ISE-TR-88-69
ISE-TR-89-79

HEURISTICS DIRECTED SEARCH IN UNDERSTANDING ENGINEERING DRAWINGS

TWO APPROACHES FOR INFORMATION RETRIEVAL THROUGH FUZZY ASSOCIATIONS

by

by
Seiichi Nishihara
Sadaaki MIYAMOTO
Jun Nishida

Shaoxing Zhang

June 3, 1988
December 20, 1989

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

ISE-TR-89-79

HEURISTICS DIRECTED SEARCH IN UNDERSTANDING ENGINEERING DRAWINGS

Seiichi Nishihara
Jun Nishida
Shaoxing Zhang

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

Abstract:

A face oriented method for restoring three dimensional scenes from a given engineering drawing is described. Restoring process is essentially a combinatorial search process to find a legal subset of candidate faces constructing a correct scene. The main purpose of this paper is to propose some novel heuristics that reduce considerably the search space. After illustrating how our heuristics are applied in understanding an example drawing, we prove the efficiency of the heuristics by experiments.

1. Introduction

Engineering drawing has been an important tool in representing three dimensional objects in manufacture and design. Recently, recognizing engineering drawings becomes a promising subject in the field of computer graphics and CAD. Many approaches to restoring solid models from engineering drawings have been reported. They can be classified into two main groups: the wire-frame-oriented approach[1,2] and the face-oriented approach[3,4]. We also proposed a face-oriented method, which demonstrated that scene restoration can be expressed ingeniously by a typical tree-search algorithm[5]. It restores exhaustively all of the possible scenes when the original drawing is ambiguous permitting more than one interpretations. However, it is time-consuming by reason that it checks every combination of faces satisfying a several fundamental rules.

In this paper, we introduce some interesting heuristics which try to let a computer follow human's understanding manner in determining initial search node and distinguishing true or false faces into the restoration algorithm, in order to improve time-consuming problem. The efficiency of the heuristics is proved by experiments.

2. Definitions and The Problem

2.1 Definitions

An original engineering drawing, or 'a drawing' in short, to be interpreted is composed of three orthographic views: top, front and right-side views. For each view, 'a viewpoint', from which the projection beam emanates, is placed at infinity far from the projection plane. Let us assume a drawing includes only two types of line segments, or simply 'segments': solid and broken. Both ends of a segment, whether it is solid or broken, are called 'points'. A closure of connected segments is called 'an area'. We call an area 'simple' if all of the boundary segments are solid and it includes no other internal solid segment at all.

On the other hand, 'a scene' expressed by a drawing is a set of one or more polyhedrons placed in a gravity-free 3-D space. A polyhedron is a

closure bounded by a nonempty set of polygons called 'faces'. A face is a closed part of a plane delimited by a series of 'edges', each end point of which is 'a vertex'.

Therefore, each 3-D component of polyhedrons, a vertex, an edge or a face, is transformed to a point, a segment/a point or an area/a segment, respectively, in the 2-D drawings of orthogonal projection.

In the preprocessing, all of the possible vertices, edges and faces are restored successively from the segments extracted from the binary image data. However, by doing so, obtained possible face set may include some faces which are unnecessary for restoring a legal scene consistent with the original drawings. This problem will be further described next.

2.2 The Problem

Here we give a concrete definition of restoring problem. Let Δ be a set of all segments, regardless of solid or broken, included in the original drawing. Being given a set Φ of all the faces, which we call 'candidate faces', our problem is to find all of the solutions each of which is a subset S of set Φ ($S \subseteq \Phi$) satisfying the following two requirements:

1. S should form a legal scene composed of one or more polyhedrons, any one of which does not contain any volumeless part, such as a floating face having no other contiguous face, or a couple of faces lying on the same plane back to back or intersecting each other.
2. Let Δ' be the set of all segments included in the orthographic views of the scene formed by S. Then, $\Delta' = \Delta$.

The first requirement is the necessary condition to make a legal scene realizable in the 3-D space. The second one is the sufficient condition that certifies the coincidence of the scene with the original drawing.

Fig. 1 shows an example of a drawing and all of the candidate faces recovered. Thus,

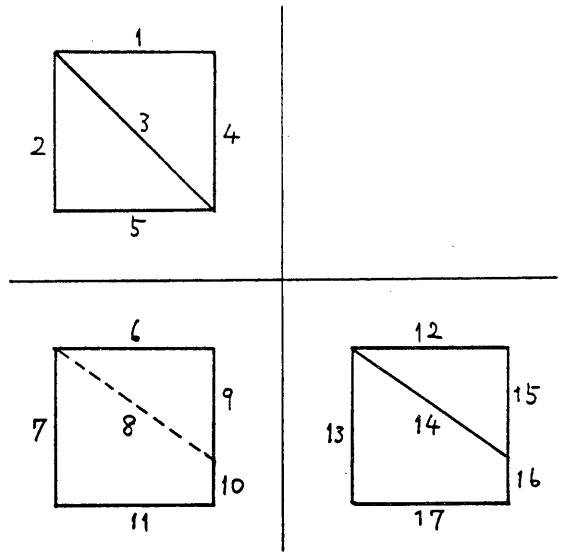
$$\Delta = \{1, \ldots, 17\} ,$$

$$\Phi = \{f_1, \ldots, f_{12}\} ,$$

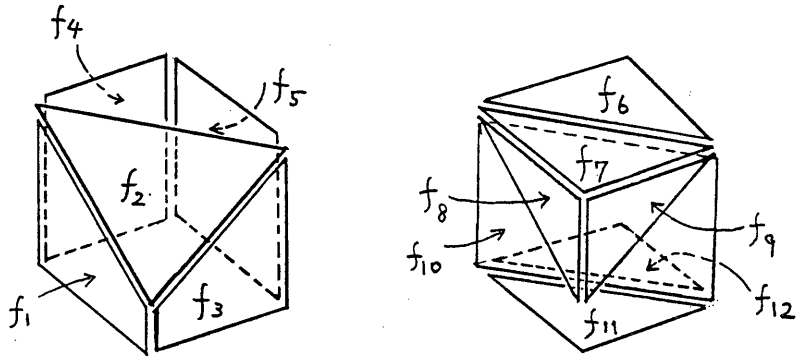
and the only one solution found is as

$$S = \{f_1, f_2, f_3, f_4, f_5, f_6, f_{11}, f_{12}\} .$$

It may easily be seen that S surely forms a legal polyhedron and the



(i) An engineering drawing.



(ii) Candidate faces restored.

Fig. 1 An example.

projection of S coincides with Δ.

3. Combinatorial Search Algorithms

3.1 A Simple Algorithm

Solving the problem defined concretely in the previous section is fundamentally a filtering process of Φ to get a subset S. Actually we can get all solutions by repeating generate-and-test operations; though, this kind of brute force method may awfully be slow.

From the requirements proposed above, we can derive a set of rules as follows, which will effectively be utilized by a basic combinatorial algorithm later.

(Rule 1) For each view, there is at least one face which is not occluded by any other faces seen from the viewpoint.

(Rule 2-1) Any infinite line orthogonal to the projected plane crosses zero or an even number of different faces.

(Rule 2-2) To each edge, two or more even number of faces are connected.

(Rule 2-3) When two or more faces are placed across each other, at most only one face among them can actually be adopted as a component of a solution.

(Rule 3-1) For each segment in a drawing, there must be at least one edge projected to it.

(Rule 3-2) If a face boundary contains an edge whose projection appears as a broken segment in a view, there is at least one other face placed nearer to the viewpoint. (That a face, f, is 'nearer' than another one, g, means f is placed nearer to the viewpoint, or is placed farther from the projection plane, than g.)

Though, strictly speaking, Rule 1 is not necessarily correct, it is used practically to get an appropriate starting node of tree search.

Rules 2-1 to 2-3 correspond to the first requirement given in 2.2, and are derived from a very fundamental characteristics as follows:

> "The necessary and sufficient condition of existence of one or more objects in 3-D space is that any closed curve crosses the surfaces of objects zero or an even number of times."

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

(a) Main program.

```
procedure search (E, F, D, S) ;
  [ if E = { } then
      [ if D = { } then
          [ if (S forms a valid scene)
            then (S is a solution) ] ————————①
        else
          [ E' : = {e (≫λ) | λ∈D} ; ————————②
            for each ε' ∈E' do
                search ({ε'}, F, D, S)
          ]
      ]
    else
      [ for each ε∈E do
            for each (eligible C (⊆ {f | f (>ε)})) do ——③
                branch (E, F, D, S, C, ε)
      ]
  ].
```

(b) Node check and expansion procedure.

```
procedure branch (E, F, D, S, C, ε) ;
  [ F : = F ∪ {ε} ; C : = C − S ;
    S : = S ∪ C ;
    E : = E ∪ {e (<f) | f∈C} − {ε} ; ————————④
    D : = D − {λ | λ (≪ε)} ;
    search (E, F, D, S)
  ].
```

(c) Branch operator.

Fig. 2 A simple search algorithm.

While these three rules certify the realizability of the scene, Rules 3-1 and 3-2, which correspond to the latter requirement given in 2.2, certify the coincidence of the restored scene with the original drawing.

Fig. 2 shows a general structure of the simple restoring algorithm, where E is a set of edges to each of which two or more even number of faces to be connected, F is a set of edges to which even number of faces have already been connected, S is a subset of candidate faces, i.e. $S \subseteq \Phi$, expressing the current node, and D is a subset of Δ, i.e. $D \subseteq \Delta$, containing segments which have not yet appeared in the projected view of S. In Fig. 2, $e(<\phi)$ means edge e is a part of the boundary of face ϕ. Conversely, $\phi(>e)$ means ϕ is a face connected to edge e. $e(>>\lambda)$ and $\lambda(<<e)$ means segment λ is the projection of edge e.

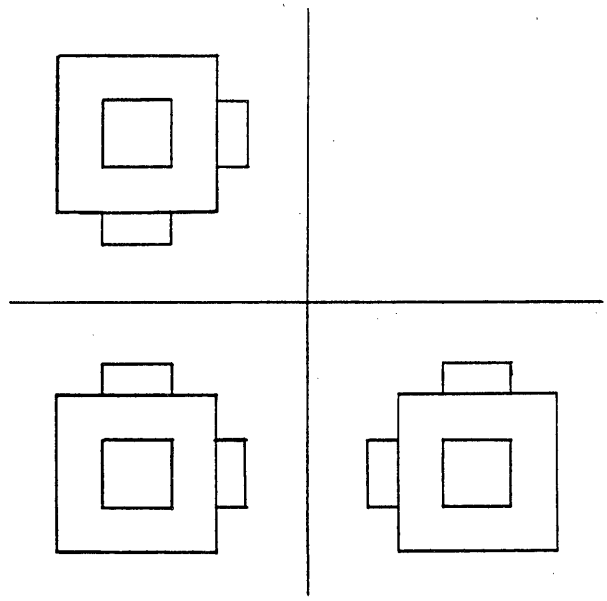
First, a starting node is determined in the main program by using Rule 1. In the search tree, each node is represented by a set of candidate faces, and the set for a node is always a superset of the set for its parent node. Thus, a branch down operation of the tree search is actually an adding operation of some candidate faces, which is shown explicitly by $S := S \cup C$; in 'branch' procedure. In ①, the current node S is checked if it is a goal or not. In ②, the search as for the original segments ($\in \Delta$) which have not yet appeared resumes. In ③, the current node is expanded to produce all of the next nodes to be branched down. In ④, unprocessed edges which are newly produced by connection operation of a face are added to E.

The above algorithm checks all of the combinations of candidate faces; therefore, all of the possible solutions are restored when the original drawing contains some ambiguity, as the example in Fig. 3 shows.

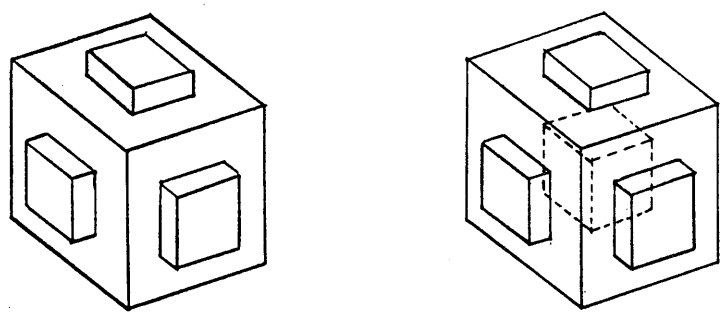
3.2 Introducing Heuristics

Two novel heuristics that try to let a computer avoid acting blindly and cultivate analysis ability to reduce the search space are developed:

Heuristics 1: Out of three orthographic views, the view which is composed of the smallest number of simple areas but composed of as many areas containing broken segments as possible is selected as the view from which a starting node, referred as the nearest face in the main program in Fig. 2, should be



(i) An ambiguous drawing.



(ii) Two possible solutions.

Fig. 3 Ambiguity of drawings.

chosen.

Heuristics 2: Instead of checking exhaustively all of the combinations of faces, more sophisticated approach making use of local constraints and characteristics should be adopted to determine true or false faces. (Hereafter, a face is called 'true' if it is included in the solution S. Otherwise, the face is called 'false'.)

The latter heuristics is decomposed into four explicit rules as follows:

(Rule 4) As for each area adjacent to the background, at least two corresponding faces must be true. Especially, if there are only two candidate faces corresponding to such an area, both of them are true, meaning both of them should be included in any final solution.

(Rule 5) If two faces which are projected to two areas adjacent to each other are the nearest faces lying on the same plane, at most only one of them can be true.

(Rule 6) A face is judged to be true, if it is the only one face connected to a true face.

(Rule 7) A face is judged to be false, if it is the only one face connected to a false face.

Let us see how the heuristics above are applied to an actual example. The engineering drawing to be processed is shown in Fig. 4.

First, the preprocessing phase of the system restores all of the possible candidate faces, which are shown in Fig. 5. Then, by using Heuristics 1, the front view of Fig. 4 is chosen as the view from which the starting node of search should be picked out. After this, many candidate faces can be judged true or false successively by using the heuristic rules given above, which is described in the following.

1) (Heuristics 1) The first search node is settled as {a}, a set of single face 'a' which is the nearest candidate face corresponding to area A.
2) (Rule 4) Faces a and b in Fig. 6(i) are judged to be true, because they are the only two candidate faces corresponding to the lower subpart of area A.
3) (Rule 3-2) Face c also in Fig. 6(ii), corresponding to the rest (i.e. upper part) of area A, and further lying on the same plane of face a, is judged to be true.

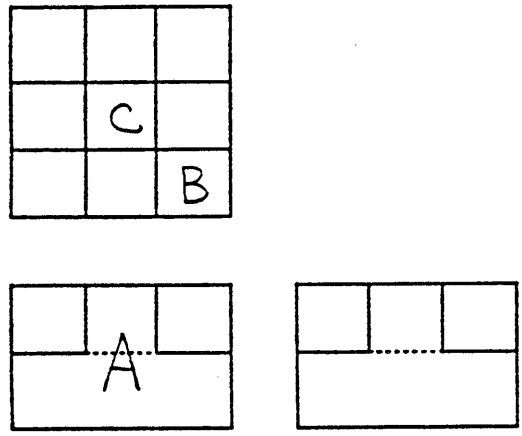


Fig. 4 A drawing.

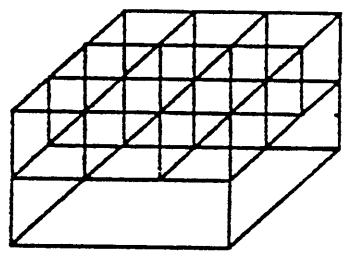


Fig. 5 Restored candidate faces.

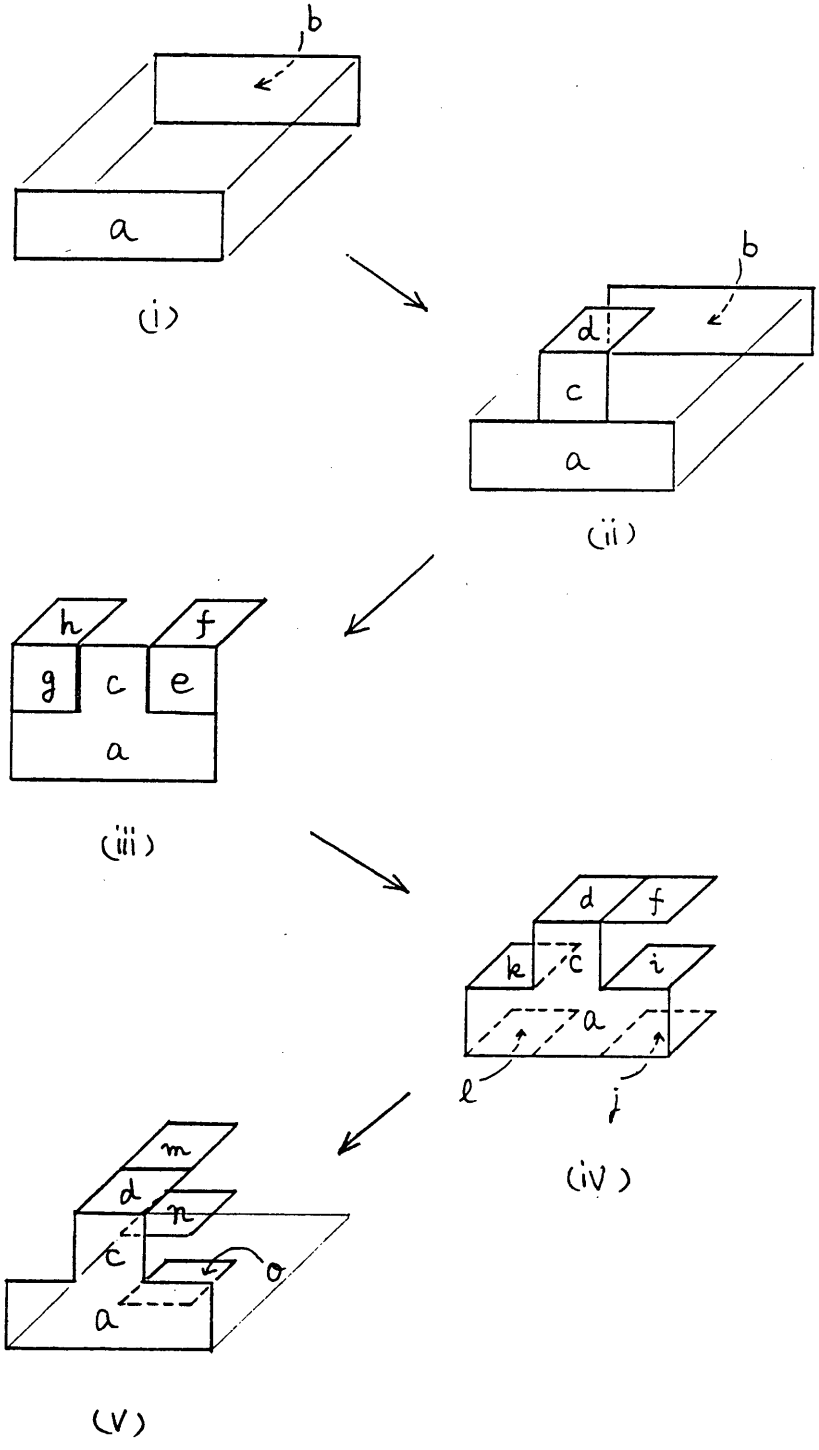


Fig.6 Determination process of truth or falsehood of faces.

4) (Rule 6) Face d in Fig. 6(ii) is true, because it is the only face connected to the true face c.
5) (Rule 5) Face e in Fig. 6(iii) is judged to be false, because e is the nearest face and is a neighbor of a and c lying on the same plane, both of which are true. Face g is also false for the same reason.
6) (Rule 7) Face f in Fig. 6(iii) is judged to be false, because f is the only face connected to the false face e, so is face h.
7) (Rule 2-1) All of the candidate faces projected to area B in Fig. 5 are f, i and j in Fig. 6(iv). Because f is known to be false, the rest faces, i and j, must be true. Similarly, faces k and l also are judged to be true.
8) (Rule 5) In Fig. 6(v), face m cannot be true for the same reason that face e is proved to be false.
9) (Rule 2-1) Thus, face n and bottom face o are turned out to be true, because they are, excepting false face m, the faces projected to area C in Fig. 4.

After pursuing this kind of heuristic inference, most of the candidate faces are judged to be true or false. In our example, the result is shown in Fig. 7, where shaded faces are the faces judged to be true. Heuristics 1 and 2 are expected to be very effective to reduce the search space; though, some candidate faces may still be left undetermined. As for these remaining faces, the basic combinatorial search shown in Fig. 2 is applied.

4. Experiments

The heuristics proposed above are incorporated into the algorithm in Fig. 2, which is implemented on Sequent's Symmetry S81 by using language C. We applied our algorithm to two examples shown in Fig. 4(case B) and Fig. 8(case A), to see how well our heuristics work to clarify truth or falsehood of all candidate faces, which may, as the result, affect the processing time needed.

The restored polyhedrons are shown in Fig. 9. As Table 1 shows, in case A, all of the 25 candidate faces are judged to be true (21 faces) or false (4 faces), while, in case B, 16 faces out of 47 candidates are still left undecided after applying Heuristics 1 and 2.

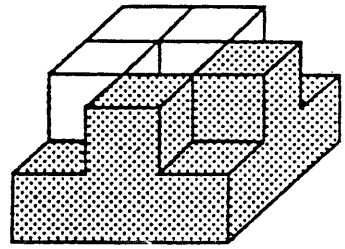


Fig. 7 Faces concluded to be true.

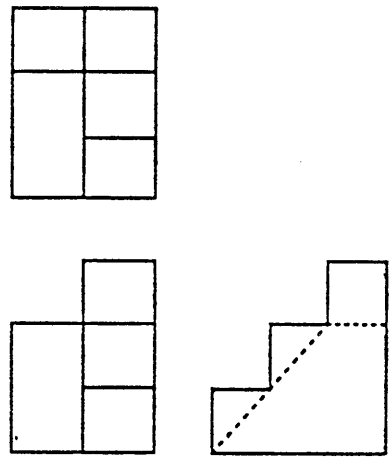


Fig. 8 Another drawing.

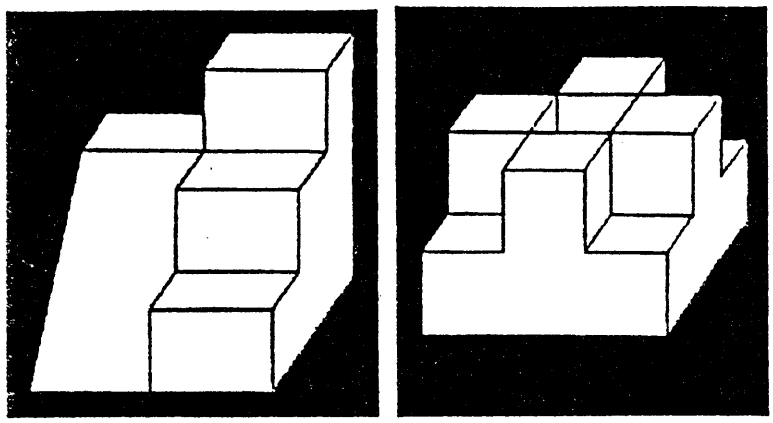


Fig. 9 Restored polyhedrons.

Table 1. Experimental results.

case	#candidate faces	after applying heuristics 1 and 2			CPU time (sec)		
		truth	false	undecided	simple algorithm (Fig. 3)	with heuristics 1	with heuristics 1 & 2
A (Fig.8)	25	21	4	0	3.1	1.7	1.6
B (Fig.4)	47	19	12	16	945.4	799.4	67.9

As an interesting result, in case A, Heuristics 1 contributes to the improvement of time efficiency much better than Heuristics 2, and vice versa in case B.

5. Conclusion

A system that restores 3-D scenes from a given drawing is described. After clarifying that the problem is essentially a kind of combinatorial search, we proposed a simple search algorithm. Then, we introduced two novel heuristics that reduce the search space to speed up the search process. The mechanism of the heuristics is explained by using an example. The efficiency of the heuristics is proved experimentally; though, more systematic evaluation of the heuristics and to find more efficient heuristics are the future problems.

References

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

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

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

[4] Sasaki,Y., Itoh,K., Suzuki,S.: Solid generation from orthographic views by non-linear pseudo-Boolean algebraic solution (in Japanese), Jour. Inf. Proc. Society of Japan, 30,6(1989).

[5] Nishihara,S., Ikeda,K.: Interpreting engineering drawings of polyhedrons, 9th ICPR(1988).

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

REPORT DOCUMENTATION PAGE

REPORT NUMBER
ISE-TR-89-79

TITLE

Heuristics Directed Search in Understanding Engineering Drawings

AUTHOR(S)

Seiichi Nishihara
Jun Nishida
Shaoxing Zhang

REPORT DATE
December 20, 1989

NUMBER OF PAGES
16

MAIN CATEGORY
Solid Modeling

CR CATEGORIES
I.3.5, I.5.4, J.6, I.2.8

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

ABSTRACT

A face oriented method for restoring three dimensional scenes from a given engineering drawing is described. Restoring process is essentially a combinatorial search process to find a legal subset of candidate faces constructing a correct scene. The main purpose of this paper is to propose some novel heuristics that reduce considerably the search space. After illustrating how our heuristics are applied in understanding an example drawing, we prove the efficiency of the heuristics by experiments.

SUPPLEMENTARY NOTES