Procedural Skin Shaders using RenderMan

Colm Doherty (e9089949)
MSc Computer Animation
NCCA Bournemouth University

Chapter 1 - Abstract

This report sets out the design, implementation and results of the creation of photo-realistic procedural shader's that represents the skin on a human hand. The procedural shader(s) were created using Pixar's RenderMan shading language. This report indicates the design approaches that were taken, as well as the implementation,

results and analysis. The procedural shader were planned to be outputted onto a polygon mesh model using Alias' Maya modelling package if time permitted. Pixar's RenderMan Artist Tools (RAT) were used as a pipeline between Alias's Maya to Pixar's RenderMan.

Chapter 2 – Contents

Chapter 3 – Introduction

The purpose of this project was to create procedural shader's to represent the various aspects of the skin surrounding the human hand. The human skin has a very complex make-up that is difficult to define for any one area of the body especially the human. The skin seems to warp and deform around the digits of the bones. The detail to smoothness ratio changes constantly around the hand. The colouration from smooth tissue under the skin, cellular activity in the layers of the skin and activity with the elements on the top layer of the skin effects how the skin appears in any one area and dictates how it changes to another. In order to create procedural skin shader's to represent the human hand, these intricacies had to be taken into consideration in the design process. During the creation of the shader's, a range of issues and problems arose. Some of these issues delayed the project and as such crippled the author in being able to combine all the shader's onto one geometric model surface to represent a human hand and for sub-surface scattering to be performed. The shader's have been created and tested using PRMan. The results and tests proved successful with anti alias elements not arising. This was a key design consideration in all the shader's. An interface between Maya, a modelling tool, and PRMan,where the shader's were created and tested, was researched. This was MTOR (MayaToRenderMan), where various methods exist for bringing shader's into a Maya scene to be applied to objects, but some proved more successful than others, and all were time-consuming to someone who had not textured a complex object before.

Chapter 4 – Literature Review

A large part of realistic image synthesizing of human skin is the modeling of subsurface light transport through the skin material. Therefore a discussion of the current and past models of subsurface light transport is in order.

One of the most recent and simplest, yet very efficient models proposed, is "A practical model for subsurface light transport" by Jenson et al [JEN01]. The model allows for efficient simulation of effects such as colour bleeding within materials and diffusion of light across shadow boundaries. The model can also efficiently simulate highly scattering anisotropic (not the same in all directions) media. The model combines single scattering solution along with a dipole point source diffusion approximation for multiple scattering. In this model, Jenson et al also designed an image-based measurement technique for determining the optical properties of translucent materials, such as skin and milk.

fig. bssrdf taken from [JEN01].

The bi-directional surface scattering distribution function (BSSRDF), describes the light transport between any two rays that hit a surface. The bi-directional reflectance distribution function (BRDF), was introduced by Nicodemus [NIC77]. Most of the BRDF models describe how a ray of light which enters a surface, will leave that surface at the same position, with any subsurface transport being approximated by a lambertian component. As Jenson (JEN01) points out, this model's approximations are valid for materials such as metals, but it falls short for translucent materials such as milk or skin, where light that enters the surface can exit at a completely different position, due to large transport of the light below the surface. BRDF is criticized, as it

doesn't consider subsurface light scattering as well as not considering the blending of surface features such as colour and geometry [JEN01]. The author argues that there is not much point in having the most complex light transport algorithms, if the local material light scattering algorithms are too simple, it will just end in unconvincing realistic image synthesis [JEN01].

fig translucent teapot. taken from [JEN01]

In order to simulate full subsurface light transport, the model needs to simulate lights scattering multiple times under the surface and exiting at a different surface point to which it entered. Pharr and Hanrahan were able to simulate subsurface light transport in the paper "Monte Carlo evaluation of non-linear scattering equations for subsurface reflection" through subsurface scattering functions [PHA00]. This model allows for subsurface light transport through multiple layers such as skin, with calculations being made at each layer as the ray of light passes through. Layers calculations are described in terms of absorption coefficients, scattering coeeficients, refraction, depth of the individual layer and the Henyey-Greenstein phase function. However, this model was computationally very expensive, as was the method by Dorsey et al. . This model fully simulates the complete subsurface scattering in the paper "Rendering of wet materials", in which weathered stone was simulated. It was Stam, in the paper "Multiple scattering as a diffusion process", who solved a diffusion equation which allowed for simulating highly scattering materials such as milk and skin, in which light can scatters hundreds of times before exiting such materials. Through solving the diffusion theory, Stam had allowed for the measurement of the optical properties of highly scattering materials such as milk and skin [STA95].

Jensen et al. (JEN01), extended the diffusion theory by incorporating an image-based appearance measurement technique for measuring translucent material. This measurement method allowed for the examination of the radial reflectance profile from a beam which illuminates the sampled material. By using an expression from the diffusion theory, solved by Stam [STA95], Jensen et al. [JEN01] were able to estimate the absorption and scattering properties of the sampled materials. This measurement technique was extended from an existing methodology for measuring biological tissues, used in the medical sciences. Jensen et al. (JEN01), also further extended the

diffusion theory by incorporating the technique for exact single scattering. Please see appendix i for approximations of BSSRDF.

Chapter 5 - Human hand

5.1 – The Human Hand

fig. bone structure. taken from [WIK06a]

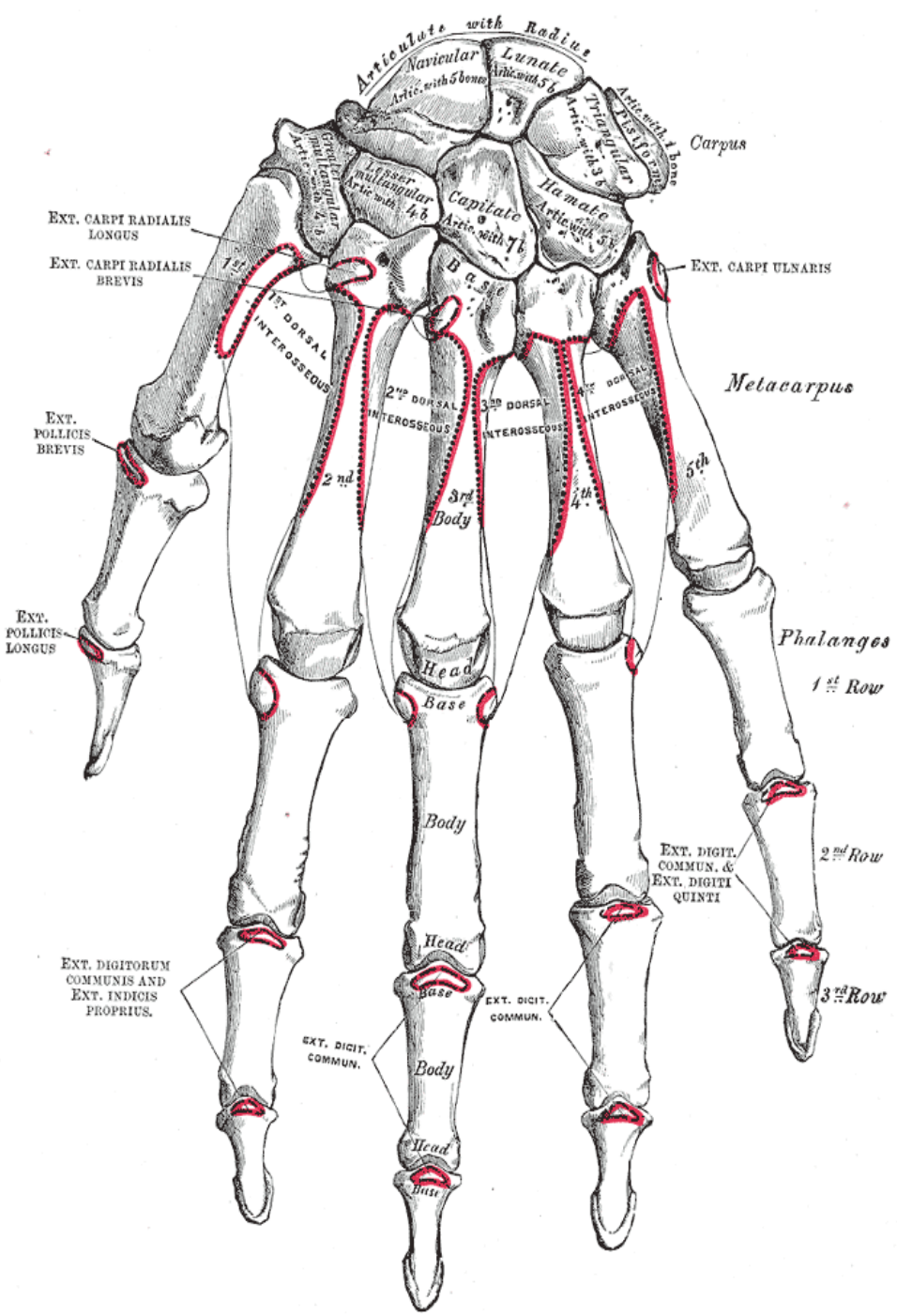
The human hand has four fingers and a thumb. The presence of an opposable thumb on the this organ, constitutes it as being a hand. Our hands are our main way of physically manipulating the environment around us, with extremely fine motor skills. As such, the human hands provide our prime sense of touch with some of the densest areas of nerves endings contained in our fingertips. Interestingly, each hand, like other paired organs, is controlled by the opposing part of the brains hemisphere. [WIK06a]

5.2 - Bone Structure

The human hand has at least 27 bones. Separated into three parts:

The wrist contains 8 carpus bones

The palm contains 5 metacarpus (metacarpals) bones, one for each digit.


Articulate with Radius
Navicular
Artic. with 5 bone
Lunate
Artic. with 5 b
Triangular
Pisiform
Artic. with 1 bone
Carpus
Greater multangular
Lesser multangular
Capitate
Artic. with 7 b
Hamate
Artic. with 5 b
Base
EXT. CARPI RADIALIS LONGUS
EXT. CARPI RADIALIS BREVIS
1st
1st DORSAL INTEROSSEOUS
2nd DORSAL INTEROSSEOUS
3rd DORSAL INTEROSSEOUS
4th DORSAL INTEROSSEOUS
EXT. CARPI ULNARIS
Metacarpus
EXT. POLLICIS BREVIS
2 nd
3rd Body
4th
5th
EXT. POLLICIS LONGUS
Phalanges
1st Row
Head
Base
Body
EXT. DIGIT. COMMUN. & EXT. DIGITI QUINTI
2nd Row
Head
Base
EXT. DIGITORUM COMMUNIS AND EXT. INDICIS PROPRIUS.
EXT. DIGIT. COMMUN.
EXT. DIGIT. COMMUN.
Body
3rd Row
Head
Base

Chapter 6 - Skin

Skin is made up of multiple layers of epithelial tissues that guard underlying muscles and organs. The skins main functions are to protect against pathogen or infectious agents as well as insulation and temperature regulating. Skin has pigmentation, or melanin, which absorb some of the potentially dangerous ultraviolet radiation in sunlight. Skin also contains DNA repair enzymes which helps to reverse the UV damage process. The pigmentation on human skin can strikingly vary amoungst different populations, and as such has led in some cases to people being classified base on the skins pigmentation colour. [WIK06b]

fig. Skin Image. authors own hand

Interestingly, the human skin is often termed as being “the largest body organ” as it covers a larger surface area then any other organ, but more importantly weighs more than any other organ constituting 15 percent of body weight.

6.1 - Skin Layers

The human skin is composed of three main layers;
Epidermis layer, providing waterproofing as well as a barrier to infection
Dermis layer, containing blood vessels, nerves, hair folicals, and other tissue

Hypodermis (subcutaneous adipose) layer, the basement membrane connecting skin to the bone and muscle structure

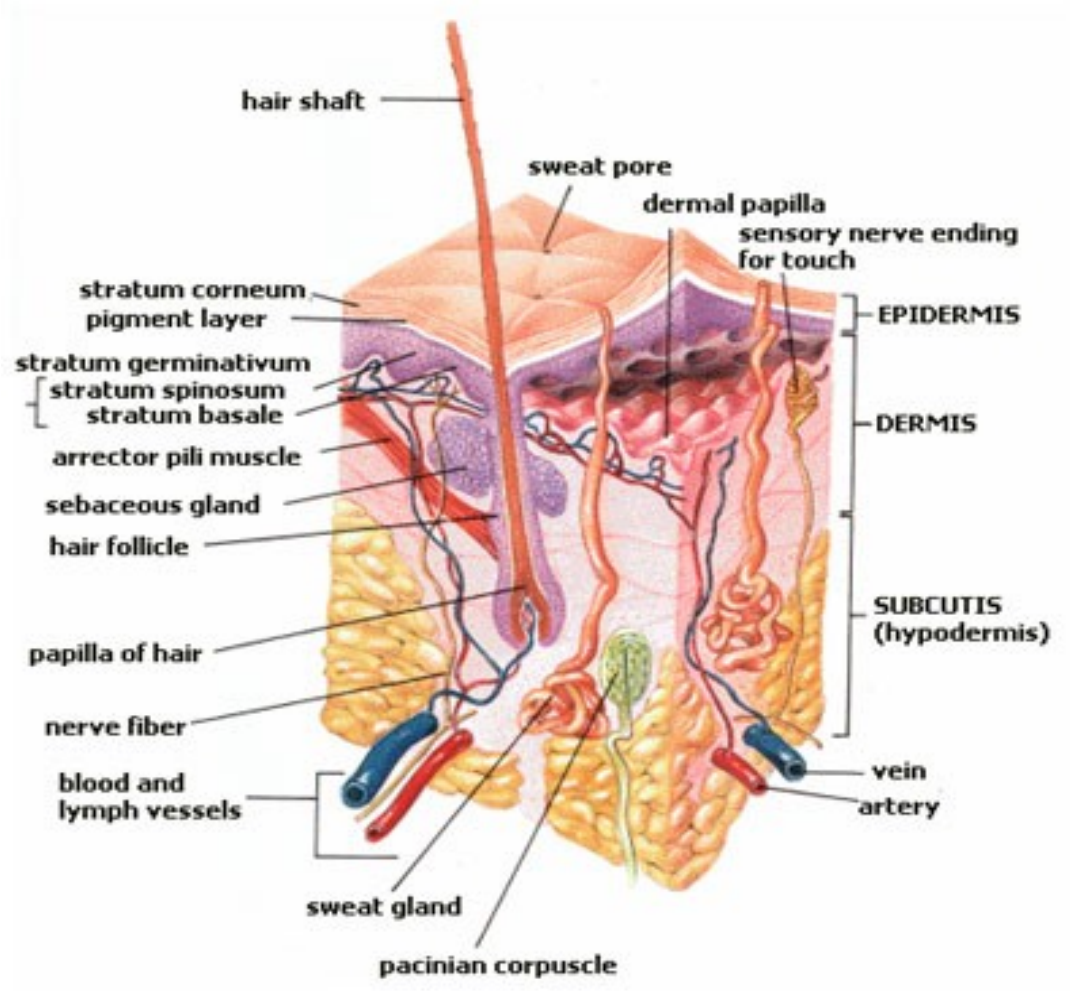


fig. Skin Structure. [WIK06b]

The epidermis layer, contains no blood vessels. The epidermis can importantly be subdivided into a further 5 layers;

- Corneum layer,
- Lucidum layer,
- Granulosum layer,
- Spinosum layer
- Basale layer.

Cells form at the innermost basale layer through mitosis, with daughter cells forming through the higher layers but eventually dieing due to isolation from blood source. Due to this, cytoplasm is released, resulting in keratin proteins inserted through layers as we go higher, ultimately reaching the upper corneum layer, resulting in a fast slough off. This process, through the layers, takes about 30 days and as is known as keratinisation. As this keratinised process moves through the layers, it is responsible for keeping water inside the body and pathogens out, providing the natural barrier to infections. [WIK06b]

The dermis layer contains blood vessels, nerves, hair folicals, smooth muscle, glands and lymphatic tissue. The connective loose tissue, areolar, has collagen, elastin and reticular fibers within. Fine erector muscles between the hair papilla and epidermis, can contract, which results in the forming of goose bumps as the hair fibre is pulled upright. Sebum is produced by sebaceous glands, used for waterproofing, antibactericidal action, and lubrication. These glands open open via a duct onto the skin by a pore.

The hypodermis layer isn't part of the skin with its purpose being to attach the skin to the bone structure and muscles, as well as providing it with nerves and blood vessels. [WIK06b]

6.2 - Structure of an anatomical Nail

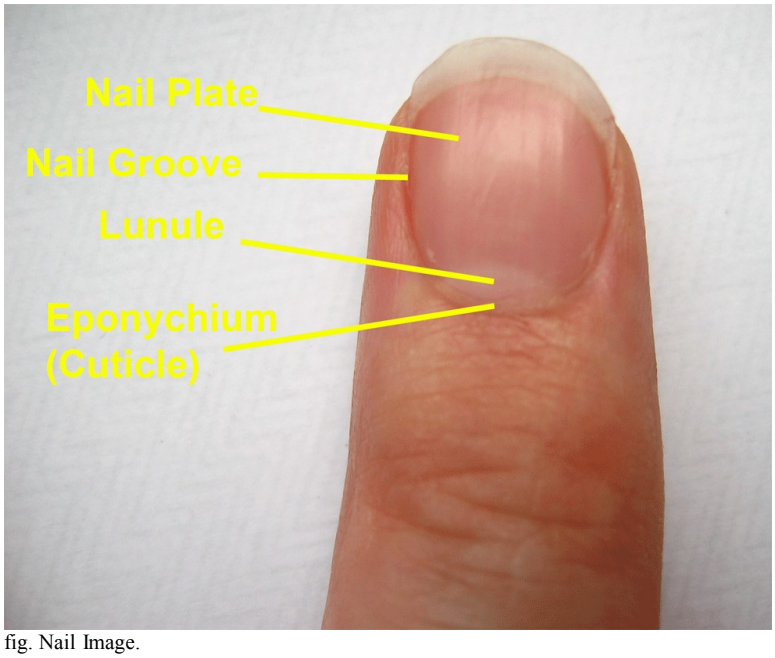


fig. Nail Image.

As (WIK06c) states, fingernails and toenails are interestingly a form of modified hair, comprising of protein. The structure of a nail consists of;

The nail matrix, still under the skin, it is the actual growing part of the nail
Eponychium, the fold of skin at the proximal end of the nail,known as the cuticle
Paronychium, the folds of skin at the side of the nail
Hyponychium, is the attchment between the skin and the the distal end of the nail
Nail plate, what we think of as being the nail, hard and transulaent consisting of keratin
Nail bed, connective tissue underlying the nail
Lunula, the crescent shape of the nail bed, whitish and a bit transparent

Chapter 7 - What is procedural texturing

One defined meaning of procedural texturing is that it is synthetic, in that it is generated from program coding and not just simply by accessing a texture image data structure. However, procedural texturing can access an image in creating the procedural texture and to good effect. The author argues, what defines it as procedural and not just texturing is the difference between the image queried and the resulting texture. It may be important to note exactly what texturing is for complete clarity. Texturing is the process of allowing a surface point's pixel colour to be calculated from a corresponding texture image. Every point on a three-dimensional surface has a corresponding two-dimensional parameter called u and v mapping. This 3-d to 2-d representation allows the u,v's of a surface to correspond to a pixel location in a given texture image. Therefore the colour of the image at any pixel position can be taken to calculate the colour at a shading point on a 3-d surface via its u and v parameter mappings. Procedural texturing has many advantages, including having no fixed resolution allowing full detail no matter how close it is looked at. Also it has no fixed size and so can cover any large area without repetition, distortion or other such effects. The surface can also change via many specified parameters instead of being stuck to one fixed image. Disadvantage however include them being hard to program and debug, prone to aliasing artefacts and can be hard to control yielding surprising results. [EBE98]

The RenderMan shading language is a programming language that allows you to write program code for procedural texturing as well as shading models. These procedures are called “shaders” and any RenderMan compliant renderer will be able to understand it in calculating sample point being coloured. It is important to briefly note what a shading model is. Shading models, also known as illumination models, calculate the colour of a pixel at a given shading point or sample point. It involves the interaction of light and the given surface colour at a specified shading point, both are taken and the resulting calculation yields the pixel colour. One of the most popular shading models is the diffuse or Lambertian model, which gives a dull appearance.

The shading language allows for calculating shading calculations for six different types of shader’s, which are distinguished by the inputs that they use and the kinds of output they produce. According to the RenderMan Companion (UPS89) these include:

- Light source shader’s
- Surface shader’s
- Volume shader’s
- Displacement shader’s
- Transformation shader’s
- Imager shader’s

The language has built-in data-types and functions that are commonly used in computer graphic calculations such point and color data types, math operators, and operators for vector calculation as well as many more.

Surface shader’s can be broken down into two parts, a pattern generating part and the shading model part. The shading model is the calculation of surface material in correspondence with diffuse and specular lighting hitting that surface at a given shading point. The pattern generation specifies a texture pattern over the surface along with surface properties to be used by the shading model in its calculations.

7.1 - Layer and Composition

Writing shaders can be a daunting task, as there are many ways to solve the same problems as with any programming situation. The flexibility of the shading language provides very powerful capabilities, but also means that it is hard to control. One way to control it is to use a method to control the madness. One such methodology is the layering and composition approach. As Stephen F. May (MAY) states, layering means that shaders surface patterns and texturing are broken down into smaller parts, with a “divide and conquer” technique towards the problem. The shaders complicated surface generated patterns are created in layers and composited on top of each other. An example for an “orange” surface shader is to break it down into smaller a parts ground upwards. First layer would be a mix of orange-yellow colour, next would be the motley-spotted surface, then bruises and scratches, then insects bites and finally illuminate. If any layer proves difficult to generate or control then simply break it down again. An example of the pseudo-code for a layered “orange” surface shader follows. It shows that the declared layer_color corresponding to the colour of the current layer is composited on top of the declared surface_color which comprises of the colour of all layers. This approach is one which I will follow in designing and creating the procedural shaders for the skin shader [EBE98],[GRI99] , [STE03], [UPS89].

surface orange(...)

```
{
 color surface_color, layer_color;

 /* background (layer 0) */

 surface_color = orange-yellow variations;

 /* layer 1 */

 layer = motley-spots;
 surface_color = composite layer on surface_color;

 /* layer 2 */

 layer = bruises;
 surface_color = composite layer on surface_color;

 /* layer 3 */

 layer = bites;
 surface_color = composite layer on surface_color;

 /* illumination */
 surface_color = illumination based on surface_color and illum params;

 /* output */
Ci = surface_color;}
```

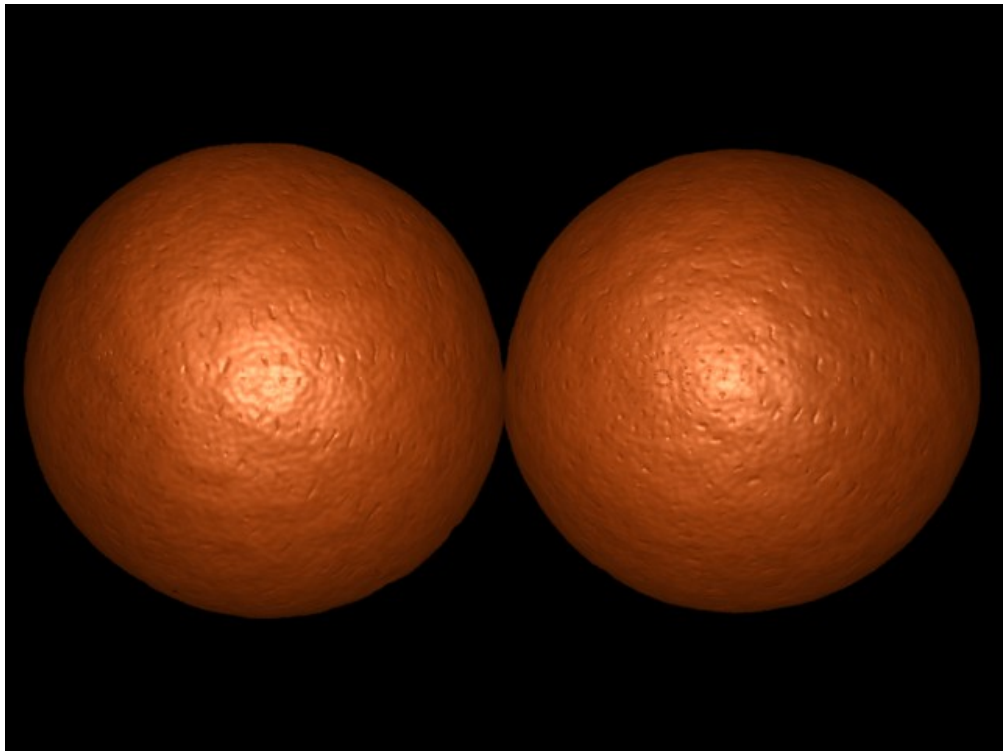


fig. oranges from authors work

Chapter 8 – Lighting

The design of my scene is perceived to be indoors. The polygon model of a hand that is to be used for the procedural shader to be applied to is set to be on top of a hard surface. The surface is to be a table surface or some sort of marble counter surface. In order to "model" this subject properly, fundamental concepts of lighting need to be applied. Firstly the pure modelling of light needs to be taken into consideration. For this a three-point lighting model is to be applied. However, before this can be accomplished, the underlying issues and goals of the three-point lighting model needs to be understood.

As Jeremy Birn states, one of the main goals of three-point lighting is to model with light. That is, to illuminate your subject so that the two-dimensional output fully represents your subject's three-dimensional form. [BIR99]

One example of modelling with light incorrectly is when a subject's lighting model is applied too uniform in all directions resulting in a flat looking lighting model, hiding the curvature of shading on the surface. To prevent this would be to model light brighter on one side than on another side, this models the subject illuminating the curvature of the subject and it's modelling.

In order to model the surface with light realistically, one must make sure that the different planes of the subject have different light values. In this it is meant that the subject should not be uniformly lit with ambient light. This results in adjacent planes not having any difference in shading, and doesn't move into a different tone of illumination as you move from one plane to another. This result's in the subjects modelling not be fully shown, as it is not modelled with light. Basically in order to avoid this, flat uniform ambience light should be avoided, with every light added to the scene having a clear purpose, resulting in the subject being shaded with different variants. [BIR99]

8.1 - Three Points to be made

As Jeremy Birn states, "the three "points" in three-point lighting are actually three "roles" that light can play in a scene, each serving a specific purpose" [BIR99]. The key light is usually deemed to be the most dominant light in illuminating the subject. This light shows the angle of lighting in the scene as well as casting the darkest most visible shadows in the scene. The Fill light softens and extends the illumination provided by the key light. The Fill light usually illuminates the subjects darker areas so that more of the subject is seen. Often representing a secondary light source in the scene, the fill light can illuminate by showing the effects of light that has bounced off or reflected from close surfaces onto the subject. The Backlight creates a defining edge to the subject, showing where the subject ends and background begins. Often providing a glint off the subject's head or hair (sometimes called "hair light"). [BIR99]

As Birn argues, three-point is a flexible principle of lighting, as is in no way set in stone. The lighting model should be adapted and tailored to meet the requirements of

the scene and the subject to be modelled with light. This means that the position and angles of these lights can also vary largely, but in order to be safe of going to far in one extreme an acceptable range should be applied.

The lighting angles can be relative to the camera, which provides a good practice. As Birn states, Three-point lighting works best when the scene is set up first with the camera set, before applying any lights. [BIR99] The range roughly states that the key light should be positioned above the subject and a little to one side; this is usually how we see view people in environments with overhead light. Positioning the Key light 15 to 45 degrees above the camera and 15 to 45 degrees to one side of the camera is a good range. The fill light should only be slightly above the camera angle, about 0 to 30 degrees. It can also be 15-60 to one side of the camera. However, to note, fill light is usually positioned to be on the opposite side and angle of the key, in order to simulate light from the key light bouncing off surfaces close to the subject and onto the subject or secondary sources of light. The backlight should be positioned behind the subject, however care should be taken that it is not positioned opposite to the camera. The backlight often has little or no effect if it is directly behind the subject, if it is slightly above the subject and slightly more towards the camera it will have more effect. The use of multiple backlights with diffuse illumination, not specular, can illuminate a subject to a great extent. Also backlights can have strong intensities, even higher than the key, as they are positioned behind the subject and take no shading away from the key lights illumination only highlighting the edges of a subject. [BIR99]

8.2 - Shadows

Shadows are extremely important in a scene. Shadows add realism to the scene, add tone richness in shading the subject, correlate different objects in the scene together and improves the composition.

Shadows often show the relationship of an object with its surrounding. Shadows can show if an object is planted on the ground or indicate how high it is above ground level. Shadows also show if subjects are close to each other, casting shadows on each other, or if one object is in front or behind another object. Shadows can also show the profile of the subject, be casting the shadow of the subject onto a wall. Shadows, can also add to the composition of the scene, adding shadows of objects not visible in the scene can break up the shot and add to a more pleasing rendering. Shadows cast from objects unseen objects can give an indication to what that object is, always increasing composition and realism. Shadows cast from objects can also add a contrast between the object and its background. It suggested that light from outdoors, the sun, often casts sharp strong shadows, whereas indoor lighting produces softer cast shadows. This should be taken into account within the scene, the author argues.

Shadow areas, are those areas within in a scene that have not be illuminated, or where light has been blocked by something. Shadow areas are basically still in shadow, even if their were no shadows rendered within the scene. However a second type of shadow, are cast shadows that are dark areas cast directly from a light being obstructed by an object. It is important to note that multiple shadows can be confusing and cluttering. Also shadows, be them shadow areas or cast ones, are usually not in total darkness, as light bounces and reflects, illuminating them slightly. Therefore shadows should not be in total darkness, which is unnatural looking, they should be a

slightly lighter color. In order to solve this, a fill light can be used to increase the shadow brightness of the shadow area, which is unlit by the object. This fill light will also brighten the shadow cast area with a natural equally toned amount. The use of a fill light is better than using increasing shadow brightness on the light or using ambient light, both of which produce unnatural results. [BIR99]

All of the points and guidelines made above, be need to be taken into account when setting the scene.

Chapter 9 - What is SSS

Sub-surface scattering is used for simulating translucent materials such as skin, fat, fruits, milk, marble and many other such materials. Translucent objects exist where light can travel through the objects surface and then diffuse underneath that surface, instead of merely bouncing off that surface. A large range of translucent materials exist around us, most of which are non-metals. Materials such as milk, skin, wax, marble are all translucent in that if you shine a laser light, perhaps, directly at and into these materials, you will notice that the light penetrates the surface, changes slightly in angular direction and diffuse mostly uniformly in all directions depending on the material. If the material has geometry under the surface, then some rays of light can bounce off this geometry and exit the material at a different point at which it entered. [BLE04]

9.1 - Light Transport

When light hits an objects surface, it react in some way, including reflection, refraction, absorption, scattering and diffusion. An example of a diffuse surface, e.g. Lambert model, would be a where a ray from a light hits the surface, makes a shade call, resulting in a color value, bounces to the camera and shades the individual pixel. In real life this is actually a bit different, where a ray of light hits the surface, the surface then absorbs some of that lights ray depending on the material, and the rest produces a colour that is reflected back towards our eyes. The materials surface has a large part to play in the process, depending on how smooth or rough that surface is. [BLE04]

However, many approximation have been made in order to make things faster within the computer industry. How light reacts when it hits a surface is down to the Bidirectional Reflectance Distribution Function (BRDF). Most of the BRDF models describe how a ray of light which enters a surface, will leave that surface at the same position, with any subsurface transport being approximated by a Lambertian component. The bi-directional reflectance distribution function (BRDF), was introduced by Nicodemus [NIC77]. Some BRDFs models can be simple approximations, such as the Lambert model whereas as other more complex do exist. However, it depends on the needs of the artist, whether an approximation may be suitable enough or a more complex BRDF model or and entirely different even more complex on is needed such as Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF). [BLE04]

9.2 - Translucency

When a light rays hits a translucent material such as skin, some of these rays simple bounce off the surface at the exact same position like the Lambert model. However, some of the rays get absorbed within the top Epidermis layer of the skin and die, whereas others will travel further within the skin. Of these rays, some will change to a slightly different direction as what they entered at when they pass into the Dermis layer, containing blood vessels, nerves, and smooth muscle tissue. Some of these rays will hit blood vessels and muscle, get absorbed and die, whereas others will perhaps bounce and reflect back to the surface. Of the rays that bounce and reflect, they may exit and a different point at which they entered the surface. Other rays will travel further into the final layer of the skin, the Hypodermis layer. These rays will eventually meet the bone where they will die. [BLE04]

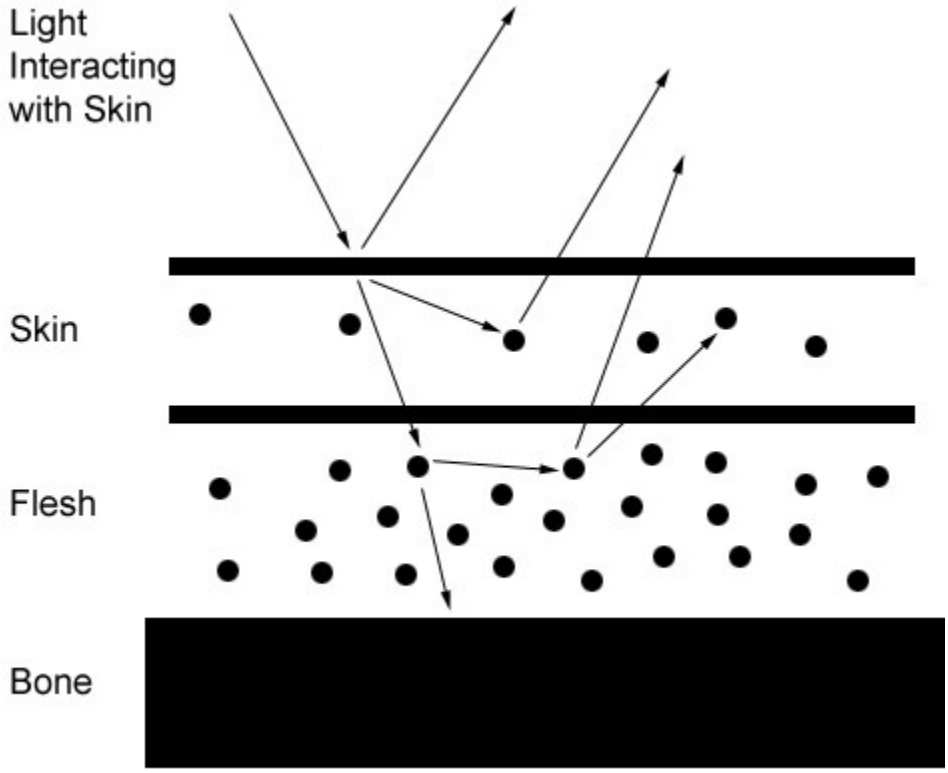


fig [BLE04]

In order to simulate this a more complex model than BRDFs models is needed. A lot of research has gone into more complex models. One of the primary ones was Pharr and Hanrahans model which simulates subsurface light transport in the paper “Monte Carlo evaluation of non-linear scattering equations for subsurface reflection” [PHA00]. However, this model is computationally very expensive. The most recent and more accurate, less expensive, model is Henrik Wann Jensens model, from the paper "A Practical Model For Sub-Surface Light Transport" [JEN01]. Jensen started his research by sending a beam of light, using a laser pen, through a glass of milk and noticing how it scattered mostly uniform through the milk material. Jensens model for simulating sub-surface scattering works well with highly scattering (multi-scattering) such as milk and skin as it scatters lights [BLE04] [JEN01].

Previous models of single scattering terms existed but did’t work well with materials such as milk and skin as these materials require multi-scattering to represent how light passes through isotropic (same in all directions). Models exist to represent this multi-scattering in way of diffusion theory. Jensen’s model works by combining both a single-scattering term with a diffusion term to create the Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF), which still has a fast speed in computation [BLE04] [JEN01].

Jensen’s model uses a generalization of the Hanrahan-Krueger Model for the direct component (due to single scattering), and a diffusion approximation (due to multiple scattering). For example, human skin's diffusion term is very high, whereas it's single scattering term is very small [BLE04] [JEN01].

9.3 - Single Vs Multi-scattering

When a laser is shot through a leaf there is a strong focused beam of light that enters and then exits in a direct manner. This shows the single scattering of light as it passes through the material. There is also a diffused glow of light around this strong focused beam, which is where light is being diffusely multi-scattered.

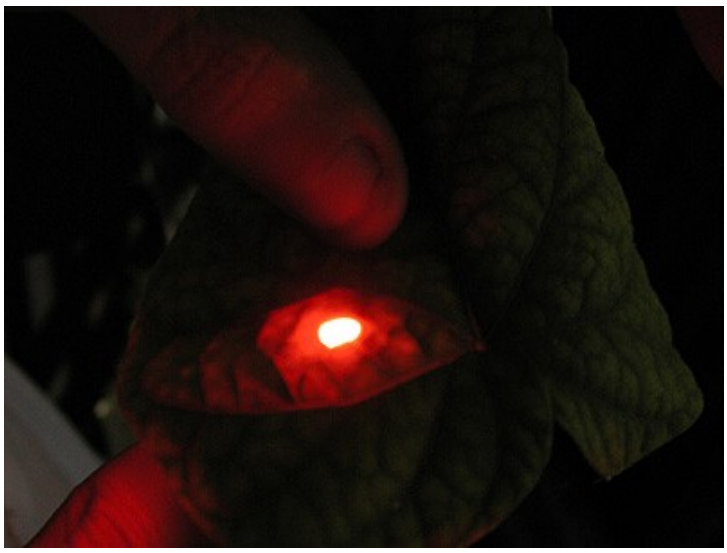


fig [BLE04]

Another example is shown where light has softly multi-scattering through grapes in an isotropic way, as they are backlit. As seen the light actually illuminates the veins, as it passes through even travelling into other grapes.

fig [BLE04]

When light enters skin it can leave at a slightly different point at which it entered, back-scattering. When light enters skin it can exit into a shadowed area, illuminating that area softly, with colored light effected by the tissue within the skin. This is where we get color bleeding. Light that travels through a material and exits the others side is known as forward-scattering, seen clearly when light passes through illuminated ear tissue. Light within skin tends to multi-scatter isotropic, as light bounces around the layers of the skin constantly illuminating until its uniform as shown below, even though the skin material is anisotropic [BLE04] [JEN01].

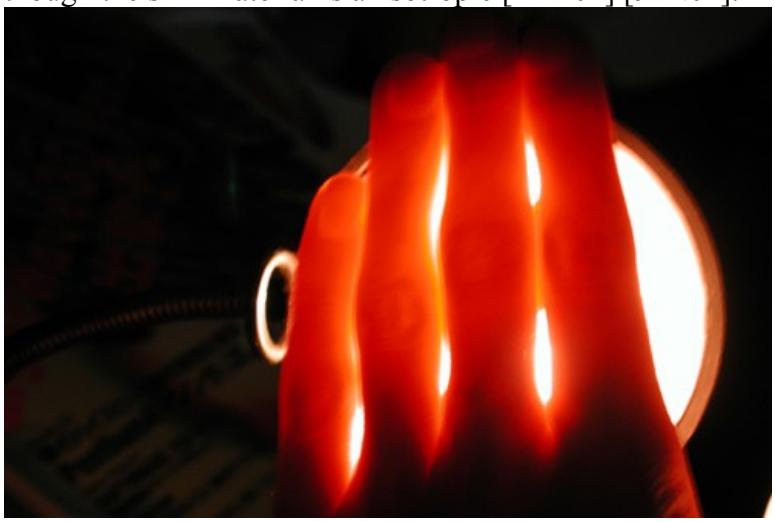


fig [BLE04]

Chapter 10 – Design

10.1- Procedural Texturing - Breaking it down

The hand was to be separated into different regions, with shader's being designed for these individual regions. Before even thinking of the coding, the author gathered as much reference material as possible in preparation of creating procedural skin shader's to represent the human hand. This is one of the most fundamental parts of shader writing, without which can lead the author down a long dark path in the wrong direction [EBE98],[GRI99] , [STE03], [UPS89],

The first step was to separate the hand into different bone and skin regions; so as to design individual shader's based upon these different bone regions as well as skin regions. This was accomplished partly by gathering information on the bone structure of the hand [WIK06a]. This also served as being good reference on the geometry under the surface, which can be taken into account on subsurface scattering calculation design. However, it was decided that because of the isotropic behaviour of light transport through skin in normal lighting conditions, this did not have to be taken into account for subsurface scattering design, useful though as in different lighting conditions it would be needed [BLE04], [JEN01], [JEN02]. The bone structure also served as an indication on whether the underlying geometry has any effect on the appearance of the surface of the material. However, it was deemed that it did not have much effect on the appearance of the surface shading.

The next step of reference material gathering was by taking photos of the different regions of the hand. Gathering photo material of the object for which you are designing a shader for, is one of the best reference material sources, it is suggested. Another reference source was on the composition of human skin, gathered from Wikipedia (WIK06a). This provided a great understanding on how the behaviour of cellular and blood vessel activity plays upon the surface appearance of skin [WIK06b]. After this, reference material was gathered on the composition of the human nail, providing insight and ideas on how to recreate this with a procedural shader [WIK06c]. Using all reference material, individual shader's for different parts of the hand were separated and categorised mainly according to bone structure regions, with shader's being designed for the front and back of the hand.

10.2 - Layers that will be needed

It is important to note that the following pseudo-type code can be subject to change during the creation and implementation stage. However, the author will probably stick fairly close to the outline of each of the shader's and their layers as follows.

backPhalange2.sl: This region is perceived to firstly have a noised colour value layer, comprised from tissue colours from the hypodermis layer. This layer is mainly based upon the perceived representation of tissue and smooth muscle colour. The next two layers are based upon the dermis layers of the skin. The colour values are perceived due to cellular activity from keratinisation through the layers, as well as the presence of blood vessels. For these two layers, noised colour values of red, whitish and cyan-

blue colours to be used. Next will be a layer to represent very slightly seen cyan-blue blood vessels within this dermis region. This may be created using turbulent noise function applied to repeating lines, colour is cyan blue its believed. The next two layers will represent the epidermis layer of the skin. For this it is perceived that a very slightly visible vertical and horizontal criss-cross effect is present. This will be accomplished by using pulsing smoothstep lines based upon distances to specified points in correlation to the shading point coordinates. The skin seems to have waved type lines in the horizontal in this region also, so this will be the next layer. The next layer will be a noised colour spline to represent the surface colour as well. Finally noising the normals on the surface will be needed as the skin surface is anisotropic with light diffusing in different directions. Finally, apply the illumnation model with high to low diffuse to specular components.

fig backPhalange1 in question. Appendix refImages

backPhalange2.sl
Layer 1 – Noise-based colour spline
Layer 2 – Spotty-Mottle effect – Colour values red noised
Layer 3 – Spotty-Mottle effect – Colour values white noised
Layer 4 – blue lines noised using turbulence
Layer 5 – Horizontal pulsing lines, noised start, end and rotation positions of lines (bump map)
Layer 6 – Vertical pulsing lines, noised start, end and rotation positions of lines (bump map)
Layer 7 – Strong Waved horizontal curve type lines (bump map)
Layer 8 – Noise-based colour spline
Layer 9 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

backPhalange1.sl: Again this region will have a noised colour value layer, to represent the colour values from the hypodermis layer. The next two layers again will have mottled red and white and cyan-blue and white based upon the dermis layers of the skin and presence of blood vessels, with red, cyan-blue and light skin colours. Next will be a layer to represent the dermis region containing slightly cyan-blue blood vessels within this region. This may be created using turbulent noise function applied to repeating lines. We are now at the epidermis layer. This layer will have a voronoi pattern, but only for a third of the surface, the first third. The layer after this shall have vertical lines with a slight turbulent value perhaps and only consumes the center of the

texture, about a third in length as perceived from the eye. Next will be straightish faint horizontal lines, representing one of the final layers. Finally noising the normals will need to be accomplished in order to represent the skins surface correctly. After this an illumination model is provided, again with low specular and highly diffuse.

fig phlanges 1. refAppedix

backPhalange1.sl
Layer 1 – Noise-based colour spline
Layer 2 – Spotty-Mottle effect – Colour values red noised
Layer 3 – Spotty-Mottle effect – Colour values white noised
Layer 4 – blue lines noised using turbulence
Layer 5 – Voronoi pattern applied(only within a third of the surface, bottom bit)
Layer 6 – Vertical pulsing lines, within 0.3-0.7 of S, with slight turb applied or simple noise. The start, end and rotation of these lines will also be noised
Layer 7 – Horizontal pulsing lines, noised start, end and rotation positions of lines (bump map)
Layer 8 – Noise-based colour spline
Layer 9 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

backMetaCarpus.sl: This is a complex shader starting off, like all other shader's with a noised value looked up into a colour spline. Next layer within the hypodermis region of the skin we shall have two layers of colour mottling. After this layer, within the dermis region of the skin we shall have turbulent vertical lines, to represent cyan-blue vessels that are present. These vessels are stronger here than the phalanges regions. Within this epidermis region of skin, an additional layer containing a voronoi pattern is applied uniformly which should be noised. Vertical lines are provided for the next layer with turbulent noise added. The next layer is for half the area of this surface the bottom half which will contain a criss cross line pattern. Finally add a layer for noised colour spline and an additional layer for freckle spots on the surface. Also very final layer noises surface normals. Illuminate.

backMetaCarpus.sl.
Layer 1 – Noise-based colour spline
Layer 2 – Spotty-Mottle effect – Colour values red and white noised (using either fBm, turbulence or just simple noise functions)
Layer 3 – Spotty-Mottle effect – Colour values blue and white noised (using either fBm, turbulence or just simple noise functions)
Layer 4 – Strong blue lines noised using turbulence
Layer 5 – Voronoi pattern applied uniformly (bump map)
Layer 6 – Vertical pulsing lines, , with slight turb applied or simple noise. The start, end and rotation of these lines will also be noised (bump map)
Layer 7 – Criss-cross pulsing line effect, only in lower half of surface. Noised start, end and rotation positions of lines (bump map)

Layer 8 – Noise-based colour spline
Layer 9 – Freckle spots, noise a little
Layer 10 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

phalanges3.sl: Start with a noised colour value layer. The next two layers are based upon the dermis layers of the skin. The colour values in this hypodermis and dermis layers of skin are real turbulent, perhaps due to the softer skin tissue than on the back of the hand. For the next two layers, red and blue will be mixed with light using turbulent noise or fBm. The blood vessels from the dermis layer also seem to be more visible so this will have to be represented in the next layer with a turbulent value added to vertical lines. The final layers will represent the epidermis layer of the skin. The first layer at the epidermis layer of the skin, contains the spiral fingerprint effect which will have to be created, probably by using concentric noised lines from a specified central point to point being shaded. The next layer will be a noised colour spline to represent the surface colour as well. Next layer involves noising the normals on the surface will be done. Then, apply the illumination model with high to low diffuse to specular components.

phalange3.sl
Layer 1 – Noise-based colour spline
Layer 2 – Turbulent Spotty-Mottle effect – Colour values red and white noised
Layer 3 – Turbulent Spotty-Mottle effect – Colour values blue and white noised
Layer 4 – blue lines noised using turbulence, stronger blue
Layer 5 – Spiral pulsing effect for fingerprint (bump map)
Layer 6 – Noise-based colour spline
Layer 7 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

phalange2.sl: The lowest layers of this shader has very mottled turbulent coours of red, blue and whites. Moving upwards layers containing the blood vessels need to be provided. The next layers, at the top layer of the skin, have a collection of wave patterns, vertical pulsing lines and horizontal lines. After these layers is another pattern, a horizontal pulsing line curves pattern. Finally we come to colouring the surface, noising the normals and illuminating.
phalange2.sl
Layer 1 – Noise-based colour spline
Layer 2 – Turbulent Spotty-Mottle effect – Colour values red and white noised
Layer 3 – Turbulent Spotty-Mottle effect – Colour values blue and white noised
Layer 4 – blue lines noised using turbulence, stronger blue
Layer 5 – Wave effect noised (bump map)
Layer 6 – Vertical pulsing lines (bump)
Layer 7 – Strong horizontal pulsing lines, more spaced and stronger (bump)
Layer 8 – Few horizontal pulse curves
Layer 9 – Noise-based colour spline
Layer 10 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

metaCarpus.sl: A very large shader, this one starts like the others, with a noised spline colour look up. Next are two layers of very turbulent colour mottling, representing the dermis skin layer. After this turbulence is used in two different layers to create blood vessels one lighter than the other. Next is an important layer where three points will need to be specified and then pulsing lines will have to be generated from these points outwards to represent the fingerprint like lines of the palm of the hand. In the following four layesr their are varying degrees and locations of turbulent pulsing lines. Then three layers will provide wave like horizontal lines of the hand, which will be displaced. Finally the surfaced will be colour splined and then the normals are noised, before illumination is carried out.

metaCarpus.sl
Layer 1 – Noise-based colour spline
Layer 2 – Very Turbulent Spotty-Mottle effect – Colour values red and white noised
Layer 3 – Very Turbulent Spotty-Mottle effect – Colour values blue and white noised
Layer 4 – Using turbulence, create stronger cyan-blue lines for blood vessels
Layer 5 – Using turbulence, create stronger blue lines for blood vessels
Layer 6 – Specify three points that emanate a wave pulse effect noised (bump map)
Layer 7 – Short Spaced Vertical turbulent pulsing lines in top left region of surface (bump)
Layer 8 – Strong close horizontal pulsing lines, more spaced and stronger (bump) in bottom left region of surface
Layer 9 – Faint close vertical pulsing lines
Layer 10 – Wave lines noised in the horizontal direction
Layer 11 – One very strong Horizontal wave travelling upwards (Disp)
Layer 12 - One very strong Horizontal wave (Disp)
Layer 13 - One very strong Horizonatal wave travelling downwards (Disp)
Layer 14 – Noise-based colour spline
Layer 15 – Noise the normals, to recreate the anisotropic effect of the skin surface normals
Illumination – High Diffuse, low rough specular

nail.sl: This will be quite different in generated textures as well as surface colours and shading. First , a ramp between the lunula and the end of the finger nail will be created. This will be ramped pinky to white as observed, believed to be called the nail matrix. Next will come the generating of a texture for the crescent like shape of the nail bed, called the lunula. This is quite transparent and provides a distinguishing colour. A line will be placed between two points to represent the ends of the nail, this will be whitish and transparent. Next will be the rough jagged lines present on the actual nail plate, the surface here is specular and transparent, issues that will need to be taken into consideration. Finally a noisey cuticles layer will be added ontop, again transparent a little.

nail.sl
Layer 1 – Create a colour ramp using noised colour spline,
Layer 2 – Create crescent shape – use in spline to colour
Layer 3 – Places line to represent end of nail – whitish and noised
Layer 4 – Create rough lines on nail plate surface
Layer 5 – Create cuticles with noised line – transparent
Illuminate – Specular like shiny.

9.3 - Maya model

The procedural skin shader, will be implemented using the RenderMan shading language and then test rendered using PhotoRealistic RenderMan(PRMan). After the author is happy with the results and time permits, the shader will be applied to a polygon mesh for mere presentation purposes. This polygon mesh will represent the human hand and modelled using Alias' Maya modelling package. The model will be created in parts using the same "divide and conquer" technique to the problem that will be used in creating the shader's. Firstly the individual fingers and thumb will be modelled against reference material photos of the fingers and thumbs set up on planes in the x, y and z of co-ordinate space. This is a frequently used technique in computer generated (cg) modelling. Next the metacarpus region of the hand will be modelled, again using reference material. Finally these will be stitched together, along with modelled nails. All parts of this process will introduce small problems due to lack of knowledge and experience in modelling.

9.4 - MTOR and its use

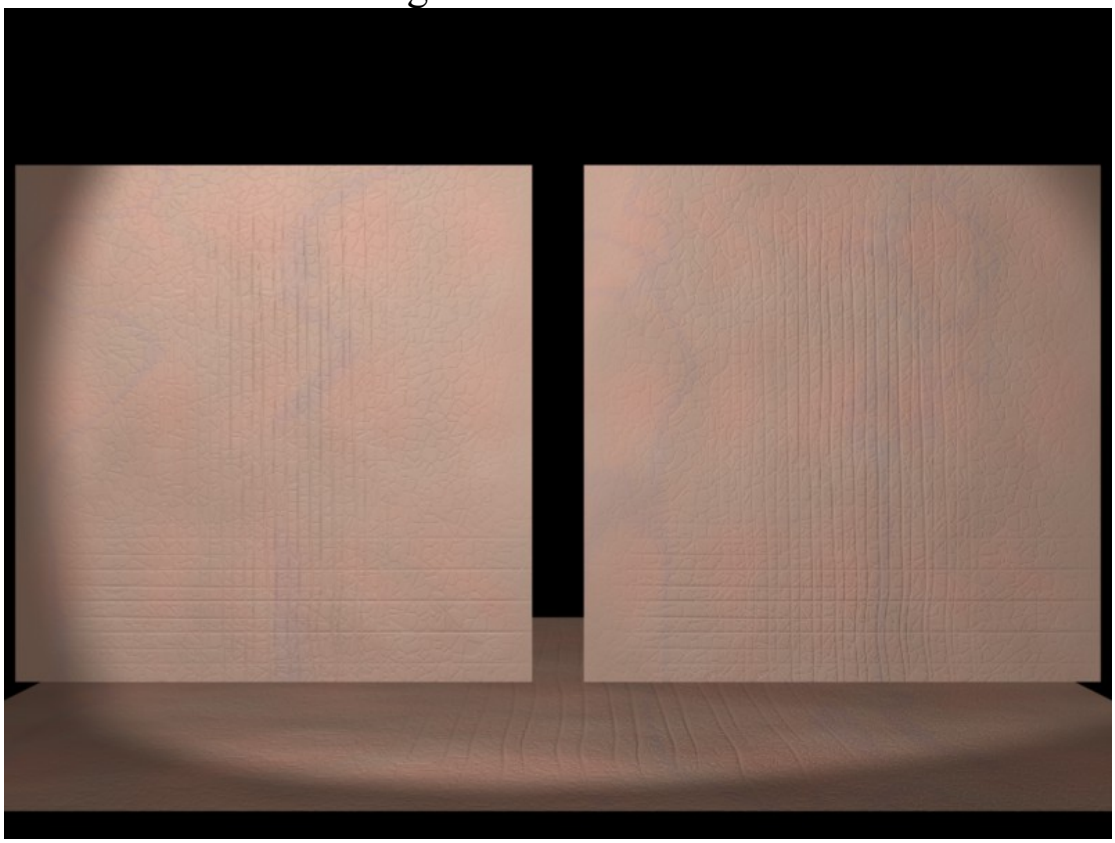
For this project, MTOR (Maya to RenderMan) was investigated as a possible interface between the shaders created with Pixars's RenderMan and Alias' Maya. MTOR provides a pipeline between the two offering file convertion and data about the subdivision surfaces used in Maya aswell as lighting objects.

MTOR also provides the Slim component. This was used to learn how to attach RenderMan attributes to objects in a Maya scene. This was one of the main areas of this project as the author needed to investigate the different ways in which RenderMan shaders can be ported and used within an environment such as Maya.

Slim allows for the import and management of RenderMan shaders in way of an interface. RenderMan shaders are organized in Slim via the use of palettes. From these palettes, shaders can be attached straight to maya objects in the scene with relative ease. However, the author found that although it was relatively easy to attach shaders to whole objects, the degree of difficulty increased as soon as shaders needed to be attached to individual faces on the objects. From here, the user needs to deal with UV texturing editing of the objects. Texture editing is an art in itself, and as such a new learner, such as the author, can find various aspects of it very difficult and time-consuming to use. As such the author ran into these problems. These will be discussed later. Back to Slim. Slim can also be used to create shaders visually with the use of a graph editor. Although this may seem great, the author suggests otherwise. Instead it is believed that one tool is being exchanged for another and thought needs to be put into both Also, the graphical user interfacing of creating shaders limits the planning, design and thought that should go into creating shaders, it is argued. Instead something is just created for the sake of it and soon as. Coding makes the author think about not only the composition of layers within a shader, but also the different functions that can be used in order to create those individual layers. It was for these reasons that the author created shaders using the the RenderMan programming language and not a GUI one such as Slim.

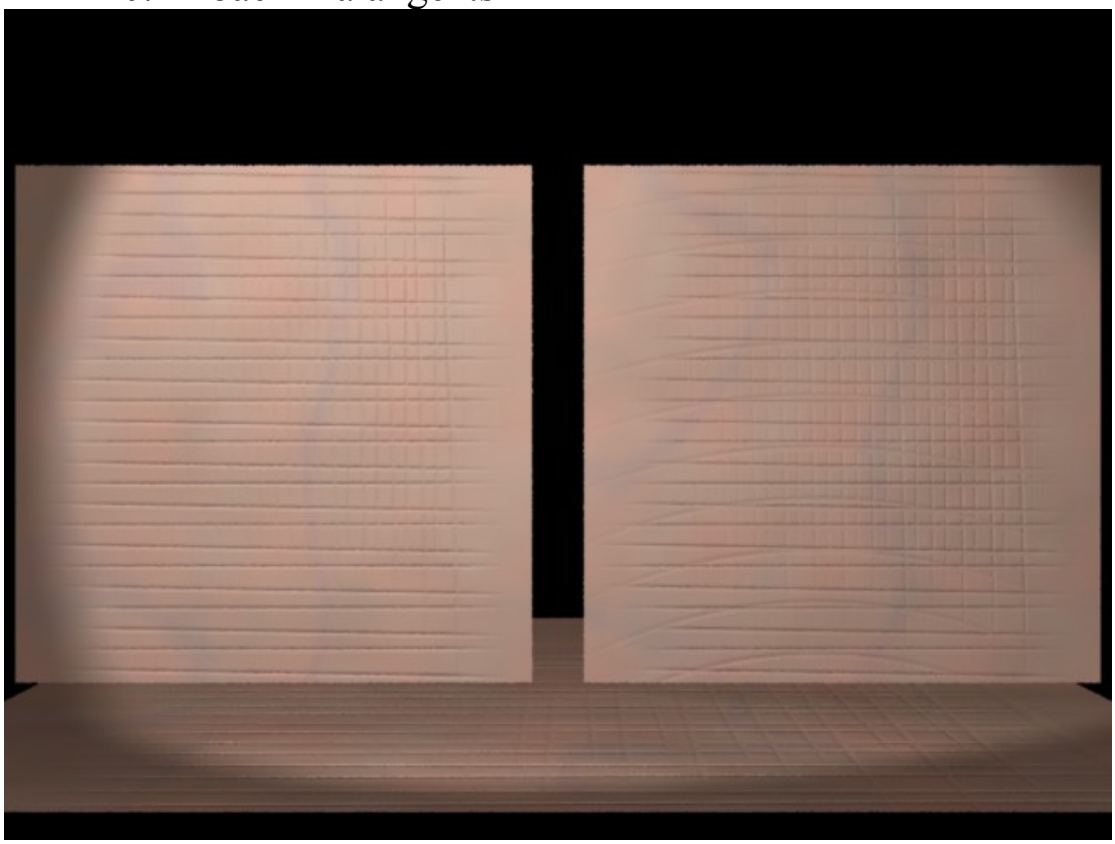
Chapter 10 – Implementation : Shader Creation and Explanation

10.1- backPhalange1.sl



This shader was created for the back of the hand for the phalange bone on the first row, hence the name. The author took various reference images of this area to get as mush clarity as possible. After this the shader was designed with detailed textures being generated for this detailed area. The shader provides a realistic look for this area of skin with pulsing lines in the horizontal and vertical which are appropriately clamped in the right postions. Noise is added to every layer as this gives a random organic look to every calculation made. This starts at the layer which is a noice spline to specify a base colour. Next a mottled appearance is given to show blood and tissue colour interaction. Texture patterns were generated and clamped in the different areas of this complex but small surface. One such pattern was the Voronoi pattern which provides an organic look just in its composition, it is suggested. This is probably why it is used to create basic but realsitic water effect patterns of the ocean[EBE98]. Other were lines intrinsically placed a specified distances and points all with nosied values.

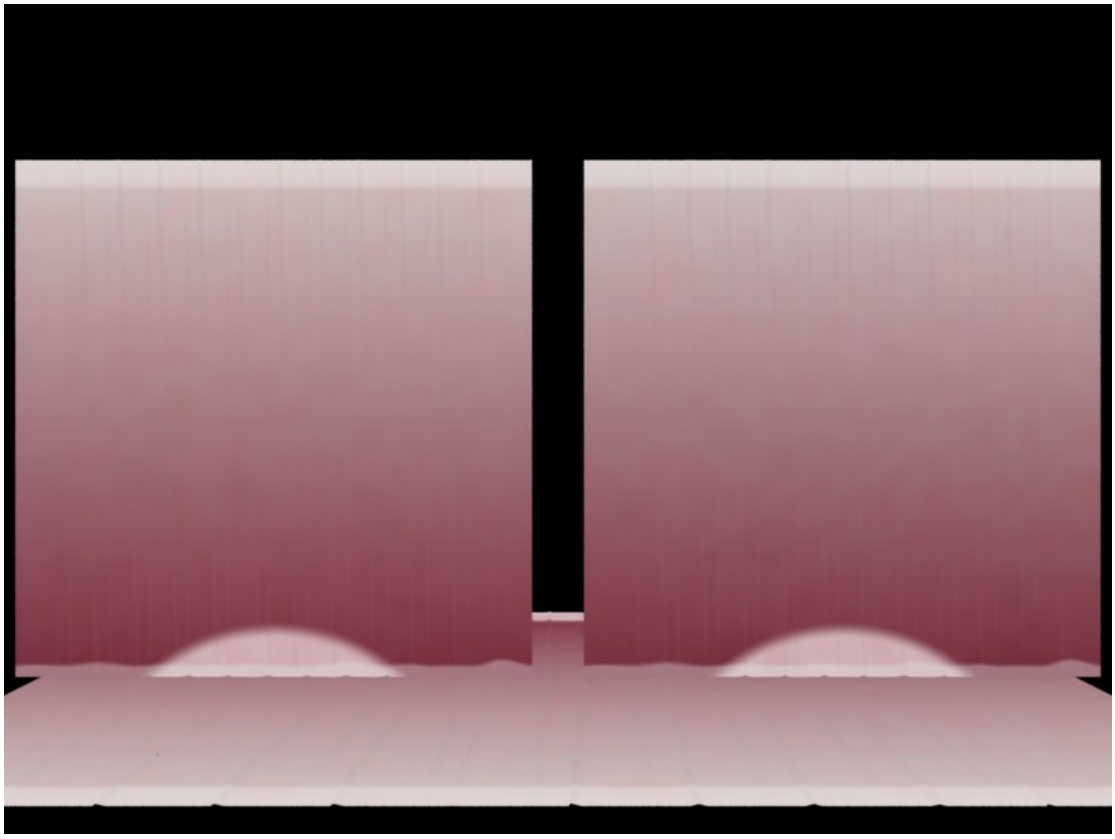
10.2 - backPhalange2.sl



This is to represent the skin surrounding the phalange bone in row two the bone structure of the hand. Its starts out like all others, a splined colour function, with a noised valuep of P as its input. The author likes starting with a base like this as it gives you something to build upon, very good for organic type surface which have different layers of skin with different tissue and colour within them layers. Next spots are created using simple calculations between points and the smoothstep functions are performed based on these distances. Inputs of s and t are noised and offset so that the coordinate copies taken are presently an organic mush of spotty colour. Another way

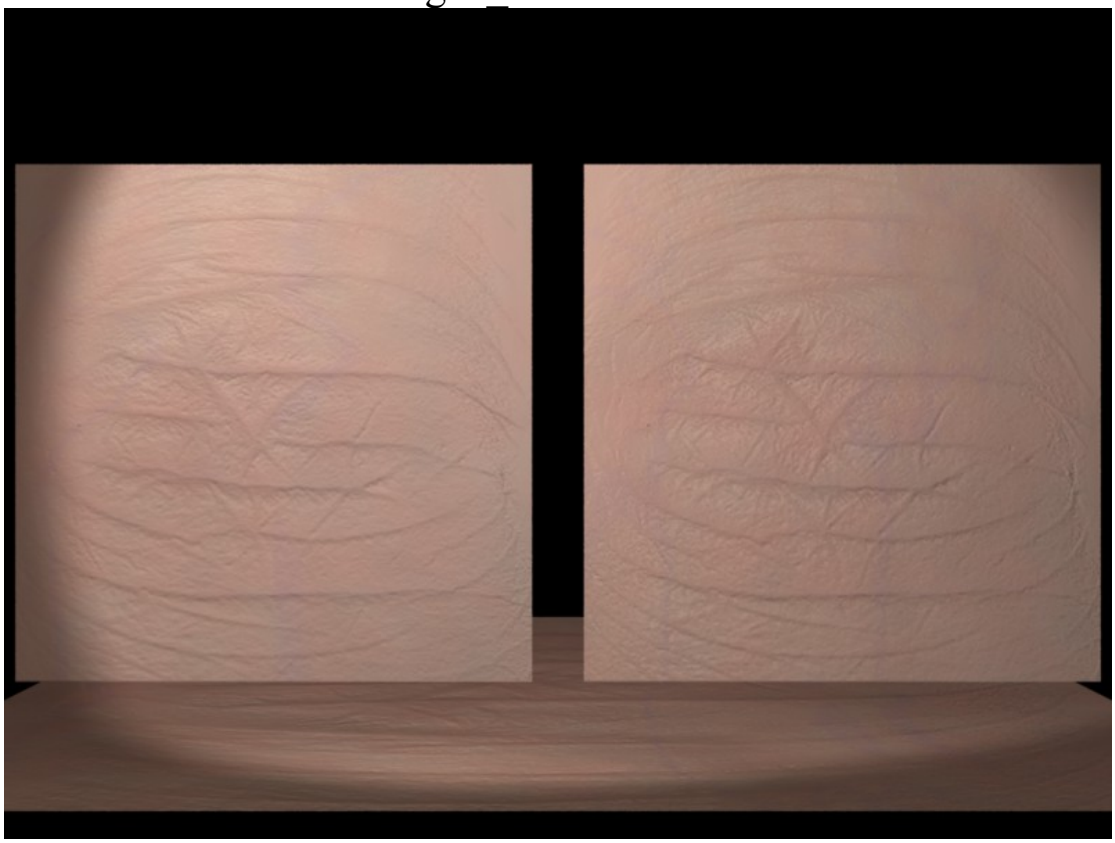
of doing this would be to use noise with frequency and amplitude controlling it, then use the value to simply mix between two values. However, this way done gives a little more control. In the next layer to follow, simple lines were created and the input values used, already passing through a turbulence function which summates noise into the value passed. This results in the lines receiving a degree of wiggliness, depending on the parameters to turbulence, octaves, amp, and gain. The following two layers create smooth lines which are noised in position and rotation to give an organic feel.

10.3 – nail.sl



The nail was created using two spline funtions to output two colours the were ramped in the t direction, A lot of trouble for mixing two colours but it gave a realistic and small range of colour needed. Next the crescent like shape was created, the proper name being the Lunula of the nail. This was created as you would a simple disk shape. By specfying a static point postion and measure to that using the s and t co-ordinates as we shade. The a colour is specify as to where we are within a certain distance of this distance range or not. If we are we colour a one, if not we colour it a different one. Lines which are created to mimic the cuticle, are noised and made very transparent as is the material on the human nail. A line is also specififed for the end of the nail. Repaeting smooth pulsed lines are created and noise to the rough line curvature on the top of the nail. These line value are displaced slightly upwards.

10.4 – backPhalange1_2.sl



This shader was designed and created as small aspects of the skin around the hand contains details that would be not be plausible creating procedurally. As Ebert (EBE98) states, texture lookup can provide fine high quality results for creating textures and their is no reason for not using them. As stated by Ebert, the value returned does not have to color, instead a float can be returned which can then be modified if needed and then assigned to a color for surface colouring. In this shader this occurs. The reason that the color is not just straight passed is so different parameters can gain access to the texture and control it in different ways, be its opacity value or whether or to blur the texture from it input. The shader starts like others, splining a base colour and then mottled colours are added. if these layers where omittted then this shaded should clearly look different or at least not tie closely with others side by side. Next , turbulent lines are added to represnt blood vessels. After this the texture is looked up, feeding into controlling parameters like transparency. By controlling and mixing the opacity of individual layersm different layers can bleed into other to specified user extents. This was a key design goal and has worked well it is suggested. The final layers add some splined colour and noise the normal. This shader can obviuosly have different textures fed in and equally work well in bringing acroos fine detail.

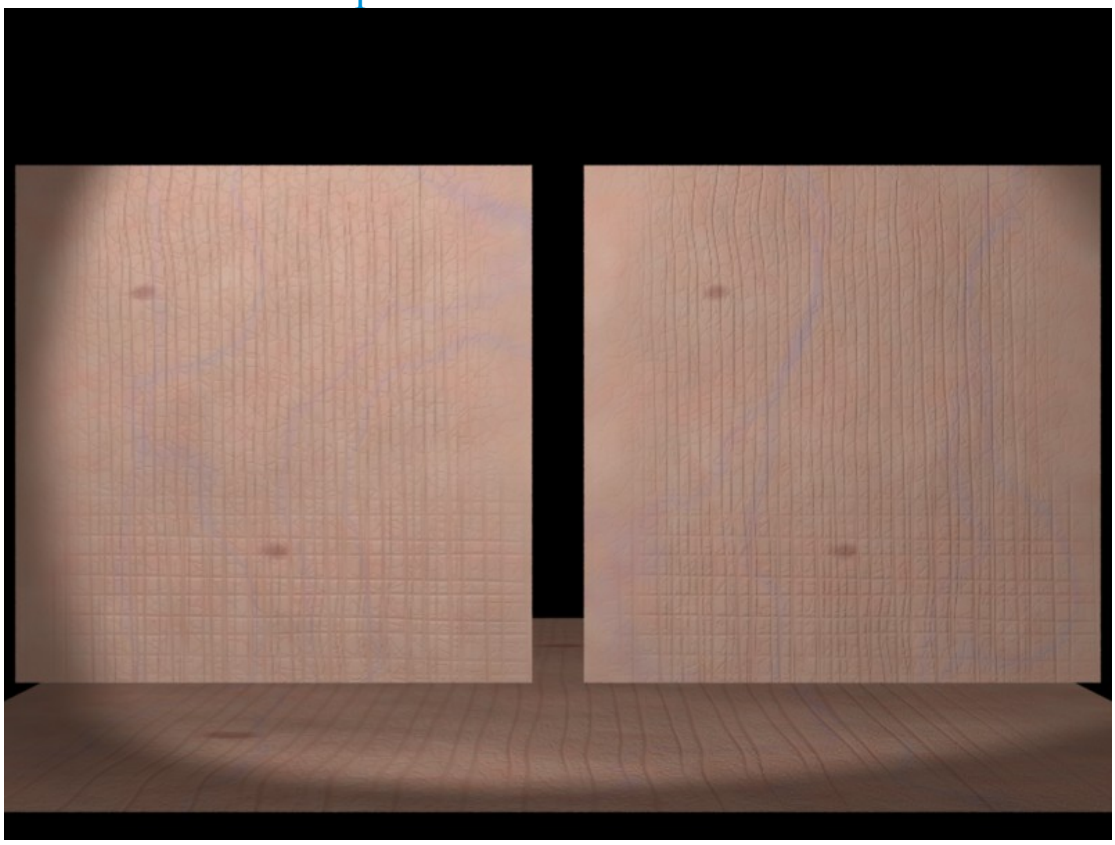
10.5 - Different ways to apply shaders badly using MTOR

There are various ways to apply shaders onto objects in scene in Maya through the use of MTOR. Of them their basically three different ways, two of them which Pixar believe are useful and one which they state you shouldn't use if you know whats good for you. This one, actually interpolates the geometry facets of where you have attached the shader. Basically, you create Maya sets which contain facets from the mesh and then RenderMan shaders are attached to these facet sets, the signal is sent to MTOR to output each facet collection as a different RenderMan primitive, each with its own appearance. Because subdivided surfaces rely on neighbor vertex locators and connectivity for the smoothing process, you have to be really careful when applying to these shder sets. Basically one single facet will be in more than one shading set. Because of this interpolation smoothing will occur causing quite a few problems with the apperance of parts of the geometry that you never even applied a shader to as the author found. Ways exist to fix these issues but their at a large cost to time needed, these issues resulted in the author over-running of the personal deadlines to get the shaders applied to geometry which was very personally disappointing.

Another way is to apply shaders to objects using MTOR projection planes. This is deemed as being usful by the authors Pixar. It works by setting up a projection plane directed towards your object. It works very well at projecting the texture or shader onto the object to start with. However, you ultimately find that texture is actually projected onto the back of the object as well, only reversed. This is deemed as being entertaining, which the author agrees with, but only useful for objects representing repeating textures on the back of them, like a coke can and only a coke can. Not very usful for a human hand.
The final way is by applying shaders to groups of faces. Most users simply need to apply a single Slim appearance(shader) to each subdiv surface and get the result required. However, the author wanted to apply shaders to different faces at different scales. Although it is possible to attach multiple shaders to objects, and results in a way that is expected. Different unwanted results occur when more and more shaders are applied and the geometry is more complex than a simple cube or cylindrical shape. Output images are given from my testing with this method. It show a small polygon cylinder with shaders attached to maya shader groups, a larger maya subdiv mesh and then a larger Mtor subdivided mesh.

Chapter 11 – In-depth Look at the backMetaCarpus.sl shader

shader backMetaCarpus.sl:



As this is one of the largest shaders that was created during this project I feel that this should be explained clearly, for the purpose that a reader and possible new-comer to RenderMan shader writing will find useful. This is a surface shader contained within the file "backMetaCarpus.sl", when compiled it creates a "backMetaCarpus.slo" file which can then be attached to geometry in a rib file that the author used for testing and implementation phases. This shader contains eleven layers, all of which hold individual layer parameters clearly organised. Firstly shader illumination parameters are setup, with specific light co-efficients such as Ka, and Kd used for shading model calculations of the surface(ambient and diffuse). Next the various layers parameters are stated. However, I shall first briefly describe what this shader produces, might help. This shader creates a synthetic look for the back of the hand, the large skin region. The shader mimics the turbulent colourations of the skin, the presence of blood vessels also. The patterns that are evident in the skin at close range are also considered with individual layer patterns layered on top of each other but never interfering due to the design of the shader. How light mottles across the skin is also given. The end result is some thing that the author personally hopes synthetically represents the human skin, and powered with displacement within the shader, believes it does. Anyway, on with the explanation.

The first layer specifies parameters baseColorFreq, label and layer1Color1. These are used for ranging the frequency and amplitude as such, of a noise function that is used to control a colour spline that provides a base colour. The last parameter of this layer is the main colour value used in this layer.

Layer 2 has parameters for repeating s and t co-ordinates. These are used for controlling noised disks that are used to mimic colouration under the skin and within

its skin layers. This layer also provides for the user, a fuzz value parameter for calculations used in controlling the disks used in the layer, along with a colour value also specified for this purpose. The layer also has a transparency parameter value as does all layers.

The parameters for the layer that follows,layer 3, are identical and as such need no explanation except that a different colour is used. Layer 4 provides noise frequency and amplitudes scaling parameters for the user. As well as this, colour parameters are provided as is a transparency toggle.This layer creates a noisey middle colour layer that the author felt was synthetically needed. The layer used a filtered version of snoise, this is to prevent antialiasing [EBE98],[GRI99] , [STE03], [UPS89].

```
"float FrequencyB=9.99;
        float noisey = amp*filteredsnoise(PP*FrequencyB,filterwidthp(PP));"
```

Layers 5 has six parameters, the first are ones are for controlling the amount of blood vessels wanted in the surface. This is what this layer creates, by using smoothpulses in the s coordinate direction. The coordinates are offset prior to smoothpulse calculations, using fBm noise funtion. Again it must be indicated that the noise function id passed the filterwidth of the point shaded to crudely antialias by averaging at the Nquiest limit. The author did perform antialias where it could be clearly applied [EBE98],[GRI99] , [STE03], [UPS89].

```
"/* Offset L5ss using fBm noise, results in wiggliness value determined by  params */
          L5ss = L5ss + fBm(PP, filterwidthp(PP), octaves, lacunarity, gain);"
```

The colours that govern the blood vessels used in this layer are also provided for full user control purposes. A transparency switch is again given.

Next is the sixth layer, which creates a voronoi type pattern in the skin [EBE98],[GRI99]. Again this is a characteristic that the author feels is evident at close inspection of the skin.

```
"for(i = 0; i < j; i+=1){

/* Apply Voronoi pattern, calculate distances and positions of the two nearest features
to the noised feature postion of each cell in 3D space */
        voronoi_f1f2_3d (PP*k,1.23,f1,pos1,f2,pos2);

        f1 += f1;
        f2 += f2;
        pos1 += pos1;
        pos2 += pos2;
        k+=noise(i*.46);
        difference = f2-f1;
}"
```

The voronoi pattern is actually calculated in two loops within this layer with values being summed together. Why, well the reason was to create high frequency by summing instead of simply multiplying the point which results in cruder visual appearances. The voronoi pattern is also culled in the s direction by multiplying the

output with a smoothstep as the author fell that this is where this pattern evidently is within the area of skin in question. Parameters for controlling and providing parameters to the user for fBm noise scaling factors, as well as fuzz values used in smoothstep calculating. The fBm control wiggliness of voronoi pattern. Layer color and transparency parametrs also provided to user [EBE98],[GRI99] , [STE03], [UPS89].

Layer 7 creates lines going in the vertical direction as this is another charcateristic of this skin area. The lines are offset by adding noise to the s coordinates before tasking copies of it. Noised rotations to the lines as well as noised starting the ending positions are given to provide a more random and organic feel. The lines are also culled in the s direction using a square smoothstep by minus one smoothstep from another. [EBE98],[UPS89],

```
"/* Used to clip the lines in the s */
float L7cull = smoothstep(0.05-L7fuzzb,.1+L7fuzzb,s)-smoothstep(0.90-
L7fuzzb,.95+L7fuzzb,s);"
```

Layer 7 gives noise frequency parameters, fuzz values for stepping and a layer colour. Importantly, a parameter for controlling the degree of displacement in this layer is also given. Transparency of this layer is also provided to the user by way of a parameter, as is the layer's displacement magnitude added in each to the overall surface magnitude of displacement to be calculated.

```
"/* Current layers disp magnitude is added to overall surface disp magnitude,
        max means that previous values don't interfere with current */

surfaceDispMag = max(surfaceDispMag, layerDispMag);"

"/* Scale the transparency of this layer */
        layerOpacity *= L7Transparency; "
```

Layer 8 is a large layer creating a culled criss-cross smooth line pattern. As before these patterns are noised offset in s and t and are rotated to a noised value using the calculations from s column.The even funtion is a boolean function that returns an true if the integer returned is even.

```
"float L8noiscale = 0.01;
float L8noifreq = 3;
float L8noi = noise(s*L8noifreq,t*L8noifreq);
float L8ss = s+snoise(L8noi + 872)*L8noiscale;"

"if(even(L8col))
{
        rotate2d(L8ss,t,radians(-25)+noise(L8sTile),0.5,0.5,L8ss,L8tt);
}
else
{
        rotate2d(L8ss,t,radians(13)+noise(L8sTile-fuzz),0.5,0.5,L8ss,L8tt);
}"
```

Layer 8 provides parameters for controlling noise frequency, displacement scaling values, fuzz for smoothsteps and layer color to be used as well as transparency parameters.

Layer 9 offers parameters again for frequency for noise and parameters for the layers colour. The noise is for input into a final colour spline function for the layer.

Layer 10 gives repeating tile coordinates values to the user, as well as the usual fuzz values for smoothsteps and layer colour used for creating freckles on the skin. The freckles are created by calculated distance between the point being shaded and at specified static point. A smoothstep returning 0 to 1 depending on a radius value is based on this distance calculation. The layers transparency is also offered to the user as always. The final layers parameters control the amount of noise in this layer to offset the s coordinates of the freckles.

Layer 11 offers noise parameters for controlling fBm noise calculations which is used to offset a copy of P, PP, which is then fed as a parametr to noise the normals over the surface depending on whether the point being shaded is within the displacement value of surfaceDispMag or outside, where it can noise as much as it is specified. This layer also offers to the user whether of not to bump-map displacement of use true displacement as a parameter. The user also controls the overall surface displacement parameter of Km. Finally the shader is illuminated with shading illumination values specified and passed. Overall surfaceOpacity is also calculated.

Chapter 12 - Problems Incurred during project

Quite a few issues arose when creating the types of procedural pattern that was needed for these various shaders. Pulses lines can give real alsiasly issues as can any pattrn that repeats over a specified range. However, this was anticipated and time was taken on prevent measures. Mainly any time a line or pulse was generated, the outputs were always smoothly interpolated between. This involved using smoothstep with fuzz values, smoothpulses, filtered steps and and pulses and other such ideas.

The input to noise was also carefully considered with noise inputs of points and floats always being passed along with their filterwidth values comprising of their combined inter pixel to pixel shading averaged value. This allows for crudely anti-aliasing at the Nquist limit which prevents sharp transitions between shaded pixels. Because the issue of anti-aliasing was perceived before and during the project, measures were constantly taken to prevent this issue so it didn't become much of one.

One large problem with creating the different procedural shaders was that they comprised of many which most of time gave surprising results when trying to composite them on top of one another. This was solved by sometimes reversing the values taken, minus a smoothstep to flip its output was soundtimes used. The use of

clamping between two layers was also used as a preventive reolution to problems making sure that one layer never exceeded the other.

The largest issue by far, was in tryin to apply the shaders to a hand model in maya. Basically when the shaders were being applied they were warping over the entire surface. The author's tutor, Mr Jon Macey offered guidance here, on fixing the u,v's of the model, mapping these out correctly and normalizing when needed. However, lack of knowledge and experience pushed the author backwards and efforts proved too late. Unfortunately the shaders were not applied to the polygon model for presentation purposes and global illumination passes. Other issues involved tryin to find a way of generating a precise pattern and detail needed, but time and effort proved fruitlful here.

Chapter 13 - Results and Analysis

backPhalange2.sl: The backPhalange1.sl shader has proved difficult. The details of it, make it that way. Detail are of different scales and this has to be mimiced ofcourse but if it is done too much, then one pattern gets lost and another is all that is seen. The colour values are complimentary to skin and seem to provide as subtle look as you would find with normal skin. The horizontal waved lines seems to be too consistent and this would be the first criticism that could be made of this shader. Although the detail should exist it seems to be too regular and uniform. Noise has been added to this value but it seems seems to hold uniformity unfortunatey. Perhaps a different angle is needed altogether, where the layer is deleted and started again, making sure that the values are noised and offset from the beginning.

backPhalange1.sl: The backPhalange1.sl shader was one of the more successful shaders in that its details seem to be more subtle or perhaps, less pinpricked. The shader provides all the texture pattern details that are eminent in this area of skin and all layers are composited well with no layer bombarding upon another. The author seeks to get this look for all other shaders at they are close but maybe the values passed into this shaders functions, such as noise values, just yielded smoother results. All the texture patterns provide no colour lookup, which the author believes is key, as when applied it provides small interfering results with a little bit of unwanted noise between patterns when they meet. This could be rectified by smoothing the individual layers patterns a bit more but it is felt that having displacement, provides enough detail.

nail.sl: This is by far the poorest shader I feel, and am quite cautious in its use. The texture pattern is not as detailed as I would like it. The patterns generated by different layers don't seem to come together at all really. The colours are also unexpected, they are either too loud or too dull, as the author has difficulty in

controlling this shader even though it is the smallest one. Perhaps this is an example of feeding poor values in and as such getting poor results back. The patterns provided by the layers get somewhere close to what is expected of a nail shader but the finer details are lacking and as such needs more work.

metaCarpus.sl: This shader, for the back of the hand is the largest of all the shaders. The patterns generated and colour value are close to what should be expected. The patterns have been culled to different areas as this occurs in the true skin area. The patterns do get very busy at the bottom of the shader so smoothing and transparency has been used to control this from getting too messy. Although trou skin is very compact and high in pattern frequency in the lower half of this area so it is closely emanating true skin.

backPhalange1_2: This is a texture lookup type shader. Although it depends on a texture for its main pattern, which is the detail between the bones of phalanges row one and two, the colour value are all clearly calculated. The texture only provides displacement and diffuse lookups. Furthermore the transparency parameter scaling the opacity of every layer of every shader works very well here. The reason is that the previous layers pass through this layer, that looks up the images, very well. The blood vessels that are being created can be controlled as to how much travels through via transparency parameters of both layers. This of course is true of real skin as the different layers all seem to provides certain amount of translucency allowing the underlying colour to bleed through. This one of the important aspects of the shader design, as the author believed it would provide high control and flexibility. Overall very controllable and successful in the purpose.

Overall the author would argue that all shaders stayed very strongly with designs set out, and all came together well to provide the details that were trying to be reached be creation and implementation. The shaders have been tested and show no signs of antialiasing artefacts, which was key in the design and creation stages. Furthermore testing of very high frequencies of either noise is repeating the textures generated still yielded no visible signs of antialiasing. It is hoped that if the shader was animated over a surface, any antialiasing would be minute or unseen, as it has shown no presence in testing. The shaders colour which is fully controllable also comes close to what the author wanted. People feel that skin is pink in intensity value when really,the author suggests, their seems to more whites and red evident in the skin, dotted arund like high frequency noise. Besides anything, the shaders come very close to this approximation and hopefully will look as photo-realistic on a polygon model, as they due under the spot light at the moment.

Chapter 14 - Conclusion

It is concluded that although the applying of the shaders onto a model for presenting in a more realistic way, was not possible. It did not result in a failed project, as at the heart of the project was to create skin shaders for the hand, and that was successful to a positive extent in the authors opinion. The colour,appearance and displacement of these shaders to cover very small areas of the hand, matched the design specification that the author created. Only some mild changes with only one additional layer being added to one shader for an added effect of noisey colour, but the texture patterns sought out were created and represented what was needed in requiring a photo-realistic shader of skin on the hand.

Chapter 15 – References

[BIR99] BIRN, J. 1999. Digital Lighting and Rendering. Publisher: New Riders (31 Jul 1999). ISBN: 1562059548

[BLE04] Blevins, N. Translucency and Sub-Surface Scattering. 2004. http://www.neilblevins.com/cg_education/translucency/translucency.htm

[BLI82] BLINN, J. F. 1982. Light reflection functions for simulation of clouds and dusty
surfaces. In Computer Graphics (Proceedings of ACM SIGGRAPH 1982), ACM, vol. 16, 21–29.

[EBE98] July 1998. David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, Steven Worley. Texturing & Modeling A Procedural Approach Second Edition. AP Professional. ISBN 0-12-228730-4

[GRI99] 1999. Anthony A. Apodaca, Larry Gritz. Advanced RenderMan: Creating CGI for Motion Pictures (The Morgan Kaufmann Series in Computer Graphics). ISBN: 1558606181

[HAN93] HANRAHAN, P., AND KRUEGER, W. 1993. Reflection from layered surfaces
due to subsurface scattering. In Proceedings of ACM SIGGRAPH 1999, ACM Press/Addison-Wesley Publishing Co., New York, Computer Graphics Proceedings, 164–174.

[HAA92] HAASE, C. S., AND MEYER, G. W. 1992. Modeling pigmented materials for realistic image synthesis. ACM Trans. Graphic. 11, 4, 305–335.

[JEN02] JENSEN, H. W., AND BUHLER, J. 2002. A rapid hierarchical rendering technique for translucent materials. ACM Trans. Graphic. 21, 576–581.

[DOR99] DORSEY, J., JENSEN, H. W. AND LEGAKIS, J. 1999. Rendering of wet materials. In Rendering Techniques ’99, 273–282.

[JEN01] JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In Proceedings of ACM SIGGRAPH 2001, ACM Press/Addison-Wesley Publishing Co., New York, Computer Graphics Proceedings, 511–518.

[LAN05] Lancaster, T. 2005. RenderManRepositoty.
http://www.renderman.org/RMR/index.html

[MAY96] Stephen F. May. "Writing RenderMan Shaders - Why follow a methodolgy?";, Copyright © 1995, 1996. *RManNotes*

[NIC77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), October 1977.

[PHA00] Pharr, M. Hanrahan, P. Monte Carlo evaluation of non-linear scattering equations for subsurface reflection. In Computer Graphics Proceedings, Annual Conference Series, 2000, pages 75–84, July 2000.

[PIX06] RenderMan Pro Server 12.5. Application Notes #37 Translucency & Subsurface Scattering. Pixar. 2006

[STA95] STAM, J. June 1995. Multiple scattering as a diffusion process. In Eurographics Rendering Workshop 1995. Eurographics,

[STA01] STAM, J. 2001. An illumination model for a skin layer bounded by rough surfaces. In Proceedings of the 12th Eurographics Workshop on Rendering, 39–52.

[STE03] 2003. Ian Stephenson. Essential Renderman Fast. Springer; 1 edition. ISBN: 1852336080

[UPS89] Upstill, Steve. 1989. The RenderMan Companion - A Programmer's Guide to Realistic Computer Graphics, 277-278; Addison Wesley.

[WIK06a] Hand. http://en.wikipedia.org/wiki/Hand. Wikipedia®. 2006.

[WIK06b] Skin. http://en.wikipedia.org/wiki/Skin. Wikipedia®. 2006.

[WIK06c] Nail. http://en.wikipedia.org/wiki/Nail_%28anatomy%29. Wikipedia®. 2006.

Appendix i

The BRDF is a generalized approximation of BSSRDF, indicating that light enters and exits the same surface point. Given the BSSRDF equation, we can see that S, BSSRDF, relates the outgoing radiance, $L_o(x_o, \vec{\omega}_o)$ at point x_o in the direction of $\vec{\omega}_o$, to the incident flux, $\Phi_i(x_i, \vec{\omega}_i)$, at the point x_i, from the direction $\vec{\omega}_i$, as shown below:

$$dL_o(x_o, \vec{\omega}_o) = S(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)\, d\Phi_i(x_i, \vec{\omega}_i).$$

[JEN01],[DOR99]

The outgoing radiance of BSSRDF, $L_o(x_o, \vec{\omega}_o)$ is computed by integrating the incident radiance over incoming directions and area, A:

$$L_o(x_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)\, L_i(x_i, \vec{\omega}_i)\, (\vec{n} \cdot \vec{\omega}_i)\, d\omega_i dA(x_i)$$

The radiative transport equation, also known as the volume rendering equation, describes the propagation of light in a medium:

$$(\vec{\omega}\cdot\vec{\nabla})L(x,\vec{\omega}) = -\sigma_t L(x,\vec{\omega}) + \sigma_s \int_{4\pi} p(\vec{\omega},\vec{\omega}')L(x,\vec{\omega}')\,d\omega' + Q(x,\vec{\omega})$$

The absorption coefficient σ_a, scattering coefficient σ_s, and phase function $p(\vec{\omega},\vec{\omega}')$ of this equation describes the properties of the medium. With $\sigma_t = \sigma_a + \sigma_s$, indicating the extinction coefficient σ_t. The phase function needs to be normalized, with $\int_{4\pi} p(\vec{\omega},\vec{\omega}')\,d\omega' = 1$, and is a function of the phase angle, $p(\vec{\omega},\vec{\omega}') = p(\vec{\omega}\cdot\vec{\omega}')$. The mean cosine of the scattering angle, indicated by g, is:

$$g = \int_{4\pi} (\vec{\omega}\cdot\vec{\omega}')p(\vec{\omega}\cdot\vec{\omega}')\,d\omega'.$$

If g is positive, then the phase function will be mostly forward scattering. If g is constant, (g = 0), the the scattering is isotropic (same in all directions). If g is negative, then the scattering will be mostly backward scattering.

With an infinitesimal beam entering a homogeneous (the same everywhere) material, the incoming radiance will decrease exponentially with distance s, referred to as *reduced intensity*:

$$L_{ri}(x_i + s\vec{\omega}_i, \vec{\omega}_i) = e^{-\sigma_t s} L_i(x_i, \vec{\omega}_i).$$

The first-order scattering of the reduced intensity, Lri, may be treated as a volumetric source:

$$Q(x,\vec{\omega}) = \sigma_s \int_{4\pi} p(\vec{\omega}',\vec{\omega}) L_{ri}(x,\vec{\omega}')\,d\omega'.$$

To gain insight into the volumetric behavior of light propagation, it is useful to integrate the radiative transport equation over all directions $\vec{\omega}$ at a point x which yields

$$\vec{\nabla}\cdot\vec{E}(x) = -\sigma_a\phi(x) + Q_0(x). \qquad (1)$$

This equation relates the scalar irradiance $\phi(x) = \int_{4\pi} L(x,\vec{\omega})\,d\omega,$ and the vector irradiance, $\vec{E}(x) = \int_{4\pi} L(x,\vec{\omega})\vec{\omega}\,d\omega.$. With the absence of loss due to absorption or gain from the volumetric light source $(Q_0 = 0)$, the divergence of the irradiance vector equals zero. This gives the 0th term, Q_0, with the 1st-order being

$$Q_0(x) = \int_{4\pi} Q(x,\vec{\omega})\,d\omega, \quad \vec{Q}_1(x) = \int_{4\pi} Q(x,\vec{\omega})\vec{\omega}\,d\omega.$$

Light transport in highly scattering material tends to become isotropic, with diffusion approximation based upon this idea. Even if the initial light source and phase function are anisotropic, the light source still results in being isotropic with high scattering mediums. With each scattering event, the light distribution is blurred, and results in it being uniform as the scattering events increases. With this, the radiance may be approximated with a two term expansion involving radiant fluence ϕ and the vector irradiance $\vec{E}$:

$$L(x,\vec{\omega}) = \frac{1}{4\pi}\phi(x) + \frac{3}{4\pi}\vec{\omega}\cdot\vec{E}(x).$$

The constants of this equation are determined by the definitions of fluence and vector irradiance. The diffusion approximation extends from this approximation, with this two-term expansion of the radiance being substituted for the radiative transport equation, then integrating over $\vec{\omega}$.

$$\vec{\nabla}\phi(x) = -3\sigma_t'\vec{E}(x) + \vec{Q}_1(x). \qquad (2)$$

Here, a reduced σ_t' extinction coefficient has been incorporated:

$$\sigma_t' = \sigma_s' + \sigma_a \quad \text{where} \quad \sigma_s' = \sigma_s(1-g)\,.$$

The scattering coefficient, σ_s, has been replaced with a reduced scattering coeeficient σ_s', with a factor of (1 - g). g being the mean cosine of the scattering angle. This is because, once light becomes isotropic, due to the propagation through a multiple scattering medium, only backward scattering terms can change the net flux; with forward scattering not being seen[JEN01],[DOR99].

When there are no sources, or where the sources are isotropic, $\vec{Q}_1$ vanishes from the radiative transport equation, with the vector irradiance being the gradient of the scalar fluence,

$$\vec{E}(x) = -D\vec{\nabla}\phi(x).$$

Where $D = \frac{1}{3\sigma_t'}$ is the diffusion constant. By substituting Equation 2 into Equation 1 we arrive at the classic diffusion equation:

$$\vec{\nabla}\phi(x) = -3\sigma_t'\vec{E}(x) + \vec{Q}_1(x). \qquad (2)$$

$$\vec{\nabla}\cdot\vec{E}(x) = -\sigma_a\phi(x) + Q_0(x). \qquad (1)$$

$$D\nabla^2\phi(x) = \sigma_a\phi(x) - Q_0(x) + 3D\vec{\nabla}\cdot\vec{Q}_1(x).$$

[JEN01],[DOR99]