NASA
Reference
Publication
1085

March 1982

A General Solution to the Hidden-Line Problem

David R. Hedgley, Jr.

NASA

NASA
Reference
Publication
1085

1982

A General Solution to the Hidden-Line Problem

David R. Hedgley, Jr.
Ames Research Center
Dryden Flight Research Facility
Edwards, California

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

A GENERAL SOLUTION TO THE HIDDEN-LINE PROBLEM

David R. Hedgley, Jr.
Ames Research Center
Dryden Flight Research Facility

INTRODUCTION

The requirement for computer-generated perspective projections of three-dimensional objects by way of line drawings has escalated in recent years. Pictures of objects which show visible and hidden lines are relatively easy to present. Unfortunately, such renderings are often ambiguous and serve as little value to the engineer or scientist. (See fig. 1.)

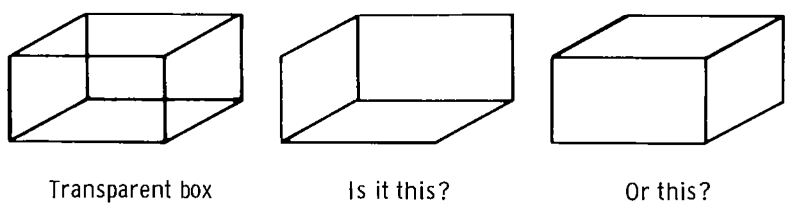


Figure 1. Ambiguous case.

Historically, much literature has appeared addressing this problem (ref. 1). However, prior solutions have exhibited some inherent limitations, one of the most significant being square-law growth, that is, the tendency of the computer execution time to grow as the square of the number of elements. Another significant restriction is the environmental limitations. (See appendix A for a list of typical limitations.)

At NASA Ames Research Center's Dryden Flight Research Facility (DFRF), the need arose to graphically represent aerodynamic stability derivatives as a function of two variables. (See the stability derivative plot in appendix B.) This requirement and the potential for further application served as a motivation for developing a general solution to the hidden-line problem which avoids the undesirable features of prior solutions.

This paper lays the theoretical foundation for the practical implementation of a general hidden-line algorithm. A theorem is presented and proved which does not assume any environmental limitations, unlike prior approaches. Furthermore the theorem allows the determination of the visibility of an entire line segment by choosing only a few points on that line, thus providing a basis for a rapid algorithm.

The author wishes to express his appreciation to Mary Bailer, Victor Pestone, and Paul Redin for their many helpful suggestions and constructive criticisms.

ANALYSIS

This section introduces basic definitions of symbols and terms, develops the equation of a plane, and discusses both the visibility of an arbitrary point and how the points are selected. The criteria of these selections and the sufficiency of the choices are predicated on a theorem which is proved.

Definitions

Let O be any scene or collection of objects which can be represented with straight lines and/or n-sided convex/concave planar polygons (internal boundaries allowed). Also, let each polygon be defined in such a way that every point is a boundary point in the topological sense. For economy of definition, vertices of polygons are sufficient and, similarly, end points for line segments.

With the admissible elements enunciated in the above paragraph, we need only discuss the visibility of a point P and its selection. The argument is then easily extended to the entire scene O.

The Equation of a Plane

The equation of a plane is given by

$$Ax + By + Cz + D = O,$$

where

$$A = b_1c_2 - b_2c_1$$
$$B = c_1a_2 - c_2a_1$$
$$C = a_1b_2 - a_2b_1$$
$$D = -(Ax_i + By_i + Cz_i)$$

where

$$a_i = x_{i+1} - x_i$$
$$b_i = y_{i+1} - y_i \qquad (i = 1,2)$$
$$c_i = z_{i+1} - z_i$$

x_j, y_j, z_j ($j = 1,3$) are three non-collinear points. A, B, and C are the coefficients of the normal vector to the plane and hence the coefficients of the plane itself.

For consistency with plotter hardware convention, x and y have their standard directions and z is perpendicular to the plane of the paper. The direction is consistent with a right-handed coordinate system.

Visibility Criteria

CRITERION I. *Let C be the projection of a planar closed polygon A on the XY-plane such that every point on C is a boundary point. (See fig. 2.) Let $P(x, y)$ be the projection of any point (x, y, z). If P lies on the boundary of C, it is clearly visible with respect to C. And hence (x, y, z) is visible with respect to A. If P lies outside the boundary of C, then a line drawn from P in any direction to infinity will intersect the boundary an even number of times. If P lies inside the boundary of C, then this semi-infinite line drawn from P will intersect the boundary of C an odd number of times. (See ref. 2.)*

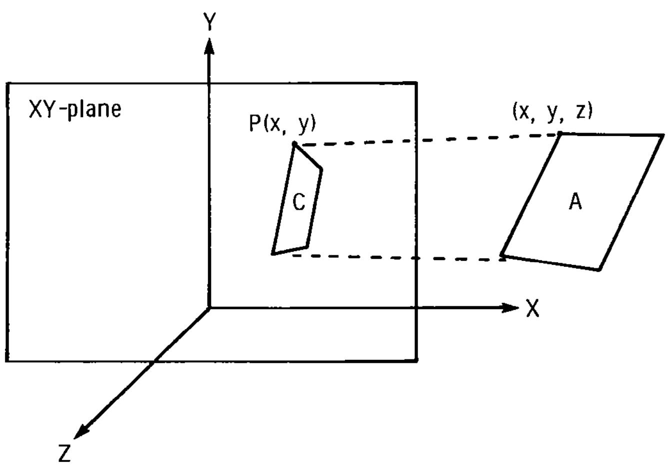


Figure 2. Plane A and its projection, C; point (x, y, z) and its projection, $P(x, y)$.

Figure 3 illustrates this idea. Note that line ℓ_1, drawn from P_1 in the XY-plane in an arbitrary direction to infinity, intersects the edges of the holes and containing

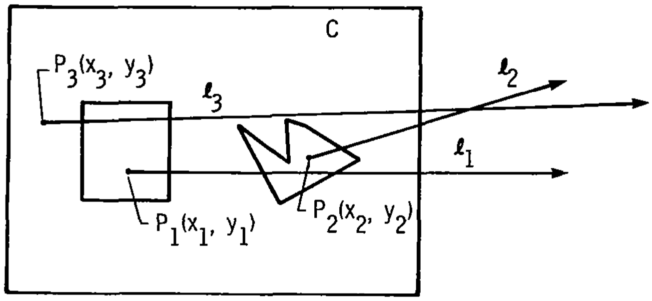


Figure 3. Polygon C with two holes.

plane, C, four times. Thus, the count is even indicating that P_1 is "outside" C and hence (x_1, y_1, z_1) is visible with respect to plane A. This is also true for (x_2, y_2, z_2). However, for $P_3(x_3, y_3)$, the count is odd indicating that the point lies in the interior of C.

If this semi-infinite line should cross a vertex, a line with a different slope should be selected. It is always possible to find such a line since the number of combinations of vertices is finite, whereas the number of slopes is infinite. Moreover, if C should have an arbitrary number of "holes" whose boundaries have the character of the external boundary of C, then no generality is lost. For if P lies inside C and outside all "holes," then the count is odd with respect to C as a polygon and even with respect to all holes. Therefore the total count remains odd. The remaining cases are argued similarly.

With the preceding observations in combination with the equation of a plane, it is now possible to determine the visibility of (x, y, z) with respect to plane A if P is "inside" C.

CRITERION II. *$z \geqslant -(A_o x + B_o y + D_o)/C_o$ implies (x, y, z) is visible with respect to A.*

CRITERION III. *$z < -(A_o x + B_o y + D_o)/C_o$ implies (x, y, z) is invisible with respect to A.*

Whether or not P is inside, outside, or on the boundary of C can be determined from Criterion I. Clearly then, if (x, y, z) is visible with respect to all A_i, it is visible. Note that if A is a line or if $C_o = 0$, then Criteria II and III do not apply. In this case, the projection is a line and hence every point is visible with respect to it.

Point Selection Criterion

Having a method for visibility determination is not by itself sufficient for a practical implementation. That is, hidden-line algorithms typically take huge amounts of computer time for even moderately simple scenes. The worst case is brute force where every line is quantified into many points and each point is tested. Hence, economy of points and efficiency of implementation are very important. The following discussion presents an ordered approach to improve computational efficiency. We first consider this important definition.

DEFINITION. Let S be a sequence of distinct triplets (x_i, y_i, z_i) belonging to a line determined by (x_1, y_1, z_1) and (x_2, y_2, z_2). Define S to be "ordered" if given any two triplets, say (x_k, y_k, z_k) and (x_n, y_n, z_n). Then $n > k$ implies that $(x_n - x_1)^2 + (y_n - y_1)^2 \geqslant (x_k - x_1)^2 + (y_k - y_1)^2$. Define a primed symbol, such as ℓ', to mean the projection of that element, ℓ, onto the XY-plane. (Fig. 4 represents a typical sequence S.)

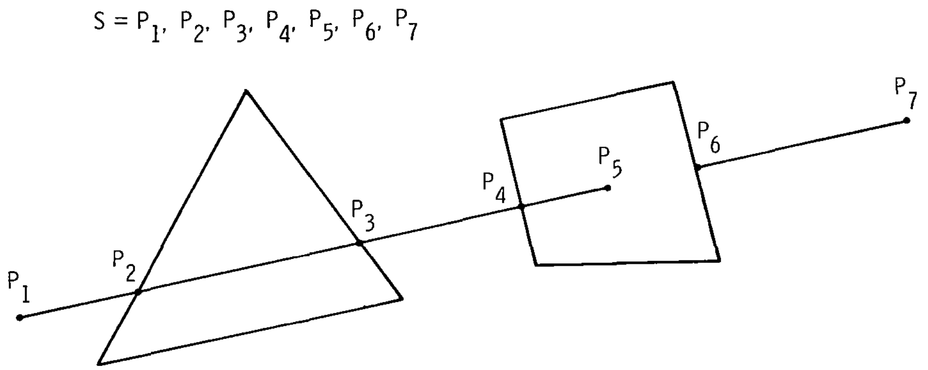


Figure 4. Spatial representations of sequence S and points that comprise set $\{P_i\}$. Note that here $\{P_i\} = P_1, P_5, P_6, P_7$, which follows from the definition of a valid intersection on ℓ_o along with its end points.

THEOREM. *Let ℓ_o be any line segment which belongs to O, any scene. Also, let S be an ordered sequence $\{P_i\} = \{x_i, y_i, z_i\}$ belonging to ℓ_o, which includes its end points along with a subset of the interior points of ℓ_o. Let these points of ℓ_o be such that their projections are a subset of all of the intersections, if any, of ℓ_o' with the interior points of other lines, say $\ell_1', \ell_2', \ldots, \ell_n'$, as well as all of the intersections of ℓ_o with the boundaries or interiors of planes $A_j, \ldots, A_k$, if any. Further, let this subset have the property that if (x_o, y_o, z_o) belongs to ℓ_o whose projection is the intersection ℓ_o' with ℓ_n', and (x_o, y_o, z_n) belongs to ℓ_n, then $z_n > z_o$. (Fig. 4 illustrates a valid selection of points that comprise the set $\{P_i\}$.)*

Then if P_i and P_{i+1} are both visible, the visibility of every interior point of

$[P_i, P_{i+1}]$ *has the same character. Moreover, if* P_i *or* P_{i+1} *is invisible, then every interior point of* $[P_i, P_{i+1}]$ *is also invisible.*

PROOF. Let both P_i and P_{i+1} be visible. Also, let P_m be any interior point of $[P_i, P_{i+1}]$ and be visible. If P_s is any other interior point, then it must also be visible. Assume the contrary whereby P_s is assumed to be invisible. Clearly then, P_s is hidden by some polygon, say A_1, which implies that P_s' lies in the interior of A_1' and satisfies Criterion III. Since P_m is visible, it is visible with respect to every polygon and in particular with respect to A_1. Therefore, P_m' either lies outside or on the boundary of A_1', or P_m' lies in the interior of A_1' and P_m satisfies Criterion II. (See fig. 5.)

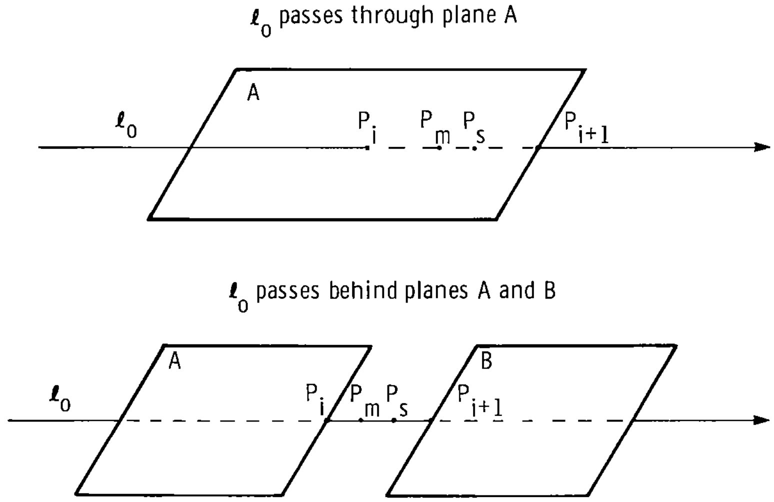


Figure 5. Visibility examples.

Suppose P_m' lies outside the boundary of A_1'. Since A_1' is closed, an edge of A_1' intersects ℓ_o' between P_m' and P_s'. That is, there is an intersection of projections between P_i' and P_{i+1}'. This implies that the z-coordinate of the point on the edge of A_1 whose projection is this intersection must be less than or equal to the z-coordinate of the point on ℓ_o whose projection is this same intersection. This observation follows from the definition of S.

Thus, there is a point on ℓ_o, say $P_t(x_t, y_t, z_t)$, such that P_t' lies in the interior of A_1' and P_t satisfies Criterion II, or P_t lies on the boundary of A_1 since P_s' lies in the interior of A_1' and P_s satisfies Criterion III. Hence, ℓ_o must intersect A_1 inside the boundary between P_i and P_{i+1}. But this is a contradiction which follows from the construction of S.

Now if P'_m lies on the boundary of A'_1, then the z-coordinate of the point on ℓ_o whose projection is the intersection of ℓ'_o with the projection of an edge of A_1 must be greater than or equal to the z-coordinate of a point on the edge of A_1 whose projection is this intersection. Clearly, this again implies there is an intersection of ℓ_o with the plane A_1 in its interior or boundary between P_i and P_{i+1}, which is impossible from the definition of S.

The same contradiction follows if P'_m lies in the interior of A'_1 and P_m satisfies Criterion II.

On the other hand, if P_m is invisible, the proof that every interior point of $[P_i, P_{i+1}]$ is also invisible follows immediately by assuming there exists an interior point, say P_s, that is visible. By letting P_m be the known invisible point in $[P_i, P_{i+1}]$, the above argument may be employed and the same contradiction reached.

Let us now define P_i or P_{i+1} to be invisible. Since, say, P_i is invisible, it lies in the interior of, say, A_1. Since P'_i is an interior point of A'_1 and P_i satisfies Criterion III, there exists another point P_s such that P'_s lies in the interior of A'_1, and P_s satisfies Criterion III. Now if we assume there exists another interior point, say P_m belonging to $[P_i, P_{i+1}]$, that is visible, again the above argument may be used and the same contradiction reached. Q.E.D.

IMPLEMENTATION

An algorithm based on the theory presented in this paper has been implemented on a CDC-6500 computer at DFRF. This approach represents a significant improvement over existing algorithms in that it minimizes the number of points interrogated for visibility without assuming any environmental limitations. Although the theorem provides a formal basis for assuring generality and rapid execution, it does not address the nuisance of square-law growth. That will now be discussed.

Initially, an $m \times n$ grid in the XY-plane is constructed whose size is $\log_2 N$ + constant, where again N is the number of elements. If an element A_j is entirely contained in a grid block, B_i, an index i, which represents the grid block, is placed in E_j. If, however, some part of the element belongs to the boundary boxes of four or less blocks and is not a proper subset of any block, then i will be

$$i = \sum_{s=1}^{k} L_s * \text{base}^{\,s-1}$$

where

$$1 < k \leqslant 4$$
L_s = the grid block number involved
base = $\log_2 N$ + constant + 1

This value of i is also placed in E_j. Thus, E_j will contain up to four of the blocks involved for the element A_j. Additionally, if A_j is inside the boundary box of a grid block but not properly contained in it, then j is stored in the matrix, $M_{k,i}$, where k is the kth element with this property and i is the grid block number.

If A_j belongs to more than four grid blocks, E_j is zero.

The indices found in E_j are sorted only once. With this arrangement, it is now possible to assign a unique address, C_{E_j}, to each sequence of like indices ignoring those values in E_j which represent more than one block involvement (i.e., $i > \log_2 N$ + constant + 1) (Scheme 1).

Thus, given an element A_j whose E_j is not zero, its relevant elements will be the elements corresponding to C_{E_j} and the M_{k,E_j} matrix. The total number of relevant elements, TN, as they relate to A_j will be

$$TN = \sum_{i=1}^{T} C_{L_i} + M_{k,L_i}$$

where

C_{L_i}, M_{k,L_i} = addresses of relevant elements

$$T = [\log_{\text{base}} E_j] + 1$$

Here, L_i are the packed block numbers derived from E_j.

If E_j is zero, then the entire collection represents the relevant elements of A_j. Note that although each C_{L_i} is mutually exclusive, this is not true for the matrix,

M_{k,L_i}. An element belonging to M_{k,L_p} may also belong to M_{R,L_w}. Thus allowances are made to eliminate redundancies.

A second scheme (Scheme 2) is adopted as follows. The minimum x- and y-coordinate values along with the maximum z-coordinate value of each element are sorted once using a method with $N \log N$ growth. This results in three lists, each arranged in ascending order. Then, given an element A, the location of its maximum x and y and minimum z in each list in turn is found logarithmically, if possible. Thus, only that part of each list such that (1) $x_{max} \leqslant x_i$, (2) $y_{max} \leqslant y_j$, and (3) $z_{min} \geqslant z_k$ is retained. The minimum $(i, j, n-k)$ and its corresponding elements become the smallest relevant set with respect to A with this scheme.

With the number of relevant elements known from both methods, the minimum count from both schemes is chosen along with the appropriate corresponding elements.

The final relevant set of elements resulting from comparisons of elements within this relevant set will be smaller yet. However, its computational efficiency is not salient since the growth pattern is ultimately predicated on the TN value. Clearly, if only the members of this class are tested against A_i, then square-law growth is avoided.

Rigorous testing at DFRF verified that the algorithm enjoys almost linear growth. It should be noted that this algorithm was benchmarked against the Watkins algorithm and Loutrel algorithm (refs. 1 and 2) and was found to be superior to both in terms of speed. This superiority increased with the complexity of the scene.

The computer program developed for the testing has about 1700 statements. The memory required for data is about $69N$ decimal words, where N is the number of elements.

Line drawings presented in appendix B illustrate the generality of the algorithm. The steps of the algorithm are presented in appendix C.

The program can be obtained from the Computer Software Management and Information Center (COSMIC), 112 Barrow Hall, the University of Georgia, Athens, Georgia 30602.

CONCLUDING REMARKS

This paper addresses a classical problem in computer graphics and presents the theoretical basis for a practical hidden-line algorithm that surmounts all of the limitations of previous solutions. Furthermore, the efficiency of the algorithm does not suffer because of its generality. To the author's knowledge, this is the most robust approach known and represents the first completely general solution to this most popular and important problem.

Dryden Flight Research Facility
Ames Research Center
National Aeronautics and Space Administration
Edwards, California, November 16, 1981

APPENDIX A

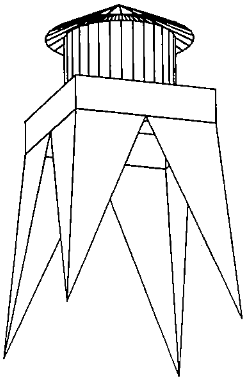
LIMITATIONS OF OTHER SOLUTIONS

The following is a list of limitations of which at least one applies to all of the known hidden-line (calligraphic) solutions to date.

Requires more than one pass.
Does not handle *n*-sided polygons.
Does not accept both concave and convex polygons as well as line segments.
Does not tolerate faces with internal boundaries (concave or convex).
Requires more topological information than the vertices.
Possesses square-law growth.
Does not accept penetrating polygons complete with lines of intersection.
Execution time grows linearly with the scale factor.
Execution time is excessive for even moderately simple structures (500 lines).
Makes mistakes frequently.
Does not handle several adjacent faces which are transparent and opaque.
Assumes a certain orientation of vertices (i.e., clockwise or counterclockwise).
Requires that polygons not be larger than a certain size.
Does not tolerate situations where an edge belongs to an arbitrary number of faces.
Has limited orientations.
Requires that every polygon belong to a polyhedron.

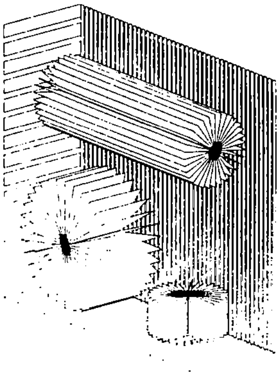
APPENDIX B

COMPUTER-GENERATED LINE DRAWINGS



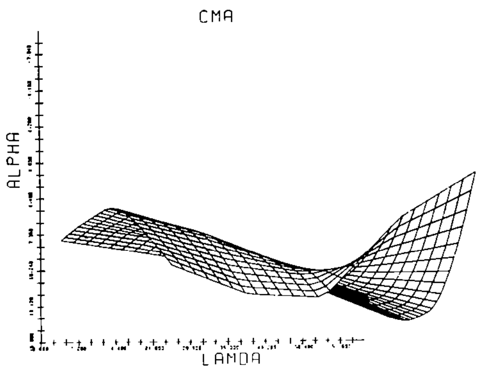
Water tower

73 polygons
Execution time = 10 sec



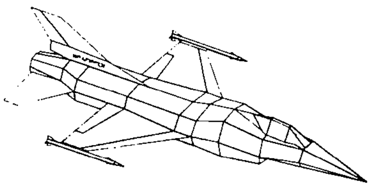
Array of planes

171 polygons
Execution time = 60 sec



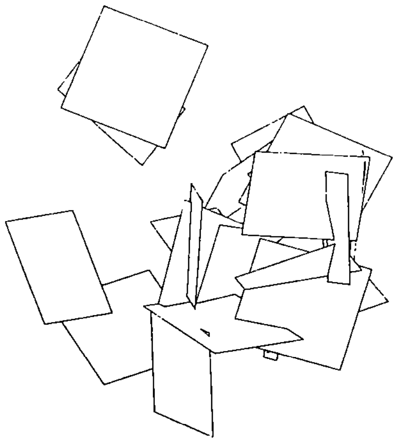
Stability derivative plot

625 polygons
Execution time = 60 sec



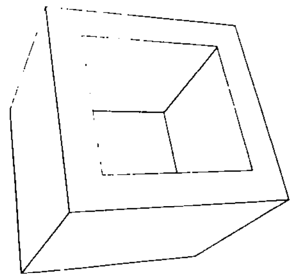
Modified F-16

159 polygons
Execution time = 12.5 sec



Intersecting planes

20 polygons
Execution time = 1.8 sec



Box with a hole

6 polygons,
Execution time = .35 sec

APPENDIX C

STEPS OF THE ALGORITHM

The sequence of the programmed algorithm is presented in the following discussion. All elements have been input in the form of (x, y, z) triplets (end points of line segments or vertices of polygons) to the hidden-line program.

Step 1. Performs the Eulerian transformations on the (x, y, z) triplets for each element. Stores all of the computed information. All further computations will be on the transformed triplets.

Step 2. Determines the equation of each plane and its edges, if applicable. All information is stored as in Step 1 and is kept in locations which correspond to that element number.

Step 3. Computes minimum and maximum x and y in the projection plane for the entire scene.

Step 4. Constructs a grid whose divisions are equal to $\log_2 N$ + constant, where N is the total number of elements. The area of the grid is predicated on the minimum and maximum values of Step 3. The divisions are formed with lines parallel to the x- and y-axes.

Step 5. Determines which elements are properly contained in a grid block and records which block. Also determines which elements are in the boundary boxes of the block but not contained in the block. Stores this information in arrays whose indices are the grid block numbers involved (Scheme 1).

Step 6. Sorts the minimum x and y and maximum z of each element.

Step 7. Begins main loop for point visibility test of each element.

Step 8. Using Scheme 2, selects revelant elements with respect to each element in turn. (Relevant elements as they relate to a given element are those elements which could possibly hide some portions of the given element.)

Step 9. Retrieves alternate set of relevant elements from Scheme 1 determined from Step 5. Chooses minimum count and corresponding elements from both schemes.

Step 10. Reduces the relevant set to a smaller subset by performing boundary box tests on the x-, y-, and z-dimensions of the given element as that element relates to its relevant set. Also performs strict overlap tests. This final set is used in the remaining calculations.

Step 11. Determines the equations of the lines of intersections, if any, between the given element and its relevant set. These equations are added to the stack of edges which bound the given element. This augmented count is used only for this particular element. The count is decremented to its original value when the algorithm proceeds to the next element.

Step 12. For each element A_j, finds the points on each edge of A_j that intersect all of the edges of the relevant elements. The actual intersection points chosen are dictated by the theorem presented.

Step 13. Sorts the intersections from left to right per line segment.

Step 14. Determines the visibility of each intersection point along with the end points of each line segment. Initially, only every other intersection point need be examined. This follows from the theorem.

The visibility criteria are described previously in the report.

The process from Step 7 to Step 14 is repeated N times.

REFERENCES

1. Newman, William M.; and Sproull, Robert F.: Principles of Interactive Computer Graphics. New York, McGraw Hill, 1973, pp. 421-454.

2. Sutherland, Ivan E.; Sproull, Robert F.; and Schumacker, Robert A.: A Characterization of Ten Hidden-Surface Algorithms. Computing Surveys, vol. 6, no. 1, Mar. 1974, pp. 1-55.

1. Report No. NASA RP-1085	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A GENERAL SOLUTION TO THE HIDDEN-LINE PROBLEM		5. Report Date March 1982
		6. Performing Organization Code 505-35-24
7. Author(s) David R. Hedgley, Jr.		8. Performing Organization Report No. H-1162
		10. Work Unit No.
9. Performing Organization Name and Address Ames Research Center Dryden Flight Research Facility P.O. Box 273 Edwards, California 93523		11. Contract or Grant No.
		13. Type of Report and Period Covered Reference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Information for obtaining the computer program is included.

16. Abstract

The requirement for computer-generated perspective projections of three-dimensional objects has escalated. A general solution has been developed at NASA Ames Research Center's Dryden Flight Research Facility. The theoretical solution to this problem is presented. The method is very efficient as it minimizes the selection of points and comparison of line segments and hence avoids the devastation of square-law growth.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer graphics Hidden-line problem Projection Planar polygon	Unclassified-Unlimited STAR category 59

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	17	A02

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1982