End Inner Loop

End Outer Loop

END

---

Algorithms such as this involve many technical issues, such as merging coplanar faces, stitching edges together into faces, regularization of faces, removing multiple versions of edges. Boundary evaluation is inherently rather complex, and depends on such things as the definition and representation of faces as well as the geometric utilities taken as basic [Voelcker and Requicha 1981]. Boundary evaluation is an example of exact conversion between significantly different representations. Such conversions are useful, since no single representation seems convenient for all geometric calculations.

## 9.5 UNDERSTANDING LINE DRAWINGS

"Engineering" line drawings have been (and to a great extent are still) the main medium of communication between human beings about quantitative aspects of three-dimensional objects. The line drawings of this section are only those which are meant to represent a simple domain of polyhedral or simply curved objects. Interpretation of "naturalistic" drawings (such as a sketchmap [Mackworth 1977]) is another matter altogether.

Line drawings (even in a restricted domain) are often ambiguous; interpreting them sometimes takes knowledge of everyday physics, and can require training. Such informed interpretation means that even drawings that are strictly nonsense can be understood and interpreted as they were meant. Missing lines in drawings of polyhedra are often so easy to supply as to pass unnoticed, or be "automatically supplied" by our model-driven perception.

Generalizing the line drawing to three dimensions as a list of lines or points is not enough to make an unambiguous representation, as is shown by Fig. 9.27,

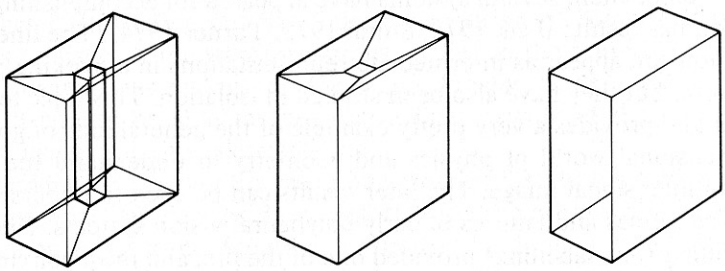


**Fig. 9.27** An ambiguous (wireframe) representations of a solid with two of three possible interpretations.

which illustrates that a set of vertices or edges can define many different solids. (It is possible, however, to determine algorithmically all possible polyhedral boundaries described by a three-dimensional wireframe [Markowsky and Wesley 1980].). A line drawing nevertheless does convey three-dimensional information. For any set of $N$ projection specifications (e.g., viewpoint and camera transform), a wire-frame object may be constructed that is ambiguous given the $N$ projections. However, for a given object, there is a maximum number of projections that can determine the object unambiguously. The number depends on the number of edges in the object [Shapira 1974]. Reconstruction of all solids represented by projections is possible [Wesley and Markowsky 1981].

Line drawings were a natural early target for computer vision for the following reasons:

1. They are related closely to surface features of polyhedral scenes.
2. They may be represented exactly; the noise and incomplete visual processing that may have affected the "line drawing extraction" can be modelled at will or completely eliminated.
3. They present an interpretation problem that is significant but seems approachable.

The understanding of simple engineering (3-view) drawings was the first stage in a versatile robot assembly system [Ejiri et al. 1971]. This application underlined the fact that heuristics and conventions are indispensible in engineering drawing understanding. This section deals with the problem of "understanding" a single-view line drawing representation of scenes containing polyhedral and simple curved objects like those in Fig. 9.28.

Our exposition follows a historical path, to show how early heuristic programs in the middle 1960s evolved into more theoretical insights in the early 1970s.

The first real computer vision program with representations of a three-dimensional domain appeared around 1963 [Roberts 1965]. This system, ambitious even by today's standards, was to accept a digitized image of a polyhedral scene and produce a line drawing of the scene as it would appear when viewed from any requested viewpoint. This work addressed basic issues of imaging geometry, feature finding, object representation, matching, and computer graphics.

Since then, several systems have appeared for accomplishing either the same or similar results [Falk 1972; Shirai 1975; Turner 1974]. The line drawings of this section can appear as intermediate representations in a working polyhedral vision system, but they have also been studied in isolation. This topic took on a life of its own and provides a very pretty example of the general idea of going to the three-dimensional world of physics and geometry to understand the appearance of a two-dimensional image. The later results can be used to understand more clearly the successes and failures of early polyhedral vision systems. One form of understanding (line labelling) provided one of the first and most convincing demonstrations of parallel constraint propagation as a control structure for a computer vision process.

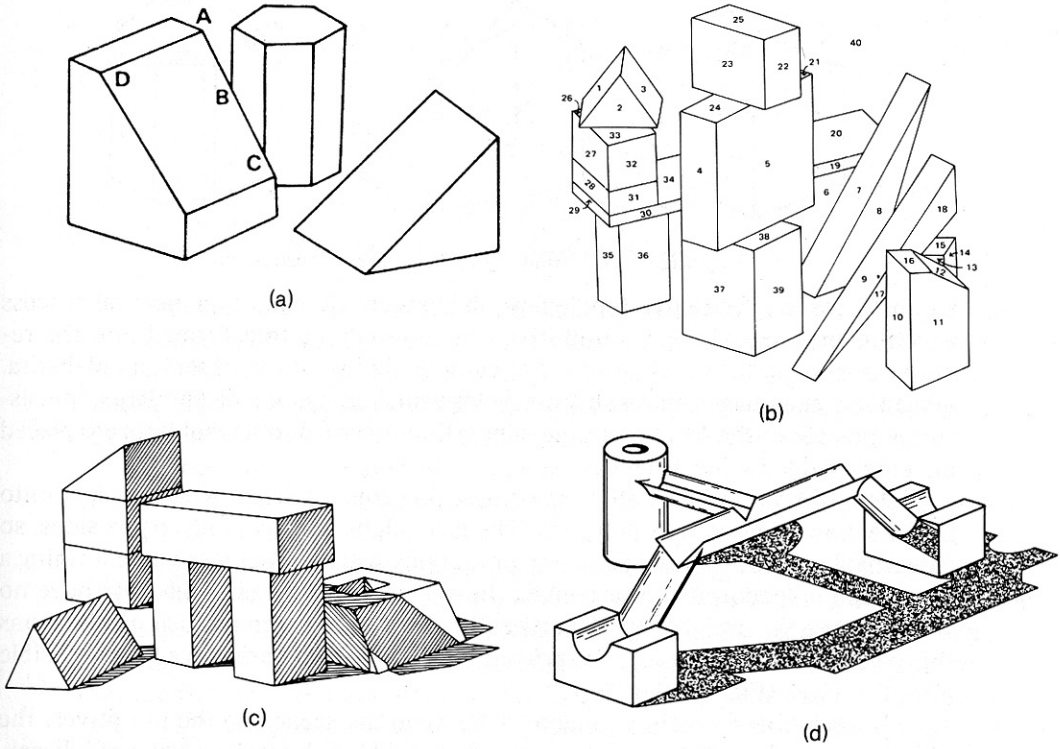



**Fig. 9.28** Several typical line drawing scenes for computer understanding.

### 9.5.1 Matching Line Drawings to Three-dimensional Primitives

Roberts desires to interpret a line drawing such as Fig. 9.28a in terms of a small set of three polyhedral primitives, shown in Fig. 9.29. A simple polyhedron in a scene is regarded as an instance of a transformed primitive, where a transform may involve scaling along the three coordinate axes, translation, and rotation. Compound polyhedra, such as Fig. 9.28a, are regarded as simple polyhedra "glued together." (A cell-decomposition representation is thus used for compound polyhedra.) The program is first to derive from the scene the identity of the primitive objects used to construct it (including details of the construction of compound polyhedra). Next, it is to discover the transformations applied to the primitives to obtain the particular incarnations making up the scene. Finally, to demonstrate its understanding, it should be able to construct a line drawing of the scene from any viewpoint, using its derived description.

To understand a part of the scene, the program first decides which primitive it comes from, and then derives the transformation the primitive underwent to appear as it does in the scene. Identifying primitives is done by matching "topological" features of the line drawing (configurations of faces, lines, and vertices) with those of the model primitives; matching features induce a match between scene and model points. At least four noncoplanar matching points are needed to derive

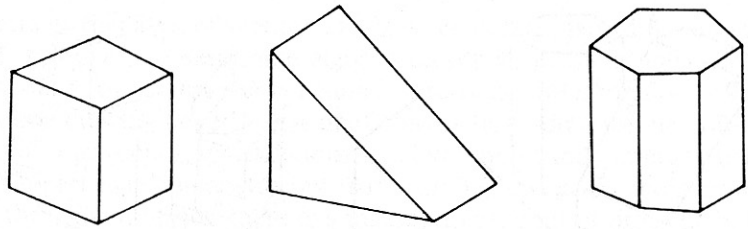


**Fig. 9.29** Primitive objects for scene construction.

a transformation. Tentative topological matches are checked by a metrical process which determines whether a primitive can allowably be transformed into the required shape, and if so whether it lies completely inside the observed polyhedra. Since the same image can result from a close small scene or a distant large one, assumptions about the location of the supporting surface and how objects are placed on it are used to fix the distance.

The three primitives all have convex polygons as faces, which project onto the line drawing as convex polygons. The faces all have three, four, or six sides, so faces that have not suffered occlusion or merging with another face while forming a compound polyhedron appear convex, have three, four, or six sides, and have no sides that are the uprights of "T vertices" (which result from occlusion). Polygons that pass these three tests are "approved" and are remembered on a list of possible primitive faces (Fig. 9.30).

In searching for points to identify between the scene and the primitives, the program looks for topological structures (Fig. 9.31) in decreasing order of efficacy, extracts the highest-quality information, reinterprets the scene, and searches again.

When a transformed primitive is identified in the scene, it is notionally unglued and removed, the resulting new visible lines are filled in, and the new scene is analyzed. Roberts's algorithm is not infallible, but it was pioneering work and is a sound starting point for the study of polyhedral scene analysis.

### 9.5.2 Grouping Regions into Bodies

A program by Guzman [Guzman 1969] takes as input a drawing of a polyhedral scene which may be quite complicated (Fig. 9.28b). The lines divide the drawing into a number of polygonal regions, and the goal of the program is just to group these regions into sets, each set corresponding to one polyhedral "block." Any reasonable description of an ambiguous scene is satisfactory. One could say that Guzman was addressing a polyhedral version of the general question of how human beings segment the world into objects.

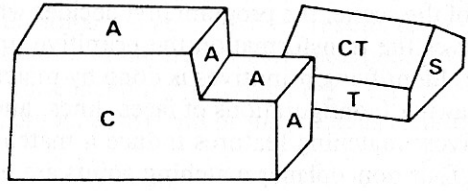



**Fig. 9.30** Approved and nonapproved polygons in a line drawing: A: approved; C: Concave; T: T-joint; S: Wrong number of sides.

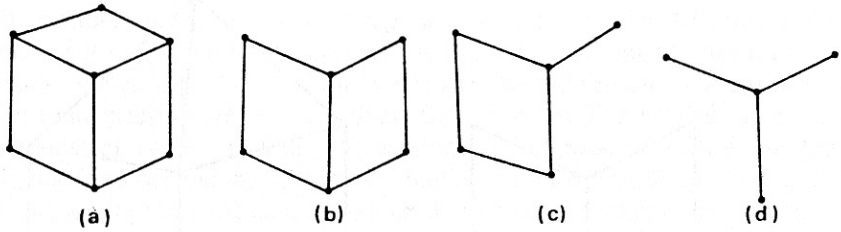



**Fig. 9.31** Topological match structures of Roberts.

The idea once again is to accumulate local evidence from the scene, and then to group polygons on the basis of this evidence. The evidence takes the form of "links" which link two regions if they may belong to the same body; links are planted around vertices, which are classified into types, each type always planting the same links (Fig. 9.32). No links are made with the background region.

Scenes are interpreted by grouping according to regions/links, using fairly complex rules, including "inhibitory links" that preclude two neighboring regions from being in the same body.

The final form of the program performs reasonably well on scenes without accidents of visual alignment, but it is a maze of special cases and exceptions, and seems to shed little light on what is going on in known polyhedral line-drawing perception. One might well ask where the links come from; no justification of why they are correct is given. Further ([Mackworth 1973]), Guzman can accept as one body the two regions in Fig. 9.33a. Finally, one feels a little dissatisfied with a scheme that just answers "one body" to a scene like Fig. 9.33b, instead of answering "pyramid on cube" or "two wedges," for example.

Guzman's method is correct for a world of convex isolated trihedral polyhedra: it is extended by ad hoc adjustments based on various potentially conflicting items of evidence from the line drawing. Ultimately it performs adequately with a much increased range of scenes, albeit not very elegantly. Further progress in the line drawing domain came about when attention was directed at the three-dimensional causes of the different vertex types.

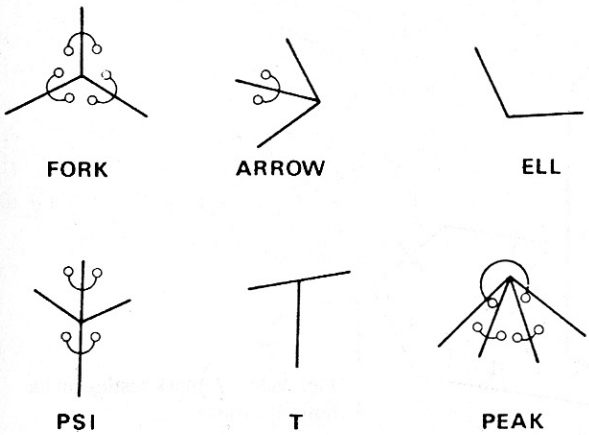



**Fig. 9.32** Links around vertices.

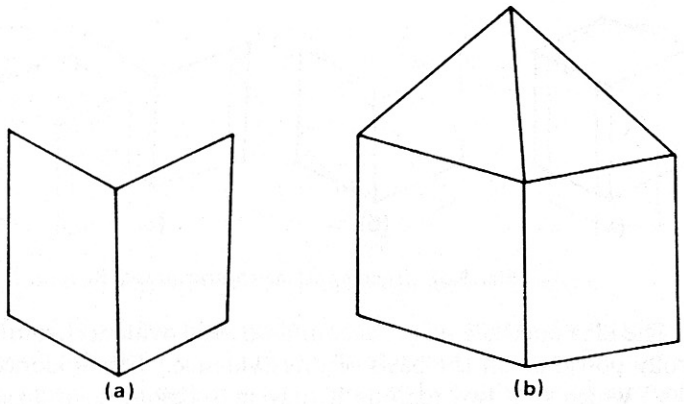



**Fig. 9.33** (a) Non-polyhedral scene. (b) Two wedges or a pyramid on cube.

### 9.5.3 Labeling Lines

Huffman and Clowes independently concerned themselves with scenes similar to Guzman's, not excluding non-simply connected polyhedra, but excluding accidents of alignment [Huffman 1971; Clowes 1971]. They desired to say more about the scene than just which regions arose from single bodies; they wanted to ascribe interpretations to the lines. Figure 9.34 shows a cube resting on the floor; lines labeled with a + are caused by a convex edge, those labeled with a − are caused by a concave edge, and those labeled with a > are caused by matter occluding a surface behind it. The occluding matter is to the right of the line looking in the direction of the >, the occluded surface is to the left. If the cube were floating, one would label the lowest lines with < instead of with −. The shadow line labels (arrows) were not used by Huffman.

A systematic investigation can find the types of lines possibly seen around a trihedral corner; such corners can be classified by how many octants of space are filled by matter around them (one for the corner of a cube, seven for the inside corner of a room, etc.). By considering all possible trihedral corners as seen from

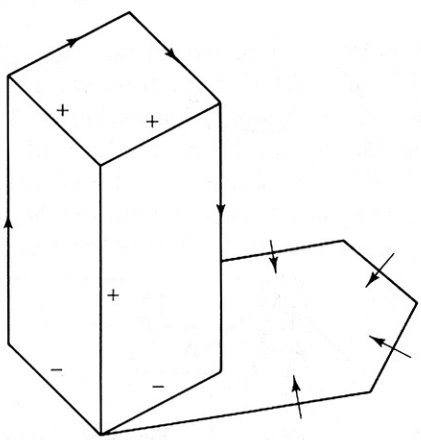



**Fig. 9.34** A block resting on its bottom surface.

all possible viewpoints, Huffman and Clowes found that without occlusion, just four vertex types and only a few of the possible labelings of lines meeting at a vertex can occur. Figure 9.35 shows views of one- and three-octant corners which give rise to all possible vertices for these corner types. The vertices appear in the first two rows of Table 9.1, which is a catalogue of all possible vertices, including those arising from occlusion, in this restricted world of trihedral polyhedra. It is easy to imagine extending the catalog to include vertices for other corner types.

It is important to note that there are four possible labels for each line ($+ - > <$), and thus $4^3 = 64$ possible labels for the fork, arrow, and T and 16 possible labels for the ell. In the catalog, however, only 3/64, 3/64, 4/64, and 6/16, respectively, of the possible labels actually occur. Thus only a small fraction of possible labels can occur in a scene.

The main observation that lets line-labeling analysis work is the coherence rule: In a real polyhedral scene, *no line may change its interpretation (label) between vertices.* For example, what is wrong with scenes like Fig. 9.36 is that they cannot be coherently labeled; lines change their interpretation within the impossible object. Perhaps the lines in drawings of real scenes can be interpreted quickly because the small percentage of meaningful labelings interacts with the coherence rule to reduce drastically the number of explanations for the scene.

How does line labeling relate to Guzman? A labeled-line description clearly indicates the grouping of regions into bodies, and also rejects scenes like Fig. 9.33a, which cannot be coherently labeled with labels from the catalog. The origin of Guzman's links can be explained this way: consider again the world of convex polyhedra; the only labels from the catalog that are possible are shown in Fig. 9.37a. Further, it is clear that a convex edge has two faces of the same body on either side of it, and an occluding edge has faces from two different bodies on either side of it. A convex label means the regions on either side of it should be linked; this is Guzman's link-planting rule (Fig. 9.37b). The inhibition rules are a further corollary of the labels; they are to suppress links across an edge if evidence that it

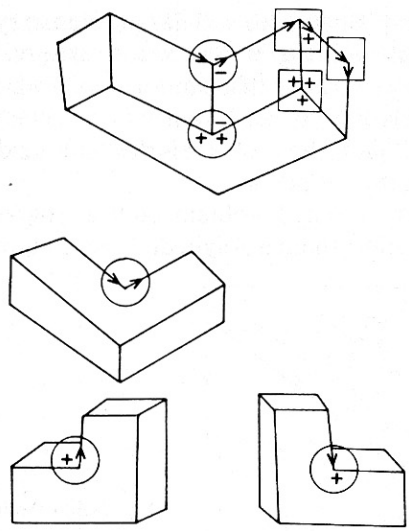
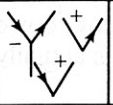


**Fig. 9.35** Different views of various corner types.

**Table 9.1**
**VERTEX CATALOGUE**

| Visible surfaces / Octants filed | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| 1 |  |  |  | — |
| 3 |  |  |  | — |
| 5 |  |  | — | — |
| 7 |  | — | — | — |
| Occlusion |  |  |  |  |

must be occluding is supplied by the vertex at its other end (Fig. 9.37c). When vertices at both ends of a line agree that the line is convex, Guzman would have planted two links; this is in fact the strongest evidence that the regions are part of the same body. If just one vertex gives evidence that the edge has a link, a decision based on heuristics is made; the coherence rule is being used implicitly by Guzman. The same physical and geometric reality is driving both his scheme and that of Huffman.

The labeling scheme explained here still has problems: syntactically nonsensical scenes are coherently labeled (Fig. 9.38a); scenes are given geometrically impossible labels (Fig. 9.38b); and scenes that cannot arise from polyhedra are easily labelled (Fig. 9.38c). It is very hard to see how a labeling scheme can detect the illegality of scenes like (Fig. 9.38c); the problem is not that the edges are incorrectly labeled, but that the faces cannot be planar.

Concern with this last-mentioned problem led to a program (see the next section) that can obtain information about a polyhedral scene equivalent to labeling it,

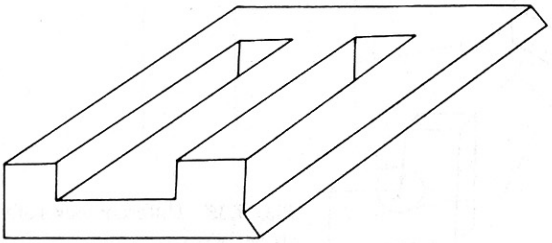


**Fig. 9.36** An impossible object.

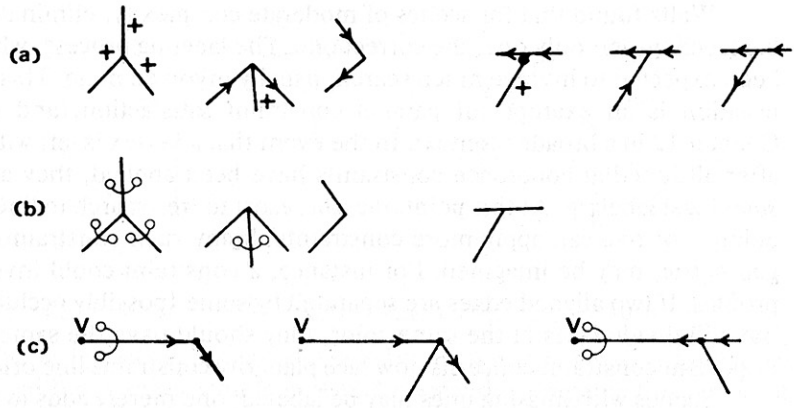



**Fig. 9.37** The relation of links to labels. (a) Line labels. (b) Link planting vertices. (c) Inhibitory links.

and also can reject non-polyhedra as impossible. There has also been an exciting denouement to the line-labeling idea [Waltz 1975; Turner 1974].

Waltz extends the line labels to include shadows, three illumination codes for each face on the side of an edge, and the separability of bodies in the scene at cracks and concave edges; this brings the number of line labels possible up to just below 100. He also extends the possible vertex types, so that many vertices of four lines occur. He can deal with scenes such as the one shown in Fig. 9.28c.

The combinatorial consequence of these extensions is clear; the possible vertex labelings multiply enormously. The first interesting thing Waltz discovered was that despite the combinatorics, as more information is coded into the lines, the smaller becomes the percentage of geometrically meaningful labels for a vertex. In his final version, only approximately 0.03 percent of the possible arrow labels can occur, and for some vertices the percentage is approximately 0.000001.

The second interesting thing Waltz did was to use a constraint-propagating labeling algorithm which very quickly eliminates labels for a vertex that is impossible given the neighboring vertices and the coherence rule, which places *constraints* on labelings. The small number of meaningful labels for a vertex imposes severe constraints on the labeling of neighboring vertices. By the coherence rule, the constraints may be passed around the scene from each vertex to its neighbors; eliminating a label for a vertex may render neighboring labels illegal as well, and so on recursively.

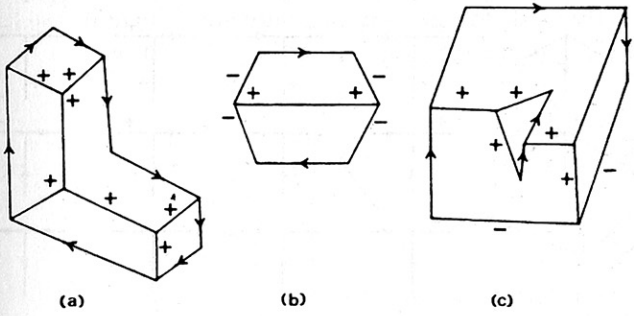



**Fig. 9.38** Nonsense labelings and nonpolyhedra.

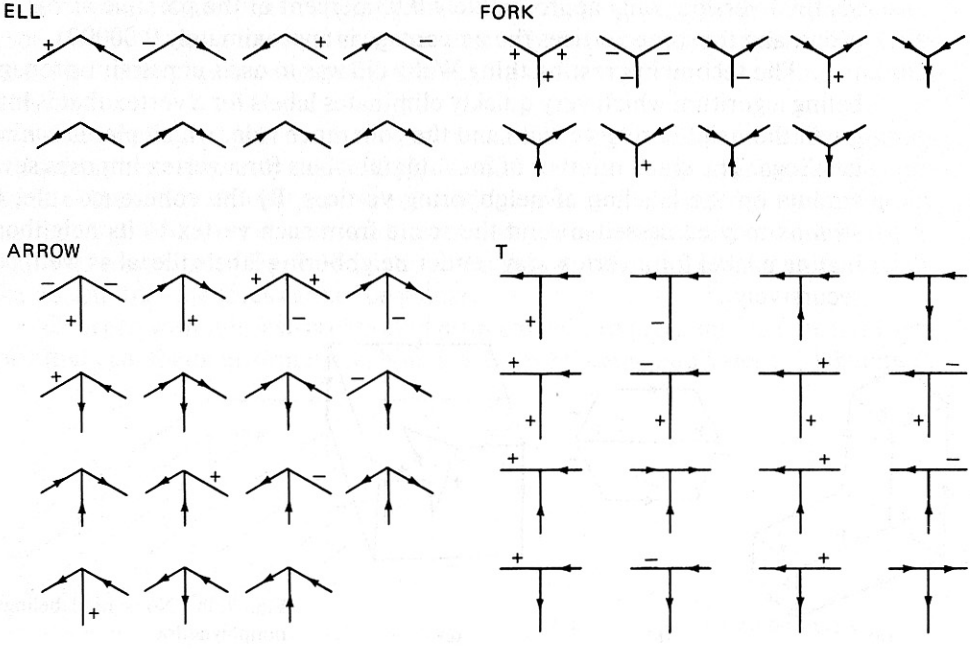
Waltz found that for scenes of moderate complexity, eliminating all impossible labelings left only one, the correct one. The labeling process, which might have been expected to involve much search, usually involved none. This constraint propagation is an example of parallel constraint satisfaction, and is discussed in Chapter 12 in a broader context. In the event that a vertex is left with several labels after all junction coherence constraints have been applied, they all participate in *some* legal labeling. At this point one can resort to tree search to find the explicit labelings, or one can apply more constraints. Many such constraints, heuristic and geometric, may be imagined. For instance, a constraint could involve color edge profiles. If two aligned edges are separated by some (possibly occluding) structure, but still divide faces of the same color, they should have the same label. Another important constraint concerns how face planarity constrains line orientations.

Scenes with missing lines may be labeled; one merely adds to the legal vertex catalog the vertices that result if lines are missing from legal vertices. This idea has the drawbacks of increasing the vertex catalog and widening the notion of consistency, but can be useful.

Another extension to line labeling is that of [Kanade 1978]. This extension considers not only solid polyhedra but objects (including nonclosed "shells") made up of planar faces. This extension has been called *origami world* after the art of making objects from folded (mostly planar) paper. An example from origami world is the box in Fig. 9.39a. A quick check shows that this cannot be labeled with the Huffman-Clowes label set. It can be labeled using the origami world label set (Table 9.2) and its interpretation is shown in Fig. 9.39b.

**Table 9.2**

**EXPANDED JUNCTION TABLE**



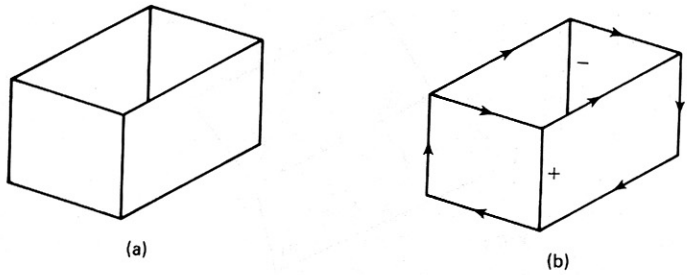



**Fig. 9.39** (a) Box. (b) Labeled edges according to origami world label set.

The vertex labels may be extended to include scenes with cylinders, cones, spheres, tori, and other simple curves. In expanded domains the notion of "legal line drawing" becomes very imprecise. In any event the number of vertex types and labels grow explosively, and the coherence rule must be modified to cope with the fact that lines can change their interpretation between vertices and can tail off into nothing, and that one region can attain all three of Waltz's illumination types [Turner 1974, Chakravarty 1979]. The domain is of scenes such as appear in Fig. 9.28d.

### 9.5.4 Reasoning About Planes

The deficiencies in the scene line-labeling algorithms prompted a consideration of the geometrical foundations of the junction labels [Mackworth 1973, Sugihara 1981]. This work seeks to answer the same sorts of questions as do labeling programs, but also to take account of objects that cannot possibly be planar polyhedra, such as those of Fig. 9.40. Neither approach uses a catalog of junction labels, but relies instead on ideas of geometric coherence. The basis is a plane-oriented formulation rather than a line-oriented one.

*Gradient Space*

Mackworth's program relies heavily on the relation of polyhedral surface gradients to the lines in the image (recall section 3.5.2). Image information from orthographic projections of planar polyhedral scenes may be related to gradient information in a useful way. An image line L is the projection of a three-space line $M$ arising from the intersection of two faces lying in distinct planes $\Pi_1$ and $\Pi_2$ of gradients $(p_1, q_1)$ and $(p_2, q_2)$. With the $(p, q)$ coordinate system superimposed on the image $(x, y)$ coordinate system, there is the following constraint. The orientation of $L$ constrains the gradients of $\Pi_1$ and $\Pi_2$; specifically, the line $L$ is perpendicular to the line $G$ between $(p_1, q_1)$ and $(p_2, q_2)$ (Fig. 9.41).

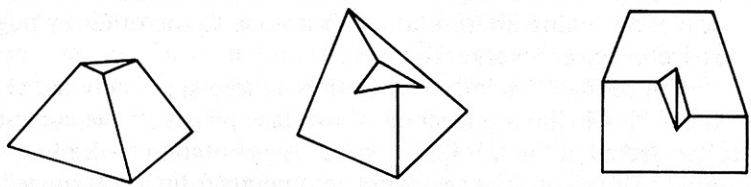


**Fig. 9.40** Labelable but not planar polyhedra.

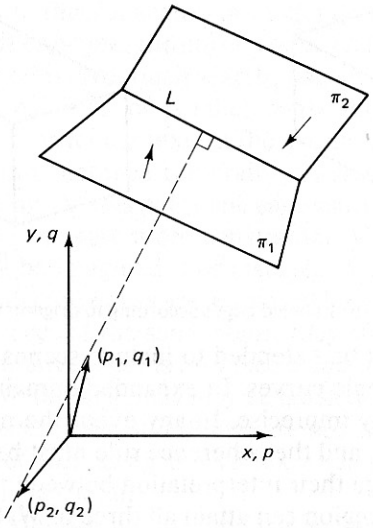



**Fig. 9.41** Gradient space constraint.

The result is easily shown. With orthographic projection, the origin may be moved of the image plane to be in $L$ without loss of generality. Then $L$ is defined by its direction vector $(\lambda, \mu) = (\cos\theta, \sin\theta)$. The three-space point on $\Pi_1$ corresponding to $(0,0)$ may be expressed as $(0,0,k_1)$, and at $(\lambda, \mu)$ the corresponding point is $(\lambda, \mu, \lambda p_1 + \mu q_1 + k_1)$. Thus moving along $M$ (which is in $\Pi_1$) from $(x, y) = (0,0)$ to $(x, y) - (\lambda, \mu)$ moves along $-z$ by $\lambda p_1 + \mu q_1$. The coordinates of a unit vector on $L$ can then be expressed as $(\lambda, \mu, \lambda p_1 + \mu q_1)$. But $L$ is also in $\Pi_2$, and this argument may be repeated for $\Pi_2$, using $p_2$ and $q_2$. Thus

$$\lambda p_1 + \mu q_1 = \lambda p_2 + \mu q_2 \tag{9.23}$$

or

$$(\lambda, \mu) \cdot (p_2 - p_1,\ q_2 - q_1) = 0 \tag{9.24}$$

Equation (9.24) is a dot product set equal to zero, showing that its two vector operands are orthogonal, which was to be shown.

Every picture line results from the intersection of two planes, and so it has a line associated with it in gradient space which is perpendicular to it. Furthermore, if the gradients of the surfaces are on the same side of the picture line as their surfaces, the edge was convex; if the gradients are on opposite sides of the line from their causing surfaces, the edge was concave (Fig. 9.42). For every junction in the image there are just two ways the gradients can be arranged to satisfy the perpendicularity requirement (Fig. 9.43). In the first, all edges are convex, in the second, concave. Switching interpretations from one to the other by negating gradients is the psychological "Necker reversal."

Notice that if an image junction is a three-space polyhedral vertex, each edge of the vertex is the intersection of two face planes. If the corresponding gradients are connected, a "dual" $(p, q)$ space representation of the $(x, y)$ space junction is formed. The connected $(p, q)$ gradient points form a polygon whose edges are perpendicular to the junction lines in $(x, y)$ space. The polygon is larger if the three-

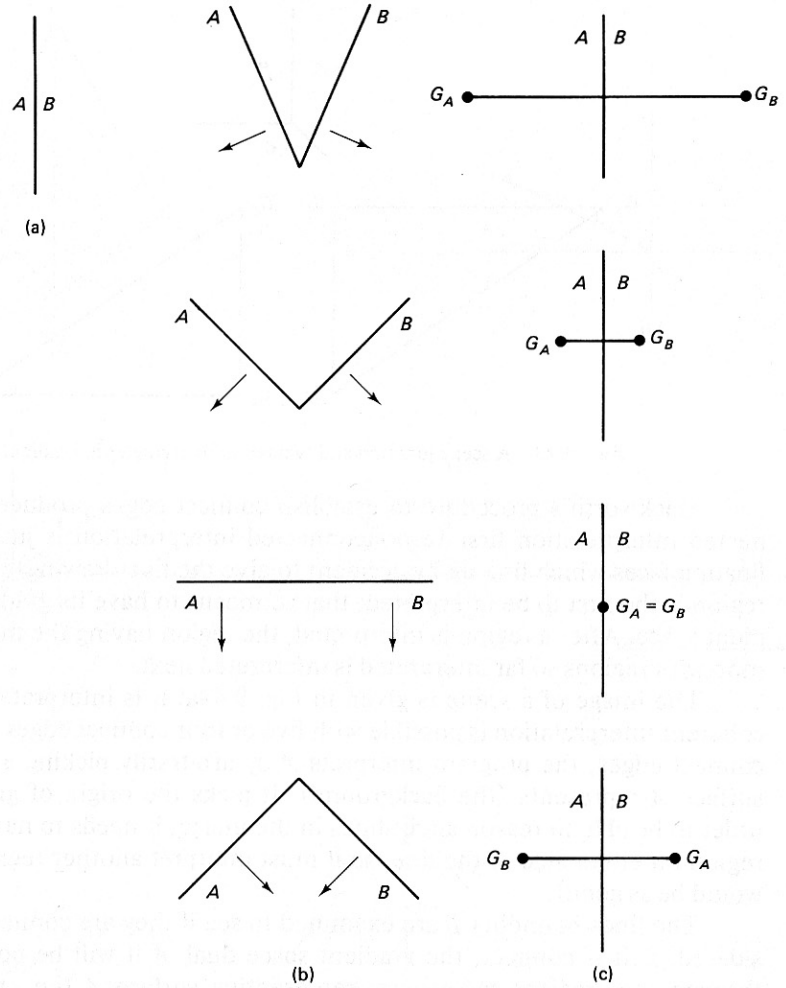



**Fig. 9.42** Relation of gradients, image and world structures. (a) Image. (b) World. (c) Gradients.

dimensional corner is sharper, and shrinks toward the junction point as the corner gets blunter.

*Interpreting Drawings*

It is possible to use these geometric results to interpret the lines in orthogonally projected polyhedral scenes as being "connect" (i.e., as being between two connected faces) or occluding. It can also be determined if connect edges are convex or concave, and for occluding edges which surface is in front. Hidden parts of the scene may sometimes be reconstructed. The orientation of each surface and edge in the scene may be found. Thus a program can determine that input such as Fig. 9.40 is not a planar-faced polyhedron [Mackworth 1973]. Sugihara's work generalizes Mackworth's; it does not use gradient space and does not rely on orthographic projection.

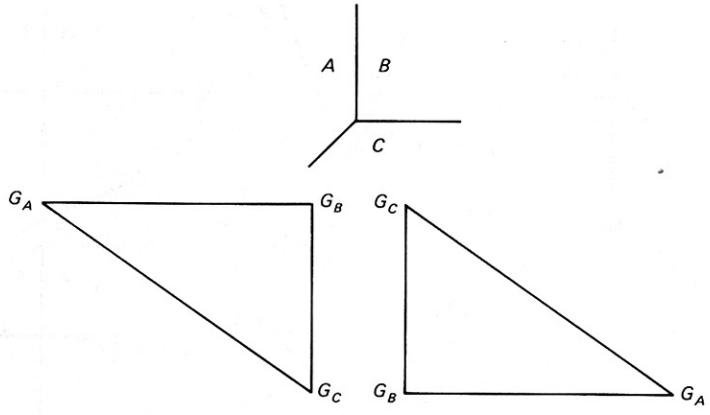



**Fig. 9.43** A scene junction and two resulting triangles in gradient space.

Mackworth's procedure to establish connect edges produces the most connected interpretation first (a nonconnected interpretation is just a collection of floating faces which line up by accident to give the line drawing). The background region is the first to be interpreted; that is, means to have its gradient fixed in gradient space. After a region is interpreted, the region having the most lines in common with regions so far interpreted is interpreted next.

The image of a scene is given in Fig. 9.44a; it is interpreted as follows. No coherent interpretation is possible with five or four connect edges. Trying for three connect edges, the program interprets $A$ by arbitrarily picking a gradient for the surface $A$ represents (the background). It picks the origin of gradient space. In order to be able to reason about lines in the image, it needs to have an interpreted region on either side of the line, so it must interpret another region. It picks $B$ ($C$ would be as good).

The lines bounding $B$ are examined to see if they are connect. Line 1 is considered. If it is connect, the gradient space dual of it will be perpendicular to it through the gradient space point representing surface $A$ (i.e., the origin). Now another arbitrary choice: The gradient corresponding to surface $B$ is placed at unit distance from the origin, thus "imagining" the second gradient in a row. From now on, the gradients are more strongly located. The arbitrary scaling and point of origin imposed by these first two choices can be changed later if that is important.

In gradient space, the situation is now shown in Fig. 9.44b. Now consider line 2; to establish it as a connect edge, $\boldsymbol{G}_B = (p_B, 1_B)$ (the gradient space point corresponding to the surface B) must lie on a line perpendicular to 2 through $\boldsymbol{G}_A$ (Fig. 9.44c). This cannot happen; the situation with 1 and 2 both connect is incoherent. Thus, with a line 1 connect edge, 2 must be occluding. This sort of incoherency result was what kept the program from finding four or five edges connect. Further interpretation involves assigning gradients and vertices into the developing diagram in a noncontradictory, maximally connected manner (Fig. 9.44d).

The next part of the program determines convexity or concavity of the lines. The final part of the program looks at occlusion. It also suggests hidden surfaces

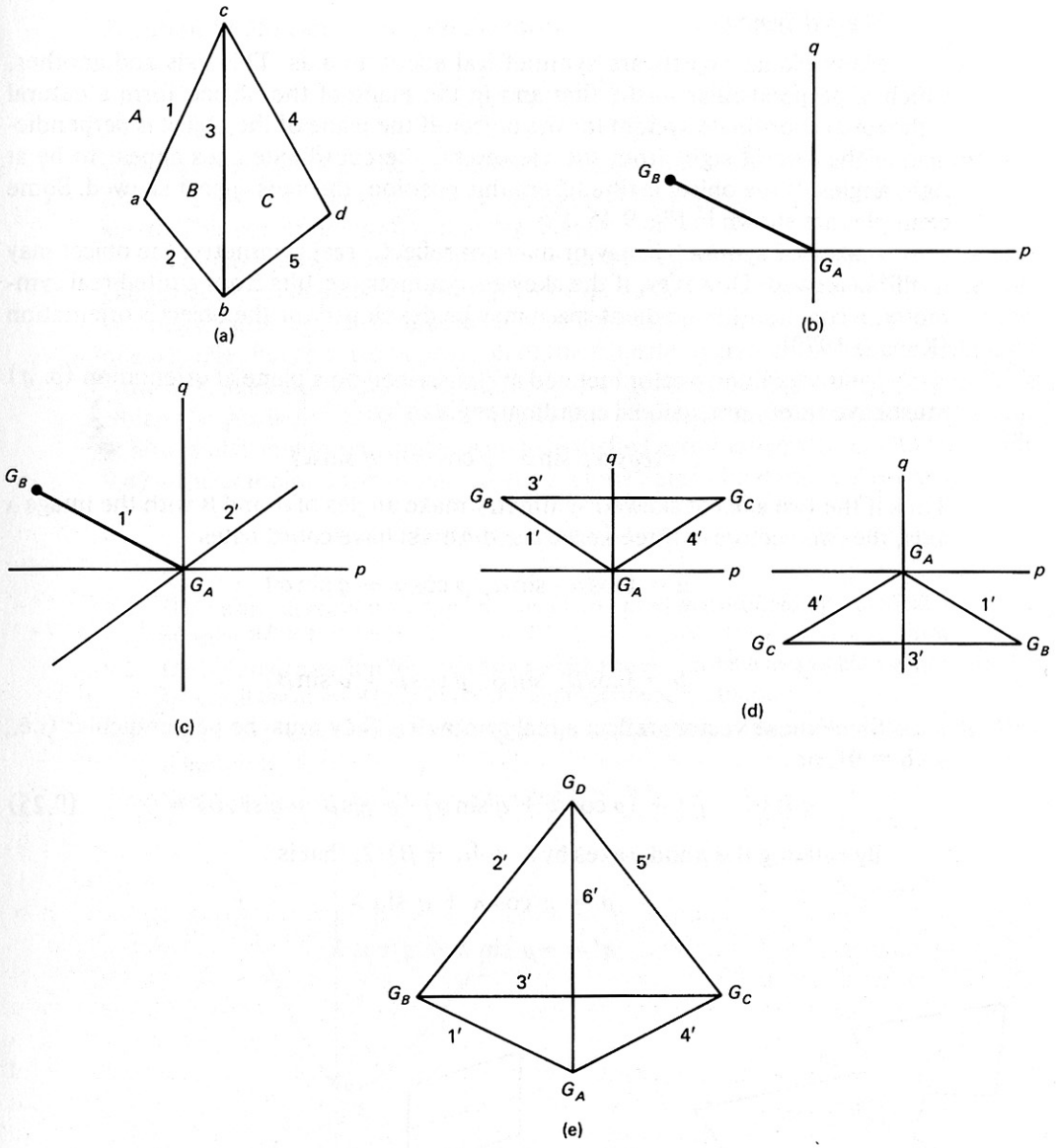



**Fig. 9.44** (a) Polyhedral scene considered by Mackworth. (b) Partial interpretation. (c) Continued interpretation. (d) Occluding and connect interpretations. (e) Final interpretation.

and thus hidden lines that are consistent with the interpretation (Fig. 9.44e). This figure in gradient space resembles a tetrahedron, as well it might; it is formed in the same way as the graph-theoretic dual (point per face, edge per edge, face per point) which defines dual graphs and dual polyhedra; the tetrahedron is self-dual. The arbitrary choices of gradient reflect degrees of freedom in the drawing that are also identified by Sugihara.

*Skewed Symmetry*

Many planar objects are symmetrical about an axis. This axis and another, which is perpendicular to the first and in the plane of the object, form a natural orthogonal coordinate system for the object. If the plane of the object is perpendicular to the line of sight from the viewpoint, the coordinate axes appear to be at right angles. If the object is tilted from this position, the axes appear skewed. Some examples are shown in Fig. 9.45.

A skewed symmetry may or may not reflect a real symmetry; the object may itself be skewed. However, if the skewed symmetry results from a tilted real symmetry, a constraint in gradient space may be developed for the object's orientation [Kanade 1979].

An imaged unit vector inclined at $\alpha$ inscribed on a plane at orientation $(p, q)$ must have three-dimensional coordinates given by

$$(\cos\alpha,\ \sin\alpha,\ p\cos\alpha + q\sin\alpha)$$

Thus if the two axes of skewed symmetry make angles of $\alpha$ and $\beta$ with the image $x$ axis, the two vectors in three-space $a$ and $b$ must have coordinates

$$\mathbf{a} = (\cos\alpha,\ \sin\alpha,\ p\cos\alpha + q\sin\alpha)$$

and

$$\mathbf{b} = (\cos\beta,\ \sin\beta,\ p\cos\beta + q\sin\beta)$$

Since these vectors reflect a real symmetry, they must be perpendicular (i.e., $\mathbf{a}\cdot\mathbf{b} = 0$), or

$$\cos(\alpha - \beta) + (p\cos\alpha + q\sin\alpha)\ (p\cos\beta + q\sin\beta) = 0 \qquad (9.25)$$

By rotating the $p$ and $q$ axes by $\lambda = (\alpha + \beta)/2$, that is

$$p' = p\ \cos\lambda + q\ \sin\lambda$$

$$q' = -p\ \sin\lambda + q\ \cos\lambda$$

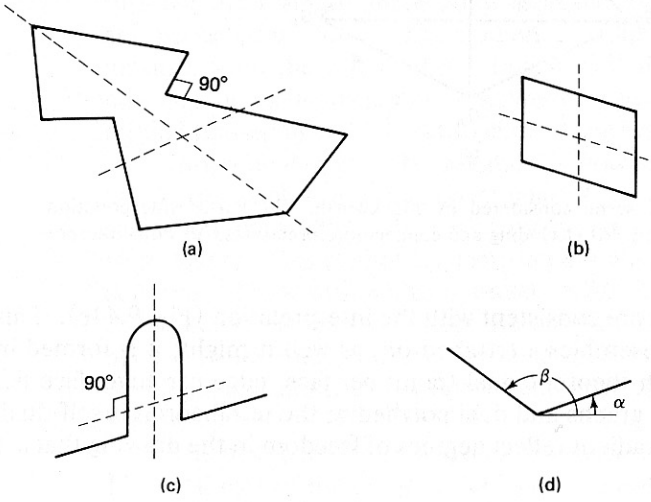



**Fig. 9.45** Skewed symmetries. (a,b,c) are examples. (d) Each skewed symmetry defines two axes.

Equation (9.25) can be put into the form

$$p'^2 \cos^2\left(\frac{\gamma}{2}\right) - q'^2 \sin^2\left(\frac{\gamma}{2}\right) = -\cos(\gamma)$$

where $\gamma = \alpha - \beta$. Thus the gradient of the object must lie on a hyperbola with axis tilted $\lambda$ from the $x$ axis, and with asymptotes perpendicular to the directions of $\alpha$ and $\beta$. This constraint is shown in Fig. 9.46.

To show how skewed symmetry can be exploited to interpret objects with planar faces, reconsider the example of Fig. 9.43. In that example the three convex edges constrained the gradients of the corresponding faces to be at the vertices of a triangle, but the size or position of the triangle in gradient space was unknown. However, skewed symmetry applied to each face introduces three hyperbola upon which the gradients must lie. The only way that both the skewed symmetry constraint and triangle constraint can be satisfied simultaneously is shown in Fig. 9.47—the combined constraints have uniquely determined the face orientations.

## EXERCISES

**9.1** Derive an expression for the volume of an object represented by spherical harmonics of order $M = 1$.

**9.2** Derive an expression for the perpendicular to the surface of an object represented by spherical harmonics in terms of the appropriate derivatives.

**9.3** Derive an expression for the angle centroid of each of the spherical harmonic functions for $M \leqslant 2$.

**9.4** Label the lines in the objects of Fig. 9.48.

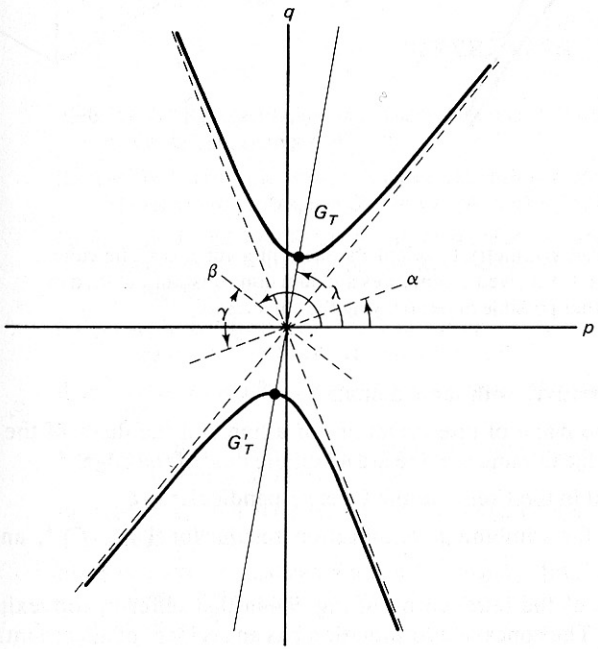



**Fig. 9.46** Skewed symmetry constraint in gradient space.

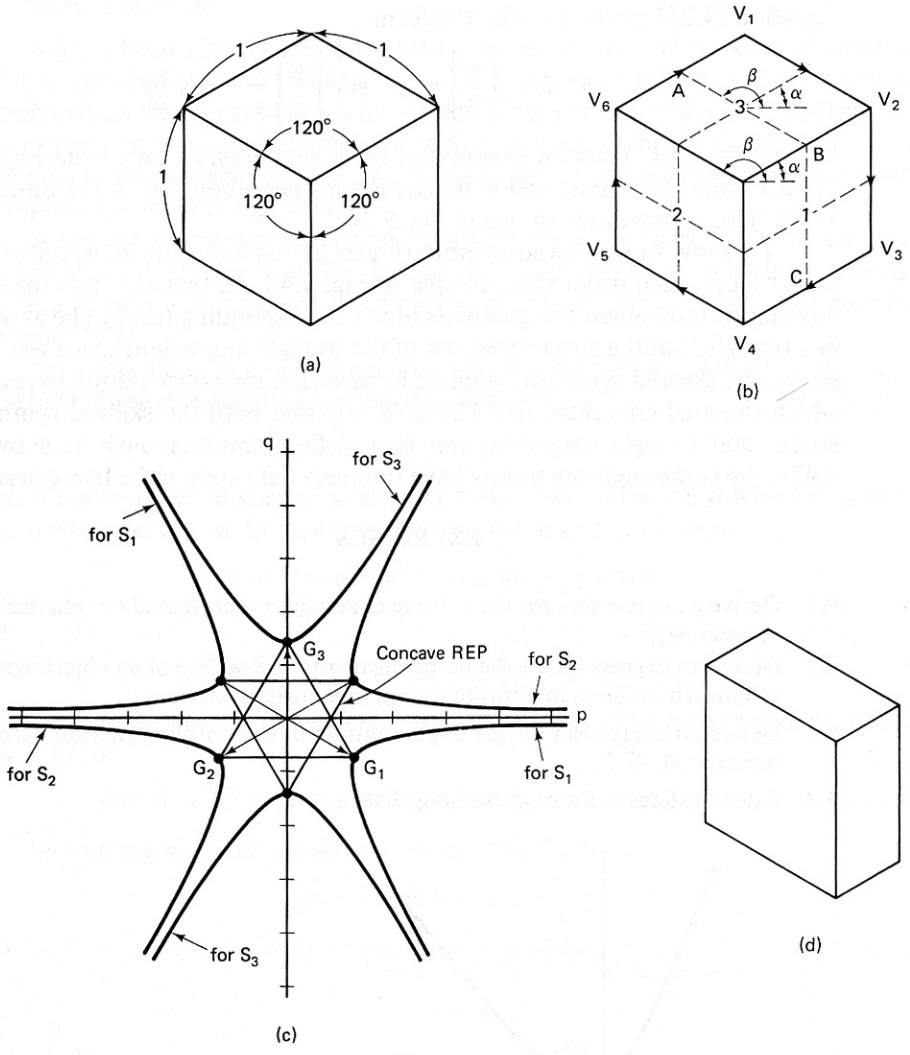



**Fig. 9.47** Using skewed symmetry to orient the faces of a cube. (a) The cube. (b) Skewed symmetries. (c) skewed symmetries and junction constraint plotted in gradient space. (d) another possible object obeying the constraints.

**9.5** Give two sets of CSG primitives with same domain.

**9.6** Show that the dual of the plane of interpretation for a line and the duals of the two planes that meet in the edge causing the line are all on the dual of the edge.

**9.7** Prove (Section 9.3.1) that in the Frenet frame $\xi$ ' is perpendicular to $\xi$.

**9.8** Write the precise rules for combining classification results for $\cup^*$, $\cap^*$, and $-$ operations.

**9.9** Find two interpretations of the tetrahedron of Fig. 9.44a that differ in convexity or concavity of lines. (Hint: The concave interpretation has an accident of alignment.)

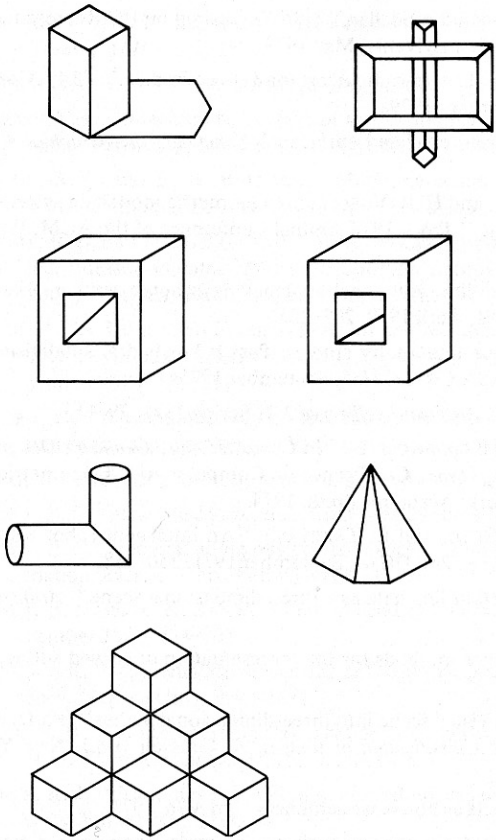


**Fig. 9.48** Objects for labeling.

## REFERENCES


AGIN, G. J. "Representation and description of curved objects" (Ph.D. dissertation). AIM-173, Stanford AI Lab, October 1972.

BADLER, N. I. and R. K. BAJCSY. "Three-dimensional representations for computer graphics and computer vision." *Computer Graphics 12*, August 1978, 153–160.

BADLER, N. I. and J. O'ROURKE. "Representation of articulable, quasi-rigid, three-dimensional objects." NSF Workshop on the Representation of Three-Dimensional Objects, Univ. Pennsylvania, May 1979.

BARNHILL, R. E. "Representation and approximation of surfaces." In *Mathematical Software III*, J. R. Rice (Ed.). New York: Academic Press, 1977.

BARNHILL, R. E. and R. F. RIESENFELD. *Computer Aided Geometric Design.* New York: Academic Press, 1974.

BAUMGART, B. G. "Winged edge polyhedron representation." STAN-CS-320, AIM-179, Stanford AI Lab, October 1972.

BENTLEY, J. L. Multidimensional search trees used for associative searching, *Comm. ACM 18*, 9, Sept. 1975, 509–517.

BINFORD, T. O. "Visual perception by computer." IEEE Conf. on Systems and Control, Miami, December 1971.

BOYSE, J. W. "Data structure for a solid modeller," NSF Workshop on the Representation of Three-Dimensional Objects, Univ. Pennsylvania, May 1979.

BROWN, C. M. "Two descriptions and a two-sample test for 3-d vector data." TR49, Computer Science Dept., Univ. Rochester, February 1979a.

BROWN, C. M. "Fast display of well-tesselated surfaces." *Computers and Graphics 4*, 2, September 1979b, 77–85.

BROWN C. M., A. A. G. REQUICHA, and H. B. VOELCKER. "Geometric modelling systems for mechanical design and manufacturing." *Proc.*, 1978 Annual Conference of the ACM, Washington, DC, December 1978, 770–778.

CHAKRAVARTY, I. "A generalized line and junction labelling scheme with applications to scene analysis," IEEE Trans. PAMI, April 1979, 202–205.

CLINTON, J. D. "Advanced structural geometry studies, Part I: Polyhedral subdivision concepts for structural applications." NASA CR-1734/35, September 1971.

CLOWES, M. B. "On seeing things." *Artificial Intelligence 2*, 1, Spring 1971, 79–116.

COONS, S. A. "Surface patches and B-spline curves." In *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld (Eds.). (*Proc.*, Conference on Computer Aided Geometric Design, Univ. Utah, March 1974.) New York: Academic Press, 1974.

EJIRI, M., T. UNO, H. YODA, T. GOTO, and K. TAKEYASU. "An intelligent robot with cognition and decision-making ability." *Proc.*, 2nd IJCAI, September 1971, 350–358.

FALK, G. "Interpretation of important line data as a three-dimensional scene." *Artificial Intelligence 3*, 1, Spring 1972, 77–100.

FORREST, A. R. "On cones and other methods for the representation of curved surfaces." *CGIP 1*, 4, December 1972, 341–359.

GUZMAN, A. "Decomposition of a visual scene into three-dimensional bodies" (Ph.D. dissertation). In *Automatic Interpretation and Classification of Images*, A. Grasseli (Ed.). New York: Academic Press, 1969.

HUFFMAN, D. A. "Impossible objects as nonsense sentences." In *MI6*, 1971.

JACKINS, C. L., and S. L. TANIMOTO. Oct-trees and their use in representing three-dimensional objects, *CGIP 14*, 3, Nov. 1980, 249–270.

KANADE, T. "A theory of Origami world." CMU-CS-78-144, Computer Science Dept., Carnegie-Mellon Univ., 1978.

KANADE, T. "Recovery of the three-dimensional shape of an object from a single view." CMU-CS-79-153, Computer Science Dept., Carnegie-Mellon Univ., October 1979.

LAKATOS, I. *Proofs and Refutations.* Cambridge, MA: Cambridge University Press, 1976.

LEE, Y. T. and A. A. G. REQUICHA. "Algorithms for computing the volume and other integral properties of solid objects." Tech. Memo 35, Production Automation Project, Univ. Rochester, Rochester NY, Feb. 1980.

MACKWORTH, A. K. "Interpreting pictures of polyhedral scenes." *Artificial Intelligence 4*, 2, June 1973, 121–137.

MACKWORTH, A. K. "On reading sketch maps." *Proc.*, 5th IJCAI, August 1977, 598–606.

MARKOWSKY, G. and M. A. WESLEY. "Fleshing out wire frames." *IBM J. Res. Devel. 24*, 1 (Jan. 1980) 64–74.

MARR, D. and H. K. NISHIHARA. "Representation and recognition of the spatial organization of three-dimensional shapes." *Proc., Royal Society of London B 200*, 1978, 269–294.

NISHIHARA, H. K. "Intensity, visible surface and volumetric representations." NSF Workshop on the Representation of Three-Dimensional Objects, U. Pennsylvania, May 1979.

O'NEILL, B. *Elementary Differential Geometry.* New York: Academic Press, 1966.

O'ROURKE, J. and N. I. BADLER. "Decomposition of three-dimensional objects into spheres." *IEEE Trans. PAMI 1*, July 1979.

REQUICHA, A. A. G. "Representations of rigid solid objects." *Computer Surveys 12*, 4, December 1980.

ROBERTS, L. G. "Machine perception of three-dimensional solids." In *Optical and Electro-optical Information Processing*, J.P. Tippett et al. (Eds.). Cambridge, MA: MIT Press, 1965.

SCHUDY, R. B. and D. H. BALLARD. "Model-detection of cardiac chambers in ultrasound images." TR12, Computer Science Dept., Univ. Rochester, November 1978.

SCHUDY, R. B. and D. H. BALLARD. "Towards an anatomical model of heart motion as seen in 4-d cardiac ultrasound data." *Proc.*, 6th Conf. on Computer Applications in Radiology and Computer-Aided Analysis of Radiological Images, June 1979.

SHANI, U. "A 3-d model-driven system for the recognition of abdominal anatomy from CT scans." TR77, Computer Science Dept., U. Rochester, May 1980; also in Proc. 5th IJCPR, Miami, December 1980, 585–591.

SHAPIRA, R. "A technique for the reconstruction of a straight-edge, wire-frame object from two or more central projections." *CGIP 3*, 4, December 1974, 318–326.

SHIRAI, Y. "Analyzing intensity arrays using knowledge about scenes." In *PCV*, 1975.

SOROKA, B. I. "Generalised cylinders from parallel slices." *Proc.*, PRIP, 1979a, 421–426.

SOROKA, B. I. "Understanding objects from slices." Ph.D. dissertation, Dept. of Computer and Information Science, Univ. Pennsylvania, 1979b.

SOROKA, B. I. and R. K. BAJCSY. "Generalized cylinders from serial sections." *Proc.*, 3rd IJCPR, November 1976, 734–735.

SUGIHARA, K. "Mathematical structures of line drawings of polyhedra," RNS 81-02, Dept. of Info. Science, Nagoya Univ., May 1981.

TILOVE, R. B. "Set membership classification: a unified approach to geometric intersection problems." *IEEE Trans. Computers 29*, 10, October 1980.

TURNER, K. J. "Computer perception of curved objects using a television camera." Ph.D. dissertation, Univ. Edinburgh, 1974.

VOELCKER, H. B. and A. A. G. REQUICHA, Boundary evaluation procedures for objects defined via constructive solid geometry, Tech. Memo 26, Production Automation Project, Univ. Rochester, 1981.

VOELCKER, H. B. and A. A. G. REQUICHA. "Geometric modeling of mechanical parts and processes." *Computer 10*, December 1977, 48–57.

VOELCKER, H. B. and Staff of Production Automation Project, "The PADL-1.0/2 system for defining and displaying solid objects." *Computer Graphics 12*, 3, August 1978, 257–263.

WALTZ, D. I. "Generating semantic descriptions from drawings of scenes with shadows." Ph.D. dissertation, AI Lab, MIT, 1972; also in *PCV*, 1975.

WARNOCK, J. G. "A hidden-surface algorithm for computer-generated halftone pictures." TR 4-15, Computer Science Dept., Univ. Utah, June 1969.

WESLEY, M. A. and S. MARKOWSKY. "Fleshing out projections." *IBM J. Res. Devel. 25*, 6 (Nov. 1981), 934–954.