object. In such cases, it is convenient to parameterize the camera transform by the measurable camera position and orientation parameter. One such technique is described in [4].

3.4 SUMMARY

Analysis of simple scenes of polyhedra with limited occlusion was described in this chapter. These techniques are strongly limited by requiring a priori models of the specific objects that may be present in the scene. In the next chapter we discuss the analysis of occluding scenes without the knowledge of such models.

REFERENCES

[1] L. G. Roberts, "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing*, J. T. Tippett et al. (eds.), MIT Press, Cambridge, Mass., 1968, pp. 159-197.

[2] F. Attneave, "Some Informational Aspects of Visual Perception," *Psychology Review*, Vol. 61, 1954, pp. 183-193.

[3] E. Freuder, "Structural Isomorphism of Picture Graphs," in *Pattern Recognition & Artificial Intelligence*, C. H. Chen (ed.), Academic Press, New York, 1976, pp. 248-256.

[4] I. Sobel, "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes," *Artificial Intelligence*, Vol. 5, No. 2, 1974.

[5] D. B. Gennery, "Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision," Stanford Artificial Intelligence Laboratory Memo AIM-339 (Ph.D. thesis), June 1980.

[6] M. A. Fischler and R. C. Bolles, "Random Samples Consensus: A Paradigm for Model Filtering with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, June 1981, pp. 381-396.

4

COMPLEX SCENES OF POLYHEDRA

The scene analysis techniques of the last chapter, though general in principle, are likely to be computationally inefficient as the scenes get more complex. As the number of models grows and large parts of objects are occluded by others, recognition by matching with specific models becomes increasingly more difficult and expensive. A major simplification occurs if the lines, vertices, and faces belonging to different objects can be separated. Such *segmentation* is the major subject of this chapter. After parts of complete objects have been segmented, complex objects or structures can be described by relationships of these parts. Structural descriptions are covered in the later parts of this chapter.

4.1 SEGMENTATION OF POLYHEDRAL SCENES

Consider the picture shown in Fig. 4-1 (the polygonal regions have been numbered for convenience). Most human observers would agree that it consists of one rectangular block occluding another. Here, we will be interested in techniques for separating the two objects, without the knowledge of specific objects in the scene (they are only constrained to be polyhedral). A simple technique that establishes relationships between regions surrounding a vertex to accomplish segmentation was devised by Guzman in 1968 [1, 2].

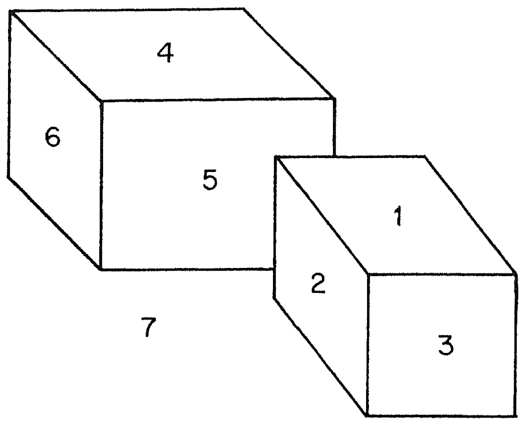


Figure 4-1: A simple scene

Guzman enumerated classes of vertices shown in Fig. 4-2. The type of a vertex is determined by the angular relationships between the lines forming the vertex. Each vertex provides evidence of whether regions surrounding it belong to the same body, as indicated by *links* between regions shown in Fig. 4-2. An absence of a link between two regions (as for a *T*-vertex) indicates that no connections between surrounding regions can be inferred.

Figure 4-3 shows a graph constructed from Fig. 4-1, by representing each region as a node and placing an arc between two nodes as indicated by Fig. 4-2 and examining each vertex. All links to the background (region 7), which is assumed to be known, are ignored. This graph separates into two groups such that the nodes in each group are linked to other nodes in that group by at least two links. One group consists of nodes 1, 2, and 3, and the other of 4, 5, and 6. These two separate groups correspond to the two desired distinct bodies.

Guzman's technique has been successfully applied to fairly complex scenes, such as the one shown in Fig. 4-4. The basic method is to construct a graph as above and group the nodes such that a node in each group is connected to at least one other node in that group by two or more links. In complex scenes this may result in some isolated groups but also some groups connected to others by a single link. The latter groups are merged if additional evidence of connection by a *weak* link is available. A weak link is formed between two faces of an arrow vertex, in addition to a strong link, if this vertex is also a *leg* vertex. A leg vertex is an arrow vertex in which one of the sides bends to become parallel to the stem or the center line of the arrow; three examples are shown in Fig. 4-5. A group consisting of a single region and connected to another group is merged into the latter.

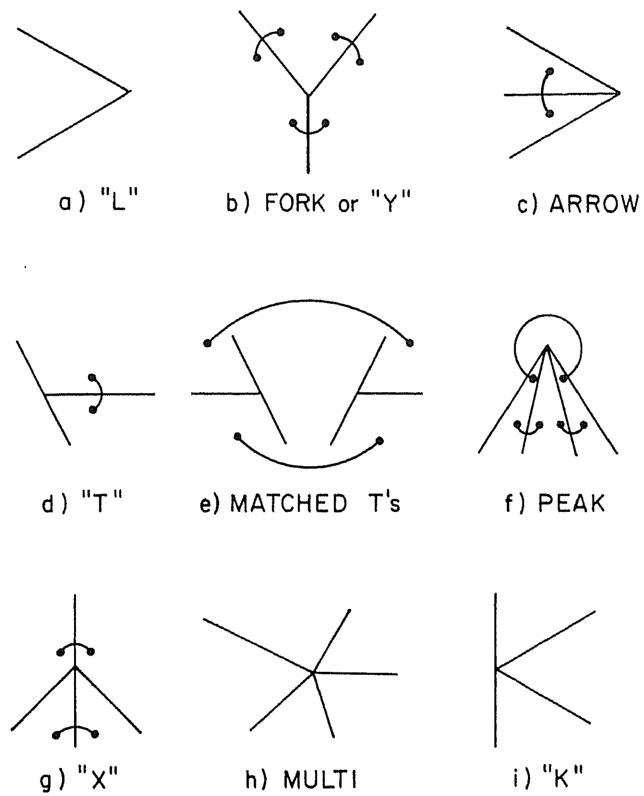


Figure 4-2: Vertex types and links

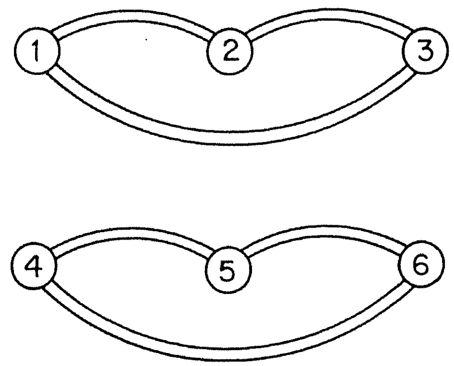


Figure 4-3: Region connectivity graph for scene of Fig. 4-1

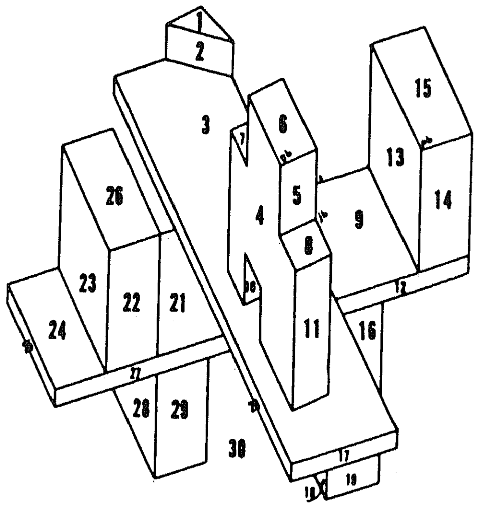
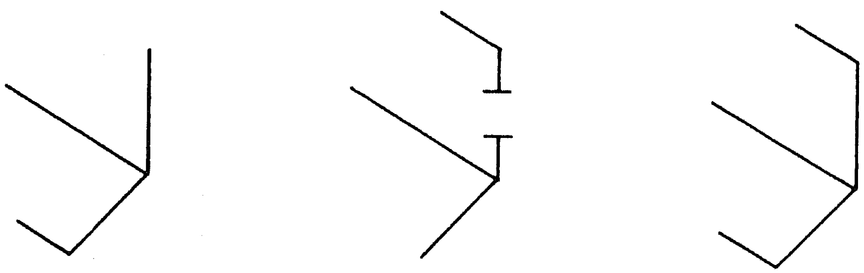


Figure 4-4: A "bridge" (from Guzman [2])



(a) (b) (c)

Figure 4-5: Three leg vertices

This simple strategy is sufficient to segment the scene in Fig. 4-4 satisfactorily and is a powerful demonstration of how global object characteristics can be inferred from rather local vertex characteristics. This technique is, however, ad hoc—hence its performance is difficult to predict and characterize. It is sensitive to certain accidental alignments and performs poorly on objects with holes. Note that perfect line drawings as input and complete absence of shadows are assumed. The reader may be interested in devising useful linking rules for more general scenes, such as the one shown in Fig. 4-6. An extension of Guzman's techniques for imperfect line drawings is given in [3].

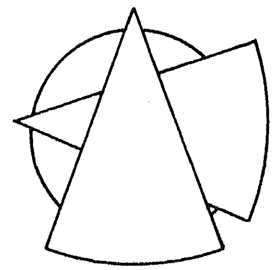


Figure 4-6: Occluding curved objects

A formal study of polyhedral objects is presented next that justifies many of the heuristic rules used in Guzman's technique.

4.2 CLASSIFICATION OF LINES

Figure 4-7 shows three line drawings that seem to represent "impossible objects" in the sense that they cannot be the image of any solid polyhedral object, yet they are constructed of seemingly permissible and ordinary line and vertex structures. Actually, the object of Fig. 4-7(b) is possible, the other two are not. Huffman and Clowes, independently, studied the problem of describing lines and vertices such that whether a line drawing represents a solid object may be determined algorithmically [4-5]. Their classification scheme also provides a method for scene segmentation.

In this analysis, it is assumed that the objects are solid, and are viewed from a *general* position, defined to be one where a small change in viewing angle does not cause lines and vertices to appear or disappear—that is, there are no accidental alignments. Under these conditions, a line in a picture corresponds to an edge in an object formed by an intersection of two faces. This edge must be one of the

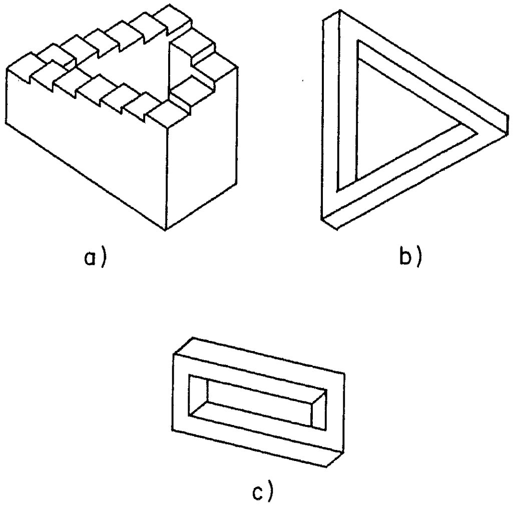


Figure 4-7: Three impossible-looking "objects"

following four types: a *convex* edge where the two faces recede away from the edge from the viewer's vantage point, a *concave* edge where the faces are toward the viewer, an *obscuring* edge where only one face is visible and the visible face is to the right of the edge and, finally, an *obscuring* edge where the visible face is to the left of the edge. Figure 4-8 shows an object with its edges labeled, where a "+" represents a convex edge, "−" a concave edge, and arrows the obscuring edges with the visible face to the right of the directed line.

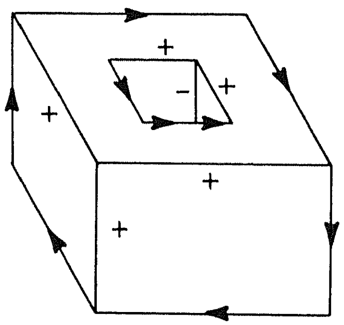


Figure 4-8: An object with labeled edges

Now, consider the vertices formed by the intersection of the faces of a polyhedron. Let the objects be restricted to be trihedral; that is, each vertex is formed by intersection of exactly three faces. If objects

are viewed from a general position, defined to be such that the vertex and the line formations do not change with small changes in the viewing angle, then only arrow, fork, L, and T types of vertices can occur.

The major observation of the Huffman-Clowes discovery is that the lines at different vertices are not allowed to have all combinatorially possible labels, but are constrained to be one of the configurations shown in Fig. 4-9. These configurations can be derived by considering the three planes forming a vertex to divide space into octants. Now the solid object may fill one or more of these octants. However, strictly speaking, the objects filling an even number of octants do not form trihedral vertices. Typical object vertices with 1, 3, 5, and 7 octants filled are shown in Fig. 4-10. The possible visible vertex configurations are derived by viewing the four different types of vertices from all eight octants (each view does not necessarily give a distinct configuration).

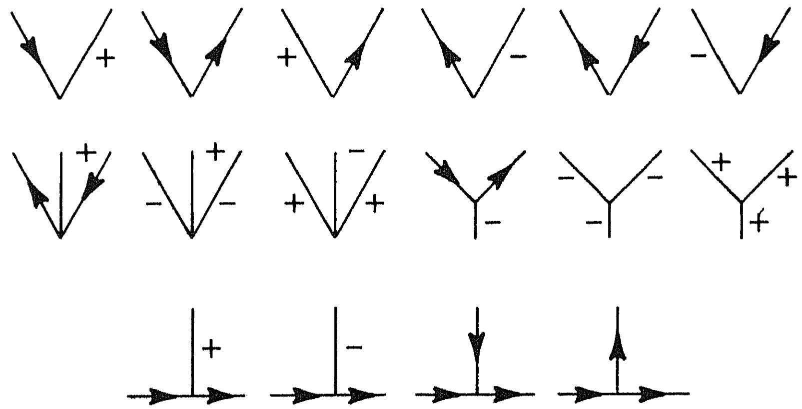


Figure 4-9: Allowed junction labels

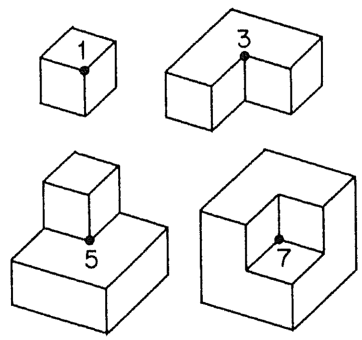


Figure 4-10: Four types of vertices with numbers indicating the number of filled octants (after Huffman [4])

A necessary condition for a line drawing to represent a possible object then is that its lines be labeled so that the vertices are labelled as in Fig. 4-9. It is clear that the label of a line cannot change between vertices. For some objects, multiple labelings are possible; such objects are ambiguous. The labeling for the object shown in Fig. 4-8 is consistent and hence it may be possible. However, no consistent labelings can be found for the line drawing in Fig. 4-11 and it cannot represent a possible solid object (to prove this, note that the lines on the outer boundary must be labeled as obscuring edges). These consistent labeling requirements are, however, only the *necessary* and not *sufficient* conditions. (The sufficient conditions are described in Section 4.3.)

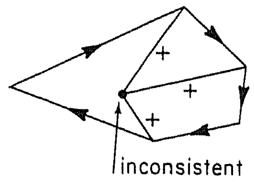


Figure 4-11: An object with no consistent labels

For a scene consisting of multiple objects, the line labels provide a straightforward segmentation; all regions connected by a line having a "+" or a "−" labeling must belong to the same body (for example, try to label Fig. 4-1). It is interesting to see that this theory supports the heuristic rules in Guzman's segmentation scheme. The center line of the arrow vertices in Fig. 4-9 has the label of "+" or "−" only. This implies that the two surrounding regions actually intersect and must belong to the same body. Similarly, two of the three fork configurations support Guzman's linking rules, but the existence of a third configuration is a possible source of errors (in Guzman's experiments, arrows were found to be more reliable than forks). The requirement of two links in Guzman's programs is also justified, as a line cannot change its label between vertices and must produce a similar linking evidence on both ends.

4.2.1 Inclusion of Shadows

Waltz [6,7] extended the types of line labels by including *crack* and *shadow* edges, as shown in Fig. 4-12. The crack edges are marked by a *C* along them and shadow edges by an arrow across them, with the dark region toward the arrowhead. Waltz also differentiated between subclasses of some types of lines. For example, a concave edge may be formed by two faces belonging to the same body or to two different bodies. The latter case is further distinguished by whichever of the two

objects, if any, supports the other. Further distinctions are added by including illumination information for the regions on two sides of a label. A region is either illuminated, or in the shadow of another object or another face of the same body. With such distinctions, 59 separate line labels were enumerated.

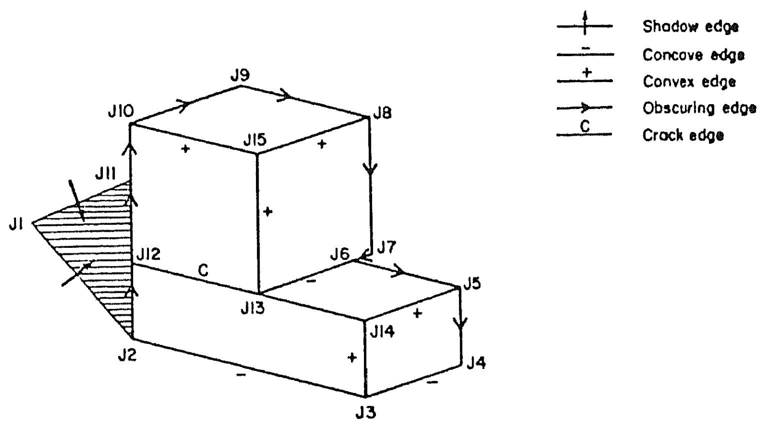


Figure 4-12: An object with cracks and shadows (from Waltz [7])

Again, not all combinationally possible sets of line labels at a vertex are physically possible. As a simple example, a shadow edge cannot occur at a type 1 vertex (the object occupying one octant). Waltz enumerated several thousand physically possible configurations for different types of vertices, including some caused by accidental alignments (such as *K* and multi). This number is a very small fraction of the combinationally possible labelings, which number in the tens of millions.

With this expanded set, certain types of vertices may have labels numbering in the hundreds, and finding a consistent set of labels for a complex line drawing might seem to be a computationally hopeless task. However, surprisingly, a simple algorithm described below has been found to converge remarkably rapidly.

Consider three vertices *A*, *B*, and *C* as shown in Fig. 4-13, which are part of a larger line drawing. Three sets S_A, S_B, and S_C of possible line configurations may be initially assigned to the three respective vertices. However, the line *AB*, must have the same label at the vertices *A* and *B* and hence those labels in S_A and S_B which do not assign a common label to the line in between them may be deleted from further consideration. The same procedure is now applied to the line *BC* and sets S_B and S_C. If any labels are deleted from the set S_B in this process, then more labels from S_A may now become ineligible. This process propagates to all connected vertices, considering one pair of

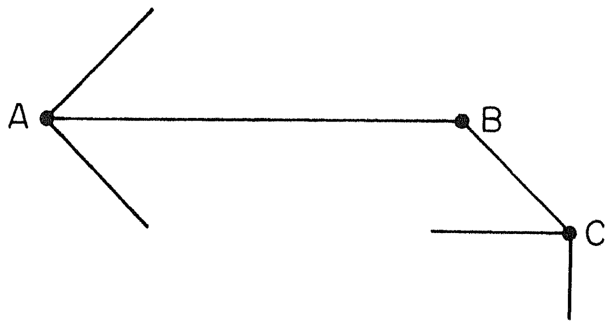


Figure 4-13: Part of a line drawing showing three vertices

vertices at a time.

In the example of Fig. 4-13, three labels can be assigned to each of the vertices *A* and *C*, and six labels to the vertex *B* without considering consistency with their neighbors. However, the labels of vertex *A* eliminate the possibility of line *AB* being occluding, and only two labels at vertex *B* can be retained, one with line *AB* being marked "+" with an arrow pointing from *C* to *B*, the other with line *AB* being "−" and an arrow pointing from *B* to *C*. However, only the latter is consistent with allowed labels of *C*. This comparison results in a unique label for both vertices *B* and *C*. This in turn also forces a unique label on vertex *A*, requiring *AB* to be "−".

It was found that the above simple algorithm applied iteratively to all vertices in a line drawing converged very fast, the number of possible labels decreased rapidly, and a unique labeling for lines was obtained in non-ambiguous figures. Thus, pairwise consistency of vertices results in global scene consistency. This agrees with our intuitive notion that a vertex does not affect the scene content very far away from it. Waltz's algorithm was successful in correctly labeling scenes of the complexity shown in Fig. 4-14. This labeling algorithm is also called *relaxation labeling*.

Labeling constraints for nonsolid objects such as wire frame objects, and objects with walls that are arbitrarily thin, have also been derived [8-10]. However, all such analysis assumes that the given line drawings are perfect and contain no missing or extra lines. Such line drawings are extremely difficult, if not impossible, to obtain even from pictures of carefully prepared, simple, homogeneous-surfaced objects. The practical utility of these procedures, even for polyhedral objects, has thus been limited. A very different technique for imperfect line drawings, using properties of a group of lines and vertices of specific objects known to be present in the scene, is described in [11].

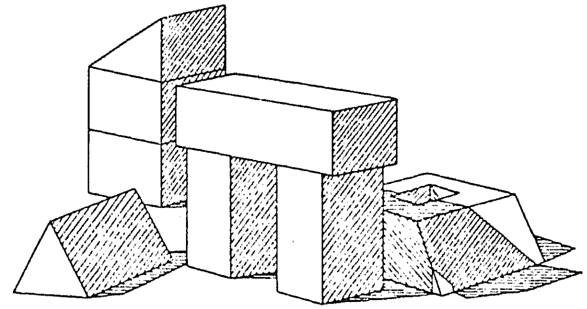


Figure 4-14: A complex scene (from Waltz [7])

4.3 GEOMETRICAL CONSTRAINTS FOR POSSIBLE OBJECTS

The Huffman-Clowes labels provide a way of verifying if a given line drawing is physically realizable. However, they provide only a necessary condition. The line drawing in Fig. 4-15 can be labeled consistently but is still not realizable as a trihedral object, as the three lines *AB*, *CD*, and *EF* do not meet in a point. We need to place some geometrical constraint on the line drawings, in addition to the syntactic constraints. The use of *dual graphs* for this purpose was suggested by Huffman [4] and further developed by Mackworth [12].

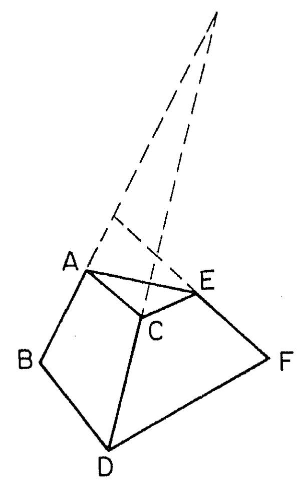


Figure 4-15: An "impossible" pyramid

4.3.1 Gradient Spaces and Dual Graphs

A convenient representation of the orientations of the surfaces of an object is needed for expressing the necessary constraints for these surfaces to belong to a possible object. The orientation of a plane defined by the equation

$$ax + by + cz + d = 0 \tag{4-1}$$

is given by the direction numbers (a, b, c) of its normal. However, a more useful representation can be derived by rewriting Eq. (4-1) as

$$-z = \left(\frac{a}{c}\right)x + \left(\frac{b}{c}\right)y + \frac{d}{c} \tag{4-2}$$

assuming c is not 0. Now let G_x and G_y be defined as below:

$$G_x = \frac{\partial z}{\partial x} = -\frac{a}{c} \tag{4-3}$$

and

$$G_y = \frac{\partial z}{\partial y} = -\frac{b}{c} \tag{4-4}$$

G_x and G_y are the gradients of the z components of the points in the plane in the x and y directions, respectively. (G_x, G_y) may be viewed as a two-dimensional *gradient space*; the orientation of each plane in the (x, y, z) space is uniquely represented by a point in the gradient space (except when c is 0). A point at the origin of the gradient space represents a plane parallel to the (x, y) plane and a point on one of the axes of the gradient space represents a tilt of this plane along the x or the y axes.

Most of the useful properties of the gradient space derive from a dual relation between the (x, y, z) space and a dual (u, v, w) space. These relationships are described as background in the next two paragraphs, and they may be skipped without loss of continuity.

A dual of the plane given by Eq. (4-1) in the (u, v, w) space is defined to be the point (a, b, c). The dual of a point (d, e, f) in the (x, y, z) space is defined to be the plane $du + ev + fw + 1 = 0$ in the (u, v, w) space. A straight line in the (x, y, z) space can be viewed as the intersection of a number of planes containing this line. The dual of this line in the (u, v, w) space is another straight line, passing through the points that are the duals of the planes determining the straight line in

the (x, y, z) space.

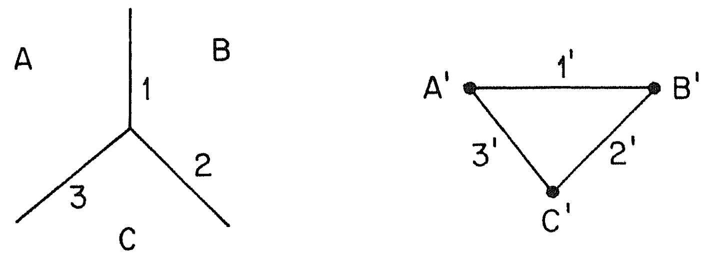
A picture of a solid object is a projection from three-dimensional space onto a picture plane. In the following we restrict the projection to be orthographic. (Orthographic projection is equivalent to a perspective projection with the viewing point at an infinite distance. For object sizes much smaller than the viewing distances, the two projections are very similar.) An interesting correspondence exists between the picture and the projection of the dual space onto a specially chosen plane called the gradient plane, or the gradient space. We state, without proof, these relations between the lines and points in a picture with their dual representations in the gradient space.

A plane (polygon) in the picture plane corresponds to a point in the gradient space (G_X, G_Y). The location of this point in gradient space is determined by the gradient of the plane in the three-dimensional space along the X and Y axes, respectively, the picture projection being along the Z axis. A line in the picture plane corresponds to a line in the gradient space such that the two lines are perpendicular to each other (assuming that the axes of the two spaces are aligned).

An example of a vertex with three lines and three faces surrounding it is shown in Fig. 4-16(a), and its dual is shown in Fig. 4-16(b). The dual of the plane A is an arbitrarily chosen point A'. B', dual of B, must line on a line $A'B'$ normal to line 1. For line 1 to be convex, $A'B'$ must be in the same orientation along line 1′ as A and B are across line 1—that is, left to right in this example (conversely for a concave edge). Length of line $A'B'$ determines the precise relative orientations of the planes A and B and cannot be determined from the line drawing. However, once A' and B' have been chosen, C', the dual of the plane C, is uniquely determined as it must lie on line 2′ through B normal to 2 and line 3′ through A normal to line 3. Note that for a consistent dual to exist, all three edges must be either all convex or all concave; the convex case being shown in Fig. 4-16 (the size of the triangle $A'B'C'$ can, under certain conditions, be determined by using the intensity values of the three faces. See Section 9.3 and [13] for more details.)

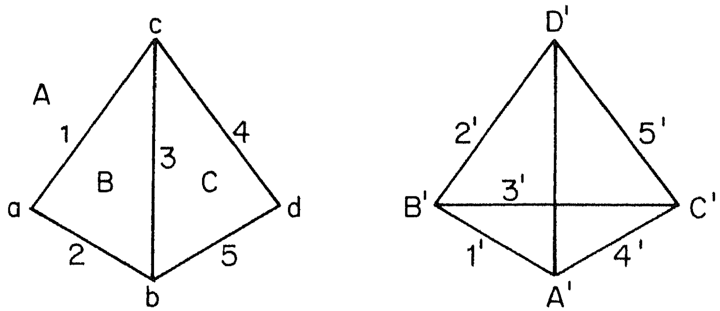
The geometrical condition for an object to be physically realizable is simple—it should be possible to construct a consistent dual for the object in the picture. This is both a necessary and a sufficient condition. For example, consider the object in Fig. 4-17(a). It appears, to us, to be a tetrahedron with two obscured faces. For simplicity, we will construct a dual in the gradient space assuming that the background plane, A, is also one of the hidden surfaces.

Duals of the planes A, B, and C are given by A', B', and C', forming a triangle as in Fig. 4-16. The position and the size of the triangle can not be fixed. Also, we are assuming that edges 1 and 4 are



(a) (b)

Figure 4-16: Three planes at a vertex and their dual graph



(a) (b)

Figure 4-17: An object and its dual graph

concave and 3 is convex (one of the few possible and consistent interpretations). This interpretation implies that edges 2 and 5 must be occluding. It is interesting to determine if the occluded planes for these two edges could be the same! For this, the dual of the hidden plane, say D, must lie at the intersection of line $B'D'$ normal to line 2 and line $C'D'$ normal to line 5. Additionally, line $A'D'$ must be normal to line *ad*, as the plane D must also intersect the plane A along line *ad*, as is the case for this example. (A dual for the interpretation where A is a background plane and another plane E is another hidden surface of the tetrahedron is easily constructed replacing A' with E' in Fig. 4-17 and a new A' as an isolated point.)

This ability to impose constraints on invisible surfaces is quite remarkable. Mackworth has implemented a computer program to hypothesize the hidden surfaces, where possible by using reasoning such as given above [12]. The reader may find it instructive to verify that a consistent dual can not be constructed for the pyramid of Fig. 4-15 by assuming just one hidden plane intersecting line AE (but can be by using two hidden planes in the back).

4.4 DESCRIPTIONS OF OBJECT ASSEMBLIES

A *description* of a given scene is a major objective of the machine-perception process. A description of a scene should include the number and the type of objects in it and their relationships to each other. As an example, the scene of Fig. 4-18 may be described as consisting of three rectangular parallelepiped blocks (bricks), with the block A supported by the blocks B and C, and with the block B to the left of block C. Each block may in turn be described in terms of its components, such as its faces.

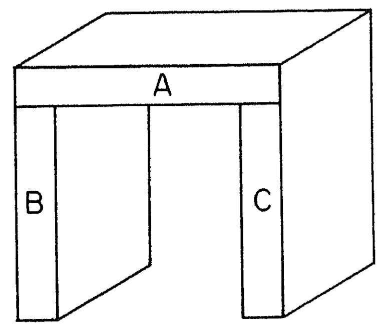


Figure 4-18: An arch

Such descriptions can be conveniently represented as a graph structure, with the objects (or components) being nodes of the graph and relationships between them being arcs of the graph. The graph representation of the above description of the scene in Fig. 4-18 is shown in Fig. 4-19. Such descriptions are useful for recognition of object assemblies.

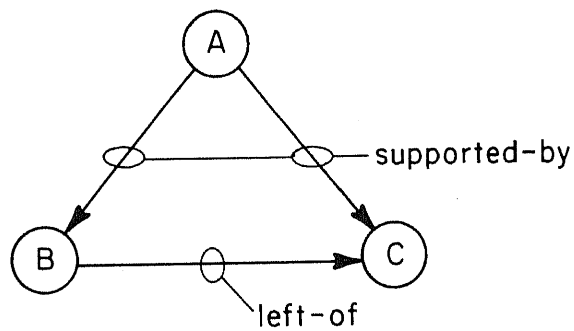


Figure 4-19: A simple graph description of the arch of Fig. 4-18

4.4.1 Computation of Descriptions

Computation of descriptions that require inference of relationships among objects is difficult. Relationships such as support are easy for us to infer, but difficult to give algorithms for. Techniques for inferring such relationships are not well developed and are, in general, heuristic in nature. A few such techniques were suggested by Winston [14, 15] and are discussed below. In the following, it is assumed that the individual objects of a scene have been separated by one of the previously described schemes.

Support. To infer that an object *A* is supported by another object *B*, object *A* must obscure some part of *B*. Also, assuming a normal viewing position, *B* will usually appear to be below *A*. Winston suggests that if the "bottom lines" of an obscuring object form an arrow vertex and share regions with the obscured object, then it may be inferred that the latter object supports the former (see Fig. 4-20). (This is easily extended to X or K vertices.) The bottom lines are those belonging to the lower vertices of the interior lines.

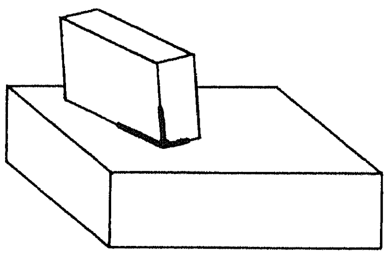


Figure 4-20: Evidence for support relation

Note that gravity and stability of objects have not been explicitly used, and the above algorithm is likely to lead to occasional errors. It is likely that these concepts play an important part in our inference of support relationships. However, they are difficult to compute from two-dimensional images.

Left and Right. Figure 4-21 shows various examples of two blocks in a scene. Some of the examples clearly depict a left-right relationship, while the others may appear to be front-behind relations. Based on these examples, we may define an object *A* to be to the left of *A* if the center of area of *A* is left of the leftmost point of *B* and the rightmost point of *A* is left of the rightmost point of *B*. The definition for *A* to be right of *B* is similar. (This definition is not necessarily symmetric; that is, *A* may be left of *B*, but *B* not right of *A*.) For ambiguous cases, such as in Fig. 4-21, it may be desirable to generate

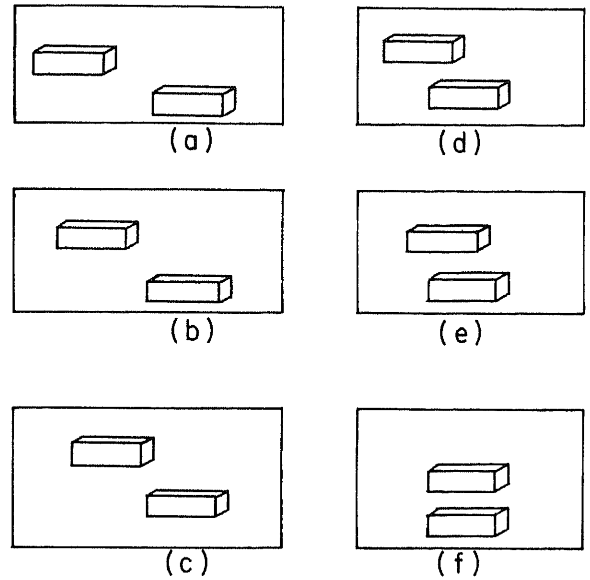


Figure 4-21: Left-right relations (after Winston [15])

more than one alternative description.

Groups of objects. A set of objects form a group if they are similar in some respects and share common relationships with others. In Fig. 4-22, blocks *A*, *B*, and *C* can be considered to belong to a group, as they are similar and are supported similarly. The latter relationship can be inferred from the chain of supported-by pointers. As another example, the four legs of a four-legged table may be considered to belong to a group. Here, the four legs must be similar, and they share the same relationship with the table top, namely that of supporting it.

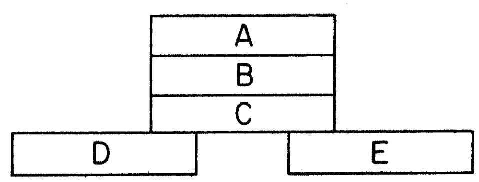


Figure 4-22: Example for a similarity relation

The relationships described above are only a subset of the relations used, by humans, and the description algorithms are far from being adequate. However, such descriptions are still useful for recognizing object assemblies. An interesting application to learning of structures was demonstrated by Winston and is described next.

4.4.2 Learning of Structural Descriptions

Suppose that a program is capable of generating the description shown in Fig. 4-19 from the scene of Fig. 4-18. If this program is next shown the scene in Fig. 4-23, it will generate a description similar to that of Fig. 4-19, with an additional link between nodes *B* and *C* indicating that they touch. A simple program that compared two graph descriptions would discover this similarity and difference (graph-matching techniques are discussed later in Chapter 5). If now the program is told that Fig. 4-23 is not acceptable as an arch, its concept of arch can be modified to contain the information that the supporting blocks *B* and *C* must not touch.

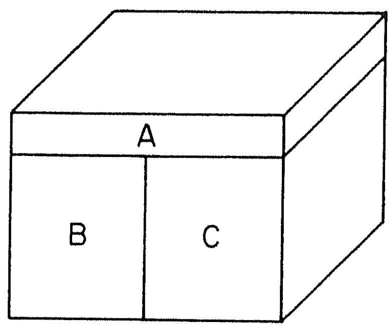


Figure 4-23: Almost an arch

The concept of an arch can be further elaborated by showing examples of other near-miss arch scenes—for example, the block *A* not being supported by the other two. On the contrary, if the top block is a pyramid and the structure is still called an arch, then the constraints on this object can be relaxed.

This type of concept learning should be distinguished from parameter learning in classical pattern recognition techniques, where only the weights of a decision function are modified. The learning described here results from a careful choice of near-miss sample sequences. It may be easier to teach a machine in this way than by reprogramming, but such learning is far from the usual notion of discovery with little or no help from a teacher.

4.5 SUMMARY

This chapter concludes our treatment of the analysis of scenes of polyhedra. In the analysis of the previous two chapters we assumed that the input to the programs was a perfect line drawing. Such line drawings are rarely available, and the techniques required for the analysis of imperfect or incomplete line drawings may need to be very different from those discussed previously. The described techniques are, thus, for the analysis of line drawings rather than for the analysis of polyhedral scenes. Also, the techniques are specific to polyhedra and may not extend to a more general class of objects.

Nonetheless, some useful lessons can be learned from the study of these techniques. The problem of three-dimensional scene analysis has proved to be much more difficult than anticipated and seems to require a deep understanding of the problem domain. It seems that general techniques that ignore the semantics of scenes are inadequate, even for simple scenes. The more knowledge one can incorporate in a program, the better is the performance (for example, Waltz's program can analyze more complex scenes than Guzman's). Finally, the description of scenes seems to require extensive knowledge of physical principles to adequately infer simple relations such as support and stability.

REFERENCES

[1] A. Guzman, "Computer Recognition of Three-Dimensional Objects in a Scene," MIT Report MAC-TR-59, December 1968.

[2] A. Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies," *AFIPS Proceedings Fall Joint Comp. Conf.*, Vol. 33, 1968.

[3] G. Falk, "Interpretation of Imperfect Line Data as a Three Dimensional Scene," *Artificial Intelligence*, Vol. 3, Issue 2, 1972.

[4] D. A. Huffman, "Impossible Objects as Nonsense Sentences," in *Machine Intelligence 6*, B. Meltzer and D. Michie (eds)., Edinburgh University Press, Edinburgh, 1971.

[5] M. B. Clowes, "On Seeing Things," *Artificial Intelligence*, Vol. 2, Issue 1, 1971, pp. 79-116.

[6] D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," MIT Report AI-TR-271, November 1972.

[7] D. L. Waltz, "Understanding Line Drawings of Scenes with Shadows," in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw-Hill, New York, 1975, pp. 19-91.

[8] P. V. Sankar, "A Vertex Coding Scheme for Interpreting

Ambiguous Trihedral Solids," *Computer Graphics and Image* Processing, Vol. 6, 1977, pp. 61-89.
[9] T. Kanade, "A Theory of the Origami World," *Artificial Intelligence*, Vol. 13, 1980, pp. 279-311.
[10] K. Sugihara, "Picture Language for Skeletal Polyhedra," *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 382-405.
[11] G. R. Grape, "Model Based (Intermediate-level) Computer Vision," Stanford Artificial Intelligence Laboratory, Memo AIM-201, May 1973.
[12] A. K. Mackworth, "Interpreting Pictures of Polyhedral Scenes," *Artificial Intelligence*, Vol. 4, 1973, pp. 121-137.
[13] B. Horn, "Understanding Image Intensities," *Artificial Intelligence*, Vol. 8, 1977, pp. 201-231.
[14] P. H. Winston, "Learning Structural Descriptions from Examples," MIT Report MAC-TR-76, September 1970.
[15] P. H. Winston, "Learning Structural Descriptions from Examples," in *The Psychology of Computer Vision,* P. H. Winston (ed.), McGraw-Hill, New York, 1975, pp. 157-209.

5

SHAPE ANALYSIS AND RECOGNITION

Scenes of interest rarely contain solely polyhedral objects. The motive in studying the problems of the blocks world was to better understand some aspects of the perceptual process. The main problems encountered were of the low-level processes to obtain object boundaries, segmentation of scenes containing multiple occluding objects, recognition of objects under the changes of scale, rotation, perspective, and varying amounts of occlusion, and descriptions of object assemblies. We now examine the generalization of these processes for nonpolyhedral objects, concentrating on the problems of shape analysis and recognition based on these descriptions.

Boundary-extraction techniques of polyhedral objects generalize to other objects, provided that the surfaces of objects are still homogeneous and shading of the surfaces due to curvature is not strong. These techniques must now be more local, as boundaries are not necessarily straight. If the objects or the background are textured, boundary extraction becomes a complex problem (for polyhedral objects as well). These low level processes will be discussed in detail in the succeeding chapters. For now, we assume that perfect boundaries, corresponding to discontinuities in the object surfaces or their slopes, are available.

Segmentation techniques for polyhedral objects were based on effective utilization of the knowledge of constraints placed on the images by the nature of the objects and the image-formation process.