Mathematical morphology

Dr. Su Su Maung
CEIT Department
Yangon Technological University

Outline of Lecture7

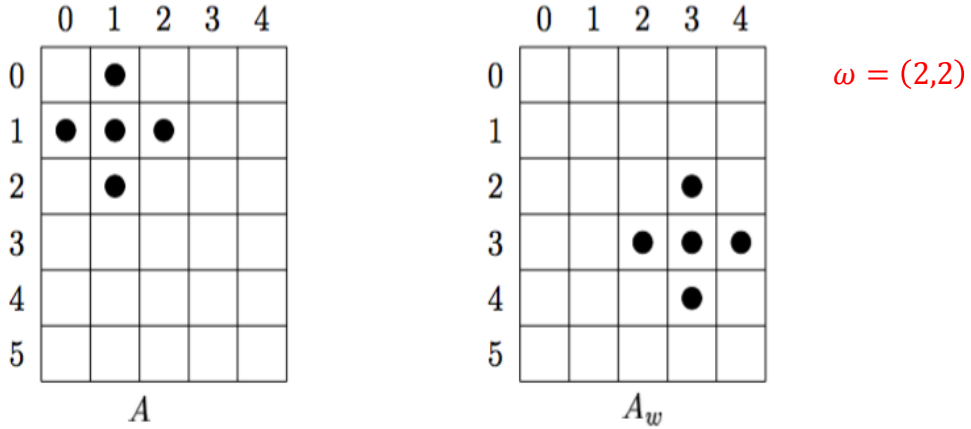
- Introduction
- Dilation and erosion
- Relationship between dilation and erosion
- boundary detection
- Opening and Closing
- Noise removal

Introduction

- Morphology is a branch of image processing which is particularly useful for analyzing shapes in images.
- These are the basic operations of morphology, dilation and erosion. All other operations are built from a combination of these two.
- Non-linear operation.

Translation

- Suppose that A is a set of pixels in a binary image, and $\omega = (x, y)$ is a particular coordinate point. Then A_ω is the set A "translated" in direction (x, y). That is $A_x = \{(a, b) + (x, y) : (a, b) \in A\}$



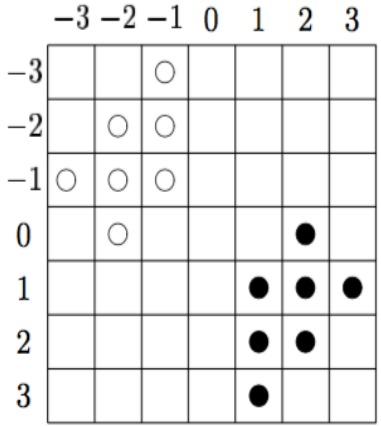
Translation

Note that here we are using matrix coordinates, rather than Cartesian coordinates,

Reflection

- If A is set of pixels, then its reflection, denoted $\hat{A}$, is obtained by reflecting A in the origin:

$$\hat{A} = \{(-x, -y) : (x, y) \in A\}$$



Reflection

Dilation and erosion

- Dilation: Adds pixels to the boundaries of objects in an image
- Erosion: Removes pixels on object boundaries
- These are the basic operations of morphology.
- Other operations are built from a combination of these two.

Dilation

- It grows or thicken objects in a binary image
- Thickening is controlled by a shape referred to as structuring element or kernel.
- Structuring element is a matrix of 1's and 0's.
- Suppose A and B are sets of pixels. Then the dilation of A by B, denoted, $A \oplus B$ is defined as:

$$A \oplus B = \bigcup_{x \in B} A_x$$

- For every point $x \in B$, we translate A by those coordinates. Then we take the union of all these translations.

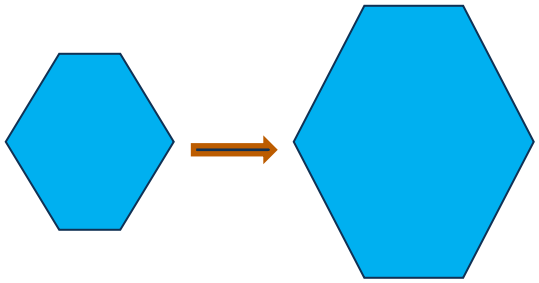
$$A \oplus B = \{(x, y) + (u, v) : (x, y) \in A, (u, v) \in B\}$$

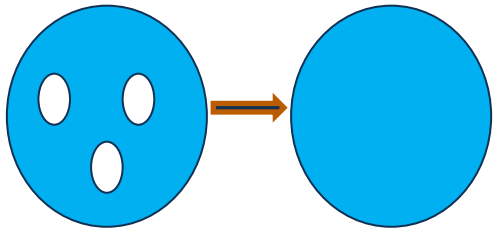
- Dilation is commutative.

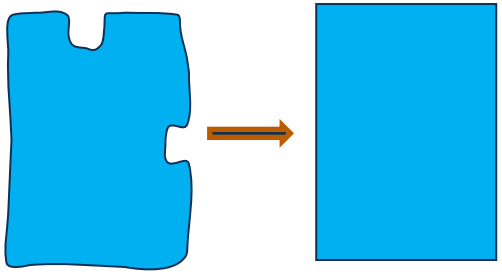
$$A \oplus B = B \oplus A$$

Dilation (cont.)

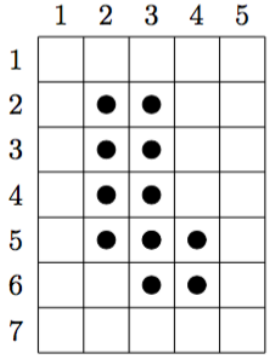
- Dilation expands the connected sets of 1s of a binary image.
- It can be used for
  - growing features
  - filling holes and gaps



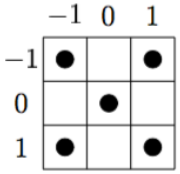




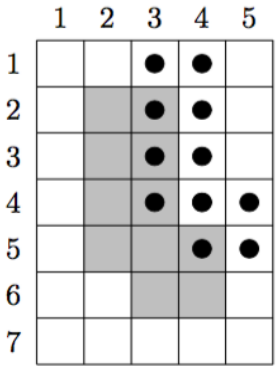
Dilation (cont.)



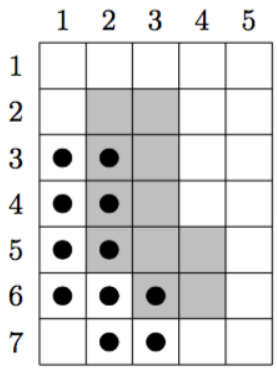
A



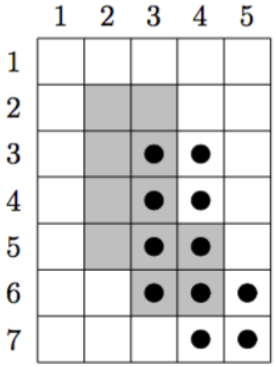
B



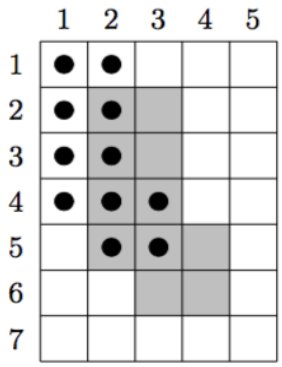
$A_{(-1,1)}$



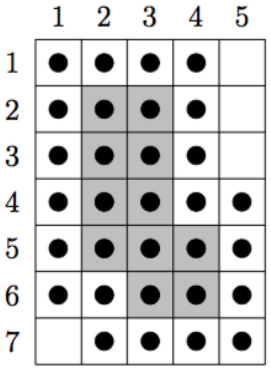
$A_{(1,-1)}$



$A_{(1,1)}$



$A_{(-1,-1)}$



$A \oplus B$

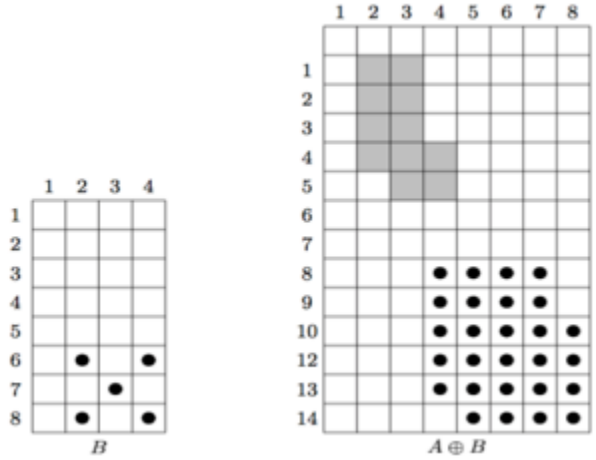
$B = \{(0,0), (1,1), (-1,1), (1,-1), (-1,-1)\}$

Dilation (cont.)

- it is not necessarily true that the original object A will lie within its dilation $A \oplus B$. Depending on the coordinates of B, $A \oplus B$ may end up quite a long way from A.

$$B = \{(7,3), (6,2), (6,4), (8,2), (8,4)\}$$

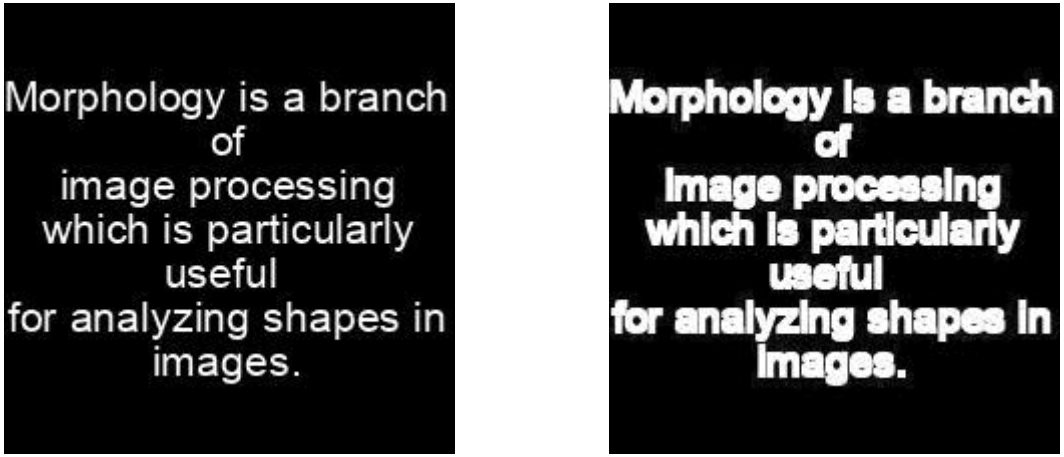
$$A \oplus B = A_{(7,3)} \cup A_{(6,2)} \cup A_{(6,4)} \cup A_{(8,2)} \cup A_{(8,4)}$$



A dilation for which $A \nsubseteq A \oplus B$

Dilation in Python

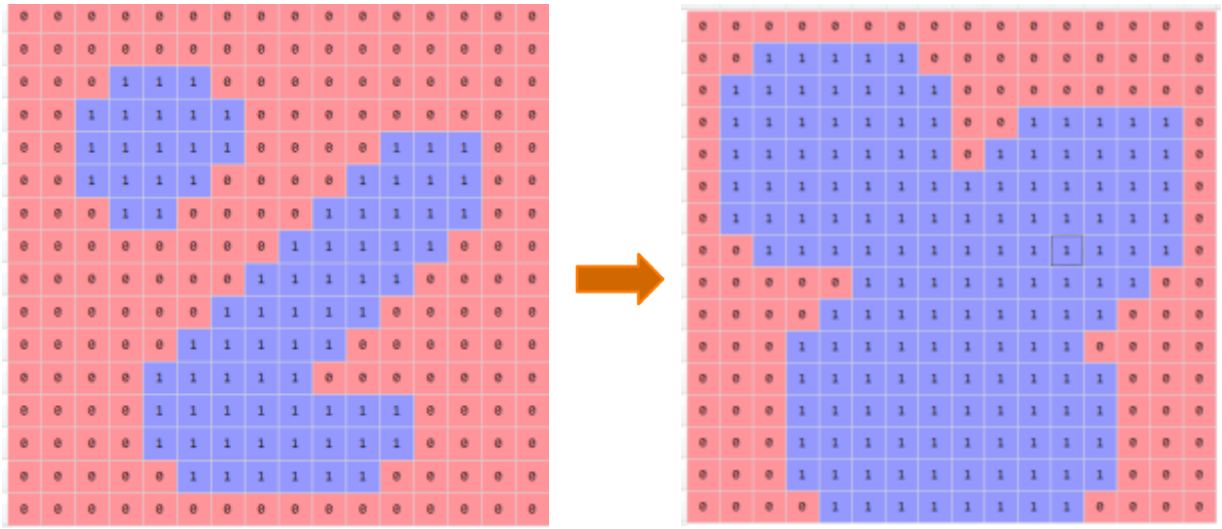
```
from skimage.morphology import binary_dilation as bwdilate
img = cv2.imread('text.png',0)
sq = np.ones((3,3))
imge = bwerode(img,sq)
```



Dilation of a binary image

Dilation (cont.)

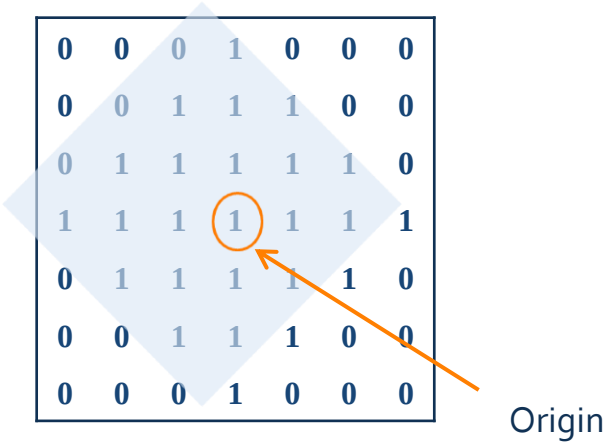
- Dilation fills gaps in an image.



Effect of dilation using 3x3 square structuring element

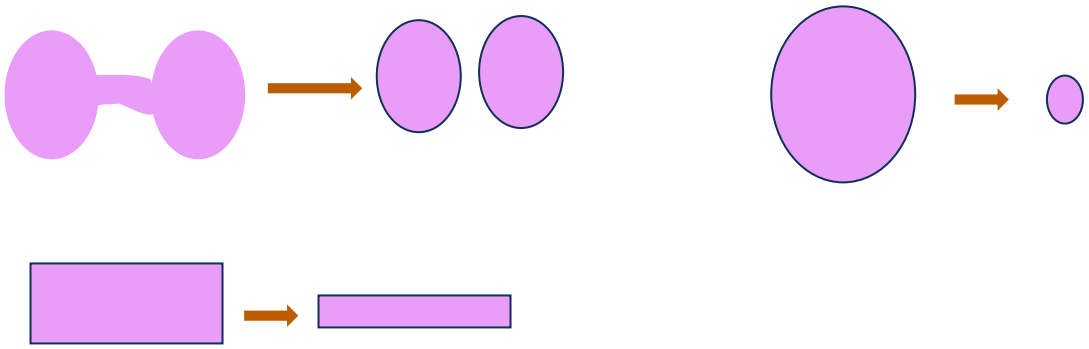
Structuring element

- The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element
- It's a matrix of 1's and 0's
- The center pixel of the structuring element, called the origin



Erosion

- Erosion shrinks the connected sets of 1s of a binary image.
- It can be used for
  - shrinking features
  - Removing bridges, branches and small protrusions



Erosion (cont.)

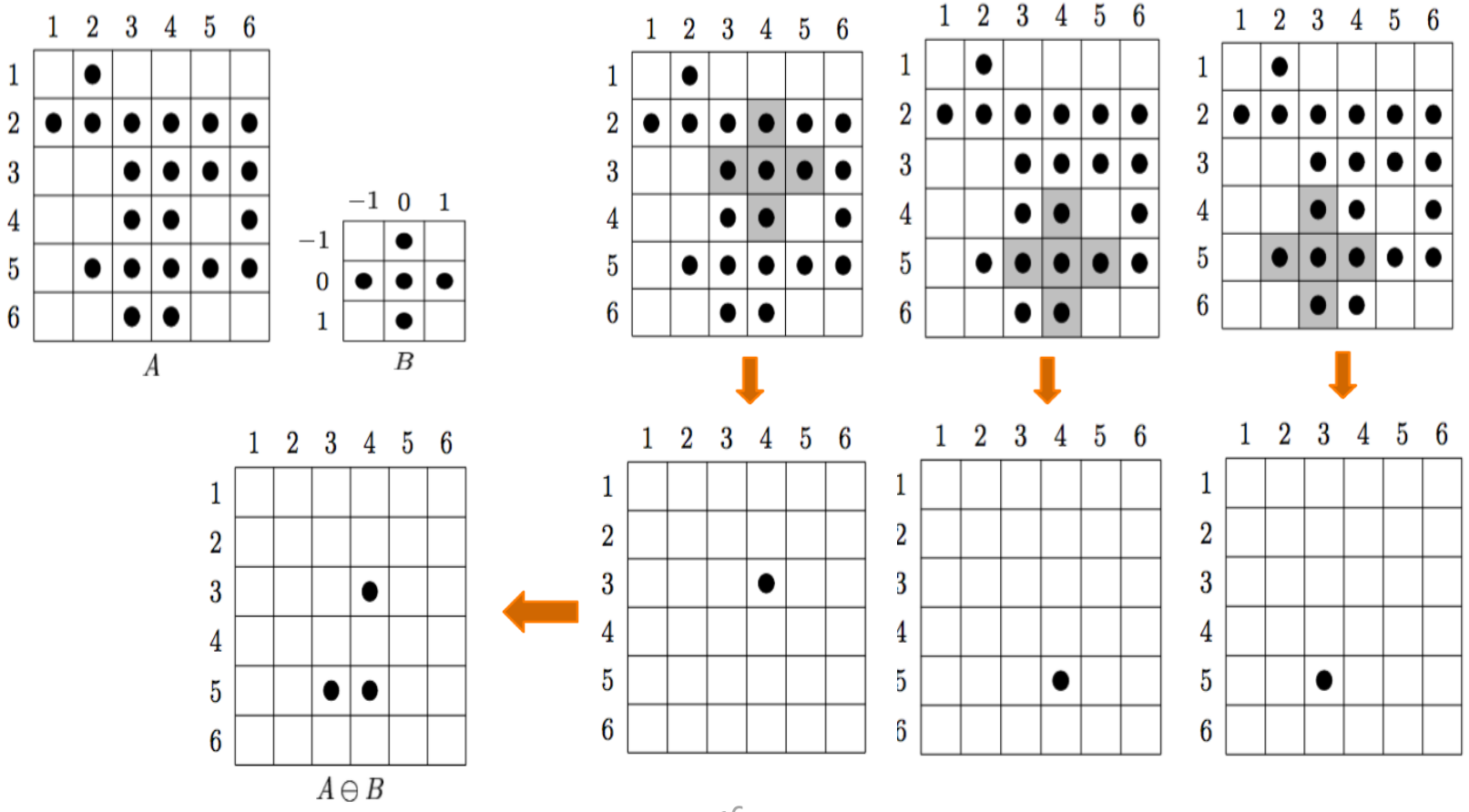
- Given sets A and B , the erosion of A by B , written $A \ominus B$, is defined as:

$$A \ominus B = \{w : B_w \subseteq A\}$$

- The erosion A by B consists of all points $\omega = (x, y)$ for which B_ω is in A.

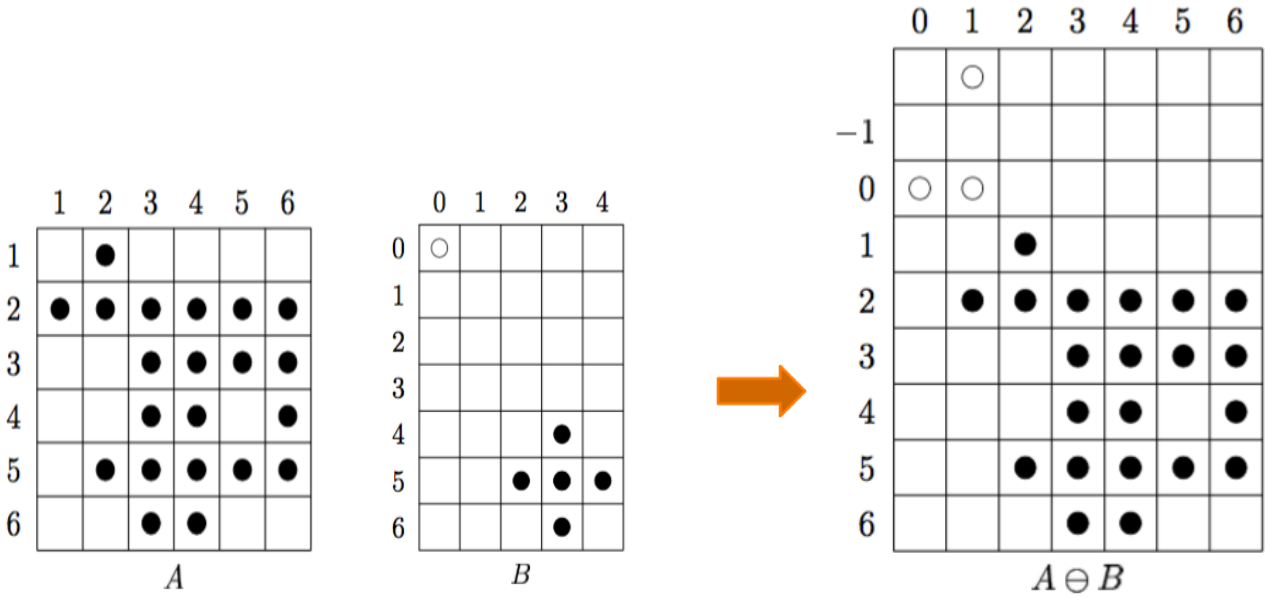
- $A \ominus B$ was a subset of A. This is not necessarily the case; it depends on the position of the origin in B . If B contains the origin ,then the erosion will be a subset of the original object.

Erosion (cont.)



Erosion with a cross-shaped structuring ele

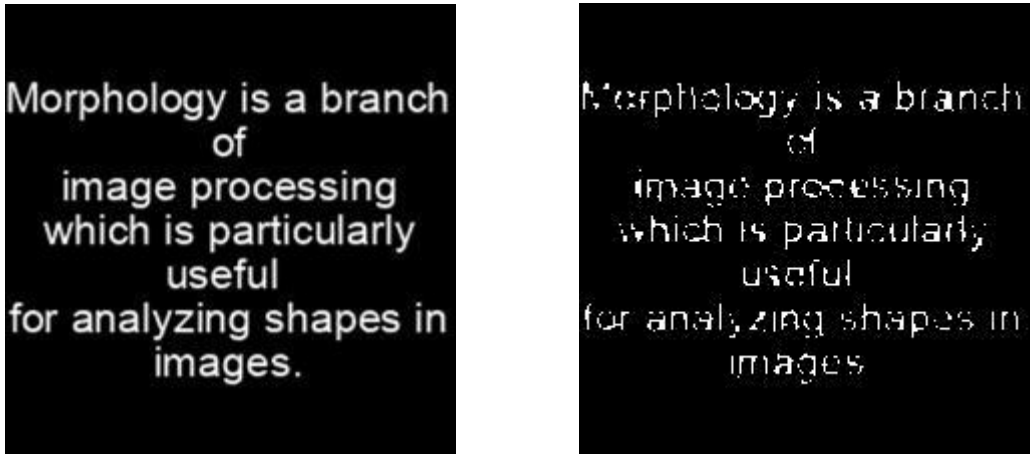
Erosion (cont.)



Erosion with a structuring element not containing the origin

Erosion in Python

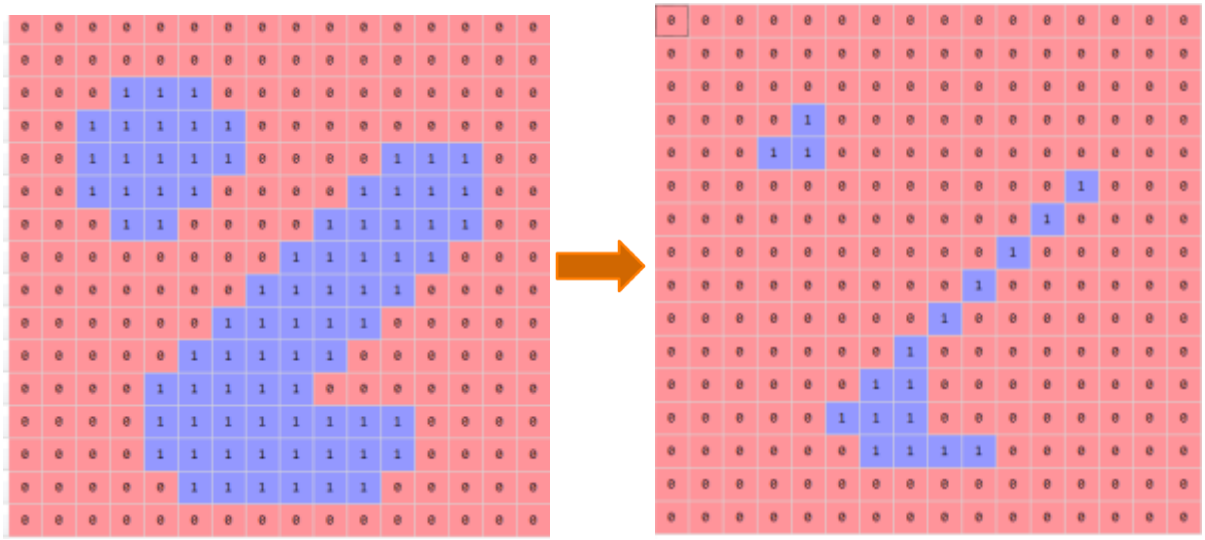
```
from skimage.morphology import binary_erosion as bwerode
img = cv2.imread('text.png',0)
sq = np.ones((3,3))
imge = bwerode(img,sq)
```



Erosion of a binary image

Erosion (cont.)

- Erosion eliminates unwanted detail.



Effect of dilation using 3x3 square structuring element

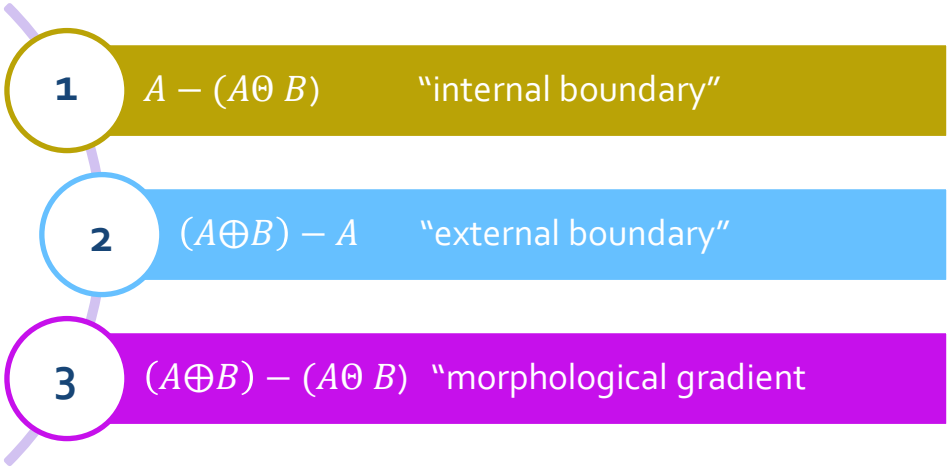
Relationship between erosion and dilation

- It can be shown that erosion and dilation are "inverses" of each other;
- the complement of an erosion is equal to the dilation of the complement.

$$\overline{A \ominus B} = \bar{A} \oplus \bar{B}$$

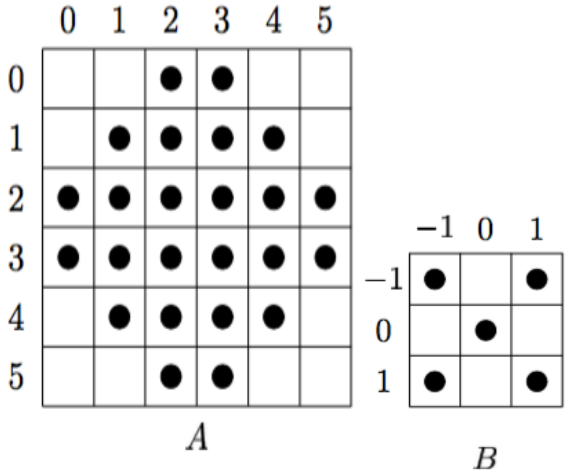
Boundary detection

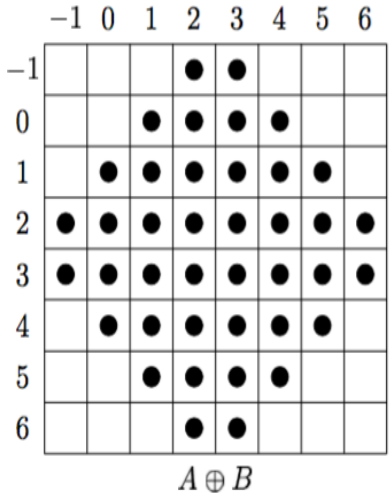
- If A is an image, and B is a small structuring element consisting of point symmetrically places about the origin, then we can define the boundary of the image by any of the following methods:

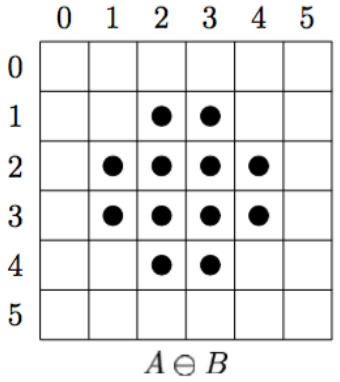


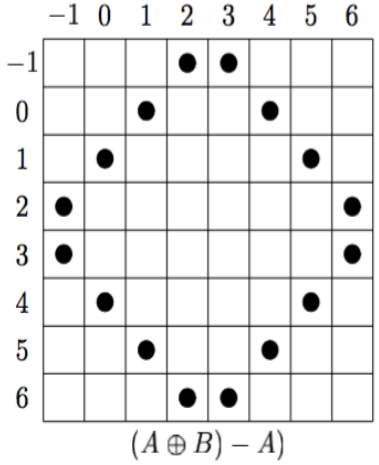
- In each definition the minus refers to set difference.

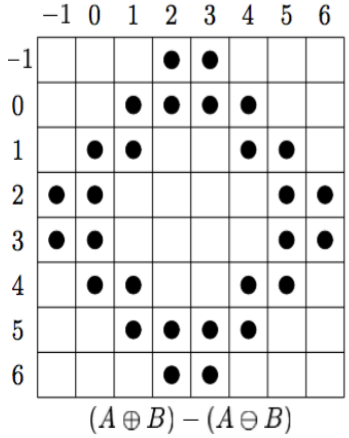
Boundary detection (cont.)





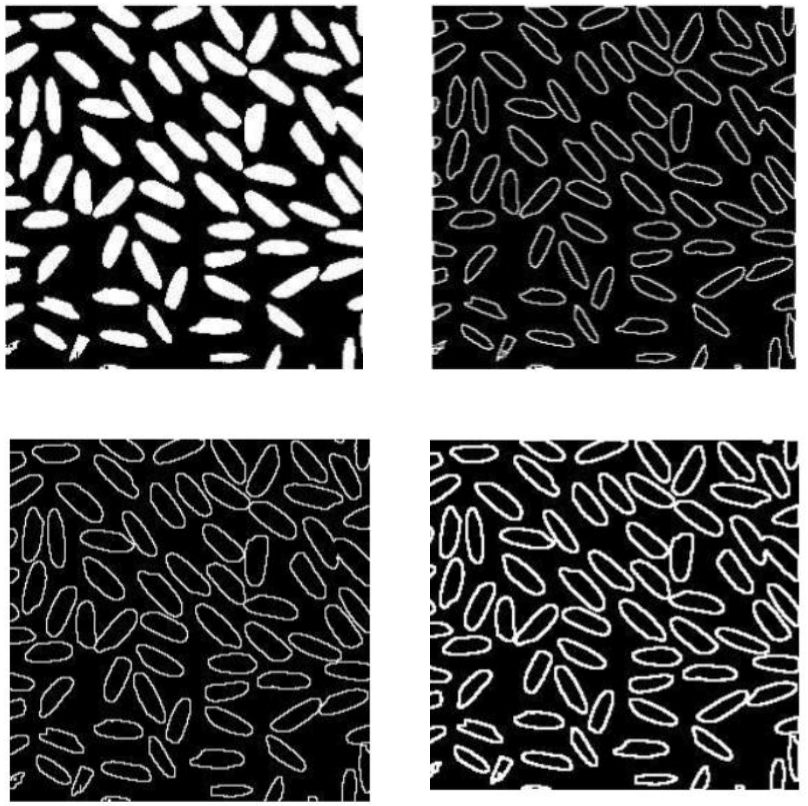






boundary detection (cont.)

```
n=io.imread('images/rice.png')
sq=np.ones((3,3))
p=~((n>40) & (n<120))
pe=bwerode(p,sq)
#Internal boundary
p_int=p&~pe
#External boundaryp
d=bwdilate(p,sq)
p_ex=pd&~p
#gradient
p_grad = pd&~pe
```



Internal boundary, external boundary and the morphological gradient of a binary image of a binary image

Opening and Closing

- Opening :
  - An Erosion followed by a dilation

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

- Closing :
  - A dilation followed by an erosion

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

Opening

- Given and a structuring element B, the opening of A by B, denoted $A \circ B$, is defined as:

$$A \circ B = (A \Theta \mathrm{B}) \oplus B$$

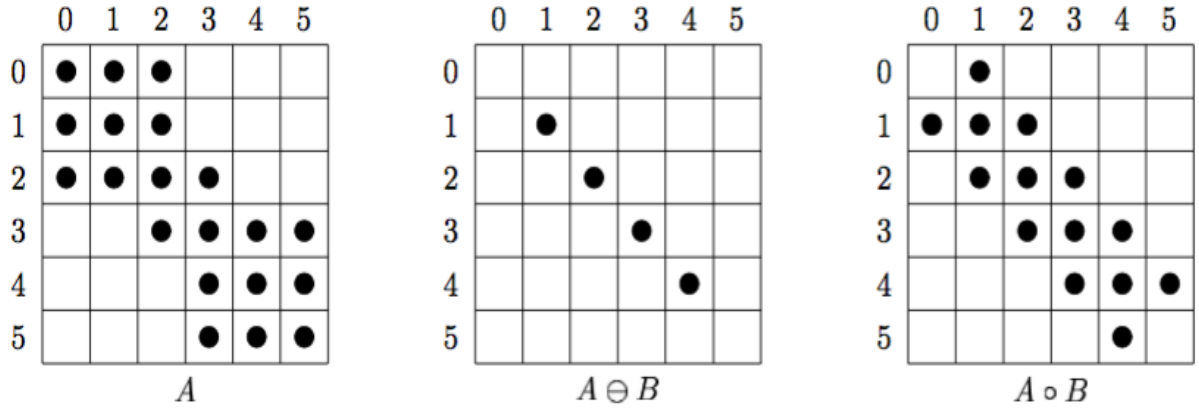
- So an opening consists of an erosion followed by a dilation. An equivalent definition is

$$A \circ B = \cup \{B_w : B_w\} \dot{\subseteq} A$$

- That is, $A \circ B$ is the union of all translations of B which fit inside A. Note the difference with erosion: the erosion consists only (o,o) of the point of B for those translations which fit inside A; the opening consists of all of B.

- Opening is implemented by the bwopen functions..

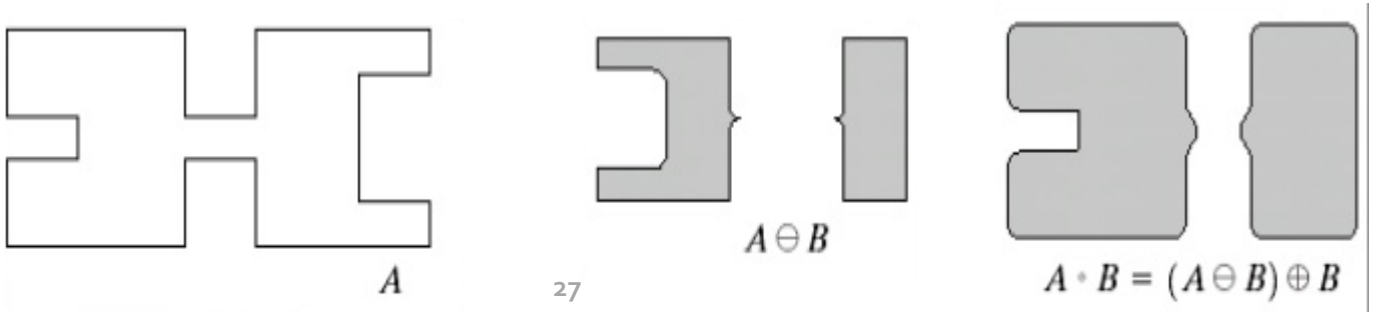
Opening (cont.)



Opening

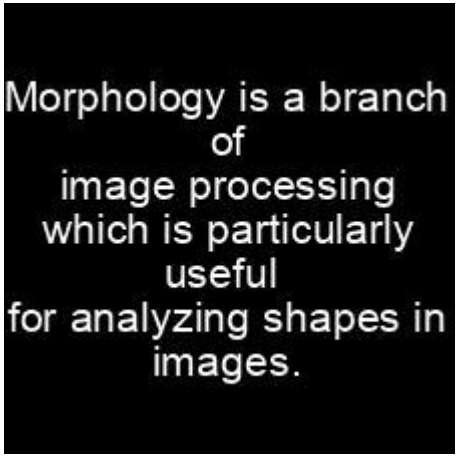
Properties of opening

- $A \circ B \subseteq A$. Note that this is not the case with erosion; as we have seen, an erosion may not necessarily be a subset.
- $(A \circ B) \circ B = A \circ B$. That is, an opening can never be done more than once. This property is called idempotence. Again, this is not the case with erosion; you can keep on applying a sequence of erosions to an image until nothing is left.
- If $A \subseteq C$, then $(A \circ B) \subseteq (C \circ B)$.
- Opening tends to "smooth" an image, to break narrow joins, and to remove thin protrusions.



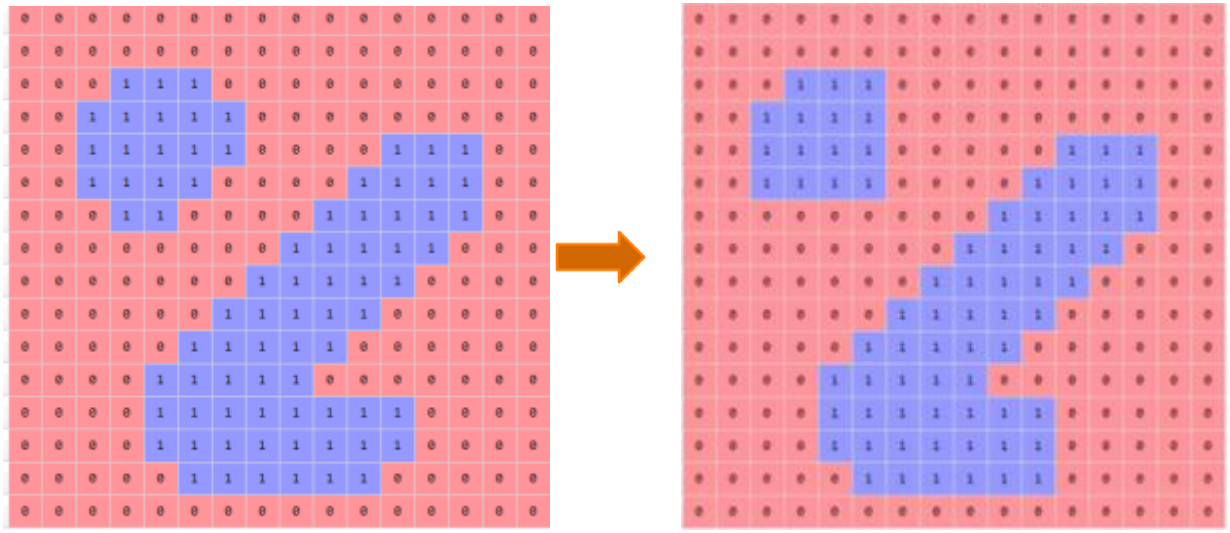
Opening in Python

```
from skimage.morphology import binary_opening as bwopen
img = cv2.imread('text.png',0)
sq = np.ones((3,3))
imge = bwopen(img,sq)
```



Opening of a binary image

Opening (cont.)



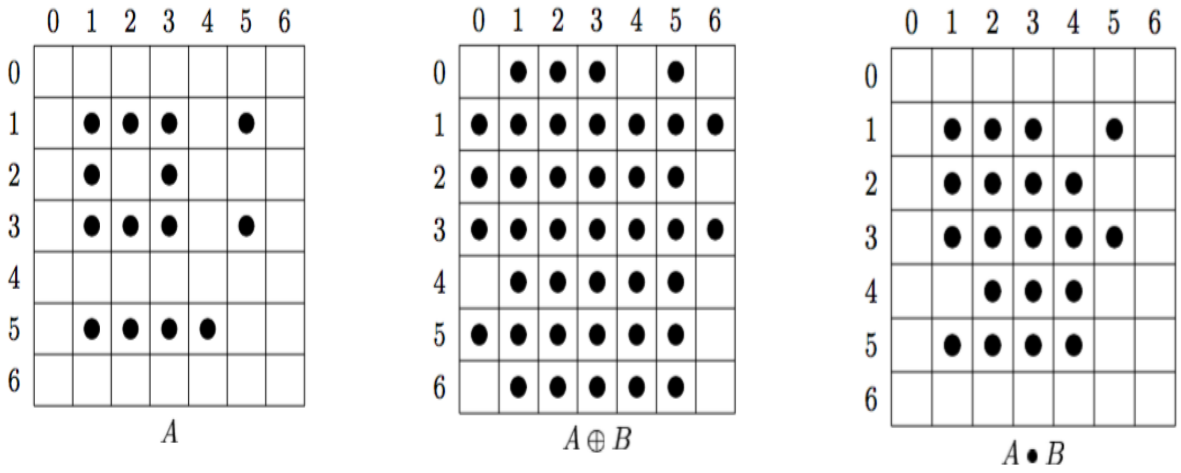
Effect of opening using 3x3 square structuring element

Closing

- a dilation followed by an erosion, and is denoted : $A \bullet B$

$$A \bullet B = (A \oplus \mathrm{B}) \Theta B$$

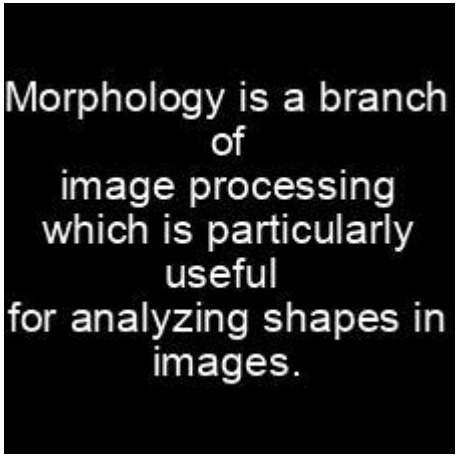
- Closing is implemented by the bwclose function.
-

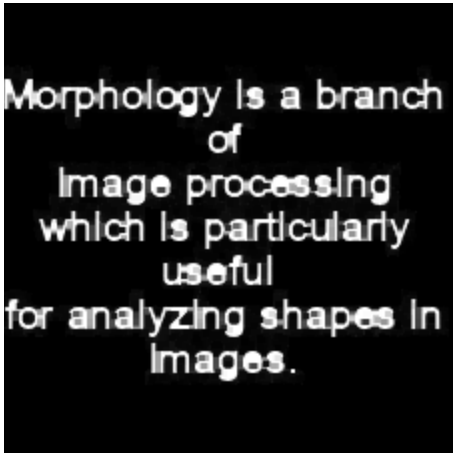


Closing

Closing in Python

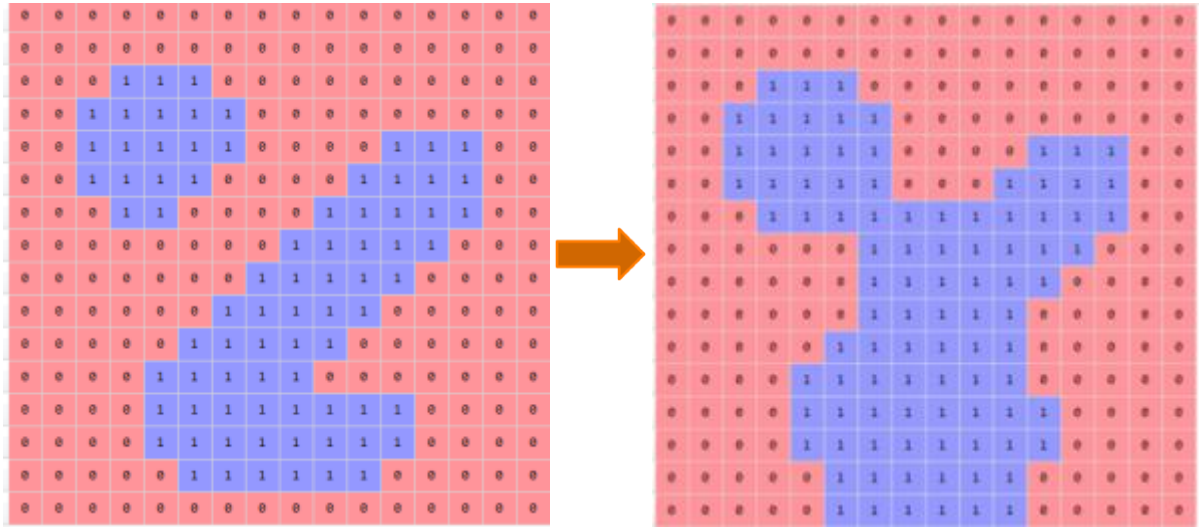
```
from skimage.morphology import binary_closing as bwclose
img = cv2.imread('text.png',0)
sq = np.ones((3,3))
imge = bwclose(img,sq)
```





Closing of a binary image

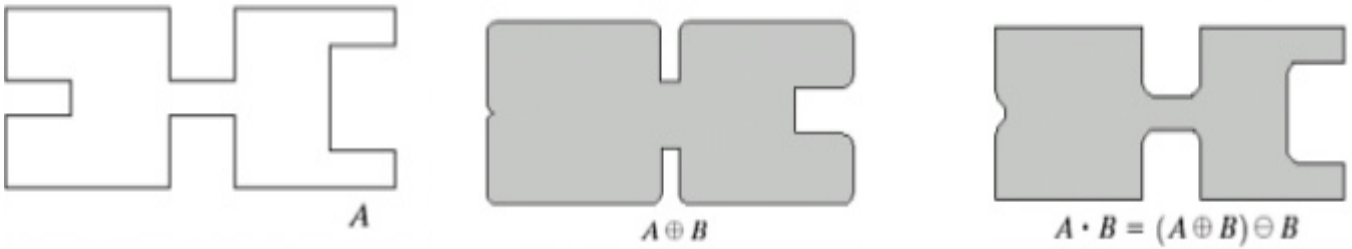
Closing (cont.)



Effect of closing using 3x3 square structuring element

Properties of closing

- $A \subseteq (A \bullet B)$.
- $(A \bullet B) \bullet B = A \bullet B$; that is, closing, like opening, is idempotent.
- If $A \subseteq C$, then $(A \bullet B) \subseteq (C \bullet B)$.
- Closing tends also to smooth an image, but it fuses narrow breaks and thin gulfs, and eliminates small holes.



Noise removal

- Suppose A is a binary image corrupted by impulse noise—some of the black pixels are white, and some of the white pixels are back.
- Then $A \ominus B$ will remove the single white pixels, but will enlarge the holes.
- We can fill the holes by dilating twice: $\big((A \ominus B) \oplus B\big) \oplus B$.
- The first dilation returns the holes to their original size; the second dilation removes them. But this will enlarge the objects in the image. To reduce them to their correct size, perform a final erosion: $\big(((A \ominus B) \oplus B\big) \oplus B) \ominus B$
- Thus this noise removal method is in fact an opening followed by a closing: $(A \circ B) \bullet B)$
- This is called morphological filtering.

An application: noise removal

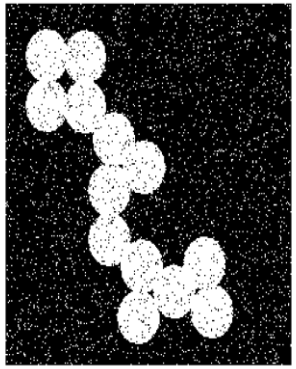
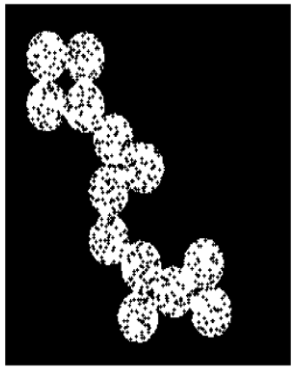
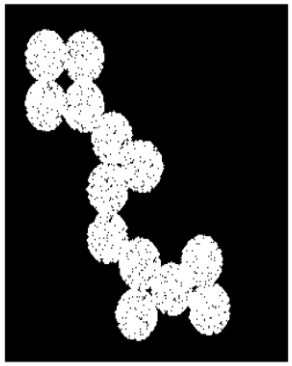


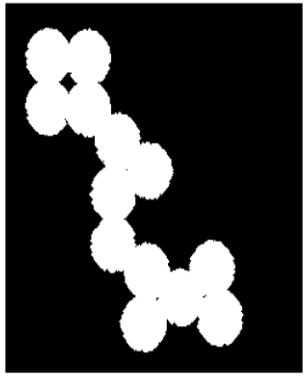
Image with impluse noise



$A \ominus B$

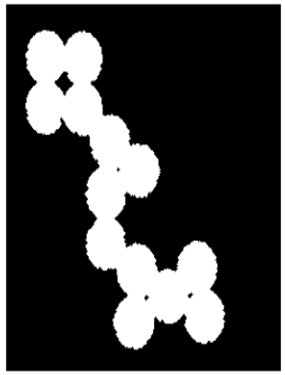


$((A \ominus B) \oplus B)$



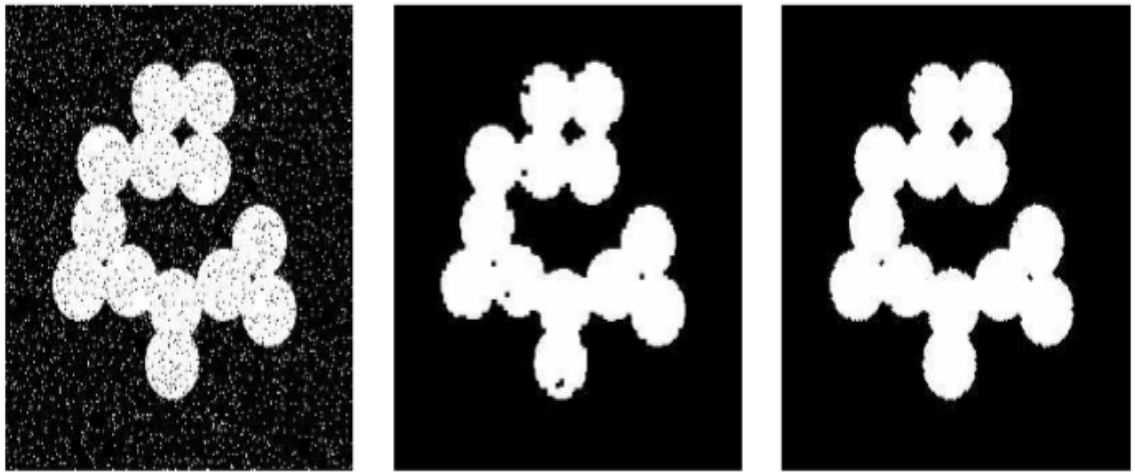
$(((A \ominus B) \oplus B) \oplus B)$

```
import skimage.io as io
import numpy as np
img=io.imread('circles.png')
x=np.random.random_sample(img.sha
pe)
img[np.nonzero(x>0.95)]=0
img[np.nonzero(x<=0.05)]=1
```



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

An application: noise removal



(a) (b) (c)

A noisy binary image and results after morphological filtering with different structuring elements.

```
import skimage.io as io
import numpy as np
from skimage.morphology import binary_closing as bwclose
from skimage.morphology import binary_opening as bwopen
img=io.imread('circle_noise.jpg')
sq = np.ones((3,3))
img2=bwclose(bwopen(img,sq),sq)
```

Relationship between opening and closing

- Opening and closing share a relationship very similar to that of erosion and dilation:
- the complement of an opening is equal to the closing of a complement, $\overline{A \circ B} = \overline{A} \bullet \hat{B}.$

$$\overline{A \bullet B} = \overline{A} \circ \hat{B}$$

- complement of an closing is equal to the opening of a complement.

Next Week Lecture (Week8)

- Lecture8:Color Processing

Thank You