Mental ray rendering optimisation

by Christoph Schinko

What is rendering?
Rendering is the calculation of your scene in Softimage's internal 3-dimensional space. For this procedure the render engine **sends out rays** from where the **camera** is positioned. The rays travel straight from the camera centre into soft space until they **hit a triangle** (objects get divided into triangles -> tessellation). If there is no object in that direction, the ray would keep going forever and Softimage would end up with infinite recursion. To prevent that, a process called **subdividing** is started in which the renderer sets borders around your scene, in the form of a box. This box will be cut in half, first in x, then in y and z. Those 8 boxes get cut in half again in the same order. Softimage continues subdividing until certain values are reached. Those values are specified in the **BSP tree** (binary space partitioning) and determine when Softimage has to stop subdividing. In mental ray there are two values called "**Max depth**" and "**Max size**". **Max depth** is the **maximum amount of subdivisions** before Softimage stops splitting a box and moves on to the next one. **Max size** sets the **maximum size of a leaf** that is in a box. If the leaf is smaller than the value, this box won't be subdivided anymore. We can manipulate the BSP tree settings in *Render -> Options* to decrease the rendering time, because the boxes don't have to be very small when there's nothing in them. This is the last fine tuning for mental ray (mr) – rendering optimisation. However, there are many things to do before you get acceptable rendering times (they're never "great", they always could be faster).

How to optimise a scene

Geometry
Cutting down on geometry is the most important part of optimising a scene. If objects are modelled in more detail than necessary, go in and **delete rows** in U and V (*Edit -> Delete Point*), but make sure that the quality of the render is still good. Go into *Info -> Selection* for every object and cut down the steps as far as possible. If an object does not cast shadows, turn off *Shadow* under *Info -> Selection -> Render Setup* and is it not reflective or refractive, also turn off *Secondary rays*. If you take off *Primary rays*, the object does not show up when rendering, because it does not get hit by the (primary) ray from the camera.

Schematics
Clean your **schematics window**! I know it's a pain, but go in and delete all the stuff that you don't need anymore. Old splines, paths, objects, hidden stuff, materials, textures, lights,… everything that won't affect your rendering.
Make sure you show all relations (light, model, motion, matter) in the schematics view in order to see what is connected to an object that you might still need.
Remember that lots of constraints, expressions and so on slow down the pre-processing stage.

Materials
Try to avoid a lot of reflective and refractive materials. A reflection faked by a **reflection map** looks pretty much the same and brings down rendering time.
Hit *Mat Oper -> Optimise all* to prevent materials with the same settings from existing more than once (watch for local textures!)

Textures
Textures, 2D and 3D, slow down rendering a lot. Go into Photoshop and **cut down the image size** of your texture images; always test render to check if you still have the same quality. Be careful with bump-, reflection-, transparency- and especially displacement mapping. They can destroy your rendering dreams just like that!
Also before you render, **delete all the temp files** on your machine.
Under **c:\temp**, throw away all:
- Temporary files (*.tmp)
- Preview-picture files (CE10.511.pic, CE7.301.pic,…) and
- Crash-backup folders (siKO-9, siKO-2,…).

If you have access, clean up your hard drive to get more room. Empty your recycling bin and **defragment your hard drive**. It takes a while longer to read the texture pic-files when they are fragmented all over the hard disk. Then restart Softimage (and even your computer) and render your scene.

Lights
If a light doesn't reach all objects in the scene make it **selective**, so the light is not being calculated where it is black anyway. Try to avoid area lights. The give very nice results, but take forever to render. Also use **ray trace on a minimum amount of lights** to get faster renders.

Camera
Under *Camera -> Settings* you can specify the **near- and far plane** of the camera. That gives you control over where Softimage starts and stops ray tracing. The near plane is the distance from the camera to where the scene starts being visible, measured in Softimage units. The far plane specifies how far away to kill the 'shot-out' ray. That way objects in the back, that would not be visible anyways are not being rendered at all. You can measure the desired **distances** with *Info -> Distance*. In the status line (where you can also access the status file), Softimage tells you how to use this tool. Middle click on the start point (camera,…) then middle click and hold down the mouse button to see the distance in Softimage units on the bottom of the screen.

Rendering in Layers
If you have a **volumetric light** in your scene, render in layers. Set up a separate scene only with the volumetric light and the objects that are interacting with it, and composite that into your scene. Same goes for shadows. Set up a **shadow pass**, so you can render your scene without them. Works really well and gives a lot more control over the intensity of the shadows. Rendering in different layers is also the way the 'big boys' and their companies are going. For how to set up a shadow pass, ask for the "How to render a shadow pass for Composer" tutorial by Noel Wright.

Render settings

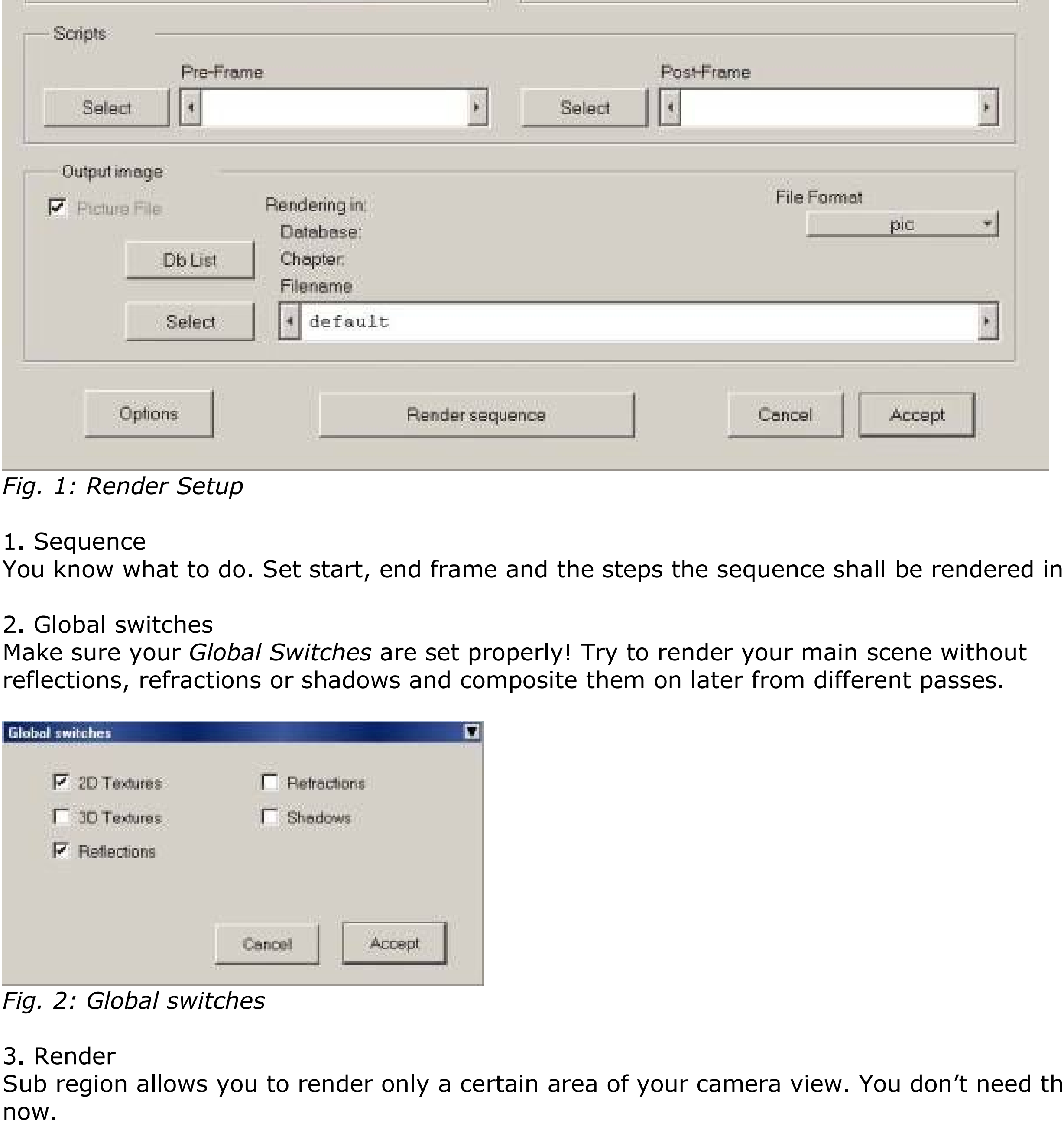

Fig. 1: Render Setup

1. Sequence
You know what to do. Set start, end frame and the steps the sequence shall be rendered in.

2. Global switches
Make sure your *Global Switches* are set properly! Try to render your main scene without reflections, refractions or shadows and composite them on later from different passes.

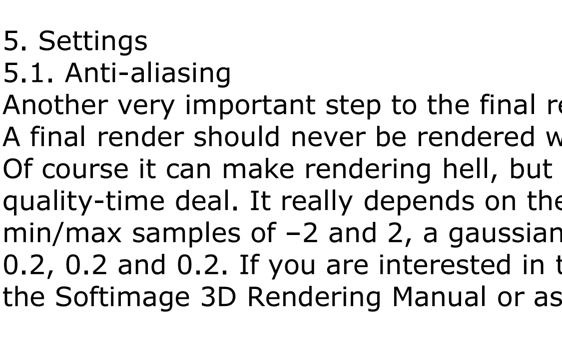

Fig. 2: Global switches

3. Render
Sub region allows you to render only a certain area of your camera view. You don't need that now.

4. Resolution
The simplest way of cutting down rendering time is make the picture smaller. If you change the aspect ratio under *Camera -> Picture Format*, you can save over 200 lines per frame to render! And you get professional looking 'wide screen'-bars on the top and bottom of your screen. It's easy to do and it looks cool. I am working with resolutions like 720x389 or 720x342. If your camera angle is already set, that might not be a good idea, because stuff on top and bottom of the screen gets cut off and more stuff on the left and right is shown.

5. Settings
5.1. Anti-aliasing
Another very important step to the final render is to decide which anti-aliasing settings to use. A final render should never be rendered without anti-aliasing, it just adds so much to the look. Of course it can make rendering hell, but with the right settings you can get a pretty good quality-time deal. It really depends on the scene but for most I would suggest to go with min/max samples of –2 and 2, a gaussian filter with 2 and 2 and adaptive super sampling of 0.2, 0.2 and 0.2. If you are interested in the different filters and their settings, please refer to the Softimage 3D Rendering Manual or ask Don Wilder for the statistics he wrote.

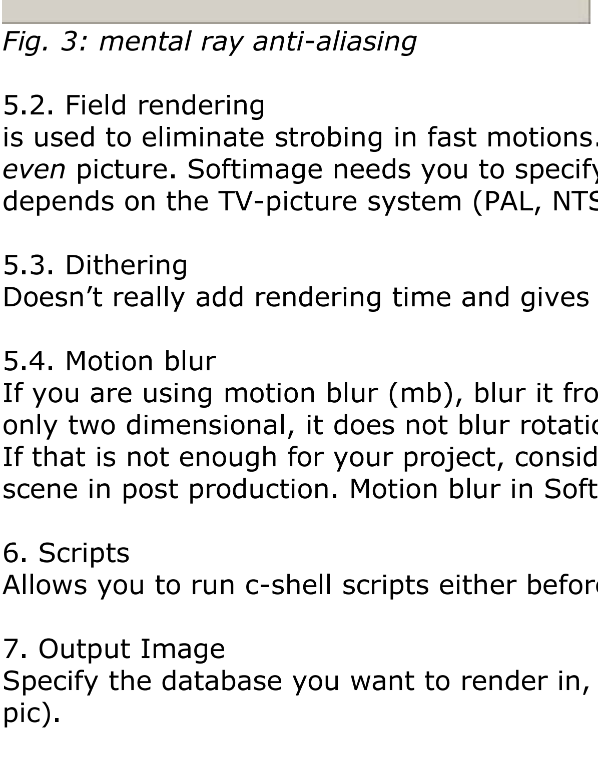

Fig. 3: mental ray anti-aliasing

5.2. Field rendering
is used to eliminate strobing in fast motions. Renders two images per frame. An *odd* and an *even* picture. Softimage needs you to specify with which one to start with, because that depends on the TV-picture system (PAL, NTSC). Starts with **ODD for NTSC**!

5.3. Dithering
Doesn't really add rendering time and gives a very settle but neat effect. I always have it on.

5.4. Motion blur
If you are using motion blur (mb), blur it from the **command prompt**. The mb-standalone is only two dimensional, it does not blur rotations or complicated movement.
If that is not enough for your project, consider **compositing** the blurred objects into your scene in post production. Motion blur in Softimage is so sloooww.

6. Scripts
Allows you to run c-shell scripts either before or after rendering.

7. Output Image
Specify the database you want to render in, the sequence name and the file format (usually pic).

Render -> Options

Getting closer to the end.

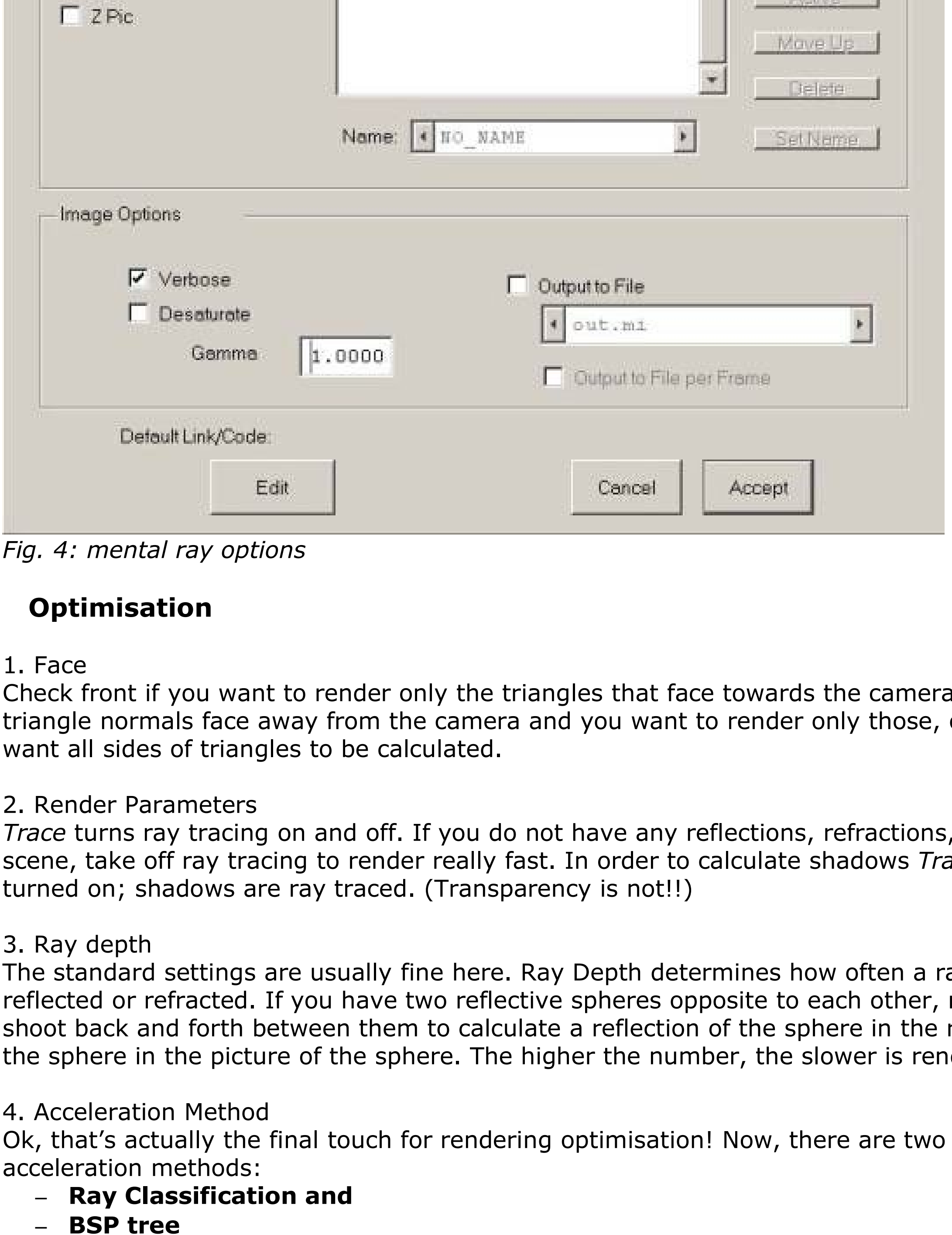

Fig. 4: mental ray options

Optimisation

1. Face
Check front if you want to render only the triangles that face towards the camera, back if triangle normals face away from the camera and you want to render only those, or both if you want all sides of triangles to be calculated.

2. Render Parameters
Trace turns ray tracing on and off. If you do not have any reflections, refractions,… in your scene, take off ray tracing to render really fast. In order to calculate shadows *Trace* has to be turned on; shadows are ray traced. (Transparency is not!!)

3. Ray depth
The standard settings are usually fine here. Ray Depth determines how often a ray can get reflected or refracted. If you have two reflective spheres opposite to each other, rays would shoot back and forth between them to calculate a reflection of the sphere in the reflection of the sphere in the picture of the sphere. The higher the number, the slower is rendering.

4. Acceleration Method
Ok, that's actually the final touch for rendering optimisation! Now, there are two different acceleration methods:
- **Ray Classification and**
- **BSP tree**

BSP tree is default setting and in most cases it works faster than ray classification. According to Microsoft Softimage, **ray classification** gives better results for scene heavier than **100.000 – 150.000 triangles** on machines with a **low amount of RAM**. Via ray classification options you can specify how mental ray uses the RAM available to prevent paging onto the hard drive, which slows down the process a lot. However, I tried ray classifications and although my example scene had 360.000 triangles, it was never faster than rendering with BSP tree. I suggest to stick with the BSP tree, it's complicated enough anyways.
BSP tree gives two settings to adjust. **Max Depth and Max Size**. We heard before that those values tell the mr renderer when to stop subdividing the scene (splitting it into boxes). Max depth and Max size are sort of dependent upon each other, if you change one value the other value shifts to somewhere else. That makes it kind of hard to figure out the exact settings. Wherever you look for information on mr render optimisation, everybody will tell you that it's more of an art than a science. There is no rule or exact way to get faster rendering times.

I read an interesting method of how to get optimum results:
Click on verbose in the *Render -> Options* dialogue box. Set your BSP tree to something like 60 and 6, render a frame.
When the render exits out to the Softimage UI, click on *Mode* on the very bottom of the screen, the status line. This opens up a viewer for the Softimage status file which contains detailed information of effects, error messages, rendering,…
The file *soft.status* is usually found under C:\temp or other temp directories. If it gets too big and you don't have access to delete it, just open it up from an NT-Explorer, cut out all the text stuff and re-save the empty file. That makes it a little bit easier to find what you're looking for back in Softimage. Sometimes, when you scroll down in the status file and you drag the little box instead of pushing the arrows, it does not always scroll to the very bottom of the file. In order to do that, use the arrows.

Max size:
Ok, now the value search begins. Before rendering, in the pre-processing stage, you get a
- value for the size of the largest leaf node. (write it down)

Then you see percentage values which show the progression of the render. After that, there is
- A number of candidate triangles per ray. (write it down)

The technique says that if those values are larger than 10, you have to **increase the max depth** of your BSP tree. Then you can make the **max size smaller**. That would cause the renderer to divide the scene up into more boxes, which increases pre-processing time, but the smaller leaves are faster to render. So you already see, that it is tough.

Max depth:
- Set max tree depth to a large number, like 50.
- Start rendering a frame, but cancel as soon as Soft is rendering on the screen.
- Go into the status file and
- Check out the *number of leaves*. (write it down)
- Set max depth to a smaller value and do the same thing again.

Keep doing that, and at a certain max tree depth the number of leaves should decrease drastically. Take that value you used +1 to get the optimum setting for max depth.

That's the theory. It unfortunately didn't work for me. The *size of the largest leaf node* and *number of candidate triangles per ray* stayed the same or increased and were nowhere near 10 (see fig. 5). And the number of leaves increased instead of decreased.

So, that's my method:
I used David Blazetich's final project as example scene. Here's the *Info -> Scene* values just for information.

363.439 triangles,
652 models,
446 materials,
492 textures,
84 pictures,
64 lights

global settings: Textures 2D, Reflections, Shadows
resolution: 720x342, no anti-aliasing

Render time for one frame when I got the scene: 1:45
Render time for one frame now: 0:59

When I got the scene, the values for max depth and max size were 60 and 6. All I did was render a frame, time it, write down pre-processing and rendering time. Set the depth value lower and rendered again. Here's a table of this procedure:

max depth	max size	size of largest leaf node	candidate triangles per ray	number of leaf objects	preprocessing time	rendering time	total time	
15	6	36060		531670	00:40	05:45	06:25	for 25% -> DOH!!
23	6	11079	208,29	1152876	00:28	00:49	01:17	
24	6	9860	153,87	1423784	00:29	00:41	01:10	
25	6	9225	116,89	1552234	00:29	00:36	01:05	
26	6	5585	88,28	1745598	00:31	00:32	01:03	
27	6	4225	71,79	2119698	00:33	00:29	01:02	
28	**6**	**2518**	**59,2**	**2340732**	**00:32**	**00:29**	**01:01**	
29	6	1678	50,51	2604048	00:36	00:26	01:02	
30	6	1405	46,49	3094568	00:44	00:26	01:10	
60	6	384	39,72	8493904	01:17	00:28	01:45	
100	6	384	39,72	11386887	01:33	00:43	02:16	
28	0	2518		2414480		01:01	01:01	
28	2	2518		2403984		01:01	01:01	
28	5	2518		2366510	00:34	00:28	01:02	
28	**8**	**2518**	**63,17**	**2267281**	**00:32**	**00:27**	**00:59**	
28	10	2518	65,96	2189374	00:31	00:29	01:00	
28	25	2518	94,74	1803180	00:30	00:32	01:02	
28	30	2518		1730572	00:28	00:34	01:02	
28	30	2518		1465214	00:27	00:43	01:10	

Fig. 5: table of BSP tree optimisation

I simply interpolated the max depth until I got the fastest setting. Changing the max size did not really bring too much. Anyway, try it out! Play around with it, have fun and I hope I could help a little. If something is missing or wrong on these pages, please let me know, this is just an overview of what I found out.
For more confusing information refer to the Softimage 3D Rendering Manual.

Questions or suggestions? Please e-mail me under stoph@christoph-schinko.com
Written 09/98 by Christoph Schinko, teaching assistant at Vancouver Film School.

Note: the e-mail address has been updated. Also, the pictures had to be re-inserted. Therefore the pictures show Softimage 3D 3.9.

Glossary

Area light - mr feature; light source that emits light from a surface instead of a point; very nice results, but long rendering times

Bsp tree - acceleration method

Defragmentation - moving clusters (little parts) on the hard drive to have all clusters of a file in a continuous order

Displacement mapping - mr feature; takes a bump map and converts the actual geometry to it

Leaf - stuff

Max depth - the maximum amount of how often a box can be subdivided

Max size - the maximum amount of leaves in a box before mr stops subdividing

Mental ray - high-end render engine for photo realistic results; thanks to the open architecture mr can communicate with custom written shaders

mr - mental ray

Pre-processing - calculations before the rendering starts; subdividing scene, loading in pic files for textures, starting shader routines,…

Radiosity - render type; each triangle gets values for how it is affected by all lights, the render is not camera dependent; once the scene is calculated, you can move freely in it. Beautiful results, but extremely slow rendering

Ray classification - acceleration method to render large scenes; good control over memory usage

Ray tracing - render type; shooting out rays from the camera and following their path as they hit objects, bounce off them (reflection) or pass through then (transparency, refraction)

Renderer - engine that subdivides and renders the scene

Shadow pass - separate sub-scene; containing only objects that cast and receive shadows -> render only shows shadows -> compositing

Subdividing - pre-processing stage; putting a box around the scene and splitting it up into smaller boxes until BSP tree tells it to stop

Tessellation - splitting up surfaces into triangles (with 3 corners) for rendering

UI - user interface

Volumetric light pass - separate sub-scene; containing only the volumetric light and the objects it affects -> compositing

Volumetric light - mr feature; spot light where the light cone is visible (beams, smoke,…)