Illumination and Reflection Maps:
Simulated Objects in
Simulated and Real Environments

Gene S. Miller
MAGI Synthavision
3 Westchester Plaza
Elmsford, New York 10523

C. Robert Hoffman 3
Digital Effects
321 West 44th Street
New York, New York 10036

23 Jul 1984
Course Notes for
Advances Computer Graphics Animation
SIGGRAPH 84

Abstract

Blinn and Newell introduced reflection maps for computer simulated mirror highlights. This paper extends their method to cover a wider class of reflectance models. Panoramic images of real, painted and simulated environments are used as illumination maps that are convolved (blurred) and transformed to create reflection maps. These tables of reflected light values are used to efficiently shade objects in an animation sequence. Shaders based on point illumination may be improved in a straightforward manner to use reflection maps. Shading is by table-lookup, and the number of calculations per pixel is constant regardless of the complexity of the reflected scene. Antialiased mapping further improves image quality. The resulting pictures have many of the reality cues associated with ray-tracing but at greatly reduced computational cost. The geometry of highlights is less exact than in ray-tracing, and multiple surface reflections are not explicitly handled. The color of diffuse reflections can be rendered more accurately than in ray-tracing.

CR Categories and Subject Descriptors:
I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism - Color, shading, shadowing and texture;
I.3.3 [Computer Graphics]: Picture/Image Generation: digitizing and scanning, display algorithms;
I.1 [Data]: Data Structures: tables

General Terms: Algorithms, Design

Additional key words and phrases: computer graphics, computer animation, shading, reflectance, illumination.

1. INTRODUCTION

The conventional point source lighting model [7] is used for efficient shading of simulated scenes, but it limits the kinds of scenes that can be rendered realistically by computer. This is because it does not adequately model the effects of area light sources (e.g. fluorescent lamps) and of objects in the environment, (i.e. the sky) that act as secondary light sources [16,19]. This is particularly noticeable on shiny surfaces.

A more realistic lighting model is also necessary when computer simulated objects are optically [17] or digitally matted [15] into photographs of real scenes. The lighting on a simulated object must be consistent with the real environment to create the illusion that it is actually in the scene.

A complete lighting model must account for all of the light that arrives at a surface which is directed towards the viewer. Blinn [2,5] obtained realistic mirror highlights by mapping the detailed reflection of complex environments. Whitted [25] provided a more accurate but slower solution for mirror highlights.

This paper presents an efficient and uniform framework for modeling light for realistic scene simulation. The idea of the illumination map is introduced -- a panoramic image of the environment that describes all the light incident at a point. Since the illumination map is an image, it can be created, manipulated and stored by conventional optical and digital image processing methods. Specific photographic and digital techniques are described.

The concept of reflection map is introduced. The shader looks up reflected light values in tables to model the appearance of specific materials under specific lighting conditions. These maps are obtained by blurring the illumination maps.

The resulting pictures have many of the reality cues associated with ray-tracing but at considerably less computational cost. The geometry of highlights is less exact than in ray-tracing, and multiple surface reflections are not explicitly handled. The color of diffuse reflections can be rendered more accurately than in ray-tracing.

2. HISTORICAL OVERVIEW

This section discusses three major illumination models. It concludes with a description of a real-time shader based on table-lookup.

2.1 Point Sources Illumination

The intensity calculation in Phong's shader [8] employs a finite number of point light sources at infinite distance. Intensity at a point on a surface is:

$$I = Ia + kd \sum_{j=l}^{ls} (N . Lj) + \sum_{j=l}^{ls} W(N . Lj)(E . Mj)**n$$

where

I = reflected intensity for the surface
Ia = reflection due to ambient light
kd = diffuse reflection constant
ls = the number of point light sources
N = unit surface normal
Lj = the vector in the direction of the jth light source
W = a function which gives the specular reflectance as a function of the angle of incidence
E = the direction of the viewer
Mj = the direction of the mirror reflection from the jth light source = 2(N . Lj)N - Lj
n = an exponent that depends on the glossiness of the surface.

The number of calculations per point is proportional to the number of light sources. Phong shading is most realistic when the simulated environment is mostly one color (e.g. the black of outer space) or when surfaces are not too glossy. It does not deal with indirect lighting.

2.2 Ray Tracing

Whitted [25] presents a simple and general method that realistically simulates mirror highlights as well as shadows, transparency and multiple reflections. The intensity of reflected light is:

$$I = Ia + kd \sum_{j=l}^{nj} (N . Lj) + ks\ S$$

where

ks = the specular reflection coefficient
S = the intensity of light incident from the reflected direction.

The cost of calculating S increase with the complexity of the environment, and ray-traced pictures are usually more costly than Phong shaded ones.

Since this method yields geometrically exact mirror highlights, the reflected environment for an object can be arbitrarily close to the object without introducing errors encountered with the illumination map method. Ray tracing is accurate for most kinds of scenes, and the results often look photographic. It does not deal, however with diffuse reflection of indirect light sources.

2.3 Mirror Highlights by Table Look-up

Blinn [2,5] simulates the mirror highlights on an object by using polar angles of the reflection direction as indices to a panoramic map of the environment. The intensity of reflected light at a point is given by a formula similar to the one used for ray tracing.

This map represents the light intensity for all indirect and direct area light sources as seen from a single point. For this reason, the results are approximate; features of the environment that are very close to the object are not correctly distorted at all points in the object. In addition, the object can not reflect parts of itself. However, for curved surfaces and restricted motion, these errors are not usually noticed, and the effect is realistic. The method is relatively fast but memory intensive.

Max [18] successfully applied a variation of this method that allows for self-reflection.

2.4 Video Lookup of Reflected Light Values

Blinn [6] and Sloan [23] present a method for real-time shading by using video lookup tables. Phong model intensity is computed for 256 different normal directions and the results are stored in the video lookup table. An image of the object is stored in a frame buffer as encoded normals, and the shaded picture is displayed on a video monitor. Lighting and reflectivity parameters can be changed quickly by editing the lookup table, however the resulting images are of low color resolution.

3. SHADING BY REFLECTION MAPS

The following algorithm illustrates shading from reflection tables. It creates a motion picture sequence of a simulated object in a real or simulated environment. For simplicity, the object is of a single surface type and the object is assumed not to move far through the environment, but may rotate about itself. The camera is free to move.

```
(1)   establish the environment illumination table I.
(2)   compute diffuse reflection table D.
(3)   compute specular reflection table S.
(4)   FOR every frame
(5)     FOR every pixel that is of the specified object:
(6)         determine unit vector N normal to surface.
(7)         determine unit vector E from surface to eye.
(8)         determine reflected direction R = 2(E.N)N - E
(9)         output intensity =
                  Wd(E.N) D[polar N] + Ws(E.N) S[polar R]
```

In this algorithm, lines 1 to 3 initialize the tables, and lines 4 to 9 loop over animation frames.

At line 1, a panoramic image of the environment is created (see section 5). The viewpoint should be chosen so that the lighting at that location is representative of the object, e.g., the object's center. The image is transformed and stored in illumination table I which is indexed by the polar coordinates of direction (see section 4).

In lines 2 and 3 table I is convolved to produce tables D and S which are indexed by encoded directions. D, the diffuse reflection map, is the convolution or blurring of the illumination table with a diffuse reflection function. S, the specular reflection map, is the convolution of the illumination table with a specular reflection function (see section 6).

Lines 6 to 9 are executed for every pixel that represents the object of interest. Lines 6 to 8 are normally provided by conventional Phong shading.

In statement 9, vectors N and R are converted to polar coordinates which index tables D and S respectively. (Square brackets [] are used here to signify indexing.) Reflected light values are obtained by table lookup, with optional bi-linear interpolation and antialiasing. Functions Wd and Ws are used to scale the diffuse and specular reflectance as a function of viewing angle. (See Sections 7 and 8.)

4. DATA REPRESENTATION AND SPHERICAL MAPPINGS

This section considers several ways of representing environmental and reflected light.

The environmental light viewed from a point is characterized as a mapping of all view directions that are points on the unit sphere into color triplets that represent the red, green and blue light intensity from each direction. The diffuse component of reflected light is assumed to be a function of surface normals, and is a mapping of the unit sphere (normal directions) into color triplets that represent reflected intensity. Likewise, for the specular component of reflected light and the mirror reflection direction.

These mappings can then be stored as tables of color triplets that are indexed by discrete polar coordinates, each representing a small area on the sphere.

For simplicity and accuracy in interpolation and integration, it is desirable to have adjacent points on the sphere represented by adjacent indices: i.e. to have a continuous transformation. The higher derivatives should also be continuous. It is also desirable, though not possible, to have all indices represent equal areas on the sphere. This is the classic cartographic Problem: mapping a sphere onto a plane [10].

Described below are three kinds of spherical projection. Data can be transformed from one projection to another by image mapping [2,9,13].

4.1 Perspective Projection onto Cube Faces

This projection is used in the preparation of illumination maps of simulated environments and is the usual picture output of most scene-rendering systems. Place the eye at the center of the sphere and project the sphere onto the six faces of the unit cube. The front face is specified by z>|x|, z>|y|, and is mapped by

$$u = y/Z$$
$$v = x/Z$$

The mapping is similar for the other five faces.

For the front face, the area of the sphere represented by each sample point is proportional to 1/z**2 = 1 + u**2 + v**2. Thus, the resolution at the center of each face is effectively one-third the resolution at the corners.

4.2 Orthographic Projection

Each hemisphere is projected onto the xy plane. In this mapping,

u = x
v = y

and there are two map arrays: one for the z >= 0 hemisphere, one for the z < 0 hemisphere.

This is how the normal component of mirror reflection is projected onto a photographic image of a mirrored sphere (see section 5.1).

The area on the sphere represented by each sample point is proportional to sqrt(1 - u**2 - v**2) = 1/z. Thus, the vicinity of the equator is severely under-sampled in the radial direction.

4.3 Polar Coordinates

Illumination data obtained in the two previous projections is transformed into polar coordinates for Illumination maps. This simplifies the interpolation and integration of light values. The north pole is identified with the y direction.

u = longitude arctan(x,z)
v = latitude arccos y

The area on the sphere represented by each sample point is proportional to sin(v) = sqrt(l - z**2). Thus, the vicinity of the poles is over-sampled in the u direction.

5. CREATING THE ILLUMINATION MAP

The first step in our method is the creation of the illumination map. The illumination map may be of real, simulated, or painted environments.

About 24 bits of precision are required for the realistic simulation of natural scenes. This is because the ratio of the intensity of the sun to that of the darkest observable point is above 1,000,000:1 on a sunny day. Although this dynamic range is not re reducible for film and video output, is important that it be available for accurate blurring and antialiasing.

5.1 Real Environments

When a simulated object is to be matted into a real environment, a representative location is selected and a conventional photograph made from that place. Two schemes are possible

1. Photograph the environment in six directions using a flat field lens with a 90-degree field of view to simulate projection onto a cube. The six pictures must be mosaiced into one seamless image.

2. Photograph the environment reflected in a mirrored sphere. This is relatively simple way to capture the whole environment. A silvered glass Christmas tree ball is nearly spherical and a good reflector. A moderately long focal length lens should be used to minimize the effects of perspective.

There may be small perturbations on the globe, and there can be much distortion near the edges. There is a blind spot, i.e. the areas directly behind the globe, which are not reflected in it. For these reasons, mirrored reflections should be confined to curved surfaces where these distortions would be less noticeable than for flat mirrored surfaces.

The photograph must then be digitized, registered, and color-corrected. If the matting is done optically, the color output of the film recorder should be consistent with the color of the film it is matted with. Color correction based on Newton iteration is described by Pratt [22].

The dynamic range of a real environment can greatly exceed the dynamic range of film and the film will saturate for high intensity regions (e.g. the sun). It is critical that high intensity regions are accurately recorded for computing the diffuse component; for the specular component, the accuracy of the Low intensity regions is similarly crucial. Bracketing is the solution: i.e. photograph several exposures for each scene, varying the f-stop of the camera. The different exposures will then need to be registered and digitally combined into a single image.

5.2 Simulated Environments

Area light sources should be modeled as bright self-luminous patches. E.g. a fluorescent lamp would be a very bright cylinder. All, reflecting objects will serve as indirect light sources. Render the scene 6 times from the point of view of the object that will be mapped, placing 6 viewing planes on the faces of a cube centered on the view point (see section 4.1).

The illumination table can also incorporate point light sources. For each point source, determine its direction and add the energy divided by distance squared into the illumination table.

5.3 Painted Environments

Environments may be created by the digitization of conventional paintings, or by digital painting.

6. CONVOLUTION - BUILDING THE REFLECTED LIGHT MAPS

For modeling each type of reflectance property, we postulate two components of reflection: diffuse and specular. Each component is represented by a reflection map or table. The diffuse map is indexed during shading by the direction of the surface normal, and the specular is indexed by the reflected direction. For mirror highlights, the specular reflection map is identical to the illumination map.

6.1 Diffuse Reflection Map

Map D contains the diffuse reflection component for each sample normal direction N. Map D is the convolution of the illumination map with a reflectance function fd.

$$D[N] = (\sum_{L} I[L] \times Area[L] \times fd(N \cdot L))/(4\ pi)$$

where

L ranges over all sample directions indexing the illumination map
I is the average light energy in direction L
Area is the angular area re-presented by direction L
N . L is the cosine between N and L
Fd is the diffuse convolution function

Some examples of the diffuse convolution function are:

1. Lambert reflection, k is the diffuse reflection constant:

$$fd(x) = \begin{cases} kd * x & \text{for } x > 0 \\ 0 & \text{for } x <= 0 \end{cases}$$

2. Self-luminous material:

$$fd(x) = kd \qquad \text{for all } x.$$

Tables of size 36x72 pixels are adequate to store the Lambert reflection map. This is equivalent to a Gouraud shaded sphere with facets at every five degrees.

6.2 Specular Reflection Map

Map S contains the specular reflection component for each sample reflected direction R. Map S is the convolution of the illumination map with a reflectance function fs.

$$S[R] = (\sum_{L} I[L] \text{ x } Area[L] \text{ x } fs(R \,.\, L))/(4 \text{ pi})$$

where

fs = the specular convolution function of R . L

Examples of the specular convolution function are:

1. Perfect mirror:

$$fs(x) = \begin{cases} 1 & \text{for } x = 1 \\ 0 & \text{for } x < 1 \end{cases}$$

2. Conventional Phong specular model,
n is the glossines parameter,
and ks is the specular reflection constant:

$$fs(x) = \begin{cases} ks * x{*}{*}n & \text{for } x > 0 \\ 0 & \text{for } x <= 0 \end{cases}$$

3. Conventional Phong specular with a clear varnish coat:

$$fs(x) = \begin{cases} ks * x{*}{*}n + .5 & \text{for } x = 1 \\ ks * x{*}{*}n & \text{for } 0 < x < 1 \\ 0 & \text{for } x <= 1 \end{cases}$$

7. INTENSITY CALCULATION

For every pixel in the scene occupied by the object, the rendering algorithm should provide:

1. Material type
2. Surface normal N
3. Direction E from the point to the eye

The intensity at that point is then given by line 9 of the algorithm in section 3. This expression has no separate ambient light term since the environment itself is the ambience. The Wd and Ws weighting functions are used to scale the reflected light as a function of viewing angle because many surfaces are more specular at low viewing angles [3,11].

7.1 Interpolation between Tabulated Values

Line 9 quantizes direction. For surfaces of low curvature this can be quite noticeable, especially if the table is of low resolution. This can be alleviated with bilinear interpolation [13] of the table values.

8. ANTIALIASING

If a surface has high curvature, then adjacent pixels in the images will index non-adjacent pixels in the reflection maps. For glossy surfaces, small bright light sources (e.g. the sun) will get lost between the pixels, leading to highlights that break up and scintillate.

High curvature occurs with simulated wrinkles [4], at the edges of polyhedra, and when objects move far away from the eye. A pixel of a highly curved glossy surface can reflect a sizeable portion of the environment and in a number of different ways depending on its curvature [23]. For these reasons, antialiasing for reflectance maps is more critical than it is for ordinary mapped images.

8.1 Recursive Subdivision

This method is inspired by Whitted [25] and is effectively antialiasing with a box filter.

1. Compute the reflected direction at each corner of the pixel.
2. If they index into the same or adjacent table entries, then use the average of the four intensities.
3. If not, subdivide horizontally or vertically.
4. Linearly interpolate the reflected direction and renormalize. Repeat until condition 2 is met or until the regions are inconsequentially small.

8.2 Integrate the Area Spanned by Reflected Corners of Pixel

Here we integrate the area within the quadrilateral defined by the reflected corners of the pixel. We divide by the area to obtain an average value. This method is faster but less accurate than recursive subdivision. Errors however, will average out over neighboring pixels: i.e. highlights may be shifted a fraction of a pixel.

This and the previous method can be extremely costly for very curved shiny surfaces. This is because each pixel maps into many pixels in the reflection map.

8.3 Pre-Integrating the Reflection Map

A significant speedup can be obtained by pre-integrating the reflection map, whereby the number of computations per pixel is constant. This was suggested to us by a fast filtering technique devised by K. Perlin [21]. The quadrilateral spanned by a pixel is approximated by a rectangle in the integrated reflection table. Values are looked-up at the four corners and the integrated value within the rectangle is obtained by differencing. The pre-integrated table must have enough precision to accurately store the highest value times the number of pixels in the map.

When the quadrilateral is not well approximated by the rectangle, then a blurring of the highlight may occur. This blurring is less than the blur produced by pre-filtering the reflection map [12,26].

9. GENERALIZING THE REFLECTION TABLE

The Bidirectional Reflectance-Distribution Fuction [11,14] specifies non-isotropic reflectance as a function of four dimensions and isotropic reflectance can be described a by function of three dimensions. Convolution with the 2-dimensional illumination map results in a 3-dimensional reflectance map that completely describes isotropically reflected light as a function of eye point and surface normal. The reflection map algorithm described in section 1.4 is deficient since it gives intensity as a weighted sum of two 2-dimensional reflection maps. Additional reflection tables may prove valuable in extending the class of materials that can be accurately simulated.

For example, certain materials reflect more light back in the viewing direction: e.g. lunar dust, reflective signs, cat's eyes. Use the viewing direction to index a specular reflection table to simulate these materials.

Another example is that the refraction of light at a single surface. This is easily simulated by computing the refraction direction as a function of surface normal and direction to the eye. Use this direction to index a mirror reflection table.

10. CONCLUSIONS

Vast amounts of computer resources are used to accurately simulate light's interaction with matter. Illumination and reflection maps provide an efficient means for realistically, simulating many types of reflections. The illumination map works because it reduces a 3-D data structure (luminance at all points in space) to a 2-D one (luminance as seen from a point). This simplification reduces the calculation time with a small sacrifice in accuracy.

Other interactions -- scattering, translucency, and diffuse shadows -- are now prohibitively expensive. Perhaps the illumination map can be applied to these areas also.

ACKNOWLEDGEMENTS

We would like to thank the production staffs and management of MAGI SynthaVision and Digital Effects for providing the motivation and, support for this work. Thanks to Ken Perlin and Joshua Pines for creating the advanced SynthaVision IMAGE program which now contains "chrome". Thanks to James Blinn and the 1983 reviewers who provided helpful comments.

REFERENCES

1. Bass, Daniel H. Using the Video Lookup Table for Reflectivity Calculations: Specific Techniques and Graphics Results. Computer Graphics and Image Processing, V.17 (1981), pp. 249-261.

2. Blinn, James F. and Newell, Martin E. Texture and Reflection in Computer Generated Images. Communications of the ACM, V.19 #10 (1976), pp. 542-547.

3. Blinn, James F. Models of Light Reflection for Computer Synthesized Pictures. SIGGRAPH 1977 Proceedings, Computer Graphics, V.11 #2 (1977), pp. 192-198.

4. Blinn, James F. Simulation of Wrinkled Surfaces. SIGGRAPH 1978 Proceedings, Computer Graphics, V.12 #3 (1978), pp. 286-292.

5. Blinn, James F. Computer Display of Curved Surfaces. PhD dissertation, University of Utah, Salt Lake City (1978).

6. Blinn, James F. Raster Graphics. Tutorial: Computer Graphics, KellOg Booth, ed., (1979), pp. 150-156. (IEEE Cat. No. EHO 147-9)

7. Blinn, James F. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. SIGGRAPH 1982 Proceedings, Computer Graphics, V.16 #3 (1982), pp. 21-29.

8. Bui-Tuong Phong. Illumination for computer generated pictures. Communications of the ACM, V.18 #6 (June 1975), pp. 311-317.

9. Catmull, Edwin A. Computer display of curved surfaces. Proc. Conf. on Computer Graphics, Pattern Recognition and Data Structures (May 1975), pp. 11-17. (IEEE Cat. No. 7SCH 0981-IC)

10. Central Intelligence Agency, Cartographic Automatic Mapping Program Documentation - 5th Edition, GC 77-10126 (June 1977).

11. Cook, Robert L. and Torrance, Kenneth E. A reflectance model for computer graphics. SIGGRAPH 1981 Proceedings, Computer Graphics, V.15 #3 (1981), pp. 307-316.

12. Dungan, William. Texture tile considerations for raster graphics. SIGGRAPH 1978 Proceedings, Computer Graphics, V.12 #3 (1978), pp. 130-134.

13. Feibush, Eliot A., Levoy, Marc and Cook, Robert L. Synthetic texturing using digital filters. SIGGRAPH 1980 Proceedings, Computer Graphics, V.14 #3 (July 1980), pp. 294-301.

14. Horn, B. K. P. and Sjoberg, R. W. Calculating the reflectance map. Applied Optics, v-18 #11 (June 1979), pp. 1170-1179.

15. Kay, Douglas S. and Greenberg, Donald. Transparency for computer synthesized images. SIGGRAPH 1979 Proceedings, Computer Graphics V.13 2 (Aug. 1979), pp. 158-164.

16. Minnaert, M. The Nature of Light and Color in the Open Air. Dover Publishing Inc., New York, 1954

17. Max, Nelson and Blunden, John. Optical printing in computer animation. SIGGRAPH 1980 Proceedings, Computer Graphics, V.14 #3 (July 1980), pp. 171-177.

18. Max, N. L. Vectorized procedural models for natural terrains: waves and island in the sunset. SIGGRAPH 1981 Proceedings, Computer Graphics, V.15 #3 (.1981), pp. 317-324.

19. Newell, Martin E. and Blinn, James F. The progression of realism in computer generated images. Proceedings of ACM Annual Conf. (1977), pp. 444-448.

20. Norton, Alan, Rockwood, Alyn and Skolmoski, Philip T. Clamping: a method of antialiasing-textured surfaces by bandwidth limiting in object space. SIGGRAPH 1982 Proceedings, Computer Graphics, V.16 #3 (July 1982), pp. 1-8.

21. Perlin, Kenneth. Unpublished personal communication.

22. Pratt, William K. Digital image processing. John Wiley and Sons, New York, 1978.

23. Sloan, K. R. and Brown, C. M. Color map techniques. Computer Graphics and Image Processing. V.10 (1979), pp. 297-317.

24. Thomas, David E. Mirror images. Scientific American (Dec. 1980), pp. 206-228.

25. Whitted, Turner. An improved illumination model for shaded display. Communications of the ACM, V.23 #6 (June 1980), pp. 343-349.

26. Williams, Lance. Pyramidal Parametrics. SIGGRAPH 1983 Proceedings, Computer Graphics, V.17 #3 (1983), pp. 1-11.