# Computer Generated Animation of Faces

Frederick I. Parke, University of Utah

This paper describes the representation, animation and data collection techniques that have been used to produce "realistic" computer generated half-tone animated sequences of a human face changing expression. It was determined that approximating the surface of a face with a polygonal skin containing approximately 250 polygons defined by about 400 vertices is sufficient to achieve a realistic face. Animation was accomplished using a cosine interpolation scheme to fill in the intermediate frames between expressions. This approach is good enough to produce realistic facial motion. The three-dimensional data used to describe the expressions of the face was obtained photogrammetrically using pairs of photographs.

KEY WORDS AND PHRASES: computer graphics, half-tone rendering, smooth shading, computer animation, flexible surfaces, polygonal surfaces, facial topology, cosine interpolation, three-dimensional data acquisition.
CR CATEGORIES: 8.2, 3.41, 4.41, 6.35

## INTRODUCTION

The human face is a challenge for computer animation for at least two reasons. First the face is not a rigid structure but is a complex flexible surface. How is the motion of such a surface specified? Secondly faces are very familiar to us, we have a well developed sense of what expressions and motions are natural for a face. We notice small deviations from our concept of how a face should appear.

This paper describes a fairly simple way of representing the face and an animation technique that allows the production of realistic half-tone animated sequences of the face changing expression. The paper also describes the method used to collect the data for the faces. These techniques could be used to animate other flexible surfaces.

## REPRESENTATION OF THE FACE

The face is a very complex three-dimensional surface. This surface is flexible. It usually contains creases, and it has color variation. What is the best way to represent such a surface that allows both animation and half-tone rendering? One possibility would be to find an analytic surface or collection of analytic surface patches (1) to approximate the surface of the face. Assuming this were feasible, there remains the problem of animating this surface or collection of patches. Again assuming that appropriate animation techniques were available for such surfaces, there still remains the problem of producing half-tone renderings of the surfaces. Hidden surface and half-tone algorithms exist for quadric surfaces (2,3) but they tend to be quite expensive. One can imagine similar algorithms for surfaces of higher degree, but would expect them to increase rapidly in expense as the degree of the surface increased. For this reason, when half-tone renderings are desired, surfaces of high degree are usually approximated by a skin of polygons.

In order to approximate the face with analytic patches, one would expect these patches to be at least quadric, and probably of higher degree. The approach of approximating the face with analytic surfaces leads to approximating the approximate surfaces with polygons. This seems a rather complex and roundabout approach.

The approach taken in this paper is one first used by Henri Gouraud (4). His approach was to directly approximate the surface of the face with a non-analytic skin of polygons. This skin was constructed by sampling the surface of the face at a number of points and connecting these points to form a skin of polygons.

In order to produce a half-tone rendering of objects in a three-dimensional space several problems must be solved.

*This research was supported in part by the University of Utah Computer Science Division and by the Advanced Research Projects Agency of the Department of Defense, monitored by Rome Air Development Center, Griffis Air Force Base, New York 13440, under contract F30602-70-C-0300.

The first of these problems is usually referred to as clipping. This is the problem of determining if all or part of an object is within a viewing space. The viewing space is a pyramid defined by the position of the view, the direction the viewer is looking and his viewing angle. An example of a viewing space would be that part of the universe visible through a window, assuming that all objects except the window frame were transparent and could not occlude other objects. The second problem is the detection of hidden surfaces. In other words, which surfaces are in front of other surfaces when seen from a given position. The last problem is one of determining the shading of the visible surfaces and producing the shaded image on some output device, normally a CRT. The shading usually depends on the orientation of the surface with respect to the viewer and the light source.

Approximation with polygons has several advantages. For polygonal surfaces, the problems listed above have been solved by a number of algorithms (5-11). These algorithms are fast and inexpensive when compared to algorithms for surfaces of higher degree. At least one of these algorithms (9) is implemented in hardware and another (11) is currently being implemented in hardware. Also, the development by Gouraud (4) of a smooth shading algorithm for polygonal surfaces makes it possible to give a continuously curved appearance to a surface made up of polygons.

For polygonal shading the shade of each polygon is constant across the polygon. This shade is a function of the angle between the normal to the polygon and a line from the light source to the polygon. For Gouraud's smooth shading, however, the shade is not constant across the polygon. It is a function of the angle between the normal at each vertex of the polygon and a line from the light source to the polygon, and the position within in the polygon. The normal at a vertex is the average of the normals of the polygons that have this vertex in common.

In smooth shaded renderings a special procedure is necessary if creases are to be visible. Creases can be made visible by "doubling" vertices. Since creases can occur only along the boundary between adjoining polygons, each vertex along the crease is doubled. One vertex of the pair belongs to polygons on one side of the crease and the other vertex of the pair belongs to polygons on the other side of the crease. This causes separate normals to be computed. When the polygons are shaded, there will be a shading discontinuity along the boundary and the crease will be visible.

Figure 1 shows a face rendered with polygonal shading and a different expression of the same face rendered using the smooth shading algorithm.

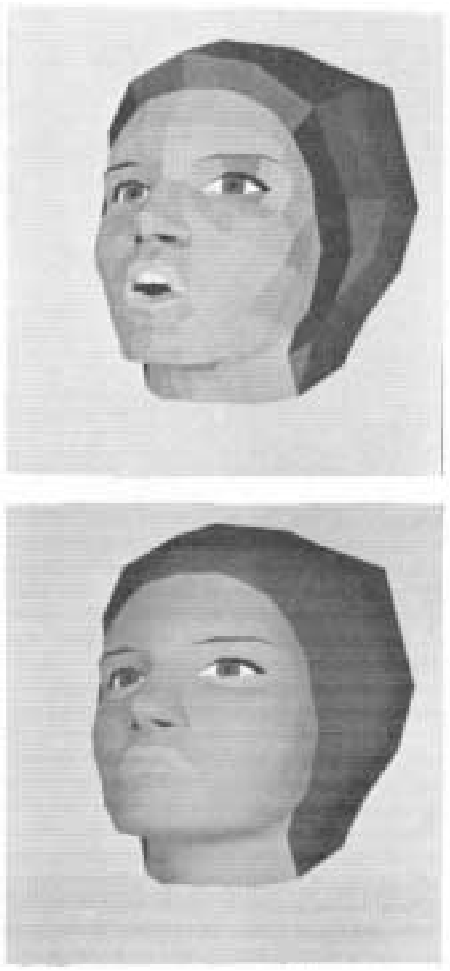

Figure 1
Two expressions of the same face. The top one was rendered using polygonal shading. The bottom one was rendered using Gouraud's smooth shading algorithm.

Having decided to use a polygonal representation, how does one go about approximating a face with polygons? There are several things to keep in mind.

1. To get good smooth shading, the density of polygons should be highest in the areas of highest curvature (the nose, mouth, around the eyes and the edge of the chin) and lowest in the areas of lowest curvature (the forehead, cheeks and neck).

2. Where creases occur on a face (under the eyes, the side of the nose, the edge of the lips and the corner of the mouth), edges of polygons must coincide with the creases. A polygon may not span a crease.

3. Use the smallest number of polygons consistent with good results. The reasons for this are obvious: a smaller amount of data, faster picture generation and minimization of the data acquisition problem.

4. If animation is desired, the polygons must be layed out in a way that allows the face to flex naturally. The polygons should remain approximately planar as the face flexes.

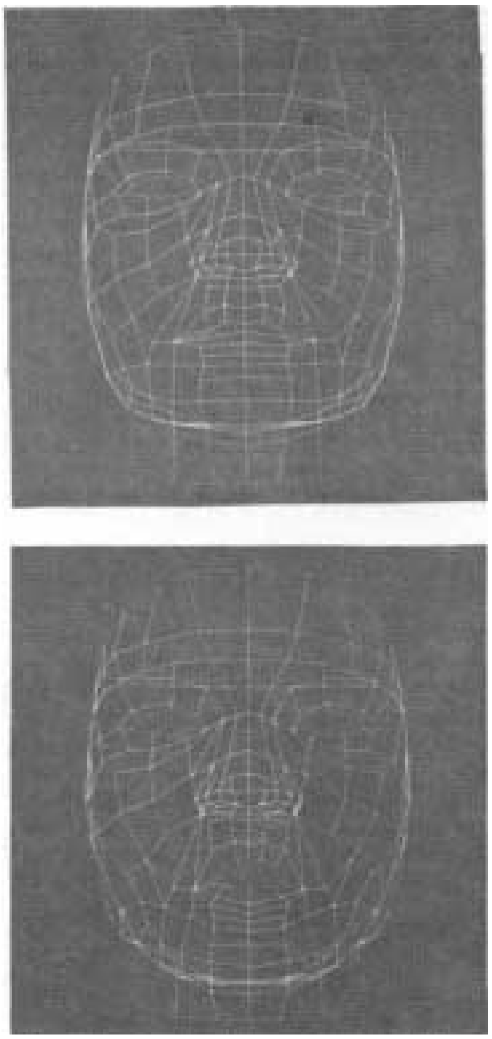

Figure 2
Two pictures showing the skin of polygons used to represent the face. Note that the polygons of the skin change shape and position as the face changes expression.

5. Since the face is approximately symmetric, we need worry only about one side of the face. The other side is obtained by "mirroring" or reflecting about the plane of symmetry.

6. Each polygon will have associated with it a color. Where color boundaries occur on the face, the lips and eyebrows, for example, polygon edges must coincide with these boundaries. A polygon may not span a color boundary.

Keeping these things in mind the next step is to find a cooperative assistant who will allow you to draw or paint a set of polygons on his or her face. After drawing the polygon skin on one half of the face, ask the assistant to assume a number of different expressions. For each expression observe how well these polygons approximate the face. After modifying the polygon set several times you should arrive at a reasonable set of polygons to represent the face. Figure 2 shows the skin of polygons used to produce the faces shown in figures 1,3 and 4. One-half of this skin contains 124 polygons defined by 202 vertices.

A unique point number is assigned to each vertex of the skin. The skin is then specified by going around each polygon in a clockwise direction and recording the point numbers of its vertices.

The details of the face are very important in achieving realistic results. Figure 3 shows the effect details, such as the eyes, eyebrows, eyelashes and teeth have on the realism of the face. The eyebrows and teeth were included simply by adding polygons of the appropriate color. The illusion of eyelashes was achieved by changing the color of existing polygons directly above the eyes. The face by itself is not very realistic. It was necessary to complete the head in order to be convincing. Figure 3 shows how a "bonnet" of hair was used to complete the head.

Color is an important feature of the face. Each polygon has a color associated with it. This color is made up of three components; red, green and blue. By

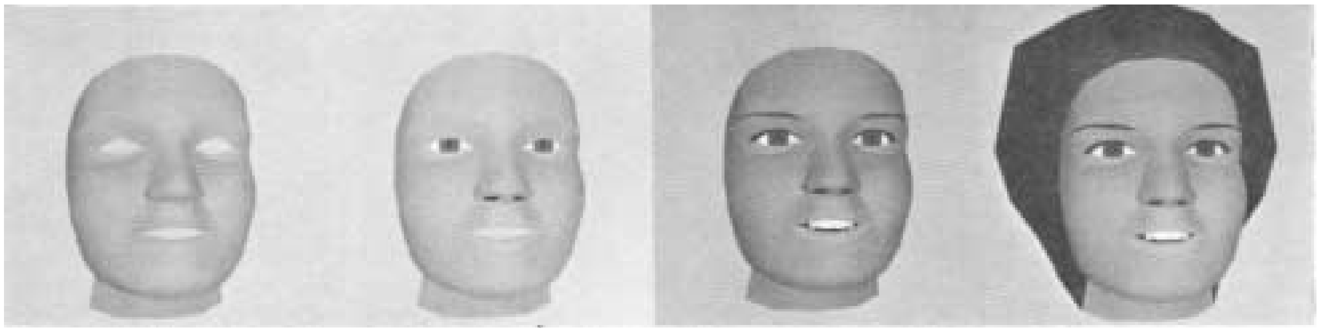

Figure 3
Four pictures that illustrate the effect of details on the realism of a face. The first picture shows the face alone. The next picture shows the face with eyes and nostrils. In the next picture, teeth, eyelashes, eyebrows and the inside of the mouth were added. The last picture shows the complete head.

specifying the value of each component it is possible to achieve the desired colors. Color half-tone renderings are produced by scanning out the picture three times, once for each of the primary colors. The appropriate color filter is placed in front of the camera lens before each scan.

After some experimentation the component values for flesh-tone and the other colors of the face were determined. These component values depend on a number of variables, including: the phosphor of the CRT, the type of filters used, the type of film used, the intensity setting of the CRT, and the compensation function used to overcome the non-linear characteristics of the CRT.

## ANIMATION

Assuming that we have a satisfactory skin of polygons for the face, how do we animate it? We would like to specify the motion of the surface in the simplest way consistent with natural motion.

The approach taken in this research is somewhat similar to the approach taken by the conventional animator. The animator specifies the desired motion by blocking it out with a series of key drawings. He then gives these key drawings to the assistant animators who generate the required intermediate frames. For the computer animation, the key drawings are replaced by data files describing the face for each of a number of different expressions. The data for each expression or "phase" of the face consists of the three-dimensional position of each point defining the polygon skin used to represent the face. Figure 4 shows two phases of a face.

The animation program takes the place of the assistant animators and generates the required intermediate frames between the phases as the face changes expression.

To change the face from one expression to another is a matter of moving each point a small distance in successive frames. The position of each point of the skin in each frame is determined by interpolating between the previous phase position and the next phase position. Figure 2 shows how the polygons of the skin change shape and position as the face changes expression.

Since the face is governed by physical laws, its motion is not linear but tends to accelerate and decelerate. A cosine interpolation scheme was used to approximate the acceleration and deceleration of the facial motions. Each frame has associated with it a phase number. This phase number is a real number whose integer part refers to the previous phase and whose fractional part indicates the position of this frame between the previous phase and the next phase. For example, if phase 2 is a smile and phase 3 is a frown then the phase number 2.5 means an expression halfway between a smile and a frown. Each component of a point's position is computed using the following algorithm.

$$\text{current position} = \text{position in the previous phase} + C * \text{difference}$$

where

$$\text{difference} = \text{position in next phase} - \text{position in previous phase}$$

$$C = (1.0 - \cos(\Phi))/2.0$$

and

$$\Phi = \text{phase fraction} * 3.14159$$

## DATA ACQUISITION

Measuring the three-dimensional position of points on the surface of a face or any other complex object is a significant problem.

If we still have our assistant whose face is painted with polygons, we ask the assistant to assume a number of different expressions. We "freeze" each expression photographically. For each expression a

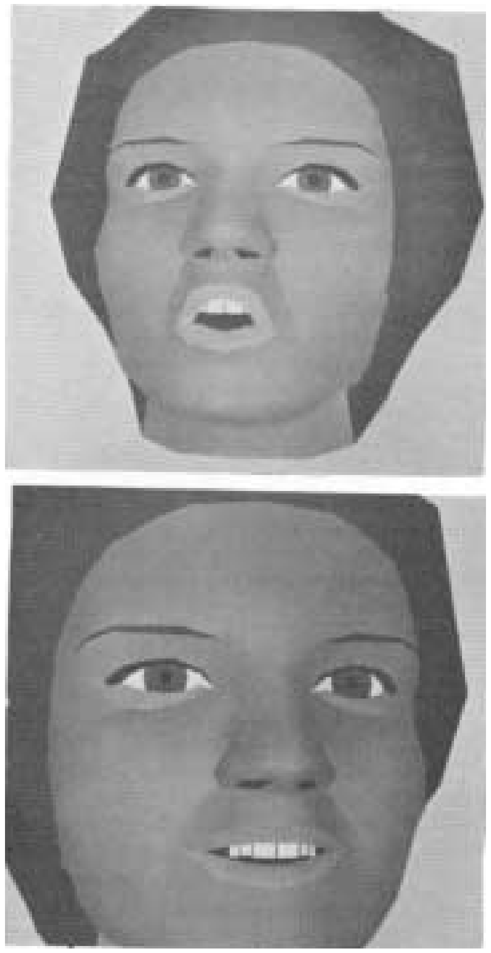

Figure 4
Two phases of a face.

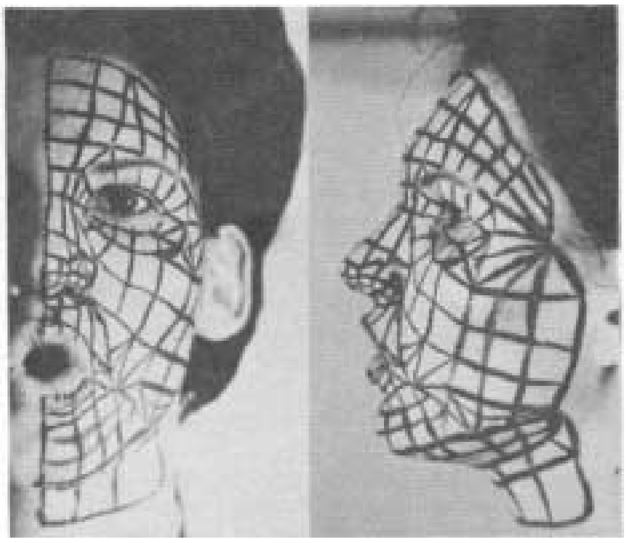

Figure 5
A typical pair of data photographs.

pair of orthogonal views of the face is taken, one from directly in front and one from the side. Figure 5 shows a typical pair of these photographs. Using these pairs of photographs we establish an origin and a coordinate system. The three-dimensional position of each point is measured directly from the photographs.

Note that the coordinate system should be chosen such that two of the coordinate axes define the symmetry plane of the face. This facilitates the mirroring or reflection operation necessary to obtain the data for the other half of the face.

This data collection method has some shortcomings. Photographs are not orthographic projections but are perspective projections. Therefore, the images on the photographs are somewhat distorted. This distortion can be reduced by using long focal length lenses when the pictures are taken. Some adjustment of the data may be necessary due to this distortion. Another shortcoming is that some points on the face may not be visible in both views. A best guess must be made for at least one of the coordinates of these occluded points.

## THE ANIMATION PROGRAM

The animation program contains arrays to store the topology and phase data for the face. Up to three phases may be stored in the program. The topology and phase data is read in from data files. The user of the program interacts with it to specify which data files he wants read in.

The phase data files consist of the data for a sequence of points. For each point there is a point number and a three-dimensional position. The topology data consists of a specification for each polygon of the face. The polygon specification is made up of the point numbers of the vertices of the polygon and its color.

After the desired data is read in, the data for the other half of the face is constructed by mirroring or reflecting the data for the first half of the face.

For each frame of a sequence a number of tasks must be accomplished in order to compute the data needed to pass on to the hidden surface and shading algorithms.

Using the phase number associated with each frame, the program interpolates the phase data to get the position of each vertex of the skin for this frame.

These point positions are specified in a coordinate system centered near the center of the head. The hidden surface algorithm requires the data to be specified in a different coordinate system. The new coordinate system, refered to as the viewing system, has its origin at the position we wish to look from. The Z axis of the viewing system must be pointing in the direction we wish to look. The position data must be transformed (12, 13) into this new coordinate system. The animation program first translates the data so the origin moves from the center of the head to the position we wish to look from. It then rotates the coordinate system so that the Z axis of the viewing system is pointing toward the position we wish to look at.

After the data is transformed into the viewing system, the normal to each polygon is computed. Using these normals, the normal at each vertex of the skin is computed. This is done for each vertex by averaging the normals of the polygons that have the vertex in common.

For each frame of a sequence the following parameters are passed on to the hidden surface and shading algorithms:

1. The viewing angle (this is used to determine the viewing space and is also used in the perspective transformation (12,13)).
2. The color of each polygon.
3. The position of the beginning and ending points of each edge of each polygon and the normals at these points.
4. The desired resolution.

The animation program was written in SAIL (14), an extended algol for the PDP-10.

## IMPLEMENTATION

The system used to produce animated sequences, and the half-tone renderings included in this paper is shown in Figure 6. This system uses two PDP-10 computers. One of these is a dedicated machine that allows only one user at any given time.

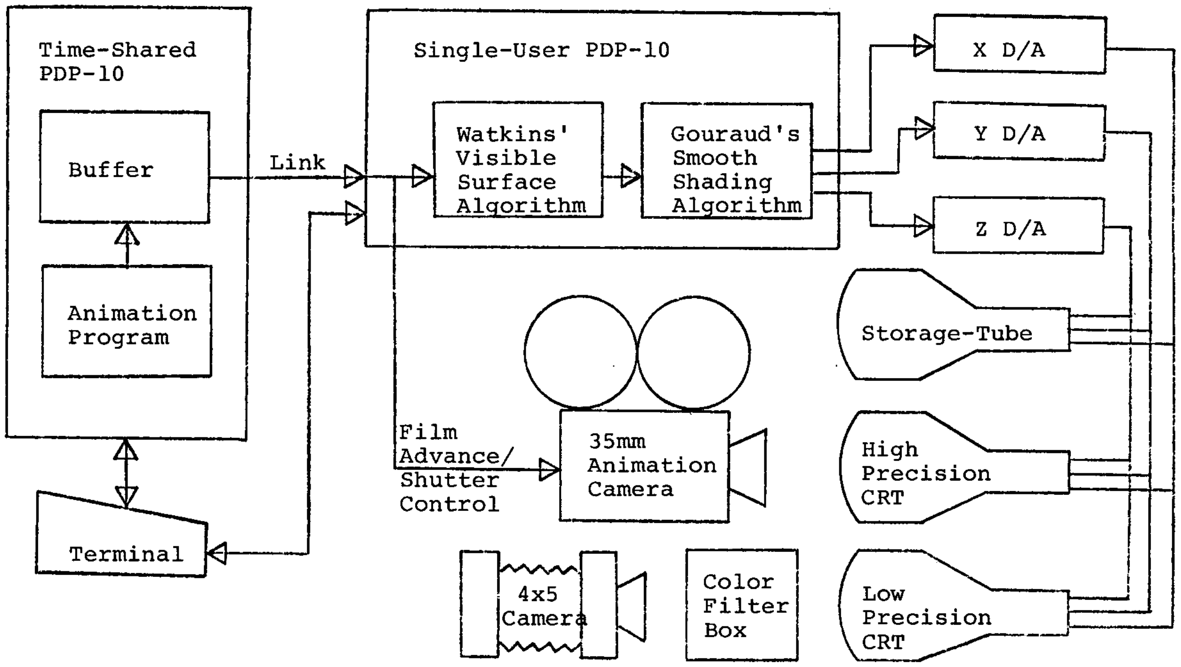

Figure 6
System configuration.

This processor is interfaced to the special equipment needed to produce half-tone pictures. The other PDP-10 is time-shared, and runs under the TENEX operating system. There is a link between the machines that allows data to be transfered between them.

This system allows us to take advantage of the TENEX operating system, particularly the file system, on one machine and the special half-tone display equipment on the other machine.

When one wishes to use this system, he connects the terminal to the single-user PDP-10 and loads a program which contains; a software version of Watkin's visible surface algorithm, Gouraud's smooth shading algorithm, a procedure to calibrate the half-tone displays and procedures to handle the single-user side of the data transfers across the link. When this program begins execution, it first allows the user to calibrate the display equipment. It then initializes the receiving side of the data link and goes to sleep. At this point the user switches the terminal to the time-shared machine. After logging in, the user loads and starts the execution of the animation program. This program asks the user a number of questions, including: which phase data files he wants read in, the desired number of frames between phases, the resolution to use, where he wants to look, where he wants to look from, and whether he wants smooth or polygonal shading. After receiving this information, the program begins processing the first frame of the sequence.

Data to be passed to the hidden surface and shading algorithms is stored into one of two buffers. When a buffer is full, a flag is set. This causes the animation program to transfer to the other buffer and wakes up the single-user program. When the single-user program wakes up, it transfers the data out of the full buffer into its memory, resets the flag and goes back to sleep. Resetting the buffer flag allows the animation program to reuse the buffer.

When all the data for a single frame has been transfered, the single-user program begins working on it to generate the half-tone image. The animation program goes on to the next frame of the sequence. While the single-user program is in the process of generating the half-tone image, it ignores the buffer flags set by the animation program. This means that as soon as the animation program fills both buffers it must wait until the single-user program completes the picture and empties a buffer.

The output of the single-user program goes to three digital-to-analog converters which in turn drive any combination of the display devices shown in Figure 6.

This system works well if the time-shared system is not heavily loaded. If the time-shared system is heavily loaded the single-user is idle much of the time waiting for data.

All of the half-tone renderings shown in this report were produced with the high precision display using a resolution of 1024x1024. At this resolution it takes about 2½ minutes to scan out a single black-and-white picture.

Animated sequences are recorded using a 35 mm animation camera. Film advance and shutter are under program control. Animated sequences of the face are produced at the rate of about 20 frames per hour.

ACKNOWLEDGMENTS

I am grateful to Professors R. E. Stephenson and I. E. Sutherland and to Barry Wessler and Ed Catmull for their help and encouragement, also to Mike Milochik for his photographic assistance.

REFERENCES

1. Coons, S. A., "Surfaces for Computer Aided Design of Space Forms", M.I.T., Cambridge, Mass., Project MAC Report MAC-TR-41, June 1967.

2. Mahl, R., "Visible Surface Algorithm for Quadric Patches", Computer Science, University of Utah, Technical Report UTEC-CSc-70-111, December 1970.

3. Weiss, R. A., "Be Vision, A Package of IBM 7090 Fortran Programs to Draw Orthographic Views of Combinations of Plane and Quadric Surfaces", JACM, vol. 13, April 1966, pp. 194-204.

4. Gouraud, H., "Computer Display of Curved Surfaces", Computer Science, University of Utah, Technical Report UTEC-CSc-71-113, June 1971.

5. Wylie, C., Romney, G., Evans, D., and Erdahl, A., "Half-tone Perspective Drawing by Computer", Proc FJCC, vol. 31, pp. 49-58, 1967.

6. Appel, A., "The Notion of Quantitative Invisibility and the Machine Rendering of Solids", Proc ACM, vol. 14, pp. 387-393, 1967.

7. Kelley, K. C., "A Computer Program for the Generation of Half-Tone Images with Shadows", Coordinated Science Laboratory, University of Illinois, Report R-444, November 1969.

8. Romeny, G. W., "Computer Assisted Assembly and Rendering of Solids", Rome Air Development Center, Griffiss Air Force Base, New York, Technical Report RADC-TR-69-365, September 1969.

9. Rougelot, R. S. and Shoemaker, R., "G. E. Real Time Display", General Electric Co., Syracuse N. Y., NASA Report NAS 9-3916.

10. Warnock, J. E., "A Hidden Surface Algorithm for Computer Generated Halftone Pictures", Computer Science, University of Utah, Technical Report 4-15, June 1969.

11. Watkins, G. S., "A Real-Time Visible Surface Algorithm", Computer Science, University of Utah, Technical Report UTECH-CSc-70-101, June 1970.

12. Coons, S. A., "Transformations and Matrices", Notes for the 1967 Summer School on Computer Graphics for Designers, University of Michigan, June 5-16, 1967.

13. Ahuja, D. V. and Coons, S. A., "Geometry for Construction and Display", IBM Systems Journal, vol. 7, pp. 188-205, 1968.

14. Swinehart, D. and Sproull B., "SAIL", Stanford Artificial Intelligence Project Operating Note No. 57.1, April 1970.