

PHOTOREALISTIC TEXTURING FOR DUMMIES



Elven armour (new)

Leigh van der Byl | <http://leigh.cgcommunity.com>

LightWave 7.5b (all textures handpainted in Photoshop)

Leigh Van Der Byl
<http://leigh.cgcommunity.com>

Index

Part 1 – An Introduction To Texturing3
Introduction 3
Why Procedural Textures just don't work 3
Observing the Aspects of Surfaces In Real Life 4
The Different Aspects Of Real World Surfaces 5
Colour..... 5
Diffuse..... 5
Luminosity 6
Specularity..... 6
Glossiness / Gloss 7
Reflection..... 7
Transparency and Refraction..... 7
Translucency..... 8
Bump 8
A Note On Making Your Own Texture Maps..... 8

Part 2 – Standard Projection Techniques vs The Painstaking Art of UV Mapping 10
What on earth is UV Mapping? 10
UV Unwrapping 13
So what do you do now that you have the unwrapped UV map?..... 15

Part 3 – How To Make Great Colour Maps 17
Introduction 17
How To Make Really Great Colour Maps..... 17

Part 4 – The Diffusion Confusion 21
What? Diffusion isn't colour?!! 21
Putting Diffusion Into Practice 22

Part 5 – Specularity, Glossiness And Reflection 27
The Need For Some Shine..... 27
Specularity – The Big Fake..... 28
Reflection – The Real Deal 29
The Big Showdown – Specularity VS Reflection..... 29
So, how about some tips for working with this stuff? 31
Ummm... wasn't this workshop also supposed to be about Gloss??..... 36

Part 6 – Translucency, Transparency and Refraction..... 38
What is Translucency 38
Using translucency in your surfaces..... 39
I keep hearing about this thing called Sub Surface Scattering – what is it??..... 40
Transparency 41
Using transparency 41
And what about refraction? 44
That's all folks! 45

Part 1 – An Introduction To Texturing

○ Introduction

Hello everyone, and welcome to the first installment of my workshop on advanced texturing. Before we go on, let me just point out that **under absolutely no circumstances are you going to find any tutorials on how to make tacky chrome surfaces or light blue glass.** There is quite enough of that out there already. Not that I think there is anything wrong with those, it's just that those tutorials will only get you so far, and when you are suddenly faced with the challenge of texturing a photorealistic shower attachment or elephant's toenail, they are going to get you nowhere.

What I intend to do is to illustrate exactly how to observe real life surfaces and recreate them digitally. With all the frills. And that means absolutely no tiled texture maps.

I am going to try and be as non-software-specific as I can, but I don't claim to know all the ins and outs of the surface editors of many high-end 3D software packages, so I will leave it up to you on how to figure out the equivalents in the software that you use. However, I will try and use the most generic terminology I can think of, to avoid unnecessary confusion.

○ Why Procedural Textures just don't work

Now procedural textures are something that I am very passionately against. I know that they are great for when you need to create a certain look in a very short amount of time, but if you want to create something which is believable, they are just not going to cut it. Procedural textures are fine for beginners, and even some intermediate 3D artists, but there comes a time when you are going to have to leave them alone. Completely.

Why is that? You may ask.

Let me explain. You see, nothing in this world remains untouched – whether it is touched by the weather, by animals, or by man, the world is interacting with it in some sort of way. When you are texturing something, you have to start off by asking yourself a few questions – what is this object used for? Where is it found? How do people interact with it?

Now if you wish to make your textures absolutely believable, they are going to have to show how it is that the world is interacting with it on a regular basis. For instance, a child's toybox, would become worn in streaks below the lid where the child's hands drag as they reach to lift it (this would be most noticeable in the specular layer); an old telephone with one of those old dials would have oily finger marks around the little holes where your finger goes in to turn it; a computer mouse would also have oily streaks on the buttons where your fingers slide up and down as you use it. These are all examples of our interaction with objects and the effect that these interactions on their surfaces.

Now, to try and recreate these very specific, familiar day-to-day effects, you are going to have to ignore the very tempting Puffy Clouds and Fractal Noise procedurals, and actually make an effort to do them yourself.

Than means, painting your own textures.

No amount of clever tweaking of a Hybrid Multi-Fractal combined with a Perlin Marble procedural pattern is going to be able to recreate the scratches and worn paint around an old screw on a tractor which has been re-tightened, over and over again over the years, where the screwdriver always drags hits and drags when the person holding the tool tries to get to the screw itself.

There are, of course, a few exceptions to this. Liquid surfaces, for instance, can only really be made and animated with procedural noise, if you have no access to a decent soft-body dynamics engine. They can also be useful for bumps maps for surfaces like skin, or cloth, that can have a very uniform bump across a large surface. However, they should never be used alone for things like these, as even a surface like skin changes in specific areas where it creases along joints or if there are any blemishes, and these details should obviously be added using a unique painted image. But apart from instances like these, the only way that you are ever going to achieve truly perfect and photorealistic surfaces is to paint them yourself.

○ **Observing the Aspects of Surfaces In Real Life**

If you wish to become a brilliant texturing artist, the first thing you are going to have to do is learn how to properly observe surfaces in the world around you. You need to look at things and, in your mind, break the surface apart into its different aspects. Look at a brick wall – study the variations in colour, notice how the light is broken up along its surface, feel the grooves, scratches and bumps on the bricks themselves as well as the grain of the cement between them. Even take note of its temperature – I know that sounds strange, but that can become very important when you are trying to make people really believe that the surface exists. People looking at your textures must be able to imagine exactly how it feels to touch, and the temperature of the surface is definitely important, but that is a detail that I will go into depth at a later stage.

You need to begin looking at the world around you in terms of the aspects that you will recreate digitally. I know this may sound really bizarre, but when I am driving to work in the mornings (this usually being the only time of day that I actually see the world outside...), I often find myself looking at an old garbage can on the side of the road, and thinking “hmmmm... that can has an awesome diffusion map on it” or looking at an old council truck and thinking “what a cool specularity map that has”. It is because I think like that, that when I sit down to work and am given something to texture, I can draw directly from my memory and experience, because I do know offhand what the diffusion of metal looks like, or how to do really great rusty streaks on the side of a truck.

Make an effort to break up things you see into Colour, Diffuse, Luminosity, Specularity, Glossiness, Reflection, Transparency and Refraction, Translucency and Bump aspects. Some surface areas also include Displacement, which is basically a Bump map that actually alters the geometry, but because it is then, technically, a geometric property, I will not deal with it, as it doesn't really affect surface attributes as such. All these different surface properties are generally, across the board, incorporated into your software's surface editor, possibly with the exception of transparency (which is not the same as Opacity), and translucency. Observe how these different aspects interact with each other – for instance, very reflective transparent glass isn't very reflective and is less transparent where it is scratched and where it has been dulled (by being handled or by being wiped by a dirty cloth).



Of course, in order to begin observing all these aspects correctly, you need to have a very good understanding of what each of these are.

Please note: When I speak of procedural textures here, I am not referring to specialised SHADERS written for Renderman, Lightwave, Maya etc. I am referring to the built in fractal-noise type procedurals. And before you begin posting complaints about these statements, please read the following few pages of discussion concerning this.

Real world surfaces are dirty and weathered! One of the biggest mistakes made by beginners is making all surfaces too clean.

○ The Different Aspects Of Real World Surfaces

Of course fully explaining every one of these properties in great depth would take too much time right now, so I am going to give a very brief explanation of each of them now, and then later on I will do a separate workshop dealing with each of them individually. But, in the meantime, here is a quick rundown:

○ Colour

Well, I guess this one is pretty self-explanatory. All objects have colour. Although it is probably the most basic of the surface attributes that you will have to make when texturing something, it is by no means an easy one to produce. Nothing in real life has a constant perfect colour. Any object has a colour which is uneven in places, even if only slightly. The colour map is usually the best one to start off with, when beginning to texture a new object, as it will give you an excellent starting point for the feel of the object. Colour maps are generally the only image maps that you will make that contain any actual colour, as the other surface properties are best created in shades of grey.

○ Diffuse

Okay, to all the Max users out there – diffuse is NOT colour. I've never fully understood why the colour attribute in Max's material editor was named Diffuse, when Diffuse and Colour are actually two very different things. Unfortunately for people who use Max, diffusion is not included in the material editor, in fact, to my knowledge, the only medium to high end programs that do incorporate it in it's correct sense, are Lightwave, Maya and Softimage. This is a shame, because diffusion is actually a critically important aspect of surface realism. Diffusion is the attribute of an object's surface that scatters light. It determines the actual amount of light that is reflected by the surface. In essence, it determines how much of the surface's colour we'll see. By diffusing an object, you limit the amount of colour that is reflected back by the light. This is completely different to simply darkening the surface of the object itself. If you were to darken the actual image used as a colour map, you would only see a change in colour, but not colour depth. Colour depth is created by scattering light across an object's surface. Take a look at human skin and you'll notice that it has a density. The colour isn't a simple continuous shade but rather many similar shades, created by scattered light. This quality cannot be made by a colour map alone, as a colour map cannot give a surface the richness that a diffuse map can. Obviously, having said that, Diffuse and Colour go hand in hand. I often find that the best way to start off a diffuse map is to take the colour map, desaturate it to grey and work from there. But, as I said previously, I will be going into much greater depth later on as to how each of them are best produced.

○ **Luminosity**

This attribute determines whether or not an object has any self-illumination properties, and how strong they are. This is only used for things like florescent lights, light bulbs, LED displays and so on. A luminosity map works best in conjunction with a renderer, which supports radiosity, so that the luminous value can become translated during the render process as a source of light, and will thus illuminate objects around it slightly.

○ **Specularity**

My personal favorite surface property. It is a shame that this extremely important aspect of a surface is all too often overlooked. In fact, I would say that Colour and Specularity are the two most important basic surface attributes. The specularity of an object determines how shiny it is and how light is broken up by its surface. Now, keeping in mind what I said earlier about being aware of how people and weather interact with and affect a surface, you make use of a specular image to show how the world has “made it's mark” on the object, so to speak. Things like smudges from a cloth, fingerprints, wetness, polishing, etc will all play a part in determining what goes into the main specular image. Specularity works most closely with three other surface properties – Glossiness, Bump and Reflection.

Specularity in conjunction with Glossiness determines how spread out or how tight an area the light's “hotspot” becomes when it come into contact with the surface. An object with only a small difference between the specularity amount and the glossiness amount have a very small “hotspot”, therefore they appear plastic-like; whereas an object with a large difference between the two will spread the light out over a much wider area, therefore appearing more dull, like metal (except chrome-plated metal, of course).

Bump works hand in hand with specularity in that generally parts of an object, which protrude more, become more worn. And depending on what the surface is, that will either increase or decrease the specularity – wood, for instance, generally becomes shinier when it is rubbed a lot, whereas metal can become duller. Another way in which bump and specularity work together is in the case of scratches along a surface – a scratch will tend to gather grime and dust, and will therefore become very dull after a while.

The interaction between reflection and specularity is pretty obvious – where a reflective object has been, for instance, touched by a person, oil from the fingertips gets left behind, and makes the surface appear less reflective in that area. Another example would be if you had to texture a car's front windscreen, where it is a cold night and the driver has wiped a cloth along the glass to clear the mist on it, the cloth leaves lines in the direction it has been pulled across the surface – those streaks will make the surface appear a lot less reflective than parts that weren't touched by the cloth. It is probably safe to say that there exists no object in this world that is 100% reflective all over it's surface, because sooner or later, someone or something is going to touch it and affect it's specularity, thereby affecting it's reflection.

You get two common kinds of specularity – normal specularity, and anisotropic specularity. Normal is just plain straight specularity, whereas anisotropic specularity is used for surfaces, which has extremely tiny bumps along its surface. But I will get more into detail about that when I deal with specularity at a later stage.

○ Glossiness / Gloss

As I explained above, Glossiness is the property that determines how widely light is spread out on an object's surface. The moment you add any specular amount to a surface, you must use glossiness to get the right balance of shininess. As previously mentioned, plastic objects have high gloss values, whereas untreated woods (like bark) and industrial metals have quite low gloss values (sometimes, in these cases, no gloss at all).

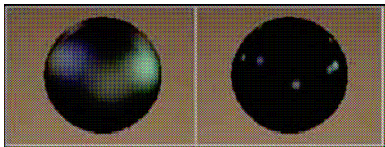


Figure A: The first sphere shows a setting with a highspec setting and a low gloss setting, while the second sphere shows the result of the amounts being closer together.

○ Reflection

Another pretty self-explanatory property, reflection determines how reflective an object is and in what areas it is reflective. As previously mentioned in the paragraph on specularity, the reflection map should vary along the surface according to how it has been interacted with. This attribute is sadly often overused – a huge giveaway that an object is cg is often due to the fact that the texture artist has made it too reflective. That is not to say that reflectivity is not a common property – in fact, most things which are in any way shiny, are slightly reflective too. Although reflectivity is not to be confused with the effect that radiosity has on objects, where the surface of one object can pick up small traces of the colour surrounding it due to bounced light. You should not attempt to create this effect with reflection, as your surfaces will just end up looking wrong.

The kind of reflection that an object has also differs greatly – a mirror, for instance, produces a perfect reflection where it is clean, whereas steel, plastic, scratched Perspex, most liquids, etc, have blurred reflections. A lot of software does have a blur option for reflection – if it's there, **use it**. All too often, you see horrible renders of perfectly reflective kettles and such, when the reflection should have been blurred.

○ Transparency and Refraction

Transparency is not Opacity. It is, in fact, the opposite. Transparency determines how “see-through” an object is (whereas Opacity determines how opaque it is. An object which is 0% percent opaque will disappear out of a scene, whereas an object with 100% transparency will just be completely see-through, yet still visible). Obviously, things like glass, Perspex, liquids, crystal, etc have varying degrees of transparency.

Transparency is also affected quite a bit by the specularity of an object – especially in the case of oily fingerprints left on a surface – obviously these areas are not going to be as transparent as the surrounding areas that have not been touched.

Most substances, which are transparent, refract light. Refraction is the bending of light through transparent bodies. This causes the effect of, for instance, if you have a scene with a glass of water in it, when you look at objects through the glass of water the objects you see are warped. Different substances have different refraction amounts, and a table of these amounts can be obtained almost anywhere – from a science textbook, to the manual that came with your software, to going to Google.com and typing in “refraction

index”. The higher the refraction amount, the more light is bent as it travels through it. Refraction amounts in reality do not exceed 2.0.

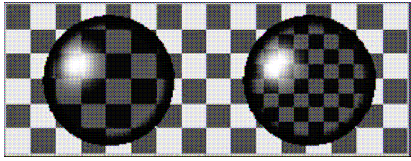


Figure B: The first sphere has no refraction, while the second one has a refraction index of 1.175

○ Translucency

This property, like Diffuse, is not included in all software. Translucency is the ability for an object to be backlit without being transparent. Take for instance, a curtain – when a light is shone through it, you can see things moving behind it, even though it is not transparent. In reality, just about everything, with the exception of metal and wood, has some degree of translucency. This property can be extremely useful for skin, in particular, so that if a bright light is shone onto it, you can catch a glimpse of the veins, which run just below the surface. Translucency works best when combined with a calculation called Sub Surface Scattering, which is a property that basically allows light to enter the surface of an object, bounce around inside the surface, and then leave it at a different angle to that in which it entered. This can usually be implemented with the use of a BSSRDF shader – Bidirectional Surface Scattering Reflectance Distribution Function. Big name, cool effect. I'll deal more with that later on though. Translucency and Sub Surface Scattering are extremely important, especially for surfaces such as skin and fabric, where they should **always** be used.

○ Bump

The most common attribute next to Colour. Everyone knows what a bump map is. Although, I must stress that a bump map should never be used as a compromise for necessary geometry. Bump maps should only be used for minor things like scratches, small dents, grooves, small carvings, minor variations in relief, and grain. **Never ever** use a bump to create something that should actually be modeled. The reason for this is that as soon as you get close to a bump map, it becomes obvious that the objects surface relief is actually flat.

○ A Note On Making Your Own Texture Maps

Something that is extremely important when making your own texture maps, is the size of the actual image you make. In order to determine what size you should make it, you need to know what the final rendered frame size of the animation is going to be.

Once you know what the final size is, then you can work out the size of your texture map as follows:

Take the maximum width that the texture map can appear on the screen (in pixels) and multiply it by two. Use this pixel size as the size of your image map, if you want to ensure that your textures do not become blurry or pixellated when viewed up close.

For example, the most common frame size when rendering for television is 720 pixels X 576 pixels (PAL D1), so if you were to make a texture image for an object that will be viewed right up close in the frame,

then the width of your texture map should not be less than 1440 pixels. I usually work with square images, as I almost always work with UV Unwrapping, so my images are generally at least 1440 x 1440 pixels. It's usually safe to leave the image at 72 Dpi, as this is the resolution that monitors and televisions display at. Of course, the drawback to this is that these kinds of file sizes for image maps do slow down the rendering process quite a bit, but on the other hand, you know they will hold close-up. So if you have a job that is going to end up on IMAX, you had better make sure your computer can handle 4000 x 4000 pixel images....

That's it for this round. I'll be putting together the chapters on the individual surface attributes, which will include picture examples, soon, and will be posting them as they are done. I hope that this was informative, and that it made sense. Please feel free to post comments or questions. Or, I have made a mistake; don't hesitate to let me know.

Part 2 – Standard Projection Techniques vs. The Painstaking Art of UV Mapping

- What on earth is UV Mapping?

"UVs" is a way of saying u,v texture coordinates (as opposed to the X, Y, and Z axis that you construct your meshes on), which are points, which define 1-by-1 positions within an image. These obviously connect to points in your 3D model, to position an image texture onto it's surface. Kind of like virtual "thumb tacks", what they do is pin an exact spot on an image that you wish to use to texture your model to a specific point on an object's surface. Between these points, your software will stretch the image smoothly. This is what is referred to as UV mapping.

So why use UV coordinates instead of standard projections? Well, once you have made your model, and are ready to texture it, the simplest way to apply your texture map is by using a standard planar, cylindrical, spherical or cubic projection. Here is a brief explanation of each of these, and how they are best used:

Planar is a method whereby an image is projected straight through the object along either the x, y, or z axis. This method is especially useful for items such as sheets of paper, posters, book covers, etc – in other words, flat objects. The problem with planar projections is that if the texture is projected along an uneven surface, or if the image reaches a side that curves away from the projection's plane, it results in unsightly lines such as in *Figure A*. When this happens, you then have to create lots of alpha-channel enabled images to cover up seams between adjacent planar projections and invariably ends up becoming a huge amount of annoying work. So never project a single image through an entire object if it has depth, like the box in the image, or if it has a very non-planar or irregular surface. Rather, as in the case of this box, create separate projections for the x and y axis as well, making sure that their edges will blend together properly. Alternatively, ensure the image to be used is tileable (seamless), and use a cubic projection (which I will discuss in a moment).

Most software has a bitmap fit/automatic sizing option that will stretch the image to fit the surface properly. Obviously, if your image is not the same shape as the surface onto which you are projecting it, this fit option will stretch it until it does. This usually doesn't look too fantastic; so ensure that you measure the size of your object before making your image map.

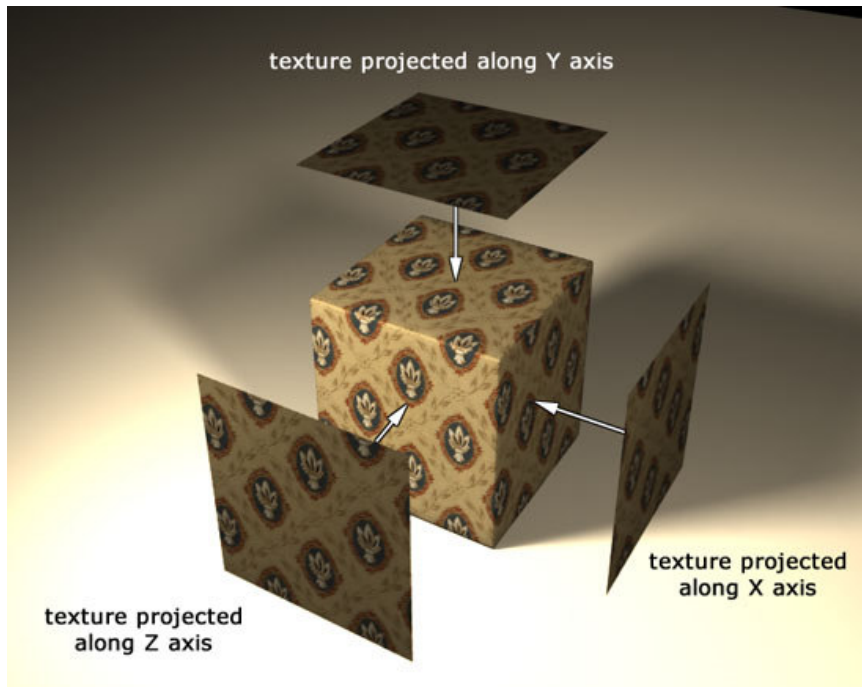


Figure A - Planar Projections

Cylindrical projection is pretty self-explanatory. Basically what happens is your image is wrapped in a cylindrical fashion around your model along one of its axis. This is really useful for one kind of object only – cylindrical objects. Please don't try and use this for anything else. When making an image that will be used for cylindrical projection, ensure that it's sides will meet correctly – in other words, where the two sides of the image wrap around and come together, make sure that there is no visible seam, as in *Figure B*.

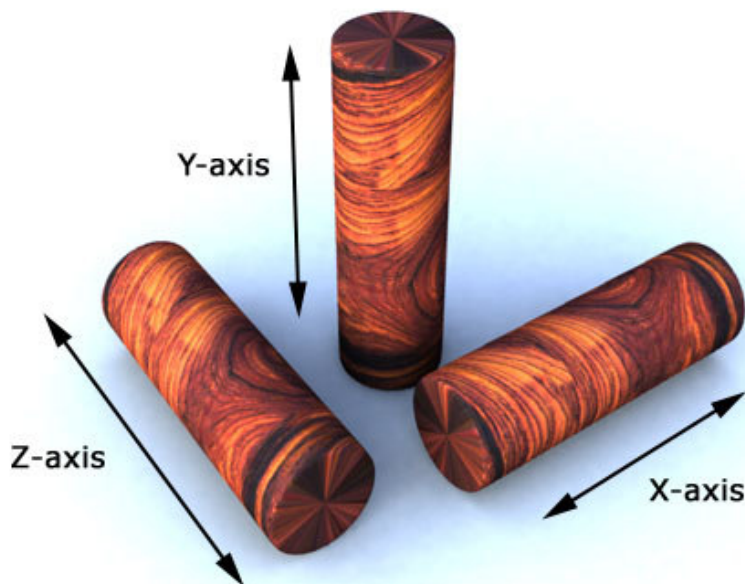




Figure B - Cylindrical Projection

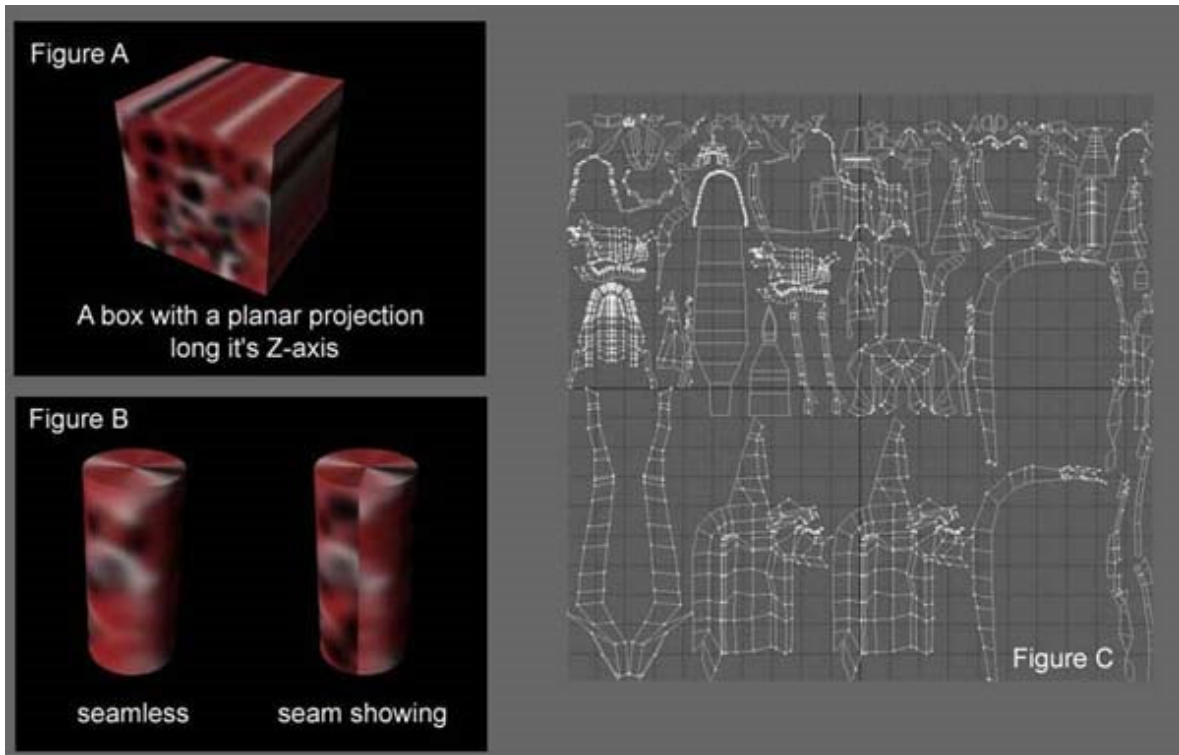
 A useful way of ensuring that the sides of an image will meet and merge properly is by using the Offset filter in Photoshop (listed under Filters/Other/Offset), and offsetting the image by however many pixels you choose, and using the Wrap Around option.

Spherical projection is when the image is stretched from one pole to the other along the axis you choose, and then wrapped around the sides from the back meridian. Make sense? Basically what I mean, is that, say for instance, you do a spherical projection along the Y axis, then the image goes straight from the top point down to the bottom point, and wraps itself around so that the two sides of the image meet along a straight line down the side of the sphere - this line is known as a meridian. Once again, only ever use spherical projection for spheres. Use it on any other kind of shape, and, well, it's not going to look very nice... Also, be sure to use the Offset filter (described above) to check that your two ends will meet properly along the meridian.

 Another useful way of checking whether a spherical map will work nicely is to use the Polar Coordinates filter (listed under Filters/Distort/Polar Coordinates) and use the Rectangular to Polar option. (However – ONLY USE THIS FILTER TO **CHECK** YOUR IMAGE. No NOT save your image that you want to use with this filter applied – it is not going to work!). This can give you an idea or whether there will be any seams when the image is wrapped around the sphere. This method is of course extremely popular for newbies to map their images onto the planets they make for those sci-fi scenes that just about everyone makes when they are starting out in 3D. Come on, admit it, you have all made sci-fi scenes, haven't you? Use the methods I've described here to ensure that your planet doesn't have any unsightly seams tearing across its surface.

Cubic projection just repeats a single image on each side of a box model it is applied to. Cubic mapping is basically a planar projection from 6 sides. Once again, ensure that the edges of the image will not form seams. Cubic projection is really only limited to perfectly square models, because if you try and use it on a rectangular shaped box, it will stretch the image on the long sides, and squash it on the shorter sides. Which looks pretty awful.

So, back to the original question – why use UV mapping instead of these options? Well, as you have seen, these projection methods are very limited. It's pretty obvious that they are not going to suffice for extremely complex models. But here is something bizarre – these projection techniques are actually, technically speaking, more accurate than UV mapping. This is because texture images will be more accurately mapped using standard projections, which have some exact, continuous values over the entire surface, whereas a UV map has accurate samples of the projection only at specific points (where the polygons join, basically), between which it then uses a linear interpolation on the surface between those points. For non-English speaking people, *interpolation* is, basically speaking, an estimation of values which go together to form a continuous series – this being a series of colour/tonal values being applied to your model, and the interpolation being the application of the parts of the image, which are not "tacked" down at those specific points, to the areas between them. The cool thing about this, however, is that once you have applied to UV coordinates to the model, pulling these points on your UV map around will pull the image with it. Pretty useful, hey?



○ UV Unwrapping

Once your model is complete (and I cannot stress enough how important it is to only ever start texturing your model once it is completely finished. **Never** begin texturing until your modelling is 100% finished. This just makes the entire process run much smoother), you are ready to decide whether your model requires UV mapping or not. And seeing as this chapter is about UV mapping, I'm going to assume your model does need it.

First, you need to decide what method of unwrapping you are going to use (unwrapping being the term most commonly used for the actual process whereby the X, Y and Z information is translated into the flat UV template). Yes, I know it can be so tiresome that everything has to have so many options, but it would all be pretty boring and not much fun at all if we didn't have all these different methods, now would it?

Now, seeing as I personally am a Lightwave user, I am only really familiar with Lightwave's unwrapping techniques. I am sure, however, that these different methods are common to all the major 3D applications, but if there are any other ones that you may use in your software, be sure to post it here too.

So, here are the different UV unwrapping options:

Planar Once again, we have a planar option. Yes, it's basically the same as before. The resulting UV coordinates are basically a flattened out straight-on projection through the surface along the desired axis. One thing to note, however, is that UV templates are always perfectly square (remember what I said about 1-by-1 image proportions?), so what often happens is that your resulting UV map can look somewhat squashed. Don't worry about that, you can change it by polling your UV points with the UV map itself without actually altering the actual geometry of the object. But more about editing UV's later on. I

PHOTOREALISTIC TEXTURING FOR DUMMIES

personally use the Planar unwrapping more than any other method, as it generally produces the simplest maps with which to work.

Cylindrical Yes, it's the cylindrical option again. And yes it's also basically the same as before. One thing to watch out for, is that if your cylinder object has a top that you wish to include in the unwrap, be sure to unwrap along the X or Z axis only, as unwrapping along the Y axis will leave you with a completely flattened top.

Spherical Ok, you guessed it. It's also the same as before. You are probably wondering why you should bother using UV unwrapping if what they do is basically the same as the standard projection types, but as I said, the power behind UV's is in the way that they "stick" to the points of your model and can be edited without affecting your models geometry.

Atlas This method of unwrapping may go by a different name in other applications, or may not be an option at all in some programs. Basically, Atlas unwrapping produces a UV template that translates the surface information into a UV map that represents the models polygons in a projection whereby once it is painted onto, will produce an image that will remain constantly perpendicular to the face normals of the surface polygons. Simply speaking, it is like taking a ball of paper that you have bunched up into an extremely irregular ball, and flattening it out again. The problem with Atlas unwrapping is that the resulting map is often a terrifying and confusing mess of disjointed polygons all over the show, as in *Figure C*. This kind of projection, although ideally useful, often ends up needing a lot of editing to get it into a state from which to work. I try and avoid it.

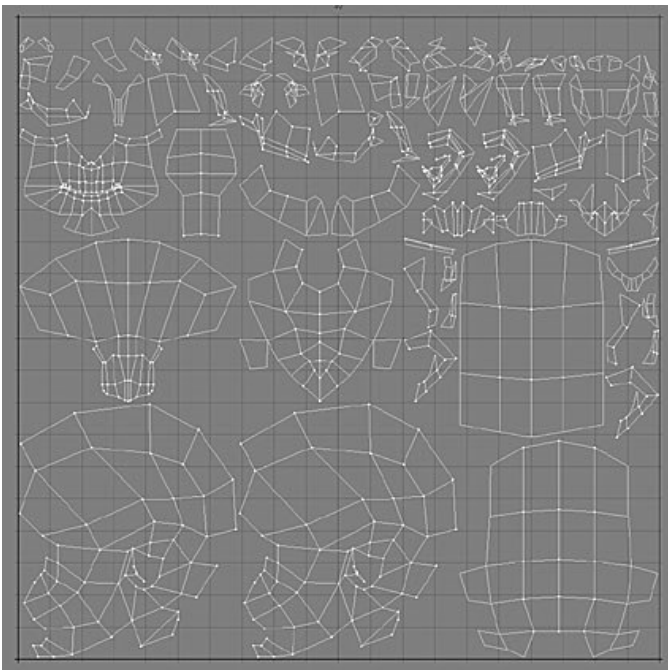



Figure C - Atlas Unwrapping. This is going to take loads of editing before you can export it to Photoshop and begin painting!

As I said before, I generally tend to use planar unwrapping the most. The problem occurring from this is that obviously when texturing, say for example, a character's body, multiple planar unwraps are going to be required (for instance, just the upper part of my Anubis character has 11 different planar UV maps applied to it), and because I make these all in separate files, what often happens is I end up with visible seams where the polygons which are using the different UV maps join. One way of covering them up is to make seam

images, which blend the different UV mapped surfaces into each other. There is another, easier method of avoiding this: When doing multiple unwraps for a single surface (as is the norm), choose a base colour that will be used as the base for all the different UV map images. Then, when painting onto the individual maps, just ensure that the detail you add lies a couple of pixels *within* the seam, that way, where all the UV maps meet, there is only that base colour between them so the seams will not be visible. The same goes for bump, spec, etc maps – make sure that no detail "breaks" the borders of the polygons within the UV map, that would then become noticeable when they do not continue on the polygons using different UV maps surrounding them.

 *MAYA: To make UV's for poly objects, first you must make sure you have the modeling menuset enabled, rather than animaton/dynamics/rendering (I'm sure you knew this already). Then you select some faces you want to lay out, then go to the menu "Edit Polygons" -> "Texture" and for an example pick the "Planar mapping". Now you open the "UV Texture Editor" found in the "Window" menu and you should now see the UV's laid on to the grid, now you can move them as much as you want, add another faces to the same plane by selecting more faces and "Planar/Spherical/Cylindrical/... Mapping", then move 'em around the "UV Texture Editor" window and finally make a snapshot of it to have the UV's saved to a file.. Et voila!*

○ So what do you do now that you have the unwrapped UV map?

Firstly, if any editing of the UV map is required, then do so. Your 3D program will have a bunch of tools that you can use to edit the map, and as I said before, remember that altering this map is not going to affect your geometry in any way. Most UV unwraps will need some editing as fragmentation often occurs, resulting in polys that actually lie adjacent to each other being displayed on opposite ends of the template. Edit your map until you are satisfied that you can work well with it, and then you are ready to apply an image to it.


You two options now:

The first option you use if you have already created an image map that you now wish to apply to the UV map. Change one of your viewports to display the UV map. Now, just stick the image you have made into the background behind the UV map, and pull the points of the map until they sit in the correct positions in correlation to the image. This is a rather bizarre way of doing it though; I must say that I personally have never used this method.

The second method is to export this template to Photoshop (or whatever painting program you use). There are two ways to do this – some Unwrap plugins will create an image for you, which you can then open up in Photoshop, or you can just get a screengrab (using the PrintScr button next to your Scroll Lock key), go to Photoshop, go to New Image (the image size will already be there, defaulted to the resolution your OS is running in), and Paste. Then just trim the square UV template (make sure you do this!! Lots of people have come to me asking why their UV maps aren't working, and it often ends up that they didn't trim the image down to the square size) and you are ready to begin painting.

As I discussed in Part 1, image size is important when making texture maps. To refresh your memory: In order to determine what size you should make it, you need to know what the final rendered frame size of the animation is going to be.

Once you know what the final size is, then you can work out the size of your texture map as follows:

 Take the maximum width that the texture map can appear on the screen (in pixels) and multiply it by two. Use this pixel size as the size of your image map, if you want to ensure that your textures do not become blurry or pixellated when viewed up close. For example, the most common frame size when rendering for television is 720 pixels X 576 pixels (PAL D1), so if you were to make a texture image for an object that will be viewed right up close in the frame, then the width of your texture map should not be less than 1440 pixels. I usually work with square images, as I almost always work with UV Unwrapping, so my images are generally at least 1440 x 1440 pixels. It's usually safe to leave the image at 72 Dpi, as this is the resolution that monitors and televisions display at. Of course, the drawback to this is that these kinds of file sizes for image maps do slow down the rendering process quite a bit, but on the other hand, you know they will hold close-up. So if you have a job that is going to end up on IMAX, you had better make sure your computer can handle 4000 x 4000 pixel images...

Part 3 – How To Make Great Colour Maps

○ Introduction

Now that you (hopefully) understand what texture projections and UV unwrapping options you have available, and you know how to use them correctly, you are ready to begin making the actual image maps to be used to make up the textures for your model.

Now, just as careful planning is always required before you begin modelling, so it is also required for the texturing process.

Being primarily a texture artist myself, when we are just beginning a project at work, I usually immediately begin researching possible ideas for the textures which are going to be used on the final models. It is very important to plan the look you want your models to have in the end, and gathering as many possible references as you can is a good start.

The internet is obviously extremely useful for scrounging for pictures of bark, metal, trees, water, Velcro, skin... whatever different surfaces you are going to need to make for your models. The problem, however, with looking on the net is that most pictures you come across are very low resolution or just plain bad quality. However, it can still be a good place to start. Once you have spotted a certain metal or bizarre animal skin that you like the look of, go down to your local library and see if you can't find some good large images of the substances you want to recreate. Also, if you have access to a decent camera, go running around and look to see if you can't find some good references in the world around you. But don't do this to the detriment of your health – many times I have wished that I could climb up to the rusty old watertanks perched on the seventh storey roof of the building where I work to get some close up shots of all that lovely rust... but alas, I fear a fall from that height is likely to leave me with my brains leaking onto the parking lot below and my limbs all mangled. Which is really going to hinder my texturing abilities...

Now, as you all know, I love painting my own textures. And because that is what these workshops are really about, I'm going to be focusing on that.

Yes, photographs are essential texturing references, but I very rarely use any portions of photos in my actual image maps. This is because most photos require a lot of editing to remove lighting from them, and secondly because painting your own textures from scratch is much more fun, and more rewarding.

Right, so we have established that before you start, you must have a really good idea of what you want your textures to look like. Have all your reference images close at hand, or, even better, on your computer so that you can refer to them as needed.

Now I guess the most obvious place to start is to make your colour images. This is because your colour images set the tone of your model and give a very clear indication to you (and your clients) of what direction you are going in, and what sort of "feel" your model is going to have.

○ How To Make Really Great Colour Maps

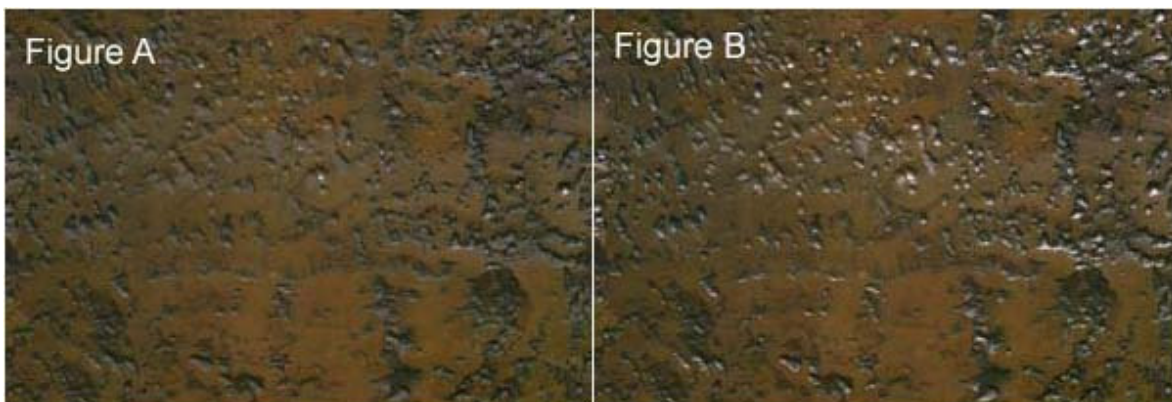
Firstly, a good reason to have all your reference images on your computer is so that you can take them into Photoshop and use that handy little eyedropper tool to sample colours out of them. That way you can

PHOTOREALISTIC TEXTURING FOR DUMMIES

ensure that your airbrushed images contain all the correct tones for the surface you are making. However, before going into too much depth about the actual painting process, just a quick word on photographs:

If you want to use a photo, or a portion of a photo, in your image map, it is **extremely** important to ensure that you first remove all trace of light from them. For instance, look at *Figure B*. Now that image will not work very well as a colour map because there is lighting in it. When looking at the actual colour of that surface, the areas where there are highlights are not actually white, and the areas where there are shadows are not actually dark like that. With a bit of editing, you can remove that light information so that it becomes more like *Figure A*. There are a number of ways that you can do this, depending on the image, and how the lighting needs to be removed. Sometimes just some simple Hue / Saturation adjustments, or Levels adjustments will do the trick, other times you will have to actually go in and manually paint it out. The same photo technique applies no matter what channel the image is going to be used in. Lots of people would just use this image as a colour map, as well as a bump map, even though, if you look at that image, you will see that it will actually not work as a bump map at all.

📌 *Another EXTREMELY important note: Never ever ever ever use any of Photoshop's texture filters on your image maps. Just don't even think about it. This causes exactly the same problem as photos that contain lighting information. I know that the little texturizer filter can seem tempting, but DON'T use it.*



Okay now you are ready to start. I am going to use the colour map that I am using on my Anubis's chest and torso front as an example here. Firstly, I load up my UV Map that I made for the front part of his upper body, see *Figure C*. Generally I find the initial colours of the screengrab a bit disconcerting because the grey in it tends to make you lose your cursor while painting on it, so I usually darken it a bit with Levels.

Then I decide on a basic colour and paint over the entire thing with that colour. Once I am happy with that base colour, I begin adding detail bit by bit.

The Airbrush tool and the Paintbrush tool can generally do the job here, but I personally also use the Dodge and Burn tools just to create different values of the base colour and well as variations of the tones that I am adding. It is important to start on a new layer whenever you add a major detail – such as the cuts that he has on his body in *Figure D*. That way, if you don't like the way it turns out, you can remove it without a long cover-up operation on the original layer.

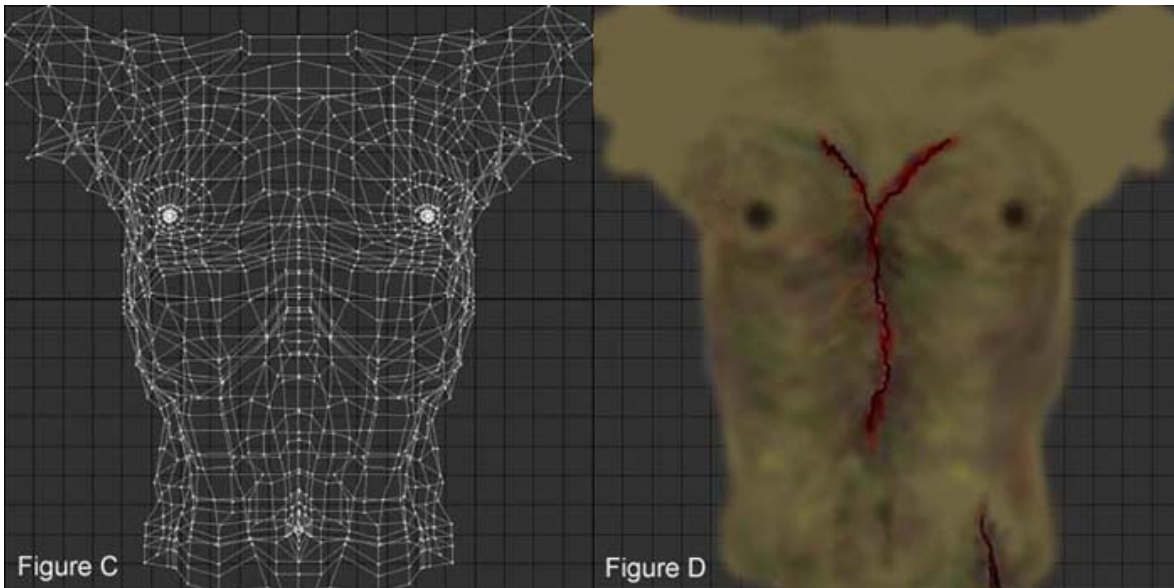
PHOTOREALISTIC TEXTURING FOR DUMMIES

An extremely strong attention to detail is, of course, always required for texturing. Even if you don't think anyone will ever notice the detail you are adding, don't let that stop you from putting it there. The chances are that if you don't add it, the final texture will seem to be lacking something...

Have a look at the areas surrounding the cuts. I've added rough bruising around the wounds, and because he is supposed to look gross and dead, I've used a rather revolting selection of browns and greens and yellows.

An important thing to keep in mind is the condition of the surface, and how to ensure that your colour map is going to work correctly with your bump and specular maps (as these two maps are often used to define the textures physical condition). Say for instance that you have to texture a piece of metal that has been painted, but is old and scratched. The areas where is has been scratched which you will be making in your bump map, must have the paint scratched off. I know this is pretty obvious, but I'm just pointing it out for beginners. So, throughout the process of making all the different image maps, always keep in mind the relationships between the different surface channel.

✎ I find the best way to do this is to do all the different images for a single UV map in the same Photoshop file, each on their own set of layers. That way I can copy, for instance, the wounds on his body (which I did on their own layer, seeing as they were a major detail) to the bump layer to ensure that the bump map has the cut positioned correctly. Unfortunately this method does result in rather large Photoshop files which can really bog down your system as well as use up a lot of hard drive space. It's really up to your own personal preference, but this really does make things a lot smoother and easier to manage.



Well there really isn't much more that I can say on colour maps, as they are actually the most straightforward of all the image maps that you will be making for your model. I'd just like to say once again, that you must always ensure that you have all the correct and realistic details that your texture needs. For instance, if you are texturing pipes, which are inside sewers, they are going to be really dirty and disgusting. If you are texturing a chair, which goes inside a dentist's surgery, make it pretty clean. If you are texturing canal walls, add drips and water damage and streaks down the sides where the walls have been weathered over the years. Think about where your object is and how the world affects it from day to day. Even the cleanliest of environments gather grime - if you look around a sterile biochemical laboratory, you are bound to find dust caked in the corners, and faint marks from people's shoes on the floor. And obviously it is the colour map

PHOTOREALISTIC TEXTURING FOR DUMMIES

that tells people how clean or dirty your objects are. Outside environments are rife with damage and dirt and streaks from rain. Make sure you incorporate these into your image maps, or your renders will end up looking fake and unbelievable. All too often people texture their models to look far too clean, resulting in a giveaway cg look - a classic example being the visual effects in Star Trek. Everything in Star Trek looks far too perfect and new, and you can't help but ask yourself *just how do they manage to get that huge spaceship through an intergalactic car wash everyday?* A gigantic hulking spaceship is most definitely going to pick up a lot of damage from flying around asteroid fields and going through rips in the fabric of time. Just look around you, notice how nothing really has a perfect solid continuous colour across its surface. Notice how and where dust and grime gathers on things. Now put these details into your colour maps! Have a look at this render I did a couple of years ago in Max - see how I added weathering damage to the walls. Without those streaks and splotches that would obviously come mostly from rain and damp, this building would not be believable as a 13-century monastery forge. Unfortunately, I do not have a huge selection of good colour maps handy to put here to show you, but I will be looking for some over the next day or two to add to this thread. Have a look at this month's challenge entries - people like Rob Pauza and OZ have made very cool colour maps, which they posted into their threads.

Once again, everybody please feel free to add all your own tips and suggestions to this thread. Post images of your own colour maps, and share your own ideas. If you have any questions, go ahead and ask. Lets hope that this chapter arouses as much participation in everyone as the last one did 😊



leigh@atomic-vfx.com

Part 4 – The Diffusion Confusion

- **What? Diffusion isn't colour??!**

Yes, I too was shocked when I discovered that diffusion is not actually colour – all those years of using Max, sticking my colour image in the diffuse channel, just as my college lecturer had taught (and my Max textbook explained). But... but... if it's not colour, then what is it?? Let me just interject the subject at hand to warn you here that this article will be pretty dull. Unfortunately, it is necessary knowledge for creating photorealistic surfaces, so bear with me. There will be more exciting chapters to come, I promise. No pain, no gain, eh?

So, as I was saying, if diffusion isn't colour, then what is it?

Well, it has got *something* to do with colour, but it is not the actual colour itself. In reality, diffusion is actually the extremely essential attribute of an object's surface that scatters light – so basically it's the surface attribute that determines how much light is reflected, as well as absorbed, by the surface – ***it determines how much of the surface's colour we can see***. And you certainly don't need a degree in physics to understand that our eyes are able to see things purely because of light reflecting off everything in the world around us. Simply put, those rays enter our beady little eyeballs and the information is processed in our brains, and we see things. Which is why we can't see too well in the dark – there is very little light bouncing around. And what little we can see, is kind of fuzzy, due to the poor quality of what little light there is available.

To drift slightly from the subject at hand, I'd just like to state here why global illumination (usually just referred to as GI) is so important. Now, for those of you that don't know much about global illumination and radiosity – here is a complicated explanation: Radiosity, technically, is defined as the amount of radiation leaving a surface per unit time per unit area.

Which, basically, to us mere animators, means that radiosity is the effect of light bouncing around – in other words, it is the indirect light that is distributed between all the objects in any environment. To illustrate this an easy example would be to set up a scene of a plain, empty room with a table in it and a light (lets say, for instance, a point/omni light) above the table. Now, without radiosity, if you were to render this scene, the underside of the table would be pitch dark. Now, in real life, this is not so. Set up a table in an empty room and hang a bare light bulb above the table. You don't even actually have to do this to know that you would definitely be able to see under the table. Why is that?

Well, that's because the light from the bulb would bounce off the walls, onto the floor, and then bounce back up off the floor and illuminate the underside of the table. Actually, it's interesting to note here that the majority of light around you actually comes from indirect light, and not the source itself.

Radiosity is perhaps the most critical element of photorealistic rendering, as it gives an ambient radiance to your scene without making it look flat. And because it is a real world effect, it's essential to include it in your rendering.

Up until very recently, most animators had to fake radiosity with complicated lighting setups consisting of anything up to 100 to 200 carefully placed point (omni) lights in a scene, usually set up similar to a concert lighting rig, in a cone like shape. Obviously this isn't a great solution though, as we all know that too many

omni lights can sometimes make things look a little flat. The same thing happens, but to an even worse degree, by using the ambient lighting option that is included in most software. If you want to do your light setup properly – DON'T USE THE AMBIENT LIGHT. Switch it off completely. In most programs, ambient lighting is on as a default, so find out where yours is and SWITCH IT OFF!! In Lightwave you'll find it in the Global Illumination panel, and in Max, it is in the environment panel.

However, as times have changed and software programming has become more intense, these days most programs are beginning to include global illumination options. And for those that don't, there are many third-party rendering solutions to choose from. So you have no excuse not to use it.

I'm mentioning this because without radiosity, diffused surfaces will not work properly. Instead, they will just appear too dark. But back to the actual subject at hand...

So where were we? Oh yes, I was explaining that diffusion determines how much of a surface's colour we can see, because it controls how much light is reflected off the surface and how much is absorbed. And although this may sound similar to specular reflection – let me quickly just clarify the difference between those two – as I said, diffusion controls how the *light's rays* leave the surface, whereas specular reflection controls how much the surface reflects the actual *light source* itself. It would be safe to say, that diffusion, in practice, is simply the opposite of specular reflection, in much the same way that opacity is the opposite of transparency. Diffusion refers to the scattering of light, whereas specular reflection is to do with its reflection.

So, in essence, by diffusing an object, you control the amount of colour that is reflected back off the surface by the light. This is completely different to simply darkening the surface of the object itself. If you were to darken the actual image used as a colour map, you would only see a change in colour, but not colour depth.

Colour depth is created by scattering light across an object's surface. Take a look at human skin and you'll notice that it has a density. The colour isn't a simple continuous shade but rather many similar shades, created by scattered light. This quality cannot be made by a colour map alone, as a colour map cannot give a surface the richness that a diffuse map can. Of course, we have to bear in mind at this point that obviously *translucency* is another critical attribute with regards to scattered light, as it determines how the light *passes through* a surface and, together with sub-surface scattering, is scattered around *inside* the surface. But I'll be discussing translucency in depth in a later chapter.

Diffusion can also be used with lighting itself. Those of you who have spent any time on film sets or in photography studios will know that it is necessary to put milky, semi-opaque plastic sheets in front of the lights. These are basically diffusion screens. They change the quality of the light, so as to prevent glare on the subjects they are lighting. This is pretty much what we are doing when we diffuse objects. We are preventing over-saturation of the colour of their surfaces.

Right, so now that we understand what diffusion is, how exactly do we use it?

○ Putting Diffusion Into Practice

How do you make a diffusion map? Well, for starters, as with most other image maps, a diffusion map's effect is created using varying shades of grey. And, keeping in mind that when working with grey scale images for surface attributes – where white always represents a positive value of the attribute, while black represents no effect at all – the lighter the shade of grey, the more diffused the surface becomes. And

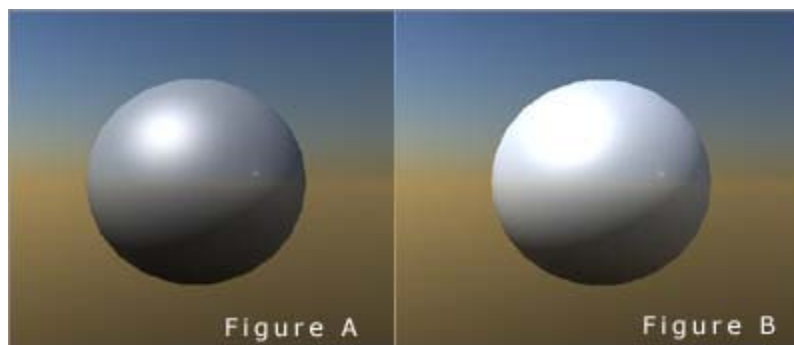
PHOTOREALISTIC TEXTURING FOR DUMMIES

obviously, the more diffused the surface is, the more light is reflected off it, carrying colour information into our eyes, and the more colour we will see.

Knowing this, you are now faced with the tricky task of determining the diffusion amount for a surface. As a general rule, **absolutely nothing** has a diffuse value of 100%. In fact, most things have a diffusion amount of 80% and lower.


Because nothing has a diffusion of 100%, leaving it at that amount will result in your surfaces interacting unrealistically with the light. This will basically result in your objects looking over-saturated, because the actual colour of your object becomes over-saturated by the light. Obviously, without light, diffusion doesn't actually make a difference, but the moment you put any light into your scene to illuminate your object, that light is going to hit your objects surface, and, depending on how shiny you've made the surface, it's going to create a hotspot of light. Now, if the object has 100% diffusion, that hotspot is going to multiply the colour of the surface, as well as adding it's own colour to the surface. This is obviously going to result in an unsightly over-saturated spot.

Take a look at Figures A and B. Figure A has a diffusion of 65%, whereas Figure B has 100% diffusion. Notice how over-saturated that light spot is on the second sphere. Yuck.




Okay, so how do we make the diffuse map to go with our surface? Well, here is something to consider – a good general rule is that ***the more reflective the surface is, the lower its diffuse amount***. Why is this? Because the more reflective an object is, the less colour it has of its own – the “colours” you see in it are merely reflections of its surroundings. And the more they reflect of their environment, the less colour they have of their own.

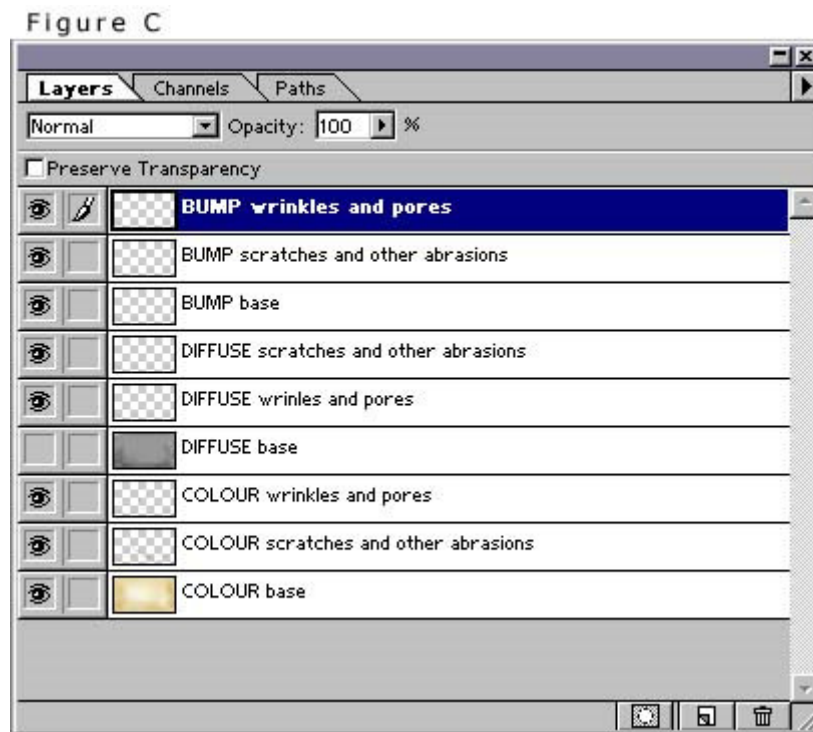
Metals, for example, generally have lower diffusion amounts, as they are usually quite reflective, whereas bark on a tree would have a higher diffusion (unless it's wet, in which case, it appears more reflective, and therefore would have a lower diffusion), as it possesses more colour of its own. Remember, the diffuse amount controls how much of the colour is seen, so if the surface doesn't have much colour of its own, it obviously doesn't scatter light.

 *Tip: Bearing this in mind, it therefore makes sense that when making an object that reflects 100% (like a mirror), you would make the base colour of it black (in other words, it has no colour). And because it has absolutely no colour, its diffusion amount would also be zero.*

Another thing to bear in mind is the topography of the surface. Any cracks, in the actual geometry of object, are going to have a slightly lower diffusion, as light gets trapped in cracks, causing less of the surfaces own

colour to show through (in other words, because the light gets trapped there, it doesn't bounce back into our eyes, carrying that colour information which would allow us to see that colour). This obviously means that a diffuse map has to be carefully made, as it has to include these sorts of details. Simply giving a surface a global value is not going to suffice. Remember then, that when making the diffuse map, you must keep in mind what the bump map is going to look like as well, as this will affect the diffusion, but obviously to a lesser extent than any cracks, holes, etc in the object's geometry itself.

 *Tip: This is why I always do all the different maps for any portion of a surface in the same Photoshop file. I usually create a “base” layer for each attribute, and then copy any details, such as abrasions, scratches, blemishes, etc onto new separate layers so that I can copy each detail separately and add it on top of the next attributes “base” layer. Every detail of each different attribute affects all the others in some way, so your method of affecting them universally is vital. Look at Figure C for an example of my Layers Palette in a skin texture image, where I have worked on 3 different attributes – Colour, Diffuse and Bump. Using this method ensures that all your different little details remain in the correct positions for each attribute.*



Now, remember how I explained earlier how it is that areas, which are shinier and more reflective, have lower diffusion values? Well, it would therefore stand to reason that the information from your specularly/reflection maps is important to include in the diffuse map. You can do this by inverting your reflection map (thereby converting the areas that were shinier and therefore lighter grey in the reflection map to the opposite – darker shades of grey and therefore less diffused) and adding it to your desaturated colour map. Blending these two together makes a great base for your diffuse map. However, the information of the colour map is more important than that of the reflection map (I'll explain why later), so make sure that you lower the transparency of the ref map before blending the two together, that way ensuring that the desaturated colour values remain predominant.

But wait a minute! If you think logically, carrying on with this process in this fashion will lead to a certain problem. Gussed what it is yet? Okay, this is a tricky one to fully explain...

We know that, in extremely simplified terms and with regards to its initial immediate effect, lowering the overall diffusion amount and adding a diffusion map *darkens the surface*.


So basically, if you do your diffuse map after you've completed your colour map, then surely when you slap your colour and diffuse maps onto your model and render it, the change in diffusion is going to result in an unwanted change in the colour value.

To put it more simply – the colours you painted onto your colour map are the colours you want your actual model to appear when rendered, aren't they? Now, knowing what diffusion does, it's obvious that when you lower your overall diffusion amount and add your diffuse map, the actual colours you painted onto your colour map are going to appear too dark. And that's NOT what you want, is it? No. So how do you compensate for this??

Well, it's actually pretty easy. It just takes a bit of forethought, some cunning, and using your brain cells a bit.

Okay, as I mentioned earlier, different substances have different base diffuse values – for example, human skin has a basic diffuse value of about 70-80%, wood is about 70-75%, and a reflective metal like chrome has a value of about 15%. So, depending on what your surface is, you would have given it a certain overall value. But how do you decide on what that overall diffusion value is???

Just take a look at its overall shininess and reflectiveness. These properties are easier to reproduce by eye than diffusion is. So, when you initially begin to make your surface, decide on how shiny and reflective it is first. This is pretty easy for anyone to do, as we all know how shiny different surfaces in real life are.

 Here is a very important tip: When creating ANY surface, make sure you have plenty of reference handy. Say for instance I'm given an object to texture, and it's a giraffe's snout. Before I start making ANY image maps or assigning any values to any of the surfaces attributes, I'd ensure that I had at least 10 different pictures of a giraffe's snout to refer to. That way I can see exactly how it's surface reacts to light. I cannot stress enough just how important reference pictures are. By referring to reference images, you can tell exactly what it's colour is, how shiny it is, how reflective it is, what sort of texture (bump) it has – basically I can see exactly how it appears in real life. This will thereby ensure that my own cg surface turns out perfectly true to its real life counterpart.

Okay, so we've decided on our overall reflection amount.

Now, decide on an overall diffusion value depending on what the overall reflection amounts are – keeping in mind that the higher those values, the lower the diffusion will be (or, on the other hand, the lower the reflection, the higher the diffuse). A rough way of determining this would be to use the leftover number from the total reflection value – in other words, if your overall reflection amount is 75%, then make your diffusion 25% (as that is the remainder of the subtraction of the reflection percentage from a possible total of 100%). And if your reflection is 15%, then make your diffusion value 85%, and so on (however, a little leeway is allowed here – sometimes you will find a bit of tweaking of a few percent here and there will be needed to get it looking right at the end). These rules aren't cast in stone, they are just guidelines. Are you following me? Good! Now keep this little equation in mind, as we use it similarly to determine the colour compensation.

Right, go back to your colour map. Okay, now say for example I'm making a human face. And I've decided that my overall diffusion value is 75%. Well, in order to compensate for this diffusion value, we are going to need to lighten the colour map by 25%. We lighten it by 25% as this is the difference between 100% and

PHOTOREALISTIC TEXTURING FOR DUMMIES

75% (of course, to put it even more simply, you could say that you would need to lighten the colour map by the same amount as your overall specular/ reflection value, but I'm referring to the diffuse amount, as diffusion is the subject at hand).

Knowing this amount by which it needs to be lightened, how exactly do you lighten your colour map? Well, in Photoshop there are a couple ways of doing this. You can use the Brightness/Contrast tool, and just drag the slider until it looks like it's a certain percentage along the line. You could also do the same using Levels. However, this simplest way of doing it is by using Curves. Just click on the graph line and adjust it until the percentage displayed beneath the graph is correct.

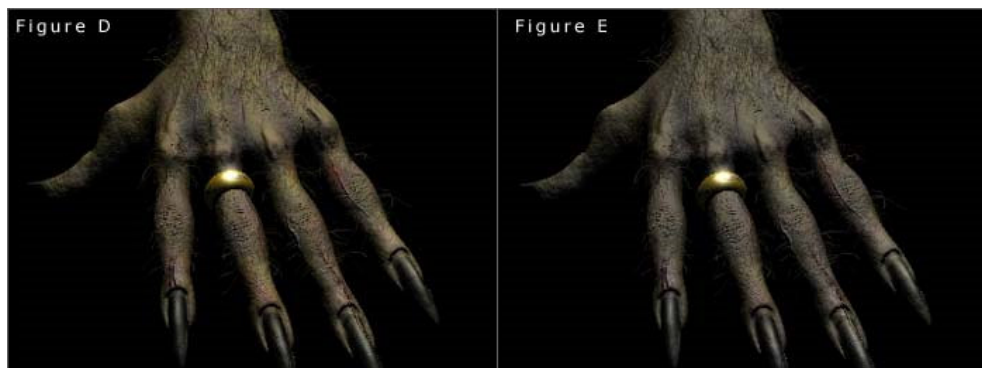
However, as I said previously, a single global value often does not suffice. Bear in mind that this compensation only correlates to the overall diffuse amount, whereas the actual map that you use in your diffuse channel has varying strengths of diffusion, obviously determined by the varying shades of grey within it. That is why it is so important to use the grayscale values from one another. Wow, painting image maps is tricky business, eh?

Okay, before you go and experiment with diffusion, let me just let you know that this is, in fact, an extremely subtle effect. So don't go expecting huge, radical differences in your renders. When you have made your diffusion maps correctly, you'll notice that the difference is very slight. However, subtle as it may be, it is nevertheless an **extremely important** effect.

Take a look at Figure D. This hand has no diffusion, whereas Figure E does. You'll see that the difference between the two is only very slight. The colours of the hand in Figure D are over-saturated – the skin of this hand is supposed to be dead, and even though I painted the colour map correctly, the fact that it has 100% diffusion results in the colours becoming over-saturated. Which is not what I wanted. Whereas, with Figure E, I've added a diffuse map, and all my overall values are set correctly, according to the method I've described. As you can, this has resulted in a more desired saturation – the skin now appears the way I want it to.

I know that a lot of you may be looking at this, and thinking “why bother?” Well, if you want to achieve photorealism, and produce high quality visual effects, these are things that are essential. However subtle and useless this effect may seem, trust me - it IS essential. A character whose skin has been textured and left with 100% diffusion is going to get very nasty looking saturated spots all over it when it gets lit, and the only way to avoid that is to use diffusion correctly. It's all part of the bigger process.

Well, that's pretty much all I can really say about diffusion. I know it's quite a boring one, but as I've said, I wouldn't have gone to all the trouble of explaining this if it wasn't extremely important. As usual, if you have any questions or anything to contribute, please feel free to do so...



Part 5 – Specularity, Glossiness And Reflection

- The Need For Some Shine

Okay, before the debate begins to rage as to whether or not to actually use specularity, or just stick to reflection mapping, let's first assess the necessity for shininess in the first place.

Basically, without shininess, an objects surface appears flat, and does not really "react" to the light shining on it (of course, the fact that it has a colour means that is it *reacting* in some way to the light, but I'm talking more in terms of visible "highlights" or "hotspots" here).



Figure A – without some kind of shininess this metal and the leather would appear extremely dull and flat.

Highlights on a surface give us an idea of how the surface feels – whether it is smooth or coarse (not in terms of the objects topography, which is generally defined by the bump map), whether it is hard or soft, dry or wet, old or new, greasy or slimy, and so on and so forth.

Another extremely important thing detail it relays to us is the objects everyday interaction with the world – by altering and breaking up the reflection of light on it's surface, we can get clues as to how the object is handled by people, or how it is used in the world. In other words, it shows us how the world and its inhabitants have left their mark on it, so to speak. For instance, a wineglass is never really 100% squeaky clean – look closely at it, and you will see oily fingerprints, faint grime from general handling, smears from the last time it was washed, and an entire host of other greasy smudges, abrasions and dusty marks. These all affect the shininess of the object by lessening them. On the other hand, interactions can increase the shininess – for instance, an apple that has just been polished, will have brighter, shinier spots where it has been polished harder.

I think that we can all agree that the surface definitely does need some attribute to show the way in which these sort of things affect and wear an objects surface over time.

Finally, the most obvious need for this shininess is to show that the way in which local light-sources are affecting it. Of course, this is where the debate begins – ***does one use specularity, or reflection?***

To add more substance to that question, here is a little refresher course on light:

As we all know, we are able to see things because of the way that light is reflected off things around us – the reflected rays shoot into our eyes, bounce around a complex setup of lenses and things inside our eyeballs, thus enabling us to see this wondrous world around us. So, basically, when setting up textures in CG, we have to bear in mind that we are dealing with *reflection of light*, and therefore creating the "instructions" for how the light should be reflected off the objects surface by creating texture maps, right? Every detail we see on a surface is there for us to see because light has bounced off it and into our eyes, showing us what the surface looks like.

To sum it all up – shininess is the reflection of light – the stronger the reflection, the shinier the object.

So, how do we set that up? This brings us back to the question – do we set this up using specularity? Or reflection??

To fully illustrate both sides of this question, let's look a little more closely at these two options...

○ Specularity – The Big Fake

Okay, so what *exactly* is specularity?? In reality, the effect known as specularity in CG is actually called **specular reflection**. A dictionary definition would describe specular as a "mirror-like quality". However, unlike it's real-life equivalent, the effect of specularity in it's 3D package incarnation, as we are familiar with, is actually quite different from the reflection option in any surface editor. Specularity is basically a way of **faking** the reflection of light on the objects surface. Let me explain.

Technically speaking, when you see a highlight on an objects surface, it is actually a *reflection* of all local light-sources. For instance, if you place a fairly shiny plastic cup onto a table in a room that is lit by a single light bulb, you will notice highlights all along the surface of the cup (and the table too, obviously, but we are concentrating on the cup in this particular example). Now, if you look really, really, *really* closely at these highlights, you will discover that in actual fact they are reflections of the light bulb itself. Obviously lots of surfaces don't have very tight, defined hotspots which are as clear as they would be on a plastic cup, but all that has happened is that the reflection has become more spread-out, a property which is controlled by the gloss amount, which will be discussed a little later on. So, having said that, you may be wondering why I said that specularity is fake. Specularity is fake because it doesn't actually *reflect* the light-source in the same way that the surface would in reality, instead it just gives the illusion that the surface is reflecting light. In other words, it shows highlights simply because there is light shining on it. It isn't actually *reflecting* anything as such. You could, for instance, shine a spotlight onto the object, and when it renders, you will see a round hotspot on the surface, not an actual reflection of the spotlight itself.

Basically, specularity gives you round hotspots, that you can break up a bit using specular maps.

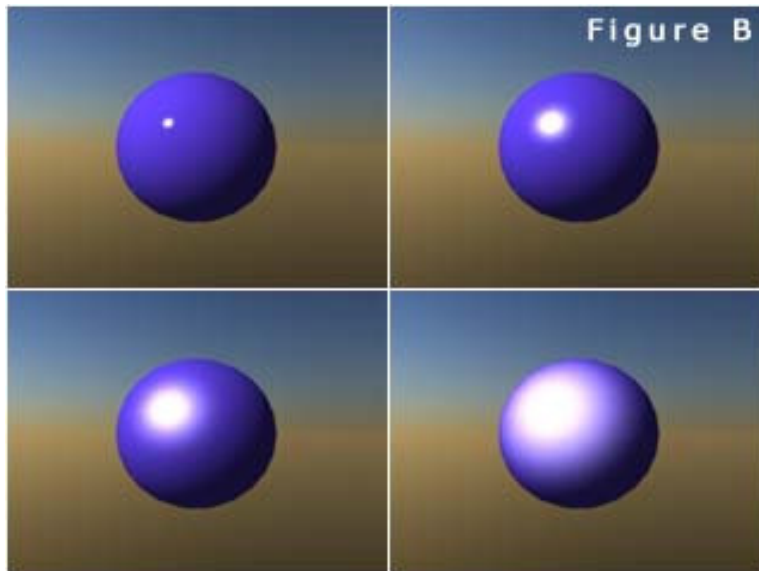


Figure B – Specularity creates round hotspots on your surfaces.

- **Reflection – The Real Deal**

Fake not good enough for you? Well, reflection is pretty self-explanatory. Using reflection on the surface will obviously allow the object to reflect its surroundings and local light-sources correctly. No need for any in-depth explanations here, as we are all aware of what reflection is and what it looks like.

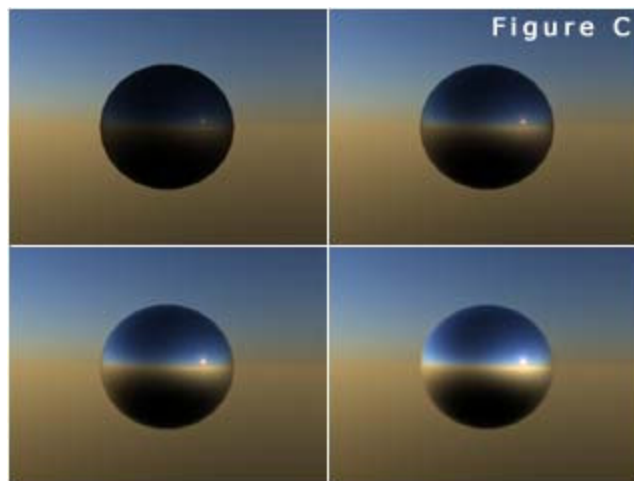


Figure C – Using reflection allows the surface to reflect its surroundings like a mirror.

- **The Big Showdown – Specularity VS Reflection**

Right, now that we understand what specularity is and what reflection is, we can look at the argument as to whether to employ one or both in a surface.

Why use specularity if it is fake?

A fair question. I guess there are 2 immediate answers to that question:

Firstly, **specularity renders faster than reflection**. Accurate reflections require complex raytracing, which takes a lot longer to render than specularity. In order to utilize reflection in most software, one has to activate a reflection option in the rendering settings, which adds time to the render in order to calculate the reflections. And we all hate waiting for renders.

Secondly, **reflection almost always makes objects begin to appear mirror-like**. To go back to our earlier example of a plastic cup, if I want to make a plastic cup that should be realistic, using reflection instead of specularity is more than likely going to make the cup look unnecessarily mirror-like. Perhaps plastic isn't the best example here, as it usually is a bit reflective as well as shiny, so let me give another example – wood. Wood that has no varnish on it, and is fairly dull, and really doesn't appear reflective at all, will nevertheless have a hotspot (however faint and spread-out it might be) on it if you shine a light directly onto it. Giving the wooden surface a certain amount of specularity will allow this hotspot to show on the surface without it reflecting like a mirror, which is what would happen were you to use reflection instead.

The same goes for cloth. Look at the clothes you are wearing – light is creating highlights on your clothes – these highlights are especially noticeable on folds in the fabric, where it is catching the light. Fabric, however, is most certainly not reflective in the sense of reflecting objects around it. So using specularity instead of reflection in this case is more feasible too.

All this is rather mind-boggling when you consider the fact that the reason that objects appear shiny in the first place *is* because they are, technically, reflecting light. So even fabric is technically reflecting light. So is dull wood. It's just that using reflection in CG tends to make these surfaces look too mirror-like.

On the other hand, if you want to recreate reality perfectly, then it would make sense to use reflection, as this is the physically correct method.

AAAARRRRRGGGHHH!!!!

So, which one do you use????

I guess this is the bit where I am going to tell you to use basically whatever you think looks good. This argument can go on and on, but in the end I always think that **if it looks good, use it**.

Yes, reflection IS the more realistic way of doing it, in terms of physics. That is easy enough to understand. It's just that, more often than not, the results of doing it like this tend to end up looking wrong.

What do I use? I tend to use a little bit of both. I have always used specularity, and I use reflection to enhance my specularity. However, I have seen some great work done without any specularity whatsoever. I guess it really comes down to individual methods of doing things.

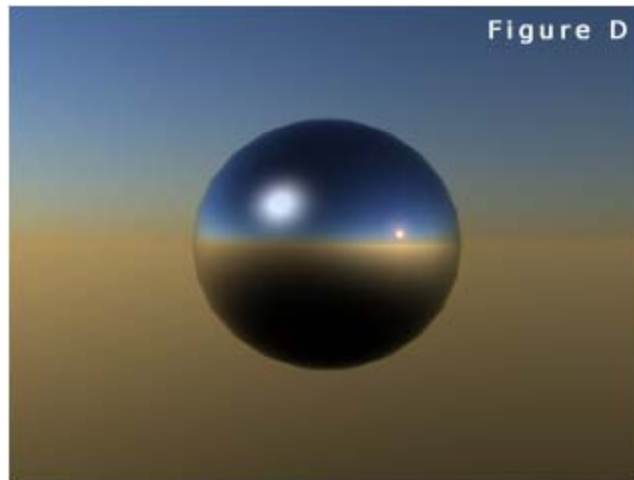


Figure D – our trusty sphere with both specularity and reflection applied.

Having said all this, the cool thing is that making maps for specularity and reflection is exactly the same, as they are logically doing the same thing – defining the shiny areas.

- **So, how about some tips for working with this stuff?**

Here are some useful tips and trivia for making specularity and reflection maps, and working with these attributes when texturing:

Variation! No surface in reality has a perfect, consistent shininess. Everything has been touched in some way by something – whether by people, animals, the weather, or anything else. These things will leave fingerprints, smudges, scratches and other artifacts that will lessen the shininess of the objects surface. It is important to include details like this, as even if they may be really small and almost indistinguishable, they are nevertheless essential details for realistic real-world surfaces.

Show some weathering. The weather, as I mentioned above, leaves obvious marking on surfaces. These sorts of marks include drips, stains, drying, damp and that sort of thing on items. Remember that this sort of damage should also be included in your colour and diffuse maps, and are further enhanced by including them into your specularity/reflection maps as well.

For instance, the paint on a house will, over years, begin to show weathering from rain and sun – in terms of damp gathering in corners, streaky drips down walls, and drying out and cracking where it has been faced by too many hours in the sun. Obviously these details will go into the afore-mentioned colour and diffusion maps, but altering the specular/reflection maps will add more detail to these effects – in terms of the dried, flaky paint will have a broader, less strong shine too it than areas which are in constant shadow, and have become damp.

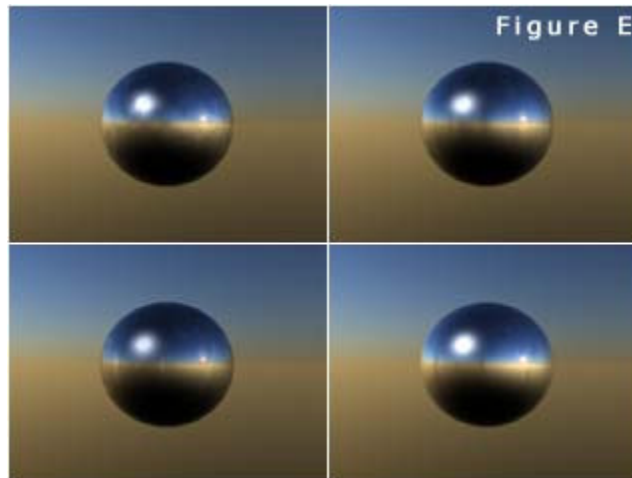



Figure E – add some variation and weathering to your reflection and specularity.

For **human skin** - Look carefully at a face. Notice how the shininess of skin is uneven. A classic example of how shininess differs in skin is to look at the area of skin where the nostril meets the cheek on the face. Almost always, the skin in this area is drier and ever-so-slightly rougher, causing it to be a lot duller than the cheek. The tip of the nose is almost always rather shiny too.

If you have any scars, you will see that scar tissue is much shinier and more reflective than normal skin too. Also look closely into the wrinkles on the joints on your fingers, and you will see that the skin there is smoother and shinier. Also, on the fingers, the bit of skin just above the cuticles and along the sides of the fingertips tends to be very shiny and smooth. If you are texturing skin that has any tattoos on it, you should note that tattooed skin is also much shinier, as it is technically also scar tissue

Surfaces that are scratched – remember that with abrasions and scratches, the shininess tends to change. For instance, if you have a piece of metal that is painted, the paint will have certain shininess. But where there are scratches in the paint, the metal beneath will show through, and the metal will have a different shininess to the paint.

Basic human interaction with objects leaves very specific and identifiable marks – particularly from fingerprints. Our fingertips are very oily, and tend to leave visible residue on everything we touch. For example, if you handle a wine glass, you will definitely leave fingerprints on it, that will alter the reflections in the glass – in other words, where the fingerprints are, you will not see as much reflection as in the untouched areas.

 *Tip: An easy way of adding these to your maps is to make a bunch of custom brushes in Photoshop, which are the shape of fingerprints. You can make prints from your own fingers and scan them in, make selections from them and define brushes from those selections so that you can literally just paint fingerprints onto your maps. Be sure to have a couple of different ones though, of varying sizes and patterns, for added realism.*

Another example of how human interaction alters surfaces is to look at things that have been used frequently, such as your computer mouse. Look at the buttons of your mouse – years of clicking away at them tends to wear the plastic down so that it is smoother, and very slightly shinier, since your fingers have essentially been polishing this area (unless, of course, you haven't cleaned your mouse recently, and instead of it being smoother and shinier, it's just a bit grimy, in which case the shininess would actually be less).

When dealing with specularity and reflection, it's **extremely** important to have a good understanding of where the object has been, and how it is used.

Blur your reflections – if your software has the option of blurred reflections, for heavens sake use it! Unless you are texturing a perfectly clean mirror, most reflective surfaces have a certain degree of blurring to them. Even just slight blurring can help to get rid of that nasty CG look.

Specular blooms – blooming is basically an effect from very, very bright highlights, where the highlight almost seems to glow. This is very noticeable on things that have been covered in some sort of lacquer – such as car paint. The finish on car paint often tends to give off extremely bright highlights when the car is in the sun. Looking at these spots usually gives you mild retina-burn – I'm sure you all know the effect I'm talking about. It's almost like a mini-lens flare, in that it is sparkly, and has lots of little streaks coming out of it. Most software has some kind of bloom effect, in the form of an extra shader, or specular parameter. If you are working with surfaces that are coated in very reflective substances, then adding a bloom can give it a nice touch.

The Fresnel Effect – in the reality, the angle between you and the surface of the object that you are looking at affects the amount of light that is reflected and refracted that you can see. This effect is particularly important when dealing with surfaces, which are transparent. For example, if you look at a lake from a far-off distance, it will appear almost completely mirror-like, yet, as you get closer, and the angle at which you are seeing the water widens, the water appears less reflective and more transparent. This is called the Fresnel Effect (pronounced "fre-nel" – the "s" is silent), an effect that gets its name from the French physicist Augustin-Jean Fresnel, who first documented it.

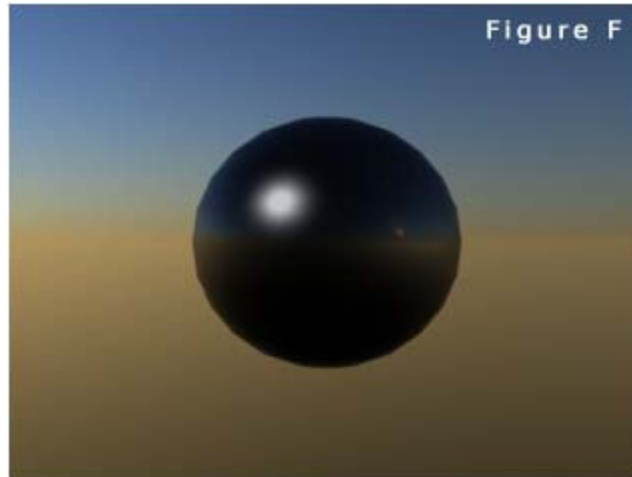


Figure F – the Fresnel effect in action. As the sphere curves away from the glancing angle, the reflectivity of the surfaces increases.

You can implement this effect in one of two ways:

Firstly, by using a **Fresnel shader** (such as the one that comes with Lightwave). The shader pretty much creates the effect for you, just leaving you to adjust it the way you want. A Fresnel shaders effect is generally controlled by adjusting the *glancing angle* for the specular, reflection and transparency amounts (some shaders offer controls for other attributes such as luminosity, diffuse, and translucency as well, but these are, in my opinion, better controlled by gradients).

PHOTOREALISTIC TEXTURING FOR DUMMIES

The glancing angle is measured from the surface normal, thus 0 degrees is any surface normal that points directly at the camera, and 90 degrees refers to any surface normal that is perpendicular to the camera.

So basically, you set the minimum glancing angle and enter in the amount of reflection, specularity and transparency that will be visible at that angle (generally speaking, the lower the angle, the lower the amount of reflection, and the higher the amount of transparency, if the object is transparent), and as the angle increases, the amount of reflectivity will increase.

The second way of creating the Fresnel effect is by faking it with gradients (ramps). Basically, this involves putting gradients into your reflection channel (as well as a gradient of opposing values into your transparency channel, if the object is transparent) that are controlled by incidence angles. In other words, you create an incidence-angle gradient going straight from dark to light, and you put it into your reflection and specular channels, so that when you look straight at the object, it isn't reflective, but as the surface slopes away from you, it becomes shinier and more reflective.

Let's say, for instance that I am texturing a chrome sphere (and for the sake of this example, let us assume that this chrome sphere is absolutely perfect in terms of being perfectly 100% reflective). I would place an incidence-angle gradient into its reflection channel, going from 100% black to 100% white. The effect that this gradient will have is basically 0% reflection when looking perfectly level at the object (0 degrees glancing angle), going to 100% reflective at any part of the object that is viewed at an angle of 90 degrees from the same point – in other words, the front bit of the sphere that I am looking straight at will appear absolutely 0% reflective, while the very edges of it will be 100% reflective.

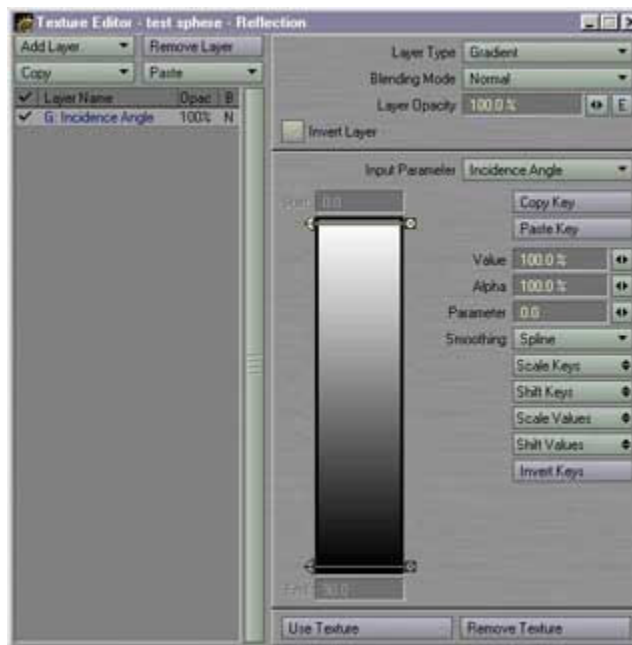



Figure G – a light incidence gradient that will be 0% reflective at a glancing angle of 0 degrees, and 100% reflective at an angle of 90 degrees.

The same technique would apply for a transparent bubble. Now that you understand the basic principle of the Fresnel effect, you should know that if you were to create a bubble floating in the air, if you point a camera straight at it, the bit facing directly level at the camera would appear more transparent than the edges, which would ideally appear more reflective. This effect would be achieved by placing gradients of *opposing* values into the reflection and transparency channels.

In other words, you would place a gradient going *from black to white* (as in the previous example) into the reflection channel, so that the reflection would be 0% when looking at it straight on. Then, you would place a gradient into the transparency channel that goes *from white to black*, so that where it is 0% reflective, the bubble will be 100% transparent, and where it is 100% reflective, it will appear 0% transparent. Make sense? Good!

Remember that this is actually a real world effect, so you should try and utilize it.

 *Tip: Gradients/Ramps are really excellent things to use in texturing. All too often their usefulness is overlooked, when in fact they are absolutely essential for creating certain effects, such as I have explained here. They are also great to use as alpha channels for image maps or procedurals that you may also be using. I highly recommend checking out how the application you use implements them, and start using them! You will most likely find them invaluable once you see how incredibly useful they are, especially for controlling the visibility and placement of other maps and effects in multi-layer texturing.*

BRDF – aaah, the most impressively named effect in texturing - the bi-directional reflectance distribution function. Complicated sounding name, fairly simple effect (in terms of execution). It is generally implemented into 3D software as an extra shader. I'm not sure which applications apart from Lightwave have this function, so I will leave it up to you to find that out.

For people who like to sound like physicists, here is an extremely technical explanation of BRDF for you to try and memorize:

The bi-directional reflectance distribution function gives the reflectance of a target as a function of illumination geometry and viewing geometry. The BRDF depends on wavelength and is determined by the structural and optical properties of the surface, such as shadow-casting, multiple scattering, mutual shadowing, transmission, reflection, absorption and emission by surface elements, facet orientation distribution and facet density.

Hmmm.... That didn't make a whole lot of sense to me.

Simply put, BRDF basically describes what we all observe every day - that the surfaces of objects look different when viewed from different angles, and when illuminated from different directions. It's to do with the directional scattering of light rays from an object's surface (as well as the bouncing of light within an object, known as sub-surface scattering, which is something that I will deal with at a later stage, when I do the workshop on Translucency).

Not to be confused with the Fresnel effect, as this is quite different, the implementation of BRDF in 3D software is basically to add detail to the specular – it allows for the effect of *anisotropic specular reflection*.

Anisotropic specularity is basically specularity with a grain in it. A distortion. It can be used to create what is often called a "brushed-metal" look. Basically what it does is create the illusion of tiny grooves on the surface, which then reflect the light in different directions, depending on the way in which the grooves run along the surface. You can also use the shader to determine which lights affect the surface, and are thereby scattered by these grooves. This results in a broader, softer specular highlight (as in Figure H).

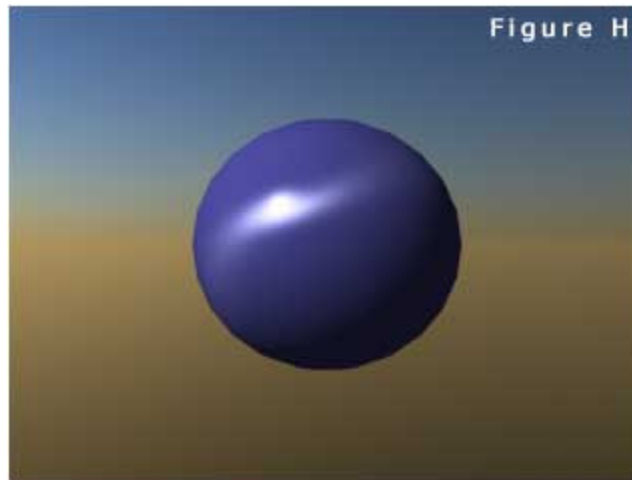


Figure H – BRDF shader with anisotropic specularity.

Another thing that (LWs) BRDF shader does is allow you to add multiple layers of specularity. This is really awesome for when you are texturing something that is coated with varnish or any other lacquer. For instance, if you are texturing a car – a cars surface is covered in paint, which is then coated in a clear seal. These two substances (the paint and the coating) each have a different specular amount, which you can use this shader to set. Pretty nifty.

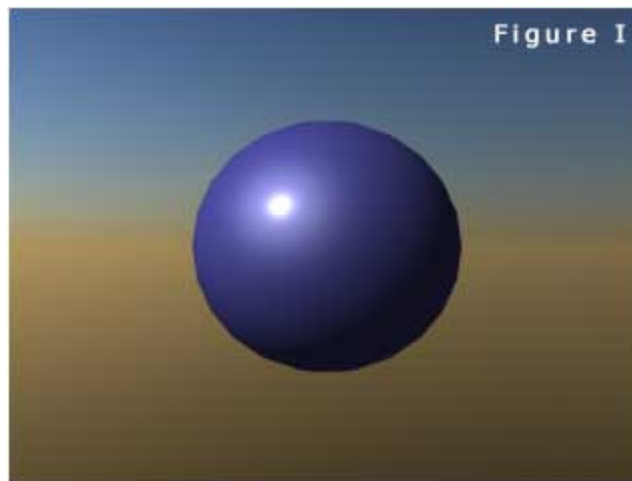


Figure I – BRDF shader with 2 levels of specularity. The lower layer has a tighter highlight while the top layer has a broader highlight.

Lastly, the most important tip of all – DON'T OVERDO IT!! One of the worst mistakes made in texturing is the tendency of individuals (especially beginners) to completely overdo reflection, and make everything look perfectly mirror like. This is a dead giveaway that the object is computer generated. Apart from that, it also looks a bit yucky. So just use it in moderation – study your references for the object you are texturing, and make sure that it's reflection/specularity matches. You will see that cars in the real world are not 100% reflective, and neither are chrome teapots (although those *are* very reflective, just not 100%).

- **Ummm... wasn't this workshop also supposed to be about Gloss??**

Well, I left glossiness to until the end because there isn't all the much to explain - basically, the gloss amount in your surface editor controls how spread out the specular highlights are. The higher the gloss amount, the tighter the highlights. For example plastic tends to have a fairly high glossiness, whereas fabric such as

cotton has practically no glossiness. Pretty simple. Usually you have to have some amount of specularity on a surface to use glossiness, or else the gloss amount won't actually have any effect, because it works hand-in-hand with specularity. And that's about it!

Part 6 – Translucency, Transparency and Refraction

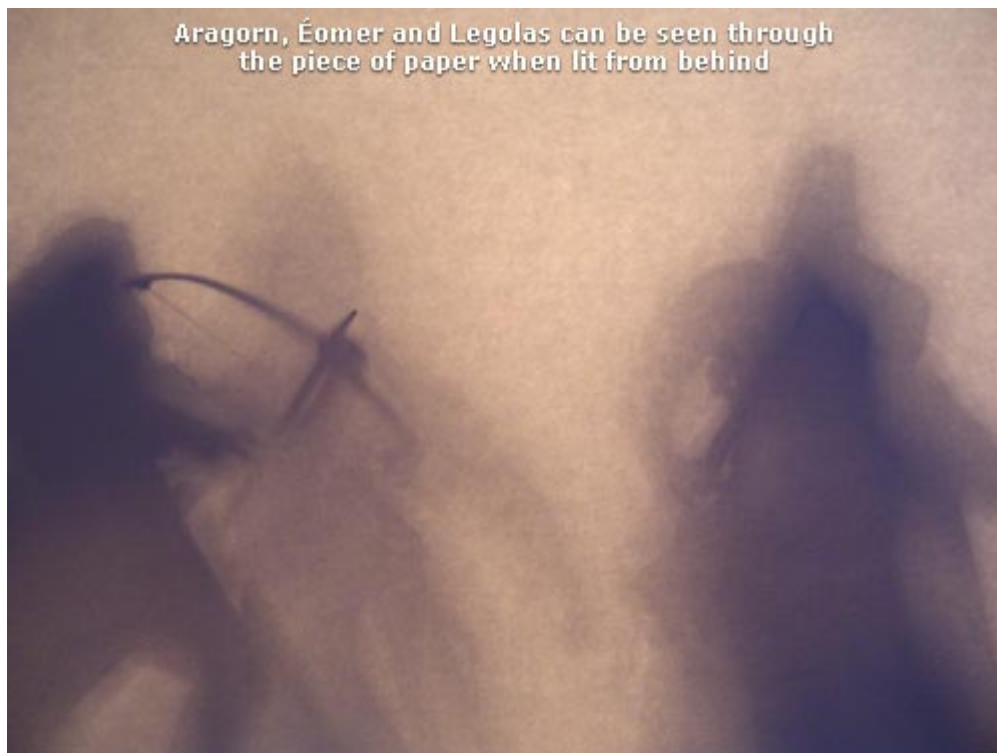
- **What is Translucency**

Yes, what is it? Translucency, simply put, is the ability for an object to have light travel through it, or into it to a certain degree, resulting in it appearing to be backlit (in the case of a curtain or sheet), or to appear to have some faint luminous quality of its own (as seen particularly in skin).

As we know, when light hits surfaces, it is bounced (reflected) back off the surface. Well, in the case of translucent surfaces, some of that light travels through the surface as well. This phenomenon occurs in most substances in this world, except metals and most woods.

Imagine a theatre stage with its curtains drawn. If you were to place strong lighting from the back of the stage, you would be able to discern shapes behind the curtain without actually being able to see any real details in those shapes. It's almost as if you can see the shadowy outlines of things. That is an example of translucency, as the material that the stage curtains are made of is translucent, allowing you to be able to see shapes through it when properly lit.

Take a look at the following image. This is my own hurriedly put-together theatre simulation experience involving a bunch of *Lord Of The Rings* action figures and a piece of paper.



This is different to transparency because the surface is not *see through* as such, and so you can't actually see things in as much clarity and detail as you would when looking through glass. This is because you are predominantly seeing the shadows of those objects behind the curtains being projected onto them. If the curtains were made out of metal or wood, which are not translucent, we wouldn't be able to see them. Apart

from the shadows, you can also sometimes make out the faint colours of things through the curtains, especially if those objects are fairly close behind the curtain.

Another great example of translucency is when you shine a very strong light under your hand when it is dark, and you can make out faint details beneath the skin. This is because your skin is quite translucent.




Notice how the veins appear darker in the photo, because they interrupt the travel of light through the surface. If the light were stronger, you would also be able to see the outlines of the bones, because the bones would also interrupt the light.

○ Using translucency in your surfaces

Okay, so now that we know what translucency is, just how do we go about making textures that make our surfaces behave in this way?

Well, one way would simply be to go to your material editor and increase its translucency amount. By default, your translucent value would be 0 in all packages, since this one is not a universal surface property. Obviously increasing this value makes your surface more and more translucent, allowing more and more light to travel through it.

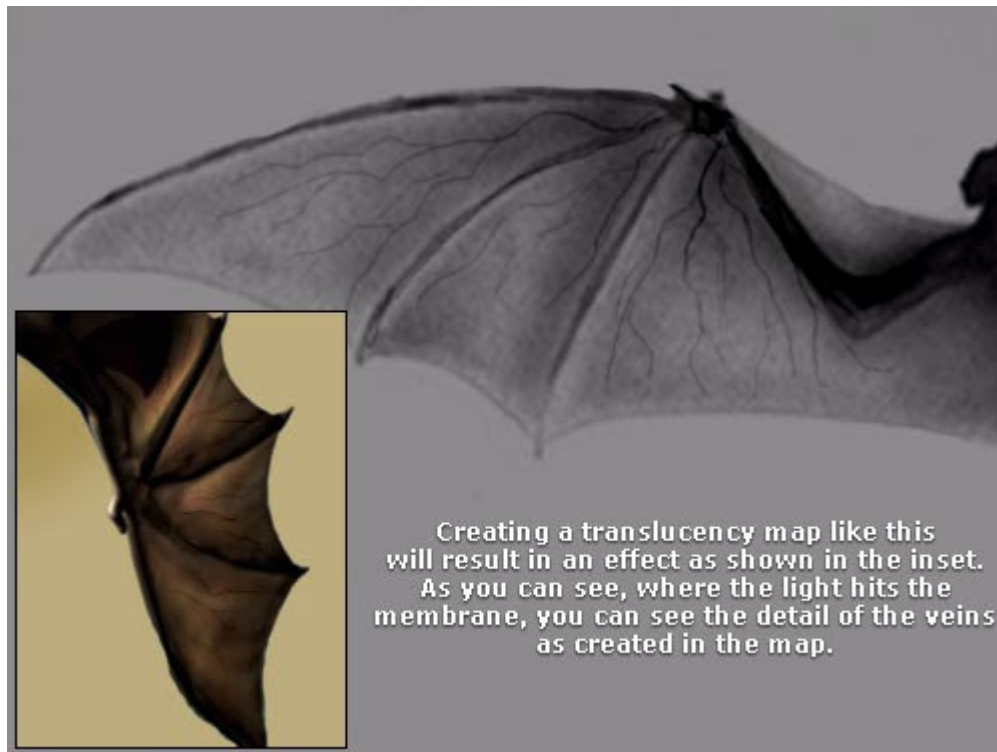
 **Note:** To my knowledge, most (if not all) of the major 3D packages have translucency included these days in their surface editors, so just look for it and you'll find it somewhere. In the case of programs like Max that separate raytracing surfaces from normal scanline ones, you might have to look in your advanced (ray traced) surfaces panels for a translucency channel.

PHOTOREALISTIC TEXTURING FOR DUMMIES

Simply altering this overall value can be cool for quickly establishing a decent translucency for the surface as a whole, when it is applied to a surface that doesn't really have major variations in this surface attribute, such as really basic clothing, curtaining, flags, candles... you get the picture. Although these could have extra details added by creating an actual texture map for it, you could still get away with just altering the overall amount in these cases.

But, of course, my favorite way of using translucency is to paint a nice translucency map! Aaaaah, nothing beats a nice detailed map with little veins and things in it.

Okay, so let's imagine that you have to make a texture for a dinosaur-like creatures wing. You could paint some really cool colour maps and whatnot for it and get it looking okay, but think about it – the membranes of the wings would be pretty translucent, wouldn't they? Well, the arm bits of the wings would be translucent too, but lets not worry too much about those in this case. So, to make all the little veins in the translucency map so that they will show up when the light is hitting the wing from behind, we would do something like the example below.



As you can see, it's actually pretty simple. Make lighter shades of gray where you want the light to shine through nicely, and "block" the translucency by adding any darker areas, such as the veins.

○ I keep hearing about this thing called Sub Surface Scattering – what is it??

Okay, now onto sub surface scattering. Where we have any degree of translucency, we find the phenomenon known as sub surface scattering, more commonly referred to as SSS. I don't have lots of time to really go into detail about this now, but very simply put, SSS is when light rays penetrate a surface at a particular angle, and then kinda bounce around a bit inside the surface and then leave it at a different angle to that which it entered at, which causes the surface to appear as if it has some illumination of its own.

Using this effect can give a lot of substance to an otherwise really flat looking surface, which is particularly useful for substances like skin, milk, wax, etc.

These days, SSS is usually implemented by using special renderers or plugins. Most of the major 3D applications have a plugin available for this, or some form of implementation.

Since I do not have any SSS plugins of my own right now, I cannot give you any examples here... boohoo. But I suggest doing a bit of research on it on your own – go to Google.com and do a search on it. There are lots of interesting articles about it on the Internet.

And that pretty much sums up the basics of translucency!

○ **Transparency**

Right, we all know what transparency is; so no in depth explanations are really required here. Transparency is the quality whereby a surface is “see through”. Glass, Perspex, water (actually most liquids really), some plastics, etc are all examples of transparency in the real world.

○ **Using transparency**

Rule number one of transparency: if your surface isn't looking correct, DON'T go and make it blue. This is something that we see all the time with beginners. When people who are new to 3D want to make glass objects, for some bizarre reason you often see them giving the surface a certain degree of transparency, and then... making it blue. Why this is, I don't know, but don't do it. It looks crap.

Okay, so sometimes transparent objects do have a colour of their own, but setting up colour for them often requires a bit of effort, depending on what software you are using.

Now, for starters, let's quickly deal with transparent surfaces such as clear glass that have no colour of their own. To get the best results, take the transparency all the way to somewhere between 95% and 100% (don't worry, your object won't disappear) and make it a bit specular and reflective. Ideally you want to implement the Fresnel Effect into this. If you don't know what that is, here is a quick recap from *Texturing For Dummies Part 5*:

The Fresnel Effect - in reality, the angle between you and the surface of the object that you are looking at affects the amount of light that is reflected and refracted that you can see. This effect is particularly important when dealing with surfaces that are transparent. For example, if you look at a lake from a far-off distance, it will appear almost completely mirror-like, yet, as you get closer, and the angle at which you are seeing the water widens, the water appears less reflective and more transparent. This is called the Fresnel Effect (pronounced "fre-nel" - the "s" is silent), an effect that gets its name from the French physicist Augustin-Jean Fresnel, who first documented it.

Remember, that in order for a reflective surface to work properly, *you have to give it something to reflect*. This could be an actual modeled environment around the object, or an image environment such as an HDR image. Pretty much anything that is transparent is also reflective, so setting up your reflection correctly is pretty important.

Right, all that understood? Cool, let's continue...

Now, take the surfaces diffusion value down to 0%. This is because we don't want any colour in this particular surface.

This should give you a decent looking glass, such as in the image below.



Now, if you want to give some colour to this glass, you obviously have to give it some diffusion again. Then, give it some colour. The problem with giving colour to glass though is that it often tends to look washed out, and unintentionally there. Ideally to get the colour looking right, you need to look at whatever special options your software has for transparent surfaces. In LW (which is what I use), we have colour tinting and colour highlights options, which work really well to make the colour stronger and more solid in transparent surfaces. Try and keep your diffuse value as low as possible to help prevent the colour from becoming washed out.





Right, now that we have dealt with simple examples of transparency, let's quickly take a look at some slightly more involved ones. Not all transparent surfaces are nice and clean and shiny and perfect like the two previous examples. Some of them can be icky and filthy. Like a window that doesn't get washed too often. The corners of windows often tend to gather dust and grime, and if we were to create a texture for that sort of thing, we would have to somehow make sure that the grime affects the transparency of the window correctly, right? Take a look at the following image.



Pretty gross, eh? Trust me, it was like that when I moved into the place...Okay, so if we were to recreate that in CG, and wanted to make sure that the glass texture looked right, we would have to take into account the fact that the grime on the window would affect its transparency, because logically where there are dirty areas or grime, those areas would lessen the transparency, because they are interfering with the travel of light through the transparent surfaces, just like we saw in the translucency example.

So if I was to paint a transparency map for that part of the window, it would look something like the following image (bearing in mind that this texture map would only be applied to the glass part of the surface).

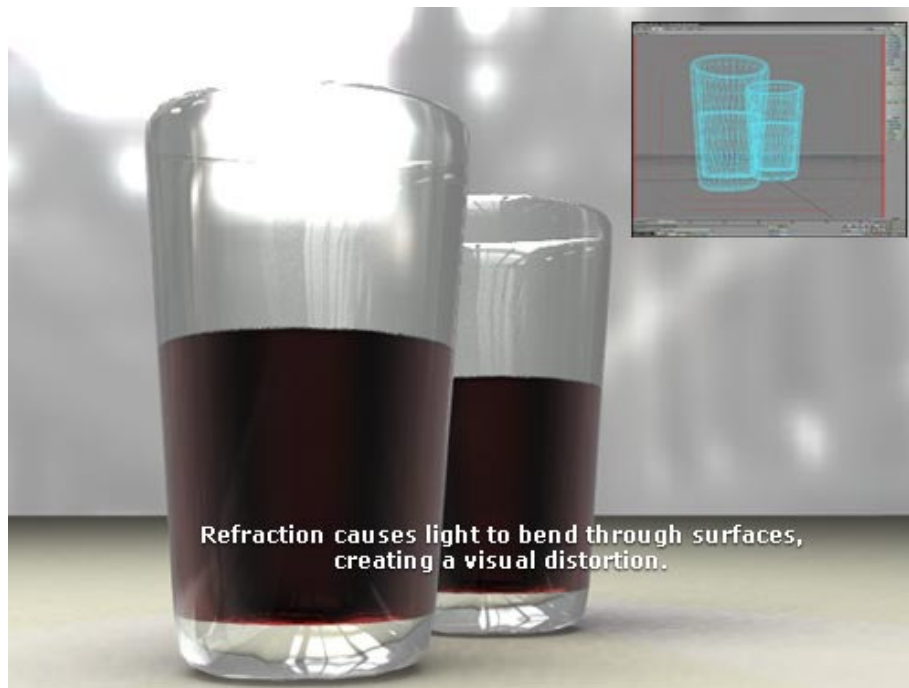
Remember that you would make sure that all your other texture maps for this window would correspond correctly with this map. In other words, you would include the colouring for the grime in the colour map, as well as assign different diffuse and specular and bump values for it as well.

When working with transparency, it's best to use double-sided surfaces, or to create both sides of the surface in the actual model. If you use normal one-sided polys you tend to lose all illusion of volume, which ends up looking fake.

○ And what about refraction?

Any surface that has any amount of transparency on it, also has a level of refraction. Refraction is basically the phenomenon of light bending through surfaces. You know how when you look through a piece of glass, things appear to be slightly shifted or even somewhat distorted? That is from refraction.

Take a look at the following image. Looking at the wireframe, we guess that we should technically be able to see the glass in the background through the one in the foreground, but when we look at the render, we cannot. This is because of the refraction in the glass, which causes the view through it to shift, resulting in us not being able to see the other glass.



The amount of refraction is determined by the surfaces *refraction index*, which is most commonly a number between 0 and 2, although some substances can exceed 2.

PHOTOREALISTIC TEXTURING FOR DUMMIES

Some common refraction indexes:

Substance	Refraction Index
Air	1.0003
Alcohol	1.329
Crystal	2.00
Diamond	2.417
Emerald	1.576
Glass	1.51714
Glass, Albite	1.4890
Glass, Crown	1.520
Glass, Crown, Zinc	1.517
Glass, Flint, Dense	1.66
Glass, Flint, Heaviest	1.89
Glass, Flint, Heavy	1.65548
Glass, Flint, Lanthanum	1.80
Glass, Flint, Light	1.58038
Glass, Flint, Medium	1.62725
Ice	1.309
Mercury	1.62
Plastic	1.460
Ruby	1.760
Water (gas)	1.000261
Water 100'C	1.31819
Water 20'C	1.33335
Water 35'C (Room temp)	1.33157

- **That's all folks!**

And that pretty much wraps up another chapter of Texturing For Dummies! As usual, please feel free to add your own tips and tricks and information to this thread. The other bunch of dummies threads have turned into goldmines of info so let's keep up the tradition and make this thread a nice big hoard of information too!