

SOLID MODEL INPUT THROUGH
ORTHOGRAPHIC VIEWS

Hiroshi Sakurai
Research Engineer
Nissan Motor Company

David C. Gossard
Associate Professor
Computer Aided Design Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology

Abstract: This paper describes the results of basic studies on procedures for creating solid models of component geometry from two-dimensional orthographic projections. An interactive graphic program was developed to allow the input of three orthographic views of a component geometry by digitizing from a drawing. The views may contain straight lines and circular arcs, solid or dashed. No restrictions are placed on the order or direction of lines and arcs in any view. Using an extension of the Wesley-Markowski procedure, the program constructs a three-dimensional solid model of the object. When the projections are ambiguous, multiple solid models are produced. The solid model may contain planar, cylindrical, conical, spherical and toroidal surfaces. Topological information of the solid model is stored in a winged edge structure. Geometric information is stored as vertex coordinates and surface equations.

The procedure for 2D-3D conversion provides a powerful new method for manual input of solid models, a common interface to all turnkey graphics systems, and, properly integrated with existing technology for scanning of drawings, a powerful new method for acquisition of CAD/CAM data bases from existing drawings.

The procedure is described, examples of typical input and output are shown, and possible extensions are discussed.

1. INTRODUCTION

Many research efforts are being conducted to realize complete geometric models of mechanical components for use in design and manufacturing. One method used to create a complete model of the shape of a mechanical part has been called constructive solid geometry. In this method, set operators are applied to a collection of primitive solids (e.g. cube, cylinder, sphere, cone, ...) to construct a solid model. Though this approach has

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

its advantages, it cannot convert white-frame models constructed by wire-frame CAD/CAM systems into solid models. Because solid modeling systems will coexist with wire-frame systems for some years to come, an interface between the two is desirable. Also, systems based on constructive solid geometry cannot utilize a variety of graphic input techniques used in wire-frame CAD/CAM systems.

This research effort was aimed at interfacing wire-frame and solid modeling CAD/CAM systems as well as at providing designers with more natural ways of constructing solid models. A program was developed in which a set of orthographic views, a very natural way for designers to express geometry which can be created easily with wire-frame systems, is processed into a solid model. Though currently only lines and arcs are accepted in orthographic views, many mechanical parts can be expressed by these two geometric entities. Hidden lines and hidden arcs must be identified as in conventional engineering drawings. No restrictions are placed on the order and direction of lines and arcs in each view. Planar, cylindrical, conical, spherical and toroidal surfaces are allowed in the solid model generated.

A graphics program was also developed to allow the input of orthographic views by digitizing and to allow visual verification of a generated solid model.

Much work has been done to generate polyhedral objects from their projections.

Sutherland's work in three dimensional input [1] was focused on hardware and software for digitizing. He introduced a tablet with multiple pens so that a 3D vertex could be generated by digitizing vertices in two views. He also discussed how to treat digitized data from perspective views. Thornton's work [2] concerned interactive techniques for three dimensional input from 2D views. Neither Sutherland nor Thornton investigated algorithms for construction of solid models from projections.

The first effort to construct solid models from projections was initiated by Idesawa [3], [4]. His approach was as follows.

1. Generate 3D vertices from 2D vertices.
2. Generate 3D edges from 3D vertices.

3. Generate 3D faces from 3D edges
4. Assemble faces into an object.

With this approach, false elements can be generated.

Idesawa identified some criteria to remove these false elements. For example, a 3D vertex belonging to only one face is false. Lafue [5] added a heuristic procedure for removing false elements and finding true elements. This approach has two drawbacks. First, the method cannot remove all the possible false elements and can remove some true elements in multiple solution cases. Second, they require input data to conform to a prescribed format. For example, Lafue required that 2D lines be input in such a way that a sequence of lines bounds a face.

Preiss [6] freed input from a prescribed format. He generated visible surfaces from polygons in orthographic views. He claims that he generated invisible faces using a scene analysis technique, but does not show the method. His approach also does not work for multiple solution cases.

G. Markowsky and M.A. Wesley [7], [8] treated the construction problem in a mathematically formal way so that even pathological cases, (not allowed in former work) can be accepted. An example pathological case is an edge shared by more than two faces. Their most important achievement is their method of assembling faces. Faces are connected into closures which divide infinite space into subspaces. The combinations of subspaces which give the input projections are solutions. In this way, their approach can remove all false elements and ensure that all the solutions are obtained.

All the above work concerns planar polyhedral geometries whose projections contain only lines. The work described here extends the approach of Wesley and Markowsky to create geometries which include planar, cylindrical, spherical, conical, and toroidal surfaces.

2. OBJECT AND ORTHOGRAPHIC VIEWS

In this section, orthographic views as the input to the solid model generation procedure and the object as its output are discussed briefly.

2.1 Object

An object is a finite segment of three-dimensional space and is expressed with the following elements.

Face: A face is a finite segment of a surface which is a part of the boundary separating an object from the rest of the three dimensional space.

Edge: An edge is a finite segment of a three dimensional line or curve which is a part of the boundary between a face and the remainder of the surface. An edge is shared by at least two faces.

Vertex: A vertex is one of the two points which separate an edge from the remainder of the line or curve. A vertex is shared by at least three edges.

2.2 Orthographic Views

Among many conventions in engineering drawing, the following are important in this work.

1. Three Orthographic Views: Basically, engineers express objects (mechanical parts) by a number of projected views of which three orthographic or principal views are the most important: top view, front view and side view. This work deals with the case of three orthographic views.

2. Identification of Hidden Edges: In engineering drawings, hidden edges are identified by dashed lines.

3. Silhouette Lines and Arcs: Another basic convention is to draw silhouette lines and arcs of curved faces in orthographic views. An artificial edge corresponding to a silhouette line or arc is called a silhouette edge hereafter. It lies on only one surface and appears in only one of the three views.

4. Tangency edge: In objects with planar and curved faces, two faces can be tangent to each other along a shared edge. This edge is called a tangency edge hereafter. A tangency edge does not appear in any views.

2.3 Geometric Elements

Lines and arcs are accepted in the input orthographic views.

On the output object, the following types of surfaces and edges are allowed.

1. Surfaces: Planar, cylindrical, conical, toroidal and spherical surfaces. The axes of cylindrical, conical, and toroidal surfaces must be parallel to one of the axes of the body coordinate system.

2. Edges: Three dimensional lines and circular arcs. The axes of circular arcs must be parallel to one of the axes of the body coordinate system.

Though these restrictions seem to be severe, most mechanical parts can be expressed by this set of surfaces and edges.

3. PROCEDURE

The object generation procedure has six steps. In each step, all the possible elements are generated. In some steps, generated elements are tested with various criteria, which do not eliminate all false elements but significantly reduce the effort in later steps. The final test is performed by comparing the input orthographic views with the three orthographic views regenerated from the generated object. If they match each other, the object is one of the objects or the only object expressed by the input views.

3.1 Generate P-Vertices

A p-vertex is a 2D point in a view and is a projection of a vertex. Because of the existence of silhouette edges and tangency edges, all the

p-vertices cannot be obtained by just calculating intersections in input views. In addition to intersections (p-vertices with the attribute of standard), 2D points where lines are tangent to arcs (p-vertices with the attribute of tangency) and 2D points where lines parallel to coordinate axes are tangent to arcs (p-vertices with the attribute of silhouette) are calculated. Then using the coordinate values of these points, lines in other views are cut to obtain candidate p-vertices on silhouette edges and tangency edges (candidate p-vertices with the attribute of tangency-generated and silhouette-generated). A p-vertex can have more than one attribute.

3.2 Generate Candidate Vertices

In this step, candidate vertices (c-vertices) are generated from candidate p-vertices. Then vertices whose projections were not recovered in the preceding step are generated.

1. Generate candidate vertices from candidate p-vertices: A 3D point is generated from a pair of candidate p-vertices which belong to different views and have the same coordinate value for the shared coordinate. Then a projection of the 3D point onto the third view is searched. If it is found in the third view, the 3D point is a c-vertex. Although there are five types of candidate p-vertices, only four combinations of them are allowed.

- A c-vertex with standard attribute is generated from three candidate p-vertices, all with standard attribute.
- A c-vertex with tangency attribute is generated from three candidate p-vertices, one with tangency attribute and the other two with tangency-generated attribute.
- A c-vertex with silhouette attribute is generated from three candidate p-vertices, one with silhouette attribute, one with standard attribute and one with silhouette-generated attribute.
- A c-vertex with double-silhouette attribute is generated from two candidate p-vertices, both with silhouette attribute.

All the vertices, except ones on both of silhouette edges and tangency edges, are recovered.

2. Generate c-vertices directly from the three views: A toroidal surface is assumed when the following facts are found in the three views.

- Two arcs in different views have the same radius.
- The center points of the two arcs have the same coordinate value for the shared coordinate.
- The distances from the center points to the center point of an arc in the third view are equal.

A spherical surface is assumed when the following fact is found in the three views.

- A 3D point generated from arc centers in two views has its corresponding arc center in the third view.

Once a torus is assumed from arcs, then c-vertices on that surface can be generated. Those c-vertices are given torus attribute. A c-vertex can have more than one attribute. Figure 3 shows c-vertices generated.

3.3 Generate Candidate Edges

Candidate edges (c-edges) are generated from c-vertices generated in the previous step. C-edges have three types, standard, silhouette, and tangency.

Standard c-edge: Two c-vertices are checked if their projections lie on the same line or arc in each view. If they do, the two vertices are connected by a 3D line or arc to form a standard c-edge.

Silhouette c-edge: Two c-vertices with silhouette attribute are checked if their projections lie on the same line or arc in a view. If they do, the two c-vertices are connected by a 3D line or arc to form a silhouette c-edge.

Tangency c-edge: A line-tangency c-edge is generated from two c-vertices with tangency attributes whose candidate p-vertices lie on the same arc or the coaxial arcs at the same angle. Arc-tangency c-edges are generated as follows. Two arc c-edges, each of which has at least one c-vertex with tangency attribute, torus attribute, or sphere attribute, are tested if they lie on the same torus or sphere. If they do, an arc-tangency c-edge is generated by connecting the two vertices of each arc c-edge.

In practice, more detailed c-edge types are required to avoid as many false edges as possible.

Once c-edges are generated, all the c-vertices are tested whether they lie on more than or equal to three edges, otherwise, the c-vertex is deleted and c-edges are generated again from a new set of c-vertices. Figure 4 shows c-edges generated.

3.4 Generate Candidate Faces

This step is divided into four parts. First, surfaces are generated from c-edges. A planar surface is generated from an arc c-edge or two line c-edges sharing a c-vertex. A cylindrical or a conical surface is generated from two arc c-edges connected by at least one line c-edge. A toroidal or a spherical surface has already been generated in the c-vertex generation step. For each surface, all the c-edges lying on the surface are collected. (Figure 5 (a)).

Second, a set of c-edge loops which divides the surface into the smallest subsurfaces is searched. C-edge loops, which bound their left side of a surface, are found by traversing c-edges with turn-to-the-leftmost rule. The procedure used by Markowsky and Wesley⁷ to find out loops on a plane can be applied to curved surfaces with small modification of the criterion differentiating between in-loop and out-loop. [9] In-loop bounds its

inside and out-loop bounds its outside (Figure 5(b)).

Third, a hierarchical tree of c-edge loops is generated for each surface and candidate faces (c-faces) are derived from the tree. The tree shows the relationships between c-edge loops, i.e., which loop includes which loop. From the tree, c-faces are found as follows. The out-loop which is not contained in any in-loop is discarded. In-loops at the root of the tree bound c-faces. An in-loop with its subordinate out-loops also bounds a face. (Figure 5(c)).

Fourth, the following conditions are tested for all the c-faces generated. Any standard c-edge and tangency-edge should lie on at least two different surfaces, otherwise, the c-edge is deleted. A c-face should be tangent to another c-face at its tangency c-edge, otherwise, the c-face is deleted. A c-edge related to a dashed entity in a view should have at least one c-face which hides the c-edge when it is projected onto that view, otherwise, the c-edge is deleted. If any of the c-edges are deleted, the process has to be carried out again from the beginning of this step. (Figure 5(d)).

3.5 Generate Candidate Objects

C-faces are connected to form c-face loops which divide the three dimensional space into subspaces in the analogical way as c-edges are connected to c-edge loops. [9] Two c-faces sharing an edge are connected to each other. When more than two c-faces share a c-edge, a c-face which gives the smallest face connection angle is chosen.

Each c-face loop forms a candidate object (c-object). One of the c-object generated is an infinite subspace and is discarded.

3.6 Generate Objects

C-objects can be categorized into certain c-objects, and uncertain c-objects. If a 2D edge, a straight line or a circular arc in any projected view, is related to only one c-object, then that c-object is a necessary part of the object and is called a certain c-object. If a c-object has no 2D edge which is related uniquely to the c-object, that object is an uncertain c-object.

All the certain c-objects are assembled and tested. Then each of the possible combinations of uncertain c-object is added to the assembly of the certain c-objects and tested.

The test is made as follows. The assembly is checked if it can give all the 2D edges in the input views, if so, the orthographic view is generated for each view and compared with the output views otherwise the assembly is discarded.

If the projection of the assembly matches the input view for every view, the assembly is one of the objects or the only object expressed by the input views.

4. IMPLEMENTATION

Based on the algorithm described above, a solid model generation program was developed on VAX 11/750. The program is comprised of six parts which correspond to the six steps of the algorithm. It generates solid models expressed in a winged edge data structure from the data of three orthographic views.

Though the focus of this research is the generation of solid models, a graphic program was also developed, using MEGATEK 7200 connected to the VAX 11/750, to allow the interactive input of existing drawings via a digitizer and to allow the visual verification and the selection of the generated solid models. As the solid model generation program was incorporated into this graphic program, all the work from the input of drawings to the selection of the results can be done interactively.

Solid models composed of planar, cylindrical, conical, toroidal and spherical bounding surfaces have been generated from the data describing their three orthographic views containing both lines and circular arcs.

Figures 9, 10 and 11 show some examples. Twenty to thirty seconds of VAX 11/750 CPU time were required to generate each object.

5. CONCLUSION

The method described here is more practical and useful than existing methods for the following reasons.

1. Circular arcs in input views: Most drawings of mechanical parts contain circular arcs. The method described here is the only one which allows circular arcs in input views.

2. Minimum input requirement: This method imposes less requirement on input than existing methods. For example, there is no requirement on the order of input or the labeling of the output.

The system based on the algorithm discussed here can serve as an interface between wire-frame CAD system and solid model CAD system. The system can be also used directly for the purpose of the input of solid models.

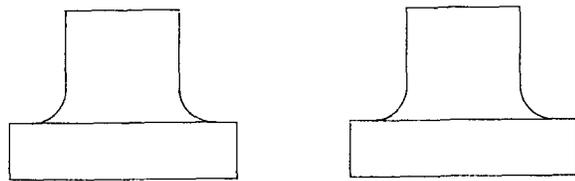
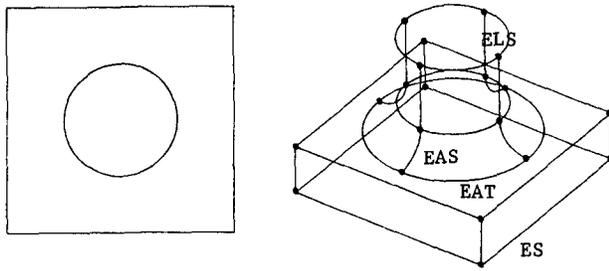
To make the system more practical, a number of problems have to be solved. In engineering drawings, there can be more or less than three projections for an object, partial projections rather than full projections may appear in some views, auxiliary views and/or sectional views may be given. It is believed that the algorithm described here can be extended to take care of the problems above, and the extension is under study.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support of the Control Data Corporation and the Nissan Motor Company.

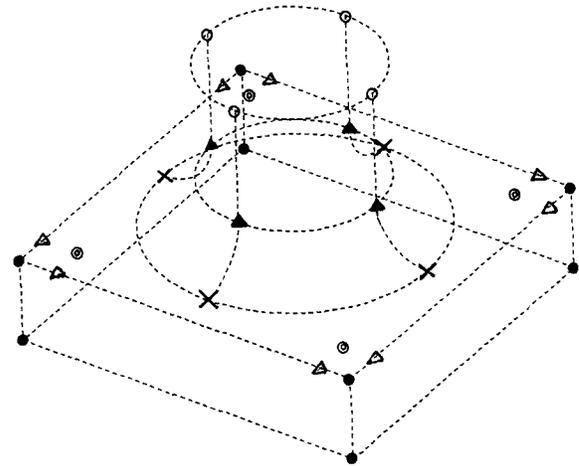
REFERENCES

- [1] I.E. Sutherland, "Three-Dimensional Data Input by Tablet", Proc. IEEE, Special Issue on Computer Graphics, Vol. 62, No. 4, 1974.
- [2] R.W. Thornton, "Interactive Modelling in Three Dimensions through Two-Dimensional Windows", Third International Conference and Exhibition on Computer in Engineering and Building Design, 1978.
- [3] M. Idesawa, "A System to Generate a Solid Figure from a Three View", Bull. JSME 16, pp. 216-225, February 1973.
- [4] M. Idesawa, et al, "Automatic Input of Line Drawing and Generation of Solid Figure from Three-View Data", Proceedings of the International Joint Computer Symposium, pp. 304-311, 1975.
- [5] G. Lafue, "Recognition of Three Dimensional Objects from Orthographic Views", Computer Graphics, Vol. 10, No. 2, 1976.
- [6] K. Preiss, "Constructing the 3-D Representation of a Plane-Faced Object from a Digitized Engineering Drawings", Fifth International Conference and Exhibition on Computer in Engineering and Building Design, 1980.
- [7] G. Markowsky and M.A. Wesley, "Fleshing Out Wire Frames", IBM J. Res. Develop. 24, pp. 582-587, September 1980.
- [8] G. Markowsky and M.A. Wesley, "Fleshing Out Projections", IBM J. Res. Develop. 25, pp. 934-954, November 1981.
- [9] H. Sakurai, "Solid Model Input Through Orthographic Views", M.S. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1982.



ES Standard edge
 ELS, EAS Silhouette edge
 EAT Tangency edge

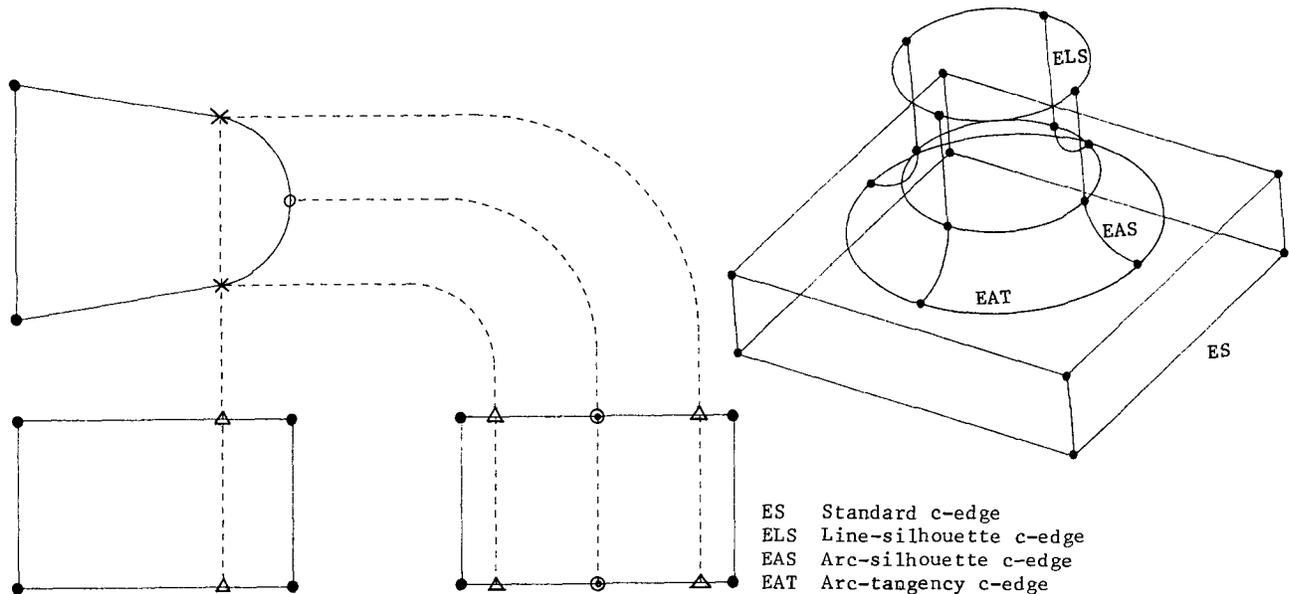
FIGURE 1 SILHOUETTE AND TANGENCY EDGES



C-vertices with the attribute of

• Standard
 ○ Silhouette
 ● Double-silhouette
 △ Silhouette and tangency
 ▲ Double-silhouette and tangency
 x Torus-silhouette and tangency

FIGURE 3 C-VERTICES



ES Standard c-edge
 ELS Line-silhouette c-edge
 EAS Arc-silhouette c-edge
 EAT Arc-tangency c-edge

FIGURE 4 C-EDGES

Candidate p-vertices with the attribute of

• Standard
 x Tangency
 ○ Silhouette
 △ Tangency-generated
 ● Silhouette-generated

FIGURE 2 EXAMPLES OF CANDIDATE P-VERTICES

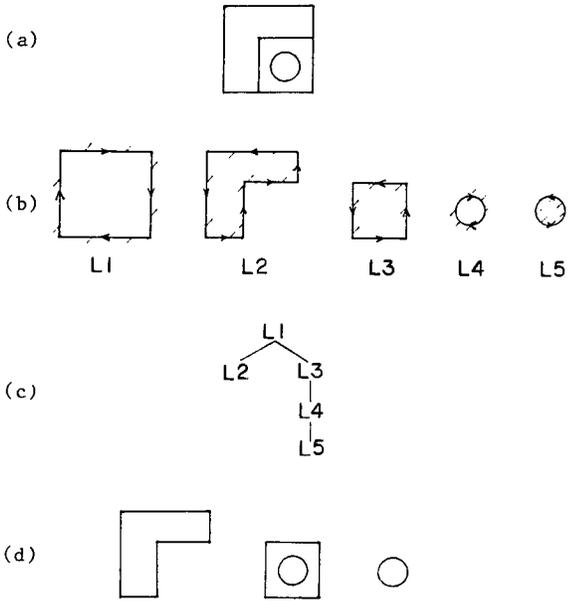


FIGURE 5 C-EDGE GENERATION: (a) c-edges on a plane; (b) c-edge loops; (c) hierarchical tree of c-edge loops; (d) c-faces generated

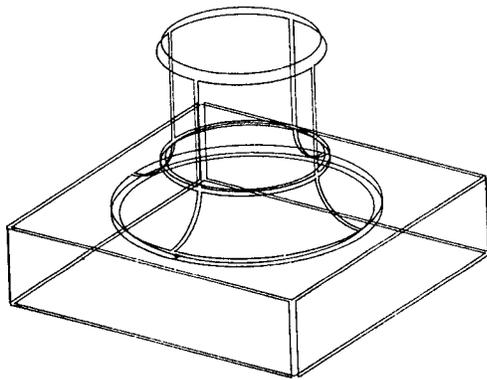
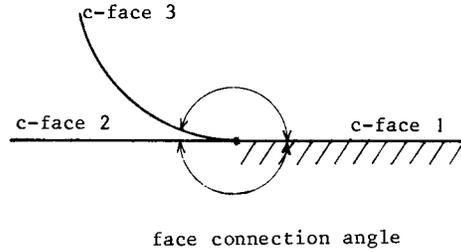


FIGURE 6 C-FACES



When c-face 1 encloses the hatched side, c-face 2 is connected, otherwise c-face 3 is connected.

FIGURE 7 C-FACE CONNECTION

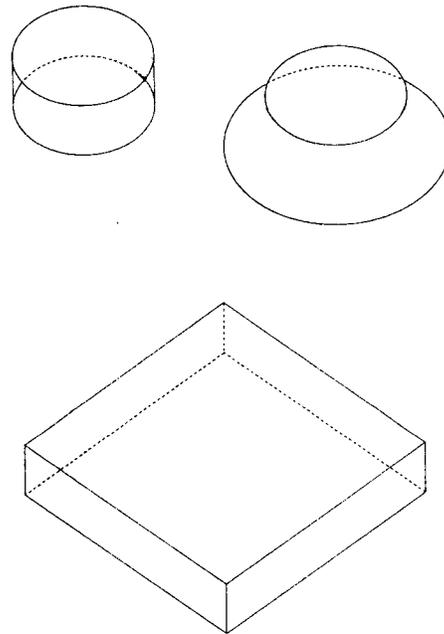


FIGURE 8 C-FACE CONNECTION

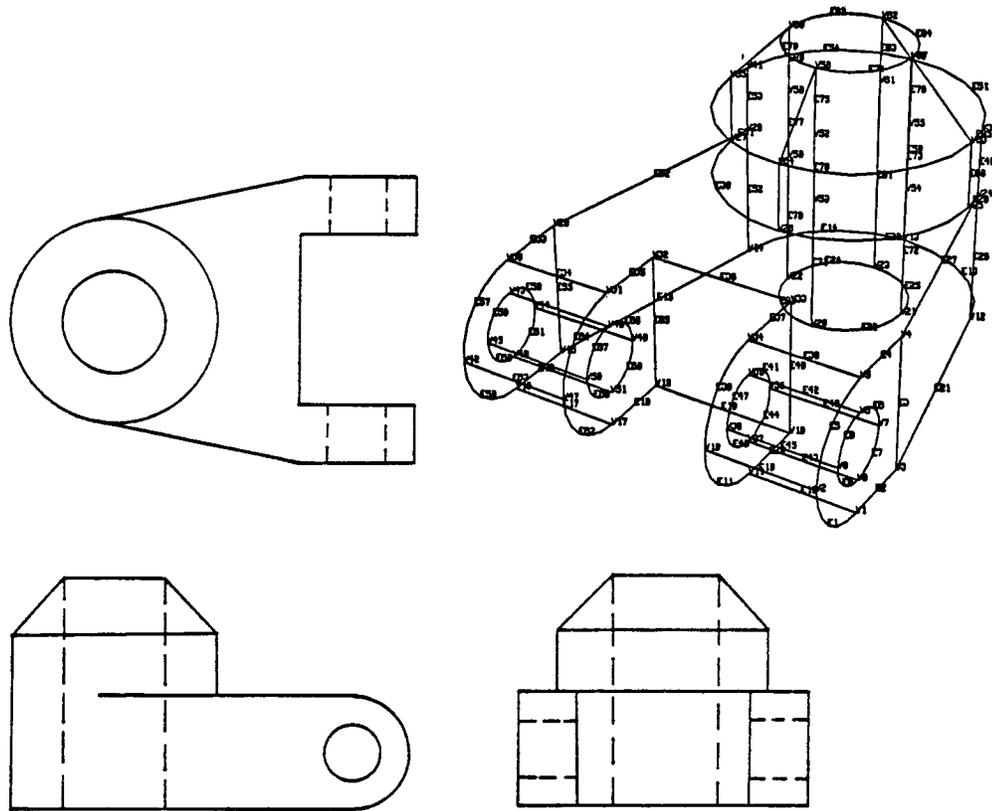


FIGURE 9 EXAMPLE 1

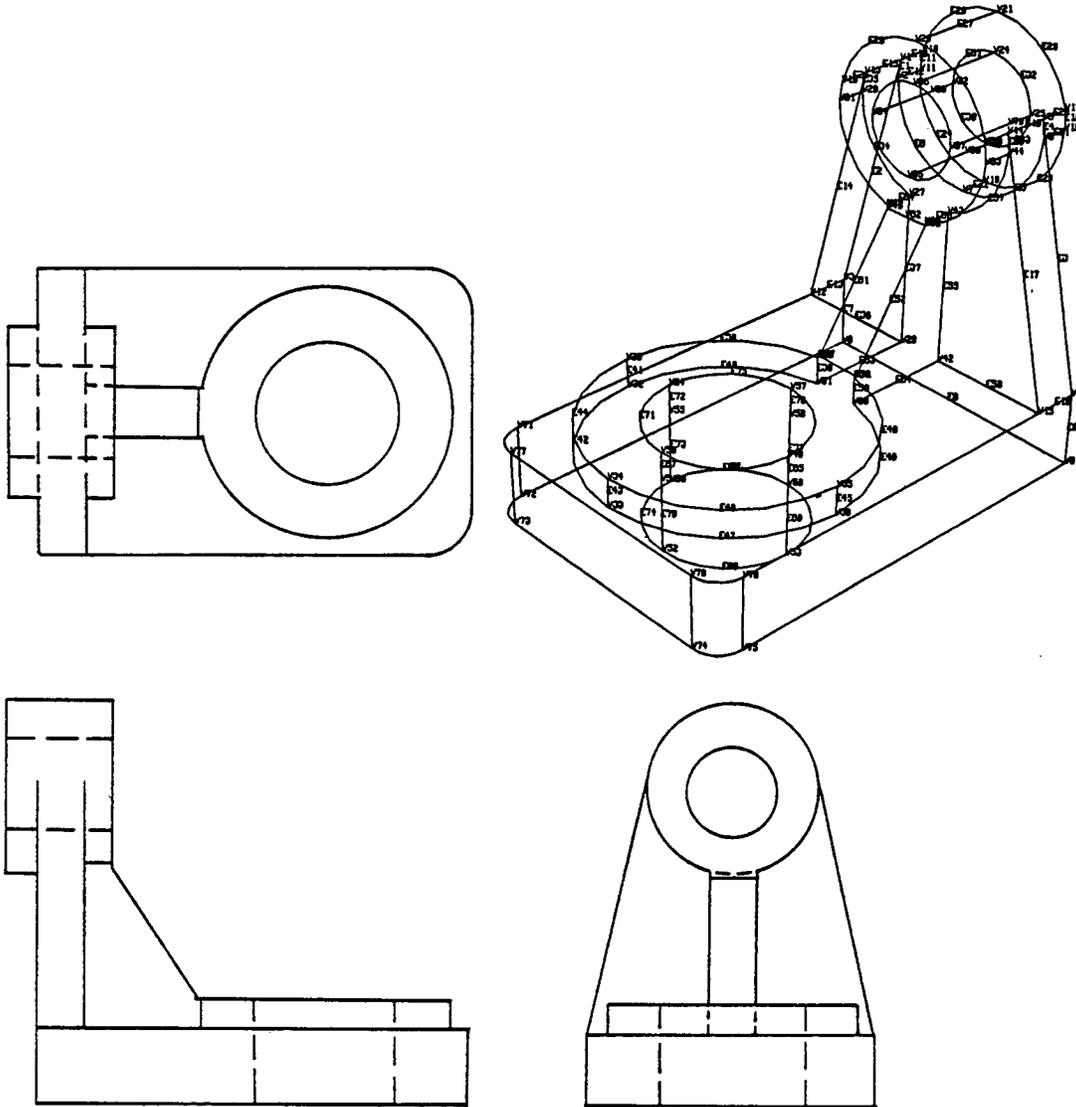


FIGURE 10 EXAMPLE 2

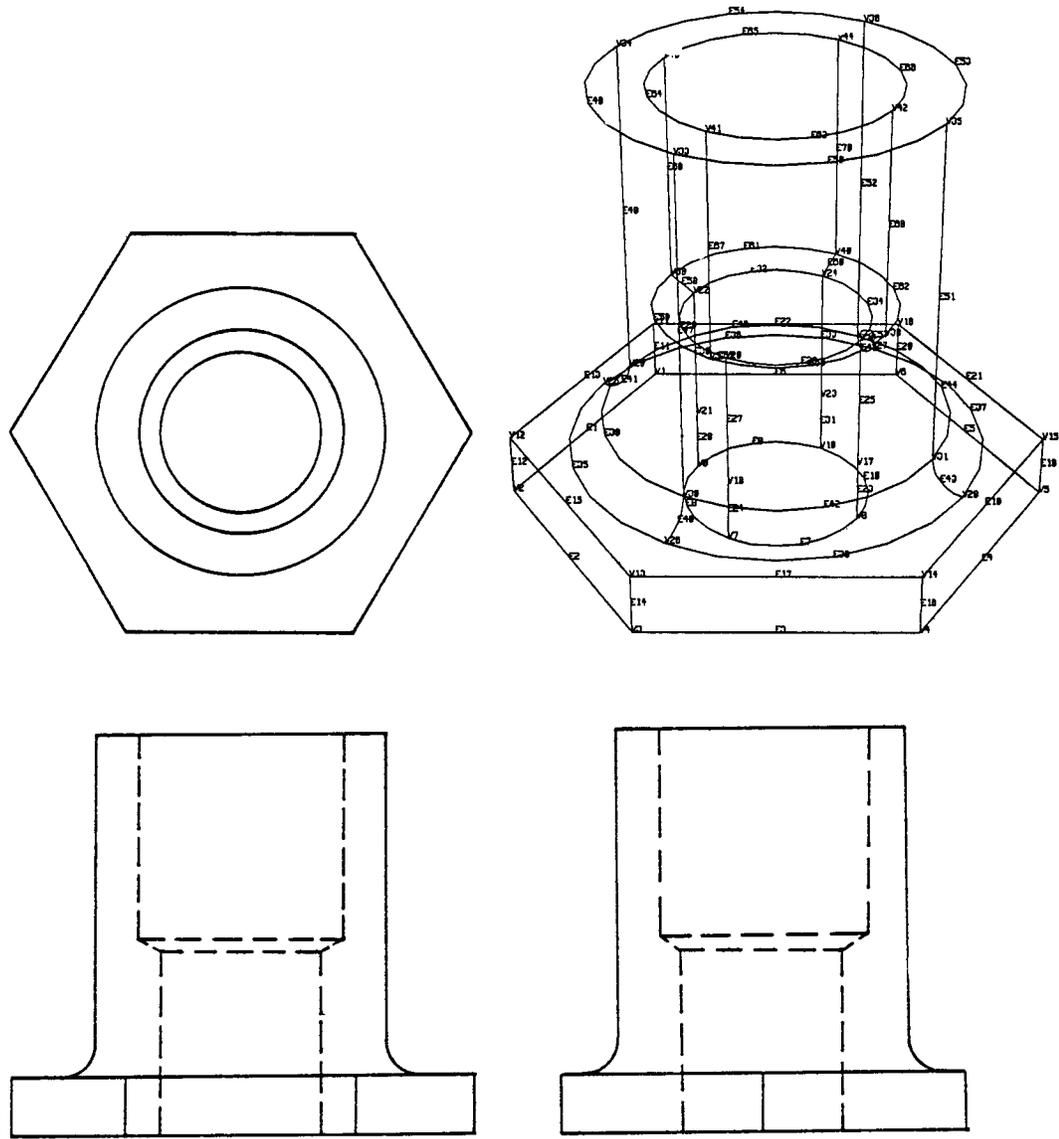


FIGURE 11 EXAMPLE 3