Fast Display of Well-Tesselated Surfaces

Christopher M. Brown
Computer Science Department
University of Rochester
Rochester, NY 14627

TR 23

March 1978

This research was supported in part by the Alfred P. Sloan Foundation under Grant No. 74-12-5, and in part by the Defense Advanced Research Projects Agency, monitored by the ONR under Contract No. N00014-75-C-1091.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR23	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Fast Display of Well-Tesselated Surfaces		5. TYPE OF REPORT & PERIOD COVERED technical report
		6. PERFORMING ORG. REPORT NUMBER 23
7. AUTHOR(s) Christopher M. Brown		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-1091
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department University of Rochester Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE March 1978
		13. NUMBER OF PAGES 35
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

three-dimensional computer graphics
raster graphics
hidden surface elimination
hidden line elimination
polyhedral objects
geodesic constructions

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Well-tesselated surfaces are piecewise planar functions on the sphere. Their planar faces are triangles which must meet a local geometric constraint. The faces may have arbitrary properties of transparency or reflectivity. These surfaces admit of simple hidden-line and -surface algorithms. In a raster graphics environment, the algorithms yield a priority ordering for painting entire faces. This order depends only upon the 3-D directions of face vertices as seen from the origin, and is independent of the radial position of vertices, hence of the function realized by the surface.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract

Well-tesselated surfaces are piecewise planar functions on the sphere. Their planar faces are triangles which must meet a local geometric constraint. The faces may have arbitrary properties of transparency or reflectivity. These surfaces admit of simple hidden-line and -surface algorithms. In a raster graphics environment, the algorithms yield a priority ordering for painting entire faces. This order depends only upon the 3-D directions of face vertices as seen from the origin, and is independent of the radial position of vertices. Hence the order is independent of the actual function realized by the surface. Since perspective distortion and an approximate version of object rotation may both be accomplished by changing only radial vertex positions, these useful transformations may be visualized without resorting.

Specific methods for creating well-tesselated surfaces are given, and general constraints defining them are stated. An efficient hidden surface/line algorithm is presented, with a simpler method for the case of opaque faced polyhedra. Proofs of correctness are provided.

Keywords and Phrases: three-dimensional computer graphics, raster graphics, hidden surface elimination, hidden line elimination, polyhedral objects, geodesic constructions.

CR Categories: 3.10, 3.41, 8.2.

1. Overview

Well-tesselated surfaces form a class which is large enough to be useful for many computer graphics applications and which has very simple hidden-surface and -line algorithms. These surfaces have planar faces, and are functions on the sphere determining at most one radial distance for any 3-D direction (i.e., longitude and latitude). Faces are triangles of arbitrary transparency or reflectivity; faces of other shape may be built up of adjacent coplanar triangles. The faces are not general triangles, but must meet a local geometric condition given below. Functions on the sphere can approximate many useful objects, such as a human head (except for most ears), a bottle, or an automobile fender. Some example surfaces appear in Figure 1; they were produced by the geodesic dome [6] construction detailed below.

<<INSERT FIGURE 1 HERE>>

Though well-tesselated surfaces have practical value, their chief point of intellectual interest is that they are the most complex surfaces which possess a particular very simple hidden-surface algorithm. The algorithm becomes even simpler when the surface is that of an opaque polyhedron. Further, applying the algorithm once solves the hidden-surface problem for all surfaces whose vertices have the same (longitude, latitude) directions. Thus one application of the algorithm suffices for an infinite number of piecewise planar spherical functions which share the same vertex directions.

One way to use the techniques given here is to express a surface for display as a function on the sphere, and pick a desired precision for the surface approximation. Apply the construction given below to obtain a polyhedron with adequate density of vertices, and distort the relevant part of its surface according to the desired function. Finally, apply necessary geometric transformations and reflectivity calculations, and display the result by one of the algorithms given here. The problem of constructing an exact well-tesselated version of a given piecewise-planar surface is not addressed here, but seems interesting, as does testing whether an arbitrary surface is well-tesselated.

Sutherland, Sproull, and Schumacker [9] have shown that hidden surface (line) algorithms have in common the operation of sorting. The two algorithms below (one a special case of the other) illustrate this observation particularly well; to display N faces the simpler algorithm just sorts N numbers. Being so simple, the algorithms themselves are of limited novelty; they are quite similar to subcases of the algorithm of Newell, Newell, and Sancha (NNS) [7]. What is more interesting is that such simple treatment suffices for a useful class of surfaces.

The algorithms (like that of NNS) are list-priority algorithms; they compute their results in object space to all available precision, and are limited in output accuracy only by the display medium. Like NNS, they depend on a raster graphics output device, in which an image memory raster is repetitively scanned out onto a viewing screen. Image information may be painted into the raster, thereby overwriting or combining with the previous contents; this capability allows for simulation of obscuration and transparency.

The algorithms of this paper differ from that of NNS in three related respects.

First, there is an underlying semantic difference. To get a tentative order for painting faces onto an image raster, NNS sort the faces on the maximum distance (depth) they attain from the viewpoint; nearer faces are to overpaint farther ones. The algorithms here sort faces instead on the maximum angle (at the origin) they attain with the viewing direction.

Secondly, therefore not only will the algorithms deal correctly with one well-tesselated surface, but the painting priority order they derive is independent of the choice of function used to produce the surface, once the vertex directions are chosen. It will be correct for any surface with the same vertex directions, including the special cases of surfaces arising from pure perspective distortion and simulated rigid rotation (see Section 3.)

Thirdly, both the algorithms given here are much simpler than that of NNS. The initial depth sort of NNS is inadequate for general objects, which may require faces to be divided and further tests (as many as six) to be applied before a correct priority order is obtained. In contrast, the opaque polyhedron algorithm presented here performs only the initial sort and needs no fixups; for general surfaces, ties in the initial sorted order must be resolved by one test. Face-dividing is never necessary.

The algorithms allow production of shaded surface visualizations if faces are painted according to some illumination and reflection model; more smoothly shaded representations may result from an interpolation scheme for face shading; faces of varying degrees of transparency may be defined. A bonus is that since faces are never divided, a hidden-line drawing may be produced by painting edges of faces black and their interiors white.

In Section 2 well-tesselated surfaces are defined and discussed. In Section 3 the algorithms are described and their timing is considered. Section 4 has constructions for well-tesselated surfaces. Section 5 contains conclusions and some open problems. The appendix has a more careful statement of tests and algorithms, and proofs of correctness.

2. Well - Tesselated Surfaces

Well-tesselated surfaces are subsets of the faces of well-tesselated polyhedra, which may be thought of as arising from polyhedra [4] inscribed in the unit sphere. These inscribed polyhedra come from certain tesselations of the sphere. The tesselations of interest are triangulated graphs embedded in the sphere, which produce a set of spherical triangles or patches partitioning its surface. Tesselations induce (spherically) inscribed polyhedra in an obvious way: the three vertices of each patch determine a plane face of the inscribed polyhedron. The inscribed polyhedron is well-tesselated if its tesselation meets the Angle Condition (AC): around any tesselation vertex on the sphere, the angle between two nonadjacent patch edges is not less than pi/2, and the angle between any adjacent patch edges is not greater than pi/2. Well-tesselated polyhedra may be derived from other well-tesselated (possibly inscribed) polyhedra by translating vertices radially with respect to the sphere center, or origin, according to some function of longitude and latitude; incident edges and faces are carried along. Finally, a well-tesselated surface is a subset of the faces of a well-tesselated polyhedron. Thus the tesselation defines a set of half-lines, or vertex directions, which radiate from the origin through each patch vertex (a longitude and latitude); the spherical function determines where the vertices lie along these vertex directions.

At least two special cases of spherical functions which do not change vertex directions are of interest. First, with the proper imaging model, the general perspective transformation is given by a radial vertex translation (Theorem IV of the appendix). Another useful application is one which provides an approximation to rotation. This can be done by considering the painted surface to be an approximation to an ideal object which it encloses. The enclosing surface has rubber faces, but its vertices can only move in and out along fixed radial vertex directions. As the object of interest rotates inside the non-rotating but deformable enclosing surface, it distorts the rubber faces into new (rotated) approximations via radial vertex translation. The accuracy of this technique depends on the characteristics of the object of interest and the tesselation; the constructions of Section 4 have good properties for this application. These two special cases are useful because they represent common geometrical transformations, and as is shown in Section 3, neither of them necessitates a resorting of surface faces.

Well-tesselated surfaces have simple hidden-surface algorithms because they are constructed to have high geometrical <u>coherence</u>. Coherence is loosely defined in [9] as a measure of predictability of what will appear next on the display (in the next picture element, on the next scan line, in the next movie frame, etc.) It arises from the geometry of the objects, from the close relation of successive movie frames, or from other physical causes. Exploitation of coherence leads to greatly improved running times in general hidden-surface algorithms, and to simple algorithms for the surfaces of interest here. Basically, a tesselation induces a set of infinite pyramids throughout 3-space (by radial projection of its patches) which are partially ordered in depth. The correctness of the algorithms depends heavily on the fact that surface faces are connected to their neighbors;

along with constraints on pyramids enforced by the AC, this guarantees that the pyramidal partial ordering will remain usable for surface faces as well. The function of the AC is to prevent certain overlaps between faces which render the simple algorithms inadequate. The effects of the locally-defined AC on the general appearance of tesselations is hard to characterize, but some facts are known. The degree of a tesselation vertex meeting the AC must be between 4 and 8; patch sizes may vary gradually over the sphere; patches with excessively small or large angles are unlikely. If a particular surface is not well tesselated, the algorithms may work anyway; however, violations of the AC allow the construction method above to produce surfaces for which the algorithms fail (see Section 3.) The interested reader is referred to the appendix for a more careful look at tesselation conditions and their relation to the algorithms.

3. Hidden - Surface Algorithms

The imaging model is that the viewpoint is on the positive z axis of a right-handed Cartesian coordinate system. The image plane is the z=0 plane, with x increasing to the right and y increasing upwards as seen from the viewpoint. The surfaces are functions of the sphere whose center is the origin. Images of surfaces are their projections onto the image plane through the viewpoint. The hidden surface algorithms determine a priority order for painting faces into an image raster representing the image plane. The general algorithm allows for transparent faces, and has as a special case the opaque polyhedron algorithm. The next two paragraphs give prose statements of the algorithms; more detailed and implementationally useful versions of both appear in the appendix.

The opaque polyhedron algorithm. Discard back faces (those whose surface normals point away from the viewpoint by more than pi/2 radians). Obtain the painting priority order for the faces by sorting them in order of decreasing maximum angle at the origin between any face point and the viewing direction (the z axis). (If the sphere is replaced by a globe, and if the viewpoint is imagined to be above the north pole, then the sort is on the minimum latitude of the spherical projection of a face.) Ties may be broken at random.

The general surface algorithm. Here, back faces may no longer be discarded. The same decreasing maximum angle sort is performed on faces, but now ties may not be broken at random. If two faces share an edge and are tied for maximum angle, then paint the one first which is farthest from the viewpoint in the sense of being on the far side of the plane formed by their common edge and the origin.

Figure 2 shows the display midway through painting one of the opaque polyhedra of Figure 1. Back faces have been discarded, leaving the central four-lobed hole, and face painting is proceeding from (angular) back to front.

<INSERT FIGURE 2 HERE>>

Note that the result of the sort in these algorithms is independent of the radial distance of vertices; thus the priority order is good for the infinite number of related surfaces sharing the same vertex directions. If it is performed on the spherically inscribed version of the surface, the sorting operation becomes the initial sort of the NNS algorithm, since the point of maximum face angle then corresponds to the point of maximum depth. This similarity is more accidental than basic; generally the painting order for surface faces yielded by NNS and by these algorithms will differ.

Figure 3 shows what can happen if the algorithms are applied to surfaces that are not well-tesselated; the image plane is the plane of the page, the viewing direction is normal to the page. The opaque polyhedron shown in Figure 3A is not well-tesselated. It arises from a tesselation based on lines of longitude and the equator of a globe, if the viewpoint is in the equatorial plane; the AC is violated at the poles. Figure 3B shows a polyhedron obtained by translating the vertex X radially (approximately toward the viewer's left eye); face P now occludes face V. However, P and V are tied at the north pole for maximum angle with the viewing direction. Since ties are broken at random in the opaque polyhedron algorithm, it may fail, incorrectly painting V after (over) P. Another case is illustrated in Figure 3C. Only two patches on the sphere are shown, but it is fairly clear that the AC must be violated somewhere if a tesselation includes them. On the sphere, lines of constant angle with the viewing direction are circles concentric with the sphere boundary; it is seen that patch (face) V has a smaller maximum angle, and will thus be painted later by either algorithm. Figure 3D shows particular faces arising from the patches; P has been projected farther than V, and is closer to the viewpoint everywhere. The later (over) painting of V is a mistake.

<<INSERT FIGURE 3 HERE>>

For piecewise planar surfaces which are functions on the sphere it is easy to tell whether a face presents its inner or outer side to the viewpoint (see the appendix.) Shading algorithms can thus easily cater to the inner/outer difference: steps may be taken to treat the interior and exterior alike, faces may have different reflectivities, transparencies, or colors on different sides, etc. Painting opaque faces with black edges and white interiors yields a hidden-line drawing.

The time complexity of the algorithms is theoretically dominated for large N (number of faces) by the sort, which is taken to be O(NlogN.) In practice, the running time has been small relative to geometric calculations taking linear time. On a minicomputer, using a high-level language with microcoded floating point, the shaded, ordered faces for each surface of Figure 1 (N is about 160) were produced in about four seconds, the hidden-surface algorithm taking about one second, the shading calculation about three seconds. The operations of creating surfaces, rotating them, and sending the faces to the display processor all take up to an order of magnitude longer. It is worth reiterating that changing the spherical function without changing the vertex directions does not change the priority order of faces. Since perspective distortion and the approximate rotation trick of Section 2 can be performed without changing vertex directions, both of these operations may be performed in linear time; this may be useful for dealing with large numbers of faces.

4. Two Constructions for Well - Tesselated Surfaces

Certain geodesic constructions [6] give well-tesselated inscribed polyhedra; they are discussed in more detail in [1]. The idea is to inscribe an icosahedron in the sphere, then to subdivide each icosahedral face into triangular subfaces whose vertices are projected onto the sphere. The icosahedron has 12 vertices, 20 faces, and 30 edges. If each edge is divided into n new edges, the polyhedra resulting from the geodesic constructions below have 10*n**2 + 2 vertices, 20*n**2 faces, and 30*n**2 edges.

Clinton [3] gives seven methods for subdividing the initial icosahedral faces. If a method existed which produced truly congruent facets, there would be many more than five platonic solids. To quantify the goodness of a method, one should quantify "congruent". The first method given here is straightforward; the second is an improvement in terms of most reasonable criteria of "congruency". The other five methods of Clinton are not as well-suited to subdividing individual faces, and yield no clear advantages over the second method given here. The simplest method is to divide each one into n**2 congruent equilateral triangles by dividing each edge into n equal lengths and connecting the division points by lines parallel to the edges of the face. During projection onto the sphere, the central subfaces of an icosahedral face are projected farther than those near an icosahedral vertex, and so produce larger inscribed faces; this symptom is more acute with finer subdivisions.

To produce more nearly congruent inscribed faces, the icosahedral edge subdivisions may be made to subtend equal angles on the sphere. After the division points on the edges have been thus located, they are connected by lines parallel to the edges of the icosahedral face. Since the division points are not equally spaced along the edges, the lines through them do not meet in points, but in small triangles. The centroid of each small triangle is projected to become a vertex. This method yields a substantial improvement in face uniformity. For four divisions per icosahedral edge, the ratio of largest to smallest solid angle subtended by inscribed faces is 1.517 for the first method, 1.146 for the second.

The following results are useful in icosahedron construction. Define:

t = the golden ratio = (1 + sqrt(5))/2

a = sqrt(t) /(5**(1/4))

b = 1/(sqrt(t) * (5**(1/4))).

The angle subtended by an icosahedral edge is arccos(sqrt(5)/5) radians. The twelve vertices may be placed (in Cartesian co-ordinates) at

(Ø, ±a, ±b)

(±b, Ø, ±a)

(±a, ±b, Ø).

5. Conclusions

Well-tesselated surfaces are piecewise planar functions on the sphere; they can be used to approximate continuous spherical functions. They have faces which are triangular (though adjacent coplanar triangular faces may may be considered as faces of other shape). The faces meet a local Angle Condition which constrains them from being general triangles. Faces may exhibit transparency. This class of functions may be used to represent many physical objects and surfaces, and is finding application as a visualization of heart volumes and histograms of 3-D vector data [2]. The hidden line and surface problem is easy for well-tesselated surfaces, since the constraints eliminate much of the computation necessary in general object-space or list-priority hidden surface algorithms. A geodesic dome construction yields well-tesselated surfaces.

A simple algorithm exists which produces hidden-surface or -line renditions of well-tesselated surfaces by establishing a priority painting order for entire faces. A simpler method may be used for opaque polyhedra. An unsophisticated minicomputer implementation of the opaque polyhedron algorithm takes a few seconds when applied to surfaces with a few hundred faces. (The computer is a Xerox ALTO, equivalent in this context to a Data General NOVA. The programs are written in BCPL. The actual display code incorporating the hidden surface algorithms is only a few pages long; interactive graphics utilities and object-creating routines are on the order of scores of pages). The priority order is independent of the spherical function realized by the surface, since it only depends on the 3-D directions of the surface vertices from the origin. This allows the perspective transform and an approximation to rotation to be applied to the surface without necessitating a resorting of faces. The algorithms here can provide a useful tool for a divide and conquer approach to painting a complex object or sets of surfaces if the surfaces can be expressed as well-tesselated surfaces partially ordered in depth (e.g. "separable" by a plane).

If well-tesselated surfaces are to be used to render given piecewise planar surfaces exactly, several interesting questions arise. How can it be determined whether a given surface is well-tesselated? Determining if all its faces are visible from some point is like the "convex kernel" problem [8], which may be solved by intersecting the interior volumes (see the appendix) of the N faces; this can be done in NlogN time. If the intersection is empty, the surface cannot be a function on the sphere. Each point in a non-empty intersection may be an origin point, and one must then determine if for any origin point the Angle Condition is satisfied for all vertices. How can a well-tesselated surface with some set of pre-assigned vertex locations and edges be produced? This is necessary to produce an exact well-tesselated cube, for instance. Can the ideas of Fuchs [5] be extended to produce an optimal well-tesselated approximation to a surface? Being well-tesselated is a very strict condition; it guarantees that the simple algorithms yield a painting order not just for one particular surface of interest, but for an infinity of related surfaces. The algorithms can clearly work correctly for an individual surface which does not exactly meet the strict condition, but this gets less likely as the surface becomes more nonconvex. What can be said about the behavior of the algorithms on surfaces which are not strictly well-tesselated? How much can the Angle Condition be relaxed as the surface exhibits milder nonconvexity?

<u>Acknowledgements</u>. Many colleagues have assisted these efforts. Deserving special mention are K. Sloan, who first stated the AC and whose constructive criticism has been invaluable, P. Selfridge for graphics production, A. Requicha, J. Nabielsky and R. Sproull for critical reading.

References

1. C.M. Brown, Representing the orientation of dendritic fields with geodesic tesselations. TR13, Department of Computer Science, University of Rochester, Rochester N.Y., (1977).

2. C.M. Brown, Two descriptions and a two-sample test for 3-D vector data. TR49, Department of Computer Science, University of Rochester, Rochester, N.Y., (1979).

3. J.D. Clinton, Advanced structural geometry studies part I: polyhedral subdivision concepts for structural applications. NASA CR-1734/35, (September 1971).

4. H.S.M. Coxeter, Regular Polytypes, 3rd ed. Dover Publications, New York, (1973).

5. H. Fuchs, Z.M. Kedem, and S.P. Uselton, Optimal surface reconstruction from planar contours. Comm. ACM 20, No. 10, 703-712, (October 1977).

6. R.B. Fuller, Building construction. U.S. Patent No. 2,682,235, (1951).

7. M.E. Newell, R.G. Newell, and T.L. Sancha, A new approach to the shaded picture problem. Proceedings of ACM

National Conference, (1972).

8. F.P. Preparata and D.E. Muller, Finding the intersection of a set of n half-spaces in time Ø(n log n). University of Illinois Report R-8Ø3 (ACT-7) (December 1977).

9. I.E. Sutherland, R.F. Sproull, and R.A. Schumacker, A Characterization of ten hidden-surface algorithms. Computing Surveys 6, No. 1, 1-55, (March 1974).

Appendix

1. Preliminaries

This appendix is relatively self-contained, but for brevity it relies on the main body of the paper for some basic terms and concepts. It contains explicit geometrical tests alluded to in the text, Angle and Edge Conditions for tesselations, detailed statement of the algorithms, and proofs that the algorithms are correct. It relies heavily on the technique of reasoning about sets of surface faces by means of reasoning about the infinite pyramids (radial projections of patches) which contain them, and about pyramids by means of the patches which produce them. This gives rise to locutions such as "painting a patch," meaning "painting any face whose spherical projection is a patch." A *face* is the face of a particular surface. A *patch* implicitly references infinitely many faces. To avoid confusion, the vertex of a patch is hereafter called a *pvertex* and the edge of a patch a *pedge*: *edge* and *vertex* refer to surface edges and vertices. Half-lines from the origin through pvertices are called *vertex directions*, since vertices must lie in them. Each patch or face P has three edge-neighbor patches, or *e-nbrs* (which share a (p)edge with P) and a number of vertex-neighbor patches, or *v-nbrs* (which share exactly one (p)vertex with P.) A non-neighbor patch, or *non-nbr*, shares no (p)vertices.

Let a spherical coordinate system (rho, theta, phi) have a common origin with the Cartesian coordinates. Phi is the polar angle, varying from 0 at the positive z axis to pi at the negative z axis. Points in the image plane have phi = pi/2. The angle theta varies from 0 to 2*pi counterclockwise from the positive x axis as seen from the viewpoint. Points in space are Cartesian 3-vectors $\underline{x}$=(x,y,z), or 3-vectors $\underline{p}$=(rho,theta,phi). pvertices are spherical 3-vectors (1,theta,phi). The functions theta(.) and phi(.) give the theta and phi coordinates of a point. A <u>line of sight</u> is a half-line emanating from the viewpoint passing through the image plane. The z axis and any other line of sight determine a plane which cuts the sphere in a line of constant theta, and thus is perpendicular to lines of constant phi. This fact is important in the sequel.

2. Well - Tesselated Surfaces

Theorem I:

If the AC is met, then if d is the degree of the tesselation graph (maximum number of patches around a pvertex), 4=<d=<8. □

The Edge Condition is in terms of face edge lengths, not angles; it gives a different insight into the constraints on well-tesselated surfaces.

Definition: The Edge Condition (EC) for Tesselations

All pedges are shorter than one radian, and the ratio of longest to shortest pedges of any patch is less than 1.27.

Theorem II:

The Edge Condition implies the Angle Condition.

Proof:

By spherical trigonometry. □

3. Algorithms and Tests

Definition: The Interior Test

Consider the plane of a face to divide 3-Space into two open half-spaces. If one of them contains the origin, it is called the interior volume of the face and the other is the exterior volume. A face meets (fails) the Interior Test iff the viewpoint is in the interior (exterior) volume of a face. The test status of a face whose plane includes the viewpoint is arbitrary (but should be consistent.)

Definition: Invisible and Potentially Visible Faces

For a surface which is a function on the sphere:

Iff a face F meets the Interior Test:

In an opaque polyhedron, opaque faces intervene between F and the viewpoint, so F is invisible (a back face).

In a general surface, the interior side of the face is potentially visible (i.e. visible except for obscurations.)

Iff a face fails the Interior Test:

In an opaque polyhedron, F is potentially visible.

In a general surface, the exterior side of F is potentially visible.

The function maxphi(.) gives the maximum phi value attained by a face or patch; only the (p)vertices need be tested to determine it.

Algorithm PaintOpaquePolyhedron

Input: The (opaque) face set OpaqueFaces of a well-tesselated polyhedron.

1. Discard a face F from OpaqueFaces if F meets the Interior Test.
2. Associate maxphi(F) with each F remaining in OpaqueFaces.
3. Sort OpaqueFaces into descending order of associated maxphi(F), breaking ties at random.
4. Paint the members of OpaqueFaces in sorted order.

Algorithm PaintGeneralSurface

Input: The set Faces of faces of a well-tesselated surface.

1. For each face F in Faces, associate maxphi(F) with F.
2. Form an ordered list L of sets of faces. Each set in L has as members all faces with a particular value of maximum phi, and the list is in order of descending common maximum phi value.

3. In order, submit the sets in L to Subroutine PaintAPhi.

PaintAPhi uses the Priority Test defined below to break ties in the initial sort. Stated for patches, it holds for faces. A given patch can have at most two e-nbrs of identica maximum phi if the AC holds. A posterior e-nbr should always be painted before its prior e-nbr, since the prior patch (hence its entire pyramid) is between the viewpoint and the posterior patch (pyramid) for every line of sight passing through both. This fact is useful in the proof of Lemma VI. The classifications "prior", "posterior", and "isolated" are given to patches on the basis of their local relationships with e-nbr patches. Thus the binary predicates prior and posterior induce unary classifications, which hold for all patches of equal minimum phi. This is shown by Lemma V, which also shows that PaintAPhi canot produce contradictory patch classifications.

Definition: The Priority Test

The origin and the two shared pvertices of e-nbrs determine a plane dividing 3-Space into two open half-spaces. If the viewpoint is (is not) in the same half-space as one of the patches, that patch is the <u>prior</u> (<u>posterior</u>) patch. If the viewpoint is in the plane, the test gives no information.

<u>Subroutine PaintAPhi</u>

Input: a set IdentPhi of faces with identical maximum phi.

1. Associate a classification with each face in IdentPhi, initially "unclassified."
2. Repeat this step until no unclassified faces remain in IdentPhi: let F be an unclassified face.
   a. If F has no e-nbrs in IdentPhi, classify it "isolated"
   b. else if F has an e-nbr in IdentPhi classified "prior" ("posterior"), similarly classify the other e-nbr (if any); classify F the opposite, viz. "posterior" ("prior")
   c. else if the Priority Test can determine that F is prior (posterior) for one of its e-nbrs in IdentPhi, so classify F and classify its e-nbr(s) in IdentPhi the opposite, viz. "posterior" ("prior")
   d. else classify F "isolated."
3. Paint faces:
   a. Paint posterior faces in any order.
   b. Paint prior faces in any order.
   c. Paint isolated faces in any order (this step can be done any time relative to steps 3a and 3b).

4. Correctness of the Algorithms

Both algorithms paint all potentially visible faces. The proofs involve demonstrating that no face is wrongly overpainted by another face. The strategy will be to classify a patch V as *suspicious* with respect to an already painted patch P if (roughly) V overlaps P, is partly behind P, and has no point of greater phi than P ("behind" and "overlap" are defined directly below). These criteria have the flavor of traditional hidden-surface tests on faces. If V is not suspicious it cannot possibly affect P when V is painted by either algorithm. A suspicious patch V is *dangerous* to P if it (i.e. any of its faces) can actually wrongly overpaint P (i.e. any of P's faces) in the course of a particular algorithm. By definition, if there are no dangerous patches for an algorithm, the algorithm is correct. It will be shown that the AC prevents many suspicious patches from existing, and that those that remain are not dangerous.

Definitions: Behind and Overlap

A patch V is *behind* P (in an *overlap* interval) if: there exists a closed overlap interval [theta 0, theta1], theta0<theta1, such that given any theta, theta0 ≤ theta ≤ theta1, then for all points p in P and v in V with theta(p) = theta(v) = theta, phi(v) >= phi(p).

If V is not behind P in an overlap interval, then P is behind V (or V is in front of P).

If patch V is behind patch P, then along any line of sight in the overlap, the pyramid of patch P is between that of patch V and the viewpoint. Thus if a face of one of these patches obscures a face of the other, it must be the face of P which does the obscuring, and hence which should be painted later than the patch of V.

Definition: Suspicious

A patch V is suspicious with respect to a patch P if both:

1. maxphi(V) =< maxphi(P)
2. V is behind P in some overlap interval.

Figure 3C shows a particular case of a patch V which is both suspicious and dangerous with respect to P.

Theorem III:

If a patch V is not suspicious with respect to a patch P, it is not dangerous to P.

Proof:

If maxphi(V) > maxphi(P), both algorithms paint V before P, so V cannot overpaint P. If there is no overlap in theta, V cannot overpaint P by Fact I. If there is overlap but V is not behind P in the overlap, then P is behind V. Thus if V overpaints P it does so correctly, and so V is not dangerous to P. □

The results below are independent of the position of the viewpoint on the z-axis. Although it is really a corollary of that observation, theorem IV explicitly shows that general perspective distortion does not necessitate a resorting of the faces.

Theorem IV.
General perspective distortion leaves vertex directions invariant.
Proof:
The general perspective distortion operating on a point $\underline{x}$ may be expressed for some focal length constants f,g, and h as

$$\underline{x}' = (1/(1 + z/f + y/g + x/h))\underline{x}.$$

Thus the magnitude of $\underline{x}$ is changed by perspective, but not its direction. □

4.1. Correctness of PaintOpaquePolyhedron

Lemma I:
An e-nbr E of a patch P is not dangerous to P in PaintOpaquePolyhedron.
Proof:
If E is invisible it is a back face not painted by PaintOpaquePolyhedron, hence it cannot be dangerous. If E is potentially visible, the edge common to E and P must separate the

images of P and E on the image plane, so E cannot overpaint P. □

Lemma II:

In a well-tesselated surface, no v-nbr V of a patch P is suspicious with respect to P.

Proof:

« Figure 4 »

Case I: the pvertex p1 of maximum phi is the same for both P and V. In Figure 4A, the AC guarantees the angle beta >= pi/2, so any overlap can be at most the singleton set theta. If phi(p2) < phi(p1), alpha must be >pi/2. If there were more than one patch "in between" V and P (as E is), then the AC forces phi(p1) < phi(p2) (V is not suspicious).

Case II: the pvertex p1 of maximum phi for P is different from p2, the pvertex of maximum phi for V. Alpha, the angle subtended by E in Figure 4B, must be <= pi/2 by the AC, so any overlap can be at most the singleton set theta. □

Lemma III:

In a well-tesselated surface, no non-nbr Q of a patch P is suspicious with respect to P.

Proof:

Suppose Q, a non-nbr of P, were suspicious with respect to P

(Figure 5).

« Figure 5 »

At least one e-nbr E of Q is in front of Q in the overlap of P and Q. E is suspicious with respect to P, since maxphi(E) =< maxphi(Q). If E is a v-nbr of P, then by Lemma II a contradiction results. E cannot be an e-nbr of P, since Q would then have been a v-nbr of P contrary to assumption (the e-nbr of an e-nbr of P is P or a v-nbr of P.) Rename E to Q and find a new (suspicious) E in front of the new Q in the overlap. Such renaming and E-finding can only be done a finite number of times before the current E becomes an e-nbr or a v-nbr of P. If E becomes an e-nbr, then the current Q is a suspicious v-nbr, a contradiction by Lemma II. If it becomes a v-nbr, a similar contradiction arises, since E is suspicious itself. □

Theorem V:

PaintOpaquePolyhedron is correct; in applying it to a well-tesselated polyhedron, no patch Q is dangerous with respect to a patch P.

Proof:

If Q is an e-nbr of P, it is not dangerous to P by Lemma I and Theorem III. If Q is a v-nbr or non-nbr of P, then by Lemmas II and III it is not even suspicious; by Theorem III it cannot be dangerous to P. □

4.2. Correctness of PaintGeneralSurface

Lemma IV:

The AC implies that if maxphi(E) > maxphi(V) with E and V e-nbrs, then E is a posterior e-nbr.

Proof:

This consequence of the AC is proved similarly to Case II of Lemma II. In Figure 4B, if phi(p1) > phi(p2) and E is prior to V, alpha must be greater than pi/2. □

Lemma V:

The AC implies that a prior(posterior) e-nbr is prior(posterior) for all e-nbrs of equal maximum phi.

Proof:

This important consequence of the AC is proved similarly to Case I of Lemma II. In Figure 4A, if E were prior to V and posterior to P, beta would be less than pi/2. □

Lemma VI:

In a well-tesselated surface, no e-nbr E is dangerous to a patch P in PaintGeneralSurface.

Proof:

Case I: maxphi(E) /= maxphi(P):

by Lemma IV, the patch of larger maximum phi is a posterior e-nbr; it is correctly (by the comment on the Priority Test) painted first by PaintGeneralSurface.

Case II. maxphi(E) = maxphi(P):

by Lemma V, no patch in any input set for PaintAPhi is both prior and posterior to its e-nbrs. Thus the prior and posterior patches may be classified by local examination and painted in correct relative order, as PaintAPhi does. Isolated patches overlap with other patches in the input set in at most a single value of theta, and can be painted independently. □

Theorem VI:

PaintGeneralSurface is correct; in applying it to a well-tesselated surface, no patch Q is dangerous to a patch P.

Proof:

As for Theorem V, using Lemma VI instead of I. □

FIGURE CAPTIONS

Figure 1. A sampling of well-tesselated surfaces, shown as opaque polyhedra shaded by basic techniques.

Figure 2. The opaque polyhedron algorithm in process.

Figure 3. The inscribed polyhedron in (A) is not well-tesselated; (B) shows a potential face misordering. The patches in (C) also cause a potential misordering (D).

Figure 4. Patches on the sphere: theta is a line of sight (constant theta), phi a line of constant phi. V is a v-nbr of P, E is an e-nbr of P.

Figure 5. Patches on the sphere: Q is a non-nbr of P. E is an e-nbr of Q, V1 and V2 are v-nbrs of Q, the shaded band is the overlap of P and Q. This situation is impossible if the AC is met.

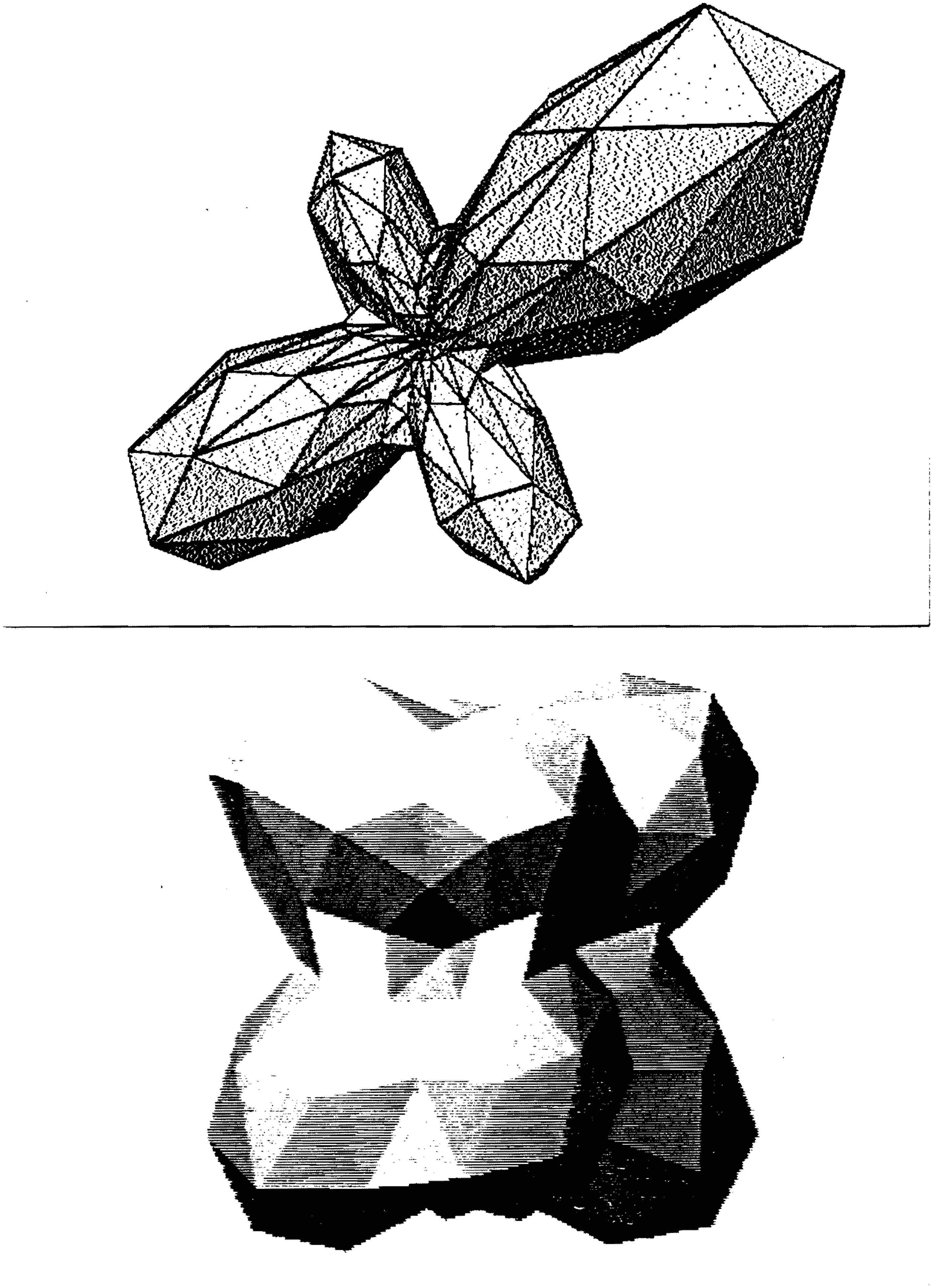
Figure 1

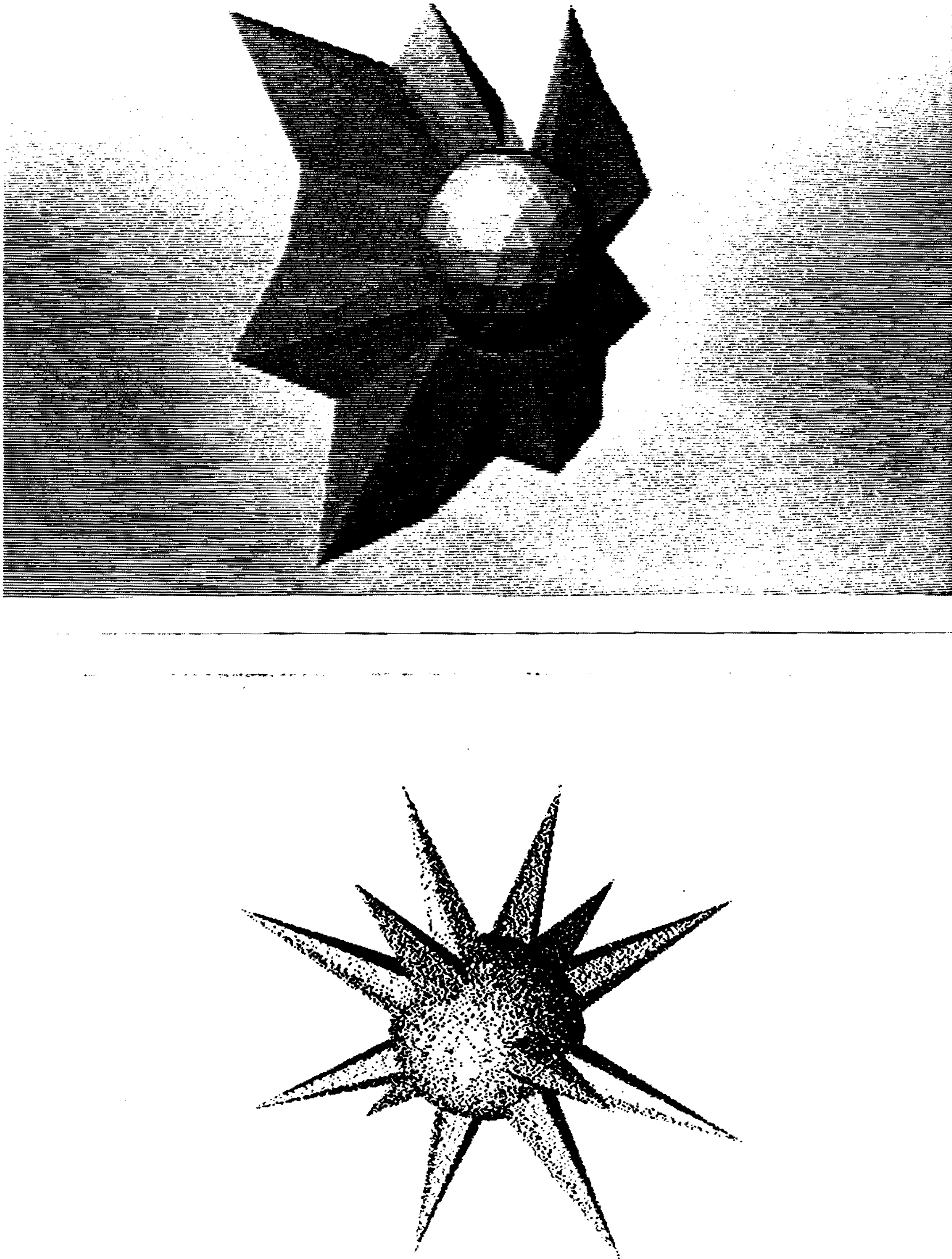

Figure 1 (Continued)

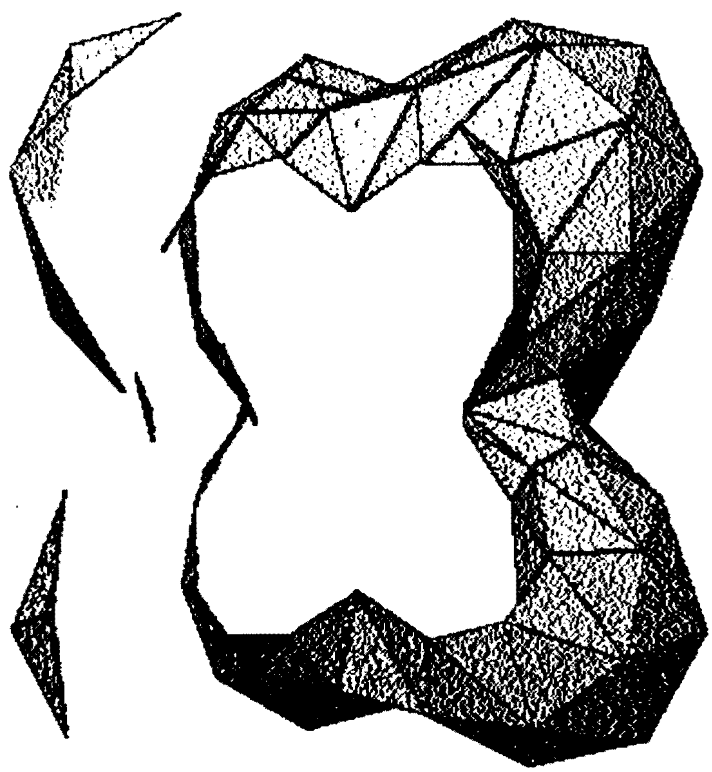

Figure 2

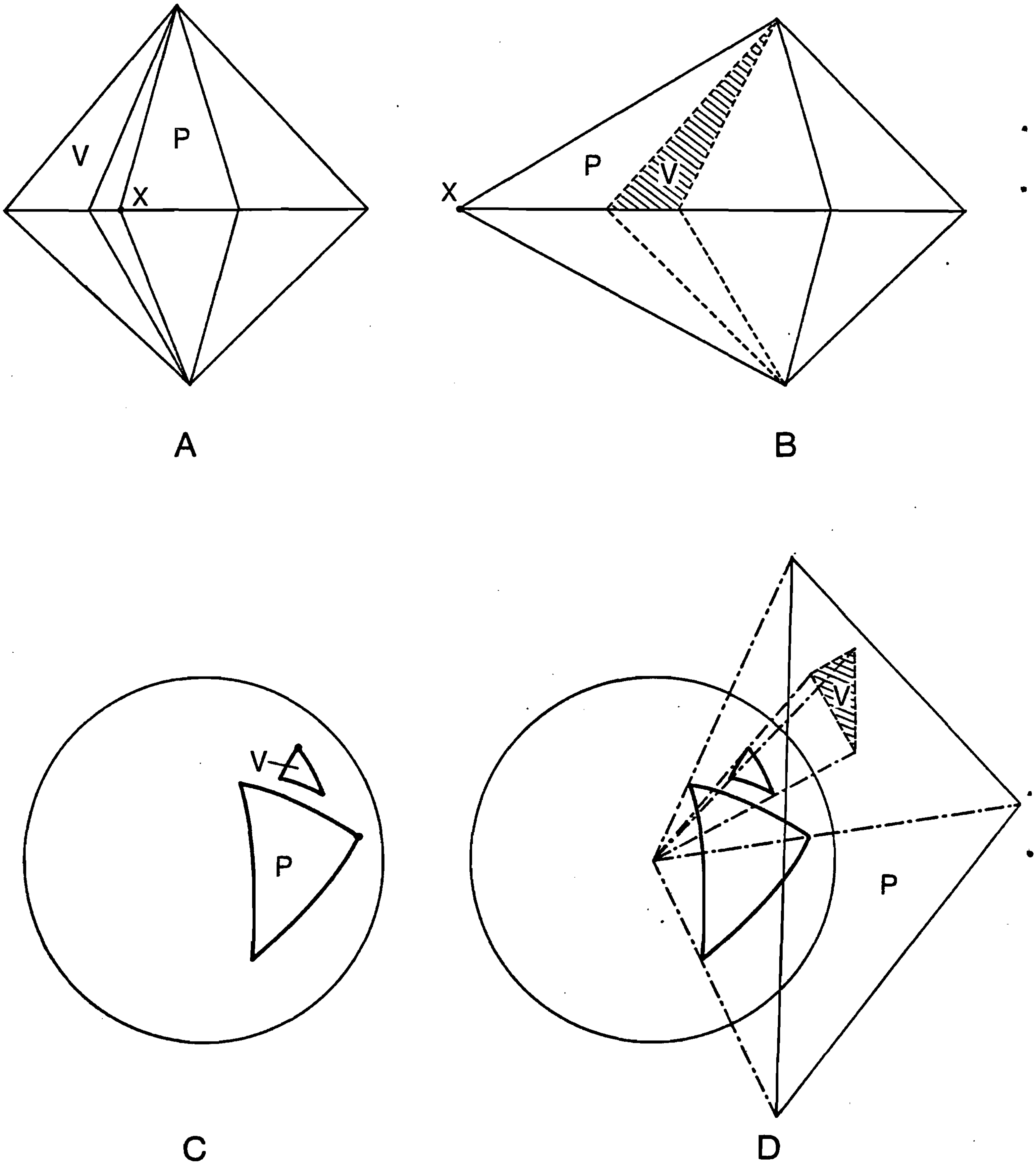

Figure 3.

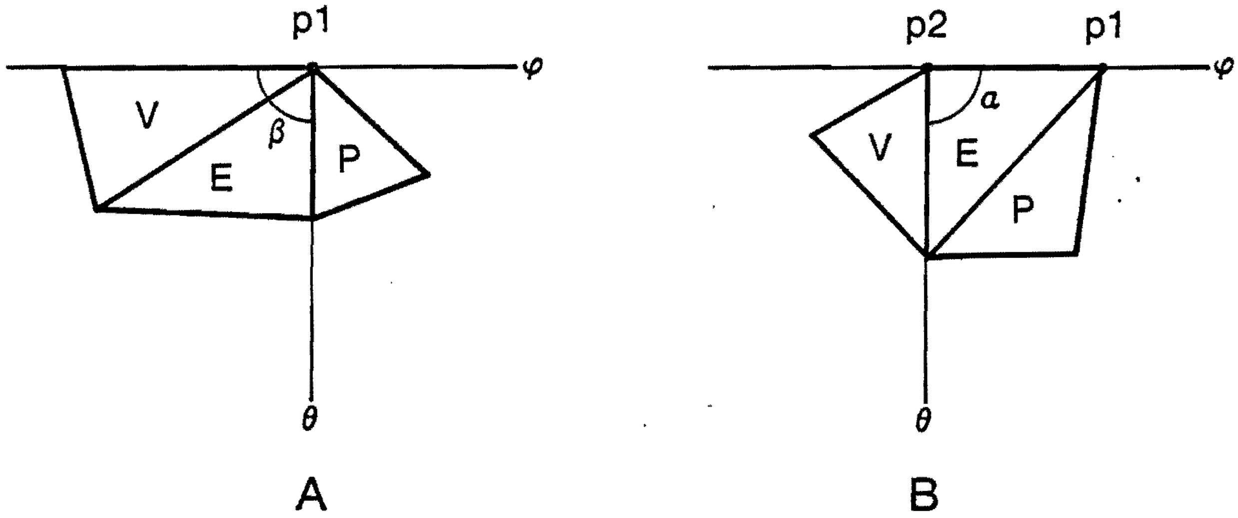

Figure 4

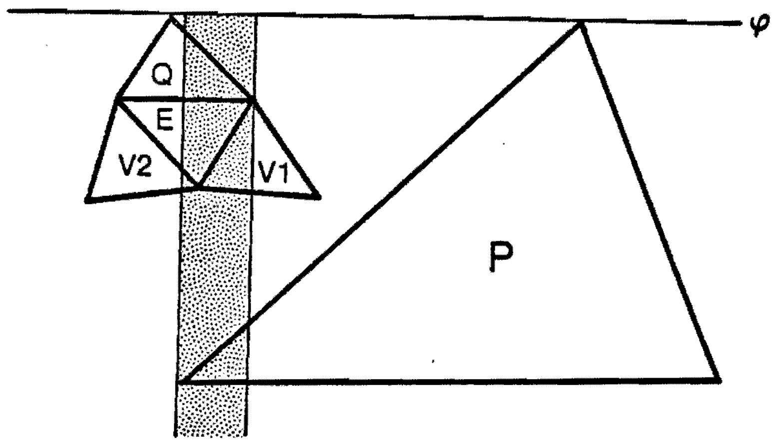

Figure 5