While the previous chapter concentrated on how low-level image processing is guided by models of scenes, this chapter describes how two-dimensional models are used for recognizing objects from image features. We start with a simple case where objects are isolated and image features are almost completely extracted. Then we consider cases where objects occlude one another and image features are not completely extracted. In the course of recognizing objects, features are modified or new features are extracted.

The cases are further classified into two categories according to the type of model used: models of images or models of objects. In the former case, segmentation of an image is not a critical problem because the whole image can be matched to models of scenes. In the latter case, however, the whole image cannot be directly matched to models. Therefore, segmentation of the image or image features is very important in order to recognize a part of an image using object models.

13.1 Recognition of Isolated Curved Objects Using a Graph Model

This section describes a method of recognizing a single curved object using the graph model described in Subsection 11.3.3 (Barrow and Popplestone, 1971). The scene includes an object on a table.

13.1.1 Scene Description

The image of the scene consists of $64 \times 64$ pixels with 16 levels. Uniform regions are extracted from the image by a region method (see Sect. 3.5). The properties of regions and the relations between them are calculated as described in Subsection 11.3.3. Thus the description of the scene being analyzed and the descriptions of models are similarly represented.

It is convenient to regard the properties of a region in terms of its relation with other identical regions. Then, the properties and the relations are represented in a uniform way. Many relations are calculated for regions either in the scene or in a model. All the relations are represented by numbers as shown in Fig. 11.5.

13.1.2 Evaluation of Matching

Let us denote regions in the scene by $S_1$, $S_2$, ..., $S_m$ and those in model $j$ by $M_{j,1}$, $M_{j,2}$, ..., $M_{j,m_j}$, where $m$ is the number of regions in the scene and $m_j$ is the
number of regions in model $j$ (generally $m$ is not equal to $m_j$). Recognition of objects is performed by matching the regions of the scene to those of the models. We need a suitable function for the evaluation of matching. The evaluation is based on the similarities between the relations in the scene and those in the model. Suppose that scene regions $S_1, S_2, \ldots, S_m$ are matched to model regions $M_{j,k_1}, M_{j,k_2}, \ldots, M_{j,k_m}$, respectively, and that the number of the region of the scene is equal to or less than that of the model ($m \leq m_j$). Then the correspondences between the relations in the scene and those in the models are established. Now let us consider how to evaluate the matching between a relation $R_{S}$ in the scene and the corresponding relation $R_{M}$ in the model. The principle is that if the difference between the values of the two relations is within a threshold, the two relations are considered to be matched. The threshold is decided experimentally to be $3 \sigma_M$, where $\sigma_M$ is the standard deviation of $R_M$.

The total evaluation function depends on the number of successive matchings ($n_\text{s}$), the number of relations ($n_r$), and the number of regions ($n_j$) in model $j$. The program tries to minimize the following evaluation function:

$$f = 1 - n_\text{s}/n_r + c/n_j$$

where a suitable value of the constant $c$ is 0.5.

### 13.1.3 Matching Strategy

Recognition of objects is now reduced to finding the match that minimizes Eq. (13.1). An exhaustive search, however, is not effective. A better way is to find correspondences one by one on the basis of the evaluation of partial matchings. In the process, when some promising partial matchings are obtained, the next step is to add one pair of regions to the most promising matching. This method is the best kind of first search. The algorithm is as follows:

1. Select an arbitrary region in the scene. Assuming the correspondence of the region to each region in the model, calculate the evaluation function. Store the partial matchings with the function values.
2. Select the best partial match (with the minimum evaluation function value). If all the regions in the scene are matched, then stop (the partial match is the answer).
3. Select one more region in the scene and add to the previously selected set of scene regions. Assuming the correspondence of the region to each of the unassigned regions in the model, calculate the evaluation function for the set of regions. Store the partial matching with the function values. Go to (2).

At an intermediate stage of this process, many partial matchings are stored along with their evaluation function values. Figure 13.1(a) shows an example of partial matchings stored at an intermediate stage (where $M_{j,1}$ is denoted by $M_1$ for convenience). Only the best three matchings are shown, with function values in increasing order. In step (2), the uppermost matching pair in the figure is selected, and in step (3) region $S_1$ is selected. Suppose the model contains only four regions. Then there are two regions ($M_1$ and $M_4$) which may correspond to region $S_3$. 