One-Shot Approximate Local Shading

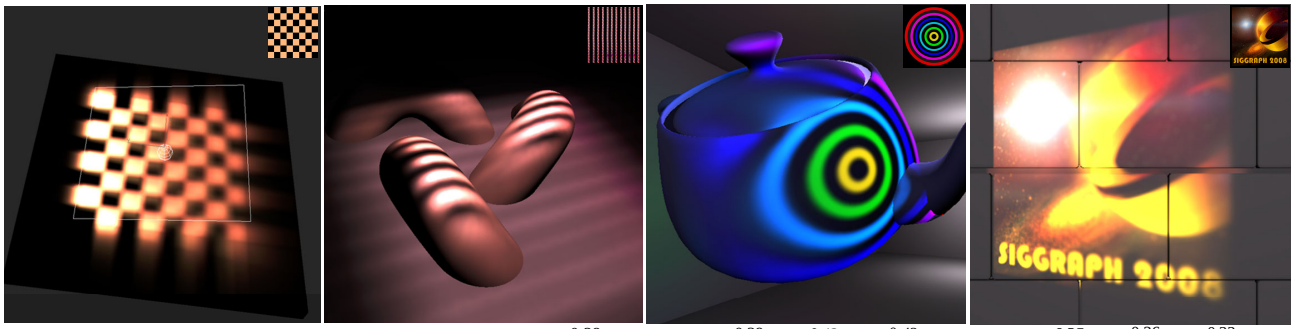
(Appear in Autodesk Technical Report soon)

Lifeng Wang
Autodesk Shanghai

Zhouchen Lin
Microsoft Research Asia

Wenle Wang
Autodesk Shanghai

Kai Fu
Autodesk Shanghai



(a) $\alpha_r = 0.35$ $\alpha_g = 0.37$ $\alpha_b = 0.29$ $\beta_r = 0.75$ $\beta_g = 0.77$ $\beta_b = 0.70$
(b) $\alpha_r = 0.41$ $\alpha_g = 0.44$ $\alpha_b = 0.38$ $\beta_r = 0.87$ $\beta_g = 0.80$ $\beta_b = 0.82$
(c) $\alpha_r = 0.39$ $\alpha_g = 0.42$ $\alpha_b = 0.42$ $\beta_r = 0.87$ $\beta_g = 0.80$ $\beta_b = 0.79$
(d) $\alpha_r = 0.35$ $\alpha_g = 0.36$ $\alpha_b = 0.33$ $\beta_r = 0.81$ $\beta_g = 0.80$ $\beta_b = 0.87$

Figure 1: Direct illumination results of area light sources with different patterns. The Phong shading model is used. The corresponding α's and β's in r, g, b channels for the light sources (see Section 3) are listed. Notice the near field and the far field effects of the light sources.

Abstract

In most rendering systems, area light sources are decomposed into many point light sources, which requires heavy computation to produce realistic results. In this paper, we propose a novel approach to approximate an area light source with a point light source for each component of the shading model. The position and intensity of the point light source is dependent on the position and orientation of scene points. Thus the scene is shaded using *on-the-fly* point light sources. The key contribution of our work is that we are able to shade with area light sources of arbitrary shape and *arbitrary intensity distribution* in real time and the shading model can be generic. And the computation cost is independent of the complexity of the area light source. Experimental results show that our approach creates results comparable to the ground truth. Our approach could be extended to other kinds of local light sources such as curved light sources and volume light sources.

Keywords: real time shading, light sources, direct illumination

1 Introduction

In computer graphics, most shading models have two main ingredients: surface properties and light source properties. Although much research has been done on shading realistic scenes, shading with a wide variety of light sources in real time is still an open problem.

There are mainly three kinds of light sources [C. Verbeck and D. Greenberg 1984]: point sources (zero dimensional), linear sources (one dimensional), and area sources (two dimensional). Point lights and directional lights are the most often used due to their simplicity. However, to increase the realism of the rendered images, more complex light sources are necessary [I. Ashdown 1993; I. Ashdown 1995; W. Heidrich et al. 1998; M. Goesele et al. 2003; X. Granier et al. 2003]. For such lights, people often have to decompose them into many point lights in order to achieve an acceptable approximation, resulting in heavy computation. In this paper, we present a novel approach to approximate a general light source, particularly an area light source of arbitrary shape and *arbitrary intensity distribution*, with a single point light for each component of the shading model[1], in order to achieve real time shading with complex lights.

There has been much work on shading with complex light sources. Verbeck and Greenberg [C. Verbeck and D. Greenberg 1984] approximate linear/area light sources by a series of collinear/coplanar point sources. Nishita *et al.* [T. Nishita et al. 1985] use multiple points to approximate the specular integral. And for the diffuse component numerical integration is also required when the linear light is neither parallel nor perpendicular to the object surface. These approaches require a large number of point lights, which makes them computationally expensive. An analytic solution for the diffuse and the specular component of linear lights is presented in [P. Poulin and J. Amanatides 1990]. Shading with area sources is also investigated in [K. Picott 1992], where the diffuse component is computed by contour integration and the specular component is evaluated at a point on the area light which has the maximum contribution to the specular illumination. The analytic methods in [P. Poulin and J. Amanatides 1990; K. Picott 1992] are valid only when the shading model is Phong [B. Phong 1975] *and* the light intensity is uniform. Although environment maps are area light sources with non-uniform intensity, they are usually treated as directional lights [R. Ramamoorthi and P. Hanrahan 2002]. Recently, based on the light field representation [M. Levoy and P. Hanrahan 1996; S. Gortler et al. 1996], several approaches [W. Heidrich et al. 1998; M. Goesele et al. 2003; X. Granier et al. 2003] have been proposed to simulate light photometry. These approaches can capture both the far field and the near field [I. Ashdown 1993; I. Ashdown 1995] illumination of a light source. However, the specific rendering requirements and the heavy storage of the data makes interactive visualization difficult.

In most previous work, it is so far not possible to use arbitrary local light sources in interactive applications. Thus offline techniques such as ray tracing have to be used for shading with com-

[1] It is theoretically possible to use a single point light source for the whole shading model. But things will be much easier if each component of the shading model is assigned a point source. See the end of Section 2.

plex light sources, including area light sources. Using closed-form solutions, the methods in [P. Poulin and J. Amanatides 1990] and [K. Picott 1992] might be the only possibility for real time shading with linear and area light sources. However, the closed-form solutions are derived by assuming modified Phong shading models and uniform distribution of the light intensity. Therefore, the methods in [P. Poulin and J. Amanatides 1990] and [K. Picott 1992] cannot be applied to general lights and generic shading models. Our paper attacks this problem by pushing to an extreme. In our approach, for each component of the shading model, the area light is represented by a *single on-the-fly* point light: the position of the point light is precisely computed to maximize the shading component, which is dependent on the position and orientation of the scene point, and the intensity is estimated by the total intensities in the local area surrounding the point light. The intensity estimation enables us to handle light sources with non-uniform intensity distribution and also naturally produces near field and far field effects [I. Ashdown 1993; I. Ashdown 1995] of the light sources. The computation cost is independent of the complexity of the area source. The basic methodology is also independent of specific shading models and can be implemented on any graphics hardware to shade the scene in real time. And it could be further extended to other kinds of local light sources, such as curved and volume lights.

However, currently our approach only handles direct illumination. The cast shadow, light interaction, and radiance transport are not involved. The remainder of the paper is organized as follows. In Section 2, we present the underlying theory of our approach. Using the Phong model [B. Phong 1975] as an example, Section 3 describes how to find the position and intensity of the point light source(s). The shading errors are also numerically evaluated. Then we show the experimental results in Section 4. Finally, we discuss the future work and conclude our paper in Section 5.

2 Basic Theories

In this section, we first present the basic theory of approximating an area light source with a point light source and then discuss some implementation issues.

Given an area light source S, its intensity distribution function $L(t)$, and a local shading model $\rho(p,t)$ which measures the contribution of a point t on the light source to a scene point p, the generated direct illumination at p is:

$$I(p,S) = \int_S \rho(p,t)L(t)\mathrm{d}S. \tag{1}$$

We prove that:

Theorem 1. *There exists a point t_0 on S and a corresponding radius $r_0 = r(p,t_0)$ such that*

$$I(p,S) = \rho(p,t_0) \int_{S\cap B_{t_0}(r_0)} L(t)dS, \tag{2}$$

where $B_t(r)$ is the disk centered at t and with a radius r. Namely, the shading result of the area light source S at the scene point p is identical to that of a point light positioned at t_0 and with an intensity $\int_{S\cap B_{t_0}(r_0)} L(t)\mathrm{d}S$, using the same shading model ρ.

Indeed, if we choose t_0 in the region of:

$$A(p,S) = \left\{ t \,\middle|\, \rho(p,t) \geq \frac{I(p,S)}{\int_S L(t')\mathrm{d}S} = \frac{\int_S \rho(p,t')L(t')\mathrm{d}S}{\int_S L(t')\mathrm{d}S} \right\}, \tag{3}$$

i.e., the part of the light source that contributes to the shading of p above the average (weighted by the intensity distribution $L(t)$), then in order to produce the same illumination at p the intensity of the point light should be:

$$\tilde{L}(t_0) = \frac{I(p,S)}{\rho(p,t_0)}, \tag{4}$$

which is between 0 and $\int_S L(t)\mathrm{d}S$ by the definition of the choice of t_0. Next, we define

$$g_{t_0}(r) = \int_{S\cap B_{t_0}(r)} L(t)\mathrm{d}S.$$

It is easy to see that

$$\lim_{r\to 0} g_{t_0}(r) = 0, \text{ and } \lim_{r\to\infty} g_{t_0}(r) = \int_S L(t)\mathrm{d}S.$$

So $g_{t_0}(r)$ is a continuous function of r (as long as $L(t)$ is bounded) and is between 0 and $\int_S L(t)\mathrm{d}S$. Therefore, by the intermediate value theorem of continuous functions, there exists an r_0 such that

$$g_{t_0}(r_0) = \tilde{L}(t_0). \tag{5}$$

Given the light source S, this radius r_0 is dependent on p and t_0. This completes the proof of Theorem 1.

To compute $\int_{S\cap B_{t_0}(r(p,t_0))} L(t)\mathrm{d}S$ in real time, we may pre-compute the convolution of $L(t)$ with the kernel function $\chi_{B_t(r)(t')}$:

$$\chi_{B_t(r)}(t') = \begin{cases} 1, & \text{if} \quad t' \in B_t(r), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

By padding the values outside S with zeros, we need not identify the region $S \cap B_{t_0}(r(p,t_0))$. The convolution can be efficiently computed using the FFT. It is easy to see that the disk in Theorem 1 can be replaced by a square. In this case, the computation is approximated by *Mipmap*, which can be easily generated by GPU.

For complex shading models, an economic choice of a single t_0 might be difficult. In this case, we may decompose the shading models into several simple components and apply Theorem 1 to each component independently. Then it is usually possible that t_0's are simply chosen as the points that maximize each component. And the corresponding radii can be estimated empirically. Hence the shading error comes from the inaccuracy of the estimated radii. Nonetheless, our method produces visually appealing shading results. All these will be shown in the following sections using the Phong model as an example.

3 Shot with Point Light Source

There have been many shading models in the literature. Popular shading models include the Phong model [B. Phong 1975], Blinn model [J. Blinn 1977], Cook-Torrance model [R. Cook and K. Torrance 1981], and Lafortune model [E. Lafortune et al. 1997]. In this paper, we use the Phong model as an example of applying the theories in Section 2, due to its simplicity and relative accuracy.

As illustrated in Figure 2, t is a point on an area light source S. $\vec{n}$ is the normal of S. p is a scene point, with a surface normal $\vec{N}$. $\vec{V}$ is the view direction and $\vec{N}_r$ is the *reflected* viewpoint vector w.r.t. $\vec{N}$. S has an intensity distribution function $L(t)$. Then Phong shading uses the following shading function

$$\begin{aligned} \rho(p,t) = \quad & \frac{\cos(\vec{pt},\vec{n})}{|pt|^2} \left\{ K_d \max(\cos(\vec{pt},\vec{N}),0) \right. \\ & \left. + K_s[\max(\cos(\vec{pt},\vec{N}_r),0)]^n \right\}, \end{aligned} \tag{7}$$

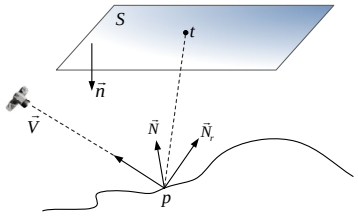


Figure 2: Illumination of a point p by an area light source S.

where K_d and K_s determine the proportions of diffuse and specular reflection, respectively.

Phong shading naturally consists of two components: diffuse and specular. Their shading models are respectively:

$$\begin{aligned} \rho_d(p,t) &= K_d \frac{1}{|pt|^2} \cos(\vec{pt}, \vec{n}) \max(\cos(\vec{pt}, \vec{N}), 0), \text{ and} \\ \rho_s(p,t) &= K_s \frac{1}{|pt|^2} \cos(\vec{pt}, \vec{n}) [\max(\cos(\vec{pt}, \vec{N_r}), 0)]^n. \end{aligned} \tag{8}$$

As mentioned at the end of Section 2, we may choose appropriate positions and radii of the point light sources for the diffuse and the specular components, respectively.

3.1 Diffuse

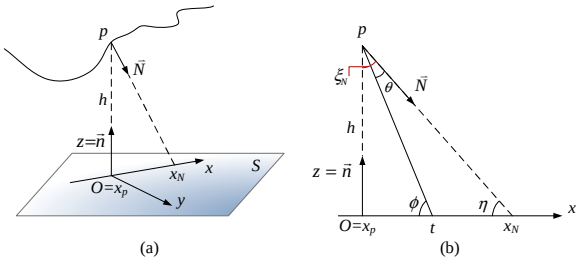


Figure 3: Diffuse illumination. (a) shows the chosen world coordinate and (b) shows the vectors in the x-z plane.

As $\rho_d(p,t)$ is a differentiable function w.r.t. t, it is easy to differentiate it to find its maximum point[2]. Choosing a world coordinate shown in Figure 3, the point t_1 that maximizes $\rho_d(p,t)$ is on the intersection line between planes $\vec{n}p\vec{N}$ and S and is a distance

$$x_p t_1 = \frac{h}{\tan(\theta + \eta)} \tag{9}$$

from x_p[3], where x_p is the orthogonal projection of p onto S. x_N is the intersection of the surface normal with S. The detailed derivation and interpretation of (9) can be found in Appendix A.

For the radius $r_d(p,t_1)$ that is used to estimate the intensity of the point light source (see Theorem 1), there may not be a closed form solution. We empirically assume that $r_d(p,t_1) = \alpha|pt_1|$, where α is a constant for a *given* area light source. With such choice of radius, when the light source is close to the surface, its pattern will appear sharply. And when the light source is far from the surface, the pattern becomes blurry.

[2]As the maximum point must be in the region that $\cos(\vec{pt}, \vec{N}) \geq 0$, we may drop the "max" operation in $\rho_d(p,t)$.

[3]As we pad zeros outside S and our algorithm can handle general intensity distributions, we need not further clamp t_1 to the real part of S as done in [K. Picott 1992].

So now we only have to choose an appropriate fixed value of α for the given area light source. Here we resort to a numeric solution. We may sample several points $p_1, p_2, ..., p_k$ with different positions and orientations and compute the exact shading at these points. Since the shadings equal those of the point lights t_1 at the corresponding positions, $\alpha_1, \alpha_2, ..., \alpha_k$ can be estimated via equation (5). Then the empirical α is estimated as the average of $\alpha_1, \alpha_2, ..., \alpha_k$. The α's are computed from r, g, b channels.

Figure 4(a) shows the numerically computed α values in r channel for different positions and orientations of p, where the light source is a disk with uniform intensity[4]. p is at a distance h from S and x_p is kept at the center of the disk. The distance h varies from 0 to 5 times the diameter of the disk light. We do not sample p's at farther distances because in that case an area light source could be treated as a point source [I. Ashdown 1993; I. Ashdown 1995]. And the normal $\vec{N}$ at p varies from pointing to S to parallel to S. With such choice of p's, the empirical α is estimated as 0.30. Using this constant α, Figure 4(b) shows the shading errors of the diffuse component in r channel, defined as the absolute error divided by the light intensity. One can see that the error is relative large only when p is very close to S and its normal is nearly parallel to S. For other combinations of the distance and the normal, the error is relatively small.

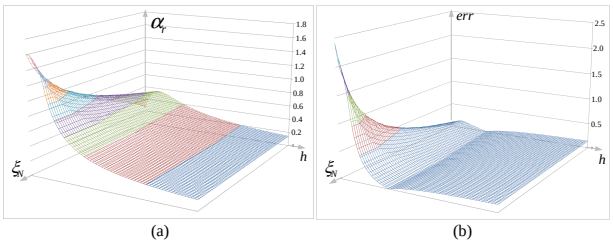


Figure 4: Estimated α's and the diffuse error for a uniform disk light. (a) The computed α's in r channel. (b) The diffuse error.

3.2 Specular

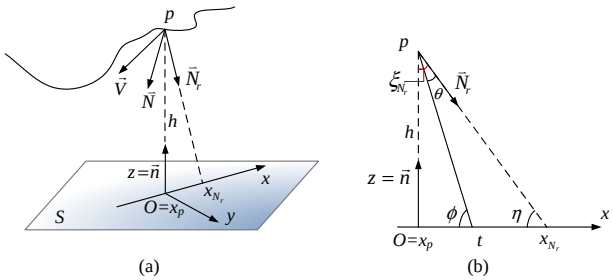


Figure 5: Specular illumination. (a) shows the chosen world coordinate and (b) shows the vectors in the x-z plane.

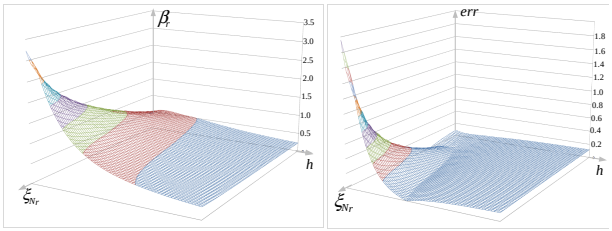
Figure 5 shows the chosen world coordinate and related parameters when considering the specular component. The computation and deduction is similar to the diffuse case, by simply replacing $\vec{N}$ with $\vec{N_r}$. Then we have a formula for the position t_2 of the point that maximizes the specular component, which is similar to (9):

$$x_p t_2 = \frac{h}{\tan(\theta + \eta)}, \tag{10}$$

[4]The purpose of Figures 4 and 6 is to show the distribution of exact α's and β's and the shading errors from the ground truth. In real applications, it is unnecessary to sample so many p's.

where the angles θ and η are now relative to the reflected viewpoint vector $\vec{N}_r$. The dependence of t_2 on $\vec{N}_r$ makes the point light source *view dependent*. As a result, the specular pattern of the area source is view sensitive. The detailed derivation and interpretation of (10) is in Appendix B. Note that the derivations for the specular component can be easily adapted for other shading models consisting of cosine lobes, such as the Blinn model [J. Blinn 1977] and Lafortune model [E. Lafortune et al. 1997].

Again, there is no closed form solution either for the radius $r_s(p, t_2)$ that is used to estimate the intensity of the point light source. We assume that $r_s(p, t_2) = \beta|pt_2|/n$, where β is a constant for a *given* area light source. The empirical value of β can be obtained numerically in a way similar to the diffuse case. The only difference is that we use different $\vec{N}_r$'s instead of different $\vec{N}$'s. Similarly the β's are computed from r, g, b channels.



(a) (b)

Figure 6: Estimated β's and the specular error for a uniform disk light. (a) The computed β's in r channel. (b) The specular error.

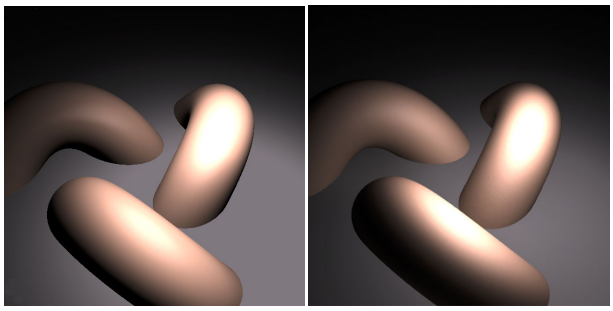
Figure 6(a) shows the numerically computed β's in r channel for different positions and reflected viewpoint vectors of p, where the setting is the same as the diffuse case except that $\vec{N}_r$ replaces $\vec{N}$ and $n = 3$ (see (8)). With such choice of p's, the empirical β is estimated as 0.57. Using this β, Figure 6(b) shows the specular error in r channel, also defined as the absolute error divided by the light intensity. One can see that for most combinations of h and $\vec{N}_r$, the error is quite small.

Note that Picott [K. Picott 1992] also used a single point to approximate the specular component. However, the point t_2 was chosen on S such that the angle between $\vec{N}_r$ and $\vec{pt}_2$ is minimized. And the intensity is simply $L(t_2)$. As a result, it cannot produce a blurry specular pattern when the light source is far away from the object surface.

4 Experimental Results

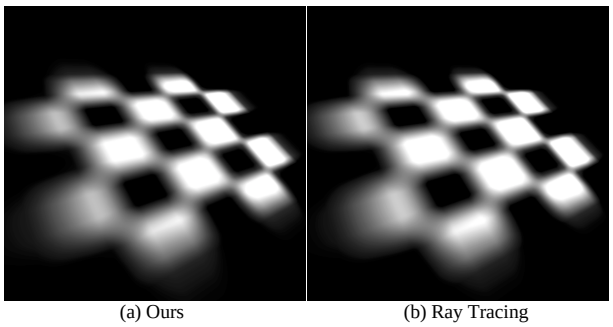
We implement our algorithm on an Intel Xeon PC with nVIDIA Qundro FX3450/4000 graphics card. We use Autodesk 3dsMAX as the rendering engine. In our experiments, the computation cost is tiny, which is the same as shading with a point light source and is independent of the complexity of the light source. And our shader can also run on other 3D graphics cards.

In our implementation, the convolution of the light source intensity distribution function $L(t)$ with the kernel function $\chi_{B_t(r)}$ is pre-computed and is stored in a 3D table with a float representation. The light source has a size of 128×128 and we store the convolution in a $256 \times 256 \times 9$ texture, which costs about 4 megabytes. For the i-th level texture, the central 128×128 pixels store the convolution of $L(t)$ with $\chi_{B_t(2^i)}$, $i = 0, 1, ..., 8$. The remaining pixels are for points outside the area source which are sampled more sparsely away from the center of the light source, e.g., at $(64 + j^2, 64 + k^2)$ $(j, k = 1, ..., 64)$ when the samples are in the first quadrant. And the corresponding convolution radii are $2^i \cdot \max(j^2, k^2)$. We use such compact storage of the convolution results because when the representative point source is outside the area source, it usually requires a larger radius than those inside the area source (because $|pt|$ will often be larger) and it could be quite far away from the area source. On the other hand, for points inside the area source, a maximum radius of 2^8 is already enough to cover the whole area source. So naively computing a huge texture to a much higher level[5] will cause an enormous waste of computation and storage. When rendering, if the representative point source and its radius are not at these samples, linear interpolation is used. And if either the position or the radius of the point source is out of range, the nearest sample is used.



(a) Ours (b) *Mental Ray*

Figure 7: Comparison with *Mental Ray*. The light source is a disk. The empirical values of α (r, g, b channels averaged) is 0.30 and β is 0.57.



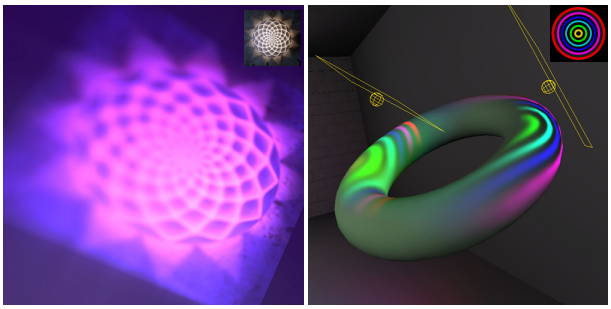
(a) Ours (b) Ray Tracing

Figure 8: Comparison with ray tracing. The light is a checkerboard pattern. The empirical values of α (r, g, b channels averaged) is 0.35 and β is 0.75.

Figure 1 shows shading results of using area lights with different patterns. From the outputs, we see that the shadings preserve visually correct light patterns. The rendering rate achieves 300 fps for a frame size of 800×600 pixels.

Figure 7 shows the comparison with the ground truth under a uniform disk light source. We use *Mental Ray* to produce the ground truth, which is widely used as an offline rendering standard in many 3D rendering systems such as 3dsMAX, Maya and MotionBuilder. One can see that our shading results are visually comparable with those by *Mental Ray*. Figure 8 shows the comparison to ray tracing with a checkerboard area light, i.e., densely sample points on the area light and then sum the contribution of these points. Our shading results are also comparable with the ray tracing results. Figure 9 shows more results using different lighting patterns. In these experiments, our simple algorithm produces visually correct light patterns, no matter how far the area source is from the object surface. More one-shot approximate shading results are presented in our submitted video.

[5]For example, to cover the same range of representative point sources and the convolution radii, the texture should be of size $(128 + 2 \cdot 64^2) \times (128 + 2 \cdot 64^2)$ and the highest level should be 20.



(a) (b)

Figure 9: More shading results using complex area lights. For the left light, α (r, g, b channels averaged) is 0.38 and β is 0.78, and for the right light, α is 0.39 and β is 0.84.

5 Discussion and Conclusions

In this paper, we have presented a novel approximate local shading algorithm for area lights. The algorithm has tiny computation cost, which is independent of the complexity of the light source, and can be implemented on general graphics hardware easily.

One key feature of our approach is that it can be applied to generic shading models represented as continuous functions, such as Phong [B. Phong 1975], Blinn [J. Blinn 1977], Cook-Torrance [R. Cook and K. Torrance 1981], and Lafortune [E. Lafortune et al. 1997]. The other key feature is that we could precisely compute the position of the desired point light, and give an approximation of the light intensity. The algorithm provides visually correct shading patterns under complex lights. Using the same methodology, 3D lights such as 3D curved and volume lights could also be represented by dynamic point lights, which hints at interesting future work.

However, as exposed in Introduction, there are limitations of the proposed approach. Firstly, we have not provided cast shadows, which improve realism. Secondly, the algorithm only handles direct illumination. Light transport or indirect illumination is currently out of scope. Thirdly, the approach approximates light intensity by using a numeric solution, which is difficult for quantitative analysis of the shading error. The above limitations suggest other possible future work of the paper.

References

B. PHONG. 1975. Illumination for computer generated pictures. In *Communication of the ACM*, vol. 18, 311–317.

C. VERBECK, AND D. GREENBERG. 1984. A comprehensive light-source description for computer graphics. In *IEEE Computer Graphics and Applications*, vol. 4, 66–75.

E. LAFORTUNE, S. FOO, K. TORRANCE, AND D. GREENBERG. 1997. Non-linear approximation of reflectance functions. In *Computer Graphics*, ACM SIGGRAPH Proceeding, 117–126.

I. ASHDOWN. 1993. Near-field photometry: A new approach. In *Journal of the Illuminating Engineering Society*, vol. 22, 163–180.

I. ASHDOWN. 1995. Near-field photometry: Measuring and modeling complex 3-d light sources. In *ACM SIGGRAPH'95 Course Notes - Realistic Input for Realistic Images*, 1–15.

J. BLINN. 1977. Models of light reflection for computer synthesized pictures. In *Computer Graphics*, vol. 11, ACM SIGGRAPH Proceeding, 192–198.

K. PICOTT. 1992. Extensions of the linear and area lighting models. In *IEEE Computer Graphics and Applications*, vol. 12, 31–38.

M. GOESELE, X. GRANIER, W. HEIDRICH, AND H. SEIDEL. 2003. Accurate light source acquisition and rendering. In *ACM Transaction of Graphics*, vol. 22, ACM SIGGRAPH Proceeding, 621–630.

M. LEVOY, AND P. HANRAHAN. 1996. Light field rendering. In *Computer Graphics*, ACM SIGGRAPH Proceeding, 31–42.

P. POULIN, AND J. AMANATIDES. 1990. Shading and shadowing with linear light sources. In *Eurographics'90*, EUROGRAPHICS Proceeding, 377–386.

R. COOK, AND K. TORRANCE. 1981. A reflectance model for computer graphics. In *Computer Graphics*, vol. 15, ACM SIGGRAPH Proceeding, 307–316.

R. RAMAMOORTHI, AND P. HANRAHAN. 2002. Frequency space environment map rendering. In *Computer Graphics*, ACM SIGGRAPH Proceeding, 517–526.

S. GORTLER, R. GRZESZCZUK, R. SZELISKI, AND M. COHEN. 1996. The lumigraph. In *Computer Graphics*, ACM SIGGRAPH Proceeding, 43–54.

T. NISHITA, I. OKAMURA, AND I. NAKAMAE. 1985. Shading models for point and linear sources. In *ACM Transaction on Graphics*, vol. 4, 124–146.

W. HEIDRICH, J. KAUTZ, P. SLUSALLEK, AND H. SEIDEL. 1998. Canned lightsources. In *Proceedings of the Eurographics Workshop on Rendering'98*, 293–300.

X. GRANIER, M. GOESELE, W. HEIDRICH, AND H. SEIDEL. 2003. Interactive visualization of complex real-world light sources. In *Pacific Graphics*, 59–66.

Appendix A: Derivation for the Diffuse Light

As analyzed, we may find t_1 that maximizes

$$\rho_d(p,t) = K_d \frac{\cos(\vec{pt},\vec{n})\cos(\vec{pt},\vec{N})}{|pt|^2} = K_d \frac{(\vec{pt}\cdot\vec{n})(\vec{pt}\cdot\vec{N})}{|pt|^4}.$$

As shown in Figure 3(a), we assume that $\vec{n}$ is the z direction of the world coordinate, x_p is the orthogonal projection of p to S, x_N is the intersection point of the surface normal with S (could be at infinity), and the plane $\vec{n}p\vec{N}$ is the plane of $y = 0$. Suppose the coordinate of p is $p = (0, 0, h)$, that of point t on S is $(t_x, t_y, 0)$, and the normal of object surface is $\vec{N} = (N_x, 0, N_z)$, where $N_x \geq 0$. Then we have

$$\rho_d(p,t) = K_d h \frac{t_x N_x - h N_z}{(t_x^2 + t_y^2 + h^2)^2}.$$

We can see that $\rho_d(p,t)$ is maximized only if $t_y = 0$ and $t_x N_x > h N_z$, i.e., t_1 must be on the axis $y = 0$. So we only have to determine the optimal t_x. Differentiating $\rho_d(p,t)$ w.r.t. t, we have

$$\frac{\partial \rho_d(p,t)}{\partial t_x} = K_d h \frac{N_x - 4\cos(\vec{pt},\vec{N})\frac{t_x}{|pt|}}{|pt|^4} = 0.$$

So t_x should satisfy

$$\cos(\vec{pt},\vec{N})\frac{t_x}{|pt|} = \frac{N_x}{4}. \tag{11}$$

To find out the geometric meaning of the above equation, we redraw the x-z plane in Figure 3(b). In the figure, we show the angles η, ϕ and θ, respectively. Then (11) is nothing but

$$\cos\theta\cos\phi = \frac{\cos\eta}{4}.$$

Using a trigonometry formula, we may rewrite the above as

$$\cos(\theta+\phi) + \cos(\theta-\phi) = \frac{\cos\eta}{2}.$$

Using the relationship $\phi - \theta = \eta$, where η is the known angle between $\vec{pt}$ and the x-axis (Figure 3(b)), we have that

$$\theta + \phi = \arccos\left(\frac{1}{2}\cos\eta - \cos\eta\right) = \arccos\left(-\frac{1}{2}\cos\eta\right).$$

Together with $\phi - \theta = \eta$, we have

$$\theta = \frac{1}{2}\left[\arccos\left(-\frac{1}{2}\cos\eta\right) - \eta\right]. \tag{12}$$

So we obtain

$$x_p t_1 = \frac{h}{\tan\phi} = \frac{h}{\tan(\theta+\eta)}.$$

Appendix B: Derivation for the Specular Light

For the specular component, we want to maximize

$$\rho_s(p,t) = K_s \frac{\cos(\vec{pt},\vec{n})\cos^n(\vec{pt},\vec{N_r})}{|pt|^2} = K_s \frac{(\vec{pt}\cdot\vec{n})(\vec{pt}\cdot\vec{N_r})^n}{|pt|^{n+3}}.$$

We choose the world coordinate as in Figure 5(a). Here the only difference from Figure 3(a) is that $\vec{N_r}$ replaces $\vec{N}$. The optimal point light t_2 also lies on the axis $y = 0$. Differentiating $\rho_s(p,t)$ w.r.t. t_x we obtain that the optimal t_x should satisfy

$$\cos(\vec{pt},\vec{N_r})\frac{t_x}{|pt|} = \frac{n}{n+3}N_{rx},$$

where $\vec{N_r} = (N_{rx}, 0, N_{rz})$ $(N_{rx} \geq 0)$.

Using the same geometric interpretation, we find that

$$\theta = \frac{1}{2}\left[\arccos\left(\frac{n-3}{n+3}\cos\eta\right) - \eta\right],$$

where η is the known angle between $\vec{pt}$ and the x-axis and θ is the angle between $\vec{pt}$ and $\vec{N_r}$ (Figure 5(b)). One can see that when $n = 1$, the above equation reduces to (12) but with different meaning of the angles. And when $n \to \infty$, $\theta = 0$, which corresponds to mirror specular reflection.

Again we have

$$x_p t = \frac{h}{\tan\phi} = \frac{h}{\tan(\theta+\eta)}.$$