ISE-TR-89-77

RESTORING SOLID MODELS OF POLYHEDRONS FROM ENGINEERING DRAWINGS

Seiichi Nishihara
Yoshikazu Matsuo
Jun Nishida
Katsuo Ikeda

January 25, 1989

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

ISE-TR-89-77

RESTORING SOLID MODELS OF POLYHEDRONS FROM ENGINEERING DRAWINGS

Seiichi Nishihara
Yoshikazu Matsuo
Jun Nishida
Katsuo Ikeda

Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba, Ibaraki 305 Japan

Abstract

A system for solid modeling is described in this paper. Being given a set of three orthographic views which are input by using a facsimile device, the system restores all of the possible solid models, each of which is a set of polyhedrons consistent with the original drawing. We propose mainly two novel techniques: the line segment extraction method and the polyhedron restoration method. The line segment extraction is performed by using a probe, a small fragment of line, which moves along the central axis of each line segment represented by a binary image. The present method, therefore, do not require any preprocessing such as noise elimination or thinning of the given binary image. The polyhedron restoration method employs a face oriented approach, rather than the conventional ones adopting wire-frame oriented approach. The present method is basically a combinatorial search procedure that finds all of the legal combinations of possible faces.

1. Introduction

Modeling, or creating, three-dimensional objects in computer systems is a fundamental problem in computer graphics. Utilization of a set of several orthogonal views of objects to be modeled is a promising approach to this problem. Many studies have been conducted to interpret two-dimensional engineering drawings for restoring three-dimensional scenes, examples of which include the top-down approach[1], the wire-frame oriented approach[2,3,4,5], and the constraint satisfaction approach[6]. The last method, although not perfect, is based on the fact that each of the line segments constructing the original drawing forms local constraints with some other line segments mutually constraining each other. Such local constraints are expressed by a consistent labeling problem (CLP) [7]; thus, all of the possible three-dimensional scenes consistent with the given drawing are derived by solving the CLP.

In this paper, a system that translates an engineering drawing represented as a binary image data into appropriate solid models is proposed. We lay emphasis on describing in detail two major algorithms adopted in the system. The first one, or the segment extraction algorithm, is an image processing algorithm that extracts all of the line segments from the given binary image. And, the other, or the polyhedron restoring algorithm, is a combinatorial algorithm that searches for one or more valid combinations of faces that compose legal three-dimensional objects. The present system of interpreting engineering drawings is based on the idea that faces, rather than wire-frames, should play the main role in defining and handling polyhedral solid objects.

2. The Problem

A concrete definition of our problem is given here by clarifying the structure of drawings to be interpreted and the structure of scenes to be

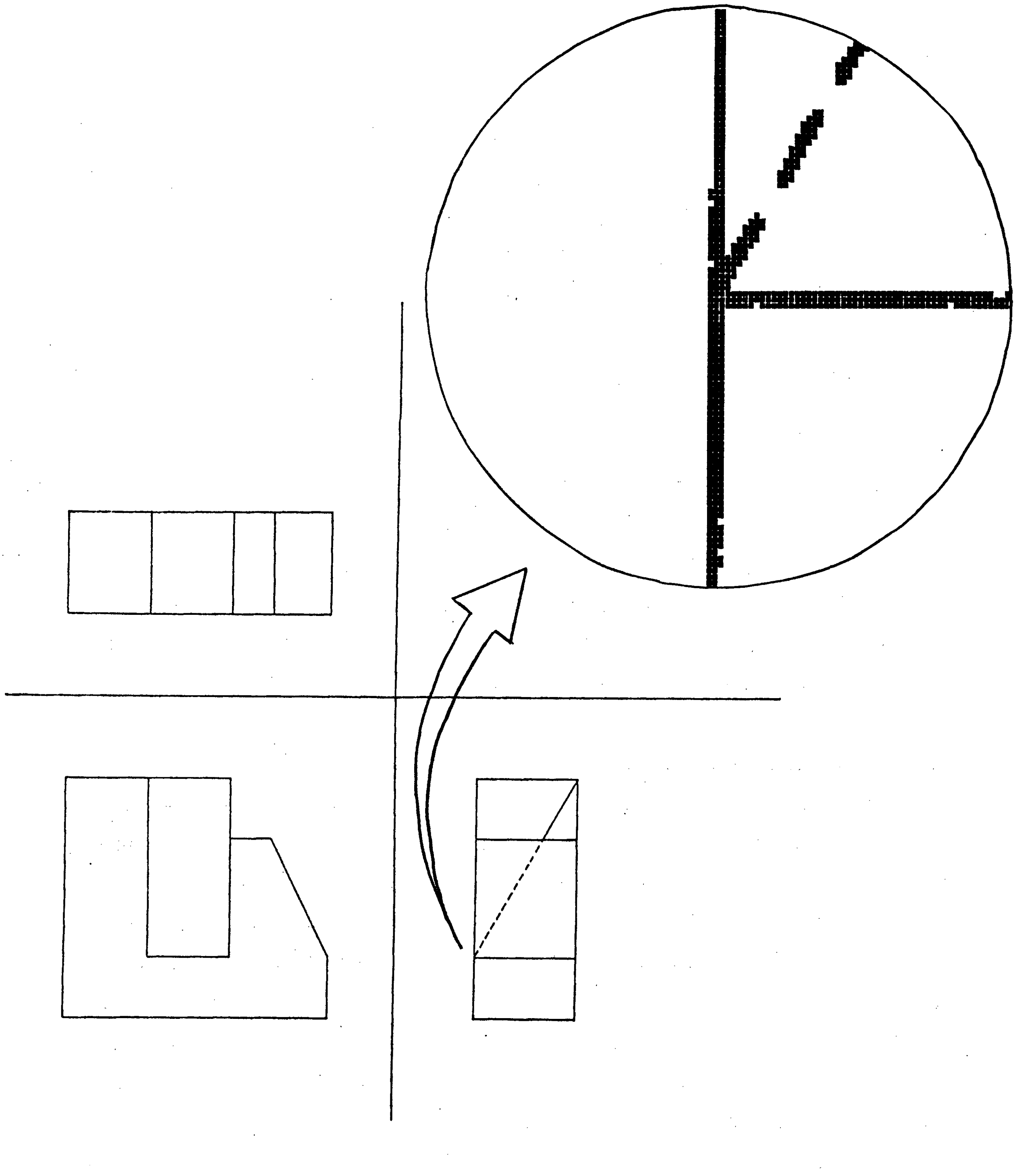


Fig. 1 An engineering drawing.

restored.

2.1 Engineering Drawings

As the example in Fig. 1 shows, the original engineering drawings are composed of three orthogonal views: top, front and (right-)side views, respectively, each of which is separated by a couple of **basic lines** intersecting each other at a right angle. Geometrical elements included in each view are only **segments** of lines. Any kinds of supplementary elements such as chain lines indicating central axis or numerals indicating sizes do not appear at all in the drawing. There are two types of segments, **solid** and **broken**. Any hidden edge should be expressed by a broken segment. A segment is defined by its two end **points**. The degree of each point, excepting end points of basic lines, is two or more, that is, at each point two or more segments meet. Particularly, at each end point of any solid segment at least two solid segments in a view is called an **area**. We call an area **simple** if all of its boundary segments are solid and further there is no internal solid segment in it.

2.2 Three-Dimensional Scenes

A three dimensional scene to be restored by our system is composed of one or more polyhedral objects, each of which is a closure of a nonempty subset of three-dimensional space bounded by a finite number of polygonal faces. A **face** is a directed finite plane bounded by a closed sequence of **edges** lying on the same plane. A **vertex** is a point where three or more edges meet. The scene should not contain any volumeless parts such as a floating face having no other contiguous face or a couple of faces lying on the same plane back to back. Simply, the system accepts any scene composed of only polyhedrons arrangeable in a gravity-free space. Fig. 2 shows an example of an acceptable scene. By orthogonal projection, each 3-D element of polyhedrons, or a vertex, an edge and a face, appears as a point, a segment/a point, and an area/a segment, respectively in any 2-D views.

2.3 Restoring Process

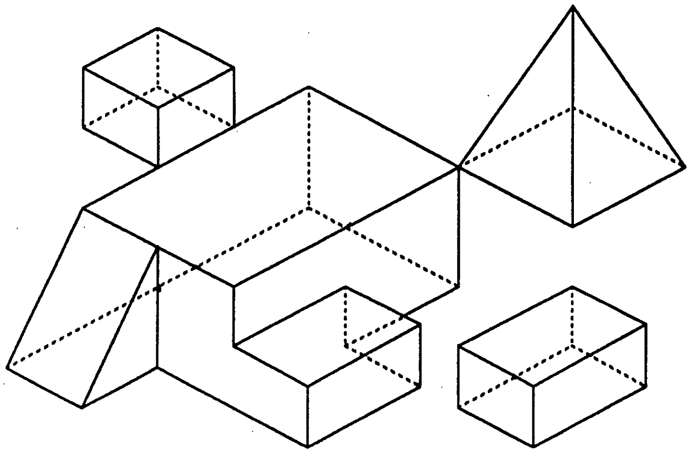


Fig. 2 A scene.

The general flow of the restoring process is shown in Fig. 3, where the types of data passed between consecutive phases are also mentioned. Looking more precisely into phase 2 and 3, which are the main objects of discussion, each of them is constructed again by a series of several steps.

[Phase 2]

Phase 2 is composed of four steps. In step 1, all the possible segment candidates are extracted by using the algorithm described in section 3. Receiving a set of segment data from step 1, in step 2 a relaxation process of unifying a set of end points lying near each other into a single point is repeated, at the end of which topological consistency of the drawing concerned is also verified as a whole. Step 3 is a simple regularization process of the input image that compensates the scanning pitch distortion caused mechanically by the facsimile itself, and also rectifies the rotational distortion of the basic lines separating each view.

The final step of phase 2 becomes the first process of the system that treats the inter-view relationship; that is, step 4 makes an overall adjustment of the coordinate of each point by examining the corresponding points in different views.

[Phase 3]

Phase 3 is divided mainly into two subparts: the first recovers all of the possible vertices, edges and faces in sequence, and the second finds every combination of faces, or every solution, that can form a possible set of legal polyhedrons consistent with the original drawing.

To see briefly how a candidate vertex is found, let three points $u=(x_u,y_u)$ in the front view, $v=(z_v,x_v)$ in the top view and $w=(y_w,z_w)$ in the side view be the projections of vertex $P=(x_p,y_p,z_p)$. Then, three equations $x_u=x_v$, $y_u=y_w$, and $z_v=z_w$ hold; which, therefore, can be utilized in restoring all of the possible candidate verticies. Similar inter-view characteristics are applied to restoring edges and faces too. But, a more detailed method of the former part of phase 3, whose basic idea owes to the one proposed by Haralick et al.[6] that converts the above characteristics to an appropriate consistent labeling problem[7], is not described

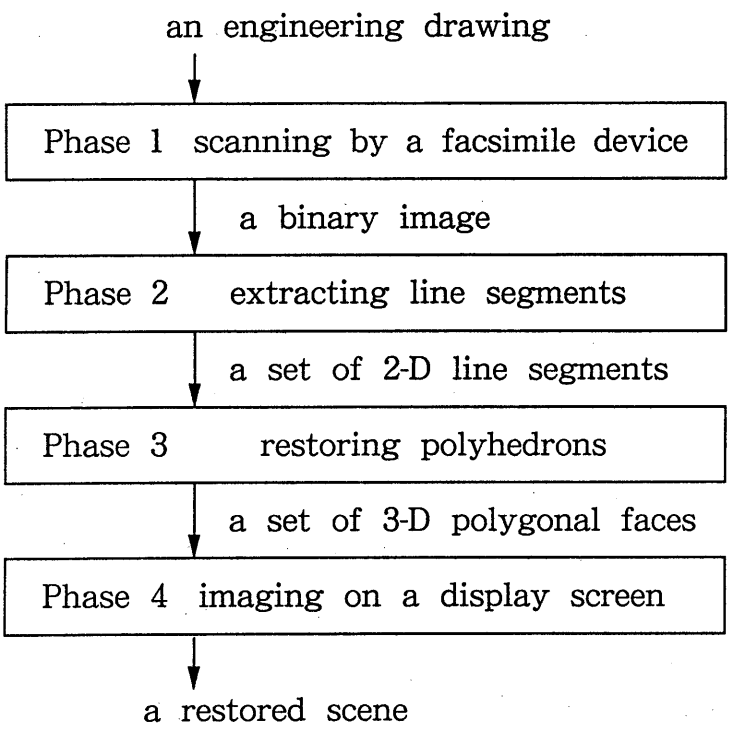


Fig. 3 Restoring process.

further.

The second part of phase 3, which seems more interesting than the first one, should essentially contain combinatorial search algorithms. As will be described in detail in section 4, our restoring strategy is characterized by a face oriented approach, which may be more natural than wireframe oriented ones.

3. Segment Extraction

Now, a method of extracting segments out of a binary image is described. As referred to in section 2.2, a segment is determined by its two end points and the line type, solid or broken. But, hereafter, we further claim that each segment should not include another point except its end points, meaning no other segment branches off except at the end points. Thus, as to the side view in Fig. 1, thirteen segments, a, b, ···, m, need to be selected, as is shown in Fig. 4, wheresegment j is the only broken segment. Nine points numbered 1 to 9 are also indicated. Notice the degree of each point is two or more. Up to now, several techniques for segment extraction have been proposed[8,9]. The novel algorithm proposed here requires no preprocessing such as noise reduction or thinning to shorten the total processing time. As shown in Fig. 1, lines are several pixels thick.

The rough structure of the algorithm is shown in Fig. 5. Statement ① is to place a probe, or a line fragment, at an appropriate general position within a new segment candidate, which will be expected to become a starting point of tracing a new segment. Fig. 6 shows some examples of the initial probes placed at a starting position for tracing a new segment, in which only probes c and d are placed appropriately. The others are not correct. As for probes a and b, they are placed too close to points P and Q with degrees 2 and 3, respectively. And, as for probe e, its upper end is not on the central axis.

Even if a probe may not be placed appropriately, at least it is sure that it forms a part of some segment. Statement ① actually finds another

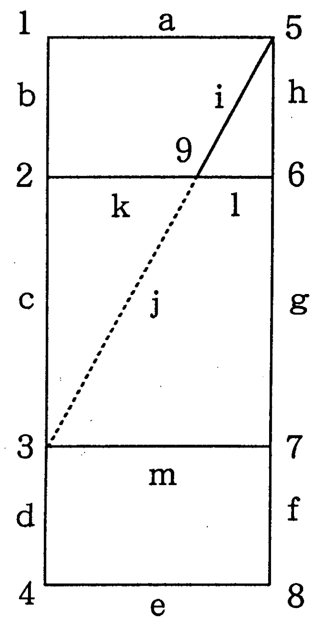


Fig. 4 Segments to be extracted.

```
begin
  while (other segments not yet visited are remaining) do
    begin
      place a probe at an appropriate position ;        ——①
      in both longitual directions of the probe do
        repeat
          make the probe move by one step incrementally —②
        until (an end point is encounted, or linearity
            of the probe movement is violated )
    end
end.
```

Fig. 5 Segment extraction procedure.

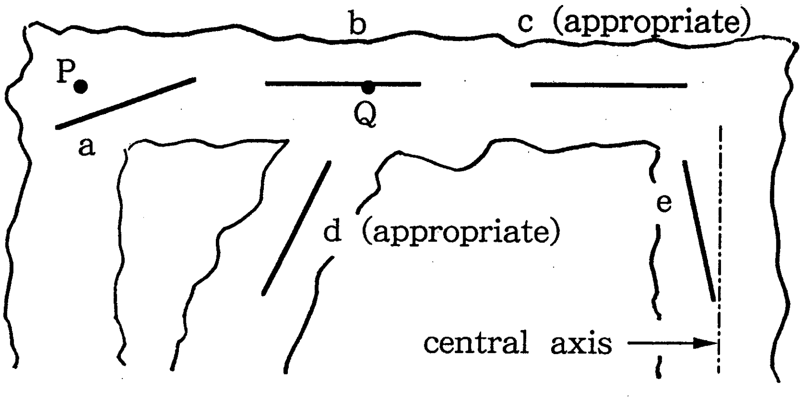


Fig. 6 Initial probe positions.

new appropriate position by making use of the inappropriate probe. In Fig. 7, let $\overline{p_1p_2}$ be the given inappropriate probe with length r. Draw a circle with the radius r round the center p_2. Then find the arc, $\widehat{st}$, composed of only pixels with value 1 and at the greatest distance from p_1. Let p_3 be the central pixel of the arc $\widehat{st}$, then probe $\overline{p_2p_3}$ may be at an appropriate position.

Next, statement ② in Fig. 5 is to move the probe step by step in the longitudinal direction of the binary image along the candidate segment. As Fig. 8 shows, the probe moves tracing the central axis; thus, the direction of the probe is directly affected by the fluctuation of the image. To absorb this kind of local anomaly, we introduced the accumulated error which is the sum of the signed difference of angles between every two consecutive probes. The requirements that assure the segment will be straight are settled as follows: (1) Any angle difference magnitude should be below the threshold α_o given beforehand. And, (2) The accumulated error should be kept small at around zero.

4. Polyhedron Restoration

Polyhedron restoration is a sort of combinatorial search problem to find all of the possible combinations of faces, each of which forms a set of legal polyhedrons. The basic idea used in our restoration algorithm is very fundamental, which is stated as follows:

> "In the three dimensional space, any closed curve crosses the surfaces of objects zero or an even number of times." ------------------ (*)

The necessary and sufficient rules to reconstruct a 3-D scene of polyhedrons consistent with the original drawing are summed up as follows:

(Rule 1) Any infinite line normal to each projection plane crosses zero or an even number of different faces.

(Rule 2) To each edge, which is a component of a face boundary, two or more even number of faces are connected.

(Rule 3) In either of the two areas adjacent each other in a view, any

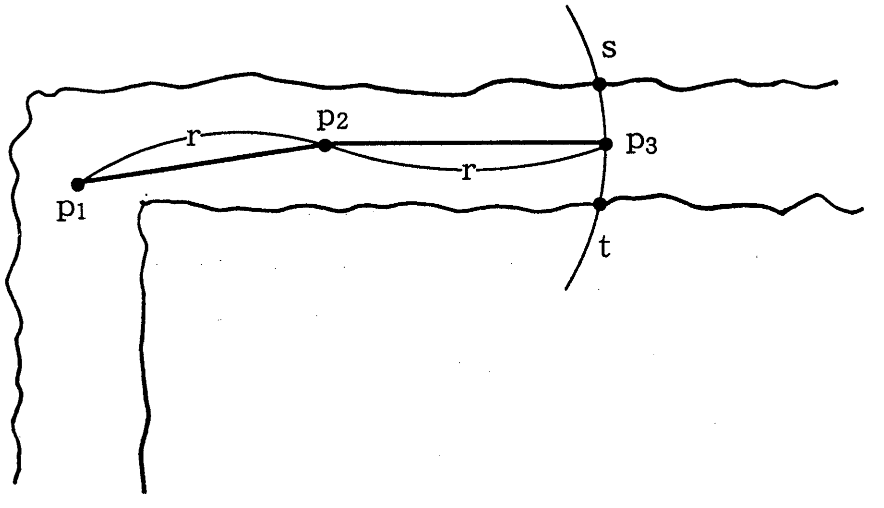


Fig. 7 Finding an appropriate probe.

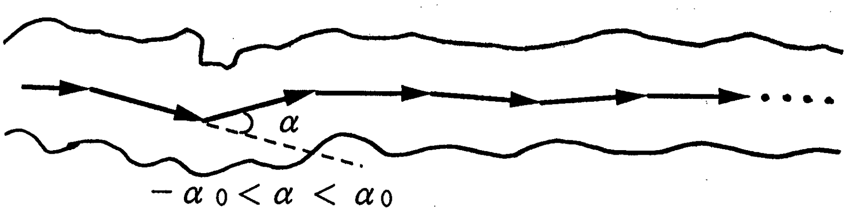


Fig. 8 Probe movement along the central axis.

infinite line as above crosses at least two faces.
(Rule 4) As to a face corresponding to an area having a broken segment as a part of its boundary, there is another face occluding the above face.
(Rule 5) At least one simple area, which isdefined in section 2.1), in each view is the projection of a face which is not occluded by any other faces.

Rules 1 and 2 are polyhedron-oriented versions of the fundamental characteristics (*); therefore, the recovered polyhedrons are all legal in the sense that they are actually realizable in a 3-D space. Rules 3 and 4 give the necessary and sufficient condition that assures the correspondence between the original drawing and the recovered scene. Rule 5, strictly speaking, is not always true; however, it is a practical rule that drastically reduces the search space.

The five rules above are fully used to carry out the search operation. First, Rule 5 is used to determine the start node of a search tree. Rules 1 and 2 are used to find all of the legal next nodes to be searched after the current node. Rules 3 and 4 are used to make the decision if the current node is a goal node giving a solution. In a search tree, each node represents a state, or a subset of possible faces.

Fig. 9 gives a general structure of the restoring procedure. By using Rule 5, the main program selects candidate faces ϕ one by one out of the faces recovered in the first part of Phase 3, to become the first node from which a depth first search starts. In the program, notation e(<f) means edge e is a part of the boundary of face f. Conversely, f(>e) means f is a face connected to edge e. Parameters E, F, C of procedure 'search' are the set of edges not yet be processed, the set of edges already be processed,and the current set of faces, respectively. Thus C reflects the current state of the search tree.

Statement ③ determines whether or not the present state is a goal, in other words, whether or not the current set of faces, C, forms a valid scene composed of one or more legal polyhedrons. Statement ④ actually produces all of the next possible states, where S is an eligible set of faces in the sense that it satisfies Rules 1 and 2 and further that each face in S is contained in C or that any part of its boundary is not

```
begin
   for each  (simple area α) do
       for each (face φ corresponding α) do
          [E : =  {e | e (< φ)} ;  F : =  {} ;
           search (E, F,  {φ})
           ]
end.
```

(a) Main program.

```
procedure search (E, F, C) ;
  [if E =  {} then
      [if (C forms a valid scene)
       then (C is a solution)]   ———③
   else
      [for each  ε ∈ E do
          for each (eligible S
                   (⊆ {f | f (> ε)})) do———④
              branch (E, F, C, S, ε)
      ]
  ].
```

(b) Node check and expansion procedure.

```
procedure branch (E, F, C, S, ε) ;
  [F : = F ∪ {ε} ;  S : = S − C ;
   C : = C ∪ S ;
   E : = E ∪ {e (< f)  |  f ∈ S} − {ε} ;———⑤
   search (E, F, C)
  ].
```

(c) Branch operator.

Fig. 9 Restoring procedure.

contained in F. In procedure 'branch', the edge ε is added to F since the processing of an eligible subset of faces connected to edge ε is completed.Statement ⑤ adds new edges necessary to be processed to E. Procedures 'branch' and 'search' call each other forming an indirect recursive procedure to branch down the search tree in the depth-first manner.

5. Some Experimental Results

While Phase 1 and 4 in Fig. 5 are implemented by using Fortran 77 on Perkin-Elmer 3220, Phase 2 and 3 are implemented by using C on Sequent's Balance 21000. As to the original drawing of Fig. 1, Fig. 10 shows the intermediate result derived after segment extraction of Phase 2 in Fig. 3. The broken segment in the original side view is expressed as a solid segment in Fig. 10; though, the line type of the segment is of course surely kept in the recovered data. The scene finally recovered is shown in Fig. 11. Another experimental result is shown in Fig. 12, where (a) is the drawing input by the facsimile device and (b) is the final result. This example shows a case that the original drawing, or a binary image, is composed of more than one connected components.

6. Conclusion

A system that interprets a given three-view engineering drawing and restores legal threedimensional solid models composed of one or more polyhedrons. In this paper, two novel techniques are mainly described: the segment extraction method and the polyhedron restoration method. In general, not necessarily only one model is restored from a given drawing. In the case two or more interpretations are possible, each of the legal model is restored and displayed in sequence.

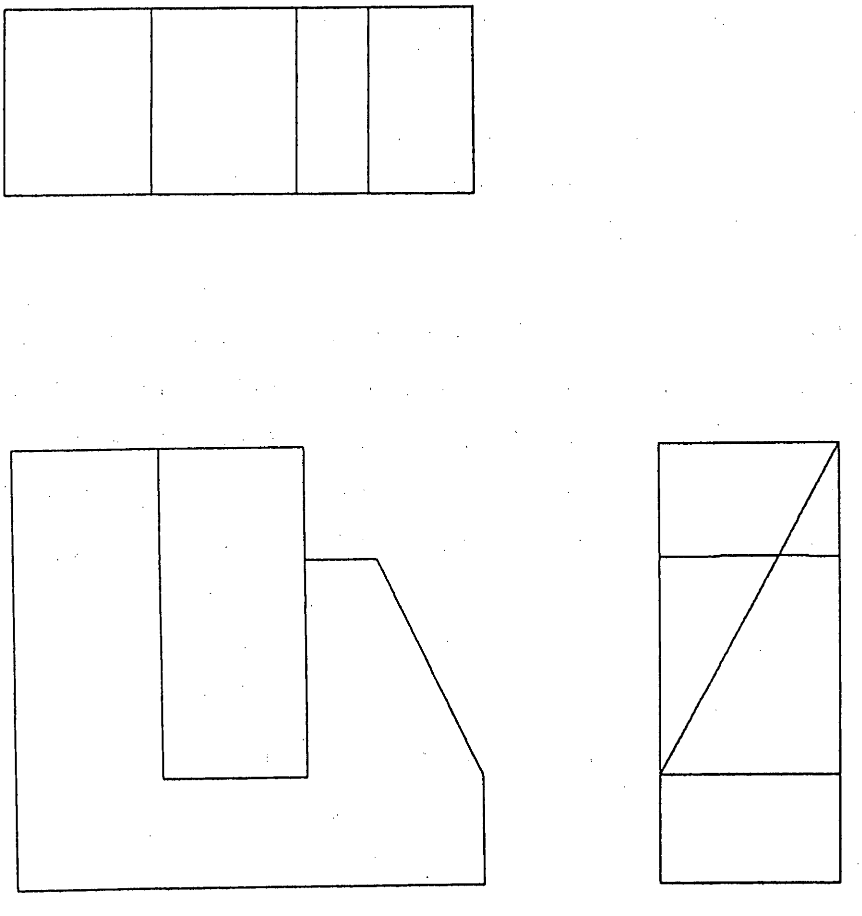


Fig. 10 Extracted segments after Phase 2.

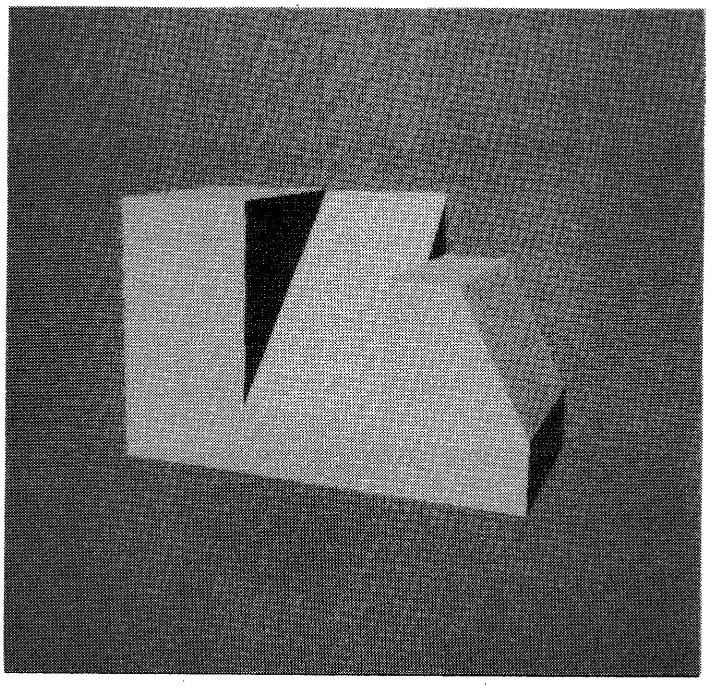
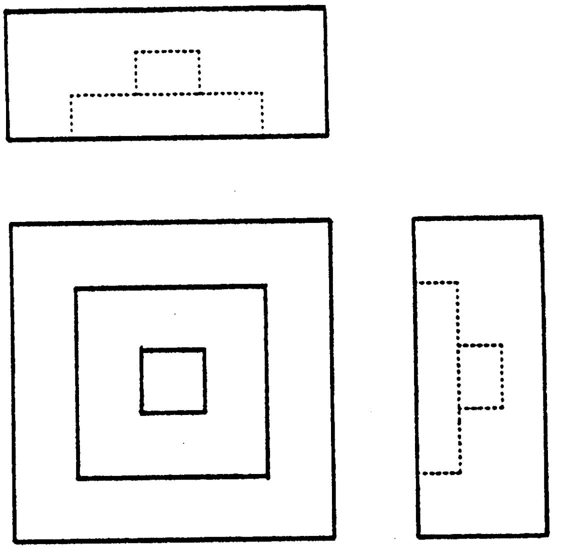
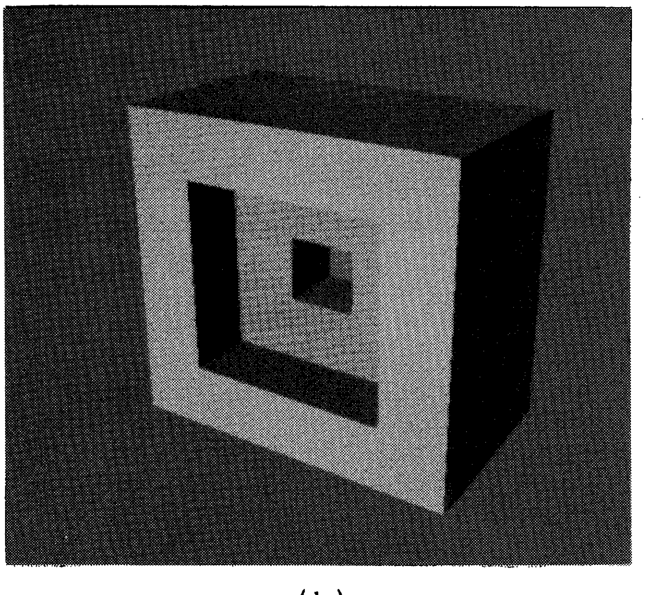


Fig. 11 The restored polyhedron.



(a)



(b)

Fig. 12 Another example. (a) The original drawing including disconnected image components. (b) The restored polyhedron.

References

[1] Yoshiura,H., Fujimura,K., Kunii,T.L.: Top-down construction of 3-D mechanical object shapes from engineering drawings, IEEE Computer,17,12 (1984),32-40.

[2] Lafue,G.: Recognition of three-dimensional objects from orthographic views, Proc. 3rd Ann. Conf. on Comp. Graphics, Interactive Techn., and Image Proc.,ACM-SIGGRAPH(1976),103-108.

[3] Markowsky,G., Wesley,M.A.: Fleshing out wire frames, IBM J. Res. Develop.,24,5(1980),582-597.

[4] Sakurai,H., Gossard,D.C.: Solid model input through orhtogonal views, Computer Graphics, ACM-SIGGRAPH,17,3(1983),243-252.

[5] Idesawa,M.: 3-D model reconstruction and processing for CAE, 8th ICPR(1986),220-225.

[6] Haralick,R.M., Queeney,D.: Understanding engineering drawings, Comput. Graphics and Image Proc.,20(1982),244-258.

[7] Haralick,R.M., Shapiro,L.G.: The consistent labeling problem: Part I, IEEE Tr. Patt. Anal. Mach. Intell.,1,2(1979),173-184.

[8] Duda,R.O., Hart,P.E.: Use of the Hough transformation to detect lines and curves in pictures, C.ACM,15,1(1972),11-15.

[9] Clement,T.P.: The extraction of line-strucured data from engineering drawings, Patt. Recog.,14,1-6(1981),43-52.

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
TSUKUBA-SHI, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE	REPORT NUMBER ISE-TR-89-77

TITLE

RESTORING SOLID MODELS OF POLYHEDRONS FROM ENGINEERING DRAWINGS

AUTHOR(S)

Seiichi Nishihara
Yoshikazu Matsuo
Jun Nishida
Katsuo Ikeda

REPORT DATE	NUMBER OF PAGES
January 25, 1989	21
MAIN CATEGORY	CR CATEGORIES
Solid Modeling	I.3.5, I,4.7, I.5.4, J.6, I.2.8

KEY WORDS

graphics, solid modeling, polyhedron, image understanding, CAD, combinatorial search, engineering drawing, pattern analysis

ABSTRACT

A system for solid modeling is described in this paper. Being given a set of three orthographic views which are input by using a facsimile device, the system restores all of the possible solid models, each of which is a set of polyhedrons consistent with the original drawing. We propose mainly two novel techniques: the line segment extraction method and the polyhedron restoration method. The line segment extraction is performed by using a probe, a small fragment of line, which moves along the central axis of each line segment represented by a binary image. The present method, therefore, do not require any preprocessing such as noise elimination or thinning of the given binary image. The polyhedron restoration method employs a face oriented approach, rather than the conventional ones adopting wire-frame oriented approach. The present method is basically a combinatorial search procedure that finds all of the legal combinations of possible faces.

SUPPLEMENTARY NOTES