

Specular and Metallic Maps

Specular maps are a tricky subject. Many write ups explaining specular maps focus on how a specific game or engine handles them, and the specific compromises and simplifications inherent to them. One thing is clear, though, specular maps focus on *shiny* stuff.

In Unity, that shininess is defined by two components, following the principles of PBR. There is the part that determines which sections are light absorbing, like metal, instead of light reflecting, like paper. That's stored in the red channels of the metallic map. And then there's the part that determines which sections have a smooth and reflective surface, like polished wood, instead of a rough surface like rusty metal. That's stored in the transparency of the metallic map. These attributes are called Metallic and Smoothness.

In proper terminology, non metallic surfaces like cloth are called dielectrics. The light that hits their surface bounces around inside and produces an even colour; albedo light. Metal, on the other hand, conducts electricity, and so reflects the photons that hit it and absorbs what doesn't reflect; producing specular light.

So that covers metallic maps. But wait, what about specular maps? Well, it turns out that it's a lot easier to have *one* map that has both the albedo and specular colours, and a second simpler map to define which parts are metal and which are not, instead of having two full colour maps that can take up more memory. However, a metallic map can't represent materials like cloth or skin well, which are neither metallic nor hard. So just consider a specular map as a way of defining a specific kind of shine, while a metalness map only defines that something is or isn't metal. This is the difference between Standard and Standard (Specular).

In a specular system, the metallic portions of the material are defined in the RGB channels of the specular map. The Alpha is still reserved for the smoothness. This takes more memory, but it's useful when a simple mask doesn't cover what you need. A good real world example are surfaces with shiny, thin metal lines. At angles, or far away, they'll lose their shine with a metallic map due to texture filtering. But with a specular map, they'll remain properly shiny.

However, if you add a metallic/specular map to an object, you'll notice the areas with specular become darker. This is because of *energy conservation*. One material can not be both fully dielectric and specular at the same time - only one, or the other, or something in between. So if dealing with specular maps made for another engine, they may need editing to fit.

To review:

- **Metallic** maps store an on/off value specifying whether part of the material is metallic or not, in the map's **red** channel. The material's albedo is used to specify how strong the specular reflectance is.

- **Specular** maps store a colour value representing the specular reflectance directly, in the **colour (RGB)** channels of the map.
- **Smoothness**, or glossiness, or the inverse of the surface's roughness is stored in the **alpha** channel, in both metallic and specular mode.

References:

- <https://google.github.io/filament/Filament.md.html> Google's guide to using PBR with their Filament engine. Written by their engineers, it covers the topic in a deeper way.
- <https://academy.allegorithmic.com/courses/b6377358ad36c444f45e2deaa0626e65>
- <https://marmoset.co/posts/pbr-texture-conversion/>
- https://www.sharetextures.com/textures/blog/physically-based_rendering/
- <https://google.github.io/filament/Material Properties.pdf> Google's reference chart to PBR materials. When making your textures, this is a useful thing to have on hand.

Later on, I'll explain how you can actually use these features.

Revision #3

Created 27 January 2022 06:28:48 by Silent

Updated 17 April 2024 02:45:25 by Silent