Università
Ca' Foscari
Venezia

Computer Vision

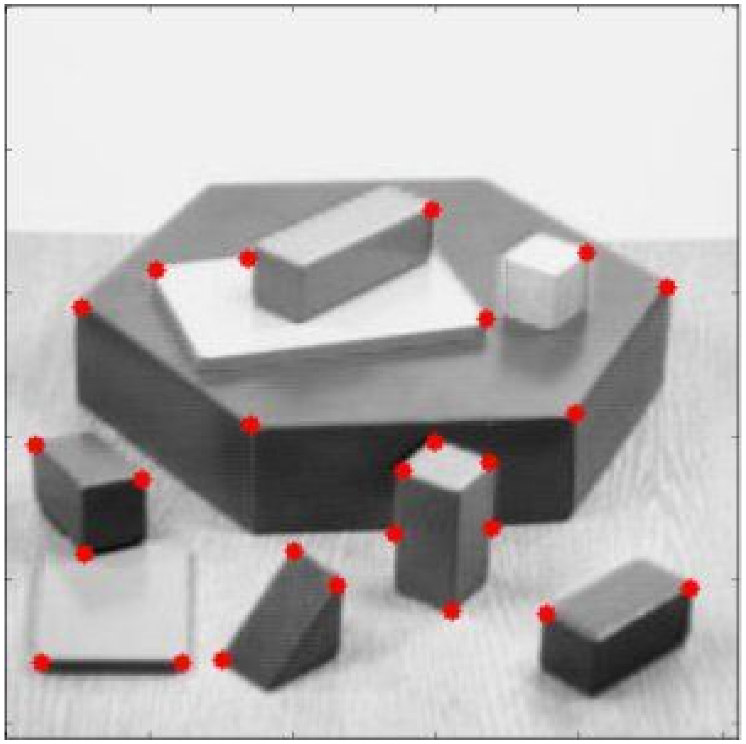
Edge Features

Filippo Bergamasco (filippo.bergamasco@unive.it)
http://www.dais.unive.it/~bergamasco
DAIS, Ca' Foscari University of Venice
Academic year 2016/2017

Feature

What is a feature?

Local, meaningful, detectable parts of the image



Università
Ca' Foscari
Venezia

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

Università Ca' Foscari Venezia

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

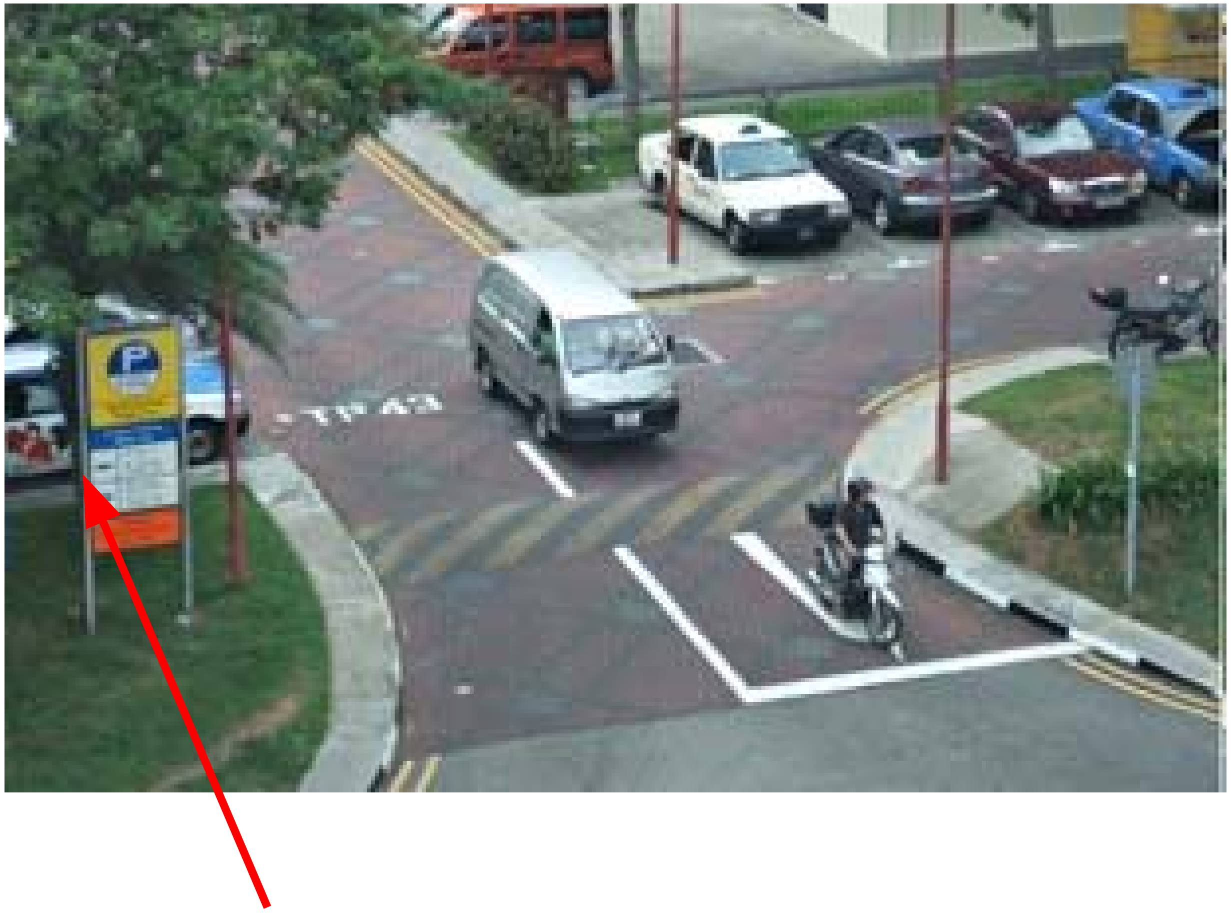
Edges

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?

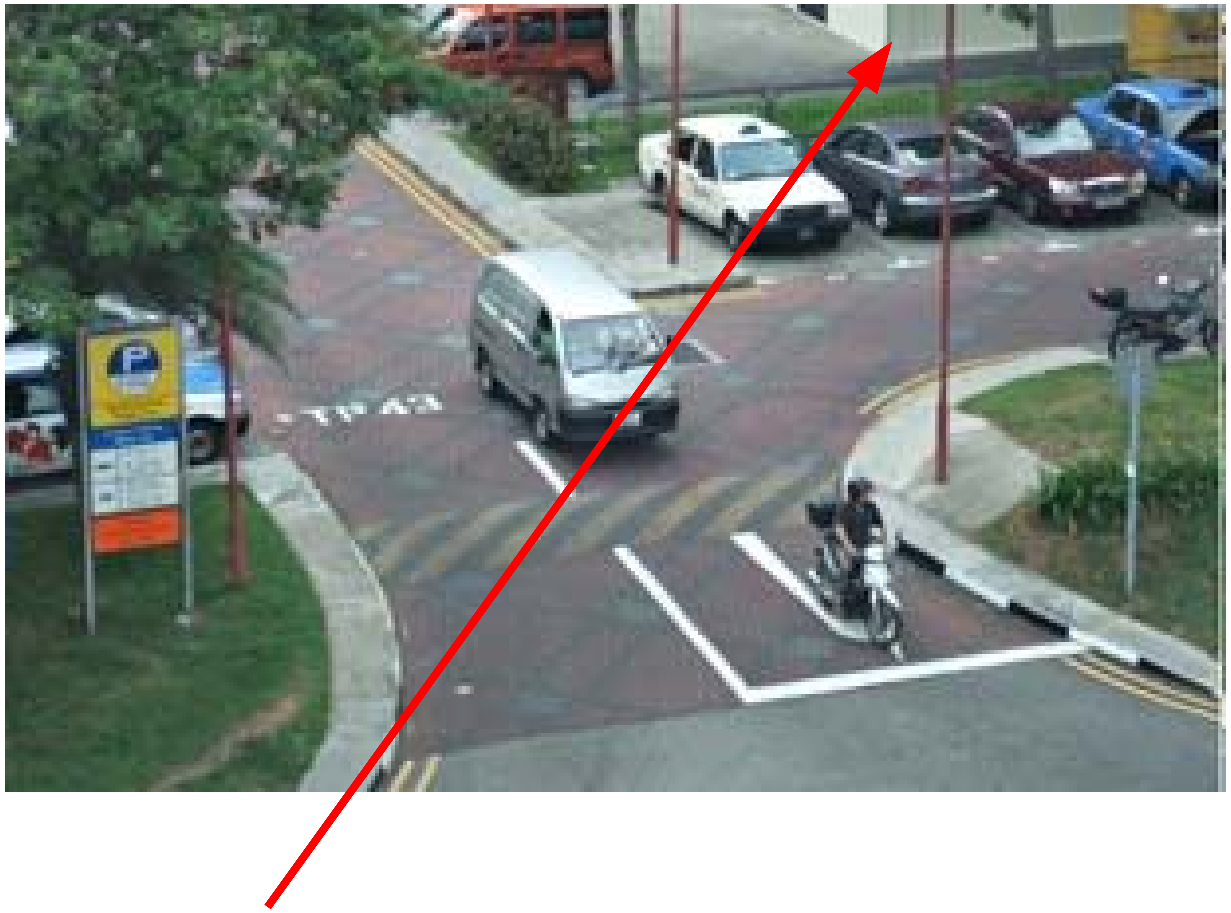
- Edges in an image have many interesting causes
- Looking at edges reduces information required
  - Look at a few pixels in a binary image as opposed to all pixels in a grayscale image
- Biological plausibility
  - Initial stages of mammalian vision systems involve detection of edges and local features

What causes an edge?



Università
Ca' Foscari
Venezia

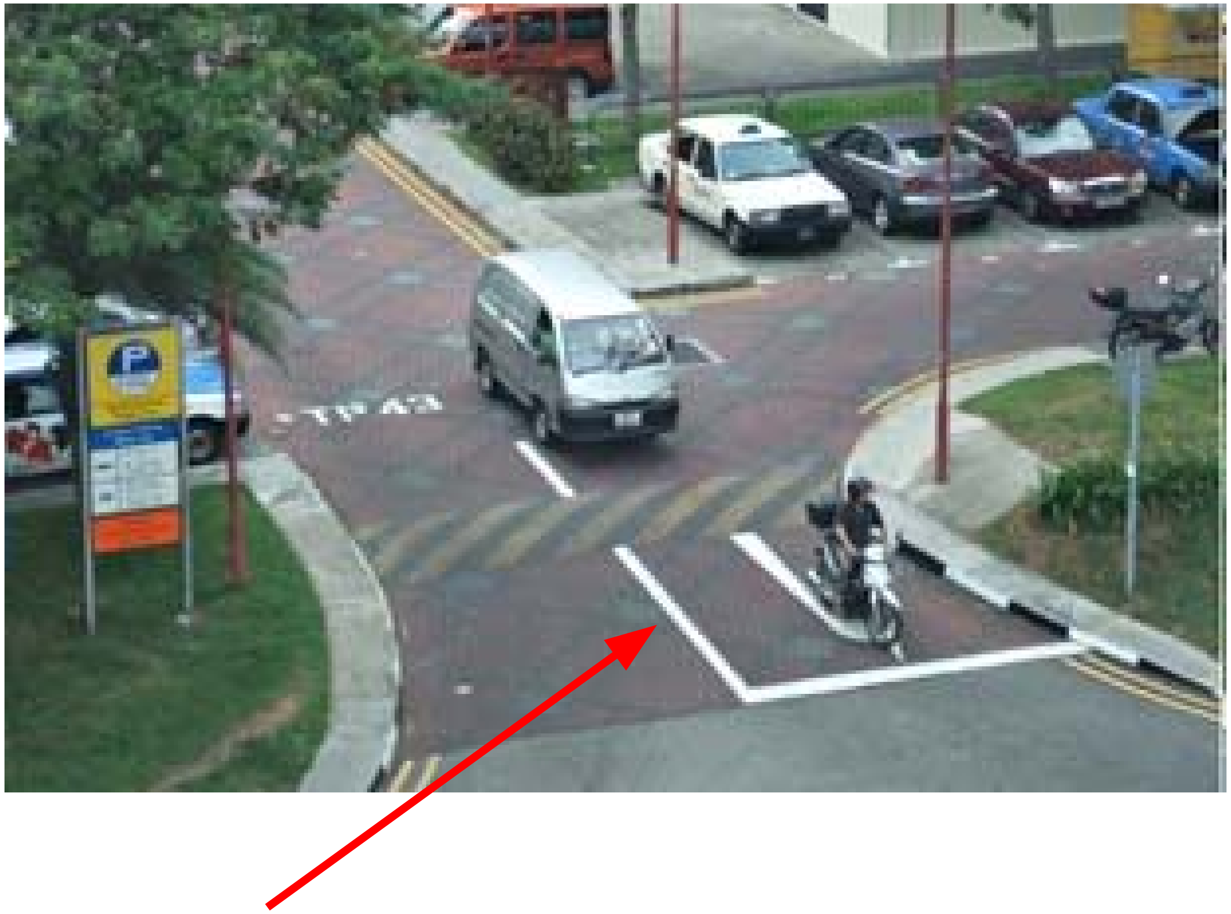
What causes an edge?



Surface orientation
discontinuity

Università Ca' Foscari Venezia

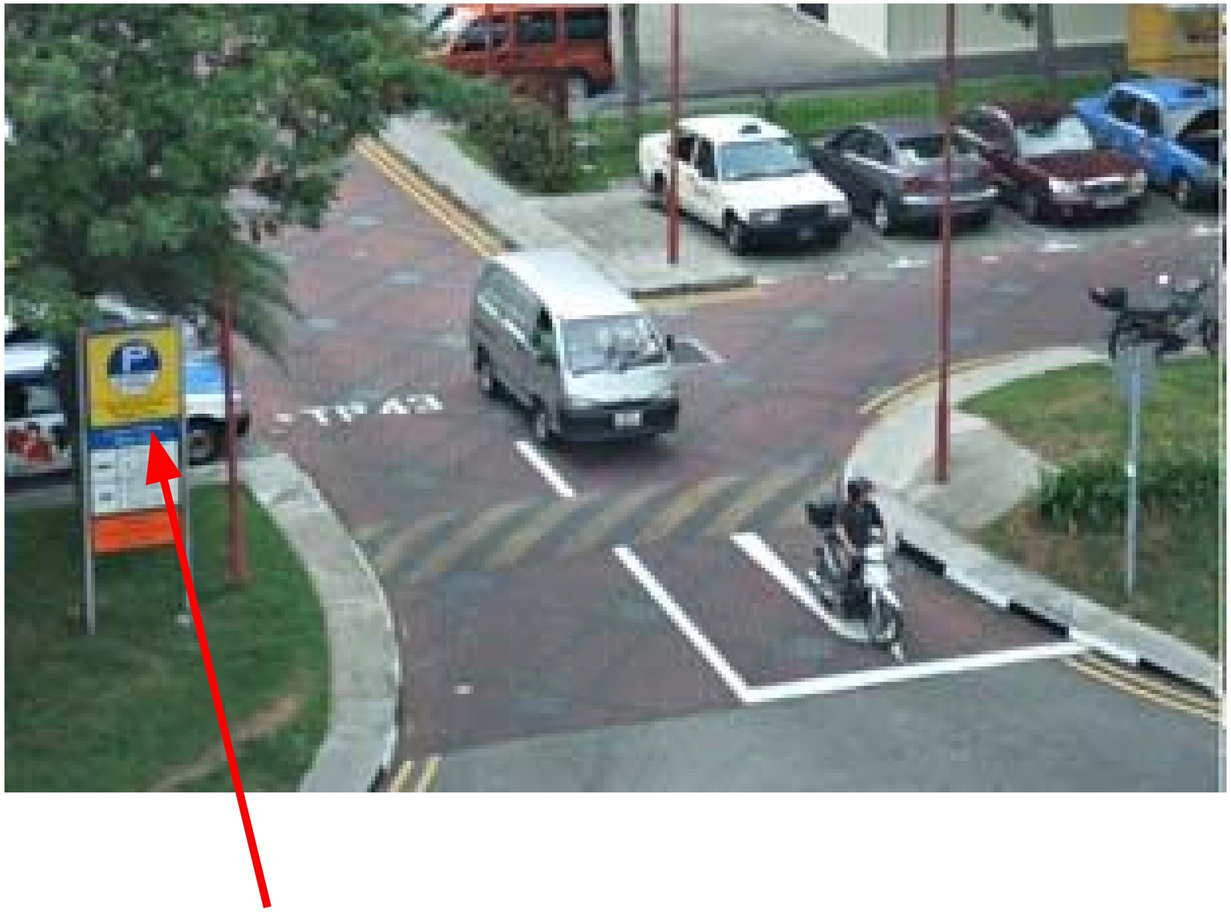
What causes an edge?



Reflectance discontinuity (ie. change of material)

Università
Ca' Foscari
Venezia

What causes an edge?



Surface color discontinuity

Edge detection

Usually, edge detection is performed with multiple steps:

1. Detection of short linear edge segments (edgels)
2. Aggregation of edgels into extended edges
3. Possibly combine the edges

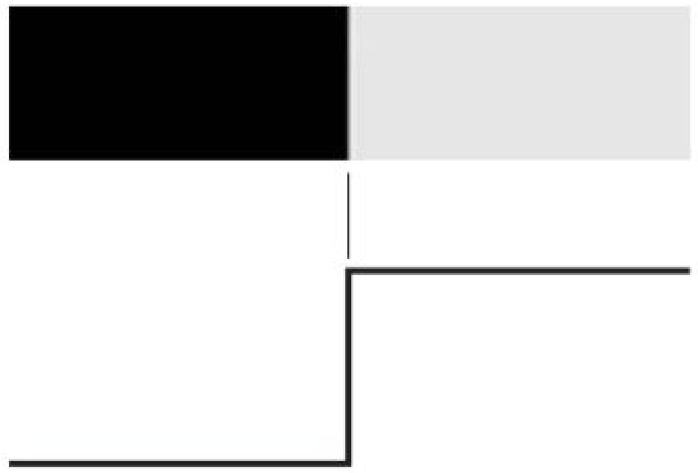
Edgels detection:

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

Università Ca' Foscari Venezia

Edge models

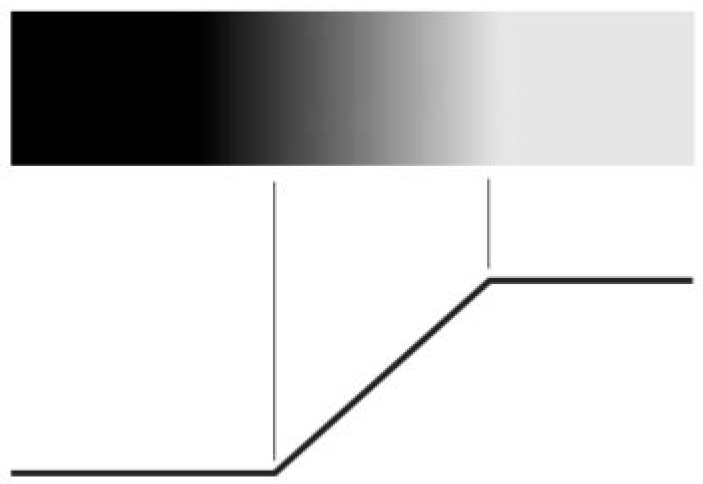
- A **step** edge involves a transition between two intensity levels occurring ideally over the distance of 1 pixel
- Never exists in practice



Università
Ca' Foscari
Venezia

Edge models

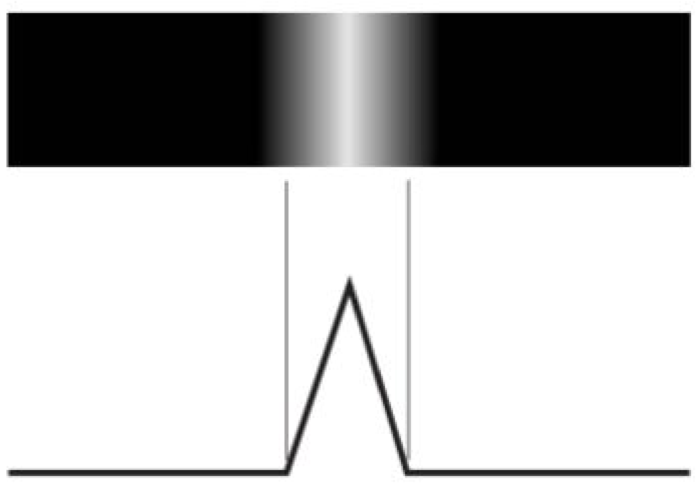
- A **ramp** edge is more common as a certain degree of blurring is introduced by limitations in the focusing mechanism
- The slope of the ramp is inversely proportional to the degree of blurring in the edge.



Università Ca' Foscari Venezia

Edge models

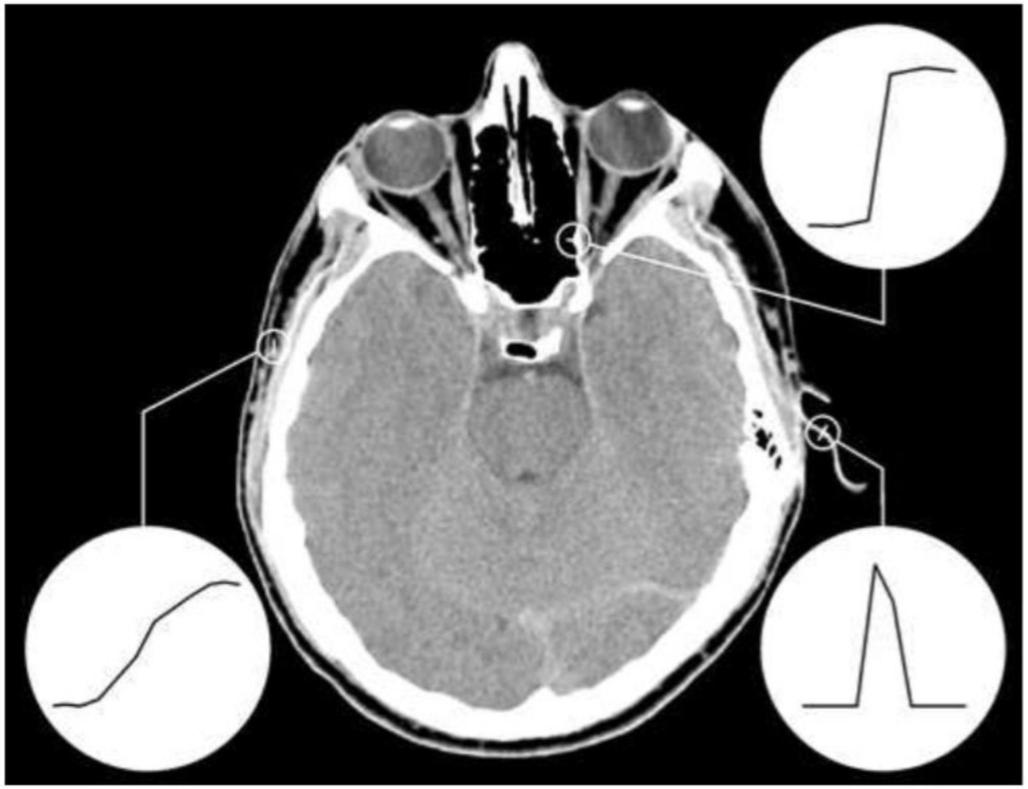
- **Roof** edges may arise for example in range imaging, when thin objects are closer to the sensor than their equidistant background.



Università Ca' Foscari Venezia

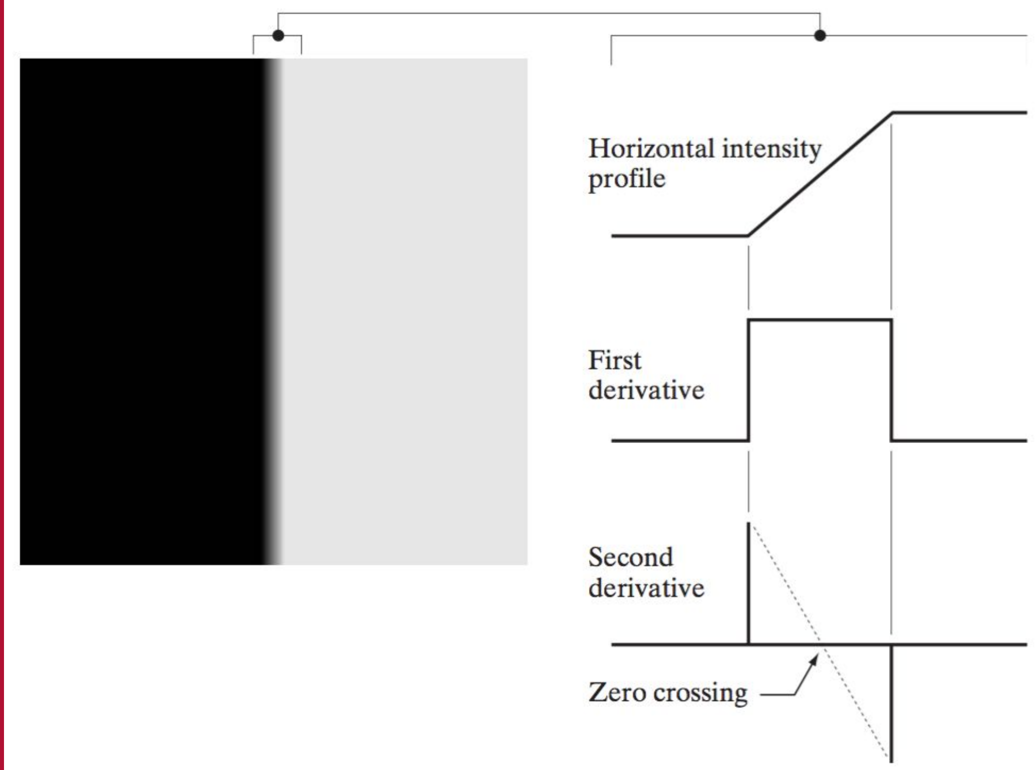
Edge models

Noise result in deviations from the ideal shapes, edges in images that are reasonably sharp and have a moderate amount of noise do resemble the characteristics of the edge models



Derivatives

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



Second derivative has the undesirable effect of producing two responses for each edge of an image
But..
its zero crossings can be used for locating the centers of thick edges,

Università Ca' Foscari Venezia

Image Gradient

The tool of choice for finding edge strength and direction at location (x, y) of an image is the gradient.

For a two-dimensional function, the gradient of f at coordinates (x,y) is defined as the vector:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

- It points in the direction of the greatest rate of change of f at location (x, y)
- Its magnitude is the amount of change in that direction

Università Ca' Foscari Venezia

Image Gradient

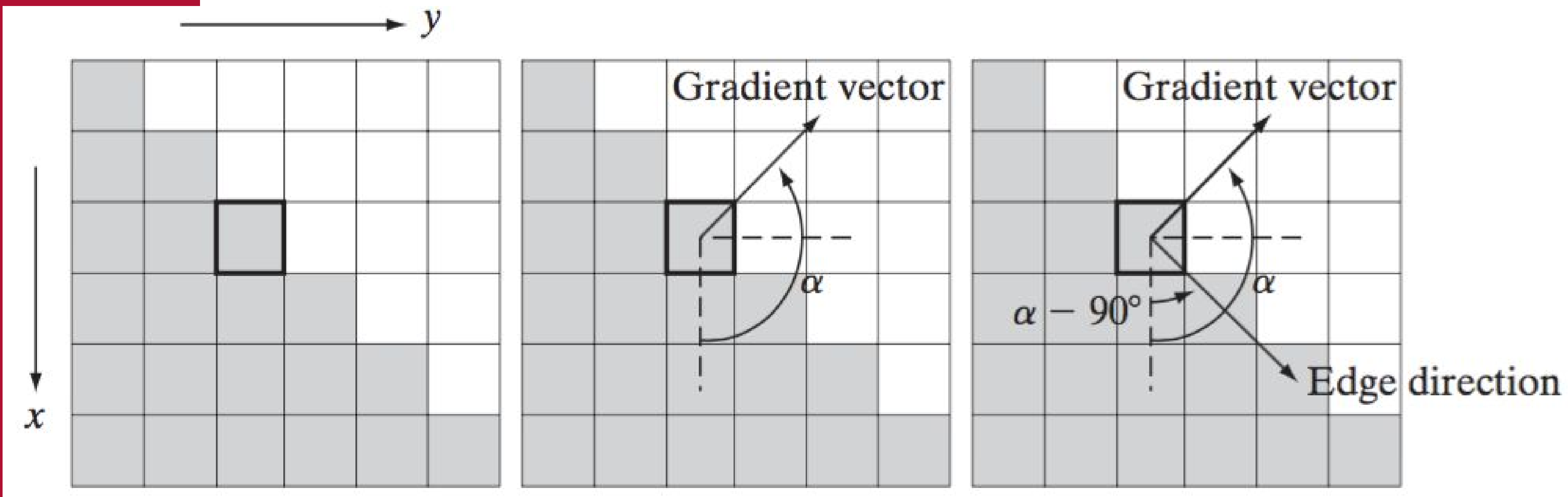
The magnitude of the gradient vector field:

$$M(x,y) = mag(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Is usually referred as “gradient image”. The direction of the gradient vector is given by

$$\alpha(x,y) = \tan^{-1}\left(\frac{\partial f}{\partial y}\Big/\frac{\partial f}{\partial x}\right)$$

Image Gradient



Computing Image Gradient

A digital approximation of the partial derivatives over a neighborhood about a point is required.

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

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

Can be implemented using *Roberts gradient operators*

Problem: Even-sized windows are not symmetric about a center point!

−1
1

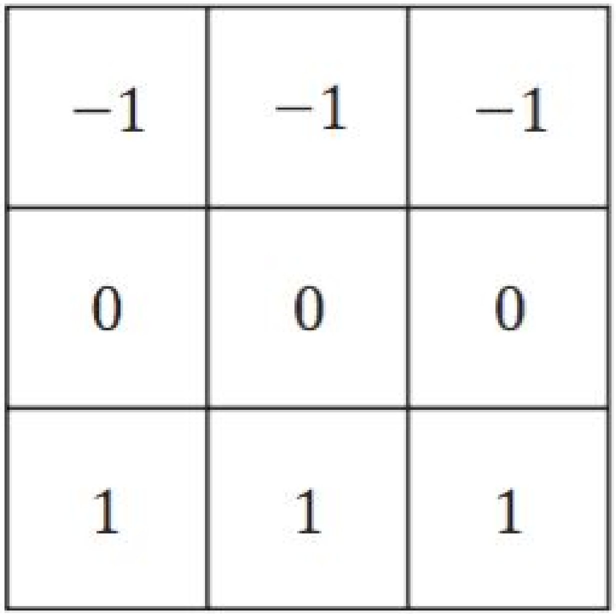
−1	1

−1	0
0	1

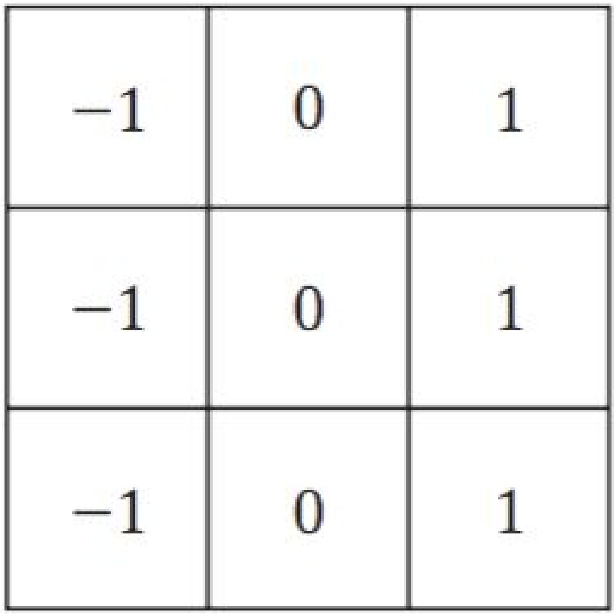
0	−1
1	0

Computing Image Gradient

The simplest digital approximations to the partial derivatives using masks of size 3x3 are given by **Prewitt** operators



−1	−1	−1
0	0	0
1	1	1

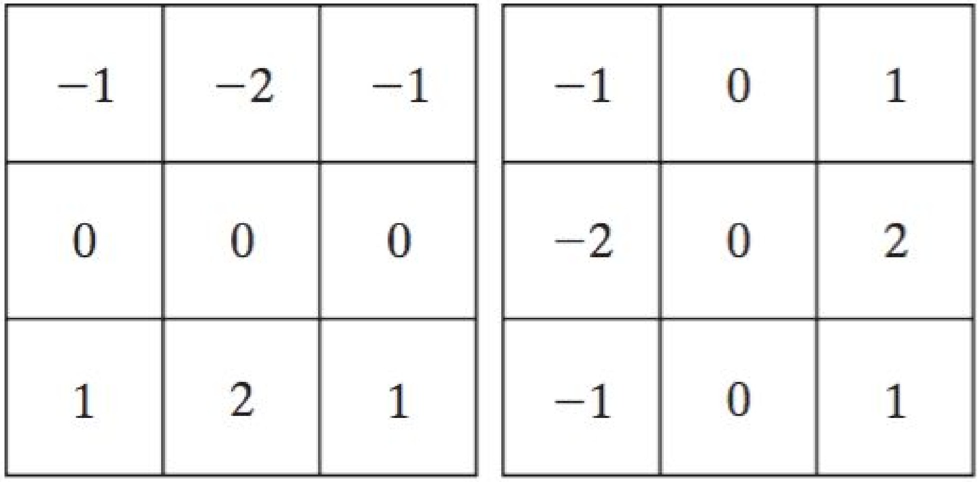


−1	0	1
−1	0	1
−1	0	1

Università Ca' Foscari Venezia

Computing Image Gradient

A slight variation of the preceding two equations uses a weight of 2 in the center coefficient, and are called **Sobel** operators



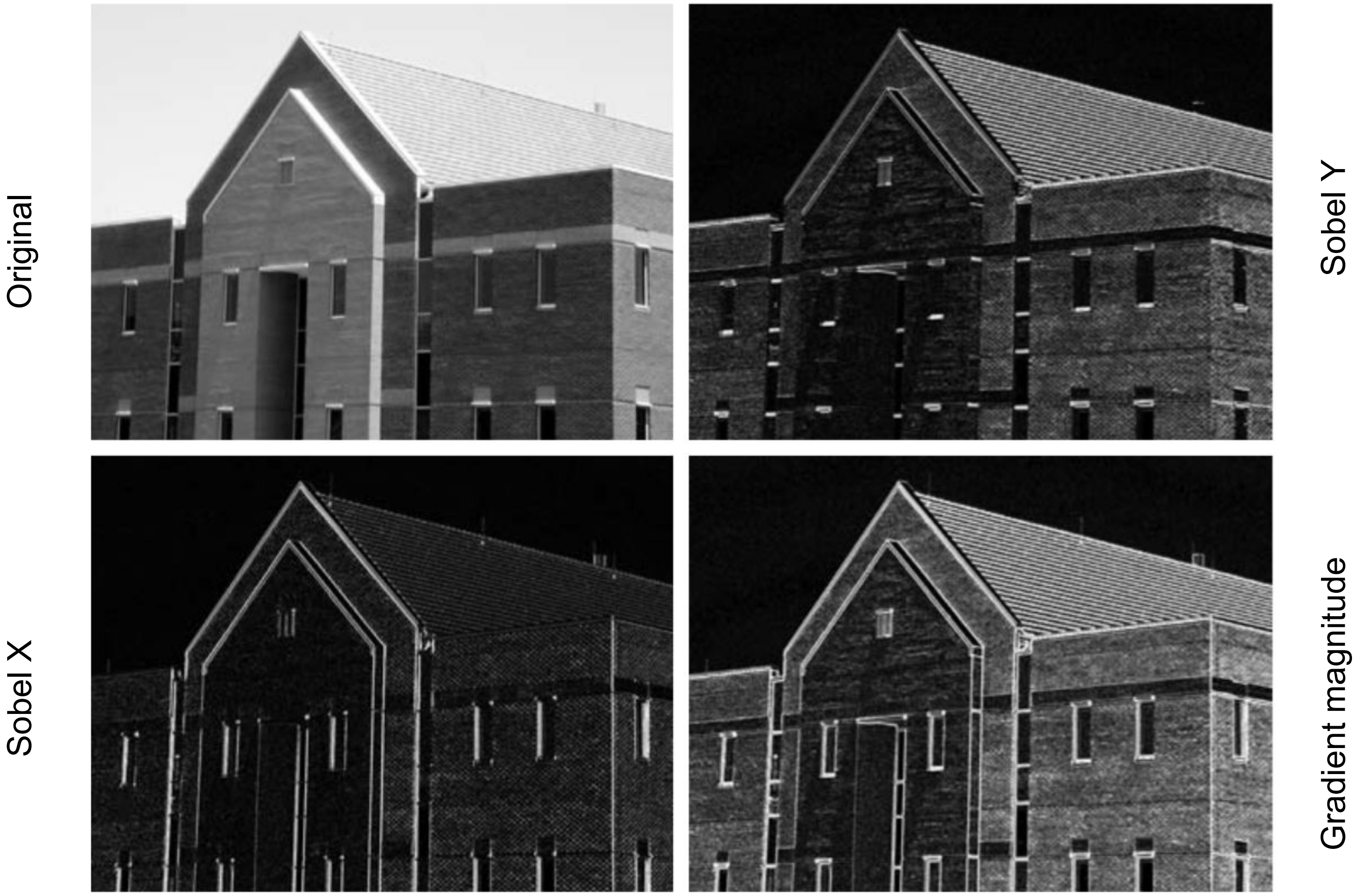
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel masks have better noise-suppression (smoothing) characteristics makes them preferable

Università Ca’ Foscari Venezia

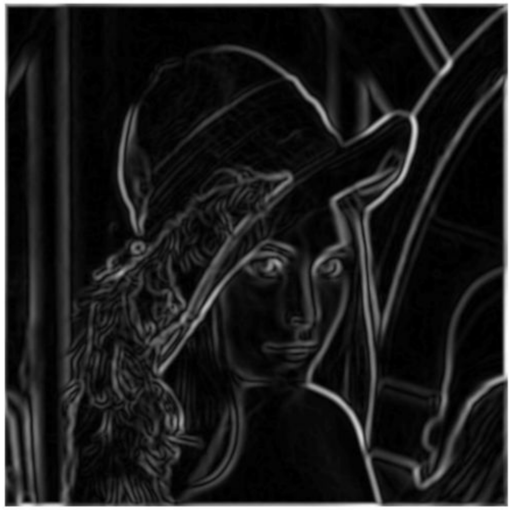
Sobel example

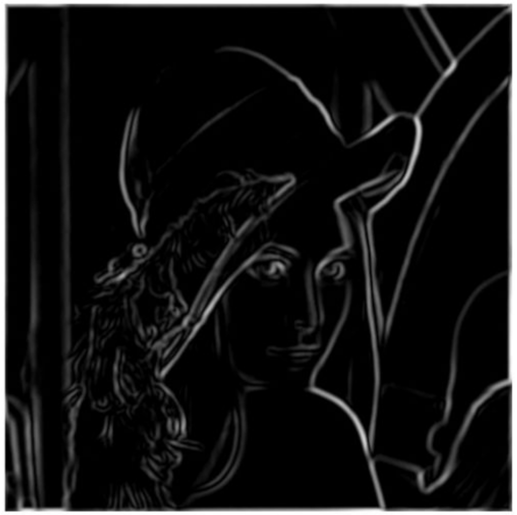


Università Ca' Foscari Venezia

Detection by thresholding

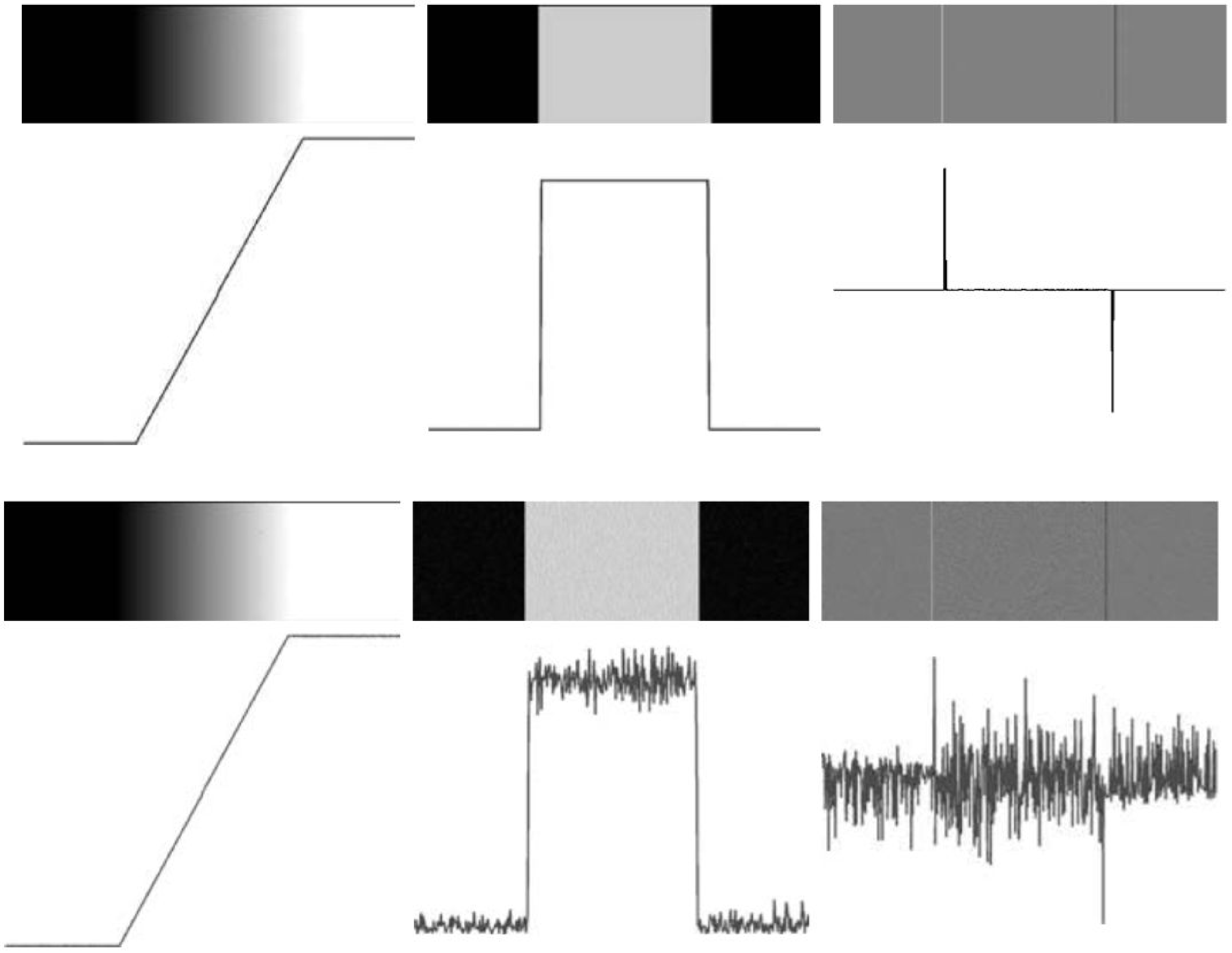
Original image (Lena)


Norm of the gradient


Thresholding

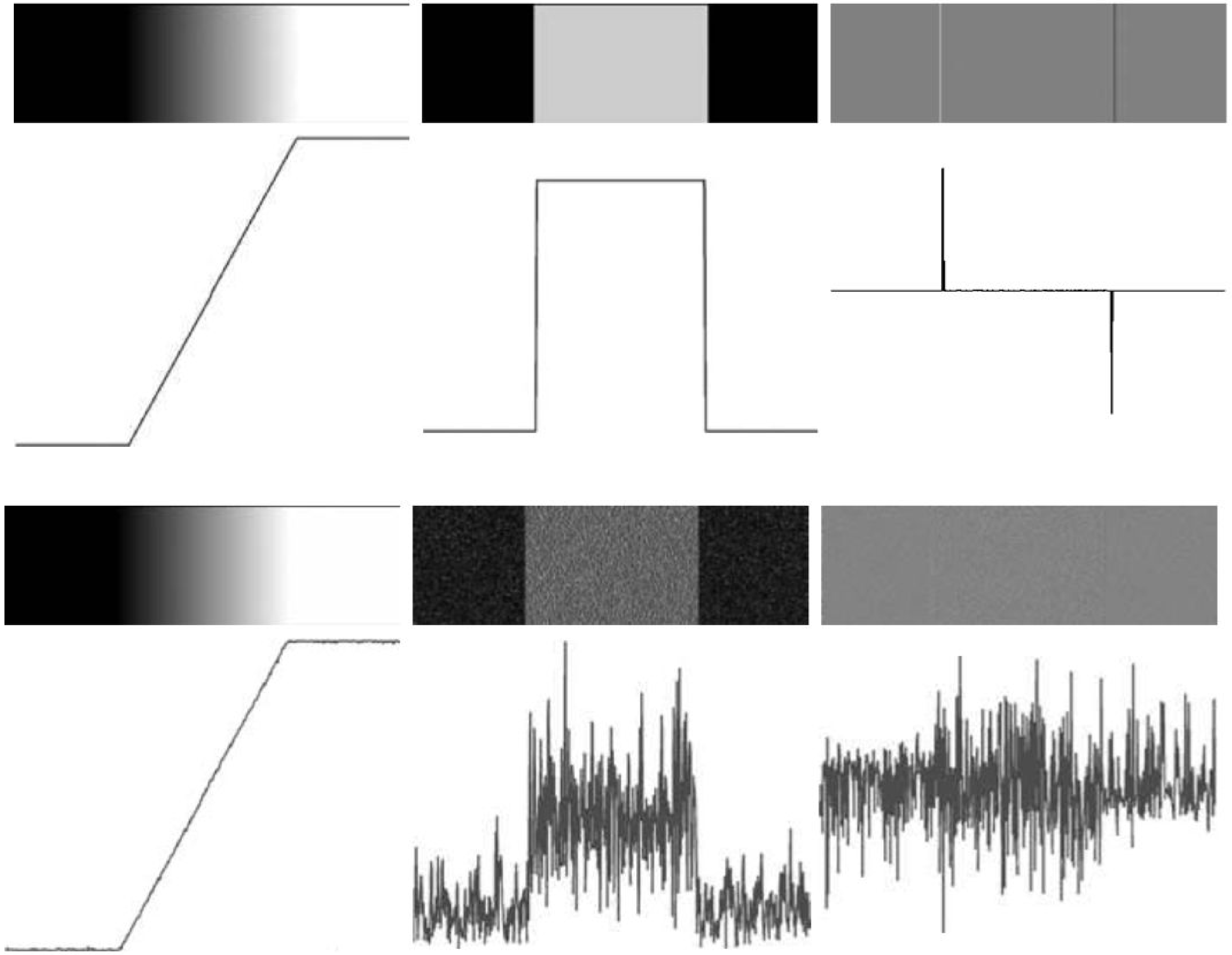
- If the threshold is too high, important edges may be missed
- If the threshold is too low, noise may be mistaken for a genuine edge

Derivatives and noise



Università Ca' Foscari Venezia

Derivatives and noise



Università Ca' Foscari Venezia

Derivatives and noise

Since noise can have such a significant impact on the two key derivatives used for detecting edges is an important issue to keep in mind

Image smoothing is usually a mandatory step prior to the use of derivatives

Marr-Hildreth edge detector

In the 1980, Marr and Hildreth argued that:

- intensity changes are not independent of image scale and so their detection requires the use of operators of different sizes
- a sudden intensity change will give rise to a peak or trough in the first derivative or, equivalently, to a zero crossing in the second derivative
- factors such as image noise and the nature of edges themselves should be taken into account

They proposed to use the Laplacian of Gaussian (LoG) operator instead of Sobel masks

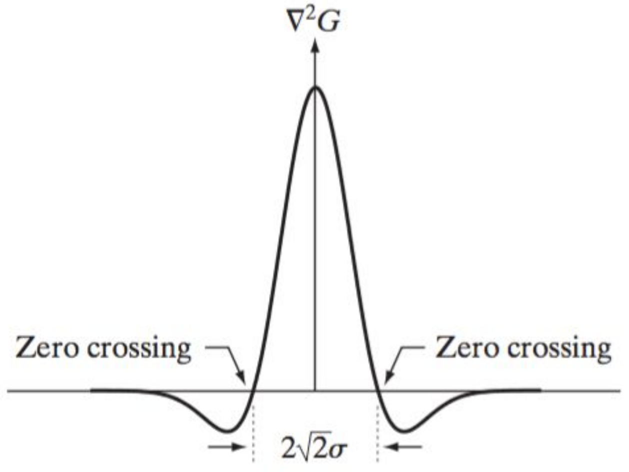
Università
Ca' Foscari
Venezia

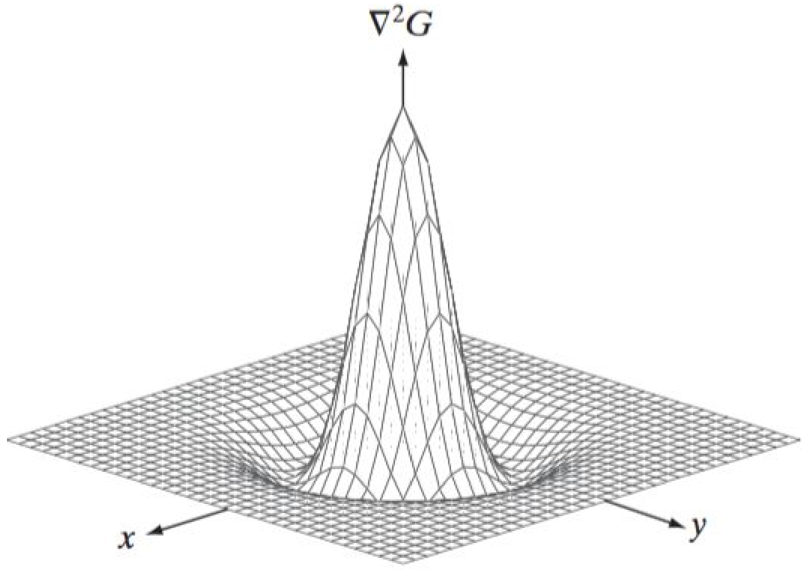
Laplacian of Gaussian

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

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

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





Università Ca' Foscari Venezia

Laplacian of Gaussian

- The Gaussian part of the operator blurs the image, thus reducing the intensity of structures (including noise) at scales much smaller than $\boldsymbol{\sigma}$
  - Gaussian function is smooth in both the spatial and frequency domains and is thus less likely to introduce artifacts (like ringing)
- The Laplacian is isotropic
  - responds equally to changes in intensity in any direction whereas the first derivative is directional and hence require multiple masks
  - corresponds to characteristics of the human visual system

Università Ca' Foscari Venezia

Marr-Hildreth algorithm

1. Convolve the input image with the LoG filter
2. Find the zero-crossing of the convolution to determine the locations of edges in the input image

Since the convolution is a linear operator, we can also divide the steps:

1. Blur the image with a gaussian filter
2. Compute the laplacian of the blurred image (with a laplacian mask)
3. Find the zero crossing

Università Ca' Foscari Venezia

Zero-crossing

How to find the zero crossing?

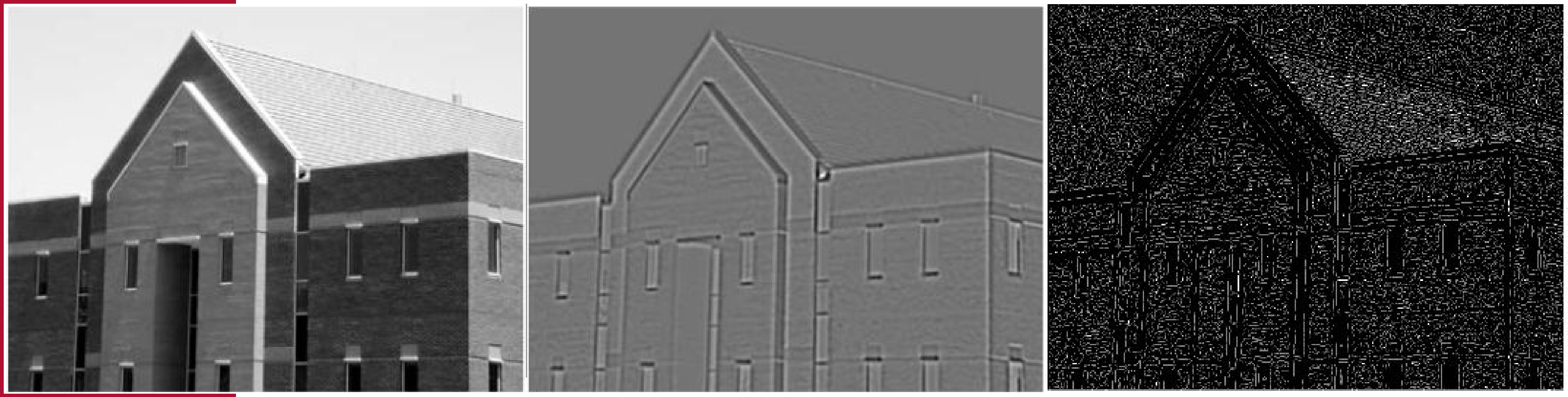
Using a 3x3 neighborhood centered at p. A zero crossing at p implies that **the signs of at least two** of its **opposing neighboring** pixels must differ

4 cases:

- left/right
- up/down
- diagonals

Zero-crossing

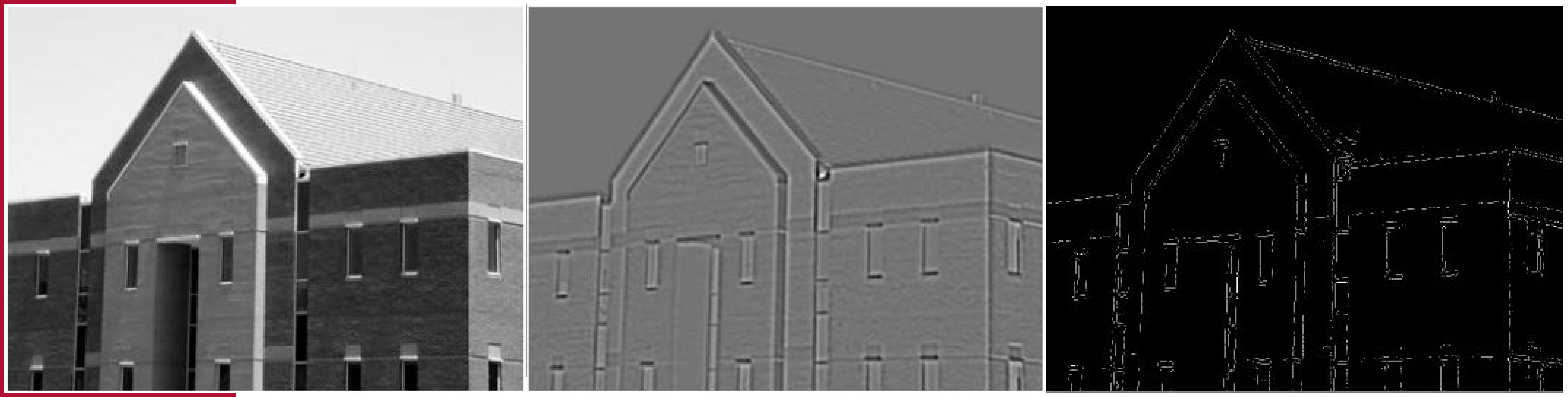
Considering only the sign, the zero-crossing detection is very sensible to noise and creates a lot to closed-loop edges



Università
Ca' Foscari
Venezia

Zero-crossing

Usually we require not only that the signs of opposing neighbors be different, but the absolute value of their numerical difference must exceed a threshold T before we can call p a zero-crossing pixel



T = 4% of the maximum value in the image

Università Ca' Foscari Venezia

Canny Edge Detector

In 1986, Canny tried to find the **optimal** edge detector, assuming a perfect step edge in gaussian noise. Optimal means:

1. *Low error rate:* All edges should be found, and there should be no spurious responses.
2. *Edge points should be well localized:* The edges located must be as close as possible to the true edges
3. *Single edge point response:* the detector should not identify multiple edge pixels where only a single edge point exists.

Canny Edge Detector

The essence of Canny's work was in expressing the preceding three criteria **mathematically** and then attempting to find optimal solutions to these formulations.

Given a filter f, define two objective functions:

$\Lambda(f)$	Large if f produces good **localization**
$\Sigma(f)$	Large if f produces good **detection**

Università Ca' Foscari Venezia

Canny Edge Detector

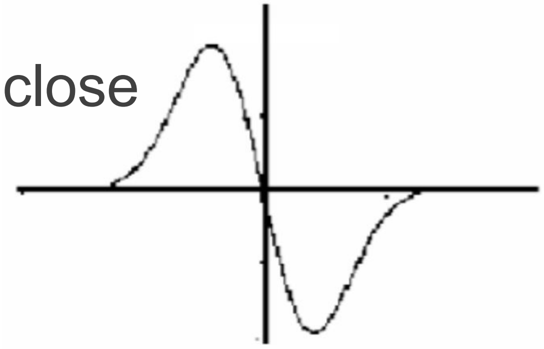
Problem:

Find the best filter f that maximizes the compromise criterion $\Lambda(f)\Sigma(f)$

With the additional constraint that a single peak should be generated at a step-edge

Solution:

The derivative of a gaussian is a very close approximation of this ideal filter



Canny Edge Detector

The derivative of a Gaussian is the optimal filter in the 1D case of a step-edge corrupted by noise

- Generalizing this result to 2D involves recognizing that the 1D approach *still applies in the direction of the edge normal* (ie. the derivative of gaussian should be applied in all possible directions)

Since the derivative is also a linear operator, the 2D detector can be implemented as follows:

1. Smooth the image with a 2D gaussian filter
2. Compute the gradient of the result

Università Ca' Foscari Venezia

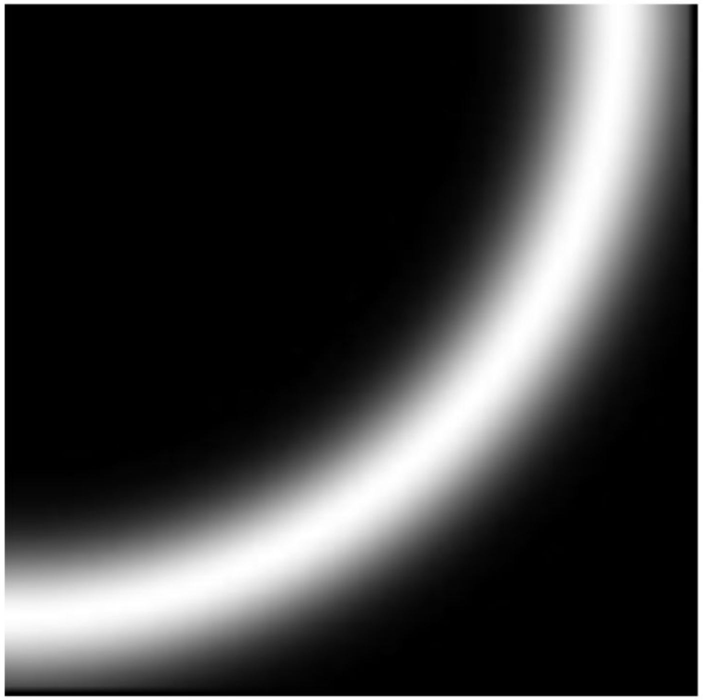
Canny Edge Detector

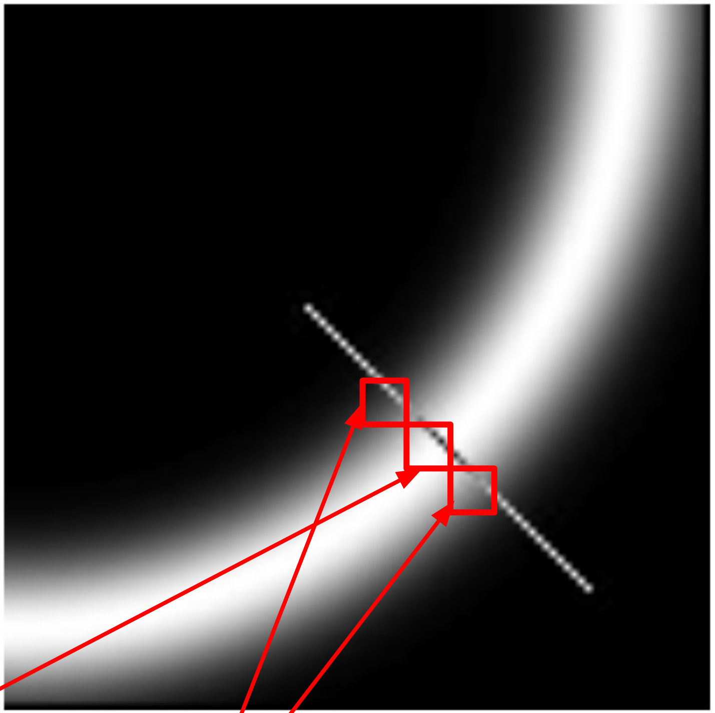
Computing the gradient magnitude with derivative of gaussian produces good results but it typically contains **wide ridges around local maxima.**

To thin the ridges Canny proposed a *non-maxima suppression* approach.

IDEA: Select the single maximum point across the width of an edge.

Non-maxima suppression

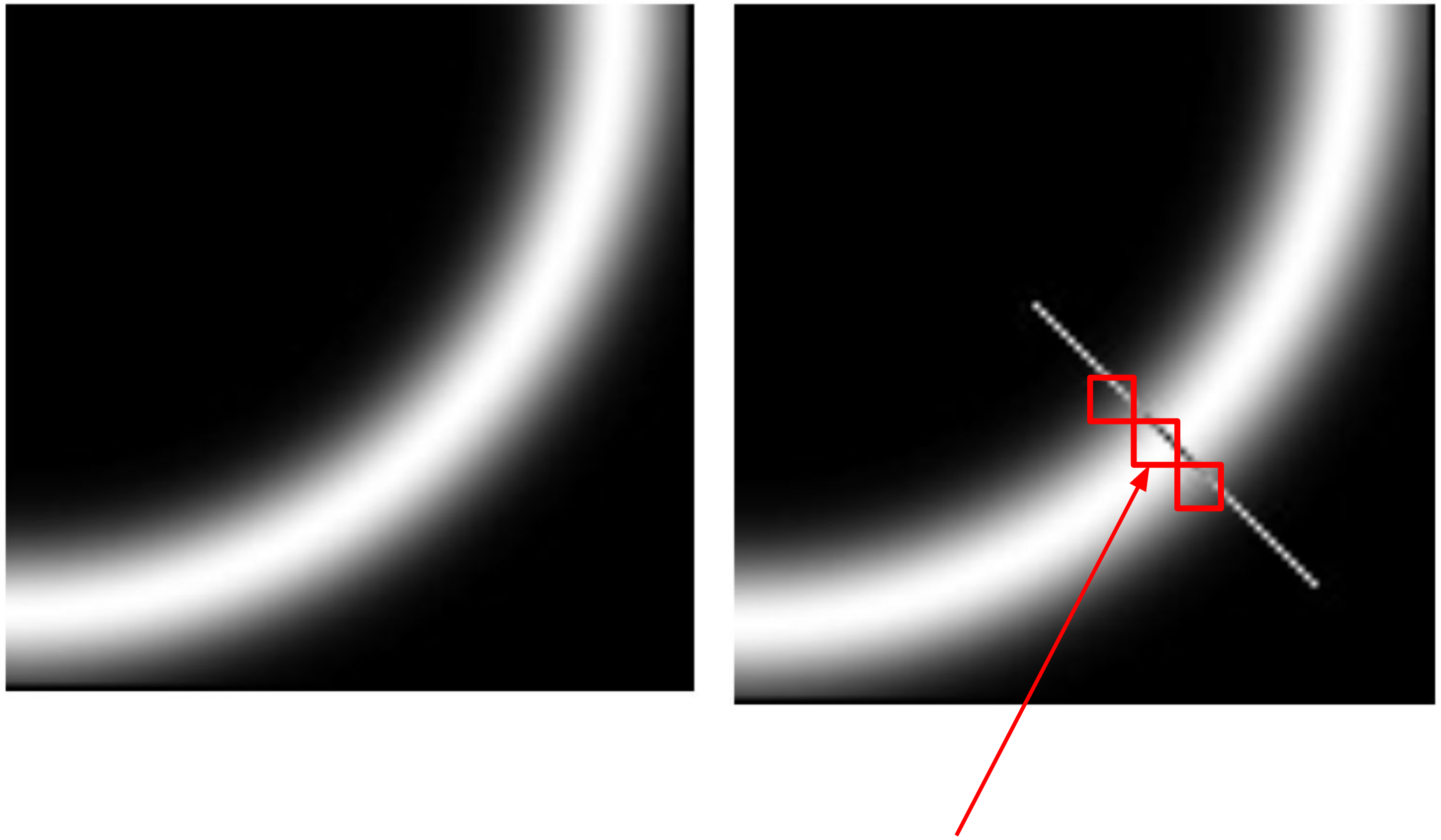




For each pixel check the neighbours depending on the direction of the gradient

Università
Ca' Foscari
Venezia

Non-maxima suppression



If the intensity of the gradient of a pixel is less than at least one of the two neighbours along the gradient, then it must be suppressed (ie. gradient magnitude set to zero)

Università Ca' Foscari Venezia

Hysteresis thresholding

The final operation is to threshold the non-maxima suppressed gradient image to reduce false edge points.

With a single threshold we have the following tradeoff:

- **Threshold too low:** there will still be some false edges (called false positives).
- **Threshold too high:** actual valid edge points will be eliminated (false negatives).

Canny's algorithm attempts to improve on this situation by using **hysteresis thresholding**

Università Ca' Foscari Venezia

Hysteresis thresholding

We use two thresholds: T_L and T_H, with $T_L < T_H$.

Pixels in T_H are called strong edge pixels and the ones in T_L are called weak edges.

The idea is to follow strong edges until they fall below T_L (ie they can be connected using also weak edges)

1. Locate the next unvisited strong edge p
2. Mark as valid all weak edges connected to p (8-neigh)
3. If there still exist unvisited edges go to step 1
4. Mark all the remaining unvisited pixels as invalid and remove then from the final edges

Università
Ca' Foscari
Venezia

Canny Edge Detector

Summary of the Canny edge detector:

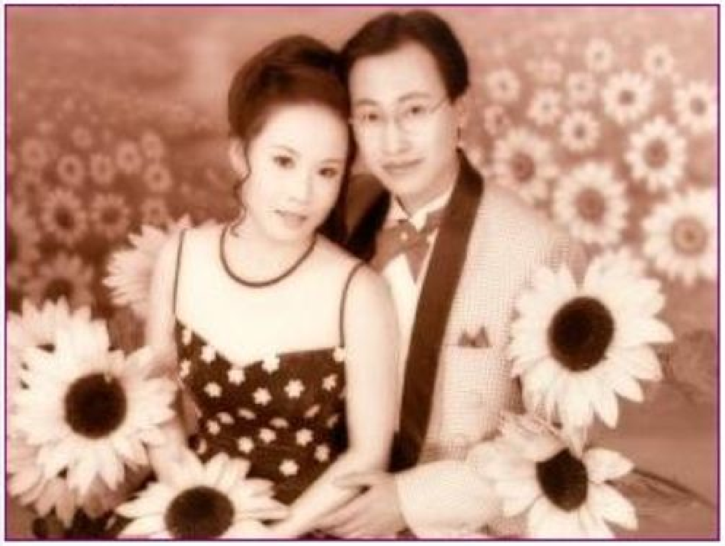
1. Smooth the input image with a Gaussian filter.
2. Compute the gradient magnitude and angle.
3. Apply non-maxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.

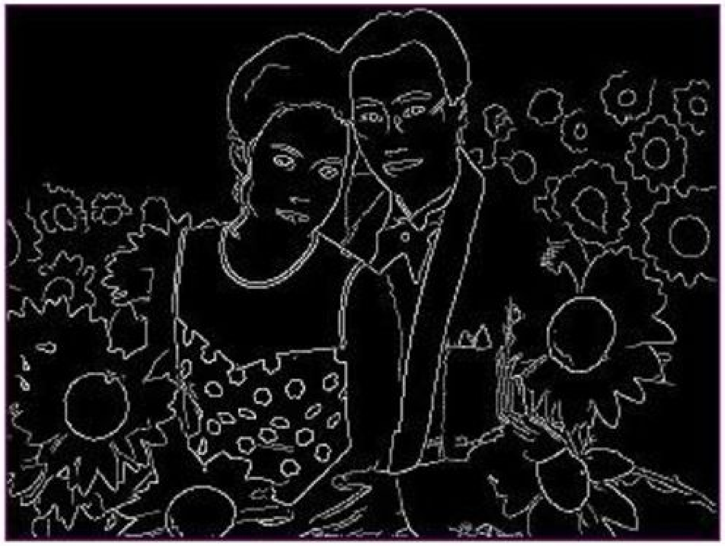
Canny is still the-facto standard after 20 years, why?

- Good theoretical background
- Hysteresis and edge linking are an important heuristic

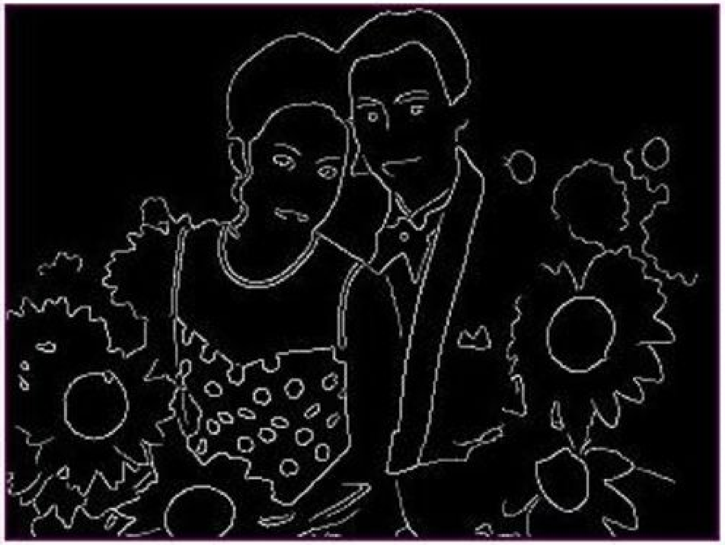
Università
Ca' Foscari
Venezia

Canny Edge Detector

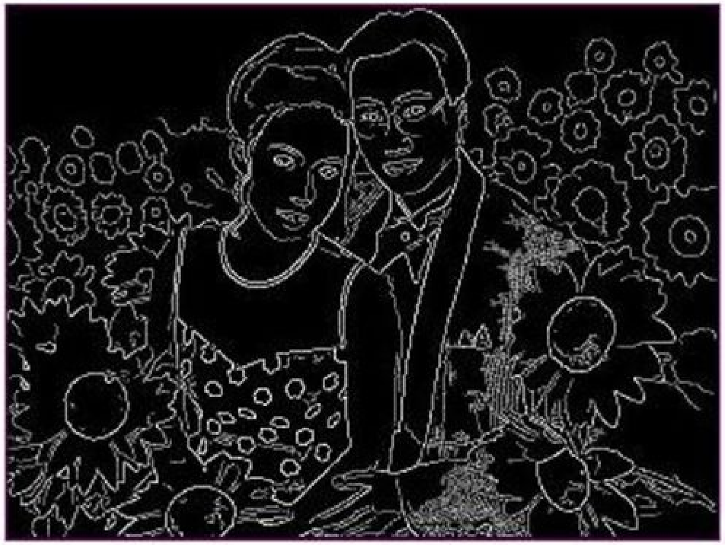




Sigma= 1, Low= 0.4, High= 0.8



Sigma= 2, Low= 0.4, High= 0.8



Sigma= 1, Low= 0.3, High= 0.7