# 26

# Recognizing Objects

In this chapter, you learn how it is possible to *identify objects by constructing custom-tailored templates* from stored two-dimensional image models. Amazingly, the template-construction procedure just adds together weighted coordinate values from corresponding points in the stored two-dimensional image models. For orthographic projections, the template is perfect—and it is nearly perfect even in perspective projections.

Previously, many researchers thought object identification would have to be done via the transformation of images into explicit three-dimensional descriptions of the objects in the images. The template-construction procedure involves no such explicit three-dimensional descriptions.

The template construction procedure does require knowledge of which points correspond, however. Accordingly, you also learn about *methods for solving the correspondence problem.*

By way of illustration, you see how the linear combination procedure handles similar-looking stylized objects: one is an "obelisk," and another is a "sofa."

Once you have finished this chapter, you will know how the linear combination procedure works, you will appreciate its simple elegance, and you will understand when it is the right approach.

## LINEAR IMAGE COMBINATIONS

In this section, you learn how identification can be done by template construction and straightforward matching.

## Conventional Wisdom Has Focused on Multilevel Description

Most of the chapters in this book present ideas without providing any explanation about how those ideas emerged. Accordingly, it is easy to imagine that solutions to hard problems improve steadily, whereas, in fact, the march toward solutions to hard problems always seems to involve long periods of little progress, punctuated occasionally by a startling advance.

To appreciate the startling advance associated with the ideas introduced in this chapter, you really need to know that many vision experts believed for years that object identification would require image processing on several descriptive levels, with matching occurring only on the highest:

- At the lowest descriptive level, brightness values are conveyed explicitly in the image.
- The brightness changes in the image are described explicitly in the **primal sketch**.
- The surfaces that are implicit in the primal sketch are described explicitly in the **two-and-one-half–dimensional sketch**.
- The volumes that are implicit in the two-and-one-half–dimensional sketch are described explicitly in the **volume description**.

Information in the primal sketch and the two-and-one-half–dimensional sketch describes what is going on at each point in the original image. Hence, the primal sketch and the two-and-one-half–dimensional sketch often are said to be **viewer centered**.

Unlike the information in the primal sketch and the two-and-one-half–dimensional sketch image, the information in a volume description often is expressed in terms of coordinate systems attached to objects. Such descriptions are said to be **object centered**. Only after constructing a volume description, according to conventional wisdom, can you go into a library and match a description extracted from an image with a remembered description.

Characteristically, conventional wisdom has turned out to be completely wrong. In the rest of this section, you learn that matching can be done at the primal sketch level, rather than at the volume description level.

## Images Contain Implicit Shape Information

In elementary geometry, you learned that a polyhedron is an object whose faces are all flat. When you look at a mechanical drawing of a polyhedron, the combination of a front view, a side view, and a top view of that polyhedron is sufficient to give you full knowledge of each vertex's three-dimensional position.

More generally, a few images, each showing a few corresponding vertexes, give you an idea of where those corresponding vertexes are, relative to one another, in three dimensions, even when the images are not the standard front, side, and top views.

For a long time, however, it was not clear how many images and how many vertexes are required to recover where the vertexes are, relative to one another, in three dimensions. Then, in 1979, Shimon Ullman showed that three images, each showing four corresponding vertexes, are almost enough to determine the vertexes' relative positions. All that you need in addition is some source of information about the polyhedron's size, such as the distance between any two vertexes.

Thus, three images carry implicit knowledge of a polyhedron's shape as long as those images all contain at least four corresponding vertexes. If you make the knowledge of shape explicit by deriving the three-dimensional coordinate values of all the vertexes, then you can construct any other image by projecting those vertexes through a suitably placed eye or lens onto a suitably placed biological retina or artificial sensor array.

Knowing that any other image can be constructed via the intermediate step of deriving three-dimensional coordinate values naturally leads to two important questions:

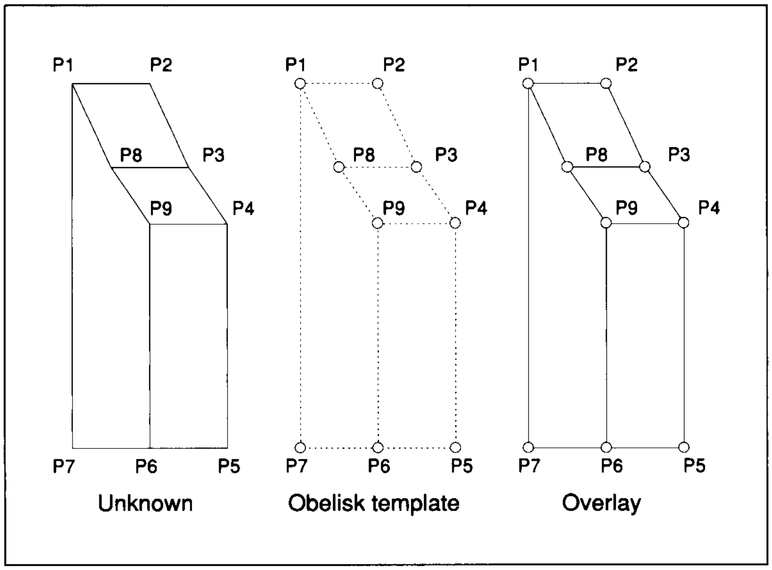
- Given three recorded images of a polyhedron, are there simple equations that predict the coordinate values of the points in a new, fourth image using only the coordinate values of the corresponding points in the three recorded images?
- If the answer to the first question is yes, is it possible to determine all the parameters in those simple prediction equations using only the coordinate values of a few of the corresponding points in the recorded images and the new image?

Happily, there are simple equations that predict the coordinate values, and it is possible to determine all the parameters in those simple prediction equations using only a few corresponding points. The coordinate values of the points in a new, fourth image are given by a linear combination of the coordinate values of the points of the three recorded images. Also, you can determine the constants involved in the linear combination by solving a few linear equations involving only a few of the corresponding points. Consequently, when presented with an image of an unidentified polyhedron, you can determine whether it can be an image of each particular polyhedron recorded in a library.

## One Approach Is Matching Against Templates

To streamline further discussion, let us agree to call each object-describing image collection a **model**, short for **identification model**, a representation specified loosely as follows:

**Figure 26.1** One possible approach to identification is to create templates from models. On the left, you see an object that may be an obelisk; in the middle, you see an obelisk template; and on the right, the two are overlaid. One question is whether the models can consist exclusively of stored two-dimensional images.



An **identification model** is a representation

In which

- ▷ An image consists of a list of identifiable places, called **feature points**, observed in an image.
- ▷ The model consists of several images—minimally three for polyhedra.

Furthermore, let us agree to call each unidentified object an *unknown*. Using this vocabulary of models and unknowns, you want to know whether it is practicable to match an unknown with a model by comparing the points in an image of the unknown with a templatelike collection of points produced from the model. In figure 26.1, for example, the nine points seen in the image of an unknown match the nine points in an overlaid, custom-made obelisk template.

Whenever a general question is hard, it is natural to deal with a special case first. Accordingly, suppose that objects are allowed to rotate around the vertical axis only; there are to be no translations and no rotations about other axes. In figure 26.2, for example, the obelisk shown in figure 26.1 is viewed in its original position, and then is rotated about the vertical axis by 30°, 60°, and 90°.

In each of the images shown in figure 26.2, the obelisk is *projected orthographically* along the $z$ axis. As explained in figure 26.3, the $x$ and $y$ coordinate values for points in the image are exactly the $x$ and $y$ coordinate values of the obelisk's vertexes in three dimensions.

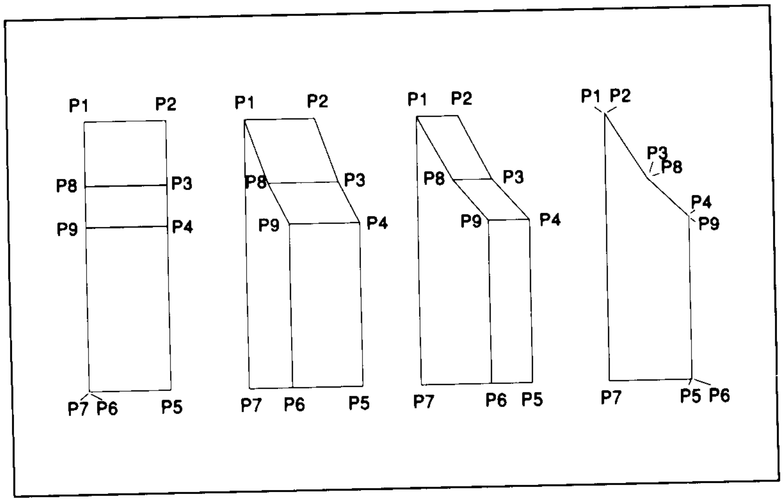


**Figure 26.2** An "obelisk" and three orthographic projections of the obelisk, one each for 30°, 60°, and 90° rotations.

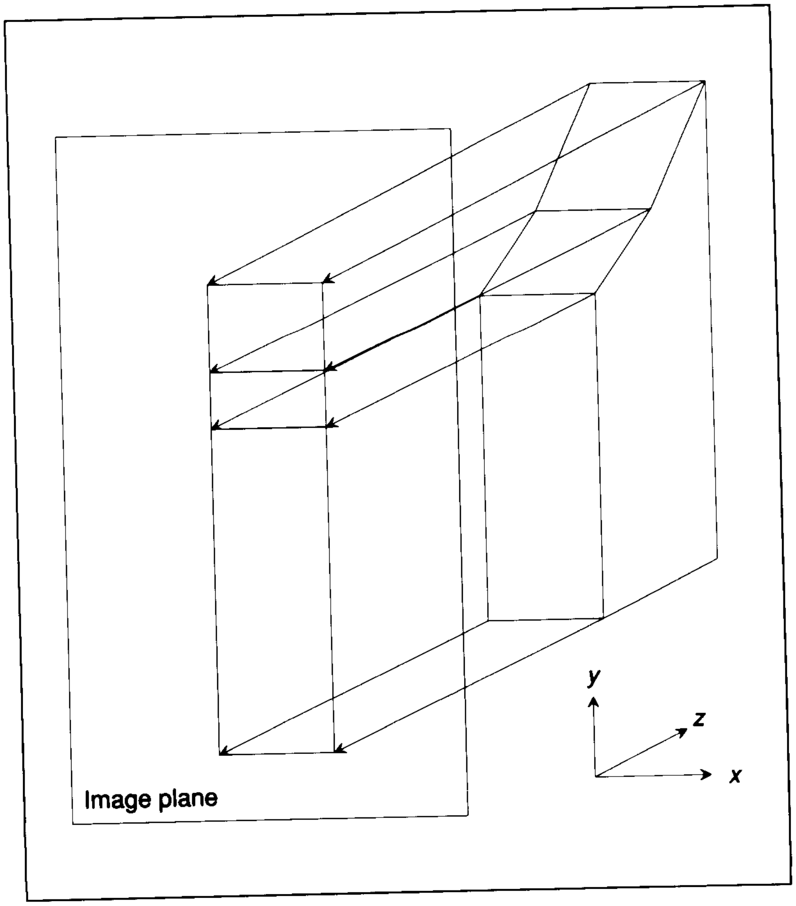


**Figure 26.3** Orthographic projection. Light moves along paths parallel to the *z* axis to an image plane somewhere on the *z* axis. The *x* and *y* coordinate values of the vertexes in an image equal their three-dimensional *x* and *y* coordinate values.

Somewhat unusually, the coordinate system shown in figure 26.3 is a left-handed coordinate system, but most people like to have the distance from the image plane increase with increasing $z$, which dictates the left-handed arrangement.

Next, note that corresponding points in figure 26.2 are labeled with corresponding numbers. In general, matching points in one image with those in another to establish correspondence may be difficult; for the moment, however, just assume that the necessary matching can be done. You learn about several approaches in the next section.

Now, by way of preview, consider a point that appears with an $x$ coordinate value of $x_{I_1}$ in one obelisk model image, and with an $x$ coordinate value of $x_{I_2}$ in another obelisk model image. Soon, you learn that the $x$ coordinate value of the same point in an observed image is a weighted sum of the coordinate values seen in the model images:

$$x_{I_o} = \alpha x_{I_1} + \beta x_{I_2}.$$

Next you see that $\alpha$ and $\beta$ can be recovered using a few corresponding points, making it easy to predict where the remaining points should be; providing, ultimately, an identification test.

## For One Special Case, Two Images Are Sufficient to Generate a Third

The nature of orthographic projection is such that a point corresponding to a vertex located at $(x,\ y,\ z)$ in space is located at $(x,\ y)$ in the orthographic image. After rotation about the $y$ axis, however, the vertex is no longer at $(x,\ y,\ z)$. Although the $y$ coordinate value is unchanged, both the $x$ and $z$ coordinate values change. As demonstrated by some straightforward trigonometry, the coordinate values, after rotation, are determined by the sines and cosines of the rotation angle, $\theta$. More precisely, the new $x$, $y$, and $z$ coordinate values, $x_\theta$, $y_\theta$, and $z_\theta$, are related to the old $x$, $y$, and $z$ coordinate values by the following equations:

$$\begin{aligned} x_\theta &= x\cos\theta - z\sin\theta,\\ y_\theta &= y,\\ z_\theta &= x\sin\theta + z\cos\theta. \end{aligned}$$

Because the $y$ coordinate value endures rotation and orthographic projection without change, and because the $z$ coordinate value does not enter into orthographic projection, your interest is exclusively in the fate of the $x$ coordinate value as an object rotates.

To keep the discussion as concrete as possible, assume that the obelisk is rotated from its original position twice, once by $\theta_{I_1}$ and once by $\theta_{I_2}$, to produce two model images, $I_1$ and $I_2$. What you want to know is what the obelisk looks like when rotated from its original position a third time, by $\theta_{I_o}$, producing image $I_o$.

Consider a particular point with original coordinate values $x$, $y$, and $z$. In the two model images of the rotated obelisk, the $x$ coordinate values of the same point are $x_{I_1}$ and $x_{I_2}$. The problem is to find a way of determining $x_{I_o}$ given the known values, $x_{I_1}$ and $x_{I_2}$. Once you can do this for any particular point, you can do it for all points and thus construct the desired third image of the obelisk. So far, however, you can only relate $x_{I_1}$, $x_{I_2}$, and $x_{I_o}$ to the original $x$ and $z$ values and the angles of rotation:

$$x_{I_1} = x \cos\theta_{I_1} - z \sin\theta_{I_1},$$
$$x_{I_2} = x \cos\theta_{I_2} - z \sin\theta_{I_2},$$
$$x_{I_o} = x \cos\theta_{I_o} - z \sin\theta_{I_o}.$$

To understand what these equations can do for you, you need to understand one subtle point. $x_{I_1}$, $x_{I_2}$, and $x_{I_o}$ vary from point to point, but $\theta_{I_1}$, $\theta_{I_2}$, and $\theta_{I_o}$ do not. The angles do not change as long as you are working with a fixed set of images. Accordingly, if you are trying to solve the equations for $x_{I_o}$ for a fixed set of images, the sines and cosines of the angles are constants.

Better still, the equations are three linear equations in the three unknowns, $x_{I_o}$, $x$, and $z$. From elementary algebra, you know that you can solve three linear equations in three unknowns, leaving $x_{I_o}$ expressed as a weighted sum of $x_{I_1}$ and $x_{I_2}$:

$$x_{I_o} = \alpha x_{I_1} + \beta x_{I_2}.$$

Describing this expression in mathematical language, $x_{I_o}$ is given by a **linear combination** of $x_{I_1}$ and $x_{I_2}$, and $\alpha$ and $\beta$ are called the **coefficients** of the linear combination.

Now, if you only knew the actual values for $\alpha$ and $\beta$ relating observed points to points in the two model images, you could predict where every point should be from where it appears in the two model images. If the predicted points match the actual points observed, then the observed object matches the model object.

Of course, you could work through the algebra and determine how $\alpha$ and $\beta$ can be expressed in terms of sines and cosines of $\theta_{I_1}$, $\theta_{I_2}$, and $\theta_{I_o}$. But that would not help you, because you normally do not know any of those angles. You need another approach.

## Identification Is a Matter of Finding Consistent Coefficients

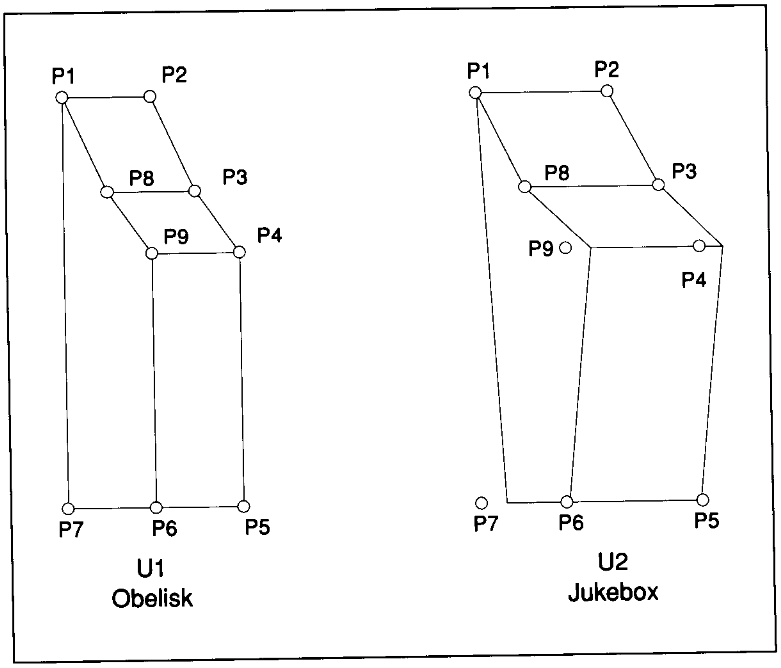
You have just learned that there must be constants $\alpha$ and $\beta$ such that the coordinate values in an observed image are predicted by the equation

$$x_{I_o} = \alpha x_{I_1} + \beta x_{I_2}.$$

Because $\alpha$ and $\beta$ depend on only the three images, $I_1$, $I_2$, and $I_o$, you need only two linear equations to determine their values.

Fortunately, two sets of corresponding points provide those equations. Suppose, for example, that you have found a point $P_1$ in the observed image

**Figure 26.4** Two unknown objects compared with templates made from obelisk images to fit points $P_1$ and $P_2$. The unknown on the left is actually an obelisk rotated by 45°. It matches the template of circles produced from the two obelisk models. The unknown on the right—the one that looks like a jukebox—does not match the template made for it.



and the corresponding points in the model images. Similarly, suppose you have found point $P_2$ and its corresponding points. Then, the $x$ coordinate values of both points in the observed image must satisfy the following equations:

$$x_{P_1 I_o} = \alpha x_{P_1 I_1} + \beta x_{P_1 I_2},$$
$$x_{P_2 I_o} = \alpha x_{P_2 I_1} + \beta x_{P_2 I_2}.$$

Now you have two, easily solved equations in two unknowns. The solutions are, of course,

$$\alpha = \frac{x_{P_1 I_o} x_{P_2 I_2} - x_{P_2 I_o} x_{P_1 I_2}}{x_{P_1 I_1} x_{P_2 I_2} - x_{P_2 I_1} x_{P_1 I_2}},$$
$$\beta = \frac{x_{P_2 I_o} x_{P_1 I_1} - x_{P_1 I_o} x_{P_2 I_1}}{x_{P_1 I_1} x_{P_2 I_2} - x_{P_2 I_1} x_{P_1 I_2}}.$$

Once you have used two of $n$ sets of corresponding points to find $\alpha$ and $\beta$, you can use $\alpha$ and $\beta$ to predict the positions of the remaining $n-2$ sets of corresponding points of the observed object:

$$x_{P_i I_o} = \alpha x_{P_i I_1} + \beta x_{P_i I_2}.$$

The predicted point positions then act like a template that the points in the observed image must match if the object in the image is the same object seen in the model images.

For the obelisk example, the following table gives the $x$ coordinate values for points $P_1$ and $P_2$ in the images of the obelisk after rotation by 30° and 60°. The table also gives the $x$ coordinate values of corresponding points on the two unknown objects shown in figure 26.4. The $y$ coordinate values are not shown, because they do not vary from image to image.

| | Position in image $I_1$ | Position in image $I_2$ | Position in image $U_1$ | Position in image $U_2$ |
|---|---|---|---|---|
| $x_{P_1}$ | −2.73 | −2.73 | −2.83 | −3.54 |
| $x_{P_2}$ | 0.73 | −0.73 | 0 | 0.71 |

Using the $x$ coordinate values, you can calculate what $\alpha$ and $\beta$ must be for each of the two unknown objects by substituting values from the table into $x_{P_i I_o} = \alpha x_{P_i I_1} + \beta x_{P_i I_2}$. For the first unknown object, you have, for example, the following equations after substitution:

$$-2.83 = \alpha(-2.73) + \beta(-2.73),$$
$$0 = \alpha(0.73) + \beta(-0.73).$$

Solving these equations, you have $\alpha = 0.518315$ and $\beta = 0.518315$. Solving the corresponding equations for the second unknown yields $\alpha = 1.13465$ and $\beta = 0.16205$. You can use these $\alpha$ and $\beta$ values to predict the $x$ coordinate values for each of the remaining points, with the following results:

| | $U_1$ predicted | $U_1$ actual | $U_2$ predicted | $U_2$ actual |
|---|---|---|---|---|
| $x_{P_1}$ | −2.83 | −2.83 | −3.54 | −3.54 |
| $x_{P_2}$ | 0 | 0 | 0.70 | 0.70 |
| $x_{P_3}$ | 1.41 | 1.41 | 2.12 | 2.12 |
| $x_{P_4}$ | 2.83 | 2.83 | 3.54 | 4.24 |
| $x_{P_5}$ | 2.83 | 2.83 | 3.54 | 3.54 |
| $x_{P_6}$ | 0 | 0 | −0.70 | −0.70 |
| $x_{P_7}$ | −2.83 | −2.83 | −3.54 | −2.83 |
| $x_{P_8}$ | −1.41 | −1.41 | −2.12 | −2.12 |
| $x_{P_9}$ | 0 | 0 | −0.70 | 0 |

Evidently, the first unknown is an obelisk, because all predicted points are where they should be, as shown in figure 26.4; the second unknown is not an obelisk, because three predicted points are in the wrong place. Although the second unknown has much in common with the obelisk, the second unknown is wider, and its front and back are tapered, rather than vertical, making it look a bit like a jukebox.

## The Template Approach Handles Arbitrary Rotation and Translation

In one special case, you have seen that identification can be done using only two model images together with two points that appear in both of those model images and in an image of an unknown object. The special case is severely restricted, however; only rotation around one axis is permitted.

You have concentrated on the one-axis-rotation special case for two reasons. First, looking at special cases is the sort of thing researchers generally do when they are trying to develop a feel for what is going on. Second, the one-axis-rotation special case is within the reach of straightforward mathematics.

More generally, however, you have to expect that an unknown may have been arbitrarily rotated, arbitrarily translated, and even arbitrarily scaled relative to an arbitrary original position. To deal with these changes, you first have to believe, without proof here, that an arbitrary rotation of an object transforms the coordinate values of any point on that object according to the following equations:

$$\begin{aligned}
x_\theta &= r_{xx}(\theta)x + r_{yx}(\theta)y + r_{zx}(\theta)z,\\
y_\theta &= r_{xy}(\theta)x + r_{yy}(\theta)y + r_{zy}(\theta)z,\\
z_\theta &= r_{xz}(\theta)x + r_{yz}(\theta)y + r_{zz}(\theta)z.
\end{aligned}$$

Note that $r_{xx}(\theta)$ is the parameter that shows how much the $x$ coordinate of a point, before rotation, contributes to the $x$ coordinate of the same point after rotation. Similarly, $r_{yz}(\theta)$ is the parameter that shows how much the $y$ coordinate of a point, before rotation, contributes to the $z$ coordinate of the same point after rotation.

If, in addition, the object is translated as well as rotated, each equation gains another parameter:

$$\begin{aligned}
x_\theta &= r_{xx}(\theta)x + r_{yx}(\theta)y + r_{zx}(\theta)z + t_x,\\
y_\theta &= r_{xy}(\theta)x + r_{yy}(\theta)y + r_{zy}(\theta)z + t_y,\\
z_\theta &= r_{xz}(\theta)x + r_{yz}(\theta)y + r_{zz}(\theta)z + t_z.
\end{aligned}$$

where the $t$s are all parameters that are determined by how much the object is translated.

Now you can repeat the development for the one-axis-only special case, only there must be three model images. These three model images yield the following equations relating model and unknown coordinate values to unrotated, untranslated coordinate values, $x$, $y$, and $z$:

$$\begin{aligned}
x_{I_1} &= r_{xx}(\theta_1)x + r_{yx}(\theta_1)y + r_{zx}(\theta_1)z + t_x(\theta_1),\\
x_{I_2} &= r_{xx}(\theta_2)x + r_{yx}(\theta_2)y + r_{zx}(\theta_2)z + t_x(\theta_2),\\
x_{I_3} &= r_{xx}(\theta_3)x + r_{yx}(\theta_3)y + r_{zx}(\theta_3)z + t_x(\theta_3),\\
x_{I_o} &= r_{xx}(\theta_o)x + r_{yx}(\theta_o)y + r_{zx}(\theta_o)z + t_x(\theta_o).
\end{aligned}$$

Plainly, these equations can be viewed as four equations in four unknowns, $x$, $y$, $z$, and $x_{I_o}$, which can be solved to yield $x_{I_o}$ in terms of $x_{I_1}$, $x_{I_2}$, and $x_{I_3}$ and a collection of four constants:

$$x_{I_o} = \alpha_x x_{I_1} + \beta_x x_{I_2} + \gamma_x x_{I_3} + \delta_x,$$

where $\alpha_x$, $\beta_x$, $\gamma_x$, and $\delta_x$ are the constants required for $x$-coordinate-value prediction, each of which can be expressed in terms of $r$s and $t$s. There is no reason to go through the algebra, however, because, as in the one-axis-only case, there is another way to obtain the values of $\alpha_x$, $\beta_x$, $\gamma_x$, and $\delta_x$ without knowing the rotations and translations.

Following the development for the one-axis-only special case, you can use a few corresponding points to determine the constants. This time, however, there are four constants, so four points are required:

$$\begin{aligned}
x_{P_1 I_o} &= \alpha_x x_{P_1 I_1} + \beta_x x_{P_1 I_2} + \gamma_x x_{P_1 I_3} + \delta_x, \\
x_{P_2 I_o} &= \alpha_x x_{P_2 I_1} + \beta_x x_{P_2 I_2} + \gamma_x x_{P_2 I_3} + \delta_x, \\
x_{P_3 I_o} &= \alpha_x x_{P_3 I_1} + \beta_x x_{P_3 I_2} + \gamma_x x_{P_3 I_3} + \delta_x, \\
x_{P_4 I_o} &= \alpha_x x_{P_4 I_1} + \beta_x x_{P_4 I_2} + \gamma_x x_{P_4 I_3} + \delta_x.
\end{aligned}$$

Solving these equations for $\alpha_x$, $\beta_x$, $\gamma_x$, and $\delta_x$ enables you to predict the $x$ coordinate values of any point in the unknown image from the corresponding points in the three model images.

Note, however, that you have to consider the $y$ coordinate values also. There was no need to consider $y$ coordinate values in the one-axis-only special case because rotation was around the $y$ axis, leaving all $y$ coordinate values constant from image to image. In the case of general rotation and translation, the $y$ coordinate values vary as well.

Fortunately, the development of equations for $y$ values follows the development of equations for the $x$ values exactly, producing another, different set of four constants, $\alpha_y$, $\beta_y$, $\gamma_y$, and $\delta_y$ for the following equation:

$$y_{I_o} = \alpha_y y_{I_1} + \beta_y y_{I_2} + \gamma_y y_{I_3} + \delta_y.$$

Thus, the identification procedure requires three model images for each identity that an object might have:

```
To identify an unknown object,
▷ Until a satisfactory match is made or there are no more
  models in the model library,
    ▷ Find four corresponding points in the observed image
      and in a model's three library images.
    ▷ Use the corresponding points to determine the coeffi-
      cients used to predict the $x$ and $y$ coordinate values of
      other image points.
    ▷ Determine whether a satisfactory match is made by
      comparing the predicted $x$ and $y$ coordinate values with
      those actually found in the observed image.
▷ If a satisfactory match occurs, announce the identity of
  the unknown; otherwise, announce failure.
```

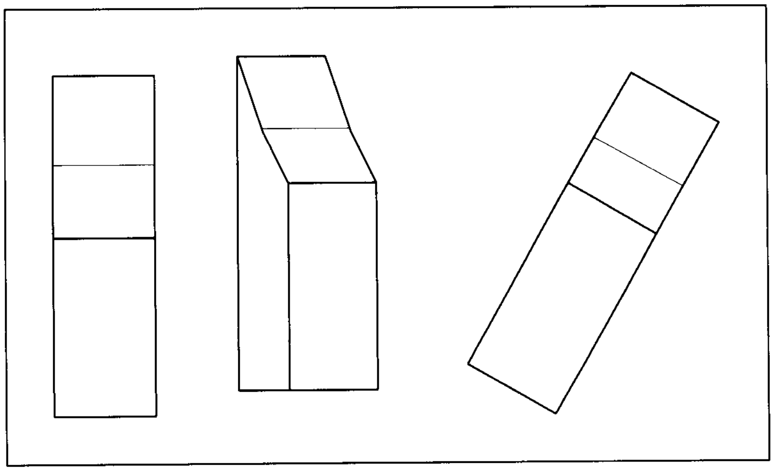


**Figure 26.5** Three obelisk images. The one on the left has been rotated 30° around the *x* axis relative to the standard initial standing position, pitching it forward; the second has been rotated around the *y* axis; and the third has been rotated around the *z* axis.

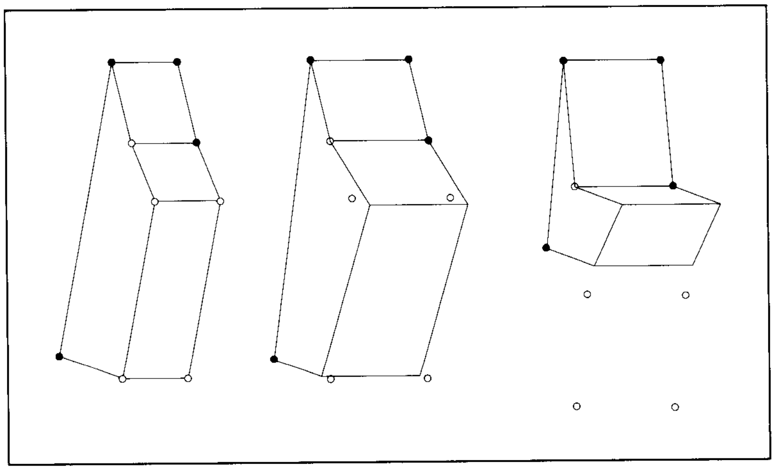


**Figure 26.6** Three unknown objects are compared to templates made from obelisk images using the points marked by black circles. The unknown on the left matches the obelisk template. The unknowns resembling a jukebox and a sofa do not match.

Thus, you need three images, like those in figure 26.5, to have an obelisk model that is good enough to recognize obelisks with unrestricted rotations and translations. With such a model, and knowledge of how four corresponding points in the model correspond to four points in each of the object images shown in figure 26.6, you identify just one of those objects as an obelisk.

## The Template Approach Handles Objects with Parts

So far, you have learned that the $x$ coordinate value of any point is given by a linear combination of the coordinate values in two or three model images. Now you learn that the $x$ coordinate value is given by a linear combination of the coordinate values in several model images, even if an object has parts that move relative to one another.

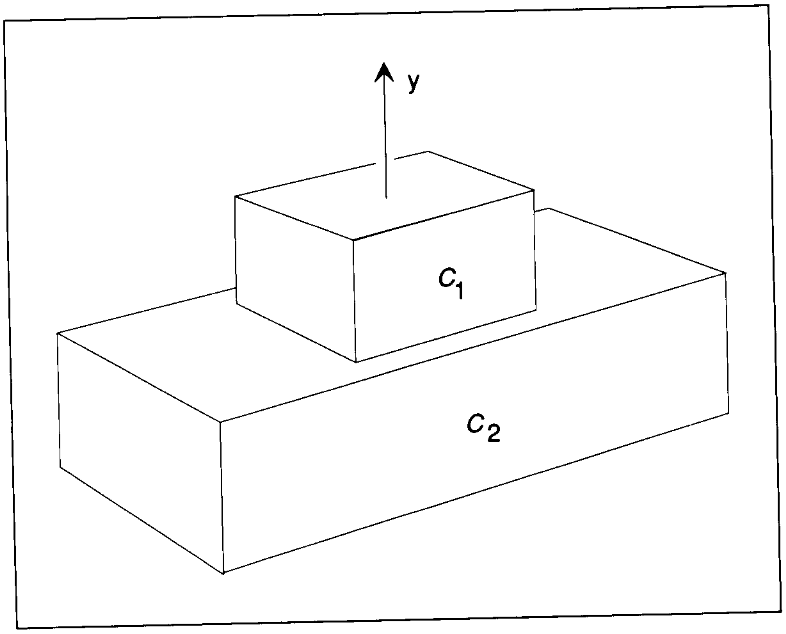


**Figure 26.7** An object with two parts, $C_1$ and $C_2$, each of which rotates independently about the *y* axis.

So as to keep the algebra as uncluttered as possible, the explanation is focused on the special case of rotation about the $y$ axis, and the example is the one shown in figure 26.7.

Note that each part is, by itself, one rigid object. Accordingly, the $x$ coordinate value of a point in an observed image, $x_{I_o}$ is determined by two model images and a particular $\alpha, \beta$ pair specialized to one part of the object, as indicated by elaborate $\alpha$ and $\beta$ subscripts:

$$x_{I_o} = \alpha_{C_1 I_o} x_{I_1} + \beta_{C_1 I_o} x_{I_2}.$$

The $C_1$ in the subscripts indicates that the $\alpha$ and $\beta$ values are for part 1; the $I_o$ subscript indicate that the $\alpha$ and $\beta$ values transform $I_1$ and $I_2$ values into $I_o$ values.

Similarly, $x_{I_o}$ is given by a different $\alpha, \beta$ pair if the other part is involved:

$$x_{I_o} = \alpha_{C_2 I_o} x_{I_1} + \beta_{C_2 I_o} x_{I_2}.$$

Amazingly, if you have enough images, you do not need to know to which part a point belongs, for there is a set of four coefficients—$\alpha, \beta, \delta, \gamma$—such that the $x$ coordinate value is determined by four model images, four coefficients, and the following equation, no matter to which part the point belongs:

$$x_{I_o} = \alpha x_{I_1} + \beta x_{I_2} + \delta x_{I_3} + \gamma x_{I_4}.$$

To see why, you have to go through a bit of algebra, but in spite of the necessary subscript clutter, there is only one mathematical insight involved: You can solve $n$ independent linear equations in $n$ variables.

Strategically, you need to convince yourself that the four-coefficient equation for $x_{I_O}$, the one that requires no knowledge of which part is involved, does the same thing as the two-coefficient equation for $x_{I_O}$, the one that does require knowledge of which part is involved. Said another way, you need to be sure that there is a set of four coefficients, $\alpha, \beta, \delta, \gamma$, such that the following is true if $x_{I_O}$ is a coordinate value of a point on part 1:

$$\alpha x_{I_1} + \beta x_{I_2} + \delta x_{I_3} + \gamma x_{I_4} = \alpha_{C_1 I_O} x_{I_1} + \beta_{C_1 I_O} x_{I_2}.$$

Alternatively, the following is true if $x_{I_O}$ is a coordinate value of a point on part 2:

$$\alpha x_{I_1} + \beta x_{I_2} + \delta x_{I_3} + \gamma x_{I_4} = \alpha_{C_2 I_O} x_{I_1} + \beta_{C_2 I_O} x_{I_2}.$$

Focus for now on the case for which $x_{I_O}$ is the $x$ coordinate of a point that belongs to part 1. Because you assume you are dealing with just one part, you know that two model images suffice to determine the $x_{I_O}$ coordinate value of any point in any image. Accordingly, $x_{I_3}$ and $x_{I_4}$ can be determined by $x_{I_1}$ and $x_{I_2}$, along with an $\alpha, \beta$ pair suited to the part and the images produced, as indicated by the subscripts:

$$\begin{aligned} x_{I_3} &= \alpha_{C_1 I_3} x_{I_1} + \beta_{C_1 I_3} x_{I_2}, \\ x_{I_4} &= \alpha_{C_1 I_4} x_{I_1} + \beta_{C_1 I_4} x_{I_2}. \end{aligned}$$

Evidently, $x_{I_O}$, given that the point is on part 1, must be determined by the following after substitution for $x_{I_3}$ and $x_{I_4}$:

$$x_{I_O} = \alpha x_{I_1} + \beta x_{I_2} + \delta(\alpha_{C_1 I_3} x_{I_1} + \beta_{C_1 I_3} x_{I_2}) + \gamma(\alpha_{C_1 I_4} x_{I_1} + \beta_{C_1 I_4} x_{I_2}).$$

Rearranging terms yields the following:

$$x_{I_O} = (\alpha + \delta\alpha_{C_1 I_3} + \gamma\alpha_{C_1 I_4}) x_{I_1} + (\beta + \delta\beta_{C_1 I_3} + \gamma\beta_{C_1 I_4}) x_{I_2}.$$

But you know that $x_{I_O} = \alpha_{C_1 I_O} x_{I_1} + \beta_{C_1 I_O} x_{I_2}$. So now you can equate the two expressions for $x_{I_O}$, yielding the following:

$$\alpha_{C_1 I_O} x_{I_1} + \beta_{C_1 I_O} x_{I_2} = (\alpha + \delta\alpha_{C_1 I_3} + \gamma\alpha_{C_1 I_4}) x_{I_1} + (\beta + \delta\beta_{C_1 I_3} + \gamma\beta_{C_1 I_4}) x_{I_2}.$$

For this equation to hold, the coefficients of $x_{I_1}$ and $x_{I_2}$ must be the same, producing two equations in the four coefficients, $\alpha, \beta, \delta, \gamma$:

$$\begin{aligned} \alpha_{C_1 I_O} &= \alpha + \delta\alpha_{C_1 I_3} + \gamma\alpha_{C_1 I_4}, \\ \beta_{C_1 I_O} &= \beta + \delta\beta_{C_1 I_3} + \gamma\beta_{C_1 I_4}. \end{aligned}$$

Going through exactly the same reasoning, but assuming that $x$ belongs to part 2, produces two more equations in the four coefficients:

$$\begin{aligned} \alpha_{C_2 I_O} &= \alpha + \delta\alpha_{C_2 I_3} + \gamma\alpha_{C_2 I_4}, \\ \beta_{C_2 I_O} &= \beta + \delta\beta_{C_2 I_3} + \gamma\beta_{C_2 I_4}. \end{aligned}$$

Now you have four equations in four unknowns, thus fully constraining $\alpha$, $\beta$, $\delta$, and $\gamma$. The only $\alpha, \beta, \delta, \gamma$ combination that works, no matter what, is the combination prescribed by those four equations.

Of course, none of this computation would do any good if you had to use the solutions to determine $\alpha, \beta, \delta, \gamma$, for then you would have to know all the particular $\alpha, \beta$ pairs appropriate to the individual parts and images.

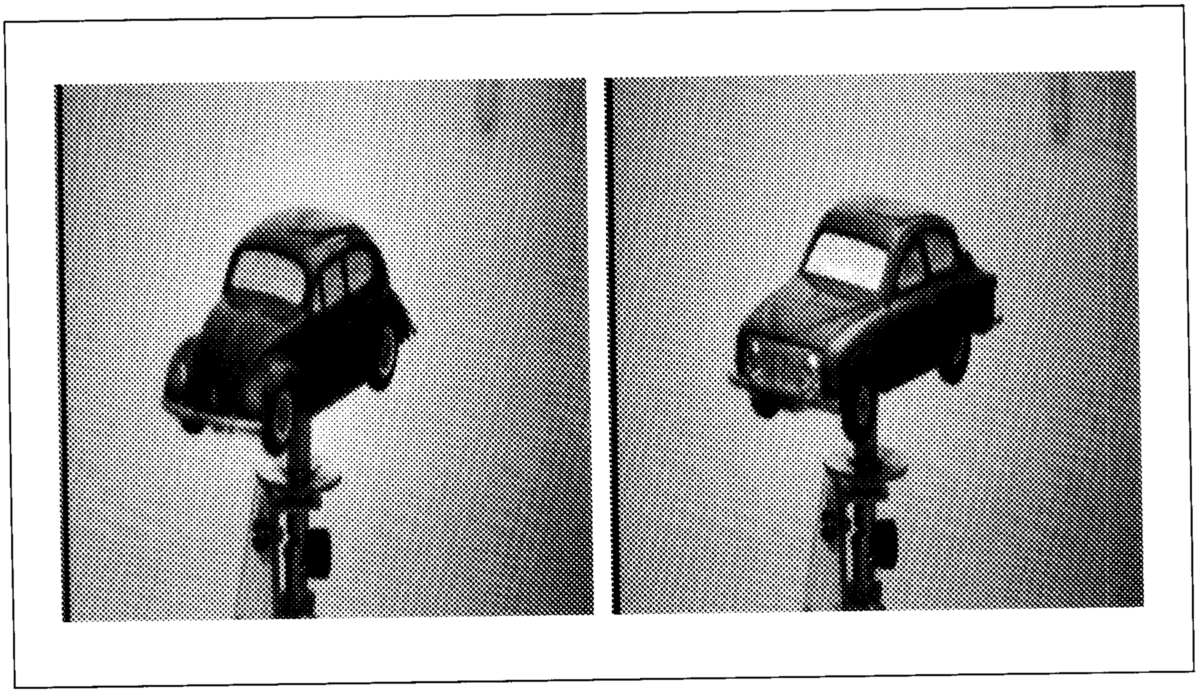


**Figure 26.8** Images of two unidentified automobiles. Images courtesy of Ronen Basri.

Fortunately, however, you can use the same idea that proved invaluable before: Given that you know that an appropriate set of coefficients exists, you can create a different set of equations, with easily obtained constants, by looking at four distinct points:

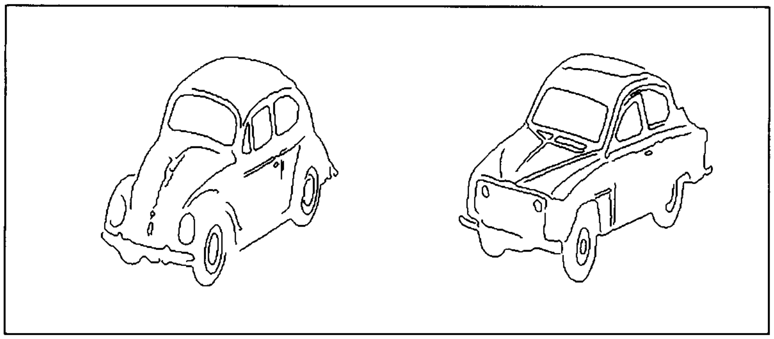
$$
\begin{aligned}
x_{P_1 I_o} &= \alpha x_{P_1 I_1} + \beta x_{P_1 I_2} + \delta x_{P_1 I_3} + \gamma x_{P_1 I_4},\\
x_{P_2 I_o} &= \alpha x_{P_2 I_1} + \beta x_{P_2 I_2} + \delta x_{P_2 I_3} + \gamma x_{P_2 I_4},\\
x_{P_3 I_o} &= \alpha x_{P_3 I_1} + \beta x_{P_3 I_2} + \delta x_{P_3 I_3} + \gamma x_{P_3 I_4},\\
x_{P_4 I_o} &= \alpha x_{P_4 I_1} + \beta x_{P_4 I_2} + \delta x_{P_4 I_3} + \gamma x_{P_4 I_4}.
\end{aligned}
$$

## The Template Approach Handles Complicated Curved Objects

To get a feeling for the power of the template approach, consider the two toy-automobile images shown in figure 26.8. To initiate identification, you have to reduce both to line drawings, as in figure 26.9.

Suppose that each automobile is known to be either an old-model Volkswagen or an old-model SAAB. To determine which it is, you need to match the drawings to constructed-to-order templates. Remarkably, the templates can be made straightforwardly, even though the objects have curved surfaces. All you need to do is to increase the number of images that constitute a model. Instead of the two model images needed for rotation about a vertical axis, you need three. Accordingly, you can form a Volkswagen model by taking three pictures and rendering those pictures as drawings. You can form a SAAB model the same way. Both are shown in figure 26.10.

**Figure 26.9** Drawings of two unidentified automobiles. Drawings courtesy of Ronen Basri.



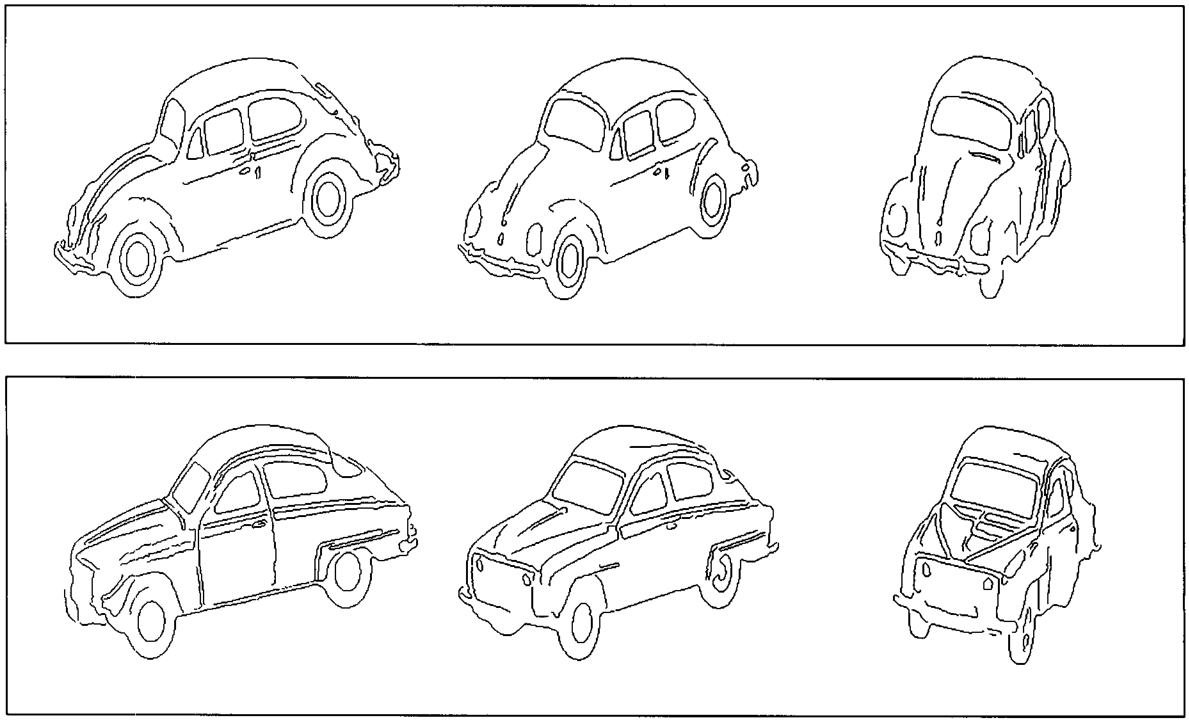


**Figure 26.10** Models consist of three drawings. Above is a Volkswagen model; below is a SAAB model. Drawings courtesy of Ronen Basri.

At the top of figure 26.11, the unidentified automobile on the left in figure 26.9 is shown together with a template manufactured for the drawing from Volkswagen images. As shown, the fit is just about perfect. At the bottom of figure 26.11, the unidentified automobile on the right in figure 26.9 is shown together with a template manufactured for the drawing from Volkswagen images. The fit is terrible, indicating that the drawing is not a drawing of a Volkswagen. Had the template been made from the SAAB model images, however, the fit would have been just about perfect.

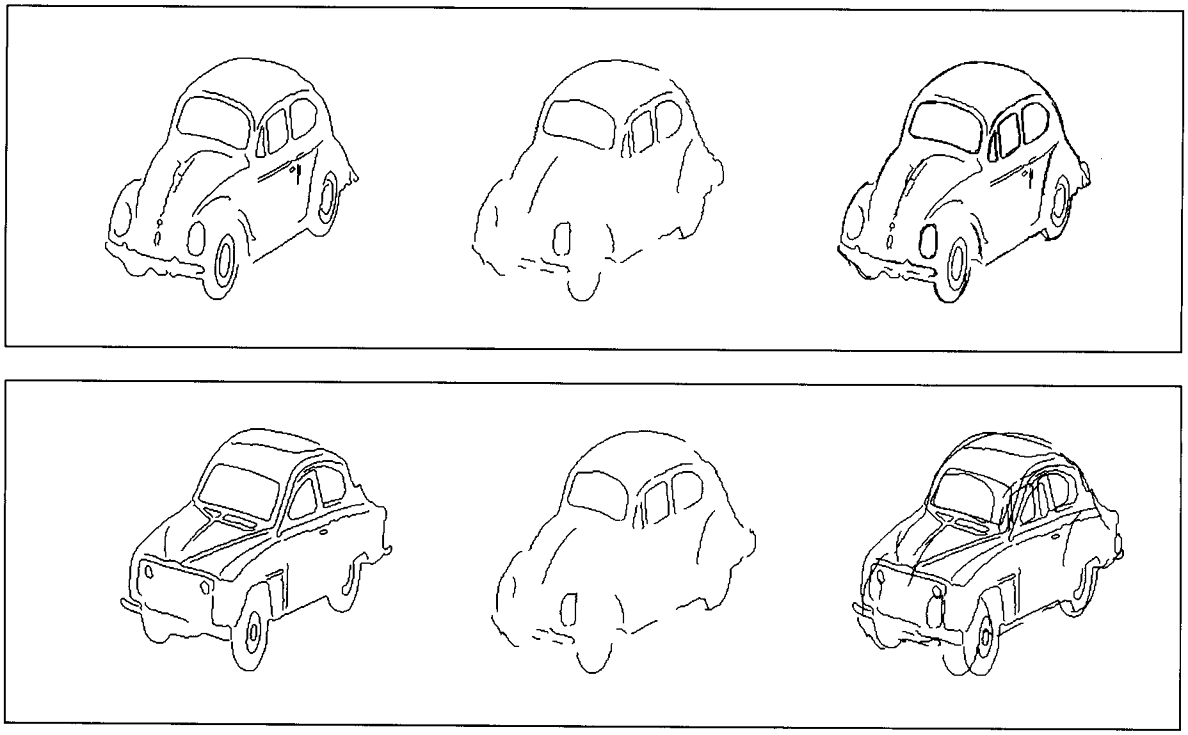


**Figure 26.11** Above is a drawing of a Volkswagen, a template manufactured from Volkswagen images, and the two images superimposed. Below is a drawing of an SAAB, a template manufactured from Volkswagen images, and the two superimposed. Drawings courtesy of Ronen Basri.

## ESTABLISHING POINT CORRESPONDENCE

In this section, you learn how it is possible to determine how the points in one image correspond to the points in another, a necessary prerequisite to template construction.

### Tracking Enables Model Points to Be Kept in Correspondence

Actually, it is relatively easy to create three images of an object, with known point correspondences, to serve as a model. All you need to do is to move the object slowly, taking many intermediate snapshots between each pair of images that is to be in the model. That way, the difference between adjacent snapshots is so small, corresponding points are always nearest neighbors, and you can track points from one model image to another through the intermediate snapshots, as suggested in figure 26.12.

### Only Sets of Points Need to Be Matched

When you are confronted with an unknown object, there can be no such thing as intermediate snapshots lying between the unknown's image and one of the model images. Consequently, matching the points is much harder.

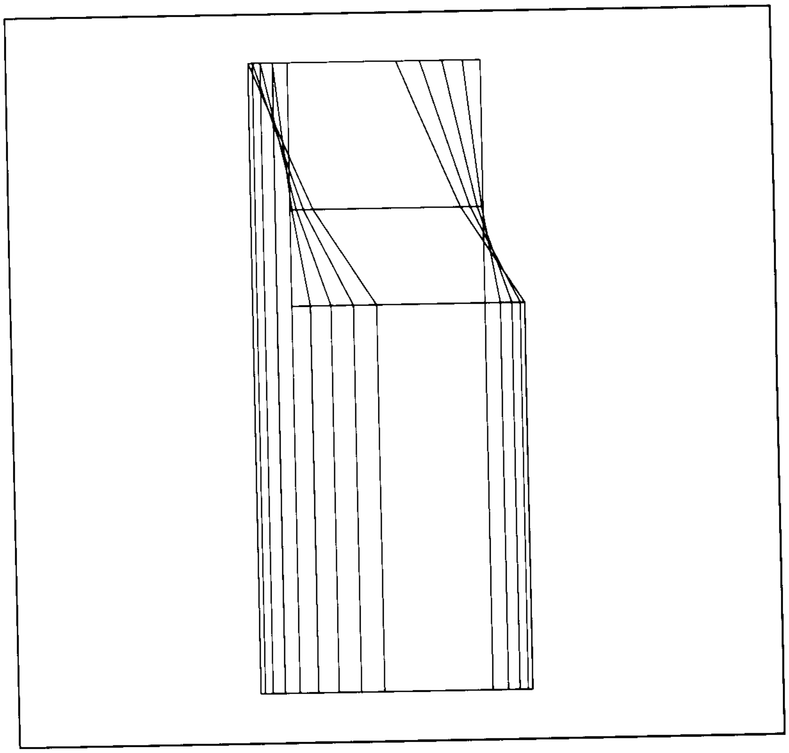


**Figure 26.12** Moving an object slowly makes it possible to track points from one model image to another through intermediate snapshots. Corresponding points are always nearest neighbors in adjacent snapshots, even though they are not necessarily nearest neighbors in the model images. Here five obelisk images are shown, with adjacent pairs rotated by 10°.

Accordingly, it is important to know that it is enough, in the general case, to establish that sets of points correspond, without knowing exactly how the points in the sets match up individually.

To see why, consider the one-axis-rotation special case again and recall that, for any point, $P_i$, the following equation holds:

$$x_{P_i I_o} = \alpha x_{P_i I_1} + \beta x_{P_i I_2}.$$

Accordingly, the equation must hold for points $P_1$ and $P_2$:

$$x_{P_1 I_o} = \alpha x_{P_1 I_1} + \beta x_{P_1 I_2},$$

$$x_{P_2 I_o} = \alpha x_{P_2 I_1} + \beta x_{P_2 I_2}.$$

Adding these equations produces the following:

$$(x_{P_1 I_o} + x_{P_2 I_o}) = \alpha(x_{P_1 I_1} + x_{P_2 I_1}) + \beta(x_{P_1 I_2} + x_{P_2 I_2}).$$

Repeating for two other points, $P_3$ and $P_4$, provides a second equation in the two unknowns, $\alpha$ and $\beta$:

$$(x_{P_3 I_o} + x_{P_4 I_o}) = \alpha(x_{P_3 I_1} + x_{P_4 I_1}) + \beta(x_{P_3 I_2} + x_{P_4 I_2}).$$

Note that, wherever $x_{P_1 I_o}$ appears, it is added to $x_{P_2 I_o}$, and vice versa. Similarly, wherever $x_{P_3 I_o}$ appears, it is added to $x_{P_4 I_o}$, and vice versa. Accordingly, there is no harm in confusing $x_{P_1 I_o}$ and $x_{P_2 I_o}$, and there is no harm in confusing $x_{P_3 I_o}$ and $x_{P_4 I_o}$. By adding up the $x$ coordinate values of the points in the sets, you eliminate the need to sort out exactly how the points in the sets correspond. It is enough to know that the points in one set are among the points in the corresponding set.

Thus, for the one-axis-rotation special case, you do not need to find two corresponding points in two images. You need only to find two corresponding sets of points in two images.

The argument generalizes easily. For arbitrary rotation, translation, and scaling, you do not need to find four corresponding points in three images. You need only to find four corresponding sets of points in three images.

## Heuristics Help You to Match Unknown Points to Model Points

Once you know that it is enough to find corresponding sets of points, you can use a few heuristics to help you find those corresponding sets. If an object has a natural top and bottom, for example, the points near the top and bottom in any image are likely to form corresponding sets.

Consider the model images and unknown shown in figure 26.13. As before, the 30° and 60° images constitute the model to be checked against the unknown. Earlier, you saw that you can use points $P_1$ and $P_2$ to find $\alpha$ and $\beta$ for the first unknown, using the following equations:

$$\begin{aligned}-2.83 &= \alpha(-2.73) + \beta(-2.73),\\ 0 &= \alpha(0.73) + \beta(-0.73).\end{aligned}$$

Any other pair of points would do just as well, of course. Instead of using $P_1$ and $P_2$, the points along the top, you could use any two of $P_5$, $P_6$, and $P_7$, the points along the bottom, which produce the following equations:

$$\begin{aligned}2.83 &= \alpha(2.73) + \beta(2.73),\\ 0 &= \alpha(-0.73) + \beta(0.73),\\ -2.83 &= \alpha(-2.73) + \beta(-2.73).\end{aligned}$$

Note that sums of equations are also equations. Thus, you can add together the equations for the top points to get a new equation. Then, you can add together the equations for the bottom points. Thus, you have two more equations in $\alpha$ and $\beta$:

$$\begin{aligned}(-2.83 + 0) &= \alpha(-2.73 + 0.73) + \beta(-2.73 - 0.73),\\ (2.83 + 0 - 2.83) &= \alpha(2.73 - 0.73 - 2.73) + \beta(2.73 + 0.73 - 2.73).\end{aligned}$$

Naturally, these new, composed equations have the same solution as do the original equations involving $P_1$ and $P_2$—namely, $\alpha = 0.518315$ and $\beta = 0.518315$. Thus, you can use corresponding sets of points to produce equations in $\alpha$ and $\beta$, instead of corresponding points. Within the corresponding sets, you do not need to know exactly which point goes with which point.

Still, even finding corresponding sets may be hard. For one thing, the general case requires four sets of corresponding points, not just two. Also,

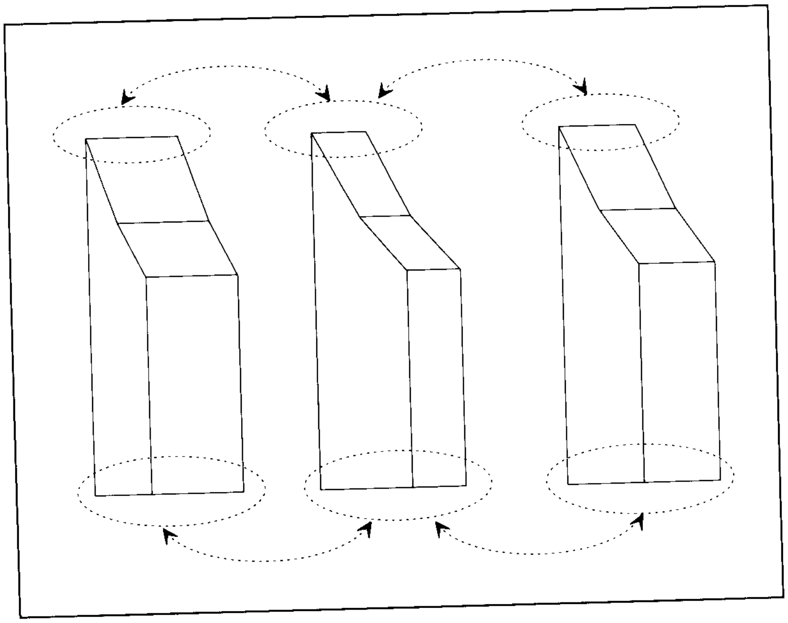


**Figure 26.13** It is sufficient to find sets of corresponding points; it is not necessary to know exactly how the points within the sets correspond.

you cannot always expect objects to have a natural standing posture that enables you to identify top and bottom sets.

## SUMMARY

- The traditional approach to object identification involves image description, followed by surface description, followed by volume description, followed by matching with library models.
- Amazingly, you can construct a two-dimensional identification template using the two-dimensional image of the object to be identified, plus a few stored two-dimensional image descriptions. Identification becomes a matter of template matching.
- You construct two-dimensional templates by using a few corresponding points to establish position-prediction coefficients.
- For pure rotation of a polyhedron around one axis, two corresponding points in two images are sufficient to establish position-prediction coefficients for an unknown object in a given image. For general polyhedron rotation and translation, you need four points in three images.
- When fully generalized, the template approach handles objects with parts and objects with curved surfaces.
- To use the template approach, you need to be able to identify corresponding features in image sets. Fortunately, you need only to find corresponding sets of points, rather than corresponding points. Sometimes, you can track points as they move, thus maintaining correspondence, rather than establishing correspondence.

## BACKGROUND

Identification procedures in which a template is generated and matched to an unknown are called alignment methods. The seminal alignment ideas described in this chapter were developed by Shimon Ullman and Ronen Basri [1989].