5.2 A LOCAL TECHNIQUE: BLOB COLORING

The counterpart to the edge tracker for binary images is the blob-coloring algorithm. Given a binary image containing four-connected blobs of 1's on a background of 0's, the objective is to "color each blob"; that is, assign each blob a different label. To do this, scan the image from left to right and top to bottom with a special L-shaped template shown in Fig. 5.1. The coloring algorithm is as follows.

Algorithm 5.1: Blob Coloring

Let the initial color, $k = 1$. Scan the image from left to right and top to bottom.

If $f(\mathbf{x}_C) = 0$ then continue
else
    begin

    if $(f(\mathbf{x}_U) = 1$ and $f(\mathbf{x}_L) = 0)$
    then color $(\mathbf{x}_C) :=$ color $(\mathbf{x}_U)$

    if $(f(\mathbf{x}_L) = 1$ and $f(\mathbf{x}_U) = 0)$
    then color $(\mathbf{x}_C) :=$ color $(\mathbf{x}_L)$

    if $(f(\mathbf{x}_L) = 1$ and $f(\mathbf{x}_U) = 1)$
    then begin
        color $(\mathbf{x}_C) :=$ color $(\mathbf{x}_L)$
        color $(\mathbf{x}_L)$ is equivalent to color $(\mathbf{x}_U)$
        end

  comment: two colors are equivalent.

    if $(f(\mathbf{x}_L) = 0$ and $f(\mathbf{x}_U) = 0)$
    then color $(\mathbf{x}_L) := k$; $k := k + 1$

    comment: new color

    end

After one complete scan of the image the color equivalences can be used to assure that each object has only one color. This binary image algorithm can be used as a simple region-grower for gray-level images with the following modifications. If in a

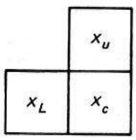

Fig. 5.1 L-shaped template for blob coloring.

gray-level image $f(\mathbf{x}_C)$ is approximately equal to $f(\mathbf{x}_U)$, assign $\mathbf{x}_C$ to the same region (blob) as $\mathbf{x}_U$. This is equivalent to the condition $f(\mathbf{x}_C) = f(\mathbf{x}_U) = 1$ in Algorithm 5.1. The modifications to the steps in the algorithm are straightforward.

5.3 GLOBAL TECHNIQUES: REGION GROWING VIA THRESHOLDING

This approach assumes an object-background image and picks a threshold that divides the image pixels into either object or background:

> $\mathbf{x}$ is part of the Object iff $f(\mathbf{x}) > T$
> Otherwise it is part of the Background

The best way to pick the threshold T is to search the histogram of gray levels, assuming it is bimodal, and find the minimum separating the two peaks, as in Fig. 5.2. Finding the right valley between the peaks of a histogram can be difficult when the histogram is not a smooth function. Smoothing the histogram can help but does not guarantee that the correct minimum can be found. An elegant method for treating bimodal images assumes that the histogram is the sum of two composite normal functions and determines the valley location from the normal parameters [Chow and Kaneko 1972].

The single-threshold method is useful in simple situations, but primitive. For example, the region pixels may not be connected, and further processing such as that described in Chapter 2 may be necessary to smooth region boundaries and remove noise. A common problem with this technique occurs when the image has a background of varying gray level, or when collections we would like to call regions vary smoothly in gray level by more than the threshold. Two modifications of the threshold approach to ameliorate the difficulty are: (1) high-pass filter the image to deemphasize the low-frequency background variation and then try the original technique; and (2) use a spatially varying threshold method such as that of [Chow and Kaneko 1972].

The Chow-Kaneko technique divides the image up into rectangular subimages and computes a threshold for each subimage. A subimage can fail to have a threshold if its gray-level histogram is not bimodal. Such subimages receive inter-

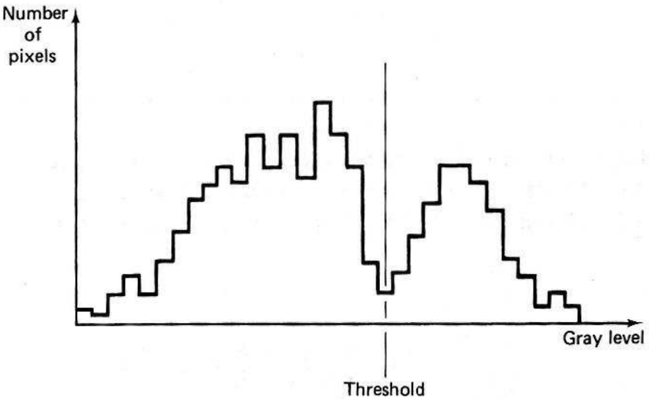

Fig. 5.2 Threshold determination from gray-level histogram.