U5. Edge Detection

SJK002 Computer Vision

Master in Intelligent Sytems

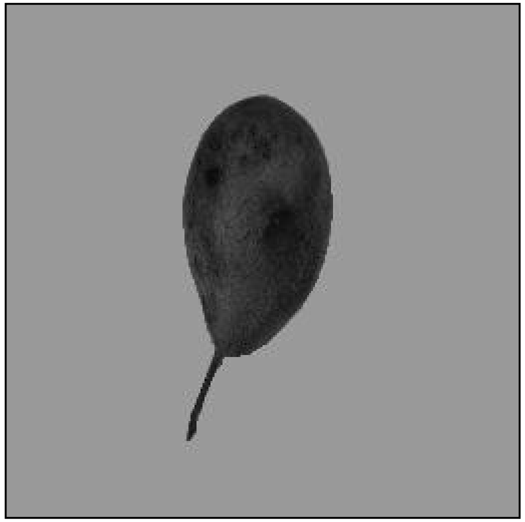
UNIVERSITAT
JAUME•I

Index

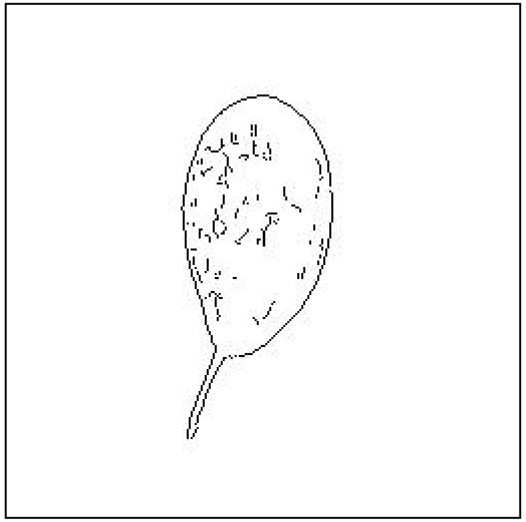
- Introduction
  - What is an edge?
  - Steps in edge detection
- Edge detectors based on image gradient
  - What is image gradient?
  - Detectors
- Edge detectors based on Laplacian operator
  - What is the Laplacian?
  - Detectors

Introduction: What is an edge?

- Edge = significant change in image intensity
- Usefulness:
  - Object features
  - Object limits



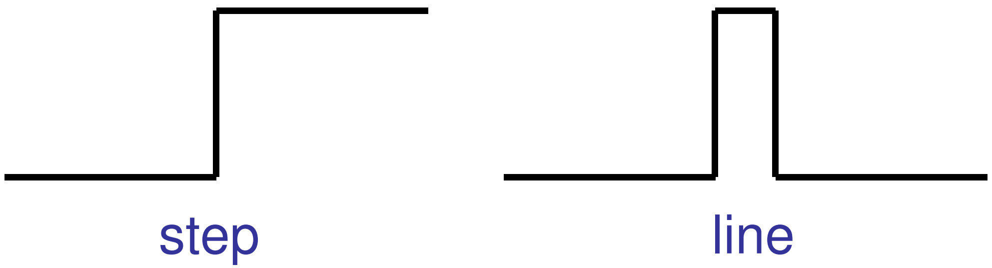
Pear image



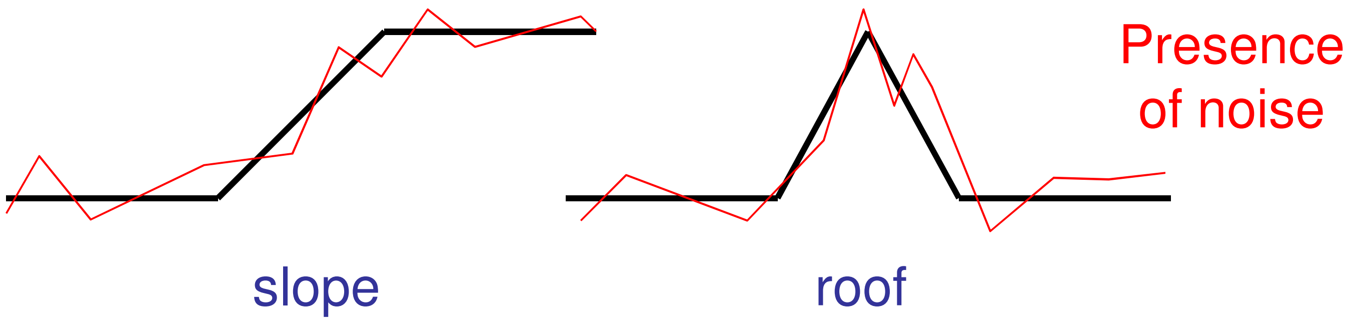
Pear image borders

Introduction: What is a edge?

- Types of edges
  - Image discontinuities

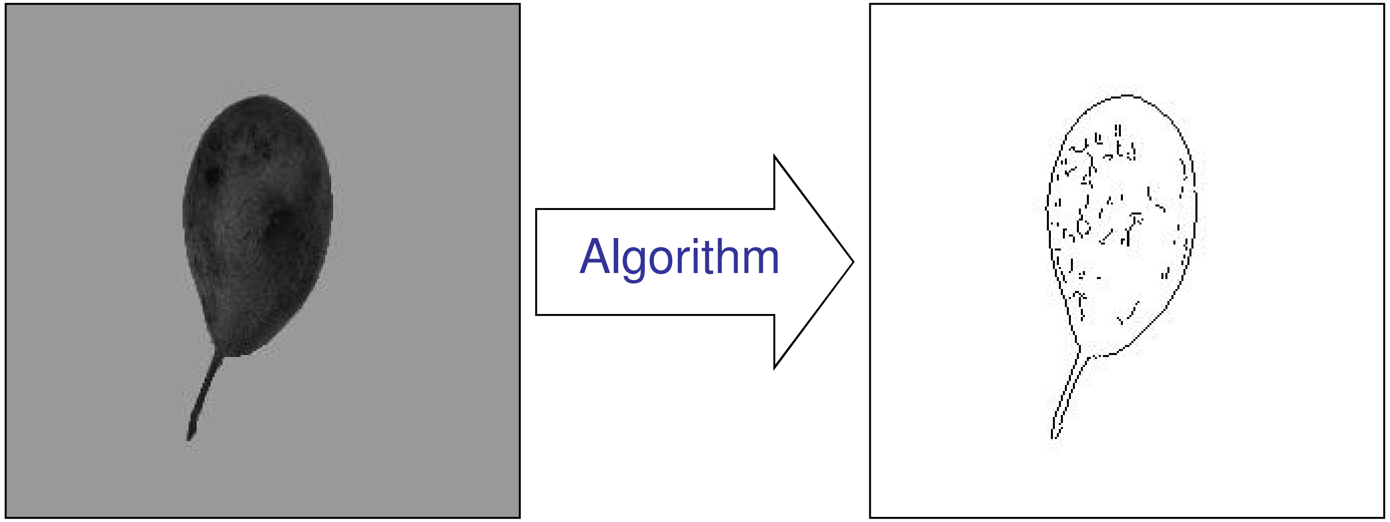


  - Smoothed versions



Introduction: steps in edge detection

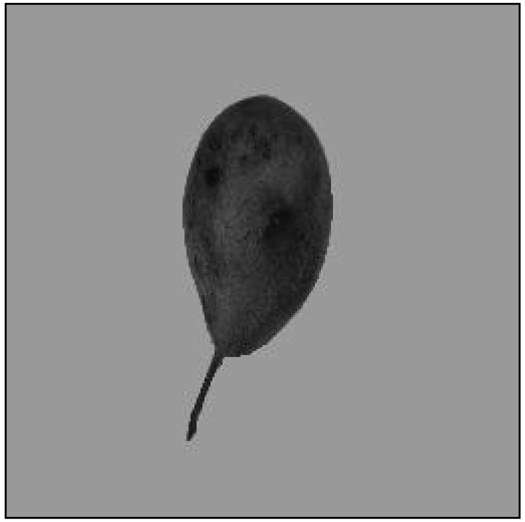
- Edge detector = *Algorithm that produces a set of edges from an image*:
  1. Image filter to enhance intensity ***changes*** +
  2. Decide which ones are either ***edges*** or not.



pear borders

Introduction: steps in edge detection

1. Image filter to enhance intensity ***changes***
   a. Smoothing
   b. Enhancement



pear

Filtering

(step a
+ step b)

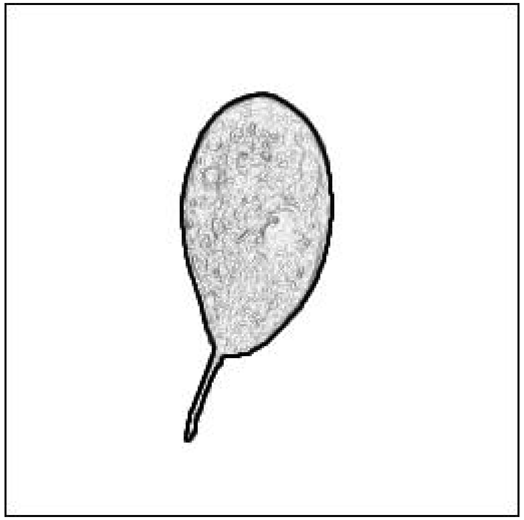


Image intensity changes

Introduction: steps in edge detection

- Locate edges:

2. Detect/locate (decide whether is an edge or not)
    - Errors: *false* edges, *lost* edges.
3. Subpixel precision estimation (optional)

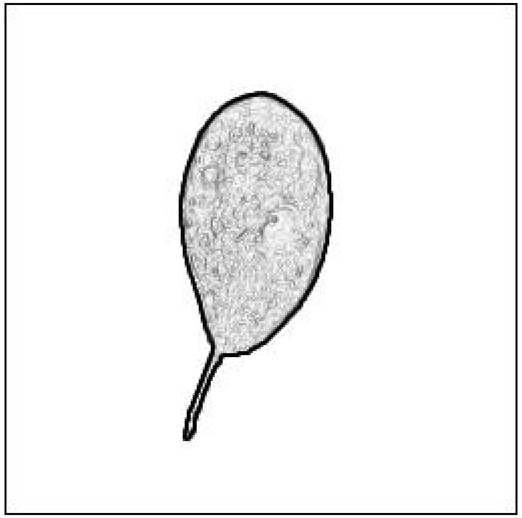
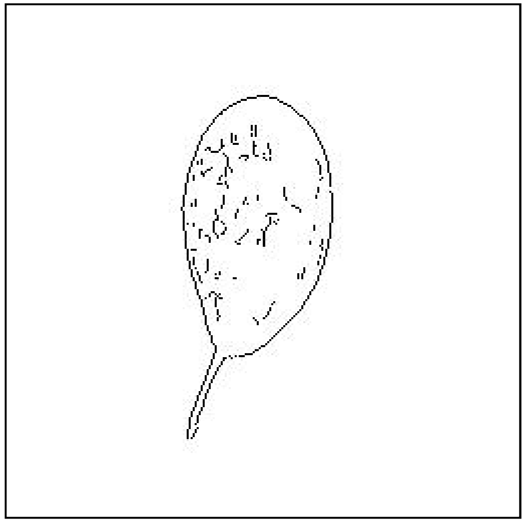


Image intensity changes

Detection

(step 2)



Edges

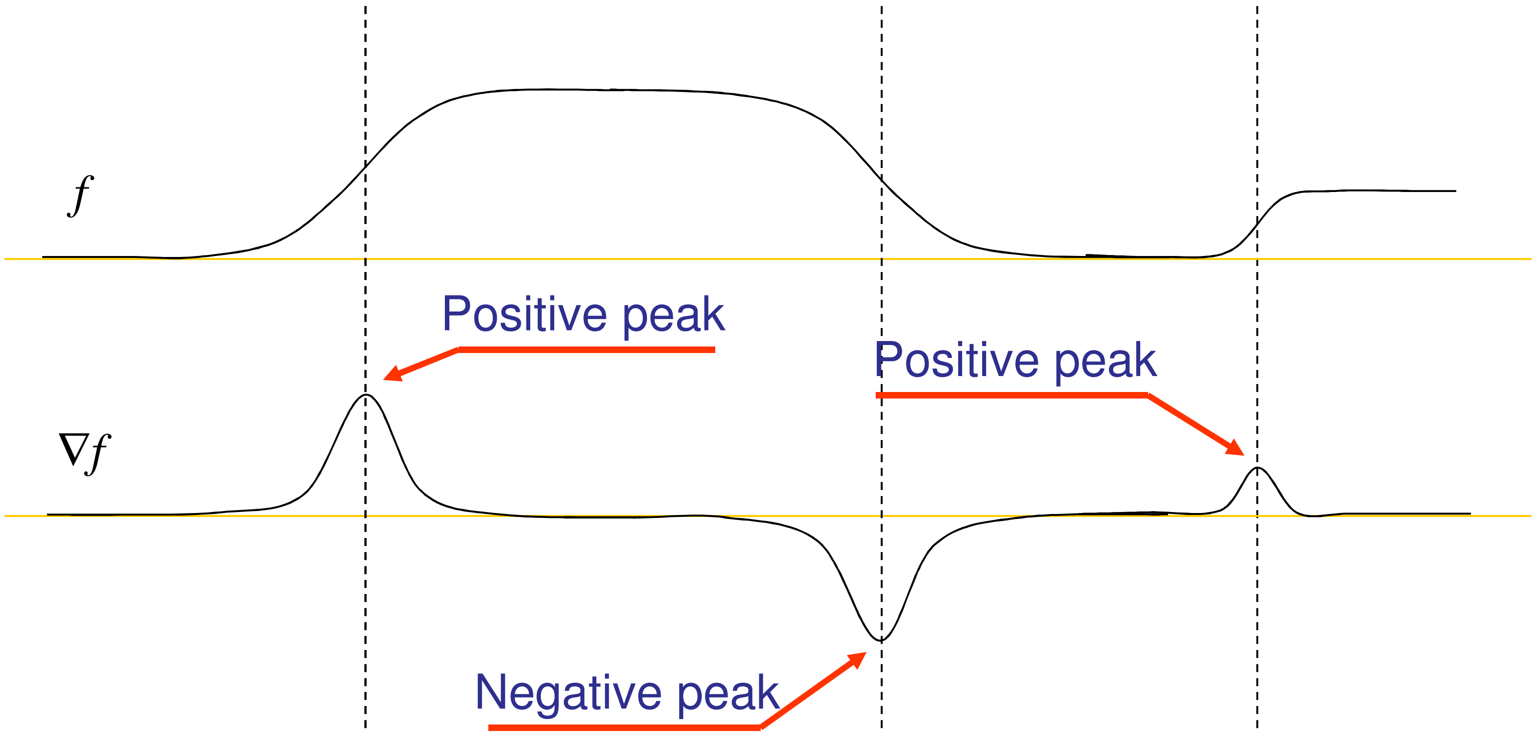
Index

- Introduction
  - What is an edge?
  - Steps in border detection
- **Edge detectors based on image gradient**
  - What is image gradient?
  - Detectors
- Edge detectors based on Laplacian operator
  - What is the Laplacian?
  - Detectors

What is image gradient?

- Gradient: measure local changes
  - $\cong$ Intensity differences

$$\nabla f(x) = \frac{\partial f}{\partial x}$$

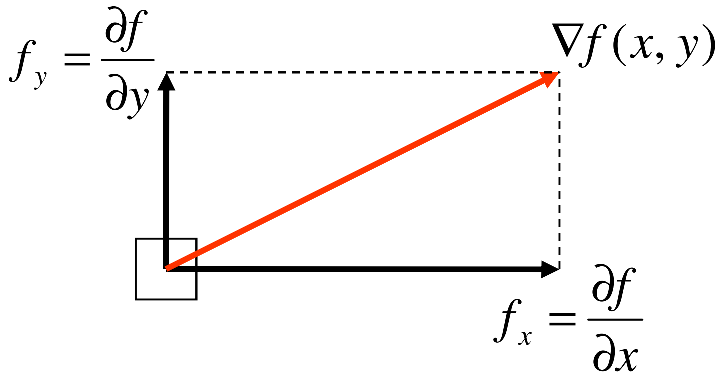


What is image gradient?

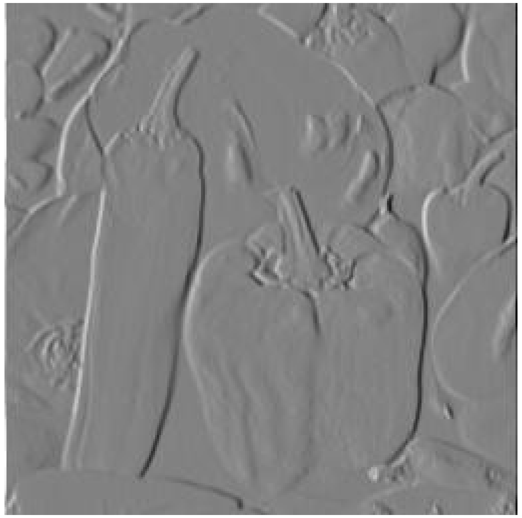
- Gradient =
  1st derivative of f

$$\nabla f(x,y) = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j}$$

- It is a 2 component vector:
  - X gradient, f_x
  - Y gradient, f_y



How to display image gradients



f_x

$|f_x|$

$\overline{|f_x|}$

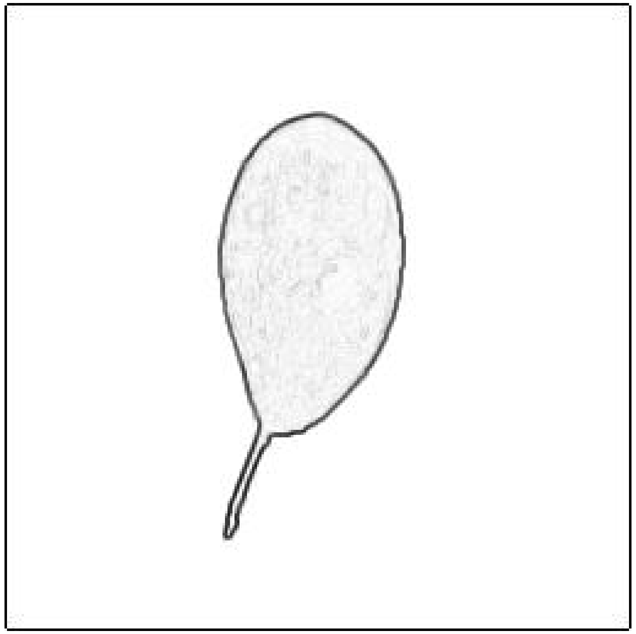
Gradient in X and Y directions

$$|f_x| + |f_y|$$

$$|f_y|$$

$$|f_x|$$

Gradient magnitude and orientation



$$|\nabla f(x, y)| =$$

$$m(x, y) = \sqrt{f_x^2 + f_y^2}$$

$$\phi(x, y) = \arctan \frac{f_y}{f_x}$$

Magnitude = Edge strength

Orientation = Edge direction

Gradient magnitude and orientation

$\left|f_x\right| + \left|f_y\right|$

ϕ

Approx.

m

Gradient filters

Enhancement operators → Sum of coefficients equal to 0

The simplest gradient operator

$$\begin{bmatrix}-1 & 1\end{bmatrix} \Rightarrow \begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}-1 \\ 1\end{bmatrix} \Rightarrow \begin{bmatrix}-1 \\ 0 \\ 1\end{bmatrix}$$

Noise sensitive

Prewitt operator

$$\begin{bmatrix}-1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 \\ 1 \\ 1\end{bmatrix} \cdot \begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}-1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1\end{bmatrix} = \begin{bmatrix}-1 \\ 0 \\ 1\end{bmatrix} \cdot \begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

Smoothing + Enhancement

Gradient filters

Sobel operator

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Emphasize pixels nearer to the center

Isotropic or Frei-Chen operator

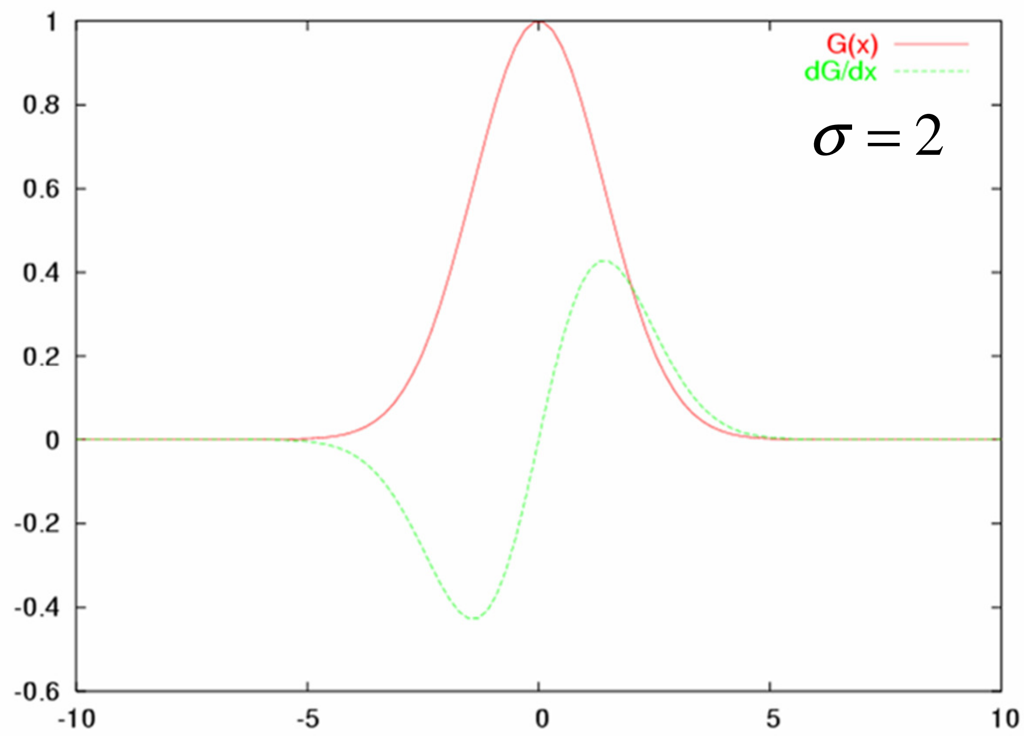
$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

Exercise: what is the decomposition?

Gradient filters

$$G(x) = e^{\frac{-x^2}{2\sigma^2}} \Rightarrow \frac{dG}{dx} = \frac{-x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}$$

Derivative of a Gaussian

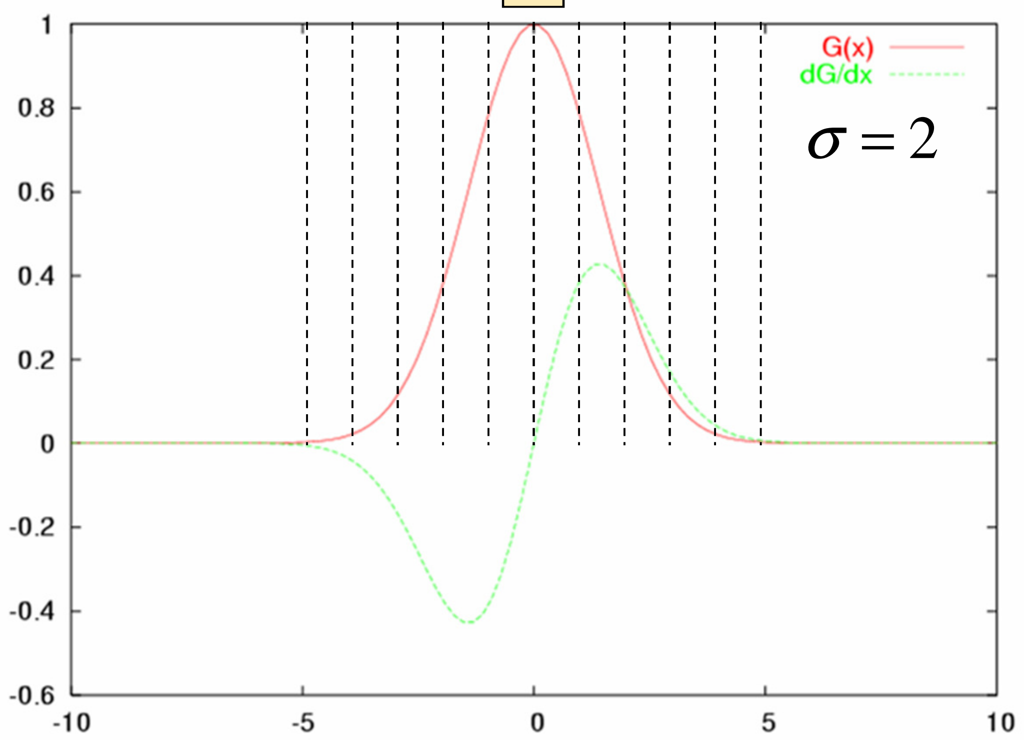


Gradient filters

$$G = [\ .02 \quad .11 \quad .37 \quad .78 \quad 1 \quad .78 \quad .37 \quad .11 \quad .02]$$

$$\frac{dG}{dx} = [-.03 \quad -.16 \quad -.18 \quad -.39 \quad 0 \quad .39 \quad .18 \quad .16 \quad .03]$$

Derivative of a Gaussian

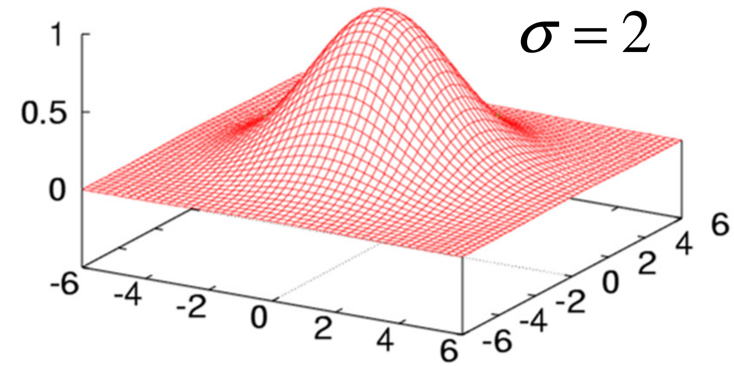


Gradient filters

Derivative of a Gaussian

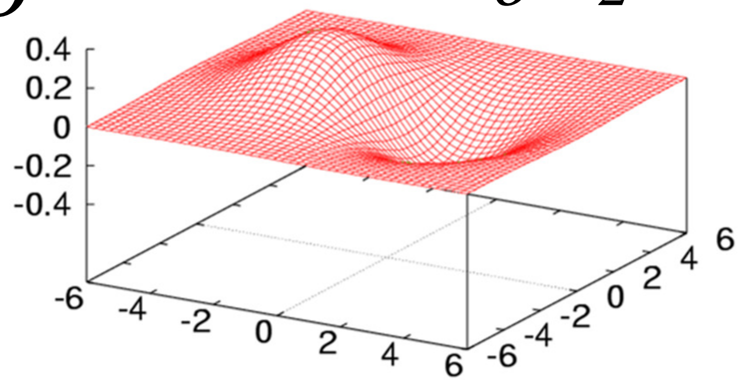
$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$\sigma = 2$



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

$\sigma = 2$



Gradient filters

Roberts' operator

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Kirsch's masks (or compass masks)

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

0º 45º 90º 135º

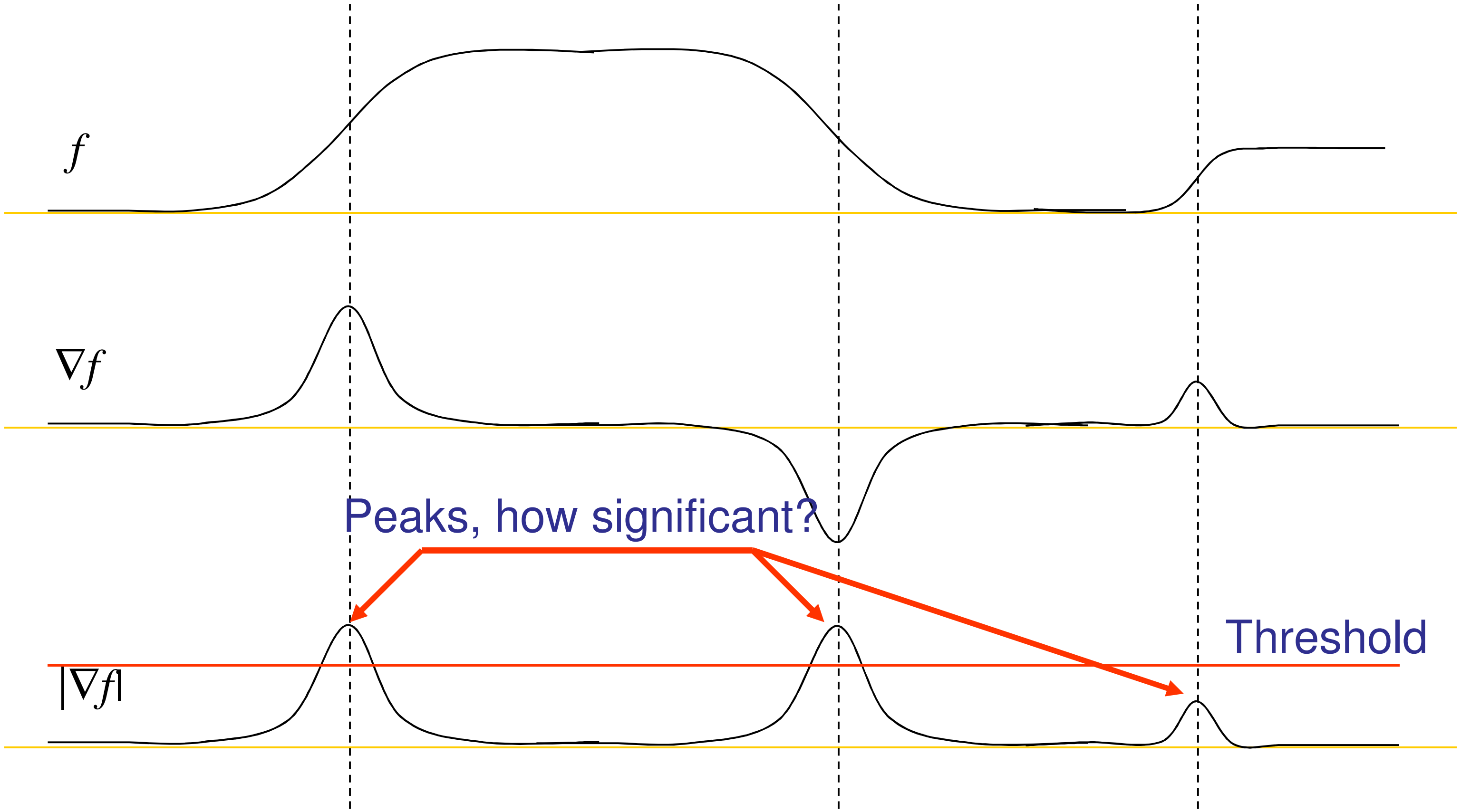
$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

180º 225º 270º 315º

Choose the máximum and associated direction

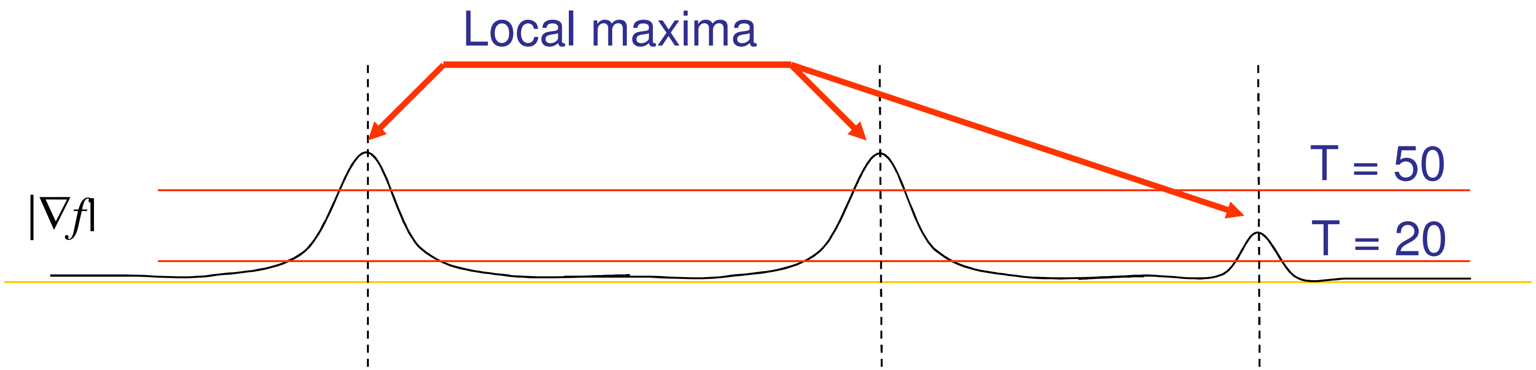
Edge detectors based on gradient

- Gradient magnitude ≅ Amount of change
  - Significant changes ≅ Magnitude significant peaks



Edge detectors based on gradient

- **Binarize** gradient magnitude
  - Pixels with value > T → borders

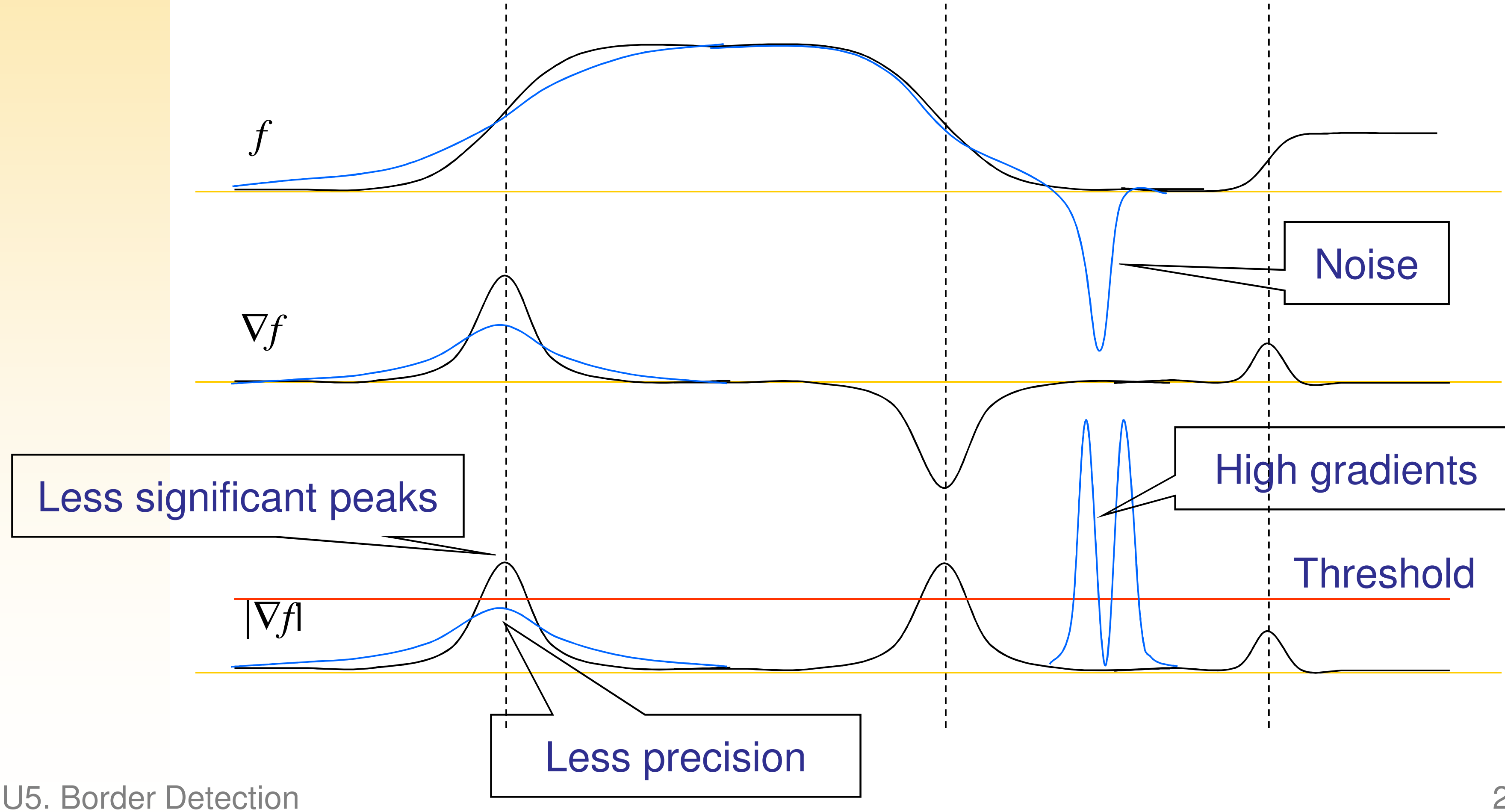


- Problem 1: Border width > 1 pixel
- Solution: Identify as border only 1 pixel per peak.

- Detection = **Significant local maxima of gradient magnitude**

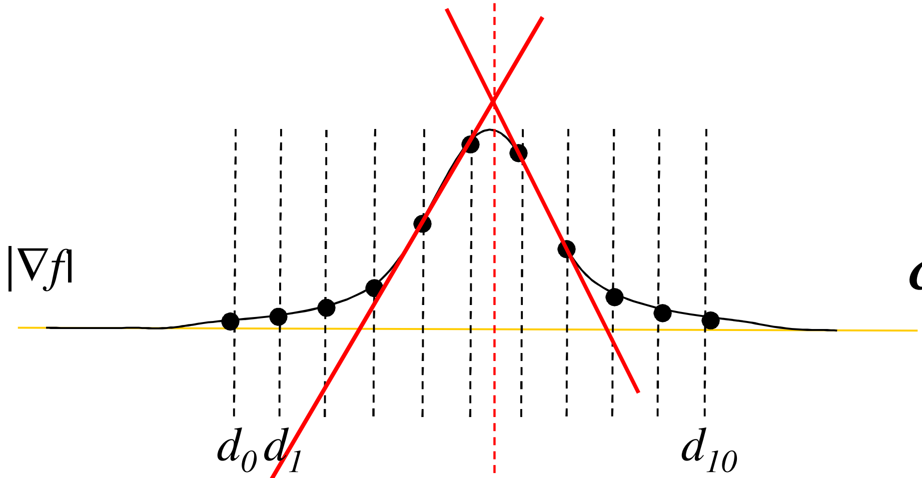
Edge detectors based on gradient

- Problem 2: Smoothed edges → lost / imprecise
- Problem 3: Noise → false edges



Edge detectors based on gradient

1. Smoothing
2. Enhancement: **Gradient**
3. Detection:
   - Significant **local maxima** of gradient **magnitude**
4. Subpixel precision estimation (optional)
   - **Weighted sum**



$$\delta d = \frac{\sum_{i=1}^{n} m_i d_i}{\sum_{i=1}^{n} m_i}$$

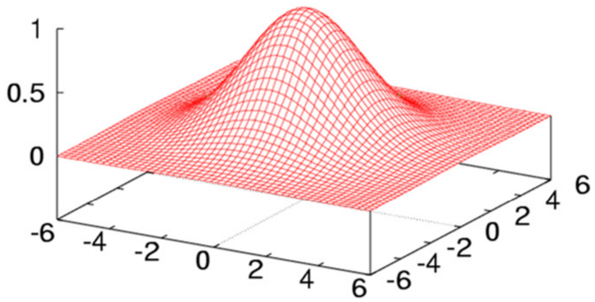
Position

Gradient magnitude

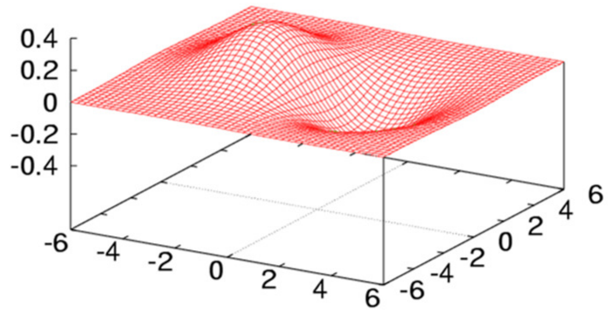
   - Approximate to a Gaussian
   - Interpolation

"Canny" edge detector

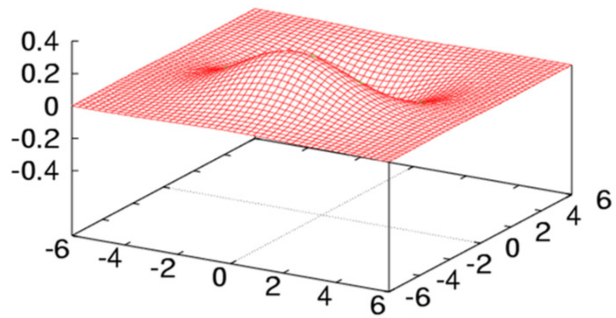
- Conflict between:
  - noise reduction and
  - border location.
- Distrubution that optimize both problems → **Gaussian** (Canny, 1986)



2D Gaussian, σ =2



1st derivatives of a 2D Gaussian, σ =2

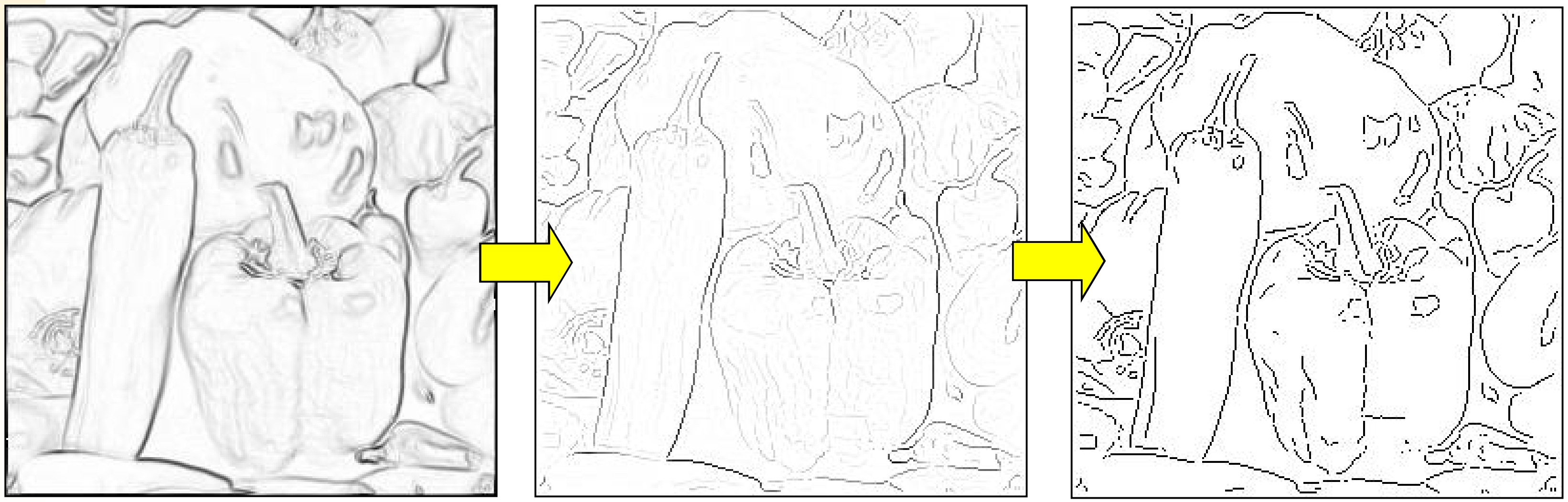


"Canny" edge detector

1. Smoothing
2. Enhancement

1st derivative of a Gaussian

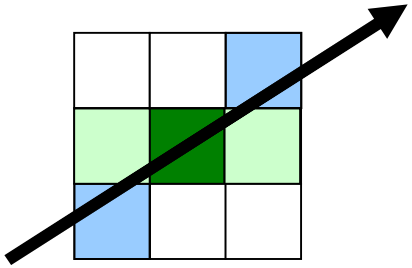
3. Detection: significant local maxima ...
   - Non-maximal suppression
   - Hysteresis threshold (double threshold)
4. Subpixel precision estimation (optional)

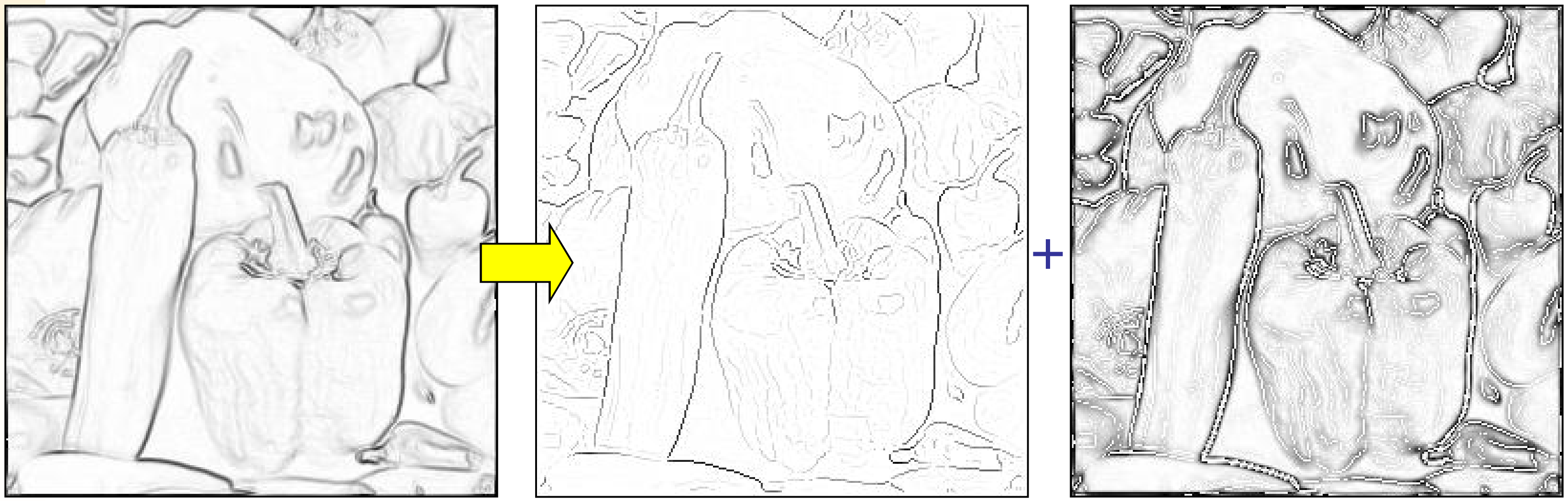


Only local maxima   Hysteresis threshold

"Canny" edge detector

- Non-maximal suppression:
  - Set to 0 gradient magnitude values which are not **local maxima** in gradient direction

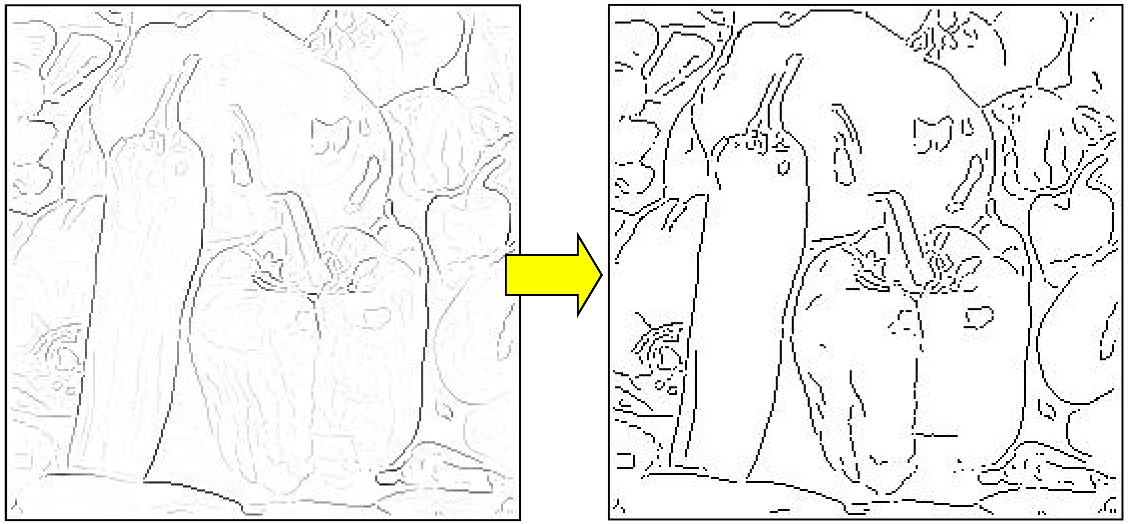




Only local maxima   Non local maxima

"Canny" edge detector

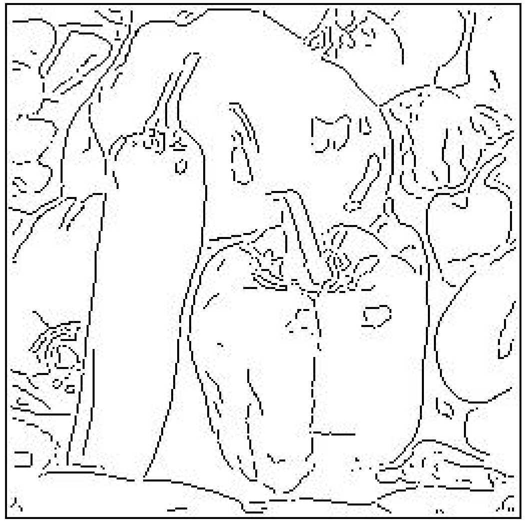
- Hysteresis threshold (double threshold):
  - Two thresholds: T1 < T2.
  - Gradient magnitude > T2 → edges
  - Gradient magnitude < T1 → no edges
  - Values in between, only if the are connected to some border → edges (iterative algorithm)



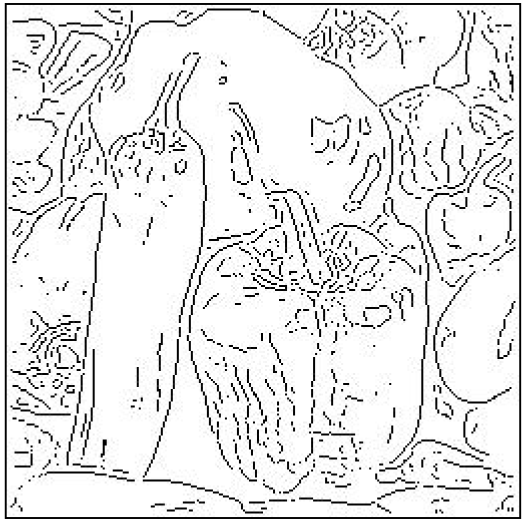
Local maxima — Significant local maxima

"Canny" edge detector

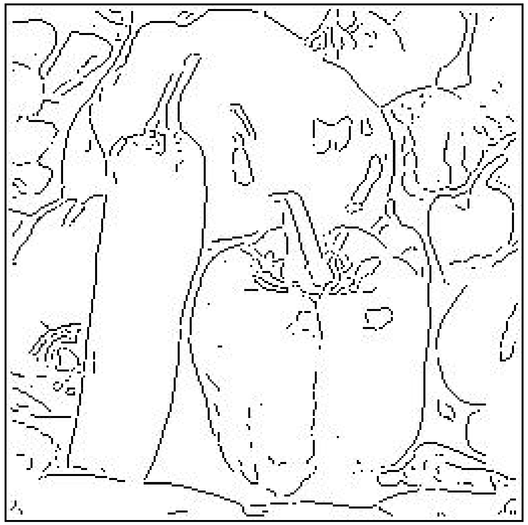
Magnitude (local maxima)



T1 = 30, T2 = 60



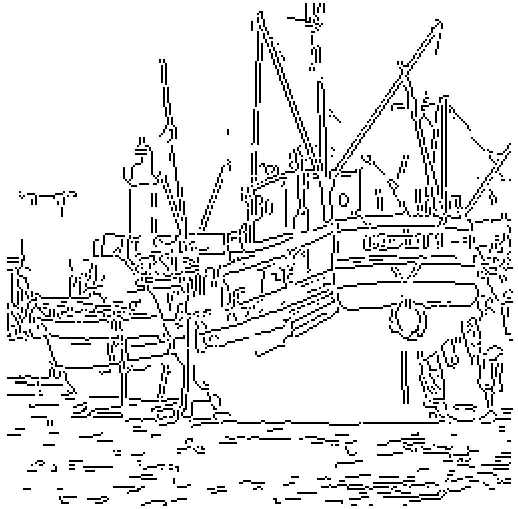
T1 = 30



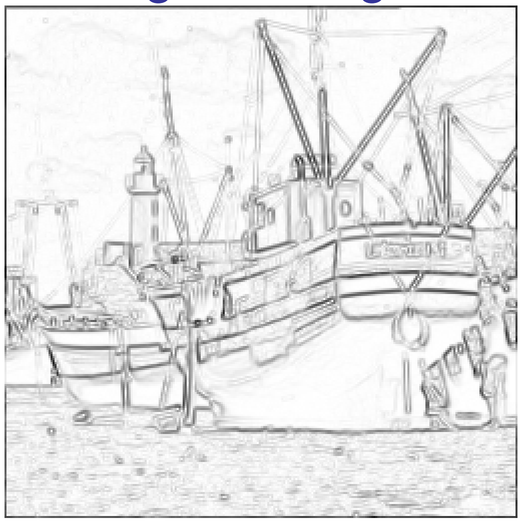
T2 = 60

“Canny” edge detector

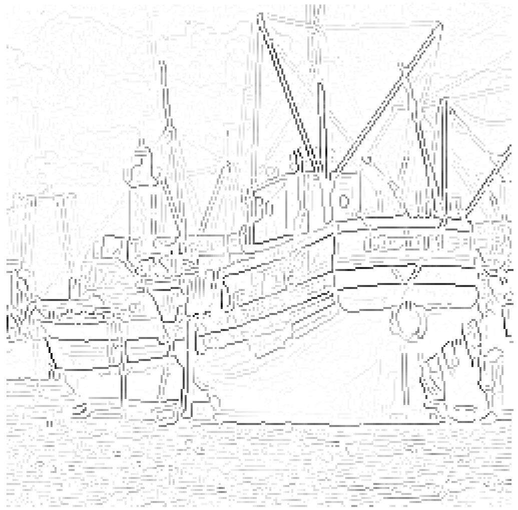
Original image



T1 = 80, T2 = 160

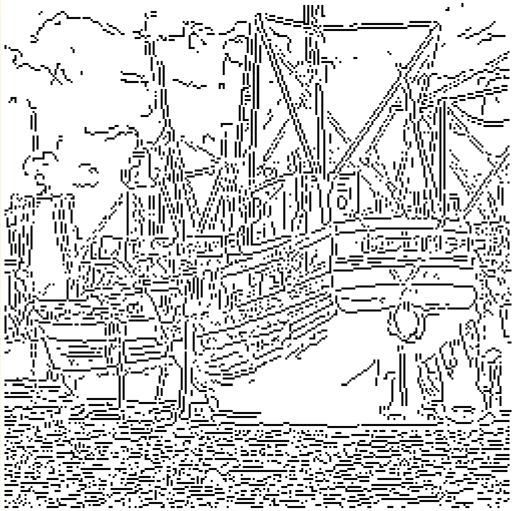


Gradient magnitude

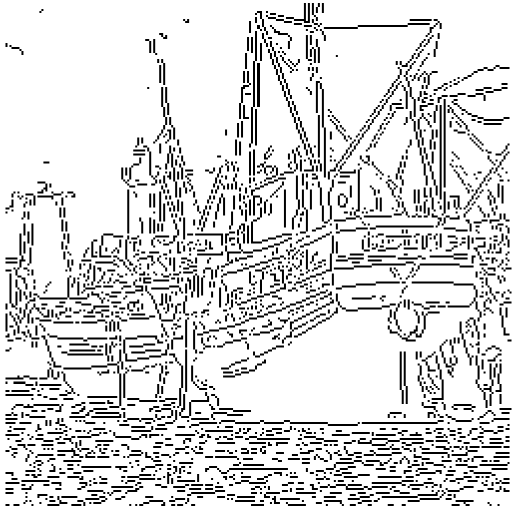


Local maxima

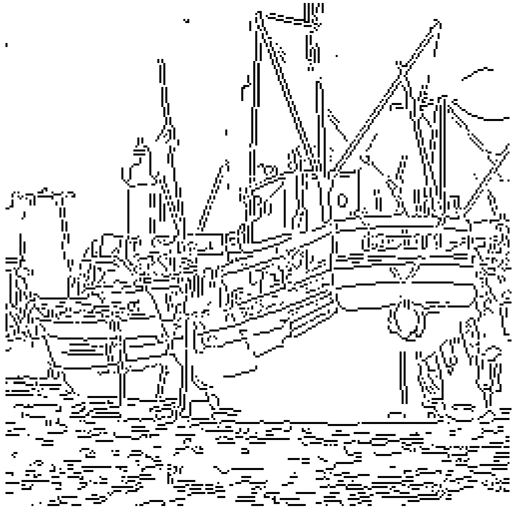
“Canny” edge detector



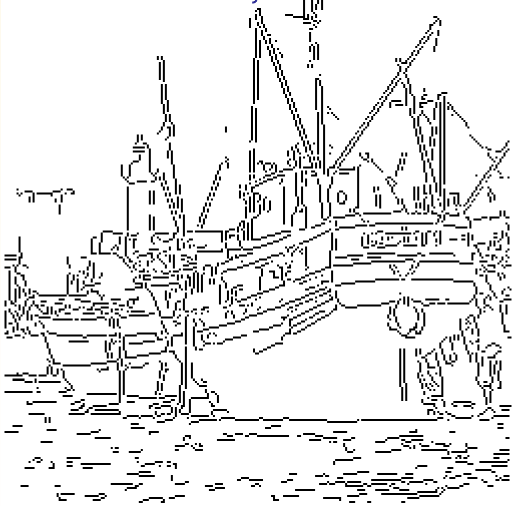
T1 = 20, T2 = 40



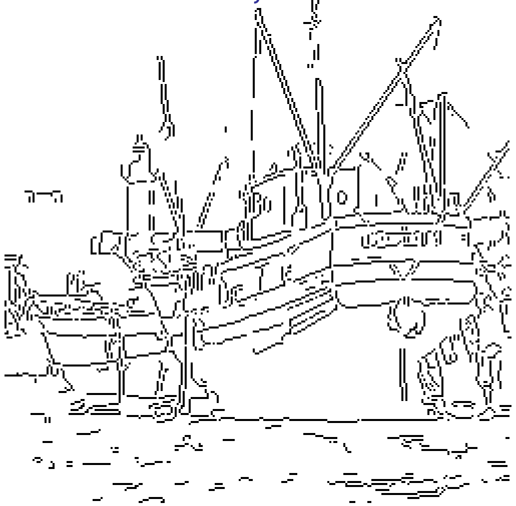
T1 = 40, T2 = 80



T1 = 60, T2 = 120



T1 = 80, T2 = 160



T1 = 100, T2 = 200

T1 = 120, T2 = 240

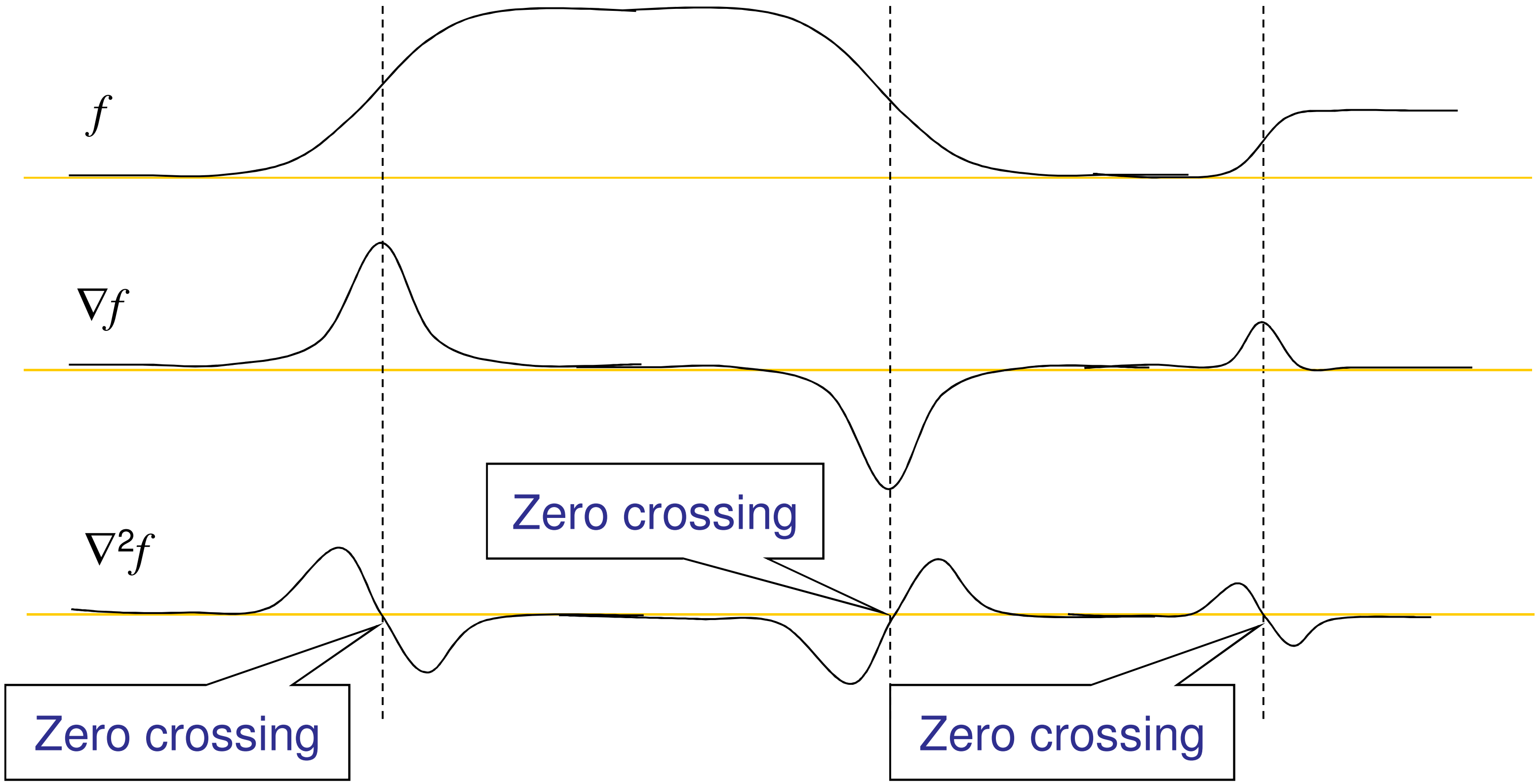
Index

- Introduction
  - What is an edge?
  - Steps in edge detection
- Edge detectors based on image gradient
  - What is image gradient?
  - Detectors
- **Edge detectors based on Laplacian operator**
  - What is the Laplacian?
  - Detectors

What is the Laplace operator?

- Laplacian: measures changes in gradient
  - ≅ Gradient differences

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



The Laplace operator

- Laplacian = 2nd derivative of *f*

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

- Simple example:
  - Gradient of *f* ⇨ $f_x(x) = f(x) - f(x\text{-}1)$
  - Laplacian of *f* ⇨ $f_{xx}(x) = f_x(x) - f_x(x\text{-}1)$

$$f_{xx}(x) = f(x) - f(x\text{-}1) - f(x\text{-}1) + f(x\text{-}2)$$

$$f_{xx}(x) = f(x) - 2\,f(x\text{-}1) + f(x\text{-}2)$$

- Convolution: $\begin{bmatrix}1 & -2 & 1\end{bmatrix}$ ⇨ $\begin{bmatrix}-1 & 2 & -1\end{bmatrix}$

Laplace

Inverse Laplace

Laplacian filters

- Convolutiion: $\begin{bmatrix}1 & -2 & 1\end{bmatrix} \Rightarrow \begin{bmatrix}-1 & 2 & -1\end{bmatrix}$

  Laplace → Inverse Laplace

Laplace operators (invers)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

More weight to central pixels:

$$\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix}$$

Noise sensitve

Laplacian + original image

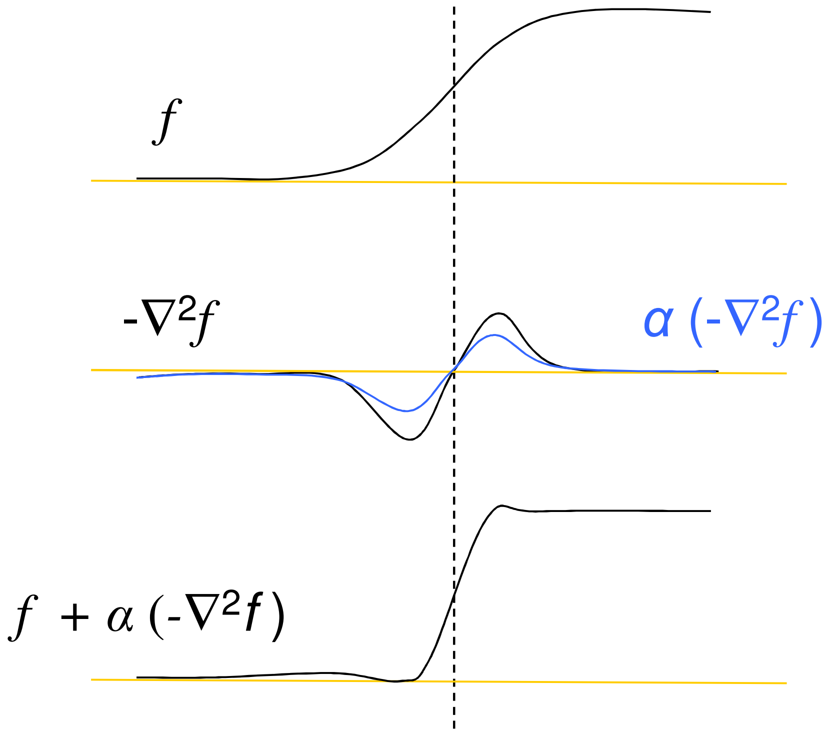
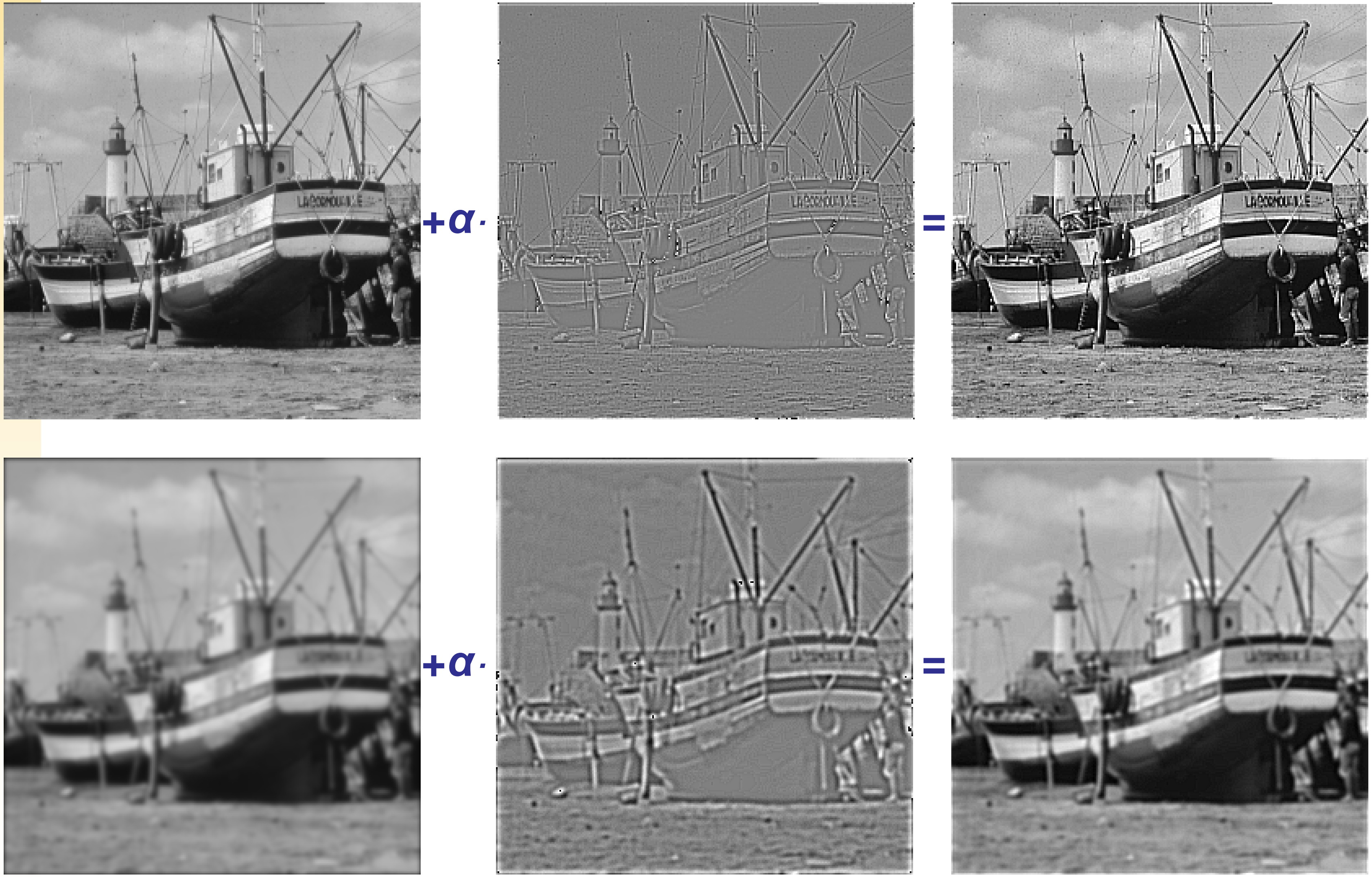


Image enhancement
(sharpening)

Laplacian + original image



LoG filter: Laplacian of a Gaussian

- 2nd derivative of a Gaussian (Marr-Hildreth, 1980)
  - Laplacian of a Gaussian (LoG)
  - Marr-Hildreth operator
  - “Mexican hat” operator

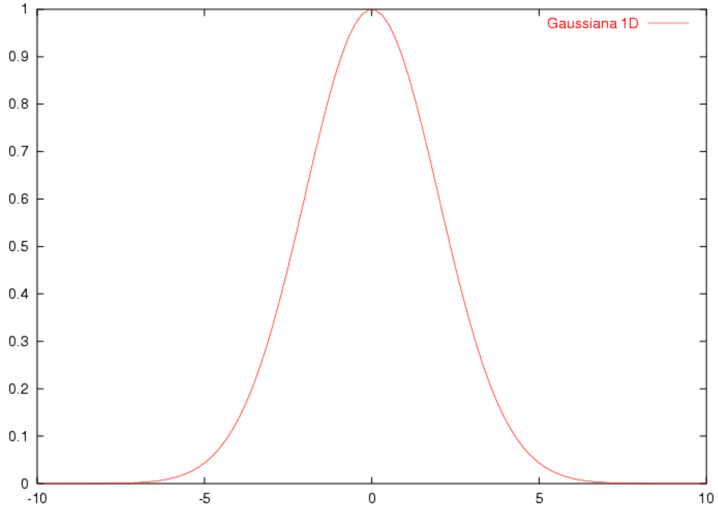
$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\Downarrow$$

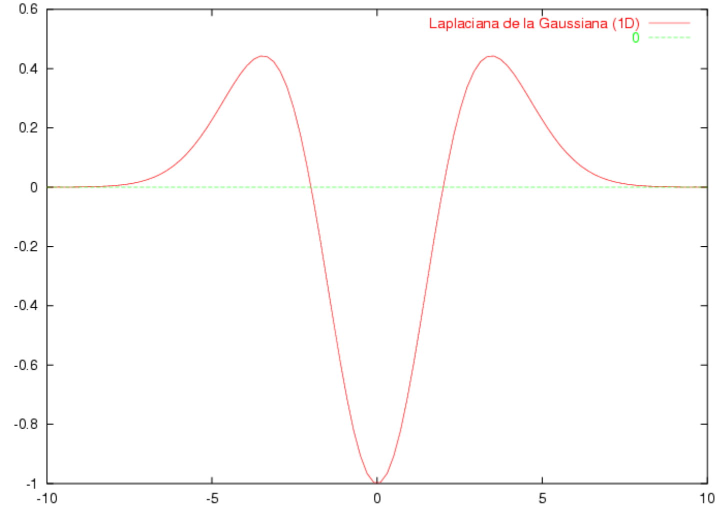
$$\nabla^2 G = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

LoG filter: Laplacian of a Gaussian

- 2nd derivative of a Gaussian (Marr-Hildreth, 1980)
  - Laplacian of a Gaussian (LoG)
  - Marr-Hildreth operator
  - “Mexican hat” operator



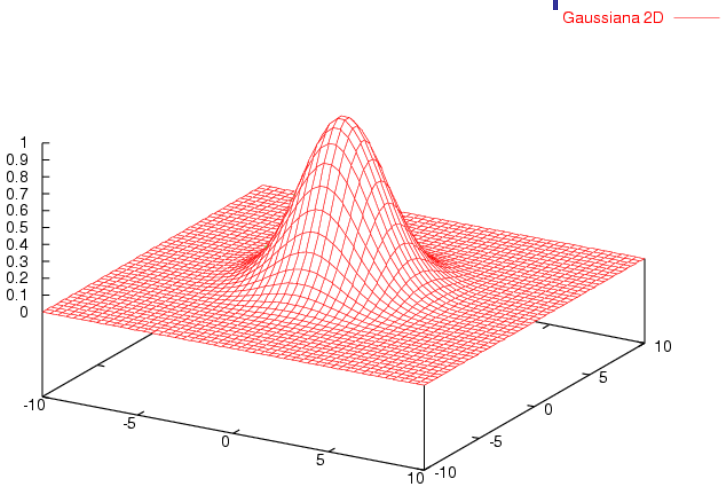
1D Gaussian



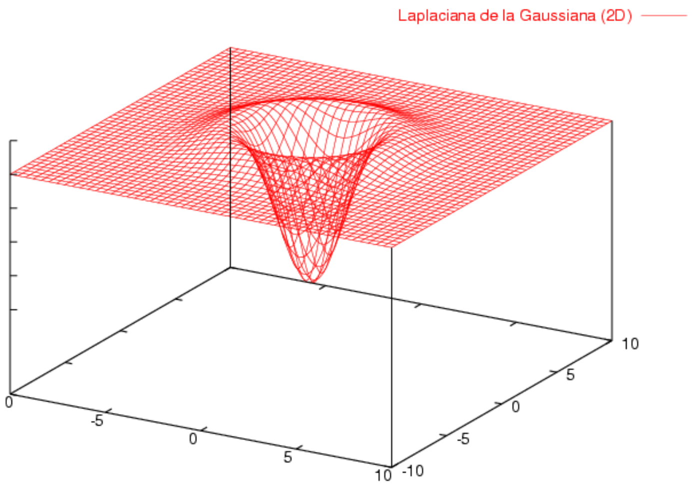
1D LoG

LoG filter: Laplacian of a Gaussian

- 2nd derivative of a Gaussian (Marr-Hildreth, 1980)
  - Laplacian of a Gaussian (LoG)
  - Marr-Hildreth operator
  - “Mexican hat” operator



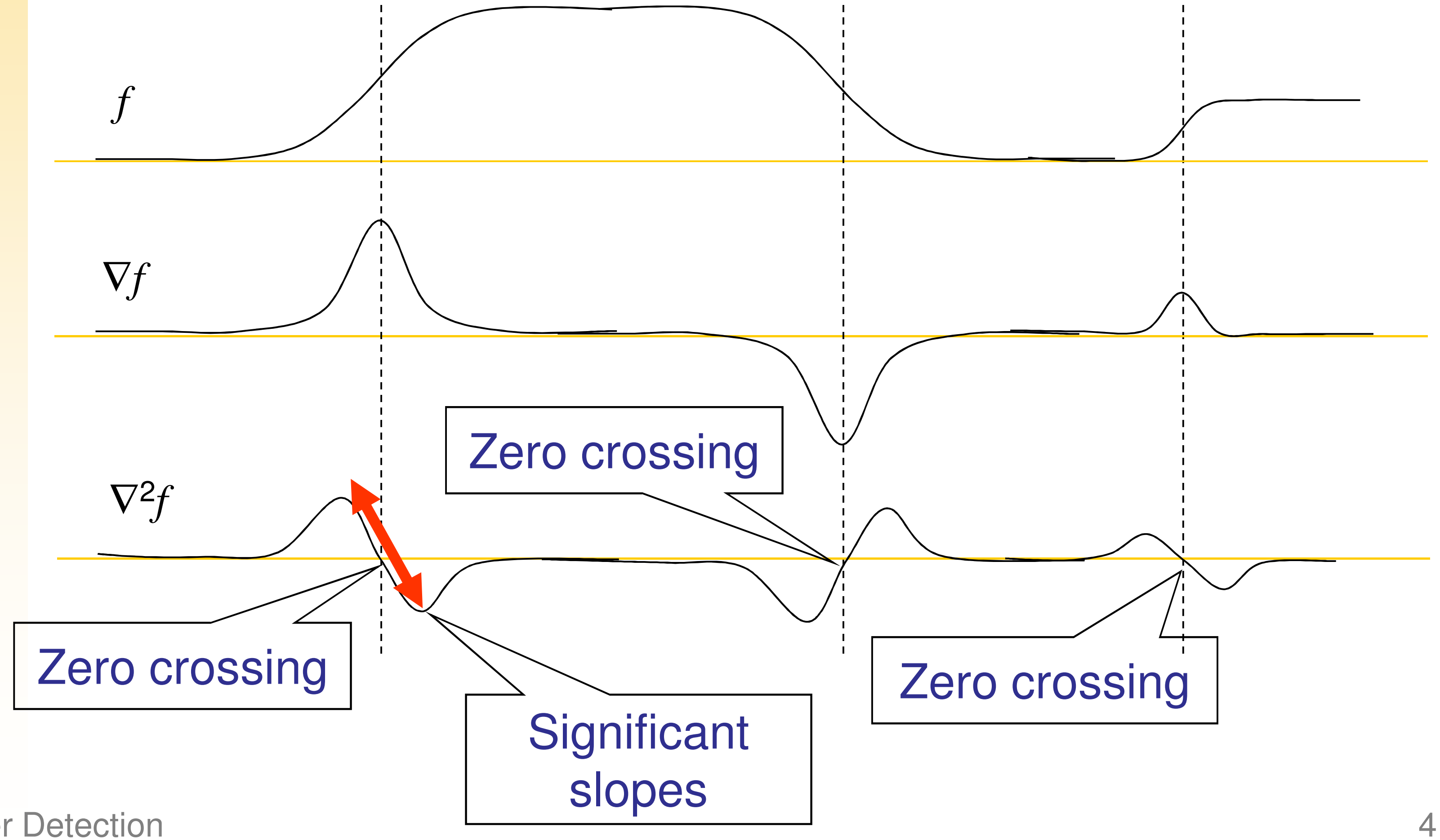
2D Gaussian



2D LoG

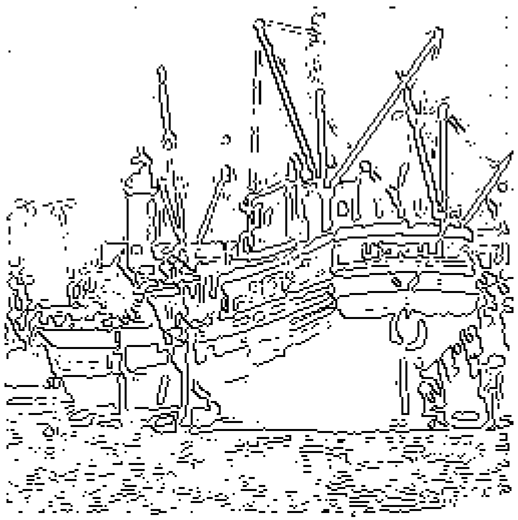
Laplacian-based edge detectors

- Laplacian ≅ Gradient differences
  - Detect ”zero crossings”

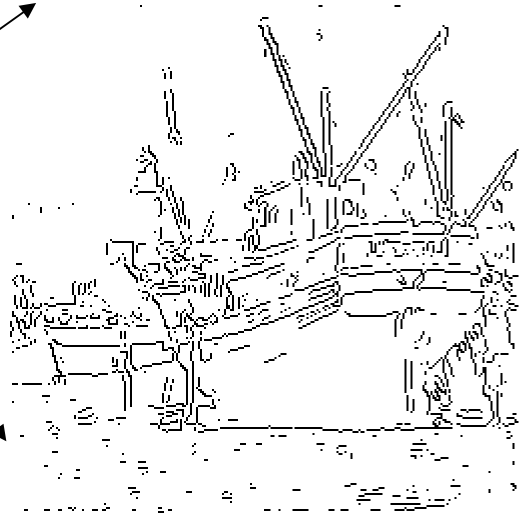


Laplacian-based edge detectors

Zero crossing slope
↓
Change strength



Different thresholds

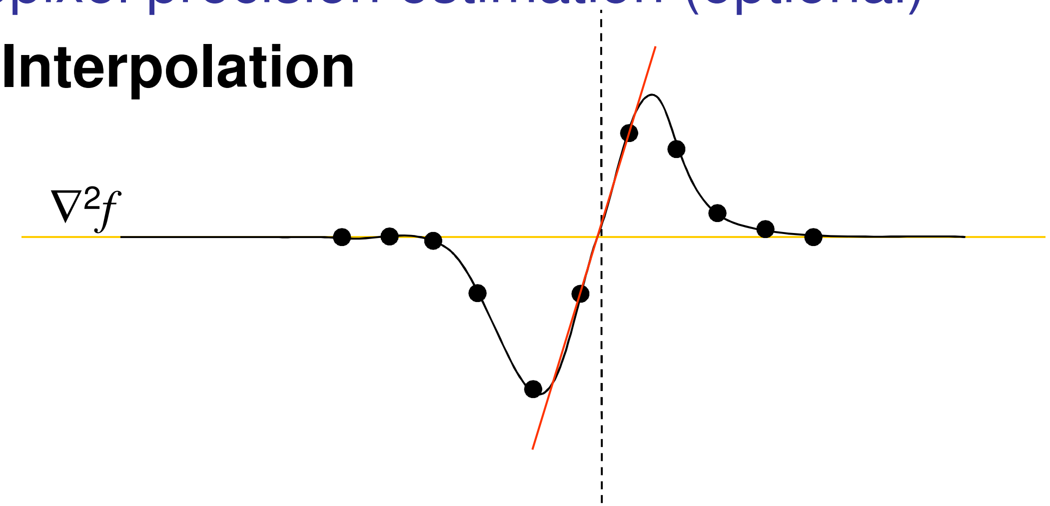


Laplacian-based edge detectors

1. Smooting
2. Enhancement: **Laplacian**

} **LoG**

3. Detection:
   - Significant **zero crossings** (significant peaks in the 1st derivtaive)
4. Subpixel precision estimation (optional)
   - **Interpolation**

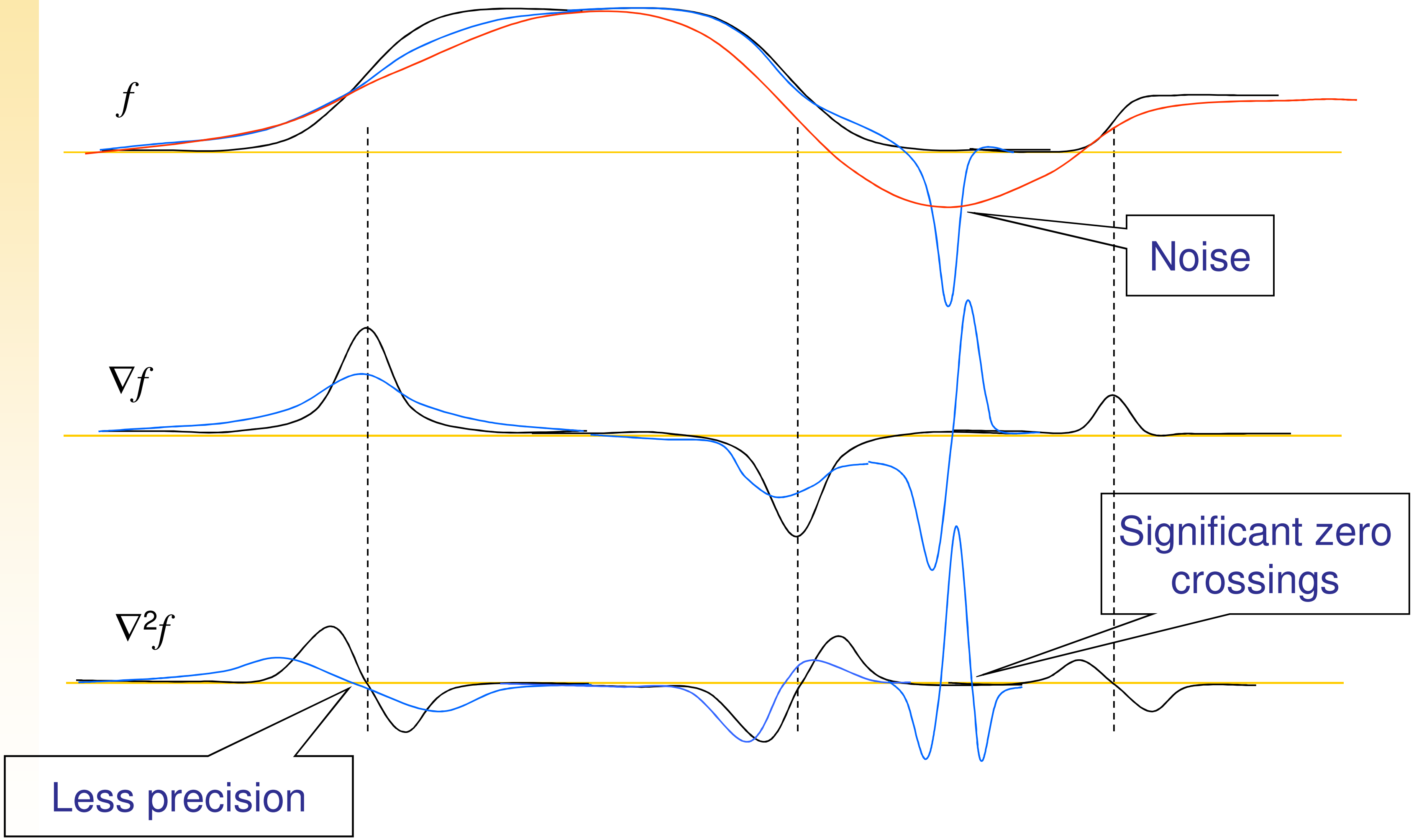


Too noisy → not used in practice

Scale space

- Slope of the zero crossing related to the change strength
  - More smoothing → less slope
- **σ controls the amount of smoothing**
  - σ small →
    - More sensitive to noise / More false edges
    - More precision to locate edges
  - σ large →
    - More lost edges / Found edges are robust
    - Less precision (edge shift) / Nearby edges can be merged

Scale space

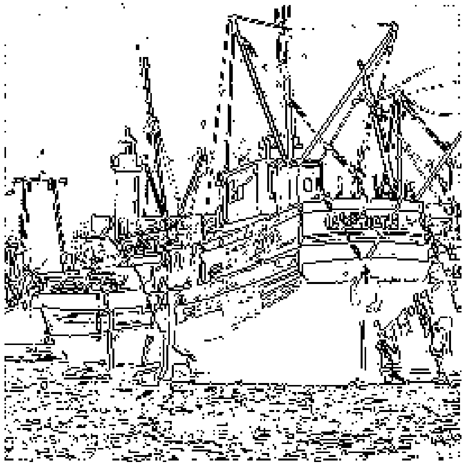


f
Noise
∇f
Significant zero crossings
$\nabla^2 f$
Less precision

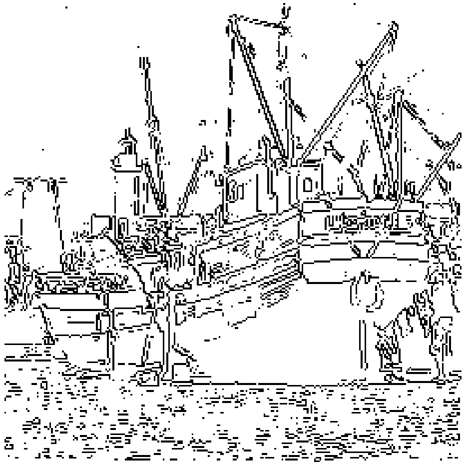
Scale space

- Solution:
  - Filter with masks of different σ
  - Analyze edges behabiour to different filtering scales (σ is related to the image scale)
    - σ large → **robust** edges,
      but **shifted**
    - σ small → better **localization**

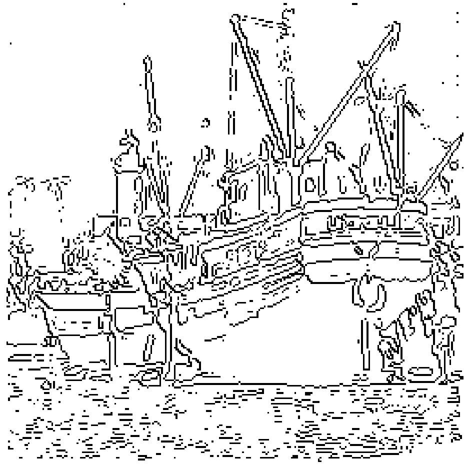
Scale space



$\sigma = 0.6$

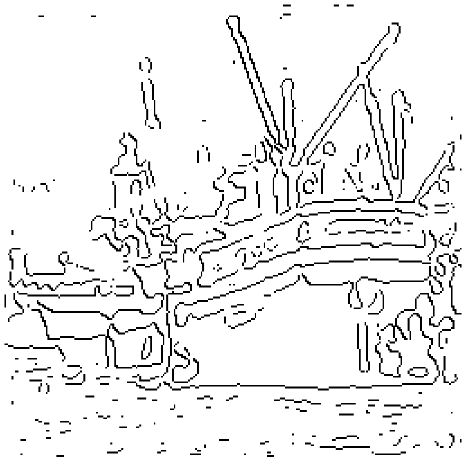


$\sigma = 1$



$\sigma = 2$

$\sigma = 4$



$\sigma = 6$