Mathematical Morphology: Greyscale, Recursive Operations

CS 650: Computer Vision

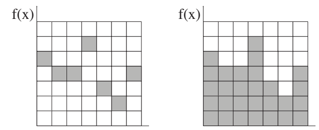
Umbras, Functions, and Images

The *umbra* of a 1-D function/signal $f(x)$ is the set of all positions/values (x, v) such that value v is less than or equal to $f(x)$:

$$\{(x, v) \mid v \leq f(x)\}$$

or for 2-D images I:

$$\{(x, y, v) \mid v \leq I(x, y)\}$$



1-D Signal Umbra

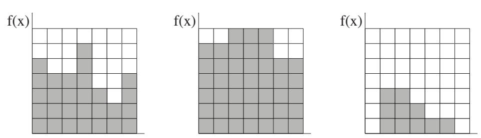
Greyscale Morphology

- Greyscale morphology involves binary morphology of the umbras:

$$\text{Umbra}\left(A \oplus_g B\right) = \text{Umbra}\left(A\right) \oplus \text{Umbra}\left(B\right)$$

$$\text{Umbra}\left(A \ominus_g B\right) = \text{Umbra}\left(A\right) \ominus \text{Umbra}\left(B\right)$$

- Example (3-wide structuring element of all 1s):



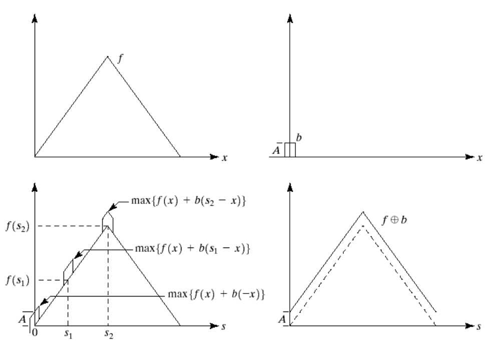
Original Dilation Erosion

Greyscale Dilation

$$f \oplus_g g = \max_z \{f(x - z) + g(z)\}$$

1. reflect the structuring element,
2. position the structuring element at position x
3. pointwise add the structuring element over the neighborhood, and
4. take the maximum of that result over the neighborhood.

Greyscale Dilation



a b
c d

FIGURE 9.27 (a) A simple function. (b) Structuring element of height A. (c) Result of dilation for various positions of sliding b past f. (d) Complete result of dilation (shown solid).

Greyscale Erosion

$$f \ominus_g g = \min_z \{f(x + z) - g(z)\}$$

1. position the structuring element at position x
2. pointwise subtract the structuring element over the neighborhood, and
3. take the minimum of that result over the neighborhood.

Greyscale Erosion

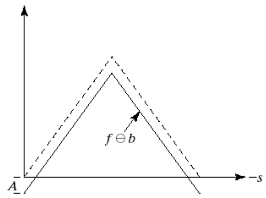


FIGURE 9.28 Erosion of the function shown in Fig. 9.27(a) by the structuring element shown in Fig. 9.27(b).

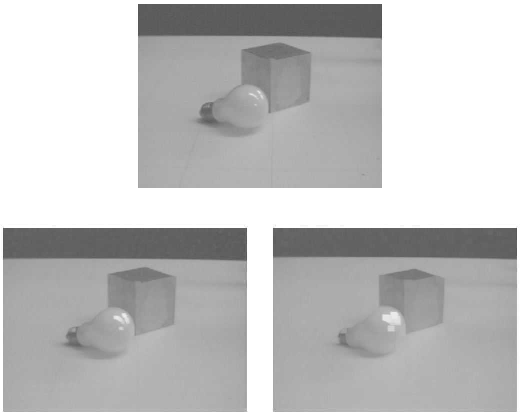
Greyscale Dilation and Erosion: Images

For two-dimensional images, these become

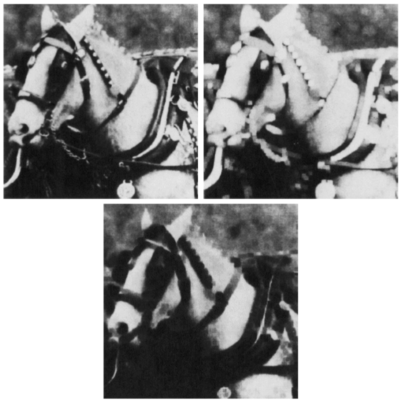
$$f \oplus_g g = \max_{a,b} \{ f(x-a, y-b) + g(a,b) \}$$

$$f \ominus_g g = \min_{a,b} \{ f(x+a, y+b) - g(a,b) \}$$

Example: Greyscale Dilation



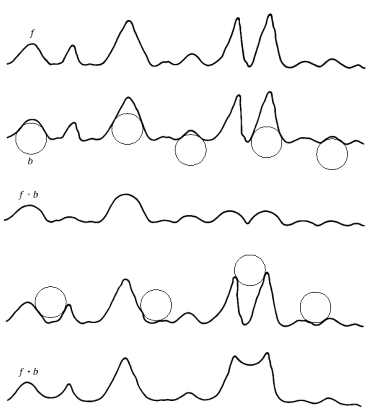
Example: Greyscale Dilation and Erosion



a b
c

FIGURE 9.29 (a) Original image. (b) Result of dilation. (c) Result of erosion. (Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)

Greyscale Opening and Closing



a
b
c
d
e

FIGURE 9.30 (a) A gray-scale scan line. (b) Positions of rolling ball for opening. (c) Result of opening. (d) Positions of rolling ball for closing. (e) Result of closing.

Greyscale Opening

Erosion followed by dilation:

$$f \circ_g g = (f \ominus_g g) \oplus_g g$$

or for a constant structuring element:

$$f \circ_g g = \max_{(a \in g)} \min_{(b \in g)} f(x - a + b)$$

Example:

	6	7	9	5	6	6	6	4	5	6	7	2
min	0	6	5	5	5	6	4	4	4	5	2	0
max	6	6	6	5	6	6	6	4	5	5	5	2

Greyscale Closing

Dilation followed by erosion:

$$f \bullet_g g = (f \oplus_g g) \ominus_g g$$

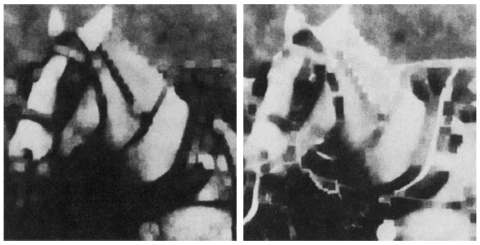
or for a constant structuring element:

$$f \bullet_g g = \min_{(a \in g)} \max_{(b \in g)} f(x + a - b)$$

Example:

	6	7	9	5	6	6	6	4	5	6	7	2
max	7	9	9	9	6	6	6	6	6	7	7	7
min	6	7	9	6	6	6	6	6	6	6	7	2

Greyscale Opening and Closing



a b

FIGURE 9.31 (a) Opening and (b) closing of Fig. 9.29(a). (Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)

Duality of Greyscale Opening and Closing

Opening a greyscale image is the same as closing its inverse image, and vice versa:

$$-(f \circ_g g) = -f \bullet_g \breve{g}$$

$$-(f \bullet_g g) = -f \circ_g \breve{g}$$

Noise Removal - Approximating Median Filtering

$$g(x) = \begin{cases} (f \circ_g k)(x) & \text{if } \left|(f \circ_g k)(x) - f(x)\right| \geq \left|(f \bullet_g k)(x) - f(x)\right| \\ (f \bullet_g k)(x) & \text{otherwise} \end{cases}$$

- If in a monotonic neighborhood, use opening or closing because neither have any effect.
- If at a neighborhood maximum, trim it off using an opening.
- If at a neighborhood minimum, fill it in using a closing.

Thickening

- One can *thicken* a binary object by finding targeted missing points and adding them:
  - Use hit-and-miss to find them
  - Union them into the object

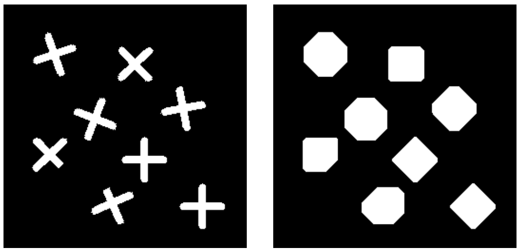
$$A \odot (J, K) = A \cup (A \otimes (J, K))$$

Example: Thickening to Produce a Convex Hull

- The *convex hull* of an object is the minimal convex shape that encompasses the object
  - Design hit-and-miss operators to detect concavities, and use thickening to fill them in
  - Apply recursively until convergence (no more concavities)
- Structuring element for a 45-degree convex hull: (plus the seven other rotations/reflections of this)

1	1	x
1	0	x
1	x	0

Example: Thickening to Produce a Convex Hull



Before After

Mathematical Morphology: Greyscale, Recursive Operations
└Recursive Morphology
└Thickening

Example: Thickening to Produce a Convex Hull

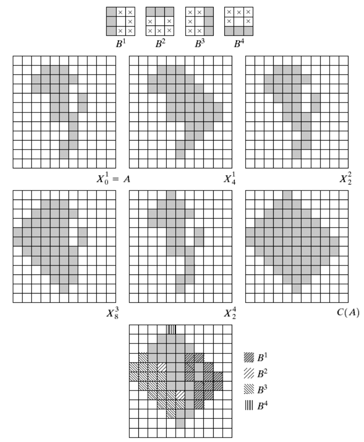


FIGURE 9.19 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

Mathematical Morphology: Greyscale, Recursive Operations
└Recursive Morphology
└Thinning

Thinning

- One can *thin* a binary object by finding targeted object points and removing them:

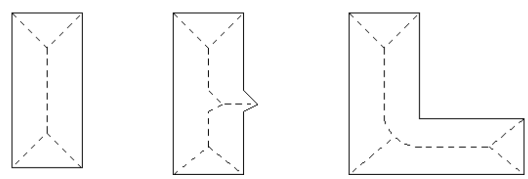
$$A \oslash (J, K) = A - (A \otimes (J, K))$$

(Finds target points $A \otimes (J, K)$ *and removes them*)

Mathematical Morphology: Greyscale, Recursive Operations
└Recursive Morphology
└Thinning

Example: Thinning to Produce a Skeleton

- A *skeleton* is a structure that is
  - Single-pixel thin
  - Lies in the "middle" of the object
  - Preserves the topology of the object



(We'll come back to this more when we discuss shape representations.)

Mathematical Morphology: Greyscale, Recursive Operations
└Recursive Morphology
└Thinning

Example: Thinning to Produce a Skeleton

- You can derive a *skeleton* of an object morphologically as follows:
  - Design hit-and-miss operators that find boundary points that can safely be removed without breaking the object in two
  - Apply recursively to strip layer by layer from the exterior of the shape until convergence
- Structuring element:
  (plus the four other rotations of this pair)

0	0	0
x	1	x
1	1	1

x	0	0
1	1	0
x	1	x

Mathematical Morphology: Greyscale, Recursive Operations
└Recursive Morphology
└Thinning

Example: Thinning to Produce a Skeleton

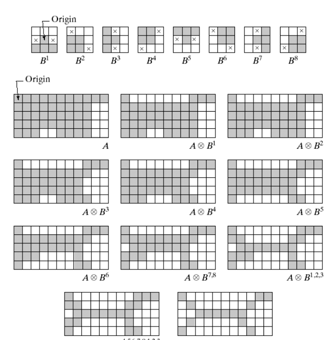
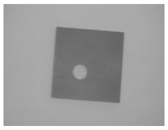
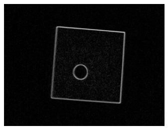
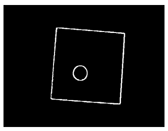


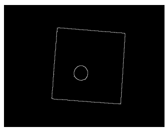
FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m-connectivity.

Example: Thinning to Produce a Skeleton


Original Image


Edge Detector (Sobel)


Thresholded


Thinned

Conditional Dilation

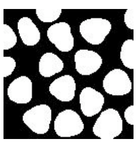
- Conditional dilation involves dilating shapes in one image A then masking it by another image I:

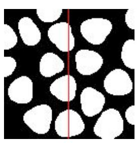
$$A \oplus |_I B = (A \oplus B) \cap I$$

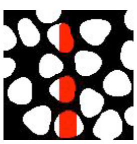
- Application: finding specific points, then "growing" back their original connected components.
  - I original image
  - A morphologically-reduced image that you want to "grow back"
  - B 3×3 structuring element containing all 1s.

$$J_0 = A; \qquad J_i = J_{i-1} \oplus |_I B = (J_{i-1} \oplus B) \cap I$$

Example: Conditional Dilation


Original


Target


Partial


Filled

Recursive Morphology

- We've seen three separate algorithms that each involve
  1. Apply a morphological operator to make some desired change
  2. Repeat until convergence (no more change)
- These are called *recursive* operators:
  - Convex hull (thickening)
  - Skeletonization (thinning)
  - Connected component finding from target point (conditional dilation)