University of Bologna

Computer Vision Laboratory

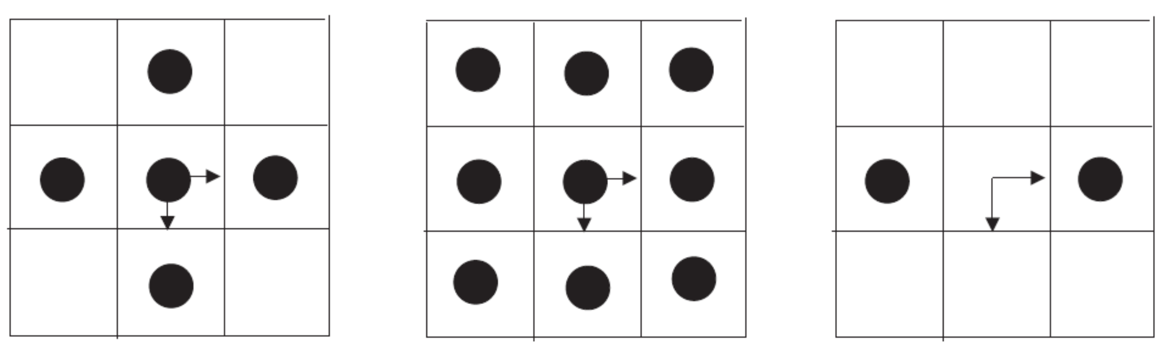
Binary Morphology

Luigi Di Stefano (luigi.distefano@unibo.it)

Definition

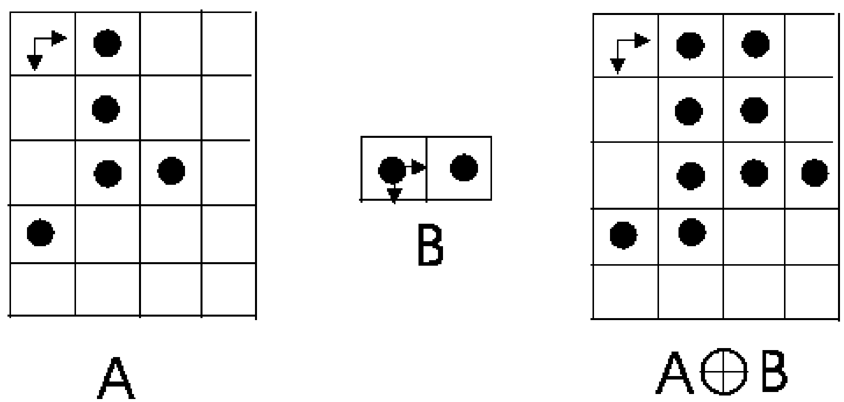
- **Binary Morphology operators are simple though effective tools to improve or analyse binary images, in particular those achieved by any kind of foreground/background segmentation (e.g. based on intensity, colour, motion estimation, joint deployment of multiple cues...).**
- **Binary Morphology operators manipulate *sets* defined over the binary image, which is itself seen as a subset of the discrete plane** $I \subset E^2 = E \times E$ **, with *E* representing the set of integer numbers and** $\mathcal{O}$ **the origin.**
- **Given *I*, the set of foreground pixels will be referred to as *A*, that comprising background pixels as** A^C **. Binary Morphology operators manipulate either *A* or** A^C **through a second set,** $B \subset E^2$**, known as *structuring element*.**

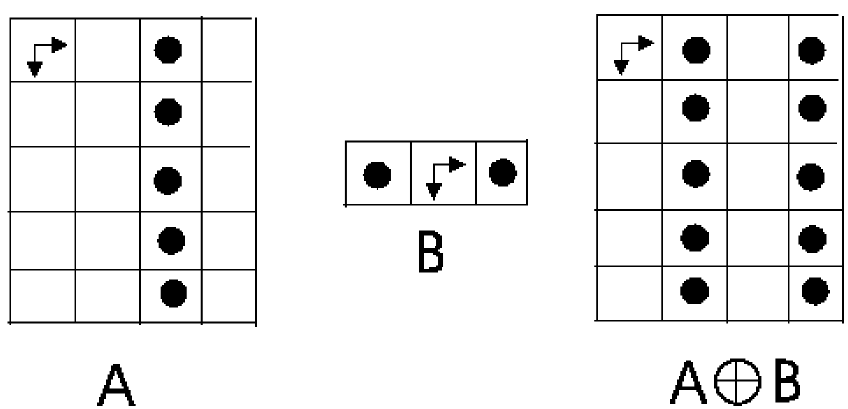
e.g.:



Dilation (Minkowsky Sum)

$$A \oplus B = \{ c \in E^2 : \; c = a + b, \;\; a \in A \;\text{ e }\; b \in B \}$$





Properties of Dilation (1)

- *Traslation* A_t of set A by t is defined as: $A_t = \left\{c \in E^2 : \; c = a + t, \;\; a \in A\right\}$

- It this follows that Dilation can be expressed as the union of the translations of either of the two sets by the elements of the other one:

$$A \oplus B = \bigcup_{b \in B} A_b \; = \; \bigcup_{a \in A} B_a$$

- Some relevant properties are as follows
  - Dilation is commutative:
    $A \oplus B = \; B \oplus A$
  - Dilation is associative:
    $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
  - If the structuring element includes the origin ($\mathcal{O} \in \; B$) then dilation is *extensive*: the initial set is contained in the dilated set ($A \subseteq \; A \oplus B$)
  - Dilation is an increasing transformation:
    $A \subseteq C \Rightarrow A \oplus B \subseteq C \oplus B$

Properties of Dilation (2)

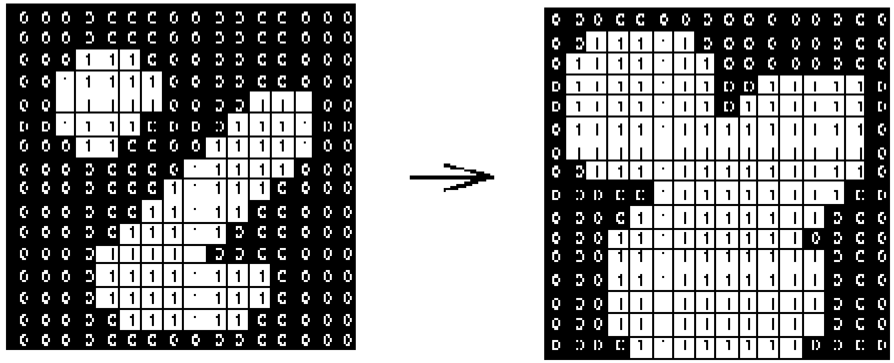
- **Similarly to convolution, associativity allow dilation by a large structuring element to be decomposed into a chain of operations by smaller elements in order to speed-up execution time. E.g., dilation by a $(2n+1)\times(2n+1)$ square can be conveniently accomplished by n successive dilations by a 3×3 square.**

- **Typical structuring elements contain the origin and are symmetric about it, so that dilation expands isotropically foreground regions.**

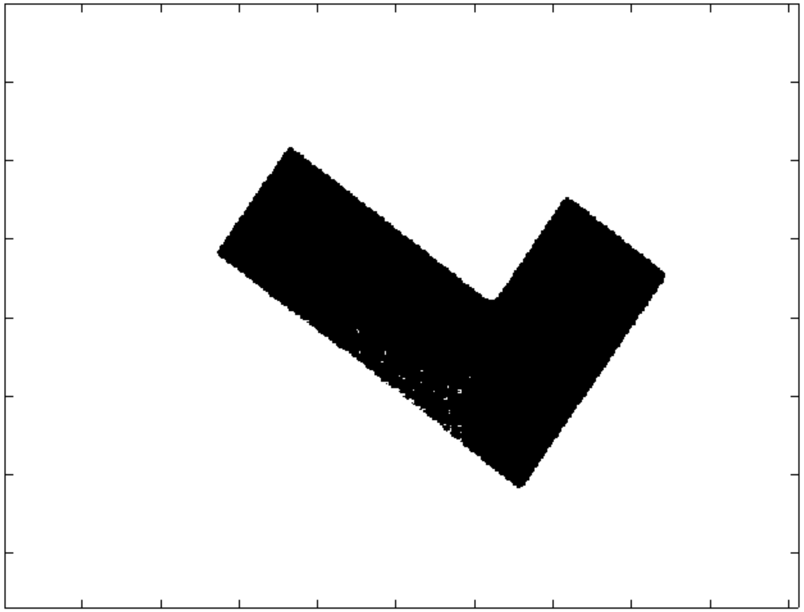
- **Such operators can be deployed to correct segmentation errors dealing with foreground pixels falsely classified as background , e.g. to connect object's parts or fill holes.**

Examples (1)

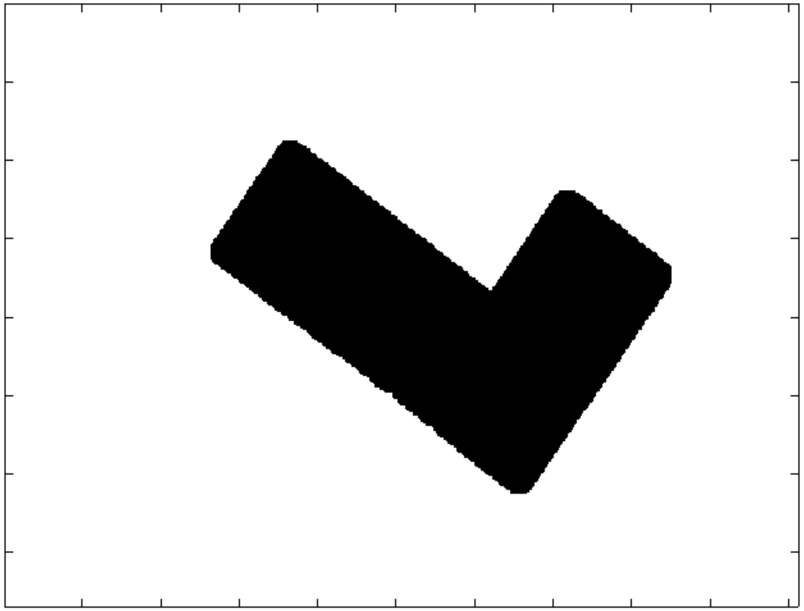
Wrongly disjoint object’s parts yielded by segmentation



Parts get connected upon dilation by a 3×3 square



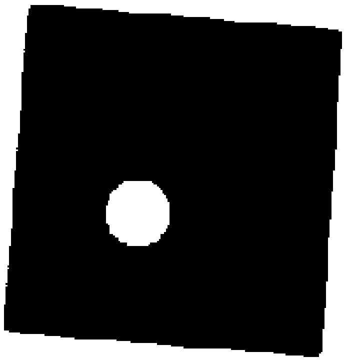
The object shows holes due to binarization errors.



4 dilations by a 3×3 square allow filling the holes.

Examples (2)

The shape of the structuring element determines that of the dilated foreground objects. In the example below, dilation by a circular structuring element results in the outer contour featuring rounded rather than sharp corners. To figure out the dilated shape one may imagine sliding the structuring element so as to traverse all contour points of the original object.



Input binary image

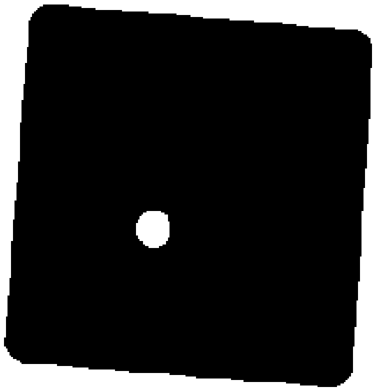
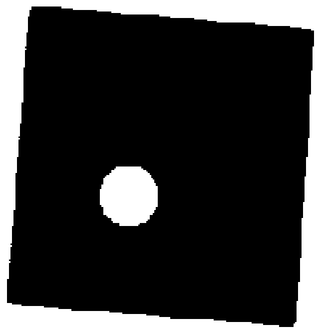


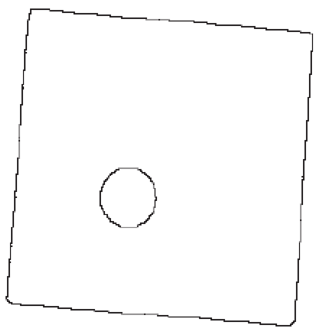
Image dilated by a circular structuring element (radius=11 pixels)

Examples (3)

Dilation by a 3x3 square followed by subtraction of the original image from the dilated one yields the outer contours of foreground regions, outer contour meaning here background pixels adjacent to foreground ones.



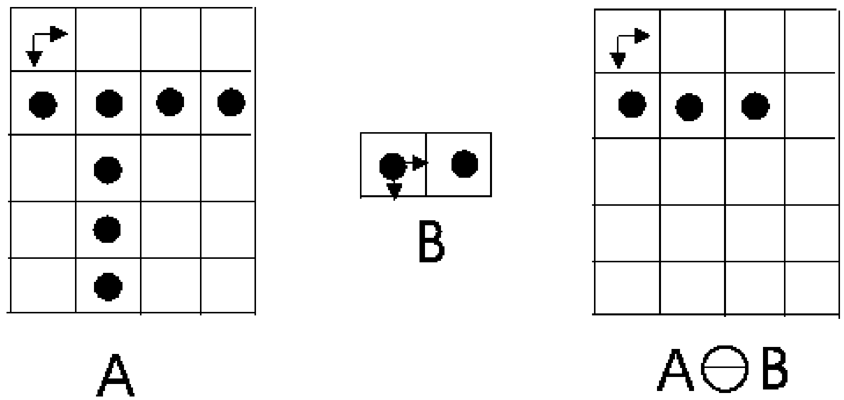
Input binary image

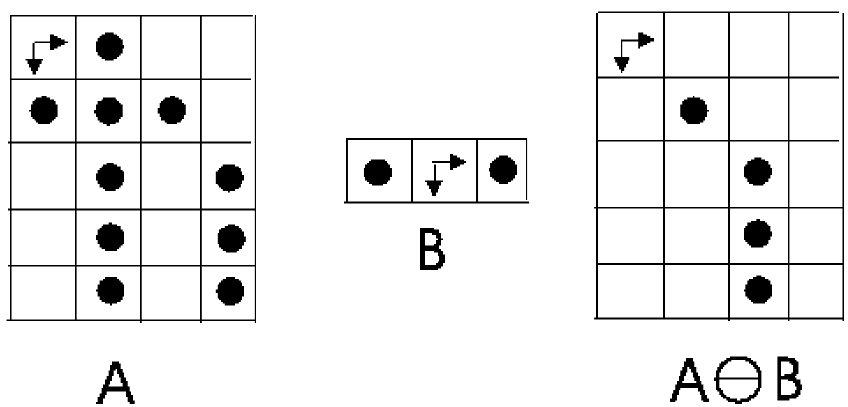


Contours extracted by dilation followed by subtraction

Erosion (Minkowsky Subtraction)

$$A \ominus B = \{c \in E^2 : \; c + b \in A, \; \forall b \in B\}$$





Properties of Erosion (1)

- Erosion can be expressed in terms of translations of the structuring element:

$$A \ominus B = \left\{ c \in E^2 : \; B_c \subseteq A \right\}$$

- Erosion involves subtraction of the elements of one set from those of the other: $A \ominus B = \left\{ c \in E^2 : \forall b \in B \; \exists a \in A : c = a - b \right\}$

- Some relevant properties are as follows

  - Erosion is not commutative:

    $A \ominus B \neq B \ominus A$

  - If the structuring element can be decomposed in terms of dilations then erosion is associative:

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

  - If the structuring element includes the origin ($\mathcal{O} \in B$) then erosion is anti-extensive: the eroded set is contained into the original one ($A \ominus B \subseteq A$)

  - Erosion is an increasing transformation:

    $A \subseteq C \Rightarrow A \ominus B \subseteq C \ominus B$

Properties of Erosion (2)

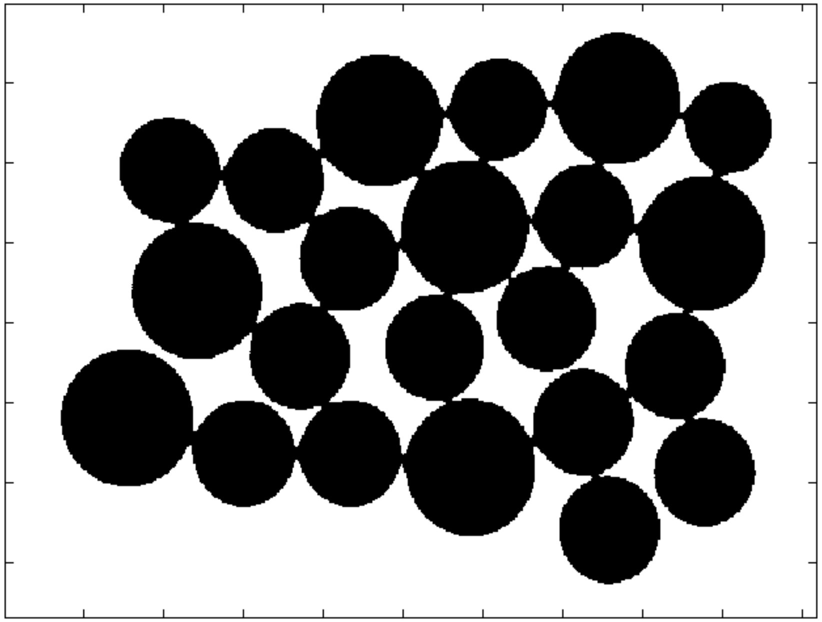
- **Associativity allow erosion by a large structuring element to be decomposed into a chain of operations by smaller elements in order to speed-up execution time. E.g., erosion by a $(2n+1)\times(2n+1)$ square can be conveniently accomplished by n successive erosions by a 3×3 square.**

- **Typical structuring elements contain the origin and are symmetric about it, so that erosion shrinks isotropically foreground regions.**

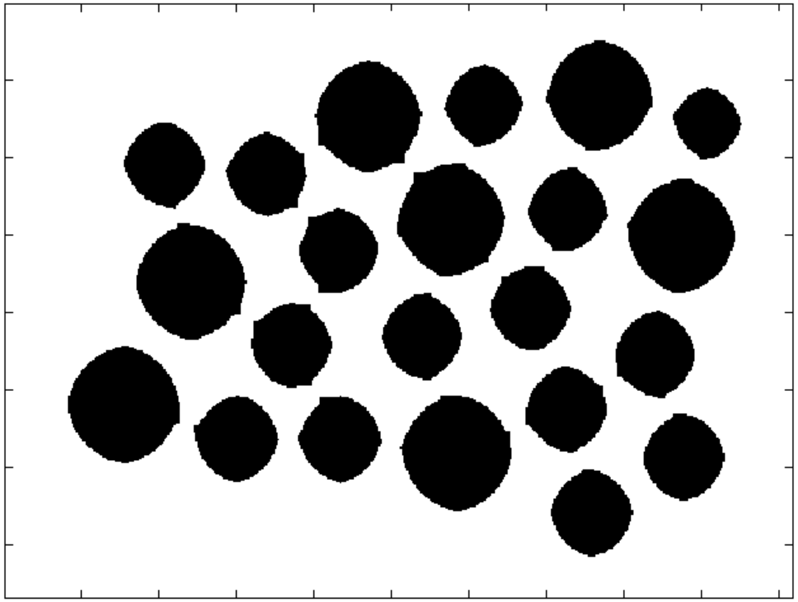
- **Such operators can be deployed to correct segmentation errors dealing with background pixels falsely classified as foreground, e.g. to split wrongly connected objects.**

Examples (1)

Wrongly connected objects yielded by segmentation

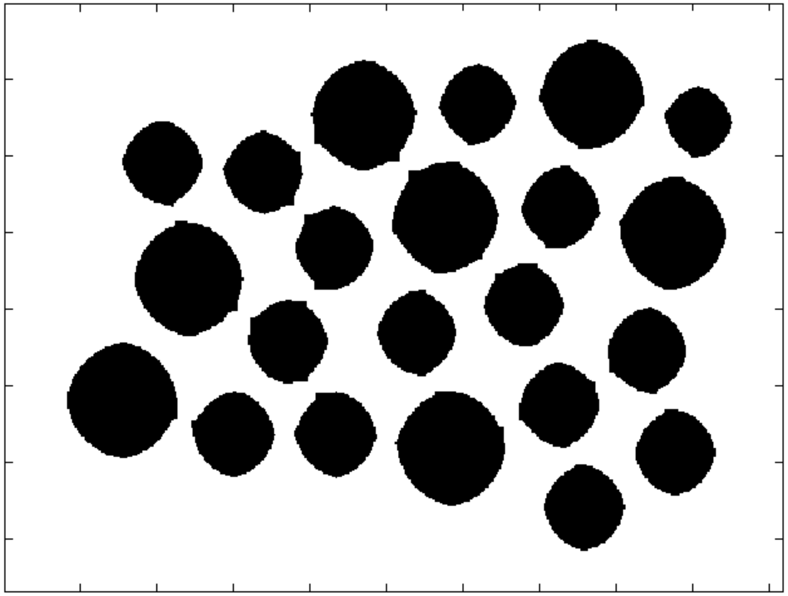


Objects can be split (e.g. to allow counting them correctly) by 5 successive erosions with 3×3 square

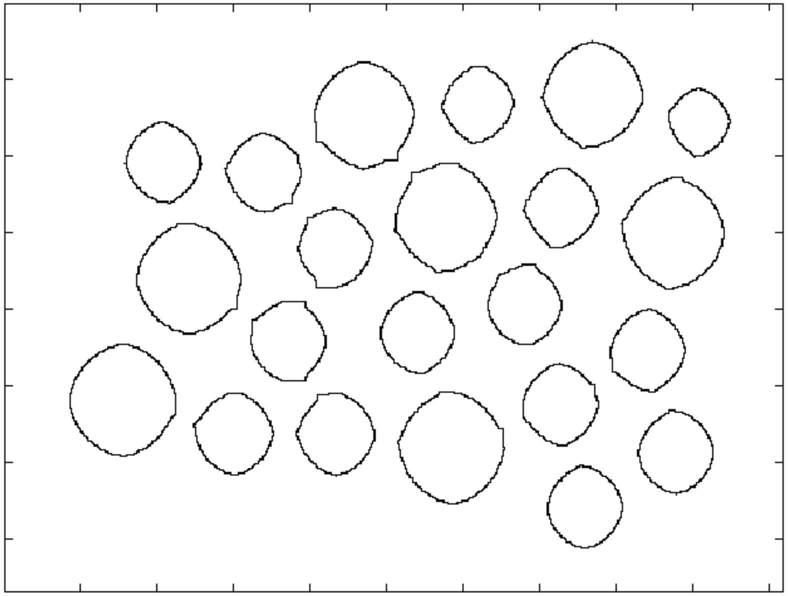


Examples (2)

Erosion by a 3x3 square followed by subtraction of the eroded image from the original one yields the inner contours of foreground regions, inner contour meaning here foreground pixels adjacent to background ones.



Input binary image



Contours extracted by erosion followed by subtraction

Duality between Dilation and Erosion

- **Given $\breve{B}$:**

$$\breve{B} = \left\{ \breve{b} : \ \breve{b} = -b, \ b \in B \right\}$$

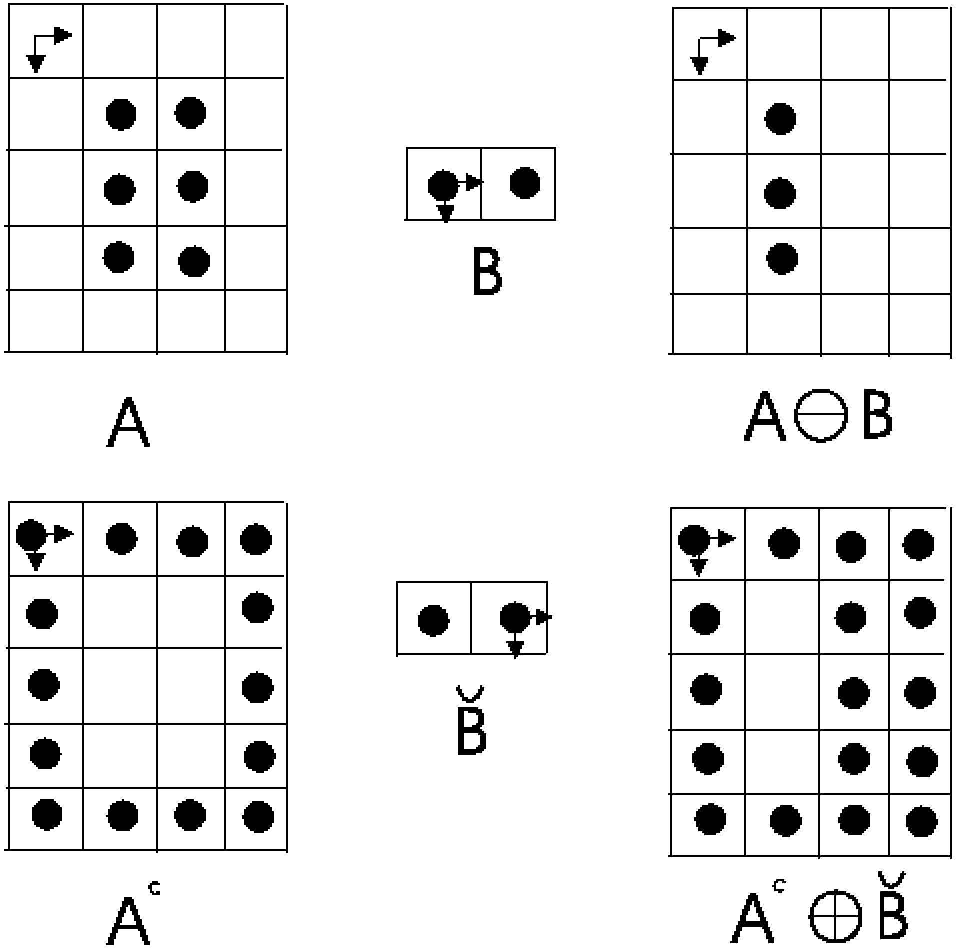
It can be shown that

$$(A \oplus B)^c = A^c \ominus \breve{B}$$
$$(A \ominus B)^c = A^c \oplus \breve{B}$$

- **If B is symmetric $\left(B = \breve{B} \right)$**

$$(A \oplus B)^c = A^c \ominus B$$
$$(A \ominus B)^c = A^c \oplus B$$

i.e. dilation of foreground is equivalent to erosion of background, and viceversa.

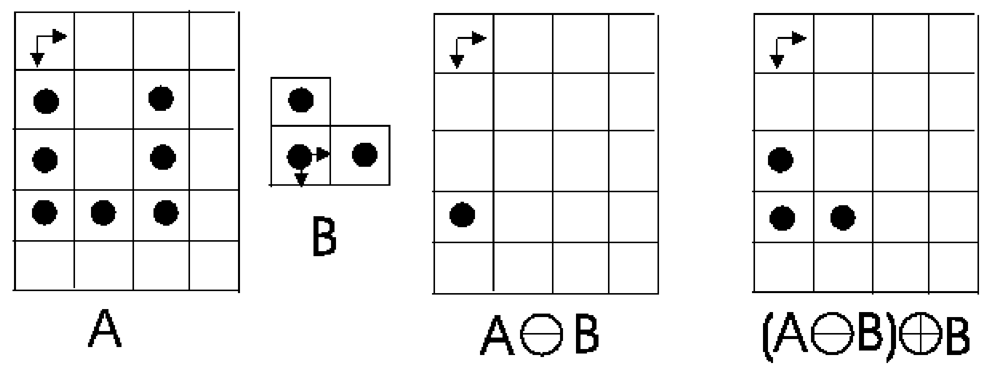


Opening and Closing

- **Erosion and dilation by the same structuring element can be chained to remove selectively from either foreground or background the parts that do not match exactly the structuring element <u>without causing any distortion to the other parts</u>.**

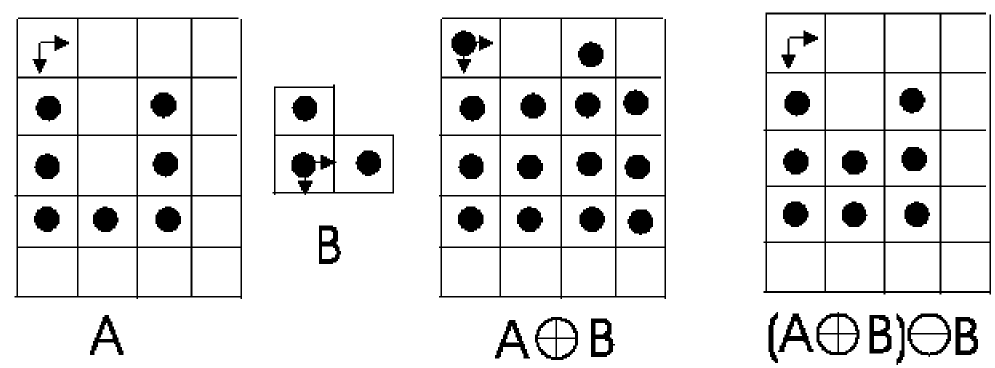
  - **Erosion followed by Dilation is known as Opening:**

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



  - **Dilation followed by Erosion is known as Closing:**

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



Properties (1)

- Contrary to erosion and dilation, opening and closing are idempotent:

$$(A \circ B) \circ B = A \circ B \qquad (A \bullet B) \bullet B = A \bullet B$$

- Opening e closing are not commutative

$A \circ B \neq B \circ A, \quad A \bullet B \neq B \bullet A$

- Opening is anti-extensive, closing is extensive:

$A \circ B \subseteq A, \quad A \bullet B \supseteq A$

- Opening e closing are increasing transformations:

$A \subseteq C \ \Rightarrow \ A \circ B \subseteq C \circ B, \quad A \bullet B \subseteq C \bullet B$

Properties (2)

- The result of an opening operation can be expressed as the union of those elementary foreground parts that exactly match the structuring element:

$$A \circ B = (A \ominus B) \oplus B = \bigcup_{y \in A \ominus B} B_y = \bigcup_{B_y \subseteq A} B_y$$

Opening can be thus thought of as comparing the structuring element to foreground parts, so as to remove those which turn out different and keep unaltered equal ones.

- Duality between erosion and dilation implies duality between opening and closing:

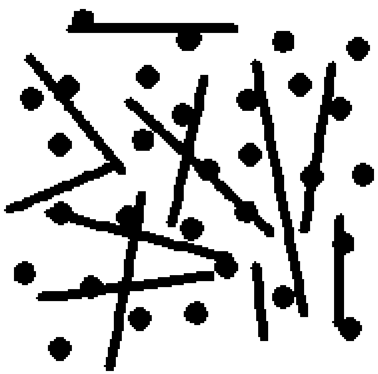
$$(A \circ B)^c = [(A \ominus B) \oplus B]^c = (A \ominus B)^c \ominus \breve{B} = \left(A^c \oplus \breve{B}\right) \ominus \breve{B} = A^c \bullet \breve{B}$$
$$(A \bullet B)^c = [(A \oplus B) \ominus B]^c = (A \oplus B)^c \oplus \breve{B} = \left(A^c \ominus \breve{B}\right) \oplus \breve{B} = A^c \circ \breve{B}$$

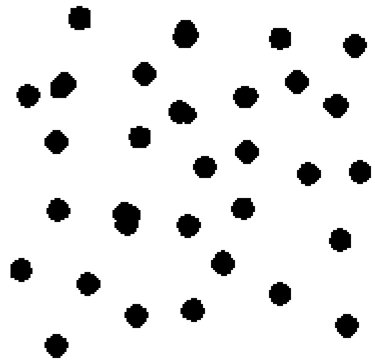
If B is symmetric $\left(B = \breve{B}\right)$: $(A \circ B)^c = A^c \bullet B$, $(A \bullet B)^c = A^c \circ B$

Because of duality, closing can be thought of as comparing the (flipped) structuring element to background parts, so as to remove (i.e. change to foreground) those which turn out different and keep unaltered equal ones.

Examples (1)

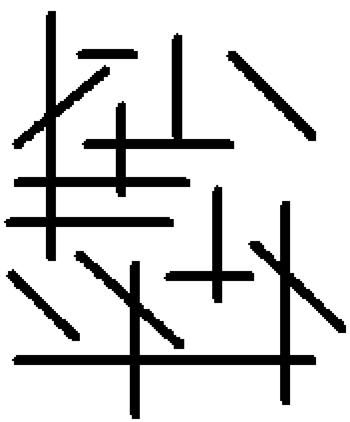
Input binary image



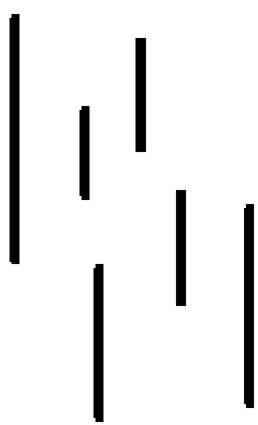


Opening by a circular structuring element (diameter=11 pixels) allows detecting circular objects.

Input binary image

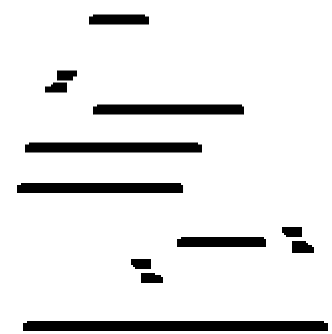


Opening by a vertical structuring element (9×3 pixels) …



.. allows detecting vertical bars.

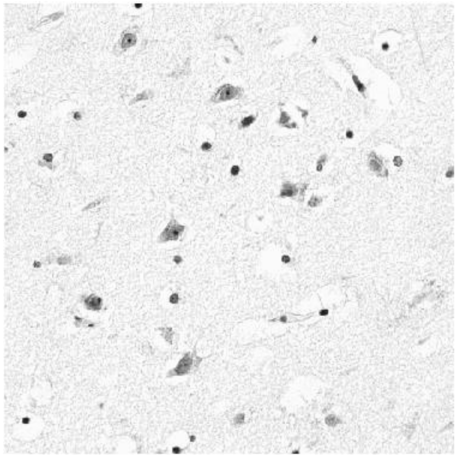
Opening by an horizontal structuring element (3×9 pixels)…

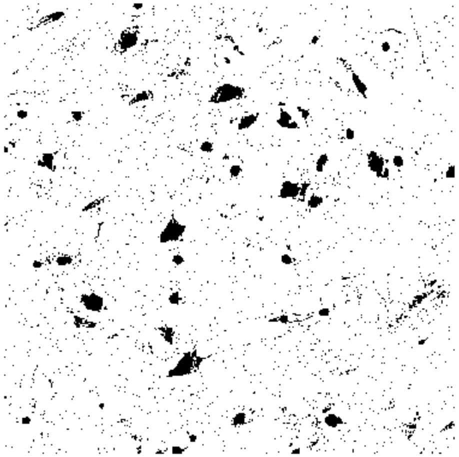


…allows detecting horizontal bars.

Examples (2)

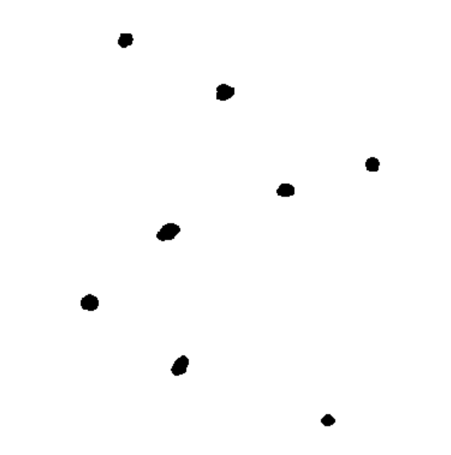
Microscopic image of cerebral tissue depicting nerve cells (larger with a graysh periphery and a dark nucleus) and glial cells (smaller, circular and dark).





Binarized image (T=210)

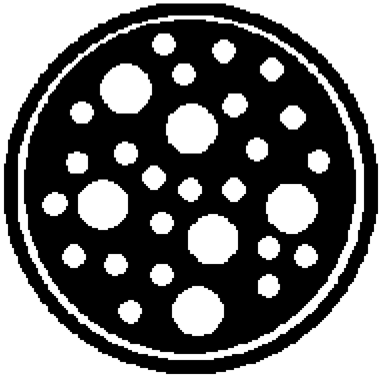
Opening by a large circular structuring element (diameter=11 pixels) allows detecting most of the nerve cells.



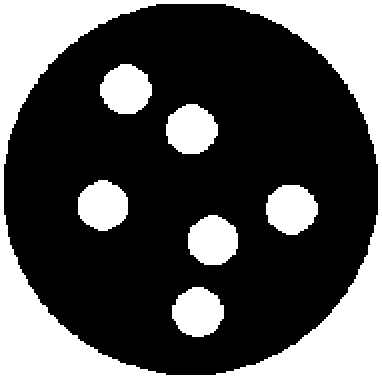
… while opening by a small one (i.e. 7 pixels) cannot isolate glial cells

Examples (3)

Input binary image



Closing by a circle smaller than the big holes and larger than the small ones …



…removes the small holes (and the external thin circular contour alike).

Hit-and-Miss Transform

- **Binary Morphology operator to detect specific patterns in binary images. The resulting image highlights the positions of the sought-for pattern.**

- **More precisely, the Hit-and-Miss transform fires at the locations where the binary image is exactly identical to the structuring element.**

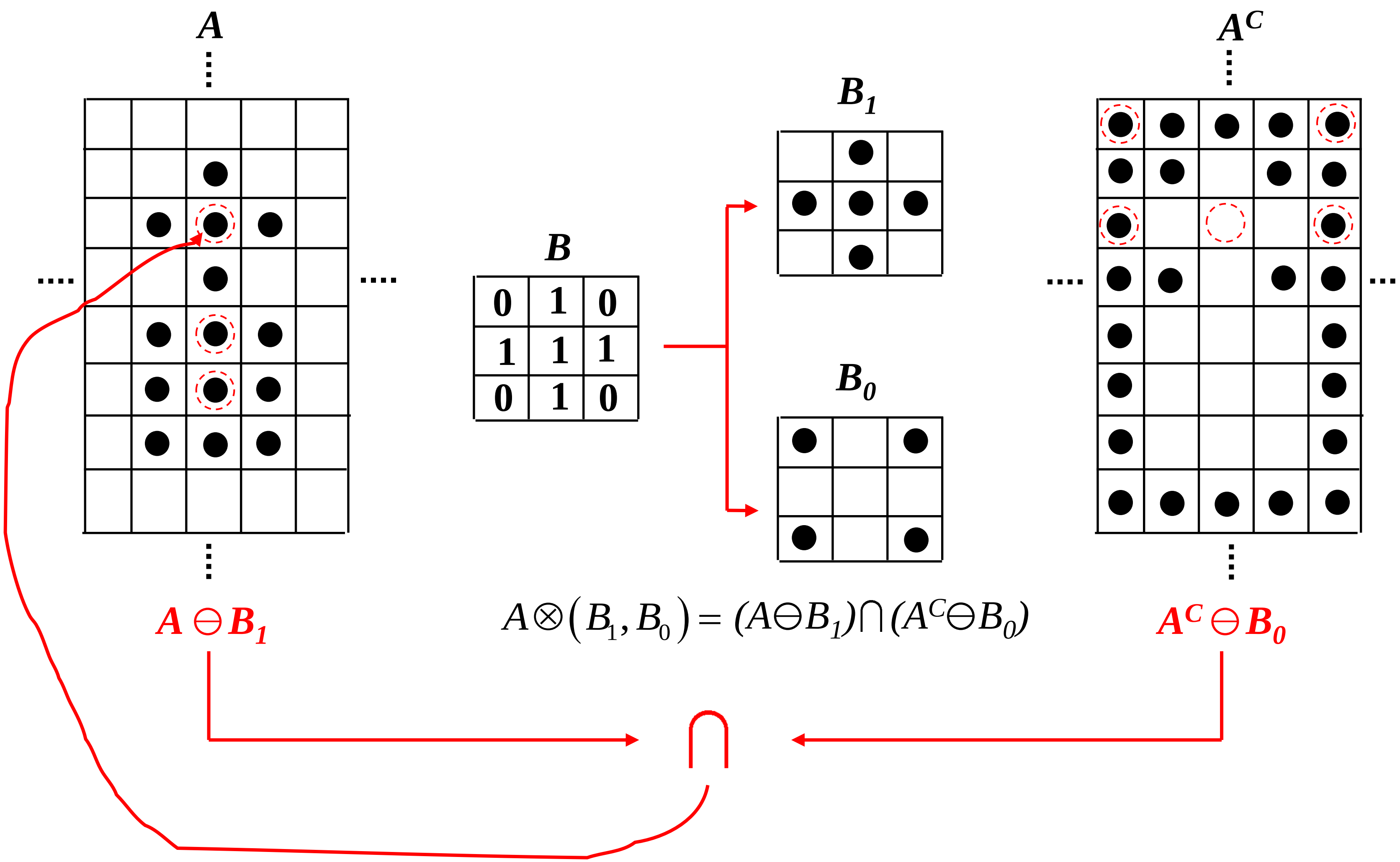
- **Given a structuring element, B comprising both foreground (B_1) and background (B_0) points, the Hit-and-Miss transform is defined as follows:**

$$A \otimes (B_1, B_0) = (A \ominus B_1) \cap (A^c \ominus B_0) \text{ with } B_1 \cap B_0 = \emptyset.$$

e.g.:

$$B = \begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix} \Rightarrow B1 = \{(0,0)\},\ B0 = \{(-1,0),(1,0),(0,-1),(0,1)\}$$

Example (1)



Example (2)

Structuring elements to detect corners in binary images:

	1	
0	1	1
0	0	

B_{c_1}

	1	
1	1	0
	0	0

B_{c_2}

	0	0
1	1	0
	1	

B_{c_3}

0	0	
0	1	1
	1	

B_{c_4}

Input binary image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	0	0	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$\longrightarrow$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$= \bigcup_{i=1}^{4} I \otimes B_{c_i}$$

Output image with detected corner structures

Main References

1) R. Gonzales, R. Woods, “Digital Image Processing – Third Edition”, Pearson Prentice-Hall, 2008.

2) R. Fisher, S. Perkins, A. Walker, E. Wolfart, “Hypermedia Image Processing Reference”, Wiley, 1996 (http://homepages.inf.ed.ac.uk/rbf/HIPR2/)

3) R. Haralick, L. Shapiro “Computer and Robot Vision – Vol. I, Addison-Wesley Publishing Company, 1993.

4) M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis and Machine Vision”, Chapman & Hall, 1993.