Ideas For A Procedural Skin Shader

by skin2279
updated 24-02-2022

It seems easy to get hold of a good-looking skin material: there are plenty of examples of such in use here on DeviantART. But they all use *image textures*, which as far as I know are obtained by photographic scanning of actual human skin.

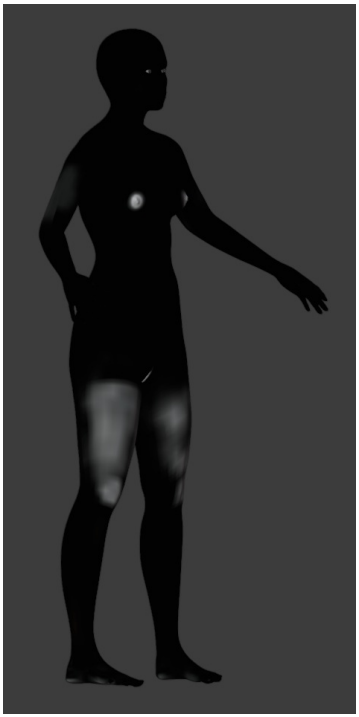
Myself, I want more variety than I can get from such canned textures. What if I want to change the skin colour? Regulate the amount of freckles? It's difficult to do that with a scanned image. Wouldn't it be nice to be able to just tweak a knob and get a whole range of colours?

So what I have been trying to do is create a *procedural* texture — that is, one made out of mathematical functions, rather than scanned images. One important thing is, you want the texture to vary somewhat over the different parts of the body, so it is shinier here, maybe darker there, and so on. I achieve this by making use of Blender's **vertex colour/vertex paint** layer functionality. I define multiple vertex colour layers, and use them, not to colour the material, but to provide controls over characteristics of the material.

You could argue that the same effect could be achieved with image textures. But proper setup of image textures requires that the mesh be UV-mapped, and I must admit I have been a bit lazy about that. Vertex colours are automatically associated with the vertices of the mesh, without requiring a separate mapping to be defined. And since there is a only a single value in each vertex colour layer associated with each vertex, there is much less data involved, so files are smaller. And painting the colours in the first place takes correspondingly less work.

(Note that, while some screenshots and renders include the eyes, those have entirely separate material setups that are not part of this discussion.)

Here are some screen shots of what the various vertex colour layers look like on top of a MakeHuman base mesh. Notice each "colour" layer is created as purely greyscale; remember they are only used as control variables, the actual colours in the material coming from elsewhere. Full black in each vertex layer represents a factor of 0, while full white represents a factor of 1.



Vertex colour layer “Skin Dark Mask”

The “Skin Dark Mask” is used to specify parts of the body that need a darker version of the basic skin colour. This can show tanning effects, and also includes the nipples.

Vertex colour layer “Skin Gloss Factor”

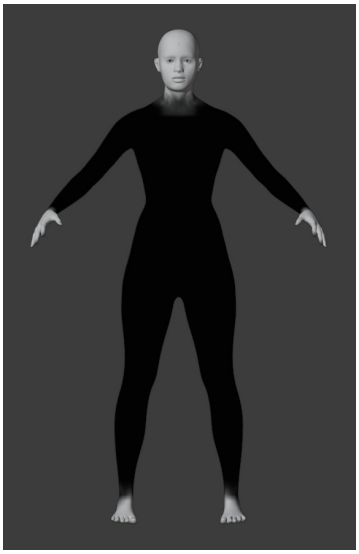
The “Skin Gloss Factor” is used to make parts of the mesh look shinier.

Vertex colour layer “Palm/Sole Mask”

The “Palm/Sole Mask” is used to give a different colour to the palms of the hands and the soles of the feet. This needs to be controlled separately, to give a greater contrast when the skin is darker.

Vertex colour layer “Face Flush Mask”

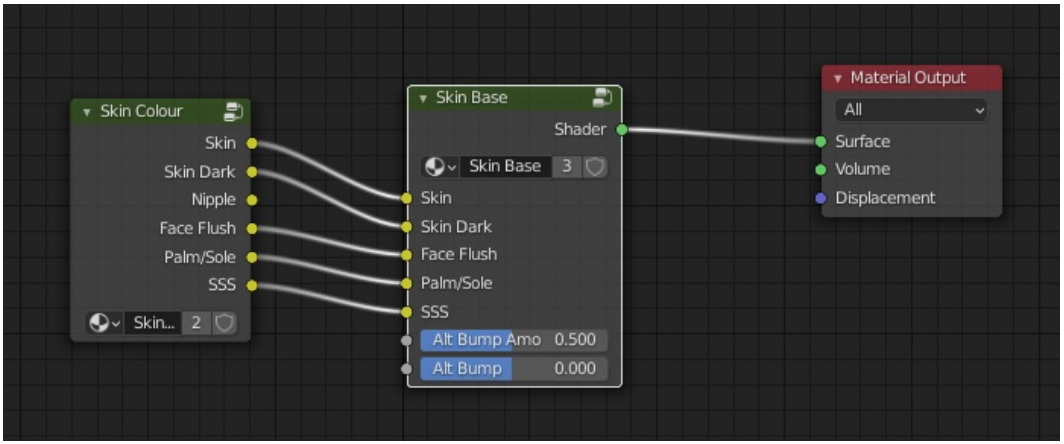
The “Face Flush Mask” is used to give a different, typically redder/darker colour to parts of the face.



Vertex colour layer “Skin Grain Mask”

The “Skin Grain Mask” is used to control the fineness of the skin patterning.

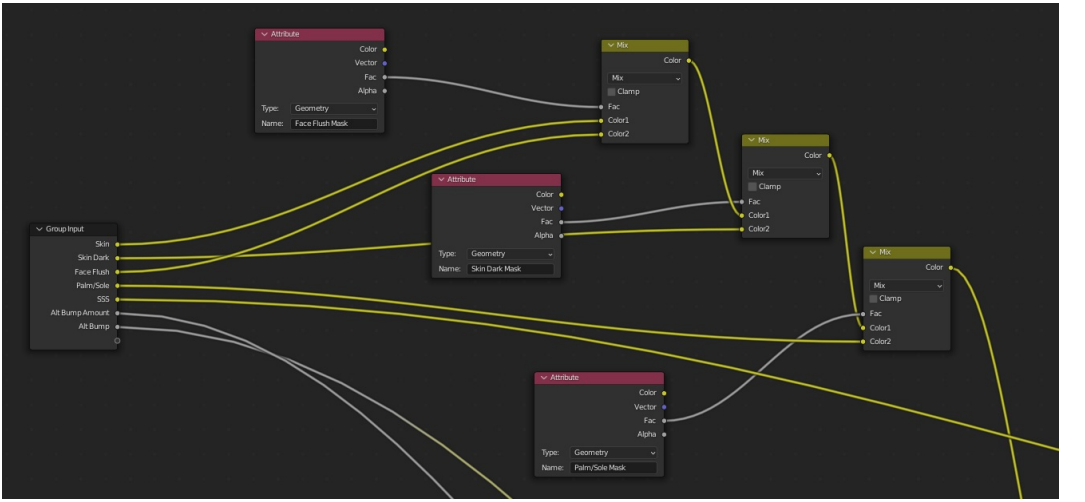
Here is what the overall skin material looks like in Blender’s Shader Editor:



Overall Skin Shader

Deceptively simple, isn’t it? OK, the complexities are hidden behind a pair of node groups. The one on the left just defines the colour scheme, while the one on the right is where all the cool stuff happens. By separating the two out in this way, I can easily change the colours, just as I mentioned at the beginning.

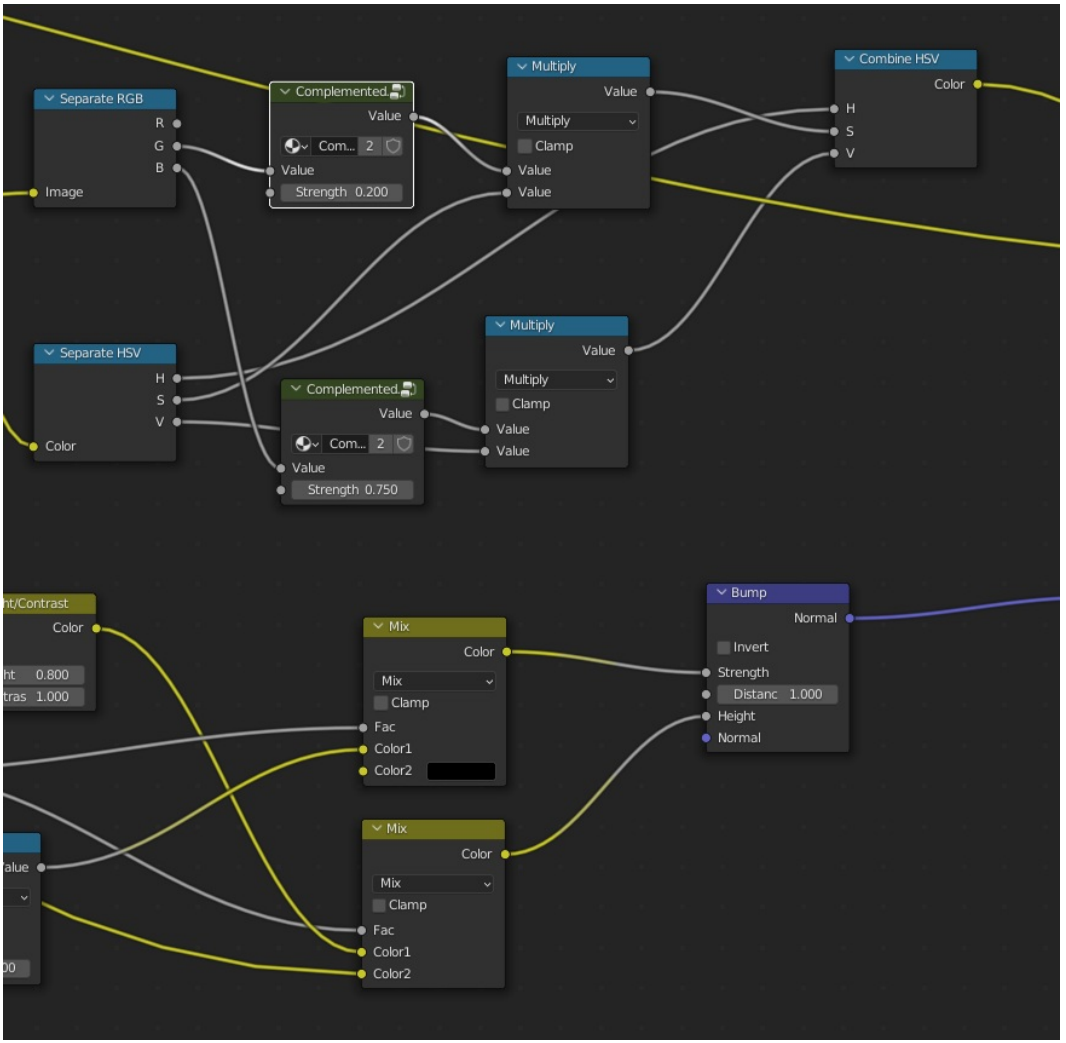
So let’s take the construction of the Skin Base node group step by step. First of all, mix the input colours together, as modulated by the appropriate vertex colour layers:



Colour Mixing In The Skin Shader

Just concentrate on the top four group inputs for now; don't worry about the other lines disappearing off the edge of the diagram. Their function will be explained later. See how the three vertex colour layers whose names appear are used as factors to control the colour mixing.

So what happens to the colour coming out of that righthandmost Mix node? It gets subject to the application of a pattern, so that the skin is no longer just a featureless colour (or gradation of colours). There are several possibilities for a suitable pattern function; the one I use here is based on the Voronoi texture function built into Blender.



Applying A Pattern To The Colour

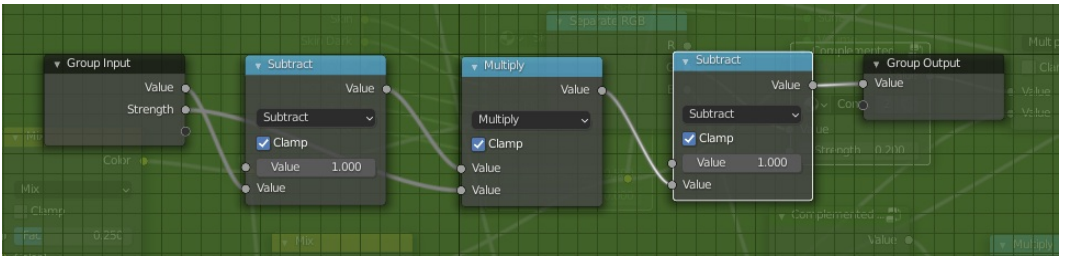
There are two parts to this portion of the node tree: the upper part creates the colouring, while the lower part is the bump map that gives a mottled effect to the skin, so it is not perfectly smooth. Let's do the lower part first.

There are two Mix nodes here: the upper one modulates the strength of the mottling effect, while the lower one produces the actual pattern. The inputs to both will become clearer below.

(The first of these two Mix nodes could be replaced with a Math node that subtracts its input from 1, since this input is actually a real rather than a colour, and the second colour being mixed with it is black, which is equivalent to zero. But never mind...)

To understand the upper half of this picture, start with the "Separate HSV" node on the left; this is where the combined skin colour comes in from the previous section. Something happens to the components of the colour here, before they get recombined into the "Combine HSV" node at the right. And what exactly is this "something" that happens? They get multiplied by factors coming from the

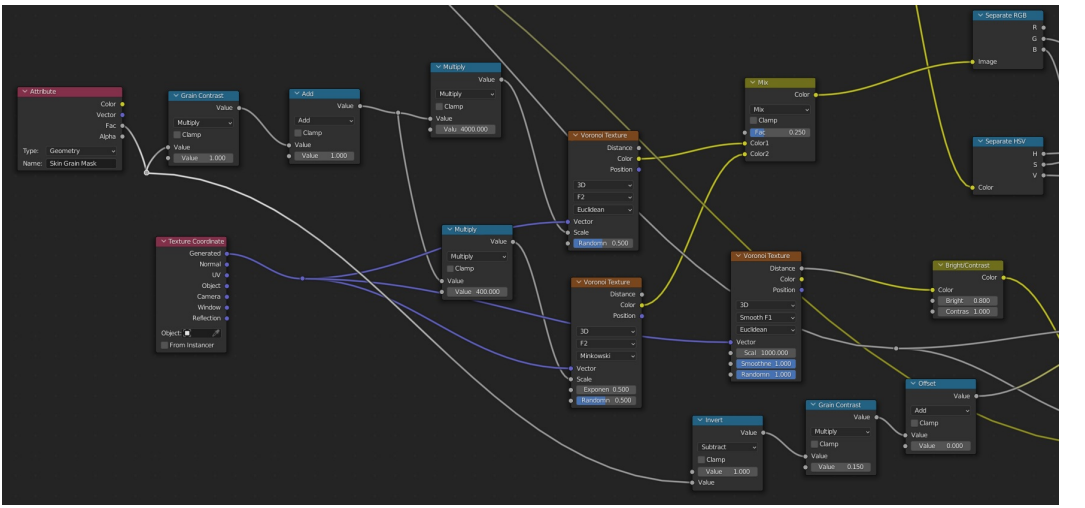
"Separate RGB" node, the strength of these being controlled by some mysterious node group called "Complemented Attenuate". What does this do?



What "Complemented Attenuate" Does

What it does is kind of an inverted-multiply operation. A straight multiply by a factor in the range 0 to 1 would produce a result close to 0 at the low end, and close to the original value at the high end. Instead, this inverted version of the multiply produces a result close to the original at the low end of values for the attenuation factor, and close to 1 at the high end. The idea is that the saturation and lightness of the original colour are maximum values that are *reduced* by the patterning, to produce darker and greyer regions, and the amount of reduction is controlled by the attenuation factors.

So what is the actual patterning that is applied?



Generating The Skin Patterning

Here, the last part of the recipe is revealed. The "Separate HSV" node at the right is the same one fed from the combined skin colour. Here we see a number of Voronoi textures at play: first, a mix of two of them, at somewhat different scales, with their relative proportions controlled by a Mix node, to produce the skin patterning effect. (Does it matter that one Voronoi uses "Euclidean" distance while the other uses "Minkowski"? Very likely not. But that's where my experimentation ended up...) A third one is used to produce a bump map for adding a texture to the skin.

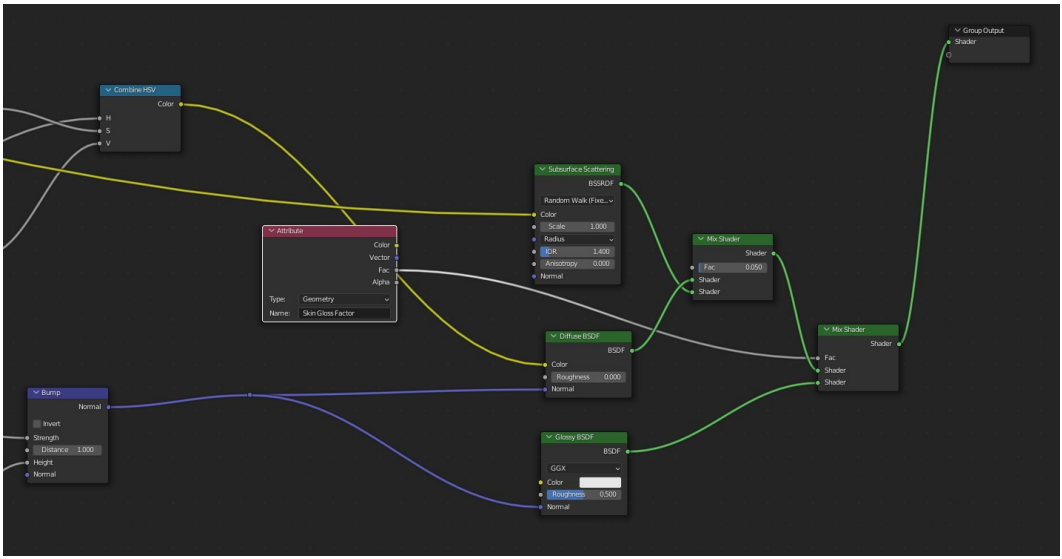
Another Mix node brings in the "Alt Bump Amount" and "Alt Bump" values from the Group Input; these can be used to add extra bumpiness, for example as part of the nipple material described below. This is mixed with the third Voronoi texture node according to those two Mix RGB nodes we previously saw, modulated by the Skin Grain Mask vertex layer, to produce the bumpiness of the skin.

What does "Separate RGB" do? The Voronoi node calculates a full RGB colour, not just a single real number. So rather than let this go to waste, I split out the individual components, and use them to modulate the saturation and lightness of the skin colour. The third component goes unused, since I decided that modulating the hue was not a desirable effect.

Mixing two different Voronoi textures at different scales adds some variety to the patterning. You can observe the effect of the separate components individually by playing with the Mix factor: set this to zero to see the upper component on its own, or to 1 so all the effect comes from the lower component.

Note also how the texture intensity is modulated by the "Skin Grain Mask" vertex colour layer; this is used to make for a subtler pattern on the face, hands and feet, which seems more realistic to me. (Previously I used this to modulate the *scale* of the pattern rather than its intensity, but for some reason that did not work the way I expected.) The use of "Generated" rather than "UV" as the coordinate source works quite adequately for a procedural texture in this case.

Putting it all together: It only remains to show the final part of the node tree.

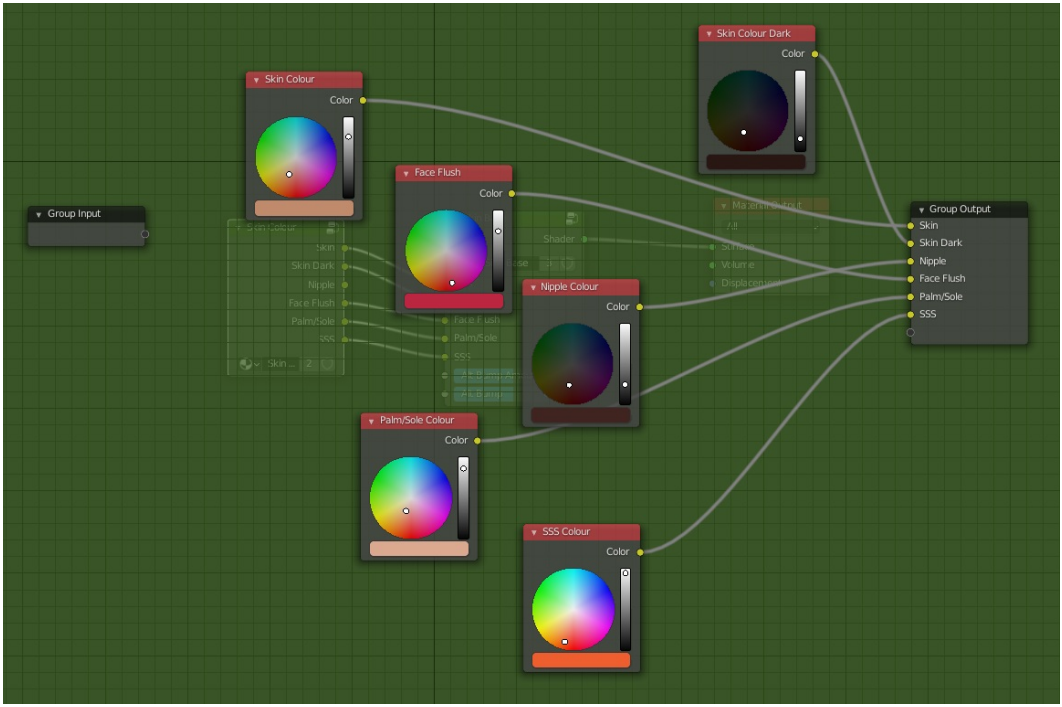


Skin Shader—The Final Output

The "Combine HSV", as we already saw, reconstructs the skin colour after the application of the pattern. That long line disappearing off to the upper left connects to the "SSS" group input in the first section of the node tree, to use as the colour for the Subsurface Scattering shader. In earlier versions of my skin material, I had a whole other vertex colour layer controlling the mix factor between the SSS and Diffuse shaders; in the end I felt this wasn't important, so you see just a constant factor used here instead. Some amount of subsurface scattering is important to give skin that subtle, lifelike translucent quality; without it, the skin looks like hard, lifeless clay.

And in the lower part, we see the mottling being applied to the "Normal" input of both diffuse and glossy shaders, to make the skin bumpy. Should I have done the subsurface shader as well? I didn't bother because, well, it is supposed to be the effect of a layer *under* the surface of the skin, after all. Does it make a difference? Try it and see.

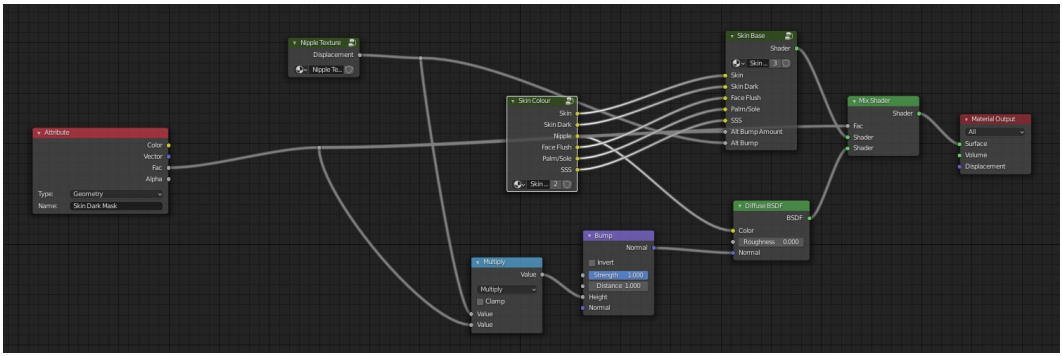
I suppose I should show the insides of that "Skin Colours" node group.



Skin Colours Node Group

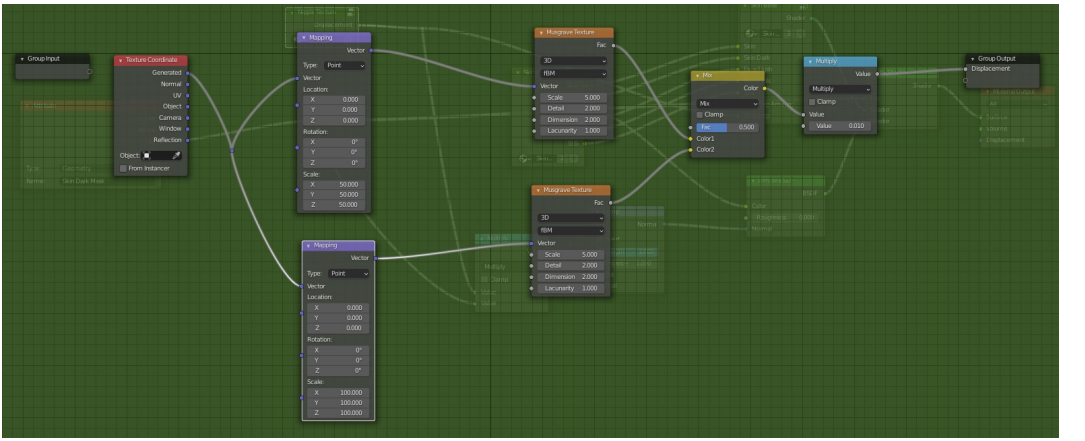
As you can see, there's nothing really magical here; it's just a collection of RGB nodes defining the colour scheme. This allows the base skin material and for the nipple material to share a common colour scheme, so changes only need to be made in one place.

The Nipple Shader



Overall Nipple Shader

Here you can see the same "Skin Base" and "Skin Colour" node groups as used in the skin material, with some extra embellishments: the "Nipple Texture" node group adds some extra wrinkliness, modulated by the Skin Dark Mask. If you don't want this, then by all means use the same Skin material for the nipples. You will still get nipple darkening, courtesy of the Skin Dark Mask.

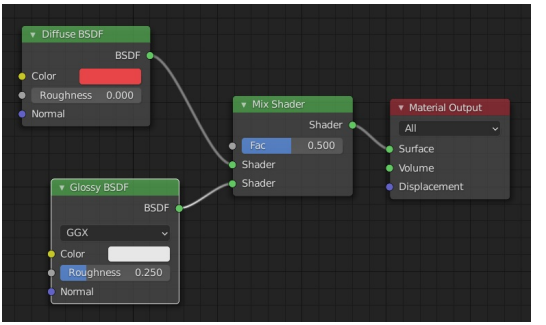


Nipple Texture Node Group

Here is the nipple texture I've been using. As I said, this may not be to your taste, so change it or even drop it entirely. Up to you.

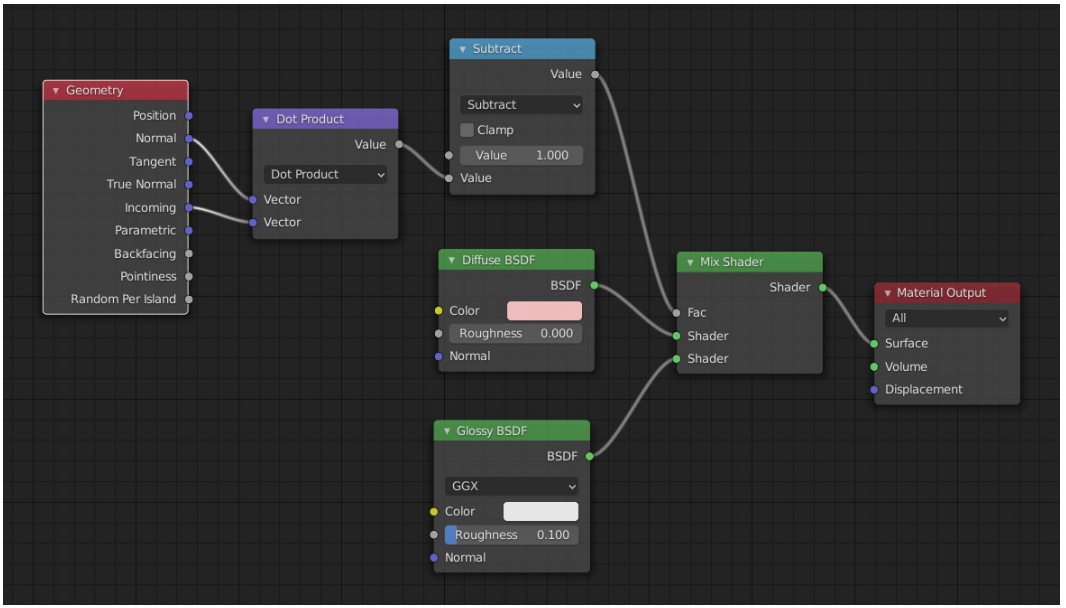
Other Materials

I use a separate material for the moist red parts inside the eyelids, and some other places. I call it "Dermis Internal", and it's nothing too complicated:



Dermis Internal Shader

For the unpainted fingernails and toenails, I wanted to simulate the effect of translucency, without actually having the nails be separate objects or protrusions on top of the underlying skin.



Unpainted Nails Shader

The “Dot Product” gives a value close to 1 when the surface is seen face-on, falling to 0 as it is seen side-on. Subtract this from 1 (or alternatively, just flip the Mix Shader inputs around), and you get a dull pink appearance in the first case, changing to a shiny white in the second case. I like the pearlescent effect this gives.

The Results

And after all that work, here are some of the colour variations I have been able to achieve with the resulting materials.

Medium Skin

Dark Skin

Fair Skin

Also-Dark Skin

Further Work

Of course this is nowhere near a properly realistic skin material. Think about the wrinkles on the backs of the knuckles, lines on the palm, wrinkles on the sole of the foot, don't forget actual pores and hairs ... the list of missing details seems endless. Could somebody use this as a starting point and take things to the next level? Maybe.

Or maybe we all just go back to using image textures...