Lesson 5: Morphology

1. Introduction
2. Expansion and contraction
3. Dilation and erosion
4. Opening and closing
5. Skeletons
6. Distance maps
1/6. Introduction

- **Morphology:** analysis of the shape of connected components.
  - **Algebraic** operators applicable to binary images in order to extract components useful in representing shape
    - Contours
    - Convex hull
    - Skeletons
- **Goals:**
  - Image simplification
  - Elimination of irrelevances
  - Preservation of useful characteristics
- **Definitions:** Let $A$ and $B$ be binary images, $p$ and $q$ two pixels with indices $[i, j]$ and $[k, l]$ respectively, and $\Omega$ the universal binary image.
  - **Union:**
    $$A \cup B = \{p | p \in A \lor p \in B\}$$
  - **Intersection:**
    $$A \cap B = \{p | p \in A \land p \in B\}$$
  - **Complement:**
    $$\bar{A} = \{p | p \in \Omega \land p \notin A\}$$
  - **Difference:**
    $$A - B = A \cap \bar{B}$$
  - **Translation:**
    $$A_p = \{a + p | a \in A\}$$
    - Vectorial sum:
      $$p + q = [i + k, j + l]$$
    - Vectorial difference:
      $$p - q = [i - k, j - l]$$
  - **Reflex:**
    $$A' = \{-p | p \in A\}$$
  - **Parallelizable techniques**
Introduction

• Example: thresholding

\[ A \]

\[ B \]

\[ C \]

\[ B \cup C \]

\[ D \]

Not touching borders

Size filter
2/6. Expansion and contraction

- Transformations to turn foreground pixels into background and vice-versa.

- **Expansion**: turn a pixel from 0 to 1 if any neighbor is 1.

- **Contraction**: turn a pixel from 1 to 0 if any neighbor is 0.

- Non-commutative

- Neither invertible:
  
  \[ S^k \text{ expanded } k \text{ times} \]
  
  \[ S^{-k} \text{ contracted } k \text{ times} \]

- Expanding a blob is equivalent to contracting the background

\[
(S^m)^{-n} \neq (S^{-n})^m \neq S^{m-n}
\]

- Neither invertible:

\[
S \subset (S^k)^{-k}
\]

\[
S \supset (S^{-k})^k
\]
Expansion y contraction

1. Expansion:

2. Contraction:

1. Contraction:

2. Expansion:

- **Expansion + Contraction:** elimination of undesired holes (salt noise).
- **Contraction + Expansion:** Elimination of noise (pepper noise).
3/6. Dilation y erosion

- **Dilation:** union of translations of an image $A$ by each pixel of image $B$, called *structural element*:

  $$A \oplus B = \bigcup_{b_i \in B} A_{b_i}$$

- **Erosion** inverse operation

  $$A \ominus B = \{p \mid B_p \subseteq A\}$$

Asociative y commutative?
Object connection

Original

Dilated twice

Eroded twice

Dilated and eroded four times
Object separation

a. Original
b. Eroded twice
d. Eroded 7 times
e. Dilated four times with XOR
f. Dilated seven times with XOR
g. Dilated nine times with XOR
h. AND with original image
Contour extraction

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

A

A \ominus B

\[ \beta(A) \]

A \ominus B

A

Z 2208 AH

Z 2208 AH

Z 2208 AH
Region filling

\[ X_0 = p \]
\[ X_k = (X_{k-1} \oplus B) \cap \overline{A} \]

\[ X_k = X_{k-1} \]

\[ X_0 \]
\[ X_1 \]
\[ X_2 \]
\[ X_6 \]
\[ X_7 \]
\[ X_7 \cup A \]
The total area might be less sensitive to noise, although the Euler number might be more discriminating.
Hit or Miss

- Selection of pixels with some particular properties (corners, isolated, border).
  \[ B = (J, K), \quad J \cap K = \emptyset \]
  \[ A \otimes B = (A \oplus J) \cap (A \oplus K) \]

- Example: sup. rig. corners
- \( J \): description of object pixels
- \( K \): description of background pixels
Computation of Euler number

\[ Eu = C - H \]

• Given a closed polygonal line, the sum of its angles should be \( \pm 360^\circ \)

• The Euler number is equal to the number of convex corners minus the number of concave corners, all divided by four:

\[ \frac{N_c - N_o}{4} = 0 \]

\( C \) and \( H \) cannot be computed separately
4/6. Opening and closing

- **Opening**: erosion + dilation with the same element
  \[ A \circ K = (A \oplus K) \oplus K \]
  - Eliminates all regions too small to contain the structural element

- **Closing**: dilation + erosion with the same element
  \[ A \bullet K = (A \oplus K) \ominus K \]
  - Fills all holes and cavities smaller than the structural element

- **Idempotent**
  \[ A \circ K \circ K = A \circ K \]
  \[ A \bullet K \bullet K = A \bullet K \]
Applications

• Template matching

1 objects in the real object models. This will make object recognition effortless. Thus, we will discuss different techniques that have been used. We will discuss different techniques that have been used.

\[ A \circ B \]

\[ A \]

\[ B \]

• Reconstruction

Original

Thresholded

Closing
Applications

- **Smoothing:**
  - Opening: smooth convex corners.
  - Closing: smooth concave corners.

- **Morphologic filtering:** B is a disc of size $\geq$ all noise components.

\[
(A \circ B) \bullet B
\]
5/6. Skeletons

- **Skeletons, axis of symmetry $S^*$:** geometric place of the centers of all at least bi-tangent circles.

- $S^*$ is a compact representation of $S$; it represents the *shape* of the region.

- Highly sensitive to noise.

The frontier is also a compact representation of shape.
Thinning

\[ A \odot B^i = A - (A \otimes B^i) \]

\[ A \odot B^{1,2} = (A \odot B^1) \odot B^2 \]

\[ X_0 = A \]
\[ X_k = X_{k-1} \odot B^{\{1,\ldots,n\}} \]

Final result
Thinning

Original

Iteration 1

Iteration 3

Iteration 5

Iteration 7
6/6. Euclidean distance maps (EDM)

- Image representing the smallest distance of e/pixel to the background.

There are several possible definitions for distance
Distance measurements

- Fundamental properties:

  \( \forall p, q, r : \)

  1. \( d(p, q) \geq 0, \)
  
  \[ d(p, q) = 0 \iff p = q \]

  2. \( d(p, q) = d(q, p) \)

  3. \( d(p, r) \leq d(p, q) + d(q, r) \)

  \[ d([i_1, j_1], [i_2, j_2]) = \]

  1. Euclidean:
  
  \[ \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \]

  2. Manhattan:
  
  \[ |i_1 - i_2| + |j_1 - j_2| \]

  3. Chess:
  
  \[ \max(|i_1 - i_2|, |j_1 - j_2|) \]

  Euclidean is closest to the real case; Mosr costly to compute
Obtaining distance maps

\[ f^0[i, j] = B[i, j] \]
\[ f^m[i, j] = f^0[i, j] + \min \left( f^{m-1}[u, v] \right) \]
\[ \forall [u, v] : d([u, v], [i, j]) = 1 \]

- **Iteration 0:** original image.
- **Iteration 1:** All pixels not adyacent to background change to 2.
- **Next iterations:** pixels farther from background change.
- No pixels changes when the distances to all have been computed.