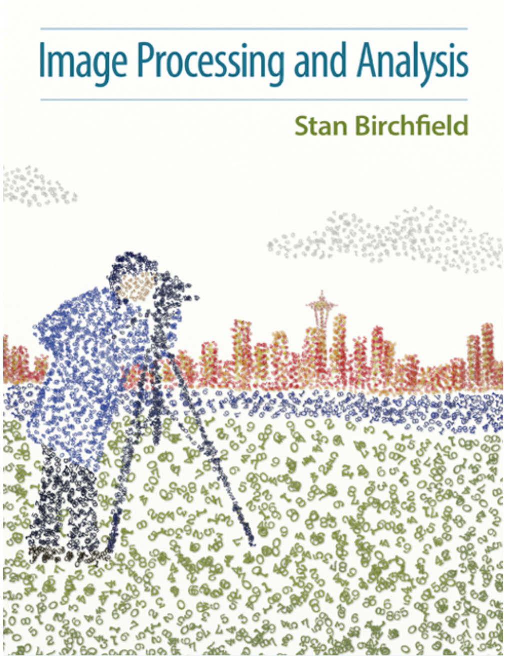
University of Stavanger

Prof. Kjersti Engan

ELE510 Image processing and computer vision

Binary Image Processing

8/25/20



Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

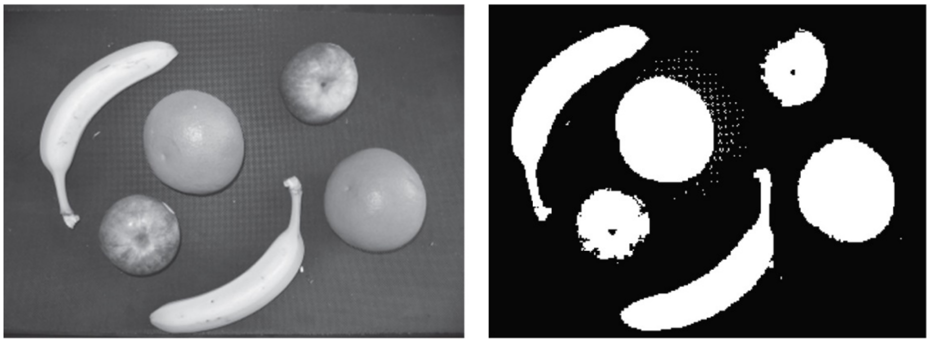
Mathematical morphology

- **Mathematical morphology:** a branch of mathematics developed to process images by considering the shape of the pixel regions.
- A **binary image** is an array of values such that $I(x, y)$ has 1 or 0 for each pixel location (x, y).

$$I = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$I = \{\boxed{(0,0)}, \boxed{(2,0)}, (1,1), (2,1)\}$$

(x, y) point



Stan Birchfield

Binary images as a set

- fundamental set operators:

Z is a point in the plane, i.e. (x,y) coordinates
A and B are sets of points in the plane

$$\mathcal{A} \cup \mathcal{B} \equiv \{\mathbf{z} : \mathbf{z} \in \mathcal{A} \text{ or } \mathbf{z} \in \mathcal{B}\} \qquad \text{(union)}$$

$$\mathcal{A} \cap \mathcal{B} \equiv \{\mathbf{z} : \mathbf{z} \in \mathcal{A} \text{ and } \mathbf{z} \in \mathcal{B}\} \qquad \text{(intersection)}$$

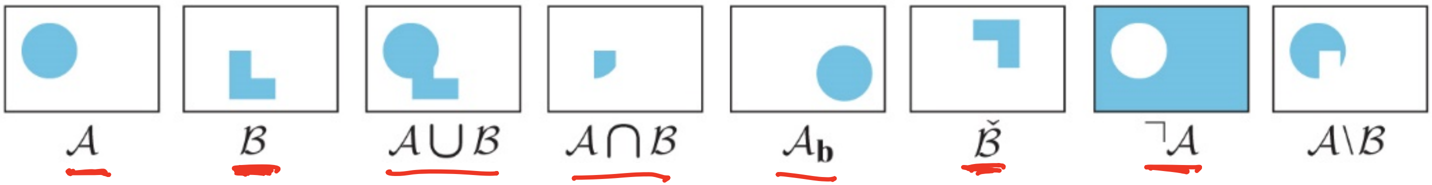
$$b = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$

$$\mathcal{A}_{\mathbf{b}} \equiv \{\mathbf{z} : \mathbf{z} = \mathbf{a} + \mathbf{b}, \mathbf{a} \in \mathcal{A}\} \qquad \text{(translation)}$$

$$\check{\mathcal{B}} \equiv \{\mathbf{z} : \mathbf{z} = -\mathbf{b}, \mathbf{b} \in \mathcal{B}\} \qquad \text{(reflection)}$$

$$\neg\mathcal{A} \equiv \{\mathbf{z} : \mathbf{z} \notin \mathcal{A}\} \qquad \text{(complement)}$$

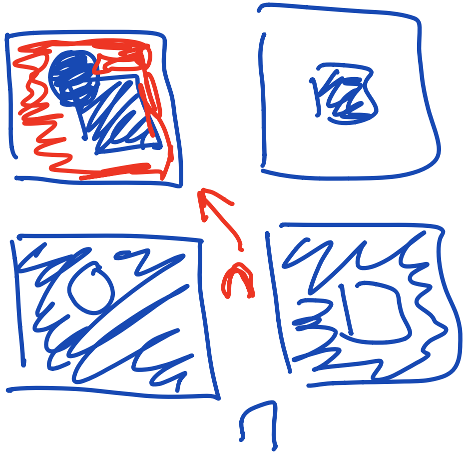
$$\mathcal{A} \setminus \mathcal{B} \equiv \{\mathbf{z} : \mathbf{z} \in \mathcal{A}, \mathbf{z} \notin \mathcal{B}\} = \mathbf{A} \cap \neg\mathcal{B} \qquad \text{(difference)}$$



De Morgan´s laws

$$\neg(A \cup B) = \neg A \cap \neg B$$

$$\neg(A \cap B) = \neg A \cup \neg B$$



$\underline{a} \in A$

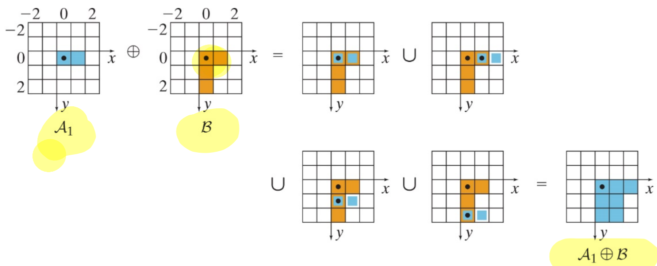
$\underline{b} \in B$

$$\underline{a} + \underline{b} = \begin{bmatrix} a_x + b_x \\ a_y + b_y \end{bmatrix}$$

Minkowski addition and subtraction

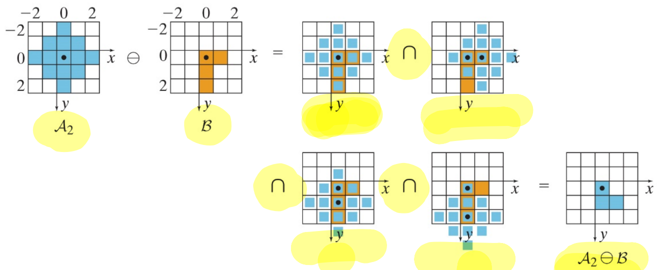
- The **Minkowski addition** of two sets *A* and *B* is defined as the set of points resulting from all possible vector additions of elements of the two sets:

$$\mathcal{A} \oplus \mathcal{B} \equiv \{\mathbf{z} : \mathbf{z} = \mathbf{a} + \mathbf{b}, \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$$
$$= \bigcup_{\mathbf{b} \in \mathcal{B}} \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in \mathcal{A}\} = \bigcup_{\mathbf{b} \in \mathcal{B}} \mathcal{A}_{\mathbf{b}}$$



- **Minkowski subtraction** of two sets:

$$\mathcal{A} \ominus \mathcal{B} \equiv \{\mathbf{z} : \mathbf{z} - \mathbf{b} \in \mathcal{A}, \forall \mathbf{b} \in \mathcal{B}\}$$
$$= \bigcap_{\mathbf{b} \in \mathcal{B}} \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in \mathcal{A}\} = \bigcap_{\mathbf{b} \in \mathcal{B}} \mathcal{A}_{\mathbf{b}}$$



Dilation and Erosion

- Based on Minkowski addition and subtraction, we define 2 fundamnetal morphological operators:
- Dilation: identical to Minkowski addition
- Erosion: the Minkowski subtraction after reflecting the second operand -> keep a set of locations where the original set fits inside the other set.

$$\mathcal{A} \oplus \mathcal{B} \equiv A \oplus \mathcal{B} = \{\mathbf{z} : \mathbf{z} = \mathbf{a} + \mathbf{b}, \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\} \quad \text{(dilation)}$$

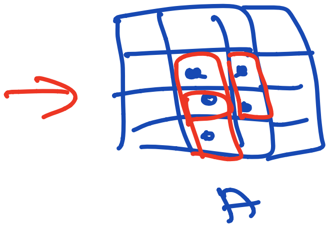
$$\mathcal{A} \check{\ominus} \mathcal{B} \equiv A \ominus \check{\mathcal{B}} = \{\mathbf{z} : \mathbf{z} + \mathbf{b} \in \mathcal{A}, \forall \mathbf{b} \in \mathcal{B}\} \quad \text{(erosion)}$$

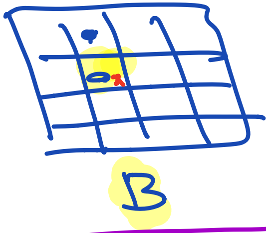
center out : $A \oplus B = \{ \underline{z} : \check{B}_z \cap A \neq 0 \}$

slide $\check{B}$ over A, find where $\neq 0$

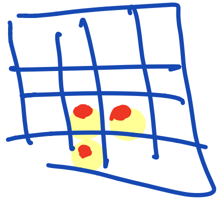
$A \check{\ominus} B = \{ \underline{z} : B_z \subseteq A \}$

Center out

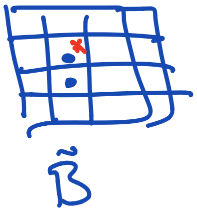


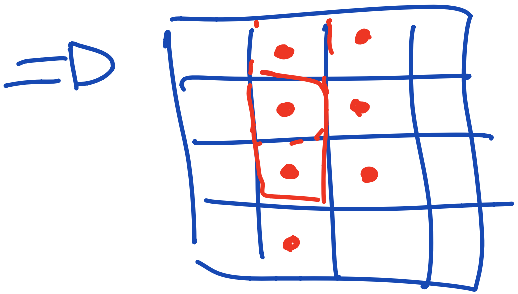


$$A \check{\ominus} B = \{z : B_z \subseteq A\}$$



$$A \oplus B = \{z : \check{B}_z \cap A \neq \emptyset\}$$



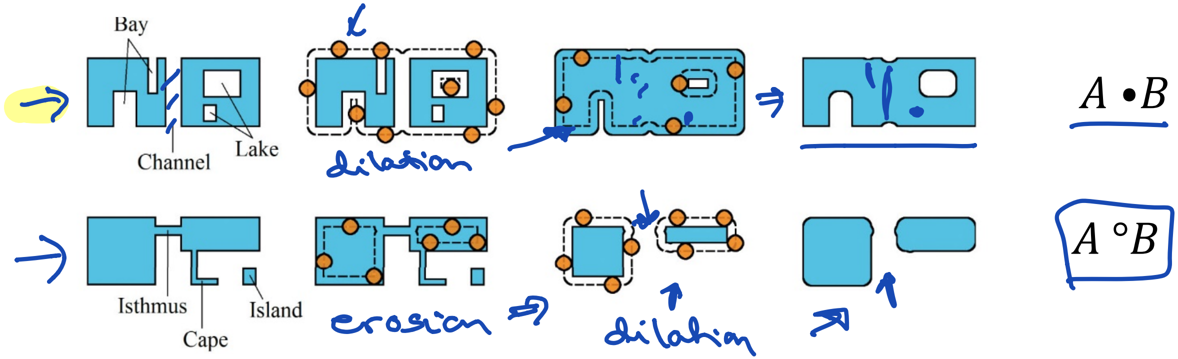


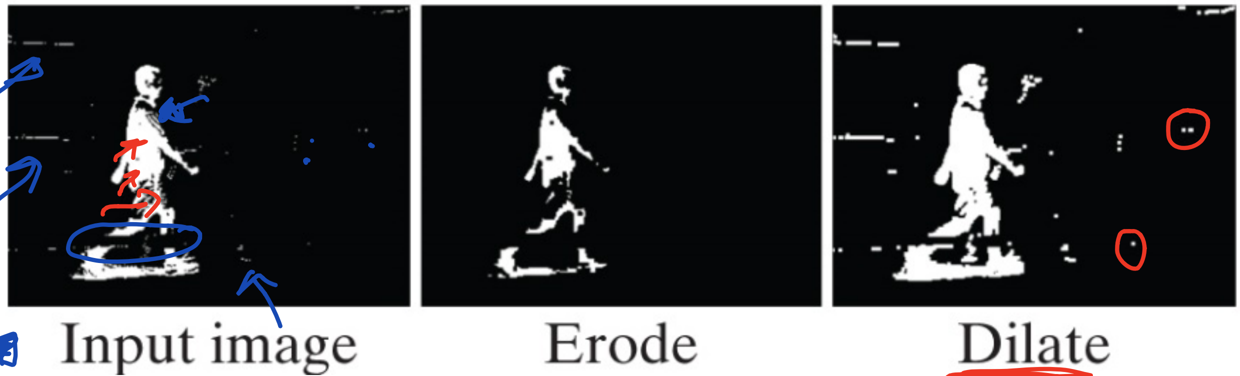
Opening and Closing

- Usually the set B is a structuring element (SE), much smaller than the image A. Can formulate dilation and erosion as translating B across the image performing test (center out).
- Closing is defined as dilation folowed by erosion
- Opening is defined as erosion followed by dilation

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

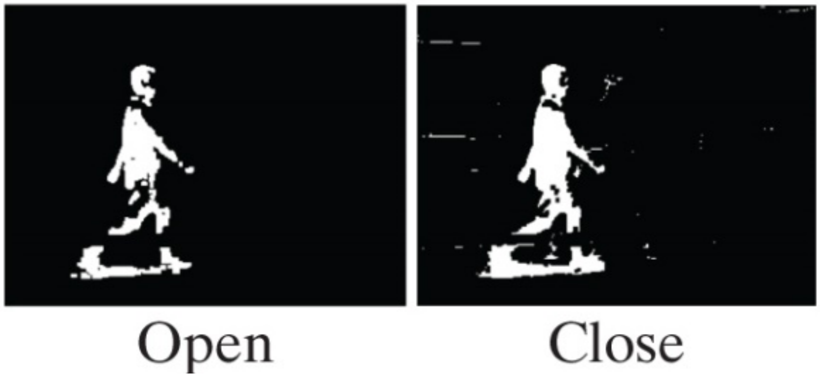
$A \circ B = (A \ominus B) \oplus B$





Erosion removes salt noise,
but shrinks foreground.

Dilate fills pepper noise but
expands foreground.

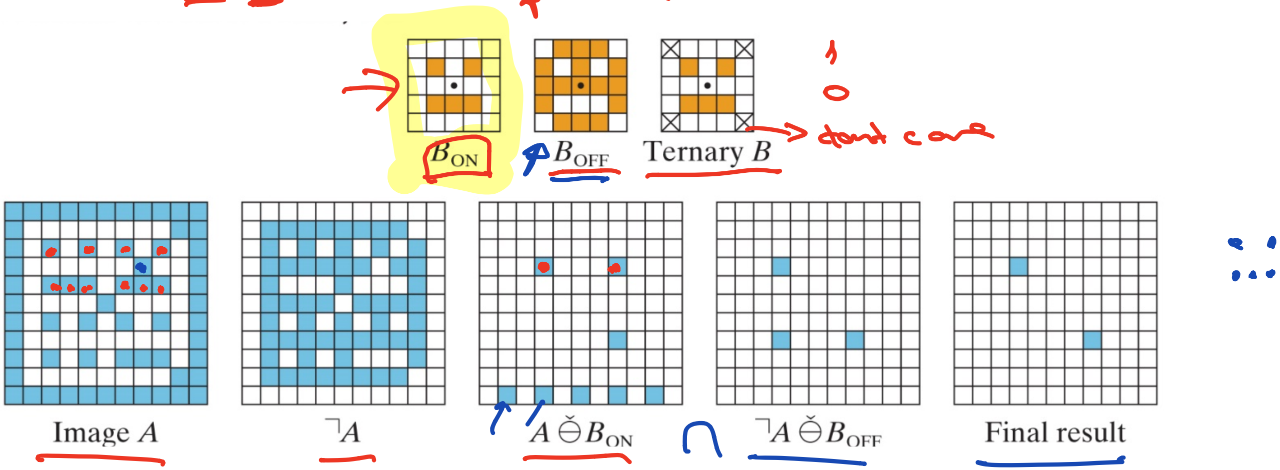


Stan Birchfield

Hit and miss operator

- To detect the shape in the image, the **hit-miss operator** uses erosion to find all the places in the image where B_{ON} matches the foreground and B_{OFF} matches the background:

$$A \circledast (B_{ON}, B_{OFF}) = (A \check{\ominus} B_{ON}) \cap (\neg A \check{\ominus} B_{OFF}) \qquad \text{(hit-miss operator)}$$



Morphological image processing

- Removing noise (salt and pepper noise)
- Thinning
- Thickening
- Labeling regions
- Region properties
- Boundary tracing – boundary Representations - signatures
- Hole filling
- Computing distances
- Skeletonization

Distance transform

Distance between two points: $(x_1, y_1), (x_2, y_2)$

Euclidean distance: $D_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Cityblock distance: $D_{cb} = |x_1 - x_2| + |y_1 - y_2|$

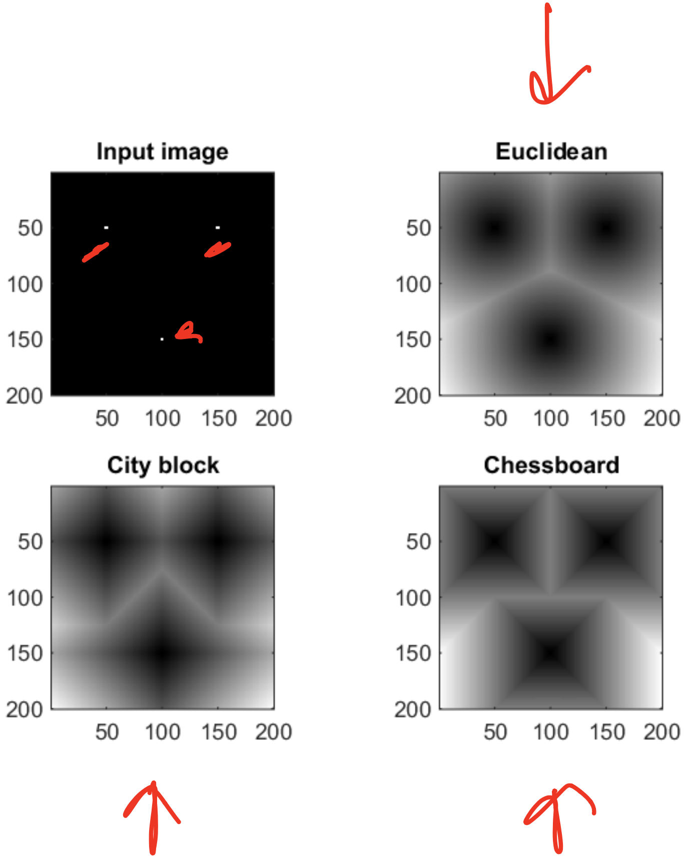
Chessboard distance: $D_{chess} = \max(|x_1 - x_2|, |y_1 - y_2|)$

The distance transform can be found as the distance from each pixel to any given feature.

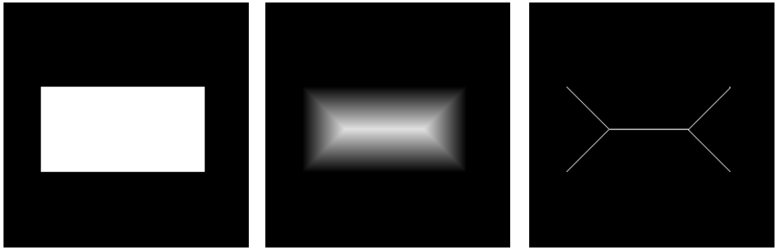
Distance transform

The distance transform on a binary image. For each pixel in the binary image it finds the distance to the nearest non-zero pixel.

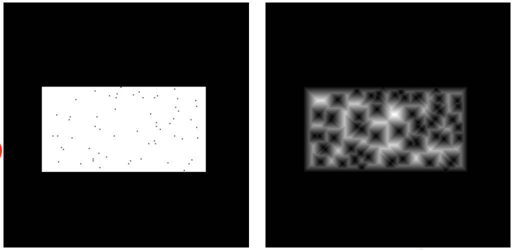
Distance can be defined in different ways. Eucledian distance is often used.



Distance transform



skeletonization



Distance transform is
Noise sensitive

Used as part of a segmentation algorithm