Edge Detection

Delivered by

Dr. Subarna Chatterjee
subarna.cs.et@msruas.ac.in

Edge Detection

At the end of this session, student will be able to

– Describe the importance of Edge Detection

– Identify types and methods of Edge Detection

Session Topics

- Detection of Discontinuities,
- Point Detection,
- Line Detection,
- Edge Detection
- Edge Detectors
  - Roberts
  - Prewitt
  - Sobel
  - Canny
  - Laplacian

Detection of Discontinuities

- Three basic types of gray level discontinuities in a digital image:
  - Points
  - Lines
  - Edges
- Most common way is to run a mask through the image. This procedure involves computing SOP of the coeff. with the image in the region encompassed by the mask.
- The response of the mask w_i (3x3) at any point in the image is:

$$R = \sum_{i=1}^{9} w_i z_i \qquad \text{.......(1)}$$

where, z_i is gray value of the pixel associated with mask coefficient w_i.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Point Detection

- Isolated Point: a point whose gray level is significantly different from its background
- Using the mask shown in the figure, we can say that a point has been detected at the location on which the mask is centered if

$$|R| \geq T \qquad \text{.......................(2)}$$

Where T is a nonnegative threshold

- Measures the weighted differences between the center point and its neighbors

mask

-1	-1	-1
-1	8	-1
-1	-1	-1

Point Detection Example

mask

-1	-1	-1
-1	8	-1
-1	-1	-1

*

original image

1	1	1	1	1
1	10	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

=

convolved image

-	-	-	-	-
-	72	-9	0	-
-	-9	-9	0	-
-	0	0	0	-
-	-	-	-	-

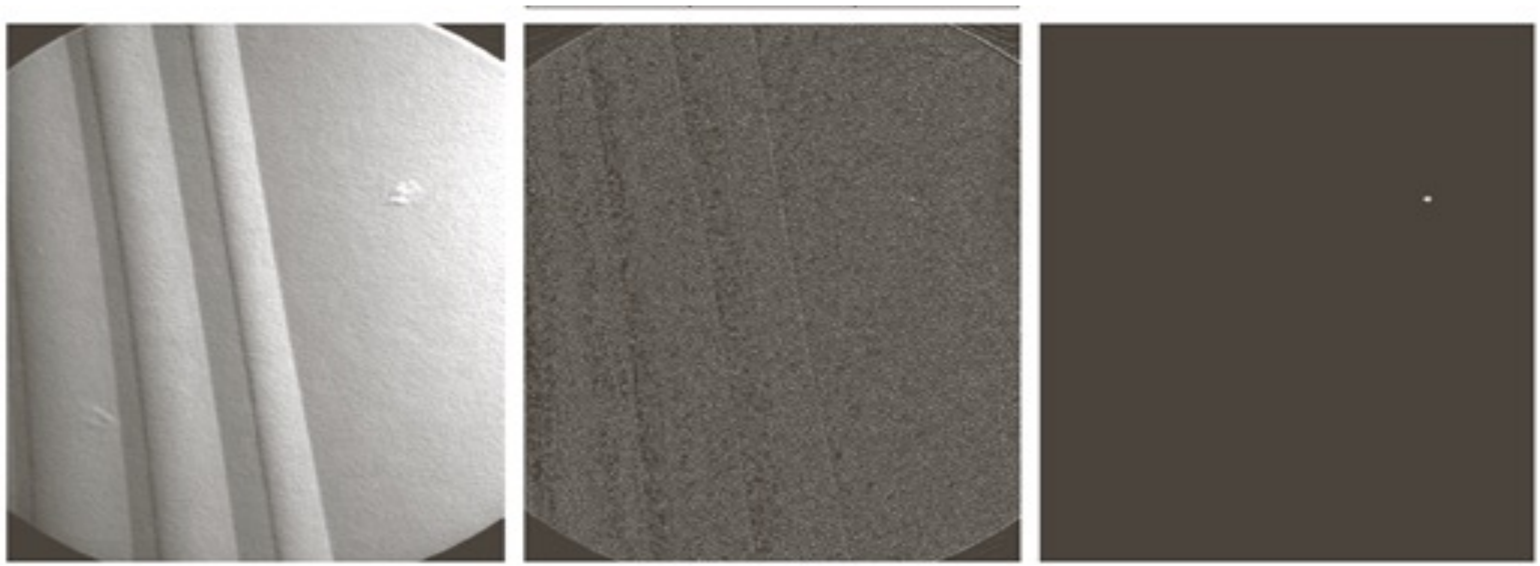
Depending on the value of *T* we can get:

- 4 points for (T >= |9|)
- 1 point for (T >9)
- 0 points for (T > 72)

Point Detection Application

mask

-1	-1	-1
-1	8	-1
-1	-1	-1



- X-ray image of turbine blade with a porosity.
- Result(point detection) of convolving the mask with the image.
- Result (of using eq. 2) showing a single point.

Background

- 1st -order derivative

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- 2nd -order derivative

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

Characteristics of 1^{st} and 2^{nd} Order Derivatives

- 1^{st}-order derivatives generally produce thicker edges in image
- 2^{nd}-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- 2^{nd}-order derivatives produce a double-edge response at ramp and step transition in intensity
- The sign of the 2^{nd} derivative can be used to determine whether a transition into an edge is from light to dark or dark to light

Detection of Isolated Points

- The Laplacian

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

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)$$

$$-4f(x,y)$$

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases} \qquad R = \sum_{k=1}^{9} w_k z_k$$

Line Detection

- 2^{nd} derivatives to result in a stronger response and to produce thinner lines than first derivatives

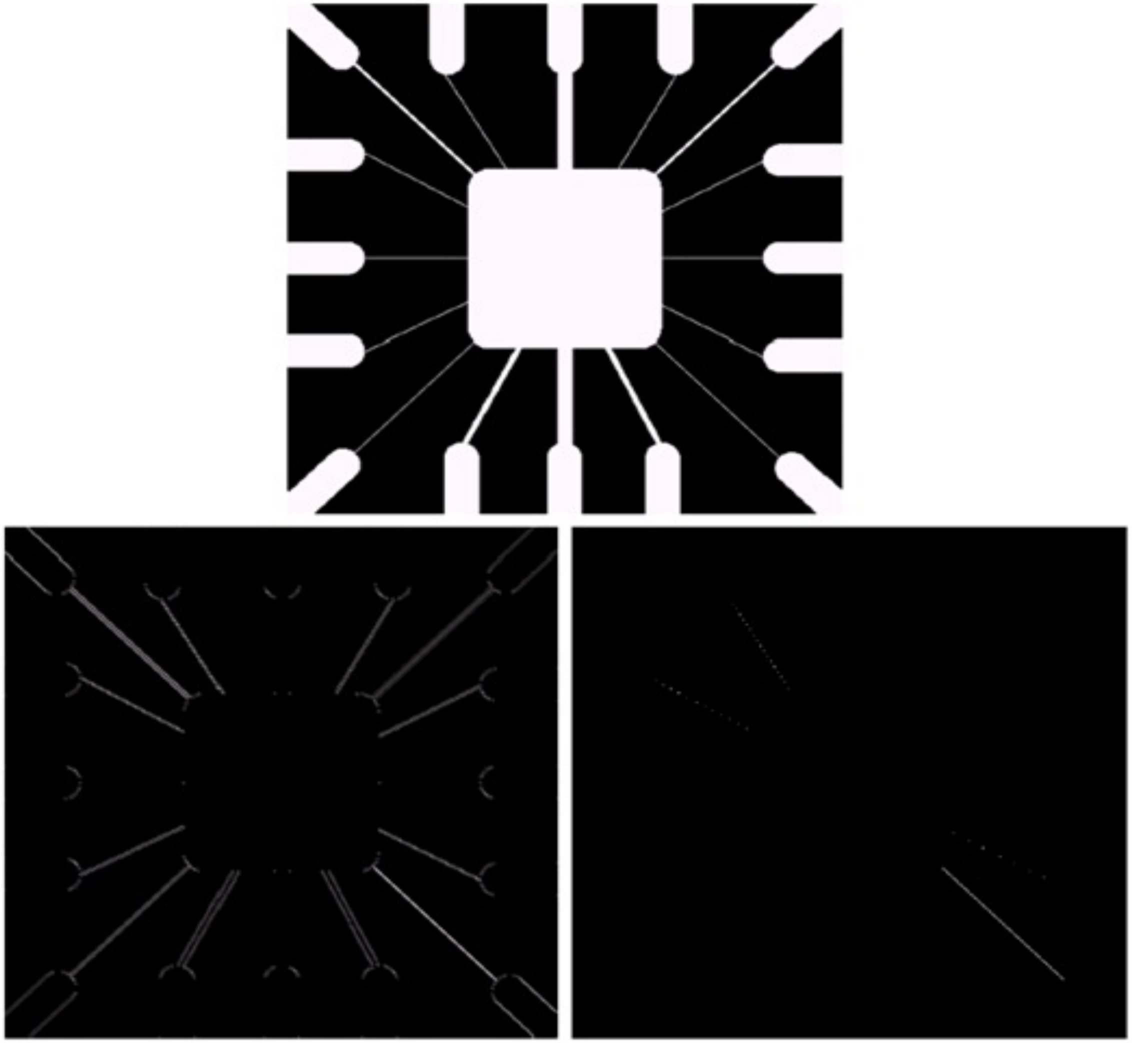
- Double-line effect of the second derivative must be handled properly

Line detection in Specified Directions

- Lines can be detected using appropriate masks
- If the 1st mask is moved around an image then it would respond to lines one pixel thick oriented horizontally
- Let R_1, R_2 , R_3 and R_4 denote responses of the masks in figure below
- 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 a line in the direction of mask k.
- For a certain point in the image, if $|R_1| > |R_2|$, then the point is more likely associated with a line in the direction of mask R_1
- To detect the lines in an image in a particular direction, use the corresponding mask and threshold the absolute value of the result

−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
	Horizontal			+45°			Vertical			−45°	

Line Detection Example



- Binary wire bound mask
- Absolute value of result after processing with -45^{o} line detector
- Result of thresholding image
- Threshold can be the maximum value of the image

Line Detection


natural image


grayscale


Applying
Edge detector


Applying
line detector

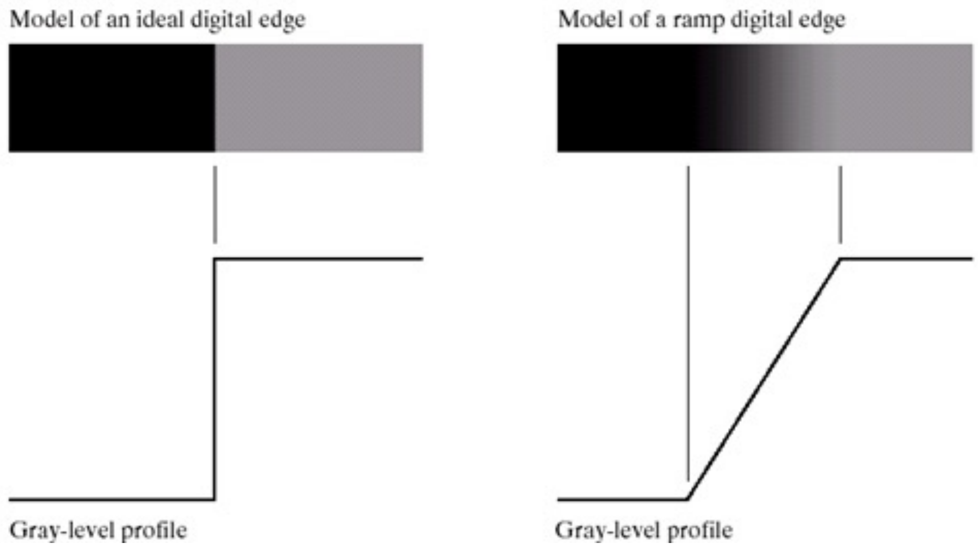
- The line detector gives a poor result compared to the edge detector because
  - there are few single pixel width lines in this image and therefore the detector is responding to the other high spatial frequency image features (*i.e.* edges, thick lines and noise)

Edge Detection

- **Edge detection** is a terminology in IP and CV, particularly in the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities
- Edges are pixels where the brightness function changes abruptly
- Edges are formed from pixels with derivative values that exceed a preset threshold
- The idea of an edge is a local concept that is based on the measure of gray level discontinuity at a point
- Edge detection operators are often implemented with convolution masks and most are based on discrete approximations to differential operators

Edge

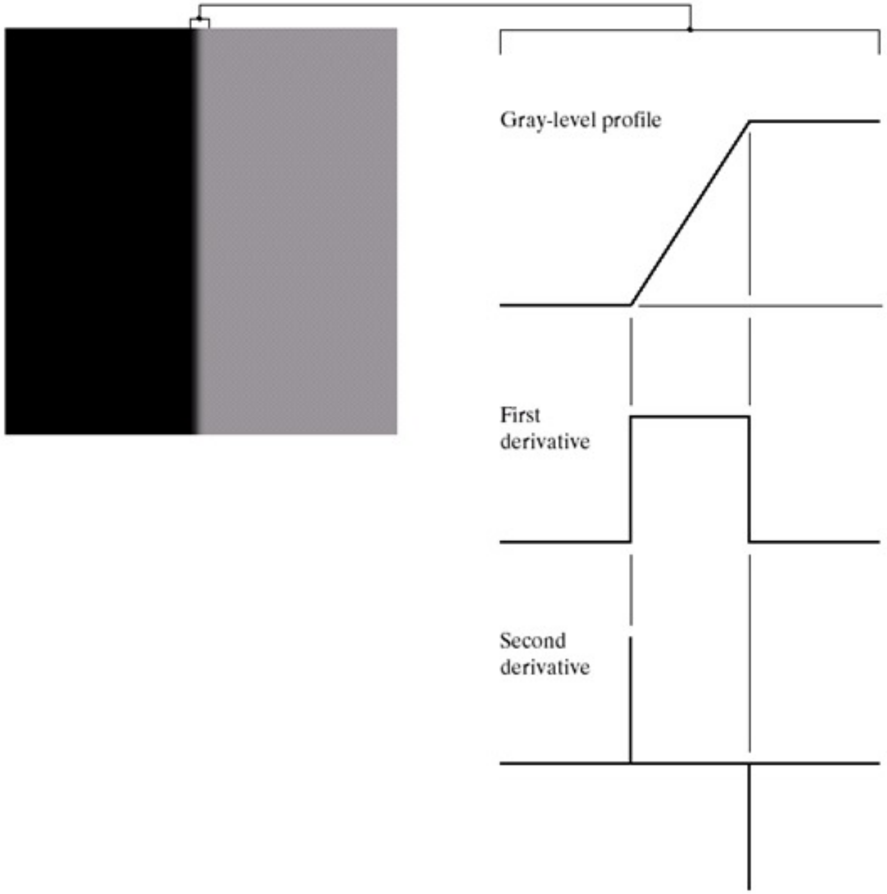
- An edge is a set of connected pixels that lie on the boundary between two regions
- Ideal edge is a set of connected pixels, each of which is located at an orthogonal step transition in gray level
- Considering practical situations, sampling and other image acquisition imperfections yield edges that are blurred. As a result edges have ramp-like profile



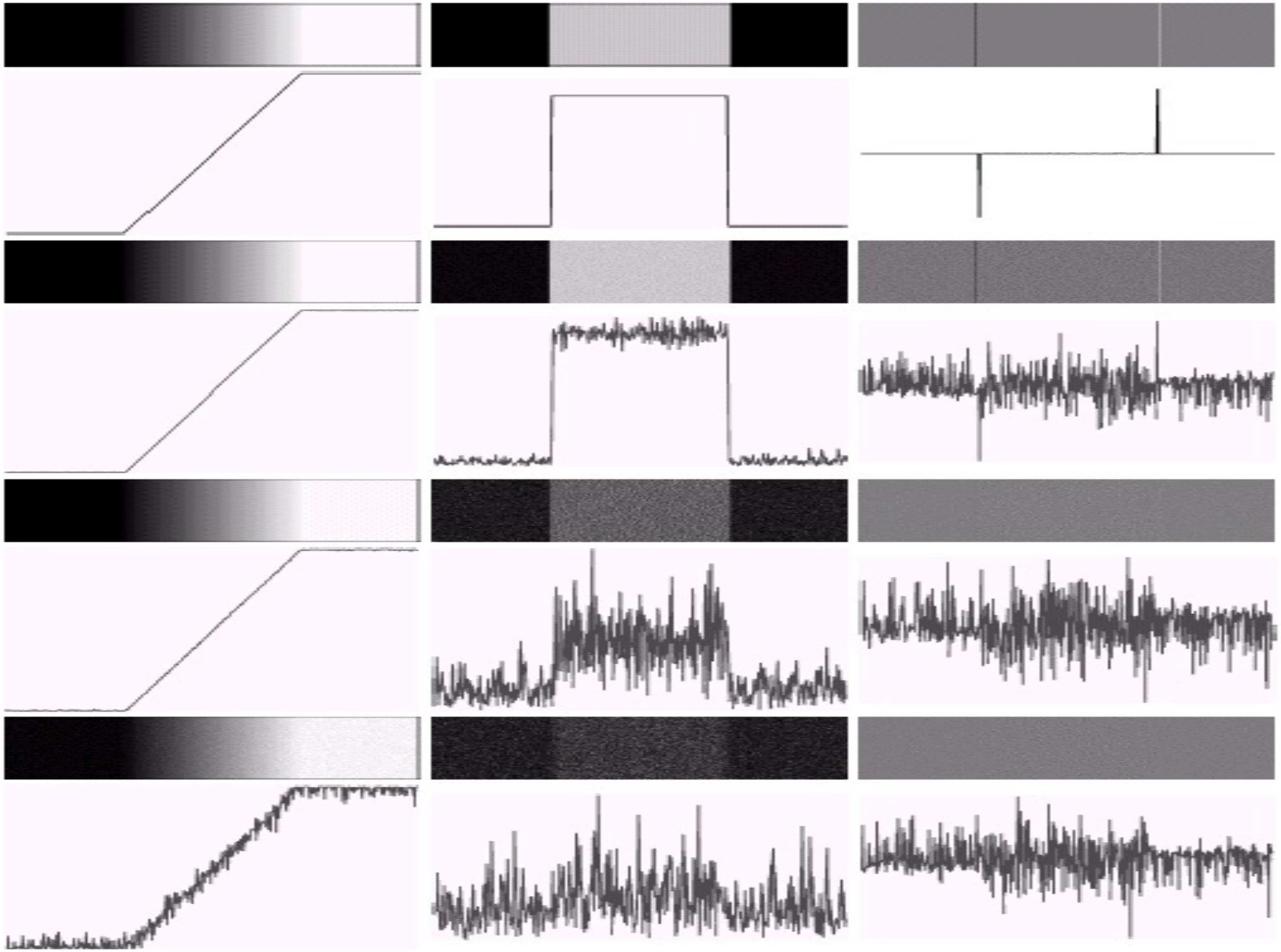
- Slope of ramp is inversely proportional to degree of blurring.

Edges and their derivatives

- 1^{st} derivative of the edge is:
  - +ve at the points of transition into and out of the ramp.
  - Constant for points in the ramp.
  - Zero in areas of constant gray level.
  - Used to detect the presence of an edge at a point.
- 2^{nd} derivative of the edge is:
  - +ve at the transition associated with the dark side of the edge.
  - -ve at the transition associated with the light side of the edge.
  - Zero along the ramp and in areas of constant gray level.
  - Thus, produces 2 values for every edge and the line joining the extreme points determines midpoint.



Edges Corrupted by Noise



- 1st column: Image and gray level profile, corrupted by random Gaussian noise of variance 0, 0.1, 1 and 10
- 2nd column: First derivatives
- 3rd column: Second derivatives

Gradient

- Edge detection operators are often implemented with convolution masks and most are based on discrete approximations to gradient operators.

- The gradient of an image f(x, y) at coordinates (x, y) is defined by the vector:

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

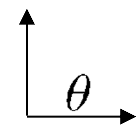
- Represents direction of most rapid change in intensity

Gradient

- Gradient direction: The angle theta represents the direction angle of the gradient at (x,y) with respect to x-axis.

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

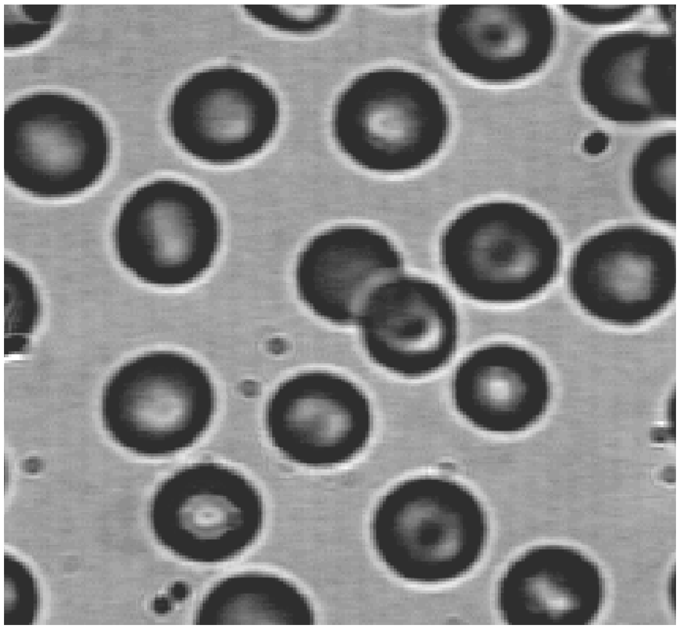
- The direction of the edge at (x,y) is perpendicular to the direction of the gradient vector at that point.
- The *edge strength* is given by the gradient magnitude



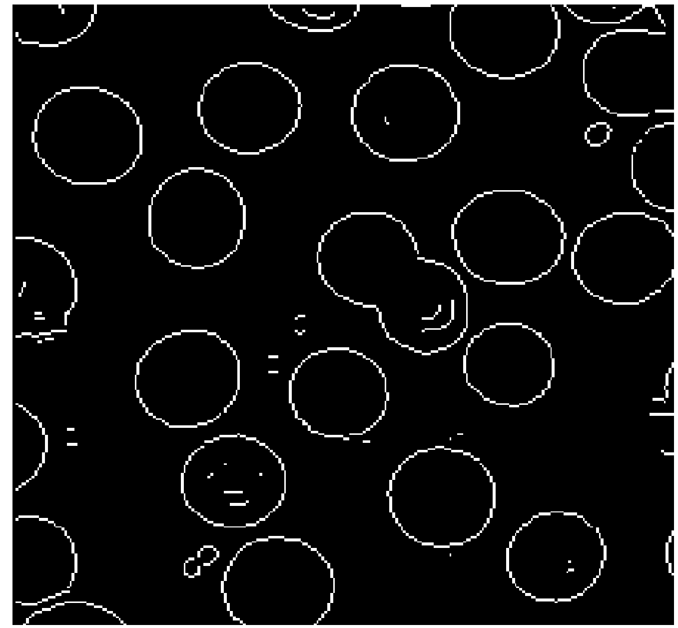
$$||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

This quantity gives the maximum rate of increase of f(x,y) per unit distance in the direction of the gradient

Edge Detection example



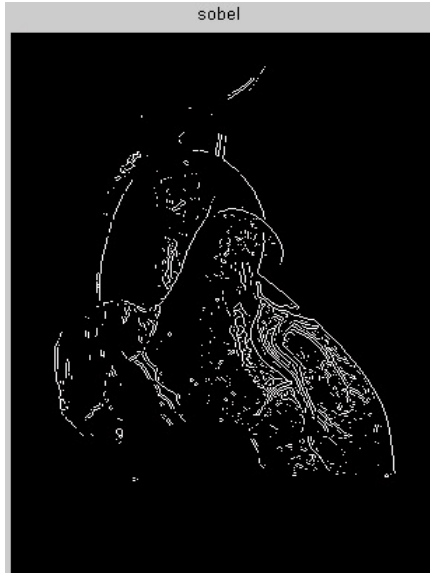
Edge detection

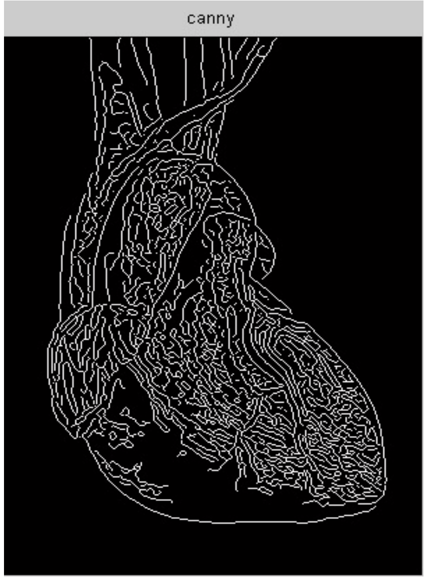


Why are we Interested in Edges?

- Edges can many times represent the information in the image about the objects
- A higher level of abstraction i.e. less data to process!
- Edges are features:
  - "Independent" from illumination. As opposed to e.g., color information
- Object recognition and detection, many times use edge features
  - This is true for electronic as well as for biological vision systems!

Common Types of Edge detectors





- Roberts, Prewitts, Sobel, Canny

Robert Edge Operator

- This is the simplest edge operator operates on a (2x2) region of pixels. This operator uses the diagonal derivatives to estimate the gradient of a point.

$$\begin{vmatrix} f(x, y) & f(x+1,y) \\ f(x,y+1) & f(x+1,y+1) \end{vmatrix}$$

- 2 x 2 image region

- The value of Gx and Gy is given by

$$Gx = f(x, y) - f(x+1, y+1)$$
$$Gy = f(x+1, y) - f(x, y+1)$$

1	0		0	1
0	−1		−1	0
G_x			G_y	

Sobel Edge Operator

- Sobel edge detection uses 'first order gradients' of intensity gradient
- First order intensity gradients can be calculated using spatial convolution kernels, which are known as 'Sobel Operators'
- The Sobel operators have the advantage of providing both a differencing and a smoothing effect
- Because derivatives enhance noise, the smoothing effect is a particularly attractive feature of the Sobel Operators
- Sobel X operator gives gradients in horizontal direction and Sobel Y operator gives gradients in vertical direction
- Sobel operator operates on a (3X3) region of pixels

a	b	c
d	(x,y)	f
g	h	i

Sobel Edge Operator

- Sobel X-operator, Gx and Sobel Y-operator Gy can be written as:

$$G_x = (a + 2d + g) - (c + 2f + i)$$

$$G_y = (g + 2h + i) - (a + 2b + c)$$

The corresponding masks may be written as

1	0	−1		−1	−2	−1
2	0	−2		0	0	0
1	0	−1		1	2	1
Sobel X Operator				Sobel Y Operator		

- These kernels are applied to the pixels of an image
- Target pixel is placed in the middle of the kernels and pixels around it are multiplied by the kernels' factors
- The X or Y gradient is the summation of all nine values of the kernel

Prewitt Edge Operator

- This operator also operates on (3x3) image region

a	b	c
d	(x,y)	f
g	h	i

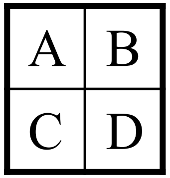
The value of Gx and Gy is given by

$$G_x = (c + f + i) - (a + d + g)$$

$$G_y = (g + h + i) - (a + b + c)$$

-1	0	1		-1	-1	-1
-1	0	1		0	0	0
-1	0	1		1	1	1
G_x				G_y		

Edge Detection



A: Original image

B: Component of the gradient in x-direction

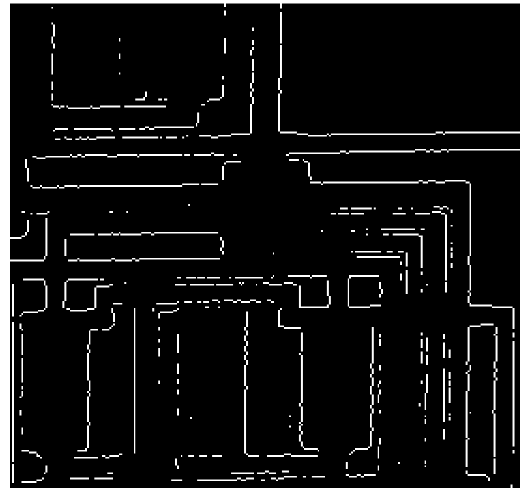
C: Component of the gradient in y-direction

D: Gradient image (sum of above two)

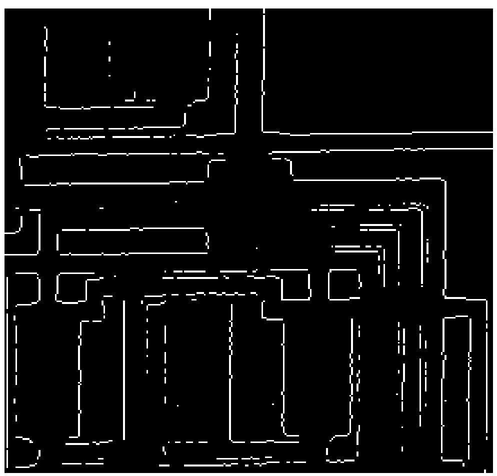
Results of Different Edge Detectors

original image

sobel edge image



prewitt edge image



roberts edge image

Laplace Edge Detector

- The following example shows a 3x3 Laplace Operator

$$\begin{bmatrix} (x-1,y-1) & (x, y-1) & (x+1, y-1) \\ (x-1, y) & (x, y) & (x+1, y) \\ (x-1,y+1) & (x, y+1) & (x+1, y+1) \end{bmatrix}$$

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

$$Where \ \frac{\partial^2 f}{\partial x^2} = [f(x-1,y) - f(x,y)] - [f(x,y) - f(x+1,y)]$$

$$= f(x-1,y) + f(x+1,y) - 2f(x,y)$$

Laplace Edge Detector

- The corresponding mask is given by

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

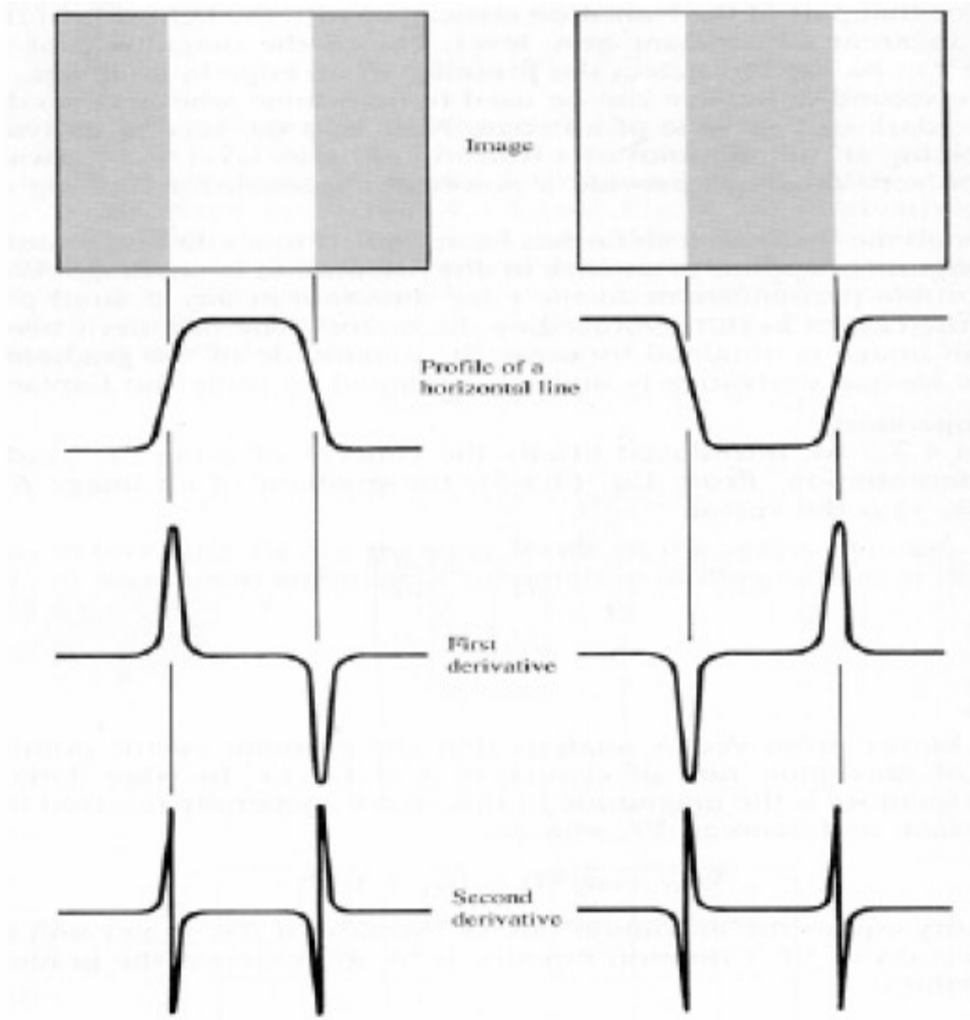
- This kernel is applied to the pixels of an image, where the target pixel is placed in the middle of the kernels
- Target pixel and those around it are multiplied by the kernels' factors

Laplace Edge Detector

- Because the Laplacian responds to transitions in intensity, it is seldom used in practice for edge detection

- As a second-order derivative, the Laplacian typically is unacceptably sensitive to noise

- Moreover, the Laplacian produces double edges and is unable to detect edge direction



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

- **Criterion 3**: Single Response Constraint: The detector must return one point only for each edge point

Canny Edge Detector

- The Canny edge detection algorithm is known to many as the **Optimal edge detector**
- Canny's intentions were to enhance the many edge detectors already out at the time he started his work
- In order to implement the canny edge detector algorithm, a series of steps must be followed
  - Convolution with derivative of Gaussian
  - Non-maximum Suppression
  - Hysteresis Thresholding

Cont..

- The **Canny edge detector** is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.
- It was developed by John F. Canny in 1986. Canny also produced a *computational theory of edge detection* explaining why the technique works.
- The general criteria for edge detection include:
  - Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
  - The edge point detected from the operator should accurately localize on the center of the edge.
  - A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

Cont..

- The Canny operator works in a multi-stage process.

1. First of all the image is smoothed by **Gaussian convolution**.
2. Then a simple 2-D first derivative operator (somewhat like the **Roberts Cross**) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives.
3. Edges give rise to ridges in the **gradient magnitude** image.
4. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as **non-maximal suppression**.
5. The tracking process exhibits **hysteresis** controlled by two thresholds: ***T1*** and ***T2***, with ***T1 > T2***. Tracking can only begin at a point on a ridge higher than ***T1***. Tracking then continues in both directions out from that point until the height of the ridge falls below ***T2***. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

Process of Canny edge detection algorithm

- The Process of Canny edge detection algorithm can be broken down to 5 different steps:
  - Apply Gaussian filter to smooth the image in order to **remove the noise**
  - Find the **intensity gradients** of the image
  - Apply **non-maximum suppression** to get rid of spurious response to edge detection
  - Apply **double threshold** to determine potential edges
  - **Track edge** by **hysteresis**: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges**.**

Gaussian filter

- Since all edge detection results are easily affected by the noise in the image, it is essential to filter out the noise to prevent false detection caused by it. To smooth the image, a Gaussian filter kernel is convolved with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector. The equation for a Gaussian filter kernel of size $(2k+1)\times(2k+1)$ is given by:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \le i, j \le (2k+1)$$

- Here is an example of a 5×5 Gaussian filter, used to create the adjacent image, with σ = 1. (The asterisk denotes a convolution operation.)

$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

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

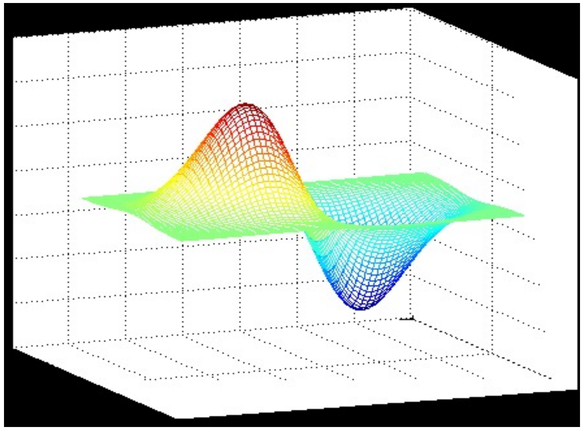
$$\nabla S = \left[\frac{\partial}{\partial x} S \quad \frac{\partial}{\partial y} S \right]^T = \left[S_x \quad S_y \right]^T$$

- Compute gradient magnitude and orientation

$$|\nabla S| = \sqrt{S_x^2 + S_y^2} \qquad \theta = \tan^{-1} \frac{S_y}{S_x}$$

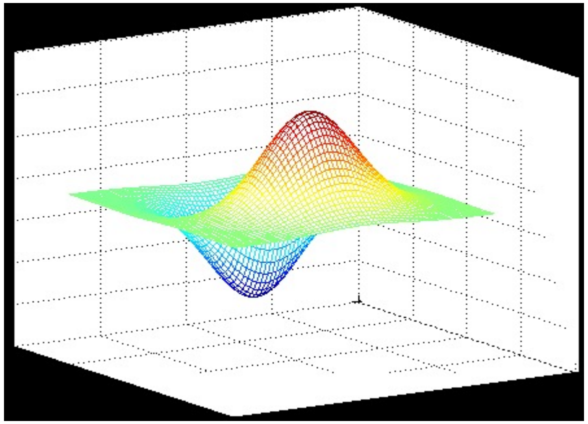
Canny Edge Operator

$$\nabla S = \nabla(G_\sigma * I) = \nabla G_\sigma * I$$



$$\nabla G_\sigma = \begin{bmatrix} \dfrac{\partial G_\sigma}{\partial x} & \dfrac{\partial G_\sigma}{\partial y} \end{bmatrix}^T$$

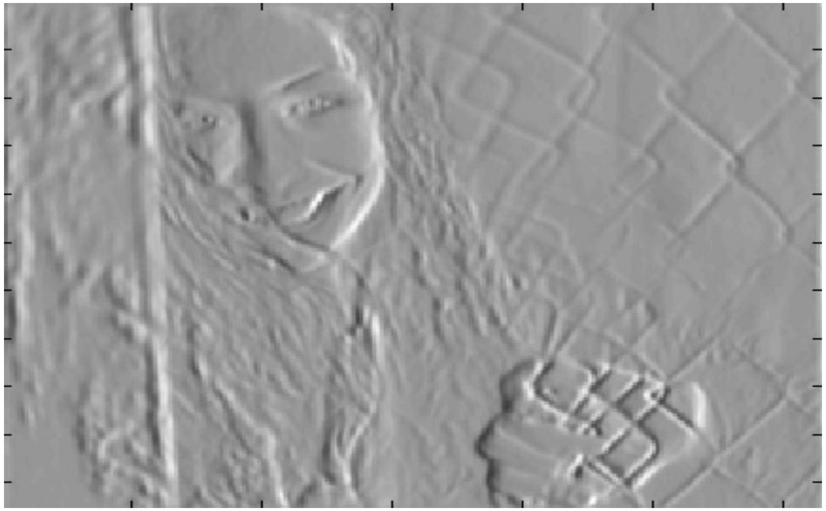
$$\nabla S = \begin{bmatrix} \dfrac{\partial G_\sigma}{\partial x} * I & \dfrac{\partial G_\sigma}{\partial y} * I \end{bmatrix}^T$$



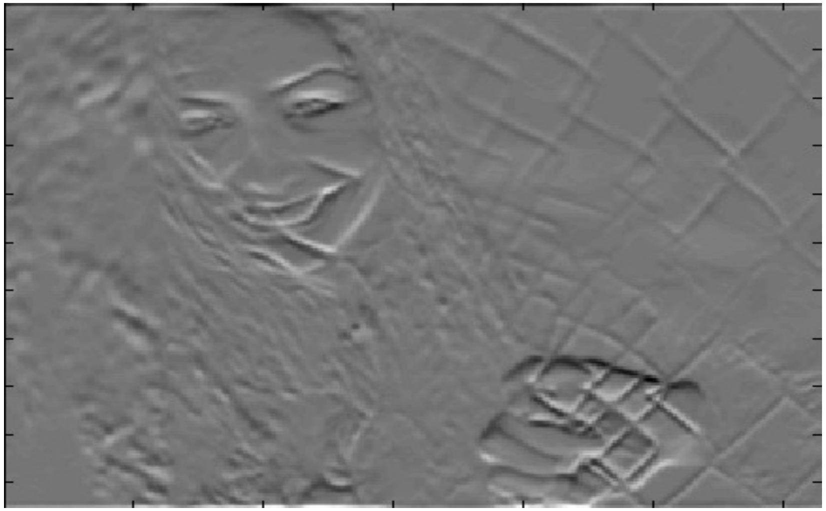
Canny Edge Detector

I

S_x



S_y



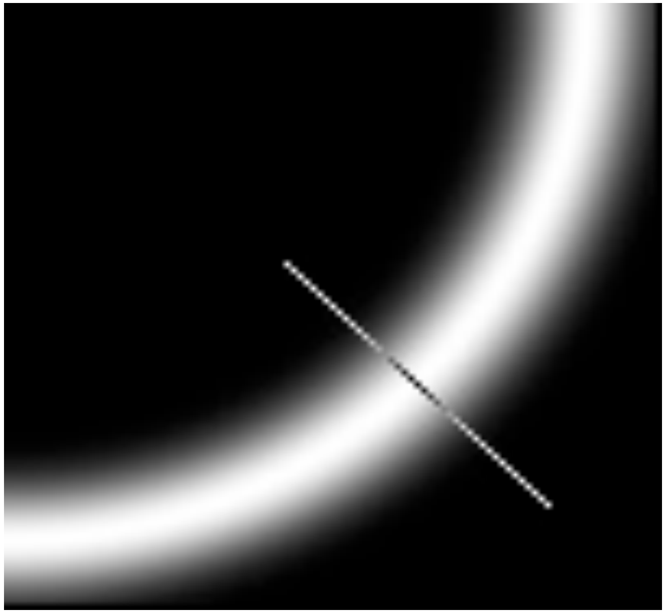
Canny Edge Detector

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

I

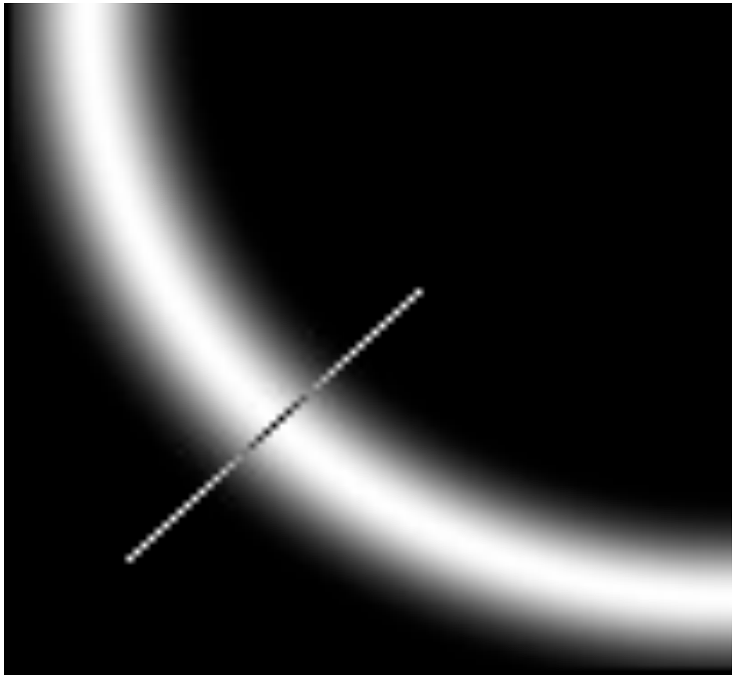
$$|\nabla S| \geq Threshold = 25$$

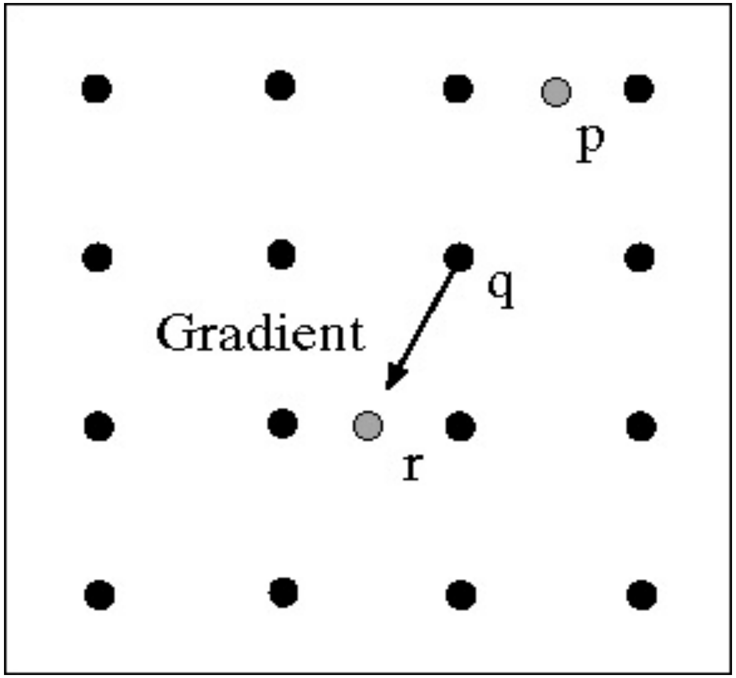
Non-Maximum Suppression



- 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?

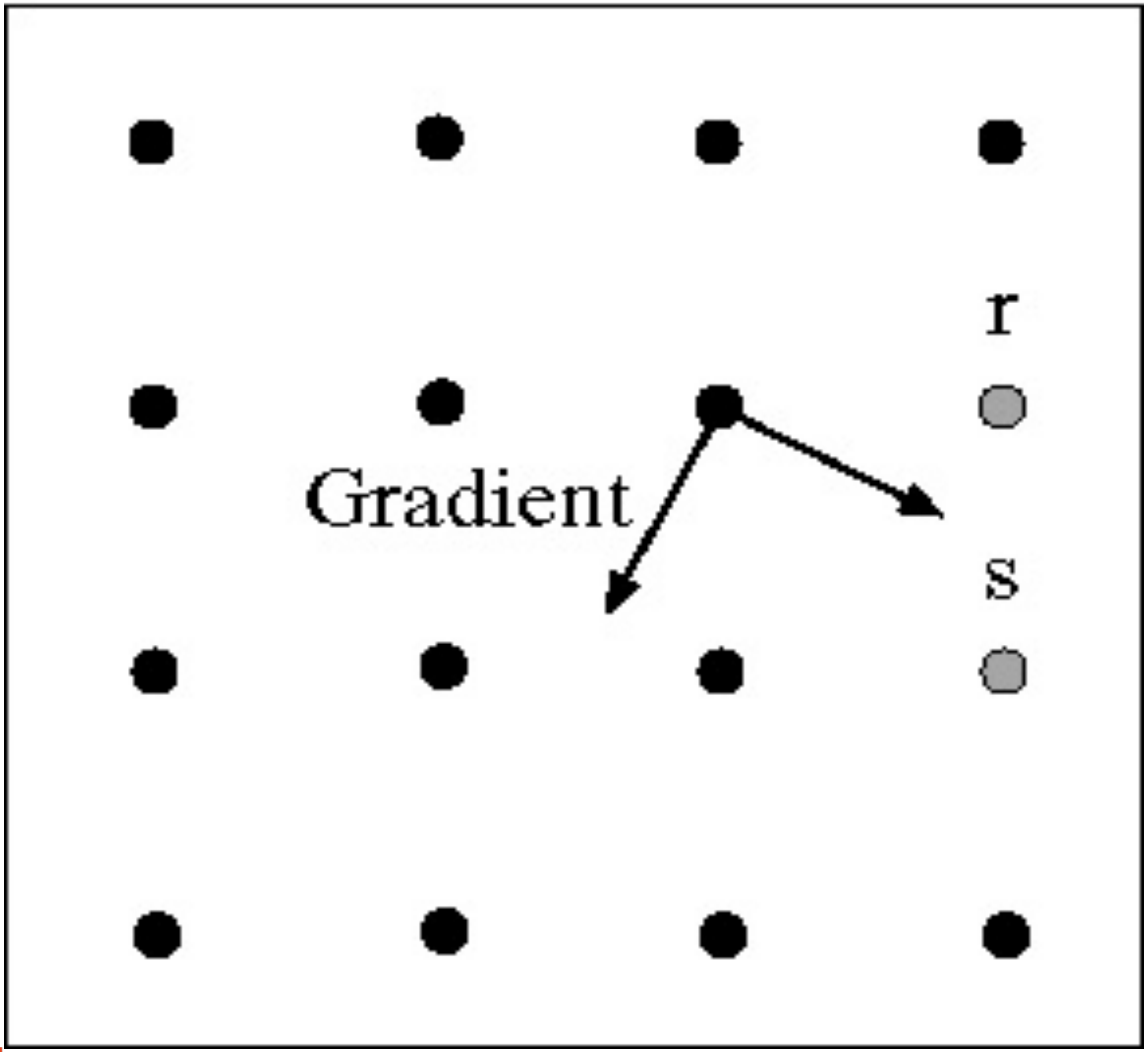
Non-Maximum Suppression





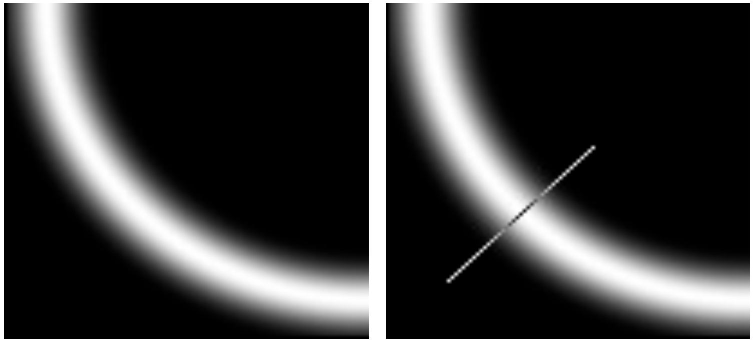
- Check if pixel is local maximum along gradient direction
  - requires checking interpolated pixels p and r

Non-maximum Suppression



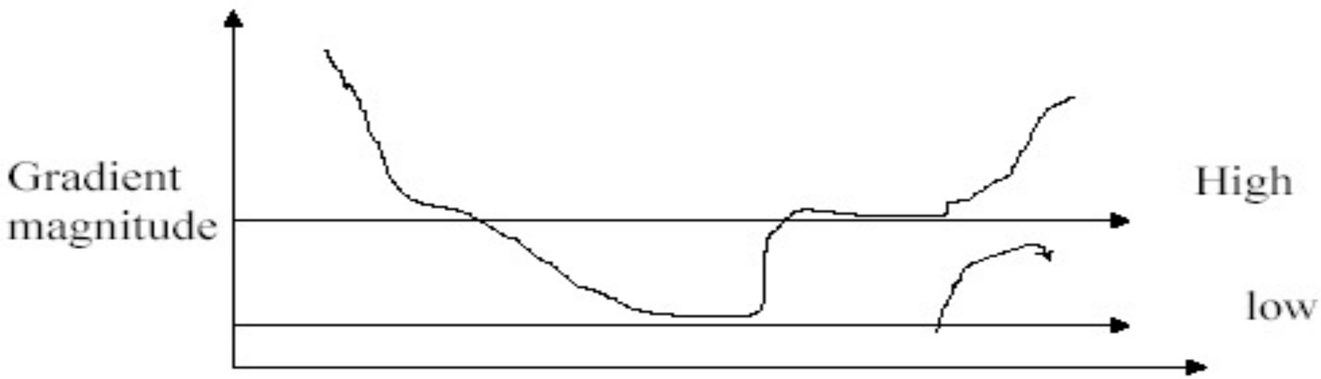
Predicting the next edge point

- Assume the marked point is an edge point
- Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).



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’



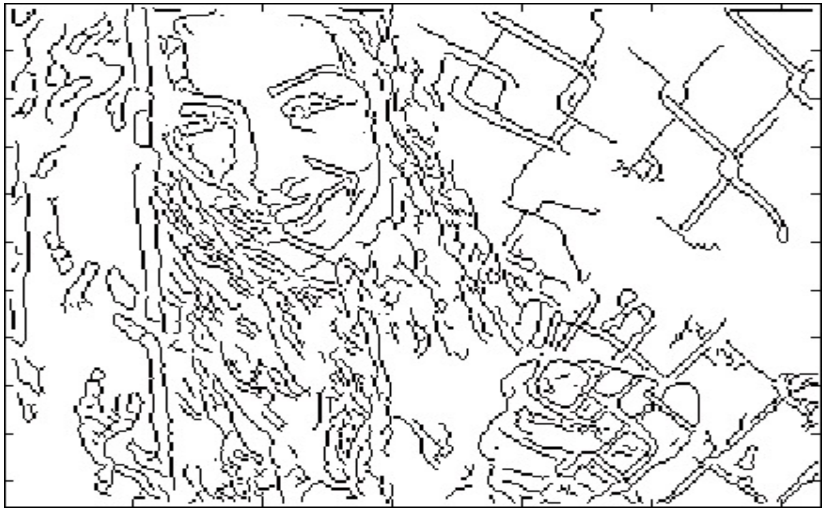
Hysteresis Thresholding

M

$M \geq Threshold = 25$

$High = 35$

$Low \ = 15$



Canny Edge Detection - Steps

- <u>Step 1:</u> Filter out any noise in the original image using Gaussian filter before trying to locate and detect any edges

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figure 3 Discrete approximation to Gaussian function with σ=1.4

Canny Edge Detection

- <u>Step 2:</u> Find the edge strength by taking the gradient of the image using Sobel operator on the image. The approximate absolute gradient magnitude (edge strength) at each point can be found using

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

Gx

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

Gy

$$|G| = |Gx| + |Gy|$$

- <u>Step 3:</u> The formula for finding the edge direction is

  theta = $\tan^{-1}$ (Gy / Gx)

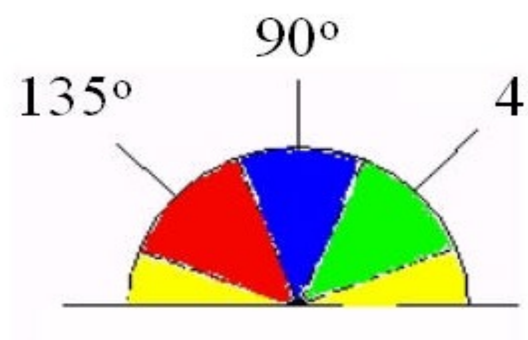
Canny Edge Detection

- **Step 4**: Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So, if the pixels of a 5x5 image are aligned as :

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

- Look at pixel "**a**", there are only four possible directions - **0 degrees, 45 degrees**, **90 degrees** or **135 degrees**

- Now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to



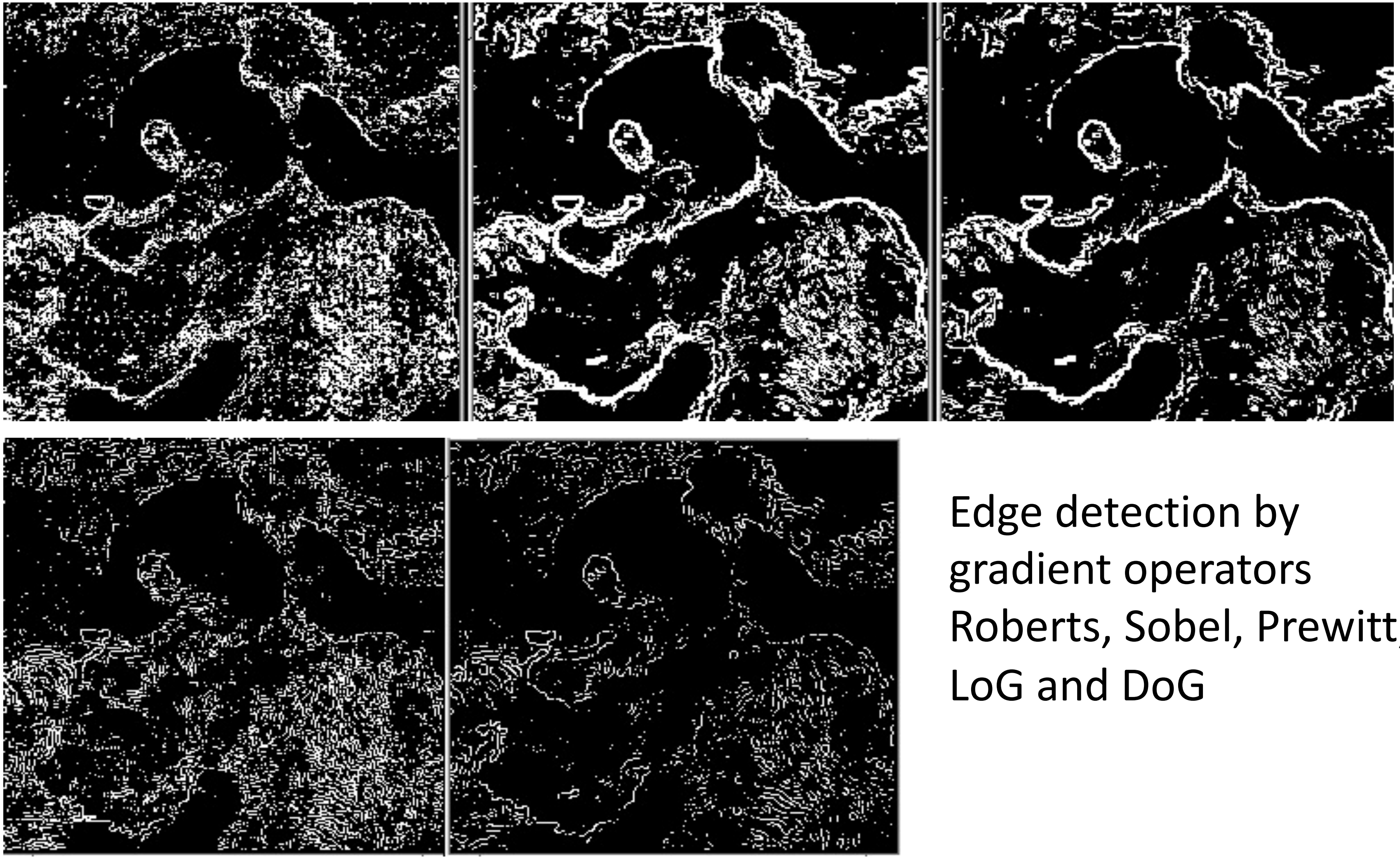
Canny Edge Detection

- **<u>Step 5</u>: A**fter the edge directions are known, non-maximum suppression has to be applied to trace along the edge, in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge which will give a thin line in the output image

- **<u>Step 6</u>**: Finally, hysteresis is used as a means of eliminating streaking which is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold.

Input Image

Canny Edge Detection



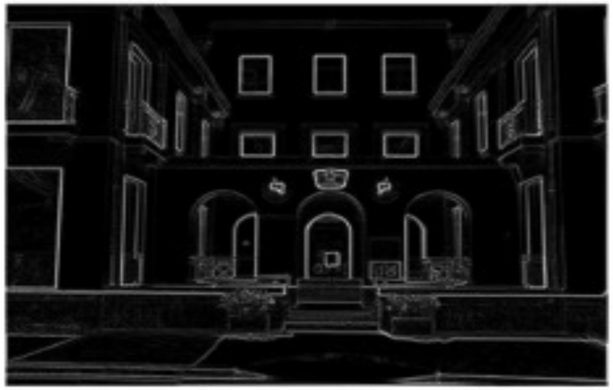
Edge detection by gradient operators Roberts, Sobel, Prewitt, LoG and DoG

Canny Edge Detection

Edge detection by Canny method for σ=1,2,3, τ_1=0.3, τ_2=0.7

Canny Edge Detection

Original image



Magnitude of gradient

Magnitude of gradient after non-maximal suppression

The Canny Edge Detector

Original Image (Lena)

Magnitude of the gradient

The Canny Edge Detector

After non-maximum suppression

Canny Edge Operator

original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

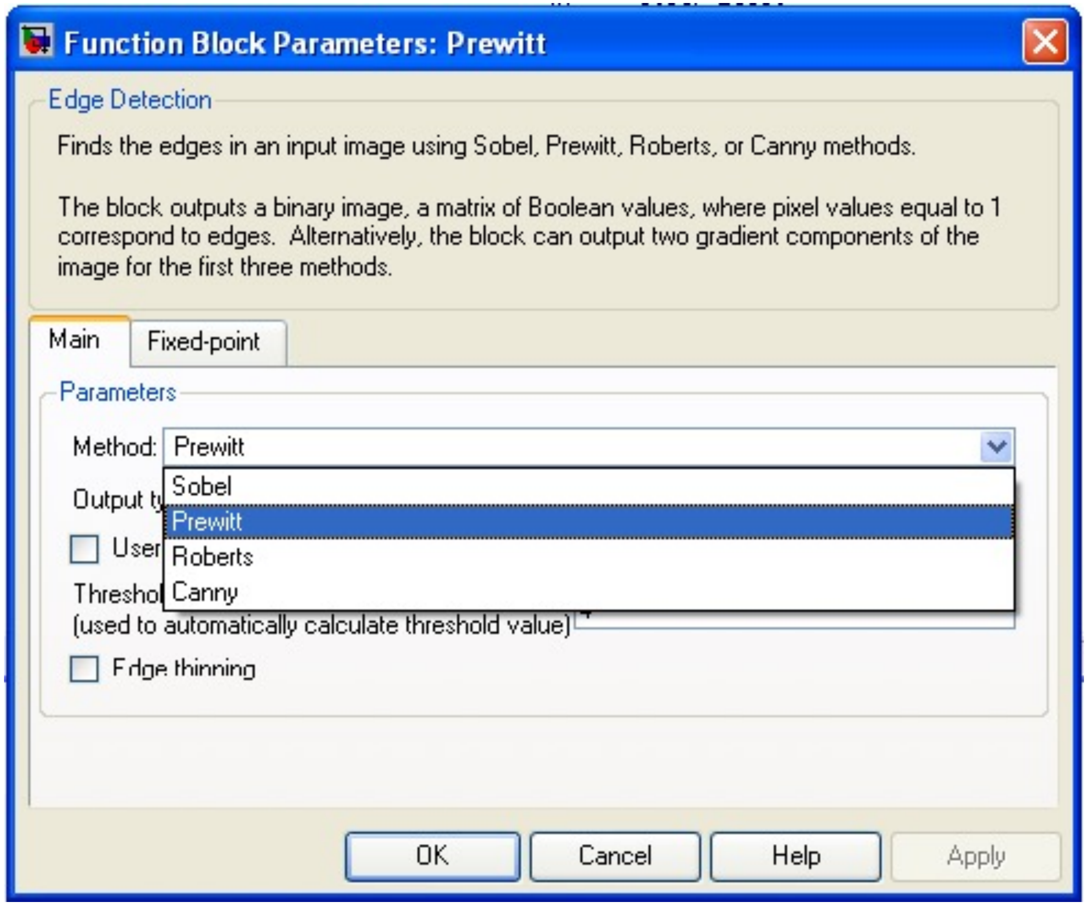
- The choice of σ depends on desired behavior
  - Large σ detects large scale edges
  - small σ detects fine features

Performance of Edge Detectors

- Objective and subjective evaluations can be useful
- Objective metrics allow us to compare different techniques with fixed analytical methods
- Subjective methods often have unpredictable results
- To develop a performance metric for edge detection operators, we need to consider the types of errors that can occur and define what constitutes success
- Success criteria use in development of Canny algorithm
  - *Detection* – find all real edges, no false edges
  - *Localization* – found in correct location
  - *Single response* – no multiple edges found for single edge

The Edge Detection Block

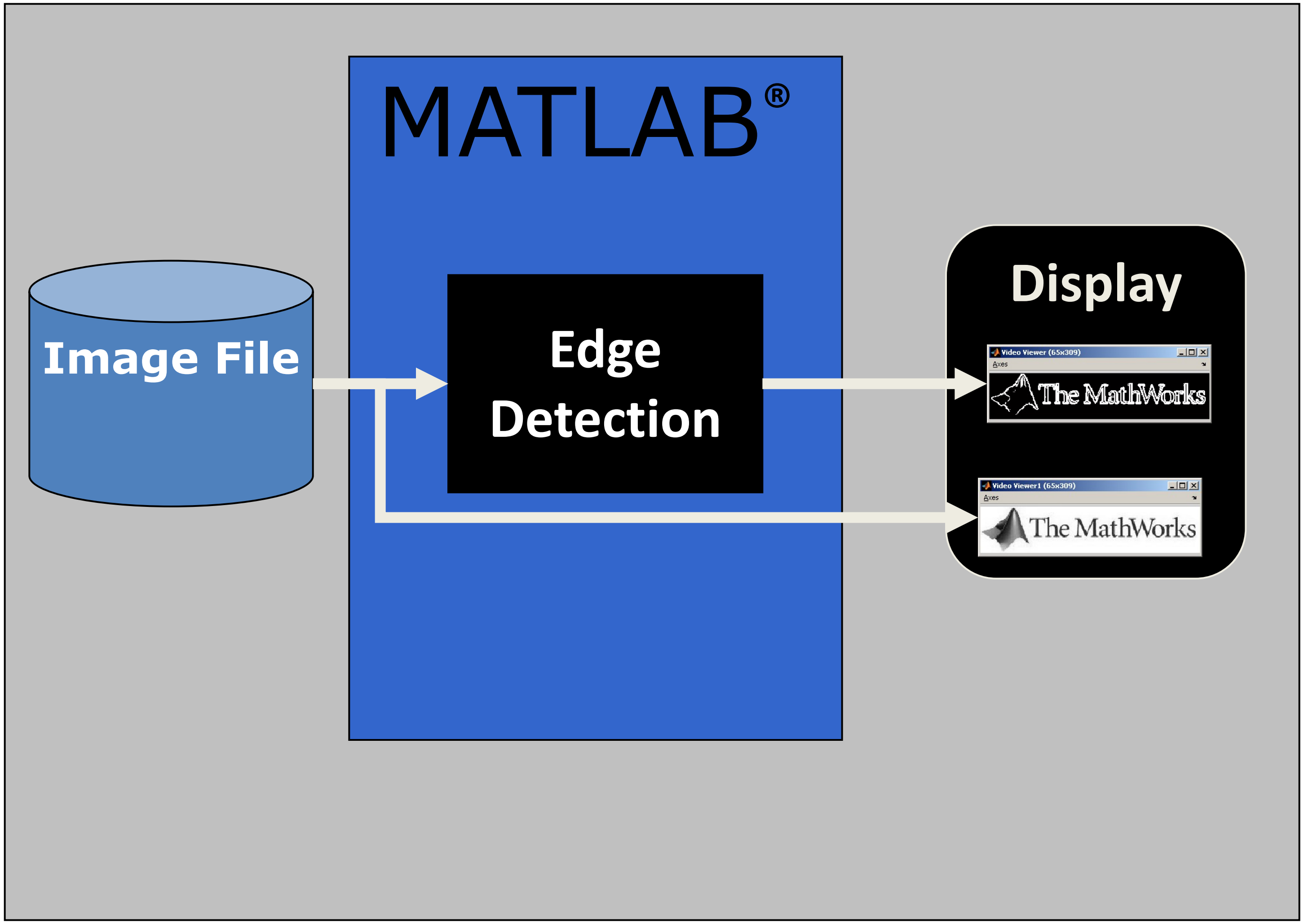
- The Edge Detection Block supports the four methods described in the pervious slides



Hands-On

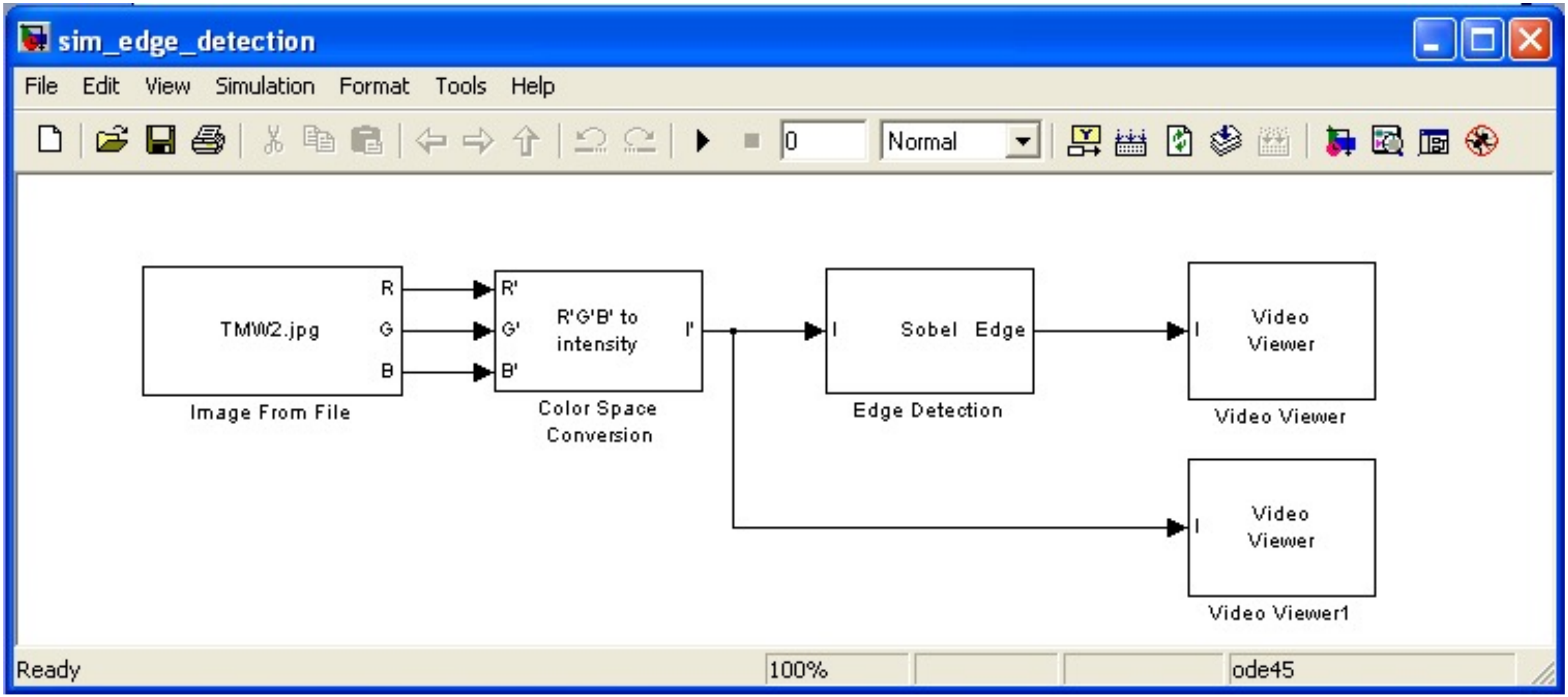
- Simulation
- Implementation using the DSK6416

Simulation

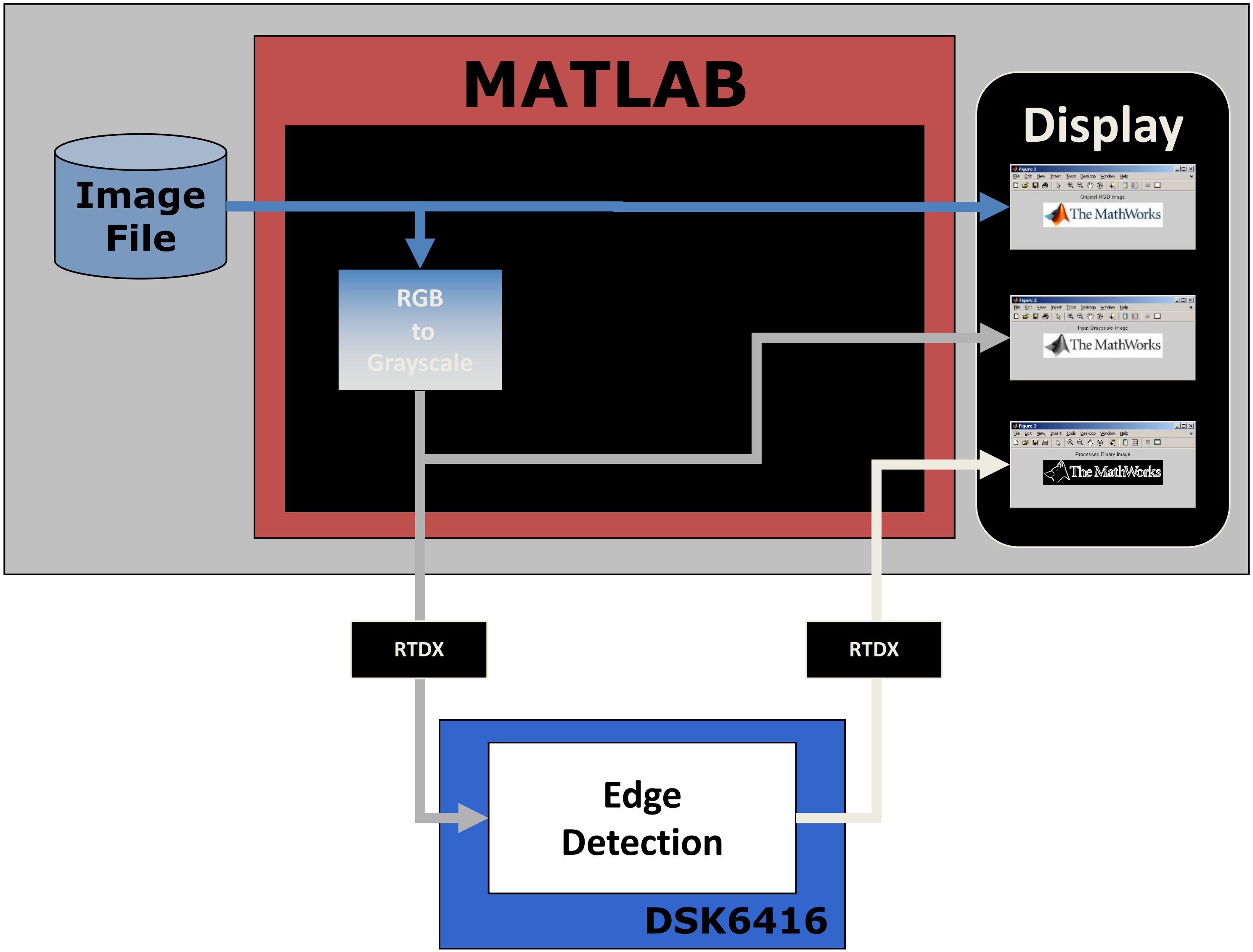


MATLAB®
Image File
Edge Detection
Display
The MathWorks
The MathWorks

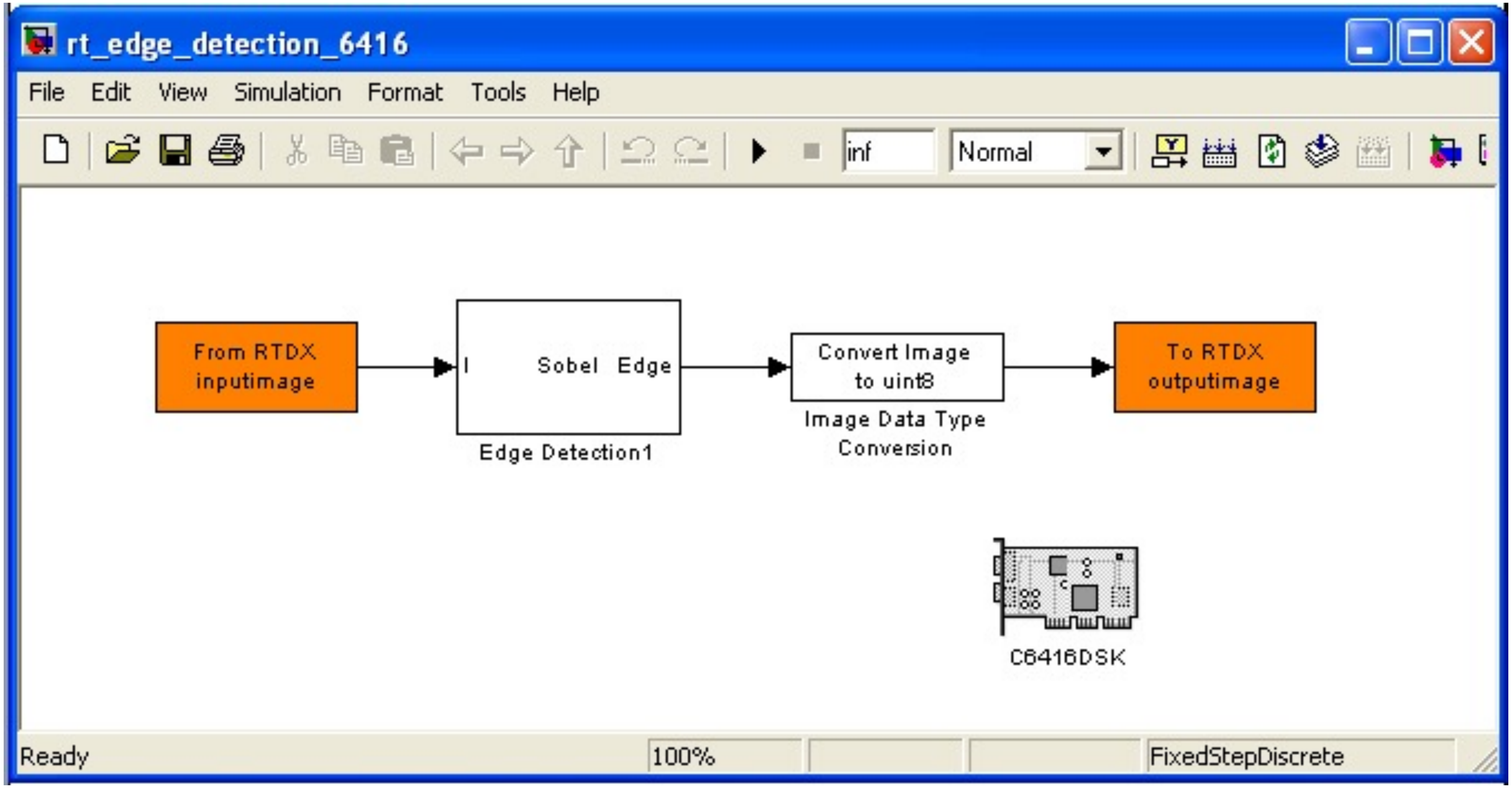
Edge Detection Simulation



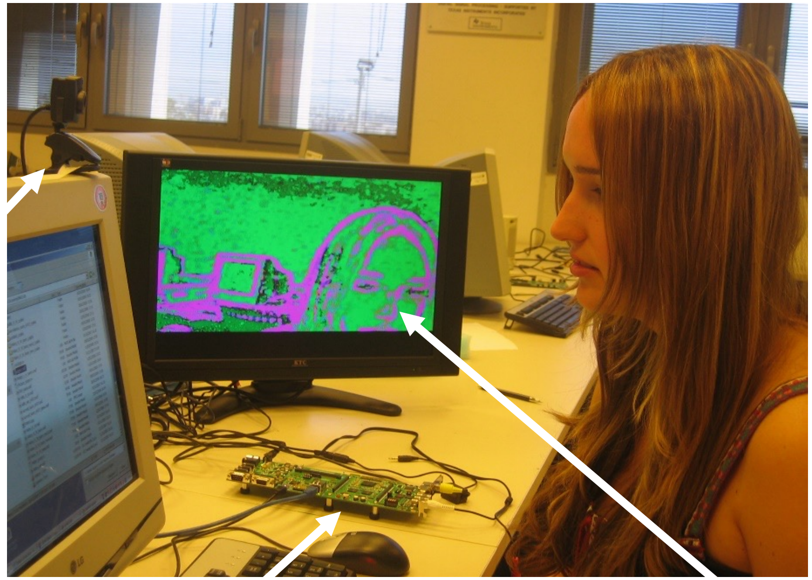
Edge Detection on Stills Images



Edge Detection Using the DSK6416



Edge Detection on Video



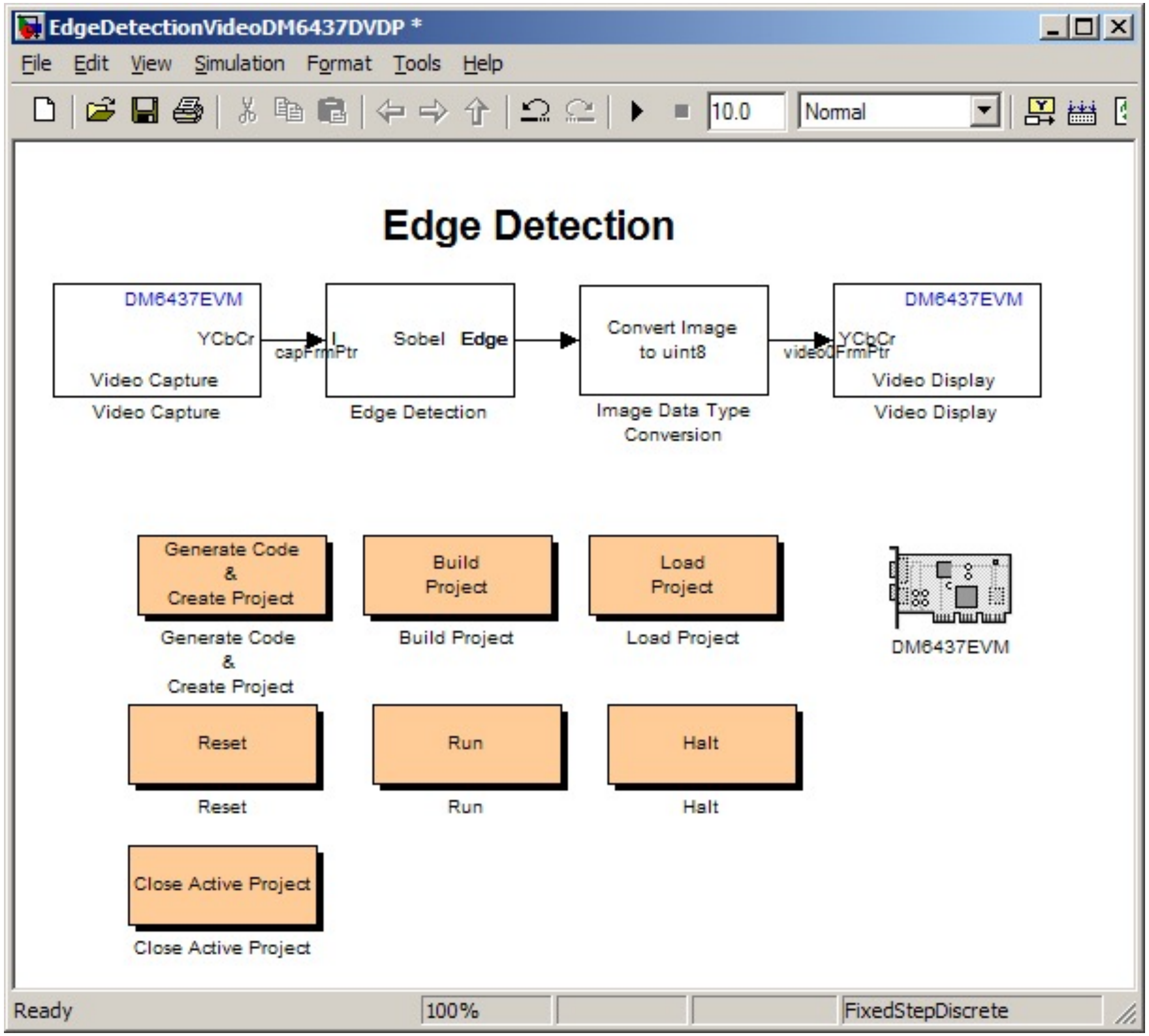
Camera

Edge Detection

DM6437 DVDP

Video Screen

Edge Detection Real Time Model for the DM6437 DVDP



Session Summary

- The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width *n*, at a particular orientation θ
- An edge is a set of connected pixels that lie on the boundary between two regions
- Magnitude of the gradient at a point (x, y) is approximated by taking the RMS of the directional differences
- Sobel edge detection uses 'first order gradients' of intensity gradient to detect edges within an image
- The Canny edge detection algorithm is known to many as the optimal edge detector
- Second order intensity gradients can be estimated using spatial convolution kernels, which are known as Laplace operators