Perception, 1976, volume 5, pages 349-370

Model-driven interpretation in intelligent vision systems

Alan K Mackworth
Department of Computer Science, University of British Columbia, Vancouver, BC, Canada
Received 7 August 1975, in revised form 22 March 1976

Abstract. With a constructive knowledge-based theory of perception as its foundation, this paper starts with a review and critique of some artificial-intelligence programs that purport to see. It is then argued that these computer programs for scene analysis offer the hope of providing a more adequate account of human competence in interpreting line drawings as polyhedra than do the current psychological theories. This thesis has several aspects. The one emphasized here is that those programs have explored a variety of methods of incorporating *a priori* knowledge of objects through the use of models. After outlining the range of models used, presenting a set of criteria for evaluating the use of model information, and sketching some psychological theories, the various proposals are contrasted. This discussion leads to two new proposals for exploiting model information that involve elaborations of an existing program, POLY.

1 Introduction

In one of its many roles artificial intelligence is cast as the vanguard of an army of psychologists who seek a new paradigm for cognitive and perceptual processes. Despite several clarion calls to this effect (Minsky and Papert 1972; Clowes 1972b; Sutherland 1973), artificial intelligence may well be a vanguard without an army. This paper attempts to show that a small part of the scouted territory is ripe for capture.

The interpretation of line drawings as polyhedral scenes has been the focus of most attempts to build artificial-intelligence vision systems. Since such interpretation is a natural human task, several psychologists have also studied it. In sketching and contrasting various resultant theories, we will concentrate on how they represent the *a priori* knowledge of the objects that exist in the world. Of necessity other essential themes such as nonmodel knowledge of the world (for example, knowledge of support and the picture-formation process itself) or the use of picture cues to access the models are slighted.

Section 2 gives the necessary background in scene analysis to the reader unfamiliar with the field. Section 3 then examines the use of models in those programs. Section 4 sketches the use of models in some theories of human vision and in section 5 a few examples are used to contrast and evaluate various proposals. This leads, via the weaknesses exposed, to the two new approaches in section 6.

2 Machine vision

Insistence on the need for descriptive adequacy of the internal representations of the perceived world and for procedural adequacy of the interpretation process itself, contrasted with the lack of both in simple classification mechanisms, is what distinguishes scene analysis from earlier work in pattern recognition. The assumption that the world consists of objects with flat surfaces is a simplification of reality for the sake of tractability that has led to 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, such as understanding the picture-taking process, models, lighting, support, and occlusion. The thesis is that once we achieve ways of dealing intelligently with those aspects for a simple, but nevertheless 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 answer this question.

2.1 *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 three-dimensional 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.

Roberts' program believes that the world consists of the models shown in figure 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. A typical compound object and its components are shown in figure 2. The models are specified by three-dimensional 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 and the subsequent removal of hidden lines. The relationships of the model, object, and picture domains are as shown in figure 3 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

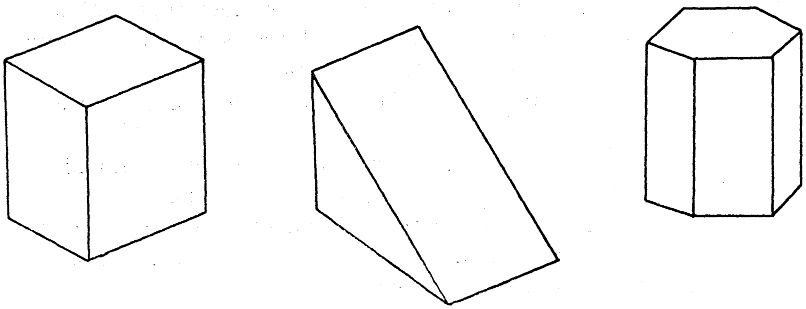

Figure 1. Roberts' (1965) simple object models.

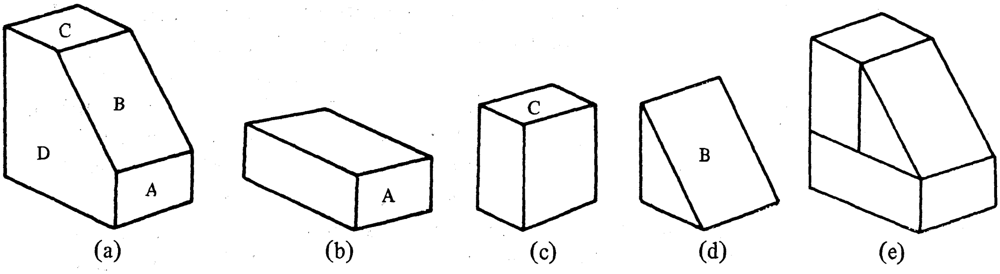

Figure 2. A typical Roberts compound model.

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 glossed over is the decision to choose a set of picture lines to interpret first. 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. When a candidate model is found, 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, a least-squares fit gives H with the residual error as a measure of the picture–model mismatch. If the mismatch is too large, that model is rejected and the topology search continues. If the transformed model is acceptable, it 'explains' a fragment of the picture which is then deleted and the entire process starts again.

Roberts' program created a scene-analysis paradigm that remains dominant. As a working theory, for that is what an artificial-intelligence 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 (1972a) remarked, the approach of such psychologists as Helmholtz (Southall 1962), Bartlett (1967), and Gregory (1974). Minsky's recent frame systems (Minsky 1975) provide a semiformalism for this paradigm of perception.

$$\begin{tikzcd}
\text{Model} \arrow[r, "R"] \arrow[rr, bend right, "H"'] & \text{Object} \arrow[r, "P"] & \text{Picture}
\end{tikzcd}$$

Figure 3. Roberts' domains and transformations.

2.2 *Guzman's body-segmentation program, SEE*

Guzman's (1968) SEE 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 figure 4 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, or part of the crossbar of a T.

The result of the first pass is a graph with regions as nodes and links as arcs and the second pass searches for two-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 counterexamples 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.

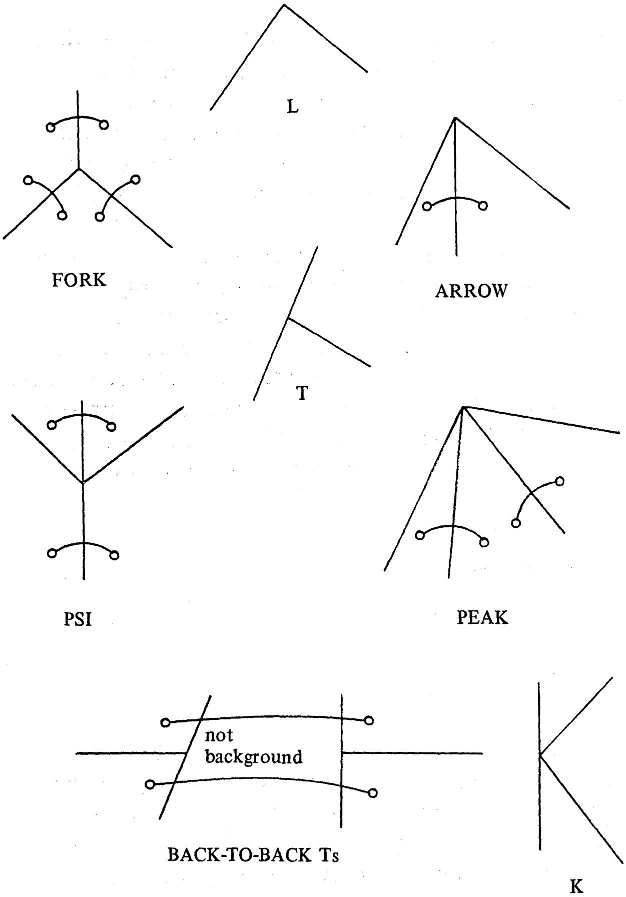

Figure 4. The link-planting rules of SEE (Guzman 1968).

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, ambiguities that are exposed by the Huffman-Clowes labelling algorithm. For example, consider figure 5a (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 figure 5a), as we shall see. The labelling algorithm also detects situations, illustrated by the picture in figure 5.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 simple polyhedra.

An interesting comparison can be made between Roberts' program and SEE. A one-to-one correspondence exists between the Roberts topology tests and Guzman's junction linking rules (see Mackworth 1975b). Moreover, Roberts used knowledge of models explicitly in the body-segmentation task. He did this in three ways, first, by looking for a feature (convex regions with three, four, or six sides) common to all the models, second, by using model-specific topology tests to identify a picture fragment as part of a particular model, and, third, having made an identification, projecting the rest of the model onto the picture to account for many more lines. Guzman, on the other hand, claims to use no knowledge of models in the segmentation. This claim may indeed be doubted on the grounds of the Roberts-Guzman parallel presented in Mackworth (1975b). SEE seems to prefer convex regions as body faces. This is confirmed in the analysis of SEE's underpinnings in section 2.4. 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).

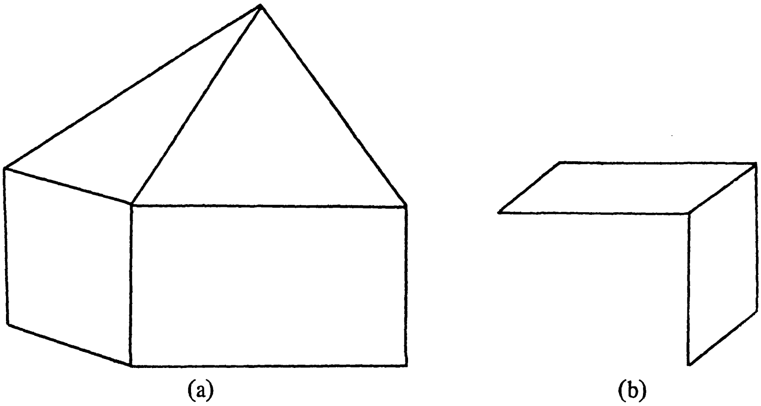

Figure 5. Illustrating (a) ambiguity and (b) anomaly for SEE.

2.3 *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 set of nine fixed-size prototypes that appear in the world (figure 6) 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 missing) to produce an exact three-dimensional representation of the scene.

The system consists of the five stages of figure 7. 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 uses 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

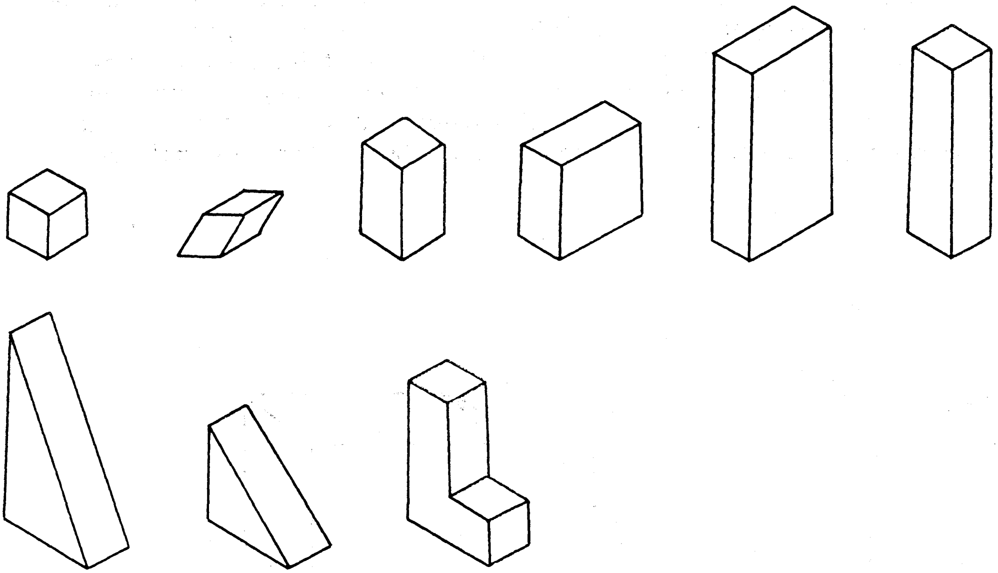

Figure 6. The object prototypes in INTERPRET (Falk 1972).

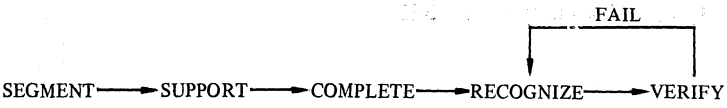

Figure 7. The organization of INTERPRET (Falk 1972).

flanking the shaft is background or if the shaft is the top of a *K* junction, otherwise it is a GOODARROW. 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.

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 three-dimensional position of the endpoints of all the base edges.

The picture of each object may be incomplete for three possible reasons: (i) the original picture had some lines missing, or (ii) the object is partially occluded, or (iii) SEGMENT failed to assign some lines to the body. COMPLETE has three routines that attempt to patch up each object before recognition.

INTERPRET does not recognize an object until all of 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 endpoints of all the base edges of the object can then be located in three-dimensional 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.

When the object has been named and three corners of the base edges of it have been located in space, 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 three 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, 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. It is easy to find examples that mislead such heuristics (see Mackworth 1975b), quite simply because in the picture *qua* picture there is no basis for deciding which of two possible segmentation, completion, or support decisions makes more sense.

"makes more sense" is a remark that applies not to the picture itself but to what is depicted, the scene. It is clear that the program must have some kind of three-dimensional interpretation before it can evaluate predicates such as 'same body', 'supports', and 'missing edge'; however, the only way Falk has of getting a three-dimensional interpretation is by recognizing the objects. This is a chicken-and-egg problem: the program needs to recognize the objects to get a three-dimensional 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.

2.4 *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 two-line and three-line junctions and three-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, applied 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 belonging 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. See figure 8 for an illustration of these labels.

Figure 9 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.

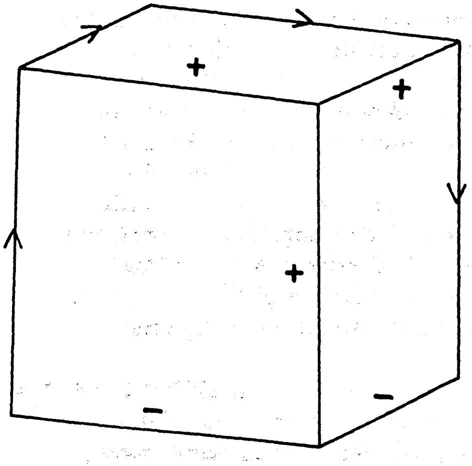

Figure 8. A labelled interpretation of a picture.

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, 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, figure 10 (from Clowes 1971), if we ourselves cannot assign a plausible three-dimensional interpretation. Moreover, this ability would be of practical use in a scene-analysis program. Figure 5b, 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

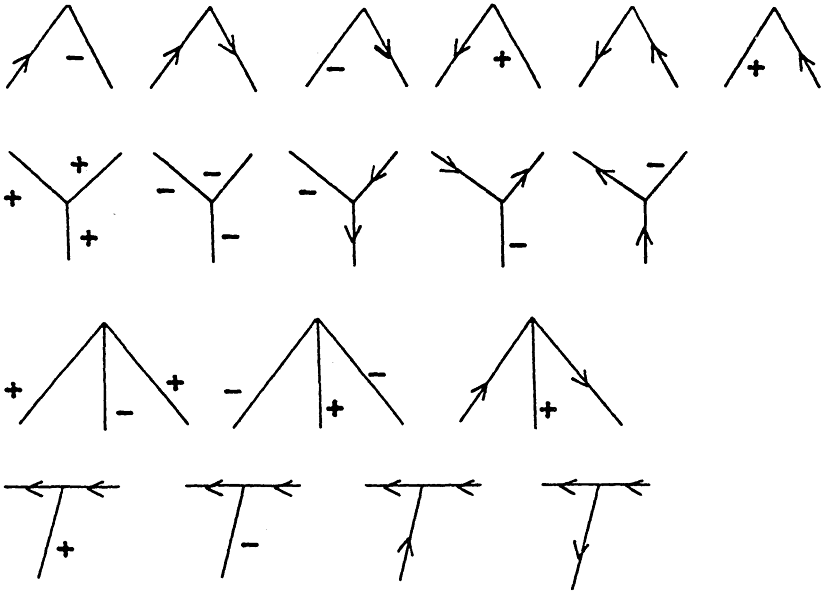

Figure 9. The Huffman-Clowes junction interpretations.

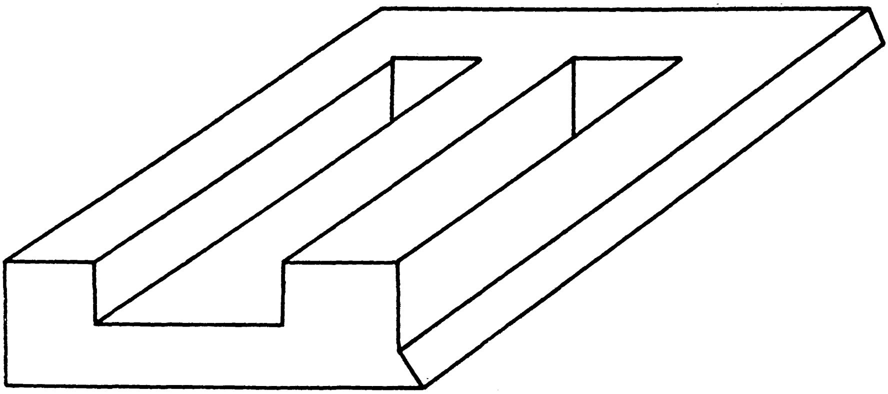

Figure 10. An impossible object (Clowes 1971).

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 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 (figure 9) eliminate all corners with concave edges, including those for the L that imply a hidden concave edge, leaving the set of figure 11. 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 with this corner set a body partitioning is easy to achieve. That is what Guzman did! The links were planted at unambiguously convex edges. The link-planting rules of figure 4 are derived from the corner interpretations of figure 11 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 figure 11 to suppress a link across an edge if its other end shows it to be unambiguously occluding. The accumulation of link evidence relies on two 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 does not work on concave objects (Winston 1968).

It is shown in Mackworth (1975b) that the Huffman–Clowes labelling algorithm does away with some but not all of the difficulties in Falk's program; however, there are some problems with the labelling algorithm as described here. It can make mistakes. In figure 12a it incorrectly labels a legitimate view of a cube (it will of course produce all the correct labellings as well) and in figure 12b (adapted from Huffman 1971) it labels an object that cannot be a polyhedron with planar surfaces. Both 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. However, that *ad hoc* extension evades the key issue, which is that the algorithm has no requirement that surfaces be planar,

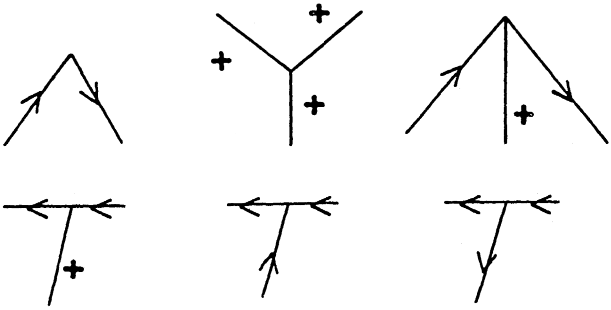

Figure 11. The junction interpretations for convex polyhedra.

nor is there any way that the requirement can be 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 in fact a partial answer to those problems.

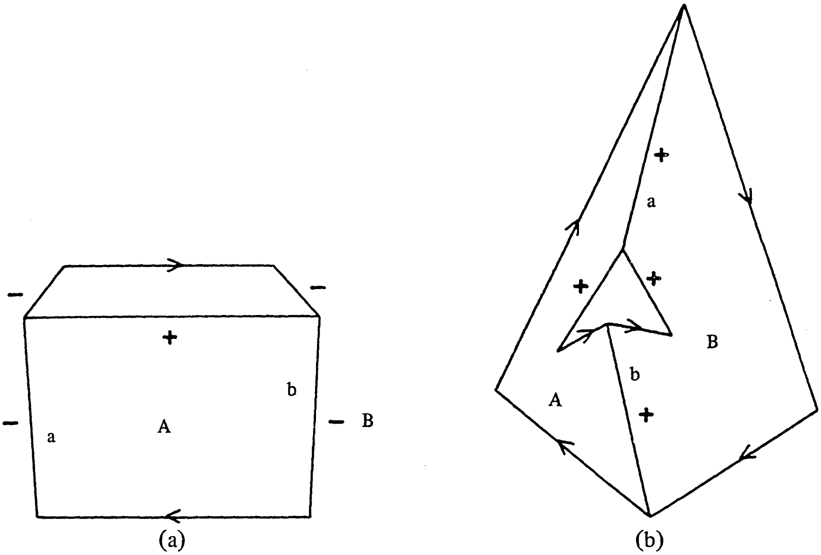

Figure 12. Labelling problems: (a) an anomalous interpretation of an object; (b) an interpretation of an anomalous object (adapted from Huffman 1971).

2.5 *Waltz' extension of the labelling algorithm*

Waltz (1972) made two important contributions to the labelling algorithm. He expanded the set of line labels from the four used by Huffman and by 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 nonseparable. 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 showing the status of the edges of the separate bodies.

The other expansion of edge possibilities derives from a crude account of lighting. If a single concentrated light is assumed, source 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 four to fifty-three.

Following a graphical representation used by Winograd (1971) 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 figure 13. In that network the choice of illumination status for each surface has not been shown, so there are only eleven distinct line interpretations (paths through the network).

For the possible corners and their picture appearance, Waltz allowed the Huffman–Clowes junction categories and also all four-line and some five-line junctions. Following a straightforward procedure, he 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 ten 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 ruled 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 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, 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, therefore 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 (1975a).

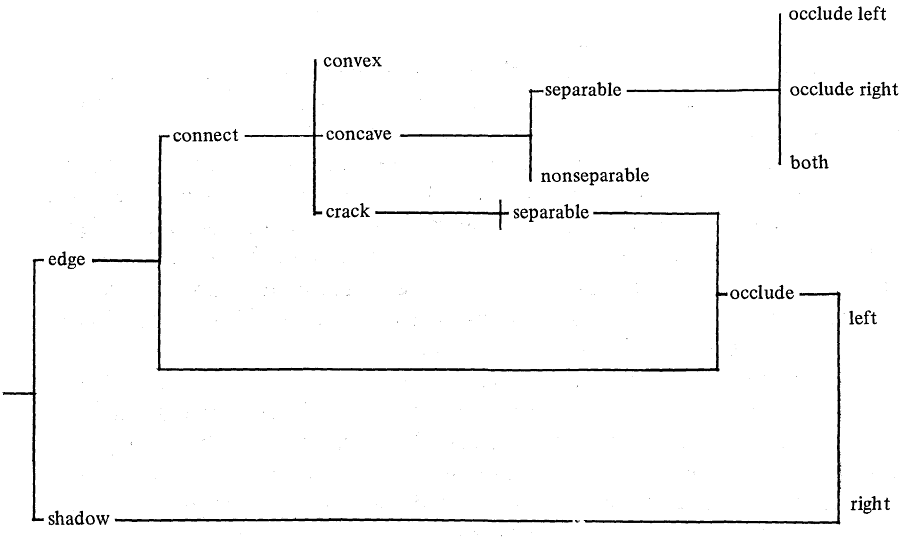

Figure 13. A network of the interpretations of a line.

Figure 14 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.

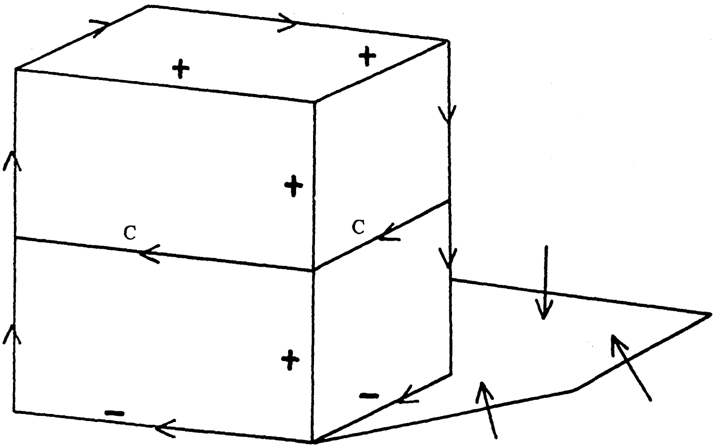

Figure 14. A scene labelled by Waltz' (1972) program.

2.6 *POLY: exploiting surface coherence and the edge hierarchy*
An approach to scene analysis that can only be summarized here is the author's program POLY (Mackworth 1973, 1974a). It uses a representation for surface orientations suggested by Huffman (1971), the gradient space. We do not have the space to reexamine the design of POLY, but, since an understanding of the underlying representation will be required later in this paper, its crucial features will be presented here.

Given a picture that is an orthographic projection of a three-dimensional line, figure 15a, then the tilt of that line from the picture plane is unknown. The tilt can be represented by a vector called the gradient, whose direction is in the direction of the picture line and whose length is the tangent of the angle the scene line makes with the picture plane. The sense of direction of the vector which is shown in

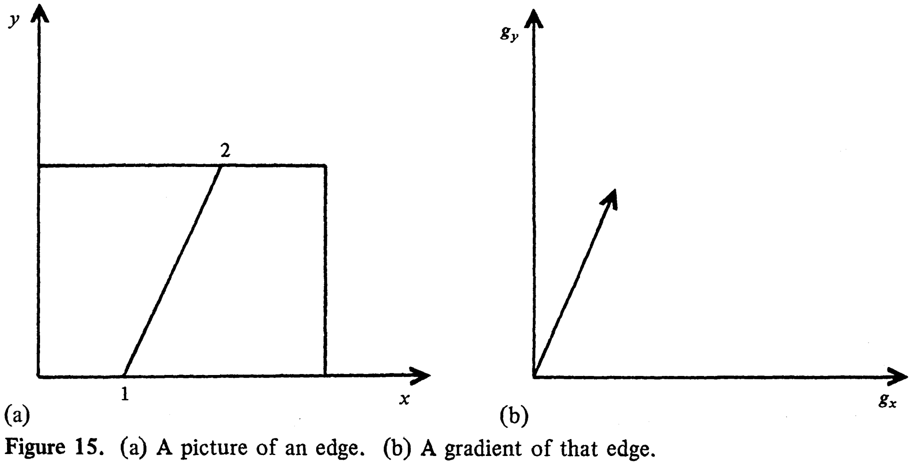

Figure 15. (a) A picture of an edge. (b) A gradient of that edge.

figure 15b is such that end two of the scene line is farther from the picture plane than end one. In general, the gradients of all scene lines that could correspond to that picture line are scalar multiples of the vector in figure 15b. In particular a scene line parallel to the picture plane has the zero gradient vector.

Consider the gradients of all the lines lying in a surface tipped away from the picture plane, figure 16a. Suppose that line a in the surface is parallel to the picture plane so that its gradient is zero. If that line is rotated in the surface, the gradient increases until the line arrives at position b where it is perpendicular (both in the picture and in the scene) to its previous position, and the gradient has its maximum value. Thereafter, the gradient decreases again. It is easy to show by simple trigonometry that the locus of the gradient of the lines in the surface is a circle passing through the origin as shown in figure 16b. By convention we call the diameter passing through the origin the gradient of the surface. Thus, the gradient

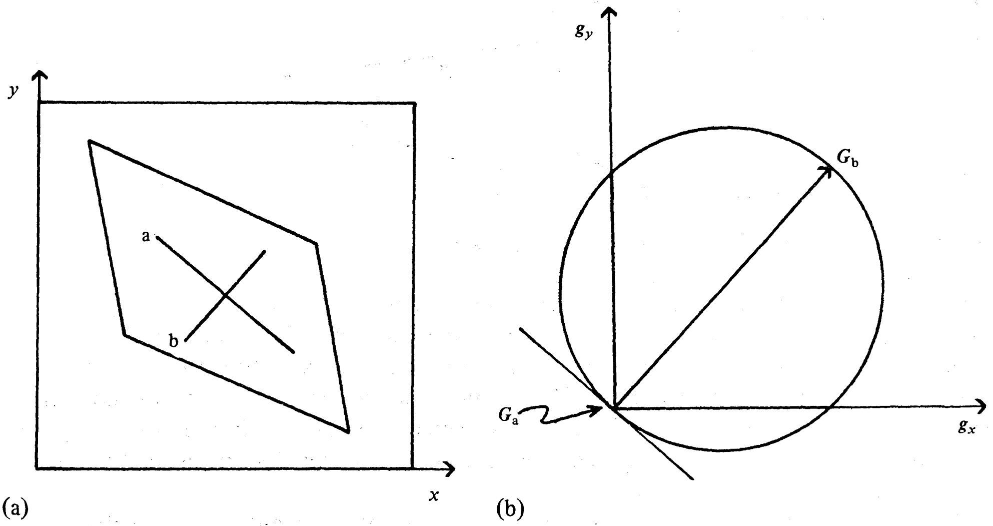

Figure 16. (a) A picture of a surface containing two lines. (b) The gradients of the lines in the surface.

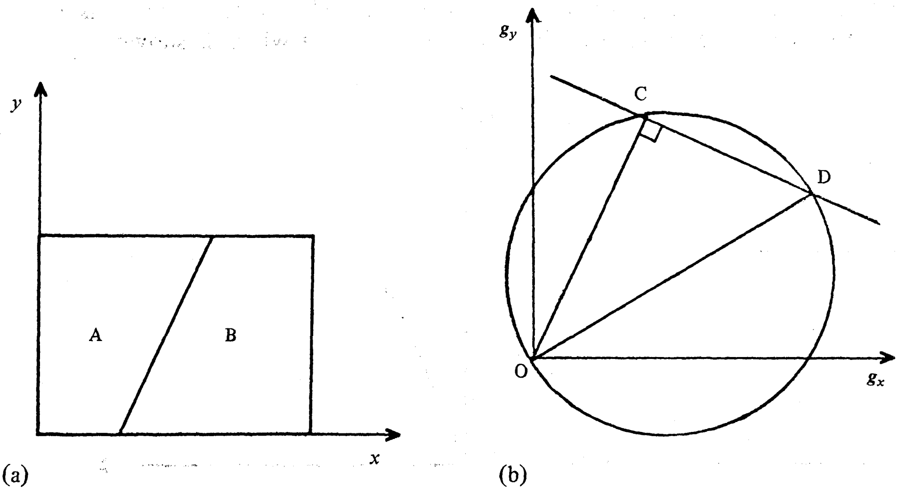

Figure 17. (a) Two surfaces, A and B, meeting in an edge. (b) The relationship between the gradients of the surfaces.

of the surface is the gradient of the set of parallel lines in the surface having the steepest inclination to the picture plane (the 'fall line' to skiers). Three points on the circumference define a circle therefore, since these circles all pass through the origin, the gradients of two lines in the surface fix the gradient of that surface.

Suppose the gradient of only one line in the surface is given, as in figure 17a; how does that constrain the gradient of the surface? If the gradient of the line is given as OC in figure 17b, then the gradient of the surface could be at D. Angle DCO is the angle subtended at the circumference by a diameter and thus it is a right angle. The result is that the gradient of any surface containing that line lies on a line perpendicular to the known line gradient and passing through its endpoint. If, on the other hand, it is only given that two surfaces such as A and B in figure 17a have the common edge shown (without being given the tilt of that edge or, correspondingly, the length of OC), then, since the edge thereby lies in both surfaces, one can infer that the line joining the gradients of those two surfaces is perpendicular to the picture line depicting the common edge.

By sacrificing grammar to brevity, an edge that connects two visible surfaces may be called a connect edge. Again, consider figure 17a where a connect edge common to surfaces A and B is shown. If the gradients of the edge and surfaces A and B are G_e, G_A and G_B, respectively, then figure 18 shows the six ways in which they can be arranged in relation to each other in the gradient space (we temporarily ignore the special cases where two or more gradients are coincident). Considering these cases individually shows that for cases (a)–(c) the edge must be convex, while for (d)–(f) the edge must be concave. This distinction is concisely captured by ignoring the position of G_e and noting for cases (a)–(c), G_A is on the one side of G_B while in cases (d)–(f), the positions of G_A and G_B are reversed on the line. If the ordering of the gradients is compared with the relationship of the regions in the picture, a rule emerges. If the gradients are in the same relative position as the corresponding surfaces appear at the common connect edge, then that edge is convex; but, if the relative position of the gradients is reversed, the edge is concave. If the gradients coincide, the edge is flat; that is, it is a crack. These crucial facts allow the exploitation of the gradient space for convex/concave/crack interpretations.

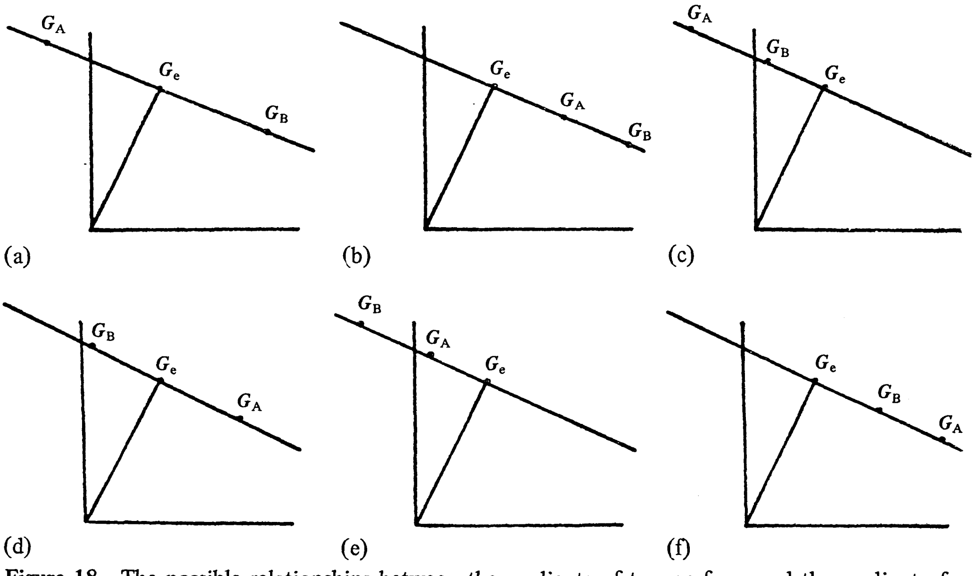

Figure 18. The possible relationships between the gradients of two surfaces and the gradient of their common edge.

But, clearly, confining its exploitation to that would represent a degradation of the descriptive power available, since the degree of convexity/concavity is also specified for any particular choice of the origin and scale of the gradient space. However, if one works only from the picture (that is, does not introduce extrapictorial information such as, for example, knowing the orientation of the support plane, or requiring some form of model to make sense of the picture), the origin and scale of the gradient space will be unspecified.

As an illustration of the use of the rules that gradient configurations must satisfy, consider a FORK junction, figure 19 (familiar from the earlier labelling procedures), for which it is known that the three edges are connect. The configuration of the gradients of surfaces A, B, and C (G_A, G_B, and G_C) can take on only one of the two forms of figure 20 if the gradients are to satisfy the requirement that the mutual vector difference of a pair of gradients be perpendicular to the line depicting the edge that connects the two surfaces. These configurations can be translated and expanded in the gradient space and still satisfy the requirement. Comparing the relative positions of the gradients in figure 20a with the ordering of the regions in the picture shows that all the edges must be convex for that interpretation, while for the interpretation given in figure 20b all the edges must be concave. That switch of interpretations, which can be achieved by mapping every gradient G into its negation $-G$, is simply the Necker reversal.

Using those aspects of the gradient space, POLY hypothesizes and makes inferences about surface and edge orientations and positions exploiting heavily the hierarchical structure of the network of interpretations of a line (see figure 13; 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

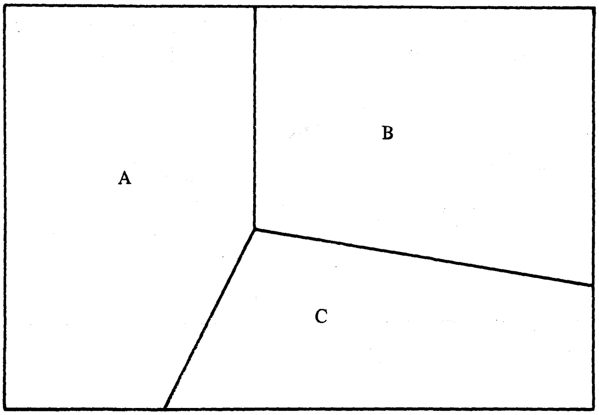

Figure 19. A FORK junction.

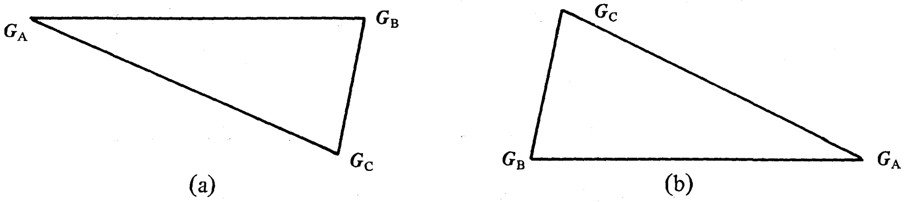

Figure 20. The possible gradient interpretations of a connected FORK.

drastically reduced, the resulting interpretation is richer in descriptive power because it includes 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 various anomalies such as those of figure 12 do not arise.

3 Models in machine vision

The noun 'model' is taken to mean a representation of a fragment of the world that a vision system uses to understand or model the reality it perceives. For our purposes we can identify four aspects of models and their use that serve as dimensions for comparison: (i) plasticity, (ii) scope, (iii) degree of embedding, (iv) variety.

By *plasticity* is meant the degree to which the model can be changed to fit the world. Both size and shape plasticity are considered. *Scope* indicates the physical extent of the world that an individual model embraces: a corner, an edge, a surface, an object, or an entire room. An *embedded* set of models form a hierarchy in which one model may be a part of or a specialization of another. None of the programs discussed in section 2 have any substantial degree of embedding. *Variety* is the variation in the world that can be explained by the set of models. Thus, the variety of each program is limited by the polyhedral assumption but some are further restricted.

Falk's nine objects have no size or shape plasticity, greater scope than any other program considered in section 2 but less scope than Winston's architectural models (Winston 1970), and the least variety. For these reasons they could be called object prototypes. Roberts' three simple models have indefinite size plasticity and some shape plasticity; their scope is an entire object or a part of an object. The programs of Guzman, Huffman, Clowes, and Waltz all have as scope an object's corner. As we have seen, they range in variety from Guzman's single trihedral convex corner through the four corners of the Huffman–Clowes algorithm (the trihedral corners in which the object occupies one, three, five, or seven 'octants') to the large number allowed by Waltz. The corners in all cases have no further shape specificity than that provided by specifying the number of surfaces and the convexity/concavity of the edges in which they meet.

The model information in POLY is minimal but varied, and includes requirements that surfaces be planar and edges either be occluding or connect, with a marked preference for connect edges. A certain amount of embedding is provided by exploiting a subset of the edge hierarchy (figure 13). It is one of the purposes of this paper to show how POLY can be extended to augment the plasticity, scope, degree of embedding, and variety of available polyhedral models.

4 Some psychological theories

Attempts to provide psychological theories of the interpretation of line drawings have not usually provided an algorithm by which interpretation may proceed; though presumably some of the usual monocular depth cues are thought to be relevant. Rather, such theories seem to assume the existence of such an algorithm and concentrate on the tension set up between the three-dimensional (scene) and two-dimensional (picture) organizations. Kopferman (1930) held that the impression of tridimensionality varies with the degree to which the scene organization is simpler than that of the picture. In extending that theory, Hochberg and Brooks (1960) provided experimental evidence for a quantified measure of pictorial complexity as the sum of (1) twice the number of line segments, ignoring intersections, (2) the number of interior angles, and (3) the number of different angles divided by the number of angles. Thus, for example, this measure correctly predicts that subjects will rank the scenes of figure 21 in the following order of increasing perceived tridimensionality: a, c, b, d. Attneave and Frost (1969) presented a similar theory in which the competition between the scene and the picture is resolved by figural simplicity criteria.

Finally, Hochberg (1968) almost anticipated the Huffman–Clowes algorithm as he demonstrated, with an ingenious experiment, that junctions act as 'local depth cues'. That is, the shape of a junction implies depth relations between the surfaces and edges at the corresponding corner, but such effects are exerted over only a limited distance.

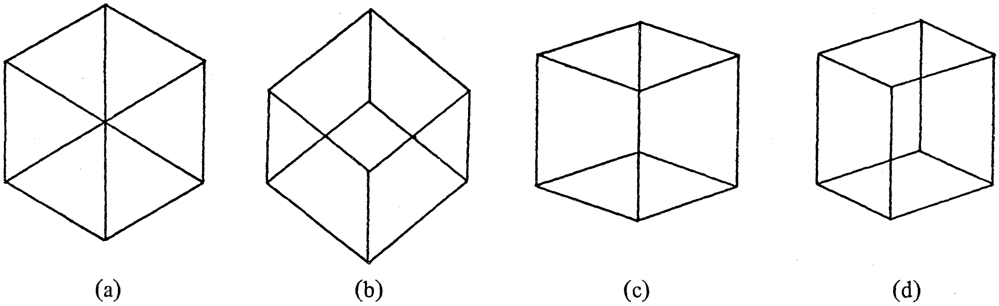

Figure 21. Illustrating variation in perceived tridimensionality.

5 Some examples

The discussion of this section uses as examples the two pictures in figure 22, which the reader should look at without reading further. The usual impression given by figure 22a is that it remains obstinately flat on the page (provided one can suppress the tendency to see an edge that is not depicted and to ignore one that is, thereby transforming the picture into figure 22b). Figure 22b has a solid three-dimensional appearance. This major difference, which holds even though both are equally faithful depictions of polyhedra, is a phenomenal fact requiring explanation.

The Huffman–Clowes algorithms do not offer that explanation. The pictures are successfully labelled with equal ease. The corners are all trihedral and both interpretations require three hidden surfaces to complete the object.

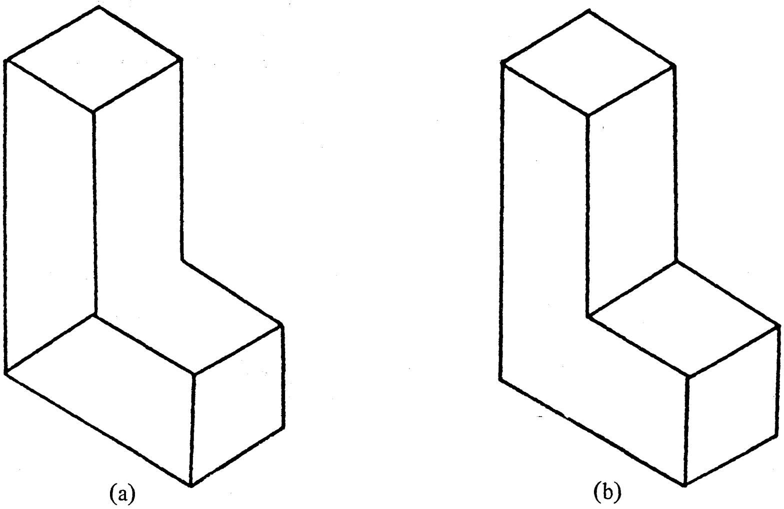

Figure 22. Two examples.

Perhaps the scene interpretation of figure 22b derives from its familiarity, as Falk's program suggests: the L-beam is one of its nine prototype objects. Look then at figure 24a, which is surely unfamiliar to the reader [although it is derived from pictures used by Shepard and Metzler (1971)]; is that object any less solid than the one depicted in figure 22b?

Both figure 22b and figure 24a are interpreted by Roberts' program as compound objects made from cuboids (two and four, respectively). However, that program cannot, contrary to expectation, similarly interpret the two objects in figure 23.

The traditional depth-cue theory of picture interpretation has nothing to say. In four of the pictures (figures 22a, 22b, 23b, 24a) there are no traditional depth cues at all! (The traditional monocular depth cues include apparent size, partial occlusion, varieties of shading and perspective, and some colour effects.) Yet, in figure 24a for example, corner 1 is clearly nearer the observer than corner 2.

The Hochberg–Brooks criterion does not contradict the phenomenon. By that criterion the pictures have equal complexity, while the scene in figure 22b is just marginally simpler than that in figure 22a. And yet that does not take us very far. What mechanisms produced those scene interpretations in the first place? Hochberg (1968) has provided an excellent rebuttal of the earlier theory of Hochberg and Brooks by pointing out, for example, that it fails to make any sense of the various experimental phenomena that surround 'impossible objects'.

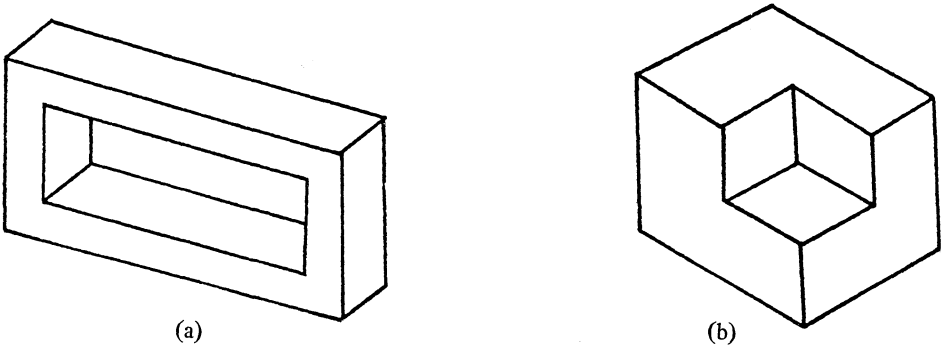

Figure 23. Two concave rectangular objects.

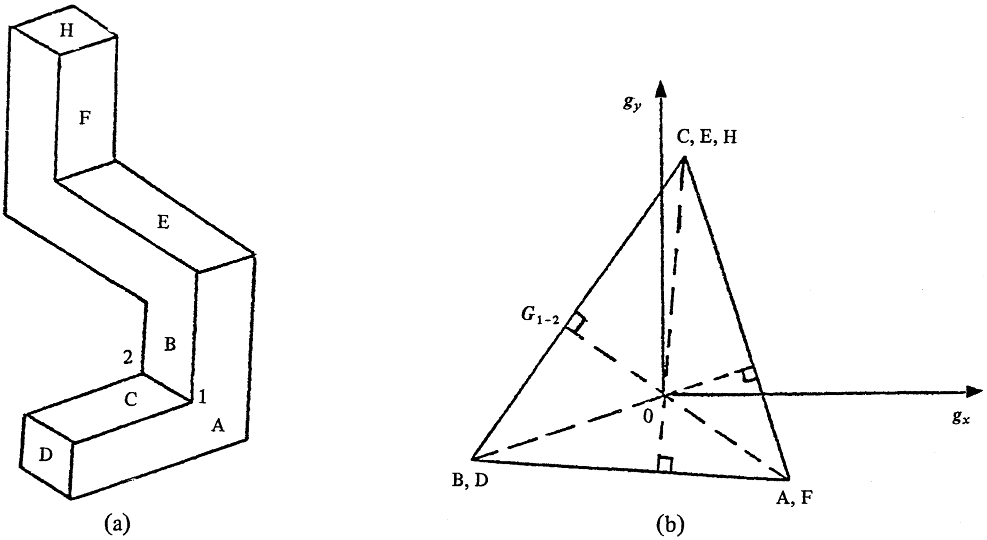

Figure 24. A rectangular object and its gradient space configuration.

On the other hand, Hochberg's claim (1968) that the junction configurations act as 'local depth cues' is not very powerful. In making that claim, Hocherg somewhat enlarged the usual meaning of the word 'depth', taking it to include such relationships as the convexity and concavity of edges in addition to its usual reference to the relative distance of two scene fragments from the viewer. The only real depth evidence given by Clowes–Huffman edge labelling is provided by the occluding edges; that is the traditional depth cue of partial occlusion. Even if the edges of figure 24a are appropriately labelled convex, concave, or occluding, there is still no evidence that corner 1 is closer to the observer than corner 2; that is, an ordinary polyhedron with that appearance and those edge labels can be constructed for which that is not true.

6 Two proposals

Surely the solidity of figure 22b and figure 24a, as contrasted with the 'flat' appearance of figure 22a, can be explained as follows: in the former cases the polyhedron interpretation can be seen as made up of surfaces of very familiar shape (in this case rectangular), whereas in the latter that is not possible. This explanation suggests extensions to POLY that use the shapes of surfaces as models. Here two such extensions are presented.

Rectangularity is often a major feature of the worlds we build for ourselves. The first proposal shows how a straightforward extension to POLY can exploit that feature. The second proposal is more of a substantial upheaval than an extension in that it suggests integrating the use of prototypes into the interpretation process.

6.1 *Rectangularity*

Consider the rectangular object of figure 24a. For this object POLY produces the gradient configuration that appears as a triangle in figure 24b. The object has three families of mutually parallel surfaces so there are only three possible values for the surface gradients. The gradients of each surface of the family are superimposed at each position. Every pair of surfaces meeting in a connect edge is joined by a line perpendicular to the picture line showing that edge. Neither the position of the origin nor the size of the gradient triangle is yet specified, but note that E and A are ordered in the gradient configuration just as they are ordered across their common edge in the picture. Therefore that edge is convex, whereas the relative positions of B and C are reversed in the picture and gradient spaces so that edge 1–2 is concave; however, as the actual values of the gradients are not determined, we still cannot say that corner 1 is closer than 2.

At any corner such as corner 1 in figure 24a there are three edges (which may not all be visible). Each pair of edges defines a surface at that corner. Each edge is normal to the surface defined by the other two. Since the direction of the gradient vector of a surface is the direction in the picture in which the surface normal appears to point, the direction of the gradient of each surface at the corner is given by the direction of the edge that does not belong to it. Thus gradient A must be in the direction of picture line 1–2. Since the vector difference between gradients B and C is required to be perpendicular to picture line 1–2, the origin must be on a perpendicular dropped from gradient A to the opposite side of the gradient triangle. Hence the origin must be at 0, as shown in figure 24b. The scale is immediately determined by the rectangularity requirement that the product of the magnitudes of the gradient of A and the gradient of edge 1–2, G_{1-2}, must be unity. Now that the orientations of all the surfaces and edges are defined, it is an obvious consequence that corner 2 is further from the picture plane than is corner 1; that is shown by the fact that G_{1-2} points up to the left (not down to the right).

6.2 *Using prototype surfaces*

The idea of using specific prototypes is attractive but, as suggested in section 5, complete polyhedral prototypes are in a sense too monolithic; we need hierarchies of embedded models of varying scope. In this section we show how the use of surface and object models can be integrated directly into the POLY interpretation process.

Consider Falk's list of nine prototype objects. They have in all fifty-four separate faces; yet those faces have only fourteen distinct polygonal shapes. The size specificity of these shapes will be dropped for the sake of this argument although it could be retained. Dropping size specificity (so that a 1 x 2 rectangle represents itself and also the 2 x 4 rectangle, etc) leaves a total of twelve distinct surface shapes.

First, a geometrical fact must be stated (Mackworth 1974a). Suppose one is given (i) the true shape of a surface in the form of a polygon whose dimensions may be uniformly scaled up or down by a factor, k, (ii) the projected shape of that surface, and (iii) three or more pairs of noncollinear points on the true and projected shapes that correspond. From this information it is easy to compute whether the true shape could produce the projected shape and, if it does, the value of k and the gradient of the surface.

For each picture region, by considering the topologically identical surfaces, a set of possible surfaces, each with a corresponding k and gradient, could be computed. If that set is empty, the region must depict a partially occluded surface.

This is now a labelling situation comparable to the corner-labelling algorithms of Huffman, Clowes, and Waltz. In those algorithms each junction has associated with it a set of possible corners; the aim of the interpretation is to discover a unique corner corresponding to each junction. Here, besides labelling each edge, the aim is to assign a unique surface to each region. Agreement between the interpretations of adjacent regions is necessary if the edge is taken to be connect. The agreement takes two distinct forms. First, the POLY coherence rules must be satisfied, and second, model-based coherence rules must be used. Such model-based rules would, at the lowest level, be of the form: Are there two such surfaces meeting at an edge in the set of prototypes? If so, do those surfaces meet at this dihedral angle? Do they agree on the scale factor? Higher levels would also be required: Are there three such surfaces meeting at a corner?

Procedurally, this approach need not be implemented in a depth- or breadth-first fashion. It is amenable to the two-stage Waltz search procedure which would first weed out the lists of possible surfaces (just as Waltz weeded out the lists of possible corners) on the basis of consideration of the mutual interpretation of each pair of adjacent regions and only then try to build complete, coherent interpretations.

7 Conclusion

World knowledge of the type incorporated as models in scene-analysis programs is an essential component of any psychological theory that attempts to explain human competence in interpreting line drawings as polyhedra. Furthermore, in those programs that knowledge is used in a procedural fashion; they demonstrate, at the very least, how a scene interpretation can be achieved.

The discussion of section 5 has pointed out some of the ways in which the available range of models is deficient for purposes of psychological explanation. The two proposals of section 6 are designed to provide mechanisms that reflect particular human competence in this task domain.

Acknowledgements. The author is grateful to Max Clowes, Rod Cowie, Aaron Sloman and Stuart Sutherland for discussion and criticism.

The research reported here was carried out in part at the University of Sussex under a grant from the Science Research Council to Professor M B Clowes and in part at the University of British Columbia under National Research Council of Canada Operating Grant A9281.

References

Attneave F, Frost R, 1969 "The determination of perceived tridimensional orientation by minimum criteria" *Perception and Psychophysics* **6** 391-396

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

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

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

Clowes M B, 1972b "Artificial intelligence as psychology" *AISB Bulletin* **1** November

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* (New York: Charles Scribner's Sons)

Guzman A, 1968 "Decomposition of a visual scene into three-dimensional bodies" *AFIPS Proceedings of the Fall Joint Computer Conference* **33** 291-304

Hochberg J, 1968 "In the mind's eye" in *Contemporary Theory and Research in Visual Perception* Ed. R N Haber (New York: Holt, Rinehart and Winston) pp 309-331

Hochberg J, Brooks V, 1960 "The psychophysics of form: reversible-perspective drawings of spatial objects" *American Journal of Psychology* **73** 337-354

Huffman D A, 1971 "Impossible objects as nonsense sentences" *Machine Intelligence 6* Eds B Meltzer, D Michie (Edinburgh: Edinburgh University Press) pp 295-323

Kopferman H, 1930 "Psychologische Untersuchungen über die Wirkung zweidimensionaler Darstellungen körperlicher Gebilde" *Psychologische Forschung* **13** 293-364

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" Ph. D. Thesis (unpublished), Laboratory of Experimental Psychology, University of Sussex, Brighton, England

Mackworth A K, 1974b "Using models to see" *Proceedings of the AISB Summer Conference* pp 127-137

Mackworth A K, 1975a "Consistency in networks of relations" technical report 75-3, Department of Computer Science, University of British Columbia

Mackworth A K, 1975b "How to see a simple world" technical report 75-4, Department of Computer Science, University of British Columbia

Michie D M, 1974 "On not seeing things" in *On Machine Intelligence* (New York: John Wiley) pp 112-133

Minsky M L, 1975 "A framework for representing knowledge" in Winston (1975) pp 211-277

Minsky M L, Papert J S, 1972 "Progress report" AI memorandum 252, Massachusetts Institute of Technology, Cambridge, Mass.

Penrose L S, Penrose R, 1958 "Impossible objects: a special type of illusion" *British Journal of Psychology* **49** 31-33

Roberts L G, 1965 "Machine perception of three-dimensional objects" in *Optical and Electro-optical Information Processing* Eds J T Tippett, D A Berkowitz, L C Clapp, C J Koester, A Vanderburgh (Cambridge, Mass.: MIT Press) pp 159-197

Shepard R N, Metzler J, 1971 "Mental rotation of three-dimensional objects" *Science* **171** 701-703

Southall J P C (Ed.), 1962 *Helmholtz' Treatise on Physiological Optics* volume III (New York: Dover)

Sutherland N S, 1973 "Some comments on the Lighthill report and on Artificial Intelligence" in *Artificial Intelligence: A Paper Symposium* (London: Science Research Council) pp 22-31

Waltz D L, 1972 "Generating semantic descriptions from drawings of scenes with shadows", MAC AI-TR-271, Massachusetts Institute of Technology, Cambridge, Mass. [Also in Winston (1975) pp 19-91]

Winograd T, 1972 "Procedures as representation for data in a computer program for understanding natural language" MAC AI-TR-84, Massachusetts Institute of Technology, Cambridge, Mass.

Winston P H, 1968 "Holes" AI memorandum 163, Massachusetts Institute of Technology, Cambridge, Mass.

Winston P H, 1970 "Learning structural descriptions from examples", MAC AI-TR-76, Massachusetts Institute of Technology, Cambridge, Mass. [Also in Winston (1975) pp 157-209]

Winston P H, 1973 "The MIT robot" *Machine Intelligence 7* Eds B Meltzer, D Michie (Edinburgh: Edinburgh University Press) pp 431-463

Winston P H (Ed.), 1975 *The Psychology of Computer Vision* (New York: McGraw-Hill)