Matt Birkett-Smith

Innovations Project - Research into photoreal head creation

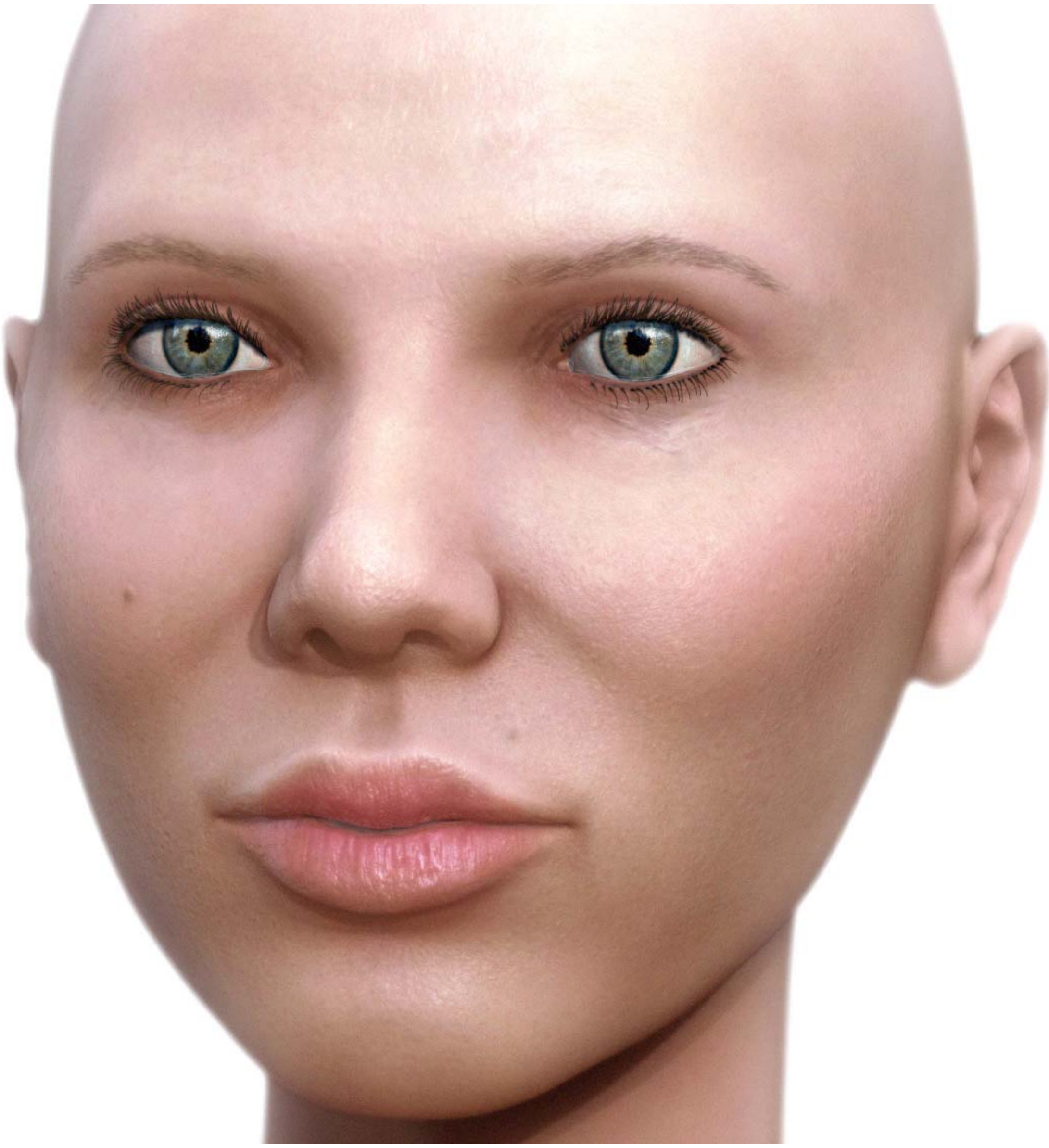
Introduction

What is this document?

This is a part research part tutorial document written for a university Innovations project. I aim to look at and discuss most topics and techniques that need to be considered when working on realistic head creation and recommend what I believe are currently the most effective workflows.

Why should I be interested?

Good question, I can't claim to be an ultimate authority on any of the topics I go through, I just feel that a lot of people would benefit from a centralised collection of techniques. As to whether this is good advice or not I can't answer in an unbiased way. However when writing this tutorial I made my own model to use as a testbed for techniques, the end result being the render seen below. If you're interested at all in how I made any of it then I would hope you will find this tutorial interesting and useful.



Latest render of my female head model loosely based on the actress Scarlett Johansson

Who is it for?

The material I will be going through will be pitched at a number of different levels. I will be assuming the reader has a certain amount of proficiency with these areas but I will also try to explain concepts that some may find confusing. What I will not be doing is going through step by step software specific instructions, as an example I will not be listing all the operations necessary to model a head, I will instead recommend different things to consider when doing it.

What does it cover?

Throughout the tutorial I will be going through a number of different areas, the rough contents are:

- Modelling Basics – Rules and guidelines for clean organic modelling.
- Modelling Intermediate – Advice on edge loop theory and placement for head modelling.
- UVing – Importance of good Uving, recommend unwrap techniques and strategies.
- Texturing – A look at what maps are necessary and advice for painting them effectively.
- Shading – Different approaches to tackling skin shading.
- Rendering and Compositing – How to render efficiently and make good use of post adjustments.
- Miscellaneous – A look at various topics such as hair, eyes, teeth, skin sliding and driven bump/displacement. Note, due to time constraints this section is still a work in progress.
- Summary – A recap of my overall workflow.

You may notice that I am not including anything on facial setup or animation, this is outside my area of experience and beyond the scope of my tutorial. However most of the pipeline I will be going through will be done in a way that allows maximum possible flexibilty with character setup.

It's also worth considering that although I am only specifically looking at techniques around facial work a lot of these ideas translate very well to a full body.

What software is it for?

Throughout the document I will be basing my comments around a Maya polygonal workflow, rendered in Mental Ray or Maya Software and touching on elements of Photoshop, Shake and Zbrush. However, most of the techniques that are mentioned aren't platform specific and should be relevant to other software packages.

Modelling Basics

What to model in?

One of the first choices to make when starting an organic model is whether to use NURBS, polygons or subD's. Each of these techniques can drastically affect your workflow and they all come with their advantages and disadvantages.

NURBS

Pros

- Instant, precise curved surfaces
- Innate UV co-ordinates
- Infinite level of detail
- Very fast in some renderers

Cons

- All topology must essentially be a plane
- To make non trivial surfaces you need to stitch together lots of patches
- Patches need to be carefully sewn together to ensure continuity
- Inflexible UV co-ordinates, modelling needs to be done with Uving in mind
- Fairly inefficient in the Maya renderer

POLYGONS

Pros

- Very fast and flexible
- Allows for arbitrary topology flow
- Can be UV'd in a very flexible way
- Essential for current game engines

Cons

- UV's need to be manually created
- High level smoothing is necessary for curved surfaces
- No automatic level of detail

SUBDIVISION SURFACES

Pros

- Essentially the same advantages as polygons
- But can be rendered with automatic smoothing
- Hierarchical editing independent of current subdivision level

Cons

- Slower viewport feedback
- Can be initially confusing to work with
- Again no innate UV co-ordinates

Personally speaking I would heavily advice against the use of NURBS surfaces when head modelling. From my own experience I've never found that the benefits of NURBS outweigh their substantial weaknesses. The main problem's with NURBS is that you cannot make arbitrary topology, everything has to be essentially an ordered grid. To get round this you have to use multiple patches which then causes more complications when it comes to surface continuity and to UVing. Each patch will have separate UV co-ordinates which creates a lot of problems when it comes to texturing.

This then leaves the choice between polygons and SubD's, personally I'm still undecided as to which offers the best workflow. I would essentially say that SubD's are an extension of polygons and that all modellers should first learn to use polygons as all the skills are transferable. Then once they feel confident with polygon modelling they should look into SubD's more thoroughly. If you do decide to use polygons then it is essential you model with smoothing in mind (particularly when working on organic surfaces), a polygon surface should be periodically checked to ensure it smooths correctly without losing its shape or volume. Also it's worth noting that a polygon mesh can always be converted to a SubD at any point, this can even be done at rendertime using Mental Ray's approximation editor.

Finally it should be reiterated someone with good 3d skills would be able to create excellent results regardless of what they are modelling in.
Throughout the rest of the document I will be referring to what is essentially a polygons/subD's workflow, however a significant amount of this should be transferable to a NURBS workflow.

An example of superb modeling skills with NURBS to show that it is a viable surface type, just that in my opinion I find them too restrictive. Note the larger blue wires were multiple patches have to be sewn together. http://forums.cgsociety.org/showthread.php?t=307349

Starting a head model

One of the main stumbling blocks people get when modelling is how to start, it is always very daunting starting a model and there are always countless different ways to achieve things. In my opinion it really doesn't matter how you start a model, no matter what you do it will take a lot of work to get it looking good and no matter what initial techniques you use you are still going to have to continually rework stuff to get it looking right. Modelling is a very iterative thing, it is by and large all about playing around with shapes until they look right. Try to avoid following step by step tutorials when modelling, you don't really learn much in the process and it means that if you do have to make something by yourself you often don't have enough confidence in what you are making as it isn't being done in an “approved” way.

Personally speaking I don't use any fancy modelling techniques, for anything organic I usually start with the create polygon tool, create the outline and then the real work is largely done with just split poly and merge vertices. A fairly good technique is outlined here

Other people start off with box modelling, others prefer to model everything separately and stitch it together, personally I prefer to get the overall shape right first and then progressively refine it from there. No particular modelling techniques are perfect though, just use whatever works for you.

Modelling tutorial links

http://www.thehobbitguy.com/tutorials/polymodeling/index.html
http://www.3dm3.com/tutorials/maya/character/
http://www.3dm3.com/tutorials/maya/dobby/

Reference:
http://www.cgtalk.com/showthread.php?t=38469
http://www.cgtalk.com/showthread.php?t=108412

Topology Theory

Topology and edge loop theory are a big topic but essentially it boils down to a fairly simple concept,- that the placement and flow of edges of a surfaces is done in the most logical and efficient way to capture the detail of the surface and also fits the way the object will deform. A good topology will have edges that flow smoothly in an easily traceable pattern and appear to capture the form of the object in the most naturally optimal way possible. A bad topology will feature irregular shapes, edges which flow against details, smoothing errors, inappropriately high poly counts and often massive inconsistencies between the size of faces.

A good topology is beneficial for many things, these include:

- Good deformation
- Efficient use of geometry
- Easy to modify
- Nicer to UV
- Simpler to get a feel for the surface by eye

On 'hard' objects, i.e. ones that don’t deform, a good topology is nice but not vital, on organic objects which have to be animated it is absolutely essential. The bottom line is that if you have edges flowing and creasing in an unnatural way then it becomes difficult to impossible to get it to deform correctly, regardless of how it is weighted. When doing facial modelling the main thing to consider about your topology is where you will encounter creases and details on the face, it is very hard to add things like wrinkles efficiently if they don’t flow with your edges. It can be done but its really bad for your polygon count. Things like facial wrinkles should be your defining start point for how your topology should be arranged. However note that there may be some tradeoff between fitting in details and getting good deformation. Its also worth noting that there is not one ultimate human topology technique, men and women differ quite a lot and faces vary greatly, however there are consistent areas that should be looked out for.

Quads, Tri's and ngons

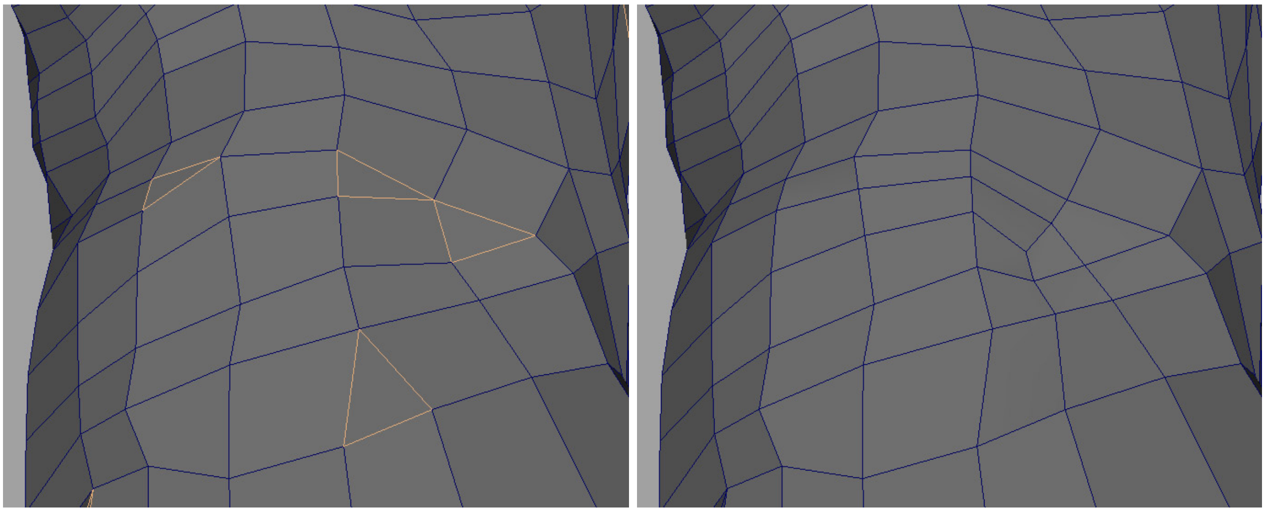
Quads:
A quad is a face with 4 edges, these are the fundamental building block of modelling and should be used wherever possible. Quads can be consistently deformed and shaded and are faster to UV, it is also much easier to visually work out the edge flow of a model if it's made out of quads.

Ngons (5+ sided):
These can create shading/smoothing glitches and are awkward to get to smooth correctly. Ngons should be used sparingly, if at all, with the understanding that they should either be hidden, or in a place that doesn't deform much. Not a good idea if you're making a low poly or game model.

Triangles:
Purely theoretically triangles are an excellent shape in that they are completely consistent in their shading, all geometry is converted to triangles at rendertime. However generally speaking triangles shouldn't be used as a primary shape for modelling, it is difficult to get a triangle mesh to shade and deform consistently unless the triangles themselves form quads. Triangles are often a very useful shape to use when merging edges together, however this can often also be done using quads. Use tris if you can tolerate their behavior. Usually, the head is a place where there should be very few, preferably none. Heavy use of triangles should only be considered in seriously low poly count scenarios.

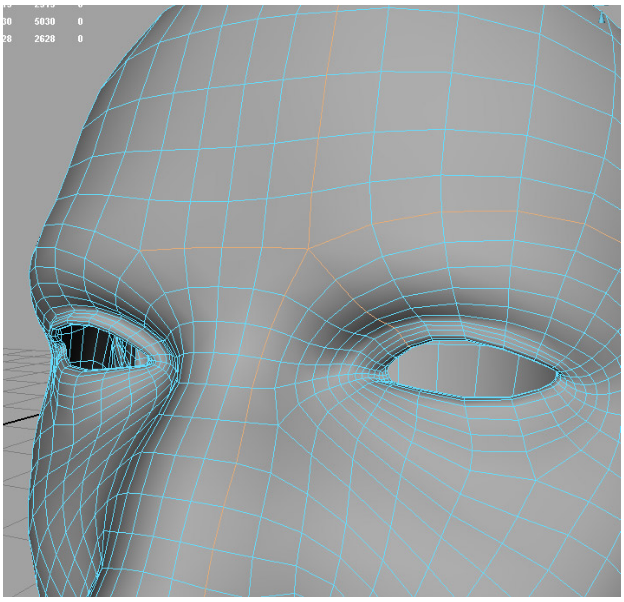
All in all, no tris or N-gons is the best. Few is acceptable, but many is lazy. As a ballpark figure a model should be at least 95% quads. It should be noted that there is often a trade off of slightly increased poly count when trying to make a pure quad mesh.

Generally speaking when modelling you should be careful to avoid making triangles from an early stage but at the same time not get too fixated on it. Once the model is more or less done it is often useful to go through a cleanup stage where you try and improve edge flow and if possible get rid of triangles. To turn triangles into quads you generally look to get two triangles together and then turn them into a quad. One triangle by itself cannot be removed just by rerouting edges. However you can achieve this by running a new edge from the triangle to a border edge but this isn't always very efficient.



A crude example of how multiple triangles can be redirected and combined to form quads

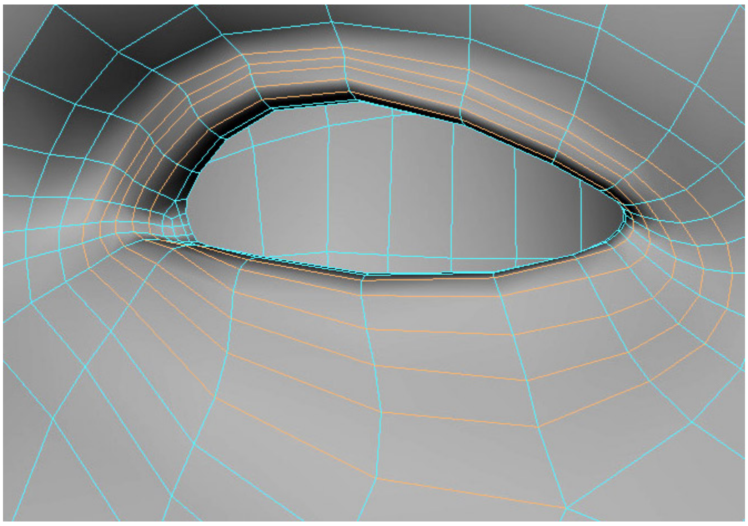
Stars:
Stars are a offshoot shape created when modelling in pure poly's, they occur when 5 edges join one vertex and are normally only present when a lot of edge flow needs to be rerouted around a small area. By and large stars are relatively harmless, they will smooth and shade correctly in most scenario's. However, when heavily subdivided stars can cause pinching (particularly in zbrush) and also tweaking points around a star can be delicate. Generally speaking stars aren't worth worrying about too much but if possible they shouldn't occur in too prominent a place on the model.



An example of a star found where many edges join

Spirals:
A spiral is where you have many edge loops running radially around a point but instead of each loop creating a closed circle they create a continuous spiral. Theoretically speaking there is nothing inherently wrong with them, they will smooth and shade perfectly correctly. However actually modelling with and around spirals can be a real headache, edge loops become harder to select and visualise plus adding localised detail in a spiral is a real problem. If you can tolerate modelling with them then there is nothing wrong with spirals but generally they should be avoided just for their awkwardness.

One other thing to be aware of when analysing your geometry is that generally speaking you should try and get your faces to have similar areas. A very large face next to a very small one will smooth oddly and can also cause problems if UVs are smoothed.

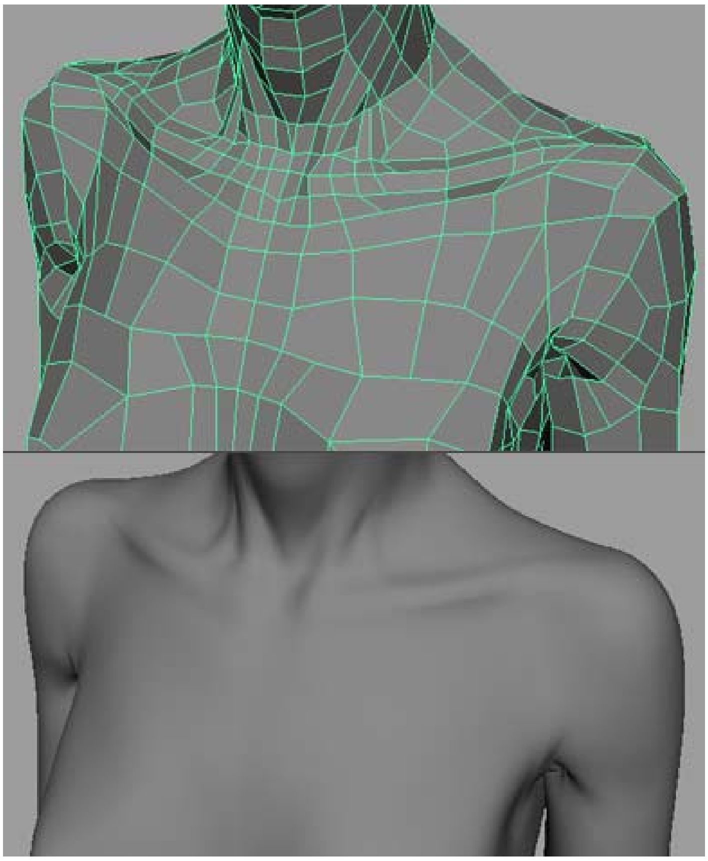


An example of a spiral located around the eye

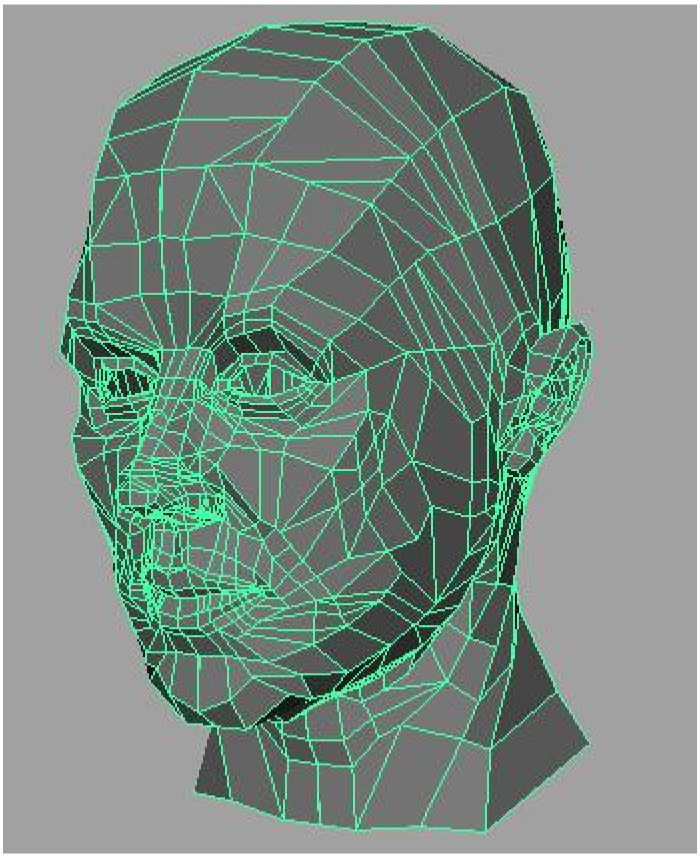
Overall you should always bare in mind what shapes you are making when modelling, however these aren't completely hard and fast rules, just use what works for you. There is a particular amount of controversy between how necessary pure quad modelling is, to quote Steven Stahlberg (a semi famous character modeller, see http://www.androidblues.com/).

“3 and 5 sided is perfectly ok. What is more important is the relative distance between edges. They have to be evenly spaced, and as few as possible, in areas that have to be smooth. Therefore, since 3 and 5 sides mixed in with quads mean the edges aren't completely evenly spaced, you try to keep them away from those areas. Like for instance, breast, butt, middle of a bulging muscle, etc. “

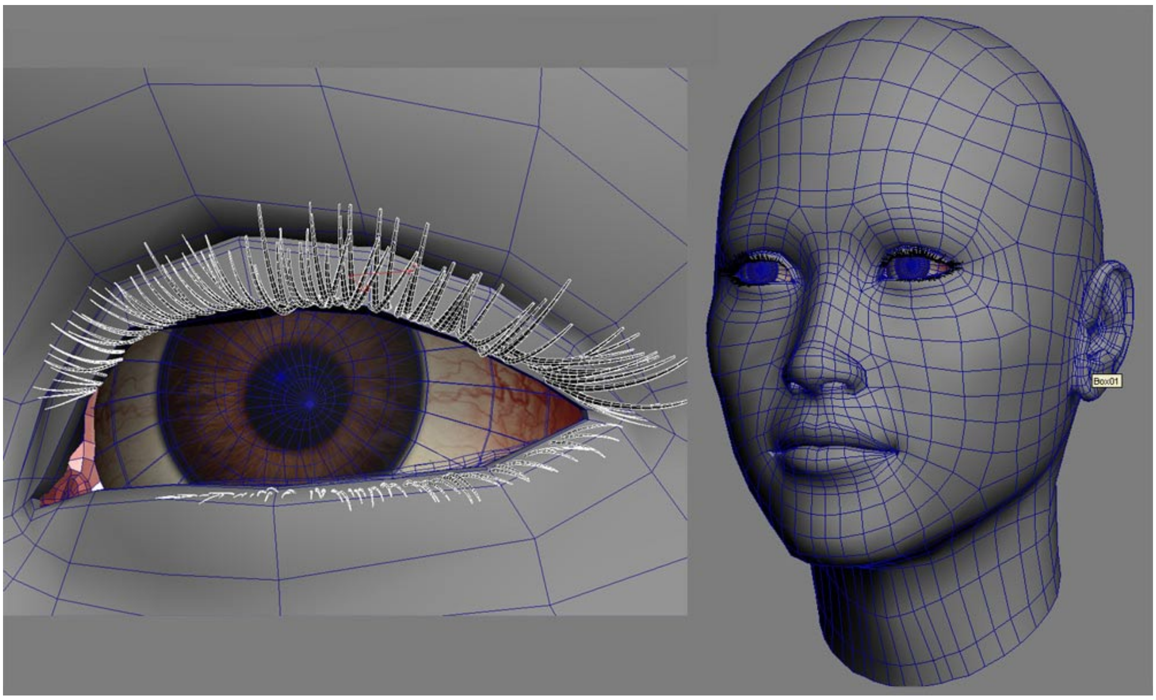
The argument for 3 and 5 sided is that when geometry is smoothed in Maya it becomes a pure quad mesh anyway so it isn't really worth worrying about. If you can tolerate the look and behaviour of triangles and 5 sided shapes then it probably isn't too much of a problem, it's likely that you can get a slightly lighter mesh using a mix than if you were sticking to pure quads. Personally speaking I'm unconvinced with this argument and I'm still a fan of using as close to pure quads as possible, if you intend to take your work into zbrush then pure quads is more or less essential, plus being able to model in pure quads is a very useful skill to learn early on and is necessary in a lot of pipelines.



An example of a typical 3 and 5 sided Stahlberg mesh



An example of a model which is proportionally well modelled but with unreadable topology



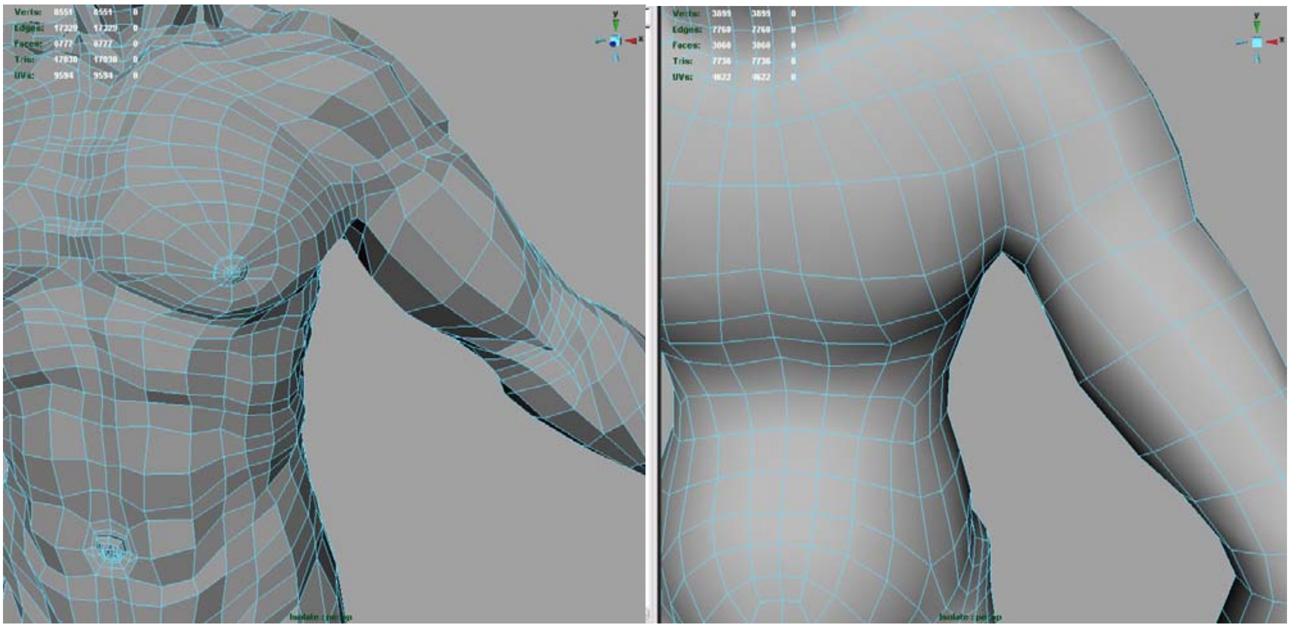
An example of a model which with a very clean and readable quad heavy topology

Modelling Intermediate

Edge Loop modelling techniques

The term edge loop modelling simply means modelling with an awareness of edge loop flow, trying to get edge loops to flow with and along key features of the model.

Generally speaking there are two different ways to approach edge loop modelling, you can either hard model edges along muscle flow or you can choose to only worry about rough body shape and mass in your topology and attempt to recreate fine detail with textures.



Two different approaches to modeling a chest and shoulder, on the left is more traditional hard edge modeling where all major muscles groups have been created so they crease correctly when smoothed. On the right is a model done with body mass modeling where the only real consideration is the silhouette and volume of the figure, surface detail is largely ignored. Note the complete absence of details like the nipples and the belly button, this allows a much cleaner and more simplified edge flow.

When hard modelling every bit of detail with edges you have to be very aware of correct anatomy and where all the main muscle groups on the body reside. Edge flow will essentially follow the outline of each muscle group. With this kind of hard modelling it is possible to get a fantastic amount of detail, however a lot of CG models suffer in that all this detail is overly sharp and defined as the modeller has made sure you can pick out how much work has gone into picking out each muscle group.

Pro's
- Possible to get extreme amounts of detail without reliance on any textures
- Muscles generally speaking can be made to react very realistically if correctly rigged

Con's
- Requires quite a large base polycount
- Time consuming
- Hard to rig accurately
- Tendency for details to be over defined
- Very hard to simulate skin sliding over these details

With body mass modelling the topology tends to be heavily simplified and much more flowing, the main emphasis is spent simply on getting the right volume and profile of the object without getting bogged down in intricate details unless they greatly affect the silhouette of the object. The idea is that these kind of fine details are created using bump or displacement maps or just through well painted colour maps.

Pro's

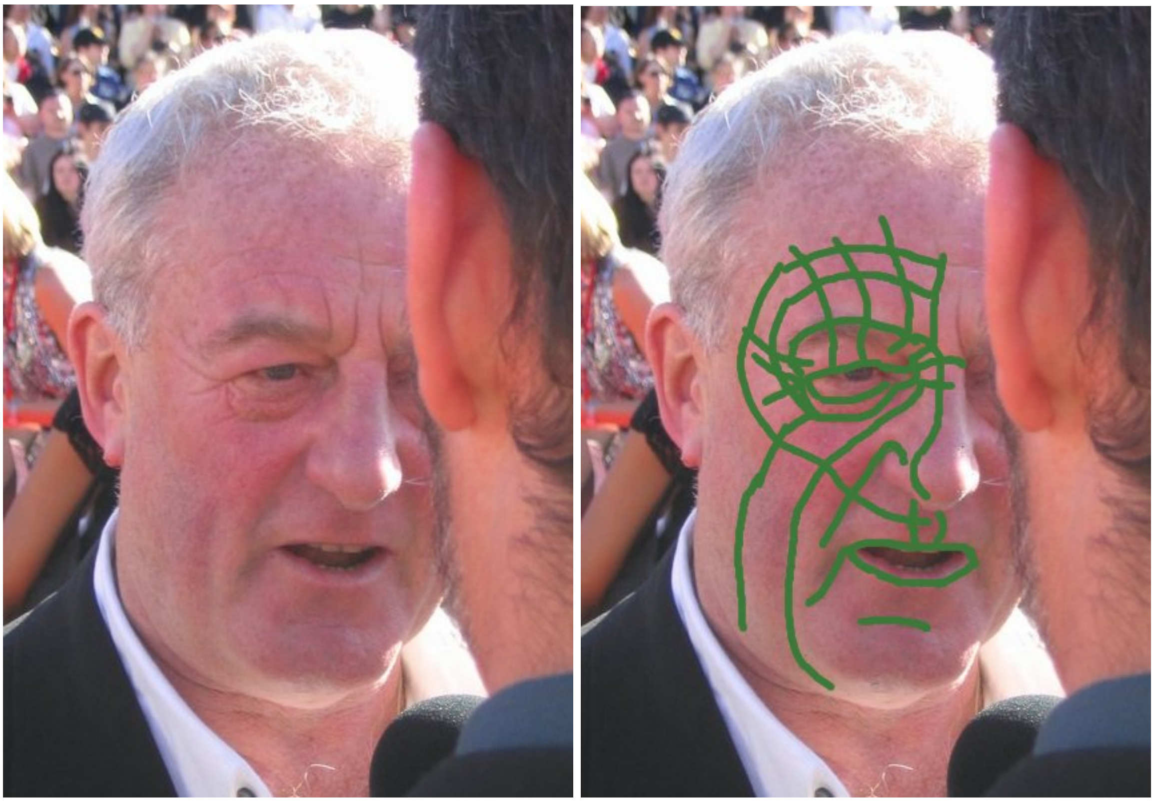
- Light base mesh – very suited to games
- Very fast to create base mesh
- Simple to UV and Rig

Con's

- Requires good bump or displacement to be effective, these can be hard to create
- To create any kind of muscle deformation different bump/displacement maps need to be blended in

With facial modelling it is wise to consider where you want to lie between these two techniques, it is generally a good idea to hard model a lot of the key features of the face so that these can be animated relatively easily. However it is also important to be aware that textures can do a lot of work for you and that you shouldn't get too bogged down in modelling every tiny bit of surface detail.

This ties in to a question which is often asked but has no real answer: How much detail do I need to model? To answer this you have to consider how much detail is required, how much setup time you have and what restrictions you have on poly count/render times. I would advise that the majority of small facial wrinkles shouldn't be hard modelled into the surface. However, the direction of these wrinkles should be considered when constructing facial topology as they are good indicators for how the skin folds.
Despite this there are at least two wrinkle areas that in my opinion should be included in facial topology, these are the nasolabial fold (often referred to as smile line) and the infraorbital fold (best described as the diagonal crease from the inner corner of your eye). Wrinkles along the forehead could be hard modelled if the character requires it as this generally isn't too much of a problem with the existing edge flow. Areas like crow's feet and nose wrinkling are definitely to be avoided when hard modelling, these will seriously confuse edge flow.



Note how wrinkles can often give a good indication to the underlying structure of a face, it is often beneficial to use them as a guideline for topology

All people share fundamentally the same facial structure and will all crease and wrinkle in more or less the same places, just that some are more prominent than others

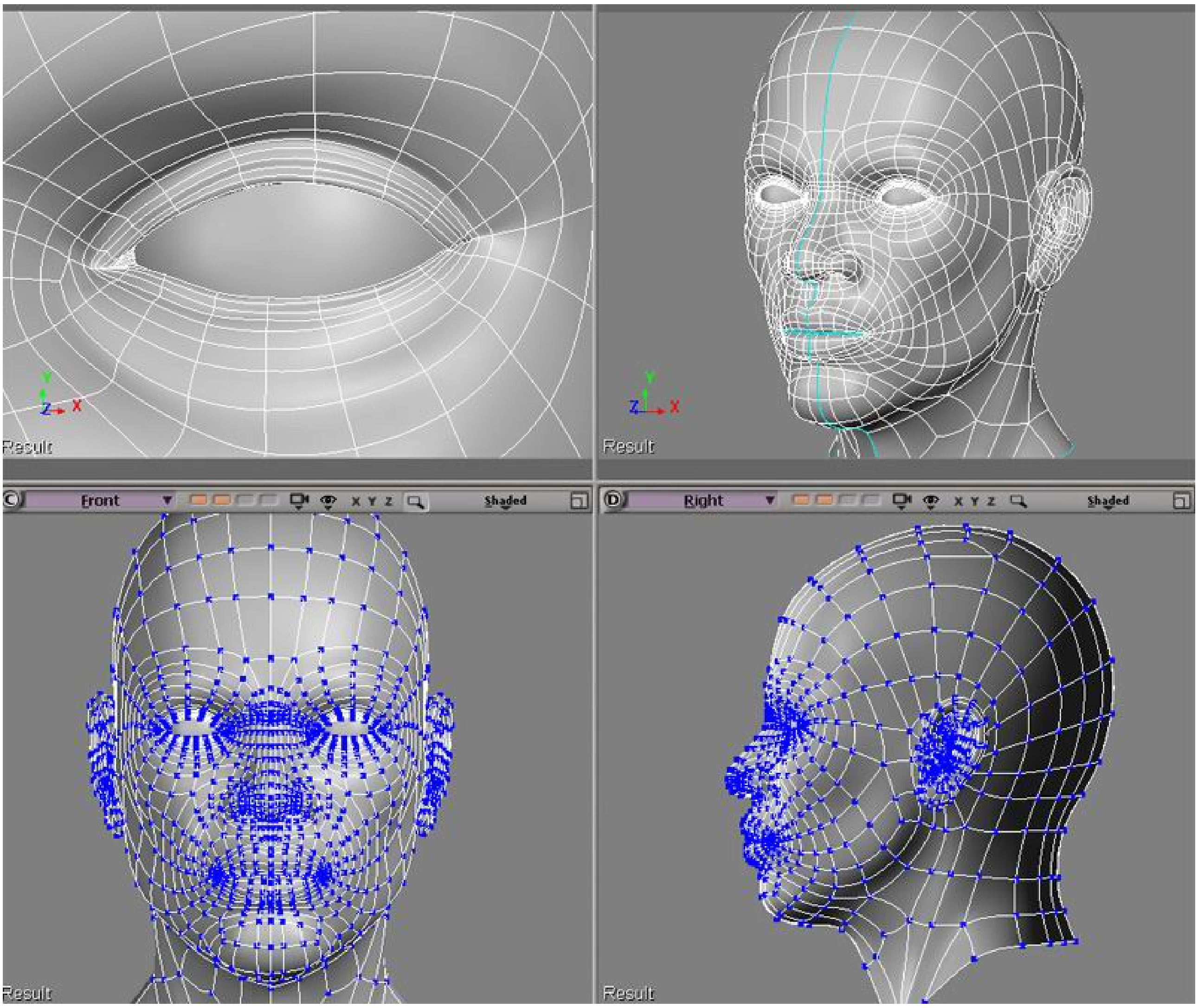
A very brief diagram of wrinkles indicating which should be considered and which discarded for facial modeling. Blue is essential and green is heavily recommended while red should be ignored

Loops for head modelling

As stated there is no hard and fast “perfect” edge flow for facial shapes, however there are a lot of consistent techniques that should be considered for the majority of face types.

“Goggle” Technique

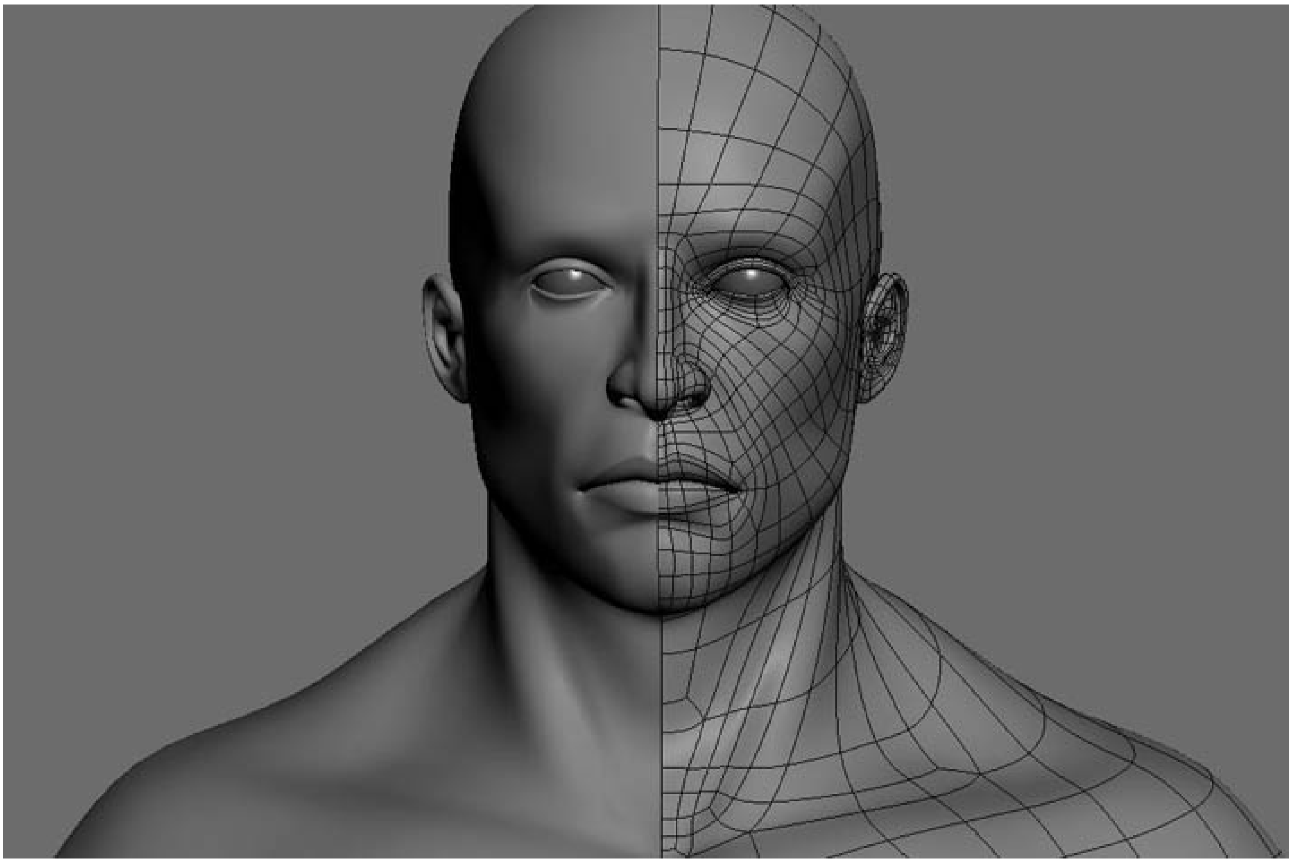
This is a broad name for a topology where the eyes are treated as the centre of a circle and many concentric edge loops follow around it. It's often used in conjunction with the mouth being the centre of a circle too. The idea being that the topology will be pretty solid if these circles are used for the basis and that most facial expressions can be created by this.
However there are quite a few drawbacks of the “goggle” techniques, the first being that the cheek lies inbetween two circles making the edge flow of this area usually quite confusing to work out. Also, because the edge loops are following a circle the definition of the nasolabial fold is often too curved and not following the anatomically correct path. The other main problem is that as the eyes are a circle it is impossible to include the infraorbital fold as this runs diagonally to the circle. The combination of these two folds often means that a character modelled with the “goggle” base topology is often too smooth as correct folds cannot be incorporated.



A very good, clean, implementation of the “goggle” topology technique. Although the model is very strong note the slightly off position of the smile line, the change of direction in the cheeks as the mouth loops meet the eyes and the lack of a crease coming from the corner of the eye.

"Crossover" Technique

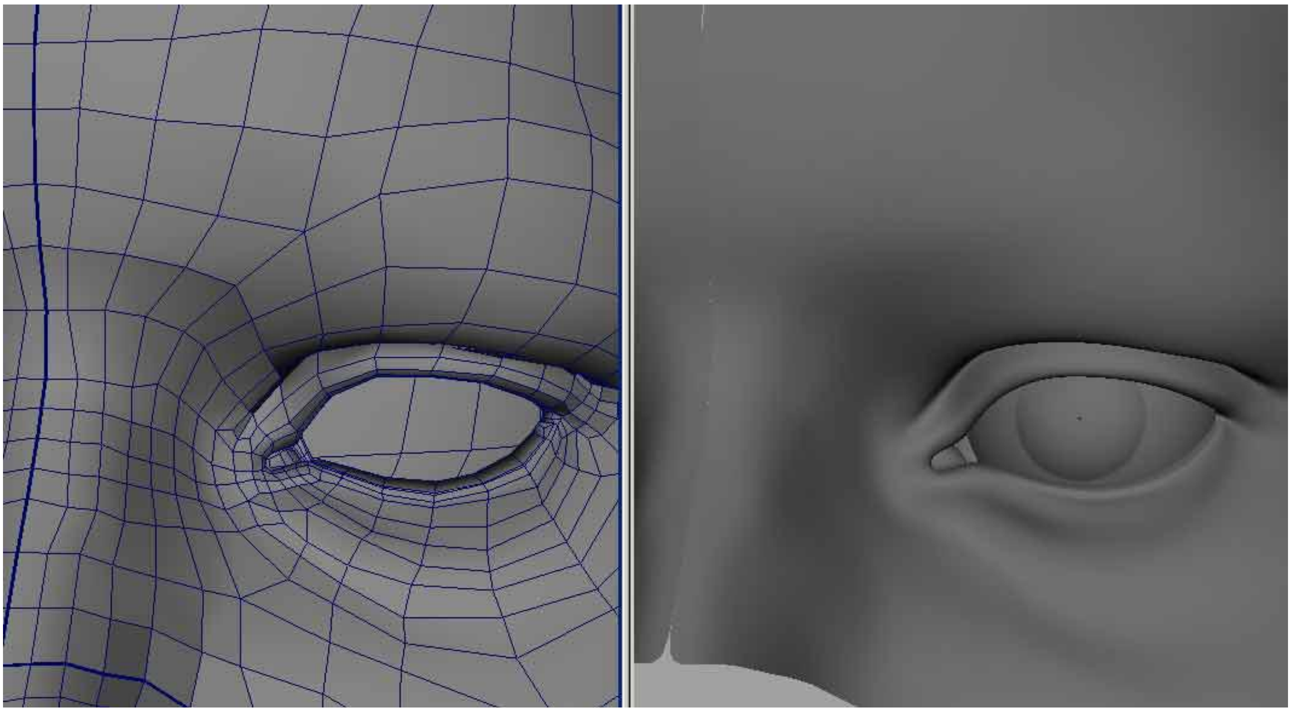
The idea of this technique is that instead of the eye being the centre of a circle it is now the centre of an edge which loops round the top of itself and crosses back over itself as it comes back down the cheek. The idea is that in this way you can get a diagonal edge following the infraorbital fold. It also becomes easier to construct the cheek and the nasolabial fold as the cheek is no longer torn between the centre of two circles and can now flow fairly smoothly.



A successful implementation of the "crossover" loop, note that there is much less visible changes in edge direction yet all major features are supported. It should be noted that on this model there is limited support for crows feet – this isn't something I'd recommend modeling.

Eyes

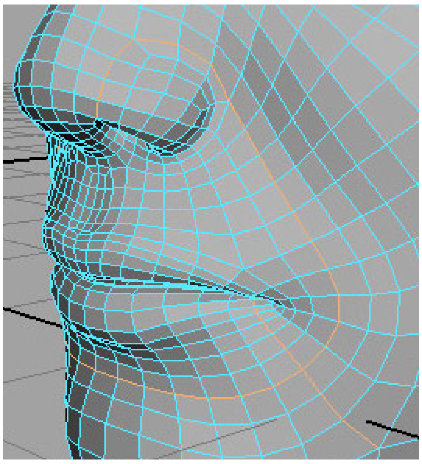
The eye sockets are topologically a fairly simple surface to construct as they are largely a series of rings, however the eyes possibly require the most amount of tweaking of any bit of the face, the hardest bit often being the upper eyelid and the weight of skin on the brow, there are no hard and fast rules for doing this but it is a lot of work to get right. It's also worth noting that you should place a lot of loops going around the upper eyelid so that when the eye is closed there is enough detail to create a believable surface. It's also worth noting that you should model the lacrimal caruncle (gunky bit in the corner of the eye) and spend a bit of time on this area getting it to look correct.



A superbly modelled eye, of particular impressiveness is the visible weight of skin above the eyelid and the cleanly implemented details at both corners of the eye

Nose

The nose can be a very hard part of the face to construct if the edges flowing into it aren't flowing correctly. One key point to consider is that the nasolabial fold most flow into the top side of the nostril, this edge should also theoretically follow round the definition of the nostril and flow down to the centre of the nose. This joining of the crease to the nostril can be seen in extreme squinting expressions. Fundamentally though the nose is one of the areas of the face that doesn't move a great deal, the only real areas that do are the nostrils and the creases at the top of the nose.



The key topology feature of the nose is the crease running up into the nostril

Mouth

The mouth is also a very delicate modelling problem, the fundamental topology of a mouth should be a set of circles flowing on and around the lips, all edges form continuous circles that expand out. This will mean that at the corners of the mouth there will be a high concentration of edges in a small area. However it is highly important that these loops stay continuous, don't be tempted to try and merge them together to save polygons, the reason being that this area of the mouth can expand and contract massively in facial animation and extreme pinching can happen very easily.
Once you are a fair distance outside of the lips it isn't as important to maintain a circular topology so it's fairly safe to start redirecting edges. In the crossover topology this means that the upper lip area's horizontal edges are defined by loops that run all the way diagonally down from the cheek. The under section of the mouth can maintain a relatively circular topology for a while. However the one complication here is that on the under corners of the mouth there is often a fairly large dimple/crease that runs diagonally out and away from the mouth. It is quite important that the edges that come out radially from the mouth follow this crease.

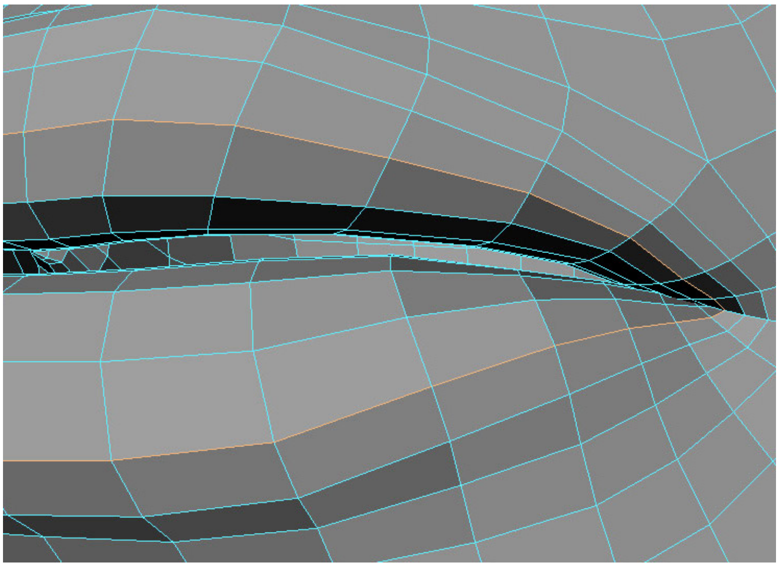
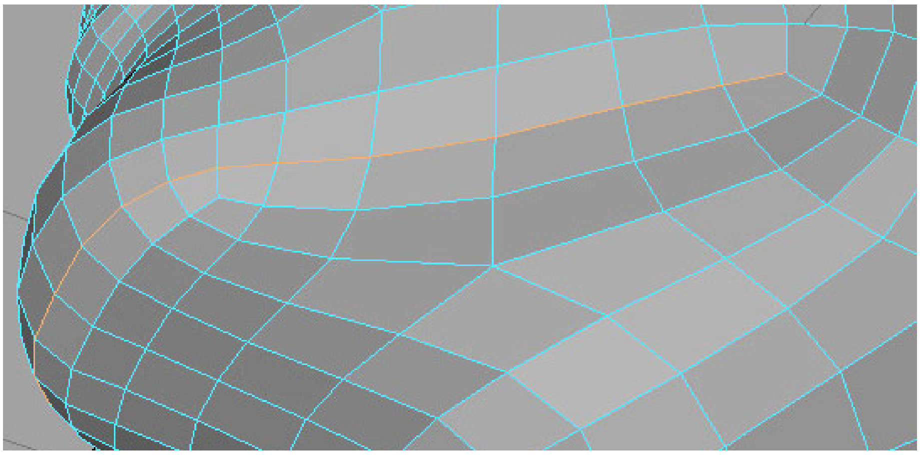


Image illustrating how loops defining the lips should form concentric rings and not be merged at the corners

Jaw

The Jaw is often a surprisingly overlooked area of facial modelling, it is generally advisable that edges follow the curve of the jawline so that a jaw can be easily hardened, it is possible to construct a relatively soft jawline with edges flowing diagonally to it but this isn't advisable.
To create edges that follow the jawline it is usually necessary to create a star as it joins the corner of the mouth. It is also necessary to do some reworking of edges in the jaw corner to get shapes to flow correctly.



Loop being used to define the edge of the jaw

Brow

The brow is a relatively easy surface to model in that the only real topology choice is to whether you include support for forehead creasing and wrinkling, by and large most of your topology here will be governed by how your eyes are constructed. On relatively young characters I'd avoid modeling forehead wrinkles and instead use bump maps if they are necessary, for older characters then hard modeling the wrinkles may be justified.

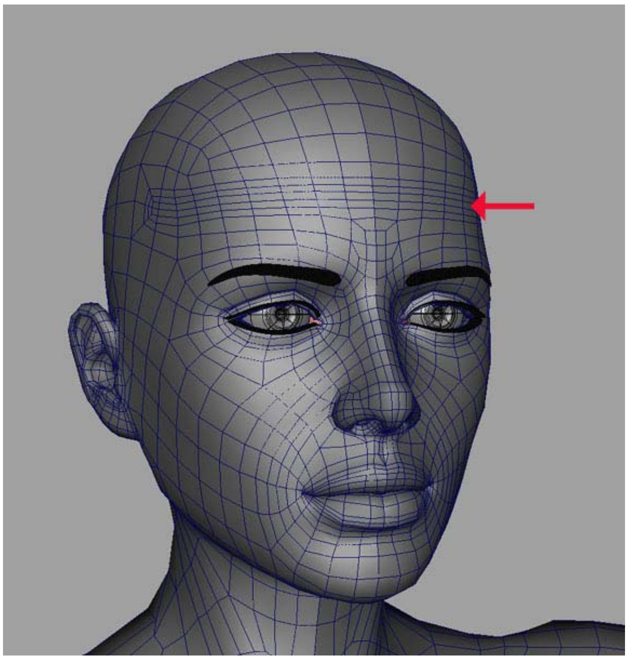
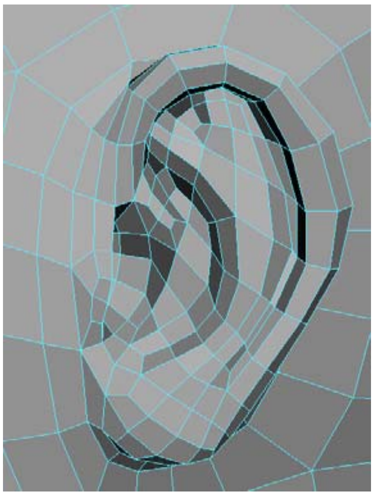


Image illustrating how supporting forehead wrinkles can disrupt your topology

Ears

Ears are one of the most complex surfaces to model on the human body and as such a lot of modellers have a really hard time recreating them. However once the basic shape has been broken down they aren't as complicated as initially expected, the other main bonus about ears is that it's extremely unlikely that they will see any substantial deformation which can relieve some of the pressure of creating "perfect" topology. Generally speaking the ear tends to sit at the centre of a rough circle of edge loops, however once these edges come "into" the ear it's often necessary for a lot of edges to be triangulated or rerouted to create more detail, this is normally particularly apparent at the front of the ear. One particular thing to be aware of is that most ear modelling tutorials recommend modelling the ear separately from the rest of the head and then combining it once it's done. I'd heavily recommend against this as it often leads to "ears on stalks" syndrome, the hardest bit of modelling an ear is properly getting it to flow into the side of the face.



One of many possible ways to approach the ear

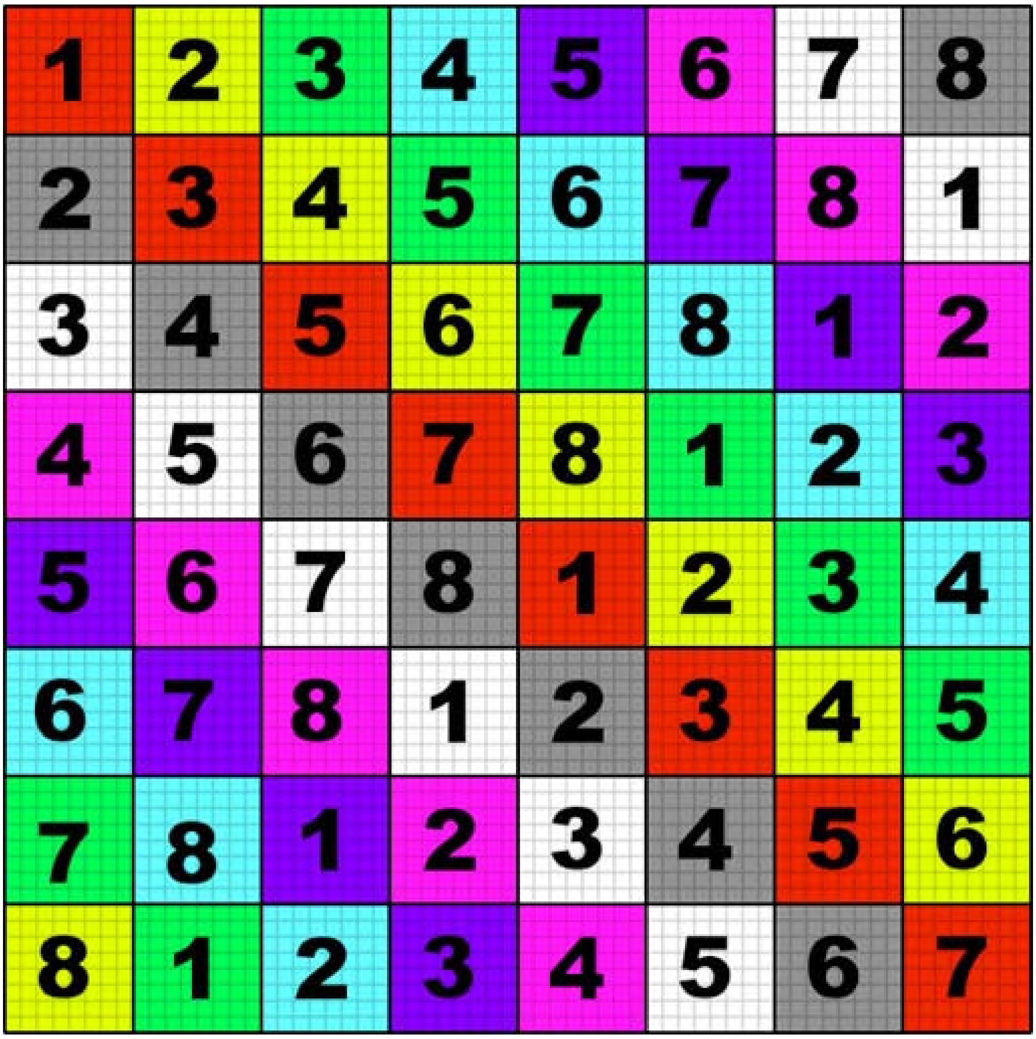
UVing

UVing is a task of vital importance which also happens to be quite tedious as well, however someone skilled at UVing will not only produce good results but will do them fast and with little mundane tweak work too. There is a lot of work that can be passed off to the computer when UVing and it's absolutely essential that you work out a good workflow for your sanity as well as the end results. The goals of a good UV unwrap are:

- Minimal amount of distortion
- UVs uniformly spaced
- No UV overlapping
- Minimum amount of wasted UV space
- Minimum number of seperate UV shells
- Logically placed seams – Hidden as much as possible

As stated the primary goal of a UV map is to make it so that a texture can be wrapped onto the geometry with the absolute bare minimum of texture distortion. The usual way to test this distortion is to apply a test image consisting of a repeating checker pattern and to see how it appears on the model. Ideally the checker should be smooth, even and undistorted plus all the squares on the checker should be approximately the same size. Theoretically you should aim to fill as much of the 0-1 UV range with used UV layout as this will make more efficient use of texture resolution. If you have a UV layout that only fills 25% of the 0-1 range then you will need a texture twice as large compared to if it had filled 100%. Usually for prerendered work this isn't too much of an issue as upping texture resolution is fairly painless. However for games work texture space comes at a large memory and performance cost and so texture usage becomes more important. For a game character it's generally expected that at least 80% of texture space is used on geometry, it's also common that an object will share and mirror textures for each side to further save texture space.
The other main issue with UVing is seam placement, a seam is a texture boundary where you've been forced to cut your UV shells into separate sections. Generally speaking a seam will cause issues when painting textures as you will be forced to take care that both sides of the seam blend together without an unsightly join line. Fixing texture seams can be done with care but it's quite a painstaking process. Ideally seams should be placed in a way that makes them hard to see and definitely not in the centre of a prominent bit of the model. In this way the texture join won't be immediately obvious as the viewer can't easily see that bit of the model.
One other side issue to consider when UVing is the usability of the end UV map, technically speaking this is meaningless but the idea is that a texture artist can intuitively work out which sections of the UV map belong to which bit of corresponding geometry. In simple terms this just means things like making sure the left foot belongs in the bottom left of the UV map and the right foot in the bottom right. Another thing to look out for on this topic is to make sure the UVs flow in a consistent direction and that if possible none of them are flipped.

An important thing to consider when UVing is that there is always going to be a massive amount of balancing and trade off between the key UVing goals. For example it's relatively easy to create a completely undistorted UV layout by UVing every face on the object separately, however the sheer number of seams created by this technique makes this texture impossible to paint in a conventional way. Another trade off which is fairly common is distortion vs. wasted UV space, it's quite common that in a games character you have a largely undistorted UV layout which you then have to deform in multiple ways to get it to fit in the UV space with least wasted space. Juggling all these different factors is something for which there are no hard and fast rules, in the end it comes down to a judgment call. Generally speaking though UVs for prerendered assets should prioritise distortion and seam placement while a games character would prioritise wasted UV space a lot higher. It's also worth noting that if your textures are going to be generated from geometry i.e. Light baked or Normal mapped then it's possible to get away with a lot more seams than if you were to try painting textures by hand.



Normally for UV testing a texture is required to be highly contrasted, consist of uniquely identifiable parts and be tileable. This is a particularly effective example. It's recommended that you turn off texture visibility within the UV texture editor as this isn't exactly easy on the eye. Image supplied by www.alias.com

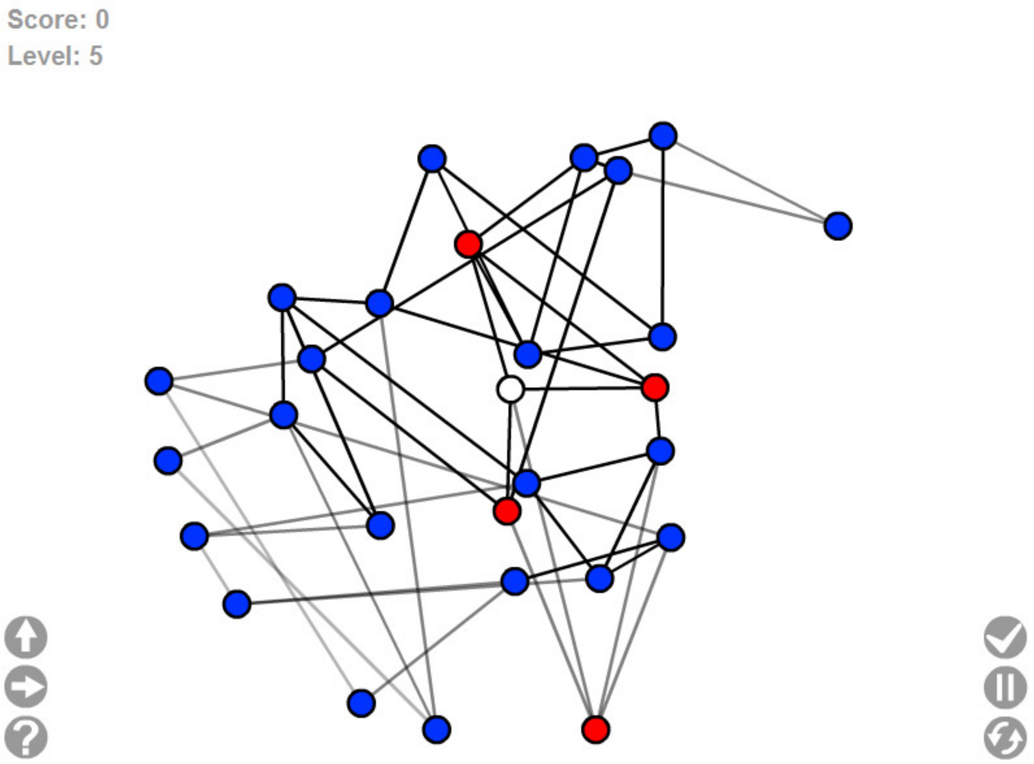
Unwrap techniques

Broadly speaking there are 6 different ways to approach UV unwrapping, these are:

Manual unwrapping
This is the process of manually moving UV points hand by hand attempting to unwrap the mesh, similar to this online game http://www.planarity.net/ . When people are very new to UVing they assume that quite a lot of this manual tweak work is necessary which is completely untrue, if you ever find yourself doing large amounts of manual unwrapping then it's probable that your workflow needs a rethink.

Pro's
- Complete control

Con's
- Dull and time consuming
- Hard to get good results
- Time taken scales pretty linearly with poly count

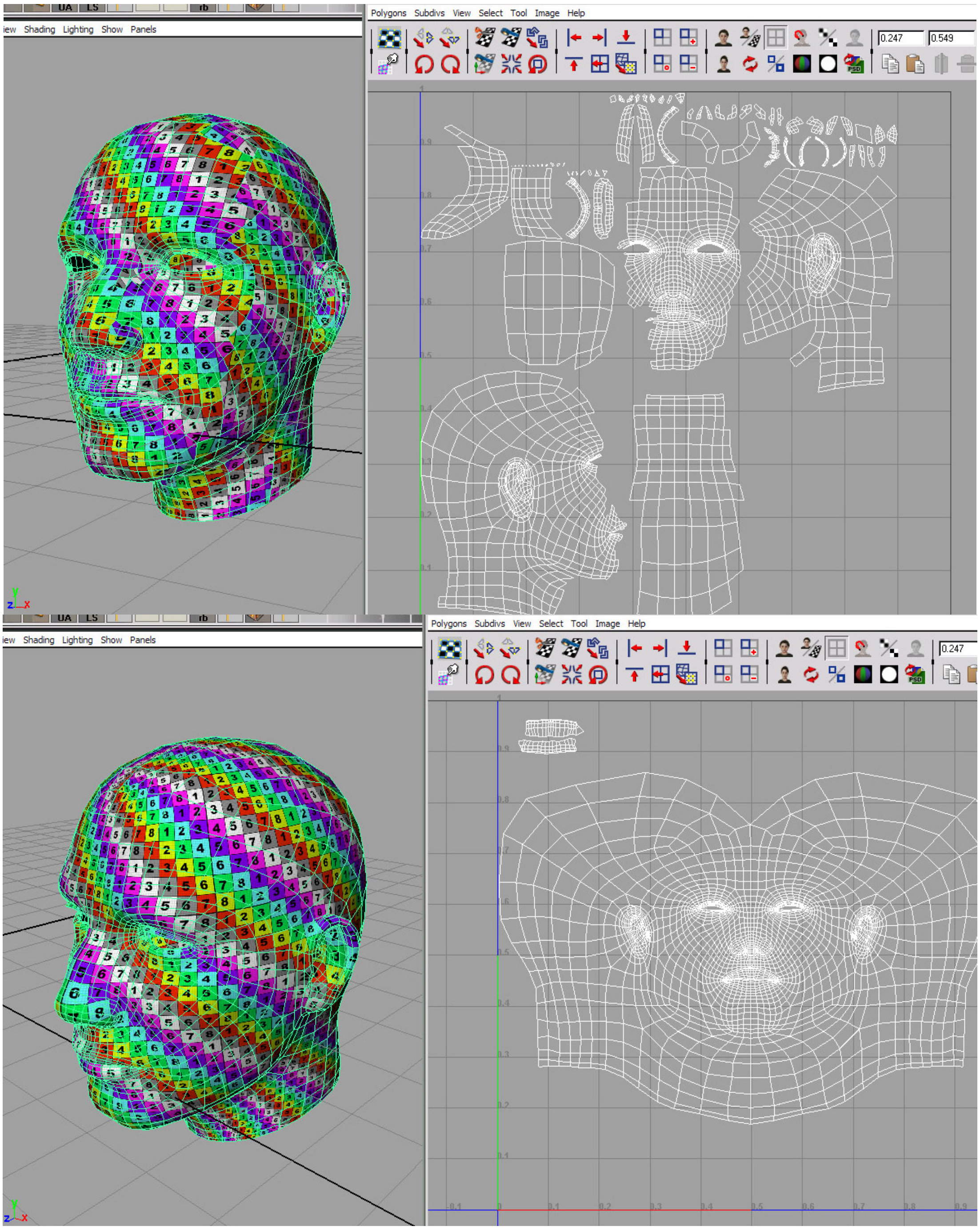


A screenshot of the planarity UV unwrap game. This game is very similar to a lot of the pain you have to go through if you unwrap an object manually point by point, you essentially start off with a series of interconnected edges that overlap and you have to pull them apart so all overlapping is removed. Like real manual UV unwrapping this requires a lot of time and effort.

Automatic unwrapping
With automatic unwrapping an automated UV layout is generated with fairly undistorted UVs but with random seam placement. It works by essentially projecting UVs for each face along it's normal to get an undistorted UV layout and then joining faces together at the software's discretion.

Pro's
- Relatively undistorted UVs
- Uniformly sized UVs
- No flipping of UVs

Con's
- Usually many seams
- Seams in the wrong place and need to be manually fixed
- Usually inefficient use of UV space



Base automatic unwrap and then after the seams have been tidied up

Planar projections

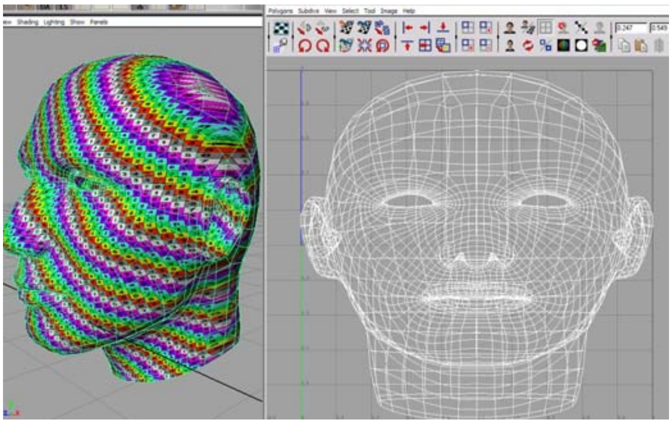
With a planar projection your UVs are projected from a fixed plane, meaning that when looked at along that viewing plane the UVs will appear perfectly aligned but when viewed from another angle the UVs may well be incorrect unless the surface itself is completely planar. When UVing a non planar object with planar projections it becomes necessary to create multiple planar maps and sew them together.

Pro's

- Fast way to get regular looking UVs
- Easy to make efficient use of UV space

Con's

- Essentially only useful on planar/non organic models
- On non trivial objects many different planar projections will need to be sewn together
- UVs are only undistorted and uniformly spaced when viewed from certain angles



Planar Projection

Camera projection

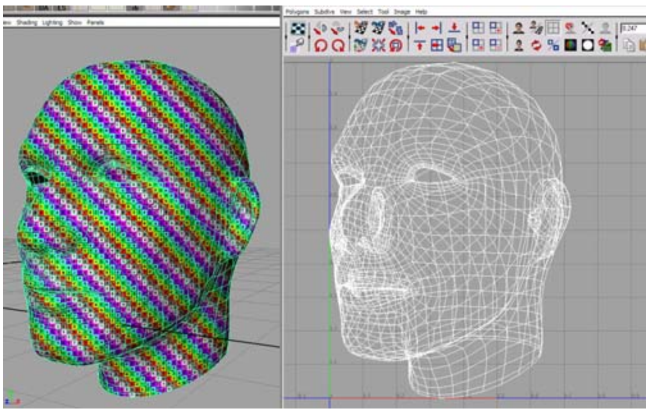
These are kind of a special case of projection's were UVs are essentially bypassed and a projection node is attached to the shader which looks up the UVs from a camera's point of view. This means that if your camera move is very simple you can just texture all objects from the point of view of a camera and project the texture on. Texture's will need to be created shot specifically.

Pro's

- Vastly speeds up the workflow if applicable

Con's

- Requires very simple camera moves for the effect not to break down
- Textures need to be made per shot
- Only really suited to environment work, not applicable to organic models

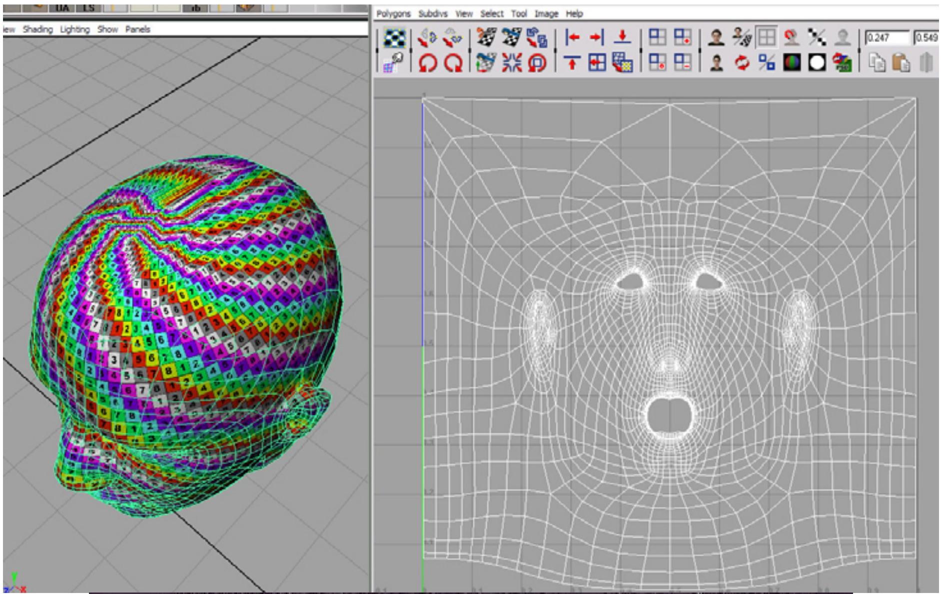


Camera Projection

Cylindrical projections
With a cylindrical projection instead of having UVs projected from a plane the UVs are created from a cylinder surrounding the object meaning that the UVs will look correct if viewed from any point of the cylinder. This is useful for cylindrical objects but for anything which has significant detail on the top or bottom it's likely that you will get significant distortion.

Pro's
- Makes good use of UV space

Con's
- Often a lot of UV overlapping
- Very hard to get non distorted UVs on non trivial objects
- Not really suited to heads



Cylindrical map with distortion at the top and bottom
Example of a predistorted game texture from half-life 2

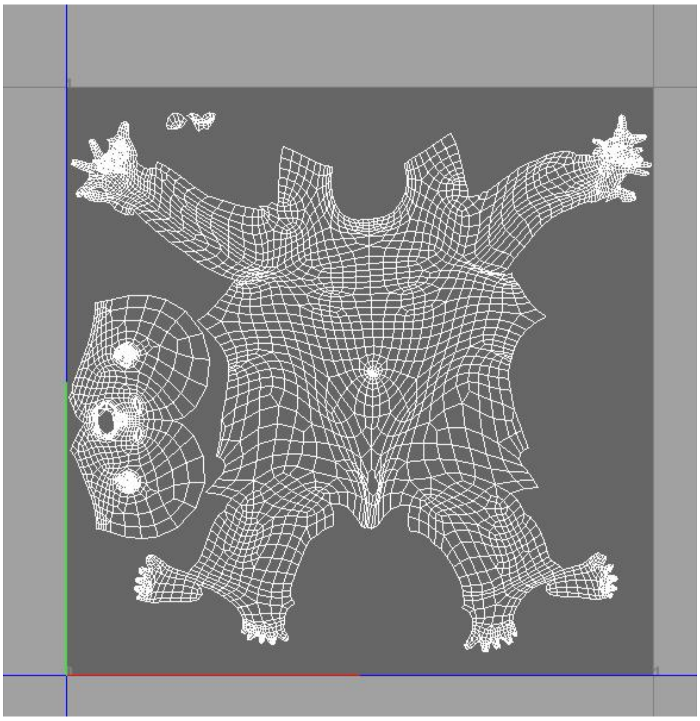
Pelt mapping
This is a relatively new technique of UV mapping where the only user input necessary is to place the seams on the object. Once this is done all the rest of the work is done by the computer in trying to work out the most optimal UV layout.

Pro's

- Generates clean and fairly distortion free layout's
- Minimal user input

Con's

- Can be slow
- Maya implementation is still fairly unpolished
- User loses a lot of control
- Generally quite a lot of wasted UV space



An example of a pelt mapped full character. Note that the entire body has been unwrapped into one piece, this is something that could not be achieved very easily using regular techniques. Also note the fairly large amount of wasted space due to the irregular unwrap shape.

For someone learning how to UV an organic model overall I'd heavily recommend they start off with a workflow that is based around automatic unwrapping and then some cleanup of the seams. This technique is a tried and tested way of UVing, the automatic component will always generate fairly undistorted UVs which only require the seams to be worked. However UV pelting is something I'd expect to see a lot of people transitioning to over time as it requires even less user input than automatic unwrapping. I'd encourage anyone new to UVing to learn automatic unwrapping first though as you will get to grips with and understand a lot more of the basic concepts and it means you have a lot more control if you need to create a UV layout with very little wasted texture space. Pelting is also largely useless for UVing mechanical/hard modeled geometry so learning more traditional unwrap techniques is an essential skill. Finally, while testing Maya's UV pelting tools the results weren't ideal and the process was still glitchy, it's definitely something worth keeping an eye on though as the tools mature. http://sunitparekh.com/pelting/
One other thing which I'd heavily recommend people get used to is using the Relax UVs tool in Maya, if you set the mode to world space and pin the UV border then in will tend to fix all mid/high frequency UV distortion as long as the UV borders are in a vaguely correct place. When using this tool effectively you actually can ignore placement of all interior UVs and just sort out the border edges and let relax UVs do the rest. I'd recommend turning on display texture borders for this, also it's worth pointing out that relax UVs isn't 100% foolproof, on heavily folded areas like the corner of the mouth it often causes UV overlapping which will have to be fixed by hand.

Conventional seam placement

When UVing a head the placement of seams isn't usually as much of an issue as when UVing an entire body, the conventional way of doing things is to cut a seam up from the back of the neck all the way round the back of the head to just above the hairline in the forehead. In this way the majority of the head can be unwrapped with minimal fuss. The only other seams necessary for a successful unwrap are ones placed around the neck and ones which terminate the lips and eyes (not necessary if there is already a border edge there). However depending on the proportions and type of head created there can be UVing issues in the following areas:

Ears
Ears are a heavily folded surface with a lot of surface area, however the area they area they occupy in UV space is often only as large as a completely planar ear, otherwise this would cause distortion to the side of the head. This causes the ears to be quite stretched and have relatively less UV space than the rest of the model. In most scenario's this is fairly acceptable as ears aren't normally prominent in hero shots and don't need a lot of texture detail. However if this does become a problem then the solution is to place a seam around the edge of the ear and place it on it's own in UV space so it can be scaled up, generally speaking though I wouldn't recommend this.

Nose
The nose also tends to have a problem with stretching due to the nostrils containing a lot of surface area but not a lot of space to unwrap it in. This causes the outer area of the nose to be fairly stretched, the normal solution to this is to either drastically scale down the texture space occupied by the nose interior (you aren't going to see it) so that the exterior can have more room. Failing that you could place a seam at the base of the nostrils and UV them separately which will free up some space for the exterior.

Eyes
The eyes should in most situations not be too much of a problem, however, there is the issue of a lot of surface area with not enough room for it to fit but its unlikely there will be as much stretching as with the ear. One issue is with the upper eyelid, it's highly likely when this is animated and the eyelids are closed there will be a lot of texture stretching, this is something you should be aware of but there isn't really an easy way to solve it.

Mouth and lips
The lips require a seam at some point as they start to flow towards the interior of the mouth, the further in you place this seam the less likely it is to be visible but the more problems you're likely to have with texture stretching, this is particularly a problem on big lipped characters. It's worth noting that when you model the interior of the mouth it's probably worth joining together this section with the interior section of the lips.
The corners of the mouth are often an issue too due to the amount of folding taking place, it can be tricky to

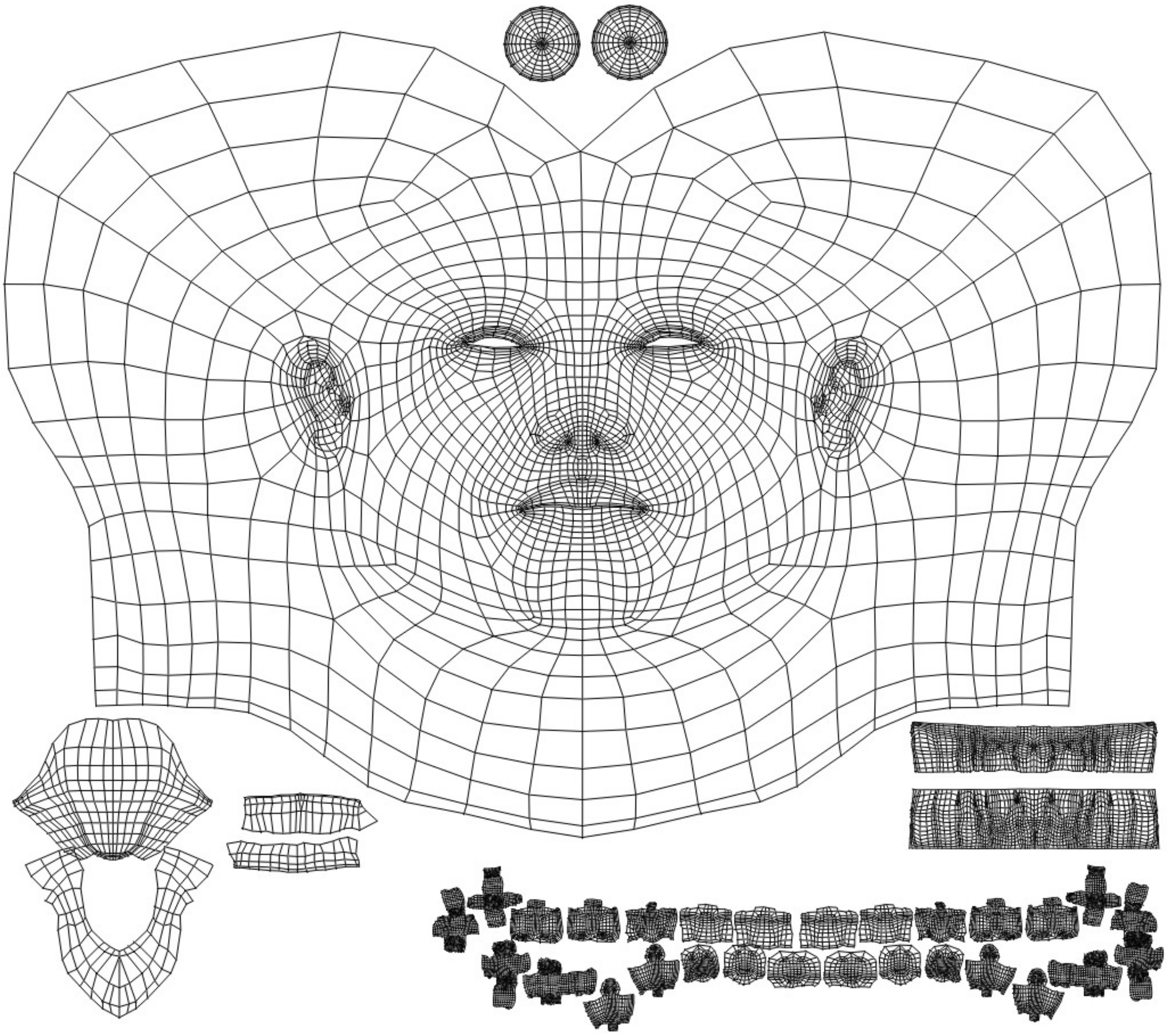
get an even UV spacing here and often automatic pelting or relaxing can fall down in this area. It's best to always check this area of the UV unwrap manually.

Hairline

The seam running down the back of the head shouldn't be too much of an issue as this is usually hidden by hair or the hairline. The only thing that can be a real problem with this area is when you are using hair or fur which often gets confused when UVs are running in different directions, unfortunately there's no real quick fix for this problem. The seam running down the back of the head and into the neck is an unavoidable problem, this area requires careful texture creation to minimise the visibility of seams.

Neck

The neck has two real problem areas, the first is the fact that the under side of the chin takes up a lot of area which cause the surrounding UVs to deform. The second is that as the neck is considerably thinner than the head it normally has to be shrunk quite a lot in UV space which can cause issues where the head and neck join. There is also the question of where to put the seam on the neckline, there is no ideal answer to this but I'd recommend either as the neck meets the collar bone (try and hide it under the collarbone) or higher up at the underside of the jaw. However both of these places can be tricky to pull off so it's quite common to see a seam running around the centre of the neck.



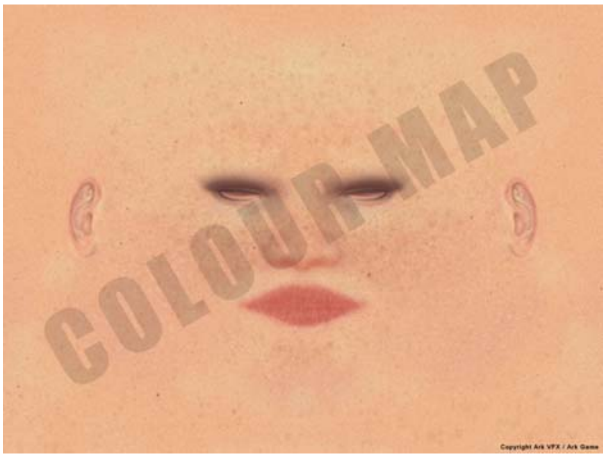
My current Unwrap approach for heads and accompanying parts

Texturing

What maps are necessary?

A common issue that people new to texturing have is what maps do I need to paint and what do they all do? At first the sheer number of mappable attributes available can be pretty daunting to a new user and also very confusing as to what is what. However just because multiple different things can be mapped doesn't mean that they need to be, I find a general rule of thumb for any realistic shader work is that the only things that really have to be mapped is colour, bump and specular. I'm not going to go in detail as to what these three do as there are already many good articles available which describe this, I would heavily recommend http://www.onona3d.com/tutorials.htm as a starting point. One thing I would also recommend is to start with you only worry about mapping colour, bump and specular, texturing and shading is one of those areas where you should learn to walk before you run. Generally speaking what you are looking for in these three textures are:

Colour – A map of the overall colour hue's of the surface WITHOUT any lighting information. Imagine you could take a photo of something in completely neutral lighting (overcast is close to this) so you would have no indication of where it is being lit from, this is how your colour map should look. If you paint lighting into a colour map then it becomes a problem if you want to light it in a different way in your scene as it can look double lit.



On the left is a colour texture with lots of baked in shading and highlight information which would be suitable for a game. On the right is a more technically correct colour map which includes little to no lighting information

Bump – A bump map is probably the hardest bit of texturing to get right and there are a lot of different ways to approach it. Fundamentally these are maps which say where the surface should be dented or raised, a black colour will correspond to a very heavy dent, a white a very heavy raise and a 50% meaning no effect.
This technique of using a black and white values to describe the amount of deformation is called a height map. Height maps can be used for bump mapping a surface but they can also be used for other stuff (e.g. displacement, terrain generation). It's important not to get confused between height maps and bump mapping, the two use essentially the same thing: a black and white image height images, just that height maps don't *just* have to be used in bump mapping. One of the main mistakes beginners make with bump maps is to just grayscale their colour map and assume that this will approximate to a bump map, in very lucky circumstances this can work to a point but bump maps are fundamentally different.
Another thing to consider is that in most circumstances the workflow for making a bump map and for a displacement map is more or less the same, if it helps for now just consider the terms height map, bump and displacement map to be interchangeable.

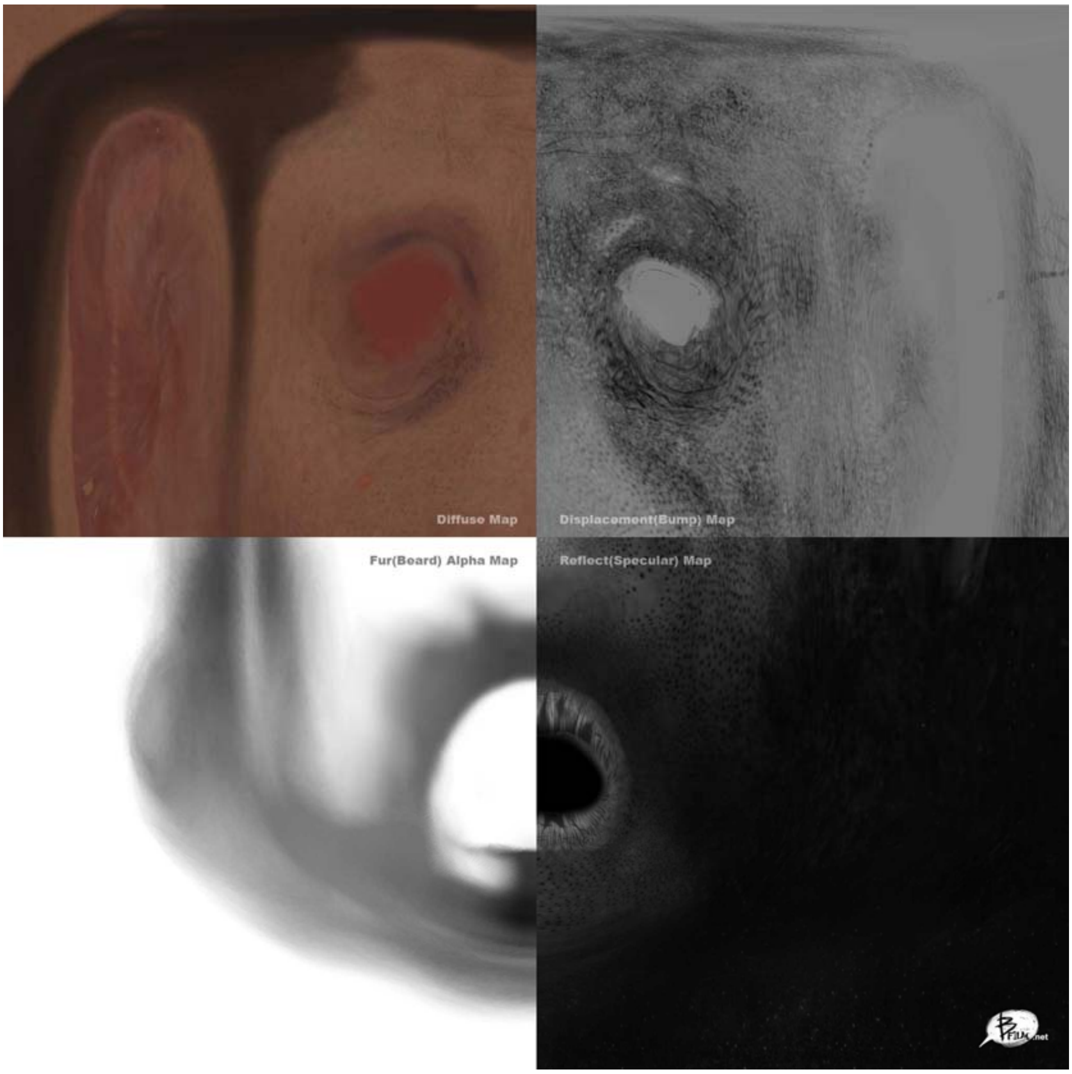
Specular – A specular map is an indication of where a surface is shiny and where it is matte, conventionally speaking specular maps are greyscale but there is no reason why specular maps cannot be coloured. In fact in skin shading it's very common to tint speculars blue to counter the reddening effect of the underlying skin. Out of the three base maps a specular map is the easiest to paint of the three but is still quite a hard one to visualise, imagining the specular properties of a surface can be very hard but it is of vital importance that there is variation in the specular map as nothing screams CG like a perfectly smooth, round blinn highlight.

Maps that can be useful:

Eccentricity/Shininess – Your specular map (technically classed as specular colour) can be thought of as the strength of your highlights, an eccentricity map could be thought of as the size of your highlights. Something like an eyeball will have relatively small, bright speculars whereas skin can be thought of as having larger, dimmer highlights. An eccentricity map is useful when the eccentricity properties vary across the surface, this could be used if the skin and the eyes shared the same shader but needed different eccentricity values.

Environment Maps – These are essentially a map of the surroundings of the object used for reflections, they can be in many different formats (cube, ball etc.). An environment map is normally used instead of conventional speculars as it allows arbitrarily shaped highlights of different colours. It is also handy for very reflective surfaces in that it provides a useful reflectance approximation to raytraced reflections and is MUCH quicker.

SSS – A sub surface scattering map is often used in most SSS shaders, this defines which areas of a surface will let through light and which are more opaque.



An example of a complete set of character textures

Maps not to paint

Diffuse – This is a confusing topic, in 3ds Max what is called a diffuse map is equivalent to Maya's colour map. In Maya there is a separate diffuse section which is effectively a multiplier of your colour value. You sometimes see people recommend making a diffuse map to define dark areas as well as using a colour map. However this is a needless thing to do as you can achieve precisely the same thing by just modifying your colour map directly, i.e. combining (multiplying) the two into one texture in photoshop.

Translucency – Maya's includes “support” for translucency, however it's important to realise that this isn't true translucency like in a SSS shader but is a very nasty hack that should be avoided in most circumstances. All translucency does is flip the normal of the surface so that you can check the lighting on the other side of the surface, this can be a sufficient approximation to translucency in some situations but is generally best left alone.

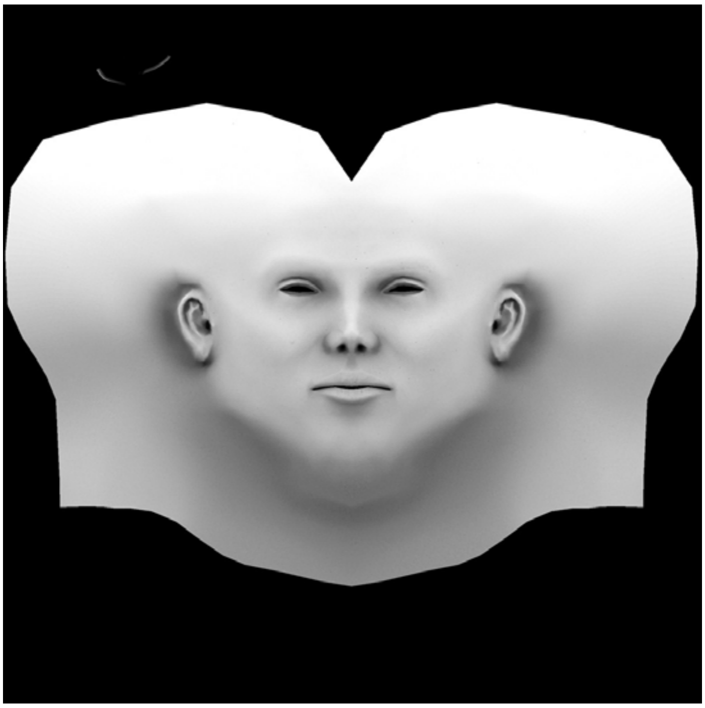
Things to Bake

Before starting to paint textures there are three different textures I would bake out to help your texture creation.

UV snapshot – This is a capture of your UV layout in the UV texture editor, it is useful for reminding you which bits of the texture correspond to which bits of the mesh and also where your seams and borders are. If you are using UV sets or are using UVs outside the 0-1 UV space then it's important you change your UV snapshot settings accordingly.

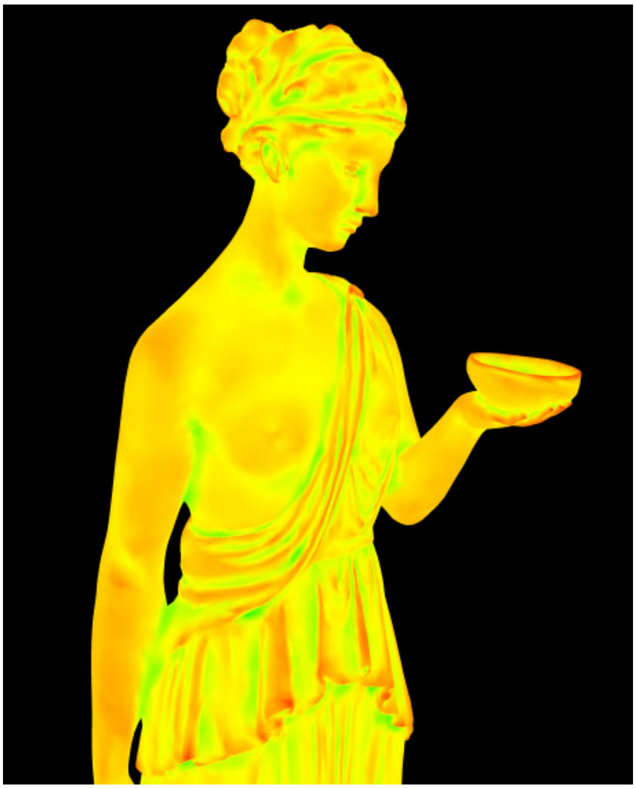
Ambient Occlusion bake – Ambient Occlusion can be thought of as indirect lighting, an ambient occlusion bake is a visualisation of ambient occlusion in texture space, a surface will appear dark in it's recesses and white in more open areas. This texture is not only very useful for recognising areas of your model in UV space but can also be helpful in itself for colouring/multiplying your textures.
When baking ambient occlusion there are a few things to consider:

- There is a bug in Maya's ambient occlusion baking in that it treats all normals as if they were hard, the only way to get round this is to smooth the surface even higher until each face becomes small enough for these edges to be unnoticed. The other solution is to just blur your bake in post.
- It often helps to place a large polygon plane a fair distance below your object so that you only capture indirect lighting from the sky as opposed to lighting from all directions.
- You will normally need to up the spread and the samples to get a usable baked result.

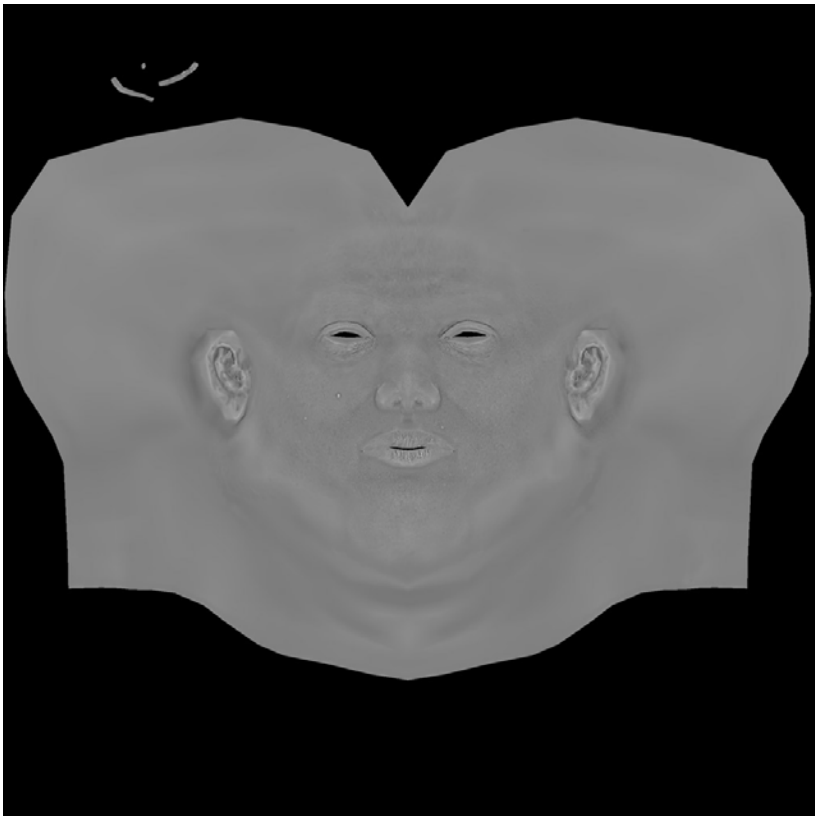


A baked occlusion Map

Curvature Bake – http://www.lightengine3d.com/index2.html, this is a very helpful plugin which analyses the curvature of a surface and produces very fast renders which visualise this result. The plugin comes with it's own baking script, when the colours are set to greyscale the resulting baked map is a very useful analysis of the curvature of the surface. It's worth noting that this is largely more useful when dealing with high resolution baking of details rather than a more simple model.



Rendering of the curvature shader at default settings, green corresponds to recesses, red to exposed areas.



Curvature bake of a head with colours remapped from black to white

Texture painting tips

I would recommend that the first map you try and paint for an object is always a bump/displacement map, once this has been finalised you can use a lot of the information on this in your colour and specular maps.

Bump

Painting a good, accurate bump map is hard, the main stumbling block being visualising what brightness corresponds to what height on a surface, as such there are a number of ways to paint a bump map:

Traditional painting in photoshop

The normal way of painting bump maps is just to paint them by hand in photoshop, photos of materials are often used and grayscaled for reference. This texture is then periodically loaded into Maya to check how the bump map is working and what heights need to be changed.

Pro's
- Conventional and comfortable way of working
- Relatively easy to get quick bump's out
- You can easily combine existing images/textures together

Con's
- Very hard to visualise what heights/brightness are necessary
- Range of bump height is often very low, working on a 16bit image doesn't greatly help as you are still limited by the texture painters ability to visualize 3d geometry depth as arbitrary brightness values.
- Small surface details are often overbumped and large areas underbumped due to problems in visualising depth
- Requires a lot of checking in a 3d program to see how well things are working
- Difficult to source existing accurate height maps to speed up texture creation

Generated from high resolution geometry

A common technique for creating bump, displacement or normal maps is to generate a high resolution version of the object and transfer the detail from the high resolution to the lower one. The workflow for doing this is essentially the same whether the intention is to make a bump map, normal map or displacement map (see Maya 7's surface sampler). This high resolution mesh can be made in any way and as the topology is unimportant it is quite common to disregard efficiency and polygon count and just subdivide the mesh multiple times and create the detail with 'sculpting'. When doing this sculpting Maya tends to have a poly limit of around 100,000 before it gets too unresponsive to be viable. For this reason a tool like Zbrush is very highly used as it can handle 2-10 million poly meshes and also has a more extensive set of dedicated sculpting tools.

Pro's
- All detail is generated from existing geometry meaning that all height values are accurate
- A proficient sculpter can generate a high res mesh very quickly
- Very easy to generate high range bump/displacement maps (16/32 bits)

Con's
- Polygon limits on high res mesh (depending on platform)
- Very high frequency surface detail is quite impractical to do with this technique as it requires huge poly counts.

Painted in 3d

This is a broad technique where a bump map is hand painted using a 3d paint tool within a 3d program. This texture can periodically be plugged into the shader and tested within the program. It is also possible to paint a texture directly as a bump map and see the results in real time as you paint without needing to swap out shaders.

Pro's

- No need to swap between programs to test bump maps
- Fine detail is very easy to achieve as it is no longer limited by polygon count, just texture resolution
- Painting directly in 3d removes the abstraction of painting a texture in 2d UV space.

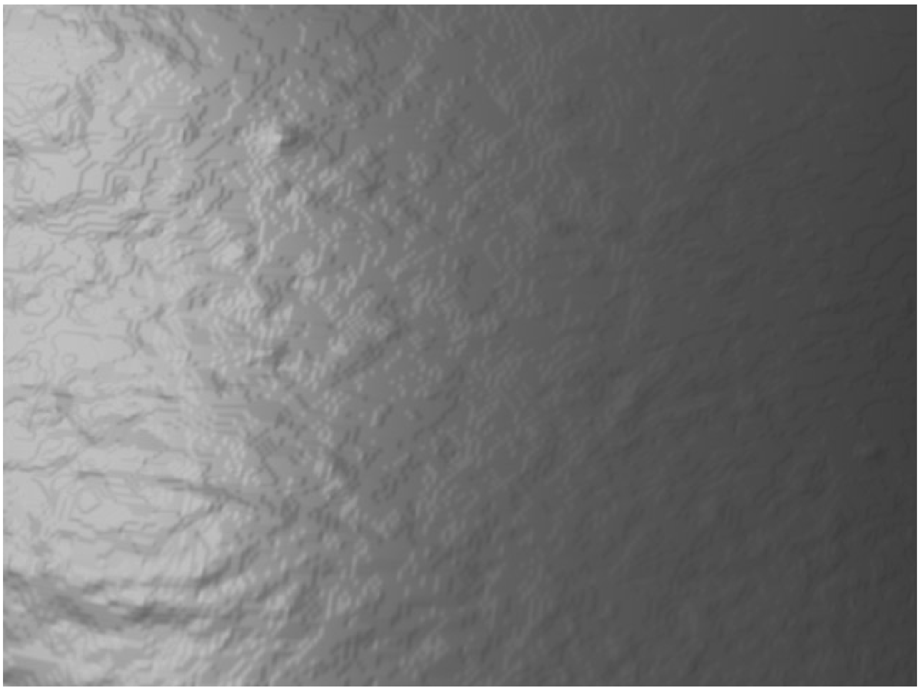
Con's

- Maya's 3d paint tool is quite slow at high texture resolution
- This process is very slow in Maya if the bump map is directly plugged into a shader and high quality render mode is turned on as the texture needs to be converted and sent to the graphics card on every stroke. It's worth noting that Zbrush has a very good implementation of this.

Overall I would currently heavily recommend generating bump maps from real high res geometry for large to mid scale bumps, this is simply due to the accuracy with which they can be made and the relative speed that you can add detail. When in comes down to very fine surface detail it gets a bit more tricky, at this point it becomes impossible to generate it from geometry from within Maya, if you have access to Zbrush then I would advise using this as it is an excellent tool, if not then I would recommend photoshop to do the fine detail painting. Using 3d paint tools to paint bump maps directly could in the future be a very good workflow but at the moment the Maya implementation is too slow to be usable.

8 vs. 16bit textures

One thing that is worth touching upon is the use of higher accuracy textures when using bump or displacement. The problem is caused by the fact that a height map is solely reliant on the grayscale brightness of a texture, a standard 8 bit image may have 16 million colours but it only has 256 levels of brightness. This means that on a height map that has extreme high or low values the mid areas may have insufficient accuracy and banding may be visible in these areas. Note that this is only usually an issue with textures generated from geometry, it's unlikely a hand painted texture will make use of this much accuracy. Fortunately it's very easy to use a higher accuracy texture format as both Maya and Zbrush have the option to export height maps as 16/32 bit file formats.

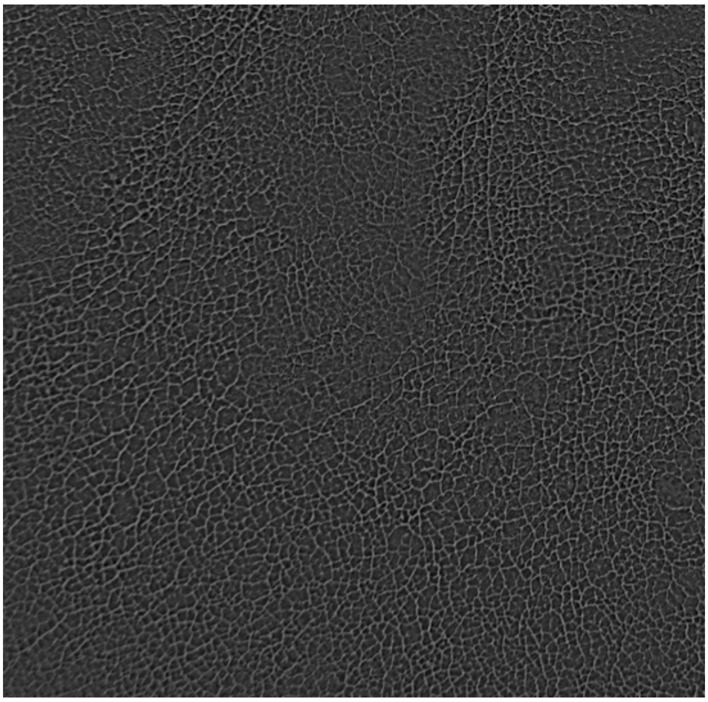


Banding problems caused by insufficient depth

Getting good height map reference

This is a real sticking point, getting good reference material for skin for use as a bump map. My first piece of advice would be try and get some genuinely high resolution photos of head closeups and to try and analyze the fine detail bumps and imperfections of the skin structure, a good site for this is 3d.sk (not free though). However, actually using these reference images directly is more or less impossible, simply greyscaling them doesn't get any kind of meaningful result. The real problem is that surface imperfections only really show up on specular highlights, however these are only ever over localised areas of the skin and they are brighter than the base colour making them hard to use directly. I would have thought it would be possible to rub dirt deeply into skin and then try and wipe off surface dirt leaving only recesses, if you took photograph reference of this it may be possible to use this as actual height map information, however this is yet to be tested. Another area to look at which may be helpful when dealing with human skin is to try and find reference of extreme skin wrinkling on elephants, rhinos, monkeys and possibly just anything made of leather. Because the skin is so much more weathered and wrinkled it is usually easier to use it in texture painting, however this is only really useful on weathered or old characters and not on people who need to have delicate looking skin.

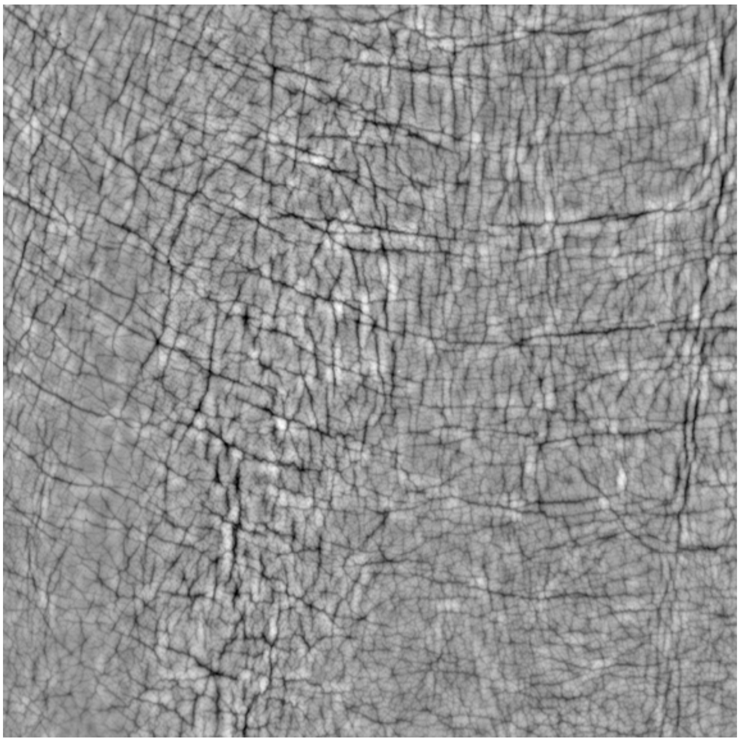
Skin reference from 3d.sk – Note how surface detail is only really noticeable on specular highlights



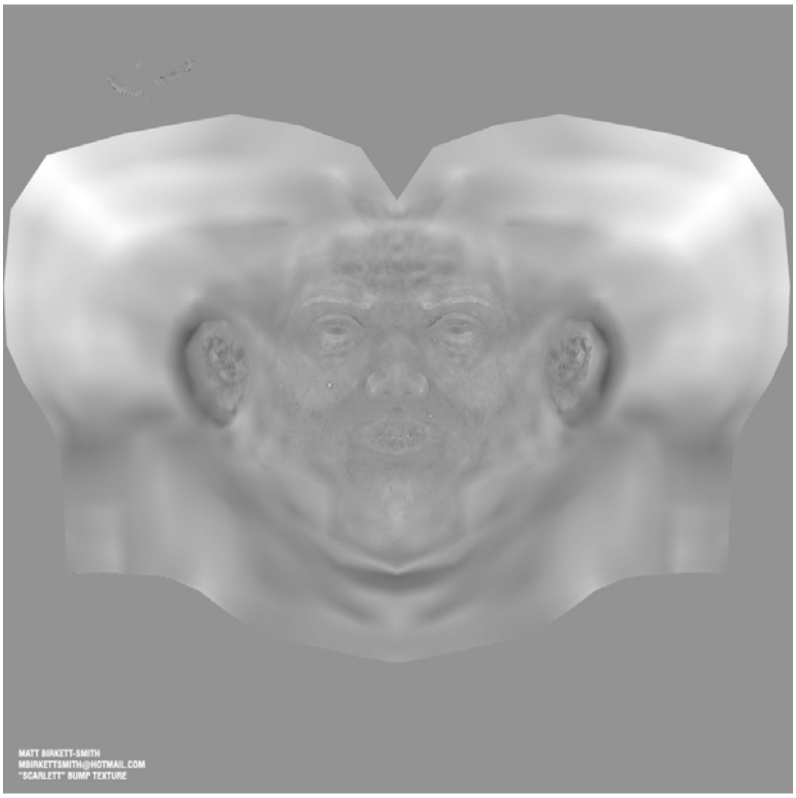
Adjusted picture of elephant skin – makes very good skin cell reference

I would instead recommend that you use photographs simply as reference pictures and try and hand paint/sculpt height maps based on what you can make out from your reference. Try to analyze which areas of the face have which properties, where you get spots, where you find pores and where you find wrinkles. It's of vital importance that your bump map isn't completely even all around the surface. The skin surface underneath the eyes for example is vastly different to that on the nose.

Another thing to consider is that the technology does exist to do very fine detail 3d scans of a surface and to record this surface as a height map, if you are lucky you may find a company giving out free samples of skin surfaces which can be incredibly useful http://www.xyzrgb.com/html/scanning.html



Sample of a height map of rhino skin available from www.xyzrgb.com

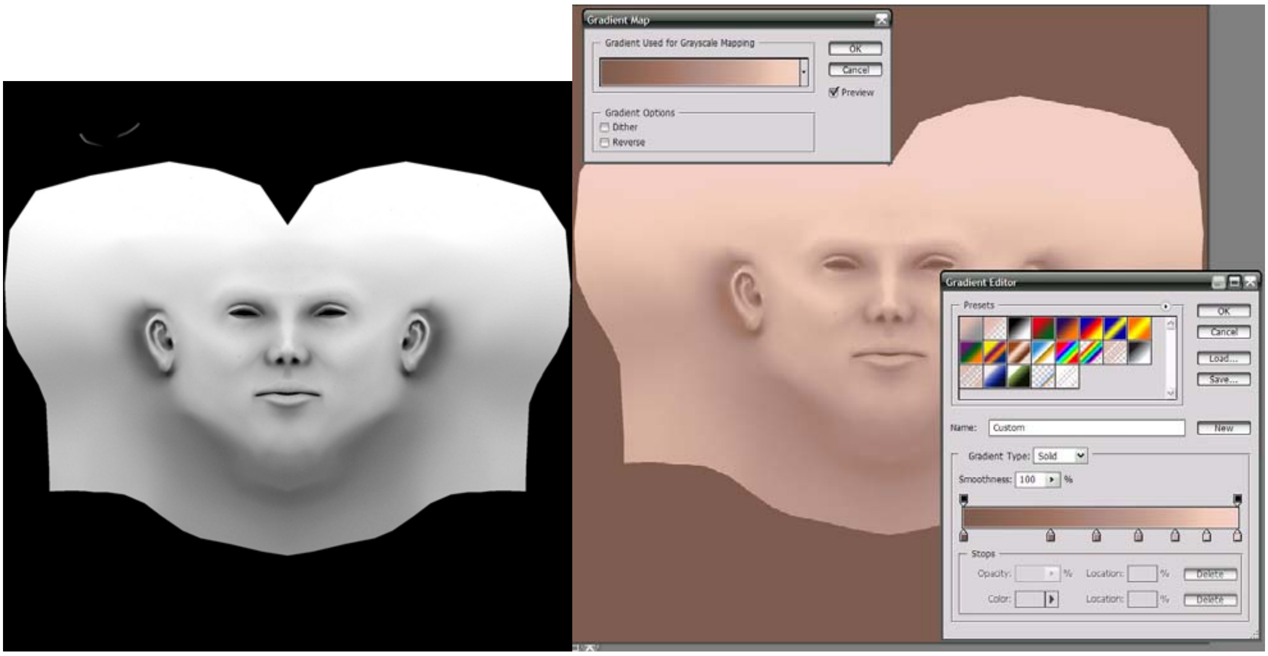


Bump map of my character – generated from high res geometry

Colour

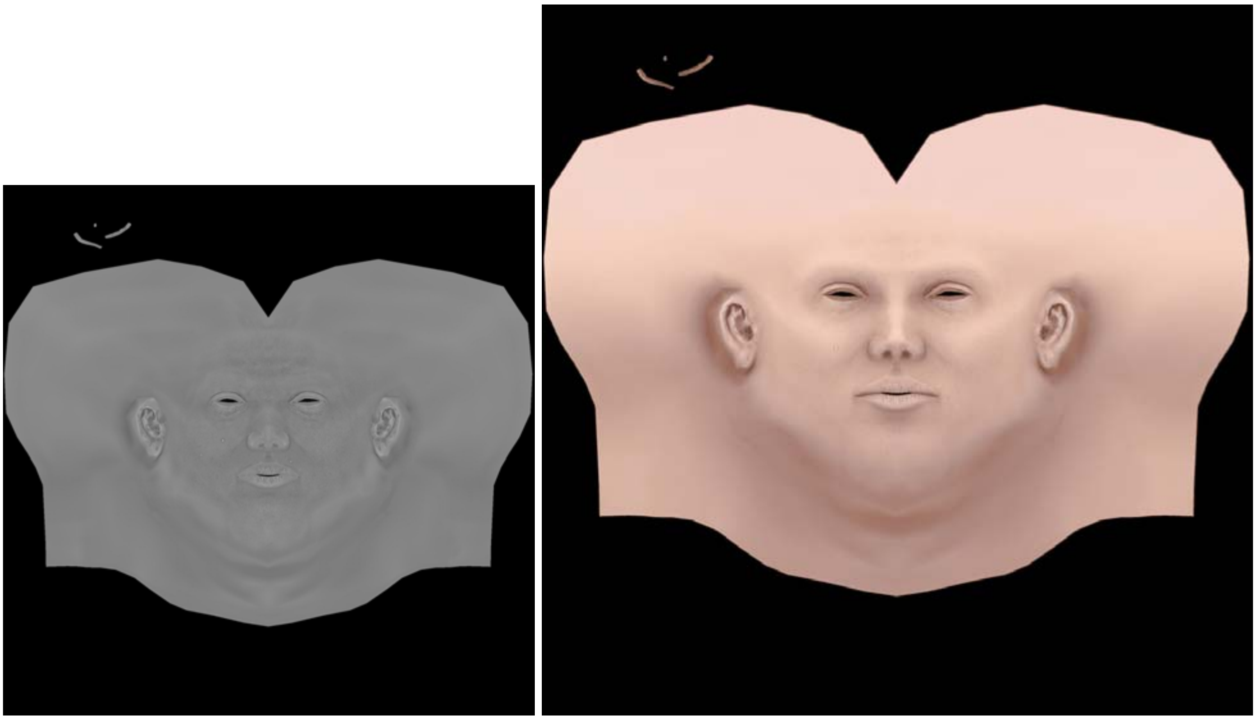
Painting a colour map is a fairly involved process but doesn't require as much abstraction as doing a bump map, it is also made much easier if you already have a highly detailed bump map from which you can take and colourise surface details. Fortunately photo reference is much more readily usable in a colour map than it is in a bump map, as long as the lighting in the photo is directionless then you should be able to use quite large amounts of the photo. Despite this I would be wary of using photographic reference too heavily as you will always run into problems with prelighting and renders being double lit. I would instead recommend using photographs just as detailed surface reference and to see what kind of colouring is found on area's in the face and then try repainting these details by hand. However if you need very quick results then I would lean more towards heavy use of photographs, just make sure they are lit appropriately.

Personally speaking I have a workflow which allows for very fast creation of a basic colour map that can then be refined on. It's at this point I contradict myself with regards to baking in lighting as the first step of this technique is to start with an ambient occlusion bake which is then gradient remapped to get an approximation of skin tones.



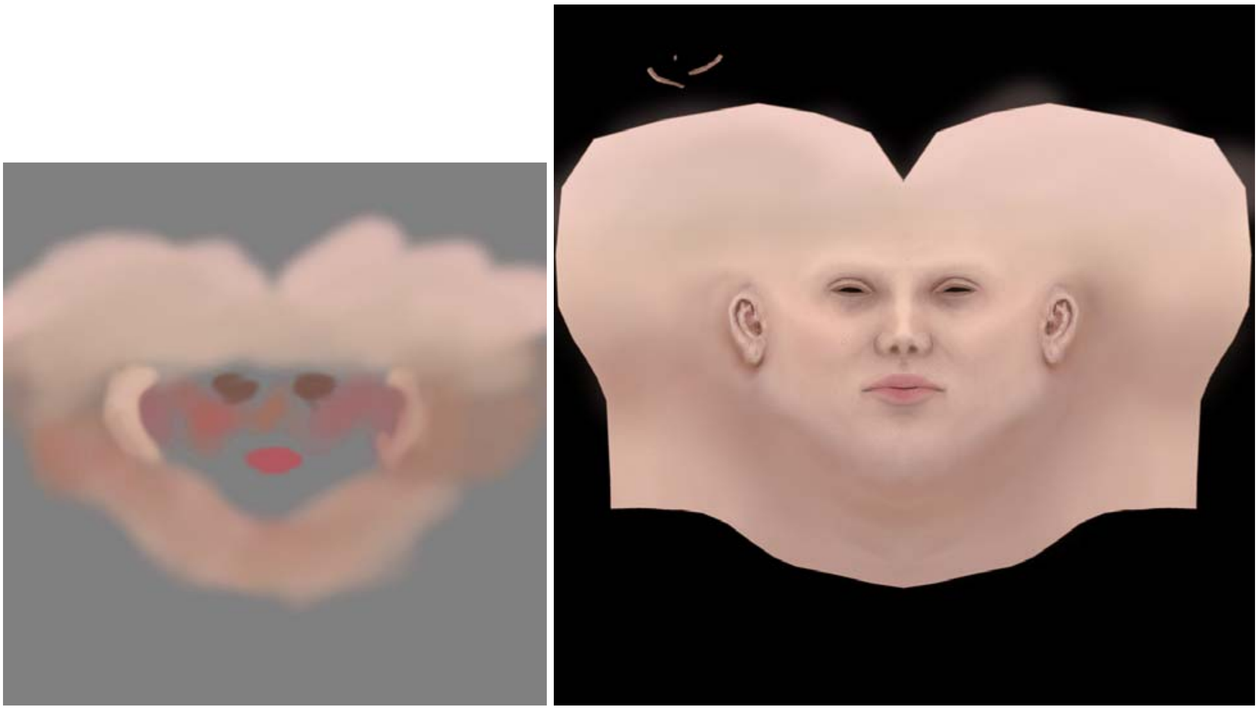
Occlusion bake which is then reramped

My justification for use of what is effectively baked lighting is that the ambient occlusion pass is simulating the effect of directionless light from the environment and the sky, in 99% of situations this will be acceptable as most light setups involve being lit from above. It's only in very specialised low level light setups that you may find a problem with directionless light being baked to the texture. In an ideal world none of this colour tinting would be painted direct to the texture and would instead be applied through the shader, however I have found it to be generally a pretty safe and fast corner to cut.
On top of this remapped ambient occlusion layer I then take a curvature bake of my high res mesh and apply it as an overlay layer, this has the effect of defining the small surface details of my colour map and provides a great base to work from.



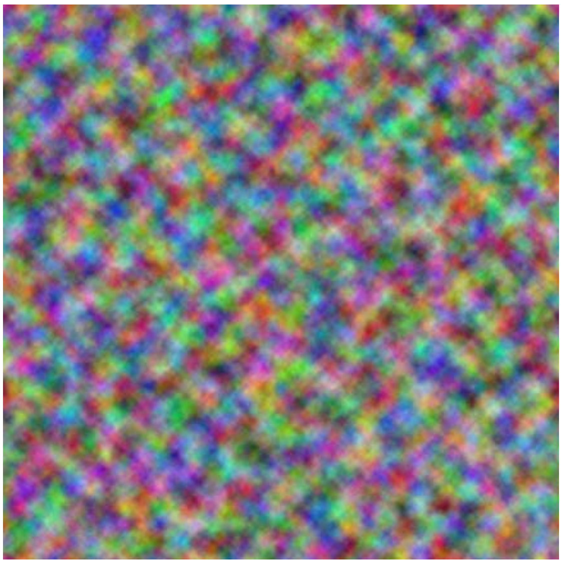
Tinted occlusion bake plus curvature bake overlay

On top of this I then go about manually painting and recolouring areas of the face, taking care to ensure things like the eye sockets are darker and have a hint of purple and the cheeks have a hint of red. It is also necessary to paint out a lot of the obvious lighting that comes from the occlusion render and some of the glitching that comes from the curvature bake.



Recolouring layer

The next step which I find helps is to apply subtle hue variation to the surface, this trick is from http://67.15.36.49/team/Tutorials/benmathis/benmathis_textures_2.asp essentially you make a new layer and in each channel you create a black and white clouds fractal, the combination of these creates a layer which contains pretty much random colour variation. If you then apply this layer as a hue blending operation and set the opacity down to 5-10% then it has the effect of applying nice subtle hue variations to your colour texture.



Hue randomization layer

With these steps you should now have a good base to work from with your colour texture, from here it's just a matter of colour refinement and adding in more surface details.

Sample finished colour Texture

Specular

Thankfully a specular map is much faster to paint than a bump or colour map. Generally speaking I find all that is necessary is to take a curvature bake as a basis and then just make a new layer and paint in rough bright and dark patches on a low opacity setting. Make sure areas like the lips and the lower eyelid are bright and recessed areas are fairly dark. Try to imagine how “greasy” your character is and try and workout which areas will see high reflectance and which will be fairly matte.

Sample specular texture – essentially just an adjusted curvature bake. Extra blue is added to this map with the shader network, this is due to the fact that with a white specular will feel unnatural when layed over a red under surface, when tinted blue the specular will contrast more with the skin and produce a more believable end result.

A note on normal Maps

On speaking to a few people I notice there is a lot of confusion with regards to the difference between bump maps, height maps, displacement maps and normal maps, people are right to be confused, there are many terms for what is essentially the same thing.

The first term to cover is height map, this is a catch all term for a black and white image used to represent the height of a surface. This can be used to render as a bump map, as a displacement map or to physically move single vertices and create terrain. It probably helps to think of a bump map as 'a height map which is plugged into the bump channel' and a displacement map as 'a height map which is plugged into the displacement channel'. They both use identical looking texture maps.

Displacement mapping is where the surface is physically altered based on the height map, bump mapping is a trick where the lighting is modified but the actual physical shape and outline is unchanged. So what is the difference between a bump map and a normal map?
When using bump mapping the renderer/engine takes the height/bump map and then recalculates these surface normals used for lighting calculations based on how much it needs to be bumped. They make it seem to the light that the surface is flowing in a different way as to how the geometry alone is describing, it's important to remember that with bump maps the surface normals have to be recalculates and that the surface can only be modified along its normal.

With normal maps you can directly modify the surface normals of a surface without it having to be recalculated based on a height map. A normal map contains a channel for X, a channel for Y and a channel for Z. Based on these 3 bits of information the normal map describes a normal vector, this vector is then directly added/subtracted to the surface normal defined by the geometry. Not only does this miss out one stage of calculation but it also means that "bumps" don't have to be just along the normals of the surface, as well as going "in" and "out" the surface can be modified to go from side to side too. Games frequently use normal maps as their calculation is less abstracted than bump maps and produce better results, however newer and complicated shader effects like parallax mapping/virtual displacement mapping require a height map and normal map.

Now, normal maps are often used for transferring surface detail from a high poly mesh to a low one, however this process doesn't have to be exclusively associated with normal maps, it can also be done with a bump map too. The only thing is it's generally mathematically easier to work out the "difference" in world space between two meshes as an arbitrary vector (normal maps) rather than a "displacement" amount which has to be relative to the surface normal of the geometry (bump maps). It is techincally possible to directly hand paint normal maps in photoshop but it's very hard to visualise as you are working with a 3d co-ordinate system. Its much easier to handpaint bump maps which are just a black and white image. For this you just have to remember that 50% grey is no effect and start painting darker and lighter areas for where you want your surface to rise and fall. For this reason normal maps are more or less solely created from high resolution geometry. It should be worth noting that you can convert a bump to a normal map using this photoshop plugin:http://developer.nvidia.com/object/photoshop_dds_plugins.html

In prerendered CG work there is generally no need to work with normal maps as bump maps are easier to create and work with, are more flexible and can be used as displacement maps if the shot requires it. Normal maps also lack default rendering support in Maya and require quite a complicated network to get rendering correctly. Therefore I would only advise using normal maps if you are doing realtime/game work.

Procedural texturing

One thing I should mention when texturing is procedural texturing, this is in my opinion a very interesting field of research but as yet not a particularly valid solution when doing photoreal character work. I have yet to see any convincing procedurally based skin textures although I admit it is theoretically possible to break skin down into component parts that are procedurally replicable.

Skin Shading

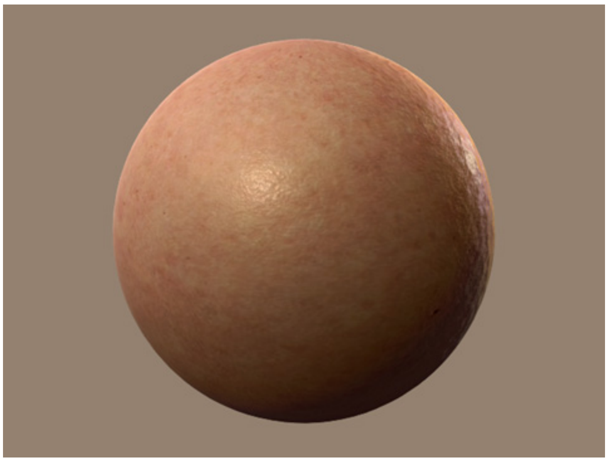
Physically correct vs. heuristic

One of the largest hurdles encountered when creating a photoreal character is the shading of human skin, this is an immensely large and heavily researched topic which is the subject of many academic research papers. There are many proposed solutions to rendering skin but they can broadly be put into two categories, either physically correct simulation of the properties of skin or heuristic workarounds that try to mimic the effects of rendering skin. To look in detail at the former is beyond the scope of this document, I will instead look at some of the more practical techniques for getting convincing looking skin renderings. It's important to bear in mind when looking at this topic that given it's complexity a lot of corner cutting is necessary, it should not be considered whether something is being done correctly, instead it should be considered whether the end result look believable.

There is one disclaimer I'll put here, regardless of what you do with your shader the most important thing for realistic skin is good textures, if your texture work is poor or even worse missing completely then it's going to be virtually impossible to create good looking skin or even see if you are on the right path. Skin is all about imperfections, without texture's it just doesn't work.

Properties of skin

The main reason that skin is such a complicated problem is that it is a substance made out of layers, each of these with differing properties. Most of the upper layers are heavily translucent and allow light to reach down into the denser lower layers, the light that passes through these lower layers goes through many blood vessels and picks up a lot of red light. Different areas of skin have different physical make ups but generally the thinner the skin the more subsurface scattering can be observed, also it's worth bearing in mind “blocking” objects like bone as if bone is very close to the surface then the scattering effect will be greatly lessened.
Skin also contains varying amounts of oil on the top layer which can have a great effect on the skin's reaction with reflected light, on area's with a large build up of oil a lot of specularity can be observed. Skin also has the property of picking up a very large amount of reflected light at glancing angles, this is compounded when the skin also has many small hairs which further the amount of light reflected.



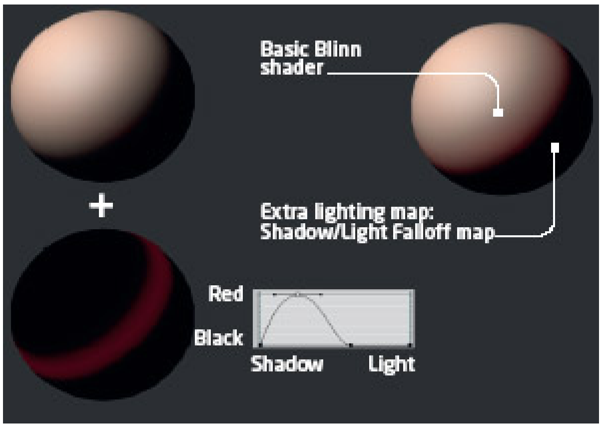
A “skin ball” from the DT3D skin shader

Layered approach

Generally speaking the traditional way of tackling the problem of skin is to split up all it's components into different shaders and then bring them all together in a layered shader network. An excellent application of this is known as the “Stahlberg” technique and it works by splitting using 4 separate networks in a layered shader. The layers are a base level lambert with the colour map, a translucency fake layer which tints the lighting, a glancing specular layer to catch heavy specular at glancing angles and a second specular layer for dealing with speculars at normal angles. http://www.androidblues.com/shadetut.html

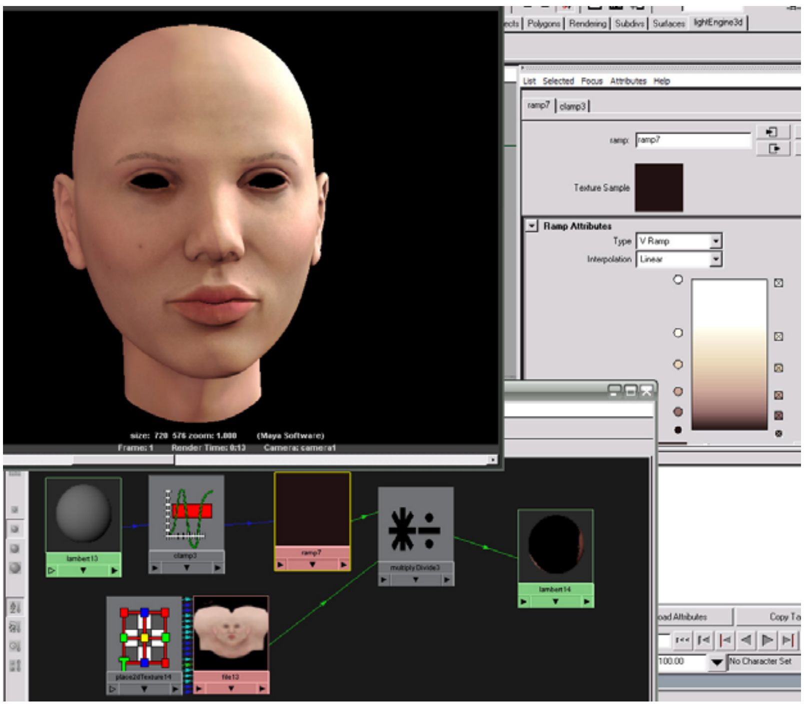
Tinting / Brightness remapping

Broadly speaking a lot of the fast and dirty approximations to skin shading fall under the category of tinting. Essentially the object is rendered with the traditional lambert shading model but instead of a fairly linear black to white output the colour is tinted depending on brightness. In the Stahlberg model a lambert shader is used to drive a colour ramp where all values below a certain threshold have a red tint added instead of being pure black, this gives the impression of scattering in low light areas as skin is shaded a deep red.



Adding red to the shaded areas – A relatively subtle tweak

In the Stahlberg model only a relatively small amount of colour tinting is used, however it is quite possible to take this approach much further and completely remap the colours of the lambert model. However it should be noted that when using this approach all tinting is effectively being added to the base colour, meaning that it bypasses anything used in the colour map, this can create a glowing look to the render (on darker areas such as the eyebrows) as colour is added uniformly to all dark areas.



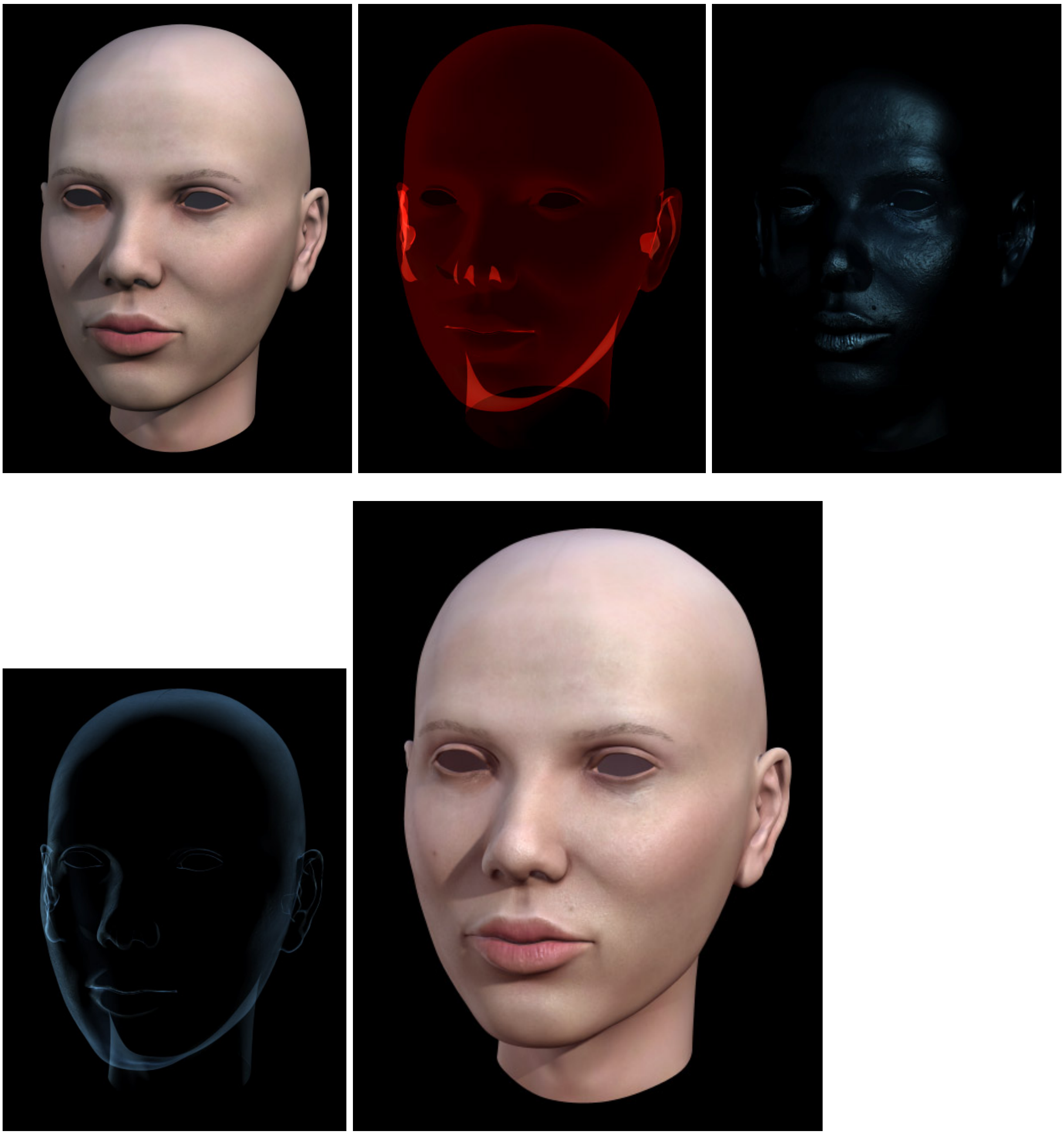
Example of a completely colour remapped diffuse curve

If the brightness curve is to be remapped very heavily then it becomes essential to change how the colour map interacts with it to avoid this uniform adding of colour. The best way to do this is to remove the base level lambert shader and instead do all of the lighting within the remapped colour shader, this can then be used to multiply the end result with the colour map being used. With this technique the user gets a lot of tweakable control for the look of the skin but there are drawbacks, one is that unless you extend the shader much further it won't have any support for coloured lighting, the other is that your overall look will be very dependent on the light levels in your scene, a slight change in lighting can have a drastic effect.

Specularity/Reflectance component

As stated when dealing with speculars it is generally best to split them into a glancing and face on component, in this way you can tweak them independently without being in danger of breaking one when improving the other. The easiest way to do this is to use the facing ratio of the surface which outputs 1 when the normals are pointing directly at the camera and 0 when they are pointing perpendicular to it, using this allows you to blend out regular speculars as they reach a glancing angle and blend in the glancing specular shader instead.

The glancing specular needs to create very wide highlights of reasonably low strength which can be easily achieved with eccentricity/shininess controls. The face on specular requires relatively small, sharp speculars which need to be used with great subtlety as they have a great effect on the perceived oiliness of the face. It should be noted that both of these two specular components should be tinted a pale blue to counter the redness of the underlying skin and should also both have a bump map applied to them to break up the shape of the highlights.



Breakdown of all the passes in the layered shader approach, in order they are diffuse+colour, translucency fake, facing specular, glancing specular and end product. Ignore the backface glitching in these picture, it isn't visible in the end product.

Bump and Displacement

When rendering skin that is based on a lambert shading model you need to be very careful with your application of a bump map, relatively small bump settings can have a disastrous effect on the believability of the skin. This problem is compounded on bump maps which have been hand painted as opposed to generated from geometry as these are more likely to have irregularities in their depth values.
However when looking at the specular components reaction to bump maps it quickly becomes apparent that really quite heavily aggressive settings can be used before it begins to be unnatural. Fortunately when using a layered approach to skin this is easily achievable as the base lambert and both speculars can be given different bump values. In fact it's quite common to completely remove the bump map from the lambert and just use it on the speculars, the contribution it makes to the base lambert is pretty minimal.

The decision to use displacement should in my opinion be made fairly early in your pipeline and preferably before you get to the shading stage, the fundamental problem is that the native Maya software is absolutely terrible at rendering fine detailed displacements and although Mental Ray is a vast improvement it is still a slow and tedious process getting good displaced renders. However if you have access to a renderer like prMan, Renderman for Maya or Mantra then displacement becomes a genuinely viable option as the performance hit is minimal and the setup process painless. If displacement is being used then most of the basic concepts for shading skin still hold true but it becomes unclear how it affects bump mapping, this isn't something I've done any real testing into but I would have thought that if you were to use displacement it would sit well with going for a subsurface scattering approach to skin (may as well push the boat out) and to largely ignore the bump component.

An example of how bad an overly bumped lambert shader can be

Subsurface scattering

This is a relatively new technique which is gradually becoming a more realistic alternative to skin rendering as implementations improve. Fundamentally this is an approximated simulation based approach to skin rendering where light rays sent into the skin are scattered beneath the surface as they hit each skin layer and eventually return from subsurface and output the skin at a different point to where they came in. The effect is that light will transmit through relatively small thicknesses of skin in a more believable way to the lambert model which is more suited to infinitely dense materials.

Maya's current main implementation of SSS is in the form of a set of shaders for Mental Ray which are built in for Maya 6.5 and above. These shaders range from more physically correct lighting models to faster more approximated ones that are specifically dedicated for the rendering of skin, the shader I tested was misss_fast_skin.
It should be noted that using SSS in mental ray isn't a completely painless experience but conversely it wasn't as awkward as I had been led to expect, the support in Maya is pretty poor (the shader doesn't even set up correctly on creation) but thankfully there are some good tutorials available which help you get started. Getting a waxy looking render of skin isn't particularly hard, it's possible to get something which looks pretty good within half an hour or so of playing. However it's during this stage that a lot of restraint needs to be exercised, in my experience the vast majority of CG renders that use SSS massively overdo the effect, the temptation is there that it needs to be pushed really far to show it is there, a good implementation of SSS shouldn't be noticeable, it should just look right.
Unfortunately there are quite a lot of controls available to you in misss_fast_skin and it's not immediately clear which one's are important and which should be mapped. The feeling I get is that the learning curve for this shader is kind of an S shape, getting it to work at all is tricky, then getting something to look ok is relatively easy, but getting something to look genuinely good is very, very hard. Personally I find there are many things in CG I don't have the patience for, I dislike tweaking one setting, doing a test render and then coming back in 10 minutes to see what it did. High level SSS tweaking is one of those areas that as yet I haven't looked into although I'm sure given time I feel I could get to grips with the shader. Despite sounding negative there are a lot of things I truly like about this shader and I was pleasantly surprised with the results.

Tutorials on misss_fast_skin :
http://forums.cgsociety.org/showthread.php?t=214189&
http://forums.cgsociety.org/showthread.php?t=163360&
http://www.lamrug.org/resources/skintips.html

The results of my initial testing with SSS

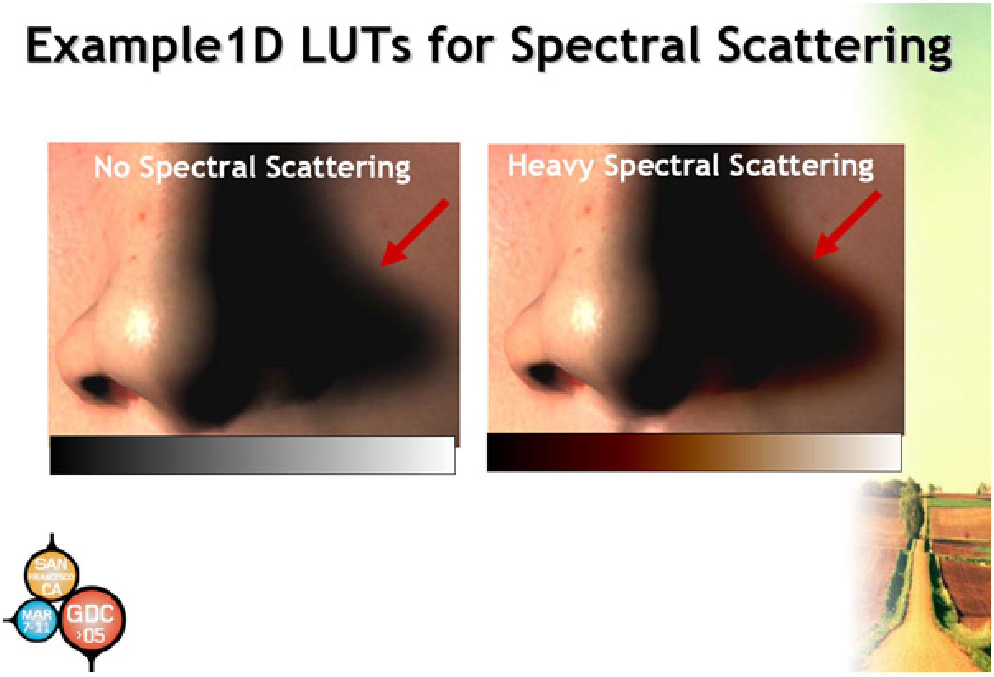
Translucency faking

As well as just using colour tinting and SSS there are other ways that have been proposed to get the effect of translucency. The most promising technique I've seen are those based on workarounds for real time translucency, the first and simplest technique involves the rendering of lighting as a separate pass (or rendering it to a texture) and then blurring this before applying it to your colour texture.

Ati presentation slides on blurred lighting

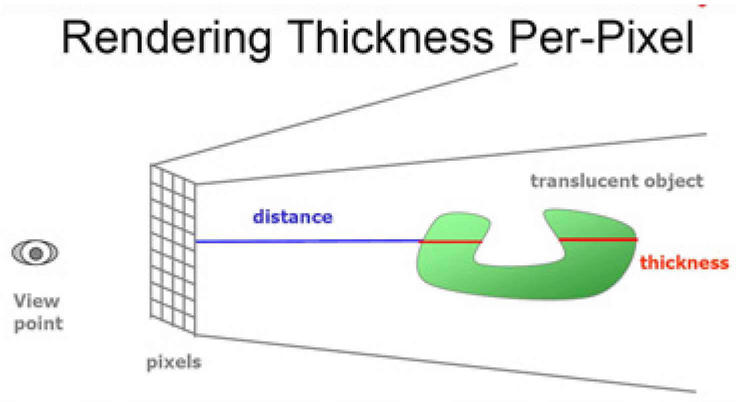
This has the effect of blurring sharp shading boundaries and can fix a lot of the problems caused by light not permeating through a small thickness of skin (such as on the side of the nose). Unfortunately in Maya there is no real easy way of rendering lighting to a texture per frame and then blurring it so the next best solution is to render your lighting pass separately and blur it in camera space in post.

Another good trick based around this technique is to identify where shadow boundaries are in your lighting and as you blur them add a red tint to the edge, all that is needed is something very subtle to have a really positive effect. This is something that can be accomplished in post but it requires rendering out separate shadow passes which are a real pain to work with.

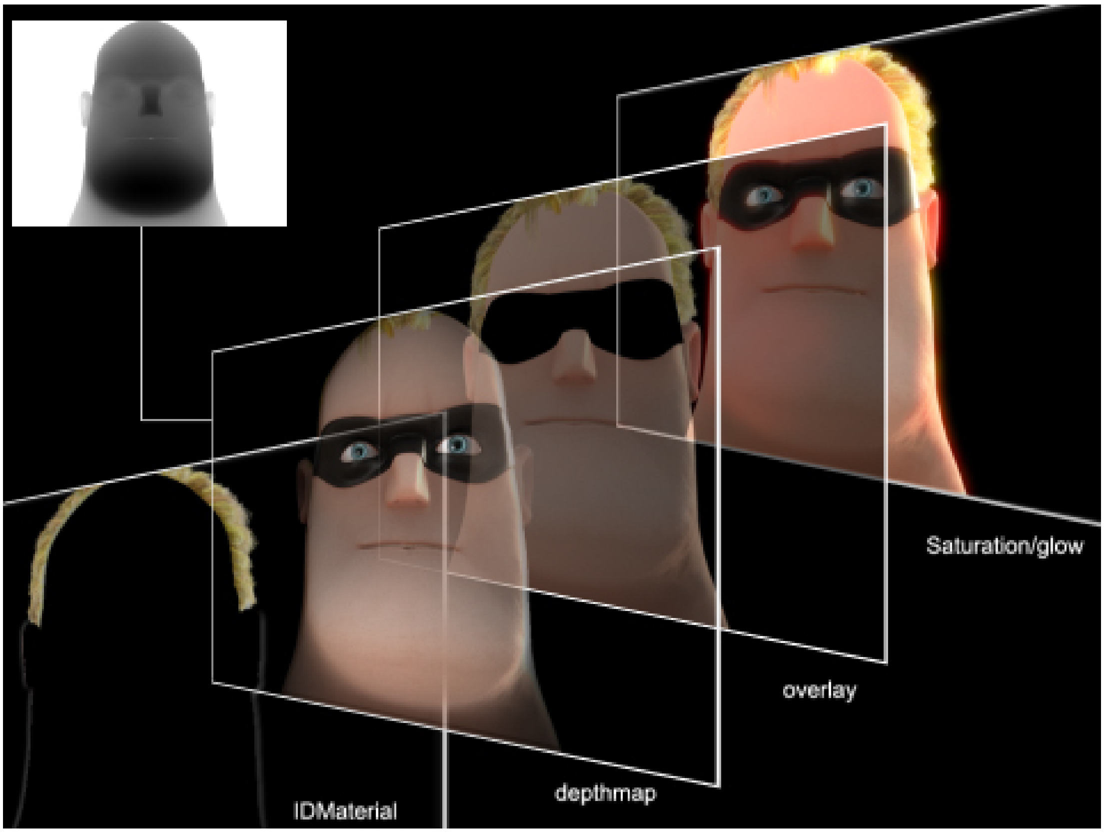


Tinted shadow boundaries

Another broad technique which has some pretty good effects is to approximately compute the thickness of an object by measuring the distance between it's near and back face, then depending on this thickness add a constant translucency value based on a thickness colour reramp. In this way you can quite easily make it so thin sections of geometry get the effect of colour bleeding.

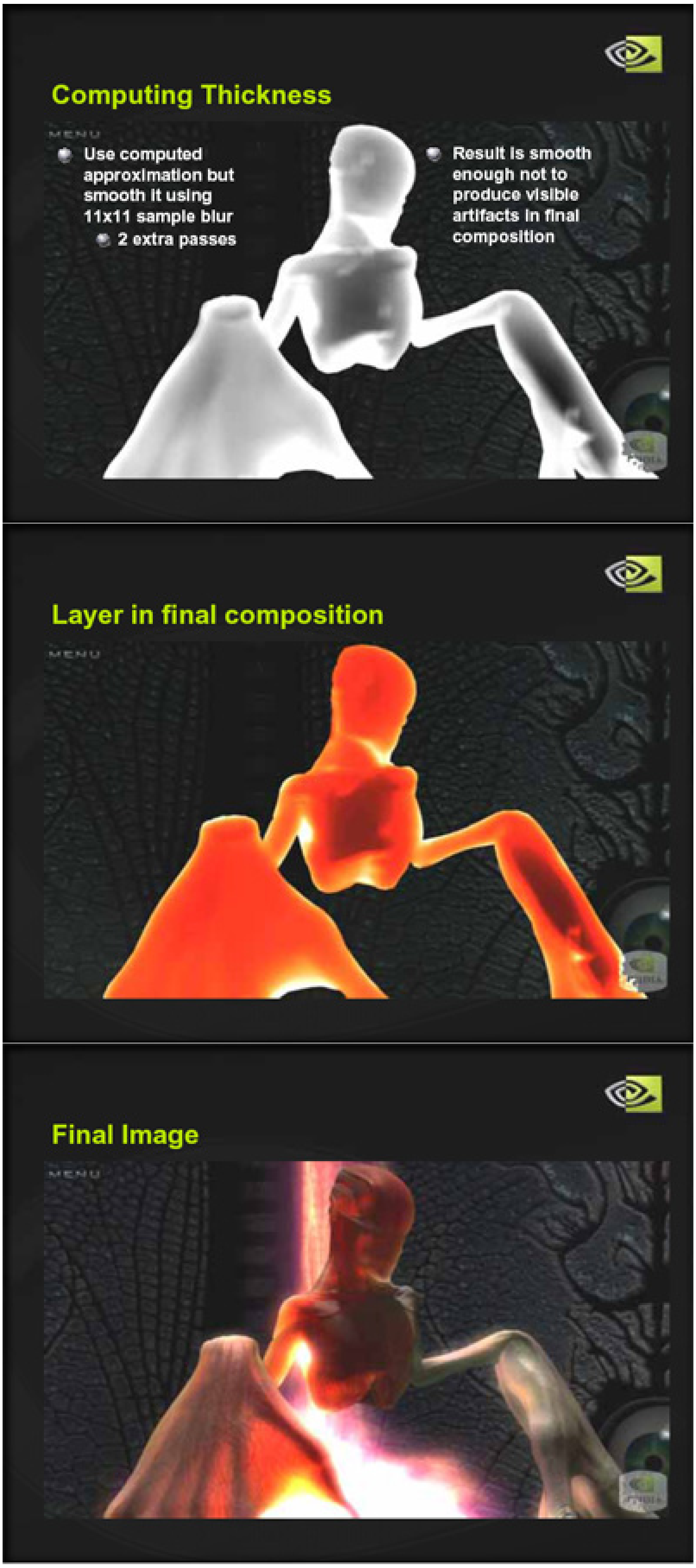


Bried diagram detailing how thickness can be calculated



Good example of this faked thickness based translucency. Note: this wasn't used in the incredibles!

This technique can either be done in camera space where you render out the thickness once relative to the camera or you can do it per light which will give more accurate results but render times will scale pretty linearly. Unfortunately both of these effects require some quite intensive shader writing so I haven't been able to test them out, it's not currently possible to render out a thickness pass from Maya without doing a lot of separate Zdepth passes.



An example of this faked depth based translucency in a realtime nvidia demo

Summary

Layer shader

Pro's
Very fast as all technique's used are relatively simple
As each render is quick it is quite easy to tweak the setup to a high level
The shader can be recreated in post to further increase flexibility
Generally provides an adequate skin rendering solution

Con's
Requires a lot of tweak work to get looking good
Will always be fundamentally a bit too crude for truly high end shots
Can be highly shot/lighting specific

SSS

Pro's
Has the potential to create amazing results
Works well pretty much regardless of light setup
Require less lights to look good
Good to have everything in one shader
Handles shadows very well

Con's
Render times are significantly slower
Setup times are equally slow
Not easy to tweak in post
SSS effect is often overused/abused
Maya's implementation could be better

At the moment the approach I'd advise for rendering skin varies greatly on the resources you have at hand and the quality level you need out of it. If you are doing something reasonably low level like a games character I'd recommend going with a basic layered shader approach and try and tint a lambert lighting model. If you can or have access to someone who can write you custom shaders (could be Mental Ray, Maya or Cg) then I'd recommend implementing some form of translucency fake component to your layered shader setup.

If you are producing characters for mid level graphics work then I would recommend investigating SSS and see how you get on with it, however make sure you don't turn it up too high and end up with people made of wax. If SSS isn't working for you then I'd go for a layered shader approach but I would recommend instead of tweaking the shader in Maya try rendering out all of it's components as a separate pass and tweak it in post. If it's a still then I would use Photoshop, for a sequence my choice would be Shake for compositing if you have access to it.

If you need truly high level character rendering then fundamentally you will need SSS, however I would heavily recommend against trying to get the whole thing done inside misss_fast_skin as you will spend countless hours tweaking and the end result may be a bit unwieldy. I would again heavily recommend rendering things in separate passes and tweaking in post, it may also be worth considering only rendering out a small section of your SSS shader, i.e. Just the scattering component without any specular to allow for more flexibility. If you have them available to you then it may be worth rendering in prMan and investigating some of the SSS options available to you there. A lot of people are often in the mindset that to do something “properly” it should come out perfectly straight from their render without any need for tweaking, while I admire the perfectionism of this goal I wouldn't recommend it as although it may always be possible to make something perfect in your render 9 time out of 10 it would be a hell of a lot quicker and easier to just fix it in post, after all in the end you are just making a bunch of pixels. On the flip side of the coin rendering out a lot of passes requires a lot of disk space and good compositing software to handle them, if you are missing either of them then by all means try and get your renders to come out perfect straight away.

Left : Rendered with a layered shader approach, render times approximately 30 seconds per frame
Right: Rendered with misss_fast_skin, render times approximately 100 seconds per frame
Note that unfortunately the lighting setups are different for each scene as the misss_fast_skin shader requires vastly different lighting setups to get a similar result.

Lighting, Rendering and Compositing

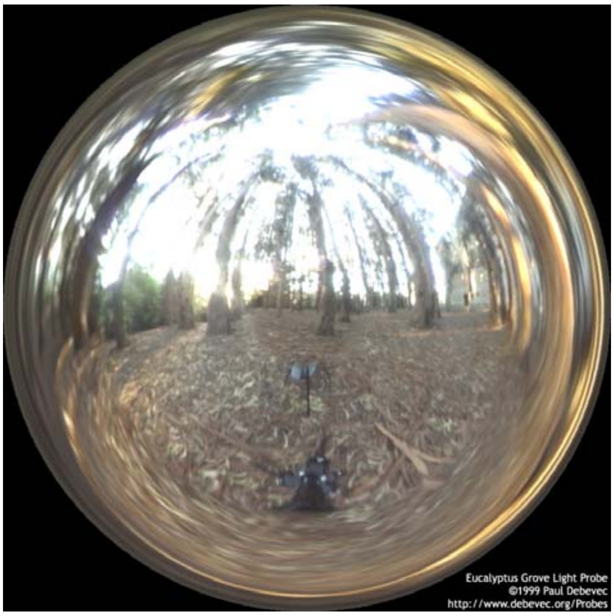
Lighting

Lighting is one of those things in CG which really make or break a piece, a mediocre model in excellent lighting will always look great, an excellent model in mediocre lighting will look dull, lighting is possibly where the shortest time input can give the greatest quality improvement so it's well worth taking a bit of care with it.

The first decision to make is what sort of lighting environment you want to put the character in and how much of the light is ambient and how much is directional. In heavily ambient light setups such as outdoors then you are going to need a lot of lights to simulate light coming from all directions, however you can take a few shortcuts with ambient and reflective occlusion as these are both good for directionless light. If the primary lights are directional then you are going to need to either go for some very expensive render techniques or setup more lights to act as bounce lights to avoid the light being too harsh. It should be noted that with subsurface scattering you can get away with a lot fewer lights and still get fairly smoothly lit results. Another thing to consider with lighting is your colouring, generally speaking skin tends to look better in fairly neutral or slightly blue or green light, if your light is harshly saturated or very red then it's going to be hard to get skin looking good. If you do need coloured lighting then personally I would render out with a relatively neutral lighting setup and then tint and recolour your lighting pass in post.

Real lights vs. Image Based Lighting

This is a big decision to make as it drastically effects how you setup your scene, with traditional real lights you get a lot of quick, fast control but you need a lot of lights to make it look good, with image based lighting you can get pretty good results with minimum effort, however to get truly good results with good render times and no glitching takes a fair bit of time and experience. One thing that is important at this stage is to try not to get confusded between terms like Image Based Lighting (IBL), High Dymanic Range Imaging (HDRI) and straight High Dynamic Range (HDR). Briefly speaking image based lighting is the important one to remember, with this instead of making lots of little light sources to get smooth lighting you create a sphere that casts light and the colour and intensity of light cast from this sphere is determined by a texture. However to define intensities as bright as the sun and as dim as a candle at the same time is tricky with a texture with only 256 levels of brightness, it then becomes essential to increase the texture depth to a higher range so it can include a higher range of values. When you combine image based lighting with a high dynamic range format then you get the process known as HDRI.
Excellent tutorial on IBL and more:
http://www.jupiter-jazz.com/wordpress/wp-content/data/tr4kv2/html/chapter1-FG.html



Classic Paul Debevec example of an HDR light probe used in image based lighting

Traditional light setups
Pro's
Lot's of flexibility
Very fast render times
Minimal render tweaking needed to get a clean result
Shadows can be easily calculated

Con's
Takes a lot of light sources to simulate ambient/outdoor lighting
Setup time for making all these lights can be high
Where lights overlap influence you can get banding, particularly in shadows
Fairly hard to match real world lighting conditions

Image based lighting
Pro's
Good results can be made very quickly
Potential to make stunningly realistic lighting – the lighting is essentially real
Fairly easy to match real lighting of an object in CG

Con's
Can take a lot of experience and time setting up without glitching
Render times are pretty slow
Shadows are either non existent or very slow

My experiments with image based lighting on skin haven't been too successful, when doing IBL you have the choice to either just used the IBL for the colour of lighting and use shadow casting real lights for shadows or you can tell the IBL to emit light which creates shadows but is very, very slow. I also had problems getting settings tweaked well enough to remove final gather glitching which is always going to be a problem. Overall I wouldn't recommend going for image based lighting for character work unless you:

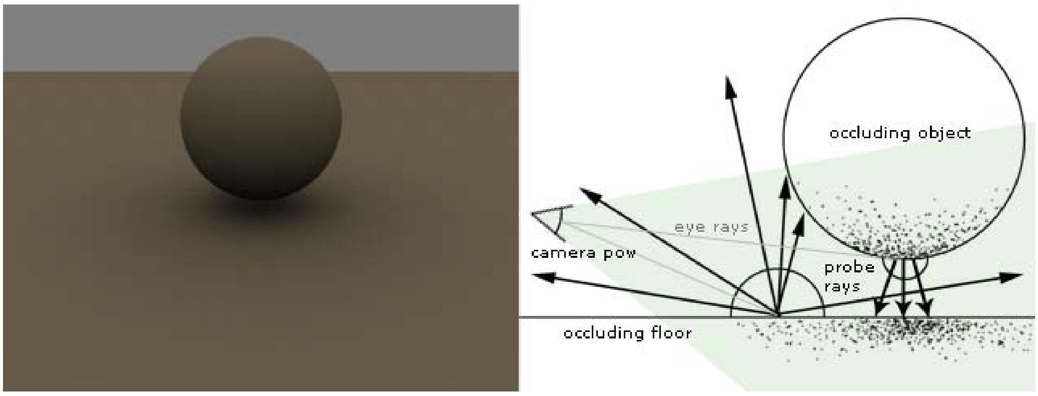
- Need to get a character integrated into real world footage
- Are aiming for very high end photorealism
- Are doing stills
- Have a lot of render time at your disposal

I would instead go for a more traditional lighting setup for the flexibility and speed it offers and possibly use occlusion passes to help give the impression of indirect lighting. One other thing worth mentioning is that in my opinion things like Final Gather and Global Illumination really aren't necessary to get realistic results, the same thing can be approximated using real lights so much quicker. Despite this I have to admit I'm not an expert in Mental Ray and it may well be that it's possible to setup things like IBL, Final Gather and Global Illumination quickly and painlessly, it's just that I had problems doing it.

Left: Imaged based lighting in final gather, no shadow casting, render times of approximately 5 minutes
Right: Imaged based lighting with emitted light. Render times approximately 15 minutes
Note both renders still exhibit significant rendering artifacts.

Occlusion

An occlusion pass can be of many forms but by far the most common is ambient occlusion which is used to represent light coming in from all directions in the scene, getting trapped in recessed area of the model and darkening them. Technically speaking what is happening here is that for every pixel being rendered a number of rays are sent out until they hit the model, when they do they bounce off and try and essentially “escape” the scene and not hit any geometry. If they do hit some geometry after they bounce then they return a black value, if they escape they return white. In this way you get the illusion of light coming in from all directions illuminating the exposed surfaces of the geometry. The conventional way of using ambient occlusion is to render it out as a separate pass and then apply it as a multiply operation in post, when doing this to get best results specular passes should be rendered out separately and applied after the ambient occlusion as speculars are reflective and are not occluded in the same way as ambient light.



Ambient occlusion diagram

When applying ambient occlusion to skin great care needs to be taken, on one hand you have a fairly fast tool which can be used to give the impression of indirect lighting, however if you apply ambient occlusion as you would on other materials then it will blacken all areas in darkness. This is something that absolutely kills the look of skin as shaded area's of skin need to be given a deep red hue to avoid feeling dead. This doesn't mean you shouldn't use ambient occlusion, just that you need to be careful how it is done, some options for this are:

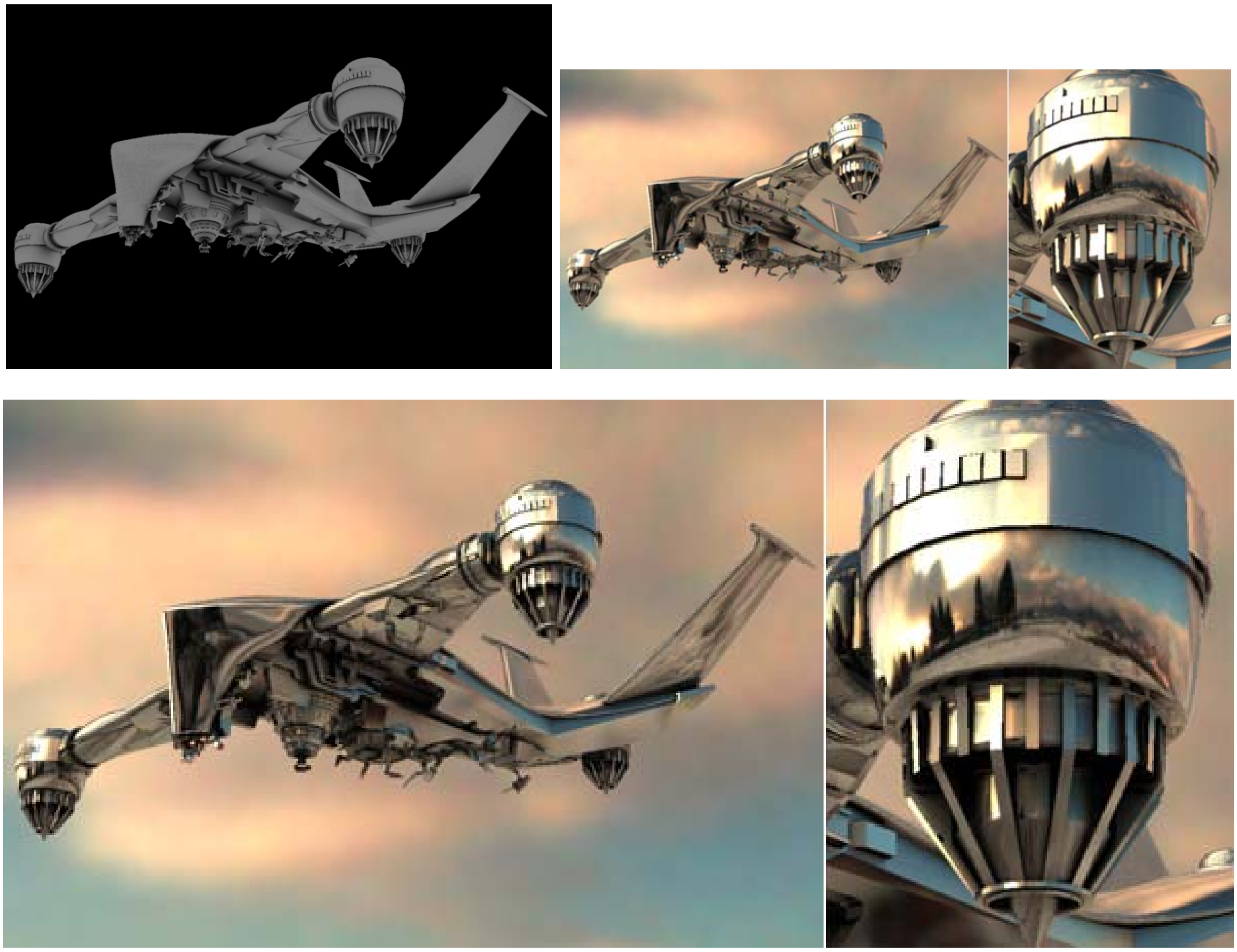
- Apply it on top of your lighting pass which is then recoloured to add red to shaded areas.
- Multiply your ambient occlusion with a dark red before multiplying with your render, in this way it won't blacken shadows.
- Use the layer blending mode “colour burn” to add saturation intensity to shaded areas, to do well requires very subtle settings.

An example of the deading black shading effect when conventionally multiplying ambient occlusion

Another thing to consider when using ambient occlusion is what other occluding geometry you wish to include in your pass. If you just include the single character you are rendering then the occlusion pass will appear fairly dull as light can escape from all angles, in reality for most cases you only really want the appearance of light coming largely from the sky so a good trick for this is to create a large ground plane and place it beneath the character (not to infinity though, just quite big). In this way you should get a pretty good approximation of something being lit from above. It is also possible to further tweak your occlusion setup to approximate more advanced lighting, if you were to block the ground plane and one half of the sky with half a sphere then you could get the illusion of being lit from one direction. However when doing these kind of tricks it's a good idea to leave small holes in these direction's otherwise the lighting will be completely black. Trying to hack directional lighting setups is something I'm still looking into to see if it's viable, the main problems so far are the more you narrow your light's ability to escape the more rays are needed to get glitch free renderings. Overall I wouldn't recommend using this yet but it may be something to play with, however I would definitely recommend using a small ground plane when rendering ambient occlusion.

Something a lot of people aren't aware of is that there are more forms of occlusion than just ambient occlusion, quite a few can be accessed from the occlusion shader found in mental ray. The most useful sub type of occlusion is reflective occlusion which is very similar to ambient occlusion, however when rays bounce off geometry instead of following a path for a diffuse light ray they follow a reflective light way which creates an angle dependent occlusion render. This render gives an approximation of self reflectance where area's on the model that would reflect themselves are shown as black and other area's white. The main two uses of this are either to multiply by your specular pass to increase realism or to multiply by an environment map pass which greatly increases the believability of being in a reflective environment.

The other output modes of occlusion are largely used in 2d relighting.

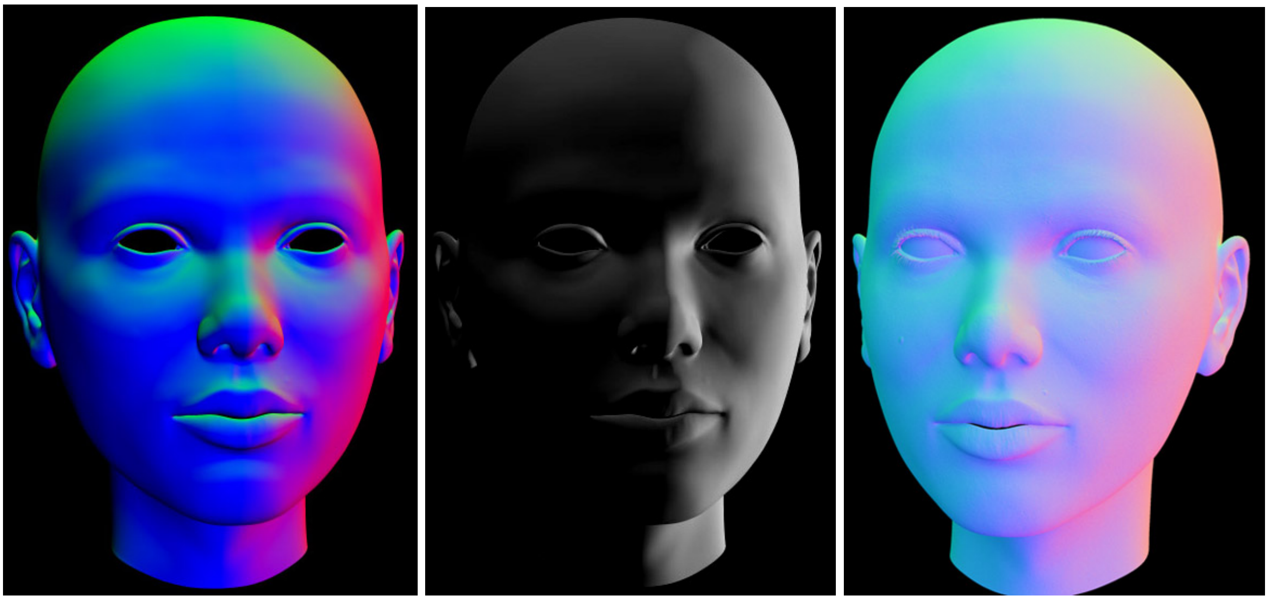


An example of reflective occlusion (top left) used to multiply an environment map (top right) to produce the bottom result

2d relighting

This is a very interesting field which is being used more and more in very compositing heavy shots, the basic idea is that instead of rendering your lighting in your scene you instead render out the surface normals of all the geometry in your scene so that you get all the information needed to know which direction things are pointing in and how they react to light. In the most basic implementation of this you then take this normals pass and essentially apply a directional light to it to relight these normals, this is computationally a very, very quick operation and it is possible to add multiple lights in post to give the impression of being lit from many different directions. The limitations of this are that none of these lights will cast shadows (a separate shadow pass is required), all lighting is diffuse and that as they are treated to be directional lights all light is being cast from infinitely far away. Despite these huge limitations it's still a useful trick as you are able to tweak your lighting setup more or less instantly without having to set off new renders.
Another limitation with rendering out a regular normal pass is that by default normals only go up to being perpendicular to the camera, nothing further back than this is considered, this means that when doing a standard multiply of the normals to get lighting it becomes impossible to simulate back or rim lighting. To get round this you need to render out what is known as “bent normals” where more extreme lighting angles can be accomplished. To do this uses some of the other output modes of the occlusion shader, output mode 2 renders out bent normals relative to their world space directions and output mode 3 relative to the camera, normally you want world space normals unless your camera isn't moving.
Another thing to bear in mind is that although in the crude implementation of post lighting you only use diffuse directional lights it is possible to create extensions of this to include things like point lights and speculars, however to do this involves a fair bit of research and the use of some fairly slow custom shake nodes.

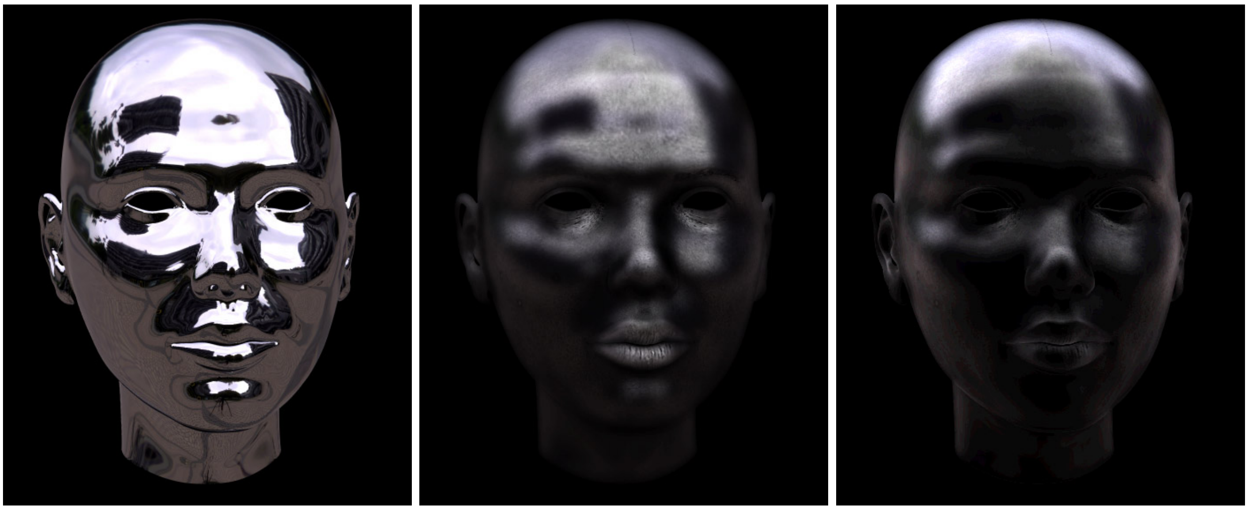
Overall 2d relighting is an interesting field to look into but at the moment I would say the limitations on speculars and shadows plus the high render times for bent normals make it unsuited for use in character work.



Left: Render of conventional normals, render time of a few seconds
Middle: Normals relit from the right
Right: Camera space bent normals, render time of a few minutes

Specular highlights vs. Reflection Maps

One of the oddest situations in the most common shader models is the way that specular highlights and reflections are considered as separate entities, in real life highlights are reflections! In CG only light sources create a specular highlight but in real life everything of any brightness will be reflected, not just light sources. The problem this causes in CG is that to reflect everything is very slow but specular highlights look fake as they are just occasionally small areas of space that give off perfectly round highlights. One solution to this is to use area lights, another is to use reflections instead of speculars, both of these are intolerably slow. Another is to use lots and lots of specular lights in clustered groups, this is functional but not ideal.
The other solution which is actually viable is to use an environment map to replace speculars. In this way the overall brightness of the scene can be captured to a texture and can be applied as if it was a reflection with very fast render. It often helps to modify the texture before hand by upping the contrast level between bright and dark levels.
To get any kind of believability it is essential to multiply your environment map by your specular map to break up the reflections the end result can be quite convincing but can then be greatly improved by using reflective occlusion as another multiplier so approximate self shadowing. However rendering reflective occlusion on top does greatly slow down renders, just probably not as much as if true reflection's were being used. Another bonus with using environment mapped reflections is that it becomes easy to blur your reflection (can be done on the texture or in post), this greatly enhances realism as most raytraced reflection's come out looking too crisp to be believable.
The main problems with reflection maps is the fact that you are using a static environment texture to represent reflections of something infinitely far away, it's quite surprising how far this holds up but in a lot of cases it just isn't good enough, if either your lighting or some nearby objects are animated then the unchanging reflections will look odd. It is technically possible to render out an environment map per frame but this isn't recommended. The other main problem is that when you use real lights for your diffuse lighting the environment maps for your speculars can be hard to match up correctly.



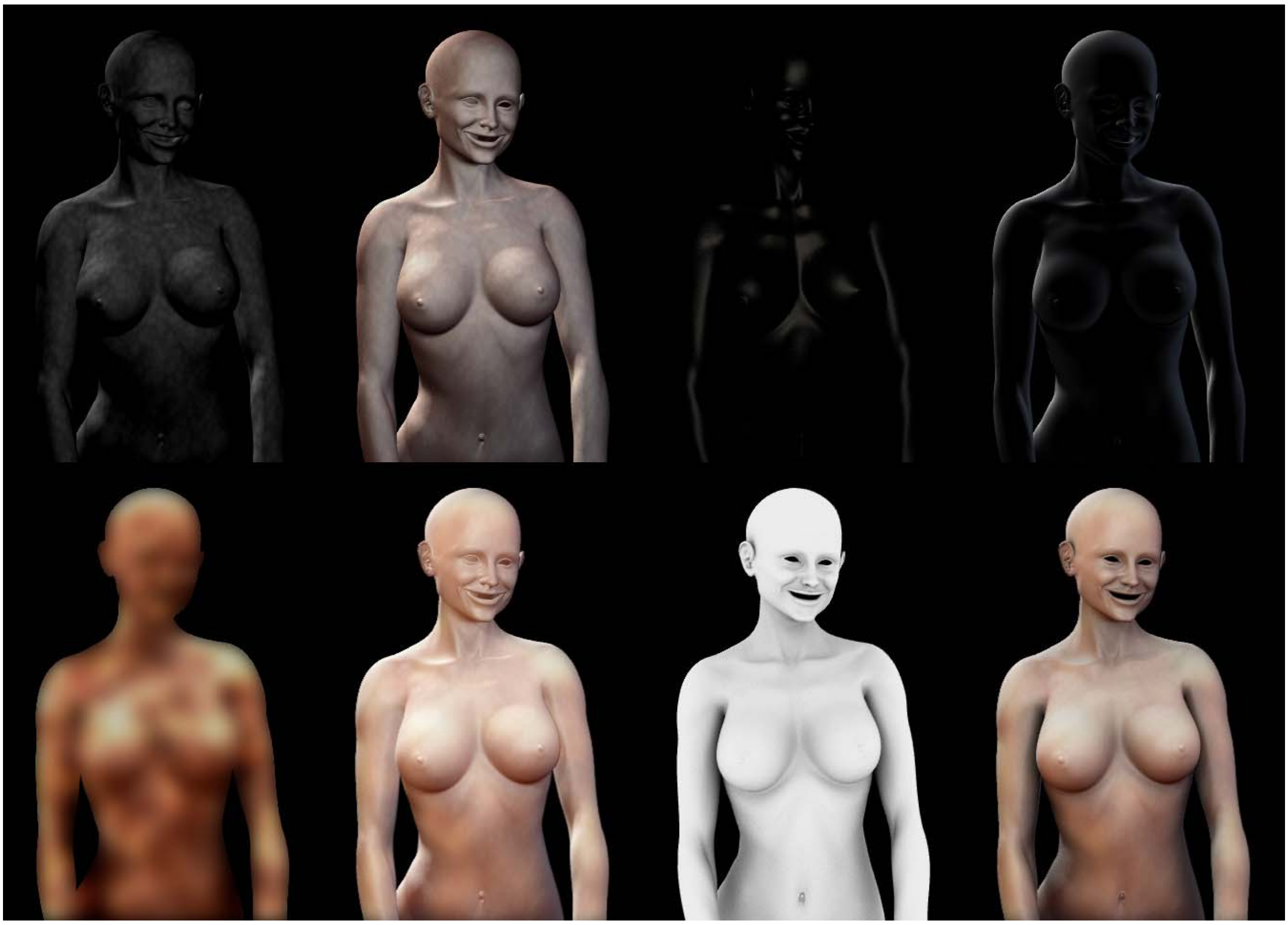
Left: Environment map
Middle: Blurred environment map multiplied by multiplied by reflective occlusion
Right: Middle multiplied by facing ratio, has been contrast adjusted

Overall I would recommend looking into replacing specular highlights with environment maps and seeing how it works for you, the end result can be greatly improved when combined with reflective occlusion but the render times do take a hit. I'd also say that environment maps are very well suited to being applied in post for greater flexibility, you can always blend between environment mapped and traditional speculars as you see fit.

Compositing

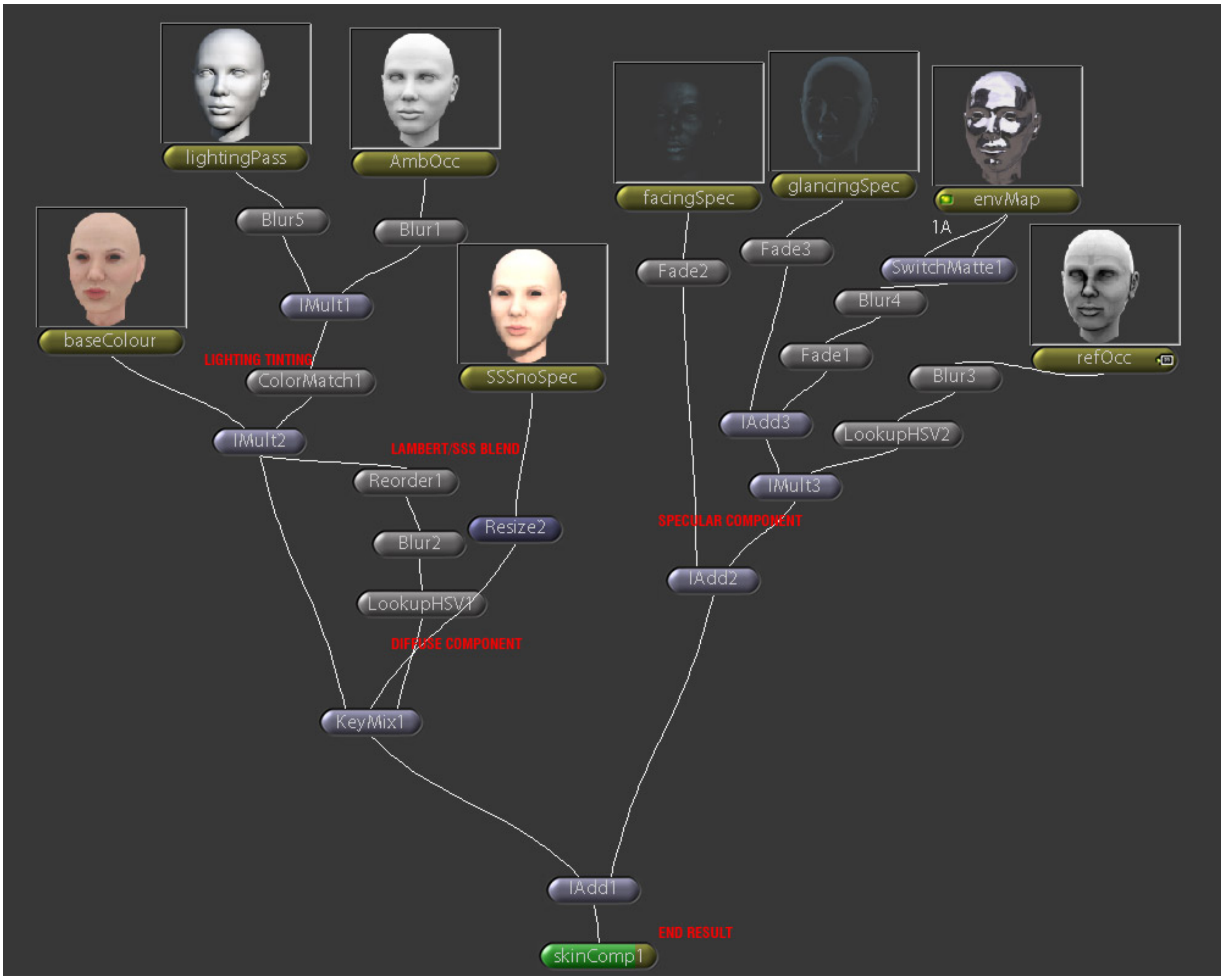
Heavily pass based skin shading

One of the areas I find particularly interesting is using various compositing tricks to approximate the look of skin, this is an area I've done a small amount of research into but I'd like to look further into. Below is an example of an interesting but not completely successful approach to using post production with skin. Here lighting is blurred to approximate the scattering element and then an ambient occlusion pass is used to make the image sit back down on the screen. Unfortunately this occlusion pass has the undesirable effect of blackening the shaded areas which kills the image.



Slighty creepy example of comped skin

When tweaking the skin shader for my character I chose to render out 9 different passes, the key ones being a separate SSS pass without any specular component, separation of speculars down into 3 different layers including an environment map and the inclusion of separate passes for ambient occlusion and reflective occlusion.
The lambert based lighting model is multiplied with my ambient occlusion and then colour tinted to give me more control over the general look of the skin, here red is added to the shadows and a slight amount of blue to the highlights. I found the strength of SSS is it's ability to transmit light through dark areas of the skin so I decided to take the control of my tinted lambert and blend it with my SSS render so that the SSS appeared stronger in dark areas and the lambert stronger in bright. The different specular components are then combined together and added to the diffuse component.



Shake network for my skin shader

Extra elements like eyeballs, eyelashes and eyebrows were then rendered separately with a diffuse pass and a specular pass. This specular pass is then sharpened around the centre of the eyes to give the impression of stronger highlights.

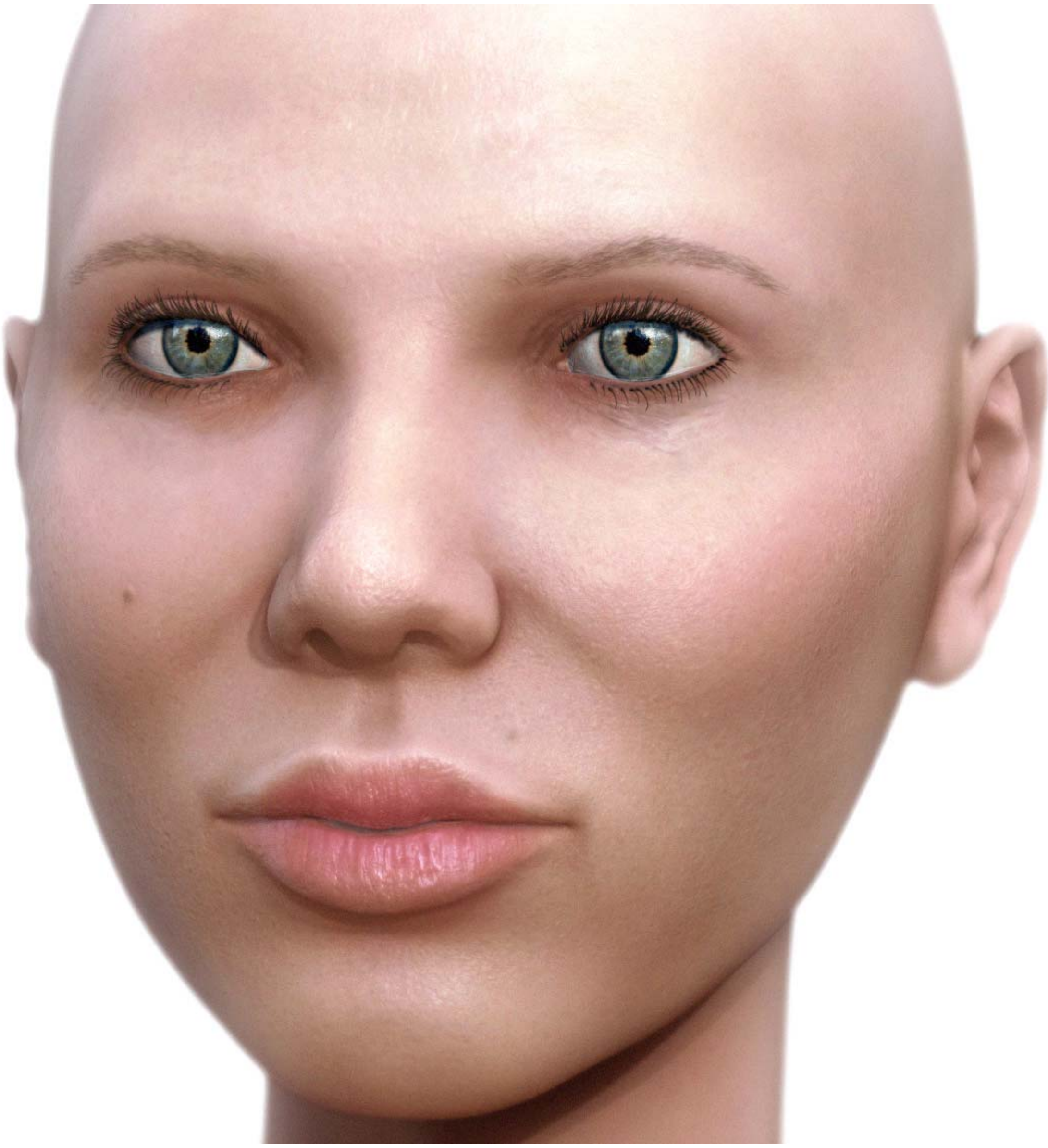
Skin rendering after going through compositing

From this point the compositing was more or less done and I just needed to do some tweak work on overall levels and colour adjustment. One particularly good trick I've found for this process is to take your renders do a blur and then overlay the image back on itself with some tinted colours. If you take the blurred overlay and tint the shadows further towards red and the highlights strongly towards blue you can set the overlay opacity down pretty low and it has the effect of desaturating the more brightly lit areas and deepening the shaded areas. The effect is fairly subtle but the overall colour improvement definitely helps.

Left: colour tinted blurred overlay
Right: overlayed result

Other areas to look into at this stage are tweaking the contrast around the eyes and also possibly adding trace elements of film grain. If you are working on a still then you can take your base render and seriously spend some time retouching areas. Particularly important areas to look at when retouching stills are adding red/blurring any sharp shaded areas such as those around the nose and ears, despite this the most important area is almost certainly the eyes as they are the focal point of the image. It's essential that these have sufficient contrast and stand out correctly.

In this example I took a still and added a stronger watery layer around the eyes plus I increased the highlights in the eye quite a bit. I then swapped out the black background for a white one and slightly blurred the edges to make it look more natural. Looking back on this image now the eyes still aren't terribly convincing and the eyelashes are too dark, to improve it I would look into these areas and probably consider adding a real background and trying to get some functional hair. Note hair is one of those things which is surprisingly easy to paint in a 2d still but surprisingly hard to get working correctly in 3d.



End result of postwork

One other thing worth noting is that I don't have any real idea how long render times are for this image as I rendered things out in separate passes without timing them. I'd estimate that the combined time would be around 15 minutes per frame but with fairly substantial setup times on top. If you needed to improve render times then the first pass you can afford to lose is reflective occlusion as this takes a long time to render and doesn't contribute that much to the end result. It's possible that if you were to try and recreate this image in one shader network in Maya you may be able to get it to render quicker but I'm firmly in the camp of fixing things in post as the flexibility you gain is in my opinion invaluable.

Miscellaneous

Please note all of this section is still work in progress.

Hair

Coming soon, I intend to go through:

- Texture based
- Paint effects
- Maya hair
- Maya fur
- Geometry
- Geometry + Alpha maps
- Commercial hair solutions
- Hair dynamics
- Lighting, Shadowing and Rendering of hair

For now, I will give you four words of advice:

Just *Wear* *a* *Hat!*

Work In Progress

Eyebrows

For eyebrows there are essentially three choices, these are to use textures, paintfx or fur. I find that paintfx or fur by themselves cause problems as they are difficult to style and often look stuck on when rendered. Textures have are conversely easy to place and style but lack depth and profile when viewed up close. Therefore in my experience the best way of going about it is to predominately just use a flat painted texture and to then place a thin layer of paintfx or fur on top just to provide volume and break up the silhouette, in this way you get the best of both worlds. One important thing to note is that if you do use paintFX it's a good idea to turn off any form of shadow casting for the eyebrow strokes.

Eyelashes

Eyelashes are a slightly trickier prospect, in this case your main choices are textured alpha planes, physical geometry, painfx or Fur. In my experience paintFX and Fur are too difficult to style and control so aren't worth the time setting up. Alpha mapped planes can work pretty well but have the problem that they lack depth and variation and that they don't shadow correctly with depth map shadows. As the shadows ignore the alpha of the geometry you get solid shadowing all around the eyes which looks very out of place. The solution for this is to either use raytraced shadows which is slow or to separate the eyelashes to a seperate piece of geometry and turn off cast shadows for that geometry.
Instead my preferred technique is to use physical geometry for eyelashes which can itself be converted from paintFX as a base. This gives complete control and doesn't require any abstraction as to where they are placed, they also shadow correctly by default. The drawback of this is that the geometry won't by default follow the eyelids so will need some clever skinning.

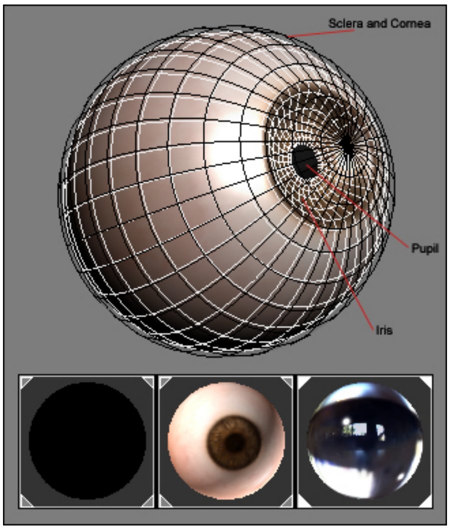
More detail and pictures to follow.

Work in Progress

Eyes

Modelling Eyes

There are fundamentally two different approaches to use when modelling the eyeball, the first is to model it accurately with separate geometry for the lens and to try and accurately replicate the shape and mechanics of the eye. The second is to approximate the modelling of the eyeball to essentially a sphere and try and tackle the more intricate parts of the eye as a shader trick. If you go for the former then it more closely mimics the behavior and rendering of a real eye but it does cause issues when moving eyes around the eye socket – as the eye is no longer a sphere it can pop through the eyelid at various locations due to it's shape.



An example of how the eye can be modeled in two distinct sections

Shading Eyes

What essentially sells an eye as being "alive" is reflections, pure and simple, the highlights on and around the eye need to be very carefully done as it is one of the main things that will sell the character as being alive. To get a truly accurate deep look you should also treat the lens as being refractive.

Eye Fluid

A small but important piece of the puzzle in making an eye look believable is to include the layer of fluid that runs along the bottom of the eyeball as it joins with the eyelid, this is also present on the top of the eye but to a lesser extent. All this liquid really does is show up in speculars/reflections but the only real way to get it to work is to physically model it and assign a largely transparent but environment mapped/reflective shader. This is a relatively easy thing to do in stills but can be a problem on animated shots as this liquid is usually done as a separate piece of geometry, however it still needs to follow the eyes quite accurately so again some clever use of skinning and blendshaping will be required. Failing that you could attach this bit of geometry to the eyelid which would simplify this process.

More detail on eyes to follow, also expect to see a section on texture creation.

Summary of Workflow

In case you don't want to read through 70 pages of talking I thought it would be a good idea to summarise the workflow I used on this project:

- Creation of base polygon mesh, in my particular example instead of starting from scratch I reused an old mesh, this isn't particularly important though.
- Sculpting and soft modification of mesh to rough proportions
- Remodelling and improving of existing topology, ensuring pure quads
- Model UV'ed using automatic mapping, sew UVs and relax UVs
- High res sculpt generated using skin reference pictures and some skin alpha maps
- High res sculpt used to generate bump map
- Ambient Occlusion map and Curvature map baked from high res geometry
- Colour map hand painted based on Ambient occlusion and Curvature bakes
- Specular map created from curvature bake base
- Lighting setup created out of a handful of spotlights
- Initial tests into layered shader techniques
- Skin is rendered using a hybrid pass based approach, half sub surface scattering, half tinted lambert
- Other features like eyes, eyelashes and eyebrows are quickly generated
- These are rendered in a separate pass and then comped
- All these images are then brought together in post and adjusted to fit

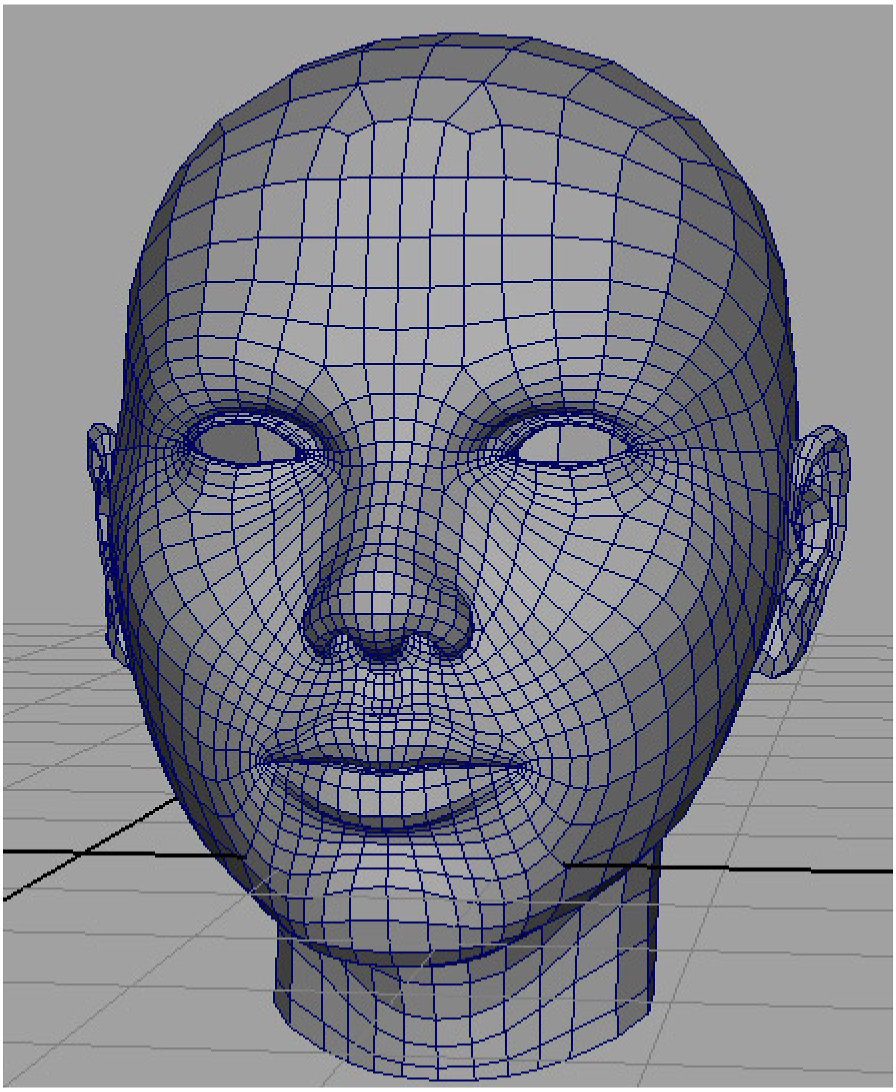
From here you now have a great base to work with for taking further and animating, everything done so far has been done with the possibility of animation in mind and there shouldn't be any problem in adapting what has been done so far to make realistic animation a possibility.

To take this model further these are the stages I will be going through, it should be noted that as of yet I don't know how much of this I will continue to document, plus the further I get into facial setup and animation the further I go out of my field of experience. I can't claim that my proposed techniques for this will be the best way of doing things. Despite this the next stages I plan to take with this model will be:

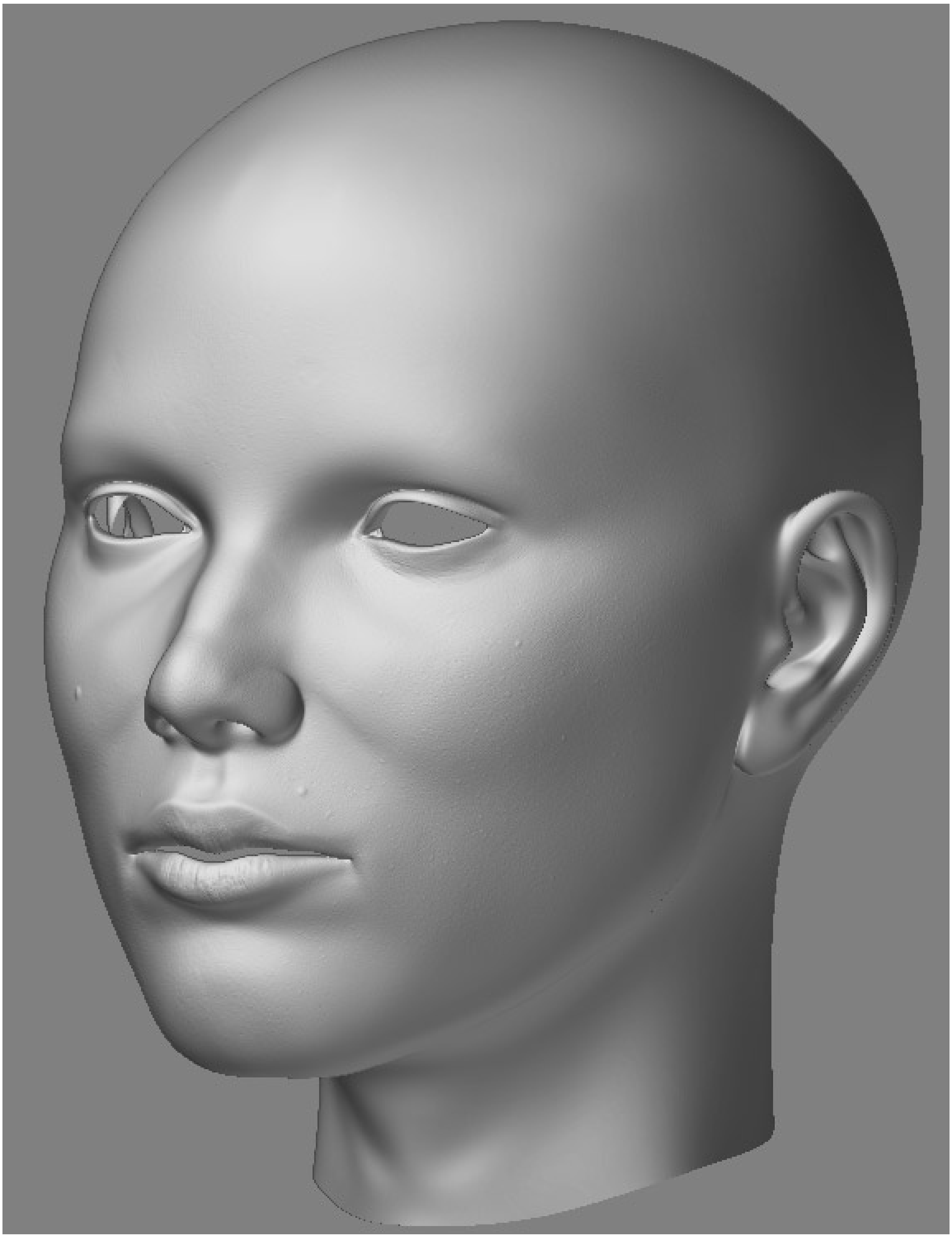
- Create some kind of hair solution
- Begin making blendshapes
- Improve the eyes
- Implement teeth and the inside of the mouth
- Create a neck and torso
- Implement a blendshape based facial rig
- Create driven bump maps for wrinkling based on facial expressions

Documentation of progress on the model

Initial reference pictures taken from Lost in Translation. It's worth noting that I'm not too worried about a perfect likeness with my reference, I'm just aiming to capture most of the facial features.



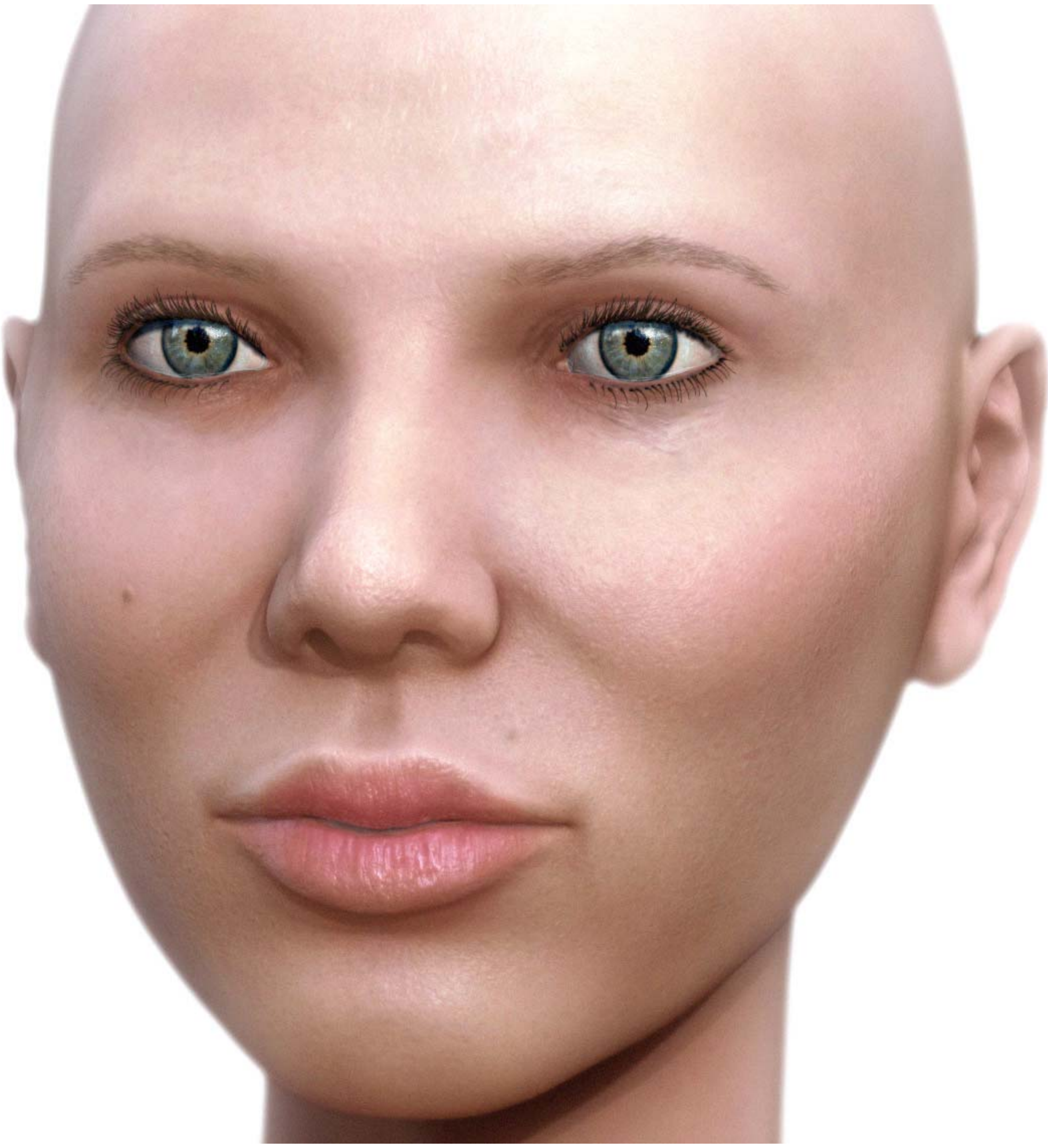
Wire mesh



Initial high res sculpting test

Left: Layered shader early render. Right SSS early render

Later Render with eyes added. This uses a layered shader



Latest render. Using a very comp heavy approach with a hybrid layered/SSS approach to skin. Note that I intend to add Hair at a later date and the eyes are still unfinished.

Link to latest turnaround:
http://www.vertpusher.com/files/scarlettTurnaround1.mov

Work in Progress thread:
http://forums.cgsociety.org/showthread.php?t=310742