

Introduction to Shading

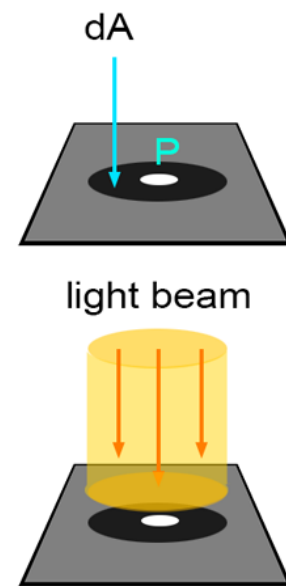
Contents

Diffuse and Lambertian Shading

Diffuse or Lambertian Shading

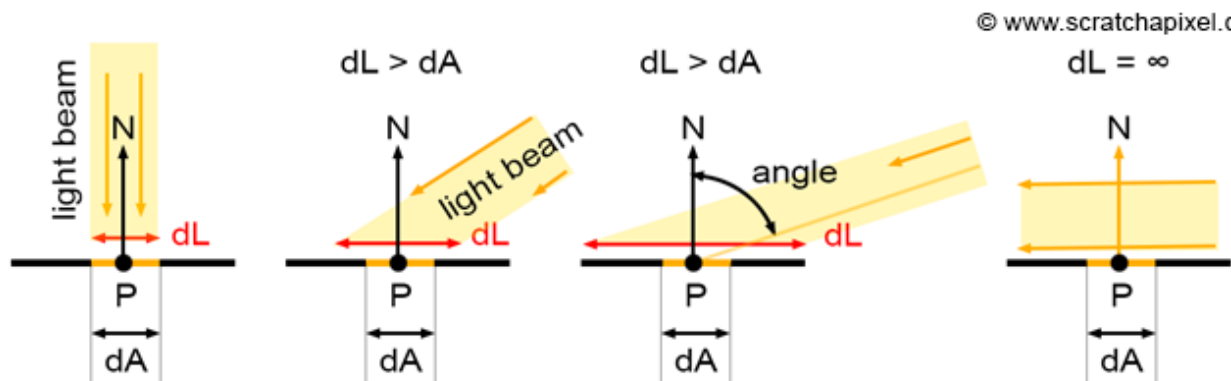
Diffuse objects are easy to simulate in CG. Though in order to understand how it works, you first need to learn something about the way light interacts with surfaces. Technically, to understand this topic properly, we should study radiometry first. Though to keep this introduction simple and intuitive we will try to do without it. To say things simply, when we actually shade a point in CG, we assume this point is not really a point but a very small surface which we call a **differential area**. Don't worry too much about the term differential here, and only pay attention to the term area instead. So, rather than considering a point which in the physical world has no meaning we will be considering a very small area around the point instead which we generally denote in CG dA . Now, what we actually consider as well is the amount of light falling on the surface of this very small area around P . Light that falls at P is not reduced to a single light ray since we are not interested in singular point but the small region dA around that point. Light that falls on this region is contained within a small volume perpendicular to P as shown in the left illustration of the following figure.

In other words all light energy contained within this volume which has the same cross section than dA falls on dA . If a constant flux of



© www.scratchapixel.com

Figure 1: a light beam illuminating a small surface dA .



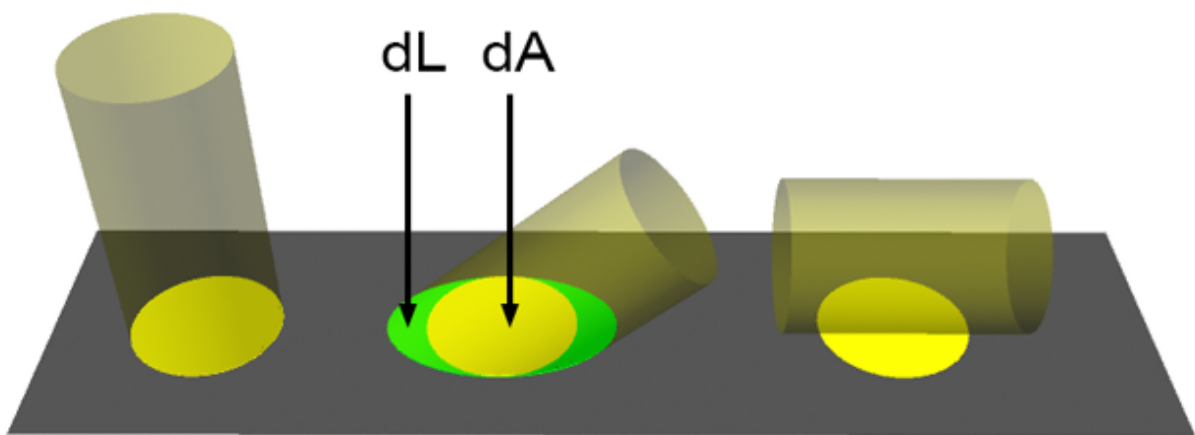
© www.scratchapixel.com

light energy travels through this volume as depicted in the left image of the above illustration, then one can assume that there's a constant amount of light energy striking the surface dA at any given time. Another way of saying this is, is to imagine that say 1000 photons strike the surface of dA at any given time. These 1000 photons will interact with the object surface. Some will be absorbed some will be reflected in (almost) the same exact time. A fraction of time later, 1000 new photons will strike the surface of dA again. If the flux of photons is constant, then this process keeps repeating itself. Then one can assume that since the cone of light perpendicular to dA has the same cross section than dA , dA is constantly bombarded by 1000 photons every fraction of time (which in CG we often refer to dt the same concept than dA but applied to time). Something interesting happens when this cone of light is angled with regard to the normal at P . As you can see in the second and third illustration of the above image, as the angle between the light beam and the normal at P increases, the cross section of the beam with the surface of the object becomes larger than dA . To say it differently, over the 1000 photons that used to strike dA only a fraction of these photons arrive at dA when the light beam

makes an angle with the normal. That fraction (the number of photons received by dA over the total number of photons contained in the beam) becomes smaller as the angle becomes larger. When the light beam is totally perpendicular to the object surface, then no photon strikes dA at all.

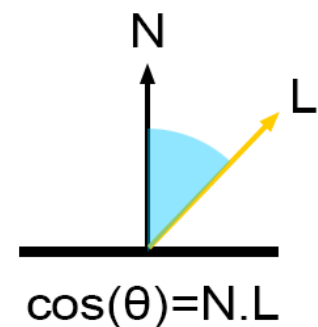
To summarize:

- When the light beam is perpendicular to dA 100% of the photons contained within the small cone of light whose cross section is the same as dA strike dA .
- When the light beam makes an angle with the normal of the shaded point, then the cross section of the beam with the surface becomes larger than dA which means that the photons or light energy is distributed over a larger region than dA . Since we are only interested in the amount of light energy arriving over the region covered by dA we can also say that fewer photons reach dA as the angle of the light beam with the normal increases. The larger the angle, the fewer the photons. Thus dA keeps receiving less and less energy as the angle between the beam and the normal increases.
- Finally, in the extreme case, when the beam is perpendicular to the normal at P , then photons do not fall on dA at all. In this particular case, dA does not receive any light energy at all.



© www.scratchapixel.com

From these observations it becomes easier to understand that a surface element dA receives a maximum of light energy when the beam is perpendicular to the surface, or in other words, when the surface normal N and the light direction L are parallel. Though when the light beam illuminates the surface from a certain angle, dA receives less light energy than when the light beam is perpendicular to the surface, which means that the surface itself will also reflect less light back into the environment. We can naturally deduce from this that the surface will get darker. As mentioned earlier, as the angle between N and L increases, dA is hit by less and less photons. In practical terms, this means that the surface gets darker and darker as the angle between the two vectors increases. Ultimately, when N and L are perpendicular, dA receives no light at all and is therefore black (since it receives no light, it doesn't reflect any either). More generally we can say that the brightness is proportional to the ratio dA over dL . Since dL , the oblique cross



© www.scratchapixel.com

Figure 2: the cosine law. The amount of light incident on the surface depends on the cosine of the angle between the light incident direction and the surface normal. The differential area dA

section of the light beam becomes larger as the angle between the two vectors increases, this ratio decreases as the angle increases.

receives more light energy when this angle is small.

This principle is one of the most basic and most well known phenomenon in CG. It is known under the name of the **Lambert's Cosine Law**. The amount of light that a surface receives is directly proportional to the angle between the surface normal N and the light direction L . This angle can be defined mathematically as:

$$\cos \theta = N \cdot L$$

Hence the term cosine in the law's name. The cosine of the angle θ (the Greek letter theta), is defined as the dot product of the vector N and L . If there is one thing you should remember from this lesson, it should probably be this law which is really fundamental. Of course the amount of light a surface reflects, and thus how bright it appears depends directly on the amount of light it receives. Thus we can somehow write, that the final color of a diffuse surface (and we will define in a very short moment what diffuse means here) is somehow proportional to the cosine of the angle between the two vectors:

$$\begin{aligned} \text{Diffuse Surface Color} &\propto \text{Incident Light Energy} * \cos \theta \\ &\propto \text{Incident Light Energy} * N \cdot L. \end{aligned}$$

The sign \propto here means "proportional to". Now, this is only one aspect of the problem. We at least now know about computing the amount of light arriving on a point of a diffuse surface, but to complete the picture and come up with a usable equation, we also need to know how much of that light a diffuse surface reflects back into the environment and especially in the view direction (the direction of the eye). First when light energy strikes the surface at P , some of that light is absorbed by the object and some of it is reflected back. As mentioned in the first chapter, this is defined by the **albedo** parameter of the surface. The albedo term defines the ratio of reflected light over the amount of incident light:

$$\text{albedo} = \frac{\text{reflect light}}{\text{incident light}}.$$

In computer graphics the term albedo is often denoted with the Greek letter ρ (rho). If we put the pieces of the puzzles together, then we have the amount of incident light which we know is proportional to the angle between the surface normal and the light direction, and we have the amount of light reflected by the surface which we know is proportional to the surface albedo. If we put these two components together, we can say that the amount of light reflected by a diffuse surface is equal to the amount of light it receives multiplied by the albedo (the ratio of incident light this is actually reflected by the surface, i.e. not absorbed):

$$\text{Diffuse Surface Color} = \text{albedo} = \rho_d * \text{Incident Light Energy} * N \cdot L.$$

This is almost the complete formula to compute the color of a point on a diffuse surface, knowing the incident light energy, the surface normal, the light direction and the surface albedo. We are only missing one term to get the complete equation.

As mentioned in the first chapter, diffuse surfaces have a unique property which is that they reflect light impinging on their surface in equal quantities in every direction above the surface at the point of incidence. Imagine a light beam striking the surface at a point and light energy being redistributed over that point in a hemisphere of directions expanding from the point of incidence outward, as shown in figure 3. From a practical point of view, this means that the energy of the light beam which we know is attenuated by the cosine of the angle between the surface normal and the light direction, is redistributed across the surface of a hemisphere. We can look at this problem the other way around and say that if we were to collect all the energy distributed across the surface of this hemisphere, it should be equal to the incident light

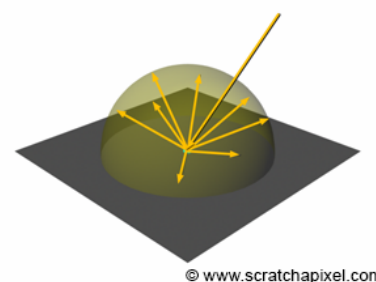


Figure 3: diffuse surfaces reflect light equally in all directions contained within a hemisphere of directions oriented about the surface normal.

energy minus of course the amount of light that was absorbed by the surface. There is as much light redistributed than light incident on the surface minus the amount of light absorbed by the surface. Mathematically, collecting the light energy across the surface of the hemisphere can be written using an integral:

$$\text{Amount of Reflected Light}(P) = \int_{\Omega} \rho_d * \text{Light Energy} * \cos \theta d\omega.$$

The integral symbol only means that we are interested in summing up in a way all the light energy that is spread across the surface of the hemisphere (imagine that you are counting the number of photons distributed over the surface of that hemisphere). The concept of hemisphere in this equation is represented by the term Ω . If you read this equation you could say "collect (the \int term) the light energy (aka the number of photons for example) of an incident light beam reflected by the surface at P , over the hemisphere (defined here by the term Ω) oriented about the surface normal N at the point of incidence". We know that the energy reflected is itself equal to the albedo of the surface multiplied by the amount of incident light energy multiplied by the cosine of the angle between the surface normal and the light direction. So if we put these two concepts together we get the integral term on the left (the green part in the equation) and what we try to collect which is the term in red on the right of the integral.

If you are not familiar with the concept of integral, we advise you to read the lesson on The Mathematics of Shading [link]. For a more formal introduction to shading we also advice you to read the second lesson from this section devoted to shading [link] as well as the lesson on radiometry [link].

We won't give much information about integral in this lesson nor explain what the term $d\omega$ means. Though you should know that collecting "some value" across the surface of some object (a half-sphere in this case) can be seen as just splitting the surface into small areas, measuring the quantity that we are interested in over the surface of these small areas and summing up the results. We do this because the quantity we want to measure might vary across the surface of the object. When the object is just a flat plane for instance, the area of a patch on this plane can just be measured in terms of square meter or inch or whatever units you wish to use to measure lengths. We would call this a differential area. We already introduced this concept earlier in this chapter and mentioned that differential areas are generally denoted dA . Though, when it comes to a sphere or a hemisphere, we don't use square meters but steradians instead. Steradian is the unit of solid angle, where a solid angle is basically a measure of a patch on a sphere expressed in terms of angle rather than length. The symbol used in CG for solid angle is most of the time ω . The term $d\omega$ thus refers to the concept of differential solid angle, or in more common terms a very small patch on the sphere but expressed in terms of solid angle. The goal with the integral mentioned above, is to measure some value (the value defined in red in our example) by splitting the surface of interest (a hemisphere in this case) into very small patches (the $d\omega$ over the surface of that hemisphere which because they are patches on the surface of a half sphere, are mathematically measured in terms of steradian), and summing up the value measured for each one of these small patches or differential solid angles.

Please refer to the lesson "The Mathematics of Shading" [link] for more information on integral and how they can be solved either analytically or using techniques such as Monte Carlo integration.

Now, hopefully you will agree that the amount of light that is reflected by the surface can't be greater than the amount of light that is effectively reaching the surface minus the amount of light that is absorbed. Logical right? In other terms if we could measure the amount of energy reflected by the surface using the equation above, it should be lower or equal to the amount of incident light energy:

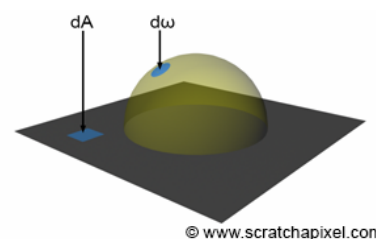


Figure 4: a differential solid angle is to a sphere what a differential area is to a plane or an arbitrary surface.

$$\text{Total Amount of Reflected Light(P)} = \int_{\Omega} \rho_d * L_i * \cos \theta \, d\omega \leq L_i.$$

Where L_i in here stands for the amount of incident light energy. Hopefully for us, the result of the integral in this particular case can be computed analytically. To say it simply, we know what the result of this integral is. Again, refer to the lesson on The Mathematics of Shading to learn about the way integrals can be solved. In this case, we can use the first fundamental theorem of calculus. The idea behind this theorem is that if you know the antiderivative F of some function f then it is possible to compute the result of the integral of f over some close intervals. This is often written as:

$$\int_a^b f(x)dx = F(b) - F(a).$$

The problem with our integral is that it is expressed in terms of solid angle (the $d\omega$ term) which is not an easy unit to work with. To make things simpler, we can replace the term $d\omega$ by $\sin \theta d\theta d\phi$ where the angle θ (the Greek letter theta) is defined over the interval $[0, \pi/2]$ and the angle ϕ (the Greek letter phi) is defined over the interval $[0, 2\pi]$. This idea is illustrated in figure 5. In other words, rather than using steradians we now use radians or angles which are easier to deal with. Though we now have to integrate over two quantities θ and ϕ which means that we need two integrals instead of one:

$$\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \rho_d * L_i * \cos \theta \sin \theta \, d\phi \, d\theta \leq L_i.$$

What we integrate over in an integral are the quantities that at the end of the equations are preceded by the letter d . In this example this would be ϕ and θ . In other words, the constant ρ and L_i are not affected and can be taken out of the integrals:

$$\rho_d * L_i \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \cos \theta \sin \theta \, d\phi \, d\theta \leq L_i.$$

Note also that we have the terms L_i on both sides of the \leq sign. If we divide both sides of the inequality by L_i we get:

$$\rho_d * \frac{L_i}{L_i} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \cos \theta \sin \theta \, d\phi \, d\theta = \rho_d \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \cos \theta \sin \theta \, d\phi \, d\theta \leq 1.$$

Also:

$$\int_a^b dx = b - a.$$

Thus:

$$\int_0^{2\pi} d\phi = 2\pi.$$

We can replace the first integral (the integral over ϕ) by 2π :

$$\rho_d * 2\pi \int_{\theta=0}^{\pi/2} \cos \theta \sin \theta \, d\theta \leq 1.$$

We can finally use the first fundamental theorem of calculus to solve the last integral. The antiderivative of $\cos x \sin x$ is:

$$-\frac{1}{2} \cos^2(x).$$

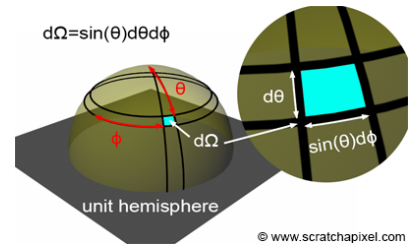


Figure 5: the differential solid angle can be defined in terms of the differential polar (θ) and azimuthal (ϕ) angles.

Thus:

$$\int_0^{\pi/2} \cos \theta \sin \theta d\theta = \left[-\frac{1}{2} \cos^2(x) \right]_0^{\pi/2} = -\frac{1}{2} \cos^2(\pi/2) - -\frac{1}{2} \cos^2(0) = 0 - -\frac{1}{2} = \frac{1}{2}.$$

Finally, we get the result:

$$\rho_d * 2\pi * \frac{1}{2} = \rho_d * \pi \leq 1.$$

The term ρ_d , the surface albedo can take any value in the range [0,1]. If we imagine that a surface has an albedo of let's say 0.5, then clearly the inequality above is wrong: $0.5 * 3.14$ is not lower than 1. So the equation we have above is not true for all possible values that the albedo can take on. The only solution for making this inequality working, is to actually divide the albedo itself by π :

$$\frac{\rho_d}{\pi} * \pi \leq 1.$$

And this would actually work. So the amount of light reflected back by a diffuse surface is in fact:

$$\text{Diffuse Surface Color} = \frac{\rho_d}{\pi} * L_i * \cos \theta.$$

Because if you integrate this equation over the hemisphere:

$$\text{Total Amount of Reflected Light(P)} = \int_{\Omega} \frac{\rho_d}{\pi} * L_i * \cos \theta d\omega \leq L_i.$$

Then you indeed find out that with this equation, a diffuse surface can not reflect more energy than it receives:

$$\rho_d \leq 1.$$

This division of the albedo by π has for effect to normalize the result of the integral in a way.

Now that we know the equation, let's put this in practice. Obviously this time, we will need to add a light to the scene. For simplicity, we will start with a distant light and one light only. We will learn about what we should do when there is more than one light in the scene and spherical lights in the next chapter. We will modify the source code of the ray tracer we developed in the previous lesson. First, each object in the scene now has an albedo parameter, which you can see as the color of the object:

```
001 | class Object
002 | {
003 |     Object(const Matrix44f &o2w, const Vec3f &c = 0.18) : objectToWorld(o2w),
004 |     worldToObject(o2w.inverse()), albedo(c)
005 |     ...
006 |     Vec3f albedo;
007 |     ...
    | };
```

Why is the default color 0.18? The reason we set the [albedo](#) default value to 0.18 is because object's from the real world reflect on average around 18% of the light they receive. This is an average value. In other words, if you look at how much light different fruits, asphalt, snow, tree leaves, grass, etc. reflect and take the average of all these values you end up with a value close to 18%. More information on this topic can be found in the lesson "Things to Know about the CG Lighting Pipeline" [\[link\]](#).

When a ray hits an object from the scene, we will first compute the surface normal at the intersection point. We then compute the pixel color associated with the camera ray using the equation we provided above:


```

001 Vec3f castRay(
002     const Vec3f &orig, const Vec3f &dir,
003     const std::vector<std::unique_ptr<Object>> &objects,
004     const std::unique_ptr<DistantLight> &light,
005     const Options &options)
006 {
007     Vec3f hitColor = options.backgroundColor;
008     float tnear = kInfinity;
009     Vec2f uv;
010     uint32_t index = 0;
011     Object *hitObject = nullptr;
012     if (trace(orig, dir, objects, tnear, index, uv, &hitObject)) {
013         Vec3f hitPoint = orig + dir * tnear;
014         Vec3f hitNormal;
015         Vec2f hitTexCoordinates;
016         hitObject->getSurfaceProperties(hitPoint, dir, index, uv,
017             hitNormal, hitTexCoordinates);
018         Vec3f L = -light->dir;
019         // compute color of diffuse surface illuminated
020         // by a single distant light source.
021         hitColor = hitObject->albedo / M_PI * light->intensity *
022             light->color * std::max(0.f, hitNormal.dotProduct(L));
023     }
024     return hitColor;
}

```

This is just a straight application of the equation. The albedo is divided by π which insures that the surface doesn't reflect more light than it receives, then the result is multiplied by the light intensity which is multiplied by the light color (this gives the total amount of light energy incident at the intersection point), attenuated by the cosine of the angle between the surface normal at the intersection point and the light direction.

Note that if the light and the surface normal are on the same side of the plane perpendicular to the surface normal, then the result of the dot product between the surface normal and the light direction is positive. Though if the light is "behind" the surface, the result of the dot product is negative. Of course, if the light is technically behind the surface it shouldn't illuminate the surface anymore, but more importantly we don't want to introduce negative values in the computation of the surface color. Thus we clamp the result of the dot product using the C++ `std::max()` function (line 20).

Here are some results:



Congratulations! You now know about two shading techniques. You know about facing ratio and simulating the appearance of perfect diffuse surface. In the next chapter, we will learn about handling more than one light source as well as using spherical lights. We will also learn about adding shadows to the image.