Segmentation

Image Segmentation

Segmentation divides an image into its constituent regions or objects.

Segmentation of images is a difficult task in image processing. Still under research.

Segmentation allows to extract objects in images.

Segmentation examples

- *unsupervised*: background subtraction, recognition
- *supervised*: photo-shop, medical image analysis

What it is useful for

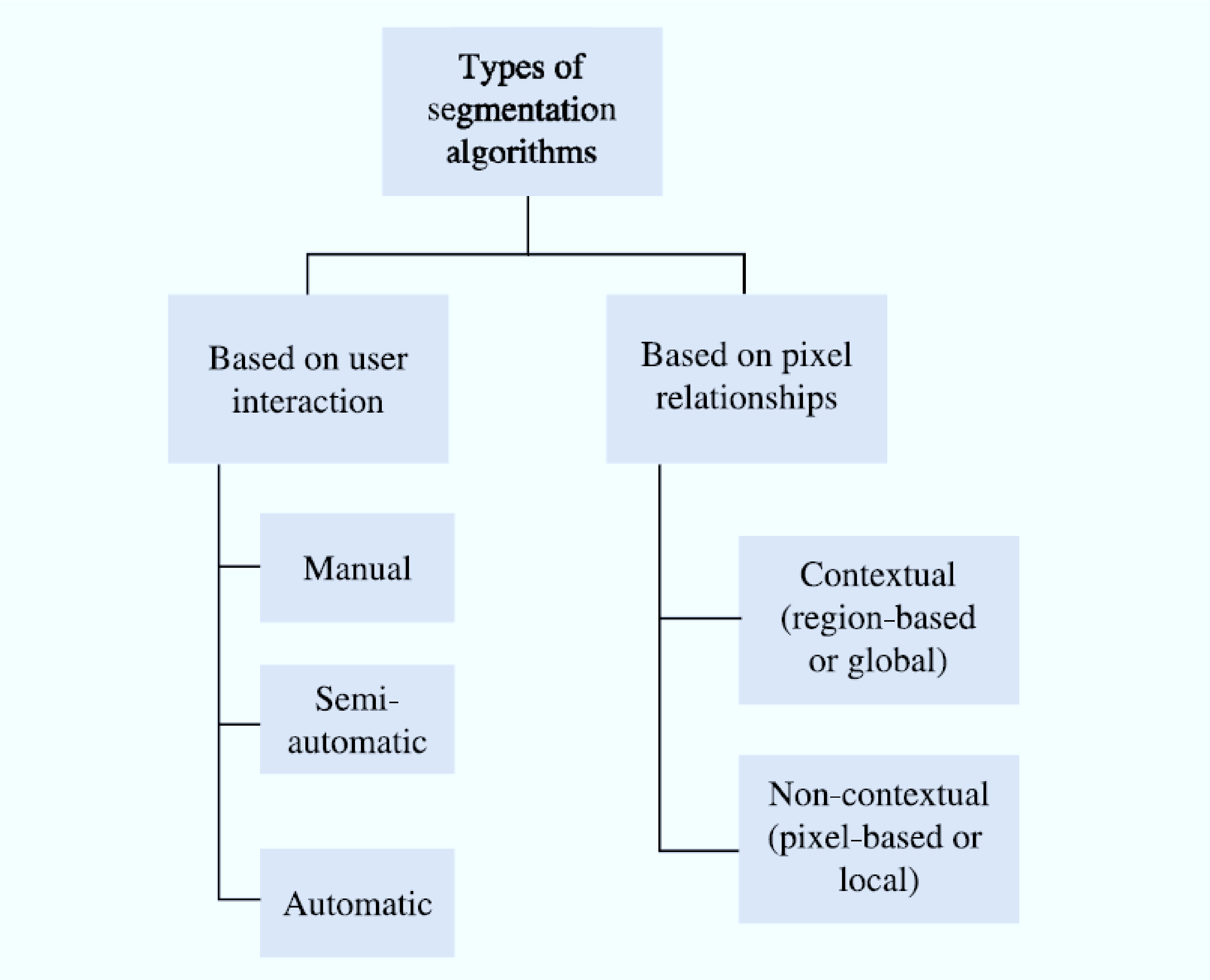
After a *successful* segmenting the image, the contours of objects can be extracted using edge detection and/or border following techniques.

Shape of objects can be described.

Based on shape, texture, and color objects can be identified.

Image segmentation techniques are extensively used in similarity searches

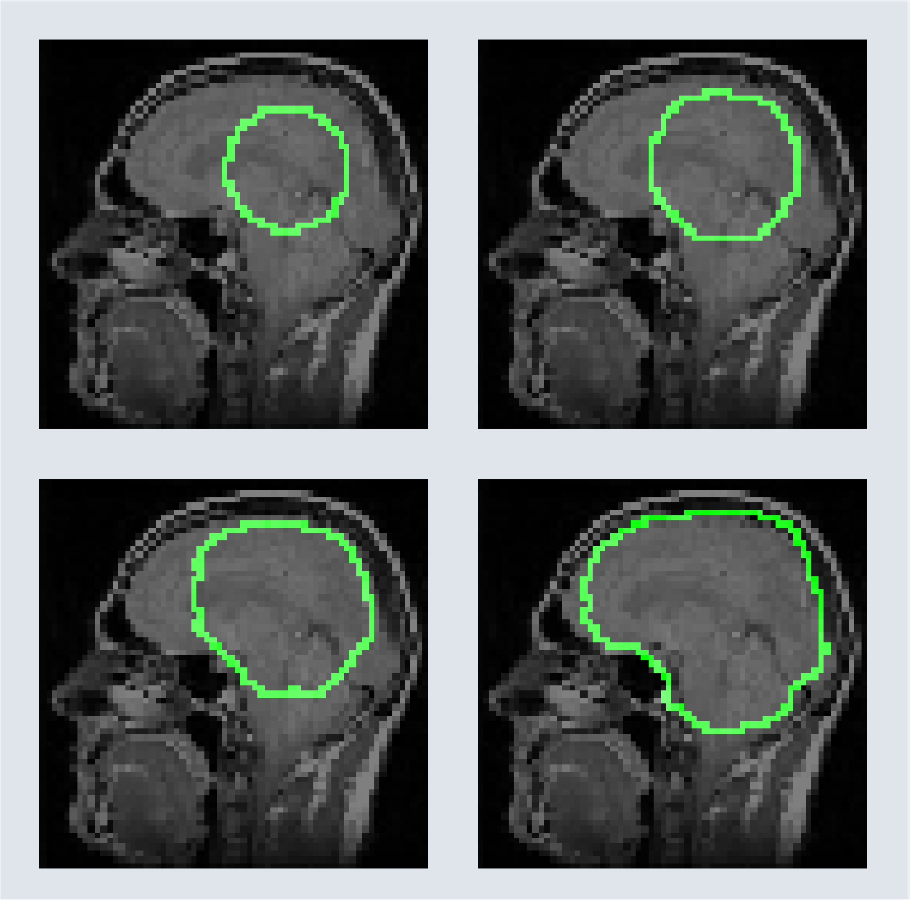
- Classification of image segmentation Techniques
  - Local Segmentation (Non-contextual)
  - Global Segmentation (Contextual)
- Region based
- Boundary-based
- Edge Based


Types of segmentation algorithms
Based on user interaction
Based on pixel relationships
Manual
Semi-automatic
Automatic
Contextual (region-based or global)
Non-contextual (pixel-based or local)

Medical Image Segmentation

Medical image analysis can be used as preliminary screening techniques to help doctors

Partial Differential Equation (PDE) has been used for segmenting medical images



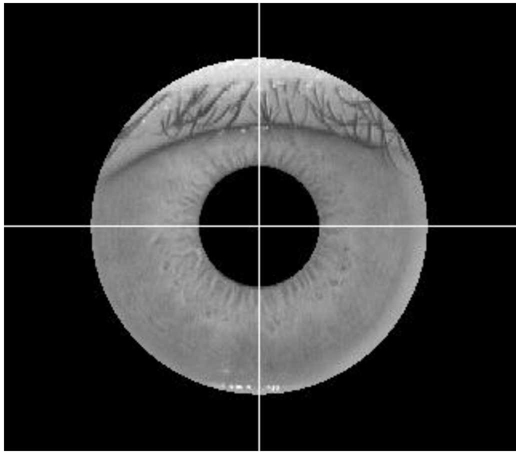
active contour model (snake)

Biometric Image Segmentation

For fingerprint, face and iris images, we also need to segment out the region of interest

Various cues can be used such as ridge pattern, skin color and pupil shape

Robust segmentation could be difficult for poor-quality images

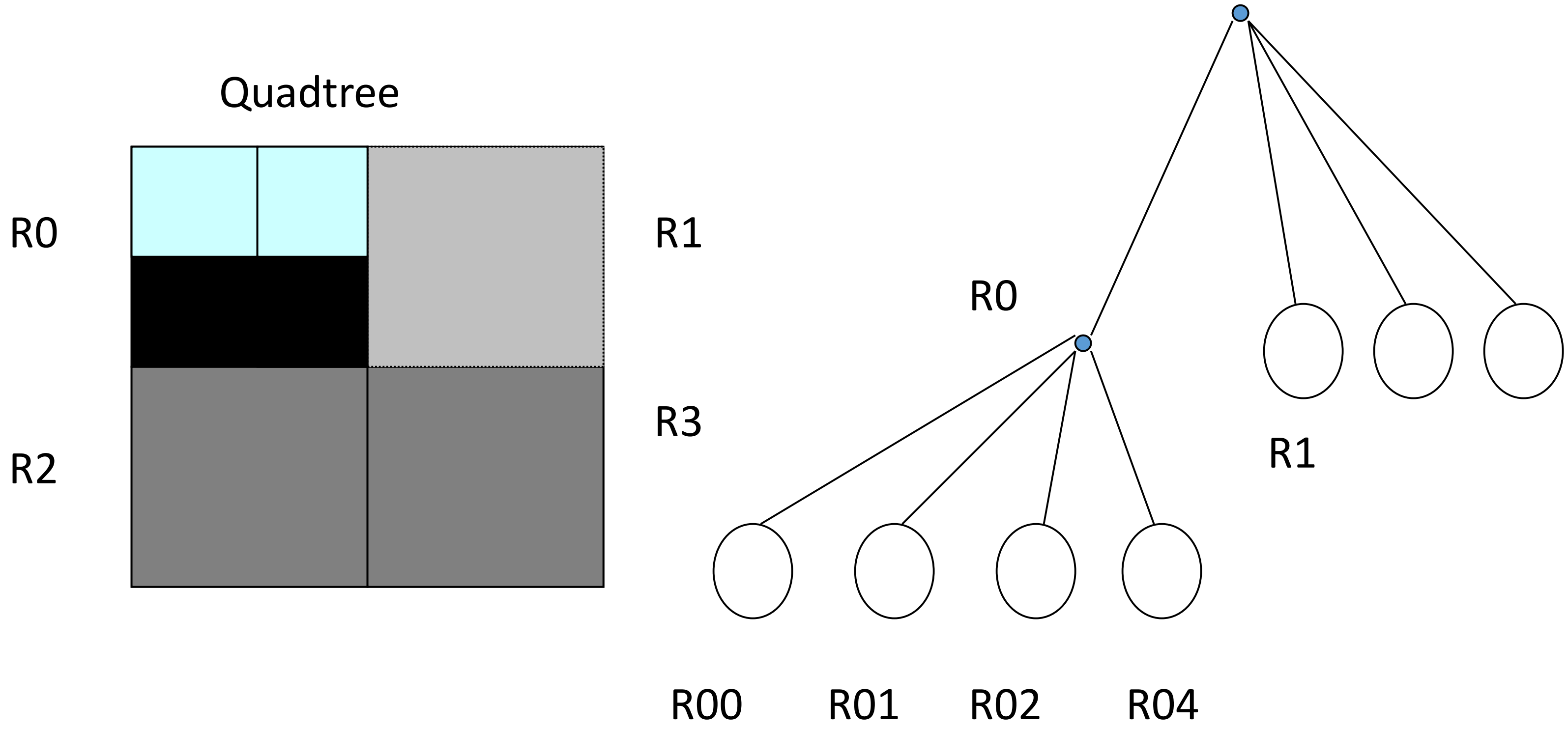


1. If the subregions are combined, the original region can be obtained. Mathematically, it can be stated that $\bigcup R_i = R$ for $i = 1, 2, \ldots, n$. For example, if the three regions of Fig. 7.1(c) R_1, R_2, and R_3 are combined, the whole region R is obtained.
2. The subregions R_i should be connected. In other words, the region cannot be open-ended during the tracing process.
3. The regions $R_1, R_2, \ldots, R_n$ do not share any common property. Mathematically, it can be stated as $R_i \bigcap R_j = \varphi$ for all i and j where $i \neq j$. Otherwise, there is no justification for the region to exist separately.

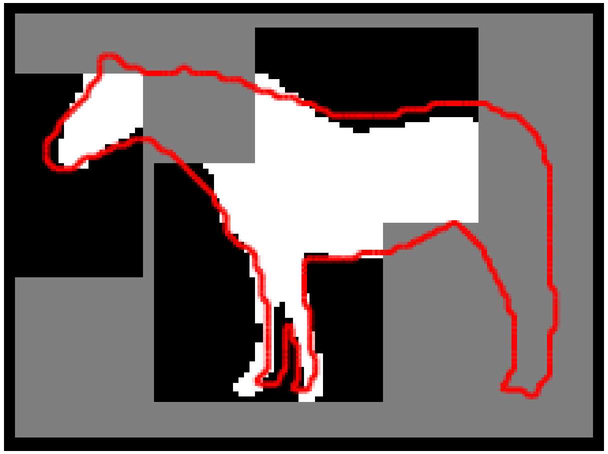
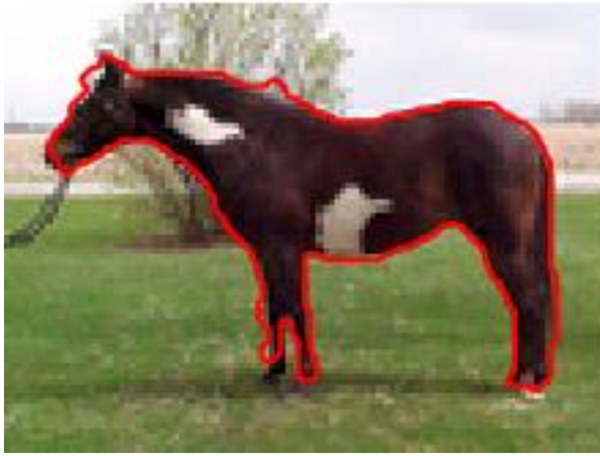
4. Each region satisfies a predicate or a set of predicates such as intensity or other image statistics, that is, the predicate (P) can be colour, grey scale value, texture, or any other image statistic. Mathematically, this is stated as $P(R_i)$ = True.

Region Based approach

- Region Growing
- Region splitting
- Region Splitting and Merging


Quadtree
R0
R1
R2
R3
R0
R1
R00
R01
R02
R04

Top-down segmentation

Detection of Discontinuities

- 3 types of image features based on sharp local changes in intensity:
  1. Edges: Sets of connected edge pixels.(Edge pixels:- Region where intensity of an image function changes abruptly.)
  2. Line: Edge segment where the intensity of the background on either side of the line is much higher or much lower than the intensity of the line pixels
  3. Isolated Point : A line whose length and width are equal to one pixel.

Approach

Using derivatives (particularly, 1^{st} and 2^{nd}) of images we find local changes in intensity

- *First derivative*
  - $\partial f/\partial x = f'(x)$ = f(x+1) – f(x) [by Taylor Series]
  - Approximation :-

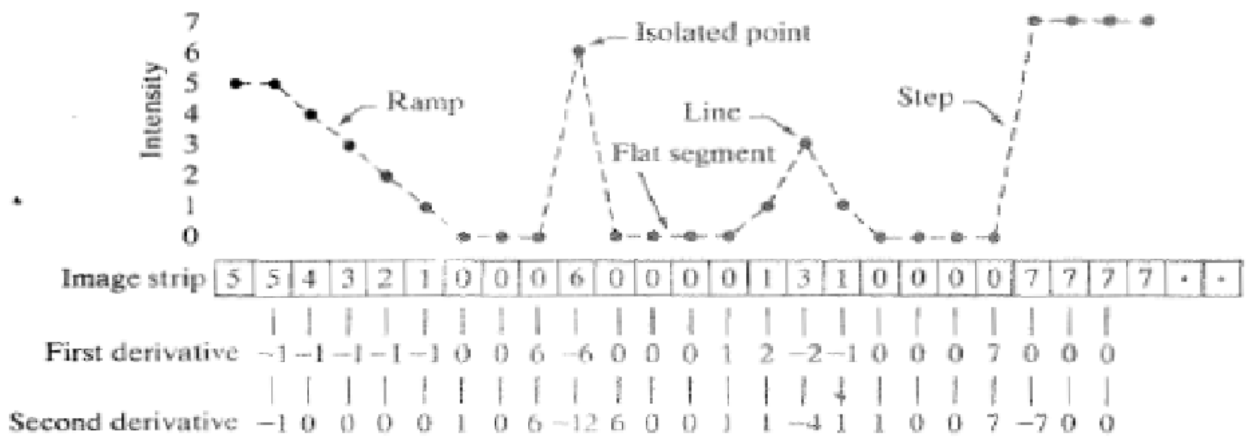
Output	Inference
Zero	Areas of constant intensity
Non Zero	Onset of intensity step or ramp and points along an intensity ramp

- *Second derivative*

$$\partial^2 f/\partial x^2 = f''(x) = f(x+1) + f(x-1) - 2f(x)$$

Output	**Inference**
Zero	Areas of constant intensity
Non Zero	Onset and end of intensity step or ramp and along an intensity ramp

- Compute 1^{st} or 2^{nd} derivative at every pixel location in an image by spatial filters



Observation	Inference
Along the ramp first derivative is non zero and second derivative is non zero at the onset and end of ramp.	First order derivative produces thick edges whereas second order derivative produces much finer ones.
At isolated noise point sod has much more response than fod.	SOD have strong response to fine details like noise, isolated points etc.
Ramp and step edges sod has opposite signs. (-1 to 1 and 7 to -7)	SOD produce a double edge response at ramp and step transitions in intensity and it's sign determines edge transition from dark to light or vice versa.

Isolated Points Detection

- Use second order derivative
- $\nabla^2 f(x, y) = \partial^2 f/\partial x^2 + \partial^2 f/\partial y^2$ (Laplacian)

  we know, $\partial^2 f/\partial x^2$ = f(x+1,y) + f(x-1,y) – 2f(x, y)

  and $\partial^2 f/\partial y^2$ = f(x,y+1) + f(x,y-1) – 2f(x, y)

  so, $\nabla^2 f(x, y)$ = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x, y)
- Implemented with mask:

1	1	1
1	-8	1
1	1	1

OR

0	1	0
1	-4	1
0	1	0

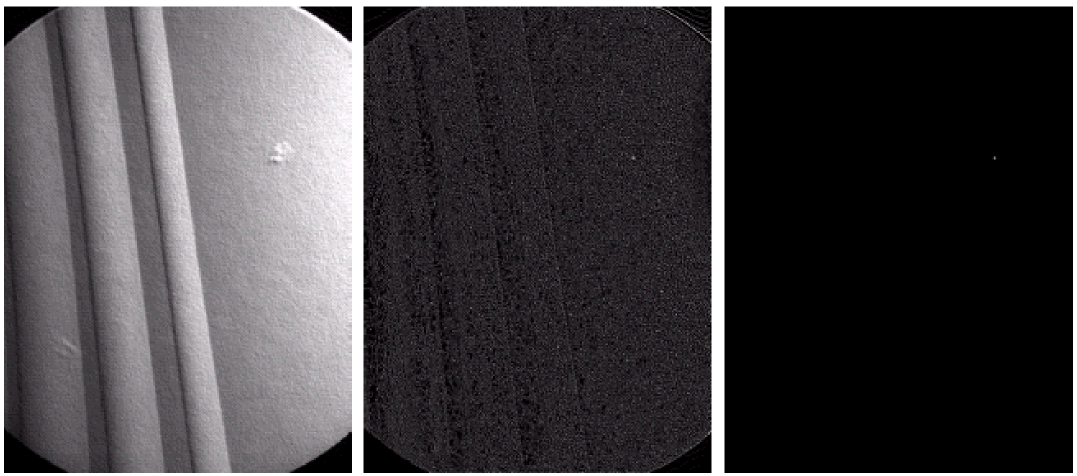
- Point output is obtained by

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

where, T = non-negative threshold

$R = \sum_{k=1}^{9} \omega_k z_k$ (ω = filter coff., z = image pixels intensity)

–1	–1	–1
–1	8	–1
–1	–1	–1



a
b c d

FIGURE 10.2
(a) Point detection mask.
(b) X-ray image of a turbine blade with a porosity.
(c) Result of point detection.
(d) Result of using Eq. (10.1-2).
(Original image courtesy of X-TEK Systems Ltd.)

Line Detection

- Use second order derivatives
- They produce stronger response and thinner lines as compared to first order derivatives
- For detecting lines in specific direction:

-1	-1	-1
2	2	2
-1	-1	-1

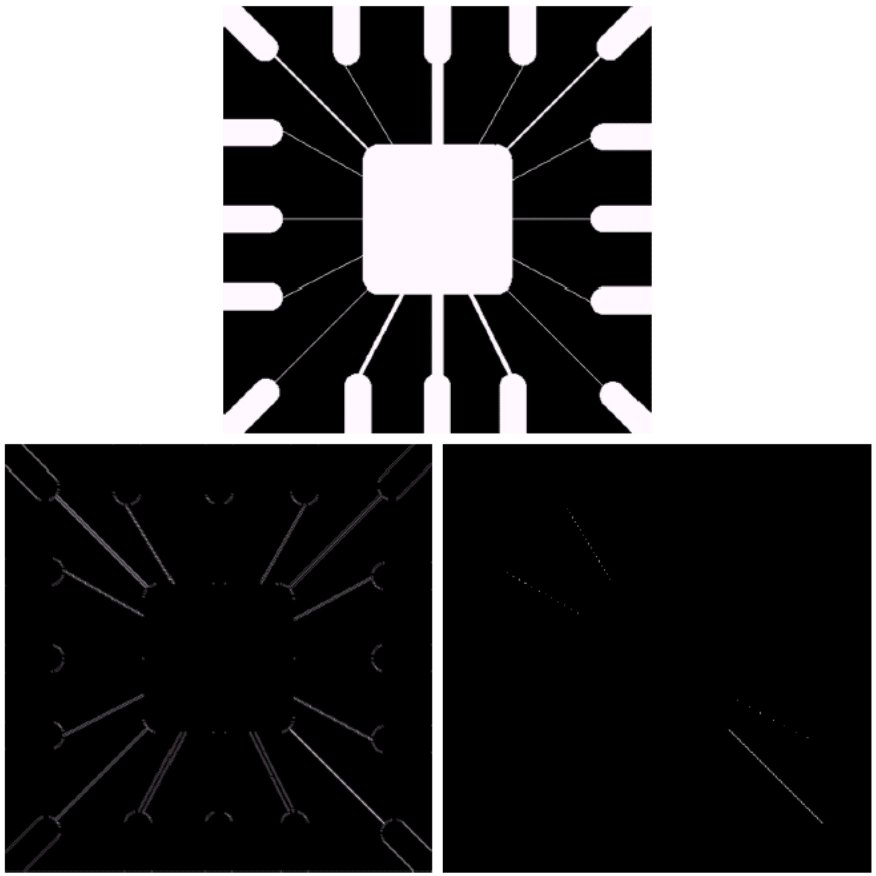
-1	2	-1
-1	2	-1
-1	2	-1

2	-1	-1
-1	2	-1
-1	-1	2

-1	-1	2
-1	2	-1
2	-1	-1

Horizontal Vertical +45
-45

- Run the mask through the image
  Threshold the absolute value of the result
  output lines are closest to the direction defined by the mask



a
b c

FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with −45° line detector. (c) Result of thresholding image (b).

Edge detection

- Edge Models:
  - Step Edge
  - Ramp Edge
  - Roof Edge

- Magnitude of first derivative is used to detect the presence of an edge
- Sign of second derivative can be used to determine whether an edge pixel lies on the dark or light side
- Second derivative:
  - produces two values for every edge in an image
  - Zero crossing to locate mid point of thick edges
- 3 fundamental steps:
  - Image Smoothing
  - Edge points detection
  - Edge localization

Detection of Discontinuities
Gradient Operators

- First-order derivatives:
  - The gradient of an image $f(x,y)$ at location (x,y) is defined as the vector:

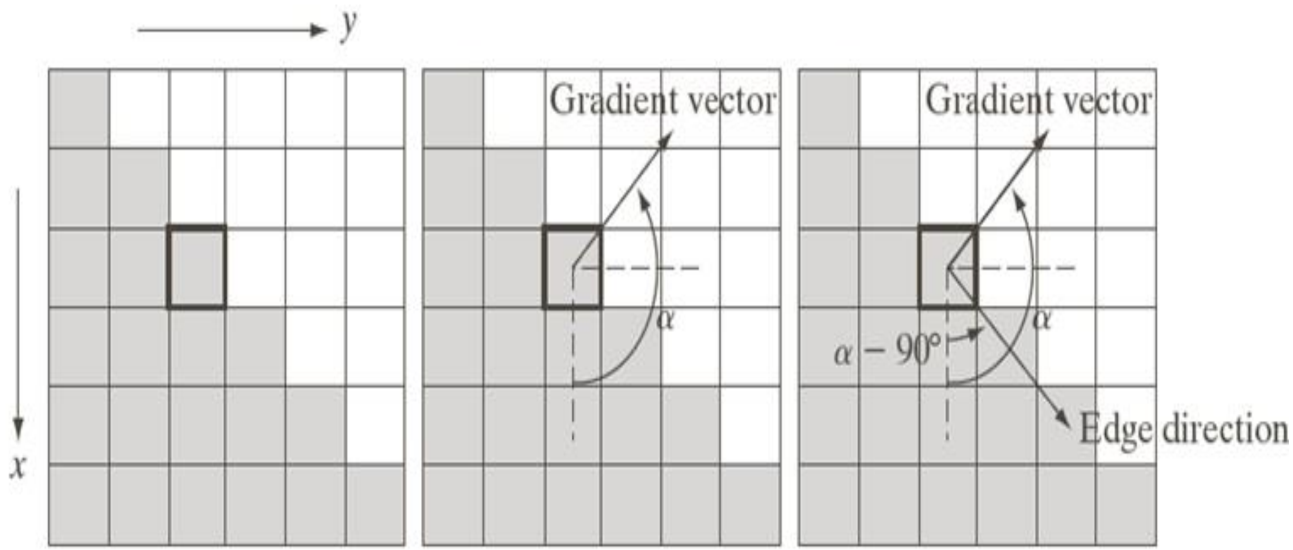
$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

  - The magnitude of this vector:

$$\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$$

  - The direction of this vector:

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$$



a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Gradient Operators

Roberts cross-gradient operators

−1	0	0	−1
0	1	1	0

Roberts

Prewitt operators

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

Prewitt

Sobel operators

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

Sobel

Prewitt masks for
detecting diagonal edges

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

Sobel masks for
detecting diagonal edges

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

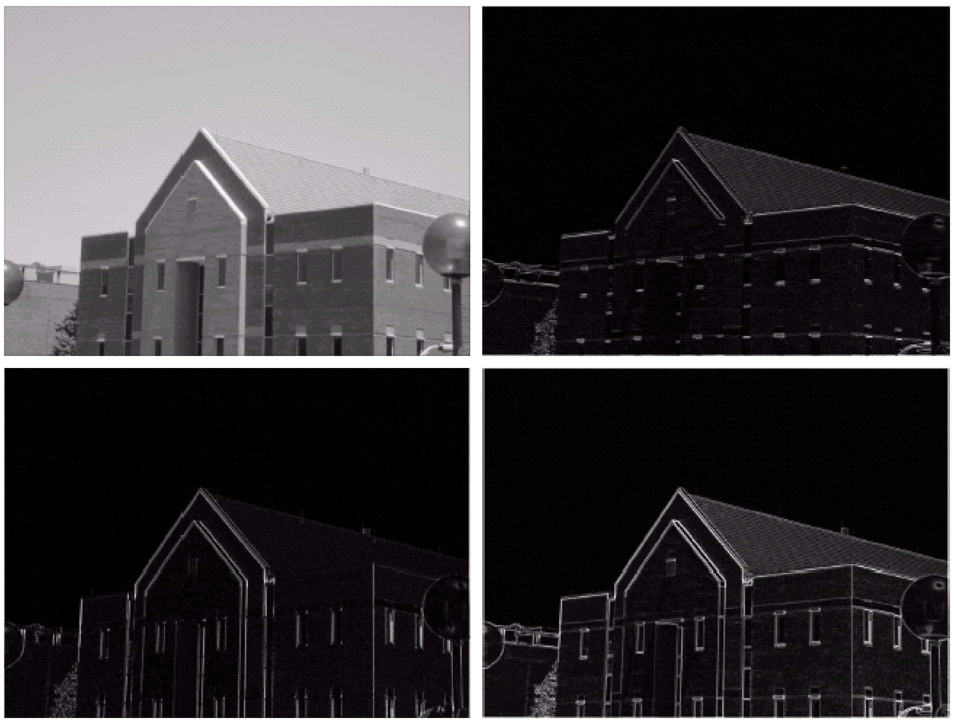
Sobel

a	b
c	d

FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.



a b
c d

FIGURE 10.11 Same sequence as in Fig. 10.10, but with the original image smoothed with a 5×5 averaging filter.

a b

FIGURE 10.12 Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

0	1	2
−1	0	1
−2	−1	0

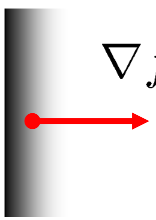
−2	−1	0
−1	0	1
0	1	2

Image gradient

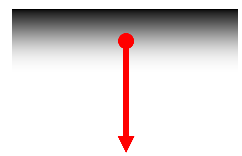
- The gradient of an image:

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

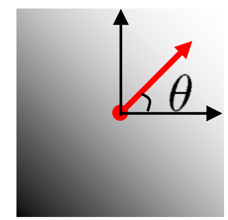
- The gradient points in the direction of most rapid change in intensity



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



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



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

The gradient direction is given by:

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

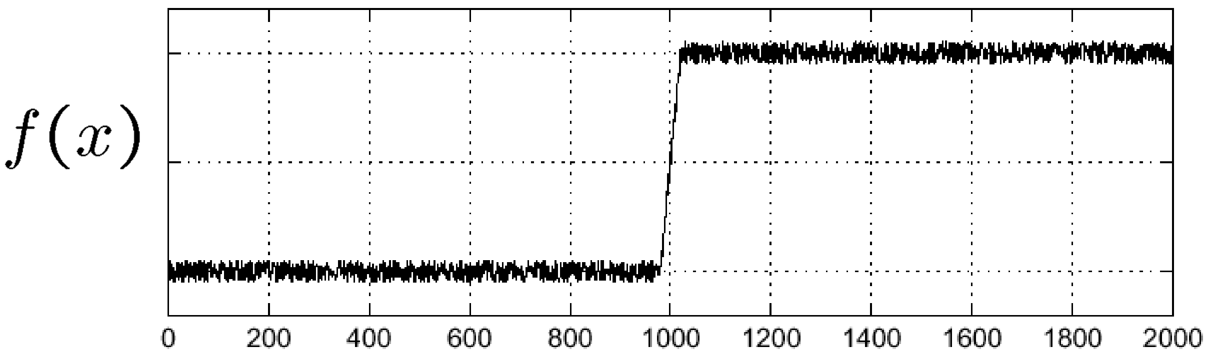
- how does this relate to the direction of the edge?

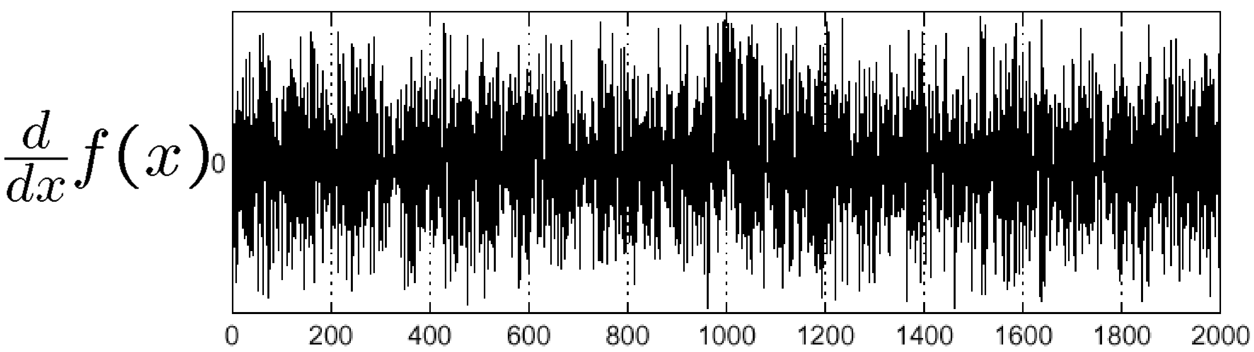
The *edge strength* is given by the gradient magnitude

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

Effects of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal





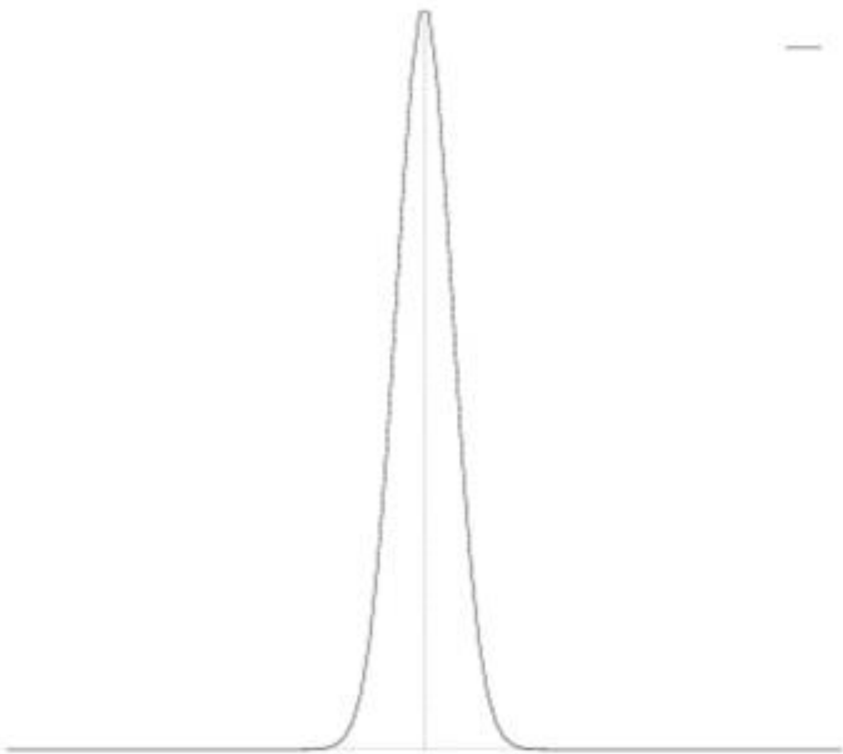
Where is the edge?

Gaussian

$$g(x) = e^{\frac{-x^2}{2\sigma^2}}$$

Standard deviation

x	-3	-2	-1	0	1	2	3
g(x)	.011	.13	.6	1	.6	.13	.011



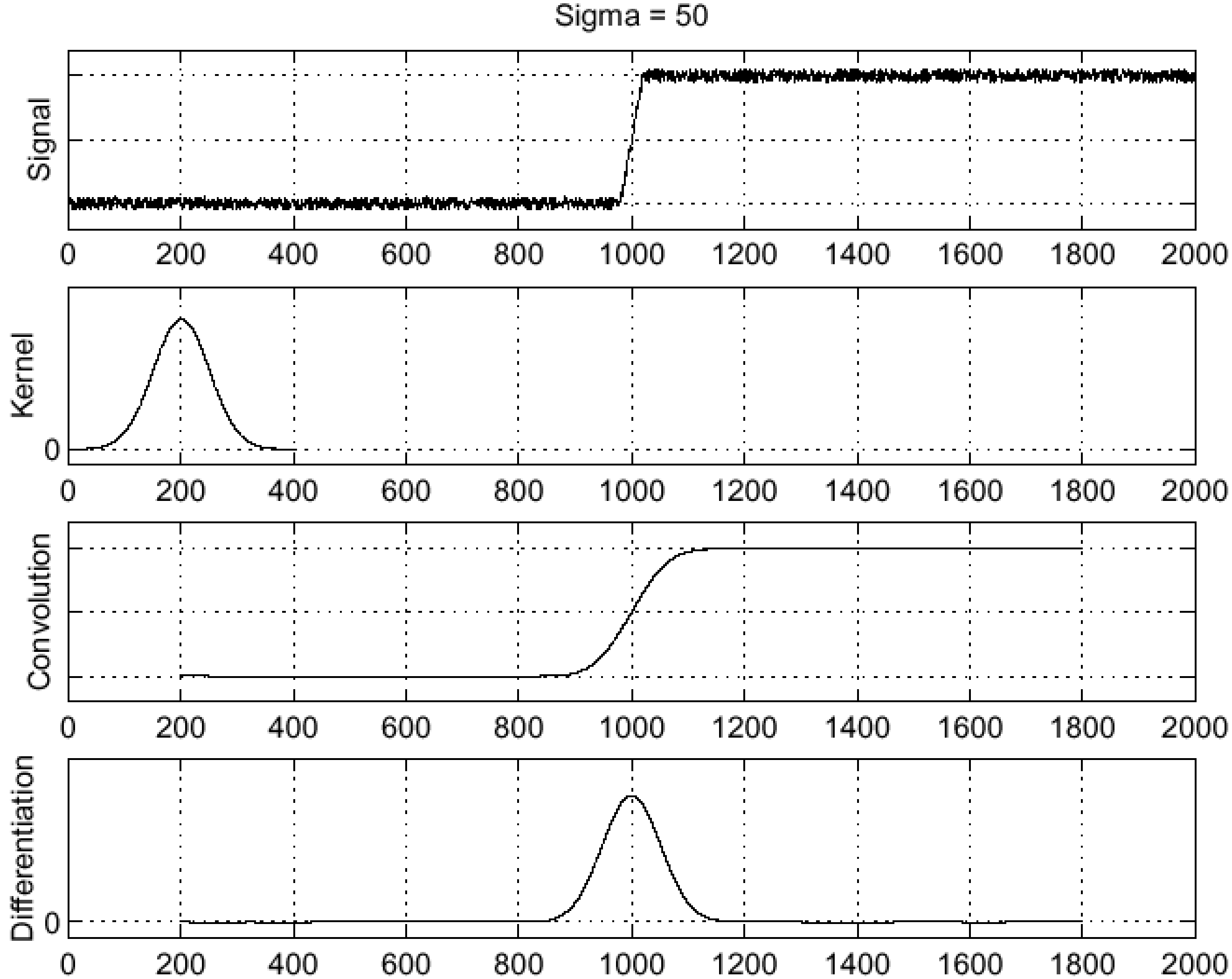
Solution: smooth first

f

h

$h \star f$

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



Where is the edge?   Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative theorem of convolution

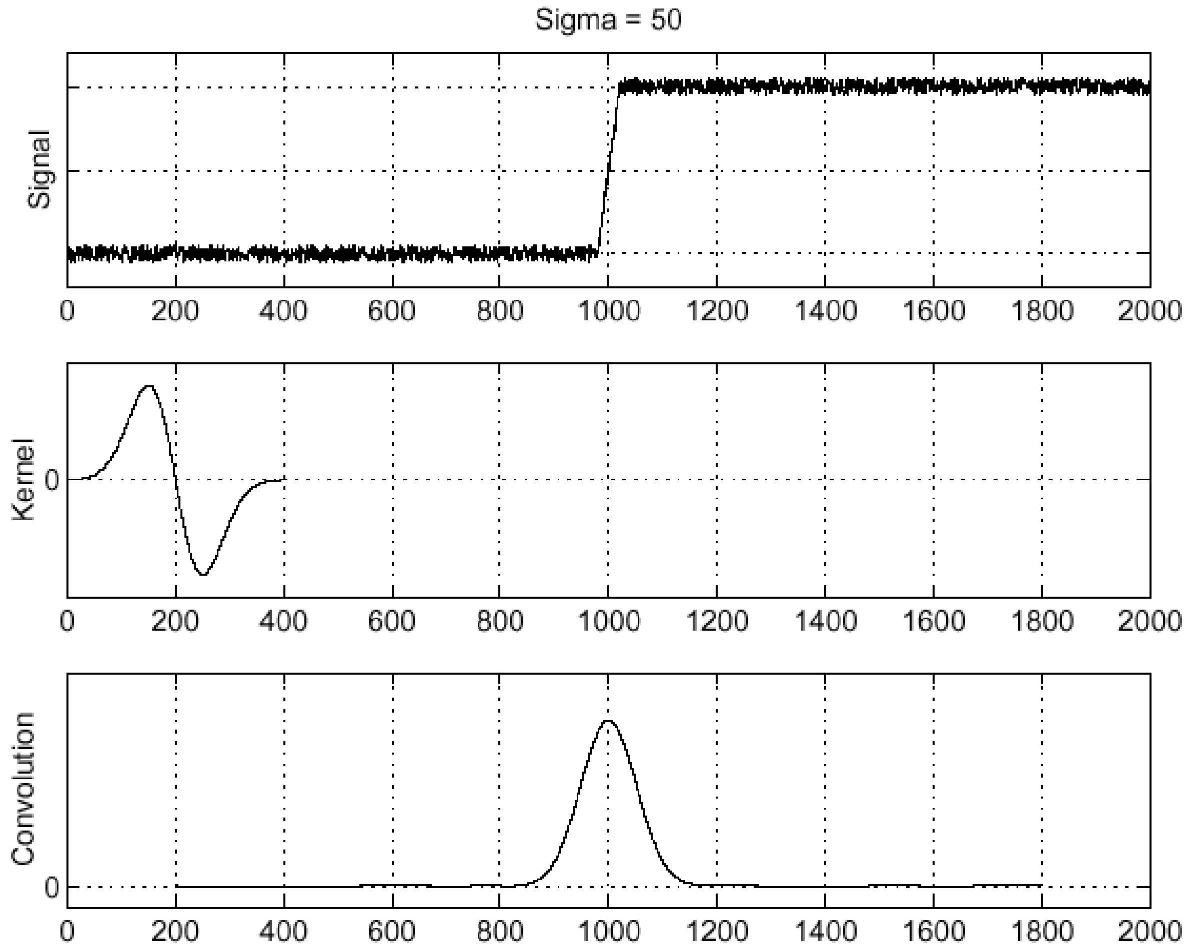
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- This saves us one operation:

f

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

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



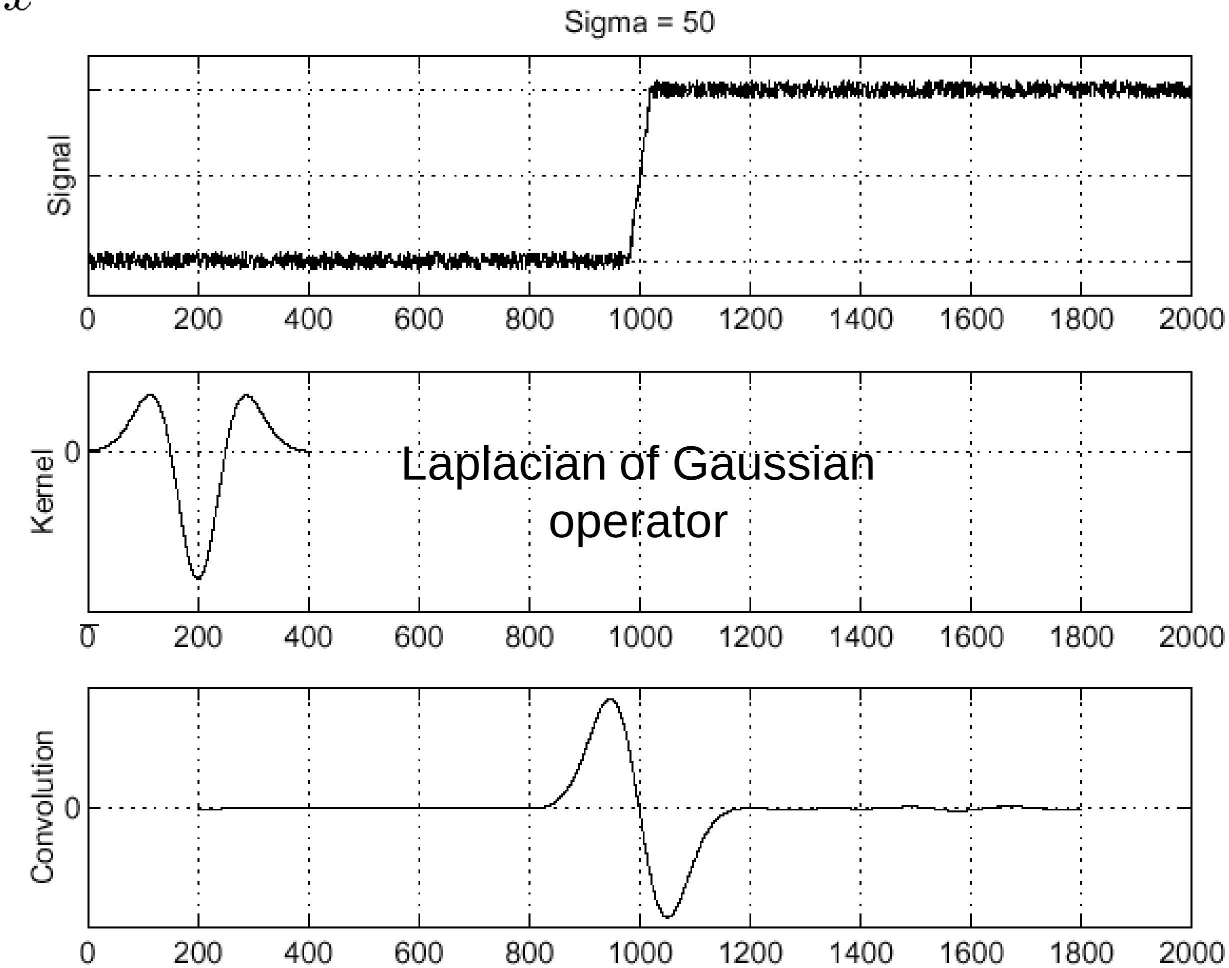
Marr and Hildreth Edge Operator **(Laplacian of Gaussian)**

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

$$f$$

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

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



Where is the edge?   Zero-crossings of bottom graph

Marr Hildreth Edge Detector

- Smooth image by Gaussian filter S
- Apply Laplacian to S
- Find zero crossings
- Scan along each row, record an edge point at the location of zero-crossing.
- Repeat above step along each column

Marr and Hildreth Edge Operator

- Smooth by Gaussian

$$S = G_\sigma * I$$

$$G_\sigma = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

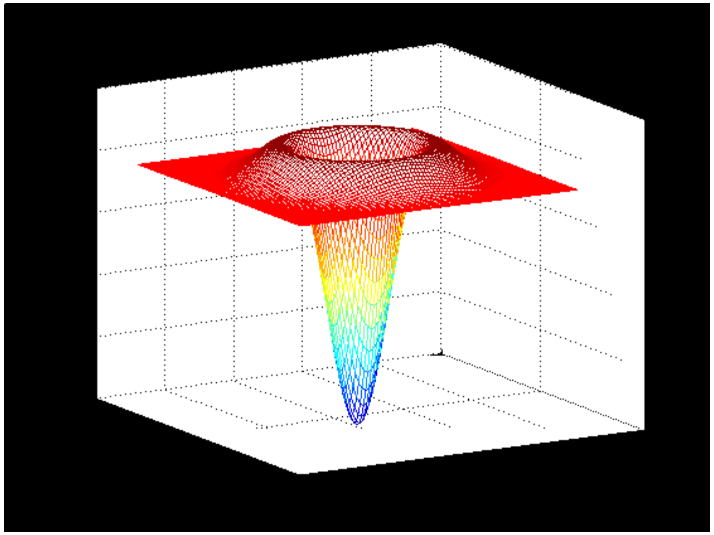
- Use Laplacian to find derivatives

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

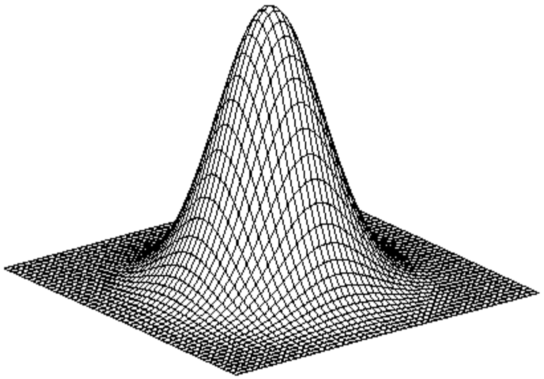
Marr and Hildreth Edge Operator

$$\nabla^2 S = \nabla^2 \left(G_\sigma * I \right) = \nabla^2 G_\sigma * I$$

$$\nabla^2 G_\sigma = -\frac{1}{\sqrt{2\pi}\sigma^3} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

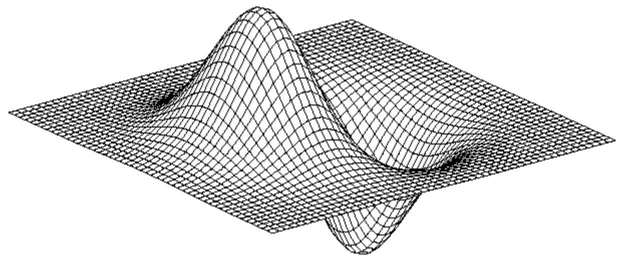


2D edge detection filters



Gaussian

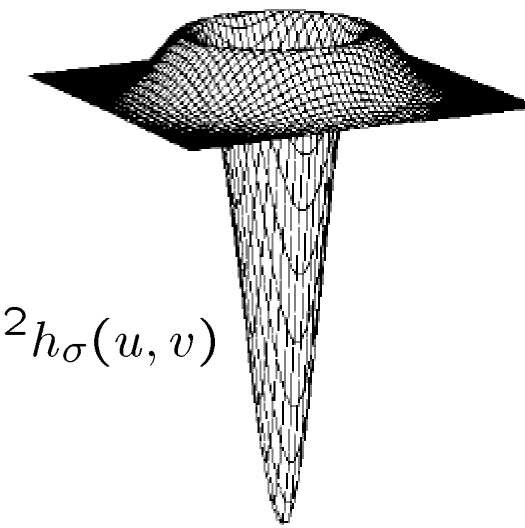
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

∇^2 is the **Laplacian** operator:

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

Marr and Hildreth Edge Operator

$$\nabla^2 G_\sigma = -\frac{1}{\sqrt{2\pi}\,\sigma^3}\left(2 - \frac{x^2 + y^2}{\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Y

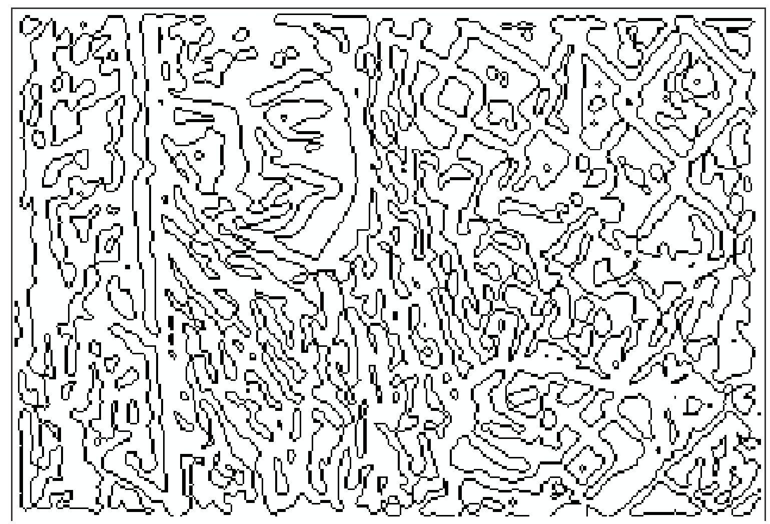
0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.031	0.108	-0.242	-0.7979	-0.242	0.108	0.031
0.0215	0.0982	0	-0.242	0	0.0982	0.0215
0.0066	0.0438	0.0982	0.108	0.0982	0.0438	0.0066
0.0008	0.0066	0.0215	0.031	0.0215	0.0066	0.0008

X

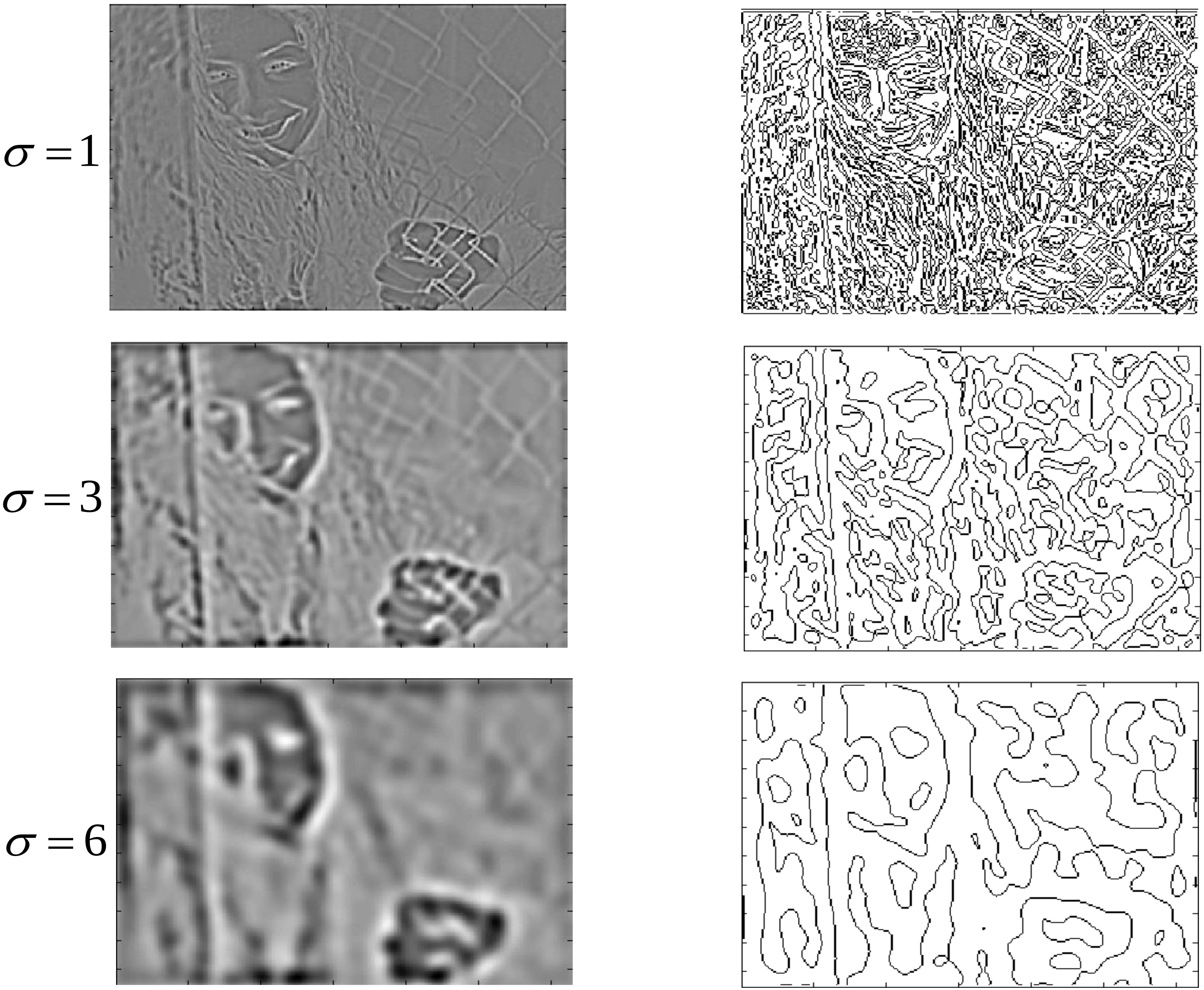
Marr and Hildreth Edge Operator

Image I → $*\nabla^2 G_\sigma$ → $\nabla^2 G_\sigma * I$ → Zero Crossings Detection → Edge Image

$\nabla^2 G_\sigma * I$



Zero Crossings


$\sigma = 1$
$\sigma = 3$
$\sigma = 6$

Finding Zero Crossings

- Four cases of zero-crossings : –
  - {+,-}
  - {+,0,-}
  - {-,+}
  - {-,0,+}
- Slope of zero-crossing {a, -b} is |a+b|.
- To mark an edge
  - compute slope of zero-crossing
  - Apply a threshold to slope

Algorithm

- Compute LoG
- Find zero-crossings from each row
- Find slope of zero-crossings
- Apply threshold to slope and mark edges

Canny Edge Detector

- Criterion 1:
  - Good Detection: The optimal detector must minimize the probability of false positives as well as false negatives.
- Criterion 2:
  - Good Localization: The edges detected must be as close as possible to the true edges.
- Single Response Constraint: The detector must return one point only for each edge point.

Canny Edge Detector Steps

1. Smooth image with Gaussian filter
2. Compute derivative of filtered image
3. Find magnitude and orientation of gradient
4. Apply “Non-maximum Suppression”
5. Apply “Hysteresis Threshold”

Canny Edge Detector

- Smooth by Gaussian

$$S = G_\sigma * I \qquad G_\sigma = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Compute *x* and *y* derivatives

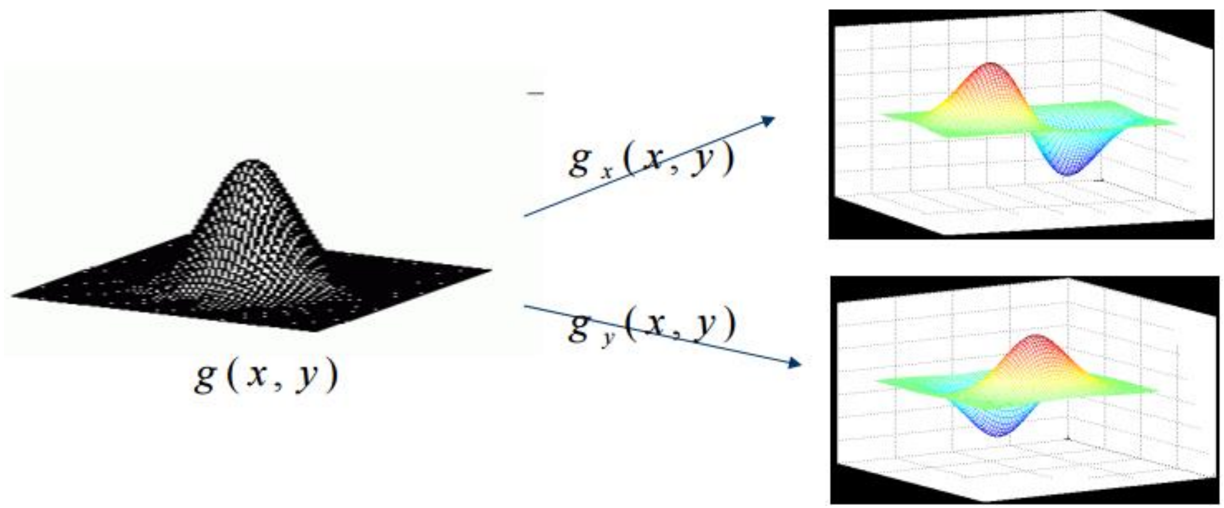
$$\nabla S = \begin{bmatrix} \frac{\partial}{\partial x} S & \frac{\partial}{\partial y} S \end{bmatrix}^T = \begin{bmatrix} S_x & S_y \end{bmatrix}^T$$

- Compute gradient magnitude and orientation

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

$$\theta = \tan^{-1} \frac{S_y}{S_x}$$

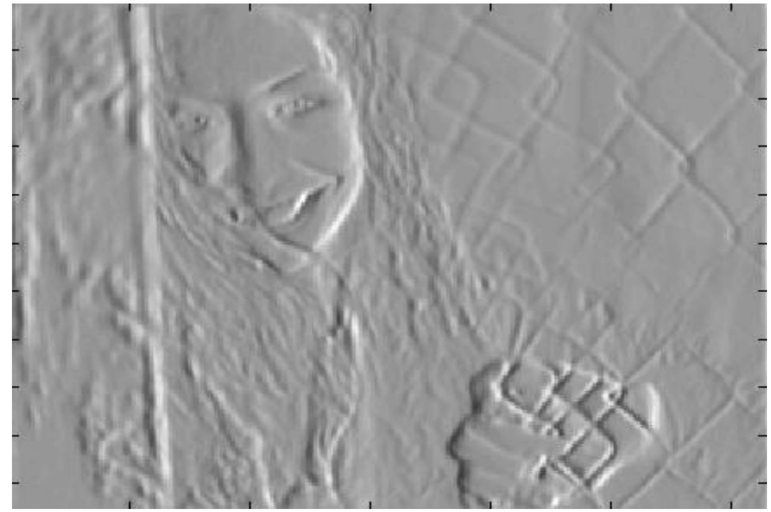
Canny Edge Detector Derivative of Gaussian



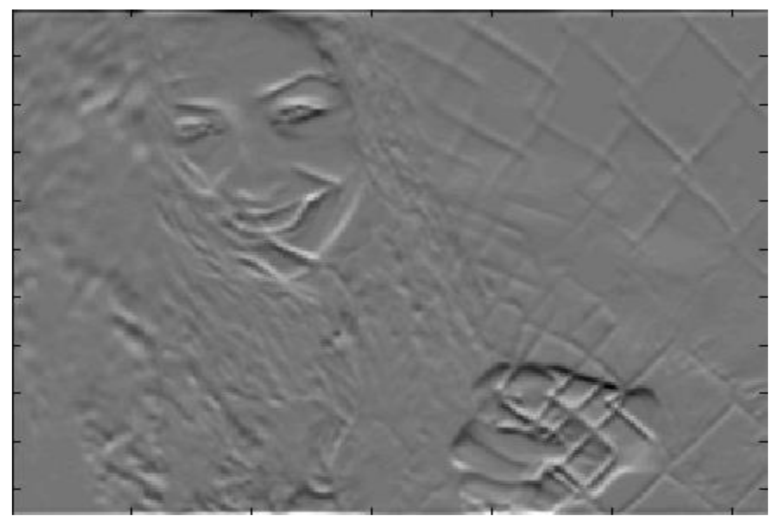
Canny Edge Detector

I

S_x



S_y



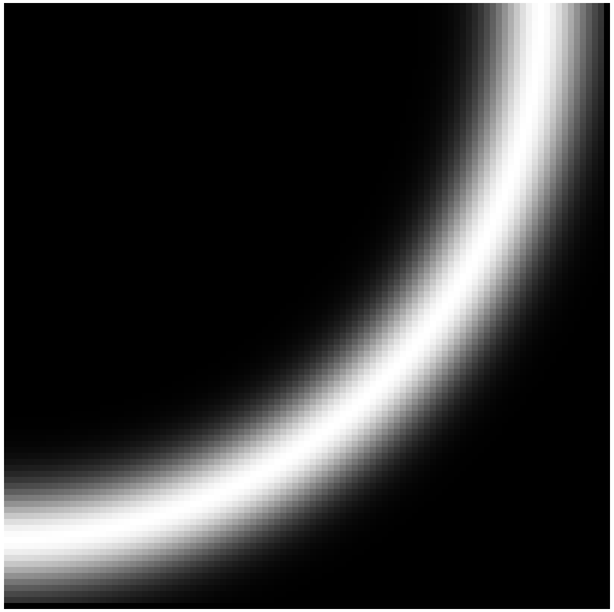
Canny Edge Detector

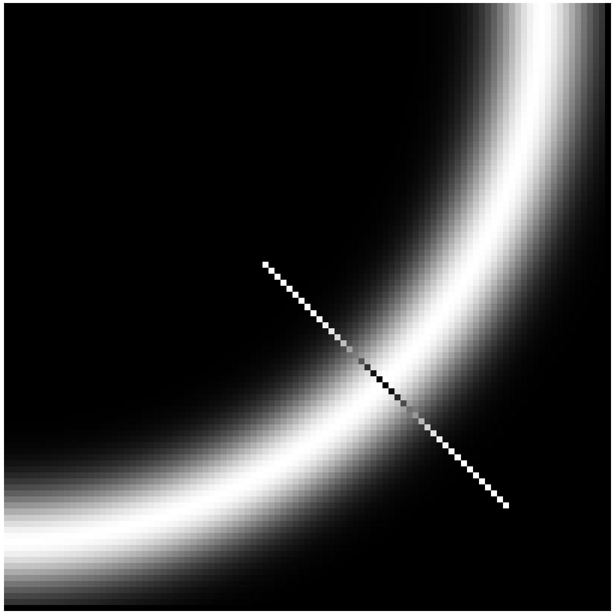
$$\left|\nabla S\right| = \sqrt{S_x^2 + S_y^2}$$

I

$$\left|\nabla S\right| \geq Threshold = 25$$

Non-Maximum Suppression



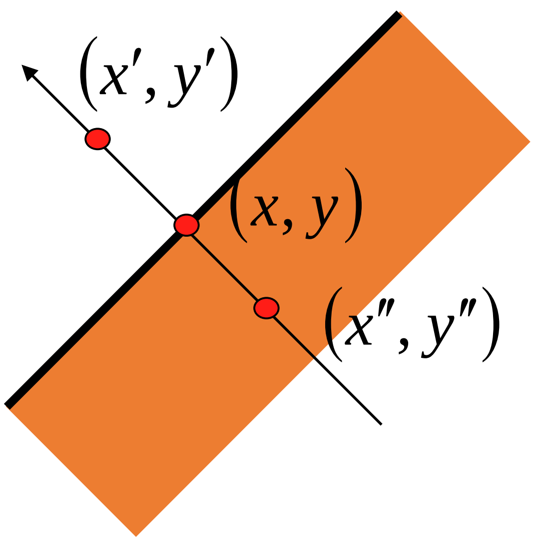


We wish to mark points along the curve where the magnitude is biggest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Non-Maximum Suppression

- Suppress the pixels in 'Gradient Magnitude Image' which are not local maximum

$$M(x,y)=\begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y)>|\nabla S|(x',y') \\ & \& \, |\nabla S|(x,y)>|\nabla S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$



(x',y') and (x'',y'') are the neighbors of (x,y) in $|\nabla S|$ along the direction normal to an edge

Non-Maximum Suppression

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

M

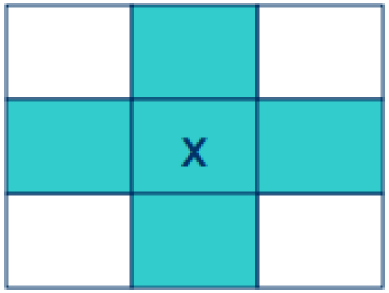
$$M \geq Threshold = 25$$

Hysteresis Thresholding

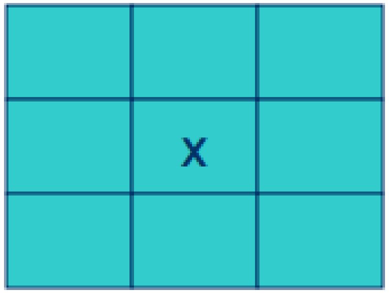
- If the gradient at a pixel is above ‘High’, declare it an ‘edge pixel’
- If the gradient at a pixel is below ‘Low’, declare it a ‘non-edge-pixel’
- If the gradient at a pixel is between ‘Low’ and ‘High’ then declare it an ‘edge pixel’ if and only if it is connected to an ‘edge pixel’ directly or via pixels between ‘Low’ and ‘ High’

Canny Edge Detector Hysteresis Thresholding

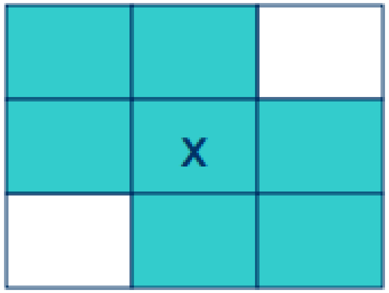
- Connectedness



4 connected



8 connected



6 connected

Hysteresis Thresholding

M

$M \geq Threshold = 25$

$High = 35$

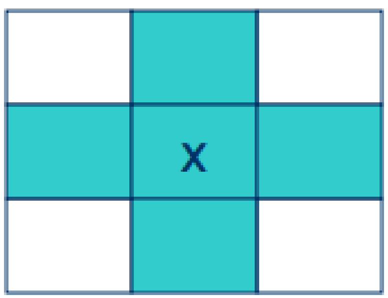
$Low = 15$

Region Growing

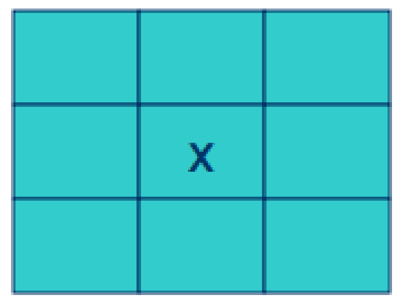
1	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6

Canny Edge Detector Hysteresis Thresholding

- Connectedness



4 connected



8 connected

Region Growing

1	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6

Region Growing

1	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6