An approach to Three-Dimensional Decomposition and Description of Polyhedra

VISION FLASH 31

by

John M. Hollerbach

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

Robotics Section

JULY 1972

Abstract

This paper presents a description methodology for trihedral planar solids that, as in Roberts' approach, decomposes an object into simpler components. The present approach, however, is more sophisticated and results in a more natural description. Hidden vertices are located in the process of description generation. Also, it is shown how the 3-D coordinates of the vertices can be obtained from the 2-D coordinates.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

Table of Contents

1. Introduction

The construction of a three-dimensional description from the two-dimensional projection of planar solids is an important goal in the development of any nontrivial vision system. Analysis of a scene using only two-dimensional information has inherent limitations; in particular, object identification and stability analysis are rendered exceedingly difficult, if not sometimes impossible.

The general problem of three-dimensional description has three aspects:

(1) recognition of distinct objects in a scene;

(2) identification of an object, e.g., cube, wedge, etc.;

(3) three-dimensional coordinate generation.

The greatest success in the past has been with aspect (1) (Guzman (3) and Huffman (4)). Guzman (2) was concerned with aspects (1) and (2), but that work cannot generally be said to have been successful. Roberts (5) attacked all three aspects, and in my opinion produced the most significant results with regard to aspects (2) and (3). The latter two works are discussed further in section 4.

This paper assumes that distinct objects in a scene have been recognized and an appropriate description set up. Only single trihedral planar solids subjected to trimetric projections are handled (these terms are defined in section 3.1). The basic motivation for the present approach is that objects can often be described as the projection of the most complex plane along parallel lines eminating from vertices of the plane (as used in this paper, plane will be taken to mean an area bounded by straight lines). In many cases where direct application of this procedure is not possible, the solid can be subdivided into parts, each of which is amenable to this procedure. By keeping track of the planes projected, it is possible to construct a description of the object in terms of these planes, thereby solving aspect (2).

Successful application of this procedure to the trimetric projection of an object results in the determination of the 2-D coordinates of hidden vertices. When this is done, it is proved in section 2.3 that for the class of objects under consideration, knowledge of the 3-D coordinates of exactly four independent vertices suffices to determine the 3-D coordinates of the remaining vertices of the solid. This takes care of aspect (3).

2. Exposition

2.1. Basic Propa-pH

The present approach grew out of the observation that one plane often provides complete information about an object. This circumatance occurs when the object can be described as a parallel projection of this plane (thus the acronym Propa-pH for project parallel plane hack). For example, the description of a cube is obtained as follows (Fig. 1).

(1) Move plane A along the parallel rays from the vertices of the plane until the ends of the rays are reached. Call the projected plane P(A).

(2) Connect corresponding vertices of A and P(A).

Many common objects, such as wedge, cube, and block, can be described in this manner. Some examples of objects amenable to this simple procedure are given in Figs. 2 and 3, along with hidden lines and vertices found as a consequence of its application.

A common feature of the objects in the previous figures is that the plane projected can be considered to be the most complex plane. To test whether a particular trimetric projection represents an object that is describable by projection of one of its planes, it suffices to find the most complex plane and to apply the afore-mentioned procedure to it. To be more precise, a binary relation is-more-complex-than is defined below. If

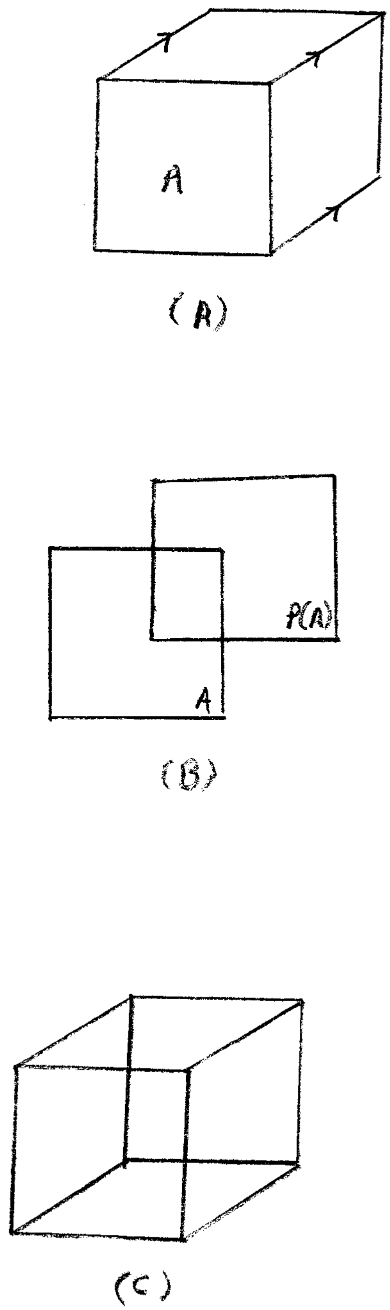


Fig. 1

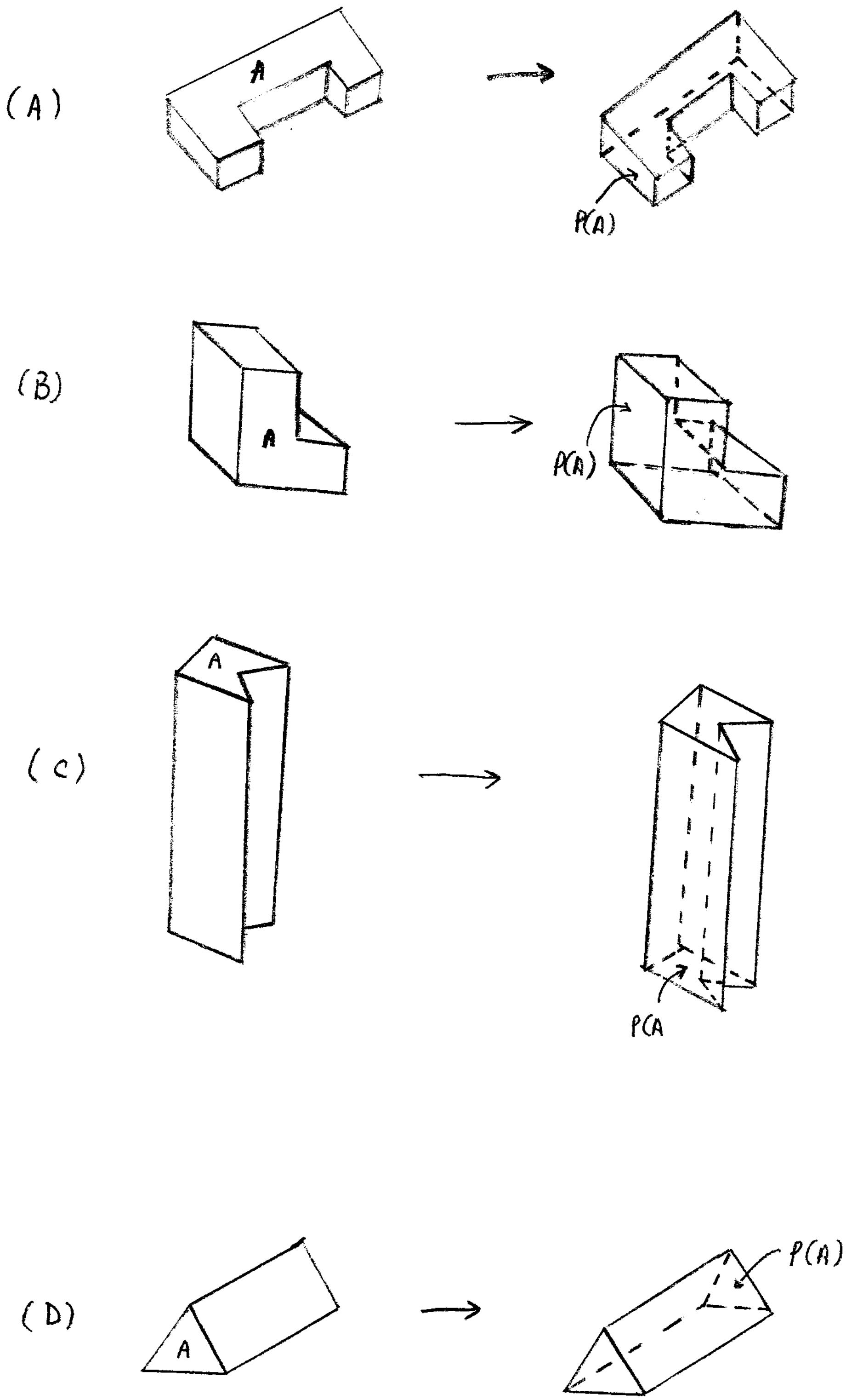


Fig. 2

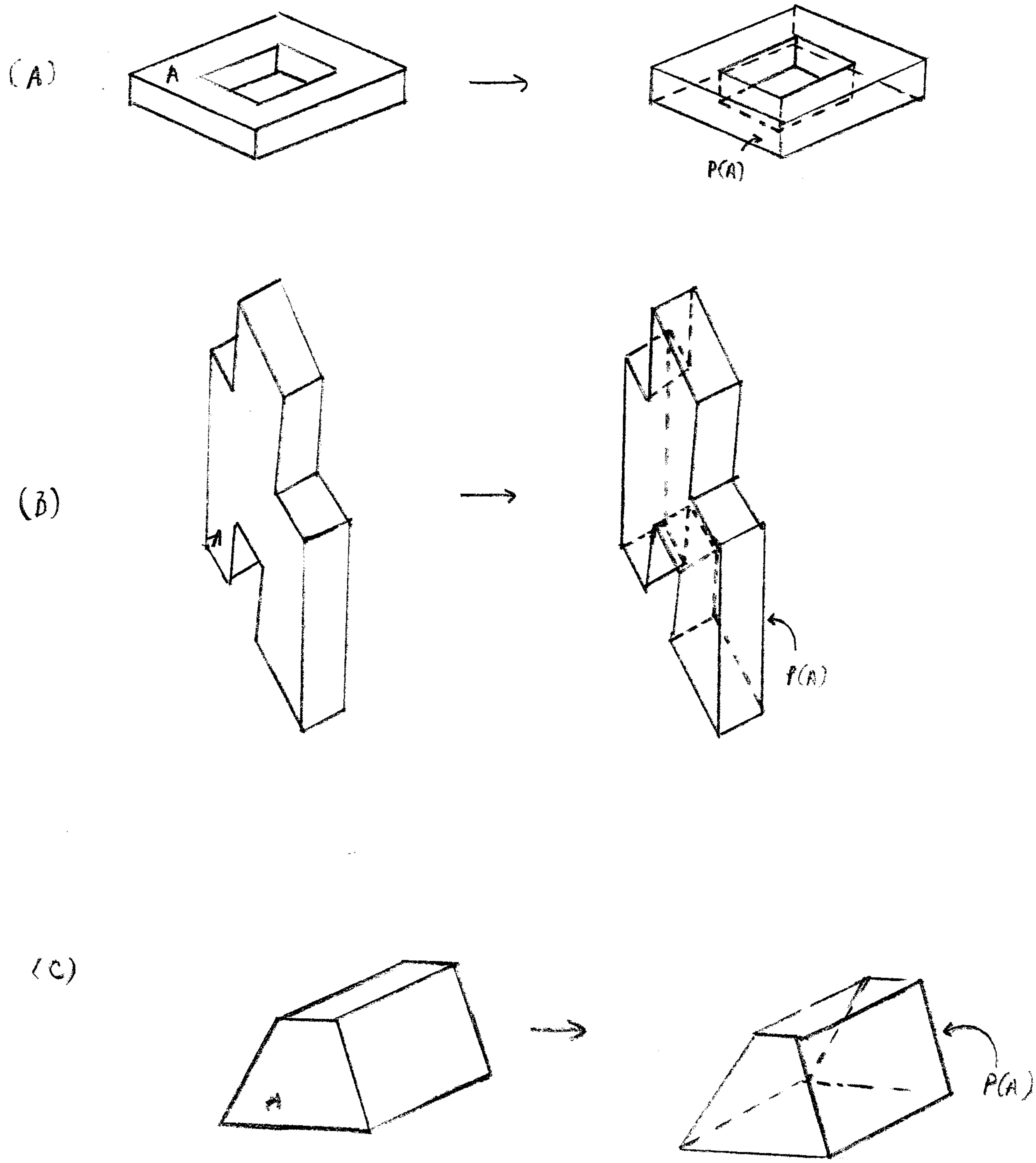


Fig. 3

a plane has as an edge the shaft of a T, the relation is not defined on it. Let n(A) be the number of edges of plane A and p(A) be the number of pairs of parallel edges of A. Then A is-more-complex-than B if and only if one of the following is true.

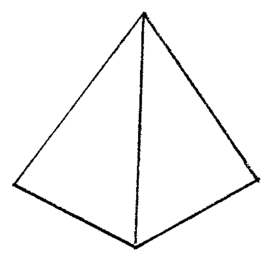
(1) $n(A) > 4$ and $n(A) > n(B)$.

(2) $n(A)=3$ and $n(B)=4$.

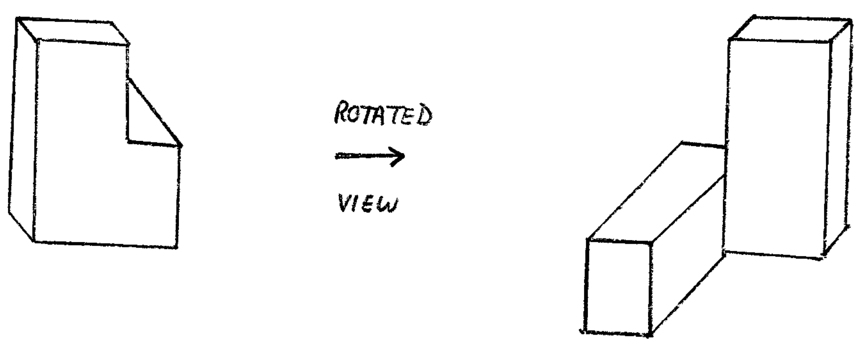
(3) $n(A) = n(B)$ and $p(A) < p(B)$.

Thus the most complex plane can generally be considered to be the one with the greatest number of edges. Note, however, that a triangular plane is considered to be more complex than a quadrilateral (see Fig. 2D). In the case that two planes have the same number of edges, we choose the one with the least parallelism among its edges (see Fig. 3C). Sometimes more than one plane can be considered most complex (e.g., Fig. 1), in which case any one of them can be selected.

Having found the most complex plane, it must be the case that all rays eminating from the plane are parallel. If not, the test fails (e.g. Fig 4A). By ray will be meant an edge eminating from a vertex of a plane that is not an edge of the plane itself. Since all vertices are taken to be trihedral, this definition is unambiguous. The rays must also be of equal length, excluding those that are shafts of T's (see Figs. 2B and 3B). Finally, those rays that are shafts



(A)



(B)

Fig. 4

of T's must be shorter than those that are not (e.g., Fig.4B).

If all these conditions are fulfilled, the object formed by connecting corresponding vertices of the most complex plane A and P(A) is compared with the original object. If there are no edges of the original object that are not also edges of the object formed, the test succeeds. Another way of saying this is that nothing should be left when the object formed is subtracted from the original object. The conditions stated in the previous paragraph are not enough to insure that the test succeeds, as Fig. 5 testifies.

If the test succeeds, the object can be considered to be described by its most complex plane. In the process of connecting corresponding vertices of A and P(A), the 2-D coordinates of hidden vertices were computed. This knowledge is necessary for finding the 3-D coordinates of all vertices by the method given in section 2.3.

2.2 Compound Object Construction

When an object is not describable as a projection of its most complex plane, it is often possible to break this compound object into pieces that can be so described. For example, the object in Fig. 6A can be considered to be an L-shaped block stuck at one end to a rectangular block. How Propa-pH actually constructs this description is also demonstrated in the Figure.

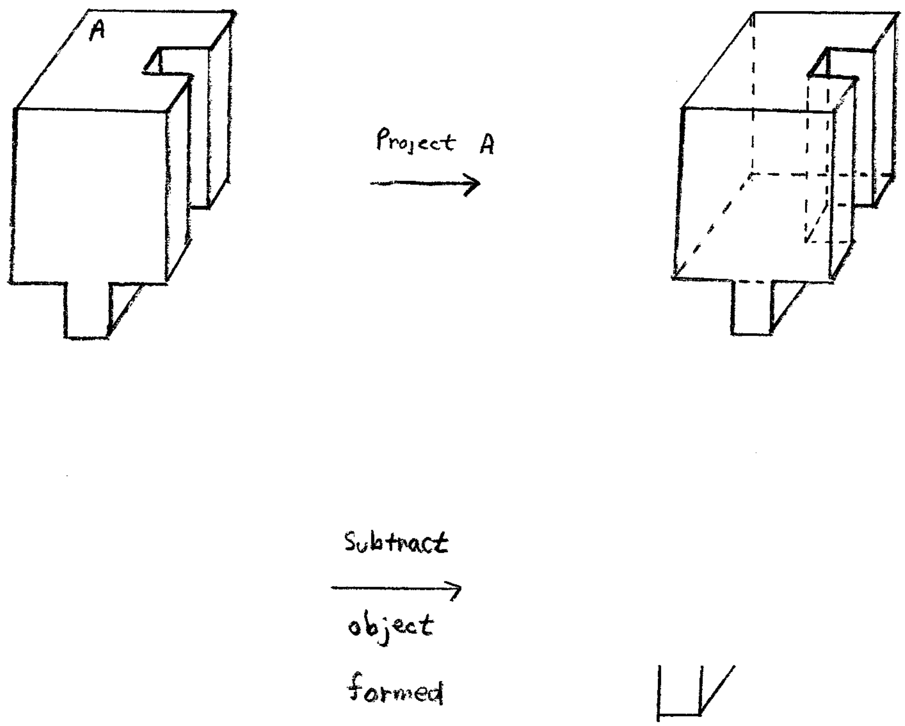


Fig. 5

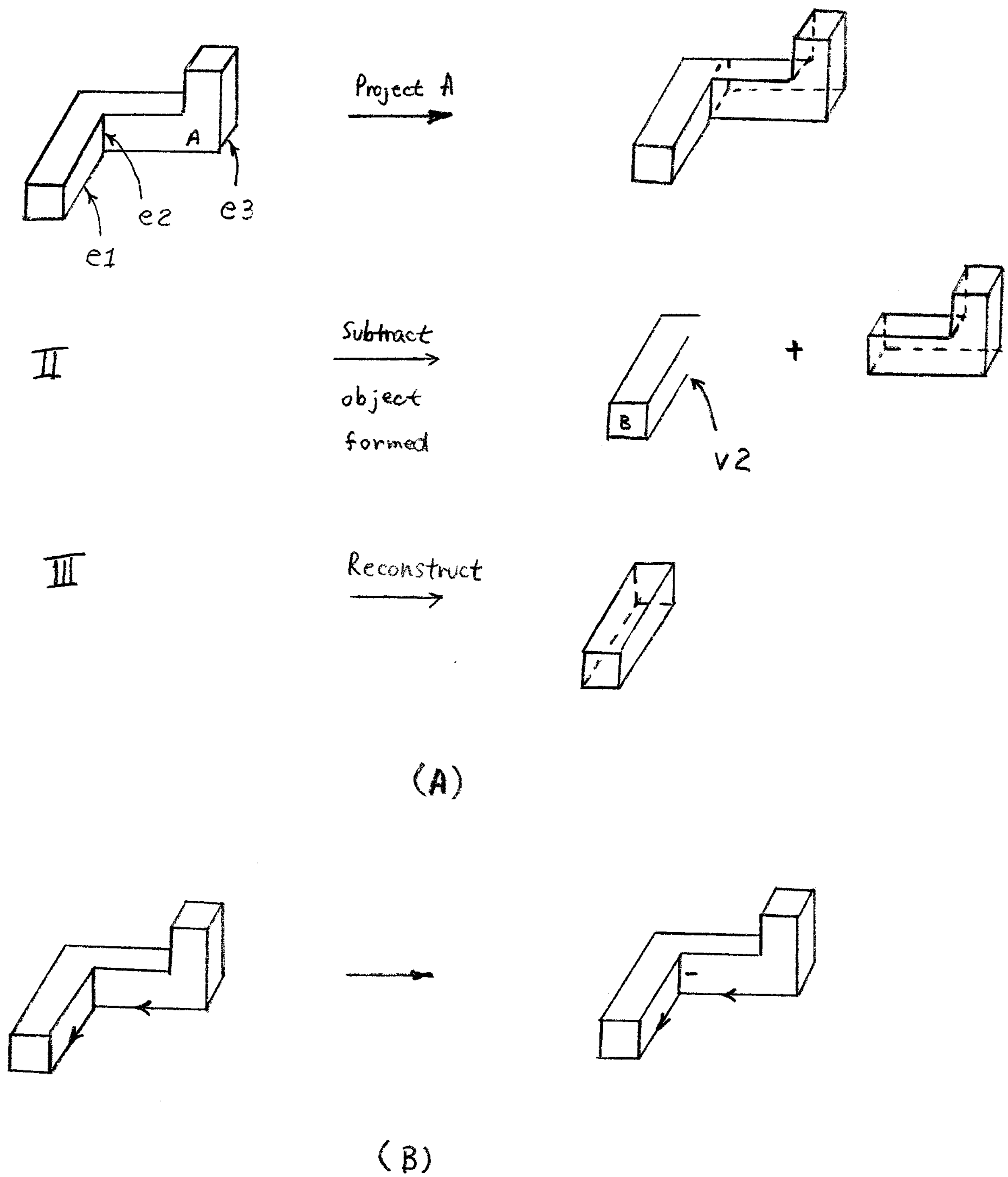

A
e1
e2
e3
Project A
II
Subtract
object
formed
B
v2
+
III
Reconstruct
(A)
(B)

Fig. 6

There are two aspects to this construction. It must first be decided what part to carve out of the object. Then the remainder of the object must be reconstructed for further processing.

2.2.1. The Hacksaw

Generally speaking, the carving process proceeds by attempting to remove the most complex part that can be described as the projection of a plane. The determination of this part can be done by ordering planes in a list in terms of decreasing complexity, and by marching down the list until a plane is found that by its projection is useful in splitting up the object. Plane A in Fig. 6A, for example, is the most complex plane, and its projection as indicated is a good way to carve up the object.

For a plane to qualify as useful, the necessary requirement is that its rays be parallel. When this condition holds, the plane will be said to be projectable. How far to project the plane is the next question, and is usually determined by the shortest ray that is not the shaft of a T. Thus in Fig. 7 the projection of plane A in Step I is limited by edge e1. The same holds for plane B and edge e2 in step IV. (The meaning of the remaining steps will become clear in the course of this section.) The distance of projection may be limited by a concavity in the object, as indicated in Fig.8. Whether the distance of projection is limited by the shortest ray or by

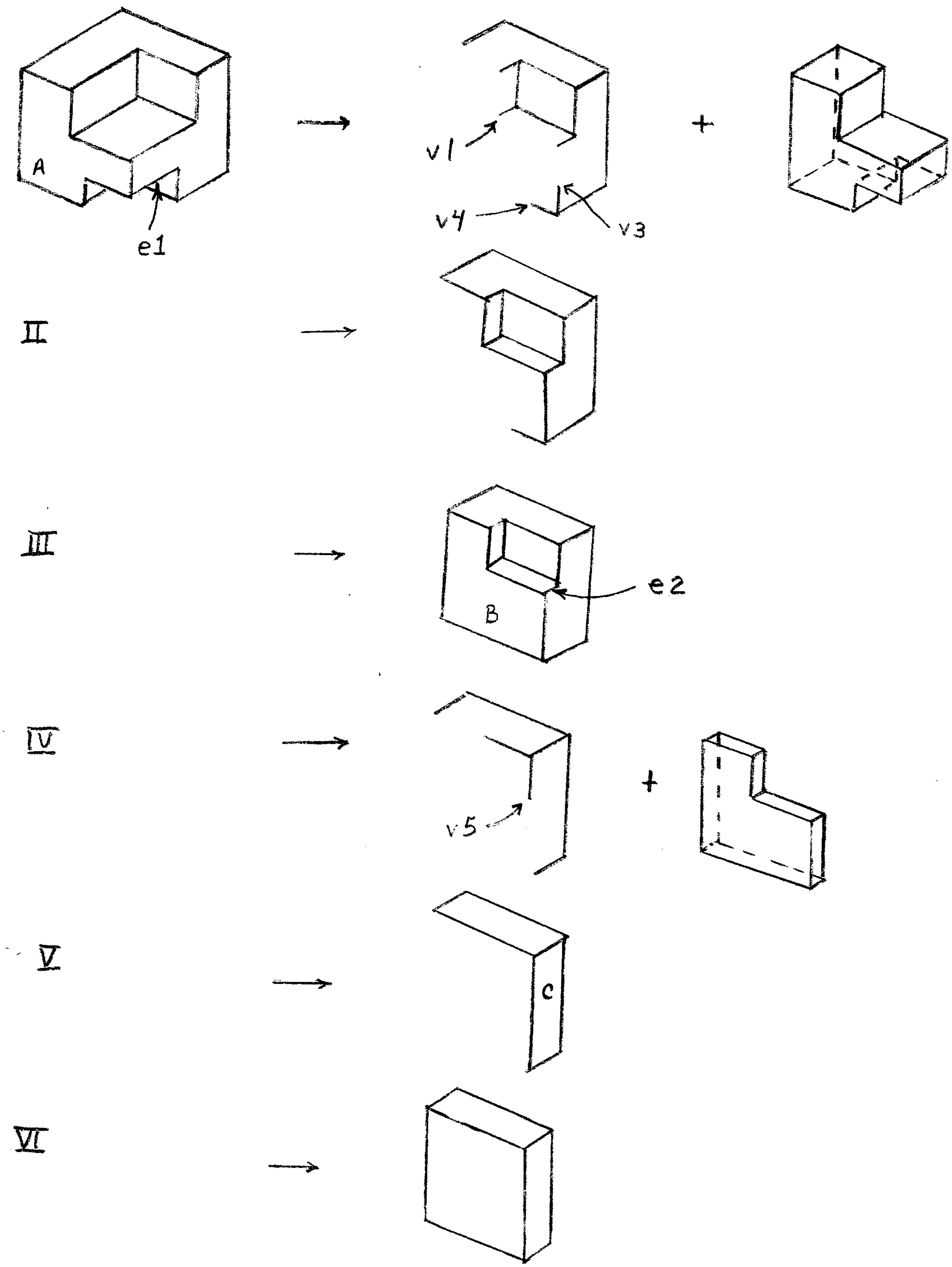


Fig. 7

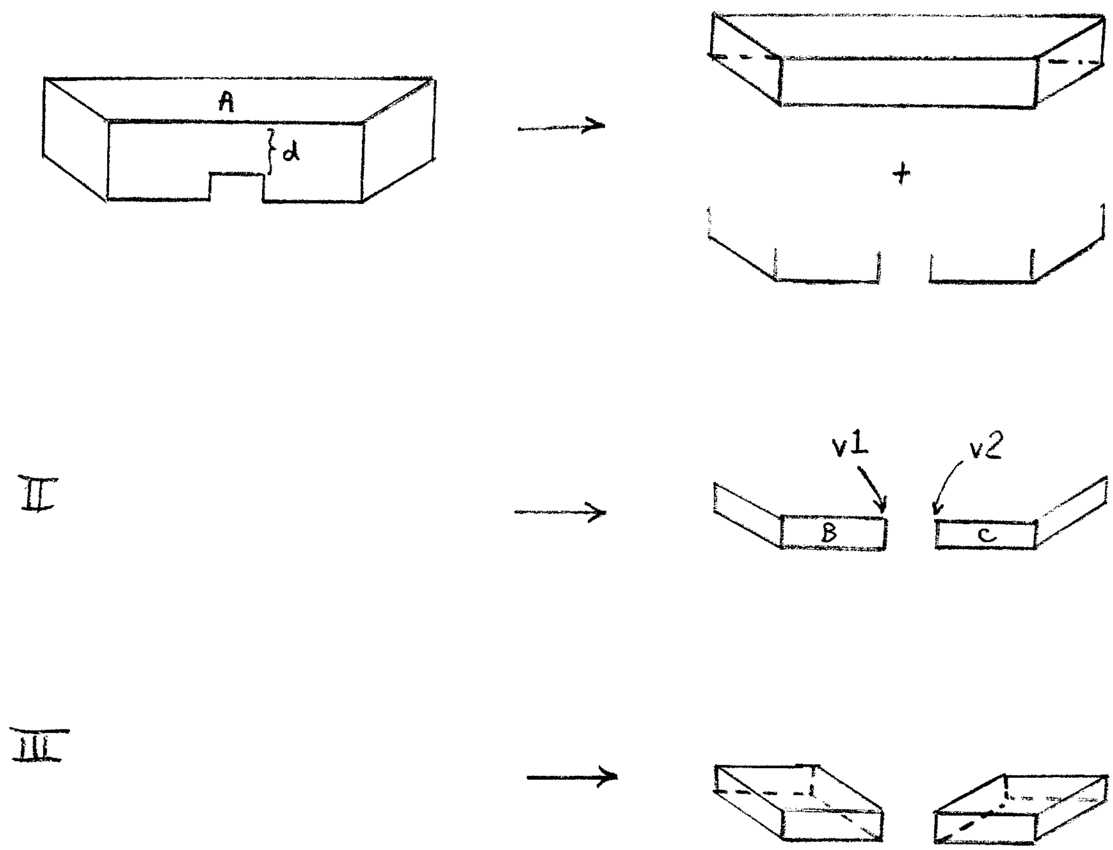


Fig. 8

a concavity depends of course on which of the distances is smaller.

Though a plane be projectable, it is not always immediately obvious in which direction to project it. This circumstance occurs when the rays of the plane are parallel and some of them are in opposite directions; e.g.,plane A in Fig. 6A. What determines the direction of projection is the concavity of the object. The general rule is that projection is along a ray that has the characteristic that neither of the two edges of the projectable plane that form a vertex with the ray are concave. Thus ray e1 of Fig. 6A does not dictate the direction of projection because edge e2 of plane A is concave. Ray e3, however, satisfies the rule and consequently determines the direction of projection. An edge can be determined to be concave by selective Huffman labeling (4); that is, not all edges of the object need receive a Huffman label, but only those that are necessary to determine the label of the edge in question. The selective Huffman labeling of edge e2 could proceed, for example, as in Fig. 6B.

Not all projectable planes can be profitably used to carve up a compound object. It may happen that object reconstruction is rendered impossible by rules to be described later. It is then necessary to try the next plane on the list of planes ordered by decreasing complexity. If all planes on the list have been unsuccessfully used to carve up the object, then

Propa-pH fails. The backup facility of PLANNER is evidently well suited to this type of procedure. Consequently, explicit PLANNER theorems are given in section 3.2 that conduct the carving up process in the desired manner.

In some situations a more intelligent choice can be made regarding what projectable plane to choose. One such situation is illustrated in Fig. 9A, where plane A has equal length rays in the same direction. The object formed by projection of A (hereafter referred to as the projected object) does not cause a visible plane of the compound object to be cut or dissected when it is removed. The projected object is thus more in the line of a protrusion, and is naturally removed first to decrease the complexity of the compound object. Without looking for this situation, the compound object construction could proceed as in Fig. 9C, which is decidedly less aesthetic than that of Fig. 9B.

A slightly different situation from the previous one is indicated in Fig. 10, where the object formed by projection of plane A does cause a visible plane to be cut. One would not exactly call the projected object a protrusion, but there is some aura of sticking-outness about it. In Propa-pH, this situation is checked for after the previous one. How this all fits together is made clearer by the PLANNER theorems in section 3.2. It should be noted that the construction in Fig. 10 is the only successful way of describing the object under the limitation

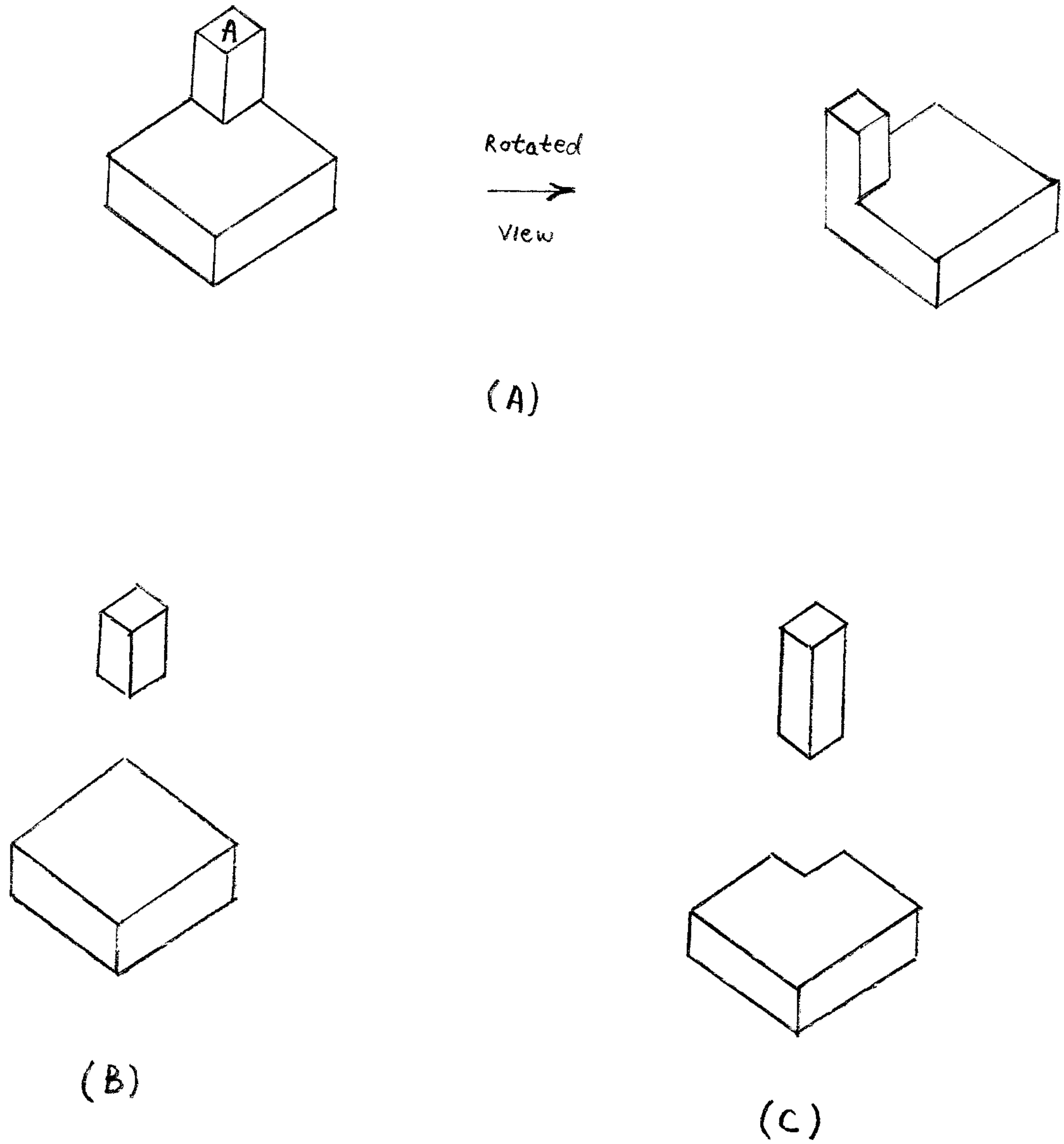


Fig. 9

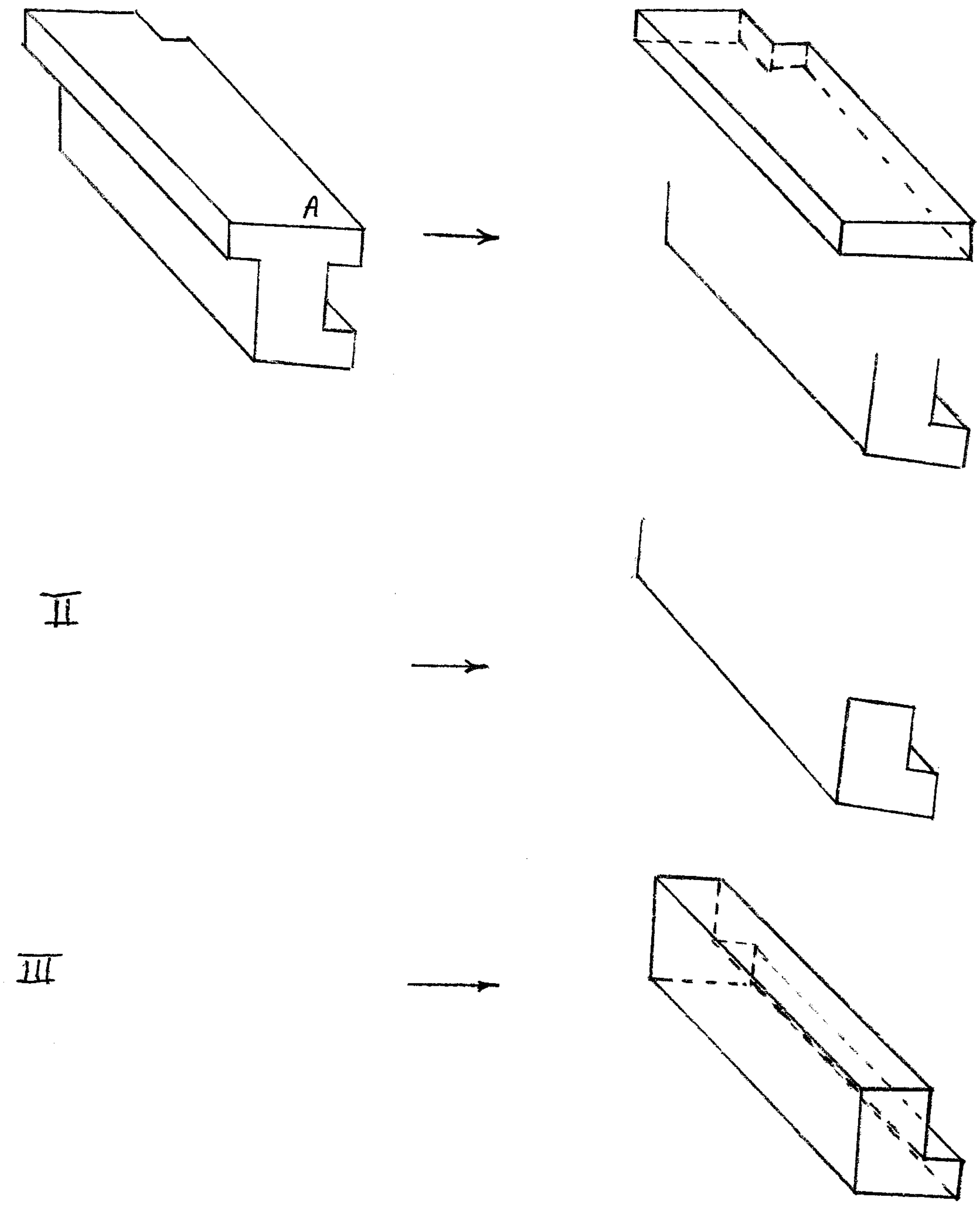


Fig. 10

of the reconsturction rules. Thus the choice of plane A was particularly a good one.

2.2.2. Reconstruction

There are two basic approaches to the reconstruction of an object once a part has been cut out of it. The first approach is to add the hidden lines of the projected object and to concoct some method of erasing the unnecessary lines or line parts. The second one is to erase the projected object entirely and to add lines as dictated by the remainder of the object and the projected object. The first approach fails to take into account restrictions imposed by the remains of the compound object; in particular, no new lines can be created. For the two nearly identical objects in Fig. 11, the reconstruction of the remains is almost entirely dictated by the shape of the remains and not by the shape of the object cut out. Both objects are included in the figure to accentuate the hopelessness of the line erasure method.

To utilize the line addition method, the dangling line segments must somehow be handled. It has been chosen to consider the ends of these lines as vertices, and to label these vertices with regard to type by how they were formed. There are six types of these vertices.

(1) A CUTEDGE is formed from a visible edge that has been cut or dissected by removal of the object: e.g. v1 in Fig. 7.

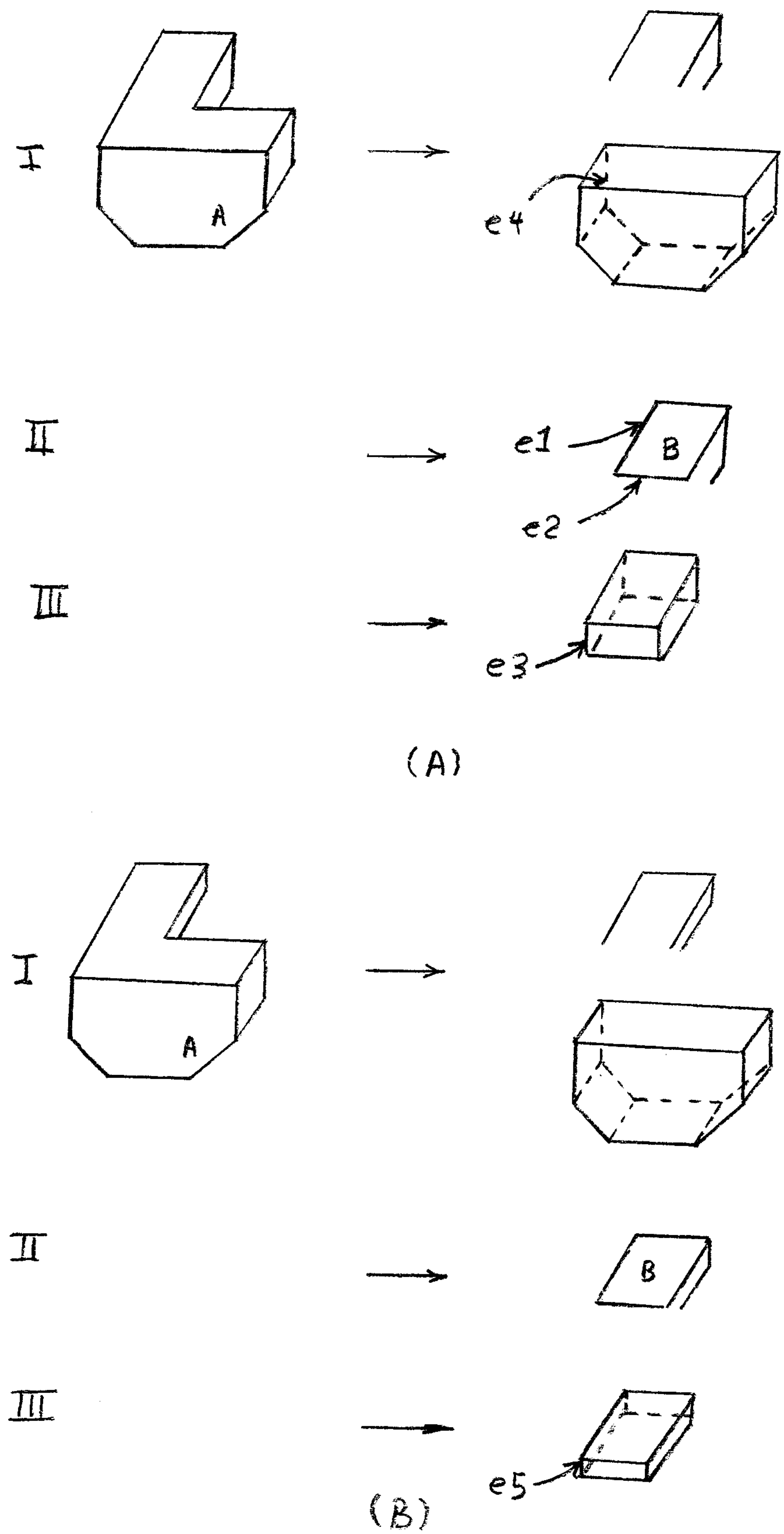


Fig. 11

(2) An XRAY is formed from a ray of the projected plane that points in a direction opposite to the projection; e.g., v2 in Fig. 6.

(3) A CUTL is formed from a dissembled L vertex; e.g., v3 in Fig. 7.

(4) A CUTT is formed from a dissembled T vertex; e.g., v4 in Fig. 7.

(5) A CUTARFO is formed from a dissembled arrow or fork vertex; e.g., v5 in Fig. 7.

(6) A CUTARFOT is formed by removing only one edge from an arrow or fork vertex; e.g., v1 in Fig. 20. This vertex differs from the previous five in that it initially has two edges.

The reason for this particular choice of categories has to do with developing criteria for determining when a reconstruction is complete. Explicit statements can be made about vertices in a particular category with regard to how many visible edges each vertex has in the reconstructed object and how long and in what direction the edges of each vertex are. Each category will now be investigated with regard to this characterization.

A CUTEDGE vertex has the following characteristics in the reconstruction process.

(1) All three edges are visible.

(2) Two of the edges are determined by the projected object.

(3) The direction, but not the length, of the third edge is determined by the projected object.

One edge that is obviously determined is the edge that was cut. A second edge falls on the edge of the projected object that produced the CUTEDGE by dissecting a plane (the latter is henceforth called the cutting edge). Its length is determined either by the shortest extension to a one-edged vertex (e.g., e2 in Fig. 11A), or by the length of the corresponding edge of the projected plane, whichever is shorter. The latter possibility is indicated by edges e1 and e2 in Fig. 12. Note that the object in Fig. 12 cannot be constructed using as one piece the projected object formed by plane A, because of limitations of the reconstruction rules. If plane B is chosen instead, the construction succeeds. Nevertheless, edge e1 is properly constructed.

The direction of the third edge is dictated by the projected plane because the CUTEDGE is a trihedral angle. The length, however, is not determined by the projected object; e.g., compare e3 and e5 in Figs. 11A and 11B.

The following statements can be made about XRAYs.

(1) The rays corresponding to the XRAYs must be extended at least the distance of projection. This forms new vertices.

(2) If there is a neighboring (with regard to the projected plane) XRAY that has formed an L, and if the

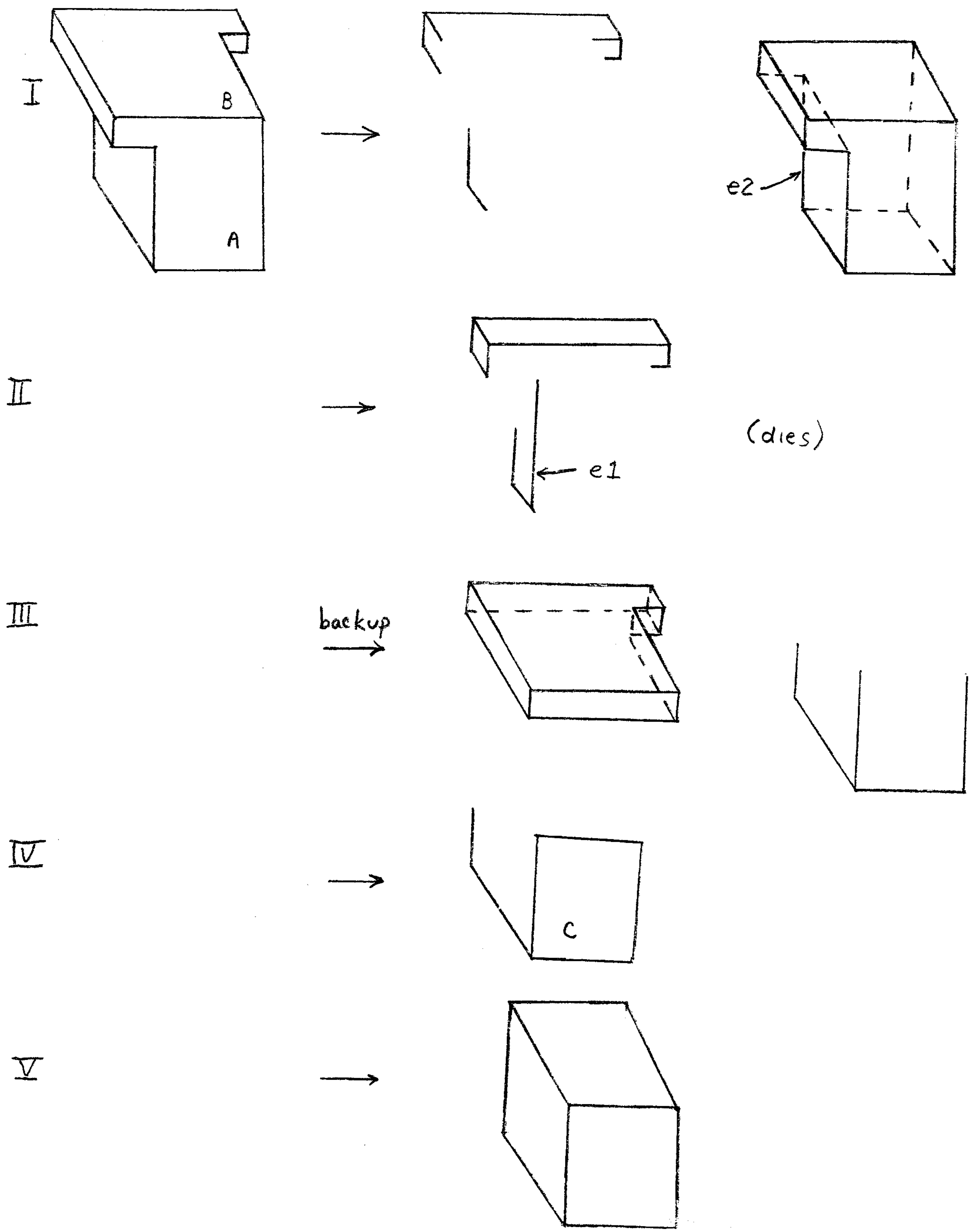


Fig. 12

connection of the extended XRAYs produces an arrow vertex from this L, of which the edge of the neighboring XRAY is the shaft; then the connection is a valid edge.

(3) If an extended XRAY has formed an L vertex, and if the corresponding ray is the edge of only one face of the original object; then there are only two visible edges of this vertex.

Properties (1) and (2) are illustrated in Fig. 13, where rays e1, e2 and e3 have first been extended in step II. The extended ray e2 is then connected to the two extended XRAYs in step III. Property (3) is illustrated in Fig. 14, where ray e1 is eventually extended to form an L vertex.

Property (2) is really all that can be said about a connection between neighboring XRAYs. In other circumstances this connection may not exist. For example, in Fig. 15 the extended XRAYs formed from rays e1 and e2 may or may not connect, as is indicated by two distinct reconstructions in steps IIIa and IIIb. There is no a priori reason to select one of these over the other, and indeed the equivalent of the object in step IIIb is eventually constructed by the present rules. What is meant by the latter statement is that the reconstruction does not succeed when begun by projecting plane A, but by projecting plane B. The eventual reconstruction is compatible with the object in step IIIb

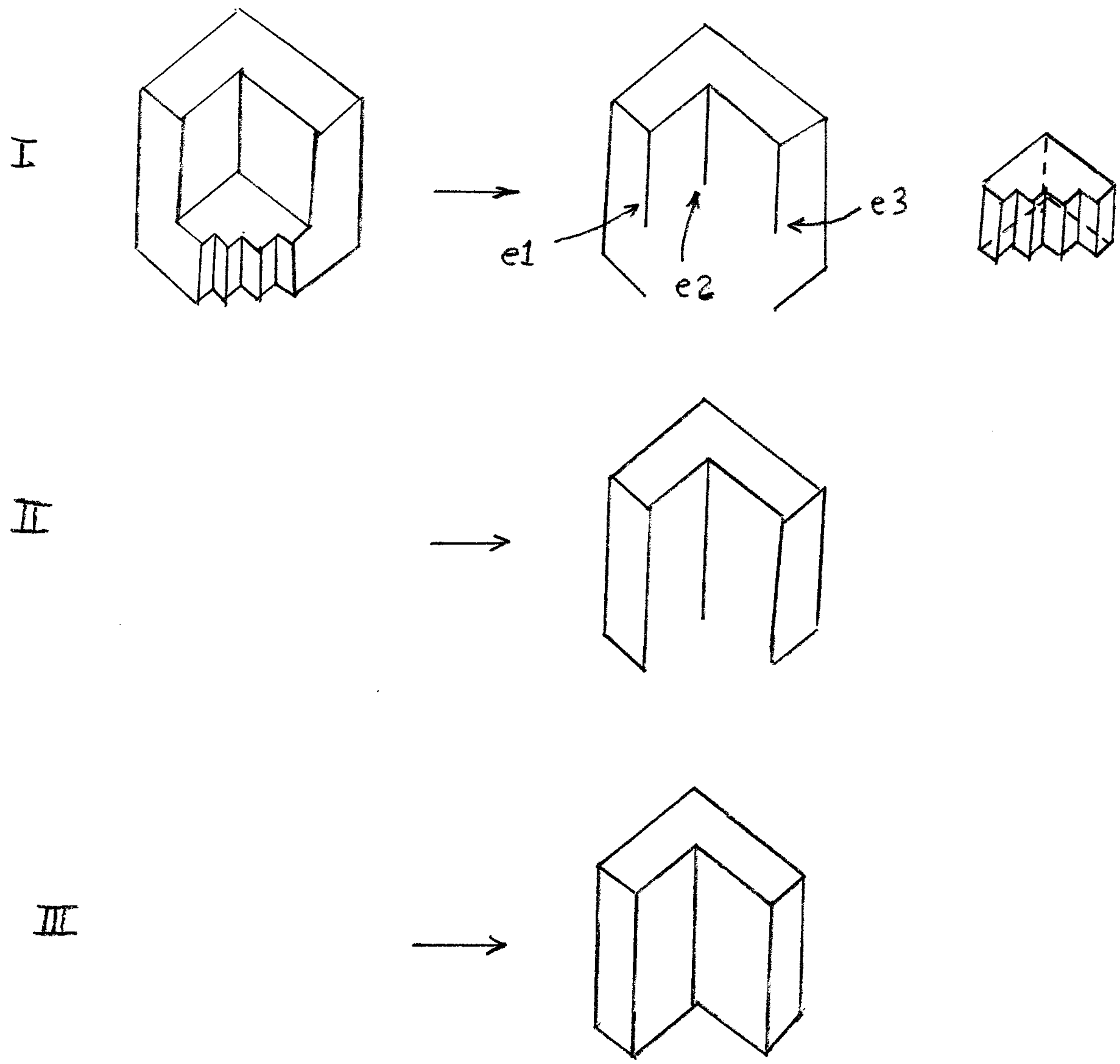


Fig. 13

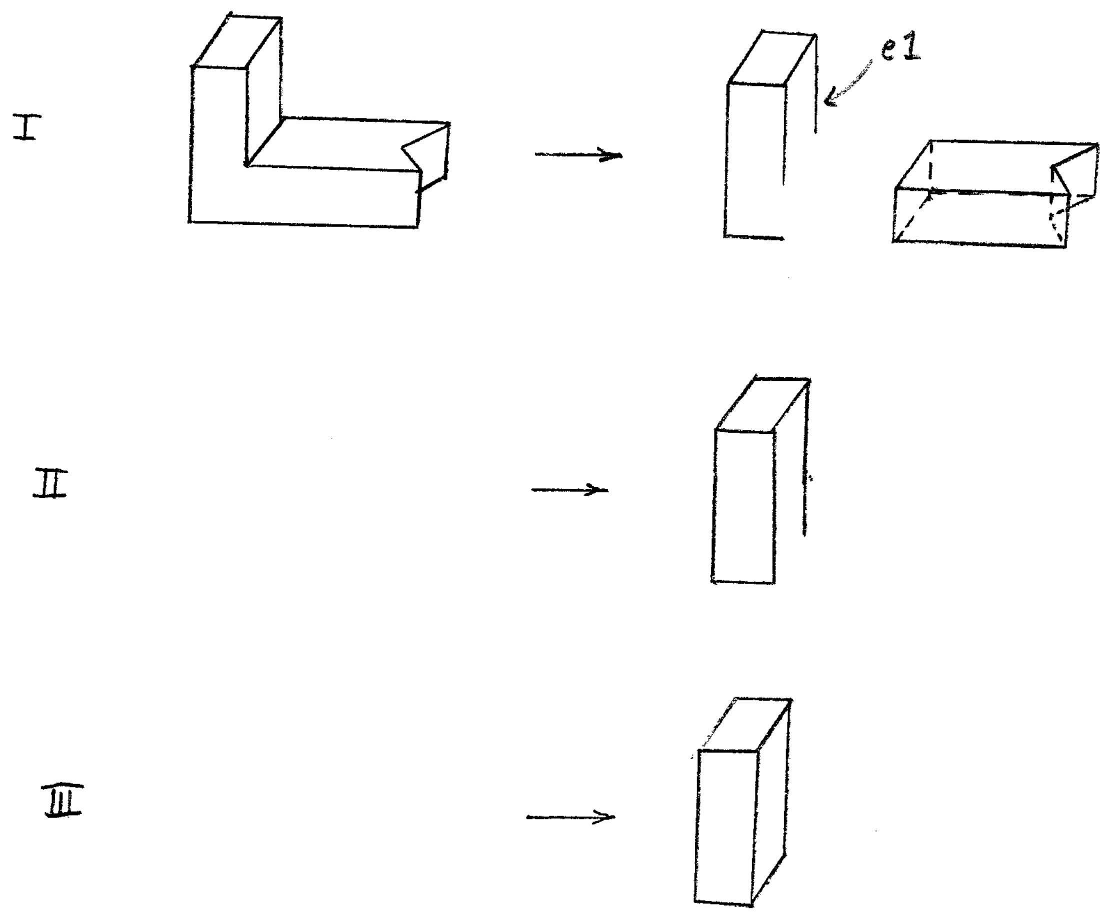


Fig. 14

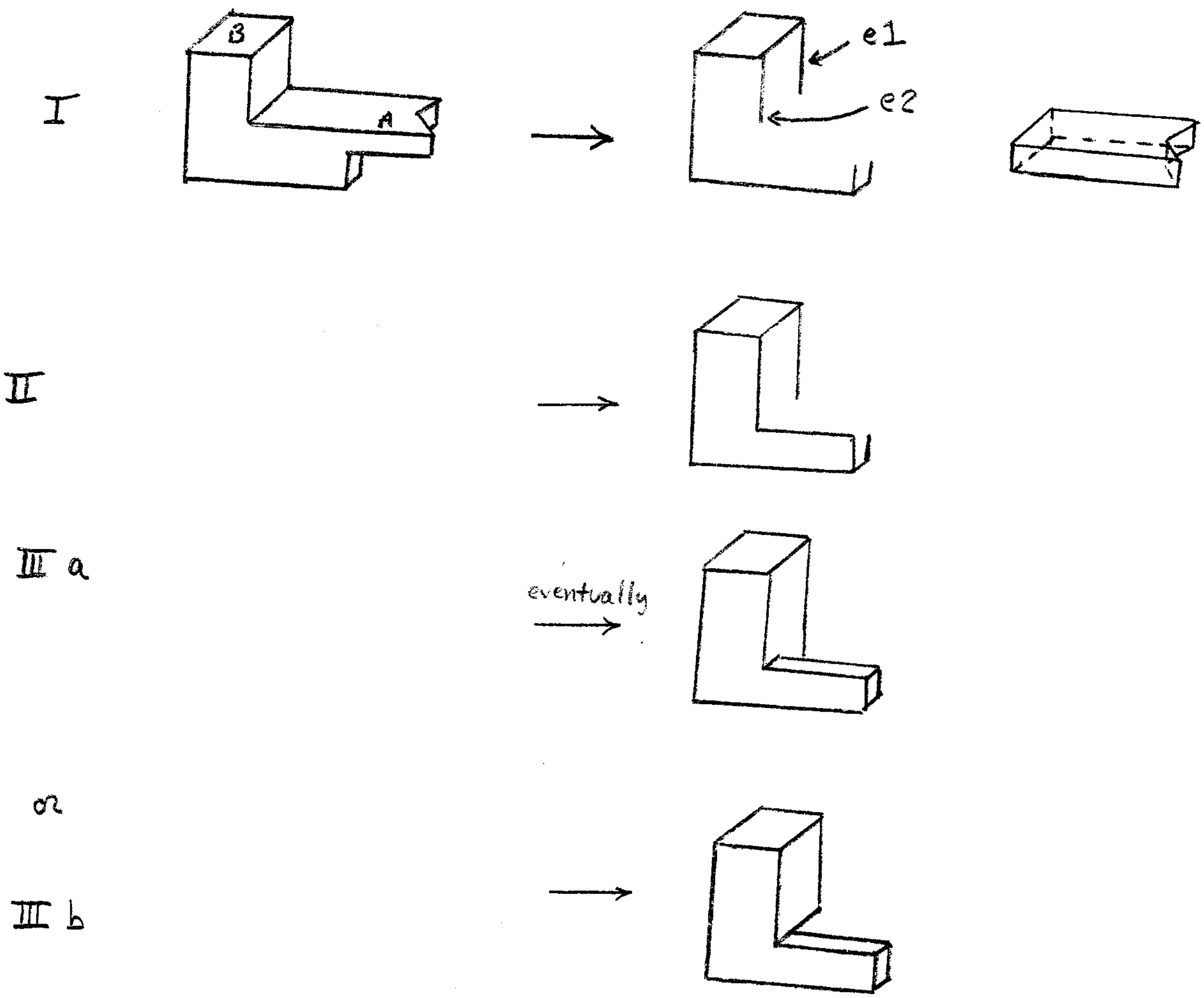


Fig. 15

rather than the one in IIIa (see Fig. 16). Nevertheless, the point remains that property (2) is the only determinate condition for connection.

This discussion brings out an important point; namely, there is not always a unique reconstruction. An impact of this observation is discussed in section 4.

Finally, property (3) depends strongly on property (2) limiting connections between extended XRAYs. Without this limitation, it is easy to construct a counterexample.

A CUTL has the following characteristics.

(1) If a CUTL does not disappear by coincidence of its edge with the cutting edge, i.e., it is a true vertex of the reconstruction, then two of the edges are determined by the projected object.

(2) If originally an edge of the L had a concave labeling, the CUTL will have exactly two visible edges.

(3) Otherwise, the CUTL has three visible edges, but neither the direction nor the length of the third edge need by specified by the projected object.

A situation where a CUTL vertex disappears by coincidence with the cutting edge is illustrated by vertex v3 in Fig. 7. If the CUTL is a true vertex of the reconstructed object, two of the edges are determined in the same way as the two of a CUTEDGE vertex. Property (2) is illustrated by vertex v1 in Fig. 17. Property (3) obtains because one edge is

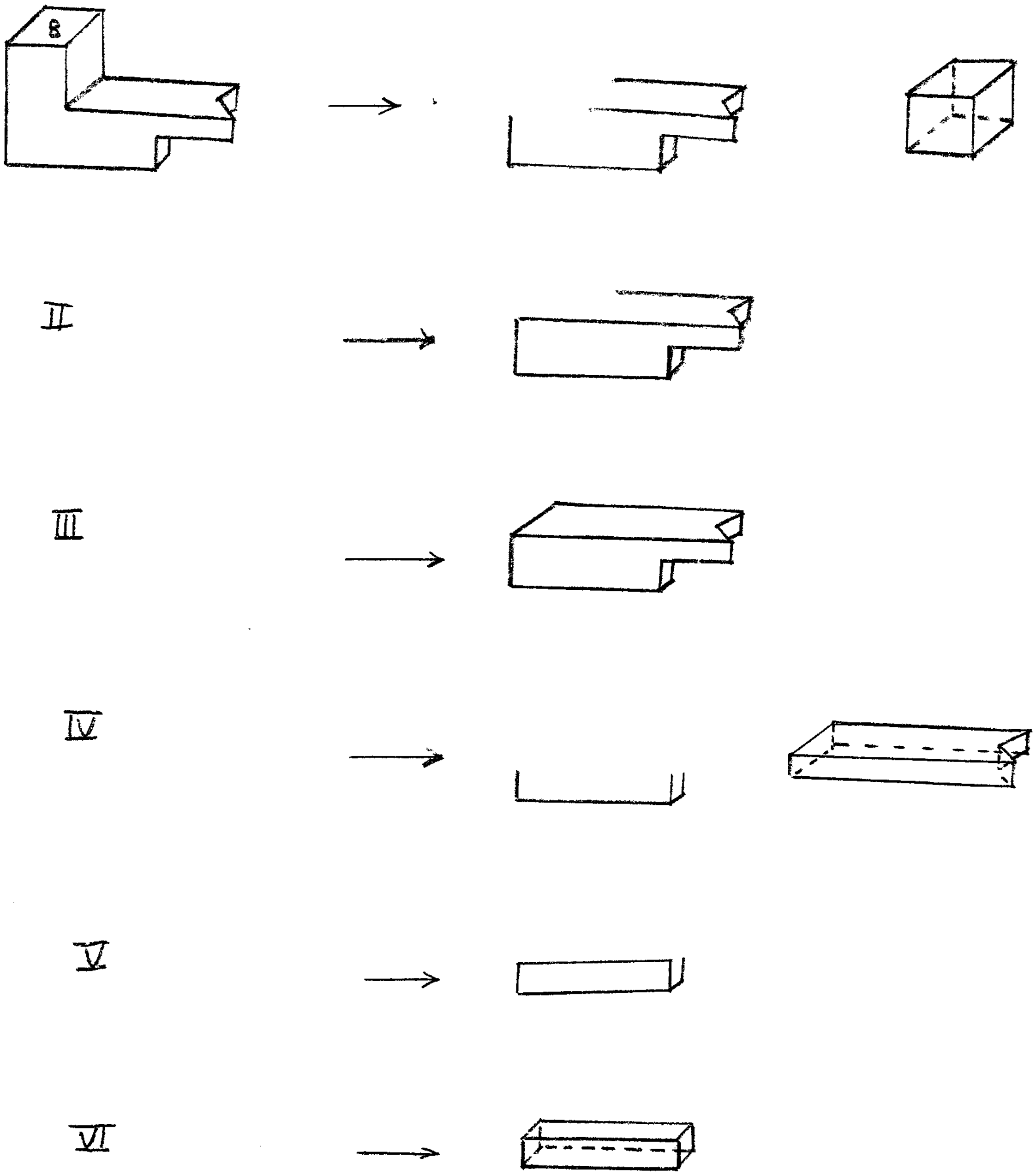


Fig. 16

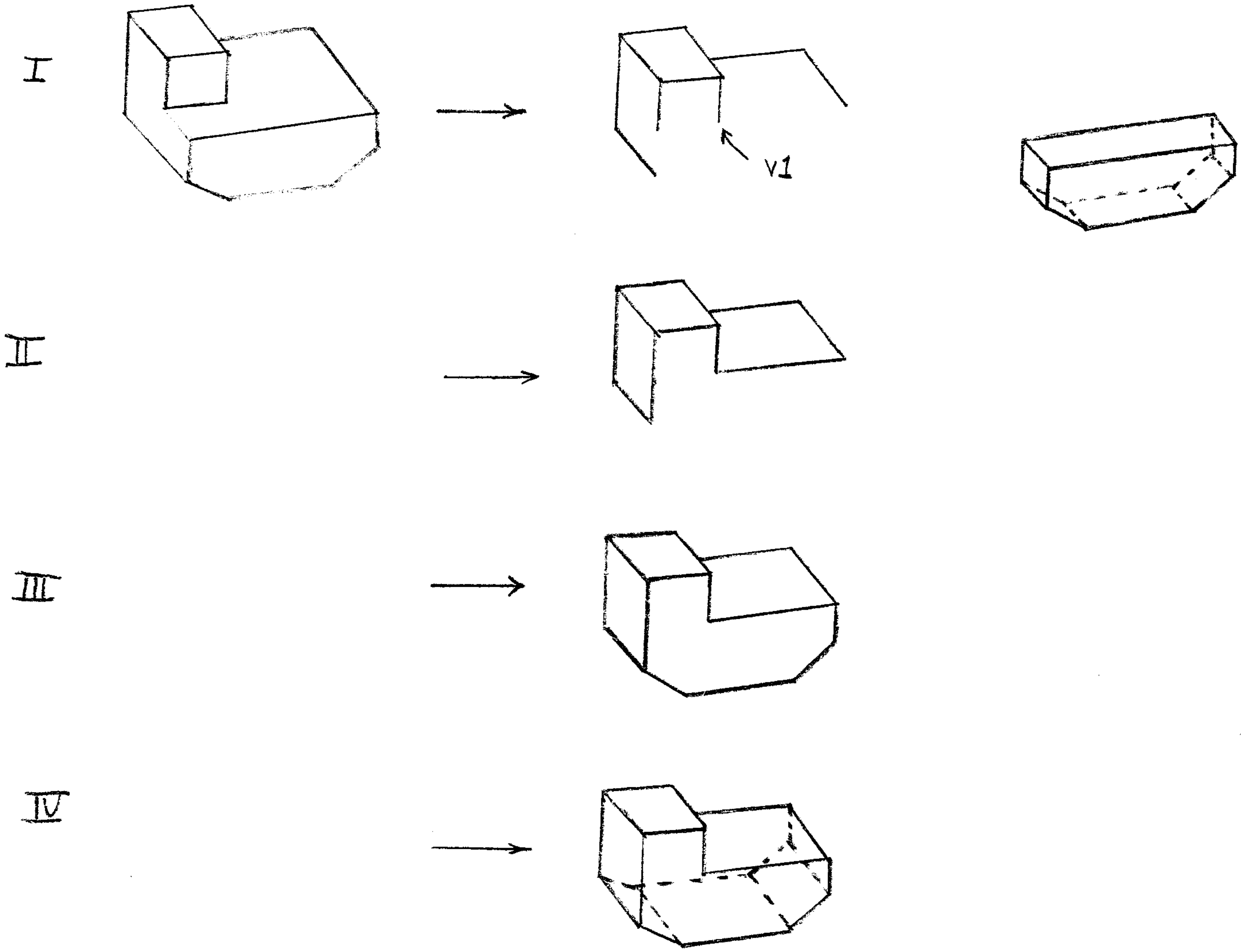


Fig. 17

merely constrained to lie somewhere on the appropriate plane of the projected object. That the length and direction are indeterminate is illustrated in Fig. 18A, which shows a valid alternative to the reconstruction of step III in Fig. 8. Indeed, there is a whole spectrum of possible objects between those in Fig. 18A and the corresponding objects in Fig. 18B, dependent only on the direction and length of the third edges of CUTLs v1 and v2 in Fig. 8.

Nothing much can be said about CUTTs. Under the assumptions of section 3.1, it is not possible for CUTTs to be permanent vertices in the reconstruction. Other rules of reconstruction must be relied on to handle CUTTs properly.

CUTARFOs have the following properties.

(1) If a CUTARFO does not disappear by simple extension, then it has three visible edges.

(2) The edges of a CUTARFO are completely specified by the projected object.

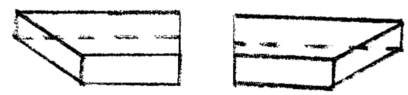
An example of a CUTARFO that disappears is v5 in Fig. 7, whereas an example of a three-edged CUTARFO is vertex v1 in Fig. 19.

Finally, the following can be stated about CUTARFOTs.

(1) A CUTARFOT has three visible edges.

(2) The third edge is completely specified by the projected object.

An example of the reconstruction of a CUTARFOT is given by vertex v1 in Fig. 20.



(A)

(B)

Fig. 18

I

II

III

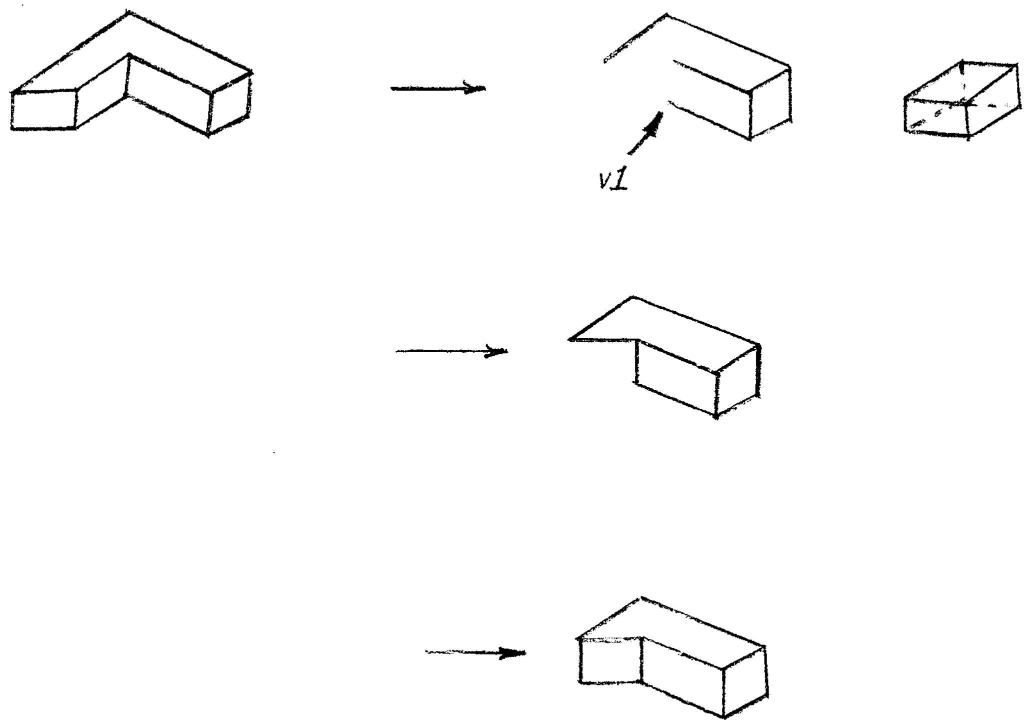


Fig. 19

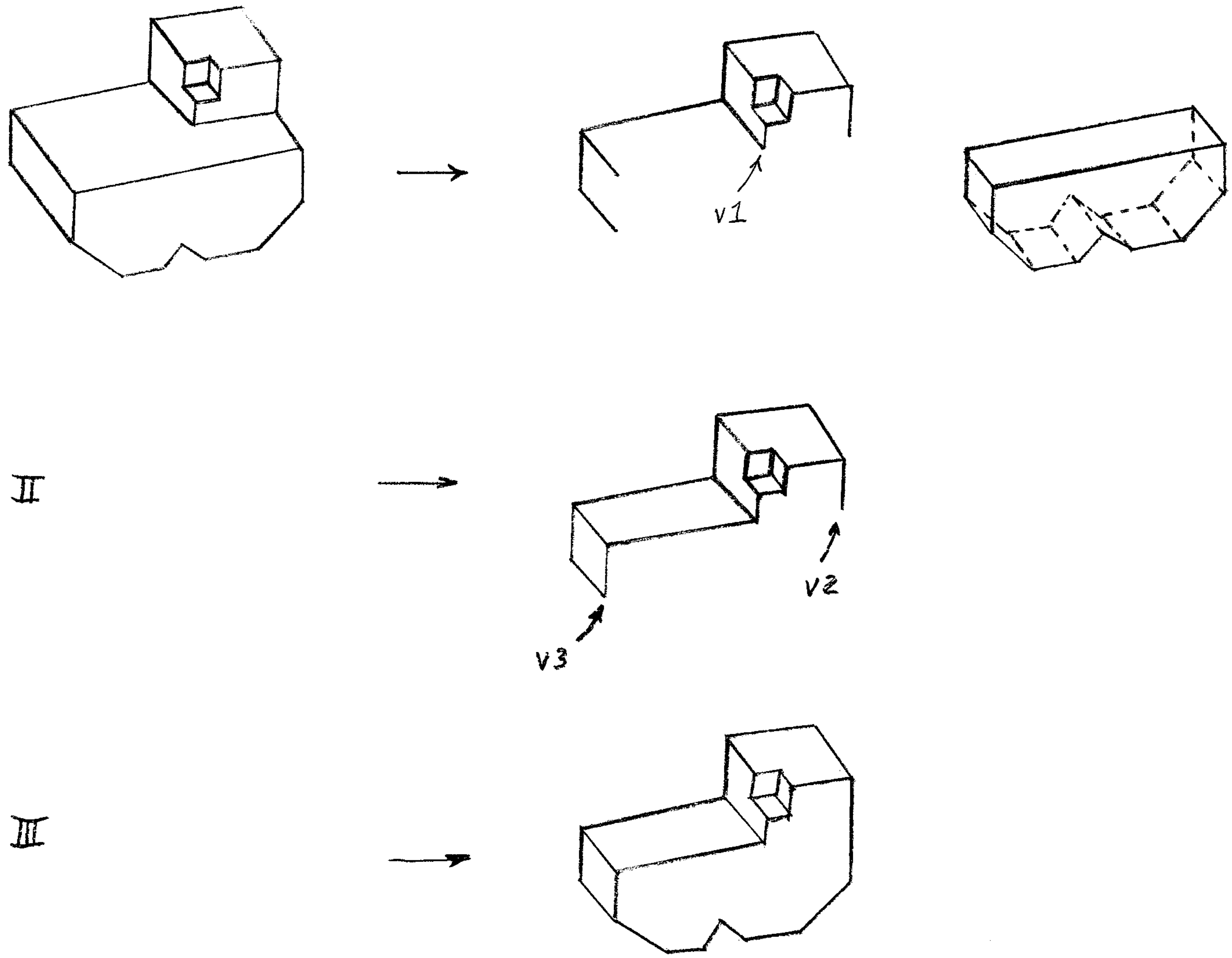


Fig. 20

How the various properties of the different vertex types fit into a reconstruction scheme is discussed in detail in section 3.2. It can be mentioned beforehand that whatever edge reconstructions are completely determined by the vertex properties are initially carried out. On the basis of the number of visible edges each vertex must have, the reconstruction is or is not judged complete. If the afore-mentioned preliminary edge reconstructions do not complete the object remains, then two different heuristics are applied. One of these involves Propagation Rules when only two vertices remain to be completely specified, and the other heuristic involves a restricted application of Propa-pH to the object remains. In the previous Figures, when two steps were required for object reconstruction, unless otherwise indicated the first step was application of the preliminary edge reconstructions, whereas the second was application of one of these heuristics.

It quite often happens that the object remains can be described as the projection of one of its planes. These remains do not necessarily have to be complete to apply Propa-pH to them. All that is required is that there be a projectable plane that can be projected in such a way as to completely specify the remains. There must exist some ray of this plane, the opposite end of which is a permanent vertex, to give the distance and direction of projection. To avoid a great deal of trouble with too many unspecified vertices, a compound object reconstruction is not attempted if a projected plane can specify only part of the remains. Thus either the projected plane can completely specify the remains, or the heuristic fails.

This heuristic is quite powerful, and has been used throughout the examples. A list of its application is given below.

Fig.	Step
6A	III
8	III
10	III
11A	III
11B	III
16	VII

A generalization of this heuristic applies to situations in which the remains are a disconnected collection of connected parts. If the simple heuristic does not work for the remains as an ensemble, then it is applied to each of the parts. To succeed, the heuristic must succeed on each part. An example of the generalized heuristic in use may be seen in step III, Fig. 8.

The other heuristic, called the Propagation Rule, applies only when two vertices are left to be completed. Under certain conditions, a connected portion of the border of some plane of the projected object is a valid border for the reconstructed object. The extent of this border portion is determined by the two uncompleted vertices. In a sense, it can be said that the edges of the border are propagated between these vertices. To make this concrete, notice that in step III of Fig. 20 a border portion of the projected object is propagated between vertices v2 and v3.

There are a number of restrictions on the application of the Propagation Rule. Not all types of vertex pairs have been found to yield valid results upon application of this Rule. Those pairs that are amenable to it are listed below, together with an example of its application to each.

Vertex Pair	Example
CUTARFO-CUTARFO	Fig. 21, Step II
CUTARFO-CUTEDGE	Fig. 20, Step III
CUTARFO-CUTT	Fig. 22, Step III
CUTEDGE-CUTEDGE	Fig. 17, Step III
CUTEDGE-CUTT	Fig. 7, Step III
CUTT-XRAY	Fig. 23, Step III

The absence of CUTL vertices from the list is noteworthy. The elusive third edge is the spoiler, in that too much variability is present for determinate propagation of edges between a CUTL and some other vertex. Similarly, not enough information is normally present to allow the Propagation Rule to complete a CUTT-CUTT pair.

A great deal is known beforehand about CUTARFOs and CUTEDGEs, and accounts for the predominance of these two vertex

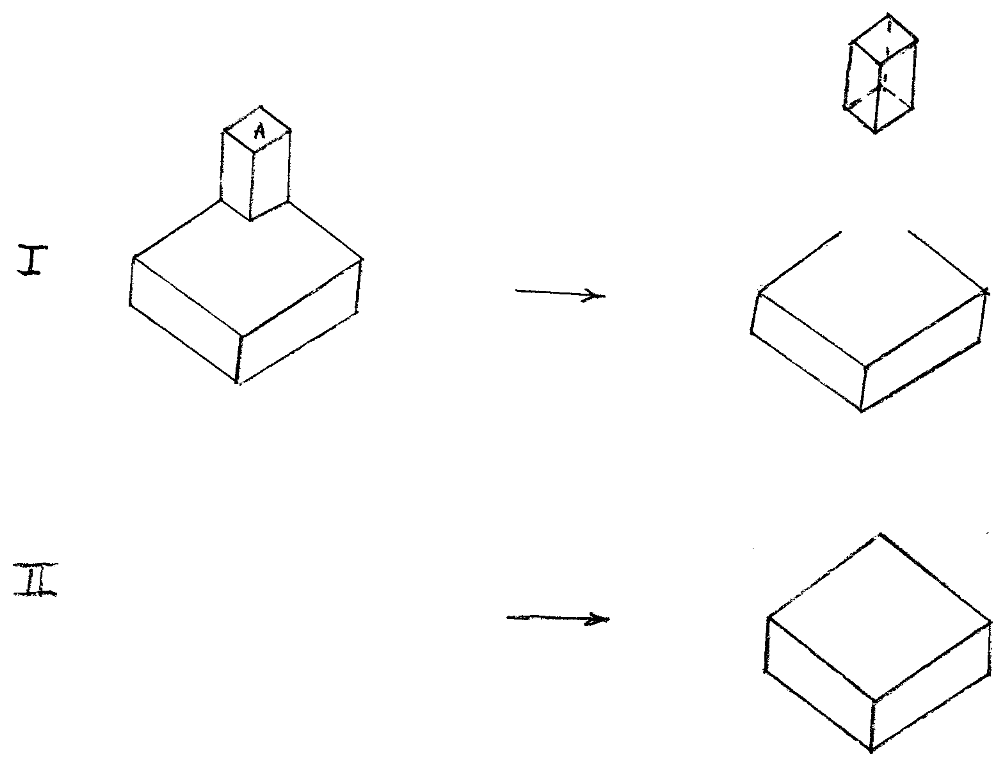


Fig. 21

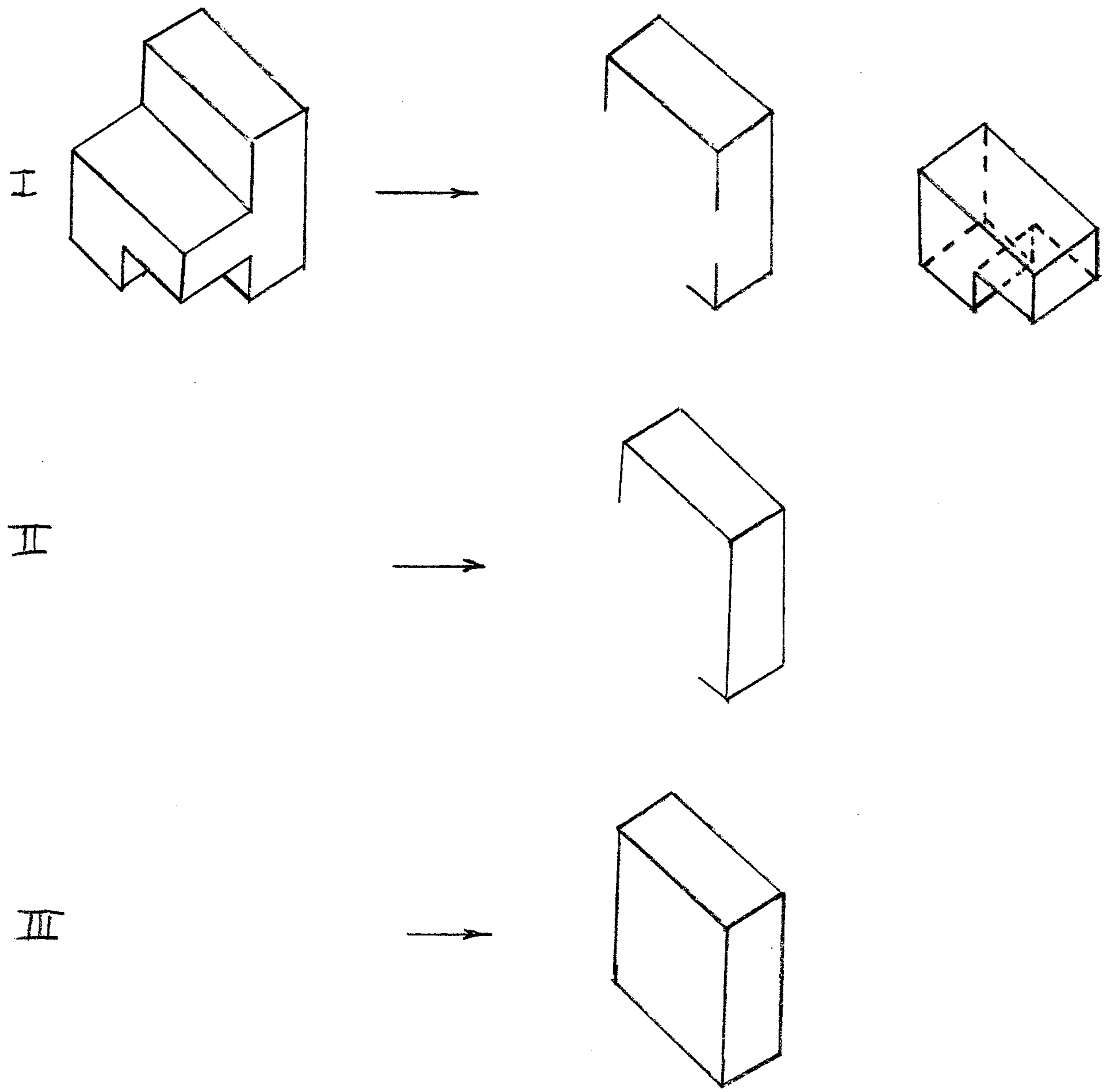


Fig. 22

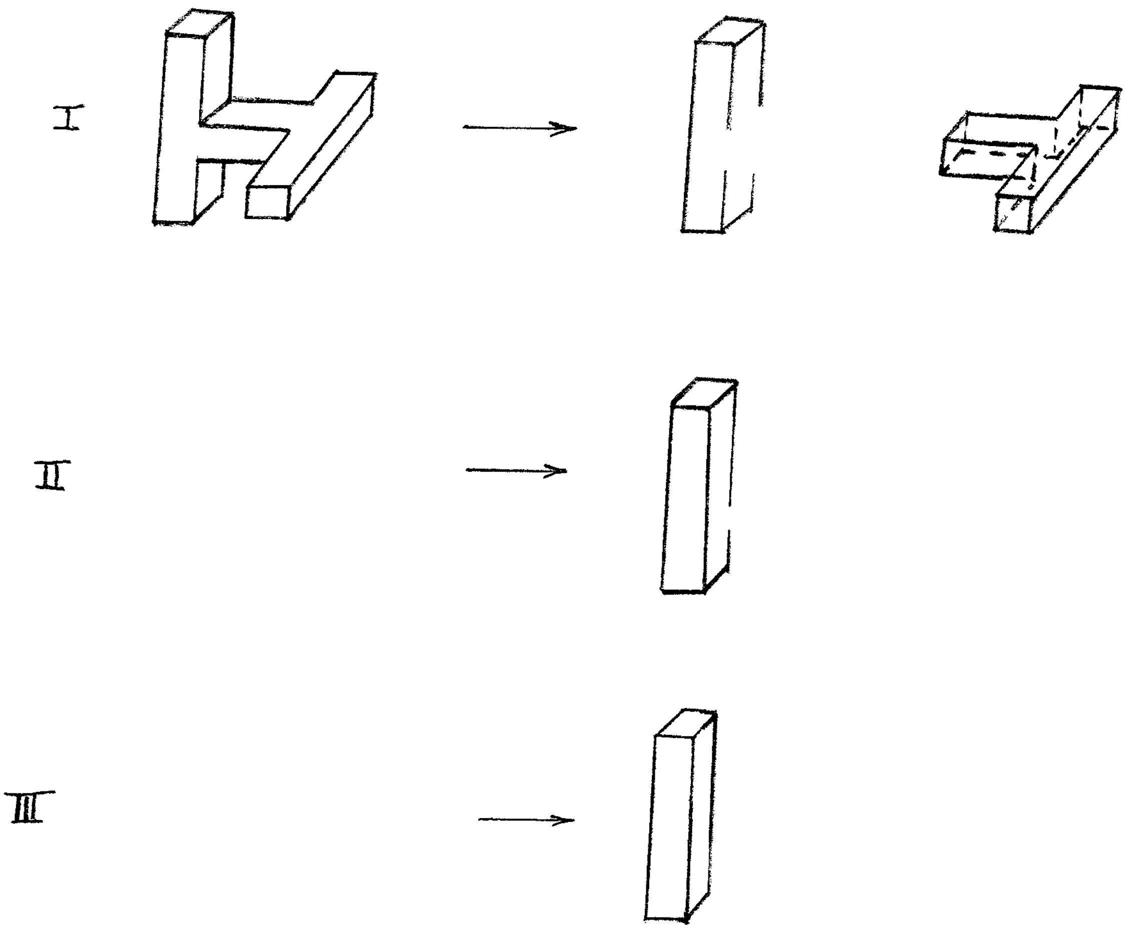


Fig. 23

types in the list. Any single CUTARFO or CUTEDGE suffices to determine which border portion on which plane of the projected object the Propagation Rule applies to. For the CUTEDGE, the plane in question is determined by the two edges that were not present originally. Remember that one edge is completely specified and the second lies along an edge of the projected object, though its length is not generally known. By following the second edge to the other uncompleted vertex, the proper border portion is obtained. For the CUTARFO, consider the corresponding vertex of the projected object. The edge to follow is the edge of this vertex that was not originally part of the object.

The propagation is deemed valid if each vertex along the border portion has the property that none of its edges or edge portions were present in the original object. This property is extremely important, and invalidates constructions that might otherwise yield nonsensical objects. For example, the presence of a portion of edge e1 in the original object invalidates the construction in step III of Fig. 24. Perhaps a more serious error occurs in step III of Fig. 25, which would otherwise be constrained by the visibility of edge e1 in that figure.

The handling of the CUTARFO-CUTT and CUTEDGE-CUTT vertex pairs is slightly modified from what the above discussion implies. A border portion cannot be propagated all the way from a CUTARFO or CUTEDGE to a CUTT. The propagation is

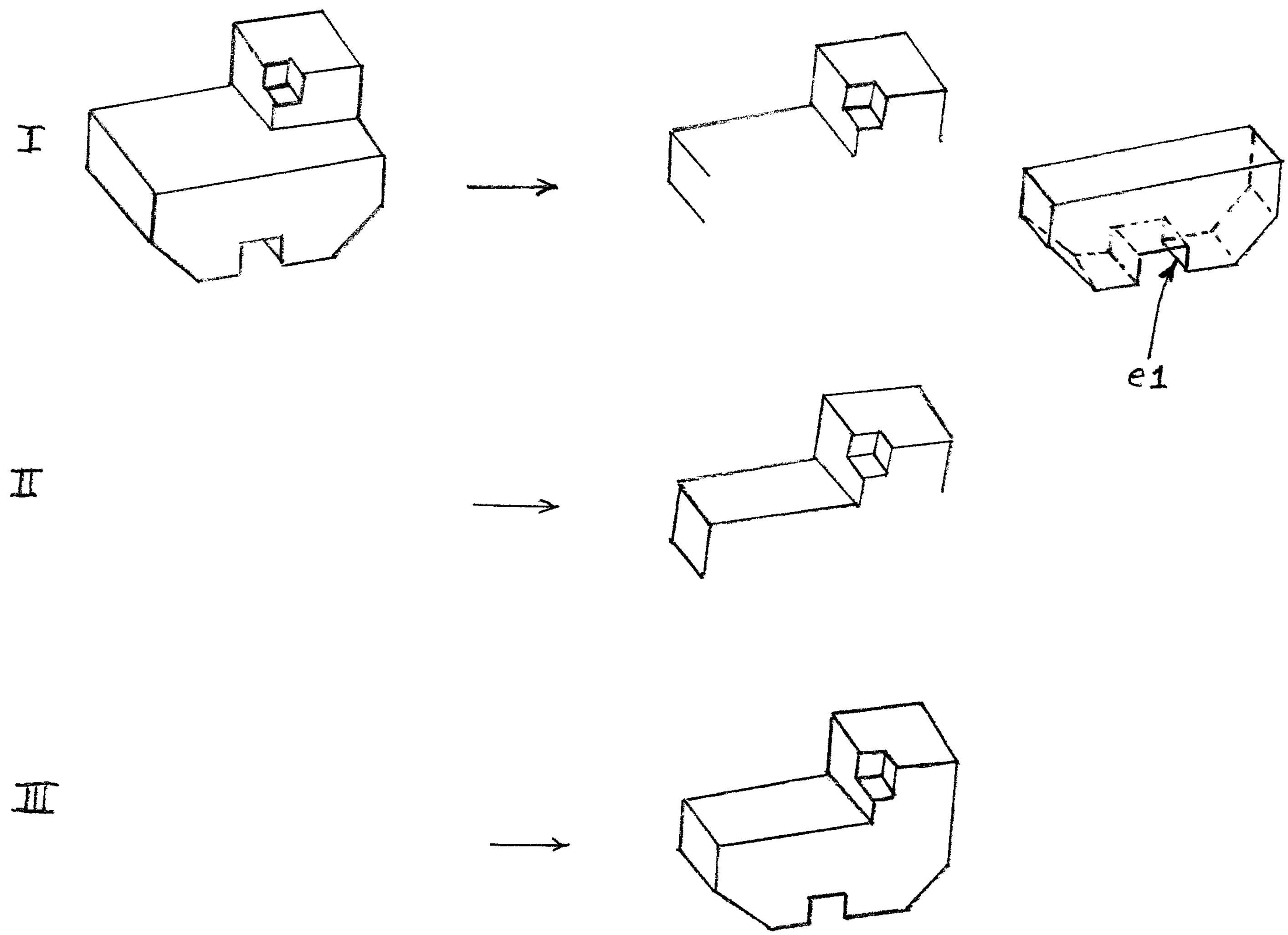


Fig. 24

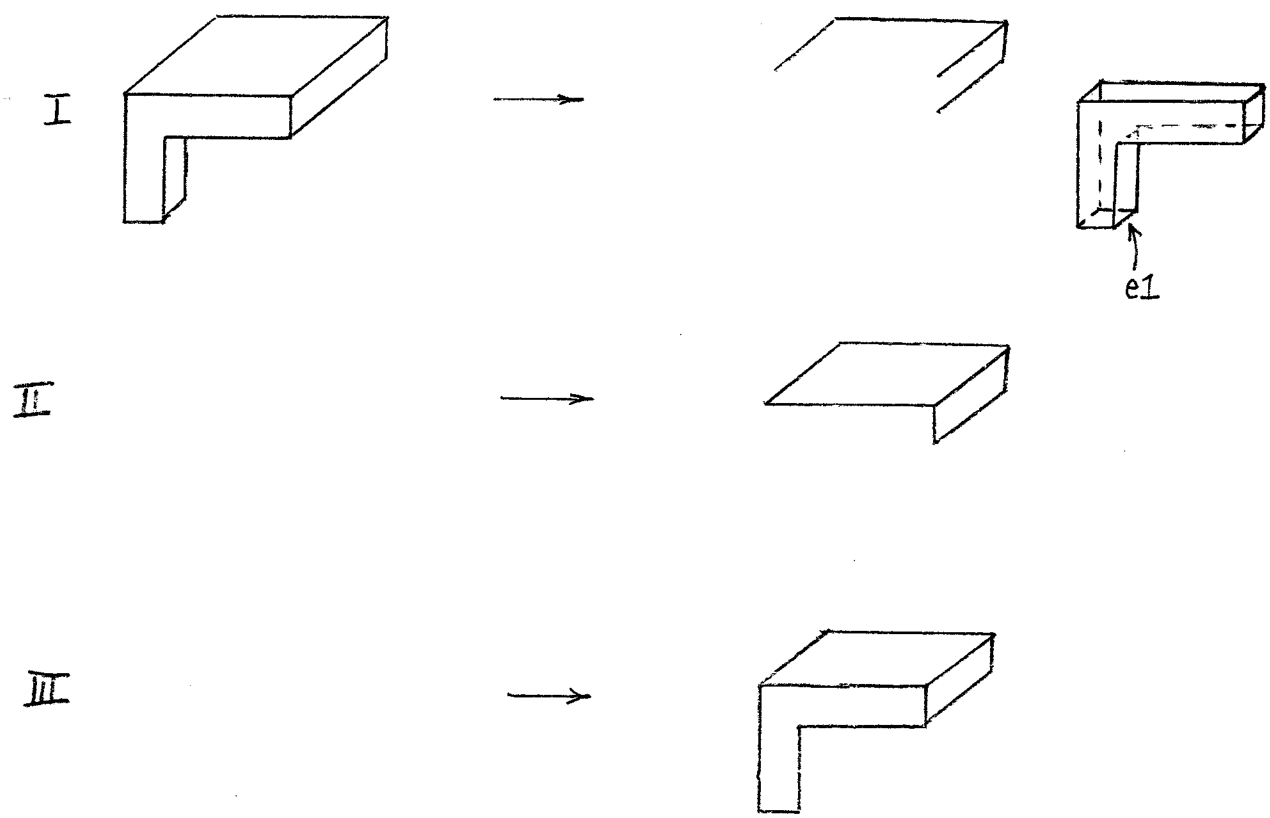


Fig. 25

carried out to such a point that an edge of the border portion is collinear with the edge of the CUTT, and the two edges can be combined into one edge by simple extension. When this cannot be done, the Propagation Rule fails.

The CUTT-XRAY vertex pair is treated somewhat differently from the others, and is included under the Propagation Rule because the exactly-two-uncompleted-vertices condition holds for it. The rule here is to join the respective edges of the XRAY and CUTT if they are collinear.

The reason for the restriction to exactly two uncompleted vertices is that invalid results can otherwise be conjured up. There are four uncompleted vertices after step II in Fig. 26, and application of the Propagation Rule to the CUTT-XRAY vertex pair produces the clearly invalid connection in step III. Examples of invalid constructions for the other vertex pairs under relaxation of this restriction will not be given here.

The inclusion of the reconstruction rules into Propa-pH is presented in section 3.2.

2.3 Three Dimensional Coordinate Generation

The main result of this section is that under certain circumstances knowing the 3-D coordinates of exactly four vertices of an object suffices to determine the 3-D coordinates of the remaining vertices. The circumstances under which the result is valid is knowledge of the 2-D coordinates of all

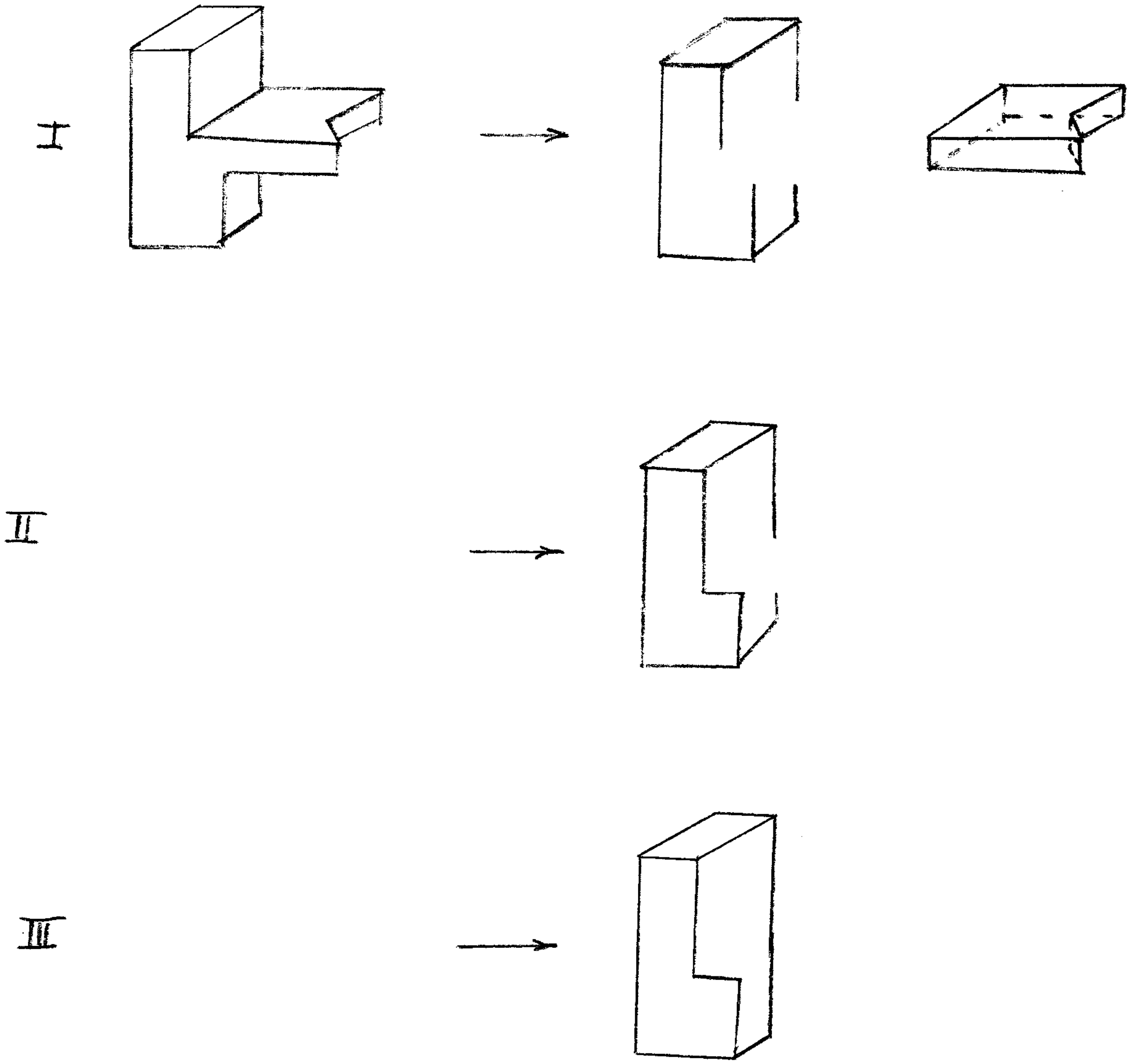


Fig. 26

vertices of a trihedral planar solid under a trimetric projection. Since Propa-pH generates the 2-D coordinates of hidden vertices, this result can be applied to an object constructed by it. It is later shown how the 3-D coordinates of four points can often be conveniently obtained in practice.

The result is developed in a succession of three theorems.

<u>Theorem 1</u>: Given 3-D coordinates of three noncollinear vertices of a plane plus the 2-D coordinates of all the vertices of the plane under a trimetric projection, the 3-D coordinates of the remaining vertices can be calculated.

<u>Proof</u>: It is assumed the plane is not projected into a line. Let T be the trimetric projection. There is a 1-1 correspondence between points on the plane in 3-space and points in the projection of the plane in 2-space.

Let $\{(x_i\ y_i\ z_i)\}$, i=1,2,3, be the coordinates of the three noncollinear vertices of the plane, and let

$$T\ (x_i\ y_i\ z_i) = (a_i\ b_i) \quad i=1,2,3$$

be the associated points in 2-space. Define vectors

$$\alpha_1 = (a_2-a_1 \quad b_2-b_1)$$

$$\alpha_2 = (a_3-a_1 \quad b_3-b_1)$$

$$\beta_1 = (x_2-x_1 \quad y_2-y_1 \quad z_2-z_1)$$

$$\beta_2 = (x_3-x_1 \quad y_3-y_1 \quad z_3-z_1)$$

Let V be the subspace spanned by $\{\beta_1, \beta_2\}$. It is clear V is nothing other than the plane in 3-space with boundaries extended to infinity. Since the vertices are noncollinear, β_1 and β_2 are linearly independent. Let us restrict T to this plane, and call this new transformation T_p. Since T is a trimetric projection, T_p is a linear transformation from the extended plane to all of 2-space. Moreover, α_1 and α_2 are linearly independent. Thus T_p is an isomorphism, and there consequently exists an inverse linear transformation T_p^{-1} such that

$$T_p^{-1}\alpha_i = \beta_i \qquad i = 1,2.$$

Let (a b) be any point in 2-space, and define a vector

$$\alpha = (a-a_1 \;\; b-b_1).$$

Since $\{\alpha_1, \alpha_2\}$ is a basis for 2-space, there exist scalars c and d such that

$$\alpha = c\alpha_1 + d\alpha_2$$

Thus

$$\begin{aligned} T_p^{-1} &= cT_p^{-1}\alpha_1 + dT_p^{-1}\alpha_2 \\ &= c\beta_1 + d\beta_2 \end{aligned}$$

If $\beta = T_p^{-1}\alpha$, we can express β as

$$\beta = (x-x_1 \;\; y-y_1 \;\; z-z_1)$$

With a little effort, it can now be shown that

$$(x \; y \; z) = (a-a_1) \begin{bmatrix} a_2-a_1 & b_2-b_1 \\ a_3-a_1 & b_3-b_1 \end{bmatrix}^{-1} \begin{bmatrix} x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{bmatrix} + (x_1 \; y_1 \; z_1)$$

Thus the 3-D coordinates of any vertex can be calculated if the 2-D coordinates of that vertex are known.

Theorem 2: Let the 2-D coordinates of vertices of a planar trihedral solid subjected to a trimetric projection be known. Suppose no plane is projected into a line. If the 3-D coordinates of three vertices of any plane A, plus the 3-D coordinates of a non-A vertex that forms an edge with a vertex of A, are known, then the 3-D coordinates of all vertices in all planes adjacent to A can be calculated.

Proof: By adjacent planes is meant planes that have at least one common point on their respective perimeters. Since the solid is trihedral, each adjacent plane B_i to A has as an edge an edge of A. Moreover, if two edges l_i and l_j of A form a vertex, then the corresponding adjacent planes B_i and B_j themselves share an edge. Suppose plane B_1 is adjacent to A and has as an edge the edge formed from a vertex of A to the fourth vertex mentioned above. Let the corresponding edge of A to B_1 be l_1, and moving in a particular direction around the perimeter of A from l_1, let the edges be consecutively labeled as $l_2,\ldots,l_k$, where k is the number of edges and $l_{k+1} = l_1$. By Theorem 1, the 3-D coordinates of all vertices of A can be calculated. Thus three vertices of B_1 are known in 3-space, and the remaining vertices of B_1 can be determined. Plane B_2 shares two vertices with A, and since B_1 and B_2 share an edge, B_2 shares a distinct third vertex with B_1. Thus B_2 has three vertices known in 3-space, and once again Theorem 1 can be applied to determine the remaining vertices. This argument can be carried out inductively to calculate the 3-D coordinates of all vertices of all planes adjacent to A.

<u>Theorem 3</u>: Given the assumptions of Theorem 2, the 3-D coordinates of all vertices of a trihedral solid can be calculated.

<u>Proof</u>: Theorem 2 can be applied to each plane B_i adjacent to A, so that all adjacent planes to B_i have their vertices determined in 3-space. It is clear that recursive application of Theorem 2 terminates with the calculation of the 3-D coordinates of all vertices of the solid.

If the solid is not trihedral, more than four vertices are sometimes required to completely specify the vertices in 3-space. For example, six vertices are required for the object in Fig. 27. The minimum number of vertices required can be determined as follows. For a given object, let us label certain of the vertices as being known in 3-space. If repeated application of Theorem 1 results in the determination of all vertices, then the number of vertices initially labeled form an upper bound for the minimum number of vertices required. It is clear that this minimum number is the least upper bound. The reader may wish to apply this argument to the object in Fig. 27, and verify that the magic number is six.

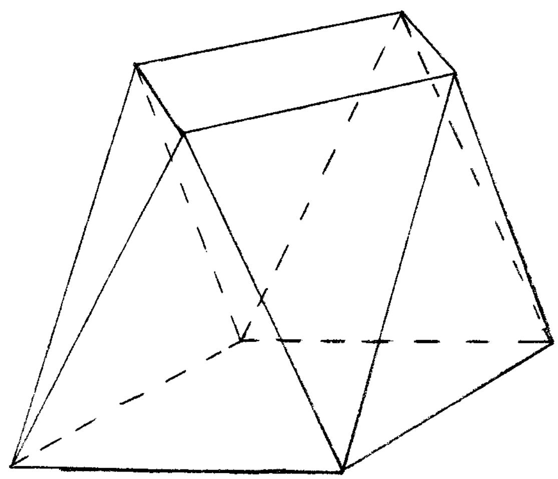


Fig. 27

The knowledge of the 2-D coordinates of all vertices of the solid under a projection is vital to the proof. If only visible vertices are known, then of course 3-D coordinates can only be calculated for these visible vertices. Moreover, the 3-D coordinates of more than four vertices are often required for this calculation. The number of vertices required has been determined by Falk(1), who has developed a formalism to treat this problem under assumptions close to those given in section 3.1. The basic idea underlying the formalism is essentially analogous to the procedure developed in the previous paragraph. For a more complete discussion, the reader is referred to Falk(1).

In real world situations, an object is resting on a table and is being "looked at" by a vidisector camera hooked up to a computer. The 3-D coordinates of the corners of the table in some coordinate system are usually known. Since the 2-D coordinates of any point on the table as it appears in the image are also known, it is possible to determine the 3-D coordinates of any vertex on the bottom plane of the object, i.e., that plane which rests squarely on the table. Thus three of the four points required by Theorem 3 for complete determination are usually easily obtainable. The fourth vertex can often be obtained by some supplementary assumption, such as assuming vertical edges on the picture correspond to real vertical edges. Thus application of

Theorem 3 to a one object scene often requires no auxilliary determination of the 3-D coordinates of some of the vertices.

3. Development

3.1. Basic Assumptions

As this is the first attempt at deriving a three dimensional description of an object by projecting the most complex plane, several restrictions were imposed in order to make the approach manageable. These are listed below.

(1) Only trihedral planar solids are considered.

A trihedral planar solid is one whose vertices are formed by the intersection of exactly three planes. Relaxation of this assumption renders many of the reconstruction rules invalid, and thereby considerably complicates the reconstruction process. No attempt has been made to derive general reconstruction rules for a less restricted class of objects.

(2) All projections are trimetric.

A trimetric projection is a parallel projection in which any initial transformation is a pure rotation, and in which orthogonality of the transformed axes is preserved. Considerable complexity to the present approach results when one point perspective pictures of objects are used. There is difficulty in recognizing when lines are parallel, since some 3-D information is usually needed beforehand for this recognition.

(3) Small changes in position of an object leave the topology of the projection unchanged.

Assumption (3) rules out degenerate projections and certain unfortunate "coincidences." These coincidences usually manifest themselves as vertices in the projection that have more than three edges. It is not expected that relaxation of this assumption will cause a great deal of difficulty, and is perhaps the next logical extension of Propa-pH.

(4) Scenes consist of single bodies only. Anticipated problems upon relaxation of this assumption are further discussed in section 4. Reconstruction of a multiple body scene when a body is deleted from it can be quite different in character from that seen in section 3.2.

3.2. PLANNER Implementation

A PLANNER program that purports to implement Propa-pH will now be presented, not without some hesitation due to my limited experience with the language. Nevertheless, it is felt that the organization of Propa-pH will be made much clearer as a result. Only limited use is made of PLANNER capabilities, of which the backup feature is the most useful. The proper choice of a plane, in order to decompose an object into a projected object and another object for which a description can eventually be obtained, is thereby facilitated. Otherwise, the operation of Propa-pH is rather straightforward, since most of the necessary computations are mechanical in nature. This accounts for the preponderance of LISP functions in the PLANNER theorems.

When the LISP functions were conceived, it was necessary to have in mind what the structure of the data base was that was being operated upon. Without going into any detail, let it merely be stated that the data base structure contemplated is very similar to that employed by Guzman(3). The reader should assume that proper bookkeeping exists to handle descriptions of objects as they are generated. The only explicit reference to the data base in the PLANNER theorems is to obtain the names of subbodies formed in the carving process by TC-HACKSAW.

As implied in the previous section, there are three distinct stages in Propa-pH: the first stage attempts to describe the object by projection of the most complex plane, the second attempts to separate the body into two simpler pieces when this cannot be done, and the third stage does some reconstruction on the object remains formed in the separation process. These stages are implemented by four PLANNER theorems, each of which is discussed at length below. For reference, the interplay of the theorems is shown in Fig. 28. The PLANNER theorems are presented on succeeding pages to this Figure.

The first theorem, TC-PROPAPH, attempts to describe the object by projection of the most complex plane. If the attempt fails, the theorem TC-HACKSAW is called to cut up the object. Upon entering TC-PROPAPH, the planes are sorted by complexity according to the LISP function SORTPLANES. The most complex plane in this list is used as an argument for a call to TC-PROJECTABLE, which checks most of the conditions

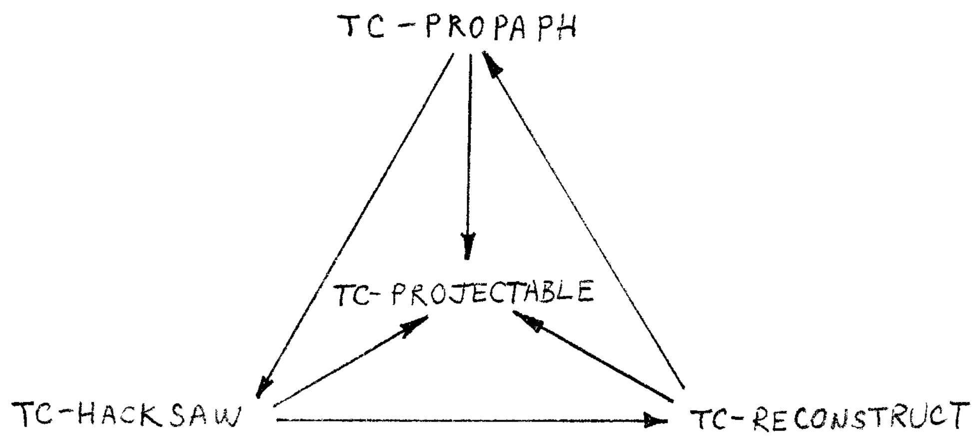


Fig. 28

```
(DEFPROP TC-PROPAPH
     (THCONSE (BODY PLIST PTLIST)
              (#SPECIFY $?BODY)
              (THSETQ $?PLIST (SORTPLANES $?BODY))
              (THSETQ $?PTLIST (LIST (CAR $?PLIST)))
              (THOR (THAND (THGOAL (#PROJECTABLE $?BODY $?PTLIST
                                                             NOCUT)
                                   (THNODB)
                                   (THUSE TC-PROJECTABLE))
                           (EQUAL (LENGTH (GETP $?BODY 'SUBBODIES))
                                  1))
                    (THGOAL (#DISSECT $?BODY $?PLIST)
                            (THNODB)
                            (THUSE TC-HACKSAW))))
    THEOREM)

(DEFPROP TC-PROJECTABLE
     (THCONSE (BODY PLIST CUT PLANE)
              (#PROJECTABLE $?BODY $?PLIST $?CUT)
              (THAMONG $?PLANE $?PLIST)
              (PARALINE $?PLANE)
              (UNIDIREC $?PLANE)
              (EQUILONG $?PLANE)
              (PROJECT $?BODY $?PLANE $?CUT))
    THEOREM)
```

```
(DEFPROP TC-HACKSAW
     (THCONSE (BODY PLIST)
              (#DISSECT $?BODY $?PLIST)
              (THOR (THAND (THGOAL (#PROJECTABLE $?BODY $?PLIST
                                                           NOCUT)
                                    (THNODB)
                                    (THUSE TC-PROJECTABLE))
                           (THGOAL (#RECONSTRUCT $?BODY)
                                    (THNODB)
                                    (THUSE TC-RECONSTRUCT)))
                    (THAND (THGOAL (#PROJECTABLE $?BODY $?PLIST
                                                             CUT)
                                    (THNODB)
                                    (THUSE TC-PROJECTABLE))
                           (THGOAL (#RECONSTRUCT $?BODY)
                                    (THNODB)
                                    (THUSE TC-RECONSTRUCT)))
                    (THPROG (PLANE)
                            (THAMONG $?PLANE $?PLIST)
                            (PARALINE $?PLANE)
                            (THCOND ((UNIDIREC $?PLANE))
                                     (T (HUFFMAN $?BODY $?PLANE)))
                            (THDO (EQUILONG $?PLANE))
                            (PROJECT $?BODY $?PLANE 'CUT)
```

```
                                   (THGOAL (#RECONSTRUCT $?BODY)
                                           (THNODB)
                                           (THUSE TC-RECONSTRUCT)))))
     THEOREM)

(DEFPROP TC-RECONSTRUCT
     (THCONSE (BODY REMAINS)
              (CUTEDGES $?BODY)
              (CUTARFOTS $?BODY)
              (CUTARFOS $?BODY)
              (XRAYS $?BODY)
              (CUTLS $?BODY)
              (THSETQ $?REMAINS (CADR (GETP $?BODY 'SUBBODIES)))
              (THOR (THAND (PROPAGATION $?BODY)
                           (THOR (THGOAL (#SPECIFY $?REMAINS)
                                         (THNODB)
                                         (THUSE TC-PROPAPH))
                                 (THFAIL THEOREM)))
                    (THPROG (PLIST BLIST PART)
                            (THCOND ((THSETQ $?BLIST
                                             (LIST $?REMAINS)))
                                    ((THSETQ $?BLIST
                                             (SORTBODIES $?REMAINS))))
                        TAG (THCOND ((NULL $?BLIST)
                                     (THSUCCEED THPROG))
                                    ((THSETQ $?PART (CAR $?BLIST))))
                            (THSETQ $?PLIST (SORTPLANES $?PART))
```

```
                    (THCOND ((EQUAL (LENGTH (GETP $?PART
                                               'SUBBODIES))
                                              1)
                             (THGO THTAG))
                            ((THFAIL THTAG TAG))))))
THEOREM)
```

that must be fulfilled to describe the object by projection of this plane. Since NOCUT was specified in the call to TC-PROJECTABLE, the projection fails if a plane of the object would be cut as a result. If these conditions are satisfied, TC-PROJECTABLE returns successfully after having carried out the projection and having set up descriptions for the projected object and the object remains. If there are object remains, the attempt has failed and TC-HACKSAW is called. If the latter fails, then TC-PROPAPH cannot be used to describe this object.

TC-PROJECTABLE seeks to project a plane along equal-length rays that are parallel and unidirectional. When the rays are as described above, the object is separated into a projected object and object remains. An arbitrary list of planes can be provided to TC-PROJECTABLE, which will go through the list until a plane is found that has the required properties. The LISP functions that accomplish this are briefly described below.

PARALINE--checks if rays eminating from a plane are all parallel. In the case of loose ends from a reconstructed body, at least one of the rays must end in a permanent vertex. T or NIL is returned accordingly.

UNIDIREC--checks if rays eminating from a plane are unidirectional, and returns T or NIL accordingly.

EQUILONG--finds the maximum allowable distance of projection along one of the rays of the plane. This distance is limited by the shortest ray that is not the shaft of a T, or by a concavity of the plane. The appropriate distance is noted in the data base. T is returned if all rays that are not shafts of T's are equally long, if the projection is not limited by a concavity, and if any ray that is the shaft of a T is not longer than the other rays.. Otherwise, NIL is returned.

PROJECT--separates an object into two on the basis of projection of a plane, and sets up proper descriptions for each. A list of names of the two pieces is entered in the property list of the object under SUBBODIES. If NOCUT is specified, i.e., no plane is to be cut in the projection, then PROJECT does not carry out the separation and returns NIL if indeed a plane would be cut. Otherwise, T is returned.

There are three different ways in which TC-HACKSAW seeks to dissect an object. The first involves removal of a "protrusion," as defined in the previous section. To find such a protrusion involves almost exactly the same tests as used in TC-PROPAPH, as may be seen by the identical call to TC-PROJECTABLE. If there are object remains, however, a protrusion has been found. TC-RECONSTRUCT is called to complete the remains and to initiate a description attempt on them. When TC-RECONSTRUCT succeeds, so does TC-HACKSAW.

If no protrusions exist or if TC-RECONSTRUCT fails, a "protrusion-like" part is sought. A "protrusion-like" part is similar to a protrusion, except that PROJECT is allowed to cut a plane during the call to TC-PROJECTABLE (e.g., step I in Fig. 10). If such a part has been found, TC-RECONSTRUCT is applied to the remains as before.

Lastly, if no protrusion-like parts exist or TC-RECONSTRUCT fails, TC-HACKSAW runs through the list of planes and chooses any plane that is projectable, i.e., a plane for which the LISP function PARALINE returns T. If the rays are not unidirectional, selective Huffman labeling is used to determine the proper direction of projection. Briefly, the LISP function HUFFMAN seeks a ray whose vertex on the projectable plane has two other non-concave edges. The proper distance of projection is determined by EQUILONG, and PROJECT carries out the separation. Once again, TC-RECONSTRUCT is applied to the remains. TC-HACKSAW succeeds if eventually a plane is chosen that results in successful description of the pieces of a separation.

The reconstruction rules in TC-RECONSTRUCT can be divided (like Gaul) into three parts: preliminary edge reconstructions, an attempted application of the Propagation Rules, and a simple projective plane hack. Initially, edge reconstructions are carried out on the various vertex types as allowed by their respective properties. These are conducted by appropriate LISP functions which are summarized below.

CUTEDGES--adds the edge(s) of the visible portion of that cutting plane which produced a CUTEDGE. The length of each added edge is limited by the first loose-ended vertex along it.

CUTARFOTS--adds the remaining edge as dictated by the cutting plane.

CUTARFOS--attempts first to connect each CUTARFO by simple extension to another loose-ended vertex. When this is not possible, an edge is added that was originally an edge of the ARFO vertex, but was not solely an edge of the projected object.

XRAYS--extends the XRAYs the length of the projection. If an extended XRAY has formed an L, then a neighboring extended XRAY is connected to it if an arrow is thereby formed, of which the first XRAY is on the shaft.

CUTLS--adds the edge of the cutting plane that produced it.

Some duplication would result if these LISP functions were executed independently, and appropriate safeguards must exist to avoid it.

After these functions act, the vertices will be in various stages of completion. The expected number of visible edges for each type, listed again below for sake of reference, determines when a vertex is complete.

(1) CUTEDGEs and CUTARFOTs have three visible edges.

(2) If a CUTARFO does not disappear by simple extension, it has three visible edges.

(3) If originally an L had a concave edge, the corresponding CUTL will have two visible edges. Otherwise, if it does not disappear by simple extension, a CUTL has three visible edges.

(4) An extended XRAY that does not disappear by coincidence normally will become part of a three-edged vertex. However, if the original ray is an edge of only one visible plane, and if the XRAY has formed an L vertex, then two edges only are visible.

(5) All CUTTs are impermanent vertices.

At this point there is an attempted application of the Propagation Rules. The LISP function PROPAGATION initially checks on the number of uncompleted vertices. If there are none, i.e., the preliminary reconstruction rules were enough, it returns T. If there are two, and if the Propagation Rules can be successfully applied as discussed in the previous section, PROPAGATION also returns T. TC-PROPAPH is subsequently applied to the reconstructed remains, and determines whether TC-RECONSTRUCT now succeeds or fails.

If PROPAGATION returns NIL, there is an attempt to describe the remains completely by projection of the most complex plane. This is very similar to application of TC-PROPAPH, except that TC-HACKSAW is not called in case of failure. If the remains exist as two or more disconnected parts and if the initial projection attempt has failed, then a similar description is attempted for each part. The LISP function SORTBODIES separates the remains into disjoint parts, sets up appropriate descriptions for each, and returns a list of their names. TC-RECONSTRUCT now succeeds only if each part is successfully described.

Thus ends Propa-pH.

4. Recapitulation

4.1. Description Building

When an object can be described by projection of the most complex plane, then the object can be identified with this plane. Thus a block can be considered a projected rectangle, a wedge a projected triangle, and so on. In the case of compound object construction, a description can be built up on the basis of the planes that characterize each of the parts. For example, the object in Fig. 23 would be described as a T-shaped object joined to a block.

Unfortunately, compound objects often lend themselves to more than one description. Given different initial planes to carve up an object, Propa-pH will separate different components for many objects. An alternate construction to the one in Fig. 17 could be produced by projection of plane A in Fig. 29.

There is nothing inherently wrong about different descriptions, but humans have a tendency to find some more natural than others. A general tendency seems to be to pick out the most prominent feature. In many cases this feature is the most complex plane, but as a general rule this observation is too simple-minded. The best interpretation of a compound object may be the one that identifies the bulkiest

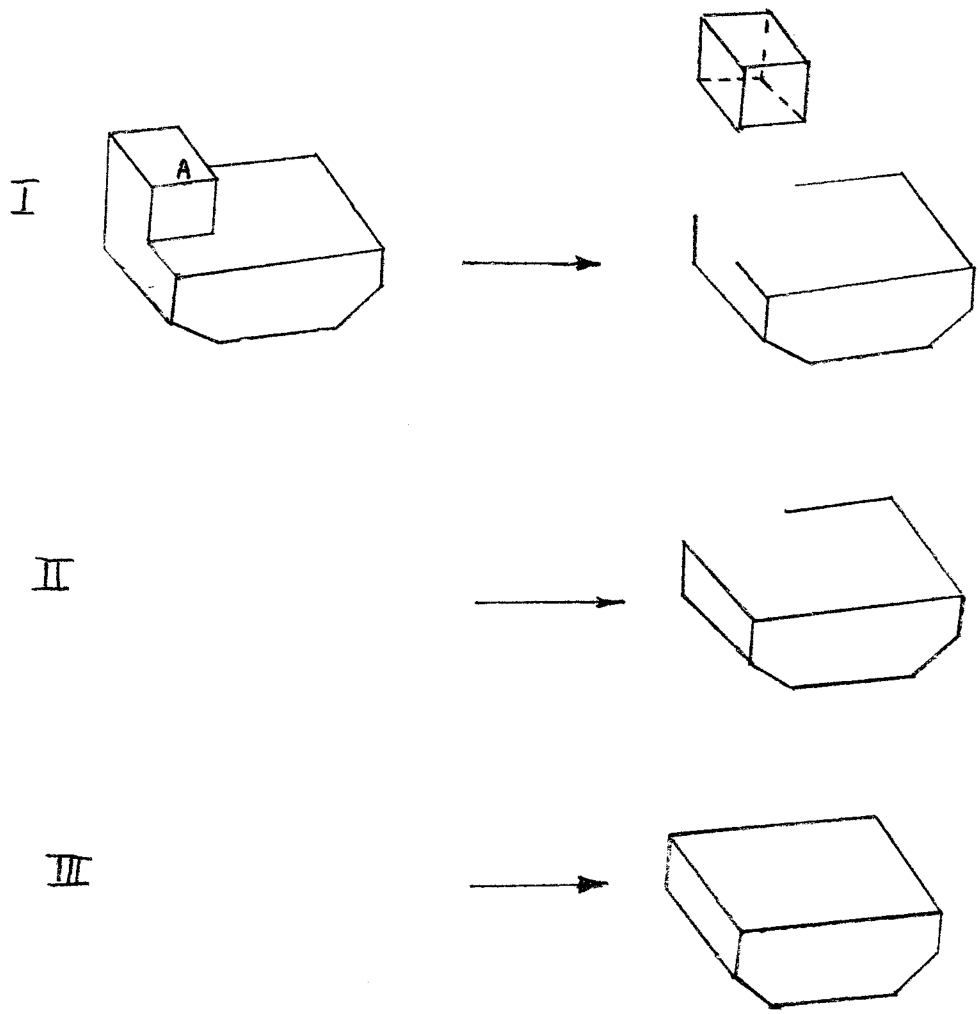


Fig. 29

component. For this reason I think humans find the interpretation in Fig. 29 more natural than the one in Fig. 17.

What is needed then are better ways of identifying components of a compound object. TC-HACKSAW makes a mild attempt at a more natural description by seeking first to remove protrusions (Fig. 9) and then to remove protrusion-like components (Fig. 10). Both of these can be considered to be prominent features. In any case, TC-HACKSAW will have to undergo extensive revision to conform with our ideas of naturalness.

4.2. Multiple Object Scenes

There are practically no problems in multiple object scene analysis that are not also problems in scenes of single objects. A single object that exhibits the problem may be very exotic, however, so that it is much more natural to consider certain problems as belonging to the multiple object scene domain. These problems are discussed in general terms below.

As exists, Propa-pH can handle a very restricted class of multiple object scenes, which usually are composed of aligned objects without visible vertex types of more than three edges. For example, Figs. 20 and 21 could just as well be considered to represent two object scenes. Nevertheless, Propa-pH at present is clearly inadequate for most multiple body scenes.

The very first problem one encounters in such scenes is deciding which object to analyze first. It would be nice to choose an unobstructed object, but such an object need not exist.

After an object has been chosen, specified and deleted from a scene, one must reconstruct those bodies that were partially blocked by the deleted object. This reconstruction is more difficult than before, since objects may not be touching and very little information need be provided by the deleted object to guide the reconstruction. It remains to be seen if adequate rules can be formulated, and it is quite possible that some auxilliary considerations must come into play.

Often one dimension is left unspecified and must be estimated somehow. Perhaps some other information in the scene can be used to establish a credible limit on this dimension, such as apparently similar unobscured objects. Another assumption that may be of utility is to extend this dimension to the maximum allowable limit as dictated by the scene; for example, the block in Fig. 30 is extended in step II as far as possible. Or one could just as well choose the miminal distance, which just shows there are no good assumptions.

Whatever assumptions one makes, it is necessary that reality not be violated. A given interpretation of an object

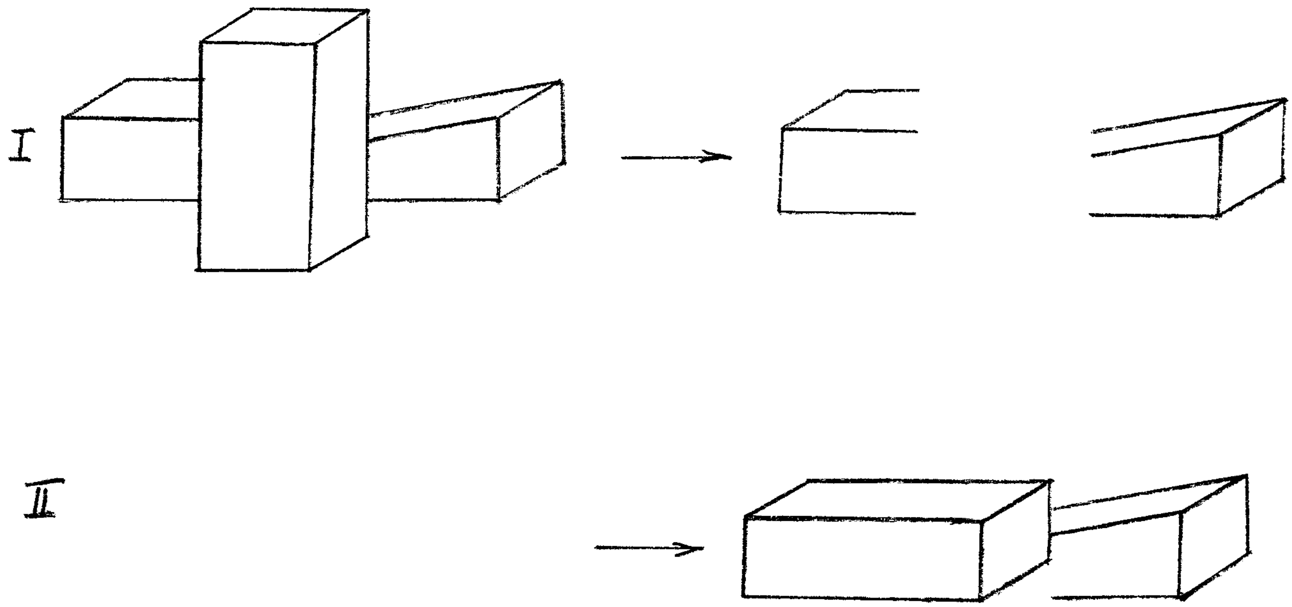


Fig. 30

may cause collision with other objects. For example, if all objects in Fig. 30 are resting on the ground, it is not now possible to interpret the rightmost object as a wedge, since it would collide with the extended block. A scene must also not be rendered unstable because of some interpretation. Generally speaking, questions of collision and stability can only be determined with complete three-dimensional information, although sometimes good estimates can be made in two dimensions. Finally, an interpretation must not conjure up parts that would have been visible in the original scene, but are not.

4.3. Comparison with Previous Efforts

As mentioned in the introduction, the two most important works in object identification from a two-dimensional picture are Roberts(5) and Guzman(2). A brief summary of each work will be given, along with a comparison with Propa-pH.

4.3.1. Roberts

The assumptions Roberts makes about objects and their 2-D projections are very similar to those enunciated in section 3.1. Objects in a scene are identified by matching with known three-dimensional models. Using topological information from the 2-D projection, a plausible model is selected. A transformation matrix is then set up such that successive rotations, translations and projection map the model exactly onto the projected object. This procedure requires some type of error analysis to detect a good fit.

In case of a compound object not describable by transformation of a single model, an attempt is made to describe it by piecing together several models. As in the present approach, the construction is destructive in the sense that the object is carved into pieces by projecting a model onto an object and deleting it. A reconstruction of the remainder of the object then takes place to prepare it for the next model mapping. How wedge and cube models are used to describe a compound object is given as an example in Fig. 31. A cube model is always tested before a wedge model to avoid splitting cubes into wedges.

The use of models can become very cumbersome if there are a large number of them. The proper selection of a model and generation of an appropriate transformation matrix then become very time consuming. What objects can be constructed are of course limited by the selection of models. There exists no such inherent limitation in Propa-pH. An object may be arbitrarily complex, so long as it is describable as a projection of the most complex plane, or as an object composed of pieces that are so describable.

His method of reconstruction of an object after a model has been found which fits part of it is one of line deletion. After all lines from the piece to be cut out have been added,

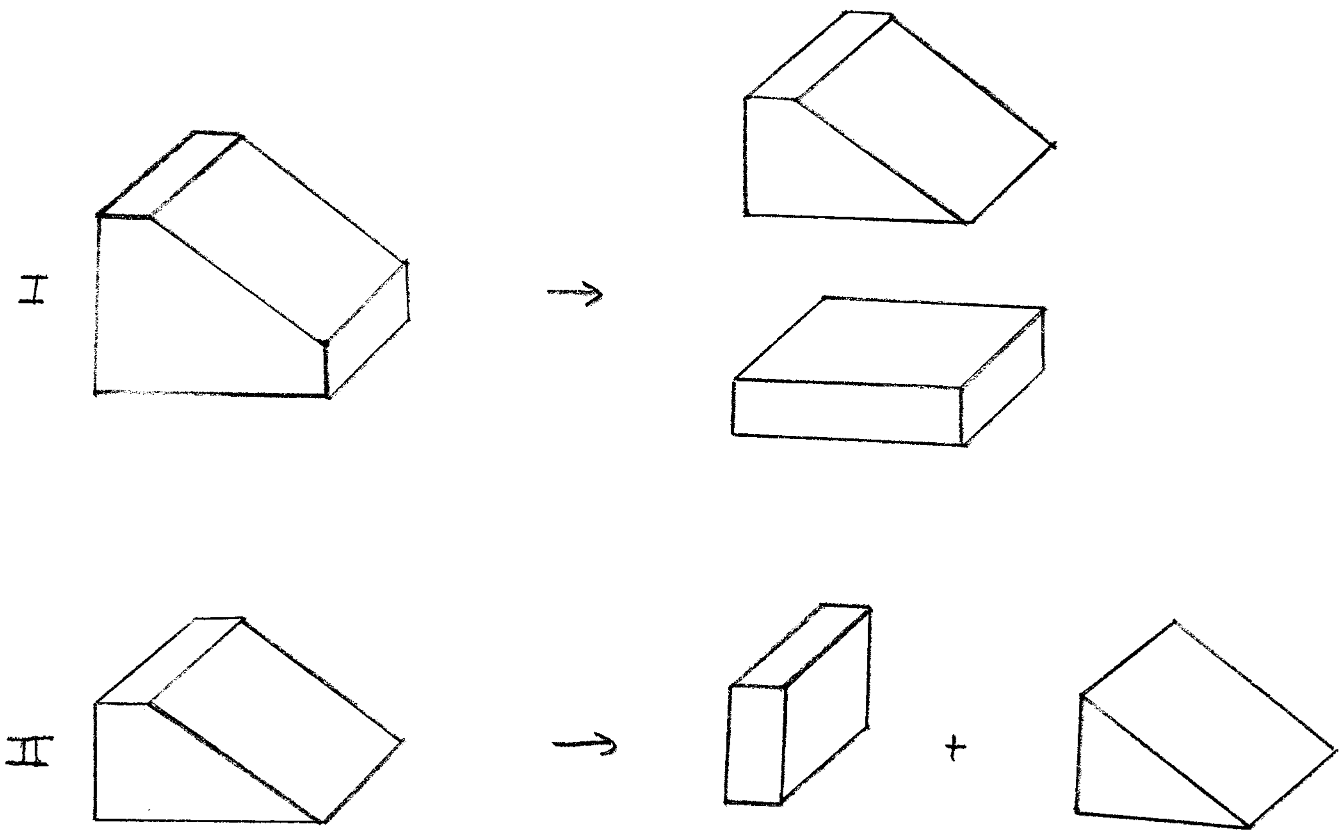


Fig. 31

certain rules are applied to determine which lines must be erased. As noted in section 2.2.2, this approach fails to take into account contextual information from the remainder of the object. Construction of new lines not part of the piece to be cut out is sometimes called for, but Roberts makes no provision for this.

It is not likely humans see an object as being composed of wedges and cubes glued together. It seems to me that an object such as that in Fig. 31 is much more naturally described as a projection of its most complex plane. Clearly many objects are naturally describable in terms of bits and pieces, but the main point of difference between Roberts' work and Propa-pH is the size of these pieces. By projecting the most complex plane, Propa-pH attempts to make these pieces as large as possible.

4.3.2. Guzman

In a program called DT, Guzman has employed a number of models for a standard object to identify constituents of a two-dimensional picture. These models are two-dimensional, and correspond to different ways in which the standard object can be projected into two-space. Matching is almost completely topological; that is, region shapes and the positions of regions with respect to other regions must be identical in model and 2-D object. For simple objects, the number of important models is relatively small. Thus a wedge can be described as an object composed of a triangle and one or two parallelograms (Fig. 32). As the complexity of the object increases, so do the number of models (Fig. 33).

If a large number of objects are to be modelled, it is clear an enormous and unwieldy data base would result. There is no provision, moreover, for describing a complex object as composed of a number of smaller, recognizable pieces. Practically every object in Guzman's thesis is easily handled by Propa-pH. Whereas his approach is characterized by a great deal of data base search, Propa-pH is almost completely procedural in nature due to a complete lack of models. Roberts' work can be considered to fall somewhere in between these two approaches.

4.4. Coda

Is Propa-pH anything other than a nice hack? Does it correspond to how humans might build descriptions? Some affirmative evidence I believe is provided by hole recognition (Fig. 34). I claim that a hole is recognized not by the interrelationship of vertices (an altogether too microscopic a view), but by the ability to project the opening backwards. Or, what is the same thing, to project the surrounding plane backwards.

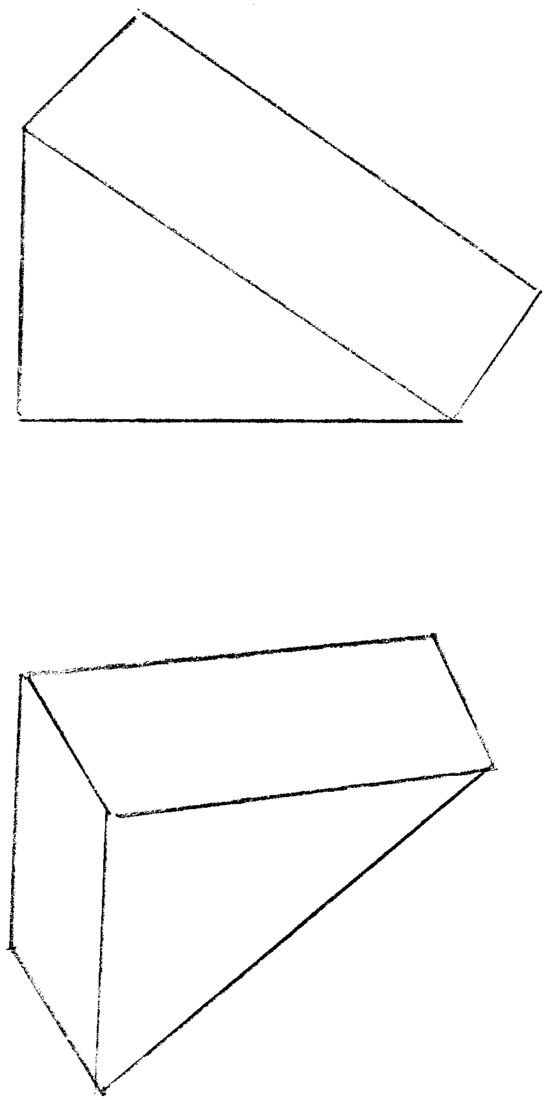


Fig. 32

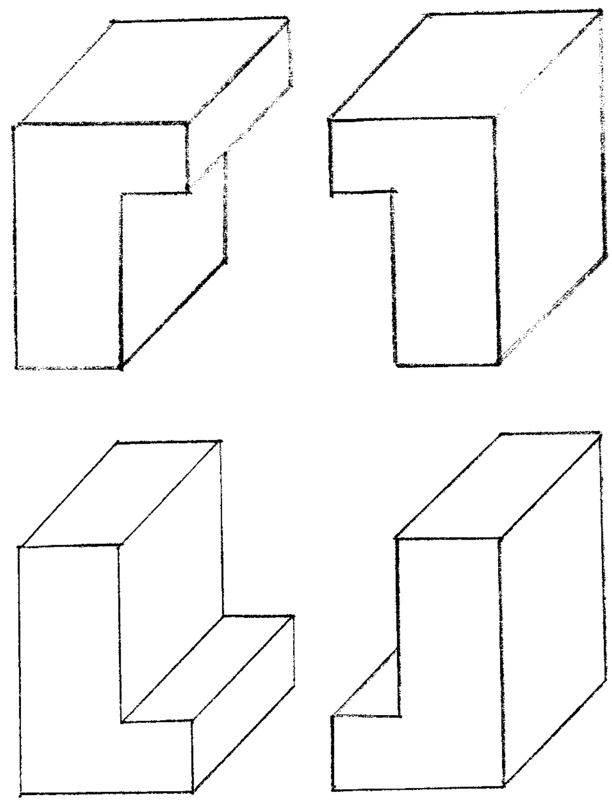


Fig. 33

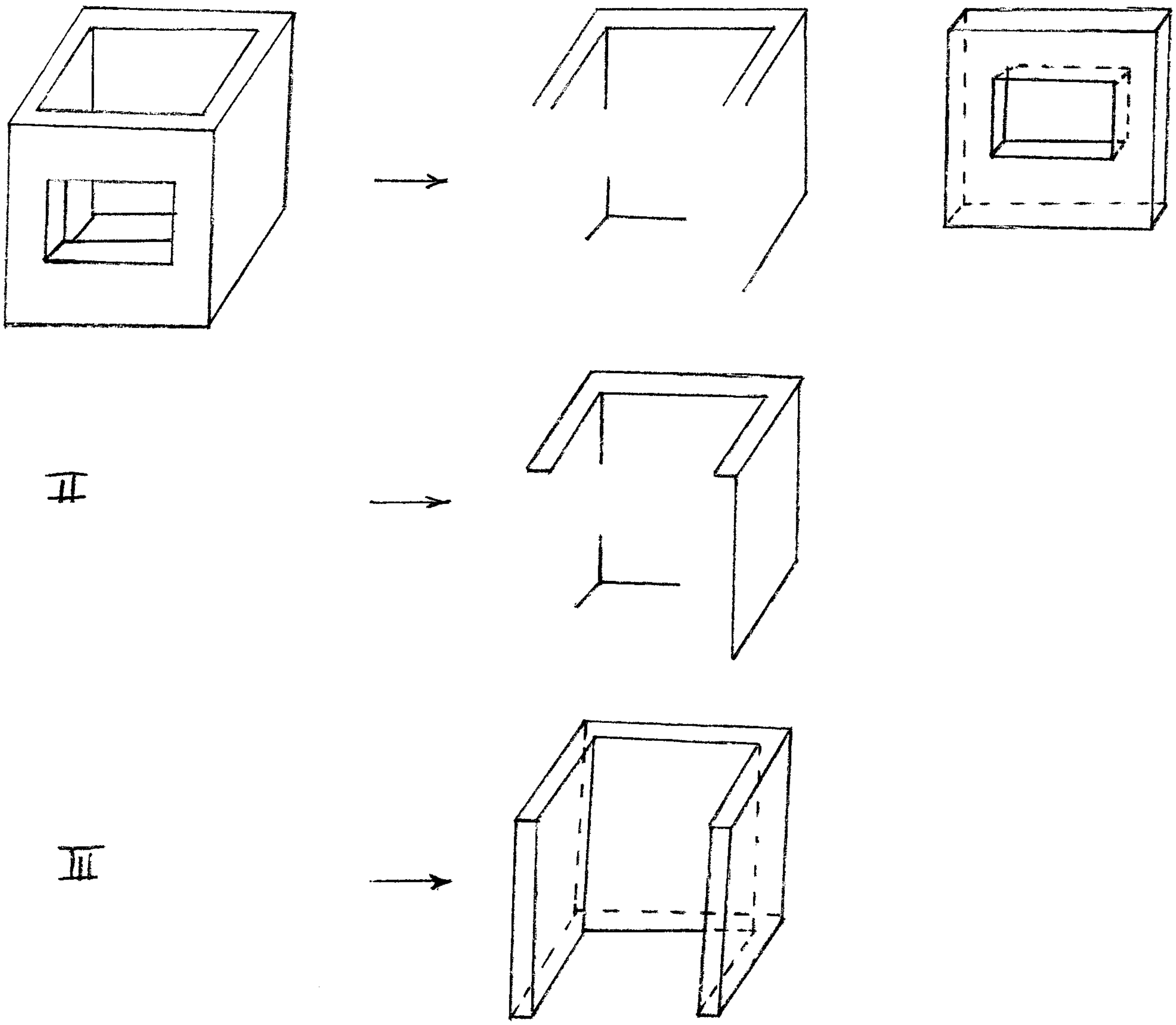


Fig. 34

The procedural nature of Propa-pH provides more insight into these questions. Rather than analyzing a scene in terms of fixed models, Propa-pH works completely with the information provided by the scene. I do not believe it is much different with humans. We look at an object in much the same way as astronomers expected the back side of the moon to look before a Russian satellite photographed it: not much different from the front. This is the essence of the present approach: to make an educated guess at what the parts hidden from view look like solely on the basis of what's in front. The words of a current pop song sums it all up. What you see is what you get.

References

(1) Falk, G. Some Implications of Planarity for Machine Perception. Stanford Artificial Intelligence Project Memo No. A.I. 107. December, 1969.

(2) Guzman, A. Some Aspects of Pattern Recognition by Computer. Project MAC Technical Report MAC TR 37, MIT. February, 1967.

(3) Guzman, A. Computer Recognition of Three-Dimensional Objects in a Visial Scene. Project MAC Technical Report MAC TR 59, MIT. December, 1968.

(4) Huffman, D.A. Impossible Objects as Nonsense Sentences. Machine Intelligence 6, 1970.

(5) Roberts, L.G. Machine Perception of Three-dimensional Solids. Optical and Electrooptical Information Processing, pp159-197. J.T. Tippett et al.(eds), MIT Press. 1965.