Optimized Phong and Blinn-Phong Glossy Highlights

Linus Källberg and Thomas Larsson
Mälardalen University

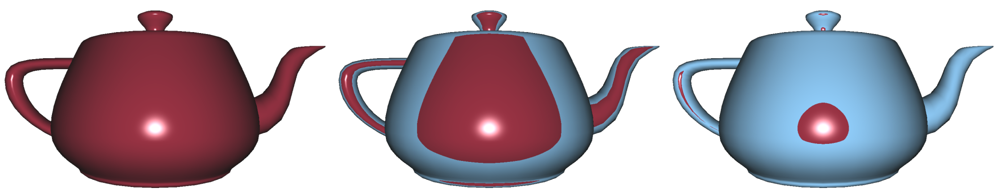


Figure 1. Phong-illuminated teapot with shininess exponent $\alpha = 50$. Blue color is used to visualize pixels where the glossy computation is skipped in a standard implementation (middle image) and using the proposed technique (right image).

Abstract

We describe a simple technique to compute the threshold at which glossy lobes in Phong or Blinn-Phong-like reflection models are negligible and thus avoid the cost of an exponentiation and a few other operations. The technique is trivial to incorporate into existing shaders. For existing shaders that already branch when the glossy term is zero, it introduces little overhead. SIMD or GPU shaders that do not already branch will incur the overhead of a branch and may experience divergence, which should be weighed against the potential advantages.

1. Introduction

In Phong's surface shading model [1975], the glossy term s is determined by the intensity i of the incoming light, the reflection coefficient m and shininess exponent α of the material, and the angle ϕ between the reflection and view directions:

$$s = \max(0, \cos\phi)^{\alpha} i m. \tag{1}$$

The intensity of light observed by the viewer is then proportional to s. This model has since been extended with energy conservation, physically-based rendering, correct accounting for the projected area of the surface, and multiple glossy lobes, but that is beyond the scope of our concern in this paper–the core idea is that a clamped

 ISSN 2331-7418

cosine power term underlies many modern shading models. Trigonometric functions are often relatively expensive to compute, however the value of $\cos\phi$ is typically evaluated efficiently by the dot product $\vec{r}\cdot\vec{v}$, where $\vec{r}$ and $\vec{v}$ are the normalized reflection and view vectors. In Blinn's extension [1977], ϕ is instead the angle between the half vector $\vec{h}$ and the surface normal $\vec{n}$. If both vectors have unit length, the cosine is again conveniently given by the dot product: $\cos\phi = \vec{h}\cdot\vec{n}$

To avoid the cost of exponentiation by α, it is common practice not to evaluate the full expression when $\cos\phi \leq 0$, because then $s = 0$ anyway. However, this condition turns out to be needlessly conservative in many cases: for even moderate values of α, the glossy contribution falls off rapidly outside the highlight areas and, although staying above 0, quickly becomes negligible. This suggests that there exists a threshold $\tau > 0$ such that the glossy term can be skipped when $\cos\phi \leq \tau$ without noticeable effects on the image.

2. Deriving the Threshold Value

When rendering colored surfaces, equation 1 is evaluated with different values of i and m for each color channel. We let the perceived contribution of the resulting glossy color be determined by its *luminance*. In RGB color spaces, luminance is commonly computed from the linear color components as $y = 0.2126\,r + 0.7152\,g + 0.0722\,b$, which accounts for the non-uniform sensitivity of the human eye to different wavelengths. Thus, the glossy luminance s_{y} is given by

$$s_{\mathrm{y}} = \max(0, \cos\phi)^{\alpha} c_{\mathrm{y}}, \tag{2}$$

where

$$c_{\mathrm{y}} = 0.2126\, i_{\mathrm{r}}\, m_{\mathrm{r}} + 0.7152\, i_{\mathrm{g}}\, m_{\mathrm{g}} + 0.0722\, i_{\mathrm{b}}\, m_{\mathrm{b}}. \tag{3}$$

Letting ε denote a lower bound on the perceptible level of luminance gives the following condition for skipping the glossy term:

$$\max(0, \cos\phi)^{\alpha} c_{\mathrm{y}} \leq \varepsilon, \tag{4}$$

which can be rewritten as

$$\cos\phi \leq (\varepsilon / c_{\mathrm{y}})^{1/\alpha}. \tag{5}$$

Thus, the threshold becomes $\tau = (\varepsilon/c_{\mathrm{y}})^{1/\alpha}$, which can be precomputed offline since it depends only on constants and parameters associated with the light source and material. Making the conservative assumption that $i_{\mathrm{r}} = i_{\mathrm{g}} = i_{\mathrm{b}} = 100\%$ for all light sources in the scene, equation 3 simplifies to

$$c_{\mathrm{y}} = 0.2126\, m_{\mathrm{r}} + 0.7152\, m_{\mathrm{g}} + 0.0722\, m_{\mathrm{b}}, \tag{6}$$

and τ needs to be computed only once for each material in the scene. Listing 1 demonstrates how to use this technique in practice by adding τ as an additional parameter in

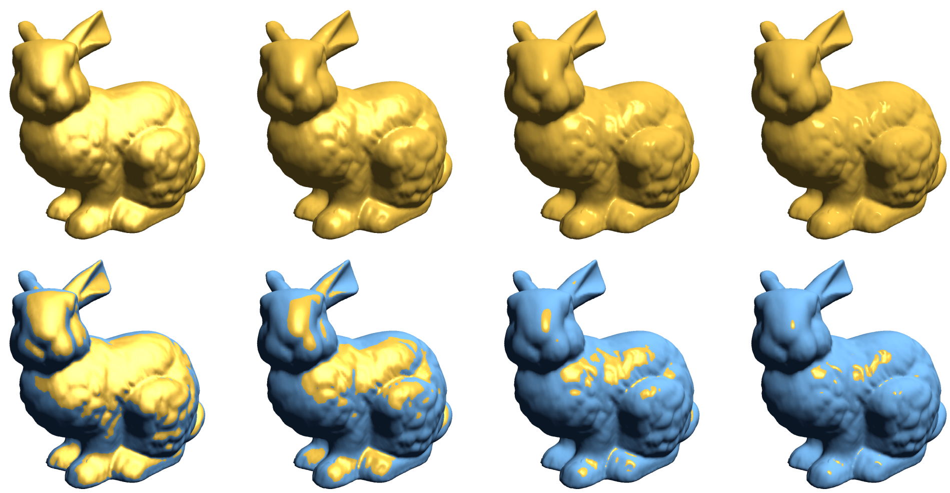
```
/* Preprocessing: initializes the 'tau' constant for all materials */
void initTau(Material mtrls[], int numMtrls) {
    for (int j = 0; j < numMtrls; j++) {
        vec3 m = mtrls[j].glossy;
        float alpha = mtrls[j].shininess;
        float cy = 0.2126f * m.r + 0.7152f * m.g + 0.0722f * m.b;
        mtrls[j].tau = pow(EPS / cy, 1.0f / alpha);
    }
}

/* Called during shading: computes the glossy term for a
   point on a surface with material 'mtrl' that is being
   illuminated by the light source 'light' */
vec3 phongGlossy(Material mtrl, Light light, vec3 r, vec3 v) {
    vec3 s;
    float r_dot_v = r.x * v.x + r.y * v.y + r.z * v.z;
    if (r_dot_v <= mtrl.tau) {
        // the glossy term is negligible
        s.r = s.g = s.b = 0.0f;
    } else {
        // compute the full glossy term
        vec3 m = mtrl.glossy;
        float alpha = mtrl.shininess;
        vec3 i = light.glossy;
        float p = pow(r_dot_v, alpha);
        s = p * vec3(i.r * m.r, i.g * m.g, i.b * m.b);
    }
    return s;
}
```

Listing 1. Example implementation of the proposed optimization technique in a C-like language.

the material description. Alternatively, memory permitting, c_y can be computed using equation 3 to give a τ for each light source/material combination, which can improve the results even further under dim lighting conditions.

Figure 1 shows an example where $\varepsilon = 10^{-3}$ is used to compute τ. The teapot is rendered with the Phong reflection model using $i = m = 1$ (maximum intensity) for each RGB channel and $\alpha = 50$, resulting in $\tau \approx 0.87$. It is clear that compared to the standard approach, which is equivalent to setting $\tau = 0$, the computations are reduced considerably. In general, with larger α, this effect is further enhanced. With smaller α, the positive effect diminishes gradually, but it is still noticeable even for $\alpha = 2$. Figure 2 shows this effect on a model with a more irregular surface than the teapot.

Since each light source is considered in isolation, the issue of overlapping glossy highlights from multiple light sources requires some consideration. Selecting ε too



$\alpha = 2$ $\alpha = 10$ $\alpha = 50$ $\alpha = 250$

Figure 2. The Stanford Bunny rendered with different Phong shininess exponents. The bottom row shows areas with skipped glossy computations in blue.

large can introduce rendering artifacts if several overlapping glossy terms are discarded although their accumulated contribution would in fact be visible. However, the largest such accumulated contribution from n light sources is $n\varepsilon$. Thus, given an ε value known to work well for scenes with one light source, the value $\varepsilon_n = \varepsilon/n$ is guaranteed to work for n light sources.

3. Conclusions

Variants of the Phong illumination model are often used in real-time rendering of massive scenes. The acceleration trick given here is applicable in both CPU and GPU systems, and may give both rasterizers and ray tracers an additional performance boost. Although the empirical results shown in figures 1 and 2 relate exclusively to the Phong glossy term, additional empirical tests suggest that even larger effects may be seen when the Blinn-Phong term is used, since the base condition corresponding to $\tau = 0$ tends to save less operations in this case, whereas the derived thresholds seem equally effective for equally sized highlights. This is easily motivated by the fact that $\cos\phi \leq 0$, where $\cos\phi = \vec{h} \cdot \vec{n}$, can hold only when either the light source, the viewer, or both are below the tangent plane of the surface.

Previous speed-up techniques for computing glossy highlights focus mainly on approximate reformulations of equation 1 to avoid the exponentiation (see, e.g., [Schlick 1994; Hast et al. 2003]). The approach taken here is not in competition with such strategies; it is rather a complement, since a combination of the approaches would reduce the computational cost even further. Of course, this assumes a corresponding derivation of τ.

Also, other glossy models may benefit similarly from the general idea presented here. For example, highlights that are more physically motivated can be created by using a Gaussian probability distribution to model the fraction of microfacets that are aligned with the half-vector $\vec{h}$ and, thus, effectively reflect the light towards the viewer [Blinn 1977]. For a simple such Gaussian glossy term, we get the following condition in analogy with equation 4:

$$e^{-(\phi/\omega)^2} c_{\mathrm{y}} \leq \varepsilon, \tag{7}$$

where $\phi = \arccos(\vec{h} \cdot \vec{n})$ and the constant $\omega \in (0,1]$ defines the roughness of the surface. By isolating ϕ, we get

$$\phi \geq \omega \sqrt{\ln(c_{\mathrm{y}}/\varepsilon)}. \tag{8}$$

As long the right-hand side belongs to $[0,\pi]$, the monotonicity of the cosine function on this interval allows the condition to be reformulated as

$$\cos\phi \leq \cos\left(\omega\sqrt{\ln(c_{\mathrm{y}}/\varepsilon)}\right). \tag{9}$$

Thus, $\tau = \cos\left(\omega\sqrt{\ln(c_{\mathrm{y}}/\varepsilon)}\right)$ can be used during shading to avoid expensive calls to the arccosine and exponentiation functions over surface areas where the Gaussian glossy term is negligible.

References

BLINN, J. F. 1977. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'77)*, ACM, 192–198. doi:10.1145/563858.563893. 2, 5

HAST, A., BARRERA, T., AND BENGTSSON, E. 2003. Fast specular highlights by modifying the Phong-Blinn model. In *ACM SIGGRAPH 2003 Sketches & Applications*, ACM, 1. doi:10.1145/965400.965406. 4

PHONG, B. T. 1975. Illumination for computer generated pictures. *Communications of the ACM 18*, 6, 311–317. doi:10.1145/360825.360839. 1

SCHLICK, C. 1994. A fast alternative to Phong's specular model. In *Graphics Gems IV*, P. S. Heckbert, Ed. Academic Press Professional, Inc., 385–387. 4

Author Contact Information

Linus Källberg linus.kallberg@mdh.se
Thomas Larsson thomas.larsson@mdh.se
School of Innovation, Design and Engineering
Mälardalen University
PO Box 883
S-72123 Västerås
Sweden

Linus Källberg and Thomas Larsson, Optimized Phong and Blinn-Phong Glossy Highlights, *Journal of Computer Graphics Techniques (JCGT)*, vol. 3, no. 3, 1–6, 2014
http://jcgt.org/published/0003/03/01/

Received:	2014-03-04		
Recommended:	2014-05-14	Corresponding Editor:	Joe Geigel
Published:	2014-07-20	Editor-in-Chief:	Morgan McGuire