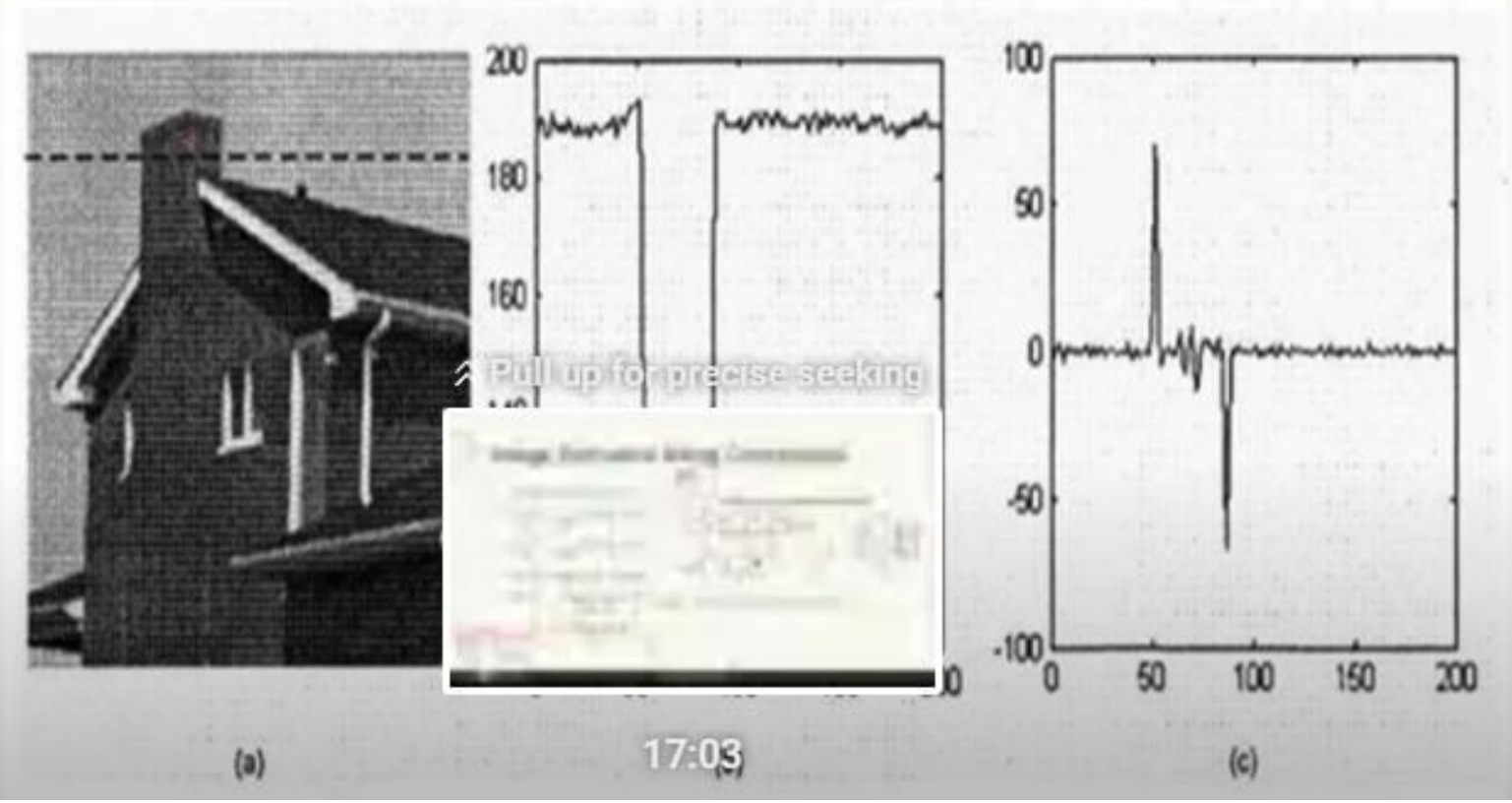
Outline

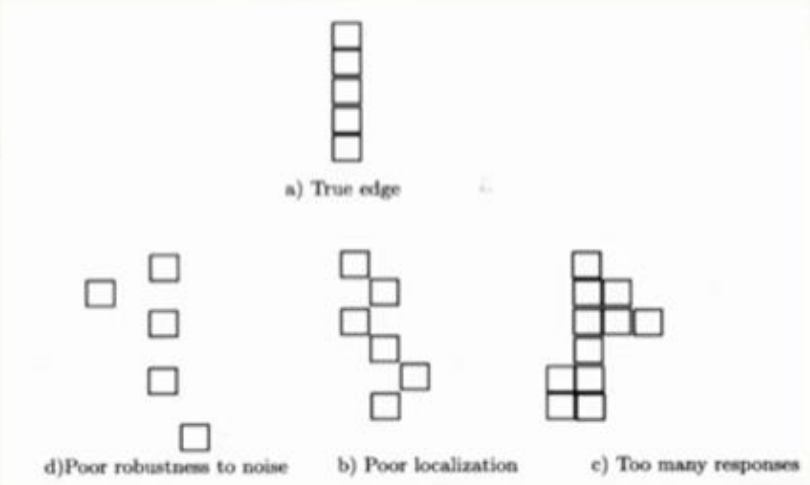
- Edge Filters (High Pass Filters)
- Canny Edge Detector (Paper published by John Canny in 1986 based on Edge Detection
  Most robust algorithm until now)

A Single Image Scanline



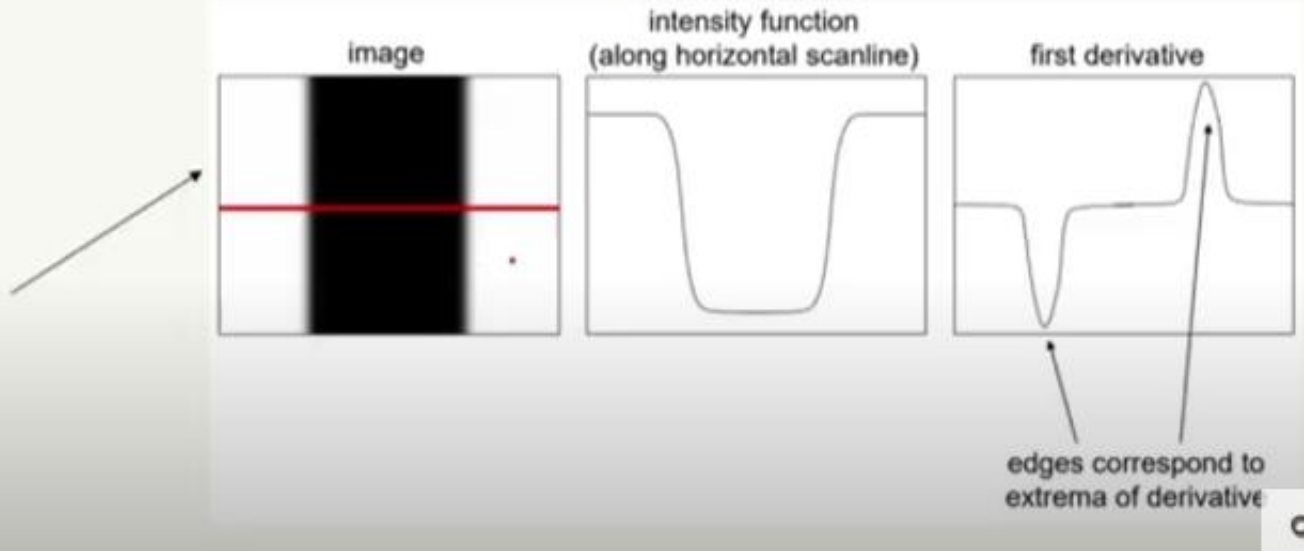
What are Edges?

- Edges are the abrupt changes in intensity values.
- Good Edge Detection Algorithm should have:
  - Filter should perfectly detect edges
  - Edges must be localized
  - One response per edge(not multiple!)

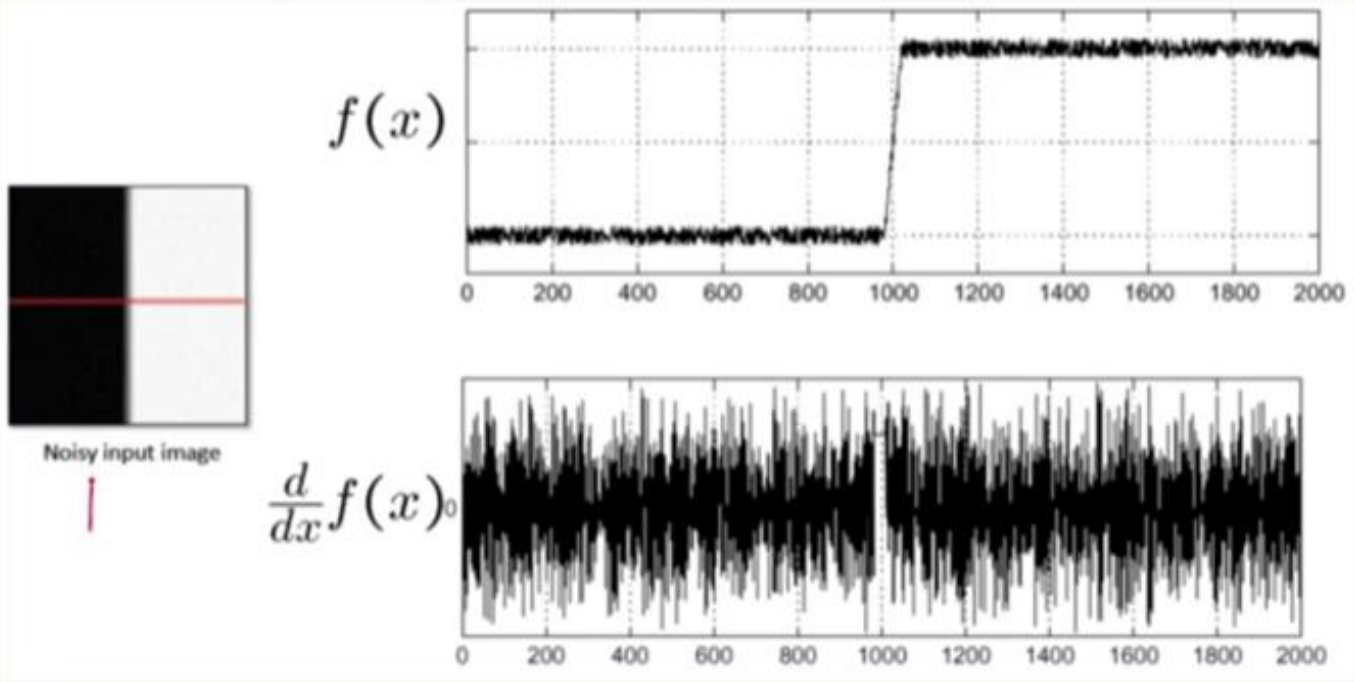


Edges(Cont...)

- Clearly, edges are the points where magnitude of derivative is large



If we don't smooth ?



A Notion of 2nd Derivative

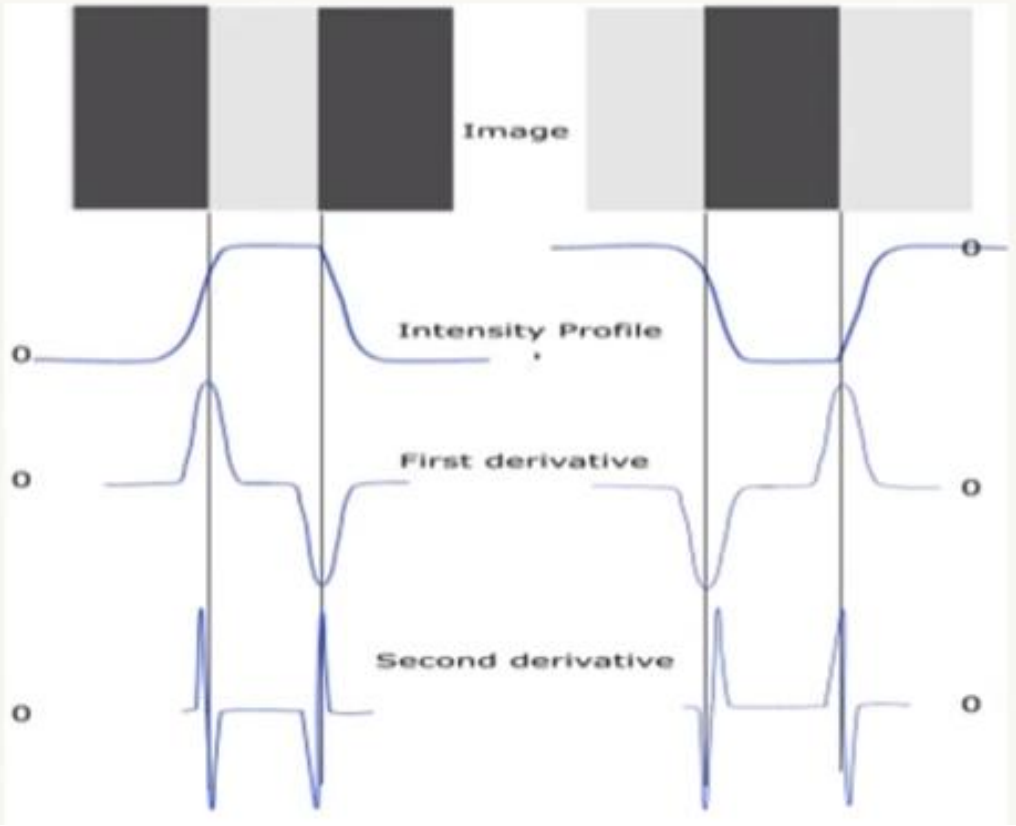


Image Derivative Using "Convolution"

- *Definition of Derivative:*
- $\frac{\partial f(x,y)}{\partial y} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon,y)-f(x,y)}{\epsilon}$
- $\frac{\partial f(x,y)}{\partial y} = \lim_{\epsilon \to 0} \frac{f(x,y+\epsilon)-f(x,y)}{\epsilon}$
- Since images are discrete,
- $\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y)-f(x,y)}{1}$
- $\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1)-f(x,y)}{1}$

$(Finite\ Difference\ Approximation)$

What are the corresponding Kernels?

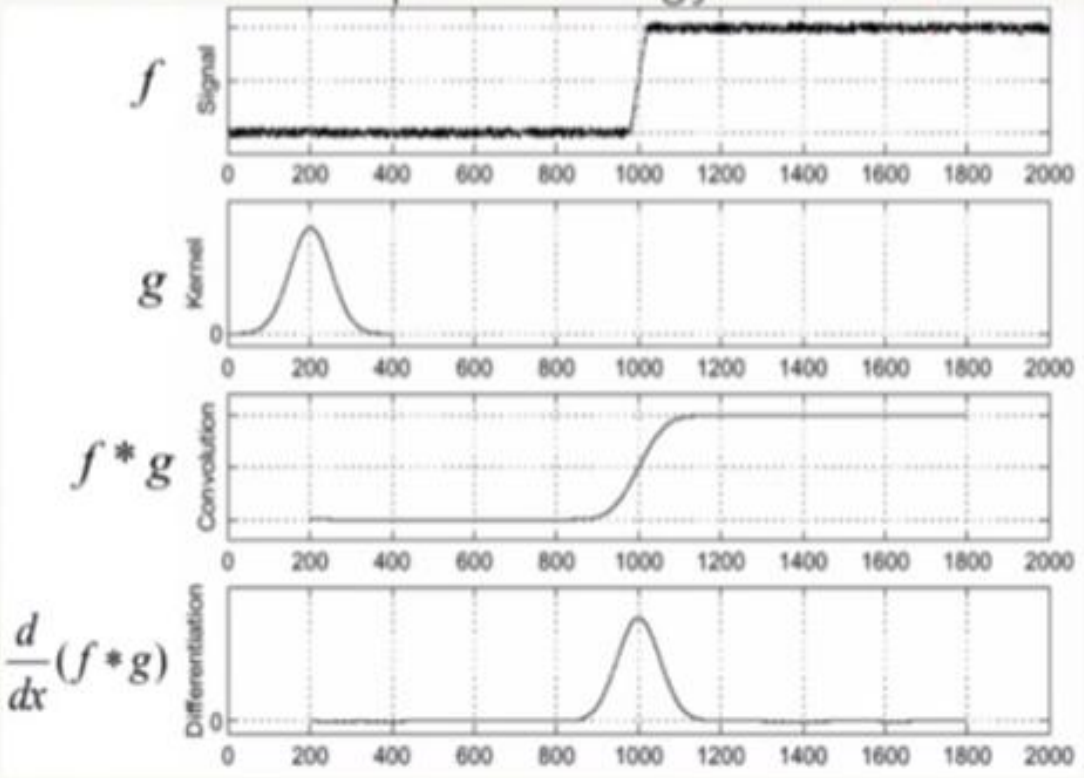
1	-1

-1	1

1	-1
-1	1

Finally Edge Detection(Based on 1st Derivative)

- Now we have the complete strategy to detect an edge



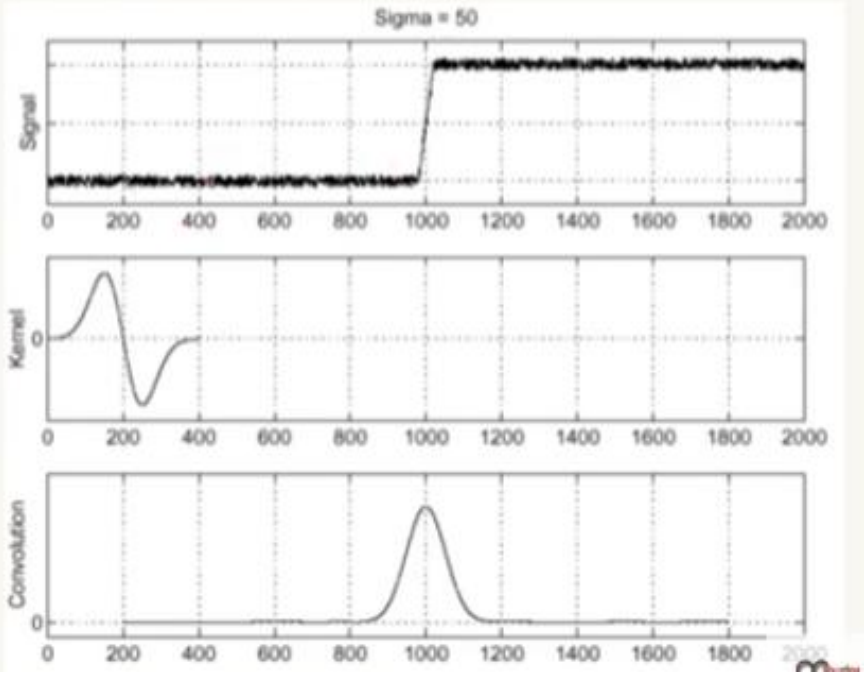
Derivative of Gaussian[DoG](More efficient)

- We used convolution for differentiation and we know convolution is associative:
  - $\frac{d}{dx}(f * g) = f * \frac{d}{dx} g = \frac{d}{dx} f * g$

f

$\frac{d}{dx} g$

$f * \frac{d}{dx} g$



Moving Towards 2D Edge Detection

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

Prewitt Filters

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

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

Sobel Filter

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

1	0
0	-1

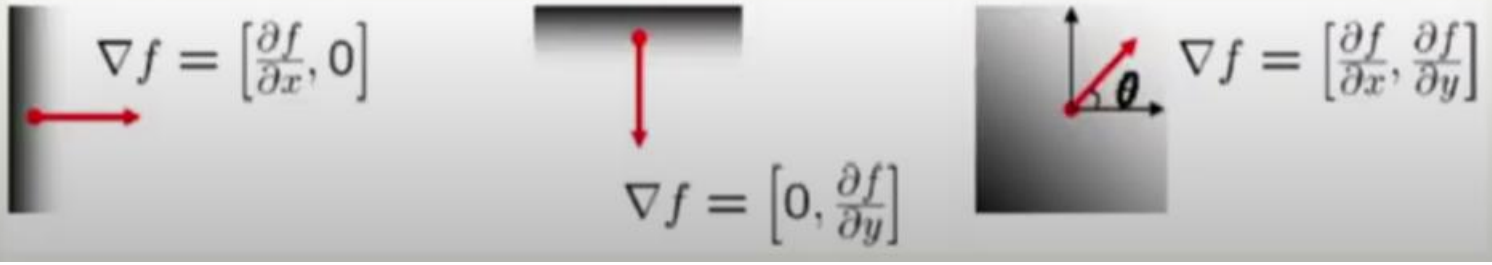
Robert Filter

0	-1
1	0

Need to find edges in all directions separately

Image Gradient

- $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$
- Gradient direction is the most significant change in the intensity
- The direction is given by $\tan^{-1} \frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}$ and magnitude = $\sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$



Canny Edge(1986) Detector(Most Optimal Detector)

- **Assumptions**
  - Only Linear Filtering is applied
  - Noise is Independent and Identically Distributed
- Algorithm
  - Find the derivative of the Gaussian(DoG)
  - Find Gradient Magnitude and Direction
  - Non-Maximum Suppression
  - Thresholding (Hysteresis)
- Note: More Smoothing can lead to Detection/localization tradeoff since more smoothing changes(more blur) the original image structure.

Canny Edge Detection

Original Image

Gradient(Edge Detection)

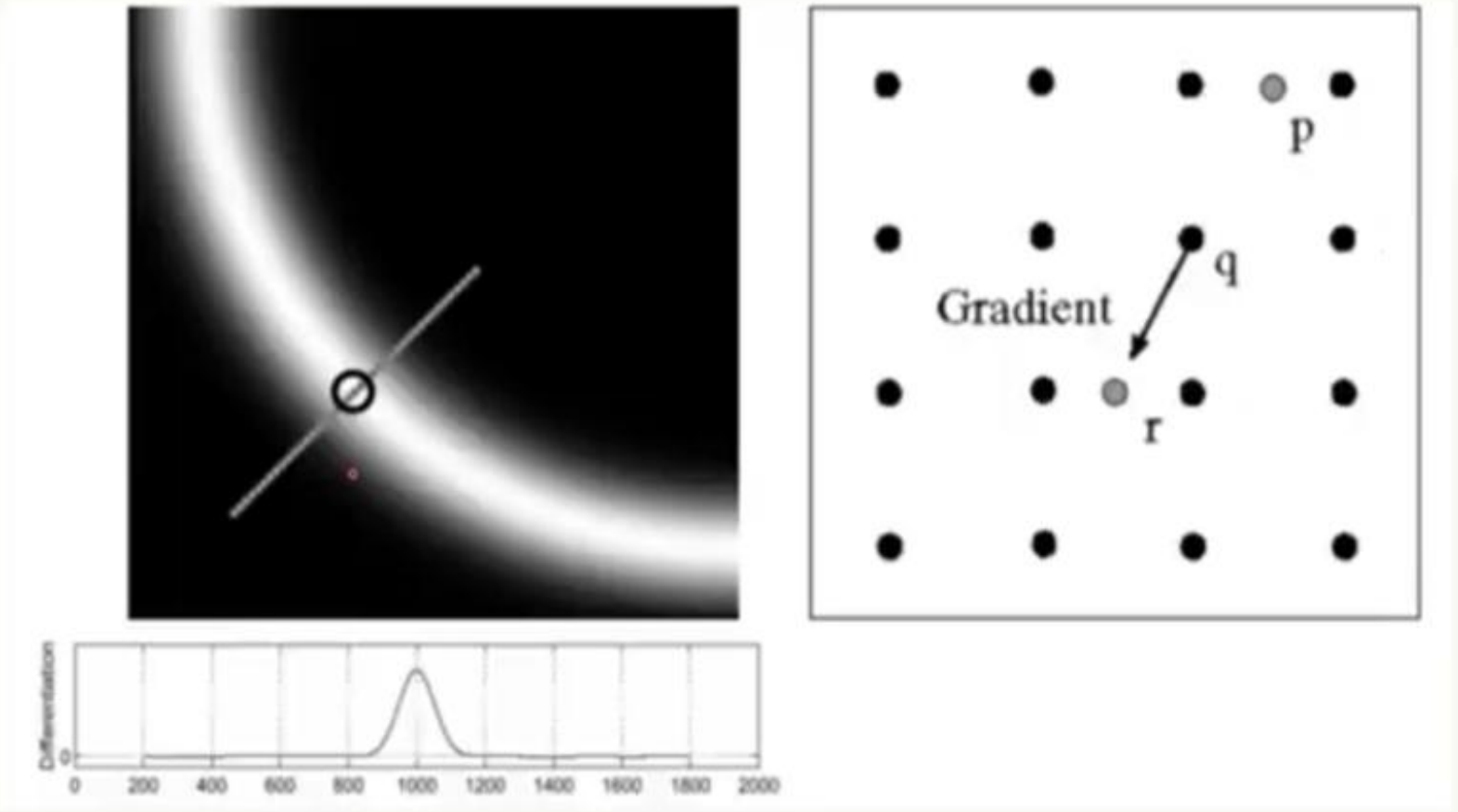
Non-Maximum Suppression

Thresholding

Non-Maximum Suppression

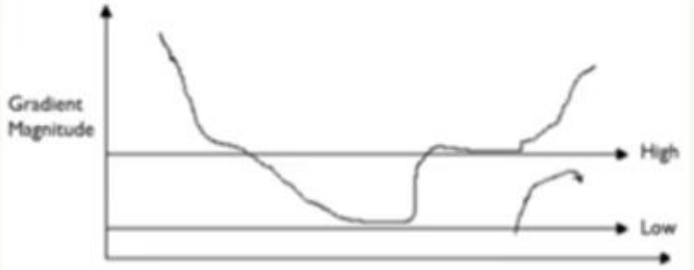
- We will check if the pixel is the local-maximum in the direction of the gradient(might need to interpolate between two pixels)

Non-Maximum Suppression



Results

Hysteresis(Thresholding)



- Threshold the gradient Magnitude(But using two thresholds!)
- Hysteresis : Use high threshold to start edge curves and a low threshold to continue them
- **What's the central idea?**
- If gradient at pixel > 'High' : Edge Pixel(Don't suppress it)
- If gradient at pixel < 'Low' : Not an edge Pixel(suppress it)
- If gradient at pixel > Low and < High : Edge pixel iff connected to an edge pixel directly or via other pixels between 'Low' and 'High'

Smoothing

Images taken from a camera will contain some amount of noise. As noise can mislead the result in finding edges, we have to reduce the noise. Therefore the image is first smoothed by applying a Gaussian filter.

The function 'GaussianFilter' multiplies each pixel in the image by the kernel generated. It returns the smoothed image in a two dimensional array.

$$S = I * g(x,y) \quad = g(x,y) * I$$

Where,

$$g(x,y) = \frac{1}{\sqrt{2\pi}\,\sigma}\; e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{1}{115}$$

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

Figure 3 Discrete approximation to Gaussian function with σ=1.4

Finding Gradients

- Here we will find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image.
- The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction and the other estimating the gradient in the y-direction .

$$\nabla S = \nabla (g * I) \quad = (\nabla g) * I$$

$$\nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I \quad = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix}$$

$$Where, \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

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

Gx

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

Gy

- Now the approximate absolute gradient magnitude (edge strength) at each point can be found as

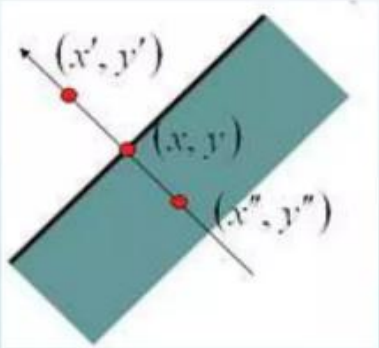
$$G = \sqrt{G_x^2 + G_y^2}$$

and the orientation of the gradient can be found using the gradient in the x and y directions

$$\cdot \quad = tan^{-1}\left(\frac{G_x}{G_y}\right)$$

Non-Maximum Suppression

- This is necessary to convert the blurred edges in the image of the gradient magnitudes to sharp edges. Actually this is performed by considering only all local maxima in the gradient image and deleting everything rest. The algorithm is applied for each pixel in the gradient image.
- Finally, only local maxima have been marked as edges.



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

x′ and x″ are the neighbors of x along normal detection to an edge

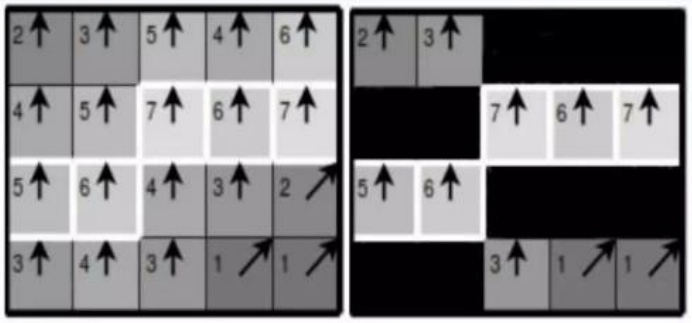


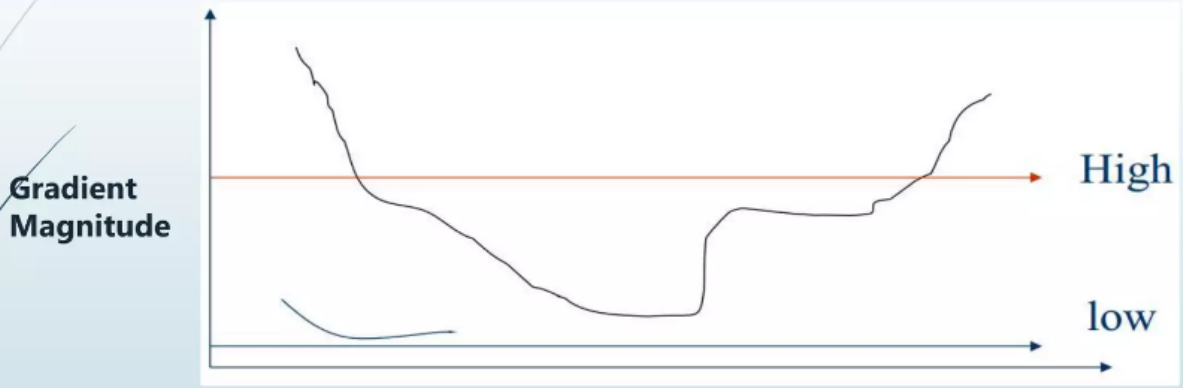
Figure: Example of non-maximum suppression

Hysteresis Thresholding

- After the non-maximum suppression step, the edge pixels are still marked with their strength pixel-by-pixel.
- The received image may still contain false edge points.Using threshold, Potential edges are determined by doble thresholding(High and Low).
- If the gradient at a pixel is
  - above "High", declare it as an 'edge pixel'
  - below "Low", declare it as a "non-edge-pixel"
  - between "low" and "high"

  Consider its neighbors iteratively then declare it an "edge pixel" if it is connected to

  an 'edge pixel' directly or via pixels between "low" and "high"



Hysteresis Thresholding

Performance of Canny Edge Detection Algorithm

- The performance of the Canny algorithm depends heavily on the adjustable parameters, σ, which is the standard deviation for the Gaussian filter, and the threshold values, 'High and Low ".σ also controls the size of the Gaussian filter. The bigger the value for σ, the larger the size of the Gaussian filter becomes. This implies more blurring, necessary for noisy images, as well as detecting larger edges. As expected, however, the larger the scale of the Gaussian, the less accurate is the localization of the edge. Smaller values of σ imply a smaller Gaussian filter which limits the amount of blurring, maintaining finer edges in the image. The user can tailor the algorithm by adjusting these parameters to adapt to different environments.
- Canny's edge detection algorithm is computationally more expensive compared to Sobel, Prewitt and Robert's operator. However, the Canny's edge detection algorithm performs better than all these operators under almost all scenarios.