11.2.2 Combining Region Labeling and Contour Finding

This method, based on [23], combines the concepts of sequential region labeling (Sec. 11.1) and traditional contour tracing into a single algorithm able to perform both tasks simultaneously during a single pass through the image. It identifies and labels regions and at the same time traces both their inner and outer contours. The algorithm does not require complicated data structures and is very efficient when compared with other methods with similar capabilities.

We now sketch the fundamental idea of the algorithm. While the main idea of the algorithm can be sketched out in a few simple steps, the actual implementation requires attention to a number of details, so we have provided the complete Java source for an ImageJ plugin implementation in Appendix D (pp. 532–542). The most important steps of the method are illustrated in Fig. 11.9:

1. As in the sequential region labeling (Alg. 11.2), the binary image I is traversed from the top left to the bottom right. Such a traversal ensures that all pixels in the image are eventually examined and assigned an appropriate label.
2. At a given position in the image, the following cases may occur:

   Case A: The transition from a background pixel to a previously unmarked foreground pixel A means that A lies on the outer edge of a new region. A new *label* is allocated and the associated *outer* contour is traversed and marked by calling the method TraceContour() (see Fig. 11.9 (a) and Alg. 11.3 (line 20)). Furthermore, all background pixels directly bordering the region are marked with the value -1.

   Case B: The transition from a foreground pixel B to an unmarked background pixel means that B lies on the edge of an *inner* contour (Fig. 11.9 (b)). Starting from B, the inner contour is traversed and its pixels are labeled with labels from the surrounding region (Fig. 11.9 (c)). Also, all bordering background pixels are again assigned the value of -1.

   Case C: When a foreground pixel does not lie on a contour (i. e., it is not on an edge), then the neighboring pixel to the left has already been labeled (Fig. 11.9 (d)) and this label is propagated to the current pixel.

In Algs. 11.3 and 11.4, the entire procedure is presented again and explained precisely. The method CombinedContourLabeling() traverses the image line-by-line and calls the method TraceContour() whenever a new inner or outer contour must be traced. The labels of the image elements along the contour, as well as the neighboring foreground pixels, are stored in the "label map" LM by the method FindNextPoint() (Alg. 11.4).

Burger, Burge: Digital Image Processing – An Algorithmic Introduction Using Java
© Springer-Verlag, 2008. www.imagingbook.com