Morphological Image Processing

April 29, 2005

Szabolcs Sergyán
sergyan.szabolcs@nik.bmf.hu

BMF NIK
IPT function \texttt{imdilate} performs dilation. Its basic calling syntax is

\begin{equation}
A2 = \texttt{imdilate}(A,B)
\end{equation}

where \(A\) and \(A2\) are binary images, and \(B\) is a matrix of 0s and 1s that specifies the structuring element.
Dilation

>> A=imread('broken-text.tif');
>> B=[0 1 0;1 1 1;0 1 0];
>> A2=imdilate(A,B);

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.
IPT function `strel` constructs structuring elements with a variety of shapes and sizes. Its basic syntax is

\[
se = \text{strel} \left( \text{shape}, \text{parameters} \right)
\]

where `shape` is a string specifying the desired shape, and `parameters` is a list of parameters that specify information about the shape, such as its size.
# Structuring Element

<table>
<thead>
<tr>
<th>Syntax Forms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>strel('diamond',R)</code></td>
<td>Creates a flat, diamond-shaped structuring element, where <code>R</code> specifies the distance from the structuring element origin to the extreme points of the diamond.</td>
</tr>
<tr>
<td><code>strel('disk',R)</code></td>
<td>Creates a flat, disk-shaped structuring element with radius <code>R</code>.</td>
</tr>
<tr>
<td><code>strel('line',LEN,DEG)</code></td>
<td>Creates a flat, linear structuring element, where <code>LEN</code> specifies the length, and <code>DEG</code> specifies the angle (in degrees) of the line, as measured in a counterclockwise direction from the horizontal axes.</td>
</tr>
<tr>
<td><code>strel('octagon',R)</code></td>
<td>Creates a flat, octagonal structuring element, where <code>R</code> specifies the distance from the structuring element origin to the sides of the octagon, as measured along the horizontal and vertical axes. <code>R</code> must be a nonnegative multiple of 3.</td>
</tr>
</tbody>
</table>
## Structuring Element

<table>
<thead>
<tr>
<th>Syntax Forms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>strel('pair',OFFSET)</code></td>
<td>Creates a flat structuring element containing two members. One member is located at the origin. The second member's location is specified by the vector <code>OFFSET</code>, which must be a two-element vector of integers.</td>
</tr>
<tr>
<td><code>strel('periodicline',P,V)</code></td>
<td>Creates a flat structuring element containing $2 \times P + 1$ members. $V$ is a two-element vector containing integer-valued row and column offsets. One structuring element member is located at the origin. The other members are located at $1 \times V$, $-1 \times V$, $2 \times V$, $-2 \times V$, ..., $P \times V$, and $-P \times V$.</td>
</tr>
</tbody>
</table>
# Structuring Element

<table>
<thead>
<tr>
<th>Syntax Forms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>strel('rectangle', MN)</code></td>
<td>Creates a flat, rectangle-shaped structuring element, where <code>MN</code> specifies the size. <code>MN</code> must be a two-element vector of nonnegative integers. The first element of <code>MN</code> is the number of rows in the structuring element; the second element is the number of columns.</td>
</tr>
<tr>
<td><code>strel('square', W)</code></td>
<td>Creates a square structuring element whose width is <code>W</code> pixels. <code>W</code> must be a nonnegative integer scalar.</td>
</tr>
<tr>
<td><code>strel(NHOOD)</code></td>
<td>Creates a structuring element of arbitrary shape. <code>NHOOD</code> is a matrix of 0s and 1s that specifies the shape.</td>
</tr>
</tbody>
</table>
Dilation

>> originalI=imread('cameraman.tif');
>> se=strel('disk',2);
>> dilatedI=imdilate(originalI,se);
Erosion

>> A = imread('wirebond-mask.tif');
>> se = strel('disk', 10);
>> A2 = imerode(A, se);
>> se = strel('disk', 5);
>> A3 = imerode(A, se);
>> A4 = imerode(A, strel('disk', 20));
>> subplot(2, 2, 1), imshow(A),...
   subplot(2, 2, 2), imshow(A2),...
   subplot(2, 2, 3), imshow(A3),...
   subplot(2, 2, 4), imshow(A4)
Erosion
IPT function `bwlabel` computes all the connected components in a binary image. The calling syntax is

```
[L, num] = bwlabel(f, conn)
```

where `f` is an input binary image and `conn` specifies the desired connectivity (either 4 or 8). Output `L` is called a *label matrix*, and `num` (optional) gives the total number of connected components found. If parameter `conn` is omitted, its value defaults to 8.
Labeling Connected Components

```matlab
>> f = imread('ten-objects.tif');
>> [L, n] = bwlabel(f);
>> [r, c] = find(L == 3);
>> rbar = mean(r);
>> cbar = mean(c);
>> imshow(f)
>> hold on
>> for k = 1:n
    [r, c] = find(L == k);
    rbar = mean(r);
    cbar = mean(c);
    plot(cbar, rbar, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 10)
    plot(cbar, rbar, 'Marker', '*', 'MarkerEdgeColor', 'w')
>> end
```
Labeling Connected Components
References

- http://www.imageprocessingplace.com