

The Phong Model, Introduction to the Concepts of Shader, Reflection Models and BRDF

Contents

The Phong Model and the concepts of Illumination Models and BRDF

Keywords: specular reflection, glossy, roughness, Phong reflection model, BRDF, reflection model.

We do not expect to be able to display the object exactly as it would appear in reality, with texture, overcast shadows, etc. We hope only to display an image that approximates the real object closely enough to provide a certain degree of realism. Bui Tuong Phong

Please read the following three chapters for a quick introduction to shading and rendering: [A Light Simulator](#), [Light Transport](#) and [Shading](#).

The Phong Model

Before we dive into the concept of BRDF and illumination model, we will introduce a technique used to simulate the appearance of glossy surface such as a plastic ball for instance. From there, it will become easier to generalise the technique which is what the concept of BRDF and illumination or reflection model are all about.

In the previous lesson, we learned about simulating the appearance of mirror like and diffuse surfaces. But what about glossy surfaces? First you should note that the plastic ball example that we have just mentioned, is more than just a purely glossy surface. As most materials, it can be described as having both a **diffuse component** and a **glossy component** (the shiny or specular reflections that you can see in the ball from figure 1). The reason why many materials exhibit this dual properties is not always the same. In some cases, it is simply because the material is itself a composite of different materials. For example a plastic ball can be made of some sort of flakes or small particles acting as diffusers but glued together by a polymer that is itself acting as reflective (and often transmissive) material. Though the flakes or small particles diffuse lights while the polymer reflects light. In other cases, objects are made of several materials layered on top of each other. This is the case of the skin of many fruits. An orange for example has a thick skin layer that is acting more like a diffuse surface which is itself covered with a thin oily layer which is acting more like a specular or reflective surface. In summary we can describe the appearance of many material has having a diffuse component and a specular or glossy component. We could put this concept in a equation form:

$$S_P = \text{diffuse}() * K_d + \text{specular}() * K_s.$$

Where the term S_P here stands for the "shading at P". in this equation, the term $\text{diffuse}()$ is nothing less than the diffuse effect that we learned to simulate in the previous lesson The term $\text{specular}()$ is

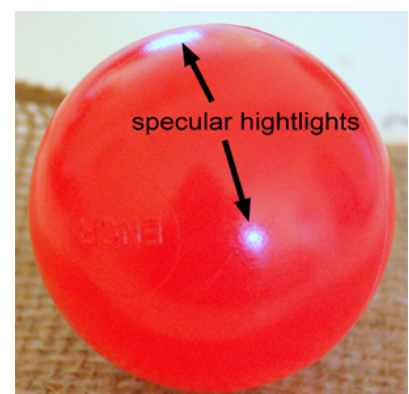


Figure 1: the specular highlights are just a reflection of the strongest sources of light in the scene surrounding the ball. The ball is both diffuse and specular (shiny).

new and will be used to simulate the glossy appearance of the object. The strength of both effects or to say it differently balancing one against the other can be controlled through the two parameter K_d and K_s . In shading and the world of computer graphics these terms are given many names and have caused a lot of ink to spill. You can look at them as the strength or gain of the diffuse and specular component. By tweaking them, one can create a wild variety of effect. But how these two parameters should be set is something we will look into later on. For now, let's focus on the specular function itself.

Bui Tuong Phong was a promising researcher in the field of computer graphics who sadly past way in 1975 soon after he published his thesis in 1973. One of the ideas he developed in this thesis was that indeed many materials could be computed from a sum of weighted diffuse and specular component. Readers interested in learning what causes some surfaces to be glossy are invited to read [the first chapter of the previous lesson](#). Glossy highlights are just the reflection of lights sources by the object. The phenomenon is similar to that of a perfect mirror-like surface reflecting an image of the light sources or an image of the object from the scene, though rather than being perfectly smooth like in the case of a mirror, the surface of a glossy material is slightly broken up (at the microscopic scale) which causes light to be reflected in a slightly different direction than the mirror direction (the waves of a water surface produce a similar effect as shown in figure 2). This has for effect to blur the light reflected off of the surface of the object. Because the surface of a rough surface acts like a broken mirror, computer graphics researchers like to define it as a collection of small mirrors which they also call **micro-facets**.



Figure 2: the waves of this water surface breaks the reflection of the background scene.

The concept of micro-facets here is purely a view of the mind and doesn't obviously "reflect" how the surface of a rough surface actually looks. Though representing rough surfaces a collection of micro-facets simplifies the resolution of mathematical equations which can then be used to simulate the appearance of rough surfaces. This is the foundation of the micro-facet shading models. You can find a lesson devoted to this topic in the next section.

Two observations are worth making at this point:

- The glossy reflection of a light source is dimmer than the reflection of that same light source by a mirror-like surface (we assume here that the viewer looks at the reflection of a light source along the light rays reflection direction as shown in figure 3). The reason for this is because only a fraction of the **micro-facets** (as we call them in CG) or small mirrors making up the surface of our glossy object, reflects light in the viewing direction, while in the case of a mirror like surface, all light rays are reflected in that direction (figure 3). In the case of a rough surface, only a fraction of the rays are reflected towards the eye when the observer looks down along the surface ideal reflection direction, hence the decrease in brightness.

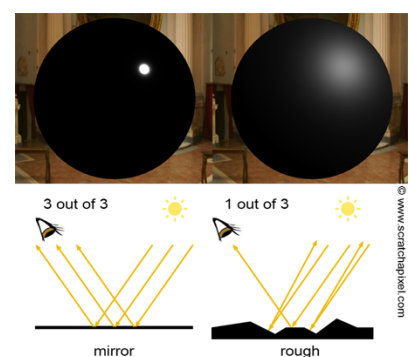
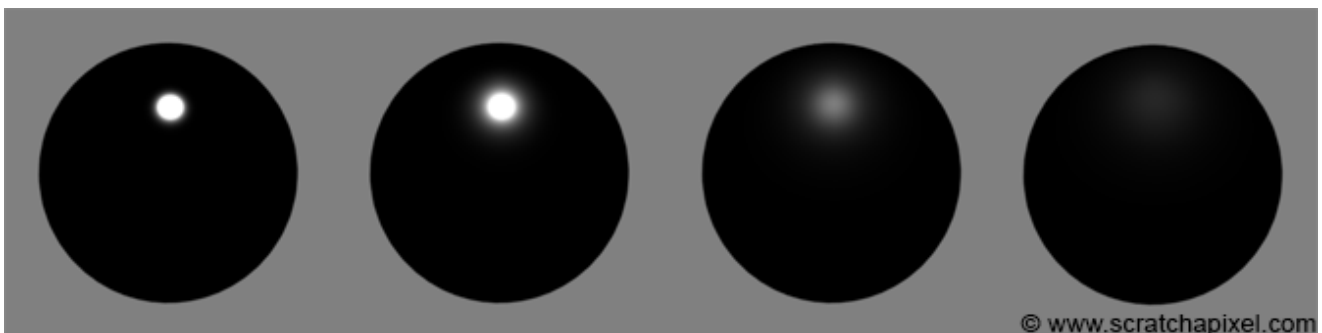


Figure 3: the rougher the surface, the larger and the dimmer the specular reflection. You can see this as the light being "blurred" across the surface of the object.

- The brightness of a glossy reflection decreases as the angle between the view direction and the ideal reflection direction increases. This is due to the fact that as this angle increases, the number of micro-facets reflecting light towards the eye decreases, hence the decrease in brightness. Basically the probability of finding a micro-facet reflecting light in the direction of the observer decreases as we walk away on the surface of the object from the point where the reflection of the light was observed when the surface was a perfect mirror. This is a statistical property of the way the micro-facets are distributed.

What's important in this last observation is basically that the brightness of the specular reflection decreases as the distance between a point on the surface of the object and the point where the reflection of the light source would be formed if the surface was a perfect mirror increases. This idea is illustrated in the following series of images. On the left you can see the reflection of a small light bulb by a perfect mirror. As we progress to the right, the surface roughness increases. Note how the reflection's brightness decreases and the light bulb's reflection spreads across a larger area. Note also that the highlight brightness decreases in intensity as the distance of points on the surface of the object to the original reflected light position increases.



Phong observed that it was possible to simulate this effect somehow by computing the ideal reflection direction of a light ray incident on the shaded point, and computing the dot product between this reflected ray and the actual view direction. As we know from the previous lesson, to see the reflection of a point on the surface of a mirror-like surface, the view direction or light of sight needs to perfectly coincide with the reflection direction. If these directions are different (even by a small amount), then the observer won't see the reflection of that point at all. When the two vectors are the same (when the view direction is parallel to the reflection direction), their dot product is equal to 1. As the angle between the view direction and the reflection direction increases, the dot product between the two vectors decreases (and eventually reaches 0).

$$\text{Specular} \approx V \cdot R.$$

Where V is the view direction and R is equal to:

$$R = 2(N \cdot L)N - L.$$

L is the incident light direction at P , the shaded point. You can see a plot of this equation in figure 5 (the red curve). Though Phong noticed that the curve has a pretty large shape which in itself would create a pretty large specular highlight. To solve this problem and shape the specular highlight, he raised the equation to the power of some value n (which is often called the specular exponent):

$$\text{Specular} \approx (V \cdot R)^n.$$

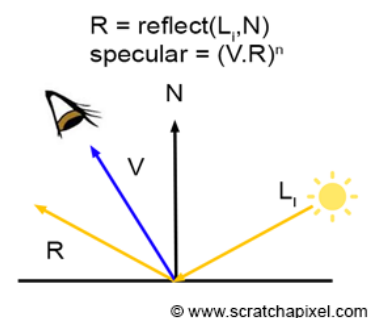


Figure 4: if the view direction is not perfectly aligned with the mirror direction, we don't see the reflection of the light source. Though we can take the dot product between the view direction and the reflection direction and this gives us an indication of how far the two vectors deviate from each other. This is the principle upon which specular reflections are simulated with the Phong model.

Figure 4 shows the shape that this equation has for different values of n . The higher the value, the narrower the curve, resulting in smaller, tighter specular highlight. If you apply this model and render a series of spheres with increasing values for n , here is what you get:

As you can see, some of these spheres start to look like shiny grey spheres. Though there is a problem. Since the probability that a micro-facet reflects light toward the viewer decreases as the roughness of the object increases, the overall brightness of the specular specular highlight should also decrease with n . In other

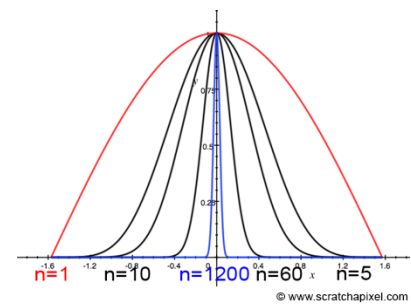
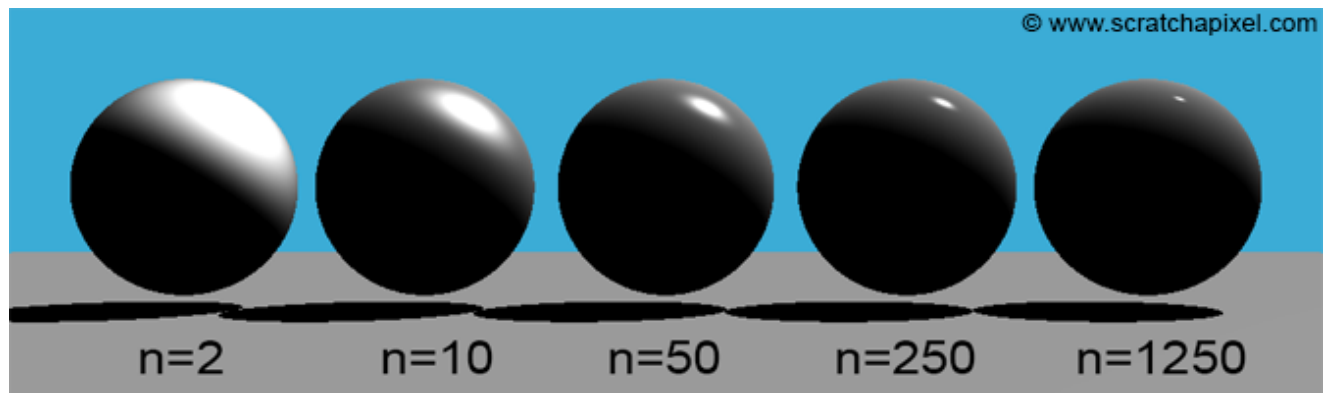
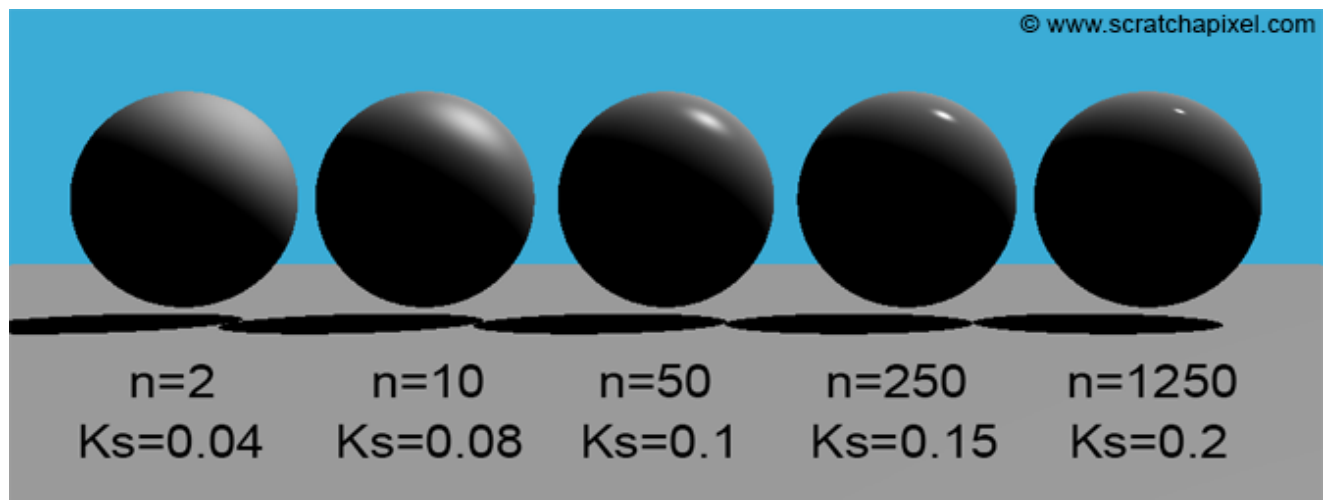


Figure 5: the shape of $(V \cdot R)^n$ for different values of n .



words, the larger the highlight, the dimmer it should be. Though this is clearly not the case in this render. Unfortunately Phong's model is **empirical** as he notices himself in his thesis, the numbers n and K_s have no physical meaning. In order to adjust the specular highlight intensity, you need to tweak the parameter K_s until you get the desired look.



Here is the code used to compute the images above:

```
001 Vec3f castRay(...)
002 {
003     ...
004     if (trace(orig, dir, objects, isect)) {
005         ...
006         switch (isect.hitObject->type) {
007             case kPhong:
008                 {
009                     Vec3f diffuse = 0, specular = 0;
010                     for (uint32_t i = 0; i < lights.size(); ++i) {
011                         Vec3f lightDir, lightIntensity;
012                         IsectInfo isectShad;
013                         lights[i]->illuminate(hitPoint, lightDir, lightIntensity,
```

```

014         isectShad.tNear);
015
016         bool vis = !trace(hitPoint + hitNormal * options.bias, -lightDir,
017         objects, isectShad, kShadowRay);
018
019         // compute the diffuse component
020         diffuse += vis * isect.hitObject->albedo * lightIntensity *
021         std::max(0.f, hitNormal.dotProduct(-lightDir));
022
023         // compute the specular component
024         // what would be the ideal reflection direction for this
025         // light ray
026         Vec3f R = reflect(lightDir, hitNormal);
027         specular += vis * lightIntensity *
028         std::pow(std::max(0.f,
029         R.dotProduct(-dir)), isect.hitObject->n);
030     }
031     hitColor = diffuse * isect.hitObject->Kd + specular *
032     isect.hitObject->Ks;
033     break;
034 }
035 default:
036     break;
037 }
}
else {
    ...
}

return hitColor;
}

```

Shading/Reflection Models

The model that Phong used to simulate the appearance of shiny material is what we call in CG a **reflection** or **shading** model. The reason why materials look the way they do is often the result of very complex interactions between light and the microscopic structure of the material objects are made of. It would be too complicated to simulate these interactions therefore we use mathematical models to approximate them instead. The **Phong model**, which is very popular because of its simplicity, is only one example of such reflection model but a wide variety of other mathematical models exist. To name just a few: the **Blinn-Phong**, the **Lafortune**, the **Torrance-Sparrow**, the **Cook-Torrance**, the **Ward anisotropy**, the **Oren-Nayar** model, etc.

The Concept of BRDF and The Rise and Fall of the Phong Model

As mentioned above what Phong essentially used to simulate the appearance of shiny materials is a function. This function (which includes a specular and a diffuse compute). This function contains a certain number of parameters such as n that can be tweaked to change the appearance of the material, but more importantly, it actually depends on two variables, the **incident light direction** (which is used to compute both the diffuse and specular component) and the **view direction** (which is used to compute the specular component only). We could essential write this function as:

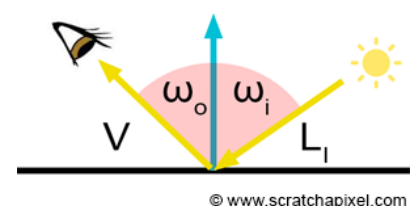


Figure 6: incoming I and outgoing V direction.

$$f_R(\omega_o, \omega_i).$$

Where ω_o and ω_i are the angle between the surface normal (N) and the view direction (V) and the surface normal and the light direction (I) respectively (figure 6). The subscript o stands for outgoing. In computer graphics, this function is given the fancy name of **Bidirectional Reflectance Distribution Function** or in short **BRDF**. A BRDF is nothing else than a function that returns the amount of light reflected in the view direction for a given incident light direction:

$$BRDF(\omega_o, \omega_i).$$

Any of the shading models that we mentioned above, such as the Cook-Torrance or the Oren-Nayar model are examples of BRDFs. You can also see a BRDF as function that describes how a given object scatters or reflects light if you prefer. As suggested, this amount of reflected light depends on both the incident light direction and the view direction. Many BRDFs have been proposed over the years: some are designed to simulate a specific type of material. For example the Oren-Nayar model is ideally suited to simulate the appearance of the moon which is not reflecting light exactly like a diffuse surface would. Some of these models were designed on either principles of optics or just to fit some physical measurements. Some other models such as the Phong reflection model are more empirical. A lesson is devoted to the concept of BRDF alone so don't worry if we just scratch the surface of the topic for now.

There are good and bad BRDFs. A bad BRDF is essentially one that breaks either one or more of the three following rules:

- First, a BRDF is a **positive** function everywhere over the range of valid incoming and outgoing directions.
- Two, a BRDF is **reciprocal**. In other words, $BRDF(\omega_o, \omega_i) = BRDF(\omega_i, \omega_o)$. If you swap the incoming and outgoing direction in the function, the function returns the same result.
- Finally, a BRDF is **energy conserving**. What this essentially means is that the BRDF can not create more light than it receives (well unless the surface is itself emissive but this is a special case). Overall an object can not reflect more light than the amount of light incident on its surface. A BRDF should naturally follow the same rule.

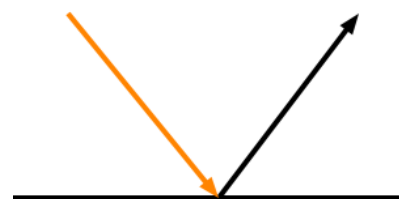
A good BRDF, is a BRDF that complies to these three rules. The problem with the Phong model, is that it is essentially not energy conserving. It would be too long to demonstrate this in this lesson, though if you read the lesson on BRDF [\[link\]](#), you will learn why. In general a few factors contribute to making a BRDF useful and good. It needs to be physically accurate, it needs to be accurate, and it needs to be computationally efficient (speed and memory consumption should both be considered here). The Phong model is not physically correct but is computationally efficient and compact which is why it was very popular for many years. The Phong model while considered still useful to teach people the basics of shading, is not really used anymore today and is replaced by more recent and more physically correct models.

The Phong model is also very useful to learn about more advanced rendering techniques. Its mathematical simplicity makes it an ideal candidate to learn and teach about importance sampling for instance. Check the lesson on importance sampling in the next section if you want to learn more about this topic.

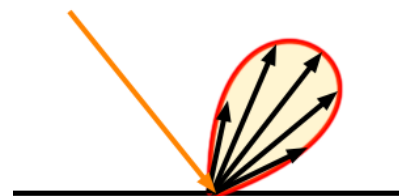
Note also that when used in conjunction with delta lights as opposed to area lights the result of the specular reflection looks nice but isn't physically accurate (beside the fact that the model is not energy conserving). The specular reflection being essentially a blurred reflection of a light source, the size of that reflection depends on the size of the source and its distance to the object (the closer the object to the surface, the larger the reflection). Since delta lights have no size by definition, the size of the specular reflection can't be physically accurate. The solution to this problem is to use area lights. You can find more information on this topic in the next section.

What Do You Mean by Specular or Diffuse Lobes?

We will finish this chapter on the concept of specular lobe which you may have heard or read about. When a surface is rough, it reflects light in directions slightly different from the perfect reflection direction but centred around it. You can draw or visualise this process by drawing some sort of elongated shape centred around the reflection direction as shown in figure 6. This represents in a way the set of possible directions that the surface reflects light into. This is what we call in CG, the lobe of the specular reflection. In figure 7 we represented this lobe as a two-dimensional shape but the shape should actually be three-dimensional. The shape of this lobe changes with the incoming light direction. This is a property that most materials have. They have a unique way of reflecting light for each possible incoming light direction. This lobe or the shape of this lobe can be quite complex for real material. We can acquire it using an instrument called a [gonioreflectometer](#). The function of this instrument is to measure how much light a given material reflects in every possible direction above its surface for a given incoming light direction. The shape of the resulting three-dimensional data totally depends on the material property. If the material is more diffuse the result will look like a hemisphere. If the material is more specular, there will be a long elongated lobe oriented about the reflection direction. Data acquired from real material is very useful to either derive mathematical models that can be used to simulate a given material or validate the accuracy of a given BRDF model. If the BRDF model behaves like the measured data (it reflects light the same way) then the model is a good candidate for simulating the appearance of the measured material. Note that when you look at the reflectance function of measured materials, you can see that they often have more than one lobe. In CG, we simulate this effect by combining several lobes with different parameters and weight. For example in figure 7, in the bottom figure we



mirror reflection



specular reflection



diffuse reflection



diffuse + specular

combined a diffuse and a specular lobe together. The resulting material should look both shiny and diffuse like with the Phong model. More information on this topic can be found in the next section.

Figure 7: diffuse and specular lobes. To simulate the appearance of complex materials you often need to combine several lobes together (like for example one diffuse lobe and 2 to 3 specular lobes with different specular exponent and weight).

And What About Other Types of Material

To the contrary of what it may seem, there is only a few types of materials. As mentioned earlier, objects' appearance is sometimes complex but only requires to combine different lobes together. The process by which we can find what the recipe for mixing these lobes together should be is generally complex, but the equations for creating the lobes themselves, are even the law of reflection or the Snell's law are themselves pretty simple. As mentioned in the previous lessons, we generally divide materials in two broad categories: dielectric and conductors. Conductors are essentially metals: they conduct electricity. In opposition, dielectrics are insulators. This includes plastic, pure water, glass, wood, rubber, wax, etc. Conductors (gold, aluminium, etc.) are essentially reflective and their appearance can easily be simulated using the reflection law. Though remember that you should use a different Fresnel equation for dielectrics and conductors. The appearance of many dielectrics also varies considerably. Water wood, plastic and wax have different appearances. Water and glass can be simulated using a combination of reflection, transmission and Fresnel. Wood can essentially be simulated using diffuse. Plastic can be simulated using a combination of diffuse and specular (the Phong model can be used to simulate plastic as it combines both components). Simulating wax is a slightly different problem because wax is **translucent**. In all the examples we studied so far, we considered that the point where a light ray strikes the surface and the point on the object where the same light is reflected back into the environment are one and the same. BRDFs are designed on the assumption that this is indeed the case. Though in the case of translucent materials, when a light ray strikes the surface on a object in a point x , it then generally travels through the material the object is made of and then eventually leaves the object in a different point x' . When the ray enters the objects, it is scattered one or multiples times by the structures that the material is made of. The ray generally follows some sort of random walk until it eventually leaves the object in some random position. BRDFs can't be used to simulate that sort of objects. We need another kind of model called a BSSRDF. This complex acronym stands for Bidirectional Scattering-Surface Reflectance Distribution Function. The phenomenon is also known as **subsurface scattering**. Check the next section, to learn more about BSSRDF and simulating the appearance of translucent materials.

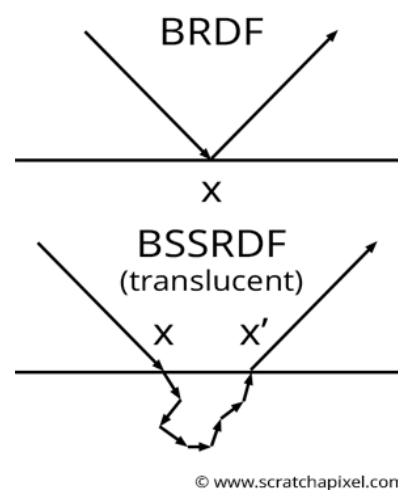


Figure 8: for BRDFs, we assume that the point where the light ray strikes the surface is the same from the point where the same ray is reflected by the object. For translucent objects, these two points are different. A light ray can strike the surface in point x , travel through the material the object is made of and then exit the object in point x' which is some distance away from x .

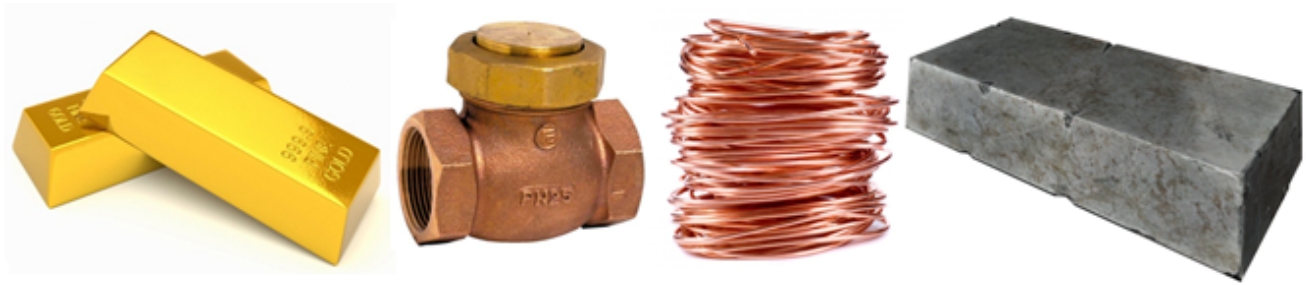
Should I Tint Specular Reflections with the Objects' Color?

This question is often asked by CG artists but even the most experimented ones sometimes fail knowing what the exact answer should be. The answer to this question is

NO

if the material is a dielectric, in other words, if it is not a conductor/metal. A specular reflection is only a reflection of a light source off of the surface of an object. Therefore the color of that reflected light

should be the same than the color of the light emitted by the light source. If the light source emits red light, then the specular highlight should be red. If the light source emits white light, then the specular highlight should be white. Never, ever tint a specular reflection especially with the color of the object. If the object is yellow and the light being reflected is white, the specular reflection won't be yellow. It will be white.



There is an exception to this rule though with metals. Some metals have a color (bronze, gold, copper) and though are purely reflective. Metals change the color of the specular reflections with their own color if you wish. For gold for example, it is a yellow color. For copper it is some dark orange-red color, etc. If you want to simulate the appearance of gold, you have to multiply the reflected light by the metal's color: yellow. Though if the metal is covered with a layer of paint, then what you simulate the appearance of is not the metal anymore but the paint layer, which on its own is a dielectric. Thus in this case, you should not tint specular reflections.