Extensions of the Linear and Area Lighting Models

Kevin P. Picott
Alias Research Inc.
110 Richmond St. East
Toronto, Ontario
M5C 1P1

ABSTRACT

Techniques exist which can realistically render the diffuse effects of both one-dimensional (linear) and two-dimensional (area) lights on a surface. These techniques are described, and then extended to add the specular components for both light types.

CGA–KPP.02–0.50
January 11 19100
lighting.art

1. Introduction

Lighting models have always been a major part of creating realistic computer graphics imagery. Kajiya's rendering equation [Kaj86] details accurate calculation of the illumination of a scene. In practice it cannot be completely calculated so some approximations must be made. The accuracy of the lighting model depends on the approximations applied to this general model.

Traditionally, certain simplifying assumptions are made in order to make lighting calculations easier to perform. These have included:

Point Light	The light has zero size and a single position in space.
Directional Light	The light is infinitely far away so that all light rays from it are parallel.
Spot Light	The light has zero size, a single position in space, and a cone of influence with a fixed angular spread.

While these assumptions simplify the lighting calculations they also tend to detract from the realism of the scene. Recently, more complicated lighting models have been proposed that eliminate one or more of these assumptions. The most accurate of these is the Radiosity model put forth in [GR85]. Unfortunately, implementation of this model does not fit well into a system that employs traditional shading models.

Working towards this model we remove one of the more prominent assumptions from the point light model. The assumption we remove is that the light has zero size. We expand it first into one dimension, and then into two dimensions.

2. Linear Lights

Figure 1: Standard diffuse reflection model

The first light type we wish to tackle is a one-dimensional light, analagous to a fluorescent tube. It has a finite length, but no width or height. We approximate the linear light with an infinite series of point lights, lined up along the direction of the "tube".

For the purposes of this discussion we will refer to the standard diffuse lighting geometry (Figure 1).

$\vec{N}$ is the surface normal, $\vec{L}$ is the vector from the surface to the light source, θ is the angle between $\vec{N}$ and $\vec{L}$, and $\vec{R}$ is the reflection of $\vec{L}$ around $\vec{N}$.

2.1. Diffuse Intensity by Integration

To calculate diffuse intensity I_d we use the simple Lambert shading model:

$$I_d = I_p k_d \frac{\cos\theta}{|\vec{L}|^n} \tag{1}$$

k_d is the diffuse reflection coefficient of the surface. I_p is the intensity value of the point light source. The constant n is a decay coefficient, usually 1 or 2.

The easiest way of calculating cosθ is by using the dot product of the two vectors. We will assume that $\vec{N}$ is normalized and rewrite (1) as:

$$I_d = I_p k_d \frac{(\vec{L} \cdot \vec{N})}{|\vec{L}|^{n+1}} \tag{1a}$$

In order to extend this formula to the linear model we must integrate over all possible values of $\vec{L}$. We write $\vec{L}$ as a parametric function of t, with a starting point of L_0 and a direction of L_d.

$$\vec{L} = L_0 + t\,\vec{L_d} \quad ,t \ \varepsilon \left[0,1\right] \tag{2}$$

Now we can write out the full diffuse equation as an integral over t:

$$I_d = I_p k_d \int_0^1 \frac{((L_0 + t\vec{L_d}) \cdot \vec{N})}{|(L_0 + t\vec{L_d})|^{n+1}}\, dt \tag{3}$$

At this point we can make a further simplifying assumption concerning n. A shading model following physical laws would require $n = 2$, since light obeys an inverse square law. Empirical studies have indicated that other values of n give better results in certain circumstances. Typically, values of n between 0 and 3 are used, so we can restrict the integration to these values only. These integrations are straightforward, although somewhat long. For the interested reader the mechanics of the integrations appear in Appendix A.

2.2. Specular Intensity Methods

Now that we have a formula for diffuse lighting we would also like a formula for the specular component in order to implement a more interesting shading model, such as the Phong model [Pho75]. Recall that the specular illumination is given by:

$$I_s = I_p k_s \cos^n \alpha \tag{5}$$

In this equation I_s is the specular illumination, I_p is as before, k_s is the specular reflection constant of the surface, α is the angle between the sight vector $\vec{E}$ and the reflected light ray $\vec{R}$, and n is the reflection value of the surface ("shininess"). In this equation n is allowed to be any real number, with typical values between 0 and 200. A quick run through a symbolic math package should convince you that this equation will not be as nice to integrate as the diffuse equation. This leaves us with two options. We can either numerically integrate the equation, using one of several known methods, or we can approximate the equation.

Since this equation will be applied to every single vertex that uses our lighting model, numerical integration is too slow for practical use. Two other possible approaches examined were the creation of Chebyshev polynomials (too expensive to compute) and approximating $\int \cos^n \alpha$ with $(\int \cos\alpha)^n$ (dropoff was too severe and appeared unrealistic).

The method yielding the best results makes use of our original assumption that the linear light is approximated by an infinite series of point lights. We further approximate the specular contribution of the light to the contribution made by the point along the line that has the largest specular value for a given light ray. The idea is to find the single point among the infinite series that makes the largest contribution to the integral and assume the rest are negligible. This assumption works better with higher n values since the contribution from neighbouring points on the light become negligible, although the results obtained with low n values are satisfactory. (One can fine tune the approximation by sampling a neighbourhood around the optimal point whose size depends on the value of n and summing the values.)

The only remaining step is to find out which point on the line will make the largest contribution to the specular illumination. In equation (5) the only variable is α, therefore the largest contribution will be made when $\cos\alpha$ is as high as possible (note that we always restrict α to be positive). Glancing at our handy cosine curve we see that for positive angles the cosine is maximized when the angle is minimized. This means that the point we are looking for occurs where the angle between the sight vector $\vec{E}$ and the reflected light vector $\vec{R}(\theta)$ is the smallest.

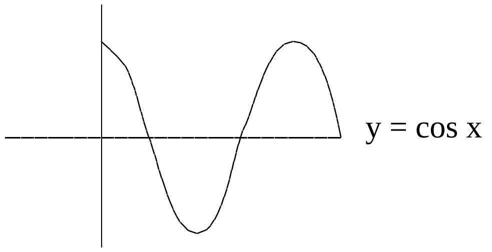

Figure 2: Linear light specular approximation

Notice that if we find the smallest angle between the light vector and the reflected sight vector we are also finding the minimal angle, θ, since the two angles are equal (Figure 2). We can now formulate the angle θ as a function of *t* using a dot product:

$$\theta = \frac{\vec{R} \cdot (L_0 + t\vec{L_d})}{|(L_0 + t\vec{L_d})|\,|\vec{R}|} \tag{6}$$

Minimizing θ we come up with a complicated, but easily calculated formula for the specular intensity: (For details on the calculation of θ see Appendix B).

$$t = \frac{((2L_0 \cdot \vec{L_d})(\vec{R} \cdot L_0) - 2(L_0 \cdot L_0)(\vec{R} \cdot \vec{L_d}))}{((2L_0 \cdot \vec{L_d})(\vec{R} \cdot \vec{L_d}) - 2(\vec{L_d} \cdot \vec{L_d})(\vec{R} \cdot L_0))} \tag{7}$$

This leaves a linear equation which is easily solved after checking for zero in the denominator. In this case a zero denominator would mean that the light is parallel to the reflected ray which implies zero specularity contribution.

We can see already that these simple additions have begun to make the image look more realistic (Plates 1a and 1b).

3. Area Lights

The next light type we wish to tackle is a two-dimensional light. This light is analagous to a light that is set into the ceiling. It has a finite length and width but no height (just as an inset light would not shine from the sides). The assumption we will make is that the area light can be approximated by an infinite series of point lights arranged in a rectangular array. In the specific calculations we will perform this is identical to an infinite series of linear lights arranged in a line perpendicular to their orientation. For greater generality the point model is used (ie. we can define arbitrary polygons instead of only rectangles using the point model).

3.1. Diffuse Intensity by Contour Integration

In the appendix of [NN85] on continuous tone representation the luminance calculation for a general polygonal light source is presented (Figure 3):

Figure 3: Contour integration geometry

The area light is defined by its boundary vectors (Q_{l+1} - Q_l, for each edge of the light source). P is the point on the surface being illuminated. β_l is the angle between $\overrightarrow{PQ_l}$ and $\overrightarrow{PQ_{l+1}}$. δ_l is the angle between the tangent plane of the surface at P and the plane defined by the three points P, Q_l and Q_{l+1}.

$$I_d = \frac{I_p}{2} k_d \sum_{l=1}^{m} \beta_l \cos\delta_l \tag{8a}$$

This equation is extended by Nishita to also include calculations of umbra and penumbra illumination by clipping the light source against the silhouette of all objects as viewed from the surface point and summing over the clipped pieces.

$$I_d = \frac{I_p}{2} k_d \sum_{pieces} \sum_{l=1}^{m} \beta_l \cos\delta_l \tag{8b}$$

These equations are very easy to calculate and take little computational power while yielding fairly realistic images. No further examination of the diffuse component of an area light source was needed at this point (Plate 1c).

3.2. Specular Intensity Methods

Since our specular approximation for linear lights was so successful we will apply the same reasoning to an area light. We will find the point on the light source where the angle between the reflected sight vector and the vector from the surface point to the light source is minimal (Figure 4). A few short mathematical derivations are used in this discussion, full details of which are shown in Appendix C.

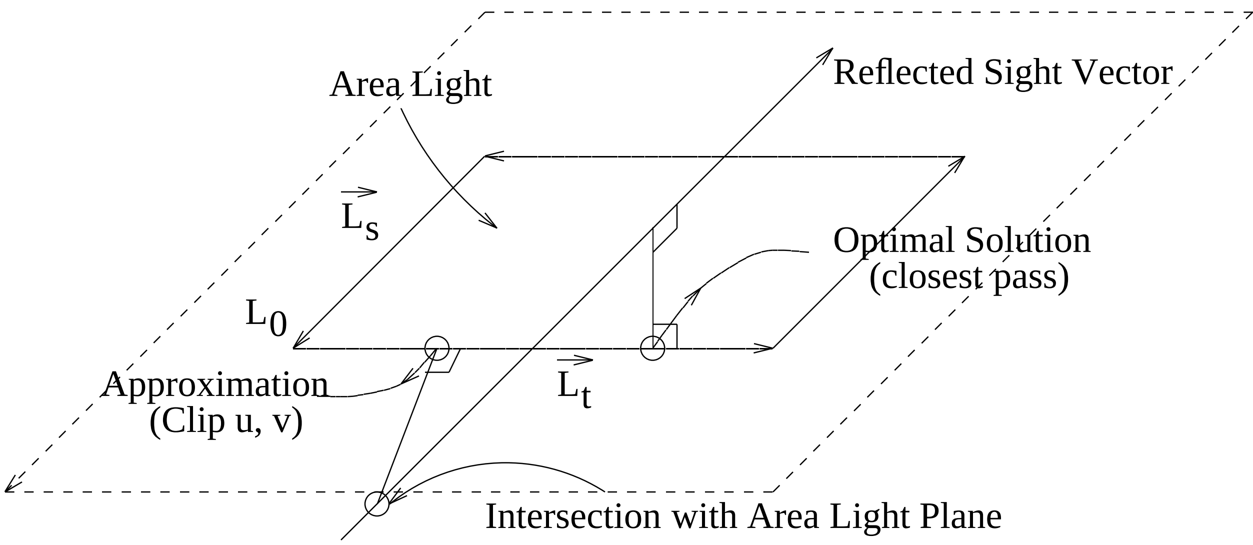

Figure 4: Area light specularity approximation

We define the light source by an origin point L_0 and two direction vectors $\overrightarrow{L_u}$ and $\overrightarrow{L_v}$ indicating the two axes that form the rectangular light. This definition ensures that the light is planar, while also giving us the flexibility of forming parallelograms. (As it turns out, parallelograms fold in nicely to both the diffuse and the specular algorithms but nonplanar lights do not. N-gons, where N>4 come for free in the diffuse algorithm but require a bit of extra work in the specular algorithms, although this work is not presented here.)

The first thing we do is to calculate the equation of the plane that the light source lies in from the three known points L_0, $L_0+\overrightarrow{L_u}$ and $L_0+\overrightarrow{L_v}$ as per [BW83]. This equation will be represented as:

$$Ax + By + Cz + D = 0 \tag{9}$$

Once we have the equation of the plane we can perform a line-plane intersection to find out where the reflected sight vector crosses the plane of the light source. Then we test the intersection point to see if it is inside the light source. This gives a trivial acceptance test. The intersection is relatively straightforward but the check for being inside the light source requires a little explanation.

If our intersection point is (x,y,z), we can convert this to (u,v) coordinates in the parametric space defined by an origin at L_0 and axes $\overrightarrow{L_u}$ and $\overrightarrow{L_v}$. After doing this, checking for the point being inside the light source is reduced to checking that the u and v parameters of the transformed intersection point are in the range $[0,1]$. This works for non-orthogonal axes since we are working solely in the u,v parametric space. For N-gons, N>4, we would require an N-axis clipping algorithm in place of this simple range check.

We will set up a transformation matrix in order to simplify calculations. In this case the transformation is going from 3-space (x,y,z) to 2-space (u,v) so the matrix will be 3 rows by 2 columns. We can define this matrix fully by noting that L_0 goes to $(0,0)$, $\overrightarrow{L_v}$ goes to $(0,1)$ and $\overrightarrow{L_u}$ goes to $(1,0)$. This gives us a linear system of six equations with six unknowns which can be solved with Gaussian elimination to give the transformation matrix $\overline{T}$.

We now apply the transformation to the intersection point:

$$(u\ v) = (x\ y\ z)\ \overline{T} \tag{10}$$

There are now three possibilities for (u,v):

Both in range We are done. The intersection point is on the light itself.

One of u, v out of range

We now have a good idea of where the closest point is. It lies along the axis that the out-of-range parameter has violated. For example, if u is in range but v is less than 0 the closest point will lie along the u axis, where u is between 0 and 1. (In real-space terms, it will lie along the vector $L_0 + t\overrightarrow{L_u}$ for $0 \le t \le 1$). It will not necessarily lie at $[u,0]$ since this implies that the reflected vector is perpendicular to the light source axis, which we cannot assume. In order to find the closest hit we can project the reflection vector on to the area light plane and intersect the projected line with the vector $L_0 + t\overrightarrow{L_u}$.

Both out of range

This will require a bit more work. Since we do not know specifically which axis was violated first we cannot tell which of the two possible axes the closest point lies along. The best thing to do here is to calculate the two closest points along each axis using the above method and then manually comparing the resulting angles to see which is smaller.

Now we can see a definite improvement in the realism of the image using the area light calculations over the point light approximations (Plate 1c).

4. Future Work

It seems that the application of Green's theorem used in [NN85] to solve the diffuse area illumination could be extended to also solve for specular illumination. A further extension would be to examine the mathematical techniques used to come up with these solutions and apply similar techniques to the calculations for linear lighting, hopefully yielding simpler formulae.

Even more interesting is the possibility of generalizing the contour integration formula to allow for 0 and 1 dimensional polygons. Currently, 0 dimensions would not work because the angle β_l is identically 0 and thus would cancel out all lighting contribution. The formula may work for 1 dimension but has not been tested by the author.

Real lights have three dimensions, not two, with the notable exception of inset ceiling lights. Since Green's theorem yielded such a nice result for two-dimensional lights it seems only natural that Gauss's theorem (the three-dimensional equivalent of Green's theorem) would give an equally nice result for three-dimensional lights. This is definitely an avenue for future investigation.

5. Conclusions

The extension of the lighting model to one and two-dimensional lights is a very powerful tool in realistic surface rendering. With a little bit of intuition and calculus we were able to implement these lighting models at a cost not much higher than traditional lighting techniques. Empirical approximations, where appropriate, also contribute to keeping the cost of lighting calculations low.

Some final images creating using the area lighting techniques are shown in the colour plates (Plates 2a, 2b, 2c).

6. Acknowledgements

I would like to acknowledge the readers of this paper, Jim Craighead, Andrew Pearce, Bob Leblanc, and Richard Sargent, all of Alias Research, without whom I would not have been able to organize this information so clearly. I would also like to thank Gary Mundell, also of Alias, for the colour plates of

the studio which dramatically highlight the practical use of these techniques.

7. Colour Plates

Plate 1a A bowling lane rendered with spotlights. Note the sharp shadows and cutoff area of the light. The crisp lighting is reminiscent of a studio setup, lacking other objects in the environment.

Plate 1b The same lane rendered with linear lights, using the techniques described above. Note the softness of the shadows and the way that the light blends more into the entire scene.

Plate 1c The same lane rendered with area lights. The shadows are almost nonexistent since the light is directly overhead. The pins do not self-shadow since some of the light reaches all parts of them.

Plates 2a-2c A more complex rendering of a studio scene. (Again plates a, b, and c correspond to spotlight, linear light, and area light renderings.)

8. Appendix A - Linear Light Diffuse Integration Calculations

To start off, we will simplify equation (3) by extracting the polynomial coefficients of t as follows:

$$A = L_0 \cdot \vec{N} \qquad B = \vec{L_d} \cdot \vec{N} \qquad C = L_0 \cdot L_0$$

$$D = 2\vec{L_d} \cdot L_0 \qquad E = \vec{L_d} \cdot \vec{L_d}$$

Equation (3) may now be written as a rational polynomial integral in t:

$$I_d = I_p k_d \int_0^1 \frac{A + Bt}{\sqrt{(C + Dt + Et^2)^{n+1}}}\, dt$$

Now *Mathematica*[TM] or a similar symbolic mathematics package can be used to solve for each value of n:

$$I_d = \frac{I_p\, k_d}{2E^{\frac{3}{2}}} \left(2B\sqrt{E}\left(\sqrt{C + D + E} - \sqrt{C}\right) + \right. \qquad \text{(n=0)}$$

$$\left. (BD - 2AE)\left(\log\left(\sqrt{C} + \frac{D}{2\sqrt{E}}\right) - \log\left(\frac{D}{2\sqrt{E}} + \sqrt{E} + \sqrt{C + D + E}\right)\right) \right)$$

$$I_d = I_p\, k_d \left(\frac{B}{2E}\left(\log(C) - \log(C + D + E)\right) + \right. \qquad \text{(n=1)}$$

$$(BD - 2AE)\frac{atan\left(\frac{D}{\sqrt{4CE - D^2}}\right) - atan\left(\frac{D + 2E}{\sqrt{4CE - D^2}}\right)}{E\sqrt{4CE - D^2}}\Bigg]$$

$$I_d = I_p\ k_d\ \left[\frac{2D\sqrt{C}(A - B) + 4AE\sqrt{C} + 4BC(\sqrt{C + D + E} - \sqrt{C}) - 2AD\sqrt{C + D + E}}{\sqrt{C + D + E}\,(4C^{\frac{3}{2}}E - D^2\sqrt{C})}\right] \qquad \text{(n=2)}$$

$$I_d = I_p\ k_d\ \left[\frac{2BC - AD}{C(4CE - D^2)} + \frac{AD - BD + 2AE - 2BC}{(C + D + E)(4CE - D^2)} - \right. \qquad \text{(n=3)}$$

$$\left.\frac{(4AE - 2BD)\left(atan\left(\frac{D}{\sqrt{4CE - D^2}}\right) - atan\left(\frac{D + 2E}{\sqrt{4CE - D^2}}\right)\right)}{(4CE - D^2)^{\frac{3}{2}}}\right]$$

Further optimization of expressions can be performed when coding the actual equations in order to minimize calculation time.

9. Appendix B - Linear Light Specular Angle Calculations

Starting with equation (6) we expand it out and rewrite it as a rational polynomial in *t*.

$$\alpha = \frac{\vec{R}\cdot L_0 + t\vec{R}\cdot\vec{L_d}}{|}((L_0 + t\vec{L_d})\cdot(L_0 + t\vec{L_d}))\,|\,|\vec{R}\,| \qquad \text{(6a)}$$

$$= \frac{1}{|\vec{R}\,|}\frac{\vec{R}\cdot L_0 + t\vec{R}\cdot\vec{L_d}}{\sqrt{L_0\cdot L_0 + t(2L_0\cdot\vec{L_d}) + t^2(\vec{L_d}\cdot\vec{L_d})}}$$

In order to keep the equations simple, we will substitute representative constants for the parameters of *t*:

$$A = \vec{L_d}\cdot\vec{L_d} \qquad B = 2L_0\cdot\vec{L_d} \qquad C = L_0\cdot L_0$$

$$D = \vec{R}\cdot L_0 \qquad E = \vec{R}\cdot\vec{L_d}$$

Differentiating with respect to *t* will enable us to find the extreme points of α:

$$\frac{d\alpha}{dt} = \frac{1}{|\vec{R}\,|}\frac{d}{dt}\frac{D + Et}{\sqrt{At^2 + Bt + C}} \qquad (7)$$

$$= \frac{1}{|\vec{R}\,|}\frac{E}{\sqrt{C + Bt + At^2}} - \frac{(B + 2At)(D + Et)}{(C + Bt + At^2)^{\frac{3}{2}}}$$

Now solve for the $\frac{d\alpha}{dt}$=0 (min/max points). (We can cancel the term $C+Bt+At^2$ because this is the length of the light. Special case handling can take over when this is 0.)

$$\frac{E}{\sqrt{C + Bt + At^2}} = \frac{(B + 2At)(D + Et)}{(C + Bt + At^2)^{\frac{3}{2}}}$$

$$2E(C + Bt + At^2) = (B + 2At)(D + Et)$$

$$2EC + 2EBt + 2EAt^2 = BD + BEt + 2ADt + 2AEt^2$$

$$t = \frac{(BD - 2CE)}{(BE - 2AD)}$$

To get the final formula, we can back-substitute the values we had for A,B,C,D and E:

$$t = \frac{((2L_0 \cdot \vec{L}_d)(\vec{R} \cdot L_0) - 2(L_0 \cdot L_0)(\vec{R} \cdot \vec{L}_d))}{((2L_0 \cdot \vec{L}_d)(\vec{R} \cdot \vec{L}_d) - 2(\vec{L}_d \cdot \vec{L}_d)(\vec{R} \cdot L_0))}$$

10. Appendix C - Area Light Specular Mathematics

10.1. Plane Equation from Three Points

The three points we can use to calculate the equation of the plane that the light source lies in are:

$$J = L_0 \quad K = L_0 + \vec{L}_v \qquad L = L_0 + \vec{L}_u \tag{11}$$

Using the notation that x_J means the x component of the point J we can write out the implicit form of the planar equation:

$$\begin{vmatrix} x - x_J & y - y_J & z - z_J \\ x_K - x_J & y_K - y_J & z_K - z_J \\ x_L - x_J & y_L - y_J & z_L - z_J \end{vmatrix} = 0 \tag{12}$$

Solving for this determinant and putting it into the planar equation form $Ax+By+Cz+D=0$ we come up with:

$$A = (y_K - y_J)(z_L - z_J) - (z_K - z_J)(y_L - y_J) \tag{13}$$

$$B = (z_K - z_J)(x_L - x_J) - (x_K - x_J)(z_L - z_J)$$

$$C = (x_K - x_J)(y_L - y_J) - (y_K - y_J)(x_L - x_J)$$

$$D = -(x_K((y_K - y_J)(z_L - z_J) - (z_K - z_J)(y_L - y_J)) +$$

$$y_K((z_K - z_J)(x_L - x_J) - (x_K - x_J)(z_L - z_J)) +$$

$$z_K((x_K - x_J)(y_L - y_J) - (y_K - y_J)(x_L - x_J)))$$

10.2. Line-Plane Intersection

This intersection is relatively straightforward. We start with the plane equation in standard form and the line equation in parametric form, where $x = x(t), y = y(t)$ and $z = z(t)$. All we have to do is substitute the parametric forms of x, y and z into the plane equation and solve for t. In our case the functions $x(t), y(t), z(t)$ can be expressed in vector form as $P + t\vec{R}$, where P is the surface point and $\vec{R}$ is the eyepoint reflection vector.

$$A(P + t\vec{R})_x + B(P + t\vec{R})_y + C(P + t\vec{R})_z + D = 0 \tag{14}$$

$$t = -\frac{AP_x + BP_y + CP_z + D}{A\vec{R}_x + B\vec{R}_y + C\vec{R}_z}$$

10.3. Transformation Between Planes

The problem is to transform a two-dimensional space defined by an origin point and two arbitrary axis vectors into a parametric space with origin (0,0) and axes (0,1) and (1,0). All we require is that this transformation is affine (preserves relative distances). Recall that what we want out of this space are the u, v coordinates of a point on the light source plane relative to its position, requiring that the range $u, v \ \varepsilon \ [0, 1]$ define exactly the transformed rectangular light source.

To solve for this transformation we set up a matrix $\bar{T}$ that will transform each point into its corresponding u-v values. $\bar{T}$ will be 3x2, consisting of 6 entries. Since we have three points to transform and each transforms to a pair of numbers we get a total of 6 equations and 6 unknowns:

$$\bar{T} = \begin{bmatrix} A & D \\ B & E \\ C & F \end{bmatrix}$$

$$L_0\bar{T} = (0,0) \qquad \vec{L}_u\bar{T} = (1,0) \qquad \vec{L}_v\bar{T} = (0,1)$$

$$L_0 \cdot [\ A\ B\ C\ ] = 0 \qquad L_0 \cdot [\ D\ E\ F\ ] = 0 \qquad L_u \cdot [\ A\ B\ C\ ] = 1$$

$$L_u \cdot [\ D\ E\ F\ ] = 0 \qquad L_v \cdot [\ A\ B\ C\ ] = 0 \qquad L_v \cdot [\ D\ E\ F\ ] = 1$$

Our 6 unknowns are A, B, C, D, E and F. The values $L_{0,} \vec{L}_u$ and $\vec{L}_v$ are constant. Given these 6 equations and 6 unknowns we can solve using traditional matrix elimination techniques, the details of which are omitted, to yield the final values:

$$W = L_{u_z} L_{0_y} \vec{L}_{v_x} - \vec{L}_{u_y} L_{0_z} \vec{L}_{v_x} - \vec{L}_{u_z} L_{0_x} \vec{L}_{v_y} + \vec{L}_{u_x} L_{0_z} \vec{L}_{v_y} + \vec{L}_{u_y} L_{0_x} \vec{L}_{v_z} - \vec{L}_{u_x} L_{0_y} \vec{L}_{v_z}$$

$$A = (\vec{L}_{u_z} L_{0_y} - \vec{L}_{u_y} L_{0_z})/W \quad B = (\vec{L}_{u_x} L_{0_z} - \vec{L}_{u_z} L_{0_x})/W \quad C = (\vec{L}_{u_y} L_{0_x} - \vec{L}_{u_x} L_{0_y})/W$$

$$D = (L_{0_z} \vec{L}_{v_y} - L_{0_y} \vec{L}_{v_z})/W \quad E = (L_{0_x} \vec{L}_{v_z} - L_{0_z} \vec{L}_{v_x})/W \quad F = (L_{0_y} \vec{L}_{v_x} - L_{0_x} \vec{L}_{v_y})/W$$

11. References and Bibliography

[BW83] Bowyer, Adrian and Woodwark, John. *A programmer's geometry*. Butterworths, 1983.

[Kaj86] Kajiya, James T. *The Rendering Equation*. In the Proceedings of SIGGRAPH, 1986

[NN85] Nishita, Tomoyuke and Nakamae, Eihachiro. *Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection*. In the Proceedings of SIGGRAPH, 1985.

[Pho75] Phong, Bui-Tuong. *Illumination for Computer-Generated Pictures.* Communications of the ACM, Vol.18, No.6, pp.311-317, 1975.

[GR85] Greenberg, Donald P. and Cohen, Michael F. *A Radiosity Solution for Complex Environments*. In the Proceedings of SIGGRAPH, 1985.

[Mathematica] is a trademark of Wolfram Research Inc.

Table of Contents

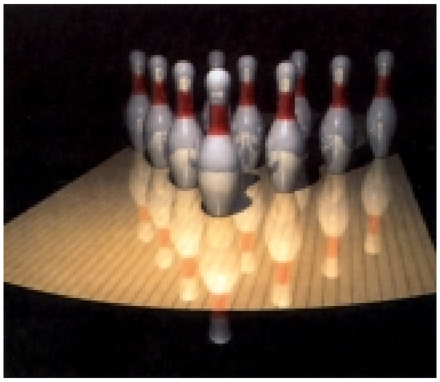

Plate 1a.

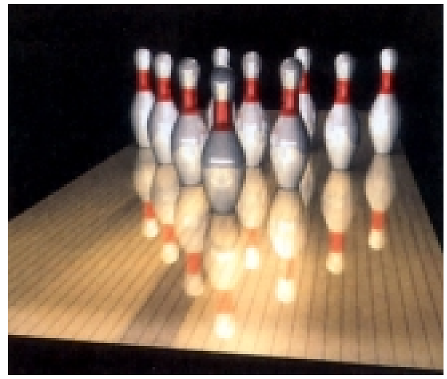

Plate 1b.

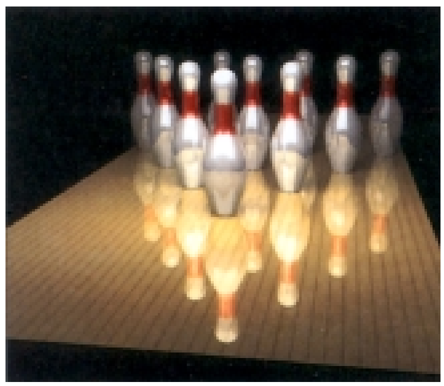

Plate 1c.

Plates 2a-2c.