Continuous Shading of Curved Surfaces

HENRI GOURAUD

Abstract – A procedure for computing shaded pictures of curved surfaces is presented. The surface is approximated by small polygons in order to solve easily the hidden-parts problem, but the shading of each polygon is computed so that discontinuities of shade are eliminated across the surface and a smooth appearance is obtained. In order to achieve speed efficiency, the technique developed by Watkins is used which makes possible a hardware implementation of this algorithm.

Index terms – Coons patches, curved surfaces, halftone, hidden-line removal, shading.

INTRODUCTION

Since computers have been used to produce perspectives of three-dimensional objects, one of the main problems has been the tradeoff between the speed at which a picture could be produced and the realism of this picture. On one hand, cathode-ray tubes are able to display line drawings very efficiently; on the other hand, images with hidden parts removed and with shading take a long time to compute. In 1963 Roberts [1] developed the first program capable of removing hidden lines. Since then other algorithms performing the same task have been developed by Galimberty [2], Kubert [3], and Loutrel [4], among others. Their algorithms solve the hidden-line problem for structures composed of planar polygons. Two algorithms developed by Comba [5] and Weiss [6] remove hidden lines for objects made of quadric surfaces. In 1967 shaded images were introduced by the University of Utah (Romney [7], Warnock [8], Watkins [9]), General Electric (Rougelot [10]), MAGI [11], and IBM (Appel [12]). More recently, Bouknight and Kelley [17], [18] presented an algorithm producing shaded pictures with shadows and movable light sources. General Electric built for NASA the first hardware capable of generating real-time shaded pictures. Combining the work of both Warnock and Romney, Watkins recently developed a fast algorithm which will shortly be implemented in hardware at the University of Utah.

Realism beyond the obvious hidden-surface removal is obtained by shading each object in black and white or in color. In the General Electric system a fixed color is assigned by hardware to each of the different polygons composing the scene. The potential for changing this color from frame to frame exists, but the author is not aware of its use. This scheme gives a "cartoon-like" appearance to the generated images. Appel developed a system to produce shaded images on a digital plotter. The shading of a particular polygon is computed only as a function of the orientation of this polygon. This could become confusing in the case of parallel polygons, but is avoided in this particular case since the visible edges are drawn on top of the shading. Warnock and Romney were the first to use a shading rule in which both the orientation of the object and its distance from the observer are taken into consideration. Warnock uses the rule

$$S = \left| \frac{\cos \theta}{R} \right|$$

Romney uses the rule

$$S = \frac{\cos^2 \theta}{R\,4} * \text{(normalization factor)}$$

Where $\cos \theta$ is a measure of the orientation of the polygon and R a measure of its distance from the observer. In both cases, the light source was located at the observer's position to avoid any need to show shadows. In the case of color pictures, Warnock also introduced the notion of specular reflectance as a term of the form

$$\frac{\cos^m \theta}{R} \quad 6 \le m \le 10$$

added to each of the three basic color components.

The essence of shaded pictures is to generate a different shade of gray for each resolution point on the projection screen, and each of the programs mentioned above has tried to reduce the time spent in computing a new shading for each point. The requirement that the objects be composed of planar polygons was mainly made to facilitate the hidden-parts computations, but it also permitted simplicity in the computation of the shading for each polygon because a part of this computation is done in common for all the points of this polygon. In the General Electric system the shading is the same on the entire projection of a given polygon. Warnock, Romney, and Watkins compute the shading at some particular points of the polygon and use linear interpolation to compute the shading at other points.

As an example, let us examine in more detail how the computation of the shading is performed in Watkins' algorithm. During a quick preprocessing of the description of the object, the orientation of each polygon is computed and stored in the data structure. The final image is then computed scan line by scan line. For each scan line the hidden parts are first eliminated and then each visible portion of a polygon is shaded according to its orientation and distance.

The distance has been introduced in the shading rule in order to make a distinction in shade between two overlapping separated parallel planes. Our experience has shown that the method used to compute this distance is not critical as long as the relative ordering of the objects is preserved.

The perspective transformation has this property, and the perspective coordinates which have already been used to solve the hidden-parts problem can be used again to compute the shading. If XYZ are the coordinates of a point, its projection has the coordinates

$$\frac{X}{Z} \quad \frac{Y}{Z} \quad \frac{1}{Z}$$

if the observer is located at the origin of the coordinate system looking in the Z direction. The coordinate $1/Z$ is a good monotonic approximation of the distance, and we can compute the shading as

$$S = \cos^2\theta * \frac{1}{Z}$$

Since the value of $1/Z$ is known only at the vertices of the polygons, it is necessary to perform a linear interpolation between two vertices to obtain the value of $1/Z$ along one edge. Once this interpolation has been performed, it is possible to compute the shading among the scan line as (Fig. 5)

$$S = (1 - \alpha)\frac{1}{Z_E}\cos^2\theta \quad + \quad \alpha\frac{1}{Z_F}\cos^2\theta \qquad (1)$$

Where α goes from 0 to 1 between E and F. It is remarkable to notice that the exact computation should be

$$S = \frac{\cos^2\theta}{(1 - \alpha)Z_E + \alpha Z_F} \qquad (2)$$

But the use of (1) does not show any noticeable degradation in the shading produced.

THE MACH BAND EFFECT

In attempting to represent a scene, the shading technique is subject to all the psychological illusions present in the visual process. Of interest to this discussion is a phenomenon thoroughly investigated by Mach [13] which explains how the retina performs some kind of two-dimensional filtering on the shading function of a scene. Each neuron, depending on the intensity of the light it receives, interacts with its neighbors and modifies their performances. The result of this interaction will be an attenuation of the low spatial frequencies and an amplification of the high spatial frequencies present in the shading. An example which is best suited to the discussion of this paper is shown in Fig. 1. Fig. 1(a) shows how the discontinuities in the value of the shading give a "fluted" aspect to each of the steps. Fig. 1(b) shows how a discontinuity in the first derivative of the shading gives the illusion of a small bump along the edge between two differently shaded surfaces.

CURVED SURFACES

In an effort to extend the class of objects that can be modeled by the computer, some techniques allowing the definition and the representations of curved surfaces have been developed. Coons introduced the Coons patch in 1964 [14]. At that time such surfaces were displayed by showing a grid of curves overlaid on the surface (Fig.2). This method presented all the disadvantages of wire frame perspectives, and no hidden-line removal method existed for this class of surfaces. At Cambridge, England, Armit [15] developed a system based on Coons patches. One of the facilities of the system was a modulation of the intensity of each segment of the curves by the distance from the segment to the observer. Without removing the hidden lines this method produced good looking pictures. At about the same time, Lee [16] developed an extension of the Coons patch called the rational Coons patch. The author is presently working at the University of Utah on the

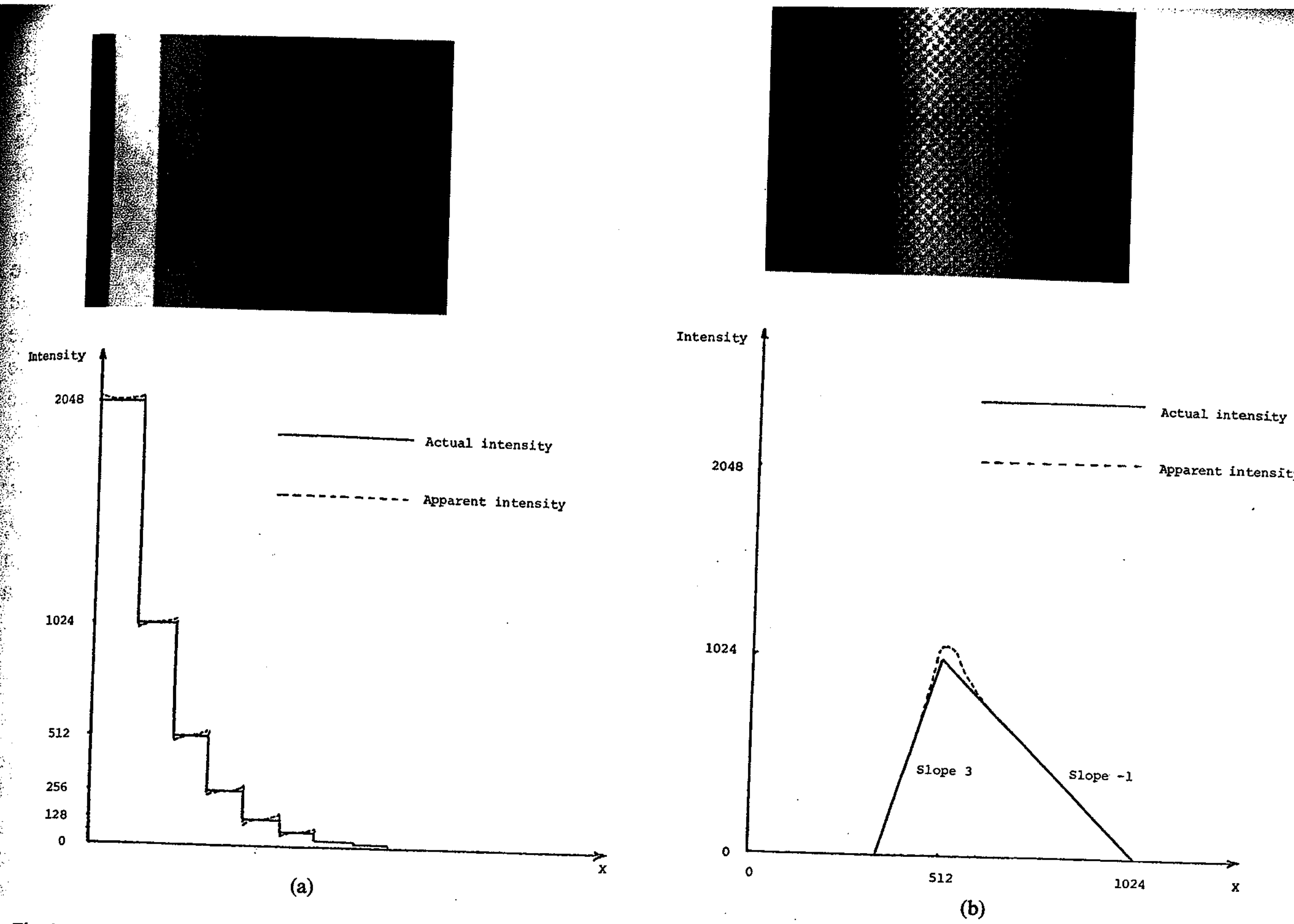


Fig. 1. (a) Mach band distortion produced by discontinuities in the value of the shading. (b) Mach band distortion produced by discontinuity in the first derivative of the shading.

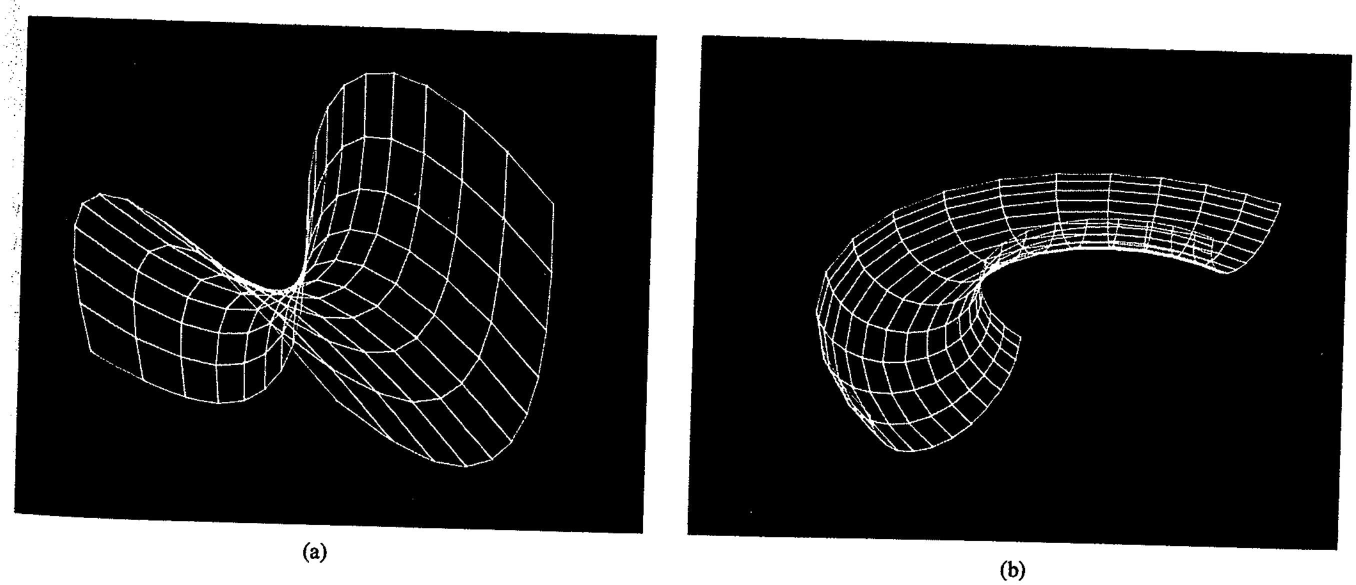


Fig. 2. Curved surfaces presented with conventional line drawng method. (a) Paraboloid hyperbole. (b) Quarter torus.

problems arising in the interactive design of rational Coons patches.

One of the properties of the rational Coons patch is the ability to reparametrize the patch without modifying its geometric shape. This can be viewed as a swashing of the grid of curves overlaid on the surface in one direction or another. In the two-dimensional perspective of such a patch, the spacing of the curves on the surface can give a cue of the depth properties of the surface. Since the reparametrization modifies the spacing it modifies also the depth cues; therefore one of the first problems the author was faced with was finding an automatic way of discovering the best parameterization for a given patch. It became rapidly evident that one should get rid of the grid on the surface by using a shading method in order to obtain the best depth representation of the surface.

Warnock and Romney had produced pictures of curved surfaces by approximating them with a large number of small planar polygons. Because of the Mach band phenomenon, this method produced pictures in which each small polygon was distinctly visible. Using Watkins' algorithm the author produced pictures of rational Coons patches, treating each grid element as a polygon (Fig. 3). The polygons thus obtained were not necessarily planar, but the fact that Watkins' algorithm accepts nonplanar polygons was very helpful at that point. As in the case of Warnock's and Romney's pictures, each polygon was very clearly visible and the grid had not disappeared.

From the explanation of the Mach band distortion it appears that in order to represent correctly the smooth aspect of a curved surface, the shading rule on this surface has to be continuous in value and if possible, in derivative. One way to achieve this would be to increase the number of polygons approximating the surface, but this is impractical for storage and time reasons. The approach described in this paper is to keep the polygon approximation of the surface, but to modify slightly the computation of the shading on each polygon so that continuity exists across polygon boundaries (Fig. 4).

Let us now examine how this continuity can be achieved. A typical data structure contains information about a certain number of lines and some more information connecting these lines into closed polygons. At a particular vertex common to several polygons, one might compute a normal for each polygon as a vector perpendicular to the plane of that polygon. To achieve continuity of the shading we have to have only one possible normal at any particular vertex. This normal could be computed as, for example, the average of the normals to each polygon associated with this particular vertex; but in the examples described in this paper an analytical description of the surface is available and it is possible to compute an exact normal at each vertex of the grid of polygons approximating the surface. Each polygon has a different shading for each of its vertices, and the shading at any particular point inside the polygon has to be computed as a continuous function of the shading at the vertices of the polygon.

If we now look at the projection of the polygon on the viewing plane, we see that one way to achieve this continuity is to compute the shading inside the polygon as two successive linear interpolations of the shading at the projection of the vertices. Given the projection of two edges AB and CD, and the scan line (Fig. 5), we assumed that the normal to the surface would be known at points A, B, C, D, which permits us to compute the shading at those four points. If E and F are the intersection of the scan line with AB and CD, respectively, and P is any point on the scan line between E and F, the shading at point E can be computed as a linear interpolation of S_A and S_B of the form

$$S_E = (1 - \alpha) * S_A \quad + \quad \alpha * S_B$$

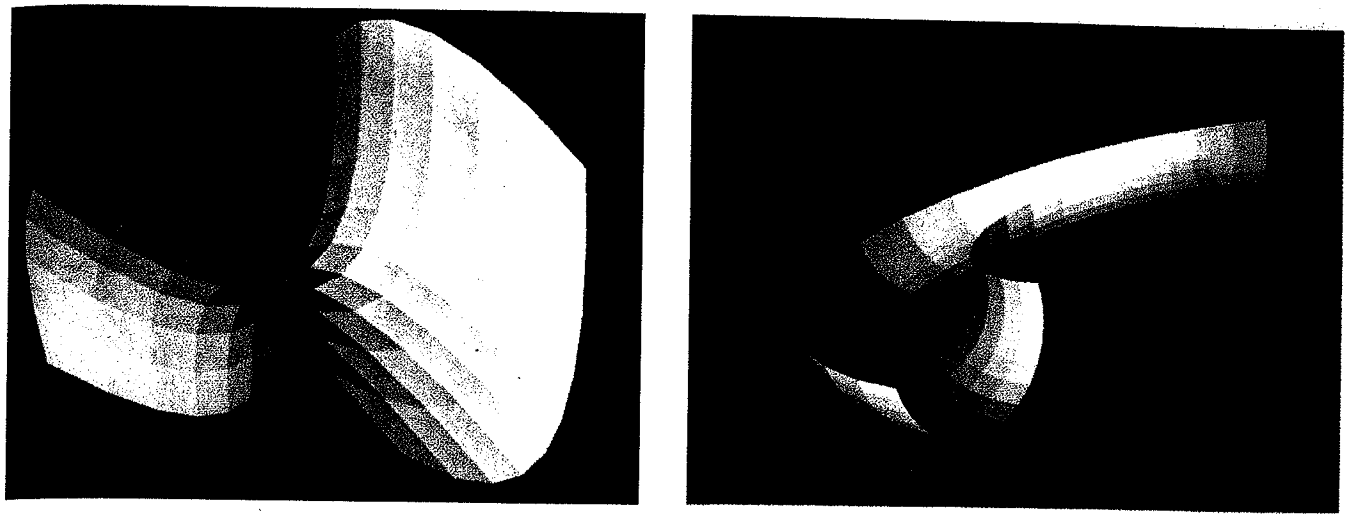
where α is the coefficient ($0 \leq \alpha \leq 1$) expressing the position of E on the segment AB. If E is identical to A then $\alpha = 0$, and if E is identical to B then $\alpha = 1$. In a very similar fashion we can compute S_F as a linear interpolation of S_C and S_D and S_P as a linear interpolation of S_E and S_F.

$$S_F = (1 - \beta) * S_D + \beta * S_C \qquad (0 < \beta < 1)$$
$$S_P = (1 - \alpha) * S_E + \alpha * S_F \qquad (0 < \alpha < 1)$$

It can be easily verified from the equations above that if

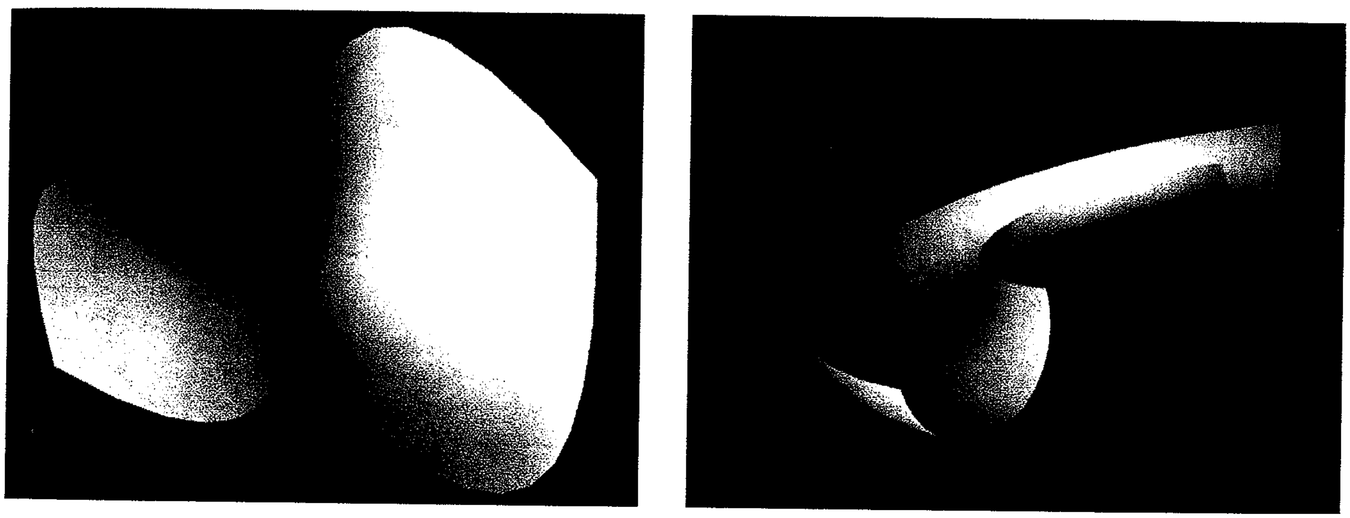
$$P \equiv A, \quad \text{then} \quad S_P \equiv S_A$$
$$P \equiv B, \quad \text{then} \quad S_P \equiv S_B$$
$$P \equiv C, \quad \text{then} \quad S_P \equiv S_C$$
$$P \equiv D, \quad \text{then} \quad S_P \equiv S_D$$

In order to reduce the computation of a new shade for each point to a minimum, the very efficient technique developed by Watkins was extended to include this computation. The following is a very concise description of Watkins' algorithm (for complete understanding of the mechanisms, refer to Watkins" Ph.D. thesis [9]). If the picture is scanned from top of bottom the following information is computed for each polygon edge:



(a) (b)

Fig. 3. Same curved surfaces presented with Watkins algorithm. (a) Computation time: 1 min 30 s. (b) Computation time: 1 min 20 s.



(a) (b)

Fig. 4. Same curved surfaces presented with author's method. (a) Computation time: 1 min 45 s. (b) Computation time: 1 min 35 s.

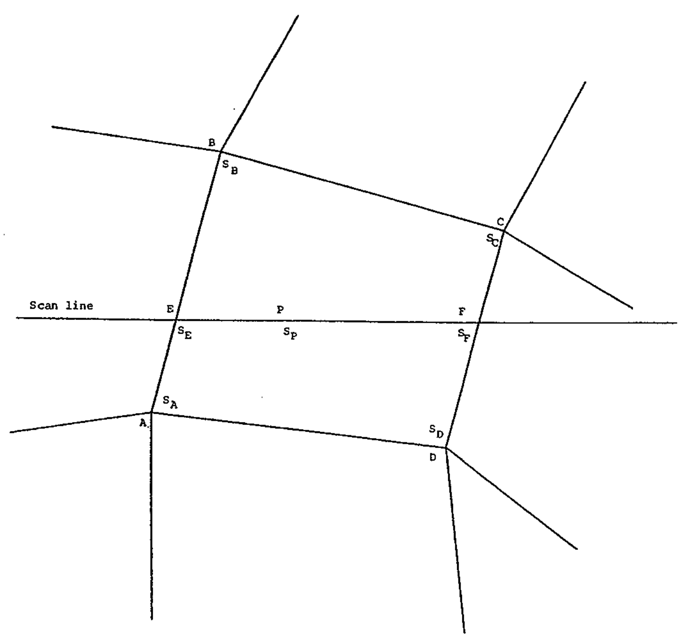
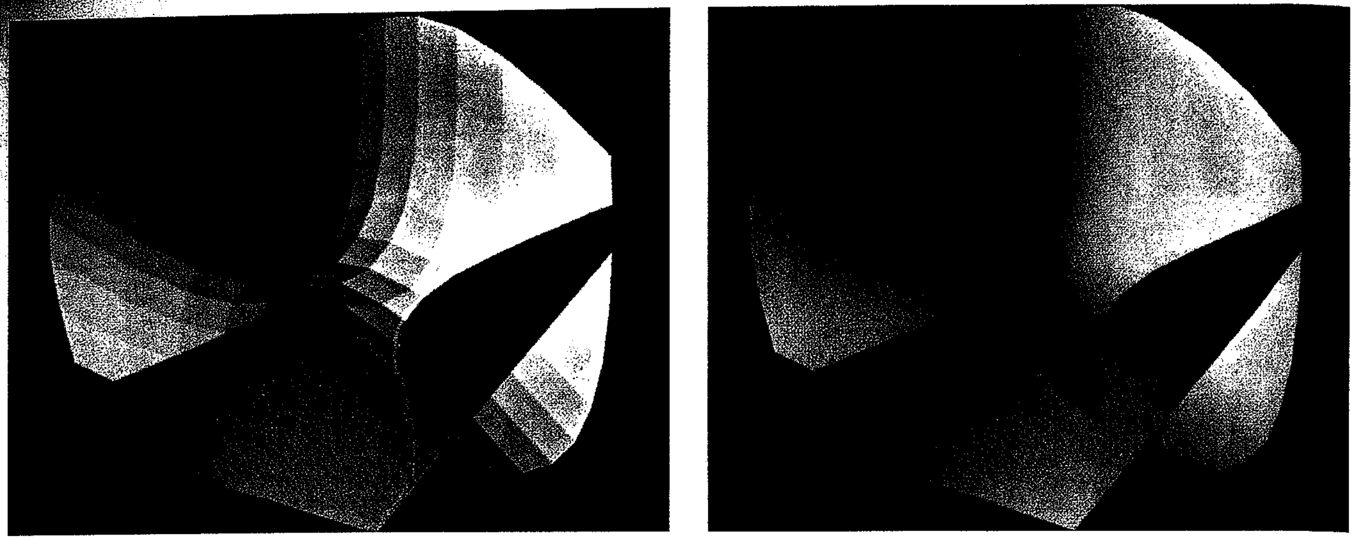


Fig. 5. Projection of one polygon intersected by the scan line.



(a) (b)

Fig. 6. Curved surface intersected by a plane presented with (a) Watkins' method (computation time 2 min), and (b) the author's method (computation time 2 min 15 s).

1) the number of the first scan line this edge will intersect;
2) how many scan lines below this one it will intersect;
3) the X and Z coordinates of the highest point of the edge;
4) 4) the slope in X and Z of this edge.

We can very easily add the following information:
5) the shading S of the surface at the highest point of this edge:
6) a "slope" of this shading along this edge.

This "slope" is computed as

$$\Delta S = \frac{S_2 - S_1}{n}$$

Where S_2 and S_1 are the shading at the two endpoints of the edge and n is the number of scan lines intersecting this edge.

Given this information it is now very easy to compute the shading on a given scan line. As the computation proceeds, an edge will become "active" when its first point is reached by a scan line. At that stage, we know the XYZ coordinates of this point and the value of the shading at the point on the surface. For the next scan line it is sufficient to add the "slope" to the coordinate information of the point as well as of the shading to find a new point and a new shading. Given a scan line and all the edges intersecting this scan line, edges belonging to the same polygon are paired to form segments (a segment could be viewed as the intersection of one of the polygons and the scan plane). A segment is created when an edge becomes "active." The segment contains the coordinates of its endpoints, the value of the shading at its endpoints, and the different slopes necessary to update this information from scan line to scan line. When an edge leaves the "active" list, it is necessary to rearrange the segments by deleting or merging some blocks of information. The hidden-lines computation is performed at that point and we are left with a number of segments totally or partially visible. For each point of the scan line on the visible part of a segment, we can compute a coefficient

$$\alpha = \frac{X_P - X_E}{X_F - X_E} \qquad \text{(Fig. 5)}$$

and the shading as

$$S = (1 - \alpha) * X_E + \alpha * S_F$$

The linear interpolation which has been used here produces a shading which is continuous in value but not in derivative across polygon boundaries. The resulting Mach band effect can be observed mostly in the vicinity of the silhouette curves and where the surface bends sharply. Interpolation schemes more powerful that the linear interpolation could probably be used but the improvement obtained with such schemes would not compensate the loss of efficiency of the present algorithm and would make a hardware implementation unpracticable.

TIMING

At this point it is important to consider the time degradation we have imposed on Watkins' algorithm. Our modifications can be split into two categories. The first category is the extra information that is requested about each edge or segment. The second is the point-by-point computation of the shading of a scan line.

The first category adds hardware cost but should not slow down the process since all segment information is handled in parallel. Indeed, this is true only for a hardware implementation and it puts some more burden on the memory requirements. In the software simulation, about 40 percent of the time is spent in the routine which creates and updates segments from scan line to scan line. The amount of information attached to each segment was previously the X and Z coordinates of the endpoints of the segments, and is now augmented by the shading value of those two points. This multiplies the time spent in this routine by 1.5, or the total time taken by the modified algorithm by less than 1.2.

As for the second category, the proposed modification uses exactly the same hardware and does not take more time than the old method since the computation to be performed at that point is still a linear interpolation between two values provided by the segment handling routine.

INTERSECTIONS

The shading rule which we have described gives the illusion of a smooth curved surface when, in fact, this surface is described by a set of small polygons. It was necessary to keep this polygon approximation so that the computation of intersections could be handled easily using existing methods. As can be seen in Fig. 6, there is no difference here between the intersection computed by Watkins and the one computed with the modified algorithm. This does not seem to be a serious drawback and the final appearance of the picture remains good even when there are intersecting surfaces.

ACKNOWLEDGMENT

The author would like to thank I.E. Sutherland for the discussions that initiated the ideas presented here and G. Watkins for his help in explaining and modifying his algorithm.

REFERENCES

[1] L. G. Roberts, "Machine perception of three-dimensional solids." M.I.T. Lincoln Lab., Cambridge, Mass. Tech. Rep. 315, May 22, 1963.

[2] R. Galimberty and U. Montanari. "An algorithm for hidden line elimination." Istituto di Electtrotechnica ed Elettronica. Relazione interna, Apr. 1968.

[3] P. R. Kubert. "A computer method for perspective representation of curves and surfaces." Aerospace Corp., San Bernardino. Calif., Dec. 1968.

[4] P. Loutrel, "A solution to the 'hidden-line' problem for computer drawn polyhedra," New York Univ., New York, N.Y., Tech. Rep. 400-167, Sept. 1967.

[5] P. G. Comba, "A procedure for detecting intersections of three-dimensional objects," IBM New York Scientific Center, New York, N.Y., Rep. 39,020, Jan. 1967.

[6] R. A. Weiss, "Be vision, a package of IBM 7090 Fortran programs to draw orthographic views of combinations of plane and quadric surfaces." *J. Ass. Comput. Mach.* vol. 13, Apr. 1966, pp. 194-204.

[7] G. W. Romney, "Computer assisted assembly and rendering of solids," Dep. Comput. Sci., Univ. of Utah, Salt Lake City, Tech. Rep. TR 4-20, 1970.

[8] J. E. Warnock, "A hidden surface algorithm for computer generated halftone pictures." Dep. Comput. Sci., Univ. of Utah, Salt Lake City, Tech. Rep. 4-15, June 1969.

[9] G. S. Watkins, "A real time visible surface algorithm." Dep. Comput. Sci., Univ. of Utah, Salt Lake City. Tech. Rep. UTEC-CSc 70-101, July 1970.

[10] R. S. Rougelot and R. Shoemaker, "G. E. real time display." General Electric Co., Syracuse, N.Y., NASA Rep. NAS 9-3916.

[11] MAGI, Mathematical Applications Group, Inc., "3-D simulated graphics," *Datamation*, vol. 14, Feb. 1968, p. 69.

[12] A. Appel, "The notion of quantitative invisibility and the machine rendering of solids," *Ass. Comput. Mach. Conf. Proc.*, p. 387, 1967.

[13] F. Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina.* San Francisco: Holden-Day, 1965

[14] S. A. Coons, "Surface for computer-aided design of space forms," M.I.T., Cambridge, Mass., Project MAC. Tech. Rep. MAC-TR-41, June, 1967.

[15] A. P. Armit, "A multipatch design system for Coons' patches, Univ. of Cambridge Computer Aided Design Group, Dec. 1968.

[16] T. M. P. Lee, "Three-dimensional curves and surfaces for rapid computer display," Harvard Univ., Cambridge, Mass., Tech. Rep. ESD-TR-69-189, Apr. 30, 1969

[17] W. J. Bouknight, "A procedure for generation of three-dimensional half-toned computer graphics presentations," *Commun. Ass. Comput. Mach.*, vol. 13, 1970.

[18] J. Bouknight and K. Kelley, "An algorithm for producing half-tone computer graphics presentations with shadows and movable light sources," in *1970 Spring Joint Computer Conf., AFIPS Proc.*, vol. 36. Montvale, N. J.: AFIPS Press, 1970, pp. 1-10.