Chapter 7 Morphological Image Processing

Agenda

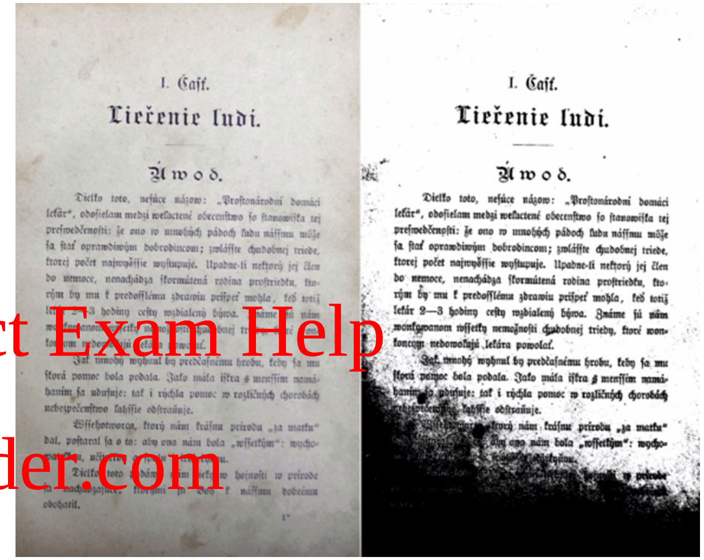
- Introduction to morphological operations
- Set theory preliminaries
- Binary morphological operations:
  - Erosion, Dilation, Opening, Closing
- Connected components and labelling
- Morphological algorithms
  - Boundary extraction, region filling, hit-or-miss transform
- Grayscale morphological operations

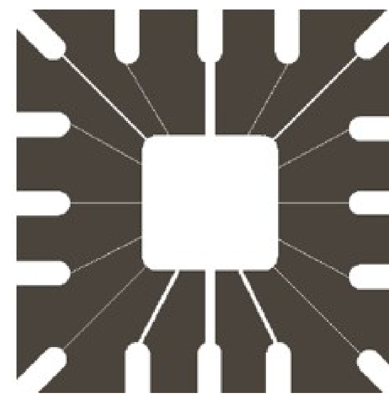
Morphological Operations

- Morphology= Shape, Form, Structure
- Morphological operations are used to extract image components for representation and description of region shape, such as boundaries and skeletons.
- Based on set theory
- Applicable to both binary and gray-level images. Application to binary (black-and-white) images is more common.

Binarization of Images

- Requires binarization of images before applying binary morphological operations.
- Binary images can be obtained from
  - Thresholding gray-level images
    - If f(x,y) > q then g(x,y) = 1 else g(x,y) = 0
  - As a result of feature detectors
- Often want to count or measure shape of 2D binary image regions

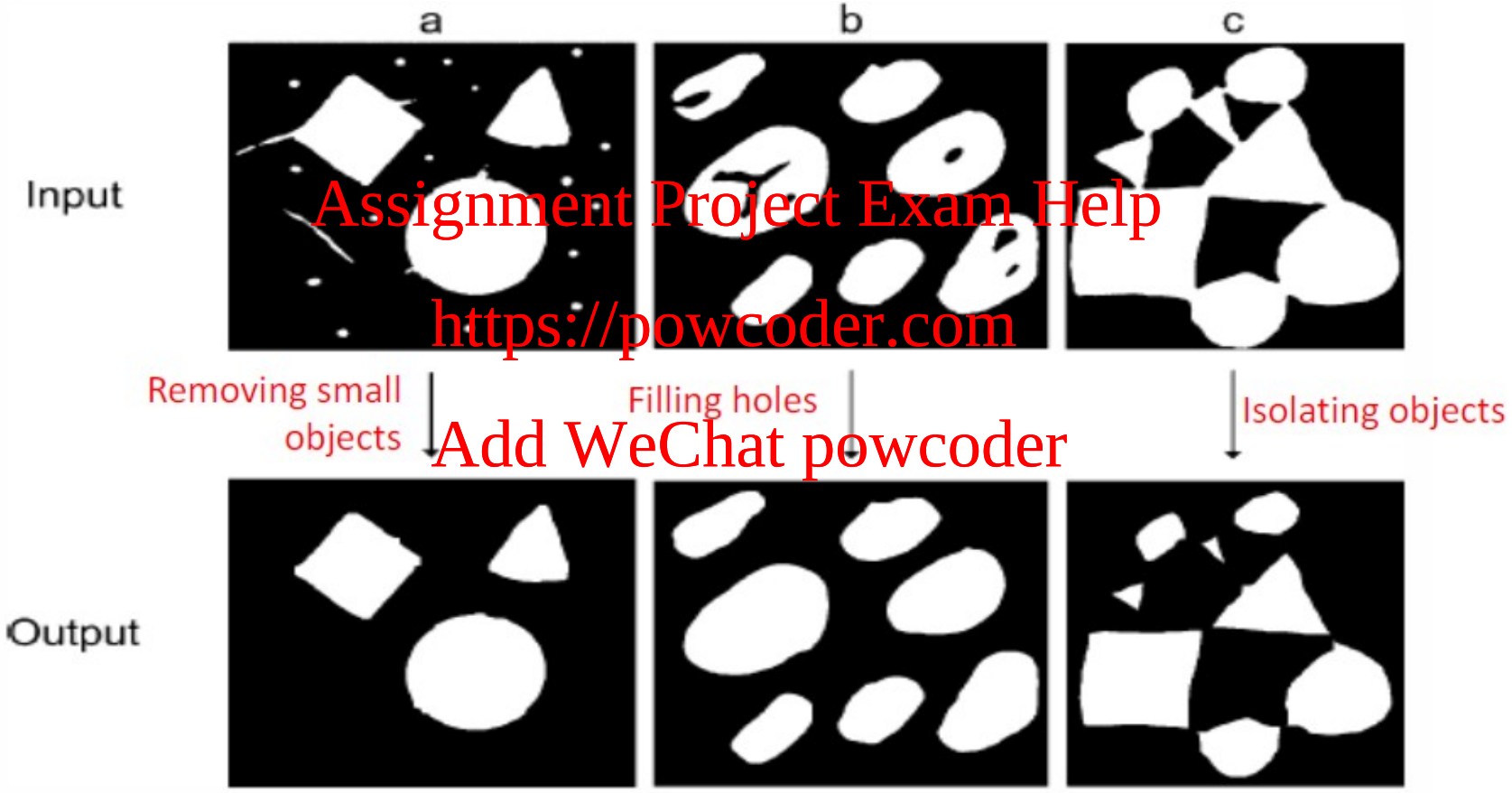

I. Časť.
Liečenie ľudí.
Úvod.



Applications of Morphological Operations

- Removing Small Objects
  - Remove noise as a side effect of thresholding
  - Reduce the effect of over-segmentation: small regions erroneously segmented
- Filling Holes
  - Remove holes inside the object due to under-segmentation
- Isolating Objects
  - Ensure that the objects are separated from each others

Morphological Processing Examples



7.1 Binary Morphological Operations

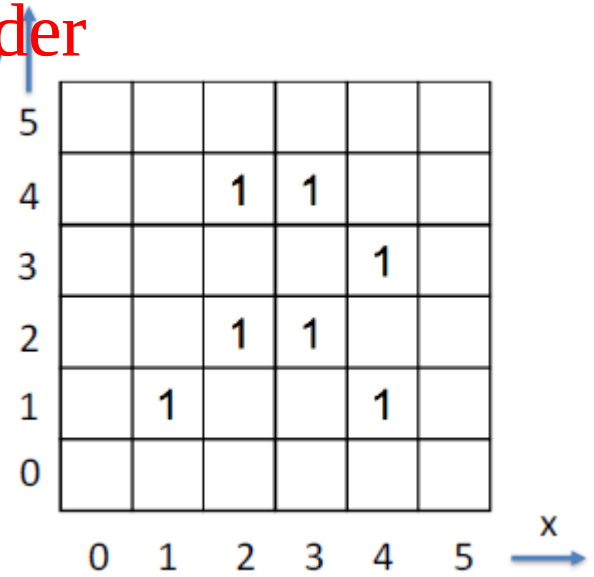
Set Theory Preliminaries

Definition:

- For a binary image , is the (unordered) set of pairs such the image value at is equal to 1:

- Example:

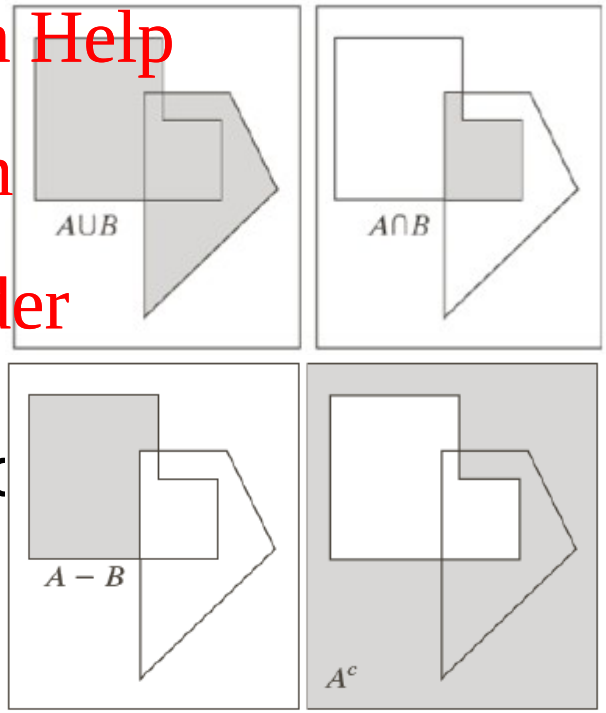
A = { (2,4), (3,4), (4,3), (2,2), (3,2), (1,1), (4,1) }



Basic Set Operations

Given and

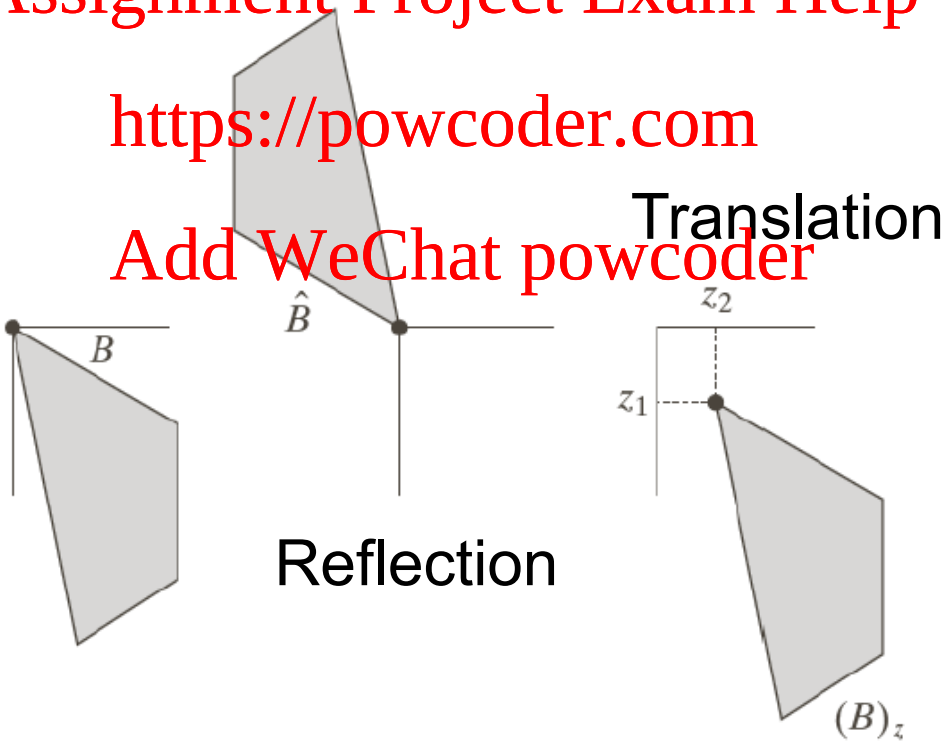
- is an element of set A:
- Union of two sets A and B:
- Intersection of two sets A and B
- Difference between two sets A and B
- Complement of A



Reflection and Translation

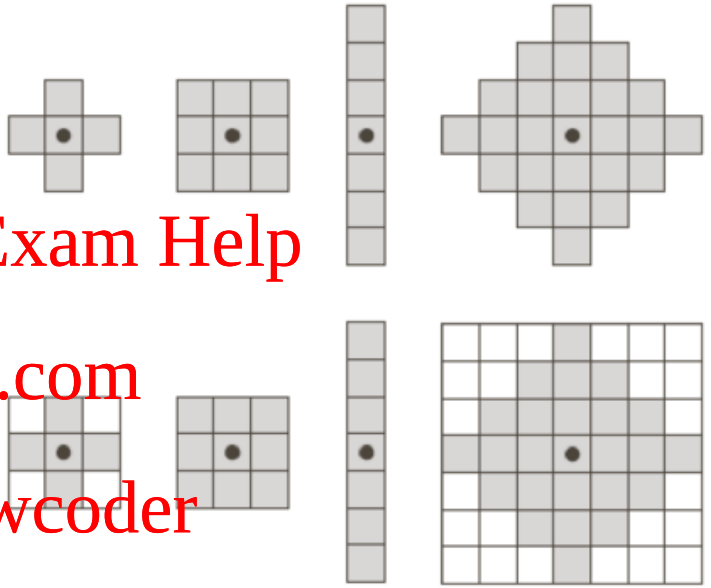
Set reflection: $\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$

Set translation by z: $(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$



Structuring Element (SE)

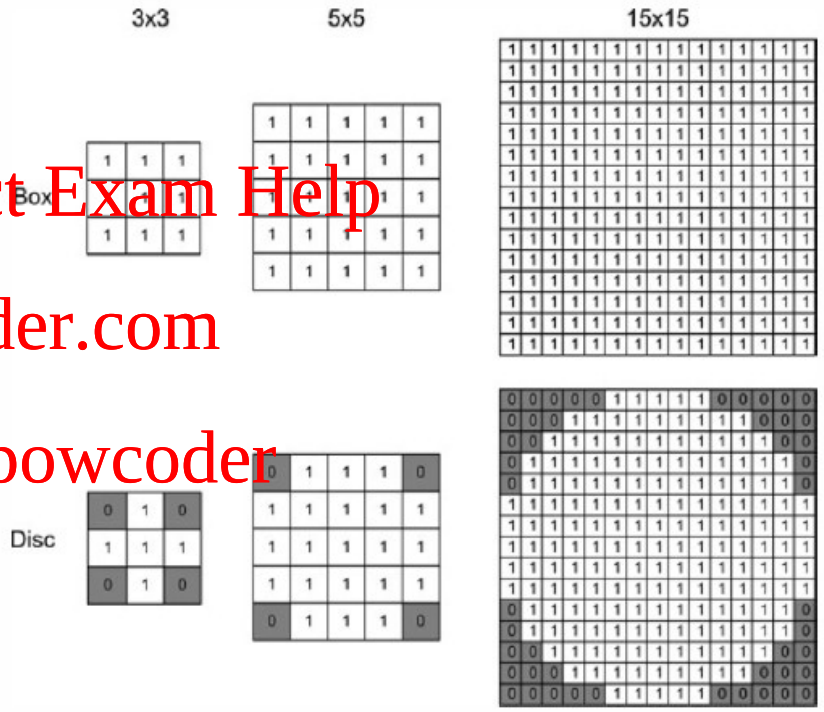
- A structuring element (SE) is applied to each pixel of the input image in morphological operations.
- The SE is small set or subimage, used to probe for structure
- Free to design the SE to fit different purposes



- Black dot denotes the origin of SE
- Gray = 1
- White = 0

Type and Size of SE

- Type and size to use is up to the user to determine
  - *Box-shaped SE* tends to preserve sharp object corners
  - *Disk-shaped* SE tends to round the corners of the objects

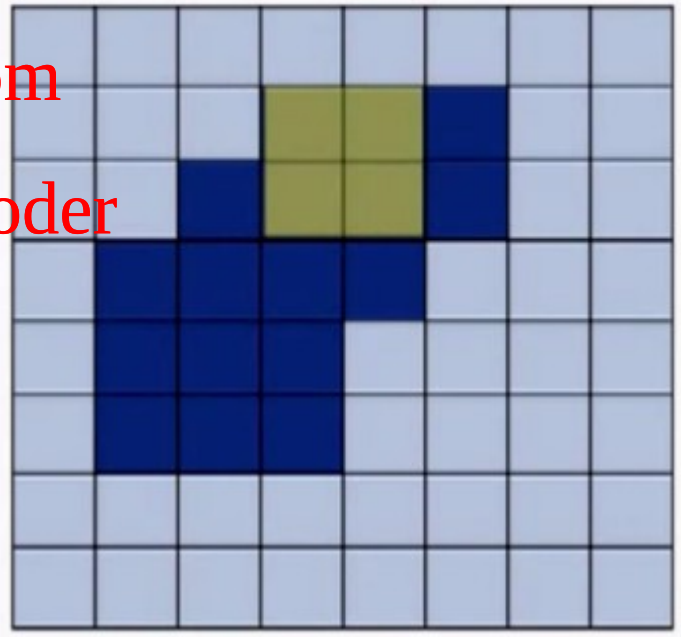


Basic Morphological Operations

- A SE is applied through either a **Fit** or a **Hit** operation.
- Applying these two operations to each pixel in an image are called **Erosion** and **Dilation**, respectively.
- Can combine these two operations to come up with compound operations:
  - **Opening**
  - **Closing**

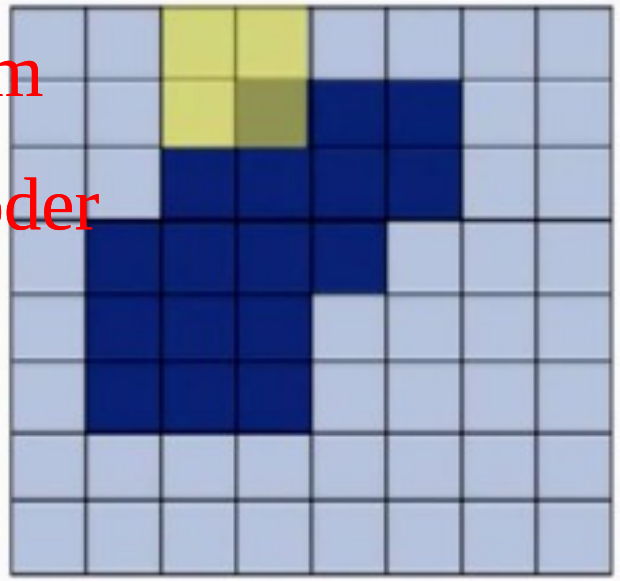
Fit

- For each ‘1’ in the SE, we investigate whether the pixel at the same position in the image is also a ‘1’.
- If ALL of the ‘1’s in the SE are covered by the image,
  - The SE **fits** the image at the pixel position in question (the one on which the SE is centered).
  - This pixel is set to ‘1’ in the output image. Otherwise, it is set to ‘0’ in the output image.

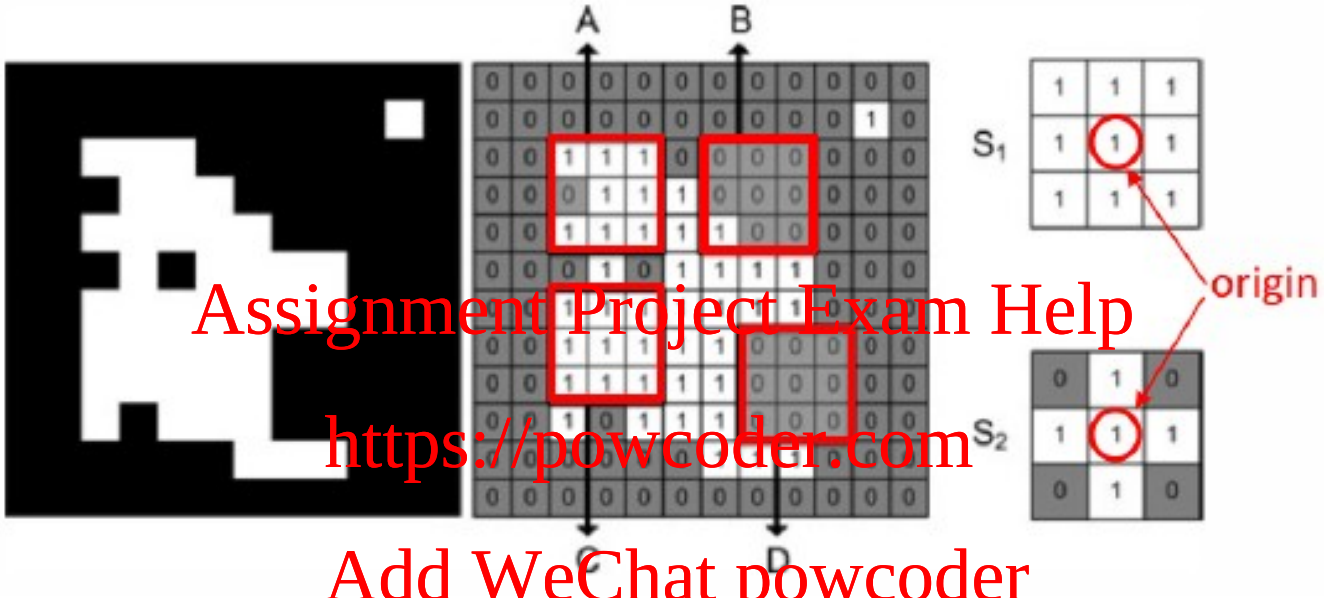


Hit

- For each ‘1’ in the SE, we investigate whether the pixel at the same position in the image is also a ‘1’.
- If any ONE of the ‘1’s in the SE is covered by the image,
  - The SE **hits** the image at the pixel position in question (the one on which the SE is centered).
  - This pixel is set to ‘1’ in the output image. Otherwise, it is set to ‘0’ in the output image.



Fit and Hit Examples



Position	*SE*	Fit	Hit
A	S_1	No	Yes
A	S_2	No	Yes
B	S_1	No	Yes
B	S_2	No	No
C	S_1	Yes	Yes
C	S_2	Yes	Yes
D	S_1	No	No
D	S_2	No	No

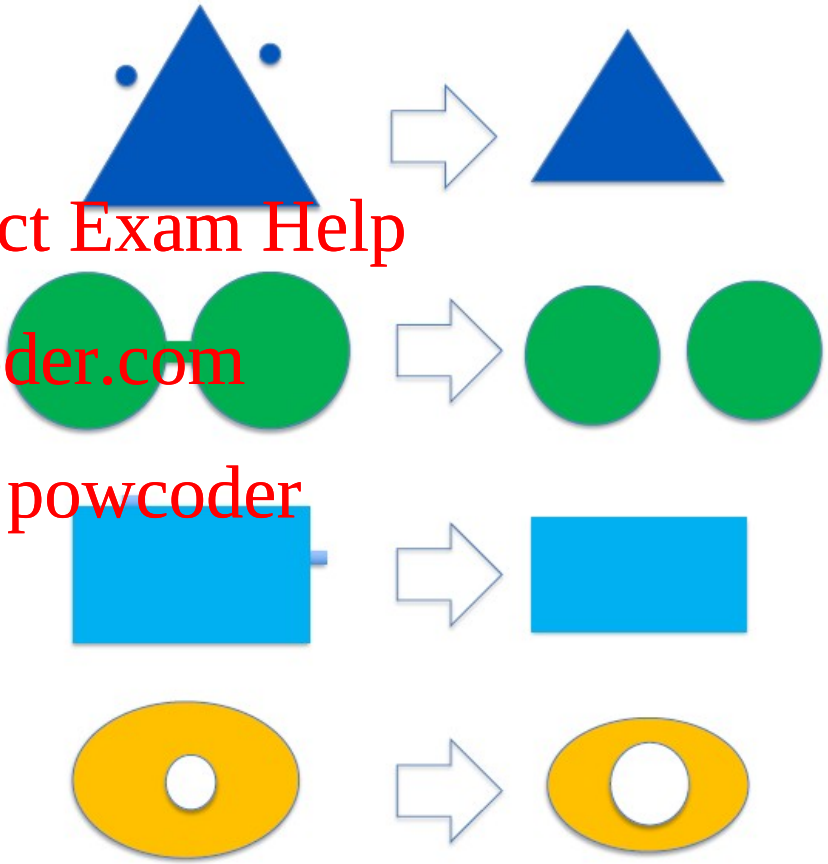
Erosion

- Erosion is the application of the Fit operation to every pixel of the image.
- The erosion of the set by a SE is defined as:

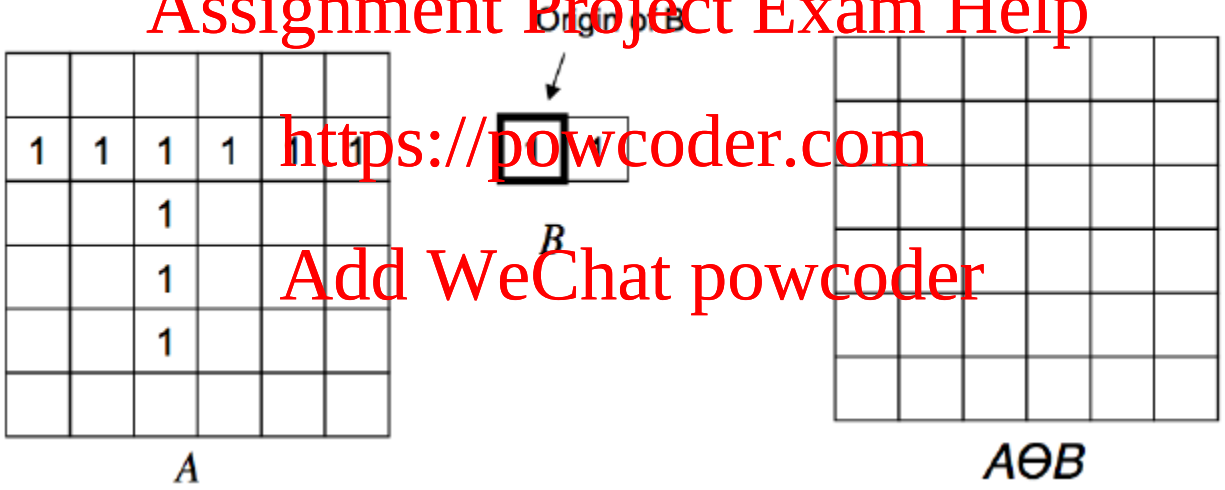
- The result is the set of all points such that translated by is contained in .
- Equivalently:

Applications of Erosion

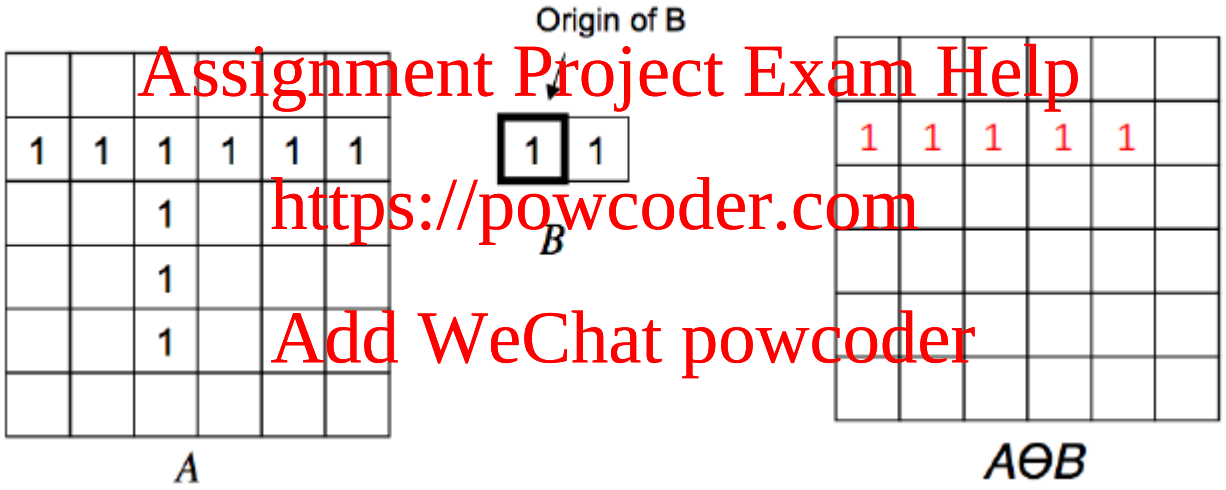
- Erosion mainly shrinks the object.
- It can be used for:
  - Shrinking objects
  - Removing small objects or noise
  - Removing bridges and branches
  - Removing protrusions
  - Enlarge holes



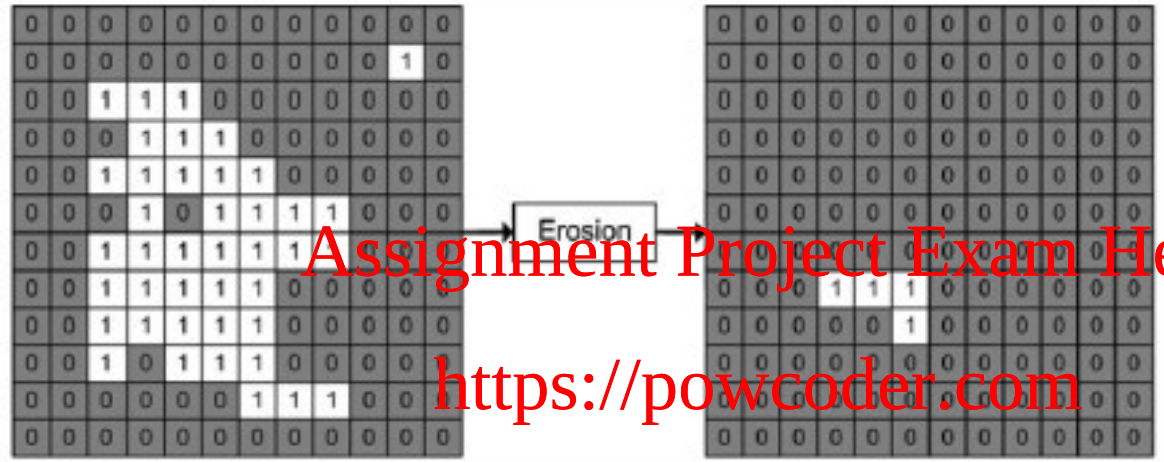
Erosion Example 1



Erosion Example 1



Erosion Example 2

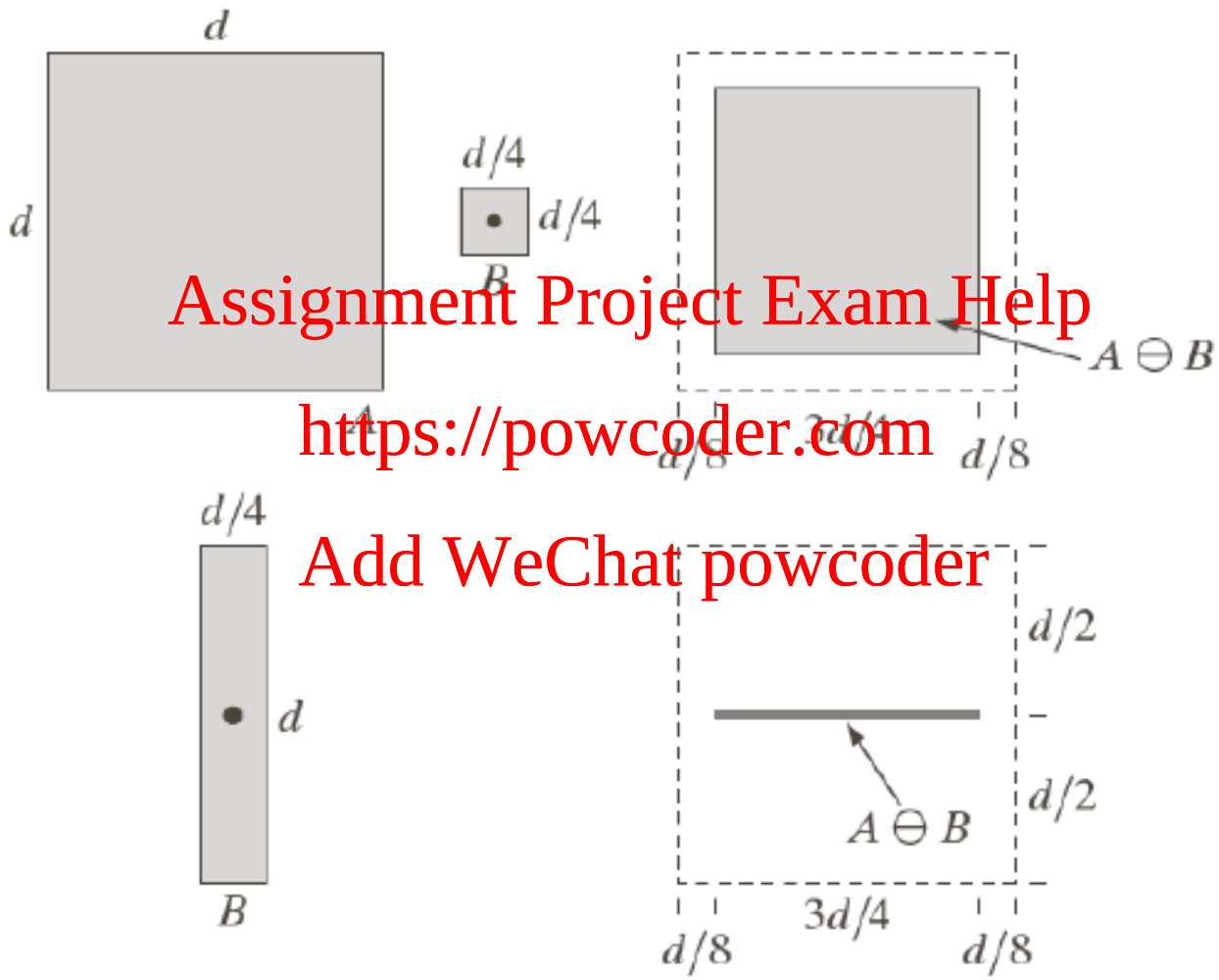


S_1

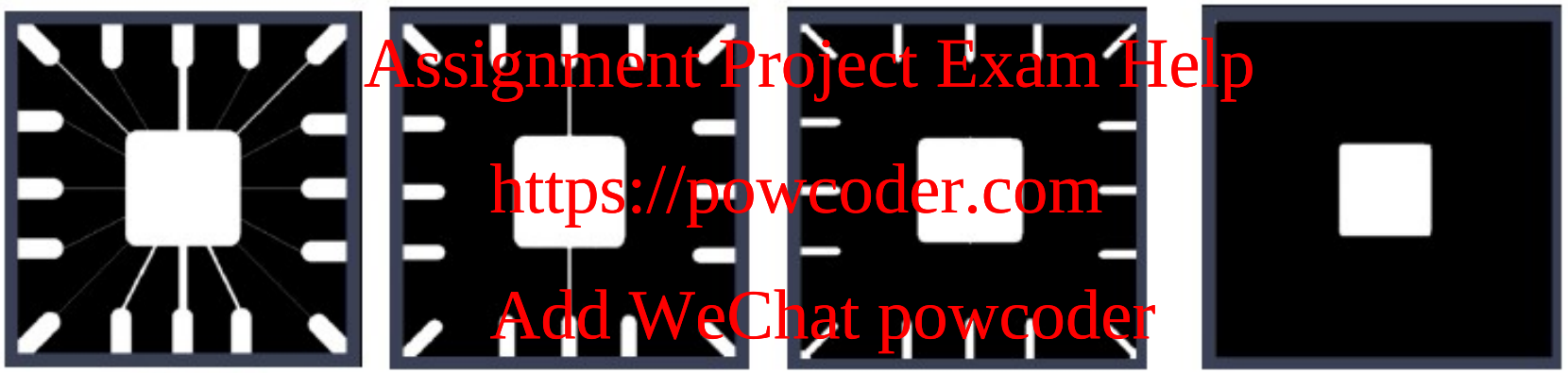
1	1	1
1	1	1
1	1	1

- Main object gets smaller. Only “core” of the subject remains.
- The size of this core depends on the size (and shape of the SE)
- The small objects disappears.

Erosion Example 3



Effect of Disk Size on Erosion



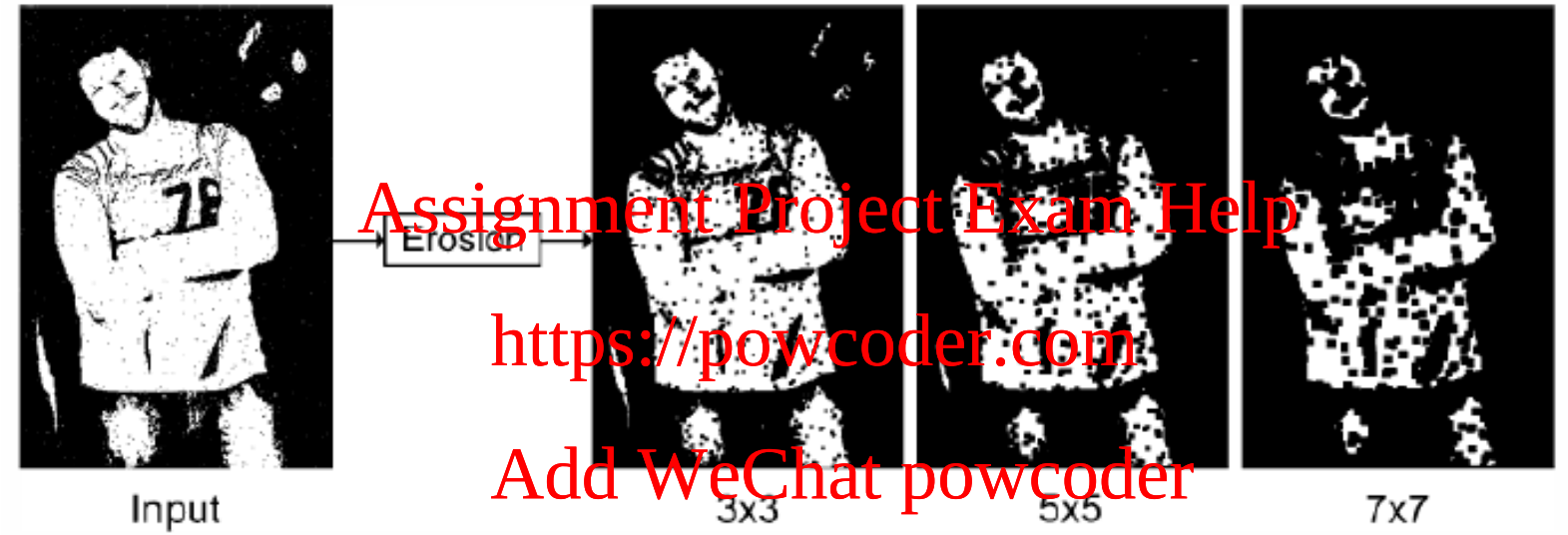
Original Image

Erosion with a Disk of radius 5

Radius 10

Radius 20

Erosion: Real Image Example



- Object becomes smaller and fractured.
- Small objects disappear.
- Effect more significant with larger SE.

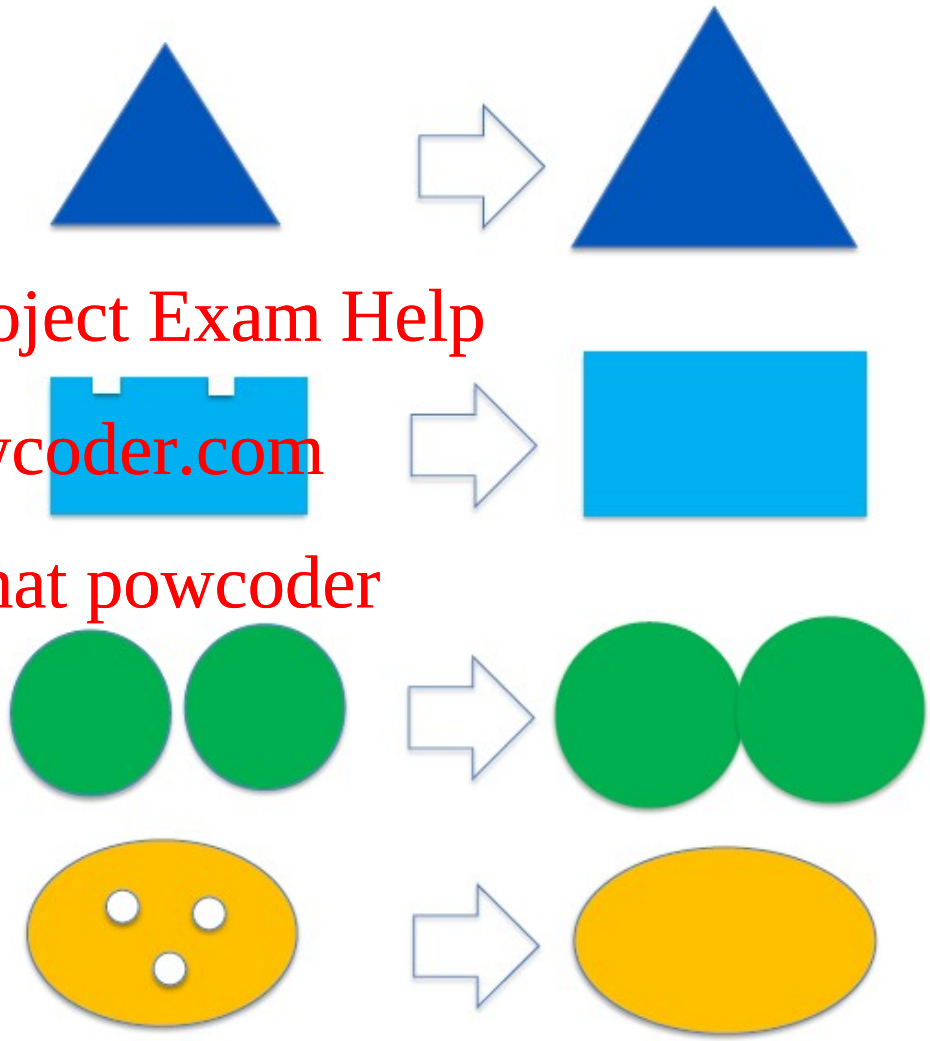
Dilation

- Dilation is the application of Hit operation to every pixel of the image.
- The dilation of a set by a SE is defined as:

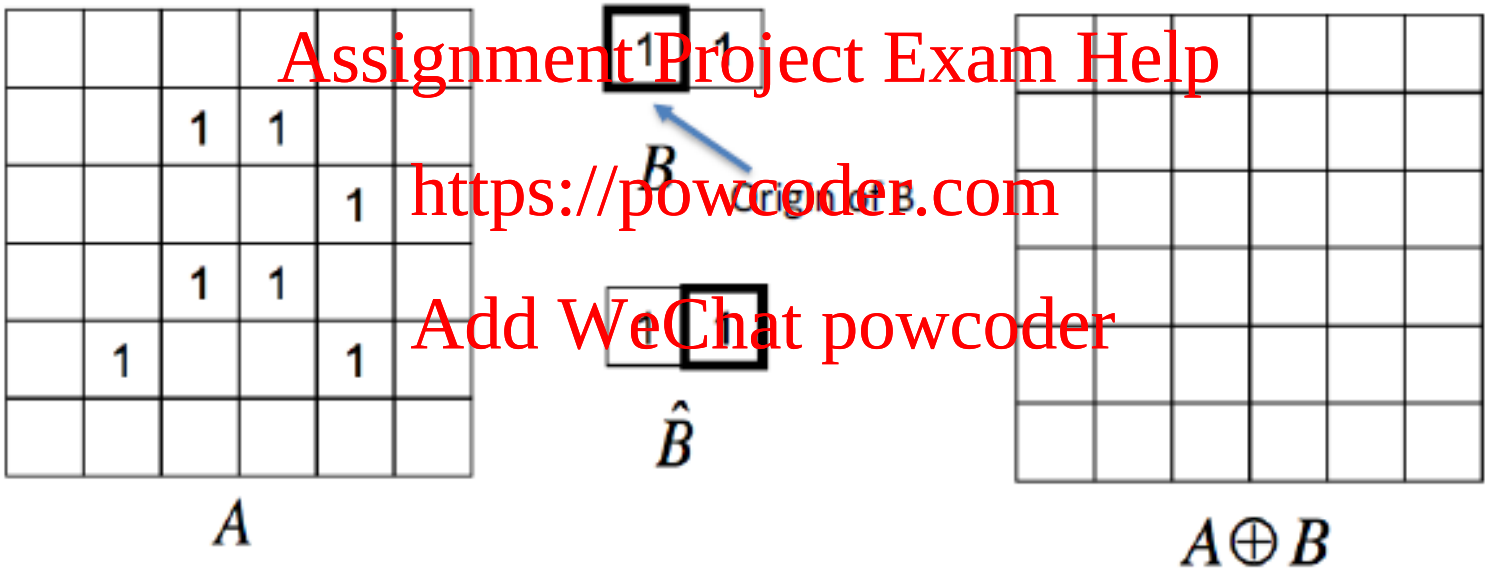
- The result is the set of all points such that the reflected translated overlap with at least one element.
- Equivalently:

Applications of Dilation

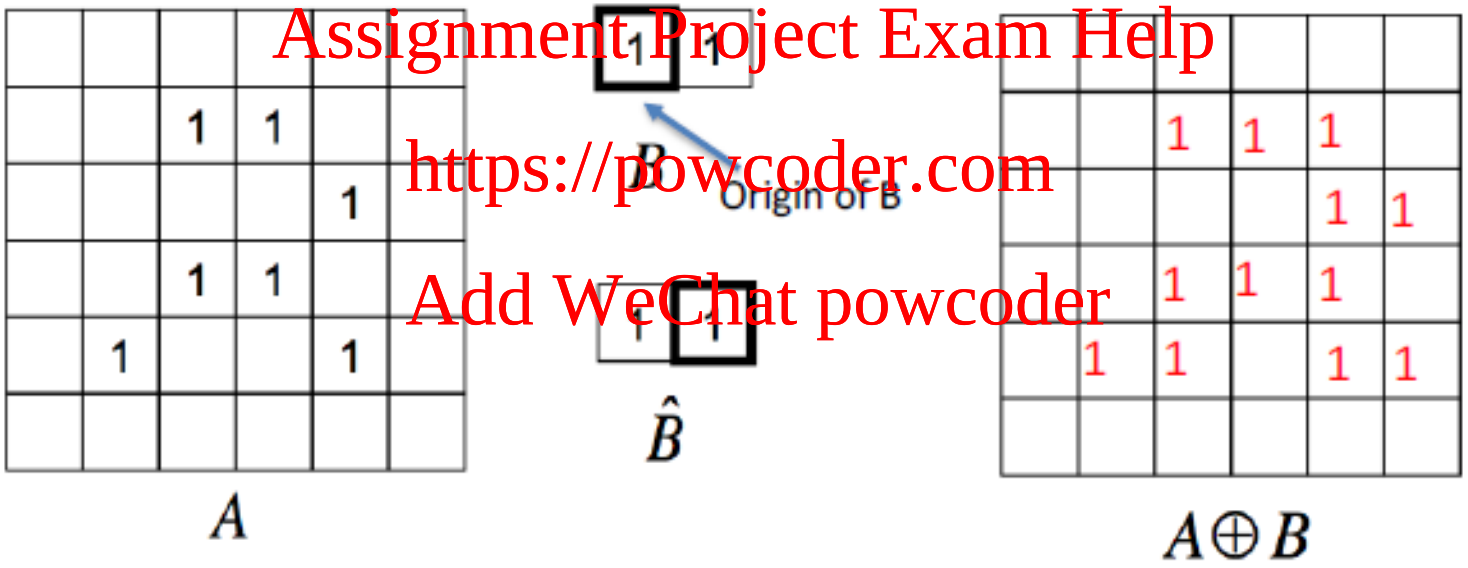
- Erosion mainly expands the object.
- It can be used for:
  - Growing objects
  - Repairing intrustions
  - Filling gaps
  - Filling holes



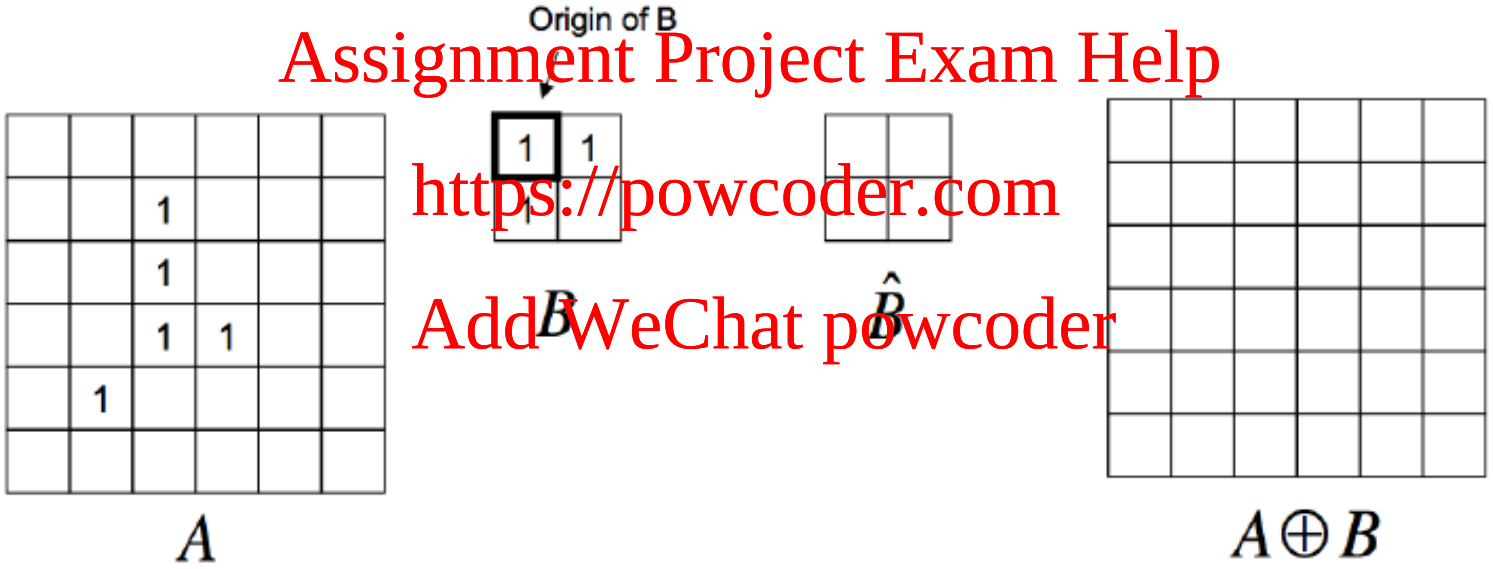
Dilation Example 1



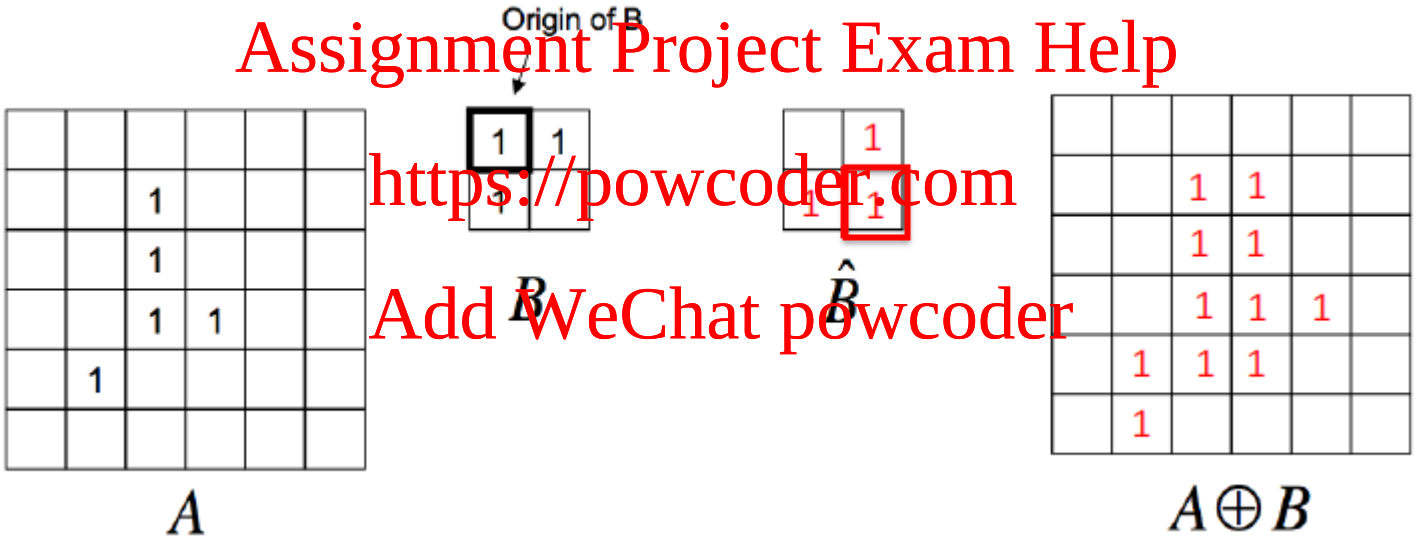
Dilation Example 1



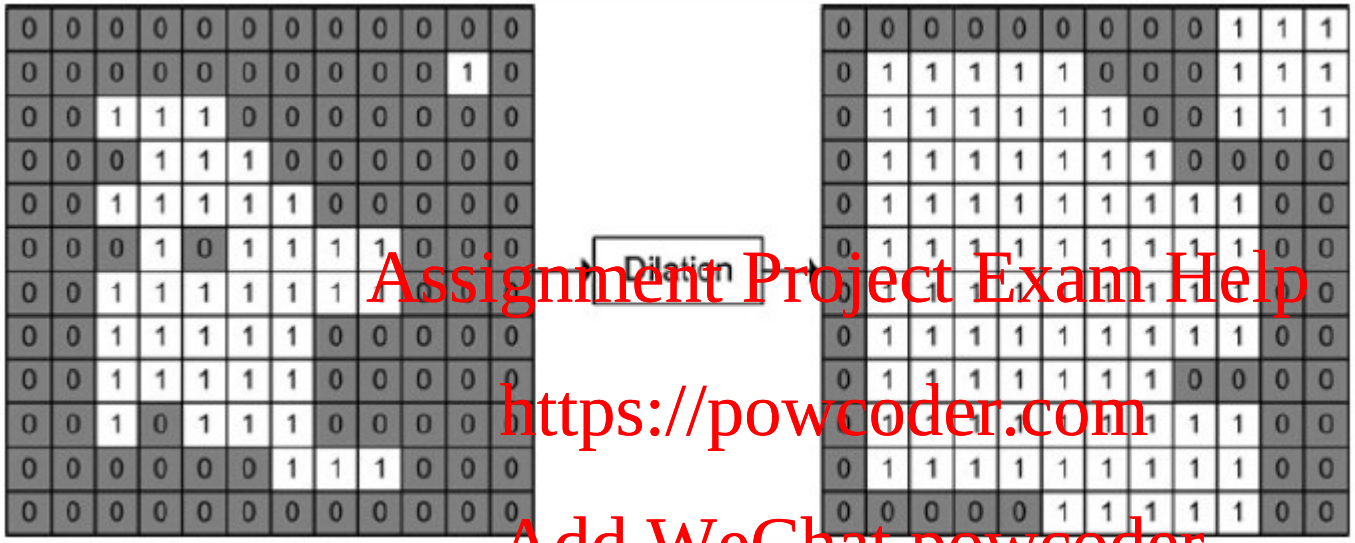
Dilation Example 2

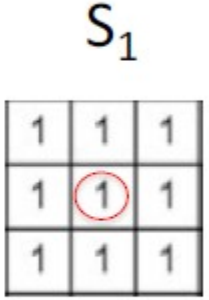


Dilation Example 2



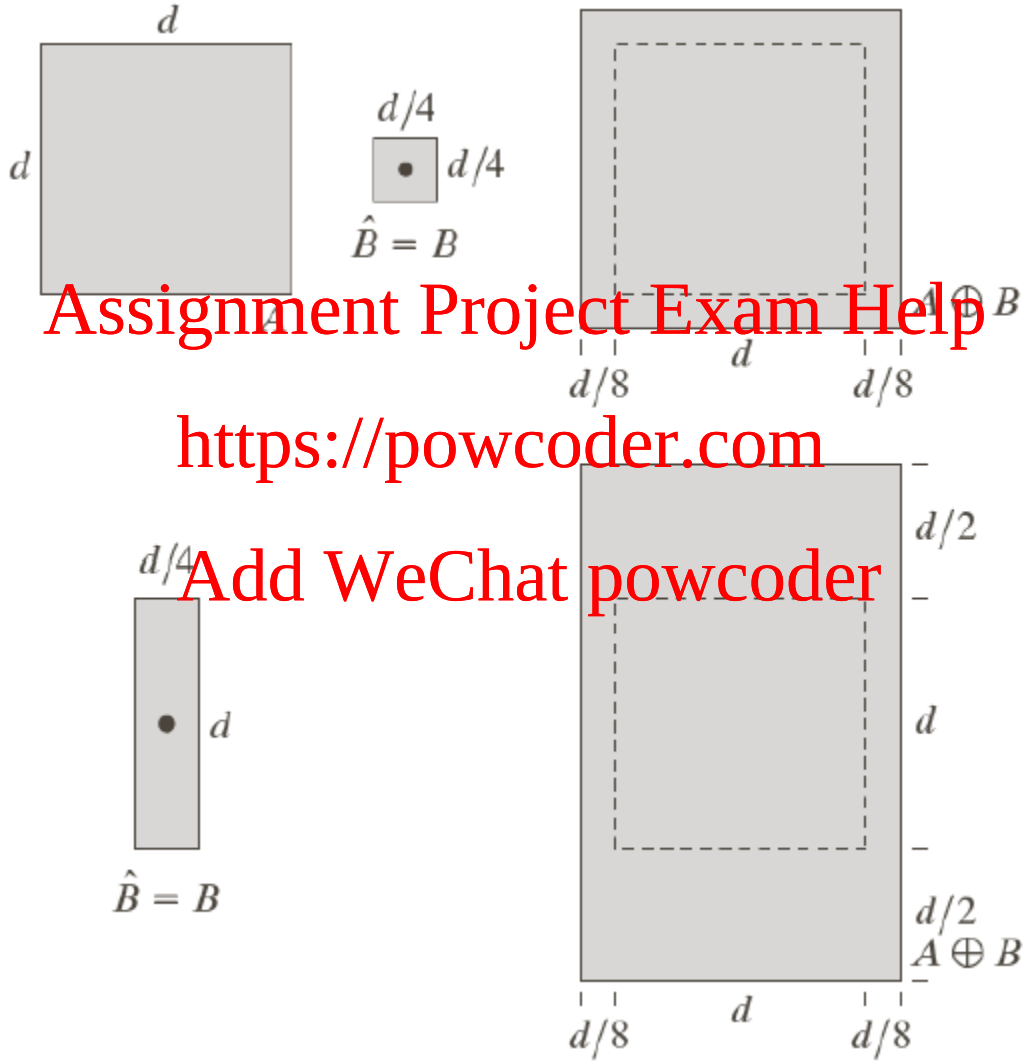
Dilation Example 3



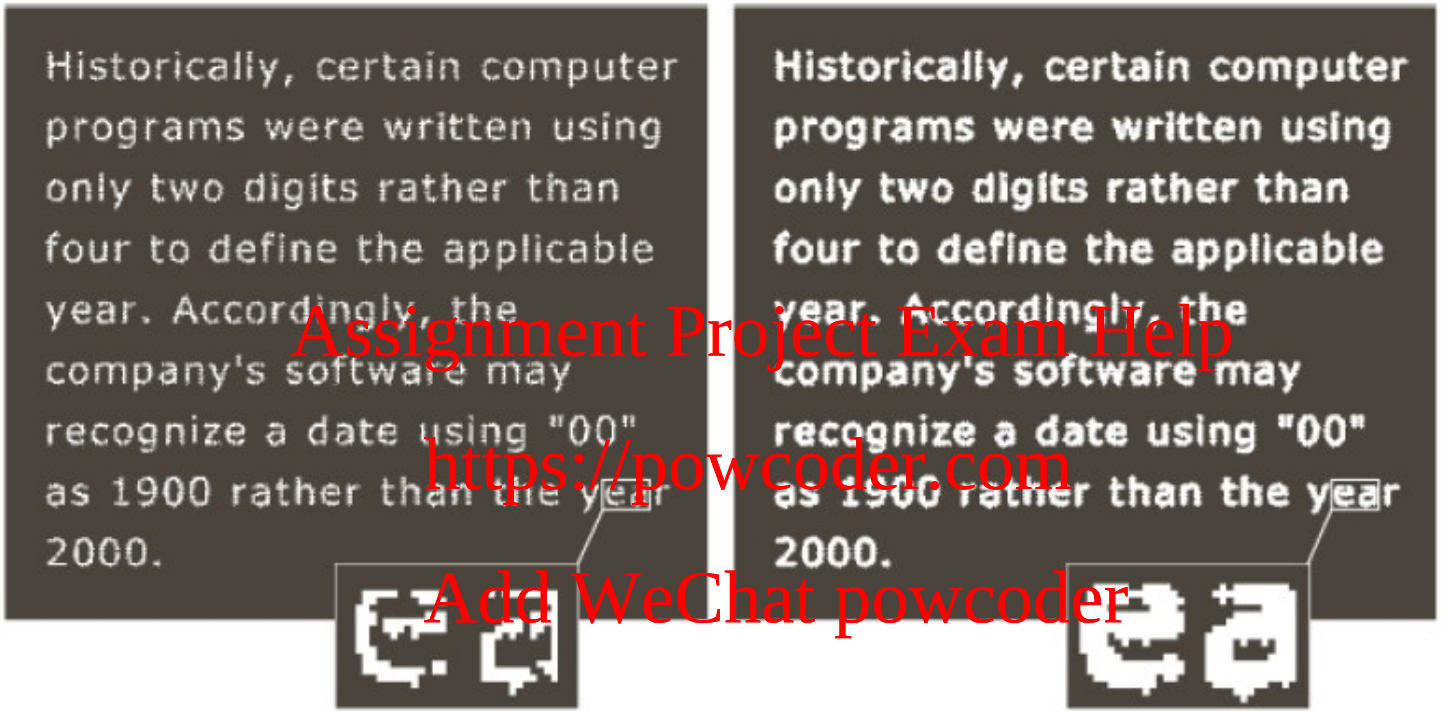


- Object gets bigger.
- The hole is filled.

Dilation Example 4



Dilation Example: Text Image



0	1	0
1	1	1
0	1	0

Dilation bridges gaps.

Dilation Example: Real Image

- Main object is becoming bigger
- The hole inside the person are filled
- Small object in the background are also enlarged

Duality of Dilation and Erosion

- Erosion and dilation are dual operations with respect to set complementation and reflection:

- Also,

- Interpretation when SE is symmetric:

  - First equation: The complement of the erosion operation of an image is the dilation of its background
  - Second equation: The complement of the dilation operation of an image is the erosion of its background.

Duality

Proof:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

$$= \{z \mid (B)_z \cap A^c = \varnothing\}$$

If set $(B)_z$ is contained in A, then the intersection of $(B)_z$ with the complement of A is empty

$$(A \ominus B)^c = \{z \mid (B)_z \cap A^c = \varnothing\}^c$$

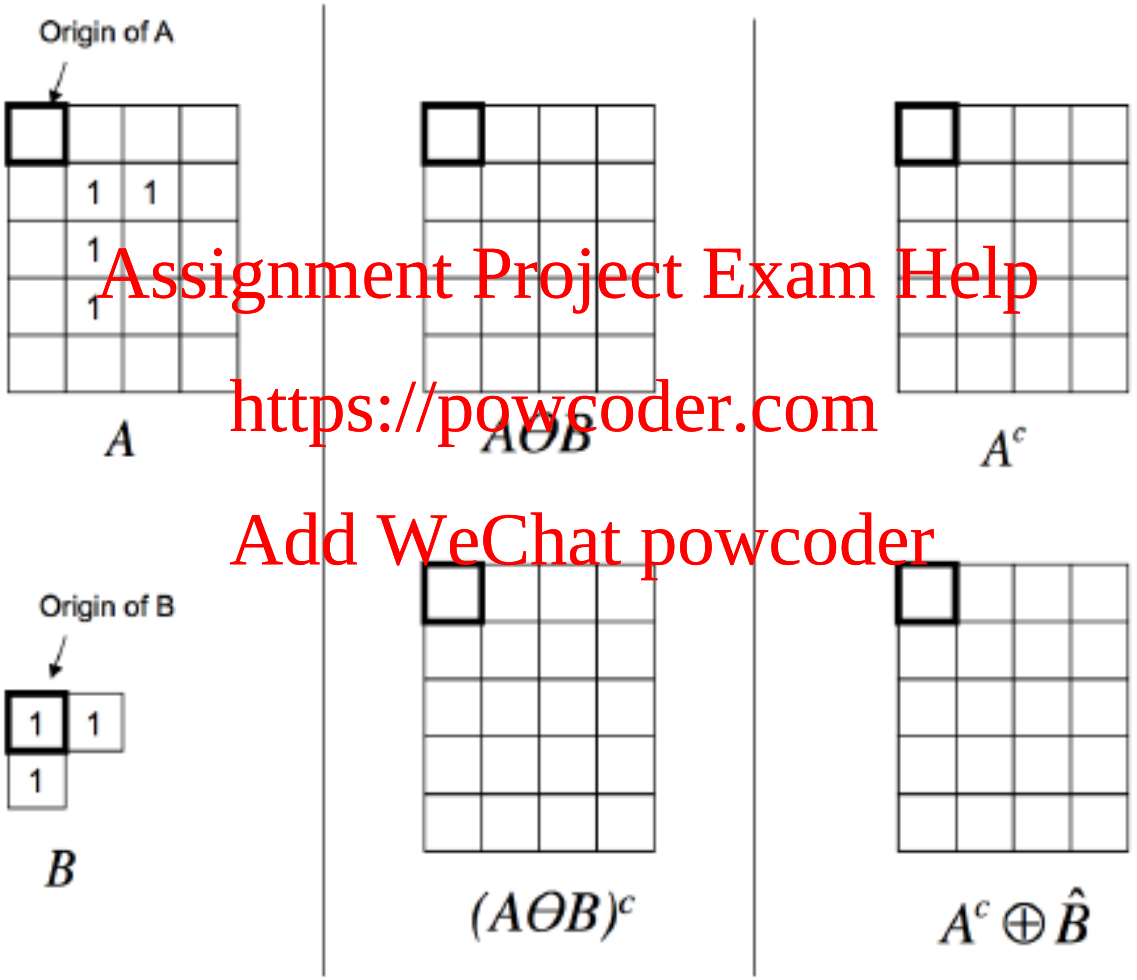
Complement both sides

$$= \{z \mid (B)_z \cap A^c \neq \varnothing\}$$

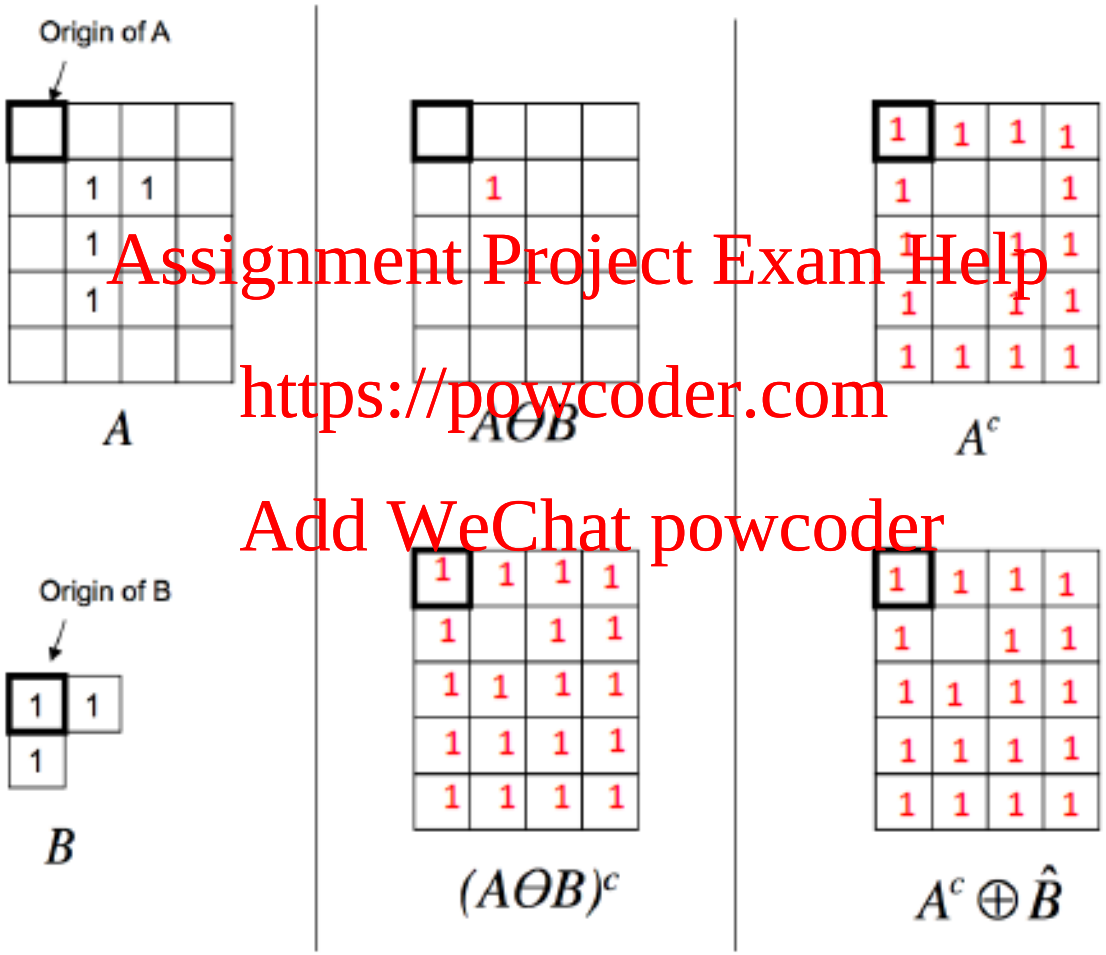
$$= A^c \oplus \hat{B}$$

By the definition of dilation

Example



Example



Compound Operations

- More interesting morphological operations can be performed by combining erosions and dilations in order to reduce shrinking or thickening.
- The most widely used of these compound operations are:
  - Opening: Erosion followed by Dilation
  - Closing: Dilation followed by Erosion

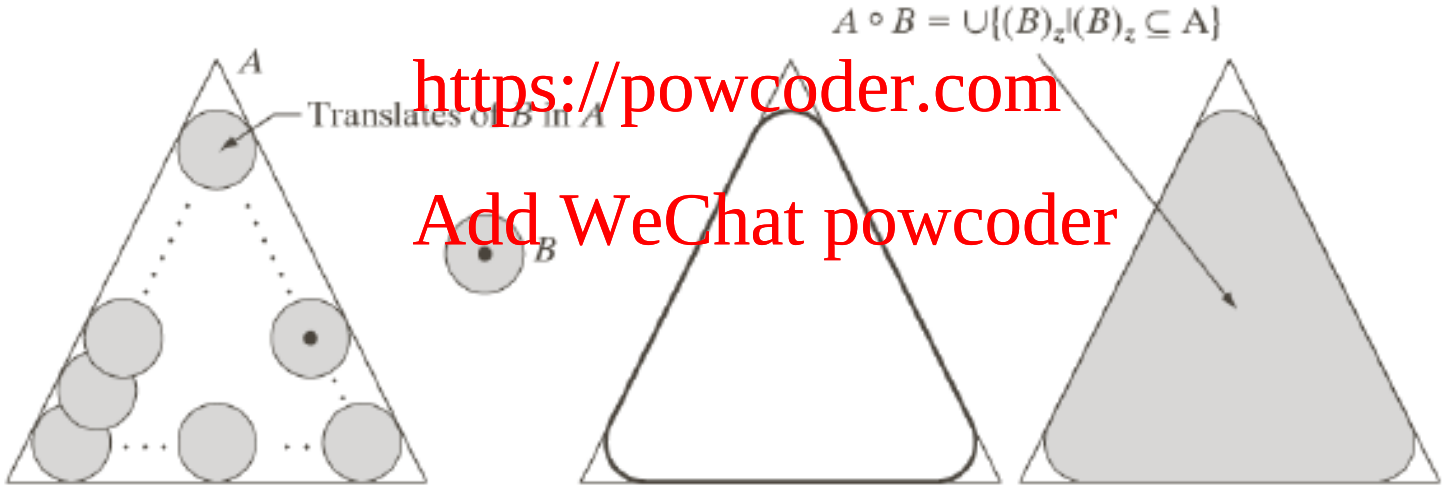
Opening

- The opening of set by structuring element is defined as

which is an erosion of by followed by a dilation of the result by .

Opening

- Geometric interpretation: The opening of by is the union of all translations of so that is fitted entirely within .



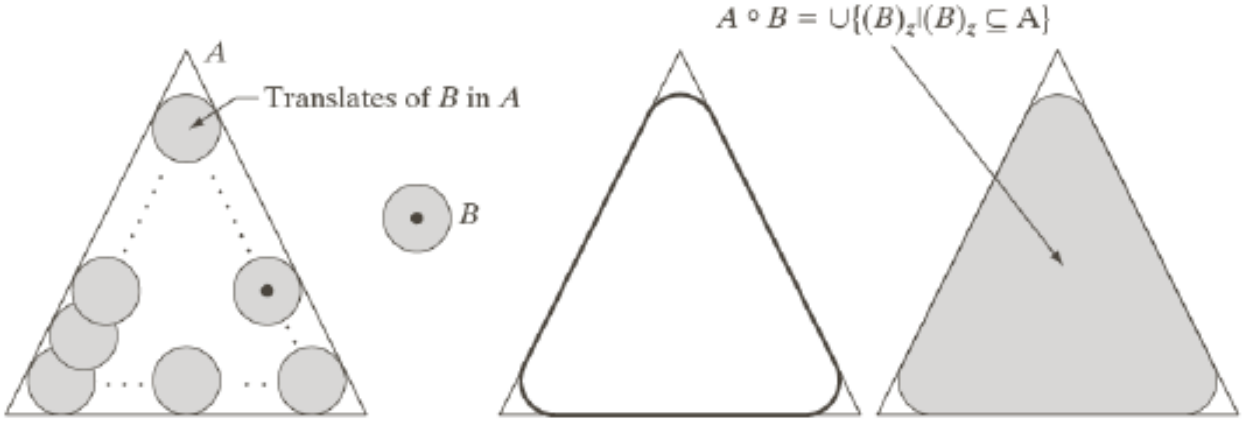
$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

Opening

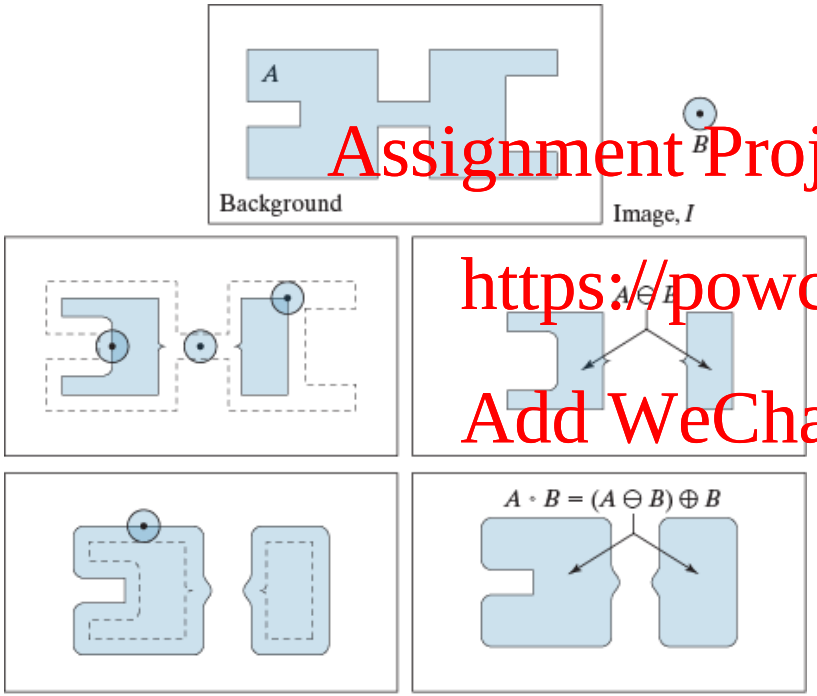
Notice the difference with the simple erosion:

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \qquad A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

If B translated by z lies inside A, then the result contains the whole set of points covered by the SE and not only its center as it is done in the erosion.



Opening Example 1

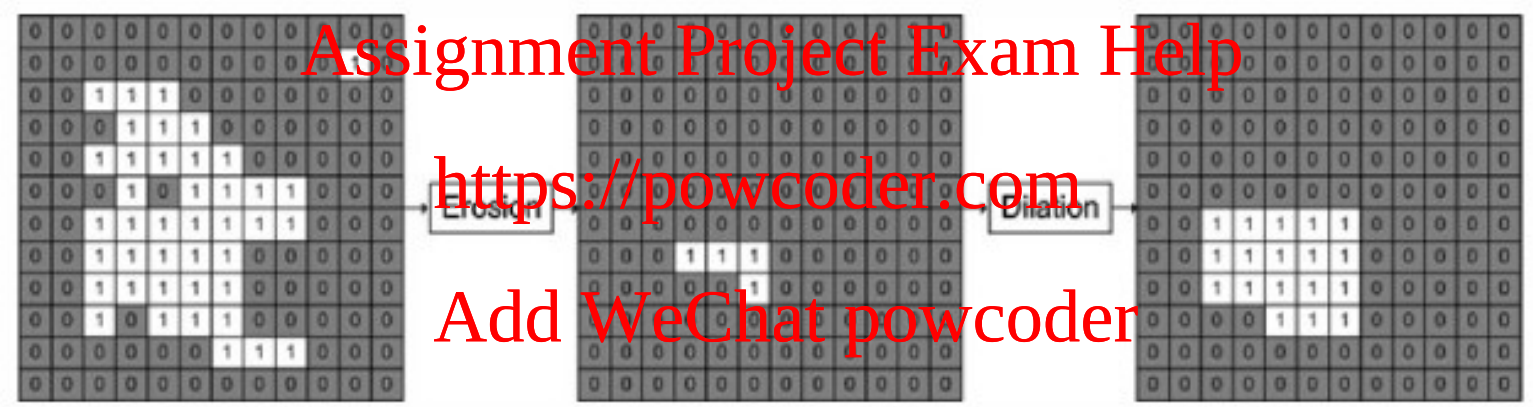


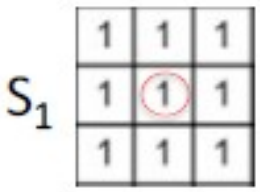
Erosion: elements where the disk can not fit are eliminated.

Opening: outward corners are rounded.

Opening Example 2

- Only a compact version of the object remains





Opening Example: Real Image

Fig. 6.12 Opening performed using a 7×7 box-shaped structuring element

- Most noisy objects are removed
- The object preserves its original size

Closing

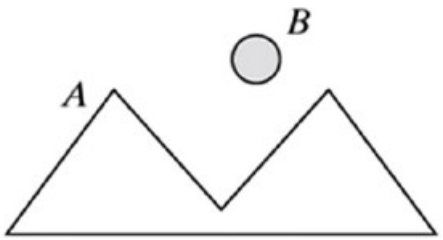
- The closing of set by the SE is defined as:

which is a dilation of by followed by an erosion of the result by .

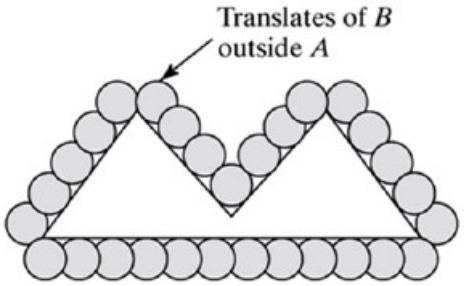
Closing

- The closing is the complement of the union of all translations of B that do not overlap with :

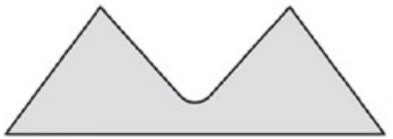
- Geometric Interpretation: Closing results in an area that we cannot paint using a brush with footprint B, when no part of the brush is allowed to overlap with the region .
- Effect: Smoothing of the boundary from the outside.



Binary image A and structuring element B



Translations of B that do not overlap A



The closing of A by B is shown shaded

Duality

- Opening and closing are dual operations.

Erosion-Dilation duality

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

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

Opening-Closing duality

$$(A \bullet B)^c = A^c \circ \hat{B}$$

$$(A \circ B)^c = A^c \bullet \hat{B}$$

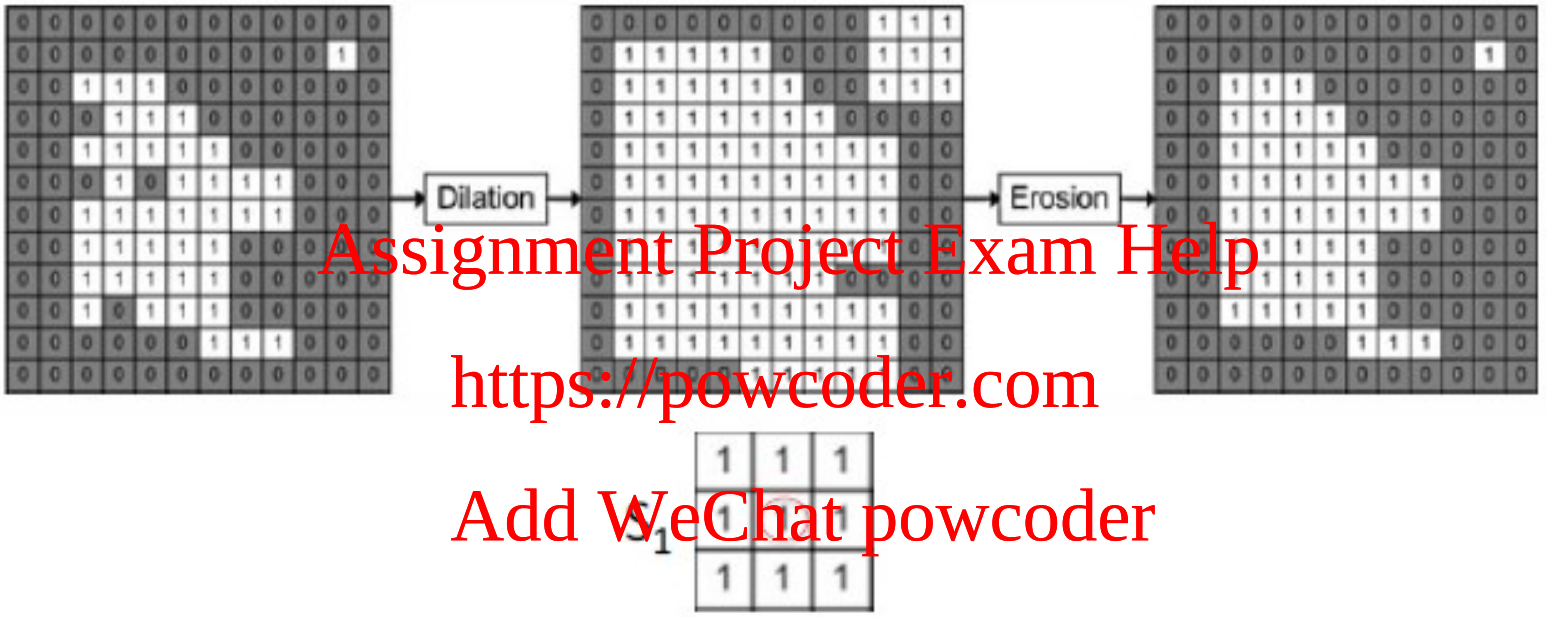
Properties of Opening and Closing

- Opening

- Closing

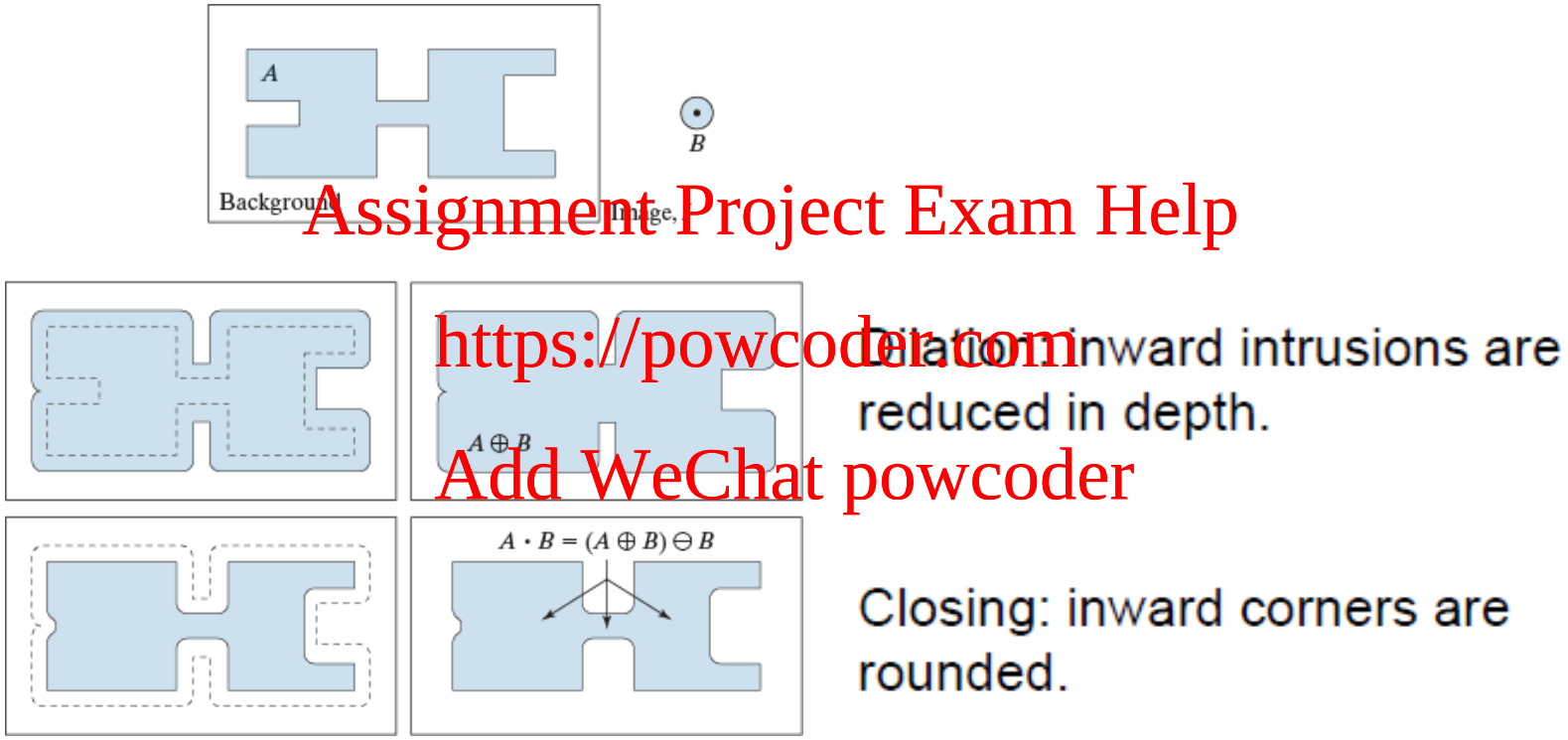
- The last properties, in each case, indicate that multiple openings or closings have no effect after the first application of the operator (idempotent).

Closing Example 1



- Holes and indentations are filled
- The object preserves its size

Closing Example 2



Dilation: inward intrusions are reduced in depth.

Closing: inward corners are rounded.

Closing Example: Real Image

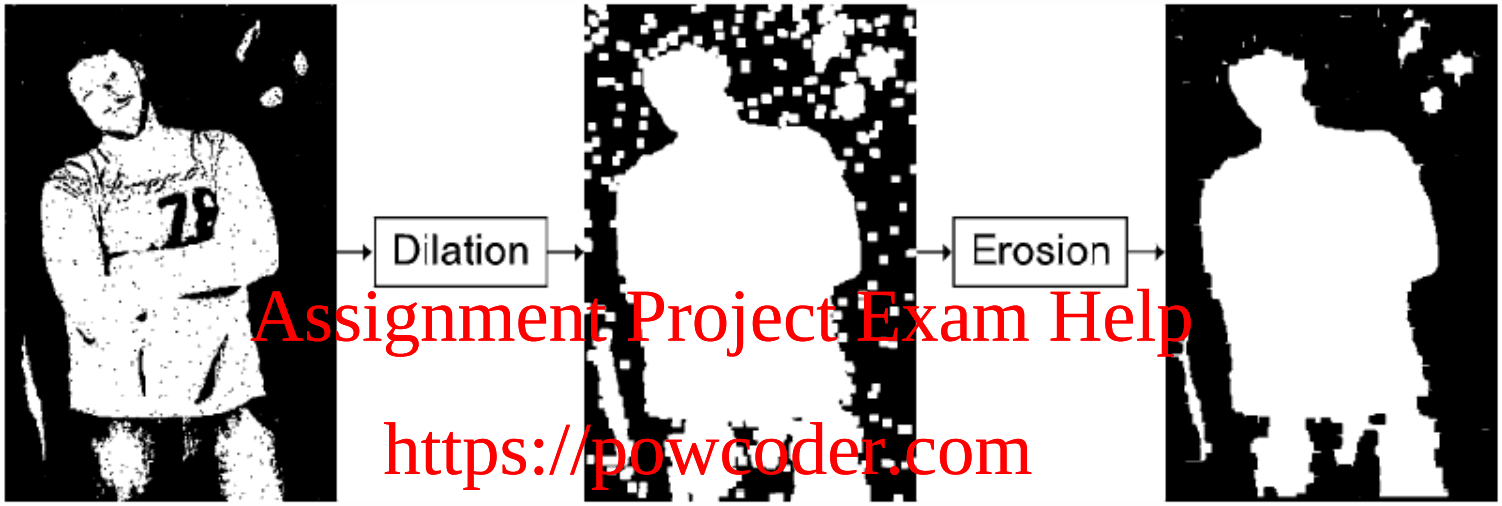


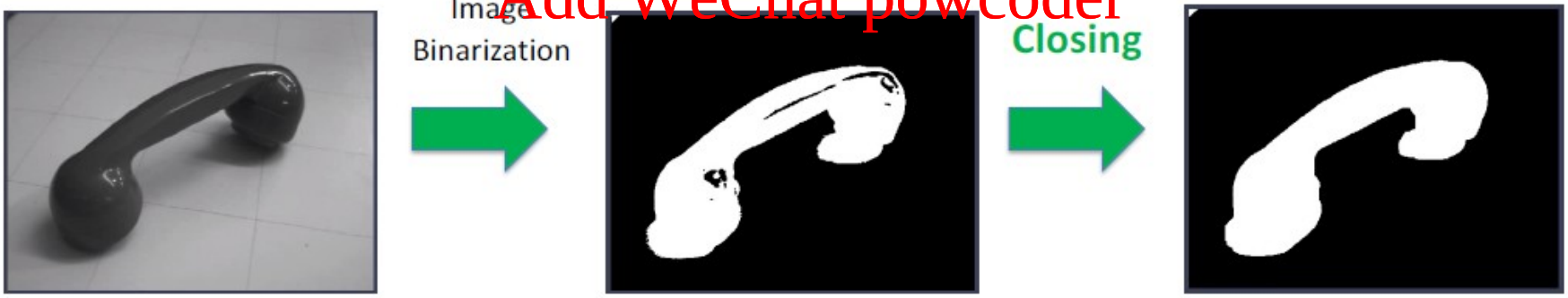
Fig. 6.10 Closing performed using two 7 × 7 box-shaped structuring elements

- Most internal holes are filled while the human object preserves its original size
- Note: the small objects in the background have not been deleted

Closing Example: Segmentation

- A simple segmentation of foreground object from a grayscale image

1. Threshold
2. Closing with disc of size 20



Closing and then Opening

- The closing was performed using 7x7 box-shaped SE
- The opening was performed using a 15x15 box-shaped SE

Python Examples: Erosion

- These examples are from the package demo6.zip

```
from skimage.morphology import binary_erosion

selem = np.ones((4, 4))
eroded = binary_erosion(orig_j, selem)
plot_comparison(orig_j, eroded, 'erosion')
```


Python Example: Dilation

```
from skimage.morphology import binary_dilation

selem = np.ones((10,10))
dilated = binary_dilation(orig_j, selem)
plot_comparison(orig_j, dilated, 'dilation')
```

Python Example: Opening

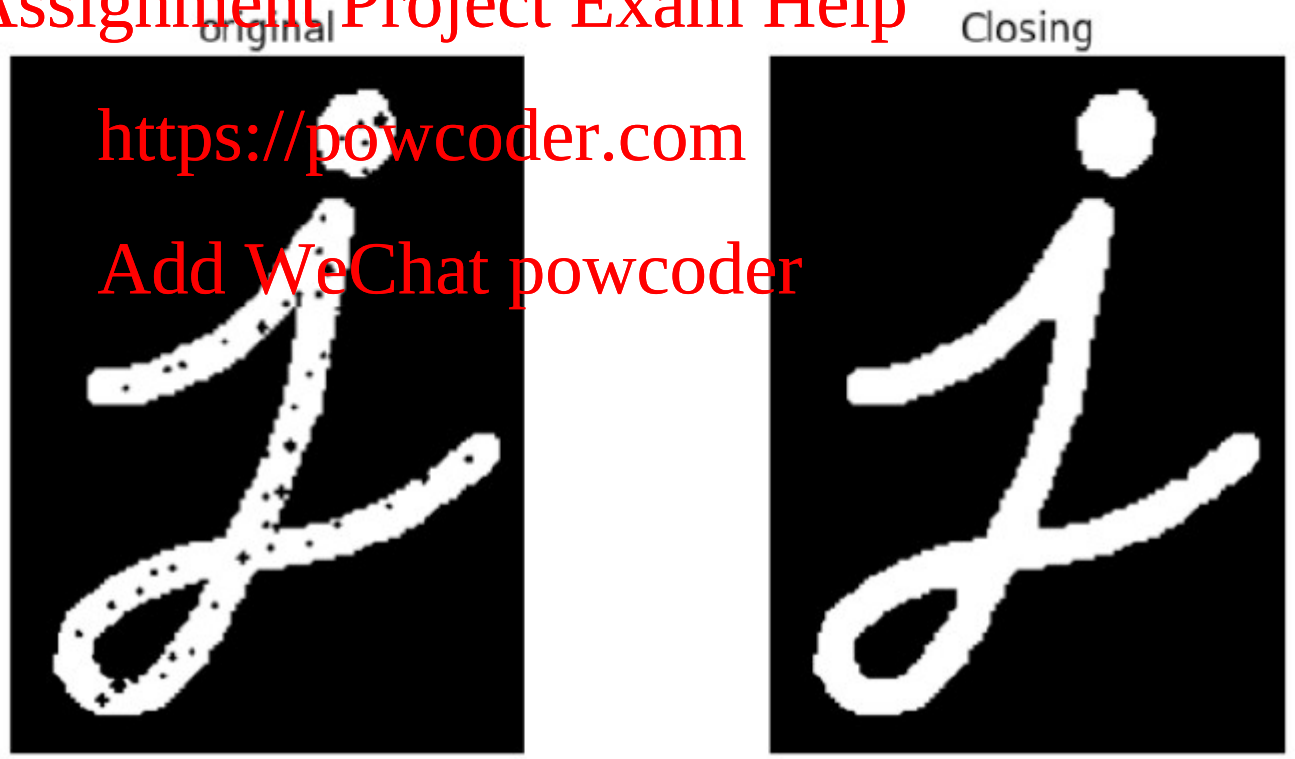
```
from skimage.morphology import binary_opening

orig_j1 = io.imread("binary_j1.png")

selem = np.ones((8,8))
opened = binary_opening(orig_j1, selem)
plot_comparison(orig_j1, opened, 'opening')
```


Opening is for removing noise.

Python Example: Closing

```
from skimage.morphology import binary_closing

orig_j2 = io.imread("binary_j2.png")

selem = np.ones((8,8))
closed = binary_closing(orig_j2, selem)
plot_comparison(orig_j2, closed, 'Closing')
```

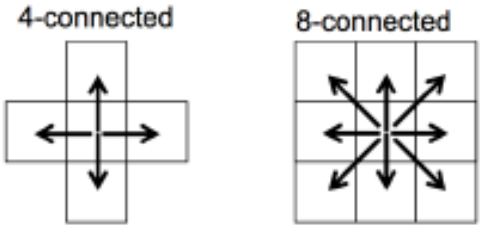

Closing is useful for removing holes inside foreground objects.



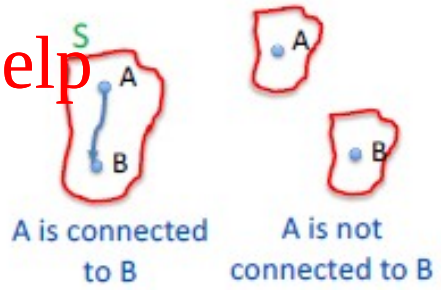
7.2 Connected Components and Labelling

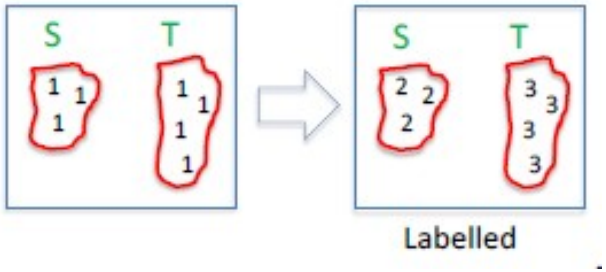
Connected Components and Labelling

- Adjacency
  - 4-adjacent
  - 8-adjacent

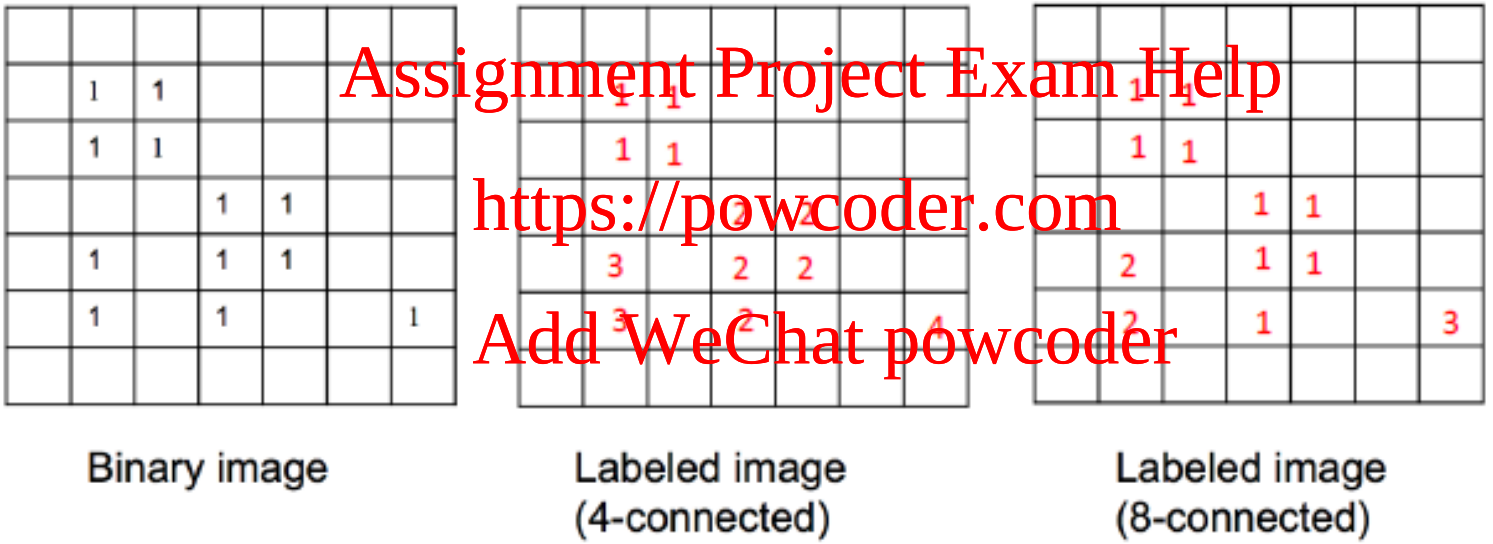


- Two pixels are connected in if there is a path between them consisting entirely of pixels in .
- is a (4- or 8-) connected component (blob) if there exists a path between every pair of pixels
- **Labelling** is the process of assigning the same label number to every pixe in a connected component





Labelling Example

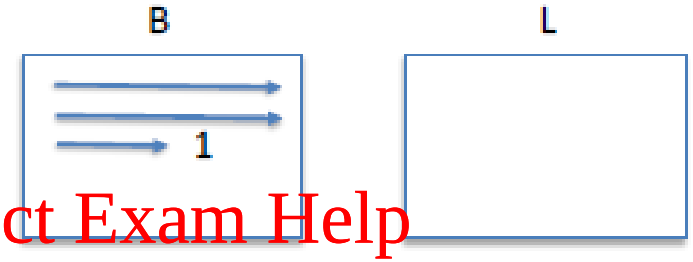


A Fast Labelling Algorithm

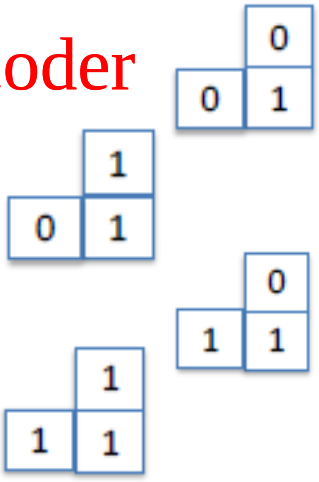
- One pass through image to assign temporary labels and record equivalent labels
- Second pass to replace temporary labels with final labels
- Let
  - B(r,c) is the input binary image
  - L(r,c) is the output image of labels

A Fast Labelling Algorithm

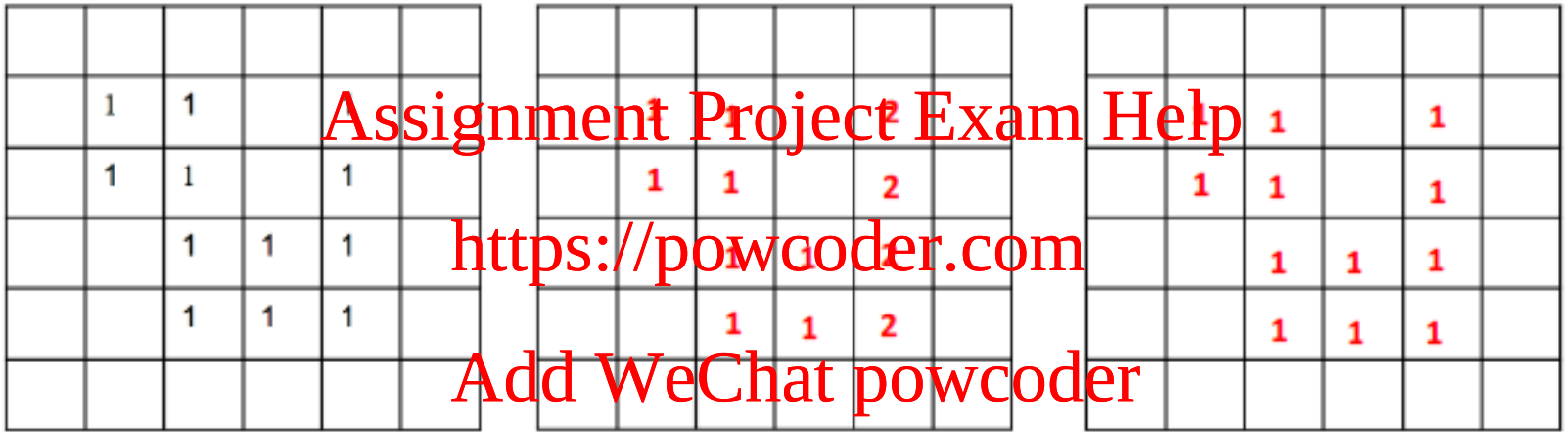
Here is the pseudo-code for the algorithm labelling 4-connected components:




```
NUMLABEL = 1
for r = 1 to MAXROW {
        for c=1 to MAXCOL {
        if B(r,c) == 0 then
        L(r,c) = 0; % if pixel not white, assign no label
        else
        if B(r-1,c)==0 && B(r,c-1)==0
        L(r,c) = NUMLABEL++;
        else B(r-1,c)==1 && B(r,c-1)==0
        L(r,c) = L(r-1,c)
        else B(r,c-1)==1 && B(r,c-1)==0
        L(r,c) = L(r,c-1)
        else B(r-1,c)==1 && B(r,c-1)==1
        L(r,c) = L(r-1,c)
        record L(r-1,c) and L(r,c-1) as equivalent labels
```

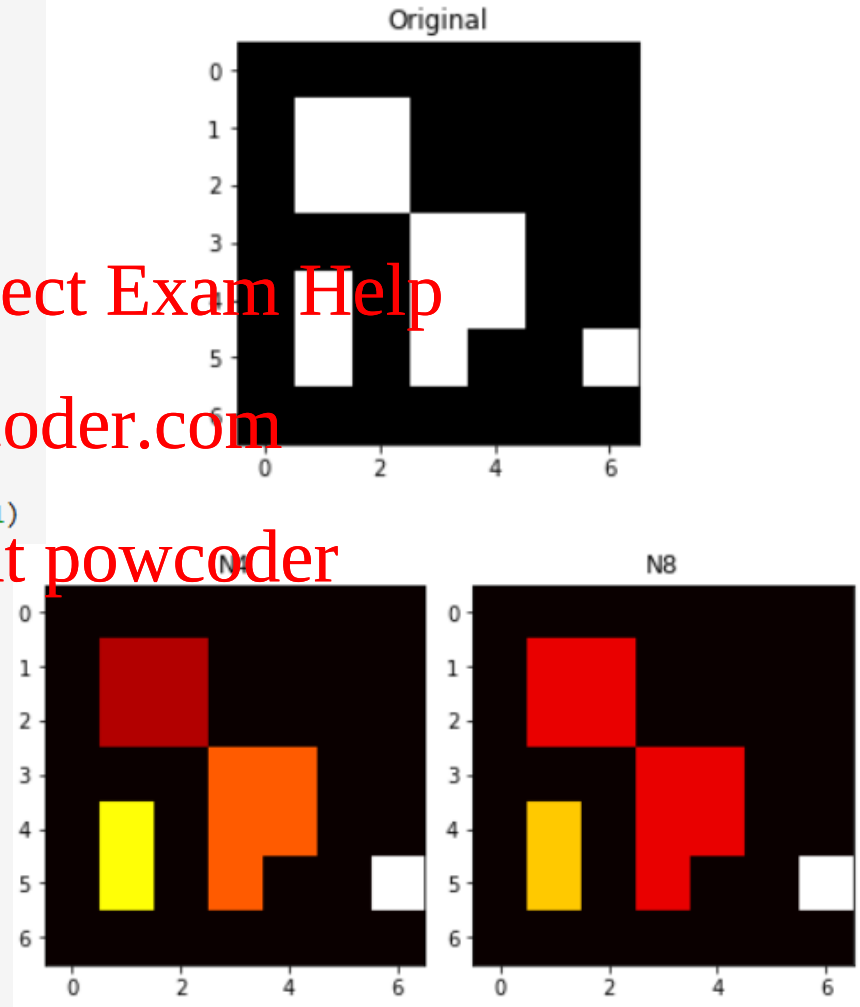
Example



Equivalence	Temp
1	1,2

Python Implementation

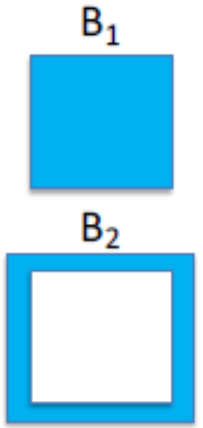
```
%matplotlib inline
from skimage import measure
from skimage import filters
import matplotlib.pyplot as plt
import numpy as np

blobs = np.array([[0,0,0,0,0,0,0],
                  [0,1,1,0,0,0,0],
                  [0,1,1,0,0,0,0],
                  [0,0,0,1,1,0,0],
                  [0,1,0,1,1,0,0],
                  [0,1,0,1,0,0,1],
                  [0,0,0,0,0,0,0]])

N8_labels = measure.label(blobs)
N4_labels = measure.label(blobs, background=0, connectivity=1)

plt.figure(figsize=(9, 3.5))
ax1=plt.subplot(131)
plt.imshow(blobs, cmap='gray')
ax1.set_title('Original')
ax2=plt.subplot(132)
plt.imshow(N4_labels, cmap='hot')
ax2.set_title('N4')
ax3=plt.subplot(133)
plt.imshow(N8_labels, cmap='hot')
ax3.set_title('N8')

plt.tight_layout()
plt.show()
```



7.3 Morphological Algorithms

Hit-Or-Miss Transform

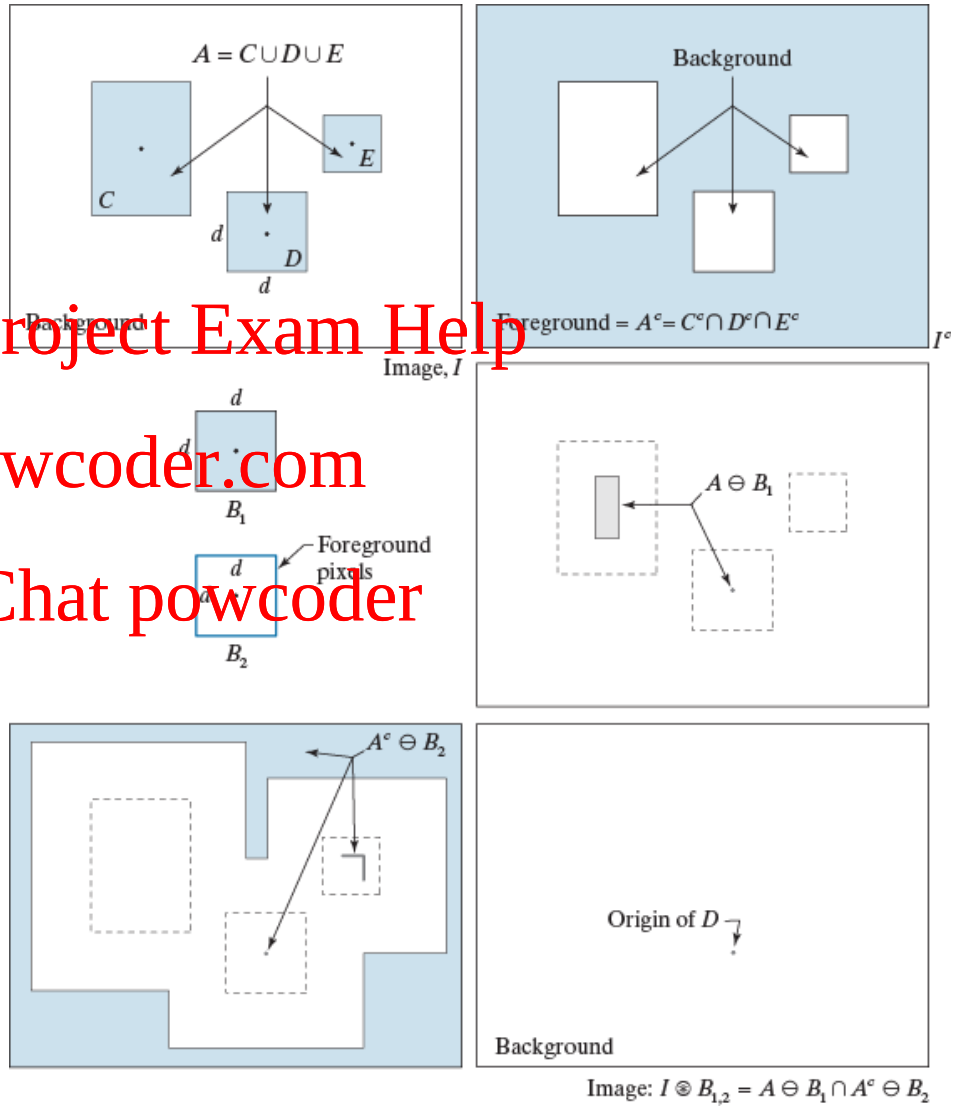
- A method to find the location of a shape in an image
- Define an SE with the same size as , called .
- gives all places where fits in
- But fits in any sufficiently large shape.
- Add one more criterion to search for : Need the background to match , which contains the boundary of .
- The intersection contains points that match both criteria



$$A \circledast B_{1,2} = (A \ominus B_1) \cap (A^C \ominus B_2)$$

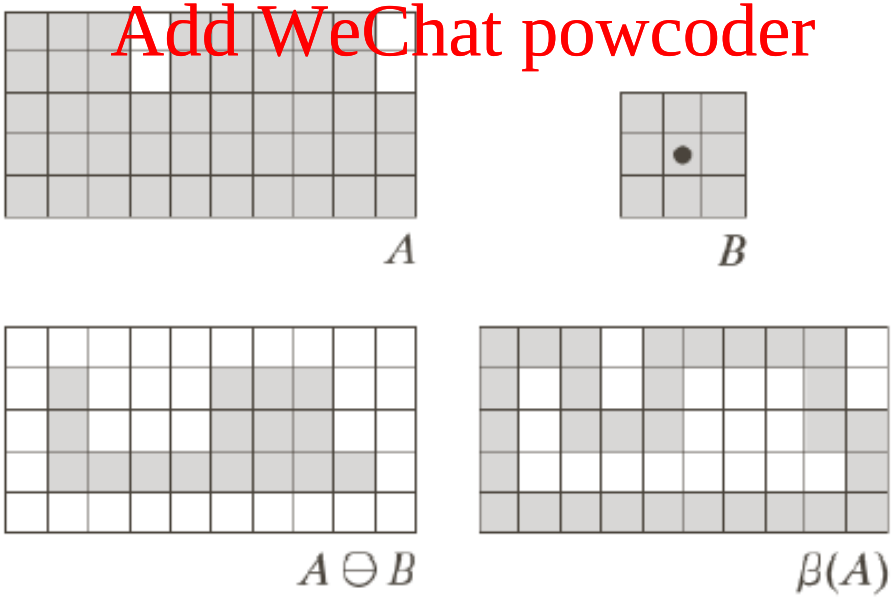
Hit-Or-Miss Transform

- contains the origin of , but also part of , as is larger than
- contains the origin of , but also part of , as is smaller than .
- Intersection gives only the origin of .



Boundary Extraction

- To find the boundary of a set , erode it by a small structuring element .
- Then take the set difference between and its erosion:



Boundary Extraction Example

- The boundary is one pixel thick due to the 3x3 SE. Other SE would result in thicker boundaries.

Original Image Extracted Boundary

Region Filling

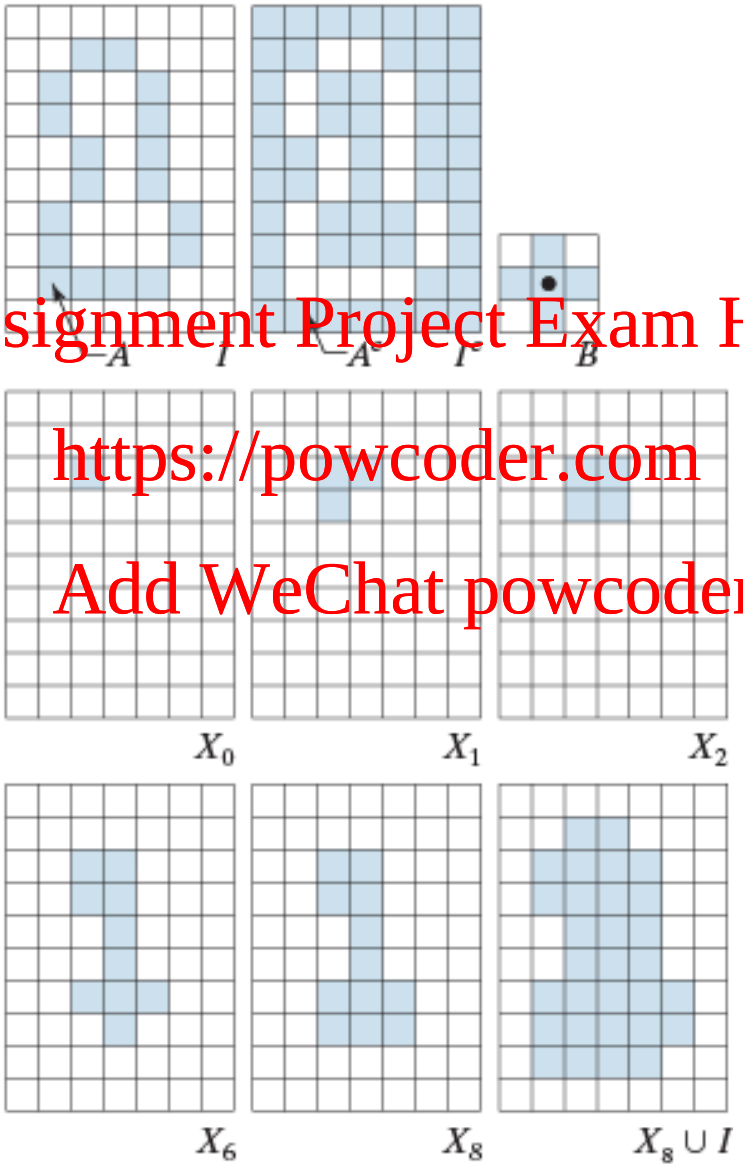
- Given a pixel inside a boundary, region filling attempts to fill the area surrounded by that boundary with 1s.

Region Filling

- Let to be an image containing boundaries.
- Form a set with zeros everywhere, except at the pixel that is confirmed to be a hole.
- Then do the following two operations iteratively:
  - Dilation with a 3x3 cross-shaped SE.
  - Intersect with
- Mathematically,

- The algorithm terminates when .
- contains all the filled holes and their boundaries.

Region Filling Example 1



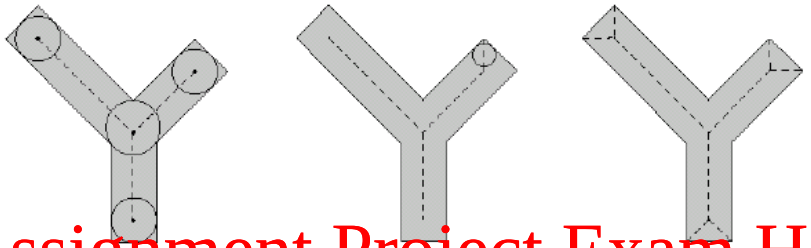
Region Filling Example 2

Original image with white dots required to start the region-filling algorithm

Output Image

Skeletonization

- Skeleton is a concise representation of shape.



- Set of all points that are equally distant from two closest points of the object boundary

- Equivalently, the union of all maximal disk centers that are contained in the object

- Analogy:
  - Start a fire at the boundary, let it burn inward (by repeated erosions)
  - Points where fire is quenched are the skeleton

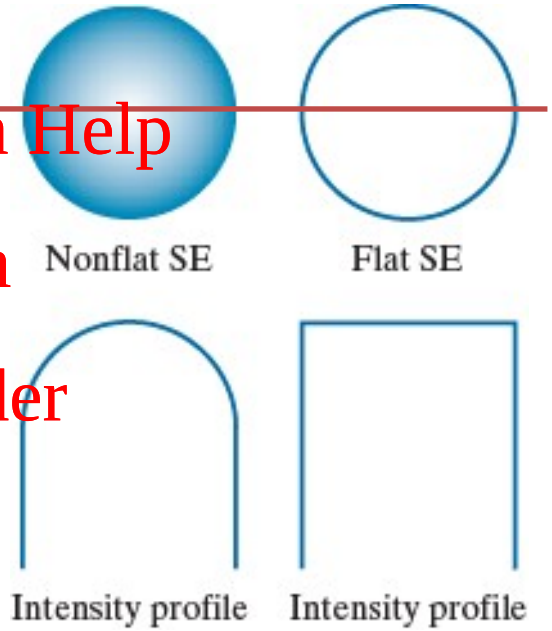
Python Example on Skeletonization

```
horse = io.imread(os.path.join(data_dir, "horse.png"), as_gray=True)

sk = skeletonize(horse == 0)

plot_comparison((horse == 0), sk, 'skeletonize')
```


7.4 Grayscale Morphology

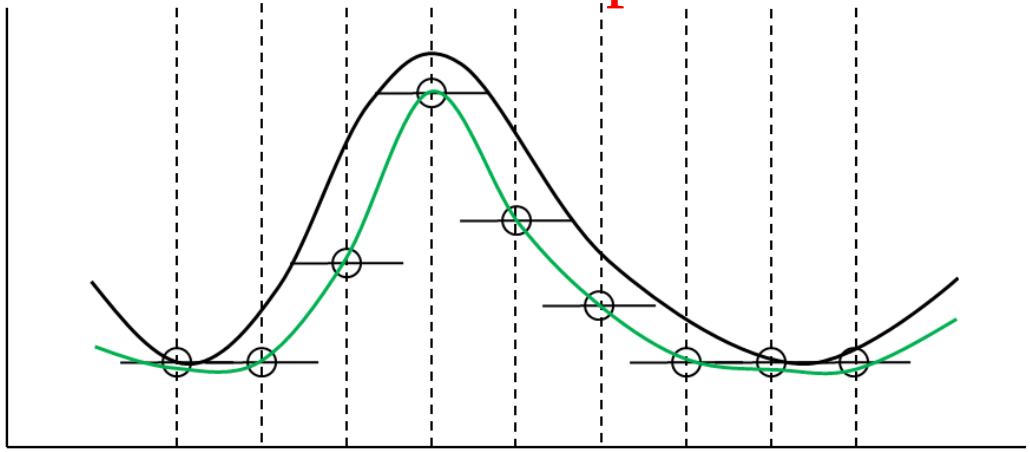
Grayscale Morphology

- Instead of binary images, we now have a grayscale image , where are integer pixels
- be the gray-level SE.
- SE can be flat or non-flat, but here we only focus on flat SE.



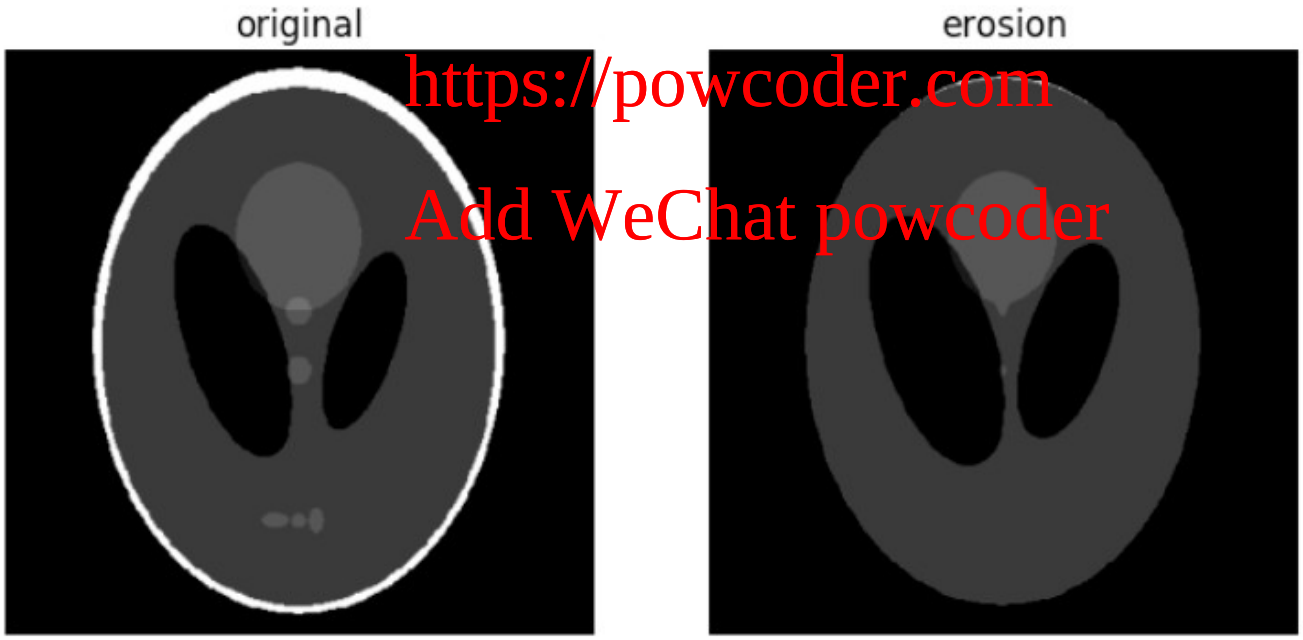
Grayscale Erosion

The erosion of image f by a SE b at any location (x,y) is defined as the minimum value of the image in the region coincident with b when the origin of b is at (x,y):



Grayscale Erosion in Python

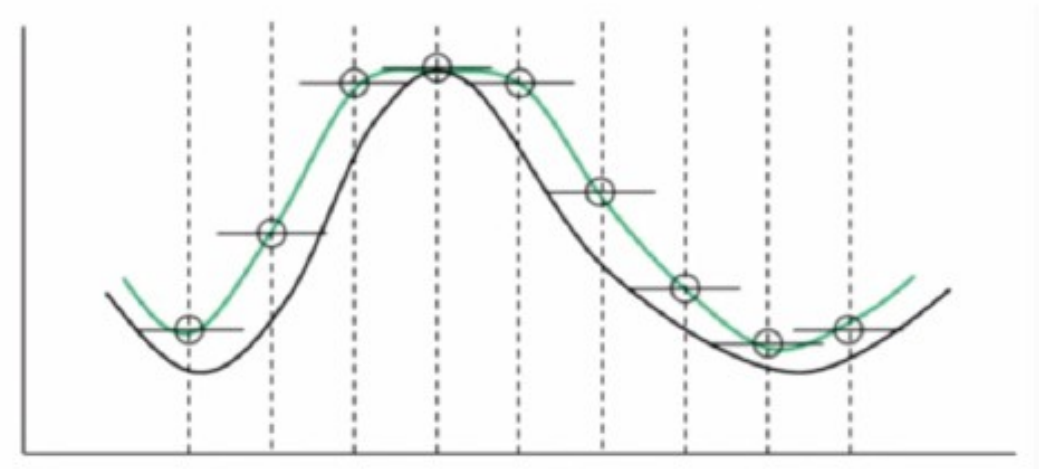
```
from skimage.morphology import erosion, dilation, opening, closing, white_tophat
from skimage.morphology import black_tophat, skeletonize, convex_hull_image
from skimage.morphology import disk

selem = disk(6)
eroded = erosion(orig_phantom, selem)
plot_comparison(orig_phantom, eroded, 'erosion')
```

Grayscale Dilation

The dilation of image f by a SE b at any location (x,y) is defined as the maximum value of the image in the window outlined by b:

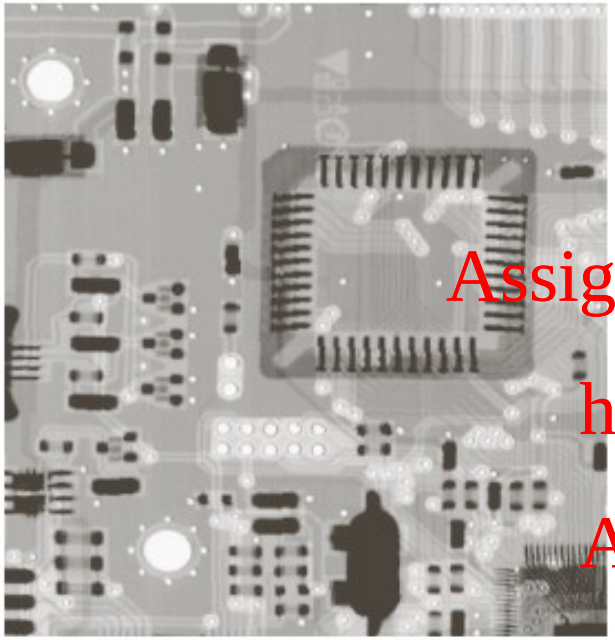
- Note: The SE is reflected as in the binary case.



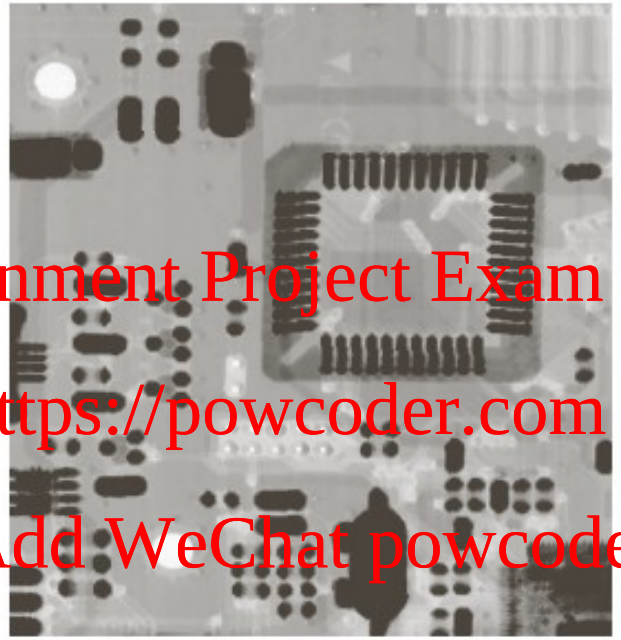
Grayscale Dilation in Python

```
dilated = dilation(orig_phantom, selem)
plot_comparison(orig_phantom, dilated, 'dilation')
```

Examples on Grayscale Erosion and Dilation



Original image



Erosion by a flat disk SE of radius 2:

- Darker background,
- Small bright dots reduced
- Dark features grew.

Dilation by a flat disk SE of radius 2:

- Lighter background,
- Small dark dots reduced
- Light features grew.

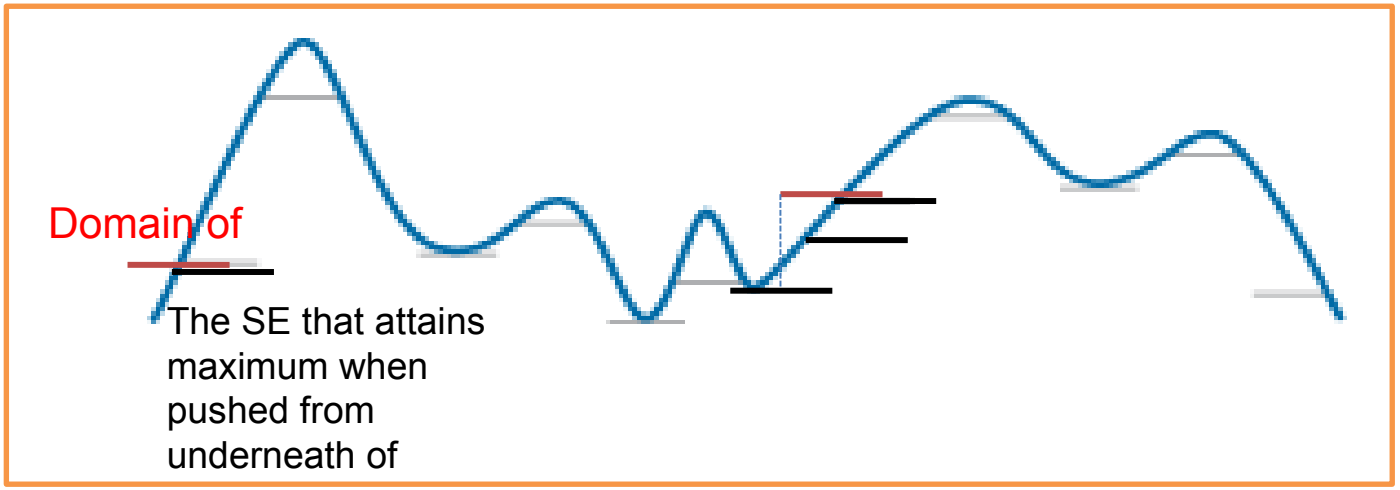
Grayscale Opening and Closing

- The opening of image f by SE b is:

- The closing of image f by SE b is:

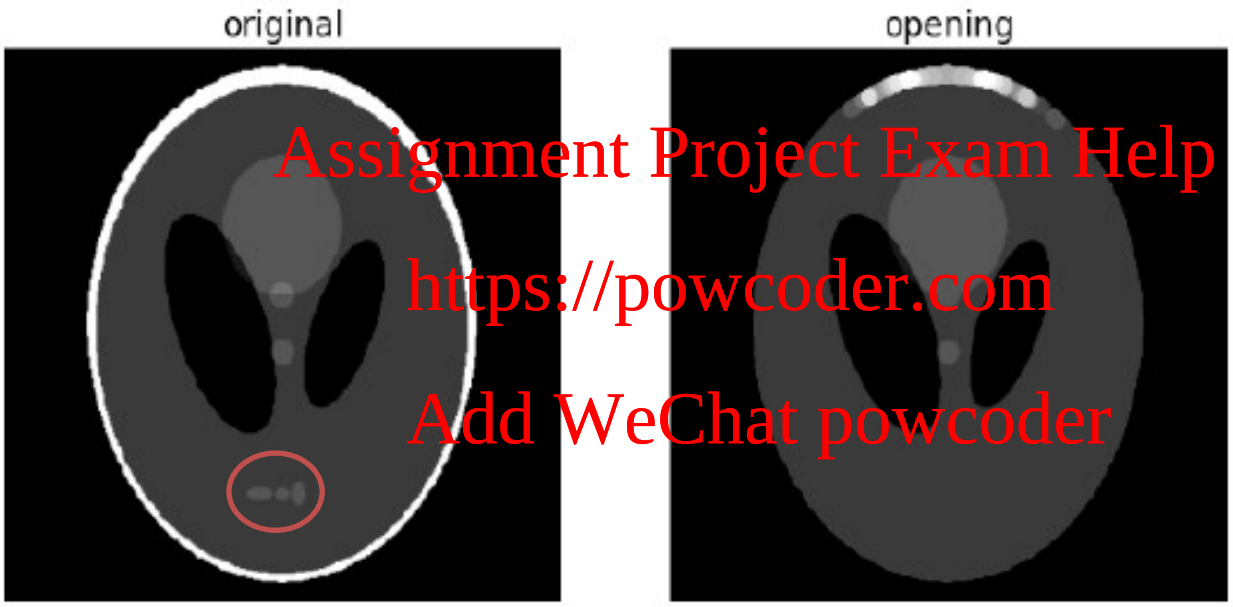
Opening: Geometric Interpretation

- Consider the domain of (SE centered at Point . Red line below) and all candidate SEs that have centres inside the domain of .
- Push all candidate SEs from underneath of (stop when touching the curve)
- There is a SE that attains maximum (Black line for the example below). That maximum is the result of the opening operation at Point .
- Effect:
  - Upward peak clipped by opening
  - Opening removes small bright details



Grayscale Opening in Python

```
opened = opening(orig_phantom, selem)
plot_comparison(orig_phantom, opened, 'opening')
```



- Bottom three blobs got completely eroded.
- Thinner portion of the outline gets eroded.

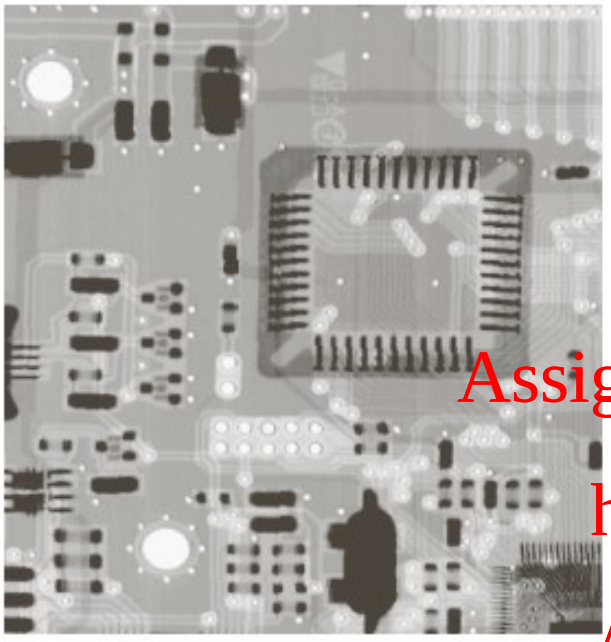
Closing: Geometric Interpretation

- Similar concept as opening
- But here, push SE from top of and take the minimum.
- Effect:
  - Valleys clipped by closing
  - Closing highlights small dark regions of the image.

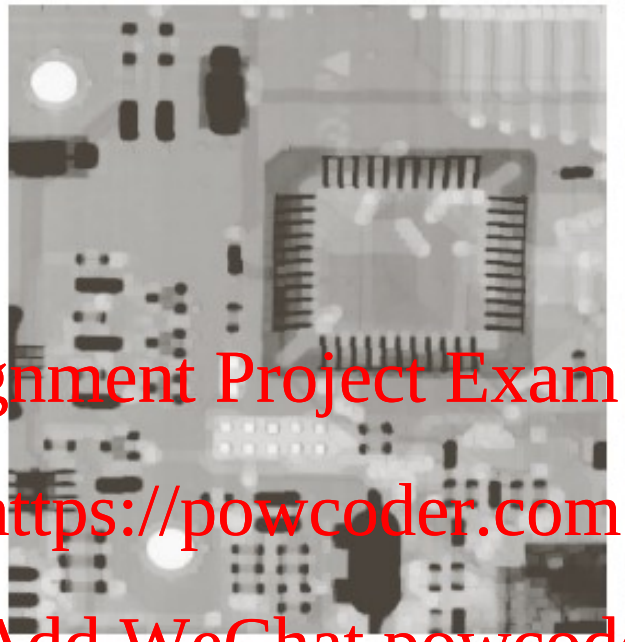
Grayscale Closing in Python

```
phantom = orig_phantom.copy()
phantom[10:30, 200:210] = 0

closed = closing(phantom, selem)
plot_comparison(phantom, closed, 'closing')
```

Examples on Grayscale Opening and Closing



Original image



Opening by a flat disk SE of radius 3:

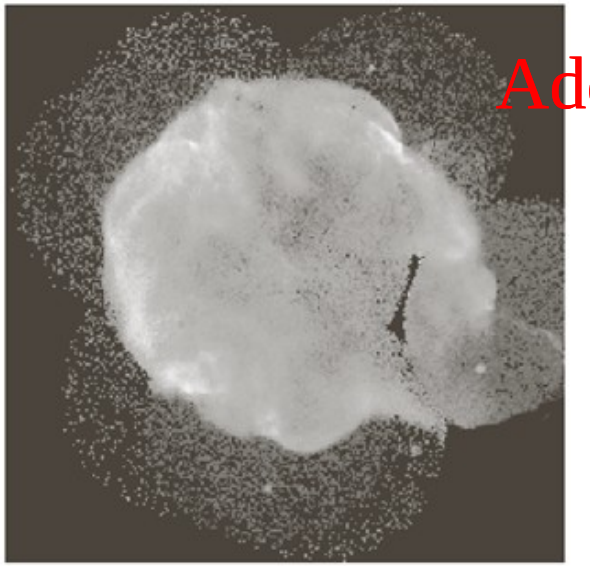
- Intensities of bright features decreased
- Effects on background are negligible (as opposed to erosion).

Closing by a flat disk SE of radius 5:

- Intensities of dark features increased,
- Effects on background are negligible (as opposed to dilation).

Morphological Smoothing

- Opening suppresses light details smaller than the SE and closing suppresses (makes lighter) dark details smaller than the SE.
- They are used in combination as *morphological filters* to eliminate undesired structures.

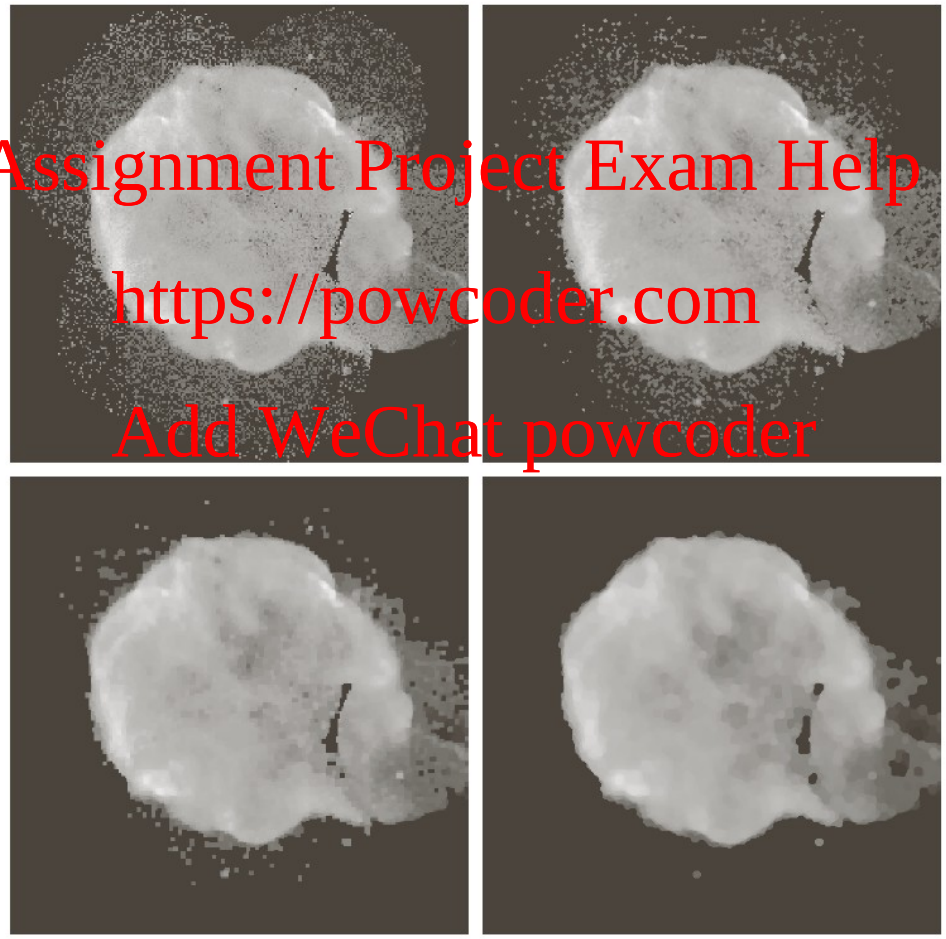


Cygnus Loop supernova.
We wish to extract the central light region.

Example on Morphological Smoothing

Opening followed by closing with disk SE of varying size

Original image



Radius 1

Radius 3

Radius 5

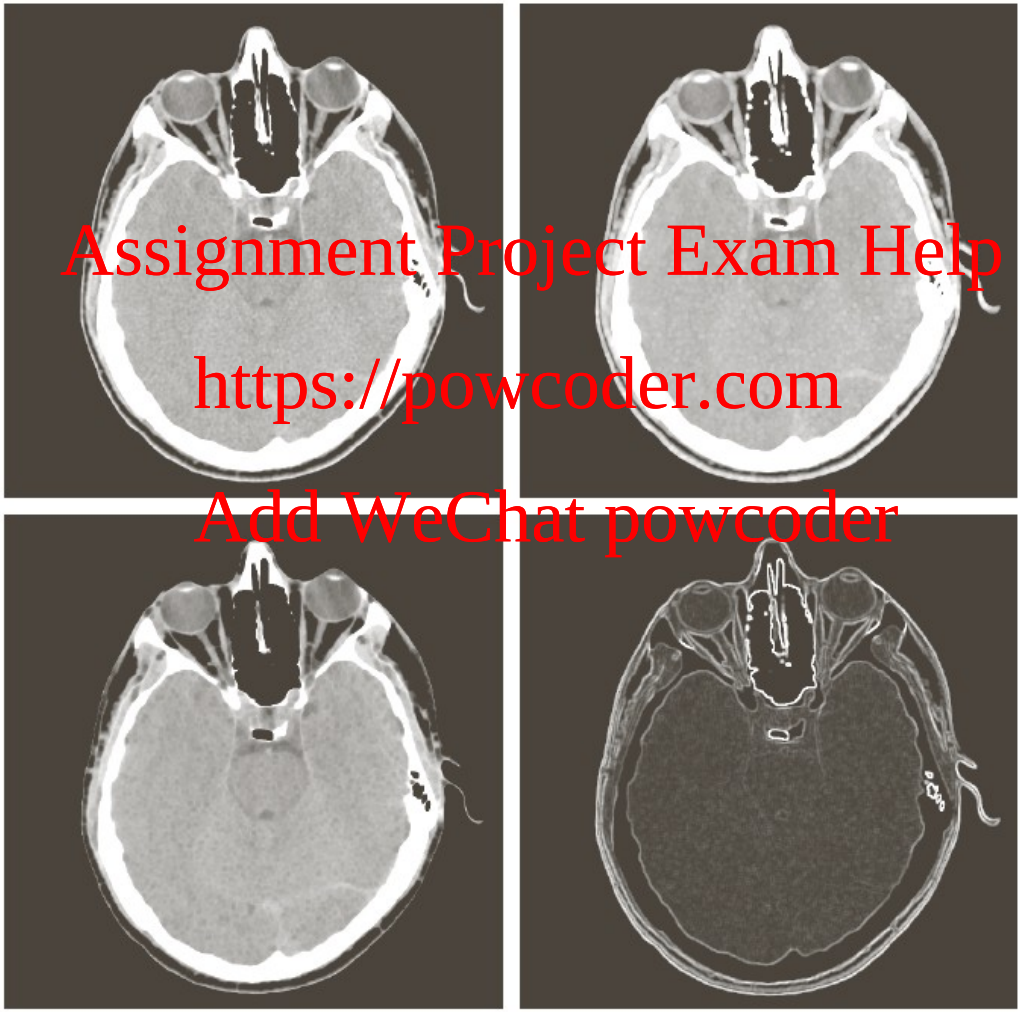
Morphological Gradient

- The difference of the dilation and the erosion of an image emphasizes the boundaries between regions.

- Homogeneous areas are not affected and the subtraction provides a derivative-like effect.
- The net result is an image with flat regions suppressed and edges enhanced.

Example on Morphological Gradient

Original image



Dilation

Erosion

Difference

Top-hat and Bottom-hat Transformations

- Opening suppresses light details smaller than the SE.
- Closing suppresses dark details smaller than the SE.
- Choosing an appropriate SE eliminates image details where the SE does not fit.
- Subtracting the outputs of opening or closing from the original image provides the removed components.

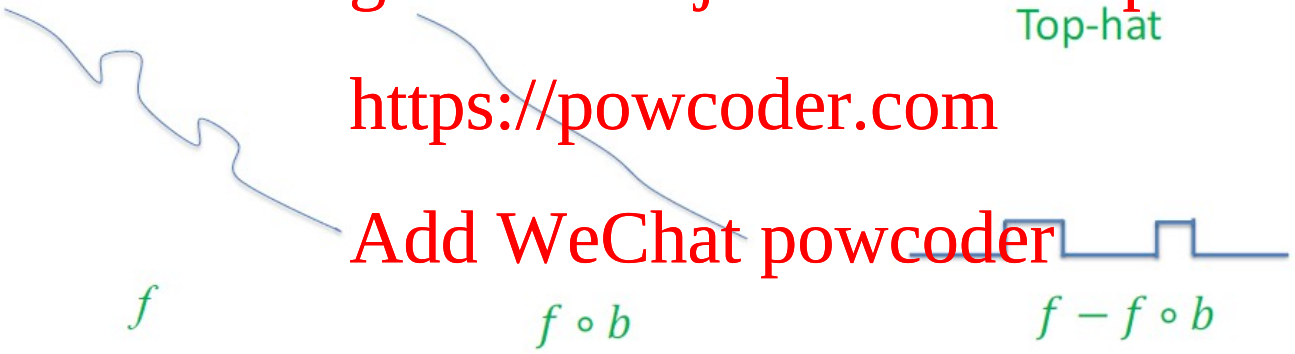
Top-hat and Bottom-hat Transformations

- The top-hat transformation of a grayscale image f is defined as f minus its opening:

- The bottom-hat transformation of a grayscale image f is defined as its closing minus f:

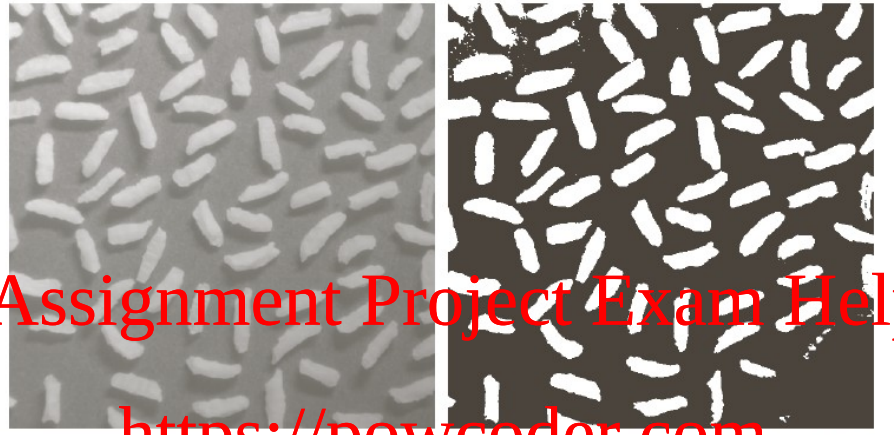
Top-hat and Bottom-hat Transformations

Because the results look like the top or bottom of a hat these algorithms are called **top-hat** and **bottom-hat** transformations



An important application is the correction of nonuniform illumination which is a pre-segmentation step.

Example on Top-hat Transformation



Original image

Thresholded image (Otsu's method)

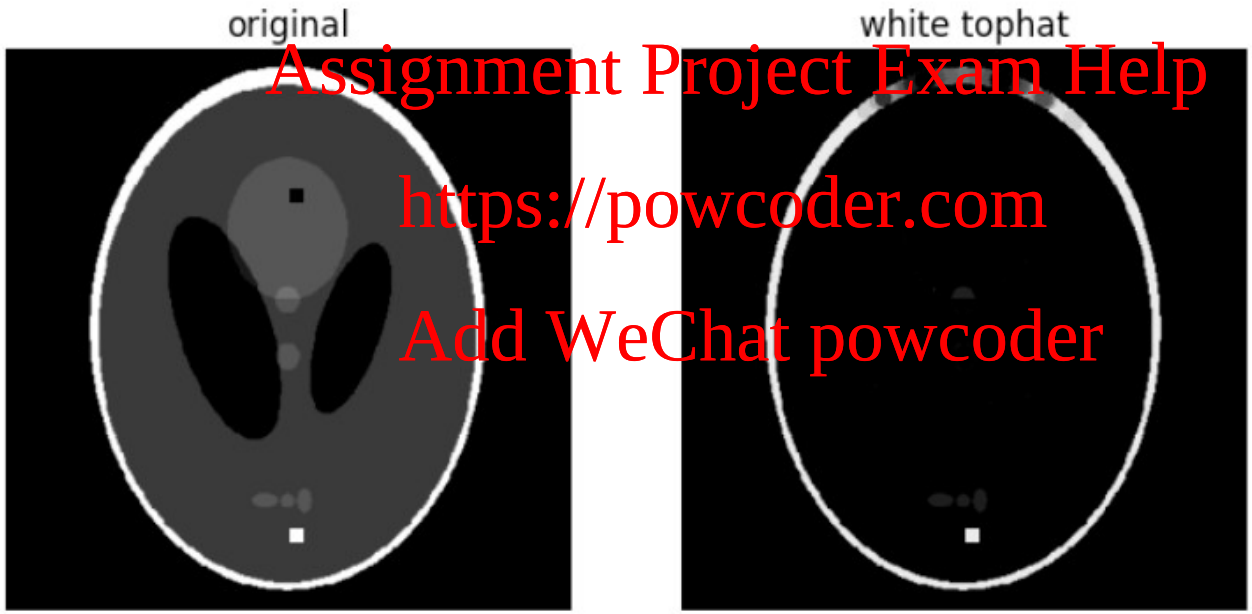
Opened image (disk SE r=40) Does not fit to grains and eliminates them

Top-hat (image-opening) Reduced nonuniformity

Thresholded top-hat

Top-Hat Transformation in Python

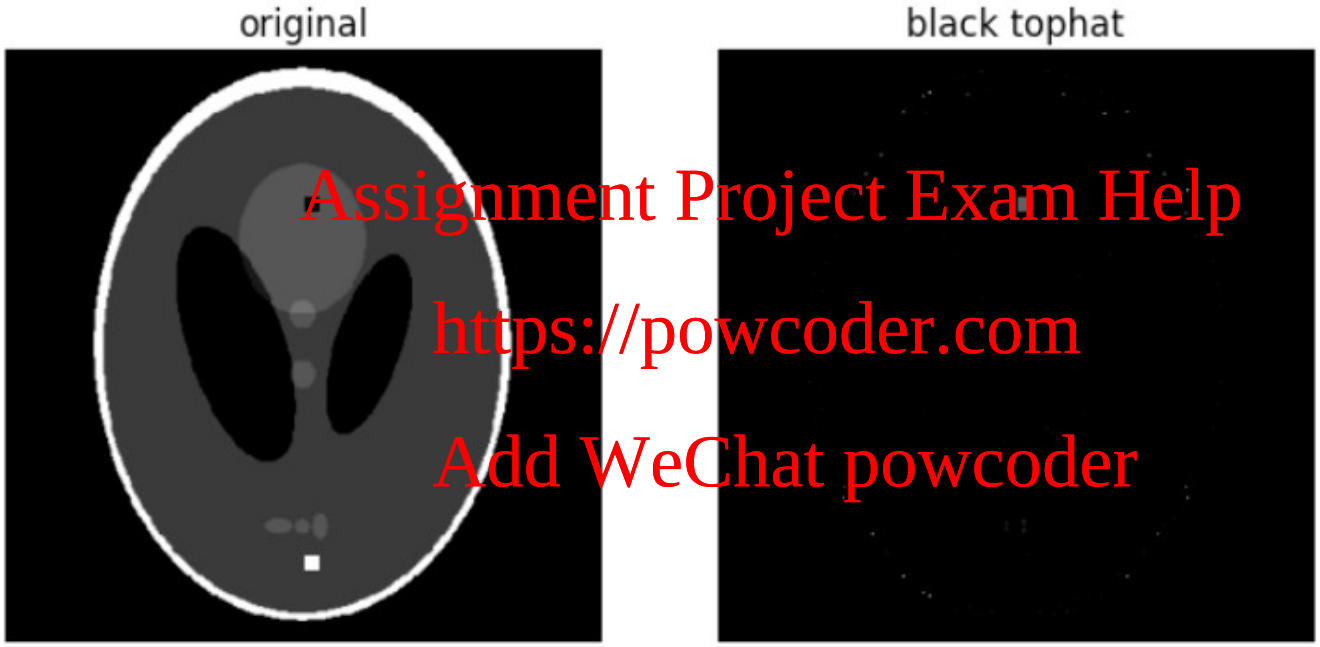
```
phantom = orig_phantom.copy()
phantom[340:350, 200:210] = 255
phantom[100:110, 200:210] = 0

w_tophat = white_tophat(phantom, selem)
plot_comparison(phantom, w_tophat, 'white tophat')
```



- 10-pixel bright element is highlighted because it is smaller than the SE
- Thin, white edges are also retained.
- Thicker regions on top disappear.

Bottom-Hat Transformation in Python

```
b_tophat = black_tophat(phantom, selem)
plot_comparison(phantom, b_tophat, 'black tophat')
```



- 10-pixel element is highlighted because it is smaller than the SE