Morphological Image Processing

Ranga Rodrigo

October 9, 2009

Outline

Contents

1 Preliminaries 1

2 Dilation and Erosion 3

2.1 Dilation 3

2.2 Erosion 5

3 Opening and Closing 5

4 Labeling Connected Components 11

Morphological image processing tools help us to extract image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and convex hull. There are morphological techniques for pre- and post-processing, such as morphological filtering, thinning, and pruning.

1 Preliminaries

Let $Z$ be a set of integers. The sampling process used to produce the digital image may be viewed as a partitioning in the $xy$-plane into a grid, with coordinates of the center of each grid being a pair of elements from the Cartesian product $Z^2$. In the terminology of set theory the function $f(x, y)$ is said to be a digital image if $(x, y)$ are integers from $Z^2$ and $f$ is a mapping that assigns an intensity value in the set of real numbers, $R$, to each distinct pair of coordinates $(x, y)$. If the elements of $R$ also are integers, a digital image then becomes a two-dimensional function, whose coordinates and the amplitude values are integers.

Let $A$ be a set in $Z^2$, the elements of which are pixel coordinates $(x, y)$. If $w = (x, y)$ is an element of $A$, then we write

$$w \in A.$$ 

Similarly, if $w$ is not an element of $A$, we write

$$w \notin A.$$ 

A set of $B$ pixels satisfying a particular condition is written as

$$B = \{w|\text{condition}\}.$$
Figure 1: Set operations.

For example, the set of all pixel coordinates that do not belong to \( A \), denoted by \( A' \), is given by

\[
A' = \{ w | w \notin A \}.
\]

This set is called the *complement* of \( A \). The *union* of two sets, denoted by

\[
C = A \cup B
\]

is the set of all elements that belong to either \( A \) or \( B \), or both. Similarly, the *intersection* of two sets \( A \) and \( B \) is the set of elements that belong to both sets, denoted by

\[
C = A \cap B.
\]

The *difference* of sets \( A \) and \( B \), denoted by \( A - B \), is the set of all elements that belong to \( A \) but not to \( B \):

\[
A - B = \{ w | w \in A, w \notin B \}.
\]

Figure 1 illustrates these set operations.

In addition to the basic operations, morphological operations often require two operations that are specific to sets whose elements are pixel coordinates. The *reflection* of set \( B \), denoted by \( \hat{B} \), is defined as

\[
\hat{B} = \{ w | w = -b, \forall b \in B \}.
\]

The *translation* of set \( A \) by \( z = (z_x, z_y) \), denoted by \( (A)_z \), is defined as

\[
(A)_z = \{ c | c = a + z, \forall a \in A \}.
\]

A binary image can be viewed as a bivariate function of \( x \) and \( y \). Morphological theory views a binary image as a set of foreground (1-valued) pixels, the elements of which are in \( \mathbb{Z}^2 \). Set operations such as union and intersection can be applied directly to binary image sets. For example, if \( A \) and \( B \) are binary images, then \( C = A \cup B \) is also a binary image, where a pixel in \( C \) is a foreground pixel if either or both of the corresponding pixels in \( A \) and \( B \) are foreground pixels. Table 1 shows Matlab’s logical operations that perform set operations on binary images.
The operations of *dilation* and *erosion* are fundamental to morphological image processing. Many of the algorithms are based on these operations.

### 2.1 Dilation

Dilation is an operation that grows or thickens objects in a binary image. The specific manner and the extent of this thickening is controlled by a shape referred to as a *structuring element*. Dilation is a process that translates the origin of the structuring element throughout the domain of the image and checks to see whether it overlaps with 1-valued pixels. The output image is 1 at each location of the origin of the structuring element if the structuring element overlaps at least one 1-valued pixel in the input image.

Mathematically, dilation is defined in terms of set operations. The dilation of $A$ by $B$, denoted by $A \oplus B$, is defined as

$$A \oplus B = \{ z | (\hat{B})_z \cap A \neq \emptyset \},$$

where $\emptyset$ is the empty set and $B$ is the structuring element. In words, the dilation of $A$ by $B$ is the set consisting of all the structuring element origin locations where the reflected and translated $B$ overlaps at least some portion of $A$. Dilation is commutative, i.e., $A \oplus B = B \oplus A$. It is a convention in image processing to let the first operand of $A \oplus B$ be the image and the second operand be the structuring element, which is usually much smaller than the image.

The Matlab image processing toolbox function `imdilate` performs dilation. The syntax is

$$C = \text{imdilate}(A, B)$$

where $A$ and $C$ are binary images and $B$ is a matrix of 0s and 1s that specify the structuring element.
Example 1. Write a Matlab program to dilate the image $A$ with the structuring element $B$ shown below.

```
0 1 0
1 1 1
0 1 0
```

**Structuring Element Decomposition**

Dilation is associative. That is

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C.$$  

Suppose that structuring element $B$ can be represented as a dilation of two structuring elements $B_1$ and $B_2$:

$$B = B_1 \oplus B_2.$$  

Then, $A \oplus B = A \oplus (B_1 \oplus B_2) = (A \oplus B_1) \oplus B_2$. In other words, dilating $A$ with $B$ is the same as first dilating $A$ with $B_1$, and then dilating the result with $B_2$. We say that $B$ can be decomposed into the structuring elements $B_1$ and $B_2$.

The associative property is important because the time required to compute dilation is proportional to the number of nonzero pixels in the structuring element. Consider for example, dilation with a $5 \times 5$ array of 1s:

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

This structuring element can be decomposed into a five-element row of 1s and a file-element column of 1s:

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix} \oplus 
\begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
1 \\
\end{bmatrix}.
$$

The number of elements in the original structuring element is 25, but the total number of elements in the row-column decomposition is only 10. This means that dilation with the row structuring element first, followed by dilation with the column structuring element can be performed approximately 2.5 times faster that dilation with the $5 \times 5$ array of 1s.
The strel Function
Matlab image processing toolbox function strel constructs structuring elements with a variety of shapes and sizes. Its basic syntax is

\[ B = \text{strel}(\text{shape}, \text{parameters}) \]

This returns a special quantity called an \textit{strel} object. The function imdilate is able to use the decomposed form of the structuring element.

Example:

\[ B = \text{strel}(\text{'diamond'}, 5) \]

creates a flat diamond-shaped structuring element such that the distance from the structuring element's origin to the extreme points of the diamond is 5. There are other shapes such as disk, line, rectangle, etc. The following code shows how to use the structuring element generated by the strel function is used to carry out dilation.

```matlab
A = imread('image.tif');
B = strel('diamond', 5);
C = imdilate(A, B);
imshow(C)
```

2.2 Erosion
Erosion shrinks or thins in a binary image. As in dilation, the manner and extent of shrinking is controlled by the structuring element. The output of erosion has a value 1 at each location of the origin of the structuring element, such that the structuring element overlaps only 1-valued pixels of the input image.

The mathematical definition of erosion is similar to that of dilation. The erosion of \( A \) by \( B \), denoted by \( A \ominus B \), is defined as

\[ A \ominus B = \{z \mid (B)_z \cap A' \neq \emptyset \}. \]

In other words, erosion of \( A \) by \( B \) is the set of structuring element origin locations where the translated \( B \) has no overlap with the background of \( A \).

The Matlab image processing toolbox function imerode performs erosion. The syntax is

\[ C = \text{imerode}(A, B) \]

where \( A \) and \( C \) are binary images and \( B \) is a matrix of 0s and 1s that specify the structuring element.

3 Opening and Closing
In practical image processing applications, dilation and erosion are used most often in various combinations. We can apply a series of dilation and erosion operations to an image, either using the same structuring element or, sometimes, a different one. Two such common operations are \textit{opening} and \textit{closing}. 
Morphological Opening

The morphological opening of $A$ by $B$, denoted by $A \circ B$, is simply erosion of $A$ by $B$ followed by the dilation of the result by $B$:

$$A \circ B = (A \ominus B) \oplus B.$$ 

An alternative mathematical formulation of opening is

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

where $\bigcup \{ \cdot \}$ denotes the union of all sets inside braces, and the notation $C \subseteq D$ means that $C$ is a subset of $D$. This formulation has a simple geometric interpretation: $A \circ B$ is the union of all translations of $B$ that fit entirely within $A$.

Morphological opening removes completely regions of an object that cannot
contain the structuring element, smoothes the object contours, breaks thin connections, and removes the protrusions.

**Morphological Closing**

The *morphological closing* of $A$ by $B$, denoted by $A \bullet B$, is a dilation followed by an erosion:

$$A \bullet B = (A \otimes B) \ominus B.$$  

Geometrically, $A \bullet B$ is the complement of the union of all translations of $B$ the do not overlap $A$.
**Matlab Functions for Opening and Closing**

Opening and closing are implemented in the toolbox with functions `imopen` and `imclose`. The following carries out opening:

```matlab
C = imopen(A, B)
```

Closing is carried out as follows:

```matlab
C = imclose(A, B)
```

In these code snippets, A is a binary image, and B is matrix of 0s and 1s that specifies the structuring element. A strel object, SE can be used instead of B.

*Example 2.* Write a Matlab program to carry out morphological opening on an image using a square structuring element.
Example 3. Write a Matlab program to carry out morphological closing on an image using a square structuring element.

Combining a closing and opening can be quite effective in removing noise.
Function `bwmorph`

Image processing toolbox function `bwmorph` implements a variety of useful operations based on combinations of dilations, erosions, and lookup table operations.

\[ g = \text{bwmorph}(f, \text{operation}, n) \]

where \( f \) is an input binary image, `operation` is a string specifying the desired operation, and \( n \) is a positive integer specifying the number of times the operation is to be repeated. The operation can be open, close, dilate, erode, thin, skel, and several others.

Thinning, (thin) means reducing binary objects or shapes in an image to strokes that are a single pixel wide. Skeletonisation (skel) is another way to reduce binary image objects to a set of thin strokes that retain the important information about the shapes of the original objects.

*Example 4.* Write a Matlab program to carry out skeletonisation.
4 Labeling Connected Components

Using connected component analysis we can extract individual objects or connected components in an image. In order to do this, we must define connectedness.

A pixel $p$ at coordinates $(x_0, y_0)$ has two horizontal and two vertical neighbors whose coordinates are. This set is known as 4-neighbors of $p$, denoted by $N_4(p)$. In a similar manner we can define $N_8(p)$ as well. Figure 3 shows this.

Two pixels $p$ and $q$ are said to be 4-adjacent if $q \in N_4(p)$. Similarly, two pixels $p$ and $q$ are said to be 4-adjacent if $q \in N_8(p)$. A path between pixels $p_1$ and $p_n$ is a sequence of pixels $p_1, p_2, \ldots, p_n$ such that $p_k$ is adjacent to $p_{k+1}$, for $1 \leq k < n$. A path can be 4-connected or 8-connected depending on the definition of adjacency used.

Two foreground pixels $p$ and $q$ are said to be 4-connected is there exists a 4-connected path between them, containing entirely foreground pixels. Similarly, two foreground pixels $p$ and $q$ are said to be 8-connected is there exists a 8-connected path between them, containing entirely foreground pixels.

**Example 5.** Identify if the pixels $p$ and $q$ are 4-connected or 8-connected.
**Connected Components**

For any foreground pixel \( p \), the set of all foreground pixels connected to it is called the *connected component* containing \( p \). The nature of the connected component depends on the definition of adjacency we choose between 4- and 8-adjacency.

**IPT Function bwlabel**

Matlab Image Processing Toolbox Function `bwlabel` computes all the connected components in a binary image.

\[
[L, \text{num}] = \text{bwlabel}(f, \text{conn})
\]

where \( f \) is a binary image and \( \text{conn} \) specifies the desired connectivity (either 4 or 8). Output \( L \) is called the label matrix, and \( \text{num} \) (optional) gives the total number of connected components found.