Lecture 7: Morphological Image Processing

Prof. YingLi Tian
Oct. 25, 2017

Department of Electrical Engineering
The City College of New York
The City University of New York (CUNY)

Thanks to G&W website and Dr. Shahram Ebadollahi for slide materials
Outline

- What is Mathematical Morphology?
- Background Notions
- Introduction to Set Operations on Images
- Basic operation
  - Erosion, Dilation, Opening, Closing, Hit-or-Miss
- Applications
  - Boundary detection, skeleton,
- Morphological operations on gray-level images
What is Mathematical Morphology?

An (imprecise) Mathematical Answer

A mathematical tool for investigating geometric structure in binary and grayscale images.

Shape Processing and Analysis

Visual perception requires transformation of images so as to make explicit particular shape information.

**Goal:** Distinguish meaningful shape information from irrelevant one.

The vast majority of shape processing and analysis techniques are based on designing a shape operator which satisfies desirable properties.
Morphological Image Processing

- Started in 1960s by G. Matheron and J. Serra
- Analysis of form and structure of objects
- Tools/Operations for describing/characterizing image regions and image filtering
- Images are treated as sets
Morphological Image Processing

\[ g = \Psi[f] \]

Image-to-image transform
Applications - filtering

1. Removal of small blobs

2. Extraction and grouping of linear objects
Applications – segmentation 1

1. Separation of connected blobs

5. Extraction of grid lines
Applications – segmentation 2
Applications - quantification

1. Pattern spectrum or granulometries

2. Analysis of directions
Background Notions: Image as a Set

**FIGURE 9.3** (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.
Set Operations on Images

(a) Two sets $A$ and $B$. (b) The union of $A$ and $B$. (c) The intersection of $A$ and $B$. (d) The complement of $A$. (e) The difference between $A$ and $B$. 

$A \cup B$ 

$A \cap B$ 

$(A)'$ 

$A - B$
Set Operations on Images

Translation

\[(A)_z = \{w \mid w = a + z, \text{ for } a \in A}\]

Translates the origin of \(A\) to point \(z\).

Reflection

\[\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}\]

Reflects all elements of \(B\) about the origin of this set.

**FIGURE 9.2**
(a) Translation of \(A\) by \(z\).
(b) Reflection of \(B\). The sets \(A\) and \(B\) are from Fig. 9.1.
Set Operations on Images -
Union & Intersection

**Union**

\[(f \lor g)(x) = \max[f(x), g(x)]\]  
\[SG(f \lor g) = SG(f) \cup SG(g)\]

**Intersection**

\[(f \land g)(x) = \min[f(x), g(x)]\]  
\[SG(f \land g) = SG(f) \cap SG(g)\]

---

(a) 1-D signals \(f\) and \(g\).

(b) Point-wise maximum \(\lor\) and point-wise minimum \(\land\).

\[(\Psi_1 \lor \Psi_2)(f) = \Psi_1(f) \lor \Psi_2(f)\]

\[(\Psi_1 \land \Psi_2)(f) = \Psi_1(f) \land \Psi_2(f)\]
### Set Operations on Images – Matlab functions

<table>
<thead>
<tr>
<th>Set Operation</th>
<th>MATLAB Expression for Binary Images</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \cap B$</td>
<td>A &amp; B</td>
<td>AND</td>
</tr>
<tr>
<td>$A \cup B$</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>$A^c$</td>
<td>~A</td>
<td>NOT</td>
</tr>
<tr>
<td>$A - B$</td>
<td>A &amp; ~B</td>
<td>DIFFERENCE</td>
</tr>
</tbody>
</table>
Set Operations on Images

Complementation

\[ f^c(x) = t_{\text{max}} - f(x) \]

Set Difference

\[ X \setminus Y = X \cap Y^c \]

Note: Only on binary images

Translation

\[ f_b(x) = f(x - b) \]

Reflection

\[ B = \{ -b | b \in B \} \]
Morphological Image Operations

- All morphological image operations are the result of interaction between a set representing an image and a set representing a structuring element.
- All interactions are based on combination of intersection, union, complementation and translation.
Graphs

Graph is a pair of vertices and edges \((V,E)\), where:

\[
V = (v_1, v_2, \cdots, v_n)
\]

\[
E = (e_1, e_2, \cdots, e_m)
\]

- Planar graph
- Simple graph

**Neighborhood of vertex** \(v\):

\[
N_G(v) = \{v' \in V \mid (v, v') \in E\}
\]

**Path** \(P\) in graph \(G\):

\[
P_G = (v_0, v_1, \cdots, v_i), (v_i, v_{i+1}) \quad \text{neighbors}
\]
Grids & Connectivity

Connectivity:
a set is connected if each pair of its points can be joined by a path completely in the set

$G^h$-Connectivity:
2 pixel $p$ and $q$ of image $f$ are $G^h$-connected iff there exists a $P^h_G$ path with end points $p$ and $q$
Structuring Element (SE): A Small Set for Probing Images

(a) A binary image.

(b) A grey scale image.

(c) Topographic representation of (b).

(d) Shape of some common structuring elements.

(e) Extraction of vertical structures of (a) using a vertical SE.

(f) Extraction of vertical structures of (b) using a vertical SE.
Morphological Image Processing

Erosions and dilations are the most elementary operators of mathematical morphology.

More complicated morphological operators can be designed by means of combining erosions and dilations.
Erosion

\[ A \ominus B = \{ z \mid (B)_z \subseteq A \} \]

"Contracts" the boundary of \( A \). (I)

Set of all points \( z \) such that \( B \), translated by \( z \), is included in \( A \)

Effect of erosion using a \( 3 \times 3 \) square structuring element
Erosion

\[ \mathcal{E}_B(X) = \{ x \mid B_x \subseteq X \} \quad \text{: Eroding set } X \text{ with SE } B \]

\[ \mathcal{E}_B(X) = \bigcap_{b \in B} X_{-b} \]

\[ \mathcal{E}_B(X) = X \ominus B \]

**FIGURE 9.4** (a) Set \( A \). (b) Square structuring element, \( B \). (c) Erosion of \( A \) by \( B \), shown shaded. (d) Elongated structuring element. (e) Erosion of \( A \) by \( B \) using this element. The dotted border in (c) and (e) is the boundary of set \( A \), shown only for reference.
Erosion: implementation

Suppose that the structuring element is a \(3\times3\) square.

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

Note that in subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

Structuring element

The structuring element is now superimposed over each foreground pixel (input pixel) in the image. If all the pixels below the structuring element are foreground pixels then the input pixel retains its value. But if any of the pixels is a background pixel then the input pixel gets the background pixel value.
Erosion Example

**FIGURE 9.5** Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.
Dilation

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

“Expands” the boundary of \( A \). (I)

Set of all points \( z \) such that \( B \), flipped and translated by \( z \), has a non-empty intersection with \( A \)

\( B = \text{structuring element} \)

NOTE: the flipping of the structuring element is included in analogy to convolution. Not all Authors perform it.
Dilation

\[ \delta_B(X) = \{ x \mid B_x \cap X \neq \emptyset \} \]

\[ \delta_B(X) = \bigcup_{b \in B} X_b \]

\[ \delta_B(X) = X \oplus B \]

**FIGURE 9.6**
(a) Set A.
(b) Square structuring element (the dot denotes the origin).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element.
(e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.
Dilation

Effect of dilation using a $3 \times 3$ square structuring element
Dilation: Examples

bridging the gaps

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.

**FIGURE 9.7**
(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.
Erosion and Dilation Example

![Erosion and Dilation Example](image)

**FIGURE 9.7** (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1’s, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

**eliminating small objects**

NOTE: white objects on black background (opposite wrt prev. slides)

NOTE: the final dilation will NOT yield in general the exact shape of the original objects
clear all;
B=zeros(256,256); % create 256x256 image
for i=128:160
    for j=128:160 B(i,j)=1; end
end
for i=20:25
    for j=30:190 B(i,j)=1; end
end
i=1;
while i<40 % generate 40 random pixels
    x=uint8(rand*255); y=uint8(rand*255);
    B(x,y)=1; i=i+1;
end
imshow(B); title('Original image');
K3=ones(3); K5=ones(5); K7=ones(7); K9=ones(9); % create kernels
B3=imdilate(B,K3); B5=imdilate(B,K5); % apply dilation
B7=imdilate(B,K7); B9=imdilate(B,K9);
figure; imshow(B3); title('Dilated image with 3 by 3');
figure; imshow(B5); title('Dilated image with 5 by 5');
figure; imshow(B7); title('Dilated image with 7 by 7');
figure, imshow(B9); title('Dilated image with 9 by 9');
Properties of Erosion and Dilation

- Erosion and Dilation are irreversible operations
- Homotopy is not preserved under either one

(a) A set $X$ and its homotopy tree
(b) A set $Y$ and its homotopy tree
(c) A set $Z$ and its homotopy tree
Opening and Closing

- Opening and Closing are morphological operations which are based on dilation and erosion.

- Opening smoothes the contours of objects, breaks narrow isthmuses and eliminates thin protrusions.

- Closing also produces the smoothing of sections of contours but fuses narrow breaks, fills gaps in the contour and eliminates small holes.

- Opening is basically erosion followed by dilation while closing is dilation followed by erosion.
**Opening and closing**

**OPENING** is erosion followed by dilation

**CLOSING** is dilation followed by erosion

<table>
<thead>
<tr>
<th>Operation</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening</td>
<td>$A \circ B = (A \Theta B) \oplus B$</td>
<td>Smooths contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)</td>
</tr>
<tr>
<td>Closing</td>
<td>$A \cdot B = (A \oplus B) \ominus B$</td>
<td>Smooths contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)</td>
</tr>
</tbody>
</table>
Opening

- An opening is defined as an erosion followed by a dilation using the same structuring element for both operations.

Effect of opening using a $3 \times 3$ square structuring element
FIGURE 9.8 (a) Structuring element $B$ “rolling” along the inner boundary of $A$ (the dot indicates the origin of $B$). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade $A$ in (a) for clarity.
Closing

Effect of closing using a 3×3 square structuring element
FIGURE 9.9 (a) Structuring element $B$ “rolling” on the outer boundary of set $A$. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade $A$ in (a) for clarity.
Erosion and Dilation
Opening and Closing

are dual operators wrt set complementation and reflection:

\[(A \quad B)^C = A^C \oplus \hat{B}\]

\[(A \bullet B)^C = A^C \circ \hat{B}\]
Opening and Closing: Example

**FIGURE 9.10**
Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.
Opening and Closing: Example

Gaps exist in the output;
Better results with a smaller SE
Hit-or-Miss Transformation

- The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image.
- The hit-and-miss transform is a basic tool for shape detection.
- Concept: To detect a shape:
  -- Hit object
  -- Miss background
The structural elements used for Hit-or-miss transforms are an extension to the ones used with dilation, erosion etc.

The structural elements can contain both foreground and background pixels, rather than just foreground pixels, i.e. both ones and zeros.

The structuring element is superimposed over each pixel in the input image, and if an exact match is found between the foreground and background pixels in the structuring element and the image, the input pixel lying below the origin of the structuring element is set to the foreground pixel value. If it does not match, the input pixel is replaced by the boundary pixel value.

\[ A \otimes B = (A \oplus B_1) \cap (A^C \oplus B_2) \]
Hit-or-Miss Transformation

Four structuring elements used for corner finding in binary images using the hit-and-miss transform. Note that they are really all the same element, but rotated by different amounts.

Matlab Command: bwhitmiss
Hit-or-Miss Transformation

\[ HMT_B(X) = \{ x \mid (B_{FG})_x \subseteq X, (B_{BG})_x \subseteq X^c \} \]

\[ HMT_B(X) = \varepsilon_{B_{FG}}(X) \cap \varepsilon_{B_{BG}}(X^c) \]

- Property:
  \[ HMT_B(X) = HMT_{B^c}(X^c) \]

where, \( B = (B_1, B_2) \)
\( B^c = (B_2, B_1) \)
Hit-or-Miss Transformation

FIGURE 9.12
(a) Set $A$. (b) A window, $W$, and the local background of $D$ with respect to $W$, ($W - D$).
(c) Complement of $A$. (d) Erosion of $A$ by $D$.
(e) Erosion of $A^c$ by ($W - D$).
(f) Intersection of (d) and (e), showing the location of the origin of $D$, as desired. The dots indicate the origins of $C$, $D$, and $E$. 
Boundary Extraction

- The boundary of a set $X$ is obtained by first eroding $X$ by structuring element $K$ and then taking the set difference of $X$ and it’s erosion.

- The resultant image after subtracting the eroded image from the original image has the boundary of the objects extracted. The thickness of the boundary depends on the size of the structuring element.
Boundary Extraction

\[ \beta(A) = A - (A \Theta B) \]

**FIGURE 9.13** (a) Set \( A \). (b) Structuring element \( B \). (c) \( A \) eroded by \( B \). (d) Boundary, given by the set difference between \( A \) and its erosion.
Boundary Extraction Example

FIGURE 9.14
(a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

```matlab
se=[1 1 1;1 1 1;1 1 1];
x=imread('Fig9_l4a.jpg');
y=imerode(x,se);
z=imsubtract(x,y);
imshow(z);
figure;imshow(x);
```
Region filling

\[ X_0 = P \]

while \( X_k \neq X_{k-1} \) do
  \[ X_k = (X_{k-1} \oplus B) \cap A^C \]
  \[ X_F = X_k \cup A \]

The dilation would fill the whole area were it not for the intersection with AC

\( \rightarrow \) Conditional dilation

FIGURE 9.15 Hole filling. (a) Set \( A \) (shown shaded). (b) Complement of \( A \). (c) Structuring element \( B \). (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].
Hole Filling Example

**FIGURE 9.16** (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.
Connected component extraction

\[ X_k = \delta_B(X_{k-1}) \cap A \quad k = 1, 2, 3, \ldots \]

**FIGURE 9.17** Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).


**Figure 9.18**

(a) X-ray image of chicken filet with bone fragments.

(b) Thresholded image.

(c) Image eroded with a $5 \times 5$ structuring element of 1s.

(d) Number of pixels in the connected components of (c).

(Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)

<table>
<thead>
<tr>
<th>Connected component</th>
<th>No. of pixels in connected comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>02</td>
<td>9</td>
</tr>
<tr>
<td>03</td>
<td>9</td>
</tr>
<tr>
<td>04</td>
<td>39</td>
</tr>
<tr>
<td>05</td>
<td>133</td>
</tr>
<tr>
<td>06</td>
<td>1</td>
</tr>
<tr>
<td>07</td>
<td>1</td>
</tr>
<tr>
<td>08</td>
<td>743</td>
</tr>
<tr>
<td>09</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>674</td>
</tr>
<tr>
<td>15</td>
<td>85</td>
</tr>
</tbody>
</table>
FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set $A$. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to $m$-connectivity.
Thickening

**FIGURE 9.22** (a) Set $A$. (b) Complement of $A$. (c) Result of thinning the complement of $A$. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.
Skeletons

Maximum disk: largest disk included in A, touching the boundary of A at two or more different places

FIGURE 9.23
(a) Set A.
(b) Various positions of maximum disks with centers on the skeleton of A.
(c) Another maximum disk on a different segment of the skeleton of A.
(d) Complete skeleton.
Skeletons

Define: \( A \ kB = (((A \ B) \ B) \ldots \ B) \)

\[
S(A) = \bigcup_{k=0}^{K} S_k(A)
\]

\[
S_k(A) = (A \Theta kB) - [(A \Theta kB) \circ B]
\]

Reconstruction of \( A \):

\[
A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)
\]

Finds the skeleton \( S(A) \) of set \( A \). The last equation indicates that \( A \) can be reconstructed from its skeleton subsets \( S_k(A) \).

In all three equations, \( K \) is the value of the iterative step after which the set \( A \) erodes to the empty set.

The notation \( (A \Theta kB) \) denotes the \( k \)th iteration of successive erosion of \( A \) by \( B \). (I)
Skeletons

It is not granted that the resulting skeleton is maximally thin, connected, minimally eroded.

Other techniques exist; e.g., the Medial Axis Transform, or conditional thinning algorithms.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bothat</td>
<td>“Bottom-hat” operation using a $3 \times 3$ structuring element; use \texttt{imbothat} (see Section 9.6.2) for other structuring elements.</td>
</tr>
<tr>
<td>bridge</td>
<td>Connect pixels separated by single-pixel gaps.</td>
</tr>
<tr>
<td>clean</td>
<td>Remove isolated foreground pixels.</td>
</tr>
<tr>
<td>close</td>
<td>Closing using a $3 \times 3$ structuring element; use \texttt{imclose} for other structuring elements.</td>
</tr>
<tr>
<td>diag</td>
<td>Fill in around diagonally connected foreground pixels.</td>
</tr>
<tr>
<td>dilate</td>
<td>Dilation using a $3 \times 3$ structuring element; use \texttt{imdilate} for other structuring elements.</td>
</tr>
<tr>
<td>erode</td>
<td>Erosion using a $3 \times 3$ structuring element; use \texttt{imerode} for other structuring elements.</td>
</tr>
<tr>
<td>fill</td>
<td>Fill in single-pixel “holes” (background pixels surrounded by foreground pixels); use \texttt{imfill} (see Section 11.1.2) to fill in larger holes.</td>
</tr>
<tr>
<td>hbreak</td>
<td>Remove H-connected foreground pixels.</td>
</tr>
<tr>
<td>majority</td>
<td>Make pixel $p$ a foreground pixel if at least five pixels in $N_8(p)$ (see Section 9.4) are foreground pixels; otherwise make $p$ a background pixel.</td>
</tr>
<tr>
<td>open</td>
<td>Opening using a $3 \times 3$ structuring element; use function \texttt{imopen} for other structuring elements.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove “interior” pixels (foreground pixels that have no background neighbors).</td>
</tr>
<tr>
<td>shrink</td>
<td>Shrink objects with no holes to points; shrink objects with holes to rings.</td>
</tr>
<tr>
<td>skel</td>
<td>Skeletonize an image.</td>
</tr>
<tr>
<td>spur</td>
<td>Remove spur pixels.</td>
</tr>
<tr>
<td>thicken</td>
<td>Thicken objects without joining disconnected 1s.</td>
</tr>
<tr>
<td>thin</td>
<td>Thin objects without holes to minimally connected strokes; thin objects with holes to rings.</td>
</tr>
<tr>
<td>tophat</td>
<td>“Top-hat” operation using a $3 \times 3$ structuring element; use \texttt{imtophat} (see Section 9.6.2) for other structuring elements.</td>
</tr>
</tbody>
</table>
Matlab examples - dilation

```
originalBW = imread('text.png');
se = strel('line',11,90);
dilatedBW = imdilate(originalBW,se);
figure, imshow(originalBW), figure, imshow(dilatedBW)

originall = imread('cameraman.tif');
se = strel('ball',5,5);
dilatedl = imdilate(originall,se);
figure, imshow(originall), figure, imshow(dilatedl)

se1 = strel('line',3,0);
se2 = strel('line',3,90);
composition = imdilate(1,[se1 se2],'full')
```
Matlab examples - erosion

originalBW = imread('text.png');
se = strel('line', 11, 90);
erodedBW = imerode(originalBW, se);
figure, imshow(originalBW)
figure, imshow(erodedBW)

originall = imread('cameraman.tif');
se = strel('ball', 5, 5);
erodedl = imerode(originall, se);
figure, imshow(originall), figure, imshow(erodedl)
Matlab examples - closing

originalBW = imread('circles.png');
figure, imshow(originalBW);
se = strel('disk',10);
closeBW = imclose(originalBW,se);
figure, imshow(closeBW);
Matlab examples - opening

```matlab
original = imread('snowflakes.png');
se = strel('disk',5);
afterOpening = imopen(original,se);
figure, imshow(original), figure, imshow(afterOpening)
```
Matlab examples - HMT

bw=[0 0 0 0 0 0
   0 0 1 1 0 0
   0 1 1 1 1 0
   0 1 1 1 1 0
   0 0 1 1 0 0
   0 0 1 0 0 0]

interval = [0 -1 -1
            1 1 -1
            0 1 0];

bw2 = bwhitmiss(bw,interval)
Basic Elements

**FIGURE 9.33** Five basic types of structuring elements used for binary morphology. The origin of each element is at its center and the ×’s indicate “don’t care” values.
Properties of morphological operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Equation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation</td>
<td>((B)_z = {w</td>
<td>w = b + z, \text{ for } b \in B})</td>
</tr>
<tr>
<td>Reflection</td>
<td>(\hat{B} = {w</td>
<td>w = -b, \text{ for } b \in B})</td>
</tr>
<tr>
<td>Complement</td>
<td>(A^c = {w</td>
<td>w \notin A})</td>
</tr>
<tr>
<td>Difference</td>
<td>(A - B = {w</td>
<td>w \in A, w \notin B} = A \cap B^c)</td>
</tr>
<tr>
<td>Dilation</td>
<td>(A \oplus B = {z</td>
<td>(\hat{B})_z \cap A \neq \emptyset})</td>
</tr>
<tr>
<td>Erosion</td>
<td>(A \ominus B = {z</td>
<td>(B)_z \subseteq A})</td>
</tr>
<tr>
<td>Opening</td>
<td>(A \circ B = (A \ominus B) \oplus B)</td>
<td>Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)</td>
</tr>
</tbody>
</table>

(Continued)

**TABLE 9.1**
Summary of morphological operations and their properties.
## Properties of morphological operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Equation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closing</td>
<td>$A \ast B = (A \ominus B) \cup B$</td>
<td>Smooths contours, fuses narrow breaks and long thin gulls, and eliminates small holes. (I)</td>
</tr>
<tr>
<td>Hit-or-miss transform</td>
<td>$A \ominus B = (A \ominus B_1) \cap (A' \ominus B_2)$</td>
<td>The set of points (coordinates) at which, simultaneously, $B_1$ and $B_2$ found a match (“hit”) in $A$ and $B_2$ found a match in $A'$</td>
</tr>
<tr>
<td>Boundary extraction</td>
<td>$\beta(\delta) = A - (A \ominus B)$</td>
<td>Set of points on the boundary of set $A$. (I)</td>
</tr>
<tr>
<td>Hole filling</td>
<td>$X_k = (X_{k-1} \ominus B) \cap A'$</td>
<td>Fills holes in $A$; $X_0 = \text{array of }0\text{s with a }1\text{ in each hole. (II)}</td>
</tr>
<tr>
<td>Connected components</td>
<td>$X_k = (X_{k-1} \ominus B) \cap A$</td>
<td>Finds connected components in $A$; $X_0 = \text{array of }0\text{s with a }1\text{ in each connected component. (I)}</td>
</tr>
<tr>
<td>Convex hull</td>
<td>$X_{k-1}^i = (X_{k-1}^i \ominus B') \cup A$; $X_{k-1}^i = A$; and $D^i = X_{k-1}^i$</td>
<td>Finds the convex hull $C(A)$ of set $A$, where “conv” indicates convergence in the sense that $X_{k-1}^i = X_{k-1}^i$. (III)</td>
</tr>
<tr>
<td>Thinning</td>
<td>$A \ominus B = A - (A \ominus B)$</td>
<td>Thins set $A$. The first two equations give the basic definition of thinning. The last equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)</td>
</tr>
<tr>
<td>Thickening</td>
<td>$A \ominus B = A \cup (A \ominus B)$</td>
<td>Thickens set $A$. (See preceding comments on sequences of structuring elements.) Use IV with $0$s and $1$s reversed.</td>
</tr>
<tr>
<td>Skeletons</td>
<td>$S(A) = \bigcup_{k=0}^{K} S_k(A)$</td>
<td>Finds the skeleton $S(A)$ of set $A$. The last equation indicates that $A$ can be reconstructed from its skeleton subsets $S_k(A)$. In all three equations, $K$ is the value of the iterative step after which the set $A$ erodes to the empty set. The notation $(A \ominus kB)$ denotes the $k$th iteration of successive erosions of $A$ by $B$. (I)</td>
</tr>
</tbody>
</table>

(Continued)
### Properties of morphological operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Equation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning</td>
<td>$X_1 = A \ominus {B}$</td>
<td>$X_4$ is the result of pruning set $A$. The number of times that the first equation is applied to obtain $X_1$ must be specified. Structuring elements $V$ are used for the first two equations. In the third equation $H$ denotes structuring element $I$. $F$ and $G$ are called the marker and mask images, respectively.</td>
</tr>
<tr>
<td></td>
<td>$X_2 = \bigcup_{k=1}^{n}(X_1 \ominus B^k)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3 = (X_2 \ominus H) \cap A$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4 = X_1 \cup X_3$</td>
<td></td>
</tr>
<tr>
<td>Geodesic dilation of</td>
<td>$D_G^{(1)}(F) = (F \ominus B) \cap G$</td>
<td></td>
</tr>
<tr>
<td>size 1</td>
<td>$D_G^{(n)}(F) = D_G^{(1)}[D_G^{(n-1)}(F)]$</td>
<td></td>
</tr>
<tr>
<td>Geodesic dilation of</td>
<td>$D_G^{(0)}(F) = F$</td>
<td></td>
</tr>
<tr>
<td>size $n$</td>
<td>$E_G^{(1)}(F) = (F \ominus B) \cup G$</td>
<td></td>
</tr>
<tr>
<td>Geodesic erosion of</td>
<td>$E_G^{(n)}(F) = E_G^{(1)}[E_G^{(n-1)}(F)]$</td>
<td></td>
</tr>
<tr>
<td>size 1</td>
<td>$E_G^{(0)}(F) = F$</td>
<td></td>
</tr>
<tr>
<td>Geodesic erosion of</td>
<td>$R_F^{(0)}(F) = D_G^{(0)}(F)$</td>
<td>$k$ is such that $D_G^{(k)}(F) = D_G^{(k+1)}(F)$</td>
</tr>
<tr>
<td>size $n$</td>
<td>$R_F^{(n)}(F) = E_G^{(0)}(F)$</td>
<td>$k$ is such that $E_G^{(k)}(F) = E_G^{(k+1)}(F)$</td>
</tr>
<tr>
<td>Morphological</td>
<td>$O_B^{(n)}(F) = R_F^{(1)}[(F \ominus nB)]$</td>
<td>$(F \ominus nB)$ indicates $n$ erosions of $F$ by $B$.</td>
</tr>
<tr>
<td>reconstruction by</td>
<td>$C_B^{(n)}(F) = R_F^{(1)}[(F \ominus nB)]$</td>
<td>$(F \ominus nB)$ indicates $n$ dilations of $F$ by $B$.</td>
</tr>
<tr>
<td>dilation</td>
<td>$H = \left[R_F^{(0)}(F)\right]^c$</td>
<td>$H$ is equal to the input image $I$, but with all holes filled. See Eq. (9.5-28) for the definition of the marker image $F$.</td>
</tr>
<tr>
<td>Hole filling</td>
<td>$X = I - R_F^{(1)}(F)$</td>
<td>$X$ is equal to the input image $I$, but with all objects that touch (are connected to) the boundary removed. See Eq. (9.5-30) for the definition of the marker image $F$.</td>
</tr>
</tbody>
</table>
Announcement

- HW3 is out today, due on 10/31.
- Keshan will present HW3 on 11/01.
- Read Chapter 9

Final project: Send me the following info today
- The project title and a short description if you choose your own project (must be image processing related).
- Your partner if you want to team with someone.