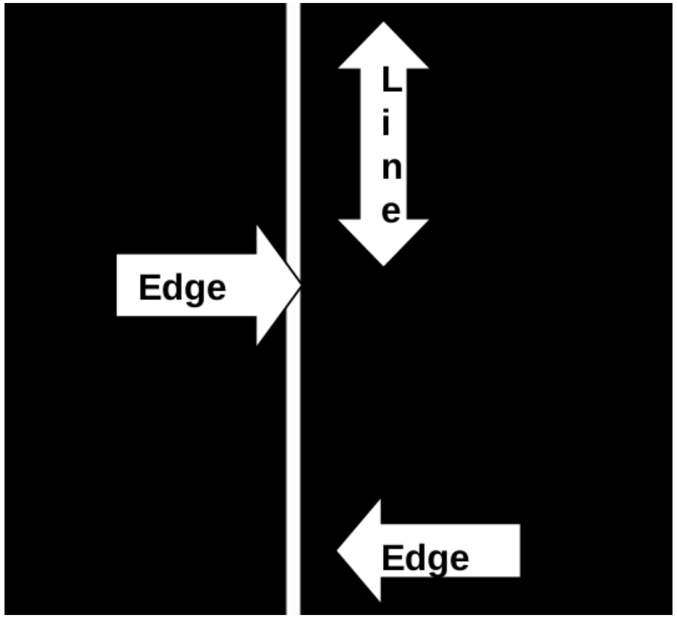
Machine Vision

Image Processing - Geometric features detection
lecture 4

Adam Szmigielski
aszmigie@pjwstk.edu.pl
materials: $ftp(public)://aszmigie/WMAEnglish$

Edges and Lines Relationship



- Edges and lines are perpendicular,
- The line shown here is vertical and the edge direction is horizontal.
- In this case the transition from black to white occurs along a row, this is the edge direction, but the line is vertical along a column.

Goals of Edge Detection

- Produce a line drawing of a scene from an image of that scene.
- Important features can be extracted from the edges of an image (e.g., corners, lines, curves).
- These features are used by higher-level computer vision algorithms (e.g., segmentation, recognition).

Goals of an Edge Detector

Goal to construct edge detection operators that extracts

- the orientation information (information about the direction of the edge) and
- the strength of the edge.

Some methods can return information about the existence of an edge at each point for faster processing

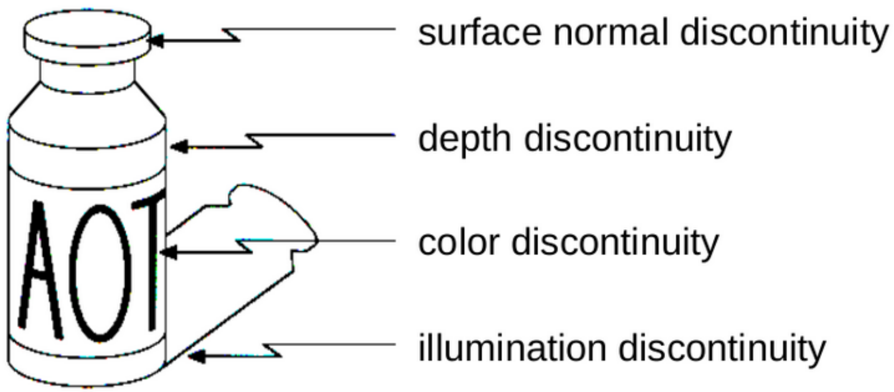
What Causes Intensity Changes?

Geometric events

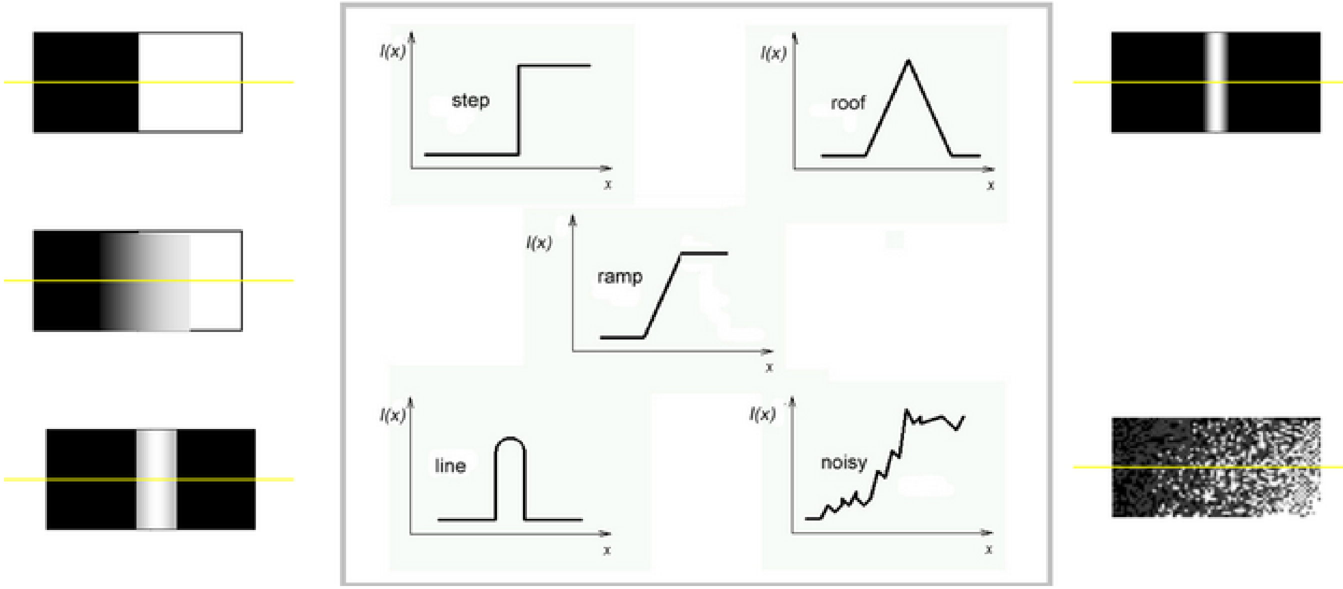
- surface orientation (boundary) discontinuities
- depth discontinuities
- color and texture discontinuities

Non-geometric events

- illumination changes,
- specularities,
- shadows
- inter-reflections

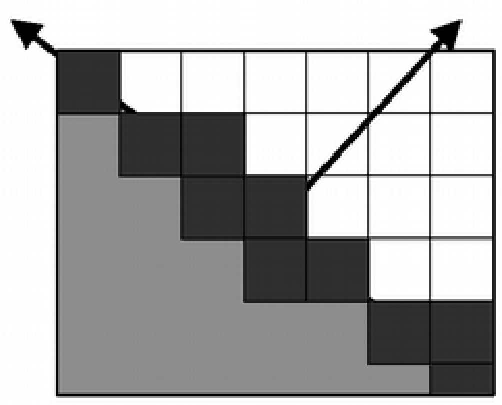


Edge Definition



- Edge is a boundary between two regions with relatively distinct gray level properties.
- Edges are pixels where the brightness function changes abruptly.
- Edge detectors are a collection of very important local image preprocessing methods used to locate (sharp) changes in the intensity function.

Edge Descriptors



- Edge Size: $S = \sqrt{dx^2 + dy^2}$
- Edge direction: unit vector to perpendicular to the edge normal. $\alpha = arctan(\frac{dy}{dx})$
- Edge normal: unit vector in the direction of maximum intensity change.
- Edge position or center: the image position at which the edge is located.
- Edge strength: related to the local image contrast along the normal.

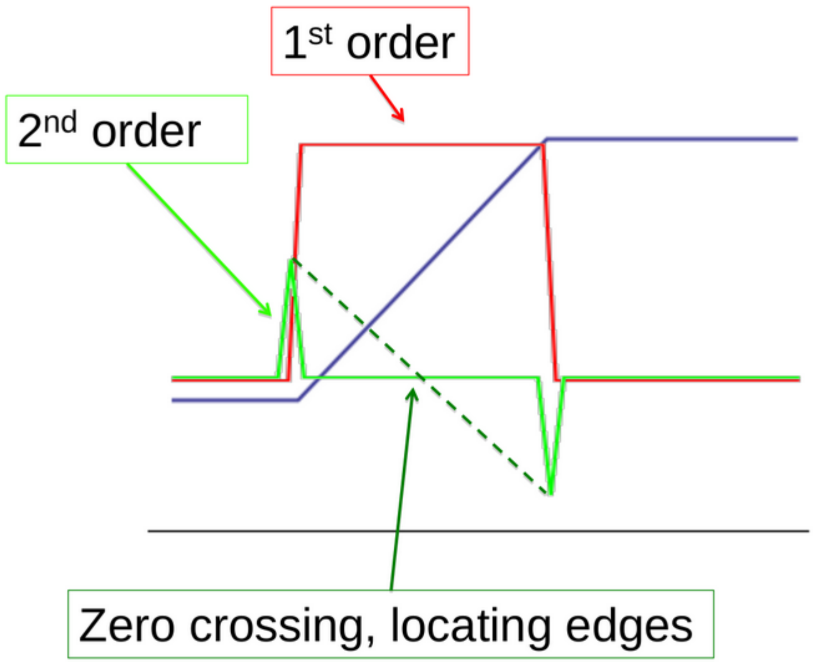
Main Steps in Edge Detection

1. **Smoothing:** suppress as much noise as possible, without destroying true edges,
2. **Enhancement:** apply differentiation to enhance the quality of edges (i.e. sharpening),
3. **Thresholding:** determine which edge pixels should be discarded as noise and which should be retained,
4. **Localization:** determine the exact edge location.

Edge Detection using Derivatives

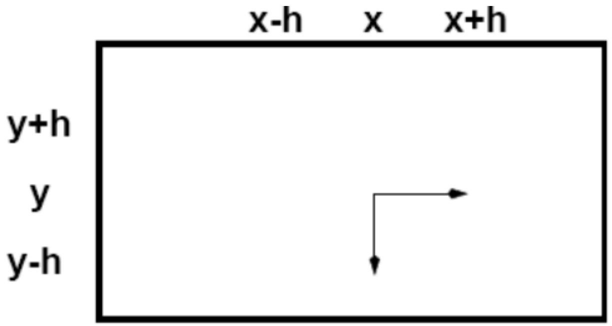
- Calculus describes changes of continuous functions using derivatives.
- An image is a 2D function, so operators describing edges are expressed using partial derivatives.
- Points which lie on an edge can be detected by either:
  - detecting local maxima or minima of the first derivative,
  - detecting the zero-crossing of the second derivative

1st and 2nd order derivatives



- 1st order derivative gives thick edges,
- 2nd order derivative gives double edge,
- 2nd order derivatives enhance fine detail much better.

Edge Detection Using First Derivative



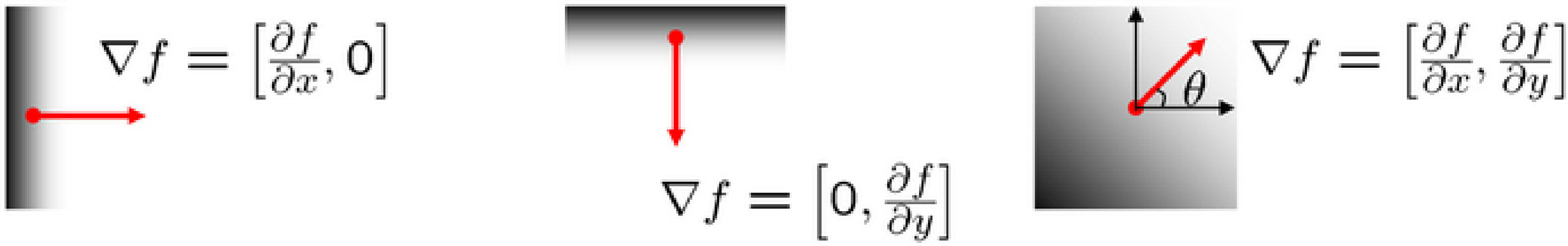
- Backward difference: $f'(x) \approx f(x) - f(x-1)$
- Forward difference: $f'(x) = f(x+1) - f(x)$
- Central difference: $f'(x) = f(x+1) - f(x-1)$

Edge detection

- Gradient-based edge operators
  - Prewitt
  - Sobel
  - Roberts
- Laplacian zero-crossings
- Canny edge detector
- Hough transform for detection of straight lines
- Circle Hough Transform

Edge Descriptors Using Gradient

- The gradient of an image: $\nabla f = [\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}]$
- The gradient points in the direction of most rapid change in intensity



- The gradient direction (orientation of edge normal) is given by:

$$\alpha[x, y] \approx \tan^{-1}(\frac{\partial f(x,y)}{\partial y} / \frac{\partial f(x,y)}{\partial x})$$

- The edge strength is given by the gradient magnitude:

$$M[x, y] \approx \sqrt{(\frac{\partial f(x,y)}{\partial x})^2 + (\frac{\partial f(x,y)}{\partial y})^2}$$

Edge Detection Using Second Derivative

- Approximate finding maxima/minima of gradient magnitude by finding places where:

$$\frac{\partial f^2(x,y)}{\partial x^2} = 0$$

- Can't always find discrete pixels where the second derivative is zero - look for zero-crossing instead.

Gradient-based edge detection

- dea (continous-space): local gradient magnitude indicates edge strength

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

- Digital image: use finite differences to approximate derivatives:

difference $\begin{pmatrix} -1 & 1 \end{pmatrix}$

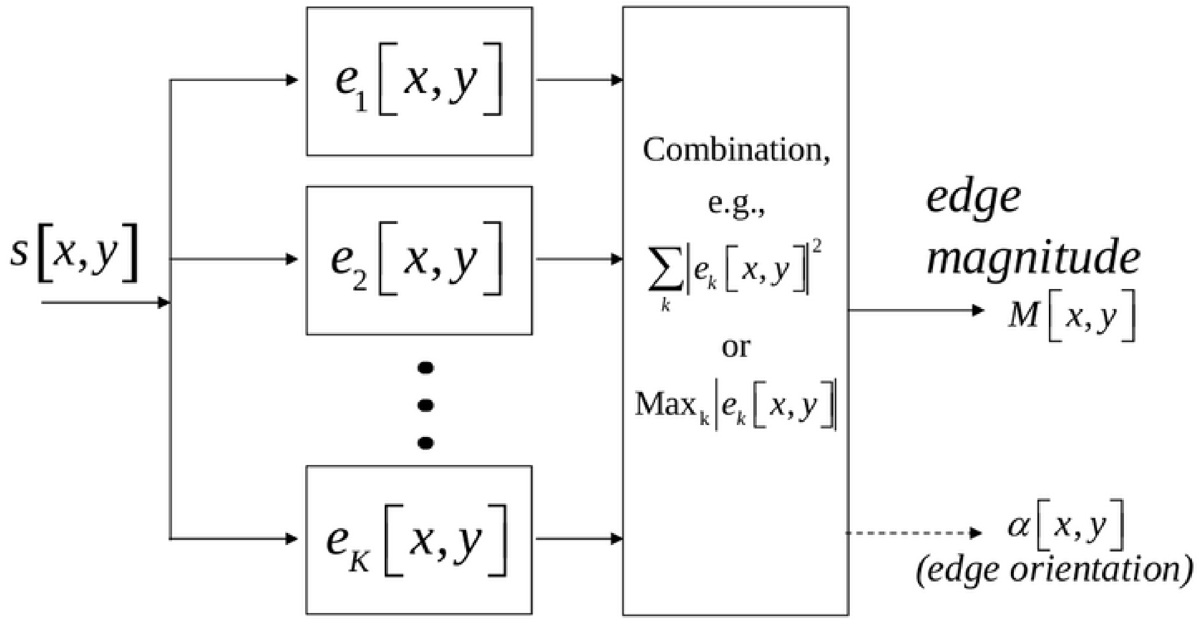
central difference $\begin{pmatrix} -1 & [0] & 1 \end{pmatrix}$

Prewitt $\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$

Sobel $\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix}$

- Edge templates

Practical edge detectors



- Edges can have any orientation,
- Typical edge detection scheme uses $K = 2$ edge templates,
- Some use $K > 2$.

Gradient filters (K=2)

Central Difference $\begin{pmatrix} 0 & 0 & 0 \\ -1 & [0] & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & [0] & 0 \\ 0 & 1 & 0 \end{pmatrix}$ Roberts $\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$

Prewitt $\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$

Sobel $\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & -1 \\ 0 & [0] & 0 \\ 1 & 2 & 1 \end{pmatrix}$

Gradient filters (K=8)

Kirsch $\begin{pmatrix} +5 & +5 & +5 \\ -3 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & +5 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & -3 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & +5 \\ -3 & +5 & +5 \end{pmatrix}$

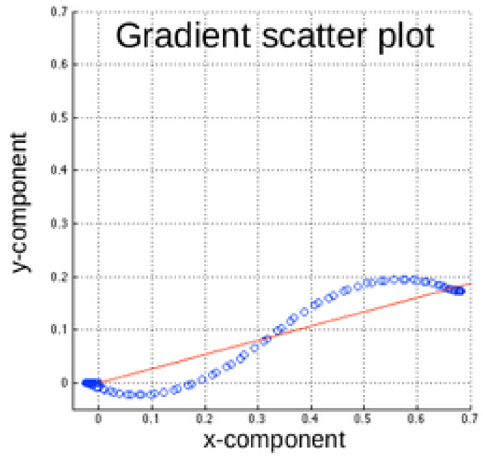
$\begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & -3 \\ +5 & +5 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & +5 & -3 \end{pmatrix} \begin{pmatrix} +5 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & -3 & -3 \end{pmatrix} \begin{pmatrix} +5 & +5 & -3 \\ +5 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix}$

Edge orientation

Central Difference

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & [0] & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

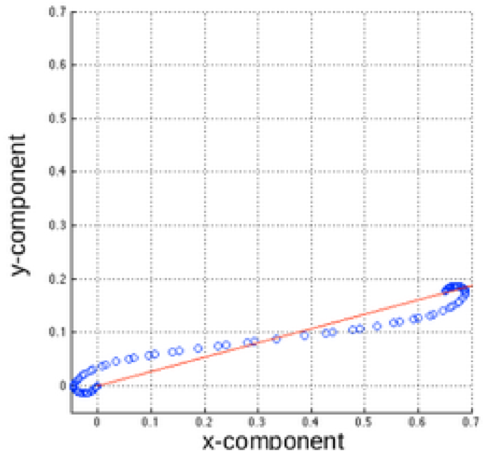
$$\begin{pmatrix} 0 & -1 & 0 \\ 0 & [0] & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



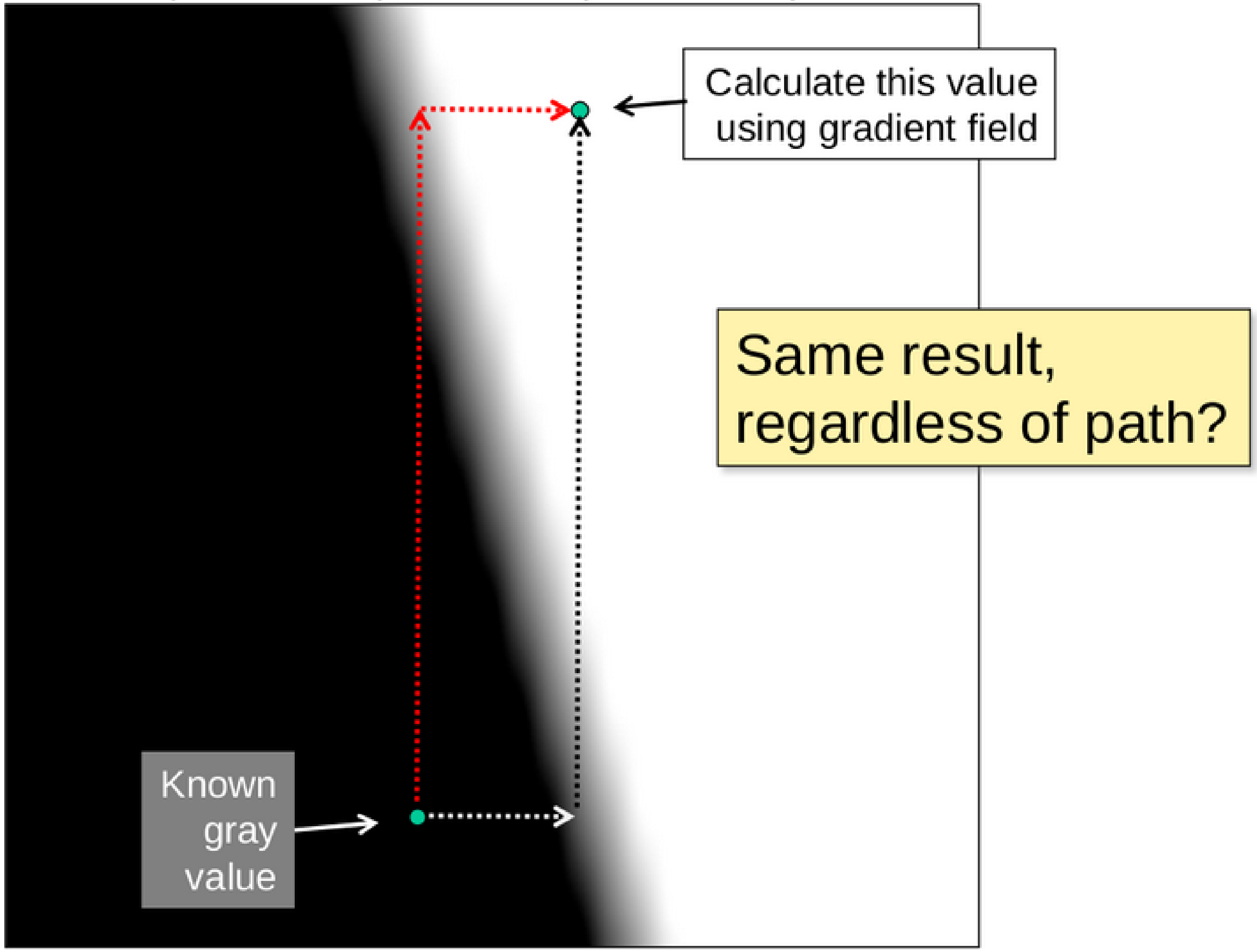
Roberts

$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$



Gradient consistency problem

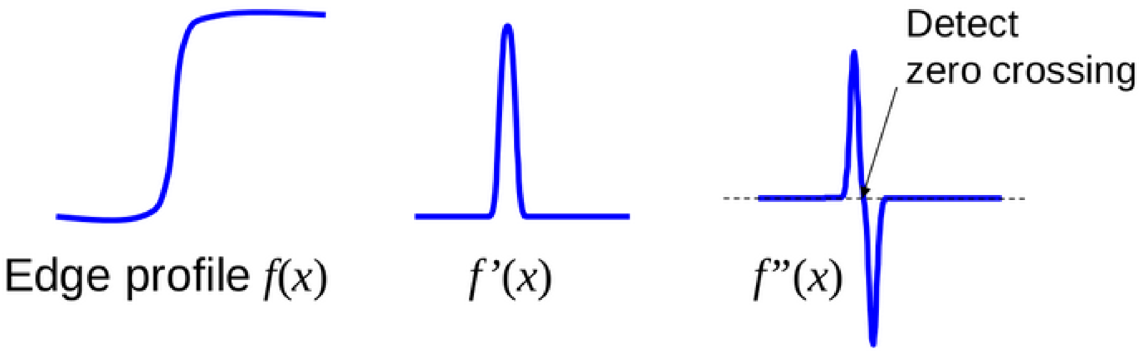


Laplacian operator

Detect edges by considering second derivative:

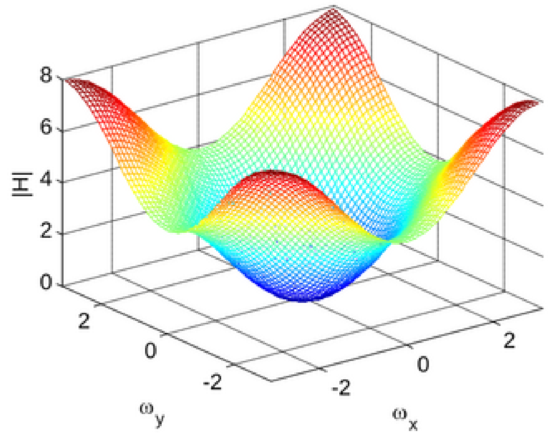
$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

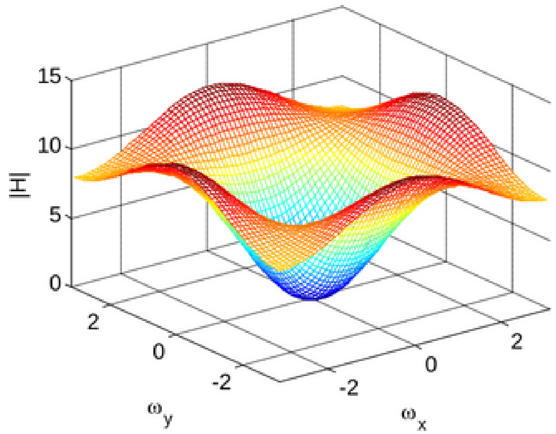
- Isotropic (rotationally invariant) operator
- Zero-crossings mark edge location



Approximations of Laplacian operator by 3×3 filter

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & [-4] & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 1 \\ 1 & [-8] & 1 \\ 1 & 1 & 1 \end{pmatrix}$$





- Sensitive to very fine detail and noise -> blur image first,
- Responds equally to strong and weak edges -> suppress zero-crossings with low gradient magnitude.

Second Derivative in 2D: Laplacian

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

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

$$\nabla^2 f = -4f(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & -2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

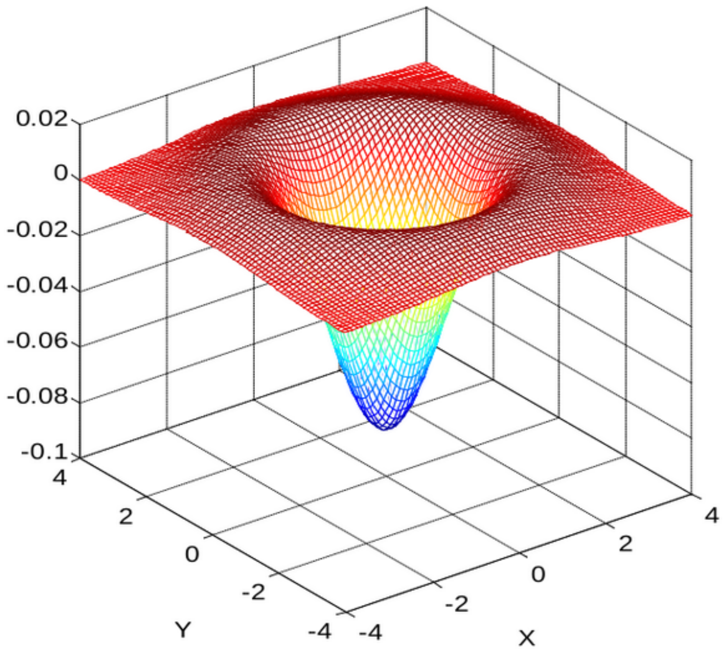
Properties of Laplacian

- It is an isotropic operator.
- It is cheaper to implement than the gradient (i.e., one mask only).
- It does not provide information about edge direction.
- It is more sensitive to noise (i.e. differentiates twice).

Laplacian of Gaussian (LoG)

Filtering of image with Gaussian and Laplacian operators can be combined into convolution with Laplacian of Gaussian (LoG) operator

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

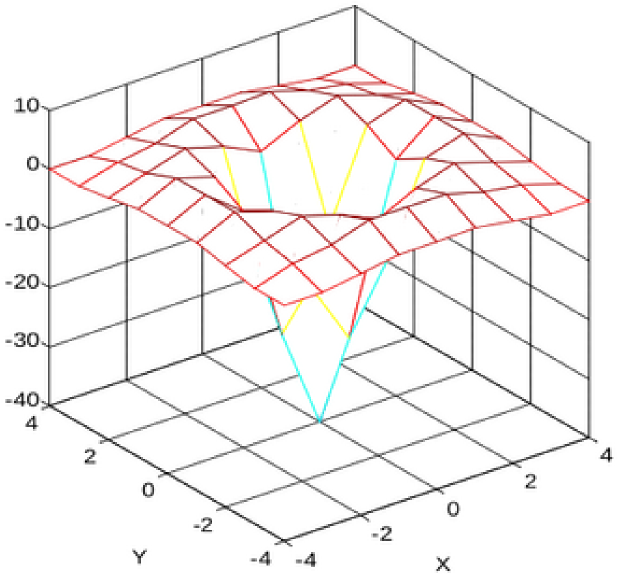


$\sigma = \sqrt{2}$

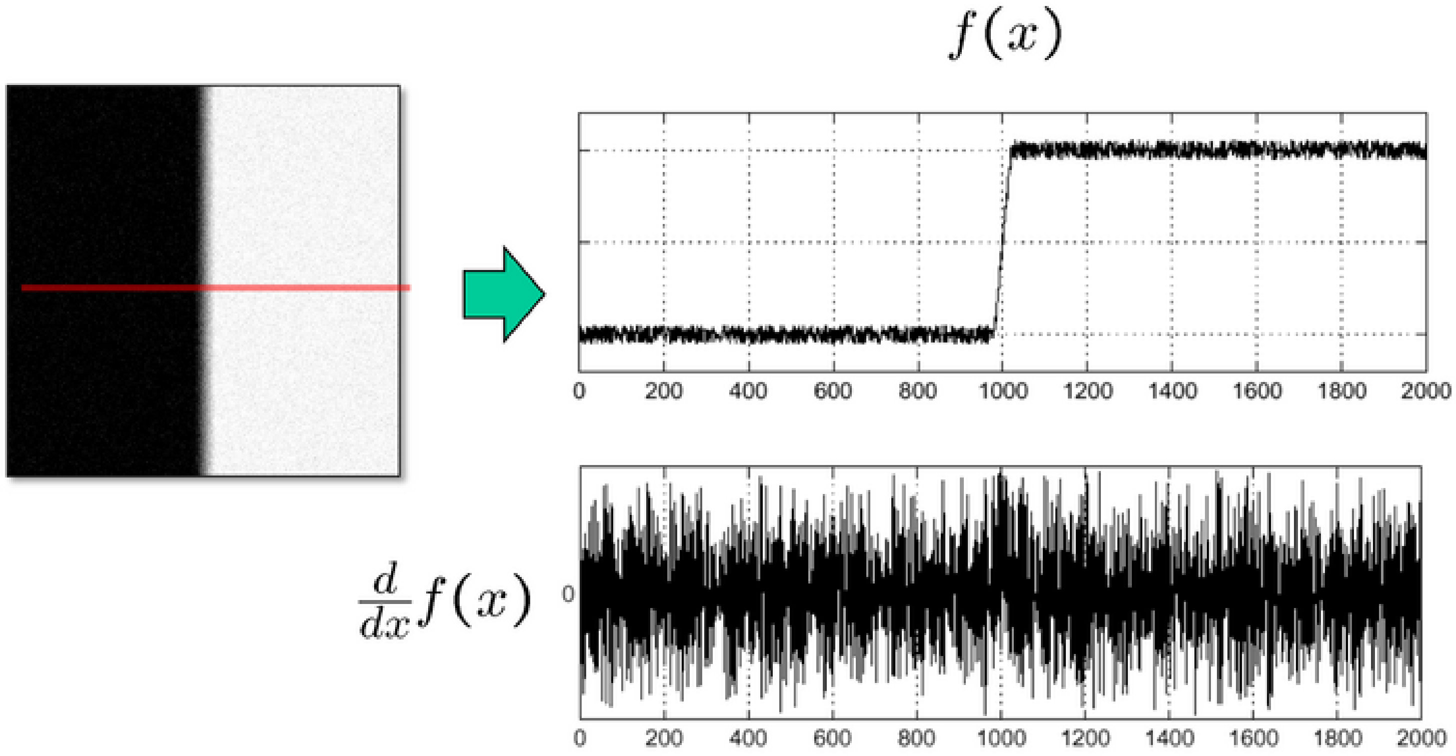
Discrete approximation of Laplacian of Gaussian

$\sigma = \sqrt{2}$

0	0	1	2	2	2	1	0	0
0	2	3	5	5	5	3	2	0
1	3	5	3	0	3	5	3	1
2	5	3	-12	-23	-12	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	3	-12	-23	-12	3	5	2
1	3	5	3	0	3	5	3	1
0	2	3	5	5	5	3	2	0
0	0	1	2	2	2	1	0	0



Noise Effect on Derivates



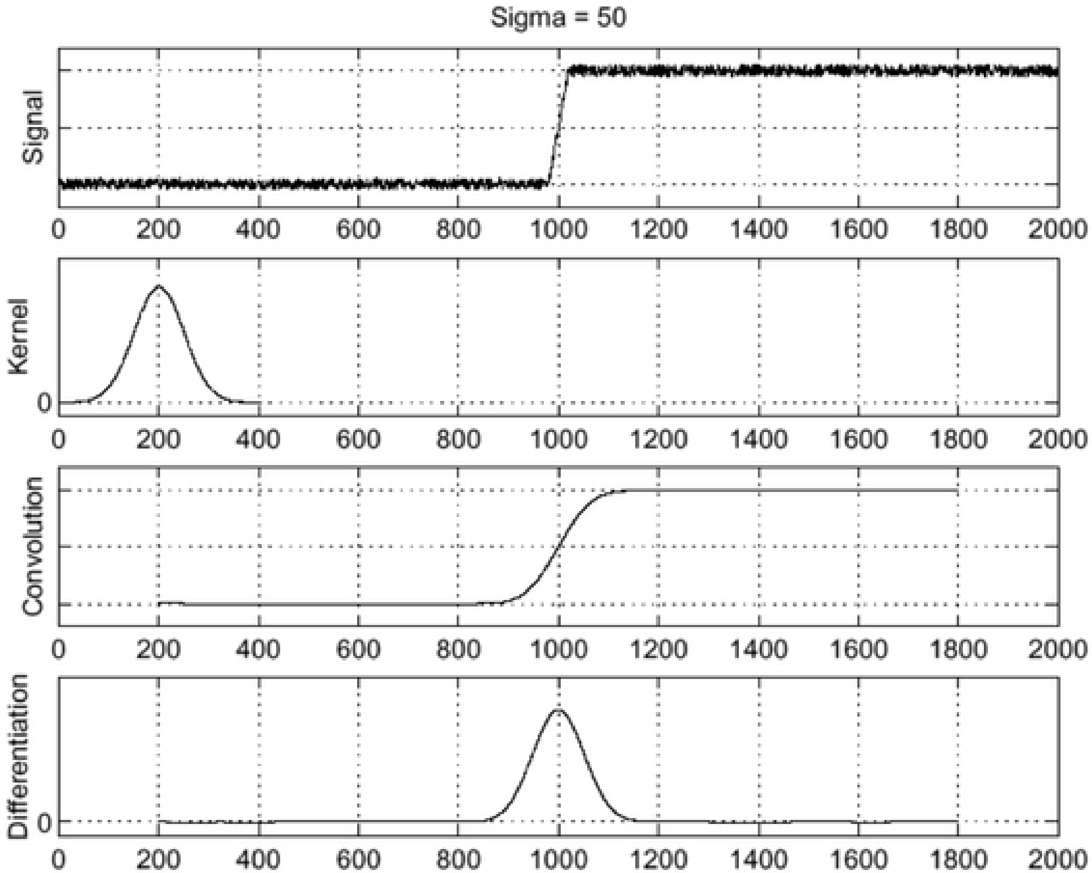
Effect of Smoothing on Derivatives

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



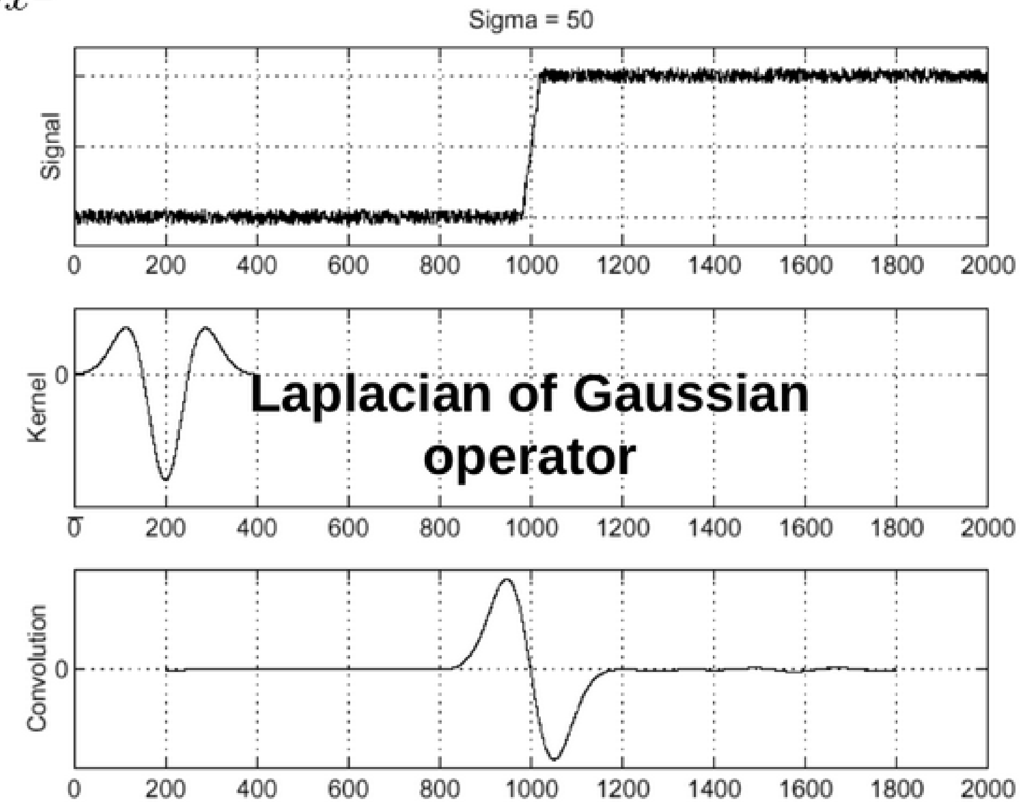
Effect of Smoothing on Laplacian of Gaussian

$$\frac{\partial^2}{\partial x^2}(h \star f)$$

f

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



Canny edge detector

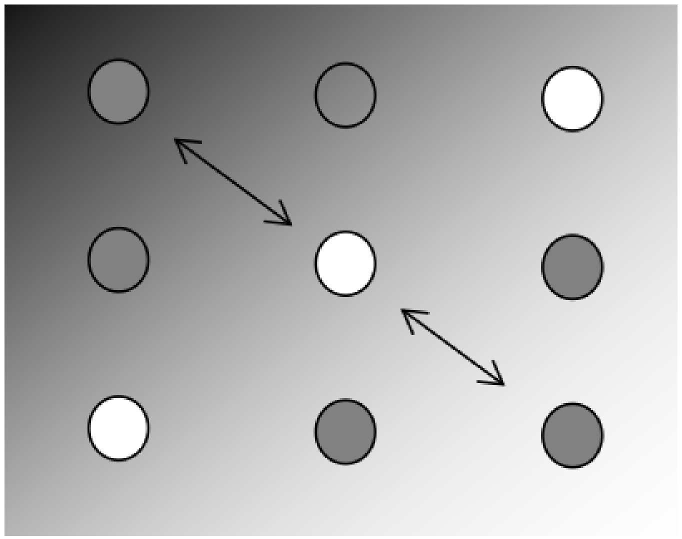
1. Smooth image with a Gaussian filter,
2. Approximate gradient magnitude and angle (use Sobel, Prewitt . . .)

$$M[x,y] \approx \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2} \quad \alpha[x,y] \approx \tan^{-1}\left(\frac{\partial f(x,y)}{\partial y} / \frac{\partial f(x,y)}{\partial x}\right)$$

3. Apply nonmaxima suppression to gradient magnitude,
4. Double thresholding to detect strong and weak edge pixels,
5. Reject weak edge pixels not connected with strong edge pixels.

Canny nonmaxima suppression

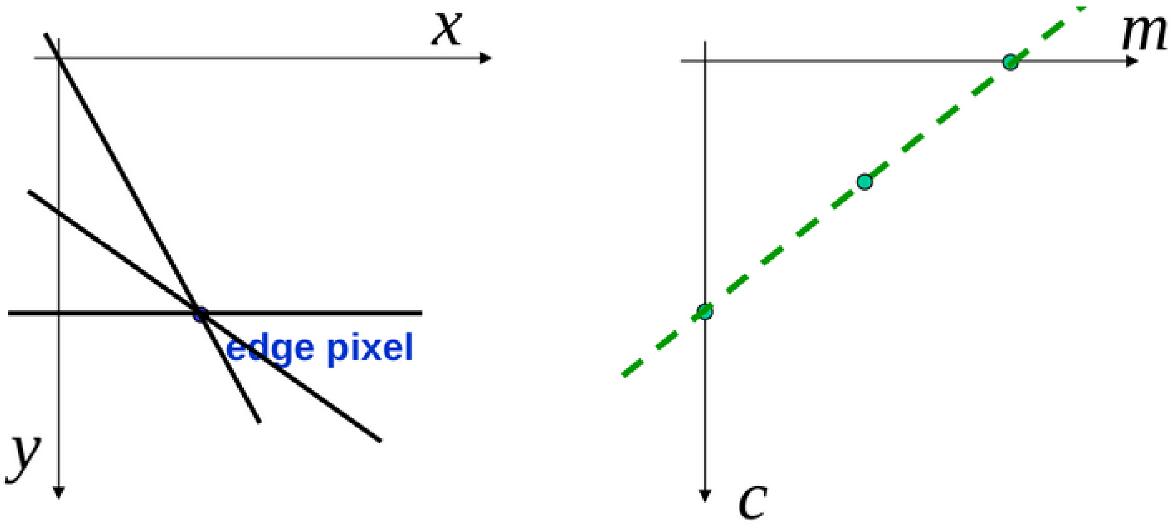
- Quantize edge normal to one of four directions: horizontal, -45° , vertical, $+45^\circ$
- If $M[x, y]$ is smaller than either of its neighbors in edge normal direction -> suppress; else keep.



Canny thresholding and suppression of weak edges - hysteresis thresholding

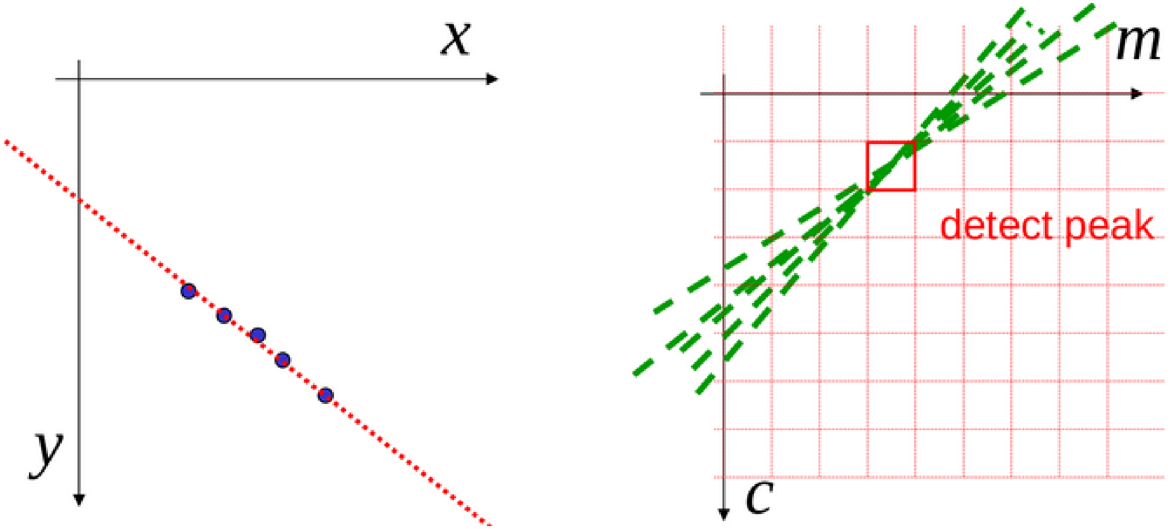
- Double-thresholding of gradient magnitude:
  - Strong edge: $M[x,y] \geq \Theta_{high}$ definitely an edge,
  - Weak edge: $\Theta_{high} > M[x,y] \geq \Theta_{low}$ maybe an edge,
  - No edge: $M[x,y] < \Theta_{low}$ definitely not an edge,
- Typical setting: $\Theta_{high}/\Theta_{low} = 2\ldots3$
- Region labeling of edge pixels
- Reject regions without strong edge pixels
- For "maybe" edges, decide on the edge if neighboring pixel is a strong edge.

Hough transform



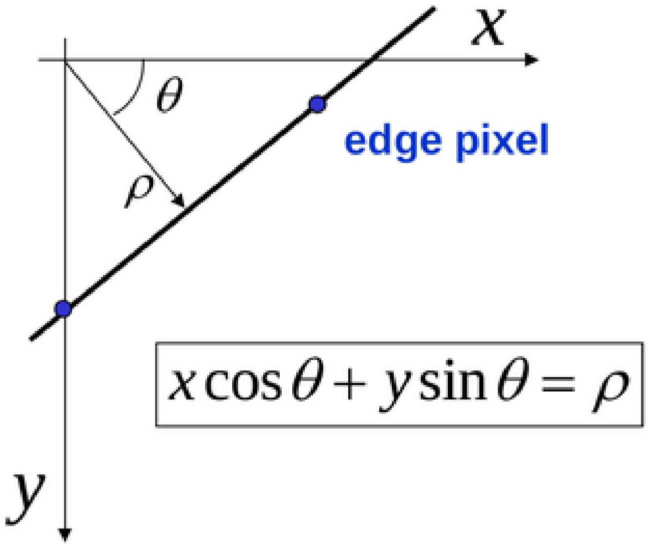
- Problem: fit a straight line (or curve) to a set of edge pixels
- Hough transform (1962): generalized template matching technique
- Consider detection of straight lines $y = mx + c$

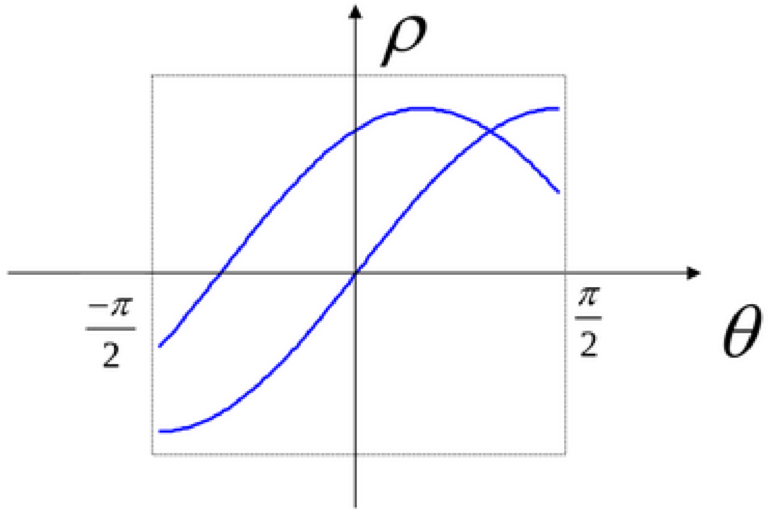
Hough transform



- Subdivide (m, c) plane into discrete "bins" initialize all bin counts by 0,
- Draw a line in the parameter space $[m, c]$ for each edge pixel $[x, y]$ and increment bin counts along line.
- Detect peak(s) in $[m, c]$ plane.

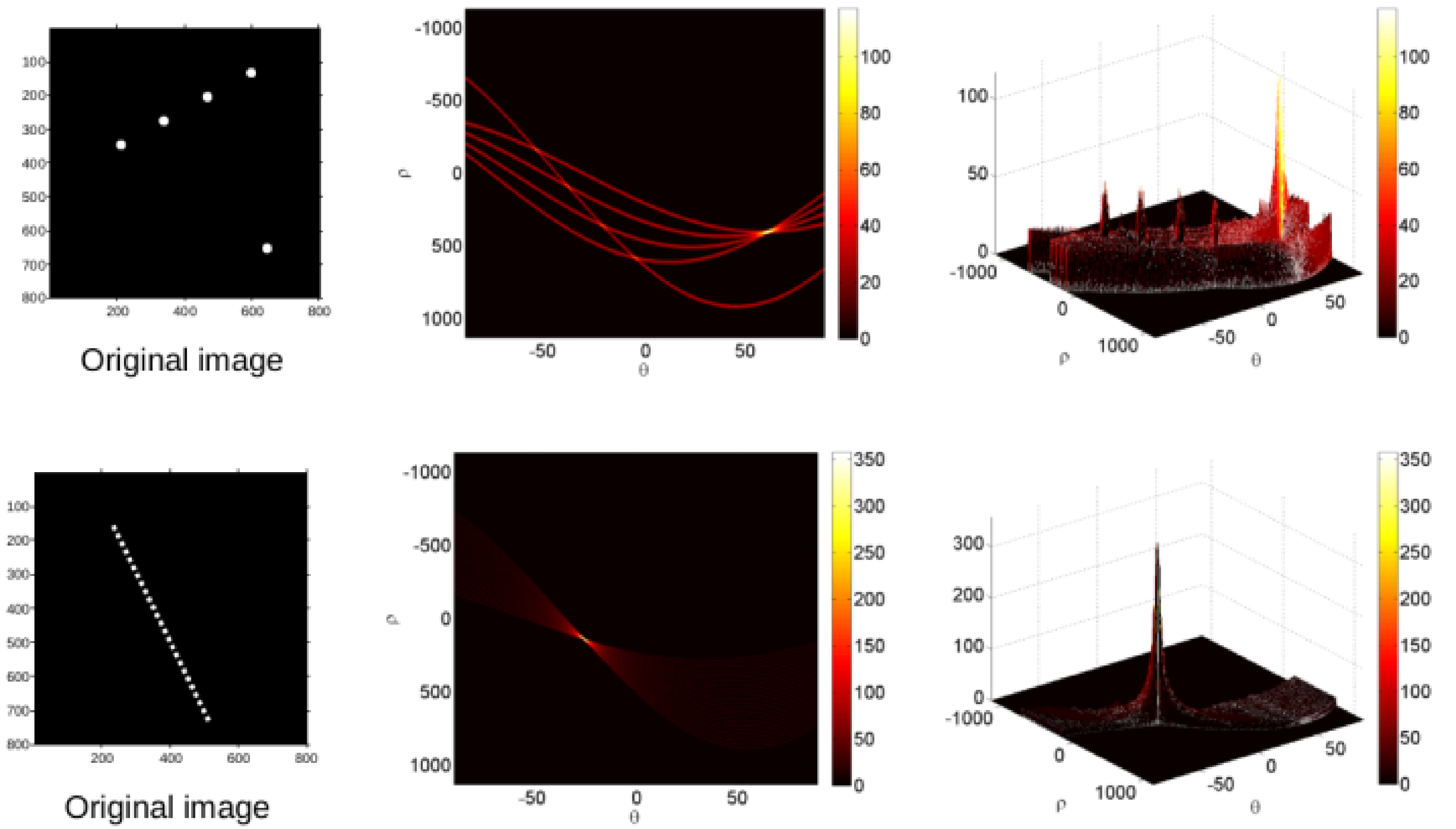
Hough transform



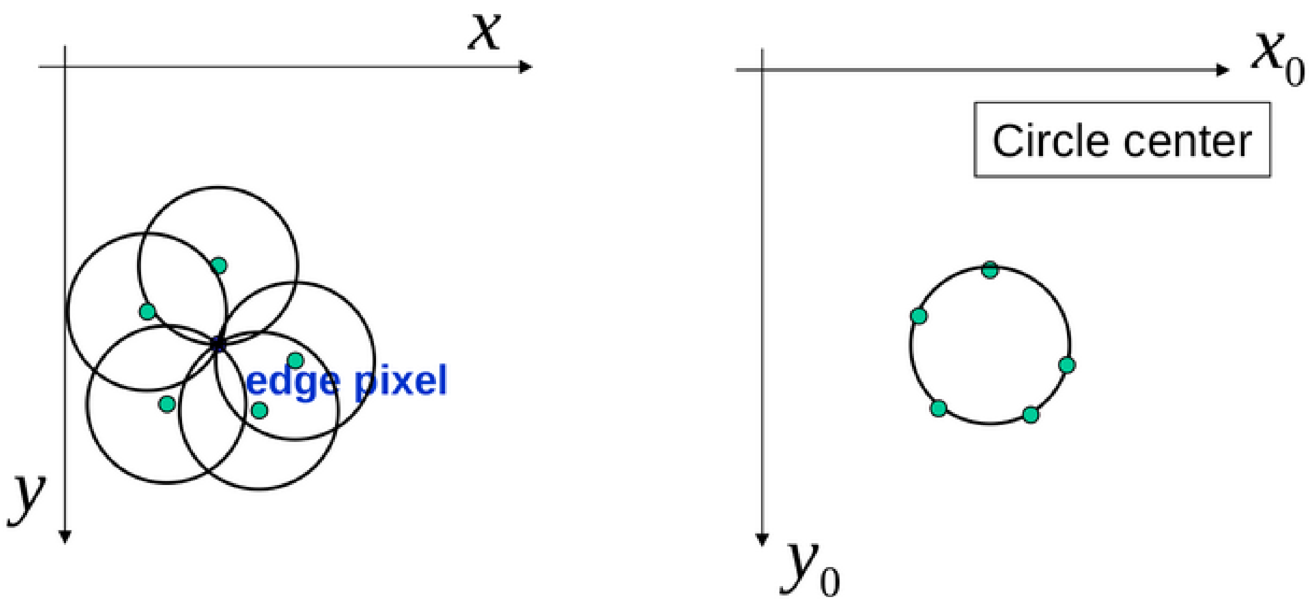


- Alternative parameterization avoids infinite-slope problem
- Similar to Radon transform

Hough transform example

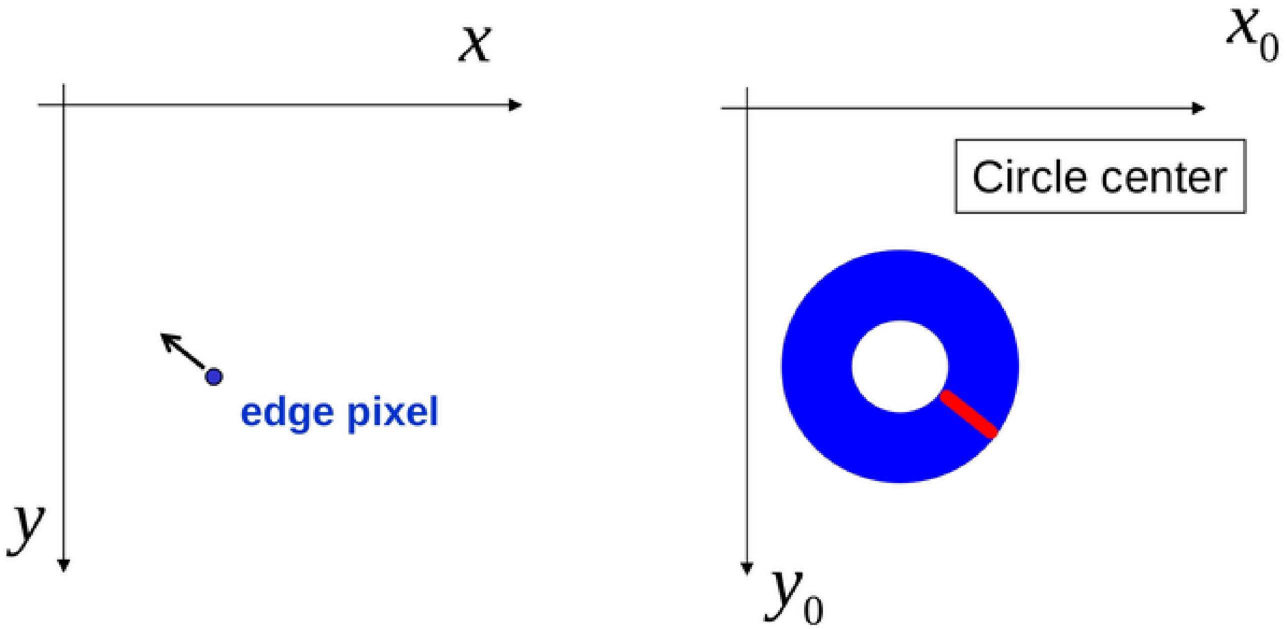


Circle Hough Transform



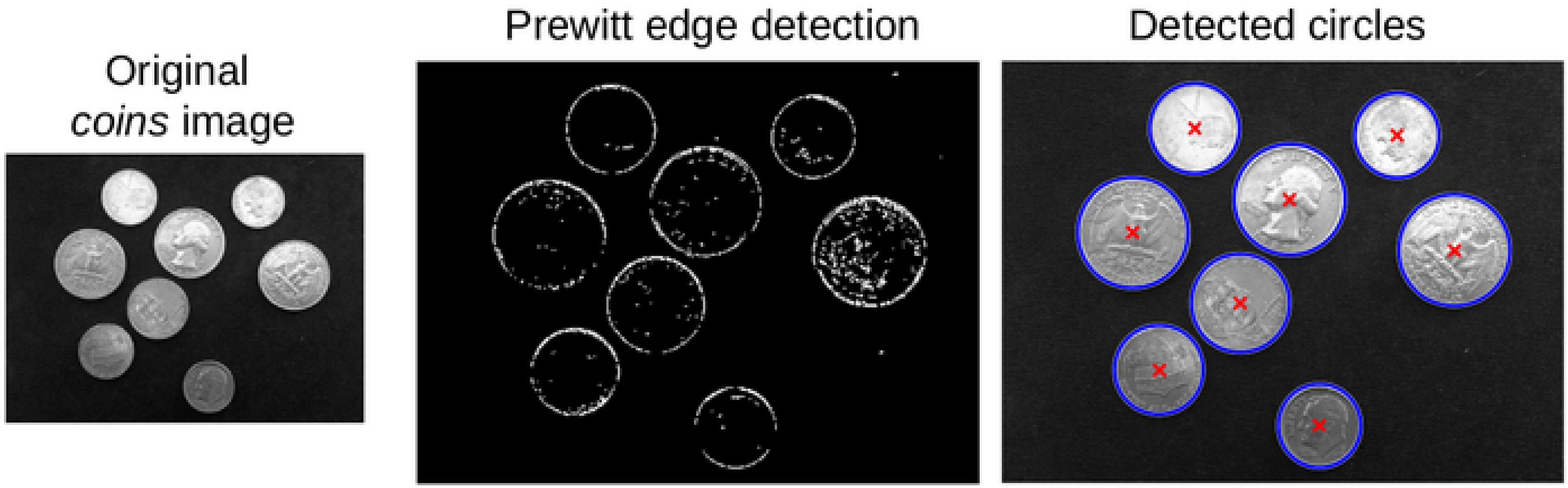
- Find circles of fixed radius r
- Equivalent to convolution (template matching) with a circle

Circle Hough Transform for unknown radius



- $3 - d$ Hough transform for parameters (x_0, y_0, r),
- $2 - d$ Hough transform aided by edge orientation -> "spokes" in parameter space.

Circle detection by Hough transform - example



Corner detection

Corner in the concept of " sharply curving edge ", intersection of 2 edges, lines

- Type of points of interest, points of interest (other: line endings, light or dark points),
- The first algorithms that detect corners:
  - edge detection,
  - tracing the extracted edges and detecting a sudden change in their direction,
- Newer generation of algorithms - high gradient curvature,
- It is recommended to smooth the image earlier.

Task for laboratory

A certain number of 5 zł coins and a certain 5 gr. coins are given. The coins can be located in or out of the tray. Write a program that will determine the number and type of coins in and out of the tray. Test the program on pictures. Tune the system to get the best possible efficiency. Consider pre image analysis (such as blurring, thresholding, edge detection, etc.). Use both Hough transformations. The effectiveness of the system will be assessed.