ensia
The National Higher School of
Artificial Intelligence
المدرسة الوطنية العليا للذكاء الاصطناعي

Course
« *Computer Vision*»

Sid-Ahmed Berrani

2024-2025

Image Features

What is a feature?

- Local, meaningful, detectable parts of the image.

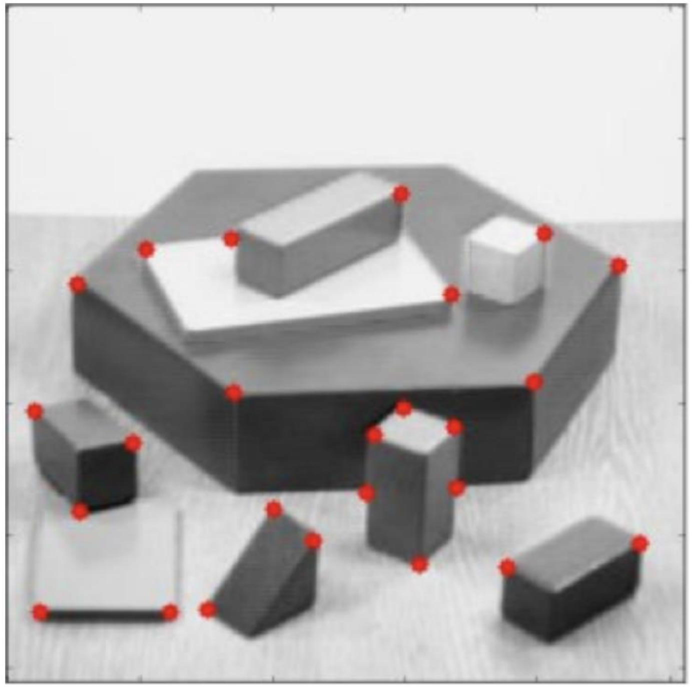


Image Features in Computer Vision

What is a feature?

- Location of sudden change

Why use features?

- Information content is high
- Invariant to change of viewpoint or illumination
- Reduces computational burden

Image Features in Computer Vision

A good feature is **invariant** to:

- Viewpoint
- Lighting conditions
- Object deformations
- Partial occlusions

and should be:

- Unique
- Easy to be found and extracted

Image Features: Edges

Edge pixels are pixels at which the intensity of an image function changes abruptly.

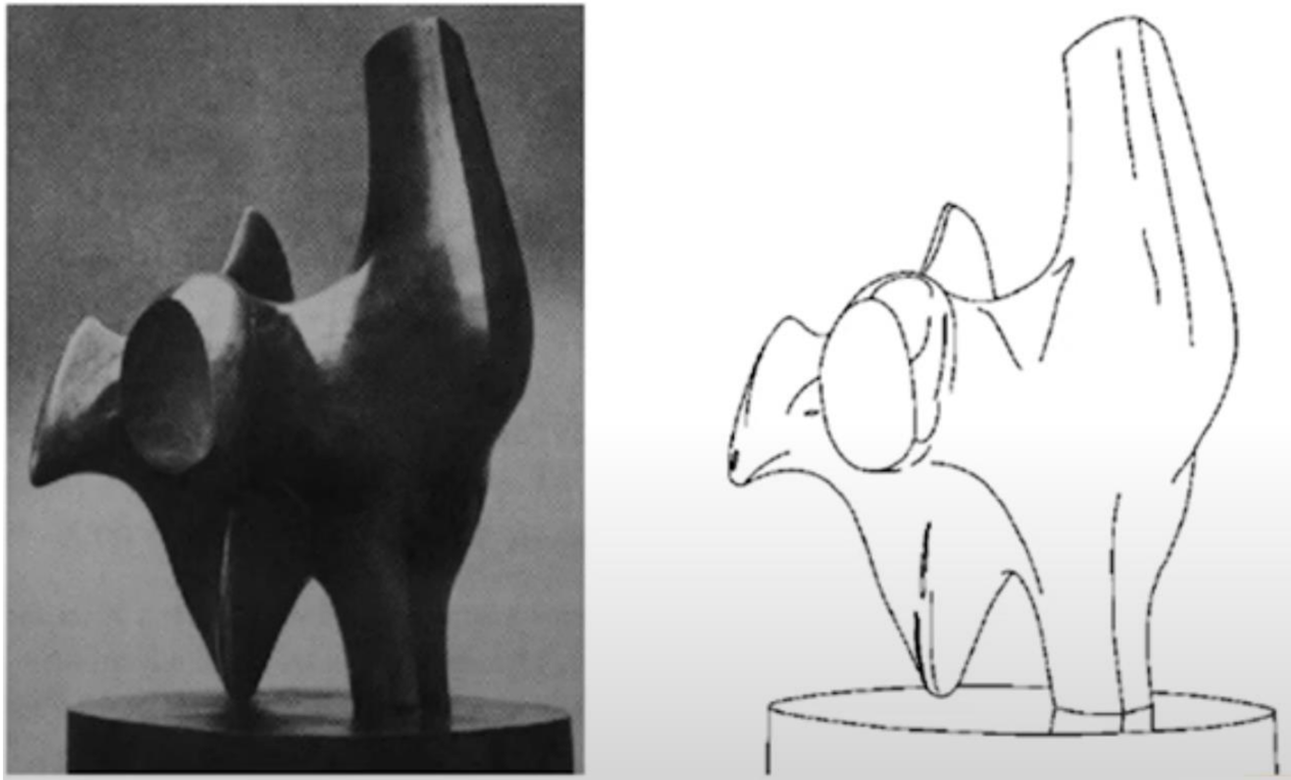
They represent a simple and meaningful type of image feature…

why?

- Edges in an image have many interesting causes
- Looking at edges reduces information required – Look at a few pixels in a binary image as opposed to all pixels in a grayscale image
- Biological plausibility – Initial stages of mammalian vision systems involve detection of edges and local features

Image Features: Edges

A sculpture *A sketch made by an artist*



Henry Moore sculpture – 1964.
https://www.wikiart.org/fr/henry-moore/working-model-for-three-way-piece-no-2-archer-1964

With a few strokes, the artist was able to convey a lot of information about the sculpture: The three-dimensional structure of the sculpture, some of the lighting effects like shading and highlights…

Image Features: Focus on Edge Detection

Objective:

Convert a 2D image into a set of points where intensity changes rapidly.

Topics:

1. What is an edge?
2. Edge detection using gradients.
3. Edge detection using Laplacian.
4. Canny edge detector.
5. Line/curve boundary detection.
6. Hough transform.
7. Corner detection (the Haris detector).

Image Features: Edges

- What causes an edge?

Depth discontinuity

Image Features: Edges

- What causes an edge?

Surface orientation discontinuity

Image Features: Edges

- What causes an edge?

Reflectance discontinuity
(i.e. change of material)

Image Features: Edges

- What causes an edge?

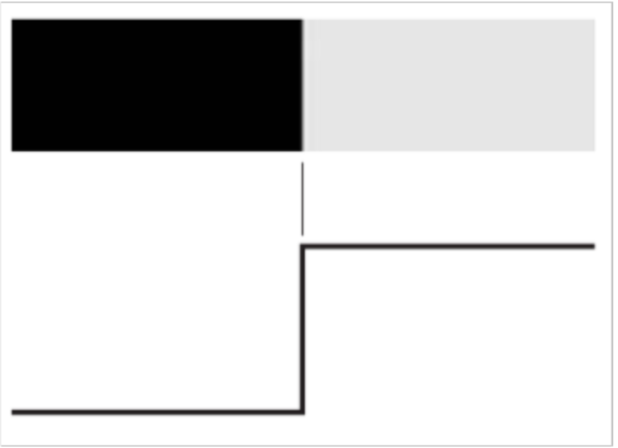
Surface color discontinuity

Edge detection: Introduction

- Find image pixels that present abrupt (local) changes in intensity.

Different edge models: Step Edges

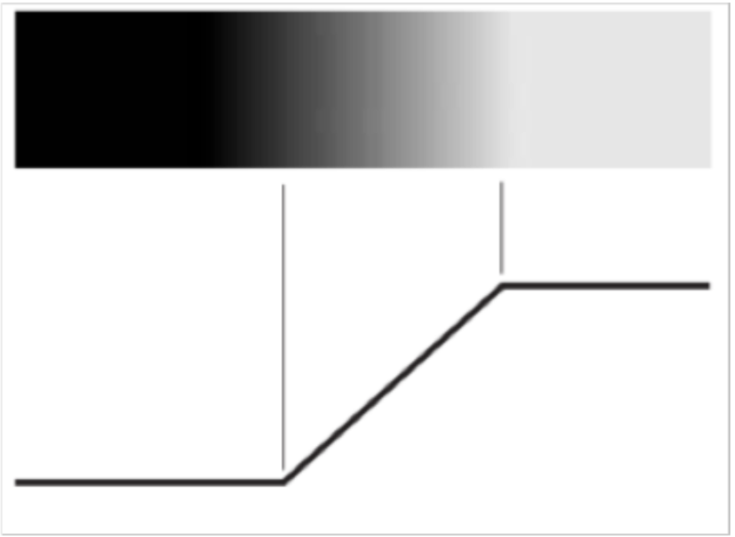
- **Description:** The intensity abruptly changes from one level to another across a boundary.
- **Model:** Ideal for modeling sharp, well-defined edges.
- **Mathematical shape:** A step function (Heaviside-like).
- **Examples:** Boundary between a black object and a white background.



Edge detection: Introduction

Different edge models: Ramp edges

- **Description:** The intensity change is gradual over a few pixels.
- **Model:** Represents real-world edges affected by blurring or gradual transitions.
- **Mathematical shape:** A smooth transition; often modeled as a linear or sigmoid ramp.
- **Examples:** Shadow boundaries or edges in out-of-focus regions.

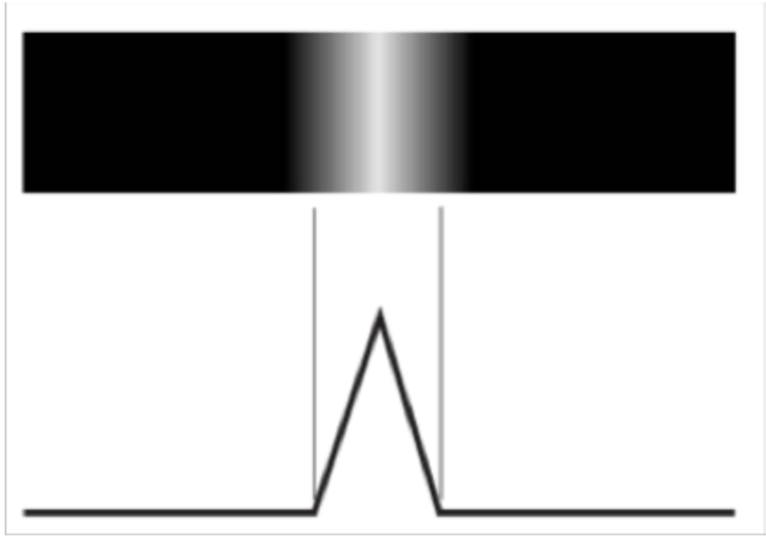


The slope of the ramp is inversely proportional to the degree of blurring in the edge.

Edge detection: Introduction

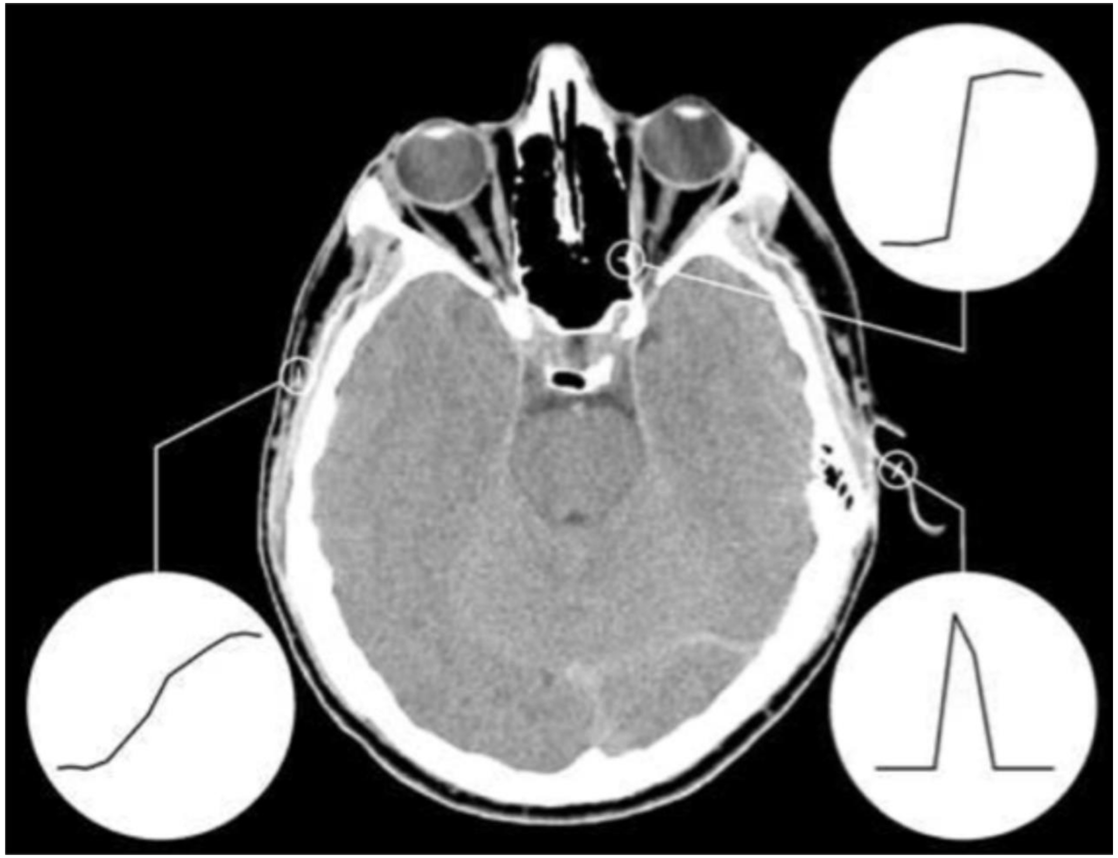
Different edge models: Roof edges

- **Description:** Intensity increases to a peak and then decreases symmetrically.
- **Model:** Represents edges formed by narrow ridges or lines.
- **Mathematical shape:** A peaked or triangular profile.
- **Examples:** Thin structures like wires, roads in aerial images, or fine hair.



Edge detection: Introduction

Edge models: Noise result in deviations from the ideal shapes.



Edge detection: Introduction

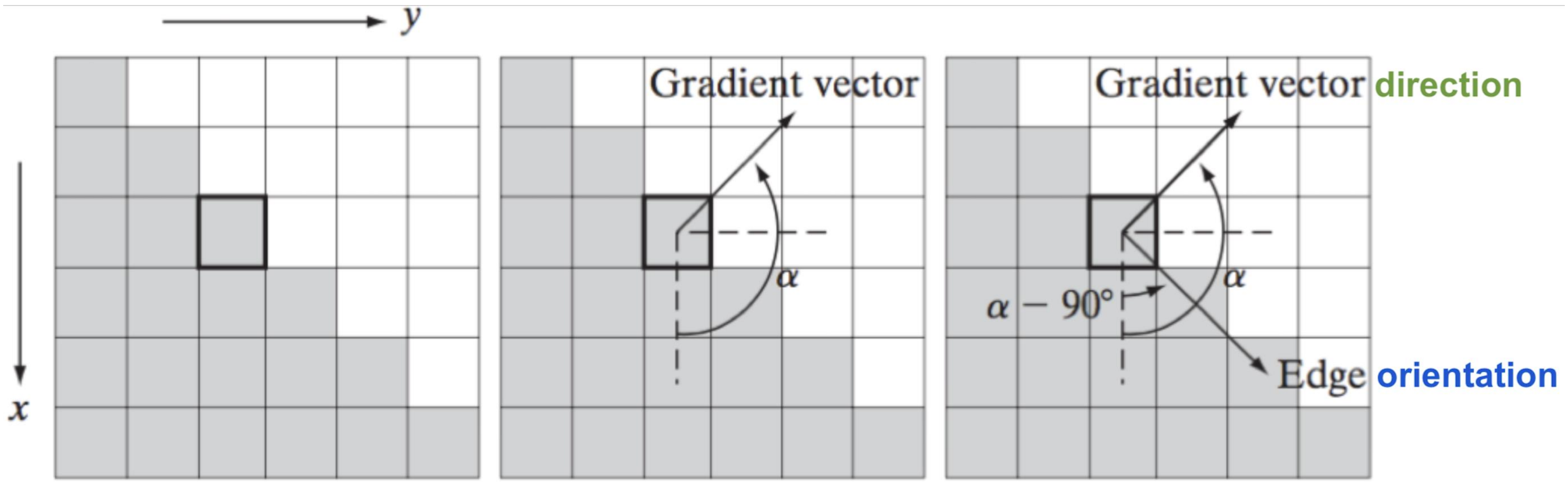
An edge detector typically produces:

1. Edge **position**.
2. Edge **magnitude** (strength).
3. Edge **orientation** (orthogonal to the gradient **direction**).

Performance requirements:

1. High detection rate.
2. Good localization
3. Low sensitivity to noise.

Edge detection: Introduction



- **Direction** of an adge refers to the way the entensity changes from one side of the edge to the other.
- **Orientation** describes the angle in which the edge is aligned.

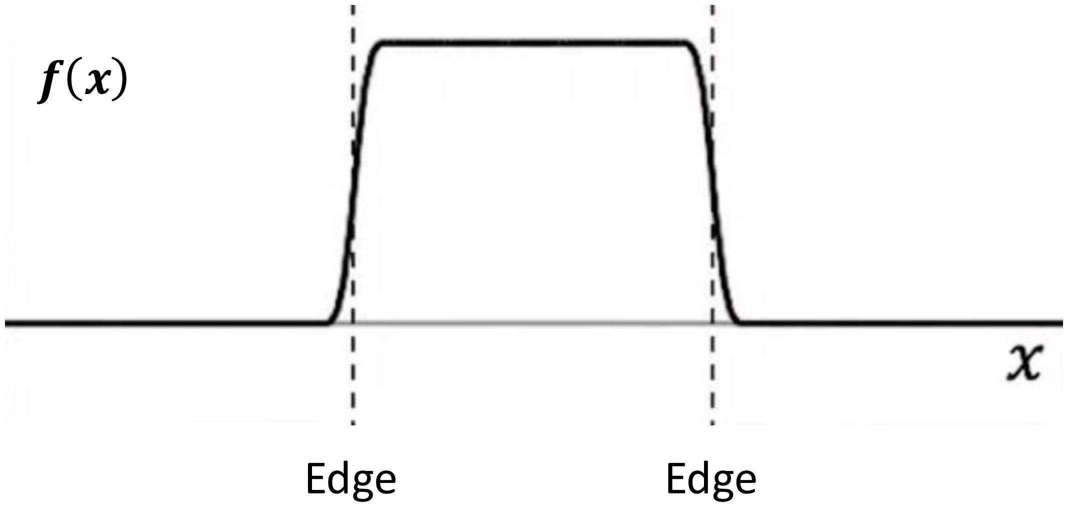
⇒ Direction is about the change in intensity while orientation is about the alignment of that change.

I. Edge Detection using Gradients

- 1-D edge detection:

  Edge: Rapid change in image intensity in a small region.

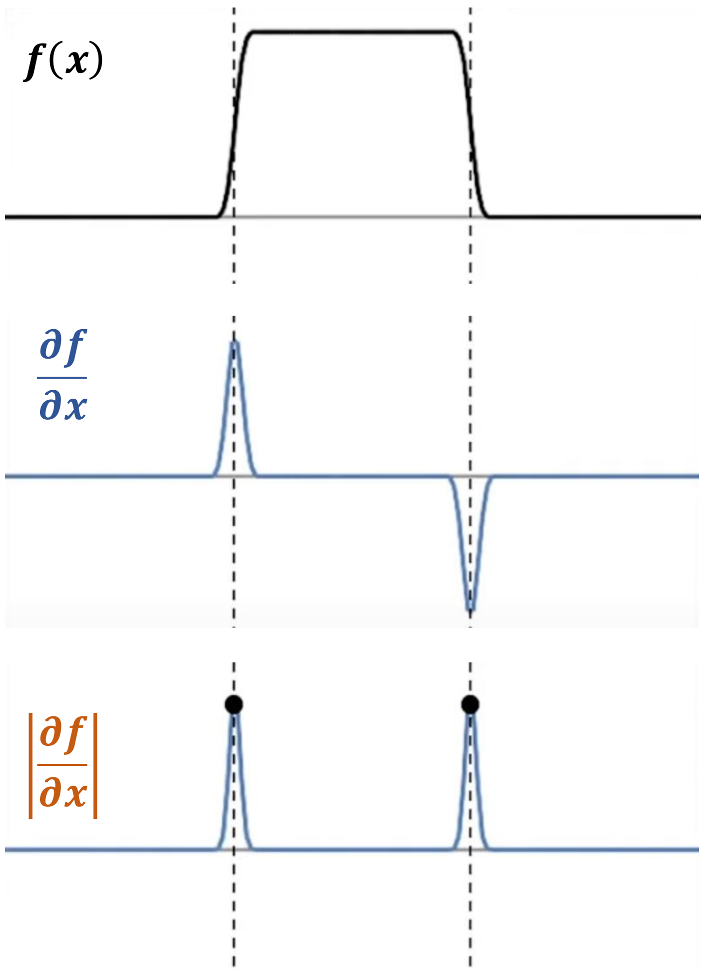
We know that the derivative of a continuous function represents the amount of change in the function.



- The **magnitude of the first derivative** can be used to detect the presence of an edge at a point in an image.

I. Edge Detection using Gradients

- 1-D edge detection using derivatives



Local extrema indicate edges

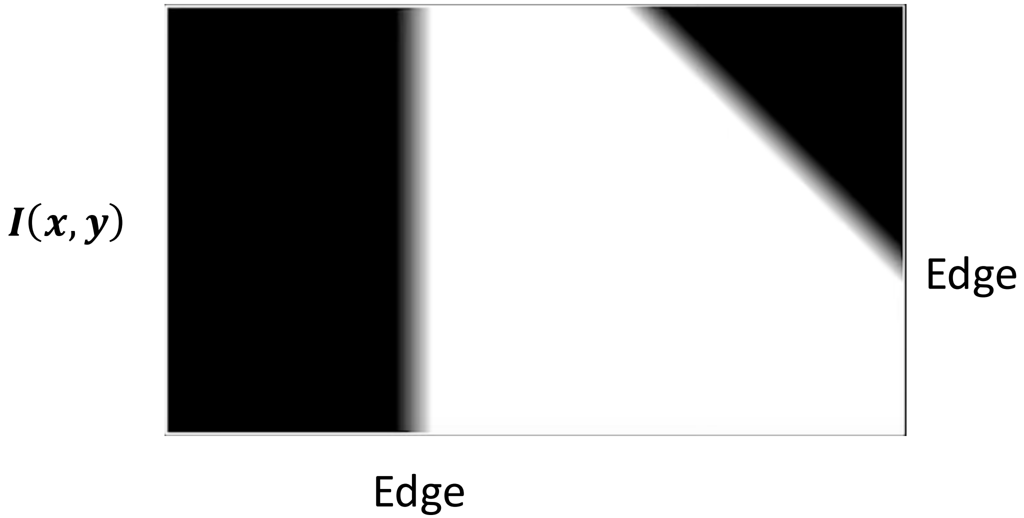
Local maxima indicate edges

$\Rightarrow$ We get both location and strength of edges.

I. Edge Detection using Gradients

- 2D edge detection using derivatives:

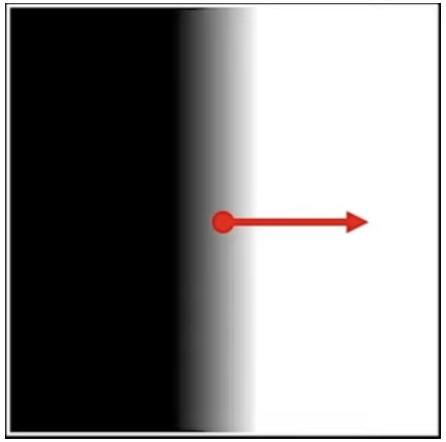
Basic calculus: Partial derivatives of a continuous 2D function represents the amount of change along each dimension.



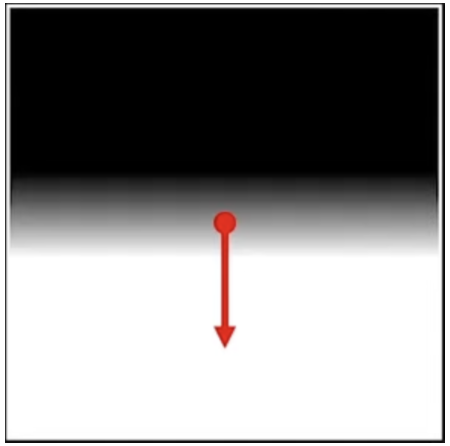
I. Edge Detection using Gradients

- 2D edge detection using derivatives: The gradient operator (∇)

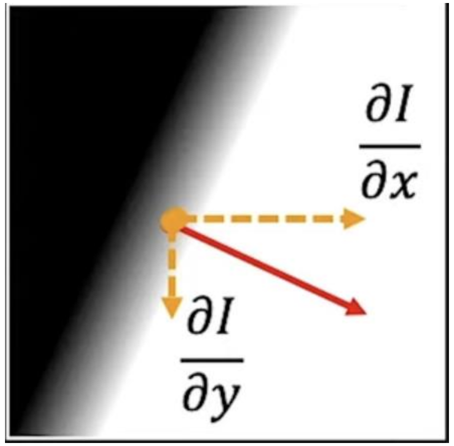
$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0\right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$



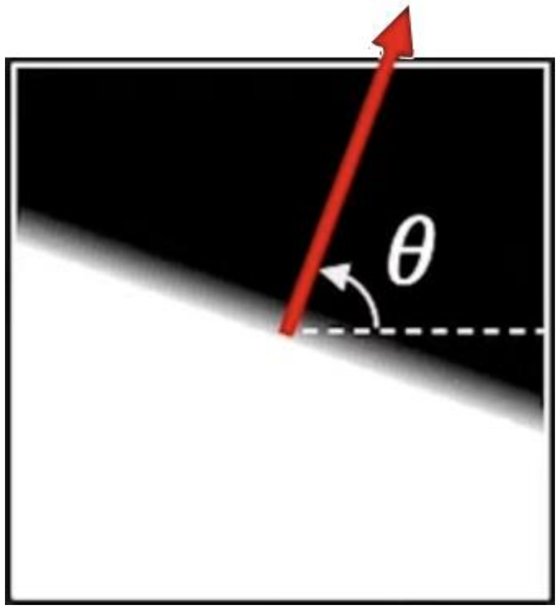
$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

I. Edge Detection using Gradients

- 2D edge detection using derivatives: The gradient operator (∇)

Magnitude: $$S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

Orientation: $$\theta = tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$$

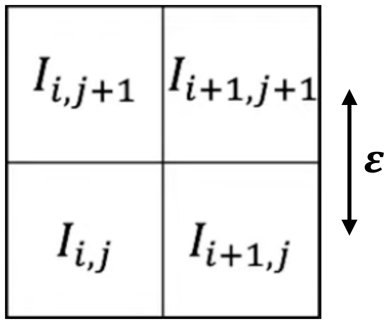


I. Edge Detection using Gradients

- Application to discrete images:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\,\varepsilon}\Big(\big(I_{i+1,\,j+1} - I_{i,\,j+1}\big) + \big(I_{i+1,\,j} - I_{i,\,j}\big)\Big)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\,\varepsilon}\Big(\big(I_{i+1,\,j+1} - I_{i+1,\,j}\big) + \big(I_{i,\,j+1} - I_{i,\,j}\big)\Big)$$



- It can be implemented as convolution:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\,\varepsilon}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \frac{\partial I}{\partial y} \approx \frac{1}{2\,\varepsilon}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

I. Edge Detection using Gradients

- A variety of gradient operator has been proposed:

Gradient	Roberts	Prewitt	Sobel (3x3)	Sobel (5x5)
$\frac{\partial I}{\partial x}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -3 & 0 & 3 & 2 \\ -3 & -5 & 0 & 5 & 3 \\ -2 & -3 & 0 & 3 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$
$\frac{\partial I}{\partial y}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 3 & 5 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -3 & -5 & -3 & -2 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix}$

Good localization $\longrightarrow$ **Poor localization**

Noise sensitive $\longrightarrow$ Less noise sensitive

Poor detection $\longrightarrow$ Good detection

I. Edge Detection using Gradients

- Gradient using a 3x3 Sobel operator

Image I

$\frac{\partial I}{\partial y}$

Gradient magnitude

$\frac{\partial I}{\partial x}$

I. Edge Detection using Gradients

Thresholding: Deciding which pixels definitely belong to an edge.

- Standard: Using a single threshold (T)

$\|\nabla I(x,y)\| < T$	Definitely not an edge
$\|\nabla I(x,y)\| \geq T$	Definitely an edge

- Hysteresis-based: Using two thresholds ($T_0 < T_1$)

$\|\nabla I(x,y)\| < T_0$	Definitely not an edge
$\|\nabla I(x,y)\| \geq T_1$	Definitely an edge
$T_0 \leq \|\nabla I(x,y)\| < T_1$	Is an edge if a neighboring pixel is definitely an edge

I. Edge Detection using Gradients

Thresholding: Deciding which pixels definitely belong to an edge.

- If the threshold is too high, important edges may be missed.
- If the threshold is too low, noise may be mistaken for a genuine edge.

I. Edge Detection using Gradients

- The Lena example.

Image I

$\frac{\partial I}{\partial x}$

$\frac{\partial I}{\partial y}$

Gradient magnitude

Threshold edge

I. Edge Detection using Gradients

- Another example.

Image I

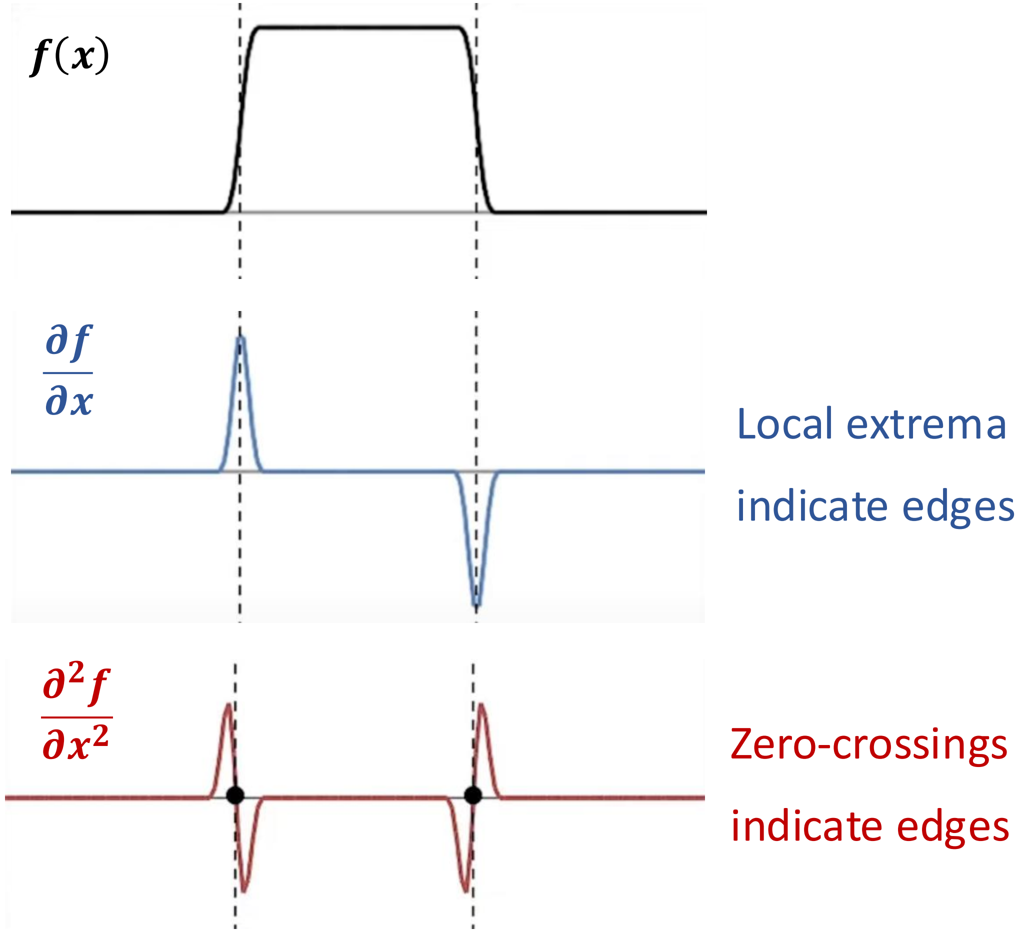
$\frac{\partial I}{\partial x}$

Gradient magnitude

$\frac{\partial I}{\partial y}$

II. Edge Detection using Laplacian

- Using the second derivative



II. Edge Detection using Laplacian

- Laplacian (∇^2) as edge detector

Laplacian: The sum of pure second derivatives.

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

This measures how much the intensity differs from its neighbors in **all directions**.

$\Rightarrow$ Edges are zero-crossings in the Laplacian image.

II. Edge Detection using Laplacian

- Laplacian (∇^2) as edge detector

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

$\Rightarrow$ A **zero-crossing** is a point where the **sign of the Laplacian response changes:** from positive to negative or vice versa.

$\Rightarrow$ These sign changes usually occur at **edges**, especially where there is a sharp change in intensity.

$\Rightarrow$ **Laplacian does not provide the direction of edges.**

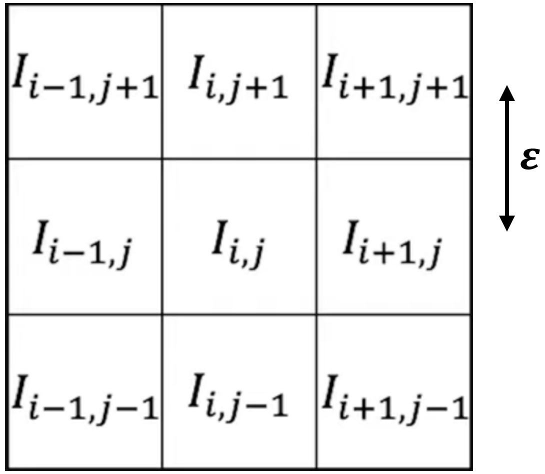
II. Edge Detection using Laplacian

- Discrete Laplacian operator (∇^2)

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{1}{\varepsilon^2}\left(I_{i-1,\,j} - 2I_{i,j} + I_{i+1,\,j}\right)$$

$$\frac{\partial^2 I}{\partial y^2} \approx \frac{1}{\varepsilon^2}\left(I_{i,\,j-1} - 2I_{i,j} + I_{i,\,j+1}\right)$$

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$



- Using convolution:

$$\nabla^2 \approx \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \nabla^2 \approx \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} \qquad \nabla^2 \approx \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

II. Edge Detection using Laplacian

Step-by-step: How zero-crossings are detected

1. Apply the Laplacian operator

- Convolve the image with a Laplacian kernel.
- This results in a **new 2D image** of Laplacian values: some positive, some negative.

2. Check each pixel and its neighbors

To detect a zero-crossing at a pixel, we have to check **its 8-connected neighbors** (horizontal, vertical, and diagonal):

- If **the sign of the Laplacian value changes** between a pixel and one of its neighbors,
- AND the **absolute difference** is large enough (to avoid tiny sign flips due to noise),

=> Then a **zero-crossing** (i.e., an edge) is detected.

II. Edge Detection using Laplacian

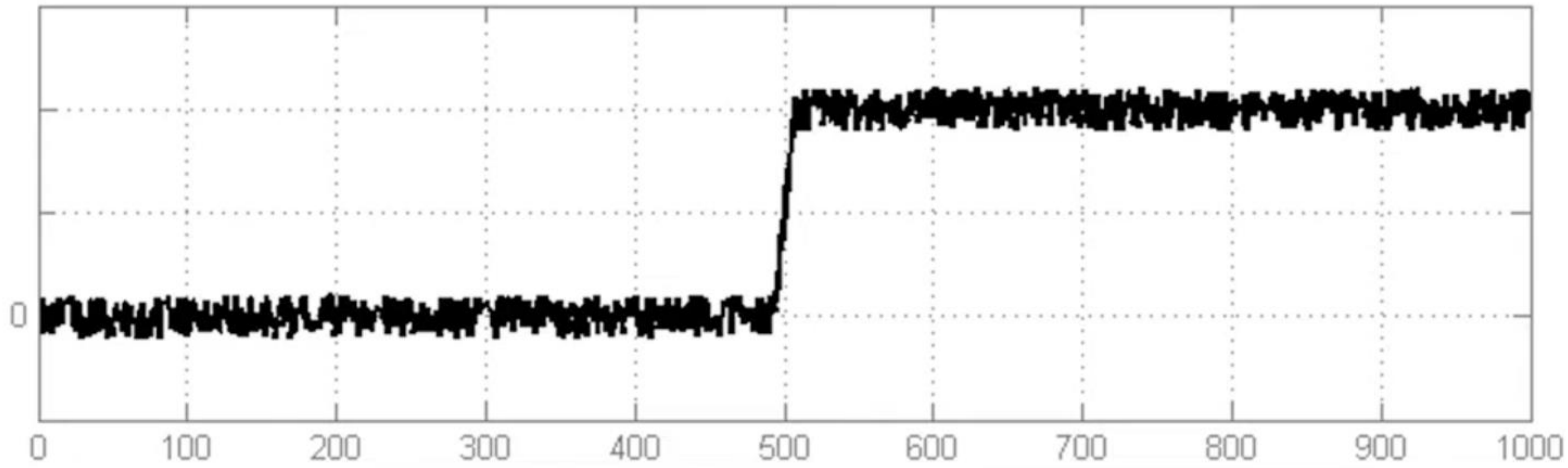
- Application to the Lena image

Laplacian
(0 maps to 128)

Laplacian
Zero-crossings

III. Edge Detection: Derivatives and Noise

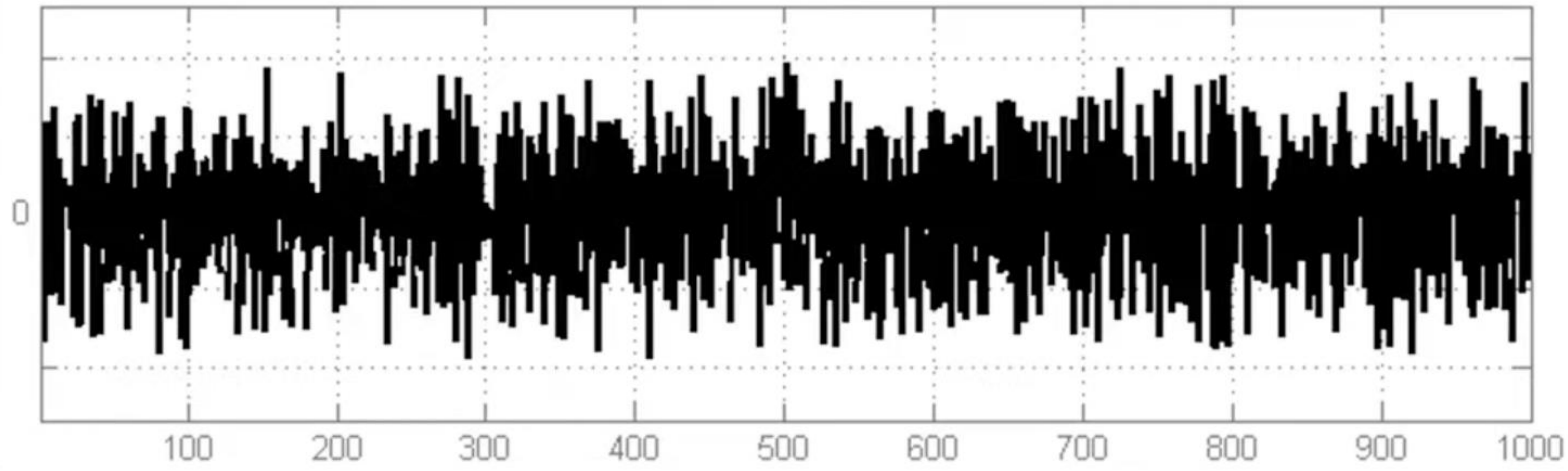
$f(x)$



The noise introduces rapid changes in the signal – and the edge is also defined as a rapid change in the signal

$\nabla f(x)$

(Gradient)



*Noise leads to a **false interpretation** of the edge.*

The edge is completely lost.

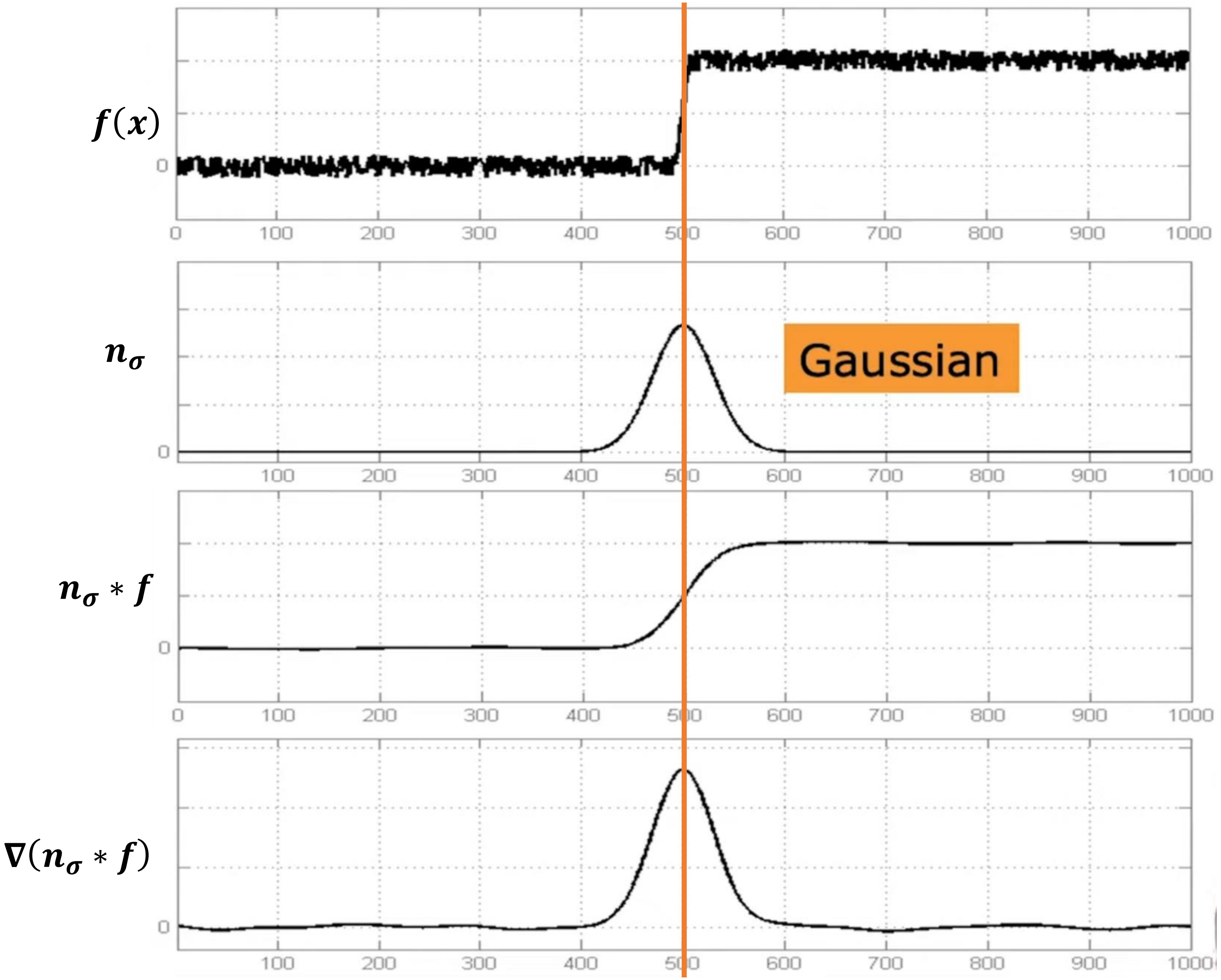
III. Edge Detection: Derivatives and Noise

- Noise is an important issue to keep in mind since it can have a significant impact on the two key derivatives used for detecting edges.
- The noise has to be removed/mitigated before applying edge detection.

$\Rightarrow$ **The solution:** Image smoothing. It is usually a mandatory step prior to the use of derivatives.

III. Edge Detection: Derivatives and Noise

- Smoothing the signal first using Gaussian filtering



III. Edge Detection: Derivatives and Noise

Using derivate of Gaussian

- We use $\nabla(\boldsymbol{n_\sigma} * \boldsymbol{f})$
- We know that:
  - ∇ is a linear operation
  - Gaussian smoothing is linear

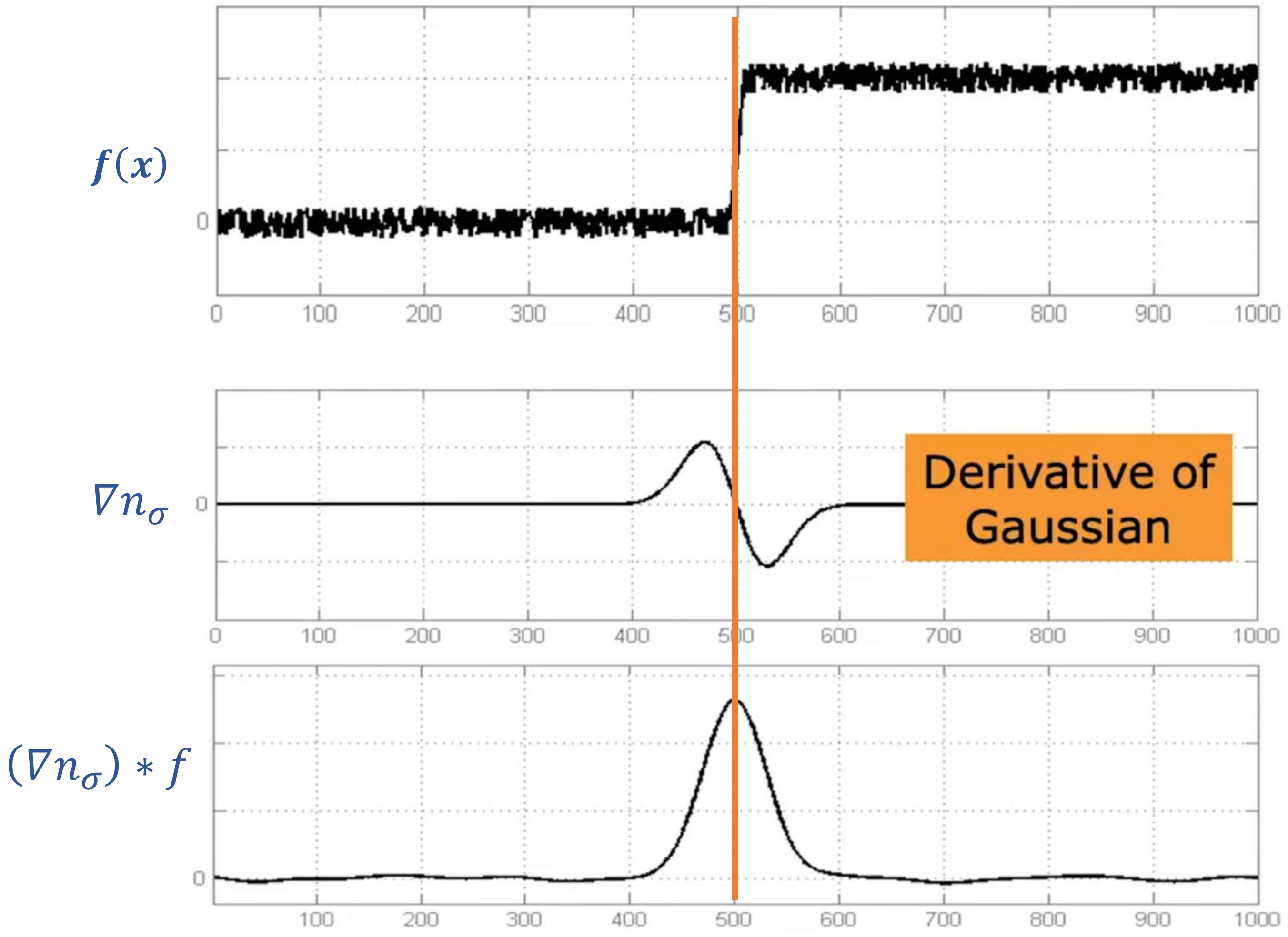
$\Rightarrow$ $\nabla(\boldsymbol{n_\sigma} * \boldsymbol{f}) = (\nabla \boldsymbol{n_\sigma}) * \boldsymbol{f}$

$\Rightarrow$ $\nabla \boldsymbol{n_\sigma}$ is a new operator that can be computed once and convolved with $\boldsymbol{f}$ to get the derivative on a smoothed version of $\boldsymbol{f}$

III. Edge Detection: Derivatives and Noise

Using derivate of Gaussian

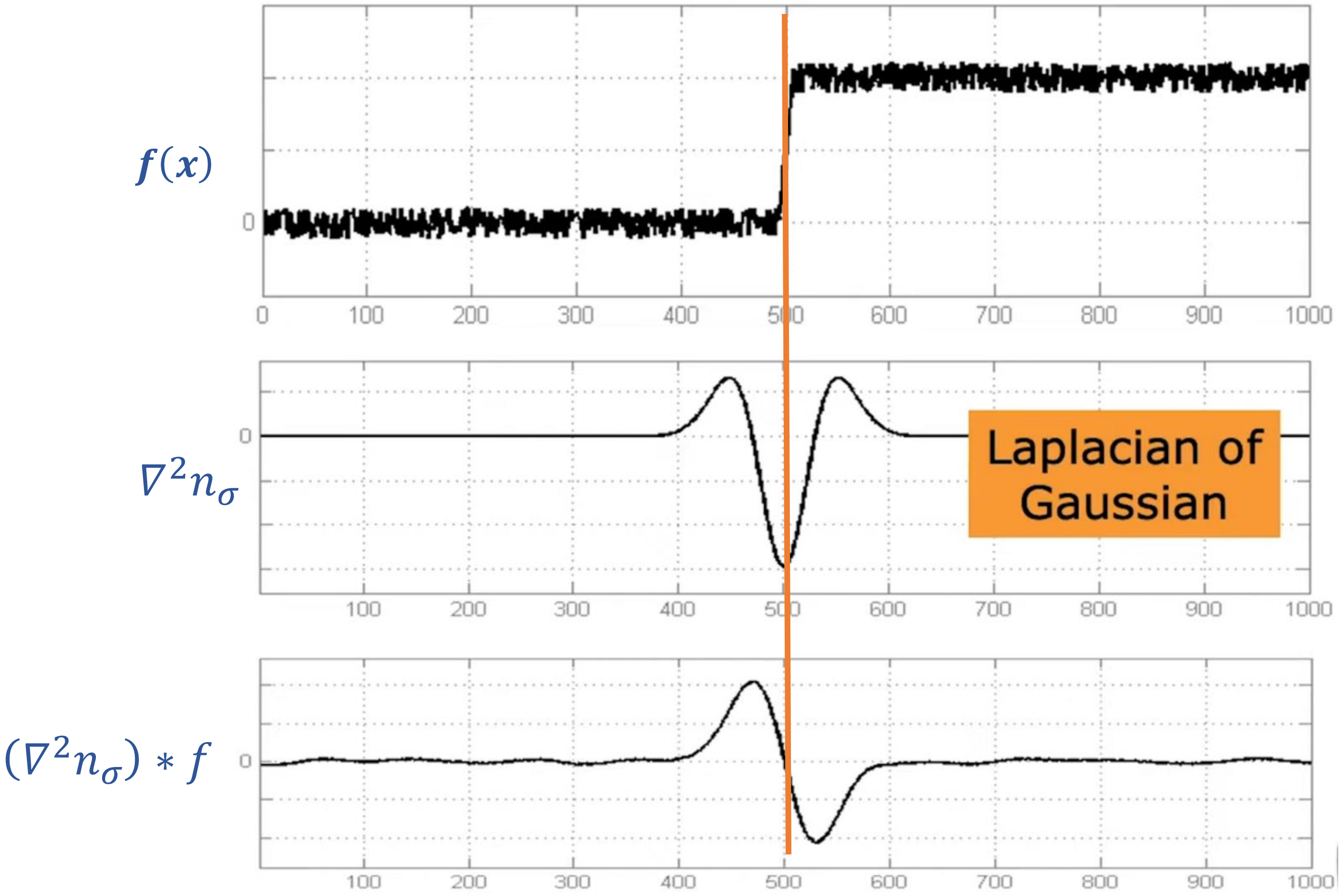
$$\nabla(n_\sigma * f) = (\nabla n_\sigma) * f$$



III. Edge Detection: Derivatives and Noise

- The same can be applied to Laplacian:
  Laplacian of Gaussian ($\nabla^2 G$)

$$\nabla^2(n_\sigma * f) = (\nabla^2 n_\sigma) * f$$

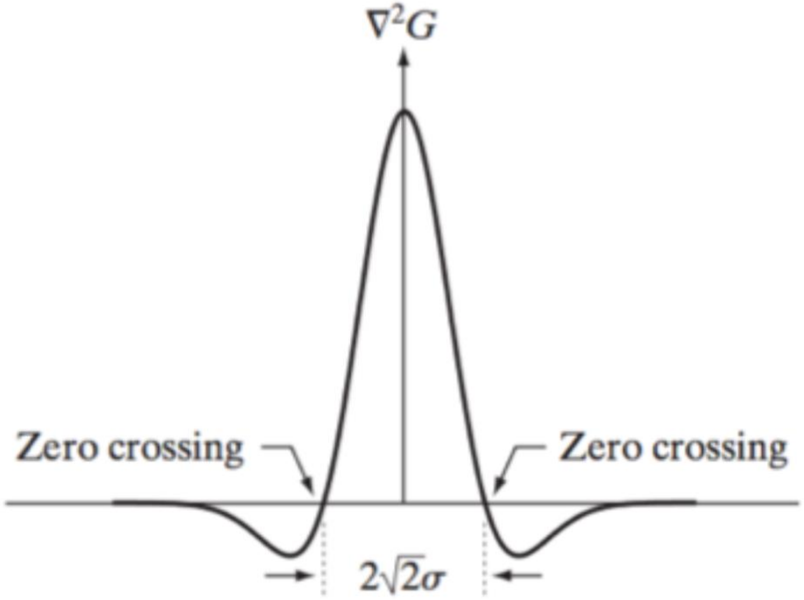


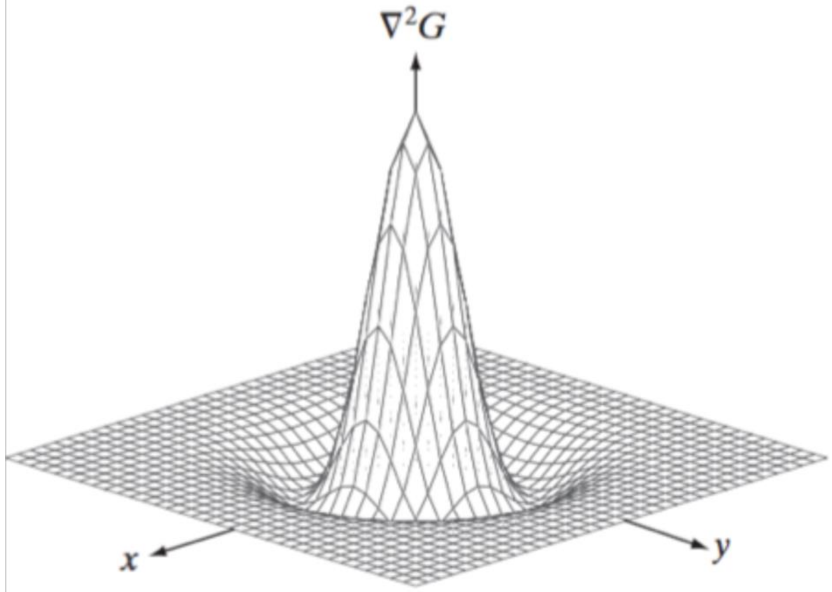
III. Edge Detection: Derivatives and Noise

- **Laplacian of Gaussian ($\nabla^2 G$)**

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

$$= \left[\frac{x^2+y^2+2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

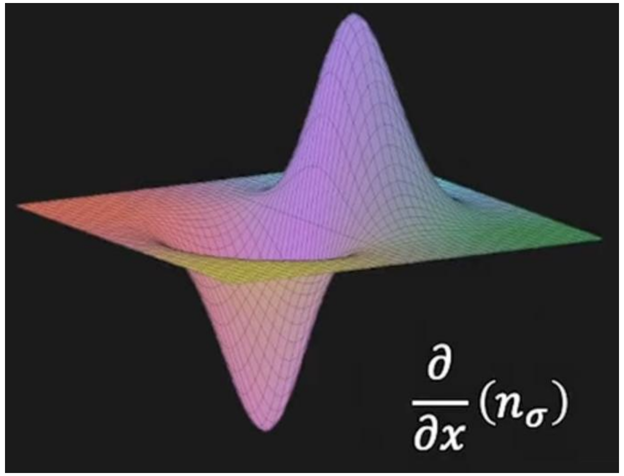


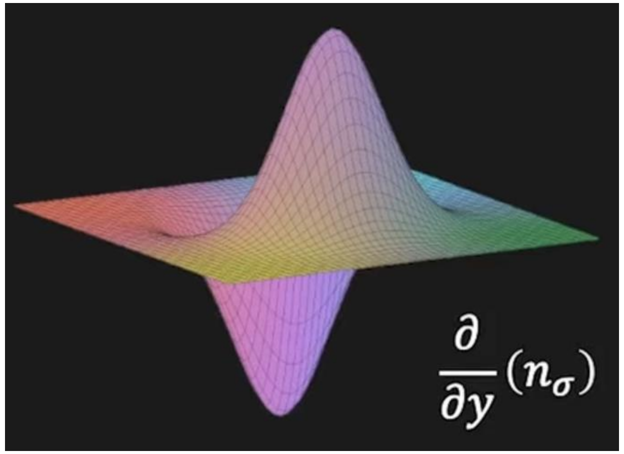


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

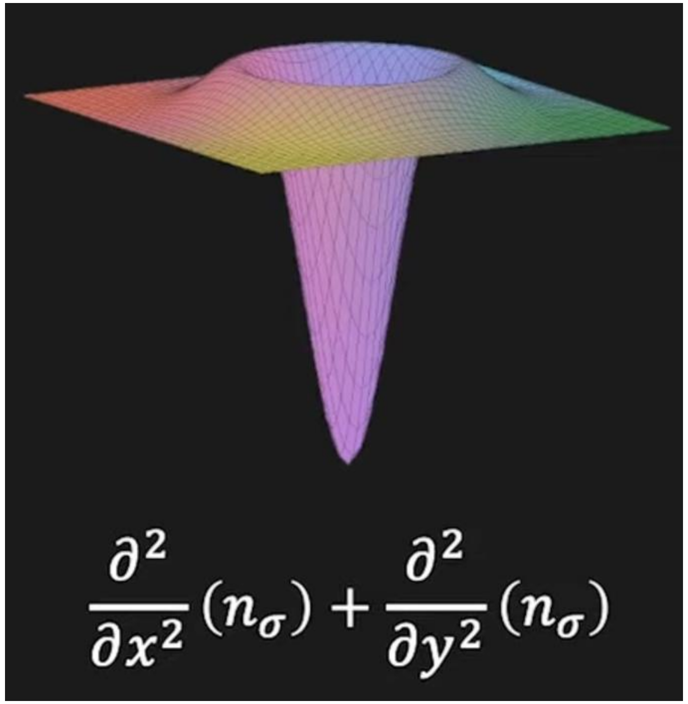
III. Edge Detection: Gradient vs. Laplacian

Gradient





Laplacian



Inverted "sombrero"

(Mexican hat)

III. Edge Detection: Gradient vs. Laplacian

Gradient	Laplacian
• Provide location, magnitude and orientation.	• Provide only location of the edge.
• Detection using maxima thresholding.	• Detection based on zero-crossings.
• Non-linear operation, requires two convolutions.	• Linear operation, requires only one convolution.