

Tutorial – Breaking subsurface scattering into render layers

Intermediate Difficulty

Author: Nick Bartone

Website: <http://www.linkedin.com/in/nickbartone>

Greetings everyone. So I've had a number of people approach me here at school about a method I've been using to break my Miss_fast_skin_maya subsurface shader into render layers so I have near absolute control of it in compositing, so I thought I'd write up a tutorial here to share.

Please note: This tutorial is more about the process of breaking the subsurface into render layers and compositing them in post, rather than how to use the miss_fast_skin_maya subsurface shader. There are several rather excellent subsurface tutorials out there on the internet, a couple of them being here on CGTalk, though one in particular I've found to be very well written and quite verbose is Jared Martin's tutorial here (http://www.highend3d.com/maya/tutorials/rendering_lighting/shaders/135-1.html) on highend. However, please note that even though Jared's tutorial is based on the miss_fast_simple_maya subsurface shader, you'll find the fast simple and fast skin shaders are almost identical, except the skin shader brings the Epidermal, Subdermal, and Back scatter attributes in to replace the fast simple's front and back SSS color attributes. Also, the fast skin shader has more specular control than the latter. But the same methods used in this tutorial will work with the fast simple subsurface shader as well.

Also note, Maya 7.0.1 was used in this tutorial, however, the general methods apply to all 3d applications where render layers and layer overrides are achievable. This tutorial is assuming you already have your subsurface shader set up and working, though, if you have any questions or complications, feel free to ask and I'm sure one of the many bright minds here at CG Talk can point you in the right direction. Or I can help you, either way.
:p

By the end of this tutorial, I will have walked you through, not only how to break down your subsurface shader attributes into their own render layers, but also how to render out an occlusion pass, and a shadow pass for compositing in post. You could take this even further on your own as a fun little experiment to render out your light passes, ZDepth, Motion vectors, etc, but I won't be delving into that here. (Perhaps for another tutorial topic? – send me a PM if it's of interest)

Without further delay, time to begin! First things first, test your subsurface shader with a beauty pass, just to make sure all is in order. I wish I could say I had a pretty model for you guys properly UVed with a color/spec/bump map, but good ol' MaleHead here via the Modo example model includes will have to do. Obviously there are some issues here, but, for the purposes of this tutorial, this will work.



And, just in case you're a little behind, you can snag my shader attributes here as a starting point, they're not far from defaults, though, remember, depending on your lighting scenario, scene scale/units, and other factors, you will need to modify these values.


Attribute Editor: miss_fast_skin_maya3

List Selected Focus Attributes Help




miss_fast_skin_maya3 | mentalrayTexture2




miss_fast_skin_maya: miss_fast_skin_maya3




Focus Presets


Sample 

Diffuse Layer




Ambient   


Overall Color   


Diffuse Color   




Diffuse Weight 


Subsurface Scattering Layer


Epidermal Scatter Color   




Epidermal Scatter Weight 


Epidermal Scatter Radius 


Subdermal Scatter Color   


Subdermal Scatter Weight 

Subdermal Scatter Radius 


Back Scatter Color   


Back Scatter Weight 




Back Scatter Radius 


Back Scatter Depth 


Specularity


Overall Weight 




Edge Factor 


Primary Specular Color   


Primary Weight 


Primary Edge Weight 


Primary Shininess 


Secondary Specular Color   

Secondary Weight 


Secondary Edge Weight 

Secondary Shininess 


Reflect Weight 

Reflect Edge Weight 


Reflect Environment Only


Reflect Shininess 

Bump Shader


Bump 


Lightmap

Lightmap 

Samples 

Algorithm control

Scale Conversion 

Falloff 


Screen Composite

Lights

Extra Attributes

Notes: miss_fast_skin_maya3

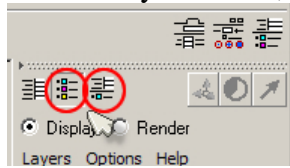
Select Load Attributes Copy Tab Close

As you can see, currently none of these attributes listed are mapped, however, it is very important to note that every one of these with the 'map' icon  can in fact be mapped. As you'll come to find, being able to map these attributes individually is very handy for controlling things; for example. For your back scatter, you'll notice that certain parts of the ear would reveal more back scatter than other denser parts, depending on your model, this can be done by tweaking your backscatter radius, weight, and depth until you get it just right, or, you can map said values and have more accurate control as to where you want the backscatter values to show up; likewise with all the other attributes. Though, once again, for this tutorial we're mainly interested on how to break this shader up into render layers.

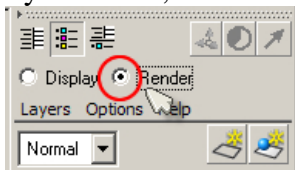
So now that our beauty pass is checked off and ready to go, it's time to start making the render layers. But first, let's take a look back up at the shader dialogue and assess what it is exactly we want to accomplish. In our composite, there are several different ways to build up our layers on top of each other to simulate the sub-surface effect in post, however, each method will require us to isolate each one of our different layer 'colors': Diffuse, Epidermal Scatter Color, Subdermal Scatter Color, Back Scatter Color, and Primary/Secondary Specular Colors. To do this, we need to create a render layer for each one of these attributes. In case you are new to the concept of render layers, you can access the render layer dialogue by click on the Show/Hide channel box/Layer editor near



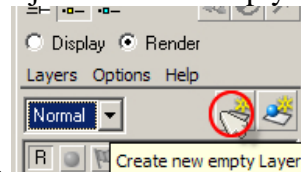
the top right of the interface and then you can choose to either show both the channel and layer editors, or just the layer editor by clicking on one of



these two buttons. Once at the Layer Editor dialogue, be sure to change your layer controls from 'Display' layers, which control the physical display layers of your scene, to Render layers, which, as you guessed it, controls the render



layers. Next, click on the icon shown in the following image to 'Create a new empty layer'. Because of some bugs that exist with Maya 7.0.1 with the other button 'Create a new layer and assign selected objects', we'll just create an empty



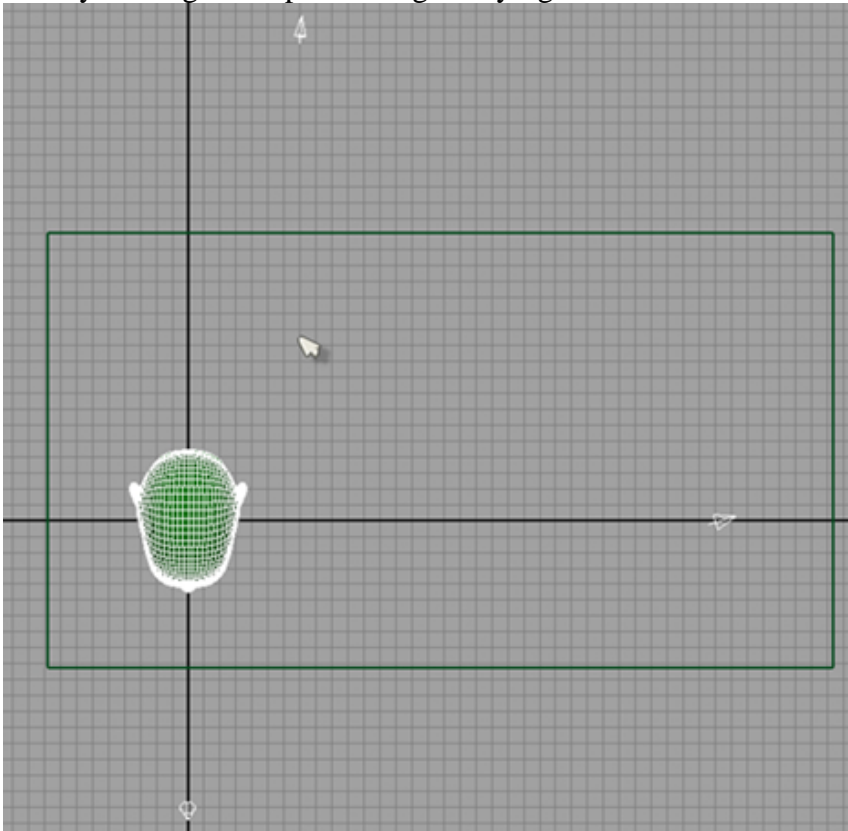
layer for now and manually add in the elements we want in there. You'll notice that two layers are added to the dialogue, a render layer called masterLayer, and a new layer, probably named layer1. What your masterLayer is, as you may have

guessed, is the base unmodified 'beauty pass' render layer that you are viewing by default. Layer1 is the new empty layer that we have just created. If you click on layer1, you'll see that there is nothing to display, why, because it's empty! We need to add in the objects we want in that render layer, but first, what to call it? A good habit to get into, is to first create what I usually call my masterCopy layer. This is a direct duplicate of my current masterLayer, just in case anything happens to fubar in my masterLayer, I have a backup! To rename this layer, you can double-click on the name layer1 to bring up the edit layer dialogue. Change the name here from layer1 to masterCopy and click save. Once that's taken care of, to add objects to this render layer, since it is a direct duplicate of our masterLayer, make sure your masterLayer is selected, Go to Edit > Select All. Go back over to your render layer dialogue, hold down right click on masterCopy and select 'Add Selected Objects'. All selected objects in the scene will be added to this render layer.

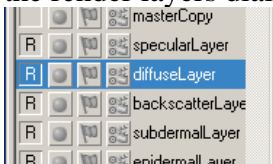
With that out of the way, the fun begins! Go ahead and create 7 more empty render layers, naming each one of them respectively: diffuseLayer, epidermalLayer, subdermalLayer, backscatterLayer, specularLayer, shadowLayer and finally occlusionLayer. Another good practice to get into when working with render layers is to name all of your layers according to relevance and also to stick to a naming convention. You don't have to use my naming convention, but find something that works for you that tells you what that layer contains but also follows good naming convention guidelines.

Let's begin with our diffuseLayer. Select your geometry, in my case, the head I'm working with, the two black spheres for the eyeballs and my lights. This view will also

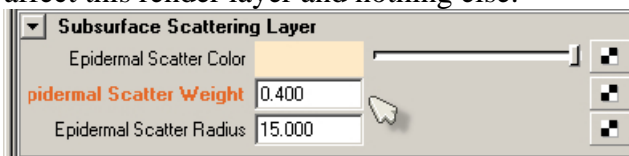
show you the general positioning of my lights for this scene.



Hold down right click on the diffuseLayer and click 'Add Selected Objects'. Now you should have a copy of your selection in the diffuseLayer. Highlight the diffuseLayer in the render layers dialogue to view that render layer, before you move on to the next step.

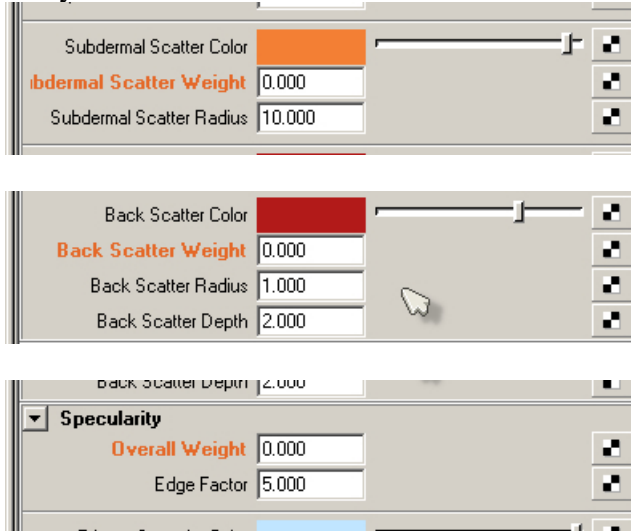


If it's not already open, open up your hypershade and bring up your attributes for your subsurface shader. What we want to do for this layer is to isolate our diffuse color and essentially hide everything else. To do this, we will create a layer override on the epidermal scatter weight, subdermal scatter weight, back scatter weight, and the overall weight of our specularity. To create a layer override on a parameter, hold down right click on the attribute title, and click on 'Create Layer Override'. Doing this will make the text of the attribute bold and orange to indicate that changes to this parameter will only affect this render layer and nothing else.

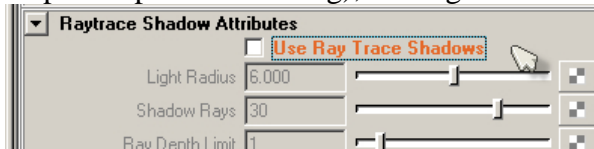


. Now, set the epidermal scatter weight to 0. This will zero out the effects of the epidermal scatter in the render and output

nothing instead, this is exactly what we want. We will continue down the list and do the same for subdermal scatter weight, back scatter weight, and then specularity. Don't worry if you notice a slight delay after changing the attributes here, remember, each time you change the values, mental ray re-renders the small display icon, this is what is causing the delay.



With all these values set to 0, your thumbnail should now look essentially like a flat lambert with your diffuse color, excellent! We have isolated the diffuse color. However, there's one additional step we need to take for compositing purposes. In your scene, select whatever lights you have that are casting shadows, and bring up their attributes. For the diffuse layer, and all layers following, we want to turn off the shadow casting, except for on our shadowLayer of course, but we'll delve into that later. To turn off the shadow casting, we will use a layer override on the ray trace shadow flag (assuming you are using ray traced shadows, if you're using depth map, the same applies but to the 'Use Depth Map Shadows' flag), un-flag the check box.



Repeat this step for every light you have casting a shadow and finally our diffuseLayer is complete.

I went ahead and rendered my frame. You can choose to render each frame and save it off after you finish each render layer, or, at the end you can just batch render all of the frames at once, which I'll explain how to do later in the tutorial.

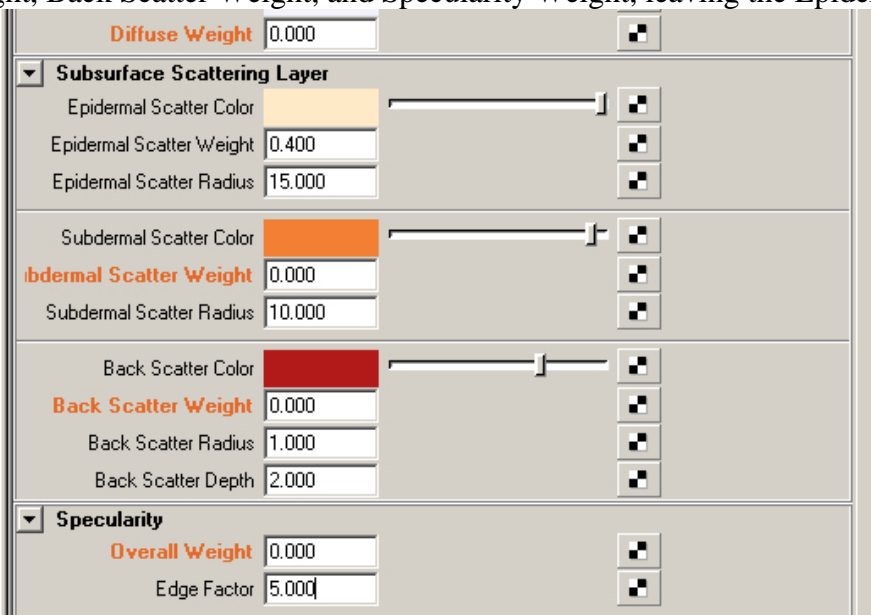


Note: You'll notice that even though shadow casting has been turned off, there is still some shadowing on our diffuse layer. More advanced users, if you know how to create a render layer for ambient color and light passes to control the diffuse shading, that is advised, otherwise this will work for the purposes of this demonstration.

Remember back in grade school when kids used to say 'Hey, your epidermis is showing, your epidermis is showing, ha-ha-ha!' ... well... in a few seconds that's ALL that'll be showing. *ba-dum crash* Alright, well, enough with the terrible attempts at reminiscent humor, back to work! Right now your epidermal layer is empty; exactly like you did with your diffuse layer, select your lights, and geometry and right click on your epidermalLayer and 'add selected objects'. Just like you had with your diffuse layer, you're going to go into your hypershade, and bring up your subsurface shader properties.

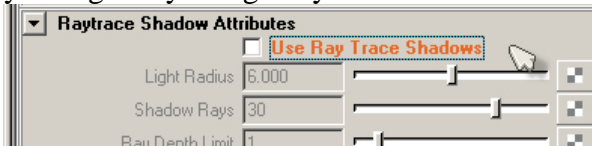
For this render layer, what we want to do is isolate our epidermal scatter color, and hide all the other layers. How do we do that you ask? Layer overrides! Except this time, you're going to want to create a layer override on Diffuse Weight, Subdermal Scatter

Weight, Back Scatter Weight, and Specularity Weight, leaving the Epidermal Weight in



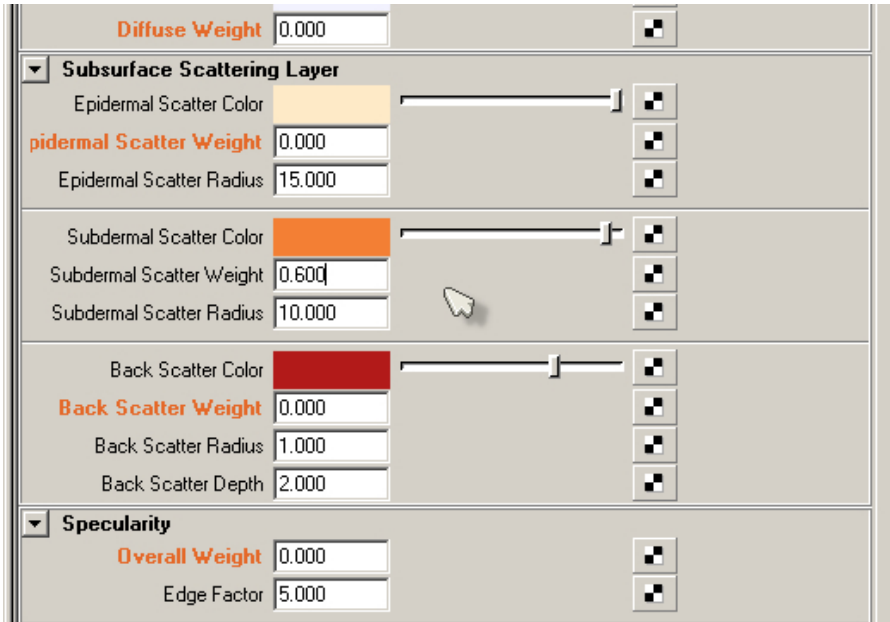
tact.

Almost done, but remember, for this layer too, you also want to turn off shadows on all of your lights by using a layer override on the 'Use XXXX Shadows' check box.

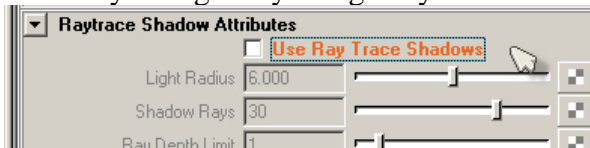


It doesn't look like much now, but it will come in handy!

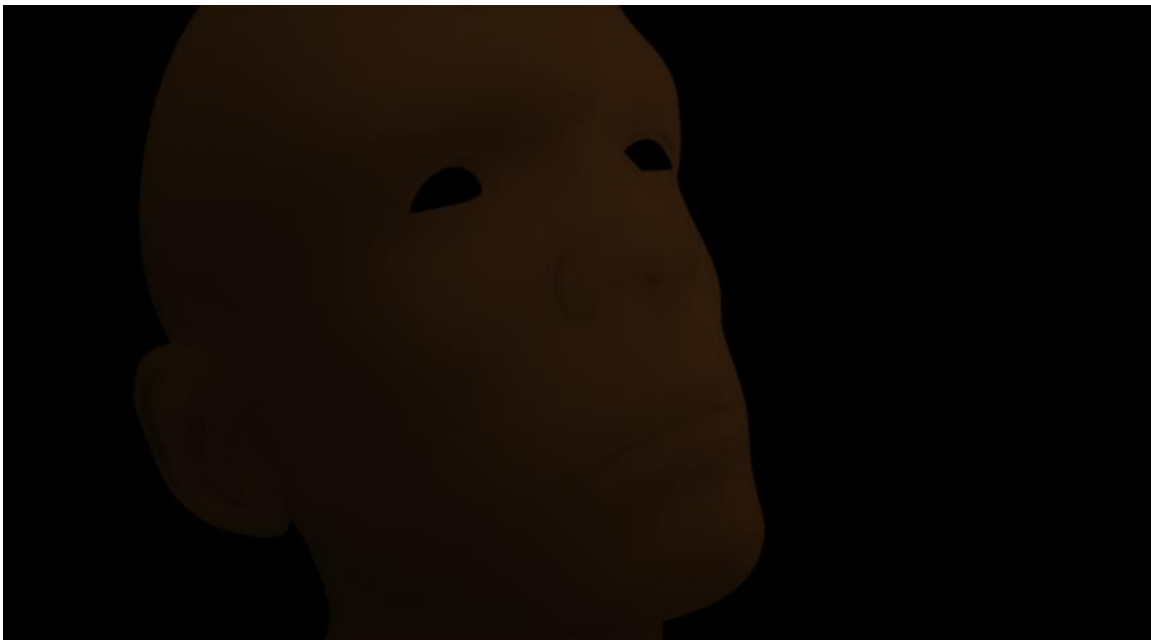
On to Subdermal. Exactly as you have previously, add in your geometry and lights to the subdermalLayer. On this layer, you're going to want layer overrides in the same manner for: Diffuse Weight, Epidermal Weight, Back Scatter Weight, and Specularity Weight. As illustrated below.



And once again, turn off shadows on all of your lights by using a layer override on the

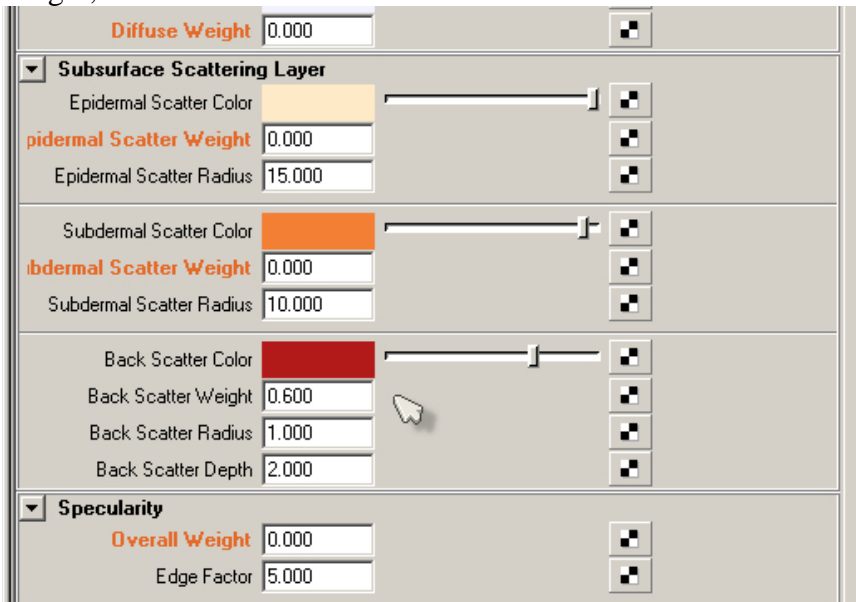


'Use XXXX Shadows' check box.

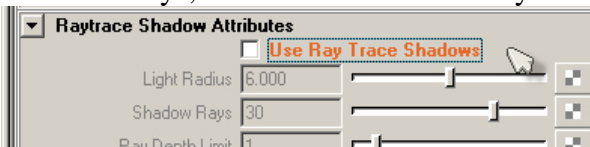


The results from our subdermal isolation.

We're getting close now! On to our backscatterLayer. Once again, copy your geometry and lights into the backscatterLayer like you have with the previous layers. And as you have with the past few layers, you're going to use layer overrides to set the following weights to 0: Diffuse Weight, Epidermal Weight, Subdermal Weight, and Specularity weight, as illustrated below.



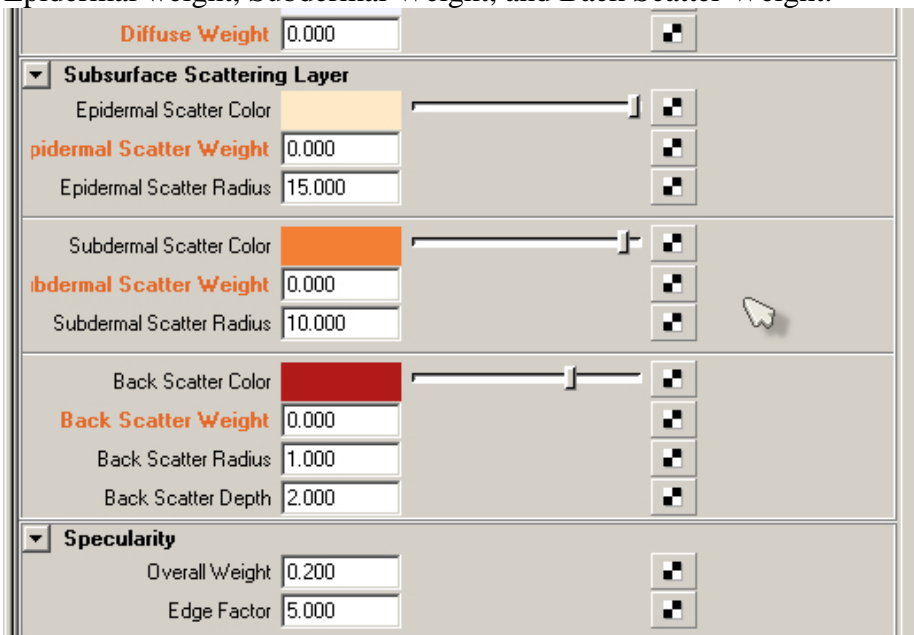
And as always, turn the shadows off on your lights. Via the layer override method.





And the results of backscatter. Note, that since the backscatter is much more subtle in our subsurface settings, it is much less prominent than the other render layers, if you look closely, you'll see this is not a black frame.

Moving right along, for your specularLayer, things get a bit more interesting. First, go ahead and import your geometry and lights like you have for all your other render layers. Open up your subsurface shader properties, and this time, you're going to create layer overrides using the same process as before for the following properties: Diffuse Weight, Epidermal weight, Subdermal Weight, and Back Scatter Weight.



Now, where this can get a little trickier, is the fact that you're dealing with two different specular controls; a Primary and Secondary specular. Now, for our more advanced users, if you wanted, you could set the Primary specular color to a pure red, and the secondary specular color to either Green/Blue, and then, isolate the red and green/blue channels to pull both speculars from the one render, then, you would color correct the now split speculars and layer them on with the rest of the stuff. For the purposes of this tutorial, however, I'm going to leave them at their defined color, and leave them as together on one render. However, using the same methods I've been going over in the tutorial, if you wanted to control both of the speculars separately, you could make a second specularLayer and use your layer overrides on the following values per layer to control whether or not they show up.



But as I said, for the purposes of this demonstration, I'm going to leave them alone and just render them both in one pass.

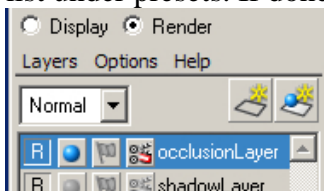
Remember to turn off your shadows!



Only two layers left! Occlusion and Shadow. These two are a little different, so now would be a good time to lift your head up and wipe the dribble off :p.

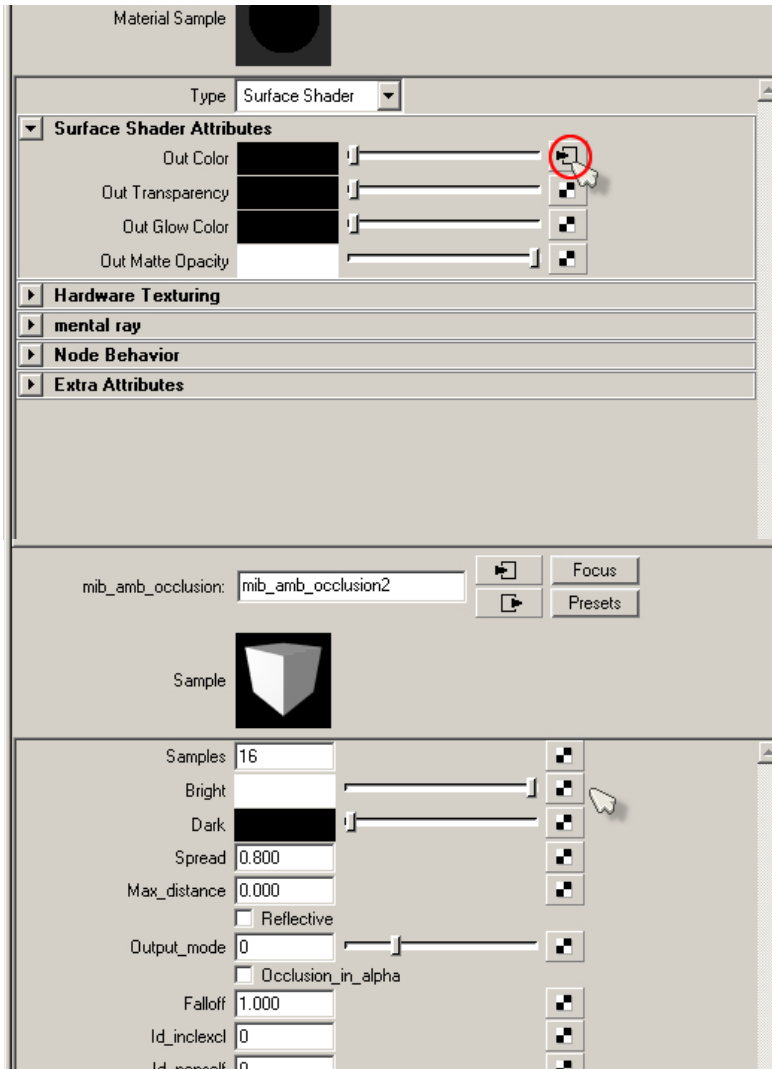
This time, just select your geometry, no lights, and add them to your occlusionLayer (you should be a pro at this by now). To reiterate, there should be no lights in your occlusionLayer.

Now, those of you who would consider yourself a more advanced user, I would advise not using the maya default presets, and instead manually drop a mib_ambient_occlusion mental ray texture into a surface shader out color and then selectively adding that shader to the objects you want to calculate occlusion on. But for the purposes of this tutorial, I am going to use the default maya preset for ambient occlusion. So, what you're going to do it go to your render layers dialogue, and select your occlusionLayer layer. Hold down right click on your occlusionLayer, hover over presets, and select 'Occlusion' from the list under presets. If done correctly, your layer will get a little blue ball icon like this.



What this will do is automatically create a layer override shader for use on every object in this layer and will apply the mental ray ambient occlusion shader. Note, this will wipe out any other shaders you have on this layer, because it is a forced material override for the entire layer. To get rid of the material override for occlusion, hold down right click on your occlusionLayer and select 'Remove Material Override'. This will not delete the surface shader it created for the ambient occlusion, that will still be in your Materials tab in your hypershade, however, it will allow you to add additional shaders into the layer if you so desire. Clicking on the little blue ball will bring up the properties of the

surfaceShader it automatically created for you. To modify your actual ambient occlusion settings, click on the connection link at Out Color, and you there you have it.



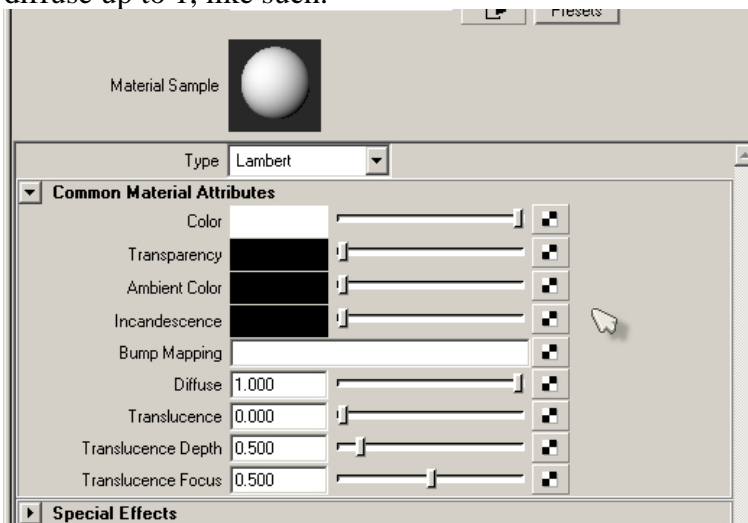
You'll want to boost up your samples to a respectable number to get rid of noise that will appear, depending on the scale of your model, usually 512-1024 samples are enough to get rid of most noticeable noise. I'm not going to get into the other attributes of occlusion in this tutorial, but your results should look something kind of like this.



The occlusion is a little harsh, but you get the concept.

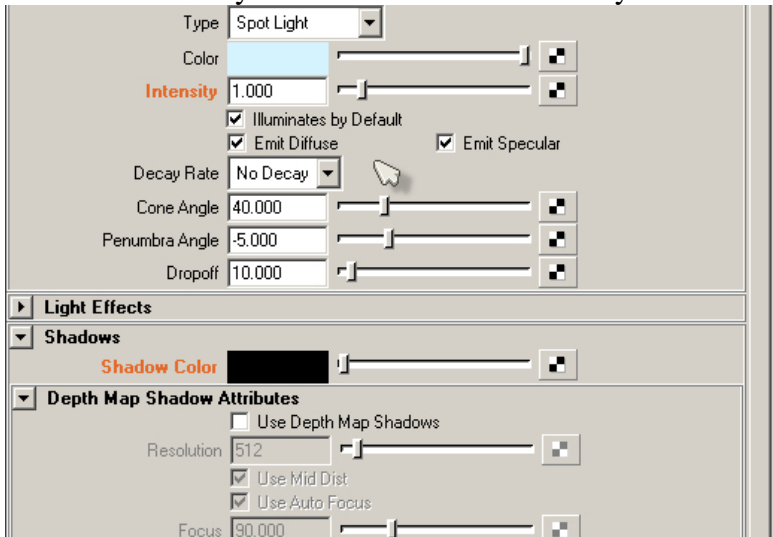
And finally, our tutorial is winding down with our shadow layer! Crank up those rays and lets make this baby pop. This one is a little trickier yet, and is one of many methods of isolating shadows, but, it works pretty well.

First, add in to your shadowLayer your geometry and only the lights that are actually casting shadows. This time, leave your shadows turned ON, this is the whole purpose of this render layer. Then what you're going to want to do is to create a Maya lambert shader. Open the properties of this lambert and change the color to pure white, and the diffuse up to 1, like such.



Once you've created your lambert, drag this onto your geometry in the shadowLayer to replace the current shader.

Now comes the cool part. Open up your light that's casting the shadow, and you're going to create two layer overrides. One on intensity and one on shadow color, like such.



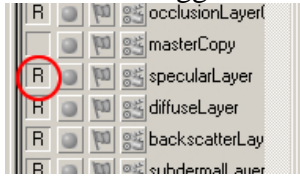
You're going to want to set your intensity to 0, and your shadow color to pure red (1,0,0-RGB). What this will create is a scenario like this. (note, in my scene I have two lights that are casting shadows)



What you're seeing here is essentially a 'mask' for the cast shadows on my object. To actually use this in compositing, in a program like Shake you can simply perform a 'reorder' on this, isolate the red channel (rrrr), and use the result as a mask for a color correct at the end of the node tree. Use the color correct to darken the area in the mask.

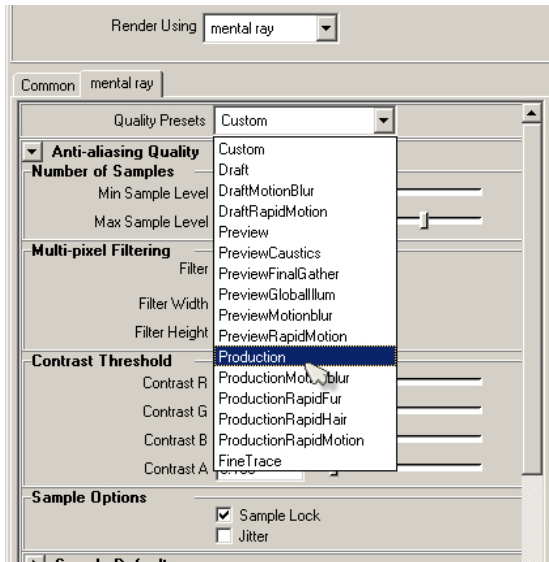
So now we have all of our render layers set up and ready to go. If you have been rendering out each frame step by step in the tutorial, you can just jump right into compositing them together. If not, lets go ahead and batch render all of our frames out.

What you'll want to do first, is make sure all of your render layers you actually want to render are flagged with the letter 'R' in the render layers dialogue, like such.

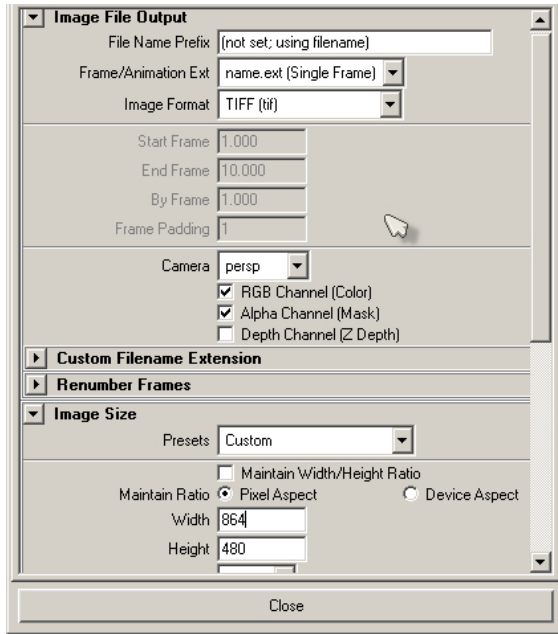


Left clicking on this little R will turn the 'render' flag on and off, basically, if the R is turned on, in a bitch, the layer will render with the scene. If the R is turned off, as you might have guessed, the layer will not render when you batch. So you see how in the image above I have the render flag turned 'off' for my masterCopy? That's because I don't want to render my beauty pass again with my layers, I just want my layers. Remember to unflag masterLayer as well.

Now that's done with. You're going to 'set' your project to your desired working folder. If you don't know how to set your project, go to File > Project > Set, then choose the folder where you want all of your output to be spit. Once your project is set, go into your render globals, change the 'Quality Presets' under the mental ray tab to production quality.



Then go back to the 'Common' tab and set your desired filename prefix, change the dropdown for frame ext. depending whether or not you have animated frames, change your image format to whatever it is you desire (usually tiffs, tga, and iff works nicely for this) and finally, set your rendered image size and close the dialogue. (Note: if you're a more advanced user and want to change around your settings a bit for optimization, that would be advised).



Now, switch your menu options in the top left from whatever they are down to Rendering



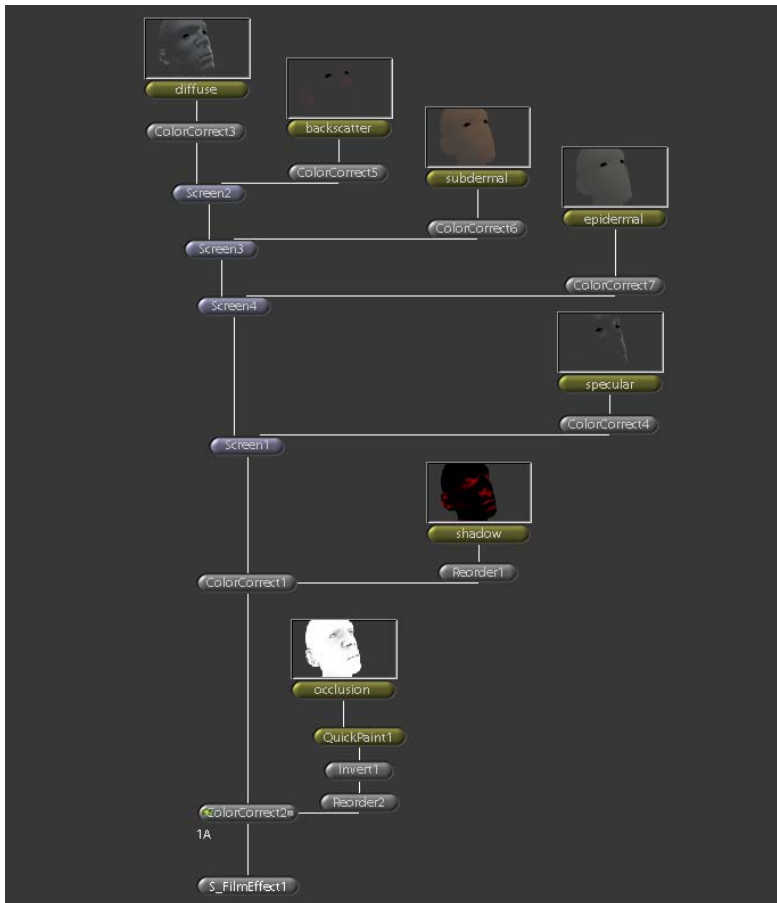
Click on the 'Render' menu at the top, and finally, on Batch Render.

This process will take some time, as it is now going to render out each one of your render layers and spit their output to your 'set' project directory. So go grab a cup of coffee, do some jumping jacks, whatever you want, until this is done rendering out. When the batch render is complete it'll tell you down in the bottom right corner on your command output.

-- Entering hard hat zone --

From here the tutorial is going to get a little looser. Since there are so many different ways to accomplish this composite, I'm going to explain my methods, but note, these are not the only ways this can be done. For my composite, I'm going to be using Shake, because it allows me a great deal more control on layering and color correcting my layers together. Though, most of the same methods are doable with Photoshop as well. Nearly everything with the exception of the occlusion and shadow layer is just screened on top of one another as you'll see in the screenshot below. For the shadow and occlusion layers, my method for integration was different from all my other color layers. For the shadow layer, as I explained earlier, the red was acting as a 'mask' for the shadow, I

reordered the shadow to isolate the red channel to act as my alpha for a color correct on top of all the layers up to this point. This way, we can add in color to our shadow by tweaking the gain on the color correct node. Likewise with occlusion; however, for this example what I've done is inverted the occlusion layer, so that the black is white and white is black. Since white acts as the 'pass-through' for a mask, this will allow us to use the contact shadow produced by occlusion as a mask for a color correction, so once again, we can use the gain on the color correct to add color into the occlusion as opposed to simply multiplying a black occlusion layer on top of our object. The quickpaint on the occlusion was used to fix some of the values on the occlusion since I didn't tweak my occlusion shader settings before rendering. And to note, the s_FilmEffect is a macro via GenArts plugins for shake used to control some color corrections at the end of the chain.



As you'll notice, I've dropped all my rendered layers into the shake workflow and placed them in order that they should be layered. Diffuse on the bottom, backscatter on top of that, subdermal on top of that, epidermal on top of that, then we have our specular our shadow, and finally our occlusion. For each of the color layers, you'll notice I've added a color correct node, this essentially allows me to individually control the color and value of each layer; this can be useful in several situations. Lets say I'm trying to integrate my character to a painting I've made. I fix up the values in maya as close as I can, and things are looking right. I render it out, try to composite it in there, but I notice that the color of my specular is off just a hair, or my epidermal is too desaturated, or my shadow is the

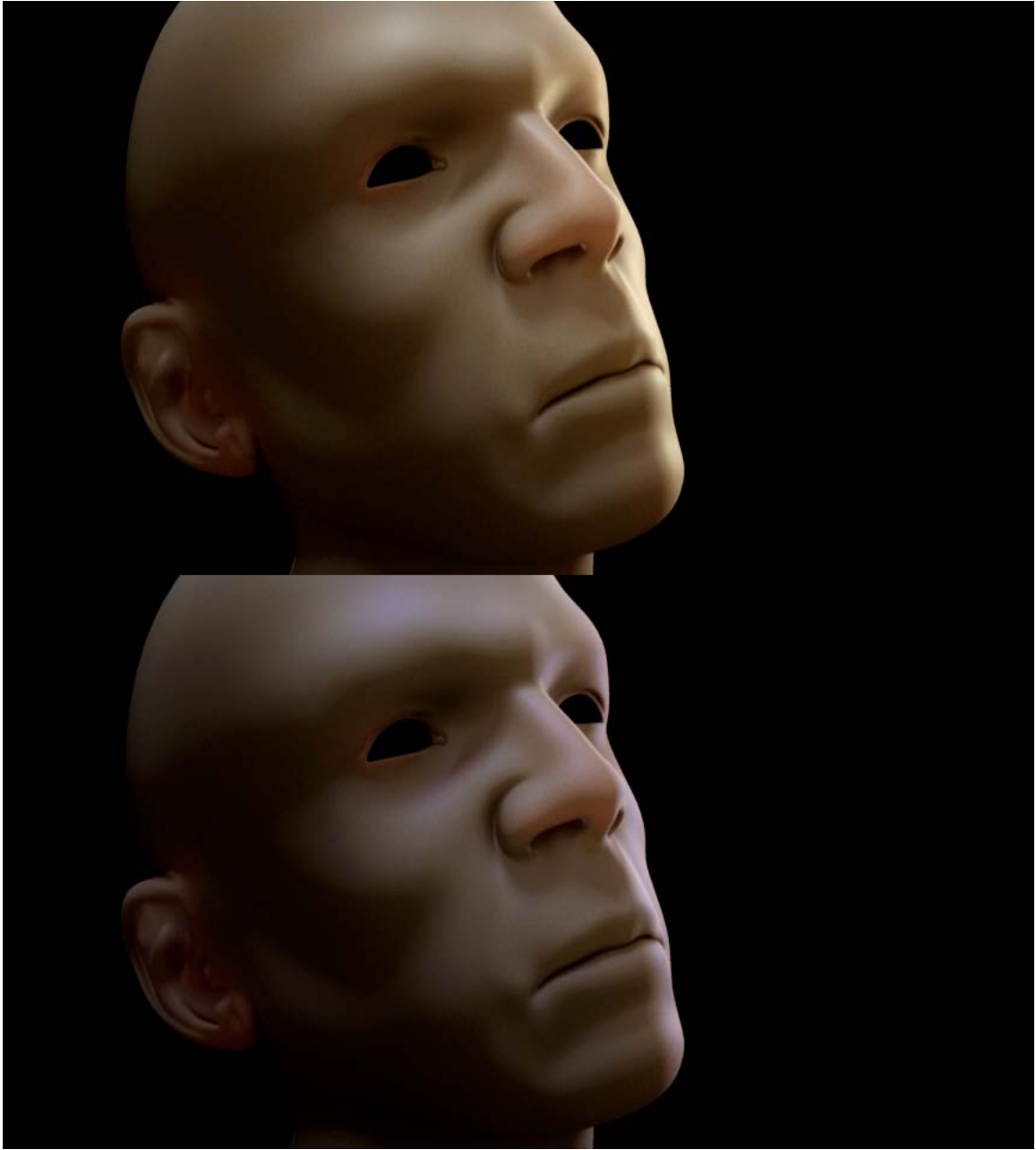
wrong color, whatever the case may be. Instead of now having to tweak a value, re-render the frame, match it up, notice it's still off, tweak some number, re-render a frame, etc. etc. I can simply tweak the color values of the epidermal or specular in compositing in real time, without having to wait on renders. This in the end is the overall advantage to rendering all of these attributes in their own layer, superior control to tweaks to color, value, and such without the grief of having to continually re-render my frames.

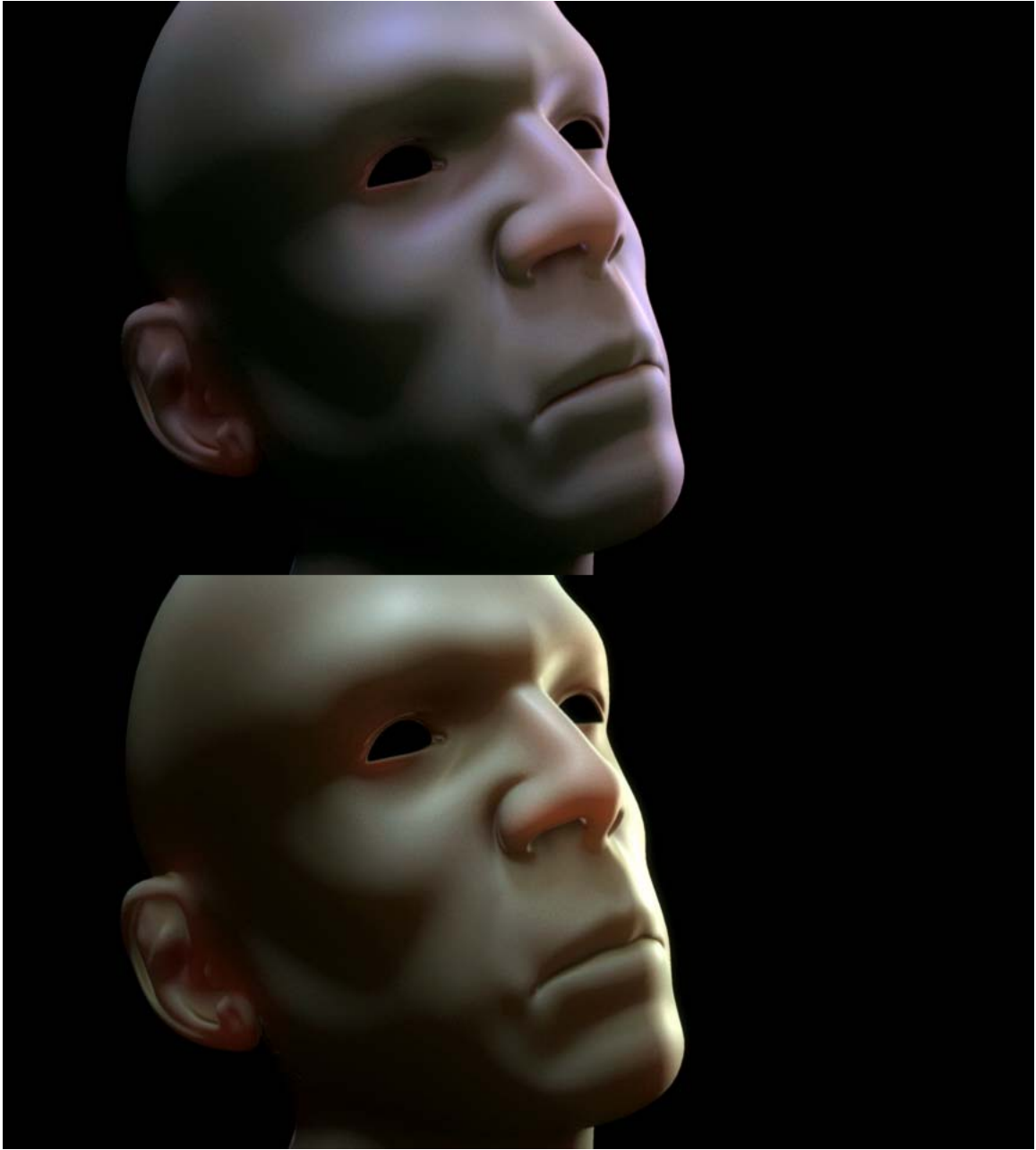
Now, in certain situations, you will still need to re-render your frames. For example, if your light position is wrong, or, overall the depth of your backscatter is too low, your shadow samples are too sharp, etc, then there are a few tweaks you 'can' do in compositing to try and compensate, but it may require you to end up re-rendering your layers anyways.

So this brings us to the conclusion of this very long 'little' tutorial I had intended on writing up. If you've made it this far, thanks for sticking through it! Here is a view of what our composite looks like with the values just layered on top of one another.



And here are a few examples of the power of using the render layers to quickly adjust the color and values of these layers to change the overall appearance and mood of the model.







Thanks again!
If you have any questions, feel free to ask away.