# EE 4211 Computer Vision

## Lecture 4B: Morphology

Semester A, 2020-2021

# Lecture Outline

- Morphological Algorithms
  - Hit or Miss Transform
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Skeletons

# 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

$$A \circledast B = \left(A \ominus B_1\right) \cap \left(A^c \ominus B_2\right)$$

$$B = \left(B_1, B_2\right)$$

$B_1$ : object $\quad$ $B_2$ : background

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

# Hit-or-Miss Transform

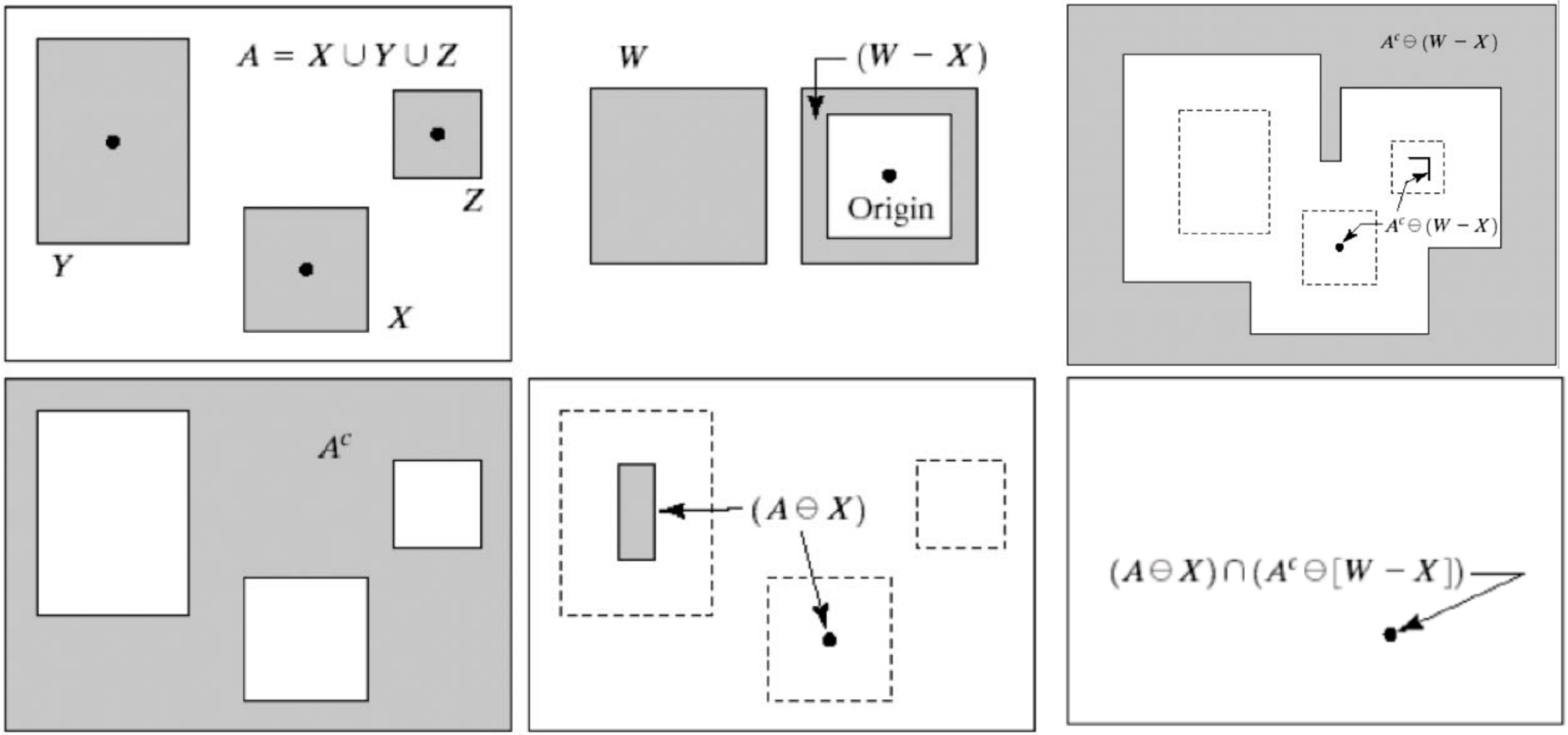
- Steps

Perform an erosion $A \Theta B_1$ 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



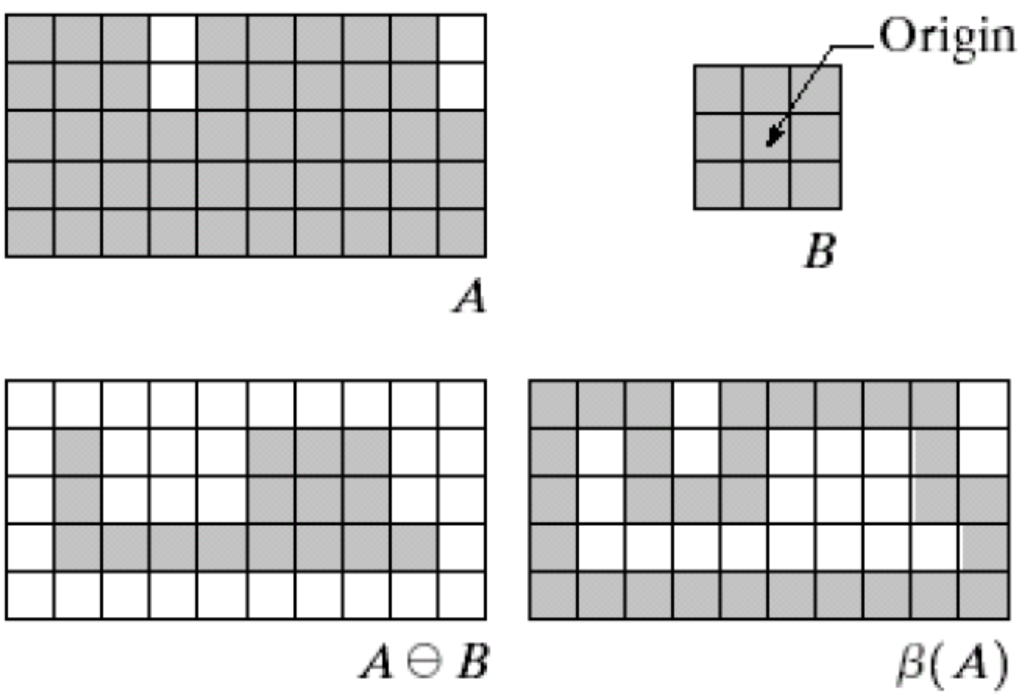
# Lecture Outline

- Morphological Algorithms
  - Hit or Miss Transform
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Skeletons

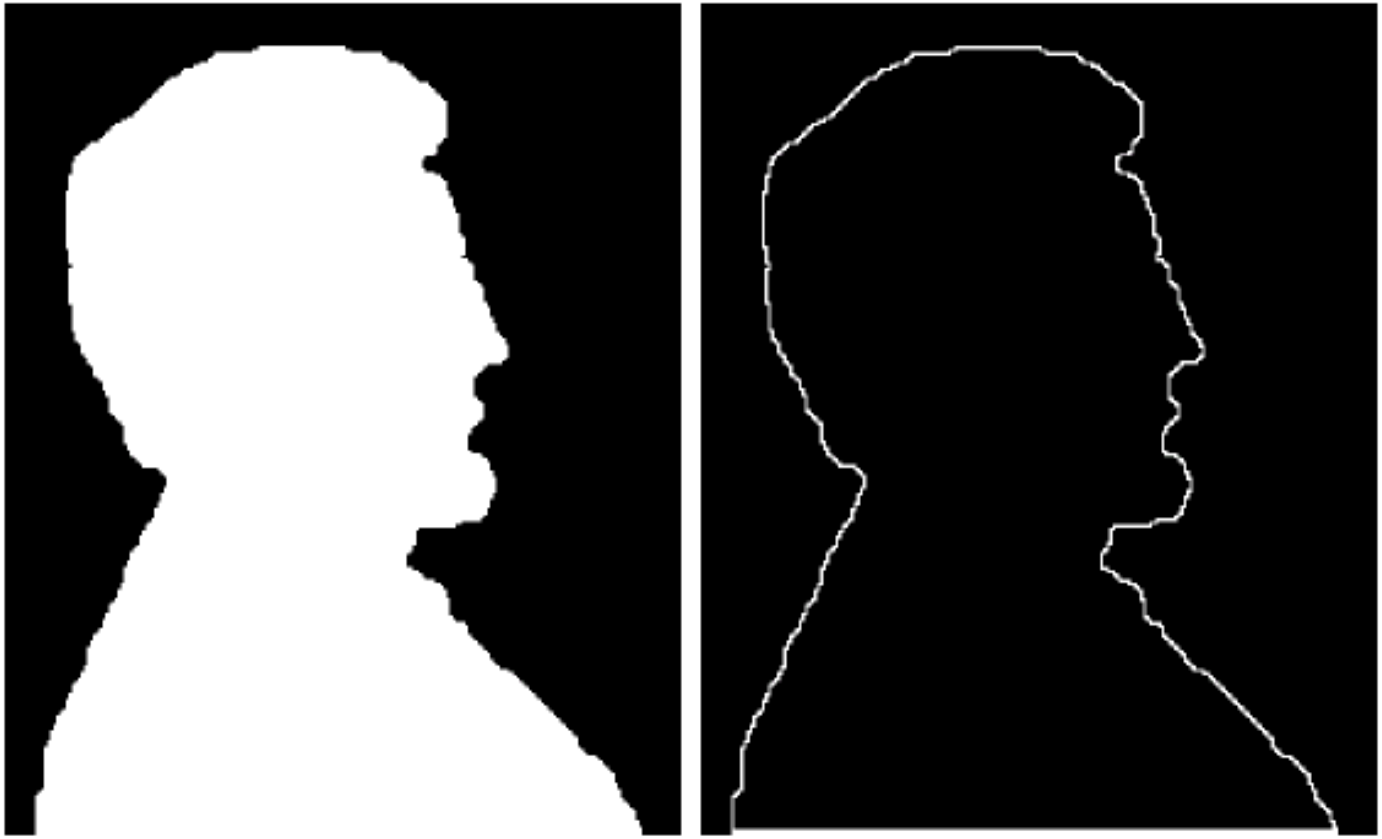
# Boundary Extraction

- A boundary of a set A, denoted by β(A), is obtained by eroding A by B, then perform the set difference between A and its erosion:

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



# Boundary Extraction



# Lecture Outline

- Morphological Algorithms
  - Hit or Miss Transform
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Skeletons

# 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

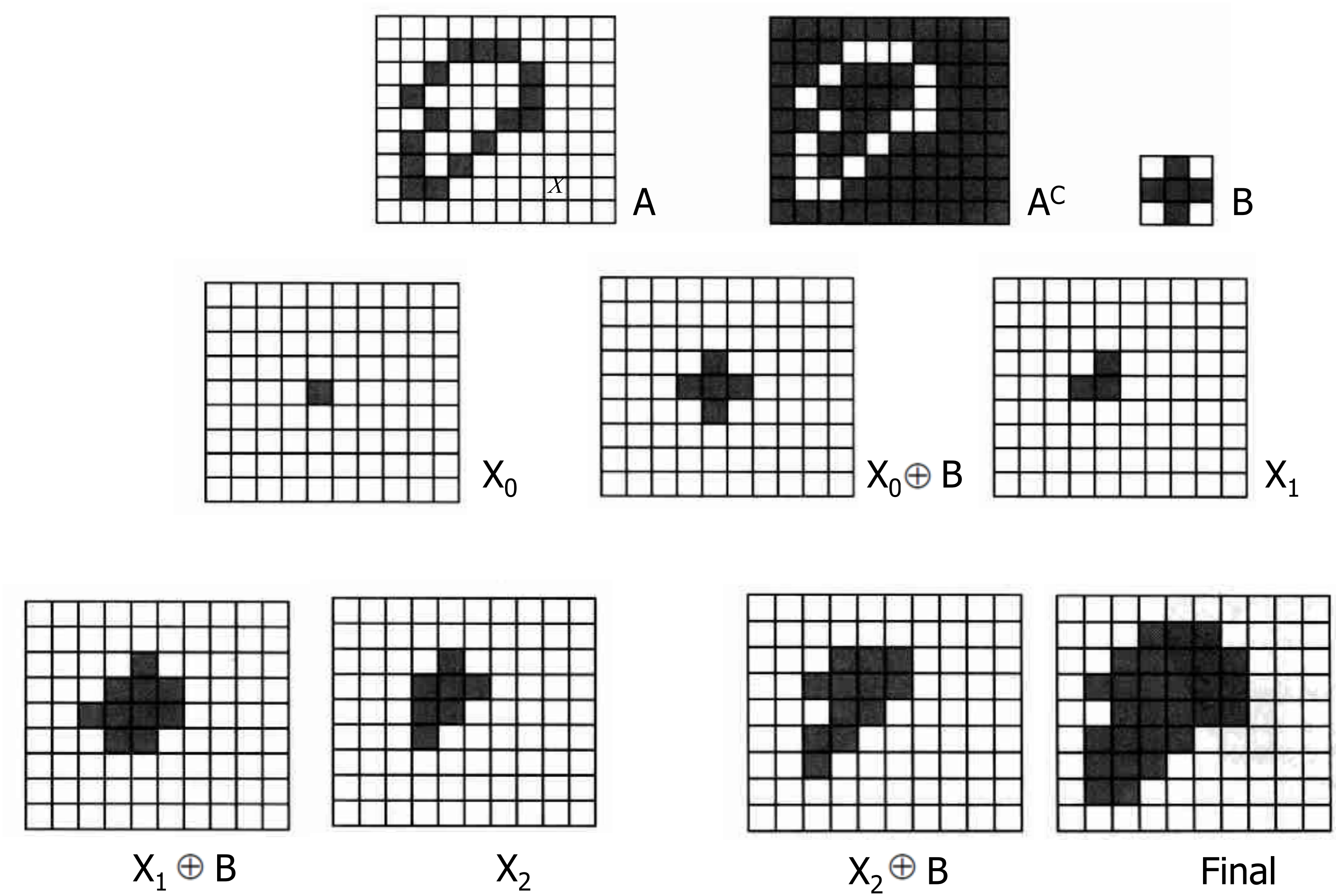
# Region Filling

- Start with a point inside the region
- Repeatedly dilate
- At each step, set to zero the points corresponding to the region

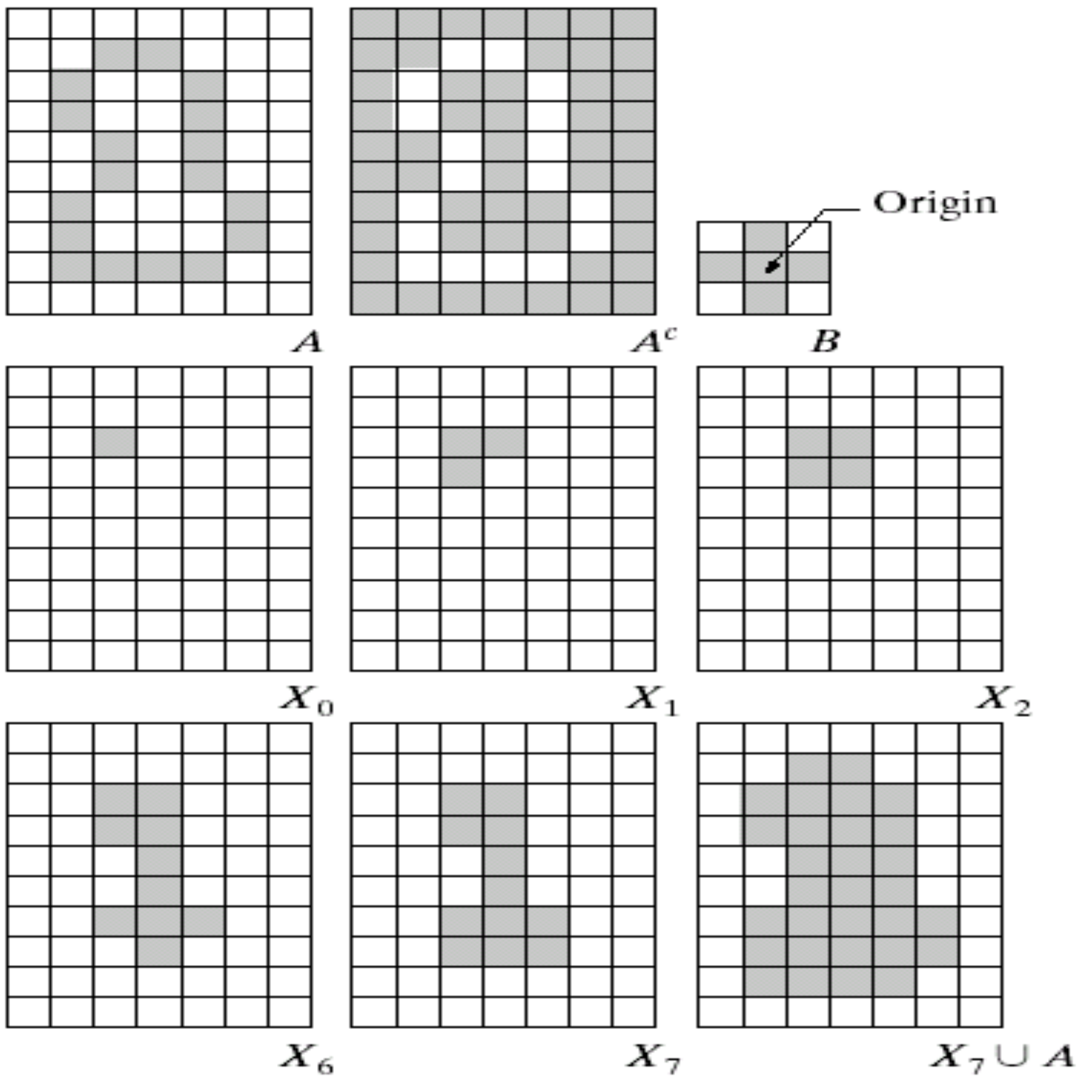
$$X_k = (X_{k-1} \oplus B) \cap A^c, \quad k = 1,2,3,\ldots$$

- $A^c$ is the complement of A
- Stop when no more changes

# Region Filling



# Region Filling



# Region 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.

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

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.

# Lecture Outline

- Morphological Algorithms
  - Hit or Miss Transform
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Skeletons

# 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

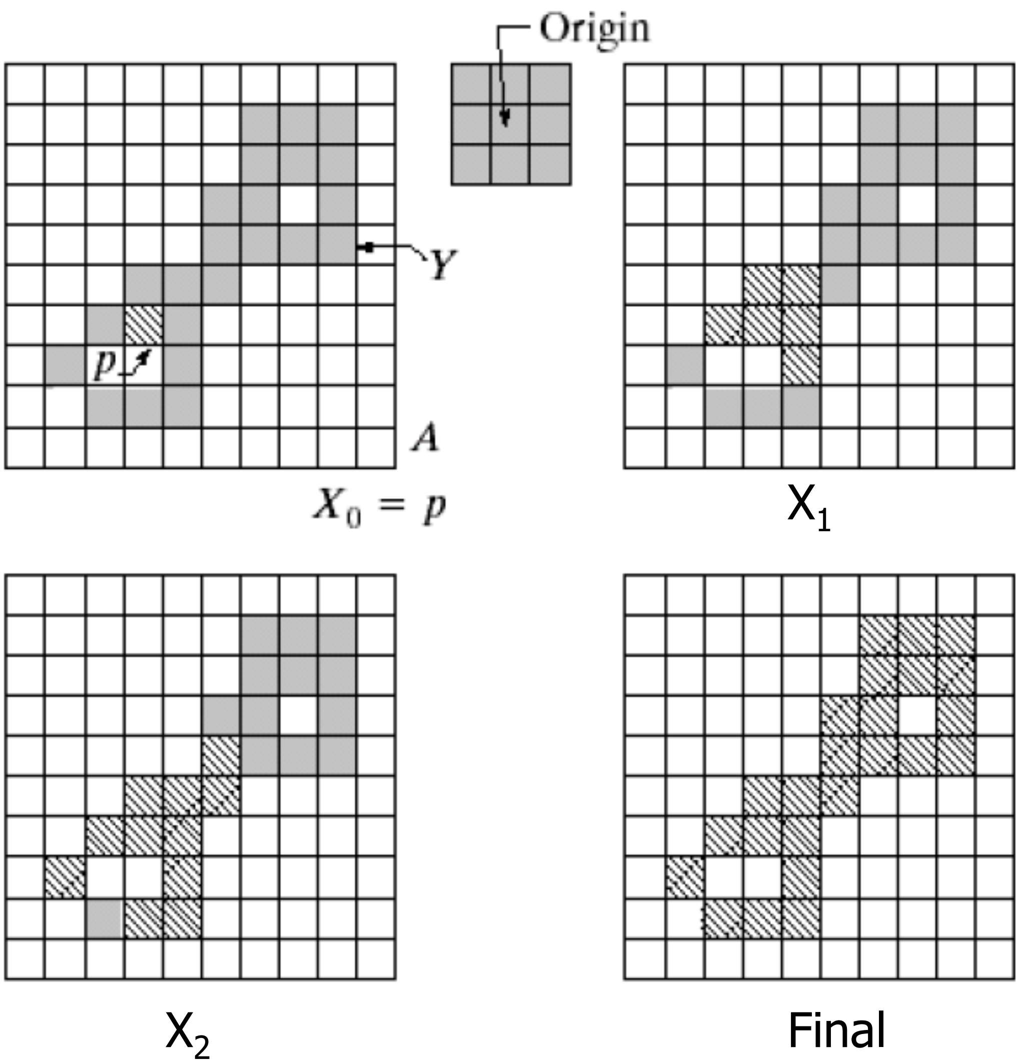
- Let A be the set of 8 connected boundary points of a region
- Start with a point inside the region, Repeatedly dilate
- At each step, set to zero the points corresponding to the region boundary

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

- Stop when no more changes

- Note its similarity to Hole Filling algorithm

# Extraction of Connected Components




Y: connected component in set A,

*p*: a known point in Y

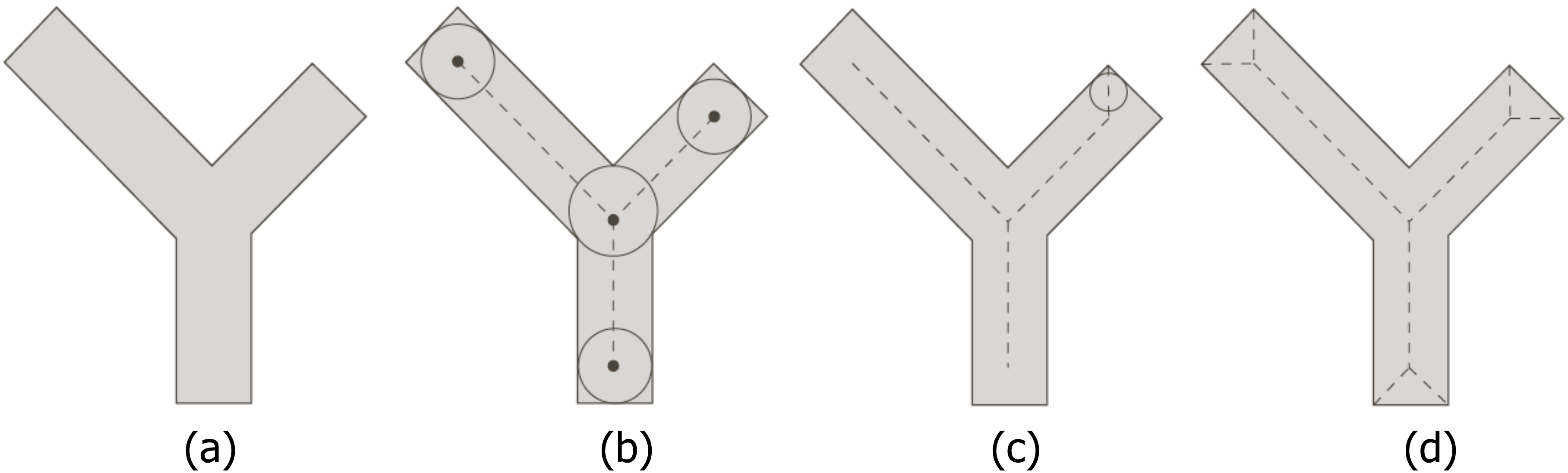
$$X_0 = p$$
$$X_k = (X_{k-1} \oplus B) \cap A$$
$$\text{if } X_k = X_{k-1}$$
$$\text{then } Y = X_k$$

# Lecture Outline

- Morphological Algorithms
  - Hit or Miss Transform
  - Boundary Extraction
  - Hole Filling
  - Connected Components
  - Skeletons

# Skeletons (Skeletonizing)

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



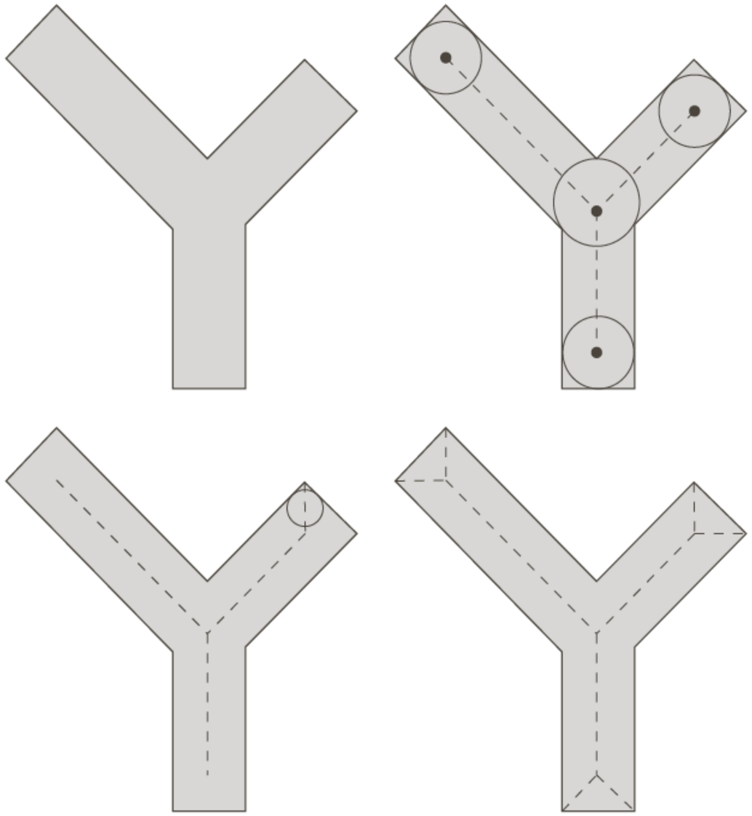

(a) Set A
(b) Various positions of maximum disks with centers on the skeleton of A
(c) Another maximum disk on a different segment of the skeleton of A
(d) Complete Skeleton

# Skeletons (Skeletonizing)

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

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

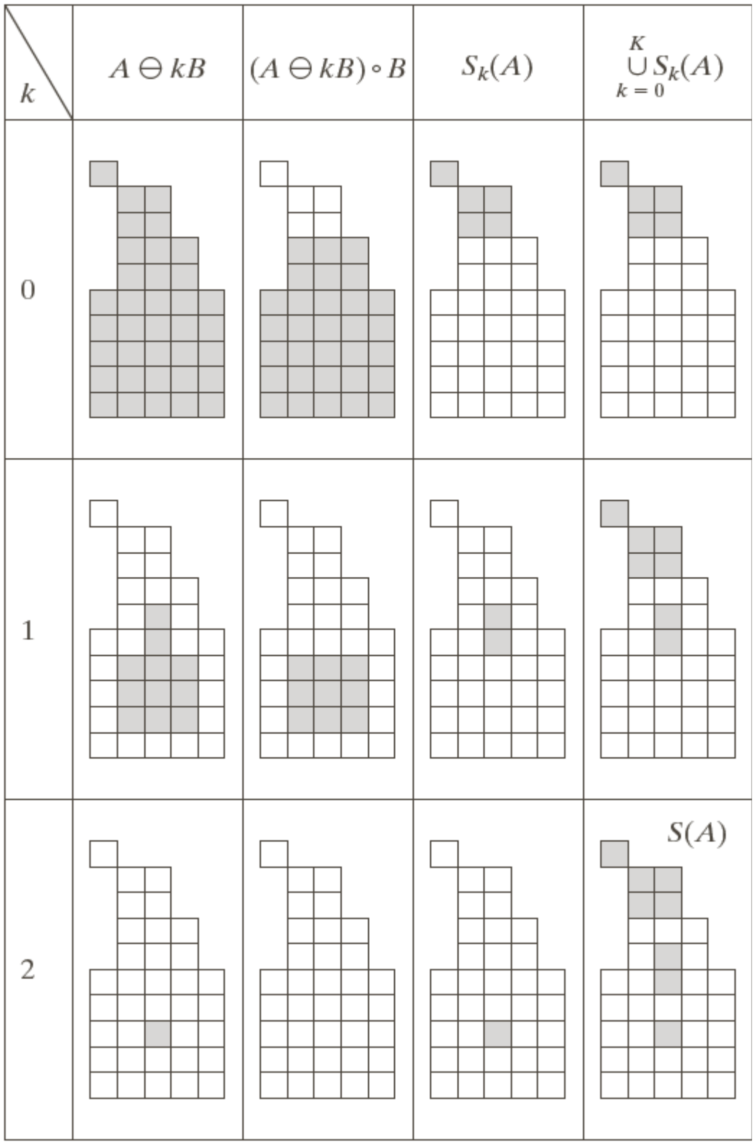
$$S_k(A) = (A \Theta kB) - \left[ (A \Theta kB) \circ B \right]$$

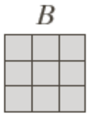
where B is the SE, and (AΘ*k*B) indicates *k* successive erosions of A:

$$(A \Theta kB) = ((\ldots((A \Theta B) \Theta B) \Theta \ldots) \Theta B)$$

*k* times, and K is the last iterative step before A erodes to an empty set.

# Skeletons (Skeletonizing)



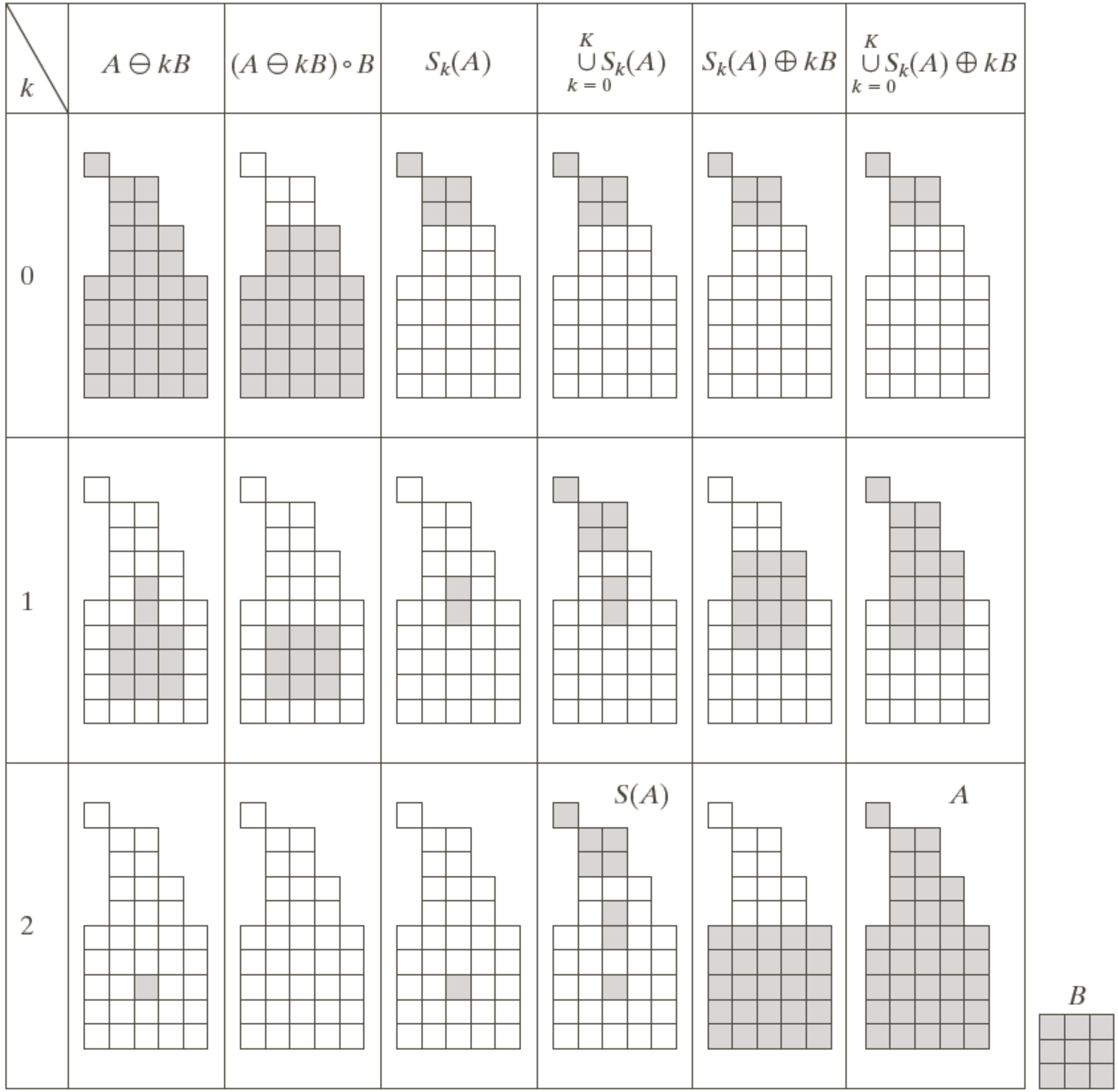
# Skeletons (Skeletonizing)

A can be reconstructed from the subsets by using

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

where $S_k(A) \oplus kB$ denotes $k$ successive dilations of A.

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B)\ldots \oplus B)$$




Skeleton Reconstructed set