Graphics and Image Processing
W. Newman Editor

Illumination for Computer Generated Pictures

Bui Tuong Phong
University of Utah

The quality of computer generated images of three-dimensional scenes depends on the shading technique used to paint the objects on the cathode-ray tube screen. The shading algorithm itself depends in part on the method for modeling the object, which also determines the hidden surface algorithm. The various methods of object modeling, shading, and hidden surface removal are thus strongly interconnected. Several shading techniques corresponding to different methods of object modeling and the related hidden surface algorithms are presented here. Human visual perception and the fundamental laws of optics are considered in the development of a shading rule that provides better quality and increased realism in generated images.

Key Words and Phrases: computer graphics, graphic display, shading, hidden surface removal.

CR Categories: 3.26, 3.41, 8.2

Introduction

This paper describes several approaches to the production of shaded pictures of solid objects. In the past decade, we have witnessed the development of a number of systems for the rendering of solid objects by computer. The two principal problems encountered in the design of these systems are the elimination of the hidden parts and the shading of the objects. Until now, most effort has been spent in the search for fast hidden surface removal algorithms. With the development of these algorithms, the programs that produce pictures are becoming remarkably fast, and we may now turn to the search for algorithms to enhance the quality of these pictures.

In trying to improve the quality of the synthetic images, we do not expect to be able to display the object exactly as it would appear in reality, with texture, overcast shadows, etc. We hope only to display an image that approximates the real object closely enough to provide a certain degree of realism. This involves some understanding of the fundamental properties of the human visual system. Unlike a photograph of a real world scene, a computer generated shaded picture is made from a numerical model, which is stored in the computer as an objective description. When an image is then generated from this model, the human visual system makes the final subjective analysis. Obtaining a close image correspondence to the eye's subjective interpretation of the real object is then the goal. The computer system can be compared to an artist who paints an object from its description and not from direct observation of the object. But unlike the artist, who can correct the painting if it does not look right to him, the computer that generates the picture does not receive feedback about the quality of the synthetic images, because the human visual system is the final receptor.

This is a subjective domain. We must at the outset define the degree of realism we wish to attain, and fix certain goals to be accomplished. Among these goals are:

1. "Real time" display of dynamic color pictures of three-dimensional objects. A real time display system is one capable of generating pictures at the rate of at least 30 frames a second.
2. Representation of objects made of smooth curved surfaces.
3. Elimination or attenuation of the effects of digital sampling techniques.

The most important consideration in trying to attain these goals is the object modeling technique.

Existing Shading Techniques

Methods of Object Modeling

Image quality depends directly on the effectiveness of the shading algorithm, which in turn depends on the method of modeling the object. Two principal methods of object description are commonly used:

1. Surface definition using mathematical equations.
2. Surface approximation by planar polygonal mosaic.

Several systems have been implemented to remove hidden parts for mathematically defined curved surfaces [1, 2, 3, 4, 5]. With these systems, exact information at each point of the surface can be obtained, and the result-

This research was supported in part by the University of Utah Computer Science Division and the Advanced Research Projects Agency of the U.S. Department of Defense, monitored by the Rome Air Development Center, Griffiss Air Force Base, NY 13440, under Contract F30602-70-C-0300. Author's address: Digital Systems Laboratory, Stanford University, Stanford, CA 94305.

ing computer generated pictures are most realistic. The class of possible surfaces is restricted, however, and the computation time needed to remove the hidden parts and to perform shading is very large. Up to the present time, these systems have usually considered the class of surfaces represented by quadric patches. Although higher degree surfaces are desirable and are sometimes necessary to model an object, they have not been taken into consideration due to an increase in computation time to remove hidden surfaces and to perform shading computations. Even when only quadric surfaces are considered, the implementation of a real time display system using this type of model is too expensive and complex.

A simple method of representing curved surfaces and objects of arbitrary shape is to approximate the surfaces with small planar polygons; for example, a cone might be represented as shown in Figure 1. This type of representation has the advantage that it avoids the problem, posed by mathematically curved surface approaches, of solving higher order equations.

Planar approximation also offers the only means of reducing hidden surface computation to within reasonable bounds, without restricting the class of surfaces that can be represented. For this reason, all recent attempts to devise fast hidden surface algorithms have been based on the use of this approximation for curved surfaces; these algorithms have been summarized and classified by Sutherland et al. [6]. The next section discusses their influence on the way shading is computed.

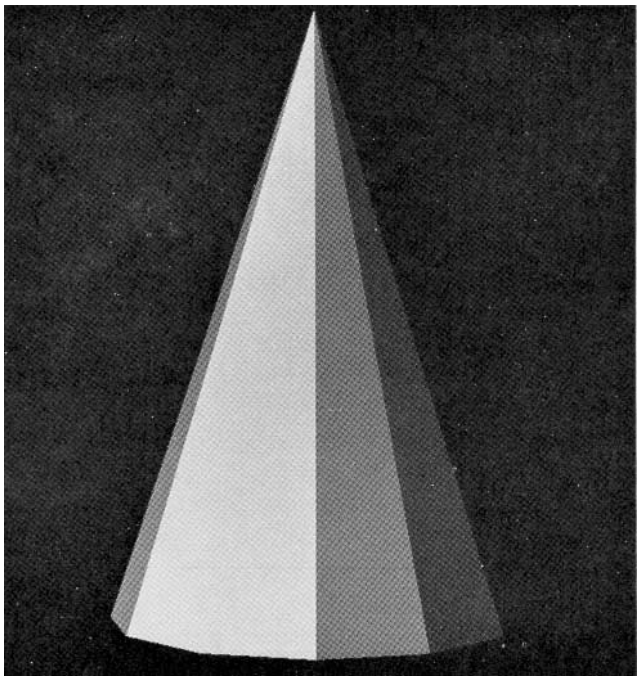
While planar approximation greatly simplifies hidden surface removal, it introduces several major problems in the generation of a realistic displayed image. One of these is the *contour edge* problem: the outline or silhouette of a polygonally approximated object is itself a polygon, not a smooth curve. The other problem is that of shading the polygons in a realistic manner. This paper is concerned with the shading problem; the contour edge problem is discussed by the author and F.C. Crow in [7].

Influence of Hidden Surface Algorithms

The order in which a hidden surface algorithm computes visible information has a decided influence on the way shading is performed. For example Warnock, who developed one of the first such algorithms [8], computed display data by a binary subdivision process: this meant that the order of generating display data was largely independent both of the order of scanning the display and of the order of the polygons in memory. This made it difficult to perform effective shading on curved objects.

The two major advances in the development of fast hidden surface algorithms have been made by Watkins [9] and by Newell, Newell, and Sancha [10]. Watkins generates the displayed picture scan line by scan line. On each scan line he computes which polygons intersect the scan line, and then computes the visible *segment* of each polygon, where this segment is the visible strip of

Fig. 1. A cone represented by means of planar approximation.



the polygon, one screen resolution unit in height, that lies on the scan line.

Newell, Newell, and Sancha adopt a different approach, using a *frame buffer* into which the object is painted, face by face. The hidden surface problem is solved by painting the farthest face first, and the nearest last. Each face is painted scan line by scan line, starting at the top of the face.

From the shading aspect, the important attribute of these algorithms is that they both generate information scan line by scan line in order to display the faces of an object. This information is in the form of segments, one screen resolution unit high, on which the shading computation may then be performed. The main differences between the algorithms, from the point of view of shading, are (a) the order in which the segments are generated, and (b) the fact that Watkins generates each screen dot only once, whereas the Newell-Sancha algorithm may overwrite the same dot several times.

Shading with the Polyhedral Model

When planar polygons are used to model an object, it is customary to shade the object by using the *normal vectors* to the polygons. The shading of each point on a polygon is then the product of a shading coefficient for the polygon and the cosine of the angle between the polygon normal and the direction of incident light. This cosine relationship is known in optics as the "cosine law," and allows us to compute the shading S_p for a polygon p as

$$S_p = C_p \cos(i), \tag{1}$$

where C_p is the reflection coefficient of the material of p relative to the incident wavelength, and i is the incident angle.

Fig. 2. An example of the use of Newell, Newell, and Sancha's shading technique, showing transparency and highlight effects.

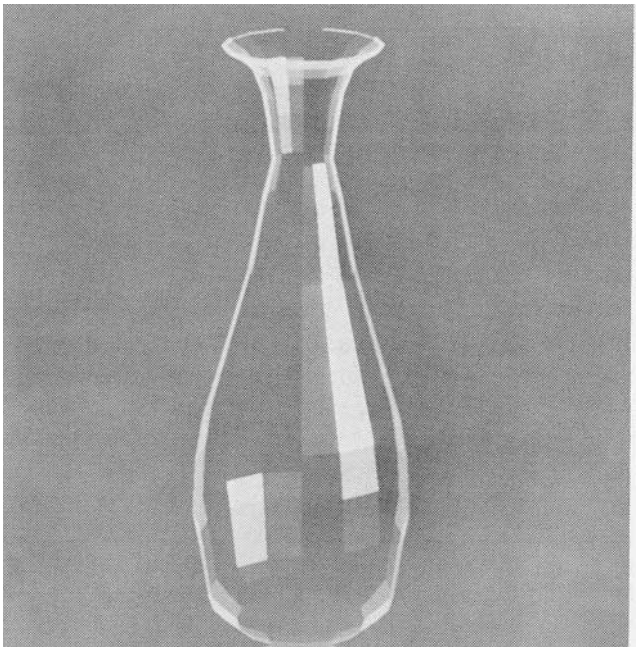


Fig. 3. Computation of the shading at point R using the Gouraud method. There are two successive linear interpolations: (1) across polygon edges, i.e. P between A and B, Q between A and D; and (2) along the scan line, i.e. R between P and Q.

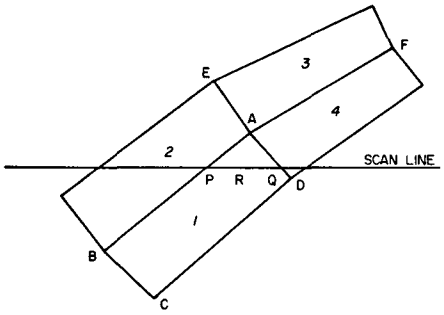
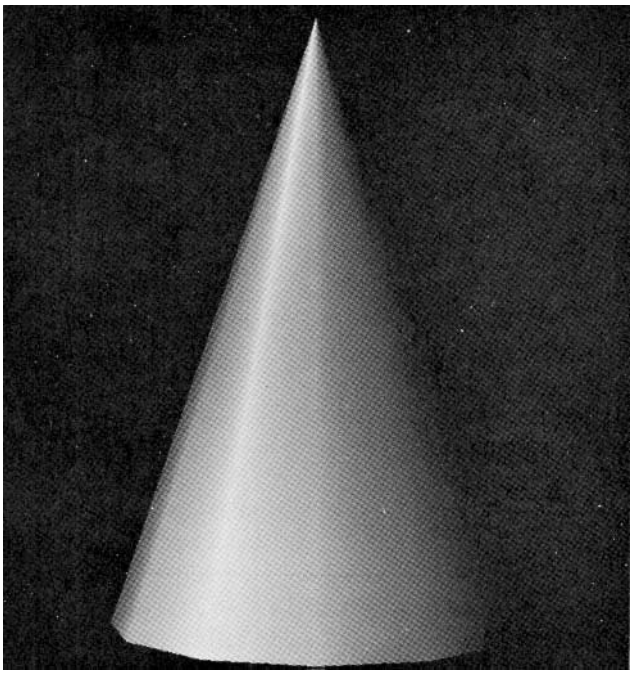


Fig. 4. Gouraud shading, applied to approximated cone of Fig. 1.



This shading offers only a very rough approximation of the true physical effect. It does not allow for any of the *specular* properties of the material, i.e. the ability of the material to generate highlights by reflection from its outer surface, and the position of the observer, which is ignored. A more serious drawback to this method, however, is the poor effect when using it to display smooth curved surfaces. The cosine law rule is appropriate for objects that are properly modeled with planar surfaces, such as boxes, buildings, etc., but it is inappropriate for smoothly curved surfaces such as automobile bodies. This does not mean, however, that we should abandon the use of such a polygon-oriented shading rule and search for a different rule for curved surfaces. Recent research in shading techniques demonstrates that significant results can be achieved by using the basic shading rule of eq. (1) and modifying the results to reduce the discontinuities in shading between adjacent polygons.

1. Warnock's shading. As three-dimensional objects are projected onto the cathode-ray tube screen, the depth sensation is lost, and the images of those objects appear flat. In order to restore the depth sensation, two effects were simulated by Warnock:

1. Decreasing intensity of the reflected light from the object with the distance between the light source and the object.

2. Highlights created by specular reflection.

Warnock placed the light source and the eye at the same position, so that the shading function was the sum of two terms, one for the normal "cosine" law, and the other term for the specularly reflected light. The resulting pictures have several desirable attributes; for example, identical parallel faces, located differently in space, will be shaded at different intensities, and facets which face directly toward the light source are brighter than adjacent facets facing slightly away from the incident light. However, the polygonal model gives a discontinuity in shading between faces of an approximated curved surface. When a curved surface is displayed, the smoothness of the curved surface is destroyed by this discontinuity. This is clearly visible in Figure 1.

2. Newell, Newell, and Sancha's shading. Newell, Newell, and Sancha presented some ideas on creating transparency and highlights. From observations in the real world, they found that highlights are created not only by the incident light source but also by the reflection of light from other objects in the scene; this is especially true in the case of objects made of highly reflective or transparent materials. In the Newell-Sancha model, curved surfaces are approximated with planar polygons. Unfortunately, the ability to generate highlights is severely limited due to the inability to vary light intensity over the surface of any single polygon. This problem is apparent in Figure 2.

3. Gouraud's shading. While working on a technique to represent curved objects made of "Coons surfaces"

or "Bezier patches," Gouraud [11] developed an algorithm to shade curved surfaces. With his algorithm, a surface represented by a patch is approximated by polygonal planar facets. Gouraud computes information about the curvature of the surface at each vertex of each of these facets. From the curvature, a shade intensity is computed and retained. For example, the shade intensity may be computed for each vertex using eq. (1), with i as the angle between the incident light and the normal to the surface at this vertex. When the surface is displayed, this shade intensity is linearly interpolated along the edge between adjacent pairs of vertices of the object. The shade at a point on the surface is also a linear interpolation of the shade along a scan line between intersections of the edges with a plane passing through the scan line (Figure 3). This very simple method gives a continuous gradation of shade over the entire surface, which in most cases restores the smooth appearance. An example of Gouraud's shading is shown in Figure 4.

With the introduction of the Gouraud smooth shading technique, the quality of computer-generated images improved sufficiently to allow representation of a large variety of objects with great realism. Problems still exist, however, one of which is the apparent discontinuity across polygon edges. On surfaces with a high component of specular reflection, highlights are often inappropriately shaped, since they depend upon the disposition and shape of the polygons used to approximate a curved surface and not upon the curvature of the object surface itself. The shading of a surface in motion (in a computer generated film) has annoying frame to frame discontinuities due to the changing orientation of the polygons describing the surface. Also the shading algorithms are not invariant under rotation.

Frame-to-frame discontinuities of shade in a computer generated film are illustrated in the following situation. A curved surface is approximated with planar facets. When this surface is in motion, all the facets which are perpendicular to the direction of the light take on a uniform shade. In the next frame the motion of the object brings these facets into a different orientation toward the light, and the intensity of the shade across their surfaces varies continuously from one end to the other. Thus the surface appears to change from one with highlights to one of uniform shade. Moreover, the position of these highlights is not steady from frame to frame as the object rotates.

Mach Band Effect

Many of the shading problems associated with planar approximation of curved surfaces are the result of the discontinuities at polygon boundaries. One might expect that these problems could be avoided by reducing the size of the polygons. This would be undesirable, of course, since it would increase the number of polygons and hence would increase both the memory requirements for storing the model and the time for hidden surface removal.

Fig. 5. Normal at a point along an edge.

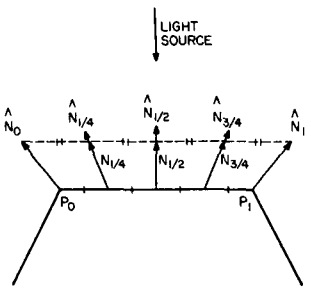
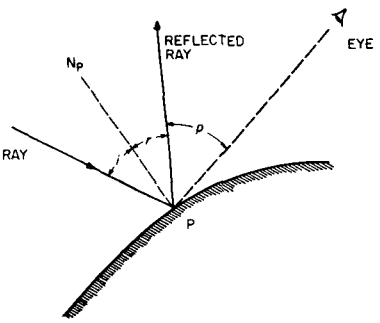


Fig. 6. Shading at a point.



Unfortunately, because of visual perception effects, the reduction of polygon size is not as beneficial as might be expected. The particular effect responsible is the *Mach Band* effect. Mach established the following principle:

> Wherever the light-intensity curve of an illuminated surface (the light intensity of which varies in only one direction) has a concave or convex flection with respect to the axis of the abscissa, that particular place appears brighter or darker, respectively, than its surroundings [E. Mach, 1865].

Whenever the slope of the light intensity curve changes, this effect appears. The extent to which it is noticeable depends upon the magnitude of the curvature change, but the effect itself is always present.

Without the Mach Band effect, one might hope to achieve accurate shading by reducing the size of polygons. Unfortunately the eye enhances the discontinuities over polygon edges, creating undesired areas of apparent brightness along the edges. Therefore unless the size of the displayed facets is shrunk to a resolution point, increasing the number of facets does not solve the problem. Using the Gouraud method to interpolate the shade linearly between vertices, the discontinuities of the shading function disappear, but the Mach Band effect is visible where the slope of the shading function changes. This can be seen in Figure 4. The subjective discontinuity of shade at the edges due to the Mach Band effect then destroys the smooth appearance of the curved surface.

A better shading rule is therefore proposed for displaying curved surfaces described by planar polygons. This new technique requires the computation of the normal to the displayed surface at each point. It is therefore more expensive in computation than Gouraud's technique; but the quality of the resulting picture, and the accuracy of the displayed highlights, is much improved.

Using a Physical Model

Specular Reflection

If the goal in shading a computer-synthesized image is to simulate a real physical object, then the shading model should in some way imitate real physical shading situations. Clearly the model of eq. (1) does not accomplish this. As mentioned before, it completely

Fig. 7(a). Determination of the reflected light.

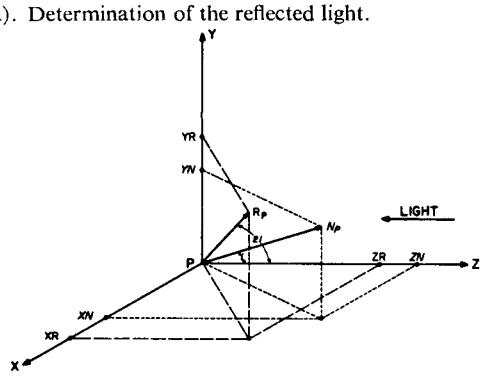
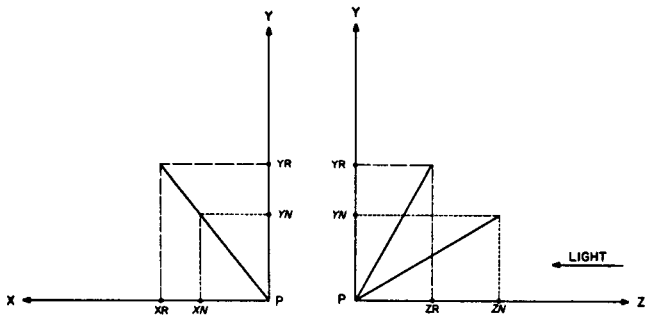


Fig. 7(b). Projections of the reflected light.



ignores both the position of the observer and the specular properties of the object. Even with the improvements introduced by Gouraud, which provide remarkably better shading, these properties are still ignored.

The first step in accounting for the specular properties of objects and the position of the observer is to determine the normal to the surface at each point to be shaded, i.e. at each point where a picture element of the raster display projects onto the surface. It is only with this knowledge that information about the direction of reflected rays can be acquired, and only with this information can we model the specular properties of objects. It is evident from the preceding discussion, however, that our polyhedral model provides information about normals only at the vertices of polygons. Thus the first step in improving our shading model is to devise a way to obtain the normal to the surface for each raster unit.

Computation of the Normal at a Point on the Surface

The normal at each vertex can be approximated by either one of the methods described by Gouraud [10]. It is now necessary to define the normal to the surface along the edges and at a point on the surface of a polygon.

The normal to the surface at a point along the edge of a polygonal model is the result of a linear interpolation to the normals at the two vertices of that edge. An example is given in Figure 5: the normal N_t to the surface at a point between the two vertices P_0 and P_1 is computed as follows:

$$N_t = tN_1 + (1-t)N_0, \tag{2}$$

where $t = 0$ at N_0 and $t = 1$ at N_1.

The determination of the normal at a point on the surface of a polygon is achieved in the same way as the computation of the shading at that point with the Gouraud technique. The normal to the visible surface at a point located between two edges is the linear interpolation of the normals at the intersections of these two edges with a scan plane passing through the point under consideration. Note that the general surface normal is quadratically related to the vertex normal.

From the approximated normal at a point, a shading function determines the shading value at that point.

The Shading Function Model

In computer graphics, a shading function is defined as a function which yields the intensity value of each point on the body of an object from the characteristics of the light source, the object, and the position of the observer.

Taking into consideration that the light received by the eye is provided one part by the diffuse reflection and one part by the specular reflection of the incident light, the shading at point P (Figure 6) on an object can be computed as:

$$S_p = C_p[\cos(i)(1-d)+d] + W(i)[\cos(s)]^n, \tag{3}$$

where:

C_p is the reflection coefficient of the object at point P for a certain wavelength.

i is the incident angle.

d is the environmental diffuse reflection coefficient.

$W(i)$ is a function which gives the ratio of the specular reflected light and the incident light as a function of the incident angle i.

s is the angle between the direction of the reflected light and the line of sight.

n is a power which models the specular reflected light for each material.

The function $W(i)$ and the power n express the specular reflection characteristics of a material. For a highly reflective material, the values of both $W(i)$ and n are large. The range of $W(i)$ is between 10 and 80 percent, and n varies from 1 to 10. These numbers are empirically adjusted for the picture, and no physical justifications are made. In order to simplify the model, and thereby the computation of the terms $\cos(i)$ and $\cos(s)$ of formula (3), it is assumed that:

1. The light source is located at infinity; that is, the light rays are parallel.
2. The eye is also removed to infinity.

With these two considerations, the values of $\cos(i)$ and $\cos(s)$ of the shading function in (3) can be rewritten as: $\cos(i) = kN_p / |N_p|$ and $\cos(s) = uR_p / |R_p|$ where k and u are respectively the unit vectors in the direction of the light and the line of sight, N_p is the normal vector at P, and R_p is the reflected light vector at P.

The quantity $kN_p / |N_p|$ can be referred to as the projection of a normalized vector N_p on an axis parallel to the direction of the light. If $|N_p|$ is unity, the previous

Fig. 8. Improved shading, applied to approximated cone of Fig. 1.

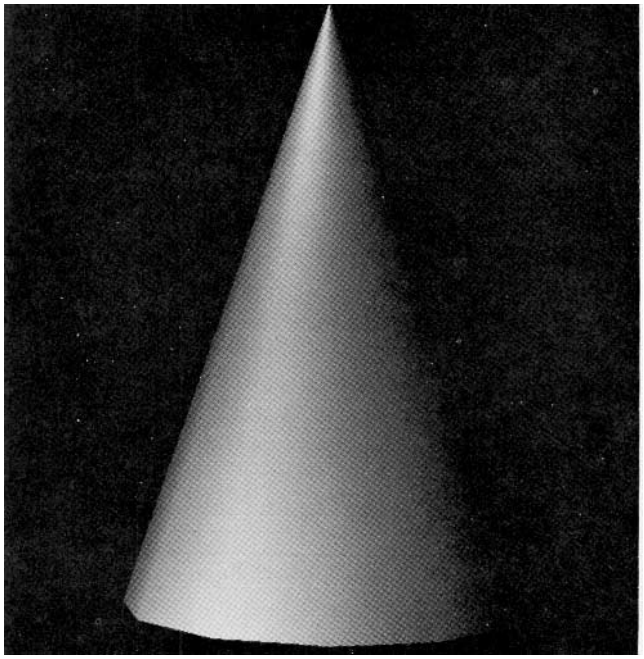
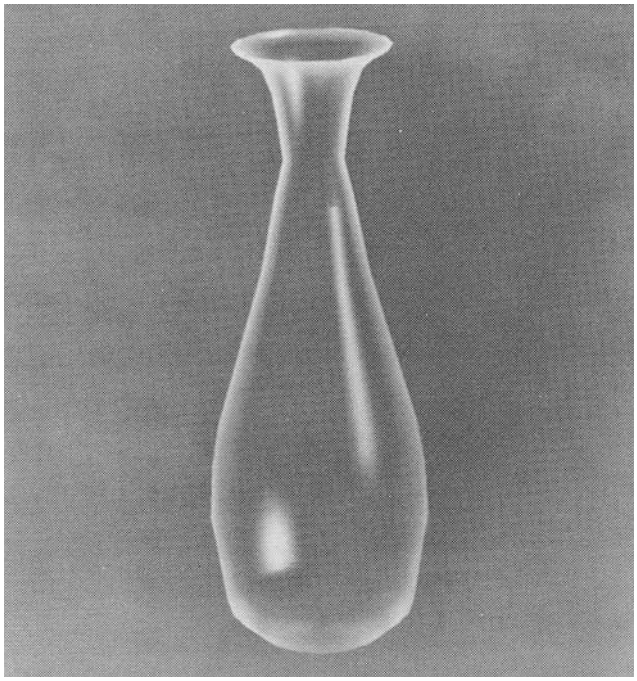


Fig. 9. Improved shading, applied to the example of Figure 2.



quantity is one component of the vector N_p in a coordinate system where the direction of light is parallel to one axis. In this case, the quantity $uR_p / |R_p|$ can be obtained directly from the vector N_p in the following way.

Let us consider a Cartesian coordinate system having the origin located at point P and having the z axis parallel to the light but opposite in direction (Fig. 7(a)).

We have the following assumptions about the model:

1. The normalized vector N_p makes an angle i with the z axis, and the reflected light vector R_p makes an angle $2i$ with the same axis.
2. Only incident angles less than or equal to 90 degrees are considered in the shading computation. For a greater angle, this means that the light source is behind the front surface. In the case where a view of the back surface is desired when it is visible, it can be assumed that the normal will always point toward the light source.
3. If k is the unit vector along the PZ axis, then by simple geometry, it may be shown that the three vectors k, N_p, and R_p are coplanar.
4. The two vectors N_p and R_p are of unit length.

From assumption (3), the projections of the vectors N_p and R_p onto the plane defined by (PX,PY) are merged into a line segment (Figure 7(b)). Therefore,

$$X_r/Y_r = X_n/Y_n, \tag{4}$$

where X_r, X_n, Y_r, and Y_n are respectively the components of R_p and N_p in the x and y directions.

From assumptions (1) and (2), the component Z_n of N_p is:

$$Z_n = \cos(i), \tag{5}$$

where $0 \leq i \leq 90$ degrees.

By simple trigonometry, we obtain the following expressions:

$$Z_r = \cos(2i) = 2[\cos(i)]^2 - 1 = 2Z_n^2 - 1, \tag{6}$$

$$X_r^2 + Y_r^2 = [\sin(2i)]^2 = 1 - [\cos(2i)]^2. \tag{7}$$

From (4) and (7), we obtain:

$$X_r = 2Z_nX_n, \quad Y_r = 2Z_nY_n, \quad 0 \leq Z_n \leq 1.$$

The three components of R_p are then known in the light source coordinate system. The projection of the vector R_p onto the z-axis of the eye coordinate system may be found by a simple dot product of the reflected vector with this z-axis. The component of R_p on an axis parallel to the line of sight is the value of the cosine of the angle between the reflected light and the line of sight. The value of this cosine will be used in the simulation of the specular reflection.

This method of calculating the direction of the reflected light for each point from the orientation of the normal is preferred over the computation of the reflected light vector at vertices and the subsequent interpolation of them in the same way as the normal. It is faster and it requires less storage space than the interpolation scheme.

With the described method, the shading of a point is computed from the orientation of the approximated normal; it is not a linear interpolation of the shading values at the vertices. Therefore, a better approximation of the curvature of the surface is obtained, and highlights due to the simulation of specular reflection are properly rendered. Examples of application of the shading technique are shown in Figures 8 and 9. Figure 10 compares a display generated by this technique with a photograph of a real object.

Fig. 10(a). A sphere displayed with the improved shading.

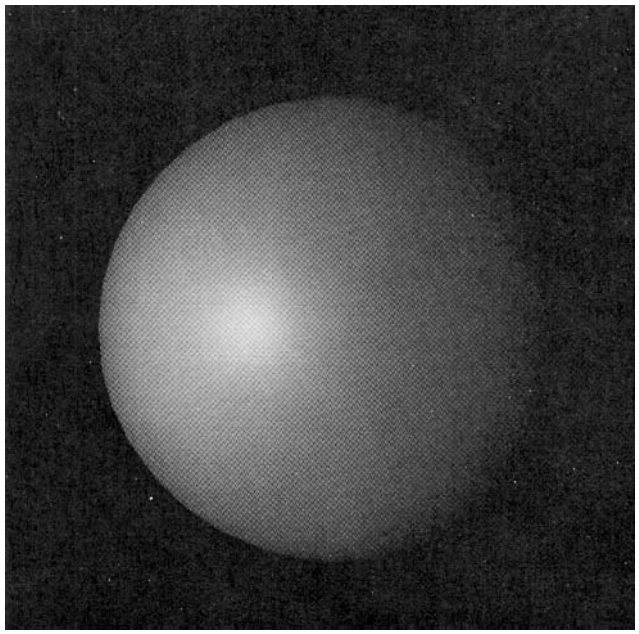
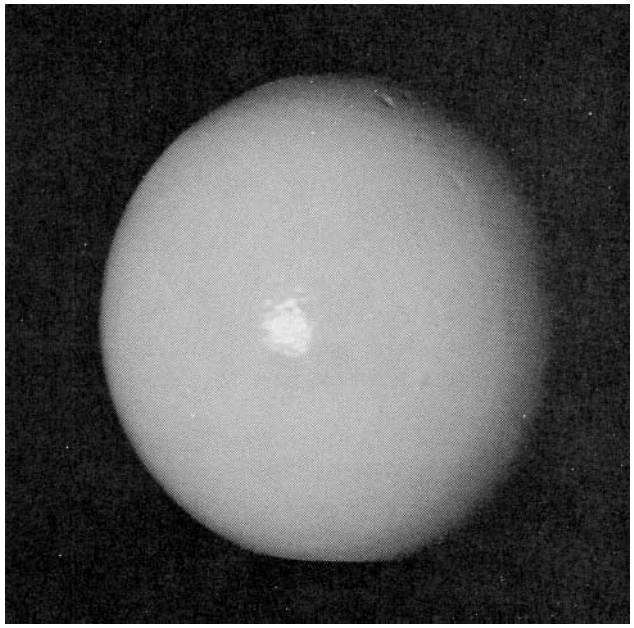


Fig. 10(b). A photograph of a real sphere.



Conclusion

The linear interpolation scheme used here to approximate the orientation of the normal does not guarantee a continuous first derivative of the shading function across an edge of a polygonal model. In extreme cases where there is an abrupt change in the orientation of two adjacent polygons along a common edge, the subjective brightness due to the Mach Band effect will be visible along this edge. However, this effect is much less visible in the described model than in the Gouraud smooth shading model. Also, an interesting fact discussed previously on Mach Band effect shows that this effect is visible whenever there is a great change in the slope of the intensity distribution curve, even if the curve has a continuous first derivative. When a higher degree interpolation curve is used, it will make the presence of the edges unnoticeable, although it will still give some Mach Band effect.

When a comparison was made of pictures of the same object generated with different shading techniques, it was found that little difference existed between pictures generated with the new shading and the ones created with a cubic interpolant curve for the shading computation. Furthermore, as time is the critical factor in a real time dynamic picture display system, the use of a high degree interpolation curve does not seem to be possible at the moment with the current techniques to compute the coefficients of such a function.

A hardware implementation of this shading model would of course require more hardware than the simpler Gouraud method. The Gouraud model needs one interpolator for the shading function. It must compute a new shading value for each raster unit, and hence must be very high speed to drive a real time display. The model proposed here requires three of these interpolators operating in parallel. In addition, since the results of the interpolation do not yield a unit vector, and since eqs. (6), (7), and (8) require a unit normal vector, some extra hardware is necessary to "normalize" the outputs of the interpolators. This requires a very fast mechanism for obtaining square roots. None of these problems is too difficult to solve; and judging from the improvements in image quality obtained using the new model, it may well be worth the extra expense to provide such hardware in applications for which real time display is important.

Received November 1975; revised March 1975

References

1. MAGI, Mathematical Applications Group Inc. 3-D simulated graphics. *Datamation 14* (Feb. 1968), 69.
2. Comba, P.G. A procedure of detecting intersections of three-dimensional objects. Rep. 39,020, IBM New York Scientific Center, Jan. 1967.
3. Weiss, R.A. BE VISION, a package of IBM 7090 FORTRAN programs to draw orthographic views of combinations of plane and quadric surfaces. *J. ACM 13*, 2 (Apr. 1966), 194–204.
4. Mahl, R. Visible surface algorithm for quadric patches. *IEEE Trans. C-21*, (Jan. 1972), 1–4.
5. Catmull, E.E. A subdivision algorithm for computer display of curved surfaces. Ph.D th., Dep. of Comput. Sci., U. of Utah.
6. Sutherland, I.E., Sproull, R.F., and Schumacker, R.A. A characterization of ten-hidden surface algorithms. *Computing Surveys 6* (Mar. 1974), 1–56.
7. Bui Tuong Phong and Crow, F.C. Improved rendition of polygonal models of curved surfaces. To be presented at the joint USA-Japan Computer Conference.
8. Warnock, J.E. A hidden-line algorithm for halftone picture representation. Dep. of Comput. Sci., U. of Utah, TR 4–15, 1969.
9. Watkins, G.S. A real-time visible surface algorithm. Dep. of Comput. Sci., U. of Utah, UTEC-CSc-70-101, June 1970.
10. Newell, M.E., Newell, R.G., and Sancha, T.L. A new approach to the shaded picture problem. Proc. ACM 1973 Nat. Conf.
11. Gouraud, H. Computer display of curved surfaces. Dep. of Comput. Sci., U. of Utah, UTEC-CSc-71-113, June 1971. Also in *IEEE Trans. C-20* (June 1971), 623–629.