MR 3D-Designer 12.25

Realistic design for the graphics professional

MR-3D Designer

MR-3D Designer is a special edition of Reinhard Epp Software's
CyberMotion 3D-Designer, exclusively licenced for MR Möbes & Ratajczak OHG.

MR Möbes & Ratajczak OHG
P. O. Box 68 02 27
30608 Hannover, Germany
http://www.mr-clipart.com

Author:
Reinhard Epp Software
Breite Straße 43
33602 Bielefeld, Germany
http://www.3d-designer.com

Special thanks to John Ridgway for translation help and to all
users helping with their beta testing and useful suggestions
on features and usability.

MR-3D Designer was programmed by Reinhard Epp
using GFA BASIC 32

Table of Contents

Part I

1 Introducing MR-3D Designer

Wherever we look, we encounter virtual worlds. Whether in Hollywood or daily advertisements, the world that is depicted is mainly an artificial creation. At last, you too can create these surreal worlds at home, practice internal or external architecture, design products or promotional logos and films or build your own virtual worlds using the built-in landscape designer and atmospheric background models. All the modules that you need to generate, model and manipulate, animate and represent the three-dimensional world are integral parts of the program. MR-3D Designer provides a high quality rendering output using raytracing and up-to-date global illumination algorithms like Ambient Occlusion and Photon Mapping. The possibilities to define object surfaces, light conditions and backgrounds are almost inexhaustible - visual libraries and fast preview windows in the main dialogs invite you to experiment and play around with this vast functionality.
Complex 3D animations can be set up very easily - you can animate all objects, camera and light settings, backgrounds, fire, water and even clouds and fog in the atmosphere. You can arrange objects in hierarchies to animate jointed models or robots and easily position joints with the help of Inverse Kinematics. With the up-to-date Skin and Bones technology even character animation is available with MR-3D Designer.

Where to start from?

A program with this vast functionality does not open up to oneself in a single day but this extensive manual will hopefully give you a good start on exploring the depths of MR-3D Designer. You should begin with the chapter "A little introduction to the world of 3D", then go on with the workspace overview and continue working through some of the tutorials.

Context-sensitive help (F1-Key)

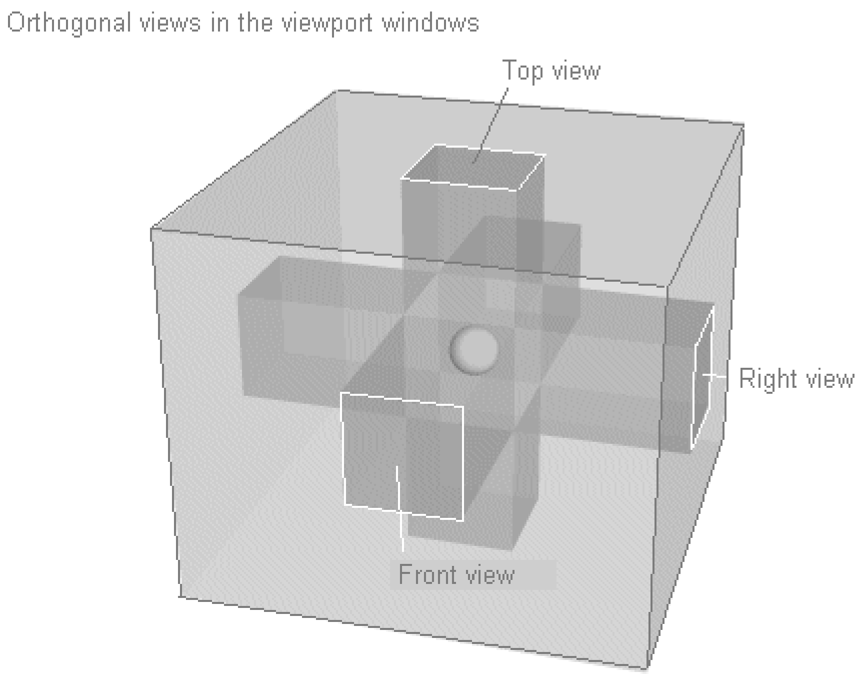
Context sensitive help can be obtained at any time pressing the F1-key. If, for example, you are in the "Rotate Object" menu and you press the F1-key, the help window with the topic "Rotate Object" opens automatically. The same goes with all open dialogs - simply press F1 and the corresponding section of the dialog automatically appears in the help window.

Part II

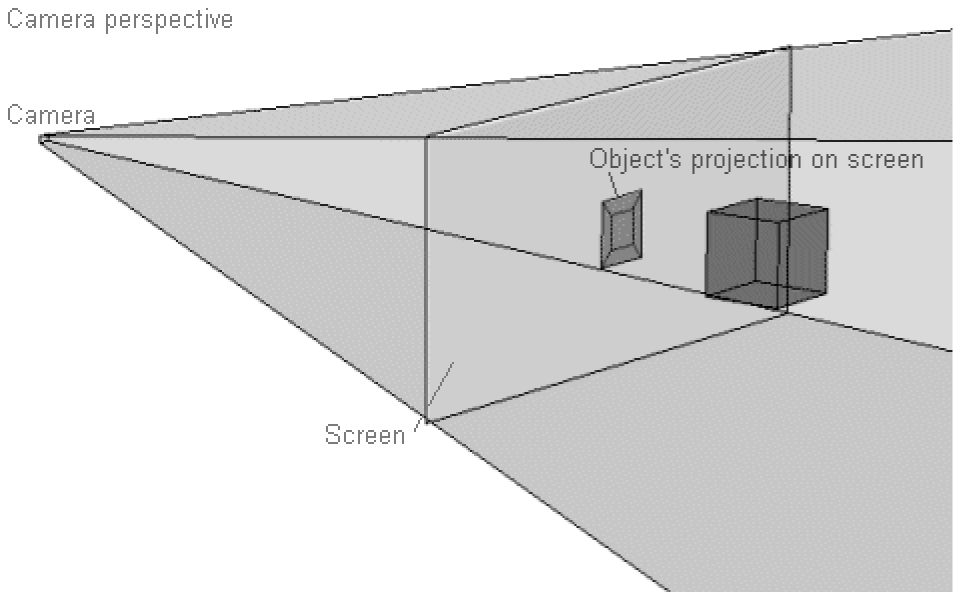
2 A Short Introduction to the World of 3D

Orthogonal View in the Viewport Window or Perspective View in the Camera Window

How do you work in 3D-space? Basically, the program distinguishes between two different types of space. Firstly, there is quasi "cubic space," in which all the objects are pictured orthogonally. Here, the objects shown in the work windows are composed of straight lines without any distortion due to perspective. While you are manipulating the objects you work in the orthogonal views and are able to view the scene from the front, back, top, bottom, right-hand or left-hand side. This, therefore, is similar to a technical drawing - without perspective.

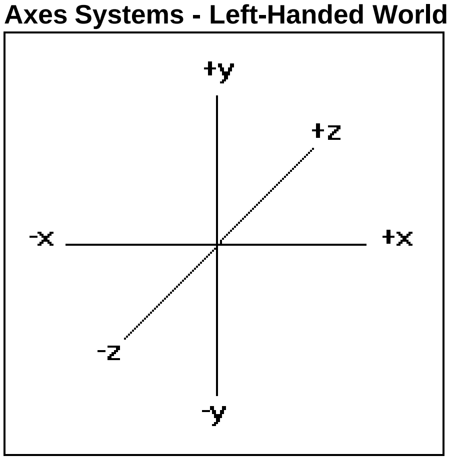


The viewport window 22 can be freely moved in the three axes directions until you reach the preset of 3D area limits, which lie at 2^{24} = 16.777.216 units. However, the window moves only to the edges of this area, so objects are always visible. The order of the X, Y and Z-axes is shown in the foregoing picture.



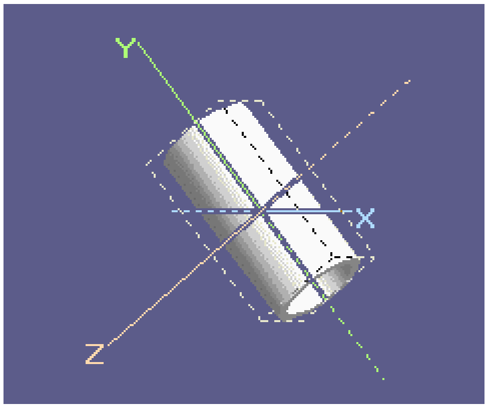
The other space is camera space. Here you can move the camera location freely. Furthermore, camera-space provides perspective, so that the objects are subjected to optical distortion. It can be thought of as a pyramid-shaped space, in which the camera lens is located at the top of the pyramid and the viewing-angle is restricted by the sidewalls of the pyramid. Within the pyramid the objects are projected onto a plane, that cuts the pyramid in front of the camera-lens.

Axes Systems - Left-Handed World Space



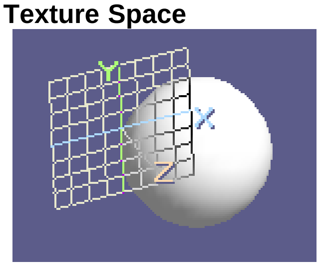
The basis of a 3D space is a Cartesian coordinate system - 3 straight lines that intersect in a single point and stand perpendicular to each other. These lines form the x-axis, the y-axis and the z-axis. Each axis consists of a positive and a negative section starting from the origin. The picture above shows the axes definition for the 3D space: left (-x) to right (+x) and top (+y) to bottom (-y). Now, the direction of the z-axis determines if the axis is called a right-handed system or a left-handed system. When the positive end of the z-axis points out of the screen the coordinate system forms a right-handed system - if it points into the screen, as it does in MR-3D Designer, it is called a left-handed system.

Object Space



Each object is equipped with an additional set of axes forming object space. These axes follow every movement of the object in world space - for instance moving and rotating with the object. You can use this axes system to scale an object along its object axes or rotate it about the axes in object space. Take a look at the cylinder in the picture above. To elongate it you simply need to scale it along its y-object axis.
Object axes are also used to define the pivot points for hierarchical 140, multisectional joints, e.g. in robot 62 constructions. Finally, all animation data is recorded in the object space in reference to the object axes system. For instance, if you move or rotate the object axes system of an already animated object, you also change the behaviour of the object in the animation.

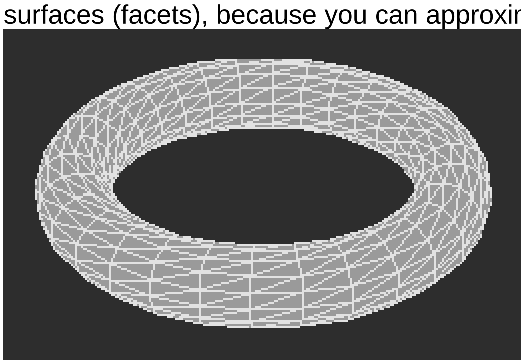
Texture Space



The texture space is used to align material 299 textures or bitmaps with an object and behaves like object space. Each object has its own texture space and the texture axes follow all object transformations, keeping the textures always in place. There are texture axes both for procedural textures as well as for each bitmap projected onto an object.

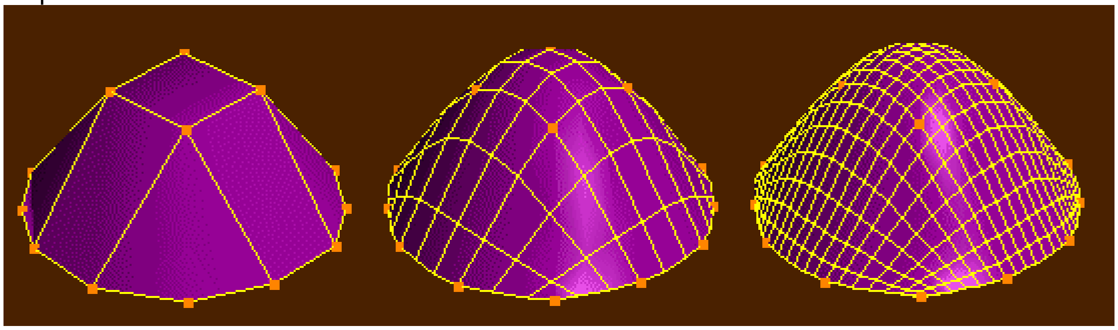
Point - We can describe the location of a point in 3D space by defining the three coordinates for the x-, y- and z-axis. The individual coordinates map the projection onto the corresponding x-, y- or z-axis.

Facet - 2 points connected together result in a line. Adding a third point that does not lie on this line results in a triangle. Almost all objects in MR-3D Designer are generated from these triangular surfaces (facets), because you can approximate almost any object's surface with them.



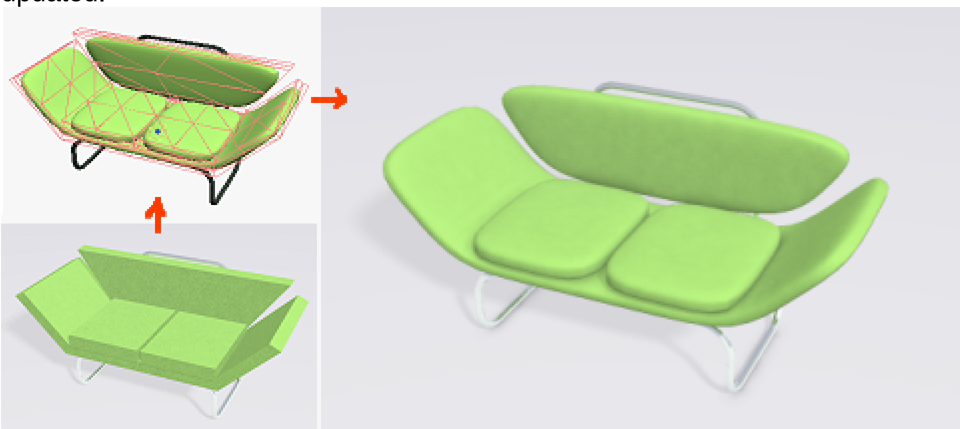
This example shows a torus object build from many small triangular facets.

NURBS - Patches - NURBS 171 stands for "Non-Uniform Rational B-Spline", a special type of deformable 3D patch. A surface is created based on a low resolution rectangular grid. The individual points of this grid represent control points used to form a surface of much higher resolution. By manipulating these control points you can very easily model smooth and organic shapes.



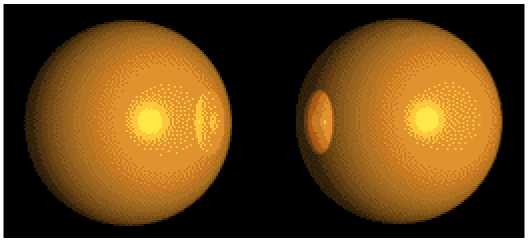
The resolution parameter defines the initial point resolution of the surface. This can be changed at any time later in the working process.

SDS - Subdivision Surfaces 163 is a real-time process which transforms a low-resolution object to a higher resolution model with an organically curved shape. Every polygon based object can be changed to a SDS model at any time in the modeling process simply by switching on or off the Subdivision Surface object property. Working with SDS is very easy. The SDS is calculated after each work step in realtime from the low resolution object hull. You cannot work on the SDS directly, you simply edit the shape of the low-polygon shape as you did before and the SDS is constantly updated.



This picture gives an impression of the effectiveness of SDS-deformation. A smoothly curved design sofa results from a few simple blocks just by activating the SDS object property for the block primitives.

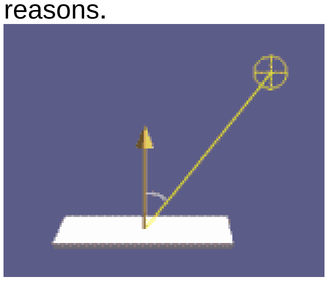
Analytical Defined Objects - There are some other basic shapes as well as triangular facets. You can, for example, create a sphere as an object that is constructed as a basic object defined only by its center and radius, instead of approximating the sphere out of hundreds or thousands of facets. When you represent this object later in raytracing-mode 264, the sphere can be calculated in a very short time, since just the basic object has to be calculated for intersection by the viewing ray. On the other hand you can not manipulate the shape of analytical defined objects, for example by deforming it, since that would destroy the mathematical description of these basic shapes.



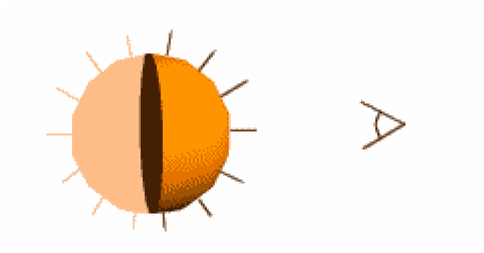
Two spheres looking just the same - on the left an analytical sphere, defined by only its center and a radius, and on the right side is a sphere constructed from 3000 facets to ensure a smooth curvature for the object.

The Surface Normal

The surface normal is a vector standing vertical to the object's surface and is important for various reasons.

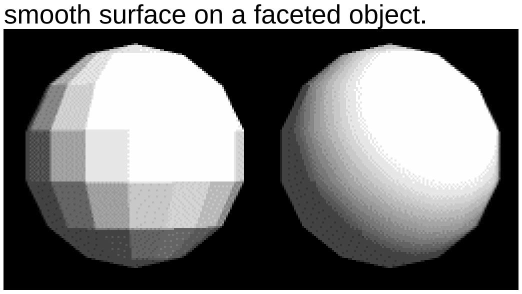


For instance, it is used to determine the surface brightness in respect of the light incidence angle.



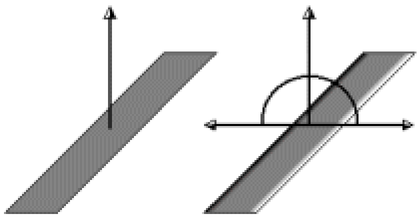
Another use for normals is to determine whether a facet has to be drawn or not. If, for example, you construct a sphere, then essentially only the front hemisphere needs to be drawn - the back half cannot be seen. On construction of an object, the facets are created so that the normals are always facing the outside. When rendering the picture, only those facets are drawn for which the facet-normal is directed towards the camera's viewpoint. The same logic applies, if you restrict bitmap textures only to the front or the back side of an object's surface.

Finally, normals can be distorted so that they no-longer stand vertical on the surface and results in distorted light intensity calculations. This is intentional and can used, for instance, to simulate a smooth surface on a faceted object.

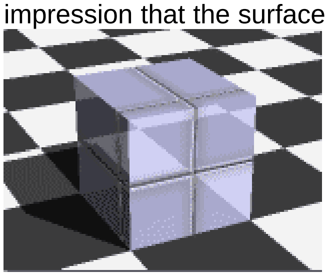


The picture above shows two identical sphere objects - except that on the right sphere was surface normal interpolation 290 switched on. The surface normals of the facet and adjacent facets are then included in the calculation of the illumination of the facets, resulting in the impression of a curved surface. This makes the object look smooth and rounded. The advantage is obvious. You can build smooth-looking objects from a smaller number of facets and save memory and time when rendering the scene.

Distorting the surface normal enables even more possibilities:



It can be used to simulate a raised appearance on the object's surface structure. If, for example, you use a stripe texture and switch on the distortion of the surface normal, this distorts the normals towards the edges of the stripe-pattern. In this way, the calculation of the light intensity creates the impression that the surface falls away at the edges of the stripe.



This example shows a tiled box. Although the box is totally smooth it appears to be constructed from several tile objects - but it is only a single object with a block texture and normal distortion assigned to the block pattern.

More examples for normal distortion are water textures 326 or the creation of irregularities in landscape textures 323.

See also

- Surface Normal Distortion 313
- Interpolation 290
- Render All Facets 290
- View - Show Normals 22
- Invert Normals 241

Part III

3 Shortcuts

Project related

F1 = Help
O + CTRL = Load project
S + CTRL = Save project
N + CTRL = Begin a new project
ALT + F4 = Quit program

Undo and redo worksteps

Z + CTRL = Undo
Y + CTRL = Redo

Start Rendering

S = Scene Rendering
S + = Scene Animation
F = Final Rendering
F + = Final Animation

Modeling <-> Animation

M = Switch to Modeling Mode
A = Switch to Animation Mode
A + = 1. Switch to Animation Mode
2. Call up Animation Dialog

Object Selection - (Un)Marking Objects or Switching Objects On or Off

- *Left Mouse Click* - Switch on and mark selected object or whole object branch
- *Left Mouse Click plus <Ctrl> key* - Add another object/branch to the current marked selection
- *Left Mouse Click plus key*
  1. Unmark object (if the object is marked)
  2. Switch off object (if the object is not marked)
- *Double Click* - Rename object

While holding the <Ctrl> or the key and the left mouse button pressed you can also drag a framework to enclose all objects you want to add to or remove from the current selection:

- *<Ctrl> key plus Framework*
  1. Switch on all inactive objects within the framework
  2. Mark all enclosed objects
- *key plus Framework*
  1. Unmark all enclosed objects

2. Switch off all enclosed objects

Let a point or facet selection float around
+, - or mouse wheel (hold wheel button pressed)

Align Viewports and Camera

V = Align viewports over selection
C = Align camera on selection

Camera Movement in Camera Viewport

- *Left Mouse Click* = Depending on camera menu settings: movement along x- and y- world axis or camera axis or circling around marked selection

- *Right Mouse Click* = Moving in viewing direction in or out
- *Right + Left Mouse Click* = Circling around selection

Switch between Work Modes

1 = Camera
2 = Move Objects and Textures
3 = Scale Objects and Textures
4 = Rotate Objects and Textures
5 = Move Viewport
6 = Edit Objects
7 = Edit Skin and Bones
8 = Deform Objects

Switch between Object and Texture Mode
(Modeling Mode, "Move-, Scale- and Rotate..."-Menus)

O = Object Mode
T = Texture Mode

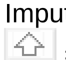
Main Dialogs

R +  = Render Options
L +  = Light Dialog
B +  = Background Dialog
M +  = Material Dialog
P +  = Object Properties
A +  = 1. Switch to Animation Mode
2. Call up Animation Dialog

and...

DEL - Delete selection
C + CTRL - Copy selection
R + CTRL - Reload Bitmaps

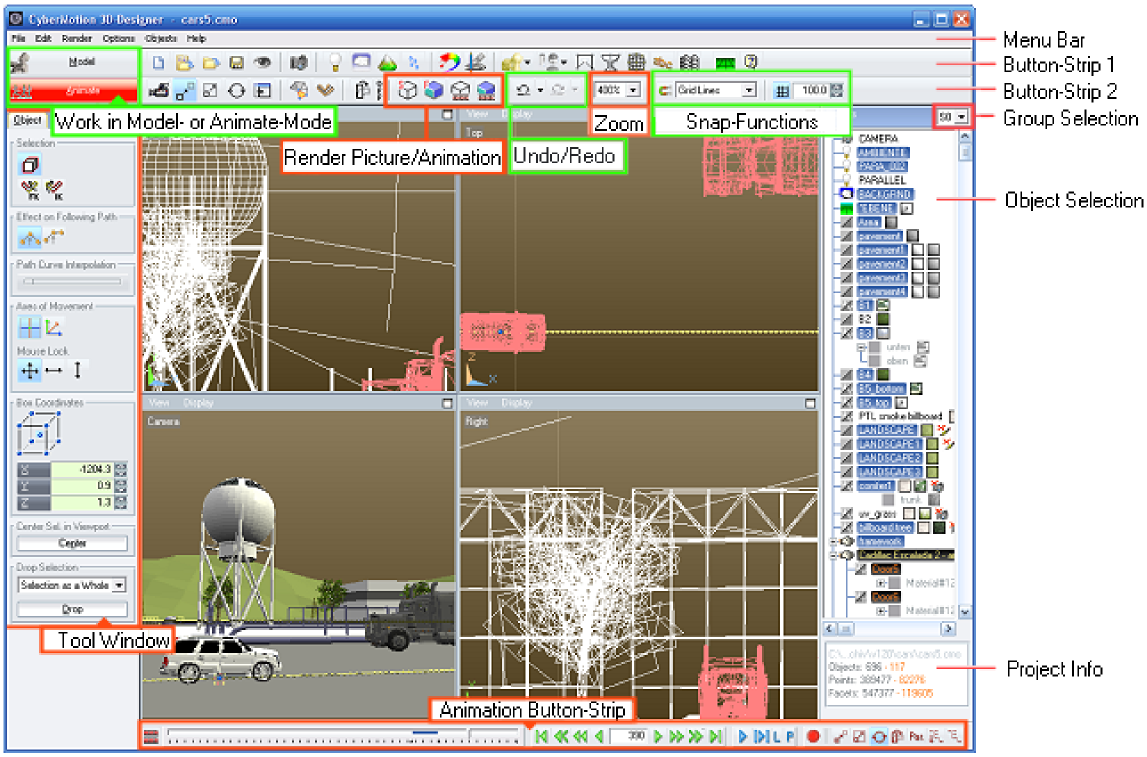
Zoom - Activate viewport and rotate mouse wheel

RETURN = Input validation
TAB = Imput validation and jump to the next edit field
TAB +  = Input validation and jump to the previous edit field

Part IV

4 Workspace, Menu Bar, Viewports, Libraries...

4.1 Workspace



When the program starts, the main window appears with the menu bar and the two button-strips 24 at the top, the tool-window to the left, the animation-button strip at the bottom and the object selection window located at the right side of the window. The main area in the center is occupied by the four depiction-windows 22 (viewports).

Switch between Modeling and Animation Mode

Work on a project is divided into Modeling Mode and Animation Mode. There are two prominent buttons at the top left corner of the MR-3D Designer window to switch between the two modes. In principle there is no great difference between Modeling and Animation Mode - you have the same tool menus for both work-modes, but some of the functions in Animation Mode are no longer accessible in Modeling Mode and vice versa. In Modeling Mode all changes made to an object - e.g. the deforming of an object by working on individual points - are permanent changes of the object's shape. In Animation Mode every action is merely a transformation of the model data and can be undone at anytime by reversing the working steps or deleting the keys that were created automatically when manipulating an object in Animation Mode.
For instance, scaling an object and its children in Modeling Mode will result in a permanent change of size of the model throughout the entire animation. If the children are deformed by this scaling this deformation will be a permanent change of the shape of the objects. However, if you scale an object and its children in Animation mode it will be only a temporary change of size. Moreover, the children will not be scaled at all - it is just their coordinate systems that are temporarily deformed by the scaling of their parents without influencing the children's object and animation data (Hierarchy Independent Animation 140).
Apart from these main differences, the working process in Modeling and Animation Mode are very similar. In both work-modes you can move, scale or rotate objects, position and align the camera and start picture rendering at any time. Detailed descriptions about the differences and restrictions, depending on whether you are working in Model or Animation Mode, are provided in the corresponding chapters of the tool menus 195.

Button-Strip 1

The first button strip, which is located directly below the menu-bar, gives you quick access to the most important dialogs, e.g. load and save projects, object-selection, lighting and picture parameters, object editors, materials, etc. These dialogs can also be found as entries in the menu bar.

Button-Strip 2

Directly below button strip 1 is a second button-strip 24, on the left of which are nine buttons that allow you to switch quickly between the different work-modes and the camera-mode. In addition to these nine buttons are 4 buttons for preview and final rendering 259, the UNDO/REDO-buttons 28, the zoom scaling, the selector box for the object-groups 133 and finally the automatic snap-functions 197 for easy positioning of objects.

Tool Window

This window is a toolbox that includes all the necessary components for editing objects. The content of this window changes depending on the work-mode. In "Rotate Object" mode, for example, there are different parameters for rotating objects, while in camera-mode all relevant data for camera positioning and alignment is available. At all times you have the facility to grab objects in the viewports with the mouse and move, scale or rotate them directly - while all corresponding parameters are updated and shown in the tool-window simultaneously . Or you can go the other way round and change the parameters in the tool window, while the scene is updated in the viewport windows.

You can change to the work-modes Camera 257, Move Objects and Textures 197, Scale Objects and Textures 206, Rotate Objects and Textures 211, Move window-detail 195, Edit objects 227, Edit Skin and Bones 242 and Deform objects 251 via the nine work-mode buttons in the second button-strip.

Object Selection Window

The Object Selection 133 window lists all objects, frozen selections and materials of the current project. Here you can select objects to work on, switch them on or off, or arrange objects in groups or hierarchies. With a right click in the window you call up a popup menu with many additional functions, for instance, to copy or delete objects or to add a marked selection of points or facets to a frozen selection.

Animation Button-Strip

The Animation button-strip 401 at the bottom of the screen is only applicable when you have switched to Animation Mode. The animation button-strip contains a slider and keyframe buttons - with which you can readily move back and forth between the individual frames of an animation. There are additional buttons to call up the animation editor, to play preview animations directly in one of the viewport windows, to record keyframes manually, and finally a further set of buttons at the right end of the button-strip to automatically create keyframes for individual tracks.

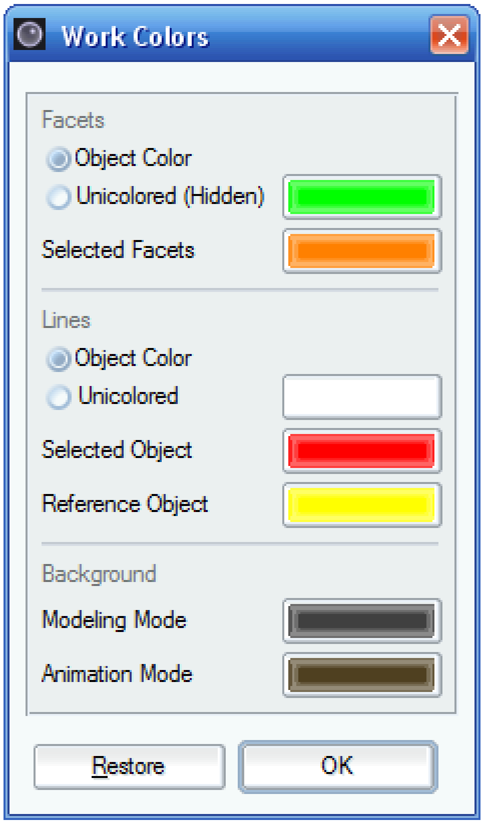
Depiction Windows - Viewports

The viewport window shows a detail of the 3D-space. You can move the window detail simply by pressing left and right mouse buttons simultaneously and moving the mouse over the viewport window. Using simple mouse-actions, you can click on objects directly in the viewport windows 138 to select, position, scale or rotate them. MR-3D Designer manages up to eleven predefined viewports sets 22. Each viewport provides its own menu bar where you can change the viewport arrangement, select an individual view
(front, top, right, back, bottom, left or camera-perspective) and a depiction type 22 (e.g. lines or flatshading).

Parameter-Inputs

Almost all parameters can be modified directly with the mouse or by the keyboard. When you have input a parameter over the keyboard, you confirm this input with a mouse-click or the return key on the keyboard. You need not be concerned about incorrect inputs - those that are outside limits are recognized and the input automatically corrected.

4.2 Work Colors



The Work Colors dialog is accessed by selecting the entry of that name under the "Options" header in the menu bar.
In this dialog you can specify the colors that are used in the wire-frame or shaded depiction modes during editing. To change a color, select the appropriate button, whereupon a color selection dialog appears, from which you can choose a new color.

Facets

- **Object Color -** When this option is checked the faces of the objects are drawn with the object's material color.
- **Unicolored** - For the unshaded hidden line depiction a single color can be defined which is used to fill all facets.
- **Selected Facets:** Selected faces of an object are drawn in this color.

Lines

- **Object Color** - The wire-frame lines are drawn with the object's material color.
- **Unicolored -** If checked, all lines are drawn with the same color you can choose here.
- **Selected Object -** The wire-frame lines of all of the objects that have been selected for editing are drawn in this color.
- **Reference Object:** The lines of objects, which serve as a reference object for the scaling,

rotation etc. of other objects, are drawn in the color indicated here.

Background

- **Modeling Mode and Animation Mod -** The background color of the view-window can be modified by this function. For a better distinction of whether you are working in Modeling- or Animation Mode you can specifiy two different background colors for the corresponding work modes.

The original colors used in the dialog can be restored by operating the <**Restore** > button.

4.3 Viewport Managment, View and Depiction in the Viewport Windows

The viewport window shows an area of 3D-space 8 through which you can freely move by using the "Move window 195" menu or by clicking into a viewport and moving the mouse while holding down left and right mouse button simultaneously. Using simple mouse-actions, you can click on objects directly in the viewport windows 138 to select them and then to position, scale or rotate them. Each viewport provides a menu bar with options to rearrange the viewports, to choose an individual view (Top, Bottom, Front, Back, Left, Right, or Camera-perspective) or depiction type (e.g. lines or flat shading).

Arranging the Viewport Windows
Use the menu entry "View - Viewport Arrangement" to select one of eleven predefined viewport sets. The viewports will be arranged automatically in combinations of two to four windows within the main window area.

Maximize or Restore Viewports
The maximize button is located at the right of each viewport's menu bar. Click on the button to resize the window so that it occupies the full working area. Click again on the button to restore the multiple viewport arrangement. An activated viewport (marked by a highlighted window frame) can also be maximized or restored using the "F11" function key.

Selecting a View for a Viewport
A view can be selected from the viewport's "View" menu. The entries "**Front**," "**Top**," "**Right**," "**Back**," "**Bottom**" and "**Left**" correspond to work-views in orthogonal drafting 8 of the relevant direction. Working direct on objects is possible only in windows with these orthogonal views. If you choose the "**Camera**" entry, the window shows the perspective view through the camera - 8 as used later when rendering the picture.

Depiction-Type in the Viewport Window

Objects can be depicted in the viewport windows in several different modes - Wireframe, Hidden line or Flat shading mode, with or without surface interpolation or textures. You can set this individually for each viewport via the "Display" entries in the viewport menu. For example, flat shading could be used for the camera-perspective, wireframe mode for all other views:

- **Grid** - Only the lines connecting surface-points are drawn (wire-frame depiction).
- **Grid - Backface Culling** - Only the lines of polygons facing the camera are drawn. (Similar to Hidden Line, but the polygons are not filled). If the object property "Render all Facets" is activated, the object is interpreted as a surface object instead of a solid, therefore all lines are drawn, because both sides of a polygon can be seen by the camera.
- **Hidden Line** - Almost all objects in MR-3D Designer are generated from triangular polygons (facets). In Hidden Line mode facets are drawn in a single object color - no illumination is applied - with correct depth information using a simple z-buffer algorithm.
- **Flat Shading** - Like Hidden Line, but with illumination of the facets.
- **Gouraud Shading** - With this algorithm objects are drawn with smoothed surfaces 290. Gouraud

Shading provides the best quality but needs longer to render. You should activate this algorithm only for the camera viewport and for scenes that are not to complex. If rendering gets to slow change back to Flat Shading or Grid depiction with backface culling.

- **Textures (uv-Mapping)** - If you switch on this option, all objects provided with uv-mapped bitmap textures can be displayed in realtime with textures in the viewport windows. Objects textured with bitmap materials using other projection modes or volumetric procedural materials are much more expensive in rendering time and will not be displayed with textures in the viewport depiction.

No Lines, Lines, All Lines

As already mentioned, in MR-3D Designer almost all objects are built from of triangular facets. However, often it is not necessary to draw all three lines to get an exact image of the object. With a quadrilateral surface, for example, which is comprised of two triangles, you need not draw the middle line. This accommodates complex objects quicker and the drawing of the scene is faster. The object editors included in MR-3D Designer support this function by retaining the necessary information for each object to obtain the best possible depiction with fewest superfluous lines.

- **No lines** - None of the lines connecting the facet points are drawn. For example, you can decide to render the polygons in Flat- or Gouraud Shading with or without drawing the lines of the polygons.
- **Lines** - Lines are drawn in accordance with the facet information.
- **All lines** - Basically, all lines connecting the facet points are drawn.

Normals

A normal 8 is a vector standing vertical to each facet of an object. The normal is important on the one hand for the visibility calculation 290 (does the facet of a solid object face the camera or not) and on the other hand for the light incidence and interpolation 290 calculations of the facets. All the normals of selected facets of an objects (in Facet Selection Mode 138) will be shown in the drawings when you switch on the "Display - Normals" entry.

Skin and Bones - Depiction in the Viewport

The following three options are available for the depiction of skeletons and skins when working on character animations:

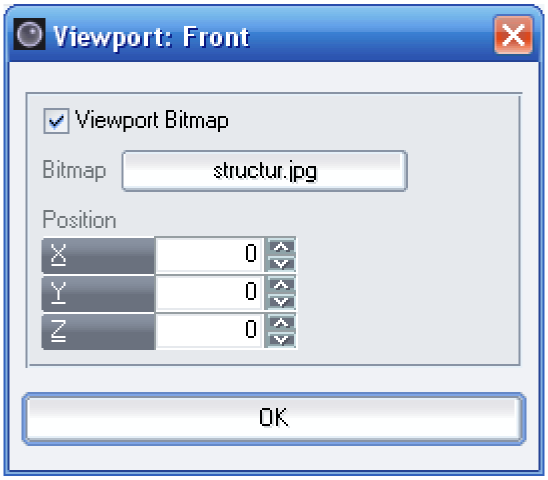
- **Bones - Transparent Skin** - In the "Edit Skin and Bones" work mode the polygons of skin objects are always drawn transparent, so that the bones within the skin can be easily recognized, selected and aligned. If you switch on the "Bones - Transparent Skin"-entry for a viewport then the polygons of skin objects are always drawn transparent, independant of the current work mode.
- **Bones - Hide Skin** - Once the skeleton is created and all skin points allocated to the individual bones, then you can accelerate setting up an animation by hiding the skin for the depiction in the viewport windows. Only the bones of the skeleton are drawn and can be easily selected and aligned in the animation.
- **Bones - Hide Bones** - This option hides the bones of a skeleton in the viewport windows. You can use this option to render fast preview animations in "Render Scene Animation 259"-mode without disturbing bones poking in and out of the skins.

Scene Background (only available for the camera view)

If the background object is switched on and you activate the "Scene Background" menu entry then a somewhat simplified version of the selected background model 374 will also be drawn in realtime into the camera viewport. For example, if an atmospheric background is chosen, then only the color range or the skymap of the background will be drawn (without clouds, fog, etc.).

Viewport Bitmap

This function copies a bitmap into the background of the viewport. The bitmap can serve as a construction plan in the Edit Object 227 work mode for adding new points or facets.

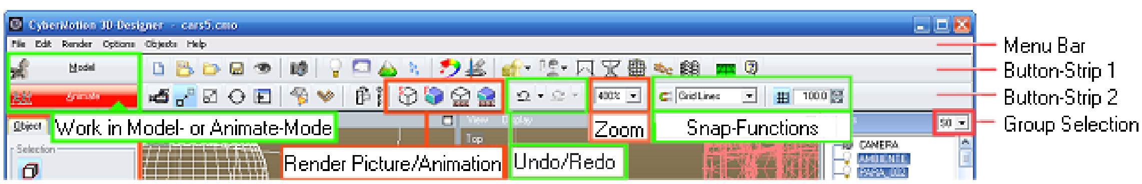


If you select the "Viewport Bitmap" entry in the menu a small dialog appears. Here you can select the desired bitmap and enter coordinates to position the bitmap in the viewport window.

Copy To All Viewports

Choosing this entry will copy the settings of the currently activated viewport to all other viewport windows.

4.4 Menu Bar and Button-Strips



In the Workspace-overview you have already seen how the MR-3D Designer window is arranged in menu- and button-strips, tool window and viewport windows. This chapter gives a detailed description of the entries of the menu bar and their corresponding icons in the button strips. In general for every important menu entry a corresponding icon is found in the button strips. Just move with the mouse over an icon and an explaining tooltip text will show up automatically. The buttons of the animation button-strip at the bottom of the screen are described separately in the chapters regarding the setting up of an animation 400.

Button-Strip 1

The first button-strip (which is located directly below the menu-bar) enables fast access to the most important dialogs, e.g. load and save projects, light and render options, object-selection, materials, object editors, etc.

Button-Strip 2

Directly below-button strip 1 is a second button-strip, on the left of which are nine buttons that allow you to switch quickly between the different work-modes and the camera-mode. In addition to these nine buttons are 4 buttons for preview and final rendering of pictures and animations 259, the Undo/Redo-buttons 28, the zoom scaling and at the right end the snap functions 197 that will help to align objects to the background grid or to the lines and points of other objects.

The entries in the menu bars and button bars:

File

Project Library 33

Load 33
Merge 33

Save 33
Save As 33
Save - Selected Objects 33

New

Show Last Rendered Picture/Animation

Customize 27

Quit

1-10 Last Project Paths 33

Edit

UNDO/REDO

Camera 257

Move Object / Texture 197

Scale Object / Texture 206

Rotate Object / Texture 211

Move Window 195

Edit Object 227

Edit Skin and Bones 242

Deform Object 251

Show Deformation 251

Show as Box

Render

Render Scene 259

Render Final 259

Render Scene Animation 259

Render Final Animation 259

Options

Render Options 263

Light 341

Background 375

Work Colors 21

Animation Editor 401

Particle System 425

Objects

Extrude 149

Sweep 158

NURBS 171

Primitives 146

Analytical Primitives 147

Landscapes 177

Plane 186

Text 174

Function 187

Billboard 190

Group Object 192

Material/Color 300

Object Properties 289

Info

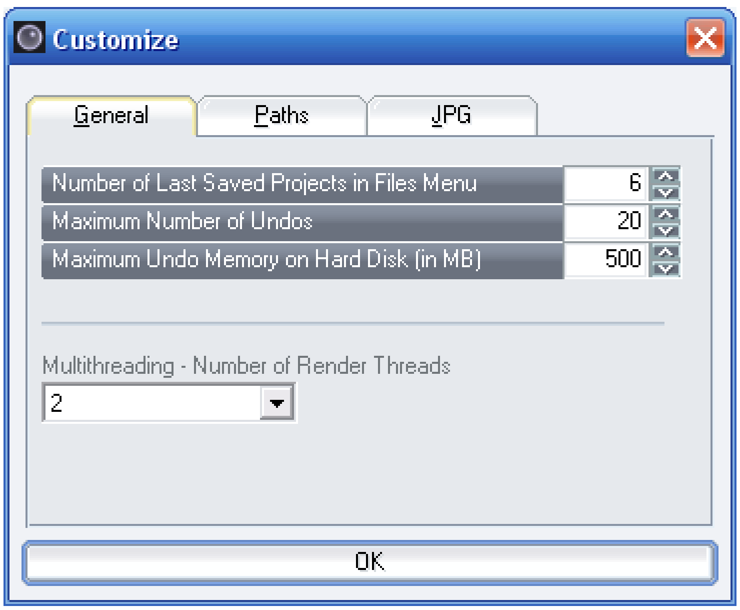
Help


Contents and Index

How to Register

About MR-3D Designer

4.5 Program Settings



Select "File - Customize" in the menu to call up a dialog in which you can define some general program settings:

General Program Settings

On the "General" page you can edit:

- **Number of last saved project files** - The files you have worked on recently are listed under the "File" menu. Here you can specify how much files you want to show up in the list (1..10).
- **Maximum Number of Undos and Undo Memory** - The maximum number of undos and redos and the maximum memory for the undo files - Each working step in MR-3D Designer is recorded and saved to temporary files. Depending on the complexity of the project and the kind of working step that has to be recorded, these files can become very large. Therefore you can limit the maximum number of files and the maximum memory used for them with these parameters. If the memory limit is exceeded, then the number of recorded undos will automatically be reduced.
- **Multithreading - Number of Render Threads** - For computers running with Multicore or Hyper-Threading processors the number of render threads occupied by MR-3D Designer can be limited. This way you can run benchmarks comparing single core with multi core processing or just save some resources for other applications.

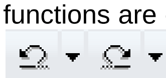
Paths for Bitmap Textures

Up to 6 different path names can be input on the "Paths" page. Simply press any of the 6 buttons and select the desired path in the Select Path dialog that then appears. The program will look under these paths later for bitmap files, which can be used to project textures onto object surfaces. You can also place your bitmaps directly in the folder that contains your project file. This way you always have the complete project saved in one folder and you can easily transfer the whole project to other computers or backup media. If the program cannot find the picture there it searches under the paths defined in the Bitmap Paths dialog.

JPG

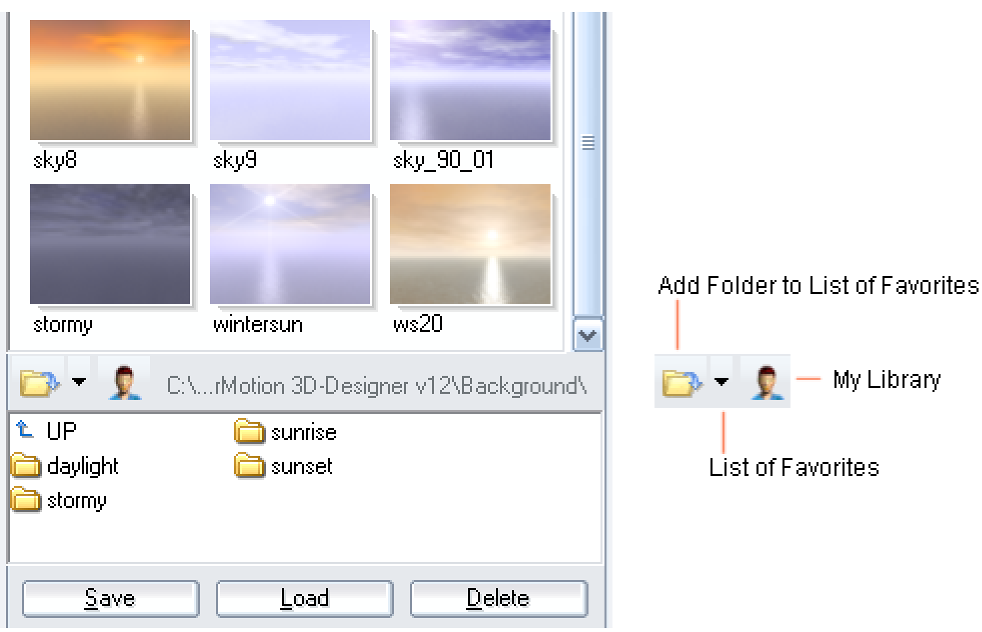
On this page you are asked for the JPEG compression rate used to compress pictures when saving them in JPEG format. Small values will yield higher picture quality but also a larger file size for the picture. High values will produce smaller files, but might introduce ugly artifacts in the rendered picture due to the lossy JPEG compression algorithm.

4.6 Undo and Redo

Undo and Redo functions are obtained over the "Undo / Redo" entries in the "Edit" menu-bar or directly from the  -buttons in the button-strip. You can also use the short cuts "Ctrl" + "Z" (Undo) and "Ctrl" + "Y" (Redo). With the Undo-function you can return to previously concluded work-steps, with the Redo-function you can repeat them again. This applies to almost all changes relevant to the project. If you, e.g., change some parameters in the light dialog, you can undo these changes right after leaving the dialog by selecting the Undo-button.
If you select the arrow buttons next to the Undo / Redo -buttons than a menu opens with a list of the last work-steps. By selecting an entry you can undo or redo several work-steps at one go.

The maximum number of operations recorded for undos and redos and the maximum amount of memory allocated for this task can be specified in the Customize-dialog 27.

4.7 Visual Libraries



The illustration shows a detail of the visual library in the background dialog.

Many of the main dialogs in MR-3D Designer contain visual libraries to provide a fast and

convenient access to already existing \projects 33, \materials 300, \backgrounds 375, \landscapes 177 or \color range 30 files. The main item of a visual library is the library window in which all available entries are represented by a thumbnail picture.
At installation time, all demo-libraries are created in the root folder of the MR-3D Designer installation. Here you can select a demo file and use it as a basis for your own projects. If you want to save or overwrite an existing demo file within the MR-3D Designer installation then keep in mind that you need administrative privileges to do so. In Windows Vista it is also required to start the software as an administrator (see Windows Vista Compatibility).
As a basic principle you should always save your newly created project files (or modified versions of the provided demo files) to your private user account or to the computers shared folders. For this purpose, on startup, MR-3D Designer creates a default library folder under each user account, for instance "C:\Documents and Settings\Username\MyDocuments\MR-3D Designer" for Windows XP or "C:\Users\Username\Documents\MR-3D Designer" on Windows Vista.'
But you can choose also any other shared directory or external media as the location for your private libraries. You can manage those locations in your list of favorites. Use the -button to add a new file to the list of favorites. Operate the arrow-button next to the folder icon to open the list of favorite collections. On initial startup there will be two entries. The path to the MR-3D Designer demo-libraries and the path to your private library created under your private user profile. This way you can easily jump to and fro between those collections. Clicking on the -button will also guide you directly to your private library.

Below the library icons is another small window listing all sub-categories of the current library folder. Simply click on the category name to open the corresponding folder or use the UP-icon to move up again in the folder hierarchy. If you want to add you own subfolders please use your Windows Explorer to create new folders in your libraries section.

Load a Library File

To take over an entry from the library simply double click on the corresponding thumbnail picture in the library window. Alternatively you can mark an entry with a simple mouse click and then press the <Load> button.

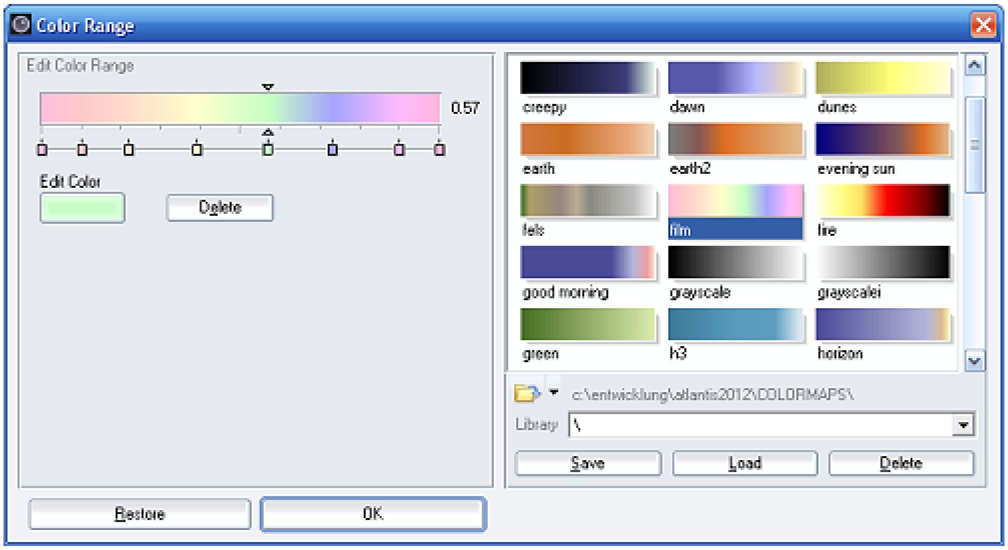
Save a File to the Library

After operating the <Save> button a file selector box appears, in which you can decide on the name for the file and the folder the file should be saved to.

Delete a Library File

First select the file you want to delete by clicking on its thumbnail picture. Then simply operate the <Delete> button.

4.8 Color Range Editor



Many functions in MR-3D Designer depend on the definition of a color range - for instance, a sky background with a color range graduating from zenith to horizon or a procedural color range texture. To edit a color range simply click on the color range bar in the corresponding dialog and the color range editor shown above appears. With it you can easily add new colors or delete existing ones. The visual library 28 on the right part of the editor provides pre-defined color ranges and the possibility to save your own creations to the library.

Edit a Color Range - Add Color Entry
A color range contains always one starting color and optional additional colors used to calculate the gradients in the color range. To add a new color, simply click on a free position in the color range. Under the color range bar is another small color button showing the current color at this position. Click on the color button to call up the windows color editor, there you can choose a matching color for the selected position in the color range. When you leave the color editor the chosen color will be inserted in the color range and a new gradient is calculated.

Edit Color Entry
Underneath the color range bar a horizontal line is displayed with small color boxes representing each color entry in the range. You can click on these small boxes to select an entry. Then simply click again on the color button beneath the line to again call up the Windows color editor where you can choose a new color for the selected entry.

Move Color Entry
Click on a color entry and move it while holding the mouse button pressed to the right or left.

Delete Color Entry
Select a color entry and simply press the <Delete> button to remove it.

Restore Color Range
Operating the <Restore> button sets the color range to the original state when the color range editor was called up.

Using the Visual Color Range Library
You can load pre-defined color ranges from the library simply by double-clicking on a thumbnail picture in the library window or save your own creations to the library by operating the <Save>

button. For a general introduction to the library functions see also: Visual Libraries [28].

4.9 Dialog Preview Options

Each of the three main dialogs for material [300], background [375] and light [341] settings contains a large preview window that provides various rendering qualities, two different resolutions and an automatic or manual update, respectively. These options are the same for all dialogs.
In addition to these basic options each preview window provides a selector box right beneath the window that offers various selections of views and object groups for the preview rendering, arranged in regard to the specific demands for the individual dialog.

Resolution - Click on the magnifying glass to change between the two possible preview picture resolutions (240 * 180 or 160 * 120 pixels).

Quality

The spheres underneath the preview window represent the six possible rendering grades for the raytraced [264] preview calculation:
1. Simple shading without shadows, reflection or transparencies in low resolution
2. Higher resolution with reflection and transparency
3. Shadow calculation is switched on
5. Activates multiple shadow sensors for soft shadows
6. Additional super-samples are calculated for antialiased depiction

Auto - If this button is activated then each change in every parameter will cause an immediate redraw of the scene in the preview window. If the **<Auto>** button is switched off you have to start a redraw manually by operating the **<Preview>** button. This is advisable when working on very complex scenes and you want to adjust several parameters in a row before starting a new preview calculation.

Stop Preview Calculation (ESC)
If a preview calculation of a very complex scene lasts to long you can interrupt the preview any time by pressing the <ESC> key on your keyboard. Of course complex scenes, for instance a landscape scene consisting out of millions of points and facets, ought to be rendered in a low grade rendering mode, preferable in raytracing mode without shadows and antialiasing.

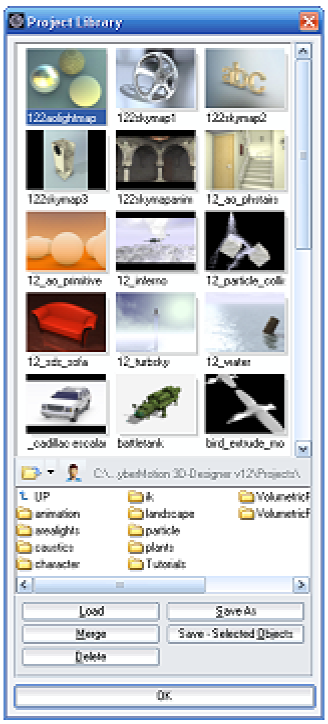
Part V

5 Projects - Import and Export

- **Project Managment** 33
  Project Library - How to load, save or merge a project file into an existing project

- **Import and Export** 34
  How to import or export foreign file formats

5.1 Load, Save and Merge Projects



Project Library 

Selecting the "File - Project Library" entry in the menu bar calls up a library window with a thumbnail list of all projects located in the current root folder of the project library. Using the library functions you can easily load, save or delete existing projects. Use the functions described in Visual Libraries to add new collection folders or to change between your existing collections.

Load, Save and Merge via the File Menu

The five file entries "Load", "Merge" "Save" "Save As" and "Save - Selected Group" call up the file selection dialog box through which projects can be loaded and saved in various forms. To load or save to 3rd party file formats you always have to use these functions - you can't save or load foreign formats with the project browser.

Load 

You can load the project-file of your choice via the file selection box. MR-3D Designer files are saved in files with the extension "**.CMO** ". For a fast acces on files you have recently worked on, the last opened files are listed also under the "File"-menu. Just click on the corresponding entry to load one of these files.

Merge
Objects are loaded and merged into a scene without deleting current objects. However, there is the following exception:
The camera, ambient light and the background are managed like all other objects in the program - which you can select and move, for example. However, there can only be one camera, one ambient light and one background. Therefore, when you select a file to merge a dialog appears in which you can choose to retain the current camera, ambient light, or background-object, or replace them with the objects included in the file. Additionaly you can choose to exclude further sun lights contained in the file.

Save  **/ Save As**
Once you have started a new project, you can save it using a suitable name by calling up the file selection box via the "File - Save" entry in the menu bar. Now, all object data, and also the settings for camera, light, background and animation data will be stored in a file with the ".CMO" extension, like "project1.cmo", for instance.
If you choose this function for a project that has already been named, it will be saved directly and without showing the file selection box.
If you want to save your project under an alternative name - to make a backup copy, for instance - you should use the "File - Save As..." menu function. This will call up the file selection box once again, allowing you to save a copy of the project under a different name.

Save - Selected Group
If you chose this entry, only those objects are saved that are selected at present in the Select Objects dialog, 133 and which, therefore, are drawn in the viewport windows. Also included are the camera, background and the light objects, if they are switched on. You should use this function when saving individual objects to library collections. Switch of all other objects, lights, the background and camera and save only the particular object designated for the library with the "Save - Selected Group" function.

Import and Export of Foreign Formats
In addition to MR-3D Designer's own object format (suffixed "* .CMO"), other foreign formats 34 can be loaded (3DS, DXF, RAW) and saved (DirectX, 3DS, DXF, RAW, VRML) with the aforementioned menu items.

5.2 Import and Export Foreign File Formats

MR-3D Designer uses its own ".CMO" object format, in which, in addition to actual geometrical data such as points and facets, a great deal of further information on object structures and materials, as well as the light, background, camera and animation settings is also saved.
In MR-3D Designer you have also the facility to save and load other foreign formats. You need only select in the file select box the corresponding extension to the relevant data format. If, e.g., you want to save a project to ".3ds" format, then use the normal "File - Save" entry in the menu bar, but before saving the project change the project extension from "project.cmo" to "project.3ds".

- Wavefront OBJ-Format (.obj) 34 - Import and export
- 3DS Format (.3ds) 34 - Import and export
- DXF Format (.dxf) 34 - Import and export
- DirectX Format (.x) 34 - Export
- VRML 2.0 Format (.wrl) 34 - Export
- RAW Format (.raw) 34 - Import and export
- Stereolithography Format (.stl) 34 - Export

Wavefront OBJ-Format

Together with the 3ds-format, the obj-format is one of the most widely-used formats for exchanging 3d files. This allows you to download not only free objects for use in MR-3D Designer from many websites on the Internet, but also objects from commercial sources. MR-3D Designer reads and writes all polygon based mesh data and basic material settings including uv-mapped bitmap textures.

3DS-Format

MR-3D Designer reads all relevant object and material data from 3DS files including uv-mapped bitmap textures. Animation data is ignored.
When exporting MR-3D Designer projects into 3DS-files take into account that 3DS can only manage objects with a maximum of 65535 points or facets, respectively. If necessary use the "Detach Object 239" function to degrade your object into several parts.
Different materials for an object and its frozen selections will be exportet properly, if at the most one basic material and one additional uv-bitmap on top of it is applied per frozen selection.

AutoCad DXF-Format

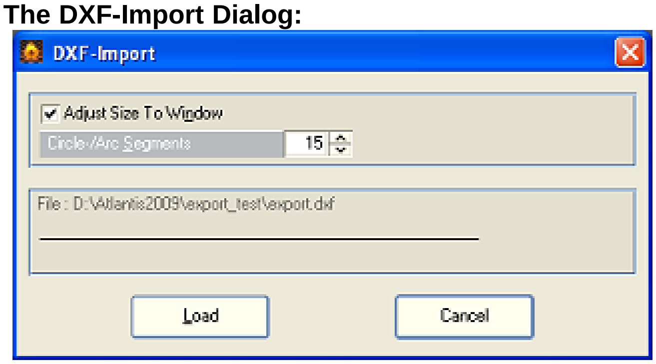
The DXF format contains a complex data configuration with a multiplicity at element-definitions. This is normally organized in different layers - to make it suitable for CAD-programs. MR-3D Designer imports most elements that have a three dimensional extent and converts them into the program's own internal object definition - based on triangular facets.

MR-3D Designer imports only ASCII DXF files without block elements**!**

DXF Elements

- high-drawn lines,
- (high-drawn) circle,
- high-drawn arc
- (high-drawn) ribbons
- (high-drawn) solids
- 3D-surfaces
- (high-drawn) (closed) (wide) 2D polylines
- approximated polygon nets
- approximated many surface nets

The DXF-Import Dialog:



In the illustration you see the dialog that appears when you want to load a DXF file.

Adjust Size to Window

If this button is selected the objects from the file are automatically scaled and adapted to the size of the viewing window.

Circle-/Arc Segments

With DXF elements such as circle, circular arcs and polylines, the circle arcs included in the DXF file are analytically defined with regard to position, radii, etc. With the Circle-/Arc Segments parameter you can determine the resolution to which these elements are broken down as MR-3D Designer objects. A high-drawn circle (e.g. a cylinder), where you have entered a segment number of 15 is subdivided into 15 segment parts (exactly as in the sweep editor). The segment number applies here, however, to an entire circle - with a segment number of 40, a circular arc of 180 degrees is subdivided into only 20 segments.

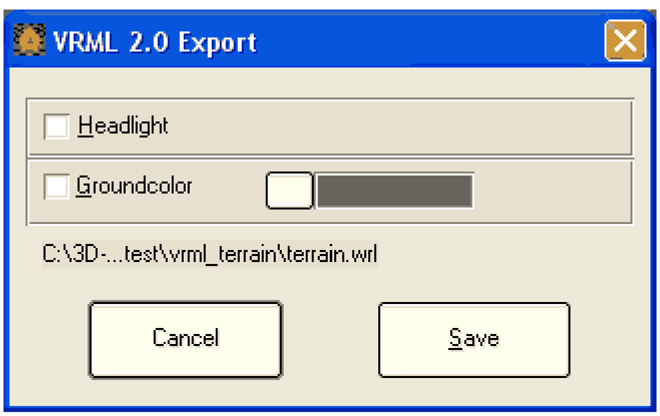
Saving MR-3D Designer objects in DXF format

If you choose "Save - All Objects 33" or "Save - Selected Objects 33" in the menu bar and in the file select box select the DXF extension, then you can export MR-3D Designer objects in DXF format. The program then generates a DXF file, in which objects are formed from 3DFACEs. A layer is included for every object (layername= object-name) defining which 3DFACEs the objects are assigned to.

DirectX-Export (Extension ".x")

This format is used by Microsoft's DirectX-3D-graphics engine. You can now export MR-3D Designer objects directly to the DirectX- object format (extension "*.x"). MR-3D Designer exports objects to 3D-meshes with regard to colors, transparency and uv-mapped bitmap textures. Different materials for an object and its frozen selections will be exportet properly, if at the most one basic material and one additional uv-bitmap on top of it is applied per frozen selection.
DirectX specifies bitmaps must be square and the size must follow the basis 2^n, e.g. 32*32, 64*64, 128*128 or 512*512. Animation data will not be saved.

VRML 2.0 Export (Extension ".wrl")



VRML-export allows the integration of MR-3D Designer projects into VRML-capable browsers. This way you can create passable virtual worlds for the internet. You can also assign URL-links (internet related addresses like http://www.3d-designer.com) to objects. Then, in a VRML-capable browser, you can change to other web pages simply by clicking on the corresponding 3D object. It is also possible to enter a link to another VRML project file (Extension '.wrl'). Clicking on the corresponding object would result in a direct jump into the next 3D world. The complexity of scenes should be kept to an absolute minimum to facilitate smooth movements in a VRML browser.
These properties will be exported:

- Objects - Color, Interpolation and Shininess will be saved. Multiple materials on different face selections are not supported, if more than one material is applied only the topmost will be exported.
- Bitmap textures - uv-mapped bitmap textures will be exported.
- Illumination - The three standard types of illumination used in MR-3D Designer are supported - lamps, spots and parallel light. Even light decrease and spot cone interpolation are known in VRML, but don't expect too much from the realization of these light types in the different VRML browsers.

- Headlight - In the export dialog you have the additional option to switch on a headlight. The headlight always points in the viewing direction while moving in a 3D world.
- Background - The 3D-sky and the fog effect can be exported to a '.wrl' file. Even all three sky gradient colors can be seen again in a VRML browser. There is no mirroring of the sky below the horizon - instead, you can specify a ground color in the export dialog. This ground color can also be used as a replacement for the plane object. Since you can't see shadows in a VRML browser, nobody will notice that the ground is just a background color (Provided you don't rotate the world.)

URL-addresses for 3D object links can be entered on the VRML-page 297 in the object properties dialog 289.

RAW-3D Format

RAW 3D objects are organized very simply. A RAW file consists only of point data (ASCI-text) for a number of triangular facets - without any other object information. MR-3D Designer objects can also be exported in RAW format - generating a file out of triangular facet data only. If you want to know more about the RAW definition just export a MR-3D Designer file in raw format and load it into a text editor. It is really very simple.

StereoLithography Export (Extension ".stl")

The .stl or stereolithography format is a file used in manufacturing. It is a simple list of the triangular surfaces that describe a computer generated solid model. This is the standard input for most rapid prototyping machines. The STL specifications require that all adjacent triangles share two common vertices. In other words, a vertex of one triangle cannot lie on the side of another. Furthermore the normals of the facets have to face outwards (facet vertexes have to be listed in counter-clockwise order - which is the usual order in MR-3D Designer) and no triangle may intersect any of the other triangles. If the model is not build according to these rules the rp-process may produce errors.

Part VI

6 Tutorials

Practise makes perfect. You should have a look at these tutorials. Once you have gone through them you will have a good idea of the capabilities of the program. However, these examples show only a fraction of the possibilities of the program. You should study the other chapters of this reference book if you want to dig out the many other possibilities the program offers.

- **Hall with Columns** 39
  The beginners tutorial
- **Facet Extrude** 51
  In a few minutes from a box to a plane model
- **Simple Animation** 55
  How to set up an animation
- **Animation and Object Hierarchies** 62
  Complex movements about several joints using object hierarchies - how to construct a robot
- **Blueprint-Modelling - Car Design** 70
  Pascal Heußner shows how to model a car from a blueprint
- **Landscape Design** 85
  Create your own worlds
- **Animation and Deformation** 93
  How to animate object deformations
- **Character Animation** 97
  How to animate characters using skins and bones
- **SDS I** 105
  Creating a designer sofa with Subdivision Surfaces
- **SDS II** 108
  Modeling faces with Subdivision Surfaces
- **Explosive Fire** 113
  Using a fire object to create an animated picture sequence of an explosion for a billboard projection
- **Smoke I** 117
  Creating dense smoke clouds with the aid of particle billboards
- **Smoke II** 121
  Creating subtle trails of smoke with the aid of 2D pixel particles
- **Particle Systems I** 125
  Spaying sparks
- **Particle Systems II** 128
  Grass-meadow with dandelions

6.1 Tutorial - Hall with Columns

This tutorial is specially written to make introduction to the program a little easier, and also to demonstrate that great pictures can be rendered with very little input. The example demonstrates mainly the modeling and arranging of a scene, no animation is involved, hence it follows that all work steps are carried out in Modeling Mode 19.

The project file for this scene can be found under "...\projects\tutorials\halle.cmo" , so you could produce a picture of the end result of this tutorial in advance.

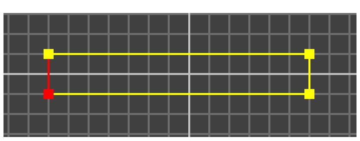
Setting Up the Objects

The picture of the hall and columns is a typical example of how a scene can be produced from only a few simple objects, using reflections, shadows and textures, which on rendering leads to an amazingly complex and realistic picture.
The object-file contains 22 objects that are built from only 5 basic shapes:

- 6 marble-columns for the hall
- 4 plinths for the columns
- 5 elements for floor, ceiling and walls.
- 4 large vases
- 3 light objects

We start with the preparation of the object-data, beginning with the production of the first column:
The column can be easily created from the analytical primitives 147 menu. Analytical objects are described just by a few parameters instead of constructing them from real points and faces - their faceted representation in the viewports is just an approximation to their real shape. When rendering later in raytracing 264 mode analytical objects are perfectly smooth and rounded and the rendering for analytical primitives is much faster as for faceted objects.
Select "Objects - Analytical Primitives >" in the menu bar and choose the cylinder icon from the sub-menu. A dialog opens with two parameters for the cylinder radius and cylinder height.
After you have entered the parameters (test data: radius 10, height 75), operate the <Create> button to create the object. Enter a relevant name for the object (e.g.:*Column1) in the dialog box that appears and confirm it with Return. As you will see, the column is now drawn in the viewport windows and the object's name printed in the Object Selection 133 window.

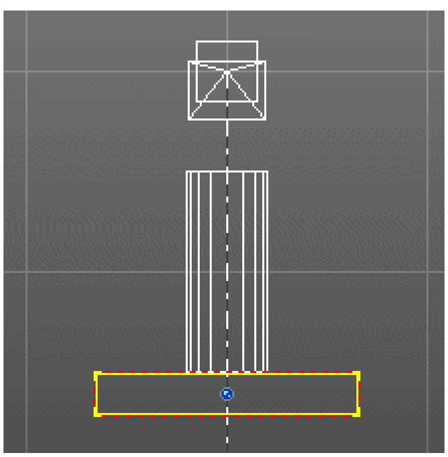
Next, we now want to set up the plinth for the row of columns:
Enter the extrude editor 149 via "Extrude" under "Objects" in the menu bar.
A new selection of buttons appears in the toolbox. Because we require only a simple rectangular base for the plinth, we can simplify the work by switching on the <Grid> and <Snap> buttons in the toolbox - points will now be constrained to a preset grid. When you move the mouse over the viewport the mouse-pointer changes to the form of a crosshairs and you can begin to draw a rectangular template in the window.



Extruding a simple rectangular shape will result in a rectangular box.

Leave the segment number on 1. The depth is unimportant, as we can later adapt the object's size to suit the columns in the "Scale Object 206" work mode.
Simply create the rectangular object by operating the "Extrude Object" <Create> button. In the dialog box that appears you can once again give an appropriate name (Plinth1) and confirm it with Return.
The large arrow-button over the viewport returns you to the main menu (or you can select the "Back to Main Menu" entry in the Options menu).
"Column1" and the "Plinth1" are now drawn in the viewports. Click onto the "Plinth1" object in the viewport window to select it for editing 138. Alternatively, you could have clicked on the "Plinth1" name entry in the Object Selection 133 window to mark the object.
In "Move Object 197" mode, a dotted-line box appears around the selected object,"Plinth1".
Click with the left mouse button into the viewport and move the plinth under the column (while still holding the left mouse button), so that the plinth stands exactly in the middle the base of the column.
There are several possibilities to help you with this:

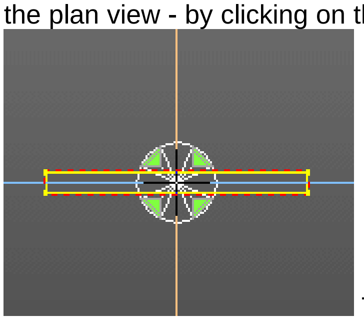
- Movement can be restricted to the vertical or horizontal direction with the three direction-buttons 197 in the tool-window.
- With help of the snap 197 functions objects are automatically catched near by the background grid lines or grid points or by the lines and points of other objects.
- You can increase or decrease the visible picture detail over the Zoom parameter in the button bar. If you have a mouse with mouse wheel at your disposal, just turn the wheel up and down to zoom in and out of a viewport window.
- You can move freely over the scene in the viewport windows when you hold down both left and right mouse buttons while moving the mouse over the viewport. In the Move Window 195 menu you only need to hold down the left mouse button. There you can also read the window coordinates or enter them directly via the keyboard,
- The last operations can be undone using the Undo 28 button or be redone using the Redo button, respectively.



The plinth and the cylinder. The object on top of the window is the camera symbol. If the camera object is disturbing you during your construction work simply switch it off by clicking on the "Camera" name in the Object Selection window while simultaneously holding the <Shift> key pressed. If an object is marked in the Object Selection window you need to click twice on the name. With the first click you unmark the object again and the second click will switch it off. Another possibility to switch off previously marked objects is to call up the pop-up menu 133 by clicking with the right mouse button into the viewport and select the "Hide" entry. The camera will still be active, but you can't select or see it anymore in the viewport. You can switch it on again in the Object Selection window with a simple left mouse click.

Now we adapt the size of the plinth to suit the cylinder:

Change to scaling mode by selecting the "Scale Object 206" entry in the Options menu bar or  in the button bar.
The plinth is still selected and crosshairs now appear exactly through the center of the object. Furthermore, the dotted-line box appears again around the plinth. The framed object can now be enlarged or reduced - again with help of the mouse. The crosshairs is the reference point of the scaling and can be moved simply by grabbing it within the 4 arrows at the crossing point. Then, when you want to scale the object, move the mouse out of the area within the 4 arrows of the crosshairs. After that hold down the left mouse-button and move the mouse to the right (left) or to the top (underneath), to enlarge (reduce) the object. The plinth is a little bit to thick, so scale down the height a little bit in "Front" view. Then activate the viewport window "View: Top" - which shows the plan view - by clicking on the viewport with the mouse.


Top view

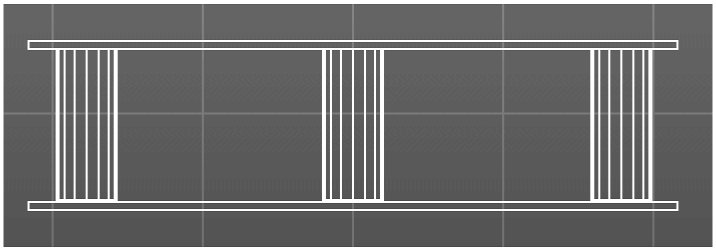
Here you can plan the plinth's enlargement in depth along the z-spatial axis 8. With the help of the mouse, you should pull the plinth to a sufficient width so that two more columns can also be placed on it. It is also advisable to enlarge the picture detail over the Zoom - function.
The scaling operations will have resulted in the plinth having moved so reposition the plinth in "Move Object 197" mode. If the cylinder is hovering above the plinth, this time choose the "Drop Selection" function to let the cylinder drop exactly on the surface of the plinth.

The bulk of the work for the hall has been completed, as the remainder is dealt with very quickly through the Copy and Move routines.
First mark the "*Column1", either with a click onto the object in a viewport window or by clicking on the name in the Object Selection windoow. Then click with the right mouse button into a window to call up the pop-up menu and choose "Copy - Normal". Repeat this work step to create a second copy of the plinth. These copies are recognizable by a "-" preceding the name. Double click on the first copy in the object list to be able to rename it and change the name to "*Column2". Change the name of the third column in the same way.

The scene appears to be unchanged as all three columns occupy the same space. In the "Right-View" viewport click once with the left mouse-button on the columns. Since they lay on top of each other a small pop up box appears listing all objects underneath the mouse pointer and so you can easily select an object from the list. Choose "*Column2". Using the mouse, move the marked column to the right, to the middle of the plinth. Restrict the movement to the horizontal direction by previously selecting the horizontal "Mouse Lock" button. The coordinates of one of the corners of the objects framework are always indicated in the tool box for exact positioning. You can stipulate the corner to be indicated just by selecting one of the corresponding corner buttons in the cube displayed in the tool window. Incidentally, the object can be precisely positioned by inputting the corner coordinates directly from the keyboard.

After you have precisely positioned the second column, you can turn to the third column to move it to the end of the base plinth. Click again once on the two columns that still overlap each other to mark the third column. After you have moved the third column to the right position, you can then select the column-center coordinates again to check the correct spacing of all the columns against each other.
Now copy an additional plinth to use at the top of the columns and place it accordingly all in the

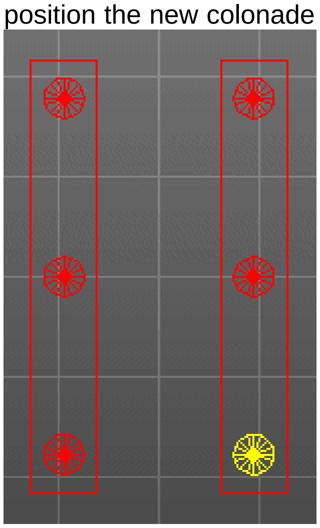
same way as you have set up the additional columns.



One row of columns with plinths at the bottom and top in the "Right" view.

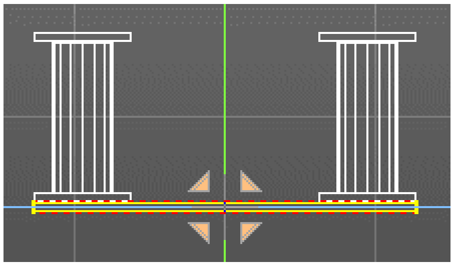
We also need three columns and two plinths now on the other side of the aisle. This is even easier as we simply copy the whole scene completely and then position the entire copy on the other side. Mark all three columns and both plinth objects with the left mouse button while simultaneously pressing the <Ctrl> button for a multiple selection (Selecting an object by clicking onto it in the viewport window or by clicking on the corresponding name in the Object Selection window is always the same. All short cuts can be used either in the viewport window or in the Object Selection window, except for switching off objects). If you move the mouse while holding the <Ctrl> key and the left mouse button pressed you can drag a framework about all objects you want to mark. Use the <Shift> key instead of the <Ctrl> key to remove individual objects from the selection.
Now choose again the "Copy - Normal" entry from the pop-up menu 133 and all marked objects are duplicated.

All copied objects will be marked automatically for work. Now, in the Front or Top view, simply position the new colonade to the right.



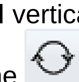
Columns and plinths centered in "Top" view

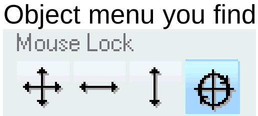
What are still missing at this time are the floor, ceiling, walls, lamps and the vases.
As we later want different materials for the walls, the floor and the ceiling we construct them from separate blocks. We have already the plinth objects with the correct length of the hall, so we just copy one of the plinths, rename it to "floor", move it down neatly under the bottom plinths and then move it horizontally to the center between the plinths. Change into the "Scale Objects" work-mode again and widen the floor, so that it fits at both ends with the end of the plinths.



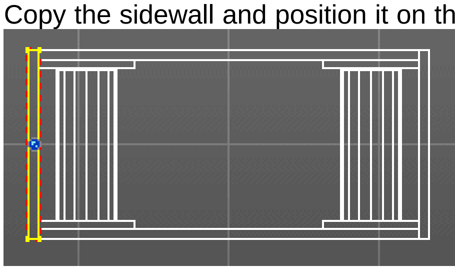
Hall with floor

After the floor has been put into place just copy it again and to move the copy as the new "Ceiling" above at the top of the columns.

Now the side walls. Again, we make a copy of the floor object. We could scale the new object now - squeeze it horizontically and stretch it vertically - to reshape the side wall. But, for practice, we choose another way. We will rotate the copied floor object about 90 degrees and move it to the side. Then we only need to scale the side wall vertically as the thickness of the wall will be identical with the thickness of the floor. Change into the -Rotate Object 211 work-mode. Now, when clicking into the viewport and moving the mouse, marked objects are rotated. We want to carry out the rotation in "Front" view about the axes standing perpendicular on the window plane. In the Rotate Object menu you find again a Mouse Lock function, that restricts movements to certain axes.

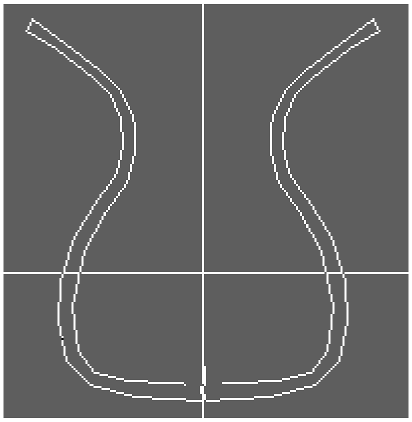


If the left button is active, objects can be freely turned about axes that are horizontal and vertical in the viewport. With the horizontal arrow button activated, objects are only rotated about the vertical viewport axes. It's much the same with the vertical arrow button - all rotations are always about the horizontal axes. But we want to rotate the wall about the axes that points directly out of the viewport window, therefore we select the last button with the circle. Now just rotate the side wall with the mouse about 90 degrees until it's standing vertical. The rotation-angles can be read at any time in the "Angle of Rotation" box. You could also have entered the angle of 90 degrees for the z-axes directly in the "Angle of Rotation" box and press the <Rotate> button afterwards.
Now, by scaling the wall vertically and moving it, you can obtain the required size and position of the sidewall.
Copy the sidewall and position it on the opposite side.



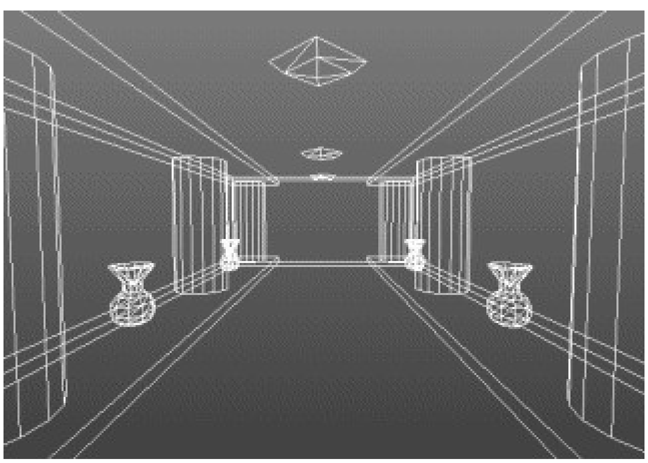
The final hall in "Front" view

Finally, we place one more wall exactly at the end of the colonnade. Later, this should create the illusion of infinite depth for the colonnade by simple reflection.



Next we want to create a number of vases for the spaces between the columns.
Go into the sweep editor 158. On the right side of the indication-surface you can produce a flat template, which when rotated about the central axis results in a sweep-object. Design a template to your liking similar to that depicted above, which, by sweeping, results in a beautiful vase. A segment-number of 12-15 should quite suffice to convey an impression of roundness later on depiction by object interpolation 290.
Generate the object and return to the main-menu.
Scale the "vase," so that it suits the size of the colonnade. Then copy it three times and position the four vases on the plinths between the six columns.
Finally we must set up three more objects that will accommodate the lamps for the walk. Go again either into the extrude or the sweep editors and construct three lamps to your taste, which approximately fit the colonnade. Fit them into the column ceiling.

 The Camera



View through the camera directly into the Hall

Next we want to position the camera. Choose the  Camera 257 button in the button-strip directly above the viewports. Aids for aligning and positioning the camera appear in the tool-window, as well as buttons for moving the camera on various axes-systems. It is the intention to move the camera to a point directly within the colonnade just in front of the first pair of columns. Positioning the camera is very flexible in MR-3D Designer:
Use the arrow-buttons beside the camera-coordinates to move the camera to the required position. The three <world-axes>, <Camera Axes> and <Circle> buttons decide if you move along the world-axes or camera-axes or circularly about an object (see also: "Camera" menu 257).

Clicking in the camera viewport window and moving the mouse is another possibility to move the visible camera detail - again along the previously chosen axes system. Holding down the <ALT>-button while moving the mouse will move the camera to the front or the rear along the world- or the cameras own z-axis, respectively. Holding down left and right mouse button simultaneously will always move the camera circularly about an object, independant of the chosen axes mode.
This way of moving the camera by clicking in the camera viewport can be done in all work-modes, not only here in the camera menu.
Then, you can go into the Move Object 197 mode again. There, the camera, like all other objects, can be selected 138 for editing and positioned with the mouse. In the "Camera" viewport window you can see directly how the camera-picture-detail changes. Back in the camera-menu you can plan a more exact adjustment of the camera-position with the arrow-buttons. You can also input the camera coordinates directly. You want, however, to capture the foyer in its entirety in the picture. This is not possible with the normal preset width of focus. Therefore, we simply put in a different width of focus with the help of the "Zoom" parameter, to achieve a wide-angle effect. In our example I have chosen a value of 23. With these settings you should see approximately the same picture in your camera viewport-window as shown in the illustration.

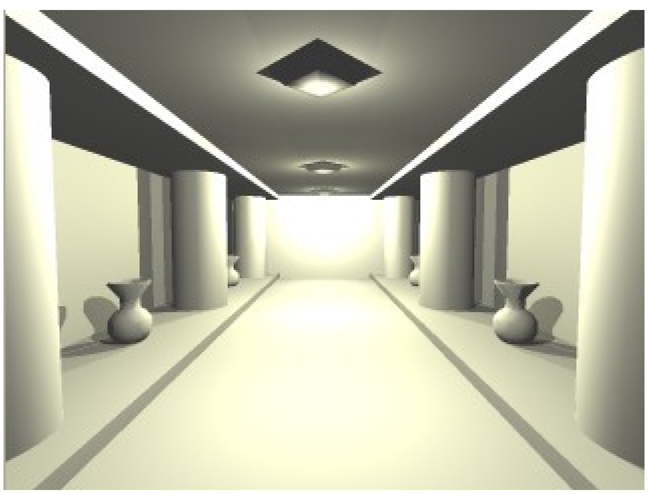
Light

The complete scene has now been set up and we can now turn to the illumination. Go into the Light dialog 341, which is reached through "Options - Light..." in the menu bar.
In the light dialog you can generate new light-objects and determine all necessary settings for the light's color, intensity and alignment. If you call up the dialog for the first time you will see that two light objects already exist in the selector box. These are the light object "AMBIENT" for the area brightness and the light object "PARALLEL", which represents a light-source with parallel light-diffusion without origin. These light-sources are generated at the program startup so that, from the start, preview pictures have basic background illumination.
For our scene we want three lamp-objects with radial light spread - as a normal lamp. The parallel light-source is superfluous for the present, so we just switch it off.
You obtain the required three lamps simple by operating the <Lamp> button in the top-left dialog field three times. The three lamps are then immediately shown in the selector-box and can be edited. Another possibility is to generate only one lamp, edit it, and then finally make 2 copies of the lamp via the "Copy - Normal" function of the pop-up menu. Light-objects are treated similarly to all other objects and can be copied as well as deleted or switched on or off in the Object Selection 133 window. When you generate a lamp a mass of new parameters for this light type are shown in the right half of the dialog. Don't worry - for our tutorial we require only the light-color and the reduction of light intensity with distance. All the other parameters on the right side are for light-effects 365 formed by lens-reflections in the simulated camera-lens and do not apply in our picture.
Now, via the Light Color button we enter the light-colors for all three lamps. The Halo Color is required for light-effects and can be ignored by us. For the lamps we enter a very bright yellow (e.g.: RGB = 255,255,220).
In addition we have to enter the light intensities 353 for the lamps. In the real world the intensity of a point light source reduces in proportional to the square of the distance, i.e. by doubling the space between the object and the light source the light intensity is reduced to a quarter. In computer graphics, however, this does not lead to satisfactory results (is there something in the real world that comes close to a real point light source?). MR-3D Designer uses a special filter to reduce the light intensity with distance and to enter an appropriate intensity for the lamp you simply have to specify a maximum radius up to the distance the light intensity will almost reduce to zero. You can enter this distance via the "Light Intensity - Maximum Range"-parameter located directly beneath the light color button. The hall is 400 units long so we enter a light intensity or maximum range, respectively, of 800 units for a first approximation. Since all 3 lamps have overlapping ranges of maximum distance we may have to reduce the intensities later on.
The area-illumination 343 in the picture must not be forgotten, because plenty of scattered radiation also originates through the many mirrored walls. Therefore select the light object "AMBIENT" in the

list box and enter a dim yellow of about R= G= 70, B= 50 for the general area brightness. But that's not enough
for our mirrored scene. Normally, Photon Mapping is used to simulate the interaction of light particles in a room, but this is a very time consuming algorithm and it is more suited for non-reflecting walls, other than our mirrored walls. In our scene all light incidence comes from above. The mirrored walls and also the light marble floor would reflect light back to the ceiling. Now we can use the parallel light again that we switched off at the beginning. Enter an inclination angle of 90 degree, so the light is directed vertically against the ceiling and switch it on again. Switch the parallel light on again and switch the shadow casting off, otherwise no light would penetrate through the floor. Enter a very low light intensity that is just enough to throw a little light back to the ceiling. Such tricks are very often used in computer industry and this special trick is called "fill lights". Even in real photography fill lights are used, e.g. during photo sessions when white boards are used to bounce spill lights back to the subject. Fill lights are also often used to produce special reflections on objects or to save time on complex shadow calculations.

We are now finished in the Light dialog 341 and can go back to the main-menu.
Here you see, there are now drawn three circles each with a cross, that represent the three light sources. These still have to be placed by moving them to the lamp-fittings. Go again to the Move Object 197 mode. There you can select and position the light objects again in exactly the same way as previously with all other objects in the scene. So that the light is not shielded by the lamp-fittings, the object-attribute "No shadows 290" has to be switched on for the lamp-fittings in the object properties 289 dialog. In addition apply a glowing material 309 effect for the lamp fitting to simulate a visible light body. Of course, the object is no real light source - it is just painted in a brighter color, the light emanates from the point light source centered within the lamp-fitting. (Note: MR-3D Designer offers the possibility to turn each object into a real light source, e.g. for area lights, but the method described above - to put a standard light into an object with material glow switched on is pretty much faster when calculating the picture.)

Adjusting light intensities - When all 3 lamps are in place we can turn back to the Light dialog to adjust the light intensities. Select the "Camera, complete scene" preview mode in the select box beneath the preview window 31. If <Auto> redraw is selected then you get immediately a first test rendering of the complete scene. On the basis of this preview picture you can adjust the light colors and intensities until you are satisfied with the illumination.



A first test rendering in raytracing-mode 264. (Start the rendering via the  "Render Final" button.)
Up to now no materials and textures are involved, but the scene looks already pretty nice.

Material

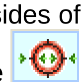
After you have dealt with the above, we can now deal with the somewhat more-complicated material adjustments:
Choose the "Material/Color" entry in the Object menu-strip to call up the Material dialog. Here you can create new materials or just choose existing ones from the materials library. Those materials can be adjusted to your needs and then allocated to the individual objects. If one material is allocated to several objects then changing this material's parameters will also change the look of all objects referencing to this material.
The Material dialog enables you to define the surface of each object using an extensive range of parameters. For instance, you can apply mathematically defined textures to generate surface structures similar to grained wood, marble, rock or multi-layered landscape textures on a fractal basis. In addition to these basic materials you can apply bitmap materials to project pictures onto an object. Furthermore up to 16 materials can be combined and freely mixed together for each object or even for each individual facet selection of an object. However, for our tutorial we will confine ourselves to the basic material properties like color and reflection or simple bitmap textures.
On the right side of the Material dialog you find the material browser. It contains two browser tabs. If the <Project> tab is activated then all materials created or loaded into the current project are listed in the browser window. At program start this is usually only one basic material. If you change to the browser's <Library> page then all external library materials will be listed in the window. Using the library functions you can transfer one or several materials into the project or, the other way round, you can save the material you are currently working on into the library.
Now change back onto the <Project> page of the browser. To select a material for editing just click on the corresponding thumbnail image in the browser window. You can also click onto the smaller material icons listed behind the object's names in the object selection window of the Materials dialog. Each time you allocate a material to an object a smaller copy of the material's thumbnail is added to the object's material list. A previously selected material icon can be removed from the object's material list simply by pressing the "Del" button of your keyboard. Attention - if no material icon of an object is selected then the currently marked material of the browser window is deleted together with all material references of objects that use this material. (Leave the dialog und use the undo-function if this happens unintentionally)

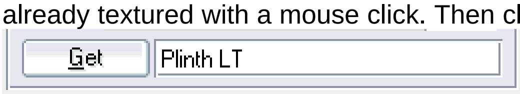
Columns, Plinths and Ceiling - For the columns and plinths we want to apply a marble bitmap-texture. To create a new bitmap material we simply click onto the corresponding <New Bitmap Material> button below the <Project> page of the material browser. A new preview thumbnail is added to the browser window and in the middle of the dialog all parameters belonging to bitmap materials appear. At the top of the page you can edit the name of the material. Change it to "Marble - Cylinder". then switch on the bitmap texture via the <Bitmap> button below the material name. Click on the button next to the <Bitmap> button to open a File-selection box in which you can select a picture to produce the texture. Select the "Boden4.tga" bitmap, a special marble bitmap suitable for our project. Thereupon the preview window of the Material dialog will be updated and the rendered picture will be copied automatically to the material's thumbnail. Now we have to choose one of the projection-types.

For the cylindrical column objects the "Cylinder" projection is correct, as you would expect. The bitmap is wrapped around the cylinder in much the same way as you would affix a label to a bottle. There are two options, "Cylinder 1" for closed cylinders and "Cylinder 2" for open cylinders with inner and outer walls. We choose "Cylinder 1". Now three icons appear next to the selection box. Here you can specify on which side of the cylinder walls the bitmap will be visible. You can choose outside, inside (camera will see the bitmap only from inside the cylinder) or both sides. For our purpose the first choice - projection on the outside wall - is the right one.
Activate now also the <Tile> button, so that the marble picture is repeated endlessly at all sides and covers the whole object.

The next step will be to allocate the material to the cylindrical column objects. Therefore, select the "Column1" object in the object selection window on the left side of the Material dialog and then just operate the <-Assing to Selection> button beneath the <Project> page of the material browser. Thereupon a small copy of the material's thumbnail is added behind the name entry of the "Column1" object. Together with the material reference an additional texture axes system has been created for the object. These texture axes allow to align materials individually on each object. For instance, by moving a texture 197 axes system you can reposition a bitmap texture on an objects surface. Or, by rotating a texture 211 axes system, you can align the direction of a bitmap projection. For example, a cylindrical projection is always projected about the y-axis of a texture axes system. The texture-bitmap includes all necessary parameters (picture-information) for colors and textures but we still need enter the shine and reflection-values for the column object. For Reflection we enter a value of about 0.25 and a highlight-value of 0.29. The reflection is important for the intensity of the highlight and the degree of reflection, while the highlight-parameter describes the radius of the reflection.
We do not switch on the mirror-button for the marble-material, however, because it could result in a confused and unappealing pattern caused by mirror reflections overlaying the texture. It is therefore only rendered with highlights. However, you might nevertheless switch on the reflection and lower the reflection share a little so as not to cover up the texture-impression.
Now that we have finished the marble material we can allocate it via the <Assign to Selection> button to each of the remaining column objects.

For the plinths we want to apply a similar material but this time with the planar "Plane" projection mode, so that the bitmap-picture is simply projected onto the front and rear of the object. Instead of creating a new material we just need to copy the "Marble - Cylinder" material by operating the <Copy> button beneath the <Project> browser. Then rename the material to "Marble - Flat" and change the projection mode to "Plane". Again a choice of icons appears to determine on which side of the object's surface the bitmap will be projected. The bitmap can be projected onto the front or backside of faces that are alternatively turned towards or turned away from the viewer. We want the picture to penetrate all sides of the object, no matter from which side the surface is viewed,

therefore we choose the  button. The direction for a planar projection is always orientated along the z-axis of the texture axes system which is created with the material reference when you allocate a material to an object. We still have to align this z-axis because on generation of the new added bitmap it will show always to the front in direction of the world z-axis. For our plinth we need the projection directed from the top downwards onto the floor space. Therefore allocate now the "Marble - Flat" material to the plinth object and then leave the dialog and change to the Rotate Object work mode again. At the top of the tool window select the "Rotate" <texture> 211 tab. If you select now the plinth object, then all materials allocated to it will be listed in the "Select Texture/Bitmap" selection box at the top of the tool window. In the "Select Texture/Bitmap" select box then choose the bitmap "Boden4.tga". Thereupon a grid appears in the viewports presenting the exact bitmap dimensions and the 3 texture axis of the bitmap. In top view you can see the z-axis alignment along the world z-axis. Now just rotate the bitmap grid with an angle of 90° about the x-axis. That's it, now the top and bottom sides of the plinth are textured with the marble bitmap. But the side walls of the plinths are not included in the projection yet. There are two solutions. You can increase the projection angle for the plane projection up to 90°. This will cover also the side walls but in the same way as it would be with a real slide projector, the parallel projection will appear as stripes along the side walls. A more elegant way is to add a second bitmap material, again the "Marble - Flat", and to rotate it in the Rotate texture mode so that it casts the projection along the x-world-axis onto the side walls of the plinth.
Actually we would have to repeat these work steps - allocating materials and aligning the texture axes - for the other three plinth objects too. But there is an easier way to copy the complete material list including the texture axes alignments over to other objects. Select the plinth object that we have already textured with a mouse click. Then click another time on the object name.



Thereupon the name of the plinth object is copied as an object reference to the <Get> field at the left bottom of the Material dialog. Now select one of the other plinth objects and then operate the <Get> button. Behind the name of the second plinth a new white icon with an arrow appears. From now on the second plinth references to the material list of the reference object. Furthermore it uses the same texture axes system as the reference object. Every change in the material list of the reference object will automatically be adopted now by all other objects referencing to that object's material list.
Click now on the white arrow icon of the second plinth object and delete it by pressing the "Del" button on your keyboard. This will delete only the object reference while simultaneously keeping the present material list and texture axes. This way you just copy the material list from one object to another. Transfer now the materials from the first plinth object over to the remaining two plinth objects.

A chechered texture for the floor
Next we want to describe the material for the floor. Instead of a bitmap we want to apply a procedural checkered texture. Operate the <New Material> button to create a procedural texture. Hereupon the bitmap parameters disappear and 4 new pages providing all parameters for procedural textures are presented in the middle of the dialog. Activate the <Material> tab. On this side you can edit the basic parameters of the new material. For the color we choose a white color, both for the diffuse and the specular reflection.
For the Reflection parameter we give a value of about 0.25 to obtain relatively weak highlights and reflections. The Highlight-value for this surface should be relatively high - about 0.76. We must switch on the Reflection-button so that it later mirrors the area in the surface. As we want to use a checkered texture here, we change to the procedural <Texture 313 > side. For the relatively simple checkered texture we choose the block pattern from the top-most dialog-field in the list box and set matching block-dimensions of (X, Y and Z= 15). Leave the Net-Width parameter on zero so that there are no spaces between blocks. Then choose a black texture color to contrast with the white material color.
Other parameters are not necessary for this relatively simple pattern, so we can allocate the material to the floor object.

Golden mirror walls
Create again a <New Material>. The sidewalls in the demo-picture are also mirrors (reflection switch on, reflection of 0.75, highlight 0.01) and have a dark golden basic color (R, G, B= about 68,51,0). The walls serve as pure mirror-surfaces in the picture and each manage without texture. Assign the new material to the side wall objects.

Back mirror wall
The material for the back wall is totally black, with a high reflection of 0.95. These are the settings for a perfect mirror reflecting all objects in front of it thus giving the illusion of an endless colonnade.

Lamp-objects - There are three point light sources directly within the lamp-objects. If the picture is rendered later with the shadow option, no light penetrates to the outside from the lamp-objects and nothing is seen. Therefore we first leave the Material dialog and change to the Object Properties dialog. There, we switch on the object attribute "No shadows" for the lamp-objects before returning again to the Material dialog. In addition, the lamp-fittings should simulate visible illumination-sources (see explanation about light settings above). We obtain that by using a self-luminosity of 0.9 for the Glow parameter and a bright yellow for the lamp's color, perfectly simulating the light object.

The material for the vases - I will leave the construction of the material for the vases to your imagination. Load a material from the library and play with the parameters. Explore the vast possibilities, you can't do anything wrong. If you want to undo your modifications just leave the dialog and press the undo button in the button bar.

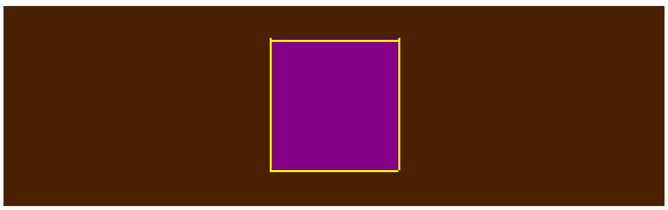
Picture Parameters

We can now set down the parameters for rendering the picture in the Render Options 263 dialog. Choose the "Render Options" entry under "Options" in the menu bar.
The scene is not very complex, thus we can start picture rendering from the beginning with the high quality raytracing 264 algorithm.
Then set the buttons for shadows and reflection in the "Raytracing" box. All effects that you choose in this box can only be rendered with the raytracing algorithm. Also switch on antialiasing 264, which improves the screen image by smoothing steps between pixels on the screen. Input a value of 1 for reflection-depth 264, which is perfect for mirroring the foyer (walls, floor and ceiling). The antialiasing default depth value of 1 is sufficient for this picture.

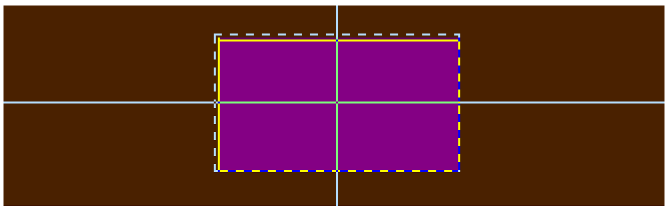
Leave the dialog and start the calculation 259 of the rendering of the first test-picture with the  <Render Final> button in the button strip above the viewports.

6.2 Tutorial - Facet Extrude

In this manual's section on "Facet Extrude 228" it is shown how to add new segments to objects simply by selecting facets and dragging them in or out of the object. You can even construct whole objects in this manner. Our example demonstrates this by modeling a plane from a simple box in no time.

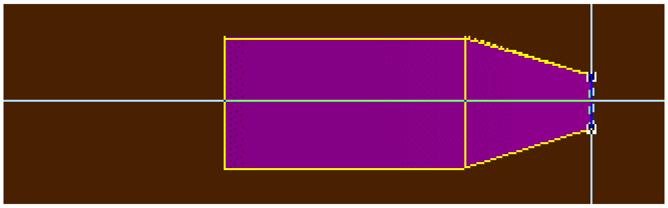


The starting point: A box generated from the primitives menu 146.

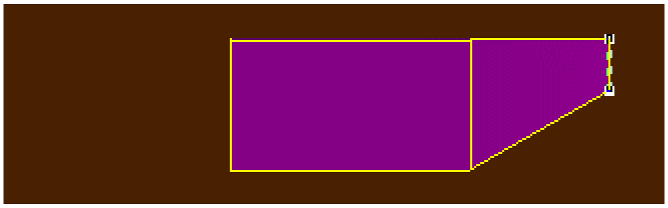


The box represents what will later be the fuselage of the plane and has to be stretched a little bit in "Scale Object 206" work mode.

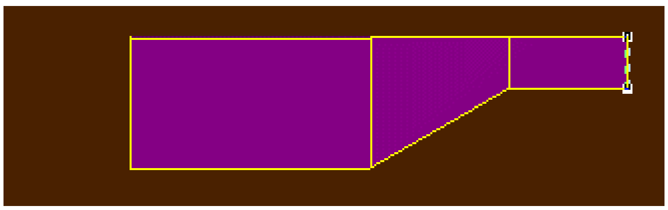
Change to "Edit Object 227" work mode. Select the facets 138 on the right side of the box and activate "Facet Extrude" mode. Now click with the left mouse button into the viewport window and drag an additional segment out of the box, while holding the left mouse button pressed.



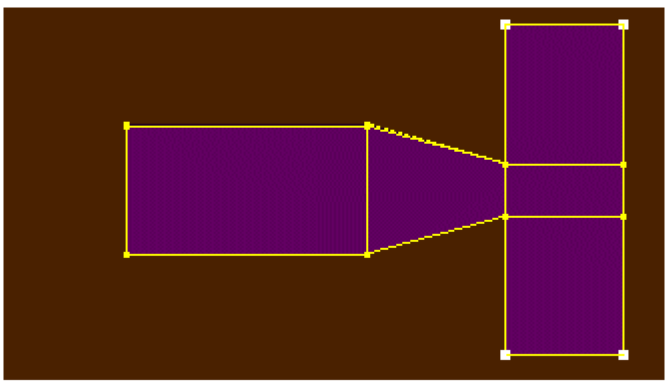
This segment forms the connection to the tail. Now scale the still selected points on the right side of the segment symmetrically (-button active) in "Scale Object 206" work mode.



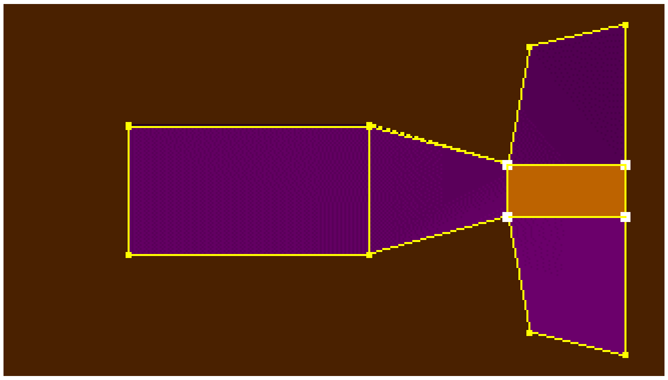
Move the point selection in "Move Object 197" work mode upwards, so that it lines up with the left segment.



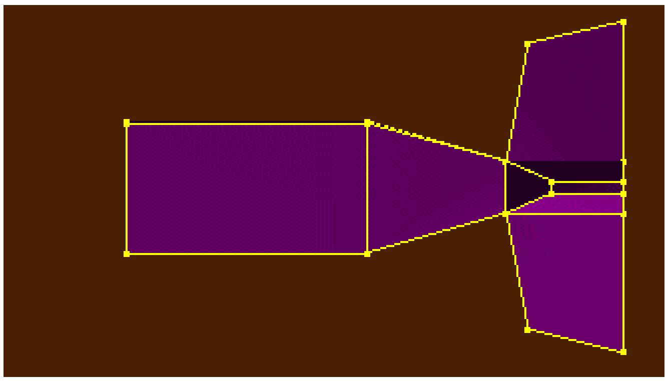
Back in "Edit Object" menu extrude the still selected facets again to the left to form the tail.



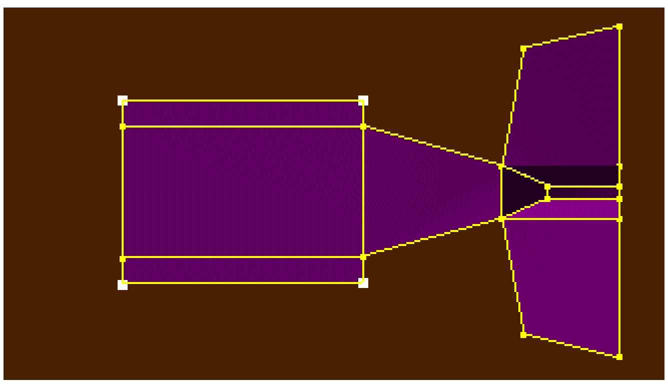
Now select the front and back facets of the new created tail segment. Then, another "Facet Extrude" operation will drag out new segments on both sides of the tail segment simultaneously, thus forming the tailfins. Scale the same point selection in the vertical direction to taper the fins at the ends. After that, deselect the points at the right end of the wings and scale the remaining points at the front horizontally to create a little beveling. Then move the points a little bit to the right until the tail resembles the illustration shown below.



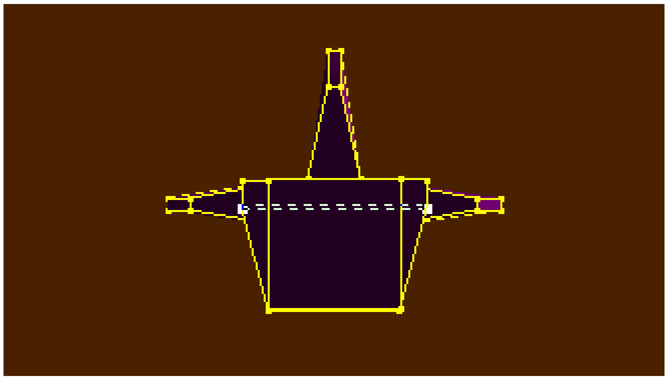
Next we select the upper facets of the tail as depicted above.



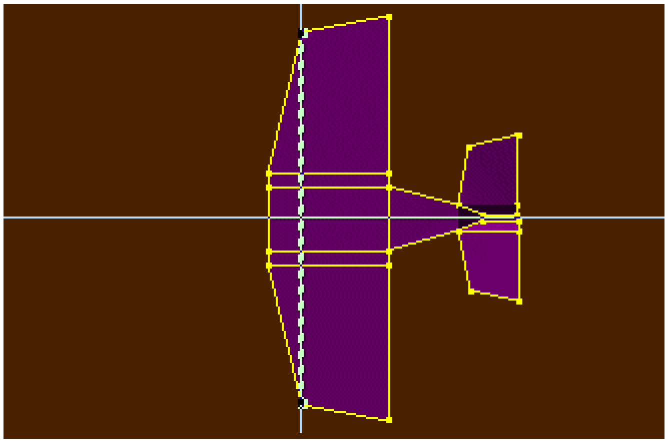
We apply again the "Facet Extrude" function to drag out the rudder segment of the plane. Scale and move again the points of the newly-created segment to taper the rudder to the top.



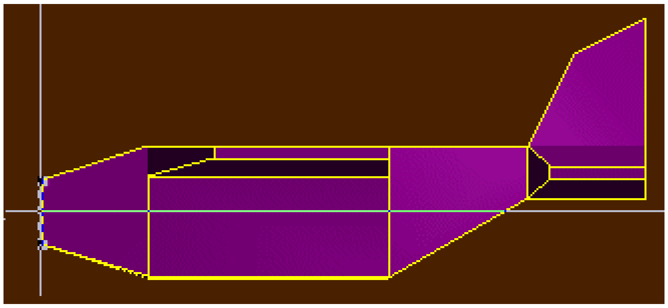
Now select the side facets of the front fuselage and drag 2 new segments out of them.



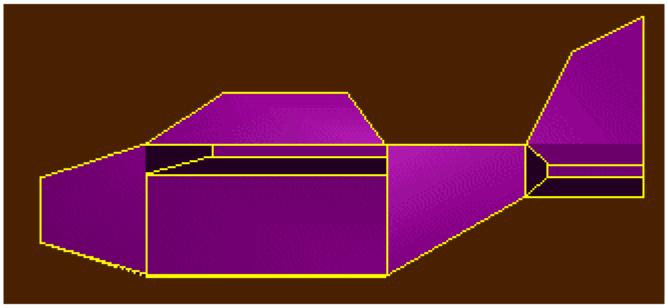
The fuselage viewed from the left. The outer, lower points of the new segments were selected and moved upwards. After that, select the facets of the new segments again and...



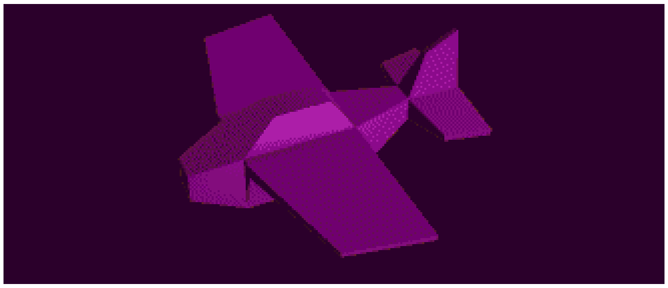
again do a "Facet Extrude" operation to drag out the side wings of the plane. Scale and move the points again to taper the wings in the same way you have done it with the tail wings.



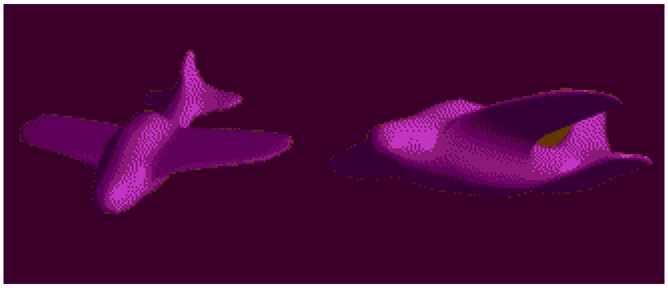
The nose of the plane is still missing, so we select the front facets of the fuselage and drag them sideways to the left. Then a symmetrical scaling of the selected points is again carried out to taper the nose.



The cockpit of the plane is produced likewise.



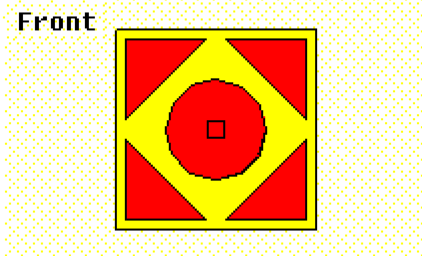
The finished plane in camera view. Somewhat bulky but quite recognizably a plane.



And this is how it looks like after applying the smooth function [238] from the "Edit Object [227]" menu twice.

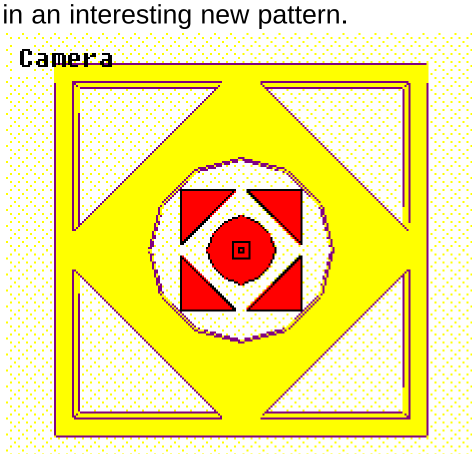
6.3 Tutorial - Simple Animation

This tutorial concentrates on the preparation of a complete animation [400] using existing objects, rather than composing and editing a scene. Load the file "anim_a.cmo" into the program. You will find the file in the folder "projects\tutorials\anm_demo\ anim_a.cmo." The file includes the objects and will serve as the starting point for the tutorial. However, if you want to make a picture of the end result beforehand, you can look at the complete ready-made object-file "projects\tutorials\anm_demo\ anim_b.cmo" in the same folder.



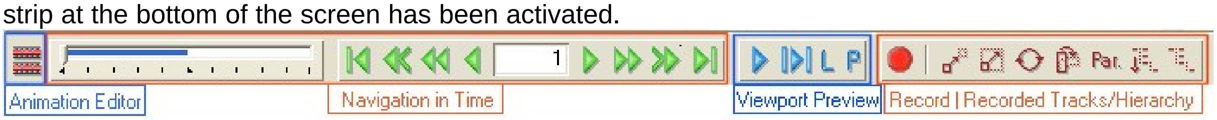
In addition to camera, light and background objects, the file "anim_a.cmo" also includes both the objects pictured above - objects "framework" and "segment". The yellow "framework" object can be set up in the extrude editor with help of the Hole <Cut out> function. The exactly-matching 5 red elements of the "segment" object can also generated in the extrude editor and combined into a single object via the Join Objects 233 function. These two objects should now serve as the starting point to set up the following animation:

At the start of the animation, the "framework" is seen on its own, rotating about its horizontal axis. The "segment" object then comes from a position lying behind and slightly below the camera viewpoint and flies into the picture, turning around its crosswise-axis. The camera approaches the objects, looking towards their slightly lower position. At its closest point it is at the same height as the objects. The "segment" object meets the "framework" object at the moment that the rotation of both objects line-up - so that they mesh together exactly, as in the above picture. However, both objects continue to rotate and the "segment" object flies on beyond the "framework". Both the "segment" and the "framework" continue to turn entirely about their axes and the animation then ends. In the Front view, in work-mode, the final frame of the animation is exactly the same picture as the start picture. In the camera-view, however, due to perspective and the setting of the zoom-parameter, in the final picture the "segment" fits exactly within the circle of the "framework", resulting in an interesting new pattern.



First we have to change from Modeling Mode to Animation Mode by pressing the 

-button in the button-strip. In the tool window some functions are hidden now or exchanged for other functions available only in Animation Mode. Furthermore the animation button-strip at the bottom of the screen has been activated.



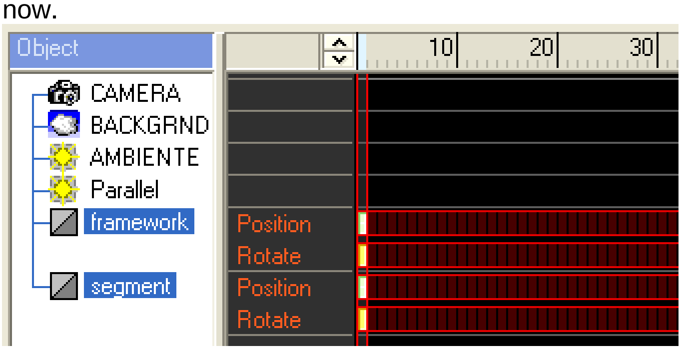
The animation button-strip - a detailed description for each button is provided in the corresponding chapter "Animation Button-Strip". Here a short summary:

- With the -button at the left you call up the animation editor in which you can edit the timelines

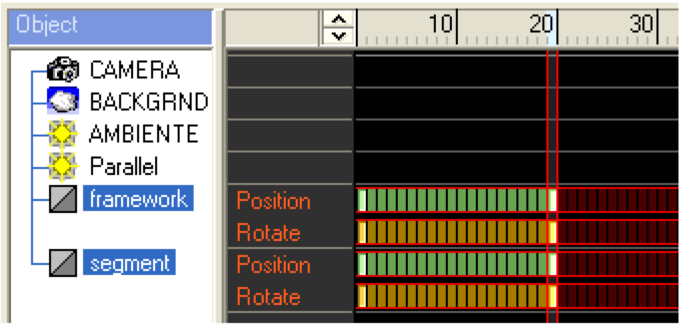
and animation tracks of the individual objects.
- Using the slider and the green navigation buttons you can move forward and back in the animation timeline.
- The blue buttons are the play-, play range-, loop- and ping pong-buttons. Use them to start preview animations directly in one of the viewport windows. Click on a viewport window to activate it before starting the preview. To stop a preview animation simple click anywhere with the mouse button or press any key button.
- The red Record button is for recording keyframes manually.
- The five following buttons behind the record button are the track buttons. Every time when you manipulate an object a keyframe is generated for all tracks that are activated here, e.g., if the rotate track is activated and you move an object, then a position keyframe is generated automatically because of the movement of the object and additionally a rotate keyframe because the rotation track was selected in the animation button-strip. This ensures a fixed position and alignment of objects in time. (See chapter on animation button-strip 401 for examples)
- The last two buttons activate the generation of keyframes for all children in a hierarchy or for the whole hierarchy up to the topmost parent, respectively.

Now let's jump in. The "framework" turns about its Y-axis, but does not change its position. The "segment" comes from the front from the negative Z-direction and flies, rotating, through the "framework". At one point "frameworks" and "segment" are at the same location and fit together exactly. This situation corresponds to how both objects have been saved on file. This situation can be used as a key-scene. So we have to create keyframes for the objects and then to move this situation forward in time, since the object should overlap in the middle of the animation and not at the beginning. Normally, keyframes are generated automatically every time you move, scale or rotate an object. But we need this scene here at this time as it is, so we have to create the keyframes manually just by pressing the -Record button. But previously we have to choose the corresponding tracks for which keyframes have to be created. In this animation objects are only moved and rotated.

Therefore we activate the corresponding buttons in the animation button-strip for position- and rotate-tracks. Then mark 138 the "framework" and "segments" objects in the viewport window. Finally just press the record button and keyframes are generated for both objects. But these keyframes hold the data we intended for a later moment in the animation. Therefore call up the animation editor now.



Here you can see the position- and rotate-tracks with the keyframes on frame 1 that we created by pressing the record button. These keyframes we copy now to a new destination point to the front of the animation. Select with a mouse click the "framework" object and then - with a mouse click and holding down the <Ctrl> button - add the "segment" object to the selection. In the timeline window choose frame 1 with a mouse click on the first frame in the timeline. Now, in the "Cut, Copy, Paste"-box select the -Copy button. All data of all marked keyframes are copied now in a temporary buffer. Now, choose frame 20 as destination point in time by clicking on frame position 20 in the timeline. Paste the copied data from the buffer to this position just by clicking on the -Paste button.

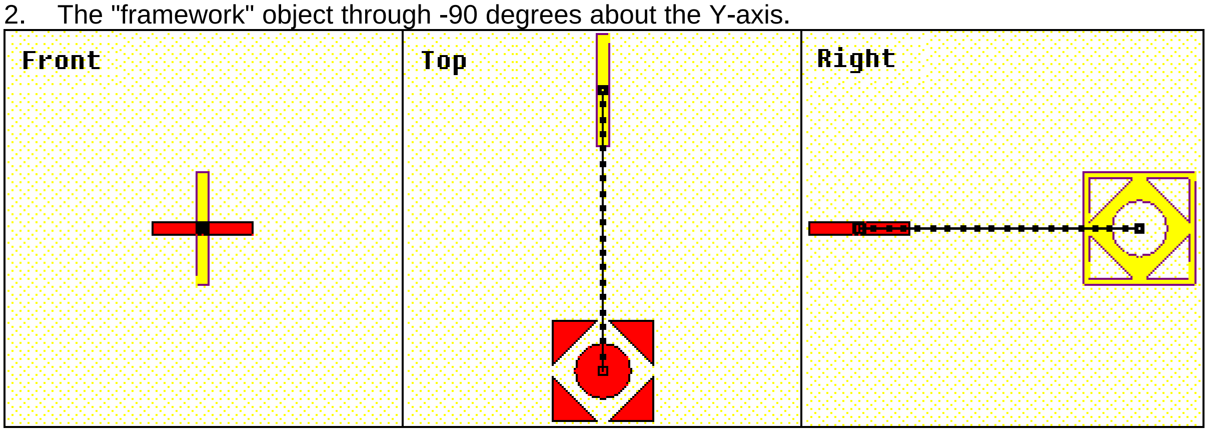


The timeline representation after the copy and paste operations.

The key data on frameposition 1 and 20 are identical now. Keyframe 20 will remain as the goal for referencing the segment with the framework. We want to change the scene in keyframe 1 to edit the approach of the "segment". We must make frame 1 the current frame and this is accomplished simply by clicking again on frame 1 with the mouse.

Back in the work-mode, select the "segment" object and move it in the negative direction of the Z-axis to a new position of -1000. For this it is best to go to the Right view. To prevent vertical movement, previously select the horizontal arrow-button in the tool-window or simply give the new Z-coordinate directly over the keyboard. After the redraw you can see movement-path of the later animation.The "segment" moves from your new position in keyframe 1 in the direction of the "framework" at key frame position 20 in 20 steps.
Now come the rotations. The special effect of the final animation come about whereby both objects rotate about different axes and then line up exactly at the moment they coincide, so that the "segment" object can fly through the "framework" object without touching it. We must change, therefore, into the "Rotate Objects" menu and rotate the individual objects about their axes in the first keyframe as follows:

1. The "segment" object through -90 degrees about the X-axis.
2. The "framework" object through -90 degrees about the Y-axis.



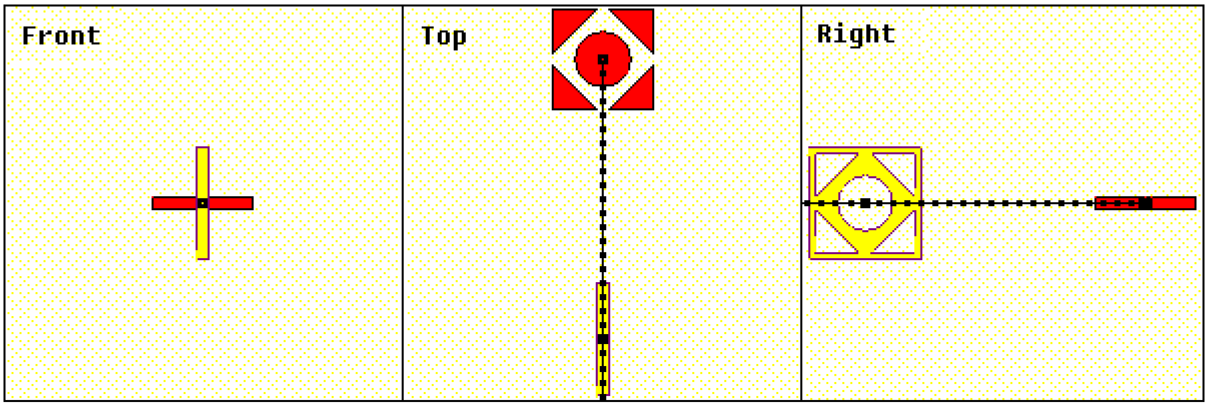
This is all that is necessary for the first keyframe. The scene should appear in the three views, as shown in the illustration above.

Now we want to turn to the third key-scene - the position after flying through the "framework".
Use the navigation-button-strip to move to the current end of the animation at frame 20. It is reached by a single click on the  -button. The next keyframe should follow 20 frames later. By operating the  - button twice - each time extending the animation by a further 10 frames - we reach frame position 40. This automatically extends the animation by 20 frames. The "segment" object is still in the same relative position to the "framework" that it occupied in the last keyframe - at frame position 20. We select the "segment" object and this time move it in the opposite direction,

(i.e. in the direction of positive Z-axis) to the position z= +1000. Because the object has changed position, a new keyframe is automatically generated at frame position 40, so we no longer need concern ourselves with the animation-editor. You can now see from the path that the "segment" moves through the "framework" on a straight line. Now we must deal with the further rotation. We therefore rotate again:

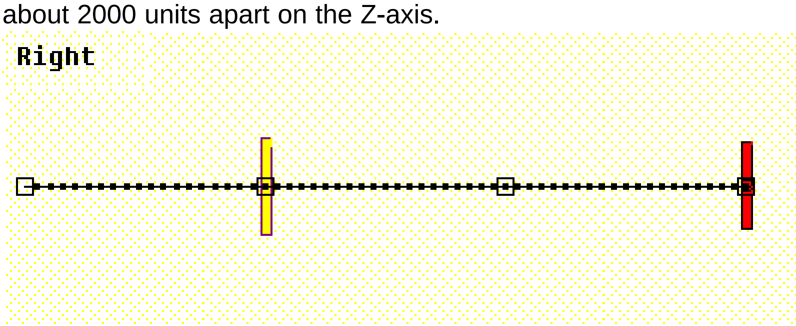
1. The "segment" object about +90 degrees about the X-axis.
2. The "frameworks" object about +90 degrees about the Y-axis.

Note that the rotation is in the opposite direction. Where we rotated through -90 degrees in keyframe 1, we now rotate about +90 degrees. This is because keyframe 1 in time is *before* our starting-situation and the present keyframe 3 refers to objects as they are in time *after* the starting-situations.



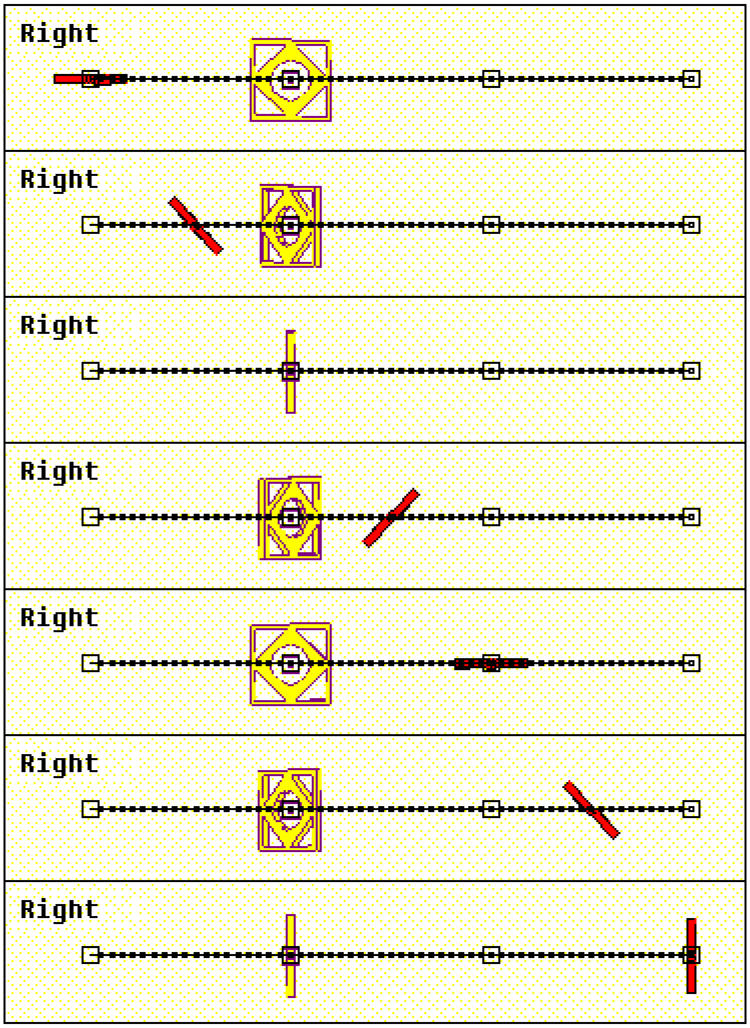
The scene representation on frame 40 after the "segment" has flown through the "framework".

Now only a forth and last keyframe for the objects remains. With it, the "segment" again moves about 1000 units further and both objects should carry out a final rotation, which brings you to the final position. Seen from in front, both objects then fit together exactly again, but now they stand about 2000 units apart on the Z-axis.



Above in the Right view you can see the goal in keyframe 4. To reach this result, exactly the same step are executed as in the third keyframe, therefore:

1. Move on a further 20 frames with the  - button to frame position 60.
2. Move the "segment" object about +1000 units in the direction of positive Z-axis. A forth keyframe is thereby automatically generated for the object.
3. Rotate the "segment" object +90 degrees about the X-axis.
4. Rotate the "frameworks" object +90 degrees about the Y-axis. Thereby a forth keyframe is also generated for this object.



The object movements in the Right view

Now that the positioning of the objects has been settled, we can move the camera and light positions. Beforehand, however, you could look again at the precise movement-path. Operate the - button, to return to the start of the animation on the first keyframe. Now choose the <Play>- button to start a Preview animation in the active viewport window. To run the animation in another viewport simply activate the corresponding window with a simple mouse click before operating the <Play> button. As the animation runs you can follow the precise movements in the different views. This is especially true for the camera-window, which shows the point of view of the animation later. The camera, however, still has to be animated and we come to that now.

Camera

Animation of the camera is accomplished quickly and simple. Change into the Camera-menu 257 and return to the start of the animation at frame position 1. In this example-animation, a simple camera-movement is implemented - from an elevated viewpoint down to the same height as the objects. Simultaneously the picture is zoomed into the scene. Only 2 keyframes are required.

1 st. keyframe

In the Camera toolbox, move the camera to the following position: x= 0, y= +500, z= 1130, using either the cursor-buttons or by inputting the coordinates directly from the keyboard. Select the "framework" object directly in the viewport with the mouse. We do not need to put in the camera-angle by hand, because, if we now operate the "Camera" <Focus> button, the camera is automatically lined up on the marked "framework" object. Now we need only adapt the enlargement. Set the "Zoom" at a value of 38 for the 1st. keyframe.

2 nd. keyframe

Move to the end of the animation - frame position 60. Move the camera down to the following position: x = 0, y = 0, z = 1130. The 2nd. keyframe is automatically generated in the animation-editor through the position change. For the camera-animation, therefore, we do not need call this a single time. Select the "framework" object again (if it has been deselected in the meantime). Operate the "Camera" <Focus> button again to line the camera up exactly on the object. Increase the picture-detail by setting "Zoom" parameter on 58.

In principle that is all that is required for the camera. We want to install one more additional effect, however. From keyframe 1, the camera turns through 90 degrees about its own longitudinal axis until it lines up exactly horizontal again. Return to the start of the animation on frame 1 and give a value of 90 degrees for the parameter for the "Camera - Roll".

Now it is time for a preview animation with a view through the camera. Activate the camera-viewport window and operate the  -button.The view through the camera-window shows the correct animation, however the visible detail may not correspond to the rendered animation, because window-size on rendering does not necessarily have the same resolution.

To open the render window and preview the animation with the correct size-relationship you can operate the  <Render Scene Animation> button in the button strip.

To produce the final True Color animation you have to operate the  <Render Final Animation> button.

Light, Background and Picture-Parameters

All lights and the background already exist in the object-file. There are two parallel light objects, whose incidence-angles and light-colors in this demo animation are not animated. You could experiment with them a little by generating a new keyframe for the light objects in frame 60. You need only move to the end of the animation, then call up the light dialog and change the parameters for the light incidence-angle or their color. A keyframe for the relevant light object is again automatically generated. The light parameter between both start and end keyframes are interpolated in the animation calculation.

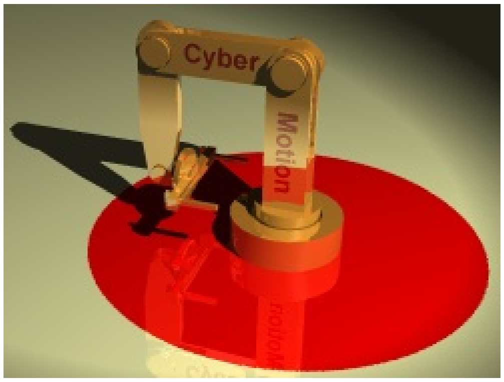
The sky can also be animated in the same manner - as you can easily try out. Choose a simple sky for the background with color path of Black to Blue.

As the scene is quite simple we can put in for picture settings everything that the program offers in the Render Options 264 dialog - switch on Raytracing, Shadow and Reflection at a recursion depth of 2 and antialiasing on step 1.

That's it. Now once again start the animation-calculation 259 .

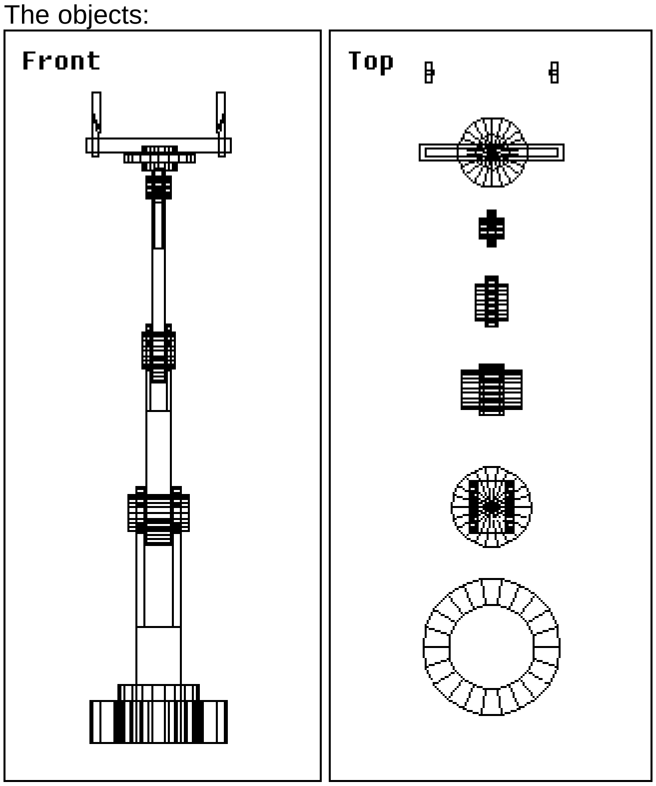
The example-animation shown here is very simple and uses only a few of the functions of MR-3D Designer. For detailed information on each function you should study the chapter on "Animation. 400 "

6.4 Tutorial - Animation and Object Hierarchies - Assembling a Robot

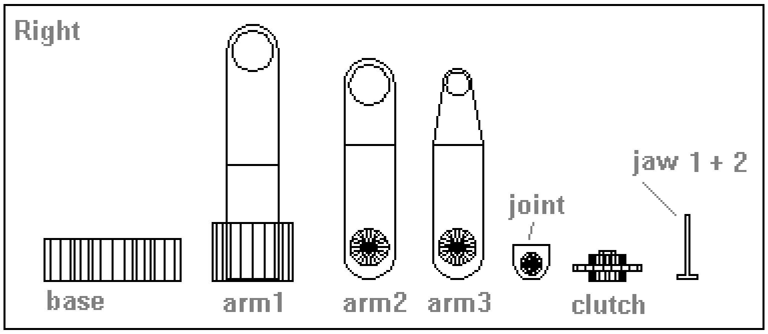


In this tutorial we set up a rather complex robot animation using object hierarchies. The object-files "robot_a.cmo" and "robot_b.cmo" for this project can be found under the directory "..\projects\tutorials\robot". The complete animation 400 defining the complex movements that the robot should execute can be set up in 10 - 20 minutes with the ready-constructed objects using the principle of object hierarchy 140. It is assumed that you are already familiar with the basic operation of MR-3D Designer and preferably have already worked through the tutorials "Hall with Columns 39" and "Simple animation 55".

The objects:



In the picture above left you can see the completed assembled robot. Load file "robot_a.cmo." This file contains the separate objects of the robot in dismantled form and serves as the start point for the tutorial. In the illustration above right you see the Top view of the separate parts.



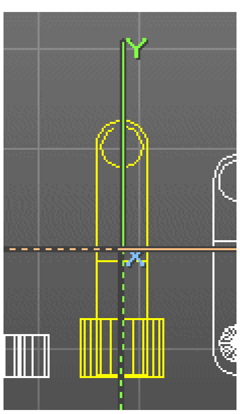
Here are all the objects again, seen in the Right view. The names of individual objects are shown to give a clearer understanding. The robot has an "arm1," which rotates in the "base" about its vertical axis and three additional joint-arms, namely "arm2," "arm3" and the object "joint." At this last joint is installed the "clutch" consisting of a pivot and a slide, which can be seen more clearly in the Top view. The slide can turn about the vertical axis of the clutch with the aid of the pivot. Lastly there are two objects "jaw1" and "jaw2." These jaws will themselves later move up and down within the clutch slide.

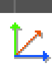
Before we go on to build the robot we must take some precautions. The principle of object hierarchy 140 is that the different lower hierarchy objects follow the movements of the higher hierarchy object. However, the lower hierarchy objects can have movements and rotations of their own. In our robot-example this happens when we turn the whole robot just by rotating "arm1" - all subordinated joints will follow this rotation - while at the same time the joints still perform further rotations about their different joints, or the jaws move in their slides. The rotations of the objects are always executed about a defined pivot point, which is identical to the focus of the object-axes.
So our first work-step in this tutorial will be to move the individual object-axes of each object to the joint position, about which the individual rotations are later executed.

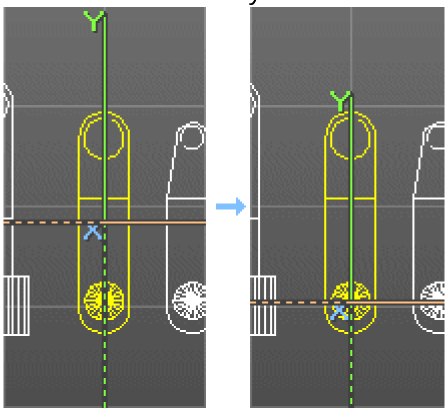
Moving the Object-Axes to Put the Pivot Point of the Objects at the Joint-Axes

The alignment of the object-axes should always take place before animating an object. The object's axes system can only be moved or rotated in Modeling Mode 19. Once an object has been animated then a later displacement of its object axes entails also an unintentional change of the objects behaviour in the animation, since all changes of position and object-angles are recorded in reference to the object-axes. So change first to Modeling Mode, if you are not already there.

Go into the "Move Object 197" mode and activate the viewport with the Right view. Select the object "arm1" with the mouse in the viewport-window.

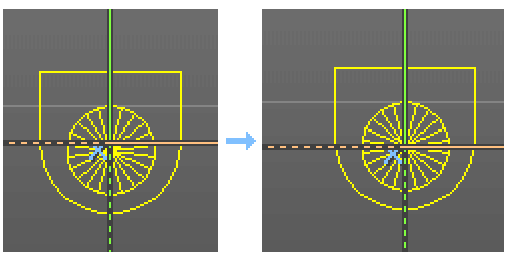


Select the  object-axes button in the "Selection" box to make the object-axes and the respective pivot point visible. In the above illustration you can see "arm1" with its object-axes. "arm1" will rotate only about the vertical axis - which here is the object's own Y-axis. Positioning the object-axes is not necessary in this case.



Change back to -object selection again and select "arm2". This arm is later connected to "arm1." The rotation of the arm follows the joint-axis at the lower end of "arm2". Switch again to axes selection, so that the object's axes can be moved again and position the axes at the joint at the lower end of "arm2" (see illustration).

Now do the same with "arm3" next to "arm2", position the pivot point of "arm3" at the joint-center at the lower end of the arm.

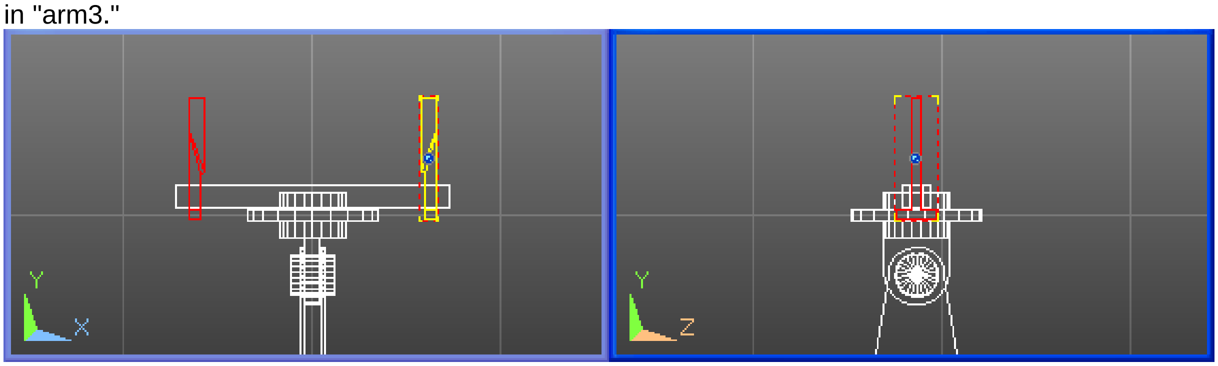


The pivot point of the "joint" object must also be moved to slightly below the mid-point of the object.

The "clutch", which is later assembled to the "joint", will only rotate about its vertical Y-axis. Repositioning the pivot point is, therefore, not necessary. The clutch- "jaws" will only move up and down along the slide. Here again, repositioning the object-axes is not necessary.

Assembling the Robot

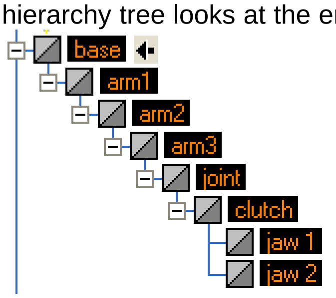
After the pivot points have been set up we can build the robot. This is quite simple. Firstly we put the "arm1" exactly into the middle of the base. This is best accomplished in the Top view. Then we change into the Right view. Position "arm2" so that its joint-axis is at the same location as the hole in "arm1", then move "arm3" to the hole in "arm2". The "joint" can now be placed relative to the hole in "arm3."



Place the "clutch" with the slide in the upper half of the "joint". Finally, there remains only the clutch- "jaws." Select both jaws simultaneous and fit them into the top of the slide. The robot is now completely assembled and looks as shown in the picture at the start of this chapter. In the foregoing illustration you can see an enlarged detail of the ready-assembled "clutch" with "joint," and the objects "jaw1" and "jaw2."

The Structure of the Hierarchy Tree

The robot is now assembled - however the individual parts are still completely independent objects. Next, therefore, we must lay down the connections between the individual objects in a hierarchy tree 140. This is dealt with in the Object Selection 133 window.   The object "base" serves us as root object, which precedes all others in the hierarchy. That is, if the "base" is later selected and moved then all the other objects are automatically moved with it. Immediately subordinate to the "base" is "arm1". Click with the left mouse button on "arm1". While holding down the button a box containing the name of the object appears. Slide the box over word base until a tool-tip indicating "Link" appears. Release the button and "arm1" is now displayed to the right side of the "base" in the window and is subordinate to that object. Next we arrange the object "arm2" under the object "arm1". Then arrange "arm3" under object "arm2," then the "joint" under "arm3" and the "clutch" under the "joint." Finally both "jaws" are subordinated to the "clutch". The complete hierarchy tree looks at the end as follows:



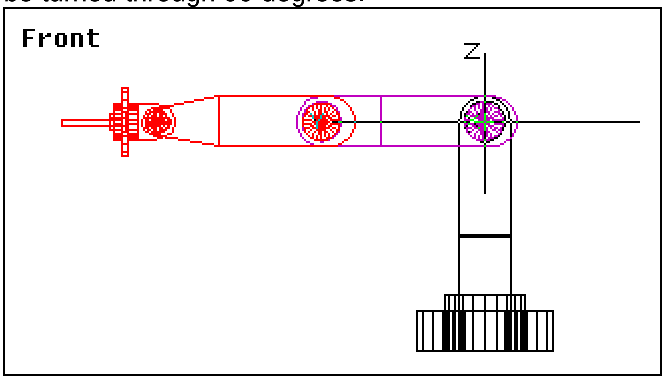
The Robot Comes to Life

In this tutorial we will carry out all movements just by selecting the individual joints and rotating them about their object axes. All children linked to the respective joint will automatically follow the rotation. This process - rotating a parent together with its children - is called Forward Kinematic 222. There is another choice of aligning a chain of joints in a hierarchy that operates the other way round - the Inverse Kinematic 222. Using Inverse Kinematics you can align a chain of multiple joints just by "pulling" at a child joint and all parent objects will automatically try to follow this movement by rotating in angle positions, that allow the child to move in the desired direction. But previously you have to define degrees of freedom (DOF) for each axes, so that the individual joints don't break out of their hinges. A demo file with initial values for the DOFs is provided under "\projects\ik\ik_robot.cmo". You can find out more about Forward- and Inverse Kinematics in the corresponding chapter refering to Kinematics 222.
Nevertheless, in this tutorial we will carry out everything with simple rotations in the Rotate work-mode - always rotating a parent with its children - first, to exercise important working methods, and second, because of the fact that when several joints are involved in the movement at the same time, the result often doesn't comes out as planned. Now and then you will have to correct arm positions with Forward Kinematics anyway. Even the makers of the great 3D-animation films often do without any Inverse Kinematics in order to maintain full control of the motions.

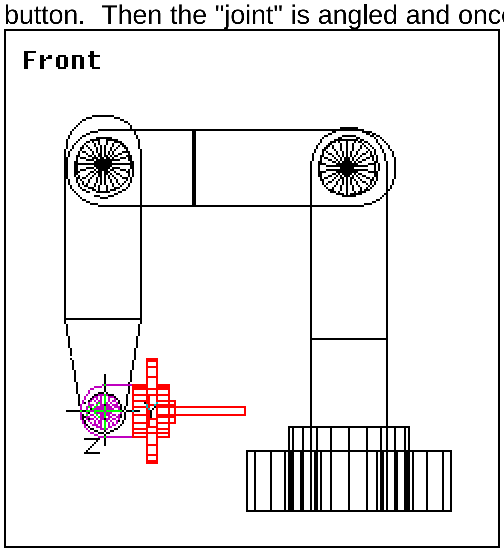
Bringing the robot to life after it has been assembled and connected together hierarchically is not difficult. Firstly, we want to bring the robot into a parked position, from which the animation should start. In the start position all the joints in the robot are turned through 90 degrees and are seen from the side in the Front view.

Go into the "Rotate Object 211" work-mode and choose the Right view. Select the "arm1" with the mouse. All subordinate objects down to the jaws are automatically selected with it. In the "Axes of Rotation"-box select the  -object-axes for the rotation axis and give a Y- angle of 90 degrees for

the rotation. Operate the <Rotate> button to initiate the rotation. Of couse you can also carry out this rotation with the mouse directly in the viewport. Select the Y-axis in the "Axes of Rotation"-box and then click in the viewport and move the mouse while holding the left mouse button pressed. The angle about which the object is rotated can be read again in the "Axes of Rotation"-box.
In the Front view you can see the robot now from the side. Now all joints of the chain have only to be turned through 90 degrees.



This is quickly dealt with. Select the object "arm2" with the mouse, enter 90 degrees for the joint angle about the X-object-axis (assuming the old angle-input has previously been deleted with <Clear>) and confirms the rotation with the <Rotate> button. The result is pictured above. Select "arm3." This also is turned about the same angle, and you again need only press the <Rotate> button. Then the "joint" is angled and once more the rotation confirmed by <Rotate>.



The robot now stands in its parked starting position.

Now the work of animation 400 begins. Change now into Animation Mode by selecting the corresponding 

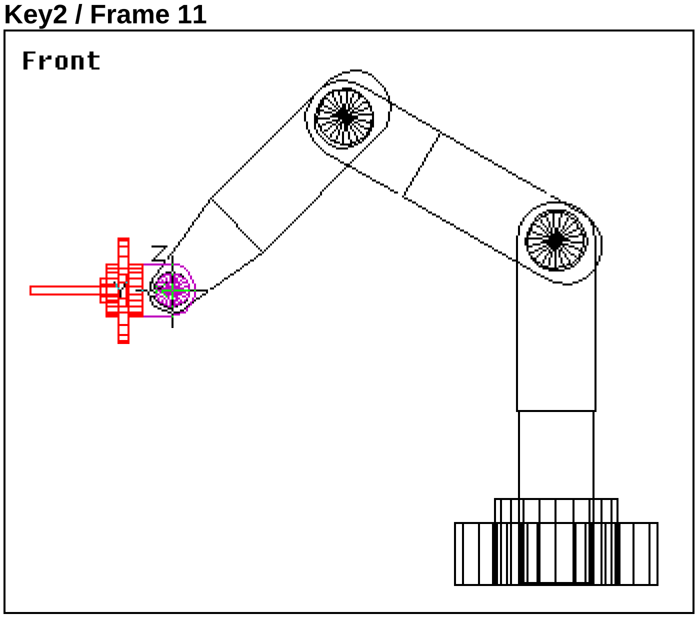
-button.

When you select an object in a hierarchy and rotate it then in the animation editor automatically a rotate track with a corresponding rotate key holding the axes alignment and rotation angles in this frame position will be created. However, this key will only be created for the particular parent object you selected in the hierarchy but not for its child objects - they will follow automatically their parent's movements. This saves a lot of keyframes and is useful in many situations, but for our robot animation this is more a handicap. An example: You move the robot arms about several rotation steps in a destination position. Once the arms have approached this position the jaws are supposed to close in to grab an object. Therefore you move forward in your animation and then move the jaws within their slides to meet together in the middle. Now, when you play the animation you realize, that the jaws already start to move in their slides from the beginning of the animation instead of from that moment in time, when the roboter arms have reached their destination point. Since you moved the jaw objects only at the end of the animation, only at this moment keyframes were generated for the jaws. The time-sequence for the jaw-movement is therefore defined by their initial start position up to the keyframes generated at the end of the animation. But we want that on every keyframe position all joints and the clutch with its jaws will hold exactly those positions that we arranged for them in that particular keyframes. Therefore we have to activate in the animation button-strip the automatic creation of keys for all objects of a hierarchy. This is the last -button at the right side in the animation button-stripe. Furthermore, always position- and rotate-keys are to be created jointly, even if an object is only moved or only rotated.

Therefore choose also the corresponding track buttons in the animation button-strip.

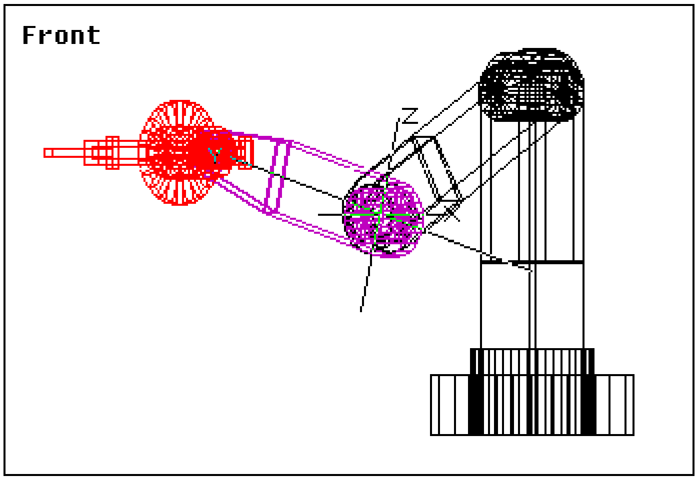
Animating the roboter:

Key2 / Frame 11



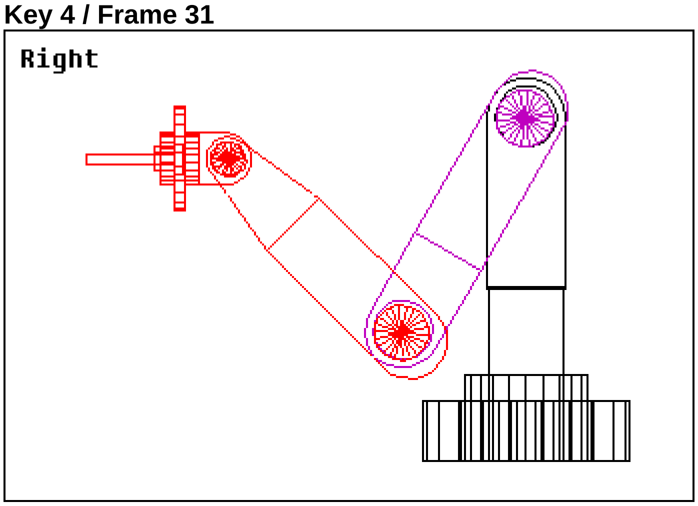
Press the  button once with the mouse to insert a further 10 frames at the start of the animation.
The individual rotation-steps:
Select the relevant object and then make the rotations about the relevant axes, as listed. The selected object is always rotated about the object's axes.
arm2 X - axis + 30 degrees
arm3 X - axis + 30 degrees
joint X - axis + 120 degrees
The result of these actions is seen in the foregoing illustration.

Key3 / Frame 21



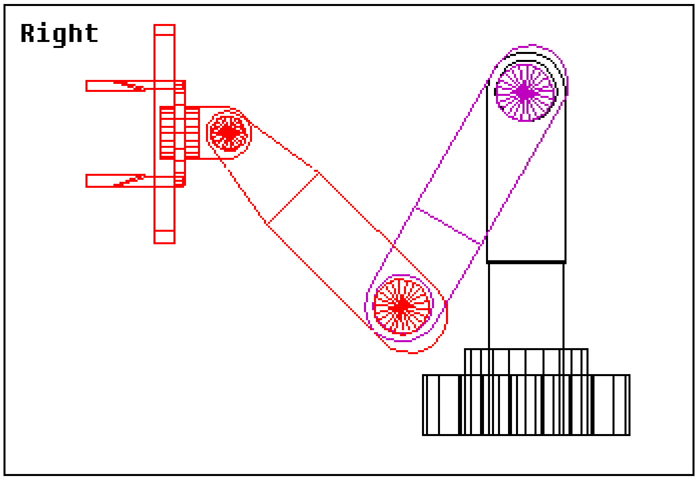
Operate the  button again to insert a further 10 frames at the front, to move to frame position 21.
Rotations:
arm1 Y - axis - 45 degrees
arm2 X - axis - 45 degrees
arm3 X - axis + 90 degrees
joint X - axis - 45 degrees

Key 4 / Frame 31



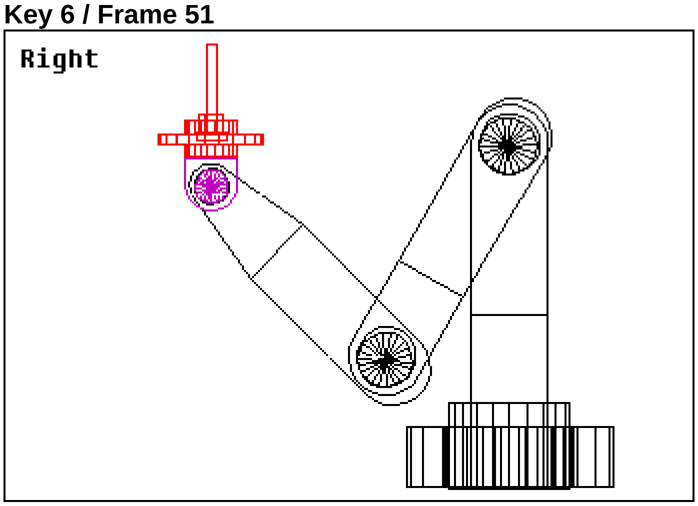
Move to the frame position 31.
Rotations:
arm1 Y - axis - 45 degrees
arm2 X - axis - 45 degrees
arm3 X - axis + 90 degrees
joint X - axis - 45 degrees
The result of these work-steps is best viewed in the Right view.

Key 5 / Frame 41



Finally, in the last two keys of the animation we want to show the operation of the "clutch". Once again, go 10 frames further with the animation to frame position 41
Rotations:
Clutch Y-axis + 90 degrees
Note that you select the "clutch" and not the object "joint." The clutch turns through 90 degrees about the "joint" and brings the slide with it into the position pictured above. However, in addition the "jaws" themselves move in the clutch. Go into the "Move Object 197" mode, and now position the "jaws" about 30 units closer to each other in Y-direction. This makes clear a further advantage of object hierarchy 140, because it is not only the rotations and movements of the parent that are transferred to child objects. Child objects, which move along a path themselves - as here the jaws move along the slide - retain their own movement path, so if the parent moves and rotates the movement path also moves. You can see during the animation that despite the rotation of parent "clutch" the "jaws" always move up and down within the slide.

Key 6 / Frame 51



The last Key is on frame position 51. It is again concerned entirely with the movement of the "clutch"
Rotations:
Joint X - axis + 90 degrees
Clutch Y - axis + 90 degrees
In the last Key, the "joint" lifts the "clutch" to the top, while this itself turns further about its Y-axis and the closed "jaws" open once more. You must enter the "Move Object 197" mode and retrace the "jaws" movements. There - because the clutch has turned - this time the "jaws" are moved 30 units apart along the X-axis.

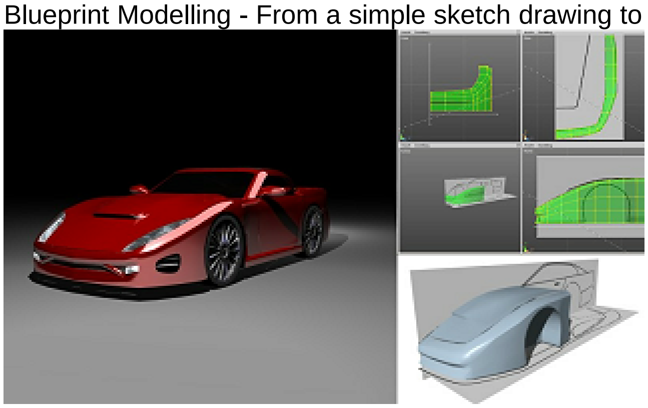
This done, the animation of the robot is ready. Using the play-buttons above the navigation button-strip you can now see a small preview animation and admire the little dance of the robot. What is still missing are the settings for camera, light and textures. The file "robot_b.cmo" includes the complete animation with all settings used to create the demo animation.

See also: Animation 400

6.5 Tutorial - Blueprint Modeling - Car Design

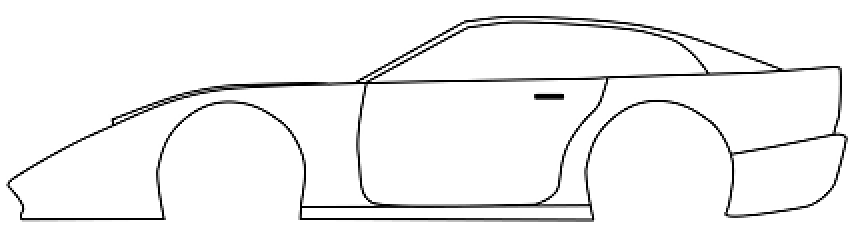
Many thanks to Pascal Heußner, who wrote this tutorial. This is no beginners tutorial. You should at least have already worked through the chapters regarding facet extrusion 228 and uv-mapping 218.

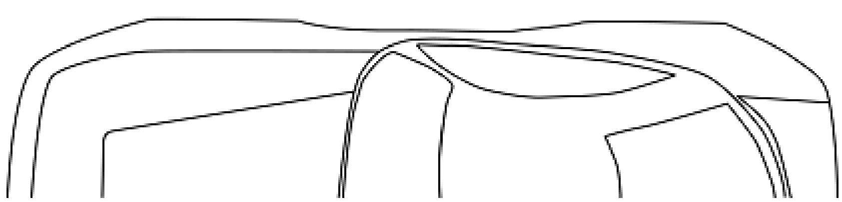
Blueprint Modelling - From a simple sketch drawing to a complete car model.



Believe it or not, the body of this noble sports car was modeled from simple boxes refined step by step using mainly MR-3D Designer's facet extrude function.

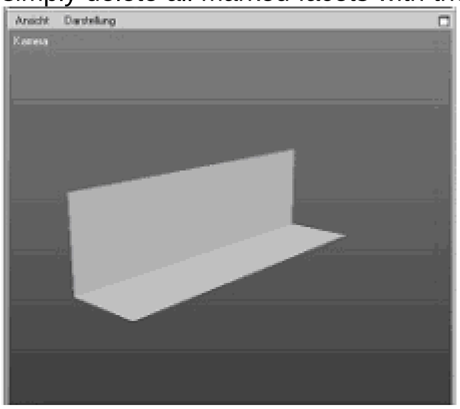
A blue print is a technical detail drawing which resembles the orthogonal views of the MR-3D Designer viewports. You can find many blueprints of original car models in the internet (you can find corresponding internet links on the 3d-designer.com link page). For clarity the blueprints used in this tutorial are rather simple drawings I painted myself. You can find this blueprint picture in the "bitmaps" folder of the MR-3D Designer installation. The final project file "car_tutorial.cmo" can be loaded from the project browser.



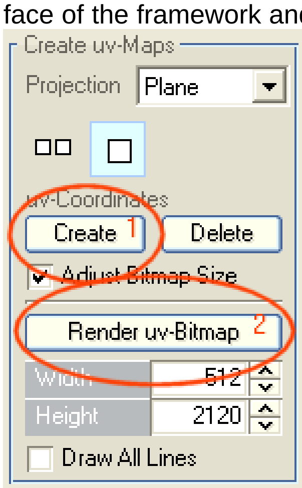


This tutorial uses only a top view and a side view of the sports car. We model only a half of the car. We will produce a mirrored copy at the end of the tutorial to complete the car.

Creating a Three-Dimensional Blueprint Framework
First we create a simple bracket consisting out of two flat faces on which we can project the blueprint pictures. We create this framework from a primitive box with the dimensions width = 127.8, height = 153, depth = 529.2. Select all facets except the facets of the left and bottom walls. Then simply delete all marked facets with the "Del"-button on your keyboard.

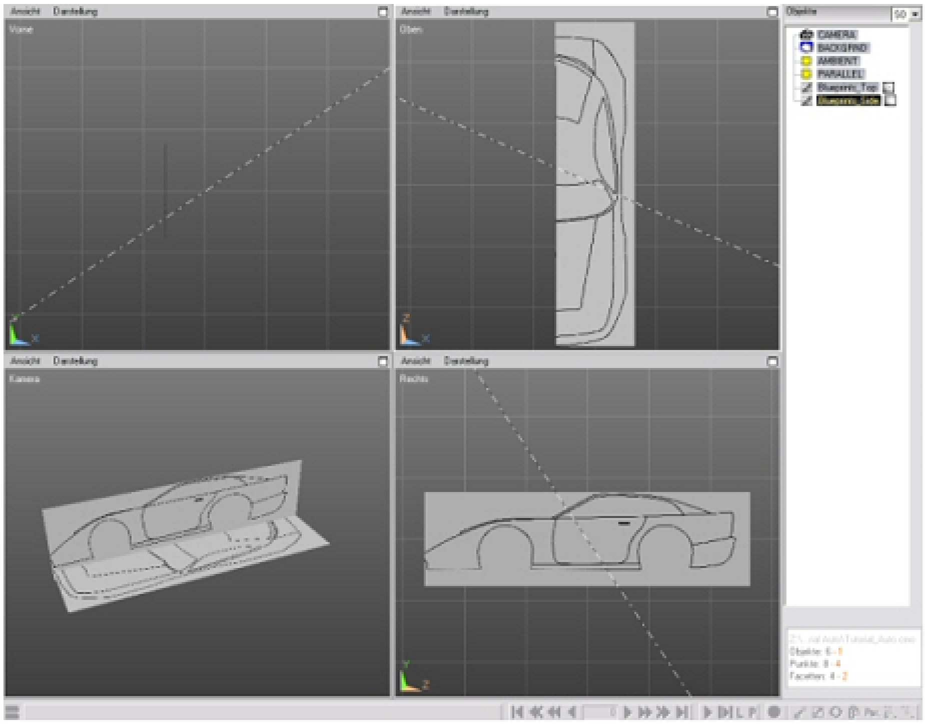


The remaining framework after deleting the box walls. It will be used as a projection plane for our blueprints. Therefore we rename the object (double click on the object name) into "Blueprint". Call up now the Material dialog and create a new bitmap material "Blueprint Top". Assign the material to the "Blueprint" object. Leave the dialog, mark the framework object, change to rotate textures, and there align the texture axes of the bitmap material so that the z-axis is facing downwards the bottom face of the framework and the x-axis is directed to the right.



Now first generate uv-coordinates 218 for the framework and immediately afterwards render the uv-bitmap from the newly created uv-coordinates. Save the uv-bitmap from the render window menu and load it into an image editor. So far only the outline of the framework's bottom plane has been drawn in the uv-bitmap. This outline serves as the template where you can paint your blueprint (or copy an existing blueprint into the borders of the frame). After the blueprint drawing is finished, save it to the MR-3D Designer bitmaps folder (or the project folder where you saved your car project). Call up the Material dialog again, select the "Blueprint Top" material and choose now the previously painted blueprint picture as the bitmap texture for this bitmap material.

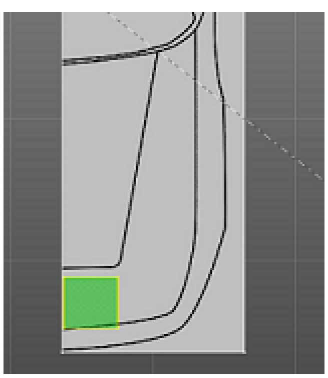
After leaving the Material dialog the bottom plane of the framework is drawn now with the blueprint's top view in the viewport windows (provided you have switched on "Display - Texture (uv-Maps)" for the viewport depiction.
Select now the facets of the side plane of the framework and use the "detach object" function to detach the side plane into a separate object. Now repeat all previously described working steps to provide the side plane with uv-coordinates and an according bitmap material for the side view of the blueprint.



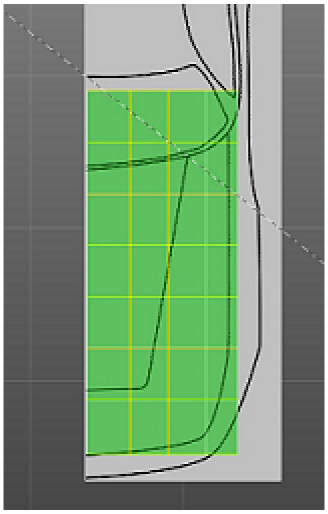
This is how the blueprint framework should be presented now in the viewports.

Modelling the Car Body

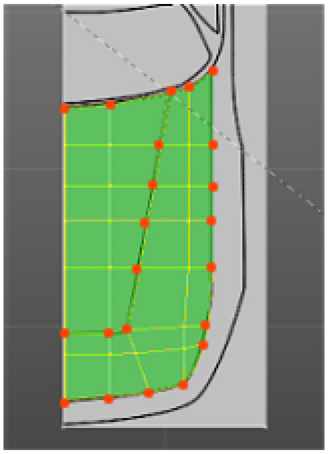
Now we can start to model the car body. First we create a new "model"-material. We choose a signal green color with high transparency. This material will be allocated to all parts of the car body so that the blueprint sketch always remains visible behind the model. Let's start now with the engine hood. Create a box primitive and assign the transparent "model" material to it.



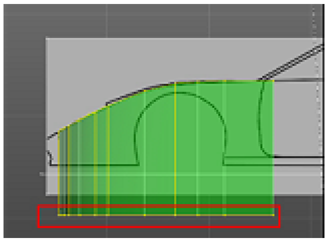
Move the box in top view to the midpoint at the front of the engine hood.



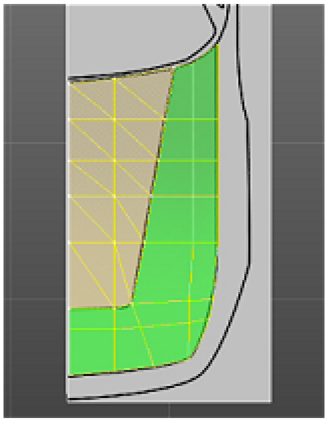
Now select first the facets at the side of the box and use several times the facet extrude 228 function to drag out three or four new segments to the left. Then select all rear facets and drag out new segments to the top until the whole engine hood is covered by the segmented box.



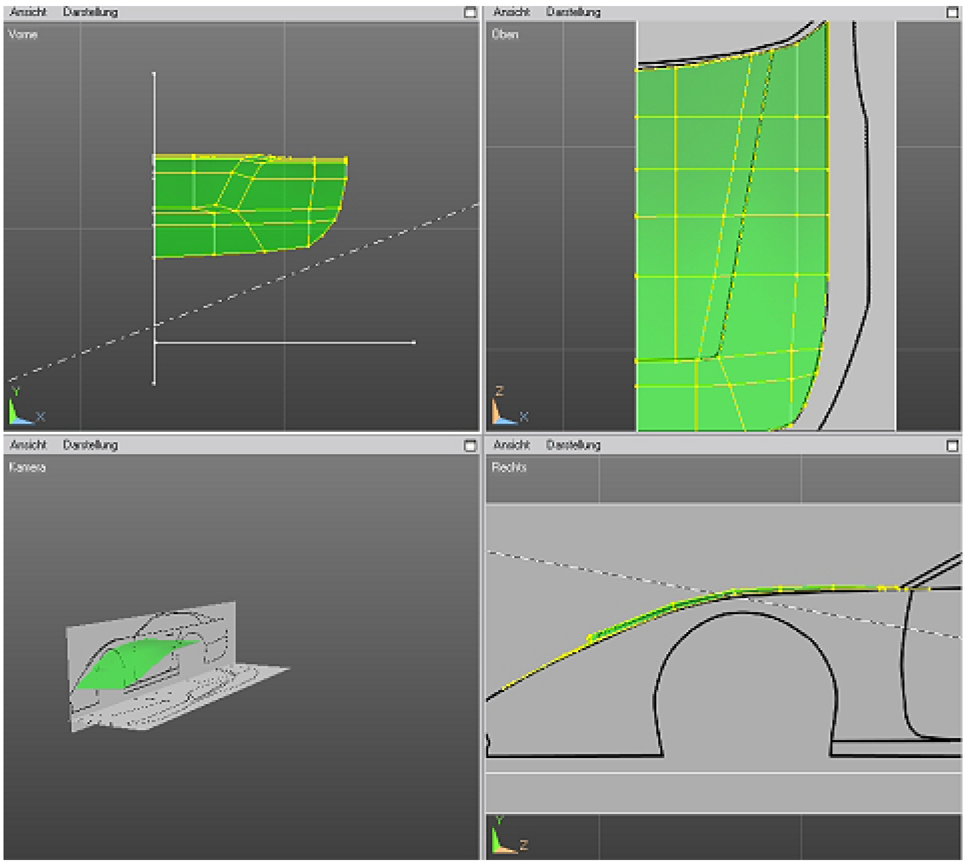
Now, one after the other, select the outer segment points and move them inwards to match the outline of the hood. Then move the points next to the inner black line onto the line, they will form later an air inlet.



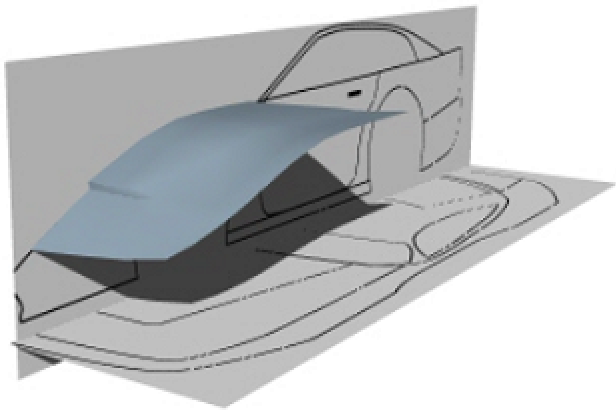
Change to the right view. Here first move all base points of the segmented box to the bottom, so that there is enough space at the top to shape the height contours of the engine hood. After that you can mark all lower points and delete them. This will also delete all side facets so that only the engine hood remains.



Now, in facet selection mode, mark all inner faces of the hood.

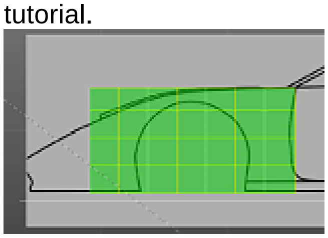


Then use again the facet extrude function to lift the inner hood a little bit up to the top. Adjust the points of the raised faces again to match the contours of the blueprint.

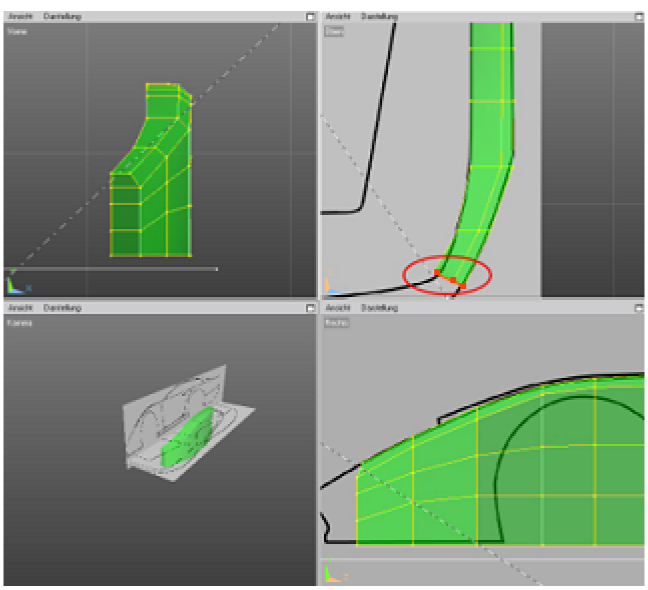


A first rendering of our blueprint framework and the engine hood.

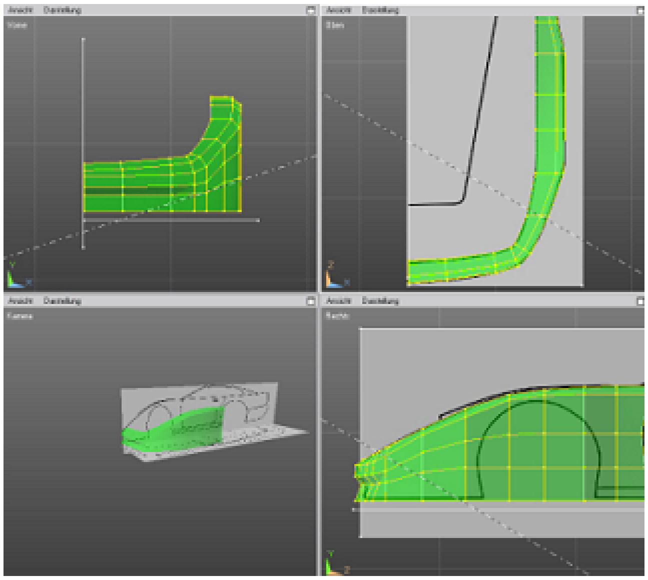
Next we create the front fender. For clarity you better switch off the engine hood for this part of the tutorial.



We start again with a little box which is extended to the side and to the top by some facet extrude operations.

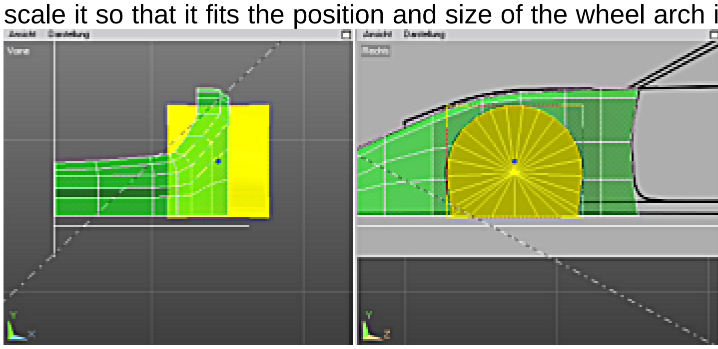


Now, just like before, adjust the segment contours to the outlines of the blueprint. Ignore the wheel arch, that will be cut out later. To model the curved transition from the side to the front part of the fender we proceed like this: Select the front points of the fender and rotate them a little bit inwards. Then drag out a new segment with another facet extrude operation and adjust the points to match the contours of the blueprint. Repeat this work steps until the curve of the fender is done and then extend the front fender up to the middle.

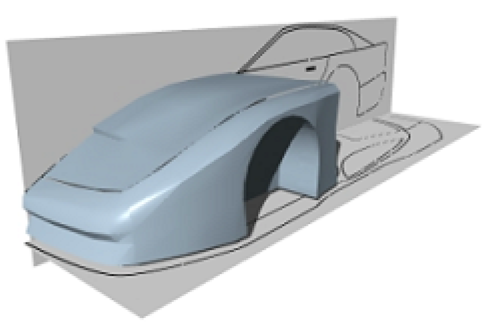


To let the car look a little bit more racy we add an indent to the front fender by selecting the lower points and moving them slightly inwards.

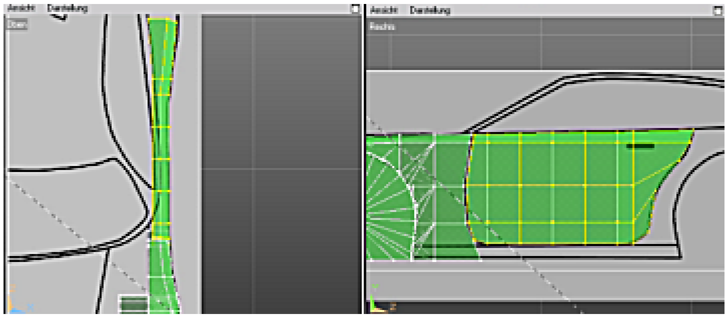
Next we will cut out the wheel arch from the fender. Create first a cylinder primitive and rotate and scale it so that it fits the position and size of the wheel arch in the blueprint.



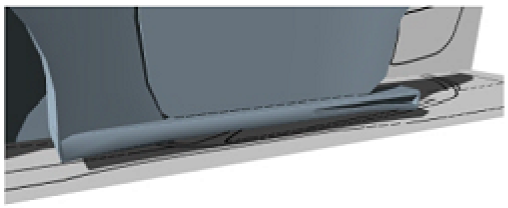
Make a second copy of the cylinder. Then subtract [233] the cylinder from the fender using the corresponding Boolean Operation [233] function. This will cut out a hole in the fender. We will use the copy of the cylinder to fill the hole. Mark the front faces of the cylinder and delete them. Then mark and delete also the bottom facets of the cylinder that overlap the lower edge of the fender. Adjust the cylinder shape by moving the inner cap of the cylinder inwards so that enough space for the wheel is left and adjust the remaining overlapping lower points of the cylinder so that they match exactly the lower edge of the fender. Finally join the fender and the wheel arch with the corresponding Join Objects [233] function.



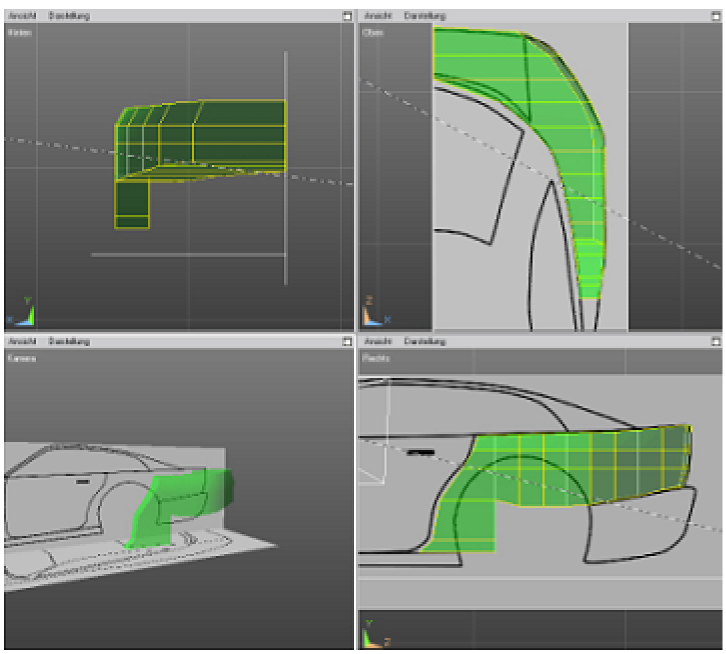
A rendering of the finalized fender with the engine hood switched on again.



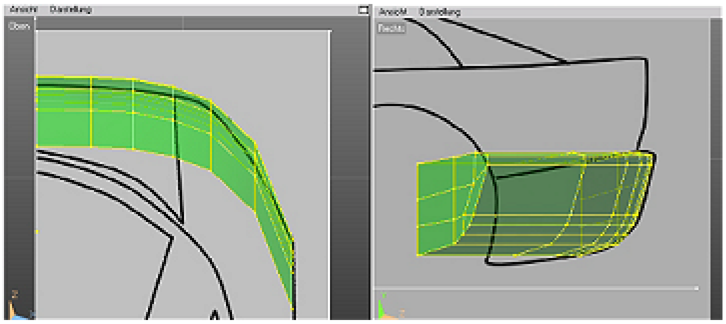
Let's continue with the door. Its the same procedure as before. Start with a box, add segments by extruding the side walls to the side and to the top and finally shape the contours by moving the outer segment points onto the outlines of the blueprint.



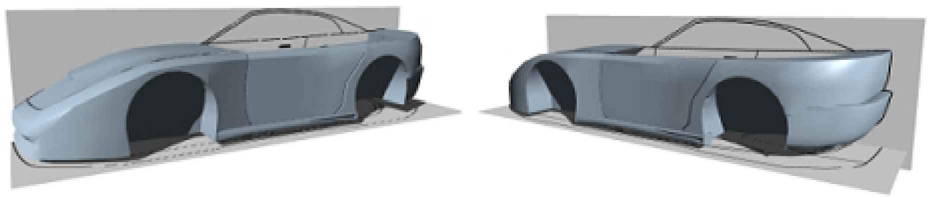
Next create a running board.



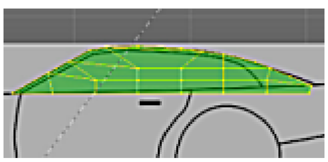
Lets follow with the rear end of the car. Start again with a box and shape the rear body with additional extruded segments like you did with the front but this time spare the lower part for the bumper...



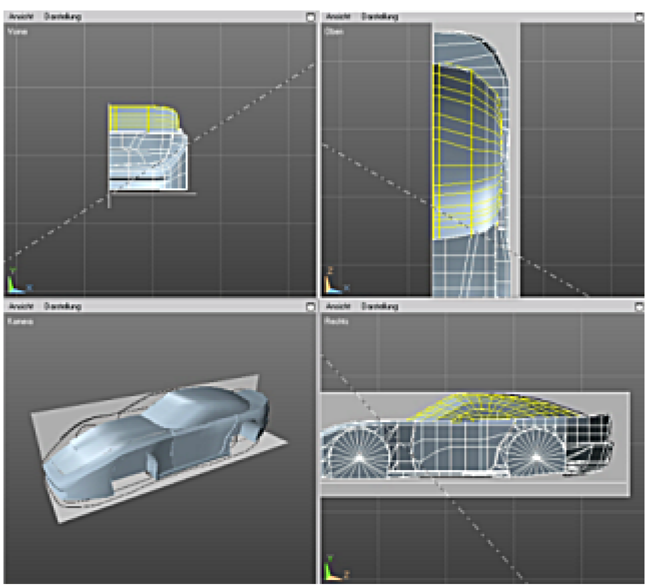
...and construct the bumper separately from another new box.



Create again cylinders to cut out and shape the rear wheel arch.



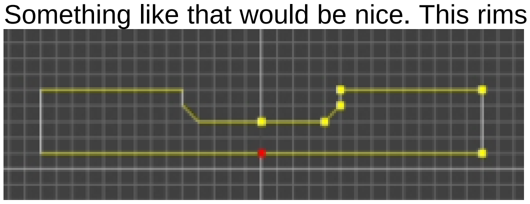
Next we design the roof. The complete top is build again from only one box extended by extruding additional segments and adjusted to the contours of the car body. The windows don't need to be cut out. Instead we will assign different materials later to the faces forming the windows.



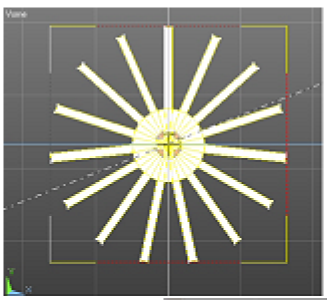
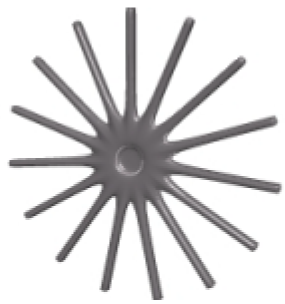
Of course, the more segments you use the more smooth the roof can be modelled but also the more difficult it will be to shape smooth curves without bumps.

The bulk of the modelling part for the car body has been completed. Let's model now something the car can roll on. A sports car needs of course wide tires with alloy rims.

Something like that would be nice. This rims were created in the sweep editor[158].

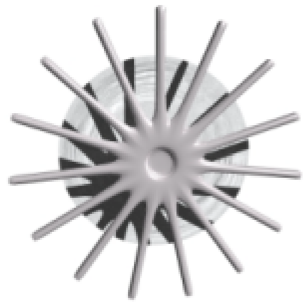


In the sweep editor draw a template like that one shown above. Enter 30 for the number of segments and create the sweep object.

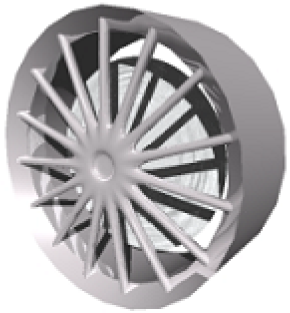
 

Back in "Edit Object" work mode mark the faces of every second segment on the outer rim of the sweep object. Now, resorting again to our nice facet extrude function, we can pull out 15 neatly spikes out of the rim at one go. Let a second extrude operation follow immediately after the first one to add another small segment on top of the spikes (This will provide better results when smoothing the spikes afterwards). Apply now the smooth object[238] function to get the spikes rounded.

Next object to create is a simple disk which is textured with a bumpmap like that shown above.

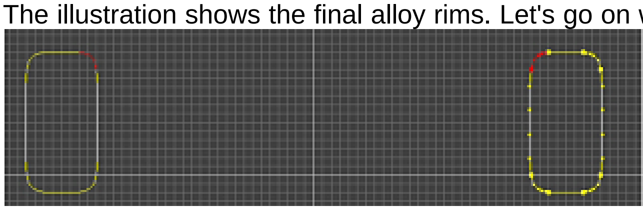


Create another material which is a reflective dark grey and assign it also to our brake disk. Next create an open cylinder, radius1 = 20 and radius2 = 23. Center it about the spikes and scale it accordingly. Then select the front points of the inner ring of the cylinder and move them a little bit backwards to get a sharp leading edge.

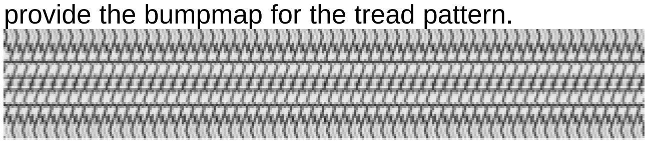


Now you can add some further details if you like. For instance, add little hexagonal nuts (closed cylinders with 6 segments) or create an additional brake shoe.

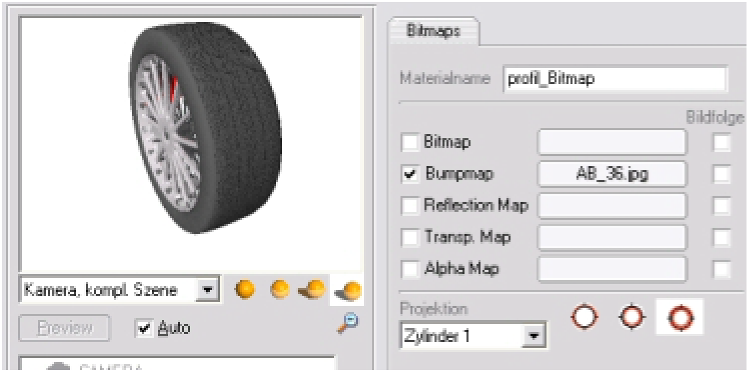
The illustration shows the final alloy rims. Let's go on with the tires.



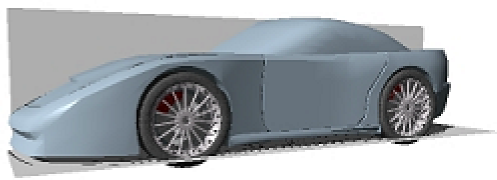
Go back into the sweep editor and draw a template like that shown above. Create the ring object described by this template and position and scale it so that it fits neatly onto the alloy rim.
Create two additional materials. One with a dark greyish color and another bitmap material that will provide the bumpmap for the tread pattern.



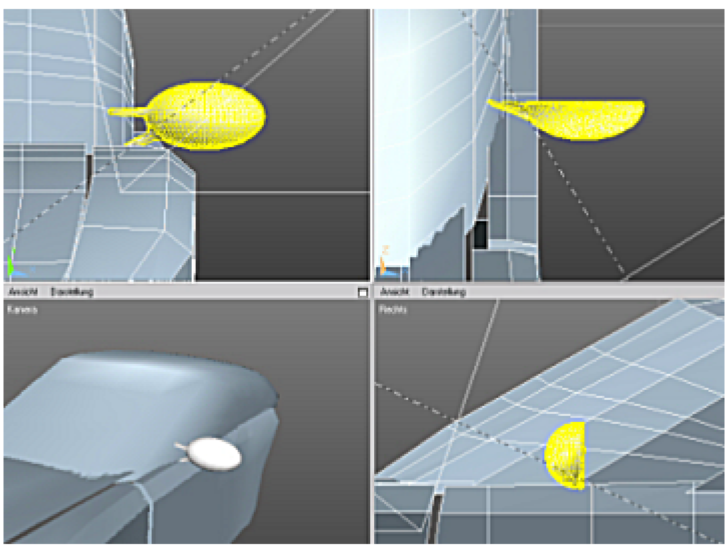
This bitmap provides the height map for the bumpmapping. It will be applied with cylindrical projection onto the tire.

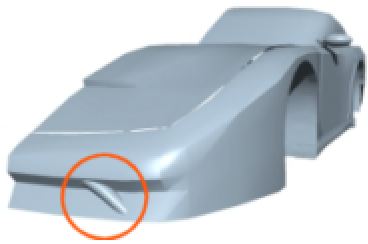


The settings for the bumpmap material in the Material dialog. The texture axes system 197 of this material has to be adjusted, so that the y-axis is aligned to the longitudinal axis of the tire-cylinder. Then scale the texture along the y-axis, so that it corresponds to the height of the tire cylinder.

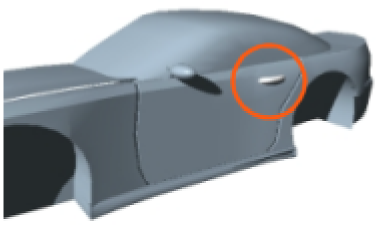


Make a copy of the complete wheel and insert both wheels into the front and rear wheel houses.


Let's add some more details to the car body. First we model a rearview-mirror. In the sweep editor we generate a circular template 160. Then reduce the rotation angle to 180° so that only a hemisphere (with front and back cover switched on) is calculated from the template. Stretch the hemisphere a little bit to an oval shaped form. Select two rectangles (two facets each) on the left side of the mirror and extrude two mounting parts from it. Finally you can use the "Smooth Object" function to smooth the mountings and the mirror.

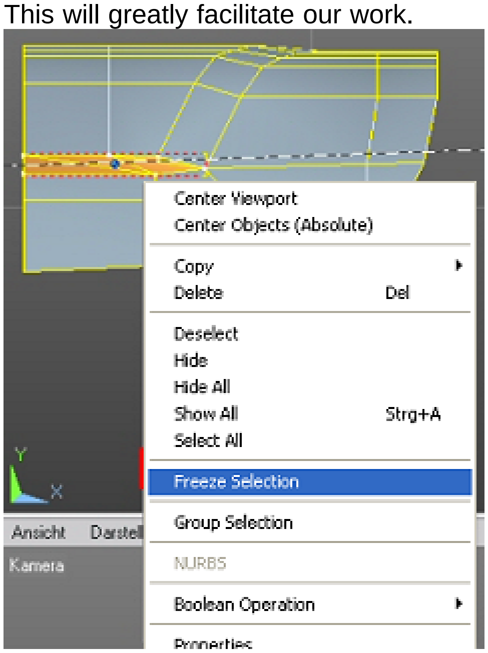


The decorative element on the front end was modeled from a NURBS-cylinder...



...as well as the door knob.

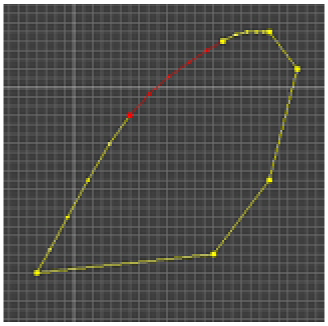
The rough modelling part of the car has been finished. But we still have to separate the windows, headlights and other details from the car body. In earlier versions of MR-3D Designer you would have to cut out these parts to separate objects to be able to apply different materials to them. But the new version supports frozen facet selections 138 which can dispose of their own material lists. This will greatly facilitate our work.



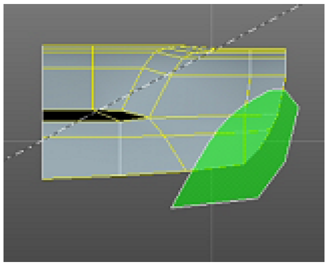
We start with the air inlet of the engine hood. Select all front facets of that part of the hood we elevated in the beginning of our tutorial for the air inlet. Then make a right mouse click in the viewport window and select "Freeze Selection" from the popup menu. Thereupon the new frozen selection will also be listed in the Object Selection window beneath its parent engine hood object. Create a new non reflective black material in the Material dialog. Select the frozen selection in the Material dialog and assign the black material to the selection. That's all.

The engine hood with the black air inlet.

For the headlights we need another two frozen selections. But choosing some of the already existing front facets of the engine hood is not satisfactory. We have to reshape the existing facets somehow to a more interesting design. For this purpose we change into the extrude editor 149.

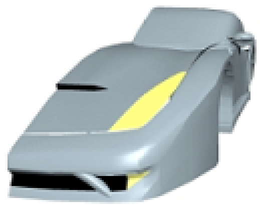


Here we draw a more pleasing template for the head lights and create a new object from it.

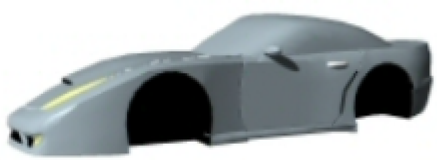


Position the headlight object in front of the engine hood and scale it backwards so that it overlaps the front of the engine hood. Now select the engine hood object and the headlight object and apply the Boolean Operation "Only Triangulate Intersections 233". This will cut new facets along the contours of the headlight into the engine hood. After that you can delete the headlight object.

Select now all facets of the engine hood forming the headlight and add them to a new frozen selection. Then create a new headlight material and assign it to the frozen headlight selection.

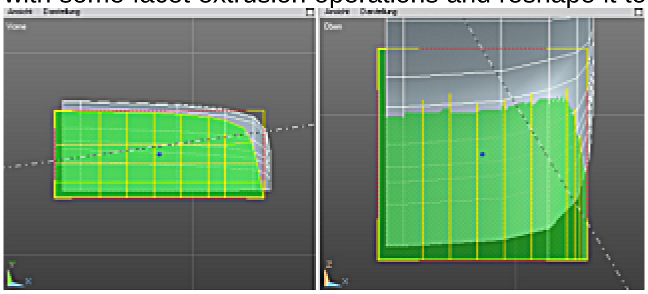


Proceed with further frozen selections for the front fender and an additional fog light.

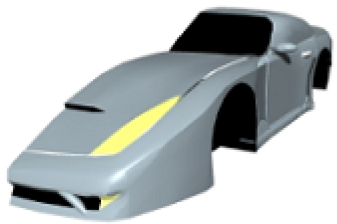


Add the wheel houses to frozen selections and paint them black.

The windows are a little bit more intricate again. We start once more from a simple box, extend it with some facet extrusion operations and reshape it to the contours of the front window.



Then proceed as you did before with the front headlights. Overlap the front cabin of the car with the segmented window object and apply again the boolean triangulation function to subdivide the front facets of the cabin.

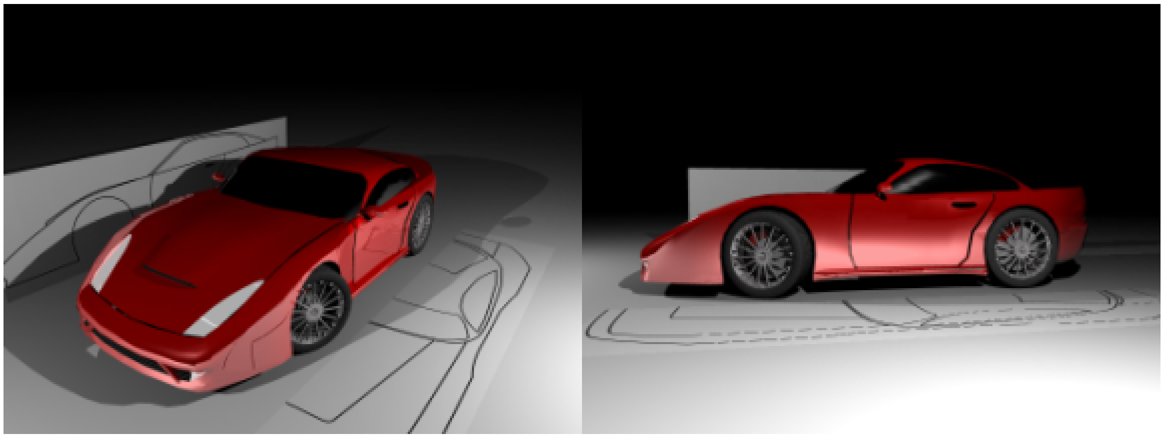


Delete the window object used for the triangulation, mark and add the triangulated front facets of the cabin to a frozen "window" selection and assign an appropriate new material to the window selection.

Add side- and rear windows as well as stop- and tail lights.

The car has reached its final state. But still it is only half of a car. For clarity we group now all parts of the car under a group object. Now, we just need to mark the group object (with it automatically all subordinated car objects will be selected), call up the popup menu with a right mouse click in the Object Selection window and choose "Copy - Vertical". This will make a complete copy of the whole object group while simultaneously mirroring the objects at the vertical x,z-plane.

After the copy operation all new objects are already selected, so moving the marked object group a little bit to the left will do the whole rest of the job, the car is ready.



Finally change the color of the model material to a finish of your liking.

6.6 Tutorial - Landscape Design

This tutorial introduces the new landscape functions. Our objective is to create a scene that resembles as closely as possible the scene shown above - hilly terrain that merges into mountain scenery with a small stream running through a pass between the mountains.

Creating the Landscape Object

Select the menu entry "Object - Landscapes..." or  in the button strip to call up the landscape editor 177 - a dialog for the production of fractal landscapes. These landscape objects are based on a rectangular grid and height information calculated with a fractal algorithm from the grid coordinates. In the editor click on the <Basic> tab to switch to the page with the basic settings 179 for the landscape definition.
The basic parameters define the fractal pattern and the dimensions of the object:

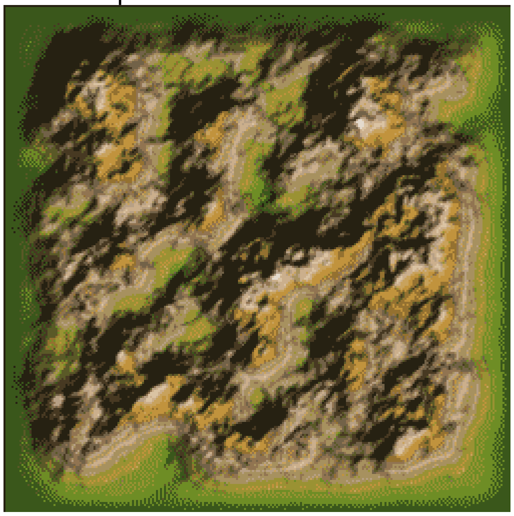
Range - The terrain we want to create covers an extensive area. With the Range parameter we can zoom out of the fractal structure used to calculate the height map. Set the Range parameter to 0.51 for our scene.

Flat Edges - With this option switched on the edges of a landscape object will smoothly run down to ground level. As we do not want to smooth down too much of the mountain details at the edges we reduce the area to be influenced by this function to a small band by entering a value of 0.34.

Random - The Random parameter enters different initial values for the fractal pattern and enables many variations in the calculation of the object. For this tutorial enter a value of 0.63.

Smooth Slope - The lower parts of the landscape still appear a little bit too much jagged to represent smooth and hilly valleys, so we increase the Smooth Slope parameter to a value of 0.69 to smooth down the jagged appearance in the valleys.

Peak - we leave the default values for the Width and Depth dimension (each 10000) as they are but we want the mountain peaks to be raised a little more. Therefore we enter a height value of 2000 for the Peak parameter.

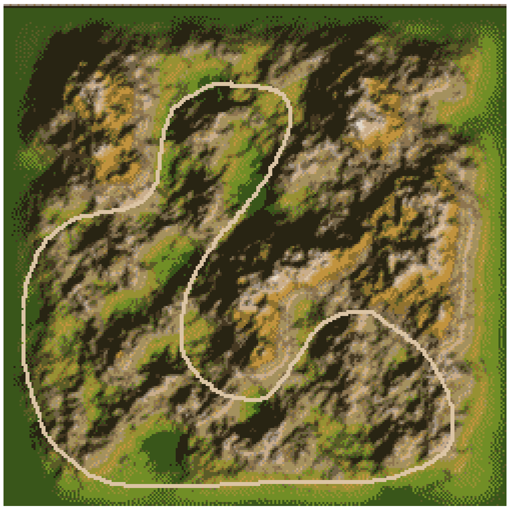


The illustration shows the temporary result as represented by the preview window.
Our aim is to generate a hilly terrain progressing into mountain scenery but there are still too many hills and mountains in the front area. We will remove the peaks later using the painting tools provided in the <Edit> work mode, but first we have to set a higher resolution for the landscape object. Since each change in one of the basic parameters calculates a wholly new fractal height map from scratch, you can't change one of the basic parameters after editing the height map with the painting tools. Any modifications made using the painting tools would be obliterated when the pattern was newly calculated.

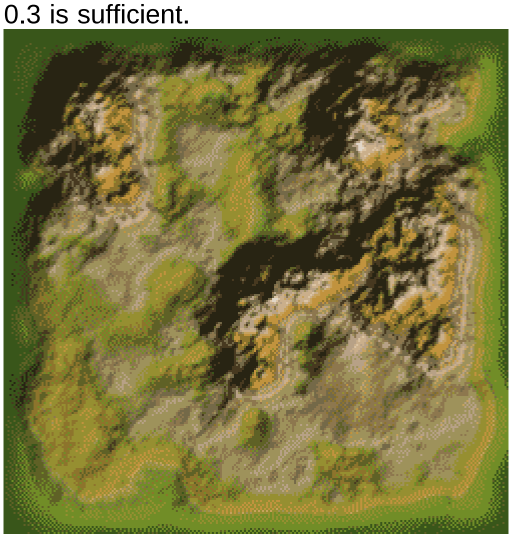
Resolution - Provided that there is enough RAM available on your computer we go the whole hog and enter a grid resolution of 700 * 700 points. Above the Resolution parameter we get the information that this results in a landscape resolution of almost a million facets, exactly 977,202. If you are low in memory (under 256 MB RAM) you should enter a somewhat lower resolution, for instance, 250 * 250 points (ca. 125,000 facets).

Now change via the <Edit> tab to the corresponding work mode in the landscape editor.

With help of the painting tools on the edit page we want to clear away some of the hills in the foreground of the height map to get a more flat and hilly terrain in this area. You can paint directly in the preview window to raise or lower the ground underneath the brush. To flatten out the bumpy hills we choose the <Lower> work mode by selecting the corresponding button on the page. For the Brush Radius enter a value of 0.3 (corresponds to 30% of the grid length) and for the Strength of the effect enter a moderate value of 0.5.



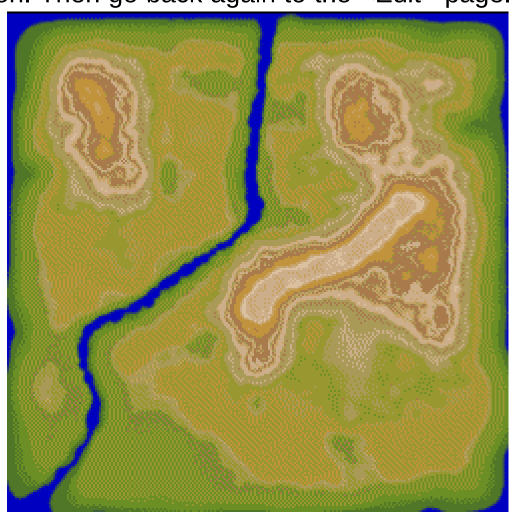
In the illustration above the area is marked where some of the mountain peaks have been taken away by the "Lower" painting effect. With the next step we want to smooth out the still jagged appearance in this area. Therefore we switch from <Lower> to <Average> work mode for the brush effect. This will level out the area underneath the brush and thus smooth the ground without removing too much of the detail. For the Brush Radius enter 0.25 and for the Strength of the effect 0.3 is sufficient.



The height map after applying the "Average" function.

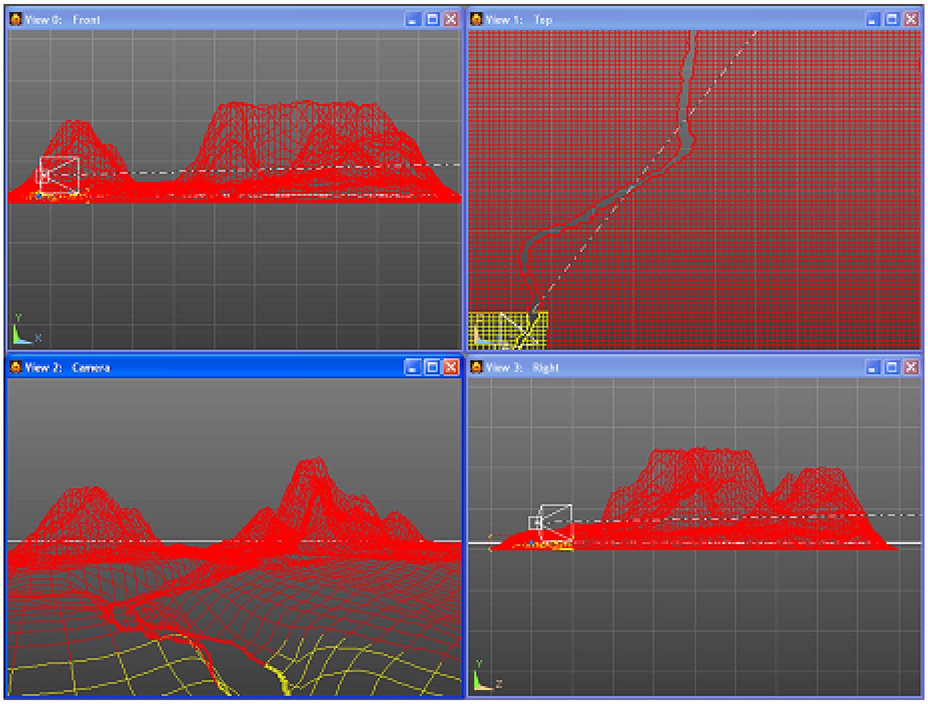
Example for the "Raise" working mode: A continuous mountain ridge was created in the illustration above with help of the "Raise" function using a very small brush radius.

What is still missing now is the little stream splitting the terrain. The course of the river can be inserted again with help of the "Lower" brush. But first we switch back to the <Basic> page and there we select the <Clipping-Height> option. Then go back again to the <Edit> page.



When we now apply the "Lower" function by painting the river course into the height map, all facets falling below the Clipping-Height will be removed and the course of the stream becomes clearly visible. Switching off the <Shaded> preview option, so that only the height map without lighting is displayed, can be helpful to determine the exact course without confusing shadows.
Finally, go over the river course once again with the "Average" brush, to smooth down the steep river banks caused by the "Lower" brush.

The main part of the work is done. Now, operate the <Create> button to generate the landscape objects. Since the <Add Plane> option was also selected, an additional plane object on ground level will automatically be created with the terrain objects.



Depiction of the landscape object in the viewport windows.

This part on object creation is completed. The landscape remains exactly where it is and won't be touched again. Instead we adjust the environment settings, like camera, background and lighting to the dimensions of the landscape object. This way you can go back to the landscape editor at any time and make some further adjustments to the height map (For that purpose you should always save your landscape settings to the visual landscape library). Then, you can create a new terrain object to replace the old one without having to adjust all settings for the rest of the scene again.

Camera - if the option "Set Camera and Atmosphere" has been activated on object generation then the camera will be positioned automatically on a good starting point in front of the landscape. Let's change now to "Move Object 197" work mode. Reduce the Zoom-parameter for the viewports to 4% so that the whole scene can be displayed in the viewport windows. Call up the object selection dialog 133 and mark the camera object with a click with the right mouse button. Back in the "Move Object" work mode again, click in the "top-view" window and move the camera - while holding the left mouse button pressed - to the front left corner of the terrain object. Then proceed to the camera 257 menu. For this vast terrain we need a panoramic wide angle effect and therefore we adjust the camera zoom to a rather low value of 35. We want to take a picture from a rather low position directly above the flowing river with the camera directed along the river towards the mountain pass and slightly inclined to the zenith. Therefore we enter a value of +3° for the camera inclination. The camera direction we adjust simply by clicking in the angle instrument for the direction and dragging the needle into a position that points to the north east. In the "top view" window you can easily follow the alignment of the camera - a dotted line originating from the camera shows the line of vision. Now comes the fine adjustment. In the camera menu select the option "Move Camera along" <Camera Axes>. Click in the camera viewport and move the camera, while holding the left mouse button pressed, into a position that represents a view similar to that depicted in the illustration above.

Light and Background

Now call up the background dialog 375 to select an appropriate atmospheric background for our terrain. To simplify matters we just select an existing background from the visual background library. Double click on the "golden sky" entry, a warm evening sun with a dense haze at the horizon. At the start the "Panorama, only planes" preview mode 31 is always selected in the background dialog. Now, of course we want to see a preview in "Camera, complete scene" mode to get a real impression of the interaction of the atmospheric background with the terrain from the current camera view. But first we have to take some precautions with a very complex scene of about one million facets, otherwise the rendering would not be fast enough to speak of a real preview rendering. Select the raytracing algorithm without shadows and antialiasing for the preview rendering (the second of the four spheres beneath the preview window). Additionally you can reduce the picture resolution by clicking on the magnifying glass button. Switching off the automatic preview update with a click on the <Auto> button is also indispensable for such complex scenes. This way you can adjust several parameters in one go without having to wait each time for the preview calculations. You can start a preview rendering any time by operating the <Preview> button then. Nevertheless, if a preview calculation lasts too long you can interrupt the preview any time by pressing the <ESC> key on your keyboard.

A first preview picture of our landscape scene in the background dialog.

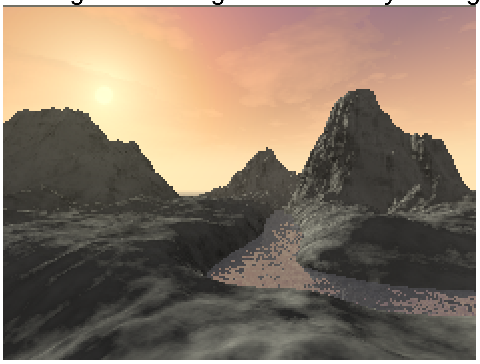
It looks quite nice already but an impression of real depth and distance is lacking. The mountains are supposed to merge much more into the distance with the atmospheric haze. As a light ray traverses an atmosphere some light is extinguished and some light may be added by emission and scattering. This yields in a change of color with distance, i.e. dark backgrounds becoming bluer and light ones redder. All these effects can be simulated using the atmospheric fog and color filter functions in the dialog. First we enter the fog parameters on the "Atmosphere"-<Fog> side of the dialog. The <Fog> button is activated so we just increase the Density to 0.20. Like in real life the fog density in MR-3D Designer is decreasing with increasing height. At Ground Height you have the maximum density and a second Height parameter defines the maximum fog height at which the density is almost zero. The ground of our scene is located at a height of zero and the mountain top at about 2000. So we enter 0 for Ground Height and 2000 for the maximum fog height. This results in a dense fog layer surrounding the foothills while the mountain peaks are clearly visible towering above the fog.

Now, change to the "Atmosphere"-<Filter> side of the dialog. For the sunset effect we enter some additive blue (0.10) and a red filter value of 0.04.

After adjusting the atmosphere parameters we have to set up a proper lighting for the scene. The sun light object loaded with the "golden sky" background from the library is a little bit too low and partly hidden by the left mountain. Leave the background dialog and call up the light dialog 341 instead.

Basically I'm satisfied with the sun settings, I just want to move the sun a little bit up and to the right, so that it peeps out right behind the mountain. We can deal with that quickly by adjusting the Inclination and Direction parameters of the parallel light object. In the light dialog at the start the preview mode "Lensflare, centered" is selected. This preview mode is best suited for displaying activated lens flare effects for a selected light object. For the adjustment of the sun position the

"Camera, background and planes" preview mode is preferable. Now, you can click directly in the angle instruments for the inclination and direction of the light incidence angles and drag the needle to a suitable position. Since no terrain objects need to be drawn in "Camera, background and planes" preview mode every change in the incidence angles is shown instantly in the preview window. For the inclination angle in my demo I've input a value of -13.3° and the direction angle of the parallel light source has been set to -110.7°. Now select the "Camera, complete scene" preview mode to get an impression of how the new light settings affect the whole scene. The picture appears a little bit to dark as the sun is still very low on the horizon. Instead of increasing the inclination angle to move the sun further up in the sky I prefer to add a second parallel light object. This light object will act as a supplementary area brightness (in addition to the ambient light) simulating the general light coming in from reflections in the atmosphere. Consequently we don't switch on the <Sun> option for the second parallel light object (yes, you can set up several suns simultaneously in the background). Instead we activate <No Shadows> for the light source in order to prevent time-expensive shadow calculations for a light source that only acts as additional area brightness. For the incidence angle of the light enter a value of -32.0° for the inclination and 57.7° for the direction. The light color ought to be a very dark gray contributing only a little additional intensity to the scene.



The preview displayed in the light dialog after adjusting the light settings.

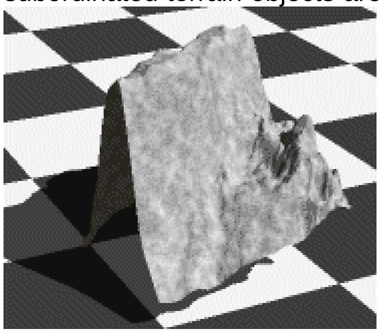
You may wonder why I did not just increase the intensity for the ambient light object instead of adding a second parallel light object for the additional area brightness. That's because of the material settings we want to add later. No matter how much points and facets are used to build a complex terrain object, the real impression of detail comes with a good surface texture. These textures ought to provide not only color patterns as realistic as possible but also an impression of bumpy and irregular surfaces. This is dealt with by normal distortion 313 The surface normal is a vector standing vertical to the surface and is used to determine the surface brightness in respect of the light-incidence. Distorting the surface normals allows a raised appearance to be added to the surface structure. And that's the point. As the ambient light object acts only as an additional intensity value representing the general area brightness - without an origin there are no incidence angles for the light - you can't use it to emphasize bumpy structures calculated from normal distortions.
The use of so-called "fill lights" is very common in computer graphics as well as in real photography. Mostly fill lights are used to simulate incident diffuse light of several directions or to emphasize particular areas in the picture.

Material

Finally the material settings for the terrain and the water plane. Leave the light dialog and switch over to the material dialog 300.

For the plane object we simply choose the "rippled" material from the visual material library on the right side of the dialog. This material represents an appropriate water texture for our river but still we have to adjust the flow direction of the streaming water. In our scene the camera is directed north east along the river passing through the mountains. Therefore the flow direction of the river should also point in that direction or exactly the other way round, up- or down-flowing. In my example the river flows towards the camera with a direction angle of -131.3°.

When creating a landscape object a preset landscape material has been automatically assigned to the terrain. The preset material provides a somewhat barren and rocky surface structure. But for our scene we need a material that provides soil and grass at lower heights and a rocky structure for the upper heights of the mountain range.
The "Landscape" entry in the visual material library is a suiting material prepared especially for this scene. As mentioned earlier in this tutorial you only need to load the material (double click on the "Landscape" library thumbnail picture) for the parent terrain object in the hierarchy - all other subordinated terrain objects are referenced to the parent's material settings.



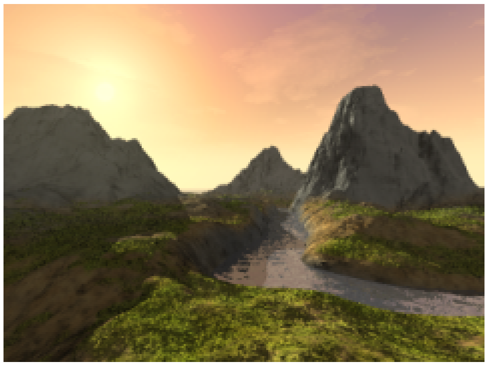
The "Landscape" material defines a rock pattern based on a procedural fractal noise texture 313. However, the material is far more complex than this. Select the <Terrain> tab in the material dialog to get to the material side for additional terrain texture layers 323. On this side you can define up to three additional fractal texture layers for the object. The way in which these layers are applied is dependent on the slope angles and the height of the surface. For instance, you can define a white snow texture that covers only areas that lie high in the mountains and have moderate slopes. A random distortion and blending parameters provide additional irregularities and smooth transitions.

For our "Landscape" material the following two layers were applied:

Layer 1 - Right at the bottom lies an earth-colored ground layer that reaches up only to the base of the mountains, mainly to cover the steep and grassless banks of the river.

Landscape preview after applying the soil-layer.

Layer 2 - Above the soil layer lies a grass layer. Besides the green colors a highly modulated normal distortion takes care of an appropriate noise on the grassy surface. Additionaly the <Patchy> function has been activated, so only patches of the grass layer will show on the ground, mingling with the underlying soil layer.

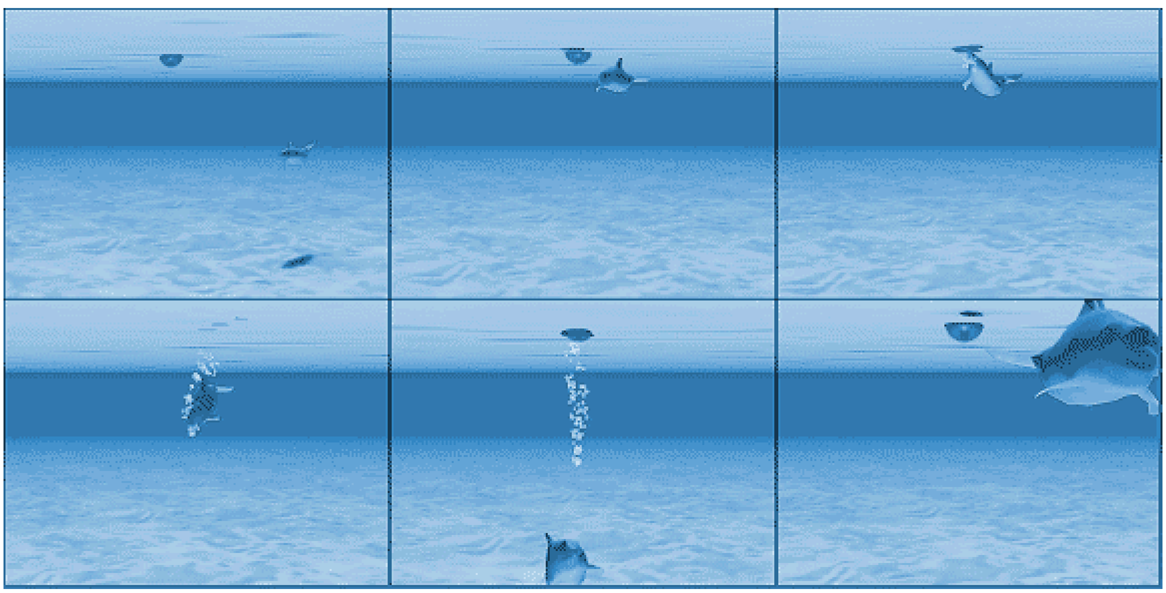

Landscape preview after applying the grass-layer.

Further details describing the individual parameters for landscape textures are provided in the corresponding chapter: Material Dialog - Landscape Textures 323.

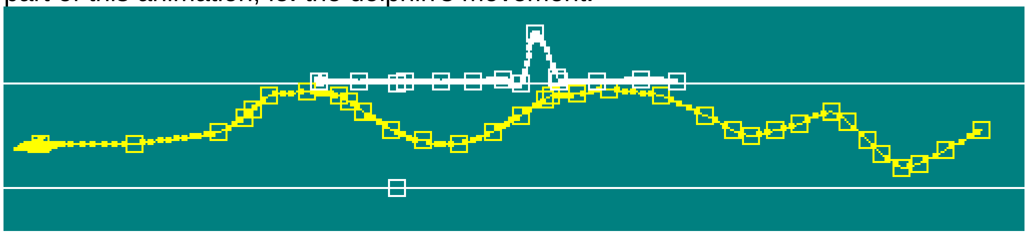
Ok, that's it. Go to the Render Options 263 dialog and select raytracing 264 with shadows and reflections for the rendering and choose a suitable picture resolution. Start the rendering 259 and enjoy your work.

See also: Landscapes and Planets 176

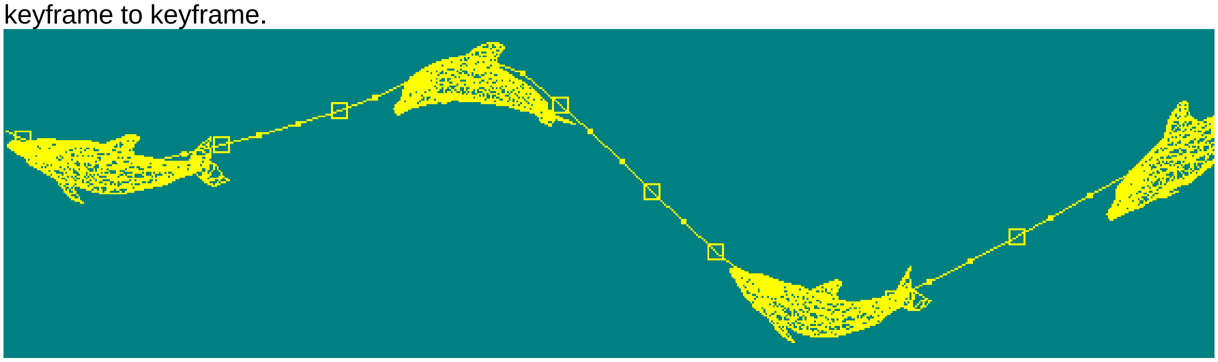
6.7 Tutorial - Animation and Deformation - Dolphin and Ball


This tutorial demonstrates the use of animated object deformation 251. You will find the associated animation file "dolphins.cmo" in the folder "..\projects\tutorials". To understand the complete demo, you should at least have a basic understanding of MR-3D Designer - especially its animation functions. Also, it won't hurt if you have also read this manual's section on Animation 400.
In this animation, the camera is fixed at a point below the water surface. A ball is drifting in the moving waves right in front of the camera. At the same time, a dolphin approaches the ball and flicks it up with his nose, followed by a steep dive. This results in a swirl of air bubbles, pulled into the water by the diving dolphin. Finally, the dolphin passes the camera with small movements of his flippers.

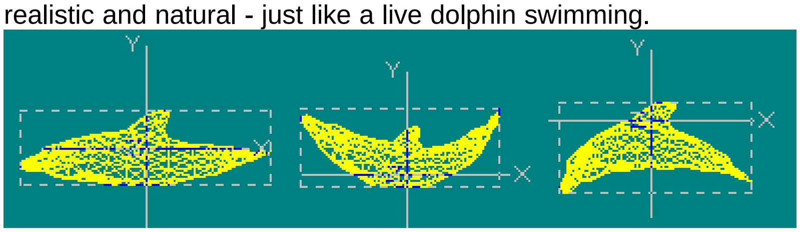
Actually, this is a very basic animation, employing two planes 186 - for the seabed and water surface, respectively. The texturing is outlined further below. We will start with the most complex part of this animation, ie. the dolphin's movement.



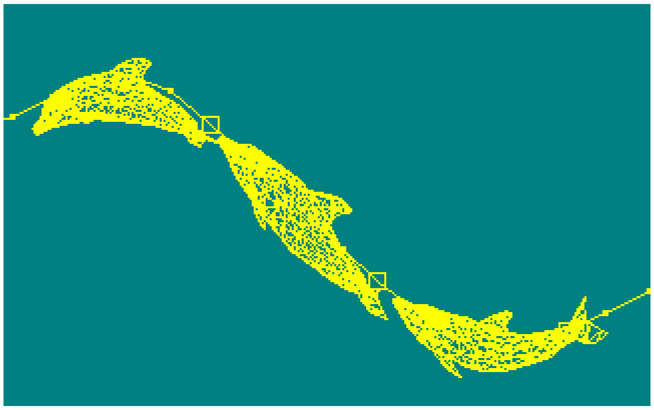
This screenshot shows the animation paths of the dolphin (yellow) and the ball (white) - as seen from the right. Both the dolphin and the ball are moving along smoothly curved paths - the dolphin controlled by its swimming movement and the ball by the bobbing movement of the waves as well as the impulse introduced by the dolphin's nose. To achieve this smooth curved movement, right at the start of the animation in keyframe 1 move the Curve-Interpolation slider in the "Move Object" menu to the right for a full interpolation of the movement path. All further keyframes we will create in the following will adopt the curve interpolation of its direct predecessor key, so we don't have to be concerned with it anymore. This is all it takes, so we can now start moving the dolphin from keyframe to keyframe.



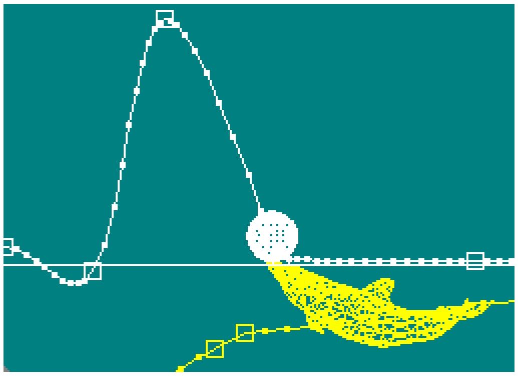
We use the animation button-stripe 401 to jump 10 frames forward in our animation and move the dolphin to a keyposition to make a nice and neat curve as it nears the water surface, where it will hit the ball later.
While the animation for the dolphin might look elegant enough at this point, the object itself still looks rather stiff and not very life-like. We will change this immediately using the "Deform Object"-working mode. Here, we will deform the dolphin smoothly at the key positions to resemble the screenshot above. When the in-between frames are calculated later, the movement should look realistic and natural - just like a live dolphin swimming.



Activate the bend function in the "Deform Object" menu. The body of the dolphin is along its x-axis - so we will use this as the axis for deformation. The body is bent up and down along the z-axis, so you will have to select "Bend Object - About Axis Z". Now, we can use the mouse to bend the dolphin up- or down for each animation keyframe.



Once this has been done, it will result in a smoothly curved animation path whose key positions should sit right at the peaks of the movement curve. When you preview 401 the animation at present, the swimming motions will not look very realistic - instead, the dolphin will just flex its body while zooming through the water like it was hurt. So we will have to change the direction at each of the key positions, which is done by adjusting the animation at the exact middle frame between every two keyframes. Erase the deformation by setting the deformation parameters to zero at each of these points, so the dolphin is balanced along its horizontal axis. Then use the "Rotate Object 211" working mode to rotate the dolphin around its z-axis to align it with the animation path so it will actually dive to the seabed. See the screenshot showing part of the animation with its keyframes. If you render a preview now, you will see that the dolphin has actually learned to swim with entirely realistic movements.



Now we return to the point in your animation where the dolphin approaches the water surface to play with the ball. Bend the dolphin in a way that makes its nose break through the water to hit the ball. Generate the key positions in a way that the ball moves in a line but also slowly wobbles up and down as it approaches the point where it meets the dolphin's snout. Generate additional keys to have the movement proceed linearly, but with a more extended vertical movement to simulate the flight of the ball. Also, the ball should be submerged somewhat after its re-entry into the water and before it starts wobbling with the wave movement again.

Now return to the key position where the ball is hit and start the animation editor 409. For the ball, enter a negative acceleration of ca. -0.50 in the "From Key"-parameter. This will make the ball go slower with the height increasing (you could also say that the movement will decrease), simulating the influence of gravity. In the next key - at the point of return - we intensify this effect by entering again a negative acceleration in the "To Key"-parameter, for the approach to this keyposition, and additionally a positive value of 0.50 for "From Key", for the behaviour after the key position when the ball begins to fall again with increasing velocity. This is all it takes to animate the objects, but you might want to make fine adjustments by repeatedly previewing 401 the animation to correct the behavior in each keyframe, changing object distance, position or deformation to achieve a smooth and realistic flow.

Particle Animation

Once the dolphin has stopped playing with the ball, it dips below the water surface, taking a swirl of air bubbles with it. This effect is created using the particle animation[425] features. Construct a small transparent sphere as the particle reference object and place it right in front of the dolphin's nose. Then put the bubble into the dolphin's hierarchical branch[140], to make it move with the dolphin's nose. Don't forget to deselect the bubble, as the reference object should be invisible during the animation.
Now proceed to the particle system editor[427]. Start a new particle action, named "Bubble1", and select the bubble object as the particle reference object. The other parameters:
Particle object - Number: 15 ±3 particle objects (air bubbles) should be generated for each frame.
Time range for this particle action: In frame 51, the dolphin's snout breaks the water surface, so this is the exact point where we want to start the particle animation, which is running for more than 60 frames. Enter frame 111 as the end of the time range.
Create particles at intervals of every n frames: This parameter defines a pause after which the particle action starts again. We only want to create bubbles at one time over a certain period of time, so we set the interval period with 500 frames beyond the end of the animation.
In each interval create particles over a period of: Once the particle action starts in frame 51, we want it to create new bubbles in every frame over a period of 13 frames.
Lifetime: 30 ±5 frames. This means that the particle action started in frame 51 will generate between 12 and 18 air bubbles for 13 frames, all of which will be destroyed after a lifetime of 25 to 35 frames.
Since the bubbles should go "up" after they are generated, we will select negative gravity under the "Particle-Action" settings. Add a small amount of rotation to make the particles behave just as a real stream of bubbles would.

Coming Together - Material, Light and Background
The swimming dolphin is only one of this animation's eye-catching features, as its impression also much depends on the moving water surface and the light reflecting off the seabed.

Waves
We will now deal with the material settings[309] for the plane object "surface", which is used to mimic the behaviour of a water surface as seen from below. In the material editor, chose a crystal clear white object color and select transparency. You won't have to filter the light coming from above by adjusting the object's material settings, as it is more convenient to use the light and background atmosphere properties. First, switch off the plane object's shadow by selecting <No Shadow>. Change to the <Waves> page in the material dialog and activate the <Waves> function with 0.11 as the recommended setting for scaling (producing a widely spread wave field) and set distortion to a moderate value, like 0.38.
Now select the "Waves" animation feature and enter a slow speed of around 0.18 for smoothly rolling waves. To match the ball movement with the water flow we adjust the "Waves" flow direction to -90 degrees.

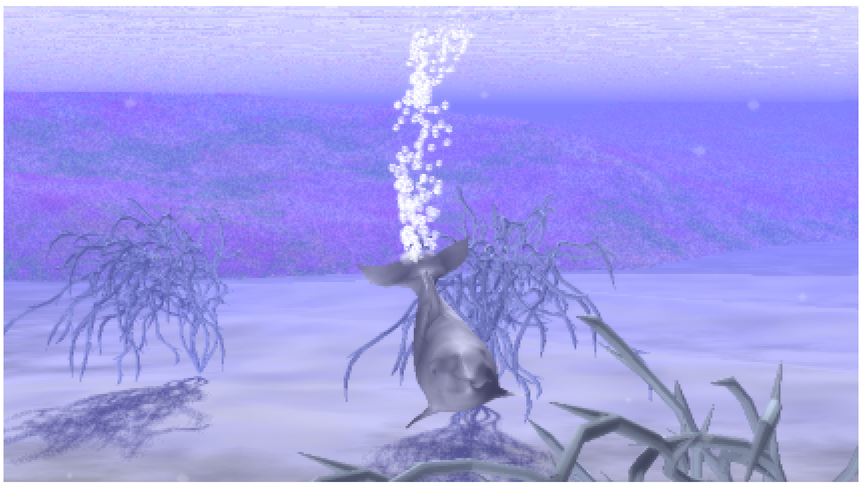
Sea Floor and Light Reflections from the Moving Water Surface
For the sea floor we apply a tiled bitmap of a sandy ground texture. Now we use a special trick to simulate light shining through the moving water surface and being reflected on the sea floor. For this purpose we simply switch on the <Waves> function for the sea ground too, this time with a wider spread for the scaling (0.03), a distortion value of 0.2 and with a maximum speed of 1.0. Although the sandy sea ground is very rough we enter a high reflection value of 0.6 but do not select the <Reflection> button, since we do not want real mirror effects on the ground. The highly specular surface in combination with the "rolling" ground results in moving highlight reflections resembling much to the light reflections coming from the moving water surface.

Background
IIn the background editor[375], we select a 3-D sky for this scene, which makes the shaded tint shine through the water surface. Additionally, we can also make good use of the atmospheric fog and color filter settings for this background model, as it allows us to make the scenery fade away into the

distance as is natural for a submarine environment. The atmosphere effect works by filtering the light depending on distance, and, without this feature, the scene would not really look like it took place in a murky, underwater environment.

Light
Two parallel light sources are used for the main illumination in the scene. The first simulates the sunlight coming from above and the second parallel light points to the opposite direction in an angle against the bottom of the water surface simulating light reflected from the sea floor underneath. Consequently a darker shade of gray is chosen as the light color and the shadow generation switched off for this source. Finally, the ambient light source is switched on, to take into account the light that is scattered around by the many particles floating in the water. To soften the under water shadows we enter for the sunlight a radius of 100 and a number of 21 shadow sensors.

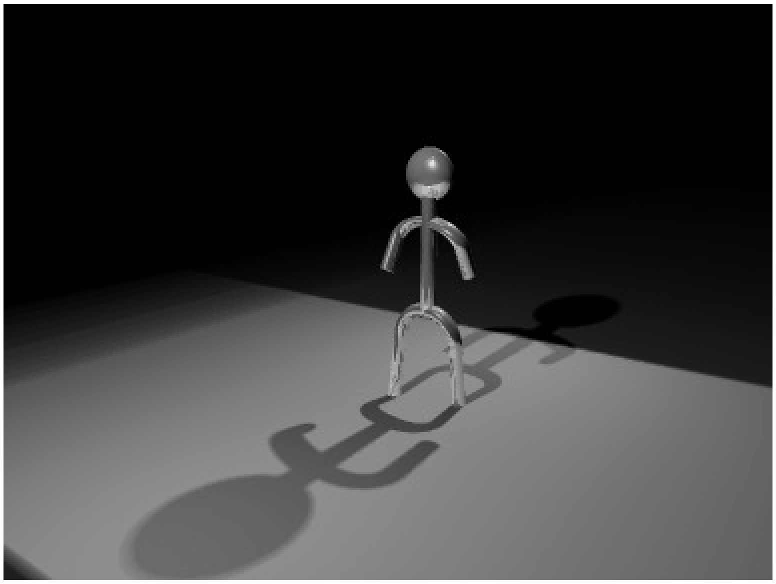
Rendering Parameters
The animation is rendered 259 using raytracing 264 with options <Reflection>, <Transparency> and <Multiple Shadow-Sensors 264 > switched on.



Here you see a snipet from the dolphin animation rendered for the 3d-designer.com gallery. Our basic scene was enriched with some further water plants and landscape objects. Furthermore, we switched on the floating particles 390 effect that is part of the atmospheric background model.

6.8 Tutorial - Character Animation - Skin and Bones

Many thanks to Pascal Heußner, who wrote this tutorial.

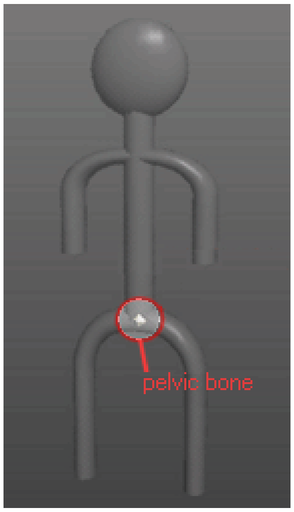


This tutorial explains the basic principles of character animation. We will animate a simple figure constructed only from some bent cylinders for the body and a sphere for the head. All parts were joined together using boolean operations resulting in the little hero you see in the picture above. This tutorial by no means explains all the possibilities you have available with MR-3D Designer. If you want to animate more complex figures and movements you will have to read everything about animation 400.

Now load the file "..\projects\tutorials\tutorial_character.cmo", which contains the little man for our tutorial.

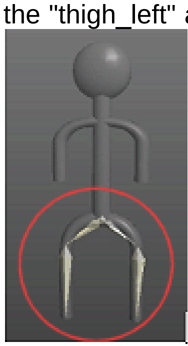
Creating the skeleton

Before we can start animating the figure we have to first create a skeleton. Change into  **Edit Skin and Bones** work-mode. First we need a root bone, from which the other bones will originate. The pelvic bone is best suited for this purpose. To create the root bone, position the crosshairs at the pelvic position of the little man. Click on **<Set Starting Point>** to define the starting position for the bone. This root bone will be without length, so we do not need to drag a length out of the starting point. Instead we operate **<Add Bone>** right away - the first bone will be added and the name dialog appears where you can give it a suitable name - in this case "pelvis". Since a complex skeleton can consist of dozens up to hundreds of bones, you should go to the trouble and assign unequivocal names for all bones. Now call up the Select Objects 133 Dialog. Link the root bone "pelvis" hierarchically 140 under the object "man". By linking bones under the "man" object this object is automatically recognized as a deformable skin for the subordinated skeleton.

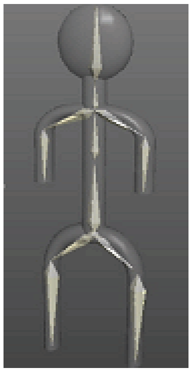


Little man with pelvic bone

Leave the Select Objects dialog again. Now we add bone after bone for the rest of the body. Select the "pelvis" again. Grab the crosshairs with the mouse between its center arrows and drag another bone for the right thigh out of the pelvic. Position the crosshairs at the knee joint and operate **<Add Bone>** . The bone "thigh_right" is created and the starting point for the next bone jumps automatically to the tip of the thigh bone. Drag another bone for the lower leg out of the thigh bone, press <Add Bone> and name it "lower_leg_right". Select again the "pelvis" root bone and drag out the "thigh_left" and "lower_leg_left" from it.

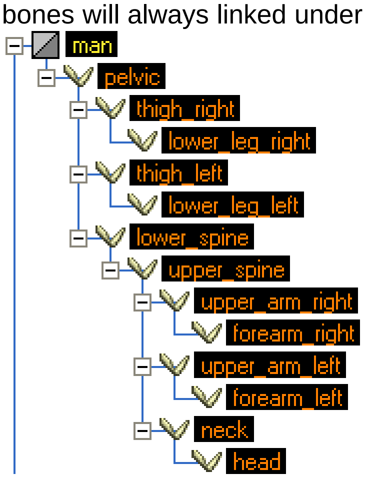

Little man with pelvis and thigh bones

Now we have to insert the bones for the upper part of the body. Select the "pelvis" again and drag out another bone up to the middle of the upper body. This bone is named "lower_spine". From this bone we pull another bone up to the neck, the "upper_spine". Since our man has a rather slender chest we can do without shoulder blades. Instead we pull out from the tip of the upper spine the bones for both arms ("upper_arm_right" and "upper_arm_left") and forearms ("forearm_right" and "forearm_left"). What is still missing now are the bones for the neck and the head. So select again the "upper_spine" and drag out the bone for the neck and finally the head bone.



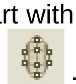
The final skeleton

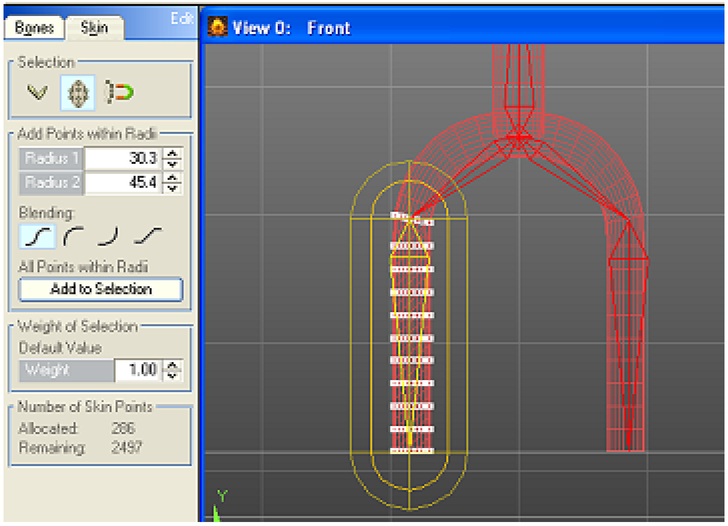
While dragging out bones from other bones the hierarchy tree is build automatically, since new bones will always linked under those bones they were pulled out from.



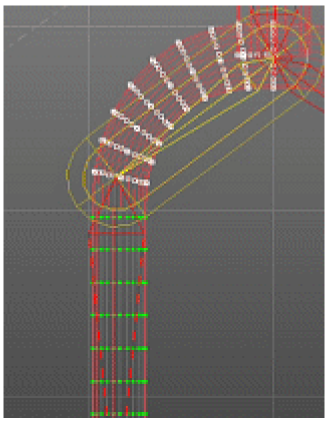
The hierarchy tree - The object "man" is the parent of the whole skeleton and serves as deformable skin. Directly subordinated to the skin is the "pelvis" root bone. All the other bones originate from the pelvis.

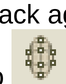
Allocating the Skin Points

Each bone is to influence only a particular part of the skin, so before we can start to animate the character we have to allocate the skin points to the individual bones. For this purpose we change now to the "Edit Skin" page by clicking on the skin tab at the top of the tool window.
We start with the right lower leg. Select the corresponding bone with the mouse and then switch over to  -"Selection - Allocate Skin Points". Now select all points 138 from the lower leg (hold <Ctrl> key pressed when adding points to the selection, hold <Shift> key down, when removing points from the selection).
You can also use the "Add Points within Radii" function to add all points within the radii of the bounding cones for a fast pre-selection and then remove or add points from this selection.



The point selection for the "lower_leg_right".



Now select the right thigh bone. You can click on the name of the right thigh bone in the Object Selection 133 window to do so. However, if you want to select the bone with a direct mouse click on the object in the viewport window instead then first switch back again to  -"Selection - Select Bones". Mark the thigh-bone with a mouse click and return to  -"Selection - Allocate Skin Points" right away. All points that are already allocated to other bones are marked in green. But you can allocate points to several bones at the same time. When this happens, the point weight - or better say the influence of a bone on this point - is changing. The point weight will be distributed in equal shares to all bones that have a reference to this point.
Now select for the bone "thigh_right" the corresponding points as shown in the illustration above.
Repeat this working steps for the left leg.
Then allocate points to the spine bones, the arm, neck and head bones:

- The "lower_spine" bone gets the lower half of the body cylinder, the "upper_spine" the upper half.
- Allocate the cylindrical area between head and arms to the "neck" bone. The "head" bone gets the whole elliptical head.
- See to it that all points of the skin are distributed to the bones, otherwise individual points will glue to their position, when the character is animated. The number of remaining points can be read in the tool window.
- You do not need to allocate points to the "pelvis". The "pelvis" only serves as the root for our skeleton hierarchy.

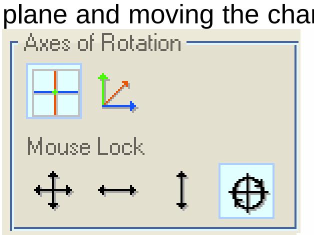
Animating the character

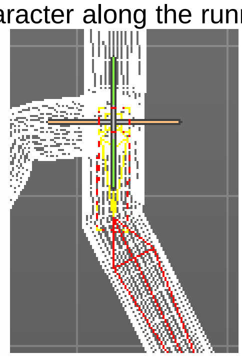
The first part of the tutorial is done. Now we will animate the character. Move the "man" to the starting point of the running track. In the Right view the man is facing now to the left. Change now into Animation Mode. At the start we want to record the position and alignment of the "man" in a first keyframe, but before that we have to select the tracks for which the keyframes are to be recorded. In each keyposition we want the whole "man" with all of his subordinated bones to be recorded in a fixed position, therefore position and rotate keys have to be created, always simultaniously, and for the whole hierarchy.

We therefore select the corresponding buttons in the animation button-stripe - position track, rotate track and key generation for all objects in the selected hierarchy.

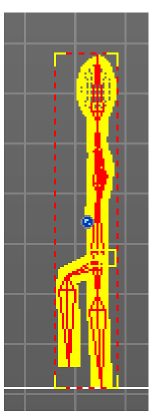
Now select the "man" in the viewport and operate the record button to create the keyframes for the character hierarchy.

Now we move forward in the animation. For each stride we need about 4 frames. Consequently we go forward to frame position 5. In this tutorial with only a simple skeleton without hands and feet a quite simple animation will be sufficient. Degrees of Freedom, Inverse or Forward Kinematics 222 are of no interest here. Actually, the complete tutorial can be carried through in the "View - Right" viewport window by simply rotating the individual bones clockwise or anti clockwise in the viewport plane and moving the character along the running track.

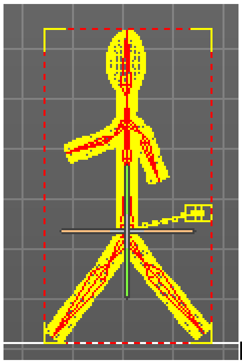




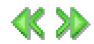
Change into the "Rotate Object" work-mode. Choose the "World-Axes" in the "Axes of Rotation"-box and for the "Mouse Lock" choose the circular-button on the right side - it will restrict rotations to the axis perpendicular on the viewport plane. If you now select a bone in the Right view, you need only to rotate it clockwise or anti clockwise to deform the skin accordingly into the desired position.


Keyframe 2 - Frameposition 5

The first step will be to bring the figure from the standing position to an intermediate stage on the way to the first stride. Lift the left thigh by rotating it about 70° to the front. Then, the "lower_leg_left" has to be rotated the other way round to bring it back in an angled position. That's it. Go forward again 4 frames in animation time.

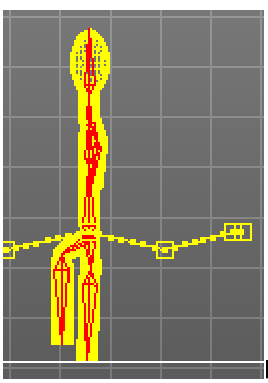

Keyframe 3- Frameposition 9

In this frame stretch the "lower_leg_left" again. Then, one after the other, select the thighs and rotate them so that they will form an open triangle. By doing this the figure will loose contact to the ground, but that's not important at this stage. If you animate a character, always concentrate on the posture, the alignment of the limbs, by rotating the bones into the correct positions. When you have done that, grab the whole character by selecting its skin and move and rotate it back to its destination position.

For our little man this means you have to move the character back to the position where the rear foot touches the ground at the same place it occupied in the previous keyframe. Use the -buttons to jump back and forth between keyframe positions to compare the position of the character in the different keyframes.

After adjusting the legs the arms have to be aligned, too. The motion dynamics will not appear realistic if the arms hang motionless at the body. Try it for yourself. If you step forward with the right leg the left arm moves forward, too. This means, when animating a walking or running sequence, you have always to counterbalance the leg movements with the opposing arm movements.

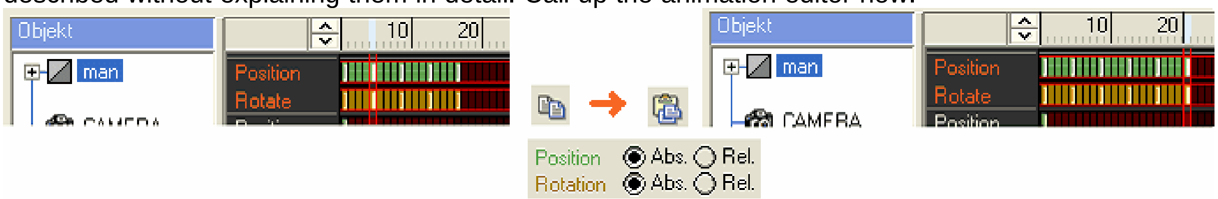

That's how it works - left leg forward, right arm back and vice versa.


Keyframe 4 - Frameposition 13

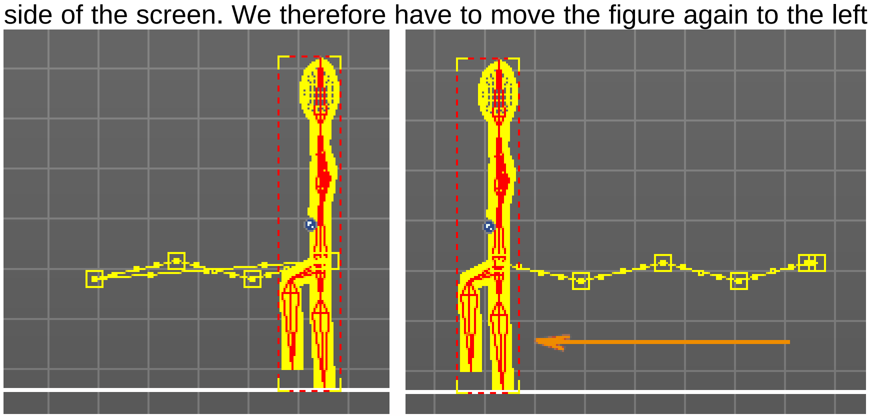
Move 4 frames forward in time again. The posture at this frameposition is almost identical to that of the second keyframe at frameposition 5, only this time the figure stands on its left leg with the right leg on its way forward. So rotate again all bones accordingly and move the figure forward until the foot position of the left leg in this key matches the foot position of the previous keyframe.

Go 4 frames forward in time again and complete the stride similar to keyframe 3, this time with the right leg in front and the left arm behind.

Now start a first preview animation. The character almost runs through a complete walking sequence, but we want our little man to make several more steps. Since we have already animated all intermediate postures lying between two strides we can simply use the copy and paste functions provided in the animation editor. The chapter "Absolute or Relative Copy of Position- and Rotate-Tracks" describes in detail the basic principles of copying absolute object positions and alignments in contrast to copying only the relative pattern of a movement. There is also an example provided of how you can extend a walking sequence. You should read that chapter thoroughly when you want to understand the following operations. In the following tutorial the individual working steps are described without explaining them in detail. Call up the animation editor now.



In the first keyframe the character is standing at the starting position. The walking sequence is contained in the keyframes two to five. For a clean loop of the walking sequence we need for the next keyframe the same character posture as in keyframe 2, when the character started to walk. For this purpose we select keyframe two on frameposition five (illustration above on the left) and copy it (Absolute Mode) over to frameposition 21. Now leave the animation editor. Since we copied the key data as absolute positions and angles the man takes over the identical body posture as in keyframe 2 - but that also means that he again stands at the starting position of the animation on the right side of the screen. We therefore have to move the figure again to the left to its new destination.

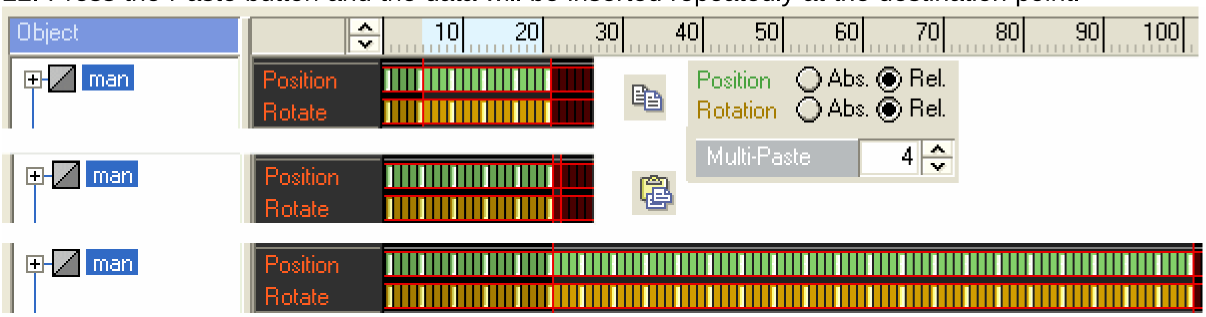


Now a whole walking sequence is ready. Starting from keyframe 2 the character moves in a complete sequence from the starting position (not the standing position but the first in-between with the left leg lifted) and takes two steps forward until he takes up the same body posture as in keyframe 2 again - just two steps further to the left. This is the sequence we can use now for a relative copy of the movement pattern. Copying this sequence in Relative Mode will make the character walk on independently so that we don't need to adjust the positions of the character further.

Call up again the animation editor.

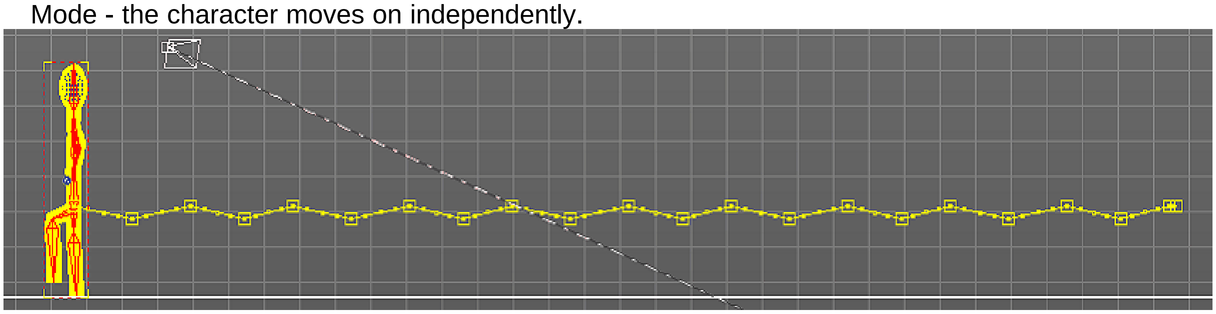
Select the frame range of the walking sequence. This is the first frame after the second keyframe, frame 6 and then with <Shift> and a second mouse click frame 21. The second keyframe on frameposition 5 has to be omitted, since he contains the same posture as the last keyframe so if you were to copy the sequence you would get two identical postures one after the other. It is also important to copy the leading empty frames 6 to 8 with the keyframe range, so that empty spaces are inserted correctly when pasting the sequence again at frameposition 22.

We want to make the character 4 more steps with both legs and instead of pasting the sequence 4 times one after the other we make use of the Multi-Paste function of MR-3D Designer. Just enter a 4 for the Multi-Paste-parameter. Before copying the data to the buffer we also have to choose the Relative Mode for the Position track as well as for the Rotate track. Now you can click on the Copy button to copy the selected frame range to the temporary buffer. Then select the destination frame 22. Press the Paste button and the data will be inserted repeatedly at the destination point.



Summary:

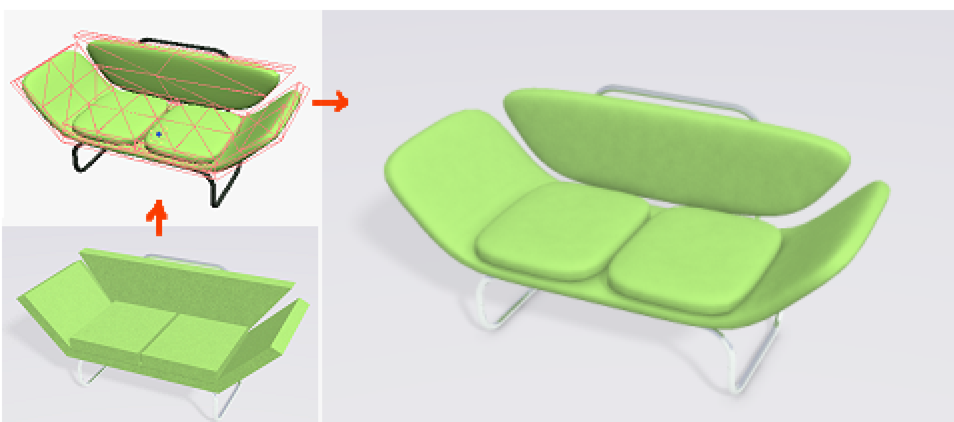
- Key 1 = Starting position, key 2 to 6 = complete walking sequence with key 2 holding the same posture than key 6.
- Copying key 3 to key 6 repeatedly behind key 6 multiplies the walking sequence.
- Since the posture in key 2 and key 6 are identical we can copy the movement pattern in Relative Mode - the character moves on independently.



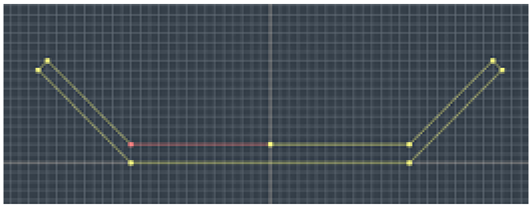
The completed tutorial - the character walking on and on...

Another example of a somewhat more complex character already animated in a simple walking sequence is provided in the projects-folder under "..projects/character/man_walk.cmo". You can also find the pure animated skeleton (without skin) of that scene in the project folder "..projects/character/skeleton_walk.cmo". The model of the character was kindly provided by the artist Stefan Danecki.

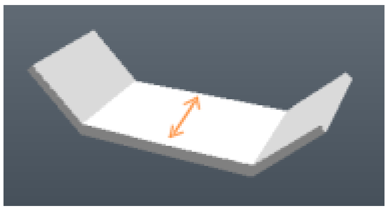
6.9 Tutorial - Modeling a Design Sofa with Subdivision Surfaces



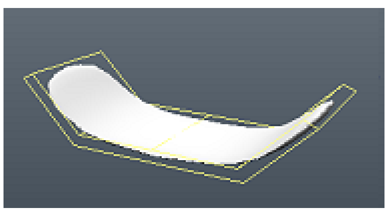
To learn more about the basics of Subdivision Surfaces you should first read the corresponding chapter - "SDS - Subdivision Surfaces 163". In this short tutorial we will create an organically-shaped designer sofa from a few simple blocks using of SDS-deformation (see project file "..\projects\tutorials\12_sds_design.cmo")



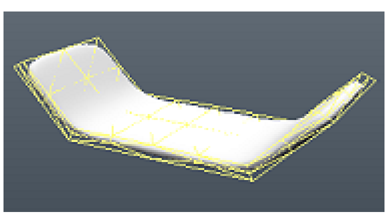
First we create the seating surface in the extrusion editor. Draw a template similar to that shown above and create the object using the corresponding <Create> button.



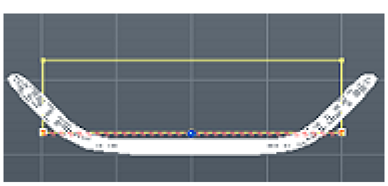
Resize the depth of the seat in Scale Object work mode . With this work step the actual modeling part for the seat is already complete. Call up now the Object Properties 289 dialog and change to the "Subdivision" page. Switch on the <Subdivision Surface> property for the seat. Then switch back to the main window by closing the Object Properties dialog.



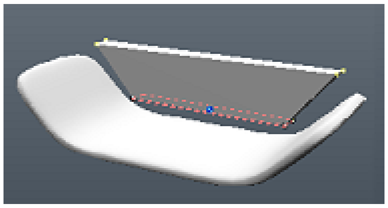
In the viewport windows the original seat object will be outlined now as a grid hull enclosing a deformed object of higher resolution and with organically-curved shapes - the Subdivision Surface. But for our purpose the Subdivision Surface is too flat and too curved. Change to Edit Object work mode and refine the object hull by operating the <Triangulate Selection> button. Since the rounding algorithms of the SDS deformation will only affect corners and edges, only those facets forming these edges are influenced by the deformation.



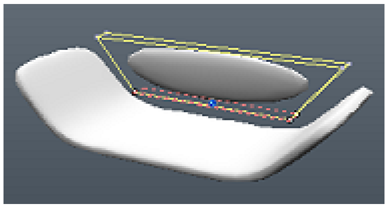
The seating surface after the triangulation operation. It is finished now.



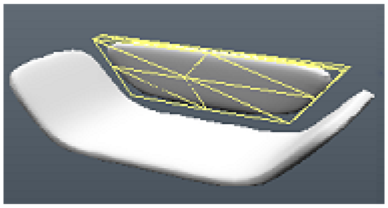
We can model the back of the sofa even more quickly. A box primitive is placed above the seat and resized to fit the dimensions of the seat.



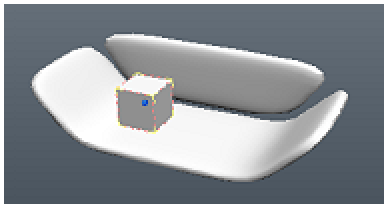
Then mark the base points of the box and pull them together with another scaling operation to form a trapezoid-shaped object.



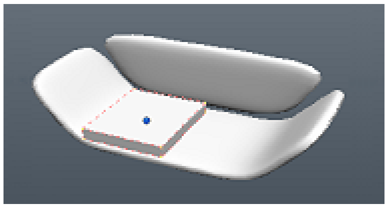
Switch the SDS-object property on again for the back object. It changes the trapezoid into an elliptical object.



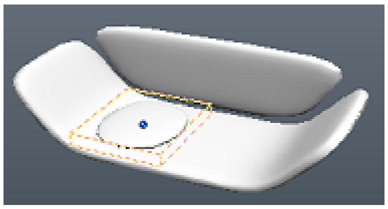
Once again, refine the back-hull using the <Triangulate Selection> function. Now we get the desired curved trapezoidal that forms the back of our sofa.



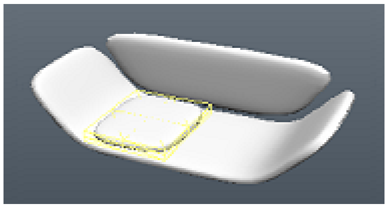
The seat cushion is also modeled from a simple box.



Scale it to a suitable dimension ...

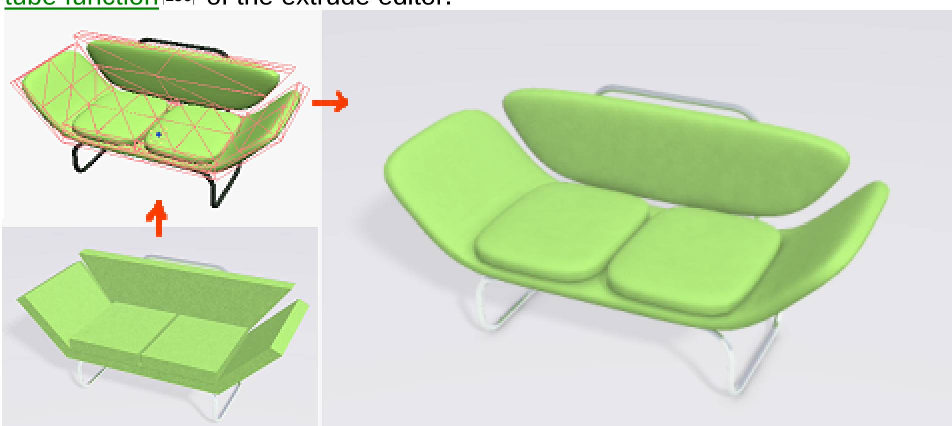


... and again activate the SDS-object property. It will change the box into a flat ellipsoid.



Apply the <Triangulate Selection> once more to refine the box hull. Now the box is deformed to a wonderfully curved seating cushion. Copy the cushion object and place the copy next to the other cushion.

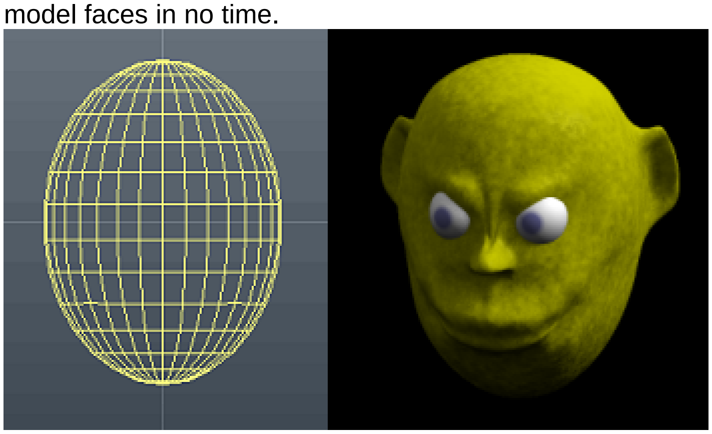
That's it. The SDS-modeling part has been finished. Now you just need to add some feet or supporting framework to the sofa. In the demo project the framework has been generated via the tube function 156 of the extrude editor.



The final scene with some materials added. In the left bottom picture you can see the simple construction of the sofa after the SDS deformation has been switched off for the primitive objects.

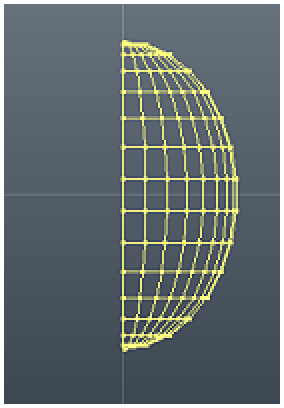
6.10 Tutorial - Modeling Faces with Subdivision Surfaces

You should first read the corresponding chapter - "SDS - Subdivision Surfaces 163" to learn more about the basics of Subdivision Surfaces. In this tutorial you learn how to apply SDS-deformation to model faces in no time.



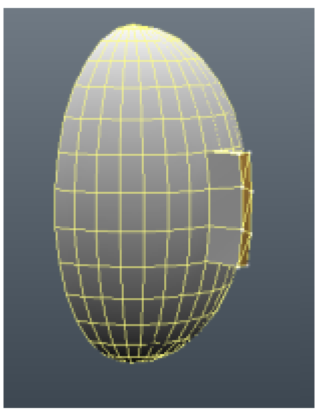
The starting point and the final aim.

We want to shape a decent monster head (see project file "..\projects\tutorials\12_SDS_face.cmo") from the ellipsoid primitive, so choose the ellipsoid primitive from the primitives menu 146 and create the new object with 32 segments. Then resize it with the scaling functions so that it resembles an egg standing on its end.

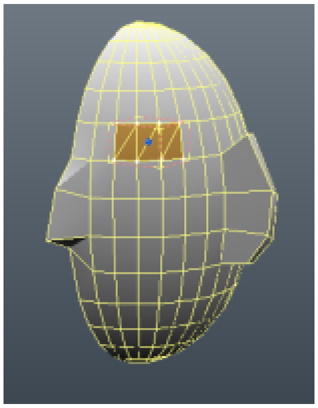


To avoid having to repeat working steps symmetrically on both sides of the face we cut the ellipsoid into two halves. Select all facets of the left side of the ellipsoid and delete the selection. Later, when the right side of the face has been completed, we will copy it mirror-reversed and join the two halves into one face.

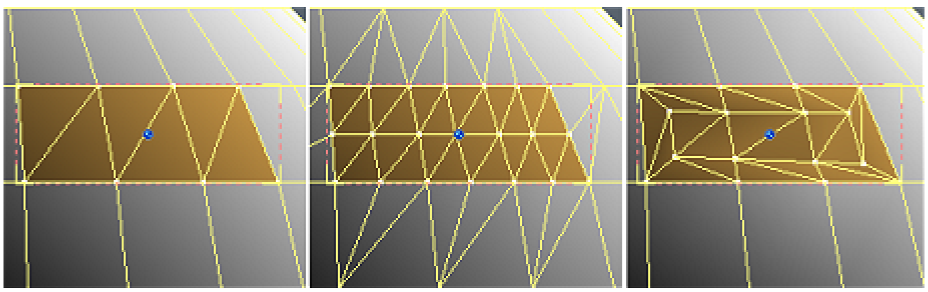
Modeling is done more easily when the face is smoothly illuminated from all directions. Call up the light dialog and switch off the default parallel sun light, then increase the light intensity for the environmental light object AMBIENT by choosing white for the light color. Switch on the <Ambient Occlusion> option for the AMBIENT object. This effect creates a soft ambient light similar to that of a cloudy day by calculating the light incidence as if it were radiated by a virtual hemisphere surrounding the scene.



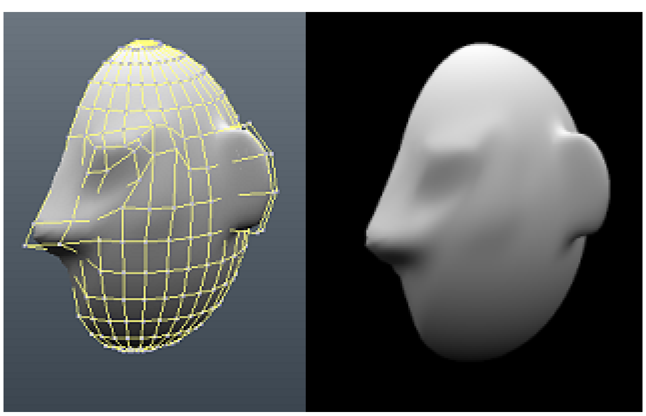
Now we can start. The procedure is as follows: By extruding facets 228 from the surface and by repositioning individual points we form a rough structure of the face. The smooth contours of the face will be accomplished in the end simply by switching on the SDS-property for the object. Let's begin with the right ear. The picture above shows a selection of 3 pairs of facets which were pulled out of the object by a facet extrusion. Afterwards, the outer points of the ear were selected to make it thinner using the scaling functions. Now select individual points of the ear and reposition them to achieve the desired rounded (or pointed) ear-like shape.



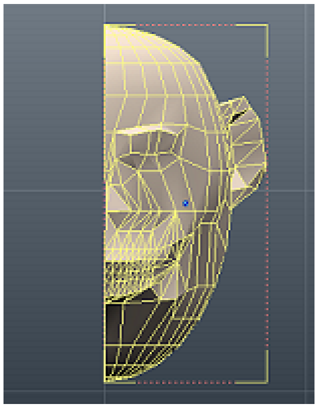
Now activate the viewport with the right view. Pull out the nose by selecting two points at the left edge and moving them further to the left. Then select the point directly under those two points and move it a little bit upwards so that the nose becomes more hooked than rounded.
Next we want to add a little bulge for the eye (the eye itself will be inserted later as a separate sphere object) and directly above it a thicker part for the brows. To form this part of the face the resolution is insufficient, so we select a little strip of facets to refine that region.



There are 2 options to do this in the "Edit Objects" menu. If you choose the <Triangulate Selection> function, each facet is divided into four smaller triangles. To maintain the homogenous net structure the points newly inserted at the outline of the marked selection have also to be connected with their adjacent facets. Therefore, those facets enclosing the marked selection also have to be cut into two halves by this operation (picture above in the middle). If you want to avoid the splitting effect of the surrounding facets you can go the indirect way using the facet extrude function. If you enter an extrusion depth of zero and a bevel value greater than zero, then no walls are pulled out of the surface (don't use the mouse but the <extrude> button). Instead new points and faces are inserted within the outline of the selection while still maintaining the homogenous net structure as shown in the picture above on the right.

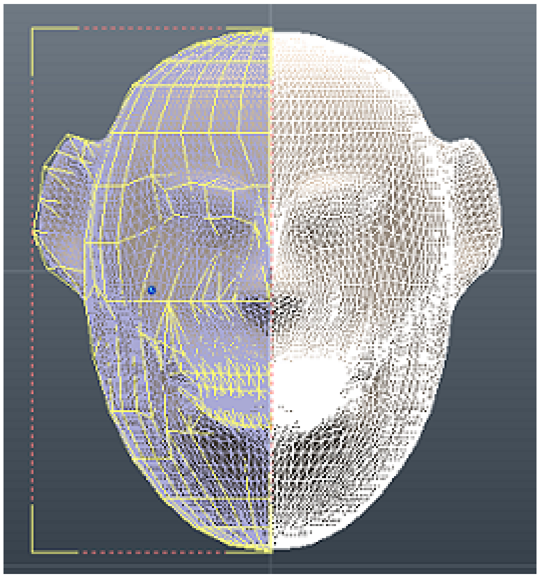


Now again select and move points to form a coarse outline of the eye socket and the eye brow. Using the magnetic deformation 231 can be of great help here. Emphasise the cheekbones by pulling out the corresponding points at the side. Now its time to get a first impression of how the face will look like with activated SDS-deformation. Call up the object properties dialog and switch on the <Subdivision Surface> property for the face object. Back in the main window you see the result - enclosed within the simple grid hull a wonderful rounded face-like surface appears. We can proceed now with the project with SDS-deformation activated. Each time we change the structure of the grid hull the underlying SDS-object will be updated in realtime. Keep in mind that the SDS-property can be switched off or on again at any time. For instance, on complex models it can speed up and simplify the work if only the object hull is displayed with the SDS-property switched off until you reach the final stage of your work.

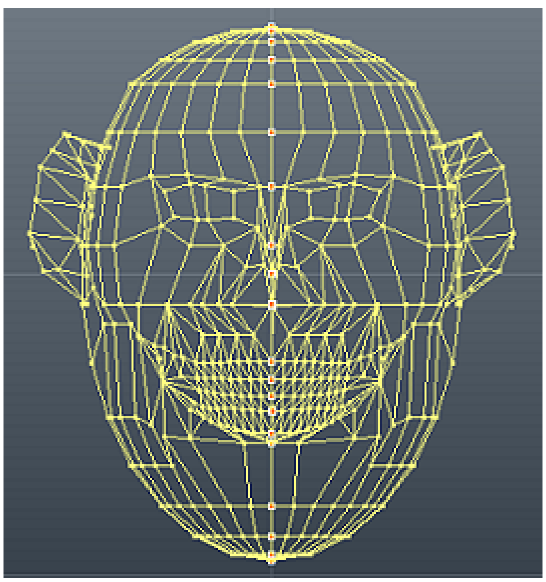


Next we need to model the area around the mouth and the chin. That requires again a triangulation of the area beneath the nose so that the resolution is high enough to form lips and the corners of the mouth.
Afterwards select all points above the eyebrows and scale them a little bit downwards to reduce the egg shaped appearance of the skullcap.
Use the magnetic function to dent in or bulge out areas at the neck, chin and the cheeks.

The finished right half of the face. It looks quite good already, but we need the whole face to get a real impression of how it will look like in the end. Create a mirror-inverted copy of the object via the corresponding function of the popup-selection 133 "Copy Selection - Mirror at Vertical World Axis".



Move the copied half to the left until the boundary points of both objects coincide (zoom in to the line of boundary points to get a very close match). Now select both objects and join them into one object using the popup function "Boolean Operation - Join Objects 233".



Now both halves of the face are combined into one single object, but still the object consists of two separated grids with doubled points along the centerline. To achieve reasonable results with SDS-deformation we need a uniform homogeneous net structure, otherwise the grid will be torn apart at the centerline of doubled points when applying the SDS-deformation. Therefore, select now all points lying on the centerline and choose the "Delete" <duplicate Points> function in the Edit Objects 227 menu. Look at the info box beneath the object selection window to verify if the number of points of the selected object has been reduced by this operation. If not, increase the tolerance value for the delete function and repeat this work step.

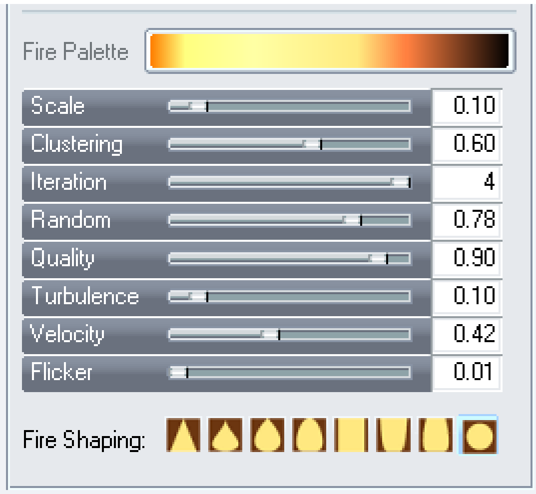

The completed head with two additional sphere objects inserted into the eye sockets...


...and finished with a green material and normal distortion for the skin.

6.11 Tutorial - Explosive Fire

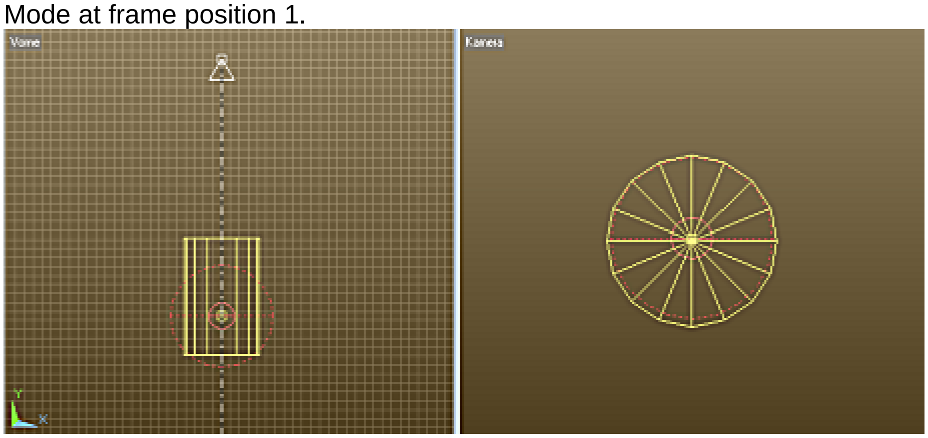
This tutorial illustrates the flexibility of the cylindrical fire object. With a simple animation of the cylinder size and the color palette we will generate an explosion sequence which also includes the transition from explosion to expanding cloud of smoke. The finished animation will be exported then as a picture sequence. You can now easily integrate the explosion sequence into other projects just by projecting it as a masked material onto auto-aligning billboard 190 objects.
This tutorial will show the way to create the explosion sequence. You should be already familiar with the basic concepts regarding the volumetric fire 358 and how to create animation 400 tracks and keys. The corresponding project file is located at "..\projects\tutorials\12_volfireexplode.cmo".

The whole animation is made from a single fire object, so call up the light dialog and create a volumetric fire object by operating the corresponding button.



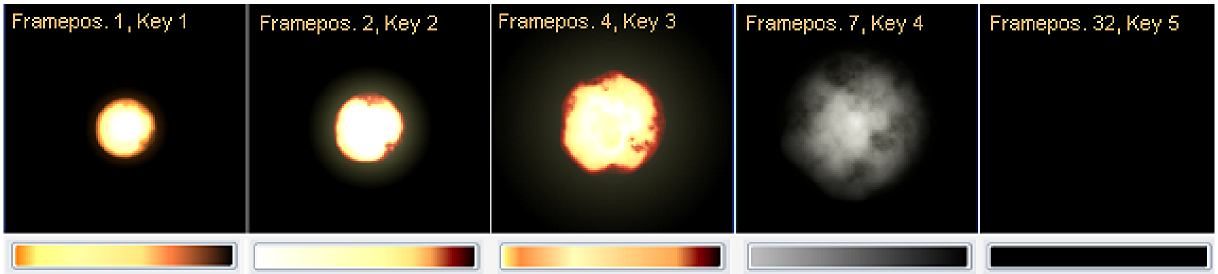
The starting parameters for the fire object.

The basic shape for the "exploding" fire within the cylinder should be spherical, activate the last button in the row of icons at the bottom of the parameter box accordingly. Then adjust the other parameters and the fire palette to the colors shown above. Leave the light dialog and select the camera object. Move the camera above the cylinder and let it face directly downwards onto the top of the fire cylinder. Select the fire cylinder and broaden it with a scaling operation along the x-,z-plane to provide a little bit more space for the explosion within the cylinder. Change into Animation Mode at frame position 1.



The starting point in front and camera view.

The animation will run about 32 frames. It will start with a rapidly-expanding fireball that changes into a smoke cloud expanding more slowly. Finally the smoke fades into the black background color. All these transitions will be animated with the same fire cylinder, all we have to do is to vary the size of the cylinder and the color range of the fire on subsequent key positions. By moving forward on the time line and changing the cylinders size and its fire parameters in the light dialog corresponding animation keys will automatically be generated, so the animation process is indeed very straightforward.



The picture above shows the course of the animation.

On frameposition 1, 2 and 4 the color range of the fire is slightly varied. It starts with a yellow to orange, then brightens up in frameposition two and reduces intensities again on its way to frameposition 4. At the same time, the size of the fire cylinder is increased on each keyposition and with it the size of the volumetric fireball inside. Our next keyposition is frame 7. Here we change the color range used for the explosion to a greyish color range. There are only 3 frames lying between the explosion and the transition to the smoke colors, but explosions happen to expand very fast so more frames would convey an impression of a slow-motion explosion. Again we increase the size for the cylinder to allow the expansion process to continue. Now we need only one last key at the end of the animation at frameposition 32. Here choose a simple black color for the color range. The cylinder is scaled up a little bit for one final time for the final expansion of the fading smoke cloud.

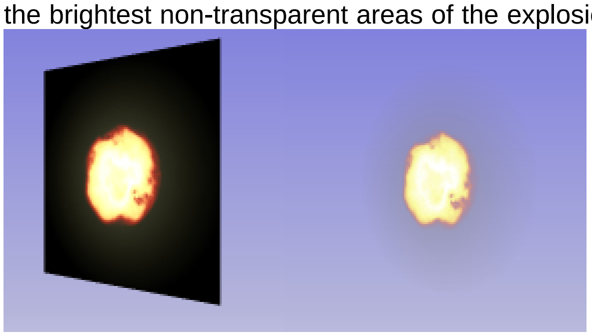
That's all, you can now start the animation and admire our nice explosion.

Integrating the explosion into other projects
After rendering the animation, we want to integrate the explosion into other projects in a more convenient way. Instead of saving the project together with its animation tracks we just save the animation itself as a picture sequence. Choose the "File - Save Animation as Picture Sequence" entry of the render window to save the animation to a particular folder of your hard disk. We want to use the picture sequence later as material bitmaps, so you have to save it to a folder that is searched by the software for bitmap materials. You can specify these folders in the Customize 27 dialog. Next we create a new billboard 190 object. It will serve as a projection plane for the picture sequence. Billboard objects are provided with some object properties especially useful for projecting bitmaps onto them. From the start they are created with a transparent basic material and a second material holding a masked bitmap (a tree by default). Now all we have to do is to replace the default bitmap with the starting bitmap of our explosion sequence.

The material parameters of the billboard's bitmap material 328.

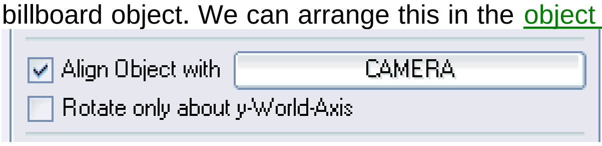
The name of the billboard's default bitmap "uv_tree.tga" was replaced with "explode01.jpg", the name of our starting picture of the exported explosion sequence. Furthermore the <Sequence> button has been switched on so that for each frame of the animation the software searches automatically for the subsequent image of the picture sequence. We also have to set a starting position for the sequence to be played. For example, If we'd enter a value of 11 for the "Seq.-

Framestartpos." then in the first 11 frames only the first picture of the explosion sequence would be used for the bitmap projection and after that beginning from frameposition 11 the complete sequence will be played. If the sequence ends then the last picture of the sequence will be used for the rest of the animation unless you have switched on the <Loop> function for an infinite loop.
If you now render a preview picture of the billboard you will find that the explosion is still projected with the bitmap's black background onto the canvas. To mask out the background also switch on the <Alpha Map> button and choose the same picture name as for the bitmap sequence. Of course the <Sequence> button has also to be activated for the alpha map. Now it works - only the explosion is visible, the black background disappears - but why? Well, the basic principle of an alpha map is - mask out those parts in the bitmap image that are black in the alpha map, leave bitmap areas untouched that are white in the alpha map and calculate a blending of the bitmap with the underlying material for all areas where the alpha map has a grey tone. (Transformation from a coloured image to a black and white alpha map is calculated automatically by the software, so you can use coloured pictures as alpha maps, too). The underlying material of the billboard is totally transparent. Since the background of the explosion is black we can use it as bitmap as well as for the alpha map - in the result the black background is masked out with half transparent transitions to the brightest non-transparent areas of the explosion bitmaps.



The explosion billboard - on the left with no alpha mapping and on the right with alpha mapping switched on.

By default, the auto-aligning object property is set for the billboard object. It ensures that the object is rotated automatically so that its z-axis always points to the midpoint of the destination object. Since billboard objects are flat projection panels the default destination object is the camera. This way billboards are always facing towards the camera. (Otherwise panning with the camera would soon reveal the flat nature of the projection panels). There is an additional property with which you can limit the rotation of the auto-aligning function to the y-world-axis. This option is useful for adding masked pictures of plants or people to architectural scenes. By rotating only on the horizontal plane it is provided that plants and people always keep ground under their feet. However, for our explosion we want the billboards to rotate about all axes so that we always look directly onto the billboard object. We can arrange this in the object properties 289 dialog.



On the <Render> page of the dialog switch off the <Rotate only about y-World-Axis> button.

We need to set another object property to control the brightness of the bitmap texture. The projected explosion should be of consistent intensity, no matter from which angle you see it and independent of the light incidence. Therefore we switch on the object property <Compositing Material 290>. Now only the pure color values of the bitmap material are used for texturing the billboard's surface while the light incidence is completly ignored.

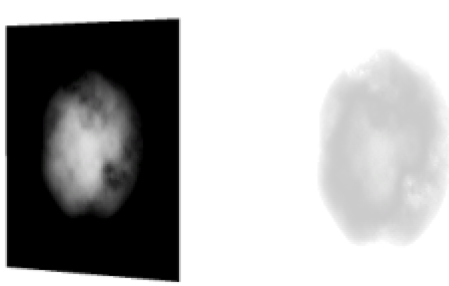
The billboard explosion sequence is now finished and can be saved. Switch off all objects, lights and background except the billboard and save the project via the menu entry "File - Save Selected Group" (only objects currently switched on will be saved). Use the merge function to import the explosion sequence into other projects. Then create an on/off animation track and corresponding keys to let the billboard appear and disappear again at particular points in time. You must also define the framestartposition for the explosion material in the material dialog so that the explosion sequence will start to play at the right moment.

Cinema like explosions and smoke trails.

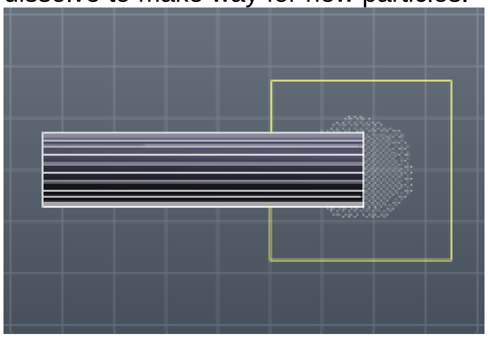
The picture above shows details from a short film sequence that was rendered with the explosion billboard we created here (you can see the film in the CyberMotion internet gallery at www.3d-designer.com). The smoke trails were also created with billboards (see tutorial - Creating dense smoke clouds with help of particle billboards). Here, only a static bitmap was used, no animation sequence. The complexity of the smoke trails is achieved by a particle system which continually emits new billboards that expand and dissolve in time. The project file of this film is also part of the MR-3D Designer installation ("..\projects\12_inferno.cmo"). In order to keep the installation file at a reasonable size, the resolution of the landscape was reduced and one of the aircrafts discarded from the demo.

6.12 Tutorial - Smoke Billboards

The aim of this tutorial is to create animated smoke rising from an exhaust pipe by means of particle billboards. The corresponding project file can be found under "..\projects\tutorials\12_smoke1.cmo".

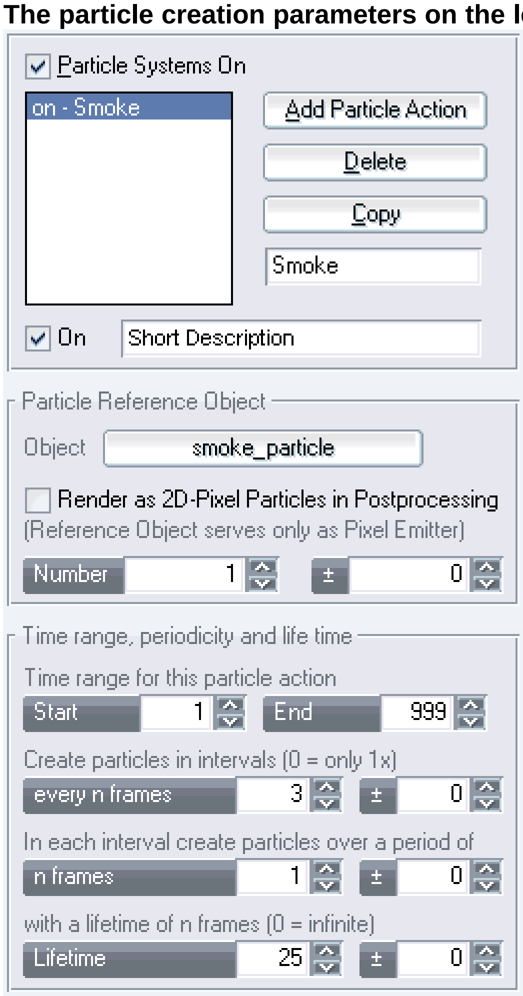


The basic object for our smoke generation - a billboard 190 with a masked smoke bitmap (the picture on the left shows the billboard without the masking material 300 so that you can see its structure - a flat projection panel). The bitmap of the smoke was created using a volumetric fire object (see tutorial "Explosive Fire" for a detailed example). Now, this smoke billboard is used as a reference object for a particle action which continually creates new smoke particles. These particles should leave the exhaust pipe and rise as they meet the cold air. Rising, they will expand and finally dissolve to make way for new particles.



First we place the billboard behind the exhaust pipe. Scale it down so that the size of the bitmap smoke fits to the exhaust nozzle. In the object selection window arrange the billboard hierarchically under the exhaust pipe. This way you can move and animate the pipe later with the billboard automatically following these movements. That's all for the set up of the scene. The rest will be done in the particle system 425 dialog.

The particle creation parameters on the left side of the dialog



Particle Reference Object and Number of newly generated particles per particle emission action
First enter a suitable name for the new particle action, e.g. "Smoke", and switch it on. Because no reference object has been chosen so far, an object selection window will automatically appear, where you can select the smoke billboard as the reference object for this particle action.
The number of new particles created per particle emission action is left to 1. The billboard is already pretty large and will expand in the course of the animation, so emitting only one particle every frame, or even only one particle every 3 frames will be sufficient.

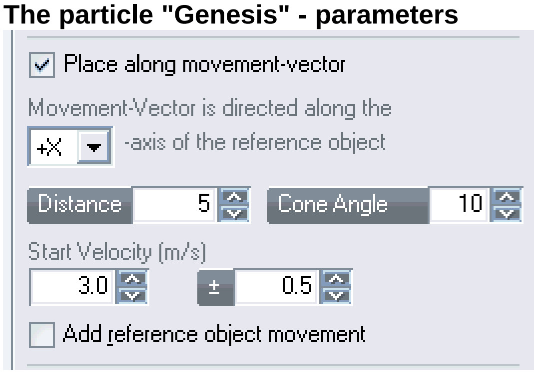
Time Range for this particle action
Start- and end frame of the particle action should cover the whole animation. Consequently we can leave the default parameters of 1 and 999 as they are.

Create particles in intervals
The interval parameter defines a pause after which the particle action starts again to create new particles. Because of the size of the reference billboard it is sufficient to start the particle action only every third frame.

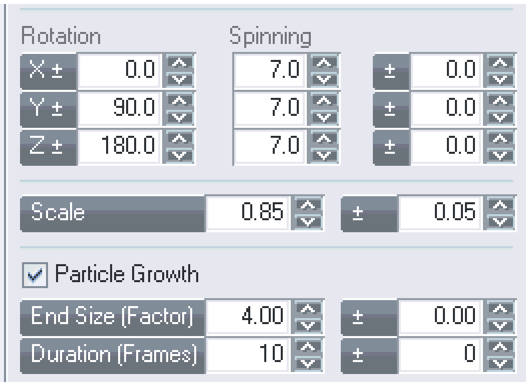
In each interval create new particles over a period of n frames
When a new interval of particle creation begins, new particles will be generated in each frame over a particular time period that is defined by this parameter. For our smoke generation we want to create new particles every third frame (interval period) over a time period of only 1 frame.

Lifetime
We want our smoke particles to disappear after a certain length of time so that they can be replaced by new ones. We set the lifetime of the smoke particles to 25.

The particle "Genesis" - parameters



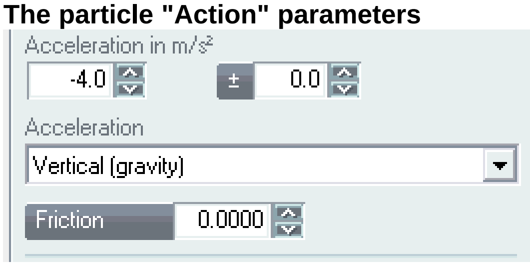
On the "Genesis" page of the dialog we can adjust the parameters for the starting position, movement direction, particle size and growth.
We want to emit the particles with a particular starting velocity along a certain emission vector - therefore we choose the option <Place along movement-vector> instead of defining a random distribution.
Since the x-object axis of the smoke billboard points directly to the right end of our exhaust pipe we can choose this axis as the movement vector for newly created particles. To add a little randomness to this direction we define also a small Cone Angle of 10° for the particle emission. The Distance parameter describes a random starting position along the movement vector, so that not all particles are generated at the center of the object axes of the reference object. The Start Velocity is set to 3 m/s with a random variance of ±0.5 m/s.
Now we have a cone shaped particle emission with particles moving with a uniform velocity away from the exhaust pipe.



Spinning - The smoke billboard was created with the auto-aligning object property 290 (with the camera as destination object) so it always rotates automatically towards the camera no matter from which angle you look at it. This object property will be adopted by the particles. To add a little bit more life to our animation we want the particles to spin around a little bit.
The auto-aligning function will always align the particle's z-axis towards the camera, so only spinning around this z-axis will make sense. Enter a random value of ±180° for the initial alignment of the particles and a spinning value of 7° for the z-axis.
Particle Growth - Next we switch on the Particle Growth. Here you can enter a factor that defines the final size of the object in relation to its initial size. Enter 4 for the End Size and 10 for the Duration. This means the particles will grow within 10 frames to four times its original size and then the growth stops.

What is still missing now is the lift that carries the smoke upwards and the fading out of the expanding smoke clouds. We can adjust these parameters on the <Action> dialog page.

The particle "Action" parameters



The ascent of the smoke can be obtained simply by adjusting a negative vertical gravity of -4.0 m/s^2. Gravity means acceleration, so the particles would become faster with every moment. Usually, this has to be compensated for by adjusting atmospheric friction, but since the smoke particles of our demo have a very short lifetime we can ignore this parameter.

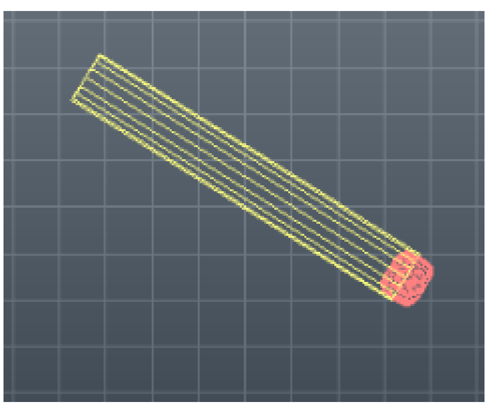
Finally switch on the <Fade Out> function. The particles will begin to fade out when they reach the specified percentage of lifetime. If, e.g., a Start value of 0.33 is defined and the particle has a lifetime of 24 frames, then the particle will start to fade from the 8th frame until it becomes invisible and is deleted in its 24th frame of lifetime.

That's it. You can leave the dialog and start the animation now.

The picture above shows details of a short film sequence (see internet gallery) which was rendered using the same smoke particle billboard and principles we applied here for our tutorial. The project file of this film is also part of the MR-3D Designer installation ("..\projects\12_inferno.cmo"). In order to keep the installation file to a reasonable size the resolution of the landscape was reduced and one of the aircraft discarded for the demo.

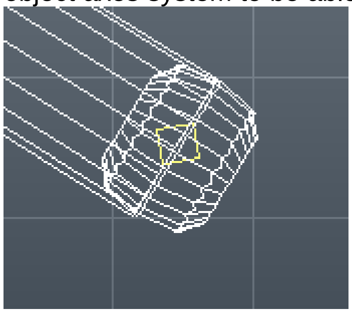
6.13 Tutorial - Subtle Trails of Smoke

This tutorial shows how to create a subtle trail of smoke with help of two dimensional pixel particles. This type of particle is rendered in postprocessing mode after the calculation of the picture has been finished. A depth buffer created during picture calculation is used to determine whether a pixel particle is visible or covered by other objects. In our tutorial we use 4 slightly offset streams of pixel particles to create a thin trail of swirling cigarette smoke. The project file of this tutorial can be found under "..\projects\tutorials\12_cigarette_smoke4p.cmo".



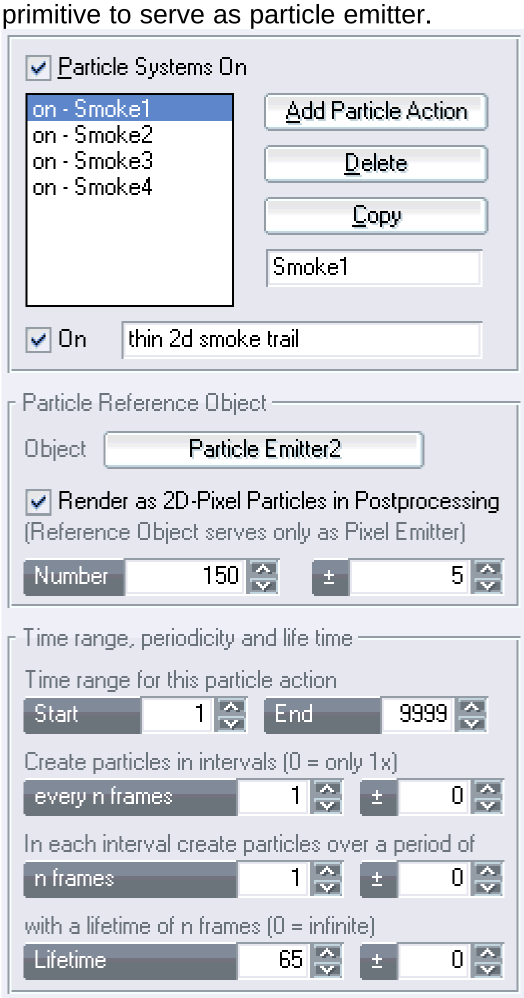
The model of the cigarette can be build very swiftly from a cylinder primitive and a flattened sphere for the cigarette glow. Then a simple procedural stripe material was applied to generate the colored end of the cigarette. The glow was produced by applying a red and yellow mixed fractal material with self illuminating glow effect.

Now let's start with the interesting part of this tutorial - the generation of the particle streams. Although we don't want to create three-dimensional particles - only streams of pixels painted after picture calculation over the already finished image - we still need to specify a three dimensional reference object. This is because the initial position and the movements of pixel particles are calculated using the same principles and physics as the three-dimensional object particles. Therefore, we need a reference object to serve as a kind of pixel emitter and the corresponding object axes system to be able to define a direction vector for the emanation of the particles.



Create a simple box primitive. Rename it to "Particle Emitter". Scale it down and place it in the cigarette glow.

Call up now the particle dialog. Switch on the global button <Particle Systems On>. Add a particle action or simply overwrite the initial "Action1". Rename the particle action to "Smoke1". Switch on the Smoke1-particle action. Because we haven't defined a reference object yet this will automatically call up an object selection window. There we can select our previously created box primitive to serve as particle emitter.



The dialog parameters on the left side of the particle dialog.

After choosing the reference object, switch on the <Render as 2D-Pixel Particles in Postprocessing> button. If you forget this work step then - instead of a thin trail of pixel particles - real copies of our box object are used for the particle action resulting in a thick column of boxes emanating from the glow of the cigarette.

Number of newly generated particles per particle emission action
We want the particle generator to create 150 new particles per particle emission action. Because in each frame of the animation new particles are to be created the amount of totally generated particles soon adds up to high numbers. We define a lifetime of 65 frames for each particle. This will result to a maximum of 65 x 150 = 9750 pixel particles in the animation. After that, the removal of "dying" particles will free memory for the newly created ones. To be able to apply really high numbers (up to millions) of particles is a main advantage of 2d pixel particles in contrast to 3d object particles. If you would try to achieve a similar effect with 3d object particles, e.g. small spheres, the additional model and material data would soon lead to memory problems and infinite render times. On the other hand pixel particles need much less memory and are rendered in postprocessing mode almost in real time into the picture.

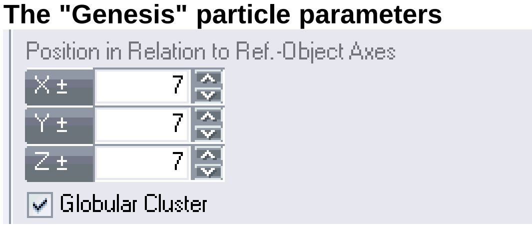
Time Range for this particle action
Start- and end frame of the particle action should cover the whole animation. Consequently we can leave the default parameters of 1 and 999 as they are.

Create particles in intervals
The interval parameter defines a pause after which the particle action starts again to create new particles. We want to create new particles in every frame so just enter 1.

In each interval create new particles over a period of n frames
When a new interval of particle creation begins, new particles will be generated in each frame over a particular time period that is defined with this parameter. To create new particles in every frame we need an interval of 1 and a creation period of 1 frame too.

Lifetime
We want our smoke particles to fade out and disappear after a certain amount of time so that they can be replaced by new ones. So we limit the lifetime of the smoke particles to 65 frames.

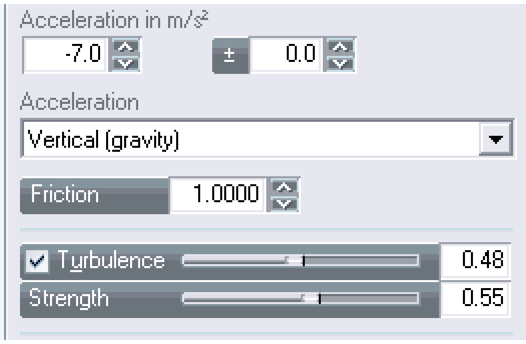
The "Genesis" particle parameters



We want new particles to appear in a spherical space around the origin of the object axes of the particle emitter. Enter a value of 7 for each of the x-, y-, and z-position parameters and switch on the <Globular Cluster> button, so that the values are interpreted as radii.

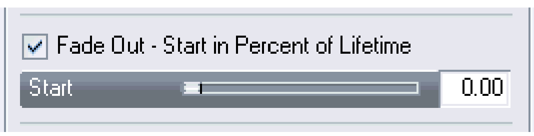
To let the particles drift in a turbulent swirl upwards we need only some gravity, friction and turbulence - no starting velocity or movement direction has to be specified. Therefore we can change now to the "Action" page of the particle dialog to set the corresponding parameters.

The "Action" particle parameters



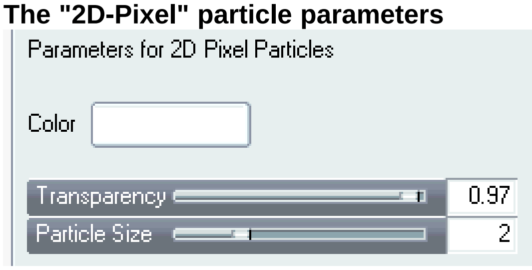
The ascent of the smoke can be realized simply by adjusting a negative vertical gravity of -7.0 m/s^2. Gravity means acceleration, so the particles would become faster with every moment. To compensate for this effect we add high atmospheric friction with a value of 1.

Without a turbulent streaming our smoke animation would become very boring, consequently we switch on the <Turbulence> button and enter a value of 0.48. This parameter controls the frequency of the turbulent streaming which results in particles flowing in narrow curls (high value) or on long drawn-out curves (small value) in the streaming.



Now switch on also the <Fade Out> option, so that the smoke slowly dissolves on its way upwards. The starting point to begin with the fade out is set to 0% of lifetime. This means the fading starts already with the creation of the new particles and is slowly progressing until the end of the lifetime.

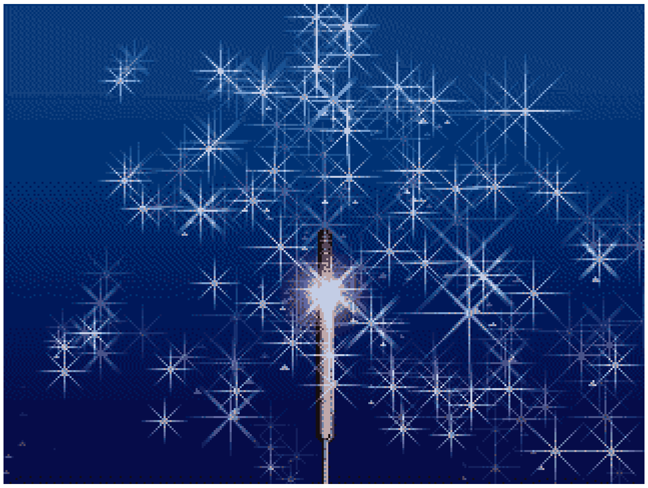
The "2D-Pixel" particle parameters



On this page we can edit the color, transparency and brush size for the painting of the pixels. The color of the smoke is set to white. The transparency is adjusted to a very high value of 0.97. This is because of the massive number of pixels we have to create to achieve a consistent smoke trail. Of course many pixels are overlapping and painted over each other, so high transparency is necessary to keep the impression of a light veil of smoke.
The particle radius defines the brush size for the pixel painting. With a smaller radius the smoke trail becomes very thin and delicate but on the other hand more particles have to be generated to produce the impression of a homogenous veil of smoke. We can do with a radius of 2 for our animation.

Close now the particle dialog and change into Animation Mode. We still need to enter the duration of our animation. This is simply done by moving to the desired end frame of our animation, e.g., by entering frameposition 200 in the corresponding field of the navigation bar 401 at the bottom of the main menu. Render a first animation and if you like the outcome then experiment with more complex smoke trails just by copying particle emitters and corresponding particle actions. First copy the emitter object and displace it slightly within the cigarette glow. Then copy the particle action in the particle dialog and choose the new emitter object as a reference object. Don't forget to initialize each newly added particle action with individual random and turbulence values to vary the particle generation and their movement paths. You can also increase the quality of the smoke veil by switching on the motion blur 409 effect.

6.14 Tutorial - Particle Sparkler



You will find the "CMO" file for this example in the folder "\projects\tutorials\sparkler\sparkler.cmo". The sparkler-animation consists of only an explosion-like particle animation, which constantly emits small triangular particles from the "burnout-point" of the sparkler. You can look at the parameters in the particle-editor if you have loaded the file. The actual trick of the animation is based on the sparkler burning down and the spray of sparks.

The sparkler burning down:

In addition to the stick, the sparkler consists of two identical rotation-objects, which are provided with different materials and overlap, one covering each other. One object is given a bright-gray porous texture - the sparkler before it burns down. The other object is given an even-more porous black-gray surface - the sparkler when it has burned down. The "burned down" object is fractionally smaller, so that it lies slightly under the surface of the un-burned object. The trick of the rod burning down is based simply on scaling and movement. The animation is 61 frames long. We want to animate the burning-down process from frame position 10 to 61 - the initial 10 frames are left for lighting up the sparkler. We therefore go to frame 10. Here use the record key of the navigation bar 401 to record 401 our initial position and size for the outer rod. Now move to frameposition 61 and scale down the outer envelope of the rod (which covers the inside burnt-down rod) to approximately half the length. Then we move the rod so that the bottom ends of both rods are covering each other. Now the upper half of the burnt down rod is exposed, while the un-burned down rod still masks the lower half of the rod. New animation keys are created here automatically by changing the position and size of the object, so you can already start a preview animation. You will see now how the outer envelope slowly moves down and exposes the burnt rod.

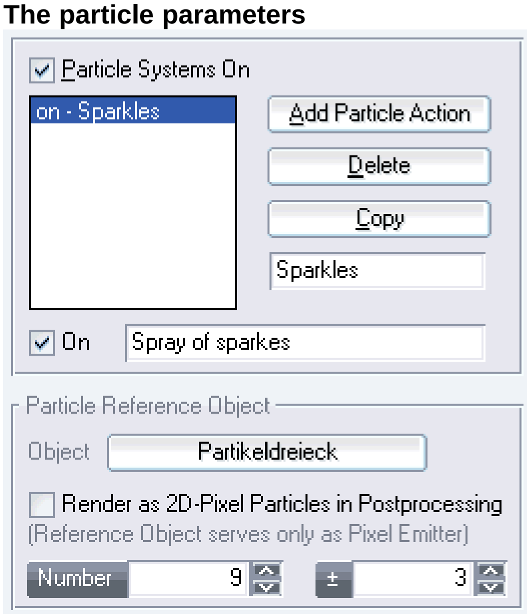
The glowing point:

For the downward-moving glowing point we use a lamp-object with the option "Visible Light-Source 365" switched on. In frame 1 the light-source is placed directly in front of the top of the rod of the sparkler. We want to add a lighting-up effect, so in the first frame we adjust the color of the light to black. Then move to frame 10 in the time line of the animation. Here enter a white light color. This will automatically create a new parameter key and in the animation a color transition will be calculated from the initial dark light color to the bright color in frame 10. Frame 10 is the starting point, from which the lamp object will move downwards, so create a position key on this frameposition for the lamp object using the record button of the navigation bar 401. At the end of the animation (frame position 61), the light-source is moved down to a point in line with the upper edge of the un-burned outer rod of the sparkler. Moving the lamp will automatically create a new position key so we can start a preview animation 259 again to see how the light-source moves with the upper edge of the outer rod.

The spraying sparks:

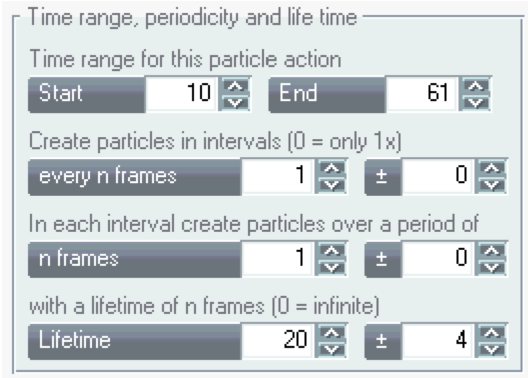
As reference object for the sparks we use the simplest 3D-object - a triangle. You can simply paint such a triangle in the extrude editor and create it without depth.
The triangle is placed in the top of the candle-rod area. We need not define a further key if we arrange the particle-object under the lamp-object in the hierarchy. The reference object then moves down with the light and new particles are always generated and emitted at the height of the glowing point.
To generate this wonderful star-spraying sparking out of the simple triangular particle, we simply use the "Sparkle 273" function, which you can switch on in the Render Options 263 dialog. In order that shining-stars are rendered on all the particles:
1. The material of the particle reference object must be highly reflective.
2. The particle triangles must directly face the light-source, so that the light-source is mirrored in the triangles and produces the shining-point intensity on the triangle surface. Here, the parallel light-source of the scene, which lines up almost front-on to triangles as seen by the viewer, fulfills these conditions.

The particle parameters



Particle Reference Object and Number of newly generated particles per particle emission action

First enter an suitable name for the new particle action, e.g. "Smoke", and switch it on. Because no reference object has been chosen so far, an object selection window will appear automatically where you can select our previously created triangle as the reference object for this particle action. Enter a value of 9 ±3 for the Number parameter, so between 6 and 12 new particles are created in each frame for the particle output.



Time Range for this particle action
The sparkling action begins in frame 10 after the glowing point has lit up and continues up to the end of the animation at frame 61. So enter a Start value of 10 and 61 for the End of this particle action.

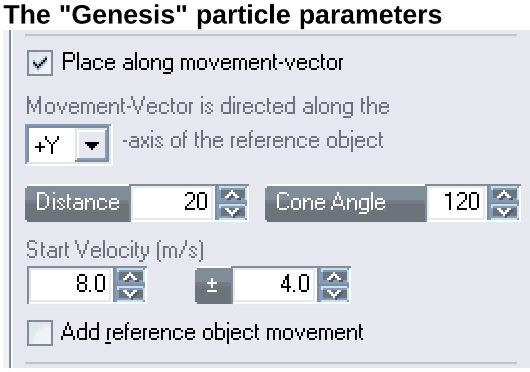
Time Range for this particle action
Start- and end frame of the particle action should cover the whole animation. Consequently we can leave the default parameters of 1 and 999 as they are.

Create particles in intervals
The interval parameter defines a pause after which the particle action starts again to create new particles. We want to create new particles in every frame so just enter 1.

In each interval create new particles over a period of n frames
When a new interval of particle creation begins, new particles will be generated in each frame over a particular time period that is defined by this parameter. To create new particles in every frame we need an interval of 1 and a creation period of 1 frame too.

Lifetime
A lifetime of 20 ±4 frames for each particle is sufficent for this particle action

The "Genesis" particle parameters

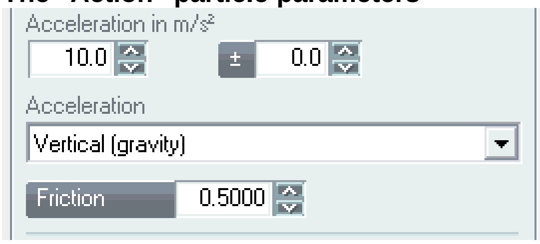


On the "Genesis" page of the dialog we can adjust the parameters for the starting position, movement direction, particle size and growth.

For the starting position we choose <Place along movement-vector>. This way new particles are generated along a line emerging from the origin of the object axes of the reference object. We choose the +Y-axis of the reference object as the direction vector for the initial movement. This will result in particles shooting upwards out of the glowing point. The length of the line along which new particles are randomly created is specified via the Distance parameter. Set a value of 20 for it. Then enter a Cone Angle of 120° for the random deviation of the movement vector. This will change the straight line to an almost spherical outburst of new particles with an opening angle of 240°.

Then enter an initial movement velocity of 8 ±4m/s.

The "Action" particle parameters

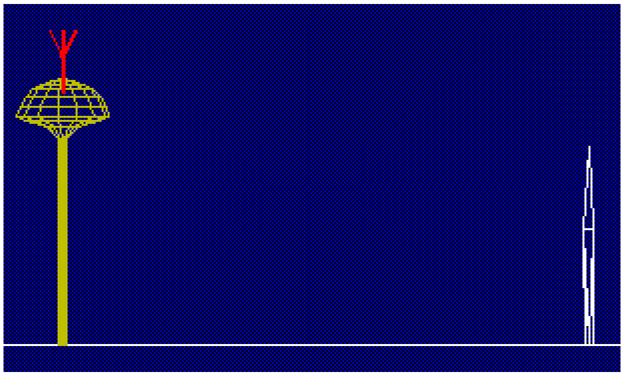


If you render a preview animation now, you will find that the particles will fly away with constant velocity on straight lines defined by the direction of their movement vectors. So we need some gravity and friction to let them slow down and fall on a nice bell-shaped curve down to earth as would happen with real gravity and atmosphere. So simply enter a vertical gravity of 10 m/s^2 (gravity on earth is about 9.81m/s^2) and a friction value of 0.5.

Now start the animation and enjoy the spray of sparks. You could also experiment with the motion blur 409 function to add some blurr to the movements of the sparks.

6.15 Tutorial - Particle Meadow with Dandelions

The objective of this tutorial is a particle generated grass meadow with some dandelions flowers swaying in the wind. The seeds of the danelions are also produced by a particle system. In the final animation the seeds of the plant will detach from the stem and drift away in a flurry. You will find the "CMO" file of this example in the folder "\projects\tutorials\meadow\meadow_anim.cmo". Before you go through this example please load this file and see for yourself the parameters described in the program. Also refer to the Particle Editor 425.



Here you can see the model-structure of this "complex" scene. All that we need is a grass-stalk, which is copied 1000 times for the particle-action, and a rotation-body for the plant's stem. On this plant's stem is a seed, which is also animated with the particle-system.

The particle-action for the grass:

Look at the parameters for the grass production in the Particle editor. The particle-action runs over the whole animation from frame 1 to frame 100. There are 1000 additional grass-stalks generated in the X-, Z- level in an area ±600 area-units around the reference grass-stalk. To bring some disorder to the meadow, the grass-stalks are randomly scaled in an area of 0.6 ± 0.25. The grass-stalks are also rotated - a full ±180 degrees around the vertical Y-axis, but only ±15 degrees the X and Z-axes respectively (higher values would lead to grass-stalks lying horizontally or to them growing over each other). Note: before animating the object-axes of the grass-stalk must be moved down to its base so that all rotations are about these axes. (If the object-axis were to remain in the middle of the object and a rotation is executed the grass stalk would not appear to be attached to the ground.)

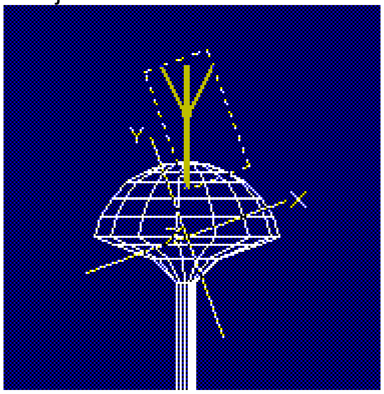
The grass meadow should wave in the wind. We animate the reference object so that it leans a little to right in some keyframes and is upright or leans to the left in other keyframes. We need only switch on the function "Overlay movement of the reference object" for this movement to be followed by the particles.

If you find the meadow to move to uniformly - all grass blades following the same movements - then just copy the grass blade with its animation track, displace it and modify the rotations a little bit and then use it as a new reference object for another particle action. Don't forget to initialize duplicated particle actions with a separate random value, so that the new particles are not created in the same places like the ones of the first particle action.

The two particle-actions of the dandelion:

We need two particle-actions for the dandelion: one for the first 43 frames - in which the dandelion waves gently in the wind like the grass - and a second in which the seeds leave the stem and blow away. The first particle-action runs from frame 1 to 43. 100 copies of the reference-seed are generated and arranged in a hemisphere about the reference object itself, and have their roots in the center of the semicircle. Again, we must move the object-axes of the seed-object to its base. We then obtain a hemisphere for the newly generated particles by rotating ±90 degrees about all 3 axes. In principle that is all. However, the flower waves in the wind exact like the grass so we animate the dandelion-stem the same way as the grass reference object. The seed object is then simply hierarchically 140 subordinated to the stem - this way the seed will automatically follow all of the stems movements. Finally switch on the option "Overlay movement of reference object" so that the seed particles will also follow all of the movements of the seed reference object.

The second particle action: In the second particle action the seeds leave the stem and are whirled through the air. Firstly we copy the first seed particle-action. The remaining frames of the animation (frame 44 to 100) are the frame range. Here, we switch off the function "Overlay movement of reference object". Now it only remains to input the movement parameters. The seeds should detach from the stem when leaning to the right. The +X object-axis of the blossom already leans to the right. You can verify this in the "Rotate Object 211" menu, by changing to the "Selection - Object-Axes" work mode.



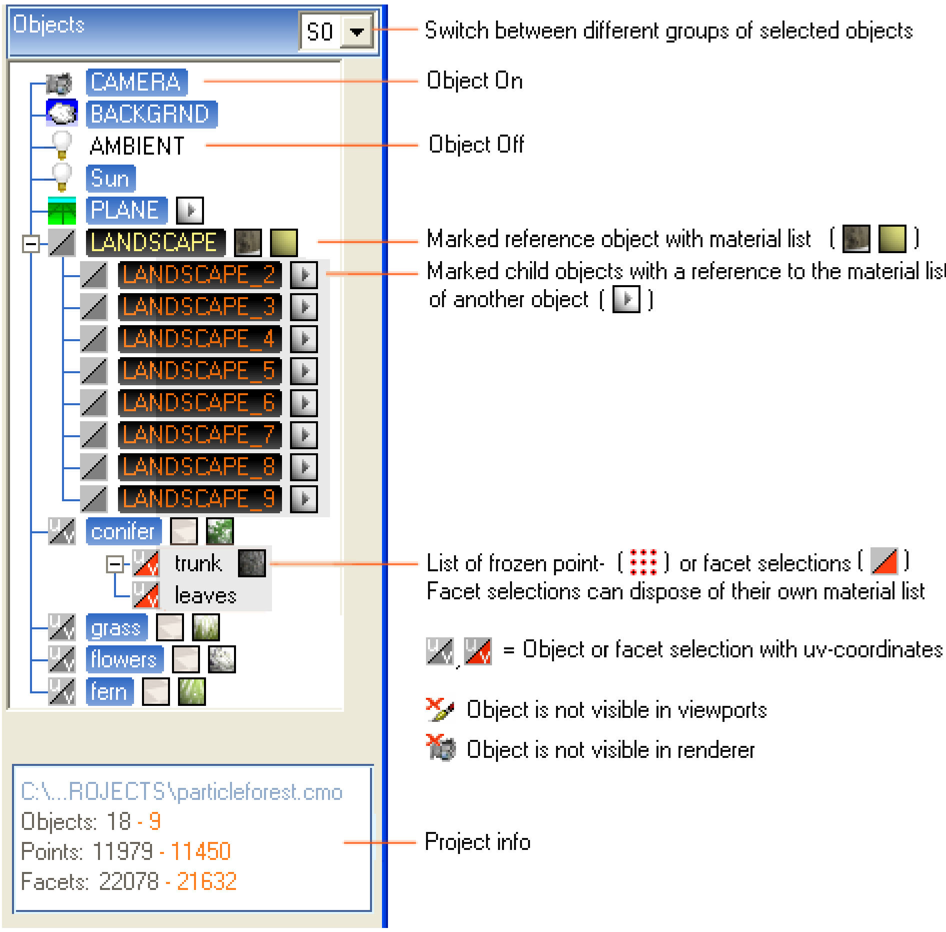
In the same menu we turn the object-axes about 35 degrees around the Z-axis, so that the X-axis is inclined to the right as shown above. Now, in the particle editor, we choose precisely this +X-axis as the movement-vector and enter a starting speed. For the chaotic swirling we switched on the turbulence and the whirl-function.
Do not forget to switch off the seed reference object before rendering the animation 259, otherwise the seed-particles fly from the stem and leave the solitary seed reference object sticking to the stalk.

Part VII

7 Object Selection, Hierarchies and Groups, Copy and Delete...

Everything about object management, groups and hierarchies, reference objects and how to work on individual facets and points.

7.1 The Object Selection Window



The Object Selection window on the right side of the main window. Click with the mouse onto the small bar between the viewport windows and the Object Selection window and pull it to the left or to the right to enlarge or reduce the visible area of the Object Selection window.

In order to work on an object, the object must first be switched on and then marked for editing (either with a mouse click on the object's name in the Object Selection window or by selecting it directly in a viewport window 138). Furthermore, it can be useful to manage several groups of selected objects, so that only those objects of the current work group are shown in the viewports. Or you arrange objects in a hierarchical structure. Using object hierarchies you can easily group your objects, e.g., under a special Group Object 192. This provides a clearer arrangement of objects in the object selection window (child nodes can be hidden) and you can select or de-select a whole group of objects just by clicking on the parent object of that group.
When you begin a new project you will note that there are already several objects displayed in the selection window - in addition to the camera 257, are two lights 340 and the background 374 object. The preset illumination is the light-object "AMBIENT 343" (general area brightness) and the light-object "PARALLEL 351" (a parallel light-source). You can switch on or off the three standard objects "CAMERA", "BACKGRND" and "AMBIENT" but since there can only be only one of these basic scene objects you cannot delete or copy them.
If a normal object or a light object is switched off it will not be included in picture generation, i.e. the lamp is not "on." Nor will a background be drawn when the picture is later generated if the background object is off. The camera is always activated as there can be no picture without a camera. However, you can also switch it off to prevent the camera-symbol being drawn in the

viewport windows.
All object types are prefixed with a special icon, to distinguish them from the "normal" faceted objects.

Camera
Background
Light Source
Volumetric Fire
Plane
Faceted Object
Subdivision Surface
NURBS Patch
Skeleton Bone
Group Object
Analytical Primitive

By the way, double clicking on one of these prefixed icons is another possibility to call up the properties dialog corresponding to the special object type you selected. For instance, if you click on the light symbol in front of a light object then the Light dialog is opened (with the choosen light object preselected in the dialog) and when you click on the background symbol then the Background dialog appears. Double-clicking on the camera symbol opens the Render Options dialog. Clicking on most other icons will call up the general Object Properties dialog.

Object Selection - (Un)Marking Objects or Switching Objects On or Off

- *Left Mouse Click* - Switch on and mark selected object or whole object branch
- *Left Mouse Click plus <Ctrl> key* - Add another object/branch to the current marked selection
- *Left Mouse Click plus key*
  1. Unmark object (if the object is marked)
  2. Switch off object (if the object is not marked)
- *Double Click* - Rename object

While holding the <Ctrl> or the key and the left mouse button pressed you can also drag a framework to enclose all objects you want to add to or remove from the current selection:

- *<Ctrl> key plus Framework*
  1. Switch on all inactive objects within the framework
  2. Mark all enclosed objects
- *key plus Framework*
  1. Unmark all enclosed objects
  2. Switch off all enclosed objects

If you mark an object within a hierarchy 140 tree then the entire object branch with all its children is also automatically marked. (Child objects inherit the movements of their parents, therefore children are always selected with their parents). However, you can switch off individual child objects (e.g., to exclude individual objects of the branch from copy or delete operations). But even if a child object is switched off temporarily in the Object Selection window it will still be a part of the hierarchy structure and therefore, internally, follows all movements of its parent object. The only difference is that it will be hidden in the viewport depiction.
See also - Arranging Objects in Hierarchies 140

Changing the Reference Object in a Marked Selection

If you mark only a single object then this object will automatically be the reference object. This means, all coordinates and object dimensions or the position of the object axes printed in the tool window's parameter fields will reference to this object.
If you mark a hierarchy branch then the reference object will always be the topmost marked parent object of that branch.
If a group of several independent objects or hierarchies are marked simultaneously, then you have

the choice to determine the reference object by clicking on the desired object or hierarchy while holding the <Ctrl>- and -shift key pressed simultaneously. This key-combination can be applied in the viewport windows as well as in the Object Selection window. In the Object Selection window reference objects will be emphasized by a yellow text color (marked objects are drawn in red). In the viewport windows, reference objects are highlighted also through a different representation color 21 (by default the same as in the Object Selection window, i.e., reference object = yellow, marked object = red).

Selecting Frozen Point- or Facet Selections 133

Each faceted object can handle up to 31 different frozen facet- or point selections in which particular selections of marked points or facets 138 are saved. Frozen selections are listed in the Object Selection window in a small list directly under the name of the object they belong to. With a click on the [+] symbol you can open or close a list of frozen selections like you can do it with objects in a hierarchy branch. To restore a facet- or point selection saved previously to a frozen selection you need only to click on the corresponding name of the frozen selection. Point selections are only applicable in point selection work mode while facet selections can be chosen in facet selection work mode as well as in the texture work modes (moving 197, scaling 206 or rotating 211 a texture axes system 197 of a material). This is because frozen facet selections can be applied with separate materials (see example 138).
If you want to (un)mark several frozen selections at a time you can use the same short cuts as when selecting normal objects.

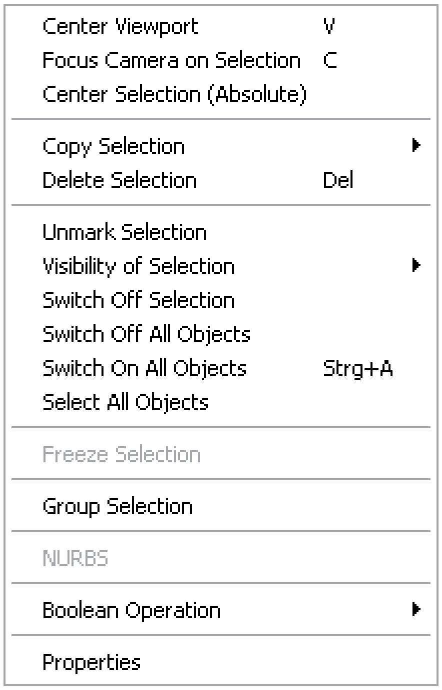
Selecting Materials

Materials assigned to an object or a frozen facet selection are listed in a row of small icons behind the object's name. You can click on a material icon to mark a material, for instance, to select the texture axes system of a material (in texture axes work mode), or, if you want to remove the material with a click on the "Del" key on your keyboard. All other functions regarding material managment are dealt with in the Material 299 dialog.

Switching between Several Groups of Selected Objects

With help of the select box at the top of the Object Selection window, you can create up to 8 different groups of selected and marked objects, and switch to and fro between them.

Popup Selection - Functions to Copy, Delete, Hide or Group Objects



If you click with the right mouse button in a viewport or the Object Selection window then a popup menu appears that gives, among other things, some additional functions for copying, deleting or grouping objects. There are two different selection modes:
Copy "Selection" - If you click in an empty area of the Object Selection window (or viewport) to open the popup menu, the function you choose will be applied to the whole selection of currently marked objects. (The entries of the menu are appendixed with a "Selection", for instance, "Copy Selection").
Copy "Name of Object" - If you click with the right mouse button directly on the name of an object in the Object Selection window, only this object will be copied or deleted - other marked objects, for example a branch of subordinated children, will be ignored. (The entries of the menu are appendixed with the name of the object, for instance, "Copy Object1").

Center Viewport - Moves the viewport over a selection of marked objects, facets or points.

Focus Camera on Selection - The camera is aligned to focus the center of the presently marked selection.

Center Selection - A selection of objects, facets or points is moved to the center of the viewport.

Copy Selection - creates a copy of the marked selection. This can be an identical or mirrored copy of the model and animation data or even a copy which transfers an animated and deformed object state to the model data of a new object:

- **Copy Model and Animation Data -** An identical copy of the model and animation data of the original object is produced.
- **Copy - Mirror at Horizontal World Axis -** The copied object is mirrored about the central Y-coordinates.
- **Copy - Mirror at Vertical World Axis** - The copied object is mirrored about the central X-coordinates.
- **Copy Current State to Model Data -** There are several ways objects can be deformed in an animation - i.e. by applying a skeletal deformation or using one of the animated bend or twist deformation tools. However, these deformations don't change the original model data of an object - they are performed rather as a part of the render pipeline immediately before rendering. After each work step an internal copy of the original model is created, deformed and passed on to the

render engine. In some situations it would be useful to apply these deformation tools not only for the animation process but also for the modelling part, for instance, if you want to bend an object permanently to alter the shape of the model. For this purpose you can use the "Copy Current State to Model Data". It copies the data of the object in its deformed state to the model data of a new object - all animation data is ignored and will not be transfered.

Delete Selection - All marked objects, point- or facet selections or materials are deleted. This function can also be applied by pressing the "Del" key on your keyboard. There is a simple logic in deleting different types of objects or selections: If a frozen selection is marked in the Object Selection window then first this frozen selection entry will be removed from the object (points and facets of the selection remain marked). If you press the "Del" key a second time then the marked points and facets will be removed from the object. Finally, if you press the "Del" key another time, the whole object will be deleted.

Unmark Selection - The current selection of marked objects, facets or points is canceled.

Visibility of Selection - There are two options to hide objects in the viewport depiction or for the final rendering. In contrast to objects which are switched off in the Object Selection window these hidden objects are still active, for instance, you can select a hidden object by clicking on its name in the Object Selection window and then move it around in the viewport by repositioning the bounding box painted around the hidden object.

- **Hide in Viewports** - Objects are hidden in the viewport windows but are still visible in the final renderings. This option helps to clear the viewport windows from already constructed objects when creating and working on new models. Objects not visible in the viewport depiction are indicated by an icon added to the name of the object. By clicking on this icon the object becomes visible again. The same result can be achieved via the menu entries of the popup menu or by setting the corresponding object properties 290 in the Object Properties dialog.
- **Hide in Rendering** - Objects are visible in viewport depiction but not in the final rendering. This way you can use additional objects to aid construction work (i.e. blue prints projected onto planes to serve as construction templates) and hide them automatically for the rendering. Objects not visible in the final rendering are indicated with an icon added to the name of the object. By clicking on the icon (or by choosing the corresponding menu entry in the popup menu or the Object Properties 290 dialog) this object property can be removed again.

Switch Off Selection - All marked objects are hidden and deactivated. To activate these objects again you must switch them on in the Object Selection 133 window.
Switch Off All Objects - All objects are deactivated and hidden.
Switch On All Objects - Activates all hidden objects, including camera, background and light objects
Select All Objects - All activated objects are marked automatically for editing.

Freeze Selection - A selection of previously marked points or facets is saved to a frozen selection. When you choose the "Freeze Selection" menu entry a little info dialog pops up where you can enter a specific name for the selection. Frozen selections are listed in the Object Selection window in a small list directly under the name of the object they belong to. To restore a frozen facet- or point selection you need only to click on the corresponding name of the frozen selection in the Object Selection window.
Replace Frozen Selection - If you want to replace a frozen selection by a new choice of marked points or facets you can use this menu function to do so.
Delete Frozen Selection - You can delete all or individual frozen selections from the selection list of the currently marked reference object.

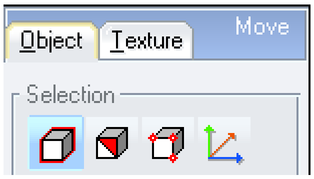
Group Selection - For clarity it is advisable to group objects belonging together under a Group object 192. Choosing the "Group Selection" function from the popup menu will group all currently marked objects in a hierarchy under a newly generated Group object.

NURBS 171 - Opens a list of additional functions to modify a marked NURBS object.

Boolean Operation 233 - Joins 2 marked faceted objects applying a Boolean Operation (Join, Union, Subtract, Triangulate).

Properties - Calls up the Object Properties 289 dialog

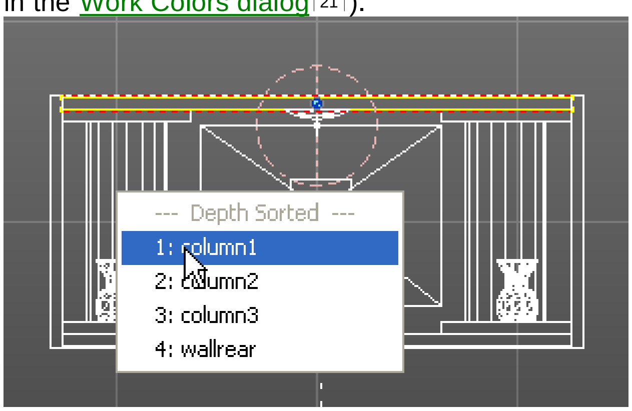
7.2 Selecting Objects, Facets or Points in the Viewport



Objects, facets or individual points can be selected simply by mouse clicking on them in the viewport windows. Hold the <Ctrl> key pressed simultaneously for adding objects to the selection or the -shift key to remove objects again from the selection. Through the first three "Selection" buttons at the top of the toolbox you can choose between object, facet or point selection mode. The fourth button - the object axes selection - is only available in "Move Object" and "Rotate Object" work-mode and enables you to move or rotate the axes system 8 of a selected reference object 138.
Individual facets, points or the object axes can only be selected and worked on in Model Mode. If you change to Animation Mode then only object selection will be available. But this doesn't mean that you cannot animate on a per-point-basis. If you want to animate deformations of an object's surface you can make use of the skeletal deformation 242 functions.

Selecting Objects in the Object Selection Window
You can also conveniently select objects in the Object Selection 133 window, which provides many other additional functions, e.g., to switch objects on or off, to copy or delete them, to rename or to group them or to arrange them in hierarchies. Some of these functions can also be accessed from a popup menu 133 by clicking with the right mouse button in a viewport window.

Selecting Objects for Processing
If you want to work on an object (e.g. move 197, scale 206 or rotate 211 it), you must first mark the object. You need only click on an object with the left mouse button in the viewport window to select individual objects. Marked objects are highlighted in a different grid color (which can be determined in the Work Colors dialog 21).



If the object you require cannot be readily selected because several other objects overlap it, then a depth sorted popup menu appears in which you can choose the desired object.

Selecting several objects - If you want to work on several objects simultaneously, then you press the <Ctrl> button on the keyboard in addition to the mouse click with the left mouse button. While holding the <Ctrl> and the left mouse button pressed you can also drag a framework to enclose all objects you want to mark.

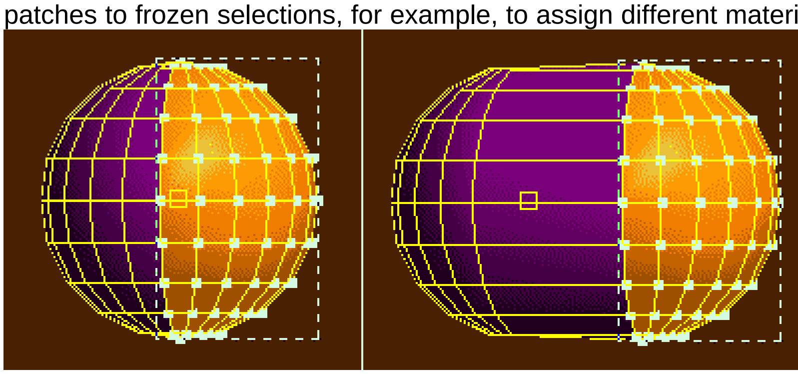
Deselecting objects - In the same manner, except this time pressing the -shift button instead of <Ctrl>, allows objects that have been selected for processing to be deselected again.

Changing the Reference Object in a Marked Selection

If you select only a single object then this object will automatically be the reference object. This means, all coordinates and object dimensions or the position of the object axes printed in the tool window's parameter fields will reference to this object.
If you mark a hierarchy branch then the reference object will always be the topmost marked parent object of that branch.
If a group of several independent objects or hierarchies are marked simultaneously, then you have the choice to determine the reference object by clicking on the desired object or hierarchy while holding the <Ctrl>- and -shift key pressed simultaneously. This key-combination can be applied in the viewport as well as in the Object Selection window.

Selecting Individual Facets or Points for Editing

You can also mark the individual facets and points of an object or a group of objects for editing in the work modes (Move Objects 197, Scale Objects 206, Rotate Objects 211, Work on Objects 227 and Edit Skin and Bones 242). First mark the objects that you want to work on, either by clicking on an object in object selection mode or by clicking on the corresponding object name in the Object Selection 133 window (this way you don't have to switch to and fro from object mode to facet- or point work mode and vice versa). Then change via the selection buttons in facet- or point work mode. Now you can select individual facets or points in exactly the same manner as before with objects. So just click an individual facet/point in the viewport to mark it. By simultaneously holding down the <Ctrl> button, you can select additional facets/points. Holding down the <Shift> button instead results in unmarking points.
You can also drag a framework with the mouse while pressing the <Ctrl> or <Shift> button to enclose several facets/points simultaneously for (de)selecting.
NURBS patches - With NURBS patches - instead of selecting individual facets - you select always whole patches lying between their surrounding control points. This way you can even add NURBS patches to frozen selections, for example, to assign different materials to them.



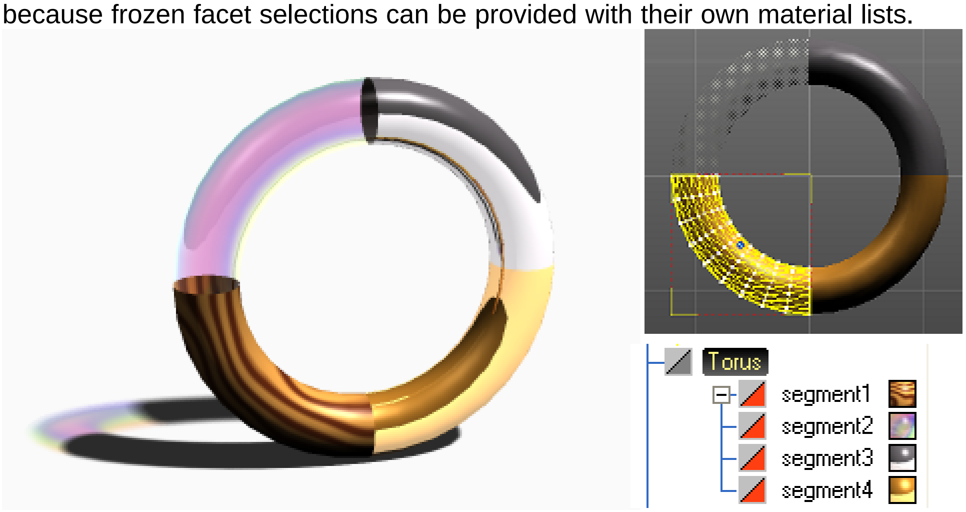
Example of a facet selection: The facets of the right half of the sphere have been marked and moved to the right, thus changing the shape of the sphere to an oval capsule.

Overlapping facets and points - let the selection float around
Elements that lie hidden behind other facets can be selected most easily in the camera viewport. There you can circle around a selection (hold left and right mouse button pressed and move the mouse in the camera viewport) and zoom in onto it until the desired element is in the forground.

There is another function that makes it possible to select overlapping facets and points. Holding down the mouse wheel button and turning simultaneously the wheel will run the last facet or point selection up or down the object (alternatively you can use the <+>- and <-> keys on your keyboard). This way you can (de)select a point at the front of an object that overlaps the actual point you wanted to (de)select and then, pressing the mouse wheel button and turning the wheel, the point selection moves to the desired point.

Frozen Point- and Facet Selections

Each faceted object can handle up to 31 different frozen facet- or point selections in which particular selections of marked points or facets 138 are saved. Frozen selections are listed in the Object Selection window in a small list directly under the name of the object they belong to. With a click on the [+] symbol you can open or close a list of frozen selections like you can do it with objects in a hierarchy branch. To restore a facet- or point selection saved previously to a frozen selection you need only to click on the corresponding name of the frozen selection. Point selections are only applicable in point selection work mode while facet selections can be chosen in facet selection work mode as well as in the texture work modes (move, scale or rotate texture axes of materials). This is because frozen facet selections can be provided with their own material lists.



The illustration shows a simple torus object with 4 different materials applied to the 4 frozen facet selections created for the torus object. The picture on the top right shows the depiction of the torus in a shaded viewport window and under it you can see how the frozen facet selections and their materials are displayed in the Object Selection 133 window.

Create, Replace or Delete Frozen Selections

The functions to create, replace or delete frozen selections 133 can be accessed via the popup menu 133 that appears, when you make a right mouse click in a viewport or the Object Selection window.

7.3 Arranging Objects in Hierarchies

Object hierarchies, in which objects are arranged in a tree like structure, help to manage groups of objects belonging together by putting them together under a single parent. The relationship between the objects in a hierarchy tree is also an essential help for building complex animation movements, where parents take along their children with their movements.

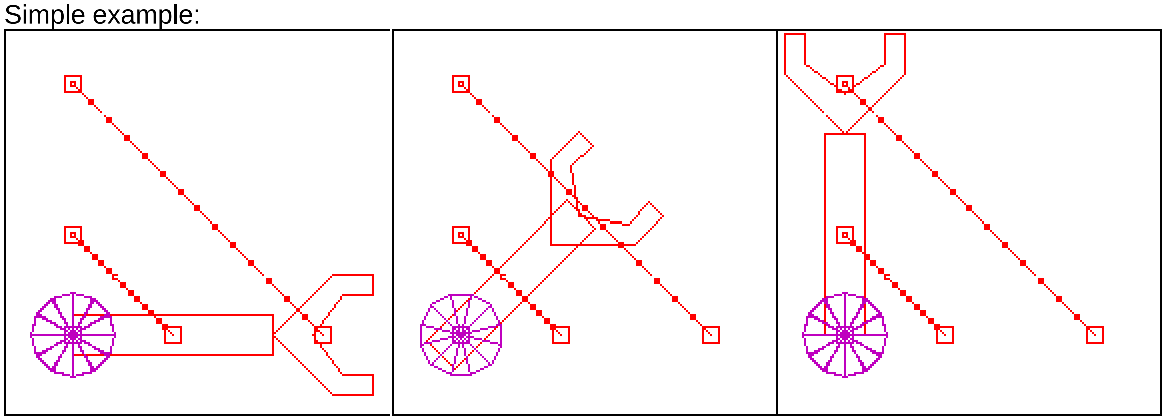
Grouping of Objects in Hierarchies

Using object hierarchies you can easily group your objects, e.g., under a special Group Object 192. This provides a clearer arrangement of objects in the object selection window (child nodes can be hidden) and you can select or de-select a whole group of objects just by clicking on the parent object of that group. For instance, if you have created a car consisting out of several hundreds of

objects, simply link them under a Group Object "car". Then, every time you want to move that car, you only need to mark the "car" Group Object and move it to its new destination, all children will follow automatically.
Incidentally, you can automatically create a new Group Object while simultaneously linking a previously marked selection of objects under it with help of the "Group Selection 133" entry of the popup menu.

Hierarchical Animation - Child Objects Inherit Movements from their Parents

Hierarchical structures are essential for animating 400 complex movements. Take, for example, an industrial-robot that is assembled out of several different rotatable arms and joints. It is almost impossible to animate this robot if you have to move all the robot's parts to their respective desired positions individually, for every movement of the robot. You could certainly mark all necessary parts in a group and move or rotate them to their final position, but the following problem cannot be solved in this way: If, in an animation, you move a non hierarchical robot arm that is built from several different parts, from one position to another, then every single part follows its own animation-path.
Simple example:

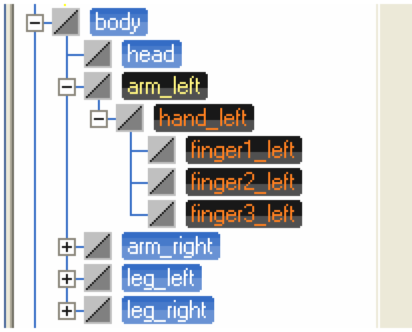


The arm pictured above should rotate clockwise around itself through 90 degrees from the start position shown in the left picture. The cylinder is used as the center of rotation. The final position is shown in the right illustration. When playing now the animation you can see, that all parts of the robot arm move on straight movement paths from the starting position to the end position, instead of circling around together with the cylinder rotation. In the center picture you see an intermediate position and an undesirable result. Moving on their own paths in the direction of the target position, the objects no longer form a single unit, but instead overlap each other or drift apart.
This type of problem is easily overcome, however, with help of object hierarchies. Individual objects are arranged hierarchically under other objects and then their movements are dependent on the parent objects. The precise concept looks as follows:

- **Each child object performs all movements of the parent object immediately superior to it, as if it were an integral component of this object.**
- **Child objects nevertheless retain their freedom of movement, and therefore can still execute additional movements independent of their parent object.**

In the simple example pictured above it results in the following: The arm and hand, which are subordinate to the cylinder, correctly rotate with the cylinder. However, you could, for example, additionally have the hand execute a rotation about its vertical axis - without it influencing the movement of the parent arm or cylinder.

Use the Object Selection Window to Arrange Objects in a Hierarchy



In the Object Selection 133 window the object hierarchy appears as a tree-structure in which child objects are always linked to their parent object. Each child object can have only one parent object. However, an object can have any number of child objects. This works in exactly the same manner as the tree structure of a file manager with its folders and files. For clarity, you are also able to open or close object nodes via the <+> and <-> buttons to hide or show child objects in the tree structure. A double click on the <+> button opens all child nodes of that parent at the same time.

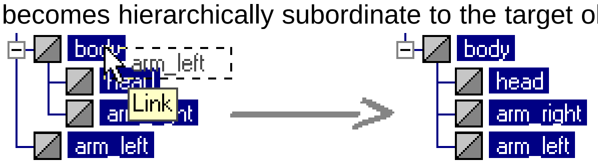
If an object is hierarchically subordinate to another object, then it performs every action of the parent object. If you mark a parent object for editing then the entire object branch with all its children is also automatically marked. Even if a child object is switched off temporarily in the Select Object dialog it will still be a part of the hierarchy structure and therefore, internally, follows all movements of its parent object. During preparation of an animation 400 you can make use of this feature to accelerate the work flow and to save some time on depiction. Then you could switch on only those objects in the hierarchy that are located prior to the branch and then move, rotate or scale them in the individual keyframes. If you then switch on all subordinate objects again in the Select Objects dialog and start an animation preview, you will see that the objects previously switched off actually all take part in the movements of the root object.

Object hierarchy in combination with camera- and light objects

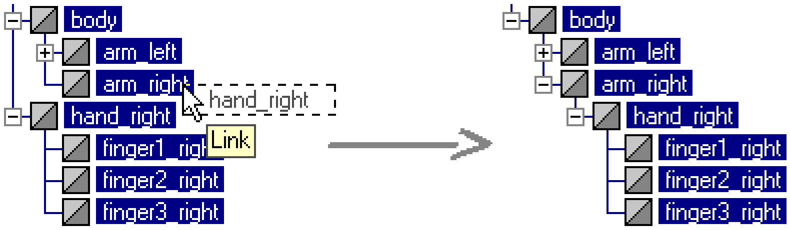
Even positional light 340 objects and the camera 257 can be part of a hierarchy. For example, a camera that is hierarchically subordinate to an aircraft automatically follows all the movements made by the pilot. This also applies to light sources such as headlights, which are hierarchically subordinate to vehicles. Objects that cannot be moved, such as parallel and ambient lights, as well as the background object, cannot be made part of a hierarchy.

Build Object Hierarchy => Drag and Drop object name

Arranging objects in a hierarchy is as simple as moving files in an ordinary file manager from one folder to another. If you want to subordinate an object or a whole object branch under another, select the relevant object or branch with the left mouse button and hold it down while dragging the object name to the target object. The 'Link' symbol under the mouse pointer indicates if a valid target object is found and you can release the mouse button. Thereupon the selected object or branch becomes hierarchically subordinate to the target object and the new hierarchy tree is drawn.

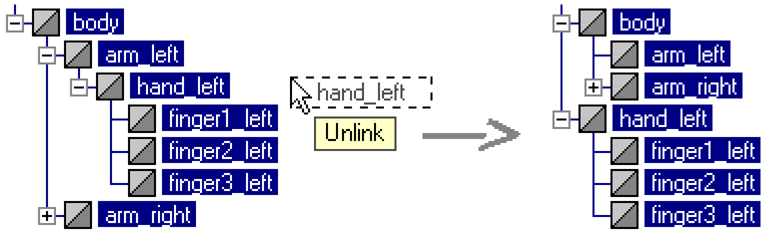


In the example depicted above you can see how the 'arm_left'-object is dragged and dropped onto the 'body'-object. On the right side you can see the result with the subordinated 'arm-left'-object.



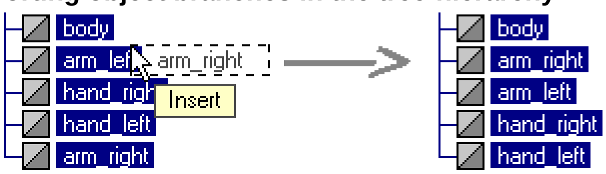
In the example above a whole object branch ('hand_right' with subordinated 'finger_right' objects) is linked to the 'arm_right' object, simply by dragging the 'hand_right' object onto the 'arm_right'-object.

Dismantling object hierarchies



The procedure for dismantling an object branch from a hierarchy is exactly the same as previously described in arranging objects under each other, except that, instead of a target object, you drag objects to any empty field in the selection-dialog. The 'Unlink' symbol below the mouse pointer indicates a valid place to drop the object branch. Thereupon the object branch is removed from the hierarchy and placed at the end of the object list once more.

Sorting object branches in the tree-hierarchy



You can move root objects up and down to change the order in which the objects are listed in the selection window. No change of the hierarchical tree structure is intended - it's just a sorting operation to clarify the arrangement. Again all you have to do is simply drag a root object branch and drop it on a position between two other root objects. All valid places are indicated by the 'Insert' symbol below the mouse pointer.

Hierarchical Independent Animation

What happens if you subordinate previously animated objects under a hierarchy, that may be also has been animated already? Read the corresponding chapter on animation - Hierarchical Independent Animation 401.

Part VIII

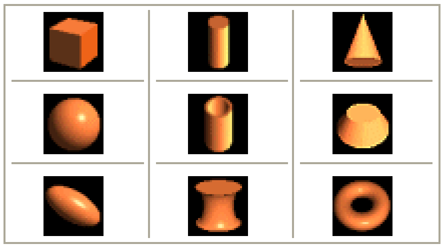
8 Create Objects

All you need to model your own 3D-worlds

- **Primitives** 146
  Create basic shapes at the touch of a button
- **Analytical Primitives** 147
  Mathematical defined basic shapes
- **Extrude Editor** 149
  Cutting out objects as with a woodworking saw
  Spiral Extrude Objects 155
  Tube Objects 156
- **Sweep Editor** 158
  Create lathe objects
  Circular-, Ellipsoid- or Torus Templates 160
  Sweep Objects with Wavy Surfaces 161
  Spiral Object 162
- **SDS - Subdivision Surfaces** 163
  With SDS-deformation, low resolution objects are transformed to a higher resolution model with organically-curved shapes
- **NURBS - B-Spline Patches** 171
  Organic shapes build from deformable patches and cylinders
- **3D Text Objects** 174
  3D text for logos or film titles
- **Landscapes and Planets** 176
  Create your own worlds
- **Plane Object** 186
  The Fundament for most projects
- **Functions Editor** 187
  Visualize mathematical functions
- **Billboard Projection Plane** 190
  A simple flat object to serve as a projection plane for masked bitmaps or picture sequences
- **Group Object** 192
  Group objects help to manage object groups or can be used to serve as reference points in animation routines

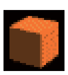
Bones 242 - See "Edit Skin and Bones - Create a Skeleton and Allocate Skin Points"

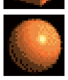
8.1 Primitives

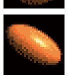


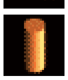
Choose the object menu's "primitives" function or click the  -button to see a choice of primitives for facet-based objects.

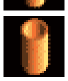
All of these could of course be constructed using the extrude/rotate editors, but it is often more convenient to make them by pressing a button. Unlike objects produced by the analytical primitives 147 function, these are based on triangular polygons. This means that they can be more readily modified using functions like scaling, deforming, blending with other objects, deleting facets etc. Once you selected the object type, a small menu allows you to set up the parameters. Apart from the dimensions, you can also set the objects resolution, i.e. the number of points and facets of the object to be generated. Note that higher resolution mean slower rendering, up to the point where you can't even handle the object because it takes up too much of your computer's processing power to calculate all the redraws. As a rule of thumb, you can get away with a low resolution when the object is small relative to the size of the picture, i.e. when it is either tiny or very far away from the camera. Also, interpolation 290 works much faster than high object resolution, so you might use this function with an object consisting of a minimal number of facets to reduce the problem. Practice makes perfect here...

 **Block** - enter the dimensions to generate a block.

 **Sphere** - enter the radius and resolution to generate a sphere.

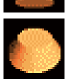
 **Ellipsoid** - enter the radius for x, y and z.

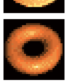
 **Closed cylinder** - define a cylinder by its radius and height.

 **Open cylinder** - two radii and the height are needed, where the difference of the radii defines the wall thickness of the cylinder. The smaller radius entered is always the inner radius, while the larger value is used for the outer radius. If both are the same, the resulting cylinder will be hollow but without having a wall thickness.

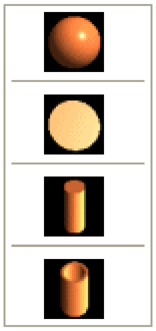
 **Hyperboloid** - a cylinder with a tapered "waistline". Enter two radii here, the smaller of which defines the "waist" thickness.

 **Cone** - defined by base radius and height.

 **Truncated cone** - like cone, but a second radius for the top surface is needed.

 **Torus** - defined by its inner and outer radius.

8.2 Analytical Primitive Object



Selecting the "Analytical Primitives >" entry under "Objects" in the-menu bar or  in the button strip calls up the selection of analytical primitive objects.

Analytically described objects are different from the objects that are designed with the other editors. Until now, all the objects that you have so far designed have been constructed from one basic-component - the triangular polygon. With a large enough number of triangles this enables practically any form to be approximated and constructed.
The disadvantage of this method is obvious, however: a high management and calculation expenditure in depicting the objects. If, for example, you want to represent a sphere you need a large number of triangular polygons to approximate the appearance of the sphere. Of course, there are procedures to calculate the shading of the surface to make it appear as if it is really rounded (and these have been built into the program), but nevertheless there remains the flaw of the angular outline and the high calculation expenditure on depiction.
In this sub menu you have the opportunity to create some other basic shapes in addition to the triangle. You can, for example, create a sphere as an object that is constructed as a basic object with a center and radius, instead of polygons.
When you represent this object later in raytracing-mode 264, the sphere can be calculated in a very short time, since just the basic object has to be calculated for intersection by the viewing ray. A sphere that could be approximated out of 1000 polygons also requires at least 1000 times as long to calculate the picture. (This is not entirely correct, as, through optimization procedures, not all polygons are checked for intersection with the viewing ray.) This is only valid, however, for the time-consuming high quality raytracing-mode that creates photo-realistic pictures - with real reflections, shadows and transparency.
All the other depiction modes implemented in this program (which are faster by far than raytracing) are based on depicting polygons.
To draw the analytically formulated objects in these other depiction modes, an additional polygon-based version of the object is generated. In this manner you can then work with these objects, as with the other objects that are based on triangular polygons. However, the number of the facets out of which an analytic object is approximated remains low.
Furthermore, it is of consideration that the polygonal depiction of an analytic object only acts as an illustration of the basic object. These objects can be rotated and moved in the same manner as all other objects. When scaling, however, they are subject to certain restrictions.
This is made clear by a sphere, for example, which if scaled along the X axis is no longer a sphere, but becomes an ellipsoid that can no longer be described simply by its center and a radius. Therefore, analytical objects will only be scaled, if this operation does not conflict with the mathematical description of the object's shape. All objects can be increased or reduced symmetrically, of course. Furthermore, cylinder objects can be scaled symmetrically along their base area as well as along their longitudinal y-axis.

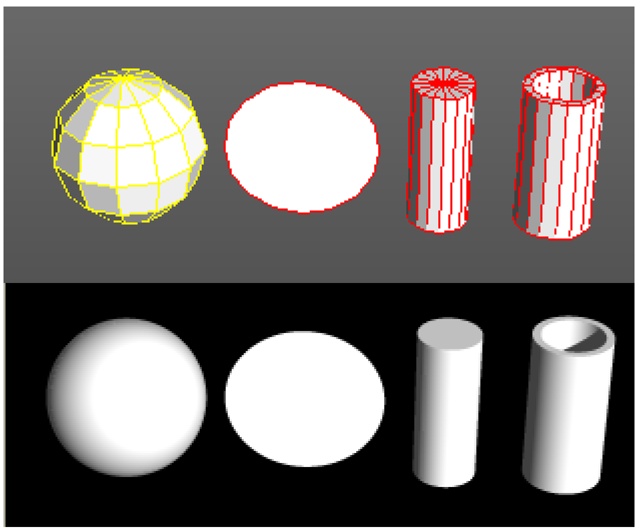
By choosing the corresponding menu entry, a dialog appears and you can construct the following basic objects:

 **Sphere:** For the definition of the sphere, only the desired radius is input.

 **Circular disk:** The statement of the radius is also sufficient for the circular disk.

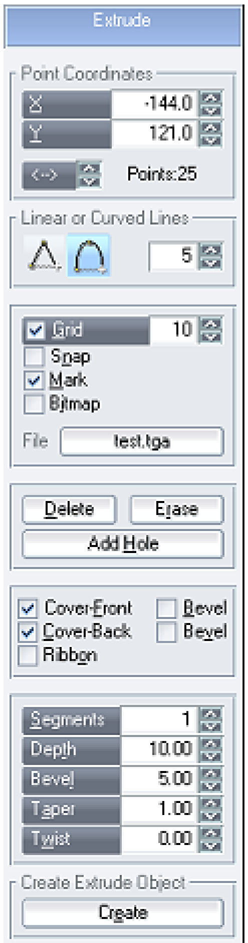
 **Closed cylinder:** You can edit the height of the cylinder as well as the radius of the base.

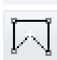
 **Open cylinder:** In addition to the radius and cylinder height, a second radius is defined. The difference between the two radii results in the wall thickness of the cylindrical tube. The smaller value therefore always becomes the interior radius and the larger value the outside radius of the cylindrical tube. An open tube without wall thickness is generated if the interior radius is same as the outside radius.



In the illustration you see the facet-based depiction in Hidden line work mode of the 4 basic object types available and, in comparison, how the same objects are represented by the raytracing 264 procedure.
To create an object, you operate the **<Create>** button.
It should be mentioned in passing that analytical objects are marked by a different icon before their object name in the Select Objects dialog 133.

8.3 Extrude Editor



 - Menu "Objects - Extrude"

The toolbox is replaced by the selection shown above. Simultaneously, the viewport windows are replaced by a window that now serves as an area on which to construct a template of your object. The menu bar and button-strips are also changed, restricted to only the functions needed here.

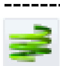
The new functions in the menu bar and button bars:

File

Save Template
Load Template

Quit

Options

 Spiral 155

Tubes 156

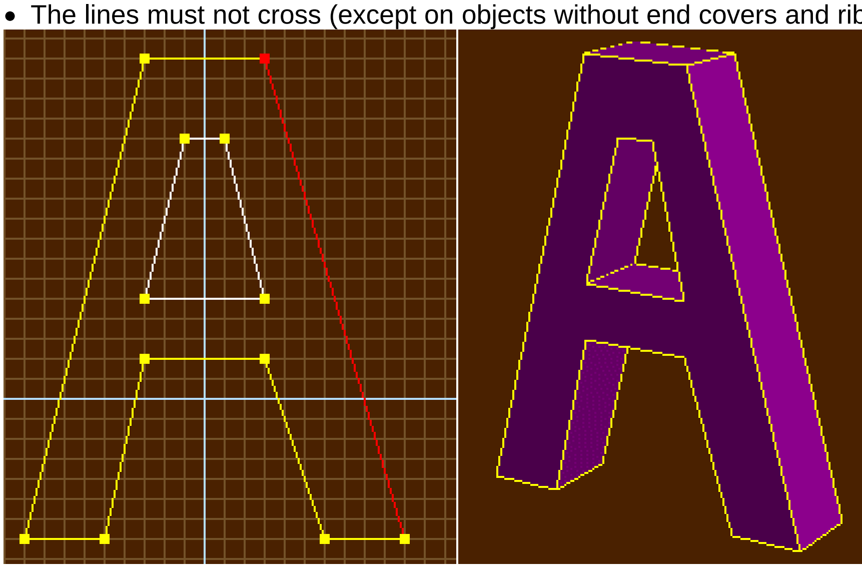
Back to Main Menu

Drawing a Pattern for an Object

The extrude editor enables objects to be generated through the following mechanism: You have the facility, to set up a drawing with the mouse in the viewport, and then provide it with a depth. Cutouts can be made, as with a woodworking saw.

If you go over to the viewport with the mouse, the mouse pointer is transformed into a crosshairs with which you can place individual points. These are then automatically connected with the help of rubber-band technique. It is important to take note of the following rules:

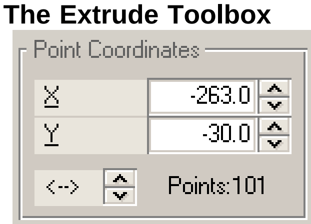
- At least 3 points must be drawn, which must not lie in a straight line.
- The lines must not cross (except on objects without end covers and ribbon objects).



The picture shows an extruded "A". The outer polygon is always drawn first. By inserting additional polygons you can easily cut holes into the shape. Look at the template on the left side of the picture. The current polygon you are working on is highlighted (the yellow one) but you can always change between the individual outlines just by selecting a corresponding point on the desired polygon. The red line indicates the two points between which the next point to set will be connected. The red line will then be deleted and the two points will connect to newly inserted point. You can move the red line around the drawing with the arrow buttons located beneath the point coordinates box or just select any point you wish on the poly to get to the position, where you want to insert new points.

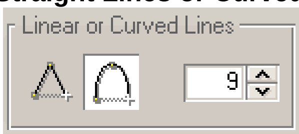
You can change a point's position, by selecting it with the mouse and moving it around while holding the left mouse button pressed. It is also possible to input the x-, y-coordinates of the selected point directly with the keyboard.

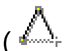
The Extrude Toolbox

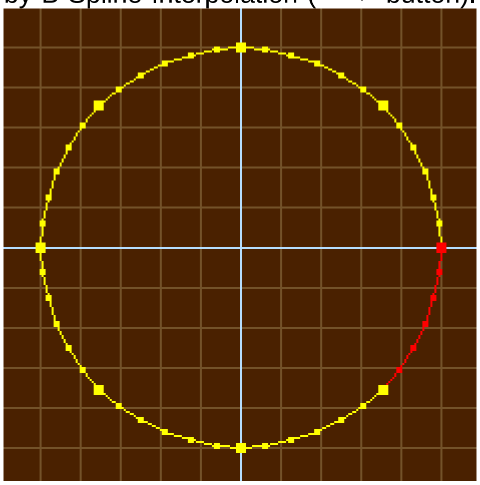


The position of the current point is stated at the top of the toolbox. Beneath the coordinates you find the arrow buttons to move the point selection around the drawing.

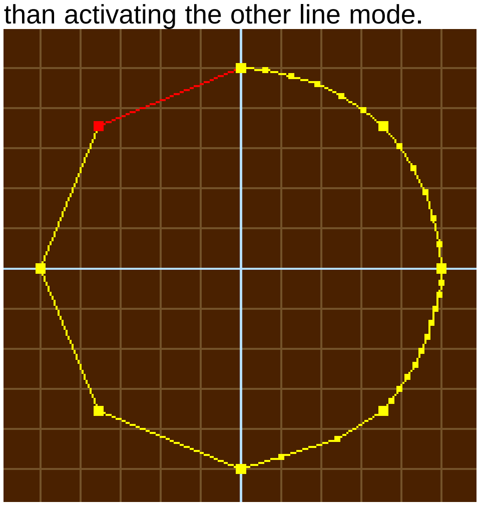
Straight Lines or Curved Line Segments



You can choose between straight-line segments ( -button) or curved line segments interpolated by B-Spline-Interpolation ( -button).



If you work in B-Spline mode and at least 3 points are set, then these points define a curve approximated by a set of additional points. The picture above shows a circular template, consisting only out of 8 points drawn in B-Spline mode. 5 additional points were inserted between each line segment to approximate the curved line segments. The amount of additional points used to approximate the curve segment can be input in the edit field adjacent to the  -button. You can change the number of points any time, just by selecting the corresponding line segment and entering a new curve resolution for that segment. You can even decide to change a curved line segment into a straight line and vice versa just by selecting the corresponding line segment and than activating the other line mode.



The picture demonstrates the flexibility of the template management. The same outline consists of 8 points, only this time with different point resolutions for the curved segments on the right side and the line segments on the left side had been switched over from B-Spline mode to straight lines.

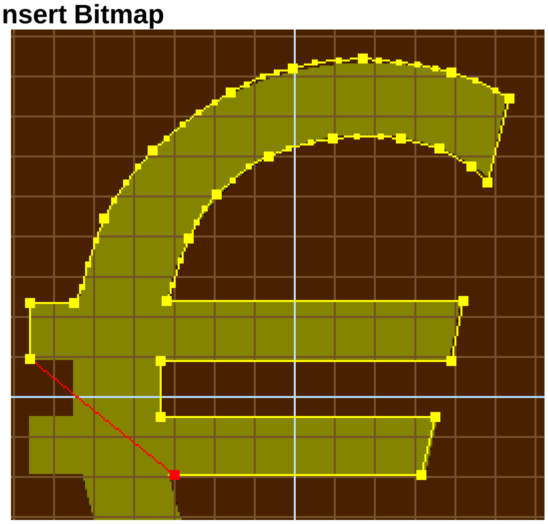
Grid, Snap Function and Marker

A grid screen can be switched on with the **<Grid>** button to simplify orientation and serve as a pattern for the Snap function. The spacing between grid lines can be input in the edit field next to the <Grid> button.

The snap function is switched on with the **<Snap>** button and ensures that points are positioned only at intersections of the grid lines.

If the **<Mark>** button is switched on, then each set point is marked. This enables superfluous points to be easily identified. They then can be deleted and the object simplified. With fewer points it follows that calculation time will be shorter when representing the objects.

Insert Bitmap



Activate the **<Bitmap>** option to place a bitmap in the extrude editor's work space, if you want to construct an object by tracing its outline. Click the **<File>** button of the file select box to load the bitmap you intend to trace. The screenshot shows how you could create an object of the Euro currency symbol by tracing a bitmap. This bitmap tracing function is very useful for creating objects from company logos.

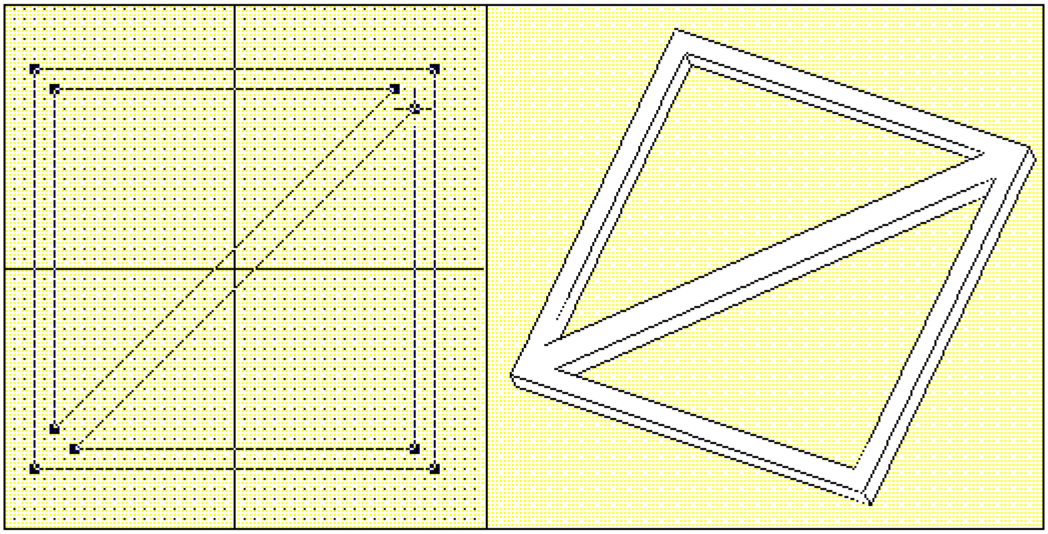
Erase Points

You can use the **<Erase>** function to remove the currently selected point.

Delete Template

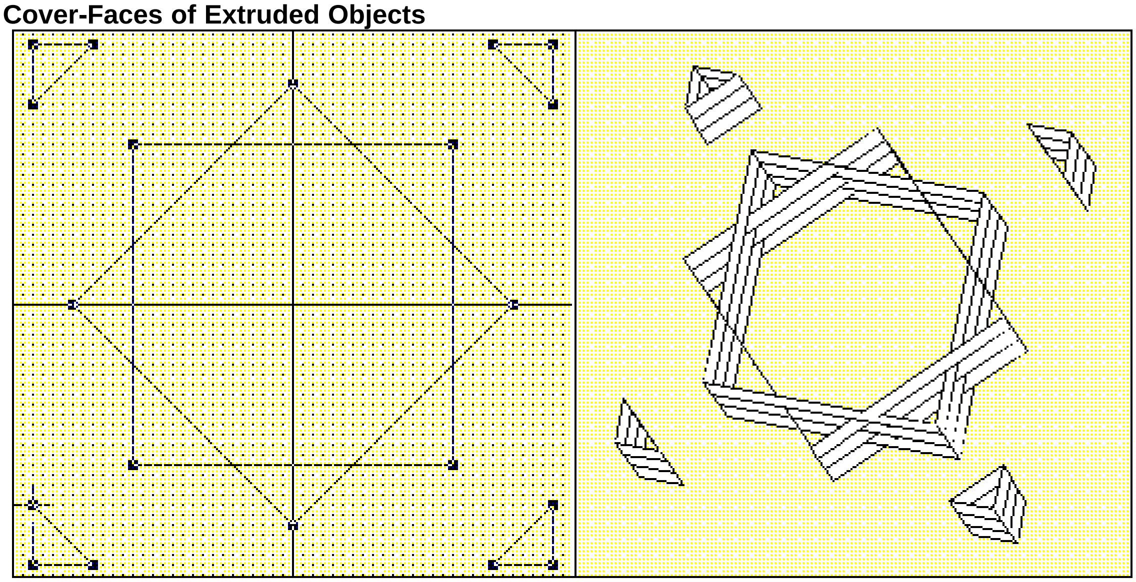
The whole of the current drawing is deleted with the function **<Delete>.**

Cut Out Holes from the Template



If you have already drawn the outline of your object's template, you can insert additional hole templates within the outline. To make clear that you want to begin with an inner hole template, simply press the **<Hole>** button once. You can then begin to draw the hole detail within the template. A started hole template can be completely deleted using the **<Erase>**-function.

Cover-Faces of Extruded Objects

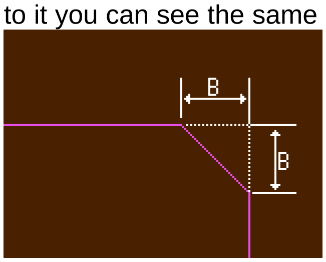


Extruded objects are usually complete with cover faces at the front and rear. You can, however, also generate objects without cover faces by simply switching these options off with the **"Cover-Front"** and **"Cover-Back"** buttons.
As already mentioned, template lines are not permitted to cross if an object is generated with cover faces. However, if no cover faces are to be generated the lines of the drawing can cross as required - as is demonstrated in the foregoing illustration.

Bevel

With covered objects you can choose to generate the cover faces with beveled edges.

The illustration shows an object generated out of an "S"-shaped template and with cover faces. Next to it you can see the same object generated with <Bevel> switched on for the front cover faces.



The distance for the beveling can be defined in the parameter box at the bottom of the tool window. You should choose the value sufficiently small that the template lines do not intersect due to inward directed beveling.

Ribbon Objects

If you activate the <Ribbon> button, the final line closing the rubber band is deleted. You no longer obtain an object with cover faces. Instead you create a line path that is the width of the ribbon.
As long as no hole templates have been generated you can still switch between the two options - ribbon or normal objects - for extruded objects.

Extrude - Segment

The number of segments or layers to the object's depth can be entered with "Segments". With a segment number of zero you create a flat object without depth.

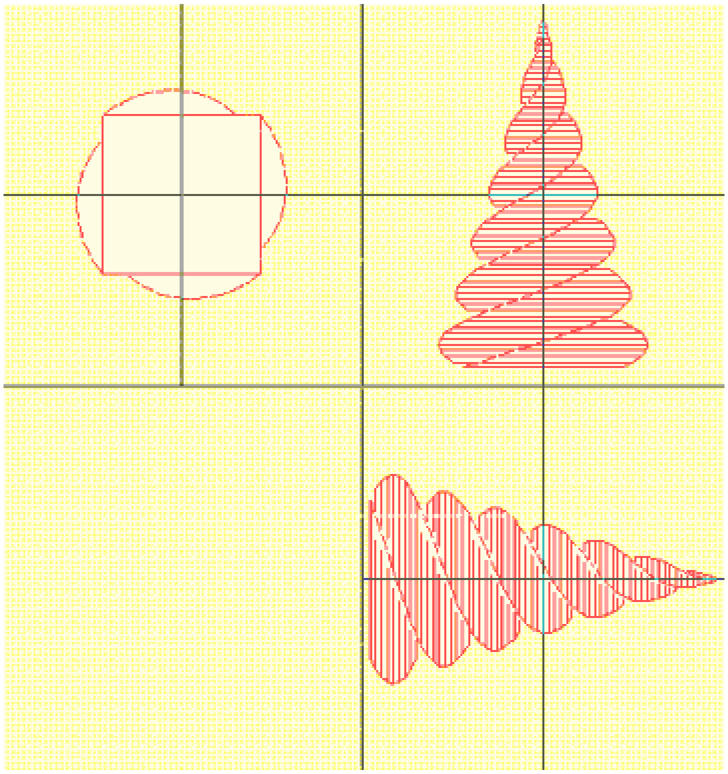
Segment Depth

With "Depth" you can enter the depth of each individual segment. The overall depth of the object, therefore, is segment number * depth. If <Bevel> is switched on you must add one additional segment per side with the defined bevel distance.

Extrude -Taper

The **"**Taper" value allows you to obtain a reduction of the object with increasing depth. If, for example, a square-shaped template measuring 100 is drawn and extruded, the object, without scaling, also produces a square shaped object. The same template, when extruded with a scaling-value of 0.01, results in a tapered, pyramid-shaped object.

Extrude -Twist Template

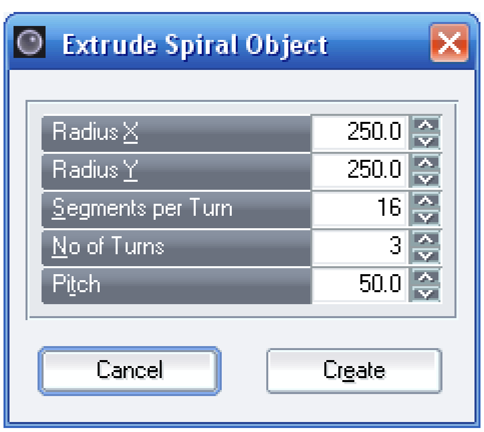


With the "Twist" parameter you can enter an angle between 0 and 90 degrees, through which the template is then distorted segment by segment. For example, with 10 segments and a twist-value of 9 degrees, the object is twisted around itself by 90 degrees. In the illustration you see an example of an object that has been extruded with a twist value of 5 degrees and a scaling of 0.01, simultaneously. A simple square template with only 4 points served as a starting template.

Create Extruded Object

To create an object out of the template you have drawn, simply operate the "Extrude Object" <Create> button. A dialog box then appears in which you can give a name to the object.

8.3.1 Extrude - Spiral Object

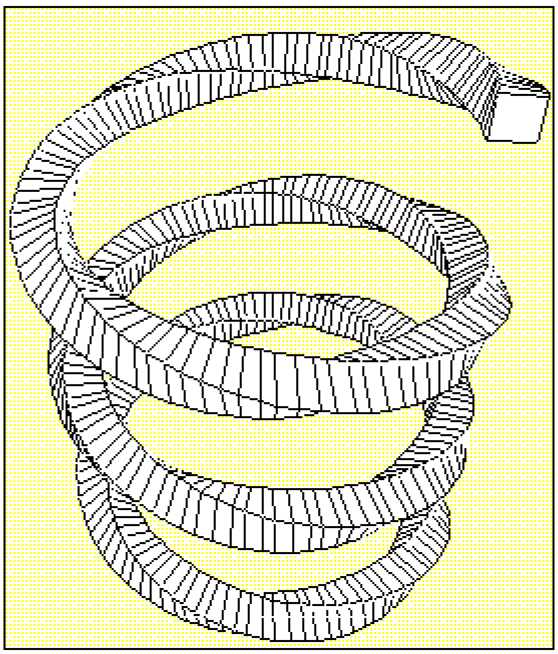


Spiral forms can be generated by drawing a template in the extrude editor 149 and then by choosing the "Extrude Spiral Object" entry in the menu bar.
The dialog box that appears enables you to generate a spiral object whose cross section has the form of the template you have drawn.

The **"Radius-"** parameters input the distance of the midpoint of the template from the center of the spiral.
Different **X** and **Y** values will produce an elliptically formed spiral.
The "**Segments per Turn** " determines how many segments make up a single turn of the spiral.
The next parameter defines the **Number of Turns** .
The **"Pitch"** sets the distance between the center points of each turn. If the pitch is less than the object-height this causes overlapping of individual turns. Experimenting with the various parameters can lead to useful effects. For example, a screw-like form can be easily generated by drawing an isosceles triangle with the apex to the right side and choosing the pitch so that individual turns touch or overlap.

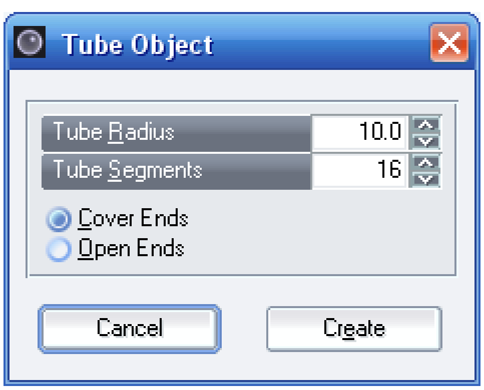
The **"Create"** button causes the object to be generated. With **"Cancel"** you leave the dialog box.

Scaling and Distorting Spirals

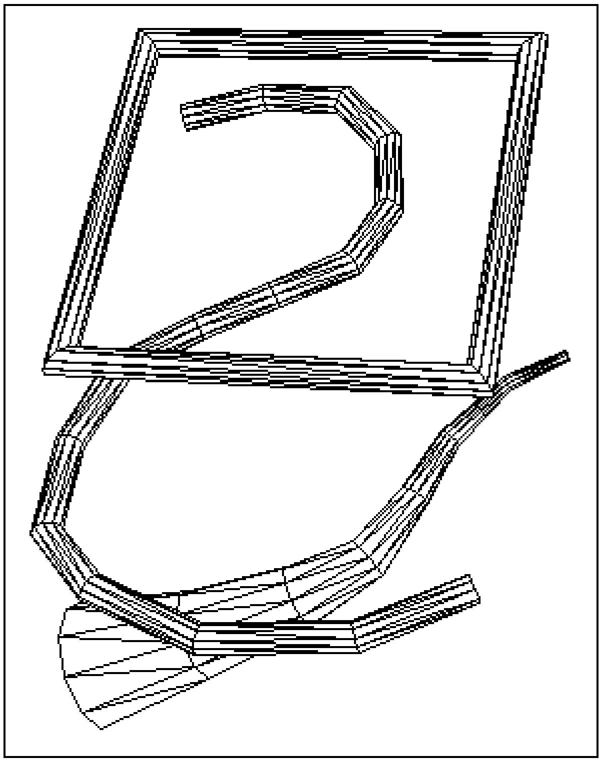


The normal extruded object and the extruded spirals can be scaled and twisted. The relevant parameters in the extrude toolbox 149 must, however, be planned before the call to the spiral dialog. The illustration shows a twisted spiral. The initial template was a simple square pattern of only 4 points

8.3.2 Extrude - Tube Object

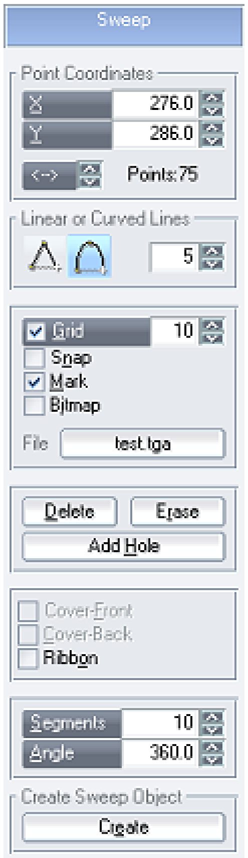


The extrude editor 149 also enables tubular objects to be created. Simply draw any line-path (lines are also allowed to cross) and then select the "Tube Object" entry in the menu bar. In the "Tube Object" editor you can then input the radius and the number of segments of the tube walls.
If you have drawn a closed path for the line then a closed tubular object is created along the line path when you operate the Object- <Create> button. If you have selected the <Ribbon> button while drawing the line path, the final connecting line is deleted. If you then call on the tube editor, a tubular object is created in which the ends are not connected together. In this event, you can decide if you require cover faces for the tube ends by selecting <Cover Ends> or <Open Ends> in the tube editor.



In the illustration you can see examples of tubular objects with interconnected and open end-segments. In addition you can see a tubular object with a taper, which is created by defining a taper factor in extrude toolbox.

8.4 Sweep Editor



 - Menu "Objects - Sweep

The contents of the toolbox and the menu bar in the sweep editor differ in only slightly from those in the extrude editor 149.

The functions in the new menu bar and button bar:

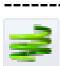
File

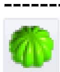
Save Template
Load Template

Quit

Options

 Circular-, Ellipsoid- or Torus Template 160

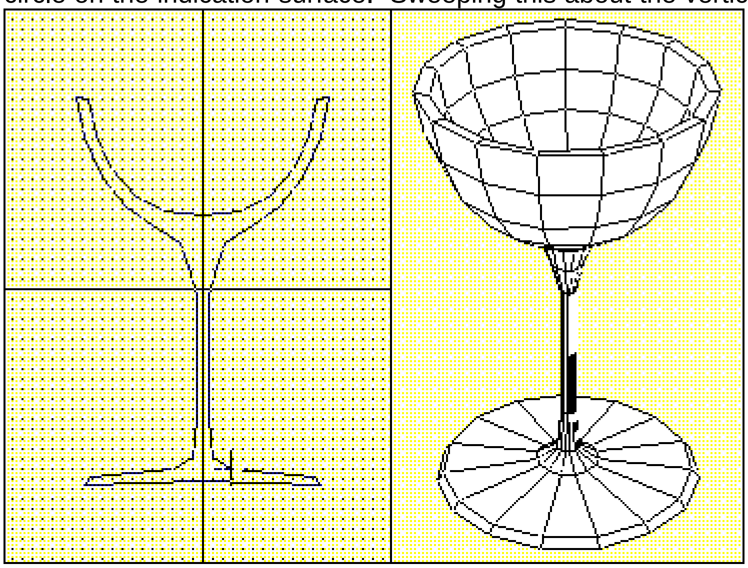
 Spiral object 162

 Waves 161

Back to Main menu

Drawing theTemplate for Sweep Objects

This time the figure is constructed around a vertical line - which represents the sweep axis - and is drawn only on the right side of the indication surface. Each point that is set appears mirrored in the left half of the surface - giving a clearer idea of the emerging object.
Sweeping the points about the vertical axis in a number of steps creates the sweep object. If, for example, you draw a semicircle on the sweep axis this, through reflection, produces a picture of a circle on the indication-surface. Sweeping this about the vertical axis would create a sphere.



An example of a glass-shaped sweep template and the object generated from it can be seen in the illustration.

The Sweep Toolbox

The content of the toolbox corresponds more or less to the content of the extrude editor 149 toolbox. The "Point Coordinates" box is again found at the immediate top of the toolbox, and beneath it the options for linear or curved line segments, the grid-, snap-, marker- and bitmap functions. <Erase> removes the currently selected point and <Delete> deletes the entire template.

Cut Out Holes in the Template

In the same way as in the extrude editor you can insert additional hole-templates within the outline. To make clear that you want to begin with an inner hole-template, simply press the **<Hole>** button once. You can then begin to draw the details of the hole within the template. A started hole template can be completely deleted using the **<Erase>**-function.
Note: Holes in the generated object are only visible later if the object is swept through less than 360 degrees (a segment object).

Cover Faces of Sweep Objects

Standard sweep objects are created without end cover faces (cover faces are not visible anyway with full rotation of the template through 360 degrees). However, for segment objects generated with an angle of less than 360 degrees you can include the respective cover faces by selecting the **<Front cover>** and **<Back cover>** buttons.

Ribbon Objects

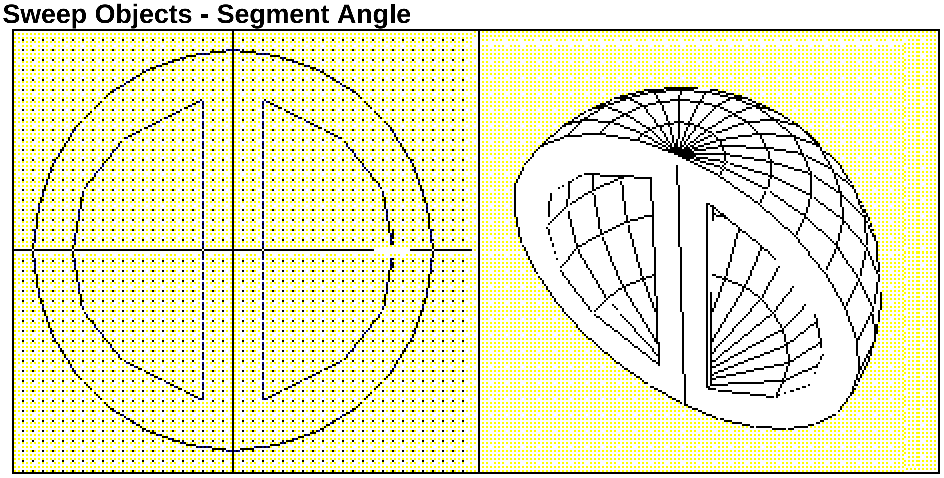
If you activate the **<Ribbon>** button, the last line completing the rubber band is deleted. You no

longer obtain a closed sweep object - instead a band is swept about the central axis.
If no hole templates have been drawn, you can still switch between the band object option and the normal sweep object at any time.

Rotate Segment

The number of steps in which an object is turned about the sweep axis is entered via the **"Segments"** parameter. With a complete sweep about the axis and a segment number set at 3, a rectangular template generates a triangular prism.
If you generate an object to the same template with a segment-number of 12, it forms a reasonable approximation of a cylinder.

Sweep Objects - Segment Angle

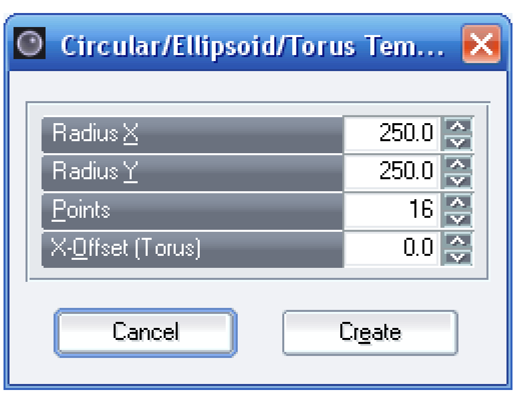


If you enter **"Angle"** parameters at a value less than 360 degrees the rotation is restricted to the stated angle and a segment object is produced. The illustration shows an example of a circle template with a hole and the resulting object generated with cover-faces and a rotation-angle of 180 degrees.

Create Sweep Object

To create an object out of the template you have drawn, simply operate the **"Sweep Object" <Create>** button. A dialog box then appears in which you can give a name to the object.

8.4.1 Circular-, Ellipsoid- or Torus Templates



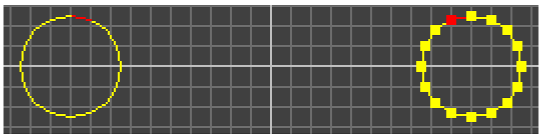
If you select the **"Circular-, Ellipsoid,- or Torus Template"** entry in the sweep editor's 158 menu bar, a dialog box appears that automatically generates templates for the sweep editor.

Generating Ellipsoid and Sphere Representations

If you want to create a sphere or an ellipsoid and the <Create> button is operated, an object is not immediately generated - only the template. This gives the facility to modify the template to generate almost any sphere or ellipsoid segment or disk segments simply by erasing some points or by inputting an angle less than 360 degrees.
All that you have to edit is the **radius** of the **X** and **Y-**axis and the number of points that will be used for the template drawing.

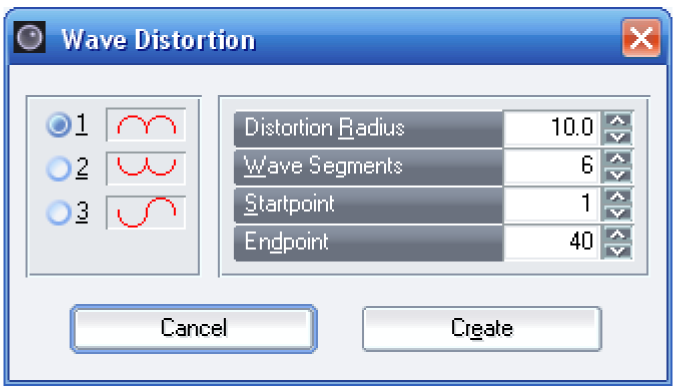
Generating a Torus Template

If you enter an additional X-Offset for the template, the circular shape will be shifted along the x-axis, thus resulting in an torus object when swept around the vertical rotation axis in the sweep editor.



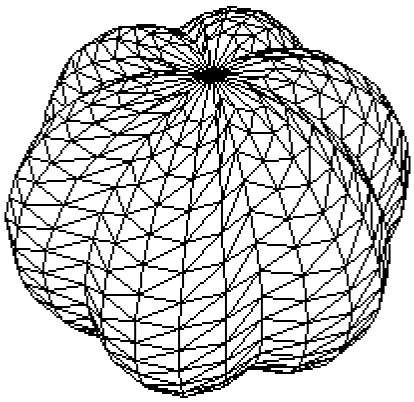
Example: torus template with radius x = y = 25, 16 points, X-Offset = 100

8.4.2 Sweep Objects with Wavy Surfaces



When you choose the **"Waves"** entry in the menubar of the sweep editor 158 a dialog box appears with which sweep objects with undulating surface can be generated. Call this function after a template has been drawn in the sweep editor. With the help of the wave function, an object produced by the sweep-movement is overlaid with sinusoidal oscillations giving the object is given an undulating surface.
However, a large number of points and segments are required to achieve this effect. The overlay follows the horizontal plane only.

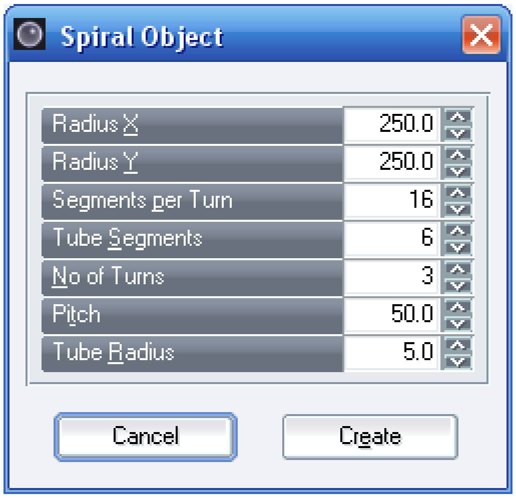
Wave Parameters:



"Distortion Radius" is the operating radius of the overlying sine wave.
The number of segments per oscillation is input with **"Wave Segments"**.
To generate a symmetrical surface, the number of the rotation steps (segments) must be divisible by the number of segments per oscillation. If you want to generate an undulating sphere out of 30 segments and enter a segment-number of 5, then 30/5 = 6 surface waves are produced. Such an object is shown in the illustration.
The wave formation need not cover the entire object:
The **"Startpoint"** determines the starting point in the template from which the overlay should begin.
Similarly, **"Endpoint"** marks the point in the template up to which the wave formation is overlaid.
In this way you can generate, for example, a glass object with an undulating stem and smooth cup and base. You need only specify as starting-point the point at which the stem begins and as endpoint the final point on the stem before the cup starts in your template drawing.
The type of oscillation superimposed, which can be selected with the buttons <1> - <3> in the "Distortion" box, are:

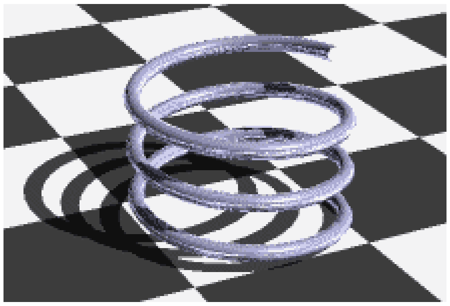
1. Only the upper oscillation of the sine wave is used.
2. Only the lower oscillation is used, so that surface waves curve inwards.
3. The complete sine wave is overlaid.

8.4.3 Spiral Object



Select the "Spiral Object" entry in the sweep editor's 158 menu bar to open a dialog that specifies a spiral with rounded walls in either circular or elliptical form.

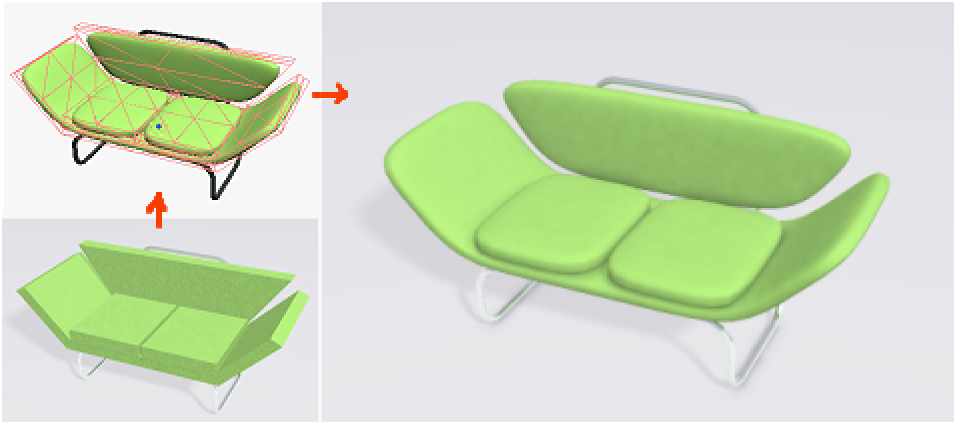
The **X** and **Y** parameters again relate to the radii of the spiral.
The **"Segments per turn"** is responsible this time for the number of the subdivisions per turn.
With **"Tube Segments"** you determine out of how many rotation steps the tube cross section is generated from. E.g. a value here of 3 results in a triangle and a value of 8 approximates a curved wall.
"Number of Turns" is obvious, and **"Pitch"** again determines the spacing of the centers between individual turns.
"Tube Radius" determines the cross sectional radius of the spiral's tube.
The spiral is immediately calculated on operating the **<Create>** button.



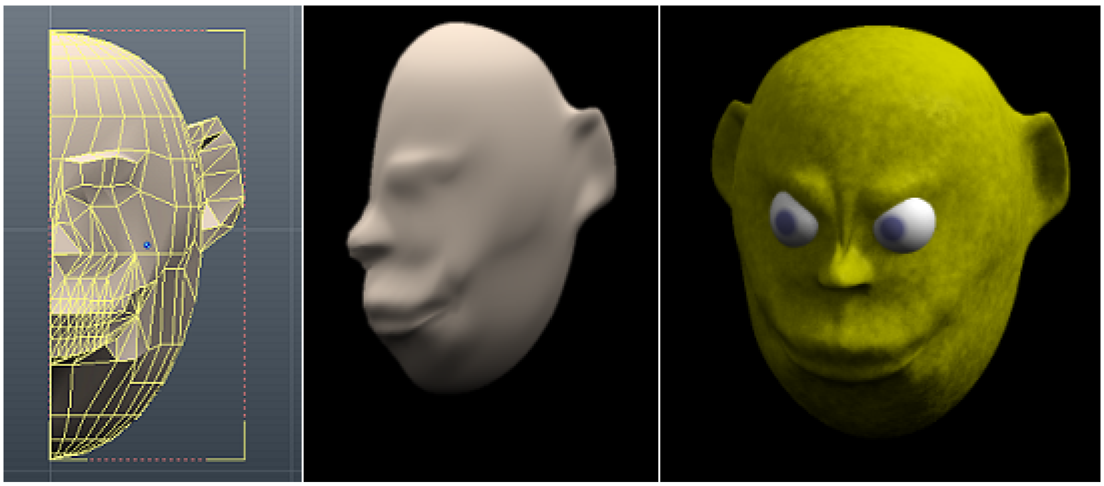
Example: radius x = y = 100, 30 segments per turn, 6 tube segments, no of turns = 3, pitch = 50 and tube radius = 8

8.5 SubdivisionSurface

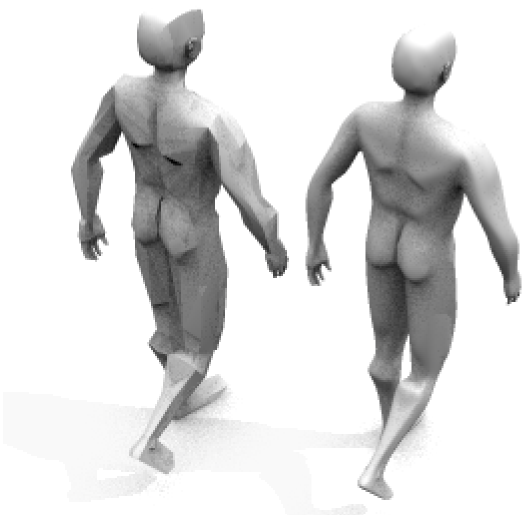
Subdivision Surface (SDS) is a real-time process that transforms a low resolution object to a higher resolution model with organically-curved shape. Every polygon based object can be changed to a SDS model at any time in the modeling process (in MR-3D Designer all objects except NURBS and analytical defined objects can serve as SDS objects). To change an object to an SDS you just activate the corresponding object property in the object properties 289 dialog together with the required resolution (separate resolutions for viewport and final rendering are available). Working with SDS is very easy. The SDS surface is calculated after each work step in realtime from the low resolution object hull. You can't work on the high resolution SDS directly, you simply edit the shape of the low polygon model as you did before, while the SDS is constantly updated.



The picture above shows the effectiveness of SDS-deformation. From a few simple blocks an organically- curved designer sofa is produced just by switching on the SDS-object property for the box objects. (see tutorial - SDS - Design Sofa 105)

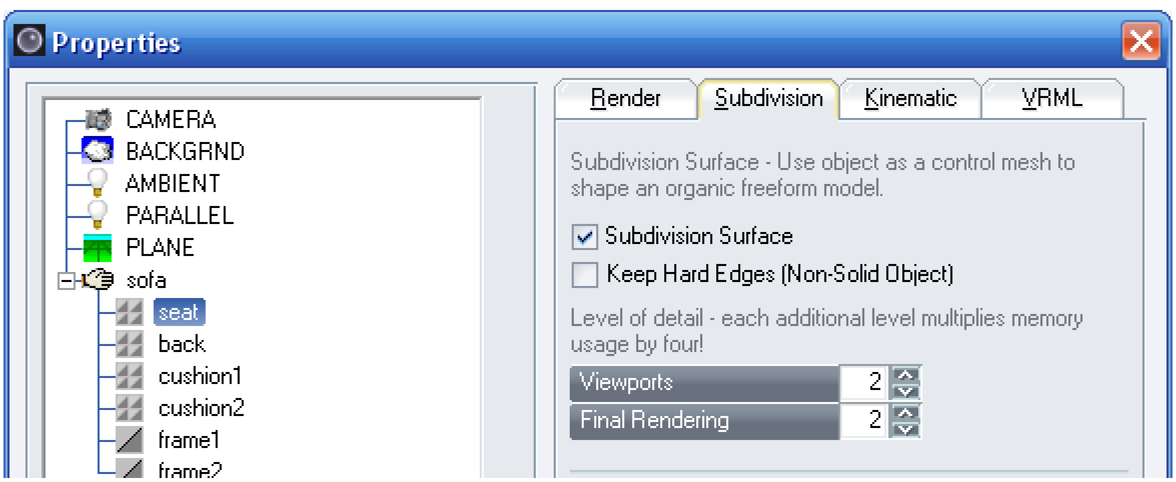


Another example: With the help of some facet extruding operations and by displacing some points and facets a rough face structure was shaped from a half ellipsoid primitive object. Then, after switching on the SDS-object property the roughly-formed features change into a smoothly curved surface imbedded within the low resolution object hull, which is now drawn in wireframe mode around the Subdivision Surface. After that you can continue to work on the wireframed object hull, you can select points or facets to move them around or to extrude new facets from the original shape - the SDS-deformation will be applied in real time to produce an updated Subdivision Surface after each work step. On slow computers or when working with complex models you can choose a lower SDS resolution for the viewport depiction than for the final rendering. The finished half of the face was then completed with a mirrored copy of the half and finally melted into one face (see tutorial - Modeling Faces with SDS 108)



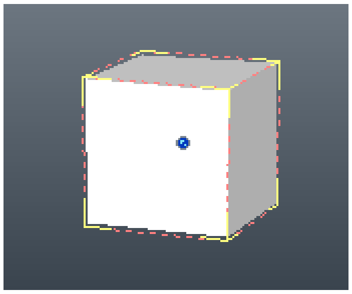
With SDS it becomes very easy to model organically-shaped objects. It also simplifies the animation of characters. With a basic low resolution model you can allocate weighted skin points to a skeleton much faster. You can test movements even without switching on the SDS-feature. Then activate the SDS-property to change the coarse model into a smooth and organically shaped figure for the final rendering of the animation. The picture above shows the demo file "man_walk.cmo" on the left without and on the right with activated SDS-deformation.

Switching On the SDS - Object Property

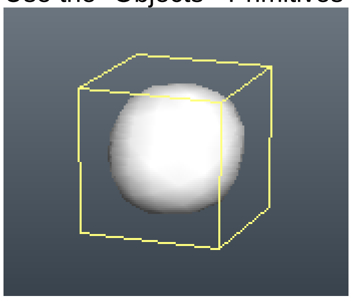


SDS is an object property. Consequently all corresponding settings can be adjusted on the <Subdivision> page of the object properties 289 dialog. Here you can switch on or off again the <Subdivision Surface> object property. Additionally, you can decide here if borders 163 of non-solid objects are to be excluded from the subdivision process and you can enter different SDS-resolution 163 levels for viewport depiction and the final rendering.

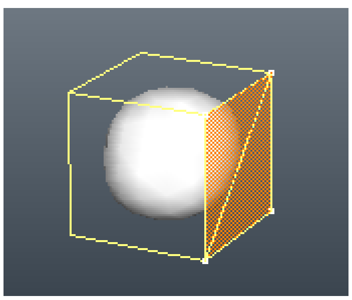
Working with Subdivision Surfaces



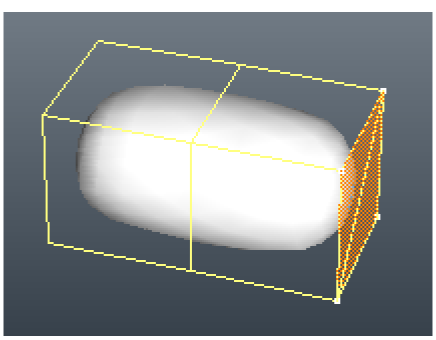
Use the "Objects - Primitives" menu to create a simple box.



Call up the object properties dialog and switch on the <Subdivision Surface> button for the box. If you now exit the dialog the scene is presented as shown in the picture above. The original box formed of 12 triangular facets was subdivided into 48 smaller triangles and the shape of the box was rounded to a sphere-like appearance. With each level of subdivision the amount of new facets needed for the subdivision process is multiplied by 4. On level 1 a triangular facet is divided into 4 smaller triangles. With the maximum subdivision level of 5 for each facet of the original model another 4x4x4x4x4 = **1024** facets are consumed for the subdivision process. Keep this in mind when working with complex models. If you need resolutions so high, only apply them for the final rendering - select a lower resolution for the viewport windows.



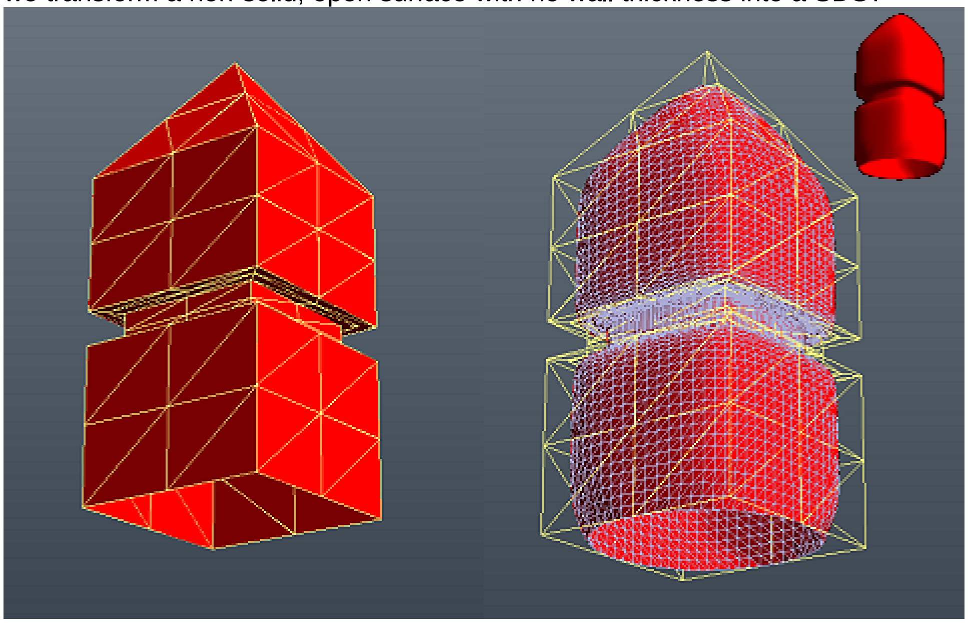
Now change to the "Edit Objects 227" menu. In facet selection work mode, mark two facets at the side of the box and pull out a new segment with a Facet Extrude 228 operation. When pulling out the new segment with your mouse you can also observe how the Subdivision Surface embedded in the wireframe box is constantly recalculated to follow all mouse movements to adapt to the new shape of the hull.



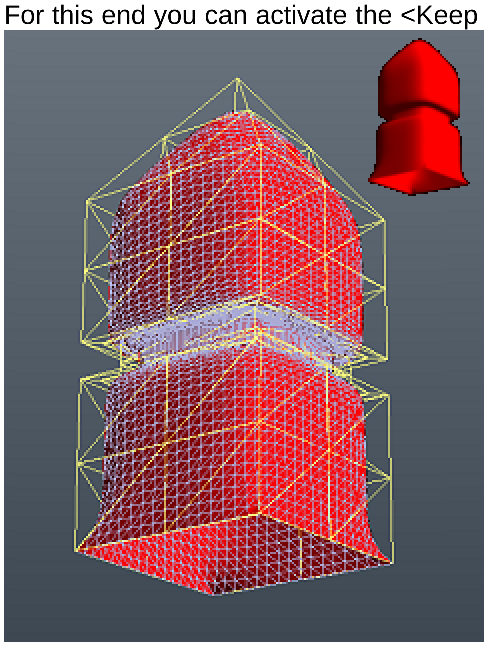
The result of the facet extrusion process - a two segmented box with a capsule-shaped Subdivision Surface.

Keep Hard Edges (Non-Solid Objects)

The subdivision algorithm works best with evenly structured and closed objects. But what happens if we transform a non-solid, open surface with no wall thickness into a SDS?

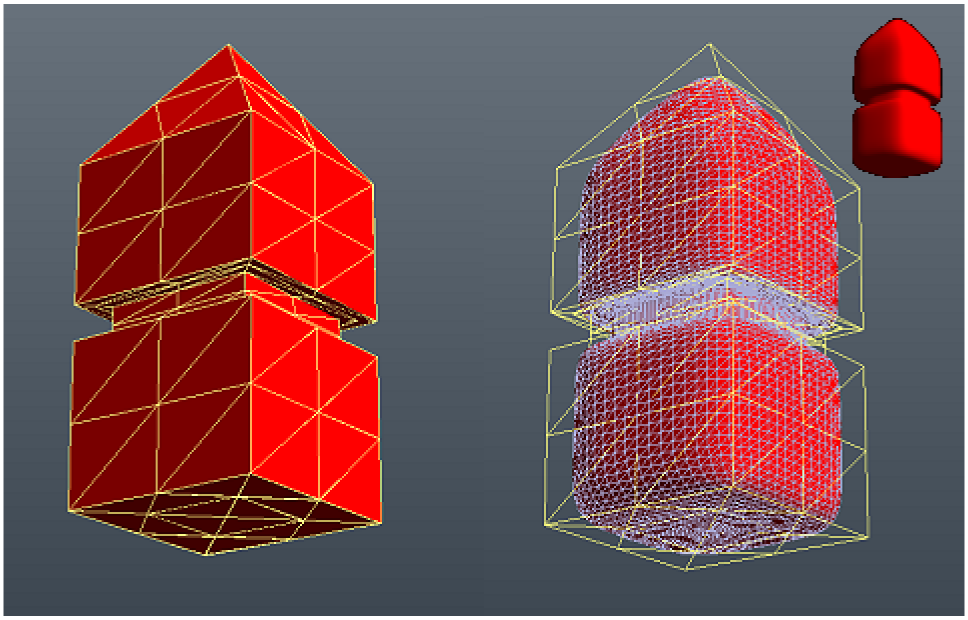


The picture above shows an object which is open at the bottom. After switching on the SDS object property we again get a smoothly subdivided version of the original object within the object hull. The previously rectangular-shaped edge at the bottom was included in the subdivision process - resulting in a curved shape now at the bottom. But this is not always desired. In some situations you may want to exclude the borders of an open surface and just refine the structures within the borders. For this end you can activate the <Keep Hard Edges> function.

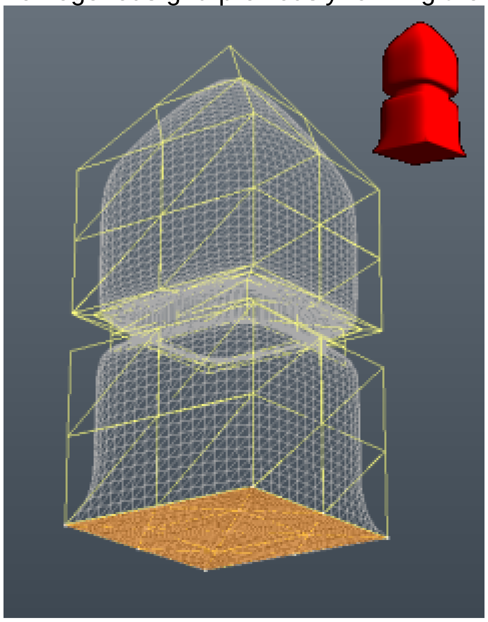


The picture above shows the result after switching on the <Keep Hard Edges> button. The rectangular shape of the bottom is retained and a smooth transition calculated towards the smoothly-curved upper parts of the object.

Hard Edges in Solid Objects



It is also possible to create hard edges in a solid object. In the picture above you see again our model from the previous demonstration, but this time the bottom faces are included to provide a closed solid object. When switching on the SDS-property we get the expected result, a rounded object with a smoothly curved base. What can we do now if we want to exclude the bottom faces from the subdivision process to keep the flat and quadratic shape of the basis? We need to use a little trick to achieve the desired result. First, all bottom faces are selected in face selection work mode. Then, in the Edit Objects 227 menu we operate the <Detach Face Selection> button. This will insert new points at the borders of the face selection to separate the bottom mesh from the homogenous grid previously forming the whole model.



At first sight the object hull hasn't changed. However, the object now consists of two separated meshes for the bottom and the upper part of the model. (Select the bottom faces and move them away to test it). Now that we have two open meshes (still managed within one object) we can apply the <Keep Hard Edges> function again to exclude the borders of both meshes from the subdivision process. The result is shown in the picture above. The base as well as the lower edges of the side

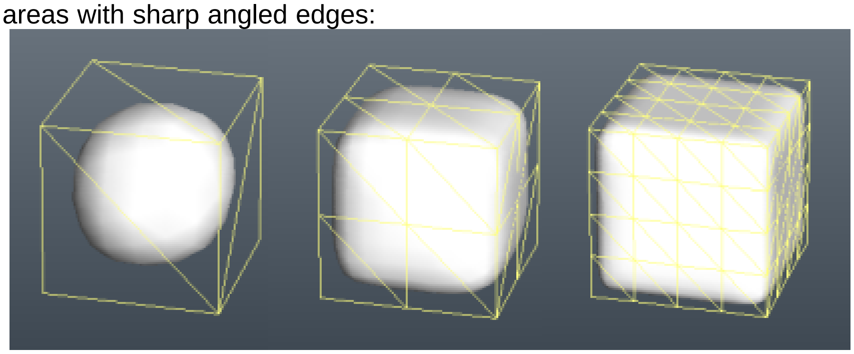
walls keep their rectangular appearance.
If, for some reason, you want to restore the two separated meshes to a single net you can try the "Delete" <duplicate Points> function of the Edit Objects menu. Provided that both meshes were not moved away from each other the net will be melted again in one single homogenous net.

Level of Resolution

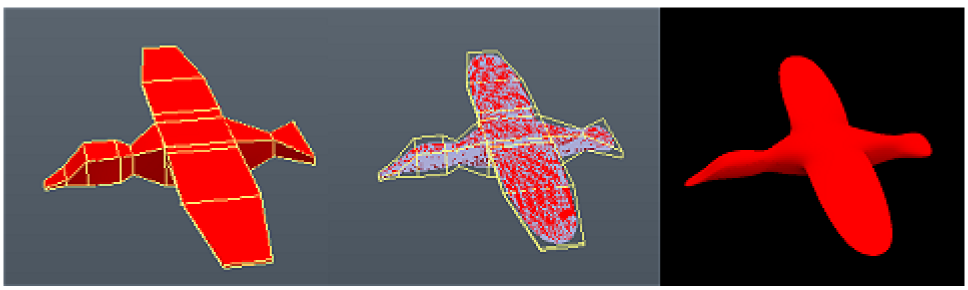
With each level of subdivision the number of new facets needed for the subdivision process is multiplied by 4. On level 1 a triangular facet is divided into 4 smaller triangles. With the maximum subdivision level of 5 for each facet of the original model another 4x4x4x4x4 = **1024** facets are used for the subdivision process. If, for example, you have modelled a face that already consisting of 1000 facets and you change it into a subdivision surface with a maximum resolution level of 5 this would result in a face containing more than one million facets! Keep this in mind when working with complex models. If you need resolutions so high, then only apply them for the final rendering and select a lower resolution for the viewport windows.

SDS - Quality

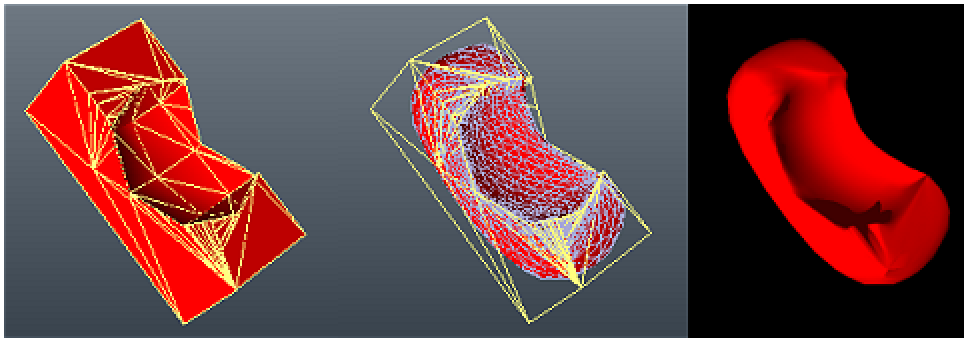
The more evenly structured the object hull the better the quality of the SDS-deformation. By refining the surface using triangulation you can also limit the influence of the subdivision process to those areas with sharp angled edges:



A cube with sidewalls consisting of only 2 triangular faces is completely rounded in the subdivision process (image above on the left). The pictures in the middle and on the right show the same cube after applying the <Triangulate Selection 237 > function once and twice, respectively. With increasing facet resolution of the original model the subdivision process increasingly resembles a bevelling function, but you pay for this with a high memory consumption and even longer rendering times.



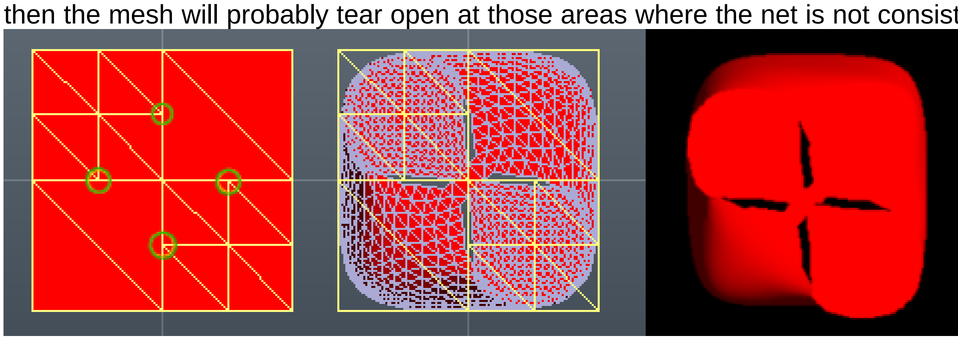
This model of a bird created by some facet extrude operations from a simple box primitive is ideally suited for the subdivision process because of the clean and uniform structure of similar-sized faces.



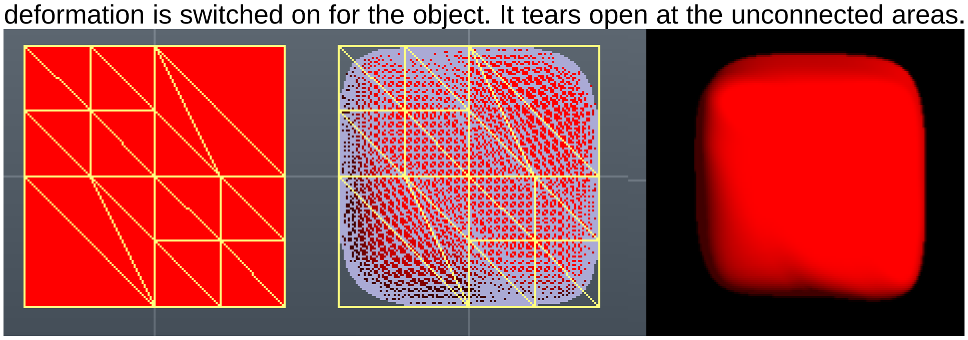
This is an example of a model containing large and pointed facets mixed with smaller facets crossing the surface in all directions. This often leads to undesired creases in the subdivision process as shown in the picture above on the right.

The Subdivision Surface Tears Open - What To Do?

SDS-deformation can only be applied properly if the object consists of a single homogenous mesh, i.e., all triangular facets have to be connected with a common edge (sharing the same edge points) with all of their adjacent facets. This cannot be taken for granted. When importing foreign files or models created with previous versions of MR-3D Designer (up to v.11.x) it can happen that a corner of a triangle is not connected to all adjacent facets. If you apply SDS-deformation to such a model then the mesh will probably tear open at those areas where the net is not consistent.

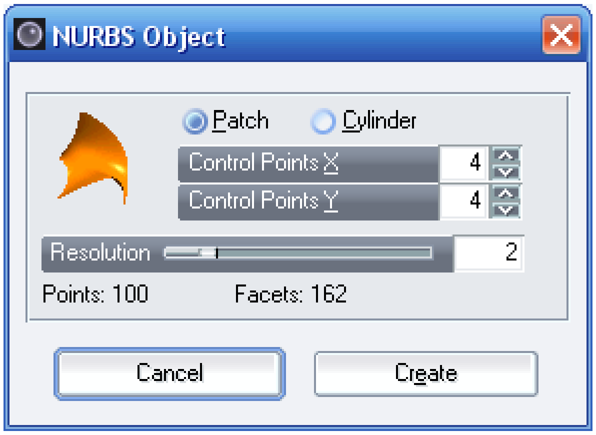


The picture above on the left illustrates the case. The problem points are marked with a green circle. These are vertices of smaller facets which are not connected with a common edge with their adjacent larger facets. The pictures in the middle and on the right show what happens if SDS-deformation is switched on for the object. It tears open at the unconnected areas.



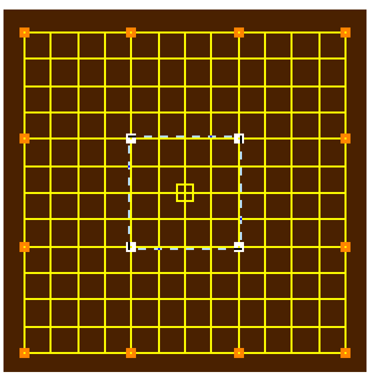
You can try to repair such a mesh with help of the corresponding <Repair Mesh Structure 240 > function found in the Edit Objects menu. The picture above shows the result after applying this function to our box mesh. At those locations where vertices of a smaller facet came to lie on an edge of a larger facet these points were used as splitting points to cut the larger facets into two halves. In cases where this function doesn't work may duplicate or points lying very close to each other have to be removed first. Try the combination "Delete" <duplicate Points> followed by <Repair Mesh Structure>. Repeat the process if necessary with a higher tolerance value for the removal of duplicate points.

8.6 NURBS - B-Spline Patches and Cylinder

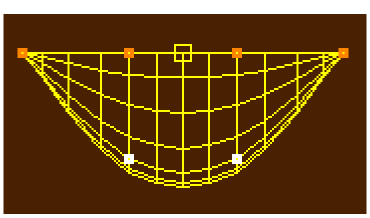


The "NURBS Object" dialog is reached by choosing the "NURBS..." entry under "Objects" in the menu bar or  in the button strip.

NURBS stands for "Non-Uniform Rational B-Spline", a special type of deformable 3D patch. A surface is created based on a rectangular grid, similar to that of the 3D function generator. But this time the individual points of the grid represent control points that form a surface of much higher resolution. By manipulating these control points you can very easily model smooth and organic shapes.



Above you can see a NURBS patch defined by 4 by 4 control points (painted in orange) but the actual resolution of the grid is much higher. You can adjust the initial resolution for a NURBS object through the resolution slider. The points lying between the control points are interpolated using B-Spline-Interpolation and are recalculated each time you move a control point. Control points can be selected in  point selection mode and then be manipulated like any ordinary point in common facet based objects. Thus you can select, move, scale, rotate or deform control points as required to shape the form of the patch. The picture shows a patch with 4 center control points selected.

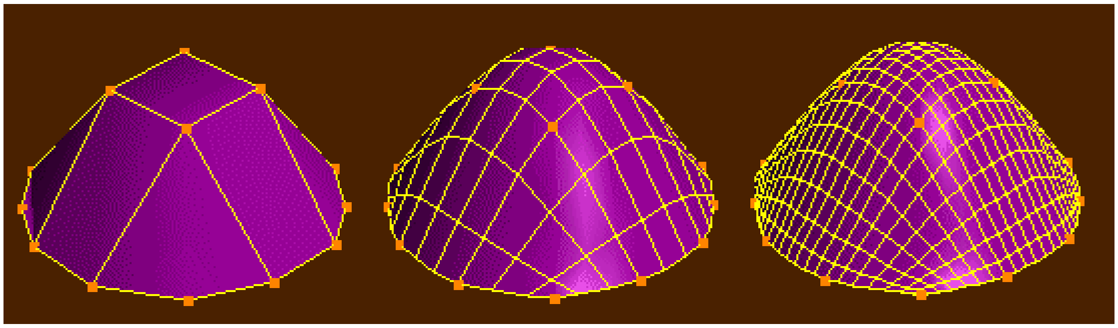


In top view these 4 points were moved to the front. You can clearly make out how the individual control points are interpolated through a curved surface. In MR-3D Designer control points are an integral part of the surface and therefore influence directly the shape of the grid. Therefore it is not necessary, as you may have known it from other programs, to adjust additional weight parameters for each control point.

The Dialog Parameters:

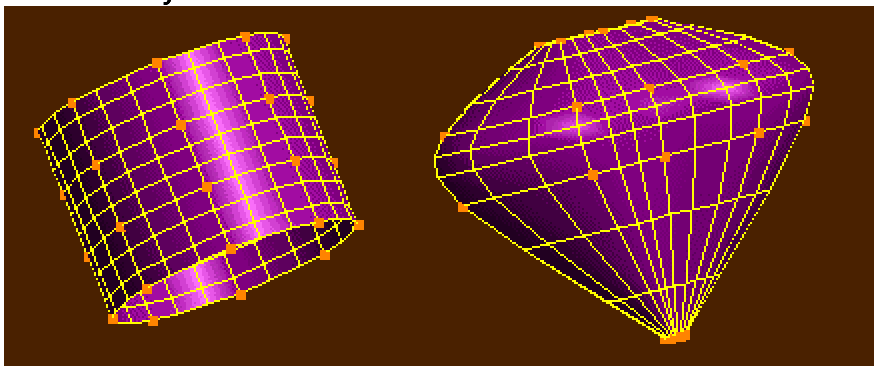
Control Points
You can define the initial number of control points in x- and y-direction for the NURBS patch. Later you will see how to add new rows of control points to existing NURBS objects.

Resolution



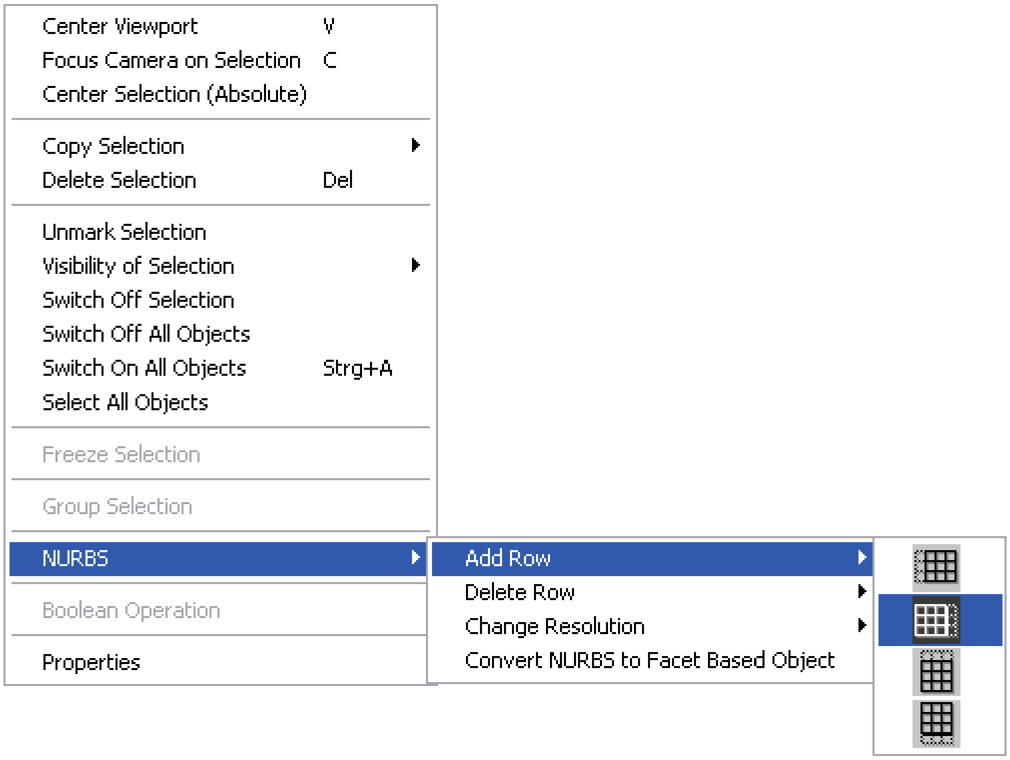
The resolution parameter defines the initial point resolution of the surface. This can be changed at any time later in the working process. With NURBS you can model quite complex objects, e.g. faces. The representation of high-resolution surfaces can considerably slow down representation on the screen therefore a lower resolution is usually choosen for the modeling process and then, at final rendering stage, a higher resolution for high quality output.
In the illustration on the left side you can see a surface with a resolution of 0 (only control points, no additional points are generated for the surface). Next to it the same object with progressively higher levels of resolution.

Surface or Cylinder



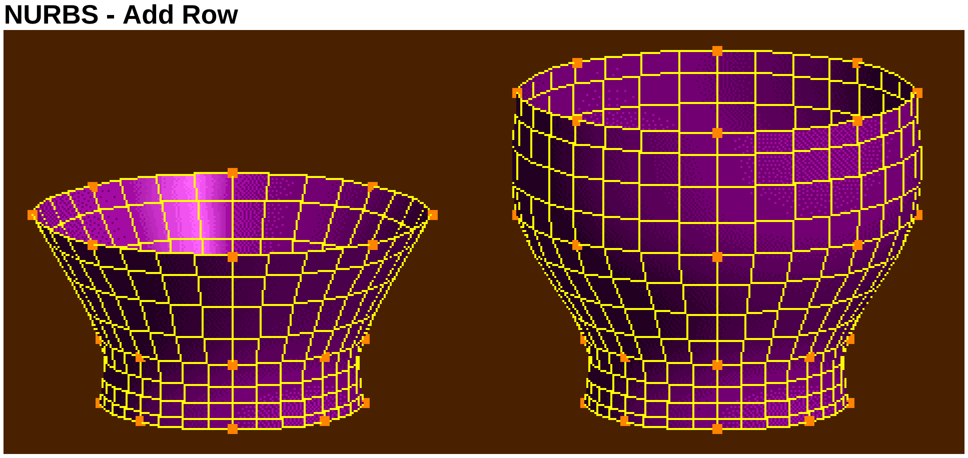
In addition to the initial flat NURBS patches you can choose to generate a cylindrical NURBS object. In that case, the "control points x:" parameter defines the number of control points forming the circular shape. The illustration above shows an example of a cylindrical NURBS object and next to it, with a few scaling and moving operations, the heart-shaped object that was formed from it.

Functions to Modify NURBS Objects



NURBS objects can be subsequently modified, e.g. by adding or deleting extra rows of control points or by changing the resolution. To modify a NURBS object first select it as a reference object. Then open the popup selection 133 by clicking with the right mouse in a viewport window. Under the NURBS-entry you find the following functions:

NURBS - Add Row



NURBS patches can be extended by adding extra rows of control points. All 4 sides of a patch can be enlarged this way. For cylindrical NURBS you can add to the top and to the bottom. Above you see an example of a NURBS cylinder after adding one extra row of control points to the top of the cylinder.

NURBS - Delete Row

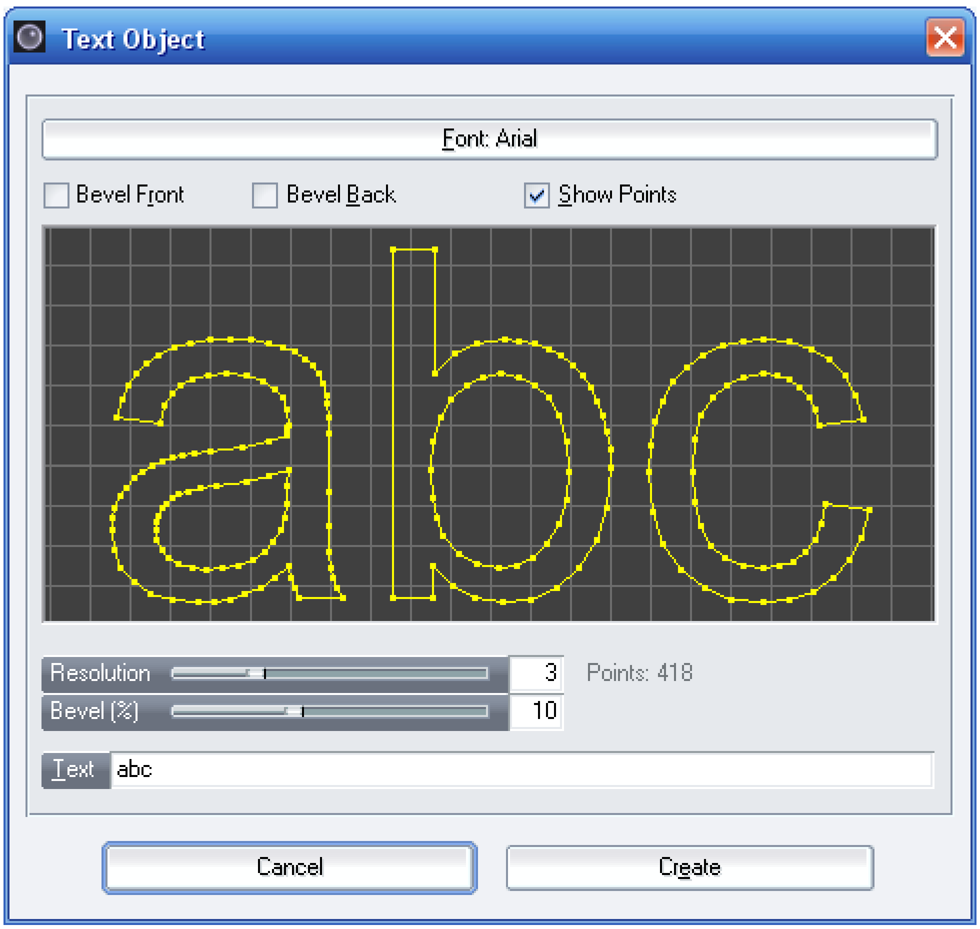
This function deletes superfluous rows of control points.

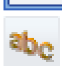
NURBS - Change Resolution

To change the resolution of a NURBS patch or cylinder, choose a value between 0 (only control points are visible) and the maximum resolution of 15.

NURBS - Convert NURBS to Facet-Based Object
Similar to analytical described objects, NURBS are restricted in some ways. The points lying between the control points are recalculated after each manipulation and cannot be selected specifically to work on them. Many functions, e.g. deleting facets and points, Boolean Operations or detaching parts of an object can't be applied to a NURBS object. Converting a NURBS object to a simple facet based object will make all this functions available again. But on the other hand no return to the NURBS object with its special modeling abilities by manipulating control points and resolution will be possible.

8.7 3D Text Objects



 - Menu "Objects - 3D-Text"

In the "Text Object" dialog, any TrueType font can be selected for generating 3D text objects. Also Beveling can be applied. Simply select the font you like and choose the output quality, input the text and create the object with the <Create>-button.


Simple example of a text object, see project folder "..\projects\font.cmo"

Font - Activate the <Font>-button to display the system's font dialog, from which you can select any TrueType font installed on your computer.

Bevel - The text objects can be generated with beveled edges. You can select this option separately for the front and rear of the object. If e.g. the rear of the text will never be shown, you can save a considerable amount of points and facets, if you only switch on the front beveling. The depth of the beveling is controlled by the "Bevel"-Slider beneath the text preview. The bevel value in percent defines the depth of the beveling in relation to the total depth of the created text object.

Resolution - You can restrict the amount of points needed for an object with the resolution slider. In the preview window you can control the output quality of the text object.
Switch on the "Show Points" option to emphasize all points from which the object will be generated. That will make it easier to find a good compromise between good output quality and the smallest possible amount of points and facets.

Text - Input the desired text for your 3D-text object in this edit field.

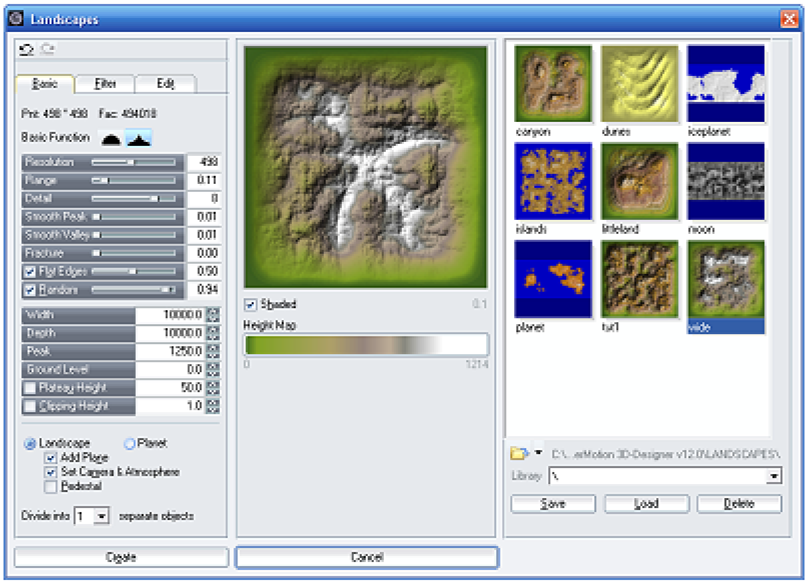
8.8 Landscapes and Planets

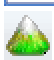
No photo, no bitmap textures, only the pure beauty of mathematics - create your own worlds with MR-3D Designer

- **Landscape Editor** 177
  Introduction and dialog layout
- **Landscape Editor - Basic Parameters** 179
  Define the dimension and structure of your terrain or planet
- **Landscape Editor - Filter** 182
  Add terraces, crater or dunes to the terrain
- **Landscape Editor - Edit Height Map** 184
  Paint your own ridges or the course of a river into the height map

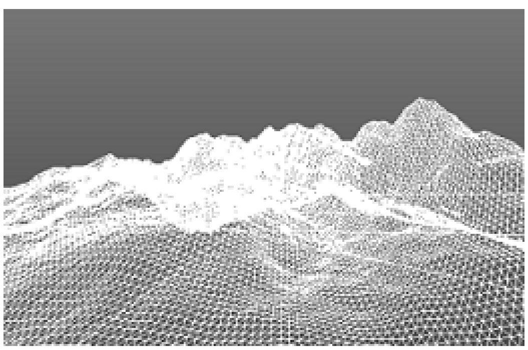
Related topics:
Tutorial - Landscape Design 85
Background Dialog - Atmosphere 378

8.8.1 Landscape Editor



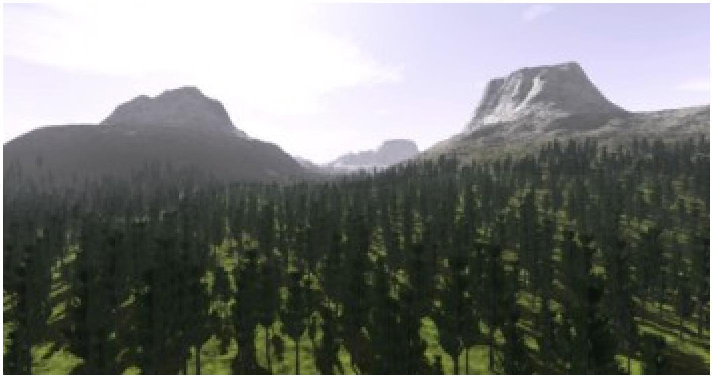
 - Menu "Objects - Landscapes"

If you select the "Landscapes" menu entry a dialog for the production of fractal landscapes appears.



These landscape objects are based on a rectangular grid and height information calculated for the grid coordinates using a fractal algorithm (similar to the Functions editor 187). The picture above illustrates such a landscape grid in camera view...


...and these two pictures show examples of final renderings after special landscape textures 323 and an atmosphere 378 have been applied.



Here a partcle system 425 was used to fill up the hill side of a landscape object with trees.

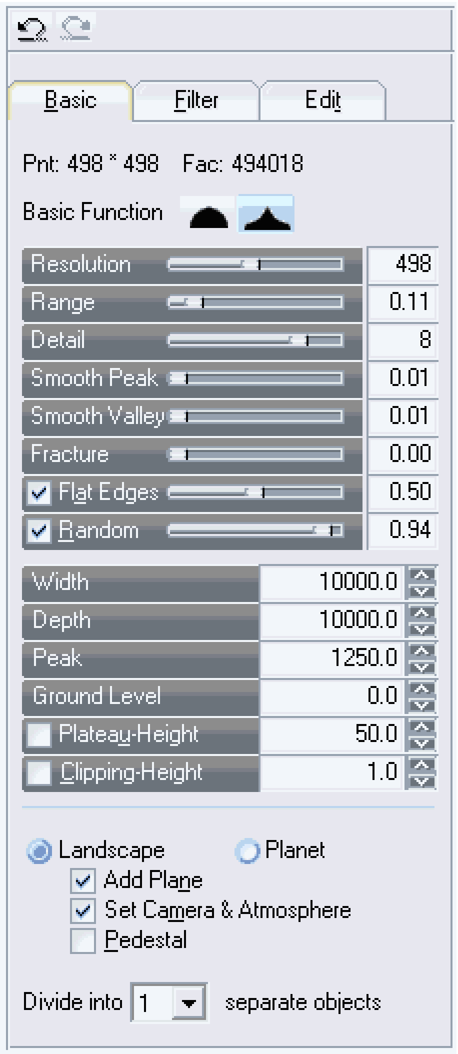
The landscape editor provides a preview window with a shaded plan view and many functions and filters (crater, terrace, etc.) for the basic generation and the editing of the height fields. By means of special painting tools you can directly "draw" in the preview window to elevate or lower the ground or to smooth eroded slopes, for instance.
In planet mode the landscape net is wrapped around a sphere to create highly detailed meteors or planets. The landscape editor can generate nets of up to 2 million facets, but this has to be handled carefully. A minimum of 1024MB RAM should be available when exceeding a million facets, otherwise the constant outsourcing of memory to the hard disk will virtually break down the whole process. However, there is no need at all to generate such large nets because the new landscape objects come with new multi-layered procedural textures to provide the necessary details.
See also our tutorial Landscape design 85 on how to plan a whole outdoor scene with landscape, atmosphere and water plane.

The Landscape Editor

The dialog is divided into 3 areas:

- On the left side you will find the tool area with all settings for the generation and editing of the landscape. It contains three sets of parameters that you can switch between with 3 tabs in the dialog's document header:
  - Basic 179 - The basic parameters define the fractal pattern and the dimension of the object. You can add a clipping plane and decide whether you want to create a landscape or a planet object.
  - Filter 182 - These filters are blended with the basic fractal pattern to add dunes, terraces, crater or crevices.
  - Edit 184 - The Edit tab provides the painting tools to directly influence the height map in the preview window.
- The preview window is located in the middle of the dialog. It shows the height map of the landscape. The height of each individual point in the height map is indicated by a different color corresponding to a color range placed directly beneath the preview window. If the <shaded> button is also switched on, then an additional light source illuminates the height map showing clearly the contours of the mountains. To edit a color range simply click on the color range button. The Color Range dialog 30 will open where you can freely define your own colors or simply load pre-defined entries from a visual library. When you move the mouse over the height map the current height is indicated below the preview window. Incidentally, when creating a landscape the current color range will be selected automatically as a color range texture for the object. See also the chapters about landscape design and procedural textures to learn more about the special capabilities of landscape textures.
- The visual landscape library occupies the area on the right area of the dialog. You can easily load existing files or add new examples to the library. The files do not contain data on actual facets (that would be rather memory consuming), only the parameters and working steps leading to the current height map. *See also:* Visual Libraries 28.

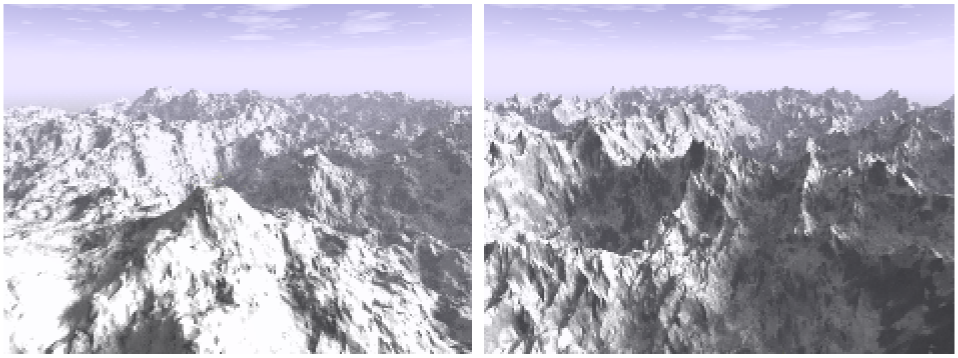
8.8.2 Landscape Editor - Basic Parameters



When you select the <Basic> tab in the landscape editor the basic parameters for the definition of the fractal pattern, the dimensions and the clipping of the landscape object appear. You can also determine here if you want to create a landscape or a planet object. At the top of the dialog are two additional Undo/Redo-buttons, with which you can restore or repeat almost every operation in the course of the landscape creation.

The parameters:

Basic Function - There are two different basic functions. If you choose the left icon the mountainside will be calculated with a more rounded and hilly appearance. Select the right icon if you want to creates rock formations with steep and sharp mountain ridges.



Example of a landscape created with a resolution of 1 million facets. On the left a curved basic function was applied. On the right the same landscape with the basic function switched to the sharp edged appearance.

Resolution - specifies the number of points used to form a row of the fractal grid. For instance, if Width and Depth of the grid object are identical and a resolution of 250 points is input, then the final grid will be made up of 250 * 250 = 62,500 points. If the Width and Depth parameters for the grid are different then for the shorter side of the grid a suitable point division is calculated from the resolution parameter.
Caution: If your computer is not equipped with at least 128 MB of free RAM you should not top the limit of half a million facets for a landscape object, otherwise the constant outsourcing of memory to the hard disk will considerably restrict a fluent working process. With 256 MB and more even 1 million facets and more can be handled rather comfortable.
Generally it is advisable to first create an object with a low point resolution (save the settings to the landscape library) and adjust camera positions, background, material settings and lighting. Then go back to the landscape editor and create the same landscape object with a higher point resolution to replace the low resolution object. Before deleting the old object transfer the material settings from it in the material dialog using the <Get> material function.

Range - the height map used to form the grid is calculated from a fractal algorithm. The range parameter acts in this context like a zooming out effect from this fractal structure. The higher the value, the more extensive is the area, with an increasing number of hills and valleys becoming visible.

Detail - the Detail parameter sets the number of iterations used by the fractal algorithm and with it the detail in the surface generation. Each of the iterations adds new detail the height map - the landscape becoming more jagged and bumpier with every step. With only a minimum value of 1 iteration the algorithm produces a very gently hilly terrain.

Smooth Valley - While the topography in low areas often appears smooth and round, with increasing height the terrain appears more jagged and stony and therefore more detailed. With the Smooth Slope parameter you can enter a percentage of the general height of the object below which the lower parts of the object will be smoothed down.

Smooth Peak - With this parameter you can smooth the higher regions of your terrain. This can be useful to make mountain ridges (basic function two) even sharper.

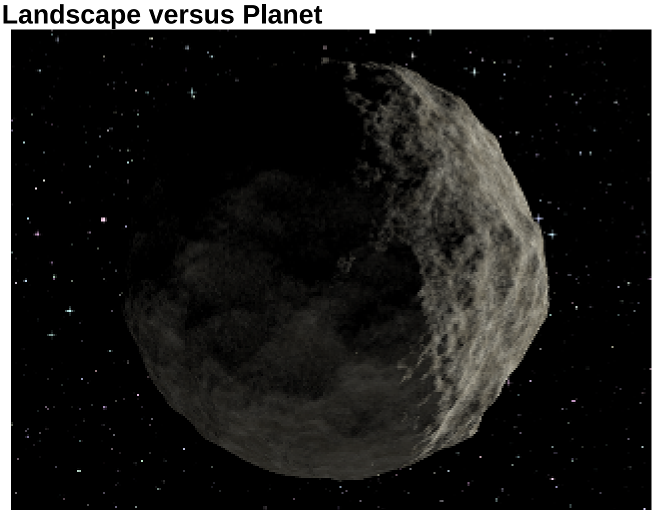
Fracture - A variation of the fractal algorithm results in a somewhat more fractured surface with sharp steps and rifts.

Flat Edges - With this option switched on the edges of a landscape object will smoothly run down to ground level. This way the terrain can be merged seamlessly with a plane object (for instance as an island on a water plane or several smaller terrain objects on a grassy plain). The parameter determines the area over which the blending of the height is calculated.

Random - The Random parameter enters different starting values for the fractal calculation and produces many variations in the production of the object. You can also switch off the random generator - this results in an absolutely flat plane object consisting out of thousands of points and facets. This makes no real sense but you can use this as a basis to study the outcome of parameter changes on the Filter page in the dialog, for instance, to make visible the overlaying of craters, dunes and crevices without being distracted by the fractal pattern. The same goes for the Edit page with the painting tools.

Landscape Dimensions
The **Width, Depth** and **Peak** parameters specify the dimensions of the generated object. Any rectangular area within the world limits of ±16,000 units can be defined. You can also enter a negative value for the **Ground Level.** With an additional plane object and the <Flat Edges> option activated (always aligned to a level of zero) parts of the object will lie below the plane. If this is intended, you should also switch on the **Clipping-Height** function to remove all superfluous points and facets below the plane level. You could also enter higher values for the Clipping-Height to generate a group of islands from the height field.
Plateau-Height - If this option is activated then all height values above this level are cut off to the plateau height. While this option is suitable for creating simple plateaus the terrace functions on the filter page offer much more possibilities for creating terraced surfaces with plateaus on top of the terrain object.

Landscape versus Planet



At the bottom of the dialog you can decide if you want to generate a landscape object or a planet. For the latter case the fractal grid is simply wrapped around a sphere and the edges merged seamlessly together. In principle the word planet is a little bit overstated. If you look at pictures showing the earth seen from outer space than our planet appears almost totally round - you recognize heights and valleys more by their color than by their irregular shape against the background. Taking this into account a simple sphere with a good bitmap texture would be much more appropriate for creating a planet object than a high resolution fractal planet object consisting out of millions of points and facets. On the other hand with the planet option you can create wonderfully irregular shapes for moonlets or asteroids - especially in combination with the crater filter 182.

Landscape - Add Plane
If your project does not yet contain a plane object and the option "Landscape" <Add Plane> is switched on, then automatically a plane object at ground level will be generated with the terrain object.

Set Camera & Atmosphere

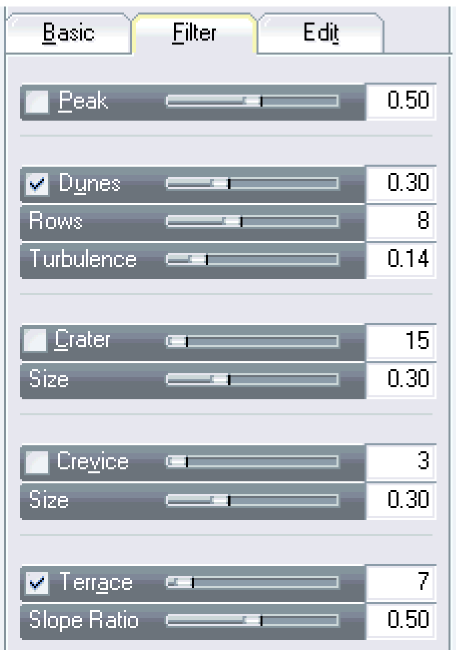
A suitable camera position in front of the generated landscape will assigned automatically and the background switched on if this option is active when creating the landscape.

Landscape - Pedestal
The edges of the landscape object are vertical. The terrain, one might say, stands proud of a base.

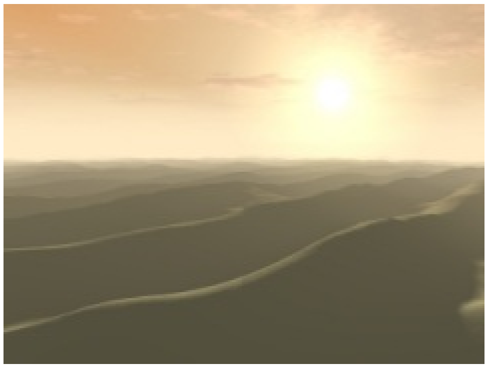
Planet with Water Surface
If the Clipping-Height function is switched on then an analytical blue sphere will automatically be generated with the planet object. This sphere represents the water surface covering the parts of the planetary surface where facets were removed by the clipping function. In the object selection dialog the water sphere will be placed hierarchically subordinated 140 to the parent planet object.

Divide Landscape/Planet into Separate Objects
On generation the object can be split into several separate objects. This way, when working with very large terrains, you can temporarily switch off parts of the landscape while working on a particular area. The separate parts of the object will be hierarchically subordinated 140 to a parent object. The highest part of the landscape will always be the parent object. This makes it easier to select the whole landscape object for working because you just have to select the highest elevation in the terrain to mark the parent and with it all children. It is also important for the later texturing of the terrain. The terrain textures 323 depend (among other things) on the total height of an object so it is the parent object that provides the total height dimension for the texturing. All subordinated objects reference the material of the parent object.

8.8.3 Landscape Editor - Filter

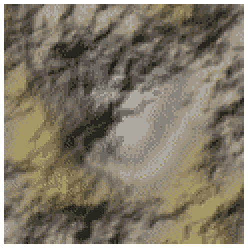


Select the <Filter> tab in the landscape editor to bring the page with the filter settings to the fore. The filter functions overlay the basic fractal landscape pattern to add further details like dunes, terraces, craters or crevices to the terrain.

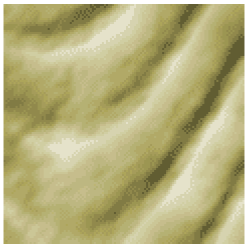


Example of a fractal grid object overlaid with a dunes filter. It consists just of 250*250 points (about 125,000 facets) and was rendered with a simple sand texture and an atmosphere.

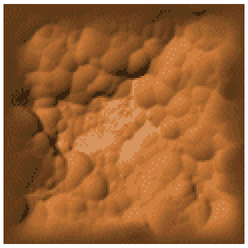
Filter Effects



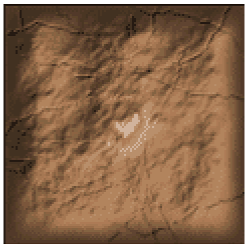
Peak - A conical mountain peak is blended into the height map. The Peak parameter controls the strength of the blending of the cone with the basic fractal pattern.



Dunes - With the Dunes option you can overlay the fractal pattern with a sinusoidal wave, thus simulating slopes similar to dunes. The number of dune Rows is entered via the corresponding parameter and with Turbulence you add a turbulent distortion to the sinusoidal wave.



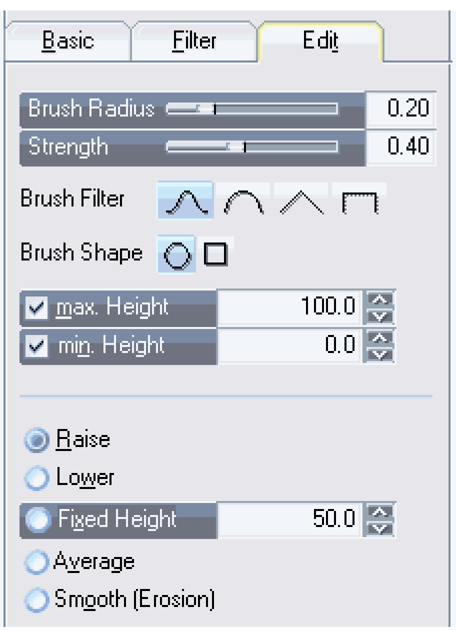
Crater - Adds craters to the landscape. You can input the number of craters and the average size.



Crevice - Adds crevices to the terrain. Again you can enter the number of crevices and their size. To see real smooth crevices a very high point resolution has to be set, so you've got to be cautions with this function.

Terrace - Subdivides a landscape into a given number of terraces. The Slope Ratio controls the transition from one terrace to the next. A low value results in steep terraces with wide level areas while a higher value calculates smooth transitions between terraces with only small areas of level ground.

8.8.4 Landscape Editor - Edit Height Map



Select the <Edit> tab in the landscape editor to bring the page with the painting tools to the fore. With these painting tools you can directly "draw" in the preview window to modify the height map. Thus you can raise or lower the ground or smooth eroded slopes, for instance.

Painting into the Height Map

Each time you change one of the various parameters on the Basic or Filter side of the dialog it results in a thoroughly new fractal pattern calculated for the height map. This changes when you switch over to the Edit work mode. In the instance you draw into the height map the landscape will

not be interpreted any more as a fractal function defined by various parameters but as a simple height map that you can paint on.
If you want to adjust any of the Basic or Filter parameters after painting into the height map the fractal pattern will be recalculated from the start and all modifications painted on the height map will be lost. The "Edited!" label that appears at the top of the dialog when painting in a height map will remind you of this fact. However, it is merely for information (also when loading edited library files), since all modifications can be undone anyway by operating the UNDO button.

Brushes and parameters:

Brush Radius - For a round brush this is the radius of the area of influence under the brush (as a fraction of the width of the landscape). For a rectangular brush this value defines the width of the brush (again as a fraction of the width of the landscape).

Strength - This parameter controls the strength of the effect, for example, how strong and how fast the surface is raised or lowered under the brush.

 **Brush Filter** - This filter specifies the distribution of the effect over the brush area.

The illustration shows the four different brush filters, in each case applied with the Raise function on a flat plane (Random switched off on the Basic side of the landscape dialog).
The top row was "painted" with a round brush and the row below with a rectangular brush. In general the round brush with the bell-shaped filter (on the top left in the illustration) shows the best results as it provides the smoothest transitions towards the edges.

Brush Shape - round or rectangular.

Minimum and Maximum Height - If you don't want to exceed a particular minimum or maximum height when applying the painting functions then switch on the corresponding option. At the start the Peak and Ground Level parameters from the Basic side of the dialog are entered automatically when switching over to the Edit side. But of course you can input any appropriate height within the world's limits of ±16,000 units.

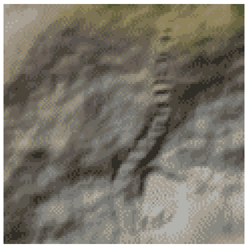
Brush Effects:

Raise - The area beneath the brush is raised.

Lower - The area beneath the brush will sink down.

Fixed Height - You can specify a fixed height up to which the area under the brush will be raised or lowered.

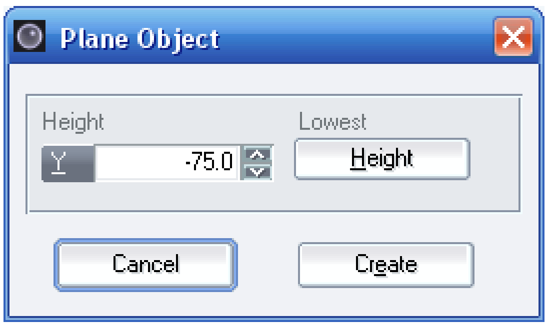
Average - This function levels out the area under the brush. The surface will be smoothed without removing to much detail.

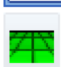


In this example the Average function was used to carve out steps in the slope by settings "points" with a rectangular brush. For further examples of use see the tutorial on landscape design.

Smooth (Erosion) - Smoothes away the rough edges on a surface.

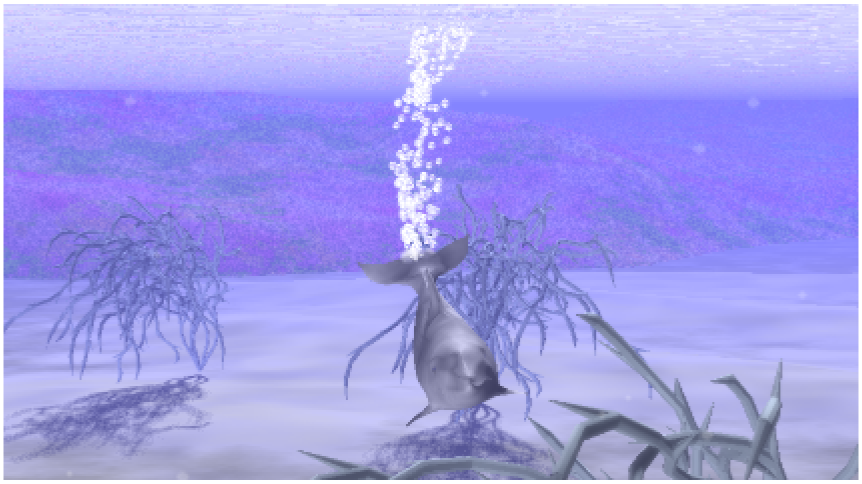
8.9 Plane Object



 - Menu "Objects - Plane"

In the Background dialog 375 you have the ability to easily select a sky model for the background. What may still be missing is a similar horizontal object. While there are a number of ways available to form a suitable surface, you can, instead, use the Plane-dialog to create a simple plane object. That is, by operating the "Plane"<Create> button you actually generate only a square facet. Now you may say that you could do exactly the same with the extrude editor 149 as well. The plane object, however, occupies a special position in picture calculation and object manipulation:

- The plane stretches to infinity at the horizon of the picture, while the sizes of other objects are restricted to the 3D area dimensions and so cannot coincide with the sky at the horizon.
- The plane cannot be scaled (it always stretches to infinity). Nor can it be turned.
- Only the height of the plane can be moved - so that you can adapt the "ground level" to suit the position of the other objects.

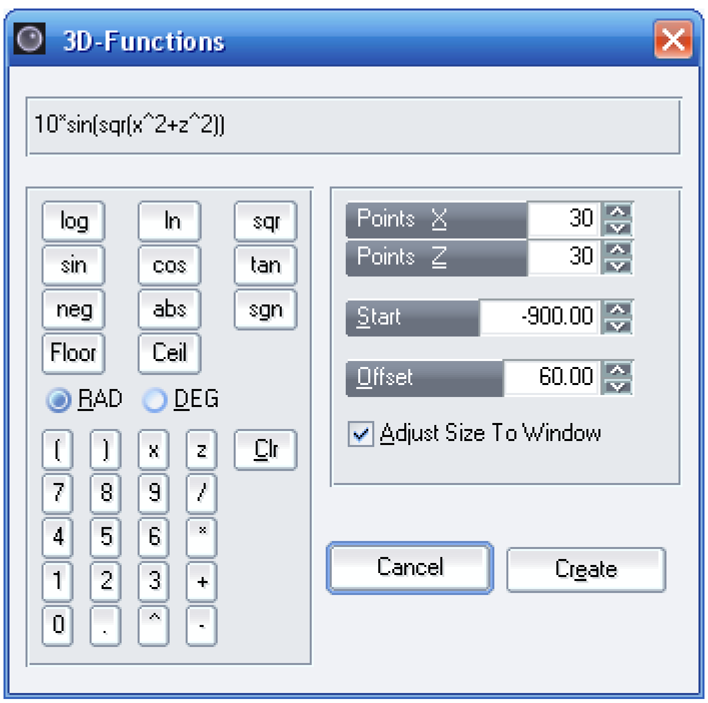


You can even set up a multi-layered plane model, in which you generate several planes that lie above one another. In the example above two planes were applied, a seabed plane and a transparent water plane above it.

With the **"Plane"-Y** parameter you can enter the height of the plane when it is created.

With the **"Lowest"<Height>** button the height of the plane is set to the lowest height of all the objects.

8.10 Functions Editor



 - Menu "Objects - 3D-Function"

A dialog box appears that contains a function "pocket calculator" and a parameter-box. By using the Function-generator you can generate three-dimensional functions relating to the X and Z

coordinates of the horizontal plane.
The procedure is quite simple: It is to generate a rectangular grid in the X, Z plane, whose grid coordinates will form the parameters of the functions. The starting point of the grid, the number of the points in X and Z direction, and the spacing of the individual points can each be edited in the parameter fields on the right side of the dialog box.
With help of the mouse you can input any function into the "pocket calculator" - which will not allow invalid input. (It will not close brackets, for example, if no brackets have previously been opened, missing brackets are automatically set, etc.). Press the **<Create>** button, and for each grid-point in the X, Z-plane the corresponding Y value is calculated from the function. These points are then assigned surface facets and the object is generated.

The Functions of the "Pocket Calculator":
You must use the mouse to input a function. Click on the required function as you would on a simple pocket calculator.

Angular Operators:
<sin> - calculate the sine of the argument.
<cos> - calculate the cosine.
<tan> - calculate the tangent.

The arguments can be interpreted as radians or degrees - determine by the **<rad>** and the **<deg>** buttons.

<sqr> - calculate the squareroot.
<ln > - natural logarithms to the base e.
<log> - ordinary logarithms to the base 10.
<neg> - corresponds to the sign for negative '-'.
<abs> - supplies the absolute value for the argument.
<sgn> - sign-function: Gives -1 for arguments less than 0. 0 if the argument is the same as 0 and 1 if the argument is greater than 0.
<Floor> - rounds down floating-point numbers to the next lowest integer number.
<Ceil> - rounds up floating-point numbers to the next highest integer number.

An opening bracket appears automatically behind each operator inserted. It does not allow function to be calculated based on invalid arguments, so at the X, Z coordinate the function value concerned is put at zero. It therefore generates a function object in every event.
Invalid arguments are:
- - Negative root expressions.
- - Logarithms of expressions that are less than or the same as zero.
- - Tangent values near to 90° or 270°.

Arithmetic Operators
<+> - Addition.
<-> - Subtraction.
</> - Division.
<*> - Multiplication.
<^> - Power.

Variables:
<x> - X grid coordinate.
<y> - Y grid coordinate.

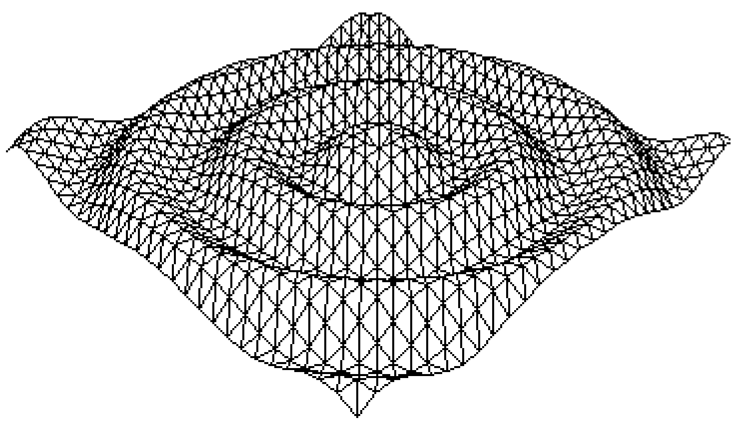
A function can be deleted with the help of the **<Clr.>** button.
The **<Create>** button generates the object from the function.

The **<Cancel>** button exits the dialog box
The **<Adjust Size to Window>** option resizes the created grid to the dimensions of the viewport window.

The Grid Parameters:

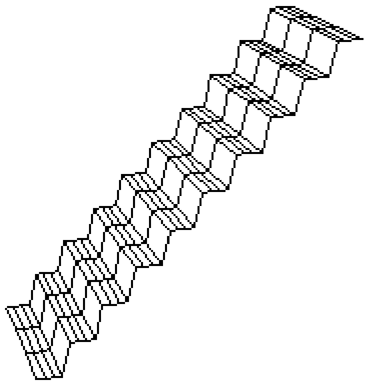
In the **"Points"** box you can input the number of points in the X and Z directions. At least 2 points must be entered for each direction. Consequently it forms a small band over each square of the grid formation to cover the entire rectangular shape. The maximum number of points that can be entered in each direction depends of the number of points available on starting and is automatically limited. The **"Offset"** box allows you to enter the spacing between individual points. For many function surfaces it is suggested that you use small offsets of about 1 or less to obtain a moderately fine structure. In this case, only a relatively small grid appears on the screen when you later view the object. To edit an object created by graphic function, you normally have to enlarge the in the Scale Object 206 menu. With the **"Start"** box you can enter the start for the X and Z parameters. Automatically scaling is enforced should the function values extend beyond the limits of the represented area.
Similarly, the function object appears centered at mid-point of the indication space, independent of the starting point and offset.

Examples



As an example, we want to generated the function **10*sin(sqr(x^2+ z^2))**.
Points in X direction: 30
Points in Y direction: 30
We enter an Offset of 60 and select the <deg> button.
To obtain a symmetrical form from the function we enter a starting value of -(30 points* 60 degs.)/ 2, i.e. -900. From the start point of (-900, -900), the function generator now produces a grid of 30* 30 = 900 points in the X, Z plane and records the Y value dependent on the function value for each point for the X, Z coordinates.
The sine function, which can assume a maximum value of 1, is stretched in the Y direction by the factor 10 in front of the sine bracket. Switch on the <Adjust Size to Window> option, as otherwise the created grid would be much too large to fit into the viewport windows and you would have to rescale it by hand. After creating the object, rotate it to using the Rotate Object 211 menu to see the surface structure clearly. Rotate the object by about -45 degrees, first about the Y axis and then about the X axis. This produces a view of the object as seen in the illustration.

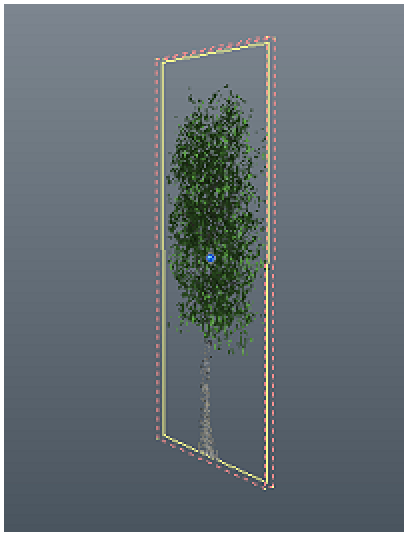
Next example: **Floor(x)**



Incredibly, with this very simple function you can easily create staircases. The parameters:
Points x: 40
Points z: 4 (actually you only need 2)
Start: 0
Offset: 0.25
With these parameters you create a small ribbon of stairs that will have to be scaled 206 in z-direction to widen it.

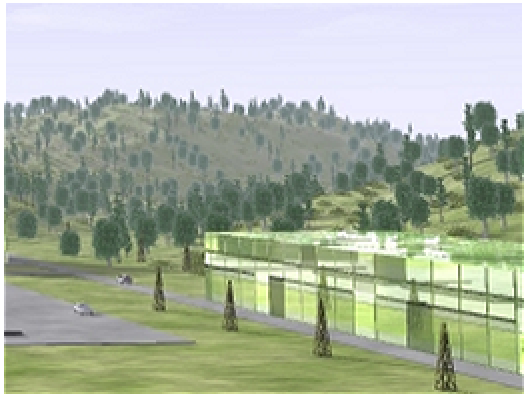
8.11 Billboard

This primitive object consists only of a flat rectangle that serves as a projection plane for masked bitmap textures. There are many application areas where you will find billboards very useful, for instance, for adding masked pictures 300 of people or trees to architectural scenes, or you can project animated sequences of explosions or smoke onto billboards to insert them into your animation. By replacing complex 3D objects with billboards you can save a tremendous amount of space and render time. Billboard primitives are created with all necessary default settings making it especially easy to use them as projection planes. They come with a basic transparent material, uv-texture coordinates and a default tree bitmap material, so you only need to adjust the size of the billboard and change the material's picture with your own bitmap. Furthermore, billboards have the auto-aligning object property 290, set so that they will always face towards the camera. The default setting is to rotate only on the horizontal world plane around the y-world axis (so that billboards of plants or people don't lose ground under their feet). If you'd rather have them rotate about all axes so that the billboard's z-axis always points perpendicular to the camera you can change this in the object properties dialog. Of course you can also change all other default settings, for instance switch off the auto-aligning function and use the billboard just as a simple rectangular primitive object for a modelling purpose.



The billboard primitive in camera view. The auto-aligning function was switched off so that the rectangular, flat structure can be seen from a side-angled view. Usually a billboard will be applied in combination with the auto-aligning function so that it always faces towards the camera to create the illusion of a real three-dimensional object.

A typical billboard application - a masked picture of a real person was placed in an architectural scene (Office design by Jürgen Kaupp, Haiterbach).

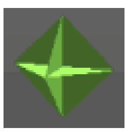


In this scene the billboard primitive with its default tree bitmap was used to enforest a whole landscape. A single billboard tree was applied as a reference object for a particle system to drop thousands of copies of the reference billboard onto the landscape. The auto-aligning function is inherent to the particle billboards so that all billboards always face the camera.

Another example for the application of billboards. Both the explosion and the exhaust smoke of the jets were realized with help of billboard projections (see tutorials: Explosive Fire 113 and Billboard Smoke 117)

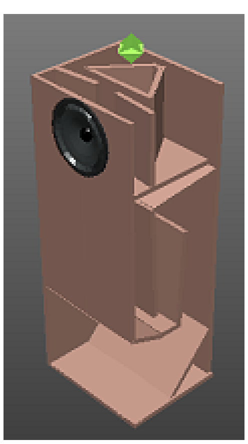
See also:
Material Blends to Mask and Cut Out Surface Areas 300
Object Properties - Align Object with 290
Tutorial - Explosive Fire 113
Tutorial - Billboard Smoke 117

8.12 Group Object



This is how a Group Object will look like in the viewport window. A Group Object is a simple primitive object that you can handle like all other objects, e.g., you can move, rotate or scale it. The only difference is that Group Objects are always hidden in the final rendering - you can only see them in the viewport windows while working on your project. To create a Group Object just select the "Objects - Group Object" entry in the menu bar. There are two major uses for Group Objects:

Use Group Objects to manage object groups
Group Objects are simply used to group together a number of objects by linking them under the Group Object in a hierarchy 140 (you can also use the "Group Selection 133" entry of the popup menu 133 to automatically subordinate all currently marked objects under a new Group Object).
This provides clarity in the Object Selection 133 window (child nodes can be hidden) and you can select or de-select a whole group of objects just by clicking on the parent object of that group. For instance, if you have created a car consisting out of several hundreds of objects, simply link them under a Group Object "car". Then, every time you want to move that car, you only need to mark the "car" Group Object and move it to its new destination, all children will follow automatically.



Example: This box consists of 30 individual elements, all of which are subordinated to the green Group Object. To move all parts of the box at the same time you just need to select the parent Group Object and use it as a "grip" to carry all the elements of the box with it.

Group Objects provide movable reference points in animations

Group Objects can serve as reference points for rotating or scaling object groups in animations. If, e.g., you want to rotate a group of objects around a common midpoint you would link this group under a Group Object and just rotate the Group Object with its child objects. It's the same for camera rotations. If you want to circle the camera in an animation around another object or group of objects then just place a Group Object at the visual focus within that group and link the camera under this Group Object. Use the "Focus" camera function to center the Group Object in the camera view, then rotate the Group Object and the camera will rotate with it in a perfect circle and with the focus always centered to the object group.

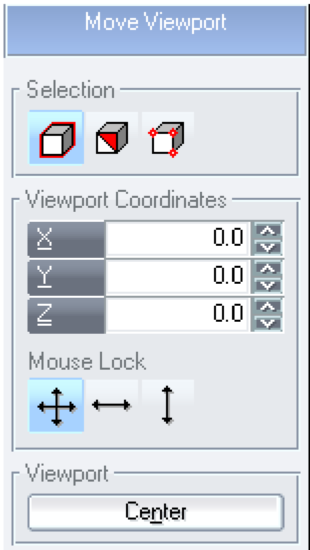
Part IX

9 Work on Objects, Skeletons and Textures

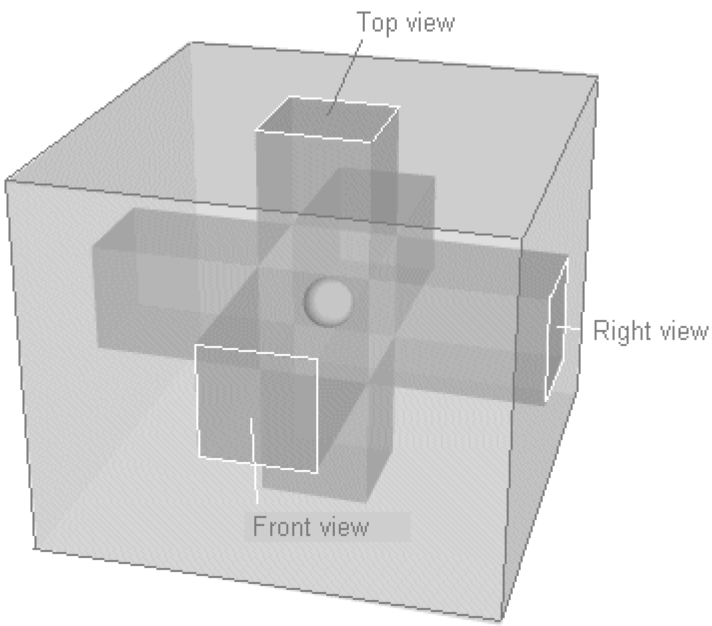
How to set up your scene by moving, scaling or rotating objects or working on individual facets and points.

9.1 Move Viewport

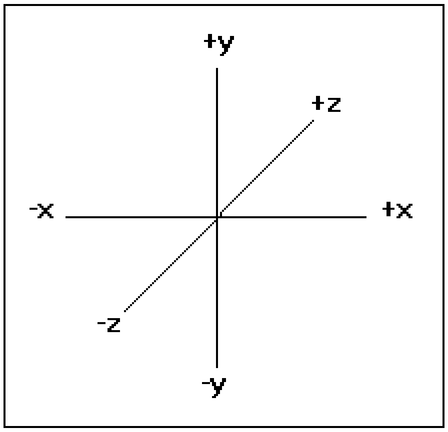
 - Menu "Edit - Move Viewport"



Orthogonal views in the viewport windows

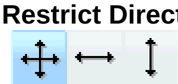


The viewport window 22 can be freely moved in the three axes directions until you reach the preset of 3D area limits, which lie at 224 = 16.777.216 units. However, the window moves only to the edges of this area, so objects are always visible in front of the viewport window.
You can move the window-detail in any work-mode simply by pressing the left and right mouse button simultaneously while moving the mouse. However, here in the "Move Viewport" work-menu it is sufficient to hold only the left mouse button pressed. Here you can also read the coordinates of the center of the window-detail or you can input it directly via the keyboard.



The order of the X, Y and Z-axes in MR-3D Designer's left-handed axes system 8.

Restrict Direction of Movement

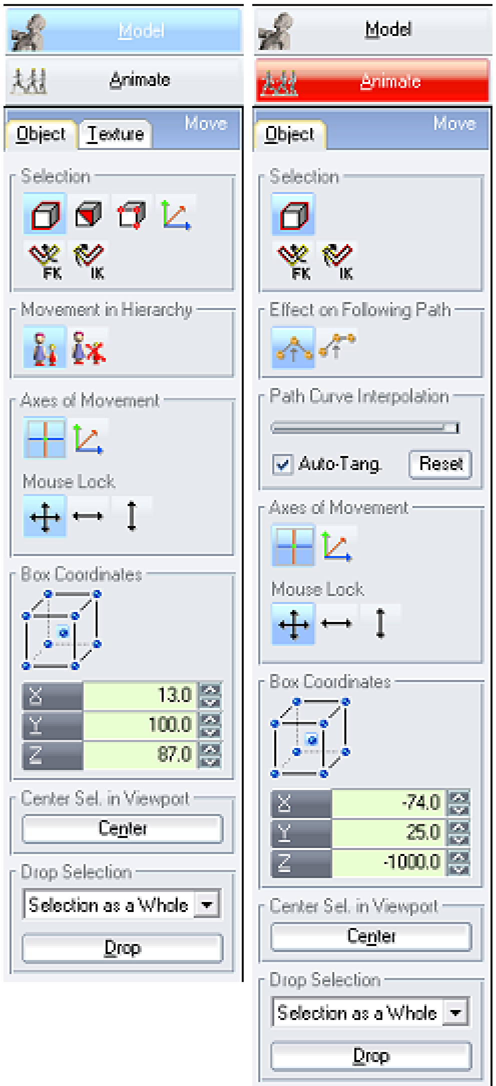
 This three buttons permit movement of the window-detail. If the left button is active, the window-detail can be freely moved. If the button with the horizontal arrow is active, then the vertical movements of the mouse are ignored, so that it permits movement of the window in the horizontal direction only. The converse is true for the button with the vertical arrow.

Line Up Viewports on a Marked Selection (Shortcut "V")

If you choose the **<Center Viewport>** button the window is precisely centered over the current marked selection 138. You can also call up this frequently used function in every work modes simply by pressing the "V" key on your keyboard. Another possibility is to choose the function from the popup menu 133 which opens after a click with the right mouse button in a viewport window.

9.2 Moving Objects or Textures

 - Menu "Edit - Move Object/Texture"



"Move Object" - Modeling- versus Animation Mode.
The "Move Texture 197" tab is only available in Modeling Mode.

Differences between Modeling Mode and Animation Mode

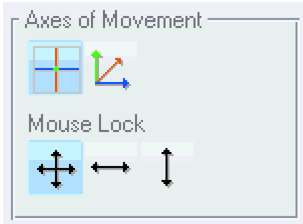
Depending on wether you work in Modeling- or Animation Mode 19 the tool window presents a slightly different selection of tools. For instance, in Modeling Mode you can change an objects basic shape by editing individual facets or points of an object. In Animation Mode you can only work in object selection mode and editing of individual facets and points is no longer available. However, you can animate objects on a per point basis if you use the skeleton deformation 242 functions (Skin and Bones).
The "Texture"-pages, where you can adjust the position 197, size 206 and alignment 211 of procedural- and bitmap textures will be hidden in Animation Mode, too. You can move, scale or rotate textures only in Modeling Mode. Based on this initial alignment, textures will behave very flexible in animations. When scaling or deforming objects or skins in an animation all textures will be deformed properly with the object.

Moving Objects, Facets, Individual Points or Object Axes with the Mouse

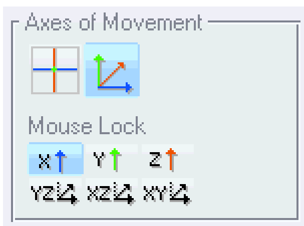
You can move a marked selection 138 simply by pressing and holding the left mouse-button and dragging the selection across the work-surface in the desired direction.

Axes of Movement

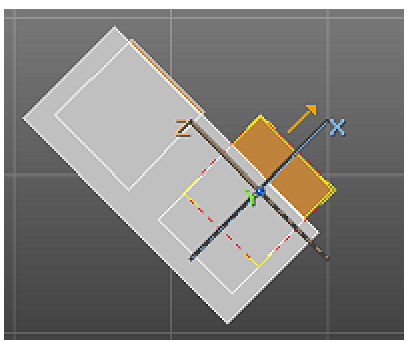
You can move a selection along the world axes or along the object axes 8 of each object:



If movement along the world axes is selected, then objects are moved only on the 2D-viewport plane of the respective viewport. The movement of the objects can be restricted by using the three "Mouse Lock" buttons pictured in the tool-window. If the left button is active, the objects can be freely moved in the viewport window. If the button with the horizontal arrow is active, vertical movement of the mouse are ignored, so that it allows the objects to move only in the horizontal direction. The converse is true for the button with the vertical arrow.

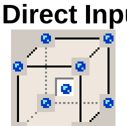


If the object axes are selected for the axes of movement, you can move a selection along the object axes of the marked reference object 138. You can restrict the movement again to one of the object axis by selecting the x-, y- or z-button. When you choose the button under the x-, y- or z-axis button, then you restrict the movement to the plane standing perpendicular on the respective axis.



Example: The illustration shows a model of a cupboard with movable drawers in "Top" view . Since the cupboard does not stand right-angled on the viewport plane, you cannot open or close the drawer precisely when moving it along the world axes on the viewport plane. The drawer would slightly run off the tracks at the left and right side. However, in object axes mode we can conveniently move the drawer in its tracks along the drawers x-axis.

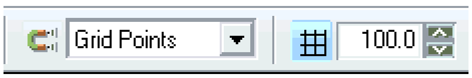
Direct Input of Coordinates from the Keyboard



The coordinates indicated in the toolbox always refer to a particular corner of the bounding box that surrounds the reference object 138 of the marked selection. You can specify the current point indicated by selecting one of the nine buttons placed at the corners and in the center of the box image. In the viewport window the corresponding corner of the bounding box will also be marked with a similar blue pin-button.

You can directly input the position for the bounding box (and thus the position of the object) via the keyboard. Each alteration to the coordinates leads immediately to a corresponding change in position and a re-draw of the scene.

Snap Functions



In the button-strip directly above the viewport windows on the right you find the different snap functions that will facilitate the positioning of the objects in the scene. If, e.g., you switch on the snap function for grid lines, then an object will be "caught" automatically by grid lines when near to them. You can switch on the snap function with the magnet button next to the snap selection box. The particular point of the object selection that is caught by the snap function is given by the selected bounding box corner of the marked reference object. You can choose to snap a selection to:

- Grid Lines (The viewport grid can be switched on or off via the corresponding grid button - the size of the grid can be edited in the parameter field next to it)
- Grid Points
- Object Axes - 2D and 3D
- Object Center - 2D and 3D
- Object Points - 2D and 3D
- Object Lines - 2D and 3D

Snapping can be done only in the 2D-working plane or in 3D mode with additional depth testing enabled. Snapping can considerably simplify working on a project. Think of a wall you want to align next to another wall. If you activate the "Object - Lines" snapping function and the corresponding corner of the wall you want to fix to the other wall, then you just need to move the wall near to the other wall and it will automatically jump into place.

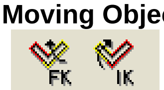
Center Selection in Viewport

The <Center Selection> function moves a selection of marked objects, facets or individual points to the exact center of the window. You can call up this frequently used function also in other work modes. A click with the right mouse button in a viewport window will display a popup selection 133 with a list of functions including the "Center Selection" function.

Drop Selection

Use the <Drop Selection> option to drop objects to the ground. If an object hovers above another object it will land on it, if not, nothing will happen. You can drop selected objects or hierarchies as a whole object or let each object/hierarchy fall down on it's own path.

Moving Objects using Inverse- or Forward Kinematics

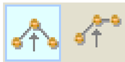


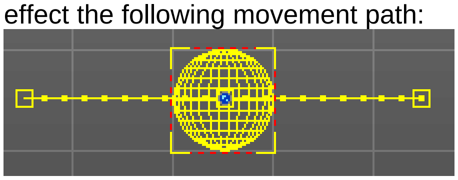
Beneath the 3 selection buttons for selecting objects, facets or individual points 138 there are 2 additional buttons for the Inverse- and Forward Kinematics working modes. With help of Hierarchical Kinematic Modeling it becomes very easy to position a group of joints arranged in a hierarchy. More details in the corresponding chapter about "Inverse- and Forward Kinematics 222".

Movement in Hierarchy *(only in Modeling Mode)*

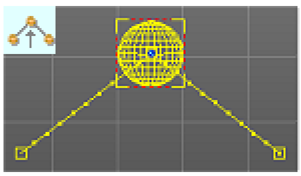
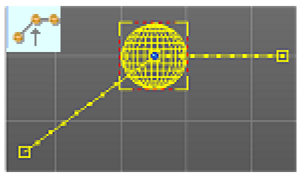
Usually hierarchical subordinated child objects follow all movements of their parent objects. However, in Modelling Mode you can temporarily switch this off. If you activate the right button of the two buttons shown above then you can move a parent object alone without moving its children with it. But this is a movement on model basis. This means that the displacement will take place relatively to its children and the new distance between parent and children will be passed to the complete animation.
In Animation Mode this function is hidden - all movements of parent objects will be inherited by their children again.

Effect on Following Path *(only in Animation Mode)*

 Every time you move an object in Animation Mode a new position key is automatically created, or, if a key already exists on this frameposition, it is updated. Now, if you are moving backwards in the animation because you want to relocate an object at a particular frame position, then you can decide via the two buttons depicted above, how the repositioning of the object will effect the following movement path:



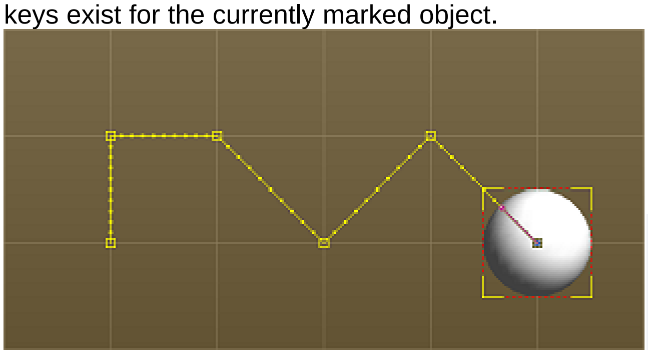
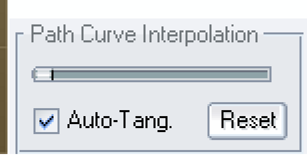
Example: The illustration above shows a sphere moving through 3 key positions from left to right. We move back to the second keyframe where the sphere is located in the middle of the window.

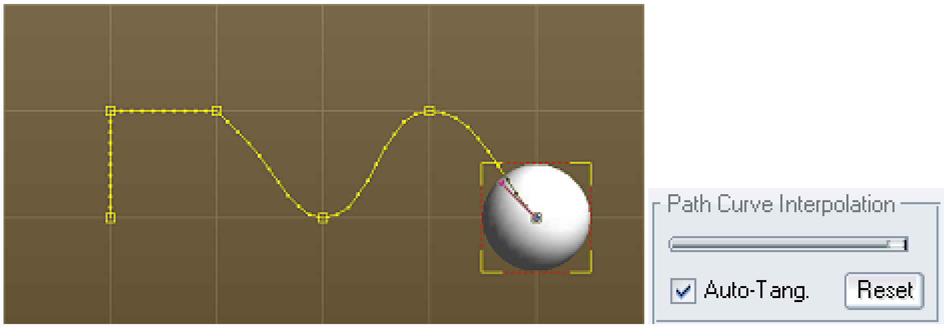
Now, if you move the object in "Absolute Position" mode (illustration on the left), the position of the object will be changed only for the current keyframe without influencing the following movement path - in the course of the animation the object returns back to the old movement path after this keyposition.
However, if you select the second button "Relative Movement - Path follows Movement" then the object will be moved with its movement path (illustration on the right).
Apart from correcting movement paths this function is very useful if you want to make copies of already animated objects. After copying an object you just need to move it a little bit to the side in "Relative" mode including its animation path, and soon you have a whole group of figures running along the same, slightly displaced, movement paths.

Path Curve Interpolation *(only in Animation Mode)*

With the "Path Curve Interpolation" slider you can define if an object moves on straight lines from one key position to the next one or if the key positions are used as reference points for the calculation of a curve that passes through these points. The amount of curve interpolation to be applied for the current key position can be continuously adjusted with the slider from the left (straight lines) to the right (full curve interpolation). The amount of curve interpolation is saved to the position keys of the objects, therefore the slider is only enabled on timeline positions where position keys exist for the currently marked object.

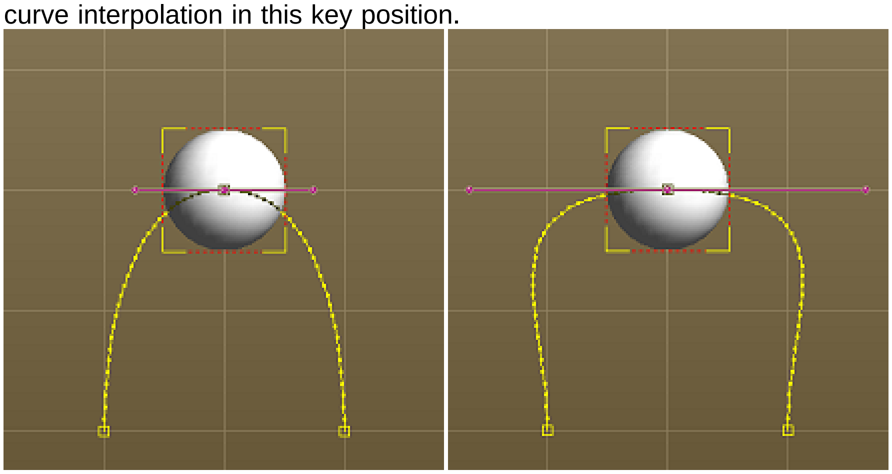
 

In the above illustration a sphere is moving along a path defined over six keyframes. Since for all keyframe positions the Curve Interpolation slider has been set to its minimum value at the left the movement between the individual keyframes will be linearly interpolated.

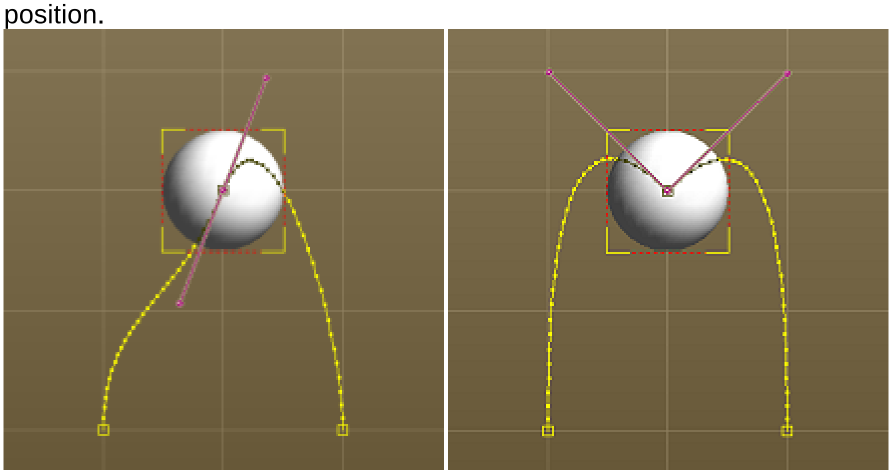


The picture shows the same key positions as before. However, for keyframes 4 - 6 the Curve Interpolation slider has been adjusted to full curve interpolation by moving the slider to its maximum value at the right. From this, you will see in the picture how a straight movement from keyframe 1, through keyframe 2 to keyframe 3 is defined, and then a curved movement through keyframes 3 - 6.

The Tangent Line to a Curve - Together with the position, the tangent line to the curve is saved in each position key of a movement path. The tangent to a curve defines the gradient slope of a curve in a given point. Each time you move (in time) to a key position of a movement path, the tangent line to the curve in this point will be painted. Now, by clicking on the end points of the tangent line you can stretch or rotate the tangent, which directly influences the curvature and the strength of the curve interpolation in this key position.

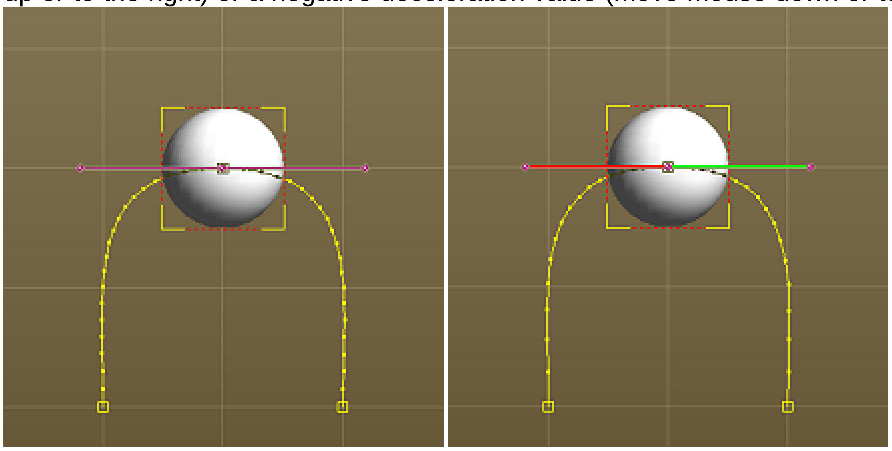


Example: A curve can be stretched by elongating or shortening the tangent line at a given key position.



Rotating the tangent line will change the curve progression through the key position (picture on the left). You can manipulate both ends of the tangent line separately, if you hold the <Ctrl> key pressed when clicking on an end point of the tangent. In the right picture above this function was used to create a sharp bend in the curve.

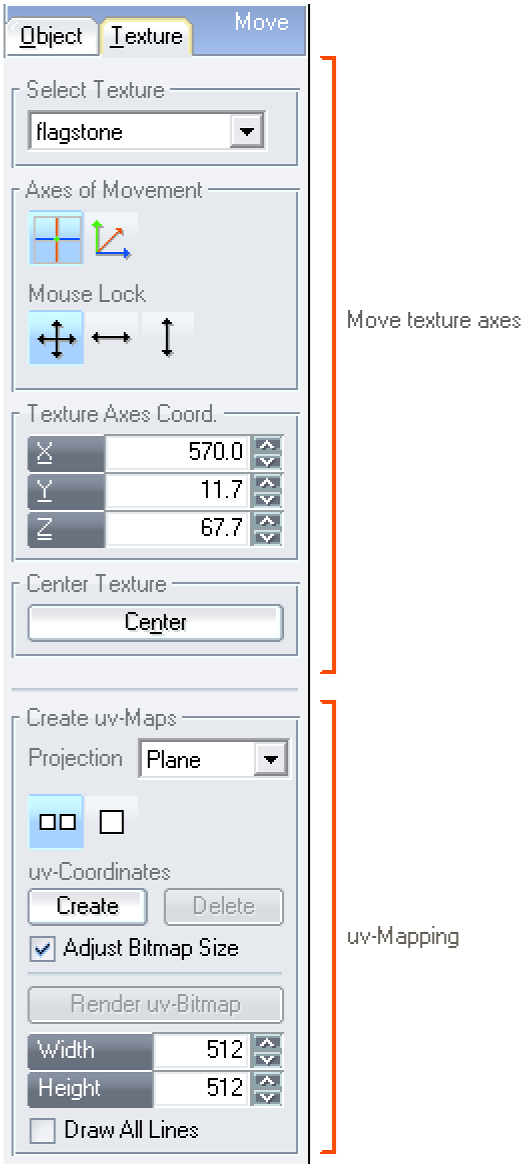
Acceleration Parameters - The amount of acceleration towards a keyposition and from a keyposition can also be controlled via the tangent's end points. Simply hold down the <Shift> key when clicking on the left or right endpoint of the tangent line and move the mouse. A green or red bar is painted along the tangent line in dependence of a positive acceleration value (move mouse up or to the right) or a negative deceleration value (move mouse down or to the left), respectively.



The picture on the left shows a uniform speed (i.e. similar step width) along the path. For the picture on the right we adjusted a deceleration towards the middle key position (step width getting smaller) and an acceleration (step width getting larger) from the middle key position to the next key. You can also manually edit the parameters for the acceleration to and from a key position in the animation editor 409, but it is a lot of easier when you use the tangent manipulation function, because here you can see in realtime how the path changes when pulling at a tangent's end point.

Auto-Tangent - If you move an object then automatically an optimized new curve progression through the changed position will be calculated. This will not only change the tangent at the present key position but also the tangents of the immediate key neighbours. You can switch off this automatism, if you want to retain the tangent alignment and thus the general shape of the curve.
Reset Tangent - If you have manipulated a tangent line you can reset it to the optimized length and alignment, as it is calculated by the Auto-Tangent function.

Moving Procedural Textures or Bitmaps *(only in Modeling Mode)*

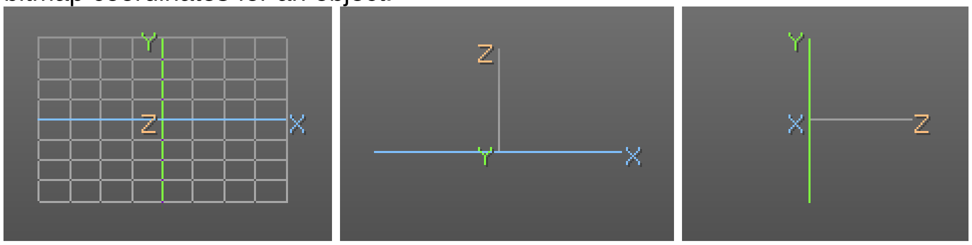


On operating the "Move"-<Texture> index tab, the content of the tool window changes. In the upper half of the tool window you will find now all functions to reposition textures. The lower part of the window provides all tools for the generation of uv-coordinates and uv-bitmaps. You will also find this section in the corresponding work menus for scaling and rotating textures. Read more about uv-Mapping here: "Generating uv-Coordinates and Bitmaps for uv-Mapping 218".

Texture Axes Systems

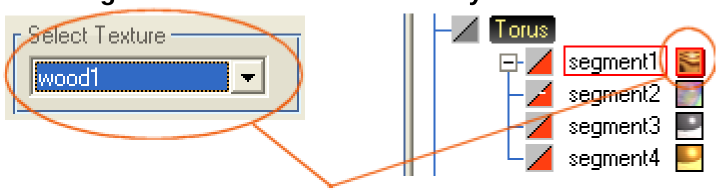
When you assign a new material to an object then always together with the material reference a new texture axes system is generated for the object. You can use this texture axes to adjust materials indivdually on every object, for example, to align a bitmap texture on an object. Moving a texture axes system will reposition the origin of a bitmap or a volumetric procedural texture, while rotating the texture axes will change the threedimensional course of a volumetric texture or the direction of a bitmap projection, respectively. Scaling a texture axes system will also result in a change of size of the texture pattern or bitmap.

Exception: When applying uv-Mapping 218 a surface point of an object is mapped directly into a bitmap picture. Therefore, manipulating the texture axes of a uv-mapped material will have no effect on the appearance of the material. However, you need a texture axes system when creating uv-bitmap coordinates for an object.



If you select a material from the material list of an object then the corresponding axes system is drawn in the viewport windows. If a bitmap material is selected then a grid indicates the picture plane with the dimension of the grid exactly matching the dimension of the bitmap. The direction of the picture projection can easily be verified by the direction of the z-axes, which is perpendicular to the bitmap grid. The location of the top or bottom of the bitmap can be determined from the y-axes. The x-mark on the x-axes indicates the right edge of the picture.
The texture axes of procedural materials are drawn in the same way as bitmap materials. Thus, if you resize a procedural material, you can easily follow the change of the texture size by the relative change of the size of the grid.

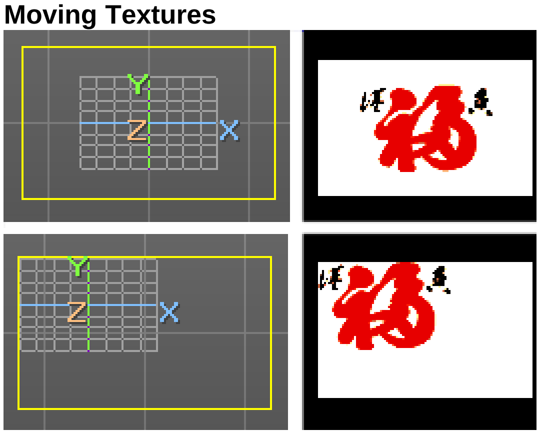
Selecting a Material's Texture Axes System



Choose a material name from the "Select Texture" list
or click on a material icon in the Object Selection window.

In the Object Selection window all materials assigned to an object or to a frozen facet selection are listed in a row of small icons behind the name of the object. Just click on a material icon to select the material and with it the corresponding texture axes system. An additional material list is provided at the top of the tool window. The "Select Texture"- select box lists all materials of the currently marked object by their material names instead of little thumbnail icons.

Moving Textures



The illustration above gives an example of a rectangular box with a bitmap projected onto its center. After that the bitmap texture axes have been moved to the upper left corner of the object, which results in the bitmap projection being similarly displaced.

Similar to moving objects, you have again the choice to move textures along the world axes or along the individual axes of the texture grid.

The coordinates in the tool window indicate the center of the texture axes.

The "Texture/Bitmap" <Center>-button returns the texture axes back to the precise center of an object.

9.3 Scaling Objects or Textures

 - Menu "Edit Object/Texture"

"Scale Object" - Modeling- versus Animation Mode.
The "Scale Texture 206" tab is only available in Modeling Mode.

Differences between Modeling Mode and Animation Mode

See - "Moving Objects or Textures 197"

Scaling Objects, Facets or a Point Selection

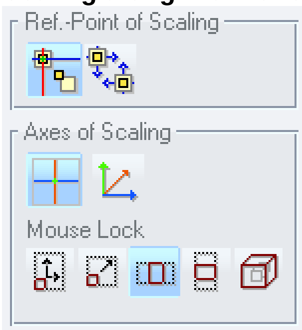
By using the new selections in the tool window you can change the size of a marked object selection 138. The program offers a variety of possibilities to do so. You can, e.g., simply enter the dimensions of the bounding box that surrounds the object selection. Using your mouse to drag the objects to the right size directly in the viewport window is another possibility.

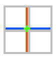
Finally, you can enter scaling factors, e.g., enter 2 for the y-axis to double the height of the object.

Reference Point and Axes of Scaling

The reference point from which scaling will take place depends on whether you are scaling along the world axes 8 or along the object's own axes system 8:

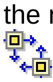
Scaling along the World Axes *(only in Modeling Mode)*



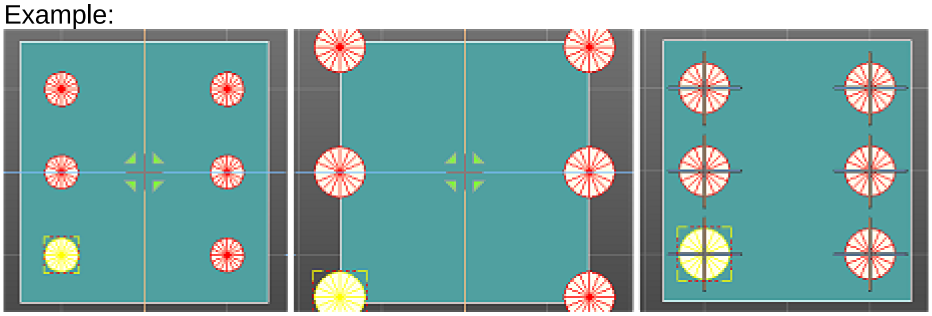
Select the  "Scale along World Axes" button in the tool-window in order to scale parallel to the X-, Y-, Z-world axes. Scaling with the mouse will take place only in the 2D-viewport plane or symmetrically in all three dimensions.
With help of the "Mouse Lock" buttons you can restrict the scaling directions again. If the left button is active, objects can be scaled in the horizontal and vertical viewport plane. If you choose the second button, objects are scaled evenly in the 2D-viewport plane.
If the third button is active, the vertical mouse-movements are ignored, so objects are only scaled in the horizontal direction. The converse is true for the fourth button. If the last button with the box is selected, objects are evenly enlarged or reduced in all three dimensions.

Reference Point of Scaling in World Axes Mode

If you scale along the world axes you have two options for the reference point from which scaling will take place:

-  Crosshairs - If you select this button, then crosshairs appear in the viewport windows. You can move them freely with the mouse to specify the reference point of scaling. To "grab" the crosshairs simply click with the mouse into the area between the 4 arrows in the center of the crosshairs. After positioning the crosshairs - if you want to scale the marked selection again with the mouse - you have to leave the area between the 4 arrows of the crosshairs again.
-  Scaling will take place from the object axes center of each marked object. In hierarchies 140 the object axes of the top most parent of the marked selection will be used.

Example:

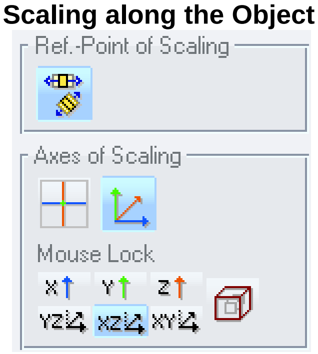


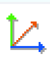
In the picture on the left you see the initial scene with 6 cylinders standing on a platform. You want to enlarge the cylinders evenly in the 2D-viewport plane, without changing the length of the cylinders. You have chosen the crosshairs as reference point of scaling and moved them to the center of the marked cylinder group. The picture in the middle shows the result after the scaling operation. The cylinders have the correct size now but the scaling operation also drifted them away from the reference point, so you have to reposition them again on the platform. The third picture shows the result after scaling along the object axes of each marked object. Since all objects were scaled from the origin of their own object axes system, all objects stayed in place. In this situation scaling along each object's own axes center was advantageous. However, if you want to scale a group of objects as a whole entity, e.g, if you want to scale a house with all its elements, then you have to scale the whole group from a single reference point, otherwise all elements would grow at their own positions and overlap each.

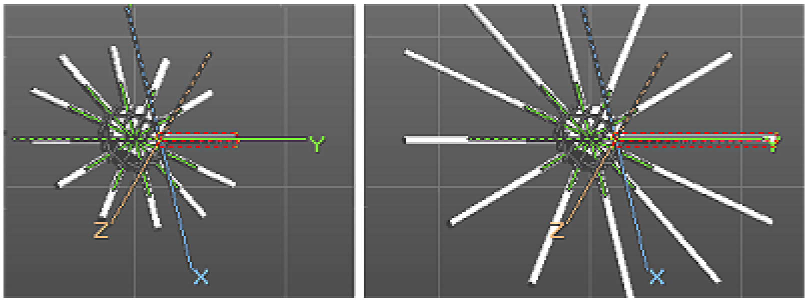
Why is Scaling along World Axes only available in Modeling Mode?
In an animation you always need a traceable reference point as well as reference axes that are used to perform the scaling operations recorded in the keyframes. These reference axes systems are always defined by the object's own object axes or the object axes of a parent object, if movements are inherited in a hierarchy.

Scaling of Analytical Objects
Analytical defined objects will only be scaled if this operation does not conflict with the mathematical description of the object's shape. This is made clear by a sphere, for example, which if scaled along the X axis is no longer a sphere, but becomes an ellipsoid that can no longer be described simply by its center and a radius. All objects can be increased or reduced symmetrically, of course. Furthermore, analytical cylinders can be scaled symmetrically in their base as well as along their longitudinal y-axis. If an analytical object is scaled in a group with other objects and the scaling would deform the shape of the analytical object then it will only displaced with the scaling movement of the group.

Scaling along the Object Axes



Select the  "Scale along Object Axes" button in the tool-window in order to scale along the object axes 8 system of each object. In this scaling mode the object's own axes center always serves as origin for the scaling - in hierarchies 140 the object axes center of the top-most parent of the marked selection will be used.
Via the "Mouse Lock" buttons you can choose an individual axis or a plane in which you want to perform the scaling. For instance, if you want to lengthen a cylinder, you would activate the longitudinal "Y"-object axis of the cylinder but if you want to enlarge only the base of the cylinder, you would activate the "XZ"Mouse Lock button below the "Y" button - then the scaling takes place only in the xz-plane standing perpendicular on the y-object axis.
If the last button with the box is active, then objects are proportionately enlarged or reduced in all three dimensions.



Example: The illustration shows a hub with 12 embedded spokes. The spokes are a little too short and we want to lengthen them along their longitudinal y-object axis, without broadening the circular base of the spokes. When constructing the cylindrical spokes, the object axes of each spoke have been moved already down to the base in the center of the hub. That simplifies the following operations. First we select the "Y"-axis button as the current "Axes of Scaling". Then we mark 138 the 12 spoke cylinders we want to lengthen. Thereupon the object axes systems for all marked objects are drawn. For clarity, only for the reference object the complete axes system is drawn, for all other objects only the selected scaling axis is painted. In the illustration you can see the green y-axis running along each of the spoke cylinders. Now you just need to click into a viewport window and move the mouse - the spokes will virtually grow out of the hub or sink back into it.

Box-Dimensions

With help of the x-, y-, and z-parameters in the "Box-Dimension" field you can exactly define the dimensions of the bounding box surrounding the marked reference object 138. If the object has been rotated and is no longer aligned to the world axes then change to "Axis of Scaling - Scale along Object Axes" mode, so that the bounding box is aligned to the axes system of the marked object. When you input the dimensions for analytical objects 147, all input values will be automatically completed, so that the mathematical description of the object is maintained. If, for instance, you enter the base radius of an analytical cylinder by changing the value for the x-dimension, the z-dimension will be adjusted automatically.

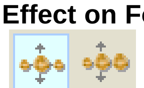
Specify a Scaling Factor

When scaling a selection with the mouse in the viewports you can simultaneous read the scaling factors from the "Scaling Factor" toolbox.
However, you can also input the scaling-values for the individual axis directions directly via the x-, y-, z-parameters in the "Scaling Factor" toolbox. Then operate the <Scale> button, to implement the change in size.
The <Clear> button in the tool-window beneath the scaling parameter resets all values to 1.

Scaling in Hierarchy *(only in Modeling Mode)*

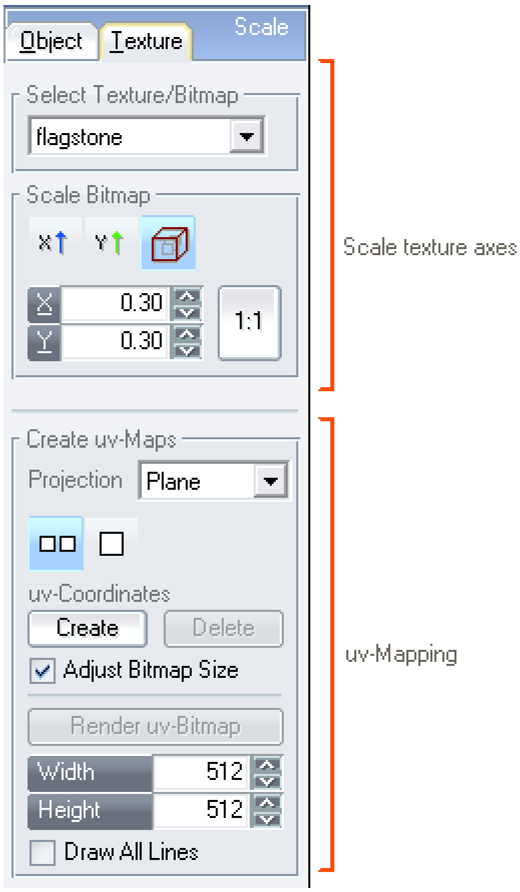
 Usually, hierarchical subordinated child objects follow all movements of their parent objects. However, in Modelling Mode this can be temporarily switch off. If you activate the right button of the two buttons shown above then you can change the size of a parent object alone without scaling the children with it. In Animation Mode this function is hidden - all transformations of parent objects will again be inherited by their children.

Effect on Following Path *(only in Animation Mode)*

 A new Scale key is automatically created every time you scale an object in Animation Mode, or it is updated if a key already exists at this frame position. If you are moving backwards in the animation because you want to adjust the size of an object at a particular frame position, you can decide, via the two buttons depicted above, if the size of the object is changed only at the current keyposition or if the object is scaled also with the same scaling factor in all following

keyframes. The second option is more or less the same as if you scale an object in Modeling Mode - the object changes its size relatively for the remainder of the animation.

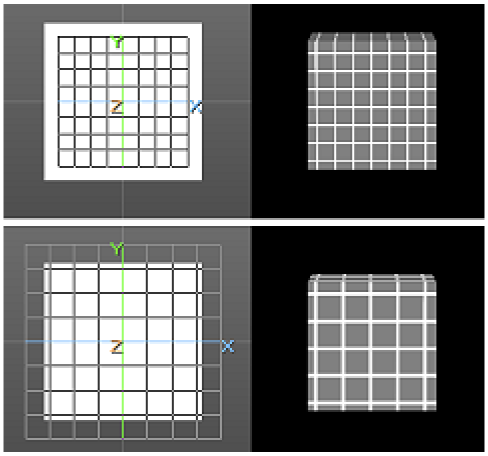
Scaling Procedural Textures or Bitmaps *(only in Modeling Mode)*



Operating the "Scale" <Texture> index tab changes the content of the tool window to the above. In the lower part of the window you will find again the tools for the generation of uv-coordinates and uv-bitmaps. Read more about uv-Mapping here: "Generating uv-Coordinates and Bitmaps for uv-Mapping 218".

Texture Axes Systems and Material Selection
See - Move Textures 197

Scaling a Texture
As explained in the chapter about texture axes systems 197 you can resize a bitmap material or a procedural texture individually on every object by scaling the corresponding texture axes generated for the object with the material reference. The scaling is carried out as usual with the mouse directly in the viewports or by direct input of the scaling factors. Bitmap materials can be scaled along their x- and y-texture axes representing the bitmap's width and height. Procedural textures are always scaled as a whole, therefore only one parameter is indicated for the scaling factor.



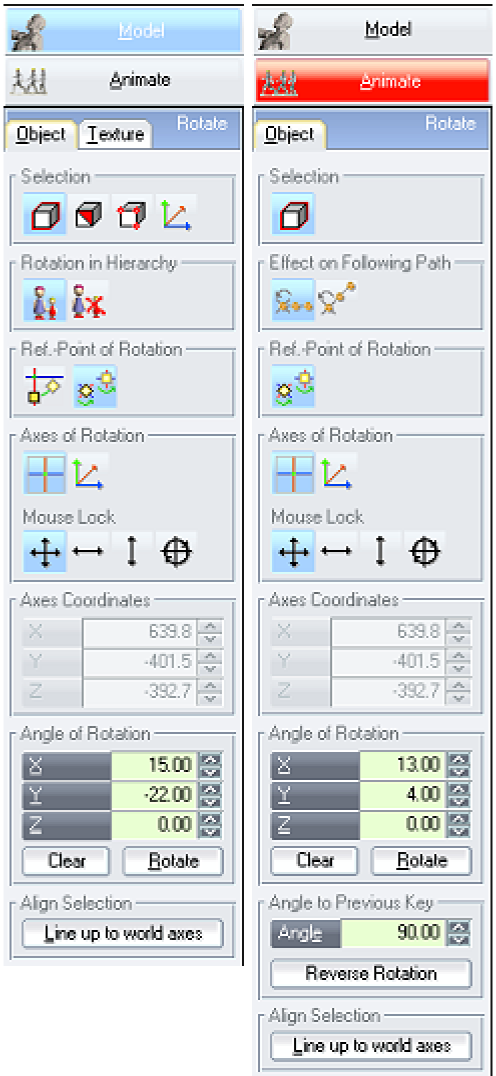
The illustration shows the scaling of a procedural texture 313.
You can also adjust the total size of a procedural pattern in the Material editor 313, but then the change of size is also applied to all objects referencing to that material. But with each material assinged to an object an individual texture axes system is created for the object. Therefore scaling the texture axes system will only resize the material for that particular object the texture axes belongs to.

Scaling a Picture to the Original Size

If you choose the <1:1> button, a scaled bitmap will be rescaled to its original size. This is primarily of importance in the particular case when you want to change the projection mode 328 in the material editor to a cylindrical or spherical projection. For example, in the case of a cylindrical projection a bitmap is scaled automatically so that a scaling factor of 1 in the X-dimension would result in a picture perfectly wrapped around the object, with the start and end edges of the picture exactly meeting each another. A value less than 1 would lead to a label effect and a value exceeding 1 would result in picture overlap. The same applies to spherical projections. A value of 1 in the X- as well as in the Y-direction would result in the bitmap wrapping to exactly fit the object. Lower values would again lead to a label effect. With these projection modes you should proceed from the original size, only the reduction in Y-direction for the cylindrical projection being meaningful, or a reduction in both directions when you want to obtain a label effect in cylindrical and spherical projections.

9.4 Rotating Objects or Textures

 - Menu "Edit - Rotate Object/Texture"



"Rotate Object" - Modeling- versus Animation Mode.
The "Rotate Texture 211" tab is only available in Modeling Mode.

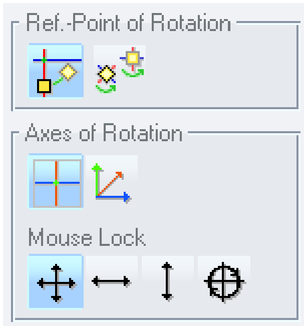
Differences between Modeling Mode and Animation Mode

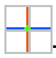
See - "Moving Objects or Textures 197"

Rotating Objects, Facets, a Point Selection or Object Axes

The tool window gives you means of rotating a marked object selection 138. This can be accomplished either with the mouse or by directly inputting rotation angles. To rotate the selection with the mouse in a viewport window, move the mouse to the right or top while pressing and holding the left mouse-button. If the mouse is moved to the left or below the rotation is in the opposite direction. The rotation-angle about which you rotate the selection is displayed simultaneously in the "Angle of Rotation" box. There, you can also input the rotation-angles directly. Then operate the <Rotate> button to apply the rotation. The <Clear> button next to <Rotate> sets all angles back to 0.

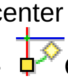
Rotation About the World-Axes



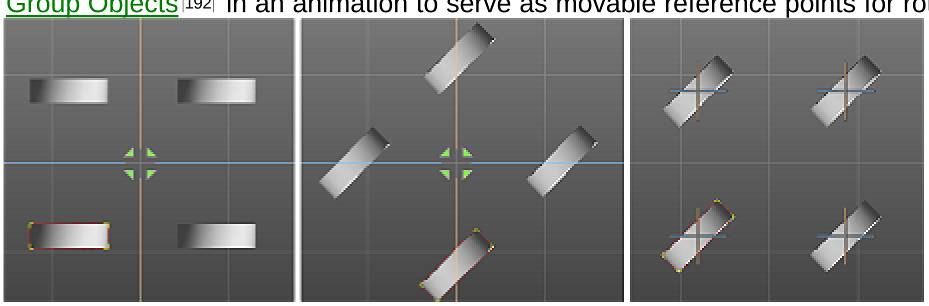
Select the -button in the "Axes of Rotation" box to rotate about the world axes. The world axes are lined up parallel to the X, Y, Z axes of the spatial cube 8. Using the "Mouse Lock" buttons you can again restrict the mouse rotation to certain directions. If the left button is active, objects can be turned about axes that are horizontal and vertical in the viewport. Which axes those are, naturally depends on the view in the viewport. If the button with the horizontal arrow is active, then only rotations about axes that are vertical in the viewport are executed. If the button with the vertical arrow is activated, it follows that the rotation is always about axes that are horizontal in the viewport. If the last button with the circle is selected, then the rotation is always about an axis that points directly out of the viewport window. For example, with the Front view it is the Z-world-axis. Try it out, the principle is quickly recognized.

Reference Point of Rotation

If you rotate about the world axes you have two options for the reference point that defines the center of the rotation:

- Crosshairs (only in Modeling Mode) - If you select this button, then crosshairs appear in the viewport windows. You can move them freely in Modeling Mode only, with the mouse to specify the reference point for the rotation. To "grab" the crosshairs simply click with the mouse into the area between the 4 arrows in the center of the crosshairs. After positioning the crosshairs - if you want to rotate the marked selection again with the mouse - you have to leave the area between the 4 arrows of the crosshairs again.
-  Rotation always takes place about the object axes center of each marked object. In hierarchies 140 the object axes of the top most parent of the marked selection will be used.

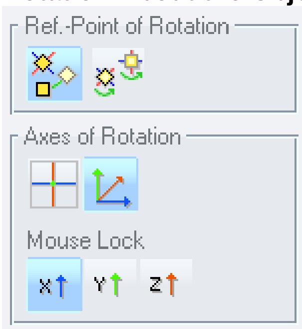
The crosshairs option is not available in Animation Mode. In an animation you always need a traceable reference point and corresponding rotation axes you can refer to in the course of the animation. These reference axes systems are always defined by the object's own object axes, or the object axes of a parent object if movements are inherited in a hierarchy. For instance, you can use Group Objects 192 in an animation to serve as movable reference points for rotations.

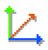


Example: In the left picture you see 4 wheel objects that were selected for a rotation. For the reference point of rotation the crosshairs option is selected and the crosshairs moved into the center of the 4 wheels. Now, if you click into the viewport and move the mouse, the selection will be rotated as a whole about the crosshairs center (picture in the middle).
The illustration on the right shows the result, after rotation about the object's own axes center has

been selected instead. All marked objects rotate simultaneously about their own object axes center.

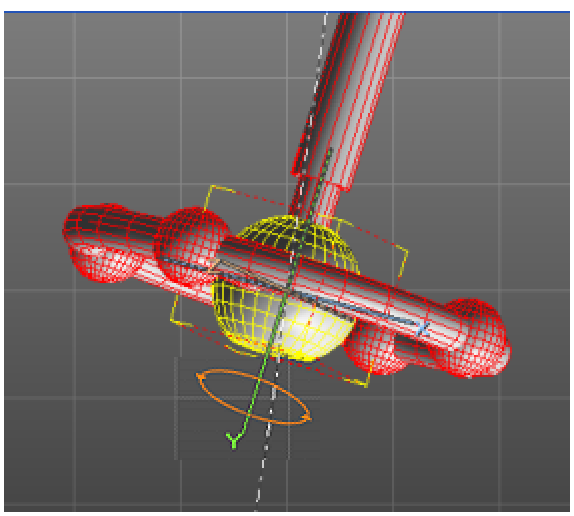
Rotation About the Object-Axes



Select the -button in the "Axes of Rotation" box in order to rotate about the object axes 8 of each object. Then, via the "Mouse Lock" buttons you can choose the X-, Y- or Z-axes to enable a rotation with the mouse about one particular object-axis only.

Reference Point of Rotation

-  Axes Centre of the Reference Object (only in Modeling Mode) - for rotating a selected group of objects about the objects axes center of the marked reference object) This option - like the crosshairs in World Axes mode - is only available in Modeling Mode, because in an animation only the object's own axes system (or the object axes of parent objects that pass their movements to their children) can serve as a traceable reference point for rotations. This connection is essential to maintain the hierarchical independence of each object. If, in an animation, you want to rotate objects about another reference point than the object's own axes or the object axes of parent objects, your objects should be linked under a Group Object 192 and use this Group Object as a movable reference point for your rotations.
-  Rotation always takes place about the object axes center of each marked object. In hierarchies 140 the object axes of the top most parent of the marked selection will be used.



Example: In the illustration you see a model of a space station. All elements of the space station are hierarchically subordinated to the large spherical object in the center. We want to rotate the space station about its longitudinal axis. This would be almost impossible if we had to do this with rotations about the world axes, which are aligned parallel to the spacial cube of the MR-3D Designer world. If, however, you choose the object axes for the "Axes of Rotation" then you simply select the y-object axis of the spherical parent object of the space station hierarchy and the space station will be

rotated automatically about its longitudinal axis.
Rotating about object axes systems is really essential, especially in all situations where the animation of joints is involved - for instance, when animating robots or the skeletons of characters.

Line up object to world-axes

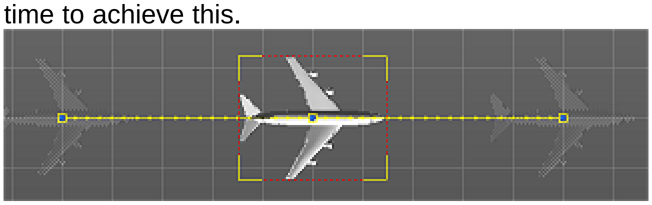
The <Line up to world axes> button rotates all marked objects so that the object-axes line up with the world-axes. This matches the initial orientation the object had when first generated.

Rotation in Hierarchy *(only in Modeling Mode)*

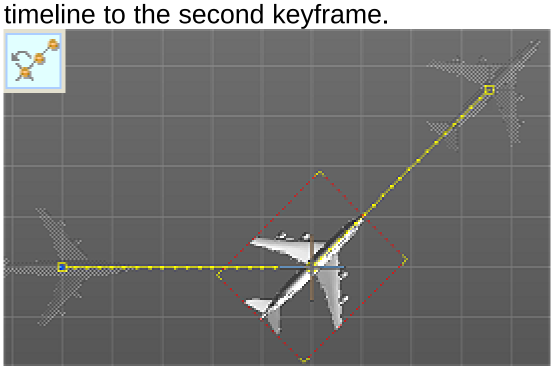
 Usually hierarchical subordinated child objects follow all movements of their parent objects. However, in Modelling Mode you can temporarily switch this off. If you activate the right button of the two buttons depicted above then you can rotate a parent object alone while all children of the parent stay in place. In Animation Mode this function is hidden - all movements of parent objects will be inherited by their children again.

Effect on Following Path *(only in Animation Mode)*

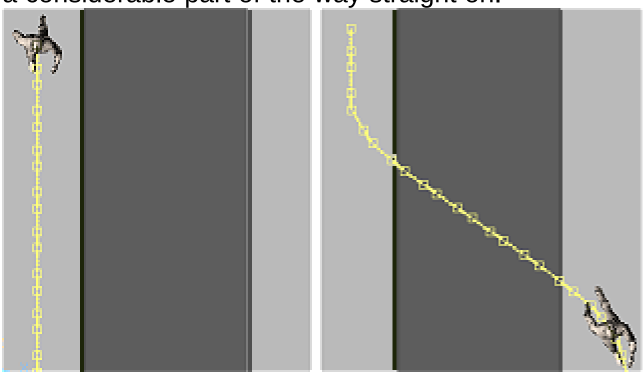
A new rotate key is automatically created every time you rotate an object in Animation Mode, or it is updated if a key already exists on this frame position. If you are moving backwards in the animation because you want to adjust the alignment of an object at a particular frame position, you can determine via the two buttons depicted above, if only the selected object is to be re-aligned or if the whole following movement path is also rotated together with the object. This function is extremely useful every time when you want to transfer animation data from one object to another. Suppose, for instance, you have animated a walking character. You want to create a second copy of this character walking in another direction. To achieve this you only have to copy the character - all animation data will be copied with the model data. Afterwards you simply need to move the copy of the character in "Move Object" mode together with its movement path to a new starting position. Then - here in "Rotate Object" mode - you just rotate the figure, again together with its movement path, so that it faces into a new direction. If you now play a preview animation you can see that the second figure really walks with the copied animation data from the first character from a new starting point in a new direction. You did not even had to step into the complex animation routines a single time to achieve this.



Example: A Boeing has been moved about 3 key positions straight forward. We moved back in the timeline to the second keyframe.

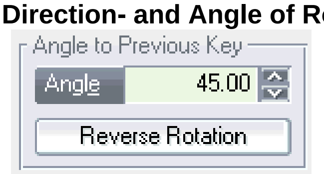


Now we activate the second button "Rotate Movement Path with Selection". Then we rotate the Boeing anti clockwise about 45° together with its movement path. In the illustration you can clearly see how the complete following movement path is bend with the rotation and that the plane is moving now in a new direction. What is demonstrated here with a single plane object can also be applied to whole hierarchies. For instance, to move a character hierarchy that consists of many bones and the skin 242 object - along a complex movement path, you could first animate a complete simple walking cycle of only two steps. You could copy this basic walking cycle in the animation editor 409 several times, one after the other, in Relative Mode 409, to let the character walk a considerable part of the way straight on.



Then, here in the "Rotate Object" menu, you move back to the starting point of the walking sequence. Move forward in animation time in several steps and rotate the character together with its remaining movement path again and again in the desired direction until the movement course has been completed. See the example in the illustration above. An animated character walks straight on along a street. Then, to let the figure walk over the street, the character and its movement path simply has been bent towards the street. On the other side of the street, the remaining path has been bent again so that the character walks again parallel to the street. All in all just a few seconds were spent to guide the figure over the street.

Direction- and Angle of Rotation in an Animation *(only in Animation Mode)*

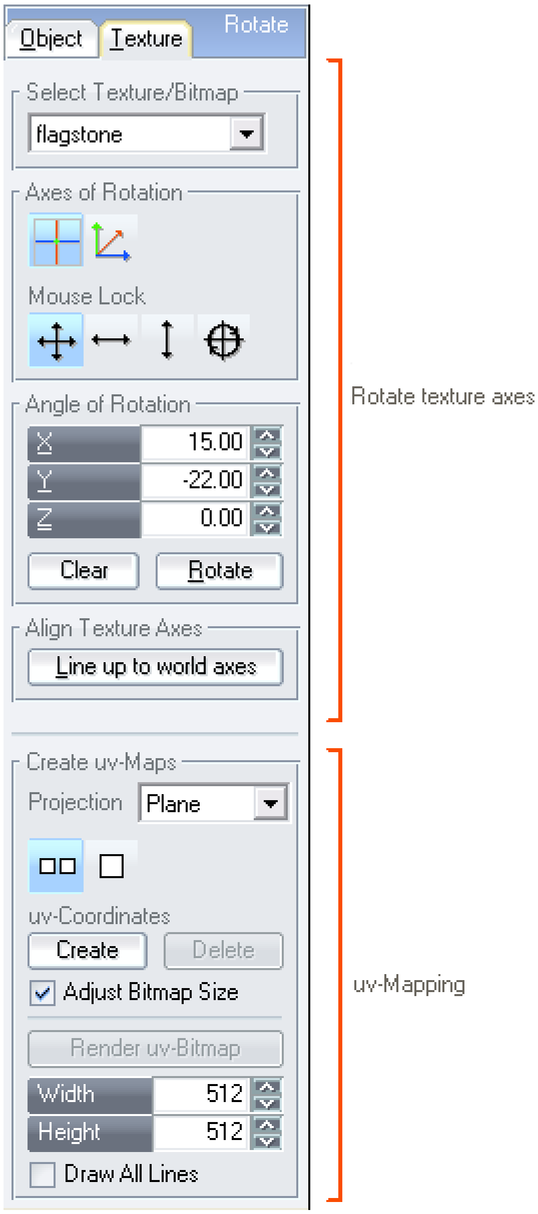


If you rotate an object in Animation Mode a key is automatically generated. This key records a rotation axis about which the object is rotated (so the object axes that were recorded in the previous keyframe will align with the object axes in the current frameposition) and a corresponding rotation angle. There are always two possible directions to perform a rotation, e.g., the pointer of a clock can rotate clockwise or anti clockwise. When a keyframe is generated the shorter angle of the two choices will be recorded. In Animation Mode you can read this angle in the "Angle to Previous Key" box. You can edit this angle, for instance, to reverse the rotation or to let the object rotate a number of times around itself.
Example: You want to animate a propeller so that it turns a hundred times around its longitudinal axis. Select the propeller object and choose the corresponding object axis for the rotation. Then use your mouse to rotate the propeller - you need only rotate it a little bit, so that a key is generated with the rotation axis and an initial angle. Only then, when a key has been already recorded, the "Angle to Previous Key" parameter will be enabled and you can input a new angle. For each revolution you have to add 360° (max. 100000°), so for a hundred revolutions 360° * 100 = 36000° is the correct value.
Note - if you move back in time before this keyframe and you rotate the propeller again, then new keyframes will automatically be calculated, first with the new rotation axis and angle to rotate the object from the previous keyframe orientation to the current orientation, and, simultaneously, a new axis and a corresponding rotation angle is calculated for the directly following keyframe, to rotate the object from the just changed orientation in the new keyframe to the next following keyframe orientation. This simply means that the previously made angle adjustments for the 100 revolutions of the propeller would have to be re-entered for the following keyframe again.

Reverse Rotation

As mentioned before, when calculating the angle that rotates an object from a previous keyframe orientation into the current orientation, the shorter angle of the two possible rotation angles is always chosen. If, for instance, you rotate an object clockwise about 270° and then start a preview you will see that the object will rotate anticlockwise 90° instead. To correct this you can input the reverse angle directly for the "Angle to Previous Key" parameter or just press the <Reverse Rotation> button, to let the program do the work for you. The <Reverse Rotation> function will even convert multiple rotations correctly in the opposite direction.

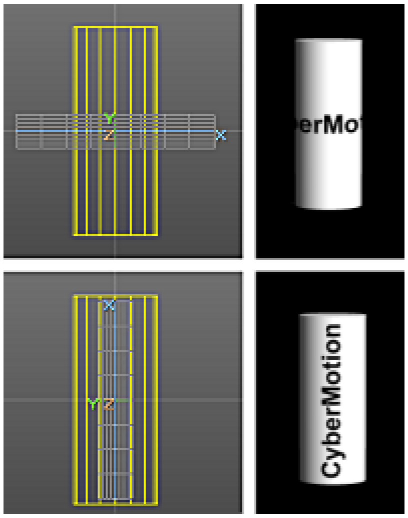
Rotating procedural textures or bitmaps



When you operate the "Rotate"-<Texture> index tab then a set of functions to align a texture axes system appears. In the lower part of the window you will find again the tools for the generation of uv-coordinates and uv-bitmaps. Read more about uv-Mapping here: "Generating uv-Coordinates and Bitmaps for uv-Mapping 218".

Texture Axes Systems and Material Selection
See - Move Textures 197

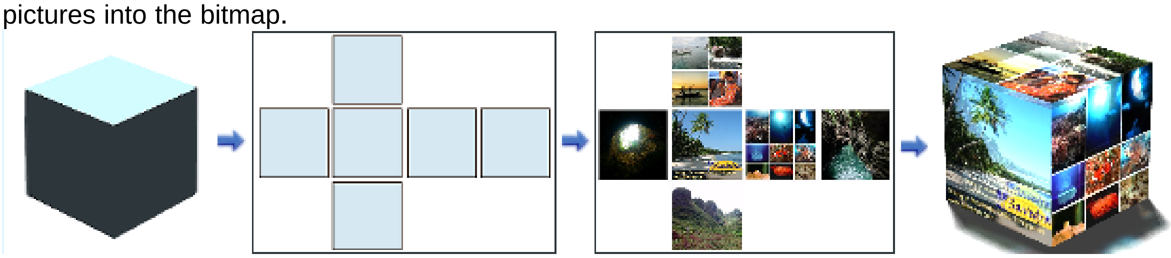
Rotating a Texture
Here in "Rotate Texture" work mode you can adjust the direction of a bitmap projection or the alignment of the three-dimensional space of a volumetric procedural texture by rotating the texture axes.



In the illustration above you can see an example for a bitmap that should be projected vertical along the cylinder. At first the bitmap is aligned horizontally. After rotation through 90 degrees around the Z-bitmap texture axis the bitmap is correctly projected along the cylinder's longitudinal axis.
All buttons and parameters have the same functions as in "Rotate" <Object> work mode. The <Origin> button realigns a rotated axes system along the 3D world-axes 8.

9.5 Generating uv-Coordinates and Bitmaps for uv-Mapping

uv-Mapping offers a very common and easy to operate method to map a picture onto an object. Creating a uv-map for an object is done by projecting and rendering the grid structure of an object directly into a bitmap. Then this bitmap serves as a template to paint in the details of a bitmap texture. For instance, if you create a uv-map for a cylindrical object with a cylindrical projection mode, then a bitmap is rendered in which you find the side walls of the cylinder neatly unrolled to a stripe and the caps of the cylinder separated in a top- and bottom view above and beneath the stripe. Simultaneously, when generating the uv-map, a pair of "uv"-coordinates pointing into the uv-map is assigned to each point of each facet of the object. These picture coordinates are always saved as relative coordinates between 0 and 1, so that they are independent of the picture resolution (which means you can resize the picture without changing the detail projected onto the object's surface). After the uv-map has been generated and saved as an ordinary bitmap from the render window menu, you can load it into common image editors and paint in the details or copy pictures into the bitmap.



Example: With uv-Mapping you can texture all 6 sides of a cube simultaneously applying only a single bitmap. You just need to render a uv-map for the cube with cube projection. The rendered bitmap will show the grid structure of the cube neatly unfolded to the 6 side views. Simultaneously, each vertex of the cube is provided with uv-coordinates pointing to the corresponding pixel coordinates in the rendered picture. Now you can easily copy your texture pictures into the uv-bitmap and then use it as a texture map for a new bitmap material.

More Characteristics of uv-Mapping:

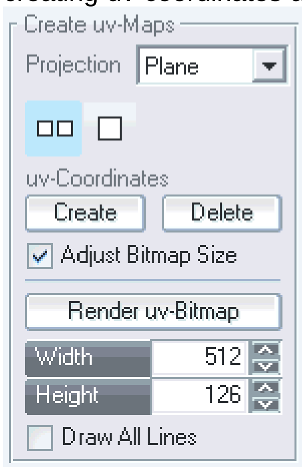
- You can generate uv-coordinates for the whole object and also additional uv-coordinates for each frozen facet selection that provides it's own materials.
- uv-mapped textures will deform with their objects in an animation. This feature is emphasized in most animation programs but nothing special in MR-3D Designer, since even the procedural volumetric textures will deform properly with the object's deformation in an animation. But, in contrast to other texture types, once you have provided an object with a uv-map, you can even deform it in modelling mode. Since all vertices of an object will preserve their uv-coordinates during a deformation the uv-mapped texture will also follow the object's deformation.
- NURBS - Adequate uv-coordinates for NURBS-patches are generated automatically on creation of the NURBS-object. Therefore you can render and save uv-maps for NURBS-patches or frozen patch selections straight away.
- You can't apply uv-maps on analytical objects yet (they don't posses individual points or facets when rendered).

Objects provided with uv-mapped bitmap textures are drawn now with textures in the viewport windows. Use the viewport's menu entry "Display - Texture (uv-Maps)" to activate this display mode. In viewports only the basic material layer (if applied) and the upmost uv-texture is applied, including color masking or alpha mapping.

Generating uv-Coordinates and Rendering uv-Bitmaps

Providing an object with a uv-mapped material is a two pass process. First we have to create uv-coordinates for each vertex of each facet of the object. After that we simply render a uv-bitmap from this uv-coordinates. To create uv-coordinates we first need a texture axes system 197 along which we can decide the projection direction of the object's grid into the uv-bitmap. Therefore we change into the Materials dialog 299. Here we create a new bitmap material. Select uv-Mapping as projection mode for the material. You don't need to choose a picture file for the bitmap material yet (this will later be the uv-bitmap we want to create), so just let the bitmap button switched off . Assign now the new bitmap material to the object. With the material reference automatically a new texture axes system is generated for the object. The projection of the uv-coordinates can be aligned now along these texture axes. For instance, if you apply cylindrical projection then the grid structure of an object will be "unrolled" about the y-texture axis. Nevertheless, if no material has been allocated to the object, you still can create uv-coordinates. In this case the body axes system of the object will be used for calculating the uv-coordinates. Sometimes this makes more sense then using the texture axes of a material. For example, if a cube has been rotated already, the body axes will still be aligned parallel along the side walls. Therefore a cube uv-projection could be processed comfortably along the object's body axes. A newly added texture axes system would have to be aligned first parallel to the side walls before a satisfying cube projection could be applied.

Now change to one of the texture work modes "Move Texture 197", "Scale Texture 206" or "Rotate Texture 211". Beneath the tool boxes to align a texture axes system another tool box for creating uv-coordinates and uv-bitmaps appears.



Next select the object for which you want to create new uv-coordinates. Then choose the corresponding material with its texture axes from the object's material list 197. In the "Create uv-Maps" tool box select now an adequate projection mode from the "Projection" selector box. For our cube example this also would be the "Cube" entry. Beneath the selector box a new choice of possible arrangements for the projection of the different side walls appears.

uv-Coordinates - Create - To generate new uv-coordinates with the selected projection mode you just need to operate the "uv-Coordinates" <Create> button now.

Objects and frozen facet selections provided with uv-coordinates are indicated in the Object Selection 133 window by a "u/v" inserted in the object's type icon.

uv-Coordinates - Delete - When you decide to use a non uv-mapped material you should free the memory used for saving the uv-coordinates in an object.

Up to now you just have created relative picture coordinates in the range from 0 to 1 for each vertex of a facet. To render a corresponding uv-bitmap we now just need to draw the grid structure of each facet using these uv-coordinates into a bitmap.

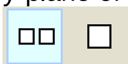
Adjusting the uv-Bitmap Dimensions and Keeping the Proportions
Under the <Render uv-Bitmap> button you can specify the width and height of the uv-bitmap you want to create. Since the uv-coordinates of the object are relative picture coordinates between 0 and 1 the uv-bitmap rendered from these coordinates will be propably distorted, dependent of the bitmap dimensions you have entered. But if you have the <Adjust Bitmap Size> button switched on before generating uv-coordinates for an object, then, in dependence of the width parameter a corresponding height parameter will be calculated and preset into the height parameter field.

Render uv-Bitmap - Once uv-coordinates are saved in the object's data structure you can render anytime a uv-bitmap from these coordinates. Click on the <Render uv-Bitmap> button and a uv-bitmap will be rendered and displayed in the render window. Here you can use the menu functions of the render window to save the bitmap to the hard disk. Any of the picture file formats supported by MR-3D Designer can be used. Now you can load the picture into an image editor to fill in the grid structure with textures or colors of your liking.
Draw All Lines - Each line connecting two points of a facet is drawn into the uv-bitmap. This is

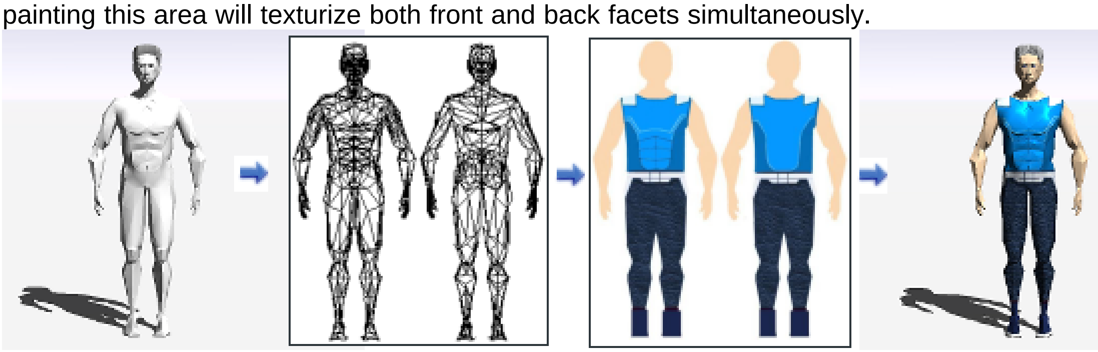
similar to the viewport depiction mode "Grid - All Lines 22".

Projection Types

Plane - The grid structure of an object will be projected in direction of the z-texture axis onto the x-, y-plane of the texture axes system.

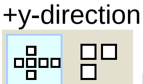


Left Icon - The facets turned towards the z-texture axis (front facets) and the facets turning away from the z-texture axis (back facets) are arranged next to each other in the uv-bitmap
Right Icon - Front- and back facets will be projected on one plane, overlaying each other. So, painting this area will texturize both front and back facets simultaneously.

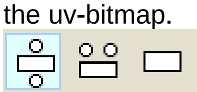


The illustration shows an example for the Plane projection. On the left you see the 3D model of a character. Next to it you see the uv-bitmap created for it applying the Plane projection with separated front and back views. This picture was saved and then loaded into an image editor. There an appropriate texture was painted over the contours of the underlying grid. Finally, after saving the picture again to the bitmap folder of the project, in the Material dialog this picture is selected for the uv-mapped bitmap material assigned to the object.

Cube - Dependent of the orientation (normal direction) of a facet it will be asigned to on one of the 6 possible side walls of a cube. For instance, if the normal of a facet points upwards then the face will be drawn in the rectangular area planned for the top cover of the cube and if the facet's normal points downwards, to the front or back or to the sides it will be drawn in that particular rectangular area arranged in the uv-map for this particular direction. Directions are relative of course, always seen in relation to the alignment of the texture axes system, where upwards means for example the +y-direction of the y-texture axis.

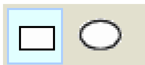
 Projection options:
Left Icon - The 6 sides of a cube are arranged as if you would unfold a cardboard box. In the middle the front side, above and under it the top and bottom covers, the side walls to the left and to the right and the last rectangle in the row is the back wall projection of the cube.
Right Icon - The opposing sides of the cube are projected onto the same area, overlaying each other. Painting one cell will texture both sides of the cube.

Cylinder - The texture axes have to be centered within the cylindrical object and the y-texture axis has to be aligned along the longitudinal axis of the object. Then the uv-coordinates of the object will be projected as a unrolled stripe with the caps of the cylinder separated in a top- and bottom view in the uv-bitmap.

 Projection options:
Left Icon - The top- and bottom caps are arranged above and unter the unrolled side wall of the cylinder.
Middle Icon - The caps of the cylinder are arranged next to each other above the side walls of the

cylinder.
Right Icon - All facets are projected to the unrolled side wall.

Sphere - First center the texture axes in the center of the object. Then, in dependency of the angle formed by the vektor from the object center to the facet vertex and the texture axes alignment the uv-coordinates for the vertex is calculated.

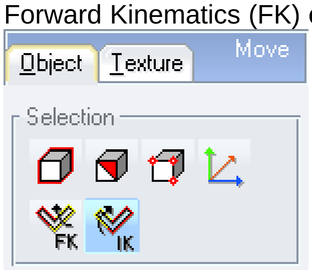


Left Icon - uv-coordinates of the spherical projection are mapped onto a rectangular area. With it the typical distortion arises you know already from world maps. At the equatorial region the width of the picture corresponds exactly to the circumference of the globe. From the equator up- and downwards the circumference of the globe decreases towards the poles. However the map is adjusted to fill the whole rectangular area so that when approaching the poles the top- and buttom points of the poles are stretched to a full line in the map. On the contrary that means that a whole line of your bitmap texture will be merged into a single pixel at the poles of a sphere. There are image editors on the market which provide special functions to pre-distort bitmap textures so that the distortion dissolves again when applying the spherical projection.
Right Icon - With this projection type the decreasing circumference of the globe towards the poles is also reproduced in the uv-map. The result is an elliptical projection without distortion.

9.6 Inverse- and Forward Kinematics

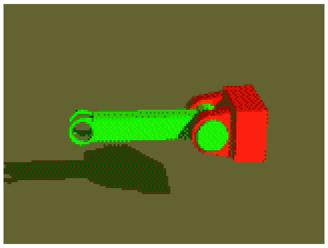
If you have already studied our tutorial concerning the assembly and animation of a little robot model [62] then you actually know everything regarding Forward Kinematics - a hierarchy [140] of objects with their object-axes serving as joints and points of rotation. Rotating an object about its object-axes will cause all hierarchical subordinated objects to follow the rotation. This is called Forward Kinematics. Inverse Kinematics can simplify the positioning of several joints simultaneously even more. You need only to pull on a finger, for example, to stretch the complete arm of a 3D-model - similar to a real jointed puppet. Furthermore you can restrict the rotation angles for each joint by setting limits for each individual rotation axes (Degrees of Freedom), thus preventing the model from unnatural twisted movements.

Forward Kinematics (FK) or Inverse Kinematics (IK) is applied only in "Move Object [197]" work mode.

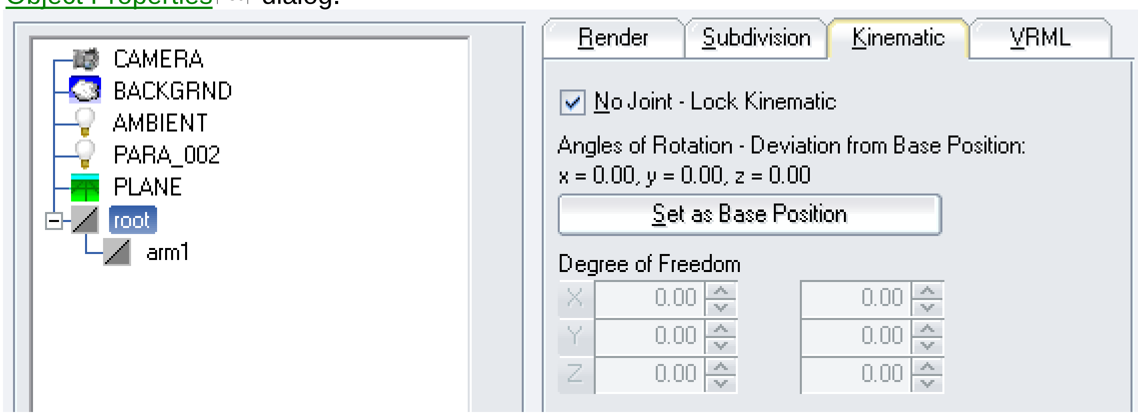


In the tool window, beneath the 3 selection buttons for selecting objects facets or individual points, there are 2 additional buttons for the  Forward Kinematics and the  Inverse Kinematics working modes. You can change anytime from the normal displacement work mode to the Kinematic work modes - for example move an object independently from its parent objects and right after that change into IK mode to pull at this object and, with it, all parent objects.

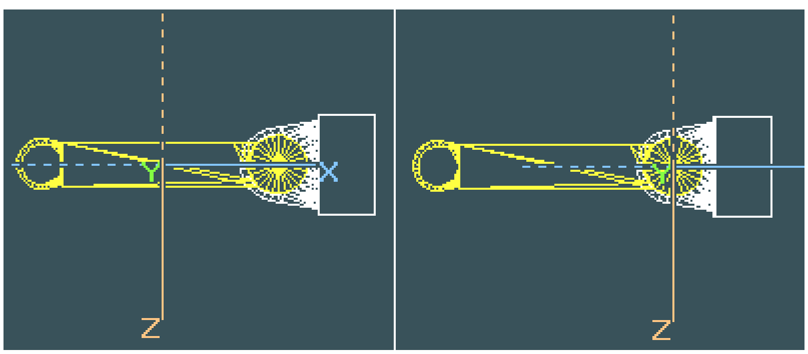
In practice: (You can find corresponding sample files to all examples in the folder "\projects\ik\".)



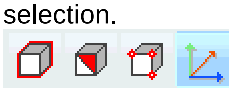
First load a simple hinge object "root" with a lever object "arm_1" attached to it (project file "ik_1_joint.cmo"). The hinge serves as root object for the hierarchy and is not movable itself. To prevent the hinge from taking part in the rotation movements of its subordinated joints we call up the Object Properties 289 dialog.



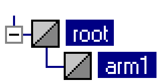
Select the "root" object and change to the <Kinematic> page of the dialog. Switch on now the <No Joint - Lock Kinematic> button. This results in a fixed hinge that will not rotate, when pulling at subordinated objects belonging to it. Now leave again the dialog.



Next we select object "arm1". The rotations of the objects are always executed about a defined pivot point that is identical to the focus of the object-axes. So first we have to move the object-axes of the "arm1" object to the joint position about which the individual rotations are later executed (shown in the right illustration above). Select therefore in "Move Object 197" work mode the object axes selection.

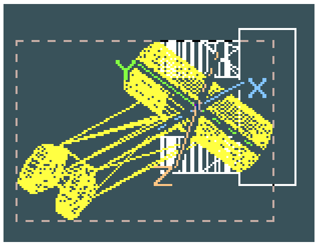


Now you can easily move the axes system with the mouse to the desired joint position.

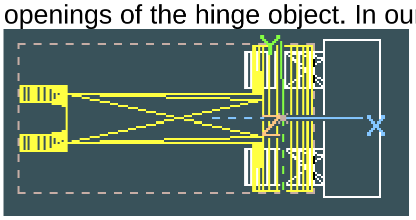


The lever object "arm1" still has to be hierarchally subordinated to the hinge object "root". In the Object Selection 133 window click with the left mouse button on "arm1". A box containing the name of the object appears while the button is held down. Drag the box over the "root" object until a tool-tip stating "Link" appears. Release the button and "arm1" is now displayed to the right side of the "root" in the window and is subordinate to that object.

Now activate the Inverse Kinematics work mode by selecting the  button in the tool window.

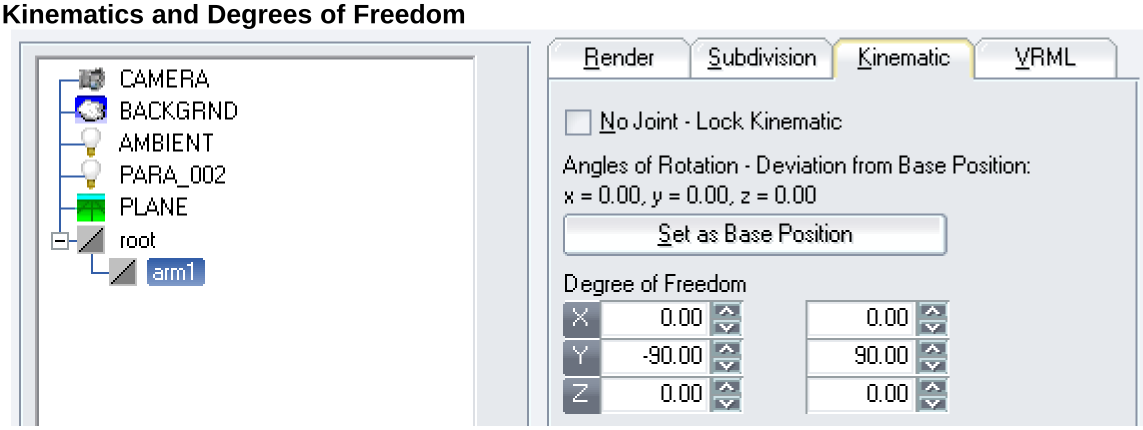

(Top view)

If you now select "arm1" and pull at the object, it really tries to follow the mouse movement by rotating about its object-axes. However, it rotates about all 3 object-axes and therefore it breaks out off the hinge. Consequently we have to restrict the rotation to the axis passing straight through the openings of the hinge object. In our example this is the y-axis of the "arm1" object.



Use the <Undo 28> function to return to the initial position of "arm1". Then call up the Object Properties 289 dialog again.

Kinematics and Degrees of Freedom



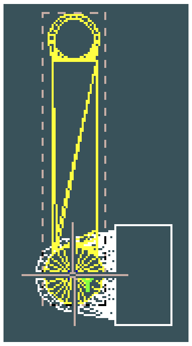
On the <Kinematic> page of the dialog use the "Degree of Freedom" (DOF) parameters to set upper and lower limits within which the object is allowed to rotate. These limiting angles are always based on the initial position of the object when it was linked to a hierarchy, so you can first position and align an object, then link it to the hierarchy and finally insert suitable DOFs based on the initial position.

However, you can also do it the other way round, first linking an object to a hierarchy, then moving and rotating it afterwards to a suitable position, then you can set this position as initial position just by operating the <Set as Base Position> button. The settings for the DOFs then apply to this base position.

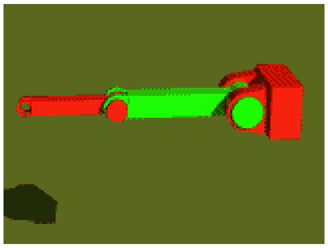
When an object has been rotated using the Kinematics functions the angles an object has been rotated through from the base position can be seen beneath "Angles of Rotation - Deviation from Base Position". If you operate the <Set as Base Position> button they will be set to zero again. However, if an object has been already animated over several key frames, then you can no longer change the initial base position.

You can still rotate objects regardless of the DOFs assigned to them - just execute a simple rotation about world or object axes in the "Rotate Object 211" work mode. These rotations are wholly independent of the Kinematic settings - neither the "Angles of Rotation - Deviation from Base Position" nor the DOF parameters are affected by it.

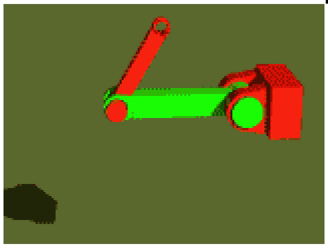
Let's go back to the hinge and the lever object "arm1". The initial DOF values are set to ±180° for each object, which is equivalent to an unrestricted full rotation. We want to lock rotation about the x- and the z-axis for "arm1", so we just enter ±0° for the DOFs along the x- and z-axis. But we also want to restrict the rotation about the y-axis somehow, so that the lever is moved only until he hits the base of the hinge and does not go through it. For that purpose we set the DOFs for the y-axis from -90° to +90°.



After leaving the Properties dialog you can immediately observe a change in the drawing of the object-axes. Locked axes are differentiated from the unlocked axes in a grayed style instead of the usual axes colors. Now pull again at the level object "arm1". It now rotates only about the y-axis and the movement stops in the vertical position - at the top and bottom respectively.



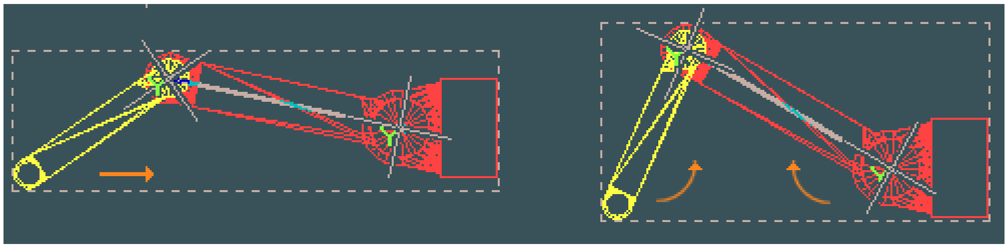
Now let's increase the complexity of the model by adding another joint object (file "ik_2_joints.cmo").



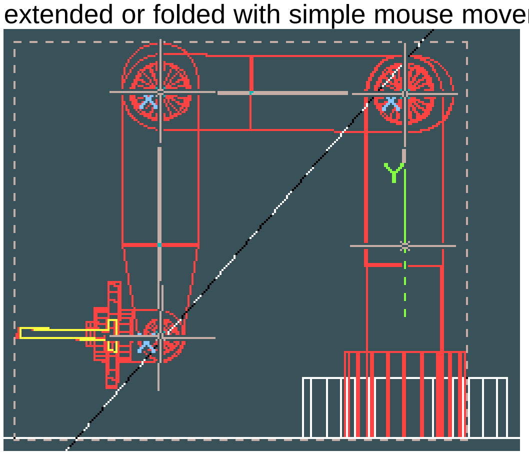
Select the new object "arm2" and call up the Object Properties dialog again. Lock the x- and z-axis again. This time, for the DOFs we enter a somewhat higher range of ±120°, so that the lever "arm2" can move exactly on both sides until it hits the base of the lever "arm1".

Forward Kinematics versus Inverse Kinematics

Let's focus again on the difference between the two Kinematic modes. With Forward Kinematics you select an object in a hierarchy of joints. When you move your mouse in a certain direction the selected object tries to follow the movement by rotating around its object-axes. All objects subordinated to this object will execute exactly the same rotation, similar to the rotation of an object group about the object-axes of a reference object.



The idea behind Inverse Kinematics is a different one. You mark an object at the end of a chain of jointed objects as reference object, in our example "arm2". Then (with a second mouse click while simultaneously holding the <Ctrl>-key pressed) you select the top-most object in the hierarchy that you want to include in the movement - here "arm1". The object-axes of all marked objects - except those which had been locked for Kinematics - are drawn. Additionally a thick line between the individual joints is drawn to emphasize the connection of all the joints that are involved. If you now give the direction of movement with your mouse, "arm2" again tries to follow the movement, but this time all marked joints participate in the movement. This way the two levers of our model can be extended or folded with simple mouse movements in the respective direction.



Finally again our example from the robot tutorial (file "projects\ik\ik_robot.cmo) with all DOFs already entered. If you select the tongs object at the head of the robot and pull at it, then a movement will be calculated that involves rotations about 3 hinge joints and 1 rotational joint at the base of the robot at the same time. The pair of tongs is locked for Kinematics and therefore will not, itself, be rotated, but it has unlocked parent objects serving as joints and the movement is passed on to them. This is always import when you subordinate objects to a parent object serving as a joint but don't want the subordinated object to be part of the chain of joints, e.g. if the added object is only an addition to the construction of the parent object.

9.7 Edit Objects

 - Menu "Edit - Edit Object"

Simple changes in the appearance of objects can be made by selecting individual facets or points and moving, scaling or rotating them. The functions in the Edit Object toolbox give you the ability to go much further in your object manipulation and modeling. Most functions are applied to selected facets or points of marked objects 138. Consequently you automatically change to Modelling Mode when entering this menu.

Functions:

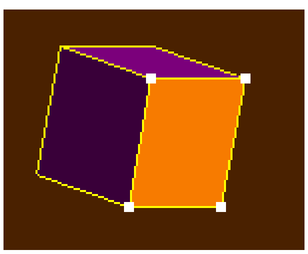
- Facet Extrude 228
- Add Points and New Facets 230
- Magnetic Deformation 231
- Delete Points, Facets and Duplicate Elements 232
- Boolean Operation 233
- Triangulate Facets 237

9.7.1 Facet Extrude

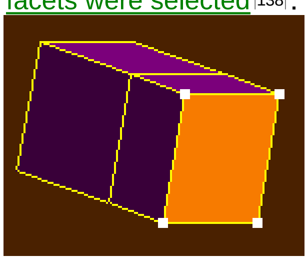
Menu: Edit Object 227 - Facet Extrude

Select the <Extrude> tab button at the top of the tool window to activate the "Facet Extrude" function.

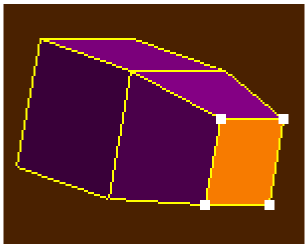
By using the "Facet Extrude" function you can conveniently add new facets to objects just by selecting facets on an object and dragging them out or into the object. This operation is carried out in real time with the mouse directly in the viewport window or alternatively manually through the parameters and the <Extrude> button in the "Facet Extrude" box. Let's see an example:



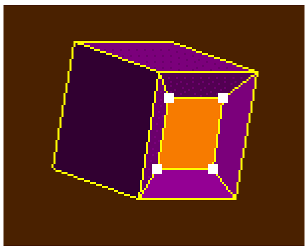
This simple box was generated from the primitives menu 146 and then both front facets were selected 138.



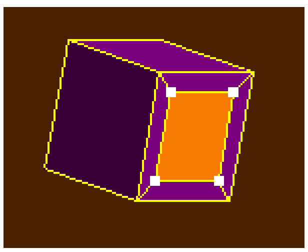
Activate the "Facet Extrude" tab. Move the mouse over a viewport and click with the left mouse button in the window. Hold the mouse button pressed and move the mouse a little bit to the right. This automatically generates an additional surface segment that is dragged out of the object with the movement of the mouse. The "Depth" parameter in the "Facet Extrude" box shows the distance of the new segment to the original surface.



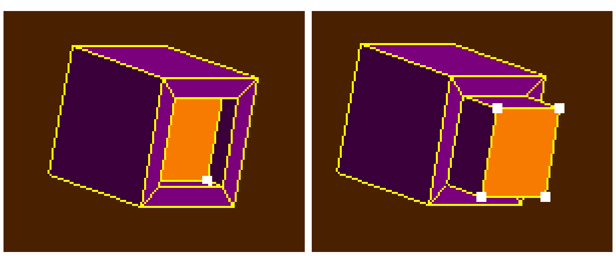
Setting the "Bevel" parameter to other than zero will result in beveled sides of the new generated segment. The picture shows the box with a bevel value set to 20 before applying the Extrude function.



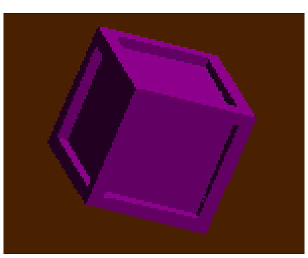
This picture shows the same process, except that the mouse was moved to the left and therefore the new beveled facet segment was pushed into the object instead out of it.



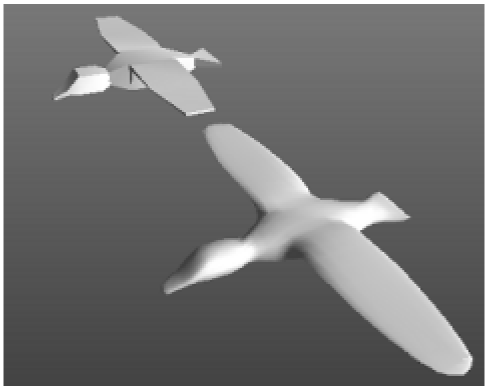
This time a "Facet Extrude" operation was performed directly on the surface, with beveling but without displacement in the depth. To achieve this you have to set the "Depth" value to zero and the "Bevel" value to something other than zero. After that, simply press the "Extrude" button. New facets are generated directly within the old ones and you can see that this time the "Bevel" value determines how far the new facets are displaced within the old ones.



After that you can do a normal "Facet Extrude" operation again, to drag new segments in or out of the object.



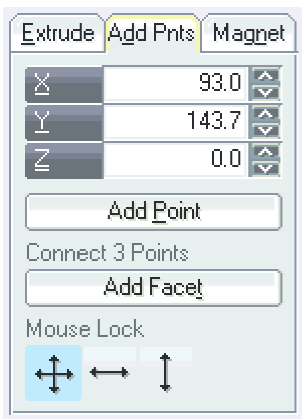
This is how the box appears, if you repeat the procedure for all 6 sidewalls.



This flying duck is an example of an object constructed from a simple box using facet extrusion and finally the object smooth function. Another example, the modelling of a plane from a simple cube can be retraced in our tutorial: Facet Extrude; in a few minutes from a box to a plane model [51]

9.7.2 Add Points and New Facets

Menu: Edit Object [227] - Add points and new facets



Add Points

Generating new points and later connecting them to form facets can extend individual object structures. Completely new constructions are also possible. You could, for example, draw a simple triangle in the extrude-editor [149] and create an extruded flat triangle object. Then add new points to the object relating to your construction requirements and connect them to form new surface facets. To generate new points for an object you must first change to the corresponding work mode by selecting the <Add Pnts> tab in the tool window. Then mark the object [138] you want to edit and activate point selection [138] by pressing the  button.

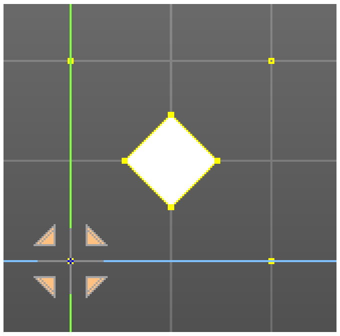
Large crosshairs appear in the viewport window. Click within the 4 arrow buttons at the center of the crosshairs to grab it, then hold the left mouse button pressed and drag the crosshairs into the position where you want to generate the new point. Use the "Mouse Lock" and the "Snap [197]" functions to assist precise positioning. The x-, y- and z-coordinates of the new point can, however, also be input directly through the keyboard. To create the new point, just operate the <Add Point> button.

Individual, unconnected points that are not components of a surface are only visibly and highlighted if the Point [22] - entry under "View [22]" in the menu bar has previously been switched on. Check on this if the newly generated points are not seen on the screen.

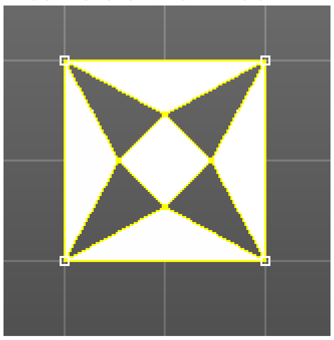
Connect Points

After you have produced new points, you will want to connect them to form new triangular facets.

Simply mark three points 138 and then operate the <Add Facet> button to create a new triangular facet from them.



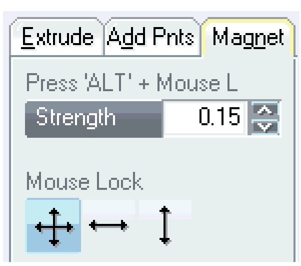
In the foregoing example you see a basic square object. A point has been generated in each of the 4 corners of the window with the "Add Point" function.



Four new facets are generated with the help of the "Add Facet" function. In each instance one corner point of the inside quadrilateral has been connected with two of the newly created outer corner points.

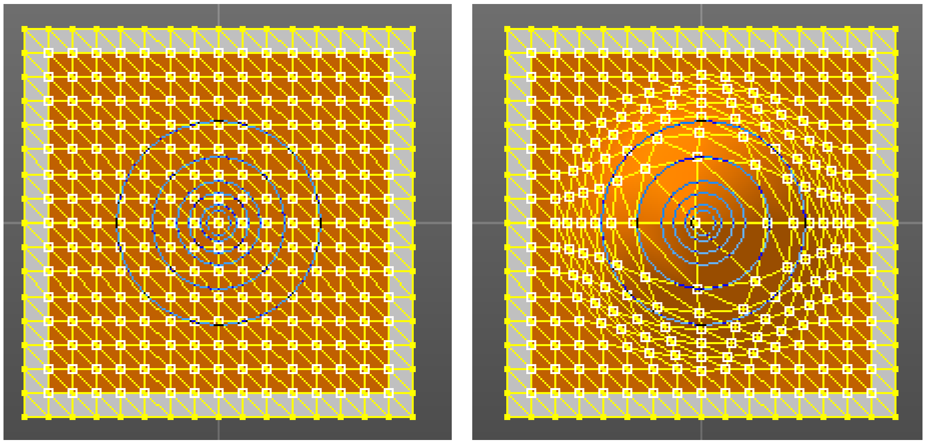
9.7.3 Magnetic Deformation

Menu: Edit Object 227 - Magnetic deformation



Objects can be distorted with the help of the magnet function. To apply this function you must first change to the **<Magnet>** work mode by selecting the corresponding tab in the tool window, then mark the object 138 you want to edit and activate point selection 138 by pressing the button. The function works on selected points only, so first mark all points 138 you want to be influenced. An annular object appears on the viewport window, which you position with the mouse while holding down the left mouse button. The different rings of the object give the area of influence of the magnetic field. The narrower the rings are, the stronger is the magnetic field. The further apart the rings, the weaker the magnetic field. For points that lie within the outer ring the distortion is only very weak, while near the center it is very strong. The strength relating to the extent of the magnetic field can be adjusted through the "Strength" parameter between values of -1 and +1. The higher the value is, the stronger the magnetic field. Negative values have a magnetic attraction, positive values, consequently, a magnetic repulsion.

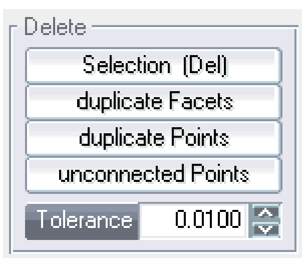
Activating the Magnet - <Alt> button + left mouse button
To produce magnetic distortion, position the magnet-object at the target point then simultaneously press the <Alt> button and the left mouse button. You can also, while pressing the <Alt> button, move the mouse, so that the influenced points also continue to move.



The illustration shows a flat frame object. All points except those lying along the edges have been selected for treatment. The magnet is switched on and positioned over the middle of the frame. In the illustration on the right you see how the object appears after operating the <Alt> key and left mouse button.

9.7.4 Delete Individual Facets or Points

Menu: Edit Object 227 - Delete Selection, duplicate Elements or unconnected Points



Delete <Selection>
All marked objects, point- or facet selections or materials are deleted. This function can also be applied by pressing the "Del" key on your keyboard. There is a simple logic in deleting different types of objects or selections: If a frozen selection 138 is marked in the Object Selection 133 window then first this frozen selection entry will be removed from the object (points and facets of the selection remain marked). If you press the "Del" key a second time then the marked points and facets will be removed from the object. Finally, if you press the "Del" key another time, the whole object will be deleted.

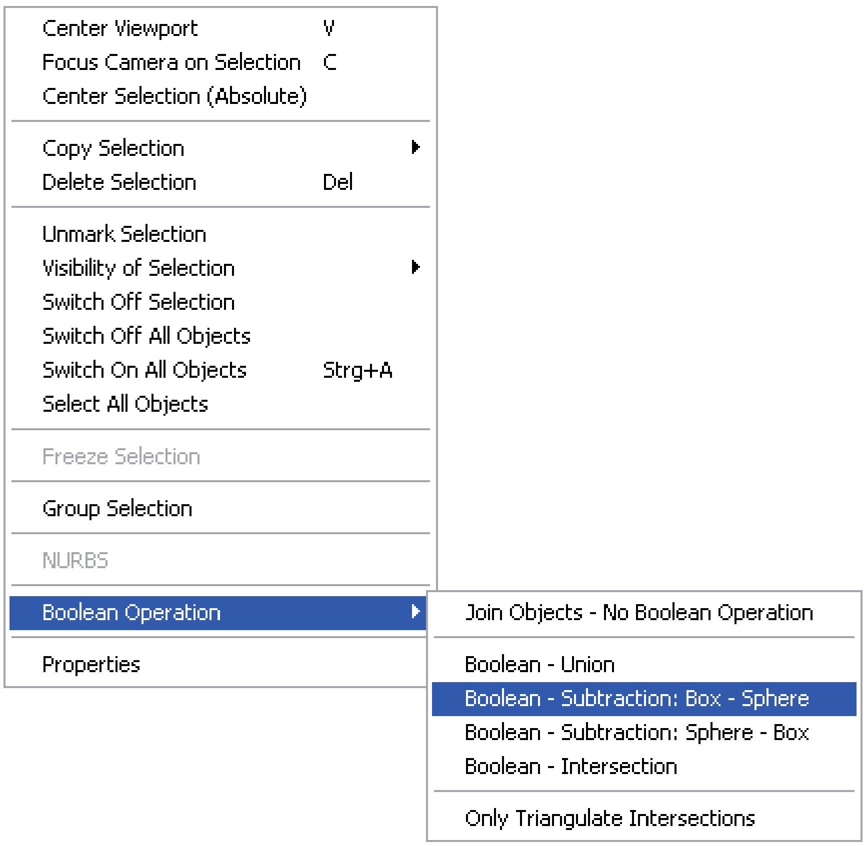
Delete <duplicate Facets> or <duplicate Points> - You can optimize a model by deleting duplicated points or facets. To decide whether a point is superfluous or not you can define a tolerance value. If two or more points lie within this tolerance distance they are replaced by a single common point. However, sometimes duplicate points were inserted deliberately to achieve certain effects. For instance, some foreign file formats provide uv-coordinates only for the vertex knots of a mesh (not separately for each face its own uv-coordinates, as does MR-3D Designer). To be able to provide these models with different materials, additional points are often inserted at the seams of regions where a separate material is required. If you delete these duplicate points you will also delete the texture information of these imported models. It is also possible that points were duplicated deliberately to detach a face selection 239 from the object mesh or to create hard edges in Subdivision Surfaces 163.

Delete <unconnected Points>

Through erasing points and facets, points sometimes remain that do not form a corner of any the remaining facets. Also, you may sometimes generate superfluous unconnected points with the <Add Point> function. These can all be removed with the function "Delete" **<unconnected Points>**.

9.7.5 Boolean Operation

Menu: Edit Object 227 - Boolean Operation

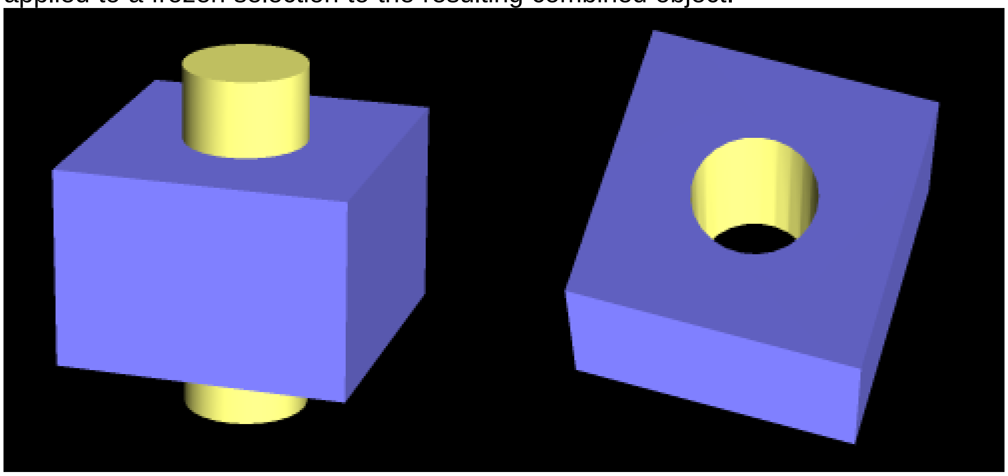


If you choose the <Boolean Operation> button in the "Edit Object" tool-window, a popup selection opens with several entries for joining objects in different ways. Those entries are found also in the popup selection 133, which opens when clicking with the right mouse button in a viewport window (see picture above) and again in the "Object Selection" dialog, thus enabling access to these functions in every work mode.

A Boolean Operation (named after the English mathematician George Boole) describes the connection of 2 objects with simple logical operators like "AND" or "OR". MR-3D Designer makes use of this logic when joining two overlapping, facet based objects, which results in complex structures that otherwise would be very difficult to construct. For instance, it is very easy to cut a hole in an object just by placing a cylinder at the desired position for the hole and then operate a Boolean Subtraction "object" minus "cylinder".

To apply a Boolean Operation, first mark 138 the two overlapping, facet-based objects. Analytical objects 147 and NURBS 171 cannot be joined, but you can convert NURBS to facet-based objects and after that a Boolean Operation will be possible. When a Boolean Subtraction is applied, the resulting object will adopt the name, material and animation data of the object from which you subtract the other object. Otherwise the data is adopted from the selected reference object.

Frozen Point or Facet Selections 138 are also incorporated in the Boolean Operation. They will be automatically adjusted and transfered to the resulting object, if not all parts of the frozen selection are removed by the combination of the two objects. This way you can even transfer the materials applied to a frozen selection to the resulting combined object.



Example: A yellow cylinder is used to cut a hole in a blue box. Normally, the resulting object would have adopted the material of the blue box. But in the example above the side walls of the hole kept the yellow material of the cylinder. This is because we assigned all facets of the cylinder previously to a frozen facet selection. Then we allocated the yellow material to the frozen selection.

When subtracting now the cylinder from the box, the remaining facets of the frozen facet selection will be transferred together with the yellow material to the box resulting in the yellow hole of the box.

Boolean Operators:

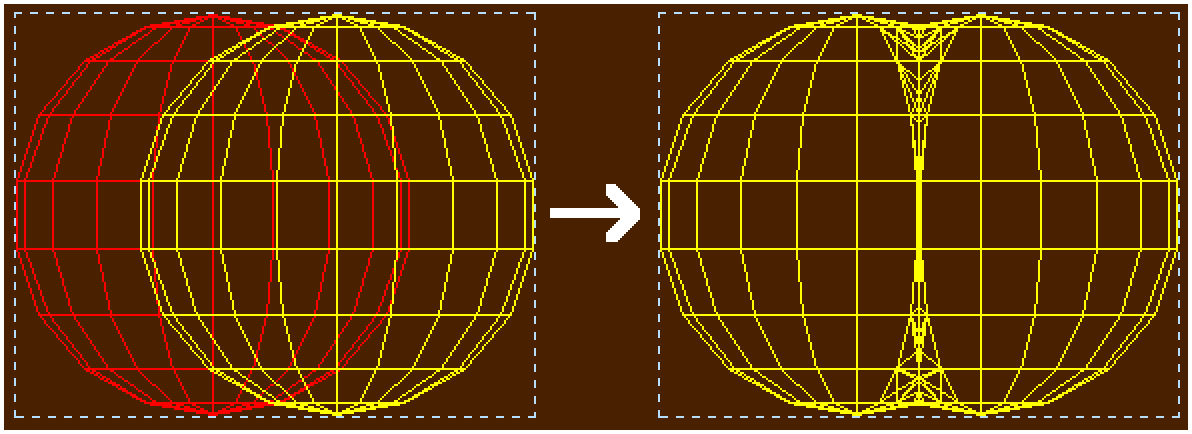
- Join Objects - No Boolean Operation 233
- Boolean Union 233
- Boolean Subtraction 233
- Boolean Intersection 233
- Problems using Boolean Operations 233
- Only Triangulate Intersection 233

Join Objects - No Boolean Operation

This function merely serves to manage 2 objects as one object. It is often simpler to manipulate objects that are composed of many different parts (but of the same material) as a single object. The object construction itself remains unchanged. No Boolean Operation is applied - just 2 objects being referenced as one single object.

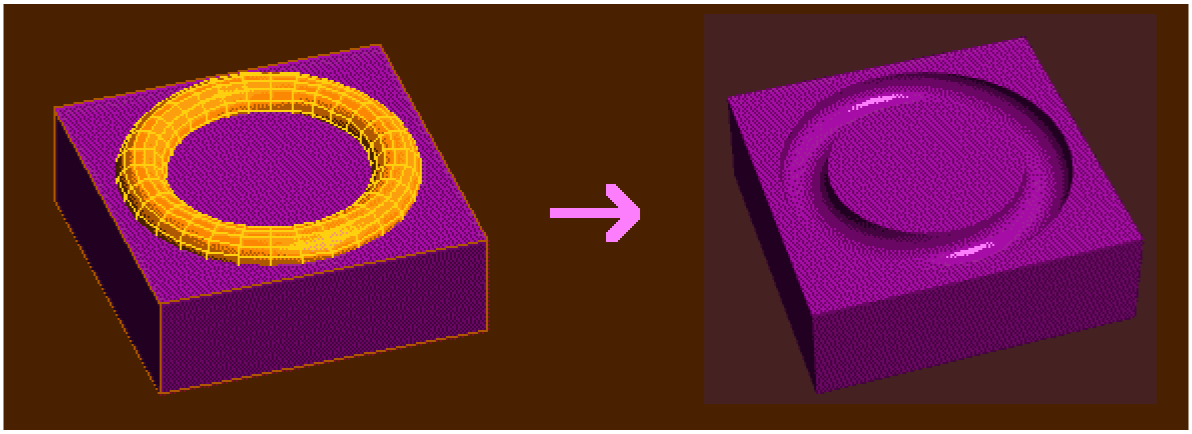
Boolean Union

The Boolean Union function combines 2 objects by removing all the facets and points of each object that lie completely within the other object. In the process, the overlapping facets will be subdivided along their intersecting edges to create a seamless transition along the overlap of objects.

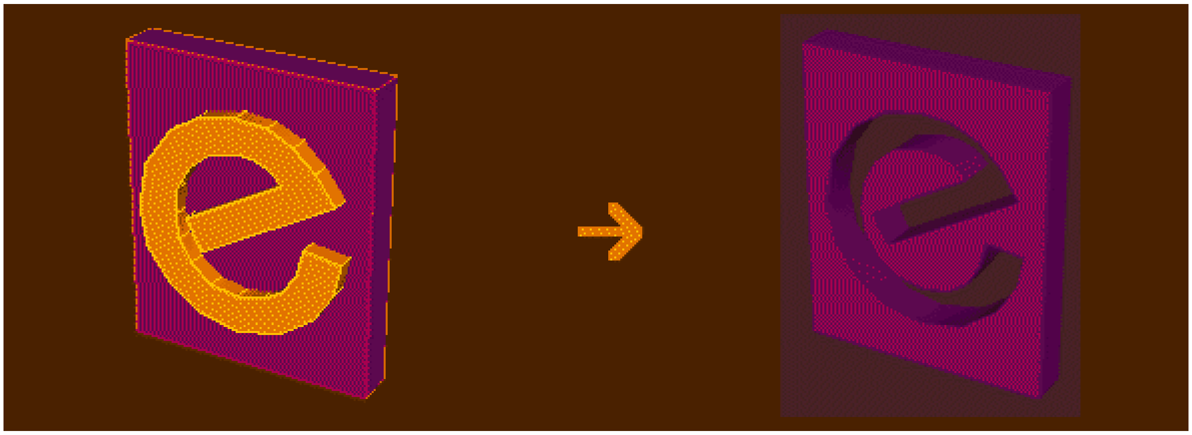


This is best shown in wire frame mode. Two overlapping spheres are merged with a Boolean Union operation. The superfluous inner parts were eliminated and the additionally generated facets and points for the joining edges are clearly visible.

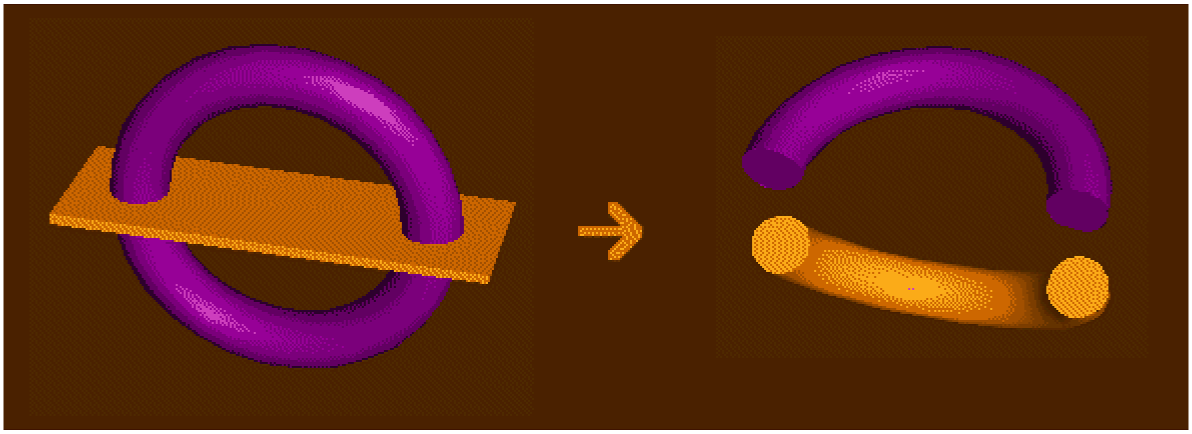
Boolean Subtraction



The Boolean Subtraction uses one object as a tool to cut a shape from the other object. Above you see an example for an operation "Box" minus "Torus".

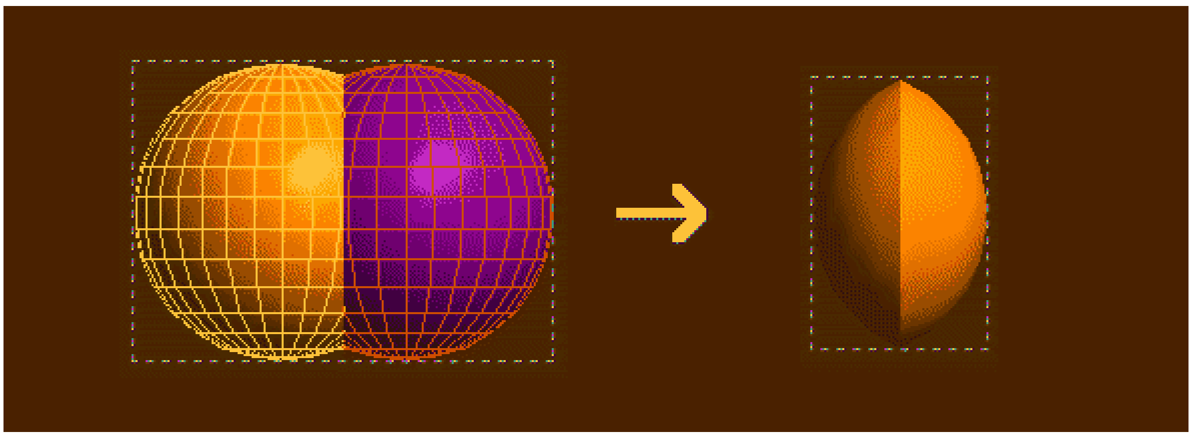


This operation is not restricted to simple primitive shapes, even complex structures can be combined. The picture shows a Boolean Subtraction of a 3D text object 174 "e" from a "box". However, because of the intensive calculation process, you should keep objects as simple as possible. If, for example, you want to subtract a complete text string of high resolution 3D text from a box, I would recommend to do this character by character and afterwards join the separate objects.



Another example shows how Boolean Subtraction can be used to produce cross-sections of objects. In the picture you can see a primitive object "box" that was subtracted from a "Torus", so that the "Torus" was cut into two parts. The points of one half of the "Torus" were then selected and transformed into an independent object by means of the Detach Object 239 function.

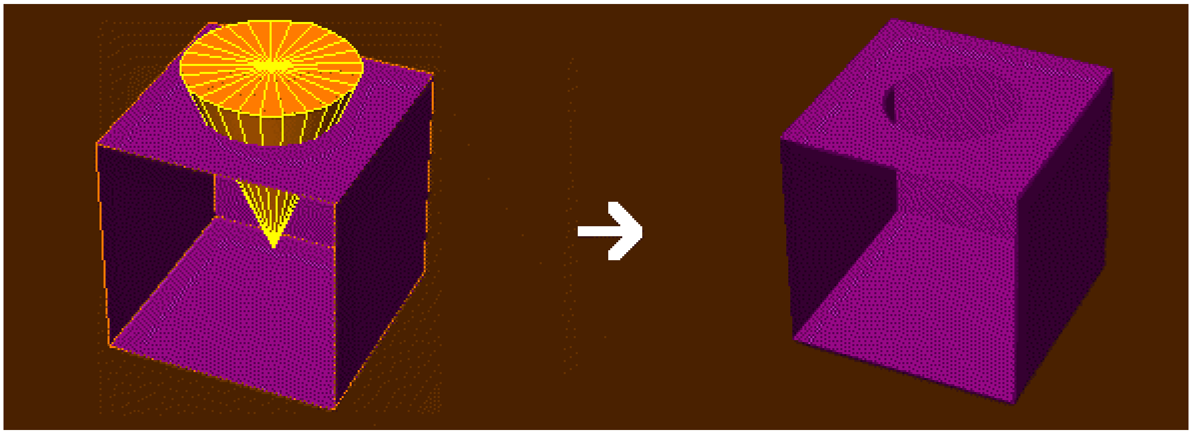
Boolean Intersection



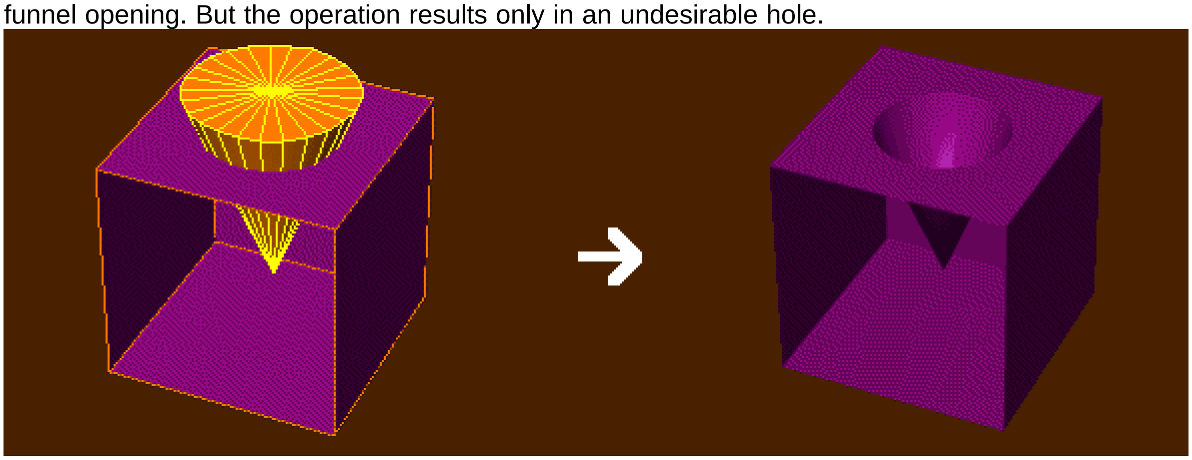
A Boolean Intersection creates a new object from the overlapping parts of the two objects.

Problems using Boolean Operations

When calculating a Boolean Operation there is one basic problem - to determine which facets or points of an object lie inside or outside the other object.
With primitive convex 290 objects this is very easy, but with non-convex objects calculations can go to great lengths.
Although Boolean Operations are intended to be for objects with closed surfaces - since only then you can determine exactly those parts of an object that lie inside the other object - you can also apply Boolean Operations to open objects. One criterion, whether a point lies inside or outside, is to check if the point is located underneath a surface facet of the other object. This can be calculated from the facet's normal 8 (a vector standing vertical to the surface). With closed objects this surface normals always point to the outside, but with open objects it depends on the view of the user, what is intended to be inside or outside.
If the result of a Boolean Operation with an open object is unsatisfying, you can try to repeat the operation after inverting the normals of the object.



The picture shows a box with an open front. A cone is to be subtracted from the box to create a funnel opening. But the operation results only in an undesirable hole.



This shows the same situation except that we have inverted the normals of the box before applying the Boolean Subtraction. Now it worked out well and the funnel was inset in the box.

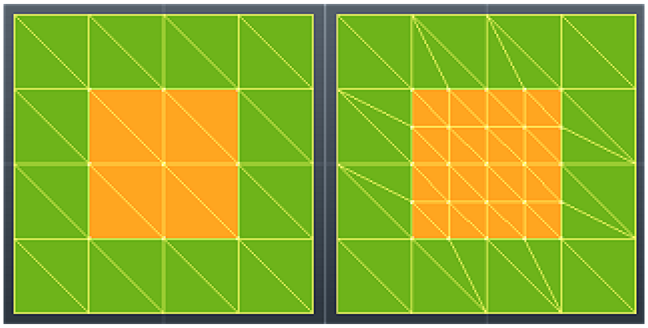
Only Triangulate Intersections

This function is the preliminary stage for all Boolean Operations. The process subdivides the overlapping facets along their intersecting edges to create a seamless transition along the overlapping objects. But after applying this function no Boolean Operation is executed, thus enabling the user to select by hand the points and facets he wants to delete. Afterwards the objects can be joined again using the "Join Objects" function.

9.7.6 Triangulate Selection

Menu: Edit Object 227 - Triangulate Selection

All marked faces of an object are broken into 4 smaller triangular facets by triangulation so that you can refine the structure of an object to improve its form or to work on it in greater detail.



On the left of the illustration you see a frame object with selected facets and next to it the same object after the function <Triangulate Facets> has been applied. You will see that the facets surrounding the marked selection were also split into two new facets - so that the newly inserted points at the rim of the selection are now connected to all adjacent facets, too. This way, the homogenous structure of the mesh is maintained, which is especially important if you want to refine the mesh later with a subdivision algorithm. Without the homogenous mesh structure the mesh would tear open at the unconnected edges when applying the Smooth Object 238 function or when changing it into a Subdivision Surface 163.

See also: Subdivision Surface - Homogenous Mesh 163 and Repair Mesh Structure 240

9.7.7 Smooth Surface

Menu: Edit Object 227 - Smooth Surface

This function smoothes coarsely-structured surfaces of facet-based objects by transforming them to a higher resolution. The facets are subdivided and new points are interpolated to round off the surface. To apply this function, simply mark the relevant objects and select the "Smooth Surface" button.

This function uses the same algorithm that is used for the realtime Subdivision Surface deformation. In contrast to the SDS-technique, a permanent change of the model data is performed, which cannot be removed later just by switching off the SDS-object property. On the other hand, you can apply the Smooth Surface function to individual facet selections of a model, so that particular regions of an object can be refined.

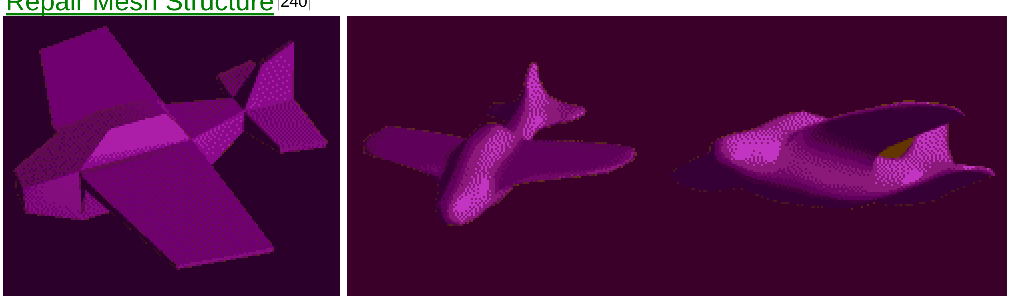
There are two different options for refinement:

Keep Hard Edges - The boundary edges of non-solid, open objects are not included in the interpolation process (see Subdivision Surface - Keep Hard Edges 163).

Include All Edges - The entire marked selection is interpolated.

Sometimes it happens that a mesh tears open when applying the Smooth Surface function. In this case the mesh structure of the object is not homogenous. You can try to repair the mesh structure using the <Repair Mesh Structure> function.

See also: Subdivision Surface - Homogenous Mesh 163, Triangulate Selection 237 and Repair Mesh Structure 240

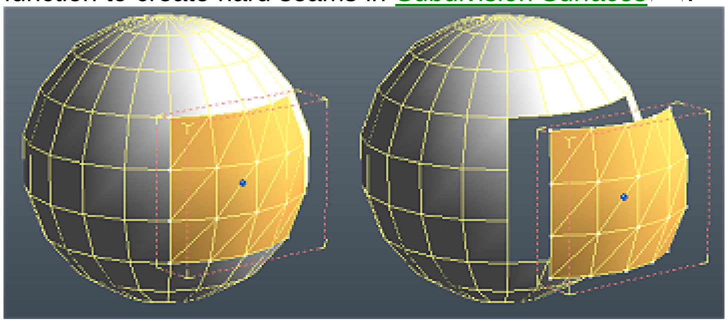


Here again is our example from the "Facet Extrude" tutorial 51. On the left side the completed object and on the right side the same object after applying the smooth function twice. But be careful with this function as each smoothing operation will significantly increase the number of points and facets in the object. On the other hand each work-step can be canceled or repeated with the Undo/Redo functions 28, so just experiment and play around with it a little bit.

9.7.8 Detach Face Selection

Menu: Edit Object 227 - Detach Face Selection

A selection of facets is detached from the object's mesh by duplicating all points at the boundary of the marked surface. This way the facets selected and the remaining facets of the original mesh are provided with their own separated mesh point (in principle it is the same function as <Detach as new Object 239> - the only difference is that the detached surface still remains in the original object structure and is not copied into a new object).
Now you can move the face selection to separate it from the remaining mesh. You can also use this function to create hard seams in Subdivision Surfaces 163.

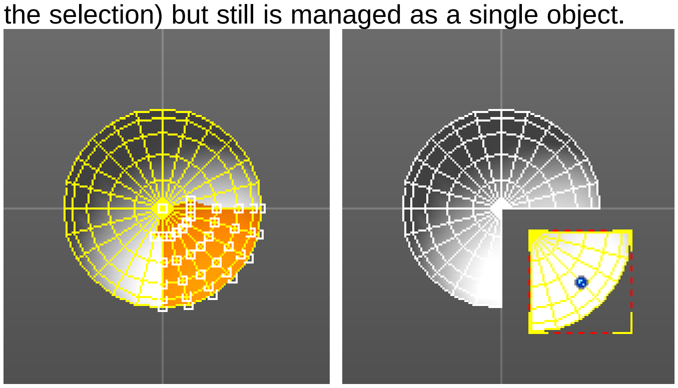


Example: A facet selection is detached from the original mesh using the <Detach Face Selection> function. At first sight the model still looks the same. Only when moving the detached selection away a little does it become clear that the object now consists of two separated meshes.

9.7.9 Detach as new Object

Menu: Edit Object 227 - Detach as new Object

You can detach a face selection from an object and copy the separated surface into a new object using the <Detach as new Object> function. This is in contrast to the <Detach Face Selection 239> function, where a surface is separated from the main mesh (by inserting points at the boundary of the selection) but still is managed as a single object.

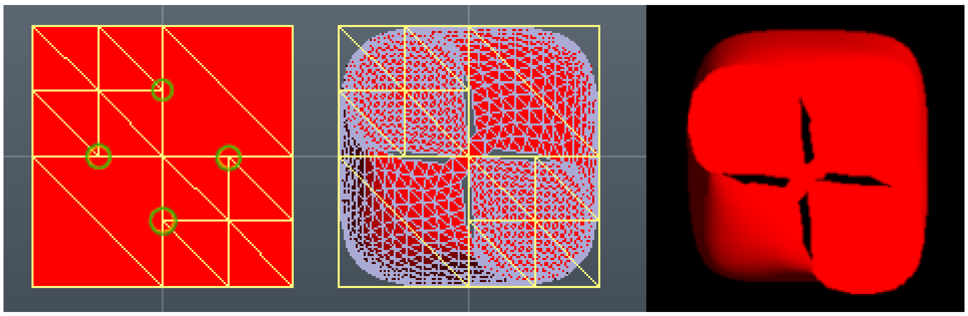


In the illustration on the left you see a sphere with a facet selection. The marked faces enclose a quarter of the sphere. Operating the <Detach as new Object> function changes the sphere as shown in the picture on the right. It now consists of two distinct objects (moved slightly apart for clarity).

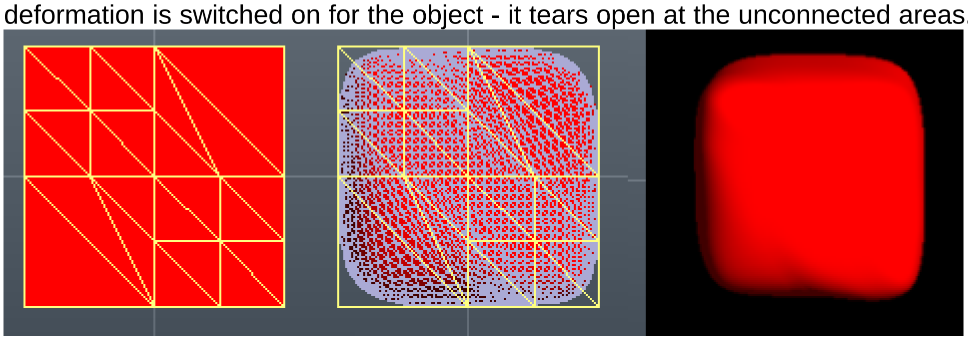
9.7.10 Repair Mesh Structure

Menu: Edit Object 227 - Repair Mesh Structure

For some modeling purposes it is necessary to have a homogenous net structure. For instance, Subdivision Surface 163 deformation can only be applied properly if the object consists of a single homogenous mesh. This means that all triangular facets have to be connected with a common edge (sharing the same edge points) to all of their adjacent facets. This can not be taken for granted. When importing foreign files or models created with previous versions of MR-3D Designer (up to v.11.x) it can happen that a corner of a triangle is not connected to all adjacent facets. If you apply a subdivision algorithm to such a model the mesh will probably tear open at those areas where the net is not consistent.



The picture above on the left illustrates the case. The problem points are marked with a green circle. These are vertices of smaller facets that are not connected with a common edge with their adjacent larger facets. The pictures in the middle and on the right show what happens if SDS-deformation is switched on for the object - it tears open at the unconnected areas.

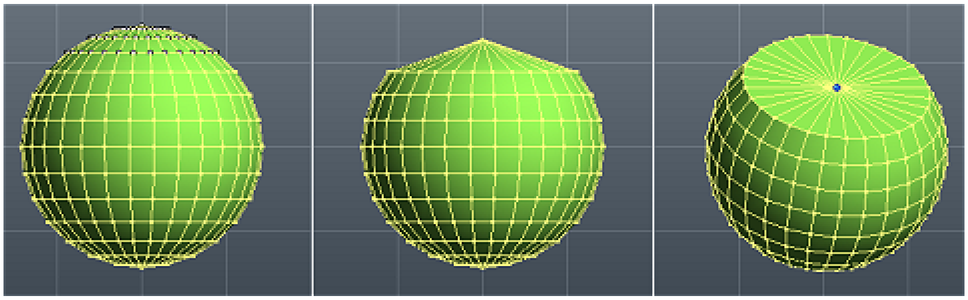


You can try to repair such a mesh using the <Repair Mesh Structure> function. The picture above shows the result after applying this function to our box mesh. At those locations where vertices of a smaller facet came to lie onto an edge of a larger facet these points were used as splitting points to cut the larger facets into two halves. In situations where this function doesn't work duplicate points or points lying very close to each other have to be removed first. Try the combination "Delete" <duplicate Points> followed by <Repair Mesh Structure>. Repeat the process if necessary with a higher tolerance value for the removal of duplicate points.

9.7.11 Melt Point Selection

Menu: Edit Object 227 - Melt Point Selection

A selection of points is melted into one single point. The new point replacing the old selection is positioned in the center of the previous point selection. Facets that degenerate to points or lines by this operation are automatically deleted.

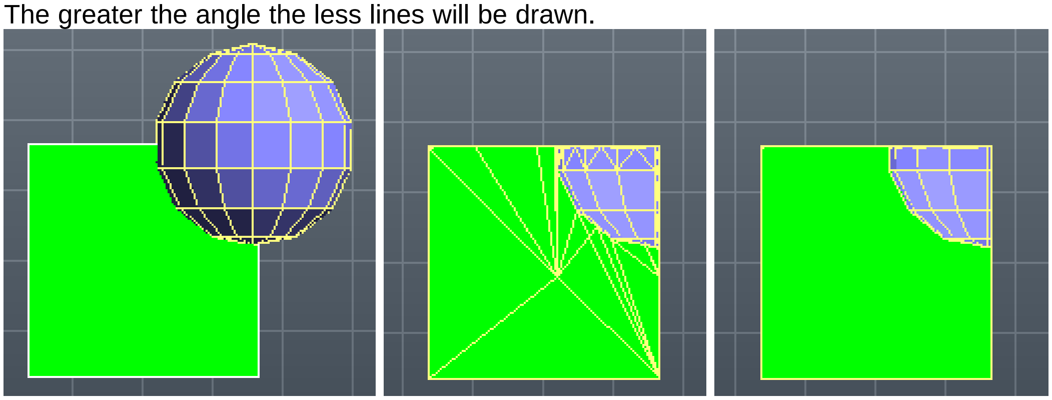


Example: The picture above on the left shows a sphere with a point selection covering the upper cap of the sphere. After pressing the <Melt Point Selection> button all marked points were replaced by a single new point and all superfluous facets were deleted (picture in the middle). After selecting the new midpoint and moving it slightly downwards, to be level with the topmost segment points of the mesh, we get a clean spherical cap (picture on the right).

9.7.12 Line Visibility

Menu: Edit Objects 227 - Line Visibility

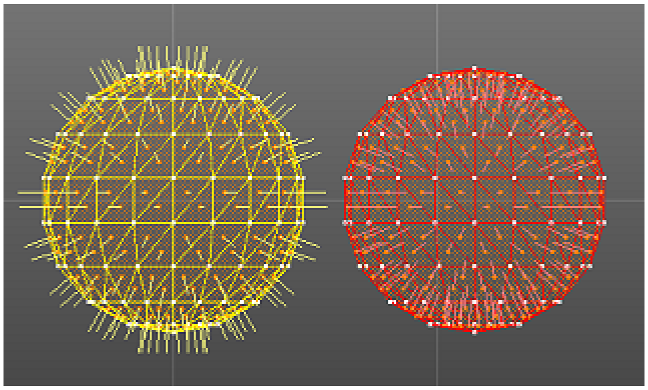
In MR-3D Designer each triangular facet holds a line flag with information indicating whether an edge should be drawn or not. Often it is not necessary to draw all three lines to get an exact image of the object. With a quadrilateral surface, for example, which is comprised of two triangles, you need not draw the middle line. The same applies to all facets forming a certain outline on an even plane (e.g. the circular cap of a cylinder). With the Set Line Visibility you can reset all line flags in dependence of the angle of a facet to a neighbouring facet. Only when the angle is greater than the specified angle will a connecting line be drawn between two adjacent facets. If you choose an angle of 0° then all lines of all facets will be drawn. With 1° you will remove all lines within a level surface. The greater the angle the less lines will be drawn.



Example: The picture above on the left shows a sphere which is overlapping with a box. The picture in the middle shows the box after applying the Boolean Operation "Subtract: Box - Sphere". The side walls were subdivided into a lot of smaller faces by this operation and this is indicated by corresponding line flags set for the new facets. After that the <Line Visibility> function was used with an angle of 1° to remove all disturbing lines lying in the same plane, so that only those lines of visual importance are drawn (picture on the right).

9.7.13 Invert Normals

Menu: Edit Object 227 - Invert Normals



The object normal 8 is a vector standing vertical to each facet of an object. It is important on the one hand for the visibility calculation 290 and on the other hand for the light incidence and interpolation calculations of the facets.
When you import objects of foreign formats 34 (e.g. of the type [DXF] or [RAW]) that do not contain the necessary information for aligning the normals, or if you edit and add new facets for a selected object, MR-3D Designer attempts to automatically assign sensible normal alignments to objects and facets. However, this is not always possible. For instance, in many cases the normals are uniformly aligned when the object is imported, but it could happen that the normals are then all shown pointing to the interior instead of to the exterior. In this case you can use the function <Invert Normals> to reverse the normal alignments of all marked 138 facets. To ease work you can switch on the "View/Normal 22 "-entry in the menu bar to show all the normals of previously marked facets of an object. In the illustration above you can see two spheres, the left one with normals correctly aligned to the outside and the right one after applying the <Invert Normals> function with normals pointing to the inside of the sphere.

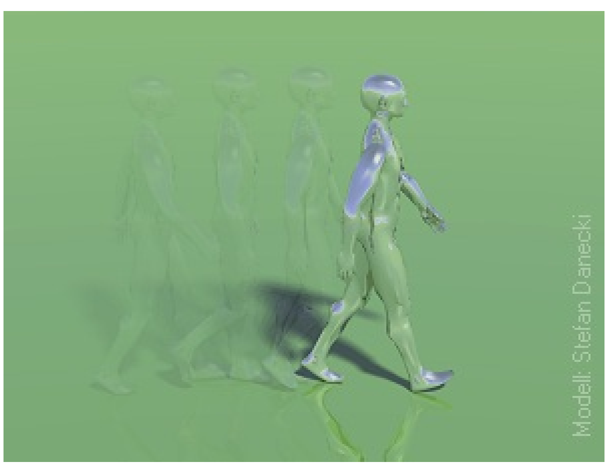
See also: The surface normal 8

9.7.14 Selective Facet Interpolation

Menu: Edit Object 227 - Facets-Interpolation

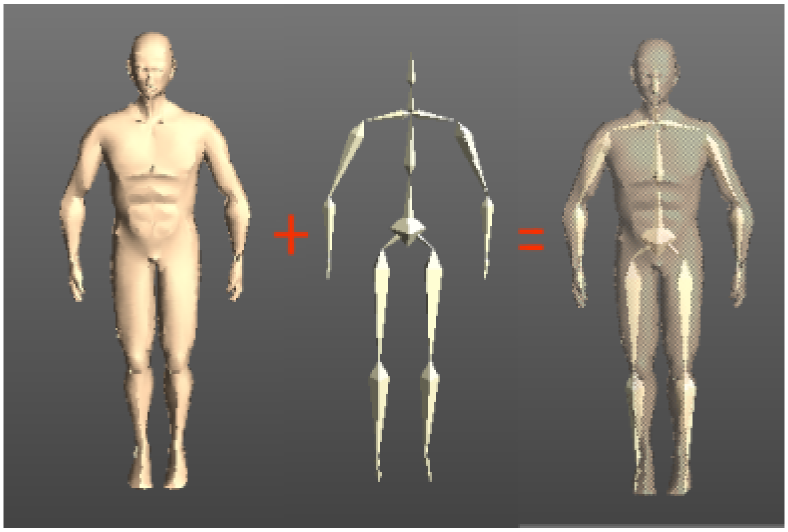
At the bottom of the toolbox are two more buttons that allow you to switch interpolation off or on again for selected facets. This function is only relevant if Interpolation is chosen in the Material-dialog for the depiction of the object. For further explanation of the purpose of this function, as well as examples see: Object Properties - Interpolation 290.

9.8 Edit Skin and Bones - Create a Skeleton and Allocate Skin Points

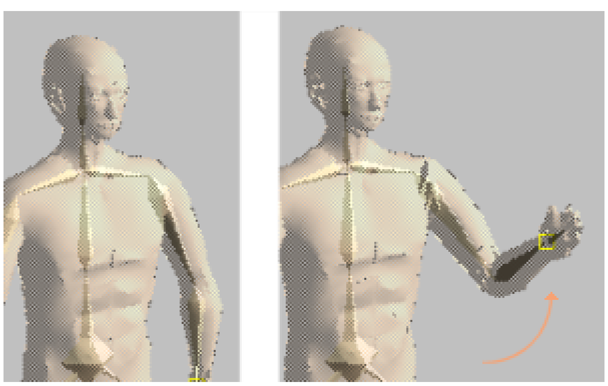


The Principle of Skeletal Deformation (Skin and Bones)

Production of modern animation films, especially the animation of characters, would be impossible without skeletal deformation . The skin and bones technique uses a skeleton which is subordinated to a corresponding skin object enveloping the skeleton. Now, every time you move a bone of the skeleton, the particular part of the skin previously assigned to that bone will be deformed and move with the bone - the character comes to life. What was usually previously left to awfully expensive animation studios, you can now also do at home with the MR-3D Designer.



A typical skeleton for a human model. The pelvis is used as a root bone from which all other bones branch off. The root bone, being the top most parent of the bones hierarchy tree 140, can be used to move the whole skeleton to align it within the skin object. Then, the root bone again is subordinated to the skin object. If you want to move, rotate or scale the whole character, for instance, to move it to a new starting position, then you always have to select the characters skin - the subordinated skeleton will automatically follow each movement of the skin. But if you just want to move a part of the body, for instance, if you want to bend an arm, then you have to select the corresponding bone of the skeleton to rotate the bone together with the skin assigned to it into the bend position.



Deformation of the arm by rotating the bone of the forearm

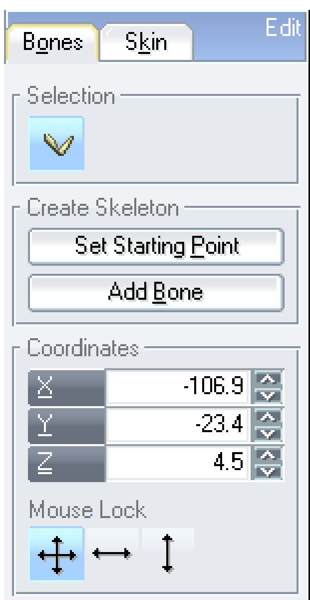
Skeletons can be easily created in the "Edit Skin and Bones" work mode. First, generate a root bone. Further bones can be added simply by selecting existing bones and pulling new bones out of the tip of these bones. Even complex skeleton hierarchies can be build this way with a few mouse clicks. Afterwards you link the completed skeleton under a normal polygonal- or NURBS-object. By doing this the skeleton will automatically recognize its new parent object as a deformable skin object. Now, in Modeling Mode, the individual bones can be aligned within the skin object. Every bone will influence only a certain area of the skin, so the next step will be to allocate the

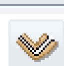
corresponding skin points to the individual bones. If you want to apply Inverse Kinematics to animate the skeleton, then you also have to define the DOFs (degrees of freedom) to restrict the bone rotations to their natural range.
After all these preparations you can start to animate the character. Change to Animation Mode for this. While in Modeling Mode 19 each movement of a bone will be interpreted as an adjustment of the bone's position within the skin. In Animation Mode, moving a bone will always produce a deformation of the skin assigned to this bone. Because of this clear distinction between Modeling Mode and Animation Mode you can always change back to Modeling Mode (e.g., to add further bones to the skeleton or to redistribute the point assignments) even in the middle of an already set up animation. The character will automatically adopt these changes for the whole animation.

Bones can also be used to simulate muscles, e.g. you can produce facial animations by adding several bones under the skin of the character's head and assigning certain areas of the face to each of them. (In commercial animation films up to 60 bones are used to control the expressions of the face alone)

Another feature: A bone object that is subordinated in a hierarchy under a normal facet-based object will automatically recognize this object as the skin object belonging to it. But you can also do it the other way round. You can link "ordinary" objects under bones, for instance, to link a tool or a weapon under a hand bone, so that it is automatically taken along with the hand's movements. Or may be you want to pack a rucksack on the top of the spine bones.

The "Edit Skin and Bones" menu



If you select the  button to change into the "Edit Skin and Bones" work mode then you also automatically change into Modeling Mode.

The menu is divided into 2 pages. The "Edit Bones" page provides all tools to create skeletons. On the "Edit Skin" page tools are provided that allow you to allocate the points of the skin to an individual bone.

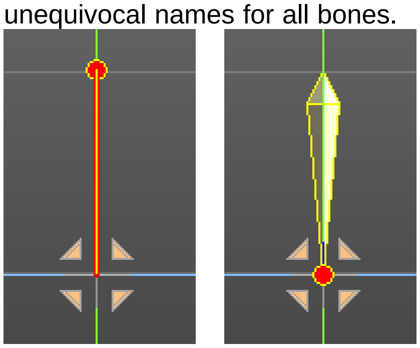
In the "Edit skin and bones" work mode, for clarity, objects that are neither bone nor skin are depicted in simple,dimmed wire-frame form. If a skeleton is already subordinated to a skin object, the skin is always drawn as a transparent object so that the bones underneath can be easily recognized.
In "Edit Bones" work mode the individual bones are always the focus of interest and therefore only

bones can be selected for editing. If you click on a skin object to select it, the top-most root bone subordinated to the skin will automatically be selected.
In "Edit Skin" work mode the points of a skin object can be selected to allocate them to the individual bones.

Note: Skins are automatically drawn transparently only in "Edit Skin and Bones" work mode. However, you can also switch this option on via the "View - Bones - Transparent Skin" menu entry for all other work modes. There are some more options, for instance, to hide skeletons or skins in the viewport depiction. See "Depiction in the Viewport Windows 22"

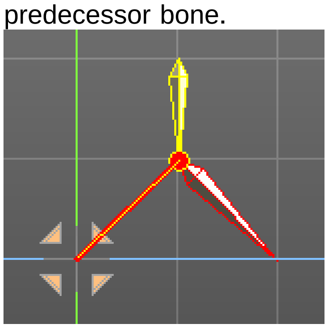
Creating a Skeleton

When you activate the "Edit Bones" page in the tool window, crosshairs appears in the viewports. Grab the crosshairs by clicking into the center between the four arrows of the crosshairs, and move it to the position where you want to set the starting point of the next bone. Then simply operate the <Set Starting Point> button to mark this position with a big red point. Grab the crosshairs again and move it to another position. Now a thick red line is drawn from the starting point to the current crosshairs position. To create a new bone running along this line just press the <Add Bone> button. Thereupon, the name dialog appears where you can give it a suitable name. Since a complex skeleton can consist of dozens, up to hundreds of bones, you should go to the trouble and assign unequivocal names for all bones.

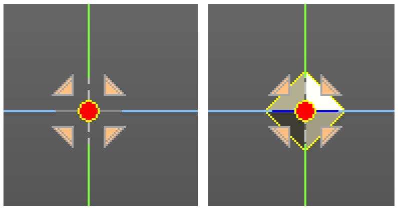


The picture on the left shows the starting point and the line dragged out off it. On the right you see the bone created from it.

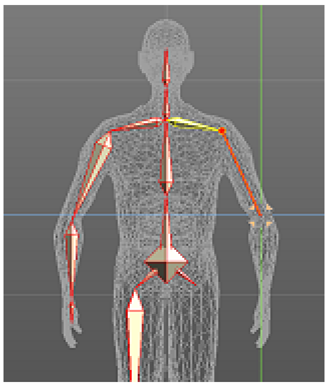
The newly created bone will automatically be selected and the starting point for the next bone moves automatically to the tip of the bone. Now you can easily drag out another bone from the tip of the currently selected bone. Every time you create a new bone by dragging it out of the tip of an already existing bone the new bone will be automatically hierarchically subordinated to his predecessor bone.



The number of bones you can drag out of an individual bone is not restricted. You just need to select a bone again in order to drag another bone out of it. The new bone will be inserted automatically into the correct place of the skeleton hierarchy.



You can also create bones that have no length, for instance, as a parent root bone, from which all other bones will branch off, like in the example with the pelvis for the human skeleton at the beginning of this chapter. To create a root bone simply set the starting point at the desired position and then immediately create the bone, without dragging a line out of it.



Of course, it's easier to create a suitable skeleton, if the corresponding skin object already exists. Then you only need to draw and create the bones along the individual limbs.

Allocate the Skeleton to a Skin Object

As explained before, you only need to hierarchically 140 subordinate the final skeleton under the corresponding skin object. In the Object Selection 133 window, simply grab the root bone of the skeleton with the mouse and drag the whole skeleton branch onto the skin object. After that the skin object will automatically recognize its new children as a hierarchy of bones that can deform the skin object and vice versa.

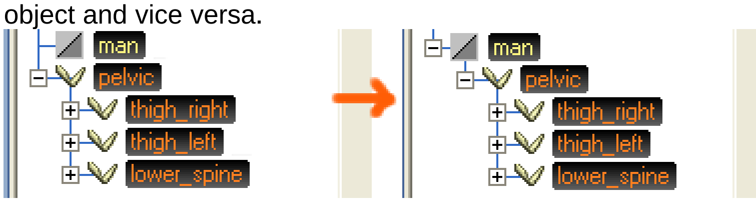
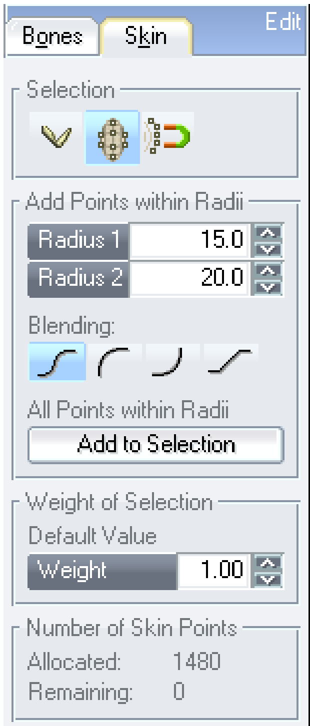


Illustration: The polygon object "man" becomes automatically a skin object just by subordinating a skeleton under it.

However, the surface points of the skin have still to be distributed to the individual bones of the skeleton. Therefore, now switch over to the "Edit Skin" page of the tool window.

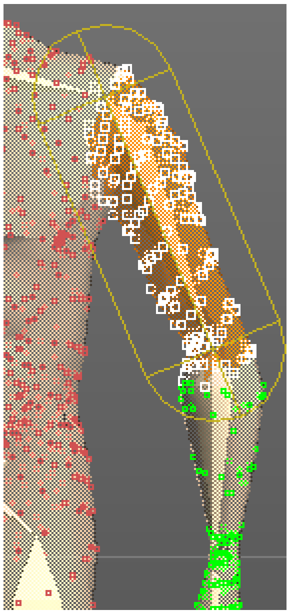
Allocating Skin Points to Bones and Assigning Point Weights



At the top of the tool window now two additional "Selection" buttons are available:

-  Activate this button to select a particular bone in the viewport window.
-  Change to "Point Selection" to be able to allocate points from the skin's surface to the currently selected bone.
-  In "Change Weight of Point Selection" work mode you can select points of a previously made point selection to change the weight of individual points.

To add a point selection to a bone you first have to select the corresponding bone in "Bone Selection" mode. Then change to "Point Selection" (You can work more swiftly when you select the bones directly in the Object Selection 133 window instead of the viewport - then you don't need to change to and fro from the viewport's bone selection mode to point selection mode and vice versa) . Now you can add and remove individual points to the point selection 138 in the same way as you do in other work modes. But, in contrast to the other work modes, this time the point selection will be saved as a permanent point selection associated with that particular bone. Points can be allocated to several bones simultaneously, for instance, in the area of joints the points overlapping the joint can be allocated to the bones on either side of the joint. Those points will be influenced proportionately by the bone movements. The weight by which a point is influenced through the movement of a bone is initialised and distributed automatically between the corresponding bones, but you can change the weight of individual points later.



The illustration shows an example of a point selection for an upper arm. For a better differentiation, those points already allocated to other bones are drawn in a separate green color. In the picture the skin points for the forearm and hand have been already allocated to their corresponding bones, so they appear in green. At the bottom of the tool window you can read the number of point already distributed and the number of remaining points. If not all points are allocated this will show up very soon in the animation, since these points will stay at their old positions while all other points will be carried along with the bone movements.

Allocating Skin Points with help of Frozen Point Selections 133

Frozen point selections are useful to quickly restore previously made point selections for a faster work flow when modelling an object. But you can also use frozen point selections to preserve a point selection for a later allocation to bones. The points allocated to bones are not saved in frozen selections but in a separate memory space together with the bones. From this it follows that if you allocate a frozen selection to a bone and later add points to or remove points from the frozen selection this will not change the point selection allocated to the bones. But with frozen selections it is very easy to update the point list allocated to a bone. You just need to change to "Allocate Skin Points" work mode and then a simple click on the name of the frozen selection will allocate the new point list to the currently selected bone. The main advantage: Frozen point selections are processed and extended automatically, for example, when refining an area of a skin object with a triangulation function. If the triangulated surface is part of a frozen selection then this frozen selection will be extended automatically with the newly generated facets and points. Now you can reallocate this extended selection easily to the corresponding bone instead of having to allocate all new points manually.

Weight of Points

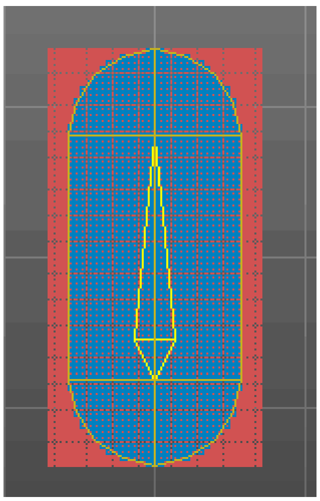
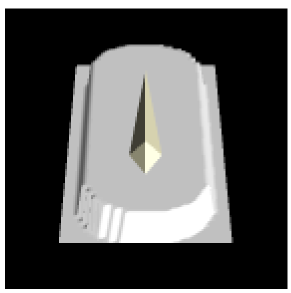
The weight of a point determines how greatly a point will be influenced by the movement of the bone it was allocated to. If the weight is smaller than the maximum value of 1, then the point falls behind the movement of the bone. If a point is allocated to several bones at the same time, then the weight will be distributed evenly among these bones, so that the total weight will not exceed the maximum value of 1. In case of character animations all points have usually a total weight of 1, because it makes no sense for individual points to fall behind a character walking straight forward. If you use bones to simulate facial muscles, however, the bones pull with different strengths at the allocated points of the face instead of just moving the assigned area back and forth.

Initial Weight of Points - Mouse Selection

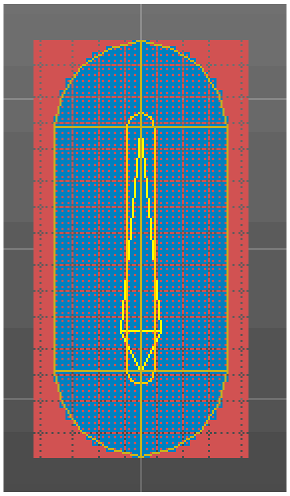
An initial value (0.01 to 1) for the point weight can be set in the "Weight of Selection" box. All points that you add to the point selection with your mouse will be initialised with this default weight. To change the weight of already selected points switch over to the  "Change Weight of Point" selection. In this work mode you can only select points from the point selection already allocated to a bone. If you now change the "Weight of Selection" parameter all currently selected points will adopt the new value.

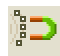
Selecting Points using a Bone Radius

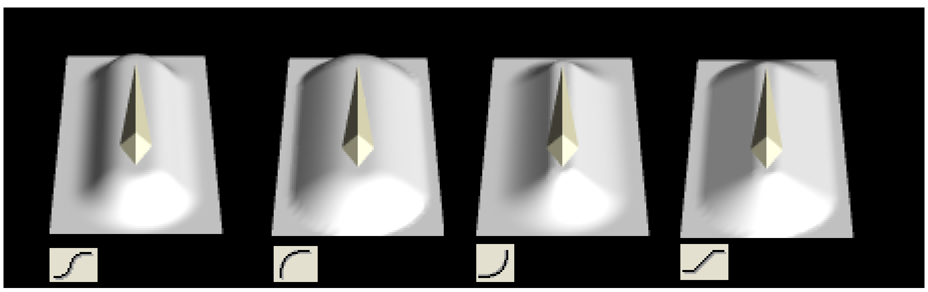
Often skin points lie in a certain radius around the bone they have to be allocated to, for instance, the skin points for legs and arms. If you select a bone in "Edit Skin" work mode two capsule-shaped outlines are always drawn around that bone. You can automatically add all points within that area to your current point selection by pressing the "All Points within Radii - Add to Selection" button. The initial weight of the points added to the selection depends on whether they are located within the inner radius or between the inner and outer radius, respectively. The default "Weight of Selection"-value will be valid for all points lying within the inner area of the two capsules. Then, from the inner to the outer radius a weight transition from the default value down to zero is calculated. Four different filter functions can be applied for this transition .

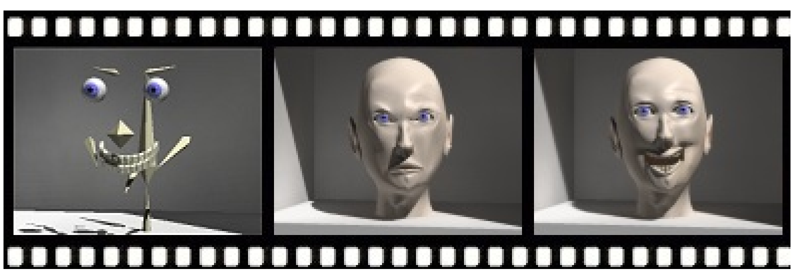
A high resolution grid is best suited to demonstrate the differences. A single bone is subordinated to this grid patch. A point selection was added using the "All Points within Radii - Add to Selection" function. Inner and outer radius were of the same size, so all points within the encapsulation were allocated with the same initial weight of 1. Then, in Animation Mode, the bone was lifted slightly above the grid patch. The second picture shows the result - all points allocated to the bone follow with exactly the same displacement.



Now we want to test the filter functions. In the  "Change Weight of Point" work mode. We first reduce the inner radius to a very small area around the bone, so that a transition can be calculated for those points lying between the inner and the outer radius. Operate the "All Points within Radii - Change Weight" button to calculate the new weights for all marked points located within the capsular outlines.



Now we change back into Animation Mode. The bone movement is adopted now by the point selection with different weights within the transition area. In the illustration you see the effects of the different filter functions.

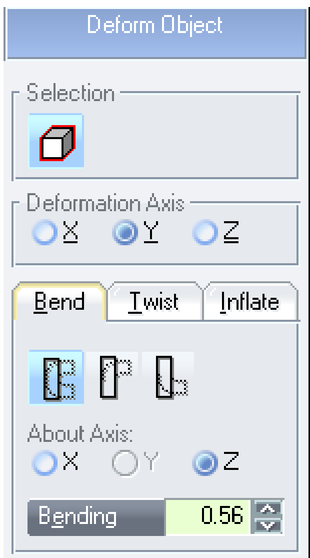


The illustration above shows some details from a short video demonstrating a facial animation (www.3d-designer.com/en/galery/galery.htm). In addition to the neck and head bones the model is provided with eyes, a set of teeth and some additional bones in order to simulate facial muscles. On each side of the mouth a bone was added to control the corners of the mouth and parts of the cheek. If the point weights have been assigned properly you only need to scale down the bone slightly to "pull" at the corner of a mouth and to conjure up a little smile on the face. To transform this smile in a full laugh a third bone was applied to control the up and down of the lower jaw. You can also rotate the mouth bones a little bit downwards to change from the smiling face to a somewhat morose appearance. Take care, that all bones of the face as well as the eyes and the set of teeth are subordinated to the head bone. Then, if you move or rotate the head bone, the whole head including all subordinated objects and bones will follow this movement. Surprisingly, only 6 additional bones were applied to animate the facial expressions in this demo video. (In contrast to commercial films, where up to 60 bones are used to animate the complex interacting of facial muscles.)

See also:
- Animation 400
- Tutorial - Character Animation 97

Example files:
An example of a character already animated in a simple walking sequence is provided in the projects-folder under "..projects/character/man_walk.cmo". You can also find the pure animated skeleton (without skin) of that scene in the project folder "..projects/character/skeleton_walk.cmo". The model of the character was provided kindly by the artist Stefan Danecki.

9.9 Animated Object Deformation



 - Menu "Edit - Deform Object"

Deform Objects
This function offers powerful and easy-to-use features to deform facet-based objects. These are fully animated and bitmaps as well as textures will follow the object's deformation like a tight-fitting skin. A number of things should be kept in mind:
- Deformation is only possible for a complete selected object 138 (or an object branch) rather than selected points/facets. In an object branch, all objects of lower hierarchy 140 will follow the deformations of higher hierarchy objects, but additionally, they can undergo their own deformation. So this function works much like any other complex hierarchical animation - see our tutorial's robot arm example 62.
- Single points 138 cannot be selected and/or deformed.
- Several types of deformation can be used at once on an object/branch.
- Deformation is temporary and can be adjusted or even completely reset by changing the deformation parameters at any time.
- This function is primarily a powerful extension of the program's animation 401 features. Producing a deformation animation is as straightforward as producing one with object rotation or movement. If you change any deformation parameter of any object, a new key frame will automatically be produced to store the animation data. Between any two key frames, deformation values will be

interpolated from the values of both keys.

- Analytical Objects 147, by definition, cannot be deformed by facet based deformation. But any such object included in a deformed hierarchy branch will correctly change its position according to the deformation applied to its master object(s). The same goes for camera and light objects included in an animation branch.

If you want to apply the deformation tools for modeling

There are several ways objects can be deformed in an animation, for instance by applying a skeletal deformation or by using one of the animated bend or twist deformation tools of this menu. These deformations don't change the original model data of an object - they are performed rather as a part of the render pipeline immediately before rendering. After each work step, an internal copy of the original model is created, deformed and passed on to the render engine. In some situations it would be useful to apply these deformation tools not only for the animation process but also for the modelling part - for instance, if you want to bend an object permanently to alter the shape of the model. For this purpose you can use the "Copy Current State to Model Data 133". This copies the data of the object in its deformed state to the model data of a new object - all animation data is ignored and will not be transfered.

A Global Switch to Deactivate Deformations

The  <OFF> button will deactivate any deformation settings. This applies to the bend and twist deformations as well as for the auto-aligning 290 feature. This is rather useful in the modeling process. For instance, if you want to move or rotate objects, these changes would be calculated for the objects before deformation is recalculated in the render pipeline. As it is rather difficult to foresee the outcome of such complex operations, you can switch off the deformation and reactivate it by pressing the button again after you have placed the object(s) precisely where you want them.

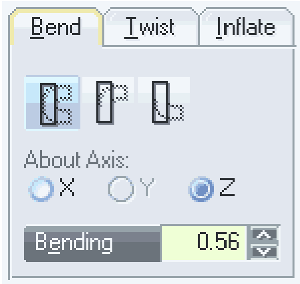
Deformation Axis

Deformation always takes place in respect to a specific axis of the object. For example, if you want to twist an object in relation to its y axis, you will have to select precisely that. But you can always switch to a different axis, because the deformation functions can always be undone by changing the parameters - keep in mind that deformation is part of the animation feature and does not permanently change the object's form. So even the most ridiculous deformation can be undone by deleting the animation, returning your object to its original form.

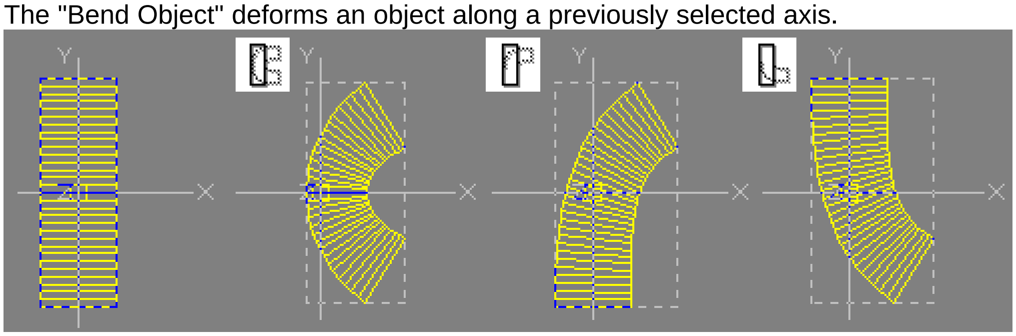
Deformation types

Much like the other working modes, deformation offers two different ways to work on the objects: Direct input of deformation parameters as well as real-time deformation by pressing the left mouse button and dragging. You can choose between one of the three deformation types "Bend", "Twist" or "Inflate" by selecting the corresponding tab button in the tool window. You can combine all deformation types, for instance, you can start by twisting the object, then bending and inflating it by the desired amount.

Bend Object

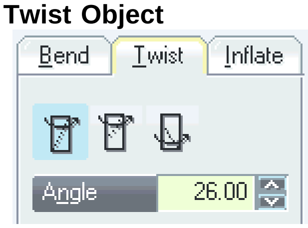


The "Bend Object" deforms an object along a previously selected axis.

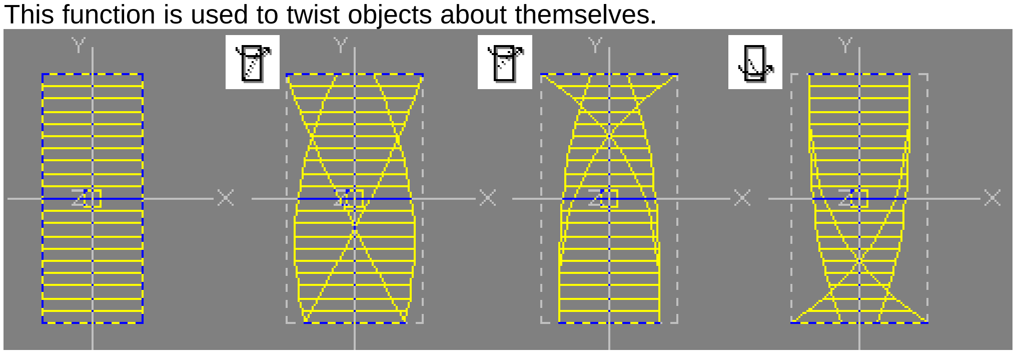


The example above shows a long block made up of several segments to form a single object. Deformation is selected for the y-axis in this case. In the "Bend Object" menu, the z-axis has been selected as the additional bending axis. See the deformed result (2nd object from left), where the object has in fact been "grabbed" by both ends of its y-axis and bent around its z-axis.
Maximum bending amount is ±1. A value of zero will return the object to its original form.
Objects can be bent from either or both of their ends. The screenshot above shows the different results of "grabbing" the object either from both of its ends or only one end. Settings for these functions are made using the three buttons also shown in the screenshot.

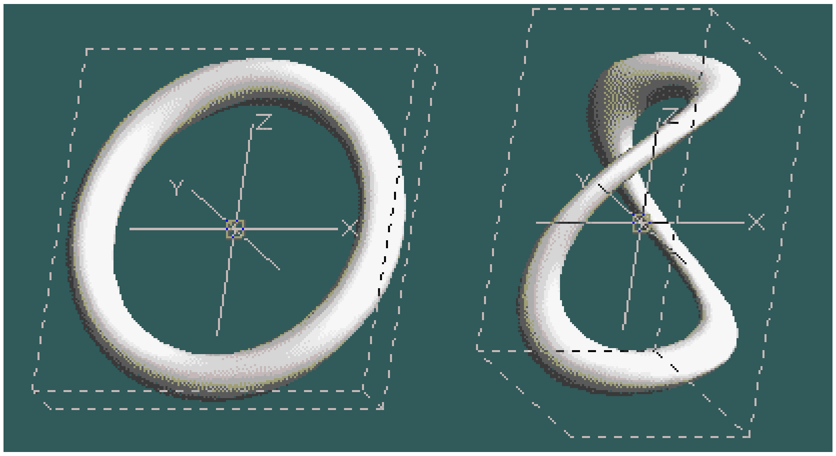
Twist Object



This function is used to twist objects about themselves.

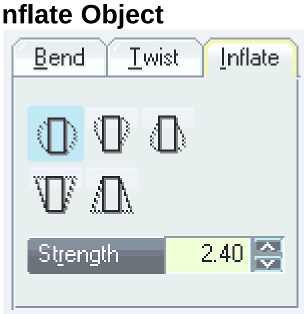


From the left, this screenshot shows the object in its original form - a solid block. The 2nd object shows the result of twisting by 120 degrees along its y-axis. As before, you can use the buttons to twist the object from both-ends or one end of the axis, with the results being shown by the last two objects in the screenshot. Deformation can be adjusted to any value around ±180 degrees. A value of zero will return the object to its original form.

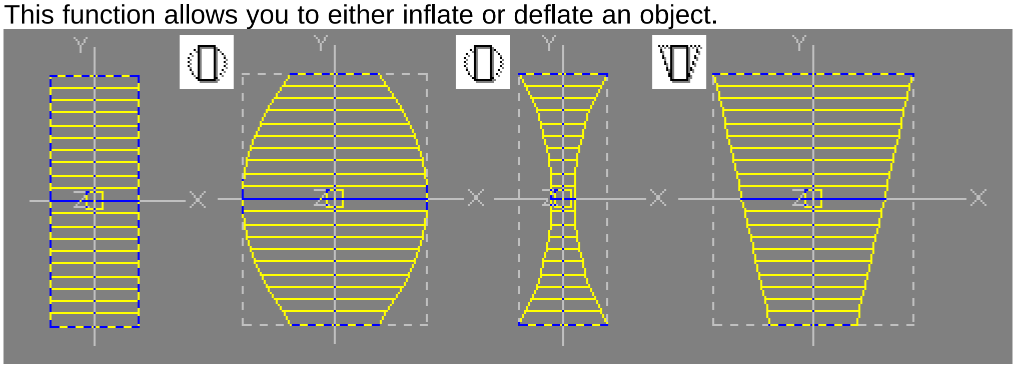


This is a rather impressive example of a torus deformed by the twist function.

Inflate Object



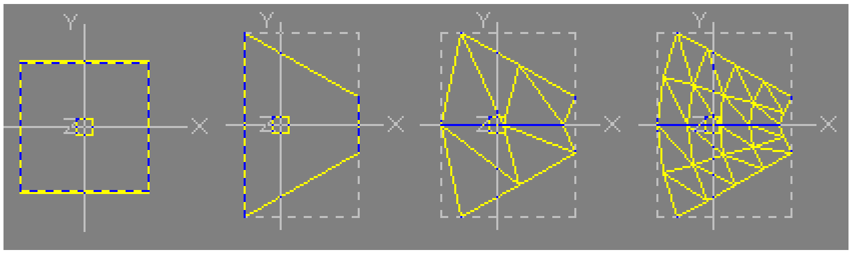
This function allows you to either inflate or deflate an object.



Again, the screenshot shows a simple block-shaped object. The 2nd object from the left has been inflated by using the "blow-up" function, where the object is deformed much like a balloon. The next object has been deformed using the same function, but set to a scale value lower then 1, thus inverting the deformation curvature. As before, you can use the buttons to deform the object from both-ends or one end of the axis - the result being shown by the last object in the screenshot, where the deformation was made from the upper end of its axis, but without the curved result.
Use the "Scale" parameter to define the amount of deformation. A value higher than 1 will inflate the object, while values lower than 1 will result in deflation. A value of zero will return the object to its original form.

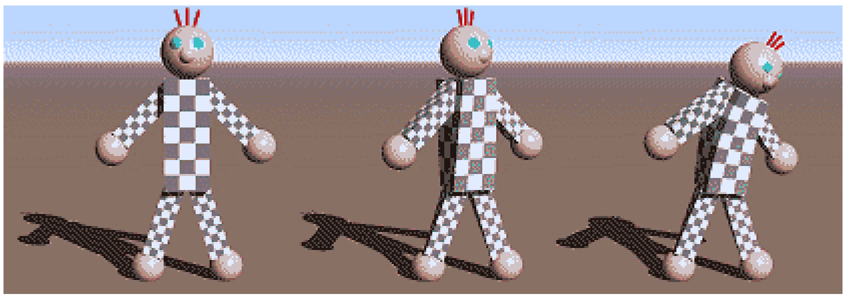
Object Resolution

As the exact shape of an deformed object will always depend on the number of points and facets, you will need a reasonably high object resolution for smoothly curved shapes. You can use the <Triangulate Selection 237> function of the Edit Object 227 menu to increase the resolution by splitting the facets, thus increasing the total number of points/facets in an object.



The screenshot again shows a simple block followed by a version deformed by bending. After using the <Triangulate Selection> function twice, the object has been finely shaped with the curvature becoming more pronounced with each step (see right). In practice, however, you would try to find a compromise between visual quality and point/facet count, as more facets will inevitably slow down the rendering process.

Hierarchical Deformation



Deformation will be inherited throughout a hierarchical object branch, meaning that lower order objects will deform according to the deformation rules applied to higher order objects. The screenshot shows a typical example that uses this feature to good effect: Rectangular "arms/legs" and spherical "hands" have been associated in an object branch with the rectangular "torso" of the puppet. The "head" consists of spheres and cylinders and is also integrated into the object branch. Now by selecting the "torso" and deforming it using the "Twist Object" and "Bend Object" functions, the deformations will also work in the lower order objects in the branch. Note that all spheres used in this animation are analytical objects! While analytical objects cannot be deformed (as they are made up of a mid point and radius in this case rather than points and facets), still, the midpoints are shifted according to the deformation applied to the higher order object, so they will in fact follow the deformation instead of staying in the same position.
Another special feature is the flexible texture function, which is not aligned with the object-axes, but smoothly follows an object's surface even when it is deformed.

See also: Tutorial - Animation and Deformation - Dolphin Movements 93

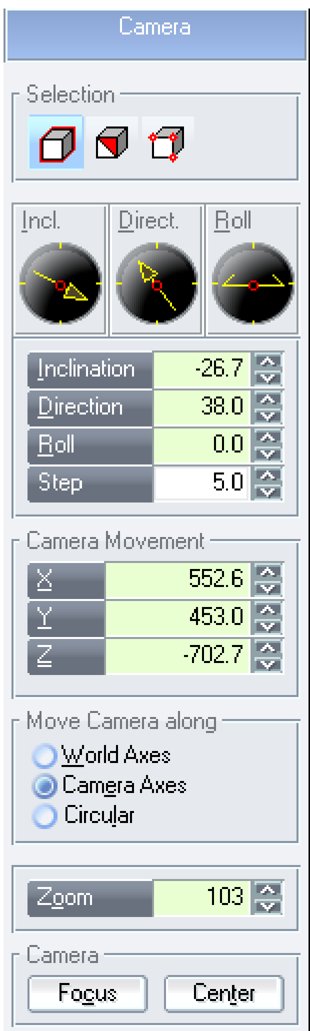
Part X

10 The Camera, Render Options and Start Rendering

The camera settings and the various render options are explained here.

- **Camera - Movement and Alignment** 257
  There are several possibilities to move a camera and align it to a scene
- **Render Options** 263
  Choose an appropriate rendering algorithms, picture resolution and special effects to determine the output quality of your picture calculation
- **Start Rendering of an Image or Animation** 259
  How to start or interrupt rendering of an image or animation and how to save pictures and animations

10.1 Camera - Movement and Alignment



 - Menu "Edit - Camera"

The Camera

The camera-menu gives you freedom to move interactively in the area and to insert any rotation and inclination angles for the camera. As with a real camera, you can implement Tele-photo or wide-angle settings over a zoom feature. Positioning the camera is very flexible in MR-3D Designer. On the one hand you can edit directly the coordinates of the camera or move with help of the arrow buttons next to the coordinates along the world-axes or along the object-axes of the camera or even in an orbit about another object.
On the other hand - since the camera can be manipulated like every other object - it can also be repositioned directly (in the "Move Object 197" mode) or rotated to set the viewing-direction (in the "Rotate Object 211" mode). If you move or rotate the camera object in a viewport window you can see in the camera viewport how the visible camera-detail changes.
Furthermore you can change the camera position just by clicking in the camera viewport and moving the mouse, independent from the work mode present.
Finally you can arrange the camera under an object hierarchy 140. Suppose you want an object to fly through an area and the camera should always line up on the flying object. If you link the camera hierarchically under the object, the camera is automatically moved with the flying object when the flying object is moved 197, scaled 206 or rotated 211 in the respective work menus.

Camera Alignment

At the top of the tool window are three angle instruments for the alignment of the camera. You can enter the angles via the keyboard or just click into an instrument and drag the pointer to the desired position.

- **Inclination** - The left instruments shows the inclination angle (pitch) of the camera.
- **Direction** - In the center box is shown the direction angle of the camera.
- **Roll** - The right instrument displays the lateral inclination-angle , which is the rotation of the camera about its viewing axis. The dial consists of a crosswise-beam, which has markings at both ends. This works precisely as the dial showing the horizontal level in an aircraft. If you roll the camera to the left or right along its axis the horizon rolls in exactly the opposite direction. Rolling through 180 degrees turns the camera upside-down and the markings on the beam point to the underside.
- **Step** - This parameter defines the step-width, with which the camera is rotated or moved along its axes when rotating or moving the camera via the arrow buttons next to the edit fields.

Moving the Camera Directly in the Camera Viewport Window

By clicking in the camera viewport window and moving the mouse you can easily change the camera position independent from the work mode present. The movement direction depends from the selection made in the camera menu (see below). If <World Axes> mode is active, the camera moves on the x- and y-planes. In <Camera Axes> mode the movement is along the x- and y-planes defined by the camera axes.
Pressing the right mouse button instead of the left one will move the camera forward or back in viewing direction.
If you hold the left and right mouse button pressed simultaneously (or left mouse button only when <Circular> movement mode is set in the camera menu) then the camera moves in an orbit about an object.

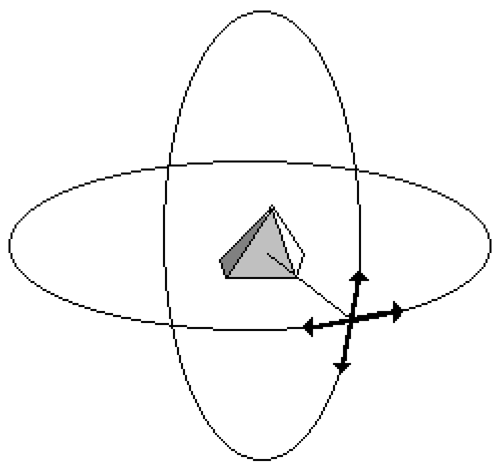
Camera Position

The current location of the camera is shown in the "Camera Movement" box.
The coordinates can also be input directly from the keyboard by choosing the X, Y, Z-buttons.

Moving the Camera using the Coordinate Arrow Buttons

You can use the arrow buttons next to the camera coordinates to move the camera throughout the area. Operate the buttons beside the X-coordinate, and the camera moves along the spatial X axis. The same is true for the other buttons. The step-width, with which you move and rotate the camera, can be put in through the **Step** parameter to suit your requirements. It would be tedious if you could only move along the world-axes. Quite often you must move around an object or in the viewing direction of the camera to get the best viewpoint and it would be very time-consuming to trace this movement along the world-axes. Therefore, in the **"Camera Movement"** box there are three buttons with which you can determine the movement-type:

- **movement along world-axes:** If the **<World Axes>** button is selected, you can move along the X, Y, Z-world-axes - up, down, forwards, backwards, right or left.
- **movement along camera axes**: If the **<Camera Axes>** button is selected, then movement is executed along the camera axes. For instance, if you have previously positioned your camera so that it is inclined to the left at a certain angle and you now operate the movement-arrow for the movement to the left, the camera - moving on the camera-axis inclined to the side - therefore moves to the lower-left. The same is true for all other movement directions. If, for example, you turn the camera on its axes to line it up on an object, then the camera automatically points in the direction of the object. Therefore, in camera-axes mode you need only operate the arrow button for movement along the Z-axis and the camera moves directly toward or away from the object.
- **circular movement about an object:** Select the **<Circular>** button if you want to move in an orbit about an object. The relevant object must previously have been marked in the Select Object dialog. If no object has been marked the camera moves in an orbit about the world center. Choosing the X-arrow buttons causes a horizontal circular-movement of the camera. Similarly, choosing the Y-arrow button causes a vertical circular movement, during which the camera lens always remains lined up on the object.



Zoom

With the "Zoom" parameter you can directly influences the "width of focus" of your camera. By increasing or reducing the "zoom" parameter you can implement wide-angle or telephoto-effects respectively. As with a real camera, there is perspective distortion of the picture.

Lining Up the Camera

The **"camera" <Focus>** button lines up the camera on marked objects, 138 turning the camera so that it points at the center of the objects. This function is also available in all other work modes - just select the corresponding entry in the main popup menu 133 or simply press the "c" key on your keyboard.

Center the Camera

By choosing the **"camera" <Center**> button the camera leaps directly to the center of the selected objects. If you are then within a closed body it is advisable to move it a little way from the object.

10.2 Start Rendering of an Image or Animation

Render Scene 259 - Camera view displayed in the render window
Render Scene Animation 259 - Preview animation in camera view

There are four entries under <Render> in the menu bar to start a calculation of an image or an animation. For fast access, four buttons corresponding to these functions can be found in the button strip directly above the viewports. You can start picture calculations at any time in each work mode simply by pressing one of these buttons.

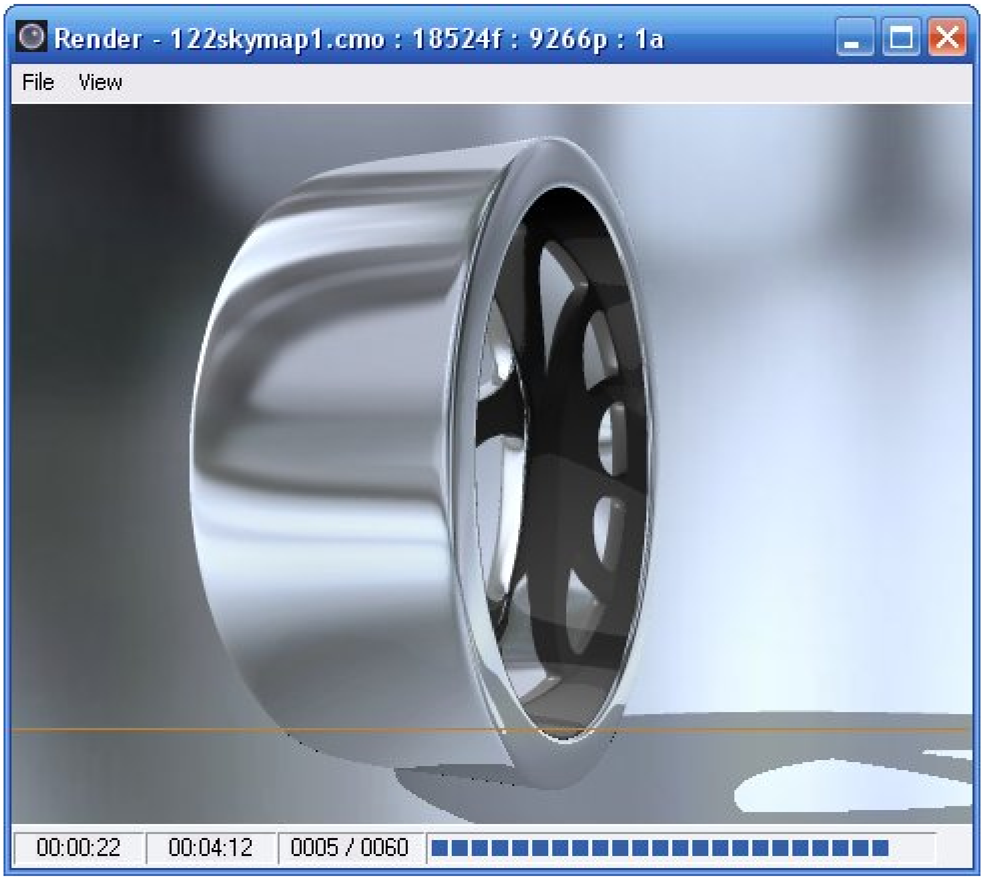
Render Scene (Short cut 'S')
The <Render Scene> button causes a re-draw of the scene in the render window and does not differ from the output in the camera viewport, except that now the size of the rendered picture is adapted to that entered in the Render Options dialog 263. The picture size can be independent of the maximum screen-resolution and independent of your current viewport size. However, if you render a picture in which the length/ height relationship differs from the length/ height relationship of the viewport, details of the picture rendered for the file are automatically somewhat different from than those shown in the viewport.

Render Final (short cut 'F')
When you operate the <Render Final> button your picture is calculated with the high quality raytracing 264 or global illumination 264 algorithm with regard to the settings planned in the material- 290, light 341 - and Render Options dialog 263. Depending on the hardware available, the complexity of the scene and the depiction mode used, the calculation of the picture can take from a few seconds to several hours.

Render Scene Animation (short cut + 'S')
This corresponds to <Render Scene>, except this time a whole animation is calculated. An entire animation with many individual pictures and a complex scene can conceivably require several hours to days to render, depending on the speed of your computer and the rendering parameters (resolution, antialiasing, reflection etc). You can try out the settings for the materials and the background lighting simply by rendering a control-picture. What a waste of time it is if, after a day of animation rendering, you find that the chosen camera or object movement is not what you anticipated. A simple preview rendering would have been helpful here.

Render Final Animation (short cut + 'F')
The final rendering process for a whole animation is started.

The Render Window
The rendering is made in an external render window with its own menu bar.



The render window in action - from the left to the right you can see in the status bar: render time | approximated remaining time | current frame number and remaining frames | and the progress bar for the current rendered picture. The caption of the render window provides additional information about the number of facets, points and analytical primitives generated to render the current frame. This way you can track how the total number of facets and points changes from frame to frame when applying particle systems or SDS-Deformation.

Saving Pictures or Video Files

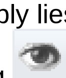
Pictures can be saved after rendering with help of the menu function "File - Save Picture as..." (as .bmp, .jpg, .png, .pcx, .tif or .tga files) - animations can be saved via the corresponding menu entries as AVI-videos or picture sequences. If the rendering of additional alpha- or depth channels was activated, additional menu entries are provided to save pictures with integrated alpha- or depth channels. Optionally, alpha- and depth channels can also be saved to separate greyscale images. Even animation renderings with activated alpha- or depth channel can be saved to separate greyscale videos or greyscale picture sequences. See also "Render Options - Rendering of Alpha- and Depth Channels 271"

The AVI-Format - Compression of the Video File

The AVI-video file is not so much a file format on its own but rather a container which can store video formats of all sorts. For instance, a video file ending with the ".avi" extension can hold a Windows-media file (.wmv) but also an mpeg or any other file format. A small dialog appears before saving an AVI file where you can select one of the video codecs currently installed on your computer system (A few codecs are included in each Windows installation, but you should search the internet for more efficient alternatives). Video codecs can compress AVI-Videos to a fraction of its original size - but be cautious. The reduction in size almost always goes along with a loss of quality that cannot be restored. Generally you have to experiment widely with different codecs and settings to achieve the best compromise between quality and size. Therefore, I recommend that you alway save your final complex animations uncompressed as 24 bit RGB videos and experiment afterwards in video processing software with the individual codecs.

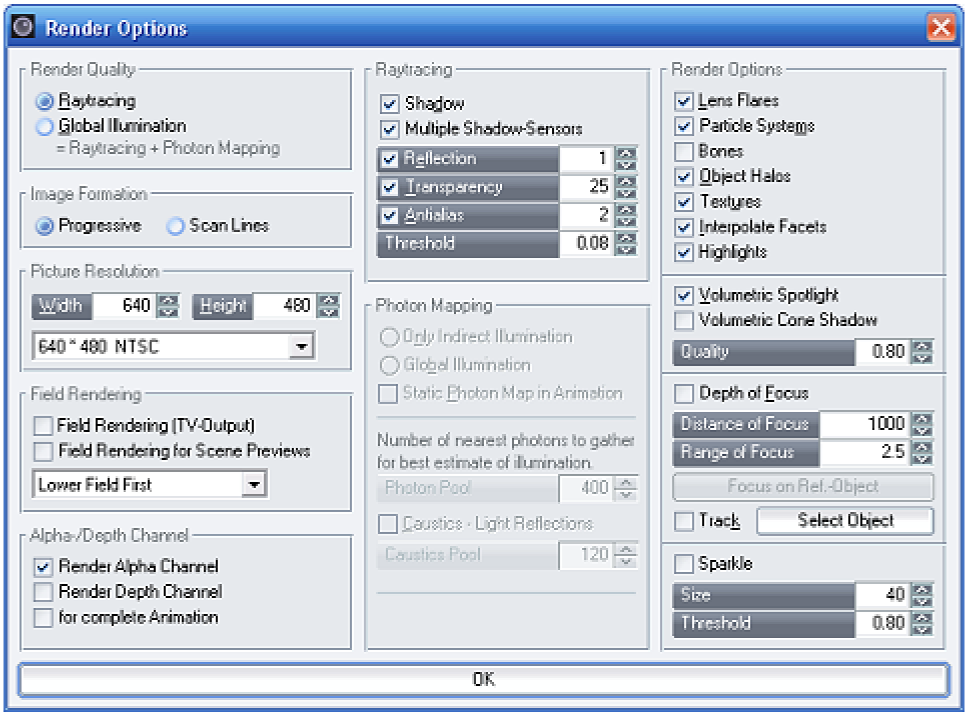
Saving Individual Pictures of an Animation - After completing the rendering of an animation the calculated video will automatically start to play in the render window. You can even pick out individual pictures of the film by stopping the video at the required points and then saving the window content again via "Save Picture as....".
A new picture calculation overwrites the last rendered picture but not the last rendered animation. The menu entry "File - Show animation" will get back the animation in the render window. Only a new animation calculation will overwrite the old one.

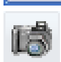
Interrupting Picture Rendering
You can interrupt the rendering process by pressing the "ESC" button, via the menu function "File - Stop Rendering" or just by closing the render window.

Show Last Rendered Picture or Animation
When the render window has been closed, minimized or simply lies below the main window you can restore the render window and its content simply by selecting  in the button strip or choosing the "File - Show last rendered picture/animation" entry in the menu bar (of course minimized windows can be also restored by selecting the corresponding button in the Windows task bar). To bring the main window to top again, just press the "ESC"-key.

Data Recovery after a System Crash
Interrupted working session (e.g. by a power failure or a system crash) will be recognized on next startup of MR-3D Designer, and can be recovered from the temporary undo files created during the previous session (if they have not been deleted in the meantime). A dialog opens where you can choose to continue your work at the point of interruption, or clean up all temporary files. With this function, you can even recover partly rendered animations. If, for instance, you started an animation rendering comprising of 400 frames and while rendering the 200th frame the computer dropped-out, then simply reboot the computer, start MR-3D Designer again and recover the old session. Then choose the "File - Show last animation"-menu entry of the render window to retrieve the already rendered part of the first 200 frames of the animation. Save this part of the animation via the corresponding "File - Save as..." menu entries. Then render the remaining part of the animation using the start- and end frame parameters of the animation editor. Finally combine both parts using third-party video processing software.

10.3 Render Options



 - Menu "Options - Render Options" - Short Cut: ⇧ + "R".

In this dialog you can plan picture resolution and settings relevant to the realism of the scene.

- **Render Quality** 264
  Choose a rendering algorithms to determine the output quality of your picture calculation:
  Raytracing 264 - High quality, real reflections, transparencies with refraction and shadows
  Global Illumination - Raytracing and Photon Mapping 264 - With the photon mapping technology the diffuse indirect illumination in a scene and caustic light reflections can be included in the calculation of the picture
- **Image Formation** 269
  By using progressive rendering, a good impression of the final picture can be seen in seconds. Normal scan-line rendering has advantages when rendering animations
- **Picture Resolution** 270
  Select a standard picture format or specify your own resolution
- **Field Rendering** 270
  Field Rendering for interlaced video
- **Rendering of Alpha- and Depth Channels** 271
  Use the information of alpha- and depth channels for compositing exported pictures and animations in corresponding video processing software.
- **Render Options** 273
  Use this global buttons to switch on or off special effects
  Lens Flares 273
  Particle Systems 273
  Bones 273
  Object Halos 273

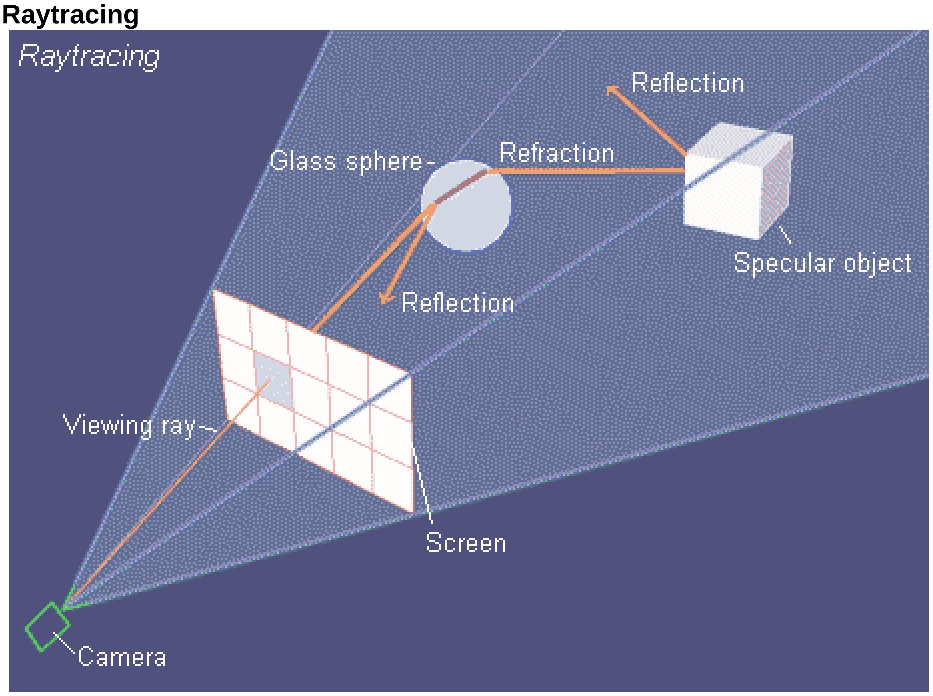
Textures
Interpolation Facets
Highlights
Volumetric Spotlight 273
Depth of Focus 273
Sparkle 273

10.3.1 Render Quality

In MR-3D Designer always the high quality **Raytracing** algorithm is used for the final picture calculation. It offers high quality and can simulate real world phenomena like mirror reflections, refractions in transparent objects and shadows. Above the Raytracing stands the **Global Illumination** algorithm. The implemented photon mapping technology combines the pros of raytracing - reflection and refraction - with the ability to render also the indirect illumination caused by diffuse reflections in the scene. At the top left of the Render Options dialog you can choose between these two rendering options.

- **Raytracing** 264
- **Global Illumination - Raytracing and Photon Mapping** 264

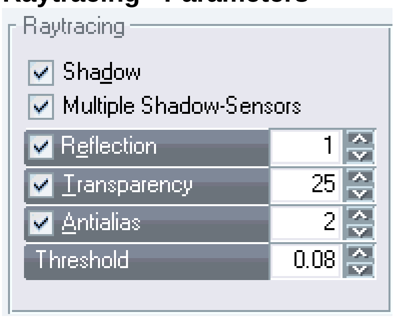
Raytracing



In nature, a light-ray leaving a light-source reflects off different objects and then falls, sometime later, into the eye. Raytracing is exactly the reverse process. A "viewing ray" is sent out from the camera-viewpoint, through a projection-plane (the screen), and then tested for an intersection with an object in the area. If an intersection occurs the relevant pixel of the screen - at the point where the viewing ray passes through the projection-plane - can, therefore, be drawn in the calculated surface-intensity of the object. The viewing ray can also be followed further, however. At a mirroring object the angle of reflections is simply calculated and a new ray started searching. The surface intensity for the first object met then comes recursively from the incident light in addition to the light

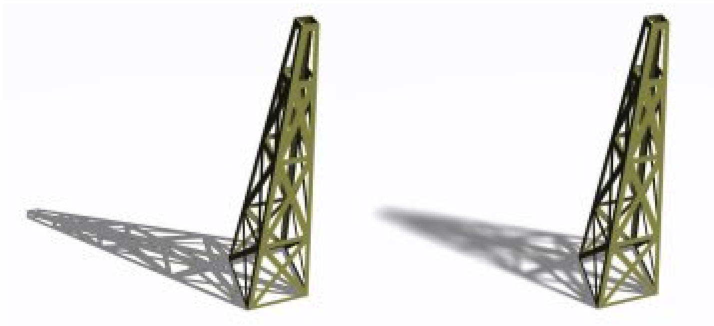
reflected from the other objects that are met.
With this algorithm it is also possible to calculate the split in the viewing rays that is necessary with transparency. In order to render transparency realistically the surface-reflections are calculated from reflections in addition to the intensity of the light coming from behind. A mirror-ray and simultaneously a split viewing ray is calculated. If these viewing rays meet another transparent or mirror object, the whole process is started again, so the color for rendering only a single point on the screen results from a whole collection of viewing rays.
The recursion levels for reflections and split rays can be entered separately to keep the rendering time within reasonable limits.

Raytracing - Parameters



Shadow - Select this button if you wish the picture to include shadows in the rendering. When calculating the light intensity for a surface, then first a light ray from the surface point to the light source is checked to see if there are any objects - or the object's own facets - between the light-source and the point being rendered. If there is, the calculation is broken off for this particular light source. If a transparent object casts a shadow then the light color is filtered by the objects surface color.
Rendering time increases greatly with each light source, especially if soft shadows (requiring multiple shadow sensors) are calculated. If an individual object is not required to cast a shadow you can save rendering time by using the <No Shadows 290> function for that object in the Object Properties dialog. Furthermore, shadow-calculation can be switched off for each light-source using the option <No Shadows 351> in the light dialog 341.

Multiple Shadow-Sensors - Soft Shadows - Switch on this option if you want to generate a soft-shadow effect. Standard light sources such as a lamp, sun and spotlight can be defined with a specific radius so that, instead of originating from a single point-light-source, the light comes from a spherical area light-source. You can enter a number of shadow-sensors 351 in the light dialog 341 for each light type . If this value is 1, a completely normal hard-edged shadow is rendered. If you use a value greater than 1 (say 20 for example), this number of shadow-sensors are used to scan the area of the spherical light area to determine how much of the light-sphere is hidden by other objects. The results of the shadow-sensor evaluations are interpolated so a soft shadow is rendered. A similar approach is used during Ambient Occlusion 343 rendering, where multiple shadow sensors are used to scan a hemispherical bounding volume to test for shadow casting caused by diffuse ambient light incidence.
The greater the number of shadow-sensors, the better will be the soft shadow effect. With smaller numbers of shadow-sensors the shadows become increasingly noisy. However, the number of additionally-calculated shadow-sensors should be kept as low as possible due to the multiplied rendering time required. By using the global switch <Multiple Shadow-Sensors>, you can deactivate rendering soft shadows until your scene is completed and you are ready to render a final image with the highest possible quality.



The picture above on the left shows a framework with a solid shadow and next to it the same scene rendered with soft shadows.

Reflection - Surface reflections are included in the picture calculation. 1 to 25 reflections can be entered, which determines the maximum number of retraces in the scene. However, in most cases a mirror-depth of 1-2 is sufficient. Too many reflections can produce quite a confusing reflected image and causes the rendering time to climb steeply. The same is true for the transparency effect described following.

Transparency - Transparent objects will be calculated with real refractions. The maximum number of retraces can be entered from 1-25 - as with the reflections.

Antialiasing - A low-screen resolution of, say, 320 x 200 pixels-points often produce an undesirable step effect. A line running diagonally looks more like a stairway than a straight, clean line. Antialiasing smoothes this effect. With it, each screen pixel to be rendered is divided into several smaller sub-pixels and a search ray is calculated for each sub-pixel. An average intensity for the screen-point is then calculated from the intensities of all sub-pixels.
Steps that depart slightly from the actual line are now plotted using the averaged intensity and altogether the impression of a slightly blurred but more representative line is produced.
The quality, and with it the number of sub-pixels to be calculated, can be set from 1-4:

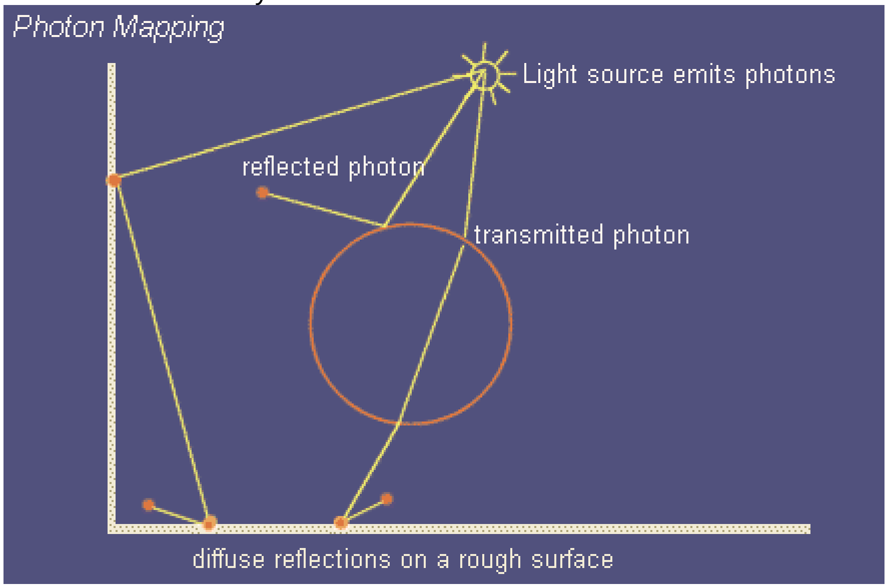
- 1 = 4 sub-pixels
- 2 = 9 sub-pixels
- 3 = 16 sub-pixels
- 4 = 25 sub-pixels

Antialiasing is only applied to certain pixels so the rendering time does not also rise by 4, 9, 16 or even 25- times because so many more pixels are calculated. It is only those pixels that deviate in their color-value by a **threshold** value from that of their neighbors that are subdivided into smaller pixels. This threshold value can be set via the corresponding parameter in the raytracing box. It is also possible to set this value to zero, thus every single pixel will be subdivided into sub-pixels and rendering time increases considerably. This should only be applied for a final, high-quality rendering, when, for instance, severe antialiasing effects arise. Don't forget to reset the value afterwards to a common value about 0.8.

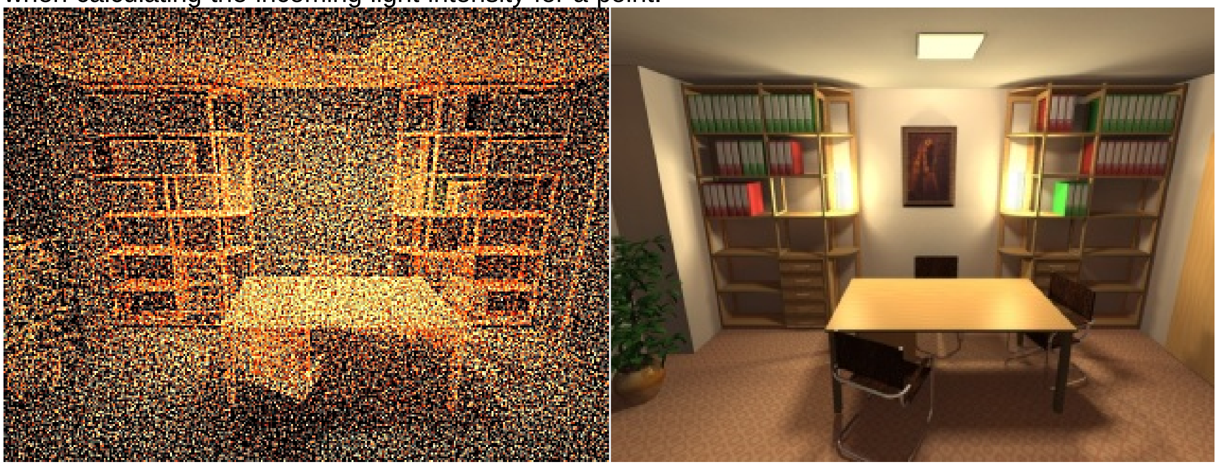
Negative Antialiasing - A negative value (-1) can also be input for antialiasing. In contrast to calculating additional intermediate pixels for positive antialiasing, negative antialiasing departs from this process. Instead of more pixels, only half the pixels to represent the rendering are calculated. The color-values for those pixels that are not calculated are interpolated from the surrounding pixels, so that halving the calculation resolution does not reduce the resolution too severely. Advantage: Up to 200% speed-gain. In this way it allows a lot of quick control-pictures to be calculated and the alignment of shadows, reflections and illumination to be checked.

Global Illumination - Raytracing + Photon Mapping

Raytracing is a standard for high picture quality and for realistic reflections and refraction. One of the major drawbacks in a general raytracing implementation is that it does not take into account indirect illumination - the light that is reflected from other objects in the scene other than the direct light from a light source. Usually an ambient light intensity 343 is defined to simulate this indirect lighting, but that is a very poor approximation. Especially in architectural scenes, the illumination in a room is dominated by indirect light reflected many times from the diffuse surfaces in a building. With its photon mapping algorithm, MR-3D Designer provides a global illumination model that combines the pros of raytracing - reflection and refraction - with the ability to also render the indirect illumination caused by diffuse reflections in the scene.



Rendering a picture with photon mapping is a two-pass procedure. In the preliminary run little packages of energy (photons) are emitted from the light sources in the scene. Similar to ordinary raytracing the path of these photons is traced through the scene and the distribution of photons is saved in a three-dimensional data structure called the photon map. After the photon map has been calculated the picture is rendered in an ordinary raytracing run and the photon map is evaluated when calculating the incoming light intensity for a point.

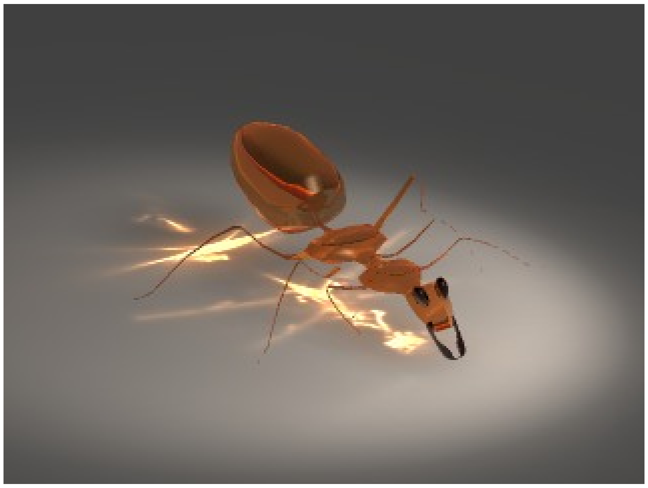


The picture above shows a visualization of a photon map and the final picture rendered with help of this light map. Each pixel set in the picture on the left corresponds to a photon saved to the photon map.

There are more advantages of photon mapping:

- Color bleeding - for instance, when a green wall casts greenish reflections on a neighbouring white wall.

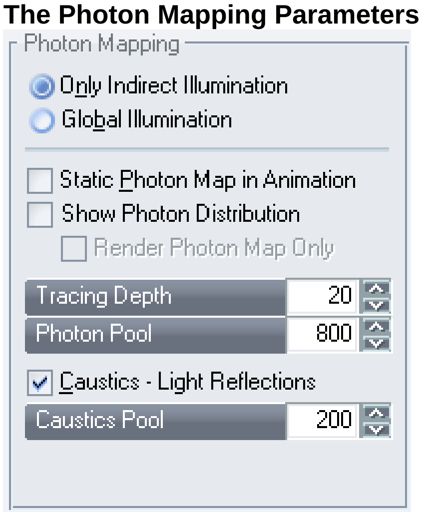
- Caustics 370 - Caustics are light reflections from highly specular surfaces or, e.g., the light gathered in a focal point after transmission through a glass lens.



Example for caustic light reflections beneath a little glass figurine

For a deeper understanding of photon mapping see also:
Photon Mapping 277 - Introduction and examples
Light Dialog - Photon Emission Parameters 370
Material Dialog 290 - Photon mapping object properties

The Photon Mapping Parameters for the Evaluation Process in the Raytracing Pass



You can make use of photon mapping in two different ways:

Only Indirect Illumination - Only the diffuse indirect light that has been at least once reflected in the scene is evaluated from the photon map. Then the incident light coming directly from the light source is calculated and combined with the indirect light.

Global Illumination - All the illumination in the scene is calculated by evaluating the photon map only. You can read about the pros and contras in the chapter - Photon Mapping - Introduction and Examples 277.

Static Photon Map in Animation - This causes the photon map to be calculated only once - at the beginning of an animation. You can use this option to animate a fly-through of architectural scene where the objects themselves do not move. It is also possible to exclude individual objects from the photon mapping process - they will be illuminated only with direct lighting. You can even render animations with a static photon map and moving objects, e.g. a car driving through a street lined with houses.

Show Photon Distribution - The photons emitted to the scene and saved into the photon map can be displayed during the preliminary run of photon distribution. This will slow down the photon emission process a little bit but on the other hand you get a good impression of the photon density and distribution in the scene.
Render Photon Map Only - If this option is also switched on, then only the photon map will be rendered. Now you can examine the photon distribution at ease or save it in a picture file. In combination with a static photon map you can even animate a flight through a photon map.

Photon Mapping - Tracing Depth - You can specify a maximum number of reflections for the tracing of photons through the scene. With a higher number of reflections even the remotest corners of a room will be reached and lit by photons, whereas a smaller number of reflections will provide a smoother shading, because more photons are saved on those surfaces hit by the main stream of photons.

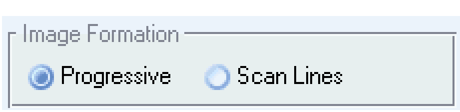
Photon Pool - A specific number of photons scattered around a point on a surface has to be gathered in the so-called photon pool for a good estimation of the incoming irradiance for that point. The maximum number of photons to search for the photon pool determines the quality, sharpness and of course the rendering time. You should take into account that searching through a photon map that might hold millions of photon entries is no trivial thing. Appropriate values for the photon pool size lie between 250 and 800 photons when photon mapping is applied only for the indirect lighting, and 500 up to 5000 photons, when Global Illumination mode is activated - with photon maps containing several million photons. As a rule you can say that the more photons in the photon map, the more photons should be gathered in the photon pool - otherwise the pictures have a somewhat spotty appearance. For examples refer to the chapter - Photon Mapping - Introduction and Examples 277.

Caustics - Light Reflections - Includes the calculation of caustic light reflections in the photon mapping process. Caustic photons are saved in a separate photon map - the so-called caustic map. This is due to the different demands for global photons (soft indirect light) and caustic photons (sharper contours created by specular reflections and transmission).

Caustics Pool - For the caustics pool you do not need as many photons as for the global photon pool. Caustic reflections often have sharp contours (e.g. light transmitted through a lens and focused in a sharp focal point) and with too many photons the caustic reflections would become blurry. Good values lie between 80 and 250 photons, depending on the number of photons contained in the caustic map.

For more examples see also:
Photon Mapping - Introduction and Examples 277
Light dialog - Photon Emission Parameters 370

10.3.2 Progressive vs. Scan Line Rendering

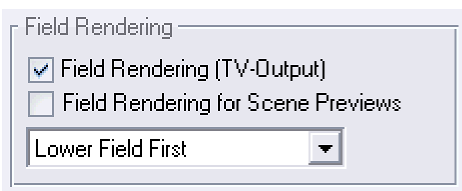


You can choose between a line-by-line picture construction (Scan Line) or a progressive procedure. In progressive mode several rendering passes are applied, first approximating the picture with a low resolution of large pixel blocks, then constantly refining the picture until the final resolution has been achieved. After that, in a last rendering step all the supersampling for the antialiasing is processed. Progressive Rendering can considerably facilitate the work on complex scenes, because you can get a pretty good impression of the outcome of the picture after only a few seconds - even if the final rendering will still require hours. This way you could decide in good time whether you want to continue the rendering or stop it to make further changes to the scene or rendering parameters. Line-by-line rendering has its advantages when rendering animations. When the last frame of the animation is overwritten line-by-line with the new picture it is much more easy to follow the movements in the film.

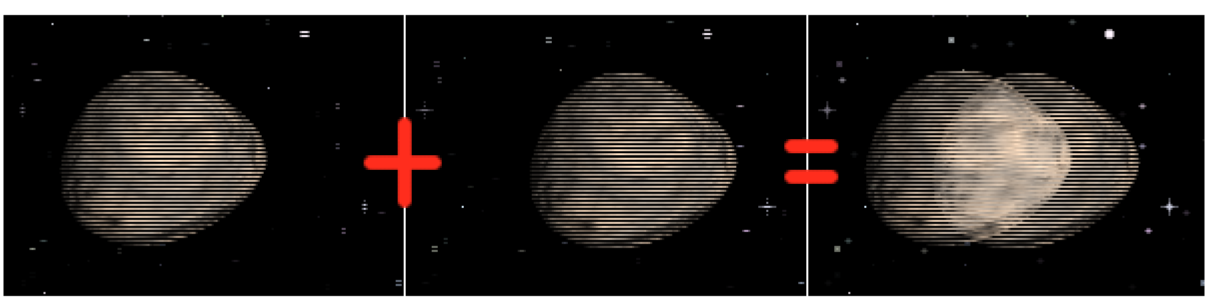
10.3.3 Picture Resolution

In this box you determine the resolution at which the picture is calculated, independent of the display window. This permits calculation to any resolutions from 32 x 32 to 12000 x 12000 pixels. Please remember that technical complexity is not necessarily very meaningful. An extremely complex picture that at a resolution of 640* 480 pixels requires a rendering time of, say, 10 minutes, will, at a resolution of 6000* 6000 points - corresponding to 117 times the number of pixels - require almost 20 hours rendering time. The same is also true for the required storage area. A picture of 640* 480 pixels with all effects switched on needs 2.5 MB while rendering for the picture and effect-buffers (maximum 9 byte per picture-point, if all effects like light reflections and depth of focus are switched on). At a resolution of 6000* 6000 pixels would require memory space of about 300 MB. You can choose from some preset standard resolutions in the list box or the resolution can be changed to meet specific requirements at any time through the keyboard.

10.3.4 Field Rendering



Interlaced video is used in regular TV. An interlaced video picture contains two fields of picture information shot at different times. In the first shot the picture information is saved to all odd numbered scanlines (1,3,5...) and in the second shot all even numbered scanlines (2,4,6...) are saved in the same video frame. When playing this video on TV, both fields are played in succession to produce the interlaced TV-picture. So, when watching television you always see only one half the strips of a picture - it's the playing frequency and the luminous characteristics of a television screen that gives the impression of full-frame pictures. If you plan to play your MR-3D Designer animations on TV you can now switch on Field Rendering for AVI-output, too. Since twice as many pictures (each of half resolution) are rendered, Field Rendering gives smoother motion and can even reduce or eliminate the need to render motion blur - which can save rendering time. But this applies only to TV-output, Field Rendering is not suited for output on computer monitors!



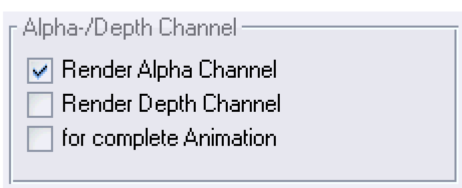
Example: The illustration shows a meteor crossing the picture from the left to the right. In the first picture only the odd numbered scanlines are rendered. Then - a moment later in time - a second picture is rendered containing only the even scanlines. Now both pictures are interlaced and saved as one single picture (illustration on the right). When playing the film on a TV, the picture information is separated again by the hardware and both fields are played in succession again, to provide the smooth movements for the television screen.

Sometimes, video cards demand the reverse order of field data. Read your video card directions and switch from "Lower Field First" rendering to "Upper Field First", if your card displays the even scanlines before the odd scanlines.

Field Rendering and Scene Previews
With the "Render Scene Preview" function you can render fast preview animations that use the same depiction mode as set for the camera viewport. These preview animations can be saved in AVI-video files in the same way as final animation films. You can also activate the field rendering output for these preview films, if you select the "Field Rendering for Scene Previews" button.

Compressing Videos
CybeMotion films are always saved as a 24 bit AVI (high quality, uncompressed) video file. These uncompressed (and therefore without quality loss) video files can become very large but can easily be converted to compressed video files using third party (free- or shareware) converter programs. You should always try a variety of different compression-algorithms/encoders before deleting the original uncompressed file because output quality and compression rates differ widely. Compressing a video containing interlaced field data is even more complex, since the compression algorithm has to take into account that each picture is composed of two time-shifted half-pictures.

10.3.5 Rendering of Alpha- and Depth Channels



Render Alpha Channel - In MR-3D Designer you can activate the calculation of an alpha channel 328 in addition to the rendering of the picture information. Alpha channels are used for compositing tasks. You could, for instance, render a picture of a 3D logo with activated alpha channel. Later, using a third party image editor, you can use the alpha channel information to copy only the shape of the 3D logo into the background of another picture, maybe a photograph or the background of a website. The alpha channel holds an 8 bit greyscale tone defining a transparency value for each pixel of the rendered image. Background areas or the rendered picture are saved in black to the alpha channel (full transparency) and object areas in white (opaque). Transparent objects and aliased transitions from object to background areas are saved in grey tones.

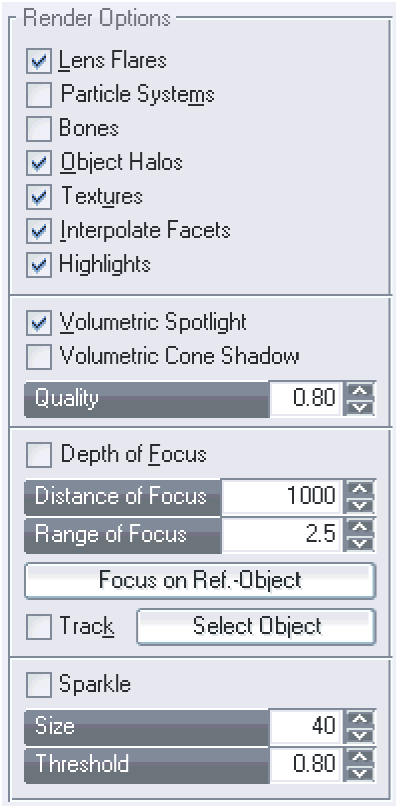
Render Depth Channel - A depth channel contains the depth buffer of a rendering, i.e., the distance of an object from the camera. Similar to an alpha channel, the distance values are converted to an 8 bit greyscale image which can be included in an 32 bit RGBA picture when saving the image (see above). Some image editors or even video processing software can make use of this information to apply special postprocessing effects, such as depth of focus or fog.

The picture above shows the rendering of a car (model: mr-clipart.com) with corresponding alpha channel (picture in the middle) and depth channel (picture on the right).

Activating and Saving Alpha- or Depth Channels
To include the render of an alpha- or depth channel you have to switch on the corresponding button **<Render Alpha Channel>** or **<Render Depth Channel>** in the Render Options dialog. Then - after picture calculation - you have several additional options for saving of an image. Via the "File" menu of the render window you can choose to save an ordinary 24 bit RGB-picture, or a 32 bit RGBA-picture with included alpha- or depth channel (supported file formats are .tga, .png and .tif) - or you can even save the alpha- and depth channels in separate grey scale images.

Rendering of Alpha- and Depth Channel Videos
Most video processing software can handle animated alpha- or even depth-channel videos. If you select the **<for complete Animation>** option in the Render Options dialog, MR-3D Designer will render additional animated alpha channel or depth channel data in separate data buffers. After the rendering you can choose again via the "File" menu of the render window which kind of data you want to export. You can either save the video with the color information, a separate alpha- or depth channel video containing only the greyscale data, or you can save the whole video as a picture sequence - here again as 32 bit RGBA, only RGB or only alpha/depth channel images.

10.3.6 Choose Rendering Effects



Global switches for additional rendering effects are found on the right side of the render options dialog.

Lens effects

A picture taken with a real camera can be flawed by camera-optics. If, for example, a light-source 340 shines directly into a camera, several reflections can be produced within the lens. This is responsible for the well-known star, circle, or annular light reflections 365 in pictures. The photo and film-industry, however, do not always view this effect with pleasure and go to considerable lengths to avoid it. On other occasions, however, these effects are deliberately aimed for and highlighted. In the meantime, everyone has become so accustomed to these picture faults that, by using these effects, you can considerably increase the impression of realism in computer-generated pictures.
You can determine if and how each individual light source produces lens reflections using a variety parameters in the Light dialog 341.
Here, in the picture-parameter dialog, the calculation of these light effects can only be switched on or off globally with the <Lens Flares> button.

Particle-Systems

If you have defined particle actions in the particle-editor 425, you can specify with the **<Particle-System>** button, if the particles are calculated for (Preview-) animations or not.

Bones

Usually, the bones of a skeleton are only visible in the viewport windows - in the final rendering skeletons are hidden automatically. Just switch on the <Bones> button in the Render Options dialog if you want bones to also be included in the final rendering of a film then.

Object Halo

The <Object Halo> button switches the halo effect on or off for all objects simultaneously. This effect creates a halo of light enveloping the outline of an object and can be used to simulate atmospheric halos around planets or, for instance, to create swirling swarms of glowing particles. The parameters for object halo are edited in the material editor [309].

Textures

If this button is selected the object's material textures (procedural or bitmap textures) that have been defined in the materials dialog [300] are included on rendering. For fast previews you can, therefore, switch the textures on or off, independent of whether the texture for the object is switched on in the material dialog.

Interpolate Facets

Facet interpolation smoothes the shading of the surface of all objects with the material attribute "Interpolation [290]".

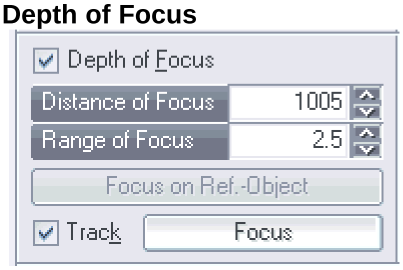
Highlights

All objects having a shiny and reflective material [309] are rendered with highlight-reflections from light sources.

Volumetric Spotlight

Switch on the option **<Volumetric Spotlight>** to render visible spot light-cones [354]. If you also activate **<Volumetric Cone Shadows>** then shadow silhouettes of objects intruding into the light cone are included in the calculation, as well as the filtering of light through transparent objects. Rendering of volumetric shadows is very calculation-intensive, therefore you can decide for each spot lamp individually if it is to be rendered with a visible light cone or volumetric shadows, respectively. You can adjust these settings in the light dialog [354]. However, the quality of the volumetric shadow calculations is a global parameter, which is defined here in the Render Options dialog:
Quality - This value defines the accuracy of the light-cone scan. A volumetric approach is applied to render samples within the light cone, which accumulate to an interpolated intensity value. The higher the Quality value, the more samples are calculated and the higher the quality with correspondingly increasing rendering time. The quality also depends on the spread-angle of the spotlight. A wide spread signifies a wide spot cone so the scan rate must also be higher if you want to obtain reasonable results.

Depth of Focus

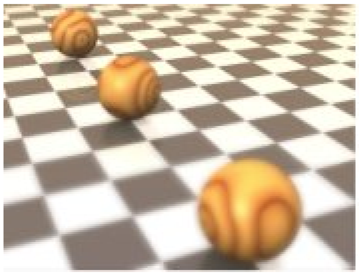
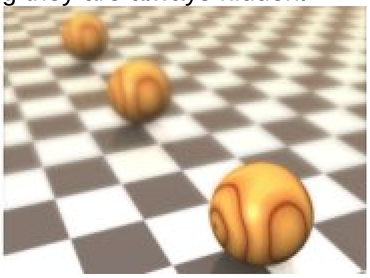


Depth of focus is another effect that is used in film and photography to accentuate areas of a picture. With depth of focus switched on only picture-regions at a specific distance are sharp, and picture-regions that lie nearer or further away become increasingly more blurred.
Distance of Focus : With this parameter you can specify the precise distance at which the picture is sharpest. If, for example, a value of 1000 units is put in here, an object that is at this precise distance from the camera is represented with completely sharpness. Objects that are before or

behind this object are represented less sharply with increasing distance from this point.
Range of Focus : With this parameter you can decide how great the area of focus will be, with respect to how rapidly the sharpness falls off with increased distance from the focus point. With a very small value (minimum 1) only a very small area about the focus is sharply represented and the sharpness falls off rapidly. With a greater value you can increase the area of sharpness and reduce the rate at which the sharpness falls off.

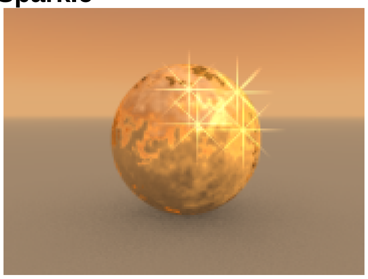
Focus on Reference Object - It is not necessary to work out the relationship between the camera and an object in order to determine the precise focal distance. Simply mark the object you want to focus as a reference object 138 in the viewport window and then, back in the Render Options dialog again, operate the <Focus on Ref.-Object> button. The distance of the camera from the reference object is automatically calculated and noted.

Animated Depth of Focus - If you switch on the <Track> button then the program automatically calculates the focal distance to a reference object. The reference object can be selected in an object selection box and its name appears when you select the button next to the <Track> button. In each frame of an animation the reference object will be used to calculate the sharpness distance. A Group Object 192 is best suited for this purpose, since Group Objects are only drawn in the viewport windows - in the final rendering they are always hidden.

Example: In the left picture the sphere lying at the back was chosen as the focus, while in the right picture the sphere at the front was the focus.

Sparkle



Sparkles are star-like lens reflections that appear if light reflects from a shiny object into the lens of a camera.
Size : The basic size of the sparkles can be entered - the size at which they are rendered, however, is dependant on the distance and also intensity of the shine.
Threshold: Here you can set a threshold value between 0.1 and 1 that corresponds to the minimum intensity of the light that must be reflected from an object before sparkles are calculated.

Part XI

11 Photon Mapping - Introduction and Examples

All about global illumination using photon mapping

Overview - where do I adjust which parameters?

Render Options 264
- Choose the rendering algorithm - photon mapping only for indirect illumination or as a global illumination model
- Static photon map in animation
- Number of photons to gather for the global photon pool and the caustics pool
- Global photon pool and caustics pool - number of photons to gather for averaging the illumination in a surface point

Light Dialog - Photon Emission Parameters 370
- Exclude light source from photon mapping - use only direct light instead
- Number of photons a light source emits
- Photon intensity correction
- Area lights and emission direction of facets

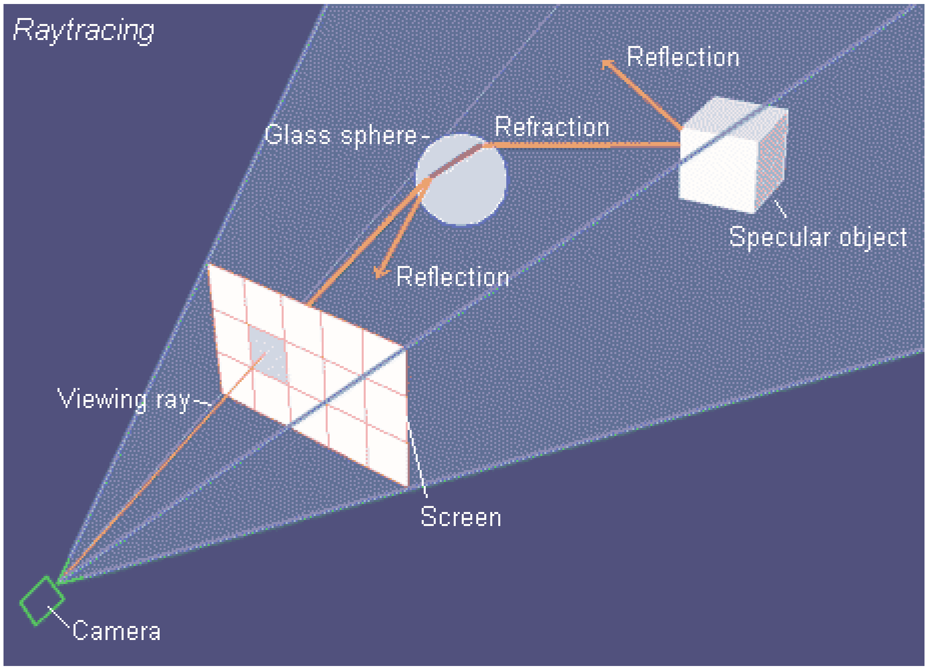
Material Dialog - Object Properties 290
- Caustics - aim additional photons at objects that cast caustic reflections
- Exclude object from photon mapping - illuminate directly instead

11.1 Global Illumination and Photon Mapping

The physically based simulation of all light distribution in a virtual 3D-model is called global illumination. A global illumination algorithm should take into account all interactions of light with the different surface materials in a scene. That sounds alright in theory, but the nature of light - having both the properties of a electromagnetic wave and a particle, is much to complex to include all phenomena in a correct physical simulation. Anyway, what counts is only the individual perception of light, since we can only "see" a limited spectrum of light and the interpretation of what we see is a mainly physiological sensation.

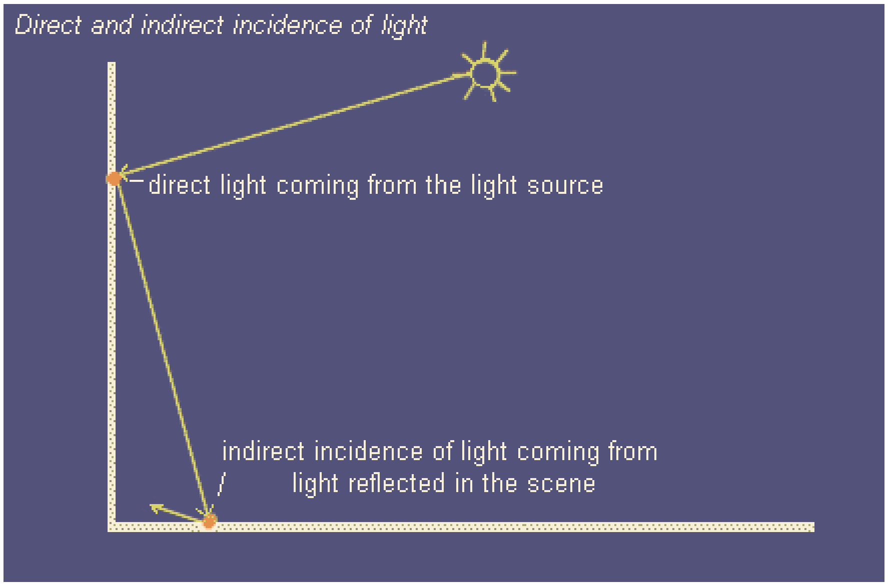
The global illumination algorithm implemented in MR-3D Designer is based on a combination of photon mapping with conventional raytracing. So, let's have a look at the raytracing algorithm first.

Raytracing

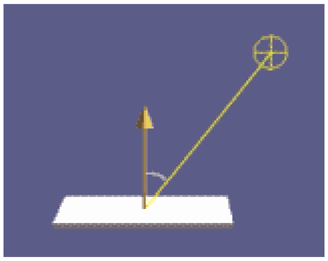


In nature, a light-ray leaving a light-source reflects off different objects and then falls, sometime later, into the eye. Raytracing is exactly the reverse process. A "viewing ray" is sent out from the camera-viewpoint, through a projection-plane (the screen), and then tested for an intersection with an object in the area. If an intersection occurs the relevant pixel of the screen - at the point where the viewing ray passes through the projection-plane - can, therefore, be drawn in the calculated surface-intensity of the object. The viewing ray can also be followed further, however. At a mirroring object the angle of reflections is simply calculated and a new ray started searching. The surface intensity for the first object met then comes recursively from the incident light in addition to the light reflected from the other objects that are met.
With this algorithm it is also possible to calculate the split in the viewing rays that is necessary with transparency. In order to render transparency realistically the surface-reflections are calculated from reflections in addition to the intensity of the light coming from behind. A mirror-ray and simultaneously a split viewing ray is calculated. If these viewing rays meet another transparent or mirror object, the whole process is started again, so the color for rendering only a single point on the screen results from a whole collection of viewing rays.

Direct and Indirect Illumination
Raytracing is a standard for high picture quality and for realistic reflections and refraction. One of the major drawbacks in a general raytracing implementation is that it does not take into account indirect illumination - the light that is reflected from other objects in the scene other than the direct light from a light source.



Direct light is coming in directly from the light source, indirect light is reflected at least once from a surface in the scene.



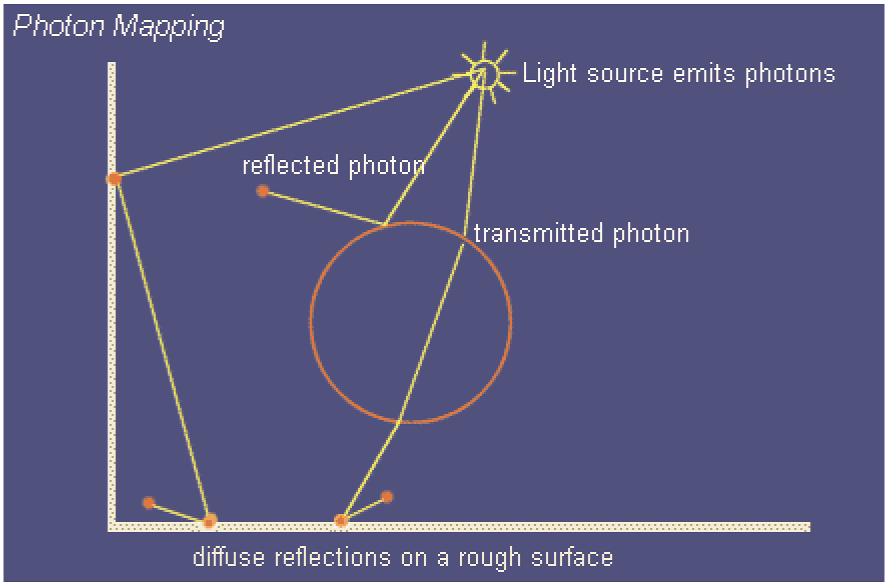
Direct light and surface brightness - In a conventional illumination model the light intensity coming directly from the light source is determined by calculating the light incidence angle between the light ray and the surface normal. This is of course only a geometrical approach, but it has produced good results for years in the computer graphics industry.

Raytracing and Indirect Illumination

Usually in 3D programs a constant light intensity can be specified to simulate the indirect illumination in non-global illumination models. In MR-3D Designer this constant general area brightness is defined with the light object "Ambient 343". Another possibility to simulate indirect light is to put additional lights with a lower intensity level directly opposed to the main light's emission direction. This will help to lighten deep shadows cast by the main light source. All in all, both methods are very poor approximations. In architectural scenes the illumination in a room is dominated by indirect light reflected many times from the diffuse surfaces in a building.

Global Illumination - Photon Mapping and Raytracing

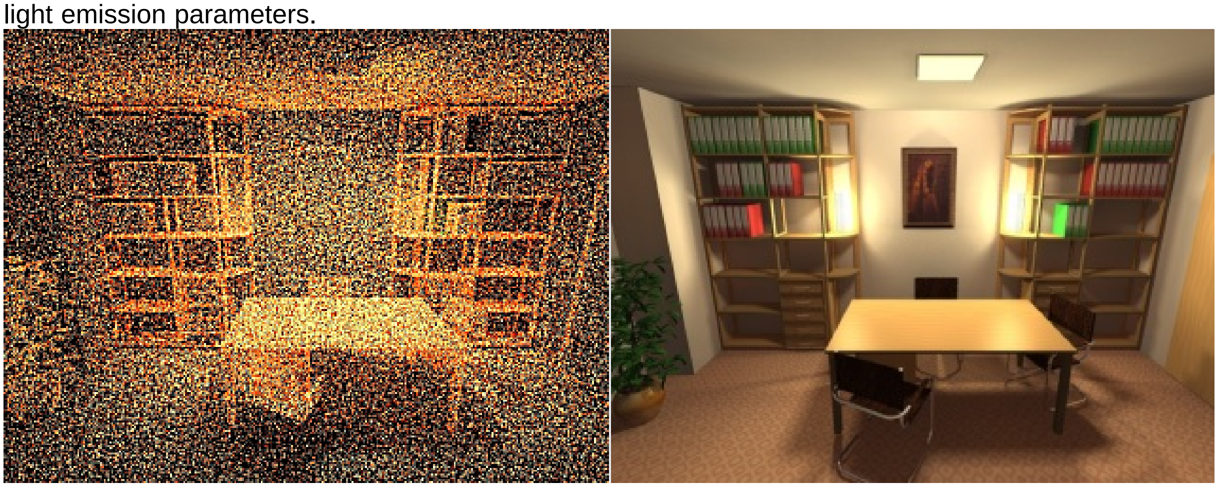
The newly-implemented photon mapping algorithm now provides MR-3D Designer with a global illumination model that combines the pros of raytracing - reflection and refraction - with the ability to also render the indirect illumination from diffuse reflections in the scene. Rendering a picture with photon mapping is a two-pass procedure. In a preliminary run small packages of energy (photons) are emitted from the light objects in the scene. As with ordinary raytracing, the path of these photons is traced through the scene, reflected from specular surfaces and transmitted through transparent objects.



Photon scattering is a more realistic simulation of light because it comes closer to the natural distribution of light from its source.

The Photon Map

Each time a photon hits a diffuse surface, the position and properties of the photon are stored in a 3-dimensional data structure called the photon map. Simultaneously a diffuse reflection is calculated and the diffuse reflected photon continues on its way through the scene until it is absorbed in the scene or lost in space. Depending on the render options, scene data and light settings, a photon map can rapidly expand to store several millions of photons in it. Therefore, be sure to have enough RAM memory available (1024mb minimum) before entering astronomical numbers of photons for the light emission parameters.



The picture above shows a visualization of a photon map and the final picture rendered with help of this light map. The photons emitted to the scene and saved into the photon map can be displayed during the preliminary run of photon distribution if you activate the corresponding option in the Render Options 264 dialog.

Evaluation of the Photon Map using the Photon Pool

After the rendering process's first pass has built the photon map, an ordinary raytracing pass is started. With the second pass, the light data stored in the photon map can be evaluated to estimate the light incidence for a point on a surface. A specific number of photons scattered around that point have to be collected in the so-called photon pool for a good estimation of the incoming irradiance .

The maximum number of photons to search for the photon pool determines the quality, sharpness and, of course, the rendering time. You should take into account that searching through a photon map that might hold millions of photon entries is no trivial thing. The size of the photon pool can be entered in the render options 264 dialog.
Appropriate values for the photon pool size lie between 250 and 800 photons when photon mapping is applied only for the indirect lighting, and 500 up to 5000 photons, when Global Illumination mode is activated with photon maps containing several million photons. As a rule you can say that the more photons in the photon map, the more photons should be gathered in the photon pool, otherwise the pictures has a somewhat spotty appearance.

Photon Mapping only for Indirect Illumination or as a Global Illumination Model

In MR-3D Designer you can choose whether you want to apply photon mapping only for determining the indirect illumination in a scene, or as a global illumination model for all of the light distribution in the scene: You can change between these two modes in the render options 264 dialog.

Photon Mapping only for Indirect Illumination

With this option the photon map is only evaluated to estimate the indirect illumination caused by diffuse reflections in the scene. The main part of the illumination and shadow calculations will be rendered using the directly incoming light, taking into account the light incidence angle between the light ray coming from the light source and the surface normal vector, as was described in the beginning of this chapter.
Advantage: You can manage with relative small photon maps when applying a photon map for indirect illumination only. Photon maps of 50,000 photons and upwards are necessary to estimate the general area brightness in a small room. You can use multiple shadow sensors 264 or insert area light objects 356 to soften the sharp shadows calculated from the direct lighting . In general, shadows calculated from direct lighting are more accurate than shadows generated from a pure photon map - when photon mapping is applied as a global illumination model. If insufficient photons are emitted towards the scene, contours of the shadows become blurry and smaller shapes in the scene may not be hit at all.
Disadvantage: The combination of the different illumination models requires an adjustment of the light intensities - on the one hand the intensity evaluated from the photon map and on the other hand the irradiance calculated from the vector pointing directly towards the light source. In the light dialog you can adjust the photon intensities via the "Intensity Correction 370 " parameter. Once an appropriate correction value has been found, you can simply adjust light intensities as usual and switch to and fro from normal raytracing to photon mapping without further adjustments of the correction parameters.

This example was rendered with photon mapping only for the indirect illumination. An area light object (NURBS patch with 49 shadow sensors) at the ceiling and 2 table lamps (36 shadow sensors each) standing in the shelves provide a warm lighting and soft shadows. For the photon mapping

process the ceiling lamp emits 100,000 photons and the two table lamps 50,000 each. The effect is a soft area brightness illuminating the otherwise dark areas beneath the table and in the shelves. You can download this project file from our internet library if you like.
.

Another example (the project file for this example is part of the installation under "../projects/volumetricfire/candles_anim.cmo"). Two candles and a third light source behind the camera are emitting 50,000 photons each for the indirect illumination. Because of the reasonably small photon map a corresponding small number of only 350 photons can be entered for the photon pool size. Again, multiple shadow sensors were applied for each light source to soften the shadows cast by the direct lighting. If you try to render the same picture with photon mapping as a global illumination model, then many more photons have to be emitted by the candles, otherwise the shadows cast by the relative thin candles would not be visible at all. This is totally independent of the complexity of the scene. Around a million photons were necessary for each light source to be sure of getting clearly visible shadows from the thin candles. With it, you would have to enter a corresponding high number of photons to gather for the photon pool to get smooth intensity transitions and to prevent a spotty appearance.

Example of an architectural outdoor scene and photon mapping only for indirect illumination (model: John Ridgway, Julia's House from the "Frontiers"-comic book series - you can download the project file from the internet library). This time a parallel sun light was applied (emitting 1 million photons, photon pool size of 1000 and 36 shadow sensors scanning the sun disc).
When rendering outdoor scenes with parallel lights, you should note that all photons emitted by the parallel light source are directed towards a bounding box surrounding all objects in the scene. Of course, the plane object will be ignored when calculating the dimensions of the bounding box - we

certainly do not want to scatter around millions of photons on an infinite plane. This is why the illumination of plane objects will always be carried out with direct lighting, no matter which rendering mode you select (see also: Material Dialog - Object Properties - Only Direct Light 290). To maintain the diffuse light interactions between ground floor and buildings you should create an additional base plate that is placed slightly above the plane and is adjusted to the dimensions of the scene. In the example depicted above this was accomplished by using the grassy subsoil plate beneath the asphalt ground.

Photon Mapping as a Global Illumination Model

If you choose <Global Illumination> in the render options 264 dialog, then all light interactions in the scene will be calculated by evaluating the photon map only.

Advantage - The whole illumination is made in one casting, there is no need to combine and coordinate the different lighting models. Also, the calculation of shadow sensors can be omitted, since all light scattering is provided by the photon map and shadows will result automatically from the distribution of photons in the scene. Because of the averaging of intensities about many photons shadows become very soft and the whole scene gets a very natural atmosphere.

Disadvantage - As described earlier in this chapter the shadows often become to soft, especially when emitting too few photons. To get accurate illuminations of minor details in the scene you have to emit a great number of photons, usually a million and more. Furthermore, you have to enter a corresponding high number of photons to gather for the photon pool to get smooth intensity transitions and to prevent a spotty appearance (about 600 up to 2500 and more, depending on the size of the photon map). This requires a fast cpu and a high memory capacity of at least 256mb and more. Before rendering pictures demanding so much computer power and time you should always render preview pictures with a lower resolution of emitted photons and a smaller number of photons to gather in the photon pool. The light intensities are distributed consistently among the whole of all emitted photons, so that the picture brightness is constant and independently of the number of photons emitted. Therefore, you can render good test images using photon resolutions of 50,000 up to 100,000 photons for the emission and photon pools with 250 up to 500 photons for the evaluation of the photon map. Only when you adjusted scene settings, render options and light intensities the final rendering can be started with lights emitting millions of photons and a photon pool picking up about 600 to 5000 photons for each shading of a point on a surface.

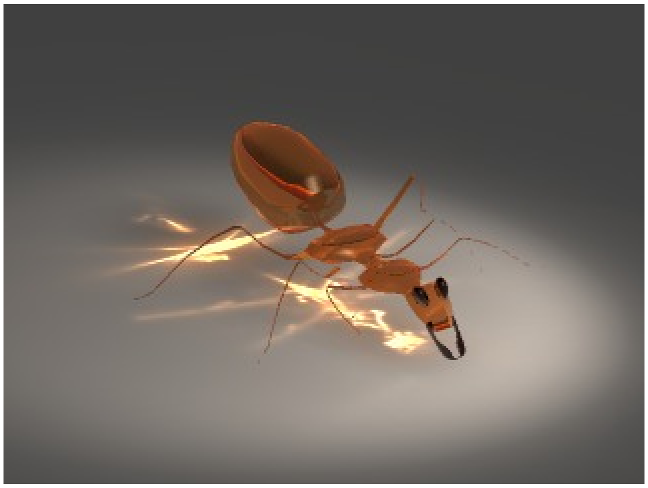
Example: This picture of a great hall was rendered in global illumination mode. The only light source is a spotlight placed in front of a side entrance to simulate sunlight penetrating through an open door. The spotlight shoots about 2 million photons through the side entrance into the hall. For the evaluation of the photon map the photon pool size was limited to 700 photons. Initially all photons are directed along the side corridor, but you can clearly see that actually all parts of the hall are illuminated, even at the end of the main corridor where the camera is positioned. This is due to the multiple diffuse reflections of the photons in tracing their way through the architecture.

This pictures shows the same scene when rendered without photon mapping - just pure raytracing and direct illumination. Because there is no diffuse interaction of light in the scene only the side corridor is illuminated. Nevertheless, you can discern the columns in the main corridor, but that's only because the fog function has been switched on, so that the shapes get brighter with distance and fog density. Of course you could also switch on the constant ambient light, but columns, walls and ceiling are designed in one basic color, so increasing the light intensity with a constant term would not add to the visible detail. In such a case, if you really want to evade using photon mapping, you would have to place some low intensity lights at the corridor ends to simulate the light reflected from the walls.

Once again the hall rendered in global illumination mode, this time illuminated by 16 burning torches 358 creating a dim atmosphere in the hall. Although using much more light objects in this example, the rendering time is not much higher than in our first example of the daylight scene with only one spotlight in front of the side door. This is due to the fact, that only the whole amount of emitted photons and with it the size of the final photon map counts for the rendering time. In the example depicted above only 60,000 photons were emitted for each torch light, adding up to only 960,000 photons. That's just half of the photon number the single spotlight emitted in the daylight example.

Caustics

The examples discussed beforehand mainly show the advantages of photon mapping regarding the diffuse interactions of light - when light is reflected randomly from rough surfaces. With photon mapping you can also trace the path of photons that are reflected from highly specular surfaces or transmit through transparent materials. The visible light patterns arising from these reflections are called caustics.



Example of caustic light reflections under a small glass figurine, caused by photons that were refracted when transmitting through the glass. The project file is part of the MR-3D Designer installation under "/projects/caustics/ant.cmo".

This picture shows caustic reflections casted from the water surface onto the walls.

Caustics Photons are stored in a separate Photon Map, the so-called Caustics Map:
Usually a photon map is evaluated only for the indirect illumination in combination with direct light for the main illumination and shadow calculations. For this purpose it will be sufficient to emit only several ten thousand photons into the scene so we can average the general area brightness at each point in the scene. Caustic reflections, on the other hand, are often sharply-outlined light patterns, as in our figurine picture shown above. It would be impossible to render these light reflections when only a few photons had been scattered around - you wouldn't even glimpse the light focused beneath the figurine. Now, it would be also ridiculous to emit millions of photons into the scene (and have to evaluate huge photon maps afterwards) only to cover the light reflections of a little specular object somewhere in the scene. That's why we have to manage different photon maps, one for the global photon map and the general illumination, and a separate caustic map only for those photons that have been reflected or transmitted via a specular surface before hitting a diffuse surface. The caustic map is build in a second photon tracing pass where additional caustic photons are aimed only towards such objects that are highly specular or transparent and with the object attribute <Caustics - Aim Additional Photons at Object 290 >. The evaluation of the two different photon maps also requires separate photon pools. For the global photon pool many more photons have to be gathered for the averaging process, so that soft and clean light transitions can be calculated for the area brightness. However, for the caustics pool we need comparatively fewer photons, because we want sharp and clearly visible light reflections.
As with the global photon pool, you can adjust the size of the caustic pool in the render options 264 dialog. The emission of additional caustic photons can be switched on or off for each light object

separately in the light dialog 370. You should activate the caustic photon emission only for lights that stand near to or are directed towards objects, that have the object attribute <Caustics - Aim Additional Photons at Object 290>. If you want to render a picture with the camera focused on a caustics object as the main part of the image, it is advisable to use a spot lamp for the illumination because the stream of photons can then be aimed directly towards the target object.
The number of additional caustic photons emitted from a light source is entered via the light emission parameters in the light dialog. Instead of specifying a certain number of photons this time you just have to enter a factor that describes how many more photons per area have to be emitted towards the caustic objects than for the global photons. Take again the glass figurine as an example. The emission of global photons was set to 50,000 photons. For the emission of caustic photons the factor was set to the maximum value of 100 via the <n-Times more Photons>-Parameter. During the processing of the photon map in the first pass, when the 50,000 global photons are emitted, about 1100 photons found their way through the glass figure and were saved in the caustic map. Then the additional emission of caustic photons is started with 100-times more photons, this is 100 * 50,000 = 5 millions photons, but this only specifies how many more photons per area are emitted in general. Since caustic photons are only directed towards caustic objects in the end "only" 200,000 photons find their way into the caustics map. This is more than enough for a sharp representation of the caustic light effects under the glass figure.

This picture was rendered under the same conditions in global illumination mode. Only 100,000 photons are emitted from a simple point light source and a factor of 100 for the additional caustic photons. As a reward, wonderful light reflections show on the wall and the base. This demo is found in the projects folder of the MR-3D Designer installation under "/projects/caustics/perfum_photonmap.cmo".

More Features of Photon Mapping:

Deactivating Photon Mapping for Individual Light Sources
Each light source in MR-3D Designer can emit photons for the photon mapping process. However, you can also exclude individual light objects from this process. If you switch on the <No Photon Mapping for this Light Object 370> option in the light dialog, then, instead of emitting photons, the corresponding light object will illuminate the scene with only conventional direct light algorithms, no matter which rendering mode is activated. You can use this function for small lamps in instrument controls or for lights far away in the background or, e.g., for spot lamps illuminating

only small parts of the scene. To save rendering time just switch on this function for all light objects that do not contribute much to the general illumination in the scene.

Deactivating Photon Mapping for Individual Objects

You can also exclude individual objects from the photon mapping process. If you switch on the object attribute <Only Direct Light - No Photon Mapping 290> in the Object Properties dialog, then the object becomes invisible for photons and is illuminated directly instead by each light source. Possible uses:

- if rendering in global illumination mode (only photon mapping, no direct lighting) and a small object is not hit by enough photons to shade it accurately. Instead of emitting more photons you can exclude these small objects from photon mapping and they will be correctly shaded again with direct lighting.
- if rendering moving objects in an animation with a static photon map. If the photon map is calculated only once at the start of the animation then objects excluded from photon mapping can still move through the scene.

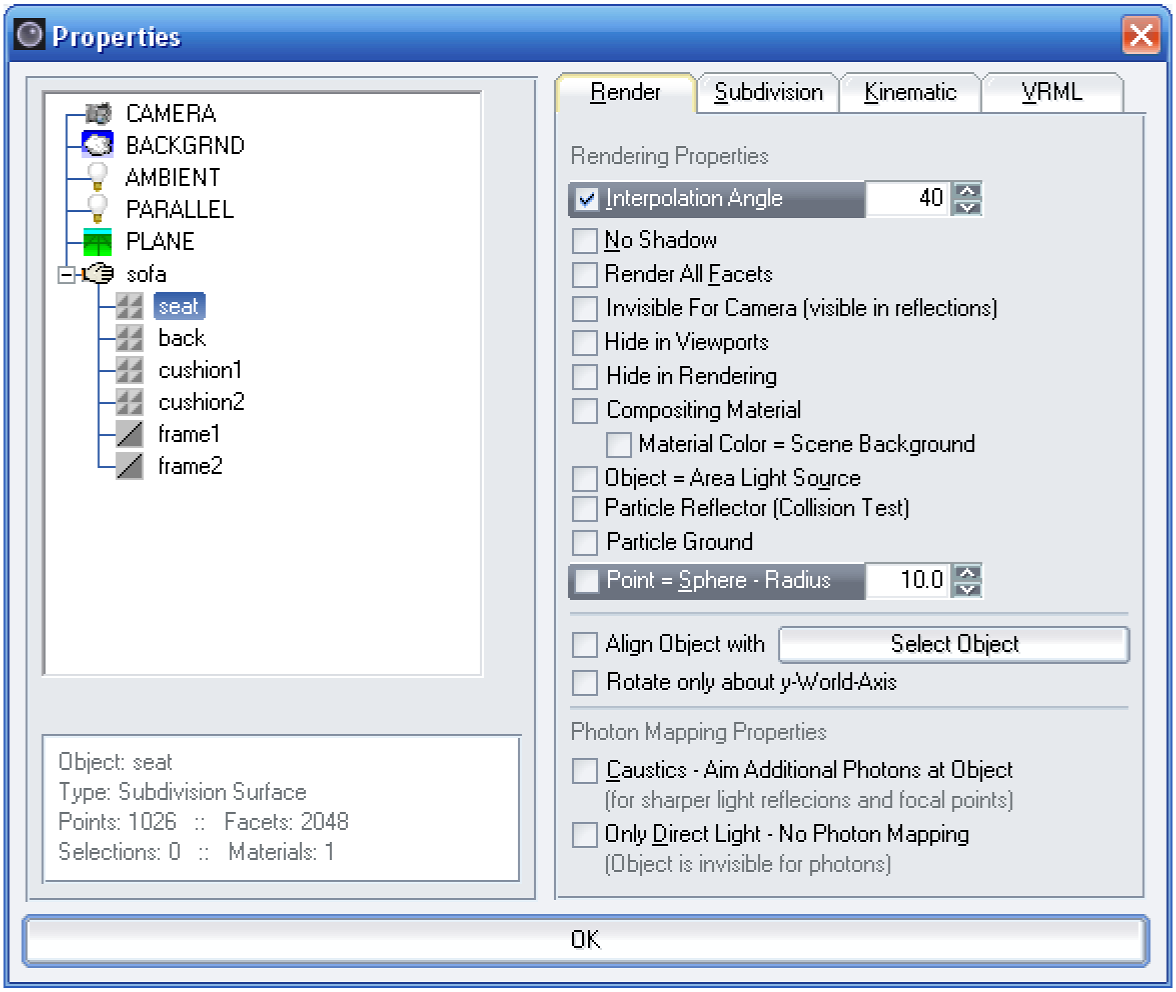
For the plane object this attribute is switched on automatically. Since the plane extends up to the horizon it would be nonsens to waste photons above this infinite plane.

Static Photon Map in Animation

If you activate this feature in the render options 264 dialog, the photon map will be calculated only once - at the beginning of an animation. You can use this option to animate a fly through an architectural scene where the objects themselves do not move. It is also possible to exclude individual objects from the photon mapping process - those will be illuminated only with direct lighting - and then you can even render animations with a static photon map and moving objects, e.g. a car driving through a street lined with houses.

Part XII

12 Object Properties



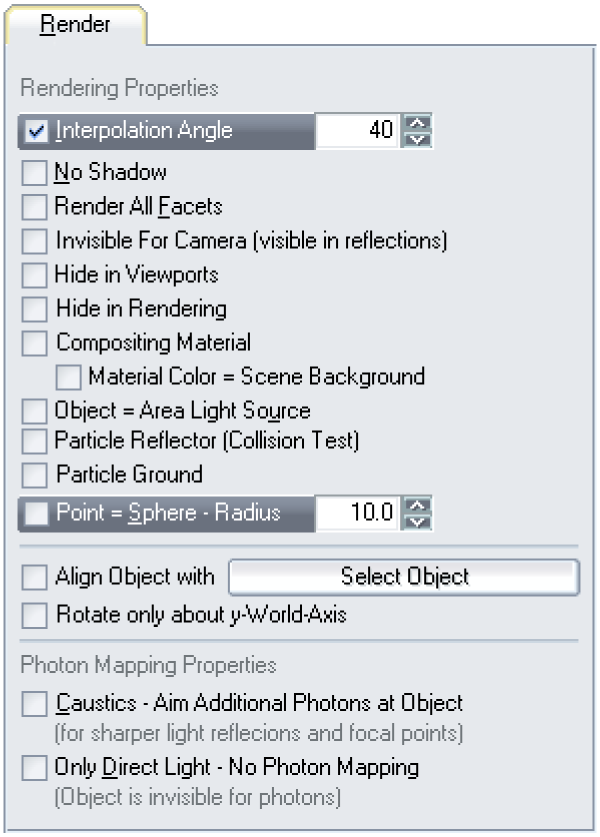
 - Menu "Objects - Object Properties" - Short Cut:  + "P".

In the Object Properties dialog you can edit a variety of object properties affecting the picture calculation or the behaviour of a jointed object in a kinematic chain.
The left side of the dialog is occupied with an object selection window. Under it an object info is provided with all relevant information regarding the number of points, facets, frozen selections and materials held by the selected object.

There are 4 register tabs to switch between the different sets of parameters on the right side of the dialog:

- Render 290 - The render-object properties affect the picture calculation.
- Subdivision 163 - The Subdivision Surface object property activates the real-time deformation that transforms a low resolution object into a higher resolution model with organically-curved shape.
- Kinematic 222 - The kinematic parameters control the degrees of freedom for joints which are moved by Inverse Kinematics.
- VRML 297 - Objects saved to a VRML-project can be provided with internet links.

12.1 Render Properties



The "Render" page of the Object Properties dialog

The page is divided in two sections. At the top you can edit different object properties that affect the picture calculation, for example, if an object casts a shadow or not. The lower section provides object properties which influence the behaviour of objects when rendered in photon mapping mode with global illumination.

Rendering Properties

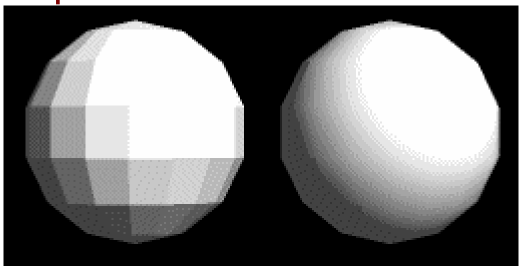
- Interpolation - Angle of Interpolation 290
- No Shadow 290
- Render All Facets 290
- Invisible For Camera (visible in reflections) 290
- Hide in Viewports 290
- Hide in Rendering 290
- Compositing Material 290
- Material Color = Scene Background 290
- Object = Area Light Source 290
- Particle Reflector (for Particle Collision Tests) 290
- Particle Ground (Drop Particles to Ground) 290
- Point = Sphere 290
- Align Object with Target Object 290

Photon Mapping Properties

- Caustics - Aim Additional Photons at Object 290

- Only Direct Light - No Photon Mapping 290

Interpolation



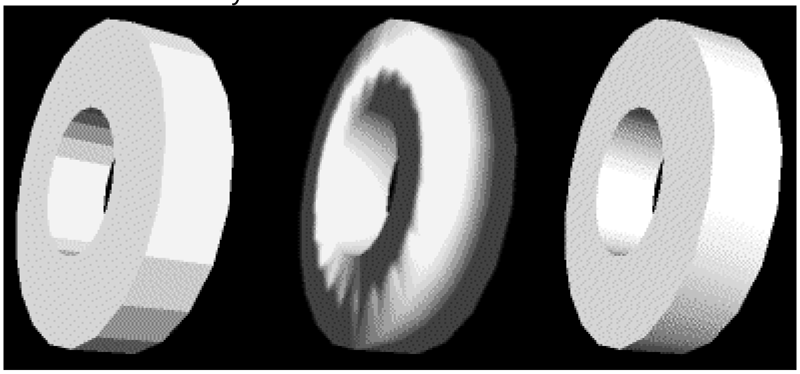
Interpolation gives the surface of an object a rounded appearance, producing a very much more realistic depiction, especially on swept objects. If you create, for example, a sphere from the primitives menu 146, which is formed from relatively few points and facets, you will see that without interpolation:

- the sphere is only approximated and the triangular facets are easily seen.
- the outline of the sphere is not round, but has corners due to the edges of the facets along the outline.

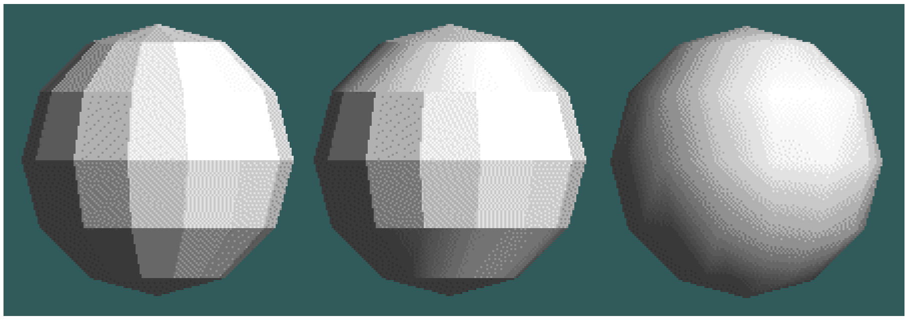
If you switch on interpolation for the object, the surface normals 8 of the facet and the adjacent facets are included in the calculation of the illumination of the facets, resulting in the impression of a curved surface. This makes the object look smooth and rounded.
However, only the facets are shaded, the outline of the body remains the same. The difference is shown in the illustration.

Interpolation Angle

It is often useful to exclude some part of an object's surface from interpolation - the top and bottom surfaces of a vertical cylinder, for instance. For this reason interpolation is made adjustable by setting the maximum angle at which adjacent facets are involved in the interpolated shading. If the angle between adjacent facets exceeds this value the surface calculations of either of the facets will not be influenced by the other.



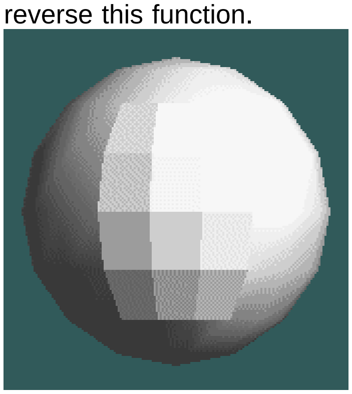
The screenshot shows three representations of the same object, but each of them has a different setting for interpolation. On the left you see the object with no interpolation. In the middle the object is interpolated to a maximum angle of 180 degrees, ie. all facets are used for interpolation calculations. However, you will see that the result is not what you would expect from an interpolated rounded surface: as the covering surfaces are also involved in the interpolation their triangular shape is responsible for making the object look rather crinkled.
To the right, you see an object with interpolation set at a maximum of 60 degrees. The angle between the sides and the covering facets is 90 degrees so the interpolation for both types of surfaces is calculated independently. The sides still appear rounded, since their respective angle is less than 60 degrees, resulting in mutual interpolation.



Here you see an example showing the same effect on a spherical object. From left to right, it is shown without interpolation, interpolation to a maximum of 22 degrees (involving mutual interpolation only for the uppermost and lowermost facets), and interpolation to a maximum of 60 degrees, where all facets are smoothed in respect to each other.

Excluding Single Facets from Interpolation

In the "Edit Object 227" menu you have the additional possibility of marking individual facets 138 and switching off interpolation for these facets. Clicking the "facet interpolation 242" <OFF> button will switch off interpolation for all selected facets. Using the "Facet interpolation" - <ON> button will reverse this function.

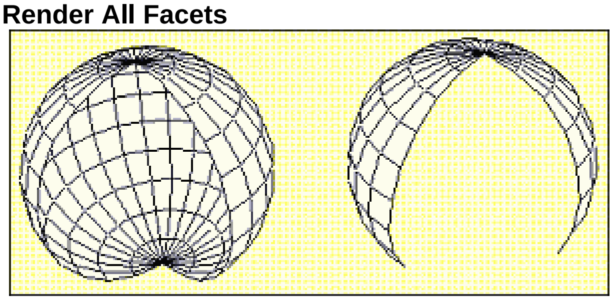


The screenshot shows a sphere with its middle facets excluded from interpolation.

No Shadows

Objects with this attribute do not cast a shadow onto other objects (i.e. self-luminous objects or object transformed to area lights or window glass that does not filter light). However, you can also decide for each light source separately in the light dialog 341 with the option <Light - No shadows 351>, whether it should contribute in rendering shadows.

Render All Facets



It is not always necessary to consider all the facets of the object when an object is drawn. If, for example, you construct a sphere, essentially only the front hemisphere needs to be drawn - the back half cannot be seen. When creating a primitive object in MR-3D Designer the facets normals (vertical vector on the facet) are always facing the outside. When rendering the picture, only those facets are drawn for which the facet-normal is directed towards the camera's viewpoint (for angles <= 90 degrees). However, this applies only on closed objects that you cannot see into. In the illustration you see an example of a three-quarter sphere - having an opening without covering facets. On the left half of the picture you see the sphere where all facets are to be represented (object property <Render All Facets> switched on). On the right side you see the sphere as it is drawn if only those facets are shown where the normal is visible to you (angles <90 degrees).

Invisible For Camera (visible for reflections)

If this option is activated, an object is hidden from the camera but it can still be seen in mirroring surfaces. Furthermore, the object is still included in the light distribution calculations of the global illumination routines (Photon Mapping and Ambient Occlusion). This property is especially useful if you want to render an indoor scene from outside the room. If you switch on the "Invisible for Camera" property for a wall of a room you can place the camera in front of it and take a look through the hidden wall at the inside of the room, while all reflections and lighting settings remain as if the wall were still there. This way, it is much easier to find a good camera viewpoint and to avoid the use of a distorting wide angle focal distance.

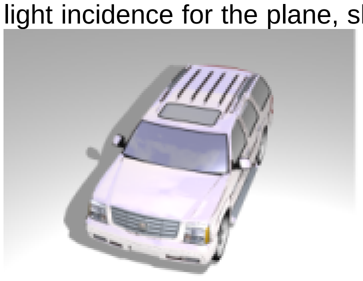
Visibility

You can hide objects in the viewport depiction or in the final rendering. In contrast to objects that are switched off in the Object Selection window these hidden objects are still active. For instance, you can select a hidden object by clicking on its name in the Object Selection window and then move it around in the viewport by repositioning the bounding box drawn around the hidden object.

- **Hide in Viewports** - Objects are hidden in the viewport windows but are still visible in the final renderings. This option helps to clear the viewport windows of already constructed objects when creating and working on new models. Objects not visible in the viewport depiction are indicated with an icon added to the name of the object. There are several ways to remove this object property. You can switch it off in the Object Properties dialog or via the corresponding menu entry of the main popup menu 133 or just by clicking on the icon in the Object Selection window.
- **Hide in Rendering** - Objects are visible in viewport depiction but not in the final rendering. This way you can use additional objects to aid construction work (i.e., blueprints projected onto planes to serve as construction templates), and automatically hide them in the rendering. Objects not visible in the final rendering are indicated with an icon added to the name of the object. This object property can be removed again by clicking onto the icon (or by choosing the corresponding menu entry in the main popup menu or the Object Properties dialog) .

Compositing Material

3D-models are often rendered in front of a single background color for later masking operations in an image editor. There the background color is used as a masking color - the object's shape can easily be separated from the background and you can copy it into another picture or, for instance, into the background of a web page. But how do you include shadows? If you place your object onto a single colored plane which receives the shadows, it is not easy to illuminate the scene so that the plane remains a single color. For this purpose you would activate the <Compositing Material> object property for the plane. Then the plane will only be drawn with its pure material color and the incidence of light will be completely ignored except of the shadow calculations. Since there is no light incidence for the plane, shadows are calculated subtractive from the material color.




The picture above on the left illustrates the problem. The car was placed on a single colored plane but because of the illumination' the plane is rendered with gradient colors. In an image editor it would be difficult to find a particular masking color to separate the car and its shadow from the background. The picture above on the right shows the same scene with <Composition Material> activate for the plane object. Now the color of the plane is painted in pure material color - without any illumination calculations - and still receives correct shadows.
The <Composition Material> property is also very useful in situations where you want a shading independent of the light incidence, for example, for billboard 190 projections of animated explosions or smoke clouds.

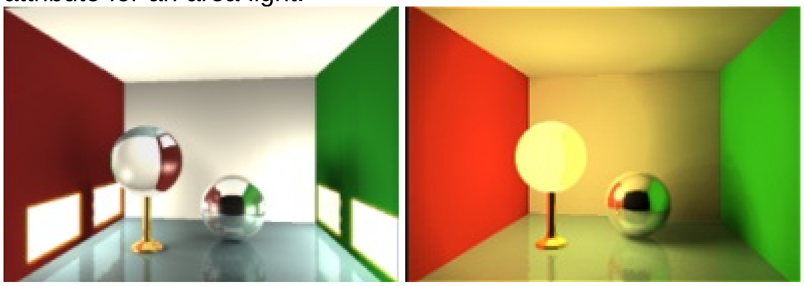
Material Color = Scene Background - This object property is only available in combination with the <Compositing Material> property. In addition, the background object has to be switched on. Then, if you activate the object property <Material Color = Scene Background>, the object will be painted in the same colors as the background. In effect, the object appears to be transparent in front of the background. But - and that's the clue - it will still receive the shadows cast by other objects.

Example: A model of a car (model: mr-clipart.com) was placed onto a plane object and a photographic environment texture (HDR-Image by Paul Debevec, www.debevec.org) chosen for the background model. The final rendering on the right shows the plane textured in the colors of the environment map 394 projection - thus becoming invisible - but still receiving the shadows cast by the car model. In this example also the reflections and actually the whole illumination information was provided by the environment texture (see Image Based Lighting 394). As a result you can hardly tell the car is a 3D-model inserted in the picture and not a real part of the photographic scene.

Converting an Object into an Area Light Source

Area light sources often used in modern architecture, e.g. in light panels, are hard to simulate with standard point light sources. However, in MR-3D Designer you can convert any object you like into an area light source, just by activating the <Object = Area Light Source> object property. Every point of an object will then be interpreted as a little subordinated point light source contributing a small share to the whole of the object's light intensity. Since each point adds separate light and shadow feelers 264 to the illumination process, the time for rendering the picture increases with the object's point resolution - especially when rendering shadows. NURBS-patches are ideal area light objects because of their regular structure with evenly-spaced points forming the surface. In addition, you can change the point resolution of NURBS-patches at anytime, e.g. to render faster preview pictures with low point resolution and then change to a higher NURBS resolution for the final rendering. As NURBS-patches do not have a thickness they can be easily installed in wall panellings.
Only a maximum number of 500 light- and shadow sensors will be calculated for each area light object. If the object construction contains more points (or a smaller number of shadow feeler has been set), then samples are picked at random from the object.

Adjusting the Light Properties for Area Lights - Once you have activated the <Object = Area Light Source> attribute, the object will also be listed in the light dialog together with all standard light types.
Then, if you change to the light dialog, you will be able to edit the light parameters for the area light object. You can define the light color, intensity and the photon emission parameters. Of course, area lights are included in the photon mapping process and therefore area lights can emit photons just like all standard lights, too.
The light color is independently of the material color. Think of the object as an ordinary body or as a container for a light source. If the light is switched of or is shining very dim, you still have to take into account the light reflections from the container, if it is illuminated by other lights in the scene. Therefore the object material is calculated and interpreted as an ordinary object surface with all of its possibilities, e.g. bitmap textures, reflection or transparencies. Only then, the self-luminosity is added to the materials color with the light color. In an animation, for instance, you can animate the light color from dark to bright and the object will slowly begin to glow and illuminate the scene.
Part of this interpretation is, that area light objects cast shadows, when they are illuminated by other brighter light sources. You can switch of the shadow casting bei activating the <No Shadow> attribute for an area light.



Two examples of area lights. In the picture on the left 4 NURBS patches converted to area lights illuminate the room. In the right picture the glass sphere was changed into an area light source.

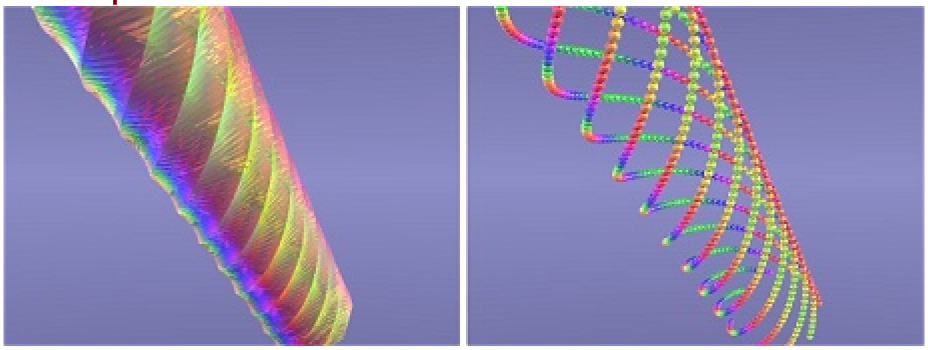
Particle Reflector

If you define particle-actions 425 and have switched on the collision-test 439 in the particle-editor 427, when an animation is generated particles are examined for collisions with all other objects that have the function <Particle Reflector> switched on.

Particle Ground

Particles 425 created with the option "Drop to Ground 431" will be placed only on the surface of those objects which have the object property <Particle Ground> switched on.

Point = Sphere



A very interesting effect can be achieved with this option, which dramatically changes the appearance of an object. If "point = sphere" is switched on, the object is no longer constructed from facets. Instead, all points of the object that previously defined the corners of the facets are used as centers for spheres. The radius of these spheres is set with the parameter beside the button. These spheres can only be rendered in raytracing mode, as they are defined as analytic objects 147 with center and radius. In the illustration you can see two identical objects, in which the only difference is that the right-hand object is rendered with the option "Point = Sphere".

Align Object with a Target Object

You can apply this object property to constantly align an object with a target object (press the <Select Object> button to call up an Object Selection window where you can choose a target object). For instance, a camera or a spot light can be aligned with a target to keep the object of interest always in focus when moving around. Or you define the camera as a target for other objects so that these objects always face towards the camera, for instance, when inserting flat billboards with masked picture projections of people or trees in architectural scenes. For this purpose you can limit the alignment to the horizontal plane (the object is only rotated about the y-world axes, so that people and plants keep the ground under their feet). Otherwise the object will be rotated around all 3 body axes so that the object's z-axis will point to the center of the target object. Other good applications for the auto-aligning function are animated billboard projections of explosions 113 or expanding smoke clouds 117 which always face the camera. This is possible even with particle systems - all particles will adopt the alignment property from their reference object.
The rotations performed to align objects do not change the model data or influence the animation data of the object (no animation tracks or keys are created or changed), it is rather a deformation operation (like SDS 163 or the animated bend and twist deformations 251) which is performed on a copy of the original model data immediately before rendering the scene.

Object Properties - Photon Mapping

For a deeper understanding of photon mapping see also:
Photon Mapping 277 - Introduction and examples
Render Options 264 - Global Illumination - Raytracing + Photon Mapping
Light Dialog - Photon Emission Parameters 370

The following two options influence the behaviour of objects when rendering in photon mapping mode:

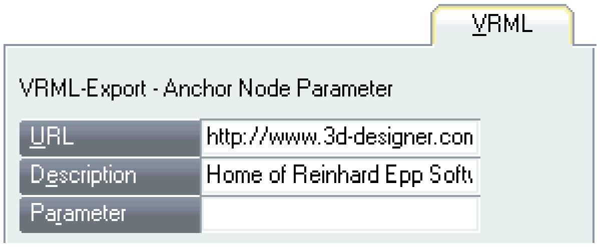
Caustics - Aim Additional Photons at Object - This option is relevant only for highly specular or transparent objects that cast caustic light reflections. If the <Caustics> attribute is activated, additional photons are aimed at this object to get a higher resolution for the light reflection patterns that arise from the photons reflected and transmitted through the object.
Deactivate this function for ordinary windowpanes - If the <Caustics> option is deactivated, the photons will transmit in a straight line through the object, without being deflected by the refractive material. Only the light color will be filtered by the material, as in conventional raytracing without photon mapping. This is particularly useful for ordinary windowpanes. In general the light transmitting through window panes is intended for the general area illumination in a room, contributing to the global photon map, and not for unnecessarily caustic reflections.

Only Direct Light - No Photon Mapping - You can exclude individual objects from the photon mapping process. If you switch on the object attribute <Only Direct Light>, then an object becomes invisible for photons and is illuminated directly instead by each light source. Possible uses:

- if rendering in global illumination mode (only photon mapping, no direct lighting) and a small object is not hit by enough photons to shade it accurately. Instead of emitting more photons you can exclude these small objects from photon mapping and they will be correctly shaded again with direct lighting.
- if rendering moving objects in an animation with a static photon map. If the photon map is calculated only once at the start of the animation then objects excluded from photon mapping can still move through the scene.

For the plane object this attribute is switched on automatically. Since the plane extends up to the horizon it would be nonsense to waste photons above this infinite plane.

12.2 VRML Export Parameters



Select the **<VRML>** register in the Object Properties 289 dialog to insert several additional parameters required for VRML 2.0 export 34.
Here you can assign URL-links (internet related addresses like http://www.3d-designer.com) to each individual object.
Then, in a VRML capable browser, you can change to other web pages simply by clicking on the corresponding 3D object. It is also possible to enter a link to another VRML project file (Extension '.wrl'). Clicking on the corresponding object would then result in a jump directly into the next 3D world. Text entered in the description field is displayed in a VRML browser when pointing with the mouse to the corresponding object.
You can also enter a target frame in the 'Parameter'-field, e.g.: *target=my_frame*

see also: VRML 2.0 Export 34

Part XIII

13 Materials and Textures

Fine feathers make fine birds. Transferred to our 3D-world we could say that without a good surface texture the most complex 3D model is not much to look at. In the material dialog you will find a vast choice of possibilities to create your own textures or simply load pre-defined textures from the visual library.

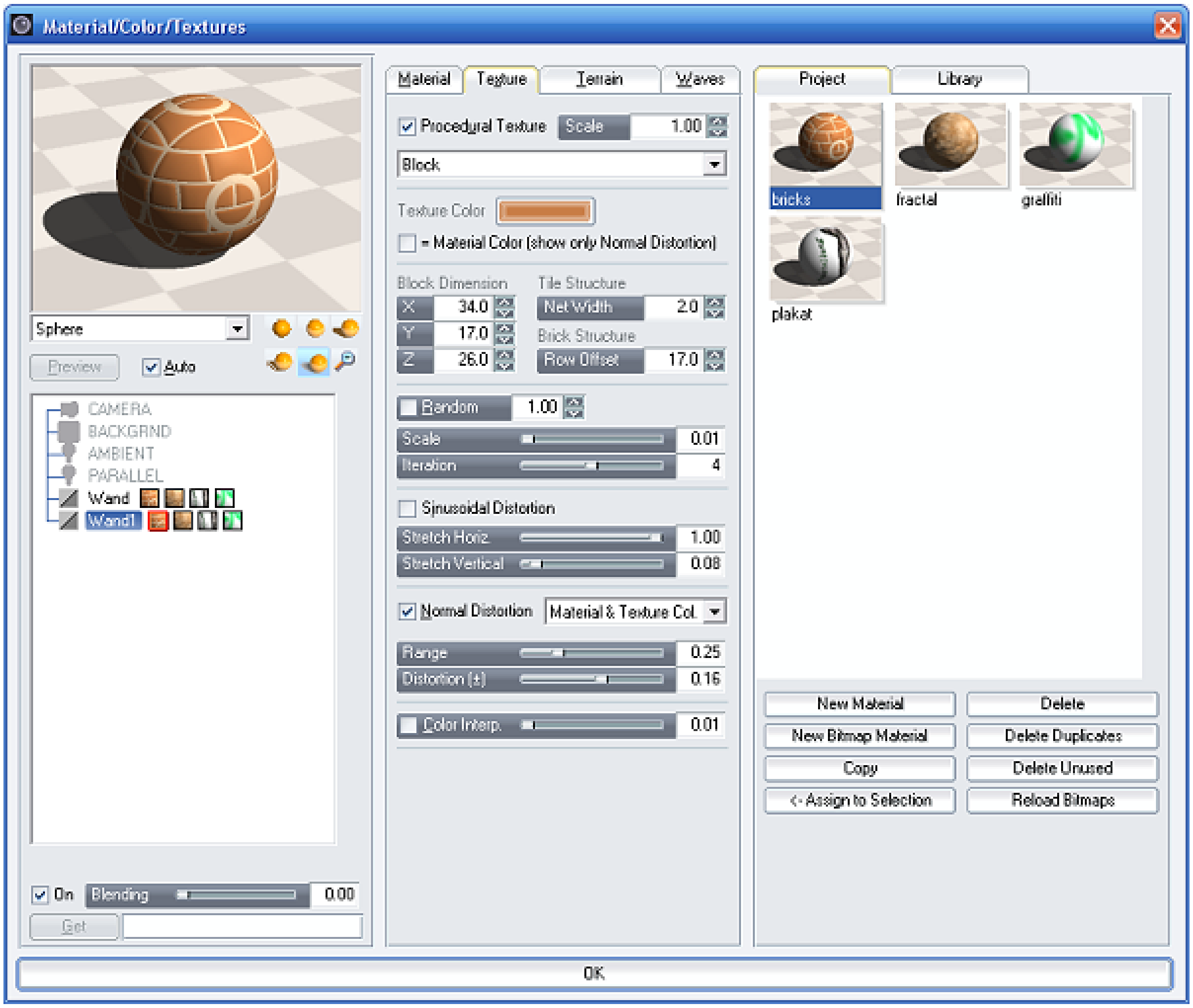
The pages in the dialog:

Basic Material, Procedural Volumetric Textures, Landscape Textures and Water

Bitmap Textures, Bumpmaps, Reflection Maps, Transparency Maps and uv-Mapping

Basic Material Properties - Reflection, transparency or self illumination 328

13.1 The Material Dialog



 - Menu "Objects - Material/Color" - Short Cut:  + "M".

This dialog enables you to define the materials for an object's surface using an extensive range of parameters. For instance, you can apply mathematically defined textures to generate volumetric surface structures similar to grained wood, marble, rock or multi-layered landscape textures on a fractal basis. These textures can be combined with bitmap materials projected onto an object. You can apply bitmaps to texturize an object but also to control the reflectivity or transparency of the objects surface. With bumpmaps you can even simulate embossed structures on an object's surface. However, is it not necessary to specify all these possibilities at the outset - simply transfer materials from the visual material library 28 into the project's material browser to select a required material. Use this material as a starting point and modify it to your own needs, before you assign it to an object or a frozen facet selection. You can assign a single material to several objects at a time. Then, when changing the parameters of this material you will change simultaneously the appearance of all objects referencing to that material. Furthermore you can apply several layers of different materials, procedural as well as bitmap textures, which can be continuously mixed together. Above all, for each object you can define different material lists on different key positions just by switching on or off materials on particular positions in time. In an animation a smooth blending will be calculated from one material combination to the next set of activated materials.

The Structure of the Dialog:
The dialog is divided into several areas:

- The **preview window** at the top left
- The **object selection** right beneath the preview window
- Under the object selection window a **<Blending>** slider and an on/off switch is located. The slider controls the material blending if more than one material is assigned to an object. With help of the **<On>** button you can switch on or off a material at different time positions.
- The **<Get>** button under the Blending slider is provided to adopt a complete material list from another specified object
- The center part of the dialog contains different sets of material parameters that you can switch between the tabs in the dialog's document header. Depending on the material type - procedural or bitmap material - a different choice of pages can be selected:

  Procedural volumetric textures:
  - Material 309 - The basic material settings and object-attributes, which are not necessarily related to the appearance of the material
  - Texture 313 - The parameters for the procedural textures
  - Terrain 323 - Additional fractal texture layers for landscape textures
  - Waves 326 - Animated water textures

  Bitmap materials:
  - Bitmaps 328 - The parameters for bitmaps, bumpmaps, alpha maps, reflectivity- and transparency maps

- The right-hand part of the dialog contains the material browser with two register tabs:
  - The <Project> browser lists all materials which are used and saved in the current project.
  - The <Library> browser displays all materials located in the current library folder of your harddisk. You can load materials from the library into the project browser or save project materials to the materials library. See also: Visual Libraries 28

Preview Options of the Preview Window
Every change of a material's parameter will be immediately displayed in the large preview window 31 at the upper left of the dialog. In the selector box below the preview window you can choose between several preview modes. You can select a primitive object for real time previews (Sphere, Cube, Cylinder or Landscape) or the "Object" mode, where the selected object is displayed centered in the preview window. Last but not least there is the "Camera, complete scene" mode that shows a preview of the whole scene in camera view. If you choose one of the primitive objects for preview depiction then only the currently selected material will be visible on the primitive object. If several material layers are assigned to an object then you must switch to "Object" or "Camera" preview mode to be able to observe the changes in the material mix.

Material Mangment - Project Material Browser versus Material Library
The right-hand part of the dialog contains the material browser with two register tabs. If the <Project> page is activated then all materials used and saved with the current project are displayed with little tumbnail pictures in the browser window. At the start of a new project the <Project> browser contains usually only a preset basic material.
If you change to the <Library> page of the browser, then all materials found in the currently selected library path 28 of your hard drive will be displayed. A new set of browser functions is provided underneath the browser window with which you can transfer previously selected library materials to the project browser or - on the contrary - save the currently edited project material to the library to make it accessible for other projects. See also: Visual Libraries 28

The <Project>-Browser Functions

- **New Material** - Creates a new basic-, procedural- or terrain material in the project browser.

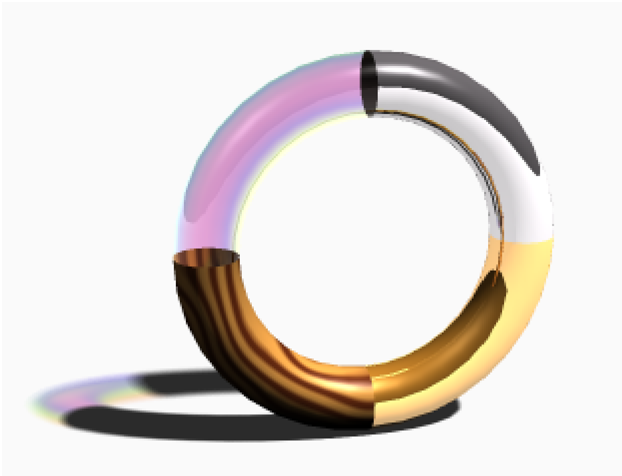
- **New Bitmap Material** - Creates a new bitmap material in the project browser.
- **Copy Material** - Adds a copy of the currently selected material to the project browser. You can modify material copies to get several variations of the original material.
- **Delete Material** - Deletes a material from the project browser and with it from all objects which reference to that material. (To simply remove a material from an object's material list, mark the corresponding material icon in the object selection window and press the "Del" key on your keyboard).
- **Delete Duplicates -** Deletes identical materials from the project and adjusts all object references to the remaining unique material.
- **Delete Unused** - Materials not assigned to any object or facet selection are removed from the project.
- **<- Assign to Selection** - The currently selected material (in the project browser) will be added to the material list of the currently selected object (in the object selection window). A double click with the mouse on a material's thumbnail will do the same. See below **-** "Assigning materials to objects or facet selections"
- **Reload Bitmaps -** If you are editing and changing a material bitmap in an external image editor while working with MR-3D Designer at the same time, you can press the <Reload Bitmaps> button to update the list of bitmaps loaded into the program. This function is also available throughout the workspace via the shortcut <Strg> + <R>.

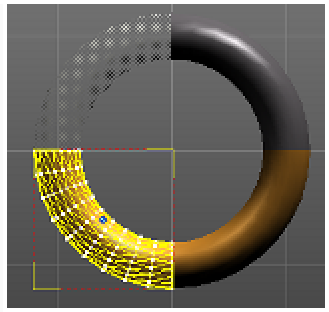
The <Library> Browser Functions
On the library page you can select several materials at a time to transfer them to the project page. Simply hold the <Ctrl> key pressed when clicking on a browser thumbnail.

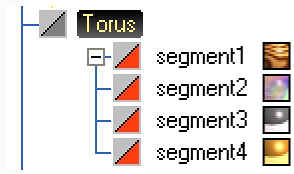
- **<Add to Project>** - All materials selected in the library browser are loaded into the project and the library browser window changes automatically to the project's browser page.
- **<Load into Material>** - This function loads a library material directly into the material currently under work in the Material dialog. The library material replaces the currently selected material on the project page of the material browser. You can apply this function also by double clicking on the thumbnail of a library's material.
- **<Save Material>** - The material currently selected for work on the project page can be saved to the materials library.
- **<Delete>** - This function will remove the selected material from the library folder on your hard disk.

Selecting a Material to Work On
Simply click on the corresponding thumbnail on the <Project> browser page to choose a material. Another possibility is to pick a material directly from an object's material list in the object selection window. There, all materials assigned to an object are listed in a row of small icons behind the object name. Clicking on an icon will select the corresponding material so that you can adjust its parameters.

Assigning Materials to Objects of Frozen Facet Selections
First select the object you want to apply with a particular material in the object selection window. Then choose a material from the project browser and assign it to the marked object by operating the <Assign to Selection> button beneath the browser window. Thereupon a small icon with a copy of the material thumbnail is added behind the object's name in the object selection window. With a mouse click onto this icon you can select this material (a red frame appears around that icon). If a material is marked in the object selection window you can remove it from the object's material list by pressing the "Del" key. But be careful **-** if no material icon of an object is marked then the currently selected browser material will be deleted from the project. This will also remove the material from all material lists of objects referencing to it. (If that happens unvoluntarily just leave the dialog and use the global undo 28 function to restore the project).
Materials can be assigned not only to objects but also to frozen facet selections of an object. This way you can apply different areas of an object's surface with individual materials.







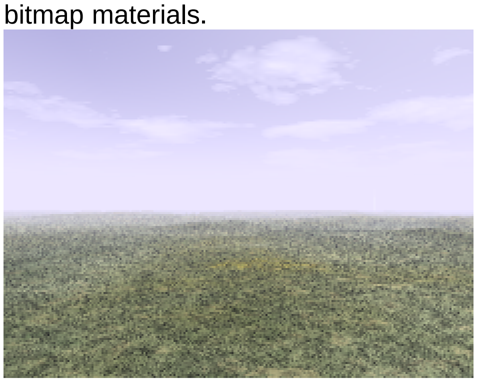
The illustration shows a simple torus object with 4 different materials applied to the 4 frozen facet selections created for the torus object. The picture on the top right shows the depiction of the torus in a shaded viewport window and under it you can see how the frozen facet selections and their material icons are displayed in the object selection window.

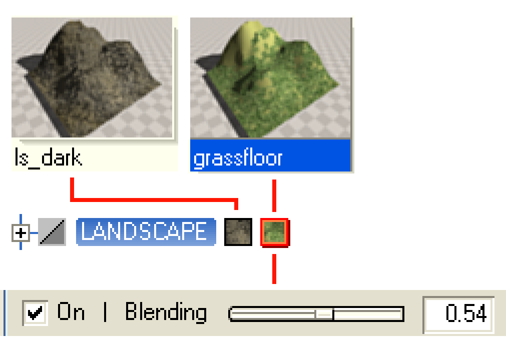
Texture Axes System - Aligning Materials on Objects

When you assign a new material to an object then always together with the material reference a new texture axes system 197 is generated for the object. You can use this texture axes to adjust materials indivdually on every object, for example, to align a bitmap texture on an object. Moving 197 a texture axes system will reposition the origin of a bitmap or a volumetric procedural texture, while rotating 211 the texture axes will change the threedimensional course of a volumetric texture or the direction of a bitmap projection, respectively. Scaling 206 a texture axes system will also result in a change of size of the texture pattern or bitmap.
See also - Moving Textures - Texture Axes System 197

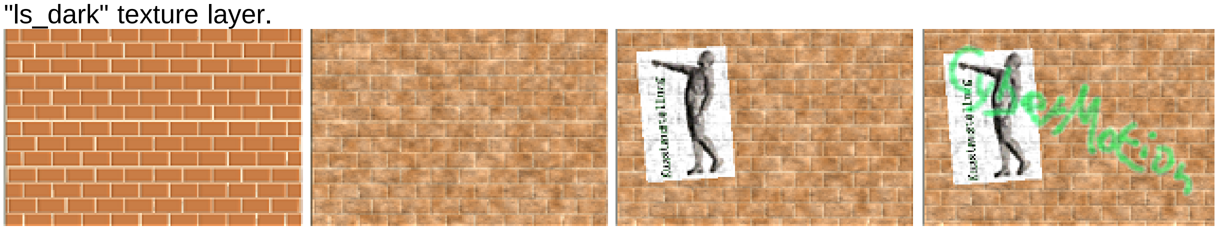
Mixing Materials

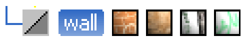
Up to 16 different materials can be applied to an object or to a frozen facet selection. These materials are listed in rows of small material icons behind the name of an object or frozen facet selection, respectively.
If you add a new material to an object's material list then at the start the new material will cover up all texture layers lying underneath it (These are all materials listed in front of the currently selected material icon). Then, with help of the <Blending> slider located under the object selection window, you can adjust a continuously blend of the currently selected material with the materials lying underneath it. Each material can be mixed with any other material, for instance, you can mix transparent materials with non transparent materials or even prodedural volumetric textures with bitmap materials.





Example: This forest soil texture (demo project "particleforest.cmo") uses two different procedural landscape textures. First the material "ls_dark" has been assigned to the "Landscape" object and then additionaly the material "grassfloor". Thereupon two new material icons appear behind the name of the "Landscape2" object in the object selection window. Choosing the second material icon with a left mouse click will enable the <Blending> slider underneath the object selection window. By moving the slider to the right the "grassfloor" material is increasingly blended with the underlying "ls_dark" texture layer.

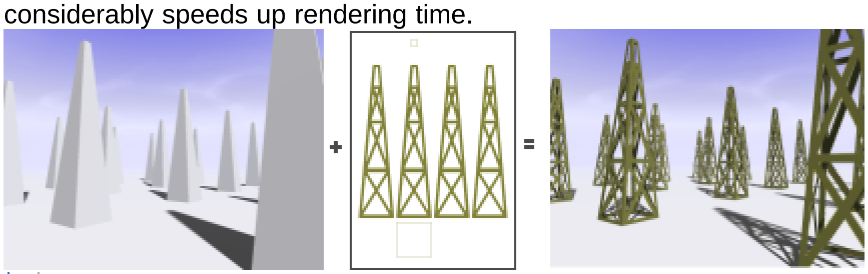




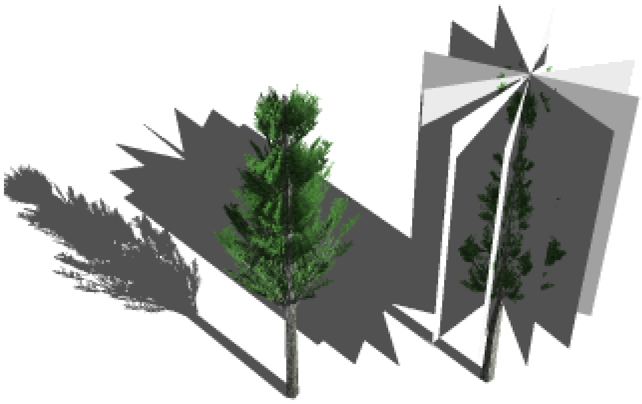
Example 2: A procedural brick texture has been assigned to a wall object. The mathematical brick pattern appears much to neatly. To distract from this artificial pattern another fractal rock texture has been added and mixed with the brick material. After that we added a bitmap material to simulate a poster fixed to the wall. The poster was applied of course without blending, it covers all underlying materials. However, the bumpiness structure caused by the normal distortion 313 of the underlying materials will show through the poster. Normal distortion is always applied additive (especially important for high detailed structures in landscape textures).
Finally another bitmap material has been added to the wall's material list. Now a graffiti writing is decorating our nice wall, spanning over the wall as well as over the poster. The graffiti picture has been combined with an alpha map 328, so that only the writing is projected onto the wall while the picture background is masked out.

Using Special Material Blends to Mask and Cut Out Surface Areas

If a totally transparent material (color white) without reflection or refraction (refraction = 1) is applied as the basic material of an object and you add an additional bitmap material, then this bitmap serves as a mask for the object, this is, the object becomes totally invisible at all places where it is not covered by a color of the bitmap material. You may say that you did not expect to see anything, since the basic material is totally transparent. But the important difference is that only a masking operation is applied to the object, no additional recursive search rays have to be traced as for colored refractive transparent objects or for mirroring surfaces. Instead the surface of the masked area of an object will be simply ignored as if it were not existent. This effect can be useful in a variety of situations, for instance, you can project people or trees on transparent panels and place them in an architectural environment. Instead or rendering millions of facets of real trees or complex human models then only simple planar billboards with bitmap textures are rendered which considerably speeds up rendering time.



Example: The picture on the left shows a simple pyramidal primitive object. As a basic material layer an uncolored and completely transparent, non-refractive and non-reflective material has been applied to the object. On top of it an additional uv-bitmap texture with a framework structure has been applied. The white background color of the bitmap has been masked out by choosing a corresponding white color as a color mask 328. On the right of the above illustration you see the result. Instead of a solid pyramidal object now only those parts of the object are visible which have the framework projected onto their surface. All other parts are literally cut out from the surface.



Example 2: You can even build complex objects like the tree illustrated above with help of this masking techniques. At the start I have rendered a side view of a "real" 3D tree I have lend from the CyberMotion's internet library at www.3d-designer.com. Next I projected the image applying uv-mapping onto a flat panel object. This panel was copied 6 times, rotated each time about 30 degrees, and finally merged in one single object using the Boolean Operation 233 "Join Objects 233" function (uv-coordinates of the panels are maintained by this function).
After that a somewhat tapered low polygon cylinder (only 5 rotational segments) has been centered in the middle of the panels to serve as the trunk of the tree. All facets of the trunk has been allocated to a frozen selection. Then a bark material has been applied via uv-mapping to the cylinder. Again the trunk was united with the 6 sided panel object using a Boolean Operation. Normally, when combining two objects then one object adopts the material settings of the other object. But frozen facet selections and their materials will be transferred to the merged object. Therefore the final tree object will keep the leaf material for the panels and the bark material for the trunk selection. If no basic transparent material is applied then the tree would look like in the illustration above on the right. On the left you see the result with the basic transparent material layer switched on - a complex conifer tree with pretty much details casting sharp shadows build from only one object consisting out of 32 facets. In contrast to real 3D-models these low polygon trees are well suited to populate terrains or even whole forrests. It is the usual way 3d game designers are creating plants for their realtime environments.

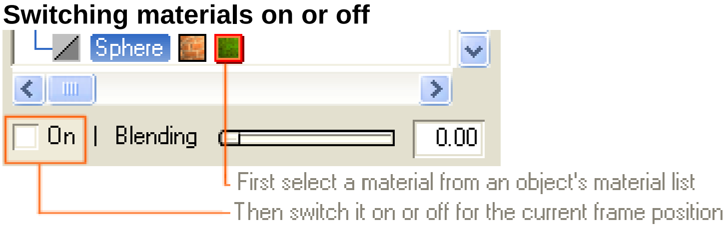
You can also use this method to create grass bushes and other plants. In the illustration above the grass bush is formed from 10 crosswise arranged rectangular planes with a simple grass pattern projected onto it and joined into one single object.
You may ask why I concentrated on merging a whole tree or gras bush into one single object with differently textured facet selections instead of arranging everything in a hierarchy? Well, since these plant objects consist only from one single object I can use these "simple" basic objects now as reference objects for MR-3D Designers Particle System 425 generator.

This wonderful picture displaying a thick grass layer consisting out of thousands of gras bushes (with millions of individual gras blades casting shadows) and hundreds of trees is rendered from merely a handful of plant objects used as reference objects for several particle actions set up in the Particle System 425 dialog. While spreading the thousands of particle copies over the terrain, the objects are rotated and resized by the particle action so that irregularities are filled in automatically. The picture shows only a detail screen from a demo animation presented in the 3d-designer.com gallery.

Animating Materials

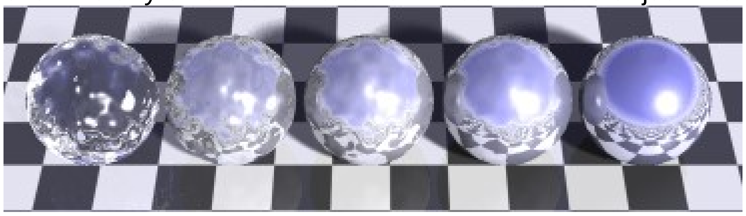
With MR-3D Designer you can animate transitions from any material mix into any other material combination, e.g., from glass to wood, from wood to stone or just to slowly fade in a bitmap label on top of other underlying materials. Setting up these effects is very simple. You just have to assign all materials you want to include in the texture animation to the material list of the corresponding object. Then, on different keypositions, just go into the Material dialog and switch on those materials that you want to appear on the object's surface at that particular point in time and on the contrary switch off all remaining materials you don't want to include at that time position. If you render now the animation then a smooth transition will be calculated from the current set of activated materials to the set of mixed materials valid for the next key position.

Switching materials on or off



Under the object selection window of the Material dialog, in front of the <Blending> slider an additional <On> switch is located. You first have to choose a corresponding material from an object's material list by clicking on one of the material icons in the object selection window. Thereupon the <On> switch will be enabled so that you can switch on or off the material for the current frameposition of your animation. Changing the material's on- or off -state at different positions in time will automatically create new material keys 401 for the corresponding object. These material keys will only save the on- or off-state of all materials of an object at a particular frame position, so you can still change the parameters of a material for a general change of a material's appearance for the whole animation.

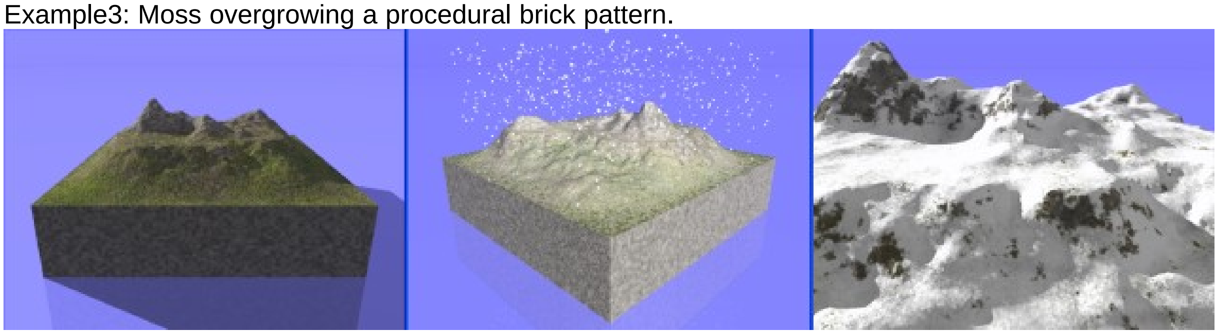
Example: A wood material is allocated to a sphere object. We want the wood texture to change in 50 steps further on the timeline to a reflective chrome texture. Therefore, in the material editor we allocate an additional chrome material to the sphere object. This chrome texture is not to show up in the beginning, therefore we select the chrome material icon next to the object name of the sphere and just switch it off. Now only the wood texture is painted in this keyframe. We move now forward in time to frameposition 50. In the material editor we switch off now the wood texture and switch on the chrome texture instead. That's all. If you play now the animation you can see how a smooth transition is animated from the wood texture to the reflective chrome texture. You can do the same with any combination of mixed materials. For instance, you can blend a set of 3 mixed materials over into any other combination of mixed materials or just into a single material.



Example1: The transformation of a bumpy glass sphere into a smooth steel ball.

Example2: A rocky surface changes into glowing lava

Example3: Moss overgrowing a procedural brick pattern.



Example4: Snow fall covers a landscape under a white blanket of snow. To render this animation only two materials have been added to the landscape object. Both materials have the same landscape texture 323 definition, the only difference is that for the second material an additional snow texture layer has been activated on the <Terrain> page of the Material dialog. Then, at the start of the animation material one (without snow) has been switched on while the snowy material has been switched off whereas at the end of the animation the status of both materials has been reversed again, the texture without snow has been switched off and the snowy texture has been switched on. That's all.

By the way, the snow fall limited to the rectangular area above the terrain is a real 3D-particle system 425 rendered from a spherical particle object. This is in contrast to the background editors "Rain/Snow" particle effects, which are rendered in post processing above the whole picture. (See demo file "/projects/snowfall.cmo").

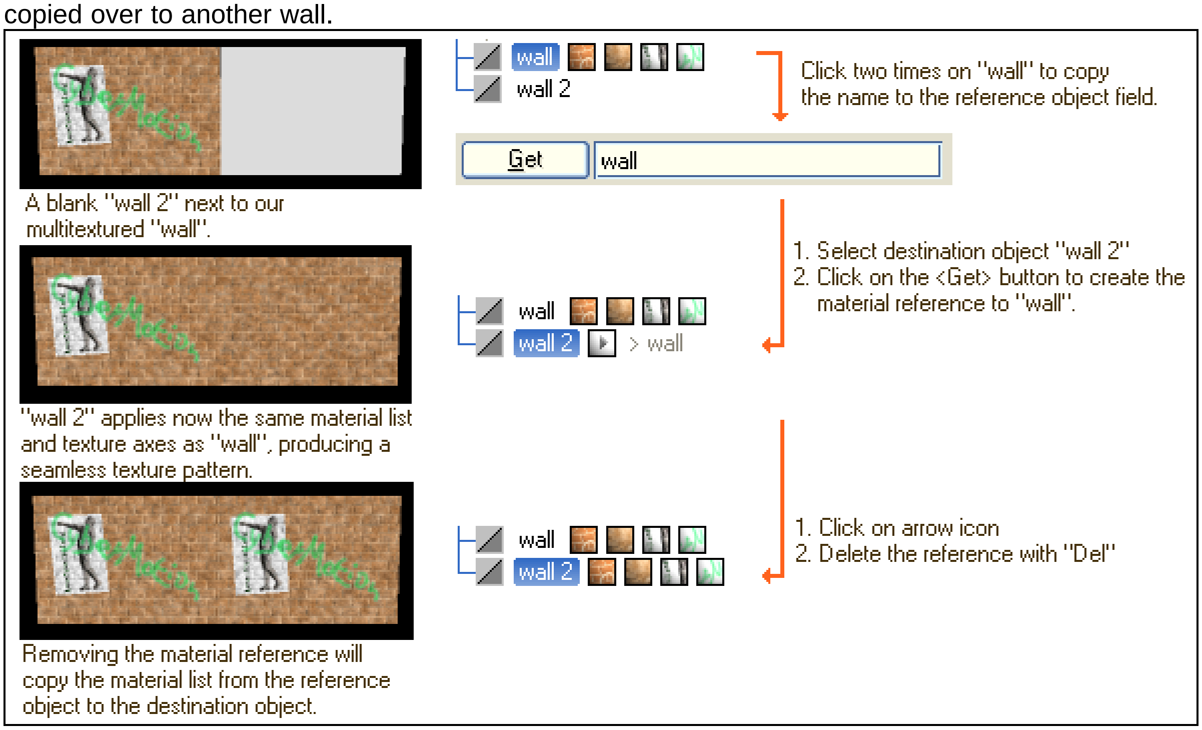
Referencing to Material Lists of Other Objects

Usually you assign materials to objects. When changing a material definition then the appearance of all objects referencing to that material will change too. But objects can also adopt the complete material settings from other objects. When you create a material reference to another object, then

the current material settings and even the texture axes system of the reference object will be used to texture an object's surface. Thus, objects that overlap each other are rendered with a continuation of the same pattern - so they appear to be formed from one piece of material.
You can also use this function to copy a material list from one object to another object. To do so, first create a material reference to another object and immediately afterwards, just delete it again. This will only delete the object reference - while all materials previously referenced to will be copied now to the current object's material list.
Creating a Reference to the Material List of Another Object - There is the <Get> button beneath the object selection window for this purpose. First select the object you want to reference to in the object selection window. Click once again on the object name - thereafter the name of the object appears in the field beside the <Get> button. Now choose the object in the object selection window that you wish should assume the material list of the reference object. Then simply operate the <Get> button. Thereupon an icon with an arrow replaces the material list of the selected object. From now on it will use the materials and texture axes of the reference object. If you select an object with an arrow icon in the Material dialog then also the name of the object to which the selected object references is printed behind the icon.

Adopting the Material for a Whole Branch in a Hierarchy - If you hold the  -key pressed when clicking on the <Get> button, then the object reference will be copied to all objects in the selected hierarchy branch.
Removing the Material Reference - Simply select the arrow icon and press the "Del" key on your keyboard.

Again the example of our multi textured brick wall. The illustration shows how the material list is copied over to another wall.



13.2 Material Dialog - Basic Material Attributes

Select the <Material> tab in the header of the Material dialog 300 to bring the material side to the fore.

Object Color, Diffuse and Specular Reflection

The diffuse reflection is the basic color of an object. Diffuse stands for a regular dull reflection of light, the object appears to be of a constant tone from all views. In contrast to diffuse reflection, specular reflection is the amount of mirrored light, i.e. highlights from light sources or mirrored objects in the scene. By specifying a particular color for specular reflection a filter function is applied for the mirrored light.
With metal surfaces the specular color usually comes close to the diffuse color, resulting, for instance, in golden reflections on a golden surface. Other surfaces reflect the whole light spectrum, resulting, for instance, in white highlights on plastic or varnished surfaces.
To edit a color simply click on the diffuse or specular color button. Select the "-->" button between the two color buttons to link up the diffuse color with the specular color. This saves time when editing metal materials with the diffuse color equal to the specular one.

Reflection, Shininess and Roughness

Reflection - Switching on this option includes surface reflections when rendering the object in raytracing mode 264, so that other objects and light-sources are mirrored in the object. The parameter value (0-1) determines how much light a mirror object reflects and is of crucial importance for the mirror attributes. A mirroring object having, for example, a reflection value of 0.3 only reflects 30% of the light striking it from other objects. Metallic objects, for example, generally have a very much higher reflection value of 90% or more and only very dark basic colors for the diffuse reflection.

A metallic object can only attract attention if there is an environment reflecting on its surface. If you do not want to model a complex 3d-environment then you can effectively apply a panoramic environment map 394 enclosing the whole scene and reflecting in all mirroring surfaces

Shininess - This parameter specifies how much of the light coming from light sources is reflected, producing these nice highlights on a surface.
Shine Width - A value of 0-1 can be entered via this parameter box that determines the radius of the highlight on a reflective object.

Roughness - Adds an amount of unevenness to the object's surface. The higher the value, the rougher the surface is rendered. With a value of 0 the surface finish is completely smooth.

Transparency and Refraction

Transparency - With this option selected, the object appears to be transparent. The transparency value (0-1) determines the relationship of diffuse surface reflection to transparency in transparent objects - the amount by which the object is truly transparent. A value of 0.7, for example, identifies the following: 70% the surface intensity of the object is determined by the light-share that comes through the object from behind and is filtered by the material color. The remaining 30% is decided by the diffuse reflection of the object. The closer this value approaches 1, the more transparent the object becomes. At a value near 0 the object is entirely opaque.
In addition there is the portion of the intensity that originates from reflection (if reflection has been switched on for this object). Reflection in transparent objects is, however, more complicated to calculate than on opaque bodies. With transparent objects this depends jointly on the angle at which the light occurs, and also on the refraction coefficient of the object. These values are then modified with the reflection coefficient to produce the final reflection-value that is used.
If, for example, you stand directly before a display window, you see comparatively little reflected in the pane - with glass and a vertical light-incidence of about 4%.
On the other hand, if you look at a very much more acute angle, the reflective ability of the pane causes it to act almost as a perfect mirror.

Refraction
The photo-realistic calculation of real refraction in transparent objects is performed only when rendering pictures in raytracing 264 mode. Optic density determines the refraction of the light-ray in transparent objects. A value of 1 corresponds approximately to the value for air, so that there is no refraction of the viewing ray worth speaking of. In comparison, the refraction value in water is approximately 1.33.
When a light-ray enters a medium of a different optic density then also the phenomenon of total reflection can occur. If a light ray passes from an optically dense to a less dense medium and the incidence-angle of the light exceeds a given material-dependent angle, the light is totally reflected. In a closed body total reflection can occur. For example, a fiber-optic element is produced from thin fiberglass that includes an optically low-density layer at the surface and relays its information through total reflection of the light pulses at the outside-wall.

Some refractive indexes:

Air	1
Water	1.33
Glass	1.5 - 1.9
Rock salt	1.54
Diamond	2.47

A Special Case - Bitmaps On Top of an "Invisible" Transparent Material
If a totally transparent material (color white) without reflection or refraction (refraction = 1) is applied as the basic material of an object and you add an additional bitmap material on top of it, then this bitmap serves as a mask for the object sufrace, this is, the object becomes totally invisible at all places where it is not covered by a color of the bitmap material. You may say that you did not expect to see anything, since the basic material is totally transparent. But the important difference is that only a masking operation is applied to the object, no additional recursive search rays have to be traced as for colored refractive transparent objects or for mirroring surfaces. Instead the surface of the masked area of an object will be simply ignored as if it were not existent. See the material introduction "Using Special Material Blends to Mask and Cut Out Surface Areas 300" for some examples of use.

Glow

If the <Glow> option is switched on, the object maintains the color assigned to the object - independent of illumination from other light-sources. The reflected and mirrored light-share from other light-sources is additional. However, the object is not considered as a light-source, i.e. there is no illumination of other objects or shadow casting from the object's glow (do not confuse this with the object attribute <Object = Area Light Souce 290 > which changes any object into a real area light source).
Glow Intensity - The glow parameter varies the strength of the self-luminosity - with a self-luminosity of 0 the object behaves exact as all other objects, therefore the color rendered uses only reflected and mirrored light from the background lighting. At a value of 0.50, half of the inserted color-brightness is used as self-illumination, and at 1.00 the object shines with the full intensity of the inserted color - again plus the reflected and mirrored light shares.

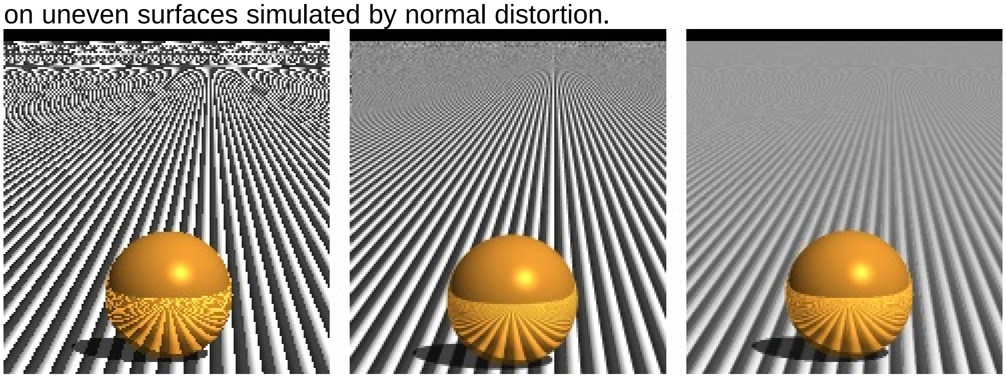
Halo

The halo effect is the natural addition to the glow function. It creates a halo of light enveloping the outline of the object. You can use this effect to simulate atmospheric halos around planets or, for instance, to create swirling swarms of glowing particles.
The halo effect is produced entirely at post processing by means of object buffers and z-buffers after the rendering of the picture. Therefore you can't see halos through transparencies nor are they mirrored by other objects. The radius of the halo is defined in pixels and can be set via the "Pixel Radius"-parameter. Next to it you can select the appropriate halo color. In case the glow effect is also applied you should adjust the halo color to the object color.
Overlay Object - If this button is switched off, the halo is only drawn around the object, otherwise the outline of the object will be partly overlaid with the light halo, similar to the reflections and filtering in a planet atmosphere illuminated from behind.

Examples: On the left 4 NURBS area lights were provided with a little additional light halo surrounding the light areas. The picture in the middle demonstrates a combined scan-halo effect - the head, with activated object halo, is uncovered slowly in an animation by a simple box object which gives the impression of an materializing head (see also project file "\projects\headhalo.cmo"). The picture on the right shows some glowing meteors with halo effect.

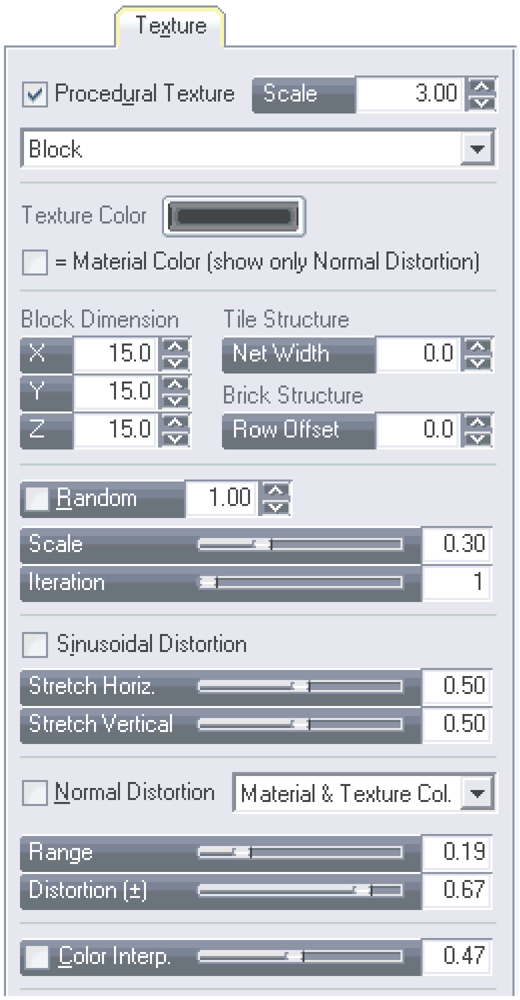
Texture Blur

When rendering objects lying deep in the background, a single pixel of the screen obviously cannot display all the texture details represented by this screen pixel. Using only a single hit point from the object's surface would be like picking a color at random from the object, resulting in a noisy and flickering appearance. This will be even more disturbing when animating the scene. You could of course reduce this effect by applying a higher oversampling rate (antialiasing 264), but that is very expensive in rendering time and at great distances, e.g., at the horizon of planes, up to thousands of subpixels have to be computed.
Instead, we apply a special filter to blur the texture pattern with increasing distance, thus reducing the noise at almost no extra cost in rendering time. The blur function will also smooth the effect of surface normal distortion 313 with increasing distance, for instance, the normal distraction of procedural textures, landscape textures or the waves function. This is necessary because disturbing noise is not only generated from colorful texture patterns but also from the flickering light reflections on uneven surfaces simulated by normal distortion.



Example: The illustration on the left shows a simple sphere standing on a striped plane. Normal distortion was switched on for the stripe pattern to highlight the problem. Neither anitaliasing nor texture blur was applied for the rendering of the first picture. The noise caused by the texture pattern is clearly visible in the background. For the second picture antialiasing was switched on at a maximum quality level of 4 (25 subpixel per pixel). The quality of the picture increases dramatically but so does the rendering time. The illustration on the right shows the same scene with a low level of antialiasing (level 1 = 4 subpixel per pixel) and the textur blur filter switched on. As a result you get the best quality and the rendering is much faster because of the lower oversampling rate.

13.3 Material Dialog - Procedural Textures



Choose the <Texture> tab in the document header of the Material dialog 300 to insert the parameters for a procedural texture.

Procedural Textures

Procedural textures are usually mathematical structures that generate a three-dimensional pattern - the coloring of an object results from the position of the surface of the object in relationship to the three-dimensional pattern. They enable simple-to-describe patterns like grid, block, and ring structures but also very complex patterns generated from fractal techniques to simulate grained wood, marble, rock or even multi-layered landscape textures.
Furthermore, procedural textures are not restricted to the manipulation of the surface colors. They also allow the surface normal of the object to be influenced, so that you can simulate surface irregularities.
The structure of the different texture patterns follows the principles below:
Starting from the center of the texture axes of the object, a three-dimensional pattern is calculated along the texture-axes. The object is therefore "within" this 3-dimensional pattern, which unrolls over the entire surface. The color of a point on the surface of a textured object is simply from the corresponding point within the 3D-matrix. By moving, scaling or rotating the texture axes system 197 created for each object when assigning a material to it, you simultaneously move, scale or rotate the 3-dimensional pattern. This is achieved in the three corresponding work modes "Move Texture 197", "Scale Texture 206" and "Rotate Texture 211".

Activate a Procedural Texture

Simply switch on the <Procedural Texture>-Button on top of the page to provide an object with a mathematical texture pattern.

Scale a Procedural Texture

Once you have correctly adjusted the size relationships for a procedural texture you do not need to repeatedly adjust the total size for all parameters. You can use the <Scale> parameter beneath the <Procedural Texture> button to apply a global scale to the texture pattern. This global change in size will appear on all objects referencing to this material. This is in contrast to resizing a texture by scaling 206 the individual texture axes system 197 assigned to an object. When scaling the texture axes system belonging to an object, the texture pattern will be resized only for that particular object.

Material Color, Texture Color or Color Range

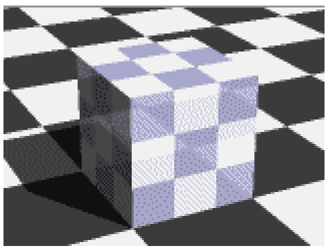
Most procedural texture pattern consist of two basic colors, the material color and the texture color. The basic material color 309 (diffuse and specular reflection) is input on the material page of the material dialog. The texture color is part of the respective texture pattern and is defined here on the procedural texture page. Only exception is the color range texture. Instead of the texture color then a color range is applied and the basic material color is ignored.

Texture Color = Material Color - This option deactivates the texture color for procedural textures. Internally, the basic texture pattern is still calculated, for instance, a block texture, but you will see a surface with only a single color because all parts of the texture pattern are painted with this color. What's the use of it you ask? Well, the normal distortion function can still be applied to the underlying pattern function, e.g., to add a raised tile structure from a block texture or just to add some irregularities to the surface with random normal distortion. The function just saves you time when you want to add such structures to a plain-colored surface by setting the texture color automatically to the material color. Of course you could also do it manually.

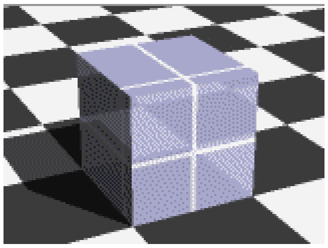
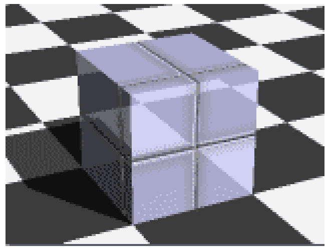
Texture Patterns

In the select box beneath the <procedural texture> button you can select the desired pattern. A set of selection parameters - dependent upon the pattern selected - appears underneath the select box.

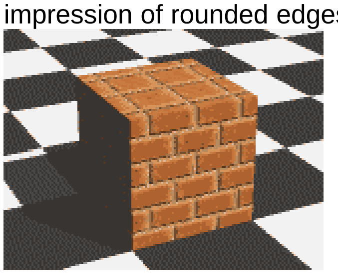
Block Texture



The block-texture is similar to the texture of a chessboard, except you not only get a structure in the horizontal direction, but also vertically.
Furthermore you are not restricted to a cube pattern. The dimensions of a block can be input individually for each axis with the "Block Dimension" - X, Y, Z parameters producing a rectangular block pattern.

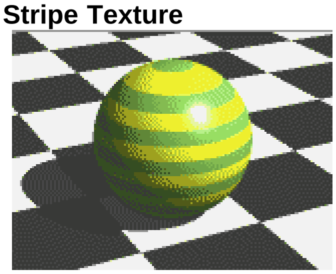
 

Next to the block dimension is the "Net Width" parameter. If this value is not 0, instead of the checkerboard-like pattern you now get a tile-like structure. The X, Y, Z parameters are again responsible for the dimensions of the tile-block. The net-width parameter determines the separation between the individual blocks. If a normal distortion is also applied to the block pattern an impression of rounded edges at the rims of the tiles is achieved. See also: normal distortion 313.



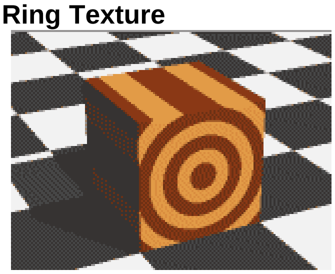
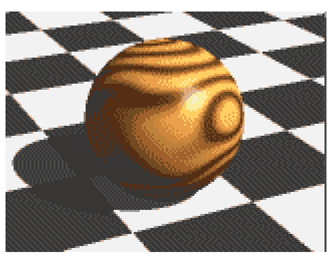
There is another parameter left for the block pattern - the "Row Offset" moves every second block row to the side by a given amount, thus giving the impression of a brick structure. The picture shows an example of this brick texture (x = 34, y=16, z=34, net width = 2, row offset = 17). Again, a surface normal distortion simulates rounded edges.

Stripe Texture



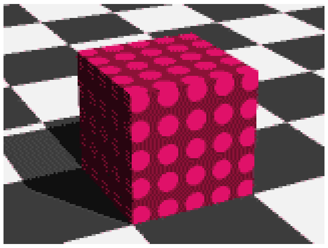
To define a stripe texture, you need only to edit 2 parameters - the widths of the two colors forming the stripe. Additionally you will still see the three X, Y, Z buttons. With them you can determine which of the three object-axes the stripes should follow.

Ring Texture

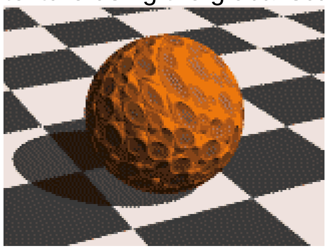
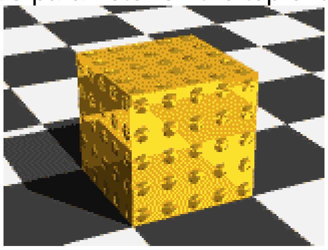
 

The parameters for the ring texture are exactly the same as for the stripe texture.
The two parameters this time determine the widths of the two colors forming the rings.
The X, Y, Z buttons determine the axis that acts as the center of the concentric rings. If with the ring texture, you additionally apply a distortion of the texture pattern and a color interpolation of the two texture colors, then you can obtain wood like structures as in the right picture above.

Sphere Texture



This texture behaves similarly to the tile-texture but, instead of blocks, spheres are used whose spacing is arranged through the Distance parameter.
The Diameter parameter defines the diameter of the individual spheres.
However, do not be surprised if you see only small circles on the object, despite having entered a large diameter - your object's surface is intersecting just the surface of the texture-spheres, which are a three-dimensional spatial formation. If this is the case, you can slightly reposition the texture using the Move Texture 197 function or change the Distance parameter or simply scale the whole texture using the global scale parameter on the top of the page.

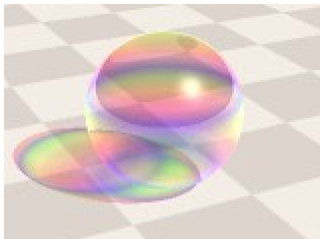



Two further examples:
The left picture shows a sphere texture combined with a texture pattern distortion and color interpolation. The block in the right picture consists of two identical colors for the material and the texture color. But by switching on the normal distortion the sphere texture becomes clearly visible somewhat resembling protruding rivets.

Color Range Texture

The color range texture resembles the stripe texture. Only, this time the colors of the material and texture are replaced by a strip of a range of colors. If the <Repeat Pattern> button is selected then one color range strip follows after the other, otherwise the areas above and below the color range strip are painted in the start or end color of the color range, respectively. When you operate the <Fit to Object> button then the width of the color range will be automatically calculated so that it precisely covers the whole object.





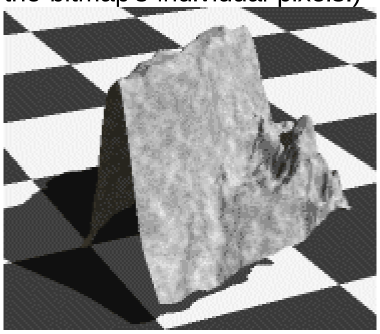
Three examples: The picture on the left shows a color range texture (combined with texture- and normal-distortion) that is intended for a landscape object with several layers of sedimented rock.
The illustration in the middle shows a color range with fractal texture distortion and glow, thus resembling lava rock.
The picture on the right demonstrates a rainbow color range on a transparent sphere to create a somewhat iridescent surface.

Fractal Noise Texture

The fractal noise texture needs only a value for random distortion in addition to the material and texture colors to produce texture patterns. This random texture distortion is based on fractal

algorithms and can be applied to all texture patterns.
In the fractal noise texture, the value provided by the fractal algorithm is used to mix the texture colors while the other texture patterns use this value for a distortion of the texture pattern itself.

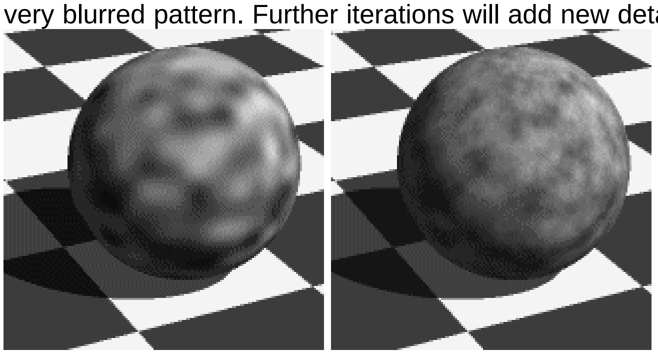
Fractal textures are the basis of every good landscape simulation. With help of iterative procedures, highly-detailed patterns resembling ground structures can be generated. This texture provides high detail at virtually any resolution. You can zoom in on a surface and new details will constantly be discovered. (This is in contrast to a bitmap, for instance, where a camera zoom would begin to show the bitmap's individual pixels.)



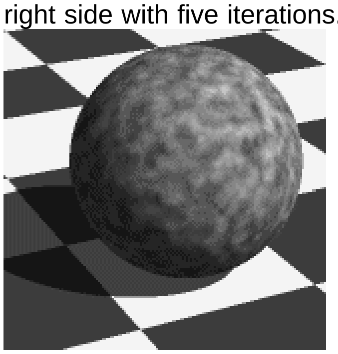
Example: fractal rock texture

A fractal texture in MR-3D Designer is described by **two colors**, the number of **iterations** and a **scale** parameter.

The number of iterations defines the level of detail for the texture. A single iteration will result in a very blurred pattern. Further iterations will add new details to the fractal texture.

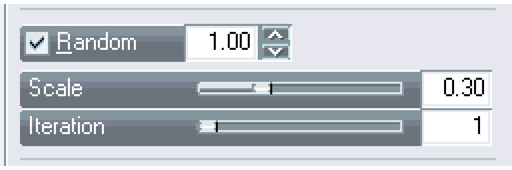


The illustration shows the same texture pattern, on the left side with a single iteration and on the right side with five iterations.

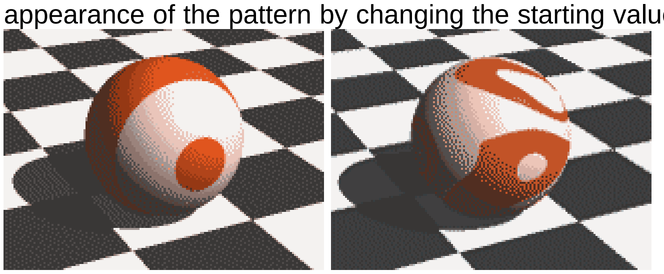


Here again the same pattern, this time with a higher scale value for the fractal noise. The individual fractal patterns are rendered close together or wide apart depending on the scale value entered.

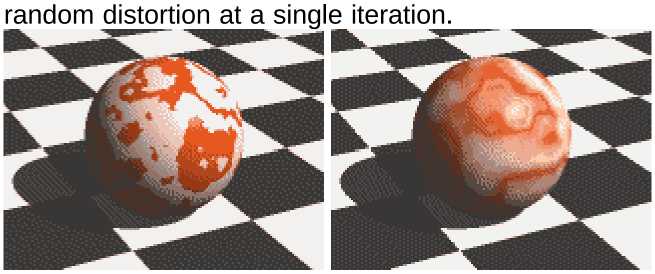
Distorting Textures



If you switch on the **<Random>** button random distortion is superimposed on the base pattern.
For the distortion of the texture exactly the same techniques are used as described above for creating a fractal texture but this time the fractal value is used to distort a pattern instead of finding a suitable texture color.
Low iteration values will result in a smooth and rounded distortion of the pattern - higher values lead to a more chaotic distortion.
To reproduce the "random" distortion you can input parameters between 0 and 1, which initializes the random number generator every time with the same starting value. You can vary the appearance of the pattern by changing the starting value.



On the left is a sphere with a procedural ring texture. On the right the same sphere with activated random distortion at a single iteration.



The next picture shows the ring texture after 5 iterations. The pattern is certainly more chaotic but it's still lacking the smooth color transitions we know already from the fractal noise texture. That's no problem since there is an additional color interpolation button at the bottom of the procedural texture page. When this option is selected a smooth transition between the two texture colors is calculated. Hard to believe, but the marble sphere in the right picture above presents the same ring texture pattern as the one in the picture on its left, only color interpolation has been switched on.

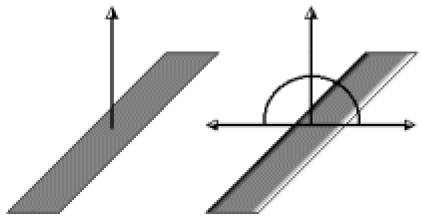
Sinusoidal Distortion

Switch on the **<Sinusoidal Distortion>** button if you want to superimpose a sine wave formation onto the pattern. Note that this is a 3-dimensional waveform. Superimposed on a stripe pattern on a cube-shaped object, for example, this produces an undulating pattern on 4 of the 6 sides of the block. On the other sides the wave peaks and troughs penetrate the cube sides and form circles or circular rings.
A full sine wave runs over an interval of 2 x pi (about 6.28). By changing the **Stretch** parameters you can make the waveform spread in the horizontal plane or in the vertical, respectively.

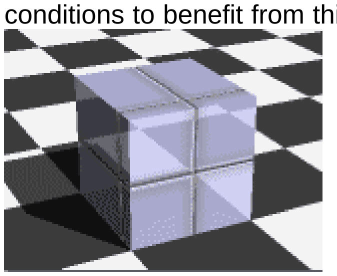
Ring texture with superimposed sine wave.

Surface Normal Distortion

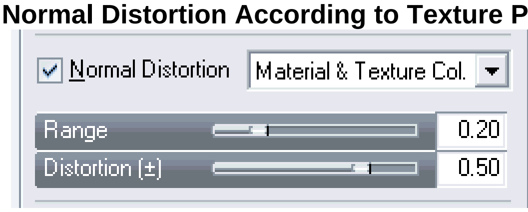
The surface normal is a vector standing vertical to the surface and is used to determine the surface brightness in respect of the light-incidence and object color.



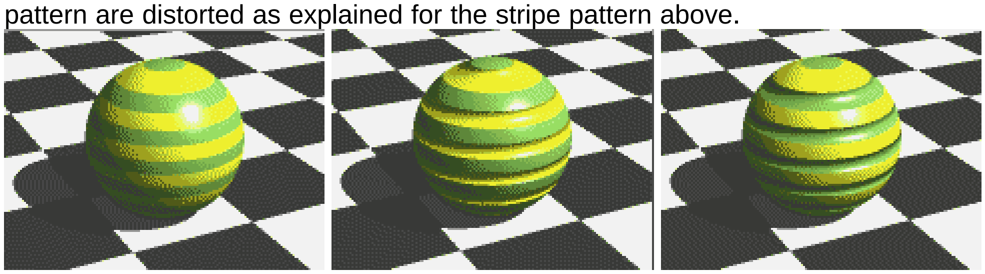
Distorting the surface normal allows a raised appearance to be added to the surface structure. If, for example, you put in a stripe texture and switch on the distortion of the surface normal, this distorts the normals towards the edges of the stripe-pattern. In this way, the calculation of the light intensity creates the impression that the surface falls away at the edges of the stripe.
This normal distortion enables a wide range of possibilities. It would be very laborious, for instance, to produce a tiled background. Without normal distortion for every single tile you would have to create and position every individual tile object. Instead, you can create them all with only a single object formed from a rectangular block, with a block texture and normal distortion assigned to the block pattern.
But keep in mind the raised appearance is only simulated and dependent of the light incidence, the surface itself remains unchanged. Therefore you should always concentrate on propper light conditions to benefit from this effect.

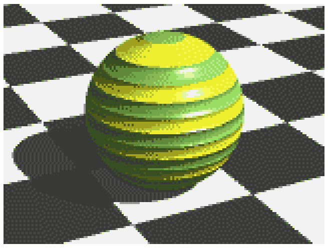


Normal Distortion According to Texture Pattern or Random Normal Distortion



In the select box next to the <Normal Distortion> button you can choose between different distortion modes for the texture. For all ordinary patterns (block, stripes, rings and spheres) you can choose a normal distortion according to the texture pattern alignment. In this case only the edges of the pattern are distorted as explained for the stripe pattern above.





The illustration shows a sphere with a ring texture, material color yellow and texture color green. The second picture demonstrates normal distortion for the material color only ("Material Color" selected in the select box). When you choose the "Texture Color" entry in the select box then only the green stripes will be distorted (3rd picture). The last picture presents the sphere after "Material & Texture Color" has been chosen for the normal distortion.

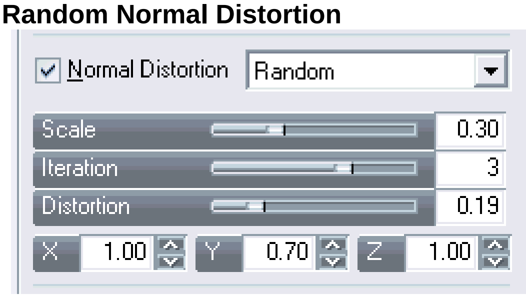
Range

With the Range parameter you can restrict the area over which the normal distortion applies. Take for example the tile pattern. With a small range only a narrow area of the edges is distorted. With a value of 1 the distortion stretches over the whole tile resulting in a large bump on the tile.

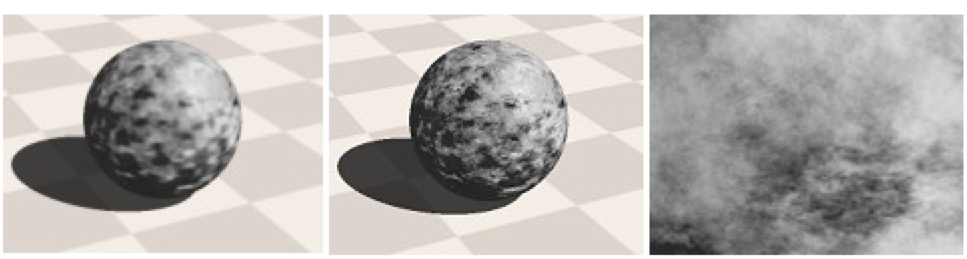
Scale (-1 to 1)

With this parameter you can influence the apparent height of the normal distortion. At low values the normal at the edges of the side curves only a little, at higher values it is more. An interesting possibility is to also use negative values.
Again, take the block pattern for the demonstration. Using a positive value the normal is distorted so the tile pattern appears to be raised. Use a negative value instead and the direction of the normal distortion turns and the pattern appears to be indented.

Random Normal Distortion



In addition to the three pattern-controlled distortion modes there are two modes for random normal distortion. When selecting a random mode the Range parameter is replaced by three separate parameters, each controlling the amount of distortion for the individual x-, y- or z-texture axis.
Scale - With a high value the distortion pattern is very narrow and at low values the random pattern becomes more stretched.
Iteration - With each iteration step the normal distortion reveals more and finer details.
Distortion - Controls the basic amount of distortion for all axes. This initial distortion is then adjusted by the three separate scale values for the individual x-, y-, z-axis.



The illustration demonstrates a fractal rock texture with a random normal distortion. The picture on the left shows a distortion pattern with only 1 step of iteration. For the next picture 3 iterations were calculated which is indicated by a much more detailed and sharper relief structure of the surface. Even if you adjust the zoom to an almost microscopic view, more and more details are revealed when zooming in on the surface (picture to the right).

An example for different distortion values along the texture axes: x=0.75, y=0.18, z=1.0. Here the y-axis is undervalued so that the ridges move vertical. An ideal texture for the bark of a tree.

High random distortion values combined with a green fractal noise texture. These are the basics for a grass texture. Although the noisy appearance is best suited for a chaotic grass layer the noise becomes very disturbing in animations. There you have to reduce scaling and distortion as well as combining it with the texture blur 309 and antialiasing 264 functions. It can also be usefull to render only a top view of the pattern into a bitmap which is applied then as a bitmap texture with Mip-Mapping 328 switched on.

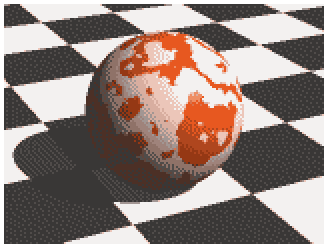
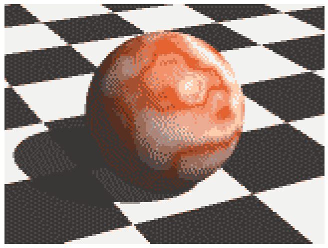
Random & Texture

There is a last normal distortion mode that combines random normal distortion with pattern controlled distortion. This is mainly intended for color range textures.

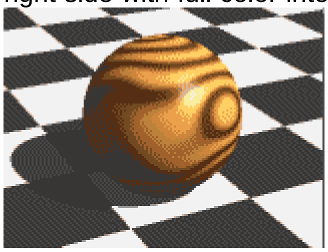
Color ranges are best suited for sedimentary rock layer textures. With additional Random & Texture-controlled normal distortion you achieve a normal distortion that follows the alignment of the color range stripes and also overlaid by a random distortion, thus better representing the sedimentary and ridged horizontal structure. To achieve this result the alignment of the color range texture must be related to the distortion of the individual texture axes. If, like in the picture, the color range runs along the y-texture axis then the random normal distortion for the y-axis should also be high, while the x- and z- parameters should be at a very low level.

Color Interpolation of the Texture Colors

The last box on the "Texture" page contains the parameter for color interpolation. If you switch on this option the two texture colors fade from one into each other (available only for block, stripe, ring, or sphere textures). The parameter only controls the area over which the color transition is applied.

Here again the example of the ring texture, on the left side without color interpolation and on the right side with full color interpolation over the whole ring width (color interpolation value = 1.0).

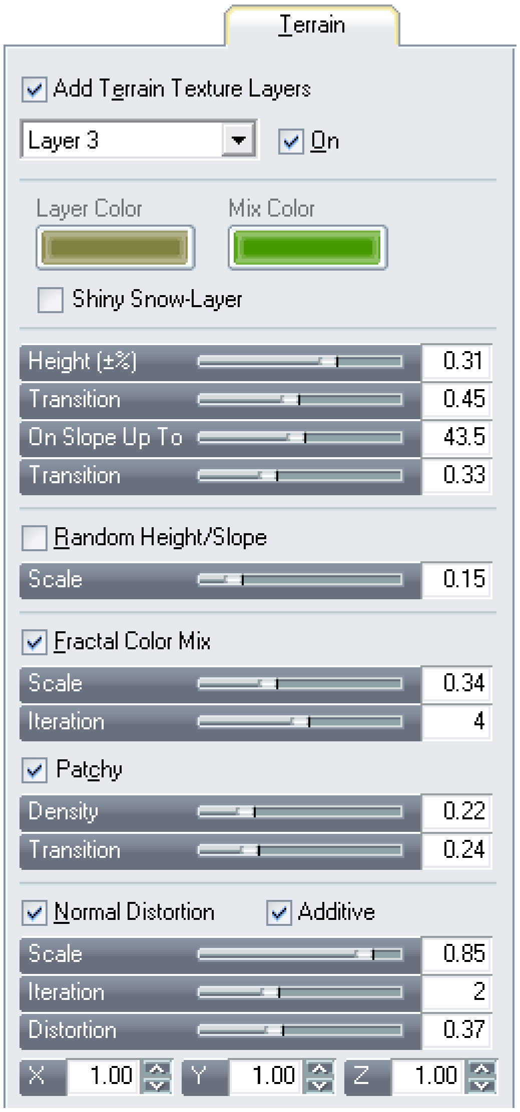


For this wood texture only half of the ring width was interpolated (color interpolation 0.5), thus only the edges of the rings fade into each other.

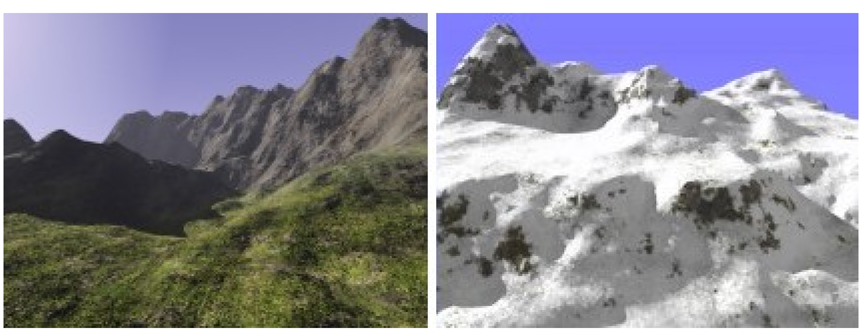
Well, I think you have realized that the possibilities of procedural textures are countless. Simply try out and play around with all the parameters - you can't do anything wrong and the preview window always gives a fast representation of the changes.

The next page "Terrain texture layers 323" will introduce an additional set of procedural texture layers especially designed for landscape textures.

13.4 Material Dialog - Landscape Textures

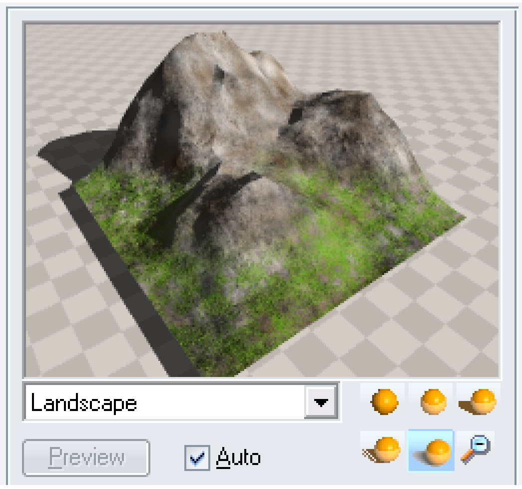


When you select the <Terrain> tab in the header of the Material dialog 300 a side providing special landscape texture layers comes to the fore.



Rock, soil, grass and snow in one single texture - the result of a combination of a fractal base texture for the rock overlaid with additional texture layers added on the <Terrain> texture side.

On this side you can define up to 3 additional texture layers overlaying the object's basic texture. The texture pattern used to build each individual layer is again based on the fractal noise 313 texture described in the previous chapter on procedural textures 313.
These additional texture layers are applied dependent on the slope angles and the height of the surface. For instance, you can define a white snow texture that covers only areas that lie high in the mountains and have moderate slopes. A random distortion and blending parameters provide additional irregularities and smooth transitions. See also: Tutorial - Landscape design 85



For editing landscape materials a special landscape primitive is provided in the preview selection box. This is especially important, since the texture structures have to be adjusted on an object with a comparative dimension rather than on a small sphere.

The Parameters:

Add Terrain Texture Layers - - Each layer can be switched on or off separately. With the <Add Terrain Texture Layers> button a global switch is provided to switch on or off all layers simultaneously.

Layer - Selector Box - the layer selector box lists the 3 possible layers in the same order in which they are applied to a surface. Simply select a layer in the select box to edit its parameters.

On - Activates the selected texture layer.

Layer Color and Mix Color - If the <Fractal Color Mix> option is switched on then a fractal color pattern 313 is calculated from these two colors, otherwise only the layer color is applied.

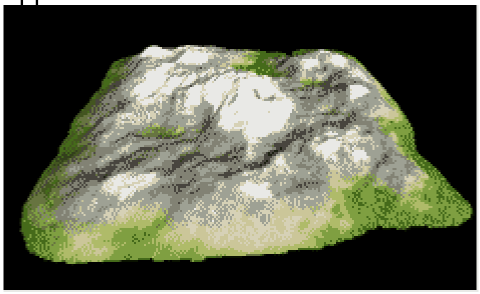
Shiny Snow-Layer - It is not sufficient to just assign a white color to a texture layer to simulate a snow covered surface. The manyfold reflections and the light distribution within the snow crystals have also to be taken into account. Simply switch on the <Shiny Snow-Layer> button and the selected texture layer will be rendered with additional shininess and selfglowing properties to let the snow sparkle in the direct light and glow somewhat in the darker areas where no direct light falls in.

Height (±%) - The Height value describes the fraction of the overall object height up to which the texture is applied. For instance, with a value of 0.5 the texture layer would cover only the lower half of the object. A special case is the snow layer. Usually snow lies in higher areas and disappears in lower and warmer heights below the snow line. To take that into account you simply enter a negative value for the height parameter. The height calculation is then reversed, starting from the top of the mountain and running downwards to the ground.

Transition - Calculates smooth transitions with the ground at the edges of a texture layer. This parameter is available twice, on the one hand for the height-controlled blending and on the other hand for the slope-dependant transitions.

On Slope Up To - An angle up to 90° specifies the slope up to which a texture layer will cling to the ground.

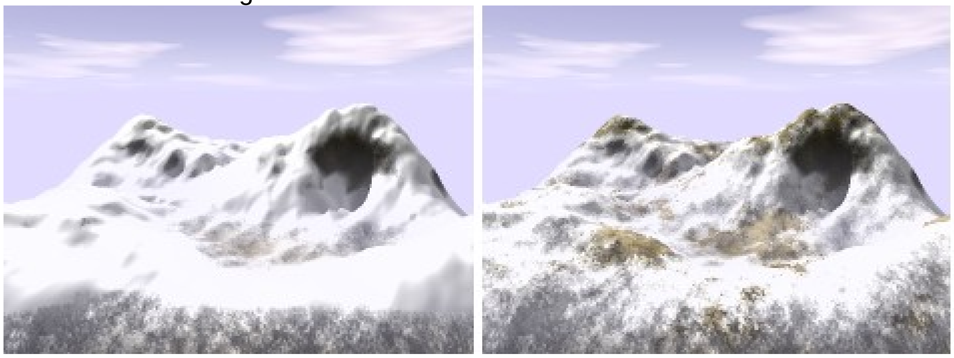
Random Height/Slope - Adds randomness to the height and slope calculation to provide further irregularities. Additionally, texture layers can overlap each other when a random distortion is applied.



In this illustration the grass layer covers the soil layer. Nevertheless, by using appropriate values for the Height, Slope and Random parameters the soil breaks through the surface in many places.

Fractal Color Mix - If this option is activated a fractal color pattern 313 is calculated from the Layer Color and the Mix Color, otherwise only the Layer Color is applied.

Patchy - This option adds even more details and variety to the landscape texture. Texture layers with this option activated become a spotty appearance, no longer covering up the layers underneath but mingling with them instead. The **Density** -parameter controls the frequency of the patches, the higher the density the less gaps showing the underlying textures will appear. The basic size of the patches is dependent on the fractal pattern of the texture layer and therefore is controled with the "Fractal Color Mix"-<Scale>-parameter. The size of the patches can be adjusted with this slider even if the fractal color mixing is switched off. The **Transition** parameter ensures again smooth transitions at the edges of the texture islets.



These pictures show the difference, on the left a snow texture without the <Patchy> function and on the right picture the same texture only this time with <Patchy> switched on for the snow texture layer.

Normal Distortion - Corresponds to the random normal distortion 313 described in the previous chapter on procedural textures 313. Each texture layer has available its own normal distortion. You can choose if the normal distortion of an underlying texture layer is covered by the upper layer or if the normal distortions of the texture layers will be summed up. If you switch on the <Additive> button then with each activated texture layer new details are added to the bumpiness of the surface. However, if you add a snow texture layer you should either switch off the normal distortion (all areas covered by the snow will cover also the unevenness of the underlying rock and soil texture layers) or you just apply a normal distortion with a high y-distortion value and low x-, z- distortion values. This will emphasize the distortion on a horizontal level, giving the impression of overlapping snow drifts.

13.5 Material Dialog - Waves

Select the <Waves> tab in the Material dialog 300 to edit the parameters for water textures.

Switch the **<Waves>** button on if you want to generate a wave-like effect on the surface of the object. This function is especially intended for planes 186 in combination with landscape objects 177. The waves are calculated only in the x-, z-plane of the texture axes of an object, so if you apply this function also to other objects than the plane object be sure to check the alignment of the procedural texture axes via the "Rotate Texture 211" menu.

Water currents, wind and opposed wave directions often create a complex interference pattern of waves of all kinds of amplitudes. Similar to a fractal pattern 313, waves of a smaller amplitude - driven by the wind - seem to be riding over the surface of the bigger waves which move with the main current. Consequently MR-3D Designer also applies a combination of a main basic frequency and additional harmonic frequencies to calculate the normal distortion 8, which simulates the water movements.

Basic Wave 

You can choose between two different types for the creation of the main basic wave

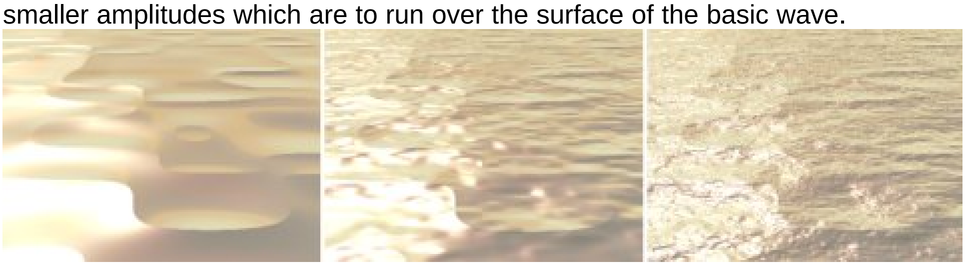
If you choose the rounded icon then also a more rounded and bumpy appearance of the water surface is created (picture above on the left). This is a correct setting for smoothly-babbling brooks and rivers. When selecting the second icon a basic wave function with somewhat sharper wave crests is calculated. This is appropriate for more troubled water on windy days.

Scale - Individual waves are rendered very close together or very wide apart depending on the scale value entered. This can be best observed in the preview window in "Camera, complete scene" preview mode.

Stretch - Waves can be stretched crossways to the flow direction to simulate broad, wind driven bands of waves.

The **<Distortion>** parameter defines how greatly the surface normal 8 is bent by the wave effect - whether the waves are flat or high.

Additional Frequencies - Here you can enter the number of additionaly calculated waves of smaller amplitudes which are to run over the surface of the basic wave.

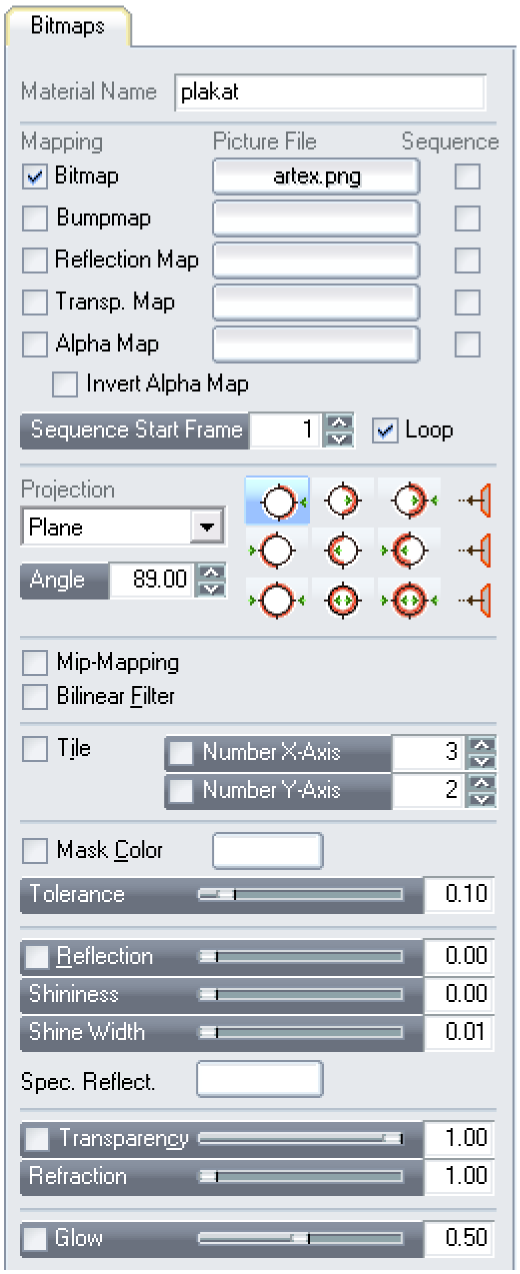


The picture above on the left shows the basic wave without any additional frequencies. Only when increasing the **Iteration** number for additional frequencies **-** two for the picture in the middle and five for the picture on the right **-** the impression of realistic water becomes apparent. The **Distortion** parameter controls again the strength of the added normal distortion.

Animated Waves

To create an animated wave flow you just need to activate the **<Animate>** button.
The **<Velocity>** value controls the speed of the wave movement.
With **<Flow Direction>** the movement direction of the wave field is input.

13.6 Material Dialog - Bitmaps



If you create a new bitmap material by operating the corresponding <New Bitmap Material> button underneath the <Project> material browser then the bitmap register containing all parameters regarding a bitmap texture appears in the middle of the Material dialog 300.

Bitmaps

Bitmap textures are by far the simplest method of generating complex surfaces. For this purpose each point on the object's surface takes a corresponding point on a pixel graphic and the pixel's color simply used to calculate the object's color at that point. You can create material surfaces very easily with the help of a paint program or use pictures imported with a scanner to project logos or surface structures. You can, for example, build a city of skyscrapers from rectangular blocks and simply project the building fronts onto the blocks. Or, on an interior furnishing, scan a "da Vinci"

and project the picture onto a "canvas" object. The list of the examples is endless.
Additionally you can make use of picture files to influence the appearance of an object in an entirely different way. The brightness values of a picture can be interpreted as information to influence material parameters and thus control reflectivity (reflectivity maps) or transparency (transparency maps). The pixel values even can be interpreted as a value for distortion of the surface normals of an object and thus can simulate surface irregularities (bumpmaps). Furthermore you can apply alpha maps or define a masking color to hide the background area of a picture, e.g., to project only the silhouette of logos, writings or other figures.
Up to 16 materials can be assigned to an object or to a frozen facet selection - procedural textures as well as bitmap textures which you can freely mix with the underlying materials or just put one bitmap on top of the other as you would affix several lables on an object.

Aligning Bitmaps on an Object

When you assign a new material to an object then always together with the material reference a new texture axes system 197 is generated for the object. You can use this texture axes to adjust materials indivdually on every object, for example, to align a bitmap texture on an object. Moving 197 a texture axes system will reposition the origin of a bitmap, while rotating 211 the texture axes will change the direction of a bitmap projection. Finally, scaling 206 the texture axes system will also resize the bitmap.
See also - Moving Textures - Texture Axes System 197

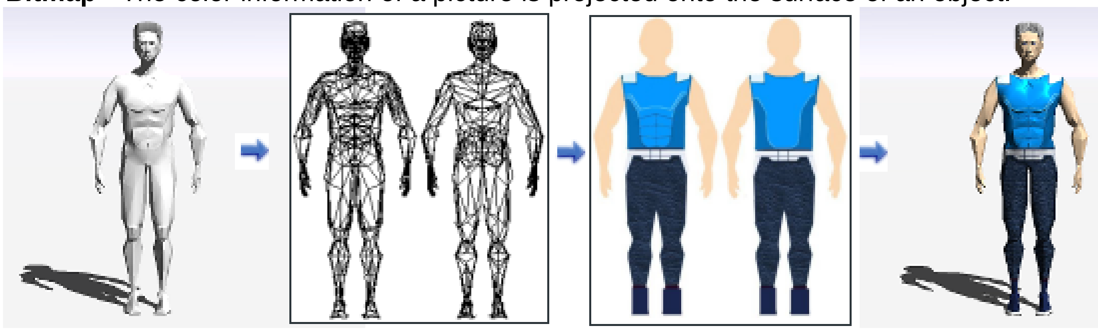
Choosing a Bitmap Type and a Corresponding Bitmap File or Animated Sequence

At the top of the bitmap page, under the material name, you find 5 rows where you can switch on the 5 different bitmap types that can be combined in each bitmap material. Integral part of each bitmap material is of course the <Bitmap> which gives the picture information for the coloring of a surface. But its not always necessary to include this picture file in a bitmap material. You can just as well choose only one or several of the other bitmap types, for instance, if you only want to control the transparency of an object's surface with a transparency map.
Alongside the buttons to switch on the different bitmap types you find another button to call up the file select box where you can choose the desired image for your bitmap texture.
Bitmap file format: MR-3D Designer supports the following picture formats: *.BMP, *.JPG, *.PNG, *.PCX, *.TIF and *.TGA
Bitmap paths: The program searches for bitmaps first in the current project folder. If the picture is not present there, the bitmap library folder (set up during the installation process of MR-3D Designer) is searched next. There are 6 additional paths for searching bitmap files, which can be stipulated in the Program Settings 27 dialog after selecting "Customize" under the "File" entry in the menu bar.
Animated bitmaps - It is possible with MR-3D Designer to project complete picture sequences onto objects while rendering an animation 401. For example, you can easily construct moving and cyclic billboards in a scene, on which a film (bitmap animation) simultaneously runs. Or you can use this function to copy masked sequnences of explosions or smoke animations into the scene. Even bumpmaps, reflection, and transparency maps can be animated, as long as the following requirements are met:

- Pictures must be available as a picture sequence in one of the designated file-paths 27.
- The pictures must be consecutively numbered, as, for example: "PIC0.TGA," "PIC1.TGA," "PIC2.TGA."
- Switch on the **<Sequence>** button in the row of the corresponding bitmap type - if the button is switched off (as normal) only a single picture is projected.
- Simply choose the first picture the sequence should start with as the corresponding bitmap file and the program then automatically recognizes if a picture sequence is available and ties it in.
- You can specify a starting position for the sequence to be played via the **<Sequence Start Frame>** parameter. Before this frameposition only the first picture of the sequence is used for texturing. When arriving at the starting position the sequence is played until the end of the picture sequence. The last picture of the sequence will be used for the rest of the animation unless you have activated the **<Loop>** function for an infinite loop of the sequence.

Bimap Types - Bitmaps, Bumpmaps, Reflection or Transparency Maps

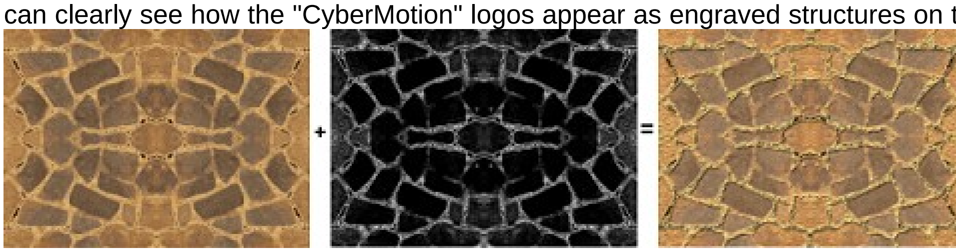
Bitmap - The color information of a picture is projected onto the surface of an object.



The illustrations shows an example for a uv-mapped bitmap material. From the 3D character model a uv-map was created 218 and used as a template to paint in the details for a bitmap texture. Finally the bitmap was applied with uv-mapping as projection mode to texturize the 3D character.

Bumpmap - The brightness values of a picture are interpreted as a value for distortion of the surface normals 313 of an object. Thus you can simulate surface irregularities. For instance, to provide an engraving for an object you simply need to project a picture in bumpmapping mode with the required text onto the object.

Example1: A bumpmap with a black "CyberMotion" writing on white background has been projected alongside the stamp (planar projection) and another time around the shaft (cylindric projection). You can clearly see how the "CyberMotion" logos appear as engraved structures on the stamp.



Example2: The bitmap texture on the left is to be used as a stone wall texture for a rectangular wall object. But without bumpmapping the wall appears more like a flat wall draped with a photographic wall paper. To add the raised structure of the rocks we apply now a bumpmap (illustration in the midlle). The bumpmap was created in an image editor from a copy of the original wall picture by changing it into a grayscale picture and adjusting the brightness and contrast values to emphasize the areas around the rock outlines. The picture on the right shows the final result. The stones are clearly standing out from the wall and are illuminated correctly depending on the light incidence.
See also - Surface Normal Distortion 313

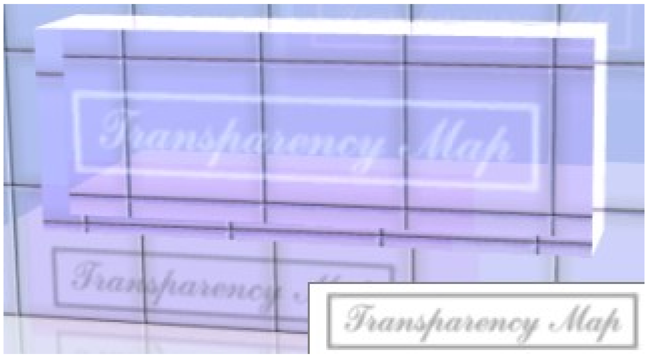
Reflection Map - In this mode the brightness values of a picture are interpreted as a value for reflectivity and shininess. If a color picture is used then the pixel colors also define the specular color 309 filter. Reflection maps can be used to apply a dull and non-reflective bitmap on a reflective material or vice versa. The higher the pixel intensity of the reflection map the higher the reflectivity value. White corresponds to a material reflection value of 1, black corresponds to a non-reflective material.

In the picture above a reflectivity map was used in combination with a bitmap and a bumpmap to produce rusty spots on a reflective sphere (file "\projects\reflection map.cmo"). These textures were used:

The left picture shows the rust texture. The white color of the picture background was masked out using the mask function. The mask color was set to white and a high tolerance value of about 40% entered. Thus the background of the picture becomes transparent and will not be projected on the object. Furthermore, all additional effect maps will only apply in that region not masked out by the bitmap's masking color.
The picture in the middle represents the reflection map. It is just a copy of the rust texture, grayed and darkened using the gamma correction function of an image editing program. The dark parts of the reflection map will reduce the basic reflectivity parameter set for the material. The white background is already masked out by the masking color of the first bitmap texture. Otherwise the white parts of the reflection map would also overwrite the basic material reflectivity and be interpreted as the maximum reflection value of 1.
The last picture shows the bumpmap that was applied to give the rust a rough structure. Again a copy of the rust texture, grayed and somewhat lightened, so that the "engravings" don't become to deep.

Another example: The picture in the middle shows a detail of a bitmap texture to be applied on an skyscraper object. We want high reflective windows on a dull store front. Therefore a copy of the bitmap texture has been created to serve as a reflection map. The facade has been tinted in black (no reflection) and the window areas have been colored in a light blue (high reflectiveness with a light blue filter for reflected light). The left picture shows the outcome when bitmap and reflection map are combined.

Transparency Map - Transparency maps behave in the same way as reflection maps, except that this time the brightness values of a picture control the transparency of an object. Black parts in the transparency map are interpreted as wholly opaque whereas white parts represent total transparency. Transparency maps will only override the transparency value of an already transparent object, that is, you have also to switch on the <Transparency> button for the basic material settings at the buttom of the bitmaps page.



(File "\projects\transparency map.cmo")
The picture above shows a complete transparent block with an additional transparency map projection. The greyish letters in the transparency map are interpreted as only half transparent while the background of the transparency map is white and therefore is be interpreted as having the same complete transparency as the block.

Alpha Map versus Mask Color

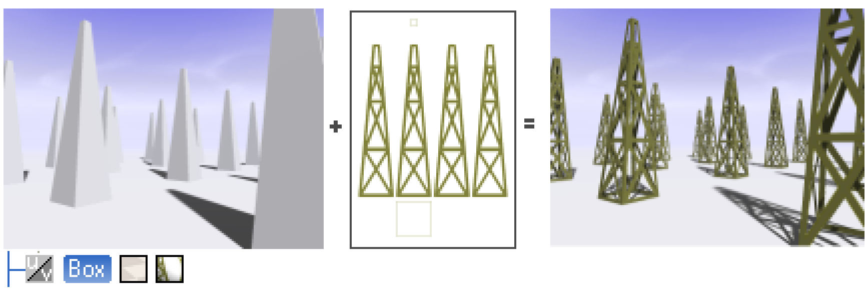
As with a mask color 328, an alpha map can be applied to mask out areas of a bitmap texture. However, in contrast to a simple masking color, the intensity values of an alpha map can also control a smooth blending to underlying material layers. Thus, the intensity value of an alpha map will also override the general blending value of the <Blending> slider 300 with which you can mix several material layers. Black points in the alpha map will be interpreted as wholly transparent, gray colors will control a fade over and white points are interpreted as complete opaque.

Let's see again the example of the graffiti-writing (see Material dialog - Mixing Materials 300). On the left picture you see the basic bitmap texture with green writing on white background projected on top of the procedural wall texture. Now we apply a simple color masking by switching on the <Mask Color> button and choosing a white color for the masking process. Thereupon the white background of the bitmap disappears but also somewhat whitish transitions at the outlines of the letters remain. This is due to the anitaliased smooth transitions in the original bitmap picture where the pixels become a mixture between green and white. You can bypass this problem by adjusting a higher tolerance value for the choosen mask color. With it also the colors lying within a small range near the mask color will be faded out. This is an acceptable solution with bitmaps showing only a small numbers of colors where you can easily determine a unique mask color to separate the background from the picture area you want to mask out. But when applying multi-colored pictures like photographs it will become increasingly difficult to find a unique color to mask out the background, especially if the range of tolerated color values is increased with help of the tolerance slider. In such a case an alpha map is the better solution.

In this example the graffiti bitmap was copied, inverted and converted to a grayscale bitmap. This grayscale bitmap was applied as an alpha map in addition to the bitmap. The background of the bitmap is masked out where the alpha map is black. The greyish transitions from black to white at the boundaries of the letters in the alpha map are interpreted as transparency values for the mixing of the bitmap with the underlying material which results in a smooth fading of the text onto the wall.

Picture Formats with Included Alpha Channel - In the graffiti example two separate pictures were applied, one for the bitmap and one for the alpha map. But some picture formats can include the alpha map as an additional 8 bit alpha channel in one single file (32 bit RGBA instead of a 24 bit RGB file). If you select a 32 bit RGBA picture as a bitmap file, MR-3D Designer will automatically recognize the alpha channel - a dialog pops up and you can choose whether you want to make use of the alpha channel or not. If yes then the <alpha map> button is automatically activated and the name of the bitmap is also entered for the alpha map. The included 8 bit alpha channel will always be used for the alpha map instead of the picture information when choosing a 32 bit file for the alpha map.

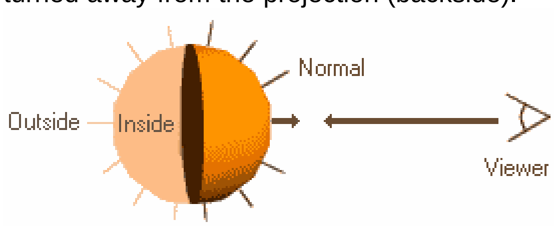
Invert Alpha Map - The interpretation of the greyscale values of an alpha map may vary in different software programs. If you import a picture with an inverted alpha map (which uses white for transparent and black for opaque picture areas) then simply activate the <Invert Alpha Map> button and MR-3D Designer will also use the inverted interpretation for this bitmap.



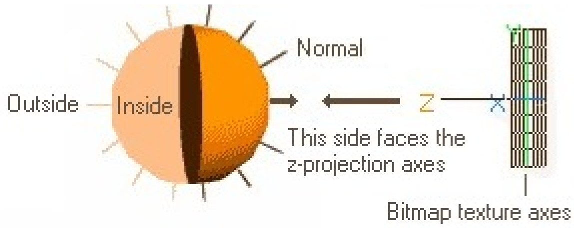
Alpha maps or masking colors can be also combined with an underlying transparent material layer to cut out particular areas from a surface of an object.
See - Using Special Material Blends to Mask and Cut Out Surface Areas 300

Bitmap Projection Planes

A picture can be projected onto an object in different ways. You can apply texture axes based projection modes like planar or box projection, cylindrical or spherical projection or you can apply uv-Mapping to map the pixel coordinates of a bitmap directly to the corresponding vertices of each facet of an object. You can select the required type of projection via the "Projection" select box. When choosing the desired projection mode another choice of icons is displayed next to the select box. Here you can decide if you want the bitmap projection to be visible only on the outside, only on the inside or on both sides of a surface, respectively. Furthermore you can make this decision separately for facets facing the projection direction (front facets of a closed object) and for facets turned away from the projection (backside).



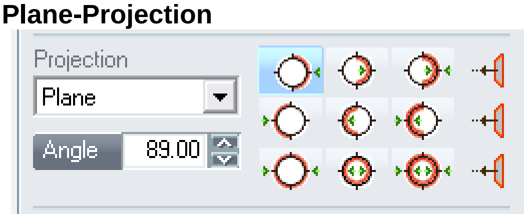
Definition of Inside and Outside: Each facet of an object is eqipped with a normal 8 vector standing perpendicular to the surface of the facet. The side this vector is standing on is the outside, accordingly the other side is the backside of a facet. On construction of an object, the facets are created so that the normals are always facing the outside. But with hollow objects which don't provide a closed shape it is not always obvious which side is the outside and vice versa. The same applies to objects imported from third party programs. May be the object data does not contain any normal information or the normal directions are reversed. In this case you can activate the normal depiction 22 for the viewports and adjust the normal alignment 241 with help of the corresponding function in the Edit Object menu.



Definition of a surface turned to the projection - If the normal of a facet is pointing towards the projection plane of the bitmap's texture axes, i.e, normal vector and z-axis vector are facing each other, then the surface is turned towards the bitmap projection.

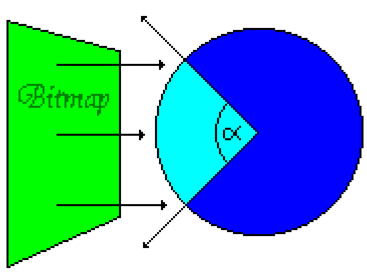
Bitmaps on a transparent surface - If a bitmap is combined with transparency then always a projection mode which applies the bitmap on both sides of a facet should be selected. Otherwise the bitmap will be visible only from one side of the transparent surface, e.g., if you project a lable on a glass front it will not be visible from behind. Also, the lable will cast no shadows on one side because the shadow feeler from behind the glass can't "see" the label on the other side of the glass surface.

Plane-Projection



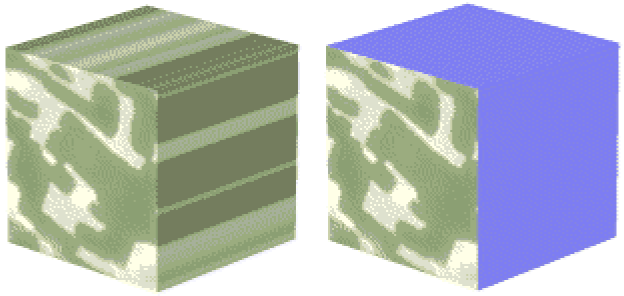
The bitmap-picture is simply projected onto the front of the object. This can be compared with a slide picture projected onto an object. Each picture-point of the picture simply becomes the corresponding object-point on the X, Y plane of the object. There are 9 icons available to determine on which side of an objects surface the bitmap is to be projected. So, if the z-axis of the bitmap's texture axes points to an object's surface then you can recognize from the green arrows and the red emphasized inner- and outer walls on which side of the surface the projection will be visible.

Projection Angle - With this parameter you can control the angle of incidence, under which the plane projection is "visible" on the object.

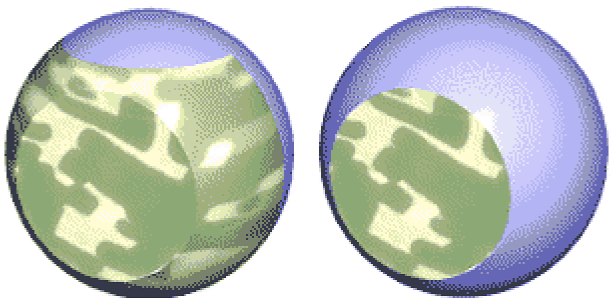


You can demonstrate that best on a spherical object. The picture shows a plane projection onto a sphere with a given angle of incidence of 45°. A point of the bitmap will be projected onto the sphere if the angle between the projection direction (always along the z-axis of the texture axes system) and the object normal (the vector standing perpendicular on the sphere) is smaller or equal than the entered projection angle parameter.

Possible uses:

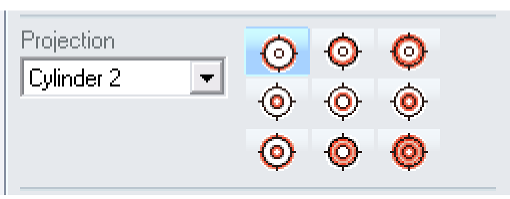


On the left you see a block that is to be textured with different bitmaps on each side of it. In front of the box a camouflage colored bitmap is applied with the bitmap dimensions slightly extending beyond the box dimensions. With a viewing angle of 90° arises the problem that the side walls of the box will be texturized too, which leads to the unsightly stripes depicted in the illustration. A simple decrease of the projection angle parameter to 89° would put things right as you can see in the illustration above on the right.



This illustration shows a sphere with a flattened front and a bitmap projected onto it. We want the bitmap only to be projected on the flattened section of the sphere. That is achieved by reducing the projection angle to 1°.

Cylindrical Projection



With cylindrical projection the picture is wrapped horizontally around the object in the way that you would affix a label to a bottle. There are two cylindrical projection modes available. "Cylinder 1" is used for closed hollow cylinders (bitmaps can be projected inside or outside or on both sides of the cylindrical wall). The "Cylinder 2" projection mode covers all possible projection types to apply bitmaps on the inner or outer walls of open cylinders with double walls. Here you can even decide to apply bitmaps only between the inner and outer cylinder walls.

Example for a transparent cylinder with double walls. With transparent surfaces always the double sided projection mode has to be chosen (red markings on both sides of the projection icons), so that the bitmap is visible on both sides of an object. In our example a simple blue stripe has been wrapped around the cylinder. From left to right: Cylindrical projection on both sides of the outer wall, then cylindrical projection on both sides of the inner wall and at least cylindrical projection on both sides of inner- and outer walls.

When using cylindrical projection you must bear in mind the following regarding picture scaling 206 about the x-texture axis (picture width):

- When x is 1, the picture is scaled so that the picture wraps once about the object and exactly joins behind. If a small line is still visible between the 2 ends of the picture, just increase the scaling in the x direction a bit, until ends join together. Usually a value of 1.01 should be sufficient.
- With x less than 1, the angle of the cylinder projection is reduced, i.e. the picture no longer fits completely around the cylinder. In this way you can easily produce a label.
- With x greater than 1, overlapping of the projection would result. The picture is therefore simply cut off at the join.
- The scaling in the Y direction causes stretching of the height of the picture - as in plane projection.

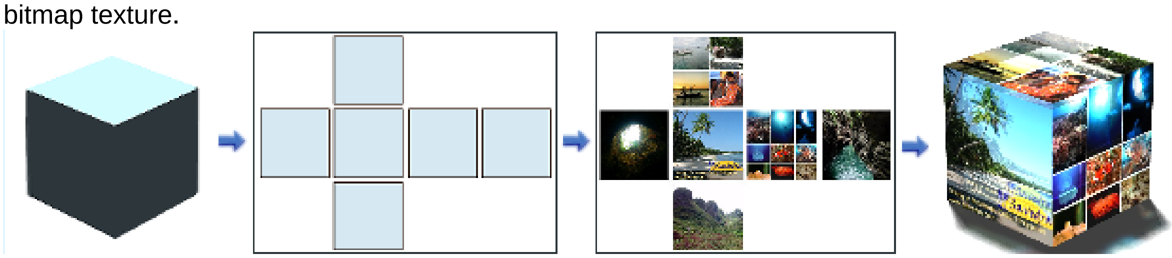
Spherical Projection

With spherical projection the picture is again projected around the object, exactly as in the cylindrical-projection. This time, however, the picture also wraps vertically, so that it is cast over the whole of the object. The X, Y scaling is, therefore, the same as for the cylinder projection:

- X=1, Y=1 : The picture is exactly matched around the object.
- X<1 or Y<1 : label effect.
- X>1 or Y>1 : overlapping.

uv-Mapping

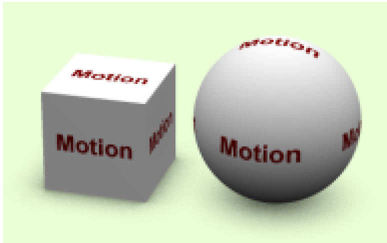
With uv-Mapping bitmap coordinates are directly mapped to the vertices of each facet of an object. Creating a uv-map for an object is done by projecting and rendering the grid structure of an object directly into a bitmap. Then this bitmap serves as a template to paint in the details of the desired bitmap texture.



Example: With uv-Mapping you can texture all 6 sides of a cube simultaneously applying only a single bitmap. You just need to render a uv-map for the cube with cube projection. The rendered bitmap will show the grid structure of the cube neatly unfolded to the 6 side views. Simultaneously, each vertex of the cube is provided with uv-coordinates pointing to the corresponding pixel coordinates in the rendered picture. Now you can easily copy your texture pictures into the uv-bitmap and then use it as a texture map for a new bitmap material.
Here you find a detailed description about generating uv-maps: "Generating uv-Coordinates and Bitmaps for uv-Mappig 218"

Box Projection

In box projection mode a bitmap is mapped from 6 different angles onto an object. In dependence of the direction of the surface normal 8 a top-, bottom-, front-, back-, left- or right projection is applied.

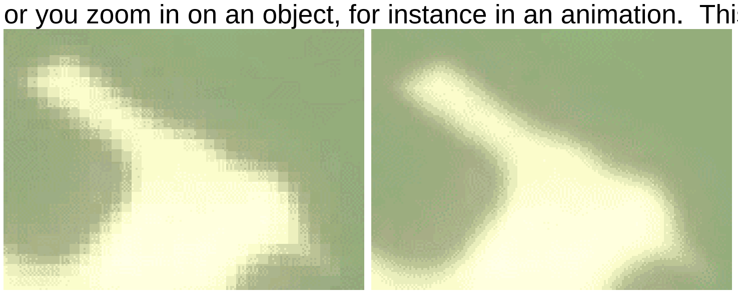


Example of a writing projected in box mode onto a cube and a sphere.

Bitmap Filters

Bilinear Filter

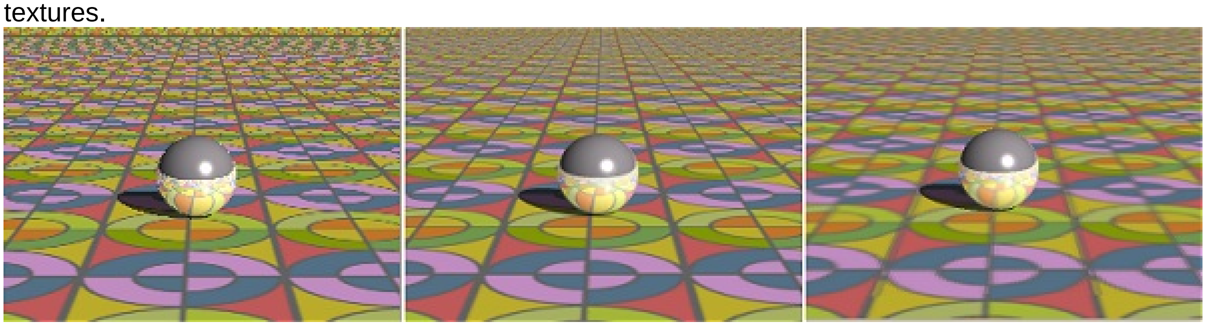
Undesired block effects can be caused when small pictures are magnified through bitmap projection or you zoom in on an object, for instance in an animation. This function smoothes a bitmap texture.



On the left you see a detail of a zoomed object with a bitmap texture. Next to it the same detail is shown, this time with <Bilinear Filter> switched on for the bitmap texture. The difference in quality is clearly visible.

Mip-Mapping

With Mip-Mapping switched on for a bitmap the bitmap texture is smoothed with increasing distance. The Mip-Mapping technique generates additional bitmaps resized and filtered to lower resolutions. The original highly-detailed bitmap is used for surfaces in the foreground. For more-distant points one of the pre-filtered lower resolution bitmaps is applied to prevent noisy texture flickering in the distant parts of the picture (see also: Texture Blur 309 for procedural textures) . It' is in fact, somewhat the reverse of the "Bilinear Filter" function, which reduces pixel steps in the bitmap when zooming into a bitmap texture. Combining both functions you get the best results and smooth bitmap textures.



Example: The plane shown in the illustration above is textured with a tiled bitmap. The picture on the left was rendered with neither antialiasing nor Mip-Mapping switched on. As a result the distant parts of the picture appear very noisy. For the second picture antialiasing 264 was switched on at a maximum quality level of 4 (25 subpixel per pixel). The quality of the picture increases dramatically but the same is true for the rendering time. The illustration on the right shows the same scene with a low level of antialiasing (level 1 = 4 subpixel per pixel) and the Mip-Mapping filter switched on. Although rendering is much faster due to the lower oversampling rate, the quality in the background of the picture has become even better.

Tile

It is often very laborious to draw a bitmap that repeats itself over and over - such as a brick wall or a grass texture. A large picture covering the entire object wastes a great deal of unnecessary storage space. Instead, you can use only a small detail that can be repeated alongside and above each other - this way you get an overall tile-like structure for the required picture. This is obtained simple by switching on the <Tile> button.

You can restrict the number of repetitions by specifying corresponding numbers for the x- and y-direction. Switch off the x- or y-button, respectively, if you want to repeat the structure infinitely.

Mask Color

Suppose you have constructed a yellow-colored sphere on which you want to project a line of black writing. If you set up the bitmap for projection you are forced to use the same background color for the writing on the bitmap as is used for the sphere. It would be better if the background used for the writing-bitmap was transparent and only the writing itself was projected. This can be obtained by applying an additional alpha map 328 or simply by switching on the option "Mask Color" and choosing an appropriate color that comes as near to the background color as possible. Then use the tolerance parameter next to the color box to allow a proportional deviation. Thus deviations caused by antialiasing effects in the picture can also be recognized.

Reflection, Transparency and Glow

The basic material parameters for reflection, transparency or material glow are the same as with procedural materials. See - "Material dialog - Basic Material Attributes 309".

Part XIV

14 Light, Fire and Photons

The right illumination is the most important thing in computer graphics. Accordingly, MR-3D Designer offers various options to light up your scene, beginning with standard light types like simple point lights, spotlights and sunlight, up to volumetric fire and area light objects of any form you like.
Of particular importance is the calculation of the general area brightness - the diffuse light incidence caused by manifold light reflections in the scene or by atmospheric dispersion. Here you can choose between different rendering algorithms, each of which can significantly increase the realism of your renderings.

- **The Light Dialog** 341
  Overview, preview options, light color and management of light objects

The individual light types:

- **Ambient Light - General Area Brightness** 343
  It's not the direct light incidence but the simulation of the diffuse area brightness which produces the impression of photo-realistic pictures
  Ambient 343 - A simple constant term for the area brightness
  Fill Lights 343 - Use additional lights of less intensity to light up and accentuate shaded areas
  Ambient Occlusion 343 - A very effective simulation of scattered light similar to the light incidence on a cloudy day. In combination with high definition range images (HDRI) - which provide not only the background texture but also the light information for the diffuse light incidence - you can produce ultrarealistic scenes with 3d-models fully integrated in a photographic environment map.
  Photon Mapping 277 - A global illumination model that takes into account all interactions of light with the different surface materials in the scene
- **Parallel Light - Sun Light** 351
  Symmetrical incidence of sunlight
- **Lamp - Point Light Source** 353
  The point light source, a standard in all 3D-programs
- **Spotlight** 354
  Cones of light
- **Volumetric Fire** 358
  Walking through fire, why not?
- **Area Light** 356
  How to convert an ordinary object into a real light source

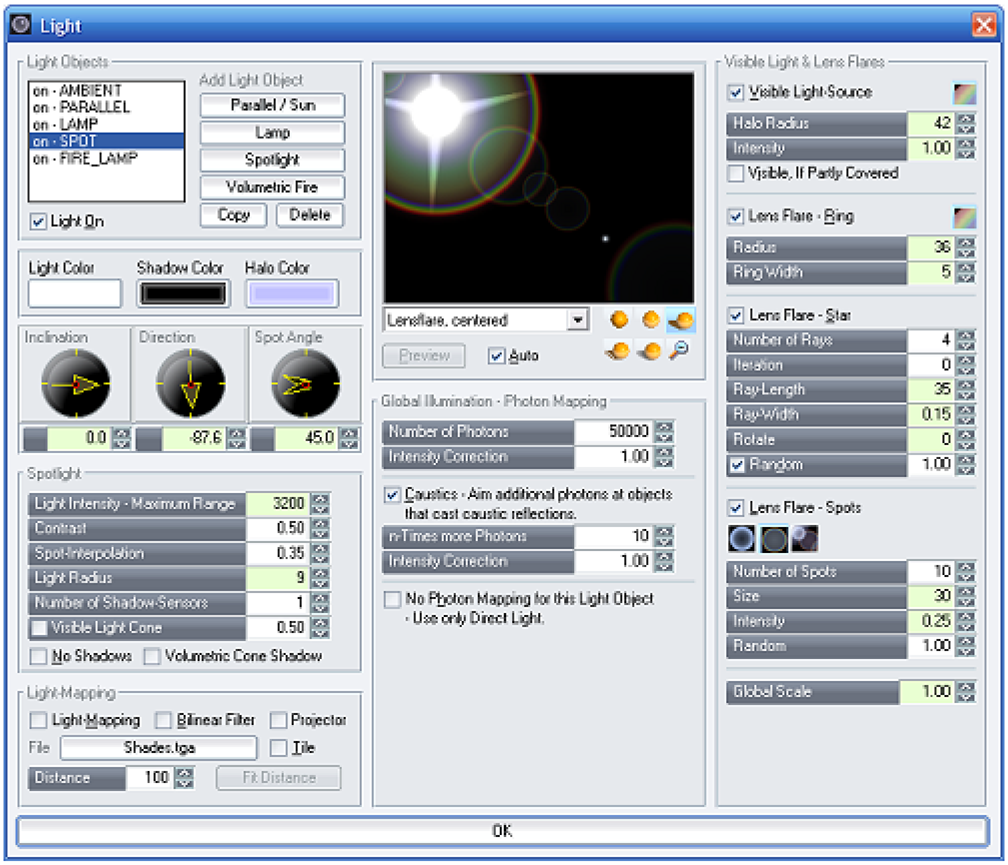
and...

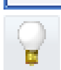
- **Photon Mapping Emission Parameters** 370

With photon tracing each light source can emit millions of light particles to simulate the light distribution in a scene

- **Light-Mapping** 363
  How to use bitmaps to project colored light patterns
- **Lens Flares** 365
  Visible light and light reflections in the camera lens

14.1 The Light Dialog



 - Menu "Options - Light" - Short Cut:  + "L".

Light sources are managed entirely as normal objects in MR-3D Designer. Like all other objects, they can be worked on (e.g. moved and rotated) or copied and deleted. The only exception is the ambient light-source 343 - which is responsible for the general brightness of the area. As there can be only one area brightness only one ambient light object is also possible, which can neither be copied nor deleted. You can insert as many light objects as you wish, provided that the maximum number of objects maintained by MR-3D Designer is not exceeded. Here, in the light dialog, new light objects can be created at any time on the press of a button. There are, however, two light objects generated at program start time, so that you already possess basic lighting for the preparation and editing of new objects and scenes. The ambient light source (for the general area illumination) and a parallel light source are generated at program start. If you have arranged a scene you can modify and develop this basic illumination with new light objects.

The Preview Window

The window in the middle of the dialog provides a quick preview of the scene when changing light settings. Depending on the selected preview mode it can also show an isolated view of the selected light object and its light effects. Directly under the preview window you can choose one of the following preview modes:

- Lensflare, centered - if lens flare effects are assigned to the light object then a centered view of the selected light object is rendered. Note that the lens flare size in the preview is always adjusted to the dimension of the preview window and thus the lens flares in the real scene can vary considerably, depending on the distance of the light source from the camera and the incidence angle of the light.
- Camera, light objects - only the visible light objects are rendered in camera view.
- Camera, background and planes - a scene preview with all lights, background and plane objects in the scene. This is an ideal setting for quick previews when changing light parameters for complex scenes like landscapes, thus preventing time consuming redraws of the complete scene when you only want to change the sun position, for instance.
- Camera, complete scene - the whole scene is rendered in the preview window each time you change a light parameter.

See also: general preview options 31 like quality, resolution or automatic update.

List of Light Objects

At the top left of the dialog is a list box, in which all the existing light-objects are displayed. A particular light object can be chosen and the current settings for this light object are displayed and can be modified.

Creating a New Light Object

There are four buttons in the dialog field "Add Light Object" to generate new light objects. The new light is then automatically provided with a name and displayed in the list box. You have the choice of generating a parallel light 351, a lamp 353, a spotlamp 354 or a volumetric fire 358 object. In addition, you can convert any ordinary object into a real area light 356 source.

Copy or Delete Light Objects

Beneath the four buttons for creating new light objects are two additional buttons with which you can copy or delete a selected light object. (Of course, it is also possible to copy or delete light objects like any other object via the corresponding functions of the popup selection 133 of the main Object Selection 133 window.)

Switch On or Off Light Objects

You can switch on or off light objects via the corresponding button beneath the list box.
As with all other objects, you can switch light objects on or off in the select objects 133 dialog, too. There you can also delete or copy them. Here in the light dialog, you can create new light objects or edit existing ones.

Light Color

Select the <Light Color> button to call up the color selection dialog. The color selected here is the color of light emitted by the selected light source. For example, a light-gray to white color corresponds to normal daylight. For a warm light in an area choose a rather yellow tone.

Shadow Color

In computer generated pictures, shadows often appear too dark. In real life these shaded areas appear brighter because of the incidence of scattered ambient light 343. MR-3D Designer offers several renderering algorithms of varying complexity to simulate this environmental light. One simple method is to adjust a shadow color, which defines a minimum light incidence for those areas lying in the shadow of the light source.

The picture above illustrates the effect. By applying a greyish shadow color we can lighten the dark shadows while the surface directly exposed to the light is not influenced by the additional shadow light.

Halo Color

In addition to the buttons for the light- and shadow colors is another button - <Halo Color>. Here you can edit a light color that is not used to illuminate the scene, but to generate light halos for visible light sources and lens flare 365 effects.

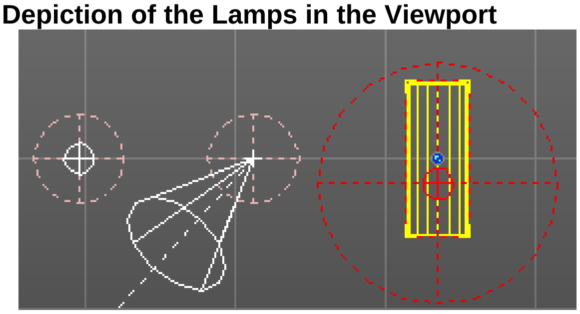
Light Parameters

Beneath the light colors are presented all parameters related to the selected light type. They will be described in detail in the corresponding chapters of the different light types.

Photon Emission Parameters

If you render in global illumination mode using photon mapping 277 to simulate the distribution of light particles in a scene, you can specify the parameters for the emission of photons 370 in the dialog area beneath the preview window.

Depiction of the Lamps in the Viewport



On leaving the dialog all moveable light sources, as lamps 353, spotlights 354 and volumetric fire 358 objects, are drawn in the viewport windows. A normal lamp appears as pictured above on the left. The dotted line circle about the lamp shows the halo radius entered for a visible light source, the solid inner circle indicates the light radius.
So that you can identify and process the spotlight cone within a scene, a spotlight source is drawn as a vector object with ray-cone and a direction line. This allows a spotlamp to be easily arranged so that a target object is within the light cone.
A volumetric fire object is drawn as a cylinder with a point light source inserted in it (picture above on the right).
Parallel light sources do not require positioning and consequently are not shown.

14.2 Ambient Light - General Area Brightness

One of the most difficult tasks in computer graphics is to simulate area brightness - the diffuse light incidence caused by manifold reflections of light in the scene or by atmospheric light scattering.

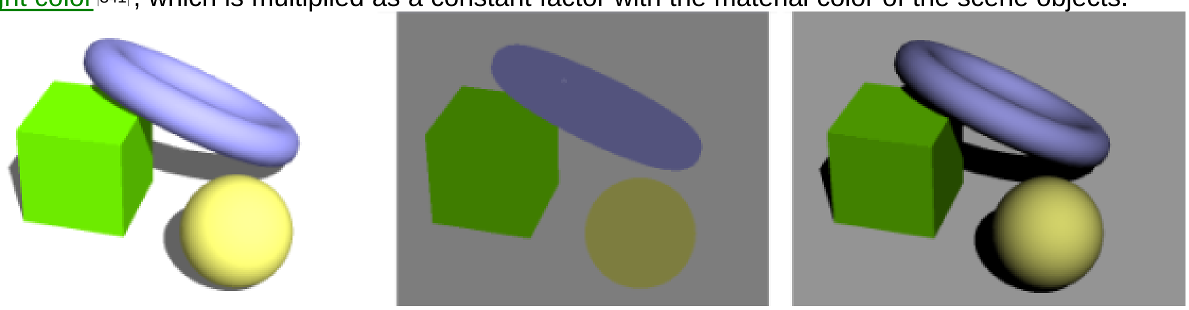
Usually, a simple constant term is applied to provide for the additional ambient light intensity. This ambient term can be calculated in no time and with it all objects in the scene are lightened slightly - but the quality of pictures rendered with a constant term for the ambient light often becomes very flat and poor. In architectural scenes especially, the illumination in a room is dominated by indirect light reflected many times from the diffuse surfaces in a building. This aspect of indirect illumination is ignored when applying only a constant ambient term for the area brightness. To achieve photo-realistic results, more complex rendering algorithms have to be employed - global illumination models like Ambient Occlusion Rendering or Photon Mapping.

The different rendering approaches for the simulation of scattered light:

- Ambient Light Object 343 - A simple constant term for the area brightness
- Ambient Occlusion 343 - A very effective simulation of scattered light, similar to the light incidence on a cloudy day. In combination with high definition range images (HDRI) - which provide not only the background texture but also the light information for the diffuse light incidence - you can produce ultrarealistic scenes with 3d-models fully integrated in a photographic environment map.
- Fill Lights 343 - Use additional lights of less intensity to light up and accentuate shaded areas
- Photon Mapping 277 - The physically-based simulation of all light distribution in a virtual 3D-model is called global illumination. A global illumination algorithm should take into account all interactions of light with the different surface materials in a scene. With the photon mapping algorithm, MR-3D Designer provides a global illumination model that combines the pros of raytracing - reflection and refraction - with the ability to also render the indirect illumination caused by diffuse interactions in the scene. Rendering a picture with photon mapping is a two pass procedure. In a preliminary run little packages of energy (photons) are emitted from the light sources into the scene. Similar to ordinary raytracing, the paths of these photons are traced through the scene and the distribution of the light particles saved in a 3D-data structure called the photon map. After the photon map has been calculated the picture is rendered in an ordinary raytracing run and the photon map is used to determine the area brightness when calculating the illumination for a surface point. The complex subject photon mapping is discussed in several chapters - see "Photon Mapping - Introduction and examples 277" for more details.

Ambient Light Object - A constant term for the area brightness

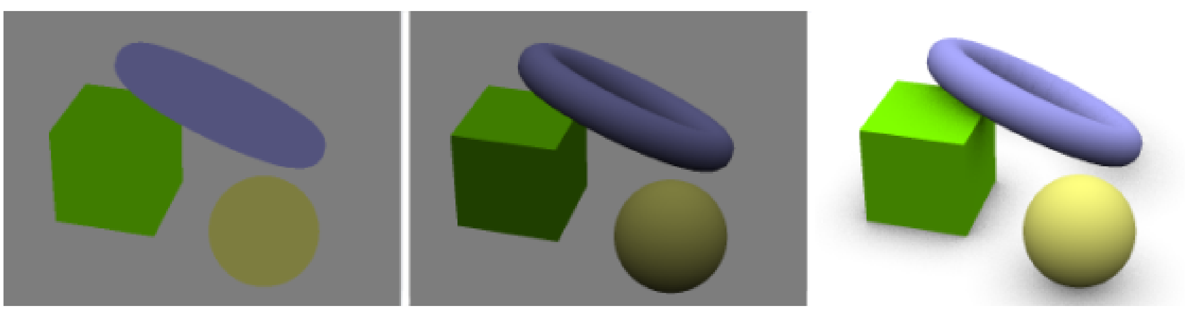
The light object "Ambient" is listed together with all other lights in the object list of the light dialog where you can select it to edit all related parameters. As there can be only one area brightness only one ambient light object is possible, which can neither be copied nor deleted. If Ambient Occlusion 343 is deactivated then the ambient light is described simply by the light color 341, which is multiplied as a constant factor with the material color of the scene objects.



The composition of light: The left picture is a summation of ambient light (picture in the middle) and direct light incidence (picture on the right). The picture in the middle reveals the great drawback of using only a constant term for the ambient light contribution - it has no incidence angle and therefore no shading gradients can be rendered, not to mention the calculation of shadows. With increasing ambient light intensity the picture would become also increasingly flat and lose much of its plasticity. We can change this behaviour when we activate the <Ambient Occlusion> option.

Ambient Occlusion(AO)

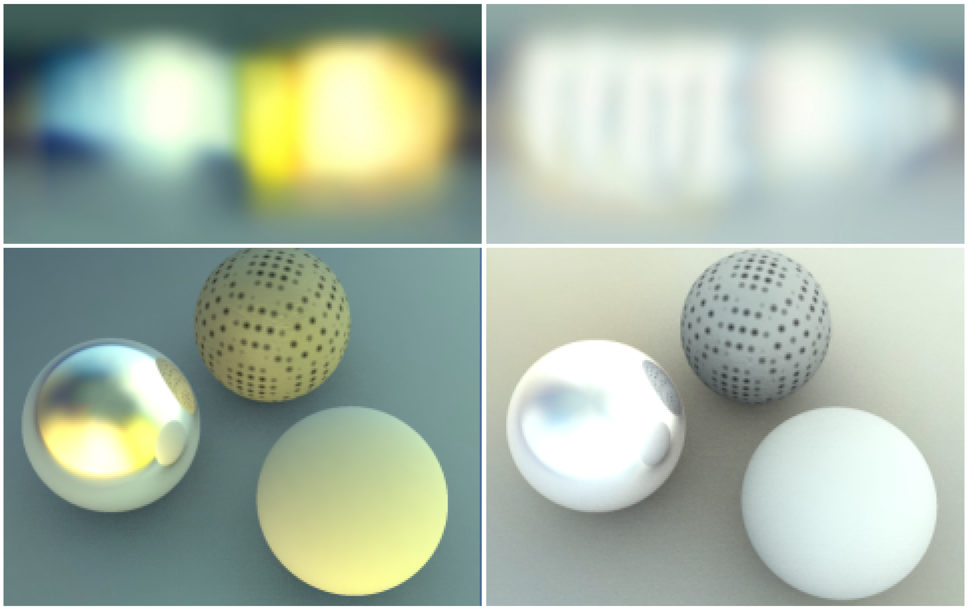
With the Ambient Occlusion shader it is possible to simulate the mutual shadow casting of adjacent objects and corners caused by diffuse ambient light. The light intensity now calculated depends on the angle of the surface normal with the perpendicular vector pointing to the zenith of the 3D-world. That corresponds to the light intensity a surface would receive if exposed to a surrounding hemisphere of uniform light intensity (Sky Dome effect). This will lighten surfaces exposed directly to the sky and darken surfaces directed towards the floor. Even without additional shadow calculations this gives a much smoother impression of ambient light then the constant ambient term, similar to the impression of diffuse light incidence on a cloudy day.
If additionally you switch on shadow calculations for the actual occlusion part of the Ambient Occlusion shader, a bunch of shadow rays is cast towards the imaginary half sphere illuminating each point of the surface. These shadow rays are limited to a maximum search radius, so they cover only the immediate surrounding area (on a cloudy day you wouldn't expect a far away tree to cast a shadow on you, only a smooth light shadow would appear under the crown of this tree). If a shadow ray hits an occluding object, the ambient light contribution is simply reduced by a fraction of its original intensity. In the result, you achieve a wonderful smooth diffuse illumination together with very soft shadows. As with all other soft shadow effects, the quality of the final picture largely depends on the number of shadow sensors applied for the shadow calculations, which means that not only the output quality is increased considerably with each additional shadow ray but also the rendering time. On the other hand the shadow calculations for the Ambient Occlusion are speeded up because of the limited search radius in contrast to the soft shadow calculation of other light sources. This limiting distance is also important when applying Ambient Occlusion for architectural interior scenes. If you define the maximum search radius to be a fraction of the size of a room then the Ambient Occlusion shader will not only generate soft shadows around all occluded areas but also emphasise all corners of adjacent walls.
In MR-3D Designer the Ambient Occlusion renderer is simply an extension of the "Ambient" light object. To edit all corresponding parameters simply select the "Ambient" light in the object list of the light dialog.



For comparison the picture above on the left again shows the light contribution of the constant area brightness in contrast to the light contribution with activated Ambient Occlusion (shadows switched off - picture in the middle). The big advantage of Ambient Occlusion is apparent - because it takes into account the light incidence angle it can render color gradients. The calculation of this angle-dependent basic area illumination is very fast (in contrast to the time consuming shadow calculations), therefore it will also be applied for the depiction in the viewport windows. For the picture above on the right the scene was illuminated only with ambient light of full intensity (light color white) and activated Ambient Occlusion including shadow casting. The picture demonstrates that with Ambient Occlusion it is possible to render complete scenes without any additional lighting but the area brightness. When activating Ambient Occlusion with shadow calculations, it is advisable to first adjust the parameters for the AO-illumination and then to add further light sources only for the accentuation of particular areas in the scene. This way it is easier to adjust the light intensities of additional lights so that they don't obliterate the soft shadow effect. It would be a tremendous waste of rendering time to calculate soft shadows which can't be seen under the bright illumination of the scene. Take into account also that with Ambient Occlusion only a diffuse area brightness is simulated - no highlights caused by light reflections on shiny surfaces are created by this type of illumination. For this purpose you always have to combine the ambient light with other light sources.

Example of an indoor scene (model by Pascal Heußner) illuminated by two lamps, the picture on the left was rendered with activated Ambient Occlusion in contrast to the picture on the right which was rendered with a constant area brightness. In the left picture clearly discernible shadows are rendered in all corners of walls and stairs and around door and picture frames - the picture appears to have more plasticity then the right one.

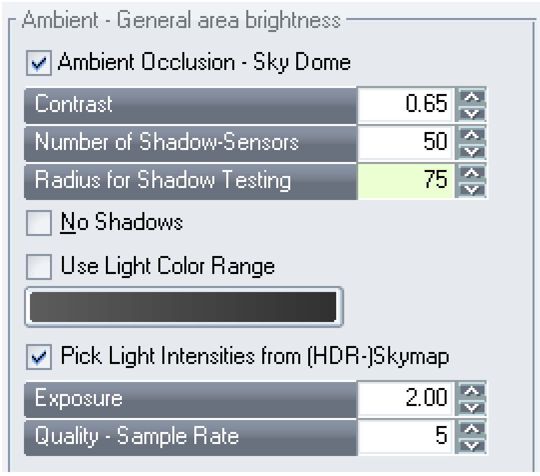
This picture of an anatomy model (model by cacheforce: www.anatomium.com) shows the difference even more clearly, on the left rendered with activated Ambient Occlusion and on the right with a constant area brightness.



This picture demonstrates Ambient Occlusion rendering in combination with skymaps 394 (environment maps) and image based lighting 394 (IBL). A skymap is a panoramic bitmap projected onto an imaginery sphere or cube surrounding the whole scene. It usually provides the information for the background texturing and environmental reflections, but skymaps can also be used as a kind of light map for the Ambient Occlusion renderer. To evaluate the ambient light incidence on a given surface point, now the hemispherical projection of the environment map above this surface point will be scanned and the pixel intensities of the skymap are interpreted as incoming light intensity. Especially in combination with high definition range images (HDRI 394), that provide a much higher dynamic range of color intensities than ordinary RGB pictures, you can achieve impressing results.

An example for an perfect integration of a virtual 3d-scene (car model: mr-clipart.com) into a photographic environment (HDRI by Paul Debevec, www.debevec.org). The only light source used to render the final picture on the right was the light information provided by the HDR-image. The ground object beneath the car was provided with some special object properties 290, so that it became transparent to the background but still received the shadows cast by the car model. See here for more information about Skymaps, IBL and HDRI 394.

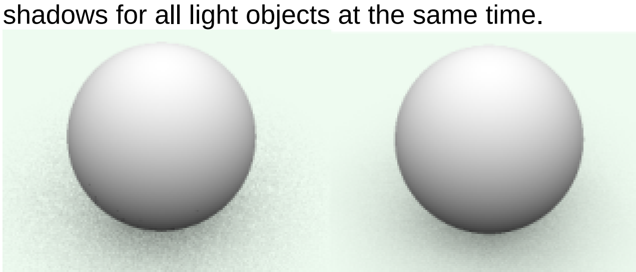
The additional parameters for the Ambient Occlusion rendering:



Ambient Occlusion - Sky Dome - This button activates the Ambient Occlusion rendering. If it is switched off only the simple constant area brightness will be included in the illumination model.

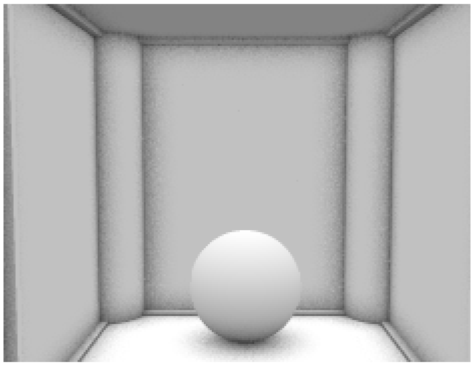
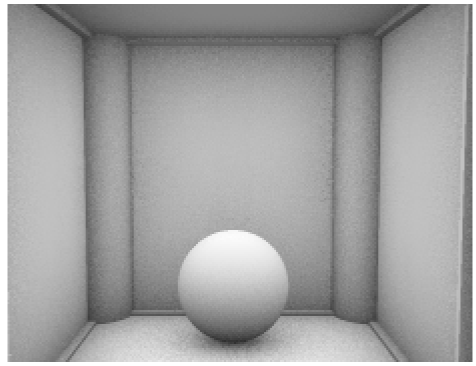
Contrast - The light intensity is calculated in dependence of the angle of the surface normal 8 with the vector pointing from the surface towards the light source. With the contrast parameter you can elevate or flatten the curve of the angle-dependent intensity progression and thus greatly influence the light distribution of the light source. See "Lamp - Contrast 353" for an example.

Number of Shadow-Sensors - Determines the number of shadow-sensors used to scan the immediate surroundings of a surface point to test for occluding objects. If the number of shadow-sensors is low, a very noisy penumbral transition from the deepest shadow to the shadeless area is produced. The greater the number of shadow-sensors and the less the radius for the shadow testing, then the better will be the soft shadow effect. However, the number of additionally calculated shadow-sensors should be kept as low as possible due to the rendering time required. To render soft shadows you also have to activate the global switch <Multiple Shadow-Sensors 264> in the render options dialog. With this button you can activate or deactivate the rendering of soft shadows for all light objects at the same time.



The picture above on the left was rendered with only 9 shadow-sensors which results in a noisy shadow pattern beneath the sphere. When increasing the number of shadow-sensors to 35 we get an almost perfect shadow transition for the sphere (picture on the right).

Radius for Shadow Testing - This parameter defines the maximum distance for the shadow-rays to search for occluding objects. With a greater search distance, the shadowy areas beneath objects and in corners will increase - this means you need a higher number of shadow sensors to achieve good results.

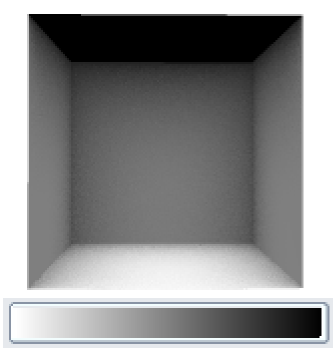
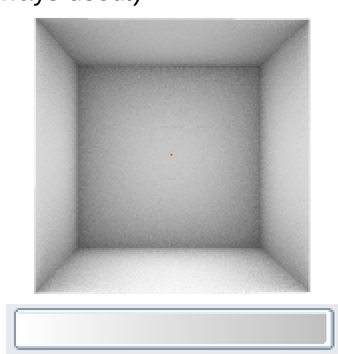



Example: A simple room illuminated only by area brightness and activated Ambient Occlusion. For the picture on the left a very short shadow radius was defined. This way, all corners and edges appear sharply pronounced. You can smooth this effect somewhat by adjusting a shadow color other than black for the ambient light.
For the picture on the right an increased search radius has been chosen. Now the shadow transitions cover larger areas, creeping from the corners to the middle of the walls and occupying a broader range beneath the sphere. To reduce noise in this picture the number of shadow sensors had to be doubled.

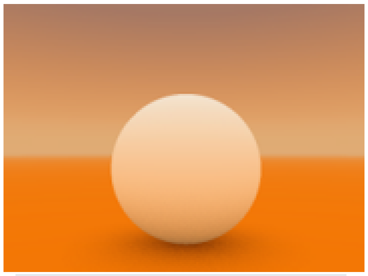
No Shadows - You can switch off the time-consuming shadow calculations for the Ambient Occlusion rendering - for instance, to render fast previews or if you just want to apply the ambient light to brighten up the scene a little bit while the main illumination and shadow casting is created by other light sources added to the scene.

Shadow Color 341 - The shadow color defines a minimum light intensity which is used to illuminate even those areas lying in the shadow of a light source.

Use Color Range - The Ambient Occlusion shader calculates the light intensity that a surface receives dependent on the angle of the surface normal with regard to the perpendicular vector pointing to the zenith of the 3D-world. This will lighten up surfaces exposed directly to the sky and darken surfaces directed to the floor. Internally, this shading is realised by a color range starting with the maximum light intensity given by the light color 341 followed by a transition to pure black at the end of the color range. But for indoor scenes this color range is not suitable, because we want ceilings, which point downwards, to participate in the area brightness. To achieve this we can activate and define a separate color range with a brighter end tone. For instance, you can set the maximum intensity (starting color of the color range) to a light gray and the end color to a somewhat darker shade of gray. To edit the color range 30 simply click on the color range button.
(Note: For the realtime depiction of the viewport windows, the internal color range with a transition from the light color to black is always used.)

The picture above on the left shows a room illuminated with AO and deactivated color range. The color range beneath the picture is the color range used internally for the intensity calculations - it ranges from the white light color to black. For the picture on the right the <Use Color Range> button was switched on. The end color of the color range was adjusted to a light gray and with this the side walls and ceiling are rendered much brighter.



Even colored area light is possible. Here an orange end color was used to adapt the area brightness to the colors of the sky and the ground below. Additionally a dark shade of orange was applied for the shadow color.

Pick Light Intensities from (HDR-)Skymap - A skymap is a panoramic bitmap projected onto an imaginery sphere or cube surrounding the whole scene. It usually provides the information for the background texturing and environment reflections, but skymaps can also be used as a kind of light map for the Ambient Occlusion renderer. To evaluate the ambient light incidence on a given surface point, now the hemispherical projection of the environment map above this surface point will be scanned and the pixel intensities of the skymap are interpreted as incoming light intensity. Some examples for the interaction of skymaps, image based lighting and high definition range pictures have been shown further up in this chapter. More information about skymaps, IBL and HDRIs can be found in the chapter describing Background Bitmaps, Skymaps or Environment Maps 394.
There are two parameters with which you can adjust the light incidence calculated from the skymap:
Exposure - This parameter is a simple factor with which you can increase or decrease the intensity of the light probes sampled from the picture. This is especially useful, if you use ordinary RGB-pictures for the environment mapping instead of high dynamic range HDRIs. When using low dynamic RGB-pictures the scene usually becomes to dark, so increasing the exposure value can contribute much to the quality of the light incidence.
Quality - Sample Rate - This parameter defines the sample rate for scanning the environment texture for pixel light intensities. To accelerate this process, MR-3D Designer internally generates a strongly blurred copy of the original skymap before rendering. This way good results can be achieved even with low sample rates. The sample rate is measured in qualitiy steps from 1 to 10.

Fill Lights

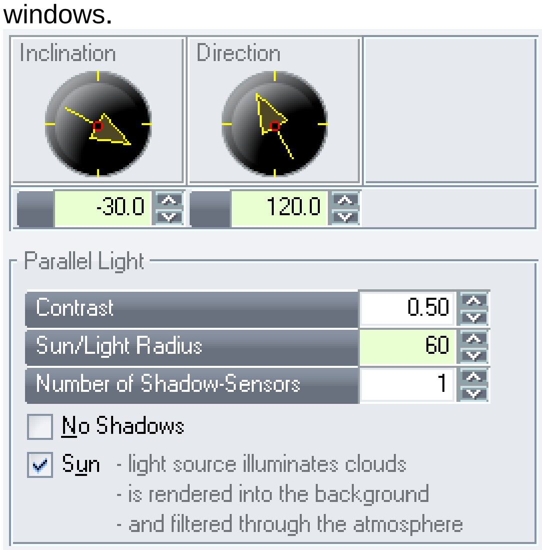
Often additional lights of less intensity are used to simulate interior reflections and to light up and accentuate shaded areas. For instance, you could direct a parallel light of low intensity to the ceiling of a room to simulate diffuse reflections from the sidewalls or the floor (shadow casting switched off - it is only to simulate diffuse area brightness). Or you place additional lamps so that they not only lighten but also accentuate the shaded areas of an object. Such tricks are very often used in computer industry and this special trick is called "fill lights". Even in real photography fill lights are used, e.g. during photo sessions when white boards are used to bounce spill lights back to the subject. Fill lights are also often used to produce special reflections on objects or to save time on complex shadow calculations.

This picture is an exceptional demonstration for the application of fill lights.German artist Jürgen Kaupp modeled this scene in CyberMotion and rendered it by means of ordinary light sources - no global illumination algorithms like Ambient Occlusion or Photon Mapping were applied.

14.3 Parallel Light - Sun Light

A new sun light object is created by operating the "Add Light Object" <Parallel/Sun> button in the light dialog 341.

This light source is similar to sunlight, i.e. parallel incident light of uniform intensity. As the light comes from outside the 3D area only the angle at which the light occurs is input. Positioning this light source later is not necessary, so a parallel light object is not represented in the viewport windows.



The parameters for a parallel light source

Incidence angle - You can directly input the inclination and direction angles to define the incidence angle of the light or just click with the mouse into the instrument boxes and drag the pointer to the desired position.

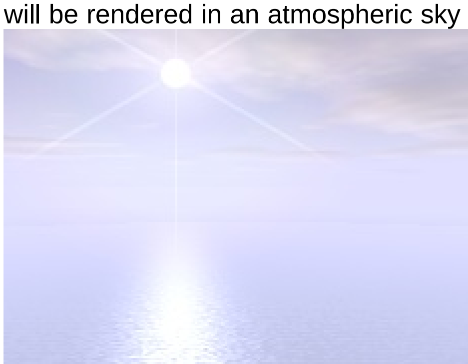
Contrast - The light intensity is calculated in dependence of the angle of the surface normal 8 with the light incidence vector. By using the contrast parameter you can raise or flatten the curve of the angle-dependent intensity progression and thus greatly influence the light distribution of the light source. See "Lamp - Contrast 353" for an example.

Sun/Light Radius - This radius defines the size of the sun disc rendered in the background of the scene (if <Sun>-interpretation is switched on) or the size of a visible lens flare produced in the camera lens, respectively (see Visible Light and Lens Flares 365).
The light radius also provides the light source with an area, which can be scanned with a number of shadow-sensors to produce the soft shadow effect.

Number of Shadow-Sensors - Determines the number of shadow-sensors used to soften shadows. Standard light sources like the lamp, sun and spotlight are defined with a specific radius and, instead of originating from a single point-light-source, the light comes from a spherical area light-source. The given number of shadow-sensors are then used to scan the light-sphere and to estimate how much of the light-area is hidden by other objects. From these results a soft shadow is interpolated.
If the radius of a light source is very high and the number of shadow-sensors low, for example, a very noisy penumbral transition from the deepest shadow to the shadeless area is produced. The greater the number of shadow-sensors and the less the radius then the better will be the soft shadow effect. However, the number of additionally calculated shadow-sensors should be kept as low as possible due to the rendering time required.
To render soft shadows with more than one shadow-sensor you also have to switch on the <Multiple Shadow-Sensors 264> option in the render options dialog.

No Shadows - The time required for the shadow calculation increases with each light source, especially if multiple shadow sensors are involved. On the other hand, you often require several light sources to correctly illuminate all the objects in your scene. Not all lamps cast shadows relevant to picture rendering, e.g., fill lights 343 of a low intensity and range used to brighten up and accentuate areas lying in the shadow of other light sources. In this event, it makes sense to switch off the shadow casting for all light sources that don't contribute to the main illumination of the scene.

Sun - Switch this option on if you want a parallel light source to be interpreted as a real sun that will be rendered in an atmospheric sky background.



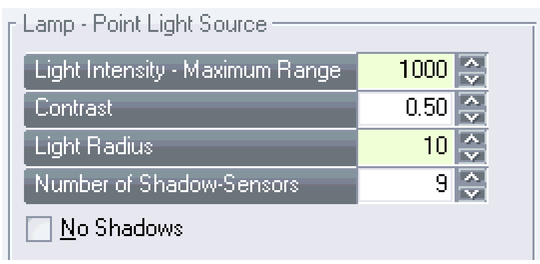
If you select the <Sun> option, then automatically the <Visible Light Source 365> effect will be switched on, too. But the visible sun object is handled differently than other visible light objects. In combination with a clouded sky background 378 the sun illuminates the clouds. The sun is always rendered as a visible disc in the background behind the clouds and its intensity is filtered by the clouds and the atmospheric haze. Furthermore the color range used to produce the

sky background 379 can be distorted in the vicinity of the sun thus creating more realistic lighting in the sky around the sun.
The normal lens flare effects 365 are rendered in addition to the visible sun disc. That is why a lower intensity for the lens flare effects should be input when combining a sun light with lens flare effects. A good example of use is, for example, a sun setting behind a mountain with the direct light causing a lens flare halo in the camera lens shedding a warm veil of light about the mountain peaks.

Other relevant settings for the parallel light object in the light dialog:
- Light-Mapping 363 - How to use bitmaps to project colored light patterns
- Lens Flares 365 - Visible light and light reflections in the camera lens
- Photon Mapping 370 - The photon emission parameters

14.4 Lamp - Point Light Source

A new lamp is created by operating the "Add Light Object" <Lamp> button in the light dialog 341.



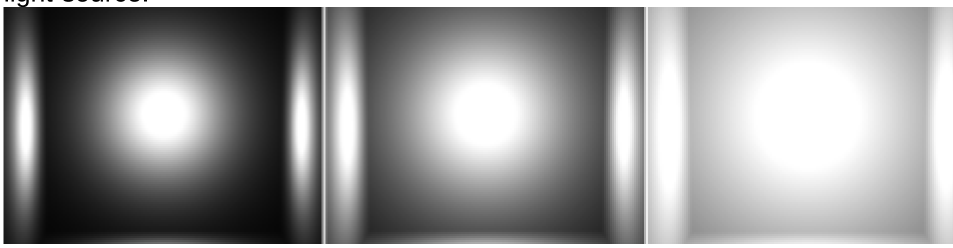
The parameters for a point light source

This light source is similar to a light bulb - the light radiates in all directions from a single point.

Light Intensity - Maximum Range
In the real world the intensity of a point light source reduces in proportional to the square of the distance, i.e. by doubling the space between the object and the light source the light intensity is reduced to a quarter. In computer graphics, however, this does not lead to satisfactory results - since most light sources in real life are not ideal point light sources. Therefore MR-3D Designer uses a special filter to reduce the light intensity with distance. To enter an appropriate intensity for the lamp you simply have to specify a radius at which distance the light intensity will almost reduce to zero. For example, to estimate the required light intensity for a room you would determine the room dimensions via the Box-Dimension function in the Scale Object menu and simply use this value as a basis for the light intensity. Then use the "Camera-complete scene" preview mode in the light dialog to adjust the light intensity until the room is lit satisfactory.

Light Intensity and Photon Mapping - If you plan to render your scene with the photon mapping 277 algorithm you ought to follow this course: Adjust the light intensities with normal raytracing 264 for the preview renderings. Then render a first test picture in photon mapping mode. To balance the intensity variations in the picture caused by the two totally different illumination methods, adjust the light intensity for the photon emission via the Intensity Correction 370 value in the light dialog's "Global Illumination - Photon Mapping" parameter box instead of changing the general "Light Intensity - Maximum Range" parameter. Thus you can switch back to raytracing mode for faster previews when extending and editing your scene and light settings and in the end you can turn again to photon tracing without having to adjust again the light intensities for the photon mapping process.
See also: Light Dialog - Photon Emission Parameters 370

Contrast - The light intensity is calculated in dependence of the angle of the surface normal 8 with the light incidence vector. With help of the contrast parameter you can raise or flatten the curve of the angle-dependent intensity progression and thus greatly influence the light distribution of the light source.



The picture above shows the same scene of a room illuminated by a lamp centered in the middle of the room. In each picture the lamp shines with the same intensity but with an increasing value for the contrast parameter so that the intensity contribution of light incidence at lower angles is increased too.

Light Radius - If you enter a radius for a point light source it is possible to render this light object as a visible spherical light source (see also: Visible Light and Lens Flares 365).
But the light radius is also required to determine the maximum deflection for shadow-sensors 351 when scanning a light sphere for the generation of a soft-shadow effect.

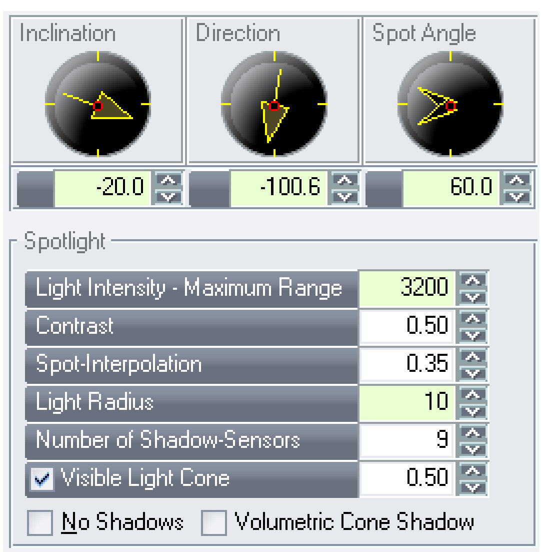
Number of Shadow-Sensors - Determines the number of shadow-sensors used to generate a soft-shadow effect. *See also:* Parallel Light - Number of Shadow-Sensors 351

No Shadows - The light source don't cast shadows. *See also:* Parallel Light - No Shadows 351.

Other relevant settings for lamps in the light dialog:
- Lens Flares 365 - Visible light and light reflections in the camera lens
- Photon Mapping 370 - The photon emission parameters

14.5 Spotlight

A new spotlight is created by operating the "Add Light Object" <Spotlight> button in the light dialog 341.



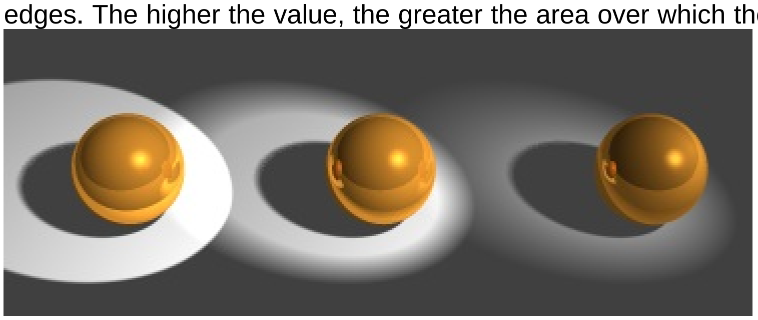
The parameters for a spotlight

A spotlight has the same type of light dispersal as with a normal light bulb, except that in this case an angle of spread can be stated in which the light is visible. This creates a cone of light originating near a spotlamp. On the left of the dialog box are now shown three angle instruments. The first two again give the inclination and direction of the spot cone, as with parallel lights (you can also change the spot direction in the Rotate Object 211 work menu where the spot object is treated as an entirely normal object that can be turned at will so that the direction of the spot cone is changed with the rotation). The third instrument contains two arrows emerging from the spot, which show the divergence angle of the light - the cone angle. Between 1 and 180 degrees can be input for the emerging cone.

Light Intensity - Maximum Range - A reduction in light intensity with distance is calculated. See also: Lamp - Light Intensity-Maximum Range 353

Contrast - See "Lamp - Contrast 353"

Spot-Interpolation - The harshness at the edges of a spotlamp beam can be reduced using the Spot-Interpolation parameter so avoiding an unrealistic extremely harsh transition at the beam edges. The higher the value, the greater the area over which the intensity is diffused.



Examples in the illustration, from left to right:

- 0- Normal light-cone without diffusion of the intensity.
- 0.3- Over the last 30% the edge area shows a reduction in intensity for slow blending and soft spot edges.

1- The light intensity falls off from the center of the beam to the cone edge.

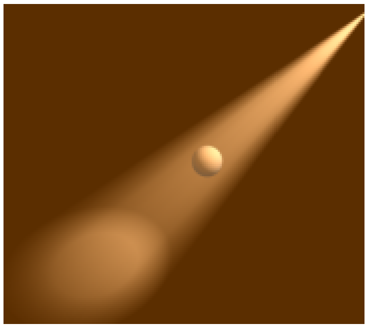
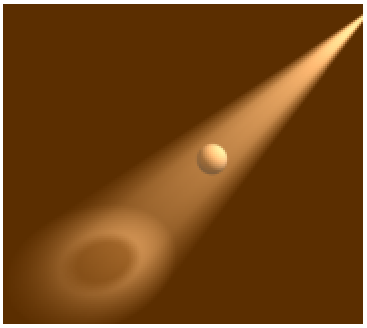
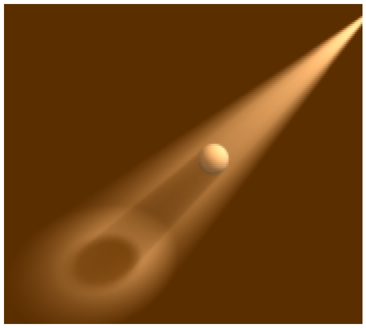
Light Radius - If you enter a radius for a spotlight source it is possible to render this light object as a visible spherical light source (see also: Visible Light and Lens Flares 365).
But the light radius is also required to determine the maximum deflection for shadow-sensors 351 when scanning a light sphere for the generation of a soft-shadow effect.

Number of Shadow-Sensors - Determines the number of shadow-sensors used to generate a soft-shadow effect. *See also:* Parallel Light - Number of Shadow-Sensors 351

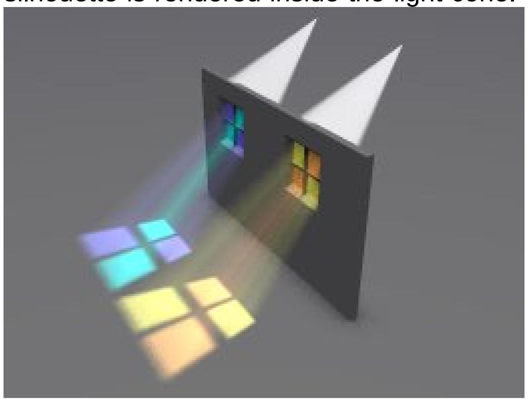
No Shadows - The light source don't cast shadows. *See also:* Parallel Light - No Shadows 351.

Visible Light Cone - Switch on this option to render a visible light-cone. In reality a light-cone becomes visible if small particles of dust or smoke reflect the light of the spot lamp. The value of the <Visible Light Cone> parameter controls the reflectivity of these imaginary particles or the brightness of the visible light cone, respectively.
Volumetric Cone Shadow - If you also activate the <Volumetric Cone Shadow> option then shadow silhouettes of objects penetrating the light cone are included in the calculation as well as the filtering of light through transparent objects. To create these effects, a volumetric approach is applied to render samples within the light cone which accumulate to an interpolated intensity value. The rendering of volumetric shadows is very calculation-intensive, therefore you can decide for each spot lamp individually if it is to be rendered with a visible light cone or volumetric shadows, respectively. However, the quality value, which defines the sample rate for the scan of the light

cone, is a global parameter that is adjusted in the Render Options 273 dialog. There you will find two additional buttons to globally switch on or off the rendering of light cones or volumetric cone shadows.

The pictures above show the different types of spot cone rendering. In the picture on the left, the light cone was rendered without any shadows (option <No Shadows> activated). The picture in the middle shows the same scene with activated shadow calculations. Now the sphere hovering in the center of the light cone casts a shadow on the ground. Then, for the picture on the right, the rendering of volumetric cone shadows has been activated. Here a clearly discernible shadow silhouette is rendered inside the light cone.



This illustration shows a simple house wall with 2 windows, behind which is a spotlight source 354 with visible light-cones. A colored bitmap with transparency has been projected onto the transparent panes, so that the windowpanes filter the light falling through them by different amounts.

Other relevant settings for the spotlight in the light dialog:

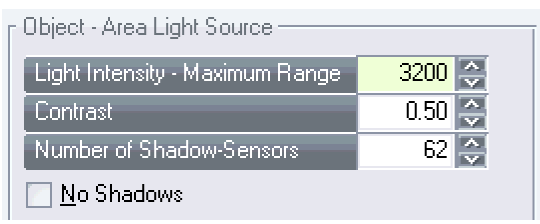
- Light-Mapping 363 - How to use bitmaps to project colored light patterns
- Lens Flares 365 - Visible light and light reflections in the camera lens
- Photon Mapping 370 - The photon emission parameters

14.6 Area Light - Convert an Ordinary Object into a Real Light Source

Area light sources, which are often used in modern architecture, e.g. in light panels, are difficult to simulate with standard point light sources. However, in MR-3D Designer you can convert any object you like into an area light source, just by activating the <Object = Area Light Source 290> object property in the Object Properties 289 dialog. Every point of an object will then be interpreted as a small subordinated point light source contributing in part to the whole of the object's light intensity. As a result, the rendering time for the picture calculation - especially when rendering shadows - increases with the object's point resolution, since each point is included with a separate light and shadow feeler 264 in the illumination process. NURBS 171 -patches are ideal area light objects because of their surface is formed from a regular structure with evenly-spaced points. Also, NURBS-patches have no thickness, so they can be easily installed in wall panellings.
Up to a maximum number of 500 light- and shadow sensors will be calculated for each area light

object (Take care, 500 light- and shadow sensors will increase the rendering time for illumination and shadows about the same factor!!!). If the object construction contains more points then samples are picked randomly from the object. The same applies if you reduce the amount of shadow sensors below the number of points contained in the object definition.

Adjusting the Light Properties for Area Lights - Once you have activated the <Object = Area Light Source> attribute, the object will also be listed in the light dialog together with all standard light types.
Then, if you change to the light dialog, you will be able to edit the light parameters for the area light object. You can define the light color, intensity and also the photon emission parameters. Area lights are, of course, included in the photon mapping process and therefore area lights too can emit photons just like all standard lights. The results will be even better than with direct lighting, because photons are emitted from the whole surface of an area light source, in contrast to the interpretation of a cluster of subordinated point light sources in the direct lighting algorithm.
The light color of an area light is independently of the material color. Think of the object as an ordinary body or as a container for a light source. If the light is switched off or is shining very dimly, you still have to take into account the light reflections from the container if it is illuminated by other lights in the scene.
Therefore the object material is calculated and interpreted as the surface of an ordinary object with all of its options, e.g. bitmap textures, reflection or transparencies. Only then is the self-luminosity added to the materials color with the light color. In an animation, for instance, you can animate the light color from dark to bright and the object will slowly begin to glow and illuminate the scene.
Part of this interpretation is that area light objects cast shadows when they are illuminated by other, brighter light sources. You can switch off the shadow casting by activating the <No Shadow [290] > object property for an area light.



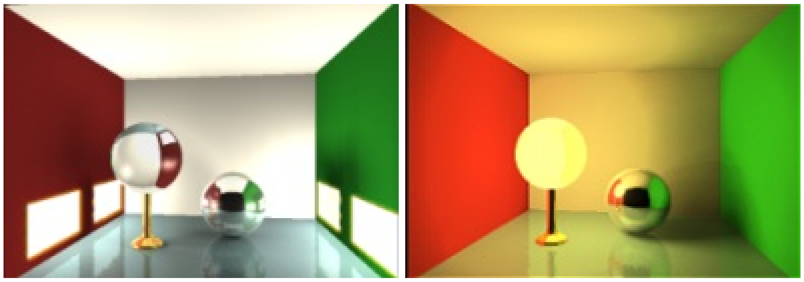
The parameters for area light objects in the light dialog

Light Intensity - Maximum Range - A reduction in light intensity with distance is calculated. *See also:* Lamp - Light Intensity-Maximum Range [353]

Contrast - See "Lamp - Contrast [353]"

Number of Shadow-Sensors - You can determine here the number of shadow-sensors [351] used to generate a soft-shadow effect for standard light objects . As already mentioned above, in an area light object for each point of the object interpreted as a subordinated light source, an individual shadow-sensor is calculated automatically. Only a maximum number of 500 light- and shadow sensors will be calculated for each area light object. If the object construction contains more points, then samples are picked at random from the object. The same applies if you reduce the number of shadow-sensors in order to keep to a reasonable rendering time - every additional shadow-sensor slows down the calculation considerably.

No Shadows - The area light don't cast shadows.



Two examples of area lights. In the picture on the left 4 NURBS patches converted to area lights illuminate the room. In the right picture the glass sphere was changed into an area light source. The picture on the left is rendered without Photon Mapping, since the light is scattered sufficiently by the many points of the 4 big light patches. The right picture was rendered with Photon Mapping for the indirect illumination. Just 25000 photons are enough to provide a soft warm ambient light for the scene. The soft shadows are achieved again by the shadow sensors directed to the points of the light sphere.
Both demo files can be found in the projects folder under "...projects/arealights/AreaLights-NURBS.cmo" and "projects/arealights/AreaLights-AnalyticalSphere.cmo".

Other relevant settings for area light objects in the light dialog:

- Photon Mapping 370 - The photon emission parameters

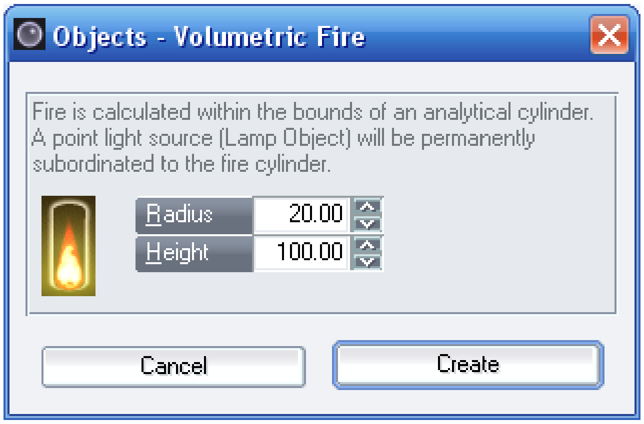
14.7 Volumetric Fire

Almost every kind of fire can be simulated with the volumetric fire object, beginning with smoothly-burning candle flames up to vividly-burning torches, camp fires or blazing seas of flames.

One primitive cylindrical fire object keeps the flexibility for many application areas. The examples above show flickering candle light, the blast of a jet engine, an explosion and open fire.

Volumetric fire is confined to a cylindrical bounding box with an additional lamp object fixed to it. Volumetric fire objects are created within the light dialog. Since a lamp object is subordinated automatically to the fire cylinder, all parameters for lamp lights can be edited when a lamp belonging to a fire object is selected in the light dialog. Furthermore, in addition to the lamp details, all parameters forming the volumetric fire are displayed. Basically, volumetric fire is calculated similarly to volumetric fog - applying a ray marching algorithm that takes samples of the fire density along the path through the fire cylinder, so most of the parameters describing the fire are similar to those describing the volumetric fog. Additional parameters define the color palette of the fire, the shaping within the cylinder, the turbulent flow and the flickering of the flame in an animation (by shifting of lamp position and varying the intensity).

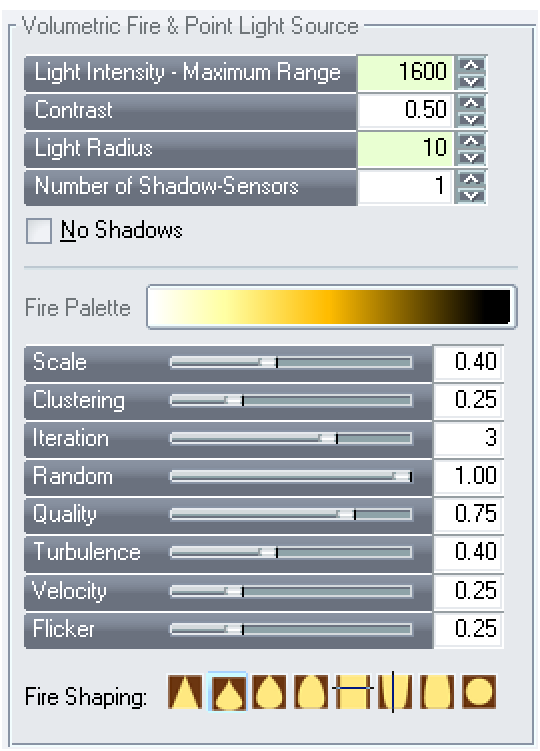
A new volumetric fire object is created by operating the "Add Light Object" <Volumetric Fire> button in the light dialog 341.



If you operate the <Volumetric Fire> button a small dialog is opened. In this dialog you can define the dimensions of the cylinder, that serves as a bounding volume for the fire object. An additional lamp object is created automatically with the cylinder and the lamp is permanently fixed hierarchically subordinate to the cylinder.

The Fire-Cylinder - The cylinder enclosing the fire is an analytical primitive 147 and can be positioned, scaled, rotated and animated just like any other analytical primitive. You can also copy or delete fire objects or insert them into hierarchies 140. The lamp object belonging to a fire object will automatically follow all modifications. Fire cylinder and lamp object always form a pair, e.g., when you switch off the lamp object in the light dialog, the fire cylinder is also switched off. The reverse also applies - if you switch off the cylinder object in the select objects 133 dialog.

Depiction in the Preview Window - Like with Lens Flare effects, a change in the various parameter settings for a volumetric fire object will be indicated by a redraw of the fire in the centre of the preview window, if the preview mode 341 "Lensflare - centered" is selected.



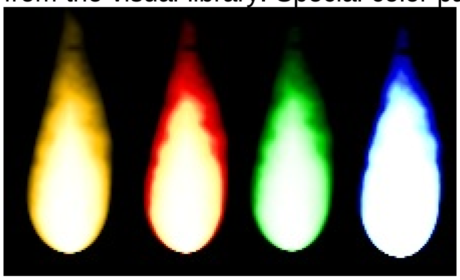
The parameters for the generation of flames

At the head of the parameter box you find again the already known settings for lamp objects. These parameters control the illumination attributes of the fire via the subordinated point light source:

Light Intensity, Light Radius and Number of Shadow-Sensors
See: Lamp - Point Light Source 353

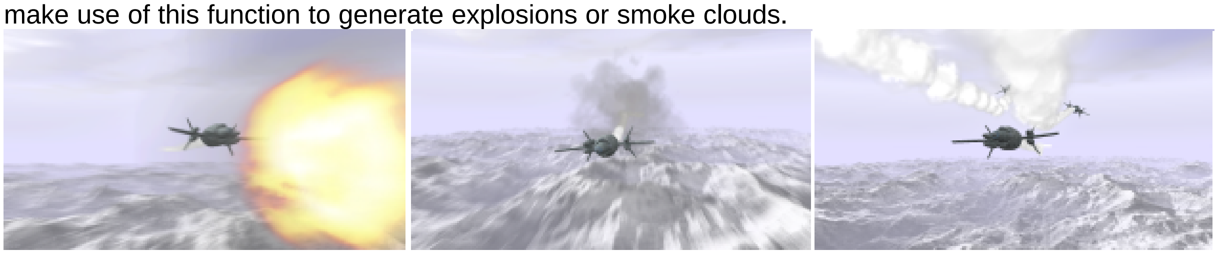
Under them you find all parameters influencing the shape, color and quality of the fire:

Fire Palette - To edit the colors for a fire object simply click on the color range bar. It opens the color range editor 30 where you can define your own colors or just load a pre-defined color range from the visual library. Special color palettes for fire are located in the fire folder of the color library.



Note! The light emanating from a fire object is defined with the light color as with other light objects. The fire color palette only controls the color range used to draw the flames.

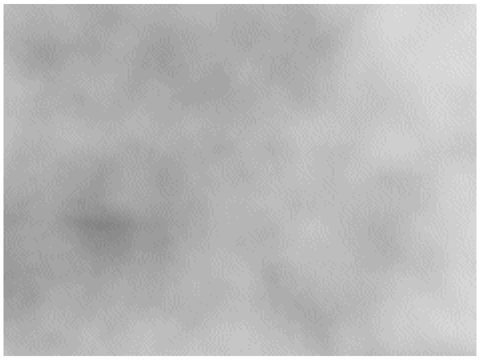
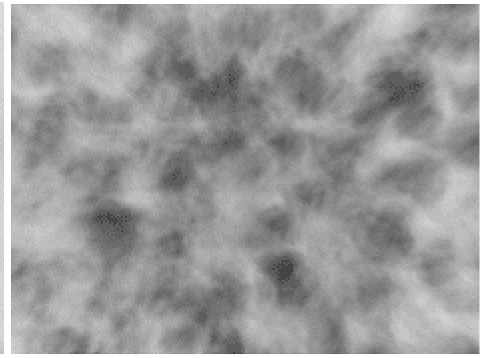
The fire palette can be animated, i.e., if you change the color range on different time positions in an animation 400 this change will automatically be recorded in a corresponding parameter key and a transition between different palettes is calculated on framepositions between keyframes. You can make use of this function to generate explosions or smoke clouds.



The picture above shows details from a short film sequence with explosions and smoke effects. The explosion sequence and the smoke picture used for the billboard projections were rendered in advance using the volumetric fire object. See our tutorials "Explosive Fire 113" and "Billboard Smoke 117" for a detailed explanation of how to create such effects.

Fire is created using a fractal algorithm 313, so most probably you will already recognize many of the following parameters from other functions in MR-3D Designer, like clouds, volumetric fog or landscape design.

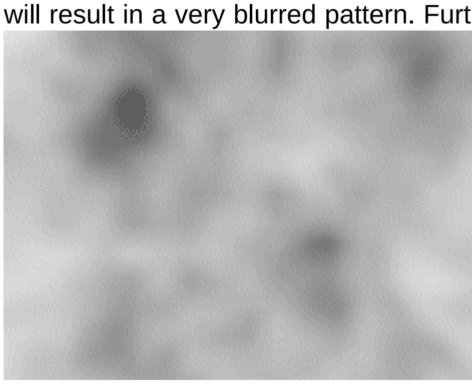
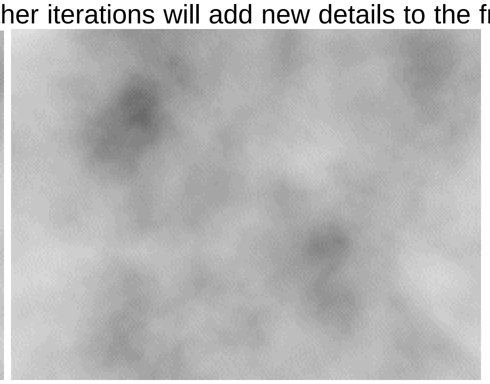
Scale - The underlying fractal patterns used in a fire calculation are rendered close together or wide apart depending on the scale value entered.

The pictures above are taken from the chapter on volumetric fog - but both volumetric fire and volumetric fog are founded on the same mathematical algorithms. These pictures show very well how the scaling function works. The picture on the left was generated with a small scale value which results in a smooth random pattern stretched wide apart. This is the right setting for soft and steady candle flames. With a higher value for the scale function, the random pattern is rendered closer together with much more details and frequent gaps in the pattern. You can say, the higher the scale value, the wilder the fire becomes.

Clustering - Adds more detail to the fire by increasing the gaps in the random fractal pattern, similar to the "Thin Out" parameter of the volumetric fog function. As a result the fire becomes even wilder with the flames collapsing and flare up again frequently.

Iteration - The number of iterations defines the level of detail for the fire pattern. A single iteration will result in a very blurred pattern. Further iterations will add new details to the fractal noise.

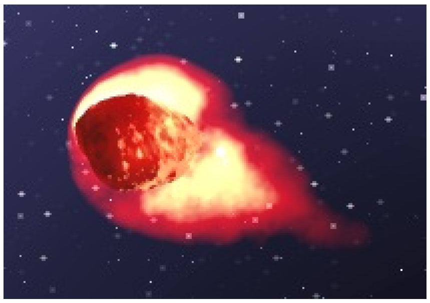
Again, two pictures from the volumetric fog section. The picture on the left shows a fractal pattern rendered from 2 iterations, while 4 iterations were used for the picture on the right. Again we can say that a small number of iterations is suitable for soft-shaped and smoothly-burning candle flames whereas more iterations are ideal for vividly burning flames with detailed, ragged outlines.

Random - Initialises the random generator for the fractal algorithm. If several fire objects are in the scene, e.g. a number of candles on a candelabrum, each candle should be initialised with a different random number so that the flames burn and dance with different rhythms.

Quality - Fire is calculated volumetrically in MR-3D Designer. This is done by tracing a viewing ray through the pillar of fire and taking many samples of fire densities along its path through the fire. With help of the quality parameter you define the intervals at which new samples are taken in the fire. With a higher quality value the step-width becomes shorter and the number of fire densities calculated in the fire cylinder increases considerably.
If you render small candle flames or torch flames you can use high quality values without hesitation, but if you want to produce a sea of flames where the viewing ray has to travel long distances through the fire, then it is better to use small values for the preview renderings and only for the final rendering increase the quality again - to a value of about 0.90 or higher.

Turbulence - This parameter increases turbulence in the fire's movement. This applies both for the fractal pattern within the fire as well as for the outline of the flame, which begins to flicker more wildly.

Velocity - Determines the speed of the fire movement. The flames always move upwards along the positive y-object axis 8. Since the fire cylinder can be freely rotated you can apply the fire object also for other effects, e.g. a jet propulsion or a flaming tail of a comet - you just have to position and align the fire cylinder so that the positive y-object axis is directed opposite to the movement direction.



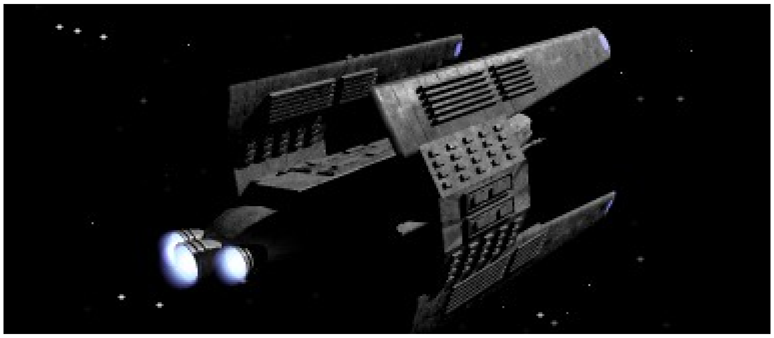
This example shows a flaming meteor entering the atmosphere

Flicker - This parameter affects the illumination attributes of the point light source of the fire object. Dependant on the flicker value and the velocity of the fire movement, the lamp origin is shifted slightly back and forth in a turbulent motion. Simultaneously, the light color changes slightly so that the restlessness of the flames is also imparted to the illumination of the scene.

This picture shows a still from a candle flame animation. In the running animation - you find the original project file under ".../projects/volumetricfire/candles_anim.cmo" - you can observe how the flickering of the candle flames is imparted to the illumination of the room.

Fire Shaping 

The cylinder defines the bounding volume in which the fire is calculated. You an apply another 8 basic shapes to further form the fire within the cylinder. For instance, you can apply an onion shape for small candle flames or a cone- or egg-shaped form for a camp fire.



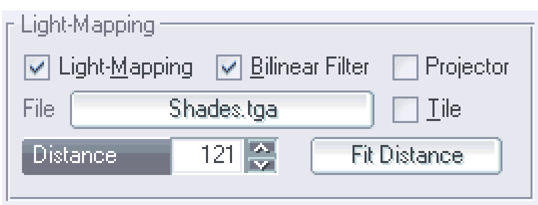
The engines of this space ship are powered by three spherical shaped volumetric fire objects.

Another demo animation. The project file is also part of the MR-3D Designer installation under "...projects/volumetricfire/fire_logo_anim.cmo". Six individual overlapping fire cylinders form a continuous line of fire. In the animation the cylinders are elongated slowly from small discs until the flames engulf the CyberMotion logo. Then the camera moves forward and dives into the flames.

Other relevant settings for the point light source subordinated to the fire cylinder:

- Lens Flares 365 - Visible light and light reflections in the camera lens
- Photon Mapping 370 - The photon emission parameters

14.8 Light-Mapping



A simple but impressive effect combines light sources with bitmaps used as light filters. This allows you to simulate the most complex shadows by using fast bitmap operations. Imagine the horizontal strips of a window screen, or fences, complex window frames, rotating disco spotlights (using a spot type light source rotating in front of a multicolored bitmap), colored shadows of tiffany lamps or windows, etc.

Interpretation of bitmap colors:

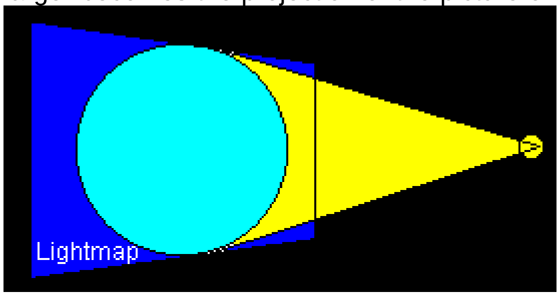
- White - completely transparent.
- Black - no transparency.
- Colored - filtering the light color.

Light mapping is mainly a feature for spotlights 354, but can be applied in the same way to parallel lights 351. Light mapping is not applied to normal lamps as the uniform radiation of the point light gives no clue about the direction of the light map projection.

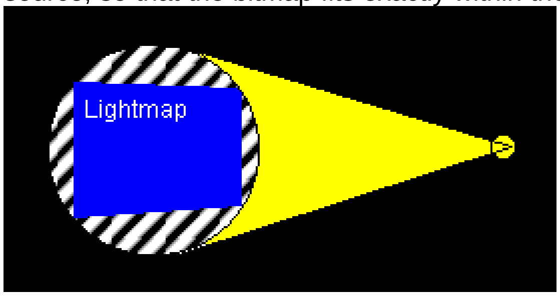
<Light-Mapping> - Activate this button to switch on light mapping.
<Bilinear Filter> - Interpolates the bitmap colors, thus smoothing undesired step effects and pixelized transitions that become especially clearly visible when the picture is magnified by the shadow casting.
Operate the **<File>** button to display the fileselectorbox and then select a bitmap suitable as a light map.
<Tile> - Light maps can be repeated alongside and above each other. This function is optional for spot lamps. However for parallel lights this option is applied automatically, since there is no origin for this type of light and therefore neither can there be a single origin for a picture.

There are some additional features that apply for spotlights only:

<Distance> - You can change the size of a projection by using the distance parameter. Imagine a slide picture you put in front of a flashlight. The closer you move the slide to the flash light, the larger becomes the projection of the picture on the wall and vice versa.



Use the **<Fit Distance>**-Button to automatically adjust the distance of the bitmap from the light source, so that the bitmap fits exactly within the light cone.

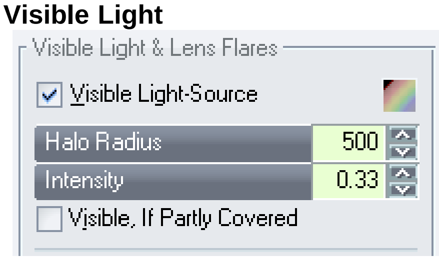


If the <Projector> option is switched on and you select the <Fit Distance> button, then the distance of the bitmap from the light source is automatically adjusted, so that the picture lies completely within the light cone. Furthermore the <projector> function changes the light cone to a rectangular projection type, just like a real projector. Light rays that don't pass through the bitmap are simply left out in the rendering process.
But be aware that the spot interpolation function is still calculated for this projection type, so that the light intensity falls off continuously from the center of the picture to the rectangle's edges if spot interpolation is not switched off.

14.9 Visible Light & Lens Flares

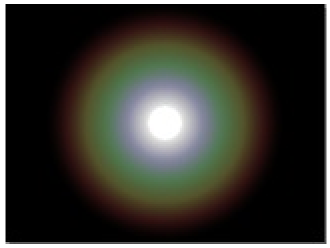
A picture from a real camera can contain faults due to camera optics. If, for example, the light from a light source shines directly into a camera, several lens flares appear in the camera's lens. This accounts for the well-known star, circle, or annular lens flares in the pictures. In the photo and film industry this effect is not always viewed with pleasure, and they often go to not inconsiderable expense to avoid this effect. However, people have become so accustomed to these picture faults that these effects you can help to considerably increase the degree of reality of computer-generated pictures.
Here, in the light dialog, you can decide if such lens flares are to be generated for individual lamps or spotlamps. On the right side the light dialog, you have a large number of different parameters with which you can determine the type, intensity, and size of the flares for each light object. Furthermore, it also allows you to animate the light effects in various ways 401. By rapidly expanding light halos and rings and rotating the star-rays, you can simulate many effects - from the rising sun to light explosions. Apart from the buttons to switch on or off the lens flares for every individual light object there is also a global <Lens Flares> button in the render options 263 dialog - which allows you to globally switch the lens flares of all light sources on or off.

Visible Light



If you switch on the option <Visible light source> a spherical light halo is created about each light source. This halo consists of an inner circle defined by the **Light Radius** and an outer halo defined by the **Halo Radius** , where the color path of the light color fades towards the halo color 341. Simultaneously, the intensity of the halo reduces more and more towards the edge. If you also switch on the rainbow colored button, then a rainbow filter will be applied to the halo color range. The light radius is part of the basic light settings located on the left side of the dialog. If the halo radius entered here in the Lens Flares box is smaller than the light radius, then only the inner light circle is drawn without producing a halo effect.

Intensity - Lens flares are additional intensities, added to the picture in post processing after finishing the picture calculation (exception - sun light 351 objects are rendered as visible light spheres in the background as well as overlaid with the lens flare effects afterwards). The intensity parameter controls the intensity of the overlaid lens flares.

Examples: On the left you see a visible light source with a white light color and a blue color for the halo. On the right you see a white light color and also white for the halo, but this time with the rainbow filter switched on.

Visible, If Partly Covered - Normally, lens flares appear only if the light source is directly visible and not masked by other objects. For many purposes, however, it is very useful if the light source with the halo shines from behind objects, so that you can still see parts of the halo. In this event, the halo is not interpreted as a lens flare in the camera, but as a halo about the light source, as it seen in fog or rain, for example. This halo originates from the light reflections of the light-source through surrounding fine water droplets.

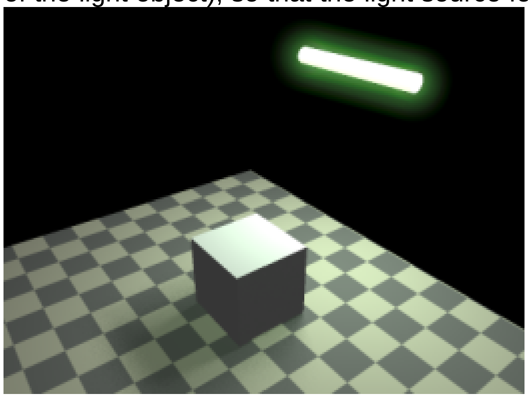
In this picture are three light sources with identical parameters. The light source above in the middle is visible and so all light effects are valid. The light-source on the lower-left is partially visible. The option <Visible, if partly covered> is switched on, and you can still see the halo surrounding the light source. The light effects that would originate through reflections within the camera lens, (i.e. light stars, rings or spots) are, however, not calculated for masked lights. With the light source on the lower right we have one more special case - it is behind a transparent object 309. In this event the lens flares are supplied but, additionally, the color and intensity of the light source is filtered through the transparent object.

Spotlights and Visibility of Lens Flares - With spot-lamps, lens flares and light halos appear only if the camera is within the light cone thrown by the spot. If, on the other hand, a visible halo is also desired, if the spotlight cone is aimed past the camera, the option <Visible, if partly covered> must be switched on. Light-stars and other effects, however, are only calculated if the camera is in spotlight cone.

Visible Spotlight-Cones - In this chapter only lens-reflections and visible light-sources are discussed. MR-3D Designer can however also produce realistically renderedspotlight-cones.
See also: Render Options - Volumetric Spotlight 273 and Spotlight - Volumetric Spotlight Cone 354

"Real" Light Objects - There are several ways to produce visible light objects in MR-3D Designer: With lens flares and light halos you can very easily add visible light effects, but in the end lens flares are only a simple post processing effect.
Objects of any shape can be converted to light objects with real illumination attributes, but rendering of these area lights 356 is very costly due to the manifold light- and shadow-sensors to evaluate. There is, incidentally, one more little trick with which you can generate a visible light source that combines wonderfully with the lens flare effects. Generate an object (for instance, a sphere 147) and

choose a material color that matches with the light color. Then, set the material's <Glow [309]> attribute to the maximum of 1.00. Also, switch on the option <No Shadows> [290] as an object attribute in the Object Properties dialog. Then place a light source precisely in the center of this object. Finally, in the Object Selection [133] window, arrange the light source in a hierarchically subordinated position to the new light object (simply drag the name of the light source on the name of the light object), so that the light source follows every movement of the light object.



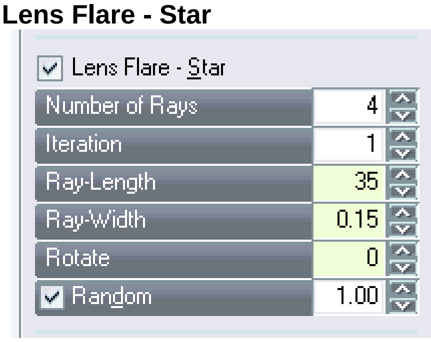
Here is an example of a luminescent tube - a simple cylinder (shadow casting switched off) with 3 lamps inside, activated soft shadows and material halo [309].

Lens Flare - Ring

Annular lens flares are sometimes seen around bright light sources. You can simulate this light effect by switching on the option <Lens flare - ring>. Again, you can overlay the light ring with a rainbow filter when you select the rainbow colored button.

Radius: The <Radius> parameter allows you to edit the size of the radius of the halo ring.
Width: The ring width is determined by this value.

Lens Flare - Star



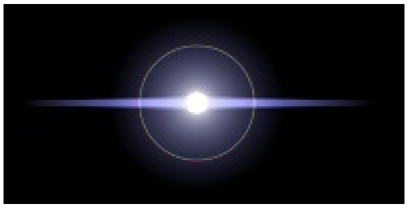
The option <Lens flare - Star> creates a star-shaped lens flare for a light source.

Number of Rays: The number of base ray-arms. A star can comprise of a minimum of 2 to a maximum of 40 rays.
Iteration: The number of iterations by which additional rays are generated on top of the base rays. For example, if you input a value of 4 for the base rays, and a value of 2 for the iteration, then the end result is star of 4 base rays plus 4 supplementary rays in the first iteration run-through plus 8 supplementary rays in the second iteration run-through. This produces 16 rays in total. The rays added are somewhat thinner and shorter per iteration. The minimum iteration number is 0, maximum is 4.
Ray-Length: The basic size of a single ray-arm.
Ray-Width: The width of a ray-arm. A value of 0.01 corresponds to very narrow rays and a maximum value of 1 produces a very wide ray - in which event the individual rays are barely distinguishable.
Rotate: The angle of the first ray of the star from the vertical. With a low or odd basic ray number (3,4, or 5) it often appears better and more realistic if you give the star a slight rotation, so that it is somewhat out of alignment with the edges of the picture. Another beautiful effect is also possible: a star with only 2 rays and no iteration, rotated through 90 degrees, lies exactly horizontal. You have probably seen this type of light effect already in many films.
The angle value can be animated 401. By the inputting different angles in keyframes, you can produce a rotation of the star effect in animations 401.
Random: The random-function generates an asymmetrical star halo. The random generator can be initialized with a value between 0.01 and 1 for different random results.
Examples:

Number of rays: 4, iteration: 1, width: 0.08.

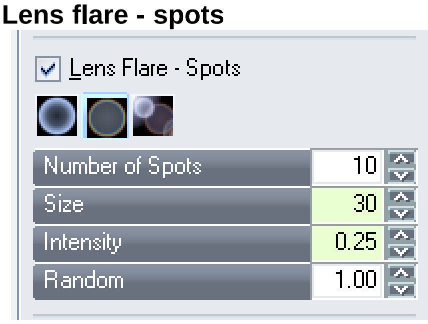
Number of rays: 4, iteration: 2, width: 0.35, length: 85.

Number of rays: 4, iteration: 3, width: 0.70, length: 85.

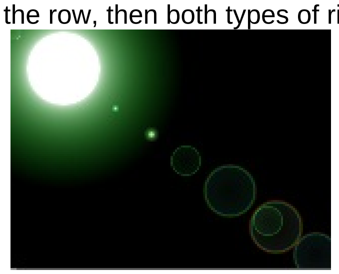


Number of rays: 2, iteration: 0, width: 0.04, length: 85, rotation: 90.
Light radius: 8, halo radius: 15.
Light ring: radius: 18, width: 3

Lens flare - spots

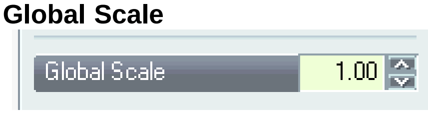


This type of lens flare generates multicolored rings and circles running out from the light source, diagonally through the focus of the picture. In films, this effect is often seen if, during filming in the countryside, the camera, in panning, catches the sun and lets these spots run through the picture. You can also use this light effect very effectively in animations 401. With the three ring-type buttons you can again choose to overlay the rings with a rainbow filter or not. If you select the third button in the row, then both types of rings will be mixed.



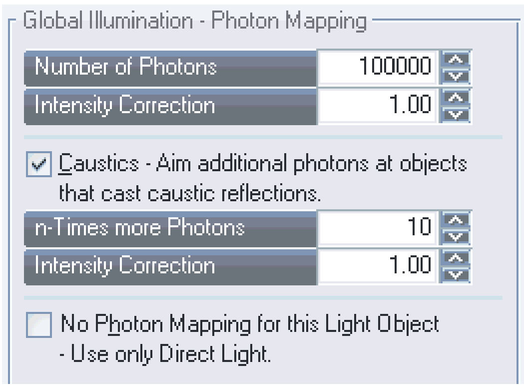
Number: This value gives the number of spots to be calculated.
Size: The maximum size of a spot. The actual size depends on the random generator and, as with all other light effects, also on the distance of the light source from the camera.
Intensity: The maximum intensity of a light ring from 0.01 to 1. Additionally, the intensity of the color of the light-ring is determined by the colors of the light and halo and is dependant on a random value.
Random: The starting value for the random generator. Changing the initialization value can generate many different variations in the light effect.

Global Scale



Once the general shape has been edited via the various effect parameters you can resize the resulting lens flare effect with the **Global Scale** parameter as a whole object.

14.10 Photon Emission Parameters



In the "Global Illumination - Photon Mapping"-box in the light dialog the parameters for the photon emission are entered.

For a detailed introduction to photon mapping see:
Photon Mapping 277 - Introduction and examples
Render Options 264 - Global Illumination - Raytracing + Photon Mapping
Material Dialog 290 - Photon mapping object properties

Each light source in MR-3D Designer can emit photons for the photon mapping process. However, you can also exclude individual light objects from this process. If you switch on the **<No Photon Mapping for this Light Object>** option in the light dialog, then, instead of emitting photons, the corresponding light object will illuminate the scene only with conventional direct light algorithms, no matter which rendering mode is activated. You can use this function for little lamps in instrument controls or for lights far away in the background or, e.g., for spot lamps illuminating only small parts of the scene. To save rendering time just switch on this function for all light objects that do not contribute much to the general illumination in the scene.

Emit Photons from Both Sides of a Facet - This button is only applicable for area light objects. Area lights emit photons from all of the facets building the surface of the object. However, if the light object is formed from a closed shape, e.g. a sphere or a cube, it is not necessary to emit photons from the inner side of the light object. Therefore, as a standard, photons are emitted only in direction of the surface normals. (The surface normals of closed objects created in MR-3D Designer always show outwards).
However, if you want to use open objects or flat surfaces without thickness as light objects, then simply switch on the <Emit Photons from Both Sides of a Facet>.
Another possibility is to apply flat NURBS-patches for light panellings. To prevent the emission of superflous photons backwards into the walls, again switch off the emission of photons from both sides of a facet. With the menu function "View - Normals" you can include the depiction of normals in the viewport windows. Then, if the normals of a NURBS-patch are faced towards the wall, simply rotate the NURBS-panel by 180 degrees.

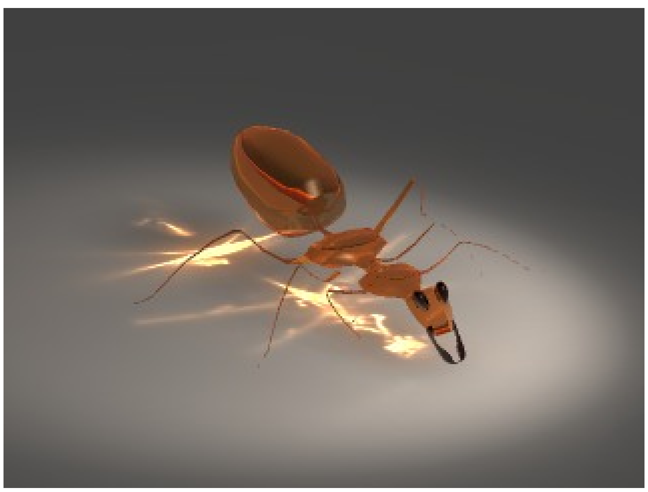
Number of Photons - For each light source you can enter an individual number of photons to emit for the photon tracing. Numbers between 20,000 up to 10,000,000 are practical, depending on the complexity of the scene and the rendering mode. When applying a photon map only for indirect illumination 277 then you can manage with relative small photon maps. To estimate the general area brightness in a small room you can do with photon maps of about 50,000 photons. However, if you apply photon mapping as a global illumination 277 model without direct lighting then usually a million and more photons are needed to cover all details in the scene. Furthermore, you have to enter a corresponding high number of photons to gather for the photon pool 277 to get smooth intensity transitions and to prevent a spotty appearance (about 600 up to 2500 and more, depending on the size of the photon map). This requires a fast cpu and a high memory capacity of at least 256mb and more.

Intensity Correction - An intensity can be entered for each light source via the "Light Intensity - Maximum Range 353" parameter. This light intensity fits best to the direct light algorithm that is used in conventional rendering modes, but, since photon mapping applies a totally different illumination approach based on a more physical model, an intensity correction value is needed to match the light intensities when switching back and forth between simple raytracing and photon tracing.

If you plan to render your scene with the photon mapping 277 algorithm you ought to follow this course:
Adjust the light intensities with normal raytracing 264 for the preview renderings. Then render a first test picture in photon-mapping mode. To balance the intensity variations in the picture caused by the two different illumination methods, adjust the light intensity for the photon emission via the intensity correction parameters instead of changing the general "Light Intensity - Maximum Range" parameter. Thus you can switch back to raytracing mode for faster previews when extending and editing your scene and light settings and in the end you can turn again to photon tracing without having to adjust again the light intensities for the photon mapping process.

Caustics

Caustics are light reflections from highly specular surfaces or, e.g., the light gathered in a focal point after transmission through a glass lense.



Example for caustic light reflections beneath a little glass figurine, caused by photons that were refracted when transmitting through the glass. The corresponding project file is part of the MR-3D Designer installation under "/projects/caustics/ant.cmo".

Caustics Photons are stored in a separate Photon Map, the so-called Caustics Map:
Usually a photon map is evaluated only for the indirect illumination in combination with direct light for the main illumination and shadow calculations. For this purpose it will do to emit only several ten thousands of photons into the scene so we can average the general area brightness at each point in the scene. Caustic reflections, on the other hand, are often sharp outlined light patterns, like in our figurine picture shown above. It would be impossible to render these light reflections when only a few photons had been scattered around - you wouldn't even see a glimpse of the light focused beneath the figurine. Now, it would be also ridiculous to emit millions of photons into the scene and having to evaluate huge photon maps afterwards, only to cover the light reflections of a little specular object somewhere in the scene. That's why we have to manage to different photon maps, one for the global photon map and the general illumination, and one separate caustic map only for those photons that have been reflected or transmitted via a specular surface before hitting a diffuse surface. The caustic map is build in a second photon tracing pass where additional caustic photons

are aimed only towards such objects, that are highly specular or transparent and own the material attribute <Caustics - Aim Additional Photons at Object [290] >. The evaluation of the two different photon maps requires also separate photon pools [277] . For the global photon pool much more photons have to be gathered for the averaging process, so that soft and clean light transitions can be calculated for the area brightness. However, for the caustics pool we need comparatively fewer photons, because we want sharp and clearly visible light reflections.

The caustics parameters in the light dialog:

Caustics - Aim additional photons at objects that cast caustic reflections - The emission of additional caustic photons can be switched on or off for each light object separately. You should activate the caustic photon emission only for lights that stand nearby or are directed towards objects, that own the object attribute <Caustics - Aim Additional Photons at Object [290] >. If you want to render a picture with the camera focused on a caustics object as the main part of the image, it is advisable to use a spot lamp for the illumination because then the stream of photons can be aimed directly towards the target object.

n-Times more Photons - Determines the multiple of additionaly emitted photons to create the caustics effect. Instead of specifying a certain number of photons this time you just have to enter a factor that describes how much more photons per area have to be emitted towards the caustic objects than for the global photons. Take again the glass figurine as an example. The emission of global photons was set to 50,000 photons. For the emission of caustic photons the factor was set to the maximum value of 100 via the <n-Times more Photons>-Parameter. During the processing of the photon map in the first pass, when the 50,000 global photons are emitted, about 1100 photons found their way through the glass figure and were saved in the caustic map. Then the additional emission of caustic photons is started with 100-times more photons, this is 100 * 50,000 = 5 millions photons. But this is only a fictitious value that only specifies how much more photons per area are emitted in general. Since caustic photons are only directed towards caustic objects in the end "only" 200,000 photons find their way into the caustics map. This is more than enough for a sharp representation of the caustic light effects under the glass figure.

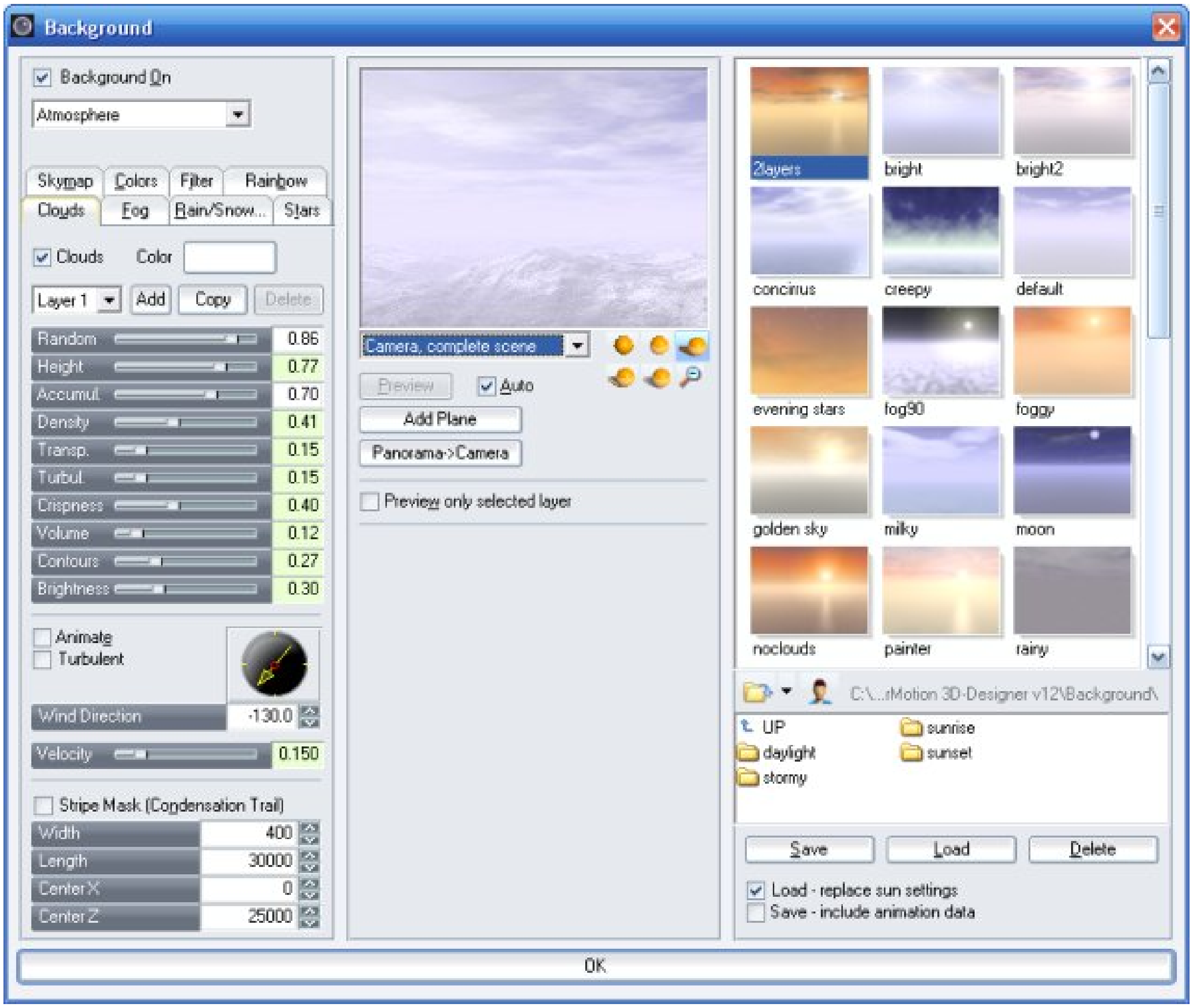
Part XV

15 Background - Colors, Bitmaps or a Complex Atmosphere

How to put a picture or a simple color range in the background or rather generate a complex atmosphere with fog, clouds and a colorful sky.

The individual background models:

15.1 The Background Dialog



 - Menu "Options - Background" - Short Cut:  + "B".

In the background dialog you can stipulate a color gradient in the background on rendering, or a complex atmospheric sky model with or without clouds and fog, or simply a bitmap.
The background is managed as an object with the following attributes:

1. The background can be switched on or off here in the background dialog via the <Background On> button or just like other objects in the select objects 133 dialog by de-selecting the background object there. If the background is switched off, it simply appears in black.
2. The background can be animated. The background object can be worked on in the animation dialog 409 in exactly the same manner as all other objects, e.g. copy, delete or insert new keyframes. Instead of object movements, however, the background colors, cloud and fog parameters are animated.

The Preview Window

The preview window in the central part of the dialog provides a quick preview of the scene when changing background settings. There is a selector box underneath the preview window providing several preview modes:

- *Panorama, no objects* - Only the background is rendered, as seen from a predefined panoramic camera view. The panoramic view is ideal for adjusting parameters of landscape scenes with atmospheric backgrounds.

- *Panorama, only planes* - In addition to the background all plane objects in the scene are displayed. Most projects contain a plane to clip the scene downwards and towards the horizon. If not, and an atmospheric background without fog is applied, the background will simply be mirrored at the horizon.
- *Panorama, complete scene* - The complete scene is rendered into the preview window.
- *Camera - no objects, only planes and complete scene* - the same as the panoramic view but except this time the current camera settings are used to render the scene.

Copy Panoramic Camera Settings to Current Scene Camera

You can copy the settings for the standard panoramic view to the current scene camera by operating the <Panorama -> Camera> button.

Add Plane Object

If you have forgotten to create a plane you don't need to leave the dialog and change into the Create Plane 186 dialog. Simply operate the <Add Plane> button to generate a new plane for your scene. This button is only available if no plane as yet exists in the scene.

Visual Background Library

On the right side of the dialog the visual background library is located. Double click on a thumbnail picture to load an existing background file and modify it to your needs or add your own backgrounds to the library using the save function. All types of backgrounds can be saved, although it would be unreasonable to save a simple color range background, for instance, since color range files can be saved separately in the color range library 30 anyway.

See also: General library functions 28 to save, load or delete entries or to create sub-folders.

In addition to the general library functions that are applicable to all libraries in MR-3D Designer, there are some special settings for the saving and loading of background files:

Load - replace sun settings - In addition to the background parameters a background library file also saves the settings for activated sun light objects, since the appearance of a cloud formation depends strongly on the light incidence of the sun. You can decide via the <Load - replace sun settings>, if you want to overwrite the current settings for the sun with the data from the library file or not. This setting effects only activated sun lights, all other light types including parallel light sources with the <sun> option switched off will not be saved or replaced by a background file.

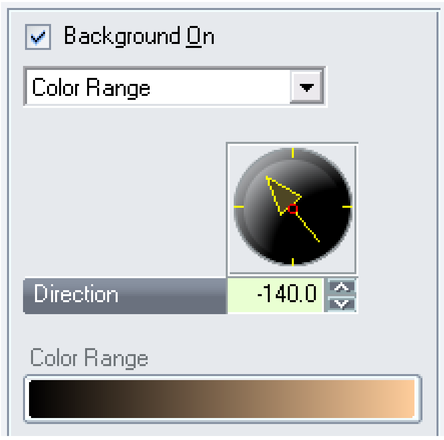
Save - includes animation data - If this button is selected the background object will be saved including all keyframes related to it (if animated). Otherwise only the parameters of the current keyframe position are saved - resulting in the background as it is presented in the preview window.

Select a Background Model

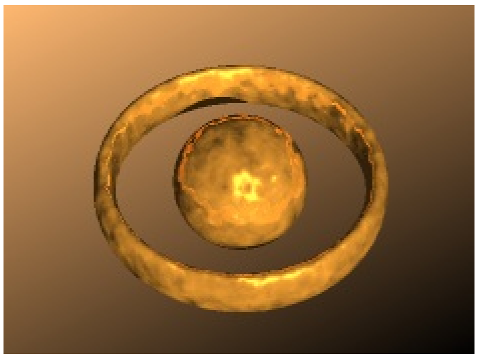
In the upper left of the dialog there is a selector box in which you can choose the type of background model you want to apply. Depending on the selected background the relevant parameters appear on the left side of the dialog:

- Simple Color Range 377
- Atmoshere 378 - Including background colors, filters, clouds, fog, stars, snow, rain...
- Background Bitmap 394

15.2 Simple Color Range



If you select Color Range in the background select box, an angle instrument appears directly beneath the select box, similar to those in the light or camera menu. This indicates the direction of the color graduations. Simply click in the instrument and drag the pointer to the desired position. To edit the colors for the color range simply click on the color range bar beneath the angle instrument - iIt opens the color range editor 30 where you can load color ranges from a visual library or define your own.



Example of a color range shining through two transparent objects

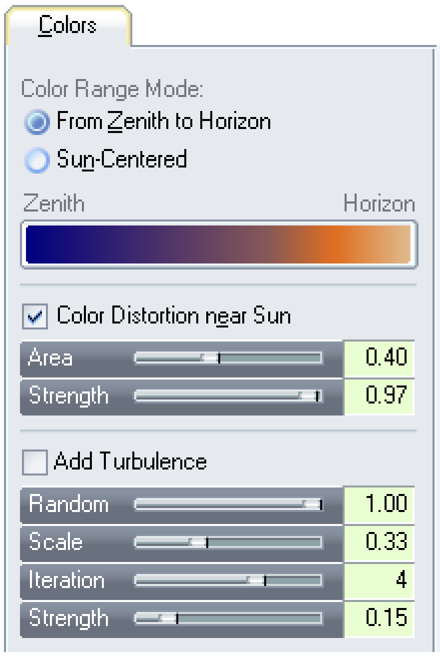
The following should be noted concerning the simple color range:
Only a simple two-dimensional color path is calculated for the background of the picture. No three-dimensional background effects apply so you cannot expect the background to change if, for example, you move the camera towards it. Nor can an object cast a shadow on the color path. However, you can see the color range through transparent objects but, as there is no real three-dimensional background, no distortion due to refraction will be seen through transparent materials. A three-dimensional background model (such as the atmosphere model) would be appropriate if the background is to be rendered with refraction seen through a transparent object - for instance, a glass sphere.

15.3 Atmosphere

The atmosphere of a planet is a complex and multi-layered matter. It provides the oxygen to breathe, the ozone layer that protects us from the hard ultraviolet radiation, transports essential humidity in its clouds and conjures up the most wonderful colors and cloud formations in the sky. Now, with MR-3D Designer you can catch some of this enormous variety in your picture compositions. Choose the "Atmosphere" entry in the background select box and the seven sets of parameters for the creation of an atmosphere will step to the fore. All effects can be freely combined or switched off, just as you like. For instance, if you choose to add the starfield to a cloudy atmosphere, it will automatically be filtered by the clouds and the fog. But if only the starfield is activated, without clouds and fog, then you can use it as a background for scenes in outer space.

- **Sky Colors** 379
  Two different approaches for rendering sky graduations
- **Skymap** 381
  Instead of a sky graduation, a simple background bitmap or a skymap providing a full range of view can be copied to the background
- **Atmospheric Filter** 381
  Why turns the sky red at sunset?
- **Clouds** 382
  Adding cloud layers to the atmosphere
- **Fog** 385
  Atmospheric fog and volumetric ground fog
- **Rainbow** 389
  The phenomenom of caustic light reflections in the atmosphere
- **Rain, Snow and Floating Particles** 390
  At the touch of a button it begins to rain or snow
- **Starfield** 392
  Use the starfield as part of the background model or alternatively as an animated starfield for space travel.

15.3.1 Sky Colors



Choose the **<Colors>** tab in the atmosphere selection to insert the parameters for the sky colors.

Edit Color Palette - To edit the colors for the sky simply click on the color range bar. It opens the color range editor 30 where you can define your own colors or just load a pre-defined color range from the visual library.

Color Range Mode
The color range of a 3-dimensional sky is generated over a large sphere surrounding the 3D area. This technique provides a true 3-dimensional color range with the attribute that the sky/horizon moves with the camera-movement - as does a real sky/horizon. Also, like any other object, the sky is mirrored in reflective objects and is correctly depicted through refractive, transparent materials. There are 2 different sky modes to render the start- and end point of the color range:

- **From Zenith to Horizon** - The color range graduates from the start color at the zenith, through the color range to the horizon.
- **Sun-Centered** - The color range graduates from the center of the sun over the complete sky sphere towards the opposite side of the sphere. This method has a big advantage when rendering animations with camera pan shots. If you define a color palette for a sunset, starting with a bright color graduating to very dark blue colors in the end, this color range will cover the whole sphere and is rendered correctly from every camera position. The area surrounding the sun will always be bright and if you turn round the other side of the horizon will be rendered automatically with the dark colors of the color range. Of course, there has to be a parallel light 351 object with activated sun mode 351, otherwise the colors will graduate automatically from zenith to horizon again.

Both color range modes have their pros and cons. For daylight shots with the sun standing high above a color range starting from a blue tone and graduating from zenith to horizon to a very bright blue or white color is often the best solution. For colorful sunsets with bright areas around the sun and dark areas on the opposite side of the horizon the sun-centered option fits better.

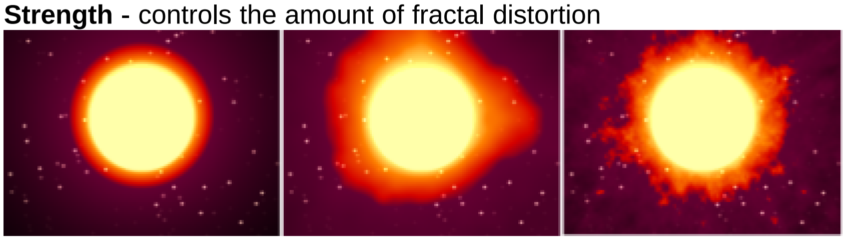
Only for the color range from zenith to horizon there are two additional parameters to lighten up the area around the sun. In principle, this is a combination of the two models described above.

Color distortion near sun - If a parallel light object is acting as a sun 351 the color range is again calculated from zenith to horizon but an additional filter function also distorts the color range around the sun. To include this effect the **<Color distortion near sun>** button underneath the color range bar has to be activated. There are two additional parameters:
Area - Defines the area of the sky around the sun that is within the distortion radius.
Strength - The strength of the filter function.

The illustration above shows a sky with a color range starting from blue in the zenith graduating to a very light blue at the horizon. For the left picture no color distortion has been calculated and therefore it resembles more a cool moon standing in the sky than a bright sun. The right picture shows the difference. With activated sun distortion the color range graduates around the sun as well to the horizon, creating the impression of a glaring firmament.

Turbulent Color Range
With the turbulence parameters you can add fractal structures to the color transitions - this way you can create coronas around a sun or the impression of an additional cloud bank at the horizon.

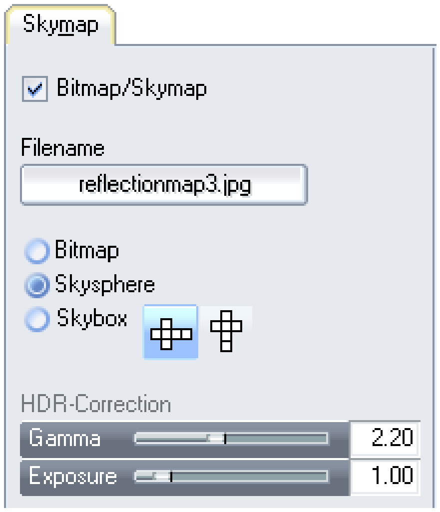
Parameters:
Random, Scale and Iteration - see fractal structures 313
Strength - controls the amount of fractal distortion



Example 1: A sun-centered color range changes with additional turbulence and increasing frequency (Scale parameter) into a flaming corona.

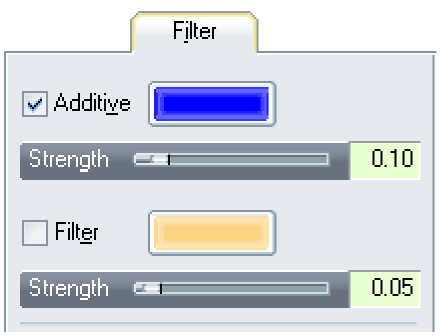
Example 2: A color range from zenith to horizon. It appears as if volumetric ground fog 385 has been switched on but its just a blue to white color range with added turbulence.

15.3.2 Skymap



Instead of painting a sky graduation you can decide to copy a simple bitmap into the background or even a skymap providing a full panoramic view. If the <Bitmap/Skymap> button is selected, the bitmap projection will replace the sky graduation adjusted on the Sky Colors 379 page. Then, all atmospheric effects like clouds, fog, stars, rain, etc., will be painted into the bitmap background. The parameters on this page are identical to those for a bitmap background without atmosphere. Read the corresponding chapter Background Bitmap, Skymap or Environment Map 394 for more information about this background model and it's vast application fields.

15.3.3 Atmospheric Filter



Choose the "Atmosphere" - **< Filter >** tab to bring the parameters for the atmospheric color filters to the fore.

As a light ray traverses an atmosphere some light is extinguished and some light may be added by emission and scattering. This results in a change of color with distance, i.e. dark backgrounds becoming bluer and light ones becoming redder with increasing distance.
Note that the atmospheric color filters build on the atmospheric fog effect - if the atmospheric fog is switched off then the color filters will also have no effect!

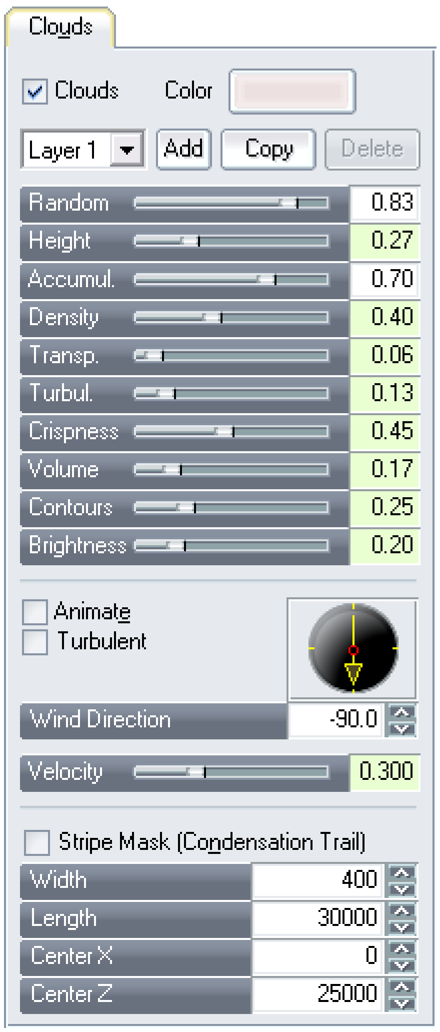
Additive - A blue component is added to the scene colors, increasing with distance. Daylight renderings, of mountain sceneries in particular, profit from this effect because only thent does the picture have a real impression of depth and distance. Click on the color button if you want to edit the additive color component.

Filter - Light colors, i.e. clouds, snow or fog, are filtered with this color with increasing distance.

With this filter you can simulate the reddening effect when the sun sets. You can also apply this filter for daylight scenes, for instance, on cold winter days when even the midday sun stands low at the horizon and the sky graduates from a deep blue to a very bright color, tinted with a trace of violet or orange at the horizon.

Strength - Controls the intensity of the filter effects. Again note, that both filters are connected directly to the atmospheric fog and so the filter effects will increase or weaken also with the density of the fog.

15.3.4 Clouds



Choose the **<Clouds>** tab in the atmosphere selection to edit the parameters for cloud formations.

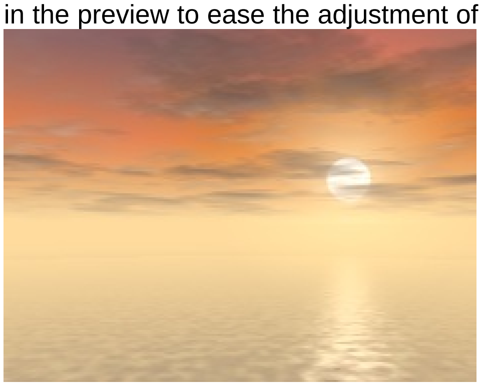
Clouds - If you select the <Clouds> button, clouds are simulated in addition to the sky model. Clouds can be influenced considerably with the parameters in the dialog. Of course you can simply load an existing file from the library and use it as a starting point to create your own cloud-filled skies.

Example: bright sky with pretty cumulus clouds (The settings shown above in the dialog picture were used for this picture).

Add Cloud Layer

A sky with clouds will contain at least one cloud layer. For complex cloud formations up to 3 additional cloud layers can be added. You can select the layer you want to edit in the "Layer" selector box at the top of the dialog. Next to the selector box are two buttons for adding new layers or deleting existing ones.

With the "Sky & Clouds" background mode an additional **<Preview only selected layer>** button appears beneath the preview window. If activated only the current selected cloud layer will be drawn in the preview to ease the adjustment of this layer.



Sunset with 2 cloud layers. A somewhat lower and therefore more-darkly shaded cloud layer and a second cloud layer high up in the sky still illuminated by the low sun.

Cloud Color and Brightness - In the **Cloud Color** box you can specify a basic color for the clouds. If a sun 351 light object is activated then the resulting cloud color will be calculated from this cloud color, the illumination of the sun light and the ambient cloud brightness (the last parameter in the upper box of the dialog).

Random - This initializes the random parameter used to generate the clouds prior to picture rendering. Each new value creates a completely new cloud field.

Height - The height of the cloud layer.

Accumulation - The smaller the value, the greater are the gaps between cloud formations.

Density and Transparency - For puffy cumulus clouds use a somewhat higher density value and no transparency while for thin cirrus clouds low density and high transparency values are recommend.

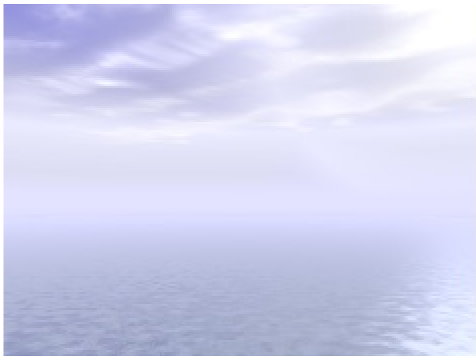
Turbulence - Turbulent flows influence the cloud formation

Crispness - A low value results in smooth rounded clouds. With a greater value the clouds become crispier and more detailed.

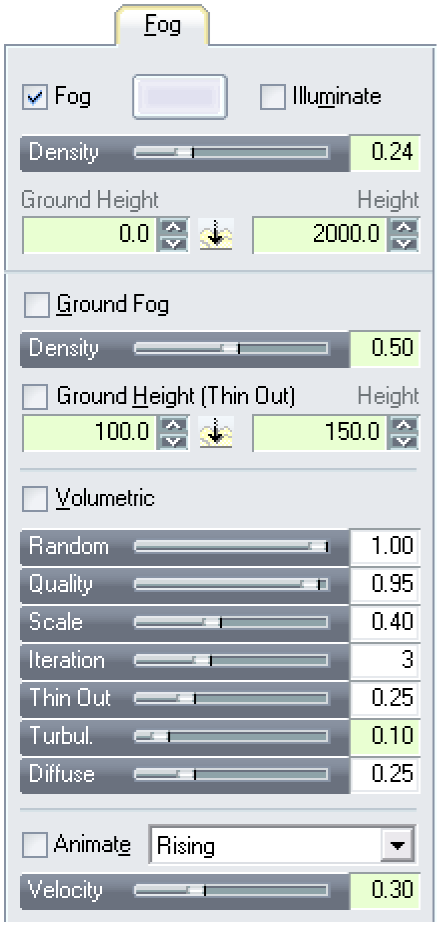
Volume and Contours - This two parameters control the 3D-effect caused by the illumination of the sun. The greater the **Volume** and **Contours** parameters the more clearly the cloud contours bulge out. For a thin cirrus cloud layer these values should be rather low.

Animate Clouds
A flow of the clouds can be switched on separately for each cloud layer. The angle instrument controls the **Wind Direction** and the **Velocity** parameter determines the speed of the cloud flow. If the **Turbulent** option is switched on then a turbulent flow with continuous changing cloud formations is calculated.
Basically all background parameters can be animated except for the Random and Accumulation parameters that define the basic random field for the cloud formation. For instance, you can stipulate in one keyframe a very low density together with high transparency and then in a following keyframe a higher density with less transparency. In the final animation a cloud bank would appear virtually out of nowhere. Similarly, you can define different color ranges at different keyframe positions to merge, for instance, a golden sunrise background into a bright blue sky.

Condensation Trails

I addition to CO_2, the combustion engine of an aeroplane also exhausts ordinary steam, which condenses to form a trail behind the aircraft.
In MR-3D Designer condensation trails can be build from normal cloud layers that are overlaid with an additional stripe mask. The position, the width and the length of the trail can be specified with the corresponding parameters at the bottom of the clouds page. The orientation of the trail is controled by the wind direction (The trail lies at right angles to the wind direction so that it floates with the wind if the <Animate>-option is switched on). With all those parameters you can freely arrange several condensation trails in different heights and different orientations in the atmosphere. As mentioned above, condensation trails are build from normal cloud layers, so all cloud parameters also affect the appearance of condensation trails. If, for instance, the accumulation or density parameters are too low, then often gaps appear in the course of the trail. On the other hand, depending on the influence of the weather, condensation trails may begin to dissolve or to mix with other air currents, so you can apply this effect intentionally.

The cirrus-rolls of this marvelous cloud formation consist of only 4 condensation trails positioned one after the other.

15.3.5 Fog



The fog-parameters of the atmospheric background model

Don't be startled by the variety of parameters presented for the fog functions. Only the few parameters in the top box are necessary if you just want to use "normal" atmospheric fog . The remainder relates to ground fog only, which, because of its cloud-like attributes, also has a corresponding choice of adjustments. Atmospheric fog and ground fog can be applied separately or in combination.

Atmospheric Fog

Fog - Switch the <Fog> button on to include an atmospheric fog effect in your scene. Atmospheric stands for a veil of mist and fog that increases exponential with the distance and, starting from ground level becomes thinner with increasing height, as with real atmosphere. The atmospheric fog effect - together with the atmospheric color filter functions - is indispensable for all realistic outdoor scenes because only with it does the picture have a real impression of depth and distance and smooth transitions between sky and horizon.

Fog Color - Click on the corresponding color button to edit the fog color. You can enter a simple gray or white or any color you like, e.g. a bright orange tone for sunset effects.

Illuminate - If this option is switched on an appropriate fog intensity is automatically calculated from the fog color and the light settings:

- **Atmospheric Fog** - Only the radiance from sun 351 lights is taken into account.

- **Ground Fog** - With volumetric rendered, the ground fog will be illuminated by all types of light sources, e.g. a lamp casting a visible light halo or a spot penetrating the mist with a visible light cone. Do not confuse this up with the simple light effects 365 or the visible spot cone 273 effect. If volumetric ground fog is activated in combination with the <Illuminate> option, then a real shading of the fog media is calculated.

Density - Controls the density of the fog.

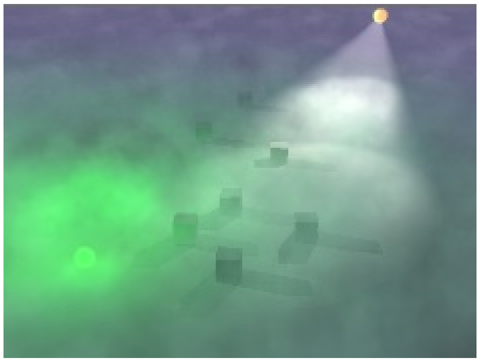
Ground Height - The maximum density of the fog occurs at ground height and decreases with increasing height. If the ground level of your plane 186 object not zero, you can enter the corresponding value of the y-position of the plane object here. The density of the fog will have a constant maximum density below this ground height. If you click on the button the ground height will be automatically set to the lowest object height in your scene.
Height - Above ground height the fog density decreases until the maximum fog height is reached. You can use the fog height parameter very effective for mountain peaks, where the massif vanishes in the distance but the peak remains clearly visible above the fog bank. But the maximum height of the fog is also relevant for the transition of the clouds and the fog at horizon level. If you want to render outdoor scenes on clear and sunny days with clouds of a high visibility a low maximum fog height is appropriate, whereas on foggy days with barely visible cloud formations a corresponding higher value for the fog height ought to be entered.

Ground Fog

The atmospheric veil of mist and fog reaching high up in the sky is indispensable for the rendering of realistic atmospheres, but we still need an additional layer of fog to simulate heavy fog limited to the ground area. The basic parameters for atmospheric fog and ground fog are much the same, with only one exception: The decrease of the fog density with increasing height is optional. If the <Ground Height (Thin Out)> button is switched off, the density of ground fog is constant and independent from the height. This is the right setting for low and dense fog banks on the ground. But if you want hills and peaks slowly emerging from a dense ground fog, then switch on again the <Ground Height (Thin Out)> button to thin out the fog with increasing height.

Example of volumetric ground fog above a mountain range. You can download the corresponding animated demo file from the internet library .

Volumetric Ground Fog - If you switch on the <Volumetric> button, the ground fog layer will be rendered volumetrically. This is done by tracing the viewing ray 264 through the foggy media and taking many samples of fog densities along its path. If the <Illuminate> option is also activated, then together with the density calculations at every sample point a shading routine will be carried out to determine the illumination at that spot. This is, of course, very time consuming but also very effective. Volumetric fog is based on similar routines to cloud formations and accordingly a lot of parameters are presented to control the appearance of ground fog, starting from a uniform media up to fluffy fog banks. Finally, you can even animate volumetric fog to let it rise from the ground and flow with the wind direction.



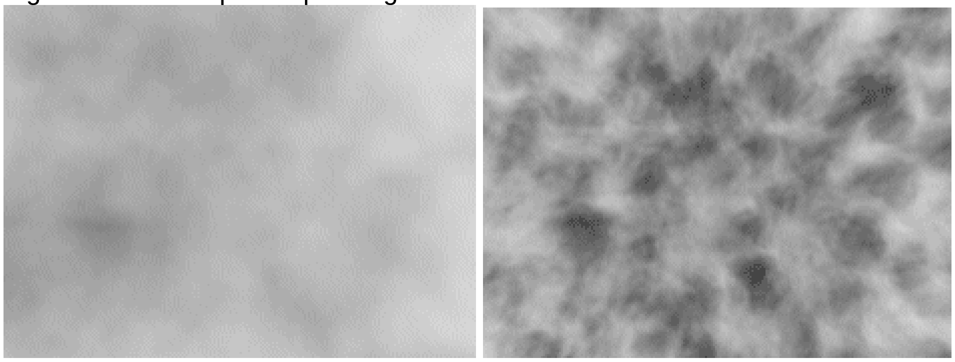
The picture above shows the shading capabilities of volumetric fog. A spotlight above the fog bank and a green lamp light directly in the fog illuminate the surrounding fog media. You can find the original demo file in "...projects/volumetricfog/fog_illumination.cmo".

The parameters:

Random - This initializes the random parameter used to generate the fluffy fog formations. Each new value creates a completely new fog field.

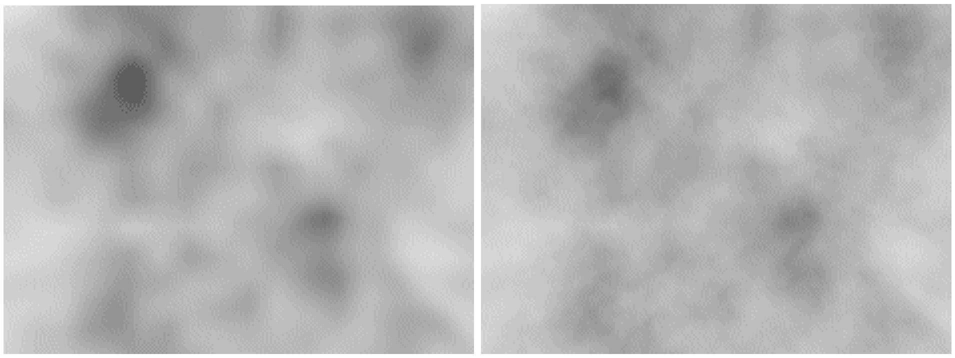
Quality - While the viewing ray is traced through the scene, many samples of fog densities and illumination values are calculated along its path through the fog media. The quality parameter lets you determine the intervals at which new samples are taken. With a higher quality value the distance between steps becomes shorter and the number of fog densities to be calculated for the general density estimation increases considerably. If fog illumination has also been switched on, the rendering time gets even higher. Therefore, it is advisable to set a very low quality (0.5 to 0.85) for preview renderings while working on the project and only for the final pictures set a relative high quality of about 0.95 to 1.00.

Scale - The underlying fractal 313 patterns used in a volumetric fog calculation are rendered close together or wide apart depending on the scale value entered.



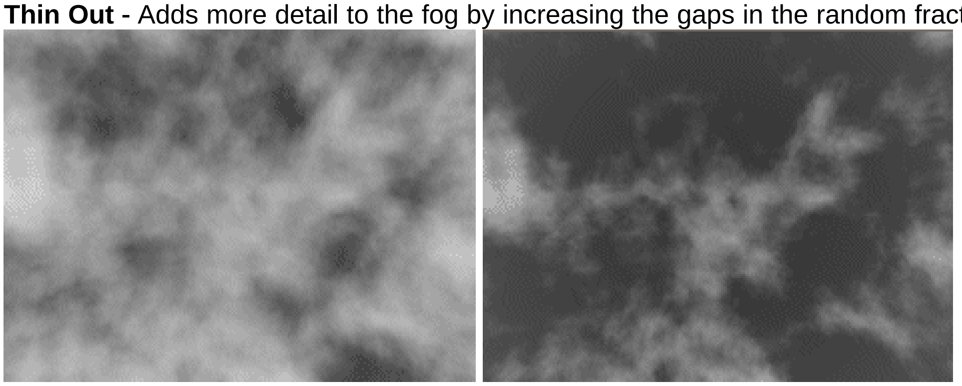
Two pictures shot from a position vertically above the ground fog with only a simple black plane object below to contrast with the fog. The picture on the left was generated with a small scale value which results in a smooth random pattern stretched wide apart. With a higher value for the scale function, the random pattern is rendered closer together with much more detail and frequent gaps in the pattern.

Iteration - The number of iterations defines the level of detail for the underlying fractal 313 pattern. A single iteration will result in a very blurred pattern. Further iterations will add new details to the fractal noise.



The picture on the left was rendered with only 2 iterations, whereas 4 iterations were used for the picture on the right. It might be a good idea to reduce the number of iterations for faster preview renderings, too.

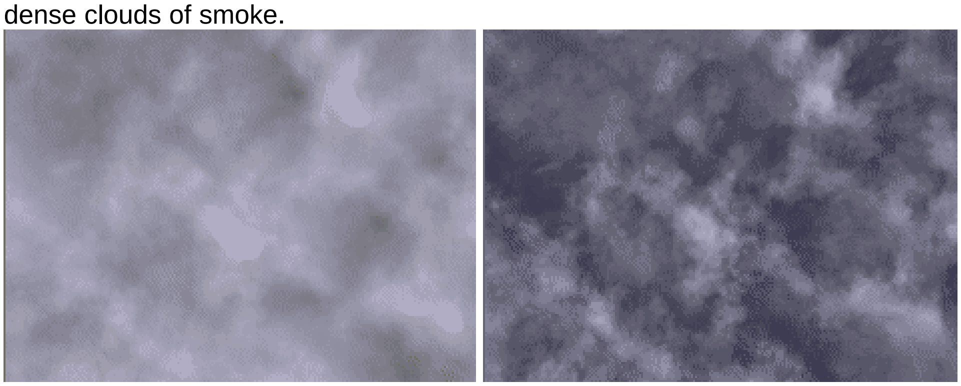
Thin Out - Adds more detail to the fog by increasing the gaps in the random fractal pattern.



The pictures above demonstrate this effect. On the left a fog bank rendered with a value of 0.5 and on the right side the same fog pattern cleared up further with a "Thin Out" value of 0.9.

Turbulence - This parameter increases the turbulent flow in the fog, especially when the fog is animated.

Diffuse - Modifies the shading calculations of illuminated fog, if the <Illuminate> option is activated. The higher the amount of diffuse reflection, the more the fog changes from a light hazy mist to dense clouds of smoke.



A ground fog pattern shown above on the left and clouds of smoke depicted in the right picture. With the exception of the diffuse parameter (0.20 for the left and 0.80 for the right picture) both images show the same fog pattern.

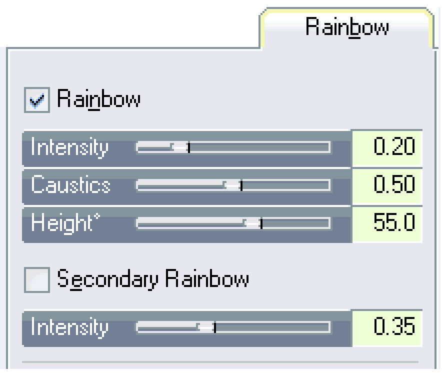
Animating Volumetric Fog

Fog patches drifting smoothly about a mountain pass or a marshland scene with vapour rising in the hot morning sun - just switch on the <Animate> button to render such scenes and to determine the direction of movement and velocity of the fog. You can select one of the following three movement types:

- **Rising** - The fog rises from the ground, vanishing as it rises.
- **Wind Direction** - The fog moves with the wind. The wind direction can be set on the clouds 382-side of the atmosphere selection.
- **Rising & Wind** - The fog rises from the ground and simultaneously moves with the wind.

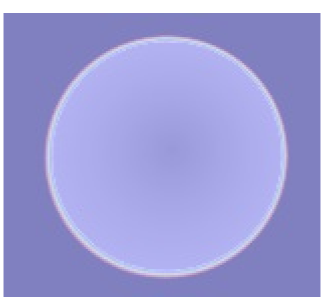
Velocity - Because we do not want the fog to chase across the ground with the speed of the wind, you can specify here a separate velocity for the fog movement.

15.3.6 Rainbow

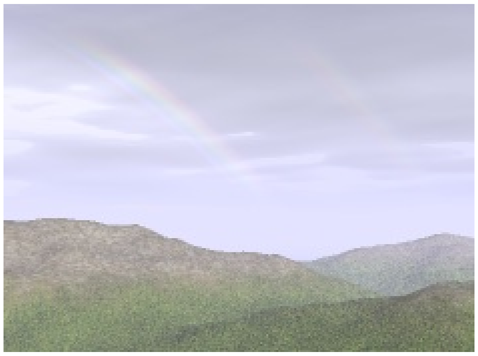


The rainbow - sun rays are refracted when entering and leaving a rain drop. Simultaneously a total reflection occurs at the rear of the drop. That's why the sun is always behind the observer and the rainbow appears as a circle around the straight line from the sun to the observer to the midpoint of the rainbow circle (a rainbow cone of 42° opens in front of the observer). Because of the reflected light the inner area of a rainbow appears brighter than the area outside of the rainbow (caustics). Additional reflections inside the rain drops may produce additional rainbows around the main bow but of rapidly-fading intensities.

This physical model is, for the most part, adopted in MR-3D Designer. If you switch on the rainbow effect then not just simple color circles are drawn in the background. Instead, the visibility of the rainbow depends on the position and point of view of the camera and the position of the sun. Therefore, if you want a rainbow to appear in the picture the sun has to shine from behind the camera. The best thing to do would be to set the camera zoom to a low value in order to locate the rainbow in the sky. Then, adjust the rainbow position by correcting the angle of light incidence for the sun and finally zoom in again with the camera.



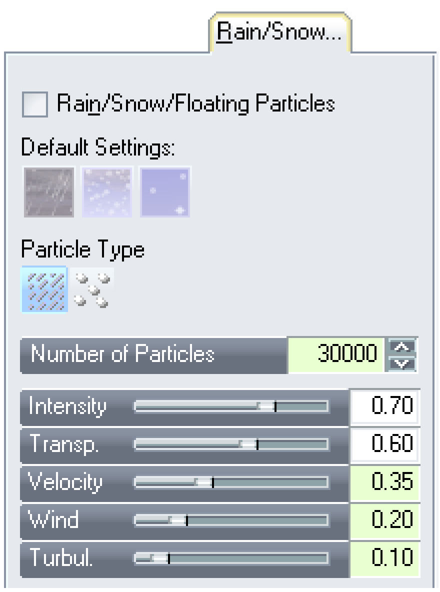
If you are lucky you can observe the full circular nature of a rainbow when looking out of an aircraft. Back on the ground the lower part of the rainbow will be hidden by the earth.



Example rendering with a bright main rainbow and a dim secondary rainbow. The caustics effect was switched on to lighten the inner area of the rainbow. Furthermore the height was limited, so that the rainbow is fading with increasing height. Seeing a complete arc depends, among other things, on the height and extent of the rain clouds.

The parameters:
Intensity - controls the intensity of the main- and secondary rainbow colors. Since the intensity of the secondary rainbow is always falling off against the intensity of the main rainbow, it is only calculated as a fraction of the intensity of the main rainbow.
Caustics - defines the brightness of the caustic reflections within the rainbow.
Height° - limits the height of the rainbow in dependence on the maximum opening angle of the rainbow (main bow 42°, secondary bow 51°)

15.3.7 Rain and Snow Particles



On the "Rain/Snow..." page of the atmospheric background model you can switch on a particle effect to simulate rain (stripes) or snowflakes or, e.g., floating particles in the water. These weather particles are not rendered as "real" objects during the picture calculation but rather painted above the picture after the rendering in post processing mode. This is in contrast to the rendering of real 3d particles you can set up in the Parcticle Systems 425 dialog.
The big advantage of the atmospheric particle system is it's rendering speed (no interaction of particles with the scenery) and the convenience in setting up rain or snow simulations just by switching on a button and adjusting a few sliders.

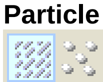
The illustration above shows some details from a small demo animation you can view in the 3d-designer.com internet gallery. Snowfall is setting in - covering the landscape with a pretty white coat. To achieve this effect material animation 300 of the terrain texture layers was also applied.

The parameters:

Initialize Standard Settings

 With these 3 buttons you can initialize the settings for the particle effect to one of 3 standard types. The first button retrieves the parameters for rainfall, the second button will set the parameters for snowfall and the last one sets the standard parameters for a floating particle effect.

Particle Types

 There are two basic types of particles - half transparent stripes for rain and a somewhat rounded shape for snow and floating particles.

Example for rain particles, rendered by Pascal Heußner.

Number of Particles - The number of generated particles for the weather effect. This number can also be zero. This is useful, if you want to start the effect at a later time in the animation. Example: You want the rain to set in at frameposition 100 with a few rain drops. On the following 100 frames up to frameposition 200 the rain becomes stronger. Then it rains with constant intensity up to frameposition 400 and finally the rain decreases again to zero raindrops on framepostion 500. To realize this animation you simply have to move to the corresponding frame positions and adjust the number of particles for the weather effect. Each time you change one of the parameters that can be animated in the background dialog (indicated by an emphasized background color of the edit field), a new parameter-keyframe is automatically generated for the animation. The settings could be as follows:

- Frame 1, Particles 0 - No rain, as yet.
- Frame 100, Particles 0 - The rain will begin to fall here. Since no changes of the parameters have been made on this frameposition, we have to manually add a new key in the animation editor.

Simply select the background object in the animation editor and add a parameter track and then operate <Add Key>.

- Frame 200, Particles 10000 - The particle stream increases from 0 particles in frame 100 to 10000 particles in frame 200.
- Frame 400, Particles 10000 - Between frameposition 200 and 400 a constant stream of 100000 particles is generated. Again, no change is made in the background dialog, so just add a key in the animation editor for the parameter track of the background object. This keyframe marks the point, from which the particle stream decreases again.
- Frame 500, Particles 0 - The rain stops.

Intensity and Transparency - This parameters control the intensity and transparency of the particles. Of course raindrops appear more transparent and less intense than shiny snow flakes. A high transparency value can also be useful for the rounded particle type, for instance, for floating underwater particles. Or think of somewhat hazy weather situation, where the snow flakes become blurred with the background.

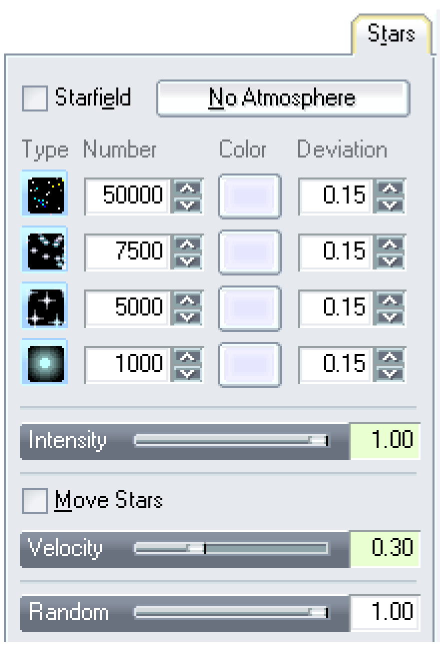
Velocity - controls the rate of fall for the particles.

Wind and Turbulence - The speed of the wind is combined with the rate of fall. The wind direction is given by the wind direction of the first cloud layer defined on the clouds page of the atmospheric background model. Turbulence adds some chaos to the particle stream which is especially useful for the snowfall.

Rendering of an Preview Animation

In contrast to the particle systems 425 based on real 3D-objects, the weather particles are rendered very quickly in post processing after the rendering of a picture. It is advisable, therefore, to switch off all unnecessary scenery objects, including clouds and other time-consuming effects, and to render complete preview animations only of the weather effect. Test and change all parameters accordingly, especially the number of particles, velocity, wind and turbulence, before you switch on again all scenery objects and effects for the final rendering.

15.3.8 Starfield



With the starfield generator it is possible to create a starry sky that you can even animate.

The starfield is a background model that can be combined with the other atmospheric effects. Thus it is possible, for example, to combine stars with a clouded sky. In this case the stars will be filtered by the clouds and fog and not simply drawn above the calculated cloud cover.

No Atmosphere

If you want to create a simple starry background for a scene in outer space, without atmospheric effects like clouds and fog, you can switch off all atmospheric effects at the same time by operating the <No Atmosphere> button. With it, the background color will also change to a simple black.

The Starfield

The starfield is a genuine 3D-starfield and not simple a 2D drawing that is put into the background. The stars will be scattered around the entire 3D-space (not all drawn in the picture area visible), so do not be surprised if you have entered 10,000 stars for a particular type of stars and you can only see a few hundred in the picture. The advantages of a 3D-starfield are evident:

·Real camera and zoom effects - The visible starfield moves along with the camera movements and you can both zoom in and out on the stars.

·The stars (being genuine 3D-objects with coordinates) can, of course, be animated.

Parameters

There are 4 basic types of stars you could create with the generator. Use the check box beside each star type image to switch this kind of star on or off. For each of the different types of stars you can choose the number of stars to generate, the basic color, and the color deviation that is randomly determined for each star.

The intensity parameter controls the transparency for the stars. This parameter can be animated 401, so you can produce a proper day (intensity 0) to night transition (intensity 1) where the stars become more intense as night falls.

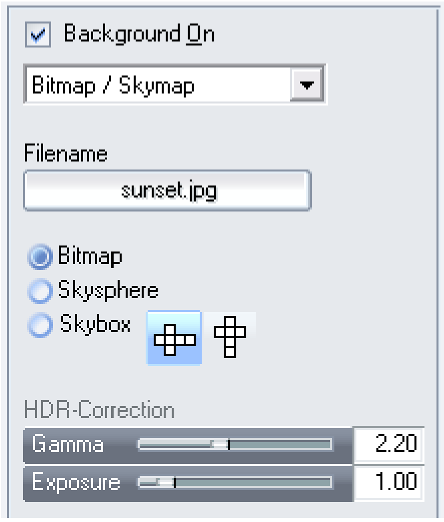
The "Random"-parameter at the bottom of the dialog initializes the random generator to produce different appearances of the starfield.

Moving Starfield

Use the option <Move Stars> if you want to animate the starfield for a film sequence. As a result the stars will move with a given speed, adjustable with the <velocity>-parameter, in the direction of the camera. This is similar to the effect you are already familiar with from corresponding screensavers. Here, however, camera movements and camera zoom are calculated in addition to the star movements. So, if the camera moves to the right, then the starfield will move to the left and vice versa. More distant stars will appear fainter than the nearer ones and will become brighter as they approach. The same applies to the size of the small suns (star type 4).

Finally switch on motion blur 409 and you will achieve the ultimate star-flight effect.

15.4 Background Bitmap, Sky- or Environment Maps



Choose the "Bitmap / Skymap" entry in the background select box if you want to copy a picture or an environment texture into the background. Operate the <Filename> button to display the file selector box and then select a bitmap suitable for your project.

- Background Bitmap 394
- Skymaps or Environment Maps 394
- Using Skymaps for Environmental Reflections 394
- Environment Maps and Ground Shadows 394
- Skymaps and Image Based Lighting (IBL) 394
- High Definition Range Images (HDRI) and Image Based Lighting 394
- HDR-Parameter - Gamma Correction and Exposure 394

Copying a Bitmap to the Background

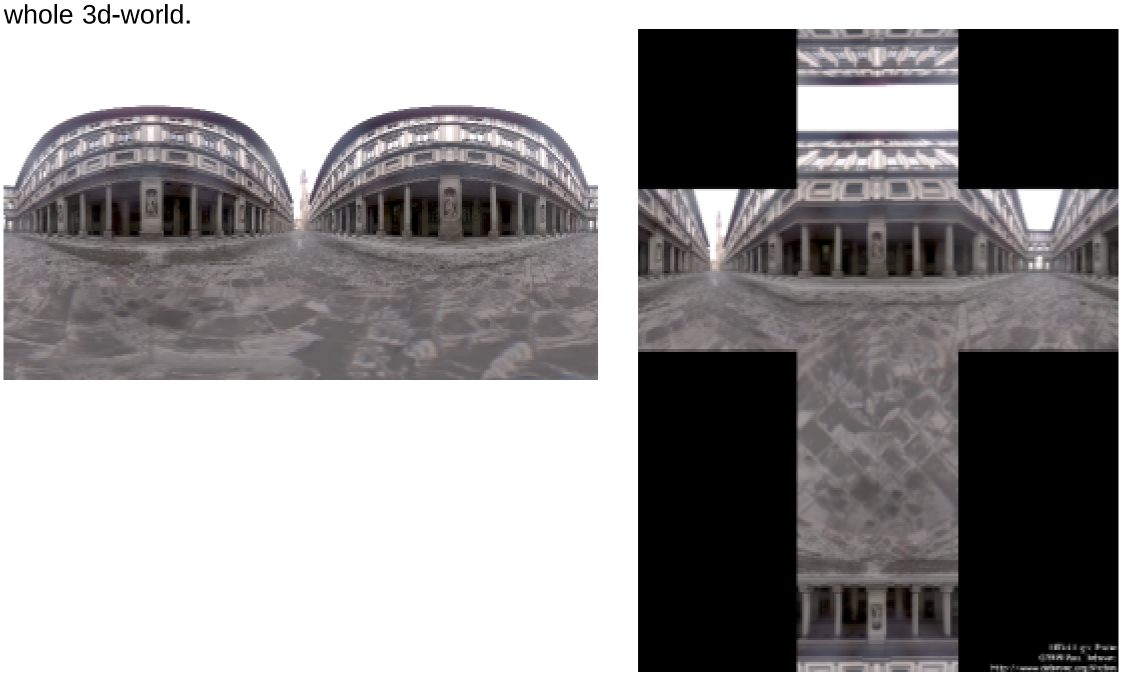
If you select the <Bitmap> projection mode then the chosen picture will be simply copied to the background of the rendered scene. The bitmap is automatically scaled to the size of the picture you are rendering. Of course, the best results are obtained when background and the rendered picture resolution 270 are the same or at least have the same proportion, otherwise the picture will appear stretched in one direction.

Using a background bitmap you can easily insert a 3d-model into a photographic picture. But the background bitmap is just a fixed picture copied behind the rendering of the scene - it is not a real part of the 3d-world. This means, you can't see reflections of the background picture on mirroring surfaces and the background will not move with the camera movements in an animation.

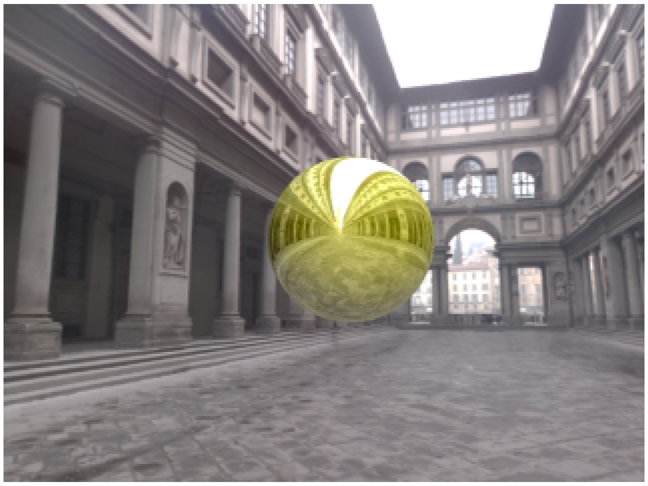
In this example a photo of a public park was copied to the background of the architectural model.

Skymaps or Environment Maps

A skymap (also known as environment map) is a panoramic bitmap providing a full range of view including ground and sky views. Depending on the given picture format (spherical projection or cross mapped) the bitmap is projected onto an imaginary skysphere or skybox surrounding the whole 3d-world.



Example of a skymap (source: Paul Debevec, www.debevec.org). On the left you see a spherical skymap used for skysphere projection mode, on the right the same environment produced on a cross mapped bitmap used for skybox projection mode.



This is how the skymap is represented in the render window when mapped to the surrounding sphere or box. In contrast to the simple bitmap background, a skymap is an adequate three dimensional object. The picture detail will change, when the camera alignment or zoom changes and the skymap is also reflected on all mirroring surfaces and can be seen correctly through refracting transparent objects.

Using Skymaps for Environmental Reflections

Skymaps are often used to provide environmental reflections on metalic surfaces. In the picture above on the left you can see a chrome-plated rim (model by Pascal Heußner), but without a surrounding environment no reflections are produced on the surface of the object. Now, instead of modeling a complex 3d-environment you can simply select an environment map for the background to achieve the desired effect (picture in the middle). By activating the object property <Invisible For Camera (visible in reflections) 290 > for the background object you can even hide the background while the reflections will still be rendered on the metallic surface (picture on the right).

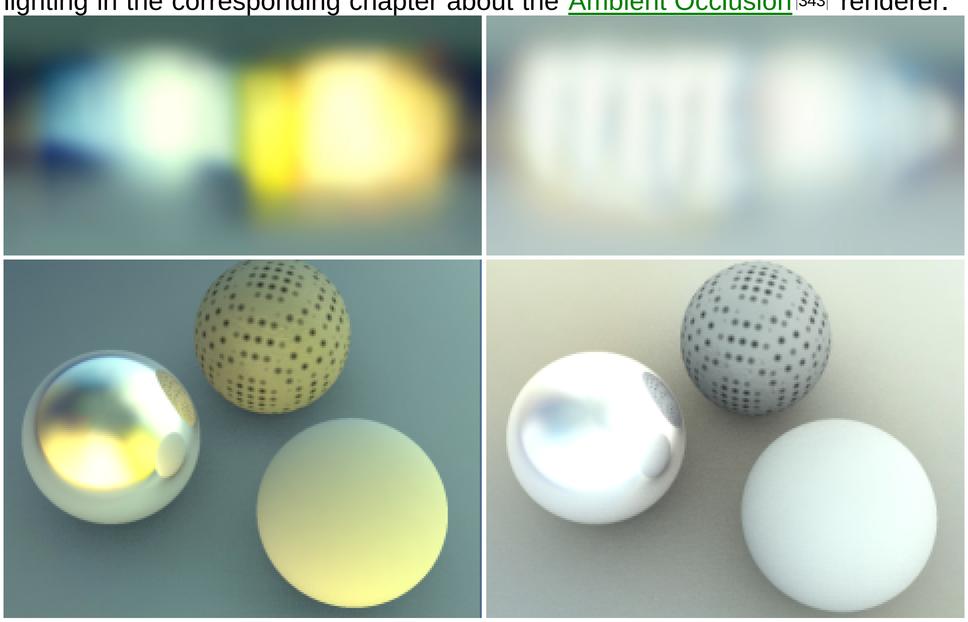
Environment Maps and Ground Shadows

Meanwhile there exist many professional sources of high quality sky- and environment maps that provide a full panoramic view including sky and underground texture. But still, the environment map is just a projection onto an infinite remotely sphere or cube, so you can't place anything on the visible ground of the texture. If, for example, you use a skymap of a street scene with a roadway visible underneath and you place a model of a car into the scene, then it would just hang in the middle of the environment projection. Since there is no real underground the car can not cast a shadow to the road and the scene appears more than a cheap montage instead of a fully integrated scene. Now, if you place a plane object under the car, then this plane object will receive the shadow, but it will also hide the underground texture of the skymap. The final solution is to combine the object properties <Compositing> (surface is drawn in pure material color without shading, but it receives shadows) together with the object property <Material Color = Scene Background>. If you activate these properties for the plane object then it will be rendered with a projection of the background texture (making it invisible in the scene) but additionally it will receive now the shadows cast by the model car placed onto it. Now the 3d model is fully integrated into the photographic scene.

Example: A model of a car (model: mr-clipart.com) was placed onto a plane object and a photographic environment texture (HDR-Image by Paul Debevec, www.debevec.org) chosen for the background model. The final rendering on the right shows the plane textured in the colors of the environment map projection - thus becoming invisible - but still receiving the shadows cast by the car model. In this example also the reflections and actually the whole illumination information was provided by the environment texture. As a result you can hardly tell the car is a 3D-model inserted in the picture and not a real part of the photographic scene.

Skymaps and Image Based Lighting

Skymaps provide the information for the background texturing, but they can also be used as a kind of light map for the ambient occlusion renderer.
To evaluate the ambient light incidence on a given surface point, now the hemispherical projection of the environment map above this surface point will be scanned and the pixel intensities of the skymap are interpreted as incoming light intensity. To accelerate this process, MR-3D Designer internally generates a strongly blurred copy of the original skymap before rendering. This will provide good results even with low sample rates. Read more about the application of image based lighting in the corresponding chapter about the Ambient Occlusion 343 renderer.



Two pictures of the same scene (3 spheres on a plane) but with different skymaps used to produce the reflections and the illumination in the rendering. This example shows how easy it is to transfer the light atmosphere from the picture to the virtual 3d-scene. Image based lighting is the number one key, if you want to perfectly integrate a virtual 3d-scene into a photographic environment.

High Definition Range Images (HDRI)

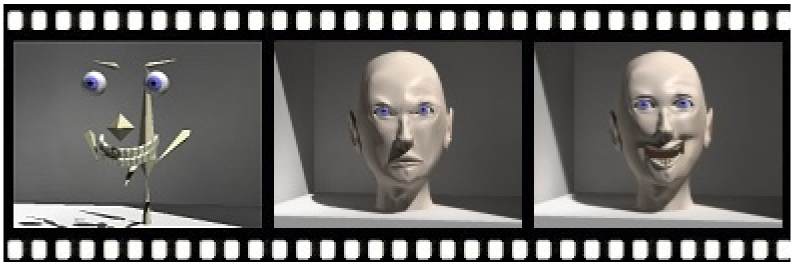
Next to common rgb-picture formats 328 supported by MR-3D Designer, you can also choose high definition range pictures in the Radiance rgbe-encoded .hdr format for the background projections. High definition range images are coded in a special floating point format, thus providing a much higher dynamic range than ordinary rgb-pictures. Think, for example, of a picture of a room. A white door is exposed to bright sun light shining through a window. Both areas of the picture, door and window, appear white on the monitor, but internally, in the hdr-definition of the picture, the bright sky visible through the window provides many times higher energy than the white door. If you would reduce the intensity in the picture (which is possible with hdr-images), the room would become darker and the formerly white sky would change to a more blue appearance, revealing perhaps some details of thin feathery clouds or condensation trails in the sky.
HDR-images have clear advantages when used as environment maps or for the image based lighting. Reflections in the scene as well as the ambient lighting generated by the ambient occlusion renderer are much more authentic and can provide much higher contrasts in the picture than renderings with ordinary rgb-pictures would do.

HDR-Parameter - Gamma Correction and Exposure

There are two parameters available if you select a picture in the Radiance rgbe-encoded .hdr format, with whitch you can control the brightness of the picture. Usually, digital images are provided with a certain amount of gamma correction (non-linear operation to encode luminance values) to adapt to the needs of a monitor, otherwise they would appear to dark on a monitor display. A HDR-image usually does not provide any gamma correction, since here real luminance values are provided for the illumination of a virtual 3d-scene. Therefore a gamma correction has to be applied to the picture before copying it to the background (whereas the linear information of the picture is used for the illumination). A gamma of 2.2 is the standard on PC video.
With Exposure you can control the general brightness of the picture.

Part XVI

16 Animation



Learn the basics to produce your own 3D-films.

See also:

- Tutorial - Simple Animation 55

- Tutorial - Animation and Object Hierarchies - Assembling a Robot 62
- Arranging Objects in Hierarchies 140
- Moving Objects and Movement Paths 197
- Rotating Objects and Movement Paths 211
- Scaling Objects in Relative Mode 206
- Tutorial - Animation and Deformation - Dolphin Movements 93
- Field Rendering for TV-Output 270
- Examples for Particle Animation 125

16.1 Animation - Introduction and Animation Button-Strip

The picture shows some details from a little demo film made with MR 3D-Designer. You can see this and other examples in our internet gallery at www.3d-designer.com.

Introduction

Computer animation is a wide field that is constantly developing. While only a few years ago an animation film was easily recognizable as such, because of the awkward movements of the animated characters and the always plastic-like appearance of the materials, nowadays its often difficult to tell if a scene takes place in a real environment, a handmade model or is even a pure computer creation - especially as these options are often combined to get the best results. Admittedly, hundreds of people are involved for years in the production of a professional animation film and the costs are exploding to many millions of dollars.
Nevertheless, I think that with MR-3D Designer you can also produce outstanding animations, since the program offers a wide range of possibilities combined with an intuitive interface that will allow even beginners to master their first steps into animation films successfully.
For instance, simple animations can be set up with a few mouse actions, moving or rotating objects to their destinations at different time positions, while the program automatically records the keypositions and interpolates the steps between these keyframes. Furthermore, almost all parameters can be animated - for instance the settings for lights, materials or the background - just by moving to the corresponding position in time and changing a parameter. With these simple parameter animations you can bring life to volumetric fires, running water, moving clouds and volumetric fog or add rain and snow to your animations.
Then there are the possibilities for hierarchical object animation - child objects inherit movements from their parents and follow these movements automatically while still retaining their freedom of movement, and therefore can still execute additional movements independent of their parent objects. Take, for example, a moving robot, that takes all his subordinated arms and joints along with its movement - while at the same time the joints still perform further rotations about their different joints, or gripping tongs open and close again. Hierarchical object animation is also the basis for the animation of characters.
The skin and bones technique uses a hierarchy of bones - a skeleton - that is subordinated to a corresponding skin object enveloping the skeleton. Moving the bones will deform the skin and so the character comes to life. For the animation of the skeleton often Inverse- or Forward Kinematics is applied, a technique that allows you too pull at the joints of an object hierarchy in the same way as if you would pull at the arms of a jointed doll. Today, most film and game productions use Motion Capturing to animate characters - an expensive technique that records the movements of real actors and transfers them on to the skeletons of virtual computer characters. However, very successful

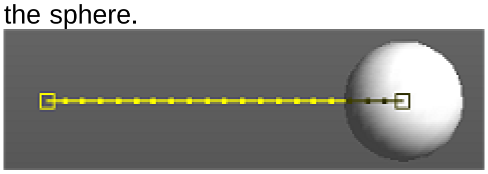
films have already been produced using nothing more than simple Forward Kinematics. Forward Kinematics appears not to be as comfortable as Inverse Kinematics on the first glance, but it provides a better control of movements, especially if many joints are involved.
Another great advantage of MR-3D Designer's hierarchically animation system is the reusability of movement patterns. For instance, in the Select Objects dialog you can remove a skeleton from its skin (or remove it from the hierarchy, respectively) and switch off all other objects except the skeleton. Then you could save only the skeleton and its animation data via the option "Save Selected Group" into a special "skeleton"-library for a later reuse in other projects.

Changing from Modeling Mode to Animation Mode 

Work on a project is divided into Modeling Mode and Animation Mode. There are two prominent buttons at the top left corner of the MR-3D Designer window to switch between the two modes. In principle there is no great difference between Modeling and Animation Mode - you have the same tool menus for both work-modes, except that some of the functions in Animation Mode are no longer accessible in Modeling Mode and vice versa. In Modeling Mode all changes made to an object - e.g. the deforming of an object by working on individual points - are permanent changes of the object's shape while in Animation Mode every action is merely a transformation of the model data and can be undone at anytime by reversing the working steps or deleting the keys that were created automatically when manipulating an object in Animation Mode.
Another Example: Scaling an object and its children in Modeling Mode will result in a permanent change of size of the model throughout the entire animation. If the children are deformed by this scaling this deformation will be a permanent change of the shape of the objects. If you scale an object and its children in Animation mode it will be only a temporary change of size. Moreover, the children will not be scaled at all - it is just their coordinate systems that are temporarily deformed by the scaling of their parents without influencing the children's object and animation data (see also: Hierarchy Independent Animation 140).
Besides these main differences the working process in Modeling and Animation Mode are very similar. In both work-modes you can move, scale or rotate objects, position and align the camera and start picture rendering at any time. Detailed descriptions about the differences and restrictions, depending on wether you are working in Model or Animation Mode, are provided in the corresponding chapters of the tool menus 195.

Animation Tracks and Keyframes - A First Simple Animation

MR-3D Designer enables animations to be implemented quickly, based on simple, intuitive procedures. Example: You have created a sphere that should move across the screen from left to right in 21 steps. First you have to change to Animation Mode. Thereupon the animation button-strip at the bottom of the screen will be enabled. With the animation button-strip you can readily move backwards and forwards within your animation. Now, in the "Move Object 197" menu you can move the sphere with the mouse to its initial starting position on the left side of the screen. This operation will automatically generate a position keyframe on a corresponding position track for this object. We will move forward in time now to frameposition 21. Simply press the  button (10 steps forward at a time) in the animation button-strip twice or input the frame number directly via the keyboard. Then position the sphere to the right at its final position. This will again generate a new position key for the sphere.



From the recorded keypositions in frame 1 and frame 21 the program can interpolate all the remaining frames lying between those keypositions by itself. In the illustration you can see the movement path of the sphere object - the large yellow squares indicate the positions in time where keyframes were generated and the smaller yellow points indicate the between-positions calculated by the program.

Now we are already finished with our little animation. Actually, you only had to move the sphere at two different times on the timeline, all the rest was done automatically in the background by the program. You did not even have to use the animation editor 409 to produce this simple animation. The tracks and keyframes, which were generated automatically in the animation editor, can be edited or you can add new tracks and keyframes, respectively.

This is how the position track for this little animation appears in the animation editor. The start and the end positions of the sphere are saved in the emphasized keyframepositions on frame 1 and frame 21.

Now its time for a first preview animation. To view the movement of the sphere directly in the active viewport window you can operate the  Play-button in the animation button-strip or, instead, select the  button to render a grid preview animation in the render window 259.

Track Types

You can set up keyframe scenes for each object in the scene - camera, background and light objects included - in which information such as position, alignment, size and parameter changes are held. The keyframes are created on separate tracks corresponding to the type of information recorded. The following track types can be added:

- *Position* - The position key holds the object position, of course.
- *Rotate* - The rotate key holds the alignment of the object axes system, a rotation axis and a rotation angle.
- *Scale* - On this track records changes in size.
- *Parameter* - This track saves parameter changes for background settings, light adjustments or the camera focus.
- *On/Off* - Objects and lights can be temporarily switched off and on again, respectively.
- *Deform* - This track saves the information for the animated deformation functions.
- *Material* - Also a parameter track, that keeps all changes of the material adjustments.

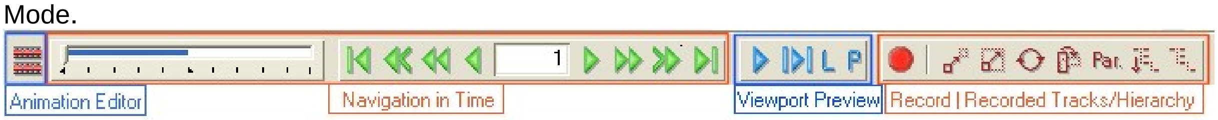
Corresponding tracks and keyframes are created automatically every time an object is manipulated, but you can also add tracks and keyframes manually in the animation editor.

Animation Editor and Animation Button-Strip

In theory you can set up complete animations in MR-3D Designer without the animation editor 409. With the help of the animation button-strip 401 you can easily move forwards and back. If you go beyond the previously existing total number of frames, the animation will automatically extend to the additional frames. Furthermore, changes to the settings of objects, the camera, the background or the lights are immediately noted and, as a result, corresponding tracks and keyframes for the relevant objects are automatically inserted into the timeline.
In the animation editor you can, however, easily edit the individual timelines of the objects. There you can add or remove tracks and keyframes, cut, copy and paste sequences or even transfer the movements of individual objects or whole hierarchies to other objects or hierarchies.

The Animation Button-Strip

The animation button-strip at the bottom of the screen will be enabled if you change to Animation Mode.



Calling Up the Animation Editor

In Animation Mode, this button on the left will be enabled and you can call up the animation editor to work on the individual timelines of the objects.

Navigation in Time

You can readily move backwards and forwards within your animation by means of the slider and the green navigation buttons. The range of the slider is limited by the current length of the animation. To extend the animation range simply use the green jump buttons to go beyond the current length or enter a frame number directly in the frame field located between the green buttons. You can also use your mouse wheel in order to move backwards or forwards in the animation. In this case you have to activate the slider box first with a mouse click, otherwise the viewport's zoom function will be executed instead.
If the movement path of a selected reference object is drawn into a viewport, then you can also klick onto the keypositions of that movement path to move directly to that position in time.

The meaning of the navigation buttons:

Jump to the start or end of the animation.

Jump to the previous or following keyframe of a selected object. This function is evaluated only for those tracks that are selected on the right end of the button-strip. If, for instance, only the rotation track is selected there, then you would only jump between existing rotate-keyframepositions.

Jump 10 frames forward or back.

Move one frame forward or back.

Viewport Preview

Next to the navigation buttons are found blue buttons for playing the animation in any of the viewport windows to get a quick overview of the movement paths of objects, lights and camera. (To start an animation rendering as a Preview or True Color animation viewed through the camera, however, you can operate the Render Scene Animation 259 - or Render Final Animation 259 button.)

Plays the Preview animation in the active viewport window. Press any (mouse) button to interrupt the animation. To play the Preview animation in a different viewport window, activate the corresponding window (mouse-click or over the "windows" menu-strip) and operate the Play button again.

Plays the animation within the designated range specified in the Animation-editor. Only the part of the animation between the start frame and end frame 409 is run through.

Loop function: If this button is activated the Preview animation will repeat, playing over and over from the start.

Ping-Pong function: The Preview animation runs alternating forward and then backwards again, until you interrupt the action.

Recording Keyframes Manually

With help of the record button, keyframes can be inseerted manually for all marked objects. Usually, new keyframes are automatically created every time you manipulate an object or when you change a parameter in Animation Mode, but sometimes keys have also to be created at timeline positions where no object manipulation is intended - for instance, to record the alignment and position of an object at a particular moment in time as a starting position for a planned movement from this timeline position.

Creating Additional Keyframes for Selected Tracks and in Object-Hierarchies

When the record-button is operated keyframes will be created for all marked objects, but only on those tracks you have previously selected via the five track buttons in the animation button-strip next to the record-button. This applies also for the automatic creation of keyframes. Every time you manipulate an object a keyframe is generated on all tracks that are activated here.
Example: If you move an object from one position to another, usually only a position key is created. But if the rotate track is activated, then a position-keyframe is generated automatically because of the movement of the object and, additionally, a rotate keyframe because the rotate-track was selected in the animation button-strip. Always generating both position and rotate keyframes ensures a fixed position and alignment of objects in time.
The meaning of the individual track- and hierarchy-buttons:

 - Position keyframes are included in keyframe creation.

 - Scale keyframes are recorded

 - Rotate keyframes are recorded

 - Deform keyframes are recorded. This relates to the deformation functions [251], not the skeletal deformation. Using the skin and bones technique will automatically deform objects when moving and rotating the bones.

 - Parameter keyframes are recorded. This track-button serves mainly to include the camera zoom when recording camera positions and alignment. Parameter changes of light or background settings are only recorded when changing the parameters in the corresponding dialogs.

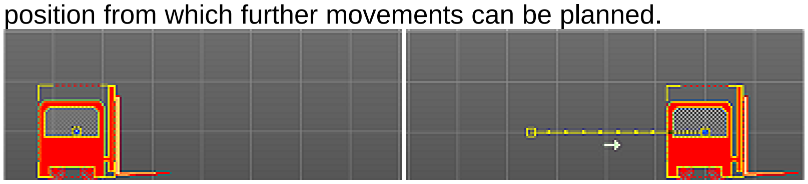
Creating Keyframes in Object-Hierarchies

If you have marked an object in an hierarchy, then you can decide with help of the last two buttons in the animation button-strip, wether you want to include key generation also for all children of the selected object or even for the whole hierarchy.

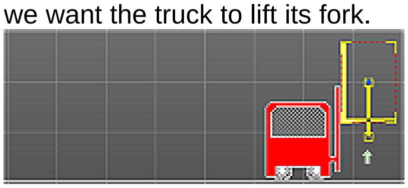
 - Keyframes are recorded for all children in a selected hierarchy-branch.

 - Keyframes are generated for the whole hierarchy - for the children as well as for the parent objects.

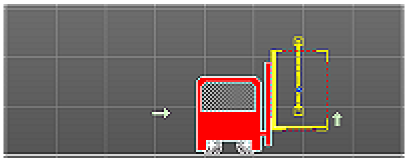
You may ask, since child-objects always follow their parent's movements anyway, why should I waste valuable memory space to generate additional keyframes for all the objects in the selected hierarchy? Well, its just to freeze the complete hierarchy at a particular moment in time as a starting position from which further movements can be planned.



Example: The forklift in the illustration above is build from a simple red truck and a fork-object subordinated to the truck-hierarchy. The forklift is moving in 10 steps from its starting position on the left to a destination position on the right (frame 1 to frame 11). After reaching the destination point we want the truck to lift its fork.



Consequently, we move again 10 steps forward in time to frameposition 21 and move the fork upwards to its raised position. What happens now, if we play a preview animation?

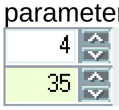


Instead of the truck driving from the left to the right and afterwards lifting the fork, the fork will start to move upwards from the beginning of the animation. This is due to the fact that only one keyframe for the fork was generated on the endframe 21, when we lifted the fork to its raised position. By moving the truck on frameposition 11 to the right, a position key was generated for the truck, but not for the fork, since the subordinated fork automatically follows the movement of its parent truck.
In this case it would have been of advantage to switch on the creation of keyframes for all objects in the hierarchy before moving the truck. Then a keyframe would have been created also for the fork on frameposition 11 and the fork would not start to lift before frame 11 is reached.

Character animation is another example where the automatic key creation for the complete hierarchy tree as well as for the position and rotate tracks is useful, so that the orientation and position of each individual limb of a character is fixed in each keyframeposition on the timeline.

Animation Options for Different Object Types

Which types of objects and parameters can be animated? In addition to normal objects and the camera, you can produce keyframes for a variety of settings for the illumination, backgrounds, clouds, fog, water and fire. To keep track of all values that can be animated, the edit boxes of those parameters are displayed in a different background color.



The illustration shows two parameters, the upper one is a fixed value that is valid for the whole animation. The second parameter, which has a light orange background color, automatically creates a new parameter keyframe on a corresponding parameter track when the current value is changed.

Objects

In each frame, objects can be given a new position 197, and can be scaled 206, rotated 211 or deformed 251. A new keyframe is automatically generated for each change in the current object data on a corresponding animation track in the animation editor, or, if a keyframe already exists, it will updated, respectively. Furthermore, objects can be arranged in hierarchies 140. Hierarchical structures are essential for animating complex movements. If an object is hierarchically subordinate to another object, then it performs every action of the parent object, but can also execute movements of its own, independent of the parent's movements. In this manner, complex movements for hierarchically linked object groups can be set up, as, for example, are required for the animation of robots or characters.
See also:

- Arranging Objects in Hierarchies 140
- Tutorial - Animation and Object Hierarchies - Robot 62
- Edit Skin and Bones 242 and Tutorial - Character Animation 97

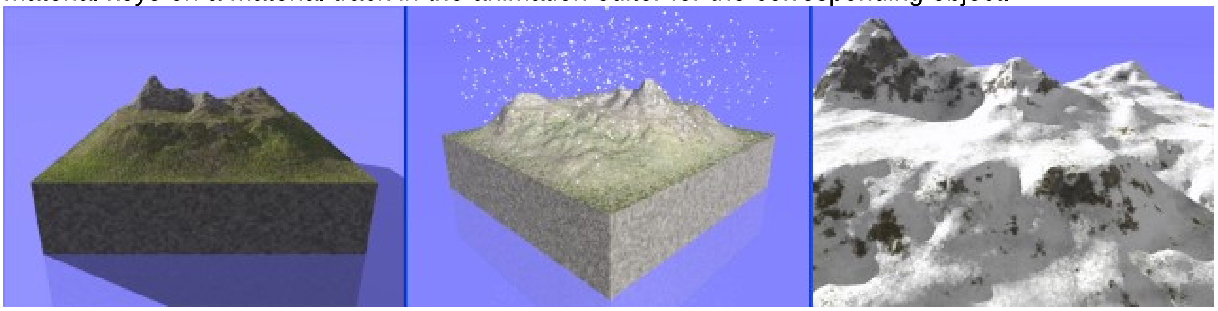
Hierarchy-Independent Animation

All objects save their animation data in an individual coordinate system belonging only to this particular object. This animation data is entirely independent of the world space and also independent of all movements, rotations or the changes of size inherited from other parent objects. Therefore, although we speak of hierarchical animation, the animation data of each individual object is recorded hierarchically independently. This is an important difference, because it enables you to insert (or remove) previously animated objects into existing hierarchies, while still keeping their own movement pattern, and, at the same time, following the movements of their new hierarchy parents. What happens for instance, if in an animation you resize a hierarchical group of objects and then, afterwards, you link a newly created object under this hierarchy? The new object will inherit all

animated changes of size from its new parents and will grow and shrink together with its parents in the course of the animation. But if you remove the object from the hierarchy once more it will immediately regain its original position, alignment and the size it had when it was first created and it will no longer follow the movements of the hierarchy it was previously linked to (if no keys have been generated for the object itself).
Another example: You have set up an animation for an aeroplane in which the plane is following a particular movement path. Separately from the aeroplane you constructed and animated a rotating propeller. Now, if you insert this previously-animated propeller in the aeroplane hierarchy the propeller will be automatically taken along with the aeroplane movements while continuing to rotate on its longitudinal axis.
The concept of hierarchical independent animation is also very useful for transferring animation data from one object or hierarchy to another. For instance, setting up movements for characters can be a very complex and time consuming task. Therefore it would be a great help, if you could save previously set-up movements for a later use in a library, or if you could copy them over to other objects or hierarchies, respectively. However, you don't want to have an exact copy of the movement, since this would result in all the characters starting together from the same position and moving uniformly in the same direction. This is the great advantages of the hierarchical independent animation into effect. Because the animation data of each object is saved hierarchically independent in a local coordinate system, the relative movement patterns 409 of objects can be copied to other objects instead of absolute coordinates and angles.
Suppose, for instance, you have animated a character in a short walking sequence. You want to create a second copy of this character walking in another direction. To achieve this you have only to copy the character - all animation data will be copied with the model data. Afterwards you simply need to move the copy of the character in "Move Object" mode together with its movement path to a new starting position. Then - in "Rotate Object" mode - you just rotate the figure, again with its movement path included in the rotation, so that it faces into a new direction. If you now play a preview animation you can see that the second figure really walks with the animation data copied from the first character from a new starting point and in a new direction.
The animation editor can also be used to copy relative movement patterns from one object to the other, or even from a whole hierarchy to another hierarchy. There, when copying a sequence from a position or rotate track to the buffer, you can decide whether this animation data is to be interpreted as absolute positions and angles or as a relative movement pattern.

Material

With MR-3D Designer you can animate transitions from any material mix into any other material mix, e.g., from glass to wood, from wood to stone or just to slowly fade in a bitmap label on top of other underlying materials. Setting up these effects is very simple. You just have to assign all materials 300 you want to include in the texture animation to the material list of the corresponding object. Then, on different keypositions, just go into the Material dialog 300 and switch on those materials 300 that you want to appear on the object's surface at that particular point in time and on the contrary switch off all remaining materials you don't want to include at that time position. If you render now the animation then a smooth transition will be calculated from the current set of activated materials to the set of mixed materials valid for the next key position.
Changing a material's on- or off-status at different positions in time will automatically create new material keys on a material track in the animation editor for the corresponding object.



Example for the combination of background and material animation. Clouds are gathering and snow

begins to fall, covering the landscape under a white blanket of snow. To render this animation only two materials have been added to the landscape object. Both materials have the same landscape texture 323 definition, the only difference is that for the second material an additional snow texture layer has been activated on the <Terrain> page of the Material dialog. Then, at the start of the animation material one (without snow) has been switched on while the snowy material has been switched off whereas at the end of the animation the status of both materials has been reversed again, the texture without snow has been switched off and the snowy texture has been switched on. That's all.
For a more detailed description and examples see - Material dialog - Animating Materials 300.

Light Objects

Lamps, spot lamps, area lights and fire objects can be animated in the same way as all the normal objects and all transformations are recorded on the respective position, rotate and scale tracks (some transformations will only displace a light object, for instance if a standard lamp is scaled in an hierarchy with other objects, it will only move with the deformation).
Parallel light-sources and spot lamps can be adjusted in alignment and in spread (spotlamp). Since parallel lights are not positional, all parameters including the angles for the light incidence will be recorded on the parameter track of the parallel light object.
The intensity of lights can be animated for all light-sources, so that they can gradually blend into different light-colors. Volumetric fire is burning anyway, so you just have to adjust the initial values. But of course size, position and alignment can be animated, too.
The light-effects 365 can also be animated. All settings for lens flares that involve the size of light halos - circular, star or spots - as well as the intensity of the light halo (color), light-circle and spots can be changed at each key. The angle of rotation for the star-shaped lens flares can also be animated. This way you can, for example, generate a radiance that rotates about a light source. Animation of size and intensity parameters enables wonderful light-explosions to be generated, for example.

Example: A meteor within a volumetric fire 358 -cylinder

Light sources can also be made hierarchically subordinate 140 to other objects, thus following their movements. This is important if you want to arrange a light source as a vehicle headlamp, for example - as you then only have to position the light source once. If the vehicle is then animated, the light-source automatically moves with the headlight object.

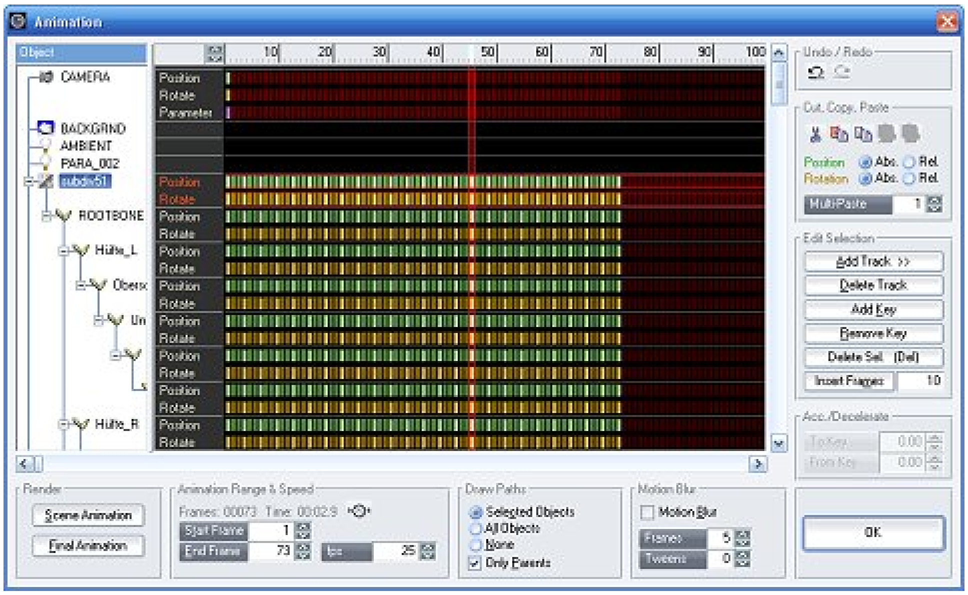
Camera

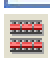
Camera position and alignment are recorded on the respective position and rotate tracks for the camera 257. The camera's zoom-values are saved to parameter keyframes on the camera's parameter track. The camera can also be made hierarchically subordinate 140 to other objects and thus follow the movements of the objects. For example, if you arrange a camera hierarchically under an aircraft, then, for the animation, the aircraft is moved, the camera automatically moves with the aircraft's movements.

Background

There are many options for animating the background. Almost all background parameters can be animated, for instance, in an atmosphere 378 background the color range for the sky can be animated as well as moving cloud layers and fog banks, shining rainbows, snowfall and rain and even the stars in the sky can be animated. Like with the light and material settings, changing a parameter in the background dialog will automatically create a keyframe on the parameter track of the background object.

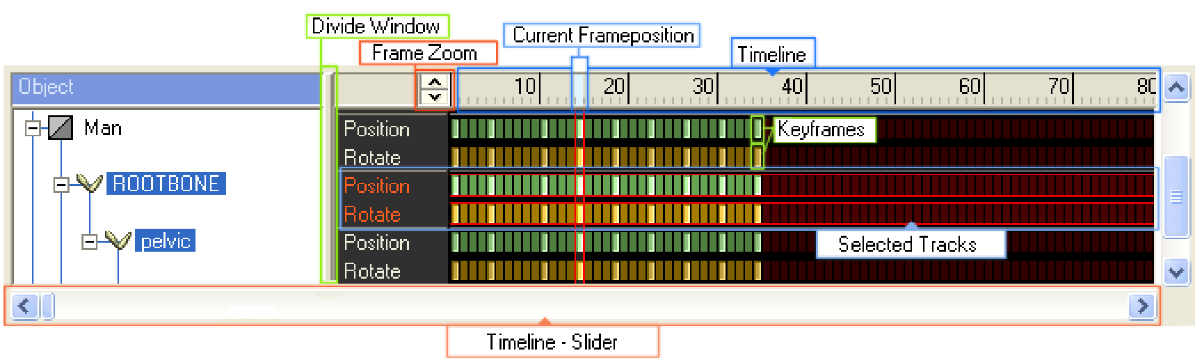
16.2 The Animation Editor



 - Menu "Option - Animation" - Short Cut:  + "A".

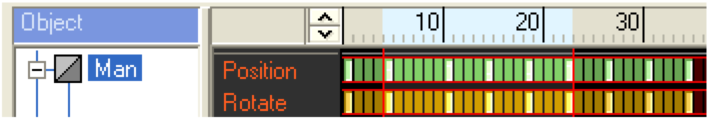
In the animation editor you can edit the timelines of individual objects and adjust the general animation settings.

The Timeline Window



The timeline window is split into the object list on the left side and the animation tracks containing the keyframes on the right. You can change the proportion of the split screen by moving the vertical bar located between the two areas.
The timeline is displayed above the animation tracks. The time is measured in frames, corresponding to the number of pictures rendered for the animation. When calling up the animation dialog, the current frameposition is highlighted in the timeline and an additionally vertical red frame

indicates the frame-column in the track window. (Selected tracks are also marked by a horizontal red frame). You can change the current frameposition with a simple mouse click on the timeline or one of the animation tracks. If you click on a frame that is behind the last frame of the animation, then the animation automatically extends the relevant number of frames.
In front of the timeline two small arrow buttons are provided to increase or decrease the visible frame zone of the timeline.
You can move backward and forward on the timeline using the scroll bar at the bottom of the timeline window.



You can also select a whole frame zone - to copy or relocate for example. Choose the first frame of the required area with the left mouse button and then, while also pressing the  -shift button, select the last frame of the required area. In the foregoing illustration the selected frame zone stretches from frame 5 to frame 23.
You can even mark the frame zone of the complete animation, from frame 1 to the last frame of the animation, by double clicking on a track name. If, for instance, you want to mark the complete position track, then double click on the "Position" text in front of the track.

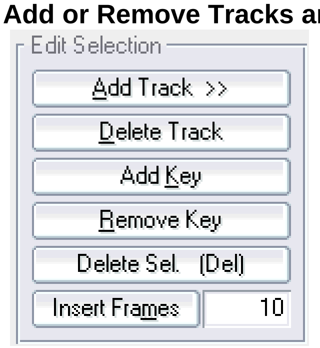
Selecting Objects and Tracks

Simply click on the object's name in the object list to select an object. All of the animation tracks belonging to the object will be automatically selected with it.
If you want to edit only a single track then just click on the track's name in the timeline window. Thereupon the track will be marked and also the respective object belonging to it.
To select (or deselect again) further objects or tracks, simultaneously press the <Ctrl> button on the keyboard while clicking. If you press the  -Shift button instead of the <Ctrl> button all objects (tracks) are selected or deselected that lie between the first and last selected objects (tracks).
Using the key combination <Ctrl> + <A> you can mark all objects and tracks at the same time.

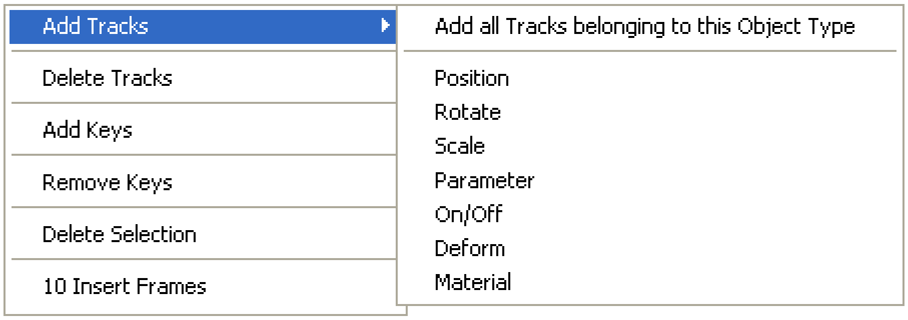
Undo and Redo

The last 50 operations in the animation dialog can be undone immediately by a separate Undo/Redo function in the animation dialog. The buttons are located at the top right corner in the dialog window. Of course, after leaving the dialog, all changes can be undone as a whole via the general Undo/Redo 28 functions in the main button bar.

Add or Remove Tracks and Keyframes



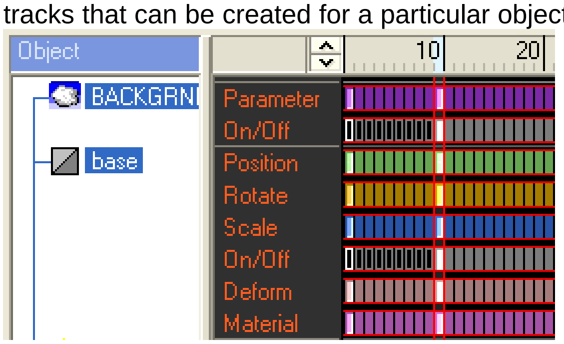
The functions to add new tracks and keyframes or to remove them again are provided in the "Edit Selection" box.



The same functions can be obtained from a selection list that opens when you click with the right mouse button in the timeline window.

Add Tracks

Every time when you manipulate objects in the viewports or when you change the parameters of an object a corresponding animation track is automatically created for the object in the animation editor. You can also add new tracks by hand, for instance, if you want to transfer an animation sequence from one object to another object for which no corresponding track has been created yet. When you operate the <Add Track> button then a list box opens from which you can choose one of several track types. Not all track types are suitable for each object type. For instance, the background object cannot be moved or rotated and therefore, no position tracks or rotate tracks can be added for the background.
Choose the "Add all Tracks belonging to this Object Type" entry from the list, if you want to add all tracks that can be created for a particular object type at the same time.



The illustration above shows all possible track types. For clarity, every track type is emphasized by a different color. The example above also shows that for the background object only two tracks are relevant - i.e. the parameter track and the On/Off track to switch off and on objects temporarily during the animation. For normal polygon-based objects all track types except the parameter track can be added. Parameter tracks are reserved for the settings of light sources, the background model and the camera zoom. But, if you would change a normal object into an area light source, then you could also add the parameter track for the area light object.

The track types:
For each object of the scene - camera, background and light objects included - you can set up keyframe scenes, in which information such as position, alignment, size and parameter changes are held. Corresponding to the type of information recorded, the keyframes are created on separate tracks. The following track types can be added:

- *Position* - The position track records position keyframes.
- *Rotate* - The rotate track records rotate keyframes which keep the alignment of the object axes system, a rotation axis and a rotation angle.
- *Scale* - On this track changes in size are recorded.
- *Parameter* - Parameter changes for background settings, light adjustments or the camera focus are saved on this track.
- *On/Off* - Objects and lights can be temporarily switched off and on again, respectively.
- *Deform* - This track saves the information for the animated deformation functions.

- *Material* - A special parameter track, that keeps all changes of the material adjustments.

Delete Tracks
Operate <Delete Track> to remove a selected track along with all its keyframes.

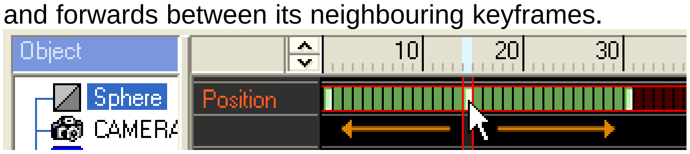
Add Key
You already know the record 401 button located in the animation button-strip 401 with which you can create keyframes for selected objects on the current frameposition. Similarly, here in the animation editor you can operate the <Add Key> button to add a keyframe to all previously selected tracks. The data saved to these keyframes will be calculated from the keyframes adjacent to the selected frame.

Remove Key
You can transform selected keyframes into normal frames by operating the <Remove Key> button. If an entire frame zone is selected all keyframes included in this area are transformed. No frames are deleted, however, and the animation therefore retains its previous length.

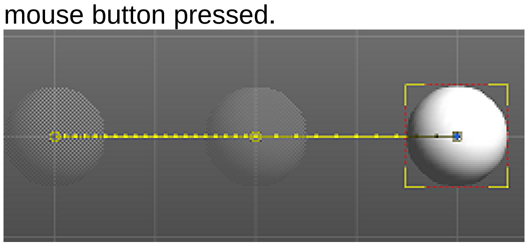
Delete Selection
If you have selected a frame zone you can completely delete it using the <Delete Selection> button. Keyframes as well as normal frames will be deleted and the remaining frames following on the selected frame zone will be moved to fill the gap.
If all objects and tracks are marked, then the whole animation is shortened by the selected number of frames.

Insert Frames
Operate the <Insert Frame> button if you want to insert additional frames at a particular point between two keyframes. Via the edit-field next to the <Insert Frame> button you can input the required number of frames that will be inserted with each click of the button.

Move Frame Range
The number of steps between two keyframes controls the speed of an object's movement. The relationship of the distances between adjacent keyframes therefore has to be adjusted quite often to ensure smooth movements. You can do this by moving a selected keyframe (or frame zone) back and forwards between its neighbouring keyframes.



Just click in the selected frame zone and move it to the left or to the right while holding the left mouse button pressed.

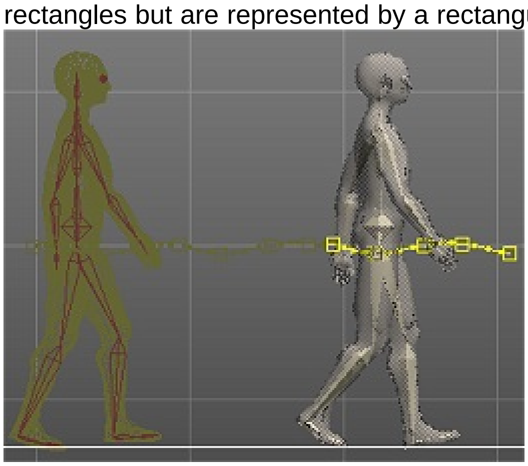


The example above shows a sphere moving in 20 steps from the left (Keyframe 1) to the center of the screen (Keyframe 2). Then, 10 further frames forward in animation time, the sphere is moved to its destination point on the right side of the screen. Now play an animation preview. The sphere moves slowly to the center of the screen and then doubles its speed for the rest of the course. Since only half of the movement steps are used for the second stage the sphere also needs only half of the time for the remaining distance.
In general - short movement steps result in slow movements and long distances between movement steps represent high-speed movements.

If, for instance, in the animation editor you were to move the second keyframe from frameposition 20 backwards to frameposition 15, then an equal number of movement steps would be used for the first stage from frame 1 to frame 15 and for the second stage from frame 15 to frame 30 - and the sphere would move with constant velocity over the complete course.

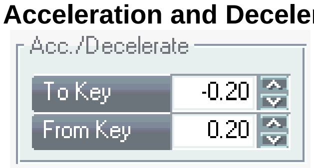
Switching Objects Off or On Again During an Animation

If you want to hide objects temporarily in the animation, then simply add an "On/Off" track. To hide an object at a particular point you simply need to insert a keyframe on the track using the <Add Key> button. To show the object again simply add another key at the required position on the track. The functionality is very easy - every newly-inserted key will reverse the visibility condition at the current frame position.
The illustration above shows the "On/Off" track of a light source. The light is switched off at frameposition 11 and switched on again at frameposition 21.
In the timeline the hidden periods are recognizable in that the relevant frames are no-longer filled rectangles but are represented by a rectangular framework only.



In the viewport windows objects on hidden frame positions are drawn only in simple grid mode and the grid color is dimmed somewhat. You can still select these objects and work with them, but you also recognize that the object is hidden in the final rendering of a picture on this frameposition.

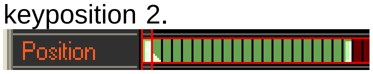
Acceleration and Deceleration



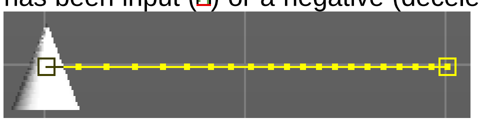
If you do not want jerky movements in your animation or the objects should not always move at a uniform speed, then you need some kind of acceleration or deceleration. The acceleration values can be input in the corresponding "Acc./Decelerate" parameter box.
For the movement track, you can also control these parameters directly with the mouse in the viewport windows by manipulating the path tangents (see: Move Object - Curve Interpolation - Acceleration 197).
Acceleration can be defined for all tracks except the visibility "On/Off" track. So you can accelerate not only movements but also rotations, changes of size or even the interpolation of parameters between keyframes. You can only edit the acceleration values of a single selected keyframe at a time. So, first select a single track of an object and then the respective keyframe. As a result, the current acceleration values for the selected keyframe will be displayed in the "To Key" and "From Key" edit fields and you can adjust the values.
You can input a value between -1 and +1:

+1 = maximum acceleration
 0 = no acceleration, which results in a uniform movement
-1 = maximum deceleration
The following examples all refer to position tracks, because from the movement path of an object you can easily recognize where an object is accelerated (movement steps get longer) or decelerated (distances get shorter).

The starting situation - a cone is moving with constant speed in 20 steps from keyposition 1 to keyposition 2.

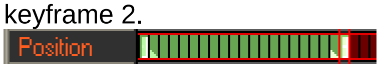


Now, on keyposition 1, a deceleration value of -1 is input for the "From Key" parameter, which results in a soft deceleration until the object comes to a halt in key 2. In the timeline you can now see a small triangle behind key 1. This triangle indicates that an acceleration value has been entered for the keyframe. If the triangle is located in front of a keyframe then a "To Key" acceleration value has been input which effects the velocity when approaching the keyframe. If the triangle is located behind the keyframe, then a "From Key" acceleration value effects the speed of an object directly after running through this keyposition.
Furthermore, you can recognize from the inclination of the triangle, if a positive acceleration value has been input () or a negative (deceleration) value ().



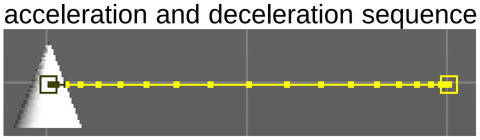
This is the movement path for the deceleration. The movement steps clearly become shorter in the course of the movement, but not uniformly. After about two third of the way the speed of the cone and with it the spaces between the individual movement steps become constant again. This is due to the fact that for the second key no acceleration value for the approaching stage has been input via the "To Key" parameter. If you also input for the second keyframe a value of -1 for the "To Key" parameter, then you get the following result:

Now the movement is continuously decelerated from the starting position to the end position in keyframe 2.

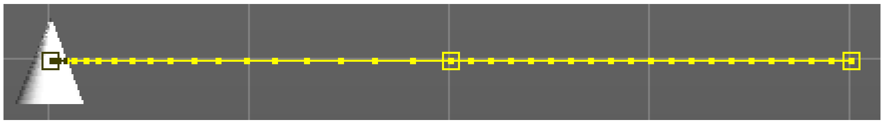


The two triangles in the timeline also show that deceleration values have been input after the first key and before the second key .

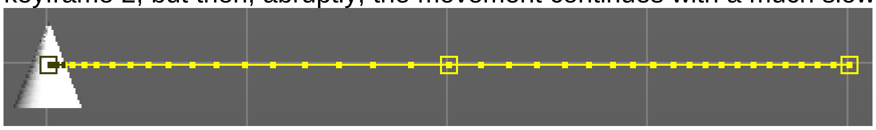
With the help of the two "From Key" and "To Key" parameters you can create a complete acceleration and deceleration sequence between two keyframes.



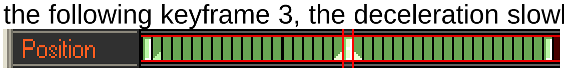
A value of +1 for the acceleration from the first keyframe and a deceleration value of -1 for the phase approaching the second keyframe results in the movement path shown in the illustration above. Starting slowly, the cone rapidly gains speed and then slows down again. This is also an ideal setting for a rotation sequence, if you want to start and end the rotation with smooth movements. Actually, you can use this setting for all animation tracks where smooth starts and end phases are required. This applies also to the scale track and the parameter track.



Take care to coordinate the acceleration values of adjacent keyframes. The illustration above shows an example of a strong acceleration towards keyframe 2 but no acceleration has been entered for the "From Key" parameter in keyframe 2. As a result the cone speeds up considerably towards keyframe 2, but then, abruptly, the movement continues with a much slower and constant pace.

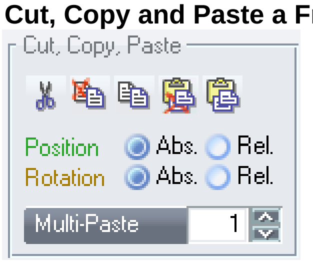


This example looks much better. To get a smooth transition in keyframe 2, in addition to the acceleration value specified for the "To Key" parameter also a deceleration value has been entered for the "From Key" parameter of this keyframe. Because no acceleration value has been entered for the following keyframe 3, the deceleration slowly returns to a constant speed.



The illustration shows the corresponding timeline for this movement. The triangles with the rising gradient after the first and before the second keyframe indicate the acceleration phase, whereas the declined triangle behind the second keyframe indicates the deceleration phase. There is no triangle in front of the third keyframe, so the object approaches with constant speed.

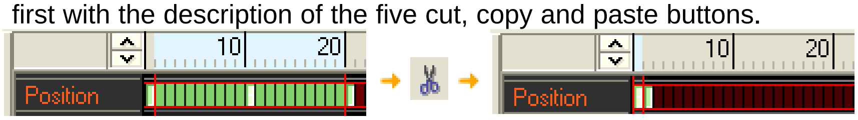
Cut, Copy and Paste a Frame-Zone



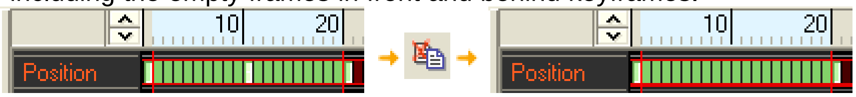
Animation sequences can be cut or copied to the clipboard and - after marking the destination selection - can be copied to another timeline position of the same object or any other object. Even the entire animation data of a character hierarchy can be copied this way, provided that the structures of the hierarchy trees of source and destination selection are similar. If, for instance, you copy the movement data of a character's skeleton over to another skeleton, then the order, in which the spine, arms and legs of the source skeleton are arranged in the hierarchy, have to match with the grouping of the destination skeleton. If the source hierarchy contains more objects than the destination hierarchy the unnecessary data of the remaining objects will be ignored. The same applies if you try to transfer animation data among incompatible track types. So just try and test everything. You can't do anything wrong. Use the undo functions frequently and when animating complex character animations never forget to make (numbered) backups of your project as often as possible.

Another special feature of the copy functions is that you can either transfer the keyframe data in absolute mode - as a fixed target position and orientation that an object will use to get in exactly the same position and orientation - or alternatively, you can copy the relative movement pattern, meaning the movement vectors and rotation instructions used to get from one keyposition to another.

Furthermore, with help of the Multi-Paste function, the animation data can be repeatedly copied to its destination position. Examples, demonstrating these powerful functions will follow, but let's start first with the description of the five cut, copy and paste buttons.



Cut - The marked frame zone is cut out and copied to the buffer - the area on the right of the marked range then moves up to fill the gap. Note, that not only the keyframes are copied to the buffer but also the number of "empty" frames contained in the marked frame zone. Later, when pasting the buffer data into an animation track again, an exact copy of the cut out frame zone will be inserted, including the empty frames in front and behind keyframes.



Copy and Remove Keys in Selection - The marked frame zone is copied to the buffer. After that the keyframes in the marked frame zone are transformed to empty frames. No frames are deleted by this action, the animation therefore retains its previous length.

Copy - The marked frame zone is copied to the buffer.

To transfer the animation data from the buffer to a destination track, select the corresponding track and mark the frameposition where you want to insert the data. Alternatively you can mark a complete frame zone again, if you want to replace the marked selection with the animation data from the buffer.

Paste - Replace Selection - The destination frame zone will be deleted and replaced by the data copied from the buffer.

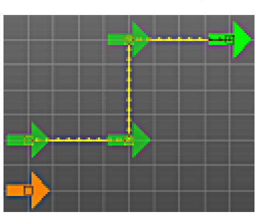
Paste - The animation data is inserted in front of the selected destination frame.

Absolute or Relative Copy of Position and Rotation Key Data

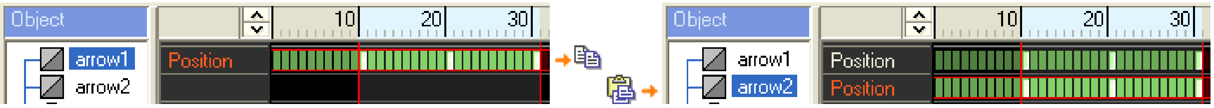
As previously mentioned, you can either transfer the keyframe data in absolute mode - as a fixed destination position and orientation - or alternatively, you can copy the relative movement pattern, meaning the movement vectors and rotation instructions used to get from one keyposition to another. By using the buttons shown above, you can decide which mode is used to copy the data to the buffer. These buttons are only relevant for the cut and copy functions. Once you have copied the data to the buffer, you can't switch over to the other mode for the paste functions.

- Absolute Mode - The keyframe data of the source object will be copied as a fixed destination position and orientation that is transferred to another object, which will use this data to arrange it in exactly the same position and with the same orientation (in regard to the object axes systems of this object).
- Relative Mode - The movement vectors and angles as seen from the source object's local coordinate system will be applied to the destination object's local coordinate system. If, for example, a character is moving "forward" along its local z-body axis and this movement is copied in Relative Mode to the destination object, then the destination object will perform this movement along its own z-body-axis. Similarly for rotations - the copied rotation data of the source object will be used to rotate the destination body about the axes of its own local coordinate system.

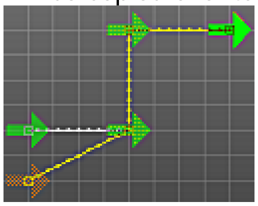
As copying a relative movement pattern instead of absolute positions and angles is such a powerful tool, some examples follow:



In the illustration you see 2 arrow objects. Three position keys were created for the green "arrow1" at 10 step intervals in the animation, just by moving the arrow to the right, then to the top and finally to the right of the screen again. Now, we want to copy these 3 position keys over to the orange "arrow2".

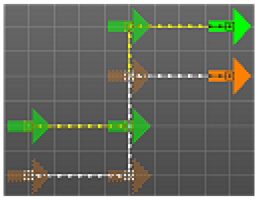


We call up the animation editor, select the position track of object "arrow1" and then we mark the frame zone on the timeline which includes all 3 keyframes. Now copy the frame zone in "Position - Abs." mode as absolute positions to the buffer. Then select "arrow2" in the object list. The selected frame zone can be left as it is, we only need to operate the paste button now and the keyframe data will be copied over to the position track of "arrow2".

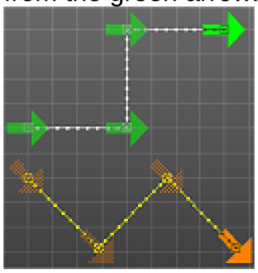


As a result, the orange arrow moves from its starting position to the first copied keyposition and from there on it moves on an identical path together with the green arrow.

Now we restore the initial project again by undoing the previous copy function. Back in the animation editor we repeat the previous work steps, but before copying the data to the buffer we switch over to the relative mode by selecting the "Position - Rel."button.

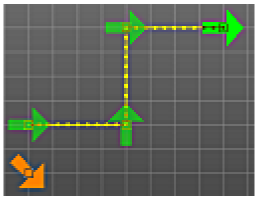


The illustration shows the scene after copying the data from the green arrow over to the orange arrow in relative mode. Since only the relative movement vectors have been transferred, the orange arrow moves on a parallel path with the green arrow.
But as mentioned before, the transfer of relative movements is dependent of the orientation of the object's local axes system. So what would happen, if we had rotated the orange arrow into another direction before copying the data from the green arrow over to it? Theoretically, it should then move "forward" along the directions of its own rotated axes system, but with the movement pattern copied from the green arrow.

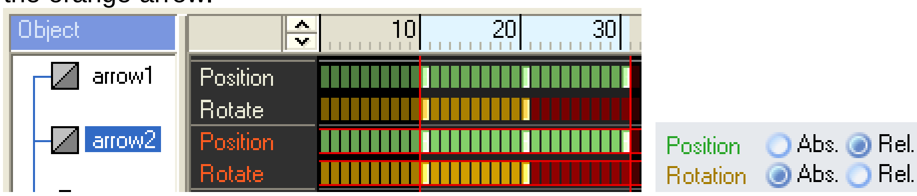


And that's exactly how it is. For the example in the illustration above the orange arrow was rotated first about 45° clockwise. After copying the movement from the green arrow over to the orange arrow, the orange arrow moves diagonal downwards along its own local axes system, but with the copied movement pattern of the green arrow.

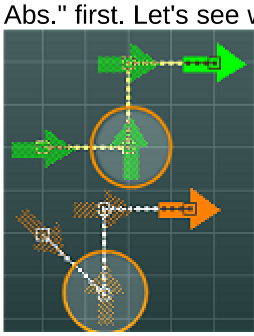
The previous examples referred only to position tracks. The arrows were merely displaced on the screen without further rotations turning them in the direction of the path. Here is another starting situation.



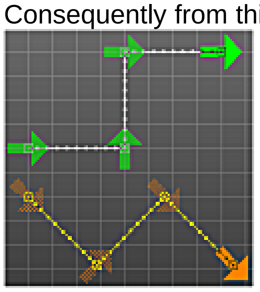
The movement path of the green arrow keeps the same, but on the first keyposition it rotates upwards and on the second keyposition it turns again to the right, so that the arrow always faces towards the direction of the path. Now let's copy again the movement including the rotations over to the orange arrow.



This time we select both position and rotate tracks to copy the selected frame zone from arrow1 to arrow2. Before copying the animation data to the buffer we have to choose the copy mode again. For the position track we select relative mode again, but for the rotate track we select "Rotation Abs." first. Let's see what happens.



As before, the orange arrow moves correctly diagonal downwards towards the first keyframe position, but from there on it moves again parallel with the green arrow. Why is that? Well, the absolute orientations of the green arrow's axes system have been copied to the orange arrow's axes system, therefore in keyframe 1 and all following keyframes both arrows face in the same direction. Consequently from this point both arrows move on parallel paths in the same direction.

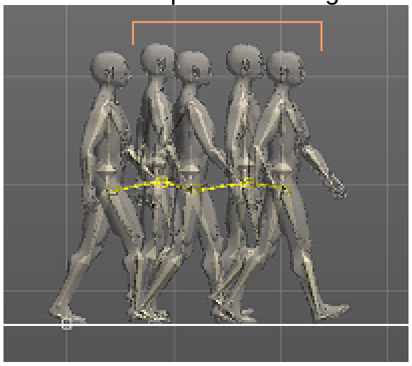


This illustration shows the scene after copying both the position and the rotate tracks in relative mode. This time everything is as we would expect. Both objects move along their own axes systems while simultaneously turning in the direction of their own movement paths. Only the movement vectors, the rotation axes and angles have been copied and transferred from the object axes system of the green arrow to the local object axes system of the orange arrow resulting in a copy of the movement pattern instead of fixed positions and alignments.

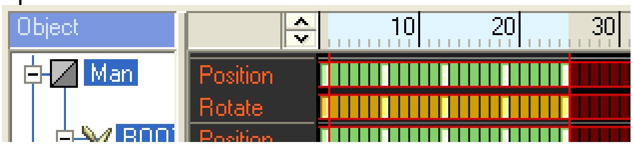
To reveal the real power of the copy functions we continue with some more complex examples. Suppose, for instance, you have animated a character in a little walking sequence. You can copy this sequence to let the figure repeat its movement or you can copy the movement over to another character. You can think of several cases:

- Creating a copy of the whole object hierarchy of the walking character and moving it to another position from where it then walks in another direction. To achieve this you only have to copy the character hierarchy in the Select Objects 133 dialog - all animation data will be copied with the model data. Afterwards you simply need to move 197 the copy of the character in "Move Object" mode together with its movement path to a new starting position. Then - in "Rotate Object" mode - you just rotate 211 the figure, again with its movement path included in the rotation, so that it faces into a new direction. If you now play a preview animation you can see that the second figure really walks with the copied animation data from the first character from a new starting point in a new direction. And the best part is you did not even have to call up the animation editor once. However, the characters are still moving uniformly with matching step sequences. Now you can call up the animation editor and move the complete walking sequence for one character a little bit to the right, so that the character starts to walk a little later.
- Copying a movement pattern (position track and rotation track in relative mode) in the animation editor from the source character over to a similarly-structured second character. The destination hierarchy is located at a different position and faces in another direction, but nevertheless, it will adopt the movement pattern from the first character and move along its own line of vision. However, copying relative movements will only work properly if both characters are arranged in the same pose, so that the alignment of the bones among each other in the source hierarchy matches with the alignment of the bones in the destination hierarchy. For instance, a movement that makes a standing figure neal down can only be transferred to a figure that is also standing. Of course, every pose can also be copied in absolute mode in the animation editor, so that transitions from any pose into any other pose can be animated. Since the destination hierarchy will adopt the absolute positions and angles of the source hierarchy, you have to move and rotate the destination hierarchy afterwards to its desired end position and direction. Example:
  - You have animated a character so that it walks a few steps straight forward, kneels down, stands up again, and continues to walk some steps forward. Now you want the character to kneal down once more. You have already animated this movement before (when the character knelt down for the first time) so you just need to copy the keyframe data (in absolute mode) from that particular frameposition to the end of the animation, where the figure should kneel down a second time. When playing a preview animation you can observe that the figure in fact kneels down at the end of the animation, but at the same time it returns to the location it occupied when first kneeling down. This is not unexpected, since we copied the absolute positions and alignments of the character. To correct the displacement you just need to move the character now in its knealing pose to the required position at the end of the animation.

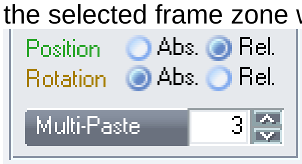
Another example: Extending a walking sequence



The illustration shows a character animated in a full walking sequence over about 5 keyframes. The sequence starts with the character standing with the right leg in front of the left leg and then making two steps forward until it stands again in the same pose with the right leg in front. Up to this stage we had to animate the sequence by hand. Now we want the character to repeat the sequence 3 times, so that it walks another 6 steps forward. We will do this via the copy functions of the animation editor. The question is now, which frame zone do we have to select for this operation? The character's pose in key 1 and key 5 is identical, both times standing with the right leg in front of the left one. If we were to now copy the complete range about all 5 keyframes, then after each full sequence we would have two succive keyframes with the same body posture and the animation would falter at these moments. Therefore, we include only the keyframes 2 to 5 for the copy operation.

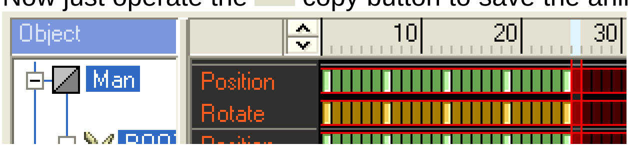


In the animation editor we select the whole hierarchy of the character by clicking on the root object of the hierarchy (usually the skeleton's skin object). Then we select the frame zone that includes keyframe 2 to 5. We also include the empty frames lying between keyframe 1 and keyframe 2, so the selected frame zone will reach from frame 2 to frame 25.

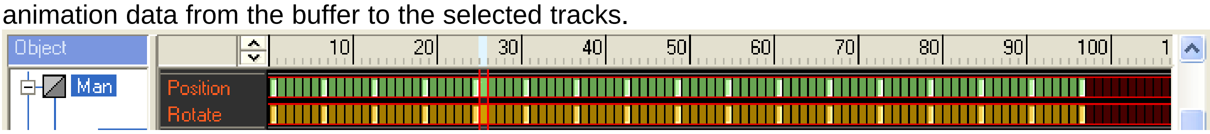


The character should walk on independently after copying the animation data, without us having to correct the positions in each keyframe. Therefore we choose the relative copy mode for the position tracks, so that only movement vectors are transferred instead of fixed positions.
In this example, the rotation track can be copied in absolute mode as well as in relative mode. If you rotate in relative mode, only a rotation axis and a rotation angle will be copied and not the alignment of the object axes itself, but since keyframe 1 and keyframe 5 hold the same axes alignments in their keyframe data, it doesn't matter if you copy the rotation instructions leading from keyframe 1 to keyframe 2 behind keyframe 5 or the absolute alignment of the axes system. Both will rotate the bones in the correct positions.

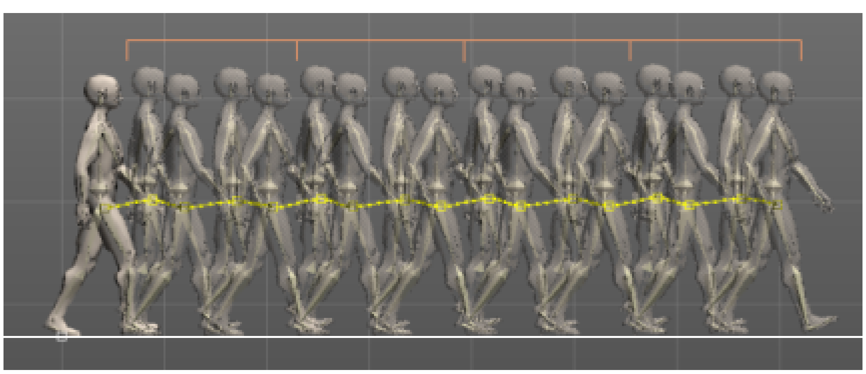
Now just operate the  copy button to save the animation data to the buffer.



Then select frameposition 26 as the new destination position in the timeline. Before pasting the animation data to this frameposition we first adjust the number of copies we want to insert. Since we want to repeat the walking sequence 3 times we also input a corresponding number into the "Multi-Paste" edit field of the copy box. Ok, that's it, now you can click on the -paste button to copy the animation data from the buffer to the selected tracks.



This is how the timeline is presented after the paste operation...



..and here you see the final animation. The character walks smoothly through his 4 walking cycles, performing 8 absolutely perfect steps.

An example of a character already animated in a simple walking sequence is provided in the projects-folder under "..projects/character/man_walk.cmo". You can also find the pure animated skeleton (without skin) for that scene in the project folder "..projects/character/skeleton_walk.cmo". The model of the character was kindly provided by the artist Stefan Danecki.

Duration of an Animation, Animation Range and Play Speed

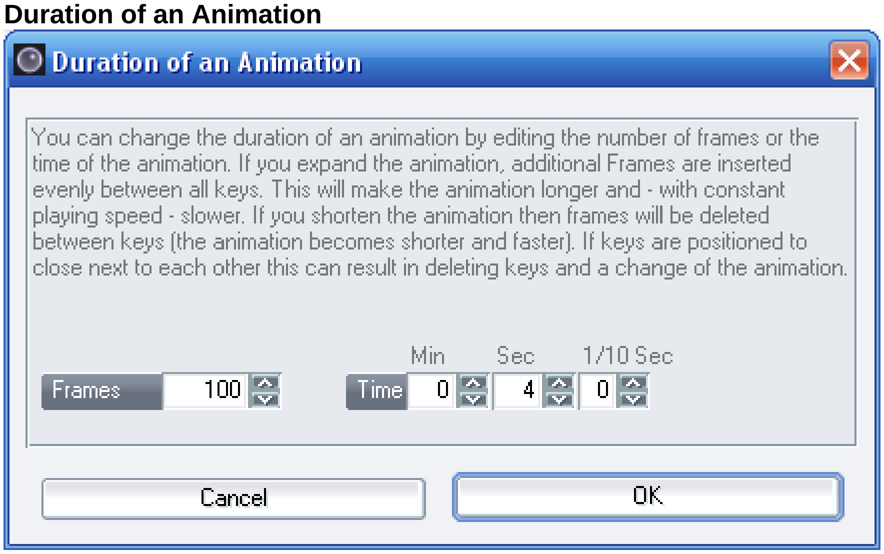
You can set the duration of an animation, the overall number of frames to be rendered, and the playing speed in the "Animation Range & Speed" box.

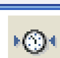
Expanding the Frame Range of the Animation

There are several possibilities to extend your animation with additional frames:

- You can simply increase the End Frame parameter which defines the target frame at which the animation should end.
- If you click with the mouse on a timeline position that is behind the last frame of the animation, then the animation automatically extends the relevant number of frames.
- In the main menu, after leaving the animation editor, use the "jump forward" navigation buttons of the animation button-strip or the frameposition edit field to extend the animation.

Duration of an Animation



If you click on the -clock icon in the "Animation Range & Speed" box, a small dialog opens and you can change the duration of the animation by editing the number of total frames or the time of the animation. If you expand the animation, additional frames are inserted evenly between all keys. This will make the animation longer and - with constant playing speed - slower. If you shorten the animation, frames will be deleted between keys (the animation becomes shorter and faster). Be careful - if the animation is shortened too drastically and if keys are positioned very close to each other this may result in deleted keys and a change of the animation.

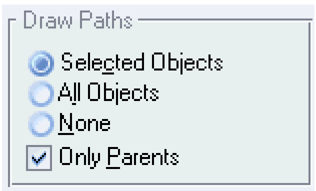
Part-Render an Animation

You can render parts of the animation by using the keyframe start and end parameters in the

"Animation Range & Speed" box. This way you may render the first part of the animation to check the quality and rendering settings and then later render the remaining part of the animation. You can also spread the animation over several computers by rendering a different part of the animation on each computer using the keyframe start and end parameters. Use a video post processing program to combine the different parts of the animation videos. In the same way, you can interrupt the rendering of an animation at a certain frame, save the video and later continue the rendering in a second video using this frame as the start parameter for the rest of the animation.

Playing Speed
The fps-parameter stands for frames per second (number of pictures per second) and determines the playing speed of an animation. This value, as well as being used during preview animations, is also saved as the playing speed with the generated AVI-file.

Draw Animation Path



With the path selection buttons you can decide which objects' movement paths are drawn in the viewport windows:

- *Selected Objects* - Only for the marked object selection animation paths will be drawn.
- *All Objects* - The paths of all objects are drawn.
- *None* - No paths are drawn.
- *Only Parents* - For clarity, only the movement paths for the top-most parents in selected hierarchies are drawn.

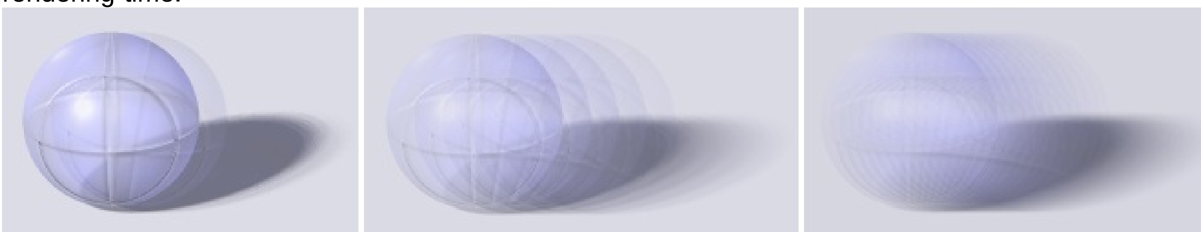
Motion Blur

Moving objects and backgrounds appear smudged on photos and films - the amount of this movement blurring depending on exposure-time, object speed and camera speed. With Motion Blur you can produce this effect over several consecutive pictures of an animation. At the bottom right of the animation-editor is a small box in which you can switch on the effect with the <Motion Blur> button. It gives you two further parameters, which decide the number of individual pictures to superimpose and the number of additional pictures to render in-between:

- *Frames* **-** Determines the number of individual pictures to apply the effect to. These pictures would be calculated in the course of the animation anyway so calculations for the Motion Blur effect is restricted to the pictures already being rendered.
- *Tweens* **-** If this parameter is greater than 0, additional between-pictures are rendered, which should reinforce the smear-effect. These pictures do not go into the animation, but serve only as storage for the Motion-Blur effect and are then deleted.

Example: You want to render a small animation of 10 pictures with Motion Blur effect. You enter a value of 3 for Frames and a value of 2 for Tweens. Two additional pictures are calculated between each existing frame, increasing the number of the pictures rendered for this animation to 10+ (10-1)* 2 = 28 pictures. To render the Motion Blur effect for each frame of the 10 picture animation, 3 Frame pictures + (3-1)* 2 Tween pictures = 7 pictures are overlaid.
Do not render tweens unless absolutely necessary as the calculation of each picture between requires the same length of time as the calculation of the "normal frames". Whether or not tweens

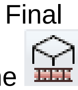
are required depends on the speed of movement of individual objects, and therefore the number of frames between two keyframes and the required effect. Large steps without additionally calculated tweens produce a stroboscopic effect (that may be what was wanted).
If you plan to render your animations as interlaced videos for TV-output with field rendering 270 switched on, then remember that with field rendering twice as many pictures (each of half resolution) are rendered, and this can also reduce or eliminate the need to render motion blur - which can save rendering time.



Here you see 3 examples for the Motion Blur effect. The sphere moves from the right to the left, the step-width is the same every time - only the parameters for Frames and Tweens are changed:
Left - Frames 2, Tweens 0. **Center** - Frames 5, Tweens 0. **Right** - Frames 5, Tweens 2

Individual Pictures with Motion-Blur Effect
The Motion Blur effect can also be used in rendering single pictures. If in an animation with Motion Blur you render only a single picture, then all pictures are automatically calculated that lie prior to the picture being rendered and are required for the over-lay.

Start Animation Calculation

To start rendering of an animation simply press <Render Scene Animation> or <Render Final Animation>. Both options are also available in the main menu under "Render" and via the  Render Scene Animation 259 and  Render Final Animation 259 buttons in the button strip. See chapter Render Image or Animation 259 for more information regarding the difference between both options and for further instructions to save animations and to convert and compress 259 large animation files to common video files using third party video processing software.

Part XVI

17 Particle Systems

Particle systems are always useful where a great number of objects are to be moved and it would be very wasteful to animate everything individually. With the help of a particle-system you simply choose an existing reference object from the program, then automatically generate up to several thousand copies, which continuously move under the influence of gravity, friction and turbulence, or simple follow the movements of the reference object. As all facet-based objects can serve as the original for particles, the list of possibilities is almost endless. Each particle is a complete object with regard to its material-attributes and reflection, transparencies or casting shadows. Animated bitmaps on the reference objects are also reproduced on the particles.
Under the influence of gravity and other physical forces it can simulate explosions, snow, meteor-swarms, insect-swarms, ballistic flying-objects, volcanic eruptions at irregular intervals or you can even use the particle system to enforest whole landscapes.
Another option is to create streams of simple two-dimensional pixel particles instead of generating copies of complete 3D objects. Using a depth buffer of the previously rendered scene, these two-dimensional pixel particles are painted in post processing mode into the picture. Two-dimensional particle streams are ideal where large quantities of small particles have to be employed, e.g., thin trails of smoke consisting of thousands and thousands of semitransparent pixels or the foam of a waterfall. (There are also special background particle systems for rain and snow 390 - these are managed on a separate register tab in the background dialog.)

Another spectacular application for particle systems. Only three basic plant objects (picture on the left) were used as reference objects for three corresponding particle actions in which thousands of object copies were created and rotated, scaled and scattered at random on the terrain to produce a complex forest.

More examples: Fog clouds generated with billboard particles, a meteor swarm, rising bubbles and a thin trail of smoke created with the help of pixel particles. Most of these examples are described in the tutorials section and can be found in the relevant project folders of the MR-3D Designer installation.

Particle System Generation

Particle actions are not especially difficult to set up. All the parameters for a complete particle action can be set up in the one dialog for particle systems. Only for collision-tests or if you want to drop particles on a particular ground object need you change to the Object Properties 290 dialog in order to select objects that are to be considered as a ground or for collisions with particles. Furthermore you can temporarily switch the particle actions you have set up off or on again with the global "Particle-Systems" button provided in the particle dialog as well as in the render options 263 dialog.

Previewing a Particle Animation

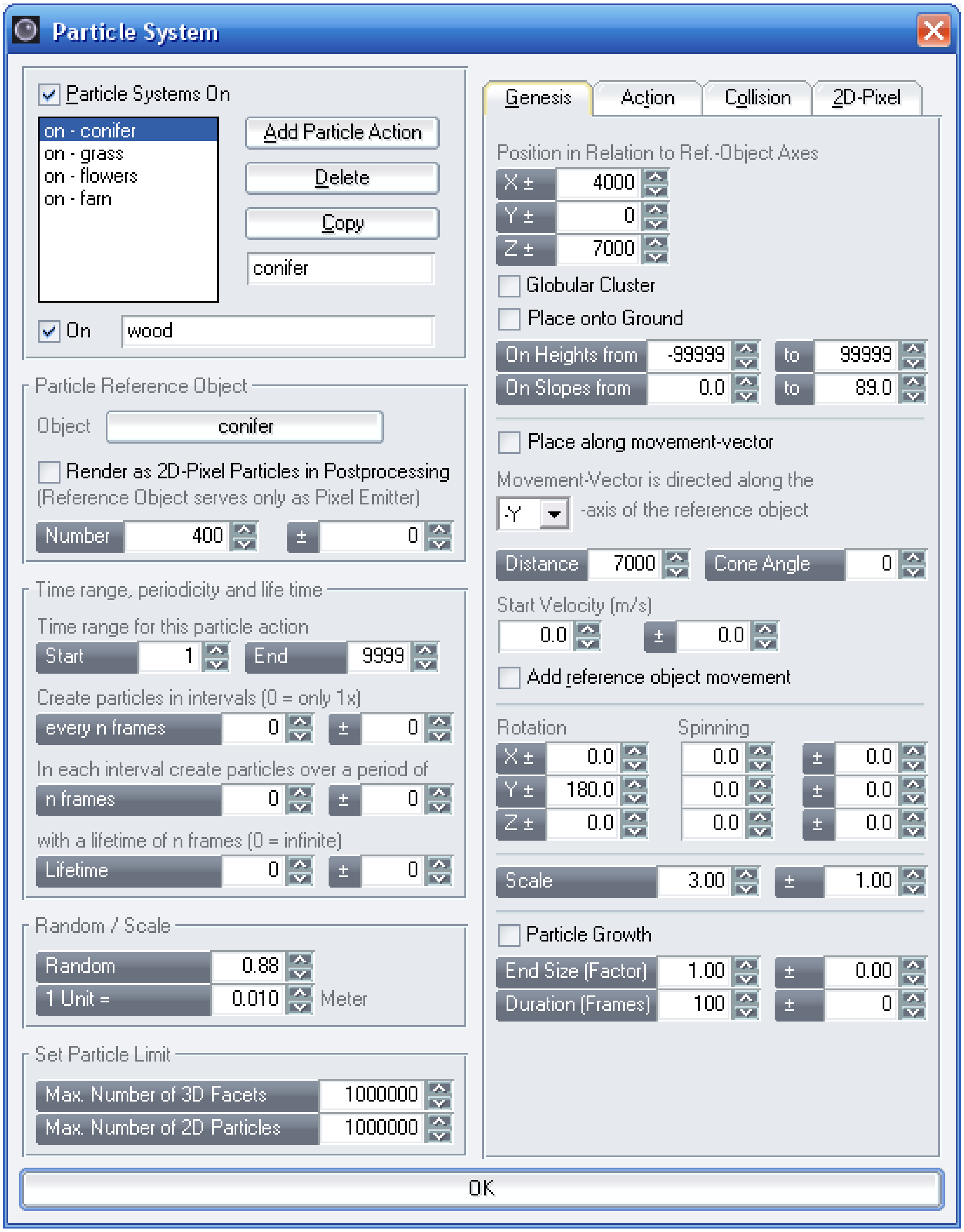
Particles are not drawn in the viewport windows where you are working. To view a preview of the particles you have to change first into Animation Mode and then start a rendering 259 as scene preview or in final quality. The particles are not displayed in the viewport windows due to the complexity of the particle animation. New particles are constantly being generated and destroyed during particle-actions. On each particle, which is moving with a certain speed and direction, additional external forces like chaotic turbulence, whirlwind and gravity are continually at work. Additionally, there is superimposed the movement of the reference object and also the changes of direction with regard to particles rebounding in collisions with other objects. Therefore, to show a particle-action correctly in the viewport-window at any one frame position, practically the entire record of the particle animation would have to be calculated and that is quite wasteful and time consuming.

Particle System - Overview

See also

- Tutorials - Particle Animation Examples 39

17.1 The Particle System Dialog



 - Menu "Options - Particles"

Switch On Particle Systems

You can switch on or off all particle actions for the picture calculation with the global <Particle Systems On> button at the top of the dialog . You will also find this button in the render options 263

dialog. Keep in mind that in order to see the particles in a preview rendering you have to be in Animation Mode on a frame position that is within the defined time range for a particular particle action.

Particle-Actions - Add, Delete, Copy

In the editor at the top left is a list-box in which all the defined particle-actions are displayed. By clicking on existing particle-actions in the list box you can change to and fro between these particle-actions and plan additional alterations.

New Particle-Action

To add a new particle-action, operate the <Add> button directly beside the list box.

Delete Particle-Action

To remove a particle-action from the list, click on the action in the list box and then operate the <Delete> button directly beside the list box.

Copy Particle-Action

If you require a second particle-action that differs only marginally from an already existing action, then select the existing action in the list box and copy it by operating the <Copy> button. You can then plan the alterations on the copied action.

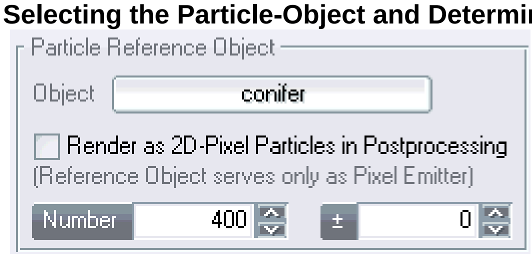
Edit Names and Descriptive Text

Click in the small bottom field beside the list box in order to change the name of the particle-action. Click in the field directly beneath the list box in order to input a short descriptive text for the particle-action.

Switch On Particle-Action

Switch on or off individual particle actions for the rendering with the <On> button beneath the list box of particle actions. You can also click on the name of a selected particle-action to change its status.

Selecting the Particle-Object and Determining the Number of Particles



Selecting a reference object

An existing reference object serves as the original for the production of particles. During the animation multiple copies of this object are created and animated dependent on the movement-restrictions and physical settings. Only facet-based objects can serve as a reference object - analytical primitives or light-objects cannot serve as reference. Operate the "Particle Reference" <Object> button to select an object for a particle-action. An object-selection window appears. Simply click on the desired object with the left mouse-button and then leave the dialog. The name of the object appears on the "Particle Reference Object" button.

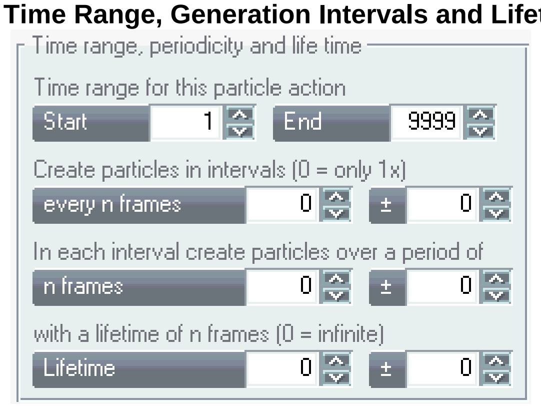
Render as 2D-Pixel Particles in Postprocessing

If you activate this option, a stream of simple two-dimensional pixel particles is created instead of generating copies of real 3D objects. Using a depth buffer of the previously rendered scene, these two-dimensional pixel particles are painted in post processing mode into the picture. Two-dimensional particle streams are ideal where large quantities of small particles have to be used - e.g., thin trails of smoke consisting of thousands and thousands of semi-transparent pixels or the

foam of a waterfall. Managing 2D pixels instead of complex 3D objects saves of course a lot of memory and rendering them in post processing mode with a depth buffer instead of tracing real objects speeds up the rendering time tremendously. On the other hand pixel particles cannot be seen in reflections or behind transparent objects.
All particle actions can be changed temporarily to two-dimensional particle systems by switching on this option. This can be helpful when rendering fast preview animations to control and adjust the settings for the particle movements and physics.
Reference Object for Pixel Particles - Even though pixel particles do not carry 3D-model data or materials you still need to define a reference object. Because all positions and movements are calculated for pixel particles in the same way as for 3D objects, we need the object axes system of a reference object to define an origin and corresponding movement vectors for the particle creation. In this way the reference object serves as an emitter for 2D pixels.

Determining the Number of Particle-Objects
In the "Particle Reference Object" box you can decide on the number of particles that are produced whenever generated by the defined particle-action.
The overall number of particle-objects during an animation depends on several factors. For example, a particle-action can be defined that generates particles once, at intervals, or even in each frame of an animation. In order not to use up all of the systems memory for the particle creation you should set upper limits for the storage of facets and pixels generated by all particle-actions. You can set these limits at bottom left of the particle dialog.
In additional to the particle-number parameter is another parameter marked "±". This serves to produce an amount of randomness into the particle-actions. If, for example, a value of 50 is given for the number of particles and a value of ±49 for the variance, every time particles are generated there are between 50-49= 1 and 50+49= 99 particle-objects generated.

Time Range, Generation Intervals and Lifetime



In the lower left of the dialog you can enter all parameters relating to the time sequence of particle generation and lifetime.

Particle Animation - Range of Validity and Lifetime
Particle-actions are independent of the settings in the animation-editor. You can specify the frame zone for the duration of a particle-action with the "Time range for this particle action - Start - End" parameters. The first particles are always generated at the start of the specified action. If the particle-action goes beyond the specified range, all particles that were generated by the action are deleted, regardless of their lifetime.

Create particles in intervals - every n frames
The interval parameter defines a pause after which the particle action starts again to create new particles.
- *Interval 0* - If you want to create new particles only once at the start of the particle action, for instance, to place particle plants on a landscape, then simply leave this parameter to zero.
- *Interval 1* - Enter an interval time of 1 if you want the particle-action to constantly create new

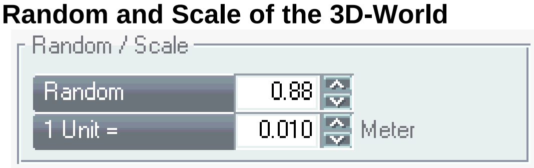
particles in each frame of the animation.

- *Interval greater than 1* - If the interval is greater than 1, then after this period the particle-action will start again to produce new particles for a particular time defined by the creation period.

In each interval create new particles over a period of n frames
When a new interval of particle creation begins, new particles will be generated in each frame over a particular time period that is defined with this parameter. If, for instance, the interval is set to 100 frames and the creation period to 10 frames, then every 100 frames new particles are created for the next 10 frames of the animation.

with a lifetime of n frames
The lifetime defines a period after which existing particles are deleted again so that they can be replaced by new ones. If a value of 0 frames is set for the lifetime the generated particles are visible for the entire particle-action; the particles, therefore, are "immortal".
If the animation exceeds the valid time range for the corresponding particle-action then all particles generated by this particle-action are deleted, independent of their remaining lifetime.

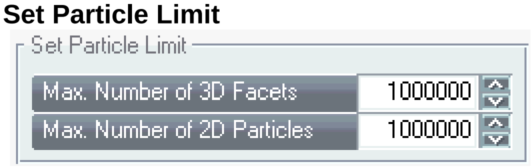
Random and Scale of the 3D-World



Initializing the Random-Generator - You will already have noticed that a certain amount of random latitude is allowed for most parameters. The random-number generator can be initialized with different starting values via the random-parameter on the lower left of the dialog. This way you can vary the particle-animation somewhat if you do not like the current random-settings or, especially important, when copying existing particle-actions. If you choose the same reference object for identical particle-actions then also all particles would appear in the same places with identical movement directions. This can be avoided by simply entering different initial random values.

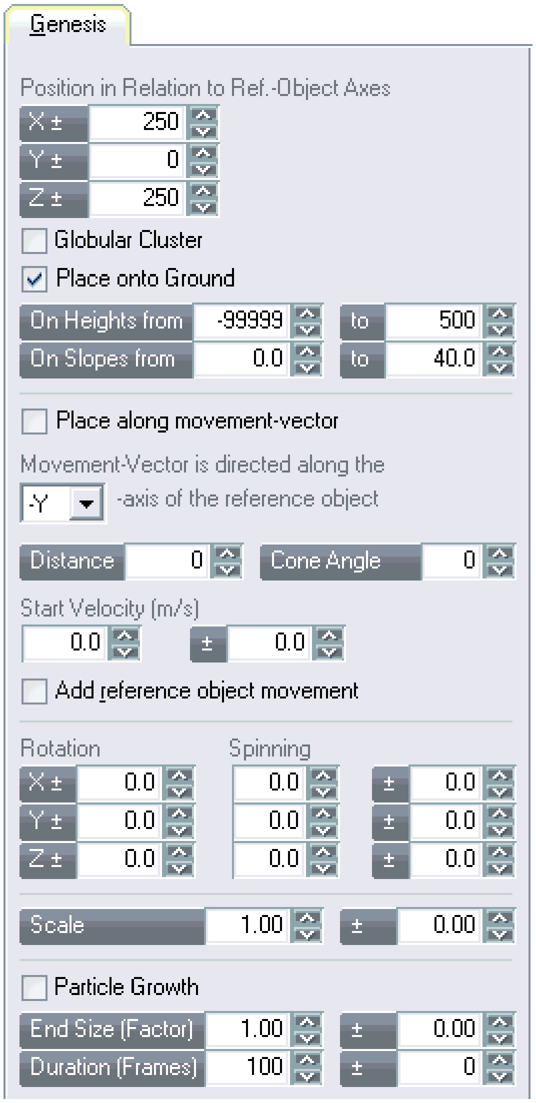
Specifying the Scale for an Area-Unit - Directly under the random parameter is the scale parameter with which you can specify how many meters correspond to an area-unit in the 3D-space. This is absolutely essential, for the following reason:
Speed and acceleration are measured in metres per second and metres per second^2 respectively. Therefore, in order to render a sensible simulation of the action of particles the program must know how many meters correspond to an area-unit in the 3D-space.

Set Particle Limit



Particle creation can be restricted to a maximum number of polygons used for 3D particles or to a maximum number of pixels stored for the 2D particles, respectively. If this limits are exceeded no further particles will be created until memory is freed again by the "death" of existing particles.

17.2 Particle "Genesis"

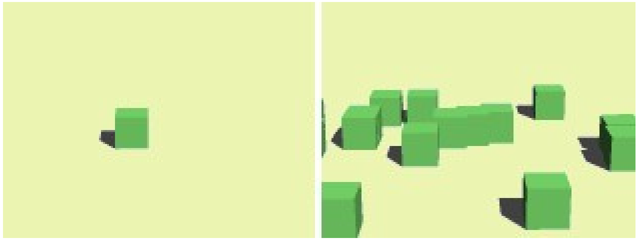


Select the "Genesis" tab in the particle system dialog 427 to bring the property sheet with all starting-parameters for the particles to the front. On this side you can edit all the parameters for newly generated particles, such as position, initial velocity, individual-rotation and so on.

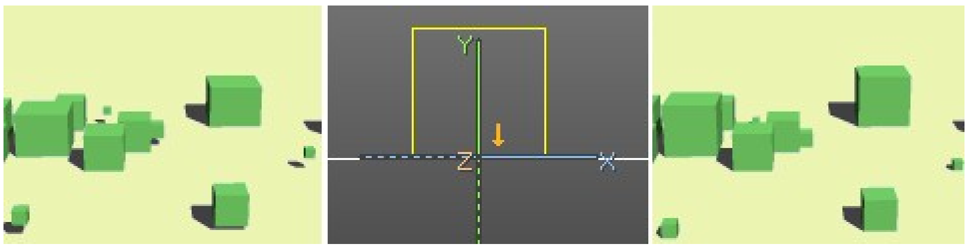
Start Position of the Particles when they are Generated

The start position for generated particles depends on the current position of the reference object. If the position parameters are zero, new particles are generated at the origin of the reference object. You can insert a random offset from this position with the X-, Y-, Z-parameters. All statements of position refer to the object-axes center of the reference object.

The position of the object axes within the reference object is particularly important if you plan to distribute particles of random size on an even surface.

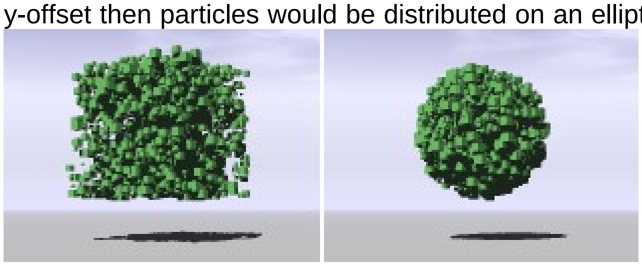


If the reference object is placed on the ground then particles can be generated on the same height level around the reference object using the x- and z-position offset parameters (no y-offset). The picture above shows a simple box and its particle copies created with random x- and z- offset.



But when applying also a random change of size to the particles the copied boxes will not stand anymore on the plane. The big boxes have sunken into the ground while the smaller copies are hovering above the plane. This is due to the fact that the center of the object axes of the reference object are used as a pivot point to scale and reposition the particle copies. Usually at object creation these object axes are placed in the center of an object.
Now its very easy to move the object axes to the bottom of the reference object (picture in the middle). After that, on particle creation this new axes position will be used as a root point for the placement and scaling of the particle copies. As a result all copied boxes of different sizes are properly placed on the ground, as you can see in the picture to the right.

Globular Cluster - Offset as Radius - The x-, y- and z-position offsets define a rectangular box in which particles are positioned at random. But if you switch on the <Globular Cluster> button then the position offsets are interpreted as the maximum radii of an ellipsoid or sphere. If, e.g., the x-, y- and z-offsets are the same then a spherical particle cluster will be the result. If you enter zero for the y-offset then particles would be distributed on an elliptical or circular area.



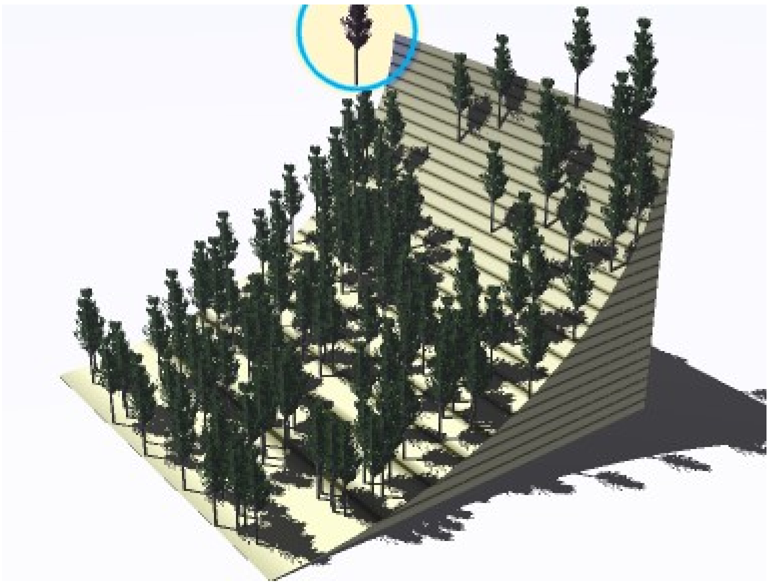
Example: The left picture shows a cubical cluster formed by 2500 small particle cubes. The right picture shows the same scene after the <Globular Cluster> button has been switched on.

Place onto Ground

You can even place particles on an uneven surface, e.g., a hilly terrain. In this case you would position the particle reference object above the surface and activate the <Place onto Ground> option. Additionally, you have to mark those objects you want the particles to land on by switching on their object property <Particle Ground 290 > in the Object Properties dialog. Now, all newly generated particle positions are checked against all marked ground objects. If a particle is positioned above a ground object it is dropped onto the surface of the ground, otherwise the particle will be deleted again.
As a reference point for the placement again the body axes position of the reference object is relevant. To avoid the particles sinking into the ground up to the half of their body you should again

move the object axes of the reference object down towards its base position. If you want the objects to sink in a little bit, for instance, if you intend to plant particle trees on a slopy terrain, then simply move the object axes somewhat above the base position. Then the tree will sink into the ground wide enough, so that even on slopes their roots will not partly hover in the air.

Height Range and Maximum Slope Angle - When dropping particles to the ground you can further limit the particle creation to areas with a particular height and a maximum slope angle.

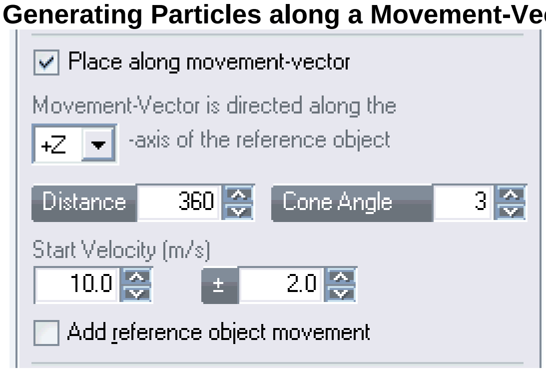


Example - Trees are distributed in dependence on the height and the slope angle of the ground object. Above the ground object, marked with a blue circle, the reference object for the tree generation is visible. If you don't want the reference object to appear in the final rendering, you have to switch it off beforehand, since it will not be dropped together with its particle children. The project file for this example is also provided in the project folder (../projects/particledrop.cmo"). You can load it and test some variations for the height range and slope angle to become acquainted with these functions.

A word regarding the tree particles used in this example. Normally, only simple polygon based objects can be used as a particle reference (no groups or hierarchies of objects). But it is quite simple in MR-3D Designer to join objects (Boolean Operations 233) into a single object while maintaining their materials in separate frozen facet selections 233. This way you can even join such complex multi textured objects like the tree in a single object which then can be used as a reference object for particle systems. By the way, the conifer tree is not that complex at all since it is constructed just from a simple cylindrical trunk and 6 planar surfaces. The branches and the green are simply projected and cut out from the transparent planes by applying a corresponding masked bitmap material. (See - Material Mix to Mask and Cut Out Surface Areas 300)

And this can a picture look like if these "simple" tree particles are combined with thousands of similar constructed particle grass bushes and flowers. The picture shows only a detail from a complete animation rendered with these techniques. You can view the animation in the gallery section of the 3d-designer.com internet site.

Generating Particles along a Movement-Vector



There is another possibility to define a particle's position on creation which is meant for particle actions with streaming particles, for instance, water drops emerging from a fountain or exhaust gas emitted from a jet engine.
Suppose you want to create a gas stream for a jet engine by continuously generating particles in each frame that should move in a set direction in an area of ±50 units behind the engine. The movement-vector must also be lined up along the longitudinal-axis of the engine. If the longitudinal-axis of the engine is the Z-body axis, you can use this as the direction in the corresponding movement vector-selection box. Give a value of 50 for the X-, Y- or Z-position and switch on the option "Place along movement-vector". If this option is switched on, then the random position offsets at the top of the page are ignored. Instead all particles are generated along a line defined by the chosen object axes and with a random distance specified by the corresponding <Distance> parameter.
You can also define a dispersal for the particle-production with the "Cone Angle" for the movement-vector so that particles are not generated in a dead straight line behind the engine. In this way the flame spreads out in a conical shape. If this value is, for example, ±180 degrees, the particles move randomly in all directions in a sphere. This can be used to simulate an explosion.
The "movement-vector" relates always to the corresponding object-axes of the reference object, which is then preset as the direction of movement. If, for example, the Y object-axis of the reference object points to the top and you choose "+Y" in the selection box, the particles will move vertically upwards.

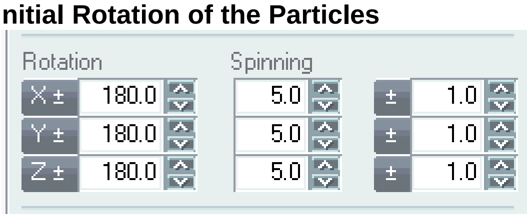
Start Velocity

The Start Velocity parameter gives the speed in metres/second with which the particles start to

move along their preset movement-vector on their creation. You can again put in a random deviation from this value over the ± parameter. Incidentally, the conversion of metres/second to kilometers/hour is 3.6. For example: 10 m/ s= 10* 3.6 km/ h= 36 km/ h
Note: Because we are simulating an accurate physical model, speed is relative to the scale of the scene and depends on the following things:
1. Speed is measured in metres/second. Both units must correspond to those found in the 3D-space. With the parameter "1 RE= 0,010 m" you can, for example, input how many area-units correspond to a metre. if a vehicle is 3 metres long and you construct it as a 200 units long 3D model, then the correct relationship is "1 RE= 3 m/ 200= 0,015 m."
2. The passage of time in an animation corresponds to the playing frame rate. The position of an object at a frame rate of only 10 pictures per second is moved a greater distance per picture than at a higher playing speed of, say, 24 pictures per second. The distance per second, however, always remains constant.

Superimposing the Movements of the Reference Object
Particle-clouds can perform extremely complicated series of movements. You need only animate the reference object in the normal manner and then switch on the option "superimpose movement of the reference object" for the particle-action. All particles generated for this action carry out the movements (positioning, rotation and animated scaling) of the reference object. Incidentally, the reference object need not be switched on during the animation. You can first switch on the reference object for the preparation of the animation-path and then switch it off again before commencement of the animation rendering. The particles nonetheless follow the animation-path of the reference object, without the possible distraction of rendering the reference object.
All other outside influences, such as starting-speed, gravity, turbulence, etc., can additionally be switched on as required and will introduce great variety into the set movement-path.

Initial Rotation of the Particles



The Rotation parameters determine a different random starting rotation-angle of ±180 degrees, about which each generated particle is turned.
Example: You have a "meteor" object as reference object model about which to animate a meteorite-swarm. All the meteorites - being based on the same reference object - are identical, but, with different views of the meteorites, due to different rotation-angles, you can bring some variation into the scene. In addition, you can also give the meteorites different sizes and provide them with different speeds of rotation.

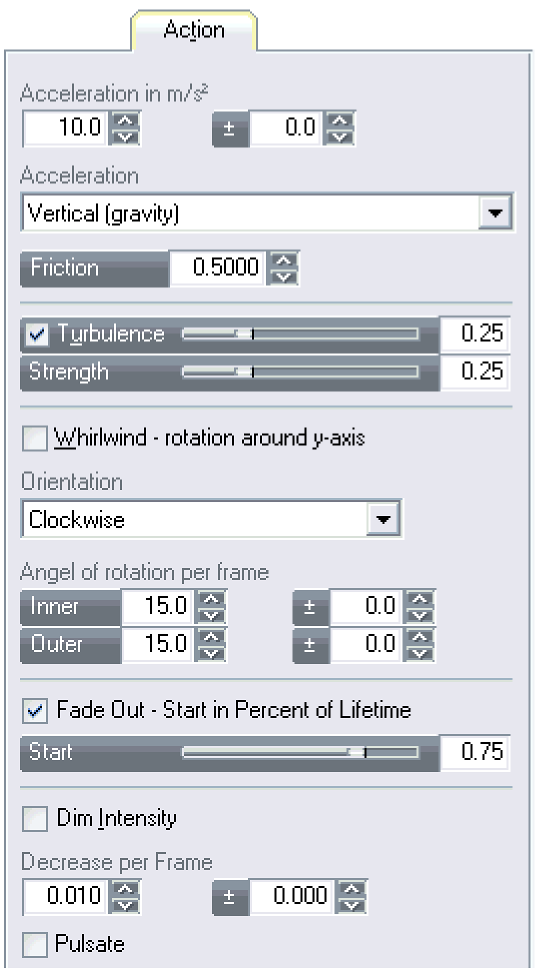
Spinning - Particles can spin about their own object-axes, independent of all outside influences and the movements of parent objects. The X-, Y- and Z-parameters give the angle through which the object turns per frame about the corresponding axis.
The object-axes do not necessarily need to be at the center of the particle-object. You can also generate particles about an imaginary focus, by positioning the object-axes outside the reference object. On particle-creation the relative position of the object-axes is also copied. By transferring the axes to the outside axes, the particles no longer rotate about their object-focus, but in a circular-movement around the imaginary axis-focus. In this way you can simulate leaves falling from a tree. With some gravity and self-rotation, the leaves hover and move towards the ground with a graceful "propeller-movement".

Scaling and Growth of Particles

Starting Size of Particles
Change the initial size of the particles with the Scale parameter. The range is given with the ± parameter. Example: Scaling: 1 ± of 0.5
All particle-copies of the reference object are scaled with a random factor that lies between 0.5 and 1.5.

Particle Growth
If you switch on the <Particle Growth> option, the particles will constantly grow over a particular period of time (duration in frames) until they reach their end size. The end size is a factor of the original size with which the individual particles were created.

17.3 Particle "Action"



Select the "Action" tab in the particle systems dialog 427 to bring to the front the property sheet with all parameters for gravity, friction, turbulence, whirls and fading.

Acceleration and Friction

Acceleration
This field determines a value for acceleration. Acceleration is measured in metres per second ^ 2. You can insert the direction in which the accelerating force should operate in the selection box. If "Vertical (gravity)" is entered here a vertical accelerating force works in the negative Y-area axis, which will simulate the acceleration due to the Earth's gravity - in the region of 9.81 m/ s^ 2.
This function is vital for many applications. Think of falling snow, bouncing balls, ballistic flight-curves and many other situations, which one can only achieve realistically through the combination of individual movement and gravity.
Here again, with the ± parameter you can generate random acceleration for the individual particles.

Friction
The acceleration due to the Earth's gravity is the same for all objects, independent of their mass - which implies that a sheet of paper, for example, falls to the ground equally as fast as a steel ball. That is true, however, only in empty space. In gas or liquid the actual speed depends on many additional factors such as, for example, the density of the medium, in which the falling body moves, the shape of the surface and the coefficient of friction of the body.
A further factor is the air resistance. Air friction increases with the increasing speed of the particles until it is then practically as great as the accelerating gravitational force. The object is then in equilibrium and no longer accelerates. The speed no longer increases and the object falls with constant speed from this point.
Forget about the physics: simply vary the acceleration with the ± parameter and experiment with the friction parameter.
For the technically-minded:
According to Stoke's law for the air-resistance to motion:
FL= 0.5 * cw-value of the body * density of the medium * surface roughness * V^ 2. The value (0.5 * cw-value * density * surface) we combine together into the constant k, which then produces: FL= k * V^2. Therefore, a constant times the speed squared produces the frictional resistance. It is this constant k that you can edit with the friction parameter.

Acceleration-Direction
The direction of the acceleration does not necessarily have to be vertically downwards. Negative values for the acceleration reverse the direction in which the acceleration works.
Other acceleration directions are available in the selection box:

- *Acceleration along movement-vector* - No matter in which direction your particles move your acceleration or deceleration is always confined to this direction.
- *Acceleration in the direction of the center of the reference object* - An interesting effect is obtained if the generated particles not only follow the movement path of the reference object 431, but additionally through the "gravity" of the reference object are pulled in elliptical paths around this object. Swarming motions can be excellently simulated.
  - A further possible effect is, for example, a solar flare in which continually ejected plasma-particles fall back again into the suns, or maybe, on an explosion directly followed by an implosion.
- *Vertical Acceleration* - will simulate the Earth's acceleration.

Turbulence

Switch to turbulence in order to bring some chaos into any movement-paths that appear too precise. The strength of the turbulence can be edited over the parameter of that name.
Example: Small smoke-particles that rise slowly from a chimney towards the sky and on their way are swirled and driven apart through air-turbulence.

Whirlwind - Rotation About the Y-Axis

Whirlwinds can be generated with this function. All particles that are generated in a frame by this particle-action rotate about the common focus of the particles. A whirlwind moving along its path is created by superimposing the movement of the reference object 431.
Whether the particles turn clockwise or anti-clockwise is determined by the "rotation-direction" in the selection box.
Over the rotation-angle-parameters you supply the rotation-angle through which the particles in each frame are rotated about the common focus. You can set different speeds of rotation for the inner and outer particles through different angle-sizes for the inner- and the outer-area. Rotation-speeds for particles between are interpolated.
Randomly varied rotation-speeds for each individual particle can again be defined with the ± parameter.
Additional disorder can be obtained through switching on turbulence.
Example: Generating a conical whirlwind. This is not at all simple in the first instance, because of how the particles are generated to result in an upright cone that can then be rotated. 2 possibilities are described in the following. The start point is a reference-particle-object on the ground, above which the whirlwind should develop.
1st option. A whirlwind at the beginning:

- Generate particles from the focus of the reference object, with a position-movement along the Y-axis and a distance along the movement vector of 100.
- Switch to the function "only positive along movement-vector"
- The movement-vector required is vertically upwards, therefore choose the + Y-axis as starting movement-vector.
- To create the particle's cone-shape, we direct the movement-vector through an opening-angle.

Ready. The vertical whirlwind can now be animated with the functions whirlwind, turbulence and by superimposing the movement of the reference object.
2nd option. A whirlwind lifting itself from the ground:

- This time, generate particles without initial distance along the movement vector, i.e. distance parameter = 0.
- Switch on the function "only positive along movement-vector"
- Set the movement-vector to vertical upwards by choosing the + Y-axis as starting movement-vector.
- To enable the particles to rise in a cone, we again direct the movement-vector through an opening-angle.
- Choose a starting-speed and vary it, so that the particles rise at different speeds and results in a good distribution of the particles in the required cone-formation.
- Give a friction-value for the particles. The particles lose their kinetic motion-energy through friction and remain for some time, thus resulting in the desired cone-formation.

Ready. The whirlwind standing on the ground can again be animated through the functions whirlwind and turbulence, and by superimposing the movements of the reference object. Naturally, gravity can also be switched on, so that the particles, depending on your vertical initial velocity and the relevant wind-relationships (turbulence) are again pulled to the ground.

Fade Out

With this option switched on, particles slowly fade out at the end of their lifetime. For instance, you could generate smoke particles which slowly expand (Particle Growth) and dissolve into nothing. The starting point from which the fade out effect begins is defined in percent of lifetime. Example: If a particle was created with a lifetime of 100 frames and the starting point for the Fade Out effect was set to 25% then from the 25th frame after creation of the particle it will begin to slowly fade until it becomes invisible and is deleted in the 100th frame of its lifetime.

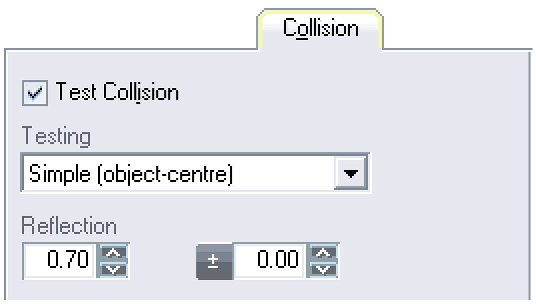
Color-Intensity Fade

For the particle-objects the material-settings are as those of the reference object. Textures, transparencies or even films projected onto the particles are therefore no problem. However, with "intensity fade" you can achieve another effect specific to particles. If the function is switched on, the particles loose their color intensity with increasing lifetime and become black. This function is

useful, for example, for the simulation of explosions, where the dispersing particles quickly loose their energy and their glow fades. With the "Fade per frame" parameter you can decide by what percentage per frame the color-intensity of the particles fade.
A good effect - especially in front of bright background - can be achieved whereby the "burned-out" particles continue to hang as clouds of black ashes in the turbulent wind. In a particle-animation 80 frames long, for example, in which the particles fade completely over 40 frames, requires a value of 1/ 40 = 0.025 for the fading per frame. In the remaining 40 frames of their short lives the particles can still continue to exist - as particles of black ash. The ± parameter relates to the random deviation that makes the whole process more chaotic and so appears more natural.

Pulsate
How would you create glowworms or a starry sky with some pulsating stars from a particle cloud? If the function <fade intensity> is switched on and you also activate the function <pulsate> the color-intensities are alternately faded out and then in again to the old value.

17.4 Particle "Collision"



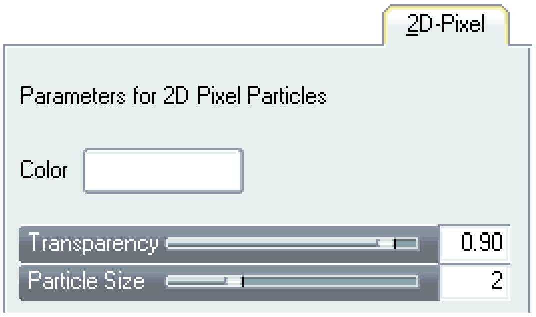
Select the "Collision" tab in the particle system dialog 427 to bring to the front the property sheet with all parameters for collision tests.

Switching on Collision-test
The collision-test is switched on if particle-objects should interact with the area - for example, bounce off the ground or walls, or remain lying on objects. However, to speed up the collision-calculations, only the objects for which the option "Particles - Reflector" 290 in the Object Properties dialog is switched on are considered. No collision-tests are executed between the particles.

Simple or extended collision-test
In the selection box can be decided if a simple or complex collision-test is used. The simple collision-test is for very small particle-objects, i.e. usually a simple small triangle, or perhaps for a fast animation preview. During the simple collision-test only the focus of a particle is examined on collision with other objects. With the extended collision-test all points of the particle are covered by the calculation.

Particle rebound
If a particle meets an object, the force of the rebound depends upon elasticity. Drop a marble onto a hard floor, and it rebounds to almost exactly the same height. A completely non-elastic lump of clay would, instead, remain on the ground.
With the Rebound parameter you decide how the particles will respond on collision. A value of 0 signifies a total lack of elasticity, so that the particles remain down on the objects. A value of 1 corresponded to an ideal elasticity. If a particle with a Rebound value of 1 falls vertically to the ground it rebounds to its original position forever more, unless somehow its movement-energy is lost. Realistic values lie generally anywhere between 0 and 1. However, values greater than 1 are permitted. In this case, particles meeting object-walls will rebound with even greater speed (trampoline-effect).

17.5 Particle "2D-Pixel"

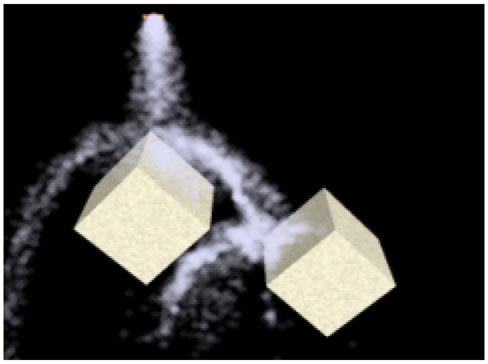


Selecting the "2D-Pixel" tab in the particle system dialog 427 brings all parameters for pixel particles to the front the property sheet .

With pixel particles, you can create streams of simple two-dimensional points instead of generating copies of real 3D objects. Using a depth buffer of the previously rendered scene, these two-dimensional pixel particles are painted in post processing mode into the picture. Two-dimensional particle streams are ideal where large quantities of small particles have to be used, e.g., thin trails of smoke consisting of thousands and thousands of semitransparent pixels or the foam of a waterfall. Managing 2D pixels instead of complex 3D objects of course saves a lot of memory and rendering them in post-processing mode (with the help of a depth buffer instead of tracing real objects) speeds up the rendering time tremendously. All particle actions can be changed temporarily to two-dimensional particle systems by switching on the corresponding button in the "Reference Object" box on the left side of the dialog. Even though pixel particles do not carry 3D-model data or materials you still need to define a reference object. Because all positions and movements are calculated for pixel particles in the same way as for 3D objects, we need the object axes system of a reference object to define an origin and corresponding movement vectors for the particle creation. In this way, the reference object serves as an emitter for 2D pixels.

Pixel parameters - For the simple pixel points we just need to enter the pixel color, transparency and the required pixel size.

Example: This thin trail of cigarette smoke consists of 4 different particle streams - only slightly displaced from each other and with slightly different turbulence values to generate separate streams of smoke whirling and dancing around each other on their way upwards. See the tutorial "Subtle Trails of Smoke 121" for more details on how to create such smoke trails.



This example demonstrates the use of pixel particles for the simulation of downward pouring water. The corresponding project-file "..\projects\12_particle_collision2d.cmo" is also part of the MR-3D Designer installation.

Index

- 3 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- R -

- V -

- W -

- Z -