Morphological Image Processing

Lecture 9

John See
Faculty of Information Technology
Multimedia University

Most portions of content and figures from Gonzalez/Woods
Lecture Outline

• Morphological Algorithms
  – Boundary Extraction
  – Hole Filling
  – Connected Components
  – Convex Hull
  – Thinning
  – Thickening
  – Skeletons
  – Pruning
Morphological Algorithms

• Last lecture: Basic morphological algorithms – dilation, erosion, opening, closing

• These basic morphological operations can be combined together to accomplish more complex tasks, such as
  – Extracting boundaries, connected components, skeletons (finding useful shape information)
  – Thinning, thickening, pruning, and filling regions (useful pre- or post-processing steps)
Boundary Extraction

• A boundary of a set $A$, denoted by $\beta(A)$, is obtained by eroding $A$ by $B$, then perform the set difference between $A$ and its erosion:

$$\beta(A) = A - (A \ominus B)$$
Example: Boundary Extraction

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

● The set difference between set $A$ and its erosion,

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

is normally the **internal boundary** of set $A$

![Diagram showing set $A$ and $\beta(A)$]

● How can we define the **external boundary** of set $A$?
Hole Filling

- Sometimes also referred to as Region Filling
- **Hole**: A background region surrounded by a connected border of foreground pixels
- Let $A$ denote a set whose elements are 8-connected boundaries – each boundary encloses a background region.
- **Objective**: to fill all holes in set $A$ with 1s
Hole Filling

• Form an array, $X_0$ of 0s (same size as array containing $A$), except at the locations in $X_0$ corresponding to the points in each hole, which are set to 1. The following procedure fills all the holes with 1s.

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

where $B$ is a symmetric SE.

• The algorithm terminates at iteration step $k$ if $X_k = X_{k-1}$.

• Set $X_k$ contains all the filled holes.

• The set union, $X_k \cup A$ contains all filled holes and their boundaries.
FIGURE 9.15
Region filling,
(a) Set $A$.
(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

• The **structuring element** (SE) here is actually used to **define connectivity**!
• Examine earlier example
Hole Filling

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

- Process that is central to many automated image analysis applications
- Connected components require connectivity to be specified (4-connected, 8-connected)
- Labelling: How many “connected components” are there in this image?
Extraction of Connected Components

- Form an array, $X_0$ of 0s (same size as array containing A), except at the locations in $X_0$ corresponding to the points in each hole, which are set to 1. The following procedure fills all the holes with 1s.

  \[ X_k = (X_{k-1} \oplus B) \cap A \quad k = 1,2,3,... \]

  where $B$ is a symmetric SE.

- The algorithm terminates at iteration step $k$ if $X_k = X_{k-1}$.

- Set $X_k$ contains all the connected components.

- Note its similarity to Hole Filling algorithm
Example: Extraction of Connected Components

\begin{align*}
Y: \text{connected component in set } A, \\
p: \text{a known point in } Y \\
X_0 &= p \\
X_k &= (X_{k-1} \oplus B) \cap A \\
\text{if } X_k &= X_{k-1} \\
\text{then } Y &= X_k
\end{align*}

\textbf{FIGURE 9.17} (a) Set \( A \) showing initial point \( p \) (all shaded points are valued 1, but are shown different from \( p \) to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.
Example: Extraction of Connected Components

**FIGURE 9.18**
(a) X-ray image of chicken filet with bone fragments.
(b) Thresholded image. (c) Image eroded with a 5 × 5 structuring element of 1’s.
(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>
Convex Hull

• Set $A$ is **convex** if the straight line segment joining any two points in $A$ lies entirely within $A$

• The **convex hull** $H$ of a set $S$ is the smallest convex set containing $S$.

• The set difference $H - S$ is called the **convex deficiency** of $S$

Area shaded blue is the convex hull of the set of points
Convex Hull

- Let $B^i$, $i=1, 2, 3, 4$, represent the four structuring elements.
- Procedure consist of implementing the equation:

$$X^i_k = (X^{k-1} \oplus B^i) \cup A$$

where $i = 1, 2, 3, 4$ and $k = 1, 2, 3, ...$

where $X^i_0 = A$
Convex Hull

\[ X_k^i = (X_{k-1} \oplus B^i) \cup A \]

- When the procedure converges \((X_k^i = X_{k-1}^i)\), we let \(D^i = X_k^i\)
- The convex hull of A is
\[ C(A) = \bigcup_{i=1}^{4} D^i \]

**FIGURE 9.19**
(a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.
Convex Hull

- Iteratively applying the hit-or-miss transform to $A$ with $B^i$ for each $i$ until converges (no changes)
- Perform union with $A$ to obtain $D^i$ for each $i$.
- The union of the four resulting $D$s constitutes the convex hull of $A$. 

**FIGURE 9.19**
(a) Structuring elements. (b) Set $A$. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.
Convex Hull

• Morphological convex hull – unnecessary growth of region to obtain convex hull (due to effect of union)

• Simple approach to reduce this effect – limit growth along vertical and horizontal directions
Thinning

• The thinning of set A by a structuring element B, denoted $A \otimes B$, can be defined in terms of the hit-or-miss transform:

$$A \otimes B = A \ominus (A \ast B)$$

$$= A \cap (A \ast B)^c$$

- As we are interested in pattern matching with SE, so no background operation is required in the hit-or-miss transform.
Thinning

- A more useful expression, is a symmetrically based sequence of SEs (just like in convex hull):
\[
\{ B \} = \{ B^1, B^2, B^3, ..., B^n \}
\]

where \(B^i\) is the rotated version of \(B^{i-1}\).

- We now define thinning as
\[
A \otimes \{ B \} = (((A \otimes B^1) \otimes B^2)...) \otimes B^n
\]

- This process thins \(A\) by one pass with \(B^1\), then thins the result with one pass of \(B^2\), until \(A\) is thinned with one pass of \(B^n\).
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) Re-
sult of using the first element again (there were no changes for the next two elements).
(k) Result after convergence. (l) Conversion to m-connectivity.
Thickening

- Thickening is the morphological dual of thinning and is defined by the expression:
  \[ A \otimes B = A \cup (A \ast B) \]

- where B is a structuring element suitable for thickening. As in thinning, thickening can be defined as a sequential operation:
  \[ A \bullet \{ B \} = (((A \bullet B^1) \bullet B^2)...) \bullet B^n \]

- The SEs used are similar to those used for thinning.
Thickening

- An algorithm for thickening is seldom used in practice
- Normally, thickening is done by thinning the background of the set of interest, and then complementing the result
- Followed by post-processing to remove disconnected pixels

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 (Skeletonizing)

- The notion of a skeleton, $S(A)$, of a set $A$:

**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 (Skeletonizing)

- If $z$ is a point of $S(A)$ and $(D)_z$ is the largest disk centered at $z$ and contained in $A$, one cannot find a larger disk (not necessarily centered at $z$) containing $(D)_z$ and included in $A$. The disk $(D)_z$ is a maximum disk.
- The disk $(D)_z$ touches the boundary of $A$ at two or more different places.
Skeletons (Skeletonizing)

• The skeleton of A can be expressed in terms of erosions and openings:

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

where

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

where B is the SE, and \( (A \Theta kB) \) indicates \( k \) successive erosions of A:

\[ (A \Theta kB) = (((...((A \Theta B) \Theta B) \Theta ...) \Theta B) \Theta B) \]

\( k \) times, and K is the last iterative step before A erodes to an empty set.
Skeletons (Skeletonizing)

- $S(A)$ is obtained as the union of the skeleton subsets $S_k(A)$
- Morphological skeletonizing does NOT guarantee connectivity

<table>
<thead>
<tr>
<th>$k$</th>
<th>$A \ominus kB$</th>
<th>$(A \ominus kB) \cdot B$</th>
<th>$S_k(A)$</th>
<th>$\bigcup_{k=0}^{K} S_k(A)$</th>
<th>$S_k(A) \ominus kB$</th>
<th>$\bigcup_{k=0}^{K} S_k(A) \ominus kB$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image1.png" alt="Table Cell" /></td>
<td><img src="image2.png" alt="Table Cell" /></td>
<td><img src="image3.png" alt="Table Cell" /></td>
<td><img src="image4.png" alt="Table Cell" /></td>
<td><img src="image5.png" alt="Table Cell" /></td>
<td><img src="image6.png" alt="Table Cell" /></td>
</tr>
<tr>
<td>1</td>
<td><img src="image7.png" alt="Table Cell" /></td>
<td><img src="image8.png" alt="Table Cell" /></td>
<td><img src="image9.png" alt="Table Cell" /></td>
<td><img src="image10.png" alt="Table Cell" /></td>
<td><img src="image11.png" alt="Table Cell" /></td>
<td><img src="image12.png" alt="Table Cell" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image13.png" alt="Table Cell" /></td>
<td><img src="image14.png" alt="Table Cell" /></td>
<td><img src="image15.png" alt="Table Cell" /></td>
<td><img src="image16.png" alt="Table Cell" /></td>
<td><img src="image17.png" alt="Table Cell" /></td>
<td><img src="image18.png" alt="Table Cell" /></td>
</tr>
</tbody>
</table>

**FIGURE 9.24** Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.
Skeletons (Skeletonizing)

• Morphological representation of skeletons presents an elegant formulation

• BUT, heuristic formulations (usually post-processing) are needed to produce a perfect, optimal skeleton
  – Maximally thin, connected, and minimally eroded
Pruning

- **Pruning** – essential complement to thinning and skeletonizing algorithms, which tend to leave parasitic components that need to be “cleaned up” by post-processing

- **Parasitic components**, also known as “spurs”.

- How do we develop a morphological approach to pruning?
Example: Pruning

**Figure 9.25**
(a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.
Pruning

• Previous example: After 3 cycles of thinning, the next step is to “restore” the character to its original form, by removing the spurs
  - Form a set containing all end points of the thinned image
  - Dilate the end points three times (can be more or less), using the original set as a delimiter to growth (intersection with original set)
  - Finally, the union of the dilated region and the original thinned image yields the desired result

• What other ways?
bwmorph

- **bwmorph** implements quite a large number of binary morphological algorithms (including those discussed today – thinning, thickening, skeletons, pruning, hole filling, etc.)

- To fully understand how they are implemented, study its Matlab code

- You can also implement them from scratch, based on the algorithms you have learnt
Recommended Readings

- Digital Image Processing (3rd Edition), Gonzalez & Woods,
  - Chapter 9: Morphological Image Processing
    - 9.5 (Week 9)
  - Chapter 10: Image Segmentation
    - 10.1-10.4 (Week 10)