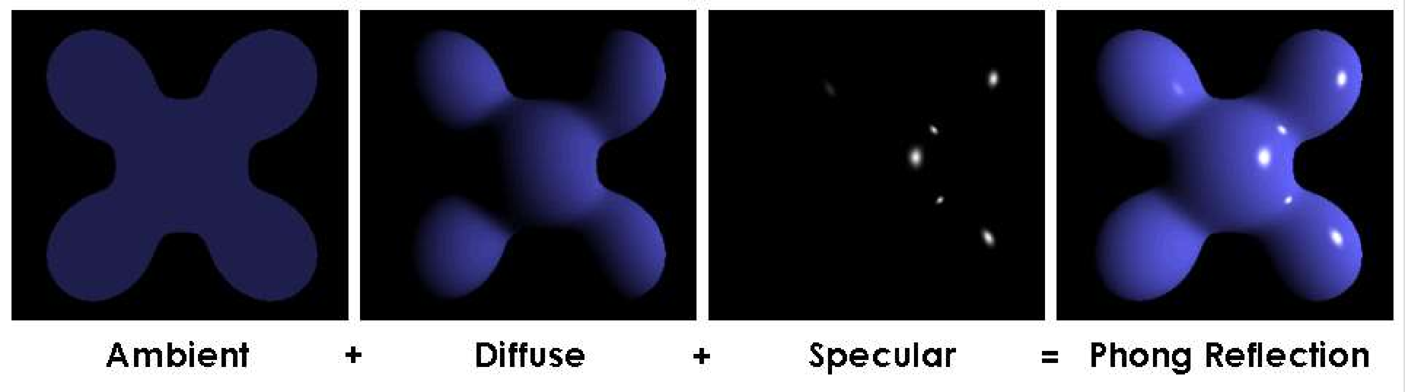
WIKIPEDIA
The Free Encyclopedia
WIKIPEDIA

Phong reflection model



Visual illustration of the Phong equation: here the light is white, the ambient and diffuse colors are both blue, and the specular color is white, reflecting a small part of the light hitting the surface, but only in very narrow highlights. The intensity of the diffuse component varies with the direction of the surface, and the ambient component is uniform (independent of direction).

The **Phong reflection model** (also called **Phong illumination** or **Phong lighting**) is an empirical model of the local illumination of points on a surface designed by the computer graphics researcher Bui Tuong Phong. In 3D computer graphics, it is sometimes referred to as "Phong shading", particularly if the model is used with the interpolation method of the same name and in the context of pixel shaders or other places where a lighting calculation can be referred to as “shading”.

History

The Phong reflection model was developed by Bui Tuong Phong at the University of Utah, who published it in his 1975 Ph.D. dissertation.[1][2] It was published in conjunction with a method for interpolating the calculation for each individual pixel that is rasterized from a polygonal surface model; the interpolation technique is known as Phong shading, even when it is used with a reflection model other than Phong's. Phong's methods were considered radical at the time of their introduction, but have since become the de facto baseline shading method for many rendering applications. Phong's methods have proven popular due to their generally efficient use of computation time per rendered pixel.

Concepts

Phong reflection is an empirical model of local illumination. It describes the way a surface reflects light as a combination of the diffuse reflection of rough surfaces with the specular reflection of shiny surfaces. It is based on Phong's informal observation that shiny surfaces have small intense specular highlights, while dull surfaces have large highlights that fall off more gradually. The model also includes an *ambient* term to account for the small amount of light that is scattered about the entire scene.

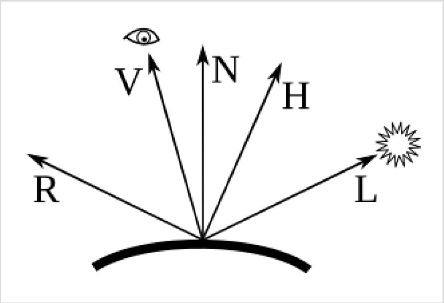
For each light source in the scene, components i_s and i_d are defined as the intensities (often as RGB values) of the specular and diffuse components of the light sources, respectively. A single term i_a controls the ambient lighting; it is sometimes computed as a sum of contributions from all light sources.

For each *material* in the scene, the following parameters are defined:

k_s, which is a specular reflection constant, the ratio of reflection of the specular term of incoming light,
k_d, which is a diffuse reflection constant, the ratio of reflection of the diffuse term of incoming light (Lambertian reflectance),
k_a, which is an ambient reflection constant, the ratio of reflection of the ambient term present in all points in the scene rendered, and
α, which is a *shininess* constant for this material, which is larger for surfaces that are smoother and more mirror-like. When this constant is large the specular highlight is small.

Furthermore, there is

lights, which is the set of all light sources,
$\hat{L}_m$, which is the direction vector from the point on the surface toward each light source (m specifies the light source),
$\hat{N}$, which is the normal at this point on the surface,
$\hat{R}_m$, which is the direction that a perfectly reflected ray of light would take from this point on the surface, and
$\hat{V}$, which is the direction pointing towards the viewer (such as a virtual camera).



Vectors for calculating Phong and Blinn–Phong shading

Then the Phong reflection model provides an equation for computing the illumination of each surface point I_p:

$$I_\text{p} = k_\text{a} i_\text{a} + \sum_{m\;\in\;\text{lights}} (k_\text{d} (\hat{L}_m \cdot \hat{N}) i_{m,\text{d}} + k_\text{s} (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,\text{s}}).$$

where the direction vector $\hat{R}_m$ is calculated as the reflection of $\hat{L}_m$ on the surface characterized by the surface normal $\hat{N}$ using

$$\hat{R}_m = 2(\hat{L}_m \cdot \hat{N})\hat{N} - \hat{L}_m$$

The hats indicate that the vectors are normalized. The diffuse term is not affected by the viewer direction ($\hat{V}$). The specular term is large only when the viewer direction ($\hat{V}$) is aligned with the reflection direction $\hat{R}_m$. Their alignment is measured by the α power of the cosine of the angle between them. The cosine of the angle between the normalized vectors $\hat{R}_m$ and $\hat{V}$ is equal to their dot product. When α is large, in the case of a nearly mirror-like reflection, the specular highlight will be small, because any viewpoint not aligned with the reflection will have a cosine less than one which rapidly approaches zero when raised to a high power.

Although the above formulation is the common way of presenting the Phong reflection model, each term should only be included if the term's dot product is positive. (Additionally, the specular term should only be included if the dot product of the diffuse term is positive.)

When the color is represented as RGB values, as often is the case in computer graphics, this equation is typically modeled separately for R, G and B intensities, allowing different reflection constants k_a, k_d and k_s for the different color channels.

When implementing the Phong reflection model, there are a number of methods for approximating the model, rather than implementing the exact formulas, which can speed up the calculation; for example, the Blinn–Phong reflection model is a modification of the Phong reflection model, which is more efficient if the viewer and the light source are treated to be at infinity.

Another approximation[3] that addresses the calculation of the exponentiation in the specular term is the following: Considering that the specular term should be taken into account only if its dot product is positive, it can be approximated as

$$\max(0, \hat{R}_m \cdot \hat{V})^\alpha = \max(0, 1 - \lambda)^{\beta\gamma} = \left(\max(0, 1 - \lambda)^\beta\right)^\gamma \approx \max(0, 1 - \beta\lambda)^\gamma$$

where $\lambda = 1 - \hat{R}_m \cdot \hat{V}$, and $\beta = \alpha/\gamma$ is a real number which doesn't have to be an integer. If γ is chosen to be a power of 2, i.e. $\gamma = 2^n$ where n is an integer, then the expression $(1 - \beta\lambda)^\gamma$ can be more efficiently calculated by squaring $(1 - \beta\lambda)$ n times, i.e.

$$(1 - \beta\lambda)^\gamma \;=\; (1 - \beta\lambda)^{2^n} \;=\; (1 - \beta\lambda)^{\overbrace{2 \cdot 2 \cdot \dots \cdot 2}^{n}} \;=\; (\dots((1 - \beta\lambda)^{\overbrace{2)^2 \dots)^2}^{n}}.$$

This approximation of the specular term holds for a sufficiently large integer γ (typically, 4 or 8 will be enough).

Furthermore, the value λ can be approximated as $\lambda = (\hat{R}_m - \hat{V}) \cdot (\hat{R}_m - \hat{V})/2$, or as $\lambda = (\hat{R}_m \times \hat{V}) \cdot (\hat{R}_m \times \hat{V})/2$. The latter is much less sensitive to normalization errors in $\hat{R}_m$ and $\hat{V}$ than Phong's dot-product-based $\lambda = 1 - \hat{R}_m \cdot \hat{V}$ is, and practically doesn't require $\hat{R}_m$ and $\hat{V}$ to be normalized except for very low-resolved triangle meshes.

This method substitutes a few multiplications for a variable exponentiation, and removes the need for an accurate reciprocal-square-root-based vector normalization.

Inverse model

The Phong reflection model in combination with Phong shading is an approximation of shading of objects in real life. This means that the Phong equation can relate the shading seen in a photograph with the surface normals of the visible object. Inverse refers to the wish to estimate the surface normals given a rendered image, natural or computer-made.

The Phong reflection model contains many parameters, such as the surface diffuse reflection parameter (albedo) which may vary within the object. Thus the normals of an object in a photograph can only be determined, by introducing additional information such as the number of lights, light directions and reflection parameters.

For example, we have a cylindrical object, for instance a finger, and wish to compute the normal $N = [N_x, N_z]$ on a line on the object. We assume only one light, no specular reflection, and uniform known (approximated) reflection parameters. We can then simplify the Phong equation to:

$$I_p(x) = C_a + C_d(L(x) \cdot N(x))$$

With C_a a constant equal to the ambient light and C_d a constant equal to the diffusion reflection. We can re-write the equation to:

$$(I_p(x) - C_a)/C_d = L(x) \cdot N(x)$$

Which can be rewritten for a line through the cylindrical object as:

$$(I_p - C_a)/C_d = L_x N_x + L_z N_z$$

For instance if the light direction is 45 degrees above the object $L = [0.71, 0.71]$ we get two equations with two unknowns.

$$\begin{aligned}(I_p - C_a)/C_d &= 0.71N_x + 0.71N_z \\ 1 &= \sqrt{(N_x^2 + N_z^2)}\end{aligned}$$

Because of the powers of two in the equation there are two possible solutions for the normal direction. Thus some prior information of the geometry is needed to define the correct normal direction. The normals are directly related to angles of inclination of the line on the object surface. Thus the normals allow the calculation of the relative surface heights of the line on the object using a line integral, if we assume a continuous surface.

If the object is not cylindrical, we have three unknown normal values $N = [N_x, N_y, N_z]$. Then the two equations still allow the normal to rotate around the view vector, thus additional constraints are needed from prior geometric information. For instance in face recognition those geometric constraints can be obtained using principal component analysis (PCA) on a database of depth-maps of faces, allowing only surface normals solutions which are found in a normal population.[4]

Applications

The Phong reflection model is often used together with Phong shading to shade surfaces in 3D computer graphics software. Apart from this, it may also be used for other purposes. For example, it has been used to model the reflection of thermal radiation from the Pioneer probes in an attempt to explain the Pioneer anomaly.[5]

See also

- Bidirectional reflectance distribution function – Function of four real variables that defines how light is reflected at an opaque surface
- Blinn–Phong shading model – Shading algorithm in computer graphics
- List of common shading algorithms
- Gamma correction – Image luminance mapping function

- Phong shading – Interpolation technique for surface shading
- Specular highlight – Bright spot of light that appears on shiny objects when illuminated

References

1. Bui Tuong Phong, Illumination for computer generated pictures (http://www.cs.northwestern.edu/~ago820/cs395/Papers/Phong_1975.pdf), Communications of ACM 18 (1975), no. 6, 311–317.
2. University of Utah School of Computing, http://www.cs.utah.edu/school/history/#phong-ref
3. Lyon, Richard F. (August 2, 1993). "Phong Shading Reformulation for Hardware Renderer Simplification" (http://dicklyon.com/tech/Graphics/Phong_TR-Lyon.pdf) (PDF). Retrieved 7 March 2011.
4. Boom, B.J.; Spreeuwers, L.J.; Veldhuis, R.N.J. (September 2009). Jiang, Xiaoyi; Petkov, Nicolai (eds.). *Model-Based Illumination Correction for Face Images in Uncontrolled Scenarios* (https://halshs.archives-ouvertes.fr/halshs-00420059/document). Lecture Notes in Computer Science. Vol. 5702. pp. 33–40. Bibcode:2009LNCS.5702.....J (https://ui.adsabs.harvard.edu/abs/2009LNCS.5702.....J). doi:10.1007/978-3-642-03767-2 (https://doi.org/10.1007%2F978-3-642-03767-2). hdl:11693/26732 (https://hdl.handle.net/11693%2F26732). ISBN 978-3-642-03766-5.
5. F. Francisco; O. Bertolami; P. J. S. Gil; J. Páramos (2012). "Modelling the reflective thermal contribution to the acceleration of the Pioneer spacecraft". *Advances in Space Research*. **49** (3): 337–346. arXiv:1103.5222 (https://arxiv.org/abs/1103.5222). Bibcode:2012AdSpR..49..579S (https://ui.adsabs.harvard.edu/abs/2012AdSpR..49..579S). doi:10.1016/j.asr.2011.10.016 (https://doi.org/10.1016%2Fj.asr.2011.10.016).

External links

- Phong reflection model in Matlab (https://web.archive.org/web/20160525135133/http://michal.is/projects/phong-reflection-model-matlab/)
- Phong reflection model in GLSL (https://web.archive.org/web/20180816064924/http://www.sunandblackcat.com/tipFullView.php?l=eng&topicid=30)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Phong_reflection_model&oldid=1253952959"