

Maya Fast Simple Subsurface Scattering using mental ray

This shader uses mental ray, so go to the render globals and change from maya software to mental ray. Then set the render quality preset to “preview”

The example scene will simply use a nurbs sphere, but you can use any object in maya. With your object selected, press F5 to change to the rendering menu set and go to “Lighting/Shading->assign new material->Misss_fast_simple_maya” (it is near the bottom)

This shader is a simulated sub-surface scattering shader, and it is optimized so that it renders fast and doesn’t flicker (and is thus safe to use in animations). While it is not physically accurate, the results are pleasing to the eye and very fast to render.

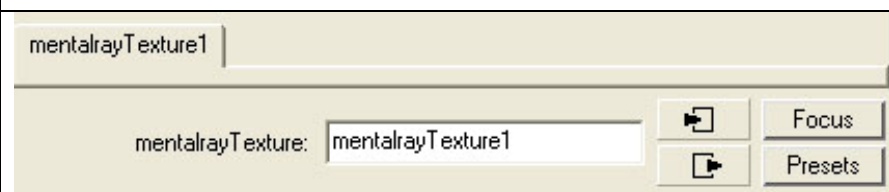
If you are in smooth shaded mode, then your object just turned green. Render and see what happens. You will see a tan Lambert shader with a specular highlight. There is no Sub-surface scattering yet. Back to your viewport:

Select your object, and press control+A for the attribute editor. At the top of that box, look on the right side and find “misss_fast_simple_maya1”

Click on that tab to bring up the attributes for the shader. Open the “lightmap” section, and click on the apply-texture checkerbox button.



Maya automatically creates and connects a lightmap_write node to your shader. You should be looking at the settings for this.



note the name of this node should be “mentalrayTexture1” - If it is not, write down or remember the name (case sensitive). you will need it later to make a connection.

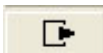
Check the “writable” box.

The width and height should be the same height as your render, and double the width. So for a 720x480 render, the lightmap should be 1440x480. Now change the file size depth to “32 bits”

You don’t need to specify an image name. But if you do, you can render the scene once, and uncheck the “writable” button. Then maya will not have to generate the lightmap each time you render the scene. This saves a considerable amount of time, as the lightmap is the most time consuming part of this shader.

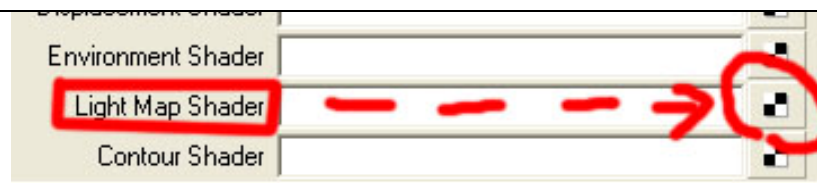
Now close the attribute editor and render again. It should still look exactly the same.

We created a lightmap, but it isn’t connected to the shader group. Select your object, and open the attribute editor. Now click on the “show output connections” button in the upper-right of the window.

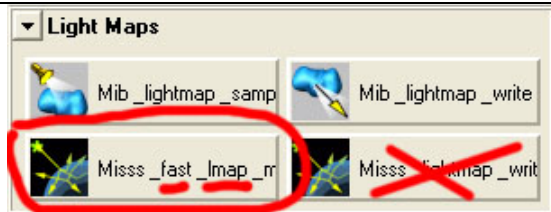
It looks like this: 

You should be taken to the shader group for the shader. Here you can specify all sorts of wonderful things for your shader. We need a lightmap.


Open the “mentalray” tab, and scroll down to “custom shaders” then find “Light Map Shader” and click on the texture apply button.



In the window that comes up, scroll down to the “lightmap” section where you should see 4 different icons. Click on the one called “Misss_fast_lmap_maya”



In the window that pops up this time open the “Lightmap Write” section and in the text field type “mentalrayTexture1”

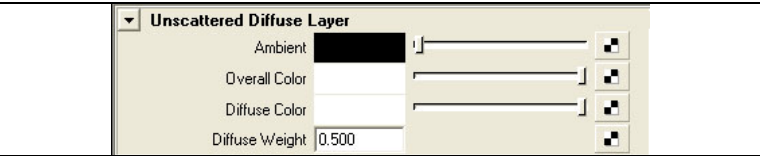
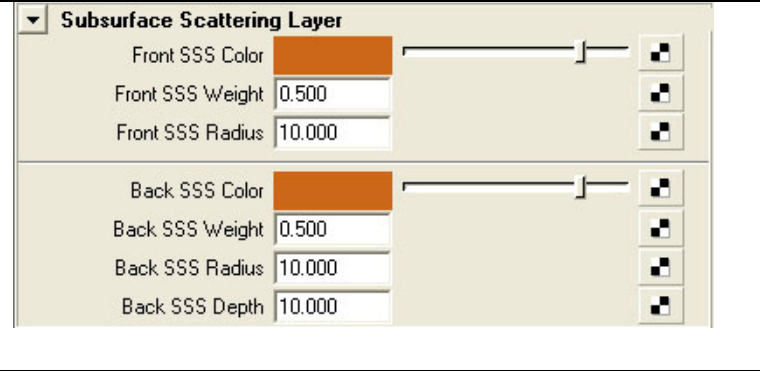

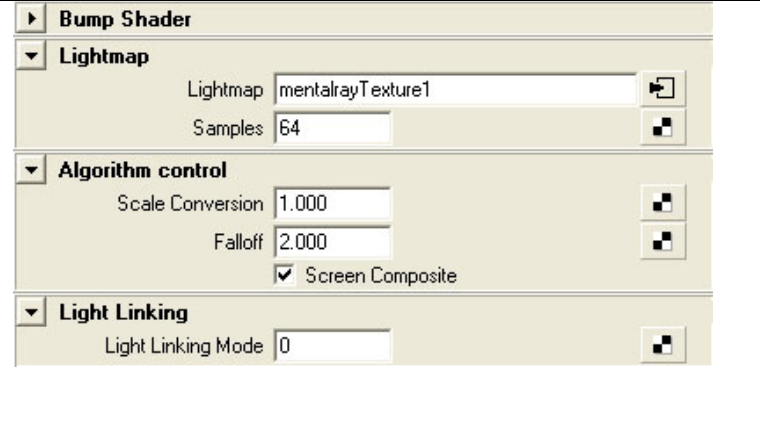
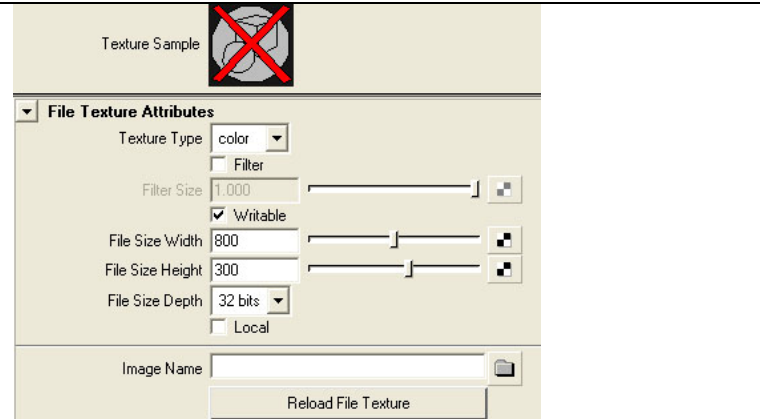
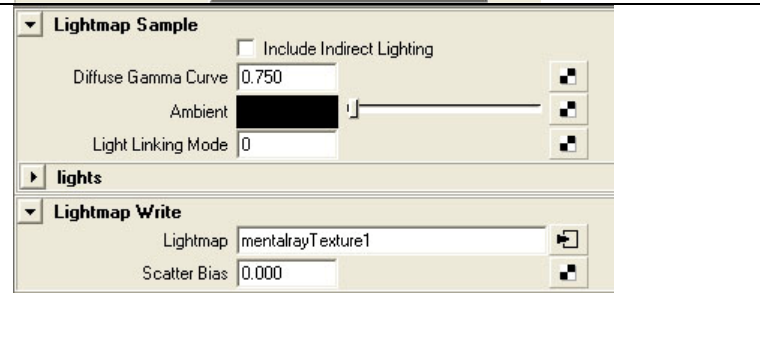
This links the lightmap node to the lightmap_write node you created earlier. If the name of that node was not “mentalrayTexture1” then type in whatever the name was. You can get the name by selecting your object, opening the attributes editor, clicking on the “misss_fast_simple_maya1” tab, and then clicking on the “show input connections” icon, which is above the show output connections button you clicked on earlier: 

Now render your scene, and you’ll see that you should have a nice tan object with very basic sub-surface scattering happening on the surface.

Here I have turned up the samples, and set the mentalray render settings to “production” quality. This rendered in 13 seconds on a dual-core Pentium 3GHz

See the next page for information on the settings for this shader.



	<p>This is the base layer, and works very much like a lambert shader does. Diffuse weight is simply a multiplier for the diffuse color.</p>
	<p>Pretty straightforward – front and back is relative to light position. So if you set the front color to orange, and the back color to red, you’ll have something like skin, and when you see light shine through the object it will be red.</p> <ul style="list-style-type: none"> - Color is the same as any other shader, this time it describes scattered light rather than diffuse or specular - Weight is a multiplier for the intensity of the scattered light. - Radius is the distance (in scene units) the light can scatter along the surface geometry. Not to be confused with Depth - Depth refers to the distance that light can travel through the object.
	<p>Specular color is the same as it is in any other shader. This is the very last layer to be added. More shininess means a more concentrated highlight.</p>
	<p>You can apply a bump shader, but if you do so, make sure you set up a bump network in the hypershade first. Mental ray shaders cannot automatically do that for you. You have to create a “bump2d” or “bump3d” node and link it to your bump material.</p> <p>Lightmap was discussed earlier. Increasing the samples is basically like increasing the quality of the scattering. More samples means smoother render.</p> <p>Scale conversion is a means of compensating for scale issues within maya. SSS is very scale dependent, so very large objects might not appear to scatter at all, unless you play with this value</p> <p>Light linking mode: 0 – use all lights : 1 – use all lights in the “lights” array below : 2 – exclude all lights in the array.</p>
	<p>Filter will soften the effect of the scattering a bit, but it also causes the shader to consume more memory. Use this if it is needed.</p> <p>Width = 2x render width Height = render height</p> <p>Filter size depth is another quality control. Lower depth means less quality, and faster rendering. Higher makes things look all smooth and nice.</p> <p>Check the “local” box if you are using multiple computers to render the scene. The render farm will need this.</p> <p>Image name is optional. But if you specify one, render your scene, and then uncheck the “writable” box, maya will use the already existing lightmap and save valuable render time.</p>
	<p>To get to this section, open the attributes for the shader, and click on the “output” button. Under “mentalray->custom shaders->Light Map Shader” click on the input button.</p> <p>Ambient color here is a powerful attribute. It has very strong effect on the render.</p> <p>Diffuse Gamma Curve controls the bias between scattering perpendicular to a light source, and scattering parallel to a light source. It’s strange, mess with it and see what happens.</p> <p>Scatter bias controls the bias between front and back scattering. -1 is all back, +1 is all front. 0 is the middle.</p>