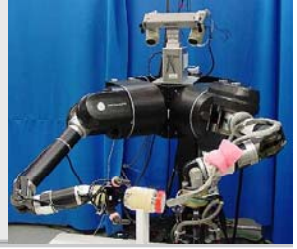
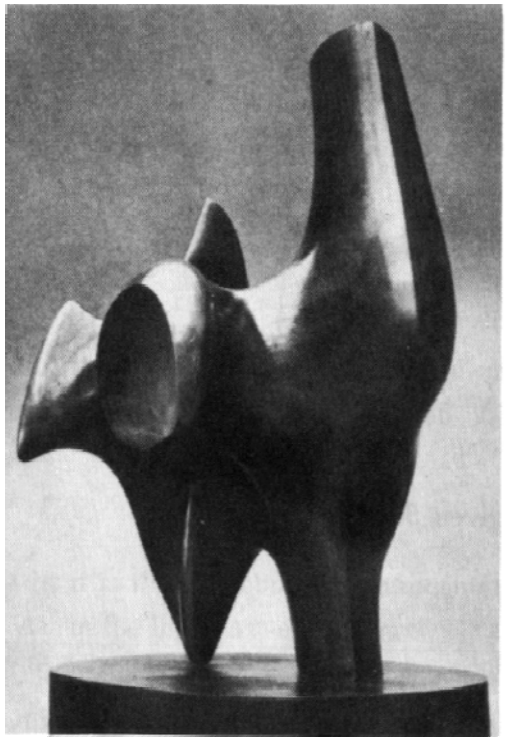
Einführung in Visual Computing

Unit 13: Local Operations – Edge Filtering



http://www.caa.tuwien.ac.at/cvl/teaching/sommersemester/evc

- Content:
  - Sharpening Filters
  - Edge Detection
  - First Derivative Filters
  - Second Derivative Filters
  - Canny Edge Detector

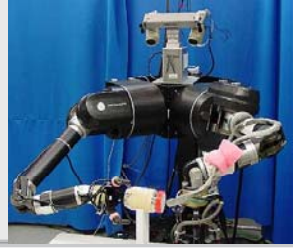


TU WIEN

CVL Computer Vision Lab

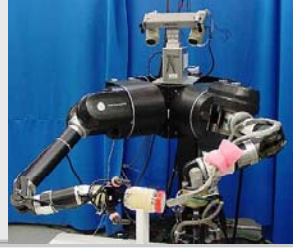
Sharpening Filters

Sharpening Spatial Filters

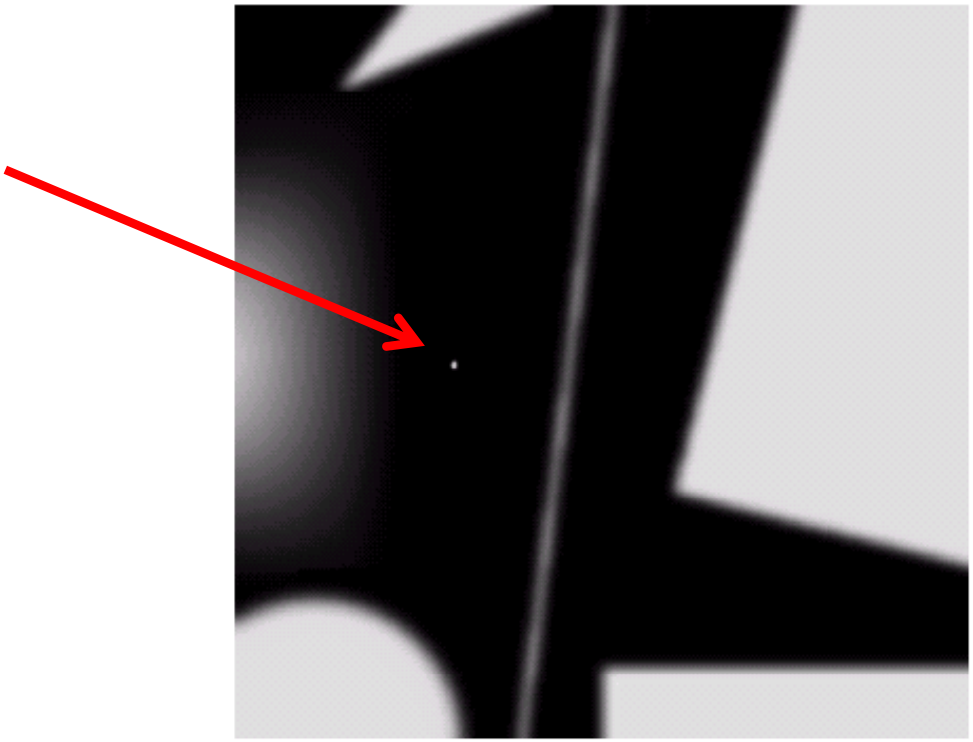


- Previously we have looked at smoothing filters which remove fine detail
- Sharpening spatial filters seek to highlight fine detail
  - Remove blurring from images
  - Highlight edges
- Sharpening filters are based on **spatial differentiation**

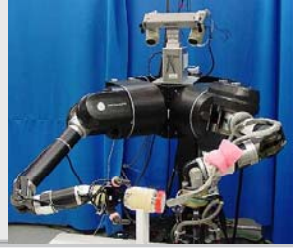
Spatial Differentiation

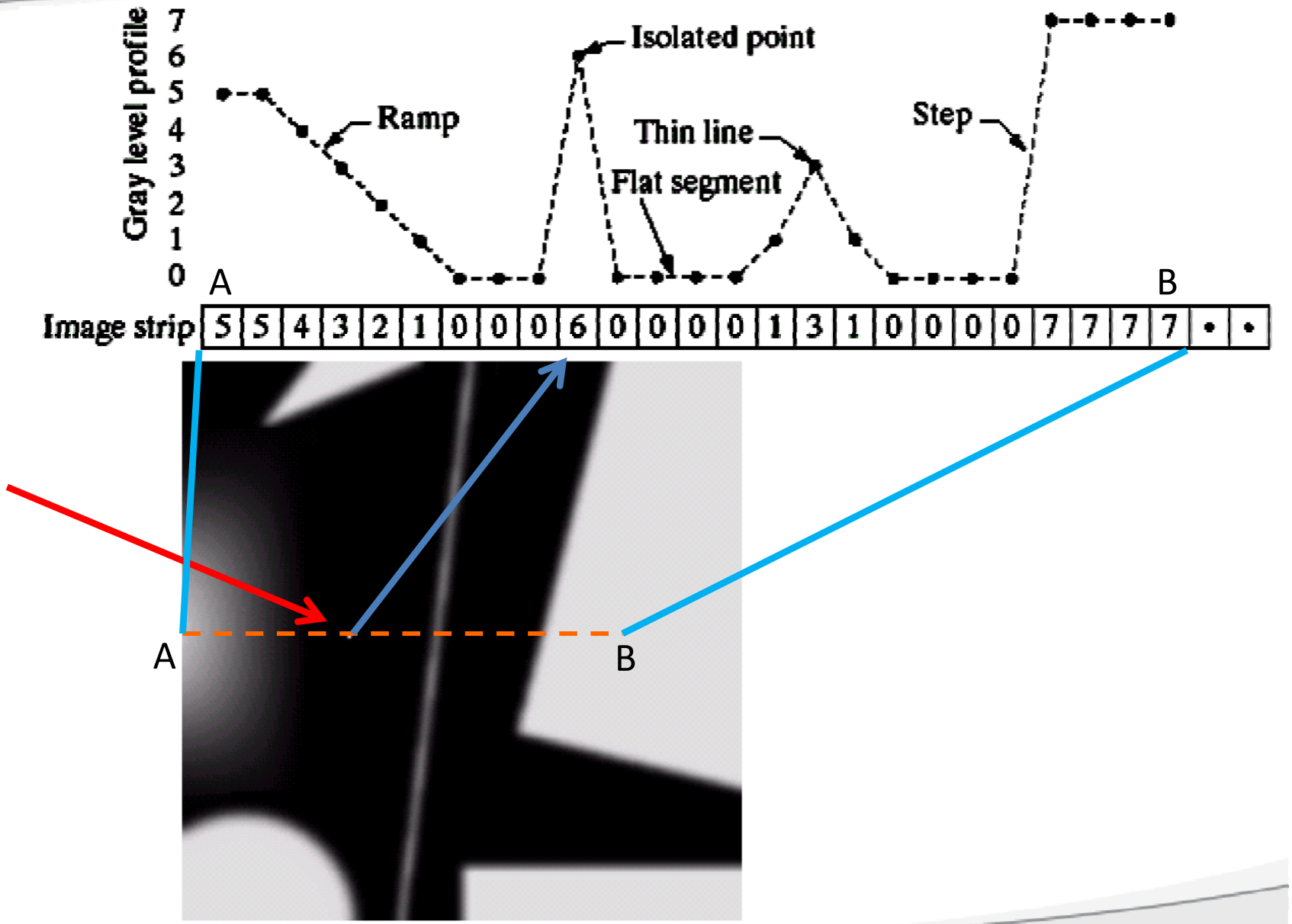


- Differentiation measures the rate of change of a function
- Let's consider a simple 1 dimensional example:

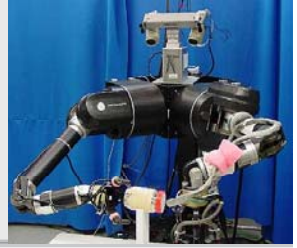


Spatial Differentiation





1st Derivative

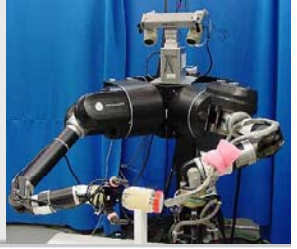


- The formula for the 1st derivative of a discrete function is as follows:

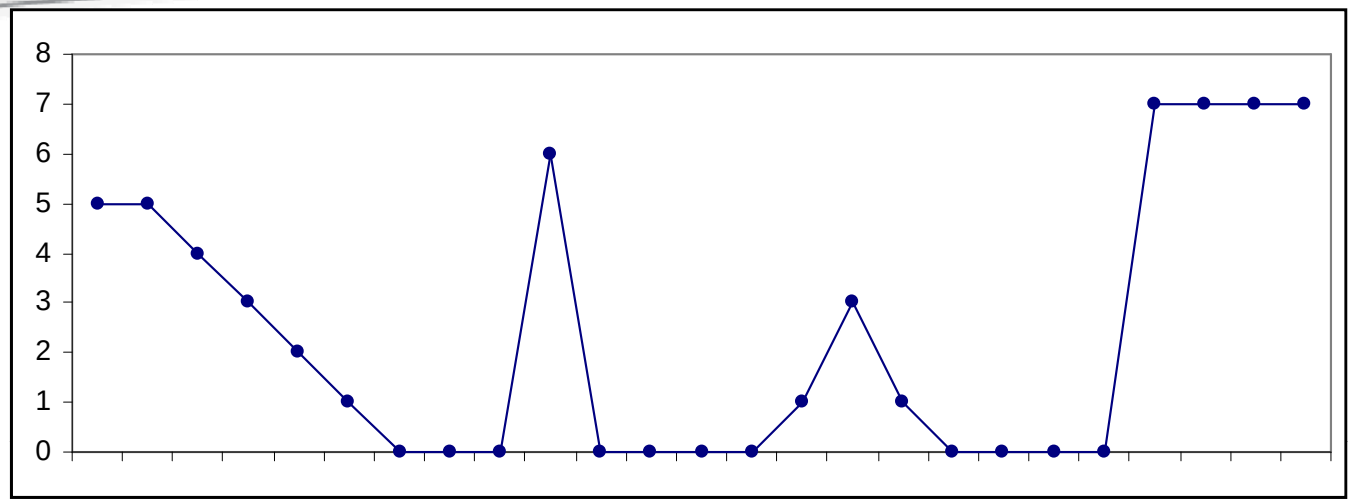
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- It's just the difference between subsequent values and measures the rate of change of the function

1st Derivative (cont…)



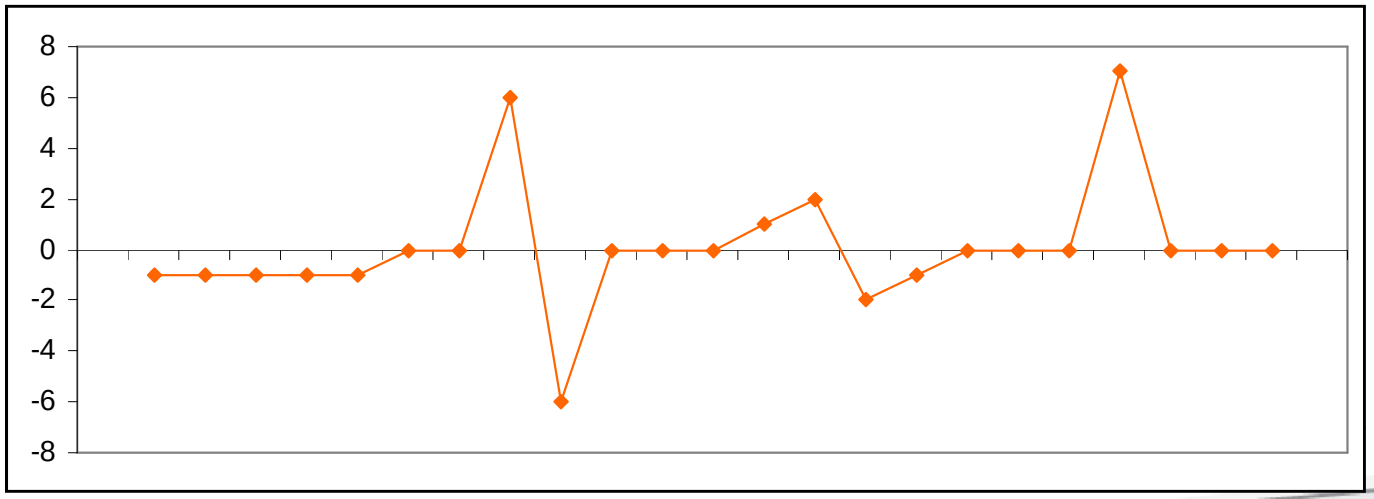
f(x)



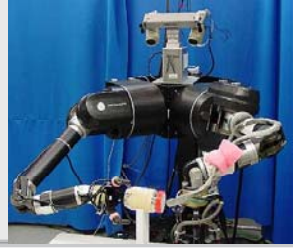
5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	**0**	0	7	0	0	0	

f’(x)



2nd Derivative

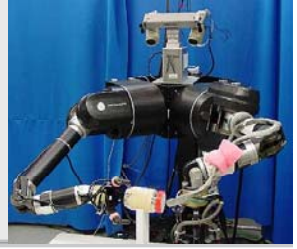


- The formula for the 2nd derivative of a function is as follows:

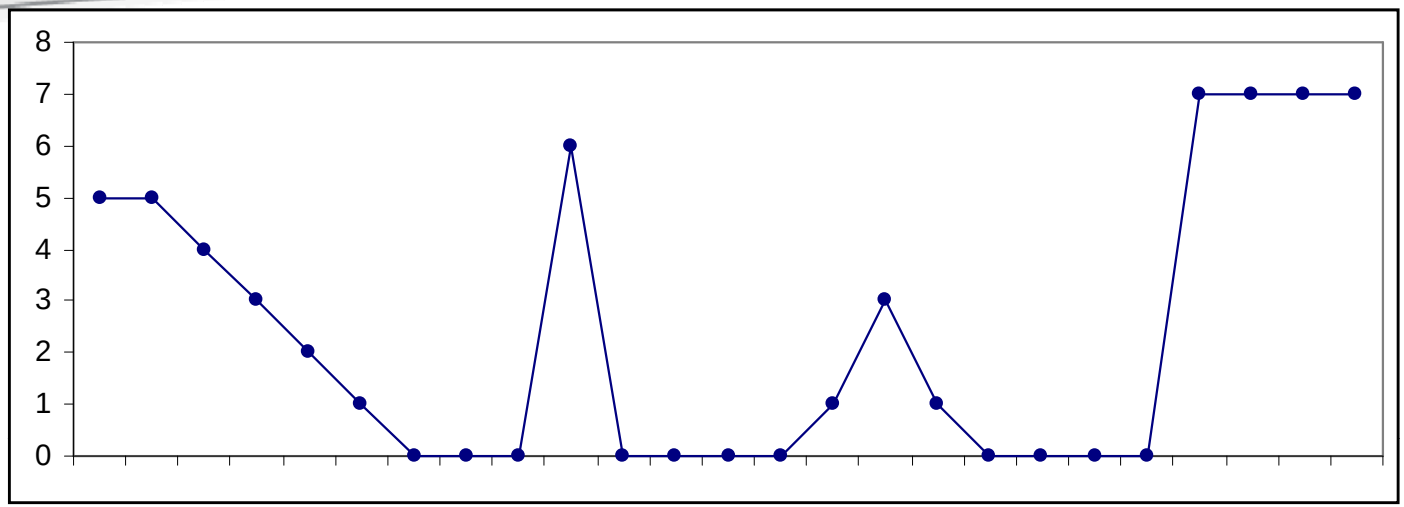
$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

- Simply takes into account the values both before and after the current value

2nd Derivative (cont…)

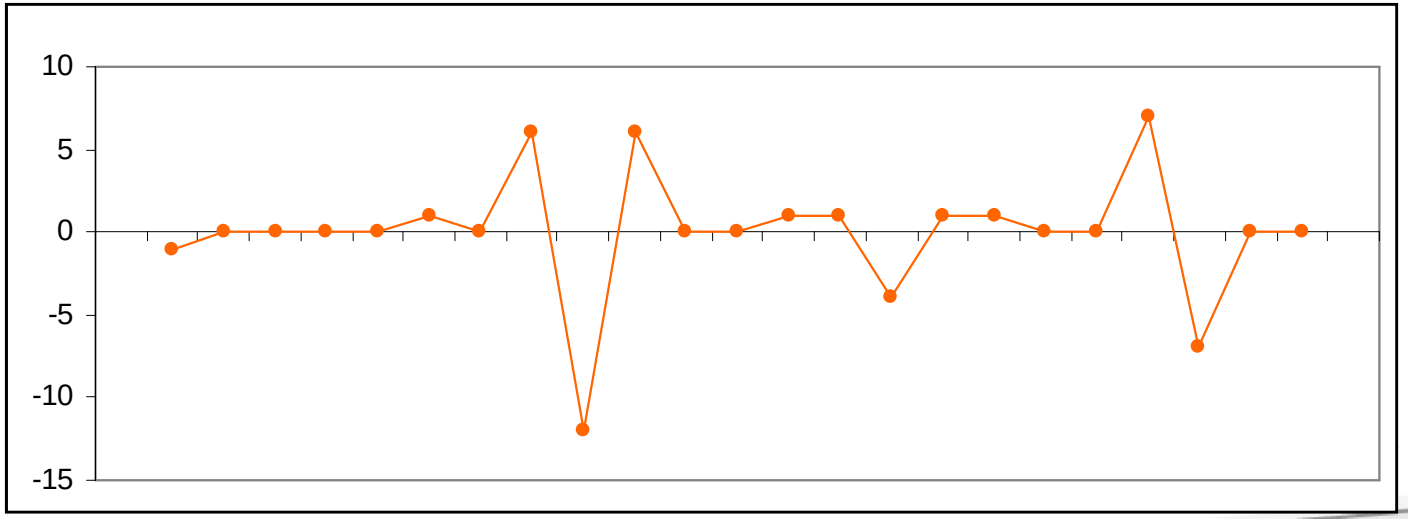


f(x)

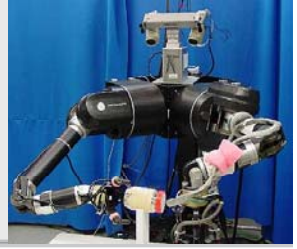


5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	

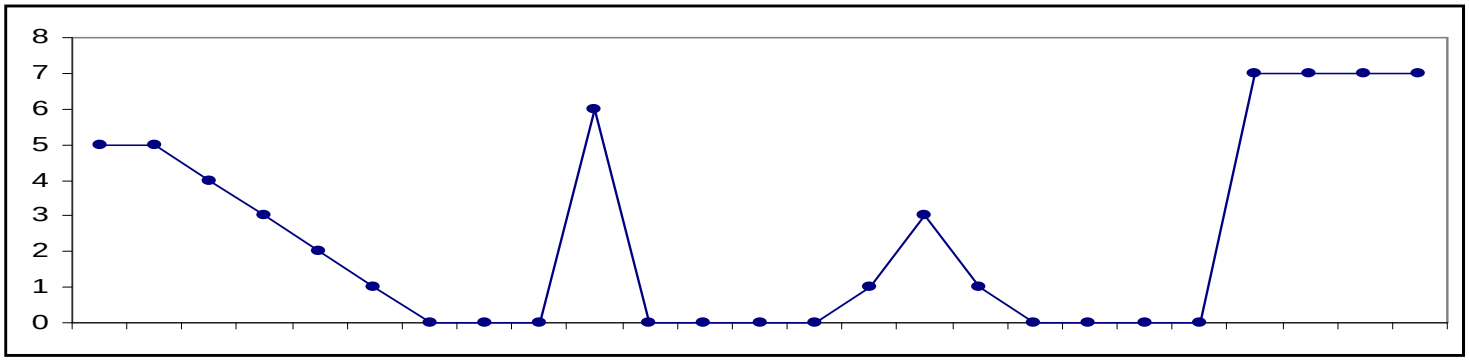
f’’(x)



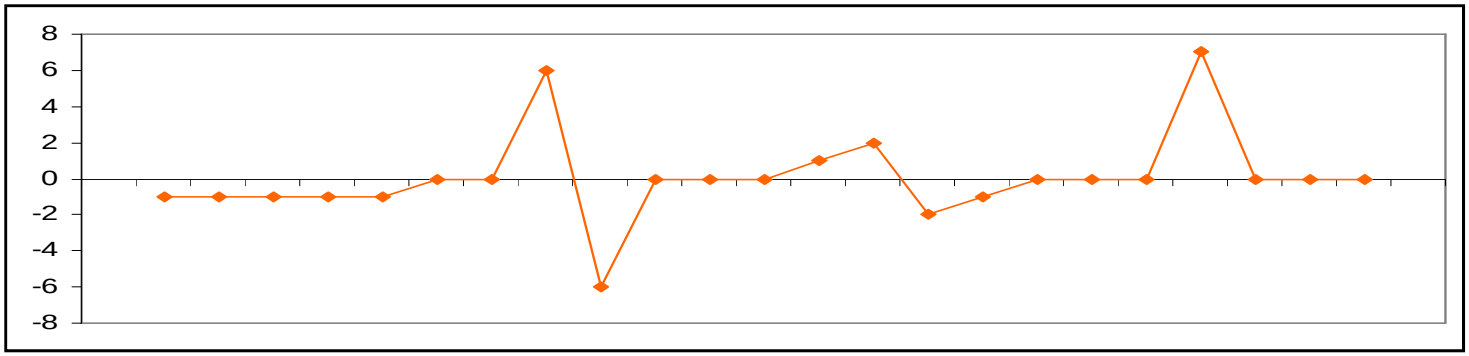
1st and 2nd Derivative



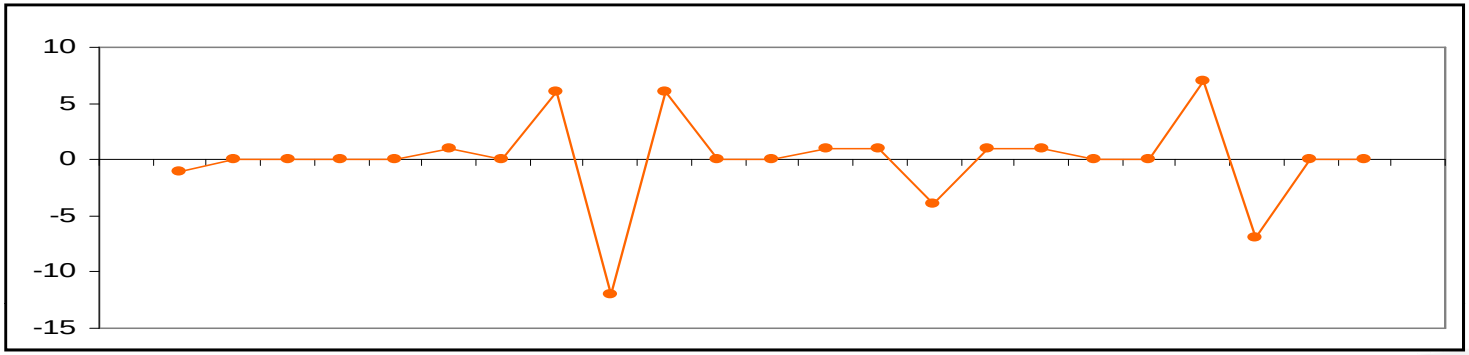
f(x)



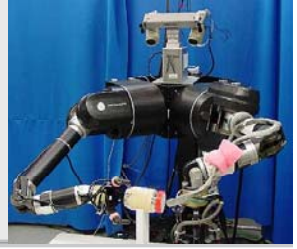
f’(x)



f’’(x)



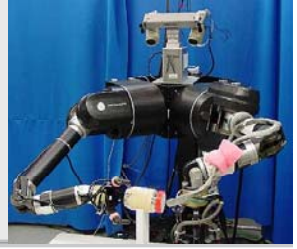
Using Second Derivatives For Image Enhancement



- The 2nd derivative is more useful for image enhancement than the 1st derivative
  - Stronger response to fine detail
  - Simpler implementation
  - We will come back to the 1st order derivative later on

- The first sharpening filter we will look at is the Laplacian
  - Isotropic
  - One of the simplest sharpening filters

The Laplacian



- The Laplacian is defined as follows:

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

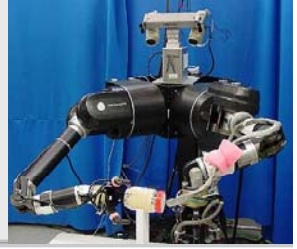
- where the partial 1st order derivative in the x direction is defined as follows:

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

- and in the y direction as follows:

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

The Laplacian (cont…)



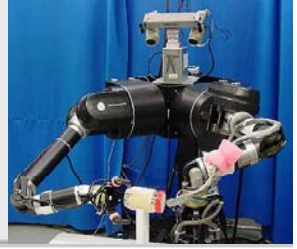
- So, the Laplacian can be given as follows:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

- We can easily build a filter based on this

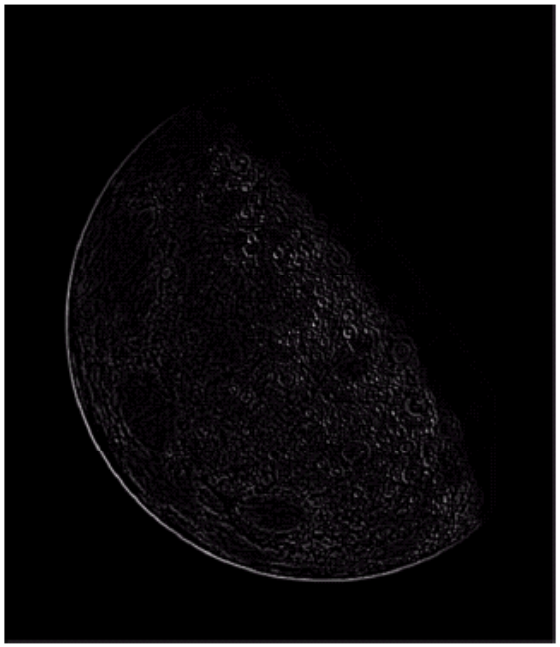
0	1	0
1	-4	1
0	1	0

The Laplacian (cont…)



- Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities

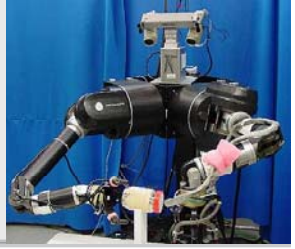
Original Image



Laplacian Filtered Image

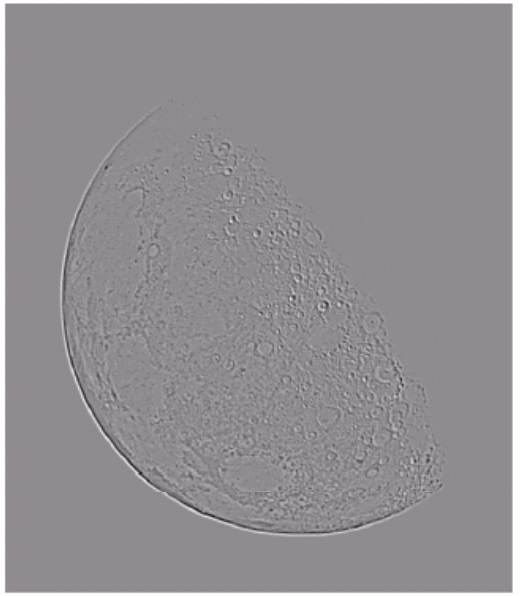
Laplacian Filtered Image Scaled for Display

But That Is Not Very Enhanced!



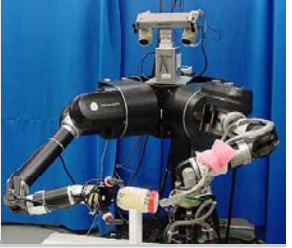
- The result of a Laplacian filtering is not an enhanced image
- We have to do more work in order to get our final image
- Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

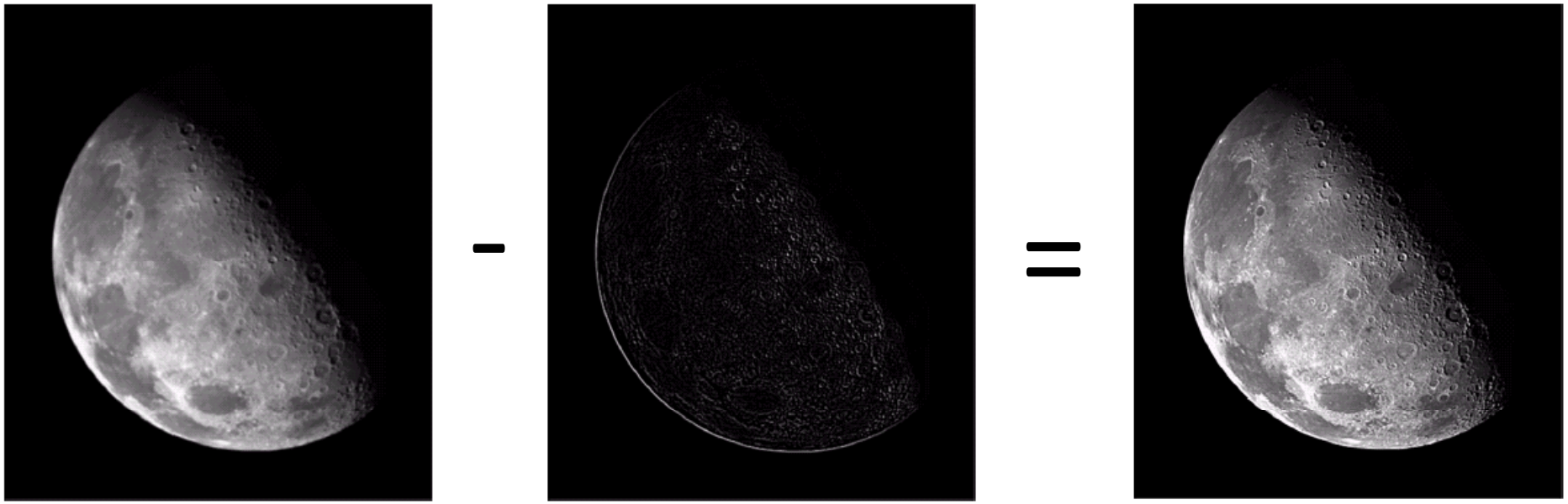
$$g(x,y) = f(x,y) - \nabla^2 f$$



Laplacian
Filtered Image
Scaled for Display

Laplacian Image Enhancement





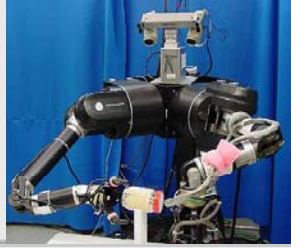
Original Image

Laplacian Filtered Image

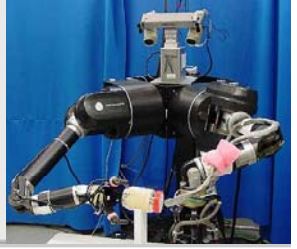
Sharpened Image

- In the final sharpened image edges and fine detail are much more obvious

Laplacian Image Enhancement



Simplified Image Enhancement



- The entire enhancement can be combined into a single filtering operation

$$g(x,y) = f(x,y) - \nabla^2 f$$

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

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

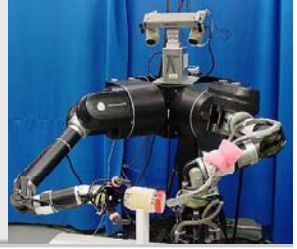
$$- 4f(x,y)]$$

$$= 5f(x,y) - f(x+1,y) - f(x-1,y)$$

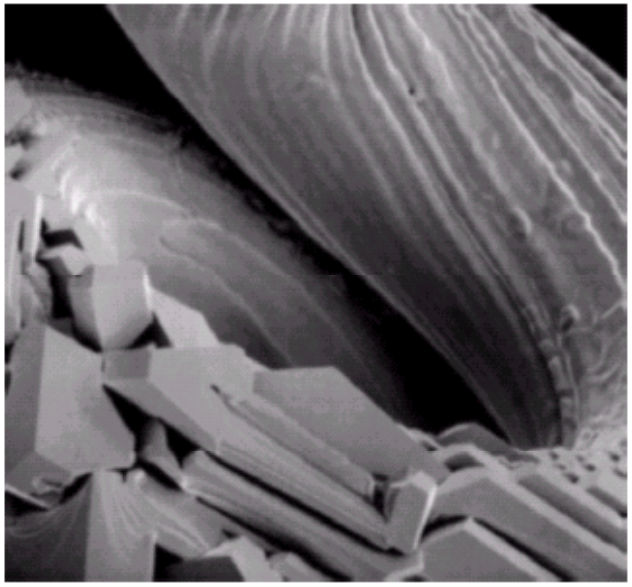
$$- f(x,y+1) - f(x,y-1)$$

0	-1	0
-1	5	-1
0	-1	0

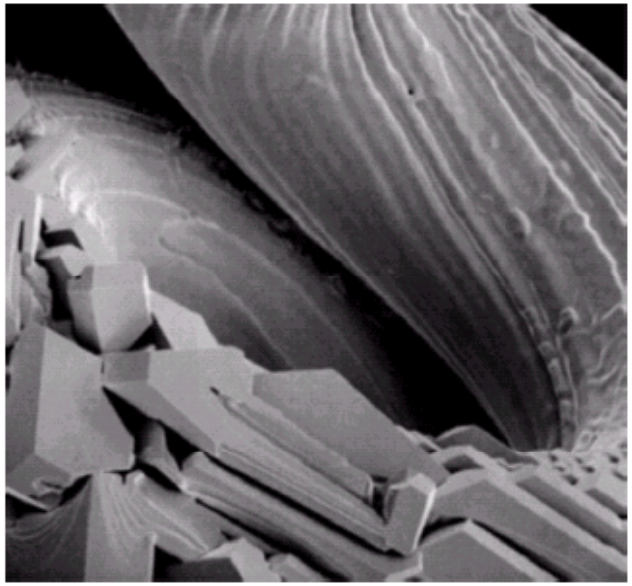
Simplified Image Enhancement (cont…)



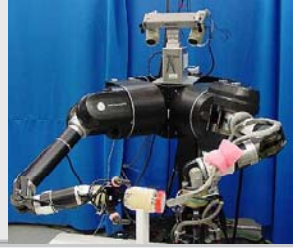
- This gives us a new filter which does the whole job for us in one step

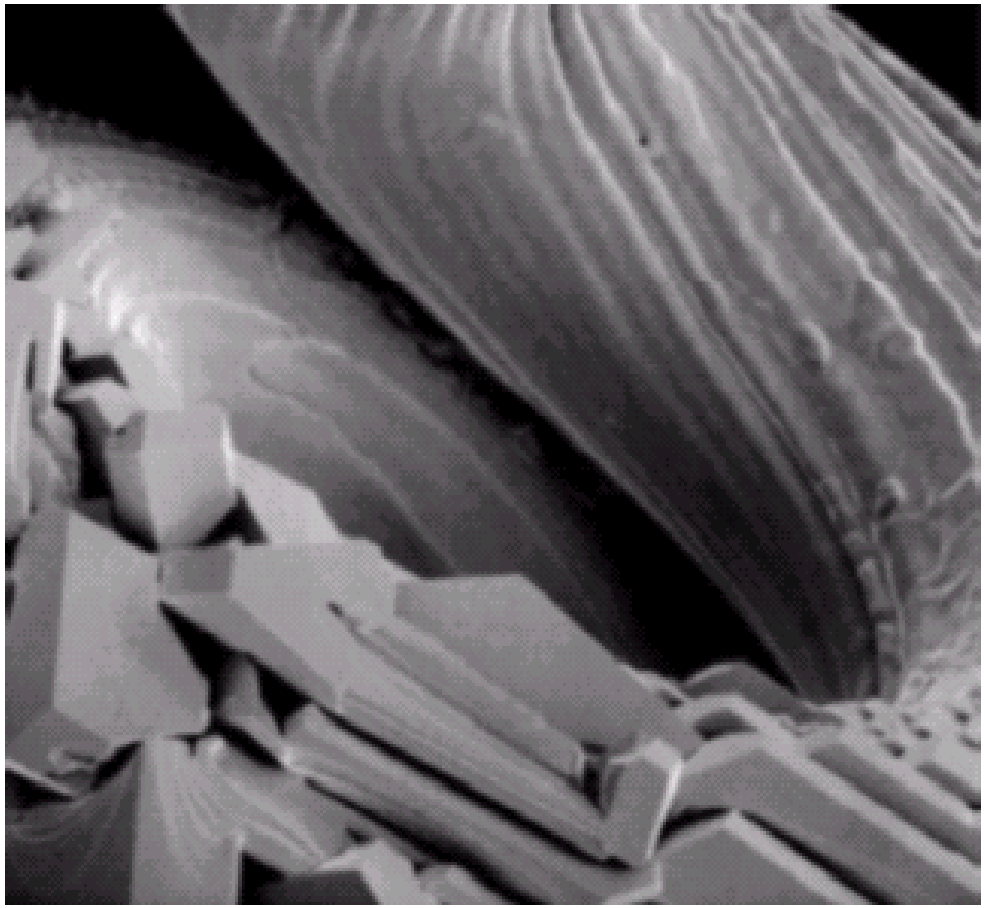


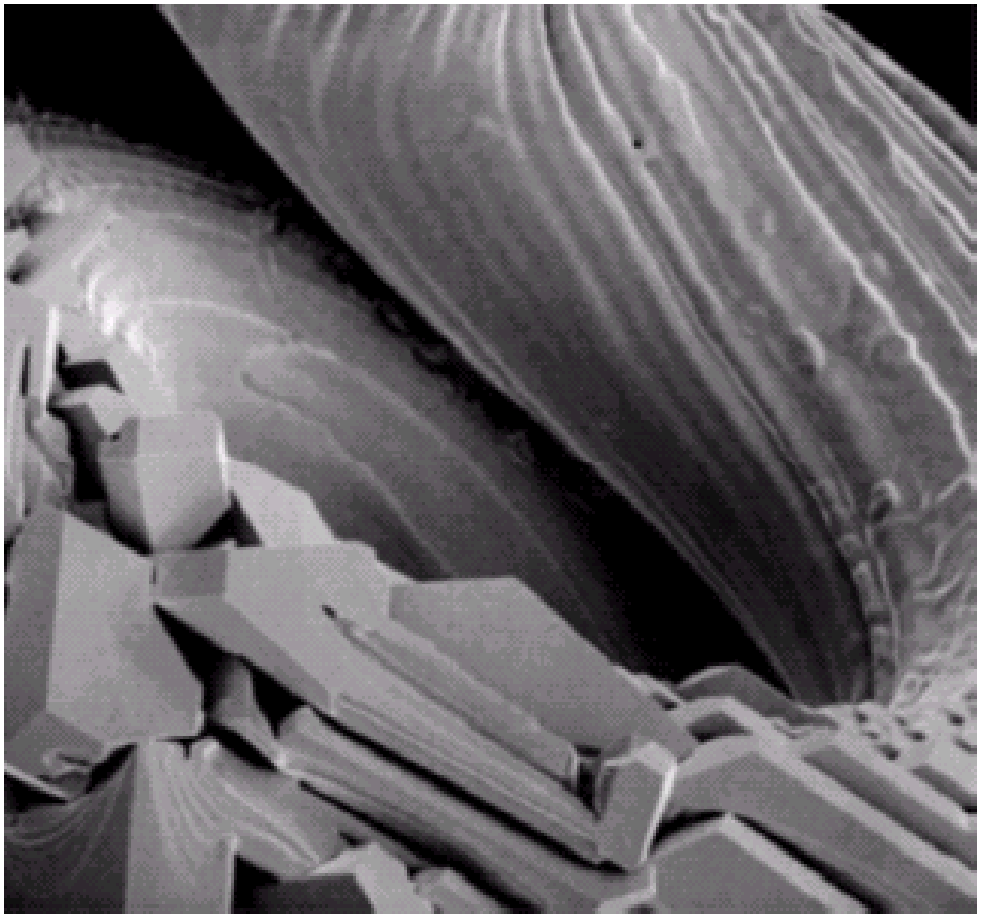
0	-1	0
-1	5	-1
0	-1	0



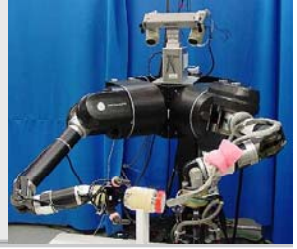
Simplified Image Enhancement (cont…)







Variants On The Simple Laplacian



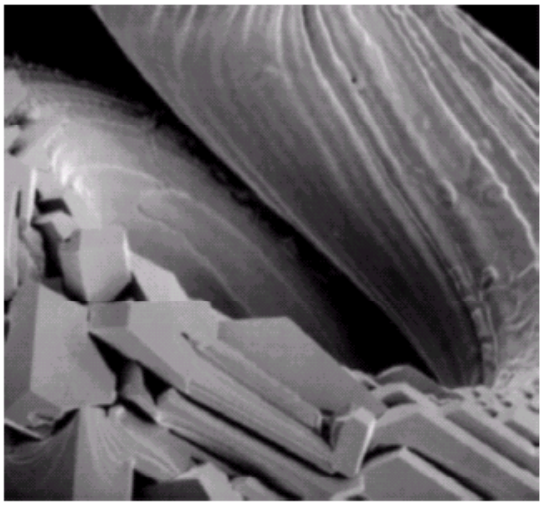
- There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

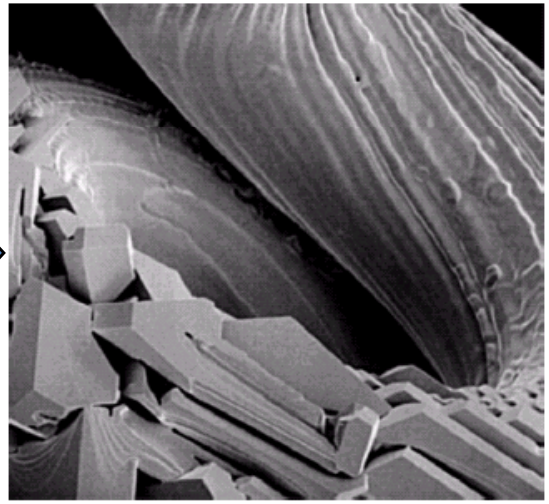
Simple Laplacian

1	1	1
1	-8	1
1	1	1

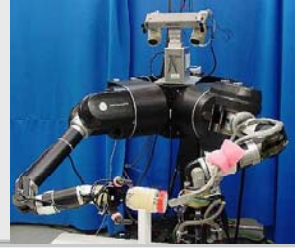
Variant of Laplacian

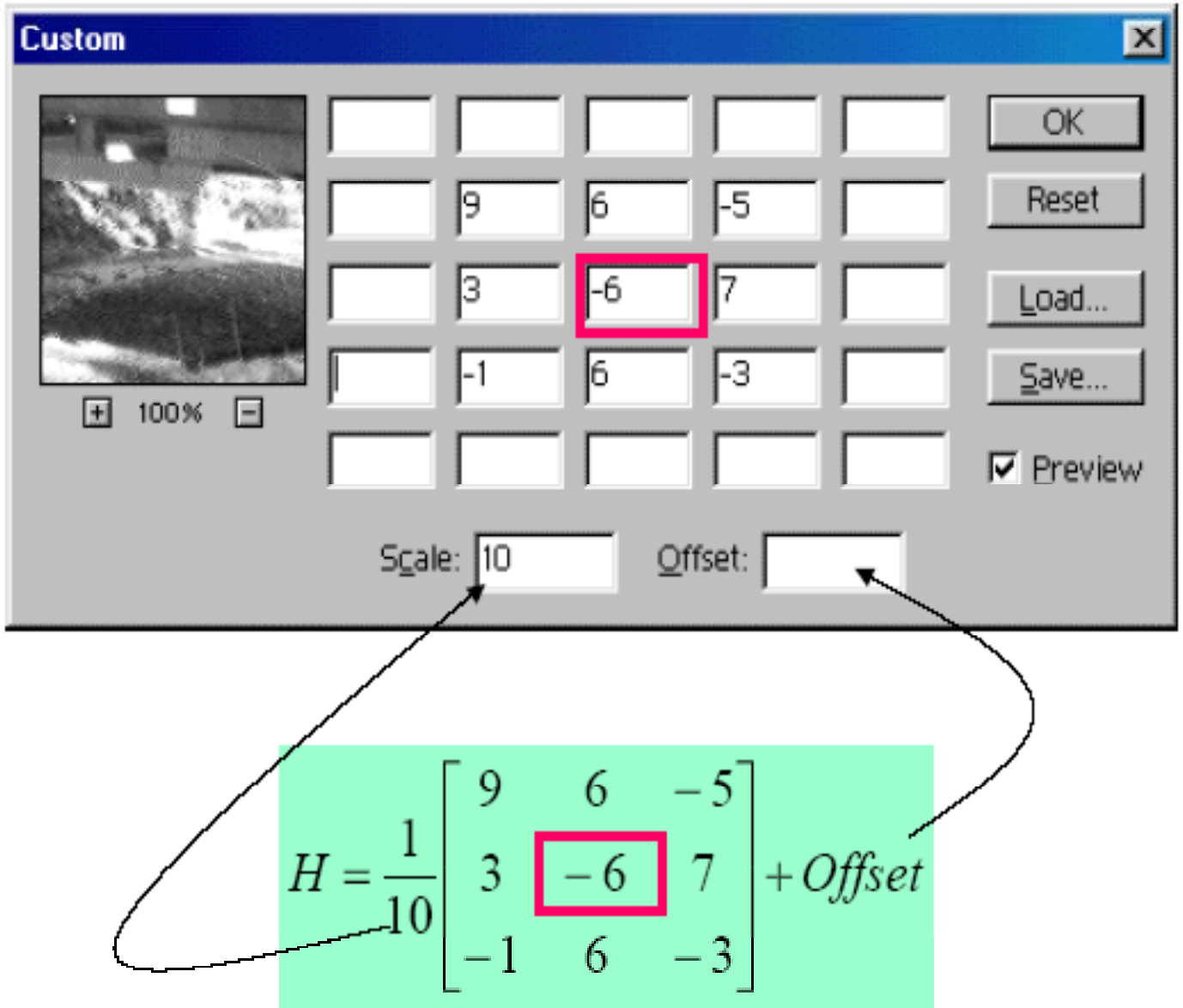


-1	-1	-1
-1	9	-1
-1	-1	-1



Photoshop: Other Filters - Custom Filter



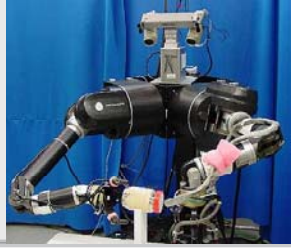


TU WIEN

CVL Computer Vision Lab

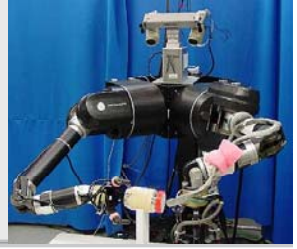
Edge Detection

Edge Detection



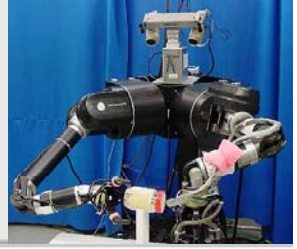
- Edges characterize boundaries of objects in image
- A fundamental problem in image processing
- Edges are areas with strong intensity contrasts
- A jump in intensity from one pixel to the next
- Edge detected image
  - Reduces significantly the amount of data,
  - Filters out useless information,
  - Preserves the important structural properties

Need of Edge Detection

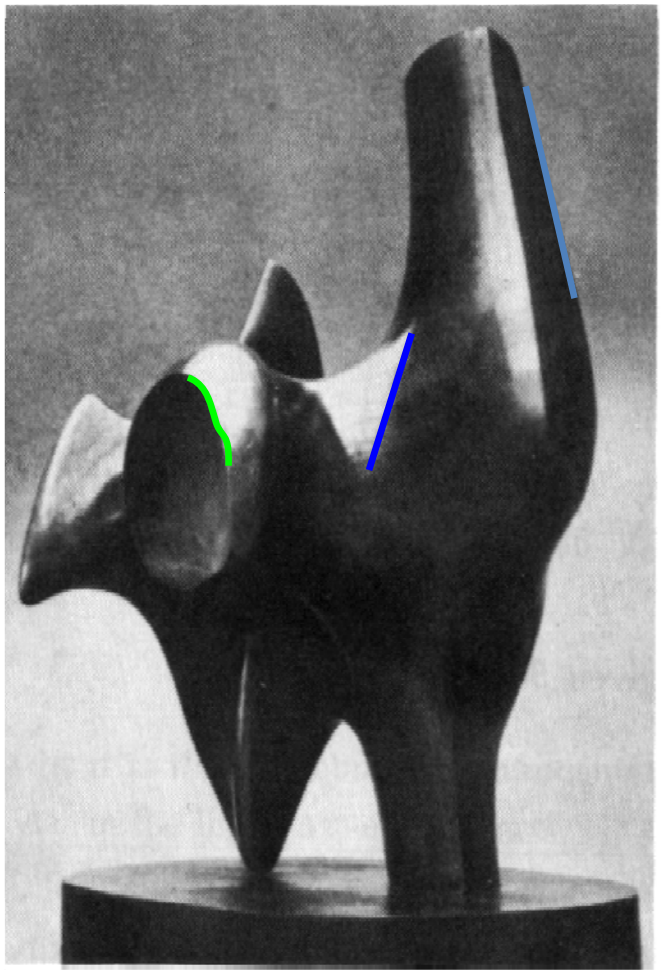


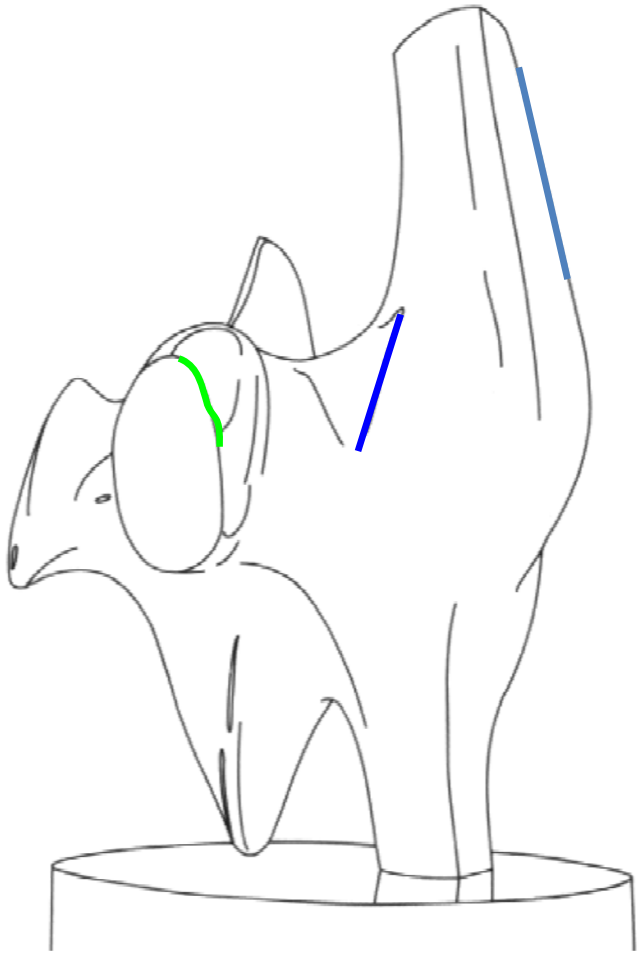
- Digital artists use it to create image outlines.
- The output of an edge detector can be added back to an original image to enhance the edges
- Edge detection is often the first step in image segmentation
- Edge detection is also used in image registration by alignment of two images that may have been acquired at separate times or from different sensors

Edge Detection



- Edges describe colloquially the edge of a surface or a significant change in orientation of the surface normals



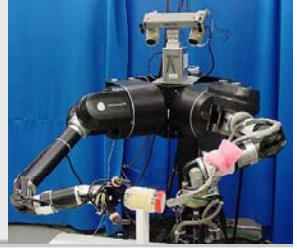


Normals

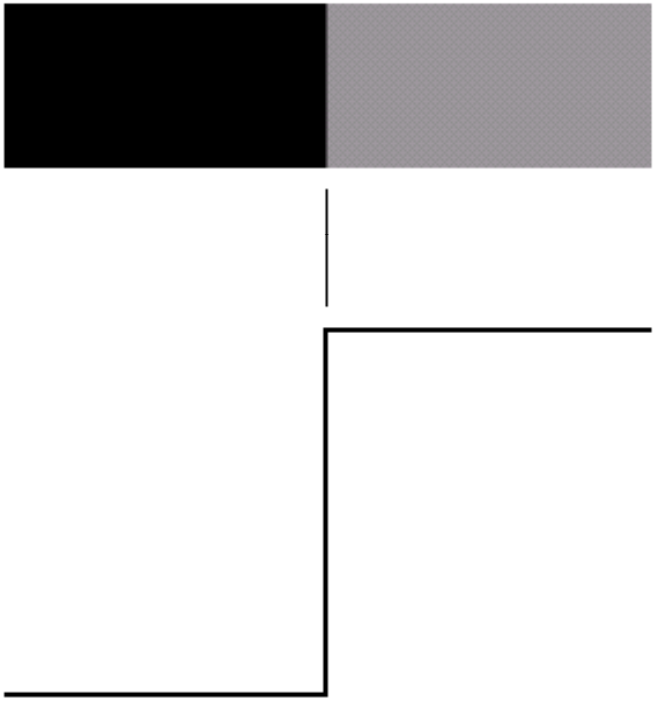
Texture

Depth

Ideal and Ramp Edges

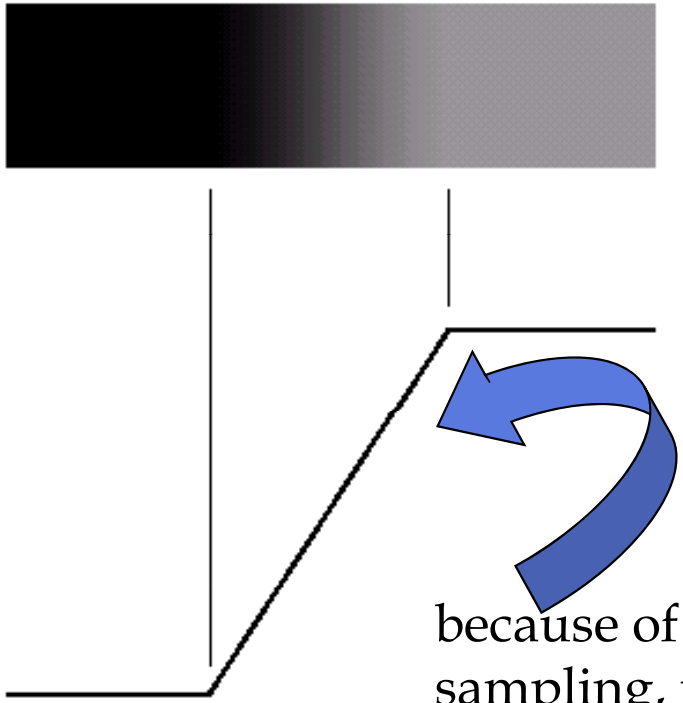


Model of an ideal digital edge



Gray-level profile
of a horizontal line
through the image

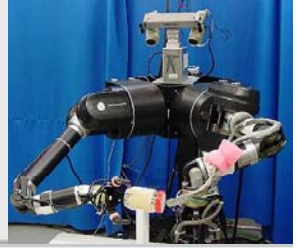
Model of a ramp digital edge



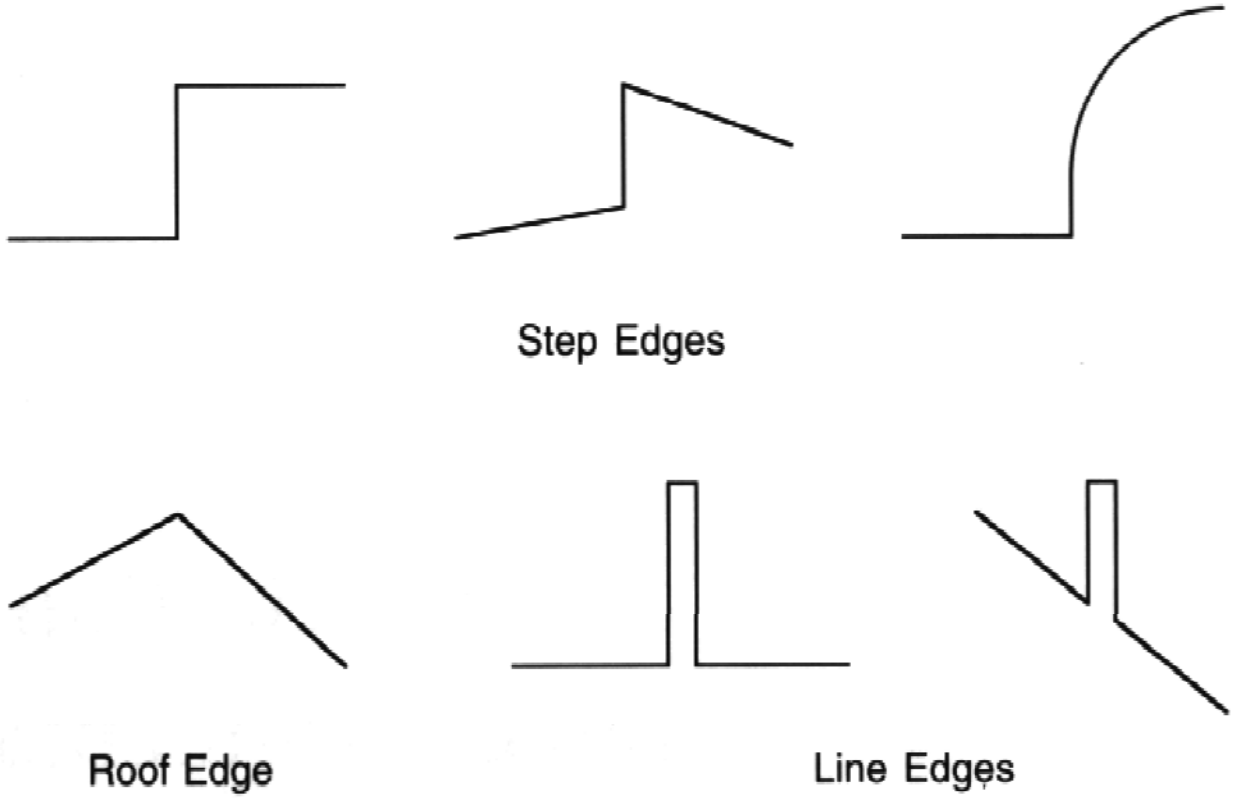
Gray-level profile
of a horizontal line
through the image

because of optics,
sampling, image
acquisition imperfection

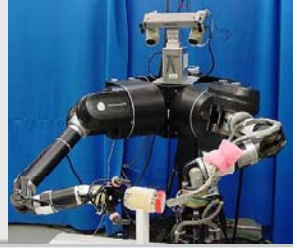
Edge Detection



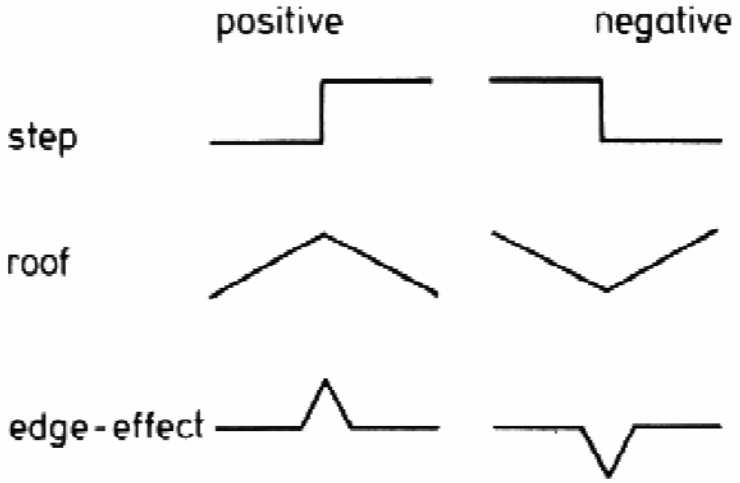
- Edges in image processing are boundary lines between two areas at which the gray value of pixels differ significantly



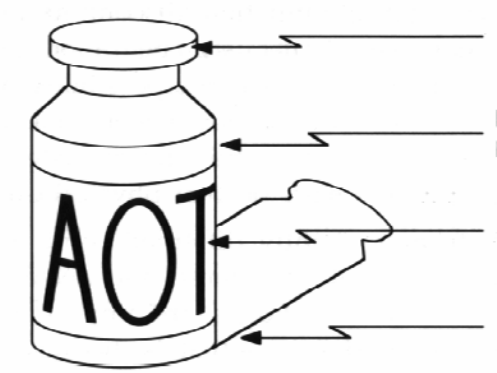
Types of Edges



- Step and "roof" edges
- Direction (positive or negative)
- Difference between gray values along an edge: **Contrast**

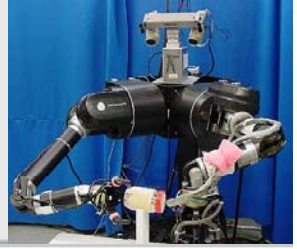


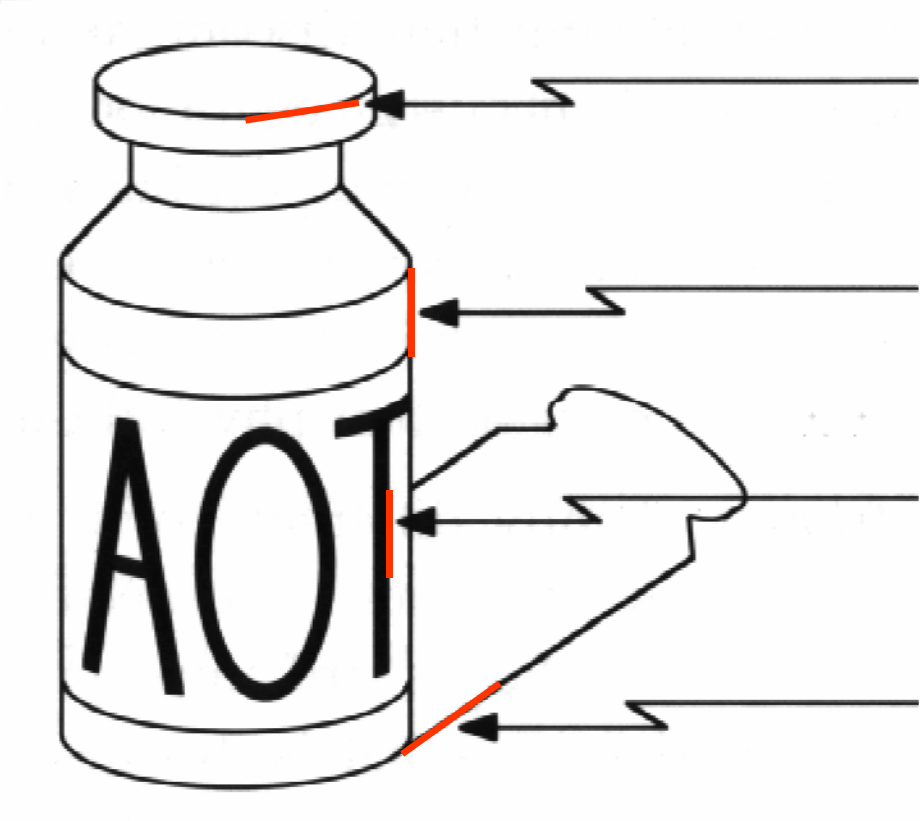
- **Discontinuities** of the brightness represent:



Discontinuities of surface **normal**
Discontinuities of the **depth**
Discontinuities of **texture**
Discontinuities of **illumination**

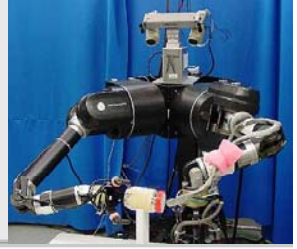
Types of Edges



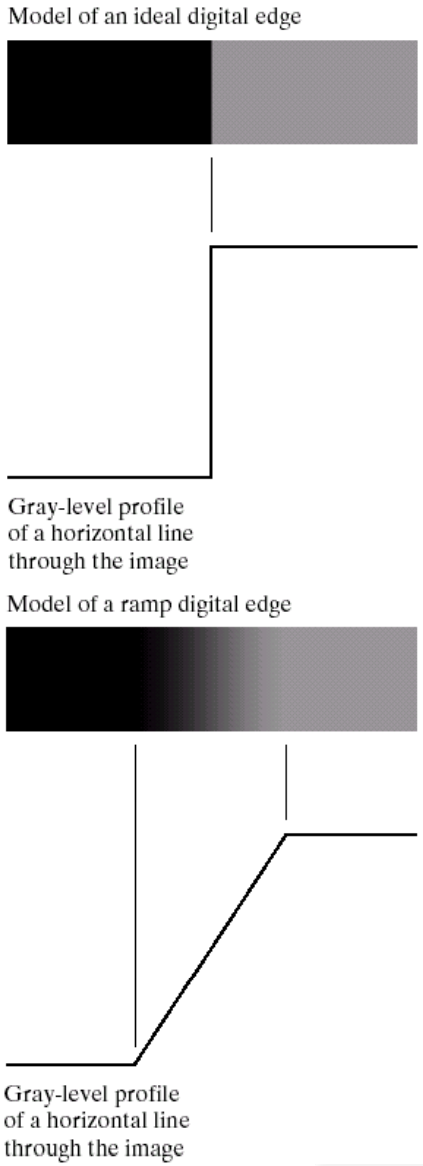


- Discontinuities of surface normals may also represent discontinuities of depth
- Edges contain non-redundant information!

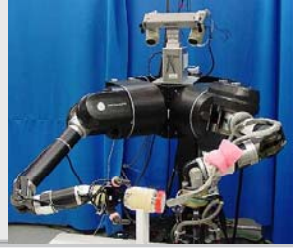
Thick Edge

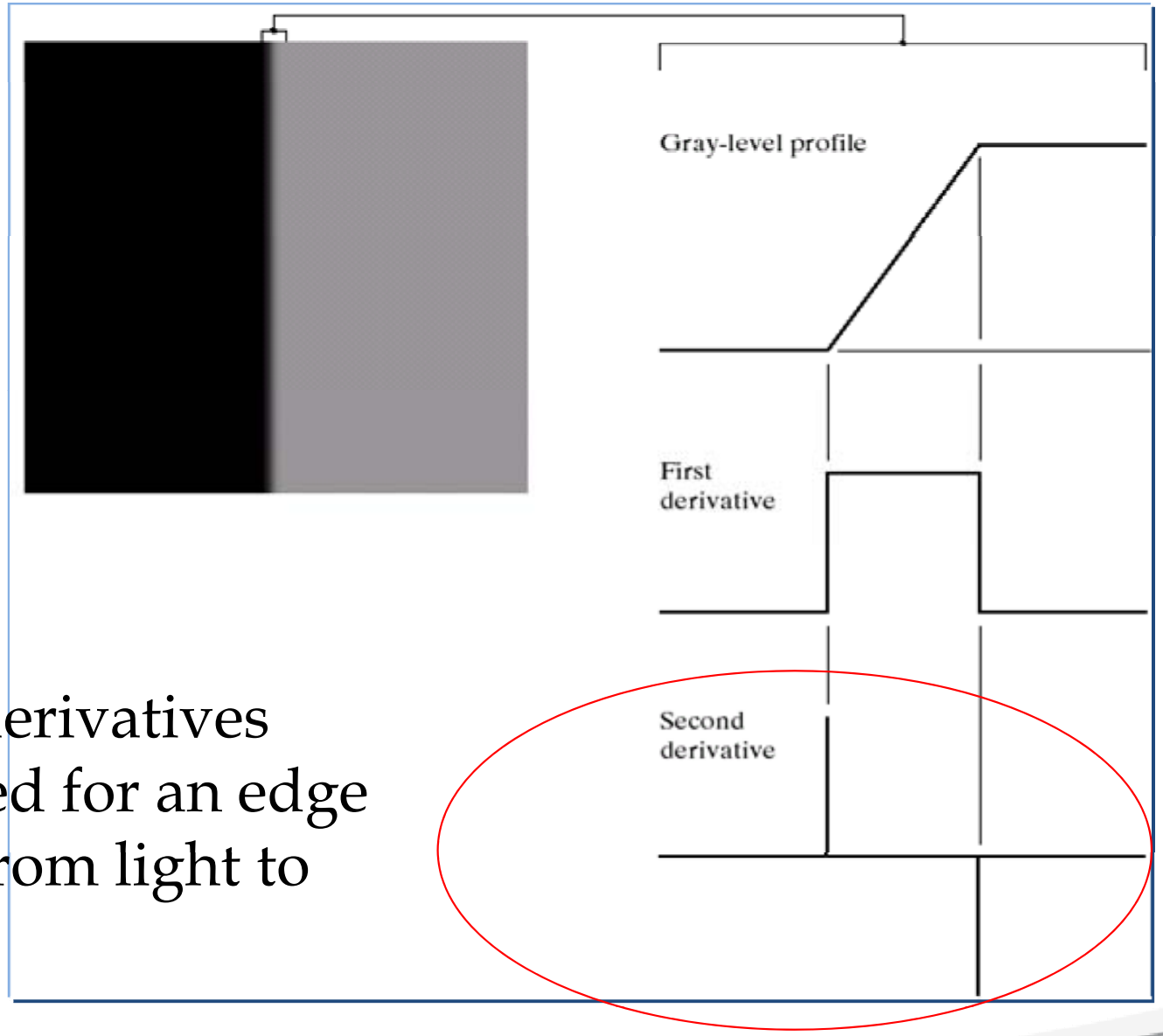


- The slope of the ramp is inversely proportional to the degree of blurring in the edge.
- We no longer have a thin (one pixel thick) path.
- Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected.
- The thickness of an edge is determined by the length of the ramp.
- The length is determined by the slope, which is in turn determined by the degree of blurring.
- Blurred edges tend to be thick and sharp edges tend to be thin



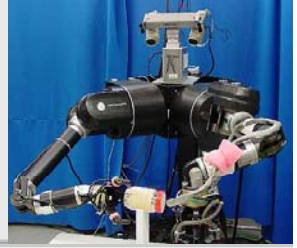
First and Second Derivatives





the signs of the derivatives would be reversed for an edge that transitions from light to dark

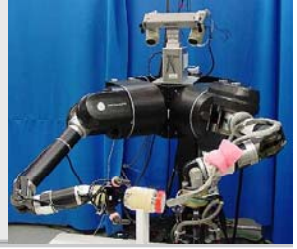
1st Derivative Filtering



- Implementing 1st derivative filters is difficult in practice
- For a function f(x, y) the gradient of f at coordinates (x, y) is given as the column vector:

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

1st Derivative Filtering (cont…)



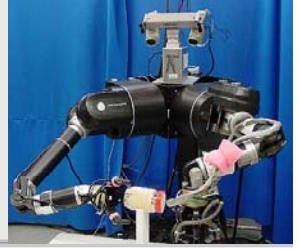
- The magnitude of this vector is given by:

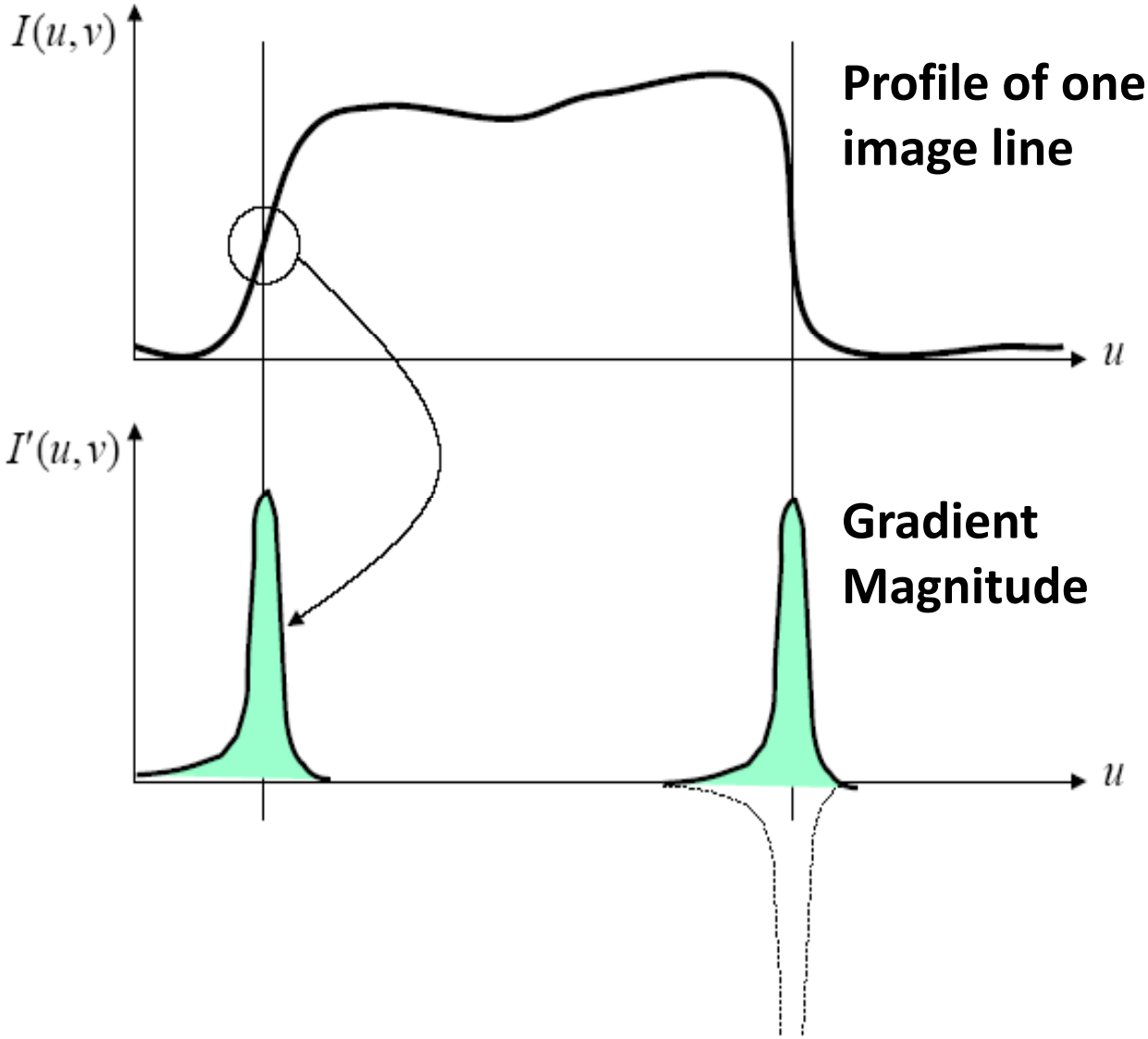
$$\nabla f = mag(\nabla \mathrm{f}) = \left[G_x^2 + G_y^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

- For practical reasons this can be simplified as:

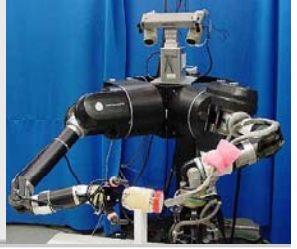
$$\nabla f \approx |G_x| + |G_y|$$

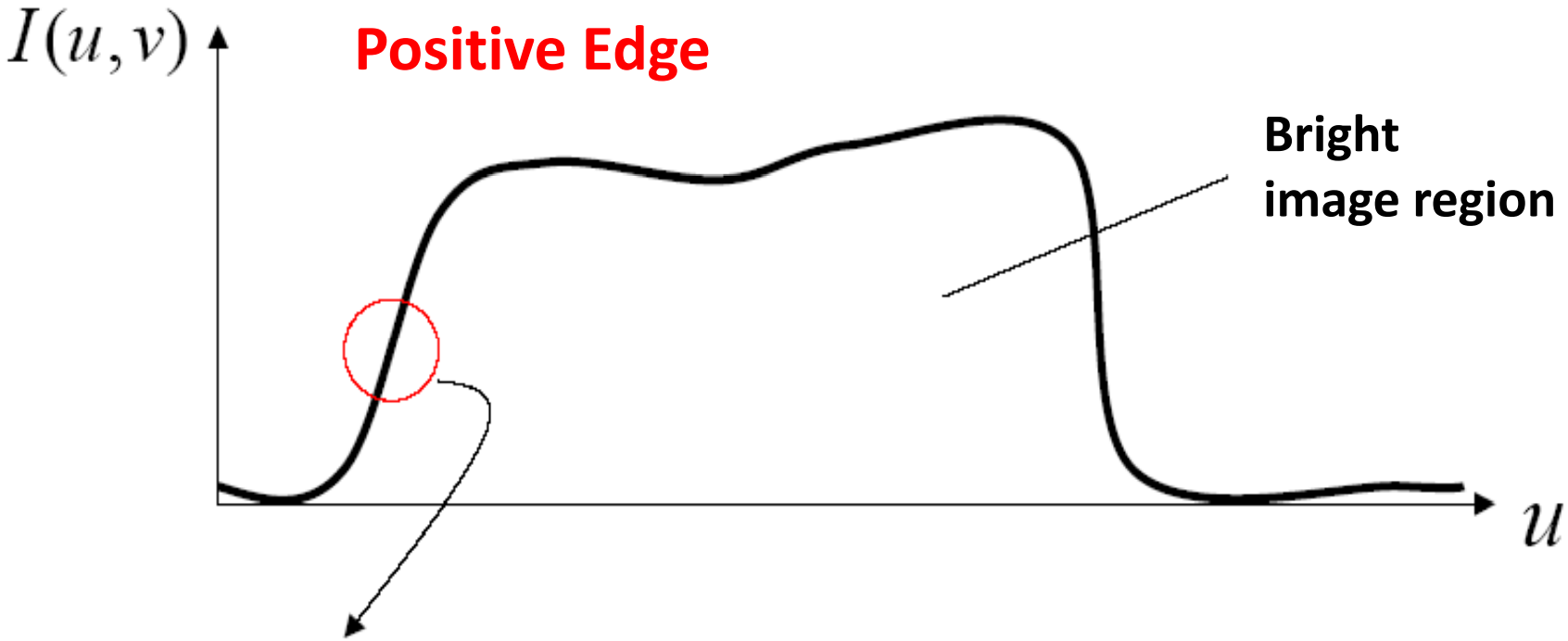
Edge-derivative





What happens at an edge?



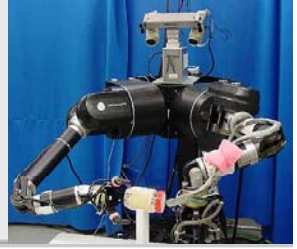


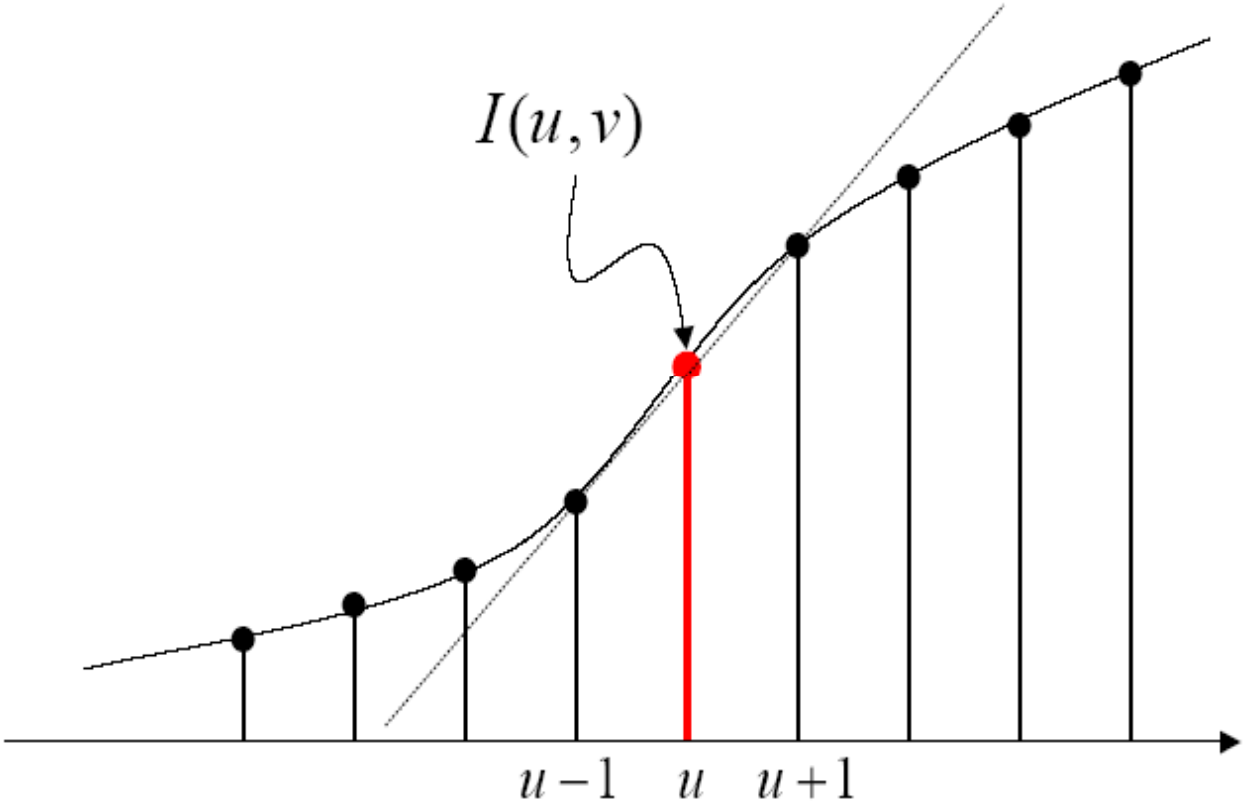
Gradient is high!

1st Derivative

$$\frac{\partial I(u,v)}{\partial u}$$

How to measure the 1st Derivative?

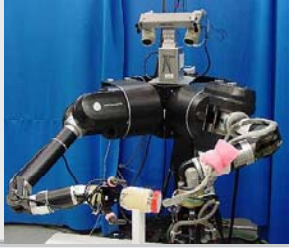


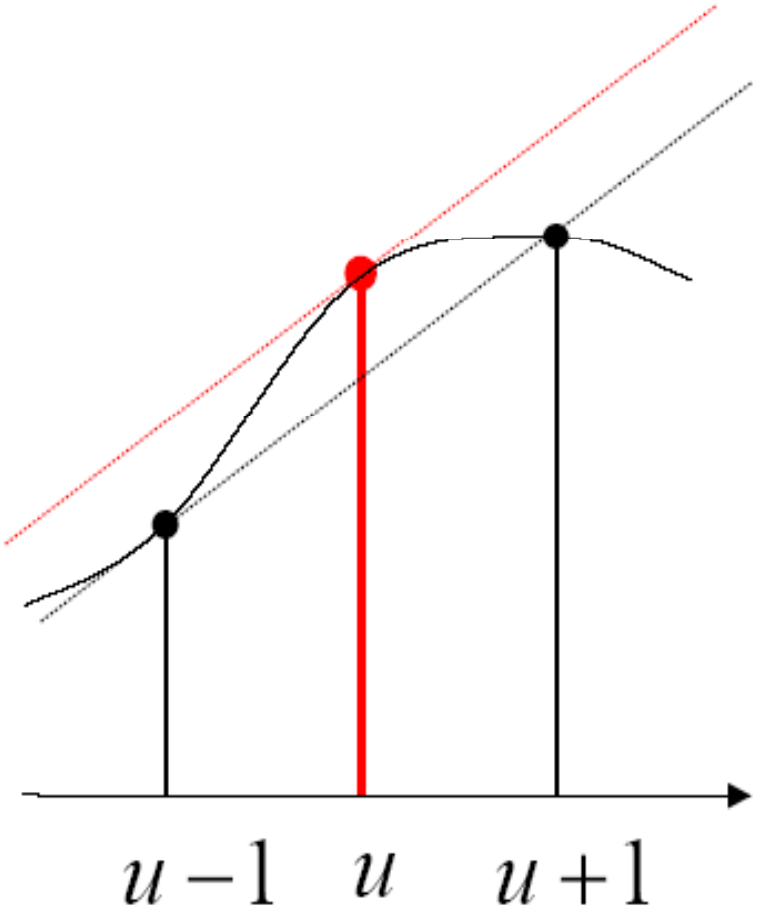


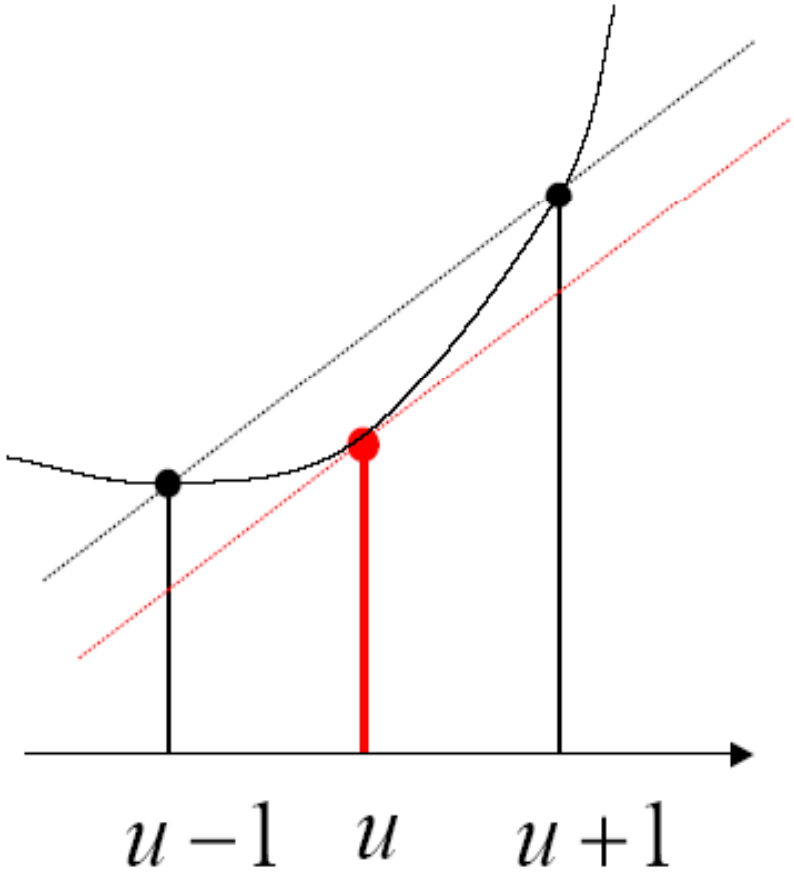
$$\frac{\partial I(u,v)}{\partial u} \approx I(u+1,v) - I(u-1,v)$$

for discrete signals only approximation possible

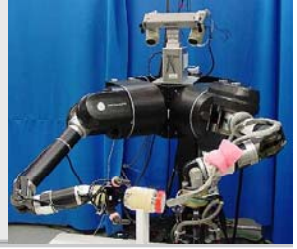
Approximation of 1st Derivative

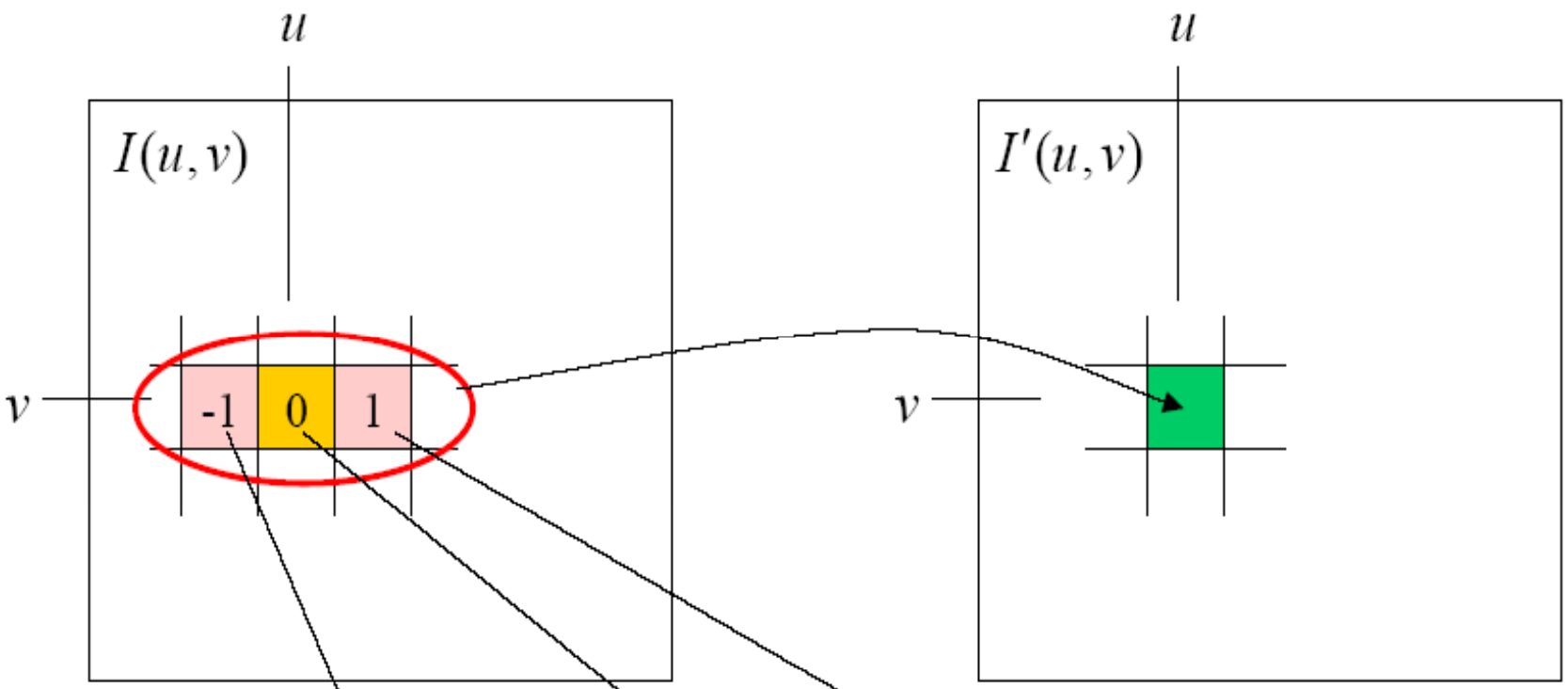






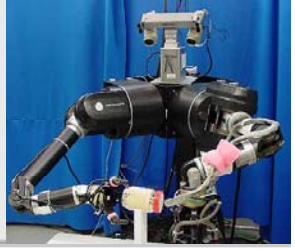
Simple horizontal Contour Filter



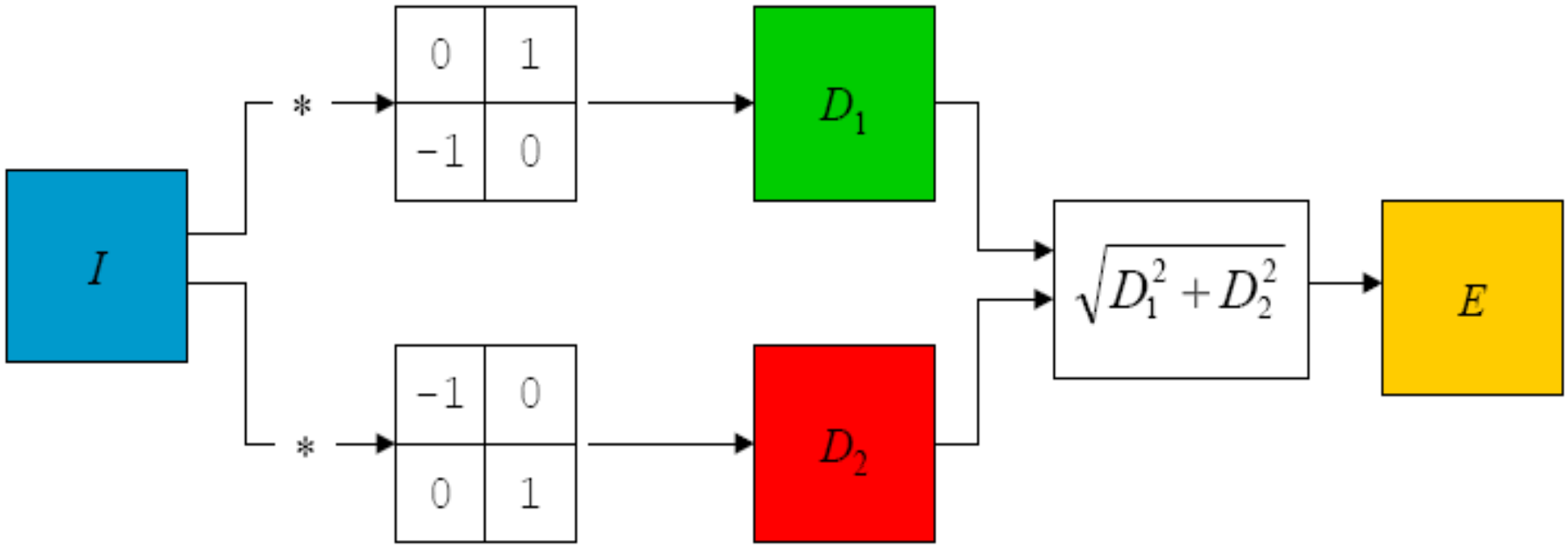


$$I'(u,v) = -1 \cdot I(u-1,v) + 0 \cdot I(u,v) + 1 \cdot I(u+1,v)$$

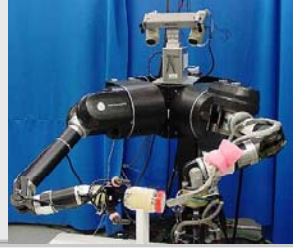
Goal: Find Edges independent of their Orientation

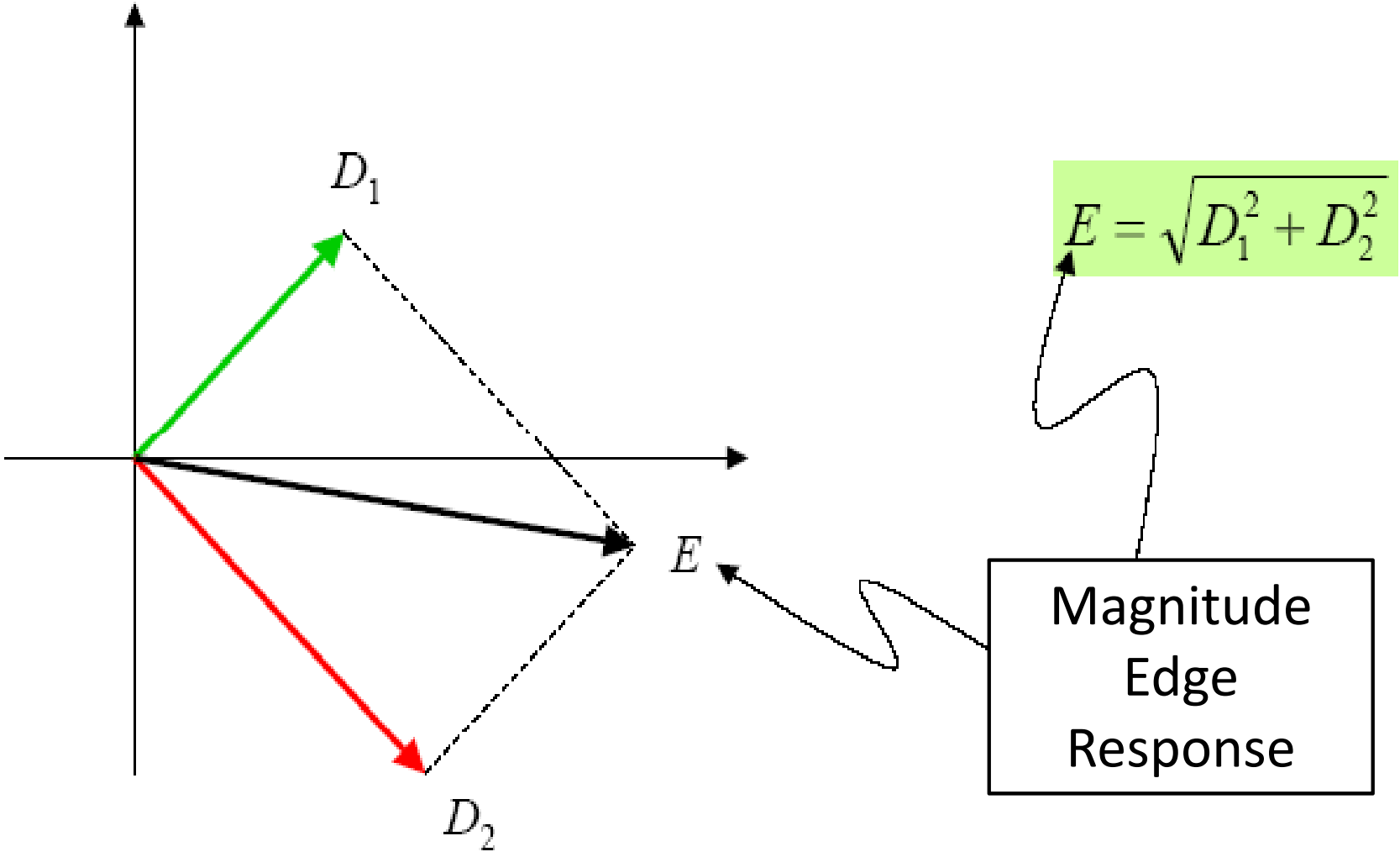


- Method by Roberts (1965)

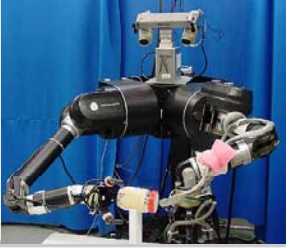


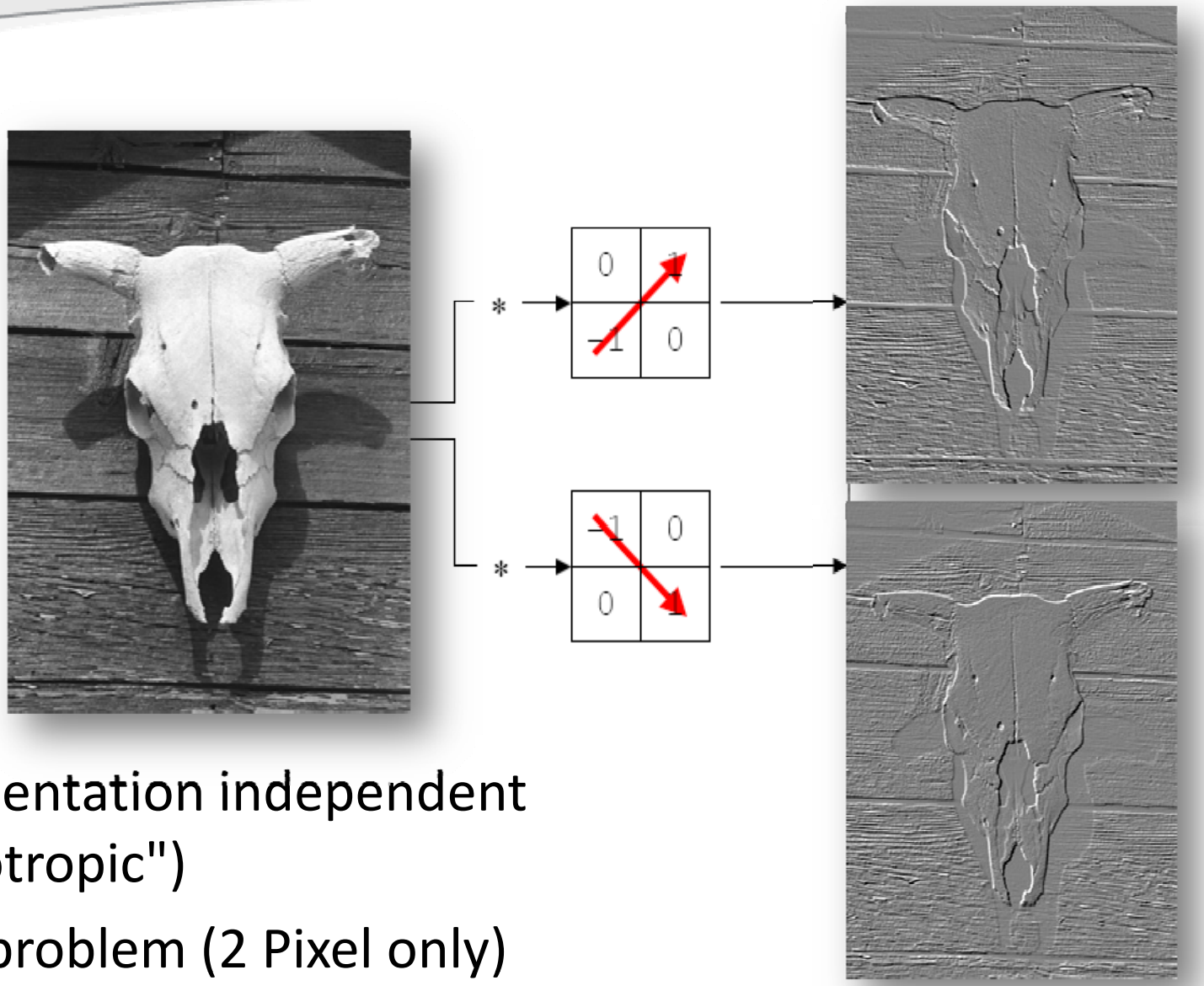
Roberts Operator





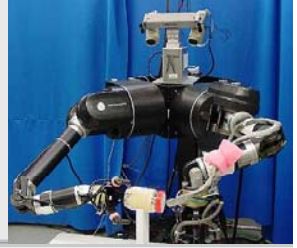
Roberts Operator

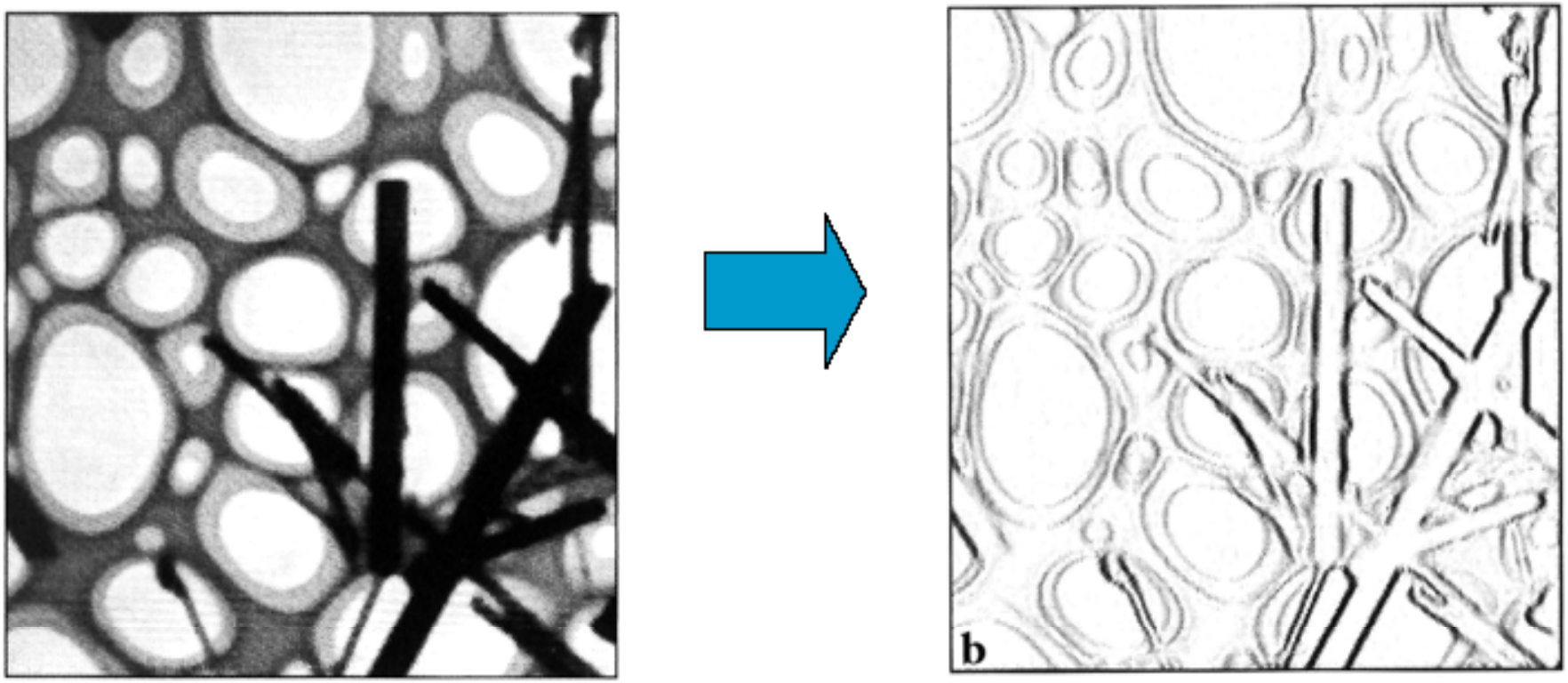




- Not orientation independent („anisotropic")
- Noise problem (2 Pixel only)

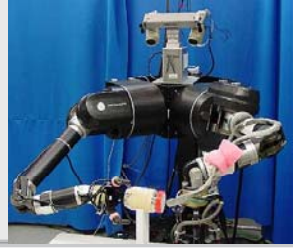
Roberts Operator





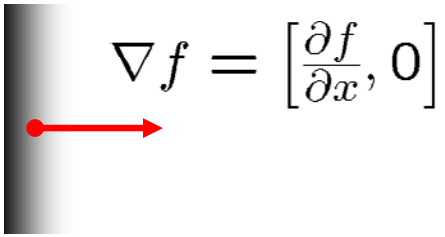
Aus: John C. Russ, *The Image Processing Handbook*, CRC Press (1998)

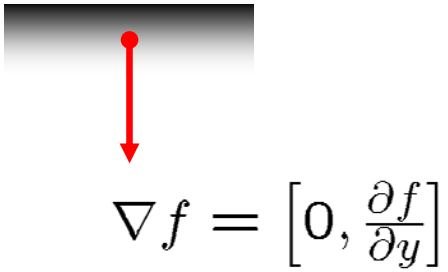
Image Gradient

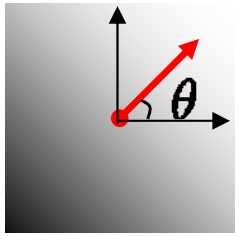


- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

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



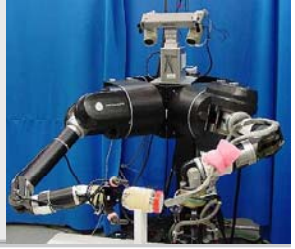


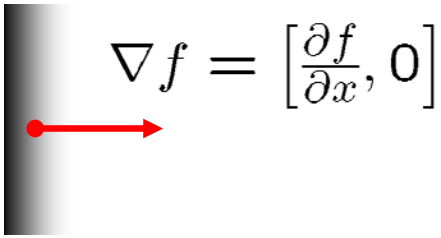


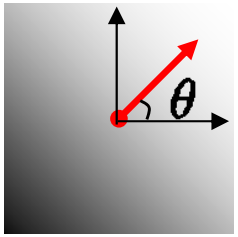
- The *edge strength* is given by the gradient magnitude:

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

Image Gradient







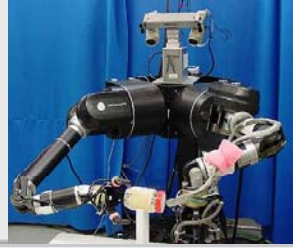
- The gradient points in the direction of most rapid increase in intensity

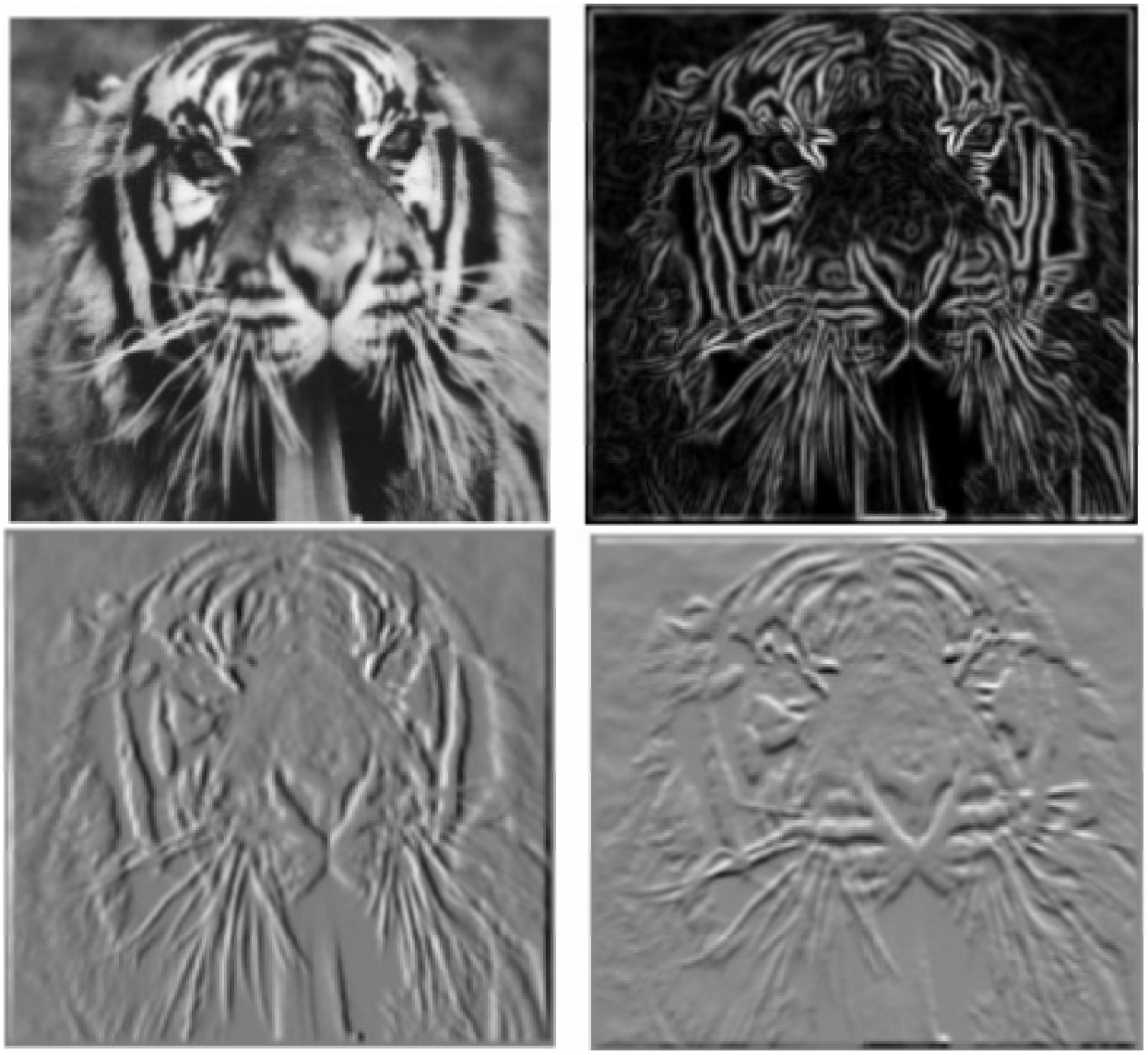
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

- **The gradient of a surface at a point defines the tangential plane to the surface at this point**

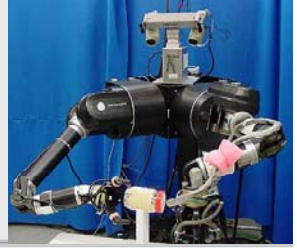
- how does this relate to the direction of the edge?

Image gradient





Sobel Operators



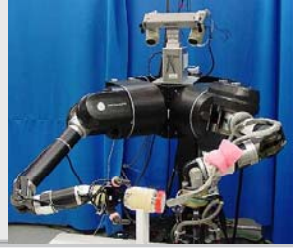
Based on the previous equations we can derive the **Sobel Operators**

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

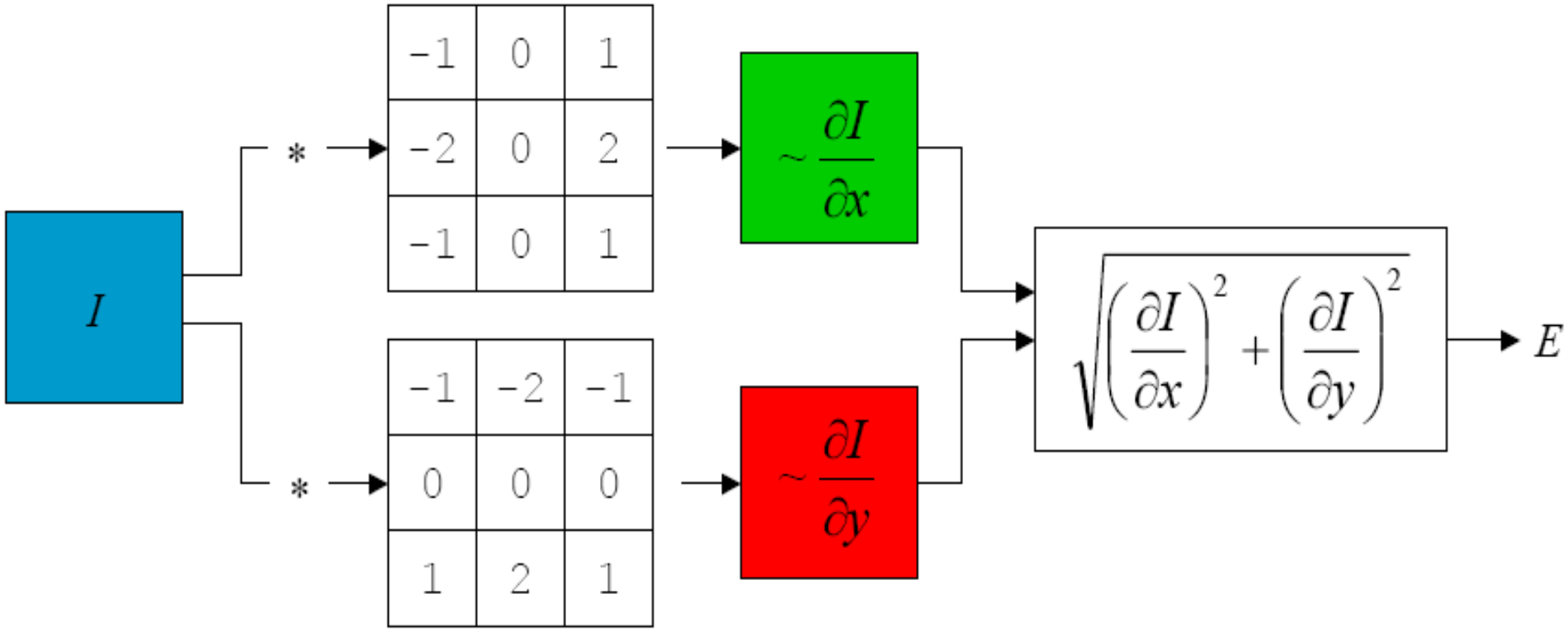
-1	0	1
-2	0	2
-1	0	1

To filter an image it is filtered using both operators the results of which are added together

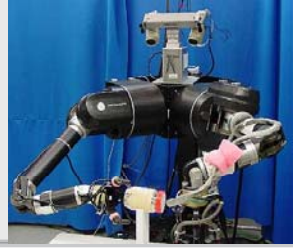
Sobel Filter



- Less noise dependent than Robert's (due to bigger filter size)



Sobel Filter



- Mathematical approximation of first derivative.
- Drawback: directional!
- **Example:** 3x3 Sobel Filter for **x and y direction**:

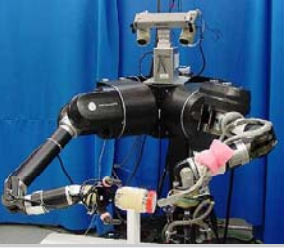
$$S_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad S_Y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

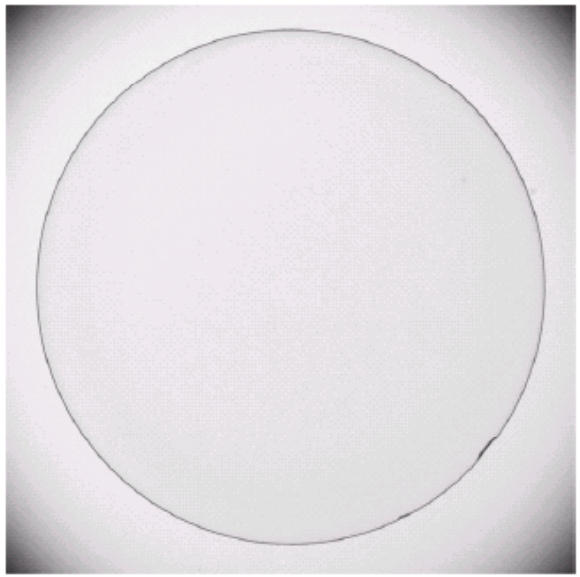
- Example: 3x3 Sobel Filter for **diagonal directions**:

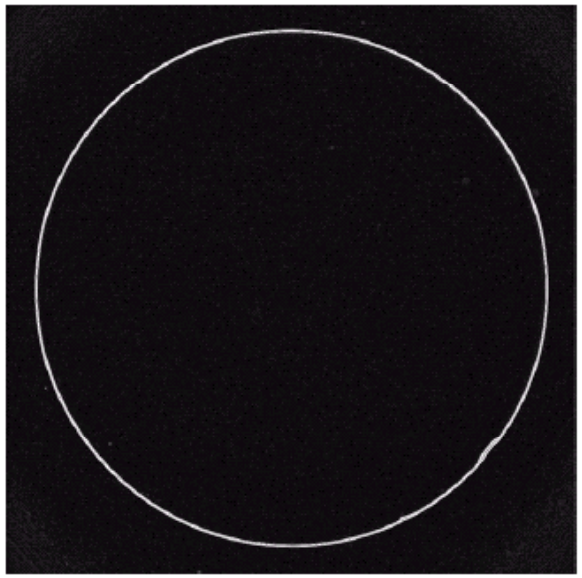
$$S_{d_1} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \qquad S_{D_2} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

- Sobel Filter can only detect edges **perpendicular to** the **direction of** the **filter**.

Sobel Example



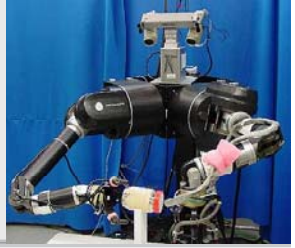


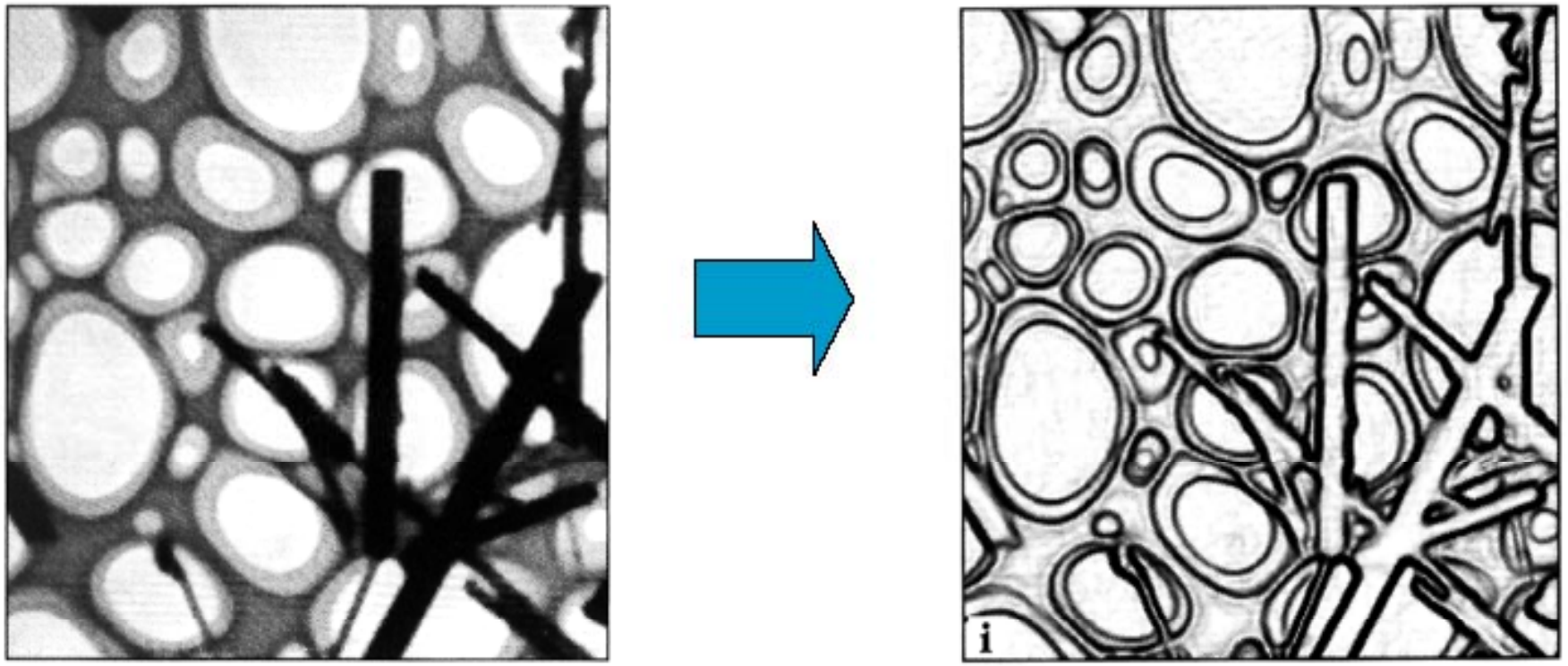


An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious

- Sobel filters are typically used for edge detection

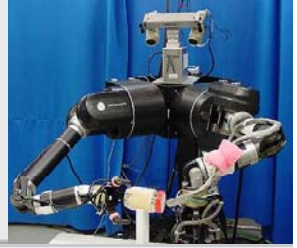
Sobel Filter





Aus: John C. Russ, *The Image Processing Handbook*, CRC Press (1998)

Sobel (noisy image)



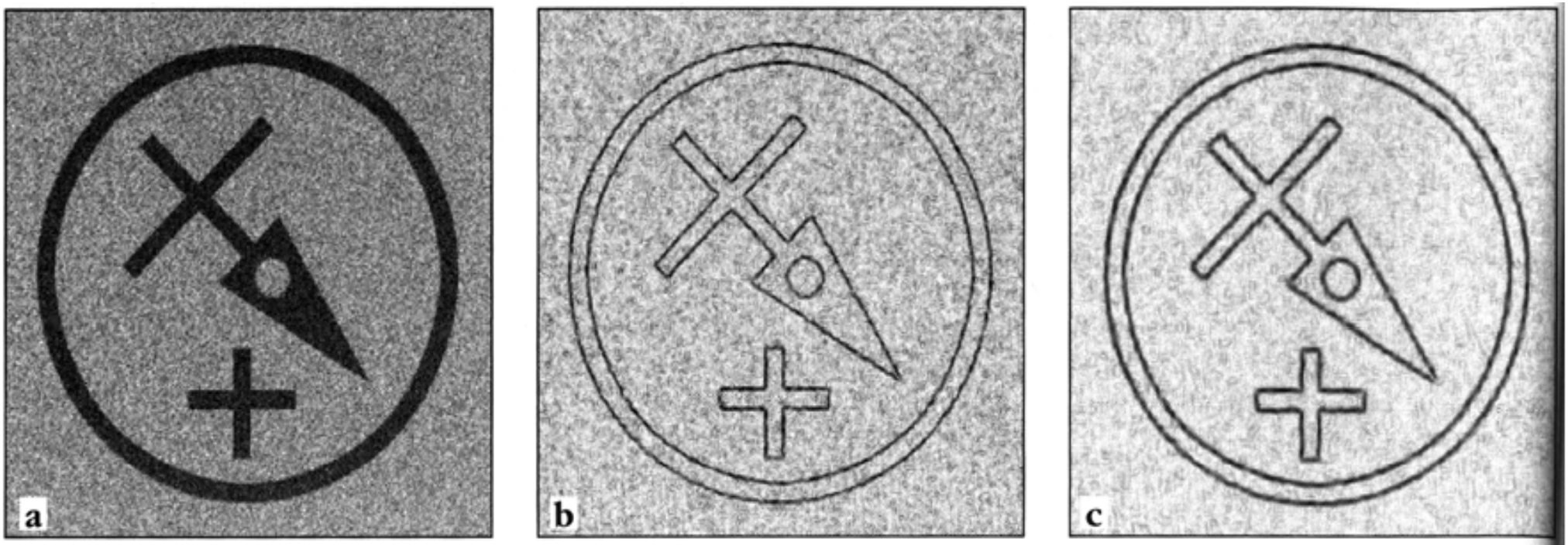
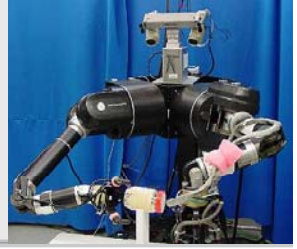
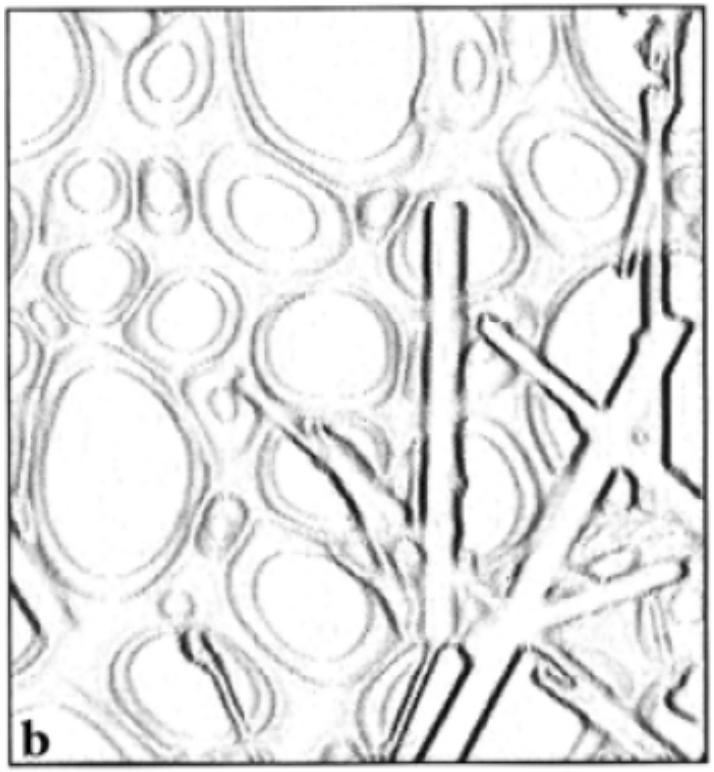


Figure 36. *Sobel magnitude images on a noisy test image* ***(a)*** *using 3×3* ***(b)*** *and 5×5* ***(c)*** *kernels.*

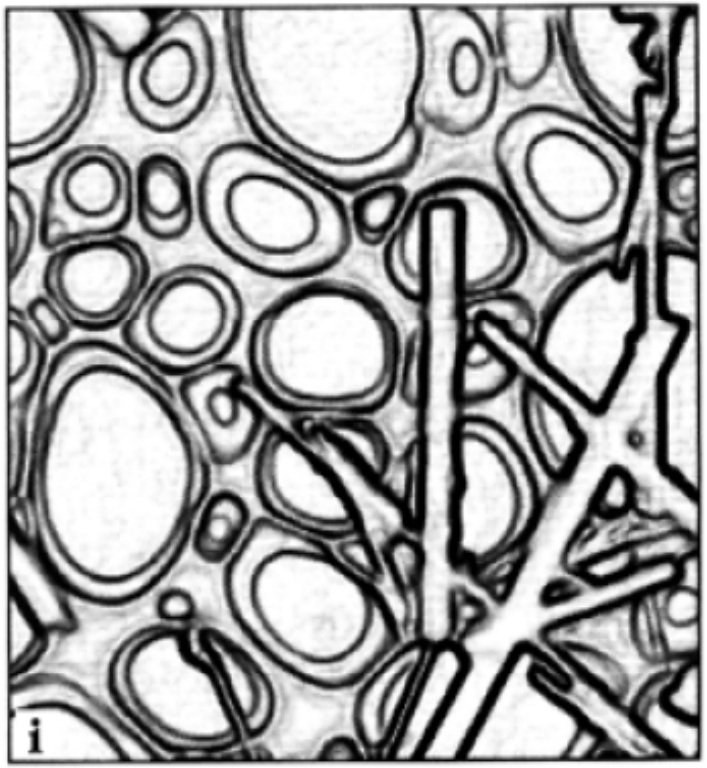
Roberts vs. Sobel Operator



Roberts

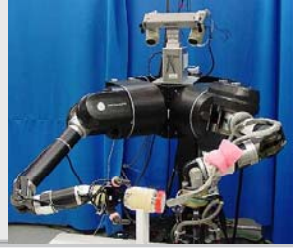


Sobel

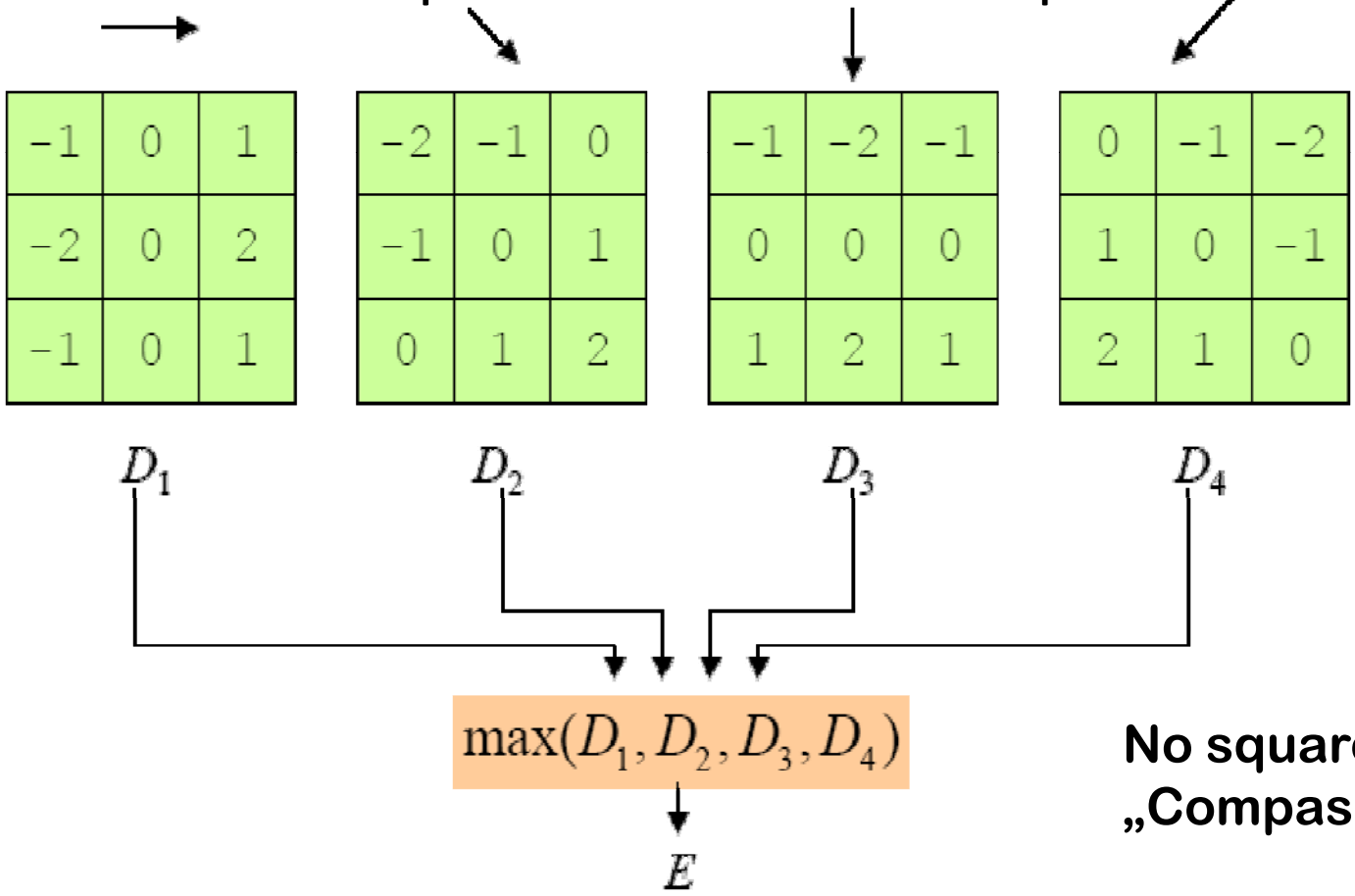


Aus: John C. Russ, *The Image Processing Handbook*, CRC Press (1998)

Kirsch-Operator (1971)

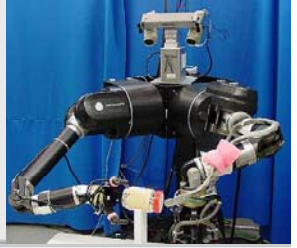


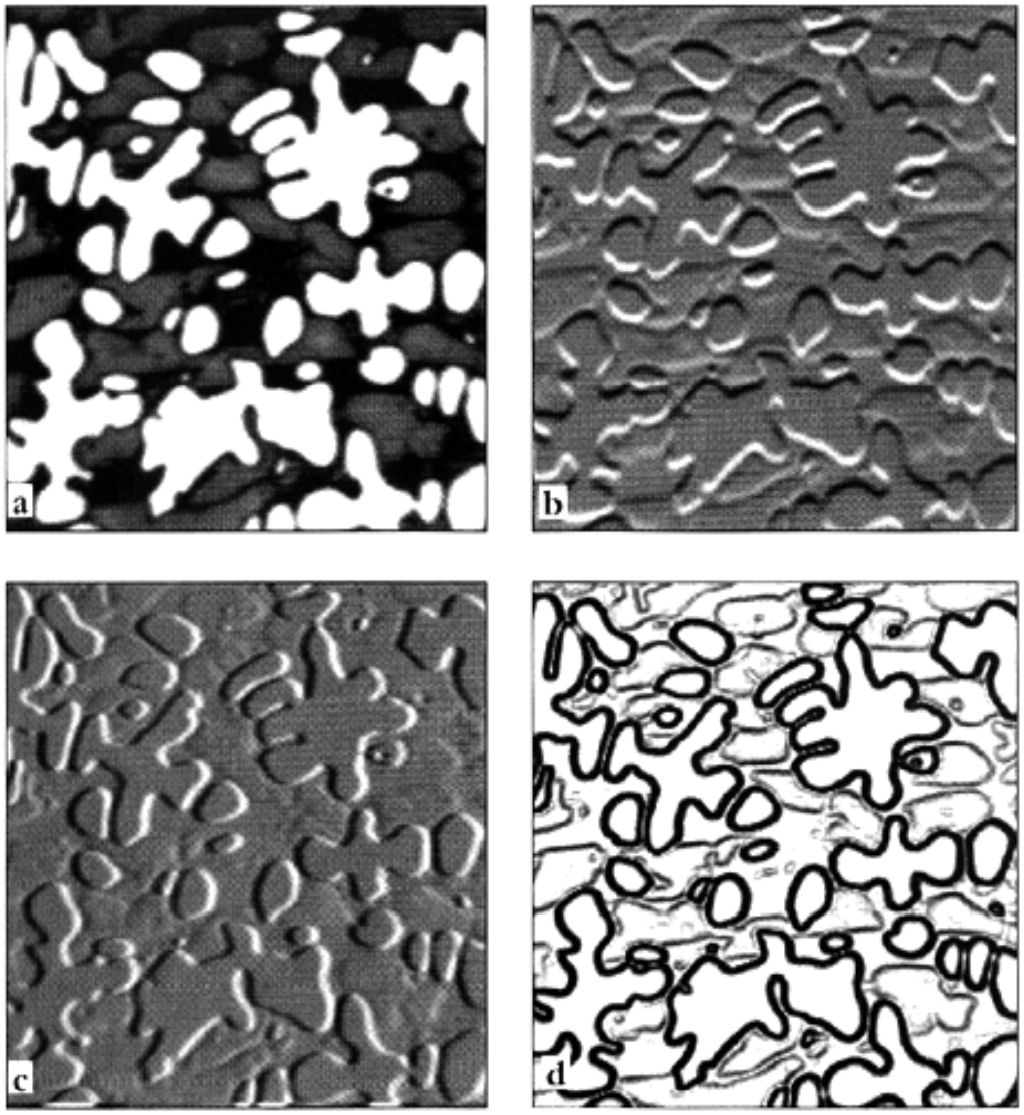
- Additional filters (different orientations)
- Less orientation dependent than Sobel-Operator



No square root!
„Compass“ Operator

Kirsch-Operator

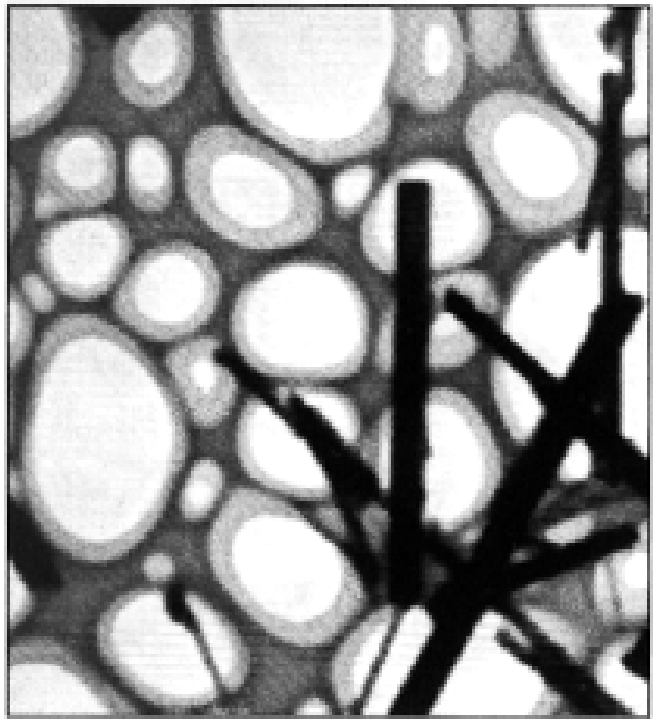


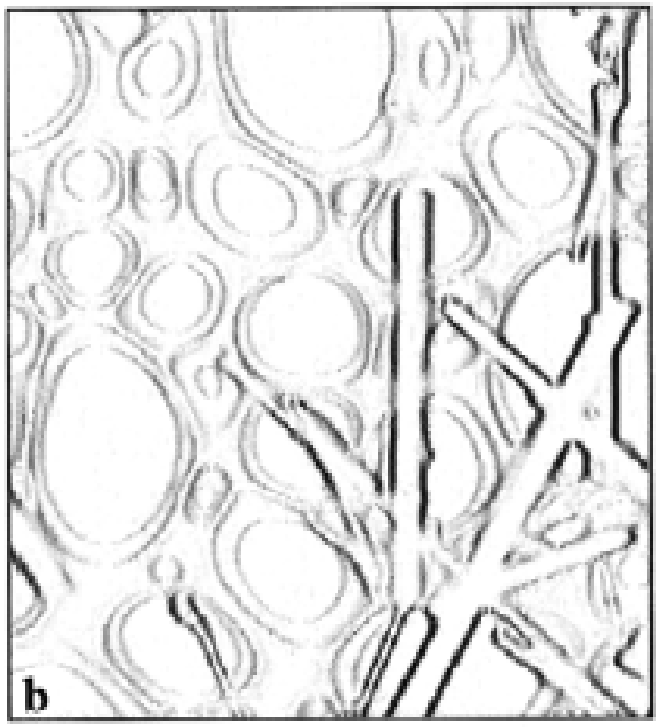


2 von 8
Orientierungen

Aus: John C. Russ,
The Image Processing Handbook,
CRC Press (1998)

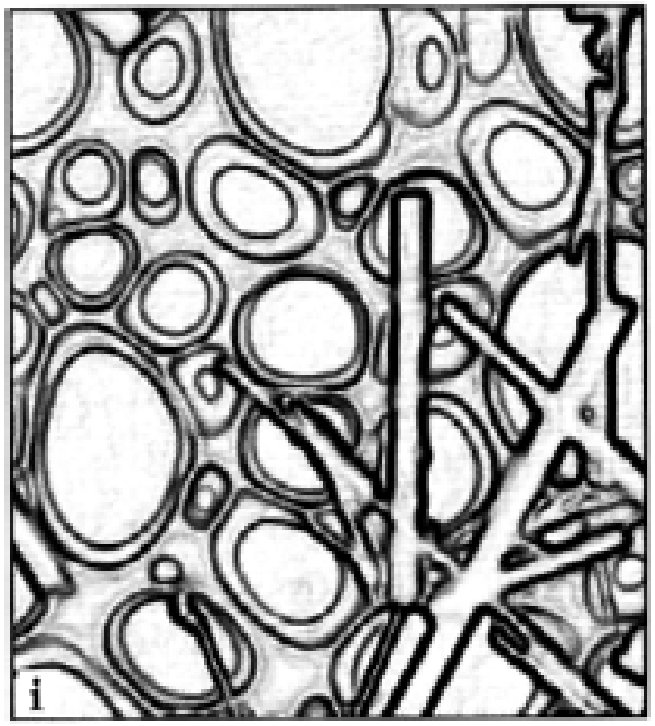
Original

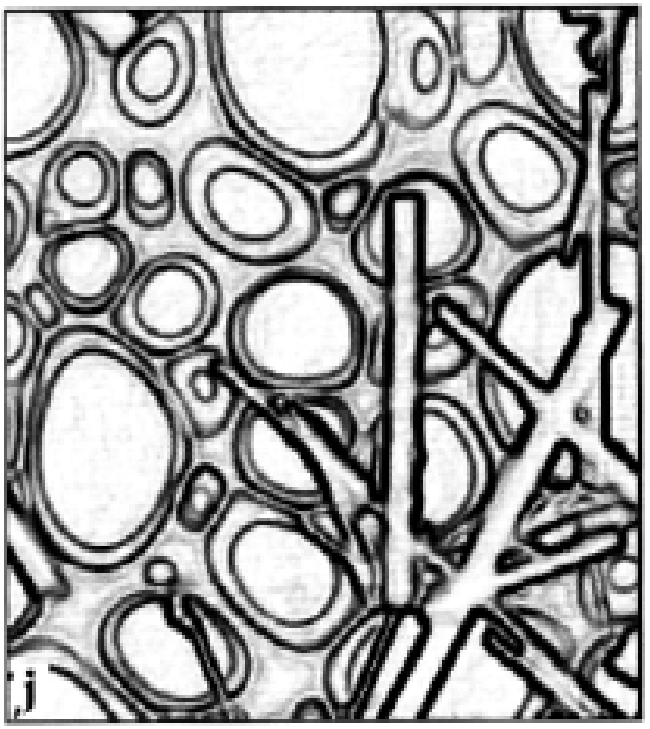




Roberts

Sobel

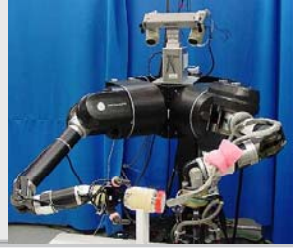




Kirsch

Aus: John C. Russ,
The Image Processing Handbook,
CRC Press (1998)

Related Operators



$$\delta x \qquad \delta y$$

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

2x2 Roberts

$$\delta x \qquad \delta y$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

3x3 Prewitt

$$\delta x \qquad \delta y$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

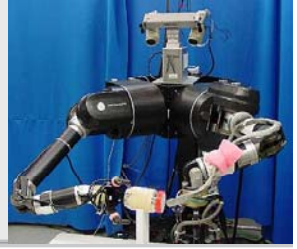
3x3 Sobel

$$\delta x \qquad \delta y$$

$$\begin{bmatrix} -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \end{bmatrix} \qquad \begin{bmatrix} 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -3 & -3 & -3 & -3 \end{bmatrix}$$

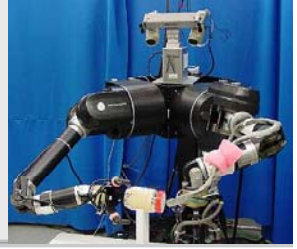
4x4 Prewitt

Edge Point



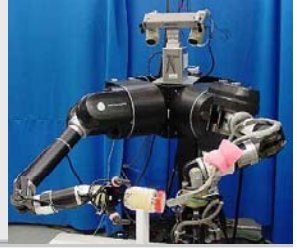
- To determine a point as an edge point
  - Determine the transition in grey level associated with the point which is significantly stronger than the background at that point.
  - Use threshold to determine whether a value is “significant” or not.
  - Note that the point’s two-dimensional first-order derivative must be greater than the specified threshold

Example



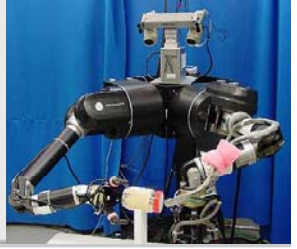
original image (Lena)

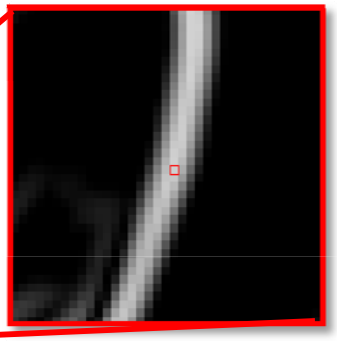
Finding Edges



gradient magnitude

Finding Edges

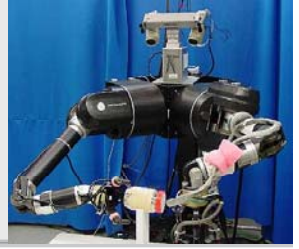


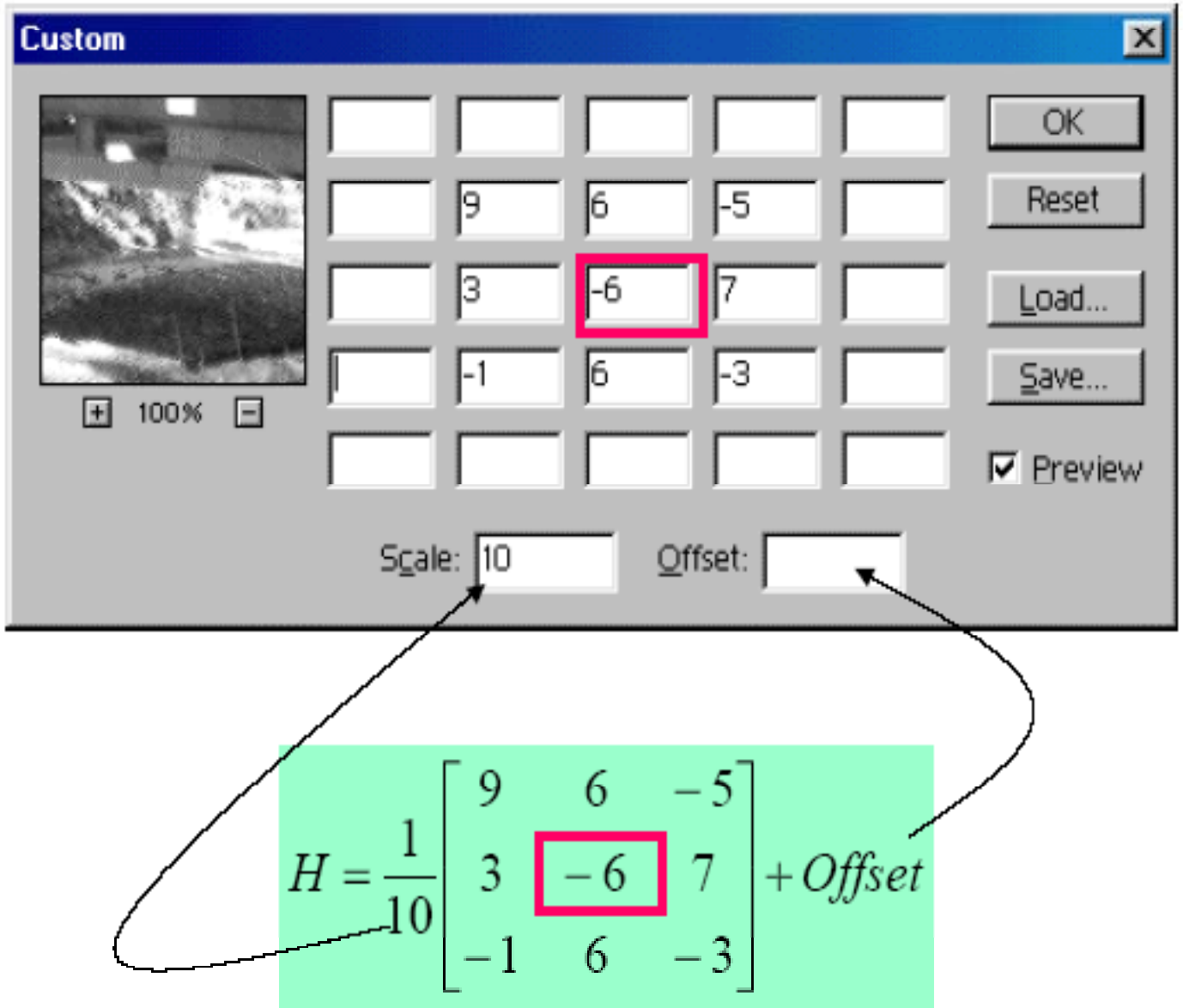


where is the edge?

thresholding

Photoshop: Other Filters - Custom Filter



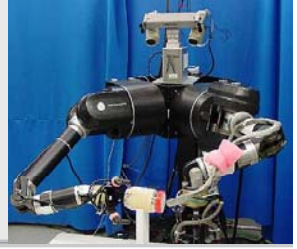


TU WIEN

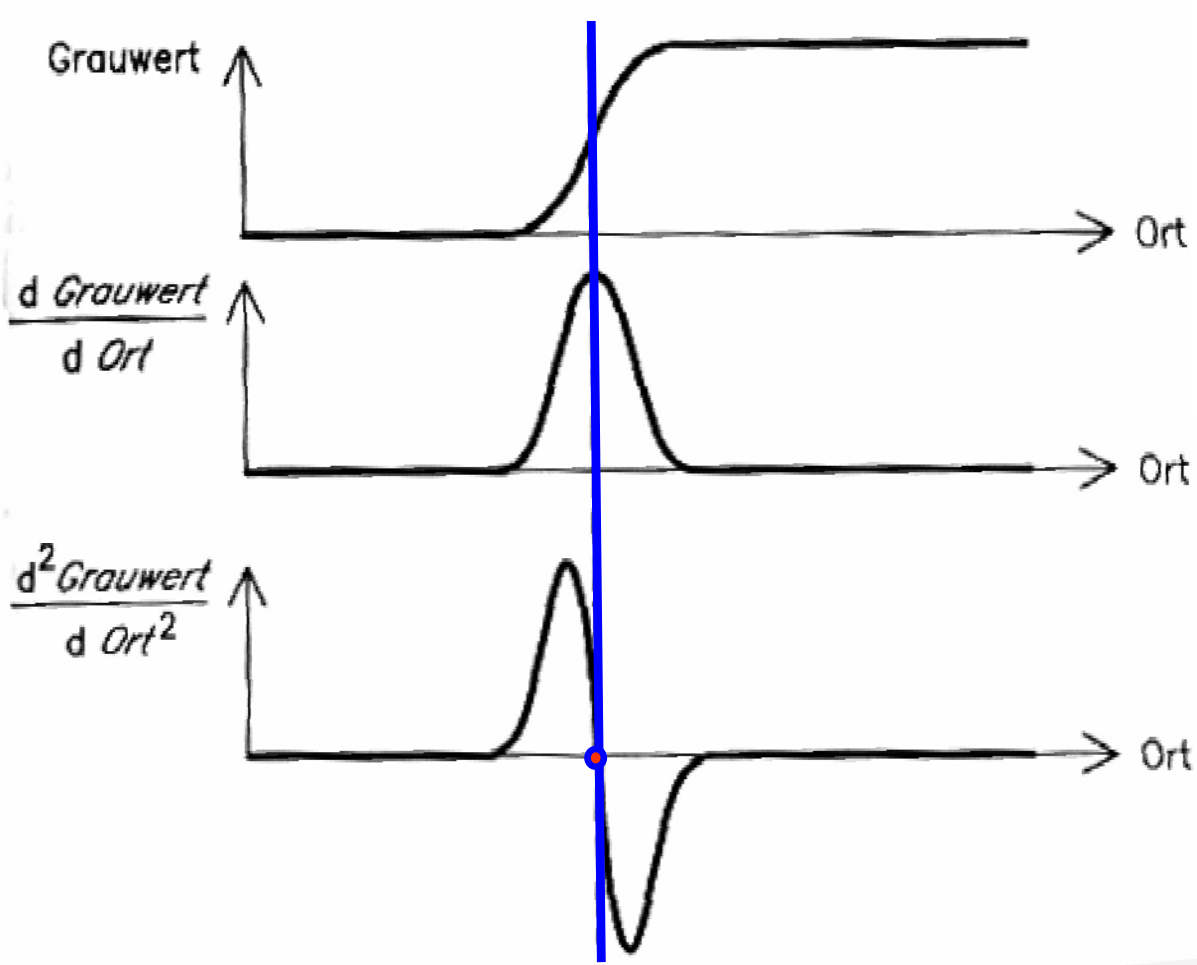
CVL Computer Vision Lab

Second Derivatives Filter

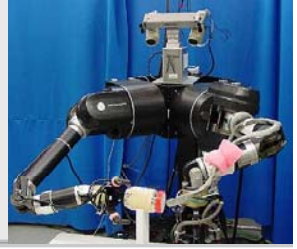
Zero Crossing



To estimate the position of the edge more precisely the zero crossing of the 2nd derivative is determined



Laplace Filter

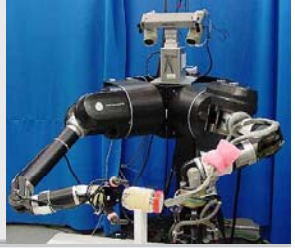


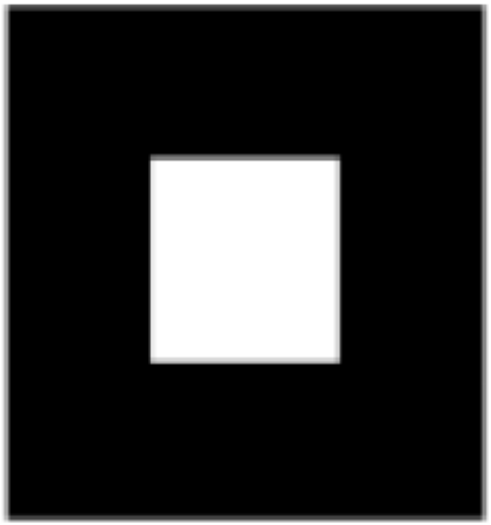
- Mathematical approximation of second derivative. Drawback: **not directional anymore!**

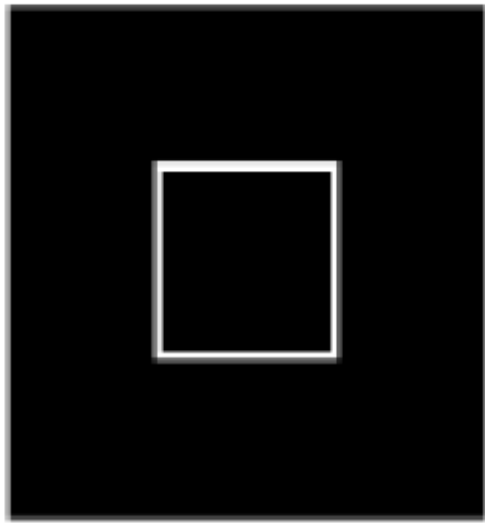
$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{or} \quad L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Because of the high pass characteristics (2nd order filter = 2nd derivative) the Laplace filter is **very sensitive to noise**.
- Therefore it is rarely applied alone. Usually it is combined with a Gaussian Filter that reduces noise before the Laplace filter can be applied.

Laplace Filter

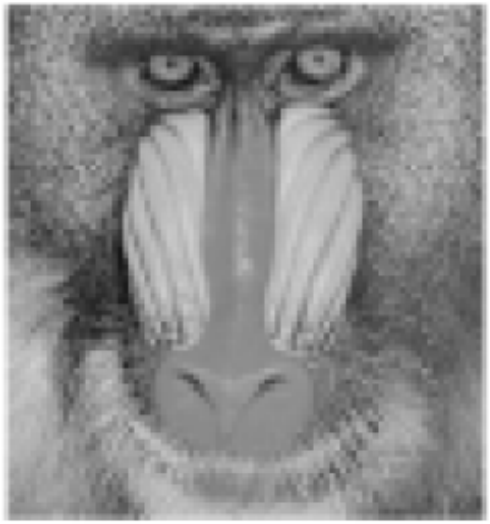




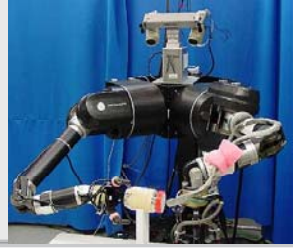


$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

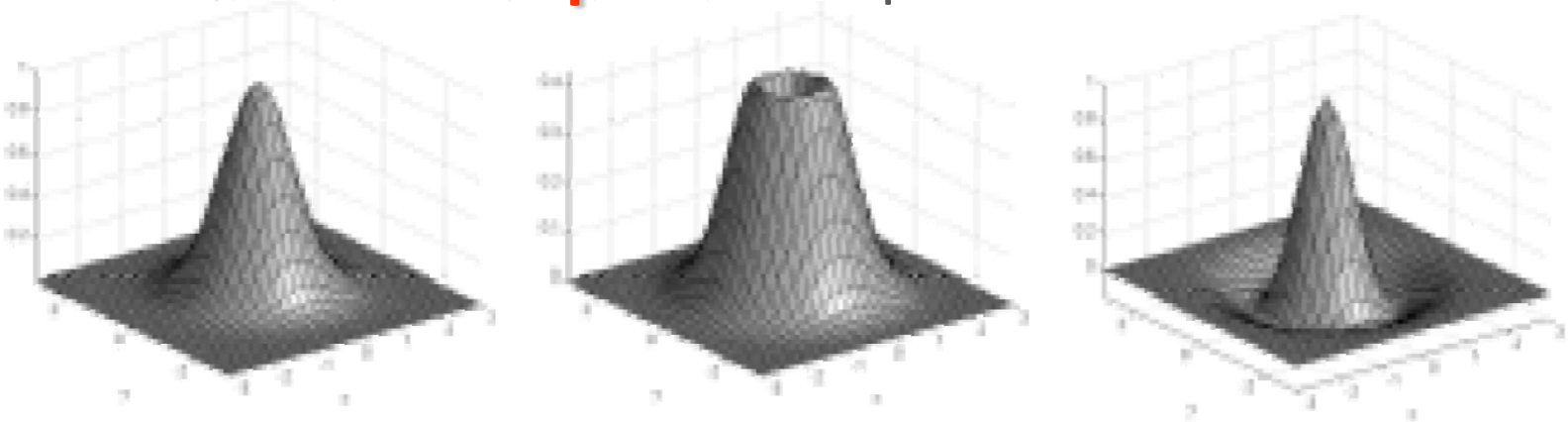
not directional



Laplace of Gaussian (LoG)



- Combination of **Gaussian Filter** and **Laplace Filter**
- Combination corresponds to second derivative of the 2D Gaussian function Laplacian of Gaussian filter (LoG):
  - Because of the shape of its kernel elements the LoG filter is usually called "**Mexican Hat**" filter
  - LoG Filter can be used for Edge Detection
  - LoG Filter **does not depend** on a particular direction.

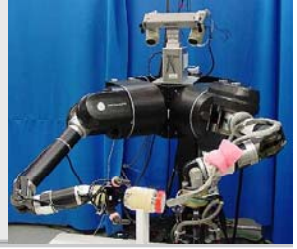


Gauss' Bell Function

First derivative

Minus second derivative „Mexican Hat"

Laplace of Gaussian (LoG)



- Second derivative of the 2D Gaussian distribution:

$$f(x,y) = \frac{1}{\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} = -\frac{x+y}{\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

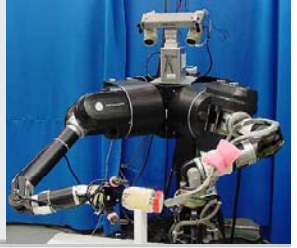
$$-\nabla f(x,y) = -\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 f}{\partial y^2}$$

$$= \frac{1}{\sigma^4}\left(2 - \frac{x^2+y^2}{\sigma^2}\right) e^{\frac{x^2+y^2}{2\sigma^2}}$$

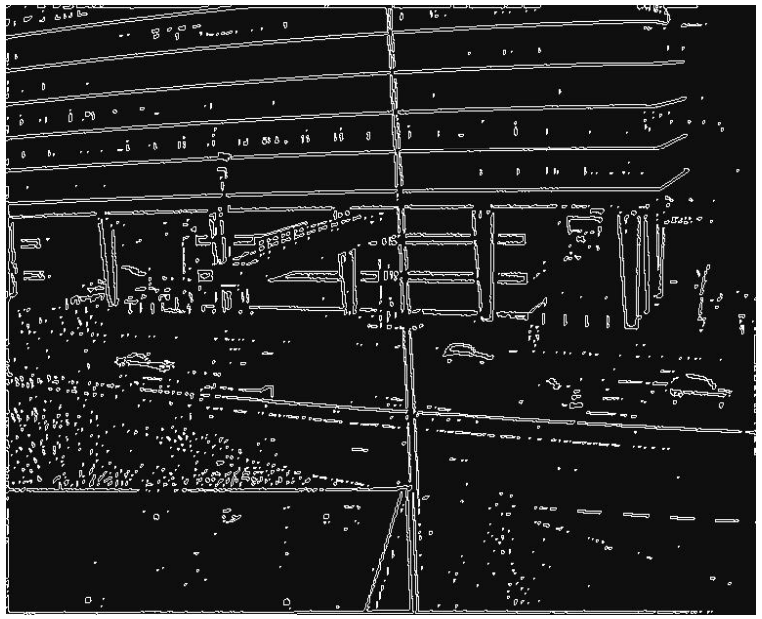
$$\mathbf{LoG} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

5x5 LoG filter kernel

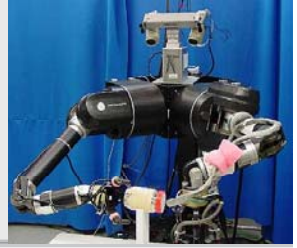
Laplace of Gaussian (LoG)



$$\mathrm{LoG} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$



Zero Crossing vs. Gradient



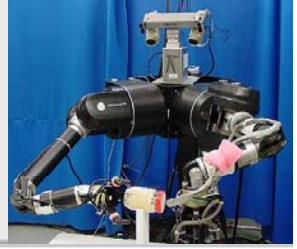
- Attractive
  - Zero crossing produces thinner edges
  - Noise reduction
- Drawbacks
  - sophisticated computation.
- Gradient is more frequently used.

TU WIEN

CVL Computer Vision Lab

Canny Edge Detector

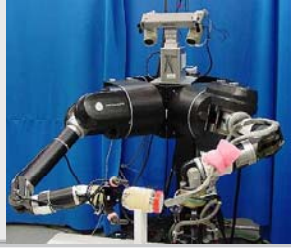
Canny/Deriche Edge Detector



- Require
  - Edges to be detected
  - Accurate localisation
  - Single response to an edge

- Solution
  - Convolve image with Difference of Gaussian (DoG)

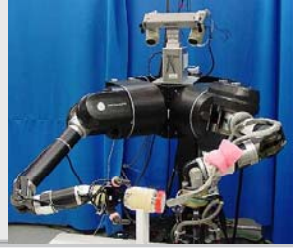
Canny Edge Detector



MATLAB: `edge(image,'canny')`

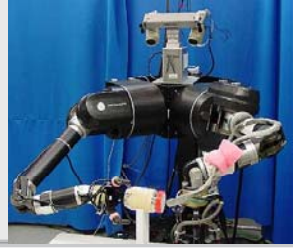
1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

Stages of the Canny algorithm



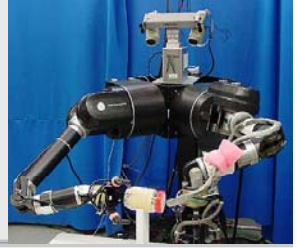
- Noise reduction: raw image is convolved with a Gaussian filter
- Finding the intensity gradient of the image
  - Intensity gradient is estimated from the smoothed image using simple horizontal and vertical difference operators
  - Gradient direction together with the gradient magnitude then gives an estimated intensity gradient at each point in the image
  - Canny algorithm uses both gradient magnitude and direction in the edge detection

Stages of the Canny algorithm



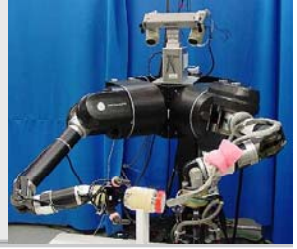
- Non-maximum suppression:
  - A search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction
  - From this stage, referred to as non-maximum suppression, a set of edge points in the form of a binary image are obtained
  - Output of this stage is sometimes referred to as "thin edges"

Stages of the Canny Algorithm



- Threshold
  - Large threshold: gives true edges but does not detect all
  - Small threshold: gives false edges
- Canny algorithm does not use same threshold for whole image
  - It does thresholding with hysteresis
- Thresholding with hysteresis requires
  - Two thresholds – high and low
  - Therefore we begin by applying a high threshold

Stages of the Canny Algorithm



Original image

↓

Smoothing by Gaussian convolution

↓

Differential operators along x and y axis

↓

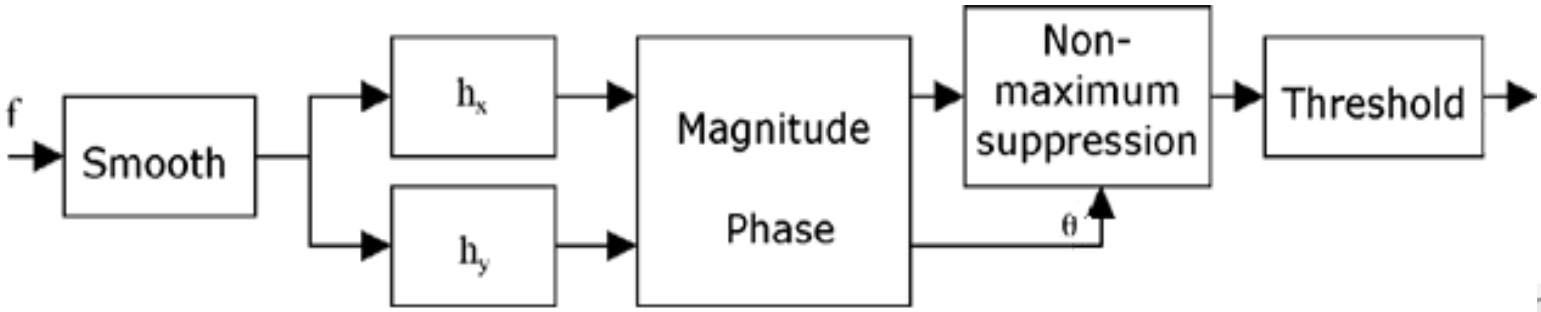
Non-maximum suppression
finds peaks in the image gradient

↓

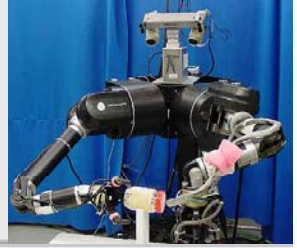
Hysteresis thresholding locates edge strings

↓

Edge map



Example

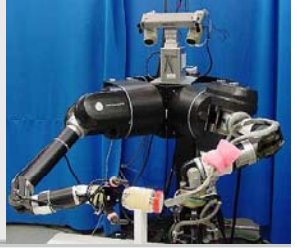


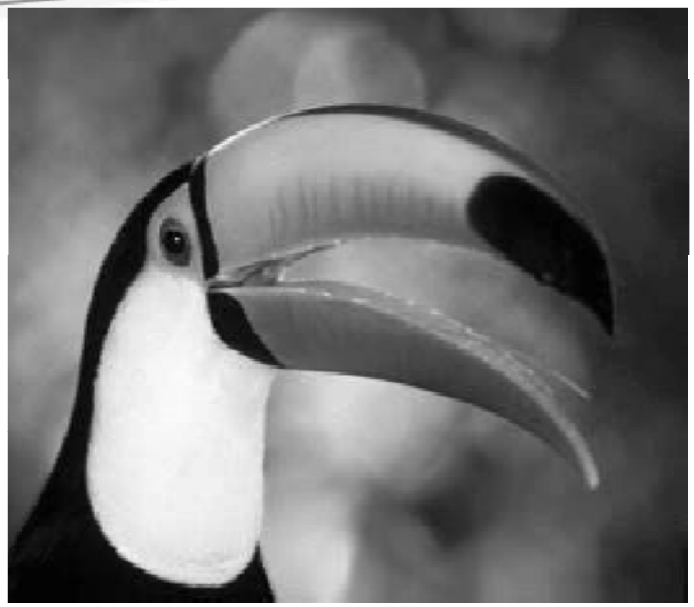
Sobel

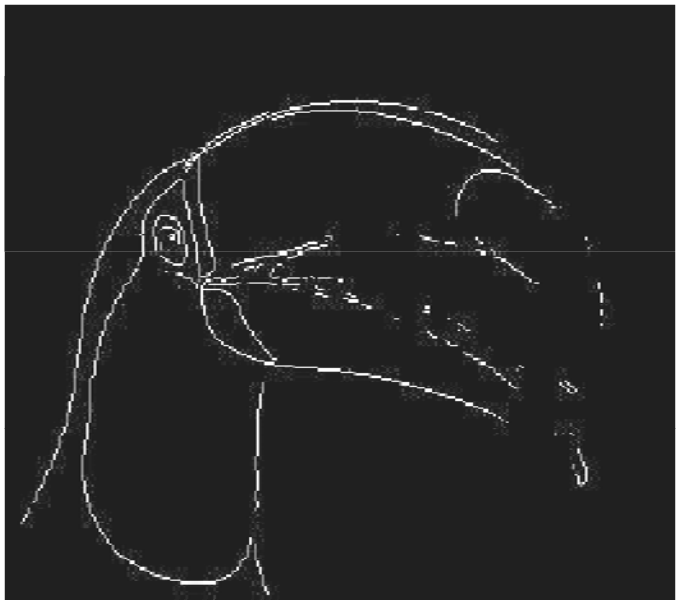
LOG

Canny

Example

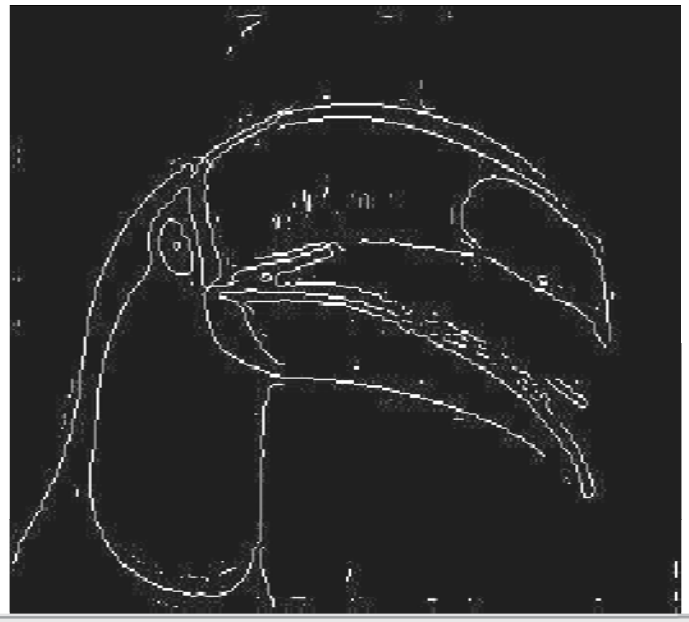


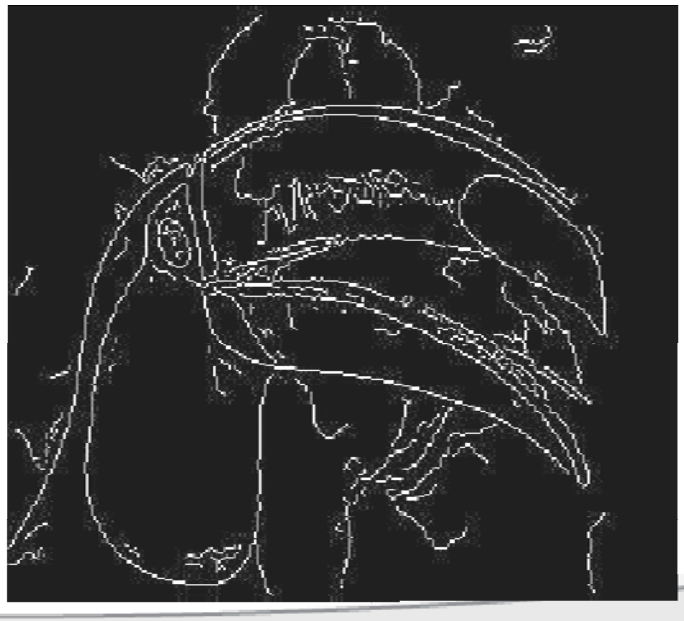




Sobel

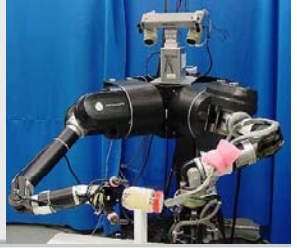
LOG

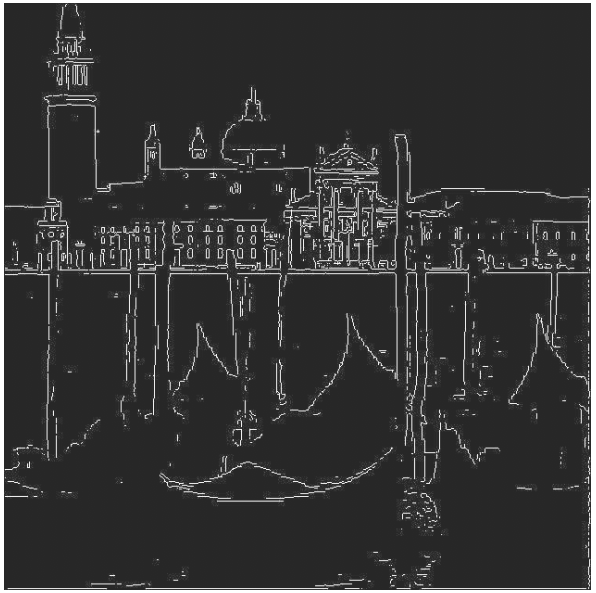




Canny

Example



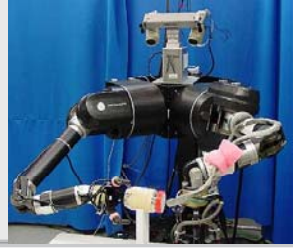


Sobel

LOG

Canny

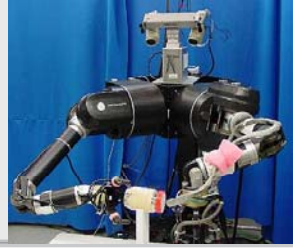
Canny Edge Detector [Canny 1986]



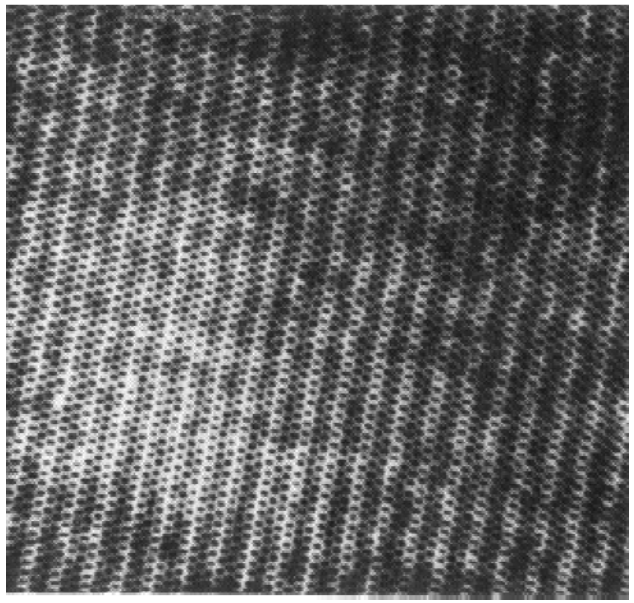
- Still one of the most widely used edge detectors in computer vision
- Corresponds to edges of different sizes
- Uses relatively large, directed filters
- Is complex (if implemented fully)
- Is preferable in very demanding applications (in contrast to Sobel and Kirsch)

[Canny 1986] J. Canny, ***A Computational Approach To Edge Detection***, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.

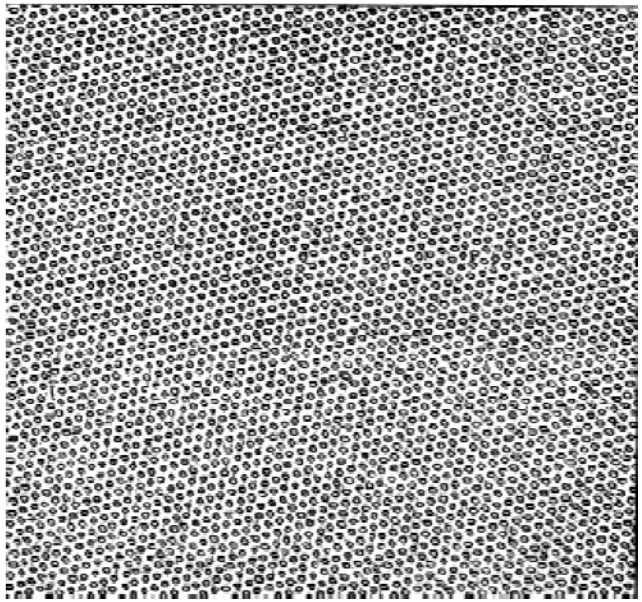
Resolution steps



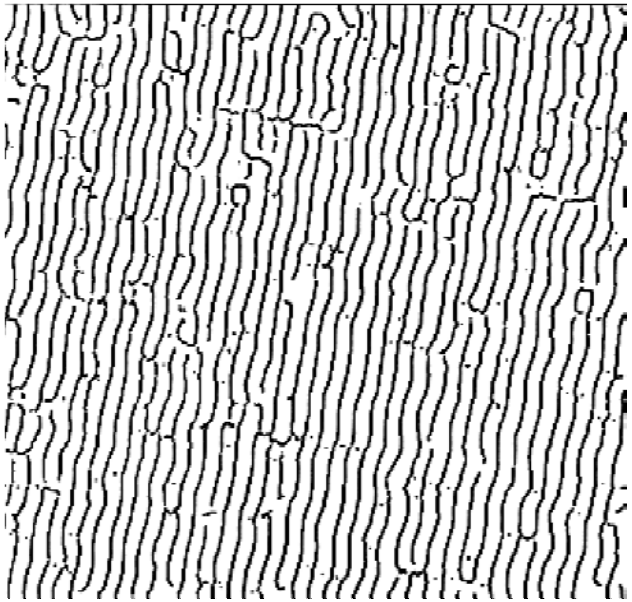
- Why different resolutions?
- Gaussian has different σ



Original

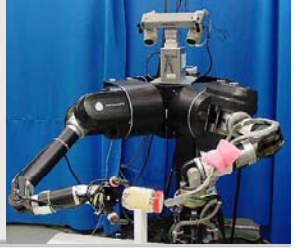


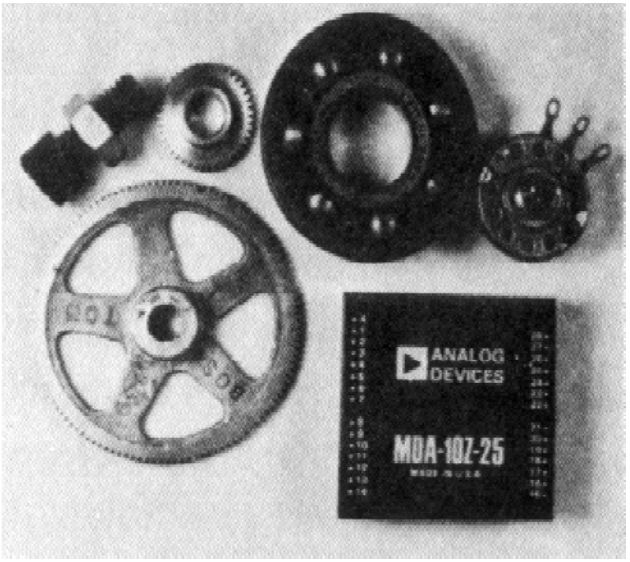
σ = 1.0



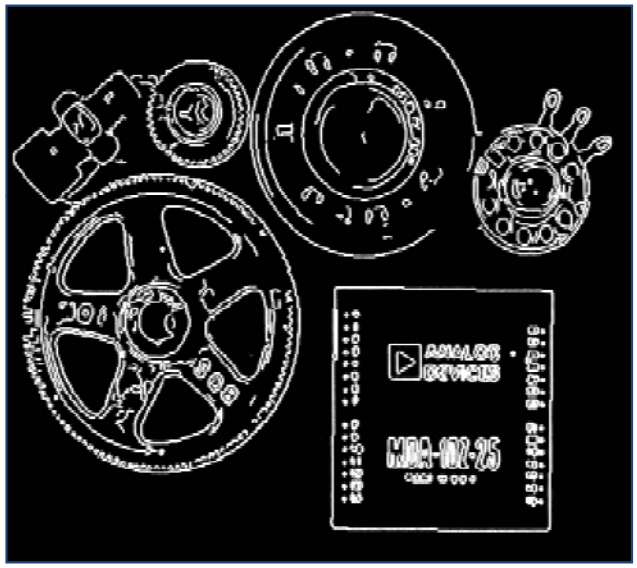
σ = 5.0

Canny Edge Detector

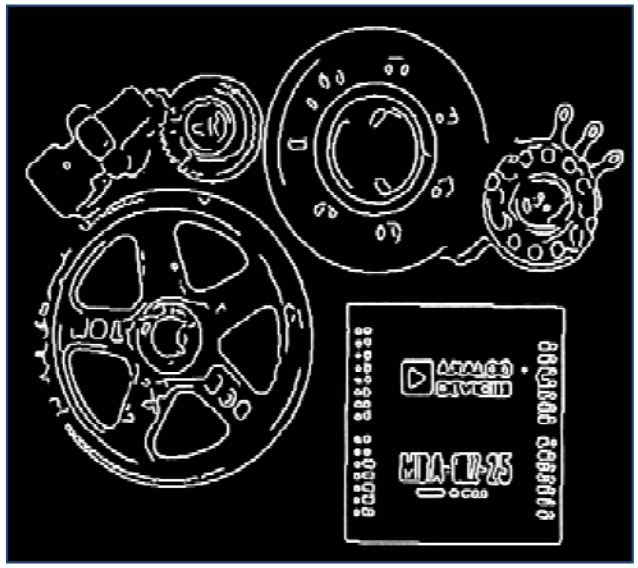




Original

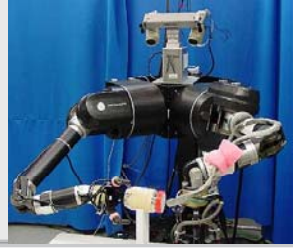


σ = 1.0



σ = 2.0

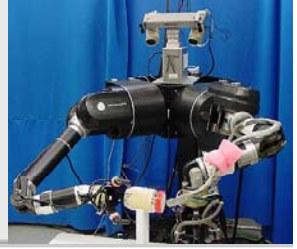
1st & 2nd Derivatives

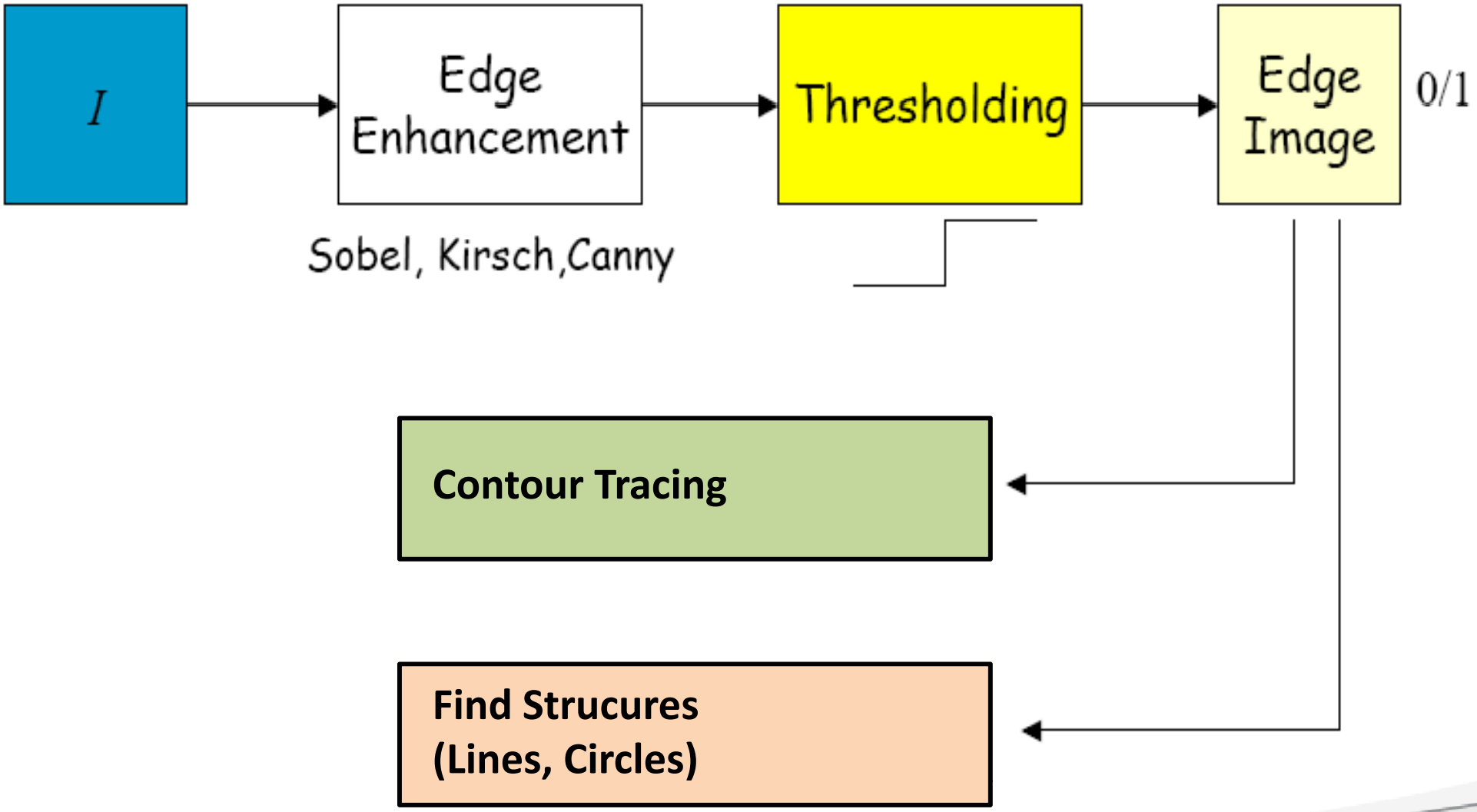


Comparing the 1st and 2nd derivatives we can conclude the following:

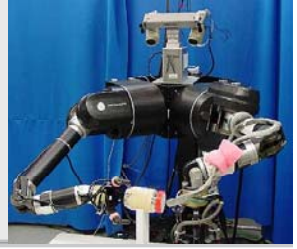
1. 1st order derivatives generally produce thicker edges
2. 2nd order derivatives have a stronger response to fine detail e.g. thin lines
3. 1st order derivatives have stronger response to grey level step
4. 2nd order derivatives produce a double response at step changes in grey level

From Edgels to Edges





Edge Linking and Boundary Detection



- Edge detection algorithm are followed by linking procedures to assemble edgels into meaningful edges.

- Basic approaches
  - Local Processing
  - Global Processing via the Hough Transform
  - Global Processing via Graph-Theoretic Techniques