Fundamentals of Computer Vision

Unit 5: Non-Linear Filtering

Jorge Bernal

Index

1

Median Filters

Median Filters

- **Limitation of Linear Filters**
  - Frequency shaping enhance some frequency components and suppress the others
  - For individual frequency component, we cannot differentiate its “desirable” and “undesirable” parts
- **Nonlinear Filters**
  - Cannot be expressed as convolutions
  - Cannot be expressed as frequency shaping
- **“Nonlinear” Means Everything** (other than linear)
  - Need to be more specific
  - They often use heuristics
  - We will study some “nice” ones

Median Filters

- **Order Statistics (OS)**
  - Given a set of numbers
    Denote the OS as
    such that

$$\boldsymbol{x} = \{x_1, x_2, \cdots, x_{2M+1}\}$$

$$\boldsymbol{x}_{OS} = \{x_{(1)}, x_{(2)}, \cdots, x_{(2M+1)}\}$$

$$x_{(1)} \le x_{(2)} \le \cdots \le x_{(M+1)} \le \cdots \le x_{(2M+1)}$$

max value

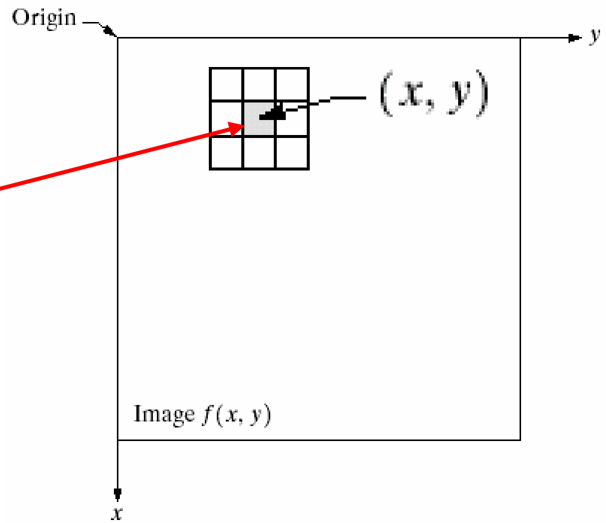
min value

middle value

- **Median**
  - Define

$$Median\{x_1, x_2, \cdots, x_{2M+1}\} = x_{(M+1)}$$

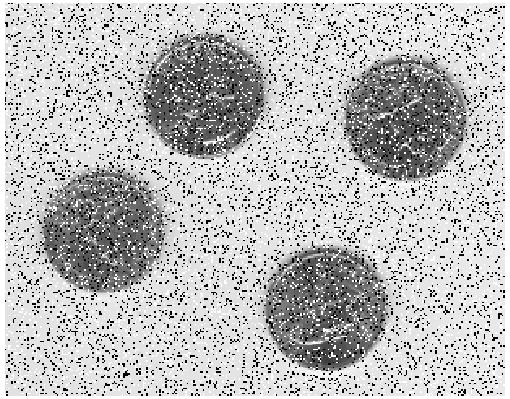
- **Applying Median Filters to Images**
  - Use sliding windows
  - Typical windows:
    3x3, 5x5, 7x7, other shapes ……



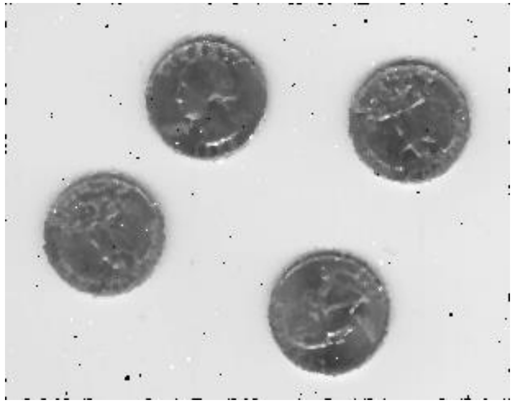
Median Filters

original

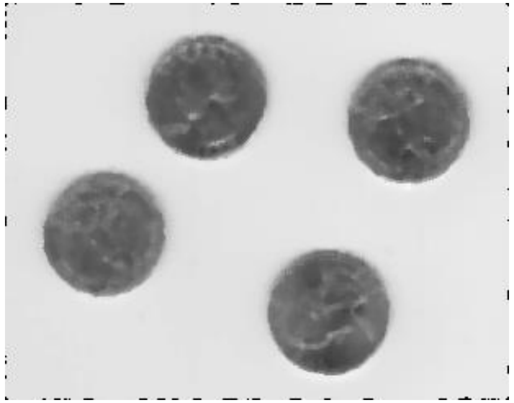
noisy ($p_a = p_b = 0.1$)



median filtered 3x3 window



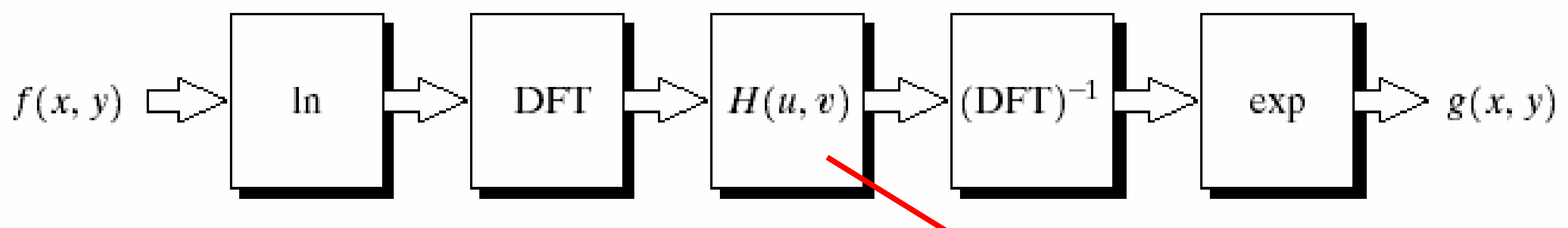
median filtered 5x5 window



2

Homomorphic Filters

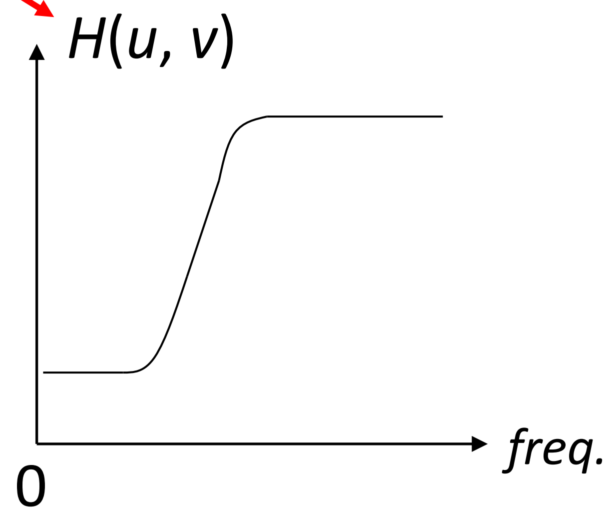
Homomorphic Filters



$$f(x, y) = \underbrace{i(x, y)}_{\text{illumination}}\ \underbrace{r(x, y)}_{\text{reflectance}}$$

illumination
(slowly varying)
(low-frequency)

reflectance
(fast variation)
(high-frequency)



$$\ln[f(x, y)] = \ln[i(x, y)] + \ln[r(x, y)]$$

Key: linear separation

Homomorphic Filters

before

after

3

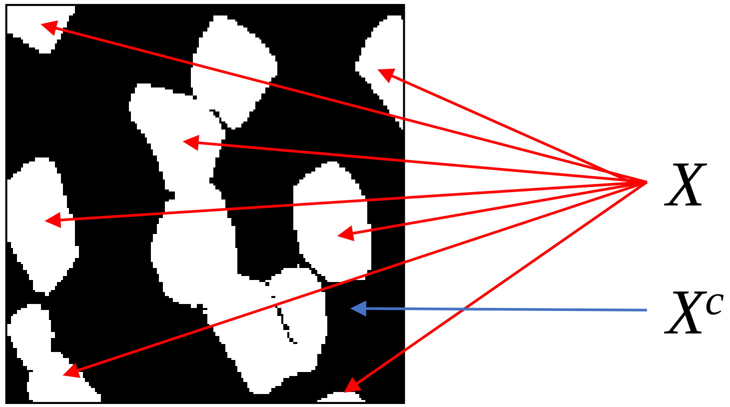
Mathematical Morphology

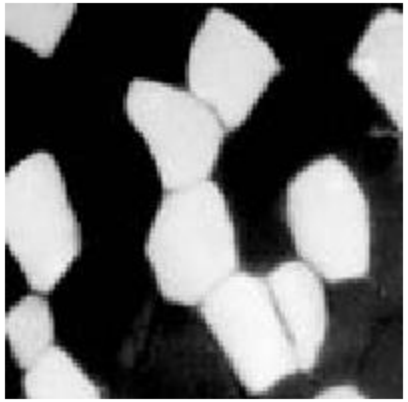
Mathematical Morphology

- Self-sufficient framework for image processing and analysis, created at the École des Mines (Fontainebleau) in 70’s by Jean Serra, Georges Mathéron, from studies in science materials
- Conceptually simple operations combined to define others more and more complex and powerful
- Operations have a clear geometrical meaning
- Powerful for image analysis

Mathematical Morphology

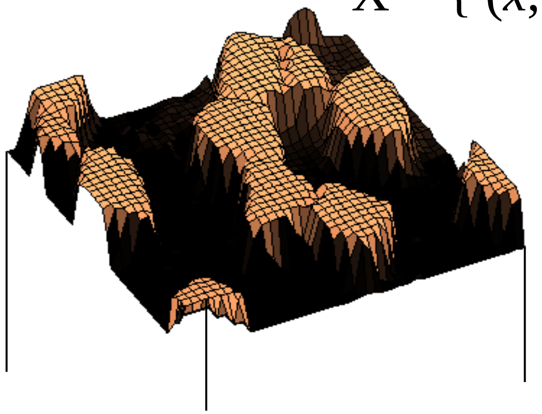
Binary and grey-level images seen as sets





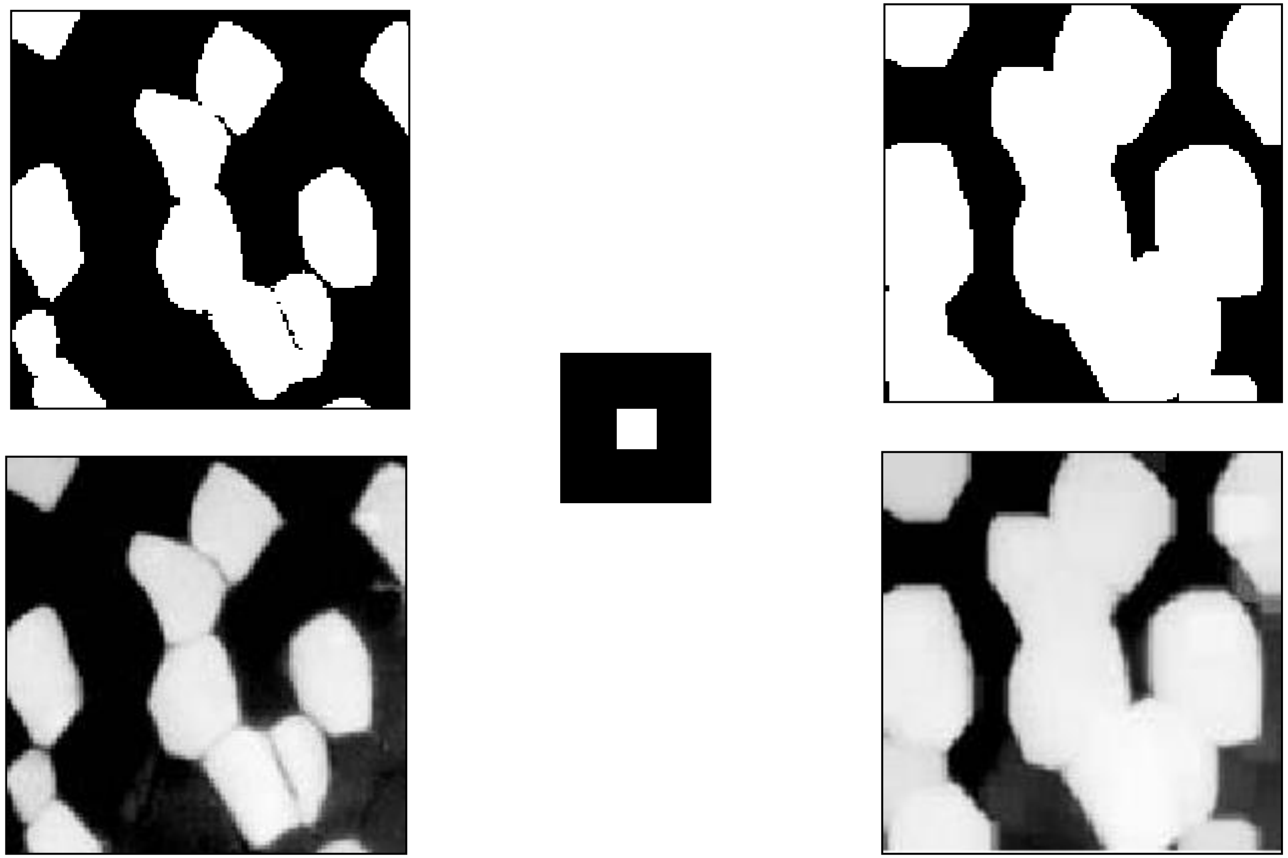
$f(x,y)$

$X = \{ (x, y, z) \;,\; z \leq f(x,y) \}$



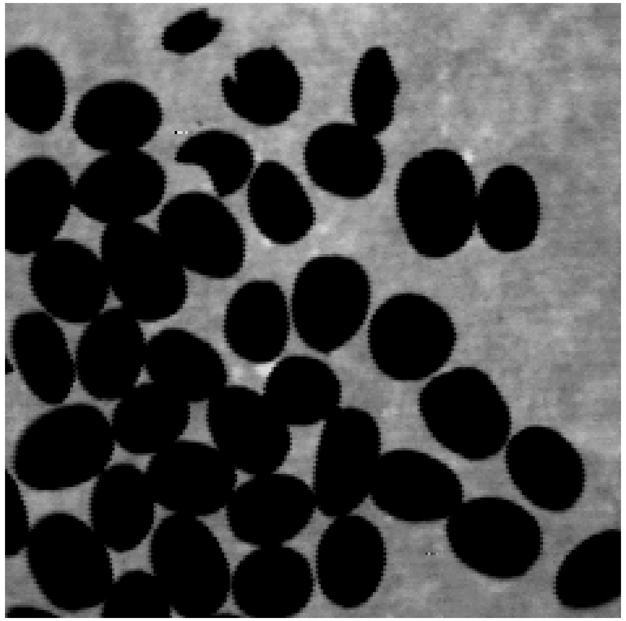
Mathematical Morphology

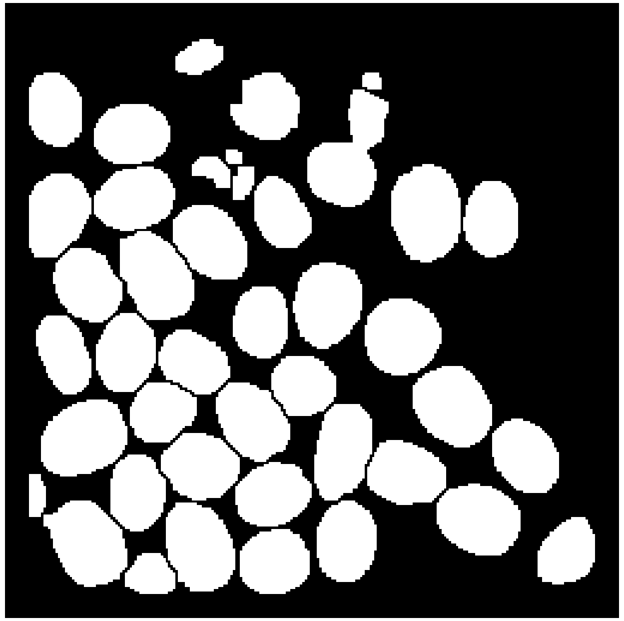
Operations defined as interaction of images with a special set, the *structuring element*



Mathematical Morphology

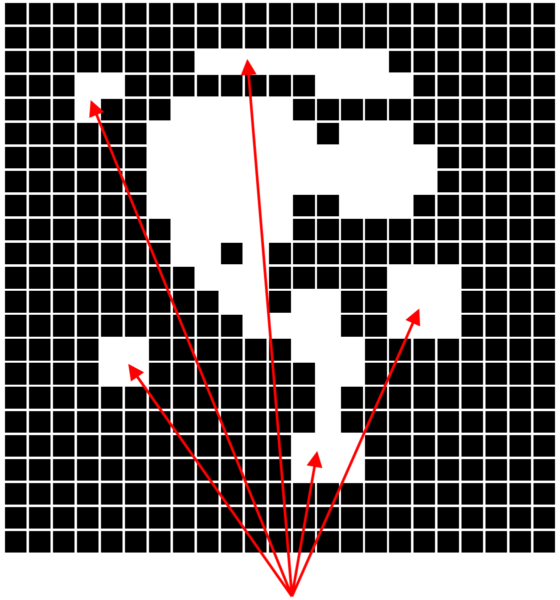
Mathematical Morphology





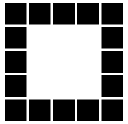
Binary Morphology

Notation



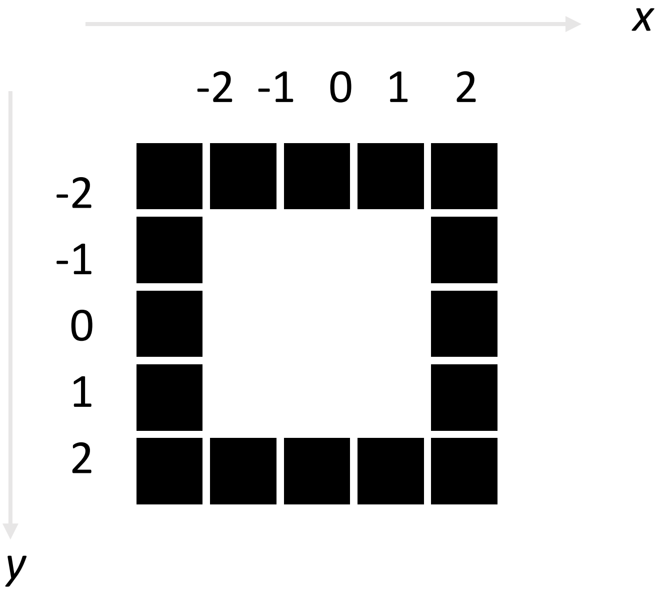
X

No necessarily compact nor filled



B

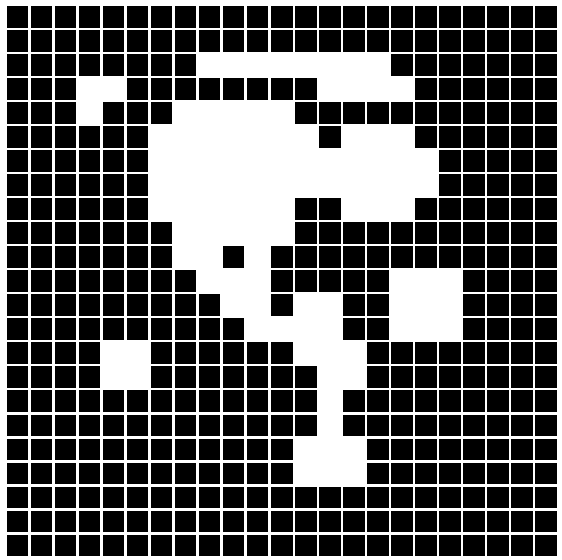
A special set : the structuring element

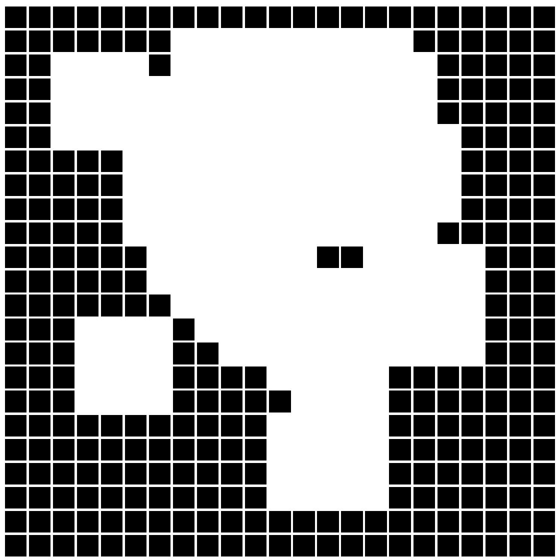


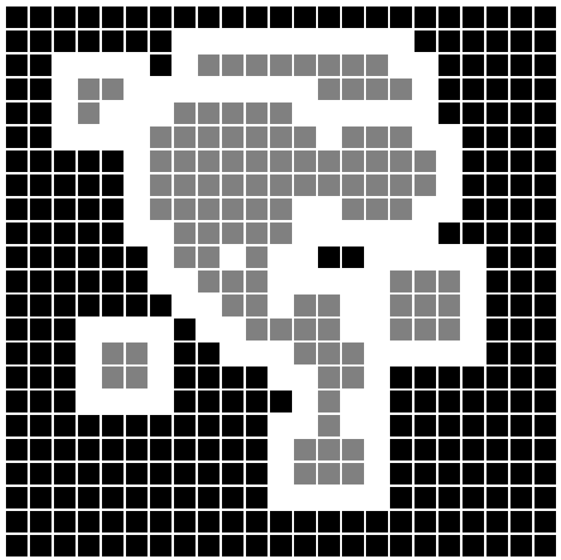
Origin at center in this case, but not necessarily centered nor symmetric

Binary Morphology: Dilate

Dilation : $x = (x_1,x_2)$ such that if we center B on them, then the so translated B intersects X.

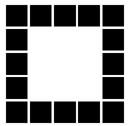






X

difference



B

Binary Morphology: Dilate

Dilation : $x = (x_1,x_2)$ such that if we center B on them, then the so translated B intersects X.

How to formulate this definition ?

1) Literal translation

$$B_x \stackrel{def}{=} \{b + x \mid b \in B\} \qquad x, b \in \mathbf{R}^2$$

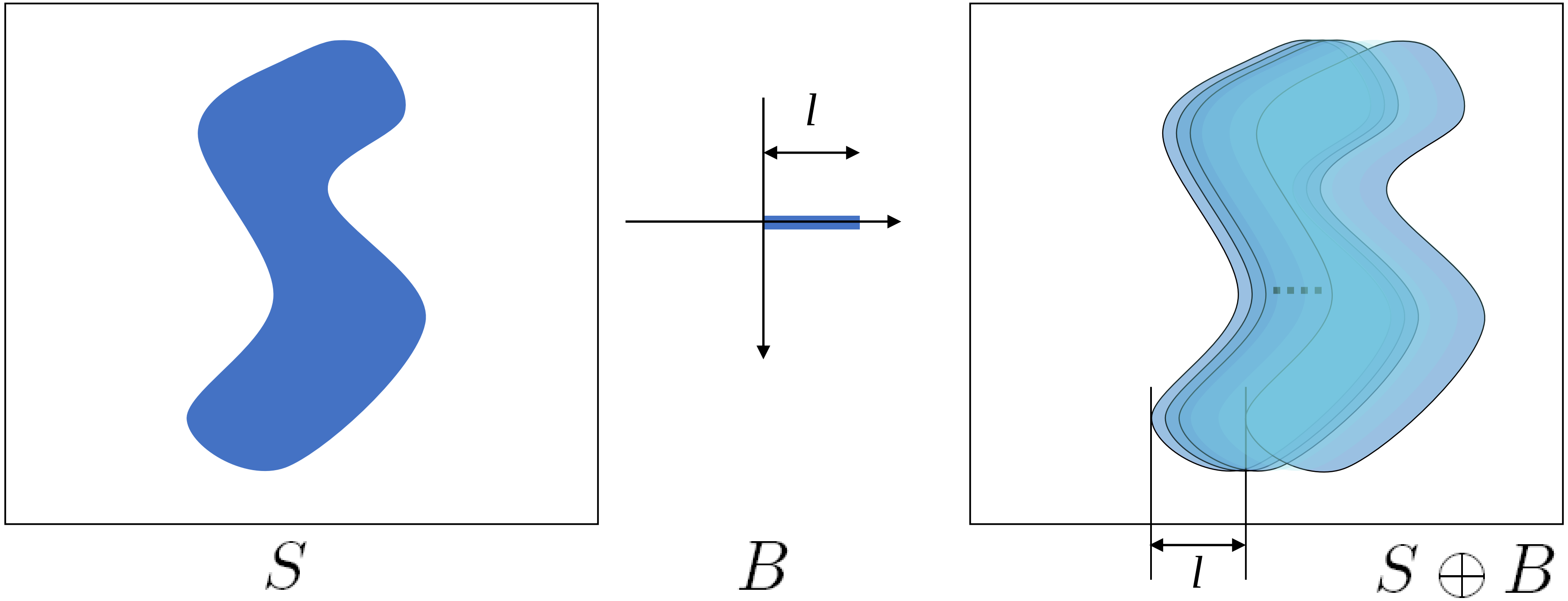
$$\{x \mid B_x \cap X \neq \emptyset\}$$

2) Better : from Minkowski's sum of sets $\quad X \oplus \check{B}$

Binary Morphology: Dilate

Minkowski's sum of sets :

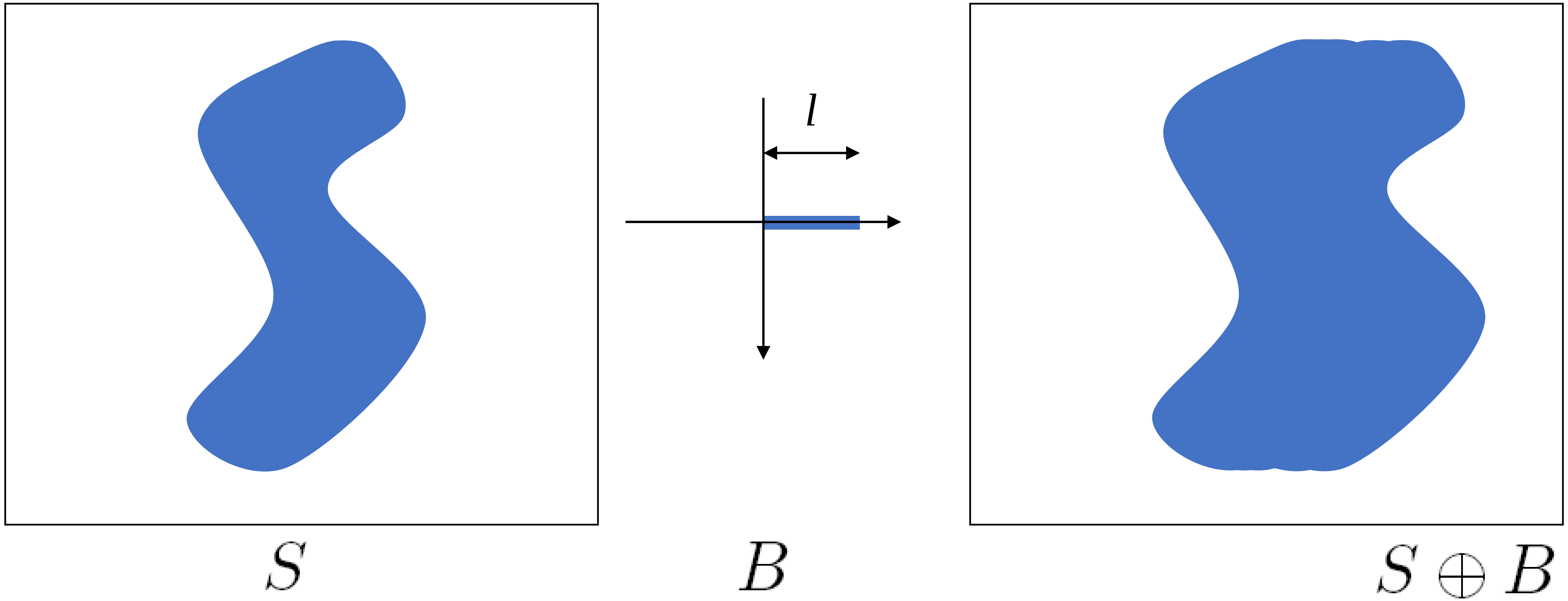
$$S \oplus B \stackrel{def}{=} \{s + b \mid s \in S, b \in B\} = \bigcup_{b \in B} S_b$$



Binary Morphology: Dilate

Minkowski's sum of sets :

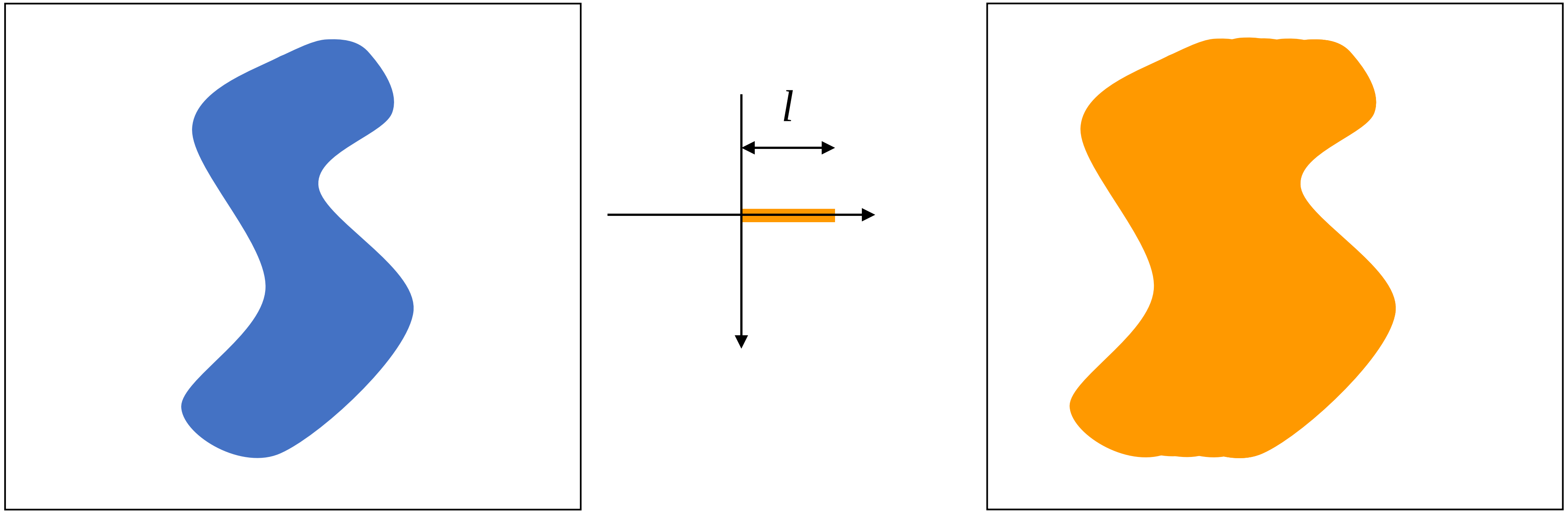
$$S \oplus B \stackrel{def}{=} \{s + b \mid s \in S, b \in B\} = \bigcup_{b \in B} S_b$$



Binary Morphology: Dilate

Dilation : $B_x \stackrel{def}{=} \{b + x \mid b \in B\} \quad x, b \in \mathbf{R}^2$

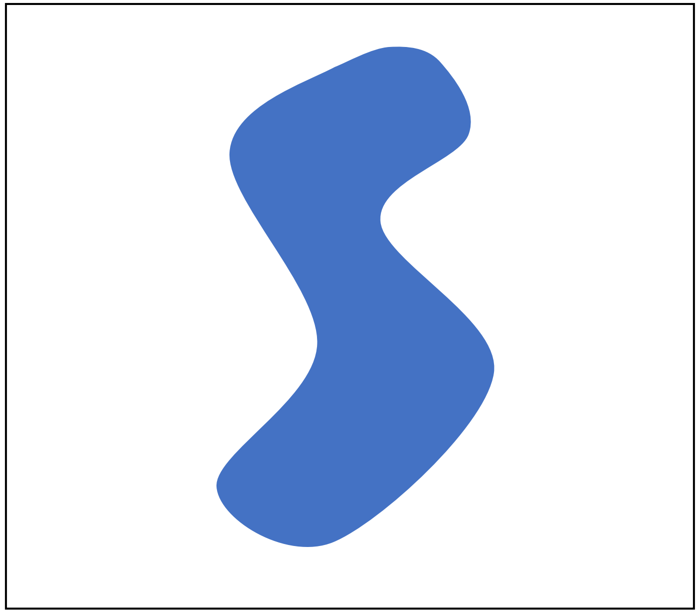
$\{x \mid B_x \cap X \neq \emptyset\}$



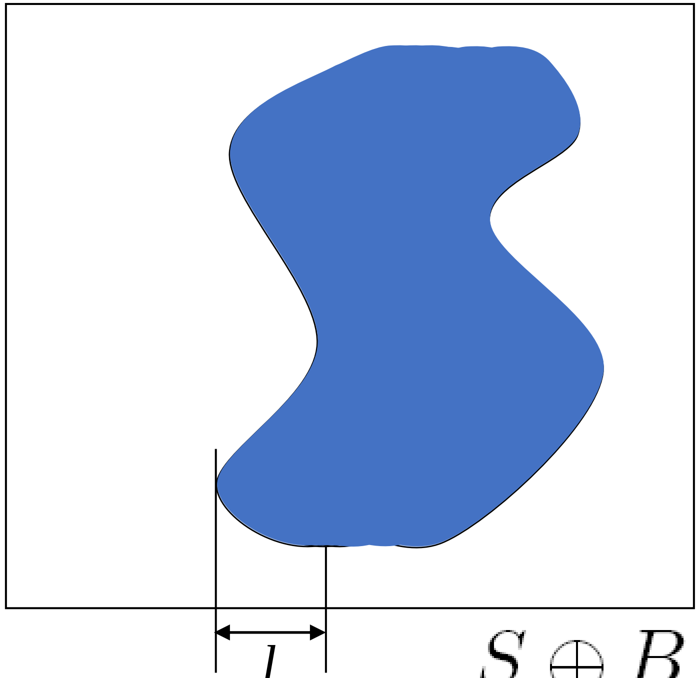
S B Dilation

Binary Morphology: Dilate

Dilation is *not* the Minkowski's sum



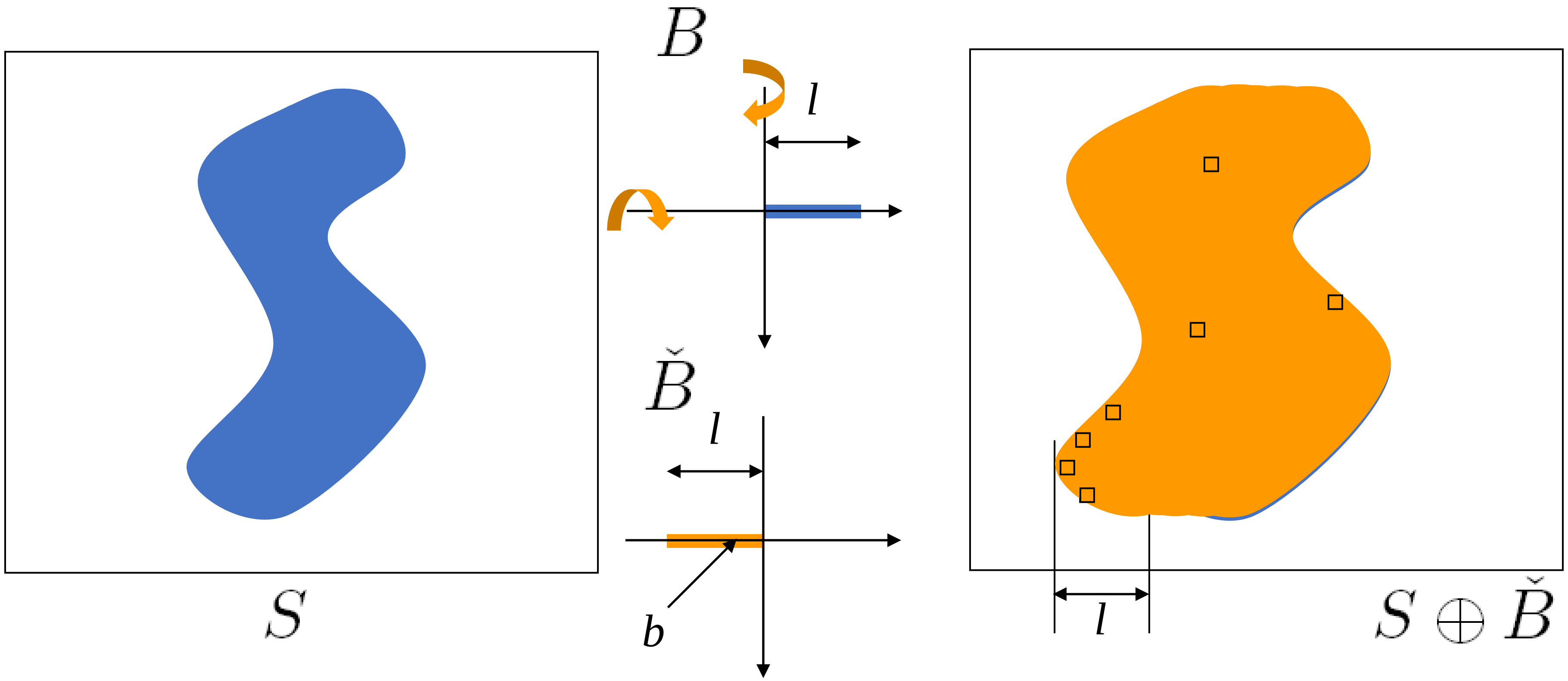
S



$S \oplus B$

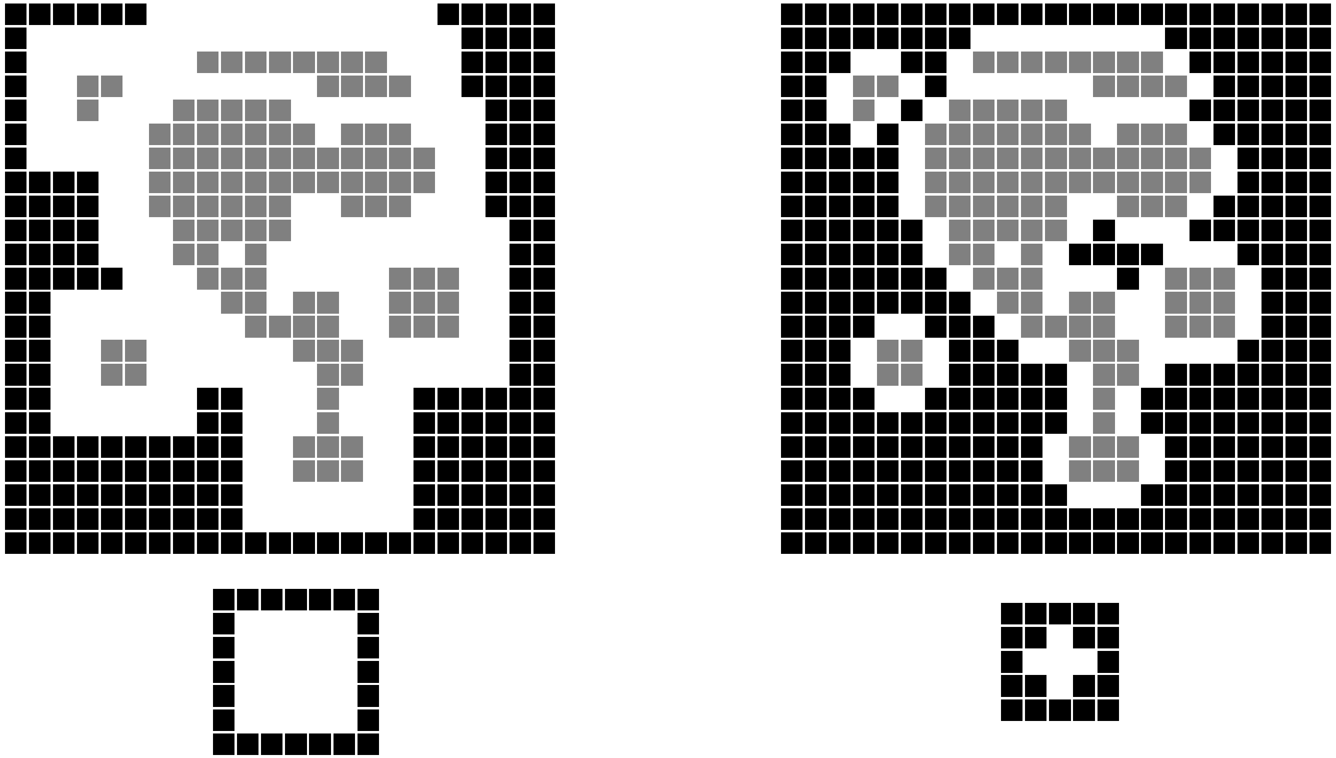
Binary Morphology: Dilate

$$\{x \mid B_x \cap X \neq \emptyset\} = \bigcup_{b \in \check{B}} X_b = X \oplus \check{B}$$



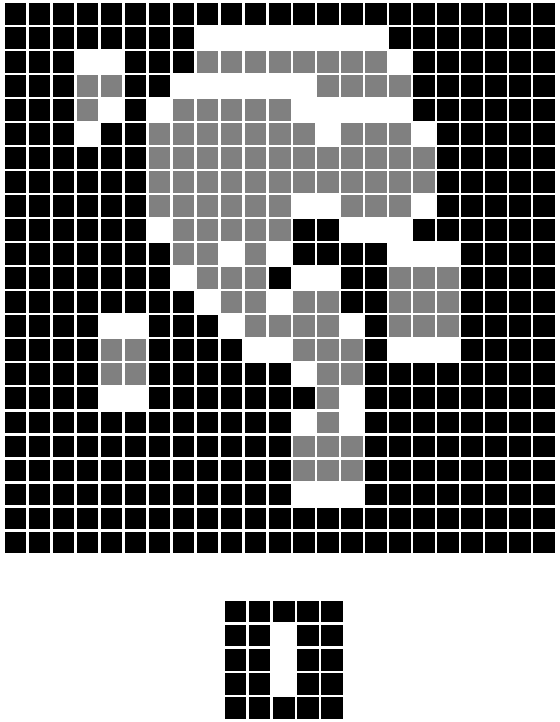
Binary Morphology: Dilate

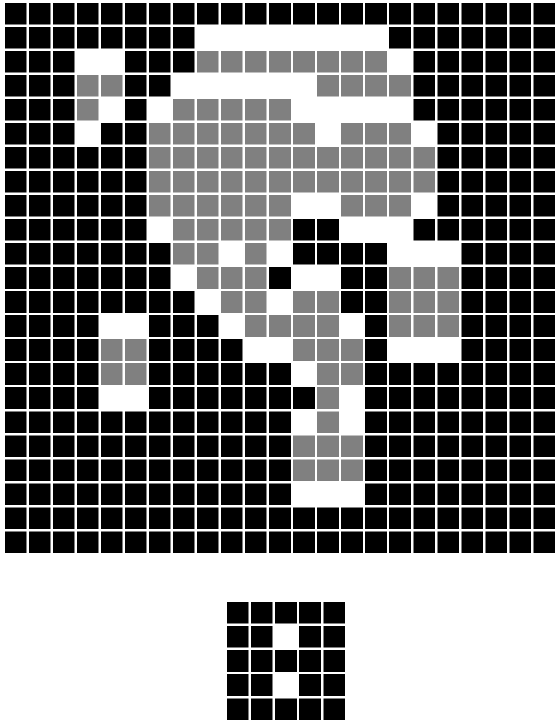
Dilation with other structuring elements



Binary Morphology: Dilate

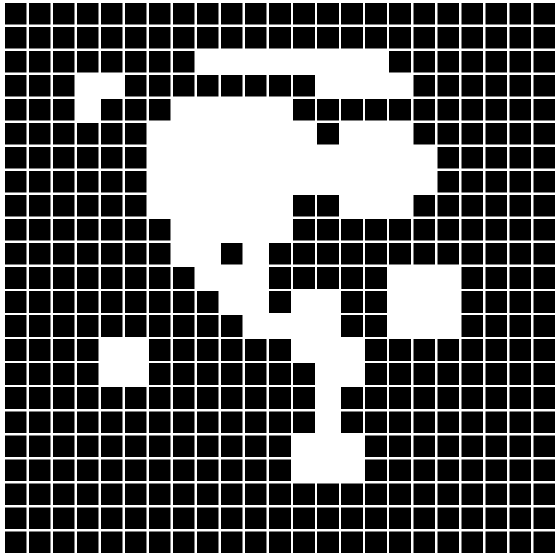
Dilation with other structuring elements

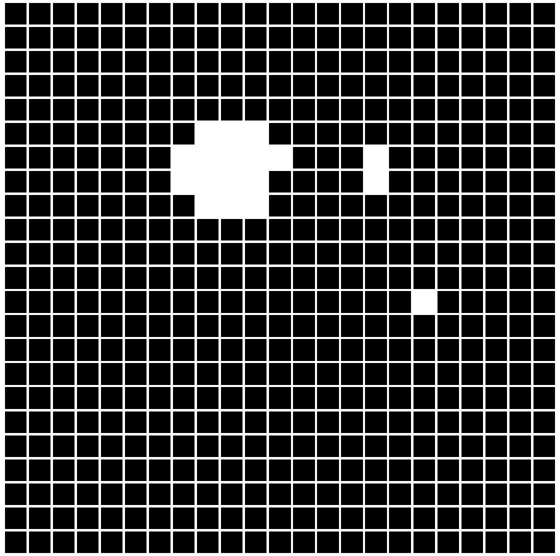


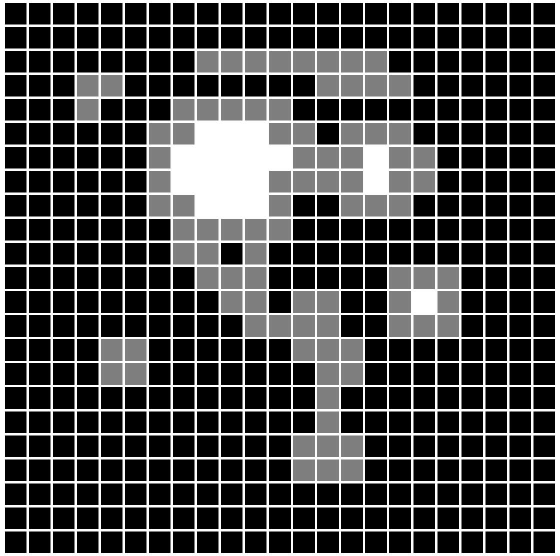


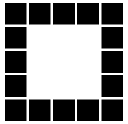
Binary Morphology: Erode

Erosion : $x = (x_1,x_2)$ such that if we center B on them,
then the so translated B is contained in X.






difference



Binary Morphology: Erode

Erosion : $x = (x_1,x_2)$ such that if we center *B* on them,
then the so translated *B* is contained in *X*.

How to formulate this definition ?

1) Literal translation

$$\{x \mid B_x \subseteq X\}$$

2) Better : from Minkowski's substraction of sets

$$X \ominus \check{B}$$

Binary Morphology: Erode

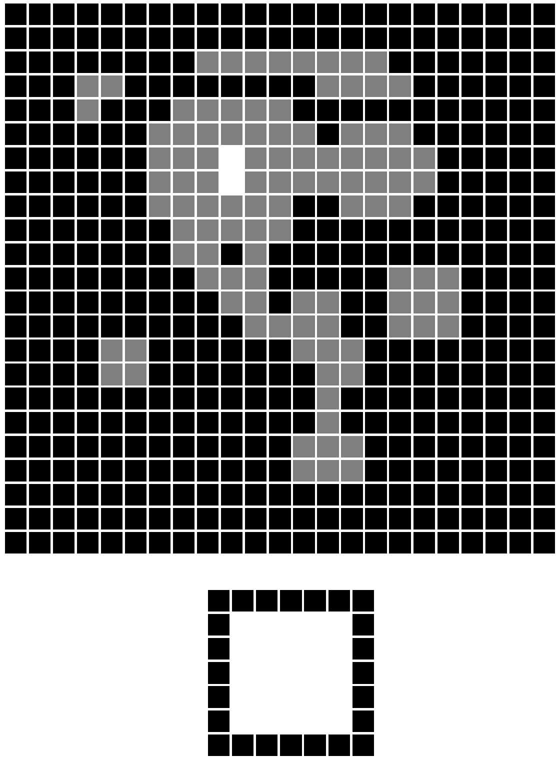
$$S \ominus B \stackrel{def}{=} \{y \mid y - b \in X, \;\; \forall b \in B\} = \bigcap_{b \in B} S_b$$

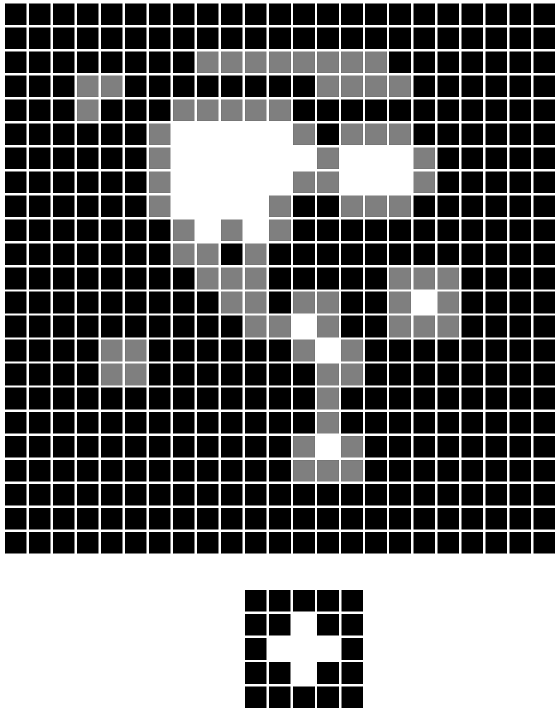
$$\{x \mid B_x \subseteq X\} = \bigcap_{b \in \check{B}} X_b$$

$$\bigcap_{b \in \check{B}} X_b = \{y \mid y - b \in X, \;\; \forall b \in \check{B}\} = X \ominus \check{B}$$

Binary Morphology: Erode

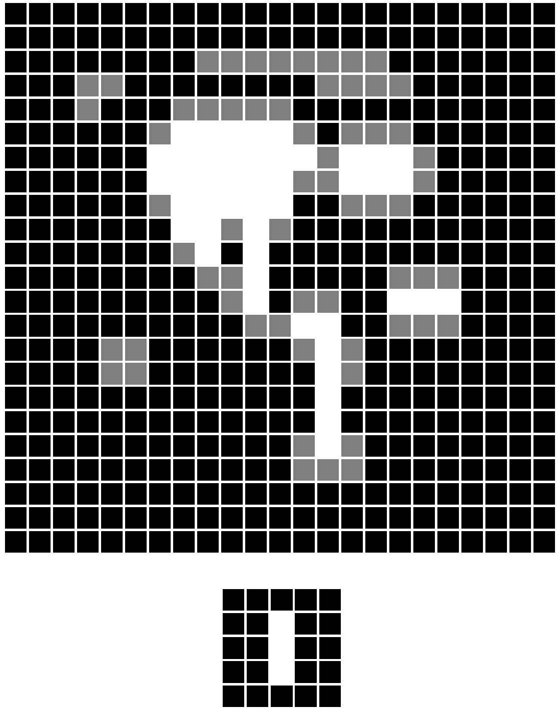
Erosion with other structuring elements

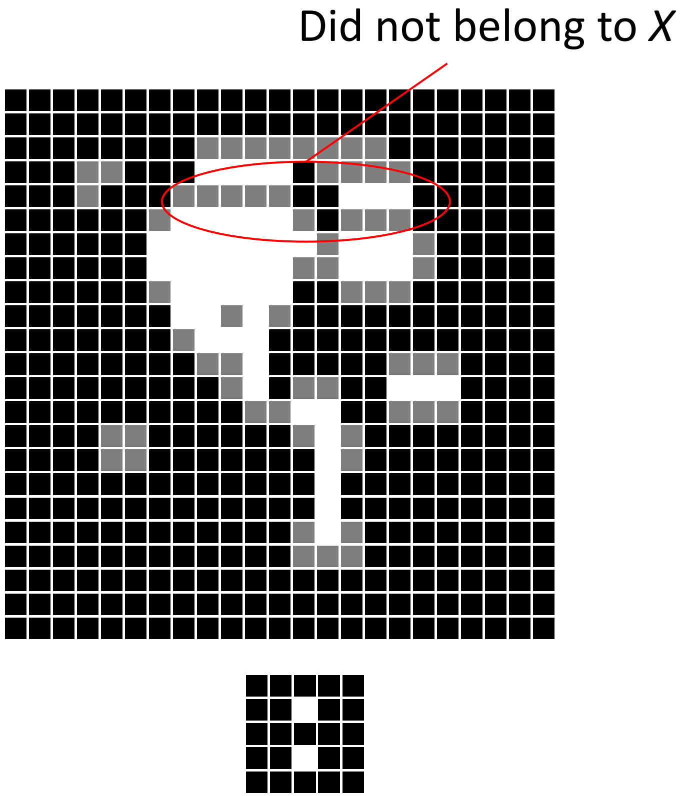




Binary Morphology: Erode

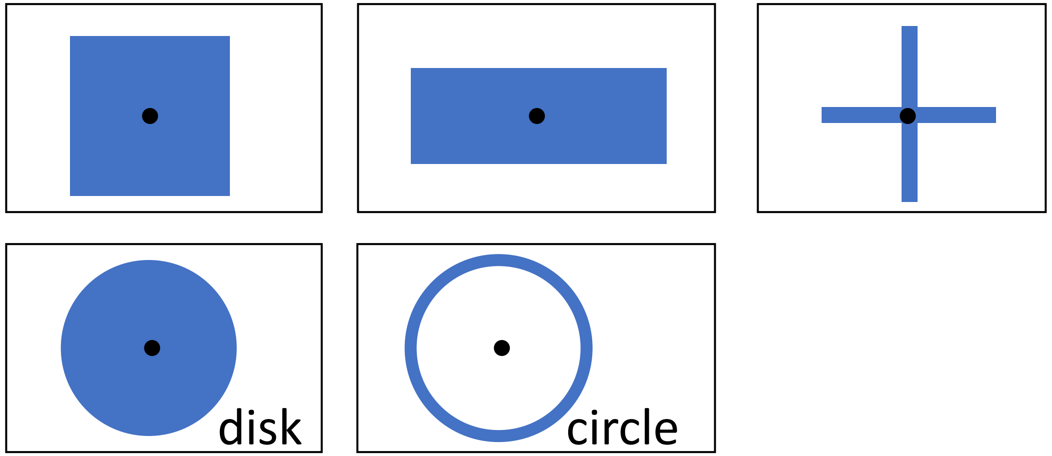
Erosion with other structuring elements

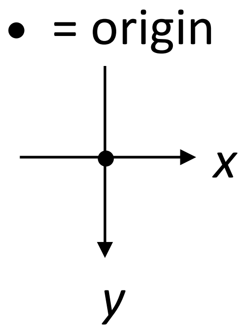


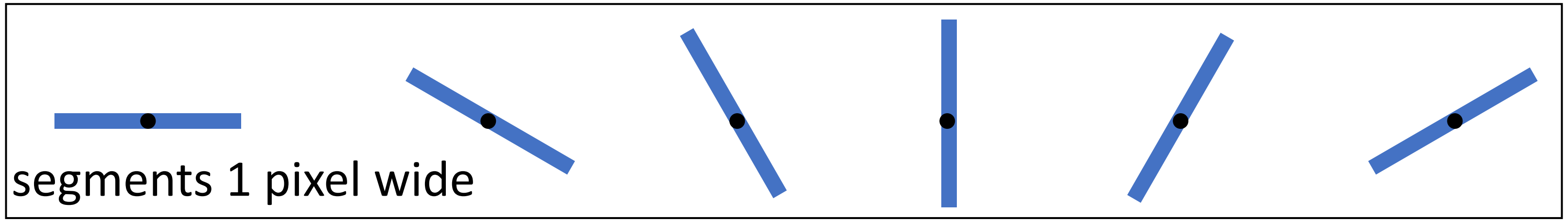


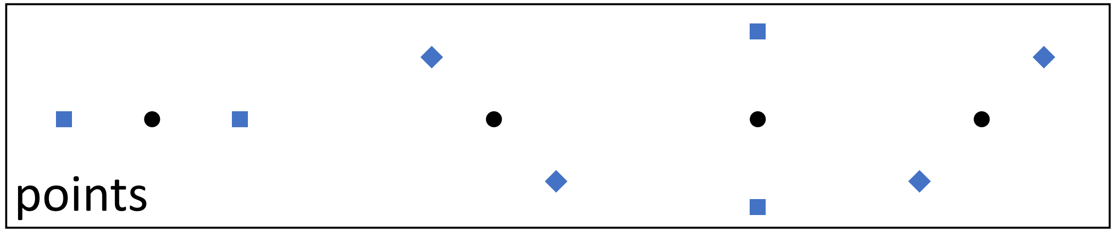
Binary Morphology: Structuring Elements

Common structuring elements shapes







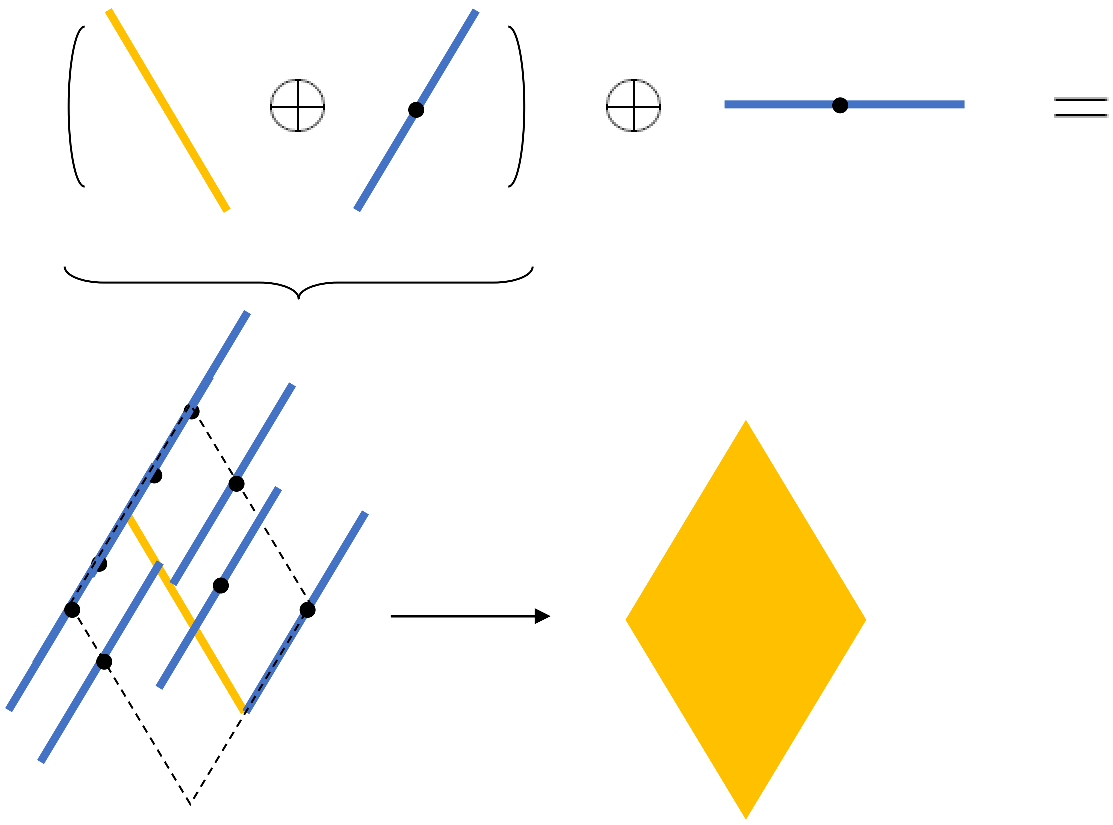


Note :

$$B = \check{B}$$

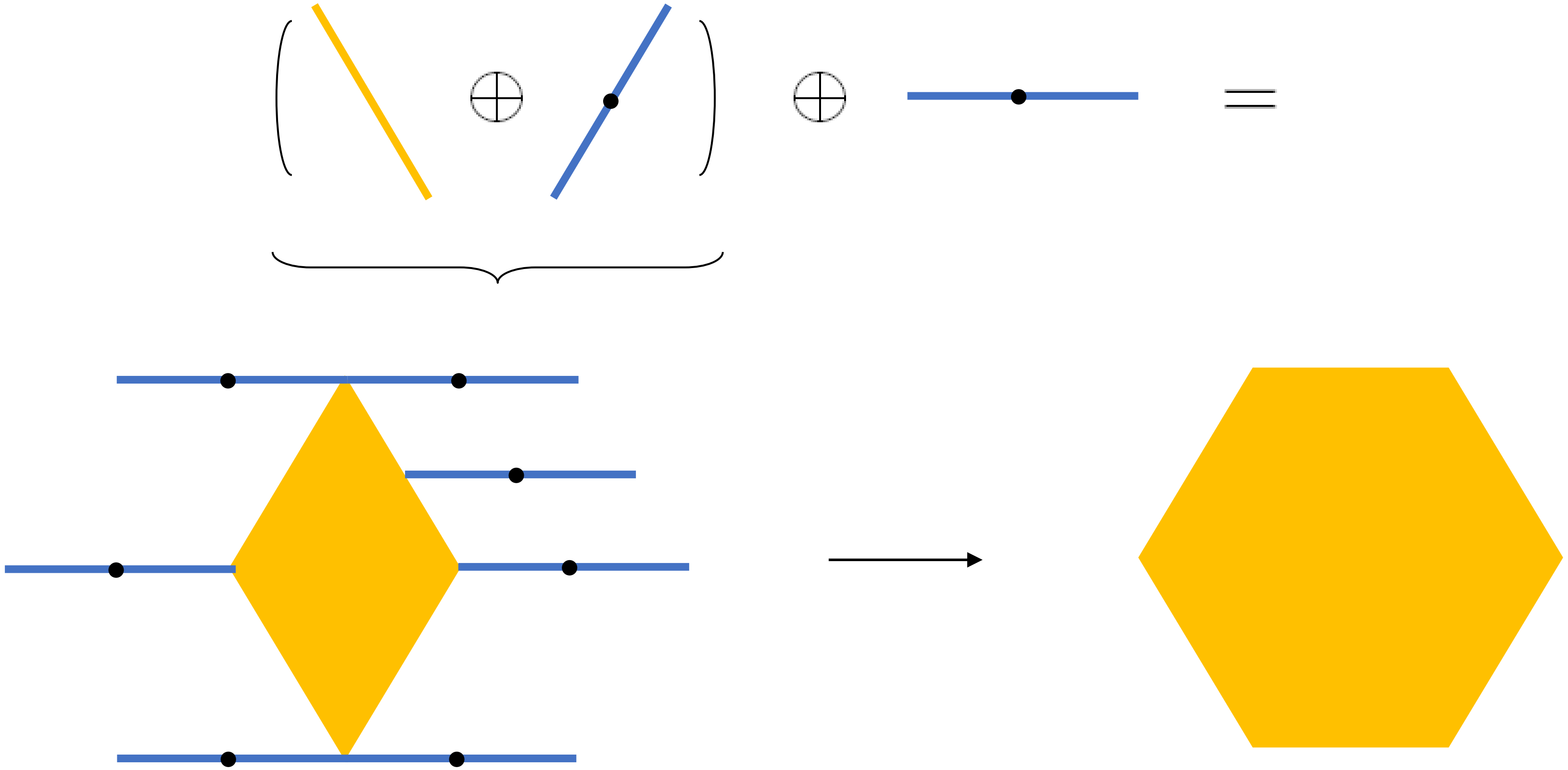
Binary Morphology

Problem :



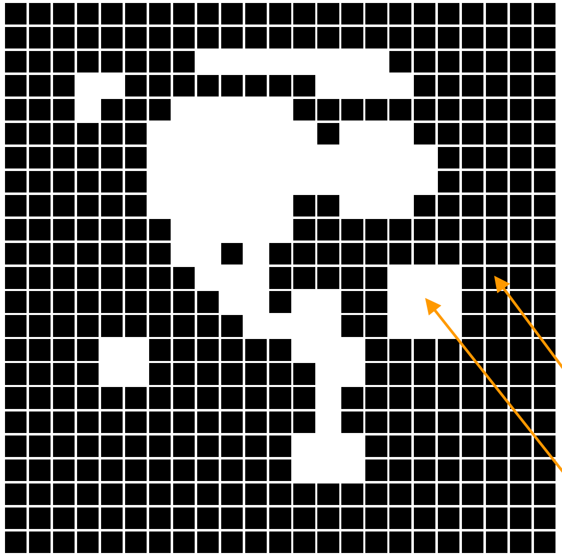
Binary Morphology

Problem :

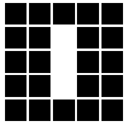


Binary Morphology

Implementation : very low computational cost



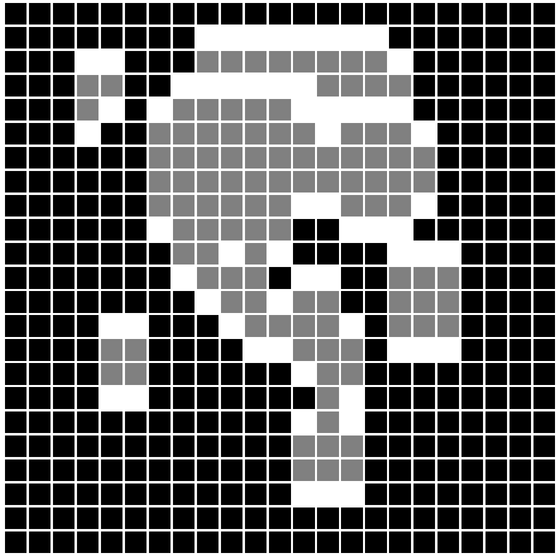
X



B

0

1 (or >0)



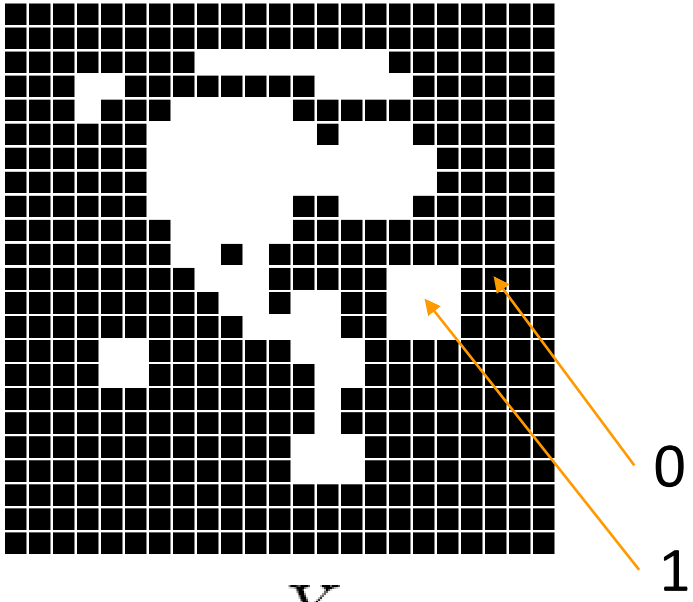
$X \oplus \check{B}$

$$
\begin{aligned}
(X \oplus \check{B})_{i,j} &= \bigvee \left(X_{i,j-1}, X_{i,j}, X_{i,j+1} \right) \\
&= \bigvee \left(X_{(i,j)+b, b \in \check{B}} \right)
\end{aligned}
$$

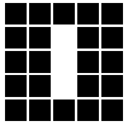
Logical *or*

Binary Morphology

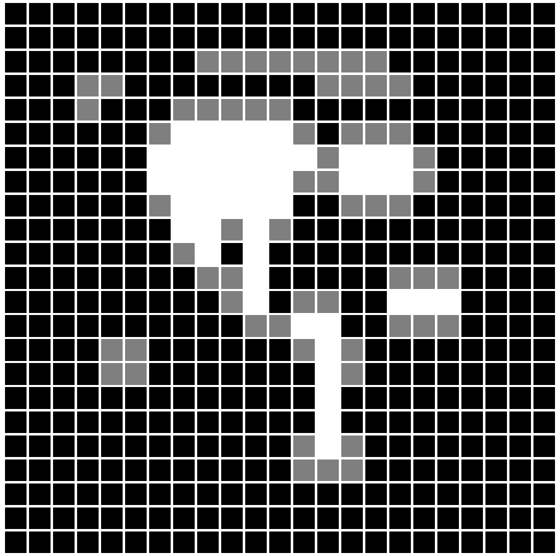
Implementation : very low computational cost



X



B



$X \ominus \check{B}$

$$(X \ominus \check{B})_{i,j} = \bigwedge (X_{i,j-1}, X_{i,j}, X_{i,j+1})$$

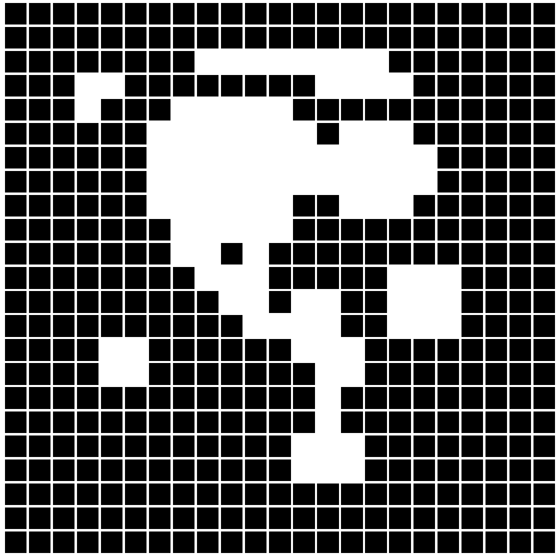
Logical *and*

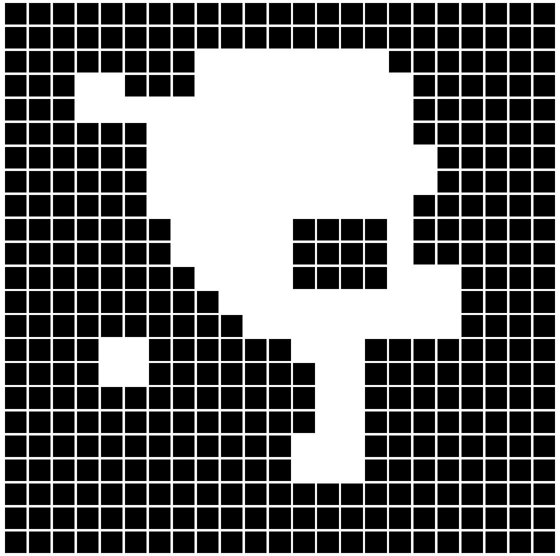
$$= \bigwedge (X_{(i,j)+b,\in \check{B}})$$

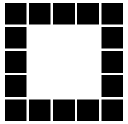
Binary Morphology: Opening

Opening : $X \bullet B \stackrel{def}{=} \Big(\bigcup_{B_x \subseteq X^c} B_x \Big)^c = (X \oplus \check{B}) \ominus \check{B}$

also X^B







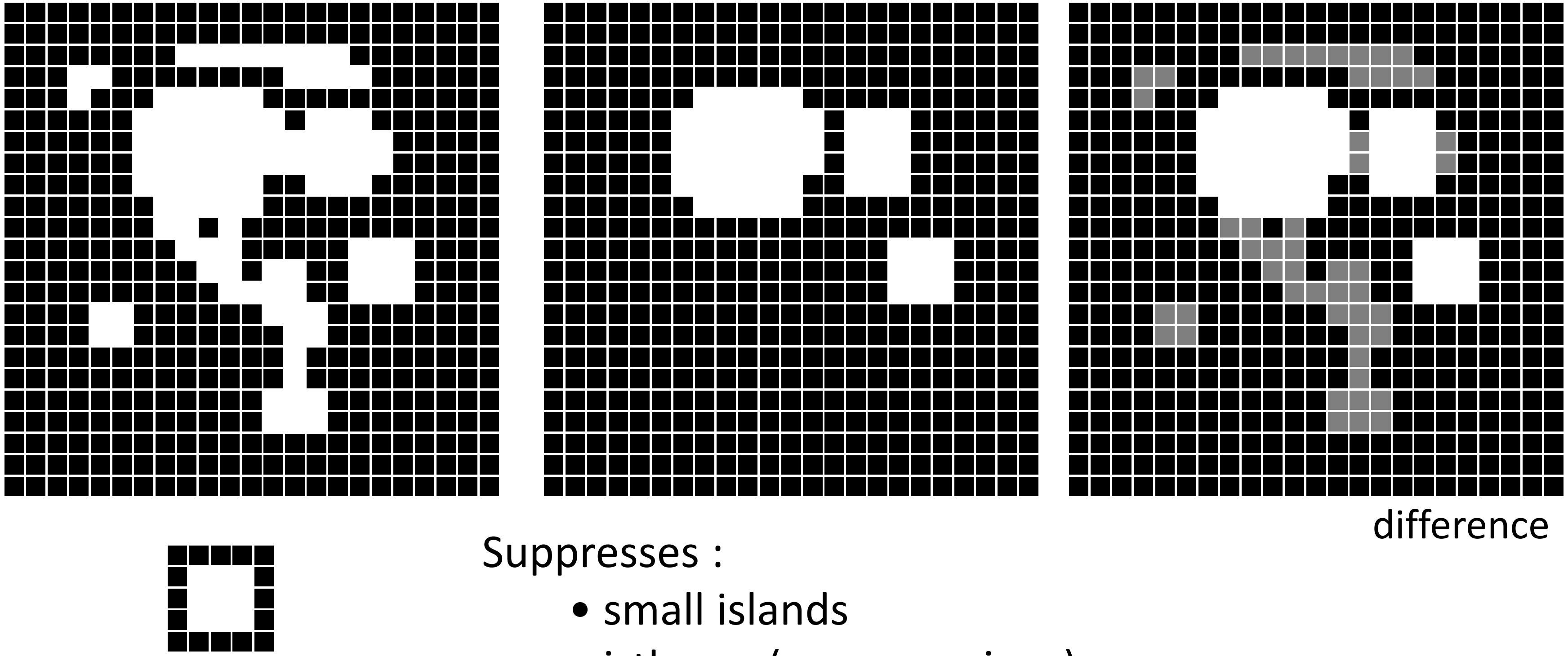
Suppresses :

- small lakes (holes)
- channels (narrow separations)
- narrow bays

Binary Morphology: Open

Opening :

also X_B



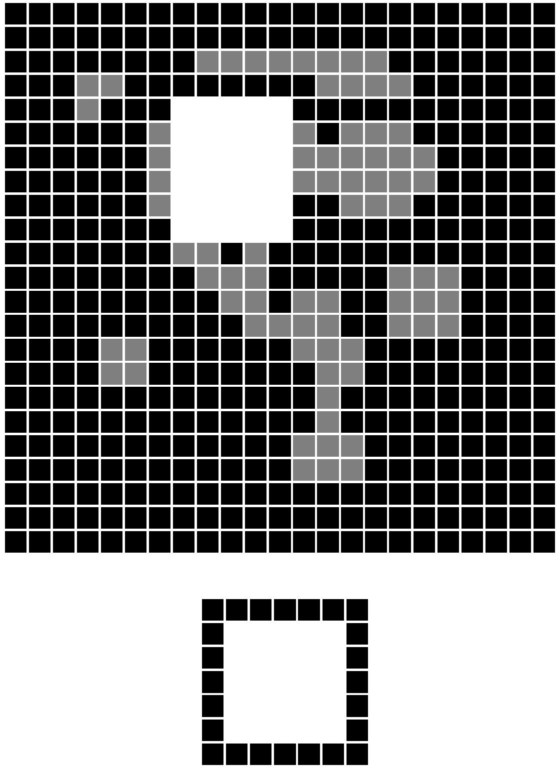
difference

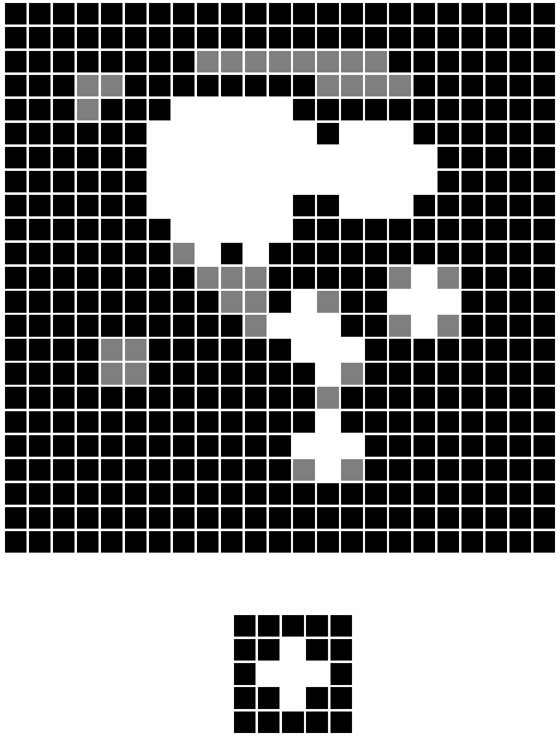
Suppresses :

- small islands
- isthmus (narrow unions)
- narrow caps

Binary Morphology: Open

Opening with other structuring elements

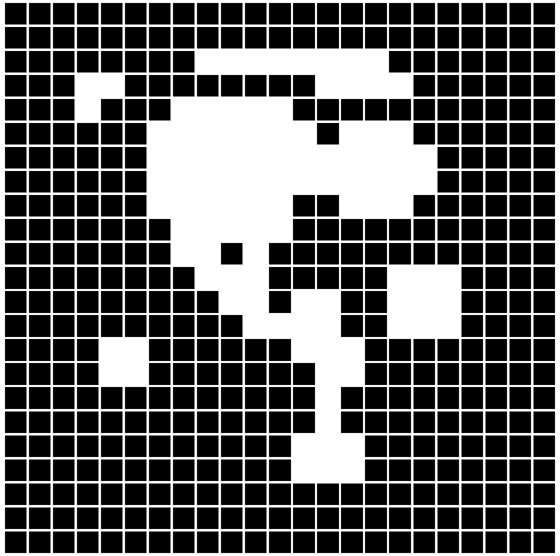


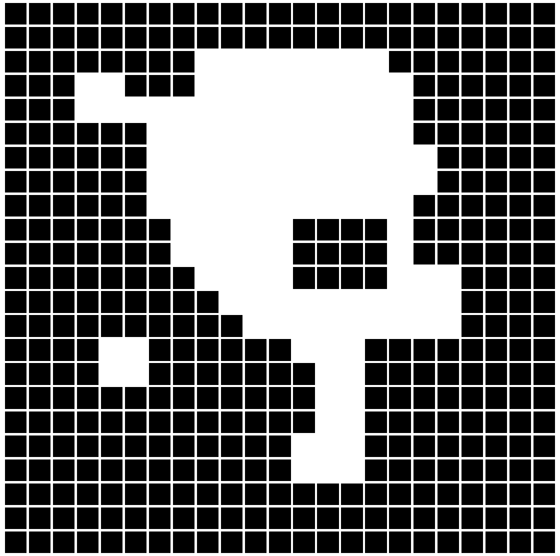


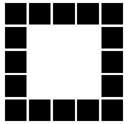
Binary Morphology: Close

Closing : $X \bullet B \stackrel{def}{=} \Big(\bigcup_{B_x \subseteq X^c} B_x \Big)^c = (X \oplus \check{B}) \ominus \check{B}$

also X^B







Suppresses :

- small lakes (holes)
- channels (narrow separations)
- narrow bays

Binary Morphology: Application

Application: shape smoothing and noise filtering

X

Binary Morphology: Application

Application: shape smoothing and noise filtering

$$X \ominus \check{B}$$

Binary Morphology: Application

Application: shape smoothing and noise filtering

X

$X \ominus \check{B}$ ✖

$X \circ B$

$(X \circ B) \oplus \check{B}$

Binary Morphology: Application

Application: shape smoothing and noise filtering

X

$X \ominus \check{B}$ ✖

$X \circ B$

$(X \circ B) \oplus \check{B}$ ✖

$(X \circ B) \bullet B$

Binary Morphology: Properties

Properties

- all of them are *increasing* :

$$X \subseteq Y \Longrightarrow \Psi(X) \subseteq \Psi(Y)$$

- opening and closing are *idempotent* :

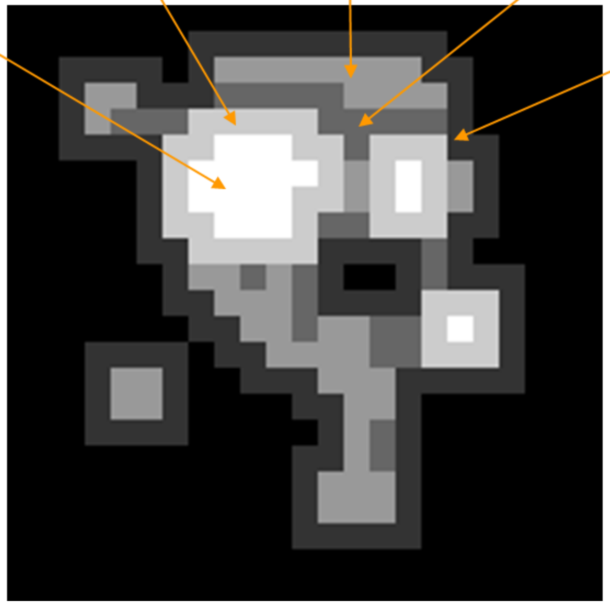
$$\Psi(X) = \Psi(\Psi(X))$$

Binary Morphology: Properties

dilation and closing are *extensive*
erosion and opening are *anti-extensive* :

$$(0,0) \in B \implies$$

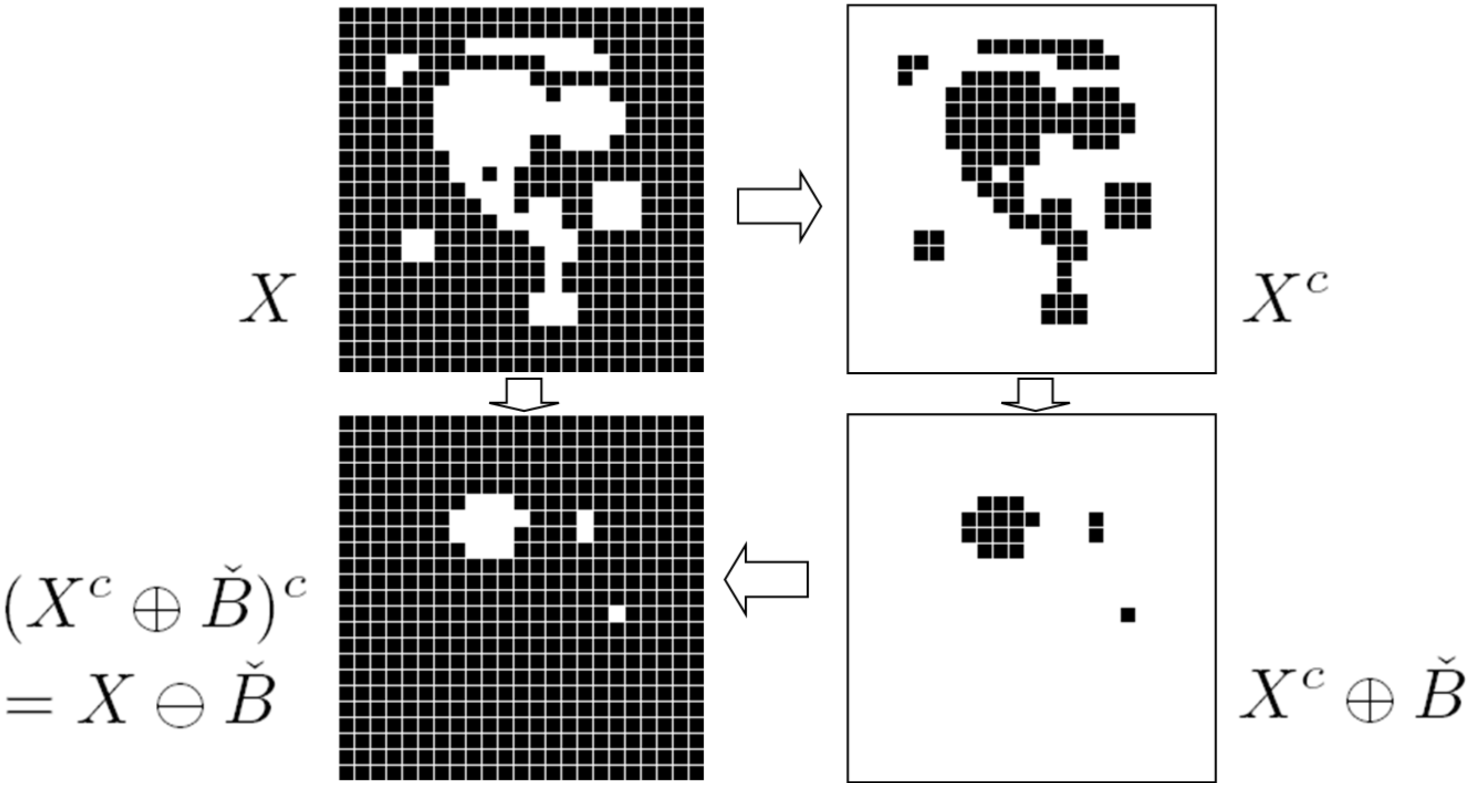
$$\boxed{X \ominus \check{B}} \subseteq \boxed{X \circ B} \subseteq \boxed{X} \subseteq \boxed{X \bullet B} \subseteq \boxed{X \oplus \check{B}}$$



Binary Morphology: Properties

- *duality* of erosion-dilation, opening-closing,...

$$\Psi, \Phi \ \text{ duals} \quad \Psi(X) = [\Phi(X^c)]^c$$



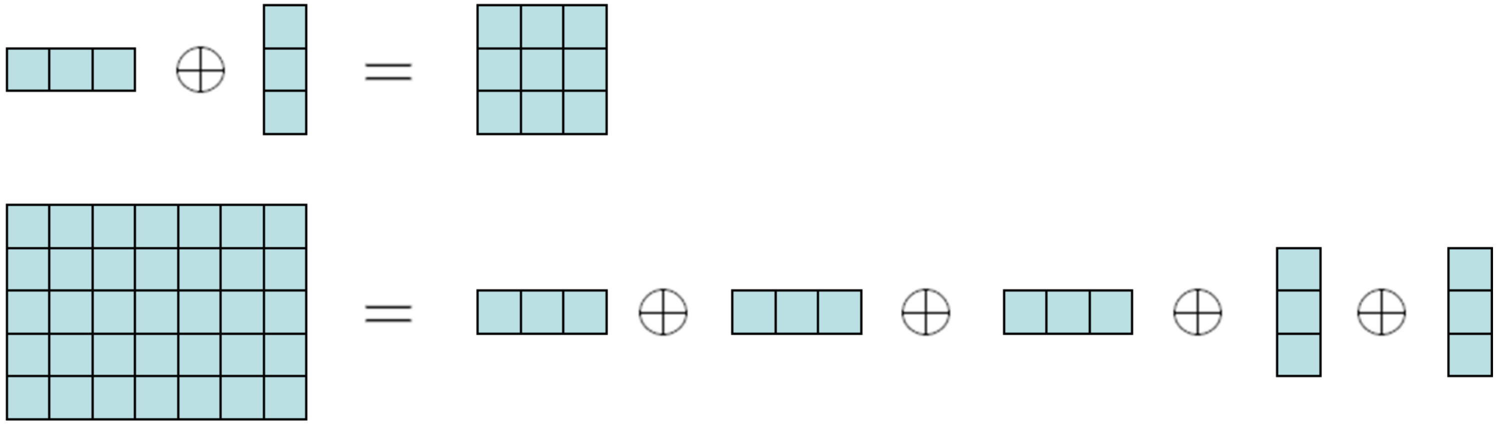
Binary Morphology: Properties

- structuring elements decomposition

$$X \oplus (B_1 \oplus \check{B}_2) = (X \oplus \check{B}_1) \oplus \check{B}_2$$

$$X \ominus (B_1 \oplus \check{B}_2) = (X \ominus \check{B}_1) \ominus \check{B}_2$$

operations with big structuring elements can be done
by a succession of operations with small s.e's

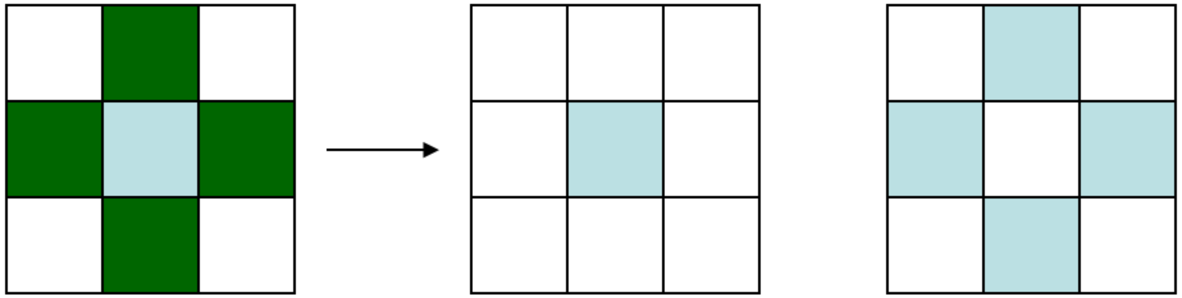


Binary Morphology: Hit-or-miss

Hit-or-miss : $$X \otimes B = (X \ominus \check{B}_1) \bigcap (X^c \ominus \check{B}_2)$$

$$B = (B_1, B_2)$$

Bi-phase structuring element



B

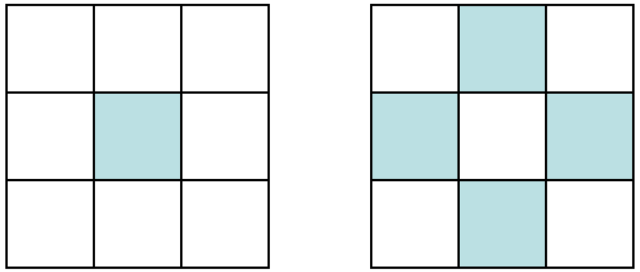
B_1
"Hit" part
(white)

B_2
"Miss" part
(black)

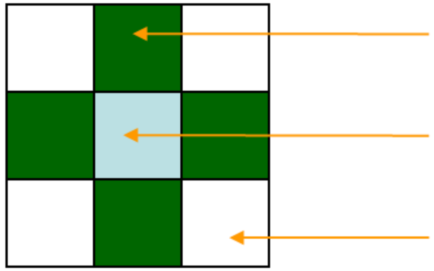
Binary Morphology: Hit-or-miss

Looks for pixel configurations :

$$\{x \mid (B_1)_x \subseteq X, (B_2)_x \subseteq X^c\}$$

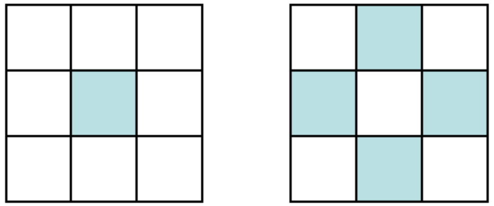


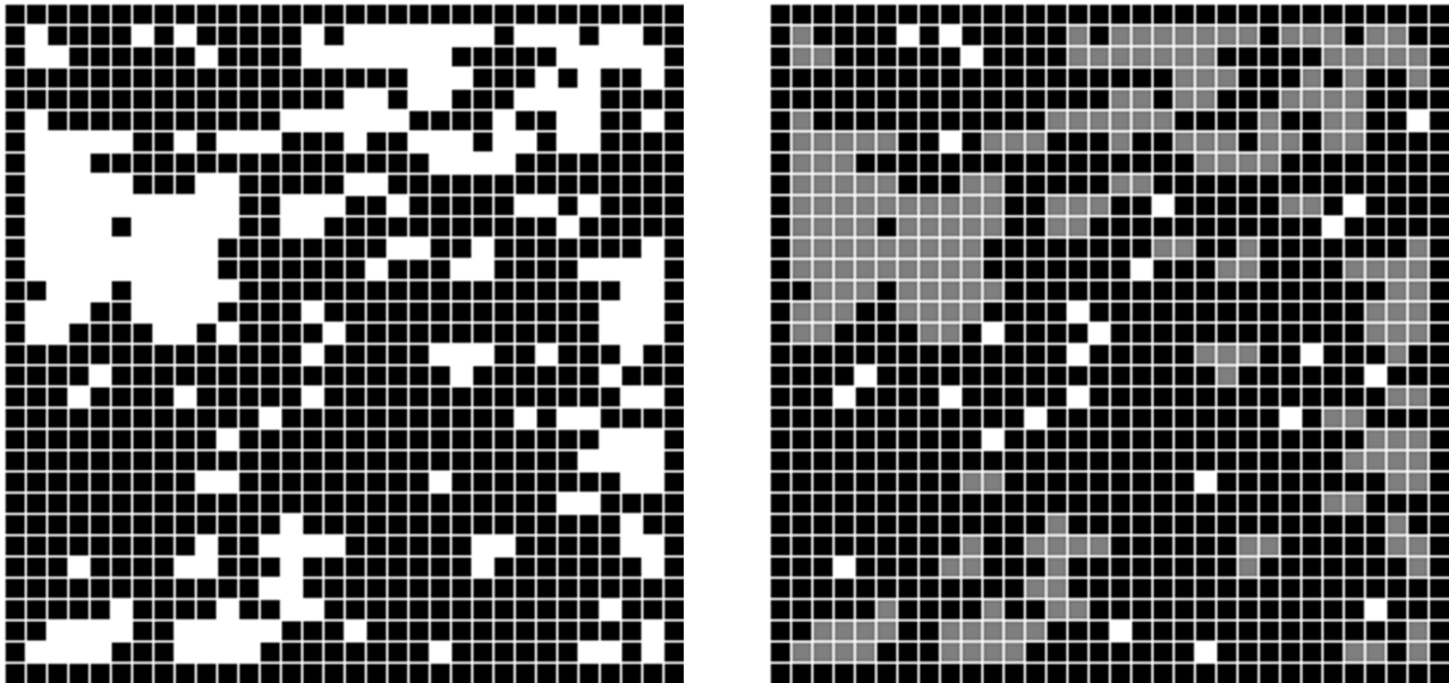
B_1 $\qquad$ B_2



Binary Morphology: Hit-or-miss

isolated points at
4 connectivity





Binary Morphology: Hit-or-miss

Thinning : $$X \bigcirc B = X \setminus (X \otimes B)$$

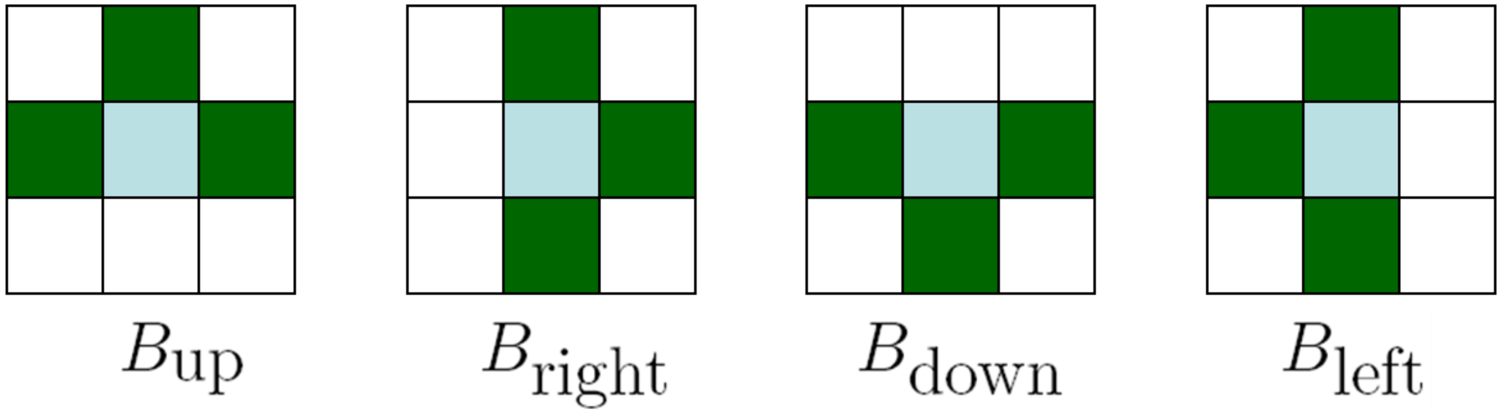
Thickenning : $$X \odot B = X \cup (X \otimes B)$$

Depending on the structuring elements (actually, series of them), very different results can be achieved :

- Prunning
- Skeletons
- Zone of influence
- Convex hull
- ...

Binary Morphology: Hit-or-miss

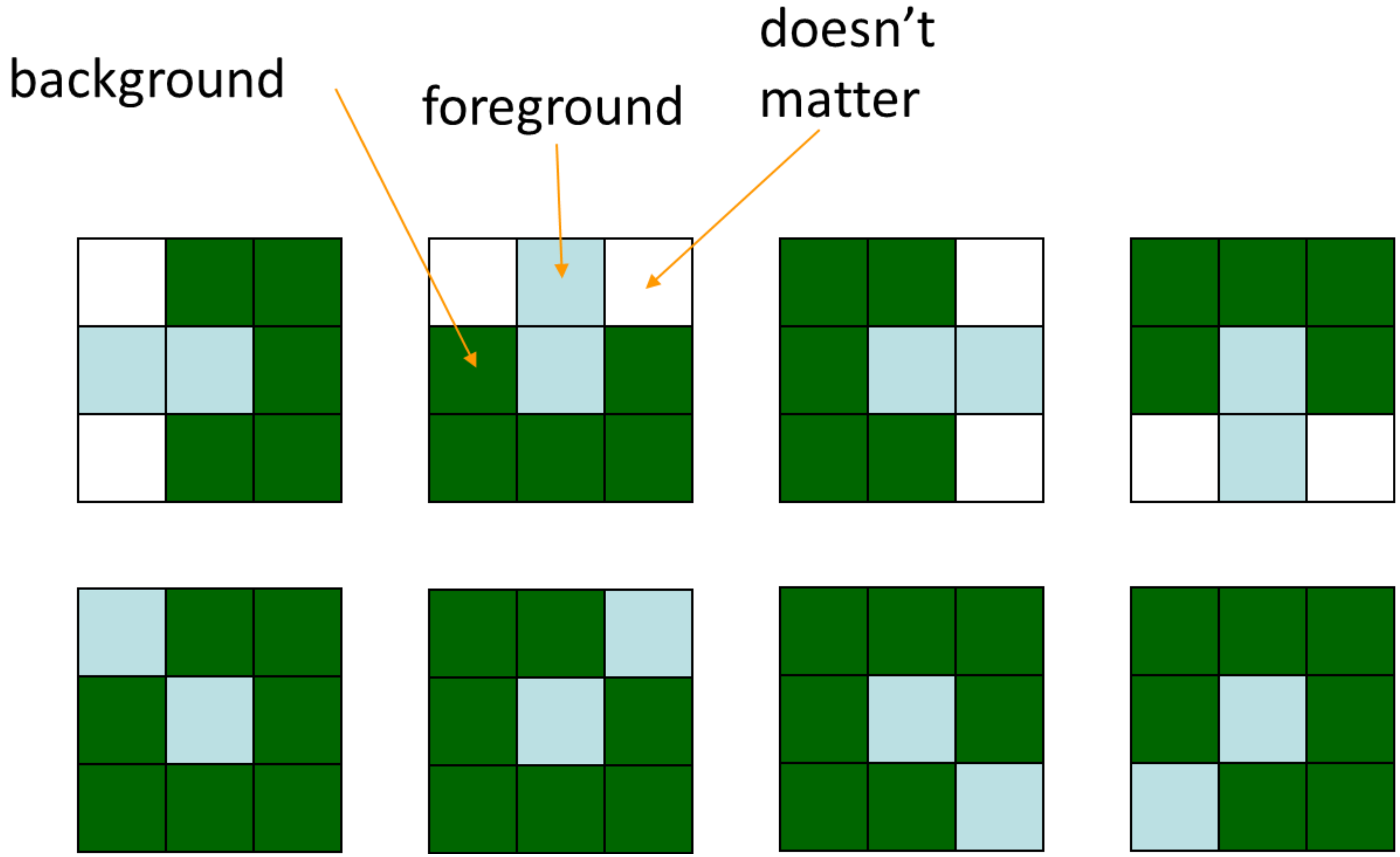
Prunning at 4 connectivity : remove end points by a *sequence* of thinnings



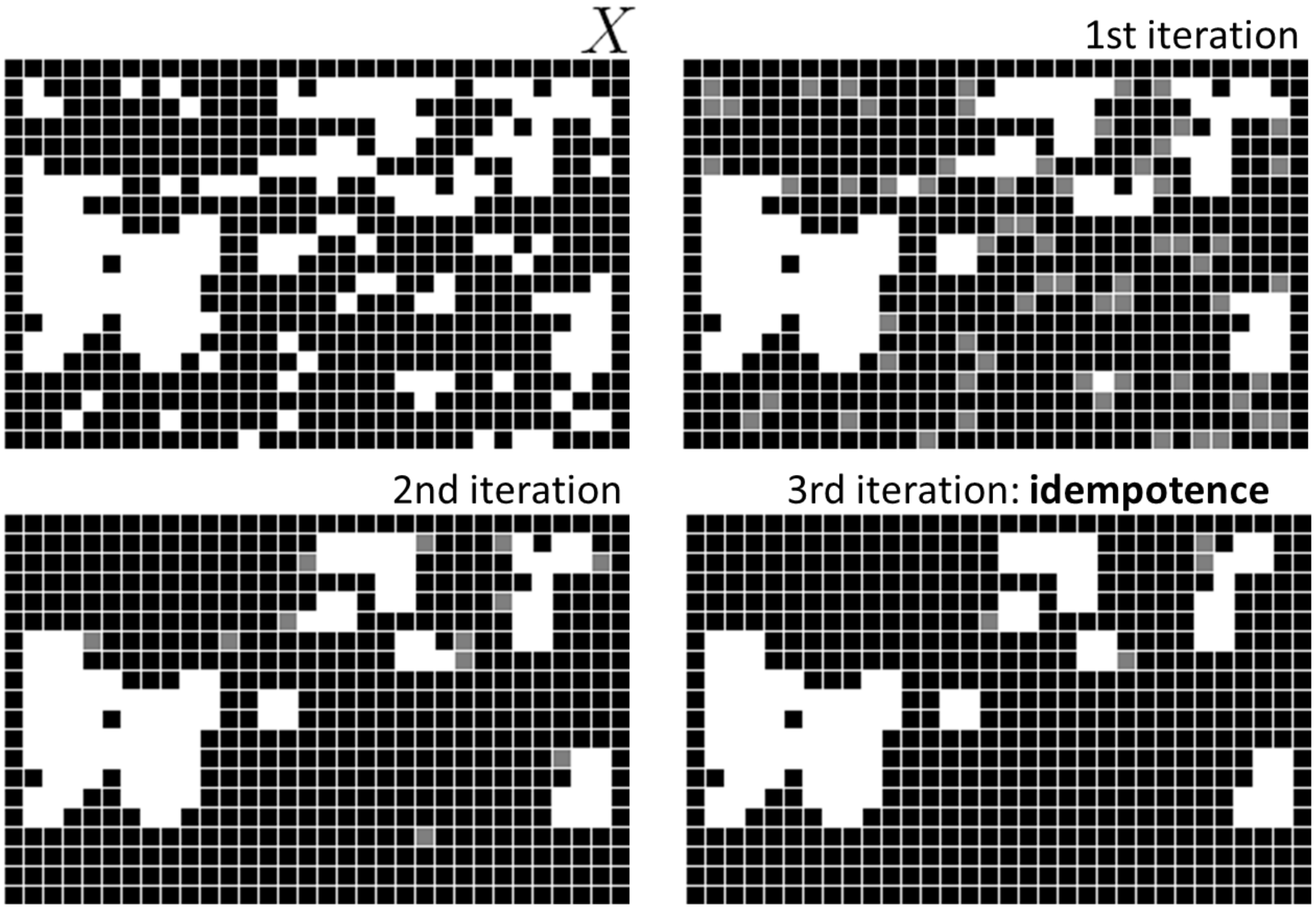
1 iteration = $(((X \bigcirc B_{\text{up}}) \bigcirc B_{\text{right}}) \bigcirc B_{\text{down}}) \bigcirc B_{\text{left}}$

Binary Morphology: Hit-or-miss

What does the following sequence ?



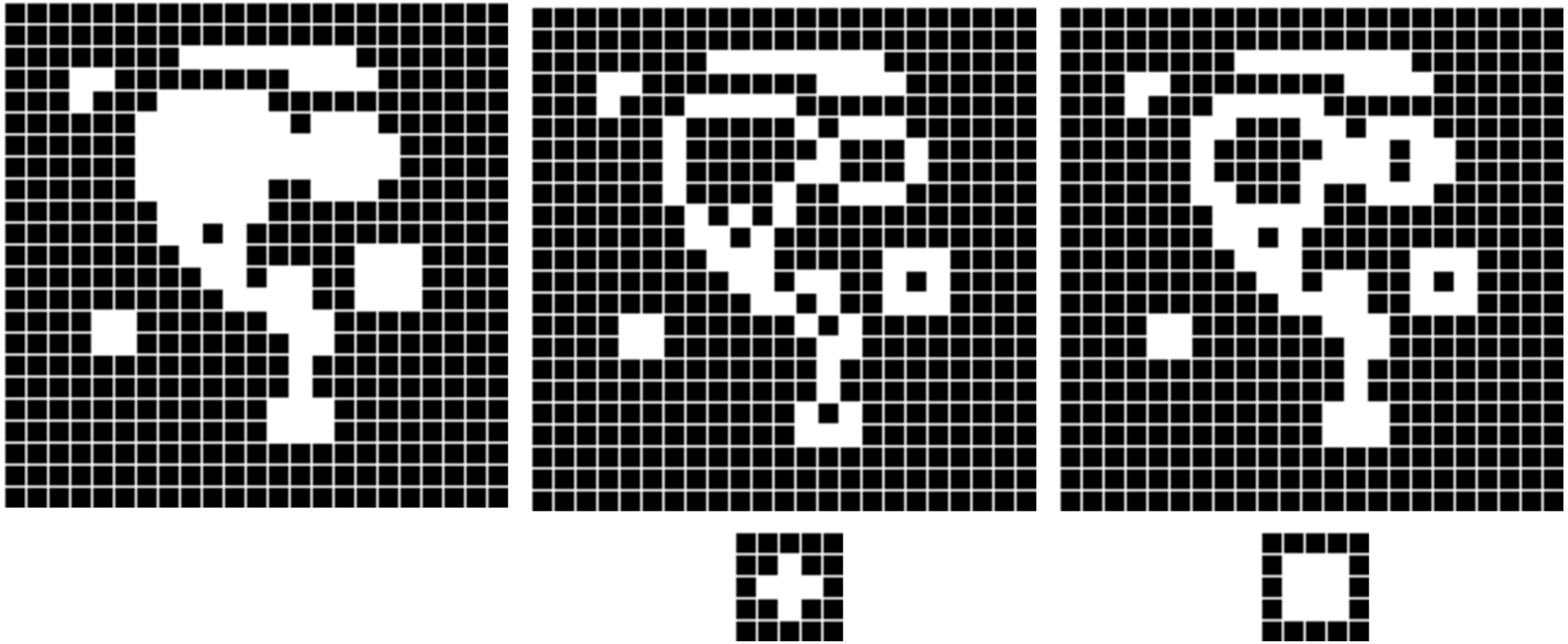
Binary Morphology: Hit-or-miss



Other useful transformations

Contours

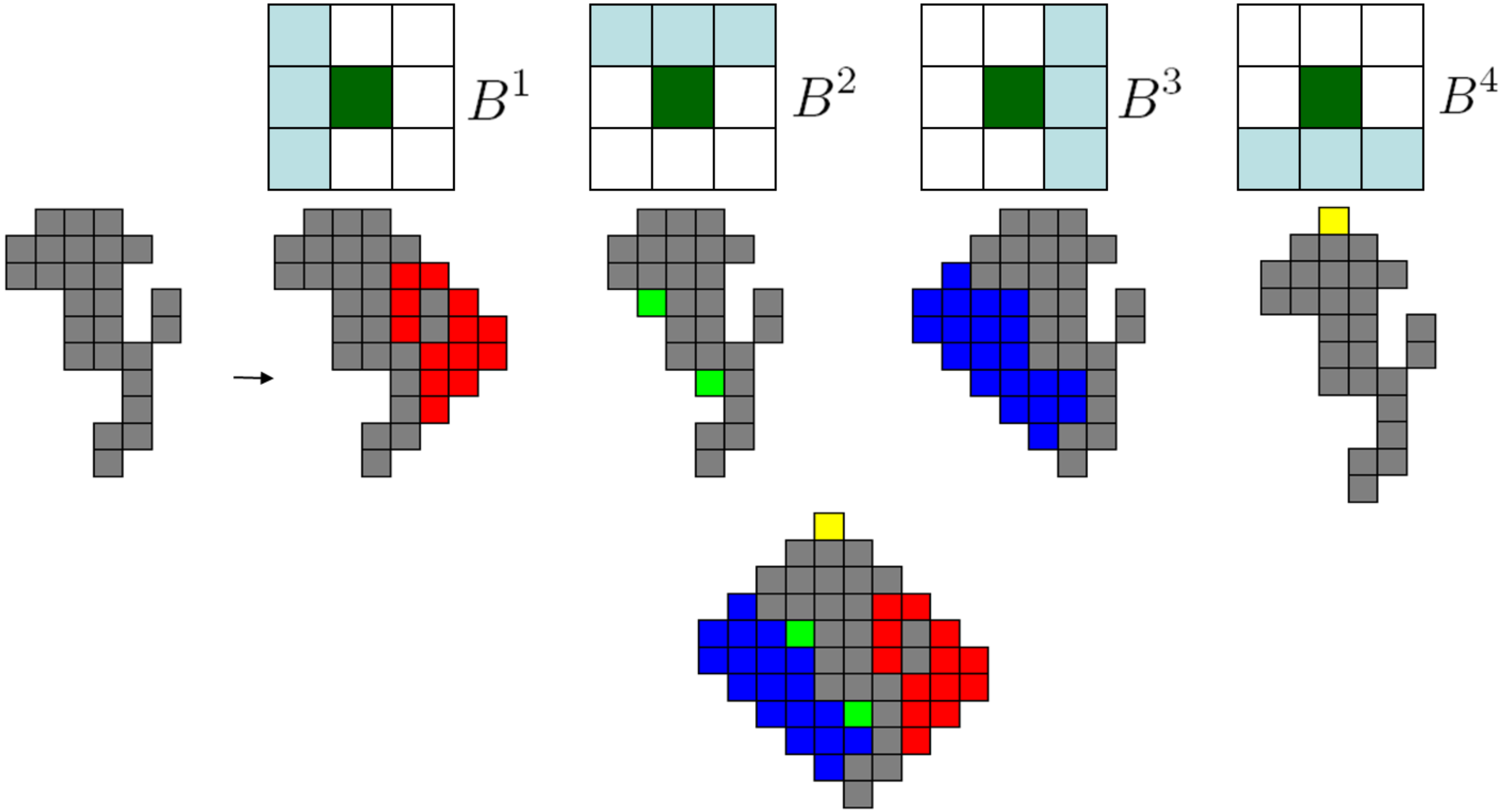
$$X \setminus X \ominus \check{B}$$



Other useful transformations

Convex Hull

union of thickenings, each up to idempotence

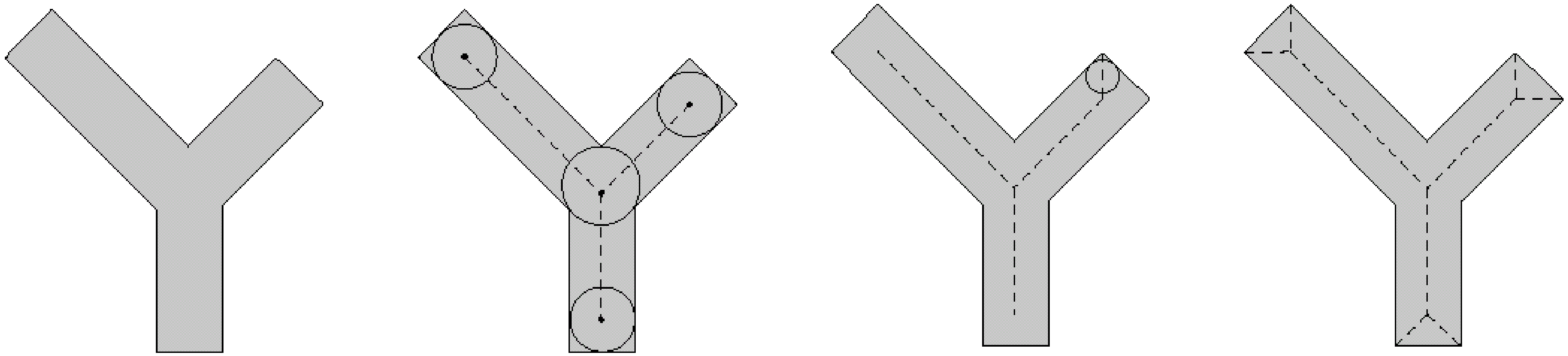


Other useful transformations

Skeleton

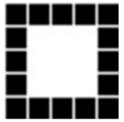
Maximal disk : disk centered at x, D_x, such that $D_x \subseteq X$ and no other D_y contains it .

Skeleton : union of centers of maximal disks.



Mathematical morphology grey level

f



B

$(f \oplus B)$

$$
\begin{aligned}
(f \oplus B)_{i,j} &= \max \left(f_{i-1,j-1},\ f_{i-1,j}, \ldots f_{i+1,j+1} \right) \\
&= \max \left(f_{i,j+b,\ b \in B} \right)
\end{aligned}
$$

Local maximum

Mathematical morphology grey level

f

B

$(f \ominus B)$

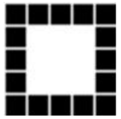
$$
\begin{aligned}
(f \ominus B)_{i,j} &= \min\left(\, f_{i-1,j-1},\ f_{i-1,j}, \ldots f_{i+1,j+1} \,\right) \\
&= \min\left(\, f_{i,j+b,\ b \in B} \,\right)
\end{aligned}
$$

Local minimum

Mathematical morphology grey level

Morphological gradient

f



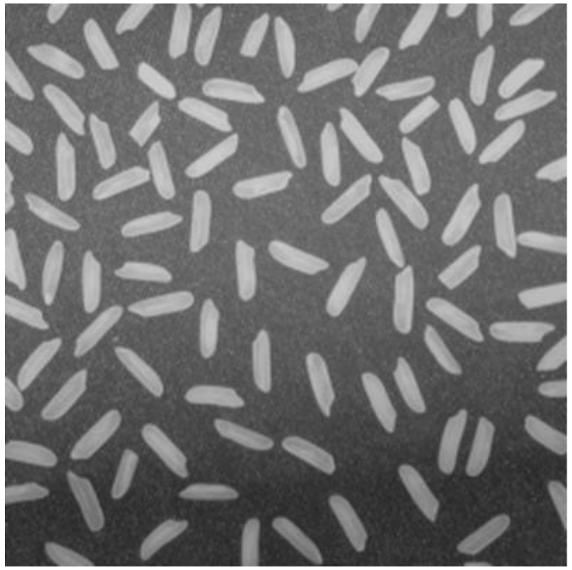
B

$\mathrm{grad}(f)$

$$\mathrm{grad}(f) = (f \oplus B) - (f \ominus B)$$

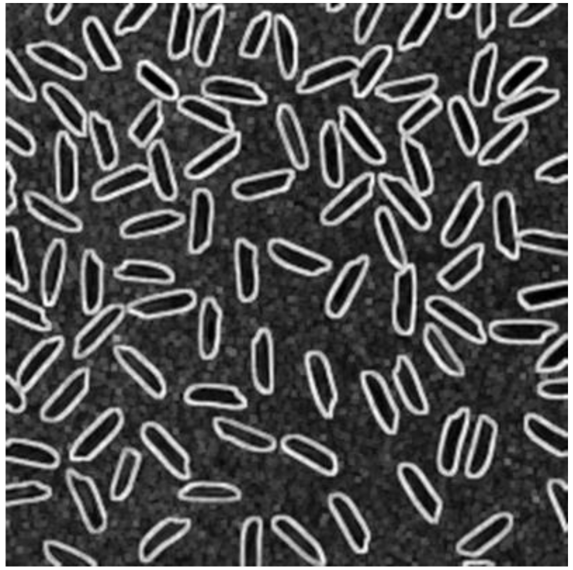
Mathematical morphology grey level

Morphological gradient



f

B



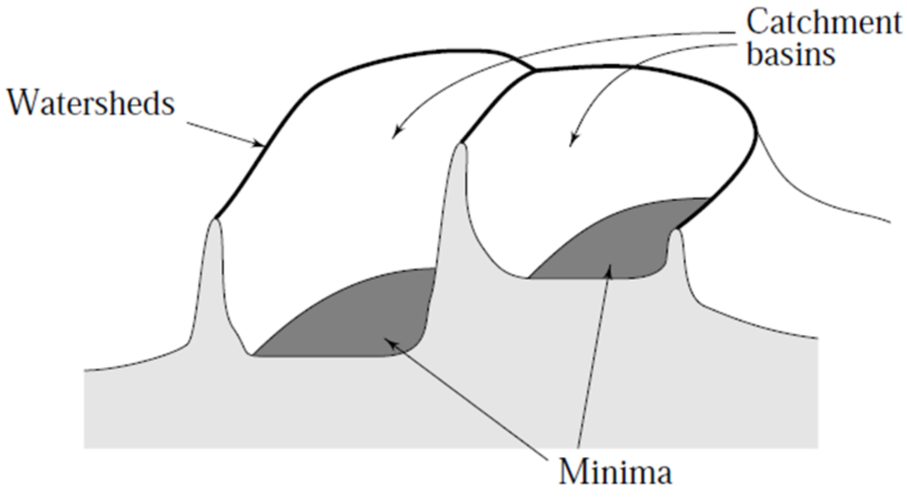
$\mathrm{grad}(f)$

$$\mathrm{grad}(f) = (f \oplus B) \; - \; (f \ominus B)$$

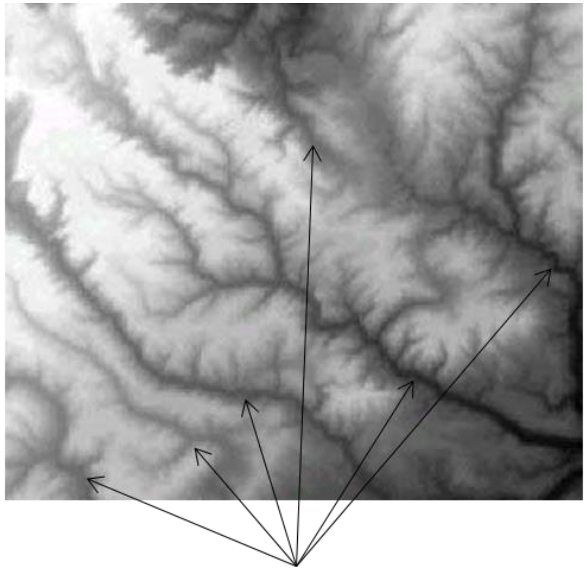
Mathematical morphology grey level: Watersheds

The **watersheds (lines)** of a grayscale image are the lines that separate the different *catchment basins*, if we look at the image as a topographic relief. Also, *divide lines*, *ligne de partage des eaux*. Interesting for *segmentation*.

The catchment basin associated with a (regional) minimum *m* of a topographic surface is the locus of the points *p* such that a drop falling at *p* slides along the surface until it reaches *m*.

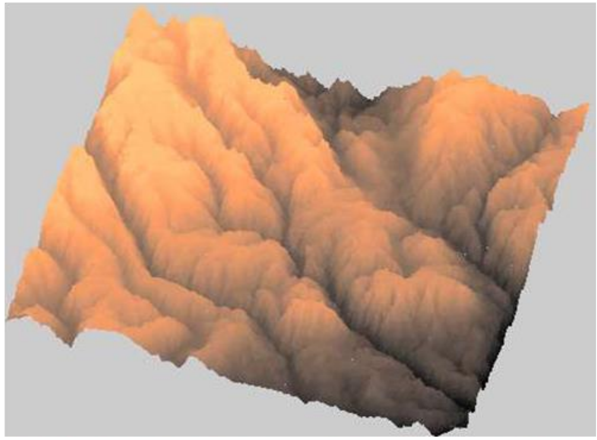


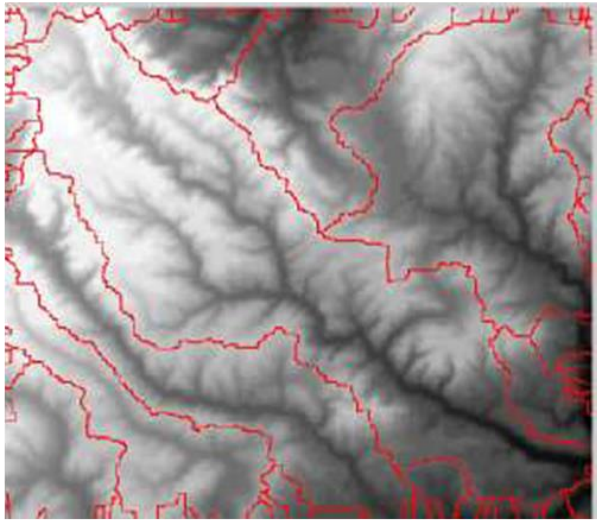
Mathematical morphology grey level: Watersheds



rivers

Digital elevation model (DEM): special image where grey level *is* terrain altitude





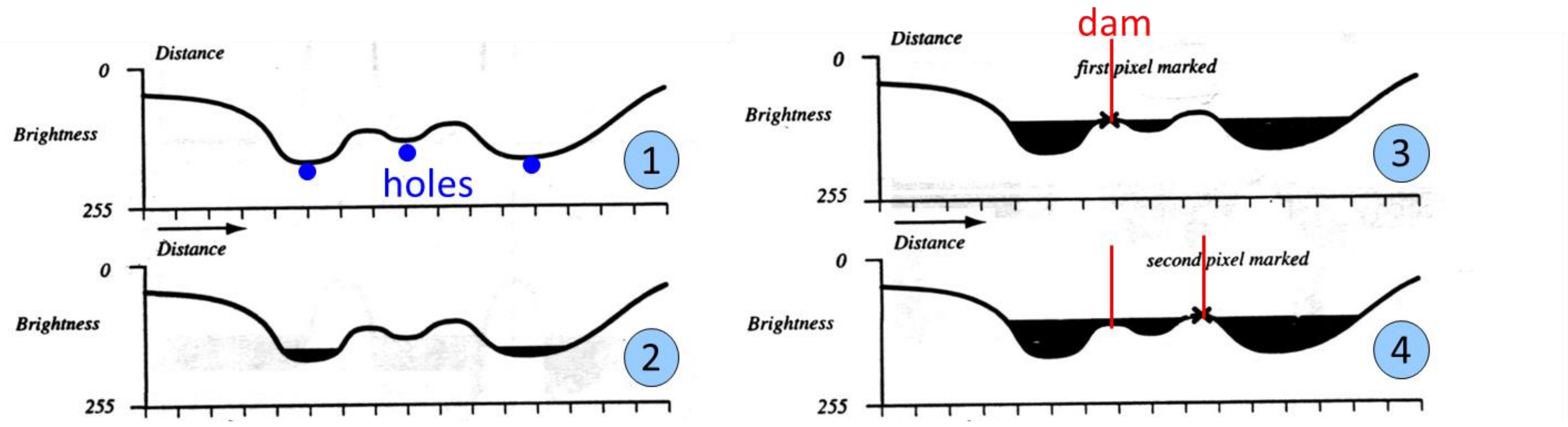
watersheds or divide lines

Mathematical morphology grey level: Watersheds

There are many algorithms to compute it. The simplest conceptually, but slow, is flood filling :

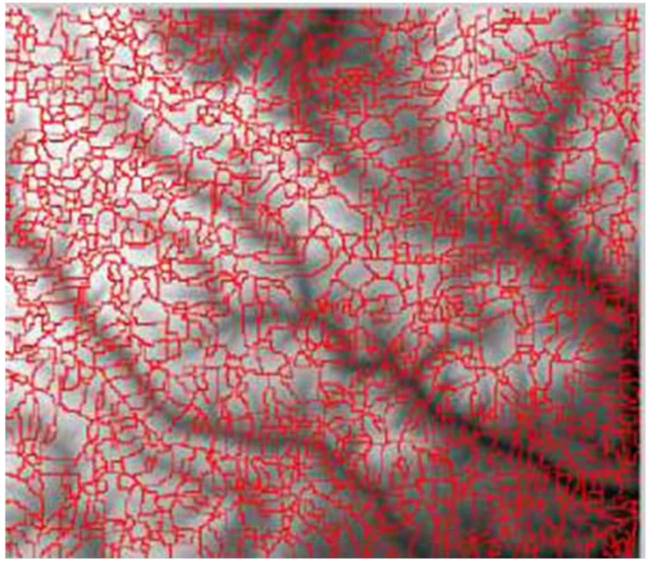
"Suppose that a hole is punched in each local minimum and that the entire relief is flooded from below by letting water rise through the holes slowly.

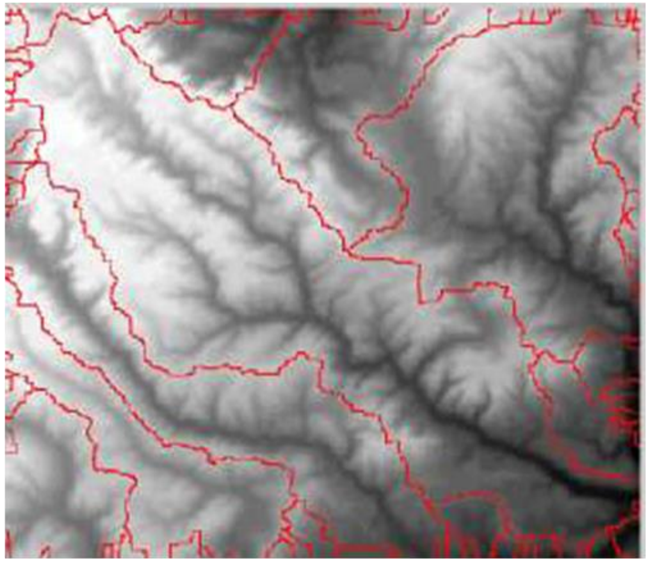
When rising water in distinct catchment basins is about the merge, a dam is built to prevent merging. Dam boundaries correspond to the watershed lines".



Mathematical morphology grey level: Watersheds

Problem 1: too local minima some processing should be done to remove them so that for each interest region there one unique minimum. We can use markers + reconstruction (see later).





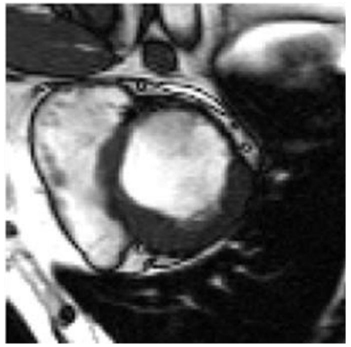
one minimum per catchment basin

Mathematical morphology grey level: Watersheds

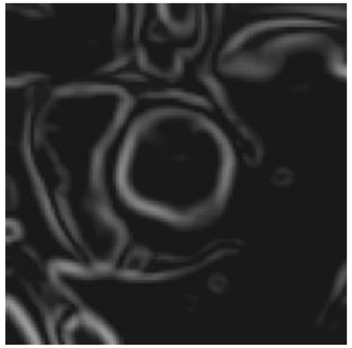
Problem 2: what if the image we want to segment is not a digital elevation model? Is the watershed still useful?

Solution: process the gradient.

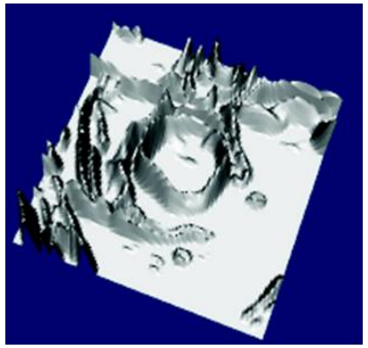
cardiac image



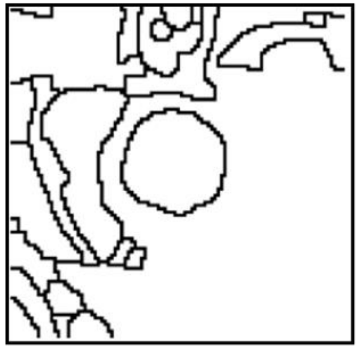
gradient

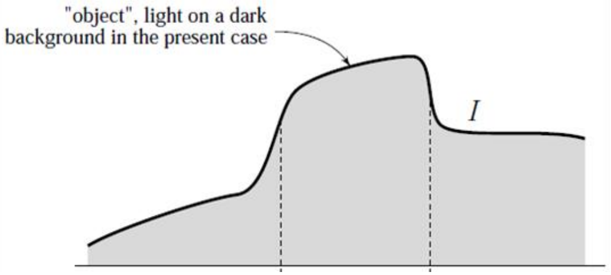


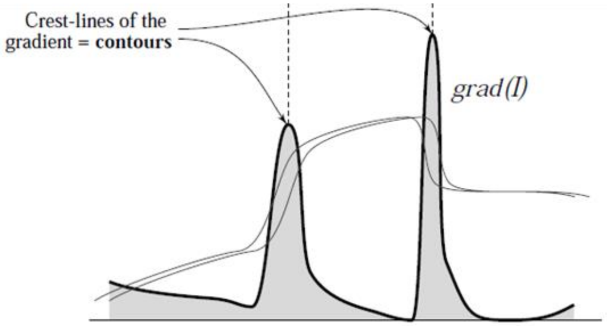
gradient as a landscape



watershed

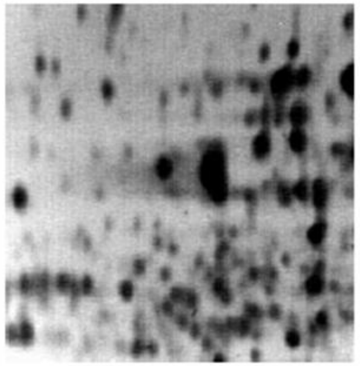




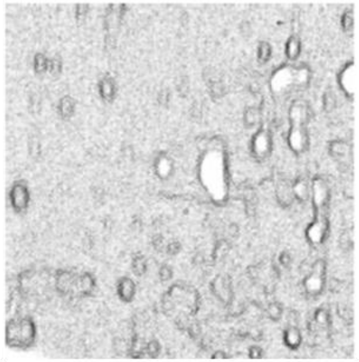


Mathematical morphology grey level: Watersheds

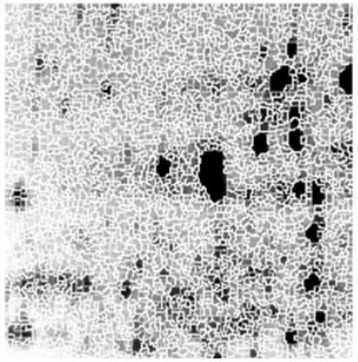
An example combining local minima filtering through markers and gradient: segmentation of electrophoretic gel images.



original image

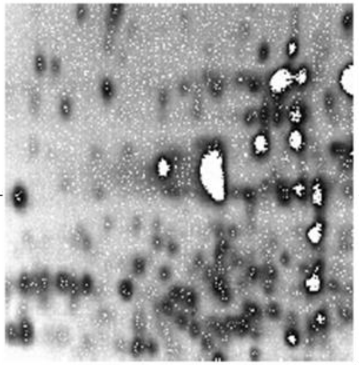


gradient of original image



watershed of gradient

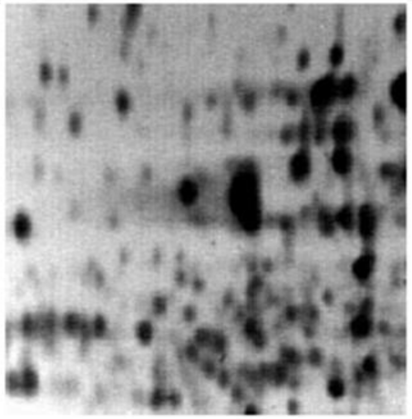
oversegmentation, why ?



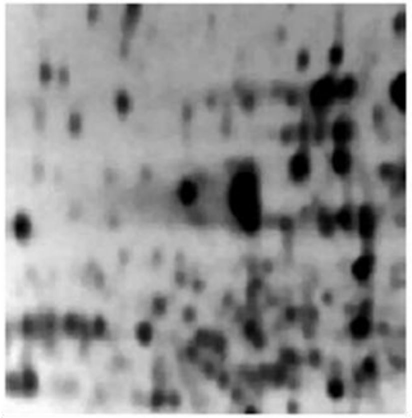
minima

It is necessary impose a single minimum region per spot in the original image.

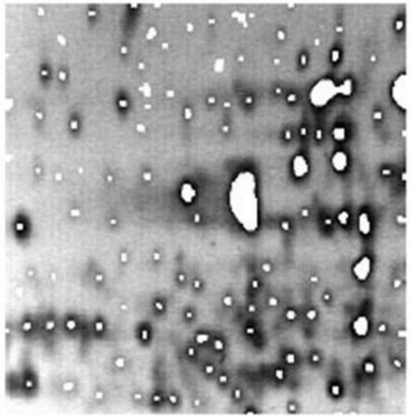
Mathematical morphology grey level: Watersheds



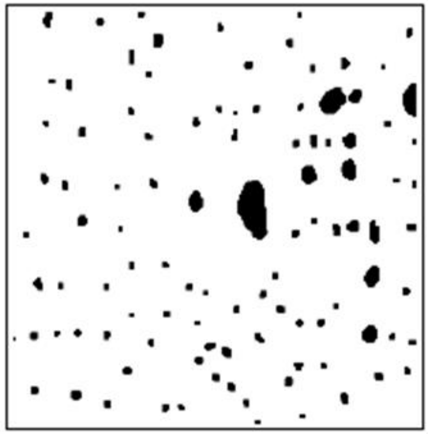
original image



filtered, removing many small local

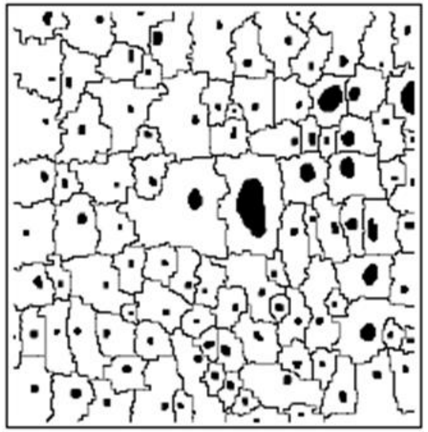


new local minima over original



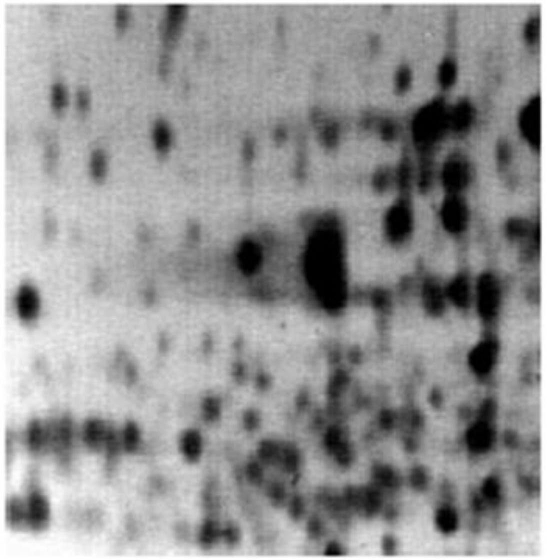
new local minima

zones of influence

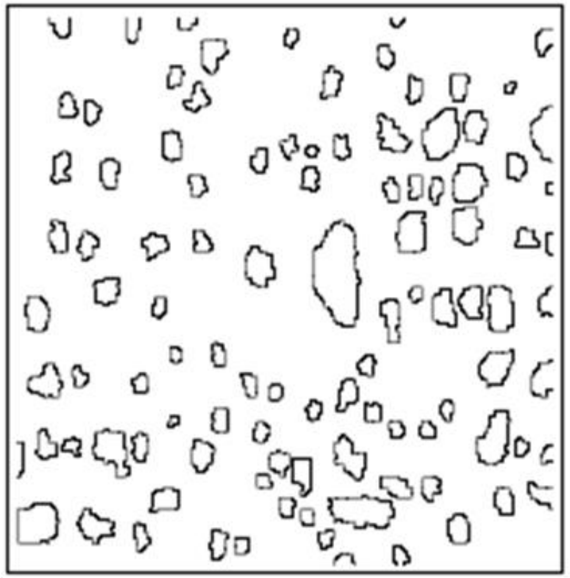


Markers: points that will be imposed as the only minima

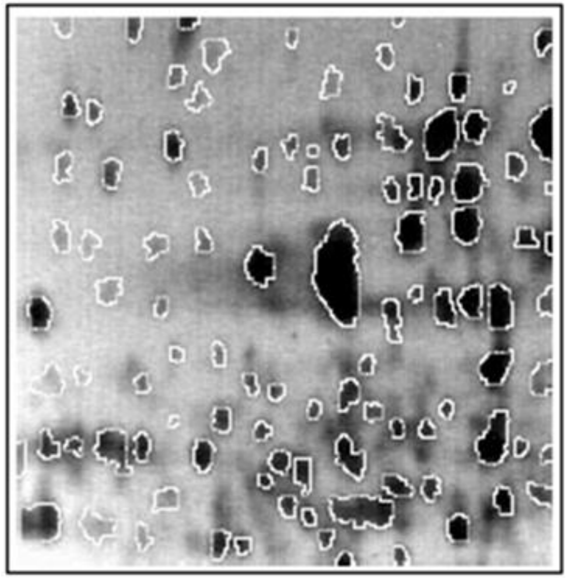
Mathematical morphology grey level: Watersheds



original image

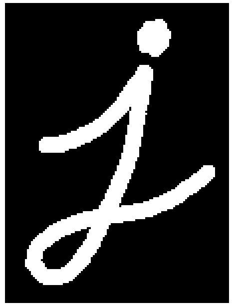


watershed of gradient
reconstructed from the markers

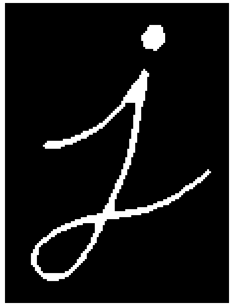


Mathematical morphology in Python

Erosion

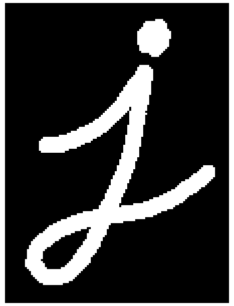


```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```

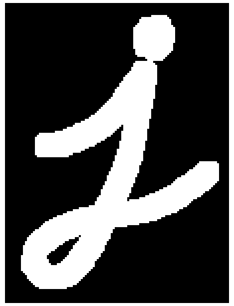


Mathematical morphology in Python

Dilation



```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.dilate(img,kernel,iterations = 1)
```



Mathematical morphology in Python

Opening

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

Mathematical morphology in Python

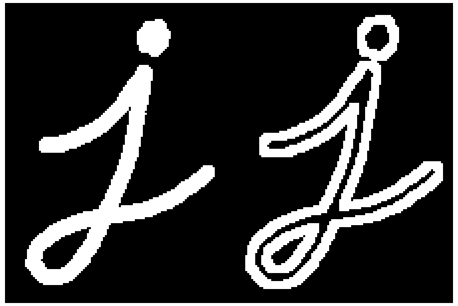
Closing

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

Mathematical morphology in Python

Morphological Gradient

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)
```



Mathematical morphology in Python

Structuring Element

```
# Rectangular Kernel
>>> cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]], dtype=uint8)

# Elliptical Kernel
>>> cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)

# Cross-shaped Kernel
>>> cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))
array([[0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

Fundamentals of Computer Vision

Unit 5: Non-Linear Filtering

Jorge Bernal (Jorge.Bernal@uab.cat)