10.5 Morphological Reconstruction

Reconstruction is a morphological transformation involving two images and a structuring element (instead of a single image and structuring element). One image, the *marker*, is the starting point for the transformation. The other image, the *mask*, constrains the transformation. The structuring element used defines connectivity. In this section we use 8-connectivity (the default), which implies that \( B \) in the following discussion is a \( 3 \times 3 \) matrix of 1s, with the center defined at coordinates \((2, 2)\). In this section we deal with binary images; gray-scale reconstruction is discussed in Section 10.6.3.

If \( G \) is the mask and \( F \) is the marker, the reconstruction of \( G \) from \( F \), denoted \( R_G(F) \), is defined by the following iterative procedure:

1. Initialize \( h_1 \) to be the marker image, \( F \).
2. Create the structuring element: \( B = \text{ones}(3) \).
3. Repeat:
   \[
   h_{k+1} = (h_k \oplus B) \cap G
   \]
   until \( h_{k+1} = h_k \).
4. \( R_G(F) = h_{k+1} \).

Marker \( F \) must be a subset of \( G \):

\[
F \subseteq G
\]

Figure 10.21 illustrates the preceding iterative procedure. Although this iterative formulation is useful conceptually, much faster computational algorithms exist. Toolbox function `imreconstruct` uses the “fast hybrid reconstruction” algorithm described in Vincent [1993]. The calling syntax for `imreconstruct` is

\[
\text{out} = \text{imreconstruct}(\text{marker, mask})
\]

where `marker` and `mask` are as defined at the beginning of this section.

10.5.1 Opening by Reconstruction

In morphological opening, erosion typically removes small objects, and the subsequent dilation tends to restore the shape of the objects that remain. However, the accuracy of this restoration depends on the similarity between the shapes and the structuring element. The method discussed in this section, *opening by reconstruction*, restores the original shapes of the objects that remain after erosion. The opening by reconstruction of an image \( G \) using structuring element \( B \), is defined as \( R_G(G \ominus B) \).

**EXAMPLE 10.8:** Opening by reconstruction.

A comparison between opening and opening by reconstruction for an image containing text is shown in Fig. 10.22. In this example, we are interested in extracting from Fig. 10.22(a) the characters that contain long vertical strokes.
Because both opening and opening by reconstruction have erosion in common, we perform that step first, using a thin, vertical structuring element of length proportional to the height of the characters:

```matlab
>> f = imread('book_text_bw.tif');
>> fe = imerode(f, ones(51, 1));
```

Figure 10.22(b) shows the result. The opening, shown in Fig. 10.22(c), is computed using `imopen`:

```matlab
>> fo = imopen(f, ones(51, 1));
```

Note that the vertical strokes were restored, but not the rest of the characters containing the strokes. Finally, we obtain the reconstruction:

```matlab
>> fobr = imreconstruct(fe, f);
```
Morphological Image Processing

Figure 9.29  (a) Text image of size $918 \times 2018$ pixels. The approximate average height of the tall characters is 50 pixels. (b) Erosion of (a) with a structuring element of size $51 \times 1$ pixels. (c) Opening of (a) with the same structuring element, shown for reference. (d) Result of opening by reconstruction.
Morphological Image Processing

Segmentation of nontrivial images is one of the more difficult tasks in image processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, certain image processing techniques can be taken to improve the probability of a correct segmentation. Such techniques include image enhancing, such as industrial inspection applications, at least some aspects of the environment are possible at times. The experienced designer invariably pays considerable attention to such aspects of the problem. There is no point past the level of detail required to identify those aspects of the problem. There is no point past the level of detail required to identify those

**Figure 9.31**

(a) Text image of size 918 × 2018 pixels. (b) Complement of (a) for use as a mask image. (c) Marker image. (d) Result of hole-filling using Eq. (9.5-29).
The result in Fig. 10.22(d) shows that characters containing long vertical strokes were restored exactly; all other characters were removed. The remaining parts of Fig. 10.22 are explained in the following two sections.

10.5.2 Filling Holes

Morphological reconstruction has a broad spectrum of practical applications, each characterized by the selection of the marker and mask images. For example, let \( I \) denote a binary image and suppose that we choose the marker image, \( F \), to be 0 everywhere except on the image border, where it is set to \( 1 - I \):

\[
F(x, y) = \begin{cases} 
1 - I(x, y) & \text{if } (x, y) \text{ is on the border of } I \\
0 & \text{otherwise}
\end{cases}
\]

Then,

\[
H = \left[ R_f(F) \right]
\]

is a binary image equal to \( I \) with all holes filled, as illustrated in Fig. 10.22(e).
Toolbox function `imfill` performs this computation automatically when the optional argument `'holes'` is used:

\[ g = \text{imfill}(f, \text{'holes'}) \]

This function is discussed in more detail in Section 12.1.2.

**10.5.3 Clearing Border Objects**

Another useful application of reconstruction is removing objects that touch the border of an image. Again, the key task is to select the appropriate marker to achieve the desired effect. Suppose we define the marker image, \( F \), as

\[ F(x, y) = \begin{cases} I(x, y) & \text{if } (x, y) \text{ is on the border of } I \\ 0 & \text{otherwise} \end{cases} \]

where \( I \) is the original image. Then, using \( I \) as the mask image, the reconstruction

\[ H = R_I(F) \]

yields an image, \( H \), that contains only the objects touching the border, as Fig. 10.22(f) shows. The difference, \( 1 - H \), shown in Fig. 10.22(g), contains only the objects from the original image that do not touch the border. Toolbox function `imclearborder` performs this entire procedure automatically. Its syntax is

\[ g = \text{imclearborder}(f, \text{conn}) \]

where \( f \) is the input image and \( g \) is the result. The value of \( \text{conn} \) can be either 4 or 8 (the default). This function suppresses structures that are lighter than their surroundings and that are connected to the image border.