Image Processing

BITS Pilani
Dubai Campus

Dr Jagadish Nayak

Image Segmentation

Fundamentals

- Let R be the entire spatial region occupied by an image
- Process that partitions R into n sub-regions R1,R2,R3….Rn, such that.

1. $\bigcup_{i=1}^{n} R_i = R$ (Segmentation must be complete)

2. $R_i \ \text{is a connected set} \ i = 1,2,3,4,\ldots..n$

   (Points in the region must be connected in some predefined sense)

3 $R_i \cap R_j = \phi \ \text{for all} \ i \ \text{and} \ j \ i \neq j$

   (Regions must be disjoint)

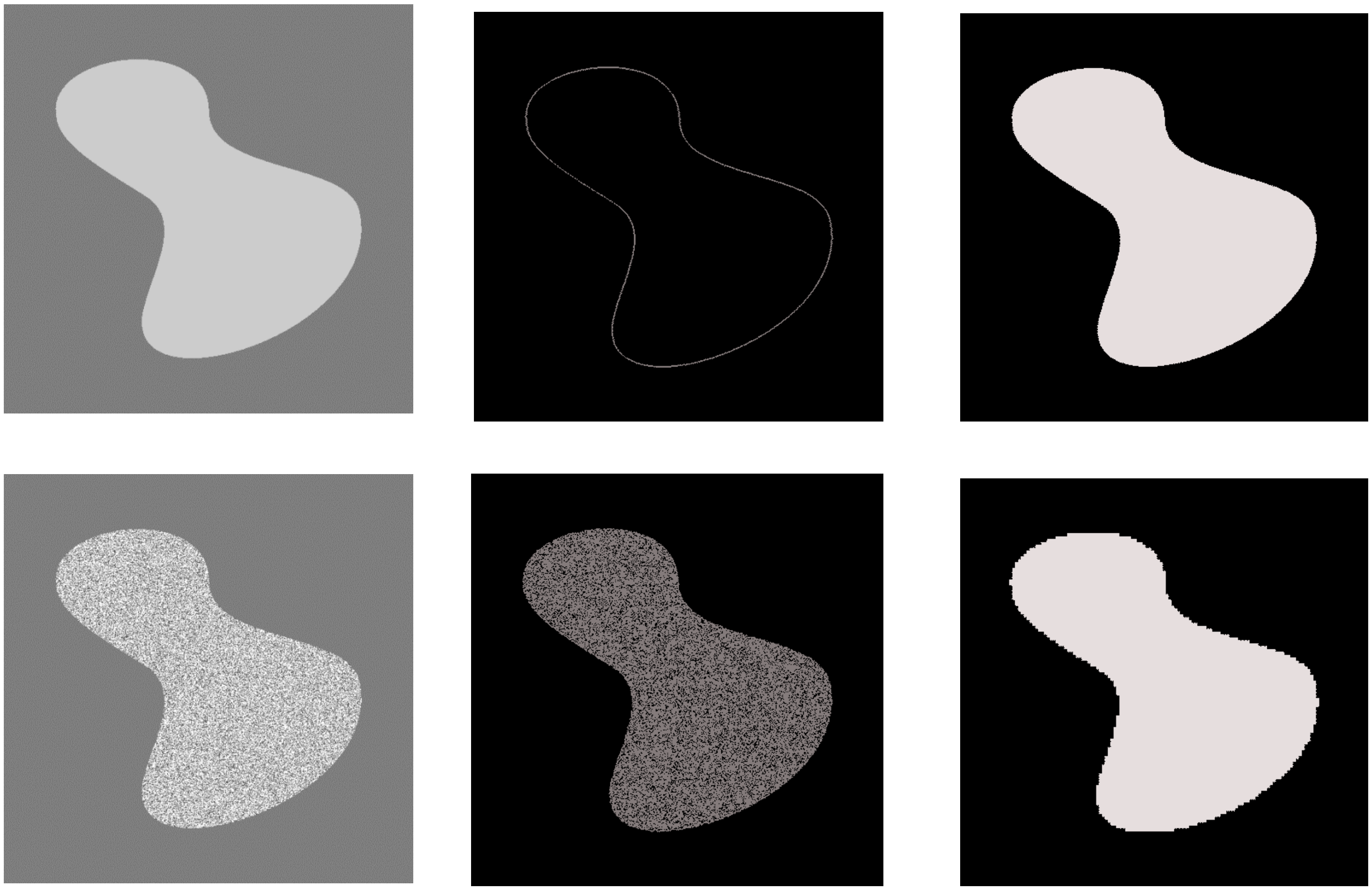
4 $Q(R_i) = TRUE \ \text{for} \ i = 1,2,3,\ldots..,n$

   (Property satisfied by all the pixels)

5 $Q(R_i \cup R_j) = FALSE \ \text{for any adjacent regions} \ R_i \ \text{and} \ R_j$

Two adjacent regions must be different in the sense predicate Q

Fundamentals



METHODS

- There are two approaches

<u>Discontinuity based</u>

Identification of
- Isolated points
- Edges
- lines

<u>Similarity based</u>

- Threshold operation
- Region growing
- Region splitting and merging

Detection of isolated points

- The Laplacian operator is given by

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

- The corresponding mask is

0	1	0
1	-4	1
0	1	0

- This is only for the vertical and horizontal directions
- Consider the one with diagonal direction , we get a mask

Detection of isolated points

1	1	1
1	-8	1
1	1	1

- Apply this mask on the image, using spatial mask processing
- Example

4	7	6	2	3
2	5	4	1	2
1	1	2	3	4
0	2	3	4	5
2	6	6	4	3

4	7	6
2	5	4
1	1	2

1	1	1
1	-8	1
1	1	1

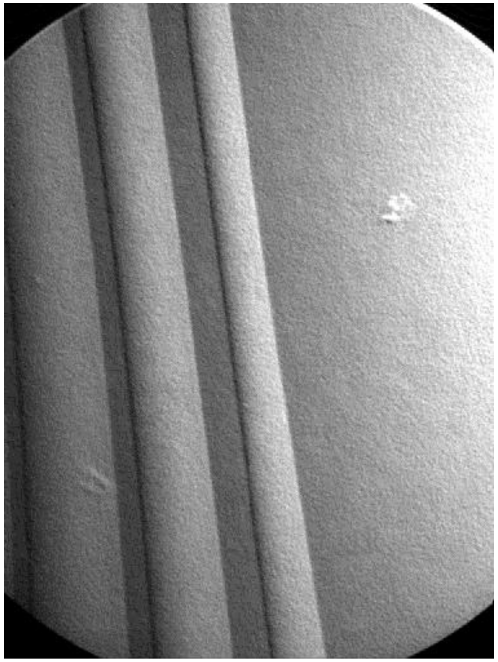
1x4	1x7	1x6
1x2	-8x5	1x4
1x1	1x1	1x2

Sum = 4+7+6+2-40+4+1+1+2
= -13

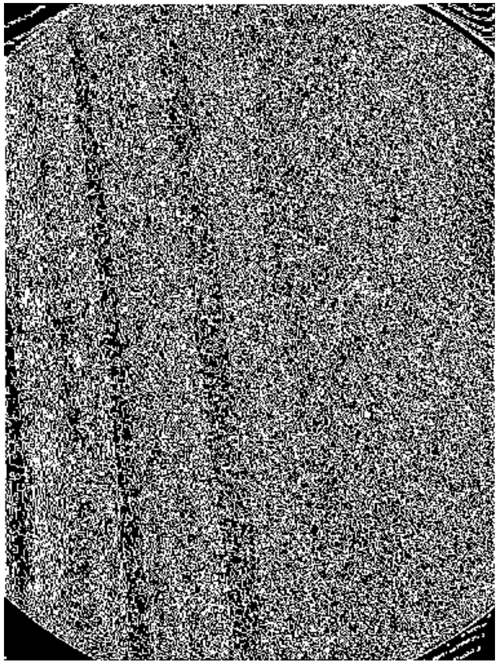
Detection of isolated points

- If the absolute value of sum is greater than some positive threshold value ,then that value is considered to be a point.
- value is set to 1 and all the other values are set to 0
- Threshold can be any value based on the selection
- Consider an example , where threshold value is 90% of the highest absolute value of the image.

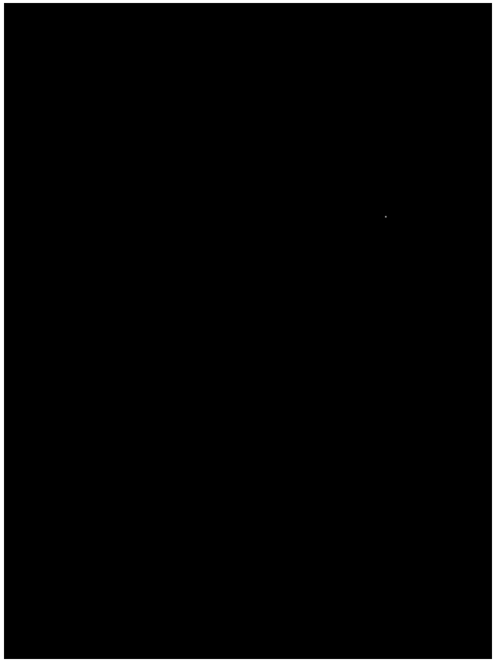
Detection of isolated points



X-ray image of the turbine blade



Result of convolving with the mask



Applying threshold to detect single point

Detection of lines

- Laplacian mask can be used
- The Laplacian, used for point detection, is isotropic and has no direction information.
- Detecting the lines in specific direction is of interest.
- Following masks are used for detecting lines

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

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

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

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

Horizontal **+45º** **Vertical** **-45º**

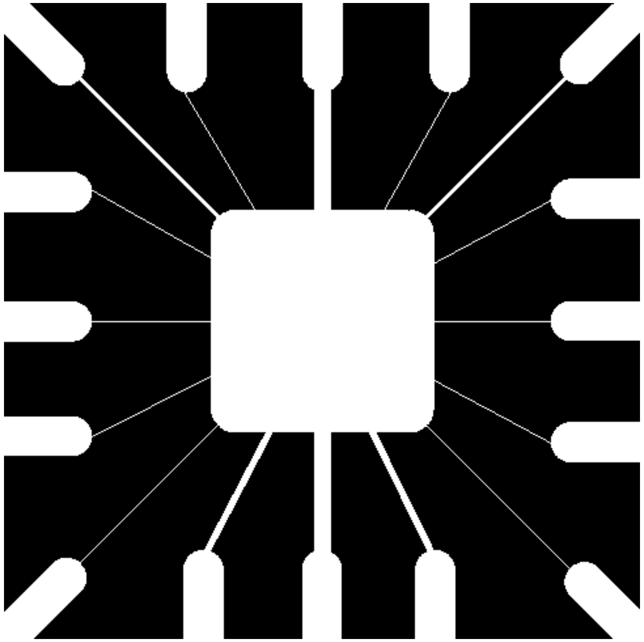
Detection of lines

- Constant intensity area yield zero response
- Let R_1,R_2,R_3 and R_4 be the responses of the masks belongs to Horizontal, +45º vertical, -45º respectively.
- When the image is filtered using all these 4 masks
- If, at a given point in the image $|R_k| > |R_j|$, for all $j \neq k$
- That point is said to be more likely associated with the linein the direction of the mask k
- Example , if at a point in the image, $|R_1| > |R_2|$ for $j=2,3,4$ that particular point is said to be more likely associated with the horizontal line.

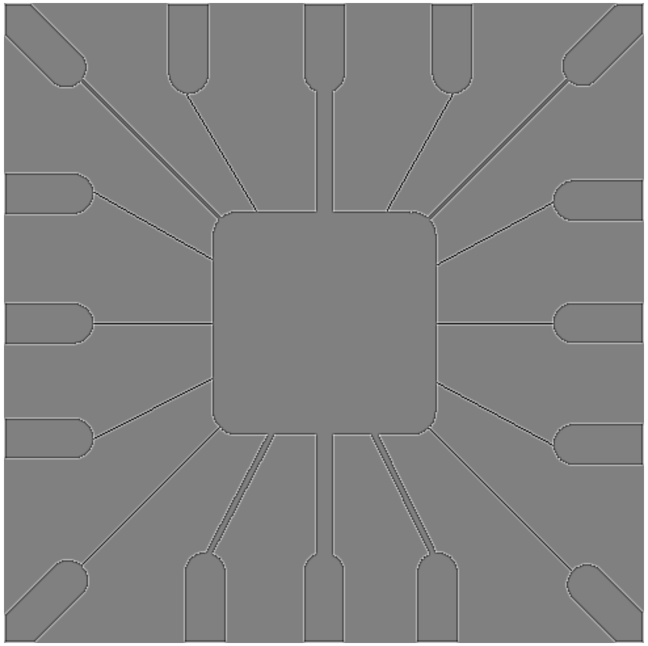
Detection of lines

- Let us apply Laplacian in the image.

1	1	1
1	-8	1
1	1	1

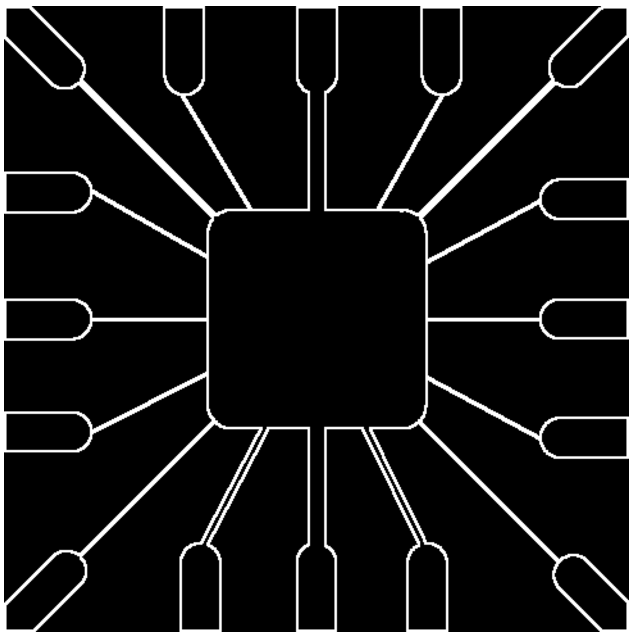


Original image

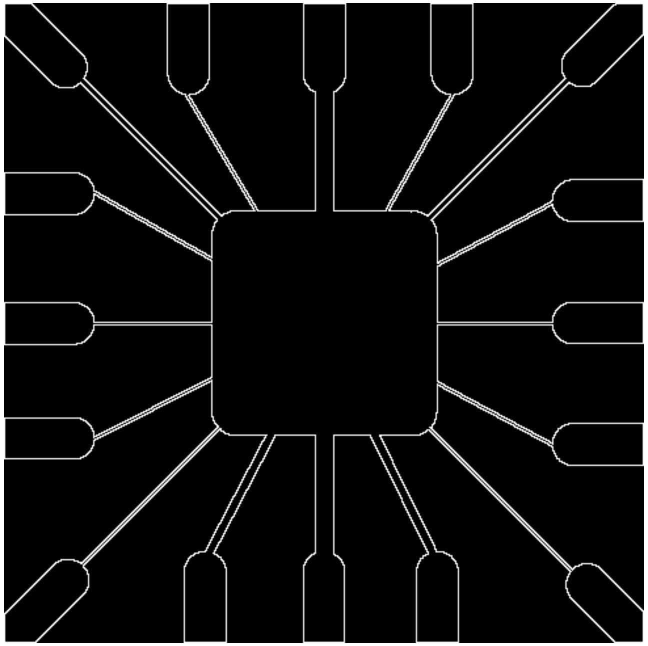


Magnitude section of the Laplacian. Positive and –ve double line effect.

Detection of lines



Absolute value of the Laplacian

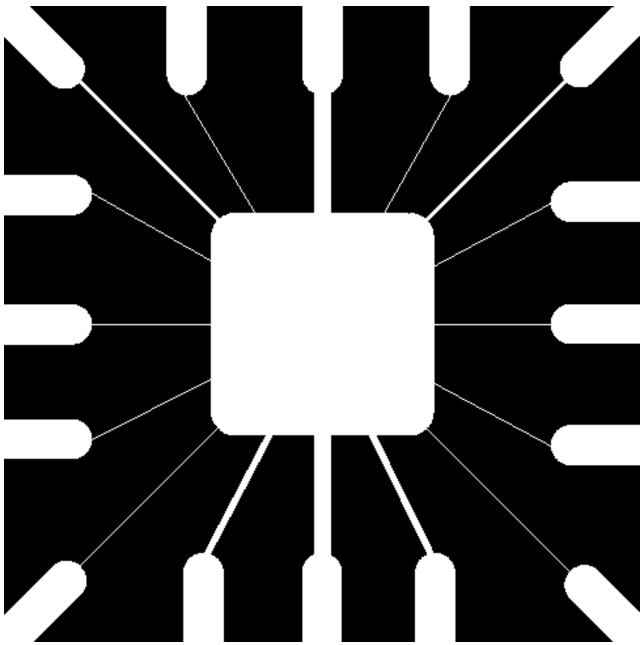


Positive value of the Laplacian
Thinner lines considerably useful

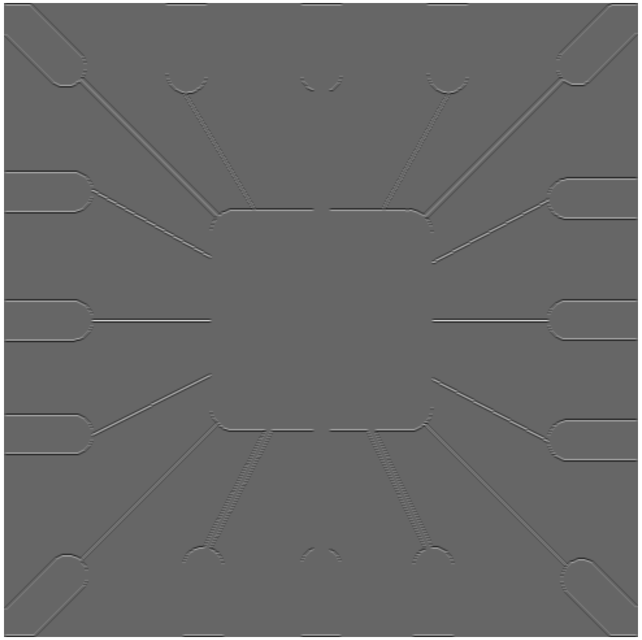
Detection of lines

- Now let us apply an line detection operator , which detects horizontal lines.

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

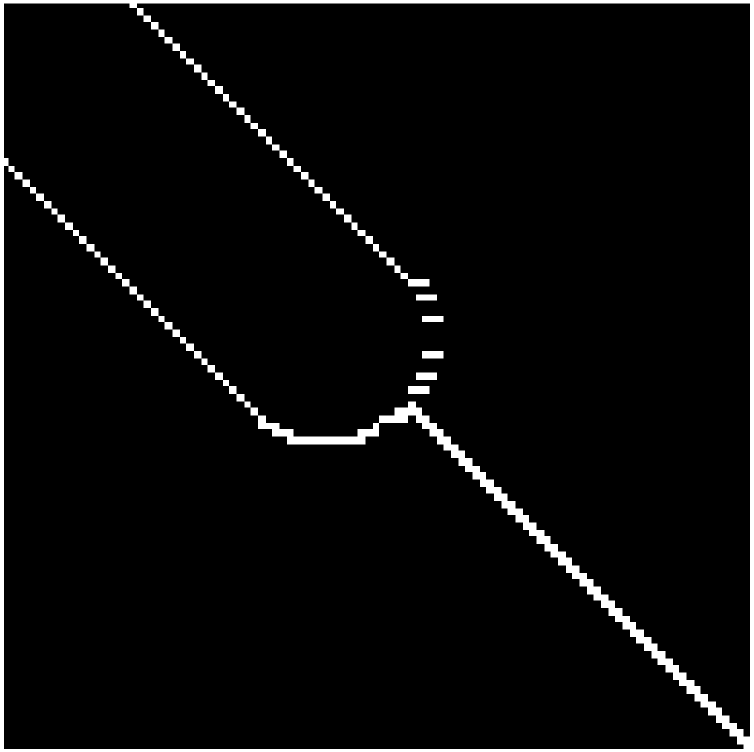


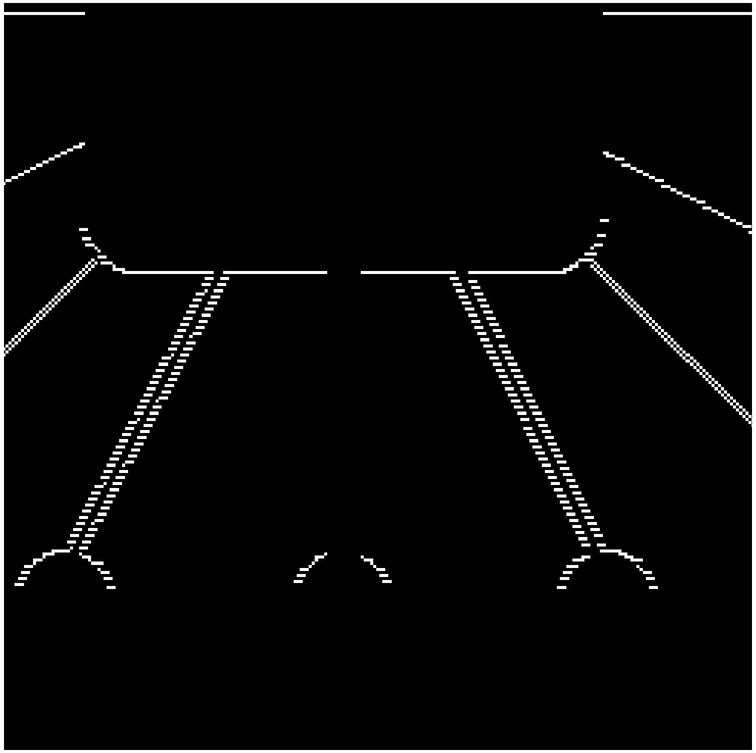
Original image



Result of processing with horizontal line detector mask

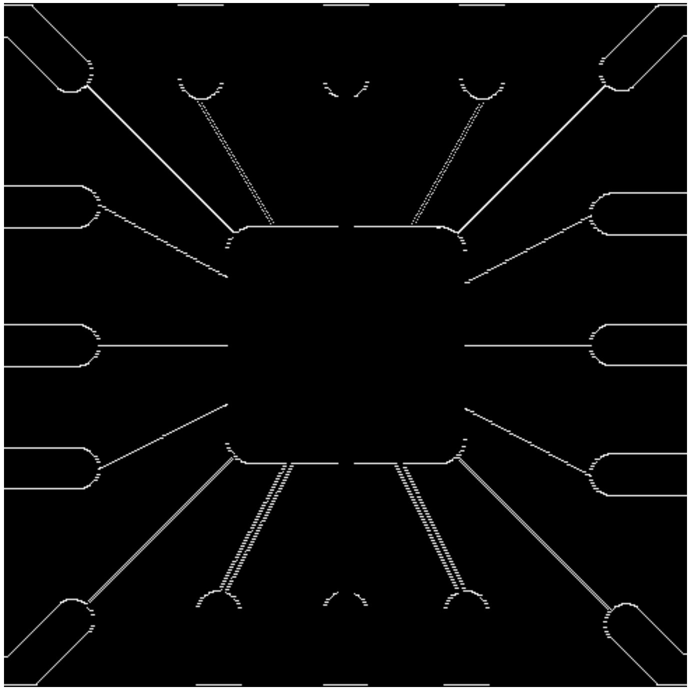
Detection of lines



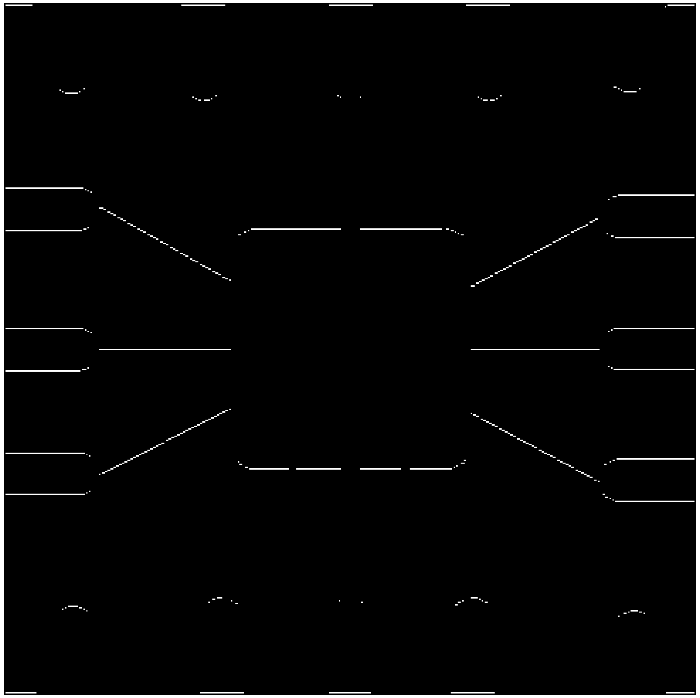


Zoomed view

Detection of lines



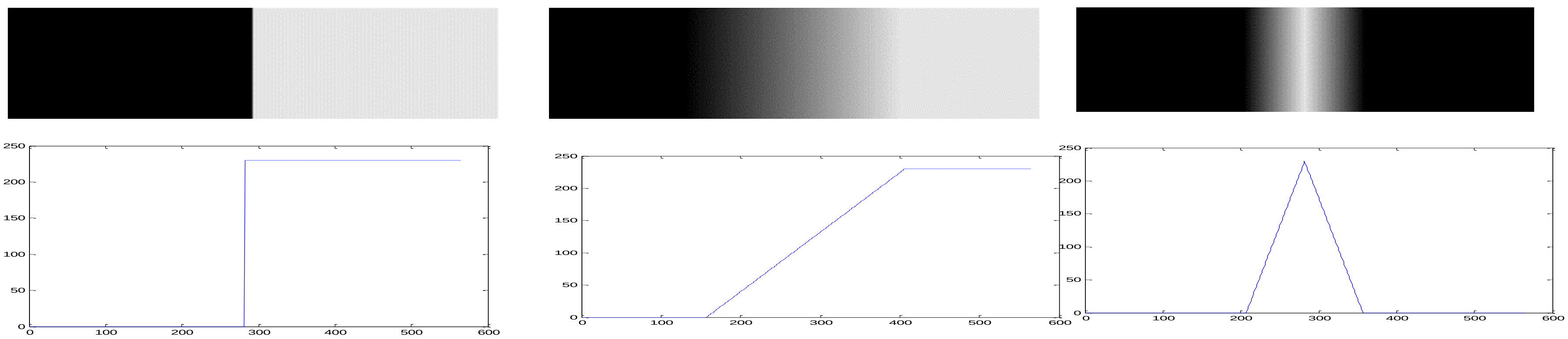
Horizontal line detected image with all negative values set to zero .



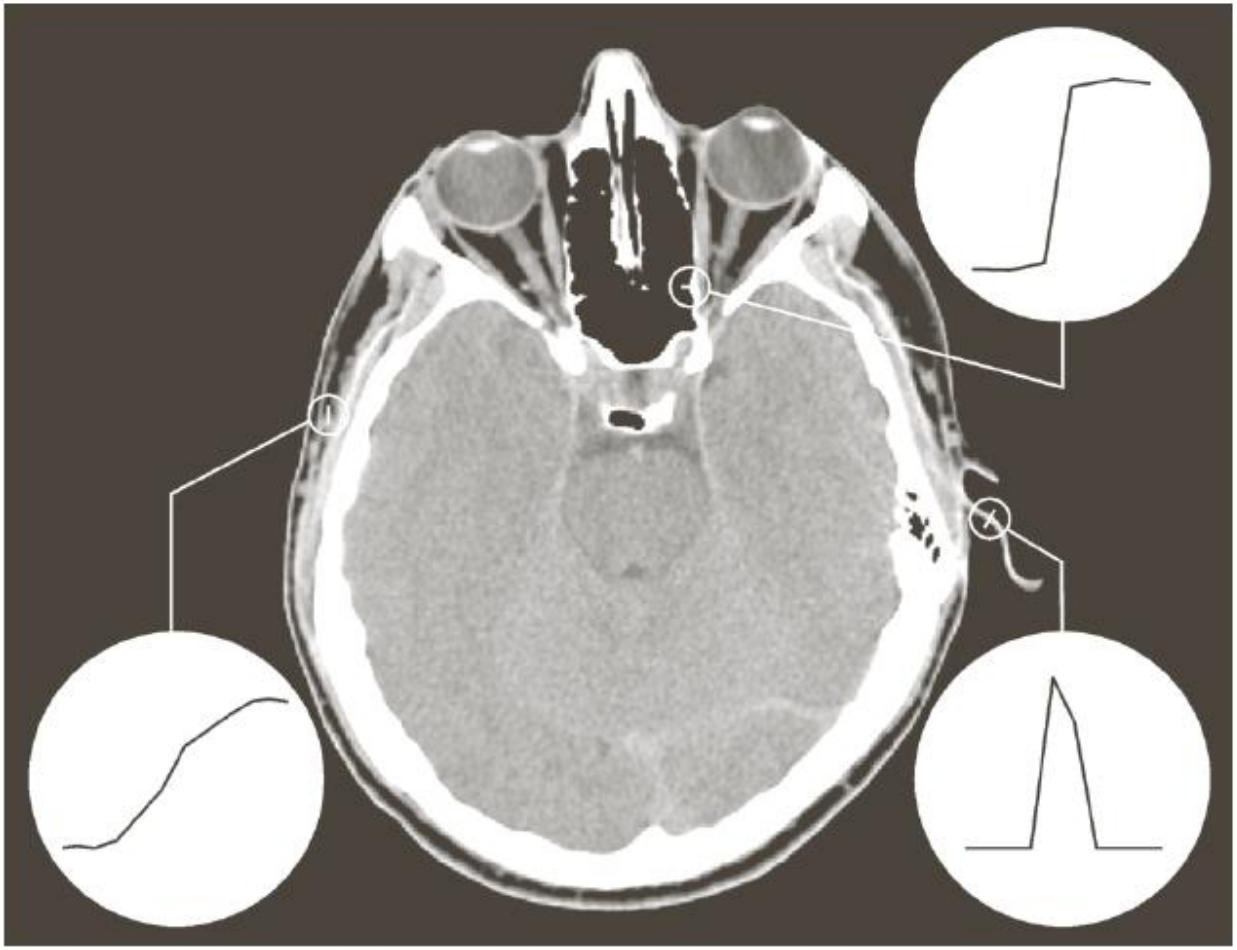
Horizontal line detected image with applied threshold

Detection of edges.

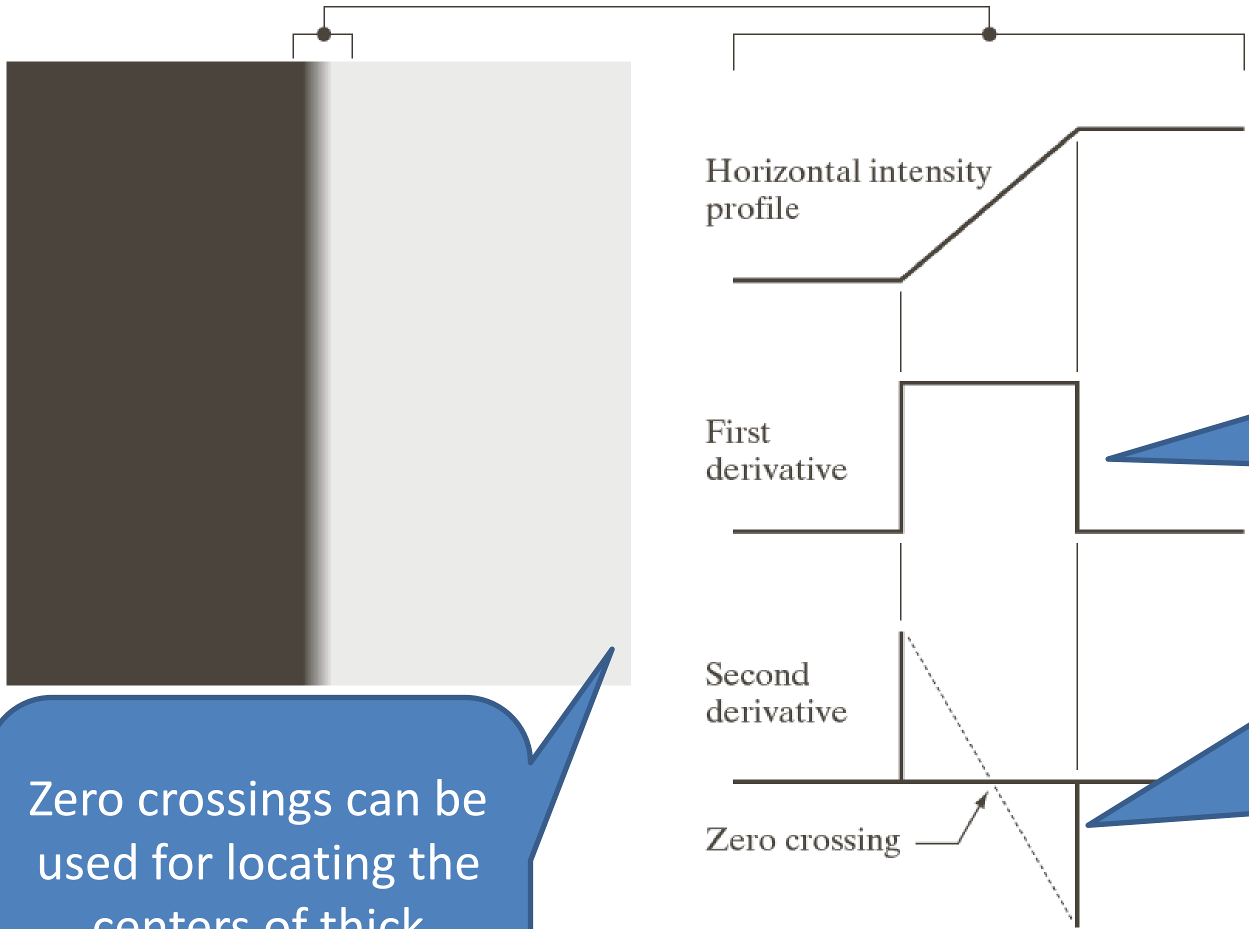
- Different types of edges



Detection of edges.



Detection of edges.

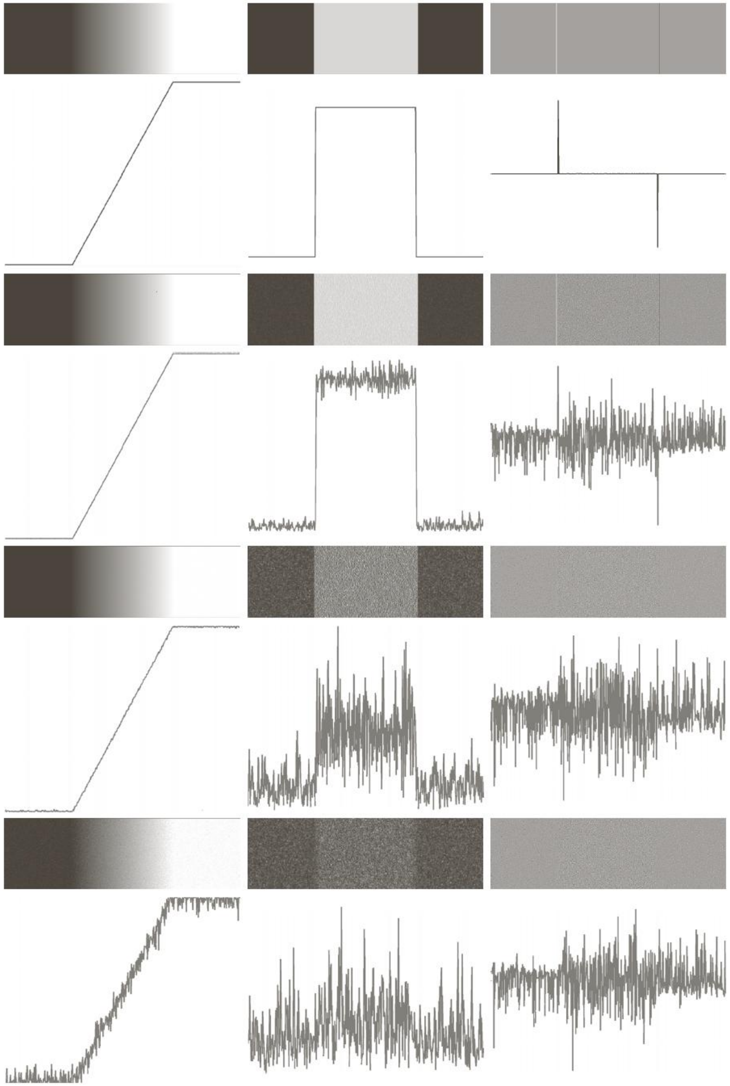


Detects the presence of edge at a point in an image

Sign can be used to detect edge is in darker side or brighter side ,Produces two edges

Zero crossings can be used for locating the centers of thick edges.

Detection of edges.



- Profile without any noiise
- Profile with Gaussian noise with zero mean an 0.1 standard dev

0.1 std

10 std

Detection of edges

- As we have seen edges can be detected using first order or second order derivatives.
- Let us start with the first order derivatives

Let us find the edge strength and direction at a location (x,y) in an image f.

Let us define the gradient as ∇f, *defined as vector*

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

- This vector points in the direction of the greatest rate of change of f at location (x,y).

Detection of edges

- The magnitude (length) of vector ∇f denoted as $M(x,y)$

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2} \; (\therefore |g_x| + |g_y|)$$

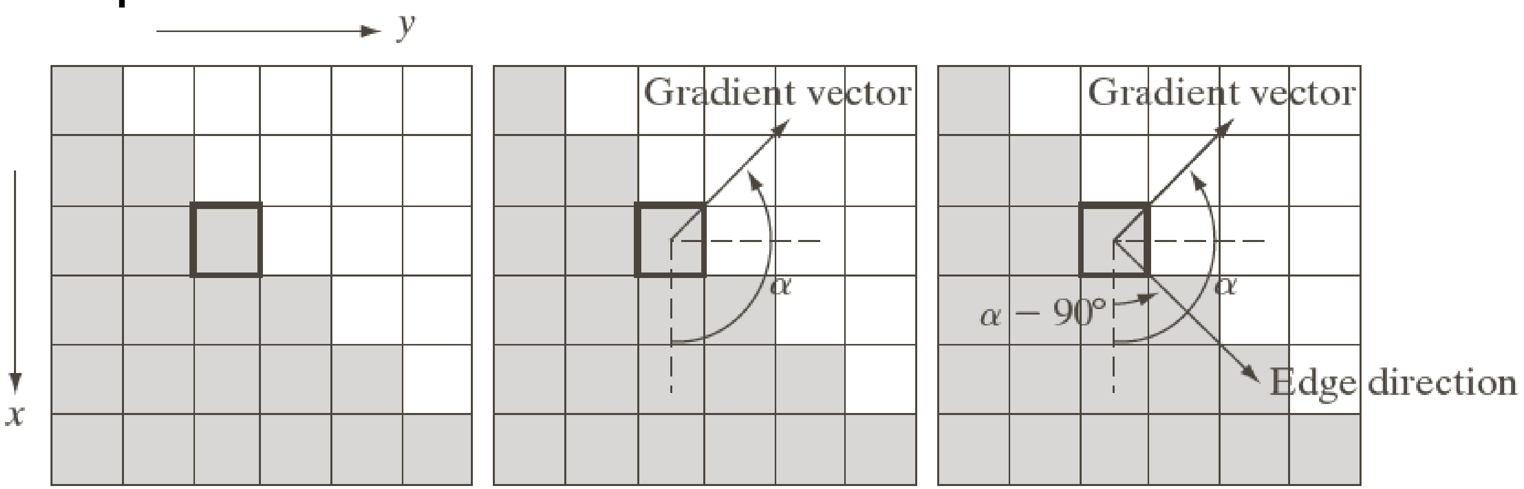
- The image which we get is a gradient image
- The direction of the gradient vector is given by the angle.

$$\alpha(x,y) = tan^{-1}\left[\frac{g_x}{g_y}\right] \; measured\; w.r.t\;\; x - axis$$

- The direction of an edge at an arbitrary point (x,y) is orthogonal to the direction , $\alpha(x,y)$, of the gradient vector at any point.

Detection of edges

- Example



- Each square is a pixel
- Let the gray pixel be 0 and white pixel be 1.
- Let us get the partial derivative along x and y direction.

Detection of edges

- To get partial derivative in x direction

  Subtract the pixels in the top row of the neighborhood from the pixels in the bottom row

0	1	1
0	0	1
0	0	0

- We get

$$g_x = \frac{\partial f}{\partial x} = (0-0)+(0-1)+(0-1) = -2$$

Detection of edges

- To get partial derivative in y direction

  Subtract the pixels in the left column from the pixels in the right column.

0	1	1
0	0	1
0	0	0

- We get

$$g_x = \frac{\partial f}{\partial y} = (1-0) + (1-0) + (0-0) = 2$$

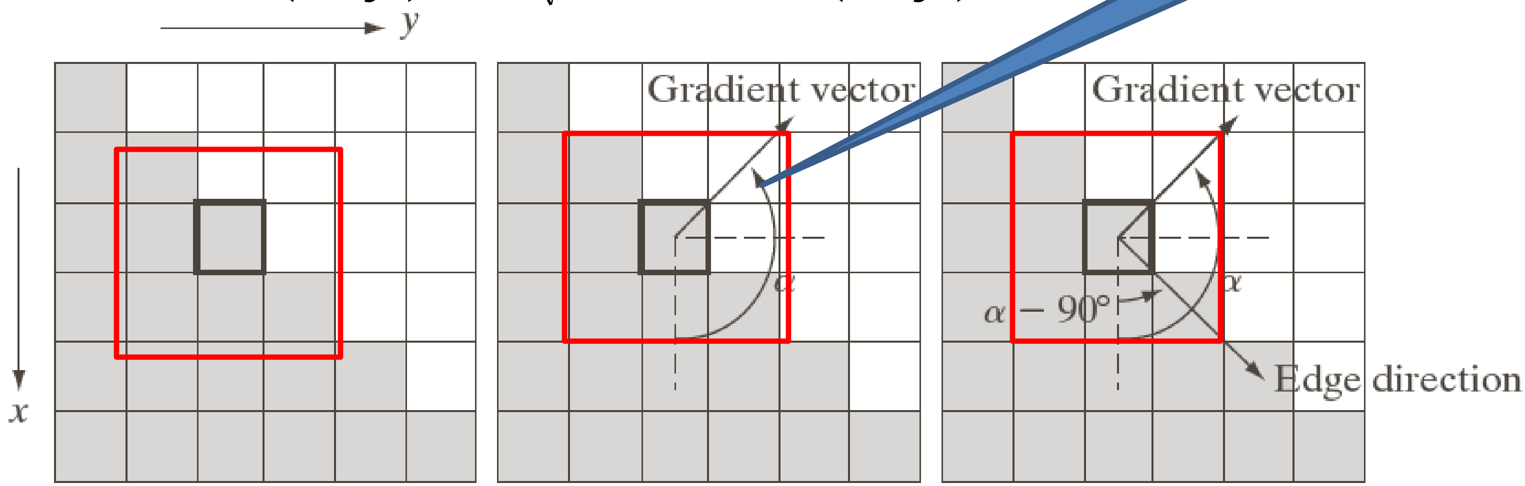
Detection of edges

- At the point of question, we get

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

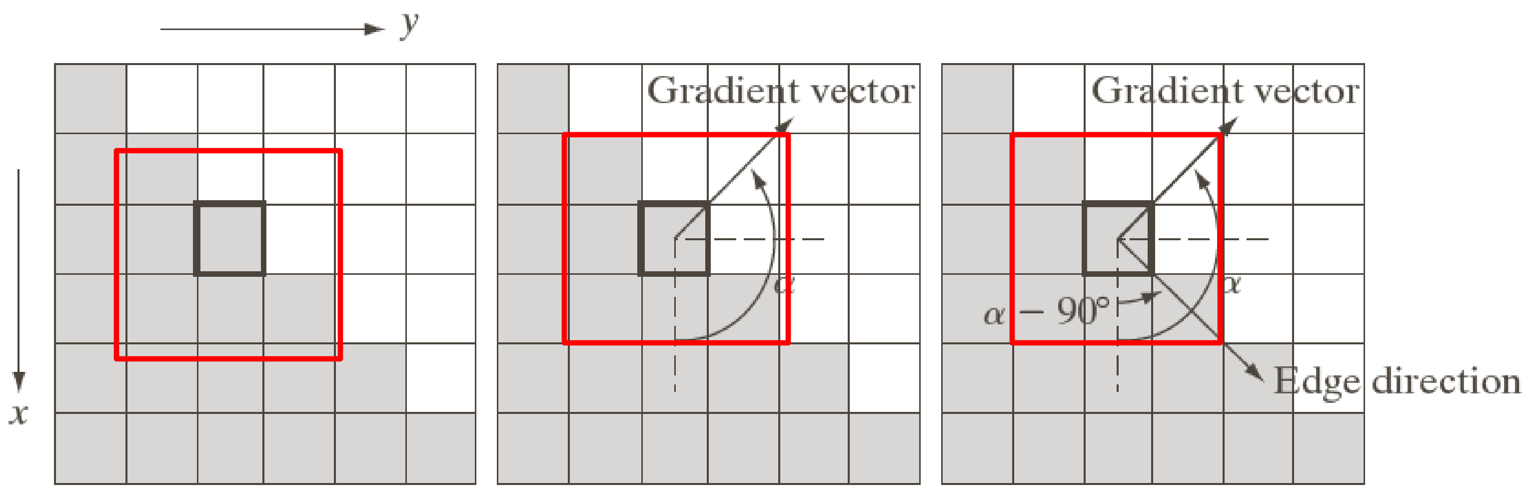
$$M(x,y) = 2\sqrt{2} \quad and \quad \alpha(x,y) = -45^o$$

Which is same as 135^o measured from the +ve axis



Detection of edges

- Edge at a point is orthogonal to the gradient vector at that point.
- So direction of the angle of the edge in this example is (α-90o)=45o
- All edge points in the example shown below have same gradient so the entire segment is in the same direction.



Gradient operator

- Gradient is computed by partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location in the image. For a digital quantity partial derivative is given by

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

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

- The above two equations can be implemented using 1D masks as

−1
1

−1	1

Gradient operator

- When the diagonal edge is required 2D mask is required
- Roberts cross-gradient operator is one of the first attempt.
- Consider 3x3 sub-image.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts operator based on implementing the diagonal differences.

-1	0
0	-1

0	-1
-1	0

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6)$$

Not useful for computing the edge direction. The mask which is symmetric bot the center point is preferred

Gradient operator

- Simplest 3x3 mask is

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_7 + z_8 + z_9)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

- Corresponding mask is called as Prewitt operator

-1	-1	1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

- A slight variation used weight of 2 in the center coefficient

-1	-2	1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Which is called as Sobel operator.

Gradient operator

- For the strongest responses along the diagonal direction, the modified Prewitt and Sobel masks are given by.

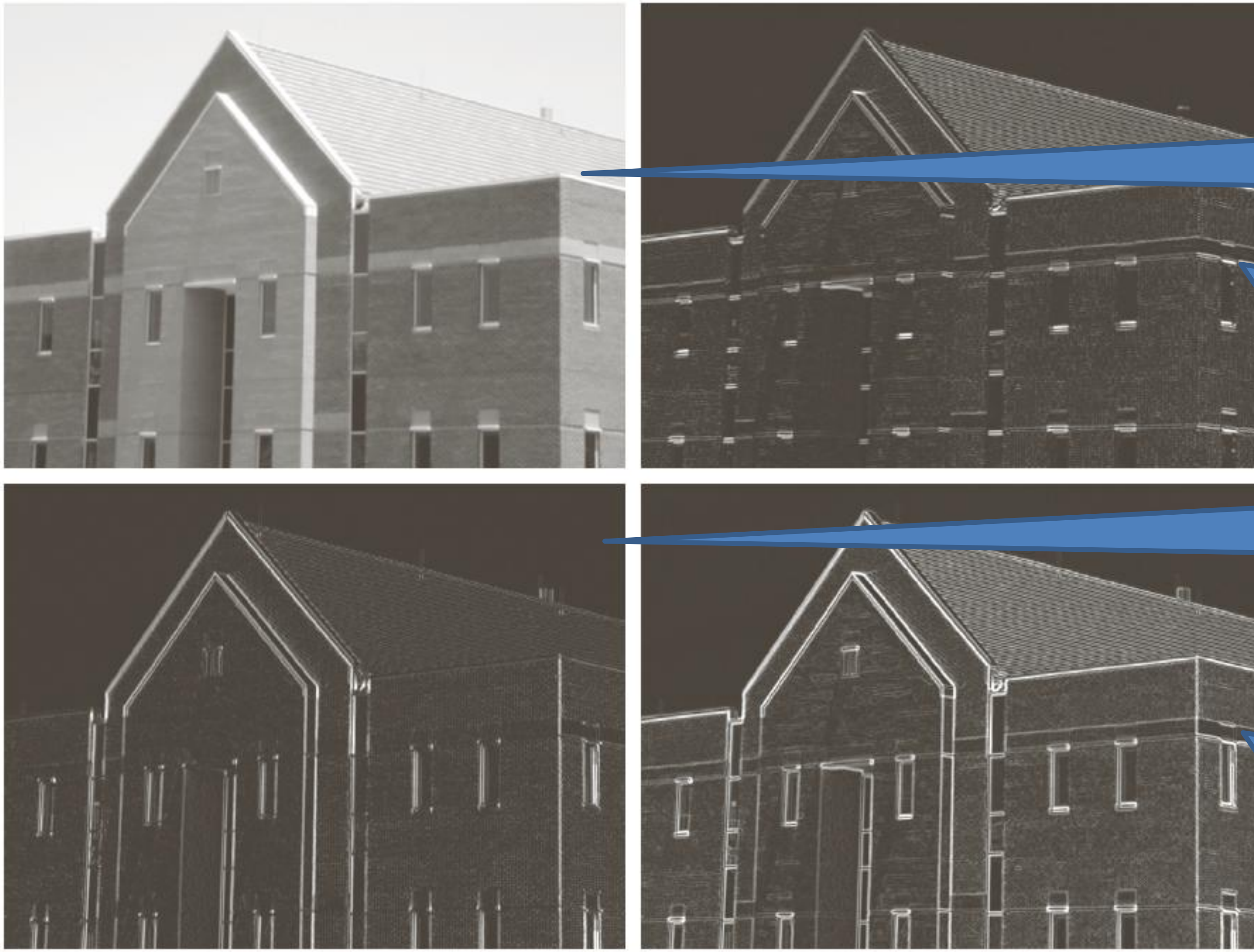
0	1	1		−1	−1	0
−1	0	1		−1	0	1
−1	−1	0		0	1	1

Prewitt

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

Sobel

Gradient operator (Example)



Original image (The intensity values scaled between [0 1]

g_x component of the sobel mask

g_y component of the sobel mask

The gradient image $|g_x| + |g_y|$

Marr- Hildreth edge detector

- Earlier method does not makes use of edge characteristics or noise before applying the edge detection operator.
- Marr-Hildreth addresses this issue
- They developed operators which changes its size based on the blurry edges and sharply focused fine details.
- The operator, which fulfill this requirement is the filter $\nabla^2 G$
- As we have seen earlier ∇^2 is Laplacian operator $\partial^2/\partial x^2 + \partial^2/\partial y^2$
- G is 2D Gaussian function

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

- Where σ is standard deviation

Marr- Hildreth edge detector

- The expression for $\nabla^2 G$

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

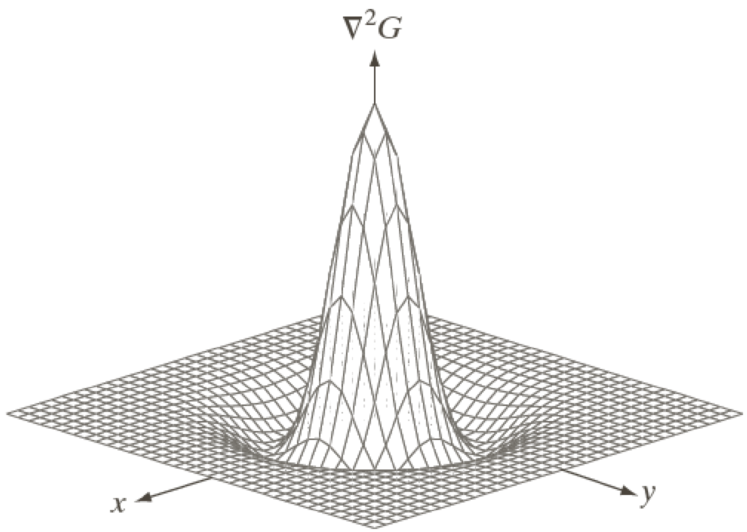
$$= \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

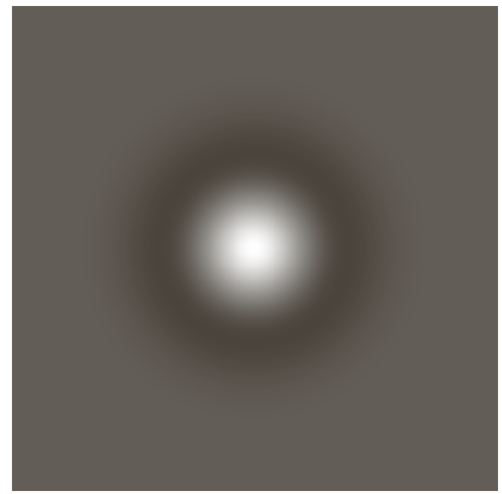
$$\left[\frac{x^2}{\sigma^2} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^2} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

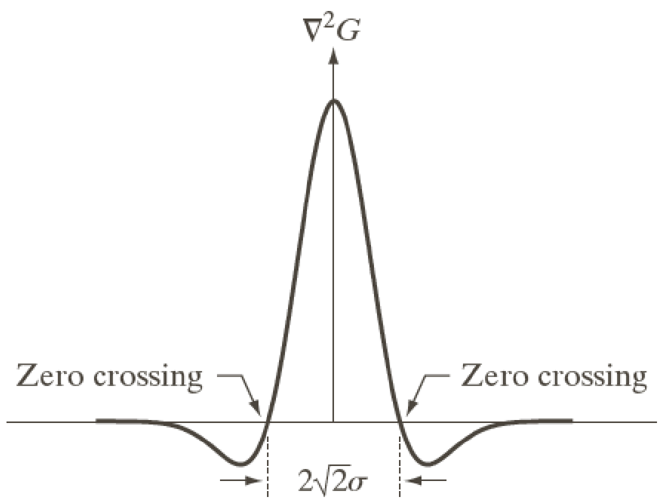
$$\nabla^2 G(x,y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- This expression is called as [Laplacian of Gaussian (LoG)]

Marr- Hildreth edge detector







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

Marr- Hildreth edge detector

- Marr-Hildreth algorithm consist of convolving the LoG filter with an input image , f(x,y)

$$g(x,y) = [\nabla^2 G(x,y)] * f(x,y)$$

- Then, find the zero crossings of g(x,y) to determine the location of the edges in f(x,y).
- These are linear process , we can write above as

$$g(x,y) = \nabla^2 \left[G(x,y) * f(x,y) \right]$$

- This indicates , smooth the image first with a Gaussian filter and then compute the Laplacian of the result.

Marr- Hildreth edge detector

- Following are the steps used

Step1: Filter the input image with an nxn Gaussian low pass filter obtained by sampling.

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

Step2: Compute Laplacian of the image resulting from above step using 3x3 mask

1	1	1
1	-8	1
1	1	1

Step3: Find the zero crossings of the image from the above step.

Marr- Hildreth edge detector

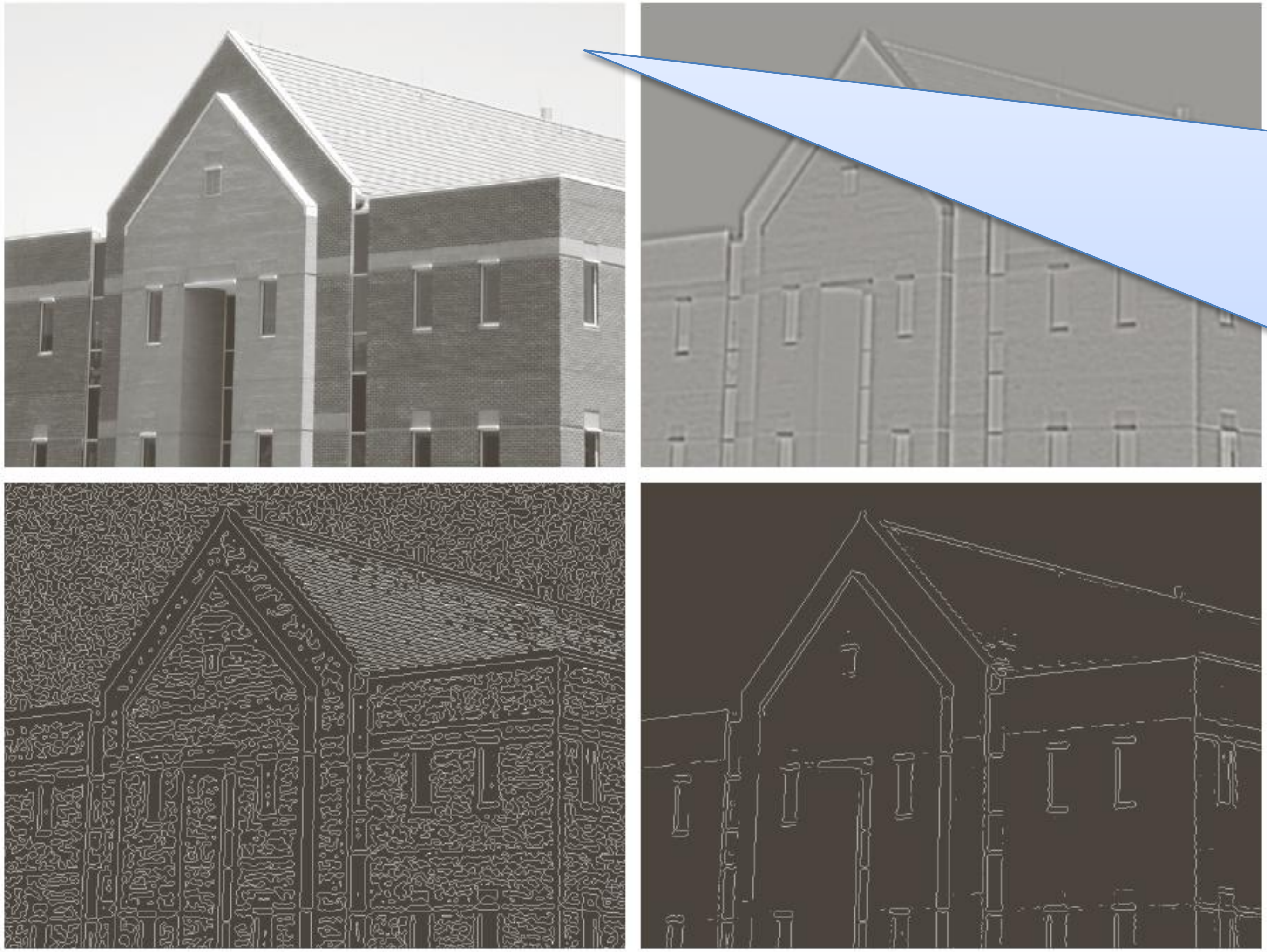
- What should be the size of the Gaussian filter mask nxn,
- Most of the volume of Gaussian surface lies between $\pm 3\sigma$, so n is a smallest odd integer greater than or equal to 6σ
- Mask smaller than this will tend to truncate the LOG function.

<u>How do we find the zero crossings</u>

- Consider 3x3 neighborhood across p,
- Zero crossing at p implies that signs of at least two of its opposing neighbor pixels must differ.

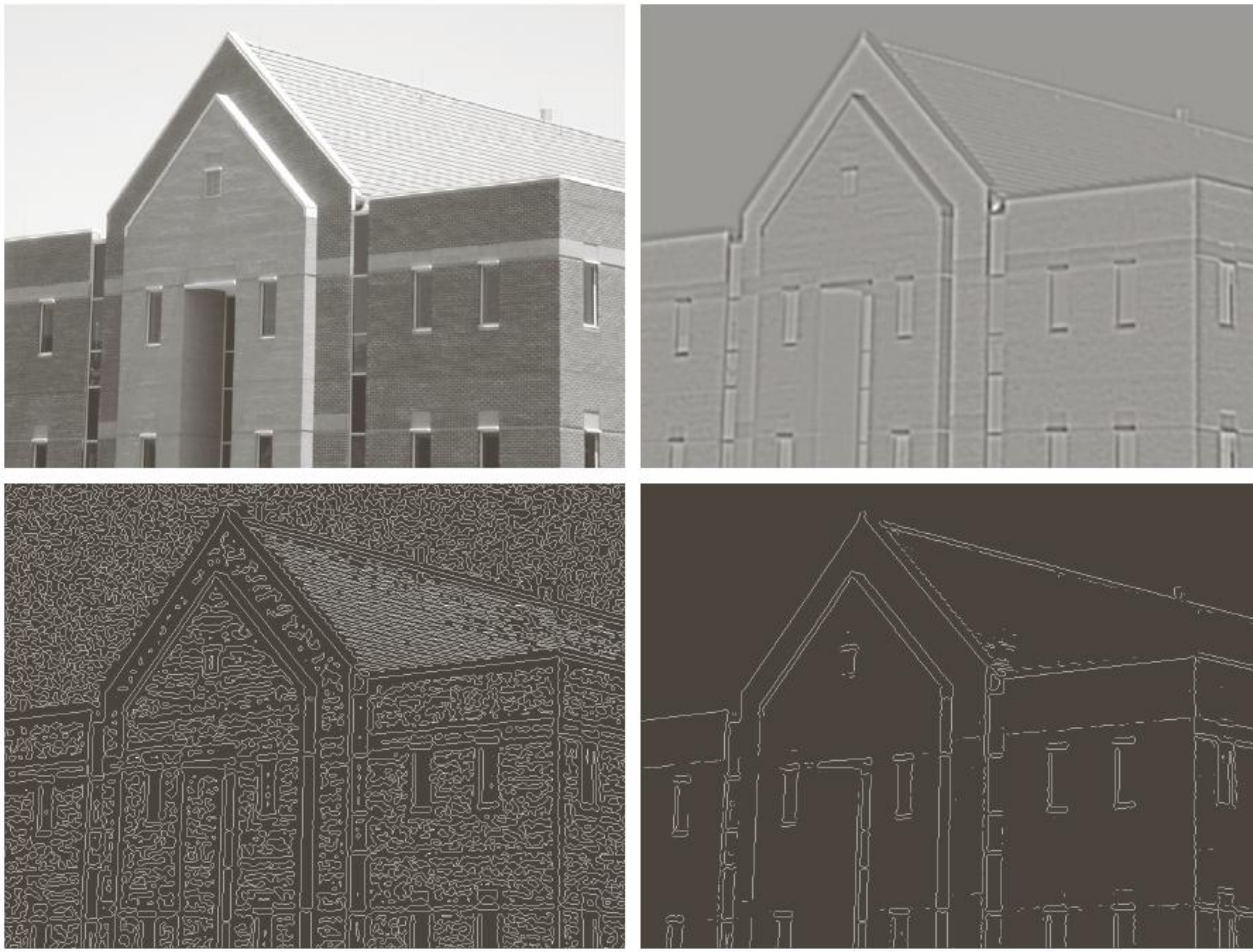
+	+	-
-	p	+
+	-	-

Marr- Hildreth edge detector (Example)



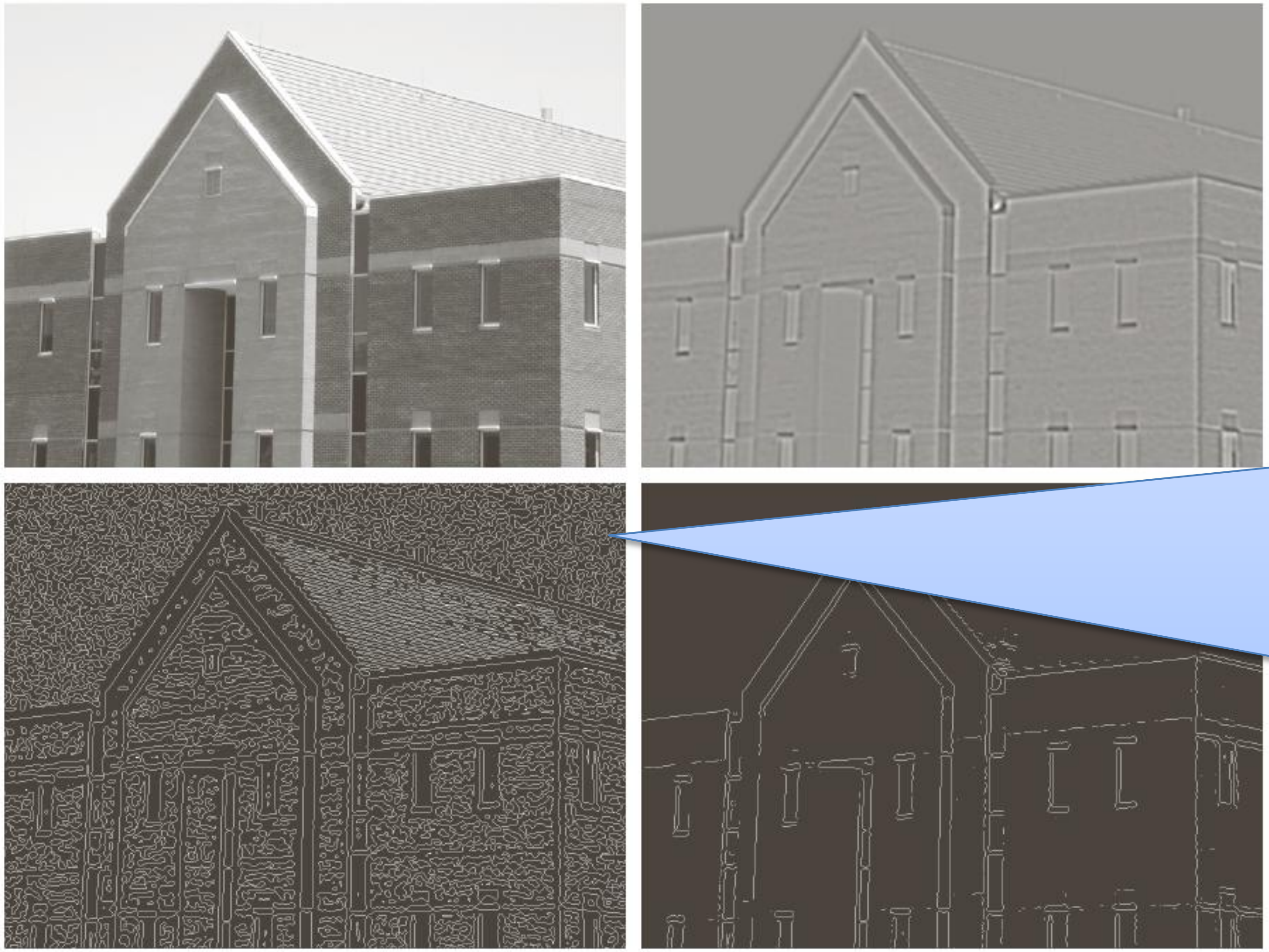
Original image

Marr- Hildreth edge detector (Example)



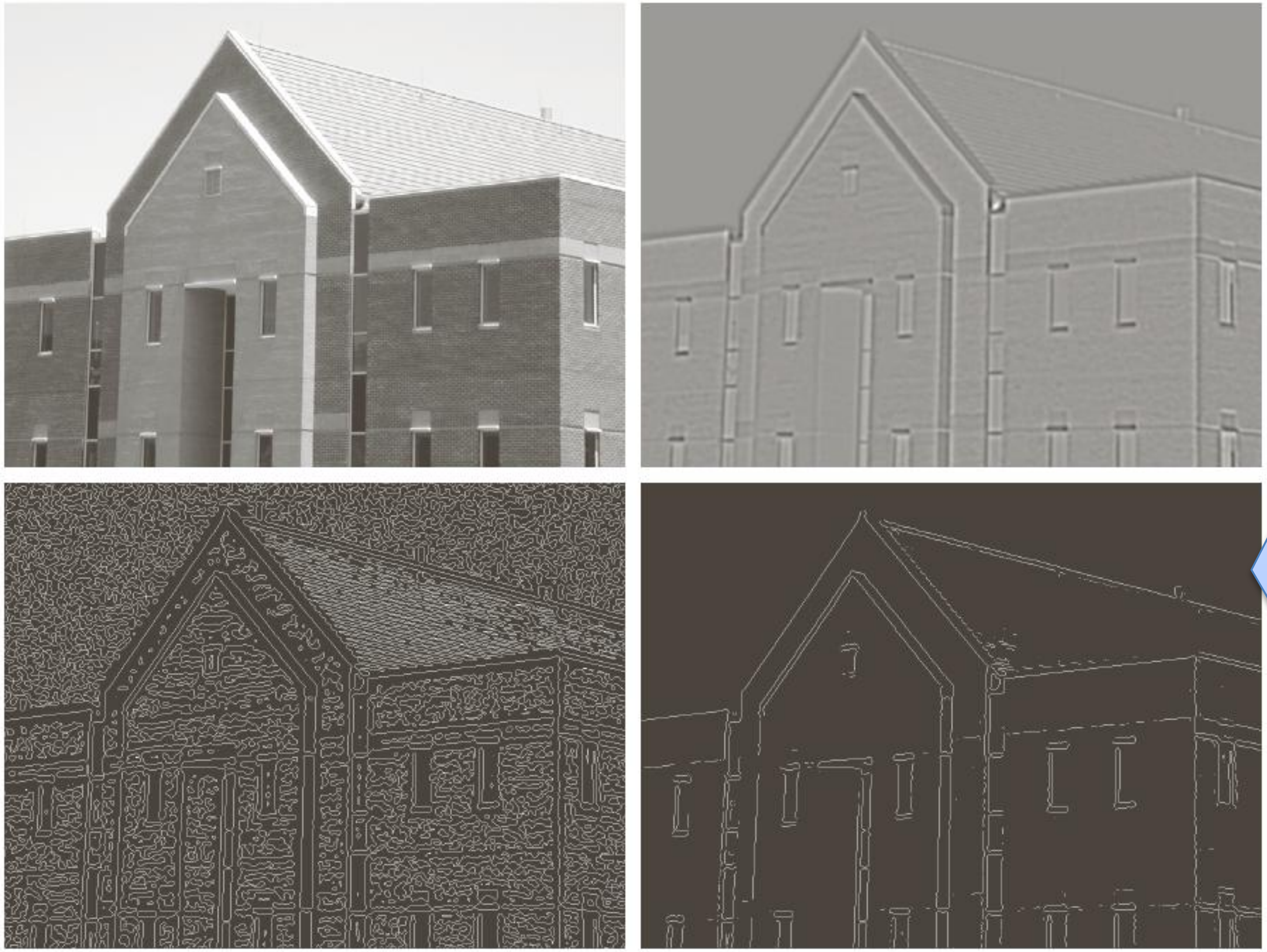
Result of step 1 and 2, using σ=4 and n=25

Marr- Hildreth edge detector (Example)



Zero crossings in 3x3 with a threshold value 0

Marr- Hildreth edge detector (Example)

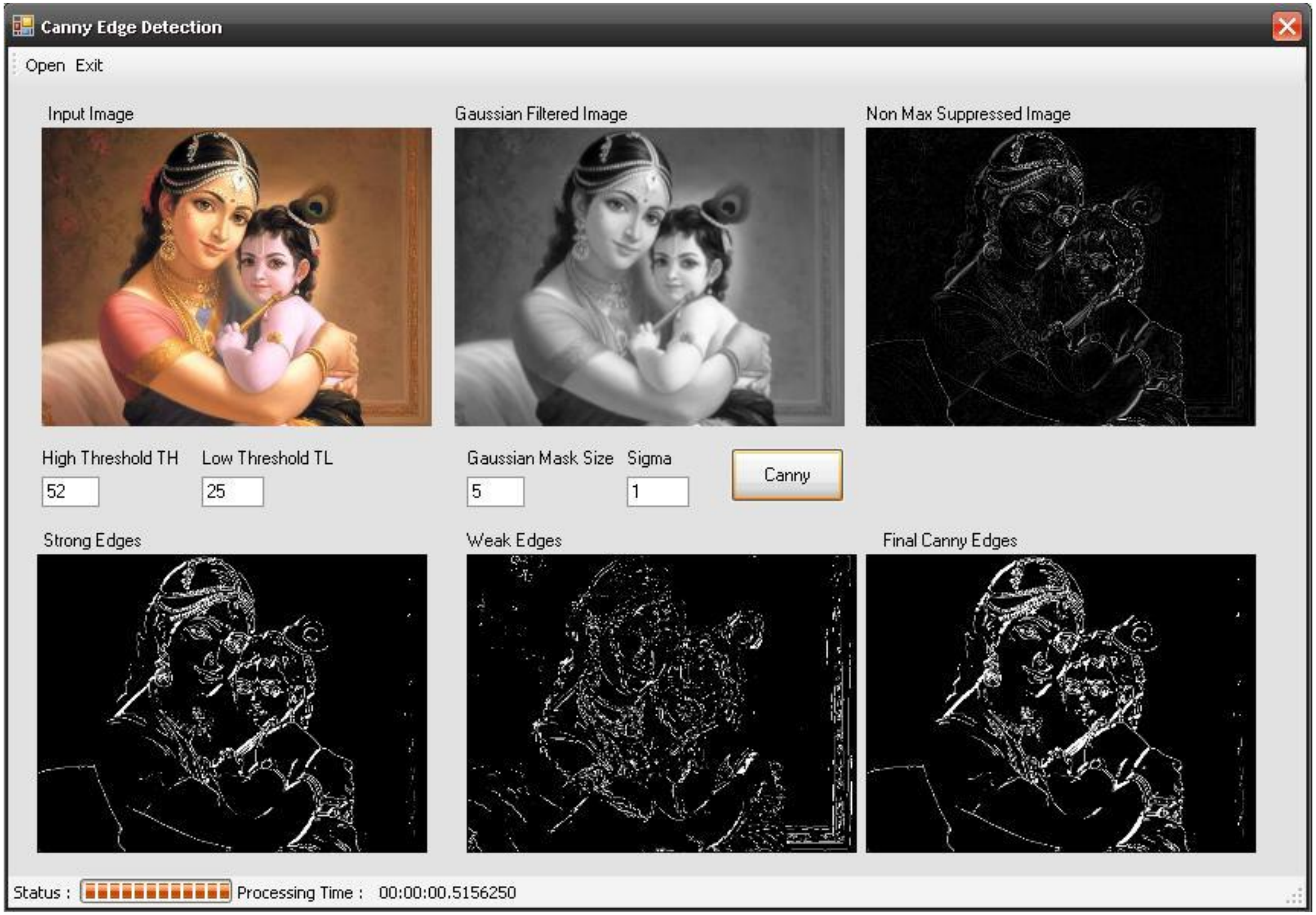


Result of using a threshold approximately equal to 4% of the maximum value of the LoG image.

The canny edge detector

- This complex algorithm is superior to the algorithms seen so far.
- It is based on the following objectives.
- Low error rate : edges are as close as possible to true edges.
- Edge point should be well localized : distance between the point marked as an edge by the detector and the center of the true edge should be minimum.
- Single edge point response : Detector should not identify multiple edge pixels where only a single edge point exist.

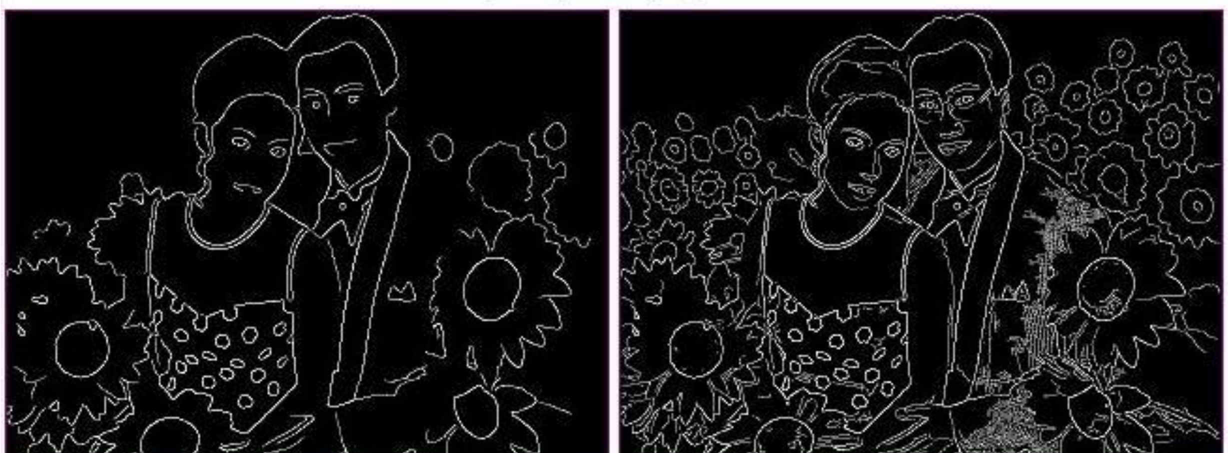
The canny edge detector



The canny edge detector

Canny Edge Detection.

Sigma= 1, Low= 0.4, High= 0.8



Sigma= 2, Low= 0.4, High= 0.8

Sigma= 1, Low= 0.3, High= 0.7