Morphological Operations-II

Siyue Yu
siyue.yu02@xjtlu.edu.cn

About last lecture

- Morphological operation
  - Structuring element
  - Erosion
  - Dilation
  - Boundary extraction
  - Opening
  - Closing
  - Hit-and-miss Operation, Thinning, Thickening

What is Structuring Element ?

- The structuring element (SE) is like a window that we process through it the input signal
- ‘1’ values define the neighborhood (of interest).
- ‘0’ values in SE is equivalent to don’t use the corresponding input pixel!
- Structural Elements have an origin, which identifies the pixel being processed.

What is Erosion ?

- Erosion is the set of all points in the image, where the structuring element “fits into”.
- If the structuring element fits in the foreground object, write “1” at the origin of the structuring element in the output image!

What is Dilation ?

- Dilation is the set of all points in the image, where the structuring element “touches” the foreground.
- If the structuring element touches the foreground object, write “1” at the origin of the structuring element in the output image!

Erosion and Dilation

```
im =

     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     1     0
     0     1     0     0     0     0     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     0     1     1     1     1     0     0
```

```
se =

     1     1     1
     1     1     1
     1     1     1
```

Do it yourself

Padding the boundary with the same value as the neighbor if it is needed.

Padding

```
se =

     1     1     1
     1     1     1
     1     1     1
```

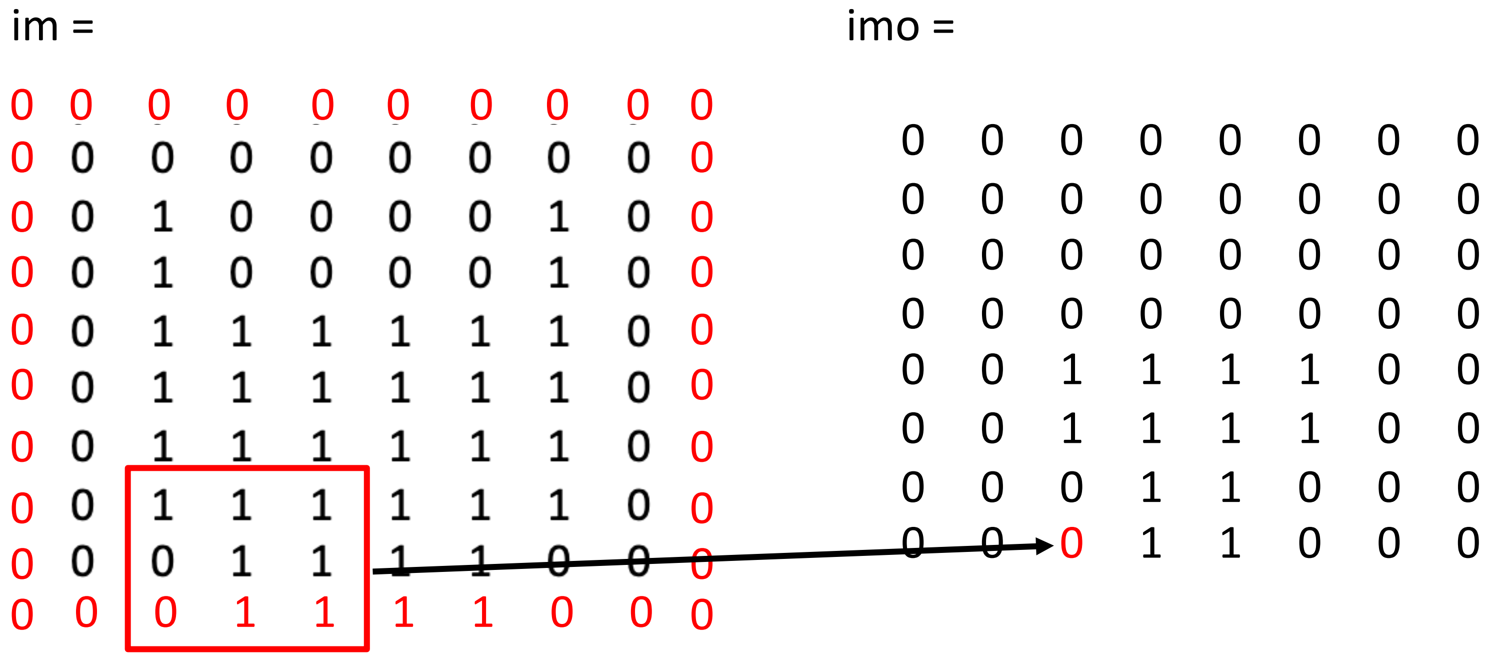
```
im =

     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     1     0
     0     1     0     0     0     0     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     1     1     1     1     1     1     0
     0     0     1     1     1     1     0     0
```

→

```
0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  1  0  0
0  0  1  0  0  0  0  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  0  1  1  1  1  0  0  0
0  0  0  1  1  1  1  0  0  0
```

Padding the boundary with the same value as the neighbor if it is needed.

Erosion Result



```
im =

0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  1  0  0
0  0  1  0  0  0  0  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  1  1  1  1  1  1  0  0
0  0  0  1  1  1  1  0  0  0
0  0  0  1  1  1  1  0  0  0
```

```
imo =

0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  1  1  1  1  0  0
0  0  1  1  1  1  0  0
0  0  0  1  1  0  0  0
0  0  0  1  1  0  0  0
```

Dilation Result

```
im =

0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   1   0   0   0   0   1   0   0
0   0   1   0   0   0   0   1   0   0
0   0   1   1   1   1   1   1   0   0
0   0   1   1   1   1   1   1   0   0
0   0   1   1   1   1   1   1   0   0
0   0   1   1   1   1   1   1   0   0
0   0   0   1   1   1   1   0   0   0
0   0   0   1   1   1   1   0   0   0
```

```
imo =

    1   1   1   0   0   1   1   1
    1   1   1   0   0   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   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   1
```

What is Boundary Extraction ?

- Detect the contour of the target object
- Dilate the input image.
- Subtract the input image from dilated image. Or we can call XOR logical operation here.

About this lecture

- Morphological operation
  - Structuring element
  - Erosion
  - Dilation
  - Boundary extraction
  - Opening
  - Closing
  - Hit-and-miss Operation, Thinning, Thickening

Opening and Closing

- Important operations directly derived from the fundamental operations
  - Dilation
  - Erosion

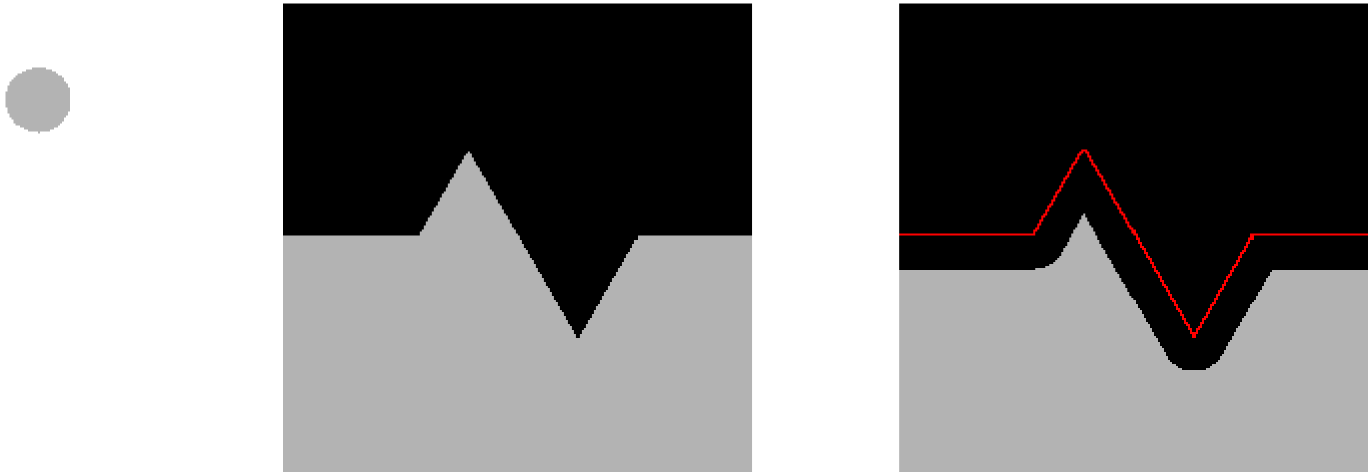
- Opening and closing are dual operations

What is Opening ?

- The morphological Opening operation is an erosion followed by a dilation, using the same structuring element for both operations
  - Similar to Erosion however it is less destructive

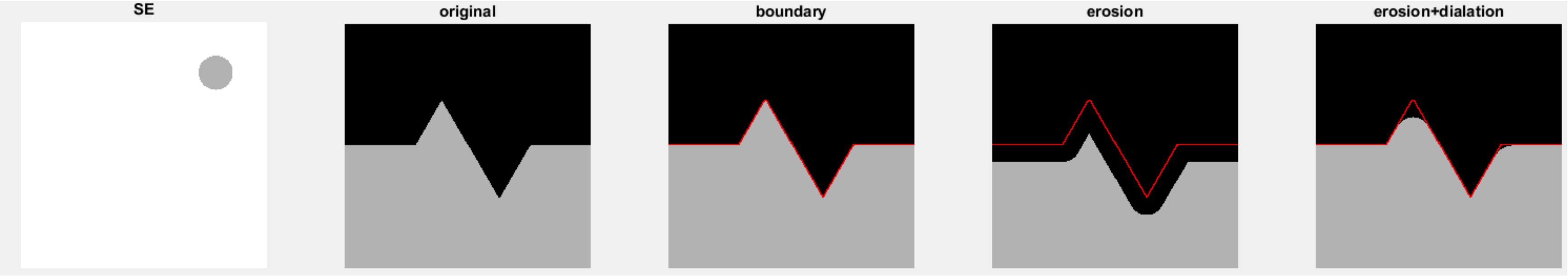
What is Opening ?

- Opening operation is an erosion followed by a …



What is Opening ?

- Opening operation is an erosion followed by a dilation



what about the dilation, imagine it

How does it work ?

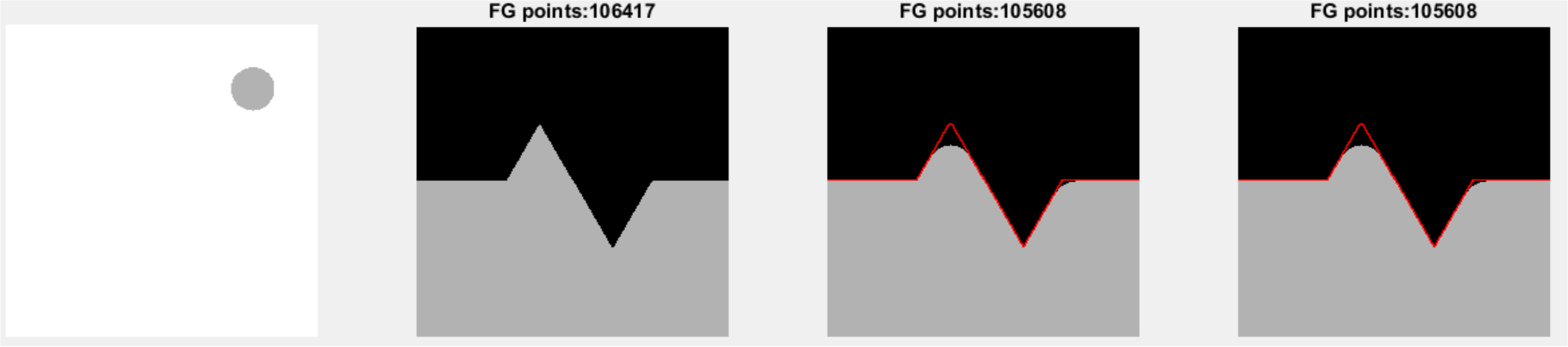
- The structuring element (SE) is moved inside each foreground region, and while doing so:
  - All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved.

How does it work ?

- The structuring element (SE) is moved inside each background region, and while doing so:
  - All pixels which can be covered by the SE with the SE being entirely within the background region will be preserved.

How does it work ?

- Opening is idempotent: repeated application has no further effects! (for the same SE)



$$SE = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

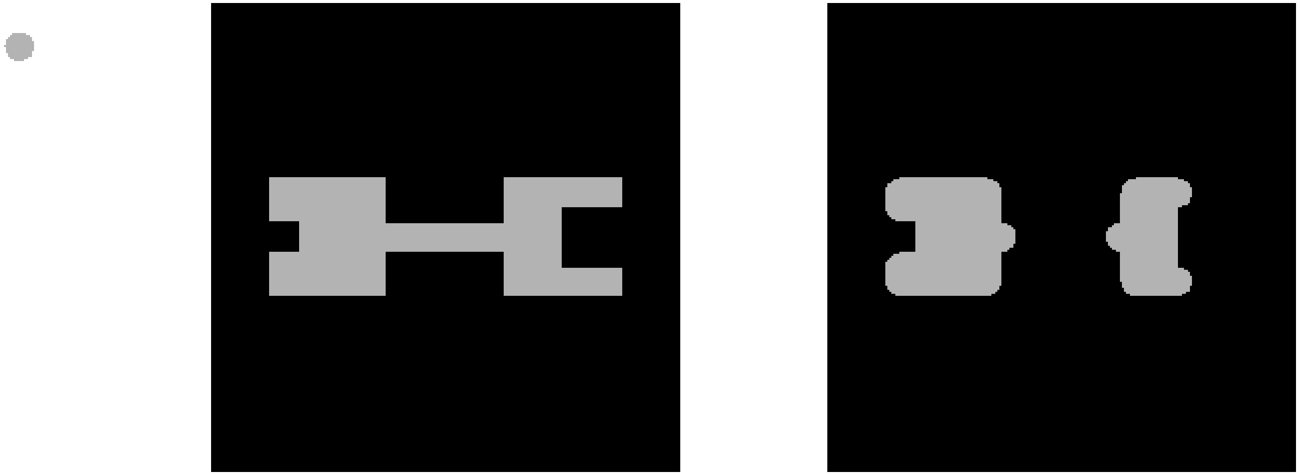
im=

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{matrix}
\xrightarrow{\text{erosion}}
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\xrightarrow{\text{dilation}}
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\xrightarrow{\text{erosion}}
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\xrightarrow{\text{dilation}}
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

Do it yourself

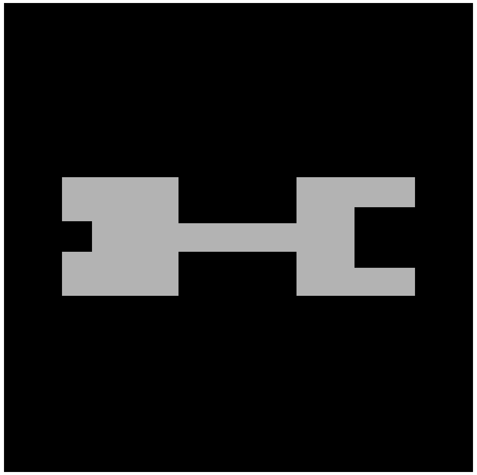
- Try it yourself with Matlab!
  - SE is disk with radius 16

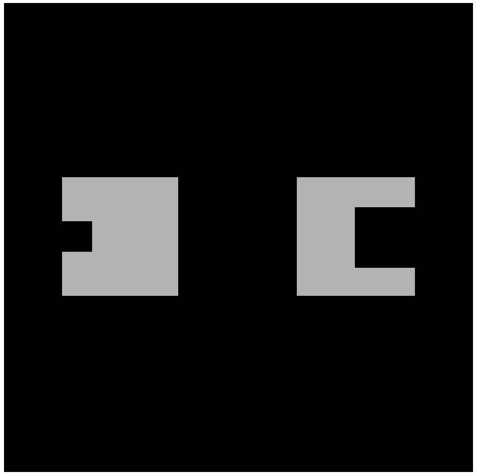


SE diameter=32,
right part width=64, height=64
middle part width = 60, height=21

Do it yourself

- Try it yourself with Matlab!
  – SE is square with width 32





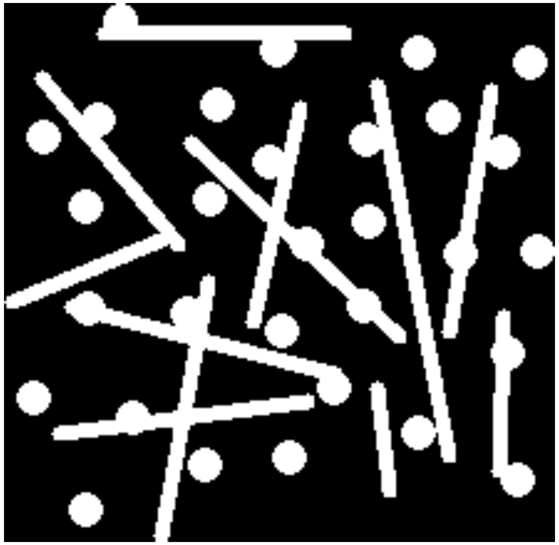
- This is why we call it Opening

What is used for ?

- The Opening Operation :
  - Smooth the contours of objects.
  - Eliminates thin protrusions.
  - Separates parts of objects that are linked by “thin” connection
  - Removes small objects from the image
  - Noise reduction, object detection and segmentation, image enhancement, shape analysis

Opening Example

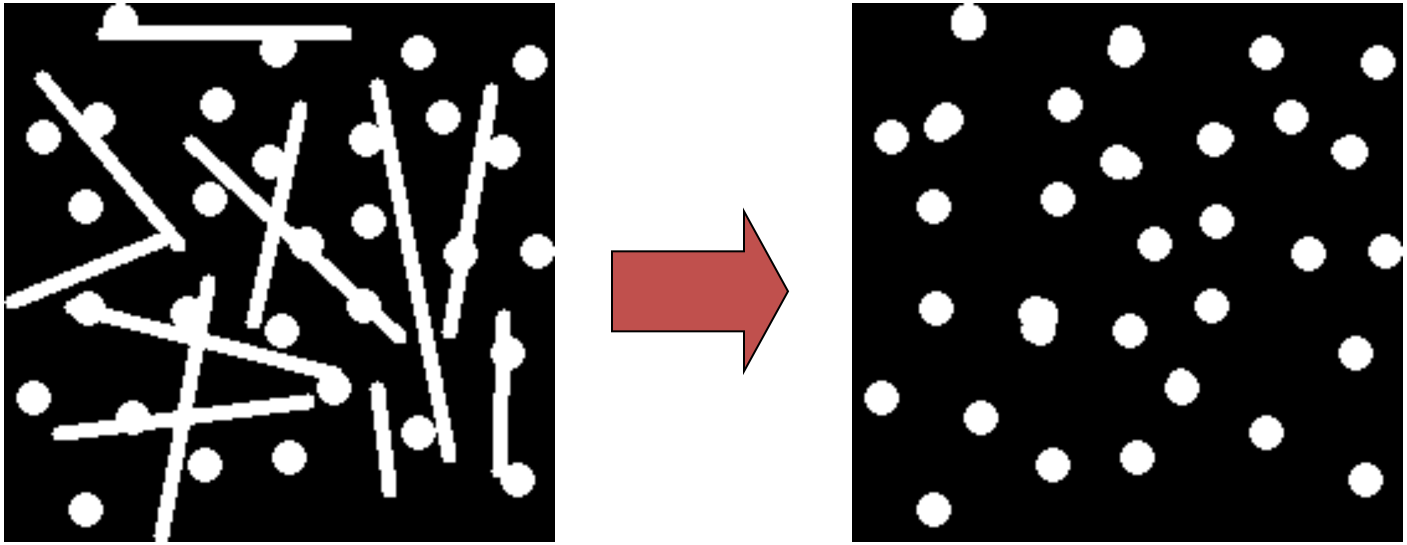
- Opening with a 11 pixel diameter disc



Opening Example

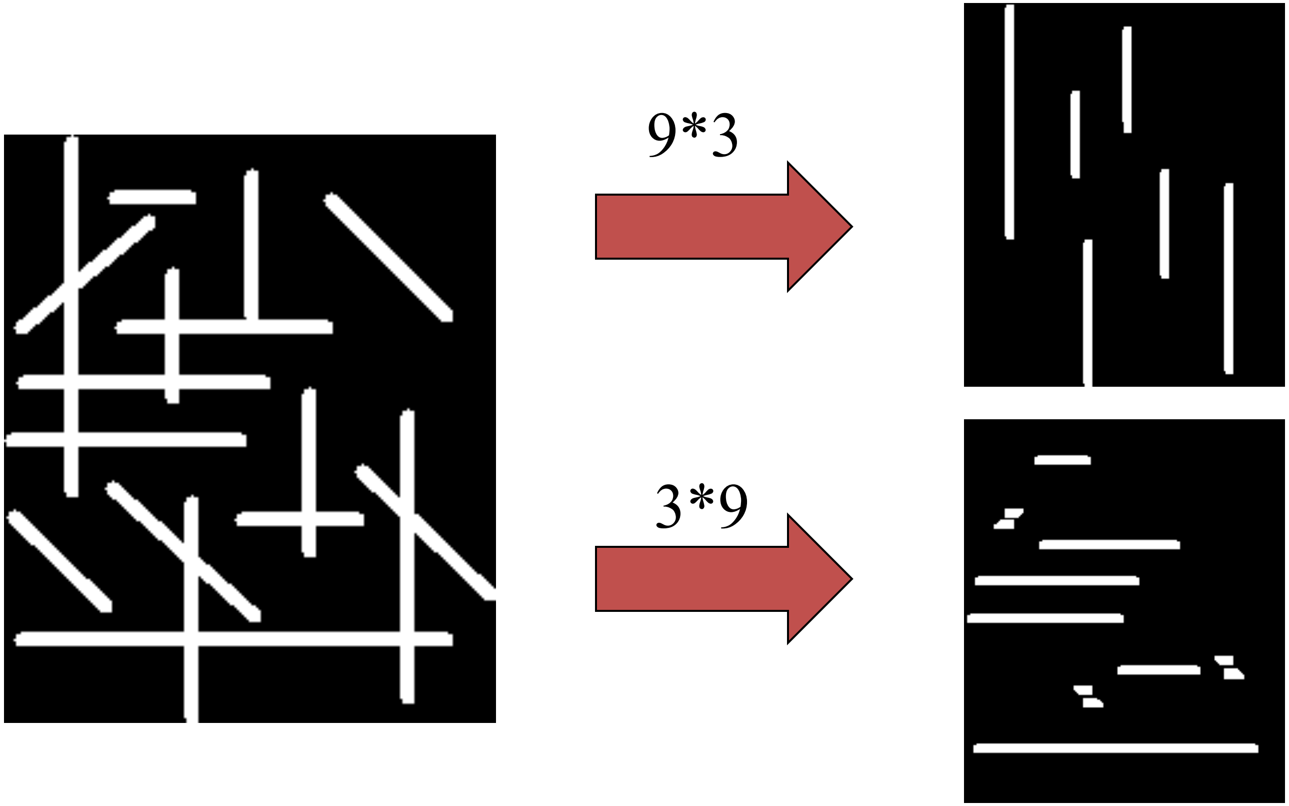
- Opening with a 11 pixel diameter disc

- The output will have balls with almost the same shape as the original one.
- If you just use erosion what will happened ?



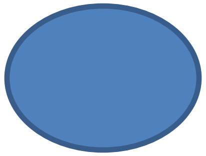
Opening Example

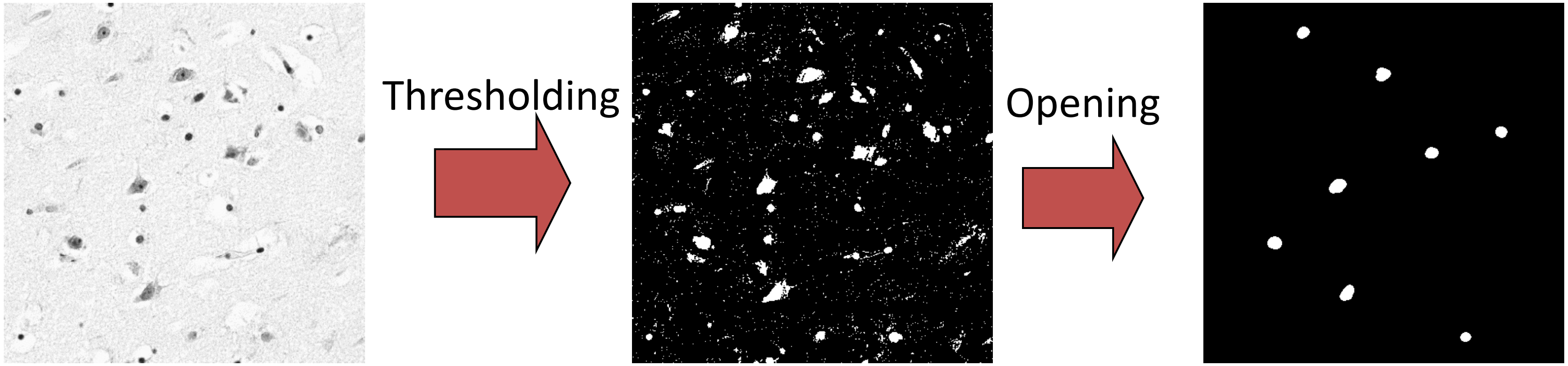
- 3x9 and 9x3 Structuring Element



Use Opening for Separating Blobs

- Use large structuring element that fits into the big blobs
- Structuring Element: 11 pixel disc

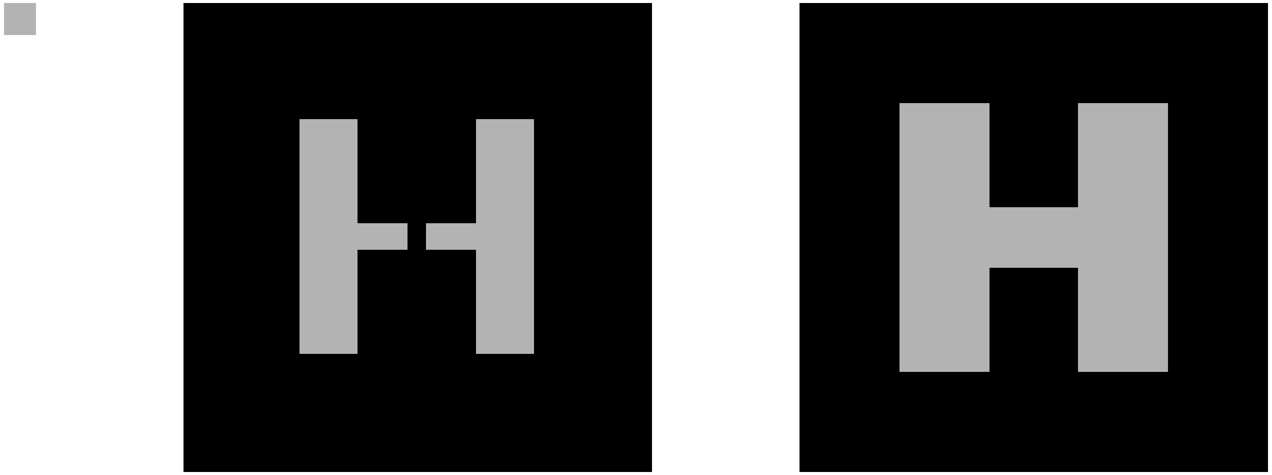




Closing

What is the problem with Dilation?

- The objects get “fatter”

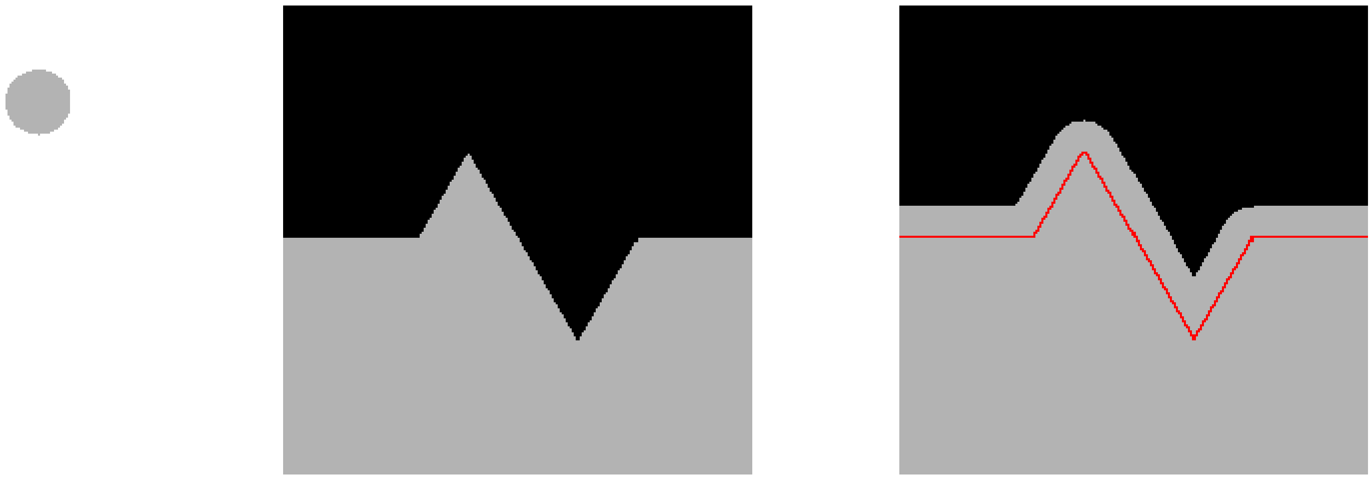


What is Closing ?

- Closing is defined as a Dilation, followed by an Erosion using the same structuring element for both operations.

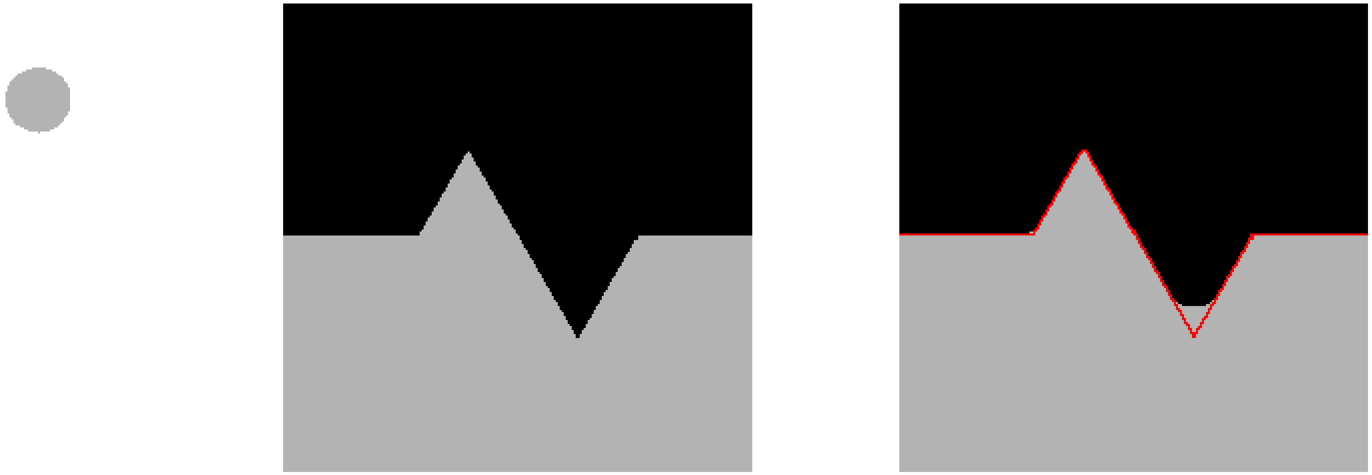
What is Closing ?

- Closing operation is an Dilation followed by …



What is Closing ?

- Closing operation is an Dilation followed by Erosion



code

How does it work ?

- The structuring element (SE) is moved inside each foreground region, and while doing so:
  - All foreground pixels which can be entirely covered by the SE will be preserved as foreground.

How does it work ?

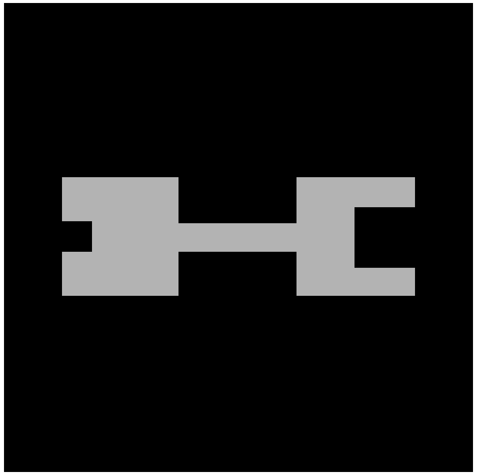
- The structuring element (SE) is moved inside each background region, and while doing so:
  - All background pixels which can be entirely covered by the SE will be preserved as background.

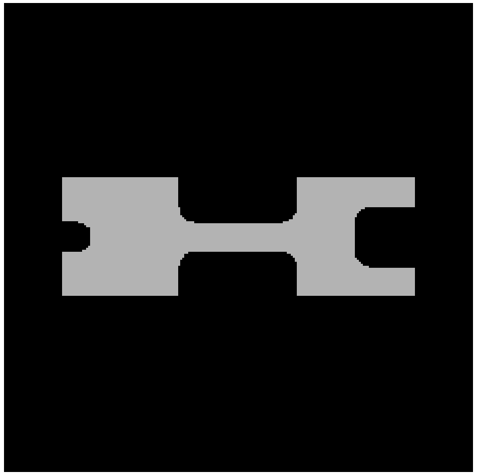
How does it work ?

- Closing is idempotent: repeated application has no further effects! (for the same SE)

Do it yourself

- Try it yourself with Matlab!
  - SE is disk with radius 16

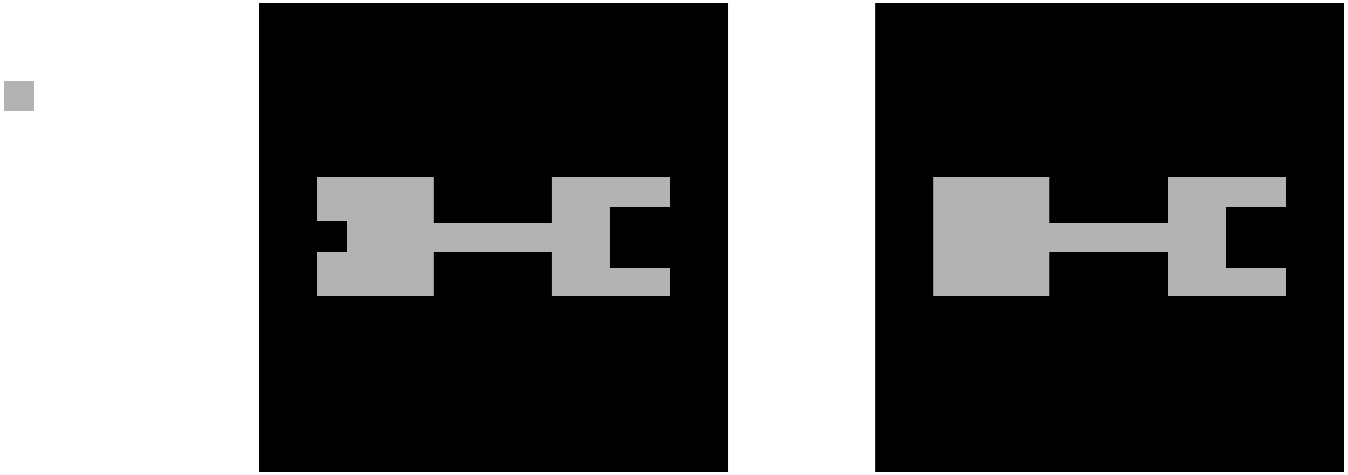




code

Do it yourself

- Try it yourself with Matlab!
  - SE is square with width 34



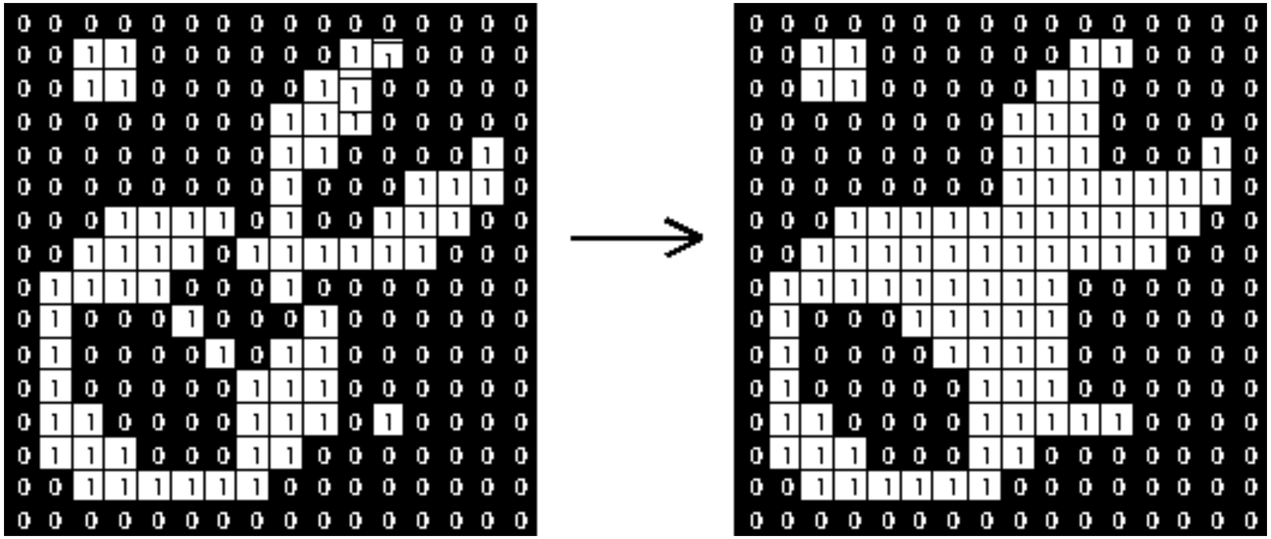
- This is why we call it Closing

What is used for ?

- The Closing Operation is similar to Dilation:
  - Removes small holes in the foreground
  - Tends to enlarge foreground regions, shrink background
  - Links close objects

Closing

- Structuring element: 3x3 square

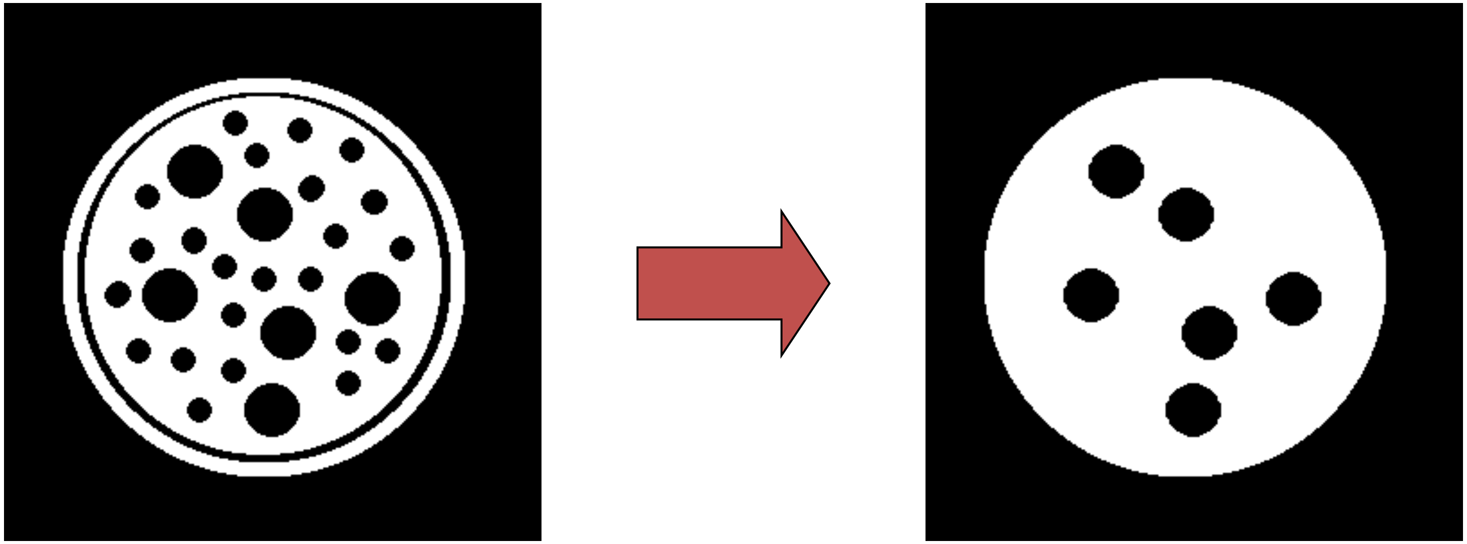


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
0	0	0	1	1	1	1	0	1	0	0	1	1	1	0	0
0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0
0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0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

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

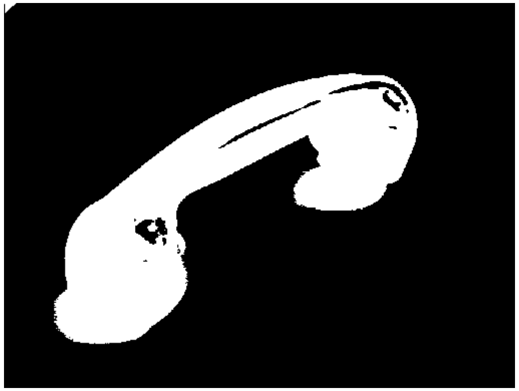
Closing Example

- Closing operation with a 22 pixel disc
- Closes small holes in the foreground

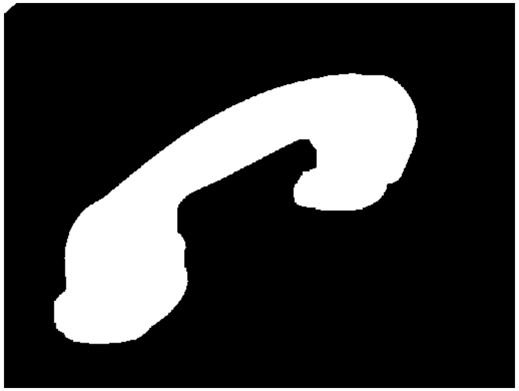


Closing Example

1. Threshold
2. Closing with disc of size 20



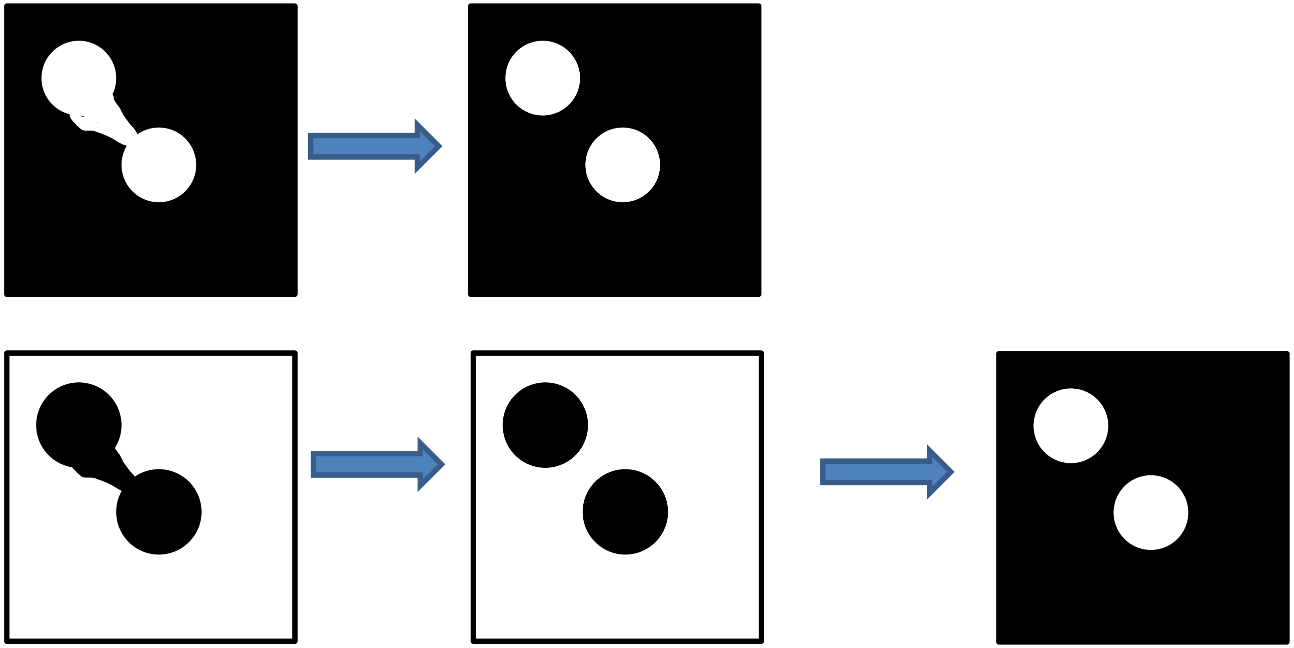
Thresholded



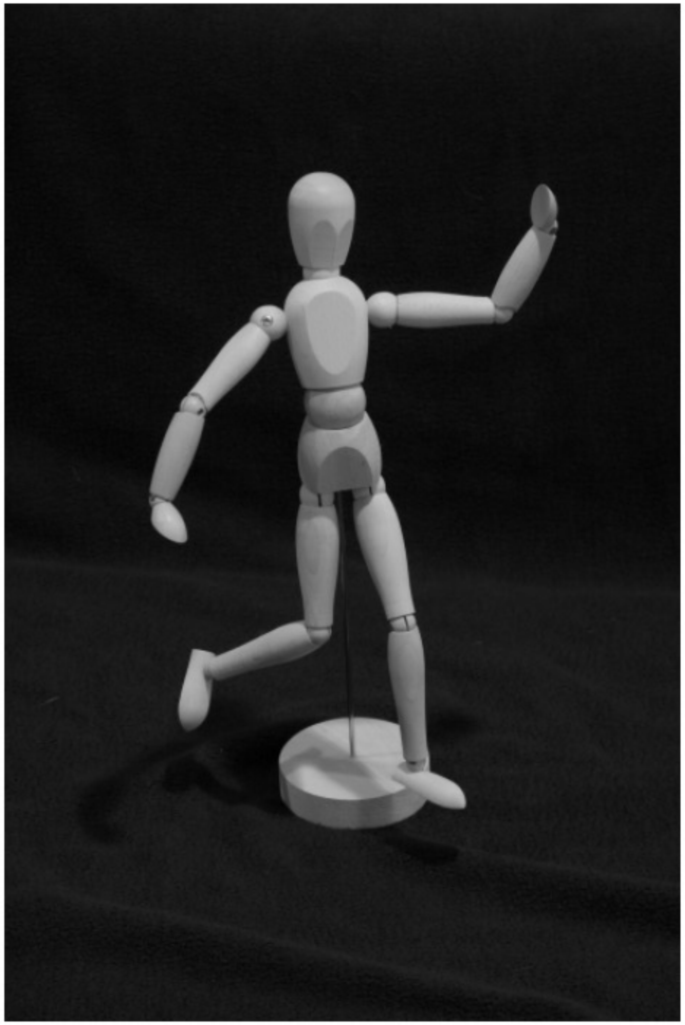
closed

Opening and Closing

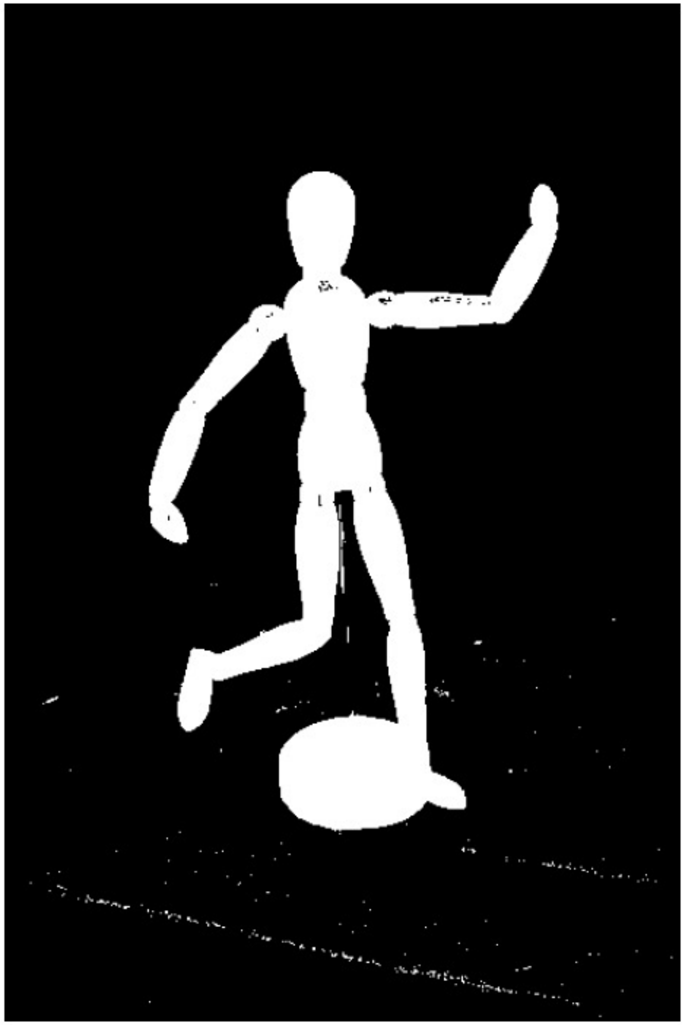
- Opening is the dual of closing
  - This means that opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same SE.



Example of Opening and Closing

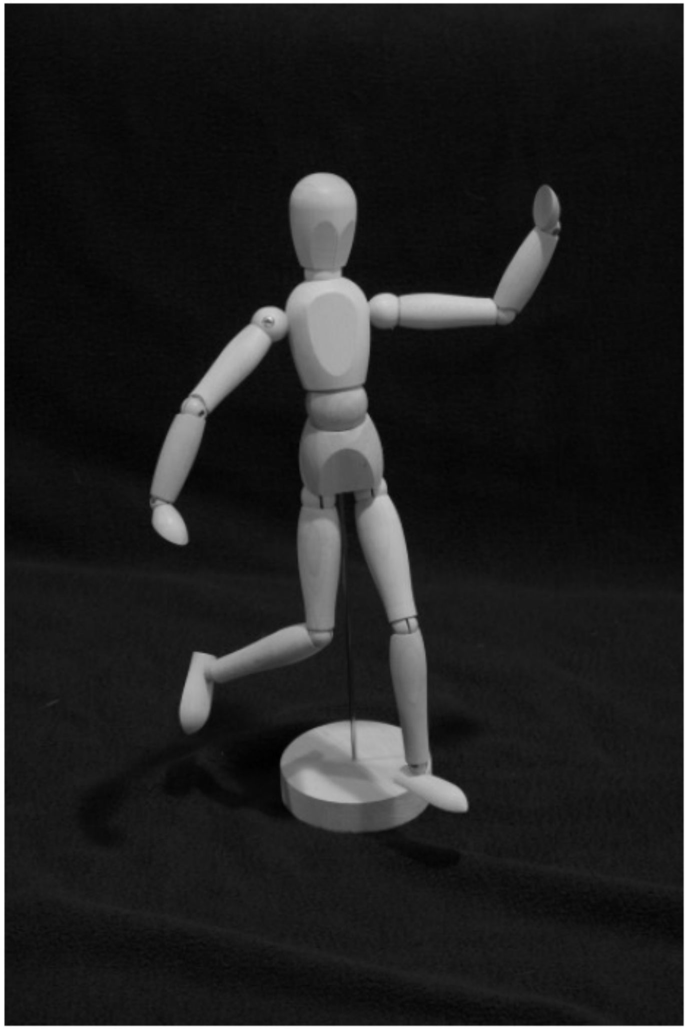


Original image

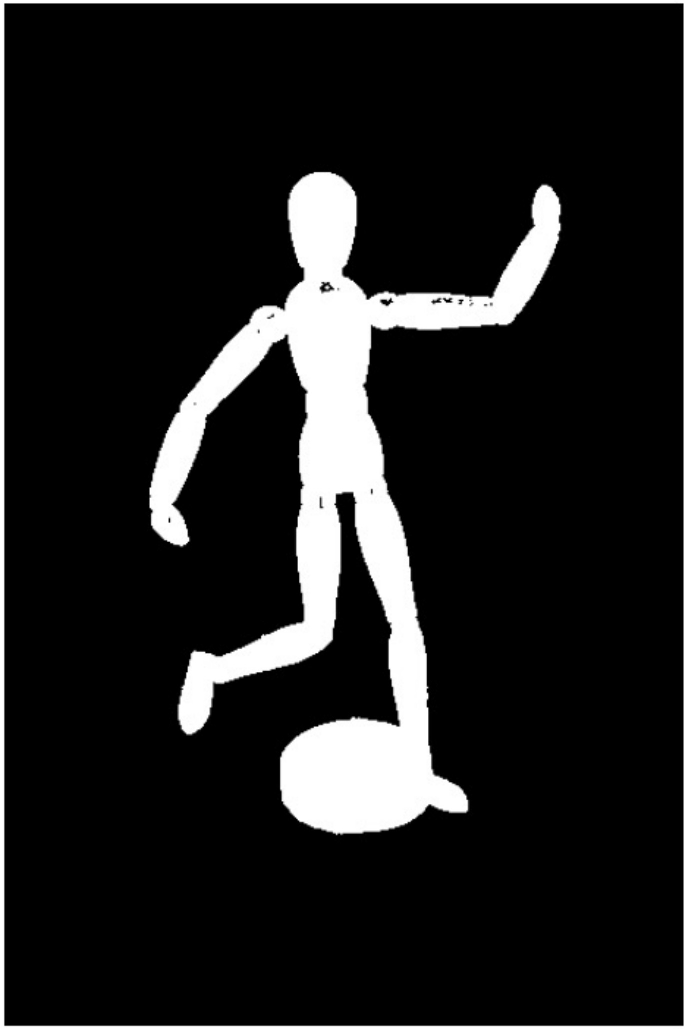


Initial threshold

Example of Opening and Closing

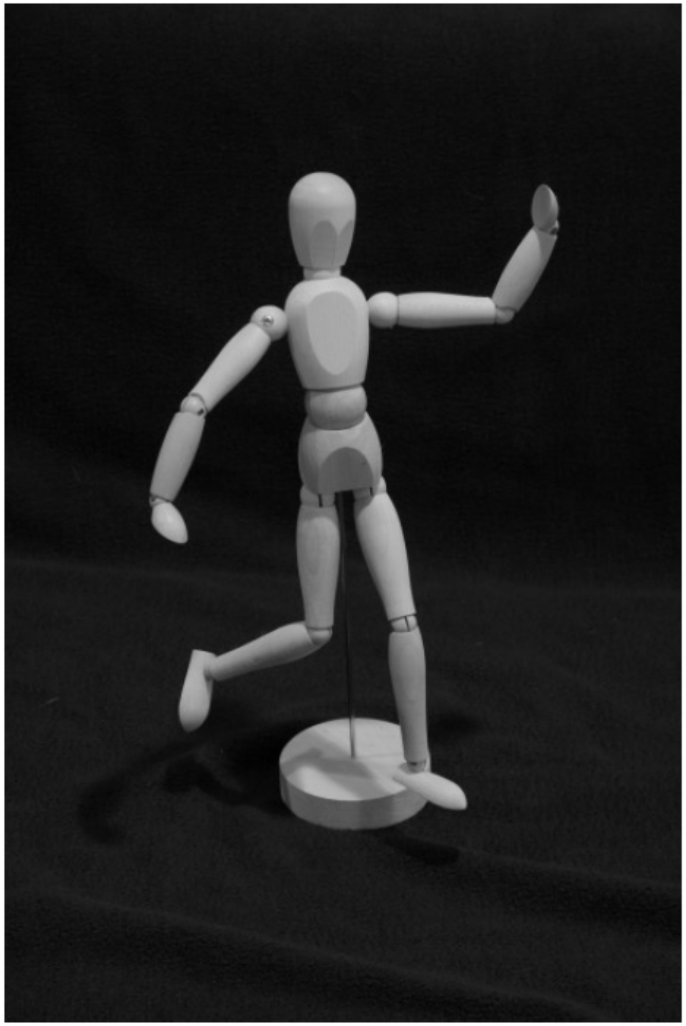


Original image

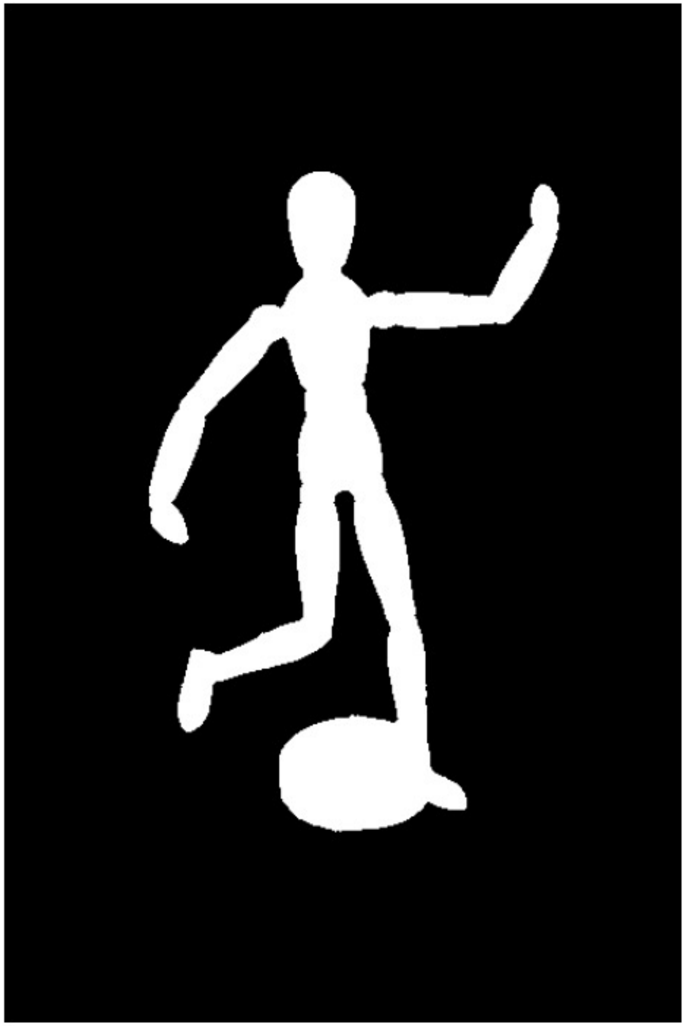


After opening

Example of Opening and Closing



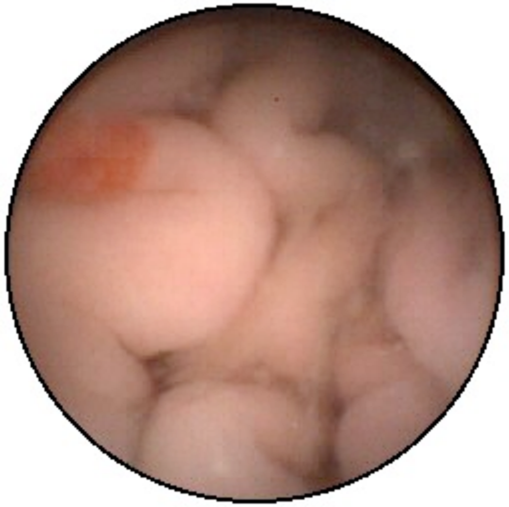
Original image



After closing

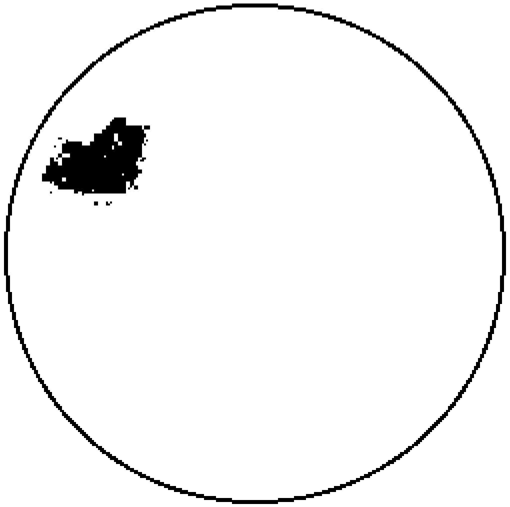
Example-bleeding detection in WCE

- Bleeding detection in the WCE (Wireless Capsule Endoscopy) images



Example-bleeding detection in WCE

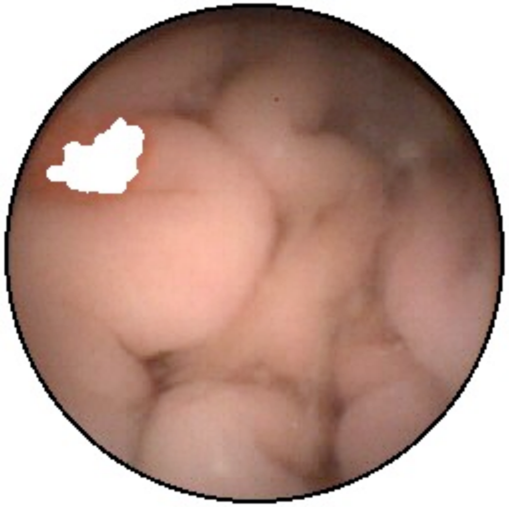
- This is the “intermediate” (not final) bleeding mask



Noise
Small hole

Example-bleeding detection in WCE

Covering the final bleeding mask onto the original image.

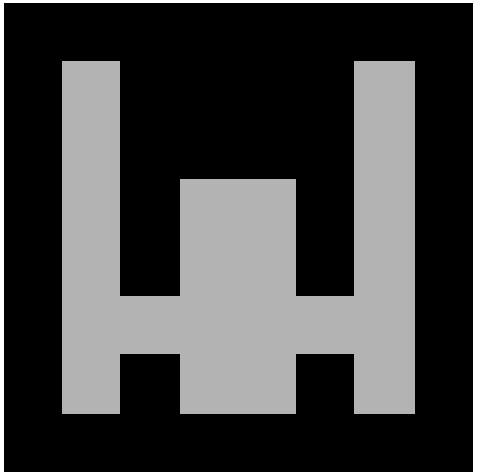


HIT and MISS

Shape detection

- Suppose you want to detect the "local maximum in a tool"

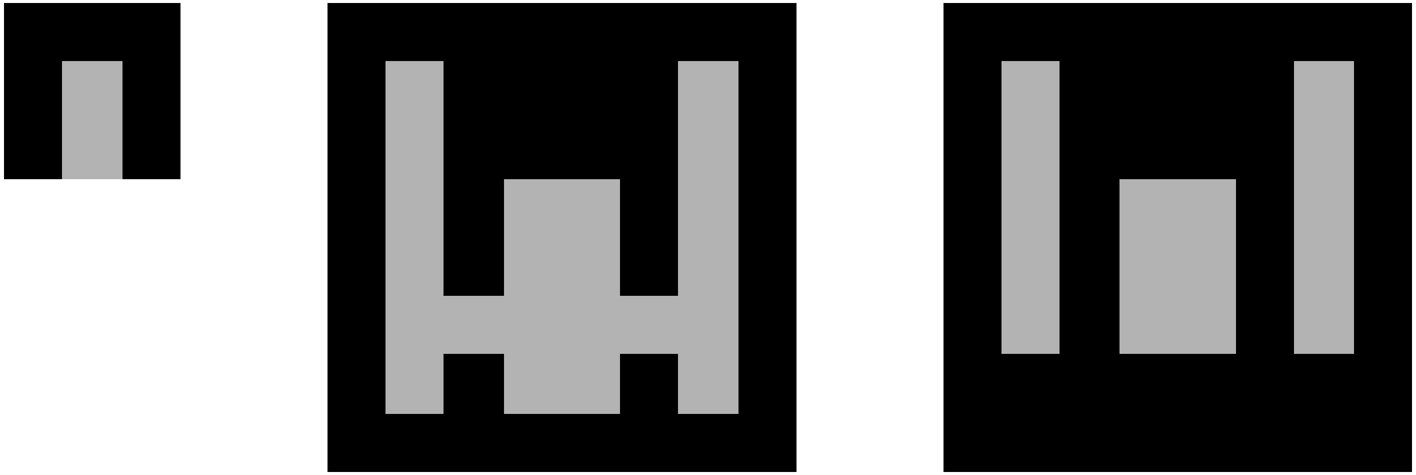
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	1	1	1	1	0
0	1	0	1	1	0	1	0
0	0	0	0	0	0	0	0



```
im = zeros(8);
im(2:7, 2) = 1;
im(4:7, 4:5) = 1;
im(2:7, 7) = 1;
im(6, 3:6) = 1;
```

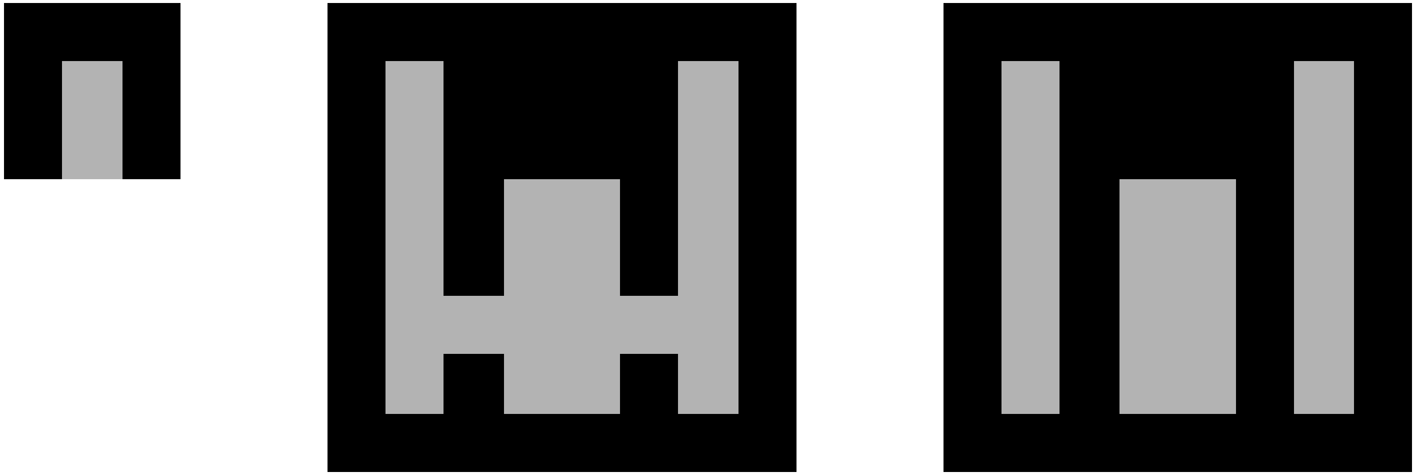
Shape detection

- Using erosion, would it work ?
  - Erosion could be used to detect some shapes



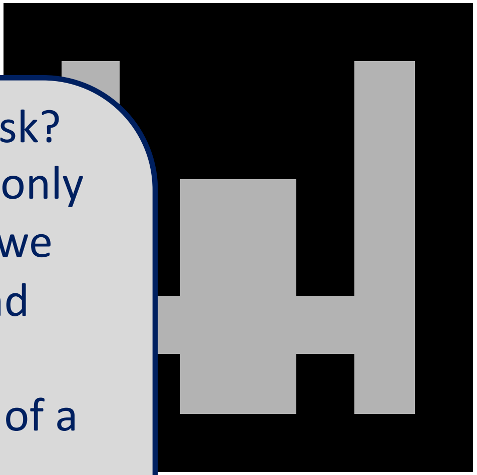
Shape detection

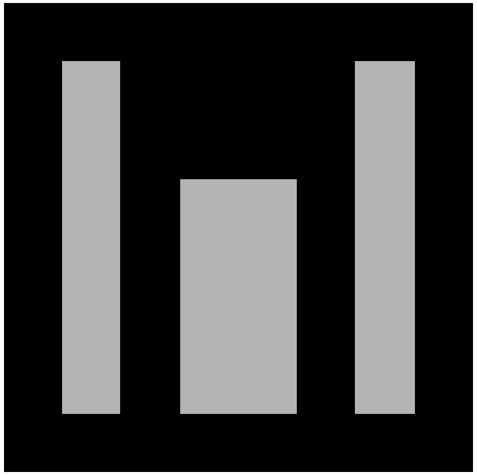
- Using opening, would it work ?
  - Opening could be used to detect some shapes



Shape detection

- Using opening, would it work ?
  - Opening could be used to detect some shapes





So what to do to address the task?
The problem here is that we are only looking at the foreground, but we didn't consider the background

What is the main characteristics of a local maximum ?
The local maximum has an upper neighbor which should be black.

Could we exploit this fact ?

Shape detection

- Using opening, would it work ?
  - Opening could be used to detect some shapes

So what to do to address the task?
The problem here is that we are only looking at the foreground, but we didn't consider the background

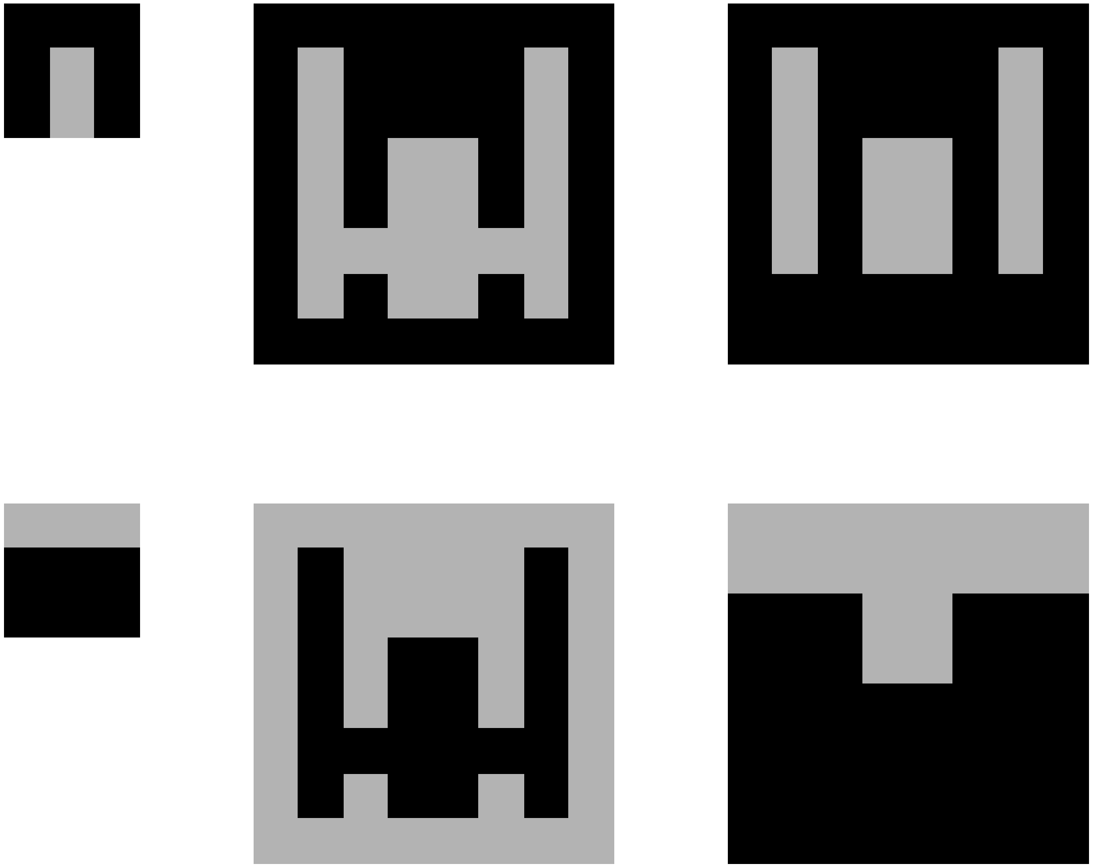
What is the main characteristics of a local maximum ?
The local maximum has an upper neighbor which should be black.

Could we exploit this fact ?

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

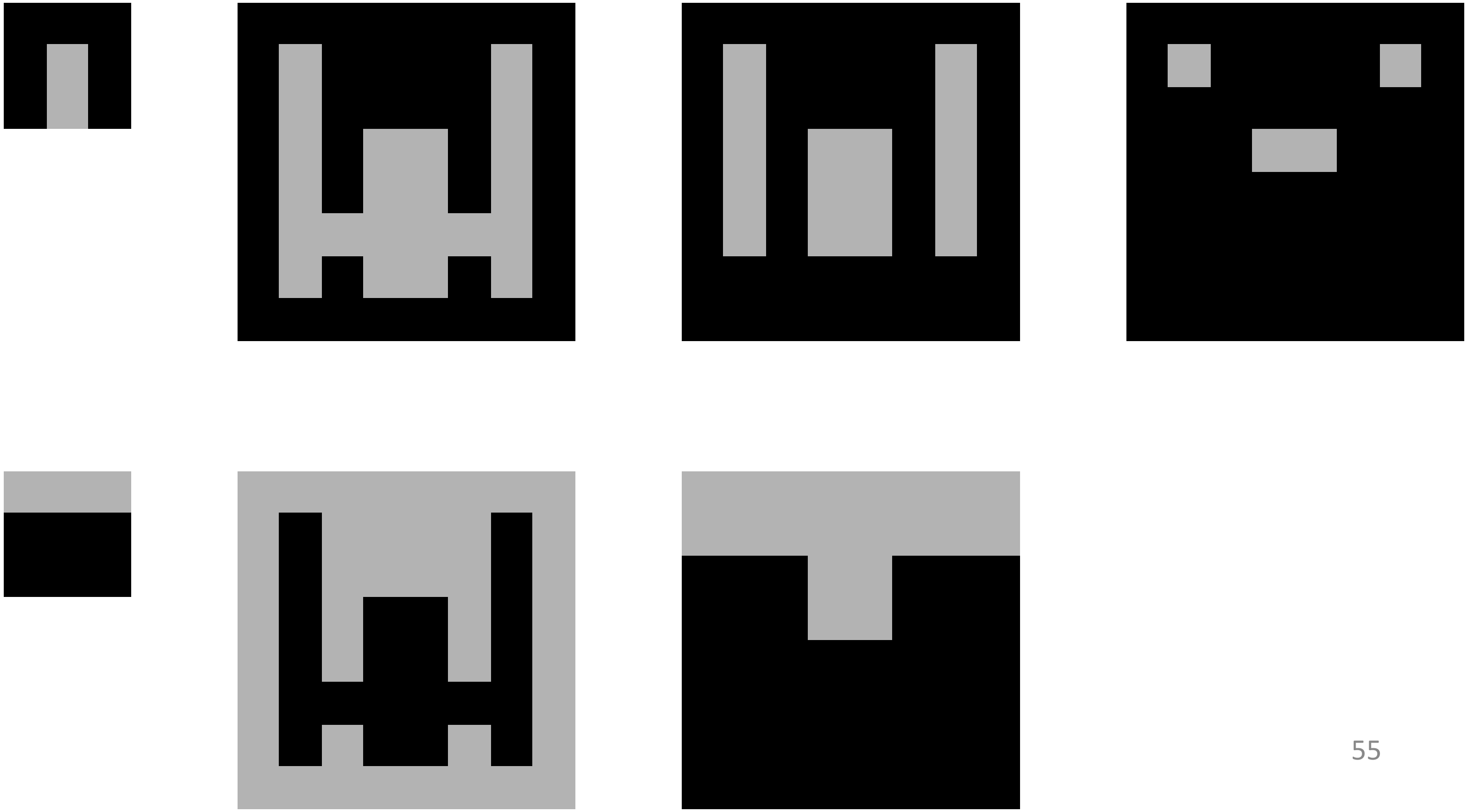
Shape detection

- Local maximums have upper neighbors in the BG → we need to detect them



Shape detection

- Local maximums have upper neighbors in the BG → we need to detect them



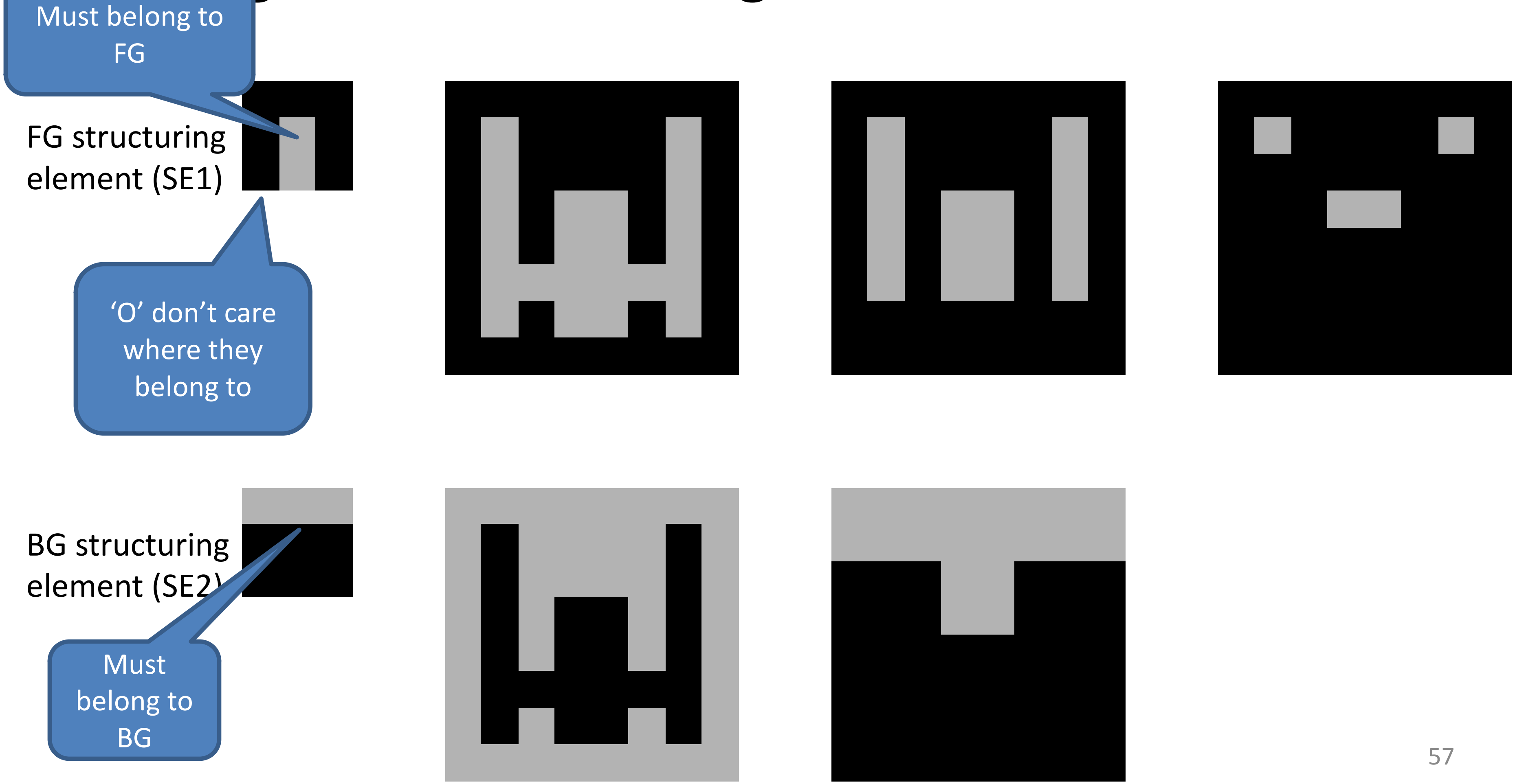
code

What is Hit-and-miss Transform ?

- Similar to Pattern Matching
- The output is set to one if:
  - foreground pixels exactly matches the foreground structuring element (SE1)
  - background pixels exactly matches the background structuring element (SE2).
- **Combine Erosion on FG and Erosion on BG**
  - bwhitmiss(BW1,SE1,SE2) is equivalent to imerode(BW1,SE1) & imerode(~BW1,SE2).

Hit-and-miss Transform

- Foreground structuring element and background structuring element



Must belong to FG
FG structuring element (SE1)
'O' don't care where they belong to
BG structuring element (SE2)
Must belong to BG

Hit-and-miss Transform

- Used to look for particular patterns of foreground and background pixels

- Very simple object recognition, detect simple geometric shapes

Corner Detection with Hit-and-miss Transform

- Structuring Elements representing four corners

SE FG

0	1	0
0	1	1
0	0	0

0	1	0
1	1	0
0	0	0

0	0	0
1	1	0
0	1	0

0	0	0
0	1	1
0	1	0

SE BG

0	0	0
1	0	0
1	1	0

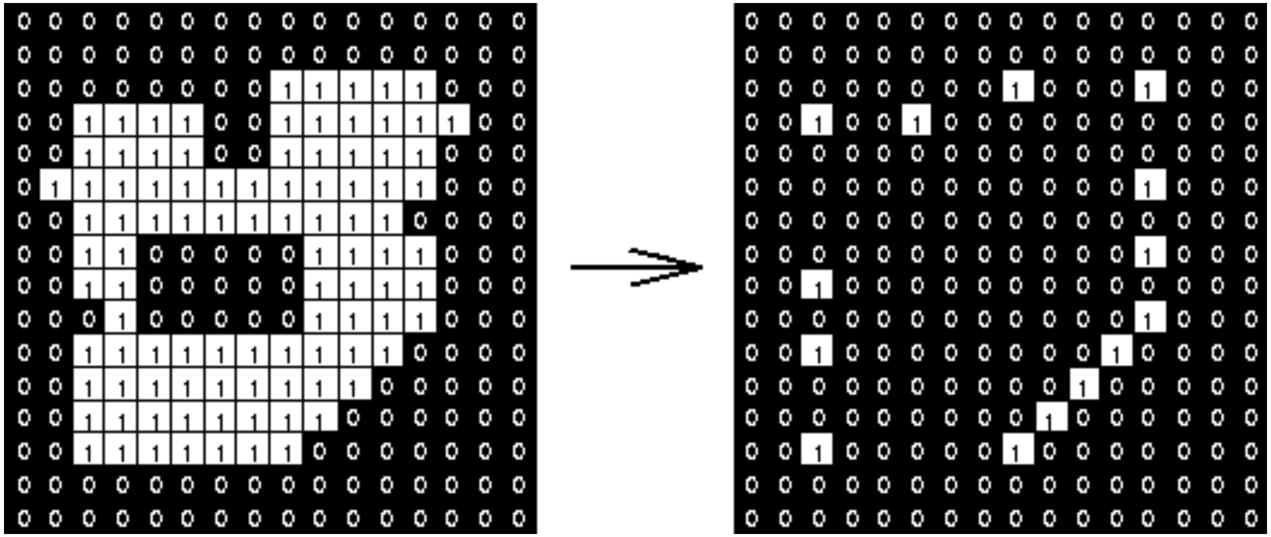
0	0	0
0	0	1
0	1	1

0	1	1
0	0	1
0	0	0

1	1	0
1	0	0
0	0	0

Corner Detection with Hit-and-miss Transform

- Apply each Structuring Element
- Use OR operation to combine the four results



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

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

Basic thinning

What is Thinning ?

- Used to remove selected foreground pixels from binary images

- After edge detection, lines are often thicker than one pixel.

- Thinning can be used to thin those line to one pixel width.

Definition of Thinning

- Let ***K*** be a kernel and ***I*** be an image

$$\mathrm{thin}(I, Kf, Kb) = I - \mathrm{HitAndMiss}(I, Kf, Kb)$$

with 0-1=0!!

- If foreground and background fit the structuring element exactly, then the pixel at the origin of the SE is set to 0

- Note that the value of the SE at the origin is 1 or *don’t care*!

What is Thinning used for?

Thinning is useful in image processing applications such as character recognition, fingerprint analysis, and medical image analysis. By reducing the width of the foreground objects, thinning can help to simplify the object representation and reduce the computational complexity of subsequent processing steps.

Thickening

What is Thickening ?

Thickening is a morphological operation that is used to increase the width of the foreground objects in a binary image while preserving their topological structure. It is essentially the opposite of thinning.

Definition Thickening

- Let ***K*** be a kernel and ***I*** be an image

$$\mathrm{thicken}\left(I,K\right)=I+\mathrm{HitAndMiss}\left(I,K\right)$$

with 1+1=1

- If foreground and background match exactly the SE, then set the pixel at its origin to 1!
- Note that the value of the SE at the origin is 0 or *don’t care*!

What is thickening used for?

Thickening is useful in image processing applications where it is desirable to increase the size or width of foreground objects in a binary image. For example, in medical image processing, thickening can be used to enhance blood vessels or other structures of interest.

Summary

- Morphological Operations
  - Erosion
    - If the structuring element fits in the foreground object, write "1" at the origin of the structuring element in the output image!
  - Dilation
    - If the structuring element touches the foreground object, write "1" at the origin of the structuring element in the output image!
    - If the SE is not symmetric, it needs reflection
  - Opening
    - erosion followed by a dilation
  - Closing
    - Dilatation followed by an Erosion
  - Hit-and-miss Operation, Thinning, Thickening
    - foreground pixels exactly matches the foreground structuring element (SE1)
    - background pixels exactly matches the background structuring element (SE2).

Thanks