Image Processing and Computer Vision Notes

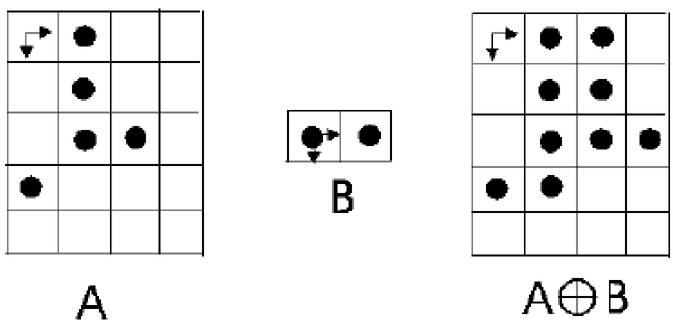
by Mattia Orlandi

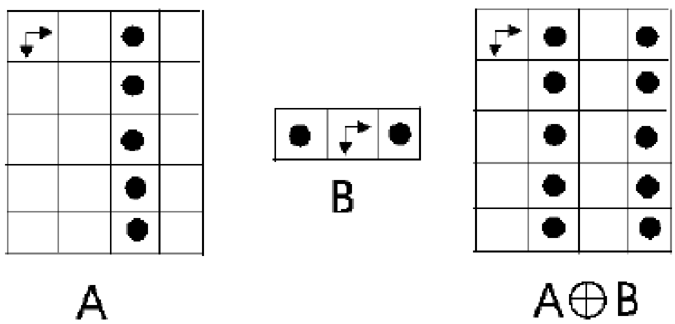
6. Binary Morphology

- Binary Morphology operators are tools used to improve or analyze binary images, in particular those achieved by foreground/background segmentation.
- They **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 with the origin in $\mathcal{O}$.
- Given I, the set of foreground pixels will be referred to as A, whereas the set of background pixels will be referred to as A^C.
- **Binary Morphology operators manipulate** either A **or** A^C **through a second set**, $B \subset E^2$, known as **structuring element**.

6.1. Dilation (Minkowsky Sum)

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





Properties

- Translation A_t of set A by t is defined as $A_t = \{c \in E^2 : c = a + t, a \in A\}$
- Dilation can be **expressed as** the **union** of the **translations of either of the two sets by the elements of the other** one:

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

- **Commutativity**: $A \oplus B = B \oplus A$
- **Associativity**: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- If the **structuring element includes the origin** ($\mathcal{O} \in B$), then **dilation is extensive**, that is the **initial set** is **contained in the dilated one** ($A \subseteq A \oplus B$).
- It is an **increasing transformation**:

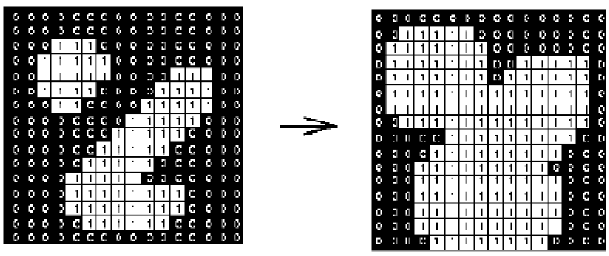
$$A \subseteq C \Rightarrow A \oplus B \subseteq C \oplus B$$

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

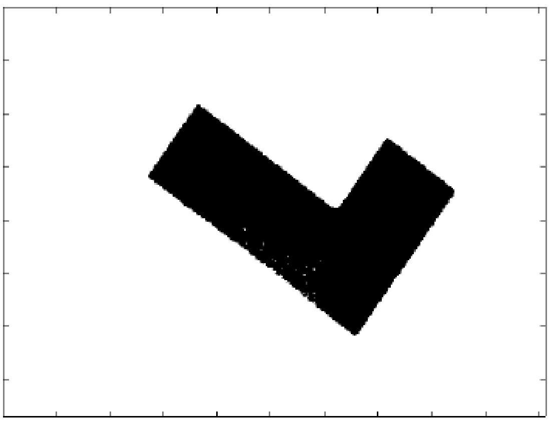
- Typical structuring elements contain the **origin** and are **symmetric** about it $\Rightarrow$ dilation **expands isotropically foreground regions**.

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

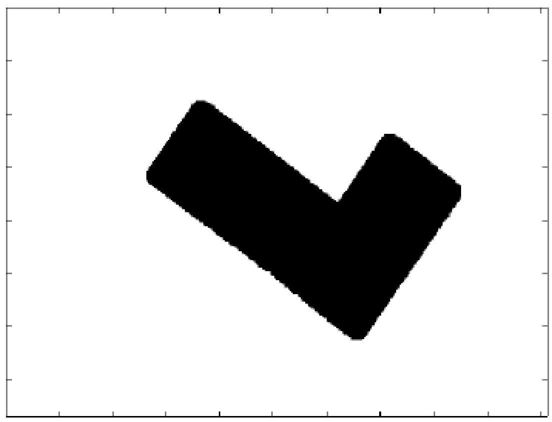
Wrongly disjoint object's parts yielded by segmentation



Parts get connected upon dilation by a 3×3 square



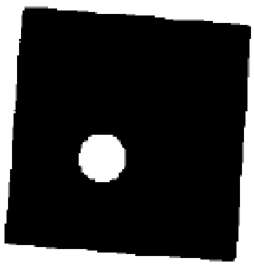
The object shows holes due to binarization errors.



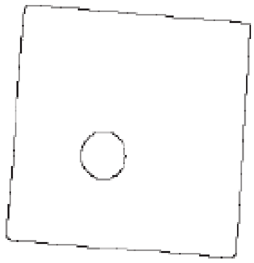
4 dilations by a 3×3 square allow filling the holes.

- The **shape of structuring element determines** the **shape of dilated foreground objects** (e.g. a circular structuring elements produce a dilated object with rounded rather than sharp corners).

- A **dilation followed by a subtraction** of original image from the dilated one **yields outer contours** of foreground regions.



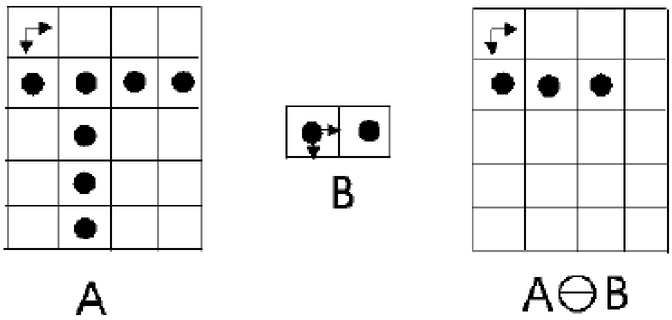
Input binary image

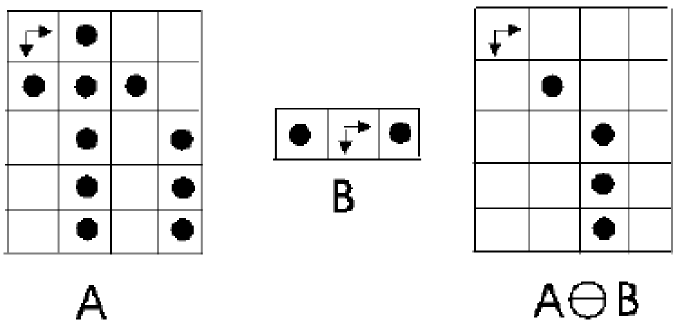


Contours extracted by dilation followed by subtraction

6.2. Erosion (Minkowsky Subtraction)

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





Properties

- It can be **expressed in terms of translations of structuring element**:

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

- It involves **subtraction of elements of one set from those of other**:

$$A \ominus B = \{c \in E^2 : \forall b \in B \ \exists a \in A \mid c = a - b\}$$

- **Non-Commutativity** (since subtraction is not commutative): $A \ominus B \neq B \ominus A$
- **Associativity** (**if structuring element** is **decomposable in terms of dilations**): $A \ominus (B \oplus C) = (A \ominus B) \ominus C$

- If **structuring element includes the origin** ($\mathcal{O} \in B$) then **erosion is anti-extensive**, that is the **eroded set** is **contained in the original one** ($A \ominus B \subseteq A$).
- It is an **increasing transformation**:

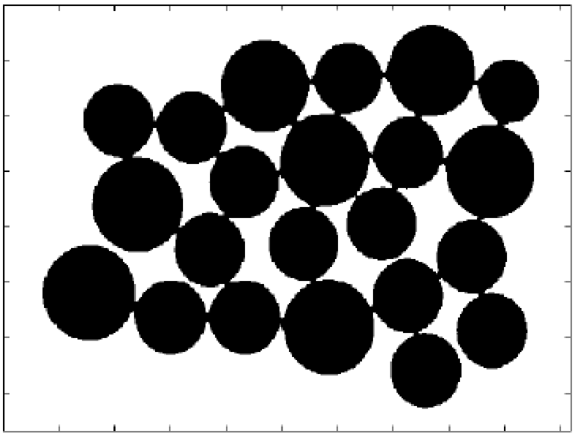
$$A \subseteq C \Rightarrow A \ominus B \subseteq C \ominus B$$

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

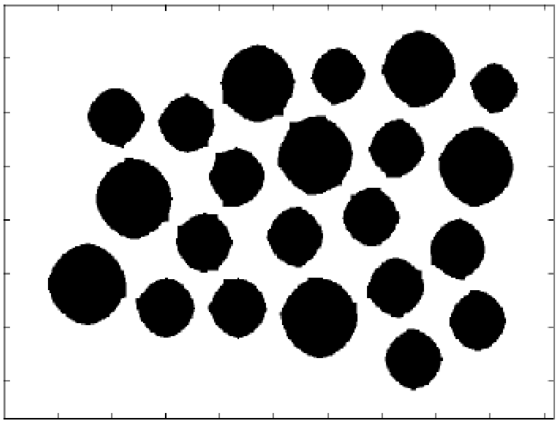
- Typical structuring elements contain the **origin** and are **symmetric** about it $\Rightarrow$ erosion **shrinks isotropically foreground regions**.

- Useful to **correct segmentation errors** dealing with **background pixels falsely classified as foreground**, e.g. to split wrongly connected objects.

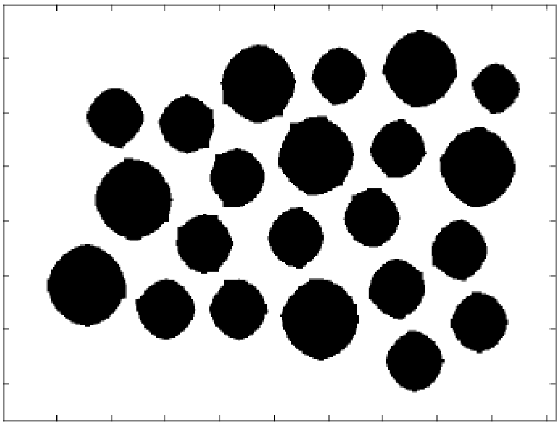
Wrongly connected objects yielded by segmentation



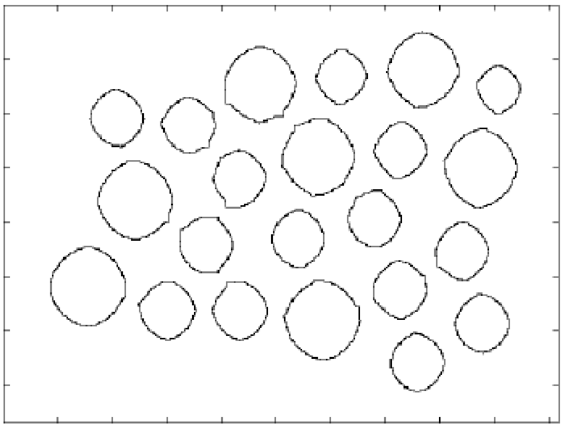
Objects can be split (e.g. to allow counting them correctly) by 5 successive erosions with 3×3 square



- An **erosion followed by a subtraction** of eroded image from the original one **yields outer contours** of foreground regions.



Input binary image



Contours extracted by erosion followed by subtraction

6.3. Duality between Dilation and Erosion

- Given $\check{B}$:

$$\check{B} = \{\check{b} : \check{b} = -b,\ b \in B\}$$

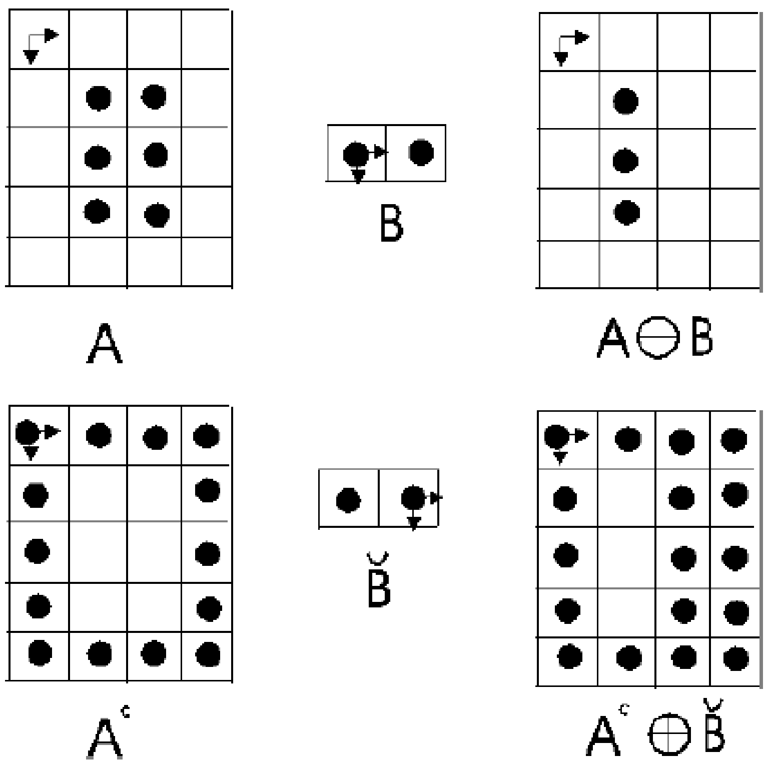
  it can be shown that:

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

- If B is symmetric ($B = \check{B}$):

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

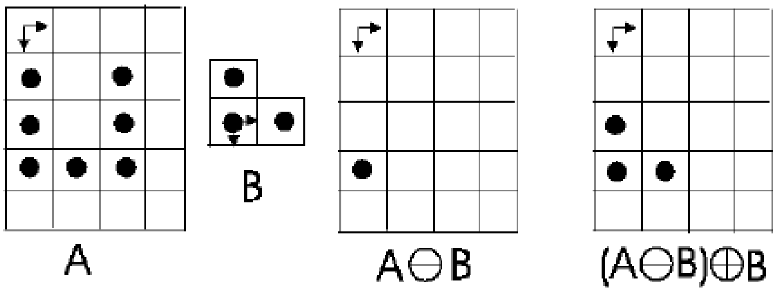
  i.e., **dilation of foreground is equivalent to erosion of background, and viceversa**.



6.4. Opening and Closing

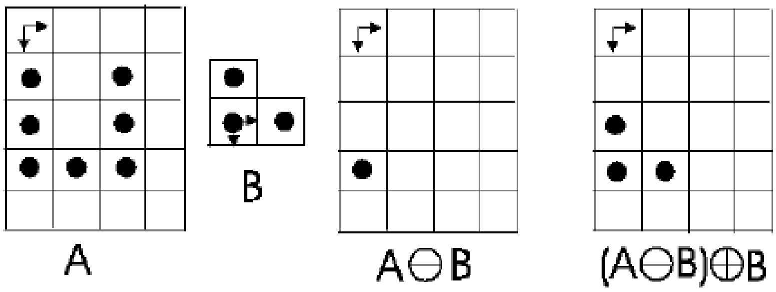
- **Dilation** grows blindly a foreground region, creating **false positives**; **erosion** changes the shape of the object, creating **false negatives** $\Rightarrow$ necessity of "smarter" versions of dilation and erosion.
- By **chaining erosion and dilation** by the **same structuring element** is possible to **remove selectively** from foreground/background the **parts that do not match** the structuring element, **without distorting the other parts**.
- **Opening** (Erosion followed by Dilation):

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



- **Closing** (Dilation followed by Erosion):

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



Properties

- **Idempotency** (unlike erosion and dilation):

$$(A \circ B) \circ B = A \circ B, \quad (A \bullet B) \bullet B = A \bullet B$$

- **Non-Commutativity**:

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

- **Opening** is **anti-extensive**, **closing** is **extensive**:

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

- Opening and closing are **increasing transformations**:

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

- Result of opening can be **expressed as** the **union** of the **elementary foreground elements** that **exactly match** the **structuring element**:

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

  it can be thus thought of as a **comparison between structuring element and foreground parts**, after which the **non-matching parts are removed**.
- Duality between erosion and dilation implies **duality between opening and closing**:

$$(A \circ B)^c = [(A \ominus B) \oplus B]^c = (A \ominus B)^c \ominus \check{B} = (A^c \oplus \check{B}) \ominus \check{B} = A^c \bullet \check{B},$$
$$(A \bullet B)^c = [(A \oplus B) \ominus B]^c = (A \oplus B)^c \oplus \check{B} = (A^c \ominus \check{B}) \oplus \check{B} = A^c \circ \check{B}$$

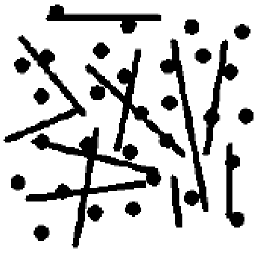
- If B is symmetric ($B = \check{B}$):

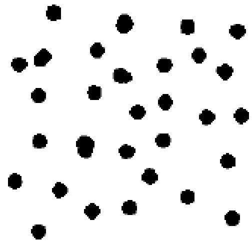
$$(A \circ B)^c = A^c \bullet B, \quad (A \bullet B)^c = A^c \circ B$$

- **Opening and closing** are the **same algorithm**, **one working on the foreground and one on the background**.

- Closing can be thought of as a **comparison between flipped structuring element and background parts**, after which the **non-matching parts are removed**.

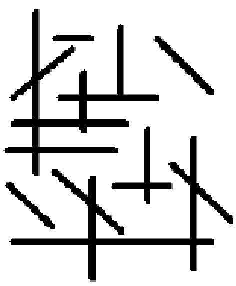
Input binary image



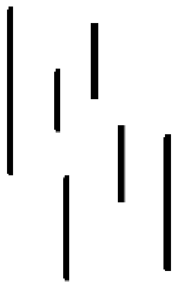


Opening by a circular structuring element (diameter=11 pixels) allows detecting circular objects.

Input binary image

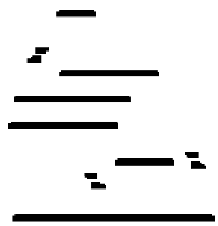


Opening by a vertical structuring element (9×3 pixels) …



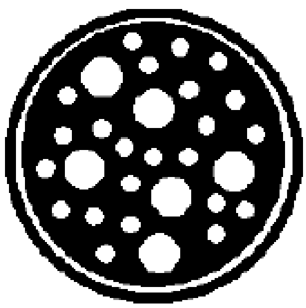
.. allows detecting vertical bars.

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

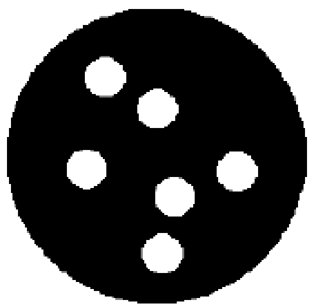


…allows detecting horizontal bars.

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).