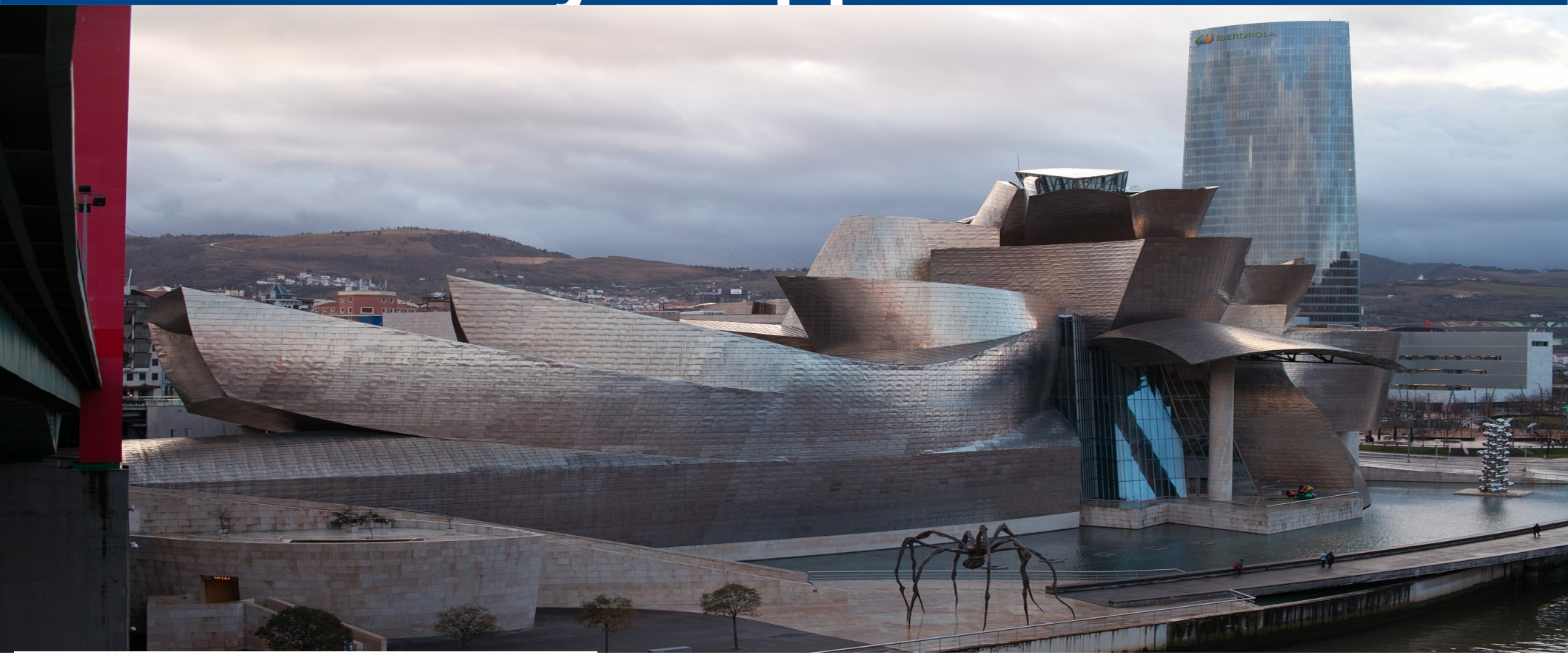
Practical Computer Vision: Theory & Applications



basque center for applied mathematics

Carmen Alonso Montes

calonso@bcamath.org

23rd-27th November 2015

Alternative Software

- Alternative software to matlab
  - Octave
  - Available for Linux, Mac and windows
  - For Mac and windows download it from: https://www.gnu.org/software/octave/download.html
  - For Linux users

    sudo apt-get install octave

    sudo apt-get install octave-image

    sudo apt-get install octave-control octave-image octave-io octave-optim octave-signal octave-statistics
    - Another alternative

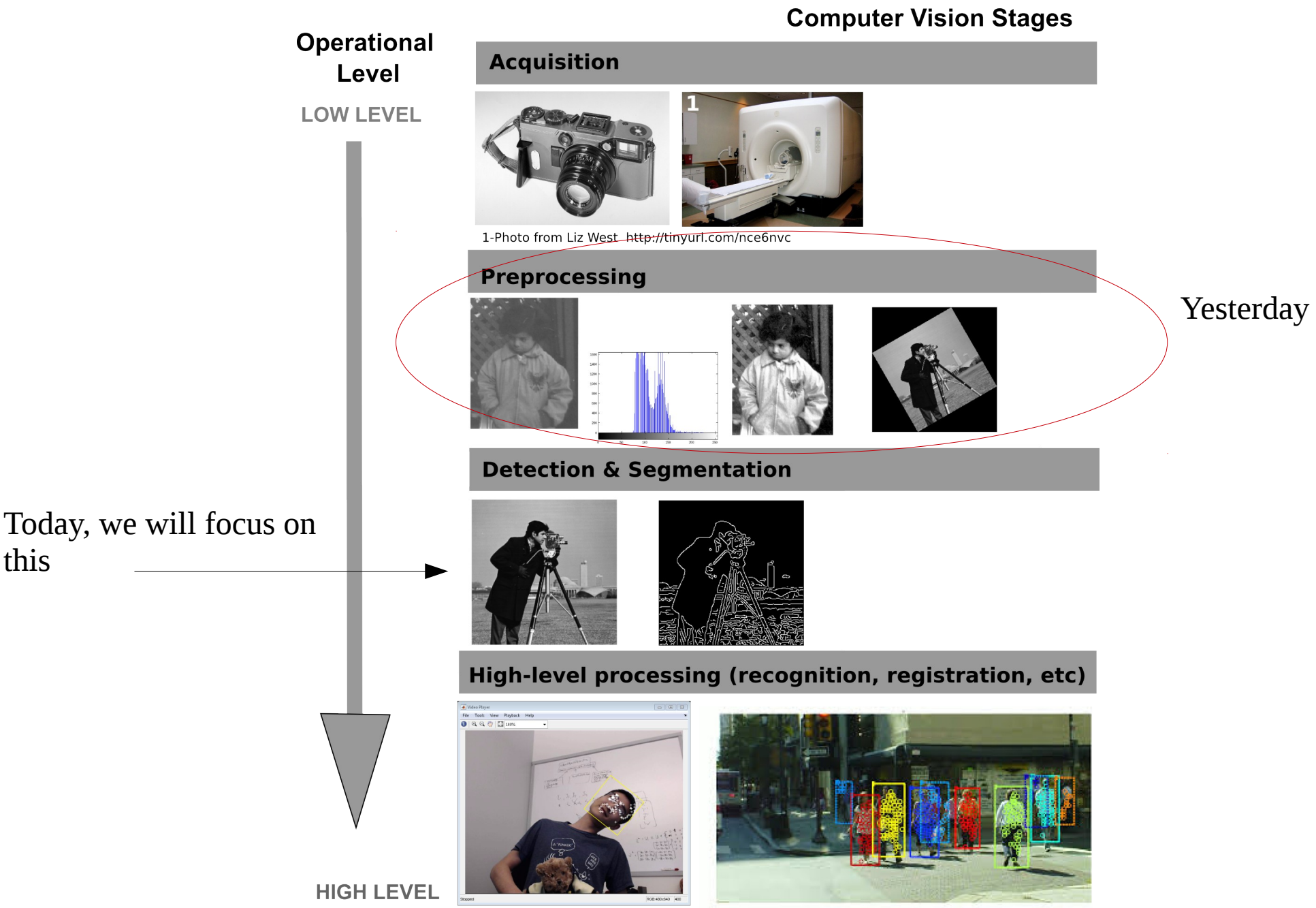
      sudo apt-add-repository ppa:octave/stable

      sudo apt-get update

      sudo apt-get install octave
  - For ubuntu users, install also the Qt-octave as graphical interface
- To load the image package (write this in octave console)

  pkg load image

Wrap up


Computer Vision Stages
Operational Level
LOW LEVEL
HIGH LEVEL
Acquisition
1
1-Photo from Liz West http://tinyurl.com/nce6nvc
Preprocessing
Yesterday
Detection & Segmentation
Today, we will focus on this
High-level processing (recognition, registration, etc)

Learned Concepts

- Noise

- Image Denoising

- Image Enhancement

- Geometric transformations

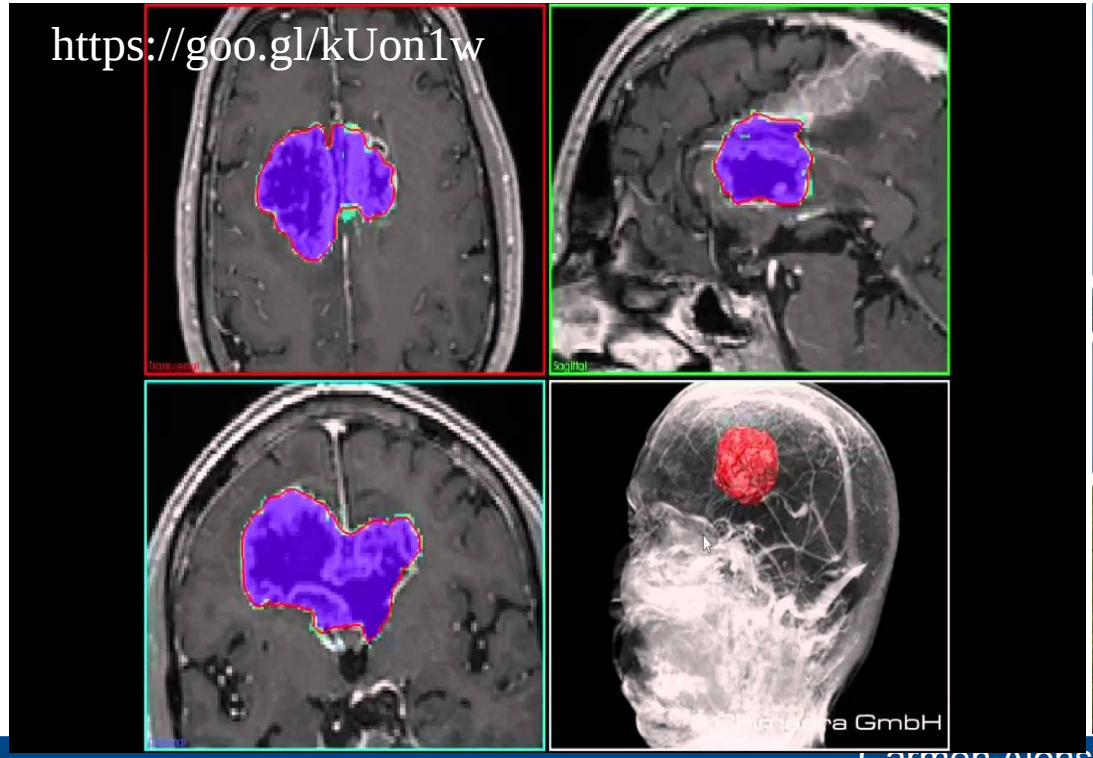
Contents

- Concepts learned last day
- **Image binarization and segmentation**
  - Thresholding
  - Adaptive thresholding
  - Sobel, Prewitt thresholding techniques
  - Canny Edge Detector
- **Morphological operations**
  - Erosion
  - Dilation
  - Opening
  - Closing
  - Skeletonization
- Summary
- Practical exercises

Image Segmentation

Image segmentation

- **Image segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels or regions), which are similar with respect to some characteristic or computed property, such as color, intensity, or texture.
  - **Goal:** to simplify and/or change the representation of an image into something that is **more meaningful** and easier to **analyze**.
- Typically used to **locate objects and boundaries** (lines, curves, etc.) in images; or to assign a **label** to pixels sharing certain characteristics.



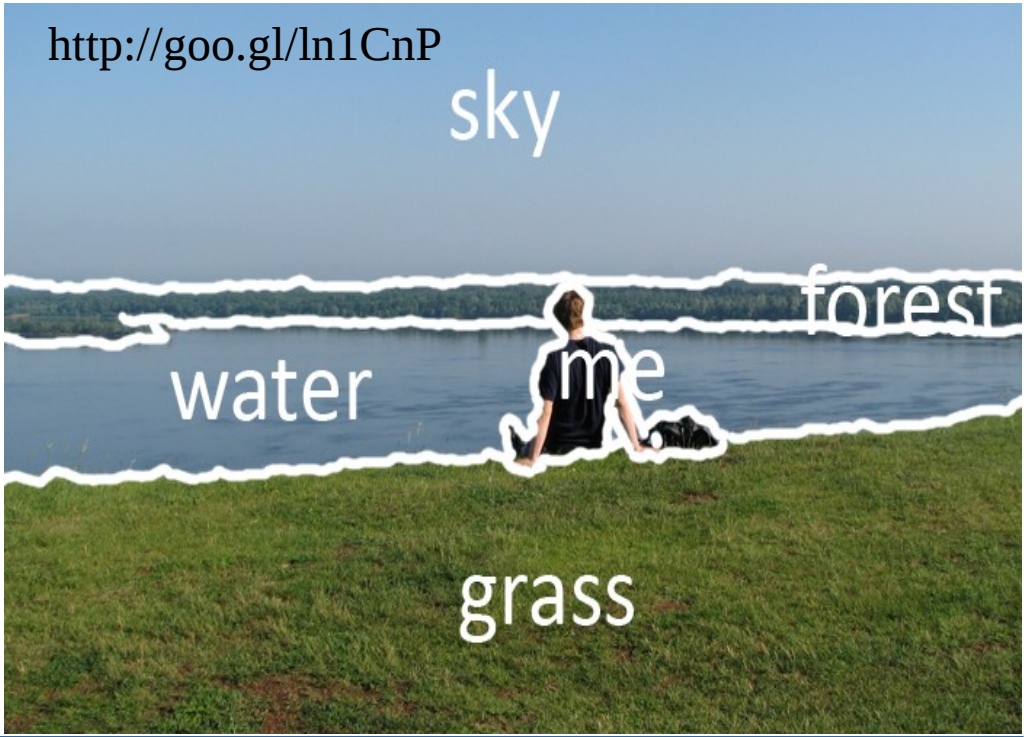
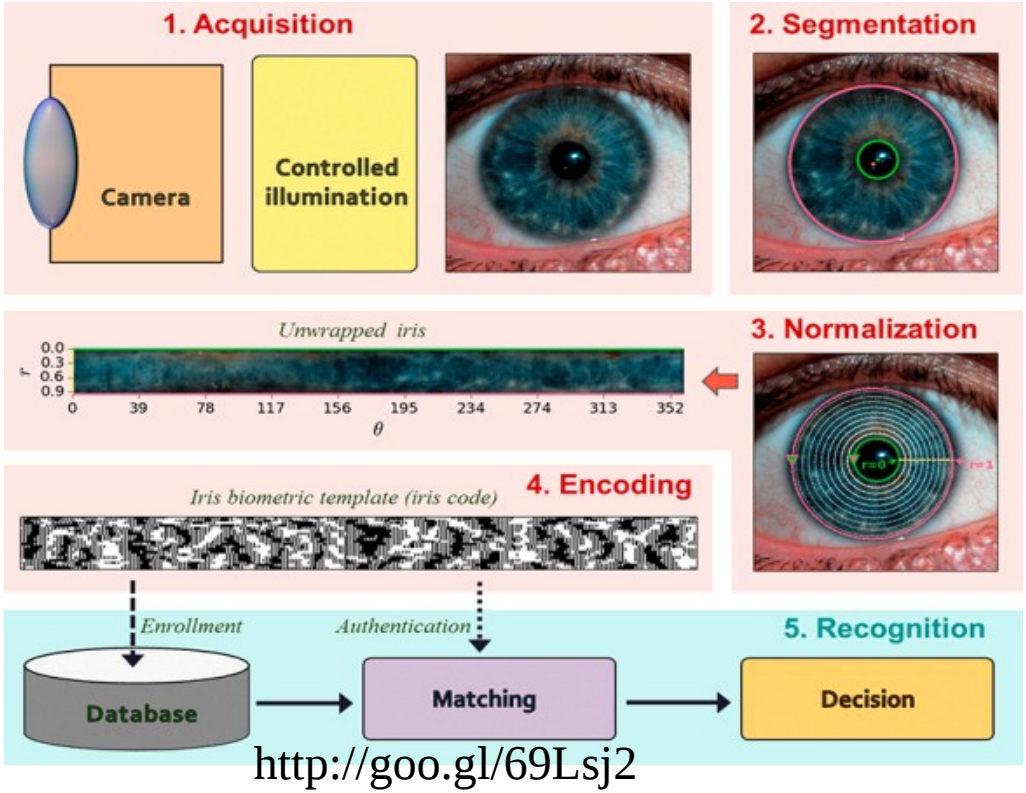


Image Segmentation: Applications

- **Machine vision** (Industry)
- **Medical imaging**
  - Locate tumors and other pathologies
  - Measure tissue volumes
  - Diagnosis, study of anatomical structure
  - Surgery planning
- **Object detection**
  - Pedestrian detection
  - Face detection
  - Locate objects in satellite images
- **Biometrics - Recognition Tasks**
  - Face recognition
  - Fingerprint recognition
  - Iris recognition
- **Video surveillance**



http://goo.gl/69Lsj2

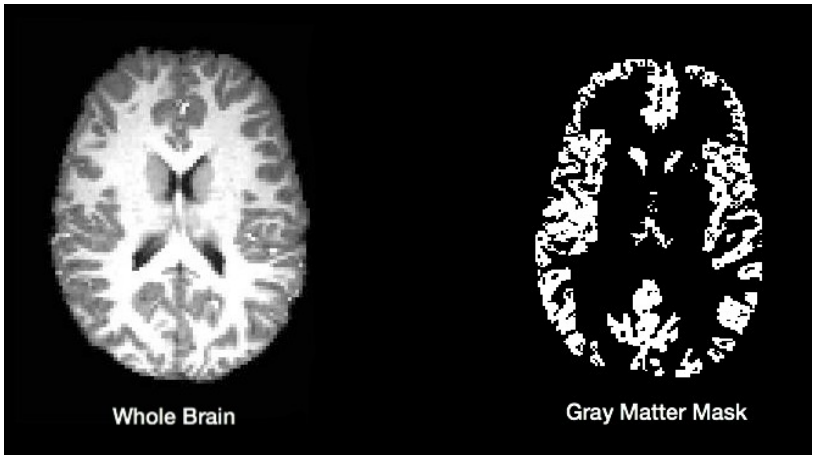
Image Segmentation: Methods

- **Thresholding**
  - Otsu's method
  - Wathersed algorithm
  - Histogram based methods
- **Edge detection**
  - Roberts
  - Sobel
  - Prewitt
  - Canny
  - Zero Crossing Detector
  - Line detection

Image binarization

Image binarization

- A **binary image (black-and-white, B&W)** is a digital image that has only two possible values for each pixel to distinguish **background** vs. **foreground**
  - Traditional [0,255], where 0 is black and 255 is white
  - Or [0,1], where 1 is white
- Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation.



im2bw

Thresholding

Global Thresholding(I)

- **Thresholding** often provides an easy and convenient way to separate background pixels (usually set to black) from those corresponding to the target objects (usually set to white).
- **Variants**:
  - **Multiple thresholds** can be specified, so that a band of intensity values can be set to white while everything else is set to black.
  - Another common variant is to set to black all those pixels corresponding to background, but leave foreground pixels at their original color/intensity (as opposed to forcing them to white), so that that information is not lost.
- **Drawbacks**:
  - Establishing a threshold is not trivial.
  - It uses a **fixed threshold** for all the pixels.
  - It works only if the intensity histogram of the input image contains neatly separated peaks corresponding to the desired object(s) and background(s).
  - It cannot deal with images containing, for example, a strong illumination gradient.

Global thresholding (II)

im2bw

Threshold

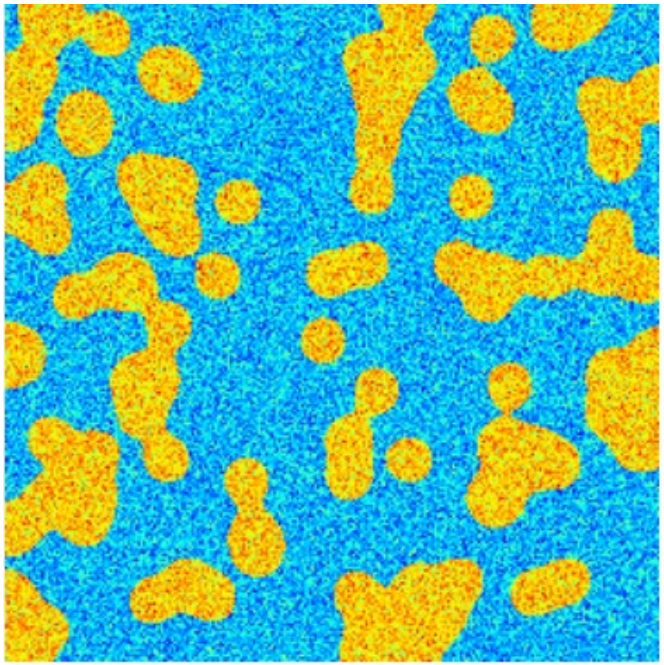
thres = 0.2

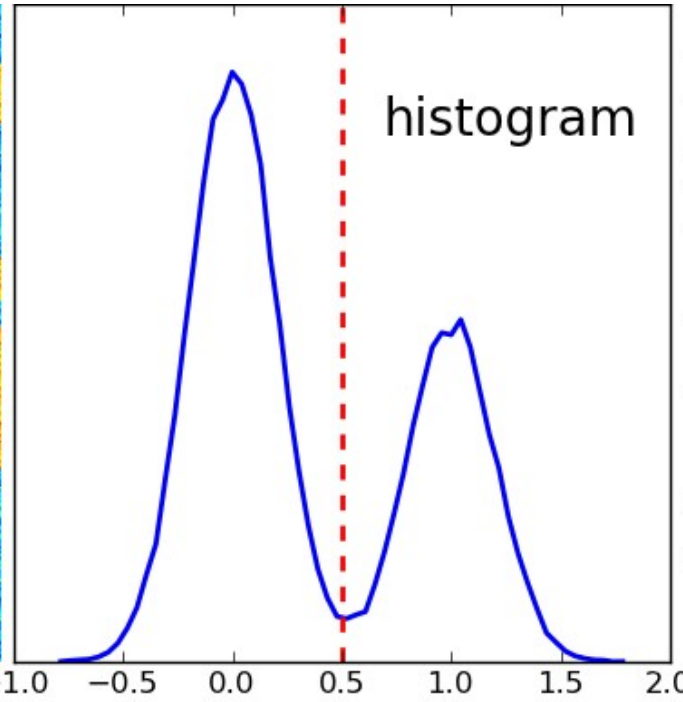
thres = 0.5

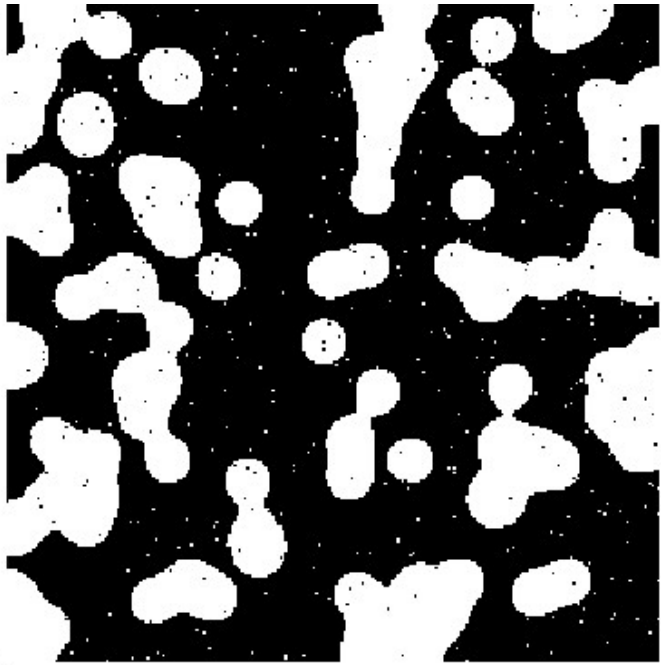
thres = 0.7

Histogram-based thresholding

- The intensity histogram can be used to determine the threshold to separate foreground from background.
  - The intensity of pixels within foreground objects must be distinctly different from the intensity of pixels within the background (peak in the histogram)
  - If such a peak does not exist, then it is unlikely that simple thresholding will produce a good segmentation







Im2bw
imhist

Otsu's method

- Otsu's method, (Nobuyuki Otsu) is used to **automatically** perform **image thresholding**, or, the reduction of a graylevel image to a binary image.
- The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels).
- Then, it calculates the **optimum threshold** separating the two classes
  - intra-class variance is minimal
  - inter-class variance is maximal
- The extension of the original method to **multi-level thresholding** is referred to as the **Multi Otsu method.**

graythresh
im2bw

Otsu Thresh=0.3451

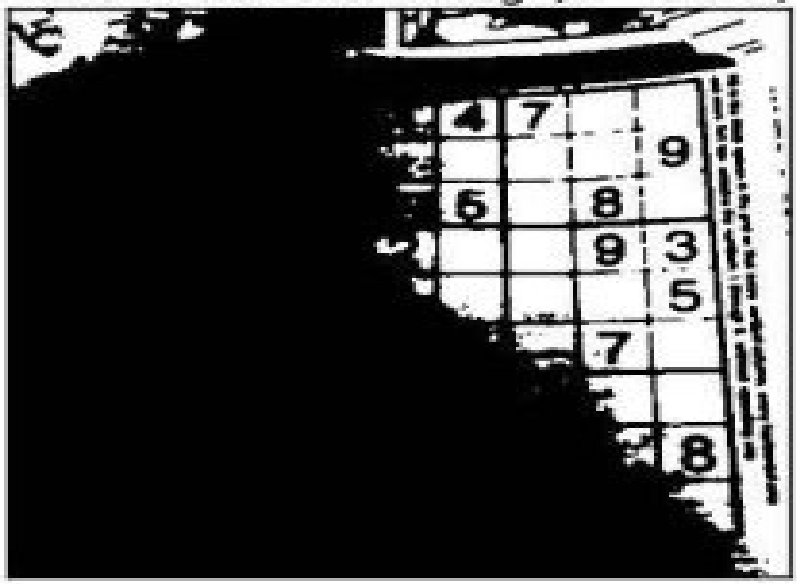
Adaptive Thresholding

- **Adaptive thresholding** changes the threshold dynamically over the image, to handle changing lighting conditions in the image, e.g. those occurring as a result of a strong illumination gradient or shadows.
- **Local adaptive thresholding** select the threshold based on the analysis of local neighbourhood area.
- The assumption is that smaller image regions are more likely to have approximately uniform illumination.
- This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks.
- Typical methods are:
  - Adaptive Mean thresholding: the threshold value is the mean of the neighbourhood area
  - Adaptive Gaussian thresholding: the threshold value is the weighted sum of neighbourhood values where weights are a gaussian window.

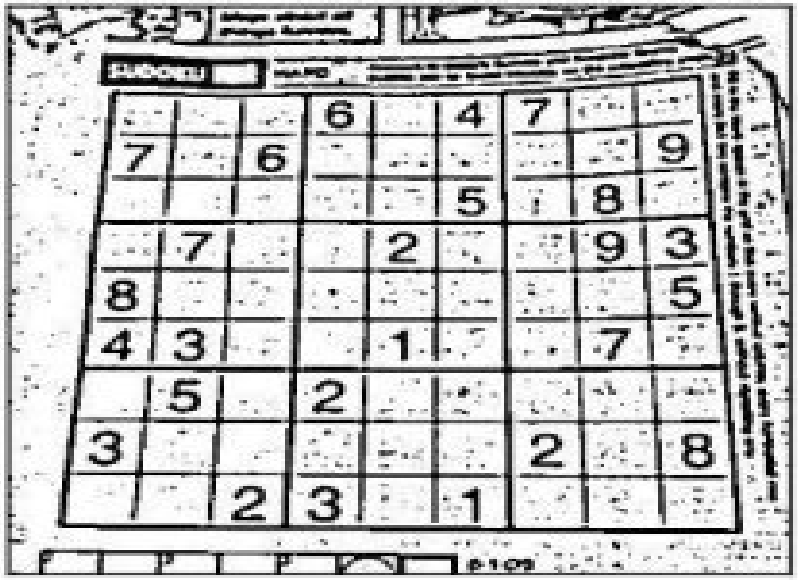
Examples

Original Image

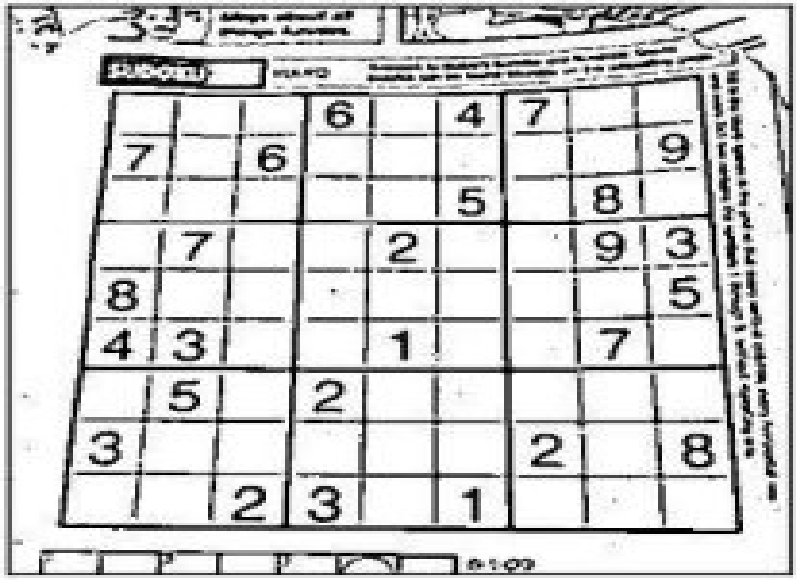
Global Thresholding (v = 127)



Adaptive Mean Thresholding



Adaptive Gaussian Thresholding

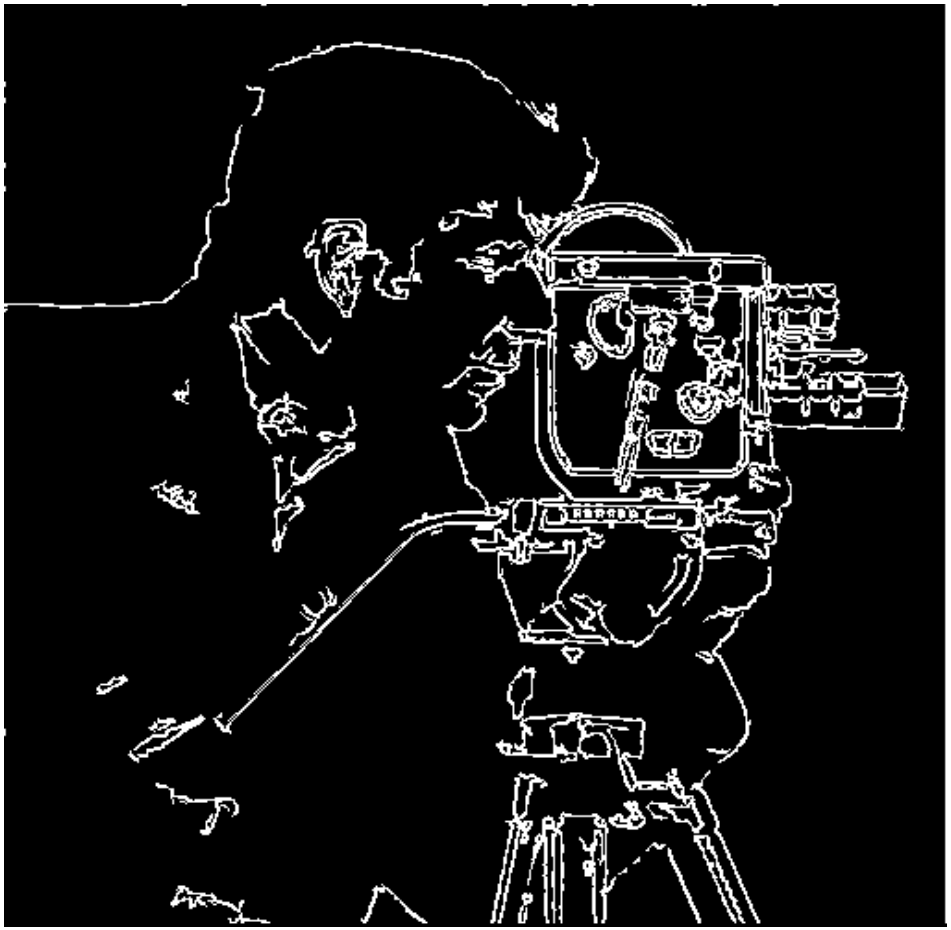


http://goo.gl/9dYIBO

Edge Detection

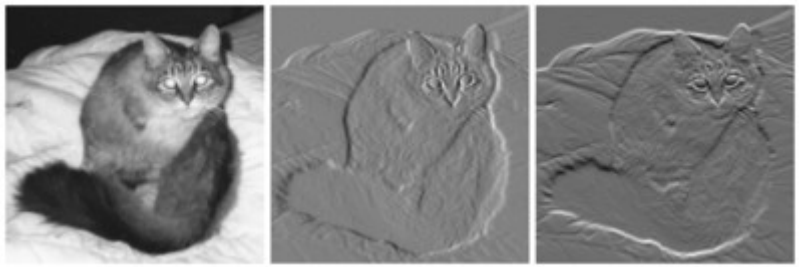
Edge Detection

- **Edge detection** is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply. The aim is to find the **boundaries** of objects within images.
- Edge detection is a crucial tool particularly in the areas of feature detection and feature extraction; of image segmentation and data extraction

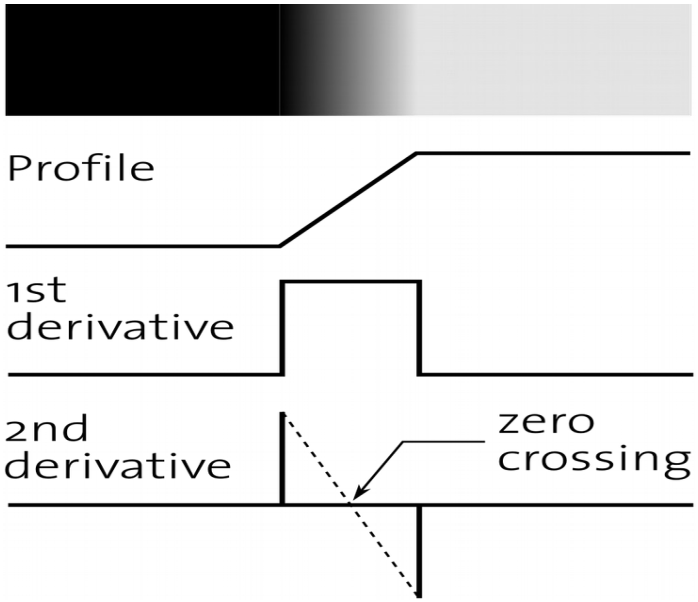


Edge definition

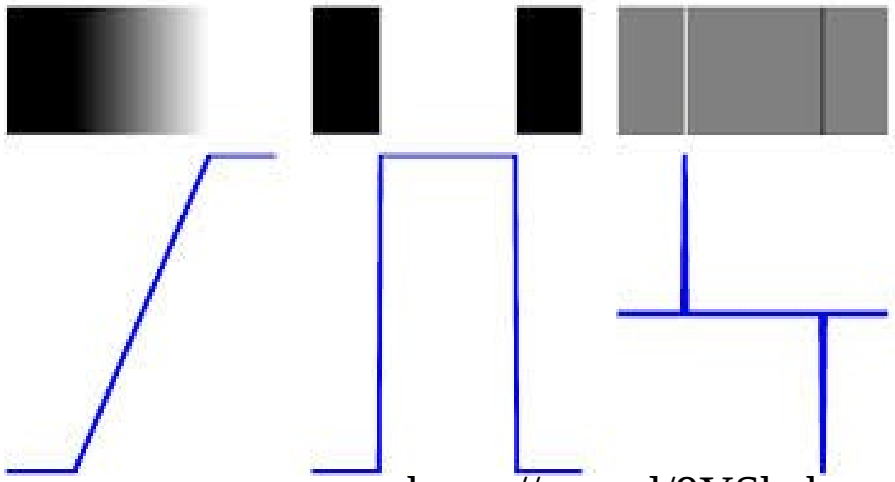
- An **edge** is a place in the image where there is a rapid change in any property of the image, e.g. intensity value
- The usage of derivatives is a good tool for the detection of rapid changes.
  - Drawback: sensitivity to noise



x direction y direction



https://goo.gl/bgI9k0



https://goo.gl/9YSbeh

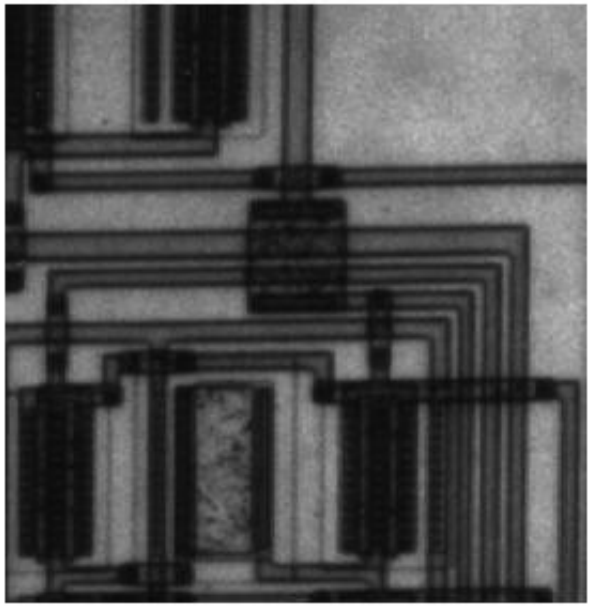
Edge Detection Techniques

- Several techniques can be applied to detect object boundaries:
  - 2-D Spatial Gradient operator
    - Roberts Cross Edge Detector
    - Sobel Edge Detector
    - Prewitt
  - Multistage operator
    - Canny
  - Laplacian / Laplacian of Gaussian detectors
    - Zero Crossing Detector

2-D Spatial Gradient operator & Canny

Roberts Cross Edge Detector

- The Roberts Cross operator performs a **2-D spatial gradient measurement** on an image
  - It will highlight changes in intensity in a **diagonal direction**.
  - Advantages:
    - The kernel is small and contains only integers
    - Very quick to compute.
  - Disadvantages:
    - It is very sensitive to noise.



$$\nabla I(x,y) = G(x,y) = \sqrt{G_x^2 + G_y^2}.$$

+1	0
0	-1

Gx

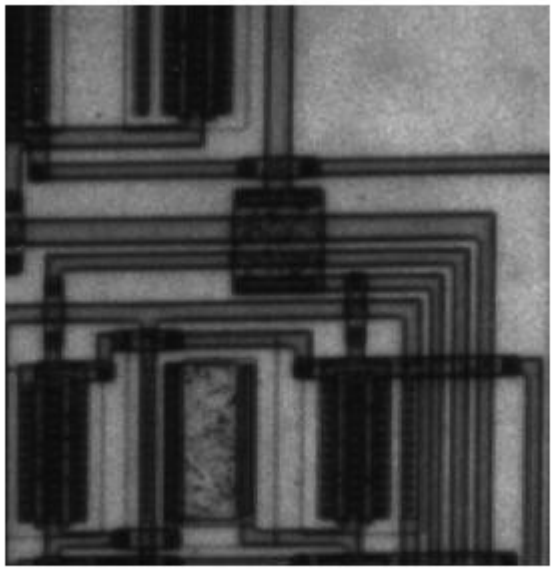
0	+1
-1	0

Gy

edge

Sobel Edge Detector

- The Sobel operator performs a **2-D spatial gradient measurement** on an image
  - To find the **approximate absolute gradient** magnitude at each point
  - Advantages:
    - **Less sensitive to noise** due to its larger convolution kernel, which smooths the input
  - Disadvantages:
    - **Slower** compared to the Roberts Cross operator
    - Final lines in the output image can be artificially thickened due to the smoothing
    - A postprocessing thinning operation is required



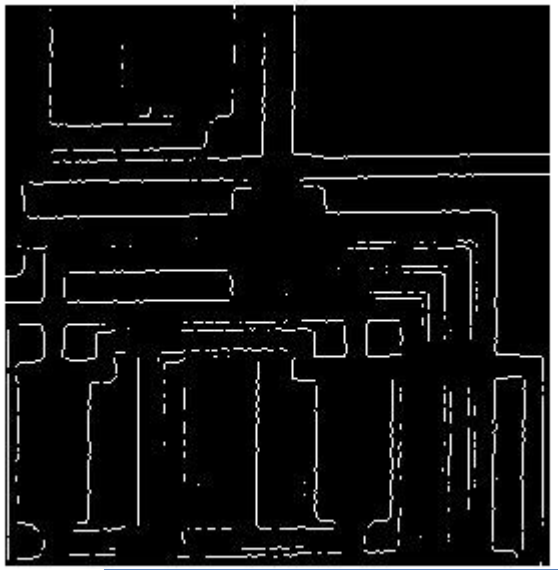
$$G = \sqrt{G_x^2 + G_y^2}$$

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

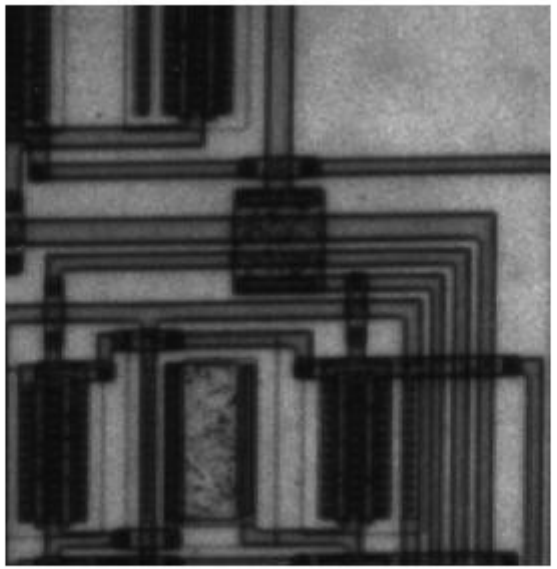
Gy

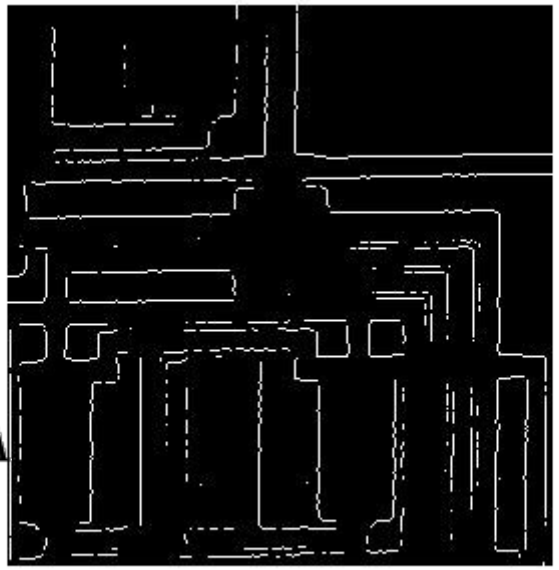


edge

Prewitt Operator

- The Prewitt operator is **2-D spatial gradient measurement that** computes an approximation of the gradient of the image intensity function.
  - At each point in the image, the result is either the corresponding gradient vector or the norm of this vector.
    - Direction of change
    - Rate of change
  - Advantages
    - Quick and fast
    - It shows the magnitude of the change and its direction, directly linked with the likelihood of being a real edge
      - The magnitude (likelihood of an edge) calculation is more reliable and easier to interpret than the direction calculation.





edge

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

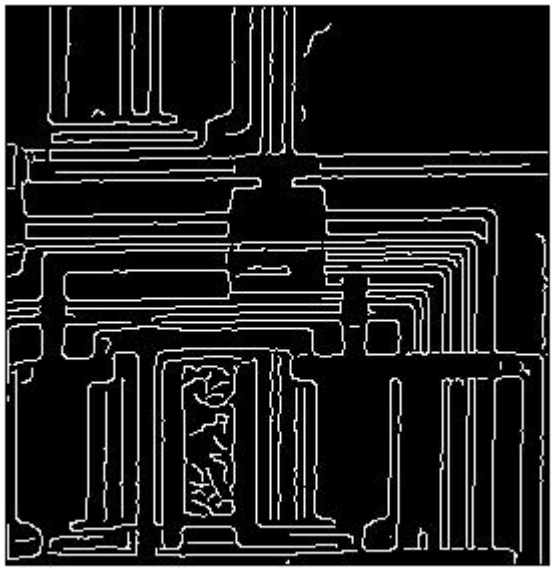
$$G = \sqrt{G_x^2 + G_y^2}$$

Canny Edge Detector

- The Canny operator is a multi-stage algorithm designed to be an **optimal** edge detector. Its steps are:
  - Gaussian convolution smoothing to remove noise (Blurring)
  - Find the intensity gradients of the image: An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to **detect horizontal, vertical and diagonal edges**
  - **Edges thinning** through non-maximum suppression.
  - Apply **double threshold** to remove spurious responses due to intensity variation.
  - **Edge tracking by hysteresis**:
- **Issues:** In Y-junctions, The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn't quite connect to, that line segment.

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A.$$

$$G = \sqrt{G_x^{\,2} + G_y^{\,2}}$$



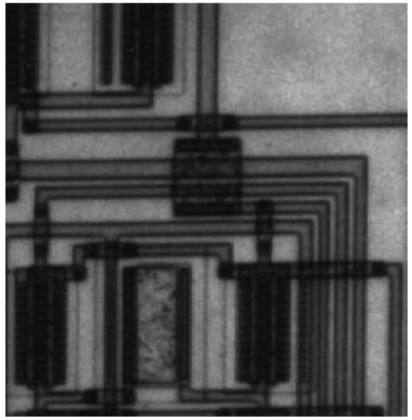
edge

Canny steps in more detail

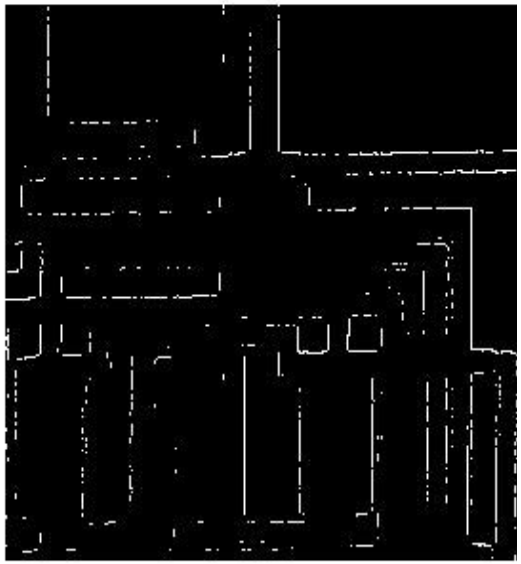
- Gaussian convolution smoothing to remove noise (Blurring)

  Tip: Increasing the width of the Gaussian kernel reduces its sensitivity to noise

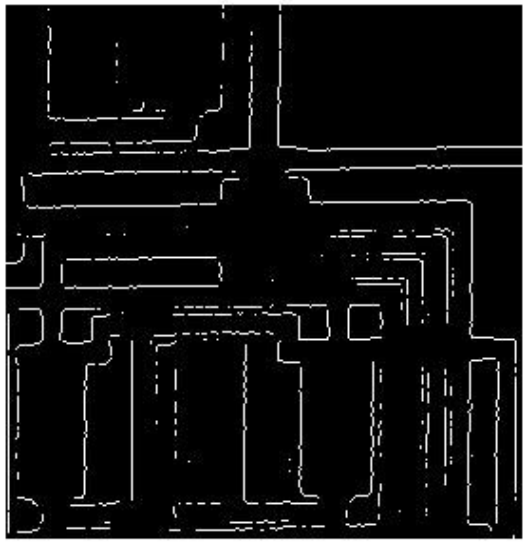
- Find the intensity gradients of the image: An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to **detect horizontal, vertical and diagonal edges**
- **Edges thinning** through non-maximum suppression.
  - The local maximal in the gradients indicates location with the sharpest change of intensity value
  - After this step, the edge pixels are quite accurate to present the real edge.
- Apply **double threshold** to remove spurious responses due to intensity variation.
  - Thresholds: T1 and T2, with T1 > T2.
  - If edge pixel's gradient > T1 → **strong edge pixels**.
  - If T2 < edge pixel's gradient < T1 → **weak edge pixels**.
  - If edge pixel's gradient < T2 → supress
  - **Tip**: T1 can be set quite high, and T2 quite low for good results.
    - Setting T2 too high will cause noisy edges to break up.
    - Setting T1 too low increases the number of spurious and undesirable edge fragments
- **Edge tracking by hysteresis**:
  - **Strong edge** pixels will be in the output
  - **Weak edge** pixels coming from real edges will be connected to the strong edge pixel. (BLOB analysis).



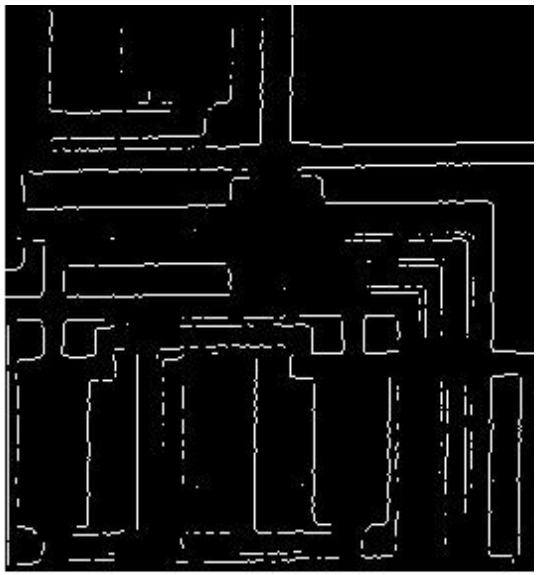
Roberts



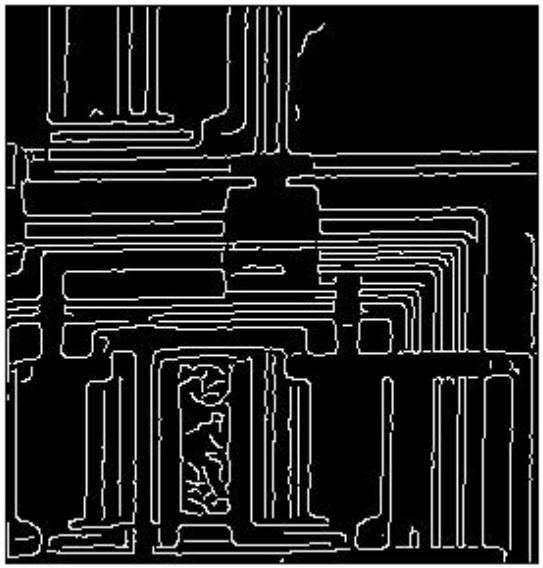
Sobel



Prewitt



Canny



edge

Laplacian

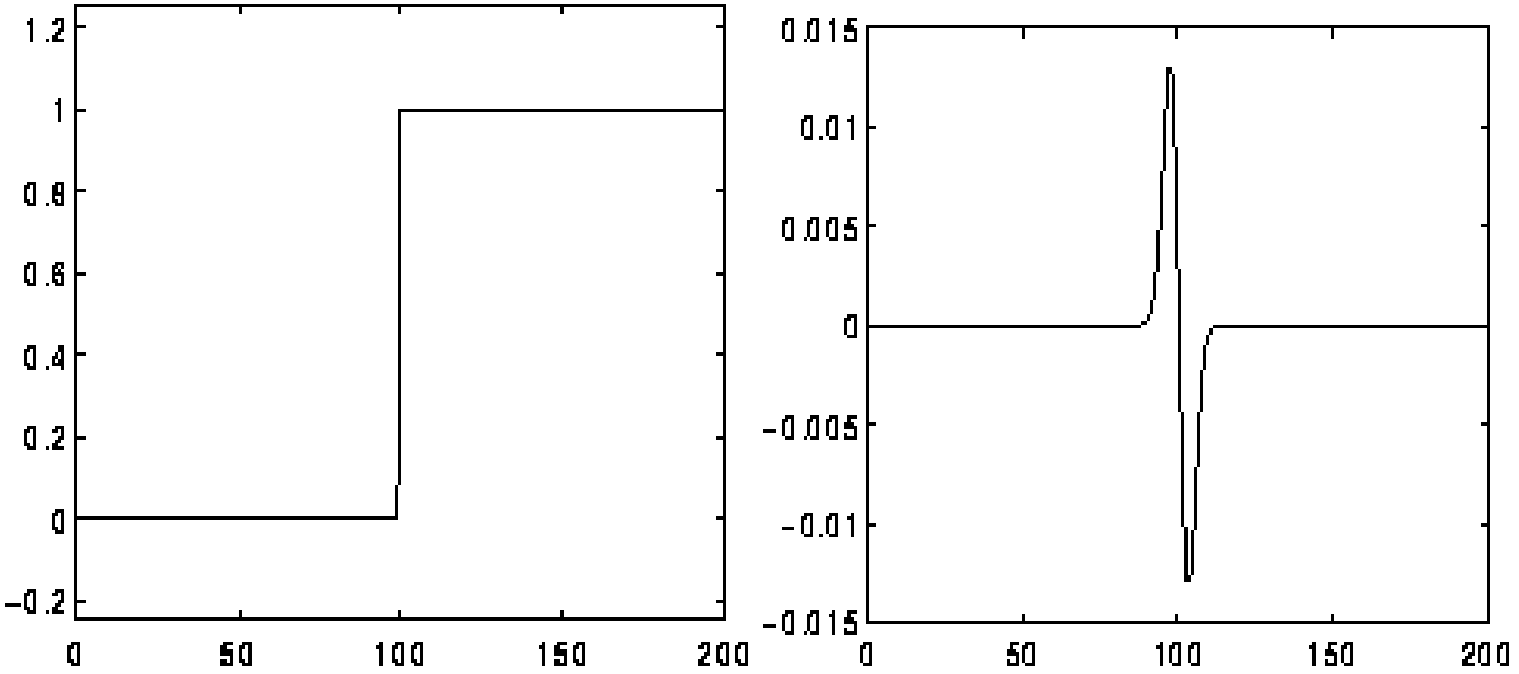
- The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image.
- The Laplacian of an image highlights regions of rapid intensity change (**edges**)
- The input image must be smoothed before to reduce noise

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

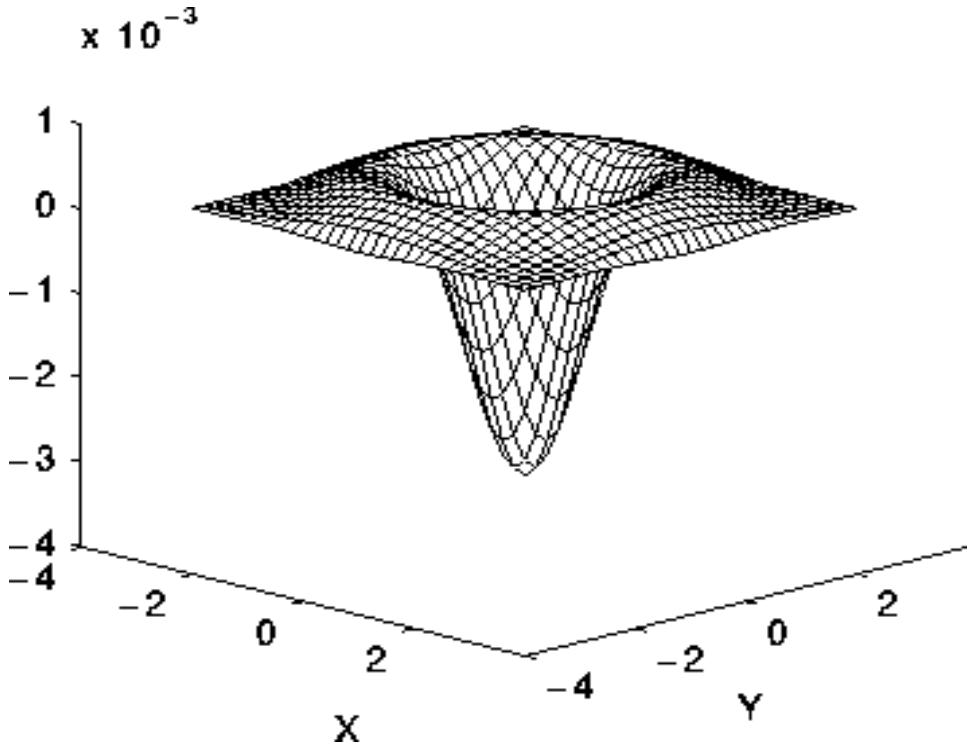
Laplacian

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1



Laplacian of Gaussian (LoG)



$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Laplacian/Log: Example

Laplacian

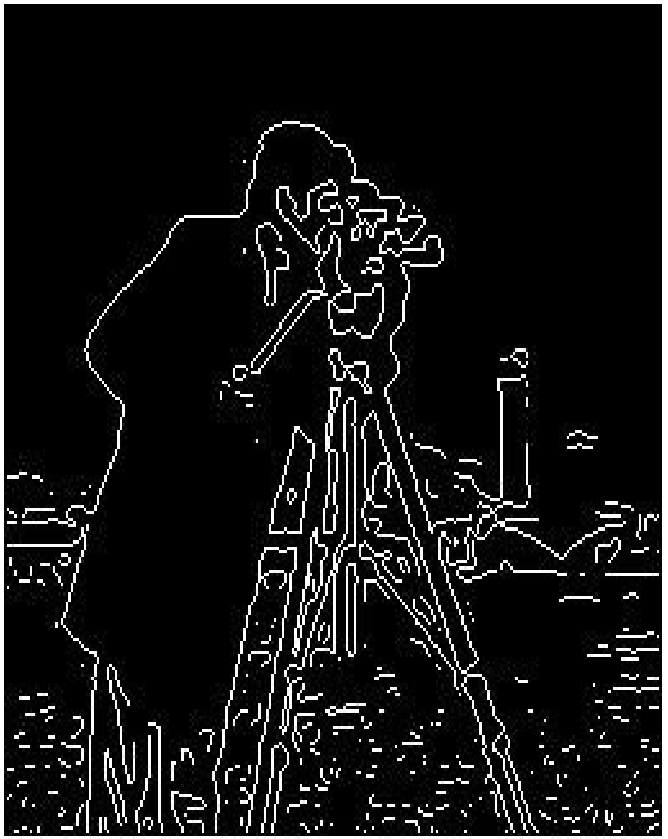
LoG

edge

Zero Crossing Detector

- The zero crossing detector looks for places in the **Laplacian of Gaussian** of an image where the value of the Laplacian passes through zero, which correspond to
  - Edges
  - Places that are not edges
- **Strongly influenced by the size of the Gaussia**n smooth kernel. Increments on the size of the smoothing kernel implies less contours to be found.
- Steps:
  - Smooth Gaussian filter
  - Apply LoG filter
  - **Identify the zero crossing points,** since the zero crossings generally fall in between two pixels
    - **Threshold** the LoG output at zero, to produce a binary image where the boundaries between foreground and background regions represent the locations of zero crossing points.
      - **Issue**: the location of the zero crossing edge maybe falls to either the light side (foreground regions), or the dark side of the edge (background regions)
    - Selection of the point with lowest absolute magnitude of the Laplacian considering both sides of the threshold boundary
    - **Interpolation** to estimate the position of the zero crossing to sub-pixel precision

Zero crossing example



edge

Line Detector

- The **line detection** operator consists of a convolution kernel tuned to detect the presence of lines of a particular width n, at a particular orientation
- **Thresholds** shall be used to remove **weak lines** corresponding to edges and other features with intensity gradients which have a different scale than the desired line width.
- An edge tracking operator shall be used to **join line fragments**.

a)

-1	-1	-1
2	2	2
-1	-1	-1

b)

-1	2	-1
-1	2	-1
-1	2	-1

c)

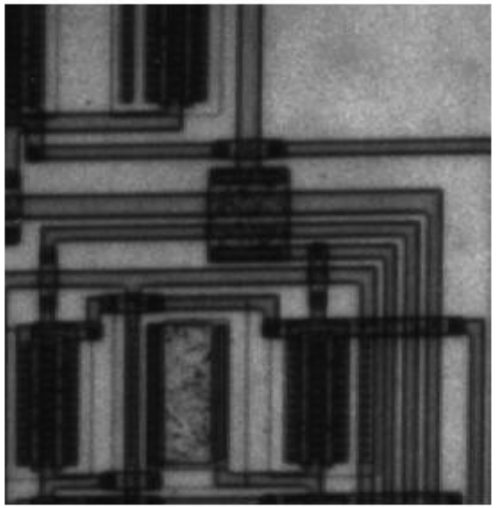
-1	-1	2
-1	2	-1
2	-1	-1

d)

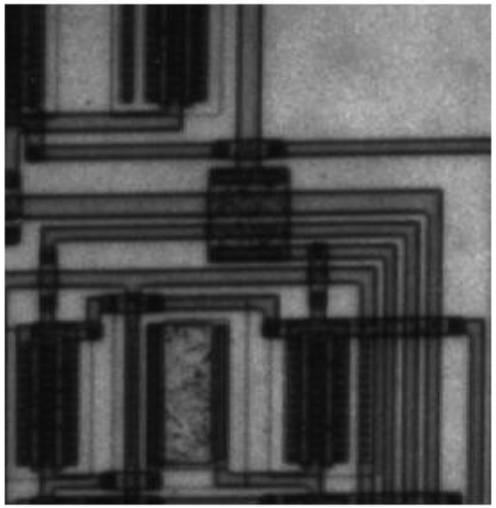
2	-1	-1
-1	2	-1
-1	-1	2

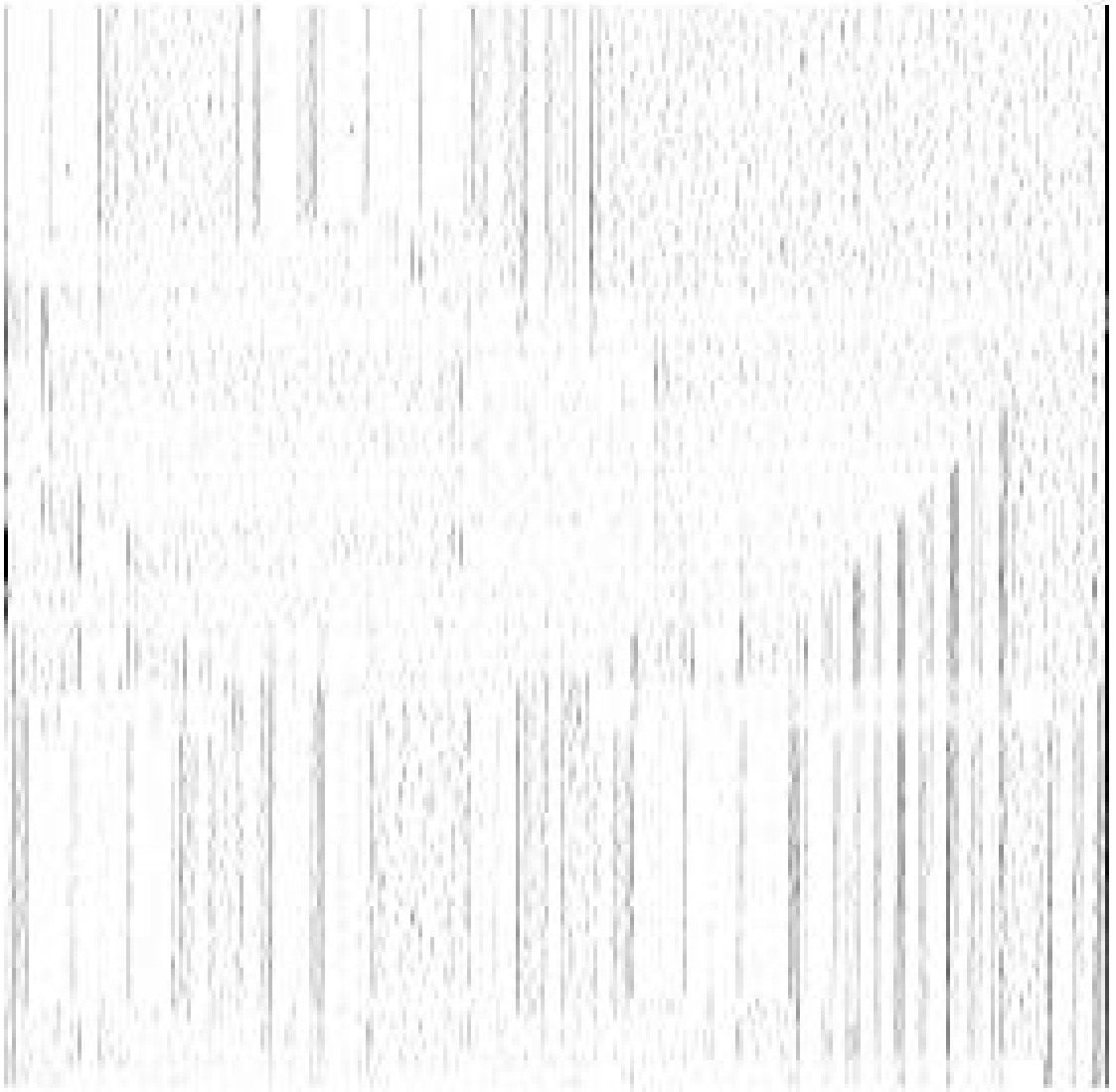
imfilter

Horizontal lines



Vertical lines





Morphological Image Processing

Morphological operations (I)

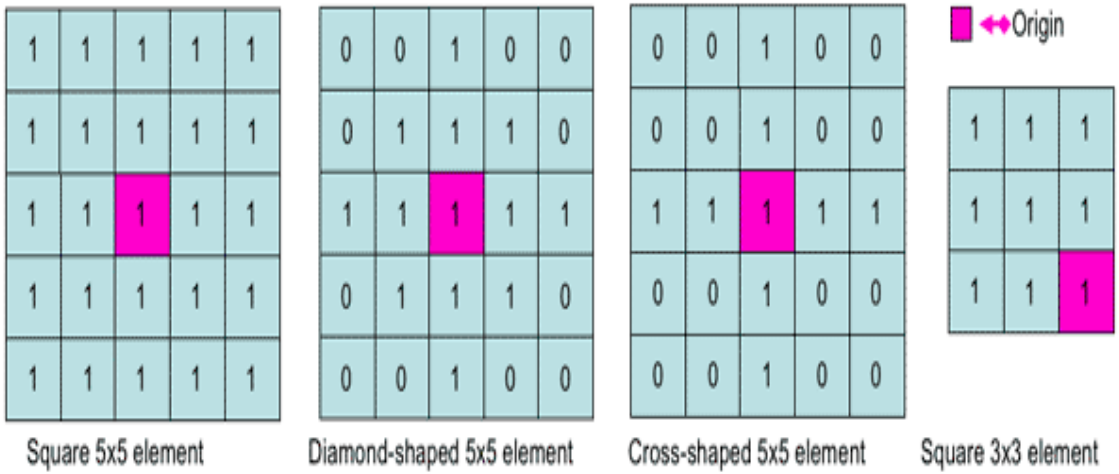
- Binary images may contain numerous imperfections and artifacts from previous processing like thresholding or edge detection, due to noise or intensity fluctuations.
- **Morphological image processing** is a collection of non-linear operations related to the shape or morphology of features in an image.

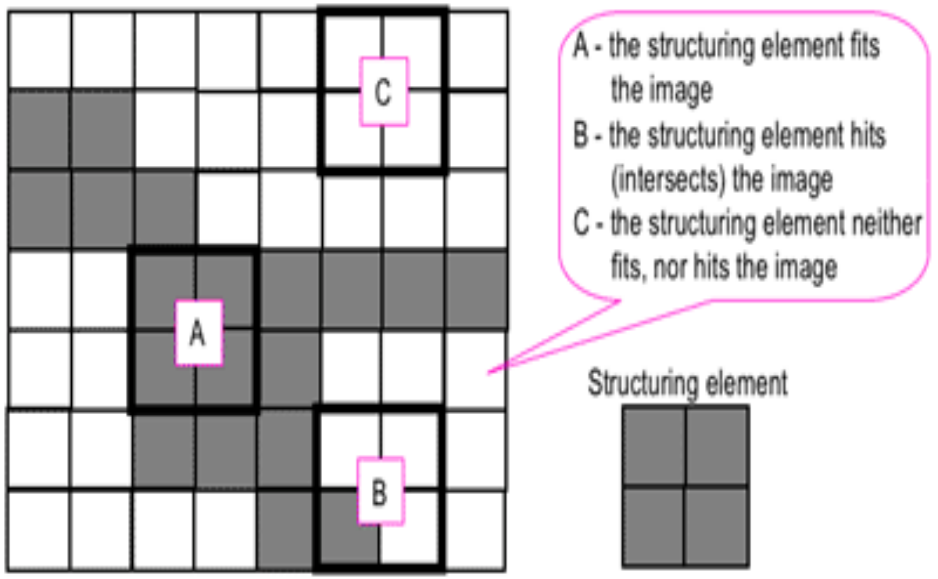
strel
bwmorph

Morphological operations (II)

- The **structuring element** is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels.
  - "fits" within the neighbourhood,
  - "hits" or intersects the neighbourhood
- A morphological operation on a binary image creates a **new binary image** in which the pixel has a non-zero value only if the test is successful at that location in the input image.
  - Notice that these operators can be also applied to gray scale images (out of the scope in this course)
- Operations:
  - Basic: dilation, erosion
  - Compound: Opening, clossing

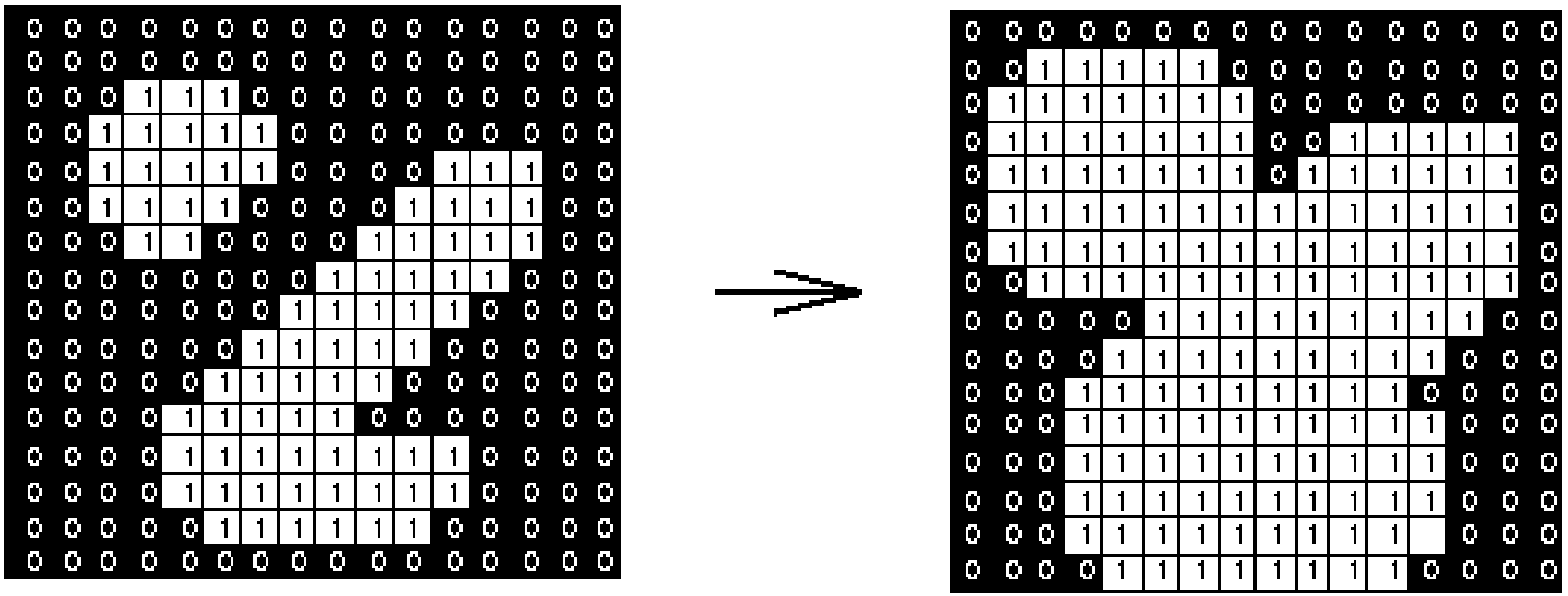
https://goo.gl/GrbE5i





Dilation

- Dilation operation is a a shift-invariant (translation invariant) operator that gradually enlarges the boundaries of regions of foreground pixels
  - Areas of foreground pixels grow in size while holes within those regions become smaller.
- It is this structuring element that determines the precise effect of the dilation on the input image.

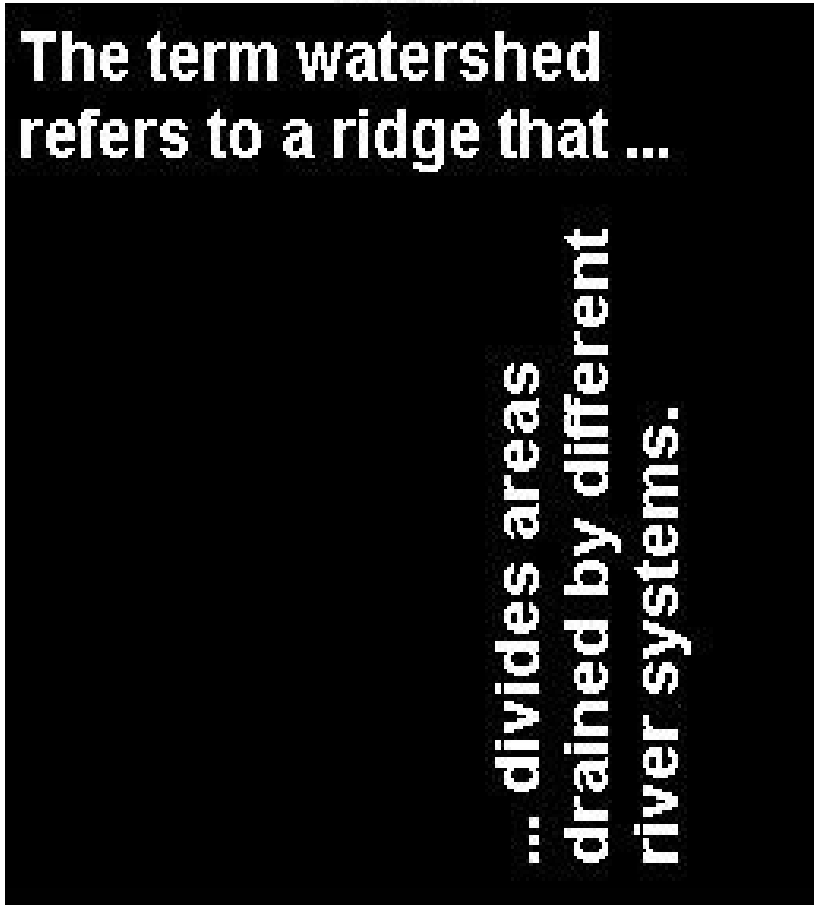


http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm

strel
imdilate

Dilation: example

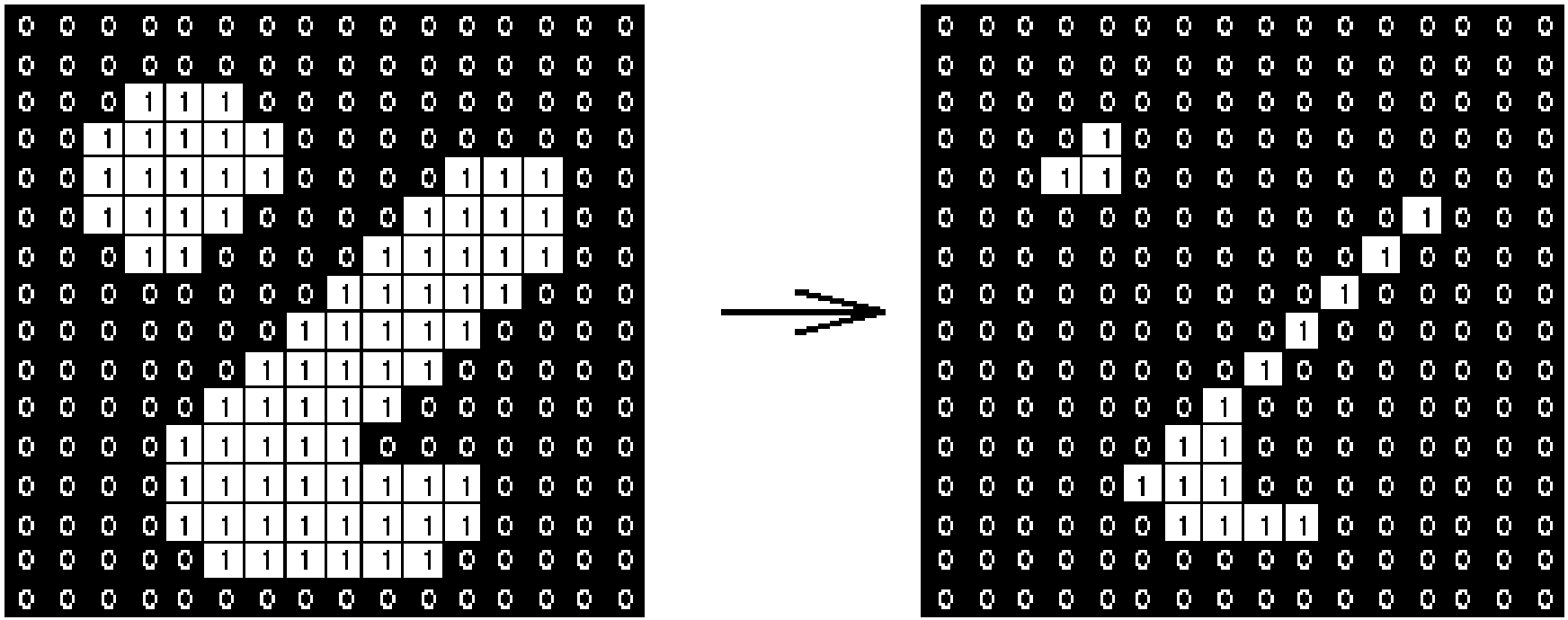
Original



Dilated

Erosion

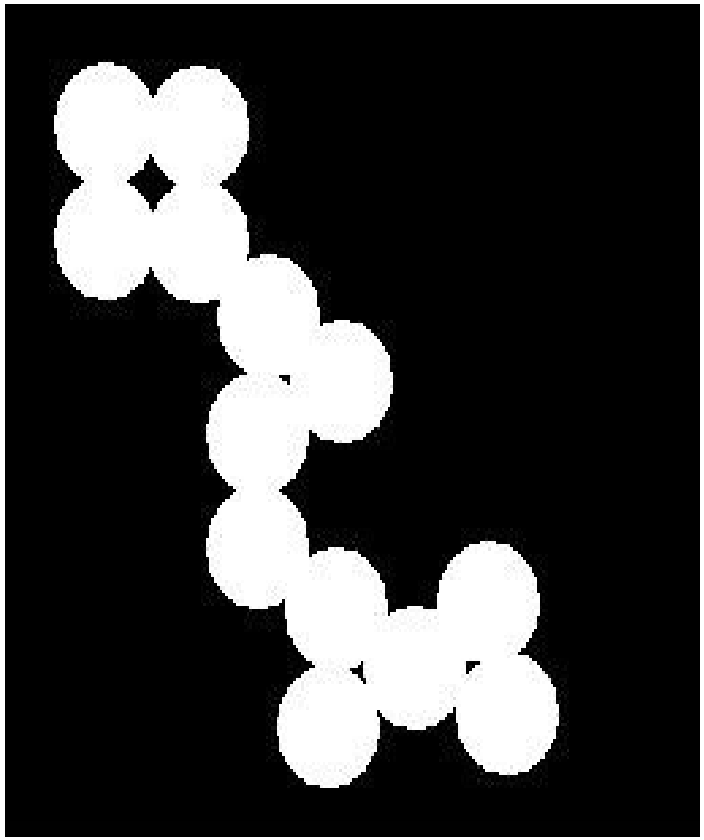
- The erosion operator erodes / removes away the boundaries of regions of foreground pixels
- Areas of foreground pixels shrink in size, and holes within those areas become larger.
- The structure is preserved
- It is usually used to remove noisy points

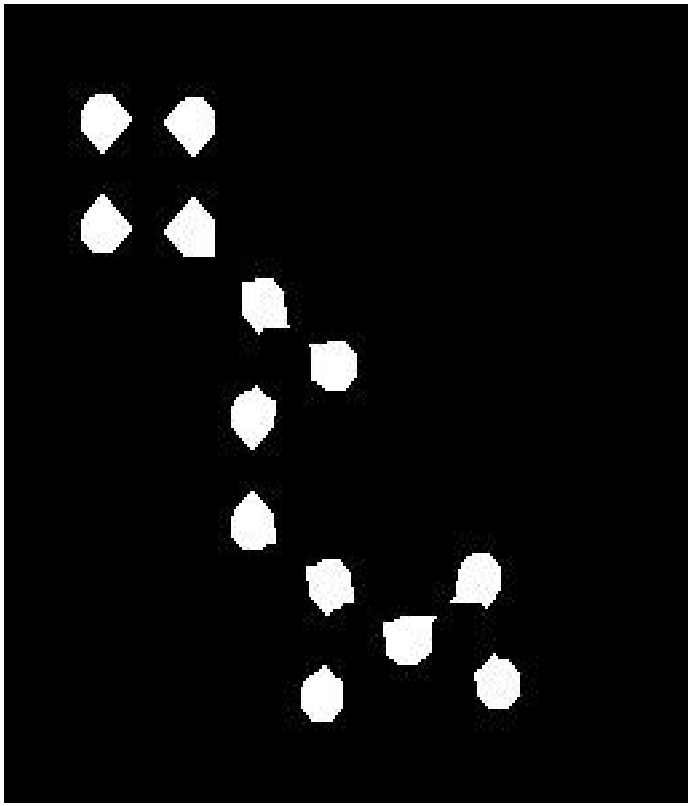


http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm

strel
imerosion

Erosion: example





Combinations of erosion and dilation

- Dilation and erosion are often used in combination to implement image processing operations.
  - **Opening** of an image is an **erosion** followed by a **dilation** with the same structuring element
    - It is used to **remove small objects** from an image while preserving the shape and size of larger objects in the image
  - **Closing** of an image consists of **dilation** followed by an **erosion** with the same structuring element.
    - It can be used to **remove discontinuities in regions**

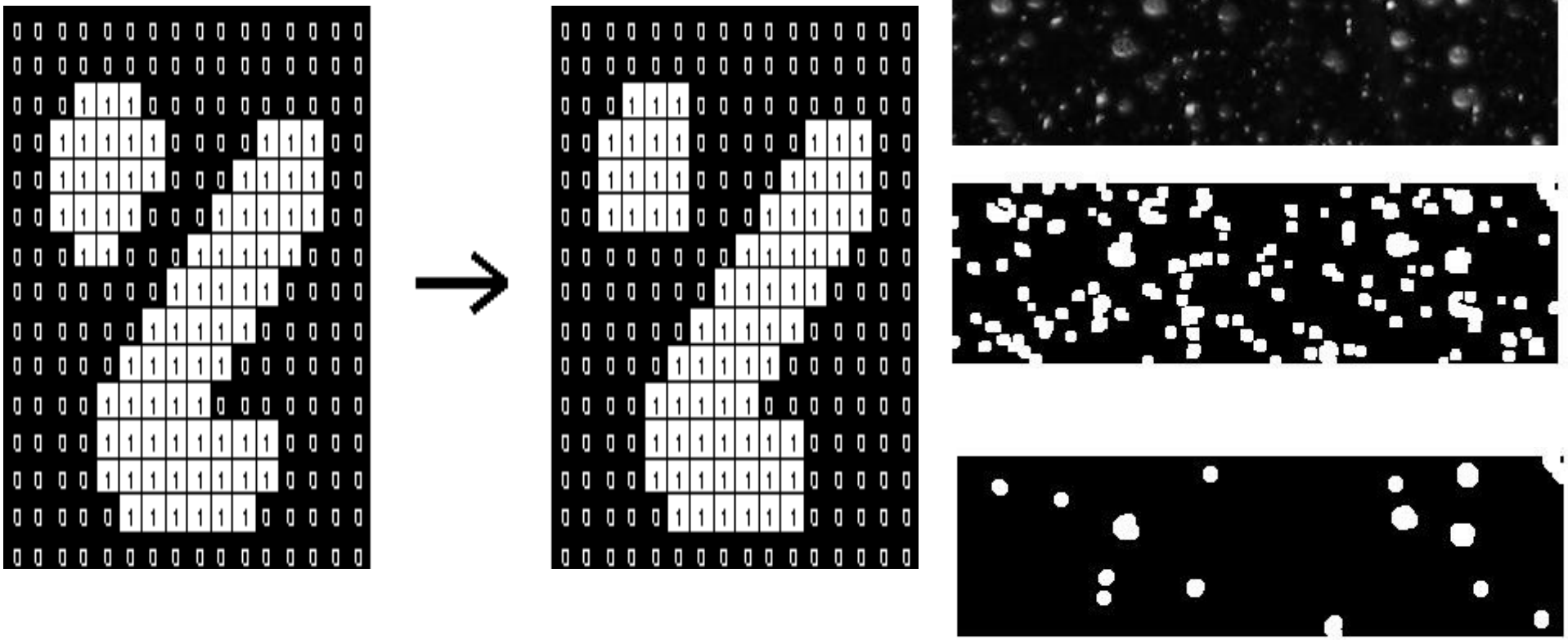
OPENING = EROSION + DILATION

CLOSING = DILATION + EROSION

strel
imopen
imclose

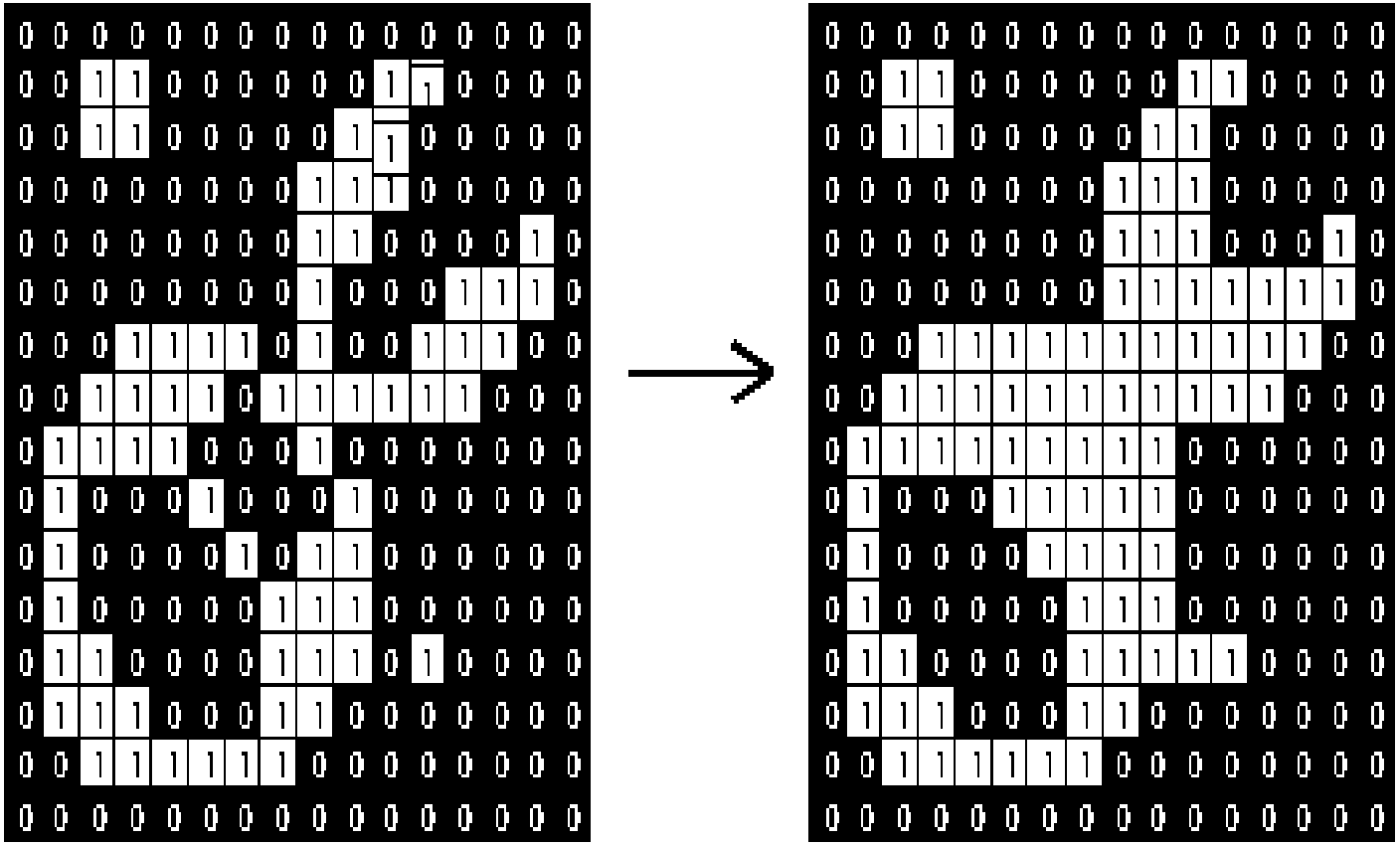
Opening

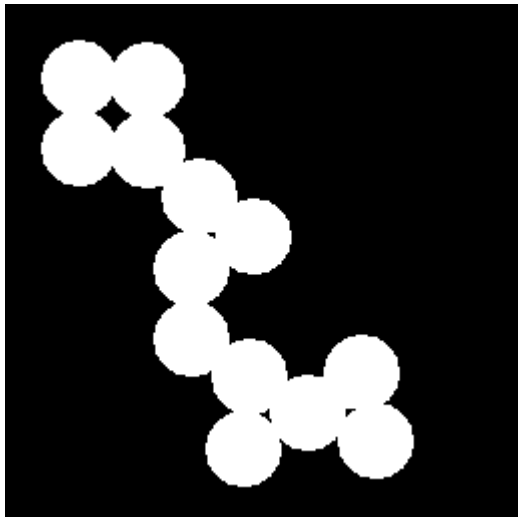
- The opening operation preserves foreground regions that have a similar shape or contains completely the structuring element, while eliminating all other regions of foreground pixels.
- An opening is defined as an erosion followed by a dilation using the same structuring element for both operations.

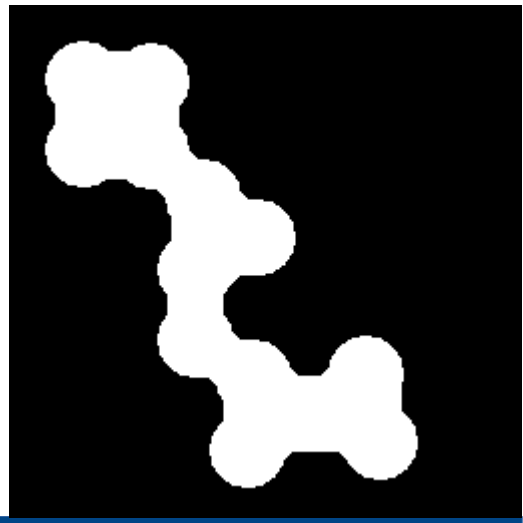


Closing

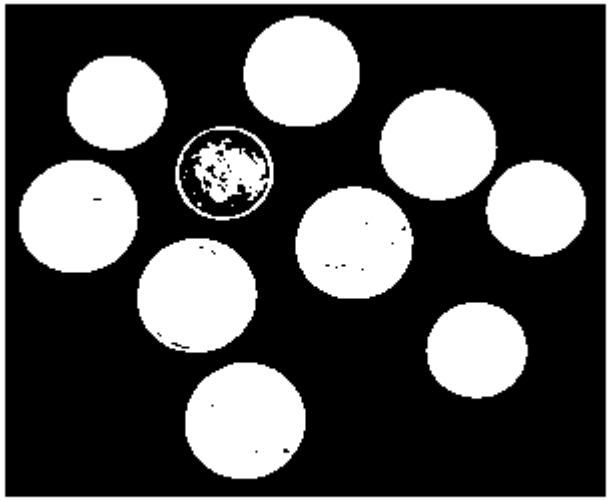
- The closing operator is to preserve background regions that have a similar shape or contains completely the structuring element, while eliminating all other regions of background pixels.
- Closing is defined simply as a dilation followed by an erosion using the same structuring element for both operations.

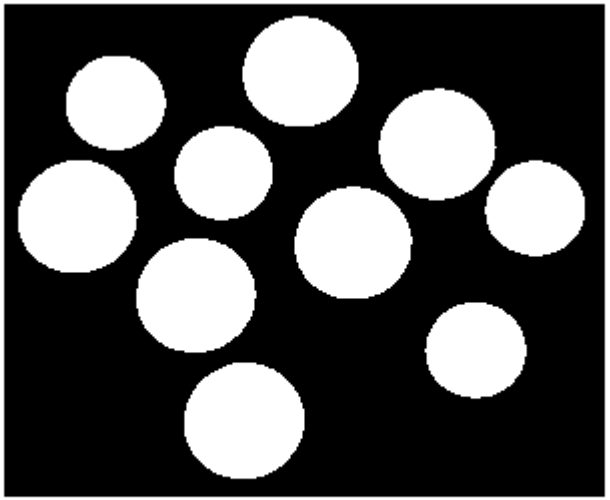






Region/Hole filling





- Region is defined as a closed contour
  - No discontinuities in the contour/edges
- This technique is used for noise or undesired artifacts removal within the target objects

imfill

Hit-and-miss transform

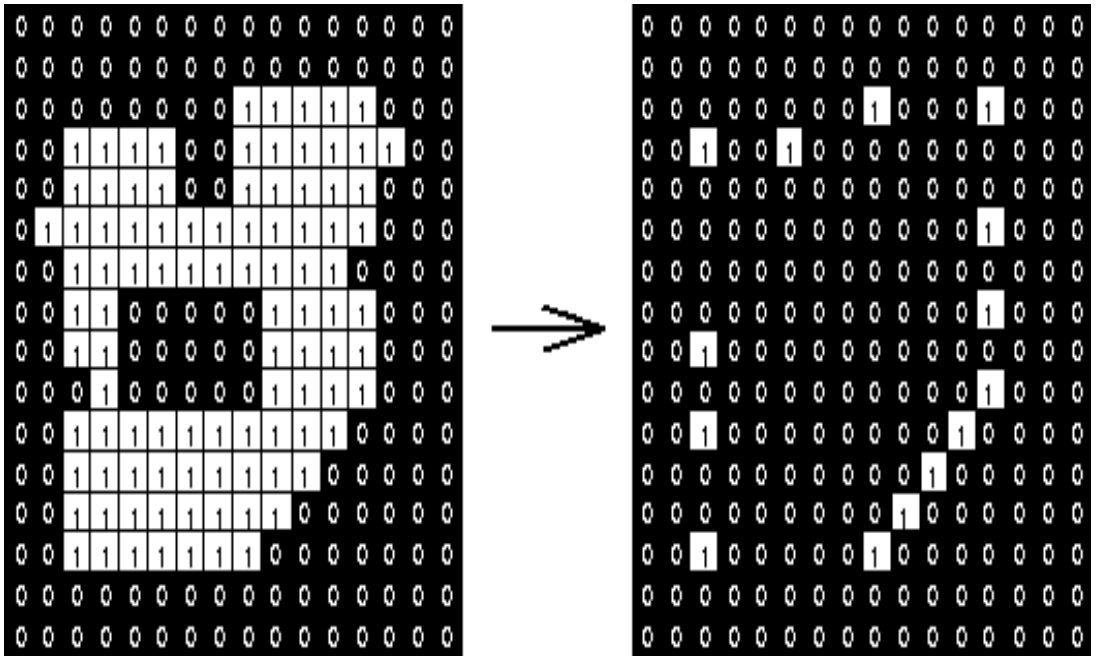
- The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image.
- **Corner detection**
  - Harris & Stephens
  - FAST method
  - Etc

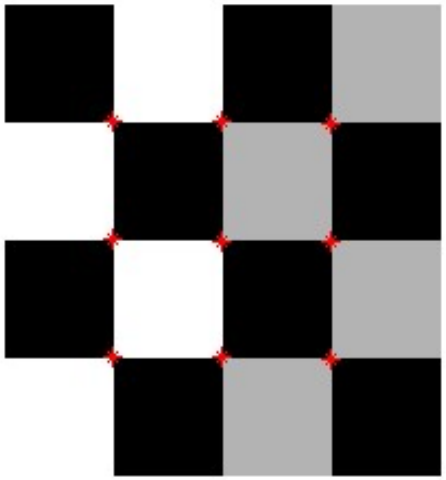
	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

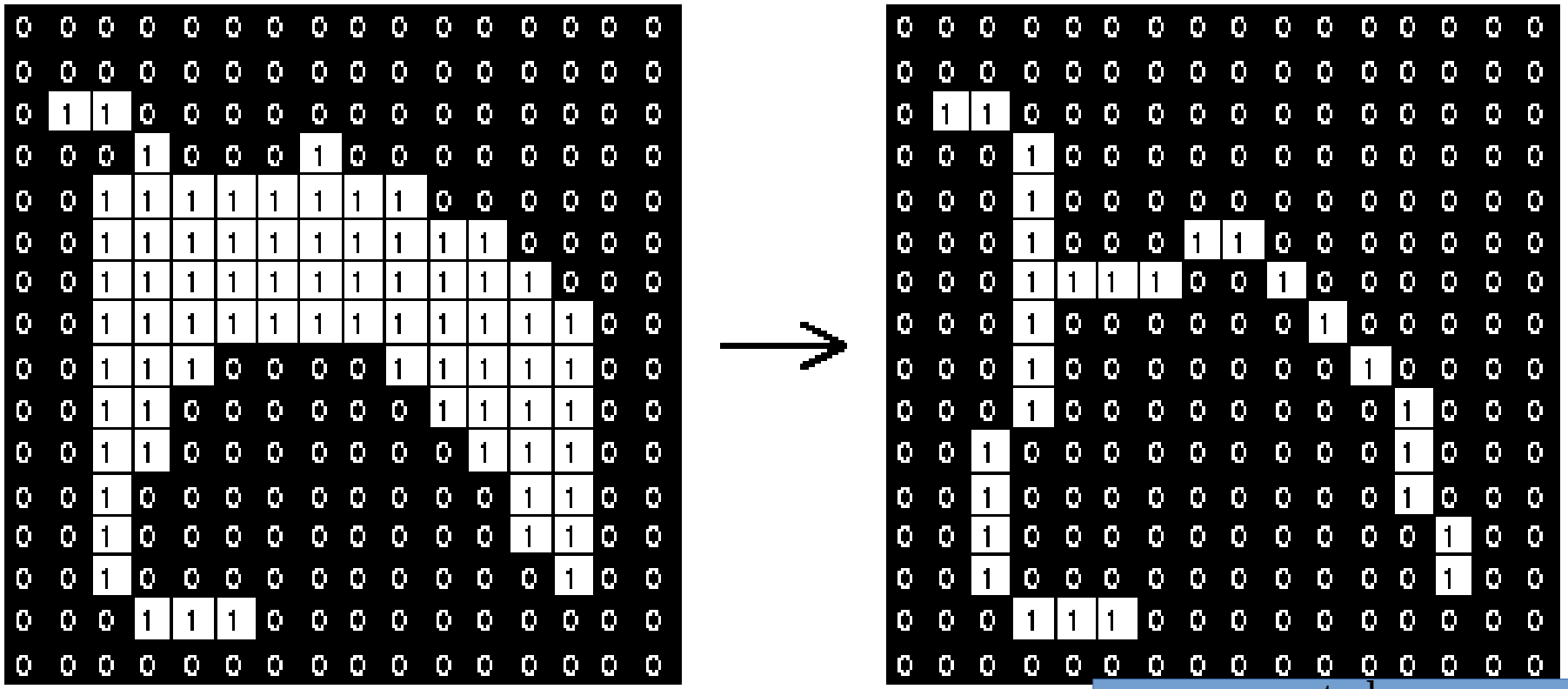
0	0	
0	1	1
	1	





Thinning/Skeletonization

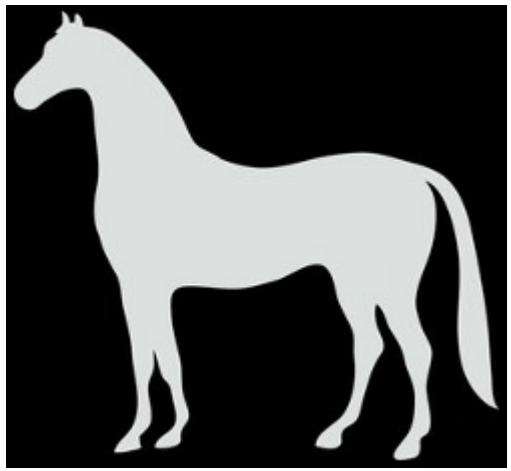
- **Thinning** is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening.
  - It is commonly used to reduce all edge lines to single pixel thickness.
  - It is particularly useful for **skeletonization**.
- Thinning is normally only applied to binary images, and produces another binary image as output.

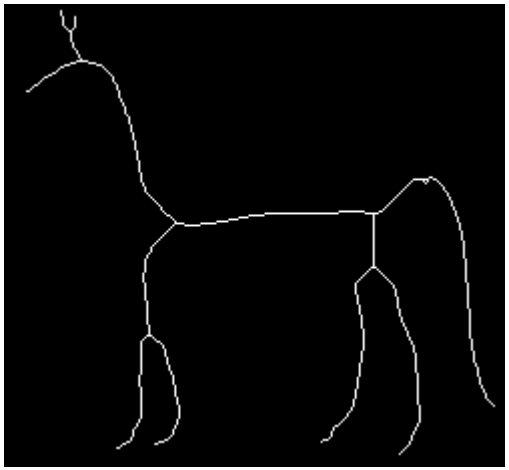


strel
bwmorph

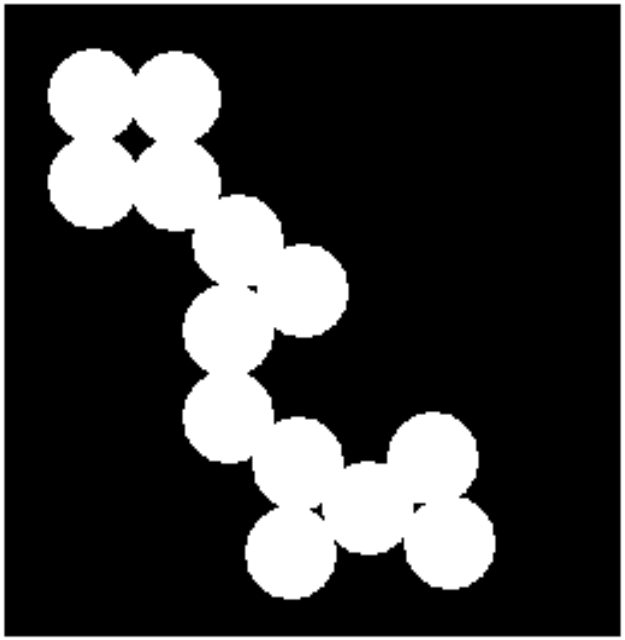
skeletonization

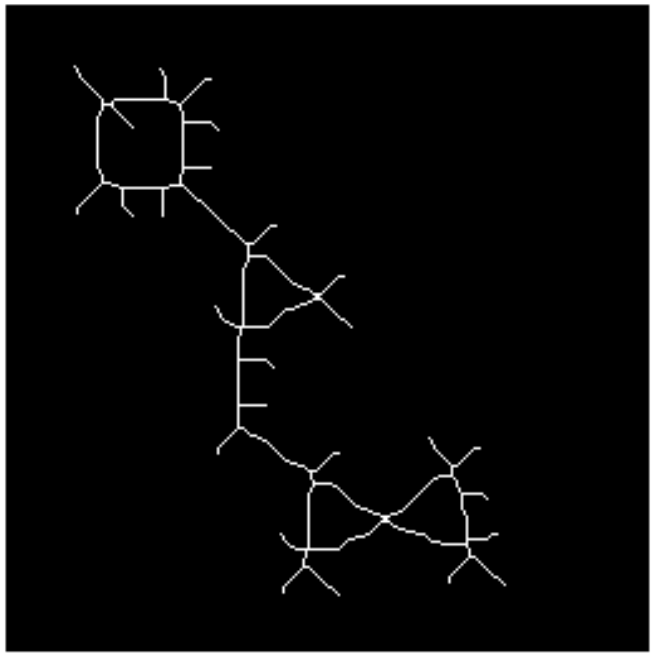
- The thinning operation is related to the hit-and-miss transform
  - **Thining**: It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary
  - **Skeletonization**: It removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image skeleton.

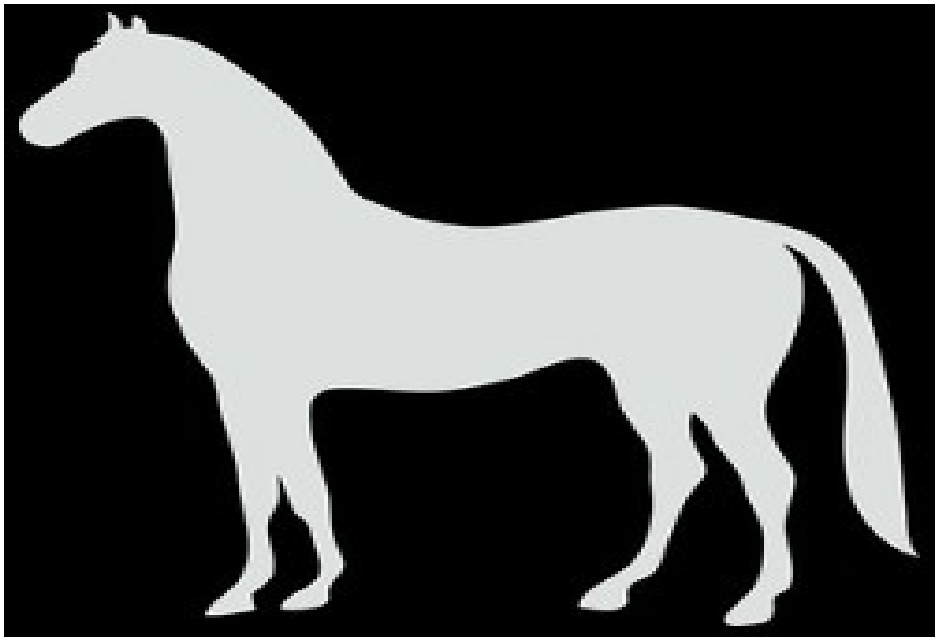


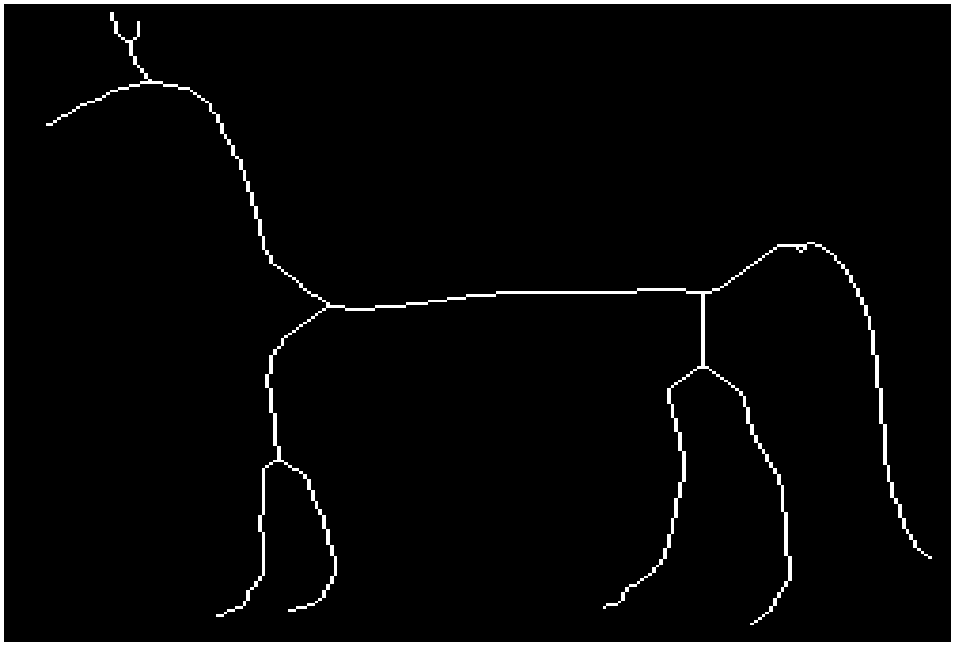


Skeletonization examples



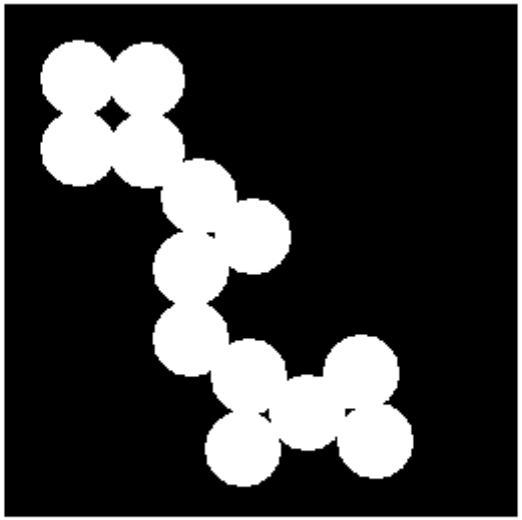


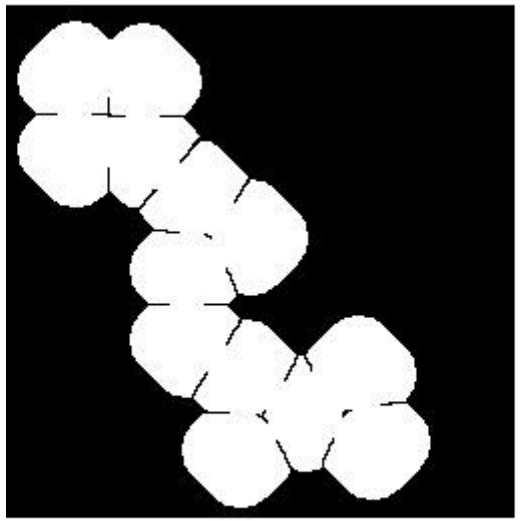




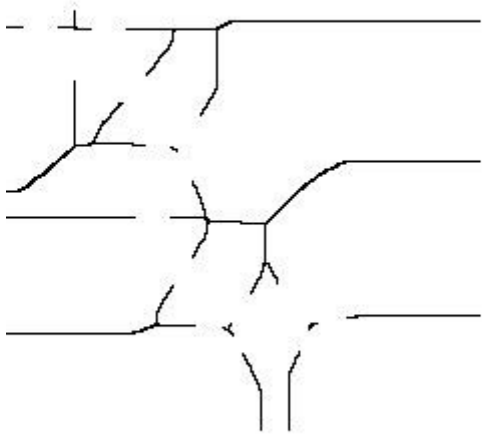
Thickening

- Thickening is a morphological operation that is used to **grow** selected regions of foreground pixels in binary images, somewhat like dilation or closing, related to the hit-and-miss transform
  - **Applications**: determining the approximate convex hull of a shape, and determining the skeleton by zone of influence.
- Thickening is normally only applied to binary images, and it produces another binary image as output.
- Note (Matlab): With n = Inf, thickens objects by adding pixels to the exterior of objects until doing so would result in previously unconnected objects being 8-connected.





Thicken=10



Thicken=Infinity

Morphological operations & Image enhancement

- A common technique for contrast enhancemen in **grayscale** images t is the combined use of the top-hat and bottom-hat transforms.
- **Top-hat transform:** It is defined as the difference between the original image and its opening.
  - You can use top-hat filtering to correct uneven illumination when the background is dark.
- **Bottom-hat transform:** It is defined as the difference between the closing of the original image and the original image.

TOP-HAT = Img - Opening(Img)

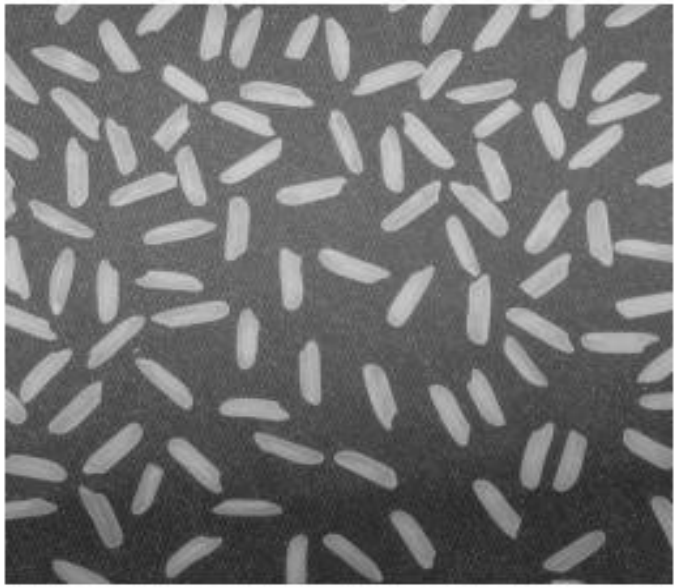
BOTTOM-HAT = Closing(Img) - Img

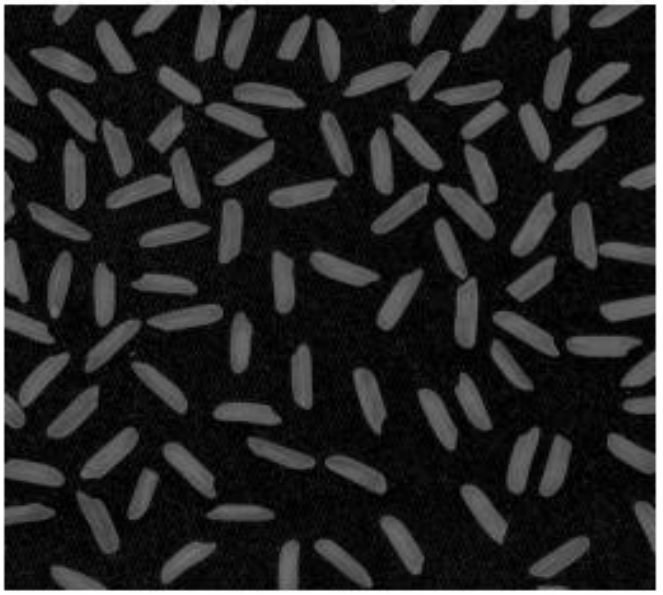
strel
Imtophat
imbothat

Examples

Bottom-hat

Top-hat





Commands used in the examples

Slide no.	Matlab	OpenCv
10,13-15	im2bw	Threshold()
15	graythresh	Threshold(),adaptivethreshold()
23-33	edge	Sobel(),Laplacian()
	34-36 imfilter	filter2D()
38	strel	-
40	imdilate	dilate()
42	imerosion	erode()
44-46	imopen	-
49-52	imclose	-
47	imfill	-
53-54	imbothat	-
53-54	imtophat	-
49-51	bwmorph	-

Summary of learned concepts

- Image binarization
- Image segmentation
  - Thresholding
    - Global
    - Otsu's method
    - Adaptive
    - Histogram-based
  - Edge Detection
    - Roberts Cross
    - Sobel
    - Prewitt
    - Canny
    - Laplacian/LoG
    - Zero Crossing
- Morphological operators
  - Dilation
  - Erosion
  - Closing
  - Opening
  - Hole filling
  - Thining
    - Skeletonization
  - Thickening
  - Top-hat
  - Bottom-hat

Practical Exercises

- **Exercise 1.** Select 2 grayscale images, one with a landscape and a person or animal on it, and a portrait (photo of a person/animal) in first plane.
  - Analyse the histogram of the images, and select a threshold you might think is the good one to separate the person/animal from the background
  - Apply Otsu's method, which is the threshold computed by Otsu's? Were you close to that number?
  - Apply some adaptive thresholding technique. Is it better the result?
  - Show all the images in a single figure to compare them
- **Exercise 2.** Edge detection
  - With the same images of the previous exercise, apply Roberts, Sobel and prewitt and visualize all of them in a single figure. Which is your impression? Which one is the best?
  - Now, apply Canny and compare to the others. Looks better, doesn't it? Or maybe, not?
  - Apply the LoG and the zero crossing detector and compare with Canny results

Practical Exercises

- **Exercise 3.** Morphological operations
  - Load the vessel files, you must apply morphological operations to try to be similar to the original one.
    - Join disconnected vessels
    - Remove noise (noisy regions outside or inside the vessels)
  - Apply thinning to the vessels, (different numbers) and then compute a skeleton
- **Exercise 4**. Closing and opening
  - Apply opening or closing to the previous images to try to get the same result as before
  - You can combine them with the usage of erosion and dilation

What did you learn today

- Segmentation of an image is needed to separate the target objects from the background
- It is not a trivial task
- The selection of a suitable threshold is highly dependent on the type of images
- Once the image is thresholded, the edges can be transformed through morphological operations
  - To remove artifacts
  - Noise
  - Undesirable noise

Octave tips