Morphological Image Processing I
Lecture 07

Milan Gavrilovic
milan@cb.uu.se

Centre for Image Analysis
Uppsala University

Computer Assisted Image Analysis
2010-04-19
Reading Instructions
Chapters for this lecture

- Chapter 9.1 – 9.5.4 in Gonzales-Woods.
Previous Lectures

### Image processing
- Point processing (Spatial domain, pixel-wise)
- Local neighbourhoods (Spatial domain, filtering)
- Fourier transform (Frequency domain, filtering)

### Image analysis
- Segmentation and labeling
Mathematical framework used for:

- Pre-processing
  - Noise filtering, shape simplification, ...

- Enhancing object structure

- Segmentation

- Quantitative description
  - Area, perimeter, ...
Some set theory

- **A** is a set in $\mathbb{Z}^2$.
- If $a = (a_1, a_2)$ is an element in $A$: $a \in A$.
- If $a = (a_1, a_2)$ is *not* an element in $A$: $a \notin A$.
- **Empty set**: $\emptyset$.
- Set specified using $\{\}$, e.g.,
  $$C = \{w | w = -d, \forall d \in D\}.$$
- Every element in $A$ is also in $B$ (subset): $A \subseteq B$.
- **Union** of $A$ and $B$:
  $$C = A \cup B = \{c | c \in A \text{ or } c \in B\}.$$
- **Intersection** of $A$ and $B$:
  $$C = A \cap B = \{c | c \in A \text{ and } c \in B\}.$$
- **Disjoint/mutually exclusive**:
  $$A \cap B = \emptyset.$$
Some more set theory

- **Complement of** $A$: $A^C = \{w | w \notin A\}$.
- **Difference of** $A$ and $B$: $A - B = \{w | w \in A, w \notin B\} = A \cap B^C$.
- **Reflection of** $A$: $\hat{A} = \{w | w = -a, \forall a \in A\}$.
- **Translation of** $A$ by a point $z = (z_1, z_2)$: $(A)_Z = \{c | c = a + z, \forall a \in A\}$. 
Logical operations
Pixel-wise combination of images (AND, OR, NOT, XOR)
Structuring element (SE)

- Small set to probe the image under study.
- For each SE, define an origin:
  - SE in point \( p \); origin coincides with \( p \).
- Shape and size must be adapted to geometric properties for the objects.
Basic idea

In parallel for each pixel in binary image:
- Check if SE is *satisfied*.
- Output pixel is set to 0 or 1 depending on used operation.

![Image with labeled parts: image, SE, and pixels in output image if SE fits]
How to describe the SE

Possible in many different ways!

Information needed:

- Position of origin for SE.
- Position of elements belonging to SE.

N.b.

Matlab assumes it’s centre element to be the origin!
Five binary morphological transforms

- Erosion.
- Dilation.
  - Opening.
  - Closing.
- Hit-or-Miss transform.
Erosion (shrinking)

Does the structuring element fit the set?

Erosion of a set $X$ by structuring element $B$, $\varepsilon_B(X)$: all $x$ in $X$ such that $B$ is in $X$ when origin of $B = x$.

$$\varepsilon_B(X) = \{ x | B_x \subseteq X \}.$$  

Gonzalez-Woods:

$$X \ominus B = \{ x | (B)_x \subseteq X \}.$$  

Shrink the object.
Example: erosion (fill in!)
Dilation (growing)

Does the structuring element hit the set?

Dilation of a set $X$ by structuring element $B$, $\delta_B(X)$: all $x$ in $X$ such that the reflection of $B$ hits $X$ when origin of $B = x$.

$$\delta_B(X) = \{x|(\hat{B})_x \cap X \not\in \emptyset\}.$$  

Gonzalez-Woods:

$$X \oplus B = \{x|(\hat{B})_x \cap X \not\in \emptyset\}.$$  

Grow the object.
Example: dilation (fill in!)
Different SE give different results

- Set $A$.
- Square structuring element (dot is the centre).
- Dilation of $A$ by $B$, shown shaded.
- Elongated structuring element (dot is the centre).
- Dilation of $A$ using this element.
Erosion and dilation are dual with respect to complementation and reflection,

\[(A \ominus B)^C = A^C \ominus \hat{B}.\]
Examples

\[ A \]

\[ A \ominus B \]

\[ (A \ominus B)^c \]

\[ B = \hat{B} \]

\[ \text{origin} \]

\[ A^c \]

\[ A^c \ominus B \]
Typical application

**Erosion**
Removal of structures of certain shape and size, given by SE (structure element).

Example $3 \times 3$ SE

**Dilation**
Filling of holes of certain shape and size, given by SE.

Example $3 \times 3$ SE
Examples

Erosion: SE = square of size $13 \times 13$.

Input: squares of size $1 \times 1$, $3 \times 3$, $5 \times 5$, $7 \times 7$, $9 \times 9$, and $15 \times 15$ pixels.

Dilation of erosion result: SE = square of size $13 \times 13$. 
Use dilation to bridge gaps of broken segments

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.

Sample text of poor resolution with broken characters (magnified view).

Structuring element.

Dilation of (1) by (2).

Broken segments were joined.
Use dilation to bridge gaps of broken segments

**Wanted:**
Remove structures/fill holes without affecting remaining parts.

**Solution:**
Combine erosion and dilation (using same SE).

- Opening.
- Closing.
Opening

Erosion followed by dilation, denoted $\circ$.

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

- Eliminates protrusions.
- Break necks.
- Smooths contour.
Example opening (fill in!)

Example opening

A

A ⊖ B

B =
Opening: roll ball (=SE) inside object
See $B$ as a “rolling ball”

Boundary of $A \bullet B$ are equal to points in $B$ that reaches closest to the boundary $A$ when $B$ is rolled inside $A$.
Closed Image Processing I

Closing

Dilation followed by erosion, denoted \( \bullet \).

\[
A \bullet B = (A \oplus B) \ominus B
\]

- Smooth contour.
- Fuse narrow breaks and long thin gulfs.
- Eliminate small holes.
- Fill gaps in the contour.
Example closing (fill in!)

Example closing

A

A ⊕ B

B =
Closing: roll ball (=SE) outside object
(Fill in border after closing with ball as SE!)

Boundary of $A \circ B$ are equal to points in $B$ that reaches closest to the boundary of $A$ when $B$ is rolled outside $A$. 
Hit-or-miss transformation (⊗ or HMT)

Find location of one shape among a set of shapes (“template matching”).

\[ A \otimes B = (A \ominus B_1) \cap (A^C \ominus B_2) \]

Composite SE: Object part \((B_1)\) and background \((B_2)\).

Does \(B_1\) fit the object while, simultaneously, \(B_2\) misses the object, i.e., fit the background.
Hit-or-miss transformation (⊗ or HMT)

Find location of one shape among a set of shapes.

\[ A \otimes B = (A \ominus X) \cap (A^C \ominus (W - X)) \]

Alternative:

\[ A \otimes B = (A \ominus B_1) \cap (A^C \ominus B_2) \]

\[ = (A \ominus B_1) \cap (A \oplus \hat{B}_2)^C \]

\[ = (A \ominus B_1) - (A \oplus \hat{B}_2) \]
Example hit-or-miss transform (fill in!)

Search for:

\[ B_1 \]

\[ B_2 \]

\[ A \]

\[ A^c \]
Basic morphological algorithms

Use erosion, dilation, opening, closing, hit-or-miss transform for

- Boundary extraction.
- Region filling.
- Extraction of connected components (labeling).
- Defining the convex hull.
- Defining the skeleton.
Boundary extraction
by erosion and set difference (boundary of $A = \beta(A)$)

Extract the boundary of:

$$\beta(A) = A - (A \ominus B).$$

8-connected boundary
$$\beta(A) = \text{pixels with edge neighbour in } A^C.$$

4-connected boundary
$$\beta(A) = \text{pixels with edge or point neighbour in } A^C.$$
Region filling

Fill a region $A$ given its boundary $\beta(A)$. $x = X_0$ is known and inside $\beta(A)$.

$$X_k = (X_{k-1} \oplus B) \cap A^C, \quad k = 1, 2, 3, \ldots$$

Continue until $X_k = X_{k-1}$.
Filled region $A \cup X_k$.

Use to fill holes! *Conditional dilation!*
Example of region filling

\[ A \quad A^c \quad X_0 \quad X_1 \]

\[ X_2 \quad X_6 \quad X_7 \quad X_7 \cup A \]
Compare with removing holes using two-pass labeling algorithm
See segmentation lecture

Connected component labeling
- Label the inverse image.
- Remove connected components touching the image border.
- Output = holes + original image.
→ 2 scans + 1 scan (straight forward...)

Mathematical morphology
- Iterate: dilation, set intersection
→ Dependent on size and shape of the hole needed: initialization!
Convex hull

- Region $R$ is convex if
  - For any points $x_1, x_2 \in R$, straight line between $x_1$ and $x_2$ is in $R$.

- Convex hull $H$ of a region $R$
  - Smallest convex set containing $R$.

- Convex deficiency $D = H - R$. 
Count the number of neighbours of a pixel, if more than three, mark the pixel! Repeat until none of the pixels has more than three neighbours.

Homework - fill in!
Reading Instructions
Chapters for tomorrow’s lecture

- Chapter 9.5.7, and 9.6 in Gonzales-Woods.