TDL2131 Digital Image Processing

Morphological Image Processing
Lecture 8

John See
Faculty of Information Technology
Multimedia University

Some portions of content adapted from CYPe and Zhu Liu's notes. Most figures from Gonzalez/Woods
Lecture Outline

• Mathematical Morphology
• Review on Set Theory
• Basic Morphological Methods
  – Dilation
  – Erosion
  – Opening
  – Closing
• Hit-or-Miss Transform
Some Announcements

• **Assignment 2** has been released. Deadline is on the **Monday of Week 11**. Total duration given is about 4 weeks (including your mid-term break), so you should already be working on it.

  – Please consult me if you need to. I'm willing to discuss if you have any doubts on your methods, **BUT** i will not give any direct suggestions of how to solve the problem.
Morpho...

- **Morphology** – Branch of biology that deals with the form and structure of organisms without consideration of function.

- **Mathematical Morphology** – Mathematical tool for processing shapes in image, including boundaries, skeletons, convex hulls, etc.

- The operations of mathematical morphology are defined as **set operations**.

- **Morphological Operations** are originally developed for bilevel (binary) images for shape and structural manipulations.
Binary Image Morphology

• A binary image can be considered as a set by considering “white” pixels as elements in the set (foreground) and “black” pixels as outside the set (background)

• Morphological operators can:
  – Thin,
  – Thicken,
  – Find boundaries,
  – Find skeletons (medial axis),
  – Convex Hull, and more
Structuring Element (SE)

- The operations of binary morphology input a binary image $B$ and a structuring element $S$
- The structuring element $S$ is usually another smaller binary image or sub-image
  - It represents a shape, can be of any size and have arbitrary structure that can be represented by a binary image
  - Members and Origin (center of gravity) of SE are specified
Review of Set Theory: Definitions & Notations

• **Set** ($\Omega$)
  - A collection of objects (elements)

• **Membership** ($\in$)
  - If $\omega$ is an element (member) of a set $\Omega$, we write $\omega \in \Omega$

• **Subset** ($\subseteq$)
  - Let $A, B$ are two sets. If for every $a \in A$, we also have $a \in B$, then the set $A$ is a *subset* of $B$, that is, $A \subseteq B$
  - If $A \subseteq B$ and $B \subseteq A$, then $A = B$.

• **Empty set** ($\emptyset$)
Definitions & Notations (cont'd)

- **Complement set**
  - If \( A \subset \Omega \), then its complement set \( A^c = \{ \omega \mid \omega \in \Omega, \text{ and } \omega \notin A \} \)

- **Union (\( \cup \))**
  - \( A \cup B = \{ \omega \mid \omega \in A \text{ or } \omega \in B \} \)

- **Intersection (\( \cap \))**
  - \( A \cap B = \{ \omega \mid \omega \in A \text{ and } \omega \in B \} \)

- **Set difference (\( \setminus \))**
  - \( B \setminus A = B \cap A^c \)
  - Note that \( B \setminus A \neq A \setminus B \)

- **Disjoint set**
  - \( A \) and \( B \) are disjoint (mutually exclusive) if \( A \cap B = \emptyset \)
Set Operations

Figure 9.1
(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 $B$. 
Set Relations

• **Translation:** $(A)_z = \{a: a = a + z, a \in A\}$

• **Reflection:** $B_r = \{b: b = -b, b \in B\}$
Logical Operations between Binary Images

**FIGURE 9.3** Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.
Basic Morphological Operations

- The basic operations of binary morphology are **dilation**, **erosion**, **closing**, and **opening**
  - **Dilation** enlarges a region
  - **Erosion** makes a region smaller
  - A **Closing** operation can close up internal holes in a region and eliminate bays along boundaries
  - An **Opening** operation can get rid of small portions of the region that jut out from the boundary (spurs, bridges)
What are morphological operations for?

• Some applications:
  – Binary morphology can be used to extract primitive features of an object that can be used to recognize/classify the object thereafter
  – A shape matching system can use morphological feature detection to rapidly detect primitives that are used in object recognition
Dilation

- Dilation of set $B$ by structuring element $S$:

$$B \oplus S = \{ x_i \mid (S_x^r \cap B) \neq \emptyset \} = \bigcup_{b \in B} S_b$$

- New set $C = B \oplus S$ is composed of all the points obtained by replacing every point $(x,y)$ in $B$ with a copy of $S$, placing the origin point $(0,0)$ of $S$ at $(x,y)$. This replacement operation works vice versa.
Example: Dilation

H, 3x3, origin at the center

H, 5x3, origin at the center

Dilation enlarges a set.
Example: Dilation

- F
- G
- H, 3x3, origin at the center

Note that the narrow ridge is closed
Erosion

- Erosion of set $B$ by structuring element $S$:

$$B \ominus S = \{ x_i \mid S_x \subseteq B \} = \{ b \mid b + s \in B, \forall s \in S \}$$

- New set $C = B \ominus S$ is composed of all the points $(x,y)$ for which $S$ is in $B$. This can be done by moving $S$ over $B$, find all the places it will fit completely, and for each such place, mark down the point corresponding to the origin $(0,0)$ point of $S$.

Eroding $B$ with a 3x3 structuring element $S$ of ones, centered at origin
Example: Erosion

H, 3x3, origin at the center

H, 5x3, origin at the center

Erosion shrinks a set
Example: Erosion

F

G

H, 3x3, origin at the center
Structuring Element

- The shape, size and orientation of the structuring element (SE) depend on application usage.
- A symmetrical one will enlarge or shrink the original set in all directions.
- A vertical one, will only expand or shrink the original set in the vertical direction.
Applications of Dilation & Erosion

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 9.5**
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.
Applications of Dilation & Erosion

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.
Opening & Closing: Intuitive Interpretation

- **Dilation** expands an object
- **Erosion** contracts an object
- **Opening**?
  - Smoothens contours, enlarges narrow gaps, eliminates thin protrusions and ridges
- **Closing**?
  - Fills narrow gaps, holes and small breaks
Opening

- **Opening**: Like “smoothing from the inside”
- **Erosion followed by Dilation**

\[ B \circ S = (B \ominus S) \oplus S = \bigcup \{ S_x \mid S_x \subseteq B \} \]

- A union of all translations of \( S \) that fit inside \( B \)
Example: Opening

F

$F \Theta H$

$(F \Theta H) \oplus H$

H, 3x3, origin at the center
Closing

- **Closing**: Like “smoothing from the outside”
- **Dilation followed by Erosion**

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

- All translations of S that have nonempty intersections with B
Example: Closing

F

$F \oplus H$

$(F \oplus H) \ominus H$

H, 3x3, origin at the center
Example: Opening & Closing

**FIGURE 9.10**
Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The dark dot is the center of the structuring element.
Duality Between Operators

- The complement of a dilation is equal to the erosion of a complement, and vice versa
  \[ A \oplus B = \overline{A \ominus B_c} \]
  \[ A \ominus B = \overline{A \oplus B_c} \]

- The complement of an closing is equal to the opening of a complement, and vice versa
  \[ A \bullet B = \overline{A \circ B_c} \]
  \[ A \circ B = \overline{A \bullet B_c} \]
Idempotency: Opening & Closing

- Applying opening and closing more than once has no further effect

\[(A \circ B) \circ B = (A \circ B)\]
\[(A \bullet B) \bullet B = (A \bullet B)\]
Hit-or-Miss Transform

• Hit-or-Miss Transform is a powerful method for finding shapes, and their locations in images

• Can be defined entirely in terms of erosion only

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

• Useful for detecting specific shapes that are intended to extract, e.g. squares, triangles, ridges, corners, junctions, etc.
Hit-or-Miss Transform

• Steps:
  – Perform an erosion $A\ominus B_1$ with $B_1$ being the SE shape that we intend to find.
  – Next, erode the complement of $A$ with $B_2$, a SE that is the border that encloses around the shape $B_1$.
  – The intersection of the two erosion operations would produce just one pixel at the center position of the found shape, resulting in a “hit”. Other parts of set $A$ which did not return anything are considered “miss”.  

Hit-or-Miss Transform

(a) Set $A$, (b) A window $W$ and the local background of $X$ w.r.t. $W$, $W-X$. (c) $A^c$. (d) $A \Theta X$

Intersection of (d) and (e) shows the location of the origin of $X$, as desired.
SE with “Don't Care” Entries

• Previous examples of SEs do not contain “Don't Care” entries
• For hit-or-miss transform, structuring elements have:
  – 1 – foreground
  – 0 – background
  – X – don't care
• In some implementations, the values in a SE are created differently to accommodate all 3 types, e.g. +1: foreground, -1: background, 0: don't care
SE with “Don't Care” Entries

• Output is 1 if there is a match for both foreground or background pixels. Likewise, output is -1 if there is no match.

• Note that the “don't care” pixels are not evaluated (anything multiplied by 0 is 0!).

\[
\begin{bmatrix}
-1 & -1 & -1 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix} \cdot \begin{bmatrix}
-1 & -1 & -1 \\
1 & 1 & -1 \\
1 & 1 & 1 \\
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix} : \text{hit/match}
\]

\[
\begin{bmatrix}
-1 & -1 & -1 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix} \cdot \begin{bmatrix}
-1 & -1 & -1 \\
1 & -1 & -1 \\
1 & 1 & 1 \\
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 \\
0 & -1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix} : \text{miss/no match}
\]
SE with “Don't Care” Entries

- Zeros in SE for erosion and dilation are actually “Don't-Cares”.
- For hit-or-miss transform, “don't-cares” are conventionally shown as blanks or 'X' in the kernel to avoid confusion. This particular element can be used to find corners:

```
\[\begin{array}{ccc}
1 & 1 & 1 \\
0 & 1 & 1 \\
0 & 0 & 0 \\
\end{array}\]
```

- 4 other SEs used for corner finding in binary images. They are actually the same element, but rotated by different angles:

```
\[\begin{array}{ccc}
1 & 1 & 1 \\
0 & 1 & 1 \\
0 & 0 & 0 \\
\end{array}\]
\[\begin{array}{ccc}
1 & 1 & 1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}\]
\[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}\]
\[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}\]
```
Simplification of Hit-or-Miss Transform

• Hit-or-miss transform can be reduced to simple erosion to simplify calculations – No background \((B_2)\) matching is needed.

\[
A \ast B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad \Rightarrow \quad A \ast B = (A \ominus B_1)
\]

• This simplification reduces the complexity of calculations but also causes a higher likelihood of inaccurate hits. (Some applications are not critical on the accuracy)
Morphological Functions in Matlab

- `dilate`
- `erode`
- `bwmorph` – special collection of pre-defined morphological operations on binary image
- Look under “Binary image operations” in Image Processing Toolbox Help for other functions that are useful for processing binary images.
Morphological Functions in Matlab

- To extract image components that are useful in representation and description of shape

- Next week:
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Convex Hull
  - Thinning
  - Thickening
  - Skeletonization
  - Pruning
Recommended Readings

- Digital Image Processing (3rd Edition), Gonzalez & Woods,
  - Chapter 9: Morphological Image Processing
    - 9.1 – 9.4 (Week 8)
  - Chapter 9: Morphological Image Processing
    - 9.5 (Week 9)