# Physically Based Shading in Real Time Rendering

Christian Schüler
MBL Development

## Introduction

### Who's me?

- Veteran: Engine programming since 15 years
- Several shipped games
- Doing "serious" work now but still love games

## Introduction

### Physically Based Shading

- Solves some real problems
- Can make authoring easier
- Can improve looks
- Feels as the “right” way to do things
- Is going to be implemented in more and more engines

Introduction

## Ingredients for Physically Based Shading

- Linear Lighting
- Energy Preserving Specular
- Metals + Dielectrics
- Fresnel Reflectance
- Image Based Reflections
- Microfacet Models

## Introduction

Ingredients for Physically Based Shading

- Linear Lighting
- Energy Preserving Specular
- Metals + Dielectrics
- Fresnel Reflectance
- Image Based Reflections
- Microfacet Models

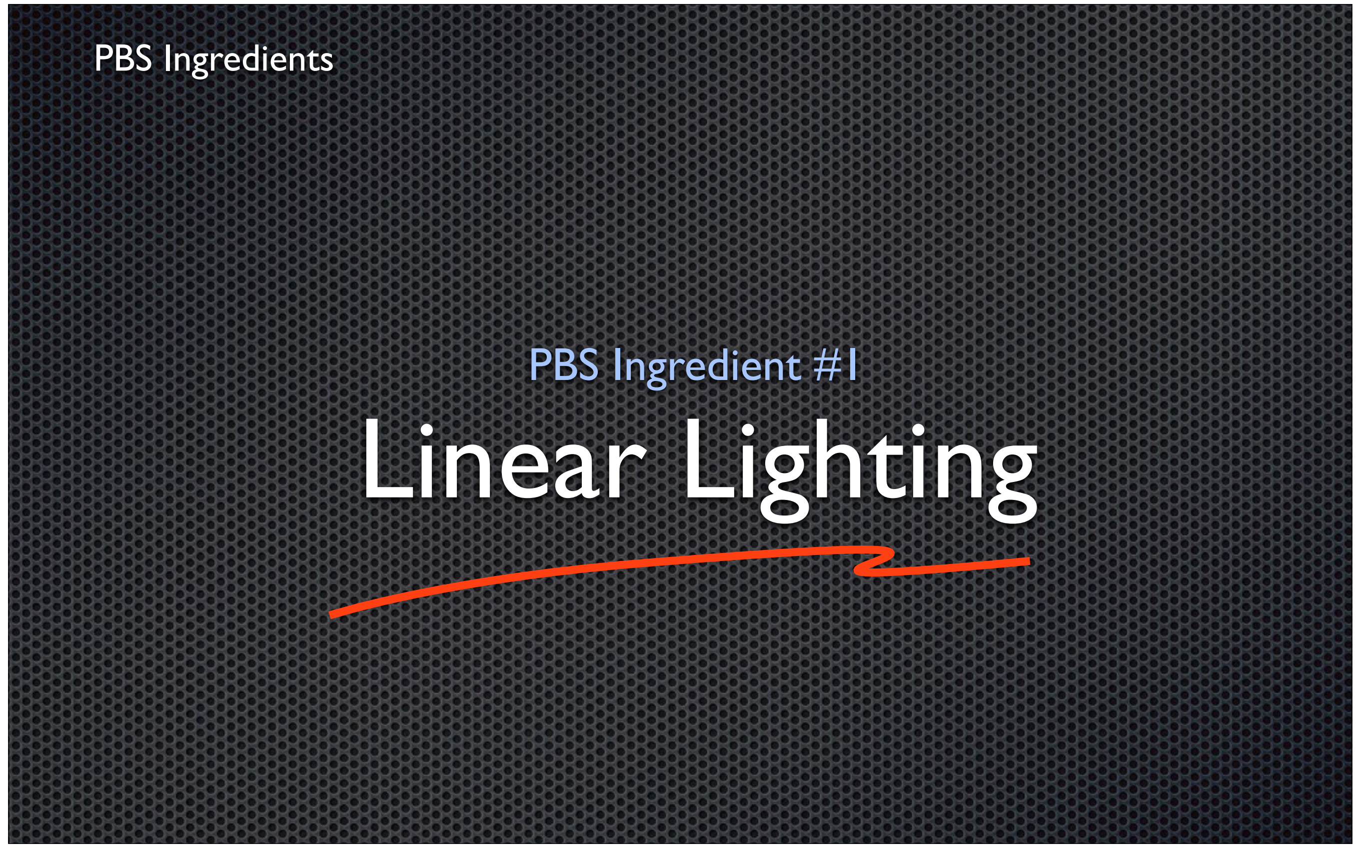
will talk about these today
important for content creation

## Introduction

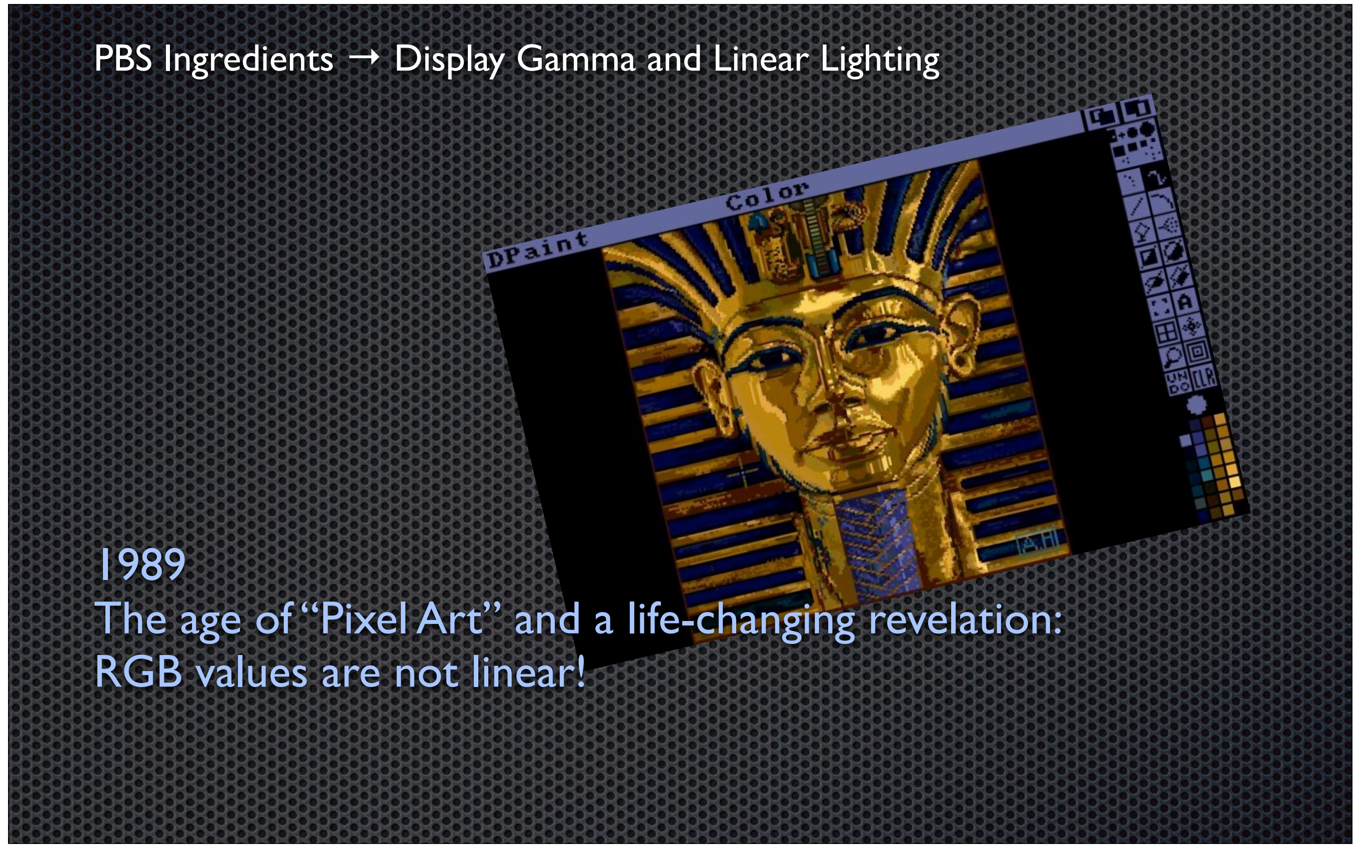
Ingredients for Physically Based Shading

- Linear Lighting
- Energy Preserving Specular
- Metals + Dielectrics
- Fresnel Reflectance
- Image Based Reflections
- Microfacet Models

programmers and shader
authors worry about these




The first ingredient of physically based shading is Linear Lighting.
This means, doing lighting computations in linear space (as opposed to gamma or RGB space).



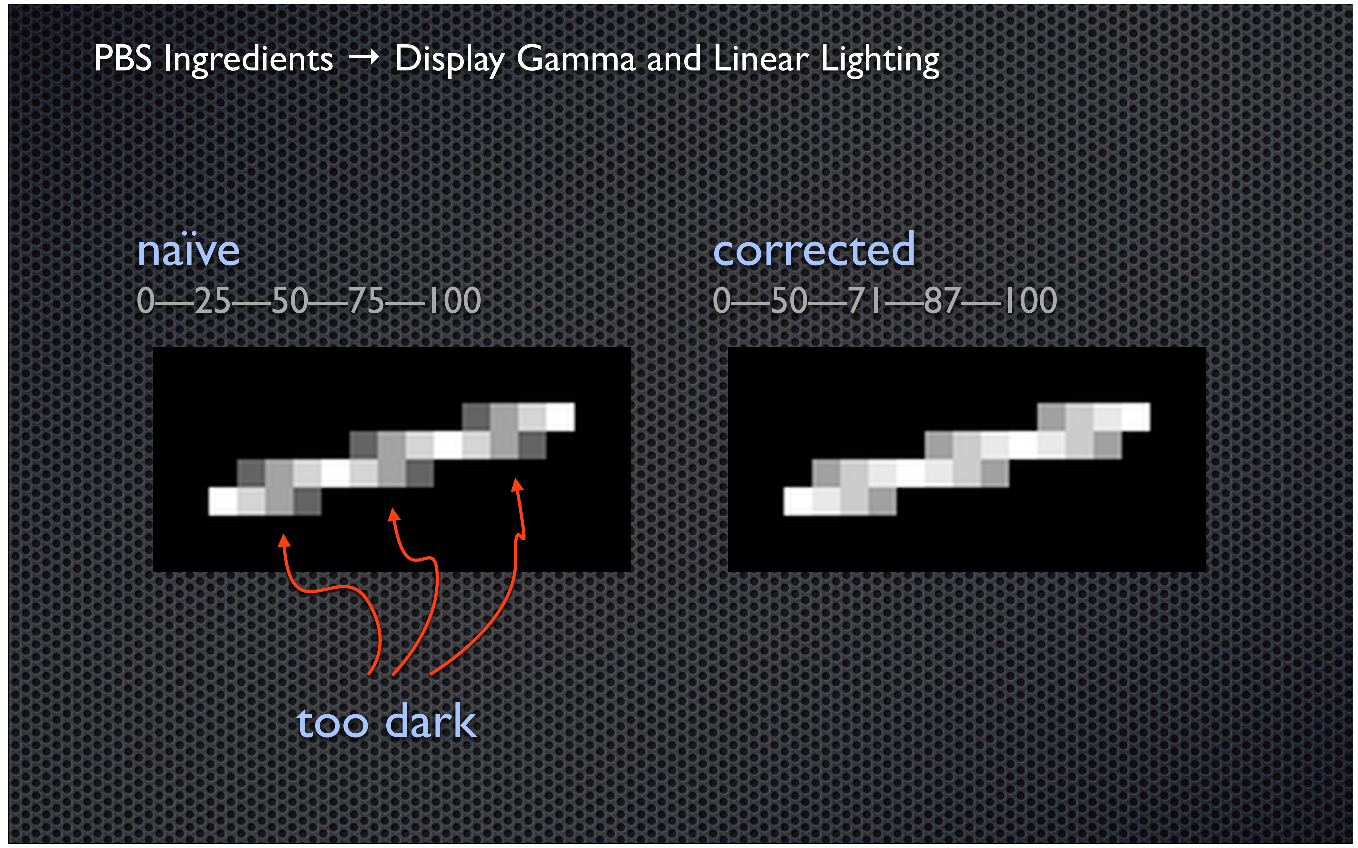

Linear lighting is connected with the topic of display gamma.

I discovered the effect of display gamma back in a time when computer graphics was dominated by ‘pixel art’, i.e. generally an artist had to manipulate individual pixels and manage the color palette by hand (images were usually restricted to a small number of different colors, like 8 or 16). Any “in-between” color used for anti-aliasing had to be included in this palette.

(Auditorium Question: Who remembers Deluxe Paint?)

One day I wondered why the RGB-values for such “in-between” colors are usually different from a simple linear average of the RGB-values of neighboring colors. So I conducted some experiments and finally hypothesized that the displayed brightness of a color must be proportional to the RGB value squared.

This was not a bad guess, and later I learned about “display gamma” and that the true relationship is not exactly a power of 2, but usually a somewhat higher power like 2.2. The important thing is that calculations done naively with RGB numbers do not yield the expected results.

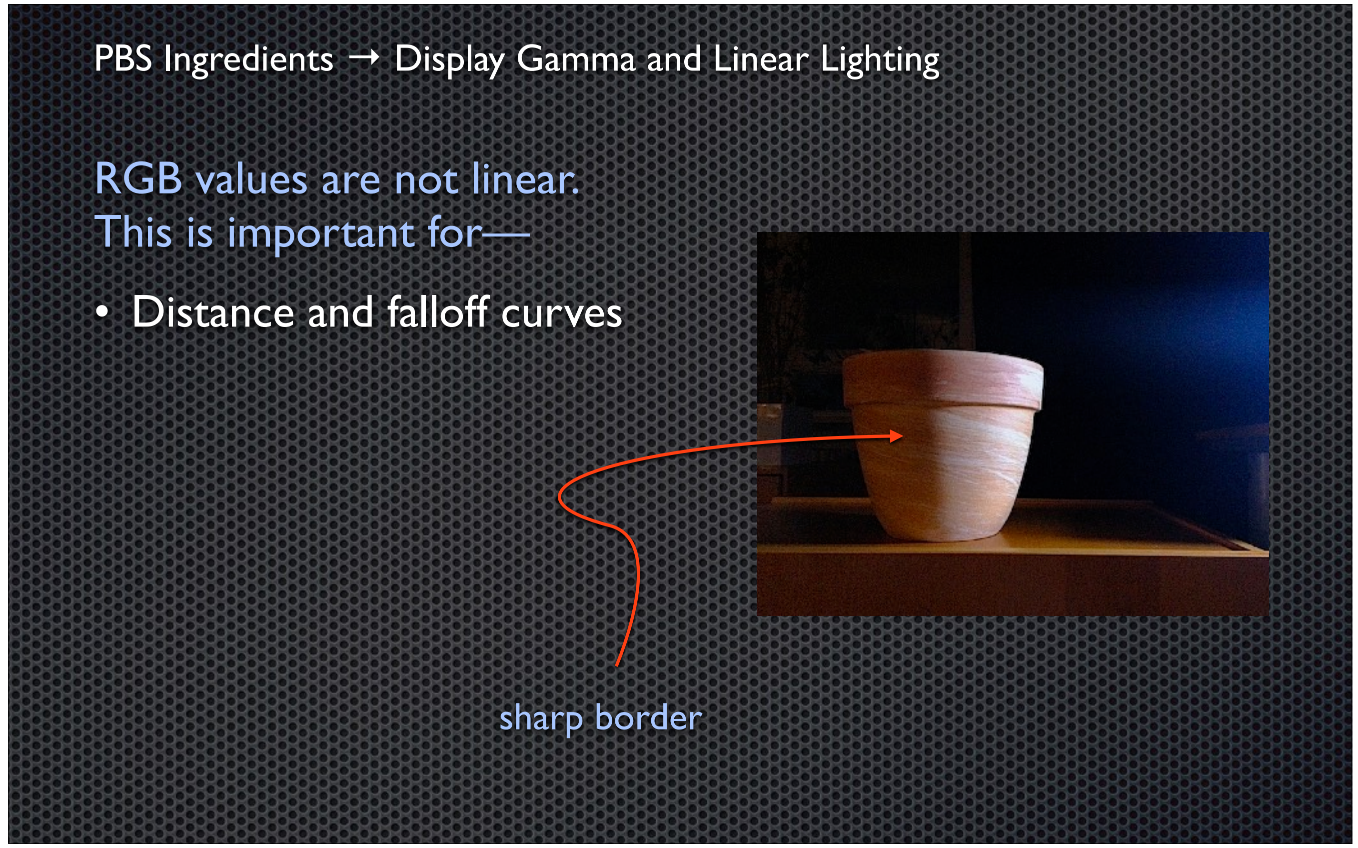



This is how one of the experiments looked like.

In both figures there is a rasterized line with grey pixels used for anti-aliasing. On the left hand side, the grey values are taken directly from the horizontal position, so they go from 0% (black) via 25%, 50%, 75% to 100% (white) and back again to 0%. When displayed on a typical computer monitor, this pattern appears too dark in the in-between parts. It almost looks like a stippled line with fluctuating brightness.

On the right hand side, the grey values were corrected using the assumption that the displayed brightness is the square of the grey value. (When I made the discovery, I initially thought this to be true) Under this assumption, 50% is displayed as 25%, 71% is displayed as 50% and 87% is displayed as 75%. It seems this assumption is not too bad, because the visual impression is already much nicer, with a line that appears to have stable brightness.

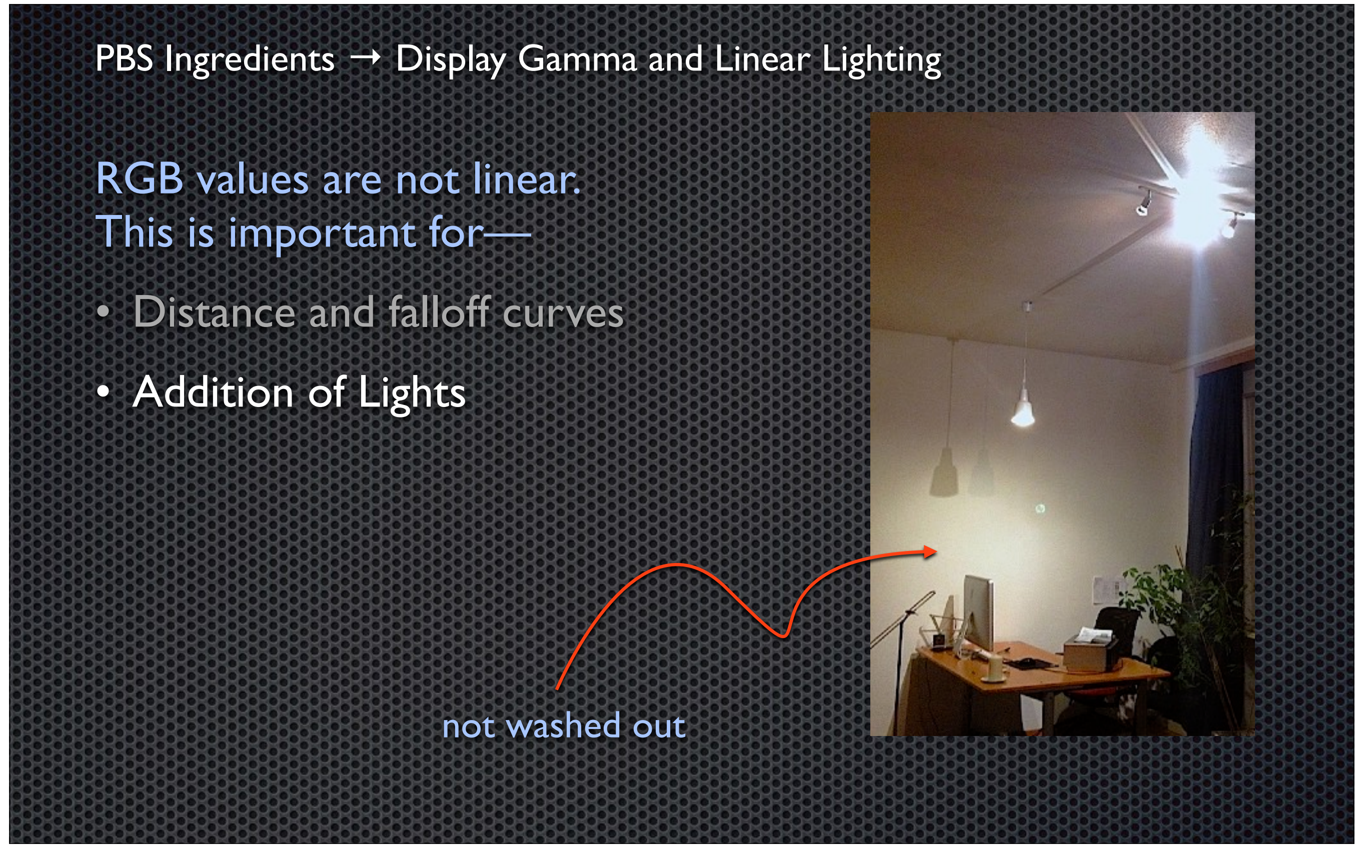
What we have just done is a gamma correction for a gamma of exactly 2. The standard display gamma value is not exactly 2 but 2.2, and so the grey values would be even slightly higher. But the take home message is here: RGB values are not linear.




So what does it mean when RGB values are not linear? When an engine or a renderer does not account for the non-linearity of RGB values in its computations, it is likely to get them wrong.

Distance attenuation and falloff curves are one example. The distance attenuation would be too strong (light gets dimmer with distance faster than in reality), and likewise the angular attenuation that follows the Lambertian cosine law would get too dark too early.

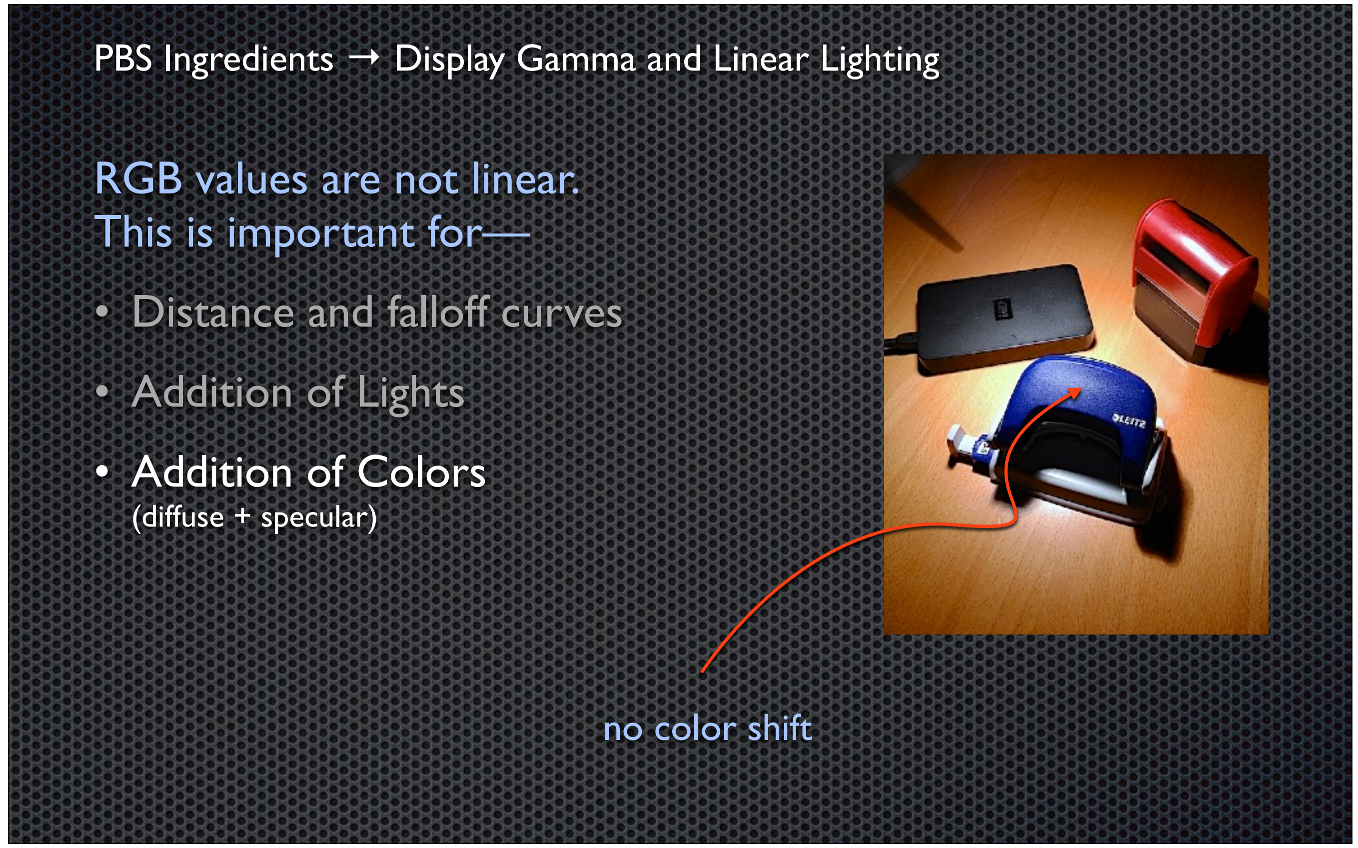
See the illuminated clay pot which acts as a real-life proxy for a Lambertian surface. In reality, the light intensity seems to vary only little with the surface angle, right up until the surface becomes perpendicular, then it makes a sharp border. Surface detail (from the normal map) will become very pronounced and noticeable in this border region. Only a linear light renderer will reproduce this behavior.




The addition of lights will be affected strongly whether this is done linearly or non-linearly.

Because the display gamma has a power greater than 1, the addition of two or more lights creates more energy than the individual lights had combined. A non-linear renderer would therefore have a tendency for washing out and clipping. Artists would have a hard time to make the individual lights bright enough to hold on their own while at the same time paying attention that the combination of multiple lights does not clip.

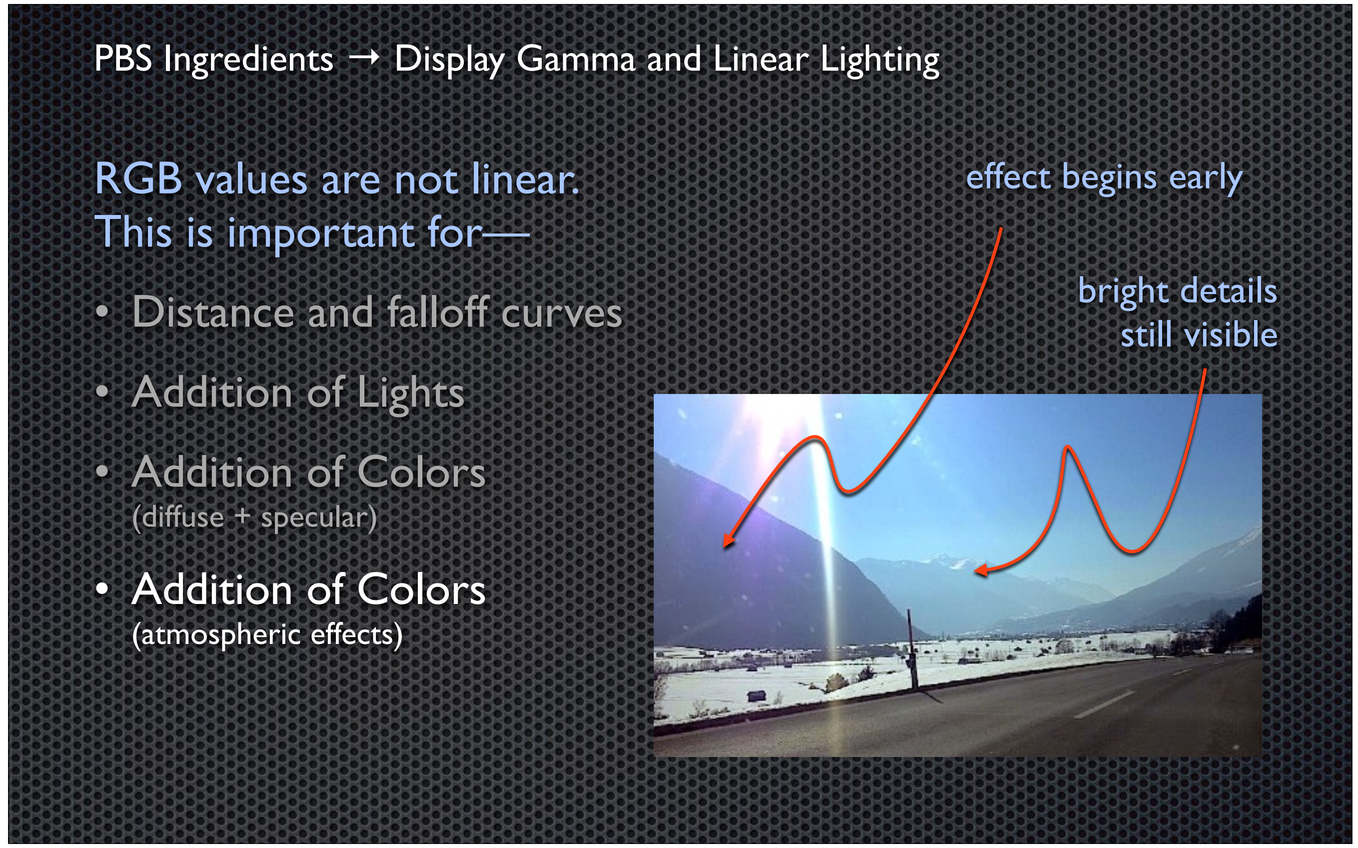
With linear lighting, things are much more relaxed and “multiple-light friendly”. Even without any tone-mapping.




What was said about the addition of lights is also true for the addition of colors, but now individually for each color channel.

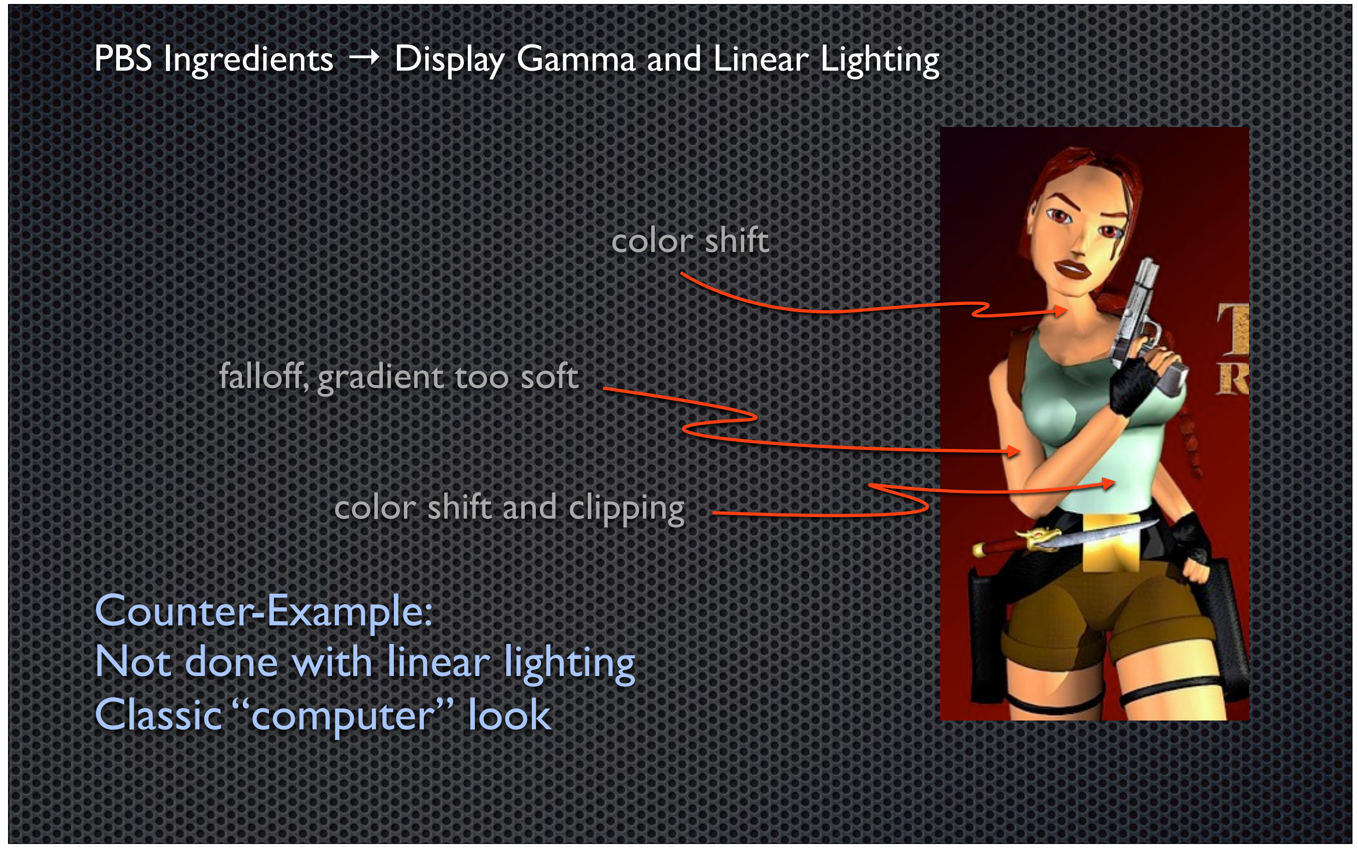
An important case for surface shading is the addition of diffuse and specular contributions. A non-linear renderer would have a tendency to wash-out and clip specular highlights, just as it does with multiple lights. Since this applies to the color channels individually, there would also be a noticeable color shift around the specular highlight.

The result looks rather cartoony and “computer-graphicy”. In reality, there is no noticeable color shift (more precisely: hue shift) around a specular highlight, except for materials that are designed to have one, like special car paints.




Things are more complex when it comes to atmospheric effects, like distance fogging.  The mixture of the surface color with the fog color looks recognizably different when done either linearly or non-linearly. A non-linear renderer would produce a perceptually uniform fogging (with a good dose of color shift).

In reality, the fog effect is highly non-uniform. It can be seen very early over dark background surfaces. Bright surfaces however do not seem to be affected much at all, even far into the distance.




Here is a counter example: A rendering which is typical for 90's renderings. The lack of linear lighting is one of the reasons why the image has this archetypical "computer graphics" look.

Of course, armed with this knowledge, lack of linear lighting can be used intentionally as a stylistic device.

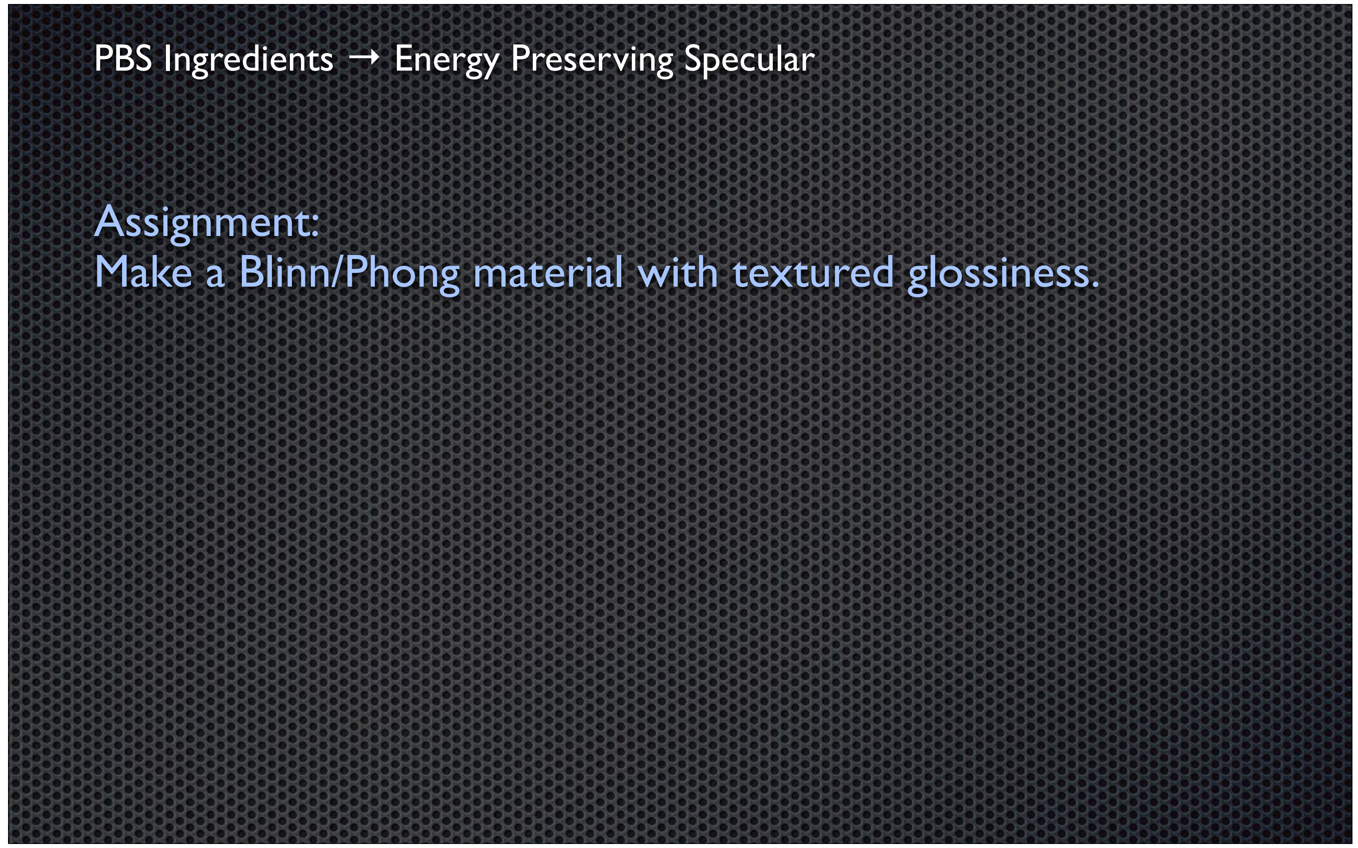
## Further reading

- Blogs
  http://filmicgames.com/archives/category/gamma
- Unreal
  http://udn.epicgames.com/Three/TexturingGuidelines.html
- CryEngine
  http://freesdk.crydev.net/display/SDKDOC4/Gamma-Correct+Rendering+(sRGB)
- Unity
  http://docs.unity3d.com/Documentation/Manual/LinearLighting.html


The second ingredient of physically based shading is an Energy Preserving Specular Highlight.

(No, this is not about being environmentally friendly.)

This means that the brightness of the specular highlight is automatically linked to its size, so that the overall energy of reflected light is conserved.



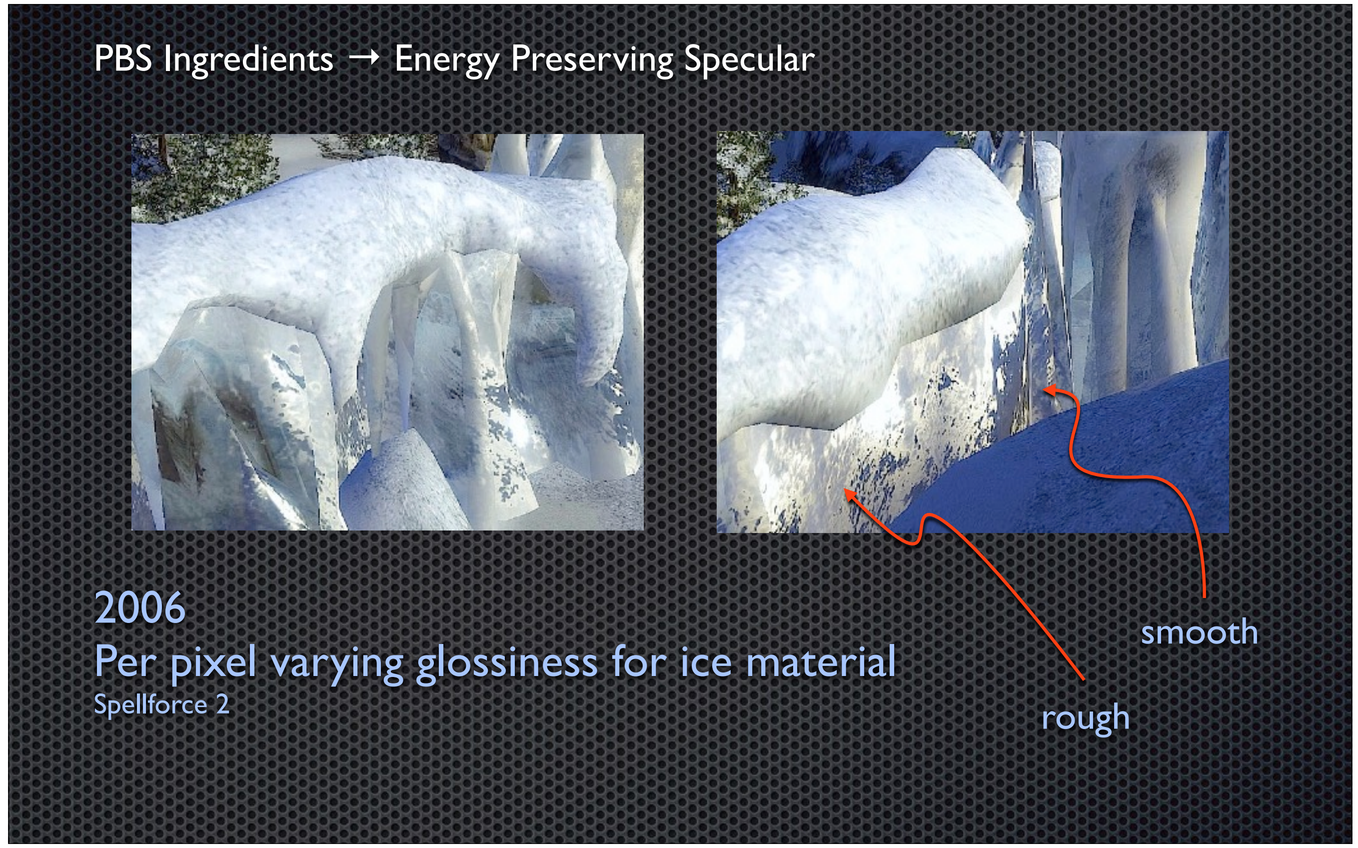

But before I’m going to explain the details of energy conservation, I would first like to talk about varying glossiness. Suppose the task is to make a material where the *glossiness* is not a constant, but textured.

## PBS Ingredients → Energy Preserving Specular

Assignment:
Make a Blinn/Phong material with textured glossiness.

Why?

(Auditorium Question: Who did this before? Why? For what occasion?)




Here is one example.

The first time I had a need for varying glossiness was when I was working on the ice/crystal material for Spellforce 2 (which was released in 2006). Varying glossiness was one aspect of the ice shader based on the observation that parts of an ice surface are very smooth (high glossiness), because of melting, and other parts may be very rough (low glossiness), because of hoar frost grown over the surface.

The screenshots show an example how this looked in game.

## PBS Ingredients → Energy Preserving Specular

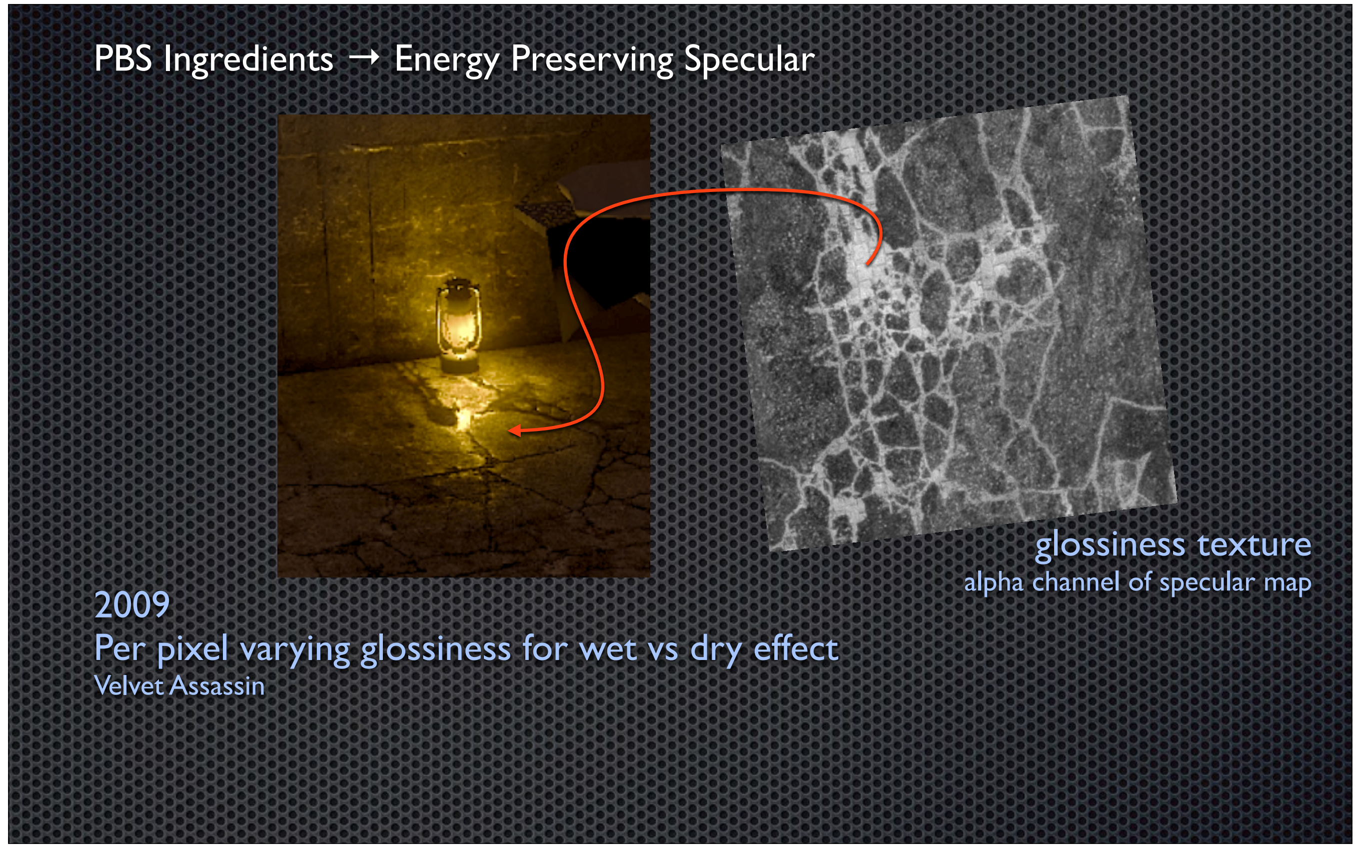
rough
low glossiness

smooth
high glossiness

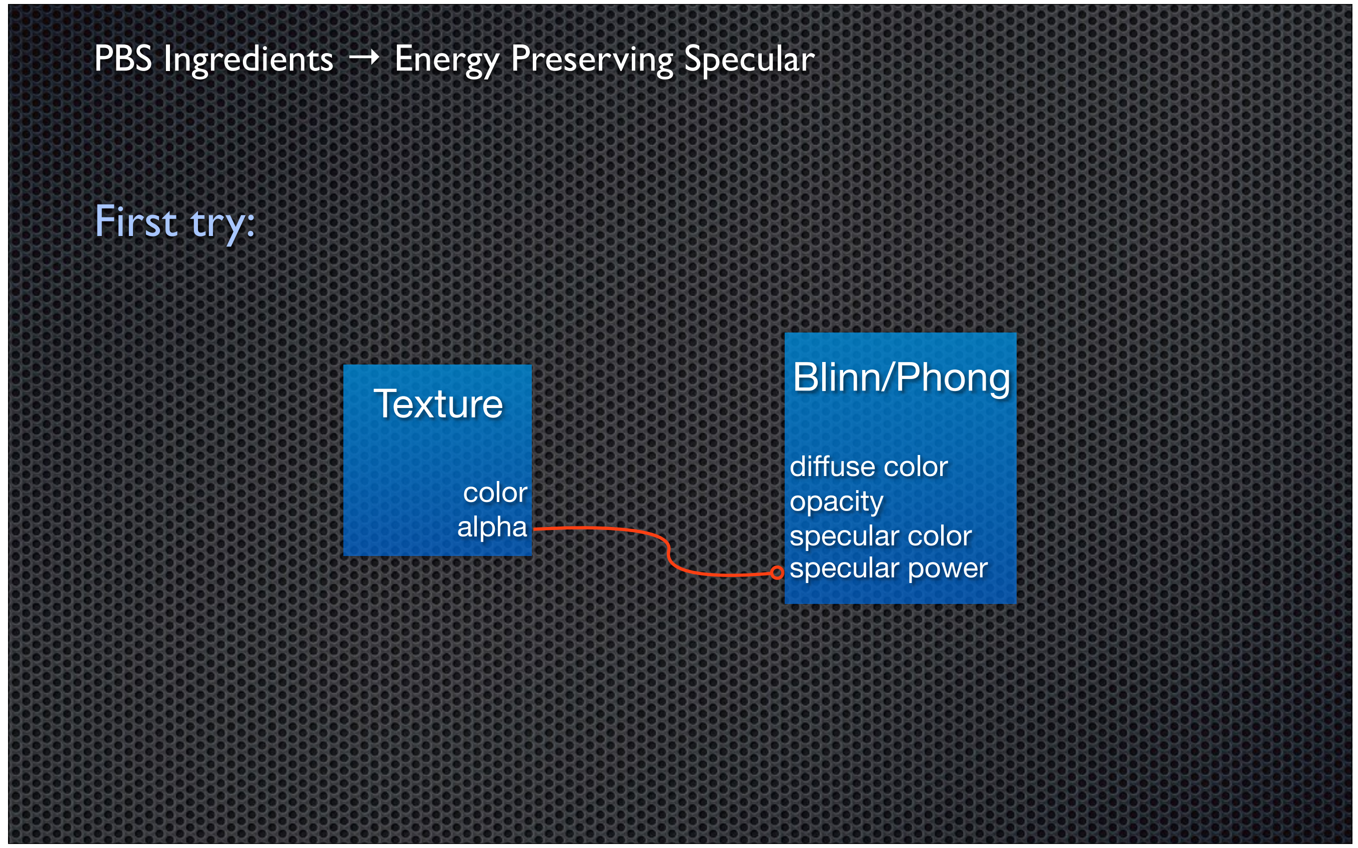
http://en.wikipedia.org/wiki/File:Frost_patterns_4.jpg


Another example. These screenshots show how varying glossiness was used to achieve a the effect of wet streaks on a surface.

Note that the main contribution for this effect is indeed a change in glossiness, not a change in the specular level (aside from a little darkening of the diffuse texture where the surface is supposed to be wet). The Physically Based Shading takes care of all the variation that goes along with the glossiness. More about that later.

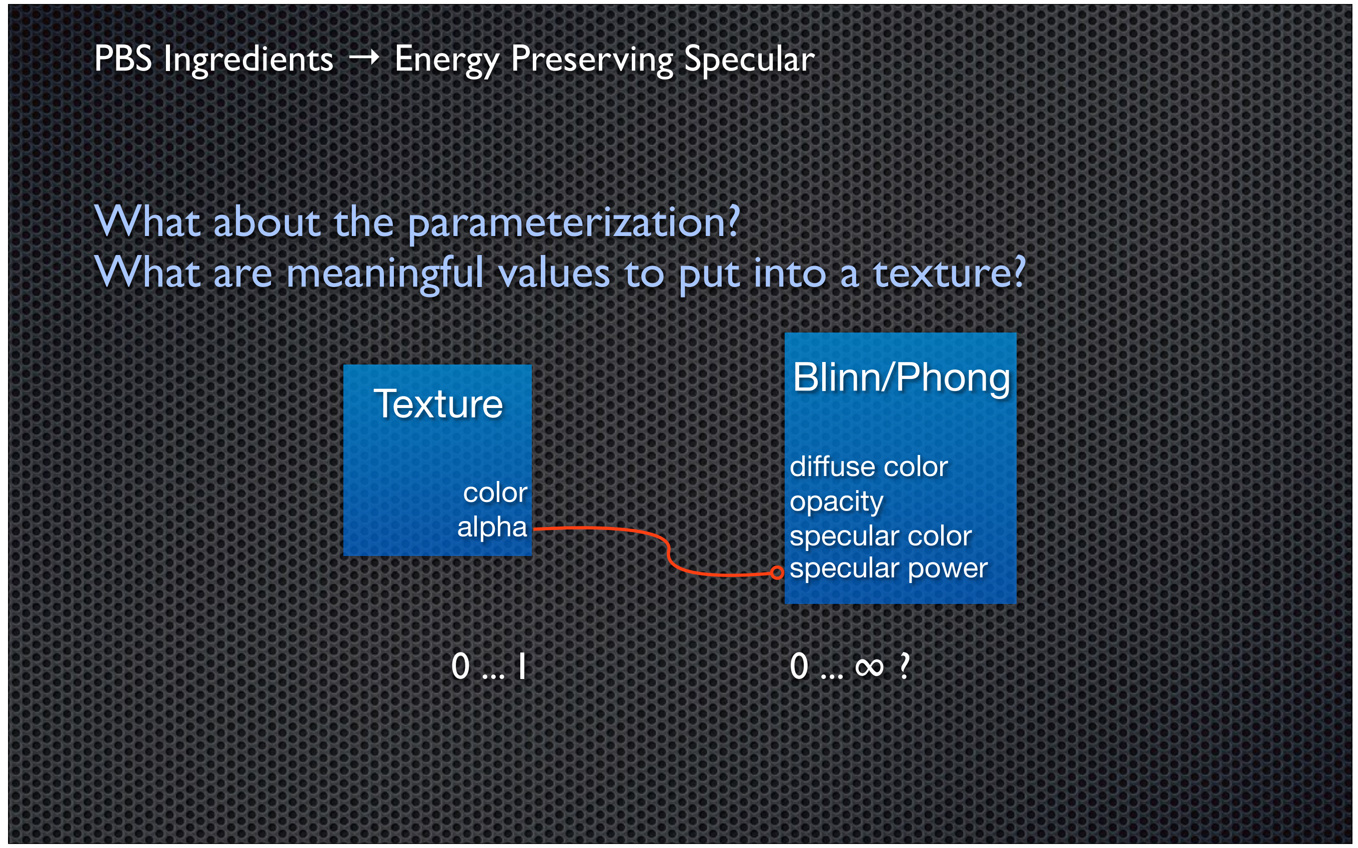



And here you can see an example of a glossiness texture and how it looked like in game. The glossiness texture was always put into the alpha channel of the specular map.




Ok, this is easy to do if you have access to a shader editor oder even to shader code.
Connect a monochrome texture output to the ‘specular power’ or ‘specular exponent’ of a shader, so that the texture channel controls the size of the specular highlight.
It would look like this.

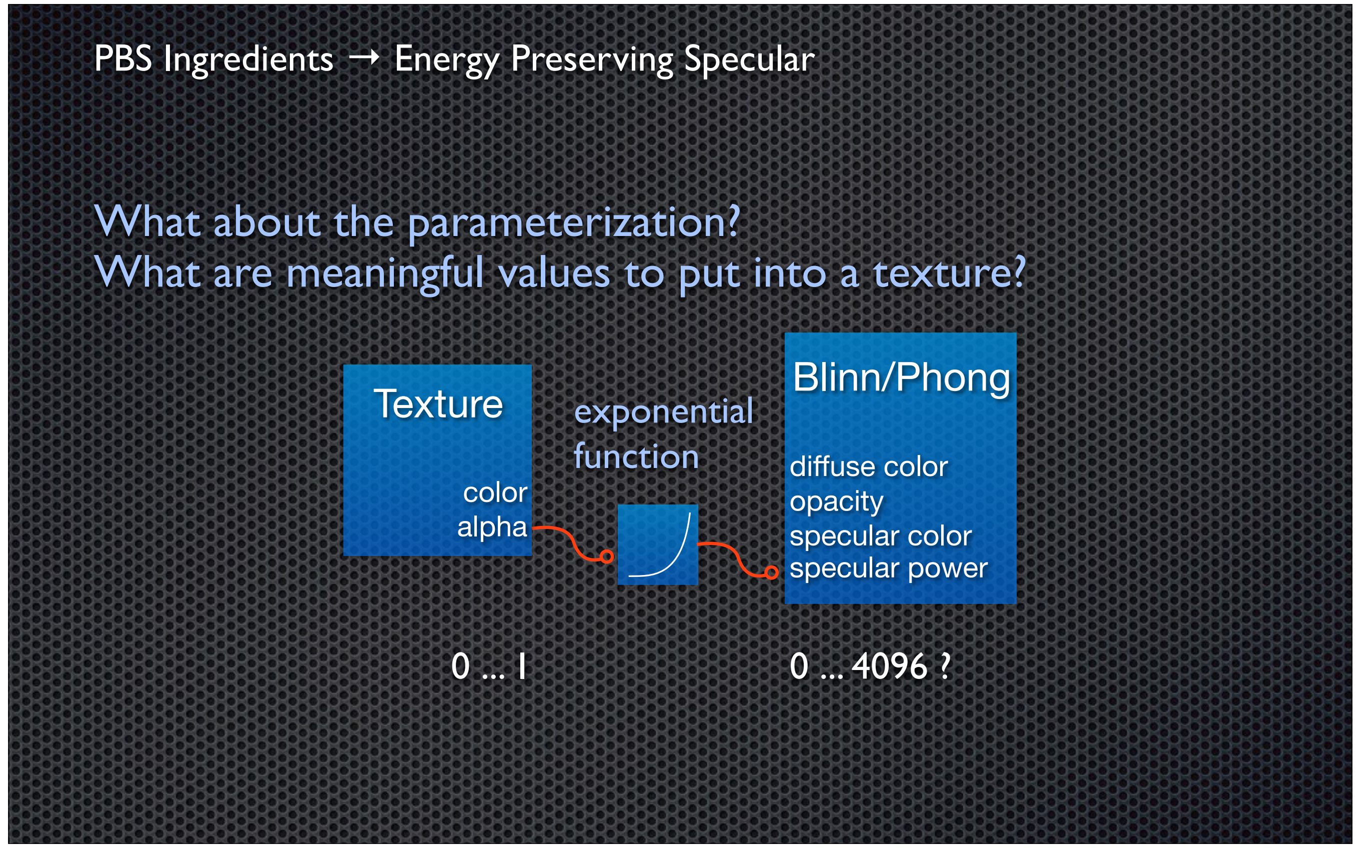
(Auditorium Question: Who does NOT know what specular power is?)




However things are not that easy, since the devil is in the detail.

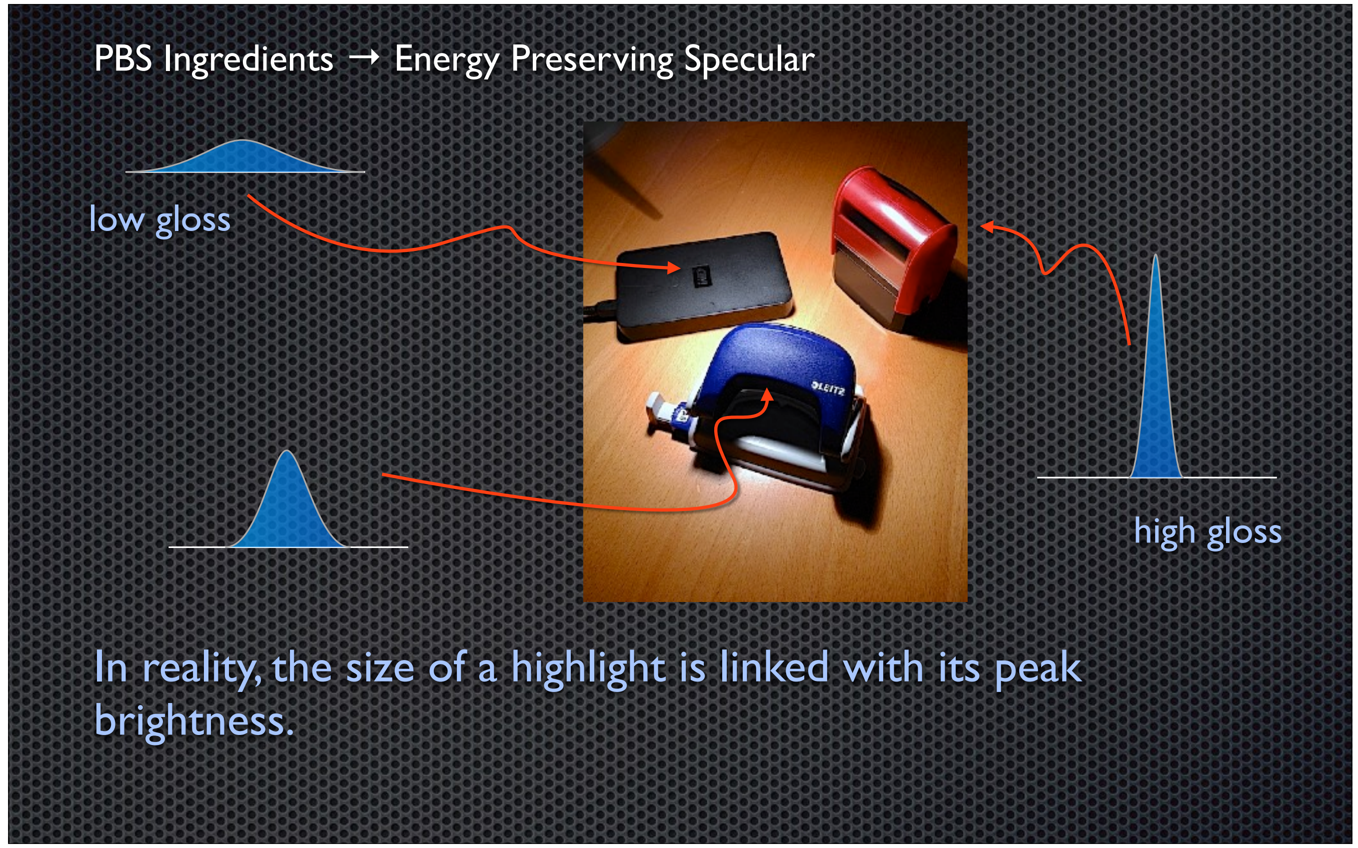
For a standard blinn/phong material, the numeric range for the specular power is unbounded. The higher the value, the smaller the highlight.
A perfect mirror would have, in theory, an infinite specular power.
An alpha or grey channel from a texture however is bounded between 0 and 1.

What is a useful parameterization to connect the two?




Turns out, that an exponential function is good mapping.
The minimum texture value of 0 (black) is mapped to a very low, but non-zero, specular exponent.
The maximum texture value of 1 (white) is mapped to a very high, but finite, specular exponent, in this example 4096.

(Auditorium Question: under this mapping, guess what specular exponent is mapped to the texture value 0.5? Answer: 64, the geometric mean of 1 and 4096, of course :-)

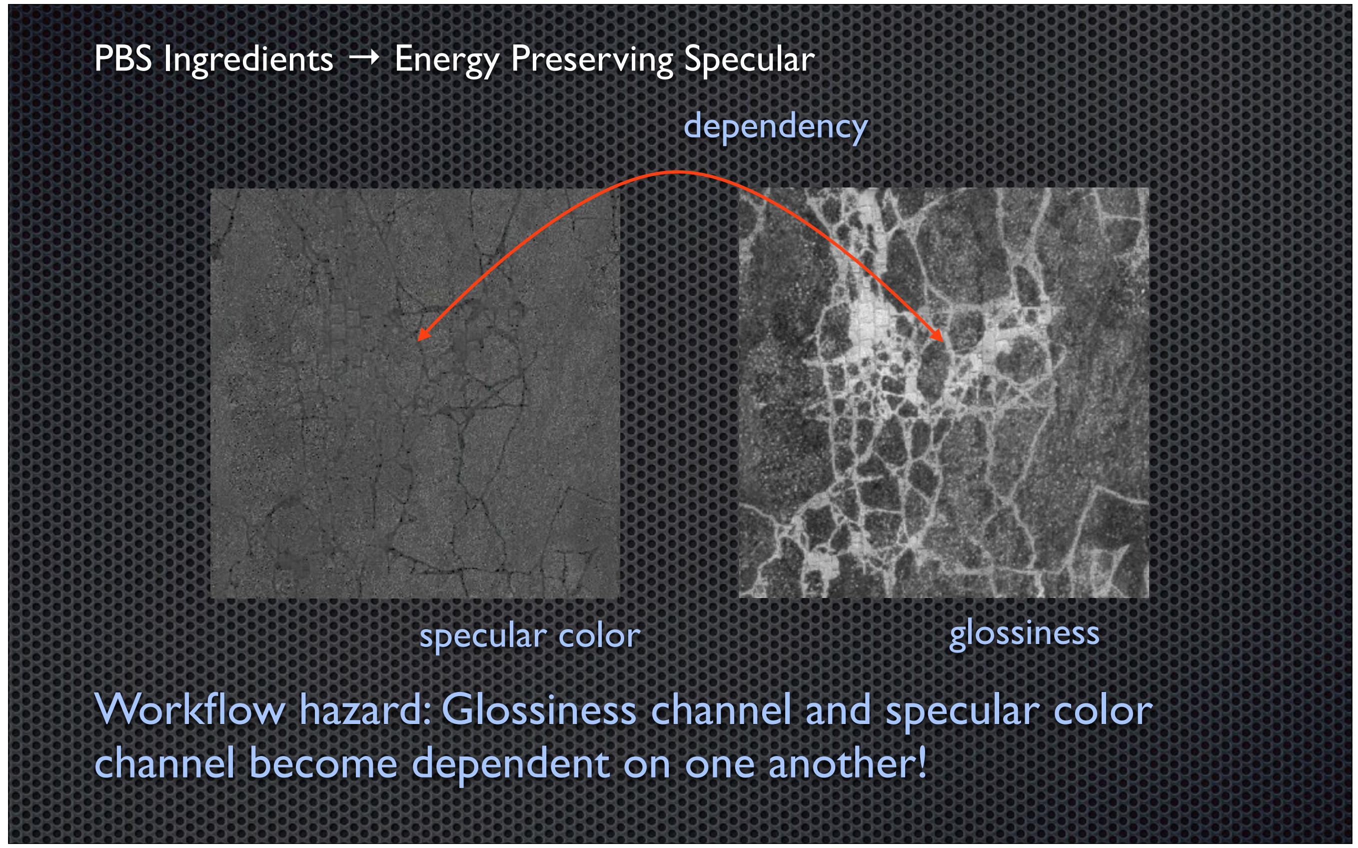



The energy reflected by a specular highlight is determined by both its size (width) and its peak brightness.

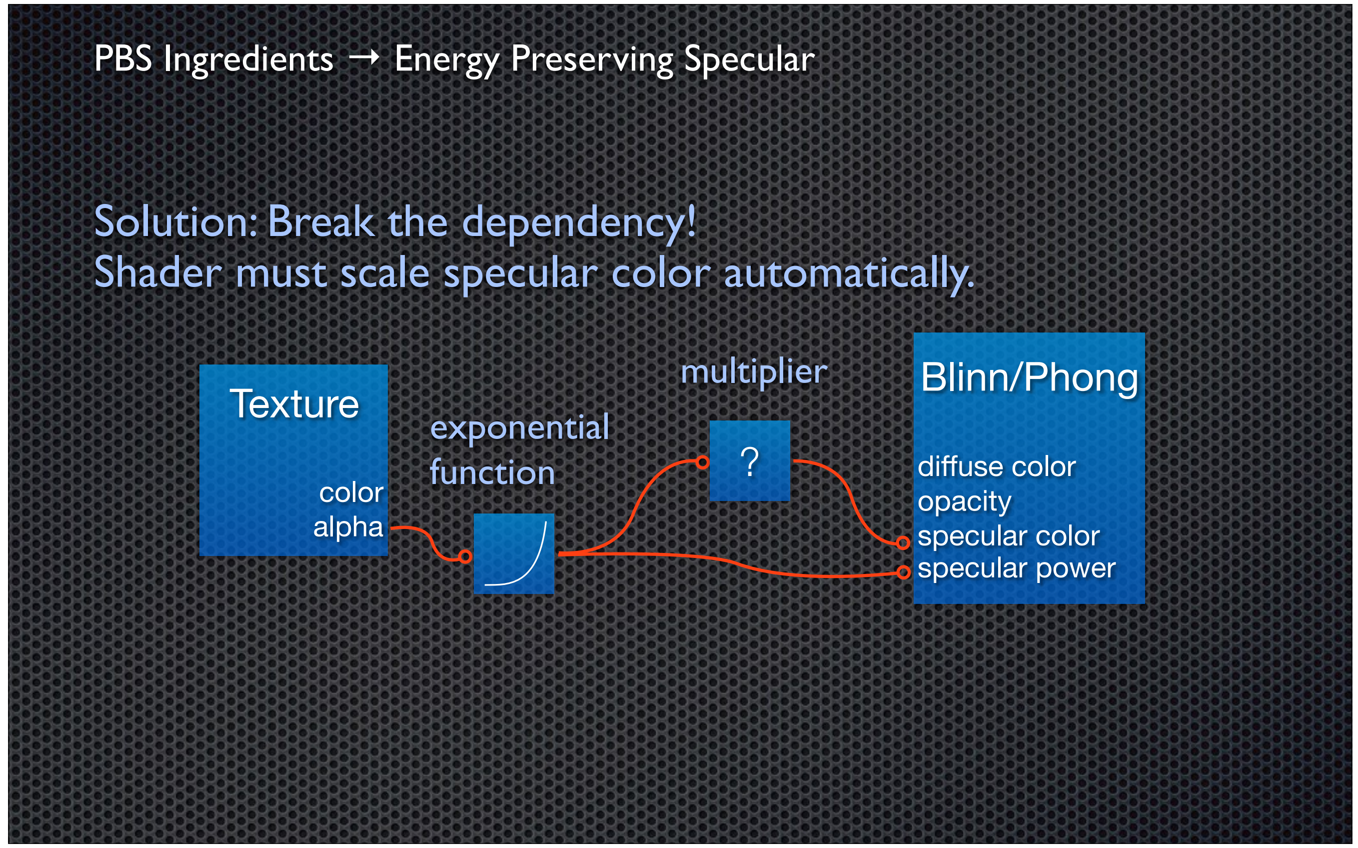
A broad highlight (on the black material, left) spreads the reflected light over a wide range of directions, therefore each individual direction only gets a small share. The profile curve is broad and shallow.

A sharp highlight (on the red material, right) concentrates the reflected light in a narrow range of directions, making it much more focused. The profile curve is tall and narrow.

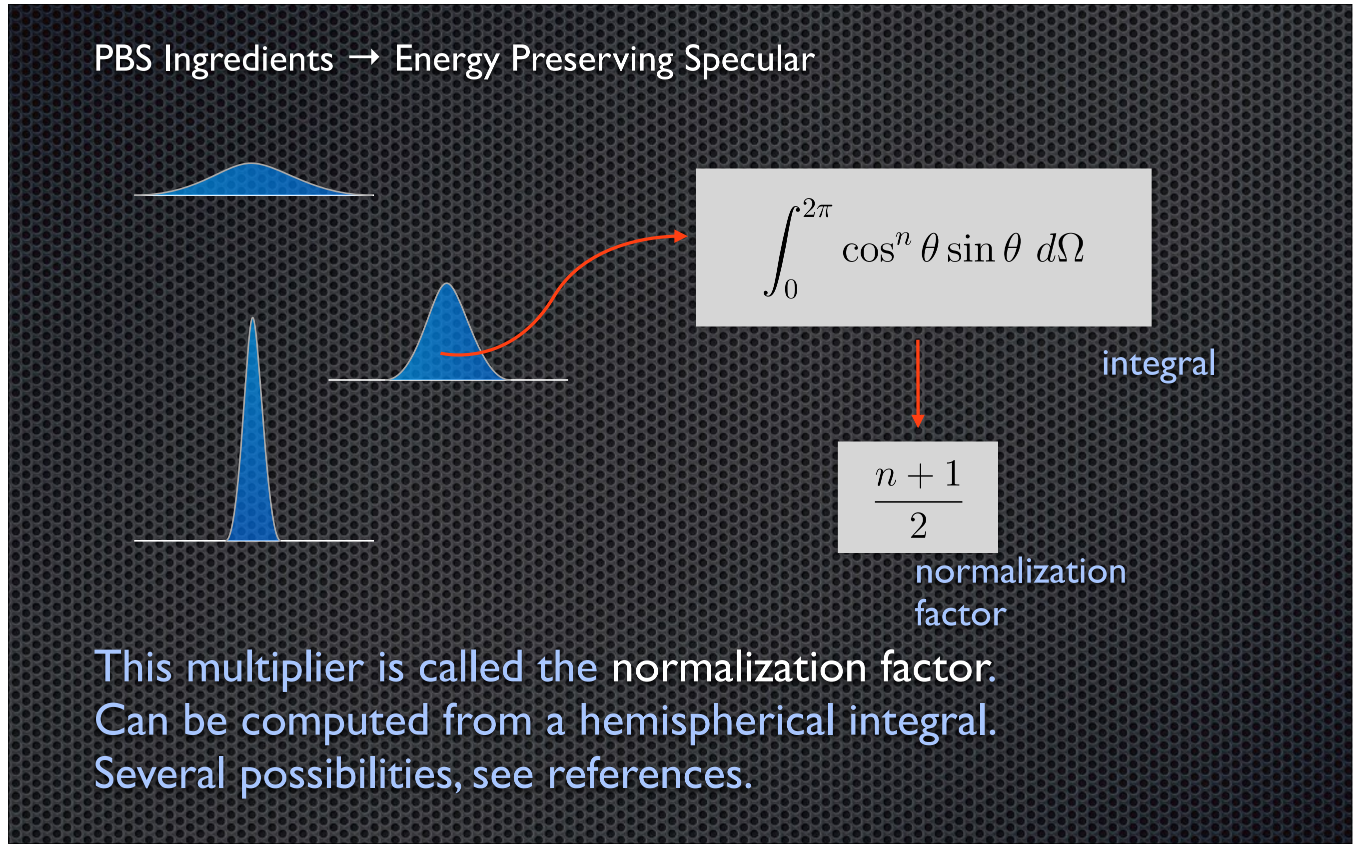
In both cases, the “area under the curve” should be about the same. In reality it is the “volume under a surface”, but that’s a mathematical detail. This is what is meant with “Energy Preserving Specular”.



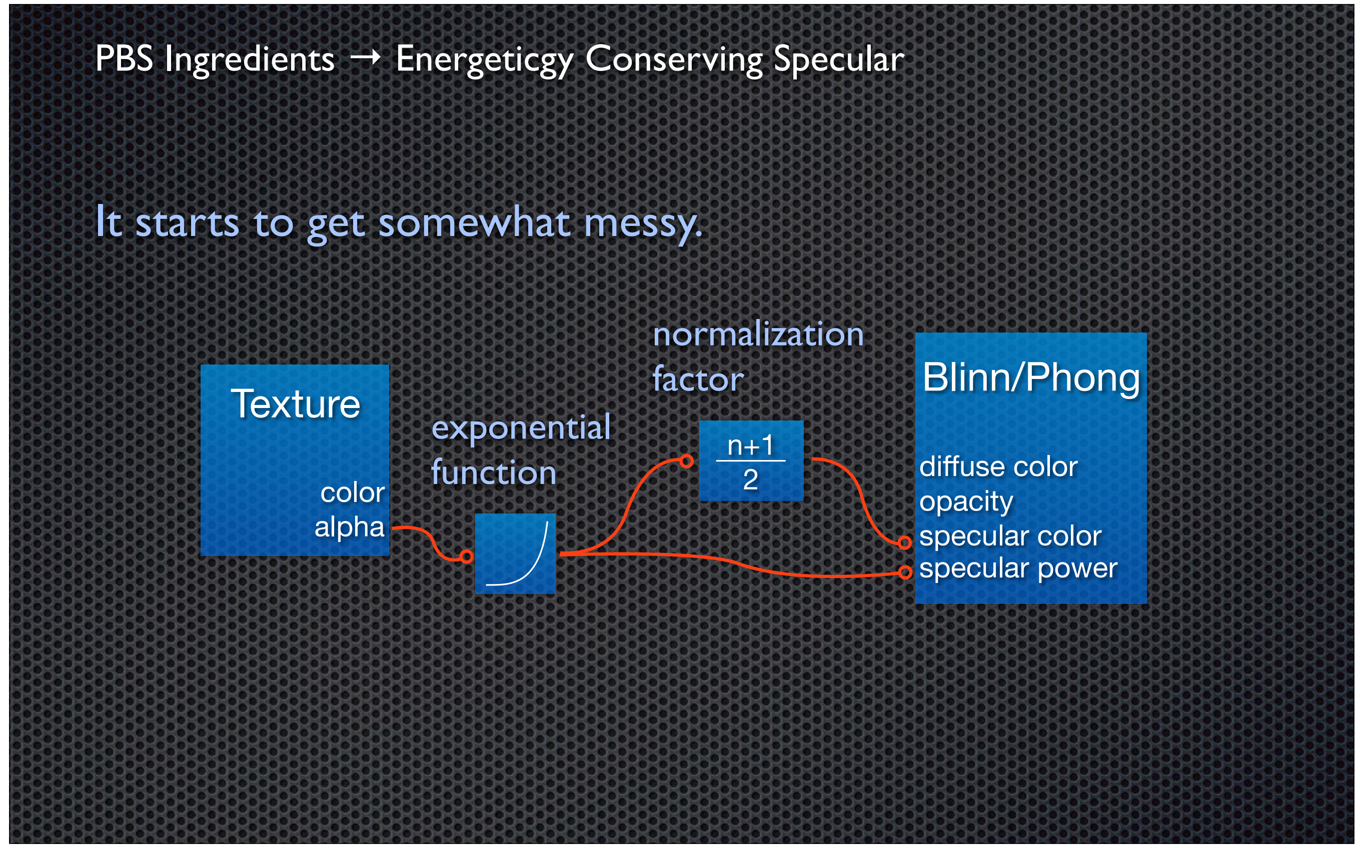

Now here we see a problem. The glossiness parameter is really linked/confounded with the specular intensity. Every time the glossiness is changed on the texture, a matching change must be done to the specular color texture. This is far from ideal from a workflow point of view.



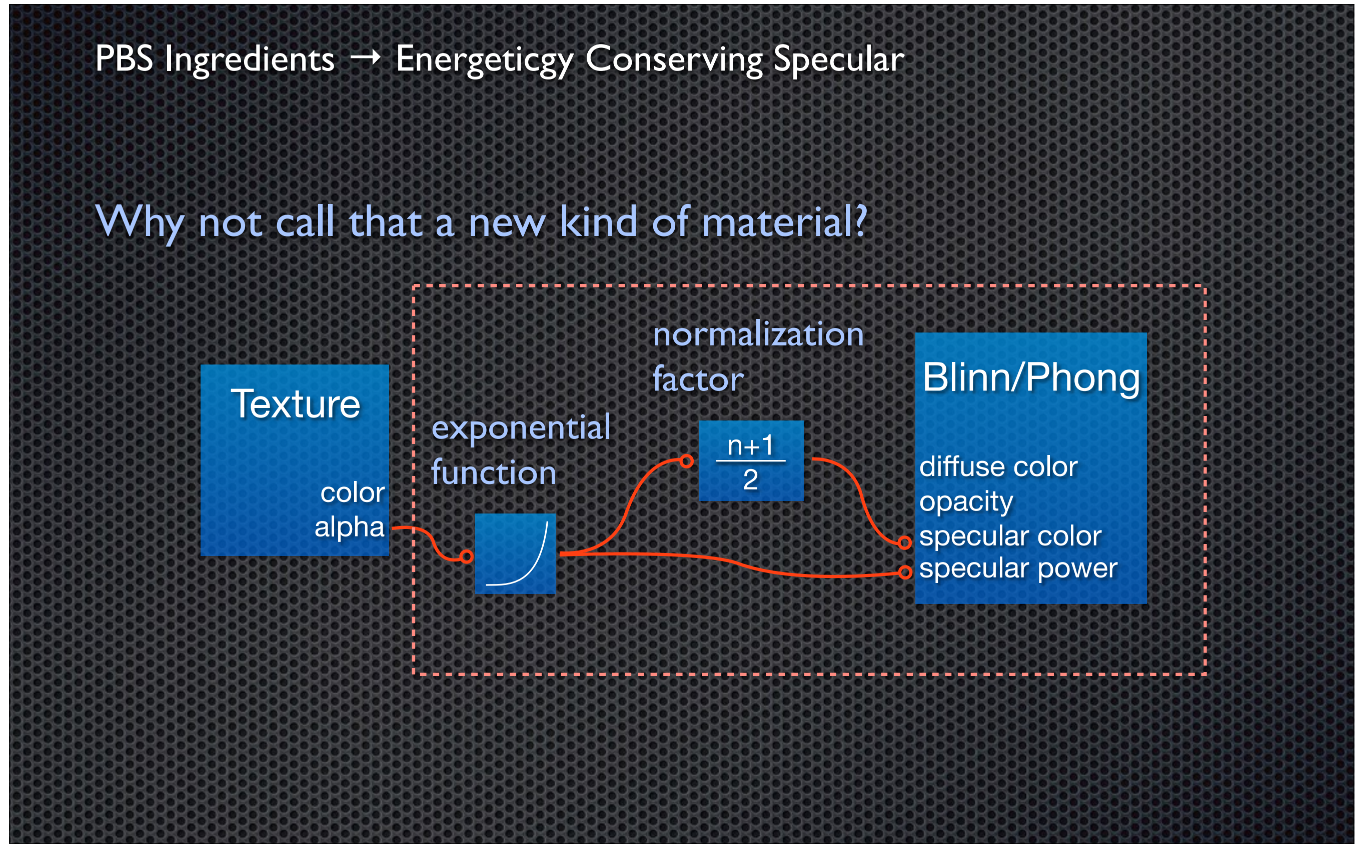

The solution is of course to make this change automatic to break the dependency.
We have to introduce a multiplier for the specular color that is dependent on the specular power/exponent.



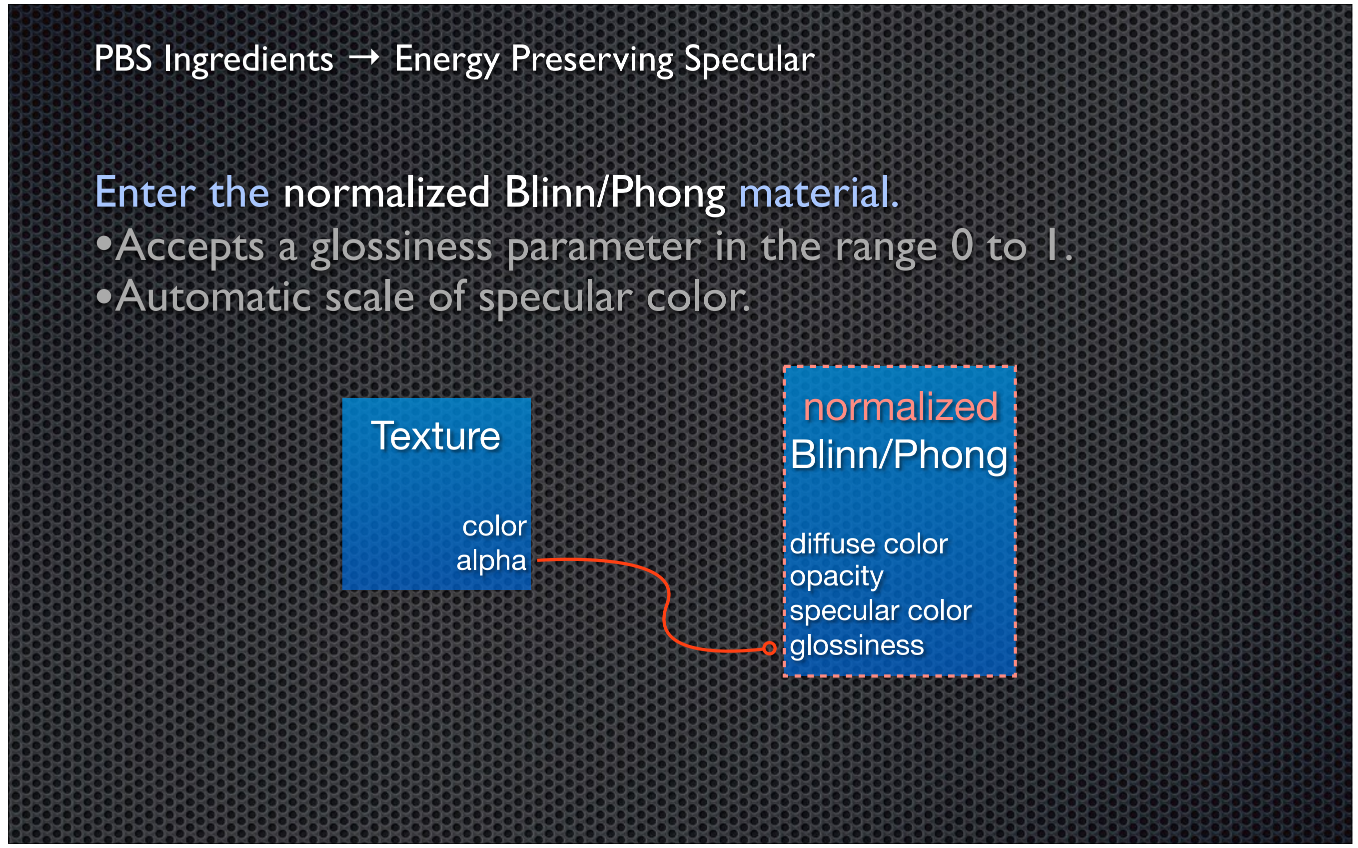

This multiplier is called a normalization factor.




Now it starts to get somewhat messy.



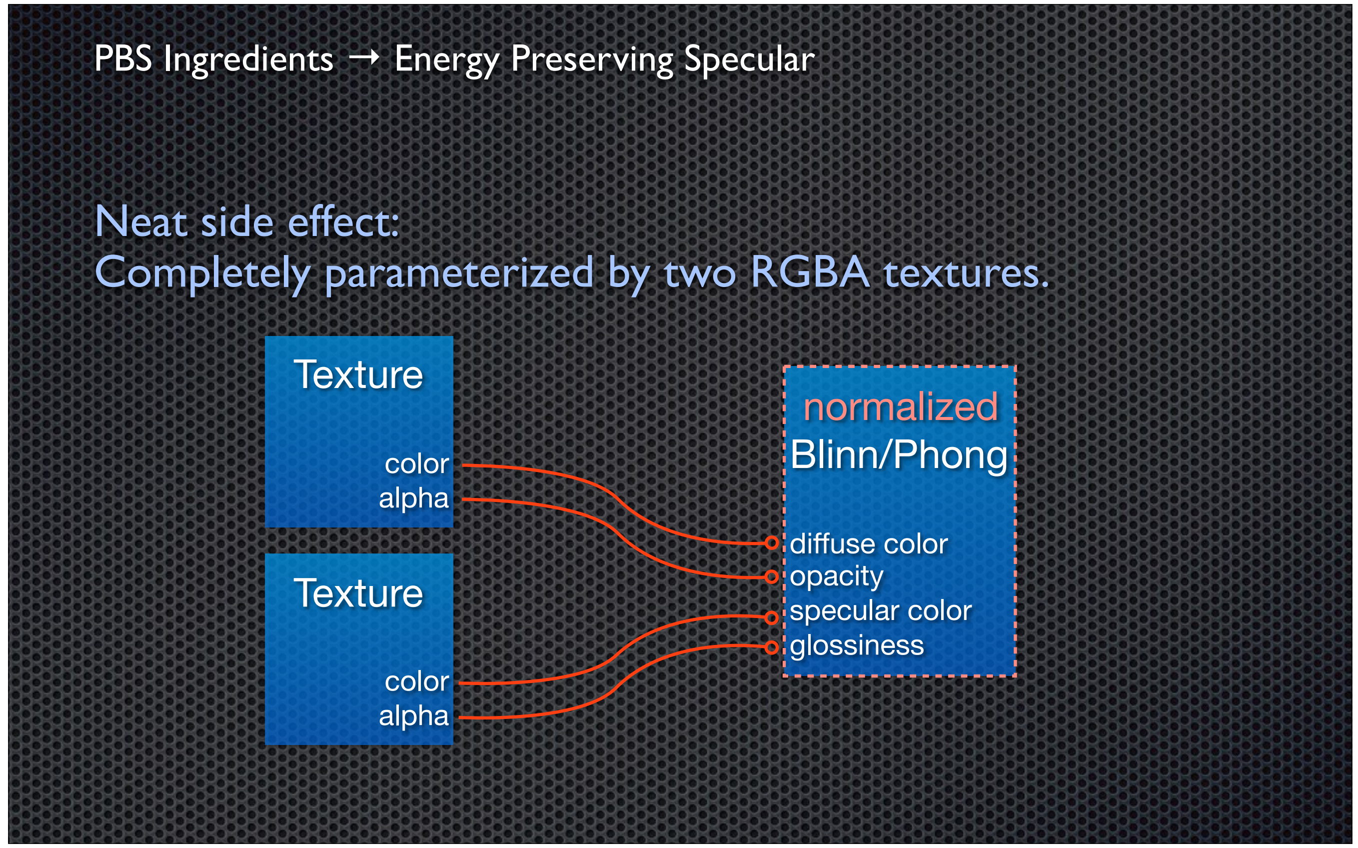

Why don’t we make the graph that we created into a new kind of material?




This would be called the *normalized* Blinn/Phong material.

It is called *normalized*, because we can control both the size and strength of the specular highlight with a single value, in a manner that preserves overall energy.

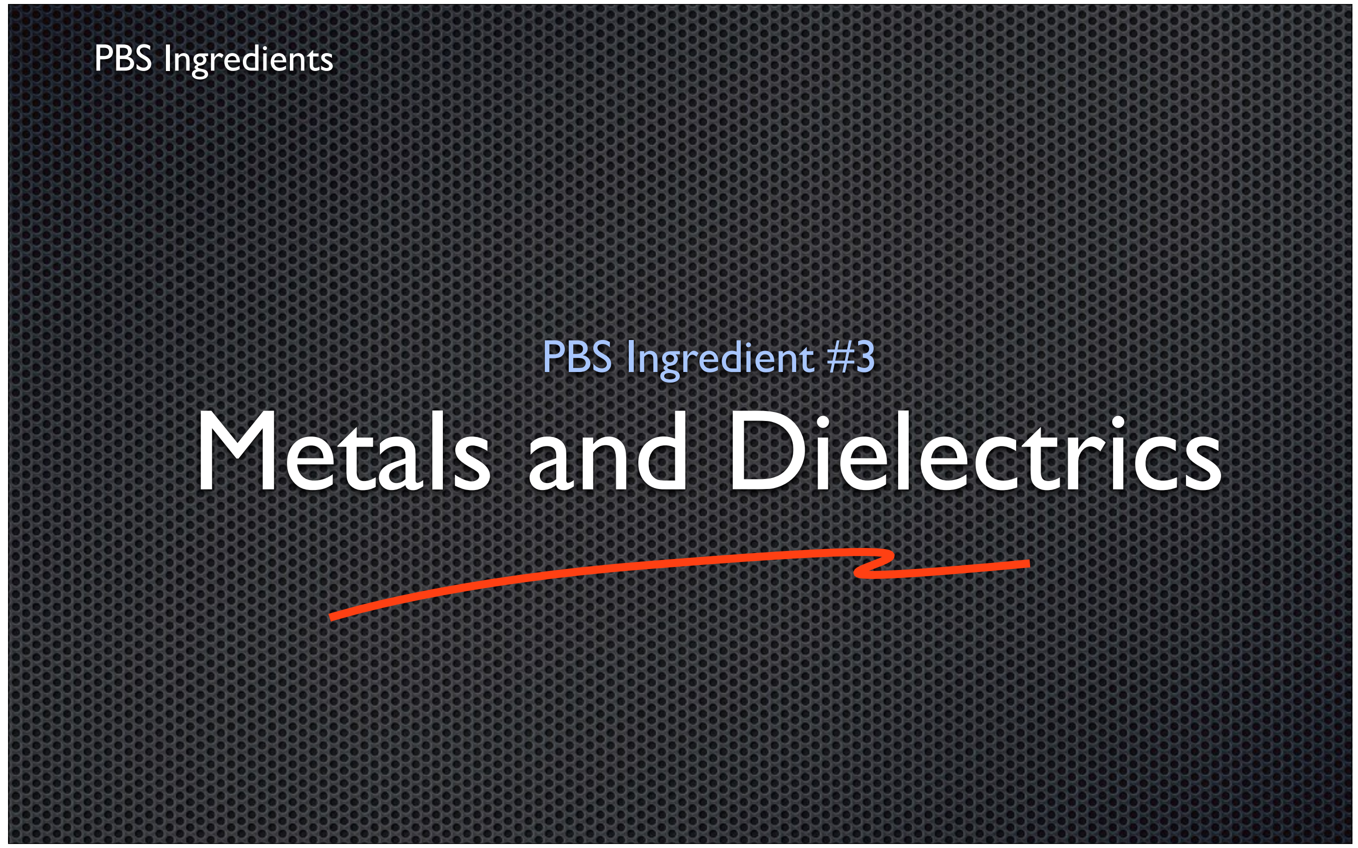
This new control is called the *glossiness*, and it is physically connected to the roughness of the surface.




The 4 material parameters map nicely to 2 RGBA textures.
All material parameters can be easily textured, which means that many different materials can be put onto the same texture, which helps batching.

## Further reading

- Blogs
  http://www.rorydriscoll.com/2009/01/25/energy-conservation-in-games/
  http://seblagarde.wordpress.com/2011/08/17/hello-world/
  http://www.thetenthplanet.de/archives/255
- Unreal
  http://udn.epicgames.com/Three/ImageBasedReflections.html
- CryEngine
  http://freesdk.crydev.net/display/SDKDOC4/Shading




The third ingredient of physically based shading is a way to incorporate both metals and dielectrics into the framework.

PBS Ingredient #3

# Metals and


Di-electrics is just another word for non-metals.


This basically reduces to the ability to drive the material from the specular color only (the diffuse color is then set to zero).

It is surprising to learn that most shaders/materials break down in this case because many times the diffuse color is used to drive other aspects of the material, for instance the ambient color.


In real time rendering, it is important to be able to "batch things up", for performance.
We want to be able to put metals and non-metals onto the same texture, rendered with the same material.

PBS Ingredients → Metals and Dielectrics

Assignment:
What is the difference between metals and non-metals?

Rhetorical auditorium question:
What do you think is the difference between metals and non-metals?

PBS Ingredients → Metals and Dielectrics

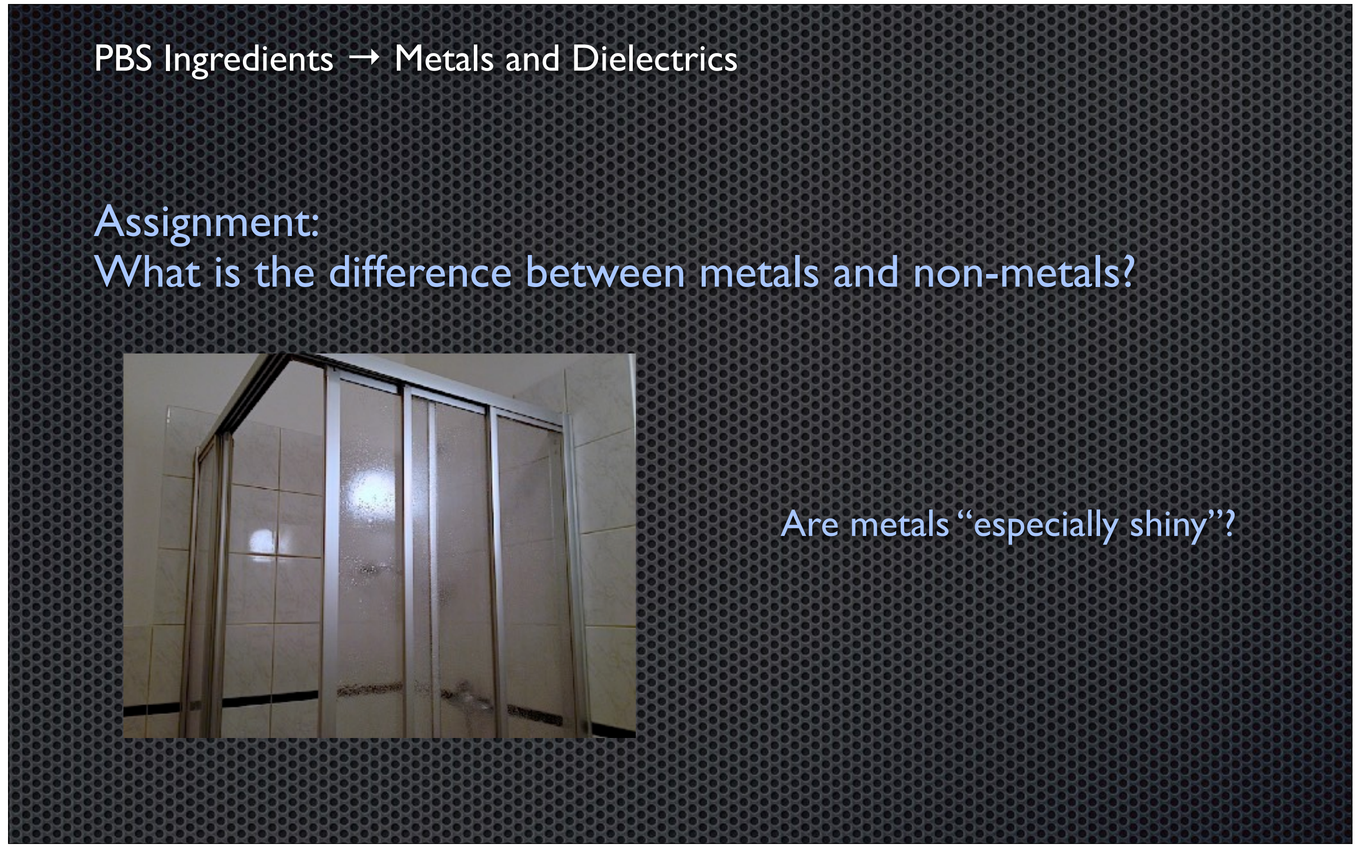
Assignment:
What is the difference between metals and non-metals?

Are metals “somehow grey”?

One typical answer is that metals are “somehow grey”.

Besides the fact that there are colored metals like gold and copper, even grey metals are not really “grey”. As can be seen on this photo, the metal appears in many colors, just as it is reflecting the environment. The metal is just being reflective!

So impression that metals are grey is based on the fact that, most of the time, the average color of the environment is grey.




Another typical answer is that metals are especially shiny. Probably this opinion is based on confusing shininess for reflectivity.

As we have seen on the previous slide, metals seem to be somehow reflective, but they do not need to be shiny. You can see in this photo how the bathroom tiles, which are non-metals, are very shiny (smooth surface), while the bars from the aluminum frame are very dull (rough surface).

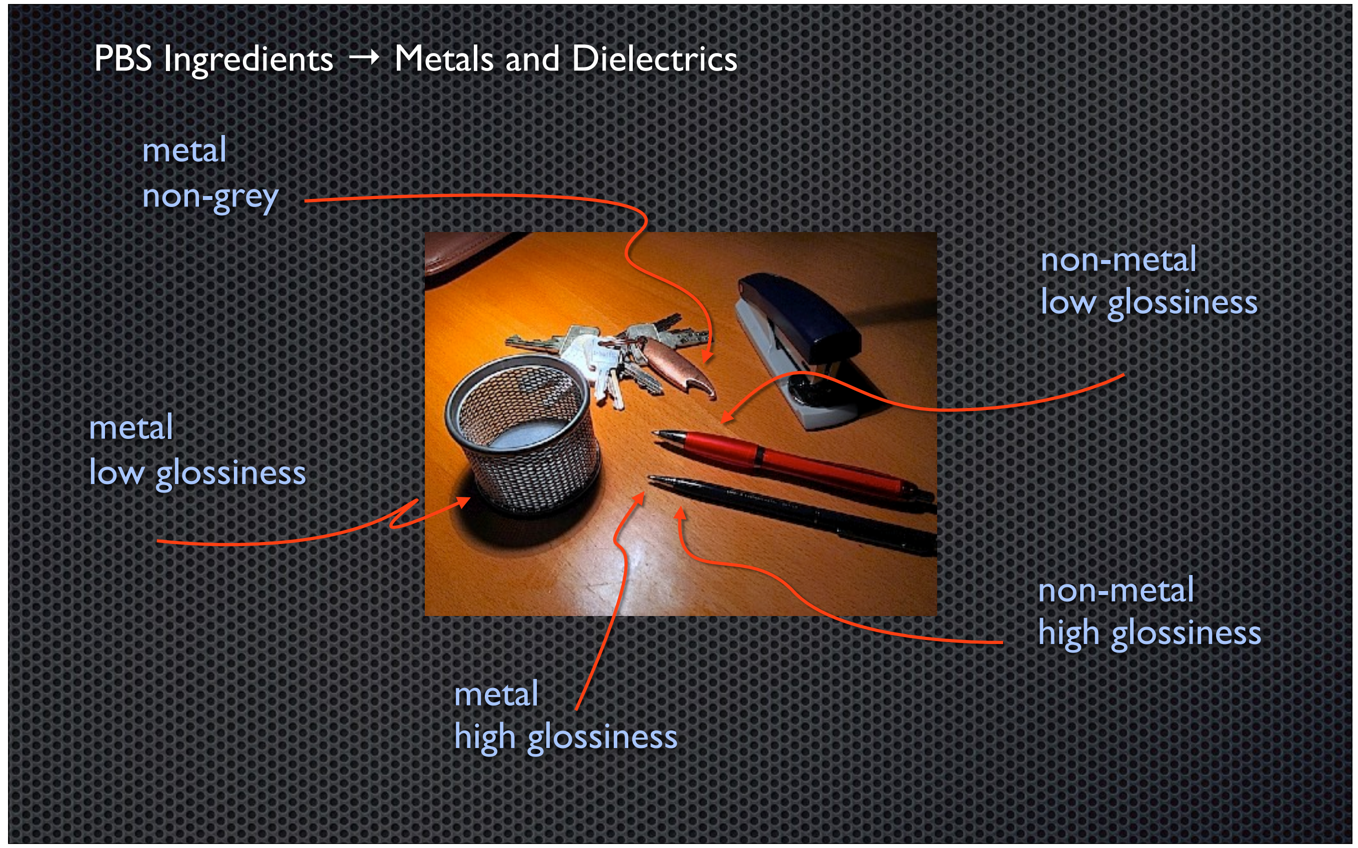
## PBS Ingredients → Metals and Dielectrics

### Assignment:
### What is the difference between metals and non-metals?

But surely they have
“lots of specular”?

There is also the quite common opinion that metals have “lots of specular”.

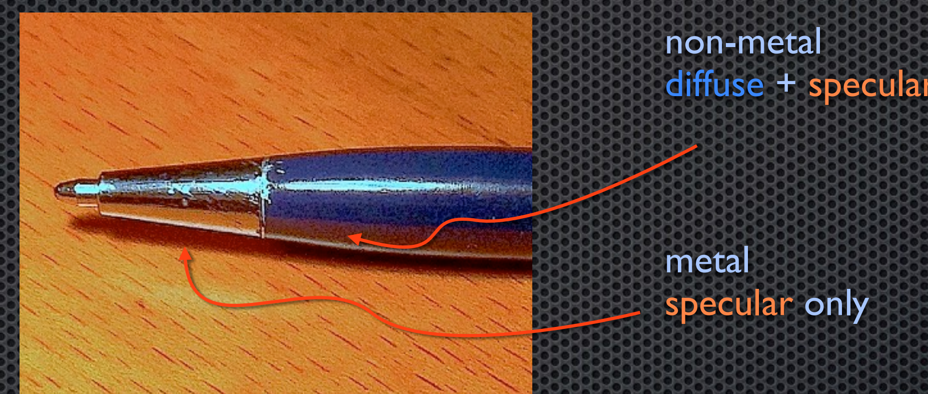
As can be seen in the photo, specularity as such is not reserved for metals. Clearly, non-metals can be “specular” too. But if “specularity” is held as a synonym for “reflectivity”, then we are onto something.




So there must be something that differentiates metals and non-metals, and it is not

- the color
- the shininess (glossiness)

Metals are pure reflection,
characterized by the **absence** of a diffuse component.

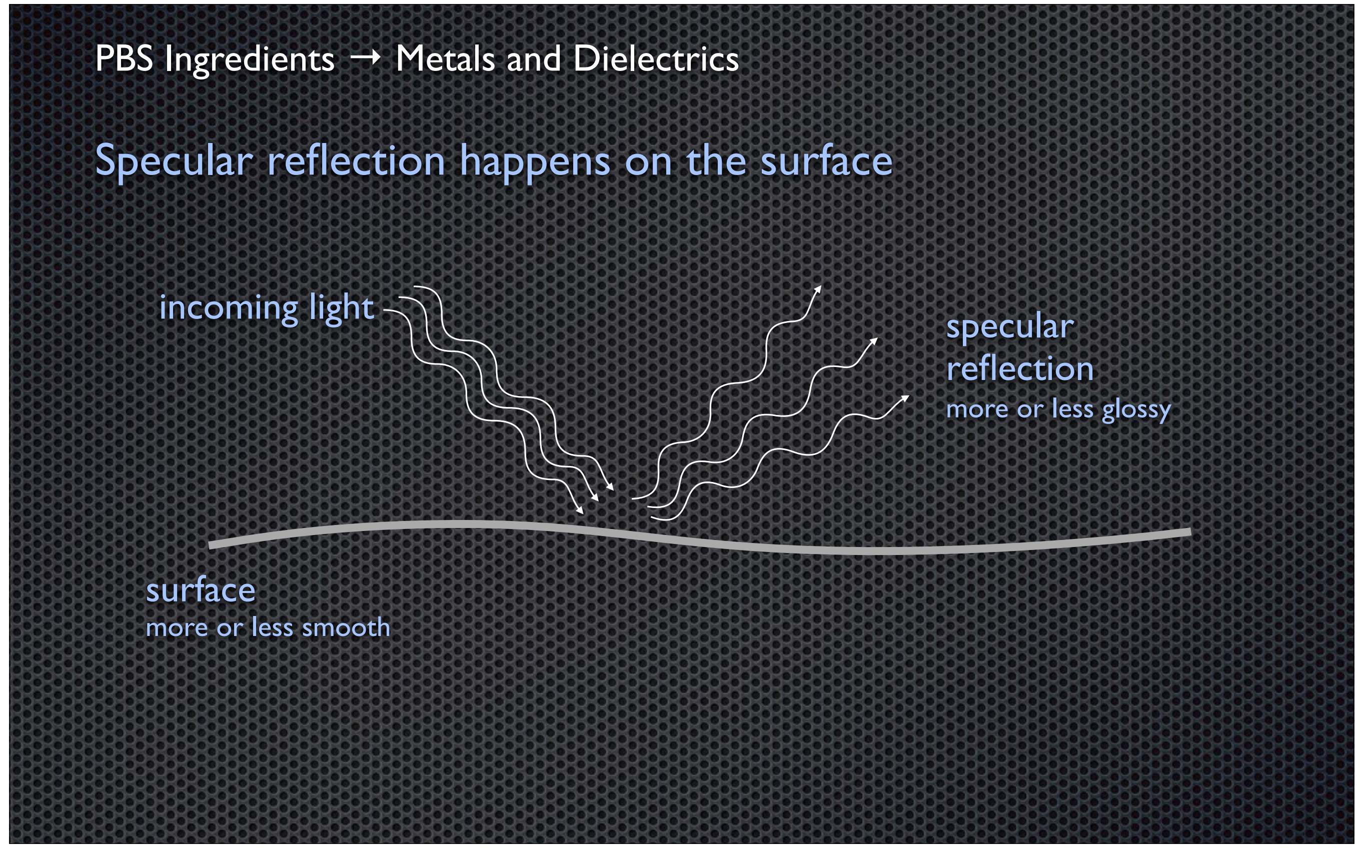



The defining characteristic of a metal is the total absence of any “diffuse” component.

That’s right, the “diffuse color” of a metal is identically zero.
Pitch black zero.

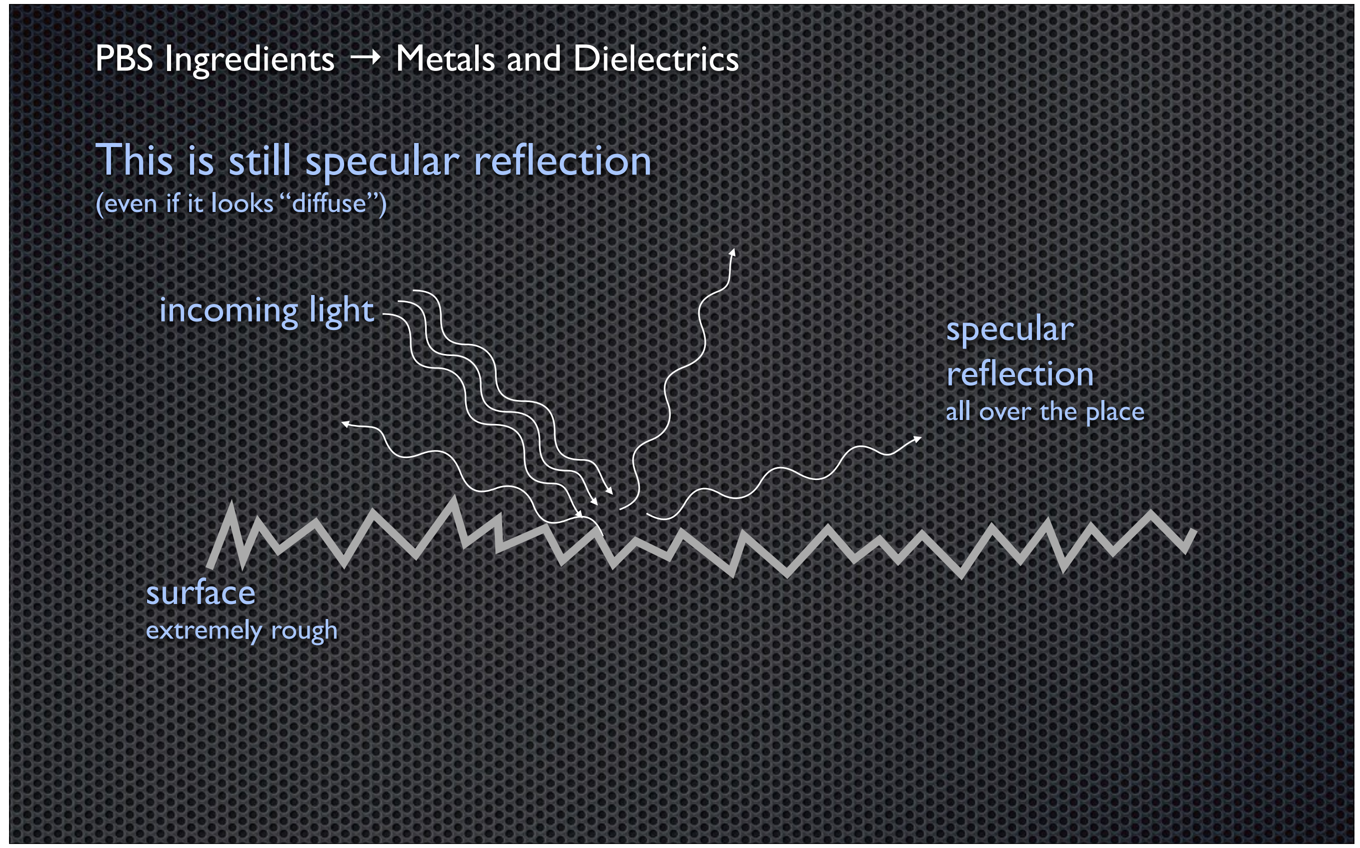
The photo is meant to demonstrate this effect. The reflection of the environment is seen on both the plastic body of the ball pen, and the metallic tip. Both have about the same glossiness (the surface is very smooth).

But while the plastic body shows a mixture between its own color (“diffuse color”, in this case a dark blue) and the specular reflection, the metallic tip does not seem to have a color of its own. It is only reflection, and nothing else.

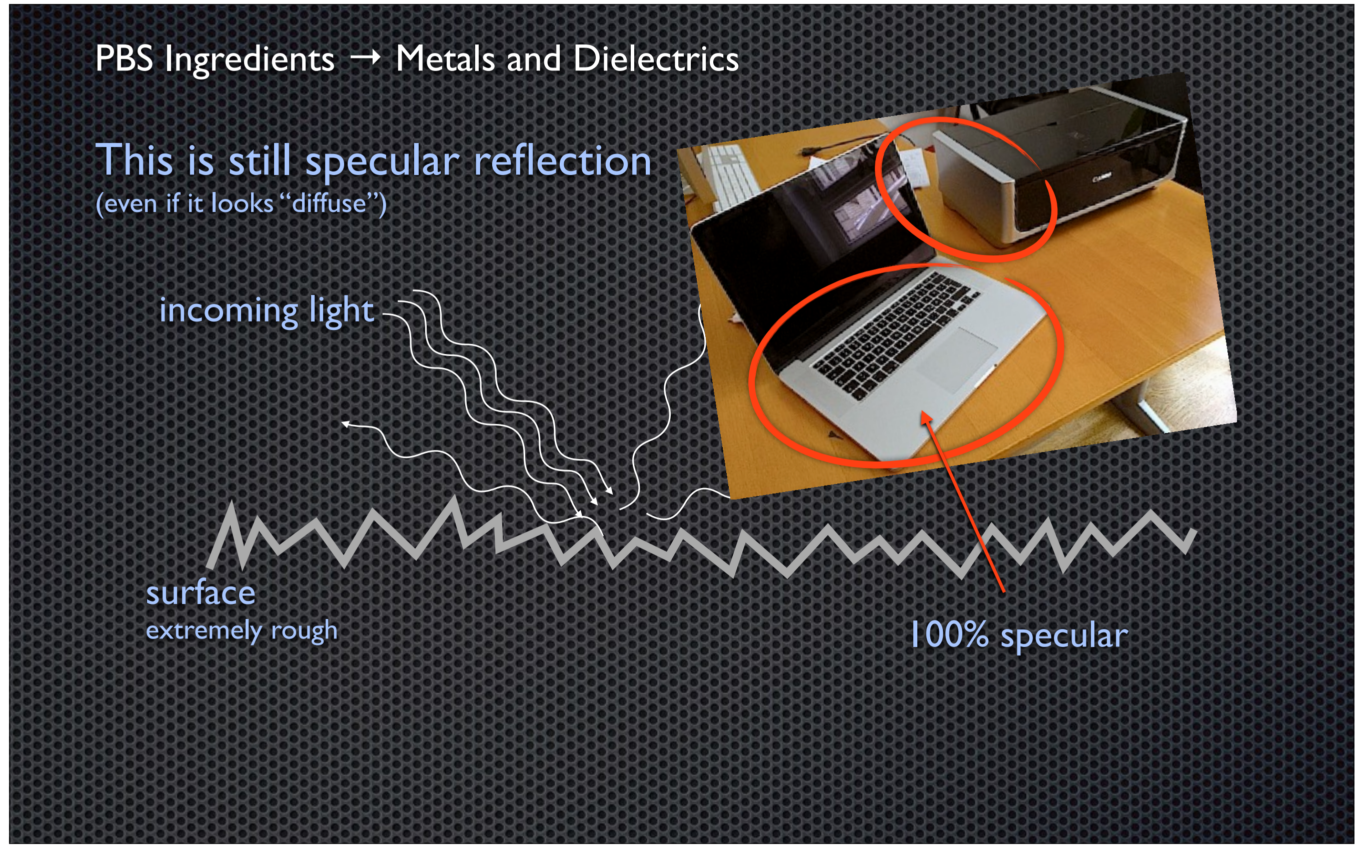



I would like to explain the physics behind this phenomenon a bit more in detail.

As you all probably know, the light is reflected on a surface and this is called the “specular” reflection. If the surface is smooth, than the light is reflected in a small cone around the perfect mirror reflection.



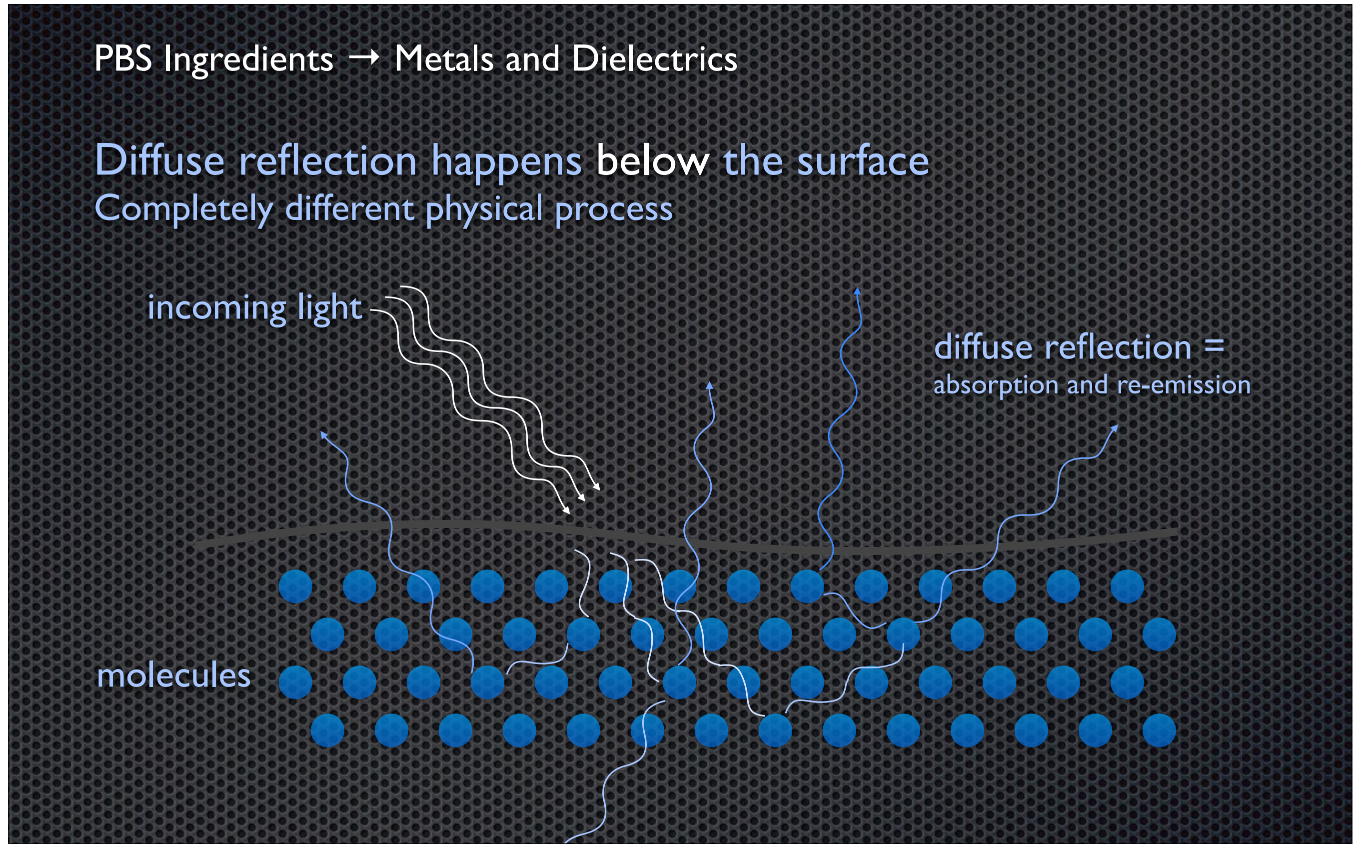

If the surface is very rough, then the light will be reflected in almost any direction. However, this is still specular reflection. It is not diffuse reflection in the technical sense.




As a real life example, think of very rough, brushed metal.

The body of my laptop, for instance, looks very diffuse, however it is still 100% specular (since it is a metal). It's just that the surface is very rough.

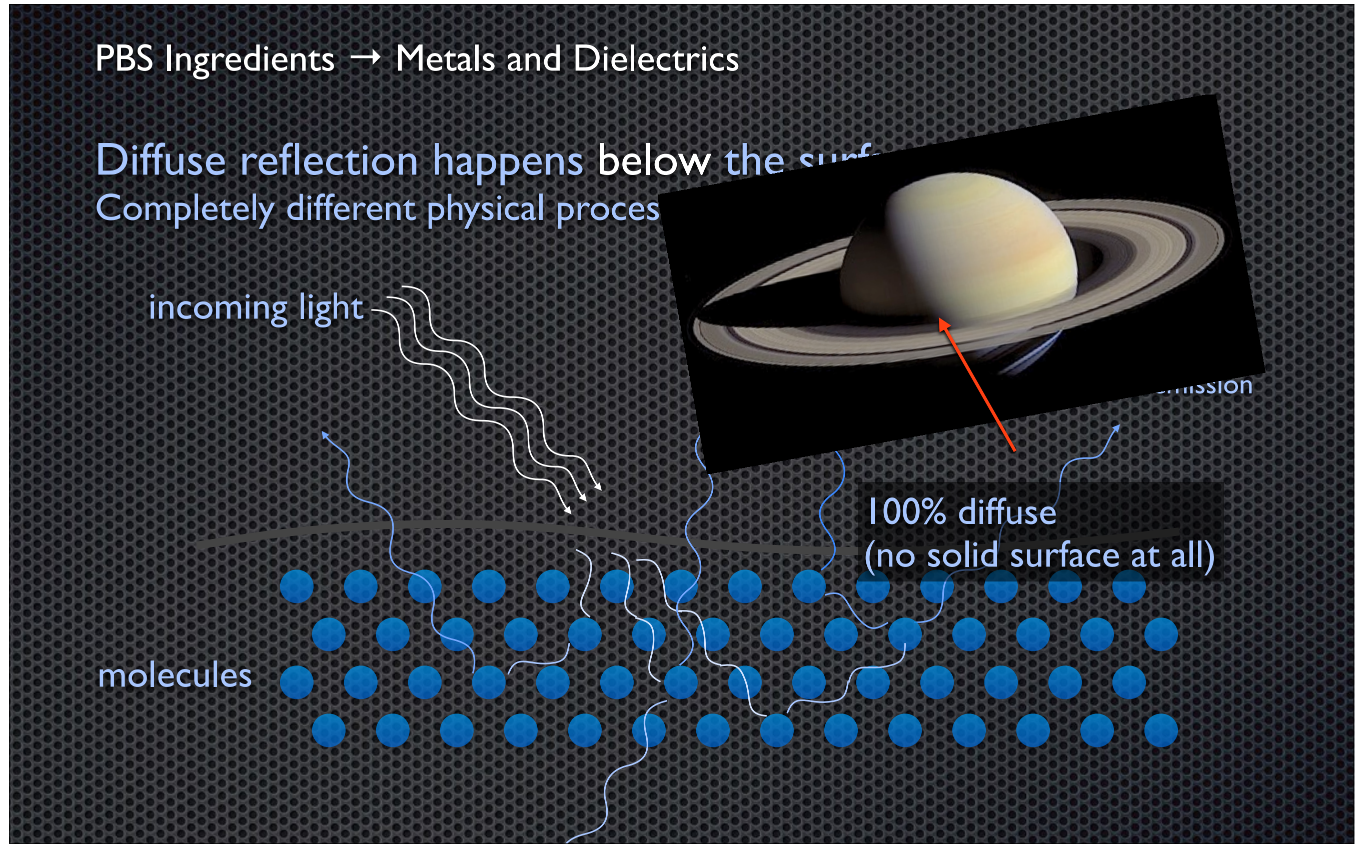
The same seems to holds for the side of the printer, but in retrospect, I'm not sure if that really was metal or some coated plastics.




Diffuse reflection, on the other hand, is an idealization of the process that happens below the surface.

Here, the light penetrates the surface into the body of the material, is absorbed and re-emitted a number of times, each time in a random direction. Along the way, it acquires the “diffuse color”.

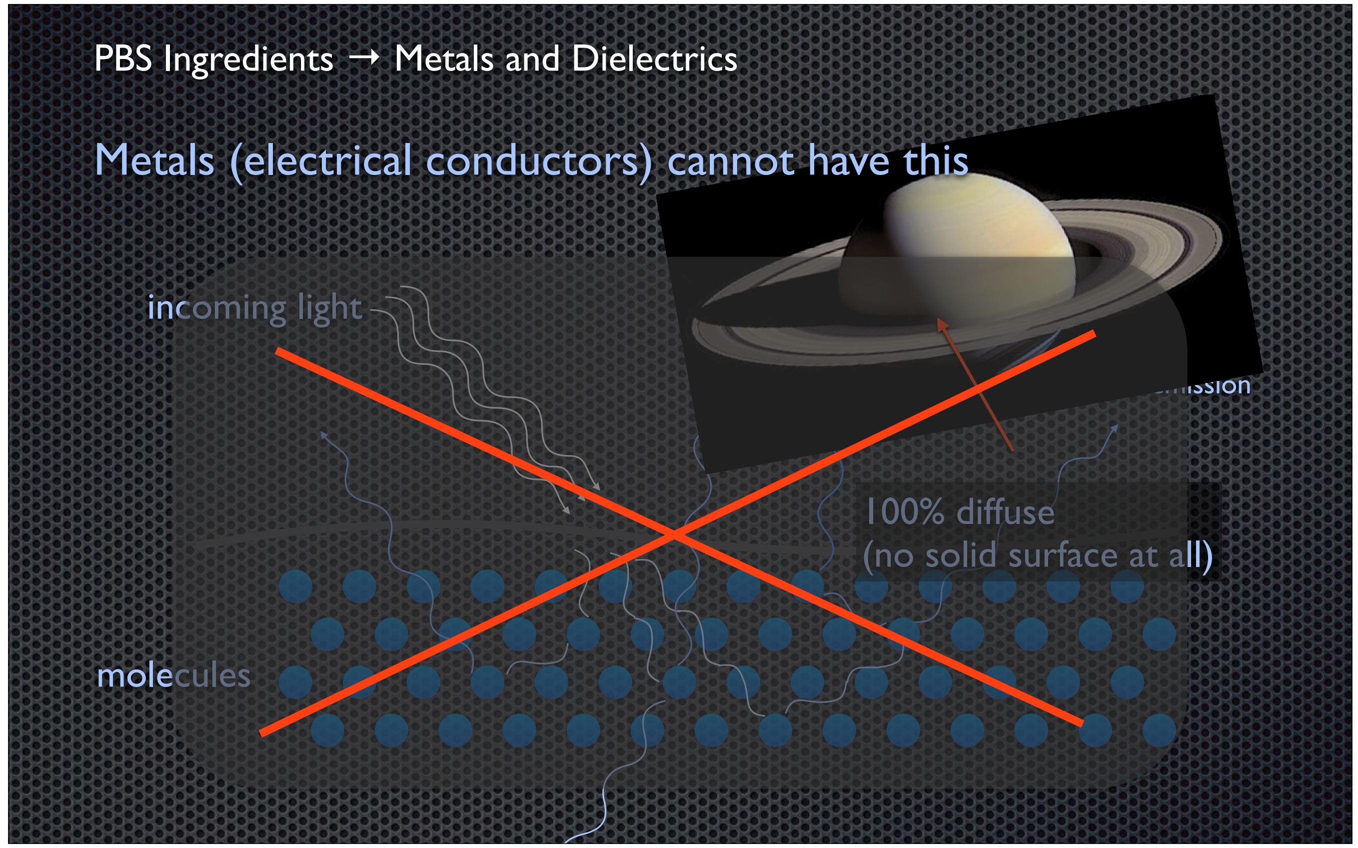
It can be thought of as “subsurface scattering on a microscopic scale”.




As a real life example, think of the atmosphere of a gas giant.

A gas giant does not even have a solid surface.So when viewed from a sufficient distance, this would be an example of 100% diffuse reflection.

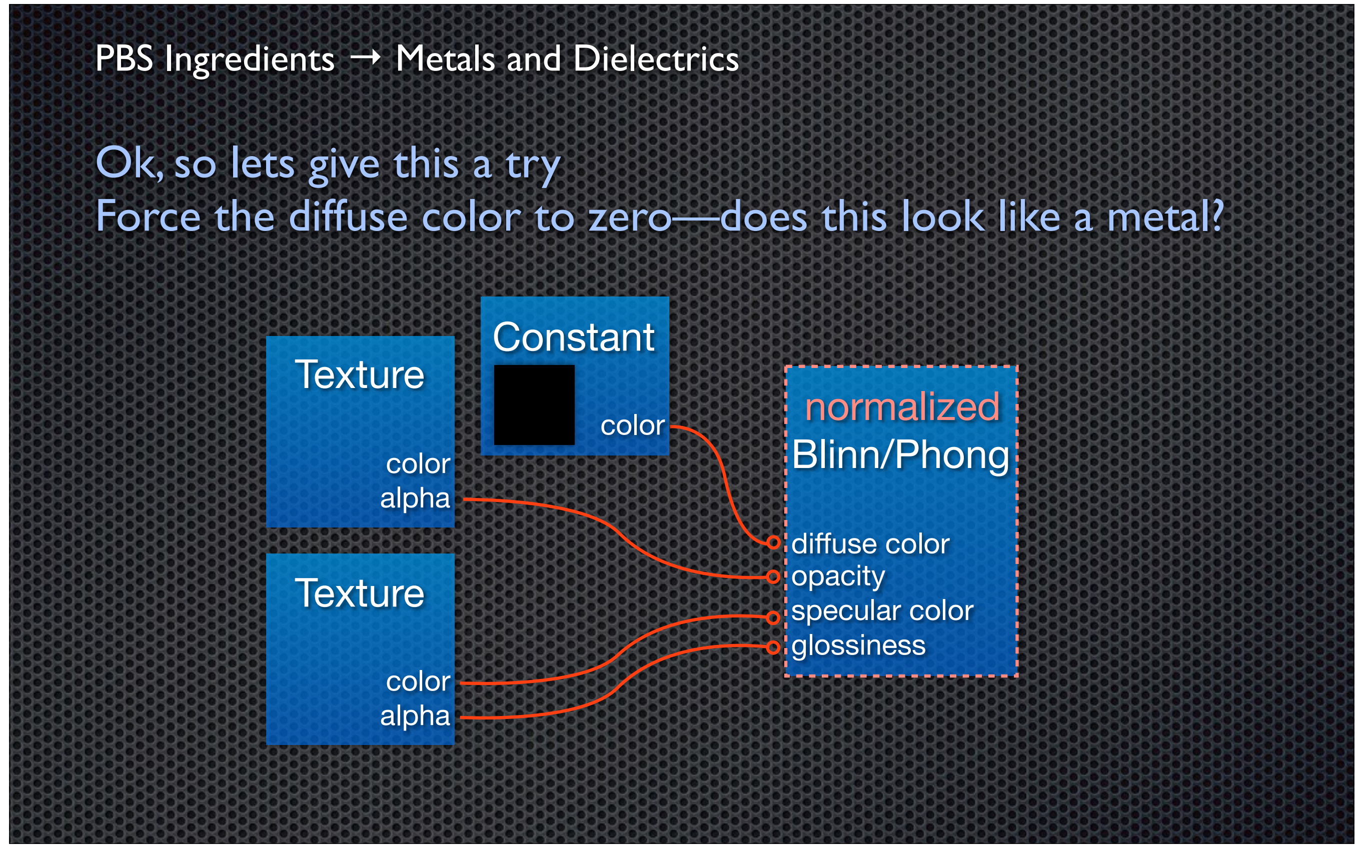
Besides, see the sharp border, another example for linear lighting.



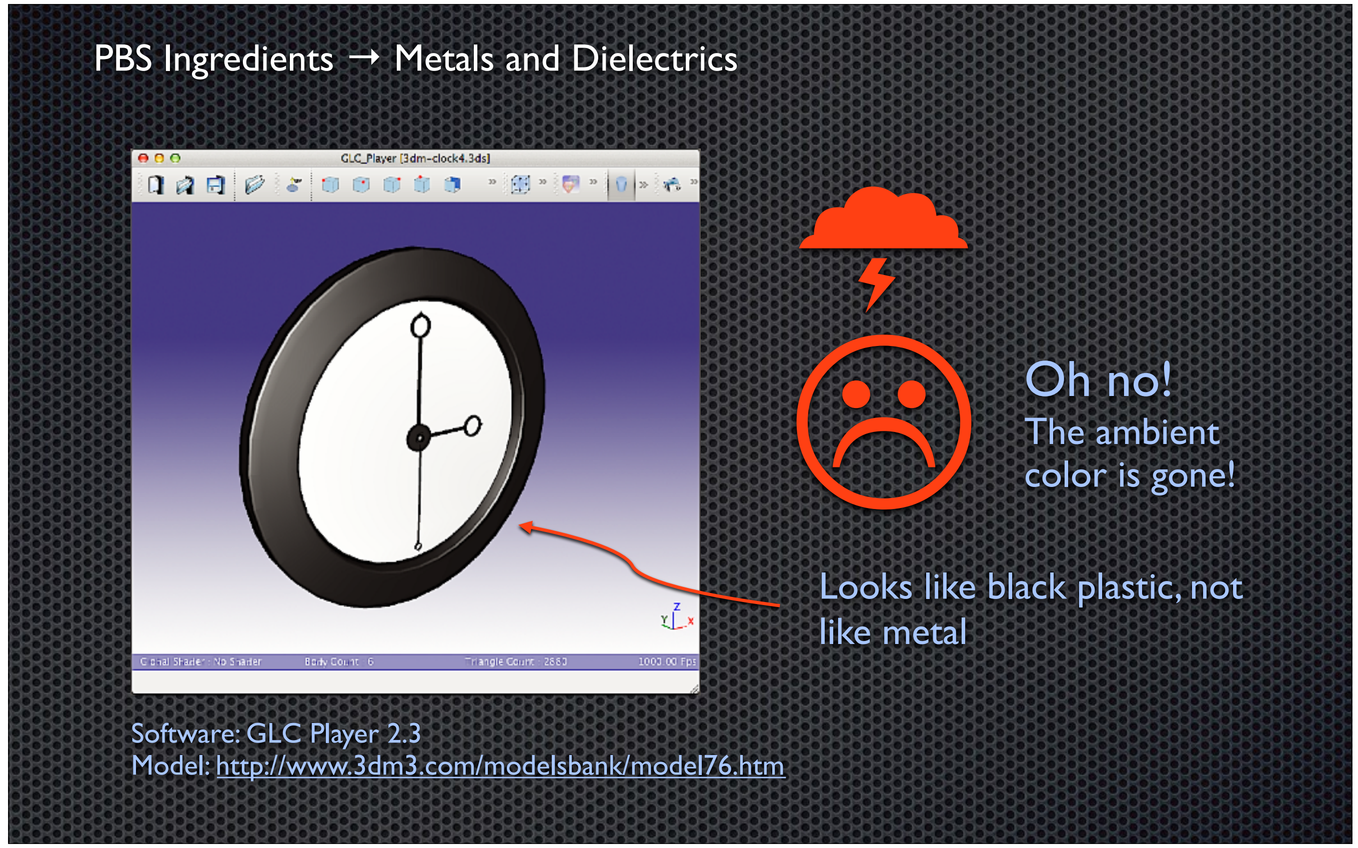

Now as for metals, they cannot have a diffuse reflection, because they are electrical conductors.

It is not possible for light—which is an electromagnetic wave—to penetrate an electrical conductor, for the same reason it is not possible to have radio communication with a submarine submerged in salt water: The wave will just be short-circuited over the course of a few wavelengths (and for light, that is a very small distance).

So for a metal, it's diffuse color must be zero and only it's specular color can be non-zero.

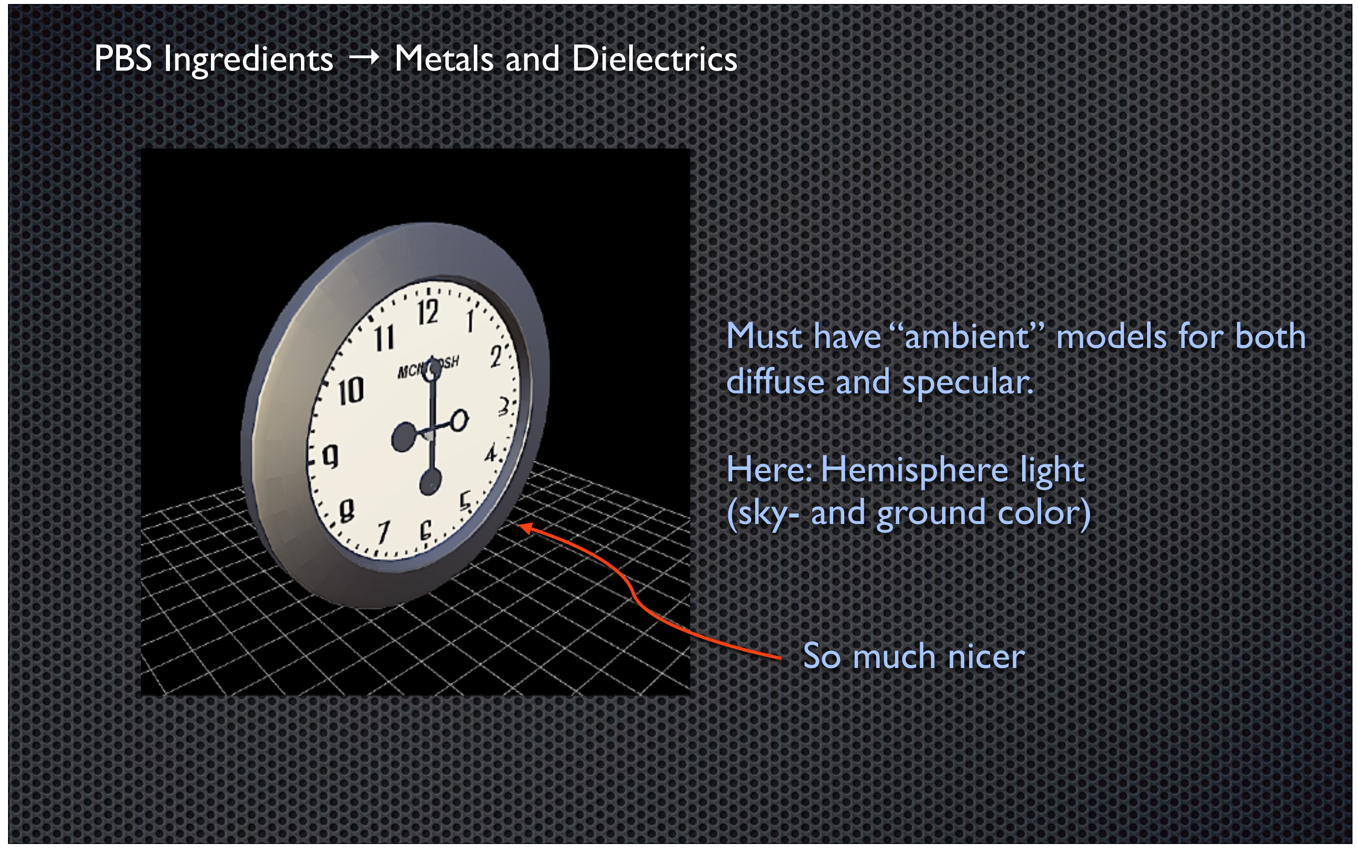



Armed with this knowledge lets try what happens when we force the diffuse color to zero. Does the result look like metal?




I have found a publicly available 3DS model of a clock, that supposedly has a metal material for rim and handles, together with a zero diffuse color. Let's load this into a common model viewer.

The result is rather disappointing. In the screenshot you can see that all ambient illumination is gone. This looks like black plastic, but not like metal! The problem is here, virtually all shaders/materials link the ambient color to the diffuse color.

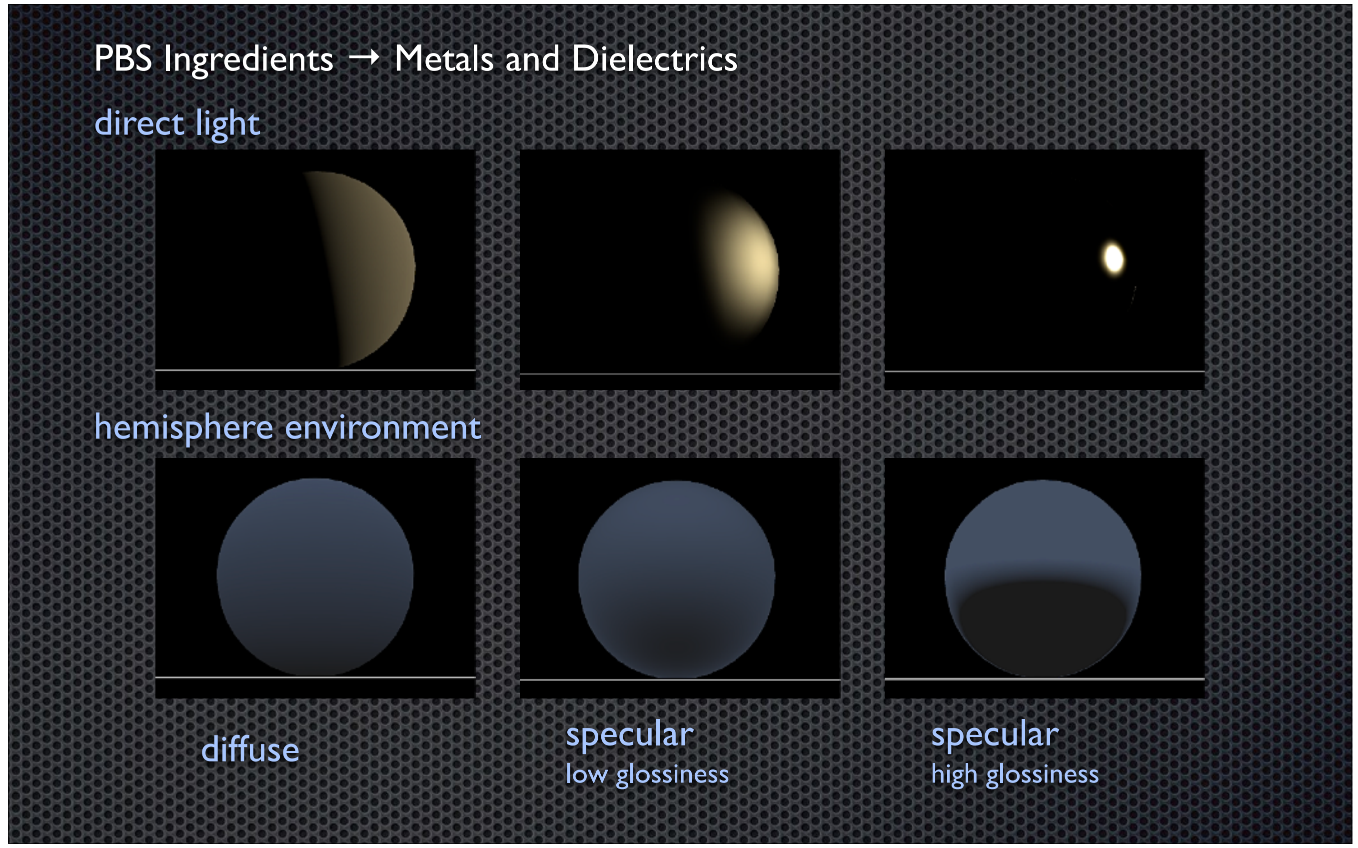



The solution is: The shader must have separate models for the diffuse and specular response to “ambient” illumination. Just as it has separate models for the response to direct illumination.

However, a uniform ambient color is not going to be very useful for this matter, since then there would be no difference between diffuse and specular response. It’d be all uniform! Therefore, we will use a hemisphere light throughout the presentation as an example.

A hemisphere light is the smallest step upwards from a constant ambient color: Two constant colors! One color is usually represents the sky color, and the other one represents the ground color.

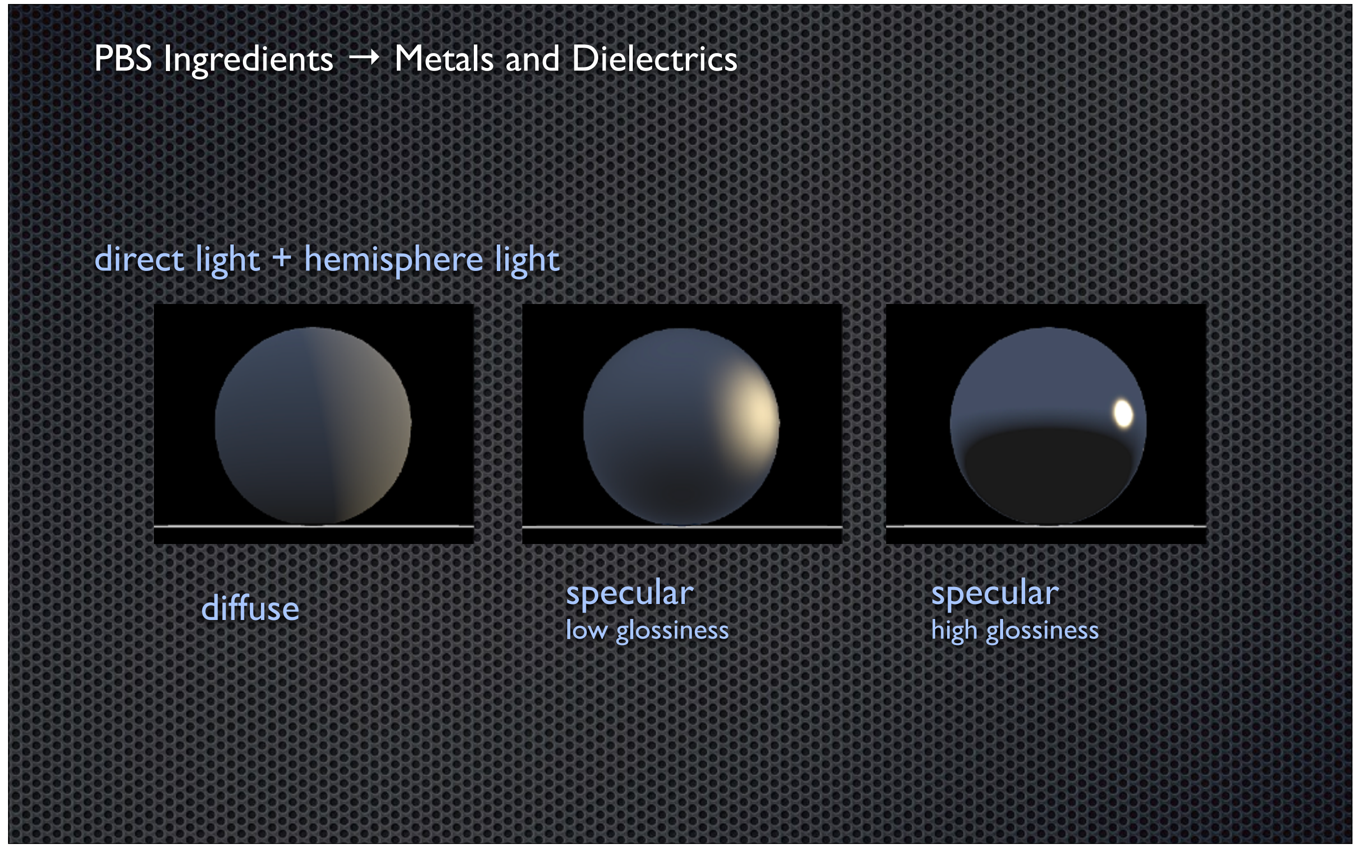
(When used in an outdoor environment, a simple hemisphere environment can give a dramatic improvement over an uniform ambient color!)




“Ambient” light is no longer existent as a category on its own. Instead, the shader just offers diffuse and specular responses to different kinds of illumination.

The top row shows the response to direct illumination (from a directional or a point source). The diffuse response shows the sharp border that was mentioned earlier as a proof of linear lighting. The energy preserving specular response gets sharper and brighter with increasing glossiness.

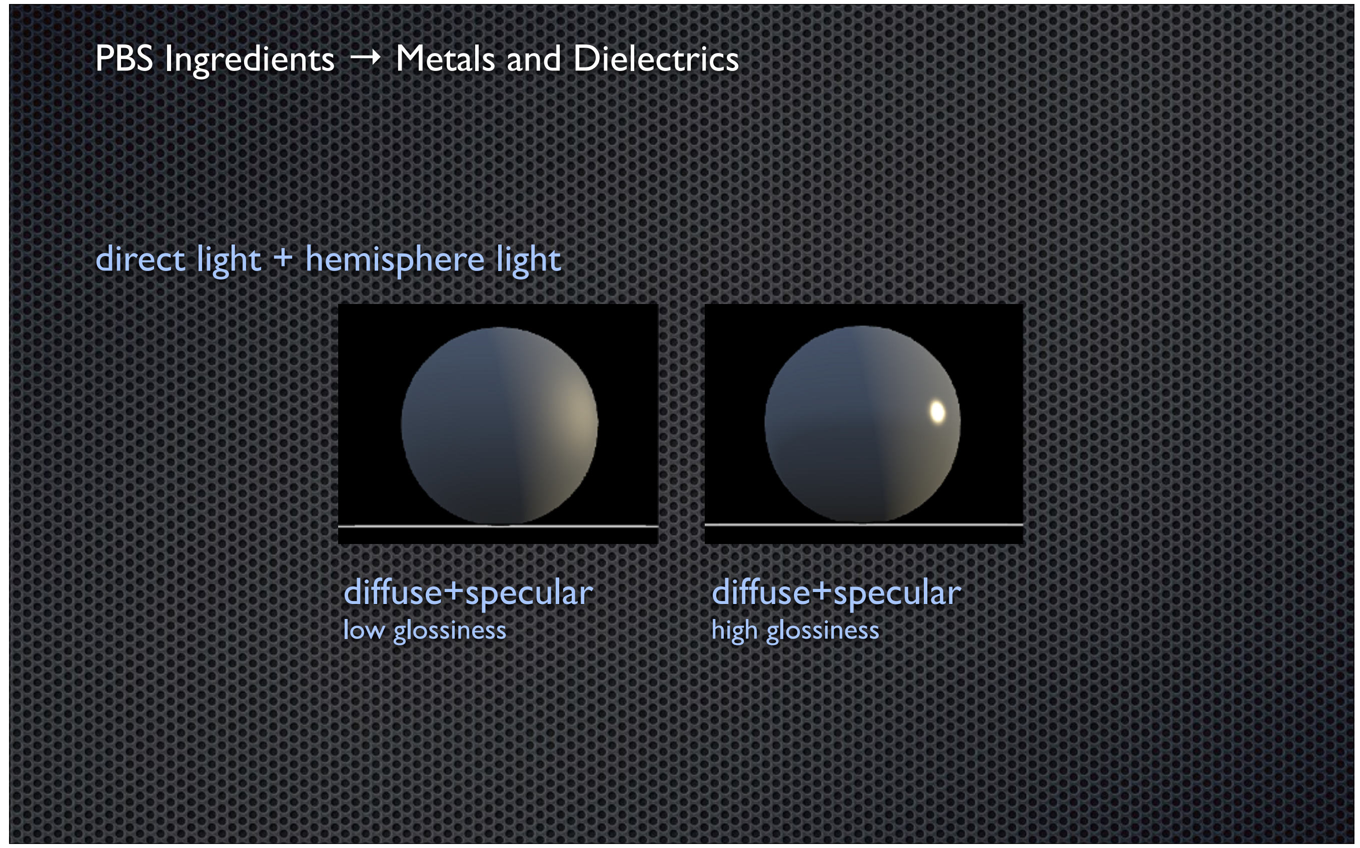
The bottom row shows the response to hemisphere illumination. In this case, the diffuse response is very similar to the specular response with low glossiness. The high glossiness case however starts to look like an environment mapped sphere, which it indeed would become in the limit of a perfect mirror!




This is the sum of direct and hemisphere illumination. (It you were following this presentation, it goes without saying that this sum must be calculated linearly …)

Purely diffuse reflection is on the left.
Purely specular reflection in the middle and on the right.

(The sphere material is 22% grey in each case.)




And finally this is the diffuse summed with the specular contributions.

In this case the diffuse color is 22% grey and the specular color is 5% grey.

Why not 22% grey for both? That wouldn't be realistic for a typical dielectric material.
I'll get to this in the next section when it comes to Fresnel reflection.

## PBS Ingredients → Metals and Dielectrics

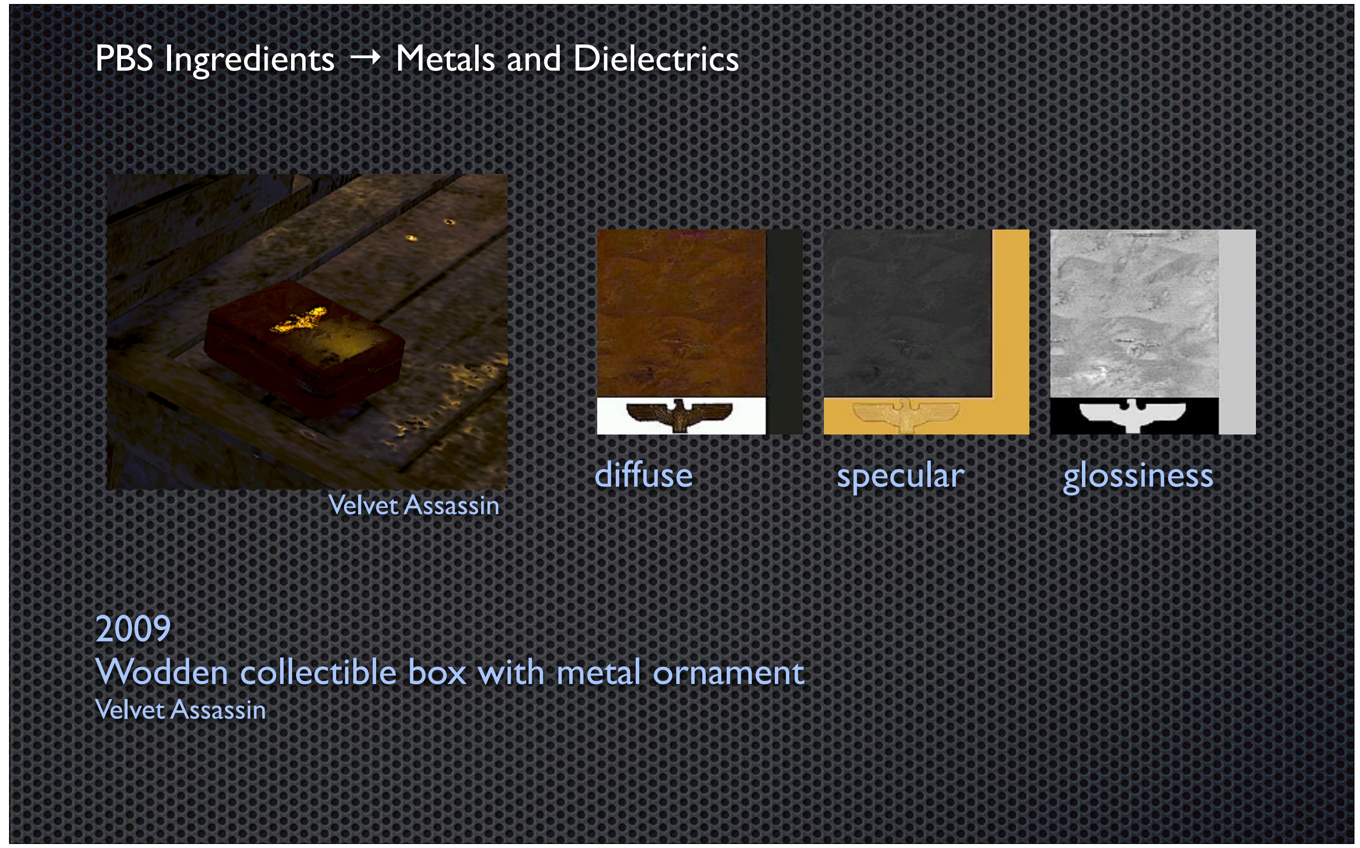

2006
World globe with purely specular surface
Spellforce 2: Dragon Storm

Here is an example of how the principles from the last slides have been put to good effect in an actual use case.

The screenshot shows a “world sphere” in Spellforce 2: Dragon Storm. The land regions are purely diffuse, while the water regions are purely specular. And both are on the same texture (which is one of the salient points).


Another example shows a wooden collectible box with a metallic ornament, this time from Velvet Assassin. There is a single texture atlas for the entire object, and is rendered with the same shader (no batch break).

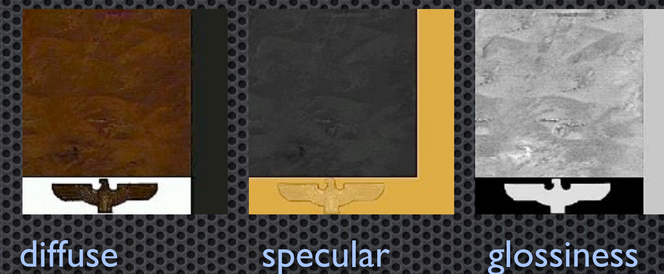
The diffuse color for the metal part is, well, not black but much darker than the specular color. You can argue that some dust on top of the metal is responsible for some diffuse part. In the end, when do you have a really pure metal in the real world?




Different perspective.

## PBS Ingredients → Metals and Dielectrics

Velvet Assassin




2009
Wodden collectible box with metal ornament
Velvet Assassin

Another different perspective.


The fourth ingredient of physically based shading is the Fresnel effect.
This will finally give a physical meaning to the “specular color”.

Fresnel effect (no news here)—

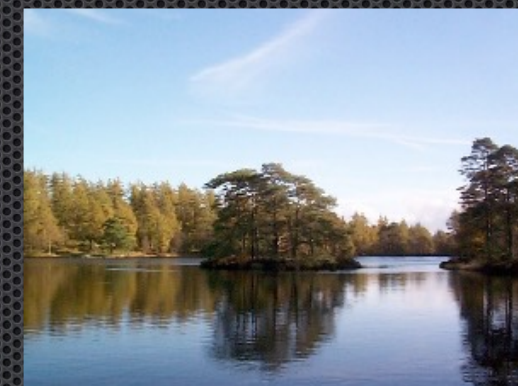
- Increasing reflectivity towards grazing angle

The Fresnel effect is known very well. Let’s just recapitulate to get everyone onto the same page: The reflectivity of a surface increases as a function of view angle.

PBS Ingredients → Fresnel Reflectance

## Fresnel effect (no news here)—

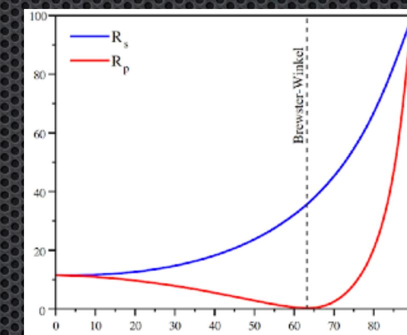
- Increasing reflectivity towards grazing angle
- Very prominent on water



This effect is very prominent on water, since water starts out with only about 3% reflectivity at normal incidence, which raises all the way to 100% on the horizon. It is no surprise then that water materials were the first ones that had to pay attention to the Fresnel effect, even when all other materials didn't.

## Fresnel effect (no news here)—

- Increasing reflectivity towards grazing angle
- Very prominent on water
- In theory, everything becomes a perfect mirror at the grazing limit



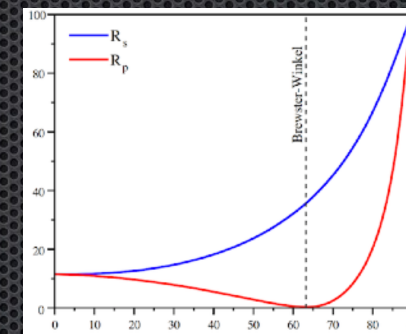

http://de.wikipedia.org/wiki/Fresnelsche_Formeln

This curve shows how the reflectivity goes to 100% at in the limit of a grazing angle.
In theory, everything becomes a perfect mirror when the view angle is just shallow enough.

PBS Ingredients → Fresnel Reflectance

Fresnel effect (no news here)—

- Increasing reflectivity towards grazing angle
- Very prominent on water
- In theory, everything becomes a perfect mirror at the grazing limit
- Corollary: Everything Is Shiny! http://filmicgames.com/archives/547

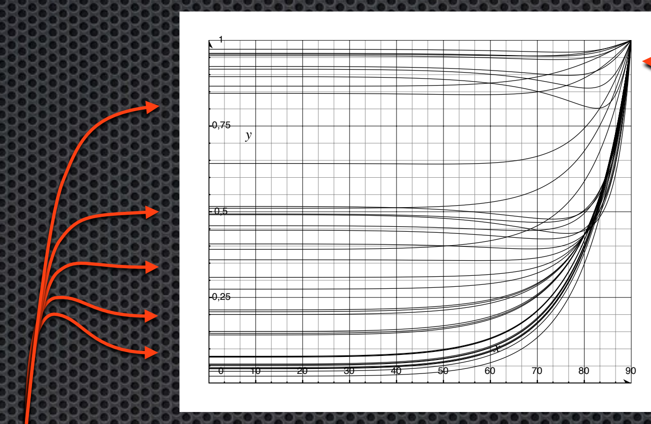



As a corollary, a perfectly diffuse material is not physically realizable. The Fresnel effect will always get in the way. Everything is shiny.

If you follow the link, you’ll see a nice proof that statement.

PBS Ingredients → Fresnel Reflectance

## Fresnel reflectance can be used to classify



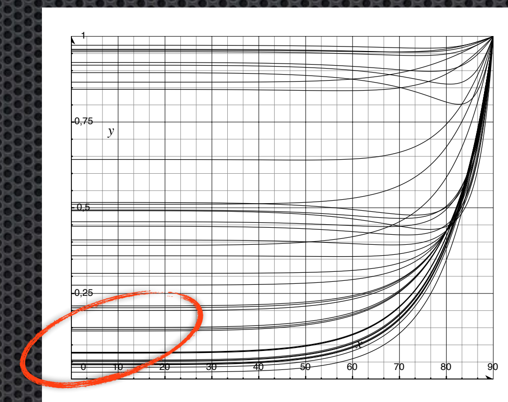

R(90°) is the same for all ...

... but R(0°) is different for each.

When we look at the Fresnel reflectance curves for different materials, we find that R(90°), the reflectance at the limit of grazing incidence is the same for all materials. But R(0°), the reflectance at normal incidence, is different for each material.

## PBS Ingredients → Fresnel Reflectance

Fresnel reflectance can be used to classify materials




Non-metals are here
(2% to 8%)

The value for R(0°) for ordinary non-metals is in the range from 2% (for liquids, like water) to 8% (for rock minerals). Almost all organic materials and plastics are around the 5% range.

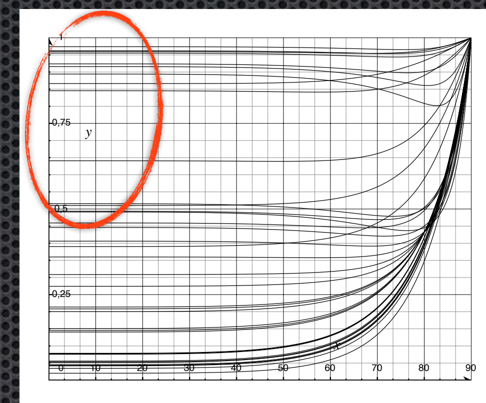
Therefore, most of the time, a specular texture is going to be approximately 5% grey. There is almost no point in texturing it.

PBS Ingredients → Fresnel Reflectance

Fresnel reflectance can be used to classify materials

Metals are here
(60% to 90%)

most are colored to
some degree

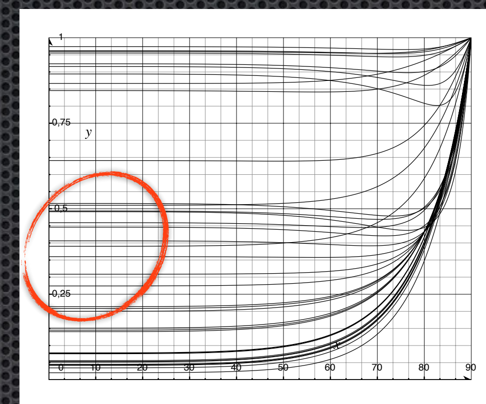



Metals live in the upper range of the reflectivity, and the reflectivity does not change much when the view angle changes. What is also unique to metals is that their reflectivity can be wavelength-dependent, so that the reflection is colored (specular color is different from grey, like in copper, gold, etc).

## PBS Ingredients → Fresnel Reflectance

Fresnel reflectance can be used to classify materials

Semiconductors and crystals are here (15% to 50%)




http://en.wikipedia.org/wiki/Sapphire

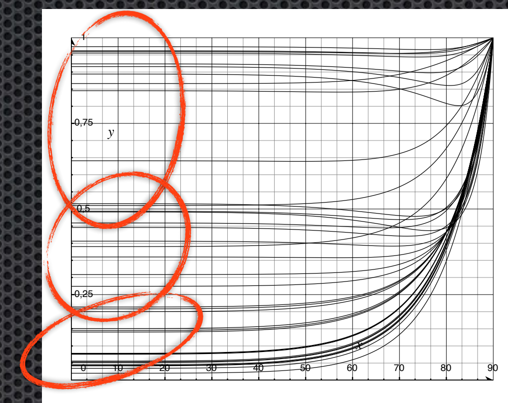
The middle range from 15% to 50% reflectivity is held by semiconductors and special crystal minerals. This includes pure diamond, aluminum oxide (sapphires) and crystalline silicon.

## PBS Ingredients → Fresnel Reflectance

Fresnel reflectance can be used to classify materials

Metals are here
(60% to 90%)

Semiconductors and
crystals are here
(15% to 50%)

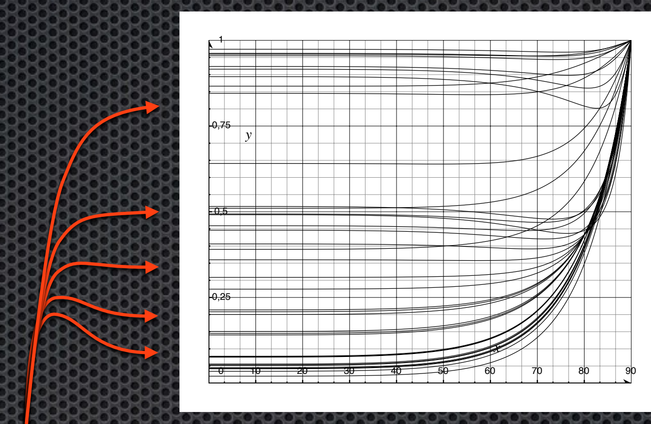



Non-metals are here
(2% to 8%)

And this is the big picture of the material categories according to Fresnel reflectance.

PBS Ingredients → Fresnel Reflectance

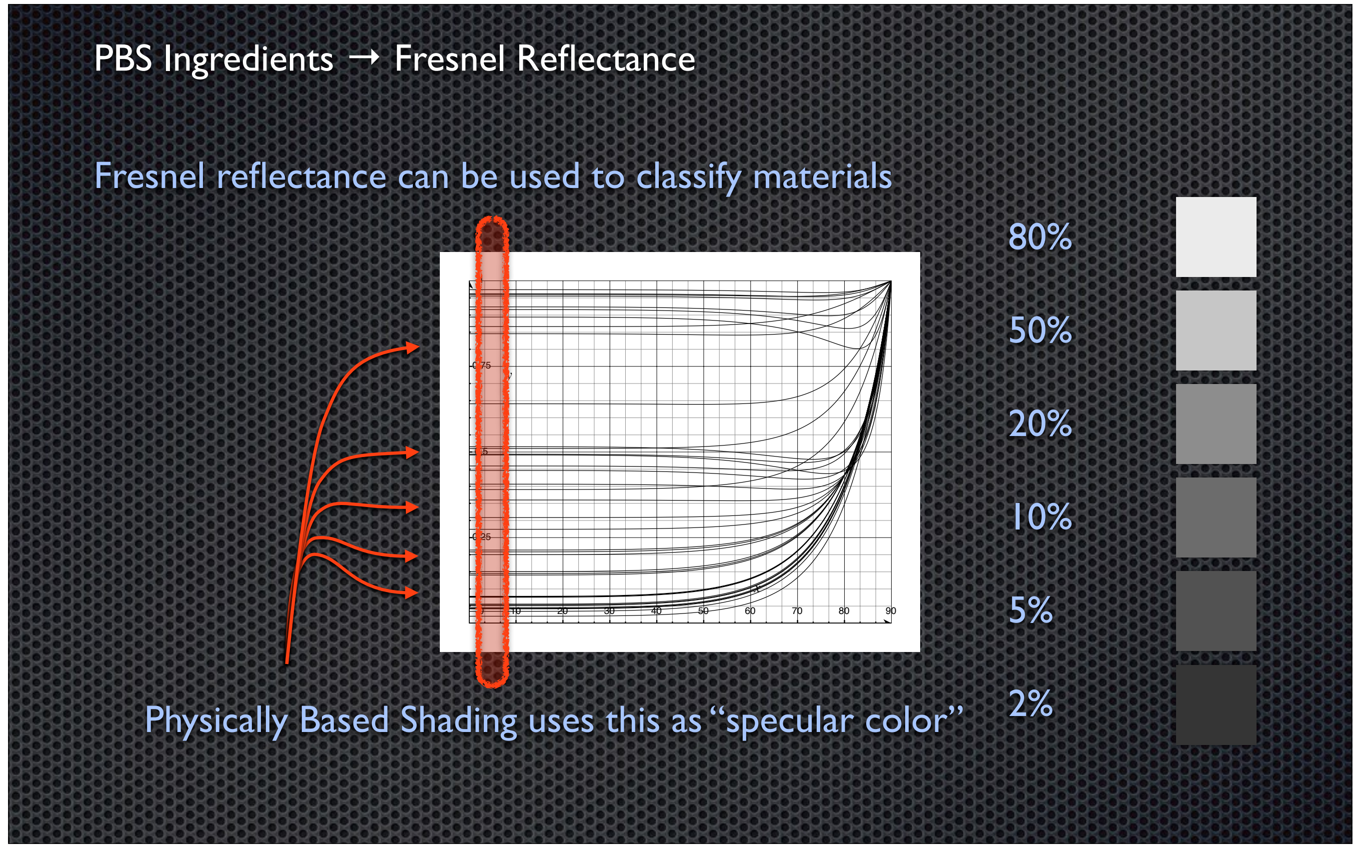
Fresnel reflectance can be used to classify materials



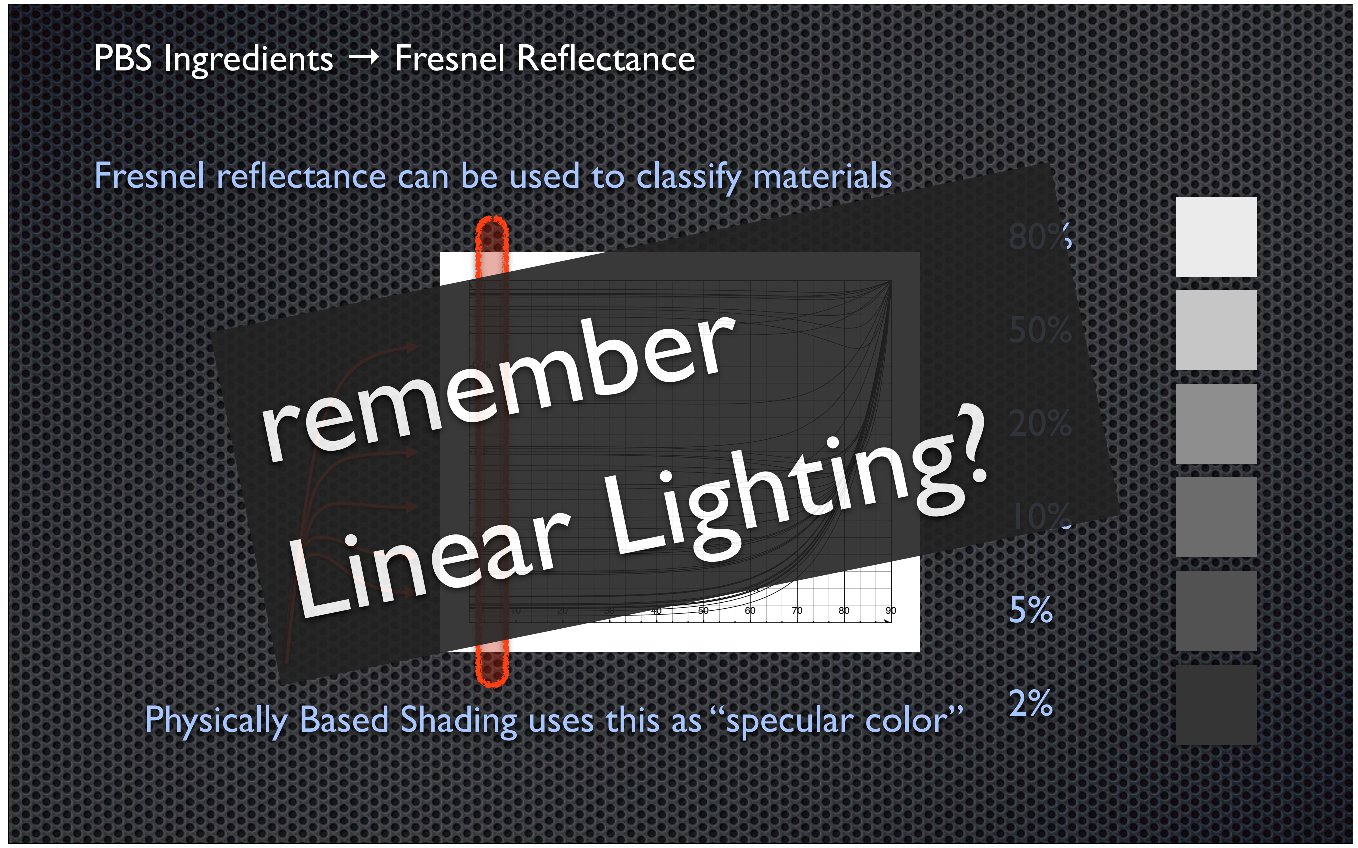

An approximation: all curves are fairly well predicted by R(0°) only

What's more, we can, to a good approximation, predict all the curves from the value of R(0°) alone. This is known as the Schlick approximation.

The small dents in that appear in the curves for metals are usually not relevant for the visual appearance.



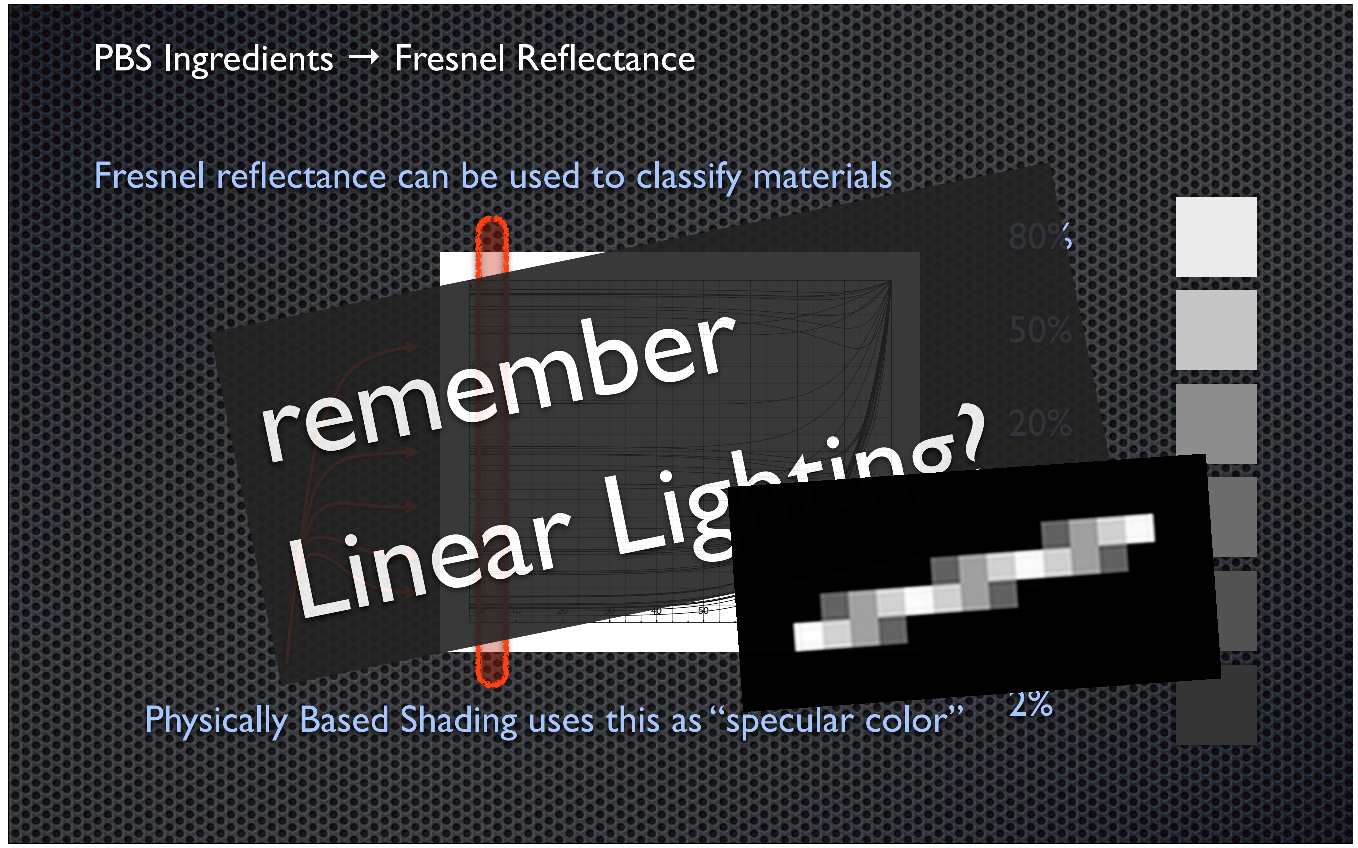

In Physically Based Shading, one takes R(0°) and simply uses it as the "specular color". The shader then approximates the curve of the Fresnel reflectance based on this value alone.




Now onto something different.

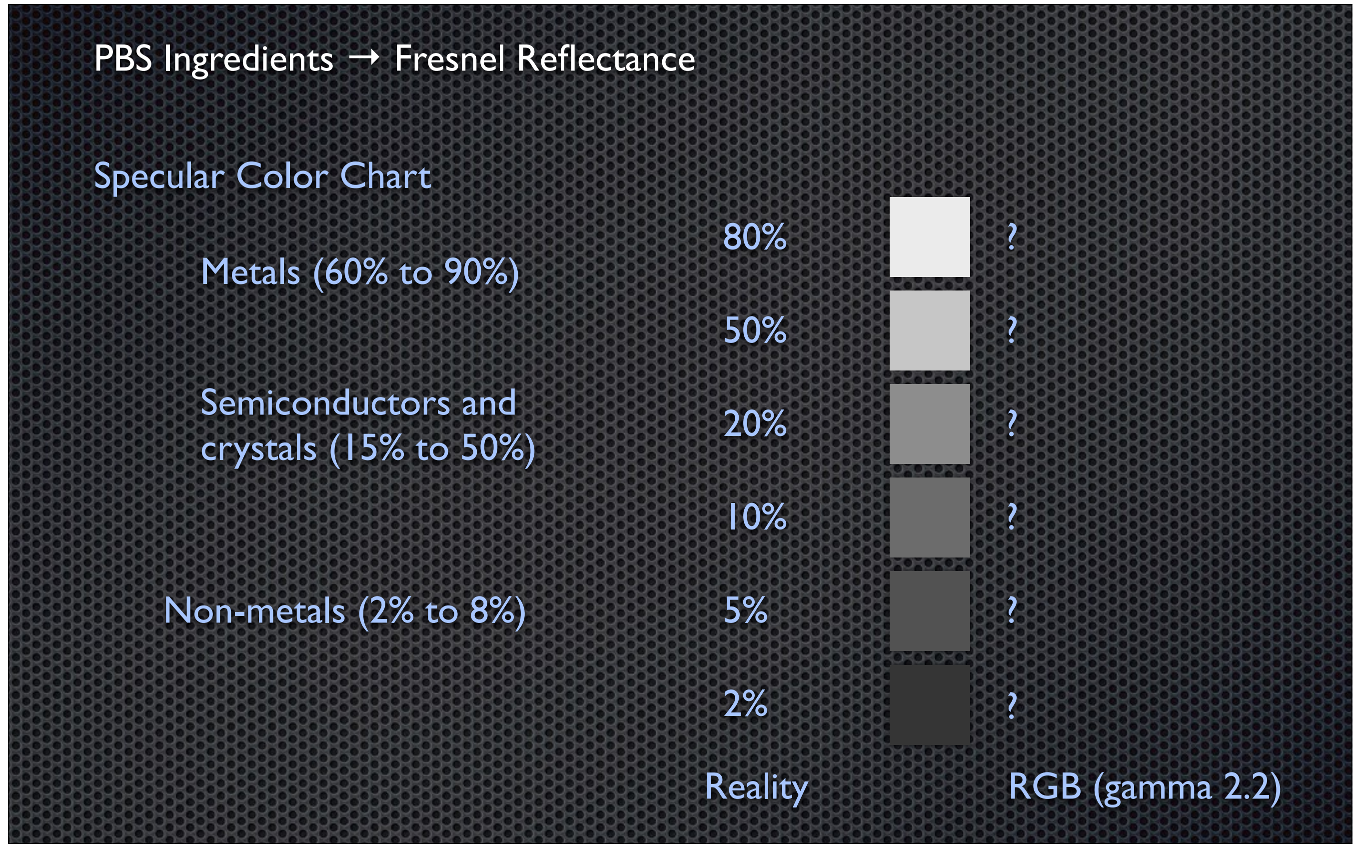
We want to put these reflectivity levels (2%, 10% etc) onto a texture, encoded as grey levels. We'll have to account for the display gamma thing.

Remember what was said in the beginning about linear lighting?
Right?



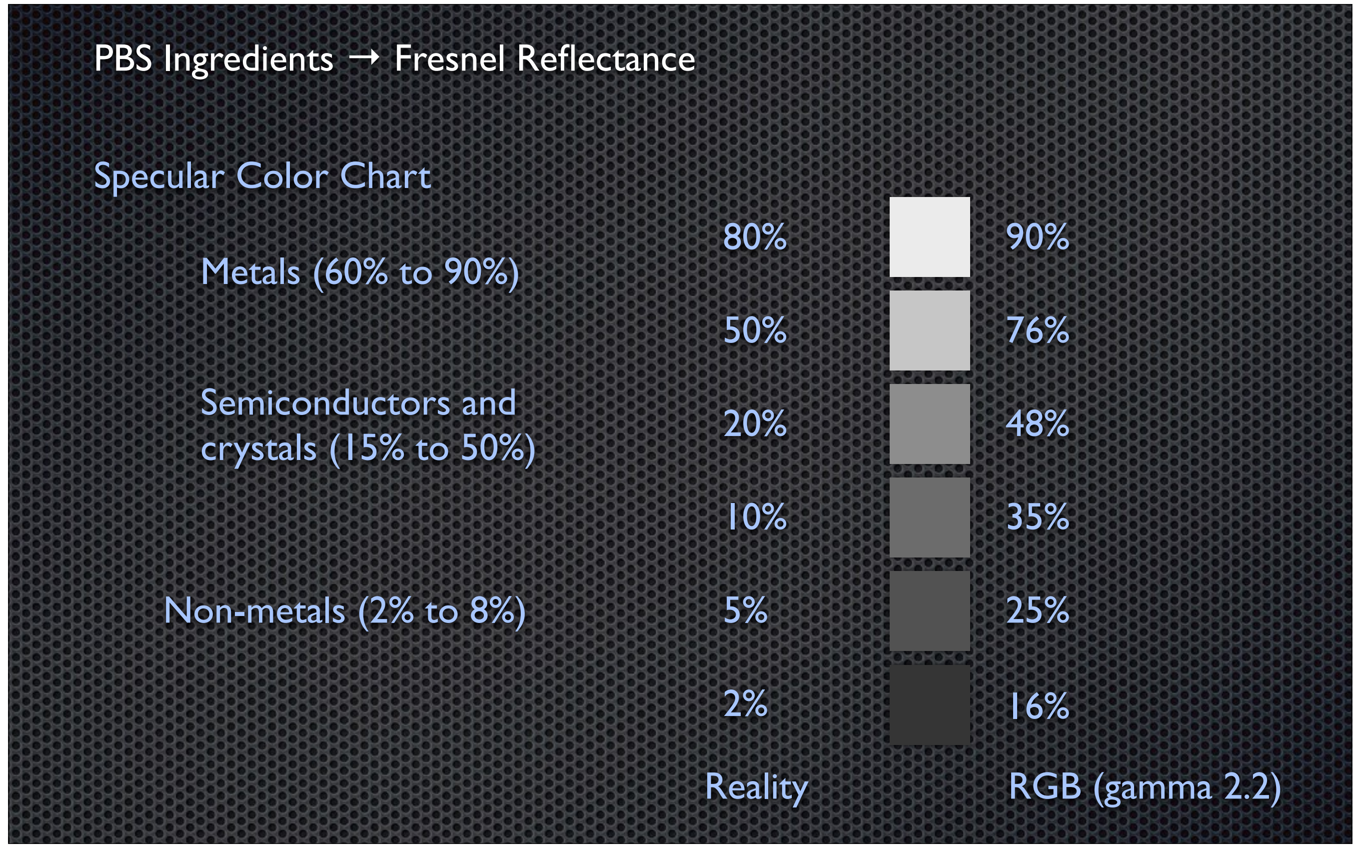

Right.

Grey levels must not be taken for their numeric value, but instead for their display value. The latter is dependent on display gamma (which is nowadays standardized to 2.2).




Auditorium question (depending on time)

QUIZ TIME!
Guess the values.
Are they higher or lower?




So here is the answer.
The RGB levels are higher than the real levels. An 48% RGB-value is displayed as about 20% grey in real terms. The difference gets more dramatic as we move to lower values. A 16% RGB-value is displayed as about 2% grey in real terms.

If you'd measure the levels of the grey boxes with a photometer, it should approximately yield the "real" values.

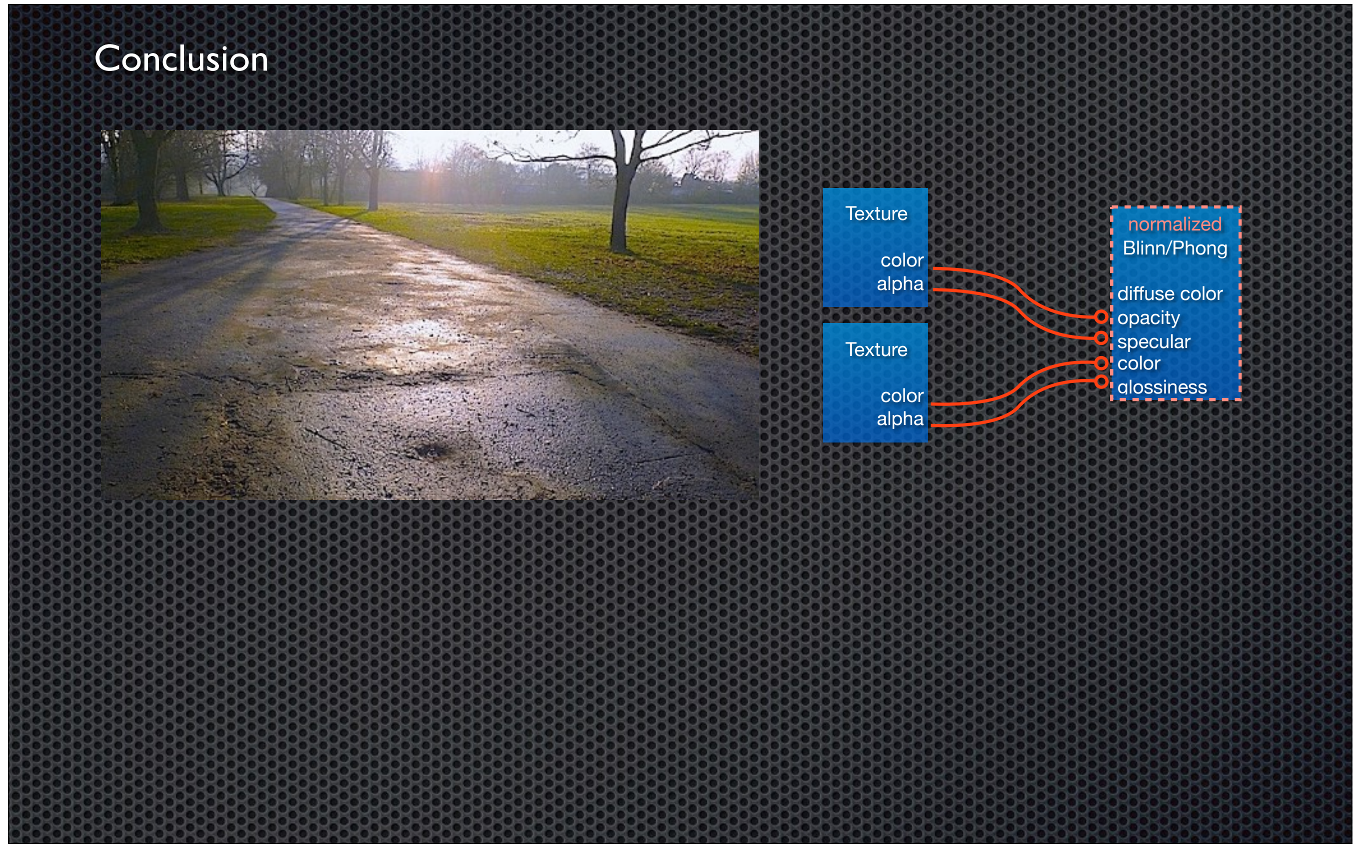
## Further reading

- Blogs
  http://seblagarde.wordpress.com/2012/04/30/dontnod-specular-and-glossiness-chart/
- Real Time Rendering, 3rd Edition
  http://www.realtimerendering.com
- ShaderX 7
  http://www.shaderx7.com
- SIGGRAPH Course on PBS
  (2010) http://bit.ly/s10shaders
  (2012) http://bit.ly/s12shaders

# Conclusion

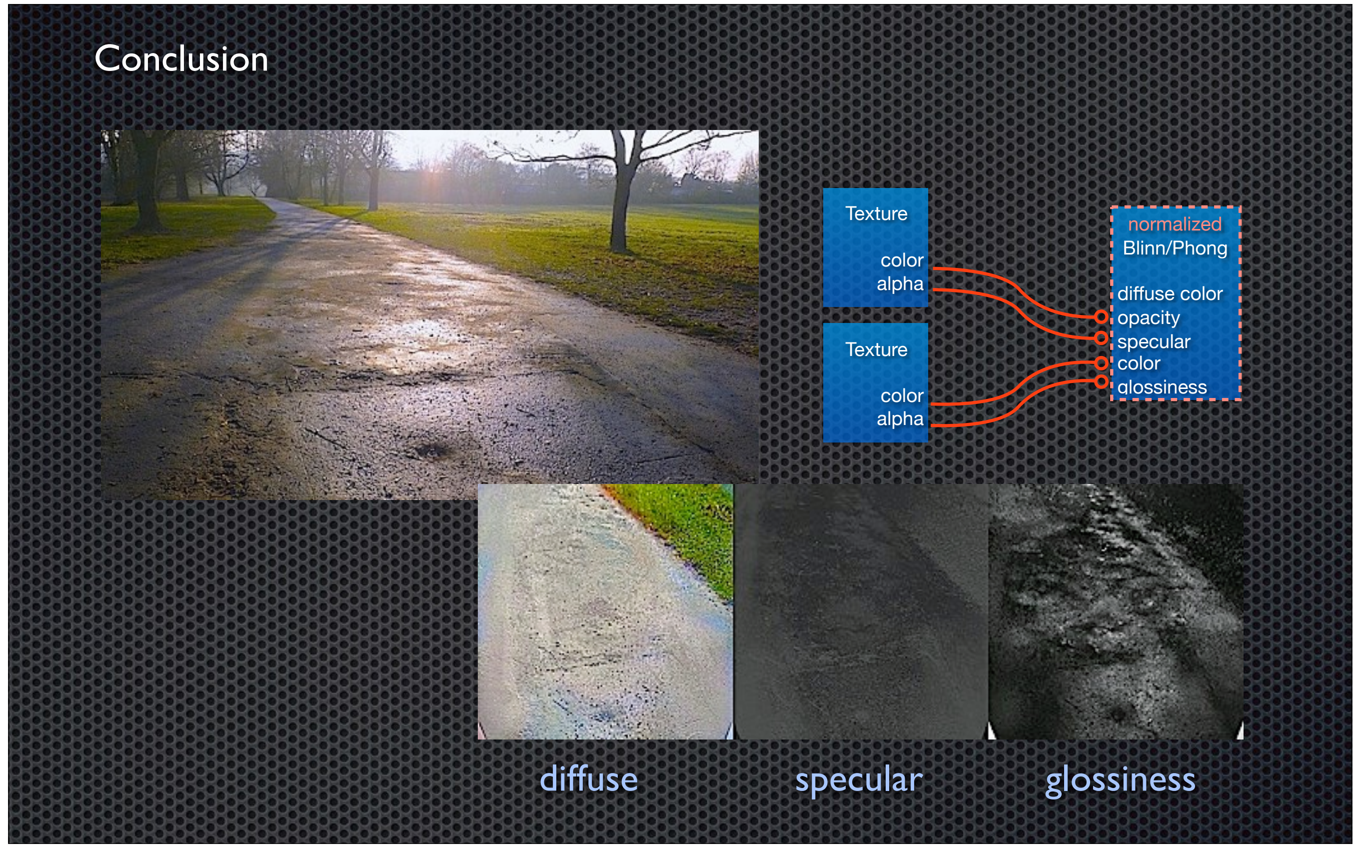
## Conclusion

So here is the take home message:




With physically based shading, the glossiness is the new important thing. This value alone determines both size and strength of the specular highlight.

Remember: smooth surface (wet) vs rough surface (dry).




The diffuse color should be devoid of any lighting variation and is an idealized subsurface color. It is usually a little bit more saturated than the observed color of the material, because it's saturation is going to be reduced when the specular component adds on top of it.

The specular color … if not for metals there is no point in texturing it. You may conserve texture memory by making this a constant color.

## Conclusion

Physically Based Shading

- Is going to be implemented in more and more engines
- Affects content creation, mostly texturing
- Affects how you think about colors and numbers

# Thank You

More information and links to related material on my blog

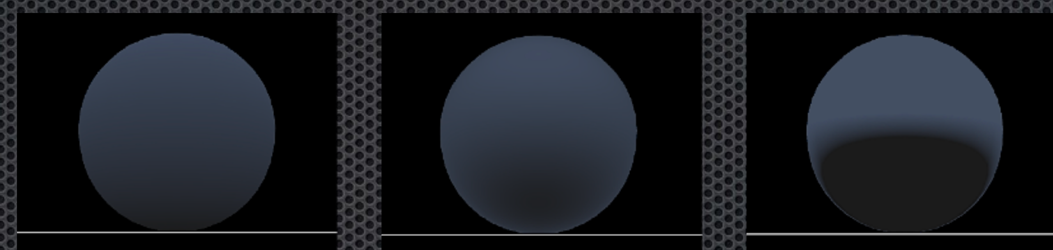
http://www.thetenthplanet.de

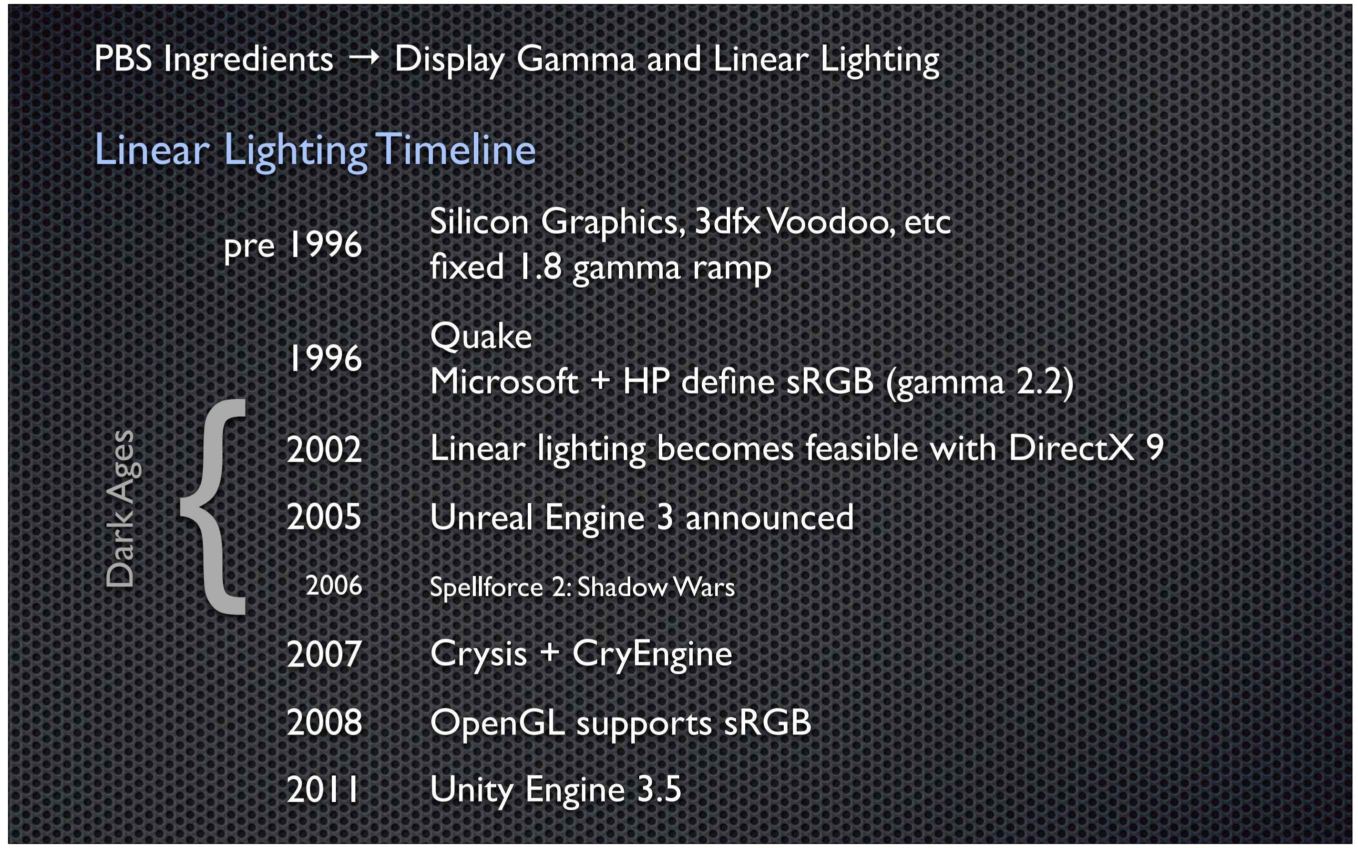
# Bonus Slides

## PBS Ingredients

### Other ingredients of Physically based lighting

- Image based reflections
  Use an environment map instead of a simple hemisphere
  Automatic mipmap selection based on glossiness






Here is a small (and very subjective) timeline for linear lighting. It is interesting to know that the early practitioners of real time 3-D were well aware of the importance of gamma and the need for linear lighting. This includes Silicon Graphics as well as the 3dfx Voodoo. As a compromise between hardware cost (bit depth) and perceptual fidelity (banding artefacts), these devices had fixed gamma ramp in the range of 1.8.

One of the first games to attempt to use linear lighting indeed was Quake! Their color palette was chosen such that addition of light sources in the light maps was approximately linear. This property was however only available with the software renderer.

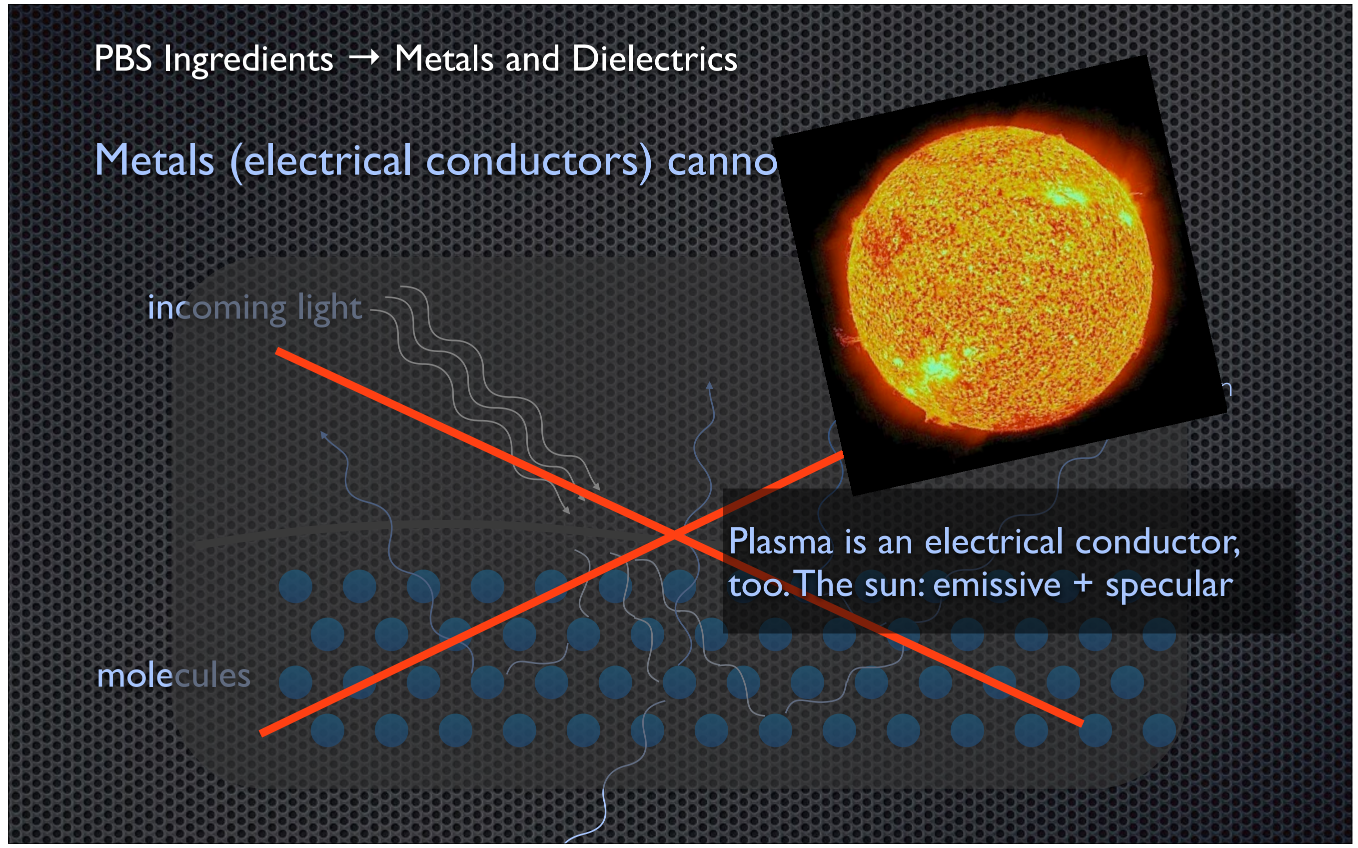
With “Hardware T&L” in graphics cards came the “dark ages” of linear lighting. Until then, lighting calculations were done in software, so that the lack of linear lighting in hardware could be worked around. Hardware T&L however meant that all lighting was done in hardware, and that meant raw RGB values with non-linear lighting, at least for the time being.

The introduction of DirectX 9 marked a turning point because it did standardize an sRGB-compatible gamma conversion in hardware. But this feature was picked up very slowly. The first 3D engine to consequently use these features is to my knowledge the Unreal Engine 3. For comparison, Spellforce 2, one of the games that I worked on, used this and was released in 2006.


Here again is a possibly incomplete and very subjective timeline of the things and and events that, as to my knowledge, feature an energy preserving specular highlight. I set the start in 1994 where a research paper appears with the theory of modifying the Phong material to behave energy conserving.

Again for comparison the release dates of the games I worked on, Spellforce 2 and Velvet Assassin. The former had mostly hardcoded constants for both glossiness and normalization factor since only a few special materials could vary the glossiness per pixel, but still.

The 2010 release of Call of Duty: Black Ops is to my knowledge the first title by a top tier brand to bring PBS to the mainstream. From there on the technique becomes ever more popular. The 2013 title Remember Me is a noteworthy example to put varying glossiness to good effect with its rainy-city-at-night look.




Oh, and plasma is an electrical conductor too.
So in theory, the sun is emissive + specular.

Now find the light source that would be bright enough to cause a visible specular reflection on the surface of the sun …

## Bonus→ Energy Preserving Specular

2006
Per pixel varying glossiness
Ice material in Spellforce 2

More screenshots from the Spellforce 2 an ice/crystal material. This is the entrance to a castle that is supposed to be made entirely from ice, in different times of day.