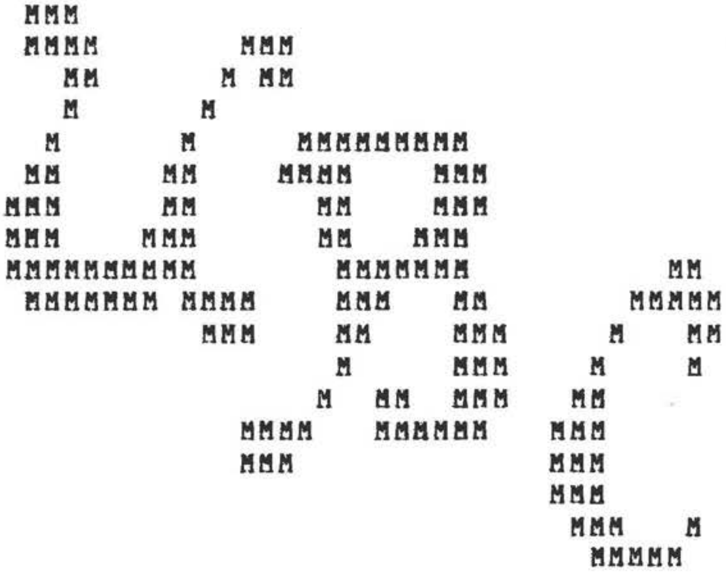

HOW TO SEE A SIMPLE WORLD

by

Alan K. Mackworth

Technical Report 75-4

July, 1975

Department of Computer Science
University of British Columbia
Vancouver, B. C.

Abstract

This is not a comprehensive survey of machine vision which, in its broadest sense, includes all computer programs that process pictures. Restricting attention to scene analysis programs that interpret line data as polyhedral scenes makes it possible to examine those programs in depth, comment on revealing mistakes and anomalies, explore the interrelationships and exhibit the thematic development of the field. Starting with Roberts' seminal work which established the paradigm, there has been an evolutionary succession of programs and proposals each approaching the problem with a different emphasis. In addition to Roberts' program thiis paper expounds in detail work done by Guzman, Falk, Huffman, Clowes and Waltz. These programs are presented, compared, contrasted and criticized in order to exhibit the development of a variety of themes including the representation of the picture-formation process, segmentation, support, occlusion, lighting, the scene description, picture cues and models of the world.

Presented at the NATO Advanced Study Institute on Machine Representations of Knowledge, Santa Cruz, Ca., July 1975.

0. Preamble

As this paper is primarily an exegesis of computer programs for polyhedral scene analysis, it should not be read as a review of all work in computational vision. The semantics of polyhedral scenes are so clean that we can re-view that body of work and see it as a coherent whole. On the other hand much recent work outside that area is so diverse and fragmented in character that it is hard to place it all within a single framework. However, the associated lectures will cover such topics as the interpretation of more complex scenes and the question of how image analysis (for example, line and region formation) can be guided by partial scene analysis. Within the area covered here the major omission is the MIT COPY DEMO which is so ably described by Winston(1973).

Caveat lector: one of the techniques used in this review is to point to non-trivial bugs in the programs discussed. These are useful for gaining insight into the weaknesses of the descriptions and inference mechanisms available to a program; however, it must be emphasised that, for the most part, these have been discovered not through running the program in question but through a careful reading of the published accounts. To seek refuge in the fact that most of these bugs could be fixed by admittedly *ad hoc* patches would be to mistake the symptoms for the disease.

1. Introduction

The Platonic assumption that the world is made up entirely of objects with flat surfaces obviously does not hold; and yet, as with so many other simplifications of reality for the sake of tractability, it has been immensely productive in establishing a paradigm for scene analysis. There is a coherent evolving body of research based on the notion that a polyhedral world is the simplest we can consider without eliminating any of the essential aspects of scene analysis, namely, the picture-taking process, models, lighting, support, occlusion and so on. The thesis is that once we achieve ways of dealing intelligently with those aspects for a simple, but nonetheless real) world we could then consider the fuzzy world of teddy bears (Michie, 1974) and the like. This should not be taken as suggesting that each of those aspects presents simply a separate, independent subproblem to be solved. The most important question to be faced was how to write programs that coordinate the use of these separate, but interrelated knowledge systems to achieve sensible picture interpretations. Roberts (1965) was the first to give an answer to this question. We shall examine his answer in some detail, because he exposed in it the issues that became themes of the first decade of scene analysis.

2. Roberts' program for scene analysis

Roberts (1965) described a program for the interpretation of photographs as images of fully three-dimensional scenes. By assuming that the scene is composed of particular instances of object models that have been transformed and combined in well-specified ways and by using knowledge of the picture taking process, support and occlusion, his system is able to compute the exact 3D position of every object in the scene. There are actually two separate programs. The first reduces the photograph to a line drawing, the second interprets the line drawing. The reduction to a line drawing does not concern us here because an adequate treatment of that topic is beyond the scope of this paper and because more recent work on line finding (Shirai, 1973; O'Gorman and Clowes, 1973) suggests that the simple, pass-oriented line-following procedures Roberts describes are not usually powerful enough to produce the complete line drawing required by the subsequent interpretation program.

Roberts' program believes that the world consists of the models shown in Fig. 1, namely, a cube, a rectangular wedge and a hexagonal prism. To create simple objects the system allows these models to be expanded along each of the model coordinate axes and then rotated and translated. Compound objects are created by abutting two or more simple objects so that each adjacent pair shares a common surface. The models are specified

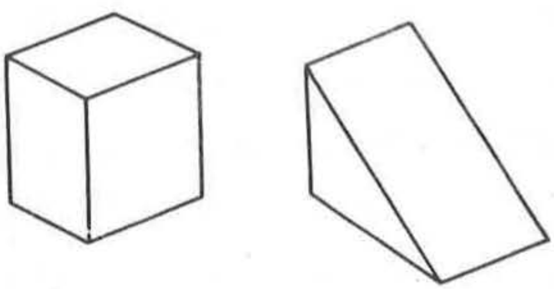

Figure 1. Roberts' simple object models

$$\begin{tikzcd}\text{MODEL} \arrow[r, "R"] \arrow[rr, bend right, "H"'] & \text{OBJECT} \arrow[r, "P"] & \text{PICTURE}\end{tikzcd}$$

Figure 2. Roberts' domains and transformations

by 3D homogeneous coordinates so that the transformation of a model to form an object is described as the transformation, by an initially unknown matrix R, of the coordinates of the corners and the normals to the surfaces. Similarly the perspective picture taking process is described as the multiplication by a known matrix P of the object coordinates to produce the picture coordinates followed by the removal of hidden lines. So the relationships of the model, object and picture domains are as shown in Fig. 2 where H, the model-to-picture transformation, is also shown. Since H = RP, if a model and a transformation H can be found that account for a set of the lines in the picture then the program maintains that the set of lines is a picture of the object given by a transformation $R = HP^{-1}$ of that model. Thus the object is identified and its location specified completely except for its actual distance from the camera. This distance is then computed from the requirement that the most downward facing surface of the object must lie in the ground plane. This is the only support hypothesis used by the program.

In this abbreviated account the most important point that has been glossed over is the decision to choose a set of picture lines to account for. This decision is followed by the choice of particular edges of a particular model to account for those lines. This is perhaps the archetypal artificial intelligence problem - the problem of relevance, by which is meant the problem of invocation of appropriately relevant models or procedures to account for the data.

The space of three models juxtaposed and transformed in all possible ways and viewed from every direction is unthinkably large for a blind search, (that is, generating all possible pictures of all possible objects until one matches the input) so the search space must be intelligently structured. Roberts noticed that all the model transformations leave the object's topology invariant and that within a wide range of viewpoints the topology of the visible aspect of an object does not change. Through this invariance the topology of the picture can be used to search a much reduced space consisting of the models viewed from a small number of typical viewpoints. On finding a candidate model, points that correspond in the model and the picture are paired. The coordinates of those pairs are used to calculate (rather than search for) the model-to-picture transformation, H. At least four pairs of points are needed to calculate H; if more are available then a least squares fit gives H with the residual error as a measure of the picture-model mismatch. If the mismatch is too large then that model is rejected and the topology search continues.

Consider the topology search in detail. It is based on the notion of an approved polygon which is simply one of the shapes of the model surfaces. For the three models used, an approved polygon is any convex polygon of 3, 4 or 6 sides. Since the topology search attempts to find the largest picture fragment that could correspond to a model, it proceeds in stages each of which looks for a smaller fragment than the one preceding. The four stages, which are called in sequence until one succeeds, are:

1. Find a picture vertex surrounded by 3 approved polygons.

2. Find a line with an approved polygon on each side.

3. Find an approved polygon with an extra line coming from one vertex.

4. As a last resort find a point with 3 lines coming from it.

When a suitable fragment is found the program searches the models in sequence (cube followed by wedge followed by prism) to find a topological structure that corresponds to the fragment recovered from the picture.

Fig. 3(a) shows a typical compound object considered by Roberts. The topology search finds no fragments of type 1, but two of type 2: both lines 2 and 3 have approved polygons on each side of them. The cube has quadrilaterals on both sides of an edge so the geometry matcher tries A and B as surfaces of a transformed cube as shown in Fig. 4, but discovers that the residual error of the least squares fit of the corresponding object-model point pairs is too large and rejects it. Similarly

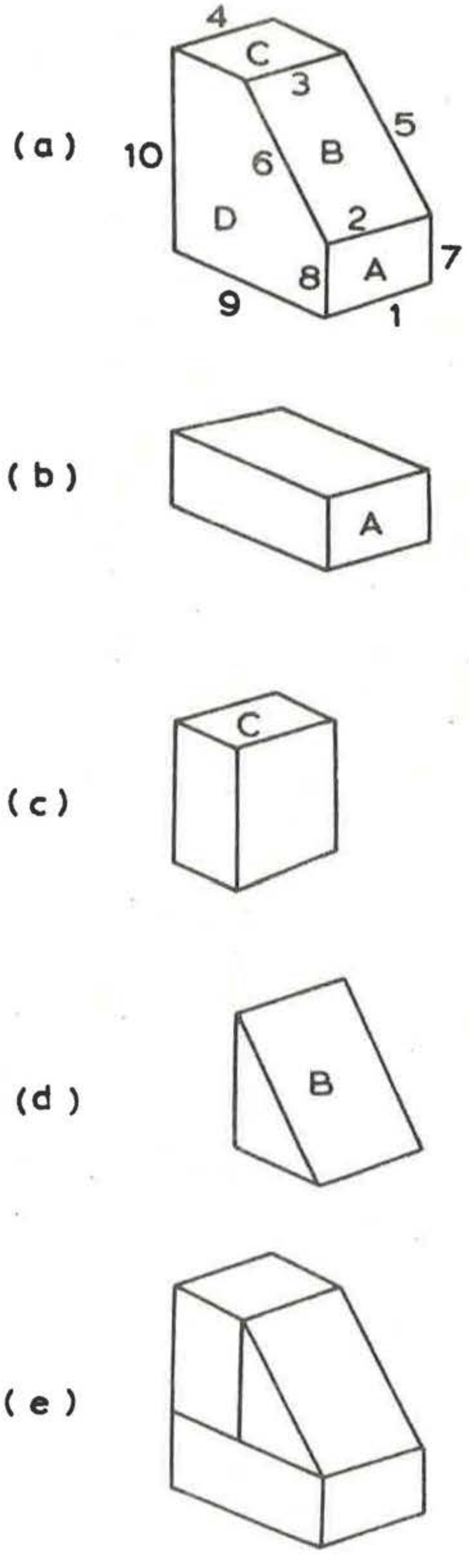

Figure 3. Interpreting a compound object

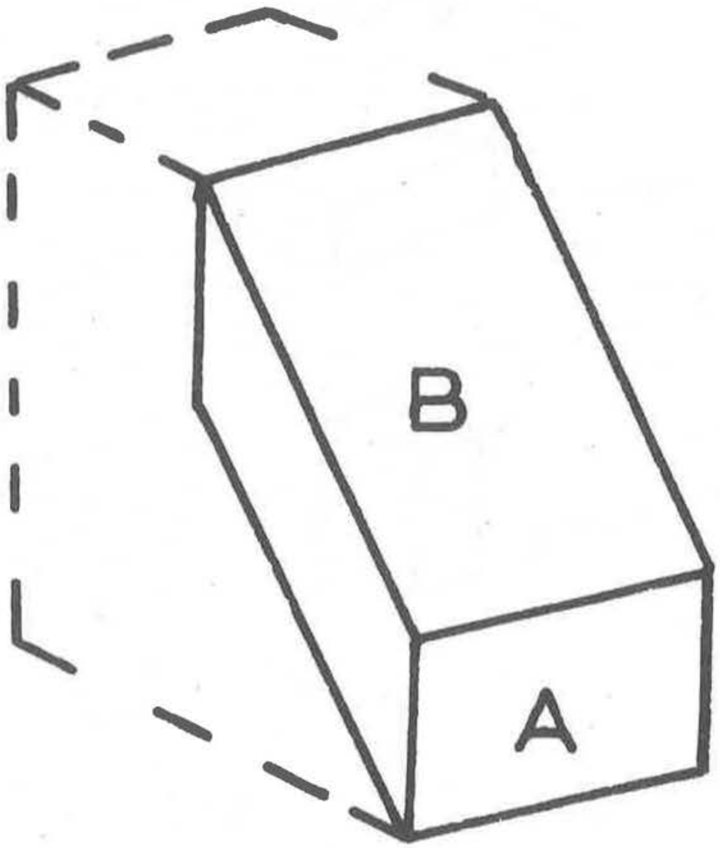

Figure 4. Seeing a transformed cube in a compound object

for line 3. The topology search then turns up a type 3 fragment: polygon A with line 9 attached. The five points defined by that fragment match a transformed cube exactly as in Fig. 3(b). This is removed from the original picture and the process continues by finding the parts shown in Fig. 3(c) and (d) with the final compound object shown in Fig. 3(e).

There are some very real difficulties with this program which can be illustrated by considering specific cases. In the example above, take the rejection of a cube model for surfaces A and B across line 2. Certainly if the projection is without perspective so that lines 1, 2 and 3 are parallel as are 5 and 6, 7 and 8 then a transformed cube fits exactly as the rectangular solid in Fig. 4 shows. This would be disastrous for the subsequent analysis. Thus Roberts' claim (1965, p.166) that "the process accounts for but does not depend on perspective information" seems to be wrong. In the perspective case the convergence of lines 5 and 6 can be used to reject it. Even assuming that the line fitting is so accurate that such fine distinctions can be made reliably, doubts must be raised about a system that depends on such distinctions.

Another example is the compound object of Fig. 5(a). Given the three basic models the program could be expected to split it into the two simple objects of Fig. 5(b). But in fact it will first remove a cuboid from the top surface as in Fig. 5(c) which leads into a muddle because it has not taken the appropriate

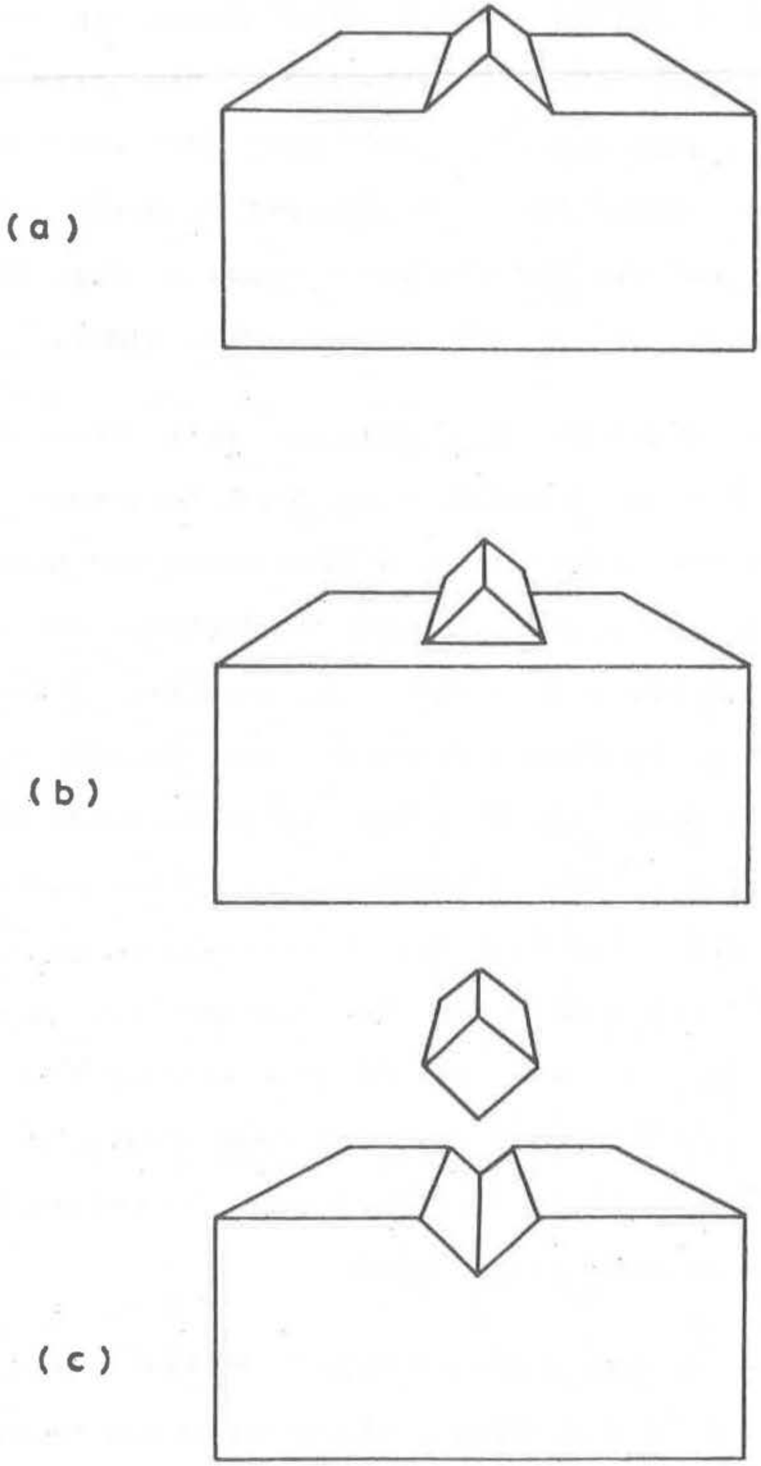

Figure 5. Two decompositions of an object

first step. This arises because the models are tested in strict sequence: cube, wedge, prism. That ordering is used to avoid splitting a cube into two wedges!

Finally consider the simple picture in Fig. 6. This object is simply a wedge on top of a cuboid. But as the program is followed through on this picture it appears that whenever the topology tests succeed the model suggested will not pass the geometric transformation test, and so the program fails completely.

The topology test finds the two quadrilaterals flanking line 4 but if one face of the cube is fitted to region A the rest of the cube will fall outside the complete figure as Fig. 7(a) shows. Attempts to fit wedges or cubes using quadrilaterals with an extra line from one corner will all fail. In particular Fig. 7(b) shows a wedge that might be thought to fit but it is incorrect as only rectangular wedges are allowed. Finally even withdrawing to just three lines from a vertex will not succeed. Looking at lines 1, 2 and 3 of Fig. 7(c) they can be seen to be three significant edges of a cube model that could be made to fit but the program does not find that context as it only looks for contexts concentrated at vertices. Finin (Winston, 1973) has defined the skeleton of a cuboid to include the sort of context needed here.

Despite the difficulties uncovered above, Roberts' program created a scene analysis paradigm that remains dominant. As a

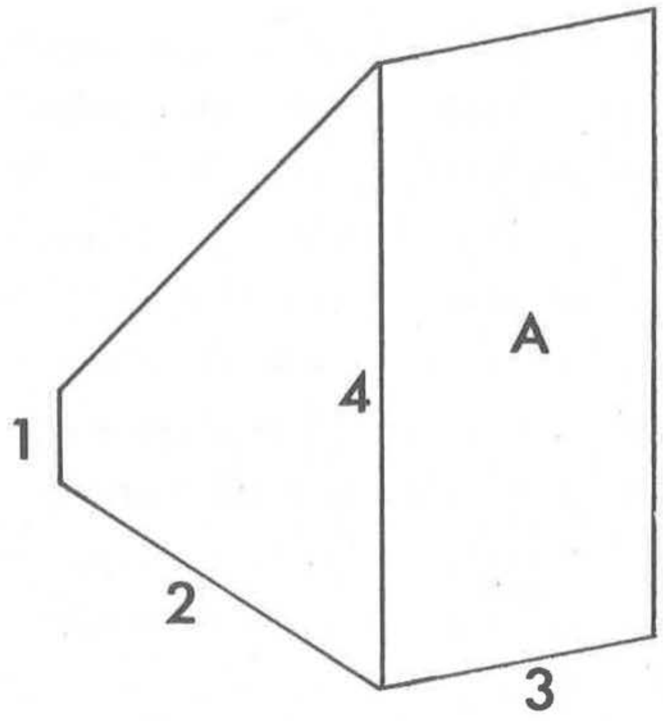

Figure 6. Another compound object

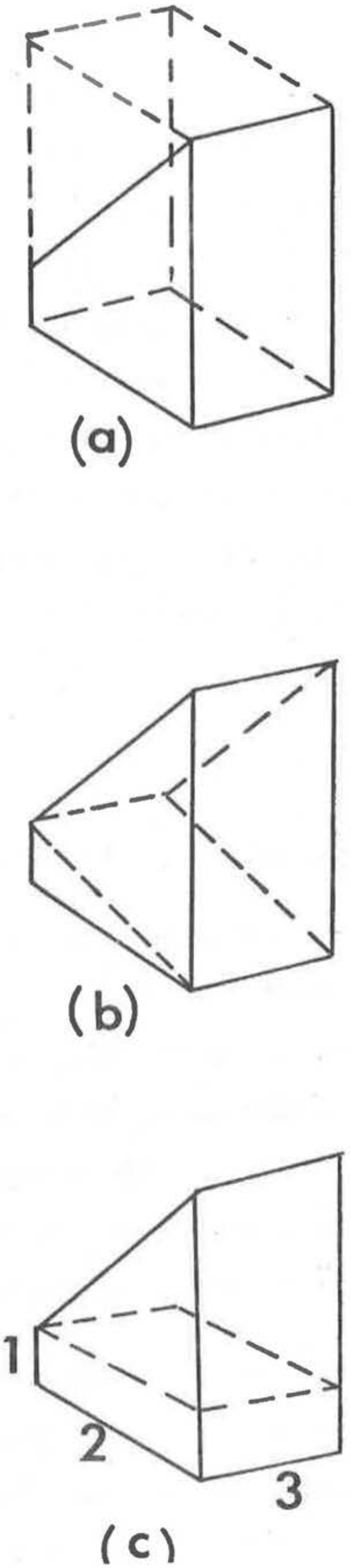

Figure 7. Possible decompositions of the object

working theory, for that is what an AI program is, it firmly established an active model of perception as a cycle of four processes: discovering cues, activating a hypothesis, testing the hypothesis, and inferring the consequences. This model of perception, so far removed from the then dominant pattern recognition paradigm for machine perception, echoes, as Clowes (1972) remarked, the approach of such psychologists as Helmholtz (Southall, 1962), Bartlett (1967) and Gregory (1974). Minsky's frame systems (Minsky, 1975) provide a semi-formalism for this paradigm of perception.

3. Guzman's body segmentation program, SEE

Guzman's SEE (1968) accepts line diagrams of polyhedral scenes as input and partitions the picture regions on the basis of the putative body membership of the surfaces depicted. The program consists of two passes over the picture. The first pass makes local guesses (called links) about which pairs of regions depict the same body. The second pass accumulates that evidence to produce a grouping of the regions corresponding to bodies.

The links are placed at the junctions shown in Fig. 8 where the links are shown as connections between two regions which are usually adjacent in the picture. An exception to these rules is the inhibition rule that no link is placed across a line at a junction if its other end is a barb of an ARROW, a leg of an L

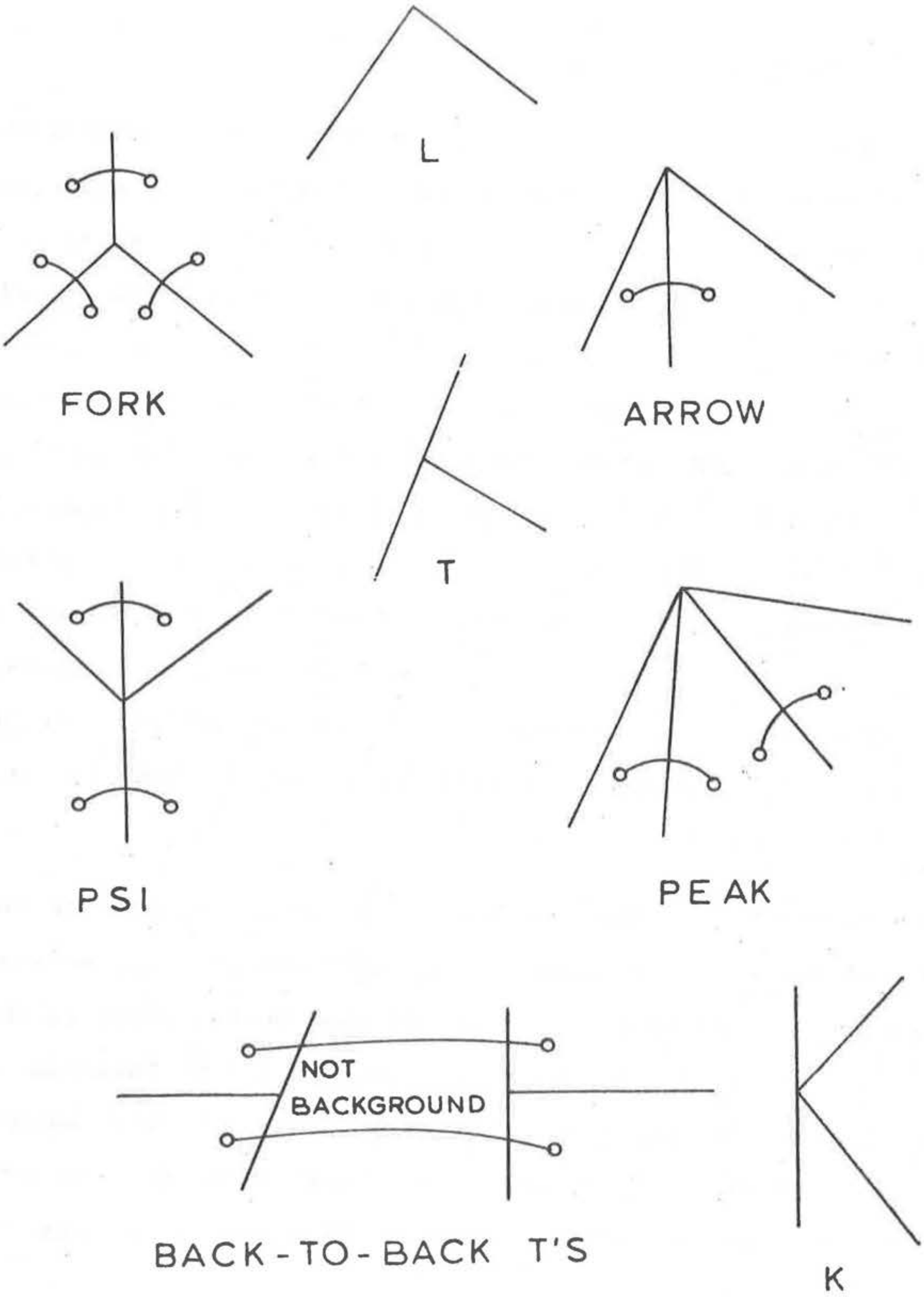

Figure 8. The junction categories and link planting rules of SEE

or part of the cross bar of a T.

Considering the result of the first pass to be a graph with regions as nodes and links as arcs then the second pass searches for 2-connected subgraphs which are declared to represent bodies. This is a highly abbreviated version of Guzman's final account which has many special case rules augmenting both passes. The rules that depend on being told which region is background can clearly be invalidated immediately by putting another block behind the scene being analyzed. That, however, is not the main point; it is merely typical of the way in which the program developed by a process of finding counter-examples that both invalidated old rules and hinted at new ones (Winston, 1973). The need to add and modify rules almost continuously to handle exceptions suggests that there is a basic flaw in the design.

The flaw seems to be that Guzman used locally computed picture predicates as evidence for global scene-based properties. To avoid this one must ask what do the lines in the picture depict? As we shall see later in the Huffman-Clowes labelling algorithm they can depict many things but only certain combinations of these things are scene coherent; this coherence decision cannot be made in the picture domain as Guzman tried to do.

SEE's tendency to see holes in objects as separate objects (Winston, 1968) is only one consequence of the fact that the program ignores ambiguities inherent in the interpretation process that are exposed by the Huffman-Clowes labelling algorithm. For example, consider Fig. 9(a) (adapted from Minsky and Papert, 1972). That can be seen in at least three different ways. The first possibility is as a simple house structure in which there is only one body. Second, as a variant of the first it can be seen as a pyramid sitting on top of a rectangular brick. Third and quite different from the first two, it could simply be two wedges abutting one another. SEE reports only the first of these alternatives and does not see the others. Moreover, SEE's interpretation consists only of "one body composed of regions A, B, C and D"; it does not provide the richness of an interpretation that reports the nature of each edge. These ambiguities and that richness are provided by the labelling algorithm (Waltz' version is needed for Fig. 9(a)) as we shall see. The labelling algorithm also detects situations illustrated by the picture in Fig. 9(b) where SEE happily partitions into bodies pictures that are syntactically correct (that is, every line bounds two different regions and so on) but meaningless as pictures of polyhedra.

An interesting comparison can be made between SEE and Roberts' program. Roberts initially hopes to find a picture fragment that corresponds to a part of one of his three prototypes so that the regions offered up should at least belong to the same body. Recalling that an acceptable polygon must be a convex region, if the first stage of the topology matching

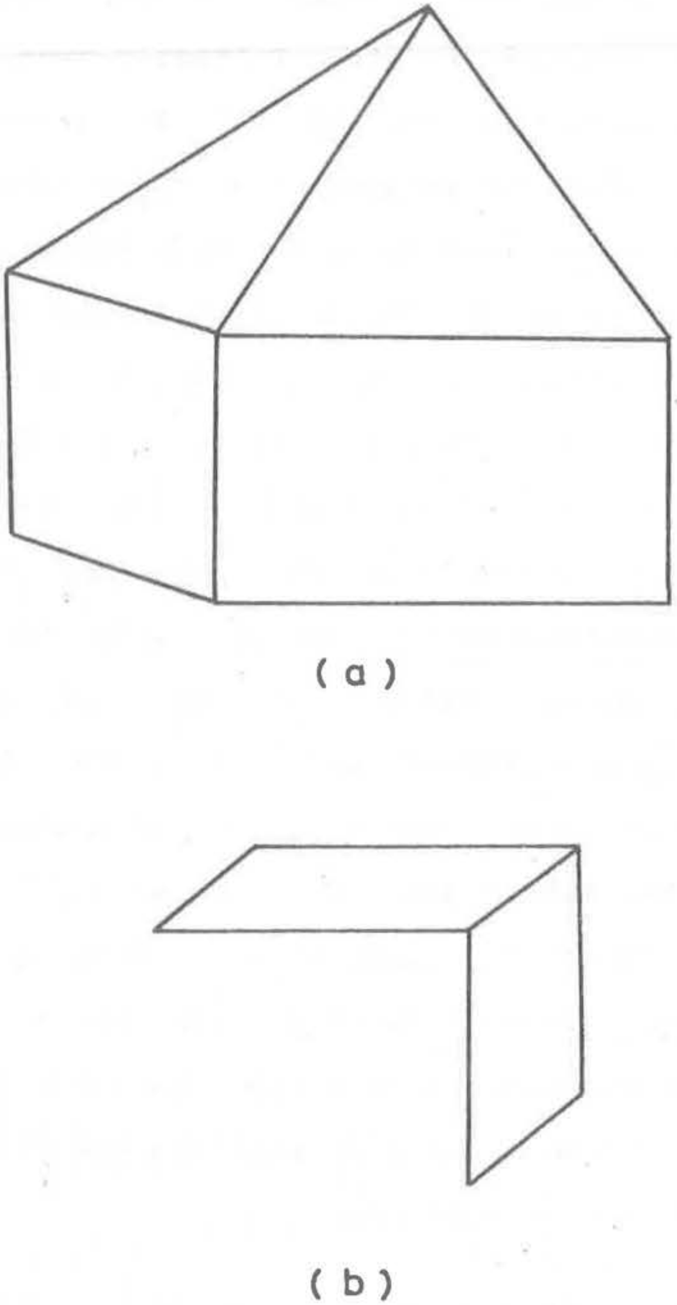

Figure 9. Illustrating a) ambiguity and b) anomaly for SEE

succeeds (3 acceptable polygons around a vertex) then it will return a FORK vertex with all three regions hopefully depicting surfaces of one body. This corresponds directly to the most powerful Guzman heuristic - the FORK that plants three links. If the first stage of Roberts' topology matching fails and the second stage (2 acceptable polygons flanking a line) succeeds then that line is almost certainly the shaft of at least one ARROW, so the second stage of Roberts' topology matching corresponds to the second most powerful Guzman heuristic linking the two regions flanking the shaft of an ARROW. Furthermore, in both the above cases, Guzman's inhibition of a link across a line at a junction if the other end of that line is a barb of an ARROW or a leg of an L corresponds directly to the convex region requirement of Roberts.

This comparison could easily be continued (consider the corresponding uses of T-junctions) but it has gone far enough to make three points beyond observing the intriguing parallels. In the first place it is now obvious that Guzman's work is not as radically new as it appeared to be. In the light of the analysis, Waltz' (1972) claim that "indeed his approach was a dramatic departure from what had been done before him" appears to be over enthusiastic. Second, we notice that Guzman did not even use such simple properties of regions as 'convex' but instead tried to express such a slightly less locally confined picture property in terms of his complicated inhibition rule based entirely on junction geometry. Third and far more important, Roberts used knowledge of prototypes explicitly in the body segmentation problem. He did this in three ways, first by using a general property (acceptable polygon) of all the prototypes, and prototype-specific topology tests to identify a picture fragment as part of a prototype and then, having made an identification, projecting the rest of the prototype onto the picture to account for many more lines. Guzman on the other hand claims to use no knowledge of prototypes in the segmentation. This claim may indeed be doubted on the grounds of the Roberts-Guzman parallel presented here. SEE seems to prefer convex regions as body faces. This is confirmed in the analysis of SEE's underpinnings in section 6. This claim to virtue (as it was seen by Guzman) in fact turned out to be an objection to SEE as it led to a vision system that was pass-structured with successive passes mapping into progressively more abstract domains (Minsky and Papert,1972).

4. Falk's scene analysis system: INTERPRET

Falk's (1972) collection of scene analysis programs operating as a system called INTERPRET represents a gathering together of the state of the art in scene analysis *circa* 1970. Given a range of nine fixed size prototypes that appear in the world (Fig. 10) and the position and orientation of the ground plane relative to the picture plane, the system is required to interpret line drawings (with, possibly, a small number of lines

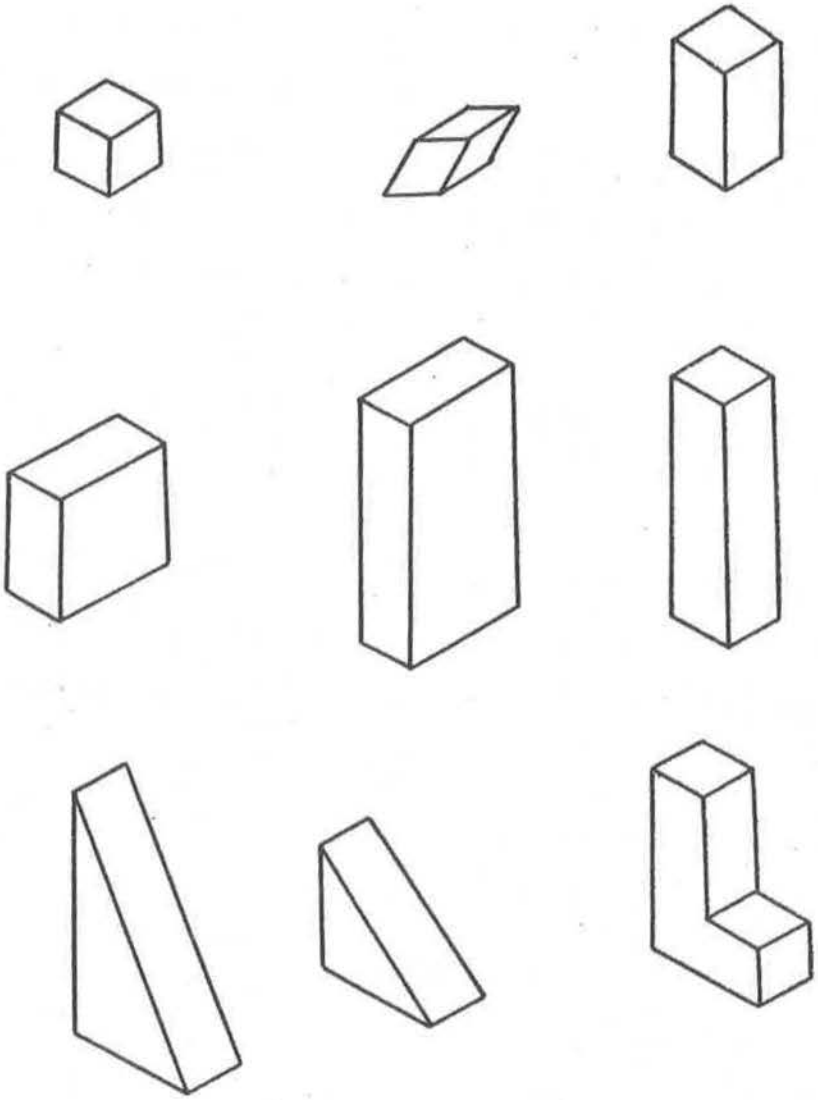

Figure 10. The object prototypes of INTERPRET

missing) to produce an exact 3D representation of the scene.

The system consists of the five stages of Fig. 11. SEGMENT partitions the set of picture lines into bodies. For each body, SUPPORT determines the set of bodies that could conceivably support it. COMPLETE tries to add lines to the picture of each object so that RECOGNIZE will find it easier to identify it as one of the prototypes. RECOGNIZE also determines the position of the prototypes so that PREDICT can say what the picture should look like. Finally VERIFY determines if the predicted and given picture match. The system is strictly pass structured with the five stages called in sequence with the exception that a failure in VERIFY requires RECOGNIZE to produce another suggestion.

SEGMENT used Guzman-type vertex classifications to assign edges to bodies. It assigns edges rather than regions as SEE did because the possibility of edges not being depicted means that a single region could correspond to two surfaces of separate bodies. Each Guzman vertex category is split into two: GOOD<category name> and BAD<category name> on the basis of local context that can include adjacent junctions. The hope is that, for the most part, GOOD junctions show edges of only one body while BAD junctions show edges of more than one body. As an example of the GOOD/BAD distinction, an ARROW is a BADARROW if one of the regions flanking the shaft is background or if the shaft is the top of a K junction, otherwise it is a GOODARROW.

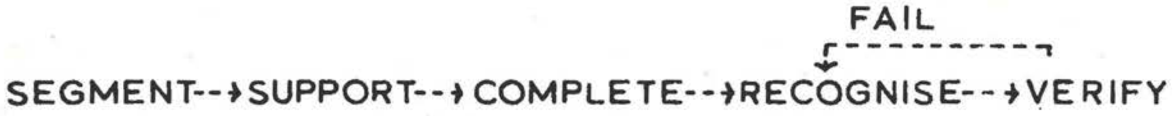

Figure 11. The organization of INTERPRET

The next step determines sets of lines such that each set connects a group of GOOD vertices. Each set then represents edges of a single body. The total set of lines thereby assigned does not necessarily exhaust the set of lines in the picture. SEGMENT then assigns regions to bodies based on the line segmentation and a few extra heuristics for splitting regions that correspond to more than one body.

RECOGNIZE needs to know which bodies in the scene could support other bodies because it infers the position of each body from the position of the body supporting it, that is, working up from the known position of the table. SUPPORT creates the set of potential supporters for each body. It starts by establishing which are the base edges of each body by applying six elimination filters to the set of exterior lines for each object. For example, eliminate both lines at downward open L vertices. These filters all depend on the local picture geometry of each line. SUPPORT then defines the potential supporters for the body as those bodies that have a face appearing adjacent to one of the base edges. If a body has only one potential supporter then that must be the actual supporter. In particular for objects supported by the background surface, RECOGNIZE will be able to establish the 3D position of the endpoints of all the base edges.

The picture of each object may be incomplete for three possible reasons: (a) the original picture had some lines missing or (b) the object is partially occluded or (c) SEGMENT failed to assign some lines to the body. COMPLETE has three routines that attempt to patch up each object before recognition. Fig. 12 shows dotted lines where ADDLINE, JOIN and ADDCORNER fill in lines. ADDLINE seems intended for case (a), JOIN and ADDCORNER for case (b). ADDLINE puts a line between two L vertices that open upwards and have parallel arms.

INTERPRET does not recognize an object until all its potential supporters have been recognized. Then the potential supporter with the highest horizontal surface is identified as the actual supporter for that object. The end points of all the base edges of the object can then be located in 3-space.

RECOGNIZE attempts to name an object by matching features of its line drawing against the stored properties of the prototypes. A succession of tests is applied to the prototypes until, hopefully, only one remains. If the line drawing is complete (which is determined by a simple heuristic picture topology test) then the first test looks at the number of visible faces and vertices, otherwise the topology of the complete visible faces is used. The second test compares lengths of base edges while the third test compares angles between the base edges. The fourth test assumes that lines vertical in the picture correspond to vertical edges if they are not labelled as base edges. The length of such an edge can be calculated and compared with the prototypes.

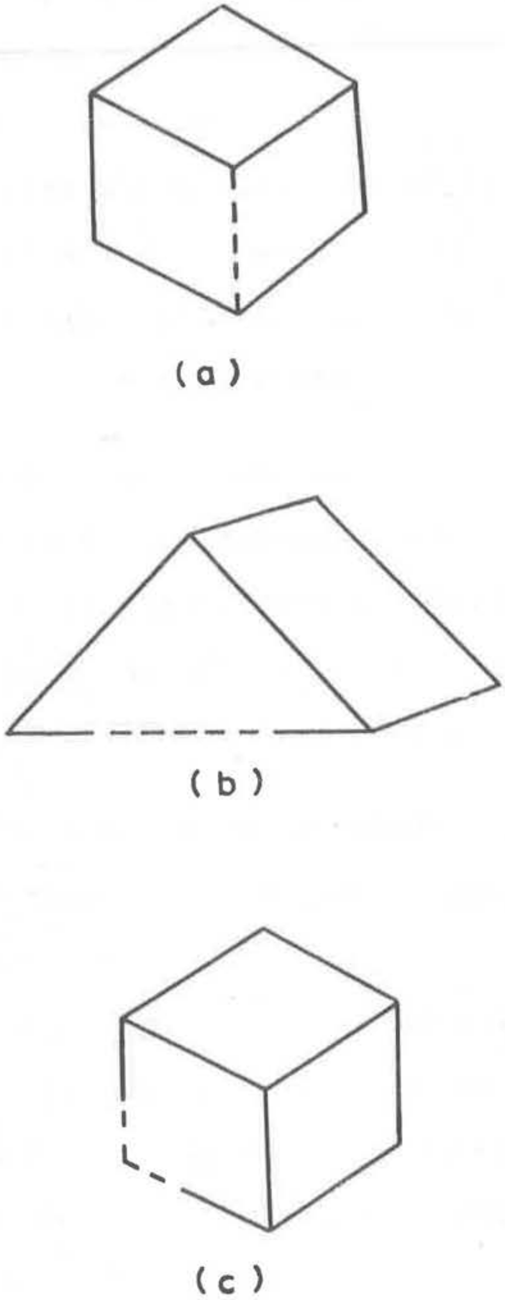

Figure 12. The contexts for COMPLETE

When the object is named and three corners of the base edges of it are located in space then the object is positioned by identifying three corresponding points on the prototype.

VERIFY predicts the picture appearance when every object has been recognized and located. If a body has more than 3 lines in the prediction that do not appear in the input or if there are any lines in the input that have not been predicted then VERIFY reports back to RECOGNIZE and asks for a new suggestion.

Falk's program is a good attempt at overcoming imperfect line data but, as he has taken from Guzman an almost total reliance on local picture-based heuristics, INTERPRET is open to the objections raised against SEE above. In fact, Falk extends their usage beyond body segmentation to include support and completion heuristics of the same general nature. To demonstrate the problems involved, we will present for each of those stages of INTERPRET a specific example of a picture where the program (at least, that version of it described in (Falk, 1972)) appears to go astray. These simple examples using only Falk's prototypes are not malevolently constructed using degenerate views or unlikely alignments, nor can the problems be attributed to insufficient data as the pictures are perfect line diagrams (except for the one missing line that COMPLETE should insert).

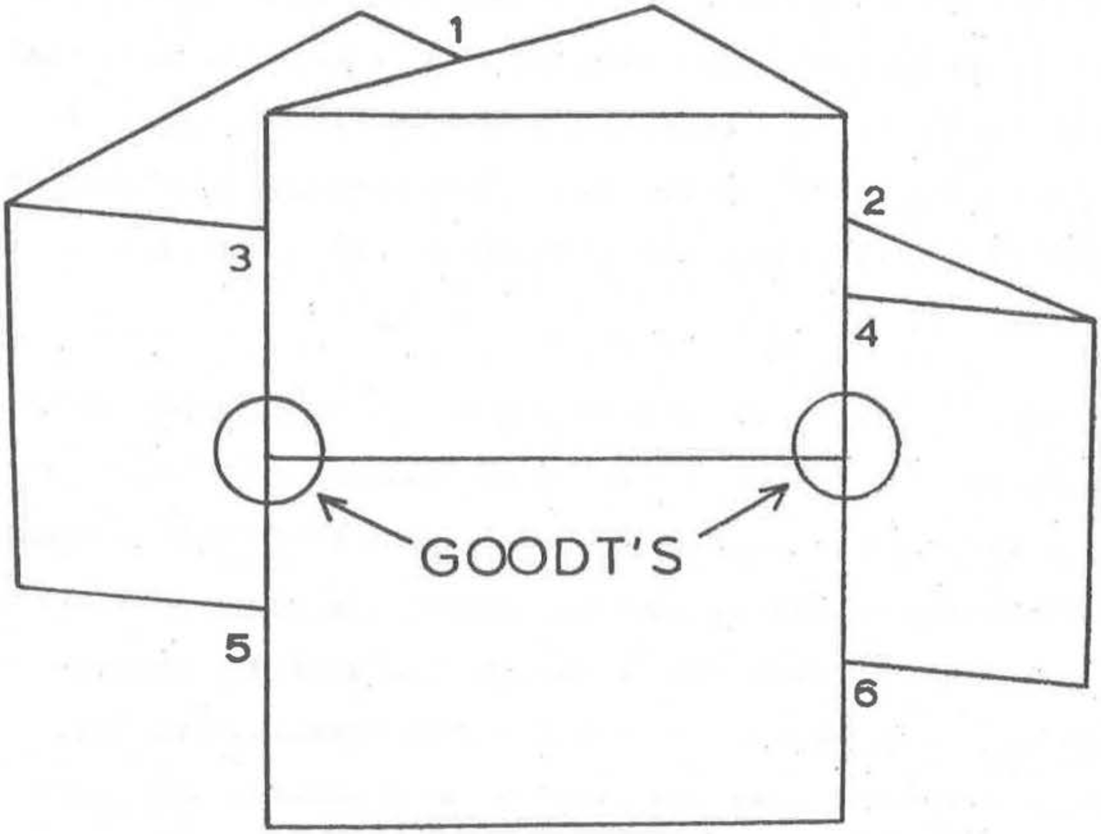

Figure 13. Illustrating SEGMENT

SEGMENT finds only 2 bodies in Fig. 13. It matches the back-to-back T's of the partially occluded wedge to get one body, (that is, it matches junction 1 with junction 2, 3 with 4, and 5 with 6) but the two stacked wedges in front are seen as one body because the 2 circled junctions are both classified as GOOD T.

SUPPORT eliminates line 1 of Fig. 14 as a base edge of that wedge because it is a line at a downward open L vertex.

Finally in Fig. 15 there is a line missing from the picture of an L-beam. COMPLETE has a routine ADDLINE to deal with this. ADDLINE is activated by a context of a pair of L vertices with parallel sides. In Fig. 15 there are two such contexts: AB and BC. The first context to be picked up is not defined but if it is AB and ADDLINE puts a line between A and B it destroys the second context, BC. Regardless of which context is found first, ADDLINE certainly has no way of knowing that line BC makes more sense than AB because in the picture domain there are no grounds for preferring one over the other; both are correct as pictures.

"makes more sense" is a remark that applies not to the picture itself but to what is depicted, the scene. Similar comments apply to the failures of SEGMENT and SUPPORT and so it becomes clear that the program must have some kind of 3-dimensional interpretation before evaluating predicates such as 'same body', 'supports' and 'missing edge'. But the only way Falk has of getting a 3D interpretation is by recognizing the

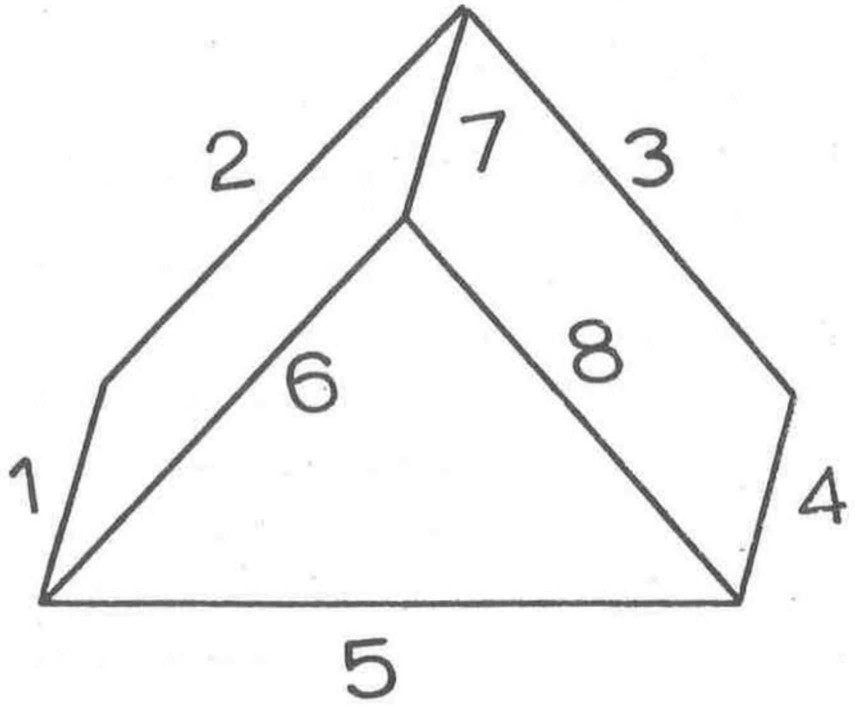

Figure 14. Illustrating SUPPORT

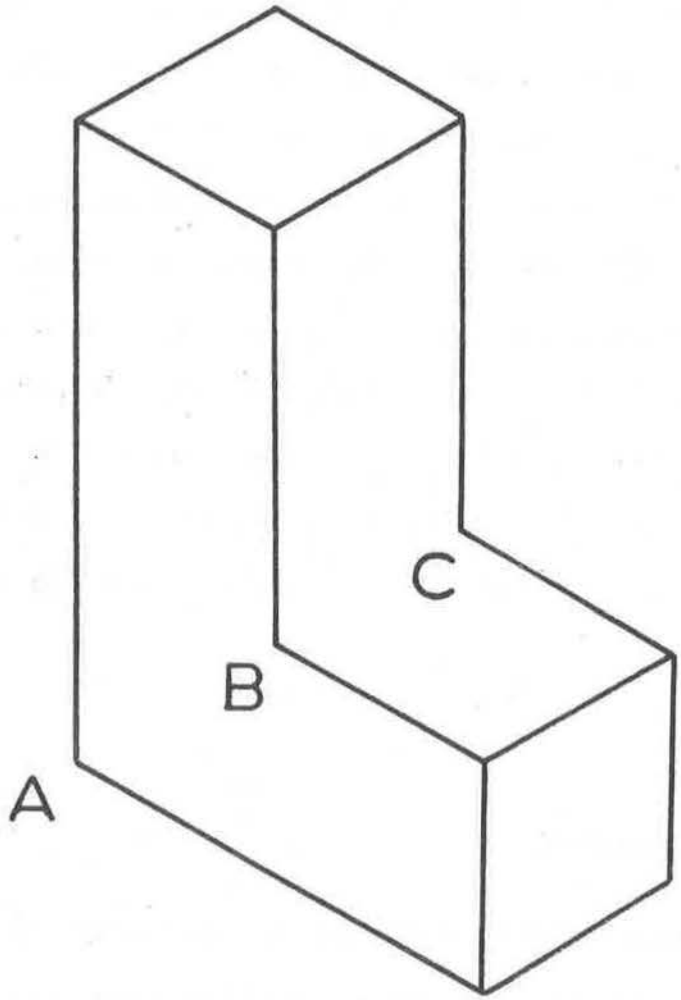

Figure 15. Illustrating COMPLETE

objects. This is a chicken and egg problem: the program needs to recognize the objects to get a 3D grip on the scene in order to recognize the objects.

The way to break this circularity is to realize that recognition, that is, the identification of an object as a particular member of a set of prototypes, is not the only way of getting a grip on the scene. There are general principles about the picture-taking process and the nature of opaque polyhedra that one can incorporate in a procedure to interpret line diagrams that does not use any specific prototypes. Huffman (1971) and Clowes (1971) working at the same time as Falk independently proposed such a procedure which can now be seen as a step towards the solution of the chicken and egg problem of scene analysis.

5. The linguistic approach

Before we examine that procedure, another approach to picture processing must be mentioned. In the nineteen-sixties a scattered group of people were trying to find suitable representations for picture descriptions as suggested by Minsky (1961). Struck by the persuasive analogy between pictures and natural language and influenced by Chomsky's (1957, 1965) account of syntactic structures, some, such as Kirsch (1964), Ledley (1964), Narasimhan (1966) and Anderson (1968) wrote grammars for restricted classes of pictures while others such as Clowes (1969), Evans (1969), Shaw (1969) and Stanton (1970) attempted more general picture description languages. Like all analogies the linguistic approach eventually collapsed and died (for the obituary notice and postmortem see Stanton (1972) and Clowes (1972a)) but it left a legacy of insights. For example, following Chomsky's emphasis on the uses of anomaly, a common technique in the linguistic approach exploited pictures of impossible objects in order to tease out the rules whereby we assign structure and meaning to pictures. Both Huffman (1971) and Clowes (1971) used this technique to examine the interpretation of line diagrams as polyhedra.

6. The Huffman-Clowes labelling algorithm

As we remarked earlier Guzman's SEE somewhat surprisingly deduces body membership of two surfaces from the appearance of the corners that they share. The most obvious question to ask is: why does it work? Another question might be: what else can we infer from the junction geometry? The answer to the latter question will indeed help us answer the former. To start with we note that it makes more sense to infer local (rather than global) scene properties from local picture evidence. In particular if we rely on the shape of junctions as evidence we should be making inferences about the corners they depict. Restricting themselves to 2-line and 3-line junctions and

3-surface corners, Huffman and Clowes observed that each Guzman junction category must have one of a small number of corner interpretations which are described by the predicates convex, concave and occluding which apply to the edges meeting at the corner. In Huffman's notation, + labels a convex edge with both surfaces visible; - labels a concave edge and an arrowhead labels an occluding edge that belongs to the surface on the right (as you move in the direction of the arrow). The surface on the left is behind the edge and partially occluded by the surface on the right.

Fig. 16 shows the interpretations for each legal junction type (L, FORK, ARROW, and T). For all but the T these interpretations are actually corners. Considering all four possible labellings for each line gives 4^2 = 16 for the L, 4^3 = 64 for the others as against the reality of 6 for the L, 5 for the FORK and so on; hence, it is apparent how useful these legal corner interpretations could be. In order to use this table of interpretations the only further scene coherence rule is that an edge must have the same interpretation at both of its visible endpoints. The labelling algorithm described by Clowes starts with the background region and constructs all interpretations in parallel whereas Huffman suggested a depth-first search, backtracking when coming upon a junction that has no interpretation consistent with the labels that have already been placed on some of its lines. Both procedures not only label the edges of the scene but also recover some of the hidden structure

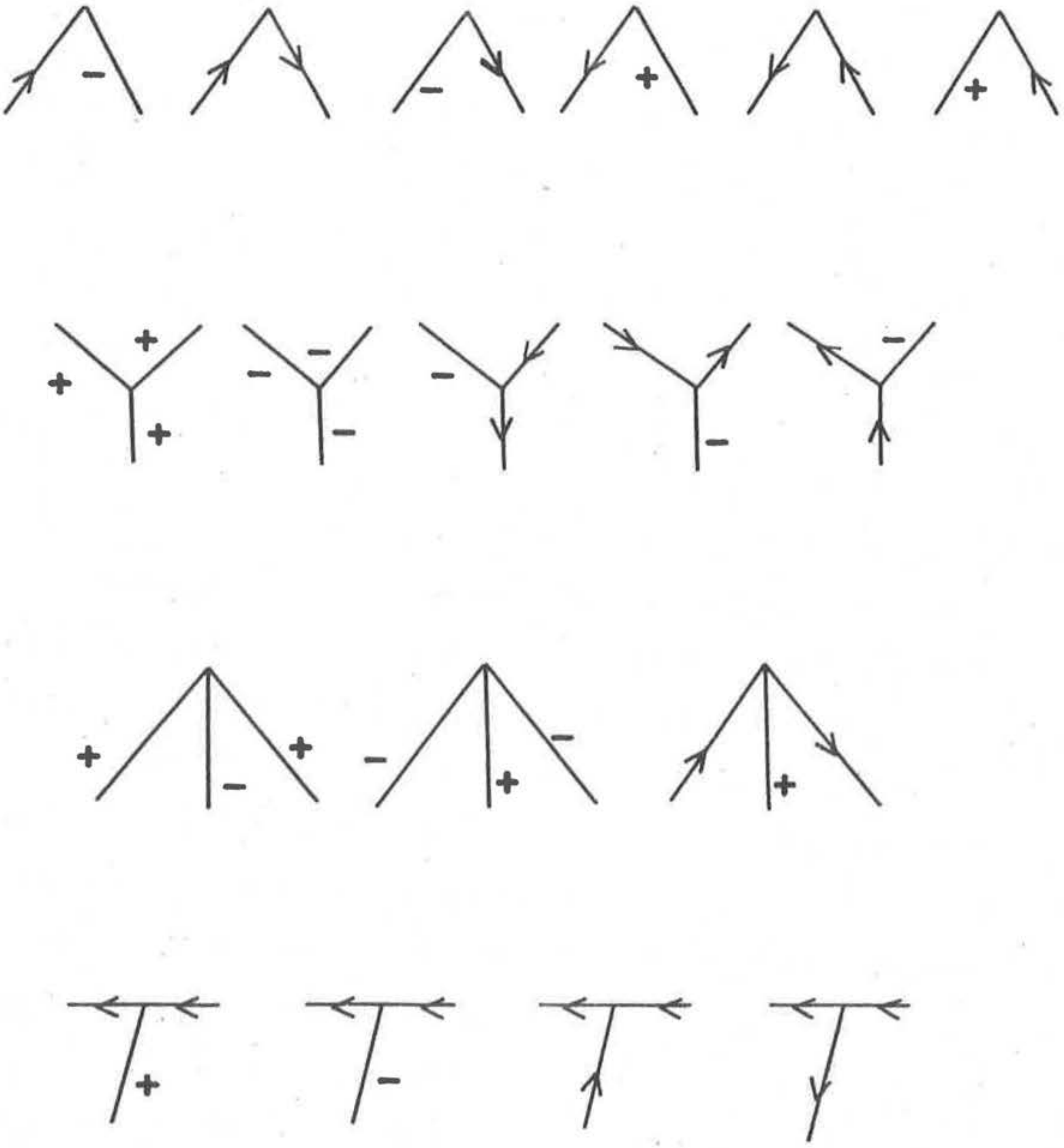

Figure 16. The Huffman-Clowes junction interpretations

in that occluding edges have attached to them surfaces that are turned away from the viewing direction.

There are several reasons to judge this algorithm to be an important step forward in scene analysis. Let us start with impossible objects. There is theoretical satisfaction in having a procedure that returns no interpretations of a picture such as the one reminiscent of the devil's pitchfork, Fig. 17 (taken from Clowes, 1971), if we ourselves cannot assign a plausible three-dimensional interpretation. But this ability would also be of practical use in a scene analysis program. Fig. 9(b), which SEE happily accepted and parsed, can be rejected as a candidate for object status because it cannot be labelled. This is a sufficient but unfortunately not necessary condition that the object be impossible as Huffman showed. But to be able to make this discrimination suggests that the method has greater descriptive power than the only other prototype-free program, SEE. A comparison of the scene description generated by this algorithm with that given by SEE shows how true that is. Here we have edges known to be convex, concave or occluding, the visible part of a surface defined by edges belonging to that surface or to another known surface and some conclusions about hidden surfaces that share an edge with a visible surface.

The question "Why does SEE work?" can now be answered in detail. Suppose that we were only concerned with convex objects, then from the set of corner interpretations used by the labelling algorithm (Fig. 16) eliminate all corners with concave edges, including those for the L that imply a hidden concave edge, leaving the set of Fig. 18. Notice that the L, FORK and ARROW junctions now have unique corner interpretations. The concave edges that appear when one body abuts or rests upon another are here taken to be occluding edges as they would be if the bodies were slightly separated. In this world of convex polyhedra, convex edges (+) join surfaces of the same body while surfaces of different bodies appear at occluding edges (> and <) so using this corner set a body partitioning is easy to achieve. That's what Guzman did! The links were planted at unambiguously convex edges. The link-planting rules of Fig. 8 are derived from the corner interpretations of Fig. 18 by replacing + by a link and occluding by no link. The link suppression rules, "no link is placed across a line at a junction if its other end is a barb of an ARROW, a leg of an L or the crossbar of a T", can be seen from Fig. 18 to suppress a link across an edge if its other end shows it to be unambiguously occluding. The accumulation of link evidence relies on 2 links between surfaces which means in effect that both ends of an edge must agree that it is convex for it to be so taken as in the Huffman-Clowes algorithm. If only one end says so there is a conflict which must be heuristically resolved. This provides a scene-coherent account of why Guzman's picture-based heuristics worked and incidentally explains why SEE doesn't work on concave objects (Winston,

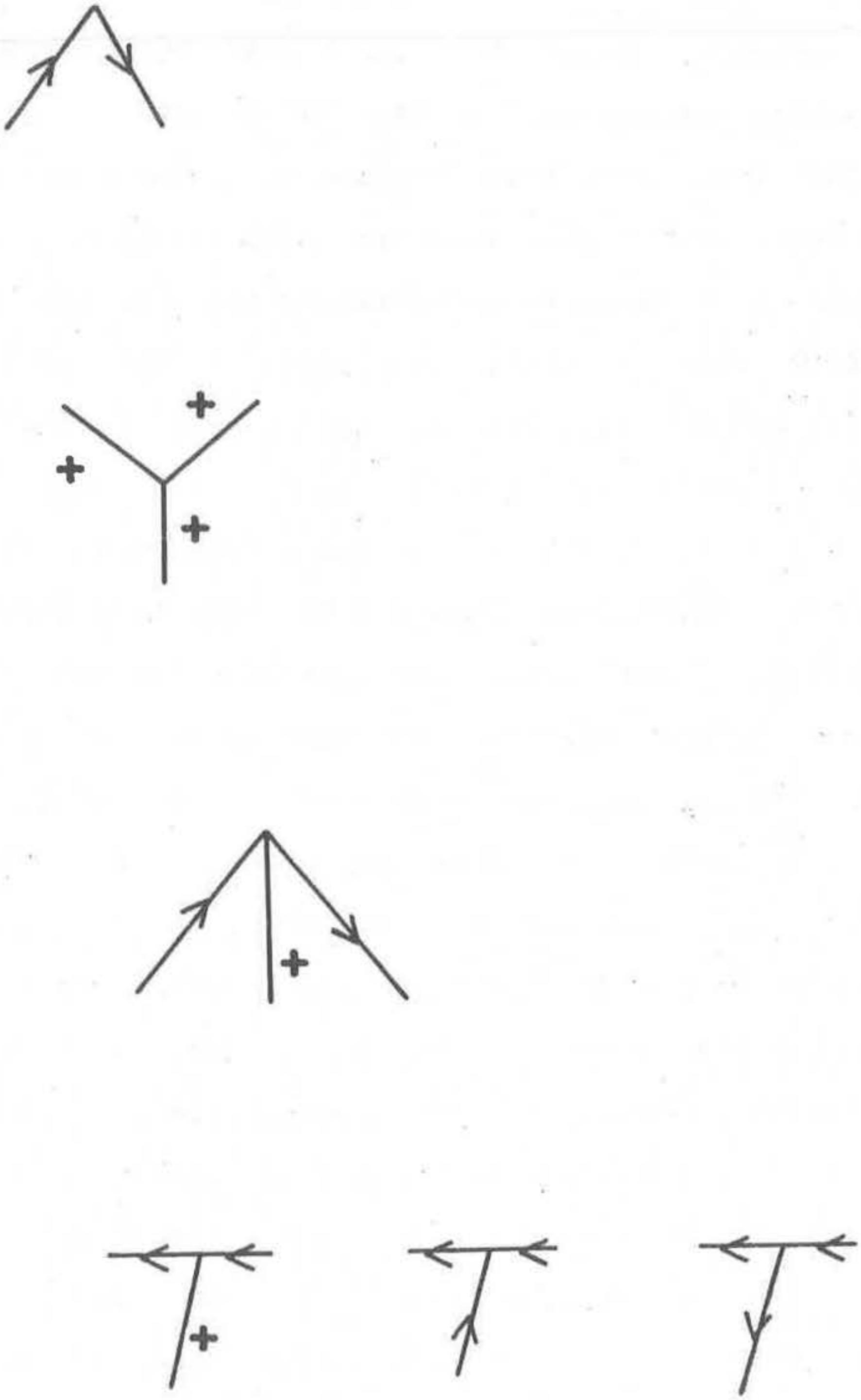

Figure 18. The junction interpretations for convex polyhedra

1968).

The next step is to use the scene as labelled by the Huffman-Clowes algorithm as a more reliable basis for body segmentation. A first guess might say: the visible aspect of a body is a maximal set of surfaces joined by convex or concave edges. This isn't quite right because by that criterion the labelled cube in Fig. 19 is part of the same body as the background, by virtue of the two concave edges. Such concave edges define body boundaries. Waltz (1972) as we shall see called them "separable" and used a further subcategorization of concave edges to solve this segmentation problem.

Returning to Falk's INTERPRET, the labelling algorithm is considerable potential help in solving the chicken and egg problem. Consider the three stages where INTERPRET was seen (in section 4) to get into trouble: SEGMENT, SUPPORT and COMPLETE. The above discussion of a scene-based approach to body segmentation applies to the problem with SEGMENT. The specific problem illustrated in Fig. 13 requires more interpretations for the T junction than shown in Fig. 16 but the extension is straightforward as will be shown in the discussion of Waltz' program.

SUPPORT rejected edge 1 of Fig. 14 as a potential base edge. A labelling of that picture gives edges 1, 2, 3, 4 and 5 as occluding edges and 6, 7 and 8 convex. Furthermore edges 1, 5 and 4 are attached to a single hidden surface while edges 2

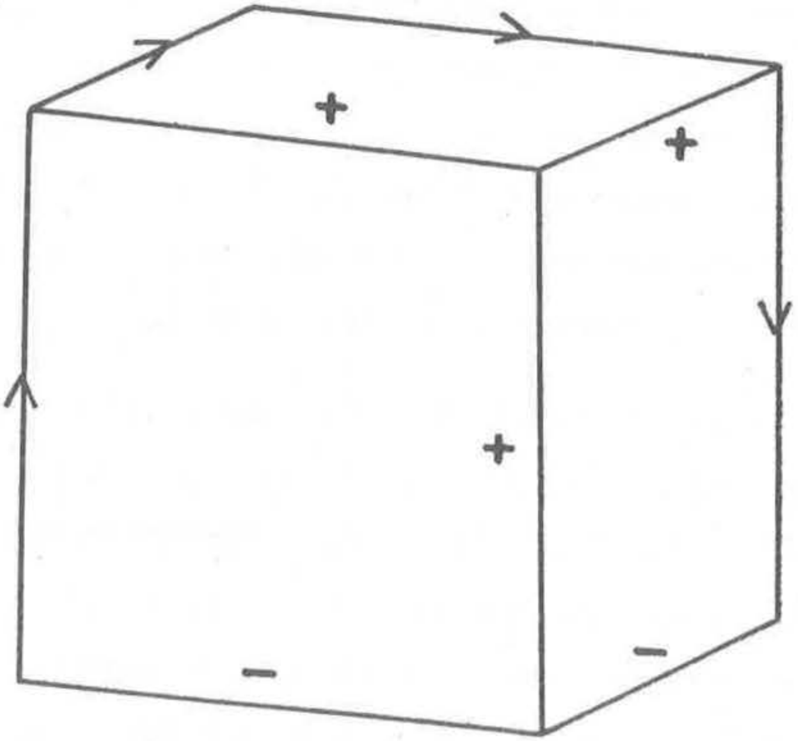

Figure 19. An interpretation of a picture

and 3 are attached to a different hidden surface of the same body. A support algorithm given that information only has to decide that the former surface is the support surface.

The first thing COMPLETE should do is decide if an edge is in fact missing. If the object cannot be labelled then that must be the case. For Fig. 15 no labelling is possible as shown by the conflict at the circled junction of Fig. 20(a). That labelling for that junction is not a legal interpretation of an L (see Fig. 16). Since lines can only be added to the picture and junctions in a picture of a single body are not allowed more than three lines, a line must be added to the circled junction of Fig. 20(a) joined to either of the facing L junctions. Either of the lines AB or BC can be inserted and the picture labelled as Fig. 20(b) and Fig. 20(c) show but clearly only (c) makes sense in terms of the prototypes. This leads us to consider the matching procedures in INTERPRET. They should operate in a domain of surfaces (visible and hidden), corners and edges (convex, concave and occluding) rather than directly in the picture, as do the picture topology matching routines of RECOGNIZE and VERIFY. Besides being more sensible, matching in the scene domain is also clearly more efficient because the program has richer structures to compare. For example, a match could be quickly aborted in the scene domain if an edge were of the wrong type.

The labelling algorithm does not sweep away all the

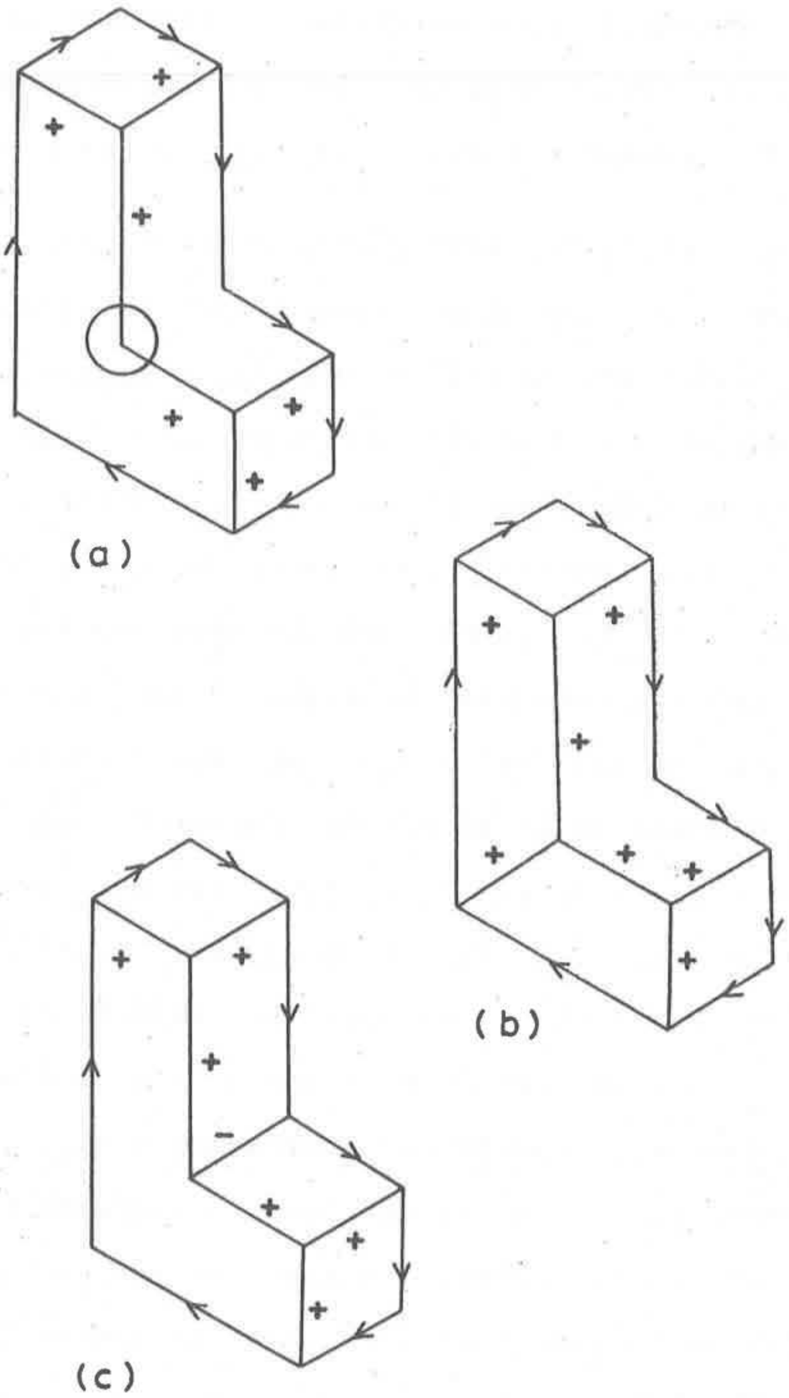

Figure 20. Completing a picture of an object

difficulties in Falk's program but it points in the right direction; however, there are some problems with the labelling algorithm as described here. It can make mistakes. In Fig. 21(a) it incorrectly labels a legitimate view of a cube (it will of course produce all the correct labellings as well) and in Fig. 21(b) (adapted from Huffman,1971) it labels an object that cannot be a polyhedron with planar surfaces. Both sorts of mistakes can be avoided by an extension of the labelling algorithm: if two lines (a and b) shared by a pair of regions (A and B) are not collinear then the lines cannot both depict convex or concave edges. But that ad hoc extension evades the key issue which is that the algorithm has no requirement that surfaces be planar nor is there any way that it can by systematically introduced without radical changes in the algorithm. Beyond saying that a surface cannot change from visible to hidden (unless, of course, it is partially occluded) there is no coherence required of a surface. This can be further illustrated by noting, as Huffman did, that the algorithm finds a labelling for the impossible triangle of Penrose and Penrose (1958). That object can only be realized if some of the surfaces are highly skewed.

In order to handle some other problems which arise such as many-surface corners, alignments of bodies in the scene, coincidence of viewing direction and object surfaces, shadow edges and so on, does one simply add ad semi-infinitum to the lists of corner interpretations? Waltz has shown that that is

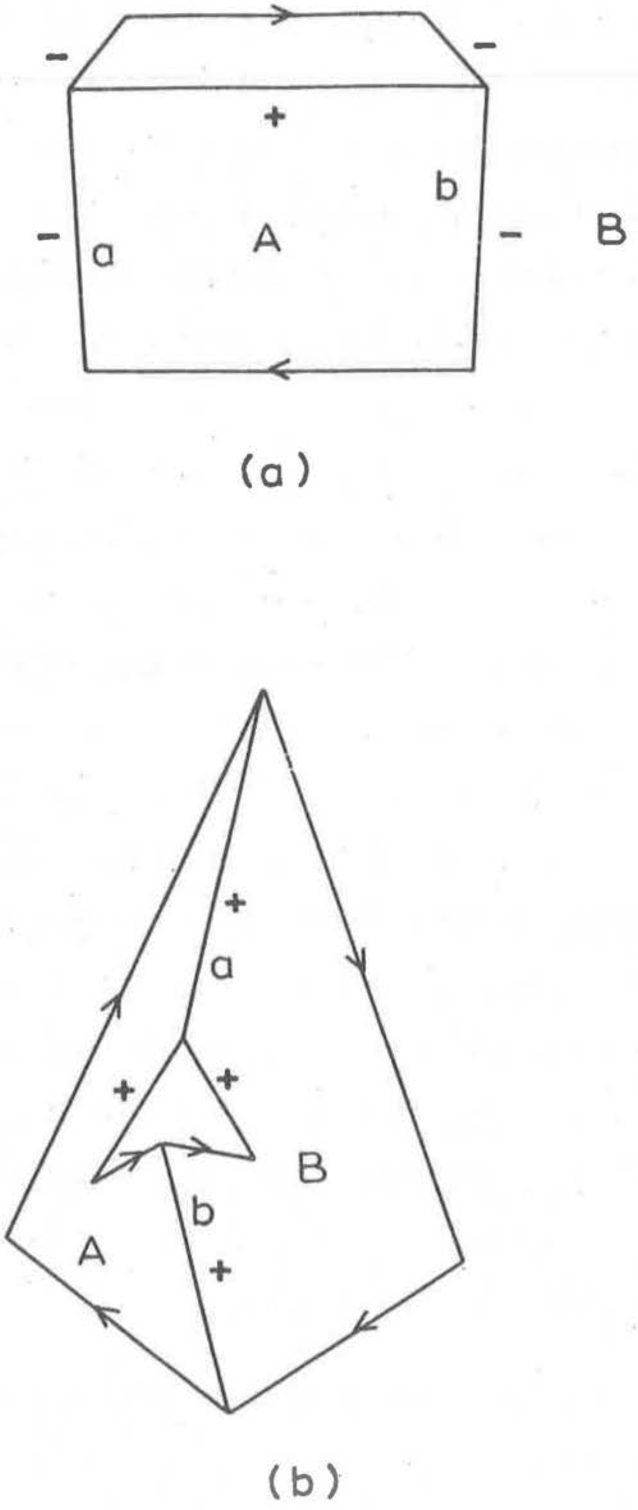

Figure 21. Labelling problems:
a) An anomalous interpretation of an object
b) An interpretation of an anomalous object

in fact a partial answer to those problems.

7. Waltz' extension of the labelling algorithm

Waltz made two important contributions to the labelling algorithm. He expanded the set of line labels from the four used by Huffman-Clowes and he improved the mechanism of search for coherent interpretations.

His first addition to the set of possible edges was the crack - a flat edge. Next, he noticed that the visible boundaries of objects usually appear at occluding or concave edges or at cracks. To account for this he subdivided the concave and crack edge categories into separable and non-separable. An edge is separable if two or three bodies meet there. All cracks are separable but some concave edges are internal edges of a body. A separable edge has, in addition to its concave/crack label, labels that show the status of the edges of the separate bodies.

The other expansion of edge possibilities derives from a crude account of lighting. Asuming a single concentrated light source then surfaces are either illuminated, turned away from the light (self-shaded) or shaded by a shadow cast by another surface. Waltz expanded the line labels to give the illumination status of the two surfaces appearing at the edge and allowed lines to depict shadow boundaries as well as real edges. The number of possible line labels has increased from the original 4 to 53.

Following a graphical representaton used by Winograd (1972) to depict the networks of features associated with grammatical units by his systemic grammar, we can more easily see the structure of the set of possible interpretations of a line in the network of Fig. 22. In that network the choice of illumination status for each surface has not been shown so there are only 11 distinct line interpretations.

Turning to the possible corners and their picture appearance, Waltz used the Huffman-Clowes junction categories and also all 4-line and some 5-line junctions. Following a straightforward procedure, Waltz considered all possible object configurations viewed and lit from all possible octants to generate the possible corners list for each junction category. The length of the corner list for each category varies from 10 to 826 with a grand total of 3256. The actual corners are all either trihedral or formed by more than one convex trihedral object but he also includes some interpretations of junctions formed by accidental alignments in the scene.

With so many possible corners for each junction, Waltz realized that time and space limitations rule out a simple depth or breadth-first search, so he devised a more efficient two pass procedure. The first pass through the junctions, the filtering procedure, is a modified breadth-first search that weeds out the

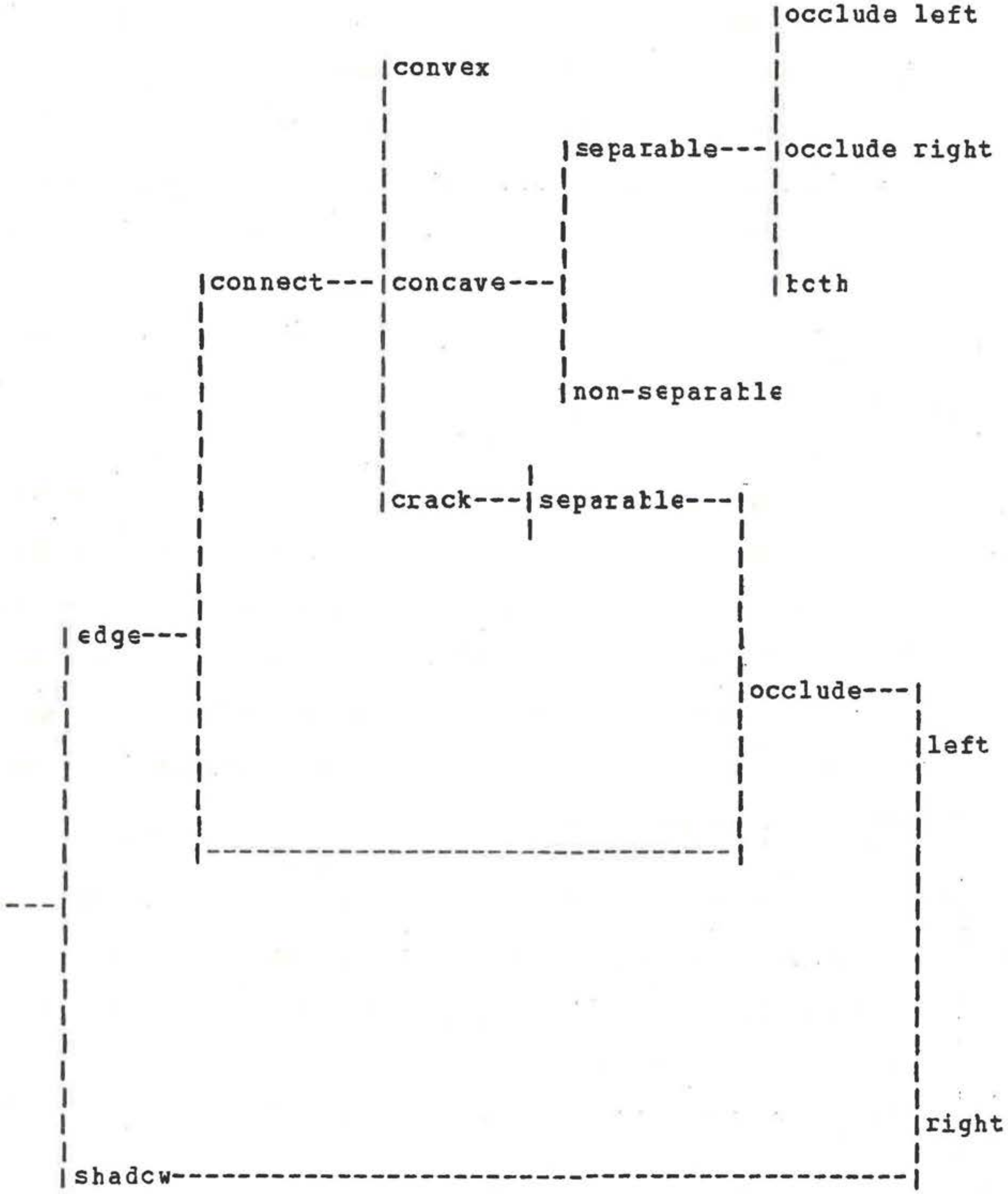

Figure 22. A network of the interpretations of a line

possible corner list for each junction by checking in the lists of every adjacent junction that has previously been processed for at least one corner with the same label for the connecting line. If that check is not successful then that possible corner is weeded out of the list for that junction. This discarding causes the program to reconsider junctions it has already looked at so the discarding action may have an effect that propagates through many junctions. Since this procedure does not actually construct complete interpretations as it goes, it need not find all pairs of corners with the same label for the connecting line as Clowes' procedure does; hence, it avoids 'the intermediate expression bulge' of the earlier procedure. This weeding process drastically reduces the possible corner lists so that the second pass can easily backtrack to find complete interpretations without requiring exponential time as Huffman's procedure does. For a detailed treatment and extensions of this and related algorithms see Mackworth (1975).

Fig. 23 shows a typical scene labelled by Waltz' program. The convex and occluding edges are shown as they were for the Huffman-Clowes labelling. The concave edges here are separable so they are additionally labelled with an occluding arrowhead indicating the sense of occlusion the edge would have if the object were picked up. Cracks are labelled with a C and a similar occlusion arrowhead. Shadow boundaries are shown with arrows pointing across the line into the shadowed region.

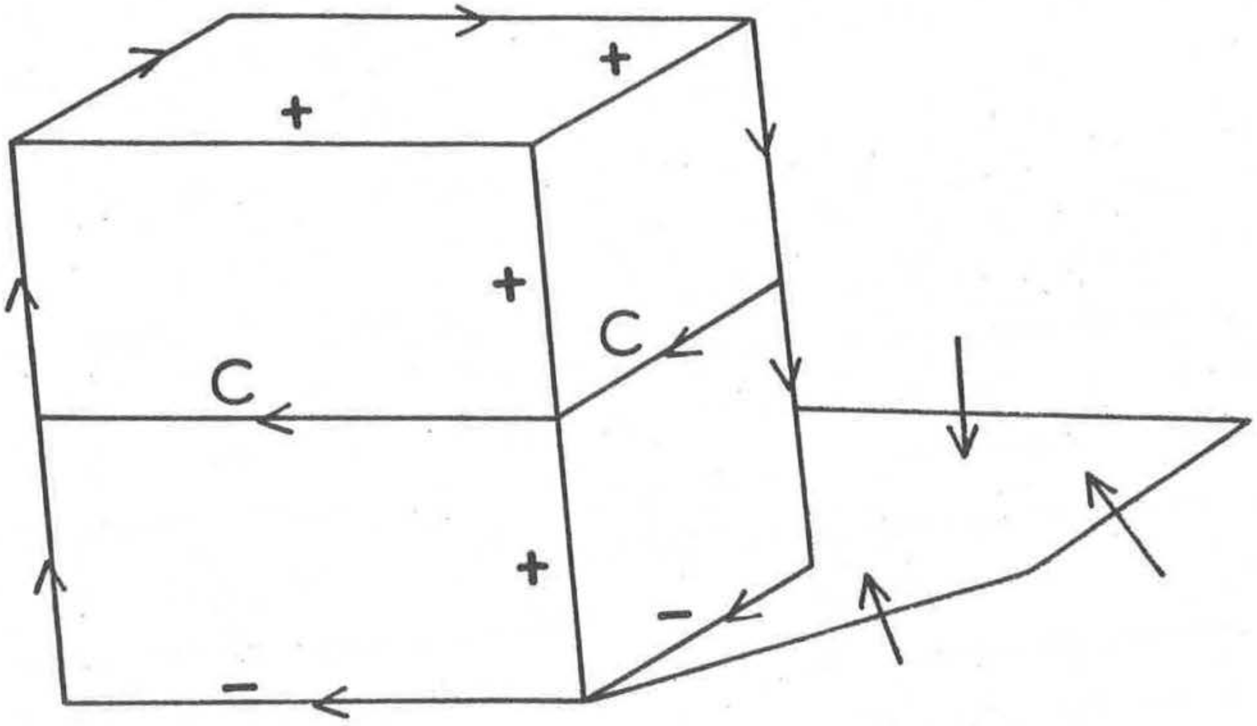

Figure 23. A scene labelled by Waltz' program

Waltz' achievement was to show that the labelling technique can be extended to handle more realistic scenes then previously although it has yet to be incorporated in a scene analysis program using grey scale picture data. Most of the remarks made above about the Huffman-Clowes procedure apply equally to Waltz' extension of it. In particular, the twin problems of anomalous interpretations of legitimate scenes and acceptance of impossible objects demonstrated in Fig. 21 for the earlier procedure still remain. In fact, there is a further scene (Fig. 24) to which Waltz' program assigns the anomalous interpretation shown. But this anomaly cannot be avoided by the simple stratagem suggested to cope with the problems of Fig. 2.21 because the requirement that the common edges of intersecting surfaces appear collinear is satisfied here. What is required to reject this anomaly is a chain of reasoning involving hypotheses and deductions about surface and edge orientations. It is left to the reader to construct the argument.

The form of Waltz' input assumes the ability to see every edge perfectly including all those inside the shadow regions even though there is only a single light source (Fig. 2.23). Is this having your shadow cake and eating it too? Waltz does consider simple cases of missing edges but as he emphasized the labelling technique uses only the topology of the line drawing and local junction shape information. He gives many good examples of pictures equivalent on that basis that seem to

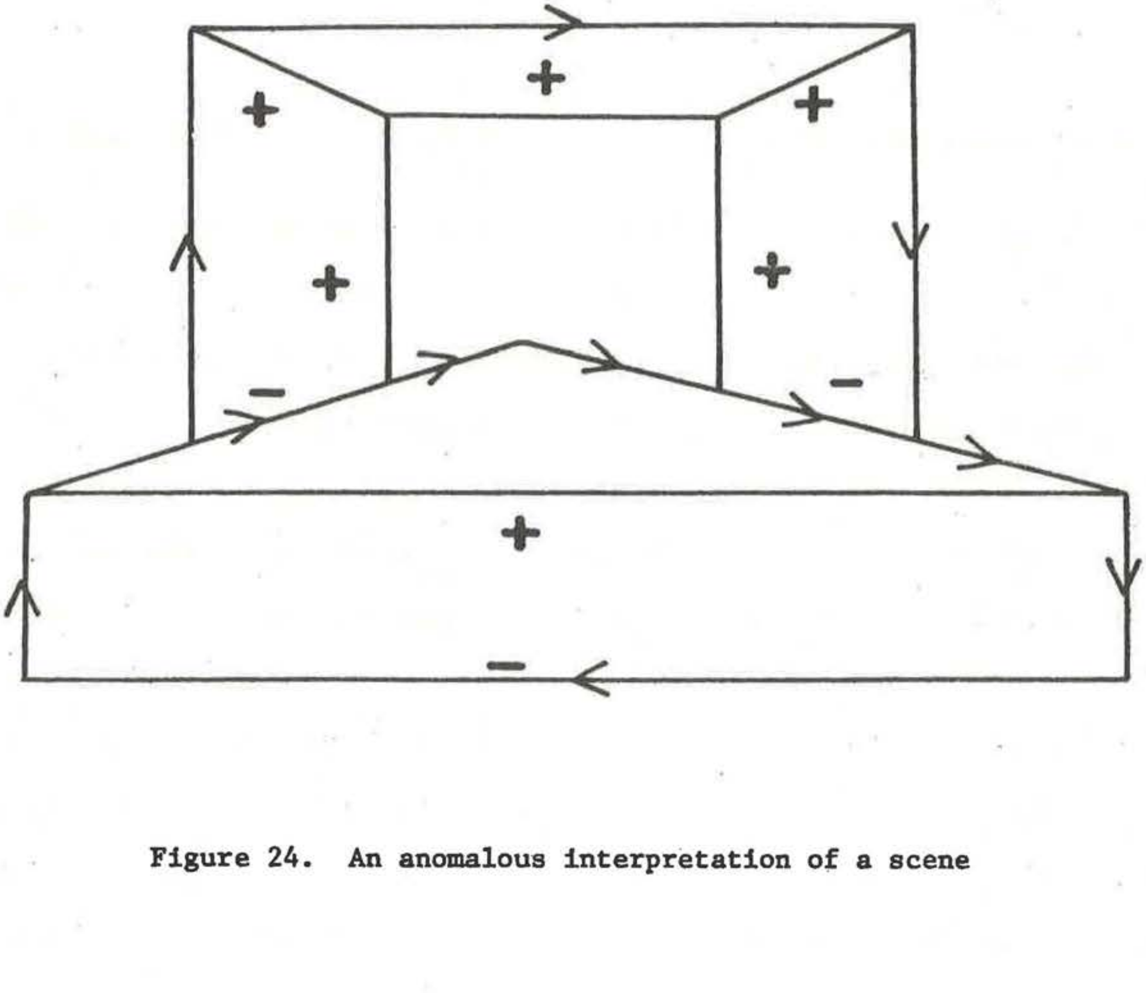

Figure 24. An anomalous interpretation of a scene

require very different interpretations or missing edge completions.

As we pointed out in the criticism of the Huffman-Clowes algorithm an interpretation procedure for line drawings must use more than the picture topology and agreement between adjacent corners if it is to be satisfactory in its treatment of all the various aspects of scene analysis discussed above.

8. POLY: exploiting surface coherence and the edge hierarchy

One approach that can only be briefly mentioned here is the author's program POLY (Mackworth, 1973, 1974a). Using a representation for surface orientations suggested by Huffman (1971), the gradient space, POLY hypothesizes and makes inferences about surface and edge orientations and positions exploiting heavily the hierarchical stucture of the network of interpretations of a line (see Fig. 22; the version of POLY implemented did not make the shadow or separable edge distinctions) thereby dispensing with the lists of possible corners. The only backtracking search in POLY is at the connect/occlude level of distinction in the edge hierarchy; the other features of the edges are then inferred directly from the surface, edge and corner representations used. While the size of the underlying search space has been drastically reduced, the resulting interpretation is richer in descriptive power including as it does relative information on surface and edge orientation and position. This descriptive adequacy or higher level of scene coherence not only makes the interpretation more useful but also ensures that the anomalies of Fig. 21(a), Fig. 21(b) and Fig. 24 do not arise.

9. Conclusion

In a paper on descriptive languages and problem solving Minsky (1968) sees artificial intelligence as an attempt to achieve adequate descriptions and procedures for manipulating them for specific task domains. This view provides the best framework for understanding the first decade of scene analysis. Starting with Roberts, there has been a continual struggle to achieve adequate picture and scene descriptions and procedures for relating the two with considerable progress being made. But, pace Chomsky, descriptive adequacy is not enough. The representation issue may be in a reasonably satisfactory state but the control issue is not. Of the work described here, only Roberts and Waltz have paid it sufficient attention. Of work not described here for space reasons, MIT's COPY DEMO (Winston,1973) and, more recently, Shirai's context-sensitive linefinder (Shirai,1973) are the most adequate from that viewpoint. Shirai's program, for example, uses a procedural model of the picture that is essentially a very loose characterization of all line drawings of scenes of convex polyhedra to direct the image analysis which consists of line and junction detection in grey-scale pictures. If we dare risk a linguistic analogy, that appears to be a syntactic model while we have an entire spectrum of semantic models ranging from Falk's size-specific polyhedral prototypes through Robert's transformable prototypes, the architectural models of Winston's thesis (Winston,1970), the Guzman-Huffman-Clowes-Waltz corner models, the hierarchy of line interpretations, to size or shape-specific surface models (Mackworth,1974b).

If we choose the active model of perception suggested to us by Roberts' program, how are we to cope with this abundance of models? How do they sensibly interrelate? How should they be invoked? When should they be invoked? And yet cope we must, for surely the availability of a wide variety of effective schemata conjoined with the ability to invoke the relevant subset of them at the appropriate time is the hallmark of intelligence.

10. Acknowledgements

The author is indebted to Max Clowes for inspiration and sound criticism. Robin Stanton, Stuart Sutherland and Aaron Sloman also helped shape these views on vision. This work was supported by the National Research Council of Canada and the Science Research Council of Great Britain.

11. References

Anderson, R. H. (1968) Syntax-directed recognition of hand-printed two-dimensional mathematics, Ph.D. thesis, Division of Engineering and Applied Physics, Harvard University.

Bartlett, F. C. (1967) *Remembering*, Cambridge University Press.

Chomsky, N. (1957) *Syntactic Structures*, Mouton and Co., The Hague.

Chomsky, N. (1965) *Aspects of the Theory of Syntax*, M.I.T. Press, Cambridge, Mass.

Clowes, M. B. (1969) Transformational grammar and the organization of pictures. *Automatic Interpretation and Classification of Images*, Grasselli, A. (Ed.), Academic Press, N.Y., pp. 43-77.

Clowes, M. B. (1971) On seeing things. *Artificial Intelligence*, *2*, 1, 79-112.

Clowes, M. B. (1972) Scene analysis and picture grammars. *Graphic Languages*, Nake, F. and Rosenfeld, A. (Eds.), North-Holland, Amsterdam, pp. 70-82.

Evans, T. G. (1969) Descriptive pattern analysis techniques. *Automatic Interpretation and Classification of Images*, Grasselli, A. (Ed.), Academic Press, N.Y., pp. 79-96.

Falk, G. (1972) Interpretation of line data as a three-dimensional scene. *Artificial Intelligence*, *3*, 2, 101-144.

Gregory, R. L. (1974) *Concepts and Mechanisms of Perception*, C. Scribner's and Sons, N.Y.

Guzman, A. (1968) Decomposition of a visual scene into three-dimensional bodies. *AFIPS Proc. Fall Joint Comp. Conf.*, *33* , 291-304.

Huffman, D. A. (1971) Impossible objects as nonsense sentences. *Machine Intelligence 6*, Meltzer, B. and Michie, D. (Eds.), Edin. Univ. Press, Edinburgh, pp. 295-323.

Kirsch, R. A. (1964) Computer interpretation of English text and picture patterns. IEEE Trans. On Electronic Computers, EC13, 363-376.

Ledley, R. S. (1964) High speed automatic analysis of biomedical pictures. Science, 146, 216-223.

Mackworth, A. K. (1973) Interpreting pictures of polyhedral scenes. Artificial Intelligence, 4, 2, 121-137.

Mackworth, A. K. (1974a) On the interpretation of drawings as three-dimensional scenes. D.Phil. Thesis, Lab. Of Exp. Psych., Univ. Of Sussex.

Mackworth, A. K. (1974b) Using models to see. Proc. AISB Summer Conf., pp. 127-137.

Mackworth, A. K. (1975) Consistency in networks of relations. Tech. Rep. 75-3, Dept. Of Computer Science, Univ. Of British Columbia.

Michie, D. M. (1974) On not seeing things. in On Machine Intelligence, Wiley.

Minsky, M. L. (1961) Steps toward artificial intelligence. Proc. Inst. Radio Engineers, 49, 8-30.

Minsky, M. L. (1968) Descriptive languages and problem-solving. Semantic Information Processing, Minsky, M. L. (Ed.), M.I.T. Press, Cambridge, Mass., pp. 419-424.

Minsky, M. L. and Papert, S. (1972) Progress Report, A.I. Memo. 252, Massachusetts Inst. Of Technol., Cambridge, Mass.

Minsky, M. L. (1975) A framework for representing knowledge. In (Winston, 1975).

Narasimhan, R. (1966) Syntax-directed interpretation of a class of pictures. CACM, 9, 3, 163-173.

O'Gorman, F. and Clowes M. B. (1973) Finding picture edges through collinearity of feature points. Proc. Third Int. Joint Conf. On Artificial Intelligence, Stanford Research Inst., Stanford, Calif., pp. 543-555.

Penrose, L. S. And Penrose, R. (1958) Impossible objects: a special type of illusion. Brit. J. Psych., 49, 31-33.

Roberts, L. G. (1965) Machine perception of three-dimensional objects. Optical and Electro-optical Information Processing, Tippett et al. (Eds.), M.I.T. Press, Cambridge, Mass., pp. 159-197.

Shaw, A. C. (1969) A formal picture description scheme as a basis for picture processing systems. Information and Control, 14, 1, 9-52.

Shirai, Y. (1973) A context-sensitive line finder for recognition of polyhedra. Artificial Intelligence, 4, 2, 95-199, also in (Winston, 1975).

Southall, J. P. C. (1962) (Ed.) Helmholtz' Treatise on Physiological Optics, Vol. III, Dover.

Stanton, R. B. (1970) Computer graphics - the recovery of descriptions in graphical communication. Ph.D. Thesis, Dept. Of Electronic Computation, Univ. of New South Wales.

Stanton, R. B. (1972) The interpretation of graphics and graphic languages. Graphic Languages, Nake, F. and Rosenfeld, A. (Eds.), North-Holland, Amsterdam, pp. 144-159.

Waltz, D. L. (1972) Generating semantic descriptions from drawings of scenes with shadows. MAC AI-TR-271, Massachusetts Inst. of Technol., Cambridge, Mass., also in (Winston, 1975).

Winograd, T. (1971) Procedures as representation for data in a computer program for understanding natural language. MAC AI-TR-84, Massachusetts Inst. of Technol., Cambridge, Mass.

Winston, P. H. (1968) Holes. A. I. Memo. 163, Massachusetts Inst. of Technol., Cambridge, Mass.

Winston, P. H. (1970) Learning structural descriptions from examples. MAC AI-TR-76, Massachusetts Inst. of Technol., Cambridge, Mass., also in Winston (1975).

Winston, P. H. (1973) The MIT robot. Machine Intelligence 7, Meltzer, B. and Michie, D. (Eds.), Edin. Univ. Press, Edinburgh, pp. 431-463.

Winston, P. H. (1975) The Psychology of Computer Vision, McGraw-Hill.