CAP 4453
Robot Vision

Dr. Gonzalo Vaca-Castaño

gonzalo.vacacastano@ucf.edu

Administrative details

- Homework due Friday 11:59pm via webcourses
- Point 5 – refers to filter center in second row

Robot Vision

6. Edge detection II

Credits

- Some slides comes directly from:
  - Yogesh S Rawat (UCF)
  - Noah Snavely (Cornell)
  - Ioannis (Yannis) Gkioulekas (CMU)
  - Mubarak Shah (UCF)
  - S. Seitz
  - James Tompkin
  - Ulas Bagci
  - L. Lazebnik

Short Review from last class

Edge detectors

- Gradient operators
  - Prewit
  - Sobel
- Marr-Hildreth (Laplacian of Gaussian)
- Canny (Gradient of Gaussian)

Gradient operators edge detector algorithm

1. Compute derivatives
    - In x and y directions
    - Use Sobel or Prewitt filters
2. Find gradient magnitude
3. Threshold gradient magnitude

Sobel

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

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

Prewitt

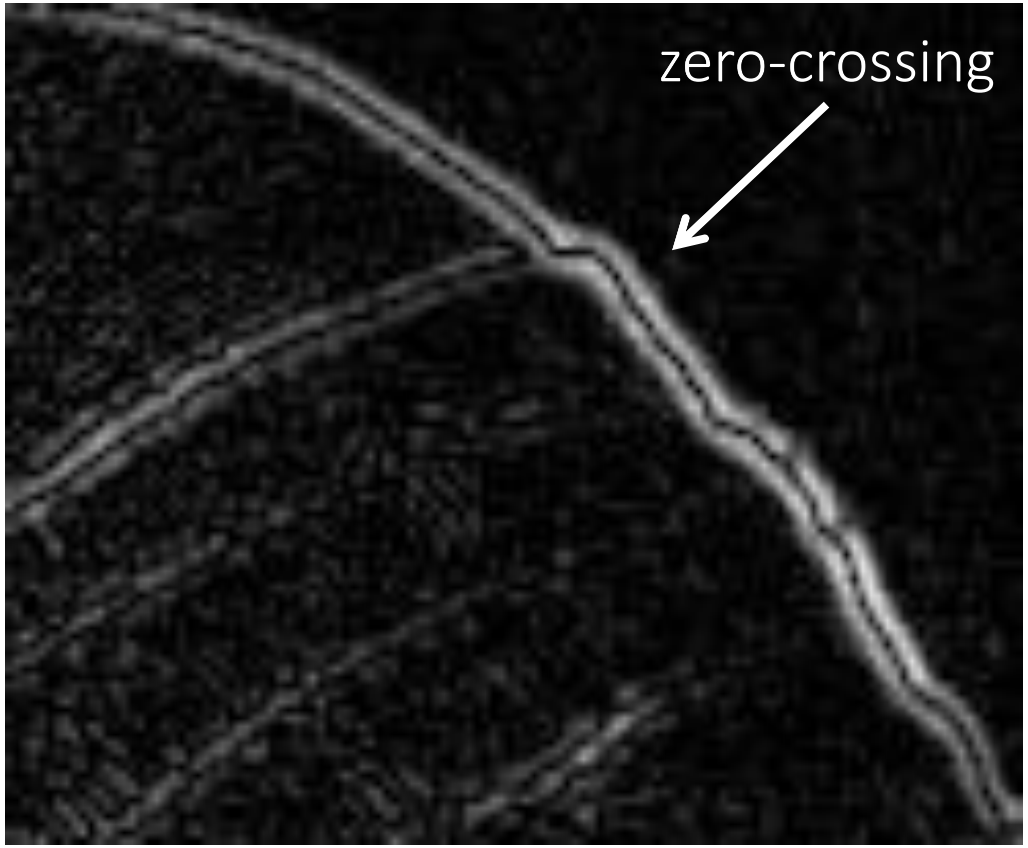
1	0	-1
1	0	-1
1	0	-1

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

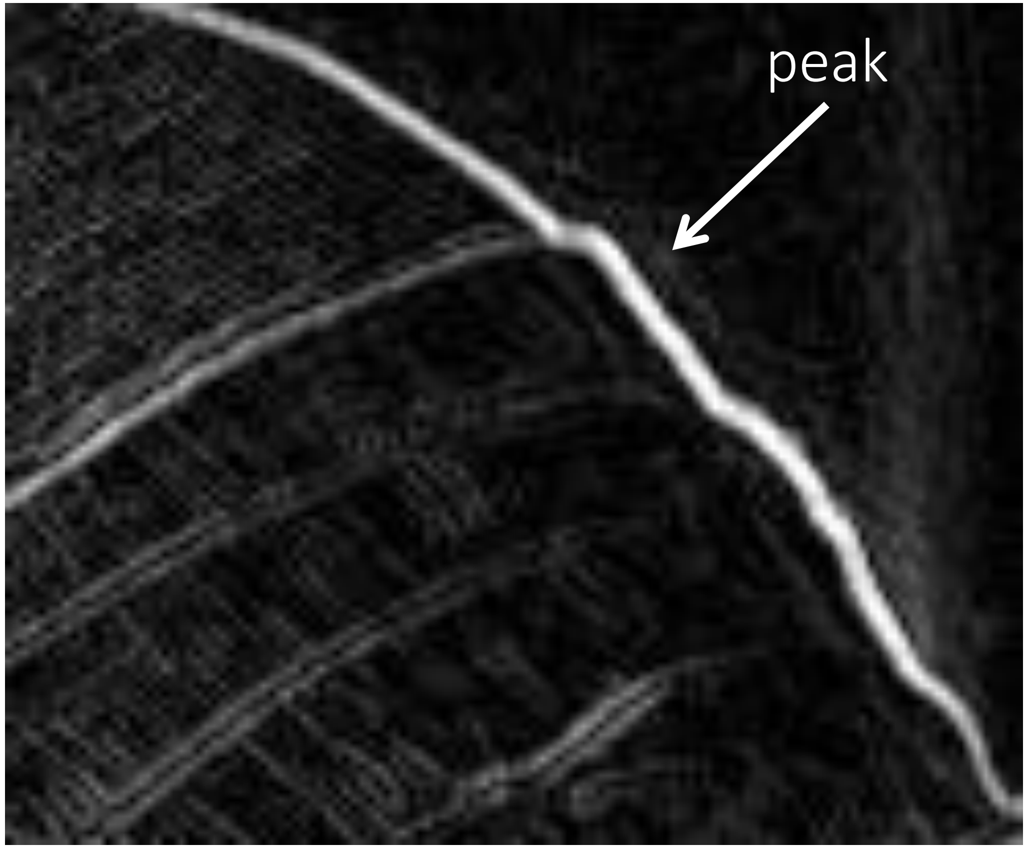
Marr-Hildreth edge detector algorithm

1. Smooth image by Gaussian filtering
2. Apply Laplacian to smoothed image
   - Used in mechanics, electromagnetics, wave theory, quantum mechanics
3. **Find Zero crossings**
   - Scan along each row, record an edge point at the location of zero-crossing.
   - Repeat above step along each column

Laplacian of Gaussian vs Derivative of Gaussian



Laplacian of Gaussian filtering

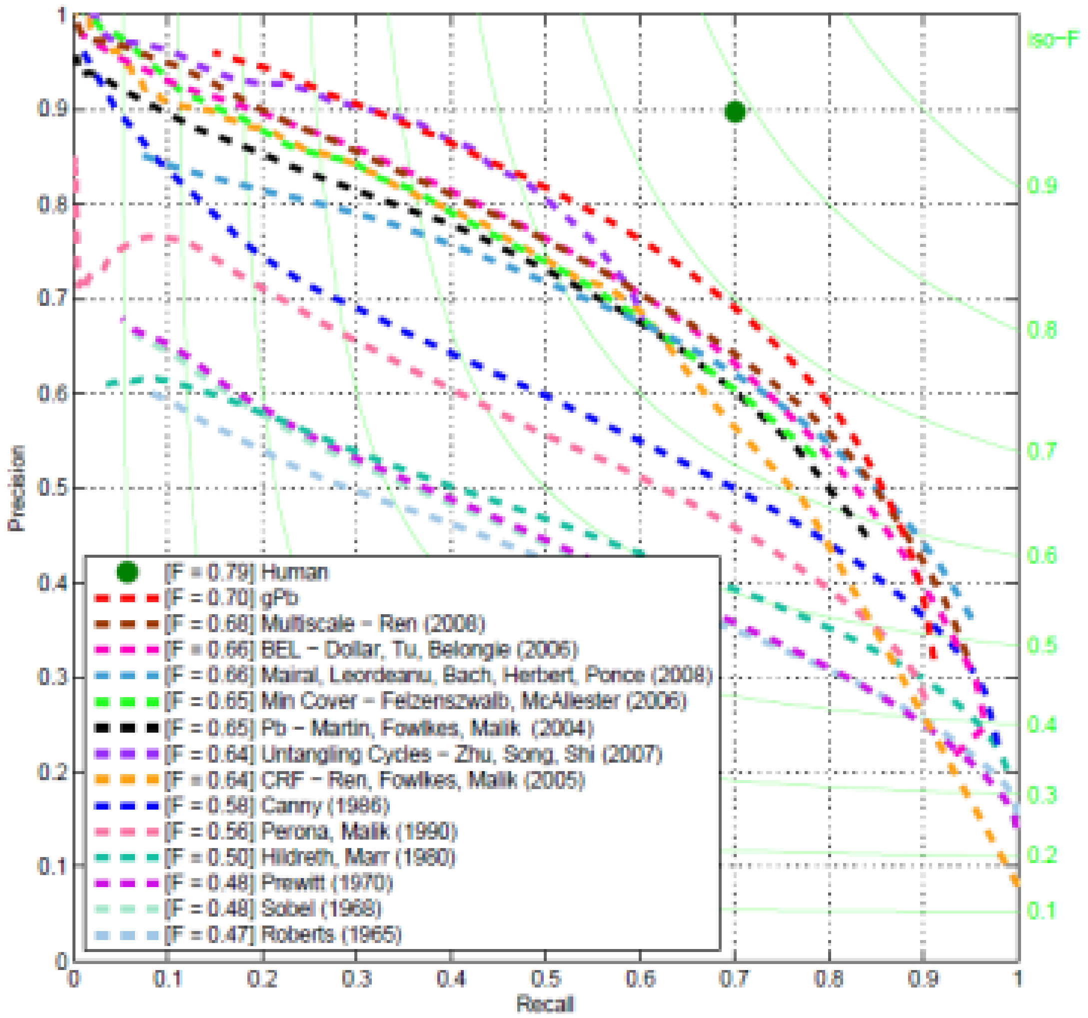


Derivative of Gaussian filtering

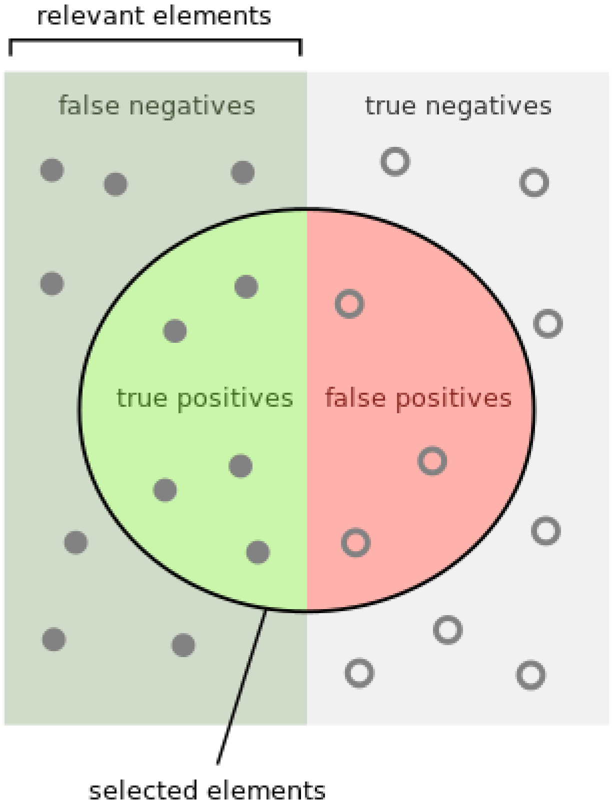
Zero crossings are more accurate at localizing edges (but not very convenient).

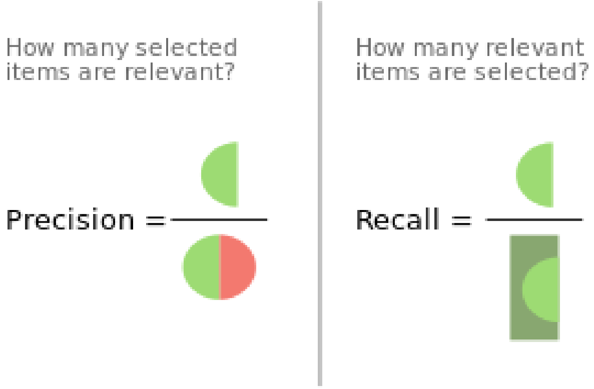
45 years of boundary detection

[Pre deep learning]

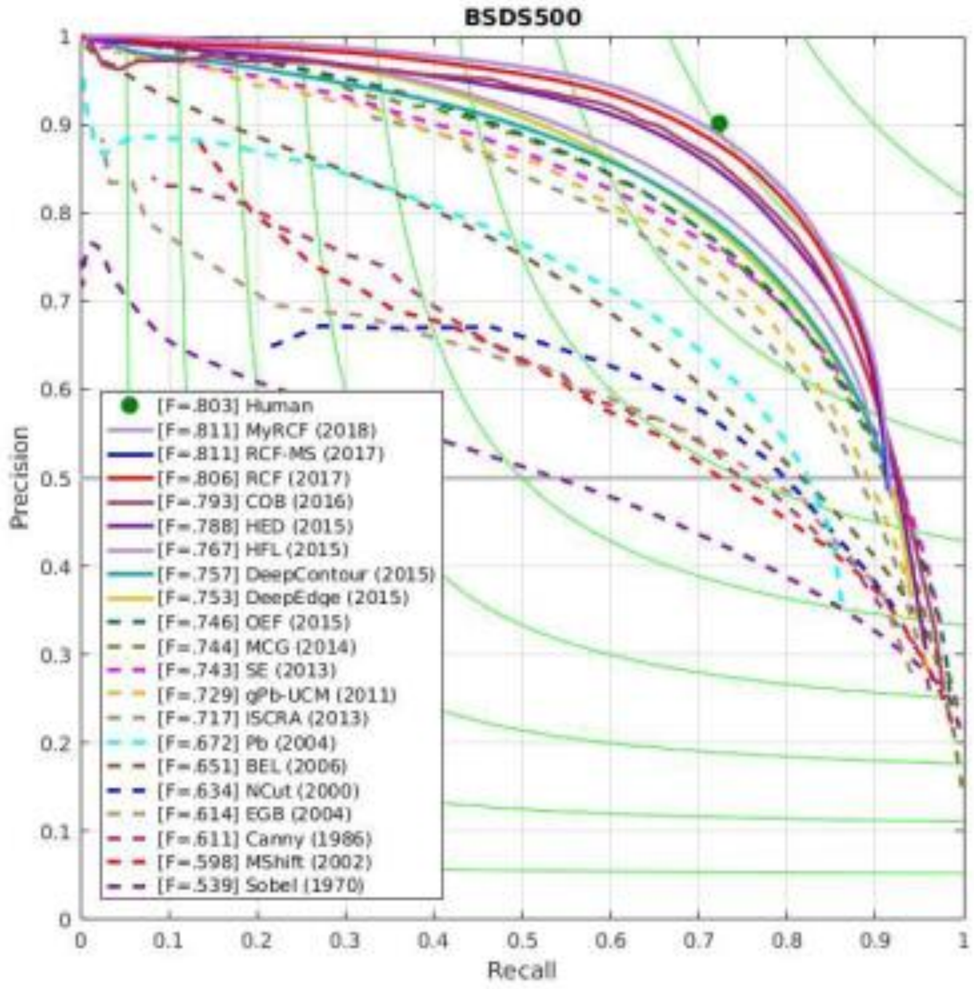


Precision Recall





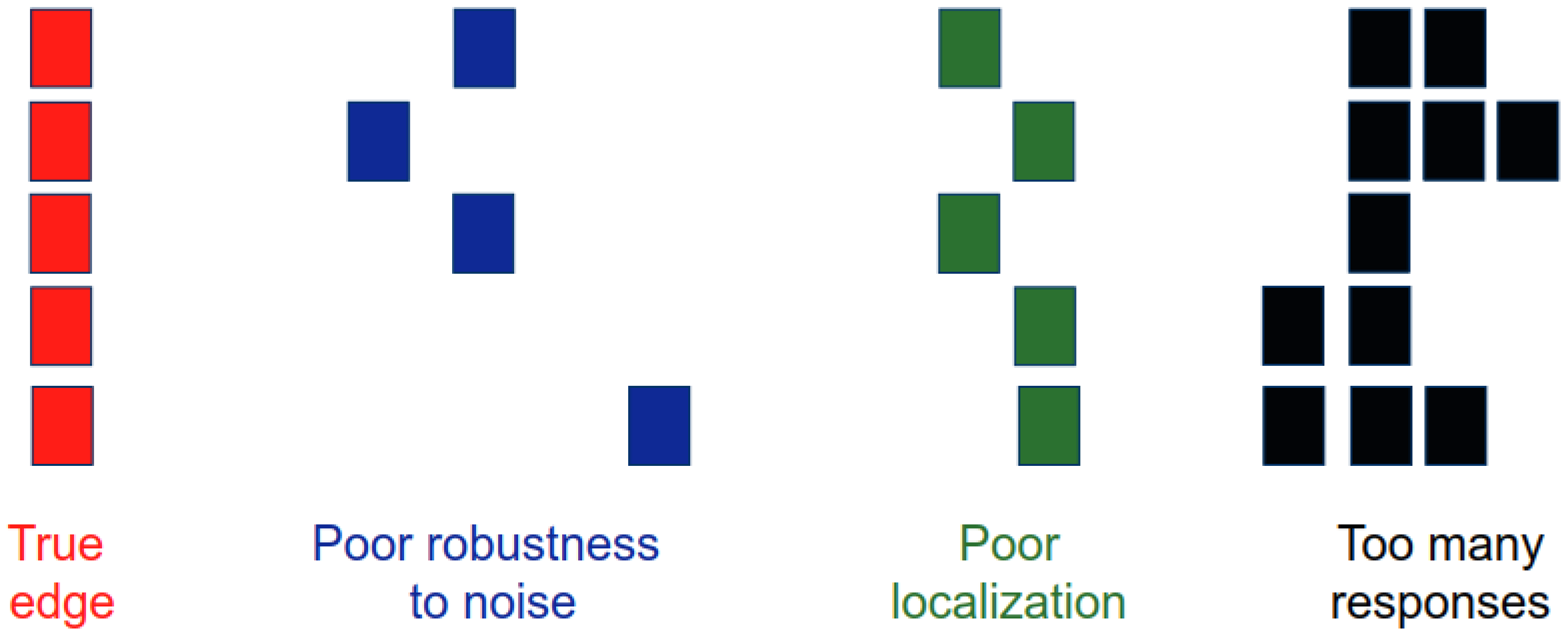
Edge Detection with Deep Learning



Canny edge detector

Design Criteria for Edge Detection

- Good detection: find all real edges, ignoring noise or other artifacts
- Good localization
  - as close as possible to the true edges
  - one point only for each true edge point



Canny Edge detector algorithm

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

Canny Edge detector algorithm

1. **Smooth image with Gaussian filter**

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

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

2. **Compute derivative of filtered ·**

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

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

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

Canny Edge detector algorithm

1. **Smooth image with Gaussian filter**

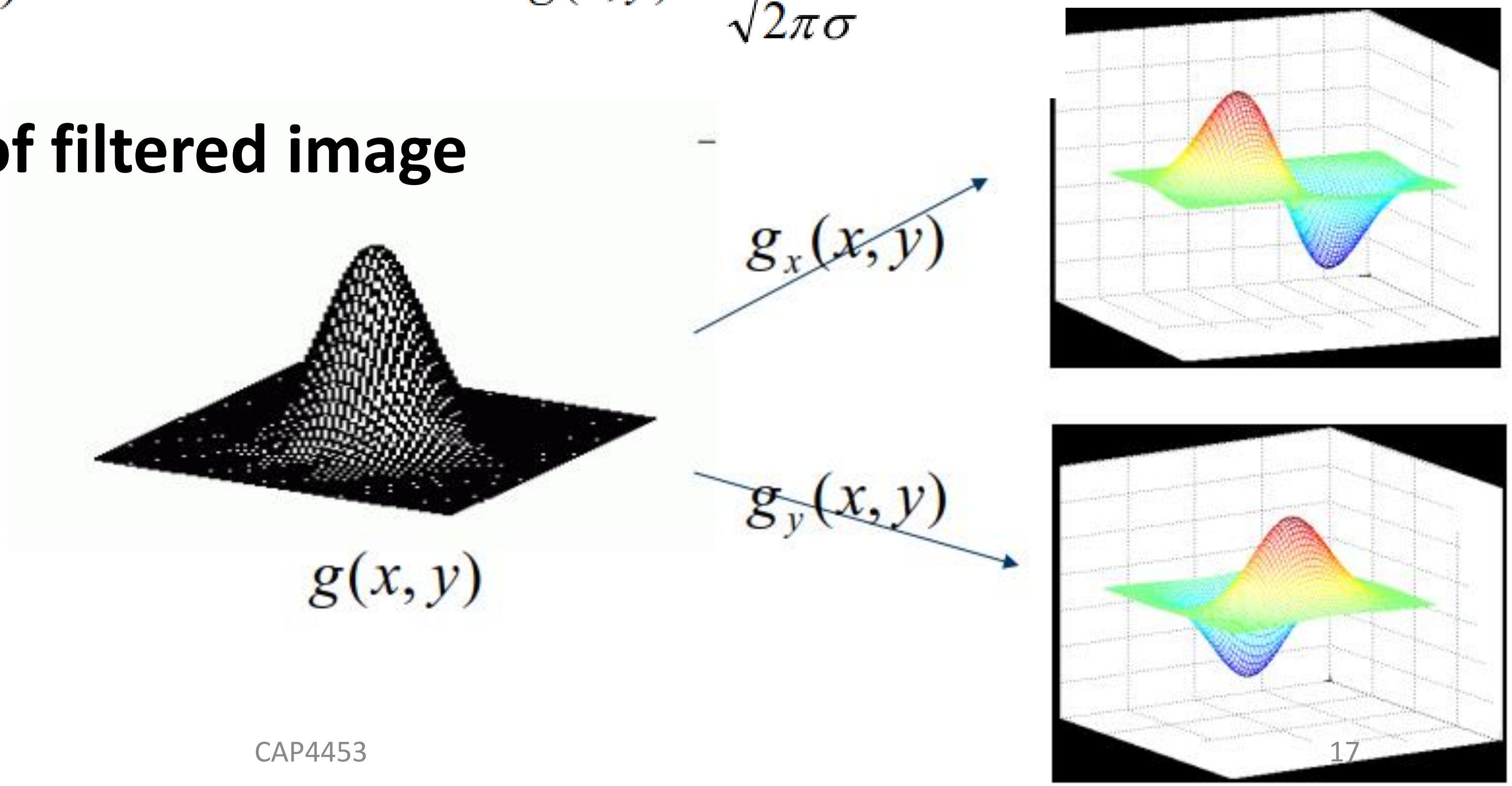
$$S = I * g(x,y) = g(x,y) * I$$

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

2. **Compute derivative of filtered image**

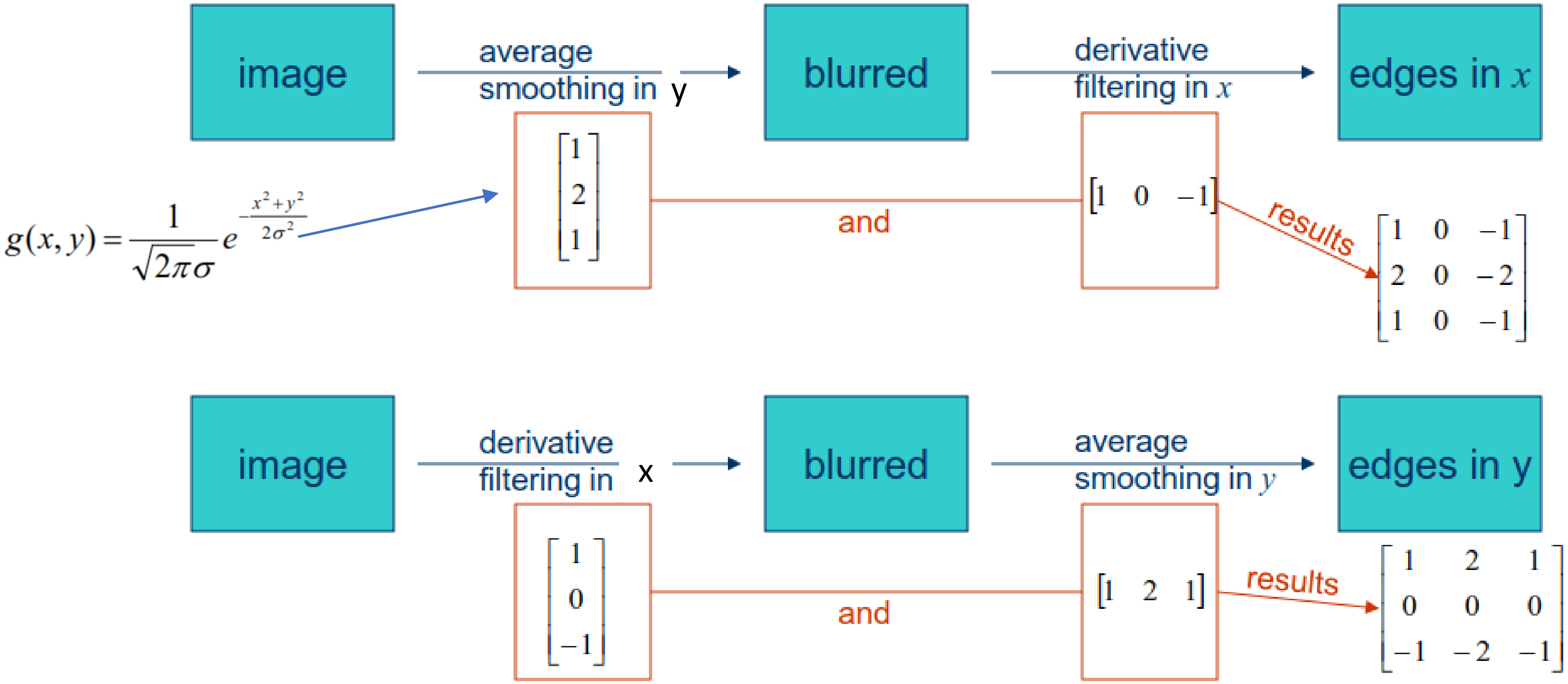
$$\nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix}$$

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



Canny Edge detector algorithm

1. **Smooth image with Gaussian filter**
2. **Compute derivative of filtered image**

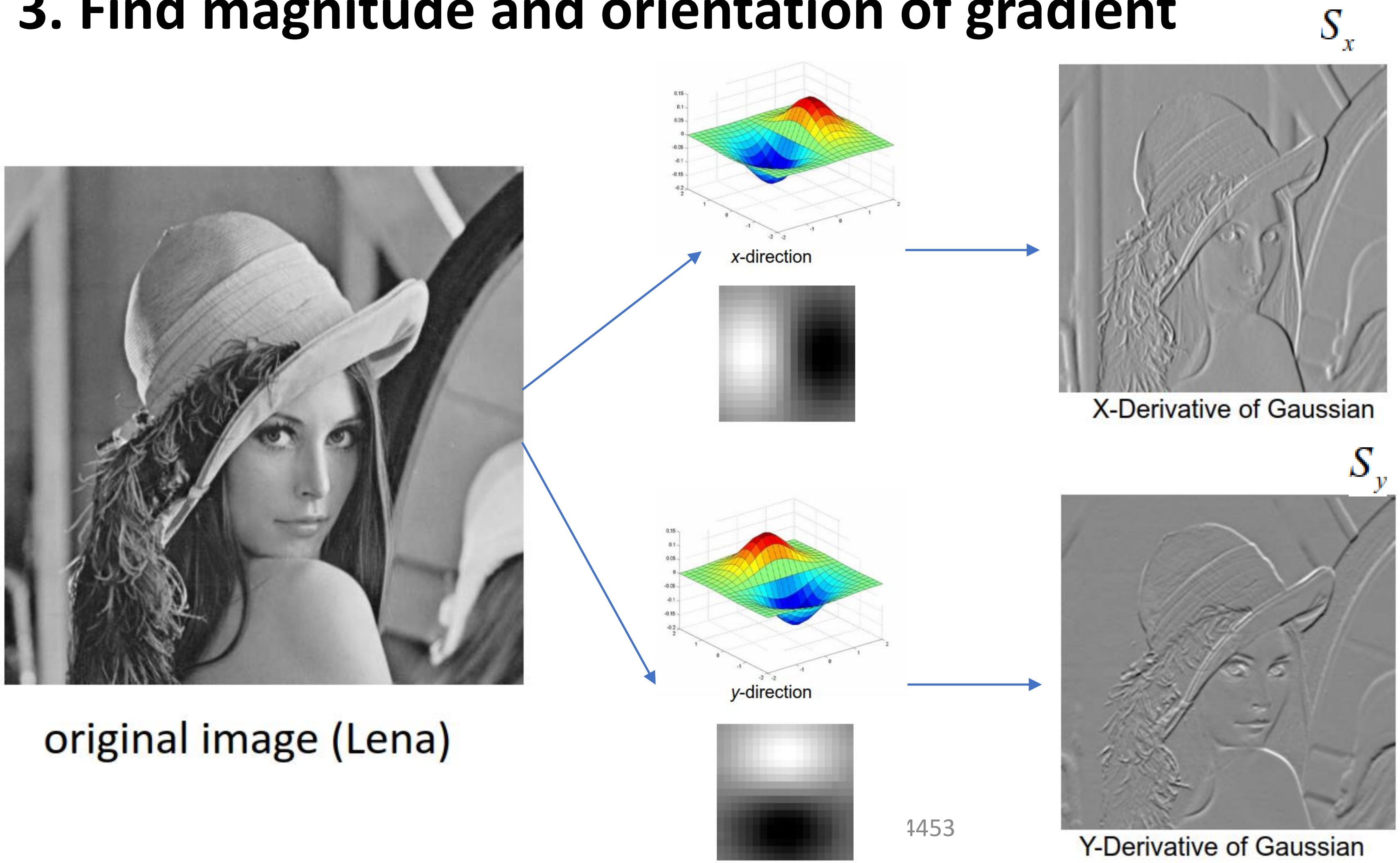


Canny Edge detector algorithm

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

Canny Edge detector algorithm

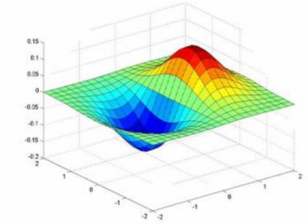
3. Find magnitude and orientation of gradient



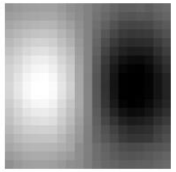
Canny Edge detector algorithm

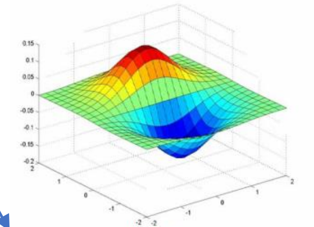
3. Find magnitude and orientation of gradient

original image (Lena)

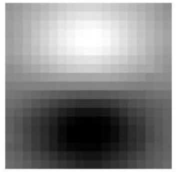


x-direction





y-direction



S_x

X-Derivative of Gaussian

S_y

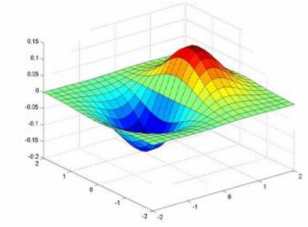
Y-Derivative of Gaussian

magnitude $= \sqrt{(S_x^2 + S_y^2)}$

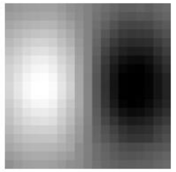
Canny Edge detector algorithm

3. Find magnitude and orientation of gradient

original image (Lena)

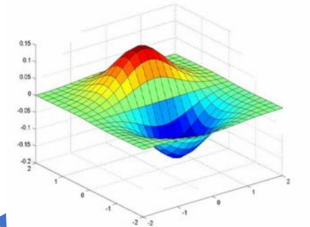


x-direction

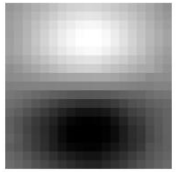


S_x

X-Derivative of Gaussian

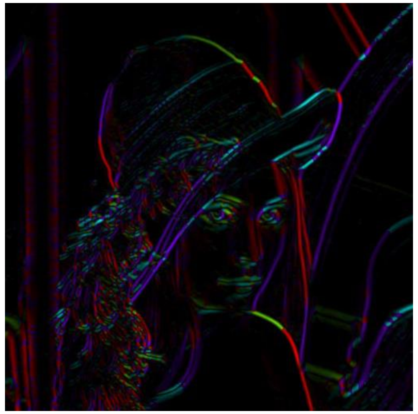


y-direction



S_y

Y-Derivative of Gaussian



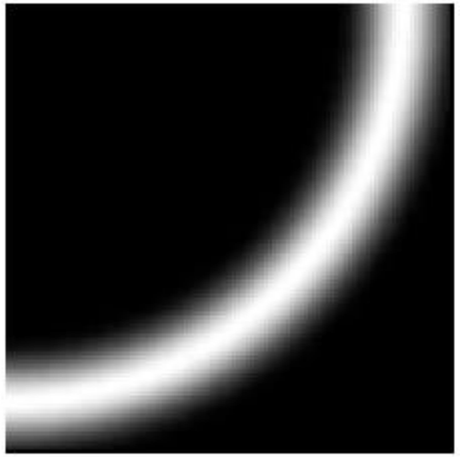
$$Theta = \text{atan2}(S_y, S_x)$$

Canny Edge detector algorithm

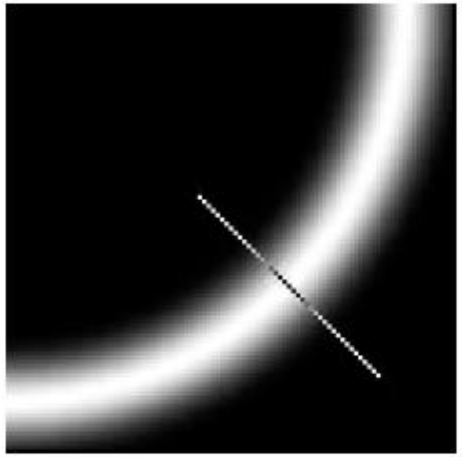
1. Smooth image with Gaussian filter
2. Compute derivative of filtered image
3. Find magnitude and orientation of gradient
4. **Apply “Non-maximum Suppression”**
5. Apply “Hysteresis Threshold”

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**



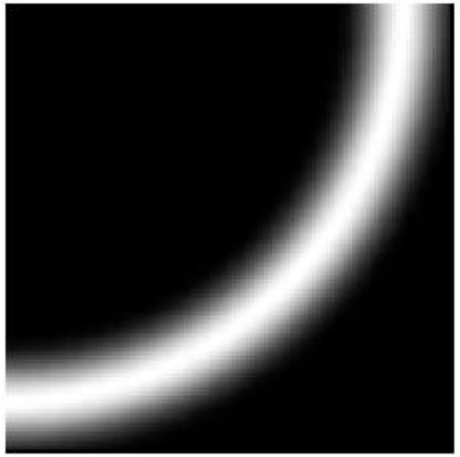
Goal: keep pixels along the curve where magnitude is largest



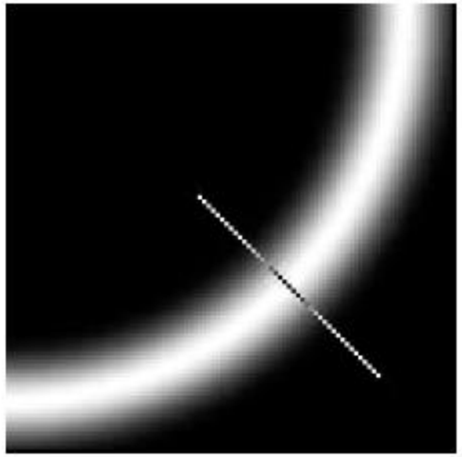
How to: looking for a maximum along a slice normal to the curve

Canny Edge detector algorithm

4. **Apply “Non-maximum Suppression”**



Goal: keep pixels along the curve
where magnitude is largest



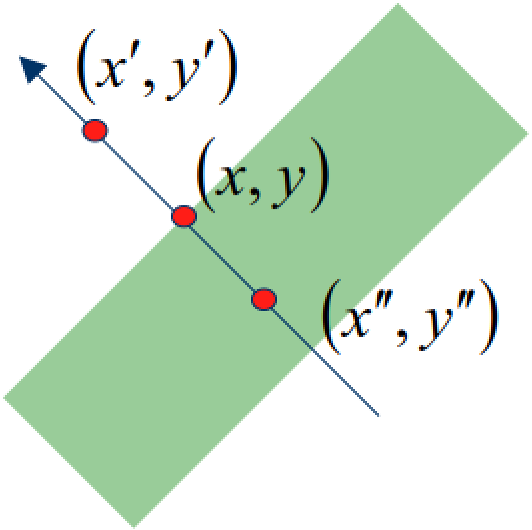
That is the direction of the gradient !

How to: looking for a maximum along a
slice <u>normal to the curve</u>

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

- Suppress the pixels in $|\nabla S|$ which are not local maximum



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

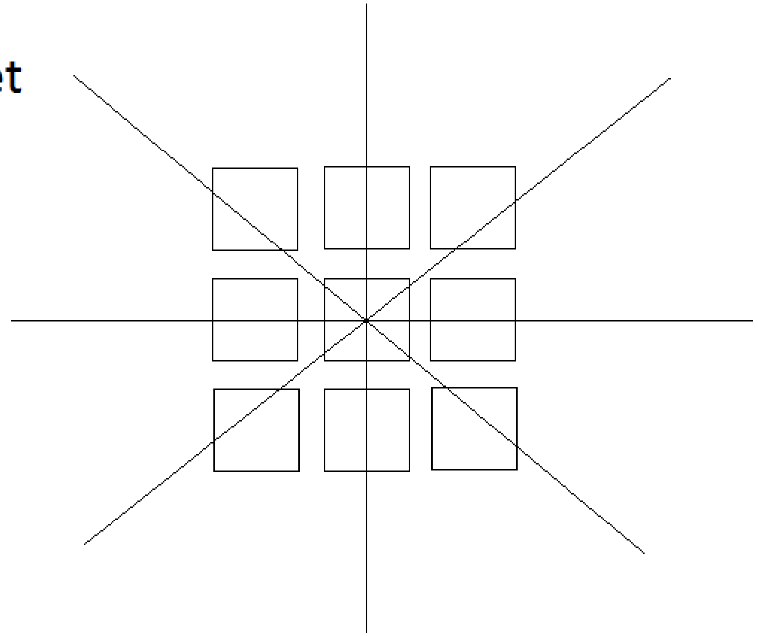
$\mathbf{x}'$ and $\mathbf{x}''$ are the neighbors of $\mathbf{x}$ along normal direction to an edge

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

Consider a 3x3 set

of pixels.



We need to examine triples

along the directions shown

to see if the center pixel is a

peak (in magnitude) compared

to its two neighbors.

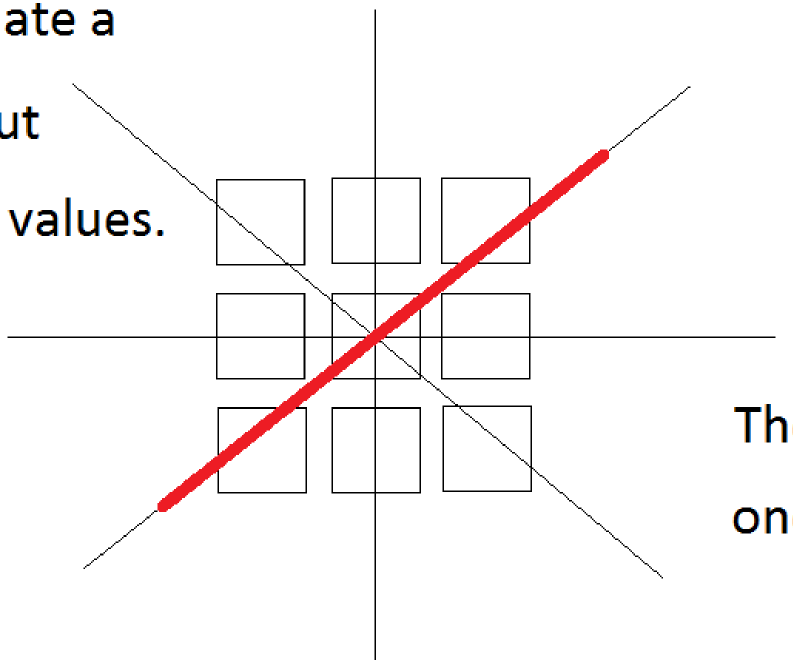
Canny Edge detector algorithm

4. **Apply “Non-maximum Suppression”**

So, we need to isolate a

triple, and ask about

the three adjacent values.



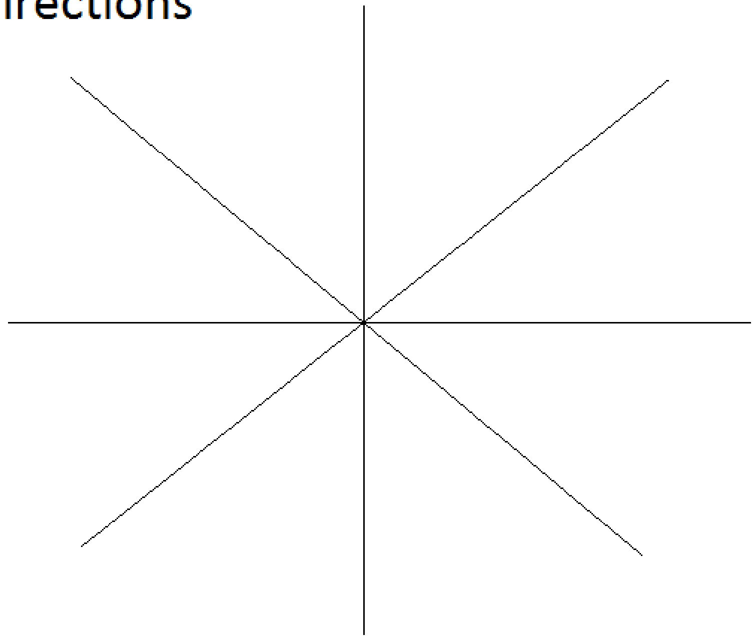
There are 4 such directions,

one triple per direction.

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

So, consider the 4 directions



For each direction, we will ask whether the center mag value exceeds the mag value of neighbor on one side, and neighbor on other side; if yes, mark as peak.

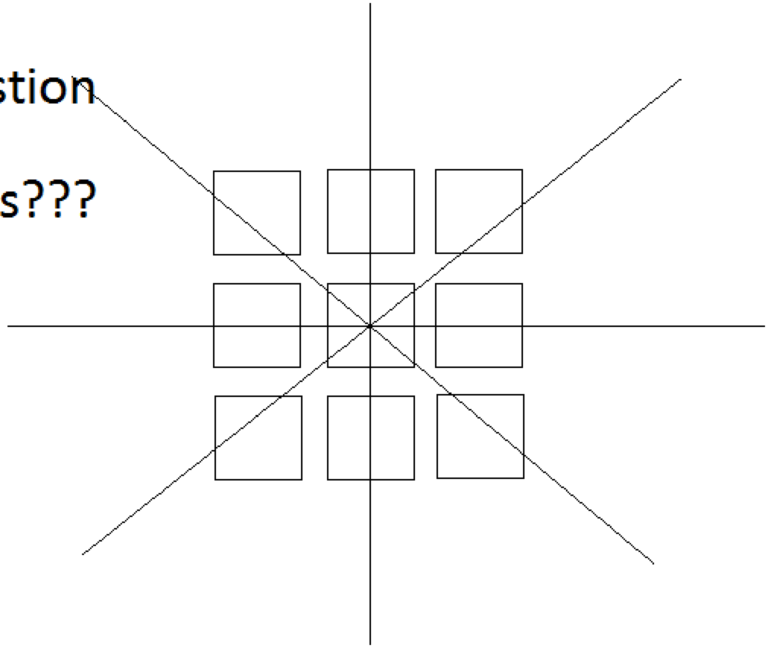
Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

For a center pixel, do we

ask the maxing question

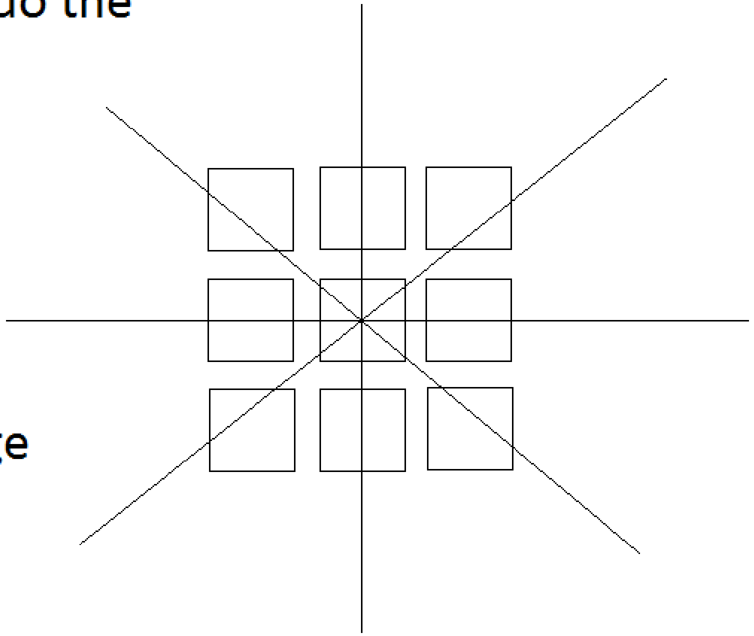
for all four directions???



Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

No, we should only do the maxing test for the direction that is dictated by the direction of the edge

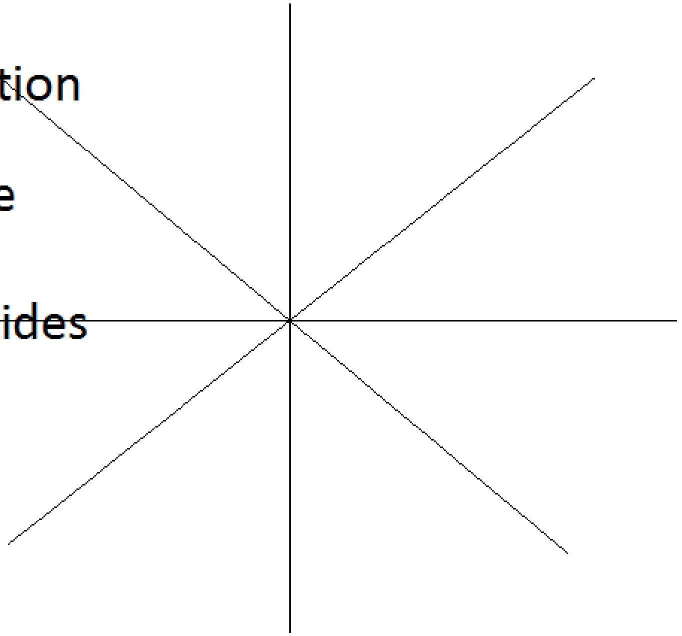


We always want to max-test in the direction perpendicular to the edge, i.e., across the edge. This means in the direction of the gradient vector.

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

So, given these 4 possible directions, each direction will be called up, if the gradient vector coincides with the max-test direction.



Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

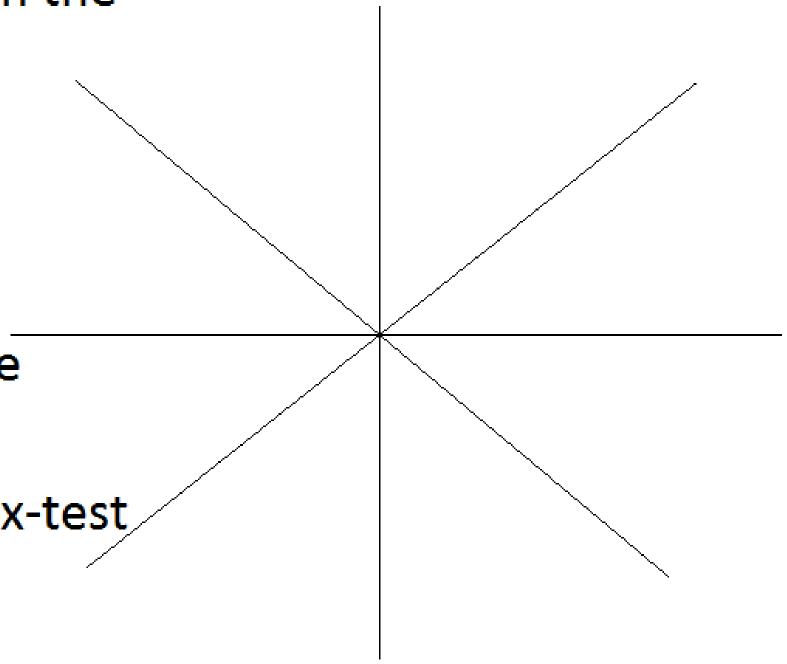
Hence, we ascertain the

gradient vector's

direction, and then

depending on it, we

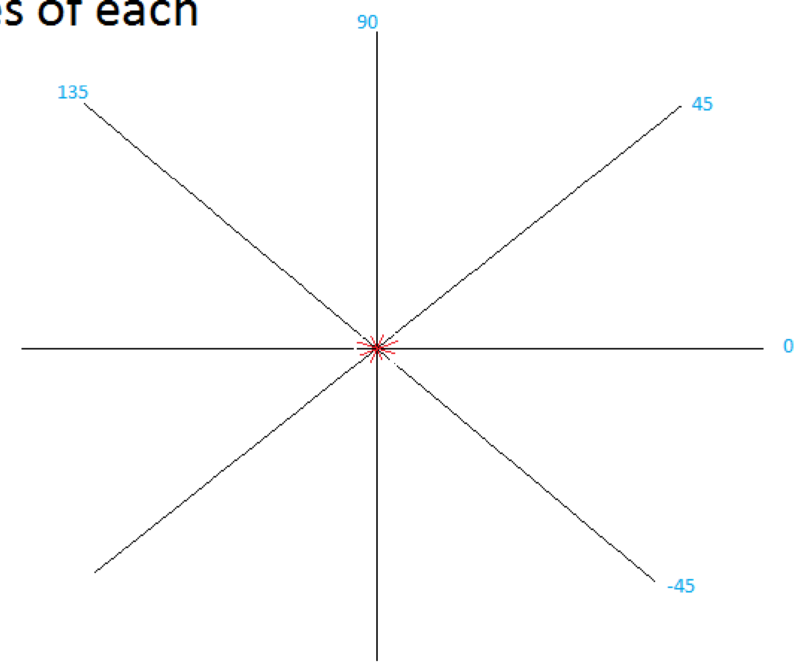
proceed to the max-test



Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

We know the angles of each direction.



Canny Edge detector algorithm

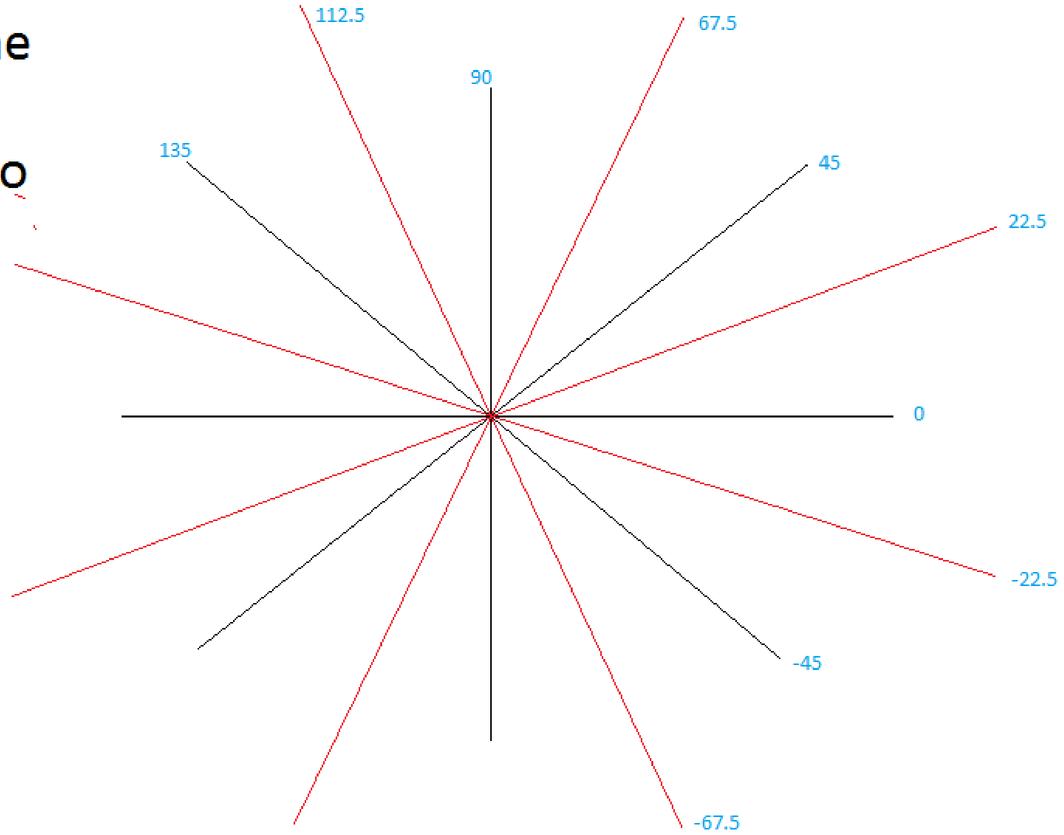
4. **Apply "Non-maximum Suppression"**

We break up the

angle space into

4 cones of

directions



Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

For example, if it is found

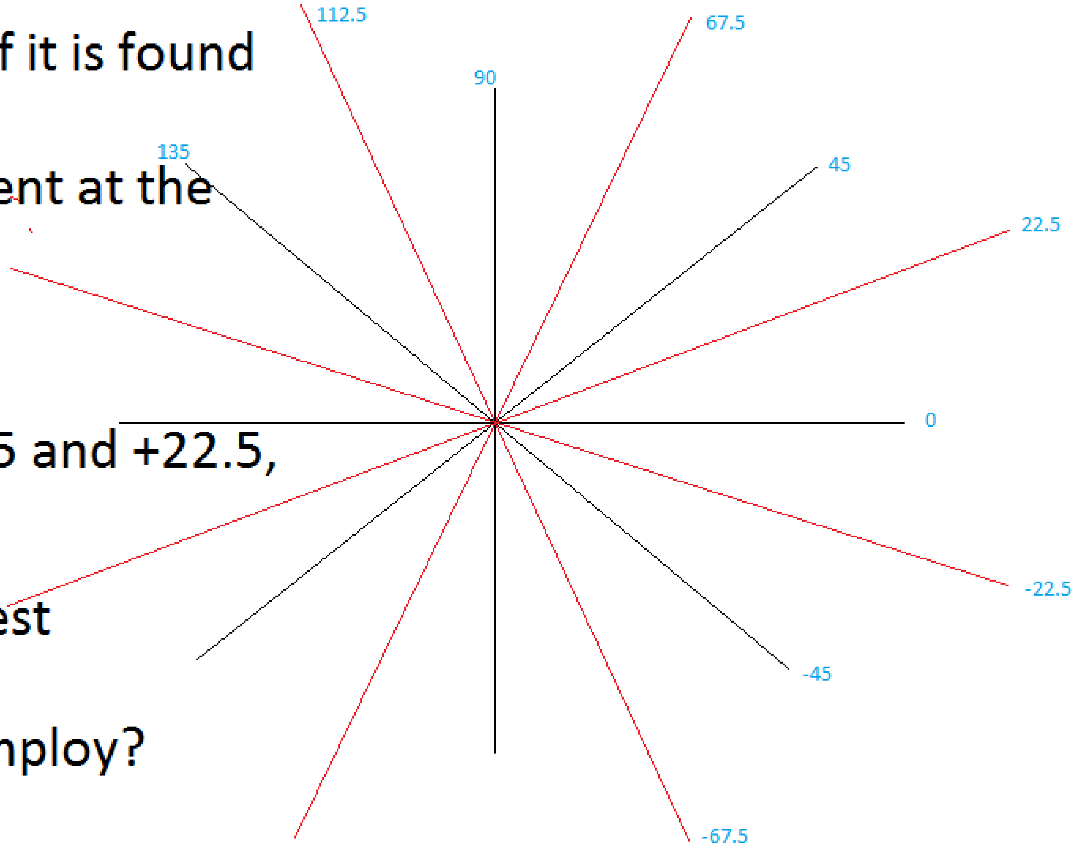
that the gradient at the

center pixel is

between -22.5 and +22.5,

Which max-test

should we employ?



Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

So, put = sign in

And, remove Vertical

cone test, use "otherwise"

For each pixel (i.e., double-for loop)

    Get pixel's gradient direction , Dir

    If - 22.5 < Dir <=22.5

        Employ Horizontal Max-Test

    else if +22.5 < Dir <=+67.5

        Employ Test involving SW and NE pixels

    else if -67.5 < Dir <=-22.5

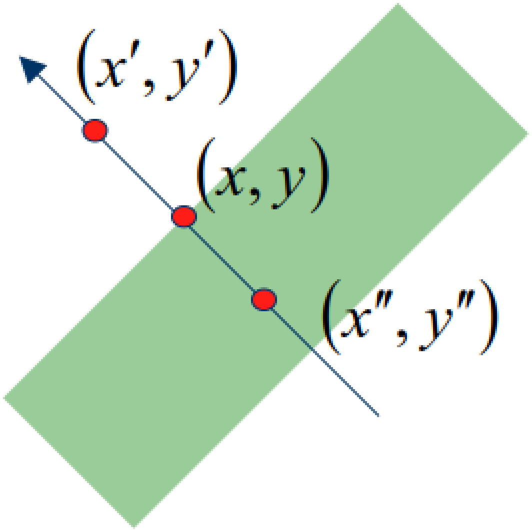
        Employ Test involving SE and NW pixels

    else Employ Vertical test

Canny Edge detector algorithm

4. **Apply "Non-maximum Suppression"**

- Suppress the pixels in $|\nabla S|$ which are not local maximum



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

x' and **x**'' are the neighbors of **x** along normal direction to an edge

Canny Edge detector algorithm

4. **Apply “Non-maximum Suppression”**

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

M

For visualization

$M \geq Threshold = 25$

Canny Edge detector algorithm

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

Canny Edge detector algorithm

5. Apply "Hysteresis Threshold"

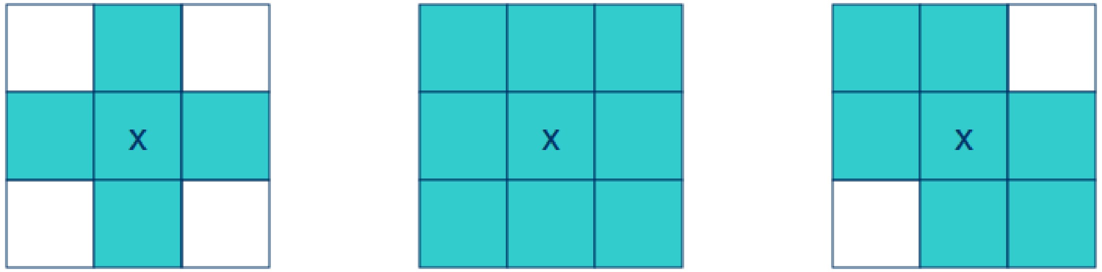
- If the gradient at a pixel is
  - above "**High**", declare it as an '**edge pixel**'
  - below "**Low**", declare it as a "**non-edge-pixel**"
  - **between** "low" and "high"
    - Consider its neighbors iteratively then declare it an "edge pixel" if it is **connected** to an 'edge pixel' **directly** or via pixels **between** "low" and "high".

Canny Edge detector algorithm

5. Apply "Hysteresis Threshold"

- 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".

- Connectedness



4 connected | 8 connected | 6 connected

Canny Edge detector algorithm

5. Apply "Hysteresis Threshold"

- Scan the image from left to right, top-bottom.
  - The gradient magnitude at a pixel is above a high threshold declare that as an edge point
  - Then recursively consider the *neighbors* of this pixel.
    - If the gradient magnitude is above the low threshold declare that as an edge pixel.

Canny Edge detector algorithm

5. Apply “Hysteresis Threshold”

M

regular

$M \geq 25$

Hysteresis

$High = 35$

$Low = 15$

Canny Edge detector algorithm

Before non-max suppression

After non-max suppression

Canny Edge detector algorithm

Threshold at low/high levels to get weak/strong edge pixels

- Do connected components, starting from strong edge pixels

Final Canny Edge

Canny Edge detector algorithm

Final Canny Edge

Effect of σ (Gaussian kernel spread/size)

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

Source: S. Seitz

Questions?