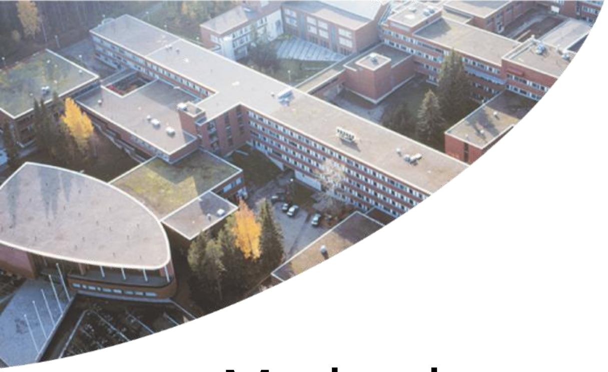
Set 7: Image Segmentation

Lecturer Arto Kaarna

Lappeenranta University of Technology (LUT)
School of Engineering Science (LENS)
Machine Vision and Pattern Recognition (MVPR)

Arto.Kaarna@lut.fi
http://www.lut.fi/web/en/school-of-engineering-science/research/machine-vision-and-pattern-recognition



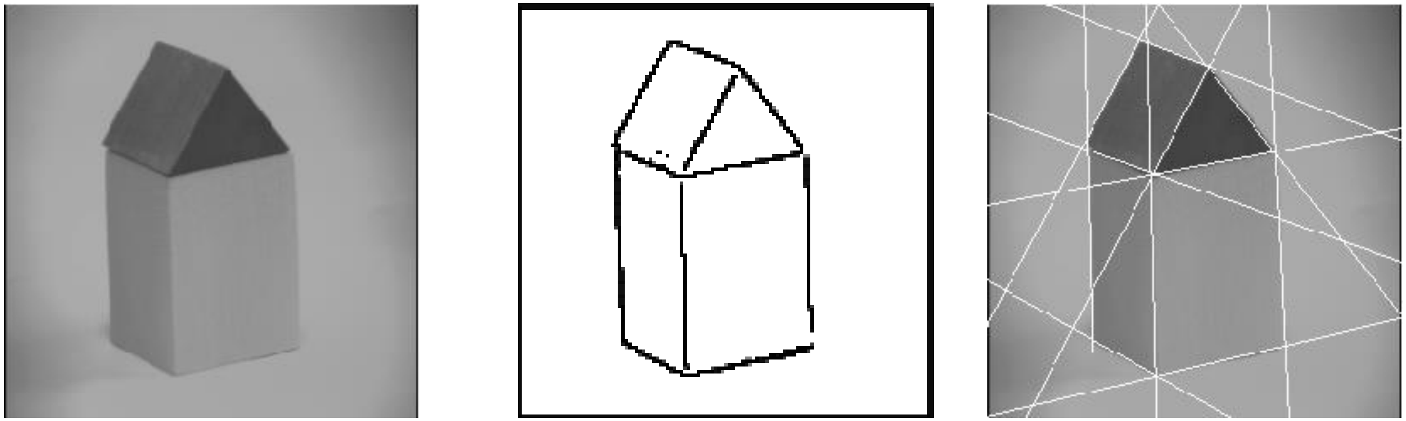
Contents

- Motivation
- Detection of discontinuities
- Edge linking and boundary detection
  - Local processing
  - Global processing: Hough Transform
- Thresholding
- Region-oriented segmentation
  - Region growing
  - Region splitting and merging

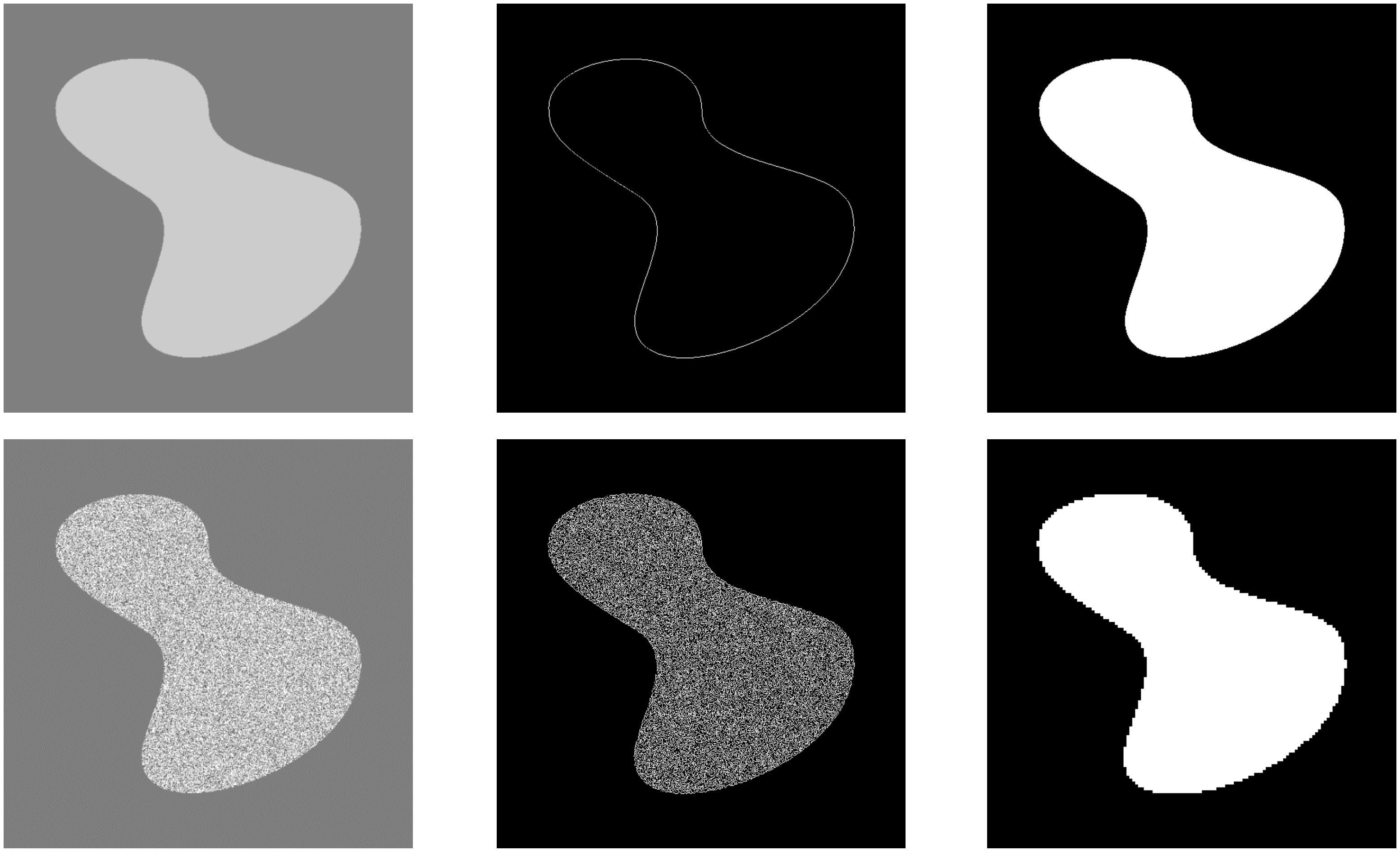
- Not included: Watersheds, Motion-based segmentation

Motivation

- To extract important details in an image for building a feature vector = feature extraction
- Feature selection => feature extraction => feature vector
- Challenging question: What details?

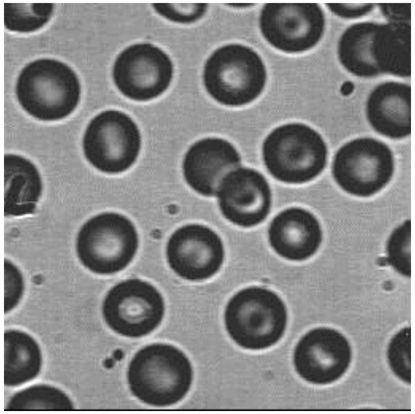


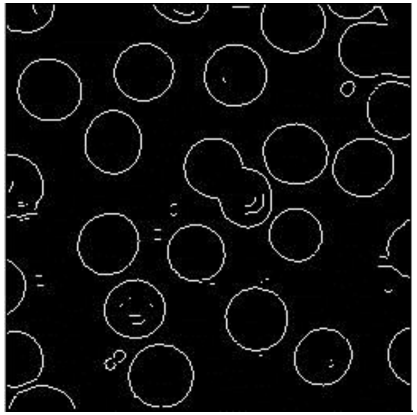
Example on Segmentation



Detection of Discontinuities

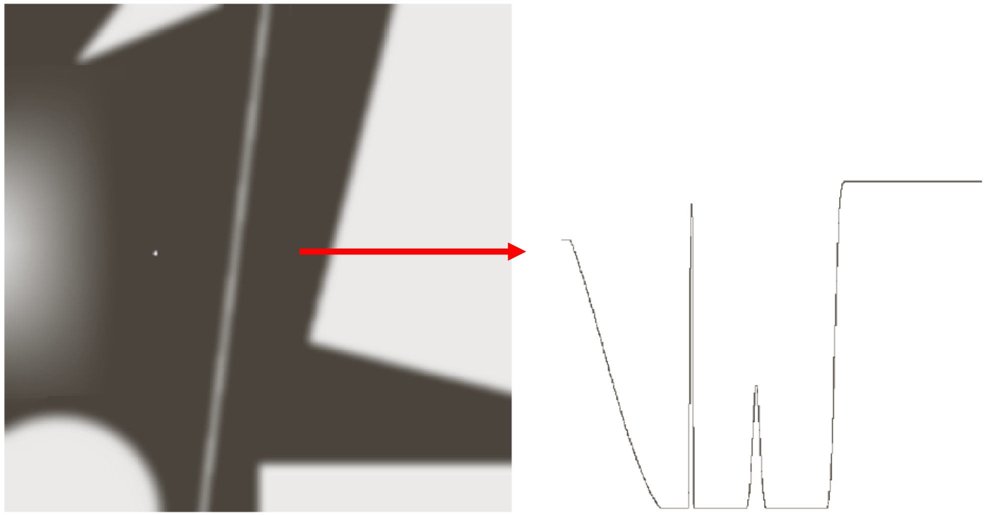
- Point detection
- Line detection
- Edge detection
  - Basic formulation
  - Gradient operators
  - Laplace operators
- Corner detection
- Combined detection

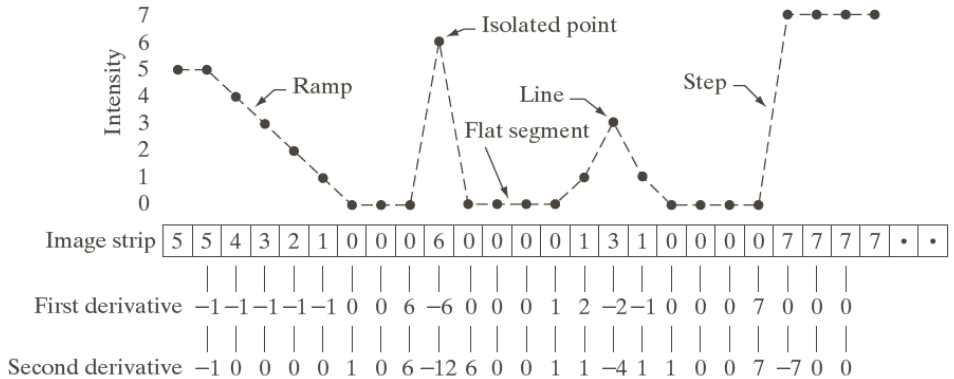




Detection of Discontinuities

- Types for discontinuities
- Pixel values and derivative information of them





Detection of Isolated Points

- Point detection based on the second derivative
- The Laplacian is

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

- and then this results in

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Detection of Isolated Points

- Detection of isolated points:

  a) Define the mask, e.g. a Laplacian or a modified

$$w_L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad w_m = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

  b) Threshold by using

$$g(x,y) = \begin{cases} 1, \text{if } |R(x,y)| \geq T \\ 0, \text{otherwise} \end{cases}$$ where

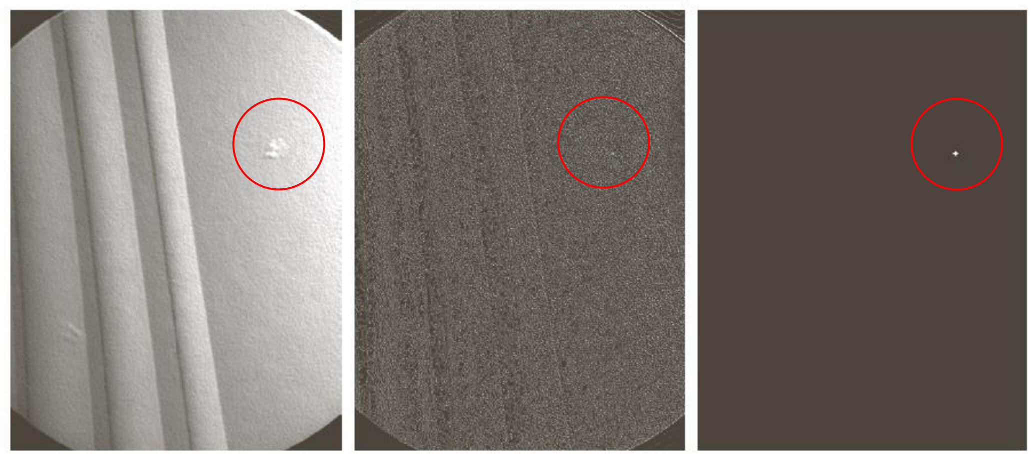
$R = \sum w_i z_i$, w_i is the mask, and

z_i is the intensity of the pixel value at location i

Detection of Isolated Points

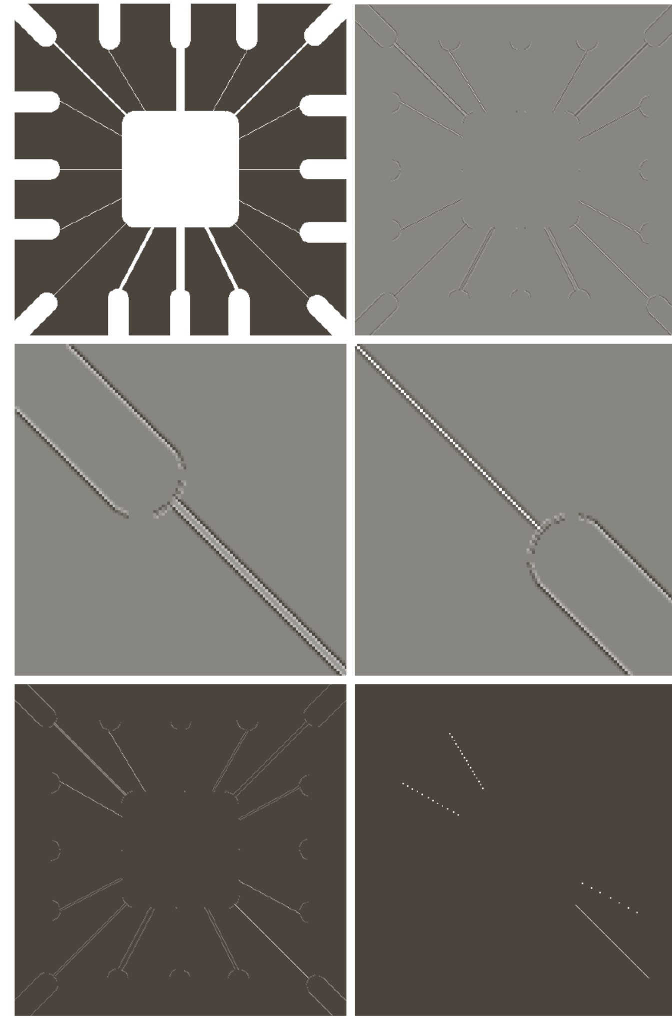
- Left: X-ray image,
- Center: modified,Laplacian mask,
- Right: thresholding, *T=90%* of the maximum intensity in the image.

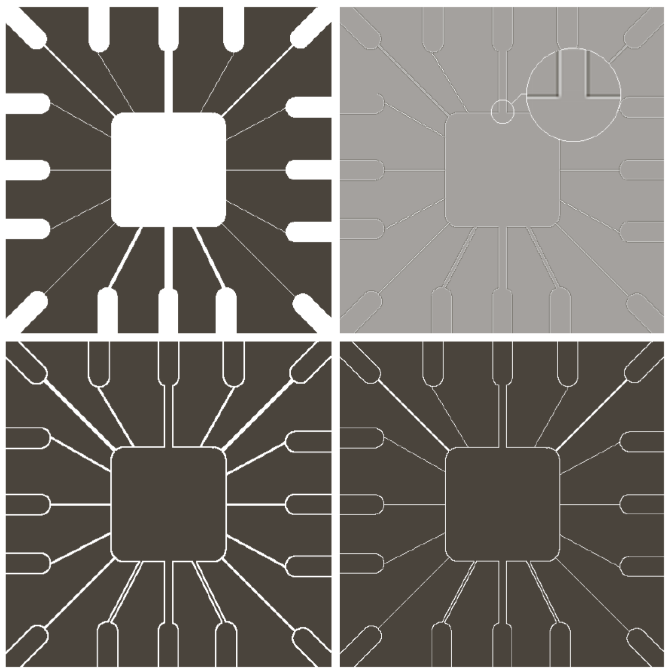
1	1	1
1	−8	1
1	1	1



Line Detection

- Laplacian detector is isotropic
- Many times, specific directions are of interest.





Line Detection

- Line detection with directed filters
  - lines with the thickness of one pixel

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

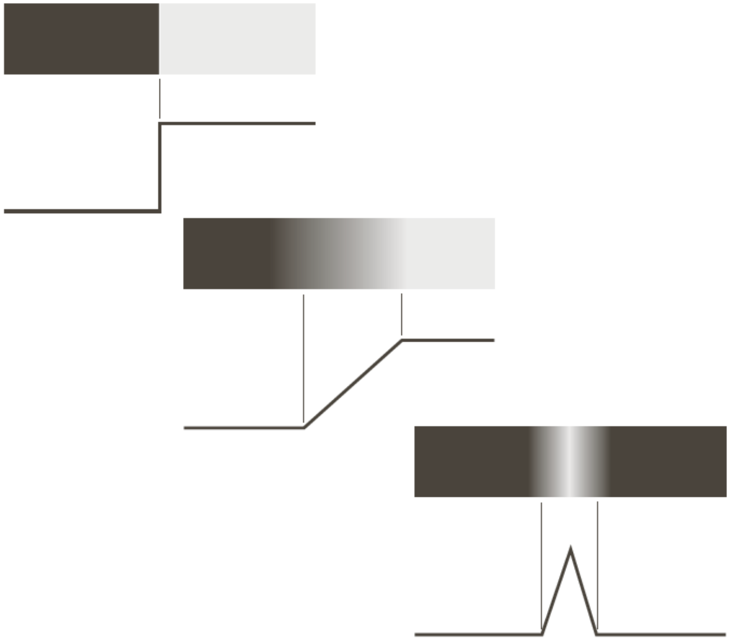
Vertical

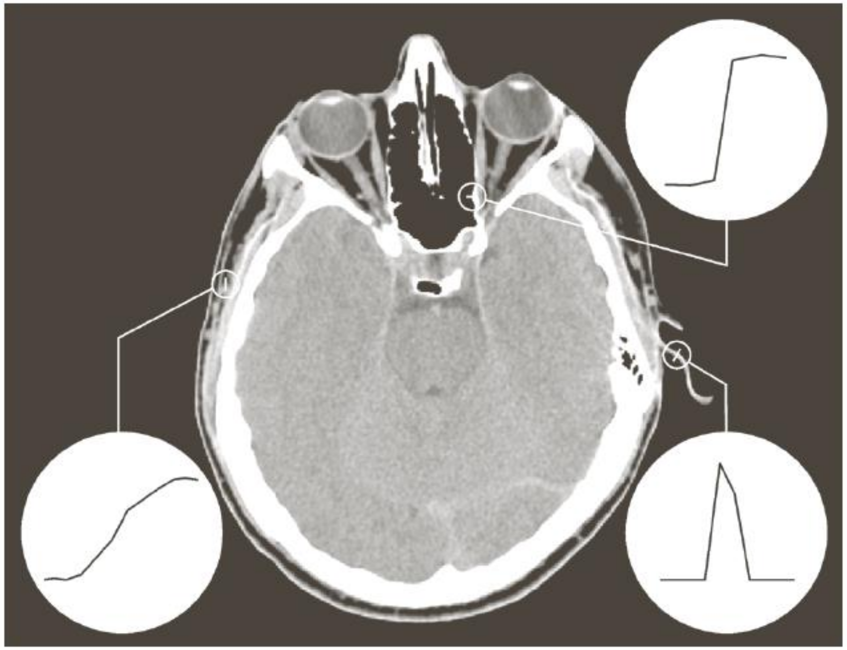
−1	−1	2
−1	2	−1
2	−1	−1

−45°

Edge Detection

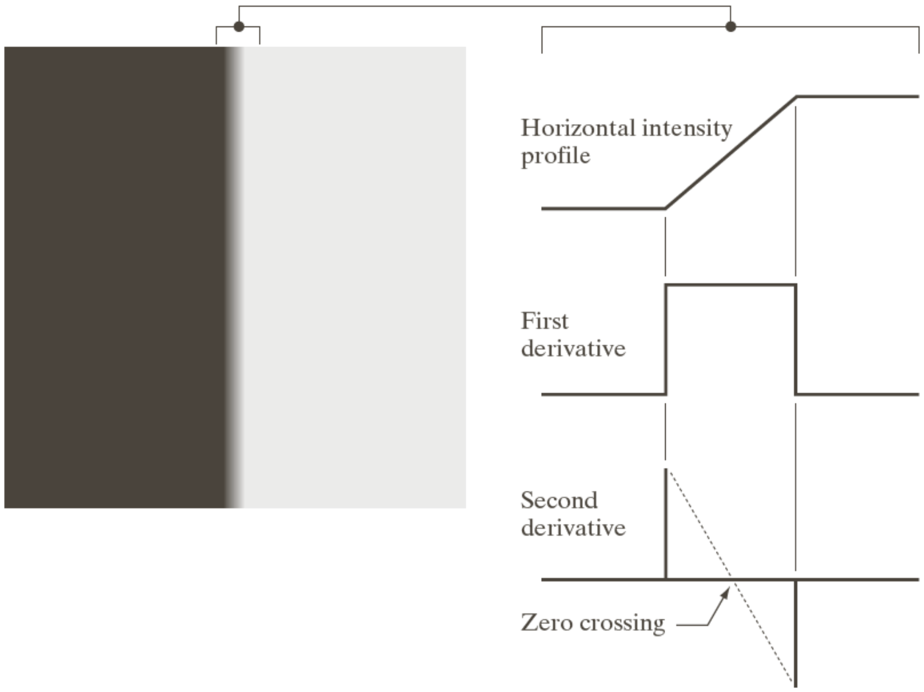
- Edge = The boundary between two regions with relatively distinct gray-level properties
- A step edge, a ramp edge, a roof edge





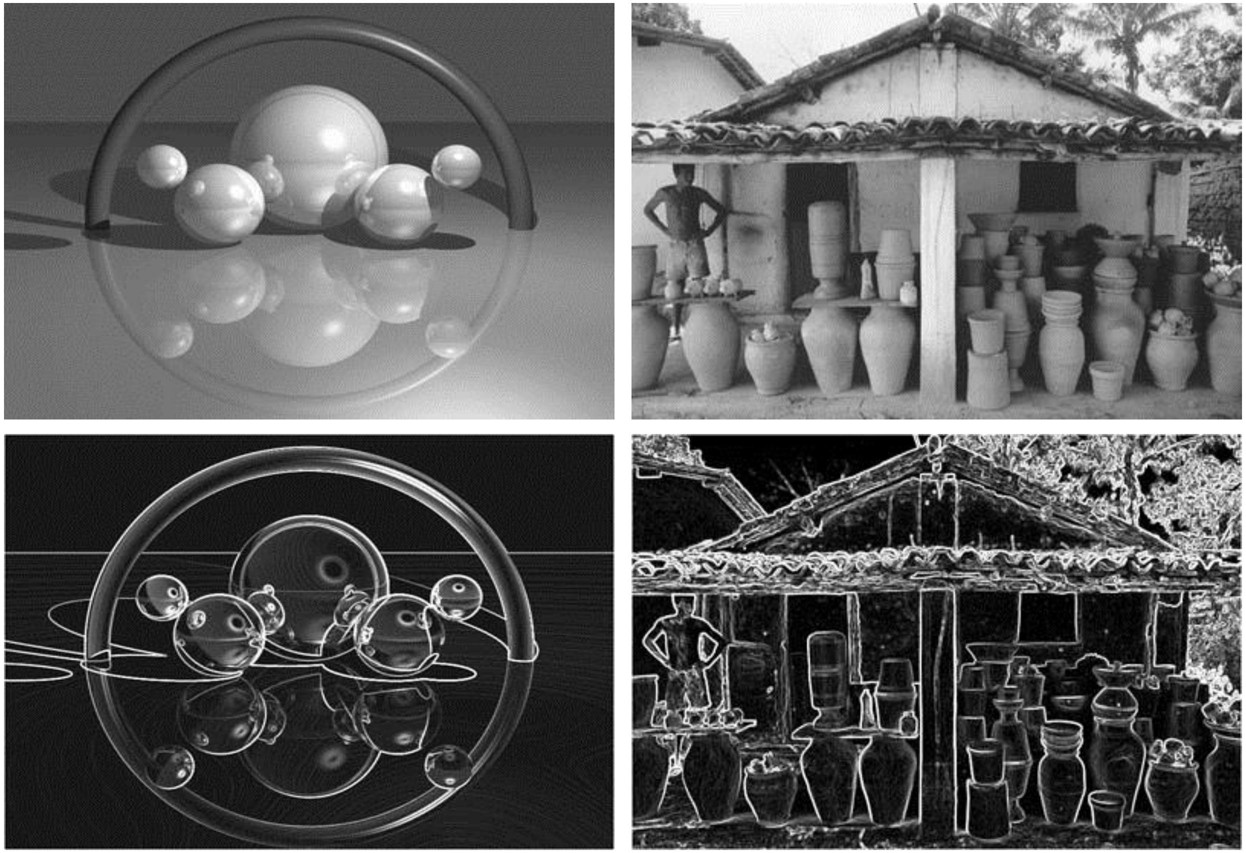
Edge Detection

- Features of an edge
- First-order derivates (Gradient operators)
- Second-order derivates (Laplace operators)
- Approaches with several techniques (e.g., Canny edge detector)



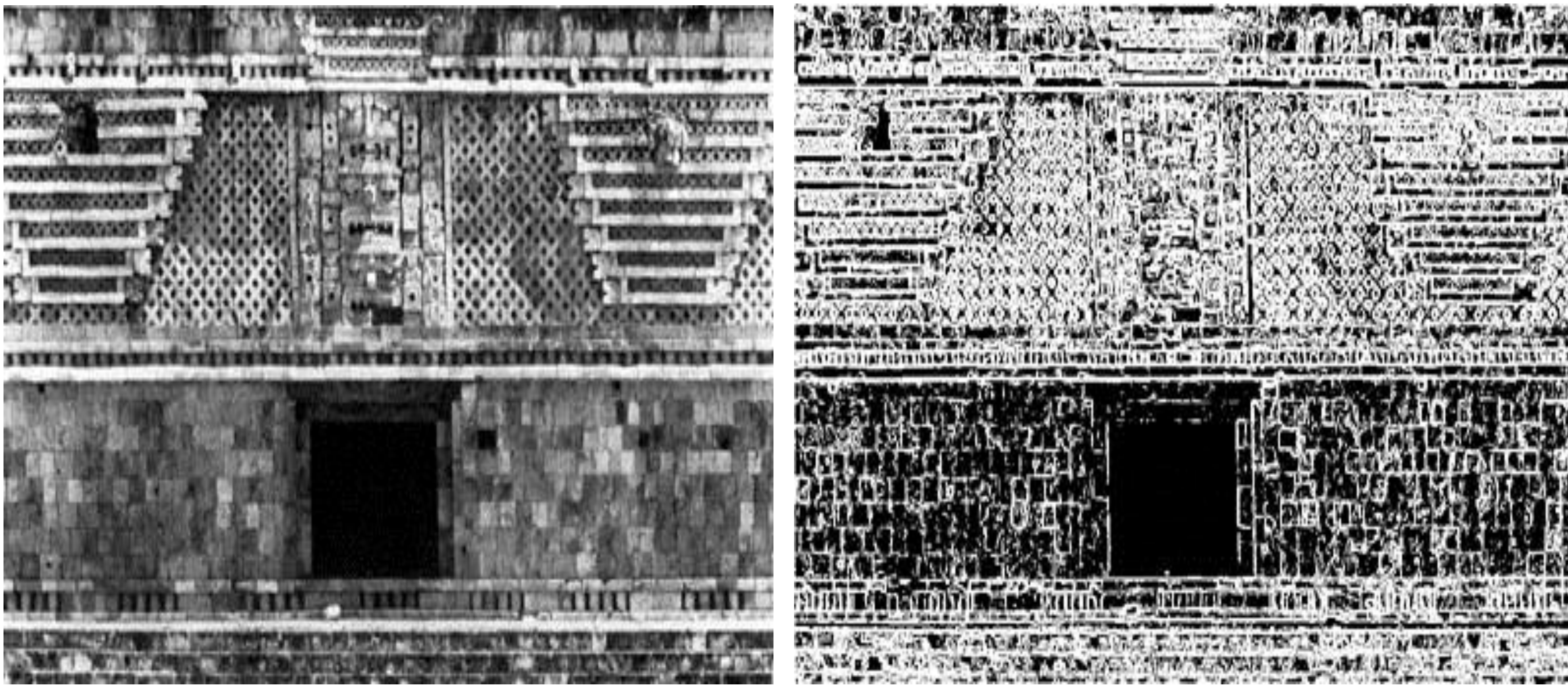
Edge Detection, examples

- Edges in practical images



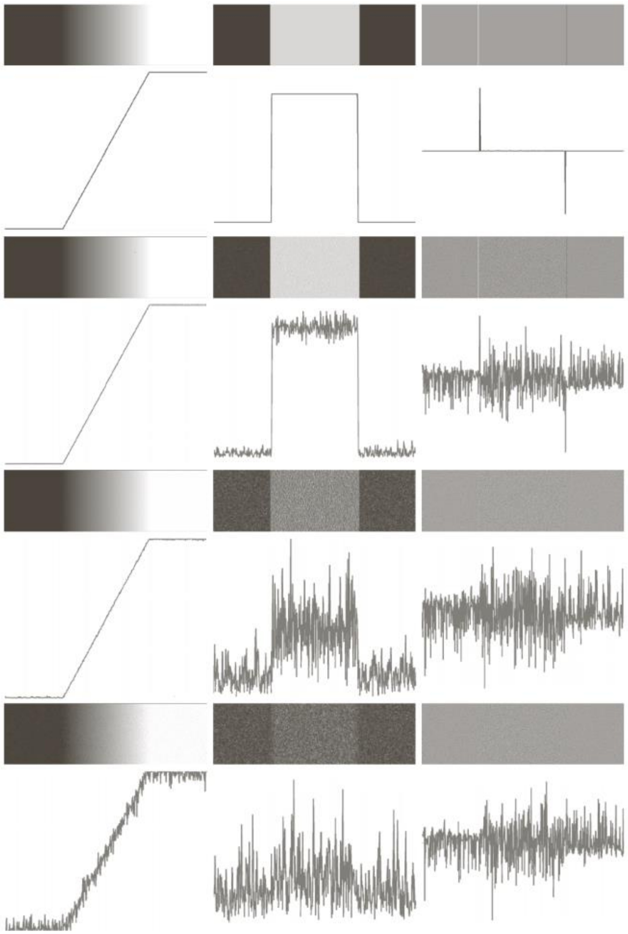
Edge Detection: various images

- Almost all pixels belong to edges



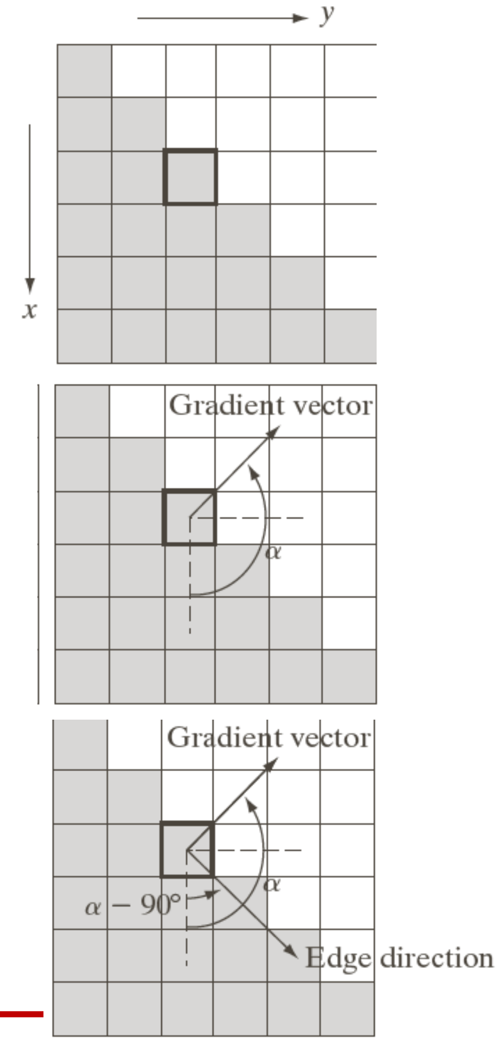
Edge Detection in General

- Basic steps in edge detection:
  1. Image smoothing for noise reduction (see the rightmost column).
  2. Detection of edge points (local operation to find candidates to become points on edges).
  3. Edge localization (finding the true members of the edges).



Edge Detection: Gradient operators

- Gradient of $f(x,y)$: $df = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$
- Magnitude: $mag(df) = \sqrt{G_x^2 + G_y^2}$
- Approximation: $mag(df) \approx |G_x| + |G_y|$
- Direction: $\alpha(x,y) = tan^{-1} \frac{G_y}{G_x}$



Edge Detection: Gradient operators

- In practice gradients are approximated by a (weighted) gray-value difference.
- Prewitt and Sobel operators also for diagonal edges

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0		0	−1
0	1		1	0

Roberts

0	1	1		−1	−1	0
−1	0	1		−1	0	1
−1	−1	0		0	1	1

Prewitt

0	1	2		−2	−1	0
−1	0	1		−1	0	1
−2	−1	0		0	1	2

Sobel

−1	−1	−1		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

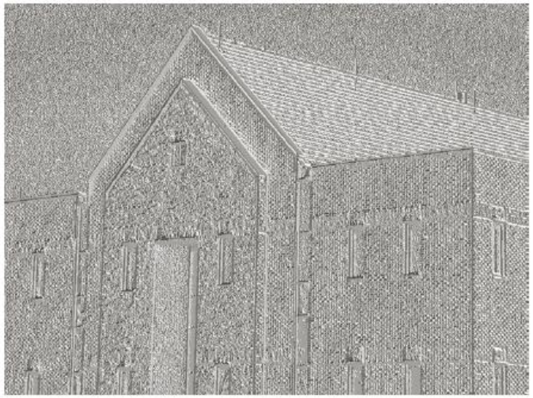
Prewitt

−1	−2	−1		−1	0	1
0	0	0		−2	0	2
1	2	1		−1	0	1

Sobel

Edge Detection, Sobel operator

- Original image, $|G_x|$, $|G_y|$ and $|G_x| + |G_y|$ with a Sobel operator;
- Gradient angle



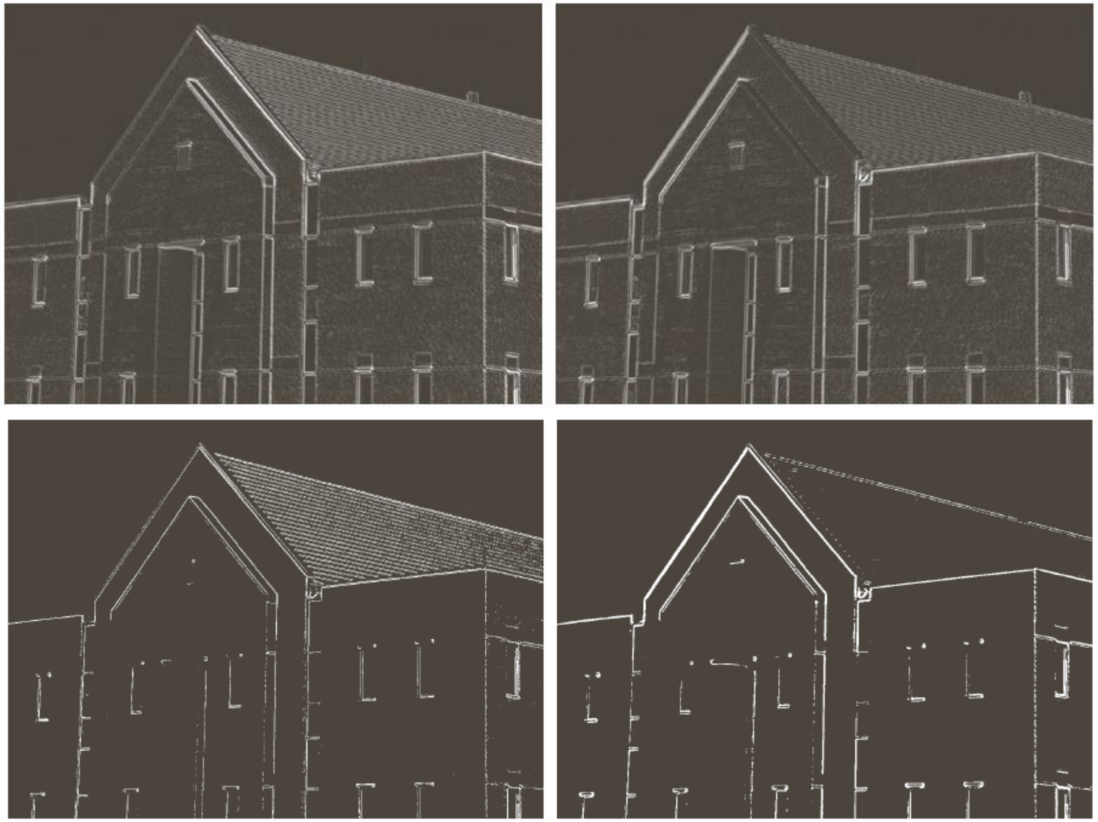
Edge Detection, Sobel operator

- Image smoothing with a 5x5 averaging filter before the edge detection.

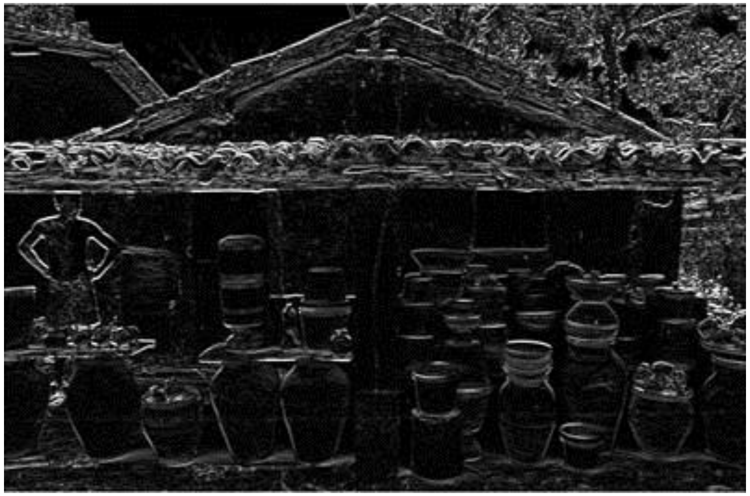
Edge Detection, Sobel operator

- Diagonal operator, -45, 45 degrees

- Thresholding of the gradient image, *T=33%* of the maximum intensity, images (-2p, lr; -1 p, lr)



Edge Detection: Prewitt operator



- 3x3 Prewitt operator
  - Horizontal mask (right)
  - Vertical mask (bottom, left)
  - Sum of the two masks (right)

Edge Detection, Marr-Hildreth

- Laplace operator is sensitive to noise
- Finding the location of edges using the zero-crossing property: convolve an image with the Laplacian of the 2D Gaussian function of the form

$$G(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

  where σ is the standard deviation.
- When $r^2 = x^2 + y^2$ then

$$\nabla^2 G = \frac{r^2 - \sigma^2}{\sigma^4} e^{\frac{-(r^2)}{2\sigma^2}}$$

- Zero-crossings are at $r = \pm\sqrt{2}\sigma$

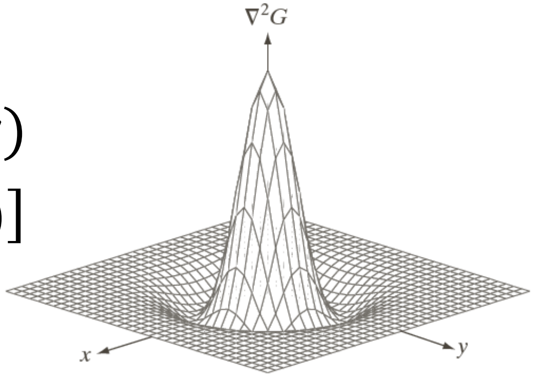
Edge Detection, Marr-Hildreth

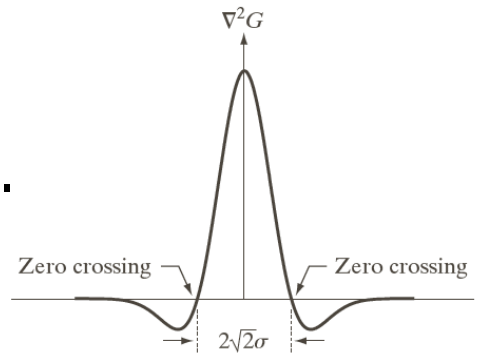
- Laplacian of Gaussian use derivative information and operators of difference sizes (various scales)

$$g(x,y) = [\nabla^2 G(x,y)] \star f(x,y)$$
$$g(x,y) = \nabla^2 [G(x,y) \star f(x,y)]$$

Algorithm:

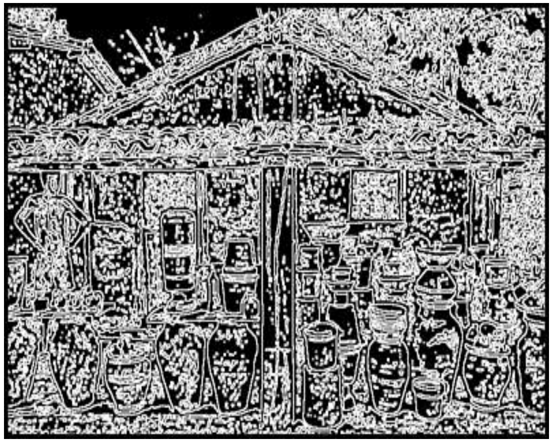
1. Filter the image with a lowpass Gaussian filter.
2. Compute the Laplacian of the image, e.g. 3x3 mask.
3. Find the zero crossings of the filtered image.





0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Edge Detection, Marr-Hildreth



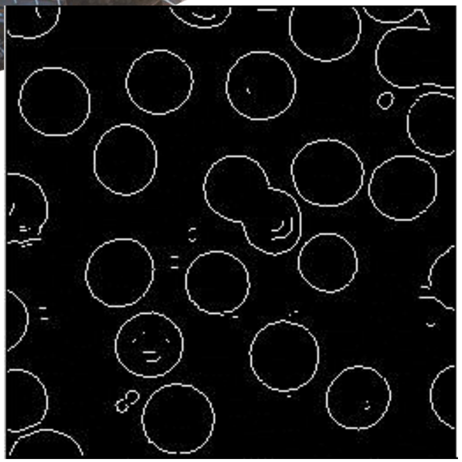
- Convolution with a filter $h(r)$ with

$\leftarrow \sigma = 1.5$

$\sigma = 5.0 \rightarrow$

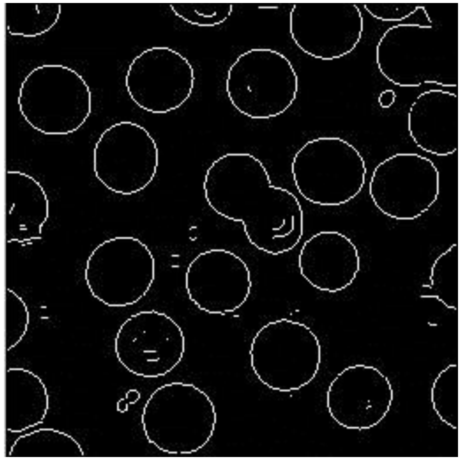
- Zero-crossings

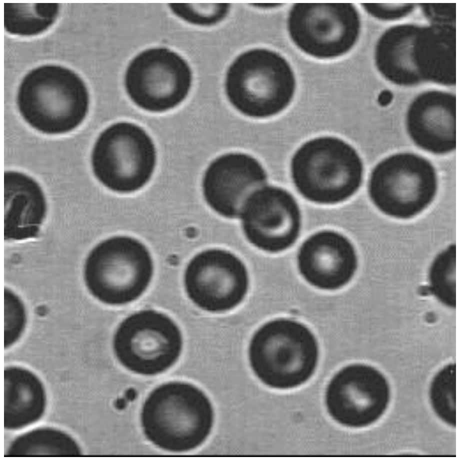
Edge detection: comparisons

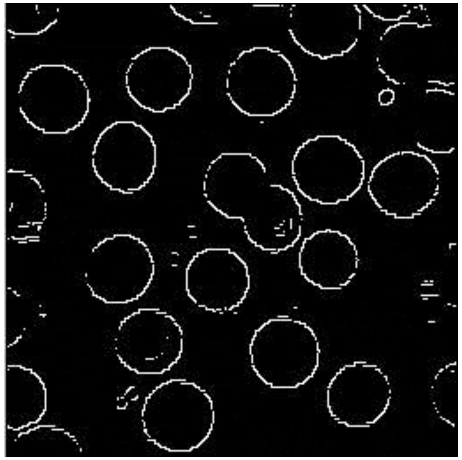


← Sobel

Prewitt →

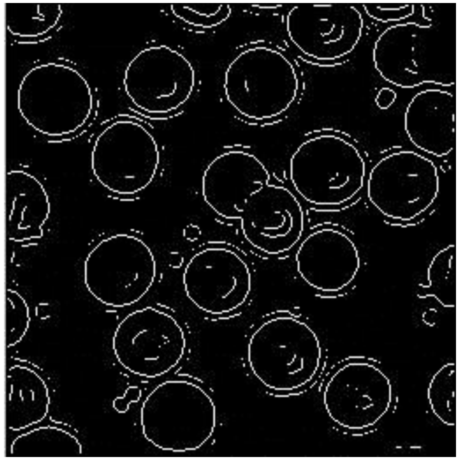






← Roberts

LoG →



Canny edge detector

- Canny edge detector
  - Requirements
    - Low error rate; All edge points should be well localized; Single edge point response.
  - Smoothing the image and eliminating any noise
    - Gaussian mask → a more blurred image
  - Finding the edge strength
    - Gradient operators
  - Defining the edge direction
    - Gradient operators
  - Thinning by nonmaxima suppression
    - Suppress the pixels by a mask → thinner lines
  - Hysteresis thresholding
    - Double thresholding and connectivity analysis to detect and link edges → less streaking

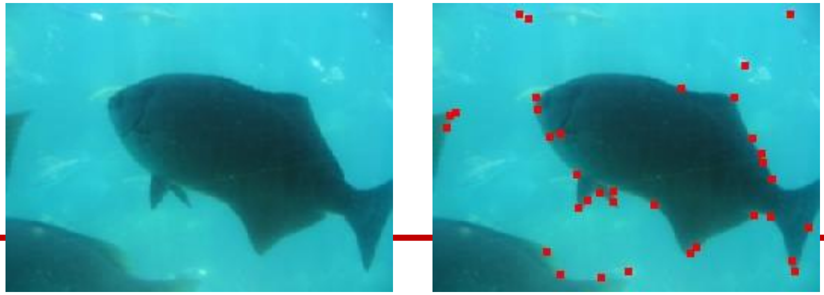
Canny edge detector

- Original image
- Thresholded gradient image

- Edges by Marr-Hildreth
- Edges from Canny edge detector

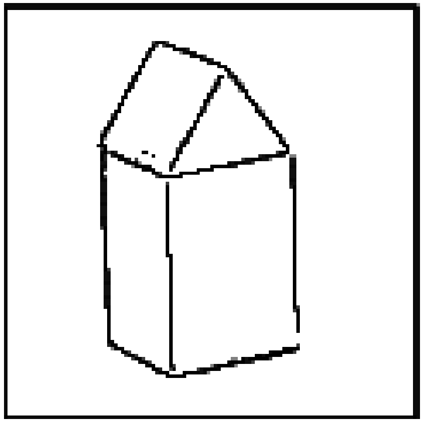
Harris Corner Detector

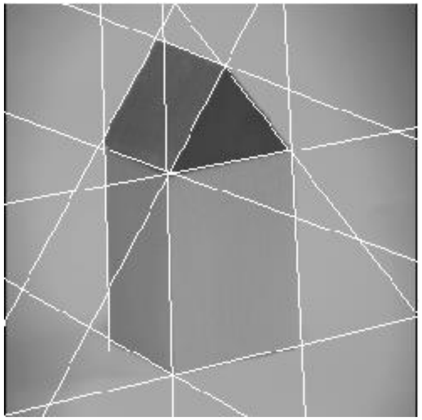
- Corners are usually very suitable as interest points
- Corners are found as the intersection of two edges or as the region of two different strong edge orientations
- Based on the local auto-correlation function of a signal $C(c, y)$ which measures the local changes of the signal with patches shifted by a small amount in different directions
- Corner response $H(x)$ defined as a function of $C(x, y)$
- Interest points can be found as local maxima in the corner response $H(x)$



Edge linking and boundary detection

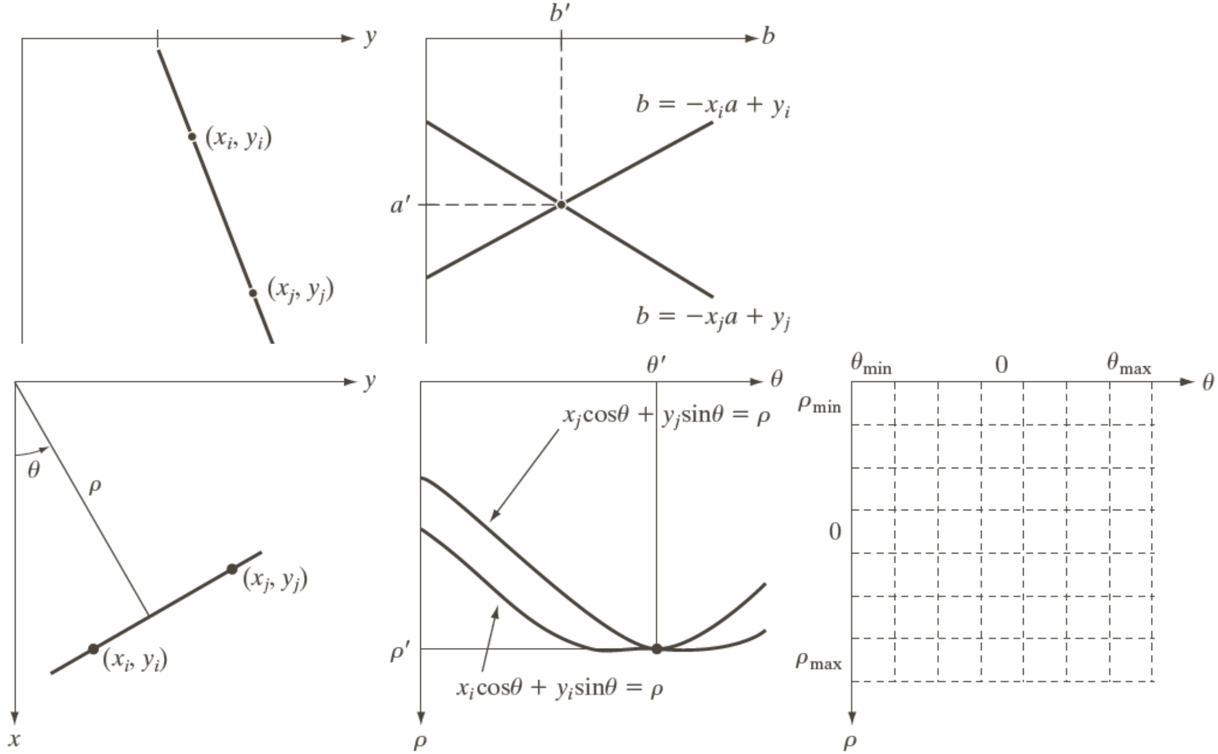
- From intensity discontinuities to more general segmentation
  - For example, from edge pixels to line segments
- Local processing
  - Analysis of a small neighborhood: strength and direction of the gradient of edge pixels
- Global processing:
  - Analysis of the whole image: global relationships between pixels
  - Hough Transform





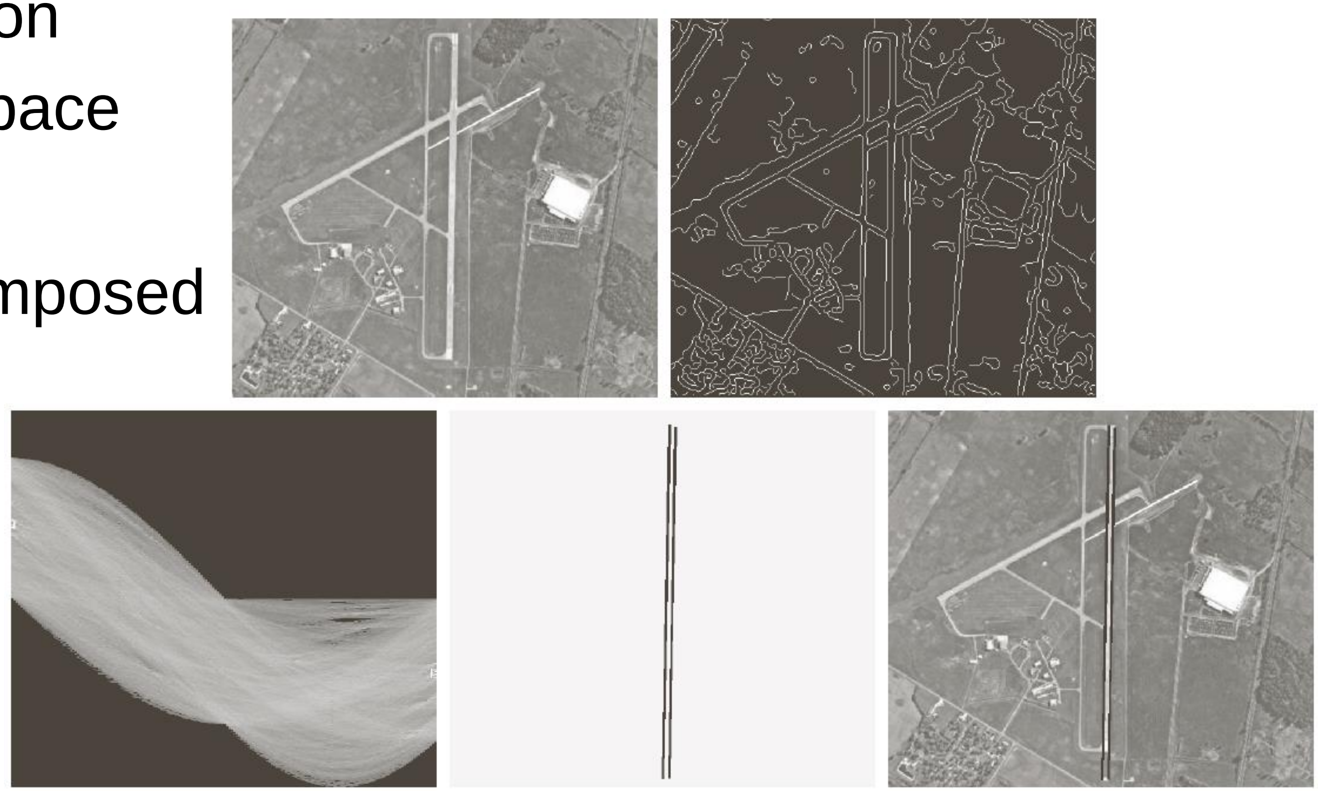
Hough Transform

- The line in xy-space is transformed to a point in ab-space or $r\theta$-space; accumulator collects information on parameters



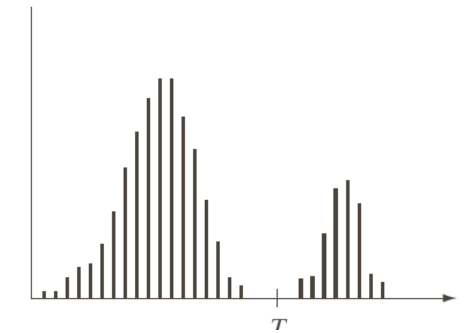
Hough Transform

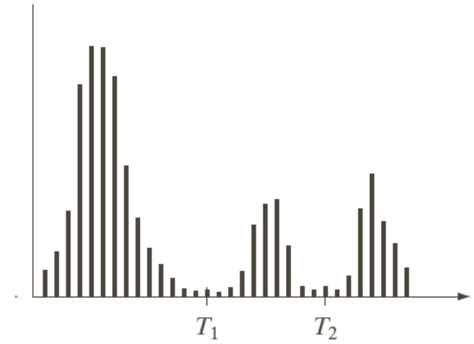
- Original image
- Edge detection
- Parameter space
- Lines found
- Lines superimposed in the image



Thresholding: Foundation

- For example, to separate an object (objects) and background
- One threshold:
  - Object point: $f(x,y) > T$
  - Background: $f(x,y) < T$
- Two thresholds:
  - Object class 1: $T_1 < f(x,y) < T_2$
  - Object class 2: $f(x,y) \geq T_2$
  - Background: $f(x,y) \leq T_1$





Thresholding: Foundation

- Operation

  $$T = T[x, y, p(x,y), f(x,y)]$$

  where

  $f(x,y)$ is the gray-level at point (x,y)

  $p(x,y)$ denotes some local property at this point

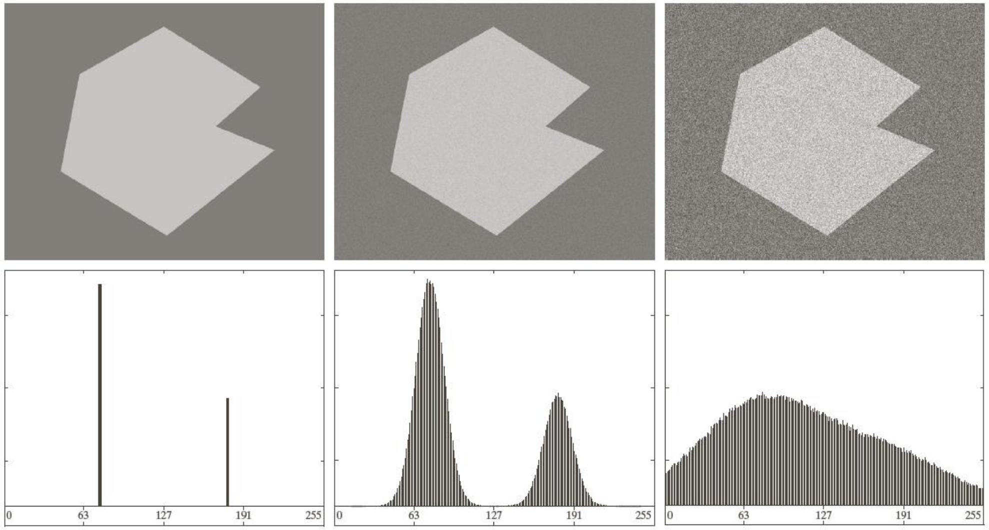
  (e.g., the average gray-level of a neighborhood centered on (x,y))

- A thresholded image is $g(x,y)$ is defined as

$$g(x,y) = \begin{cases} 1, \text{if } f(x,y) > T \\ 0, \text{if } f(x,y) \leq T \end{cases}$$

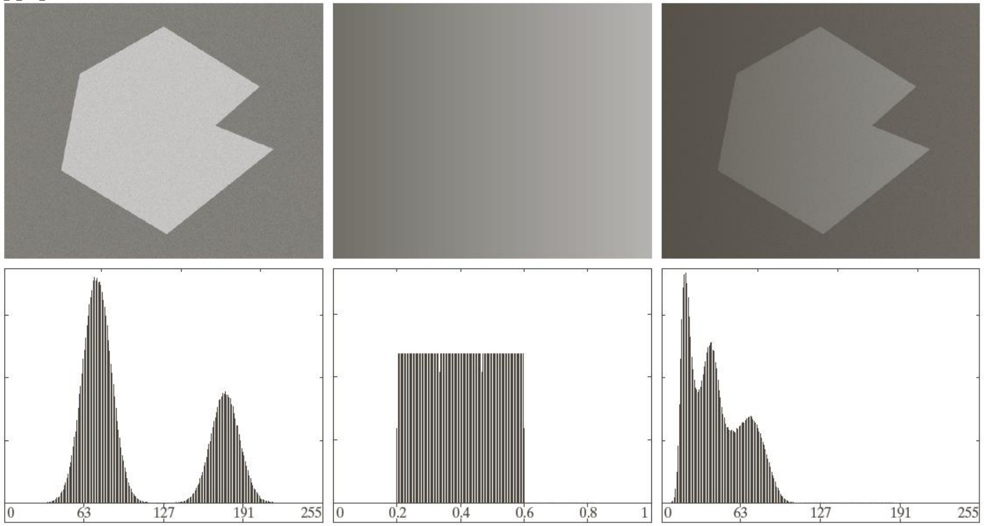
Thresholding: Role of Noise

- The content of the image changes due to noise
  - → The shape of the histogram changes due to noise
  - → The thresholds become more difficult to define
- How to solve problems?
  - Preprocessing, like image smoothing
  - Using edges



Thresholding: Role of illumination

- An image $f(x,y)$ is a product of a reflectance component $r(x,y)$ and an illumination component $i(x,y)$
- Original image $\rightarrow$ suitable histogram to thresholding
- Original image + illumination changes $\rightarrow$ unsuitable histogram for thresholding

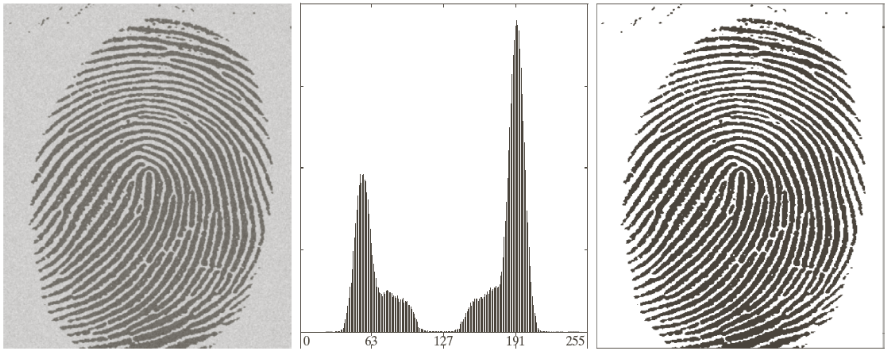


Thresholding: Role of illumination

- Solutions:
  - Correct the shading pattern with an inverse operation
  - Use morphologigal operators for correctiing the full image
  - Use variable thresholding in various parts of the image
    - A simple approach: partition into subimages and threshold each separately
    - HOW to partition is critical !

Thresholding: Global and optimal

- How:
  - Simple thresholding: T may be selected manually if the objects and the background are sufficiently distinct



Algorithm:

1. Set initial threshold T.
2. Segment the image using T.
3. Compute averages m_1 and m_2 for the two groups
4. Find new threshold as $T = 0.5 * (m_1 + m_2)$
5. Continue from step 2 until no changes in T.

Thresholding: Global and optimal

- How:
  - Optimal global thresholding: optimally by probability densities (Gaussian)

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{-(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean value and σ is the standard deviation.

  - Otsu’s method maximizes the between-class variance
    - Classes have separate intensity ranges
    - Working on the 1D histogram only

Thresholding: Global and optimal

- Mixture probability density of two classes:

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

where P_1 and P_2 are the *a priori* probabilities for the two classes, $P_1 + P_2 = 1$

- The threshold value T where the error is in minimum comes from the relation

$$P_1 p_1(T) = P_2 p_2(T)$$

- Two special cases:

$$\sigma = \sigma_1 = \sigma_2 : T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} ln \frac{P_2}{P_1}$$

$$P = P_1 = P_2 : T = \frac{\mu_1 + \mu_2}{2}$$

Thresholding: Global and optimal

- Generally $P_1 p_1(T) = P_2 p_2(T)$ can be formulated as

$$\mathrm{A}T^2 + BT + C = 0$$

where

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 ln(\sigma_2 P_1/\sigma_1 P_2)$$

- Now T is solved as

$$T = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

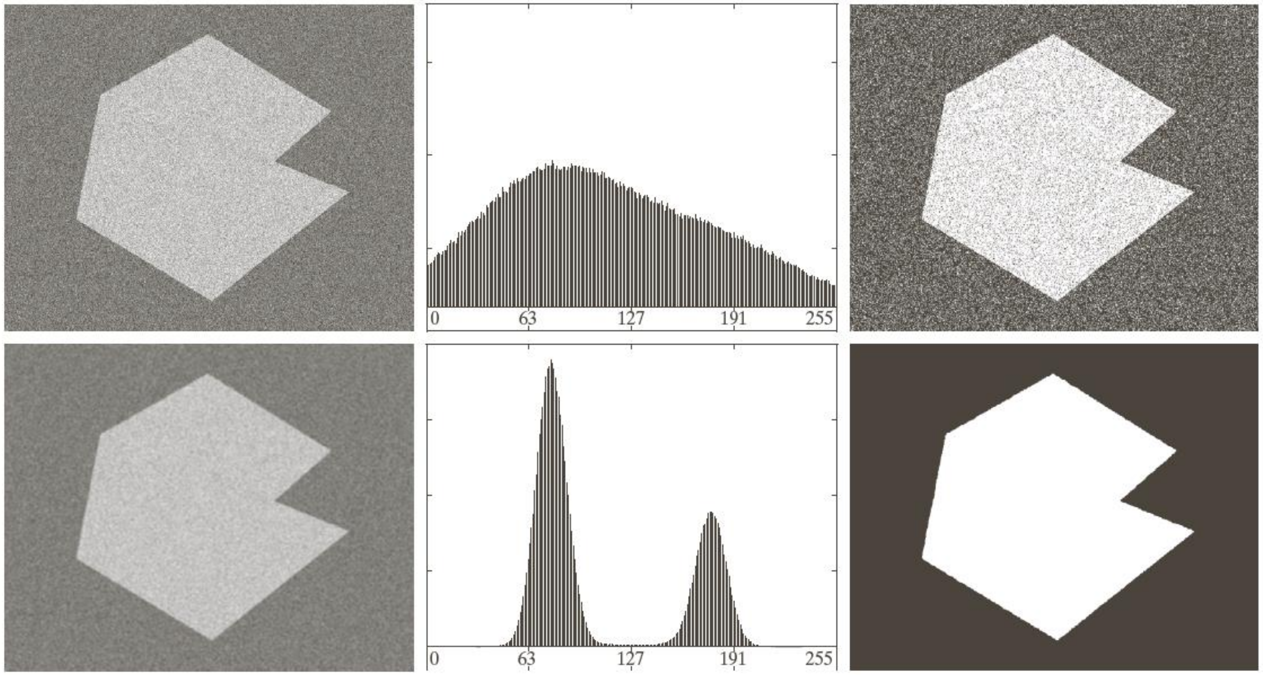
Thresholding by Otsu

- Clustering based image thresholding
  - gray-level → binary
- Computes the threshold that minimizes intra-class variance (maximizes inter-class (between-class) variance)
  - Exhaustive search for the optimal threshold: Find the normalized histogram; compute cumulative sums P_k and means m_k; compute global mean m_G; compute the between-class variance $\sigma_B^2(k)$; find the threshold k^* as the maximum for $\sigma_B^2(k)$; perform segmentation
- Multi-Otsu-method for multi-level thresholding

Thresholding by Otsu

- Noisy image
- Histogram
- Segmentation

- Smoothed image, 5x5
- Histogram
- Segmentation



Using Edges in Segmentation

- When the peaks in the histogram are tall, narrow, symmetric, separated by deep valleys, then the threshold can be found.
- What is we look at the pixels near the edges only?
  - The effect of the different sizes of the classes disappear → peaks of the histogram of equal size
  - Probability to belong to the object or to the background is equal → improves the symmetry of the histogram modes (highest peaks)
  - There is a tendency to deepen the valley between the histogram peaks

Using Edges in Segmentation

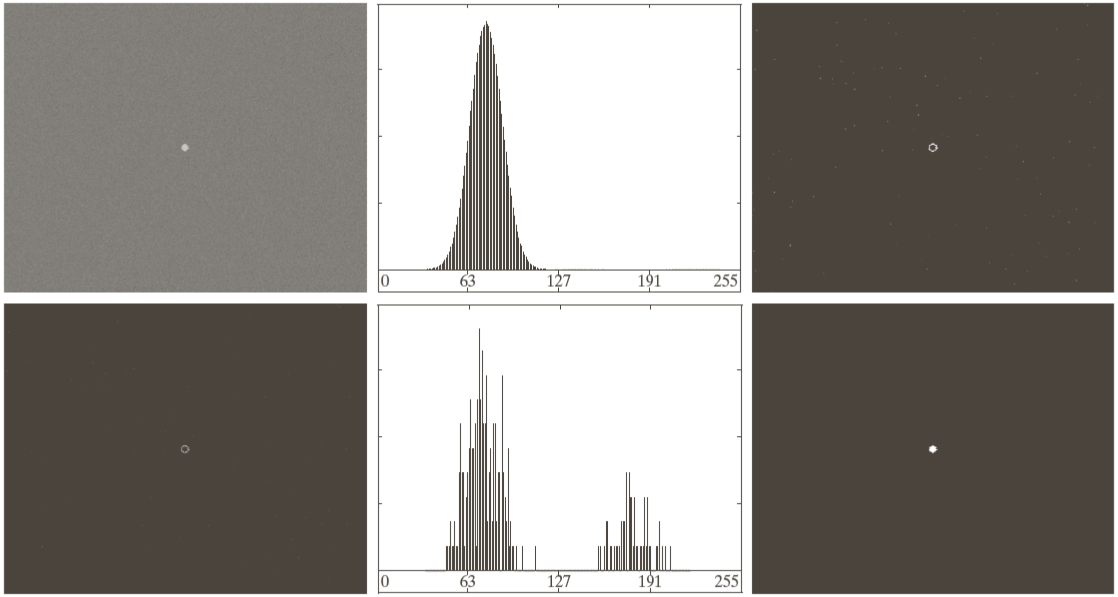
- Using edges:
  1. Compute the edge image $g(x,y)$ of $f(x,y)$.
  2. Specify the threshold value T.
  3. Threshold $g(x,y)$ with T, the output is a binary image $g_T(x,y)$. This image becomes a mask image.
  4. Compute the a histogram using pixels from $f(x,y)$ that correspond to the locations of the 1-valued pixels in $g_T(x,y)$.
  5. Use the histogram from Step 4 to segment $f(x,y)$ globally, e.g. with Otsu's method.

Using Edges in Segmentation

- The threshold value T should be larger than the minimum values in the image $f(x, y)$. Why?
- Many times T is a high percentile, e.g. high 90s

- Noisy image
- Histogram
- Gradient magnitude thresholded with 99.7%

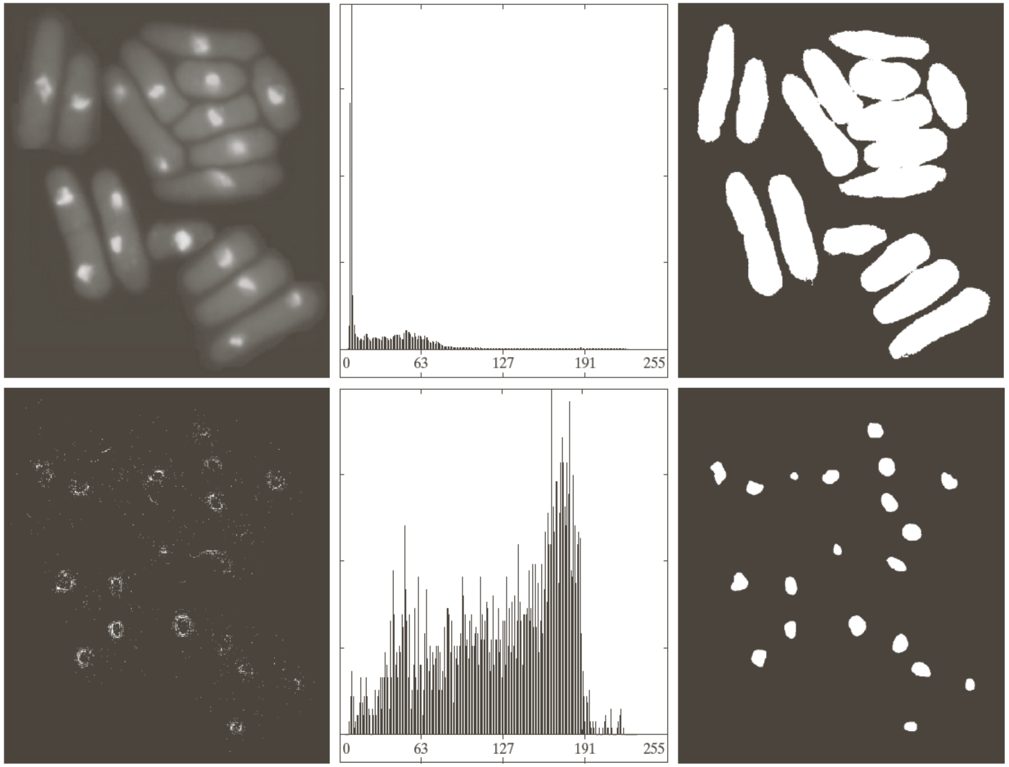
- a) times d)
- Histogram
- Segmentation



Using Edges in Segmentation

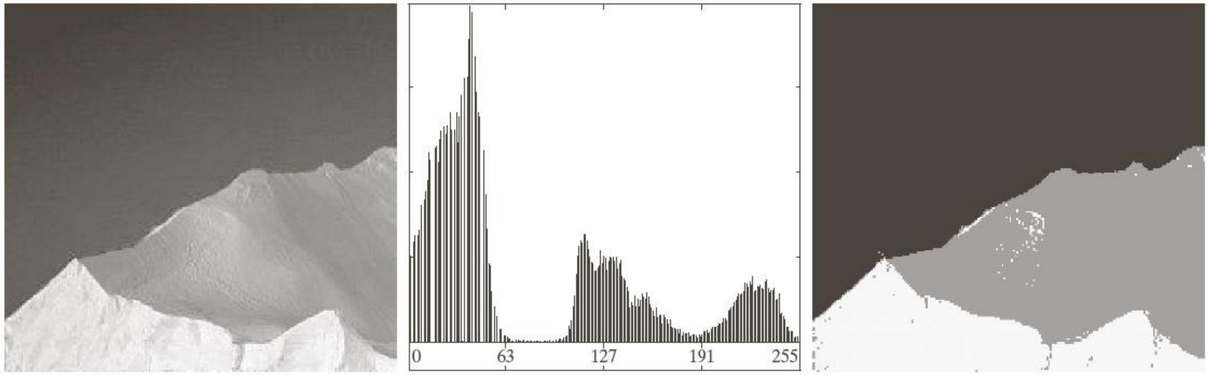
- Image of yeast cells
- Histogram
- Segmentation by Otsu

- Thresholded absolute Laplacian
- Histogram of a)*d)
- Segmentation



Multiple Thresholds

- Multiple classes can be found with a generalization of the between-class variance



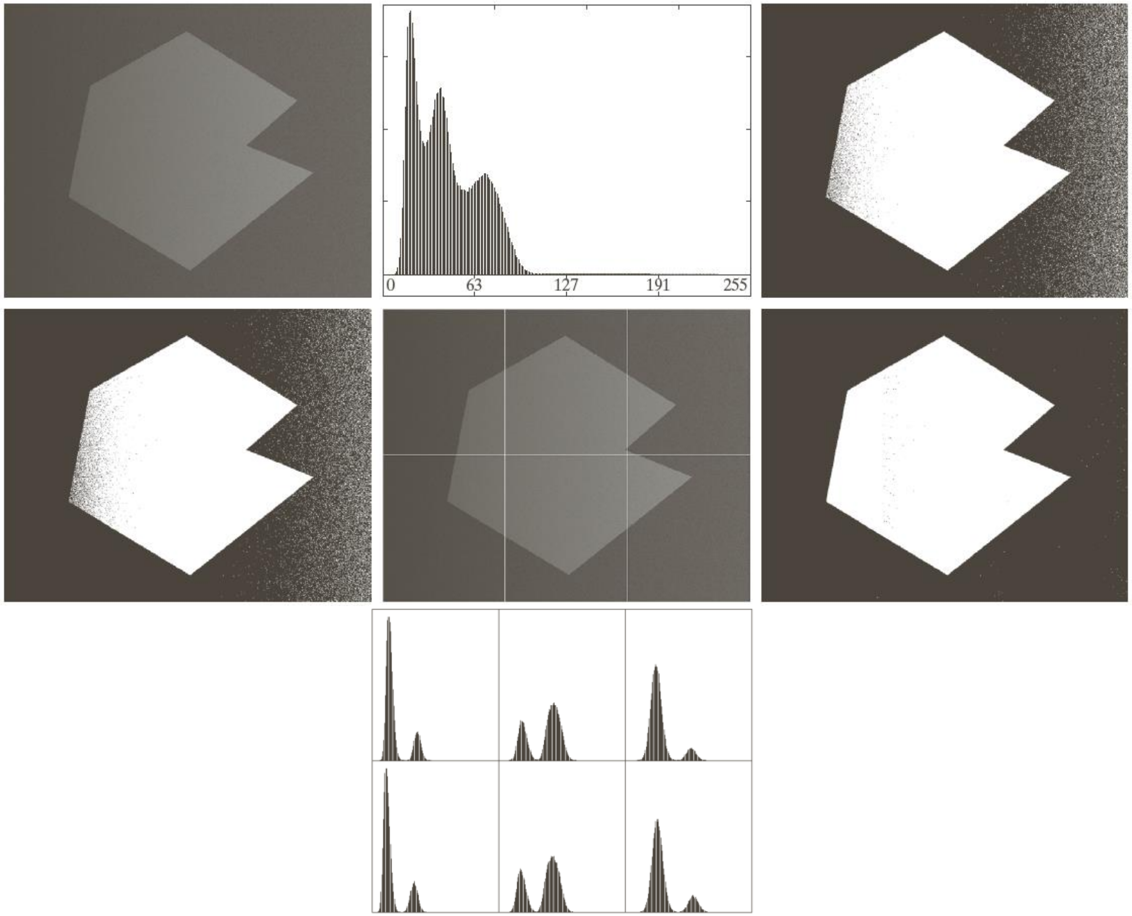
- Image of an iceberg
- Histogram
- Dual thresholds, Otsu’s method

Variable Thresholding

- Image partioning as one solution.

- Noisy image
- Histogram
- Segmentation (a)

- Segmentation by Otsu
- Image subdivision
- Segmentation of the subimages by Otsu
- Histograms of the subimages



Region-based Segmentation

- Approaches for region-based segmentation
  - Region growing
    - Bottom-up
    - Pixel by pixel
    - Pixel aggregation
  - Region splitting and merging
    - Top-down
    - Subdivisions of the image
    - Split and merge

Region Growing

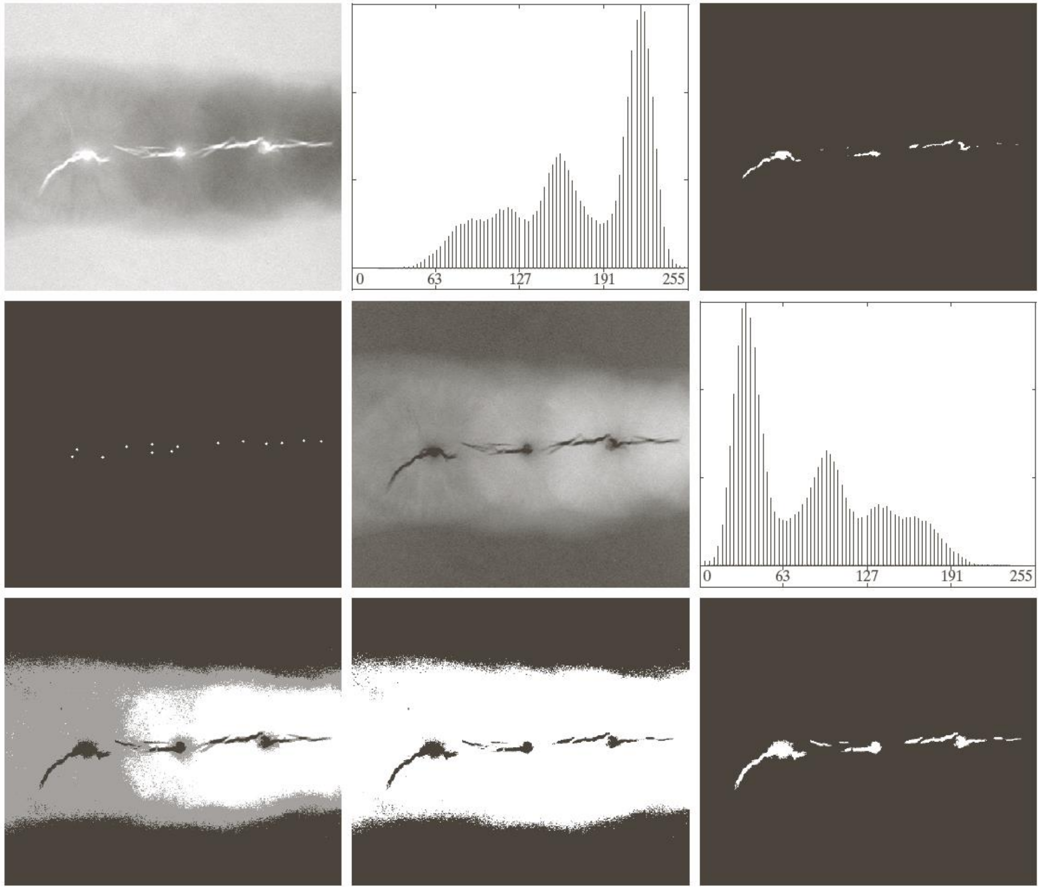
- Pixel aggregation
  - Start with a set of seed points and from these seed points grow regions by appending to each seed point those neighboring pixels that have similar properties (predicate Q, such as similar gray levels, texture, color)
- Which seed points?
- Similar properties? Should the criterion be constant or change as a function of current pixels?

Region Growing

- Image $f(x, y)$, seed array $S(x, y)$ with 1's, predicate Q
- Algorithm
  1. Find connected components in $S(x, y)$ and erode each connected component to one pixel.
  2. Find image f_Q such that at (x, y), set $f_Q(x, y) = 1$ if predicate Q is satisfied.
  3. Find image g by appending to each seed point in S all the 1-valued points in f_Q that are 8-connected to that seed point.
  4. Label each connected component with a label $1, 2, \dots, n$.

Region Growing

- X-ray image; histogram; initial seed image.
- Final seed image; difference between a) and c); and histogram of e).
- Thresholding with dual thresholds; one, smaller threshold; region growing

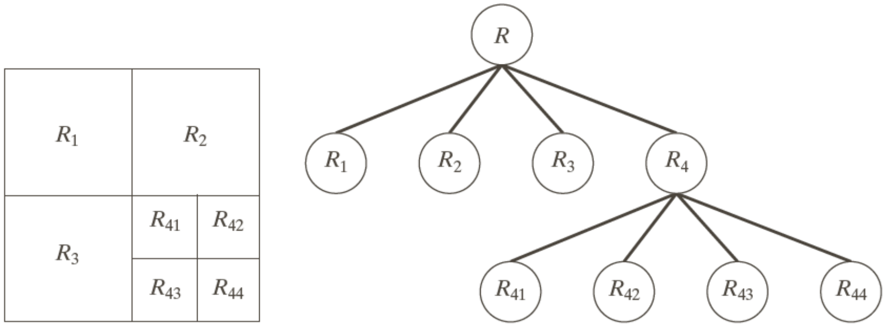


Region Splitting and Merging

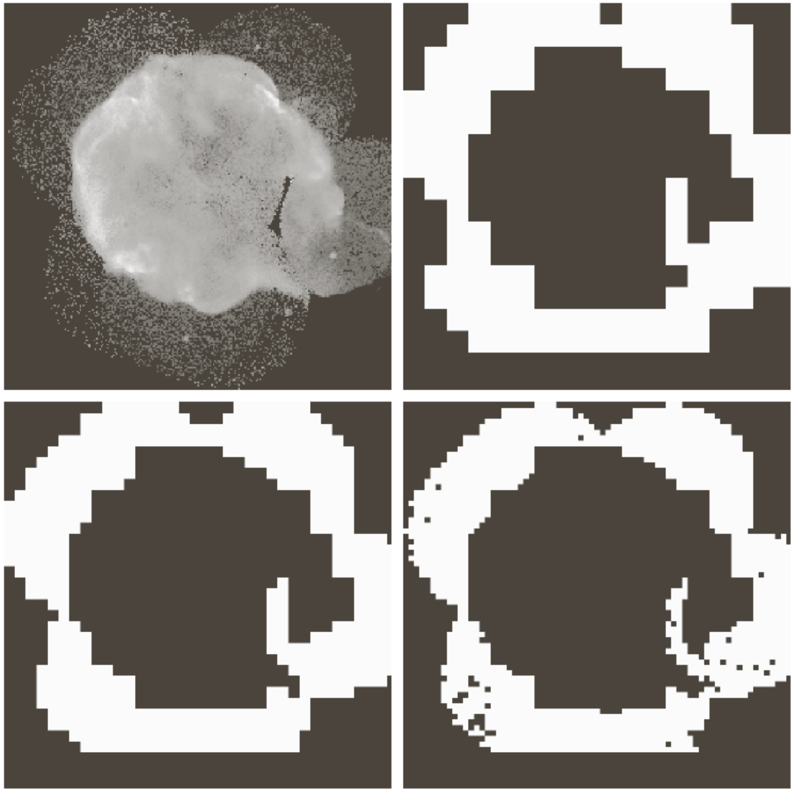
- Subdivide the image initially into a set of arbitrary, disjoint regions and then merge and/or split the regions in the attempt to satisfy the given conditions
- Method using a logical predicate $Q = Q(R)$ over the points in set R (similar values or not):
  1. Split into four disjoint quadrants any region R_i where $Q(R_i) = False$
  2. Merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k) = True$
  3. Stop when no further merging or splitting is possible

Region Splitting and Merging

- Quadtree from splitting the original image



- X-ray image, limiting a quadregion to 32x32, 16x16 and 8x8 pixels

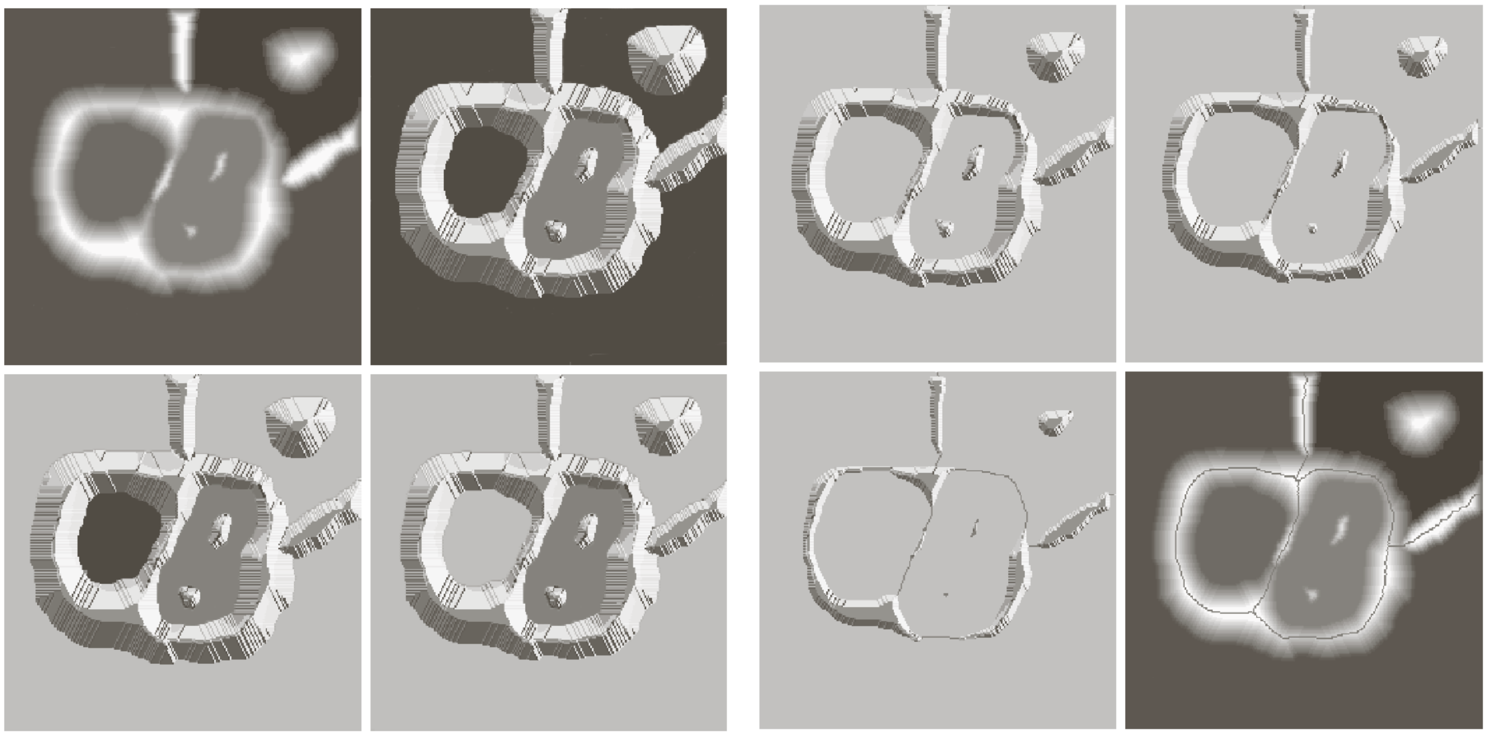


Segmentation using Watersheds

- Image have spatial coordinates and intensity, this can be seen as a topographic model
- Points have types
  - A point belong a regional minimum
  - The water would drop to one minimum from a point
    - Catchment basin, watershed
  - The water would fall to more than one minimum from a point
    - Divide lines or watershed lines

Segmentation using Watersheds

- Original image, topographic view; flooding in two stages
  - Further flooding, merging, dams, segmentation lines



Motion in Segmentation

- Relative displacement between the imaging system and the view.
- Basic idea: finding differences from multiple images taken at different time steps

$$d_{ij}(x,y) = \begin{cases} 1, \text{if } \left|f(x,y,t_i) - f(x,y,t_j)\right| > T \\ 0, \text{otherwise} \end{cases}$$

- Accumulative images are formed from a sequence of images as differences between subsequent images

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y) + 1, \text{if } |R(x,y)) - f(x,y,t_k)| > T \\ 0, \text{otherwise} \end{cases}$$

Motion in Segmentation

- Object moving to south-east direction
  - absolute, positive and negative accumulators

- Constructing a background image

Summary

Image acquisition → digital image

↓

Preprocessing → better image quality

↓

Segmentation → features for classification/clustering