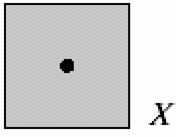
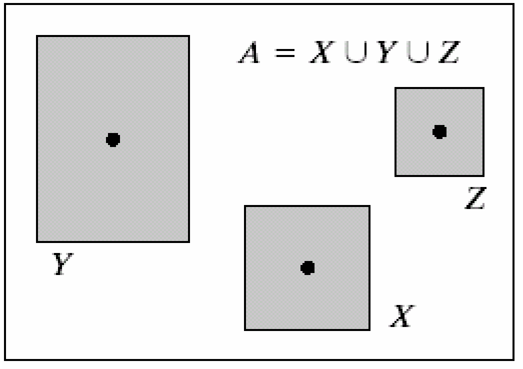
Chapter 9

Morphological Image Processing

Hit or Miss transformation

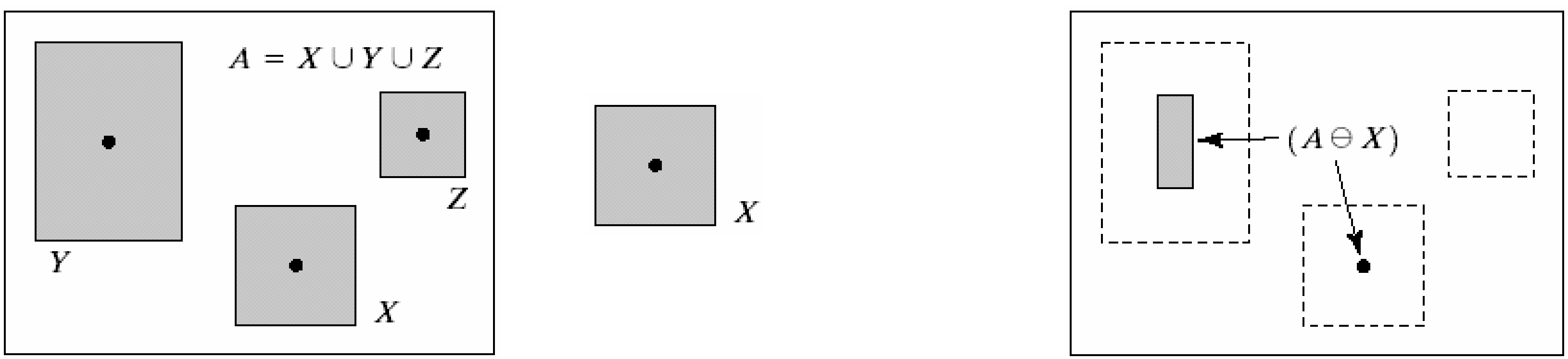
- Finds locations where the given pattern matches with the pattern in the image
  - Can be used to find and locate simple objects.
- Find whether 

is present in the following image



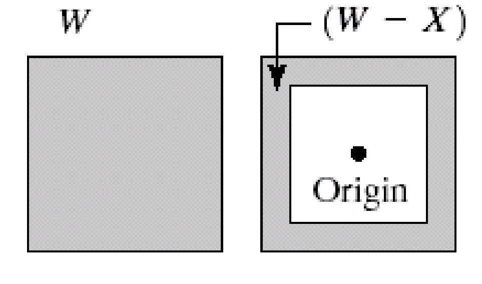
Hit or Miss transformation

- If we are interested only in finding locations where the pattern X is present in the image, then erosion is enough.



Hit or Miss transformation

- This finds points where the pattern is *contained* in the neighborhood.
- We are interested in finding points where the pattern exactly matches the neighborhood, i.e, there should be background pixels at points outside the given pattern.
- This tells us that we should also try and find points where a frame of background pixels match the neighborhood surrounding the pattern pixels.



Hit or Miss transformation

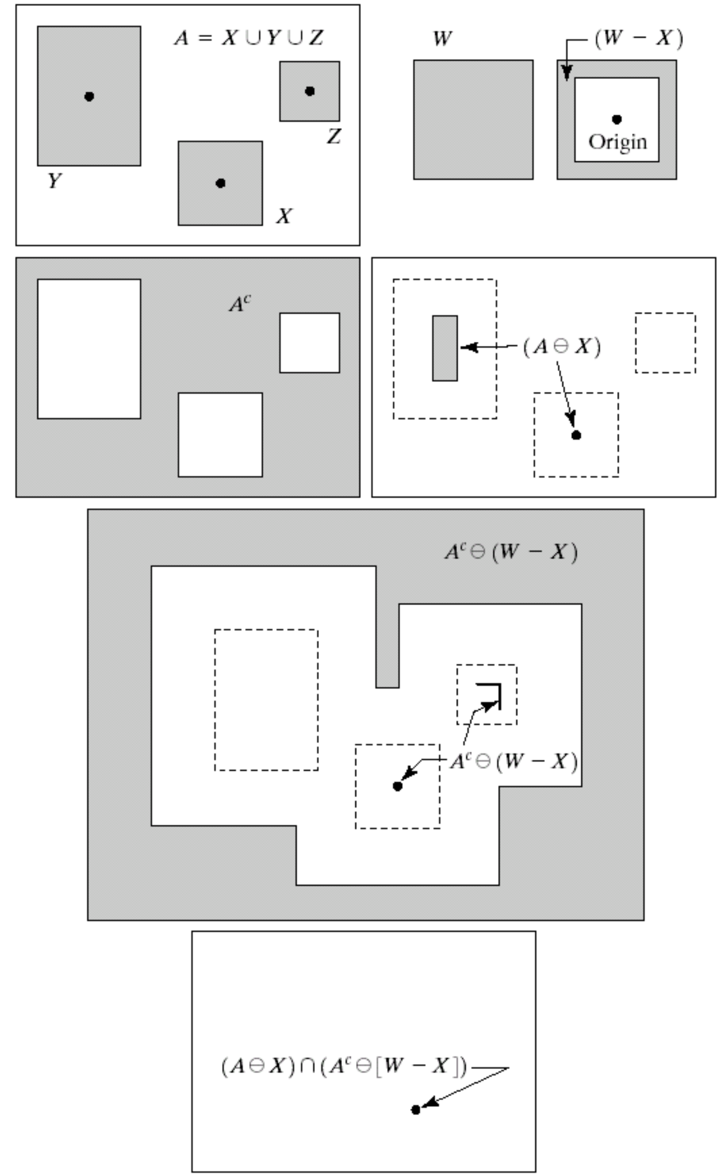
- Look for points where X matches with A *and* (W – X) matches with A^c.

$$A \circledast B = (A \ominus B) \cap (A^c \ominus (W - X))$$

- In general, let B = (B_1,B_2)
  - B1: Object set
  - B2: Background set

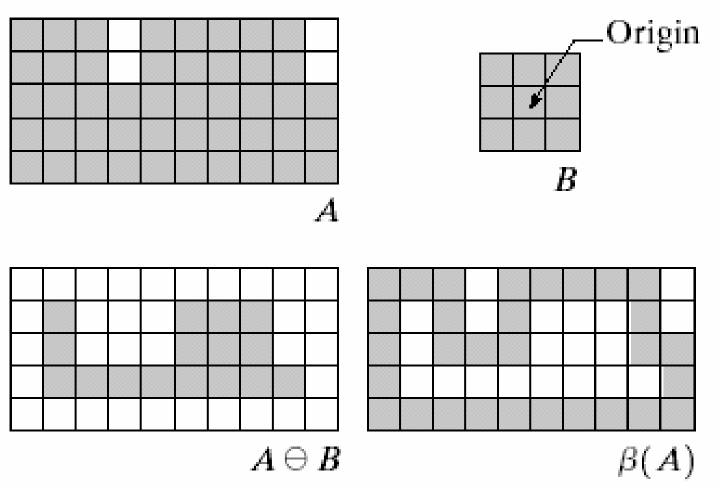
$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- Can be used to identify disjoint locations of the given pattern.

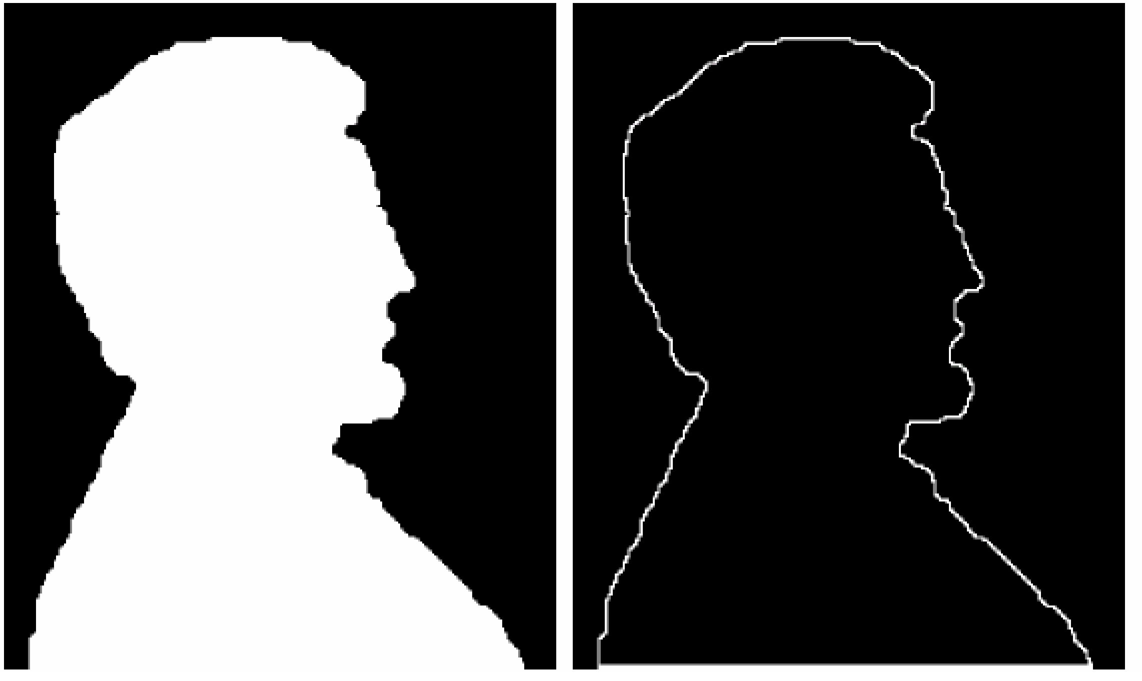


Boundary Extraction

- Given a set A, the erosion of A with a structuring element B shrinks A a *little*.
  - $A \ominus B \subset A$
- The set difference $A - (A \ominus B)$ then gives the boundary $\beta(A)$, of A.



Boundary Extraction



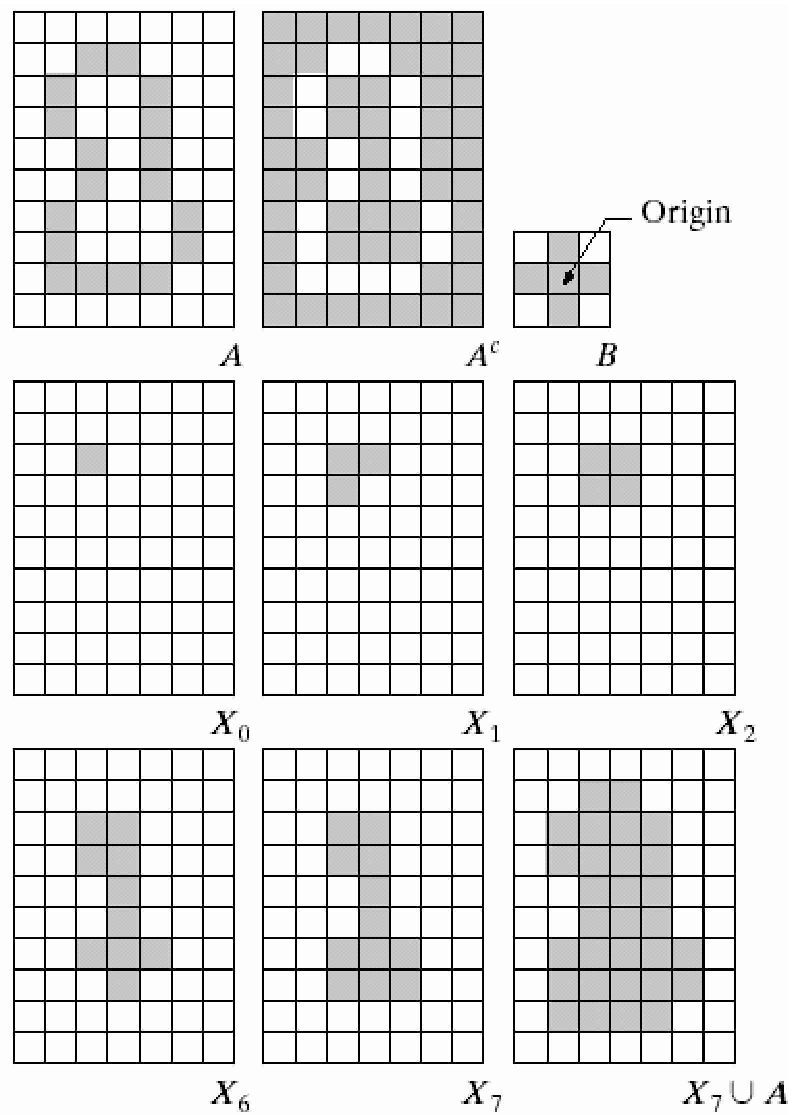
Region filling

- Given a boundary of a set and a point (p) inside the boundary, the objective is to fill the entire boundary.
- We should start working from the given interior point, dilating the set till it meets some condition.

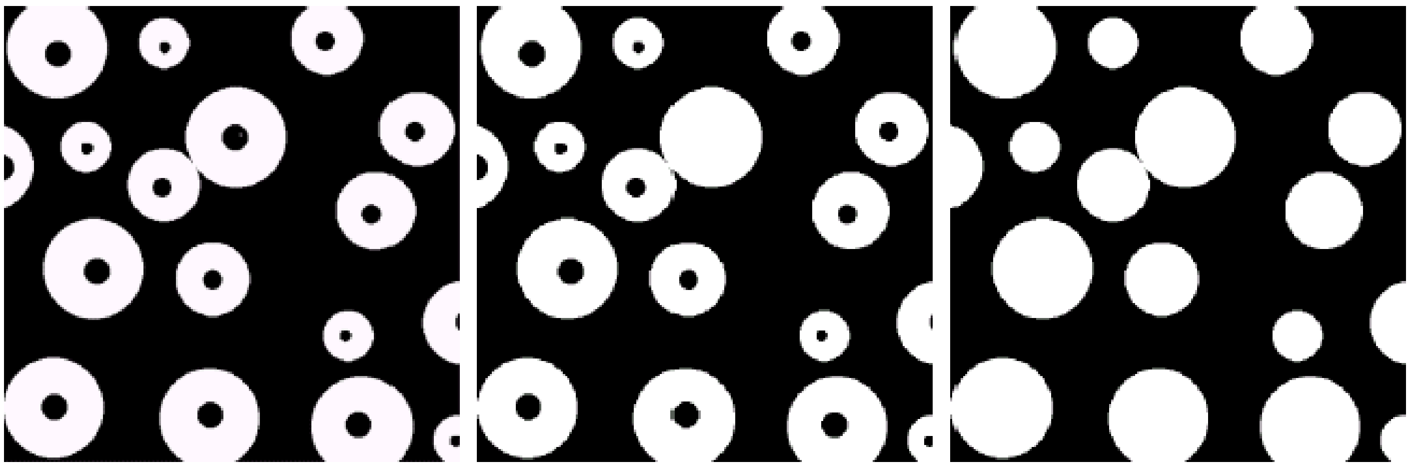
$$\text{While } X_k \neq X_{k-1}, \text{ Repeat } X_k = (X_{k-1} \oplus B) \cap A^c, \text{ with } X_0 = \{p\},$$

Region filling

$$\begin{aligned} \text{While } X_k &\neq X_{k-1}, \\ \text{Repeat } X_k &= (X_{k-1} \oplus B) \cap A^c, \\ \text{with } X_0 &= \{p\} \end{aligned}$$



Region filling



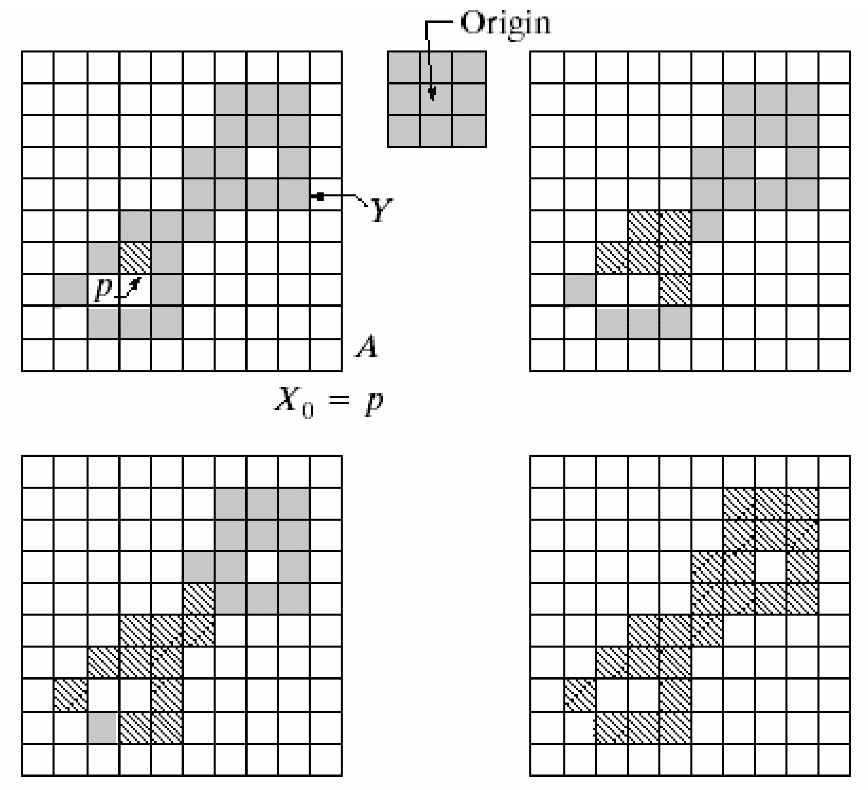
a b c

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.

Connected components

- Given a point p in a set Y, find all pixels connected to p, i.e., find the connected component of p in Y.

$$X_k = (X_{k-1} \oplus B) \cap Y$$



- Can be used to count number of objects in the image.

Thinning

- A typical application would be to process an image obtained after edge-detection.
- In natural images the edges are not sharp which gives a high gradient value in a band of pixels around the actual edge.
- We can use the thinning procedure to reduce the thickness of the edge.

Thinning

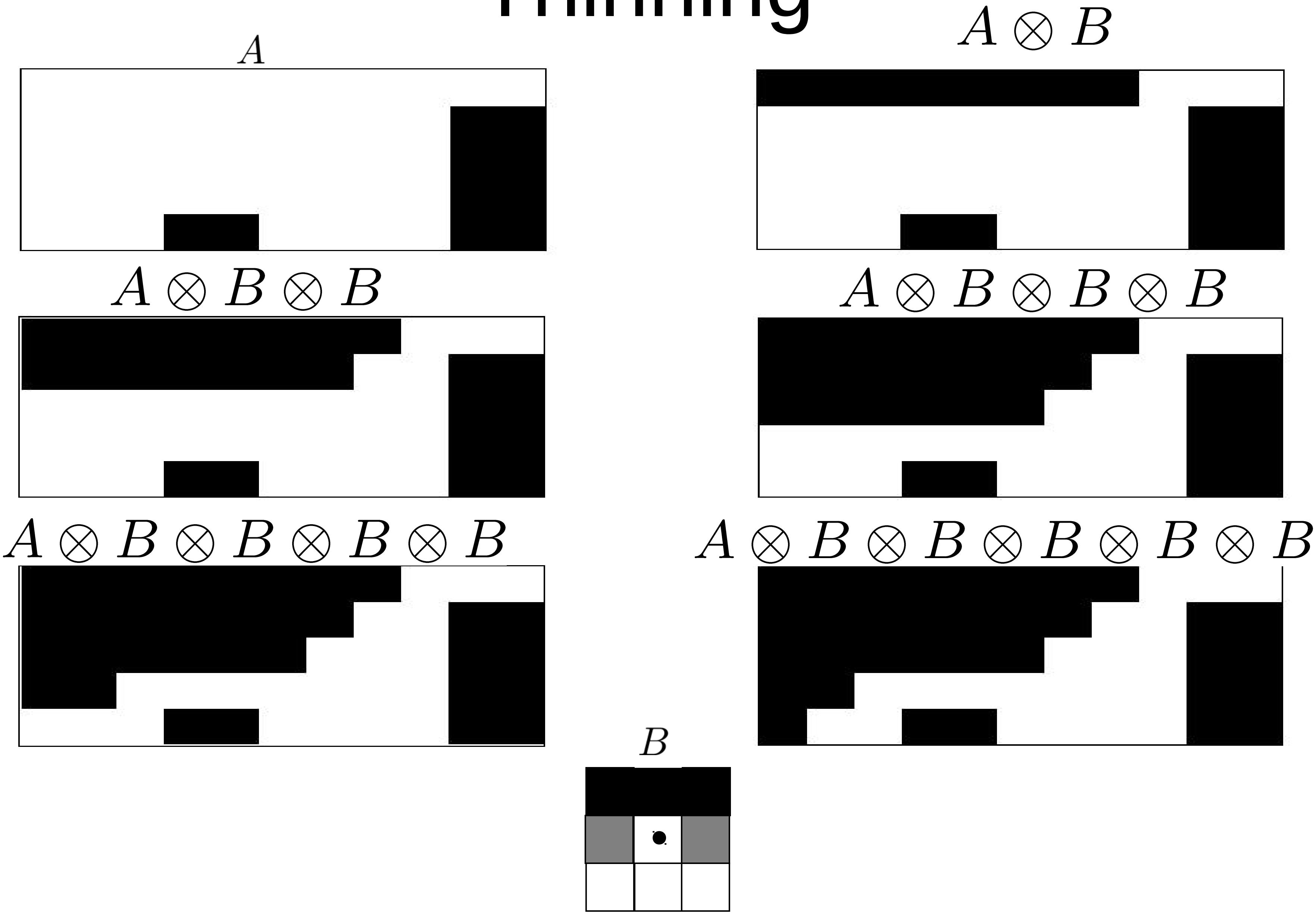
- The set of pixels with high gradient value forms the set of our interest.
- We want to remove pixels in the above set whose neighborhood contains both, edge and background pixels.
- This idea tells us that the structuring element should contain both foreground and background pixels, i.e. $B = B_1 \cup B_2$

Thinning

- Next step is to find pixel locations in the set of edge pixels whose neighborhood *matches* the structuring element B
  - Hit or Miss transformation!
- If the Hit or Miss transformation is successful at a pixel, this pixel should not be a part of the output set.
- $A \otimes B = A - (A \circledast B)$

  $A \otimes B = A \cap (A \circledast B)^c$

Thinning

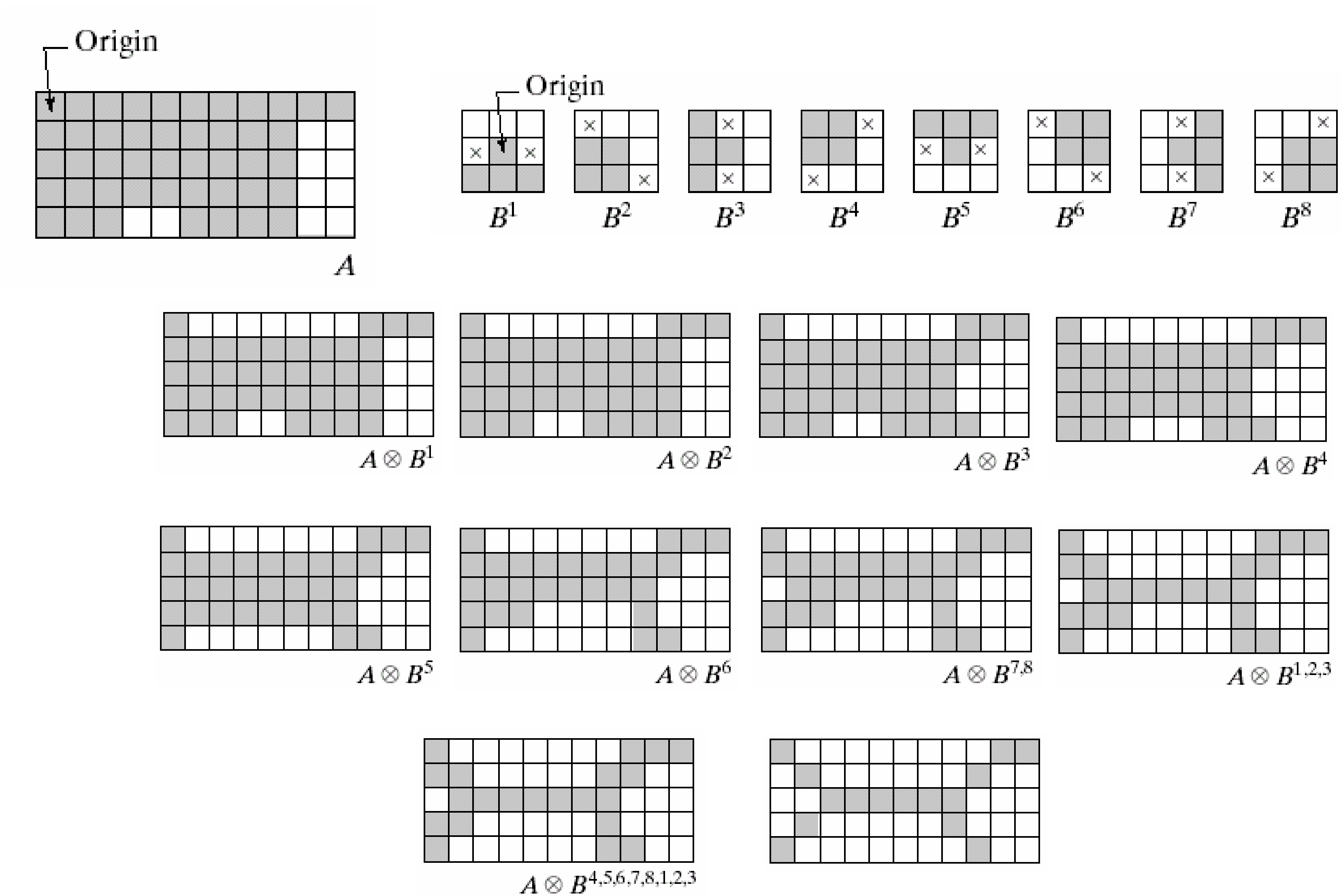

A
$A \otimes B$
$A \otimes B \otimes B$
$A \otimes B \otimes B \otimes B$
$A \otimes B \otimes B \otimes B \otimes B$
$A \otimes B \otimes B \otimes B \otimes B \otimes B$
B

Thinning

- Consider a horizontal edge that gives a horizontal band of pixels.
- We cannot use the same structuring element B for both, the top end and the bottom end of this band.
- We may need more then 1 structuring element: $B = \{B^1, B^2, \ldots, B^n\}$.
- Thinning of A by B:

$$A \otimes B = ((\ldots(A \circledast B^1) \circledast B^2) \ldots) \circledast B^n)$$

Thinning


Origin
A
Origin
B1
B2
B3
B4
B5
B6
B7
B8
A ⊗ B1
A ⊗ B2
A ⊗ B3
A ⊗ B4
A ⊗ B5
A ⊗ B6
A ⊗ B7,8
A ⊗ B1,2,3
A ⊗ B4,5,6,7,8,1,2,3