CMT107 Visual Computing

VIII.1 Edge Detection

Xianfang Sun

School of Computer Science & Informatics
Cardiff University

Overview

- Origin of Edges
- Characterising Edges
- Derivatives with Convolution
  - Finite Difference Filters
  - Image Gradient
- Canny Edge Detector

Acknowledgement

The majority of the slides in this section are from Svetlana Lazebnik at University of Illinois at Urbana-Champaign

Edge Detection

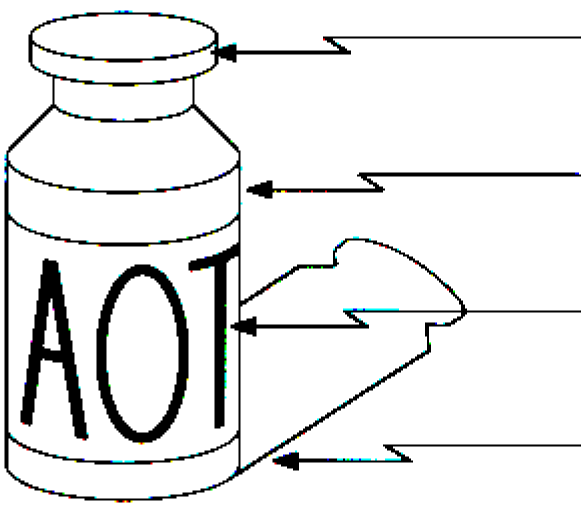
- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels

- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

Source: D. Lowe

Origin of Edges

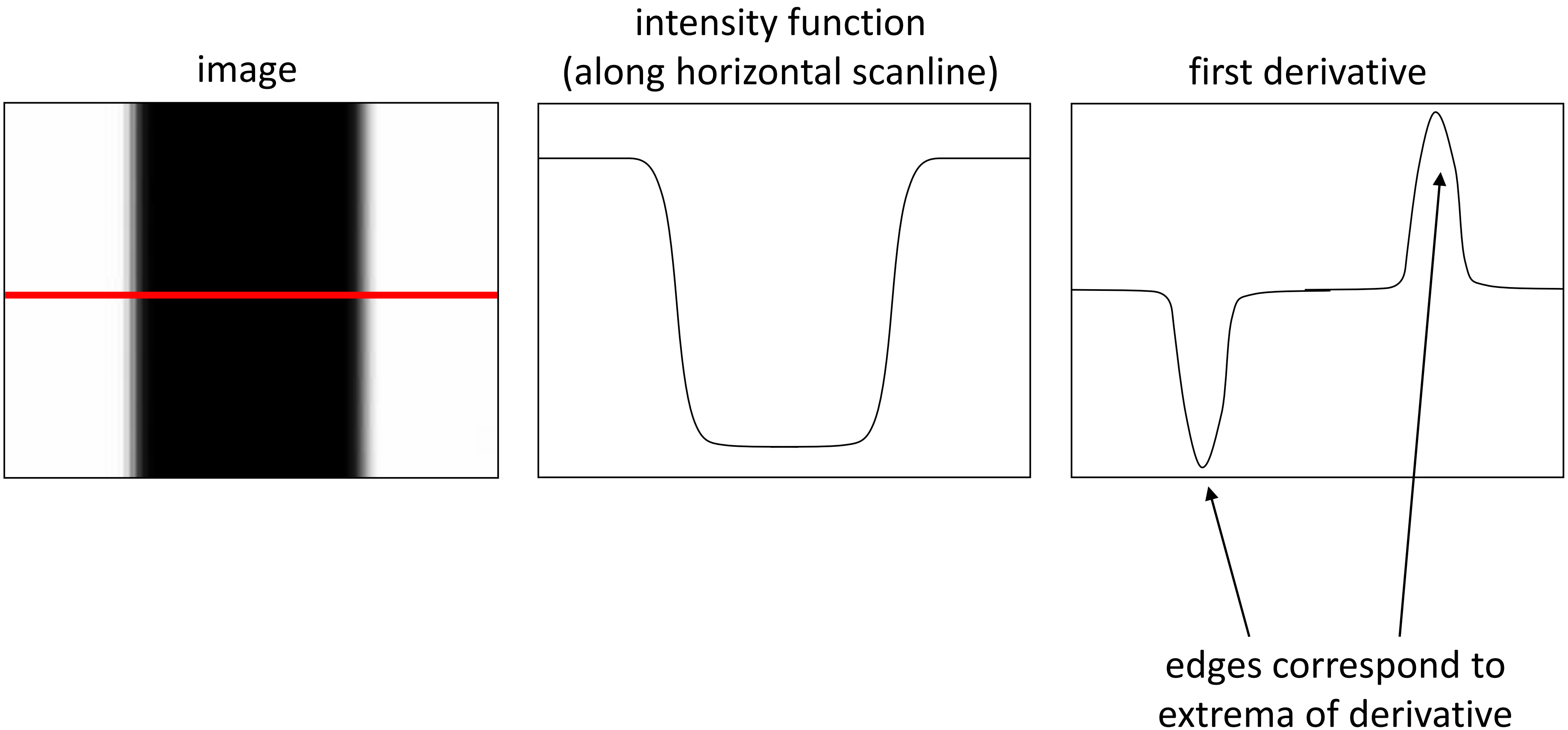
- Edges are caused by a variety of factors:



Source: Steve Seitz

Characterising Edges

- An edge is a place of rapid change in the image intensity function



Derivatives with Convolution

For 2D function f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

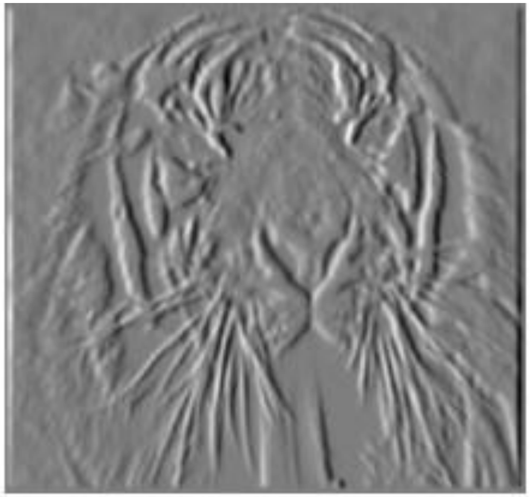
To implement above as convolution, what would be the associated filter?

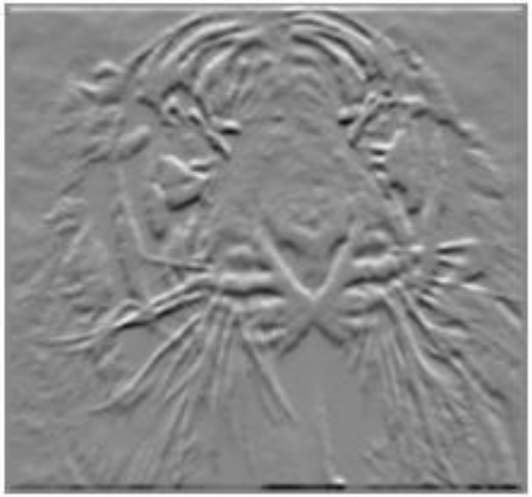
Source: K. Grauman

Partial Derivatives of an Image

$$\frac{\partial f(x,y)}{\partial x}$$

-1	1





$$\frac{\partial f(x,y)}{\partial y}$$

-1
1

or

1
-1

Which shows changes with respect to x?

Finite Difference Filters

- Other approximations of derivative filters exist:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

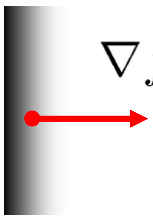
Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

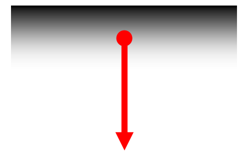
Source: K. Grauman

Image Gradient

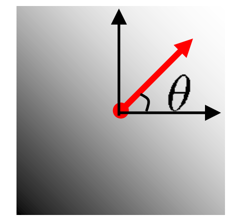
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient points in the direction of most rapid increase in intensity

- How does this direction relate to the direction of the edge?

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

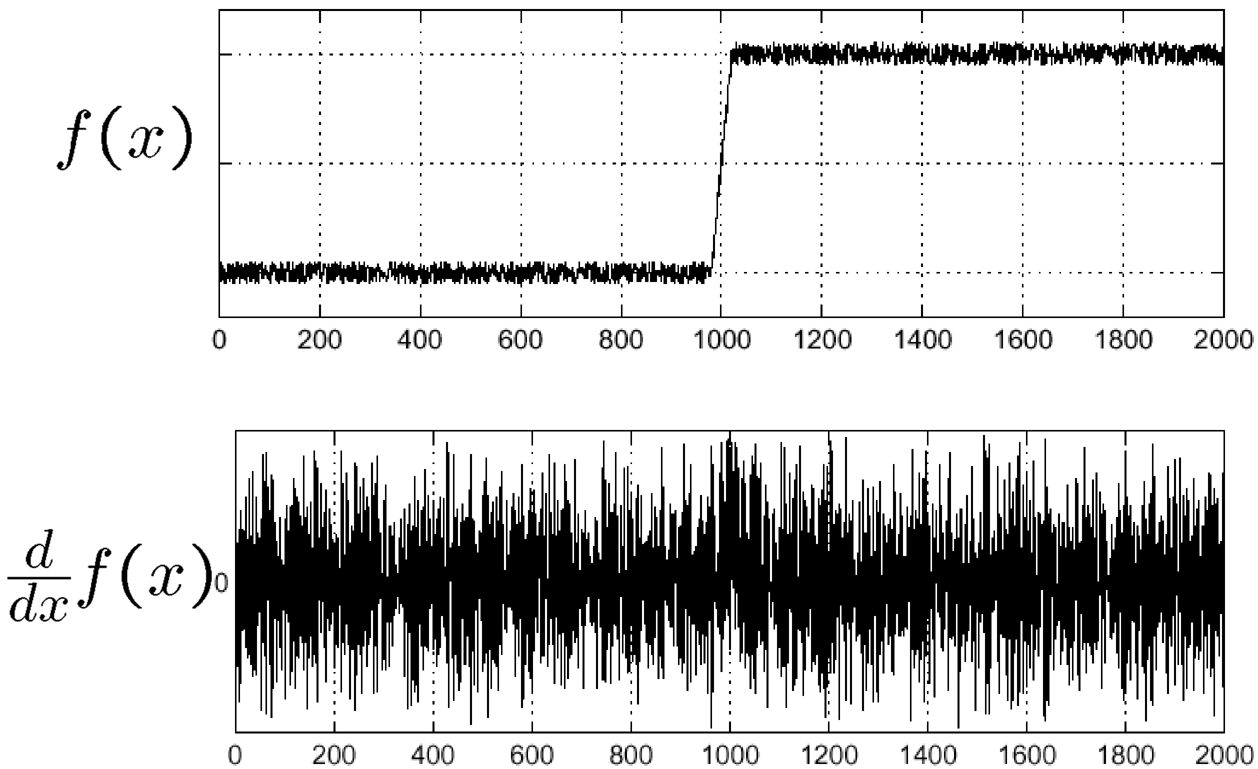
The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Effects of Noise

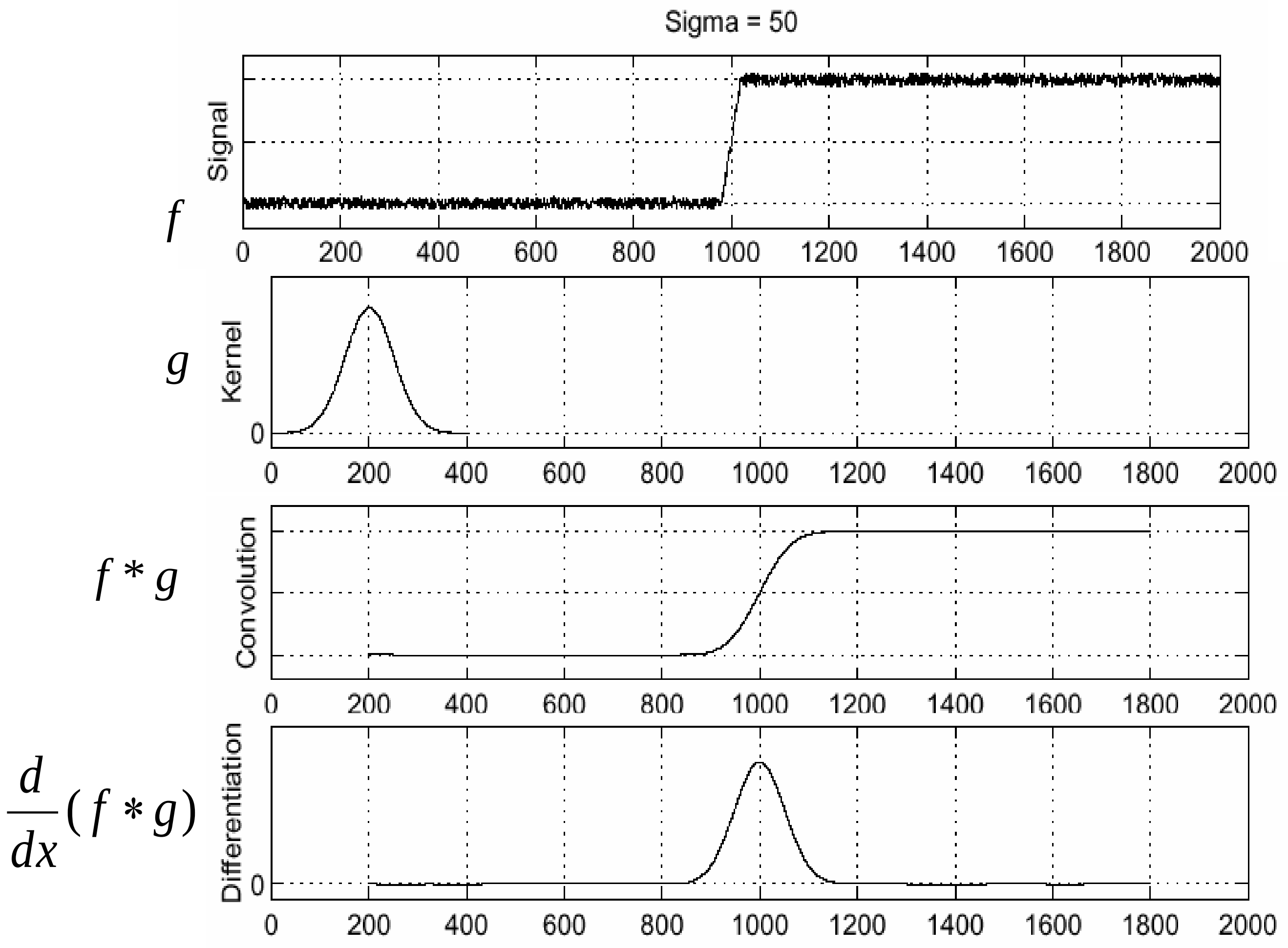
- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



Where is the edge?

Source: S. Seitz

Solution: Smooth First



- To find edges, look for peaks in $\frac{d}{dx}(f * g)$

Source: S. Seitz

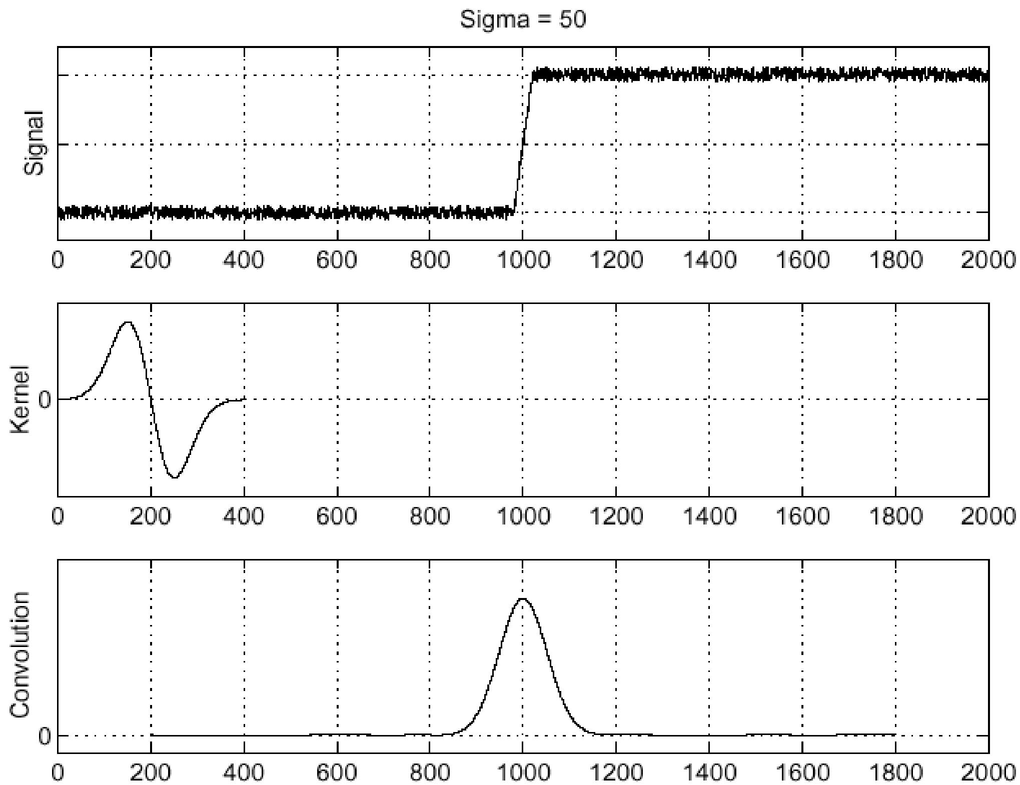
Derivative Theorem of Convolution

- Differentiation is convolution, and convolution is associative: $$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$
- This saves us one operation:

f

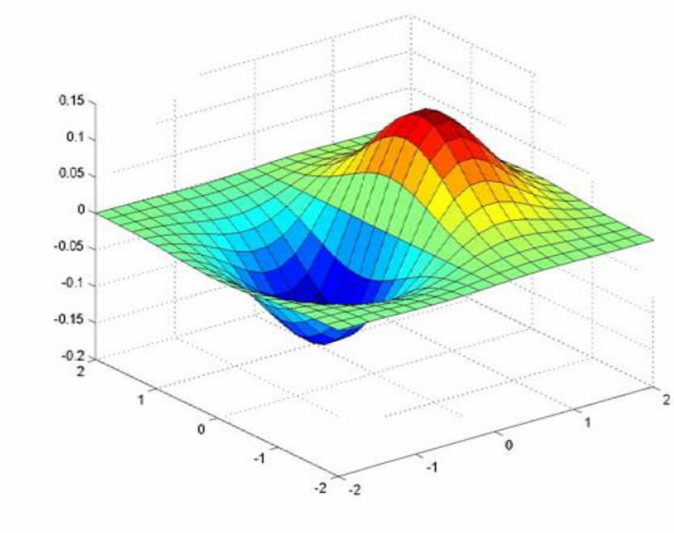
$\frac{d}{dx} g$

$f * \frac{d}{dx} g$

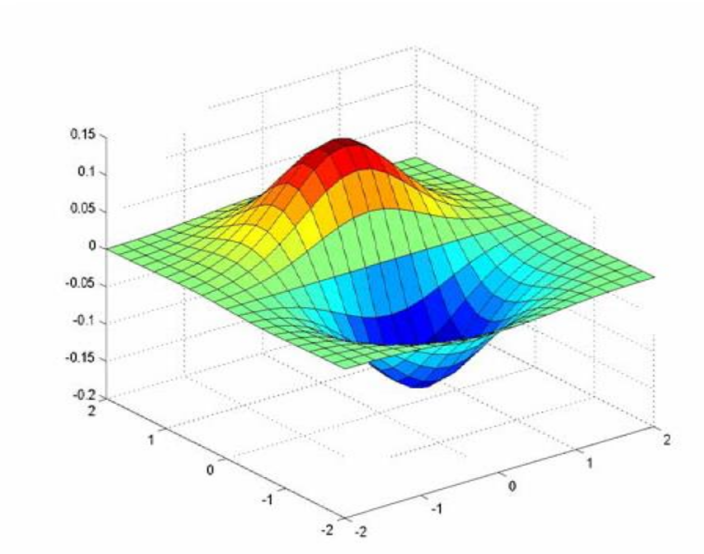


Source: S. Seitz

Derivative of Gaussian Filter



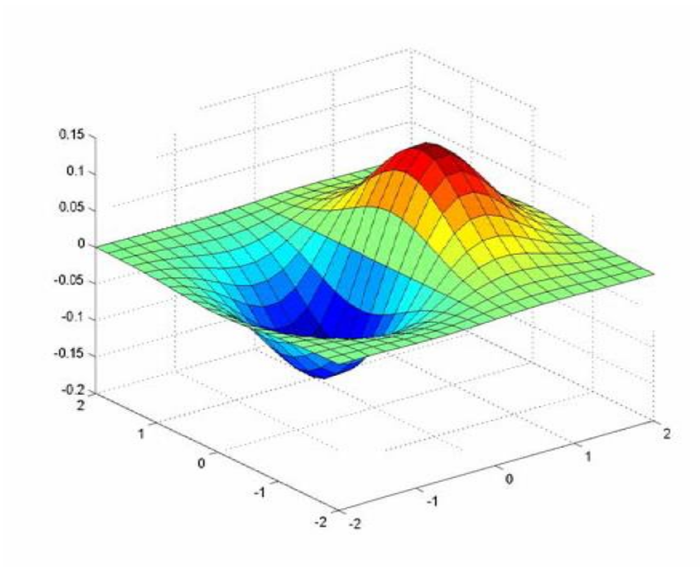
x-direction

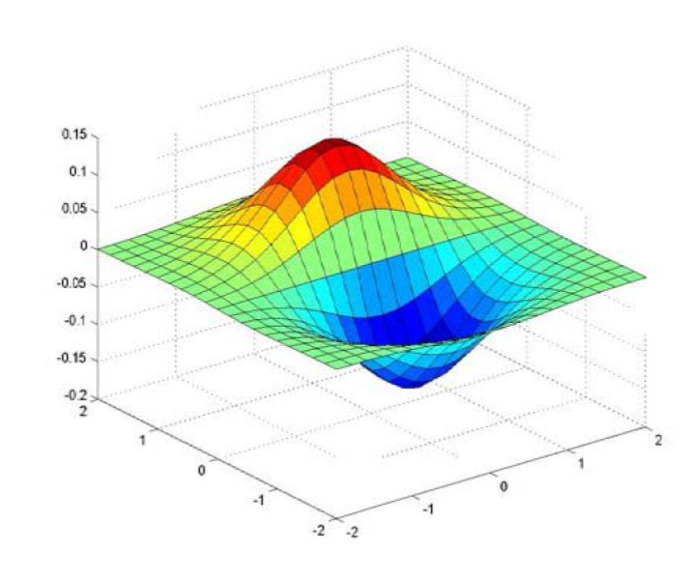


y-direction

- Are these filters separable?

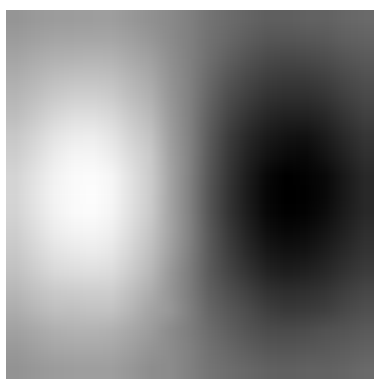
Derivative of Gaussian Filter

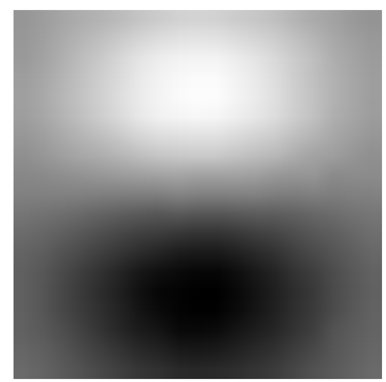




x-direction

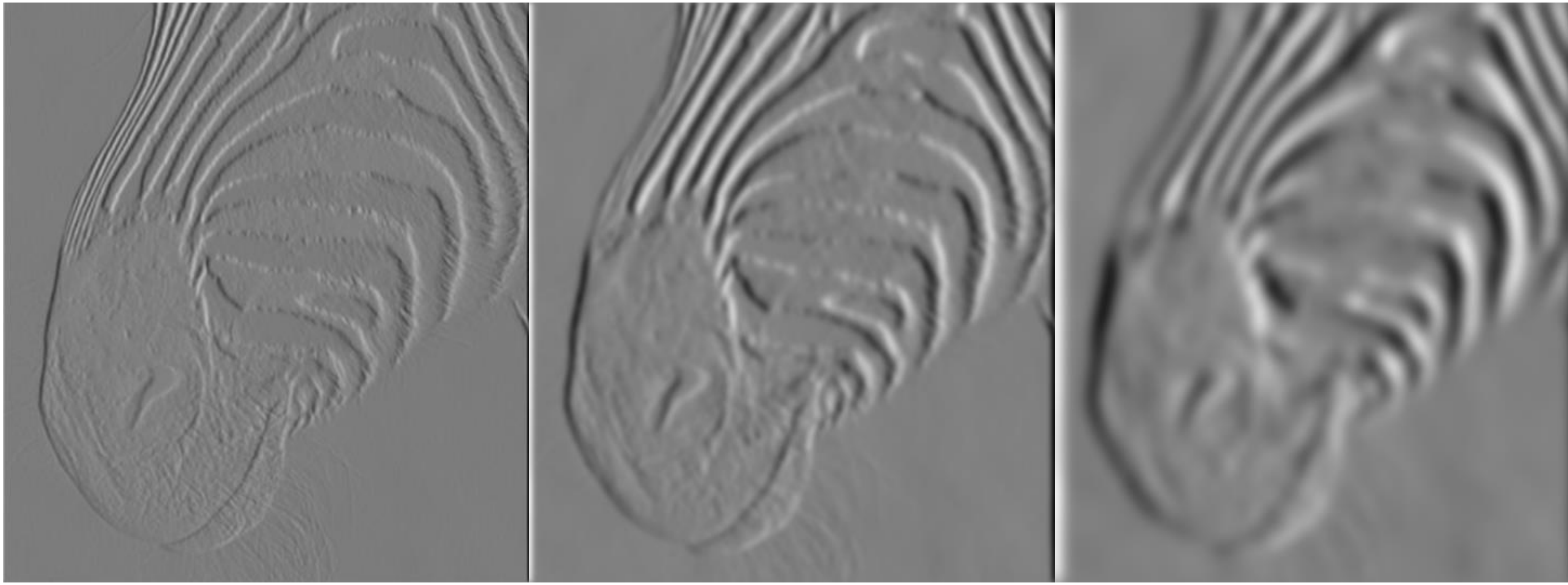
y-direction





- Which one finds horizontal/vertical edges?

Scale of Gaussian Derivative Filter



1 pixel　　3 pixels　　7 pixels

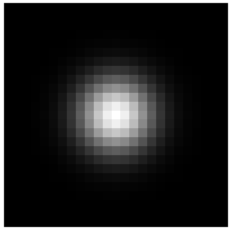
- Smoothed derivative removes noise, but blurs edge. Also finds edges at different “scales”

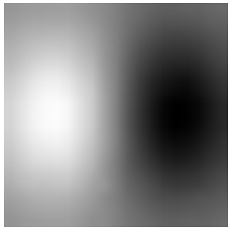
Source: D. Forsyth

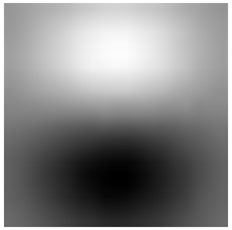
Review: Smoothing vs. Derivative Filters

➢ Smoothing filters

- Gaussian: remove “high-frequency” components; “low-pass” filter
- Can the values of a smoothing filter be negative?
- What should the values sum to?
  - — One: constant regions are not affected by the filter







➢ Derivative filters

- Derivatives of Gaussian
- Can the values of a derivative filter be negative?
- What should the values sum to?
  - — Zero: no response in constant regions
- High absolute value at points of high contrast

Canny Edge Detector

original image

Slide credit: Steve Seitz

Canny Edge Detector

norm of the gradient

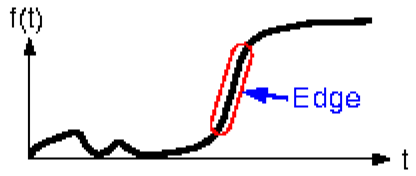
Slide credit: Steve Seitz

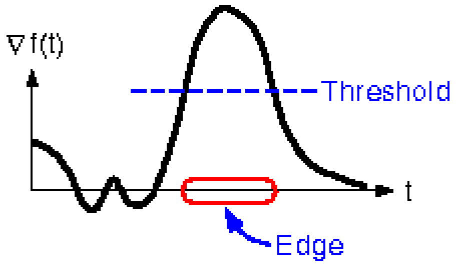
Canny Edge Detector

thresholding

Slide credit: Steve Seitz

Canny Edge Detector

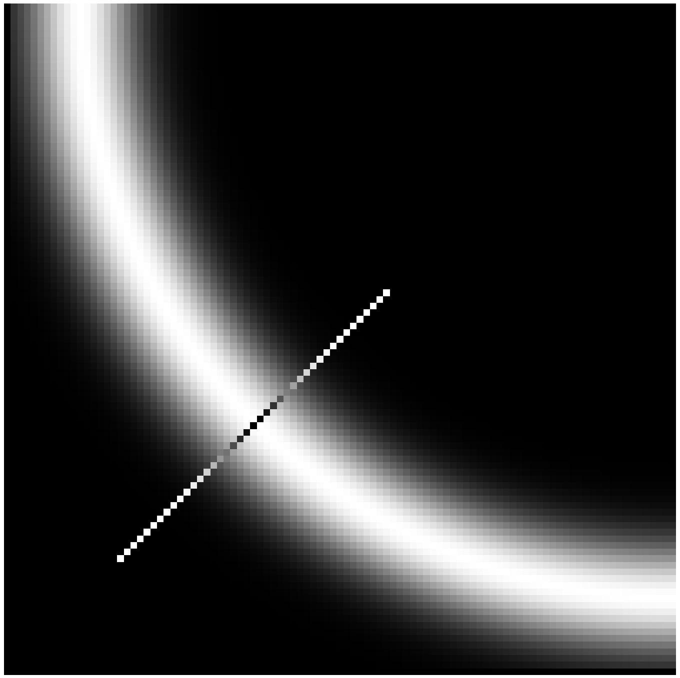


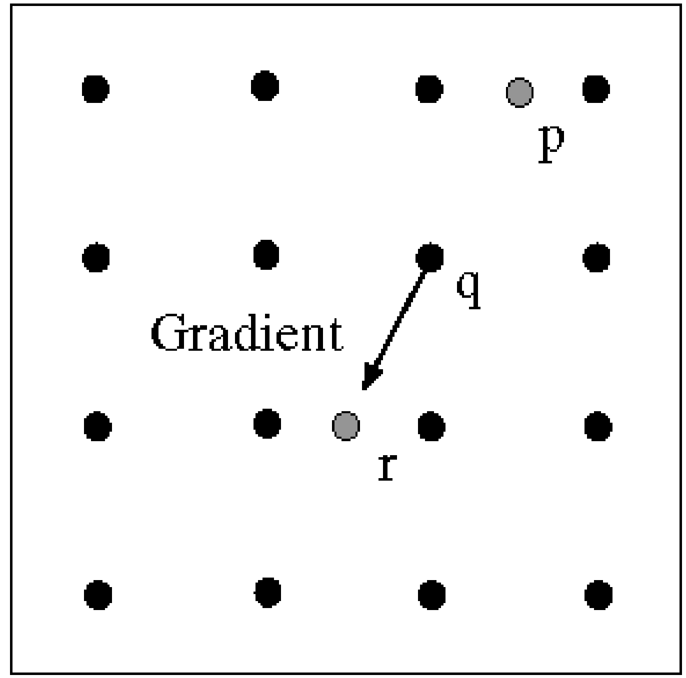


How to turn these thick regions of the gradient into curves?

thresholding

Non-maximum Suppression





Check if pixel is local maximum along gradient direction, select single max across width of the edge

- requires checking interpolated pixels p and r

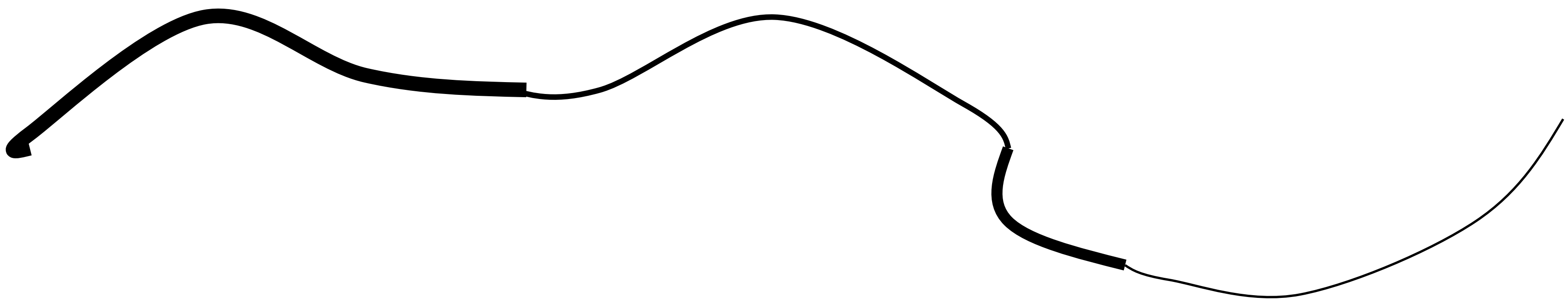
Canny Edge Detector

Problem: pixels along this edge didn't survive the thresholding

thinning
(non-maximum suppression)

Hysteresis Thresholding

- Use a high threshold to start edge curves, and a low threshold to continue them.



Source: Steve Seitz

Hysteresis Thresholding

original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Source: L. Fei-Fei

Summary of Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin wide "ridges" down to single pixel width
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

J. Canny, *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Summary

- What is edge detection?
- Describe different origin of edges.
- How to characterise edges?
- How to calculate image gradient using Prewitt, Sobel, or Roberts filters?
- Describe the steps of canny edge detector.

CMT107 Visual Computing

VIII.2 Image Morphology

Xianfang Sun

School of Computer Science & Informatics
Cardiff University

Overview

- Morphology
  - Dilation
  - Erosion
  - Duality of Dilation and Erosion
  - Opening
  - Closing
- Hit-Or-Miss transformation

Acknowledgement
The majority of the slides in this section are from Punam K Saha at University of Iowa

Morphology

- Morphological operators often take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement).
- The structuring element is shifted over the image and at each pixel of the image its elements are compared with the set of the underlying pixels.
- If the two sets of elements match the condition defined by the set operator (e.g. if set of pixels in the structuring element is a subset of the underlying image pixels), the pixel underneath the origin of the structuring element is set to a pre-defined value (0 or 1 for binary images).
- A morphological operator is therefore defined by its structuring element and the applied set operator.

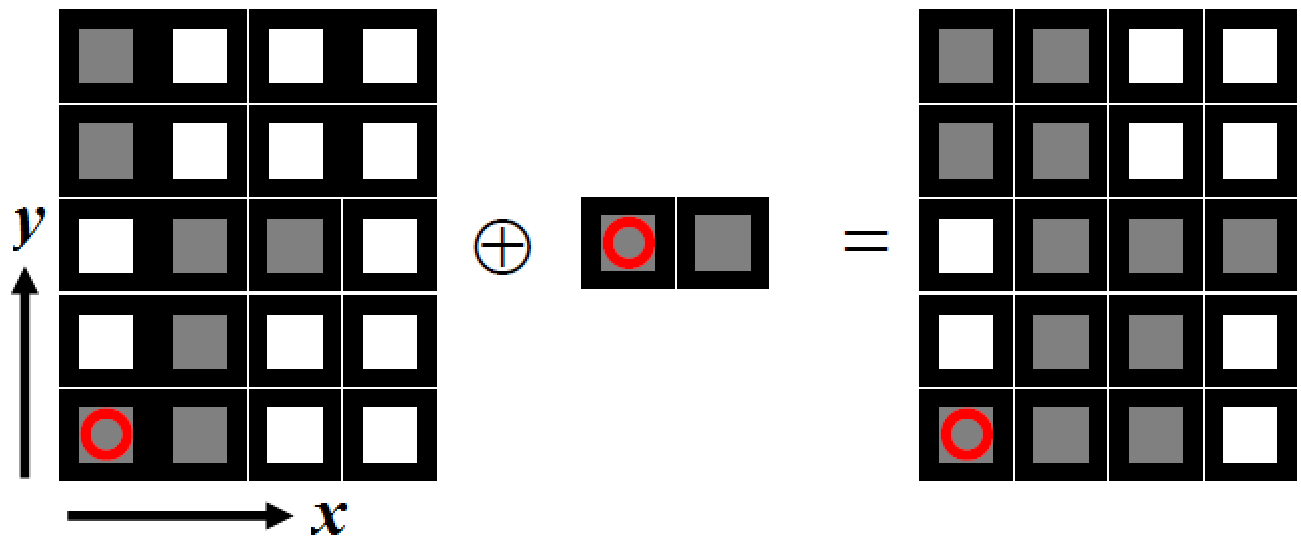
Morphology Applications

- Image pre-processing
  - noise filtering
  - shape simplification
- Enhancing object structures
  - Skeletonisation
  - Thinning
  - convex hull
  - object marking
- Segmentation of the object from background
- Quantitative descriptors of objects
  - Area
  - Perimeter
  - …, etc.)

Example: Morphological Operation

- Let '$\oplus$' denote a morphological operator

$$X \oplus B = \{p \in Z^2 \,|\, p = x + b,\ x \in X,\ b \in B\ \}$$



B ={(0,0),(1,0)}

X ={(0,0), (1,0), (1,1), (1,2), (2,2), (0,3), (0,4)}

$X \oplus B$ ={(0,0),(1,0) (1,1),(2,1) (2,2),(3,2) (0,4),(1,4) }

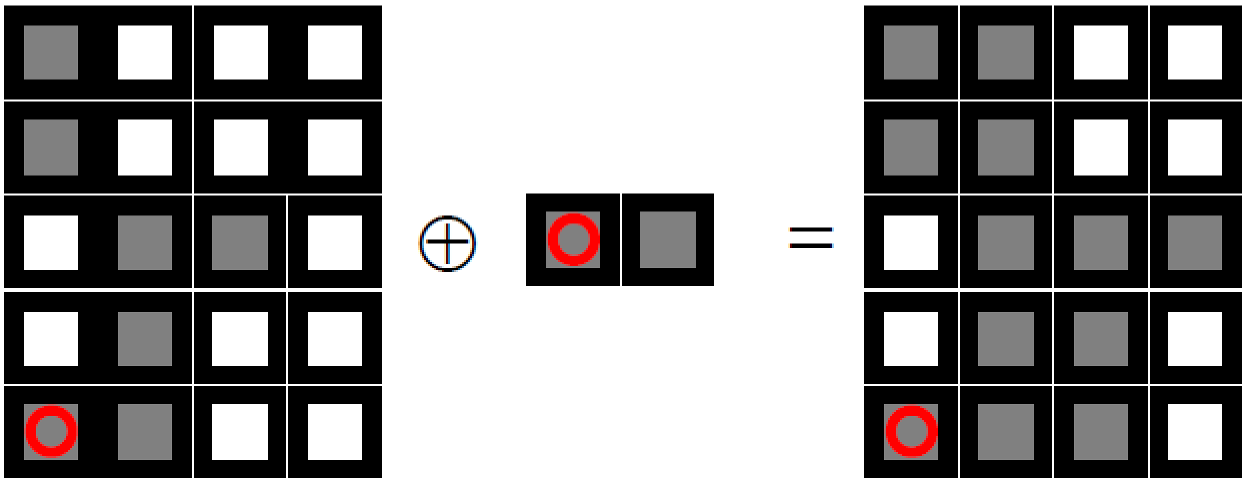
(1,0),(2,0) (1,2),(2,2) (0,3),(1,3)

$X \oplus B$ ={(0,0), (1,0), (2,0), (1,1), (2,1), (1,2), (2,2),
(3,2), (0,3), (1,3), (0,4), (1,4)}

Dilation

- Morphological dilation '$\oplus$' combines two sets using vector addition of set elements

$$X \oplus B = \{p \in Z^2 | p = x + b, x \in X, b \in B\}$$

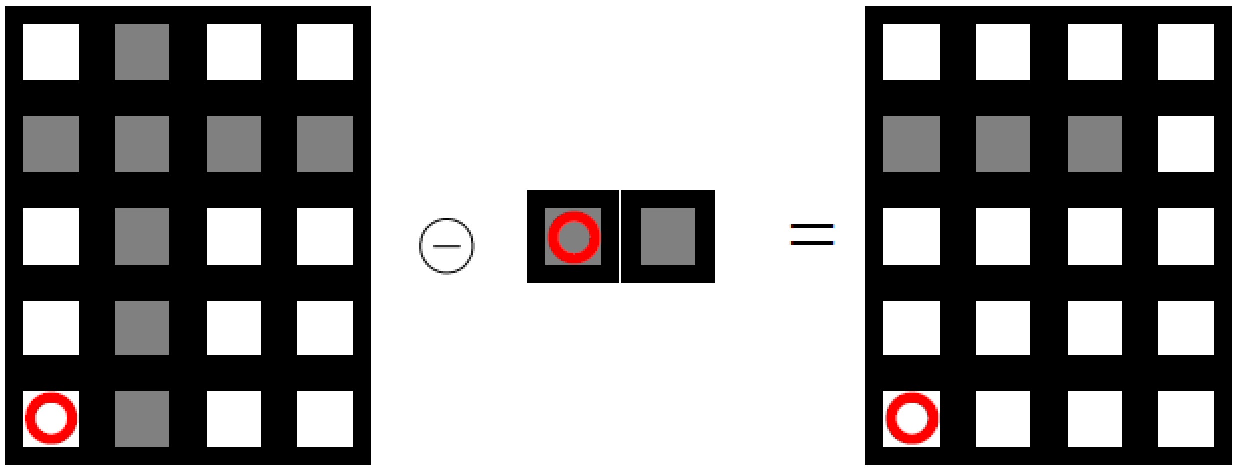


- Commutative: $X \oplus B = B \oplus X$
- Associative: $X \oplus B \oplus D = X \oplus (B \oplus D)$
- Invariant of translation: $X_h \oplus B = (X \oplus B)_h$
  - $X_h = \{p \in Z^2 | p = x + h, x \in X\}$
- If $X \subseteq Y$ then $X \oplus B \subseteq Y \oplus B$

Erosion

- Morphological erosion '$\ominus$' combines two sets using vector subtraction of set elements and is a dual operator of dilation

$$X \ominus B = \{p \in Z^2 | \forall b \in B, p + b \in X\}$$



- Not Commutative: $X \ominus B \neq B \ominus X$
- Not associative: $X \ominus (B \ominus D) \neq (X \ominus B) \ominus D$
- Invariant of translation: $X_h \ominus B = (X \ominus B)_h$ and $X \ominus B_h = (X \ominus B)_{-h}$
- If $X \subseteq Y$ then $X \ominus B \subseteq Y \ominus B$

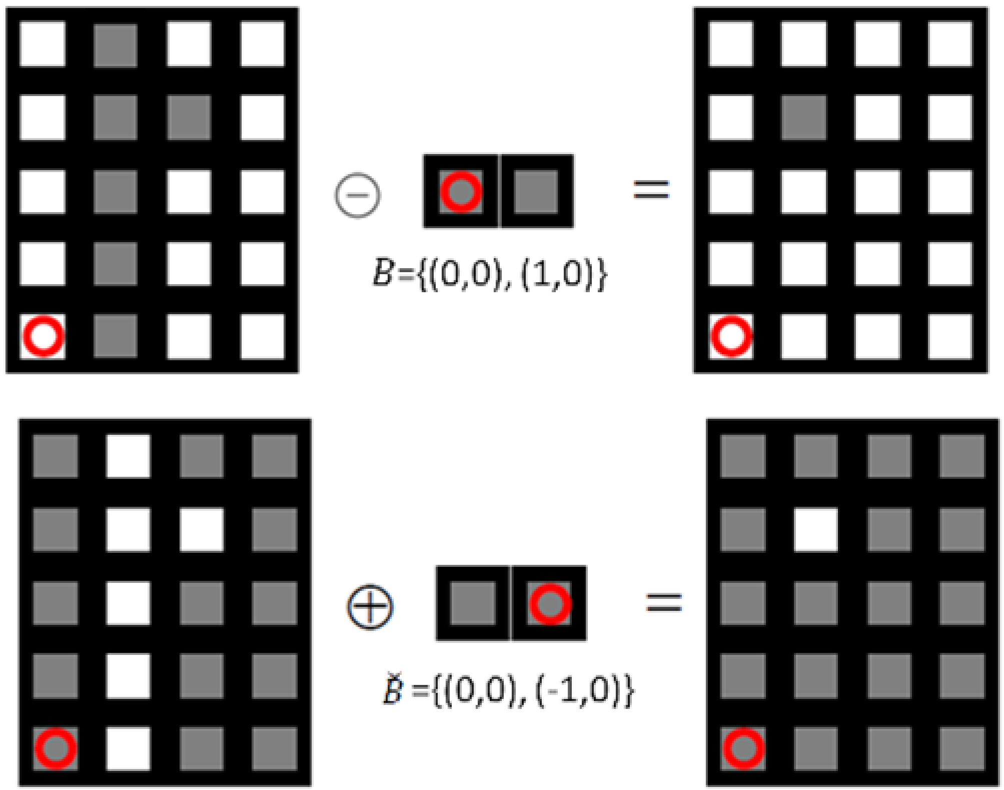
Duality: Dilation and Erosion

- Transpose $\breve{A}$ *of a structuring element A is defined as follows*

$$\breve{A} = \{-a \mid a \in A\}$$

- Duality between morphological dilation and erosion operators

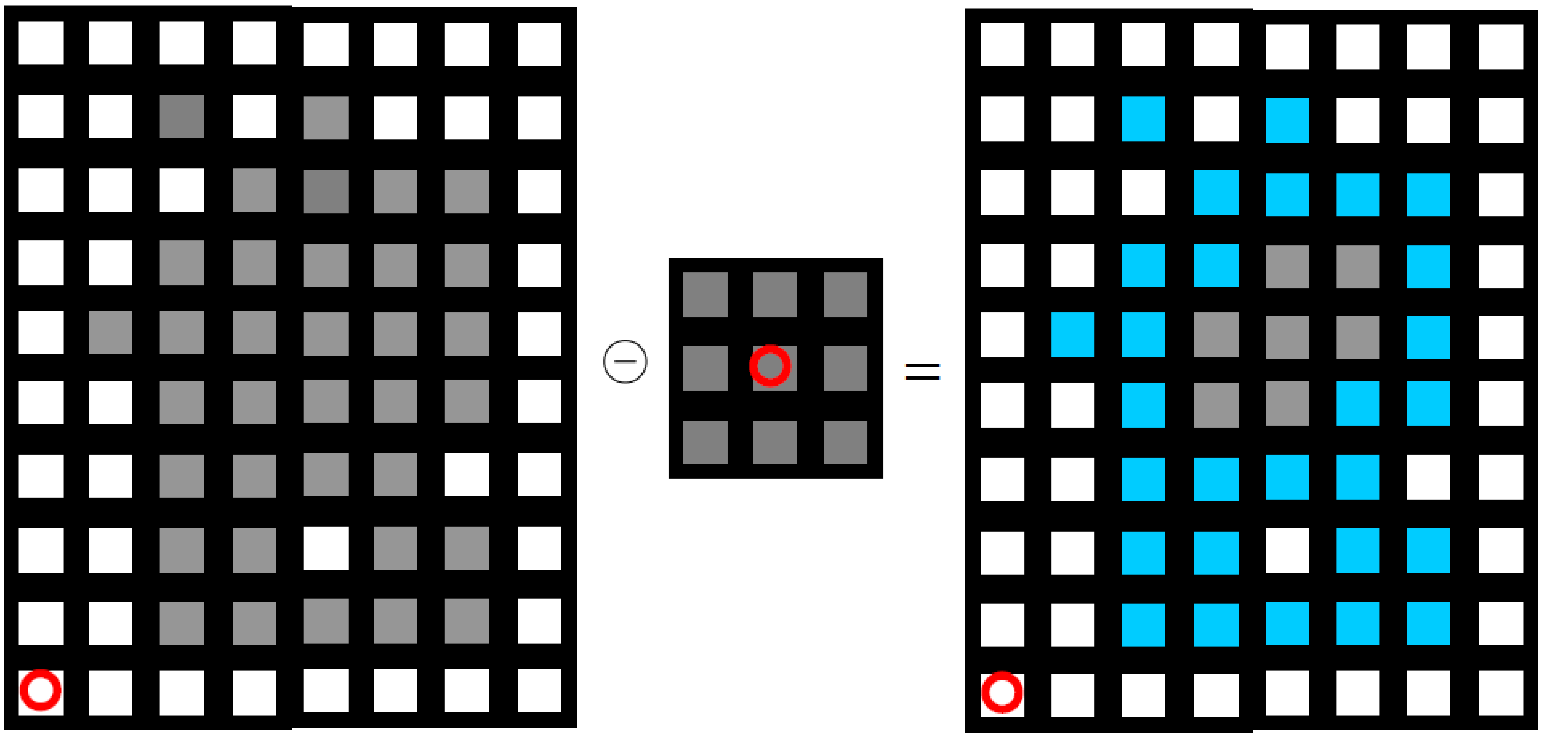
$$(X \ominus B)^c = X^c \oplus \breve{B} \qquad \text{(c – complement)}$$



Opening

- Erosion and dilation are not inverse transforms. An erosion followed by a dilation leads to an interesting morphological operation called opening

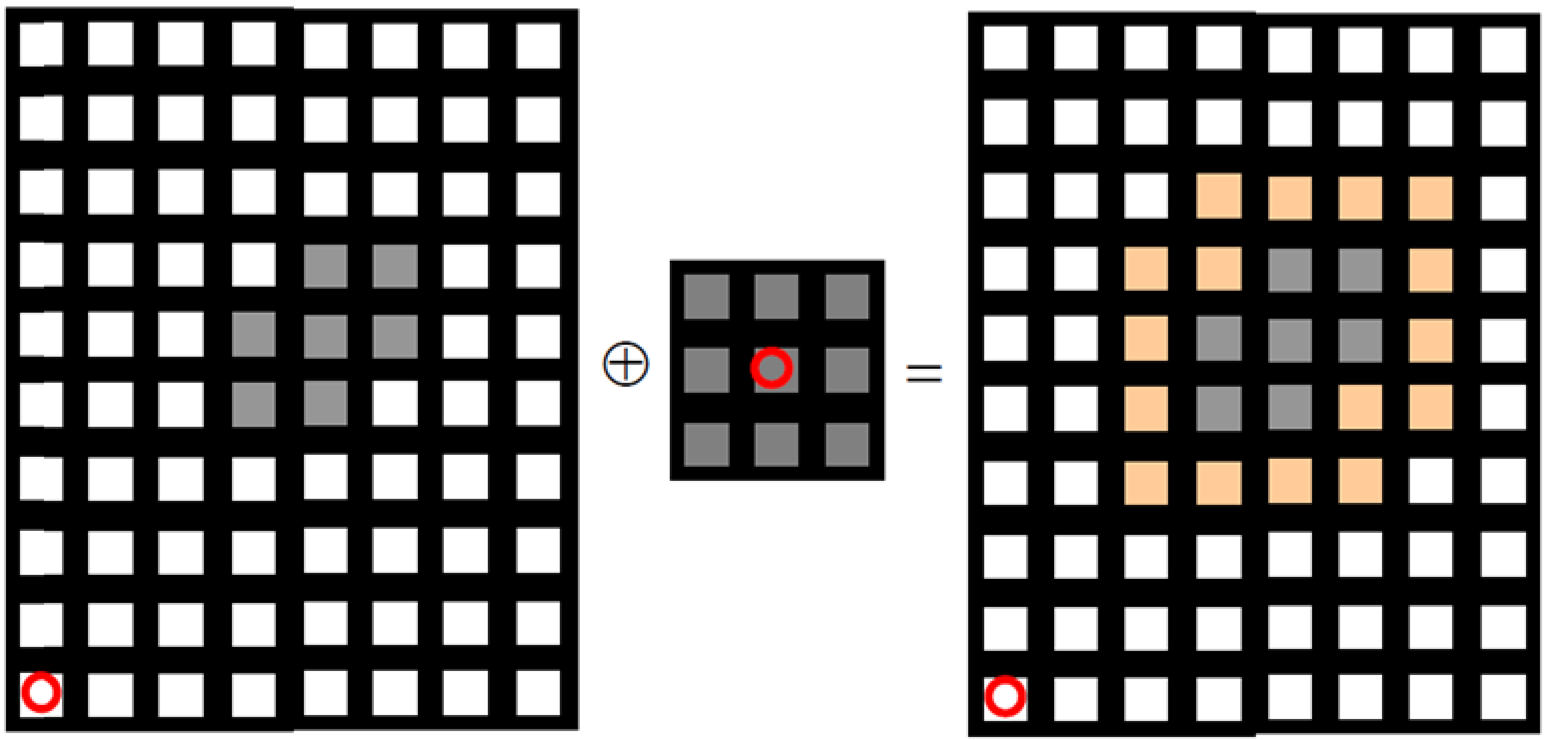
$$X \circ B = (X \ominus B) \oplus B$$



Opening

- Erosion and dilation are not inverse transforms. An erosion followed by a dilation leads to an interesting morphological operation called opening

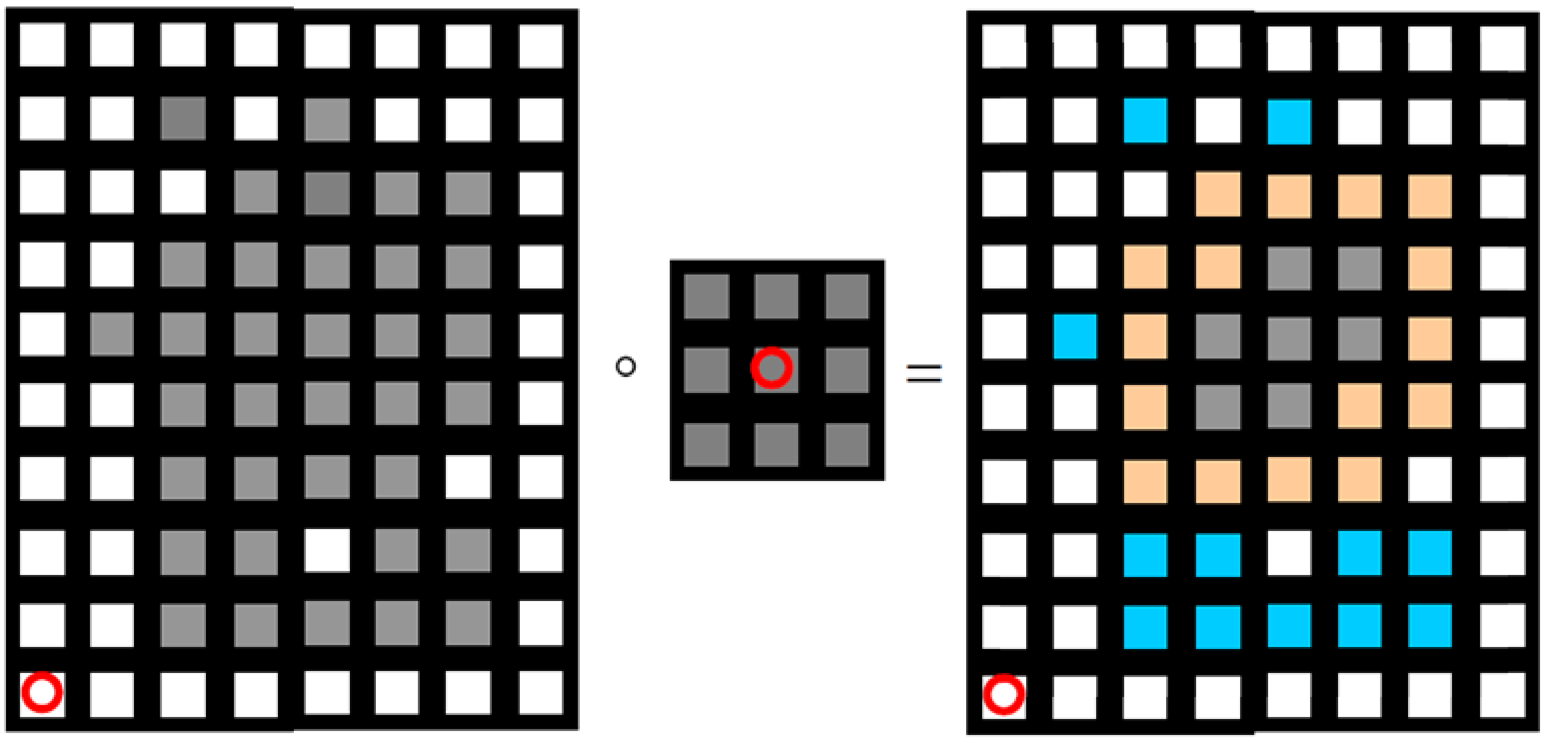
$$X \circ B = (X \ominus B) \oplus B$$



Opening

- Erosion and dilation are not inverse transforms. An erosion followed by a dilation leads to an interesting morphological operation called opening

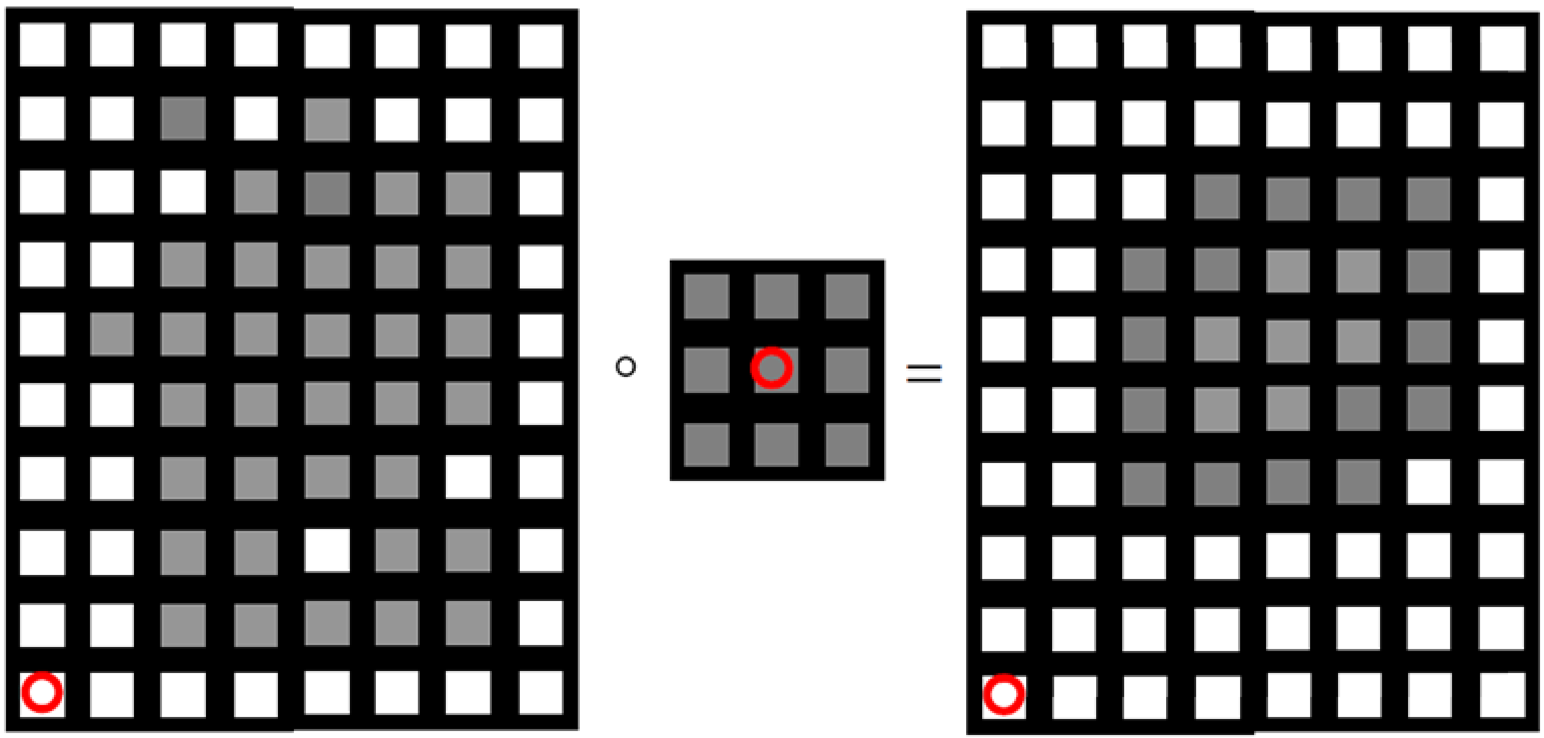
$$X \circ B = (X \ominus B) \oplus B$$



Opening

- Erosion and dilation are not inverse transforms. An erosion followed by a dilation leads to an interesting morphological operation called opening

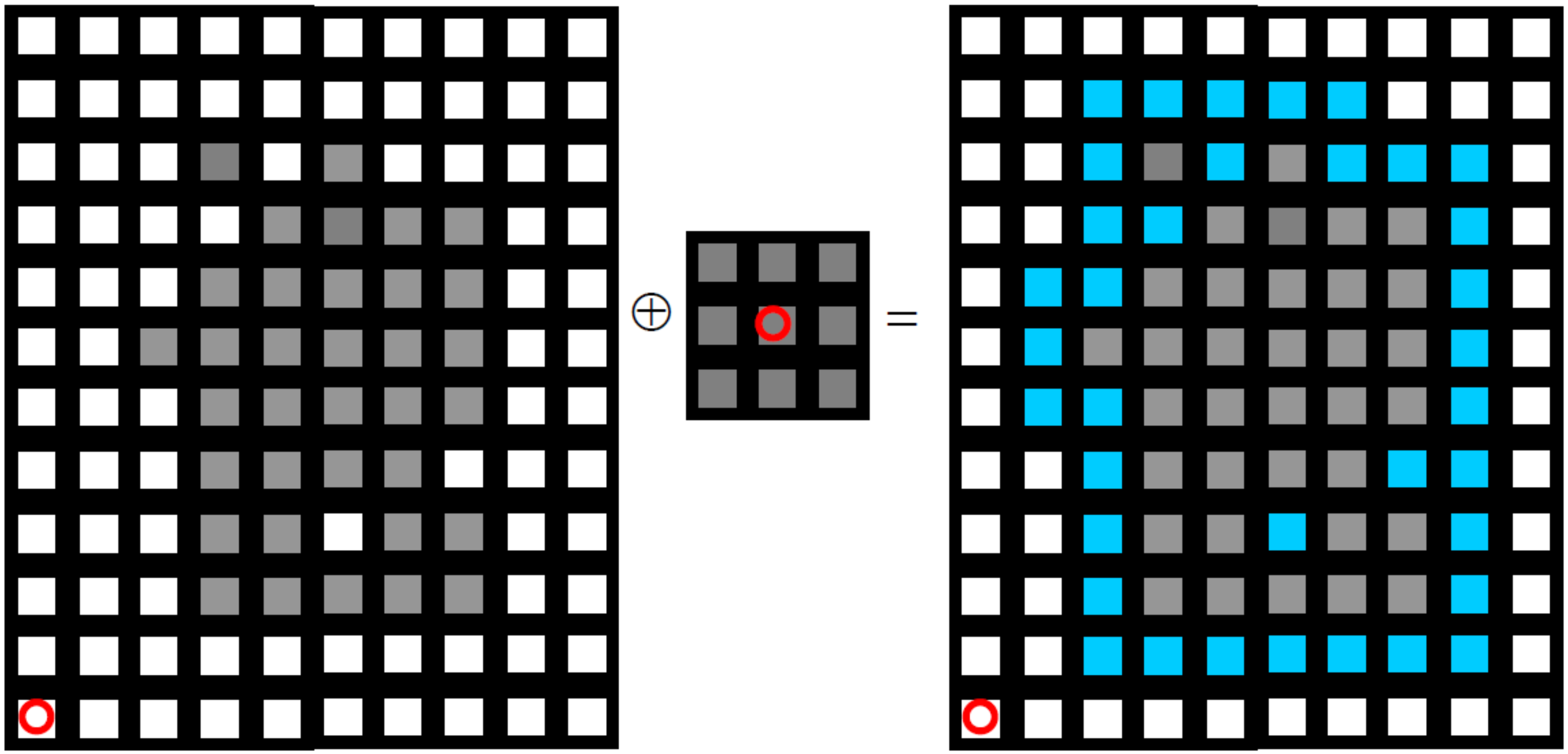
$$X \circ B = (X \ominus B) \oplus B$$



Closing

- A dilation followed by an erosion leads to the interesting morphological operation called closing

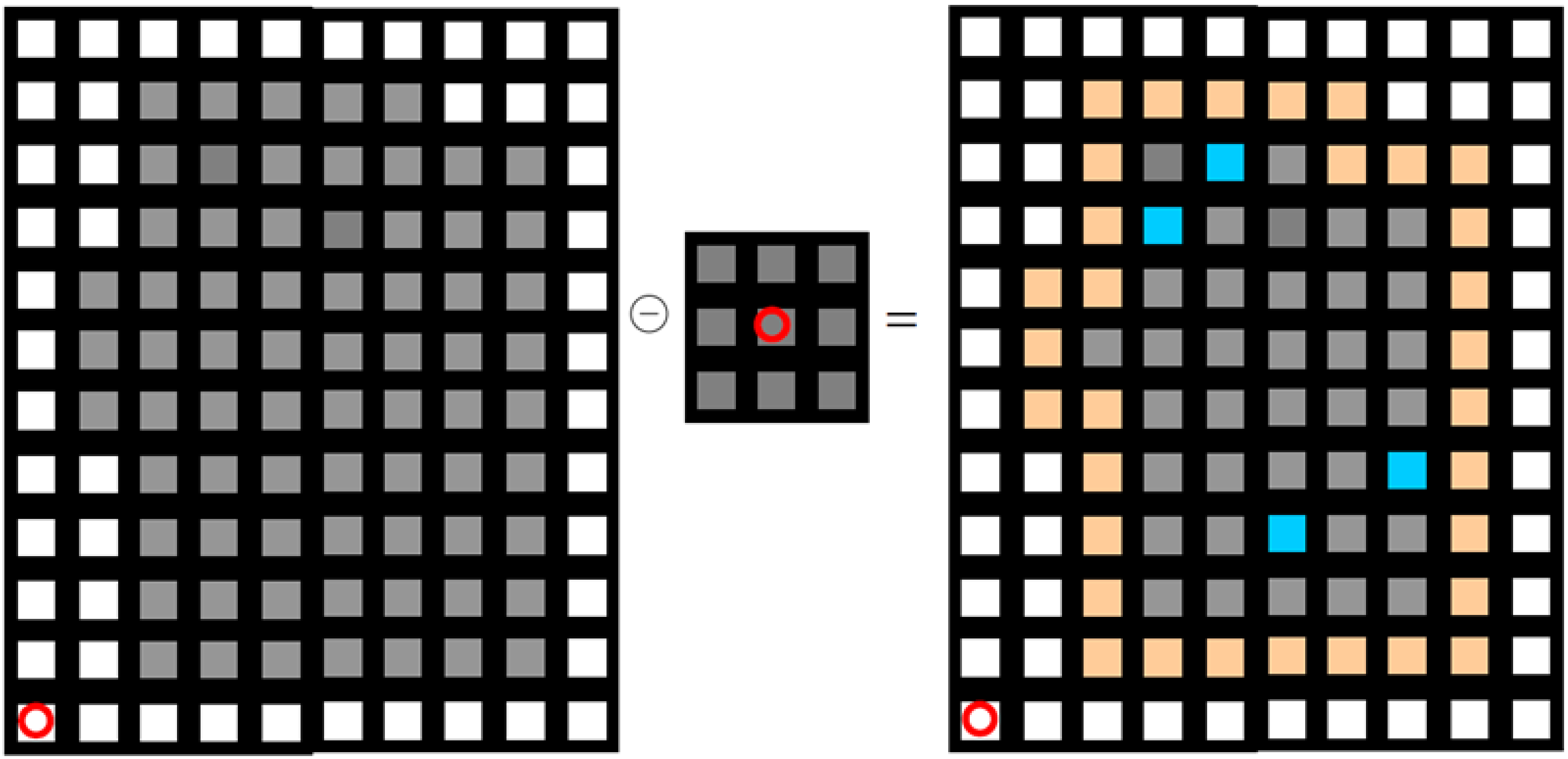
$$X \bullet B = (X \oplus B) \ominus B$$



Closing

- A dilation followed by an erosion leads to the interesting morphological operation called closing

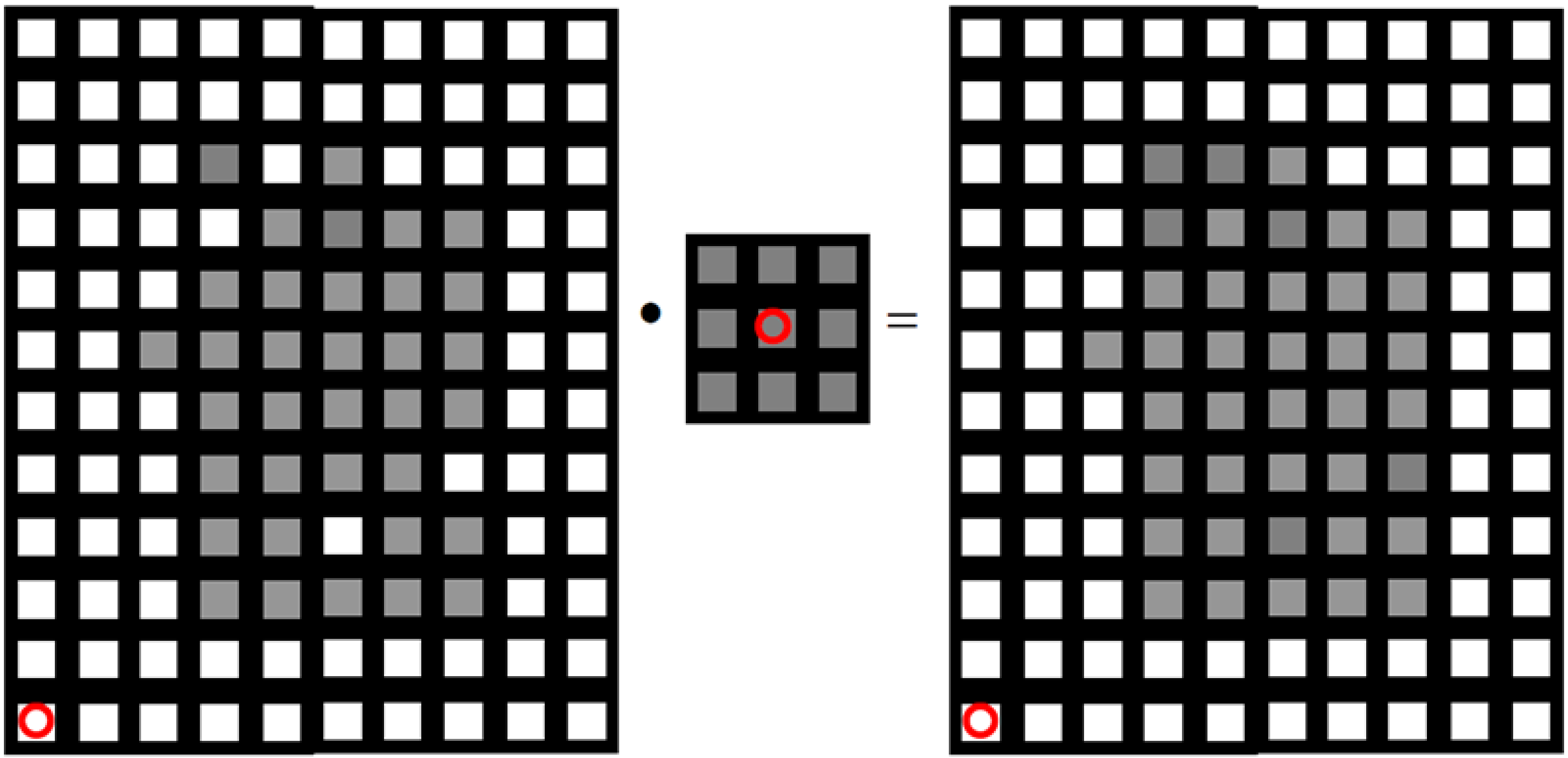
$$X \bullet B = (X \oplus B) \ominus B$$



Closing

- A dilation followed by an erosion leads to the interesting morphological operation called closing

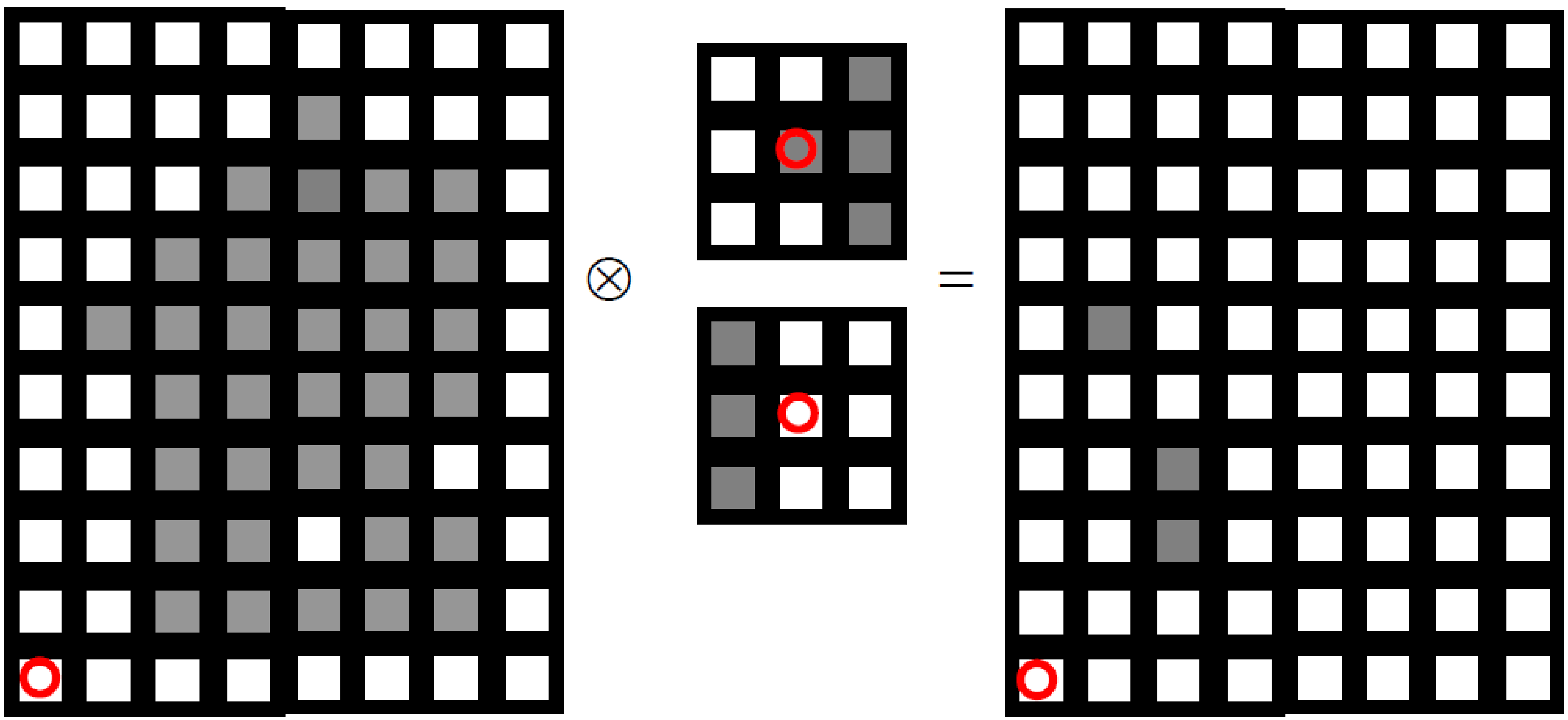
$$X \bullet B = (X \oplus B) \ominus B$$



Hit-Or-Miss transformation

- Hit-or-miss is a morphological operators for finding local patterns of pixels. Unlike dilation and erosion, this operation is defined using a composite structuring element $B = (B_1, B_2)$. The hit-or-miss operator is defined as follows

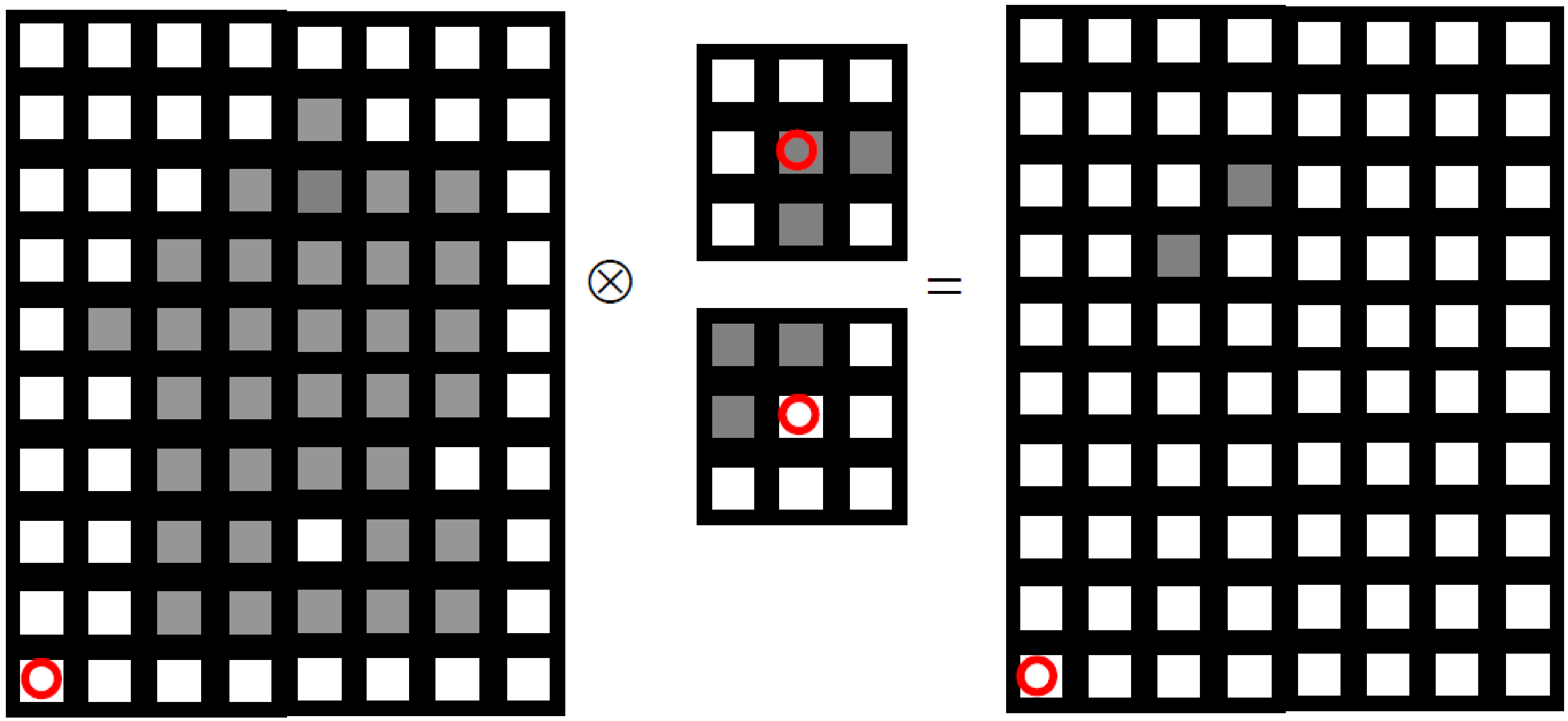
$$X \otimes B = \{x \mid B_1 \subset X \text{ and } B_2 \subset X^C\}$$



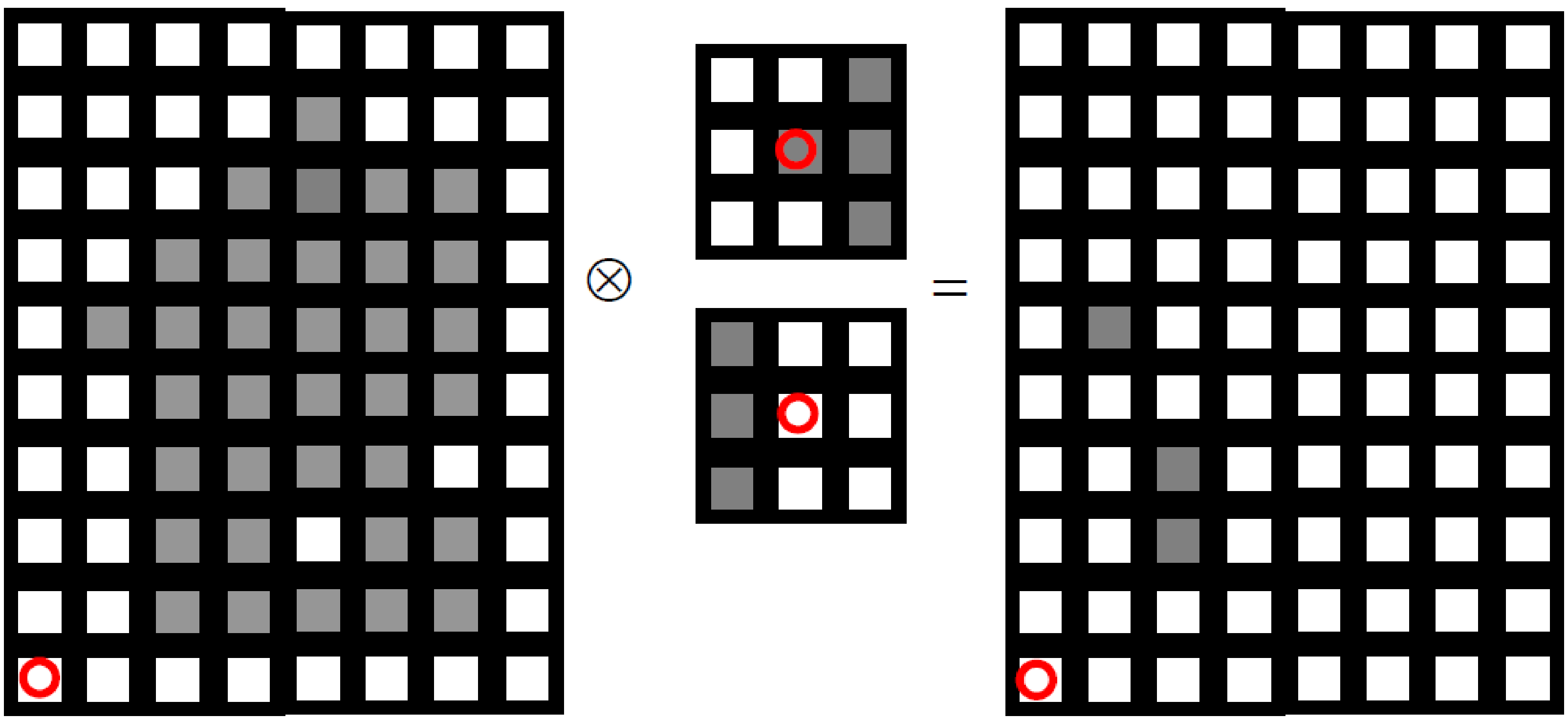
Hit-Or-Miss transformation: Another Example

- Relation with erosion:

  $X \otimes B = X \ominus B1 \cap X^c \ominus B2$



Hit-Or-Miss transformation: More Example



Summary

- What is Morphology? What are the applications of morphology?
- What is dilation, erosion, opening, and closing operators?
- What is hit-or-miss transformation?