3

SIMPLE POLYHEDRAL SCENES

The pattern classification techniques described in the previous chapter have been applied primarily to the recognition of images of two-dimensional objects. The world of our everyday perception is, of course, three-dimensional. Perceiving a three-dimensional scene from a single point of view, which gives a two-dimensional image, adds some new and unique difficulties. Interpretation of these two-dimensional images is inherently ambiguous; the same image can be formed by an infinite number of three-dimensional scenes. Also, the image formed by a particular object changes with the viewing angle; this is also known as perspective change. Above all, in scenes with multiple objects, parts of otherwise visible surfaces of some objects may be occluded by others. A perceptual system needs to separate the objects in the image and recognize them from the partial information.

3.1 PERCEPTION OF SIMPLE POLYHEDRAL SOLIDS

The study of machine perception of three-dimensional objects was launched by the classic work of L. G. Roberts [1]. In his work, and much of the early work in three-dimensional scene analysis, the scenes were restricted to consist of polyhedral solids with homogeneous surfaces against uniform backgrounds. Such scenes can be adequately characterized by the intersection lines of the objects. His work contains

many concepts that extend beyond the simple scenes considered, and a detailed study of these methods is appropriate.

Consider an image of a simple cube, painted uniformly white against a dark background. We may consider recognition of the cube by the methods described in Chapter 2. As the observed picture changes with the viewing angle, simple geometrical properties are not invariant. A template-matching process could, in principle, be used. We would need to store (or generate) templates for each known object from different viewing angles and different viewing distances. Matching with such a large set of templates is clearly prohibitive in computational cost and still does not account for changes in the lighting conditions.

Instead of operating on the picture directly, it may be simpler to extract a line drawing from the picture, corresponding to the intersection of the planar surfaces of the three-dimensional object, and to attempt recognition from the line drawing. There is evidence that boundaries suffice for many perceptual tasks in human perception [2]. Certainly, for polyhedral objects, the boundaries directly determine the visible faces. Ideally, the boundaries shown in Fig. 3-1 would be extracted from the image of a cube and are adequate to distinguish it from objects with boundaries shown in Fig. 3-2, for example. (Of course, since the projection of three-dimensional scenes is inherently ambiguous, the distinction between the objects is limited.)

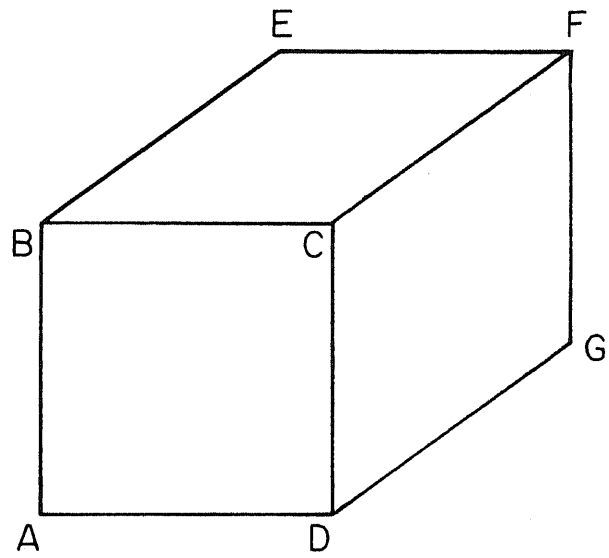


Figure 3-1: Outline of a cube

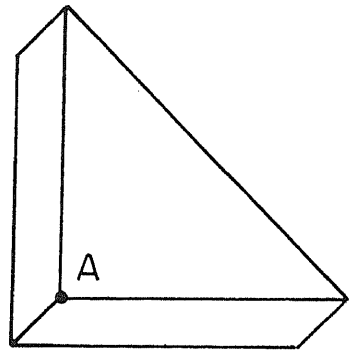


Figure 3-2: Outline of a wedge

3.1.1 Extraction of Line Drawings

Extraction of lines that correspond to object edges is based on the assumption that the light intensity is constant or smoothly varying over the image of an object face and jumps discontinuously at the intersection with the image of another face. This assumption is valid if the object surfaces are smooth, homogeneous, and opaque and the lighting is uniform and arranged to eliminate shadows. It is also assumed that the object surfaces do not have mirrorlike reflections.

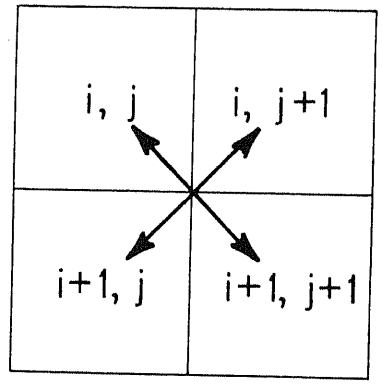


Figure 3-3: Roberts' gradient operator

In a continuous image plane, points at which the intensity changes discontinuously are easily identified to be those where the gradient of the intensity function is infinite (or larger than a threshold). An approximation for this gradient for a digital picture is given by

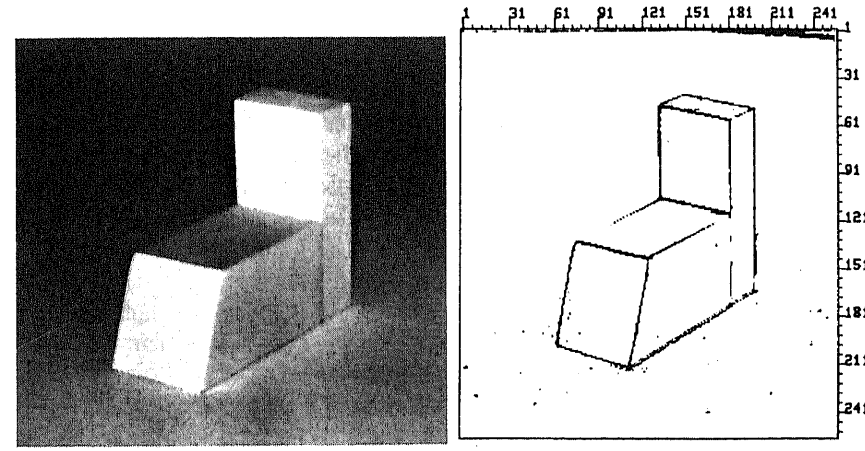
$$R(i, j) \simeq \nabla g(i, j)$$

$$= \sqrt{\{g(i+1, j+1) - g(i, j)\}^2 + \{g(i, j+1) - g(i+1, j)\}^2} \qquad (3\text{-}1)$$

where $g(i, j)$ is the image intensity at pixel (i, j) (see Fig. 3-3). Absolute values may be used instead of squared values in Eq. (3-1) above. Also the direction of the gradient is given by the angle α, where

$$\alpha = -\frac{\pi}{4} + \tan^{-1}\left[\frac{g(i, j+1) - g(i+1, j)}{g(i+1, j+1) - g(i, j)}\right] \qquad (3\text{-}2)$$

The above definition of gradient is due to Roberts [1], and this operator is often called Roberts' cross-operator. An *edge* is said to be present at pixel (i, j) if $R(i, j) > \tau$, where τ is a chosen threshold. If pictures were noise-free, τ could be chosen to be 0. In the presence of noise, τ is chosen by a trade-off between obtaining all desired edges and picking too many noise edges. Figure 3-4(a) shows the image of a complex block and Fig. 3-4(b) the edges detected in it by using the above-described method. (The threshold was chosen interactively for best subjective performance.) In this example, the block is carefully painted and has smooth surfaces. Any markings on the surface or the texture of the material would, of course, also show in these cases.



(a) (b)

Figure 3-4: (a) An image and (b) edges detected in it

The next step is to connect the computed edge points in straight

lines and determine the vertices from their intersections. Each edge is specified by a position as well as a direction (normal to that of the gradient at the point) and should be connected only in a line in that direction. Spurious noise points can be removed by eliminating short line segments. Gaps in lines caused by missing edges are bridged by extension up to predetermined lengths. Roberts was able to obtain "perfect" line drawing as in Fig. 3-1, for a limited class of scenes. However, such line detection has proven to be difficult for real scenes. More sophisticated techniques of determining object boundaries are given in Chapter 7. For the remainder of this chapter, it is assumed that perfect boundaries are available for further processing.

3.1.2 Model Matching

Once a line drawing is obtained, recognition can be achieved by determining which of the models can generate, under some permissible transformation, a line drawing that is most similar. The two line drawings must match *topologically* (or structurally)—that is, in the number of lines and vertices and their interconnections. The distances between vertices should also be as predicted by the model transformation.

For a topological match between two line drawings, it is useful to extract *polygons* in them. Further, the polygons corresponding to an object face should be distinguished from others, if possible (for example, in Fig. 3-1, only polygons *ABCD*, *BEFC*, and *CFGD* are desired and not *ABEFGD*, *ABEFCD*, *ABCFGD*, or *BCDGFE*). A clever algorithm to separate interior and exterior polygons was developed by Roberts. As an example, consider starting from an arbitrarily chosen line in a chosen direction, say from *A* to *B* along line *AB*. When vertex *B* is reached, we choose the line making the largest exterior clockwise angle with *AB* (that is, *BC*). If this procedure is repeated, polygon *ABCD* will be traced. Starting along *AB* in the other direction from *B* to *A*, and repeating the same steps, polygon *ADGFEB* is traced. However, this time the traversal of the polygon is in a counterclockwise direction (this can be determined from the scan of the exterior angles of the traversed polygons, $+2\pi$ for counterclockwise and -2π for clockwise traversal). If the above procedure is applied until each line has been traversed in both directions, the interior polygons will be traced in one direction and the outer ones in the other.

For a polygon to possibly correspond to an object face, the number of sides and the number of convex and concave angles of this polygon must be the same as for some polygon in at least one of the models. Such polygons will be called *approved* polygons (partially occluded faces are not approved polygons).

The topological match proceeds by first looking for a vertex completely surrounded by approved polygons (vertex *C* in Fig. 3-1 is such a vertex). If such a vertex exists, it is characterized by the number of sides of the surrounding polygons and is matched with a model vertex with the same characteristics. Matching of the vertices then leads to the matching of the polygons around them and the other vertices of these polygons.

In a partially occluded object, no vertices of the above types may exist, as would be the case if vertex *A* were hidden in Fig. 3-1. In this case, the topological match uses a line surrounded by approved polygons (such as line *CF* for the cube example) to obtain matching lines and then matching polygons and vertices. In case of further occlusion, the desired line may also not exist; in this case, a single approved polygon is used. Failing this also, any vertex with three (or more) lines is used for matching. (A more formal approach to such structural matching is given in [3].)

Once a topological match with a model is found, a geometric transformation for the best geometric match with that model is computed. Suppose that

$V_p = \{v_{1p}, v_{2p}, \ldots, v_{np}\}$ are the vertices in the picture,

$V_m = \{v_{1m}, v_{2m}, \ldots, v_{nm}\}$ are the matching model vertices and,

$V'_p = \{v'_{1p}, v'_{2p}, \ldots, v'_{np}\}$ are the predicted vertices, under a transformation $\mathbf{T}_m$.

We wish to choose $\mathbf{T}_m$ to minimize E_m, which is defined as

$$E_m = \sum_{i=1}^{n} \|v_{ip} - v'_{ip}\| \tag{3-3}$$

(Details of a transformation and minimization are given in Section 3.2 below.)

The model producing the minimum matching error is chosen, consistent with one more requirement. The picture positions for the unmatched model vertices must *not* fall outside the picture line drawing—that is, they must be predicted to be hidden. This is verification by picture synthesis.

If the scene contains more than one object, the above procedure will find one object at a time. Once an object has been recognized, the lines corresponding to it are removed and the remainder of the scene processed repeatedly as before. Complex objects formed by composing simpler objects are analyzed similarly. A sequence of such processing is

shown in Fig. 3-5. This method is effective only if parts of each object sufficient for recognition are visible. (Analysis of complex, occluding scenes is discussed in the next chapter.)

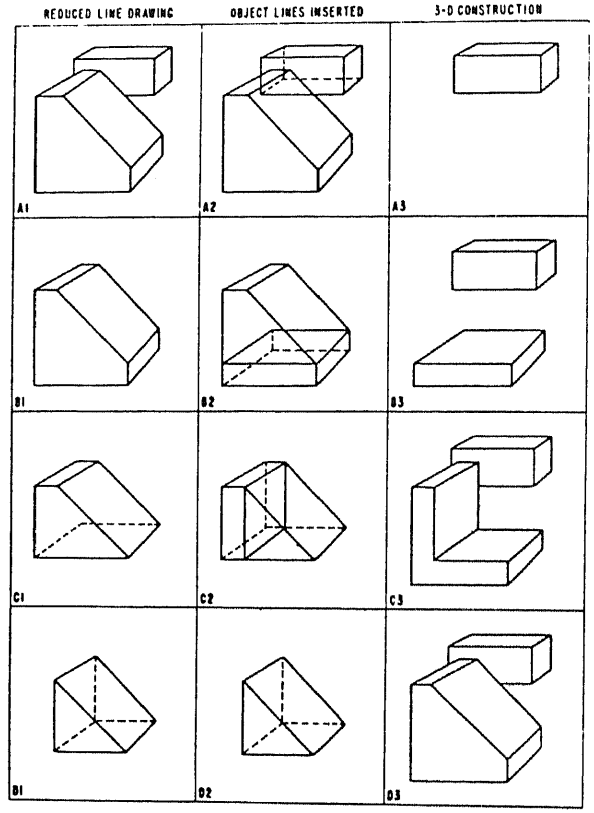


Figure 3-5: Successive analysis of a complex object (from Roberts [1])

3.2 MODEL TRANSFORMATIONS

A component of object recognition is to verify that a hypothesized model produces an image similar to the observed image, under some permissible transformation. We will consider the transformations of scaling, translation, rotation, and image formation—that is, the perspective transformation.

3.2.1 Perspective Transformations

A typical camera consists of a lens and a plane on which the image is formed. For the purpose of geometrical optics (that is, ignoring diffraction effects), an ideal camera can be modeled as a "pin-hole" camera. Such an ideal imaging system is shown in Fig. 3-6 and consists of a lens center C and an image plane I, a distance f from C. f is known as the focal length of the imaging system. The image of a given point P is formed on the image plane I at point P' determined by the intersection of the ray connecting C and P with the plane I. In Fig. 3-6 the ideal image plane is shown to be in *front* of the lens center; in normal camera systems the physical image plane is *behind* the lens center, and the image is inverted. We assume that for algebraic simplicity the inverted image has been corrected to correspond to the geometry shown. The image using a TV camera, as seen on a TV monitor, corresponds to the image plane being in the front.

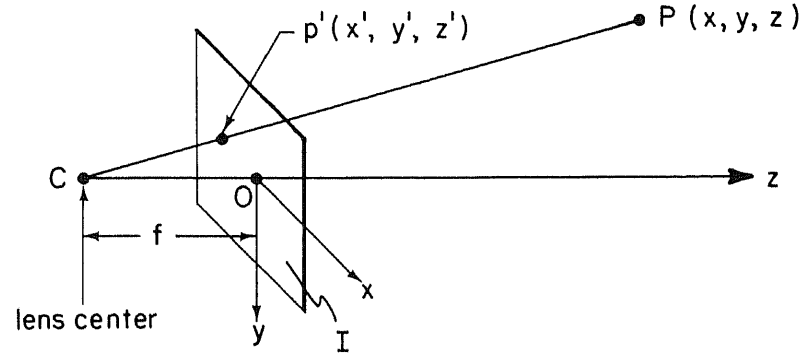


Figure 3-6: An ideal imaging system

Let a Cartesian coordinate system be chosen with the z axis normal to the image plane. Let the origin be on the image plane along the principal ray, which is the line from the lens center perpendicular to the image plane. Then the image $P'(x', y', z')$ of an object point P at location (x, y, z) is given by

$$x' = \frac{fx}{f + z} \tag{3-4}$$

$$y' = \frac{fy}{f + z} \tag{3-5}$$

$$z' = 0 \tag{3-6}$$

The above transformation from (x, y, z) to (x', y', z') is known as a perspective transformation. Note that the transformation is not invertible; that is, given a picture point (x', y', z'), we cannot completely specify the corresponding object point but can only constrain it to lie along a certain straight line.

3.2.2 Homogeneous Coordinates

The perspective transformation involves a division and is thus nonlinear. However, it can be linearized by use of homogeneous coordinates represented in a matrix form. Homogeneous coordinates of a point are defined by appending an extra component to the coordinate vector of the point and are related to the ordinary coordinates as follows. If (x_h, y_h, z_h, w_h) are the homogeneous coordinates of a point P with normal coordinates (x, y, z), then

$$x = \frac{x_h}{w_h} \tag{3-7}$$

$$y = \frac{y_h}{w_h} \tag{3-8}$$

$$z = \frac{z_h}{w_h} \tag{3-9}$$

The choice of w_h is arbitrary and thus the homogeneous coordinates of a point are not unique. Now consider the following matrix transformation

$$\begin{bmatrix} x'_h \\ y'_h \\ z'_h \\ w'_h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} x_h \\ y_h \\ z_h \\ w_h \end{bmatrix} \tag{3-10}$$

Let the 4-by-4 matrix in the equation above be called **P**. This equation expands to

$$x'_h = x_h \tag{3-11}$$

$$y'_h = y_h \tag{3-12}$$

$$z'_h = 0 \tag{3-13}$$

$$w'_h = \frac{z_h}{f} + w_h \tag{3-14}$$

Dividing Eqs. (3-11), (3-12), and (3-13) by w'_h and rearranging them, we get

$$x' = \frac{x'_h}{w_h'} = \frac{fx}{f + z} \tag{3-15}$$

$$y' = \frac{y'_h}{w_h'} = \frac{fy}{f + z} \tag{3-16}$$

$$z' = 0 \tag{3-17}$$

These are identical with the transformation defined earlier [Eqs. (3-4), (3-5), and (3-6)]. Thus we have defined a linear transformation **P**, operating on homogeneous coordinates of a point and its image, that is equivalent to a perspective transformation.

The transformation defined above does not preserve any information about the distance along the z axis of an object point P. It is sometimes useful to augment the matrix **P** to be

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix} \tag{3-18}$$

to give

$$z' = \frac{fz}{f + z} \tag{3-19}$$

The value of z' no longer corresponds to the actual image point. However, the image point is known to be at $z' = 0$, and the additional information contained in Eq. (3-19) above is useful for shading and hidden line elimination in computer graphics.

3.2.3 Geometrical Transformations

The perspective transformation given above applies when the object and the image points are specified in a coordinate system aligned with the camera. It may be more convenient to express the objects in an independent coordinate system, sometimes called a *world* coordinate system. In this case, the object coordinates must be first transformed to a system aligned with the camera, before the perspective transformation can be applied. The two systems can be aligned by a translation and sequential rotation about the three coordinate axes. The three rotation angles are sometimes referred to as *pan*, *tilt*, and *swing* (or *roll*). (Usually, pan refers to rotation of the principal ray in a horizontal plane tilt to its rotation in a vertical plane, and swing to a rotation of the image plane; see Fig. 3-7.)

Rotation. Any rotation of a model can be decomposed into three consecutive rotations about the three coordinate axes. Let us consider a rotation of axes about the z axis by an angle θ, as shown in Fig. 3-8. The direction of the rotation is from the x axis to the y axis. The coordinates of the point (x, y, z) in the rotated coordinate system are given by

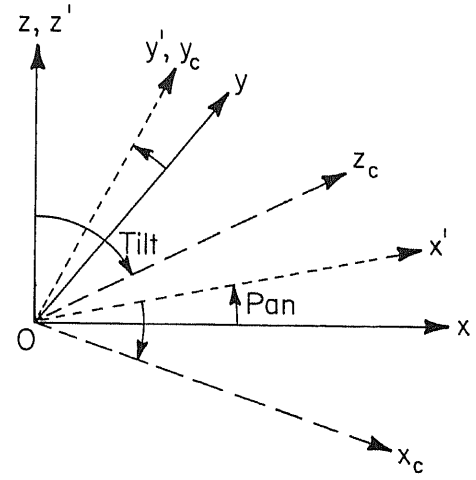


Figure 3-7: An example to illustrate pan and tilt angles

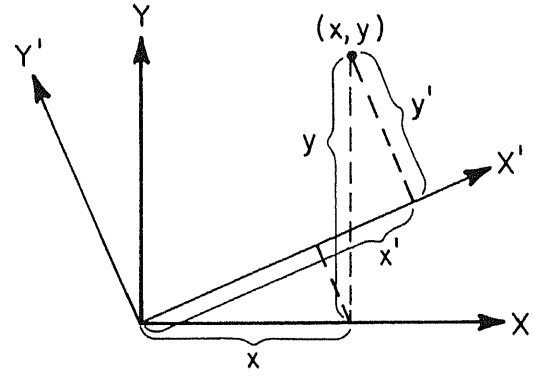


Figure 3-8: Rotation of coordinates about the z axis

$$x' = x \cos\theta + y \sin\theta \tag{3-20}$$

$$y' = -x \sin\theta + y \cos\theta \tag{3-21}$$

$$z' = z \tag{3-22}$$

The above transformation can be represented as the following matrix (for non-homogeneous coordinates):

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-23}$$

Similarly, rotation by angles ϕ and ψ about the x and y axes (in the directions of y to z, and z to x) are given by the following two matrices, respectively:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3-24}$$

and

$$\begin{bmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix} \tag{3-25}$$

Transformation about more than one axis is obtained by a multiplication of the above matrices. The general form of this matrix for homogeneous coordinates is

$$R = \begin{bmatrix} & & & 0 \\ & [R'] & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-26}$$

where $[R']$ is a 3-by-3 matrix, corresponding to rotational transformation in nonhomogeneous coordinates.

Translation. Translation of axes by (x_0, y_0, z_0) is represented by the matrix **T** where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-27}$$

Scaling. Scaling of the model by amount $|s|$ is given by the matrix **S**, where

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{s} \end{bmatrix} \tag{3-28}$$

Note that different scalings along the three axes may be represented by nonunit terms in the diagonal of matrix **S**.

Any composite transformation of the object to a picture is now represented by a product of the matrices of different transformations. Owing to use of the homogeneous coordinates, it is possible to include the transformations subsequent to the picture formation, such as scaling and rotation in a printing process, into a single matrix transformation. Such a transformation is often called a *camera model.*

3.3 FITTING OF MODELS

We seek to choose a model, m, such that the predicted image of the model under some transformation $\mathbf{T}_m$ is most similar to the observed image. $\mathbf{T}_m$ is chosen to minimize the error E_m between a set of picture points V_p, and the predicted image points V'_p from a set of model points V_m as defined in Section 3.1.2 earlier [see Eq. (3-3)]. Fortunately, this optimal transformation $\mathbf{T}_m$ can be determined analytically.

Let a model point, v_{im}, be represented in homogeneous coordinates by a column vector $(x_{im}, y_{im}, z_{im}, w_{im})^t$ and an image point v_{ip} by a vector $(x_{ip}, y_{ip}, w_{ip})^t$. Note that the two vectors use different coordinate systems, the model points are given in a world coordinate system, chosen for convenience of measuring model coordinates, and the image points are given in a coordinate system with the x and y axes in the image plane. Let $\mathbf{V}_m$ be a 4-by-n matrix whose n columns are the

coordinates of the n model points in the set V_m, and let $\mathbf{V}_p$ be a similar 3-by-n matrix corresponding to the picture points in the set V_p. Let matrix $\mathbf{H}$(3-by-4) represent the transformation of model points to picture points. If error E_m could be reduced to zero, we would have

$$\mathbf{H}\mathbf{V}_m = \mathbf{V}_p\mathbf{D} \tag{3-29}$$

where $\mathbf{D}$ is a diagonal, n-by-3 matrix. (This matrix is necessary because the scaling of each point in the homogeneous coordinates may be different.) Note that we have twelve unknowns in the $\mathbf{H}$ matrix and n unknowns in the $\mathbf{D}$ matrix, one for each point. At least six points are necessary for a nondegenerate solution of Eq. (3-29). Without proof, we present the optimal solution. (This solution ignores the interdependence of the elements of matrix $\mathbf{H}$ and is also known as the *pseudo-inverse* solution). Let

$$\mathbf{Q} = \mathbf{V}_p^{\mathrm{T}}\mathbf{V}_p \tag{3-30}$$

$$\mathbf{A} = \mathbf{V}_m^{\mathrm{T}}(\mathbf{V}_m\mathbf{V}_m^{\mathrm{T}})^{-1}\mathbf{V}_m - \mathbf{I} \tag{3-31}$$

where $\mathbf{I}$ is an identity matrix. Define a matrix $\mathbf{S}$ such that $s_{ij} = a_{ij}q_{jj}$. Then $\mathbf{D}$ is determined by the solution of the linear equation given by

$$\mathbf{S}\mathbf{D} = 0 \tag{3-32}$$

and

$$\mathbf{H} = \mathbf{V}_p\mathbf{D}\mathbf{V}_m^{\mathrm{T}}(\mathbf{V}_m\mathbf{V}_m^{\mathrm{T}})^{-1} \tag{3-33}$$

After the transformation $\mathbf{T}_m$ has been determined, the images of the non-matched model vertices are predicted. These vertices should be nonvisible, and their images should not fall outside the object outline, else the model is an unacceptable match.

The match computed by Eq. (3-32) will be ambiguous with respect to the scale factor. A larger model farther away from the camera produces the same image (ignoring absolute brightness levels). If the distance to the object is known, the size can be computed and vice versa. In simple situations, the distance may be computed by assuming that an object rests on a known plane or on another object. More complex depth-measurement schemes are described in Chapter 9.

The illusions of Figs. 1-6(a) and (b) can now be explained, if we

assume that the human system tries to interpret line drawings as representing three-dimensional scenes when possible. In Fig. 1-6(a), the two converging lines may be interpreted as two parallel lines in 3-D viewed with a perspective projection. Then the two horizontal bars must be at different distances from the viewer, and since their lengths in the image are the same, the top bar must be longer in 3-D. In Fig. 1-6(b), the left figure may be interpreted as representing an inside corner of a room, and the right figure as the outside corner. Again, differences in the interpreted distance explain the difference in observed lengths. Note that 3-D interpretation provides only one explanation of these illusions; other explanations can be found in the psychology literature.

3.3.1 Camera Calibration

The transform relating the coordinates of the objects in a certain world coordinate system to the image coordinates, such as the one represented by the matrix $\mathbf{H}$ above, is also known as the *camera model* or the *camera transform.* For many applications, the camera is fixed relative to a world coordinate frame, and it is useful to measure or calibrate the parameters of the camera transform. The parameters relating the image and the world coordinate systems such as pan, tilt, and swing may be difficult to measure directly. An alternative is to observe a known 3-D object in a known position and orientation and measure points in the image corresponding to known points on the object. The transform can then be computed as in Eq. (3-33) above.

However, this solution may be inaccurate, as the twelve elements of $\mathbf{H}$ are assumed to be independent. In fact, the matrix $\mathbf{H}$ is completely determined by the three angles, pan, tilt and swing, and the location of the lens center (three coordinates), assuming that the focal length and the scaling of the image are known. More accurate estimates of $\mathbf{H}$ can be obtained by using standard, though computationally expensive, nonlinear optimization techniques for minimizing E_m defined in Eq. (3-28). Also, if nonlinear techniques are used, three points are sufficient for determining the parameters of $\mathbf{H}$ rather than the six needed for the linear solution of Eq. (3-33). Another important consideration is to ignore the effects of isolated points that are in gross error and can have a large effect if the least-mean-square criterion is used. Some non-linear calibration techniques may be found in [4-6].

For some applications, the camera is mounted such that its position may be changed—for example, along a horizontal plane with the camera looking down—or the camera orientation (pan, tilt, or swing) changed to observe different parts of the scene or track an

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].