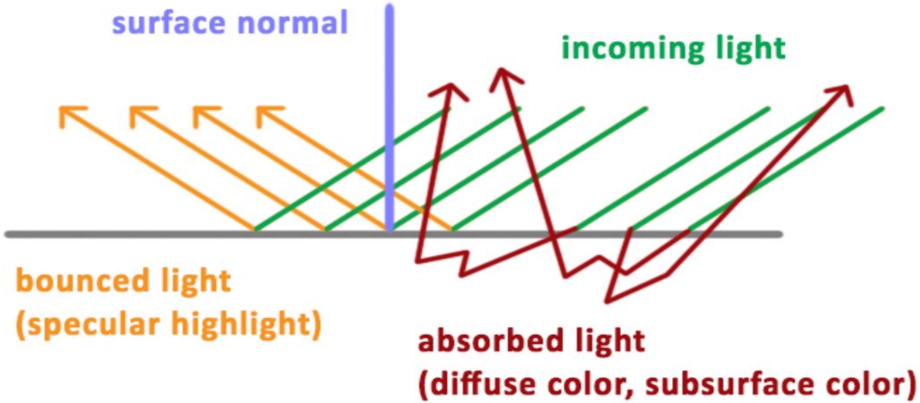
Diffusion and Reflection

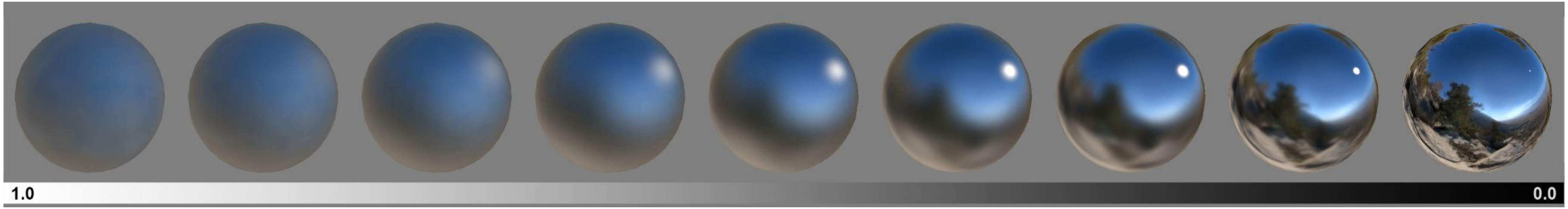
- Real-world surfaces *reflect* or *absorb* light in different ways
- **Reflected** light bounces off of the surface at an opposite angle, reflected (flipped) across the surface normal
- **Absorbed** light becomes *scattered* (or *diffused*) within a surface, exiting in a random direction
  - Some of the light may be retained within the surface and converted to heat
- Light which reflects or bounces off of a surface is represented by a ***specular*** highlight (surface appears reflective or shiny)
- Light which scatters or diffuses throughout a surface is represented by a ***diffuse color*** (the *albedo* or *base color)*
  - Our perception of a surface color is determined by which wavelengths of light are entirely absorbed by the surface, and which ones scatter or diffuse within it.



Conservation of Energy

- Physical principle which states that a surface may never reflect more light than it receives.
- It explains the behaviour of specular highlights on a given surface:
    - Rough or diffuse surfaces have a “wider” highlight, which is less bright
    - Smooth surfaces have a “tighter” highlight, which is more intense
- Mostly a concept for tech-artists or engineers creating PBR pipelines and workflows (but still useful to comprehend)

rough smooth

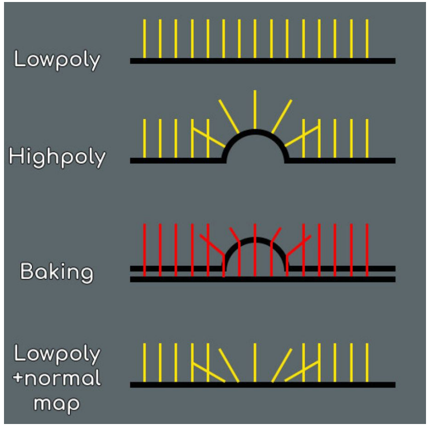


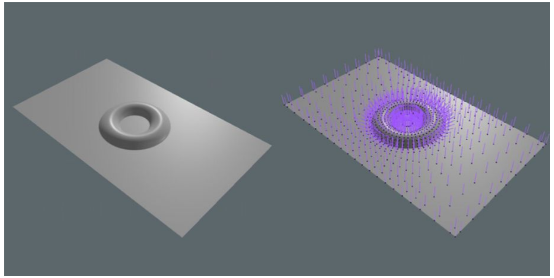
plaster glossy plastic steel mirror

Original diagram from https://docs.unity3d.com/Manual/StandardShaderMaterialParameterSmoothness.html

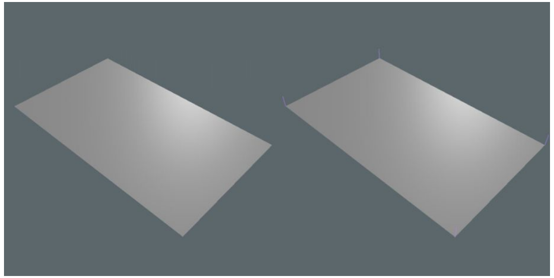
Normals

- A ***normal*** is a maths term for a directional vector which is perpendicular to a given surface
- The appearance of lighting or shading on a surface is dependent on its normal (eg, the direction it faces, its orientation)
- ***Normal maps*** are used to provide detailed surface information at a smaller scale than would be practical to represent using actual geometry (eg,, face/vertex normals)
- A flat surface with a normal map is treated by the renderer as if it has many perturbations or variations in the surface
  - This is due to the fact that each texel in the normal map is interpreted as a vector which represents some deviation from the direction of the geometry normal(s).
- Baking is simply the process of encoding the high-poly geometry normals into the normal map texture





high-poly normals

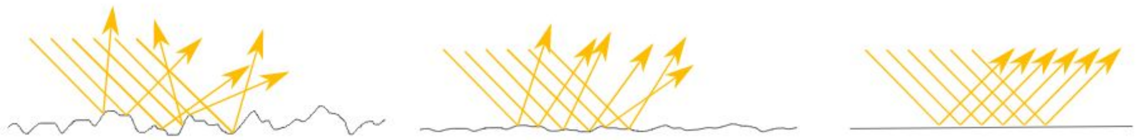


low-poly normals

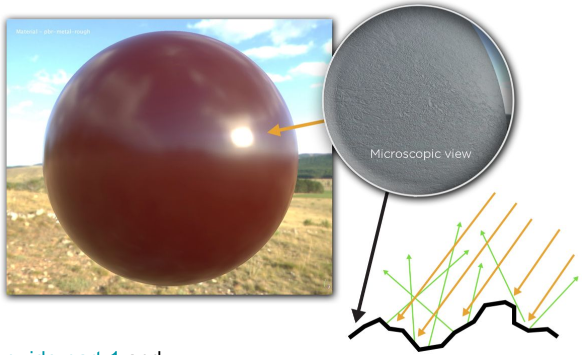
Diagrams copied from https://80.lv/articles/tutorial-how-normal-maps-work-baking-process

Microsurface

- Real-world surfaces have an infinite amount of tiny imperfections, bumps, or other surface details
- Surface detail at this scale is impractical (or impossible) to represent using the normal map
  - We’re constrained by the texel density or texture resolution (eg, 512px/metre, 2048^2 textures)
  - The individual details may be too small for our eyes to even perceive
- We perceive the “roughness” or “smoothness” of a surface largely due to its ***microsurface*** details:
  - Surfaces with lots of imperfections appear “rough” because the reflected light scatters in many directions.
  - The specular highlights on a *rough surface* are more diffuse, blurry, and spread across a wide area.
  - Surfaces with fewer imperfections appear “smooth” because incoming light is reflected back in the same direction.
  - Specular highlights on a *smooth surface* appear “tighter” and more defined - making it appear glassy or mirror-like
- Details in a *microsurface* map are separate (but complementary) to normal map details. Both affect a surface’s interaction with light, but they control different aspects of shading:
  - Normal maps represent deformations or perturbations in the surface
  - Microsurface maps define the “roughness” or “smoothness” of the surface



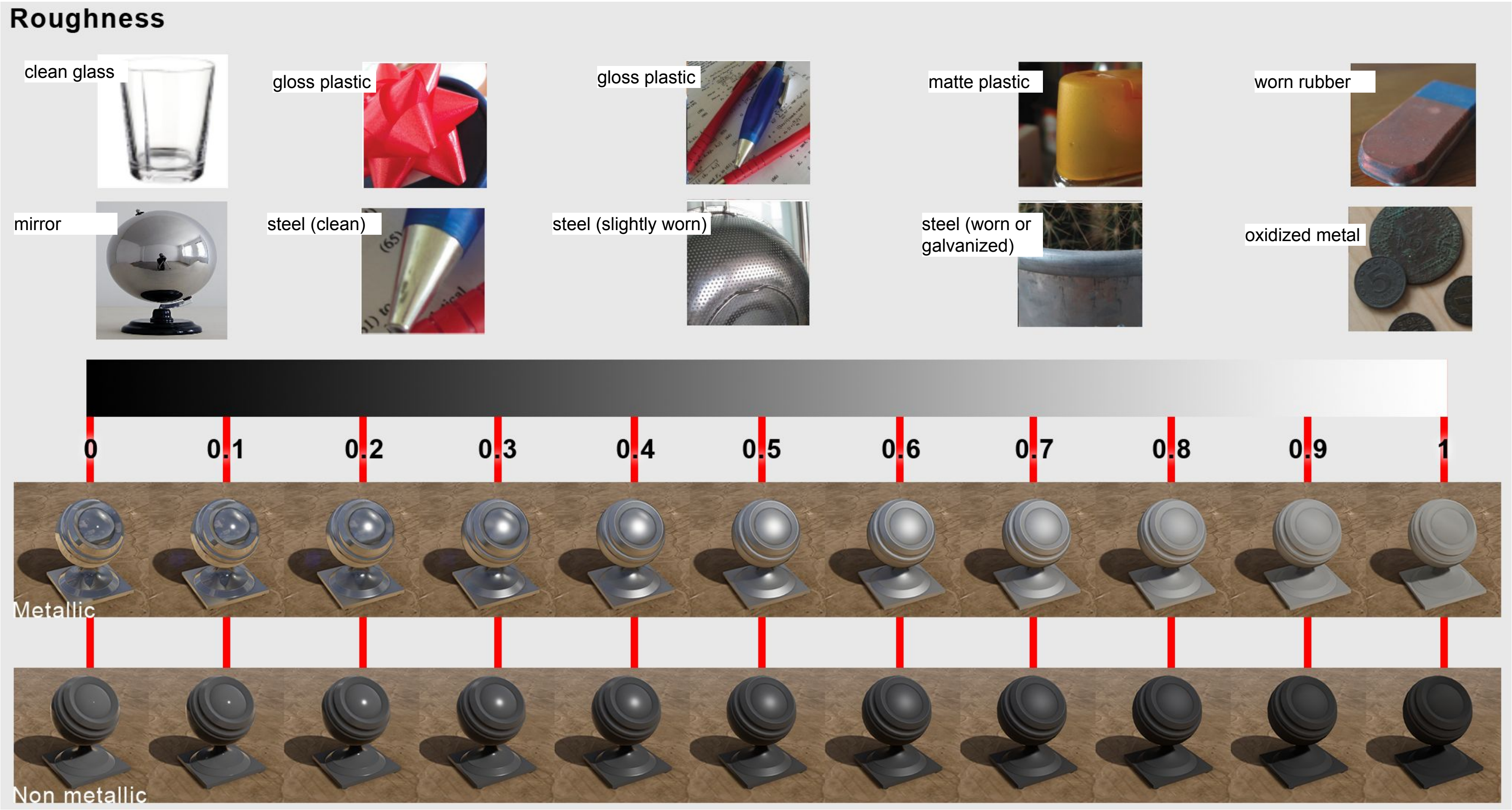
rough surface smooth surface



Diagrams copied from https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-1 and https://docs.unity3d.com/Manual/StandardShaderMaterialParameterSmoothness.html

Roughness/Glossiness

- In a PBR environment, the *microsurface* details are defined using either a ***roughness*** or ***glossiness*** map.
- Unreal uses a *roughness* map, but Unity and others (eg, Quixel) tend to use a glossiness (*gloss*) or "smoothness" map.
  - Both types of map represent the same information but are the inverse of each other.
  - You can always convert gloss to roughness (or vice versa) by inverting the map.
- In a roughness texture, values of **0** RGB are *entirely smooth* while values of **255** RGB are *entirely rough*.
- Physically speaking, it is rare to find real-world surfaces which are entirely rough or smooth.
  - Therefore, avoid authoring roughness maps with values close to either extreme.
  - The PBR validator utility will highlight extreme roughness values.
- You can use calibration charts or scanned materials (eg, Megascans) as a *reference* or *starting point* for roughness/smoothness of various real-world surfaces.
- Measured values are **not** a hard rule for surface roughness however. Surfaces of the same type can have various degrees of weathering, different coatings, etc, which affect the resulting roughness value.
  - Therefore, use observations of real surfaces, references, and your intuition as an artist when authoring roughness.
- Consider how to layer and represent details across the different channels (albedo, roughness, normal) - sometimes it may make sense to apply detail like stains, scuffs, patterns, etc in ***only*** the roughness channel rather than the normal or albedo.
  - This is similar to authoring specular maps in previous generations - except now, you are defining the surface *roughness* or *smoothness*, rather than the *specular intensity*.
  - The renderer itself will have control of specular intensity, which is discussed in the upcoming slides

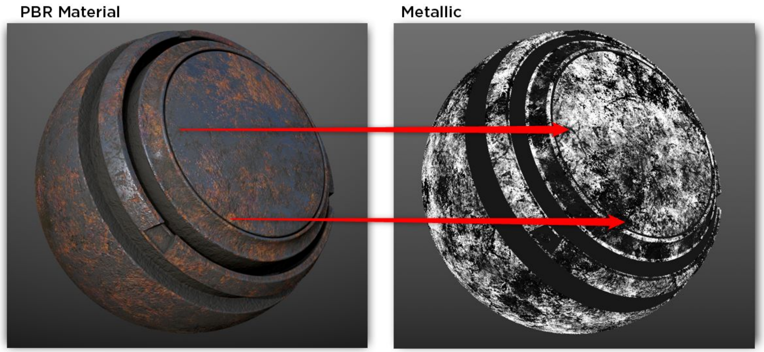


Original diagram copied from https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/

Original diagram copied from https://marmoset.co/posts/physically-based-rendering-and-you-can-too/

Insulators and Conductors

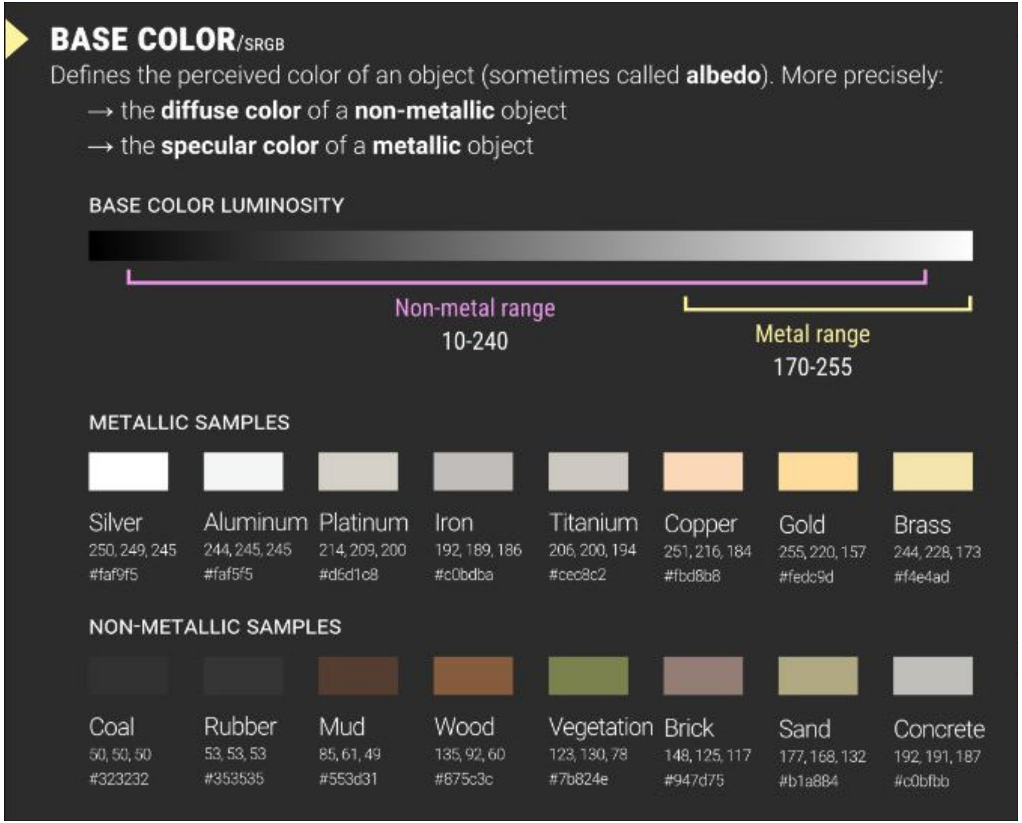
- Studies of real-world surfaces (and physics) highlight their differences when it comes to reflecting and/or absorbing light:
  - Nonmetallic surfaces (***Insulators***, or ***Dielectrics***) have low-intensity specular reflectance (measured between 2-5%) and do not impart a tint to specular highlights (they remain the same color as the originating light source).
  - Metallic surfaces (***Conductors***) have high-intensity specular reflectance (~60-95%), and may impart some tint to the specular highlight depending on the specific type of metal.
- While the terms "insulator" and "conductor" relate to electrical conductivity, this is a rough classification, not a binding one:
  - Metal which has been painted or coated is classified as an insulator (nonmetallic) in a PBR environment.
  - Likewise for dust, dirt, grime, oxidization (rust) or other weathering on metallic materials.

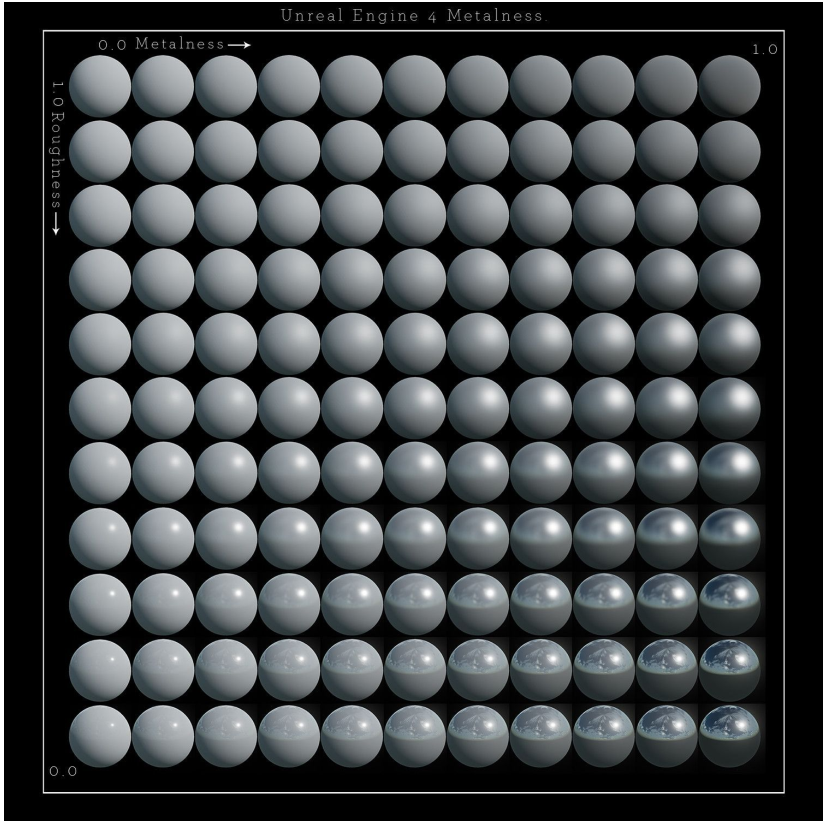


Original diagram copied from https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-2

Specularity and Metalness

- These are possibly the most confusing concepts for artists transitioning to a PBR workflow.
- In an old style specular workflow, both ***specular color*** and ***intensity*** are defined with individual texture maps.
  - This requires observation, guesswork, and artistic intuition to create realistic looking surfaces - particularly metals
- PBR environments remove a lot of this guesswork due to the inherent physical differences between the two material types:
  - For ***dielectric*** materials, the specular intensity is uniformly low (~2-5%), and the specular color is neutral.
  - For ***conductive*** materials (bare metals), specular intensity is uniformly high (~60-95%), and there may be a tint to the specular color depending on the specific type of metal (steel, gold, copper, etc).
- We use a ***metalness*** map to mask parts of the material, defining them as conductors (white) and dielectrics (black).
  - Partially metallic materials (*metalloids*) are rare, so avoid authoring metalness maps with gray values.
- The *albedo* map is treated as both a *diffuse color* and *specular color/intensity* map, based on the metalness mask:
  - Conductive parts of the material (white metalness) receive specular color/intensity from the albedo.
  - Dielectric parts of the material (black metalness) have neutral specular color and a *fixed* specular intensity.
- The fixed specular intensity for dielectric materials is termed the "***Specular F0***" (Fresnel zero) value.
  - This concept is (again) mostly useful for technical artists or engineers working on PBR pipelines.
  - The F0 value is the "minimum" specular intensity, when viewing an object "head-on". The intensity increases up to 100% at grazing angles - hence the term "fresnel zero".
  - **TL;DR for artists** - Dielectric materials receive constant specular intensity (F0 ~4%), conductive materials receive specular intensity and color from the albedo.
  - This is why it is *imperative* to use measured or otherwise calibrated color values for metals, as you aren't defining a diffuse color, but rather the *specular color* -and- intensity.





Original diagram from https://google.github.io/filament/images/material_chart.jpg

Sources/Further Reading

- Wes McDermott (Allegorithmic) - The PBR Guide by Allegorithmic - Part 1 and Part 2
- Joe Wilson (Marmoset.co) - Physically-Based Rendering, And You Can Too!
- Jeff Russell (Marmoset.co) - Basic Theory of Physically-Based Rendering
- Luis Mesquita - Everything About PBR Textures And A Little More (Part 1)
- Thibaud Tournery-Bachel (Sketchfab) - Substance Painter: F0/Reflectance
- Sébastien Lagarde - DONTNOD PBR chart for Unreal Engine 4