Graphics and
Image Processing

Texture and Reflection in Computer Generated Images

James F. Blinn and Martin E. Newell
University of Utah

In 1974 Catmull developed a new algorithm for rendering images of bivariate surface patches. This paper describes extensions of this algorithm in the areas of texture simulation and lighting models. The parametrization of a patch defines a coordinate system which is used as a key for mapping patterns onto the surface. The intensity of the pattern at each picture element is computed as a weighted average of regions of the pattern definition function. The shape and size of this weighting function are chosen using digital signal processing theory. The patch rendering algorithm allows accurate computation of the surface normal to the patch at each picture element, permitting the simulation of mirror reflections. The amount of light coming from a given direction is modeled in a similar manner to the texture mapping and then added to the intensity obtained from the texture mapping. Several examples of images synthesized using these new techniques are included.

Key Words and Phrases: computer graphics, graphic display, shading, hidden surface removal
CR Categories: 3.41, 5.12, 5.15, 8.2

A version of this paper was presented at SIGGRAPH '76: The Third Annual Conference on Computer Graphics, Interactive Techniques, and Image Processing, The Wharton School, University of Pennsylvania, July 14–16, 1976.

This work was supported in part by ARPA under Contract DAH15-73-C-0363. Author's address: Computer Science Department, University of Utah, Salt Lake City, UT 84112.

Introduction

In 1974 Edwin Catmull [2] developed an algorithm for rendering continuous tone images of objects modeled with bivariate parametric surface patches. Unlike most earlier algorithms [6, 8, 9, 10], which require that objects be approximated by collections of planar polygons, Catmull's algorithm works directly from the mathematical definition of the surface patches. The algorithm functions by recursively subdividing each patch into smaller patches until the image of each fragment covers only one picture element. At this stage, visibility and intensity calculations are performed for that picture element. Since the subdivision process will generate picture elements in a somewhat scattered fashion, the image must be built in a memory called a depth buffer or Z-buffer. This is a large, random access memory which, for each picture element, stores the intensity of the image and the depth of the surface visible at that element. As each patch fragment is generated, its depth is compared with that of the fragment currently occupying the relevant picture element. If greater, the new fragment is ignored, otherwise the picture element is updated.

This paper describes extensions of Catmull's algorithm in the areas of texture and reflection. The developments make use of digital signal processing theory and curved surface mathematics to improve image quality.

Texture Mapping

Catmull recognized the capability of his algorithm for simulating variously textured surfaces. Since the bivariate patch used is a mapping of the unit square in the parameter space, the coordinates of the square can be used as a curvilinear coordinate system for the patch. It is a simple matter for the subdivision process to keep track of the parameter limits of each patch fragment, thereby yielding the parameter values at each picture element. These parameter values may then be used as a key for mapping patterns onto the surface. As each picture element is generated, the parametric values of the patch within that picture element are used as input to a pattern definition function. The value of this function then scales the intensity of that picture element. By suitably defining the pattern function, various surface textures can be simulated.

As Catmull pointed out, simply sampling the texture pattern at the center of each picture element is not sufficient to generate the desired picture, since two adjacent picture elements in the image can correspond to two widely separated points in the patch parameter space, and hence to widely separated locations in the texture pattern. Intermediate regions, which should somehow influence the intensity pattern, would be skipped over entirely. This is a special case of a phe-

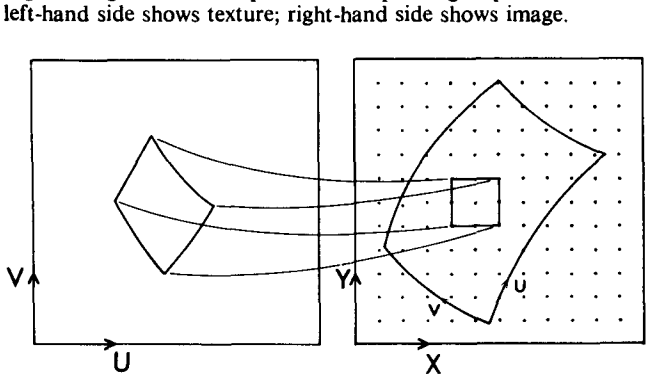
nomenon known as "aliasing" in the theory of digital signal processing. This theory [7] treats the image as a continuous signal which is sampled at intervals corresponding to the distance between picture elements. The well-known "sampling theorem" states that the sampled picture cannot represent spatial frequencies greater than 1 cycle/2 picture elements. "Aliasing" refers to the result of sampling a signal containing frequencies higher than this limit. The high spatial frequencies (as occur in fine detail or sharp edges) reappear under the alias of low spatial frequencies. This problem is most familiar as staircase edges or "jaggies." In the process of texture mapping, the aliasing can be extreme, owing to the potentially low sampling rate across the texture pattern.

To alleviate this problem we must filter out the high spatial frequency components of the image (in this case the texture pattern) before sampling. This filtering has the effect of applying a controlled blur to the pattern. This can be implemented by taking a weighted average of values in the pattern immediately surrounding the sampled point. Digital image processing theory provides a quantitative measure of the effectiveness of such weighting functions in terms of how well they attenuate high frequencies and leave low frequencies intact.

Catmull achieved the effect of filtering by maintaining an additional floating-point word for each picture element. This word contained the fraction of the picture element covered by patch fragments. For each new fragment added to the picture element, the texture pattern was sampled and the intensity was averaged proportionally to the amount of the picture element covered by the patch fragment. Examination of the spatial frequency filter effectively implemented by this technique shows that it is much better than point sampling but is not optimal.

The method discussed here does not require the extra storage and uses a better anti-aliasing filter. This filter is implemented by a weighting function originally used by Crow [3] to minimize aliasing at polygon edges ("jaggies"). It takes the form of a square pyramid with a base width of 2×2 picture elements. In the texture mapping case, the 2×2 region surrounding the given picture element is inverse mapped to the corresponding quadrilateral in the u,v parameter space (which is the same as the texture pattern space), see Figure 1. The values in the texture pattern within the quadrilateral are weighted by a pyramid distorted to fit the quadrilateral and summed.

Fig. 1. Region of texture pattern corresponding to picture element: left-hand side shows texture; right-hand side shows image.



The derivation of the quadrilateral on the texture pattern makes use of an approximation that the parametric lines within one picture element are linear and equally spaced. The X,Y position within a picture element can then be related to the u,v parameters on the patch by a simple affine transformation. This transformation is constructed from the u,v and X,Y values which are known exactly at the four corners of the patch fragment.

Given this algorithm, we now investigate the effects of various texture definition methods. We will use, as our sample object, a plain teapot constructed of 26 bicubic patches. First, the pattern may be some simple function of the u,v parametric values. A useful example of this is a simple gridwork of lines. The result is as though parametric lines of the component patches are painted on the surface, Figure 2. Note that the edges of the pattern lines show very little evidence of aliasing in the form of staircases. Second, the pattern may come from a digitized hand drawn picture, Figure 3. Third, the pattern may come from a scanned-in photograph of a real scene, as in Figure 4. Incidentally, this picture makes the individual patches very clear. This type of pattern definition enables the computer production of "anamorphic" pictures. These are pictures which are distorted in such a way that when viewed in a curved mirror the original picture is regenerated, Figure 5. The patch itself is defined so that the parametric lines are stretched in approximately the correct fashion and a real photograph is mapped onto the patch. Figure 5 should be viewed in a cylindrical mirror (e.g. a metal pen cap) with the axis perpendicular to the page. The fourth source of texture patterns shown here is Fourier synthesis. A two-dimensional frequency spectrum is specified and the inverse Fourier transform generates the texture pattern. This is a simple way of generating wavy or bumpy patterns. Certain restrictions on the form of the input spectrum must be followed to ensure that the pattern has an even distribution of intensities and is continuous across the boundaries. An example of this type of texture is shown in Figure 6. The texture patterns used here were generated before picture synthesis began and stored as 256×256 element pictures in an array in random access memory.

Reflection in Curved Surfaces

The second topic discussed in this paper concerns lighting models. Typically, visible surface algorithms

Fig. 2. Simple gridwork texture pattern: left-hand side shows texture pattern; right-hand side shows textured object.

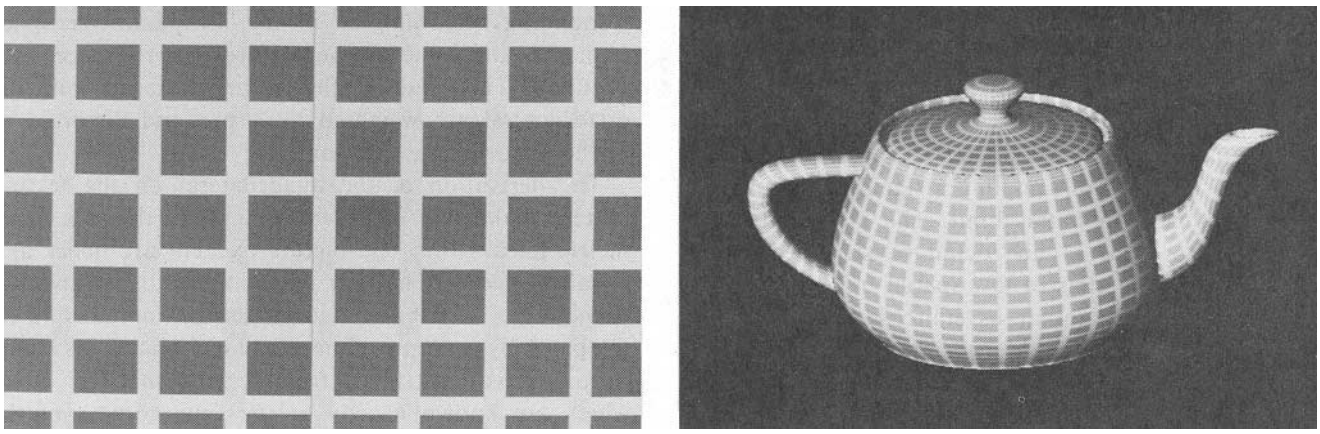
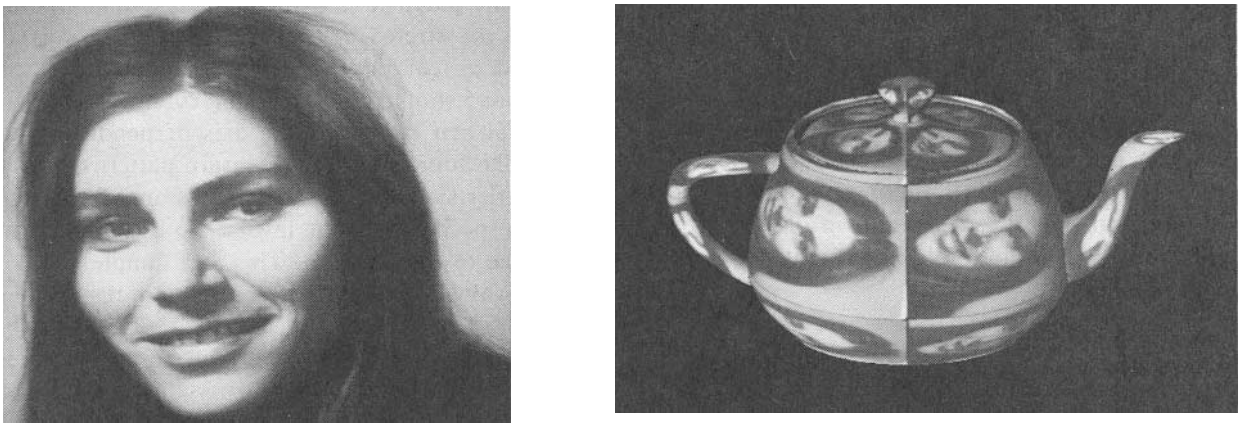


Fig. 3. Hand sketched texture pattern: left-hand side shows texture pattern; right-hand side shows textured object.

Fig. 4. Photographic texture pattern: left-hand side shows texture pattern; right-hand side shows textured object.



determine intensities within an image by using Lambert's (cosine) law: $i = s(L \cdot N)$, where i = intensity, s = surface shade, L = light direction vector, N = surface normal vector, and "$\cdot$" denotes vector inner product.

Variants on this function, such as

$i = s(L \cdot N)**n$, for $n > 1$

have been used to give the impression of shiny surfaces, but there is little physical justification for such functions, and the range of effects is limited. The modeling of more realistic lighting was first investigated by Bui-Tuong Phong [1]. His model of reflection incorporated a term which produced a highlight over portions of the surface where the normal falls midway between the light

source direction and the viewing direction. This is motivated by the fact that real surfaces tend to reflect more light in a direction which forms equal angles of incidence and reflectance with the surface normal. This can be easily implemented by simulating a virtual light source in a direction halfway between the light source and viewing directions, and raising the result to some high power to make the highlights more distinct:

$$i = s(L \cdot N) + g(L' \cdot N)^{**}n$$

where L' = virtual light source direction, g = glossiness of surface (0 to 1). Figure 7 shows an image generated using the above function with $n = 60$. These techniques work well for satin type surfaces but images of highly polished surfaces still lack realism. This is largely due to the absence of true reflections of surrounding objects and distributed light sources.

The simulation of reflections in curved surfaces requires an accurate model of the properties of the surface and access to accurate normal vectors at all points on the surface. The approximation of curved surfaces by collections of planar polygons is inadequate for this purpose, so extensions of the techniques of Gouraud [5] and Bui-Tuong Phong [1] hold little promise.

The subdivision algorithm, however, provides accurate information about surface position and can be made to give accurate surface normals at every picture element. This is the first algorithm that provides the appropriate information for the simulation of mirror reflections from curved surfaces. For each picture element, the vector from the object to the observer and the normal vector to the surface are combined to determine what part of the environment is reflected in that surface neighborhood. It can be shown that, for surface normal vector (Xn, Yn, Zn) and viewing position (1, 0, 0), the direction reflected, (Xr, Yr, Zr), is

$$Xr = 2*Xn*Zn, \quad Yr = 2*Yn*Zn, \quad Zr = 2*Zn*Zn - 1,$$

Having established the direction of the ray which is reflected to the eye, it remains to find what part of the environment generated that ray. For this, a model of the environment is needed which represents surrounding objects and light sources. Clearly, the view of the environment as seen from different points on the surface will vary. However, if it is assumed that the environment is composed of objects and light sources which are greatly distant from the object being drawn, and that occlusions of the environment by parts of the object itself are ignored, then the environment can be modeled as a two-dimensional projection surrounding the drawn object. Stated another way, the object is positioned at the center of a large sphere on the inside of which a picture of the environment has been painted. These simplifications allow the environment to be modeled as a two-dimensional intensity map indexed by the polar coordinate angles of the ray reflected

Fig. 5. Anamorphic image.

Fig. 6. Fourier synthesis of texture: top shows texture pattern; bottom, texture object.

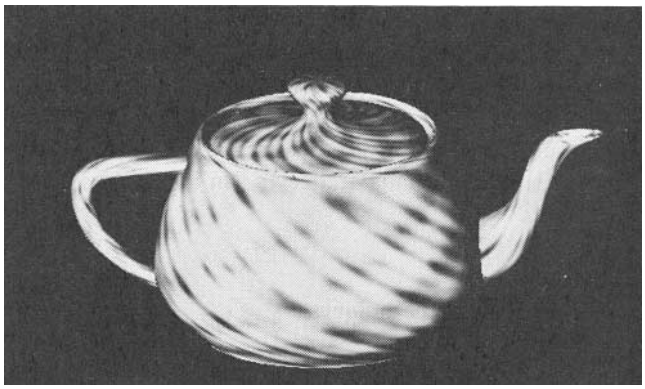


Fig. 7. Plain teapot with highlights.

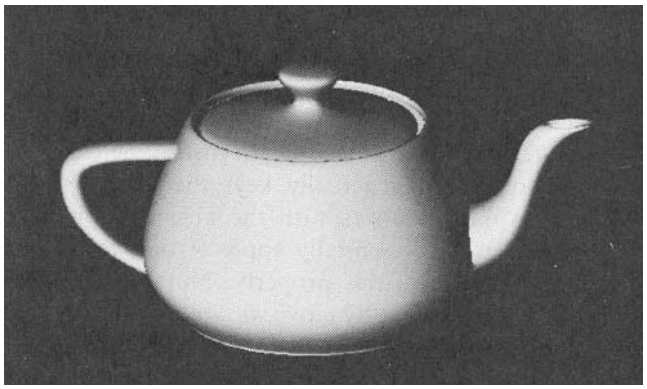
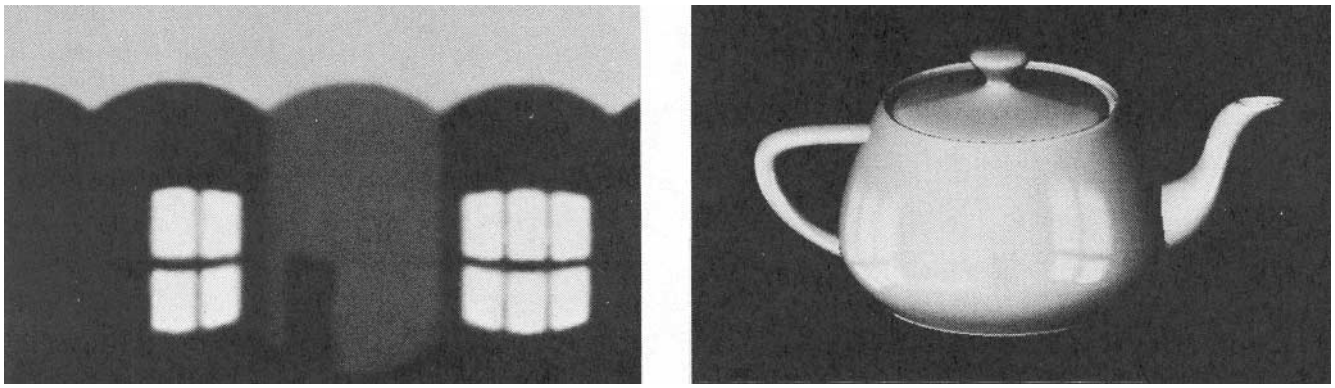


Fig. 8. Computer generated reflections: left-hand side shows environment map; right-hand side shows reflection in teapot.



(Xr, Yr, Zr). (Such maps will be shown with azimuthal angle plotted as abscissa and polar angle plotted as ordinate.)

When a reflection direction is computed, it is converted to polar coordinates and the reflected light intensity for that direction is read from the map. This is similar to the technique used to map texture onto a surface, except that the reflection direction, instead of parametric surface position, determines the coordinates in the map. Figure 8 shows an image generated using these techniques.

Use of the surface normal alone is tantamount to modeling the environment on an infinitely large sphere. This has the undesirable effect that the reflected intensity at all silhouette points on the object is the same, and corresponds to the intensity of the environment model diametrically opposite the eye. This deficiency can be corrected by using both the surface normal and surface position to determine what part of the environment map is reflected in a given surface fragment.

Combinations of Techniques

The techniques described above for simulating texture and reflection can be combined to produce images of objects having patterned shiny surfaces. When highlighting is combined with texture mapping, only the component from the real light source should be scaled. This models the highlight as being specularly reflected at the surface and not being affected by the pigment within it. In Figure 9 note how the highlights wash out the texture pattern underneath them. The technique for texture mapping actually keys the texture to the surface so that it moves with the object. Some other techniques, which essentially apply texture to the 2-D image, do not have this property. Note, in Figure 9, how the highlights hardly move with the teapot, whereas the texture does.

Given the texture mapping technique and the environment reflecting technique, we can combine them

Fig. 9. Textured object with highlights, two orientations.

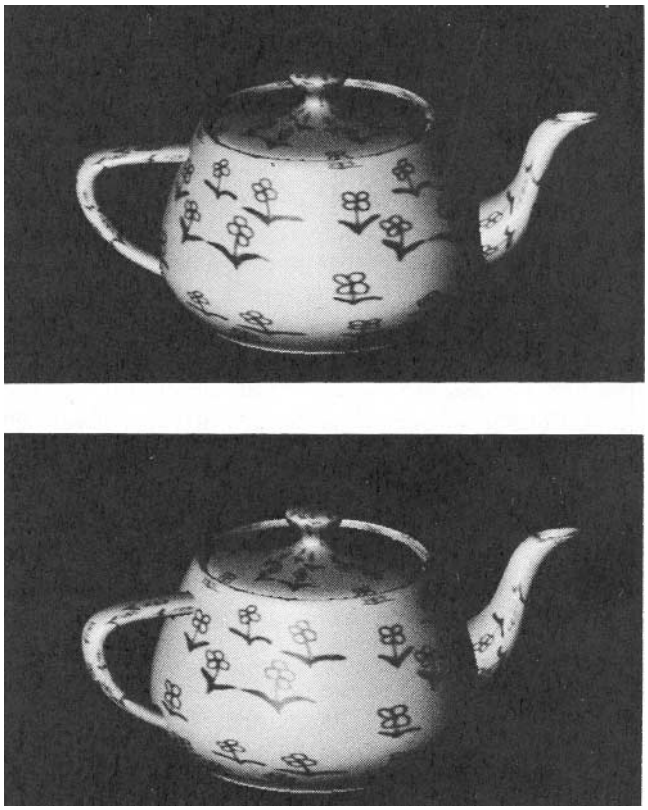
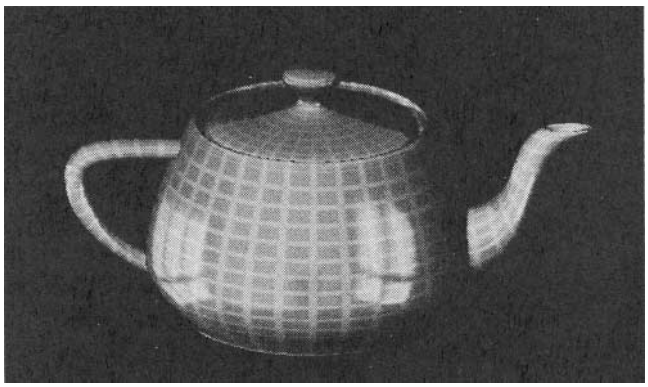


Fig. 10. Highly glazed patterned teapot.



to produce an image of a highly glazed patterned teapot, as in Figure 10.

Resource Requirements

The images shown in this paper were all generated on a PDP-11/45 computer having a 256K-byte random access frame buffer which was used as the depth buffer. The main routines were written in Fortran and the critical parts were written in assembly language. The computation time of the extended subdivision algorithm is roughly proportional to the area covered by visible objects. Images of nontextured objects of the type used in this paper take about 25 minutes. The addition of texture or reflection increases this time by about 10 percent. All images have a resolution of 512×512 picture elements.

Conclusions

By refining and extending Catmull's subdivision algorithm, images can be generated having a far higher degree of naturalness than was previously possible. These generalizations result in improved techniques for generating patterns and texture, and in the new capability for simulating reflections.

References

1. Bui-Tuong Phong. Illumination for computer generated images. *Comm. ACM 18*, 6 (June 1975), 311–317.
2. Catmull, E.A. Computer display of curved surfaces. Proc. Conf. on Comptr. Graphics, Pattern Recognition, and Data Structure, May 1975, pp. 11–17 (IEEE Cat. No. 75CH0981-1C).
3. Crow, F.C. The aliasing problem in computer-synthesized shaded images. Tech. Rep. UTEC-CSC-76-015, Dep. Comptr. Sci., U. of Utah, Salt Lake City, Utah, March 1976.
4. Forrest, A.R. On Coons and other methods for the representation of curved surfaces. *Computer Graphics and Image Processing 1* (1972), 341.
5. Gouraud, H. Computer display of curved surfaces. Tech. Rep. UTEC-CSC-71-113, Dep. Comptr. Sci., U. of Utah, Salt Lake City, Utah, June 1971.
6. Newell, M.E., Newell, R.G., and Sancha, T.L. A solution to the hidden surface problem. Proc. ACM 1972 Ann. Conf., Boston, pp. 443–450.
7. Oppenheim, A.V., and Schafer, R.W. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1975, pp. 26–34.
8. Sutherland, I.E., Sproull, R.F., and Schumaker, R.A. A characterization of ten hidden-surface algorithms. *Computing Surveys 6*, 1 (March 1974), 1–55.
9. Warnock, J.E. A hidden-line algorithm for halftone picture representation. Rep. TR 4–15, Dep. Comptr. Sci., U. of Utah, Salt Lake City, Utah, 1969.
10. Watkins, G.S. A real-time visible surface algorithm. Tech. Rep. UTEC-CSC-70-101, Dep. Comptr. Sci., U. of Utah, Salt Lake City, Utah, June 1970.

Graphics and
Image Processing

Hierarchical Geometric Models for Visible Surface Algorithms

James H. Clark
University of California at Santa Cruz

The geometric structure inherent in the definition of the shapes of three-dimensional objects and environments is used not just to define their relative motion and placement, but also to assist in solving many other problems of systems for producing pictures by computer. By using an extension of traditional structure information, or a geometric hierarchy, five significant improvements to current techniques are possible. First, the range of complexity of an environment is greatly increased while the visible complexity of any given scene is kept within a fixed upper limit. Second, a meaningful way is provided to vary the amount of detail presented in a scene. Third, "clipping" becomes a very fast logarithmic search for the resolvable parts of the environment within the field of view. Fourth, frame to frame coherence and clipping define a graphical "working set," or fraction of the total structure that should be present in primary store for immediate access by the visible surface algorithm. Finally, the geometric structure suggests a recursive descent, visible surface algorithm in which the computation time potentially grows linearly with the visible complexity of the scene.

Key Words and Phrases: visible surface algorithms, hidden surface algorithms, hierarchical data structures, geometric models

CR Categories: 5.31, 8.2

A version of this paper was presented at SIGGRAPH 76: The Third Annual Conference on Computer Graphics, Interactive Techniques, and Image Processing, The Wharton School, University of Pennsylvania, July 14–16, 1976.

Author's address: Information Sciences, University of California, Santa Cruz, CA 95064.