Edge Detection

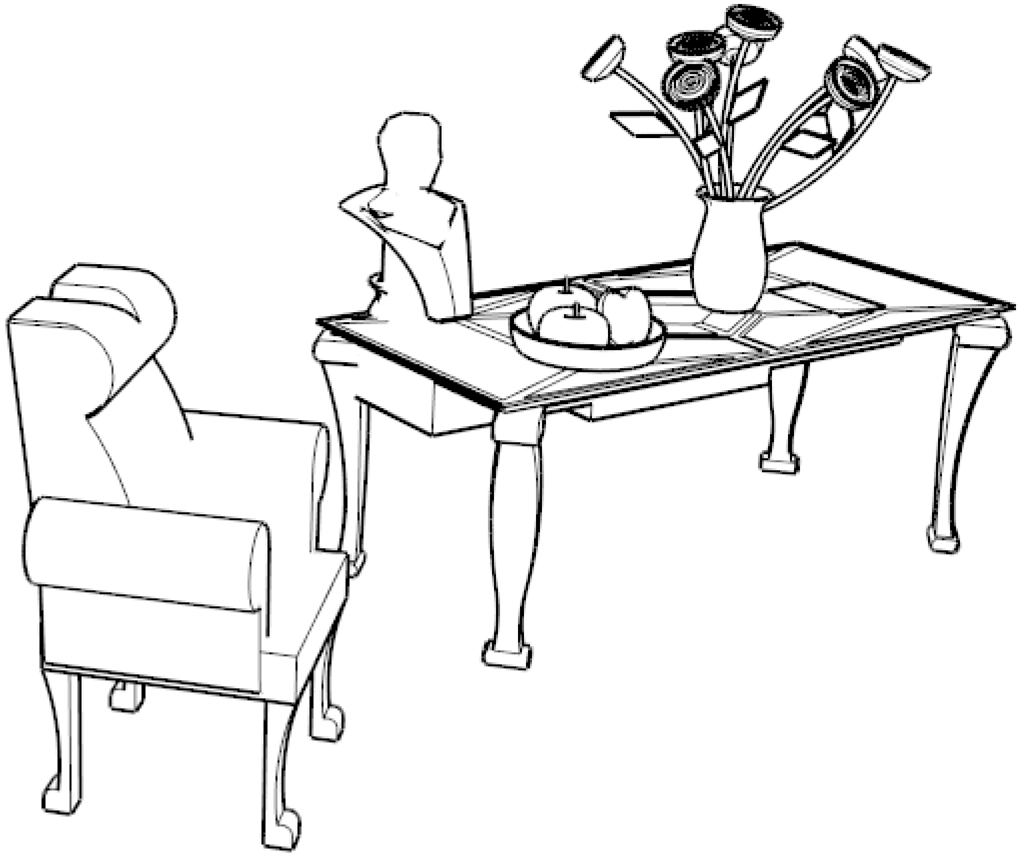
What is an Edge?

- An edge is a location of rapid intensity variation
- The often mark boundaries of objects, occlusion contours, shadow boundaries or surface contours
- Edges are very important in human perception

Find its edges?

- 

-



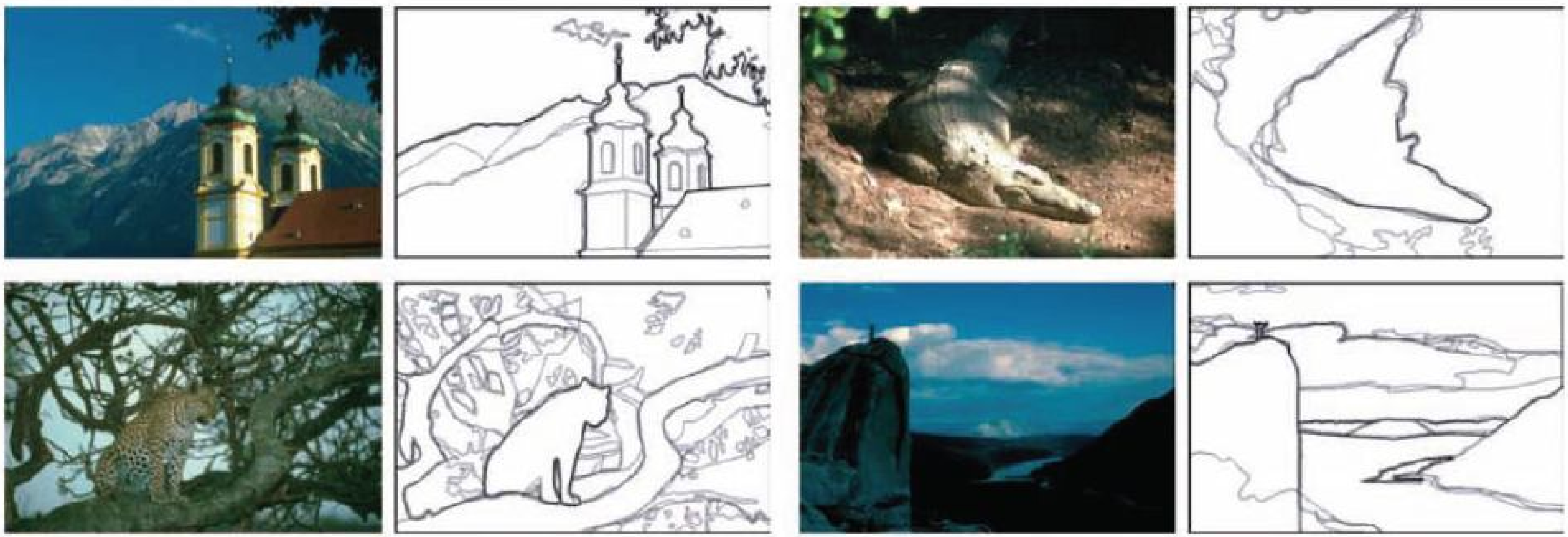


Figure 4.31 Human boundary detection (Martin, Fowlkes, and Malik 2004) © 2004 IEEE. The darkness of the edges corresponds to how many human subjects marked an object boundary at that location.

Applying Masks to Images

- Convolution Operation
- Mask
  - Set of pixel positions and weights
  - Origin of mask

1	1	1
1	**1**	1
1	1	1

1	2	1
2	**4**	2
1	2	1

1
1
1
1
1

Applying Masks to Images

- $I_1 \otimes \text{mask} = I_2$
- Convention: I_2 is the same size as I_1
- Mask Application:
  - For each pixel
  - Place mask origin on top of pixel
  - Multiply each weight with pixel under it
  - Sum the result and put in location of the pixel

Applying Masks to Images

40	40	40	80	80	80
40	40	40	80	80	80
40	1/9	1/9	1/9	80	80
40	1/9	1/9	1/9	80	80
40	1/9	1/9	1/9	80	80
40	40	40	80	80	80

1/9 x

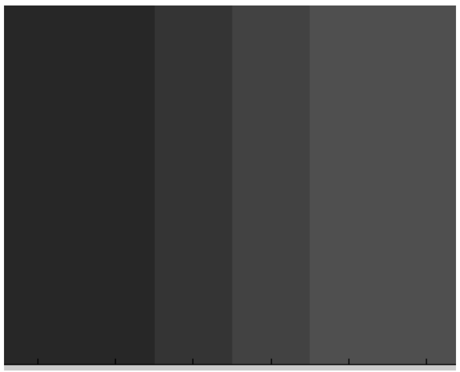
1	1	1
1	1	1
1	1	1

$$6*(1/9*40)+3*(1/9*80) = 53$$

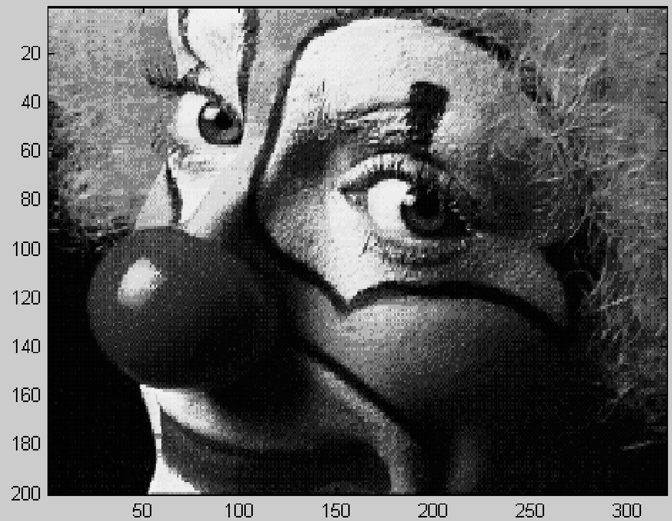
Applying Masks to Images

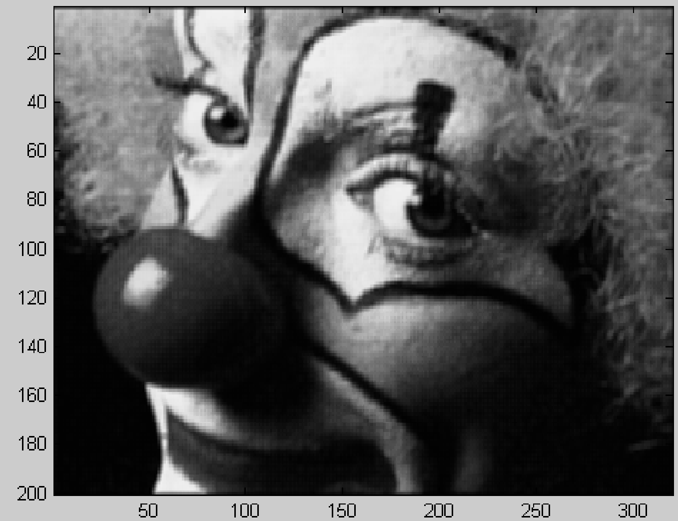
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80

- Overall effect of this mask?
- Smoothing filter

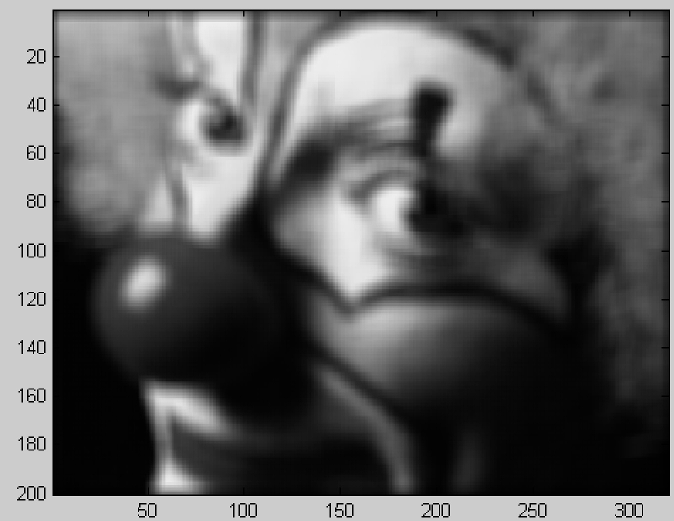


Result of applying Smoothing Mask





40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80
40	40	53	67	80	80

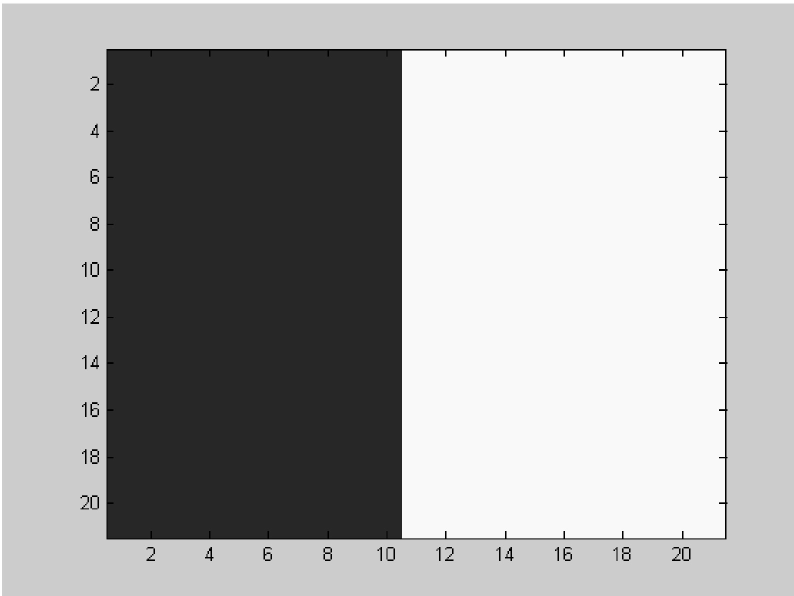


What about corner pixels

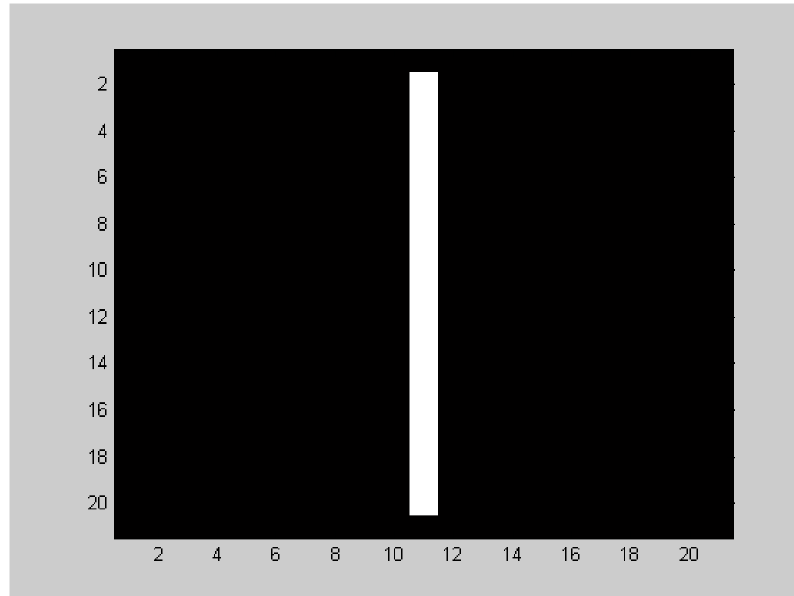
- Expand image with virtual pixels
- Options
  - Fill with a particular value, e.g. zeros
  - Fill with nearest pixel value
- Or just ignore them

Edge Detection

Input



Output



-1	0	1
-1	**0**	1
-1	0	1

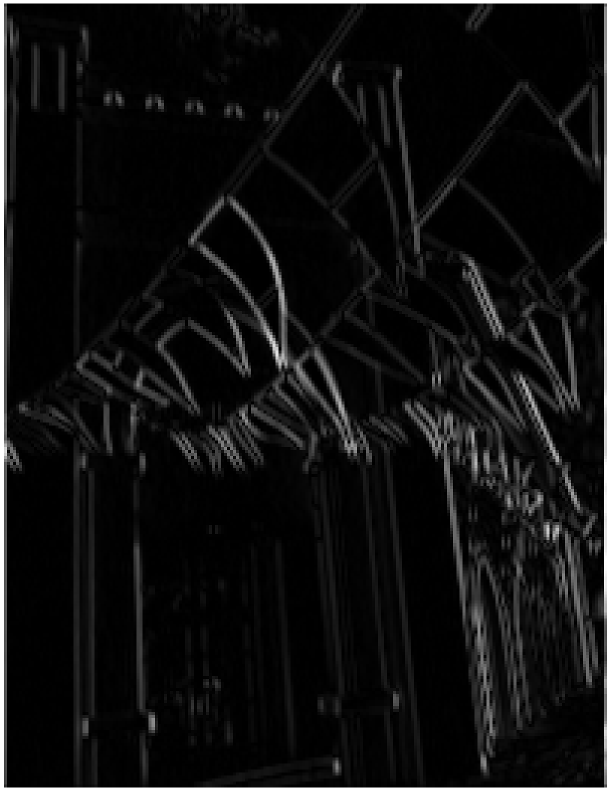
- Should give a zero on smooth output
- Should give a high value on non-smooth regions

How this mask would do that?

Original Image, I

$$e = I \otimes \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

abs(e)



Finding Edges

- Edges are locations where intensity variation is high
- OR rate of change of intensity is high
- How do we find rate of change of intensity?
- DIFFERENTIATION

$$f' = \frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

$$f' = \frac{df}{dx} = f(x) - f(x-1)$$

- Continuous form
- Discrete form

Discrete Derivatives

$f(x)$

40	40	40	40	40	80	80	80	80	80

$f'(x)$

0	0	0	0	0	40	0	0	0	0

If $g(x)$

-1	1

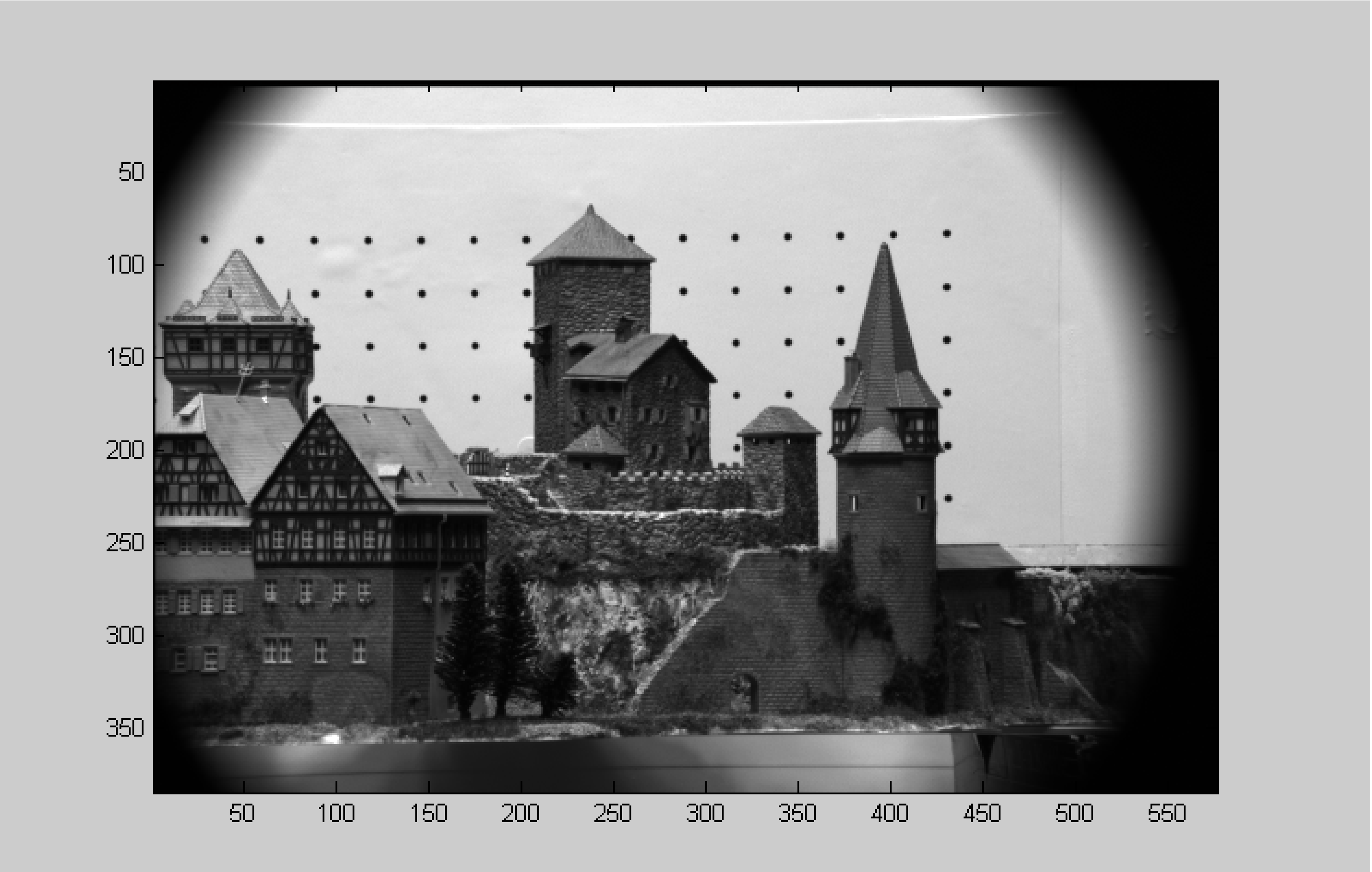
origin

then $f'(x) = f(x) \otimes g(x)$

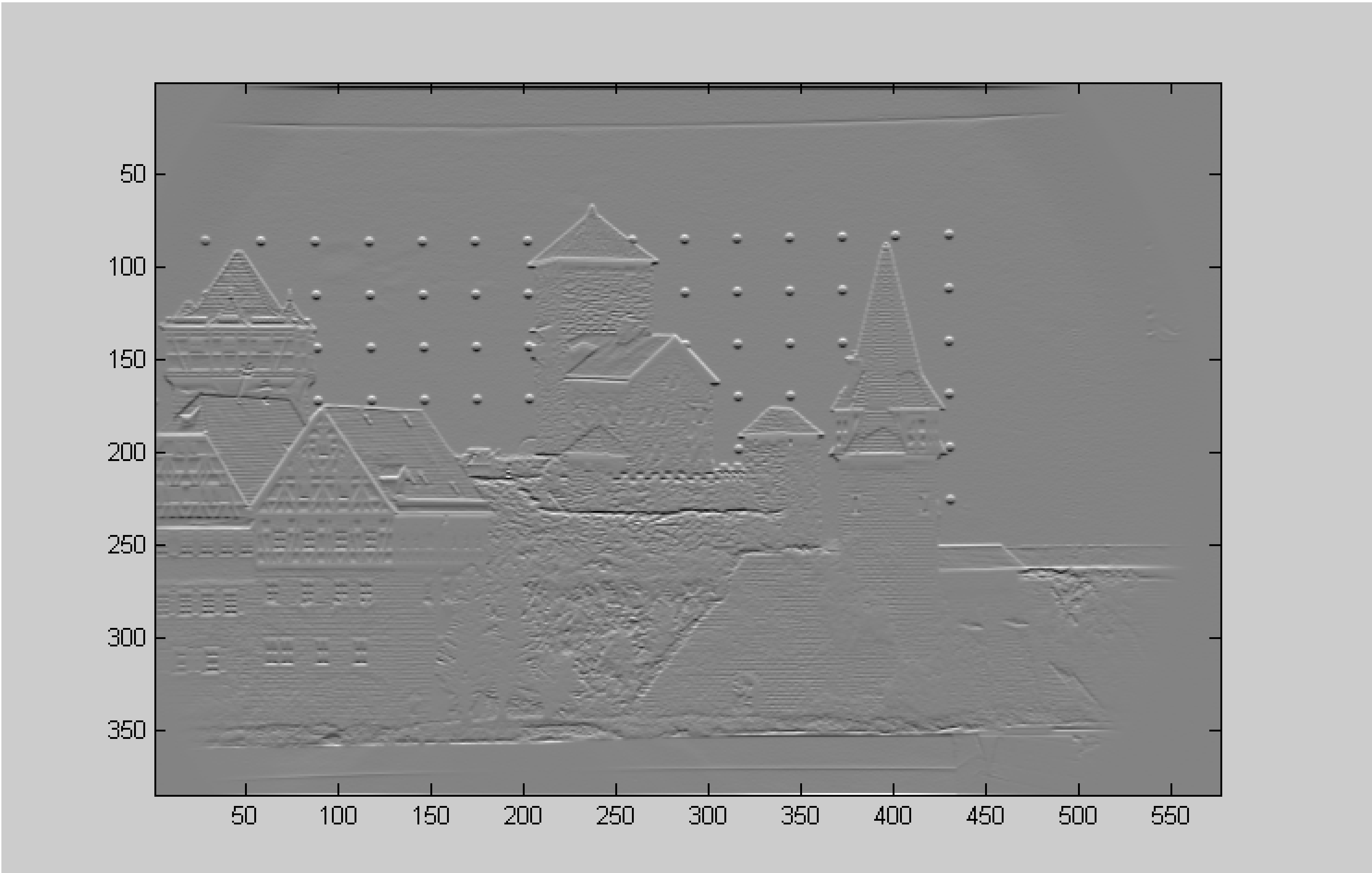
$f''(x)$

0	0	0	0	0	40	-40	0	0	0

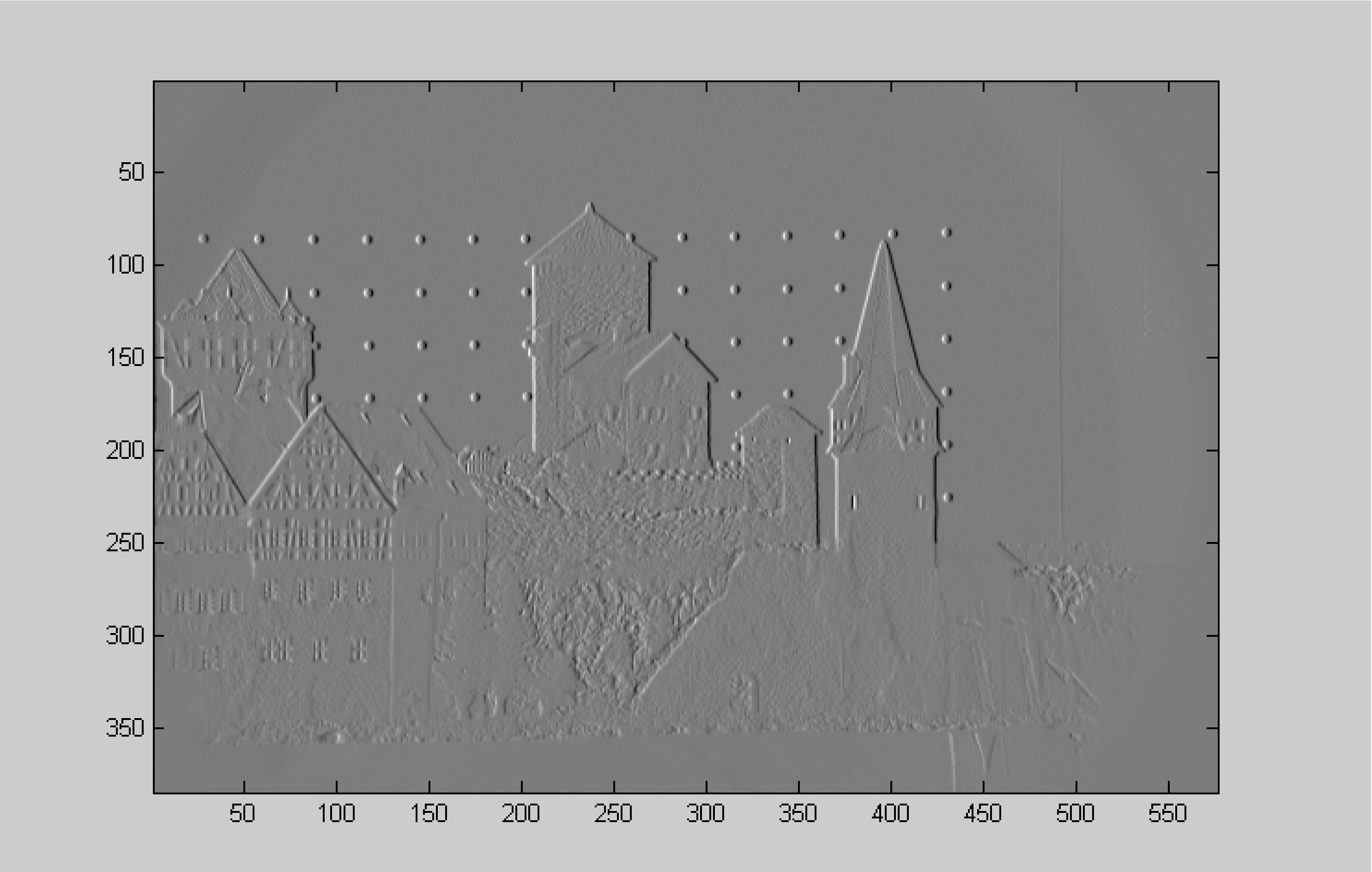
Original Image



First Derivative of the original image



Second Derivative of Original Image



Stages in Edge Detection

- Stage 1: Smoothing
  - Because Noise is enhanced in differentiation, we often apply a smoothing filter first
- Stage 2: Differentiation
  - Enhances edge areas
- Stage 3: Detection
  - Produces a binary output, with edges marked by 1, other areas as zero

Stage 1: Smoothing (Smoothing Masks)

- Mean Filter
- Gaussian Filter

1/9 x

1	1	1
1	1	1
1	1	1

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

5	21	35	21	5
21	94	155	94	21
35	155	255	155	35
21	94	155	94	21
5	21	35	21	5

Stage1 (CONT.):Filtering

- Gaussian Filter
- Pixel weight is inversely proportional to distance from origin

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

- Value of σ has to be specified

Stage1 (CONT.): Gaussian Filtering

- Example

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

- x = -1, y = -1, σ = 1

$$g(-1,-1)=e^{-\frac{(-1)^2+(-1)^2}{2}}\approx 0.367$$

0.367		

Stage1 (CONT.): Gaussian Filtering

- Implementation problem
  - Float multiplications are slow
- Solution
  - Multiply mask with 255, round to nearest integer
  - Scale answer by sum of all weights

Stage1 (CONT.): Gaussian Filtering

94	155	94
155	255	155
94	155	94

$\approx$

1	2	1
2	4	2
1	2	1

Round(g/70)

5	21	35	21	5
21	94	155	94	21
35	155	255	155	35
21	94	155	94	21
5	21	35	21	5

Stage1 (CONT.): Gaussian Filtering

0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	1	3	6	9	11	9	6	3	1	0	0
0	1	4	11	20	30	34	30	20	11	4	1	0
0	3	11	26	50	73	82	73	50	26	11	3	0
1	6	20	50	93	136	154	136	93	50	20	6	1
2	9	30	73	136	198	225	198	136	73	30	9	2
2	11	34	82	154	225	255	225	154	82	34	11	2
2	9	30	73	136	198	225	198	136	73	30	9	2
1	6	20	50	93	136	154	136	93	50	20	6	1
0	3	11	26	50	73	82	73	50	26	11	3	0
0	1	4	11	20	30	34	30	20	11	4	1	0
0	0	1	3	6	9	11	9	6	3	1	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0

Figure 2.3: Gaussian mask with $\sigma = 2$.

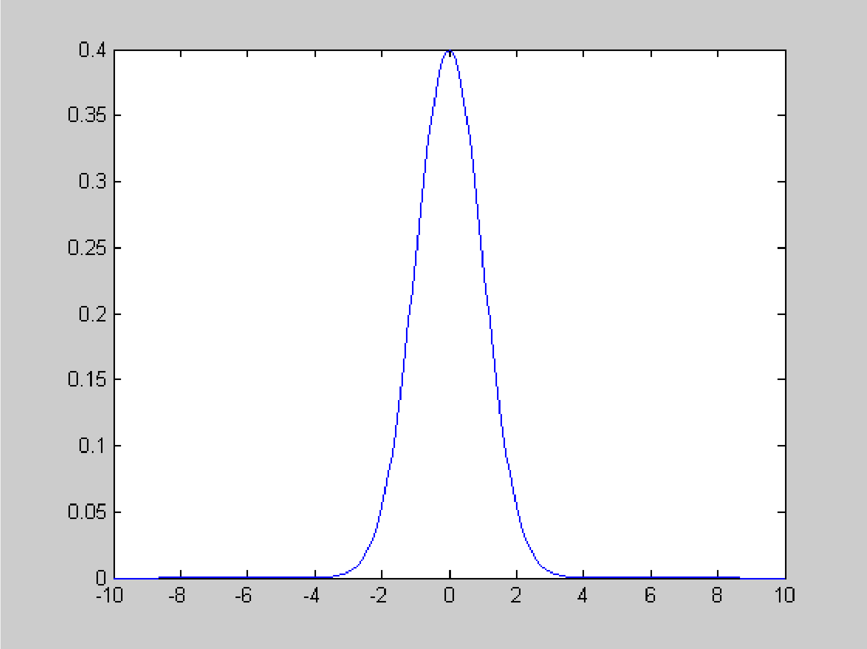
Stage1 (CONT.): Continuous Gaussian Function

- 1-D Gaussian

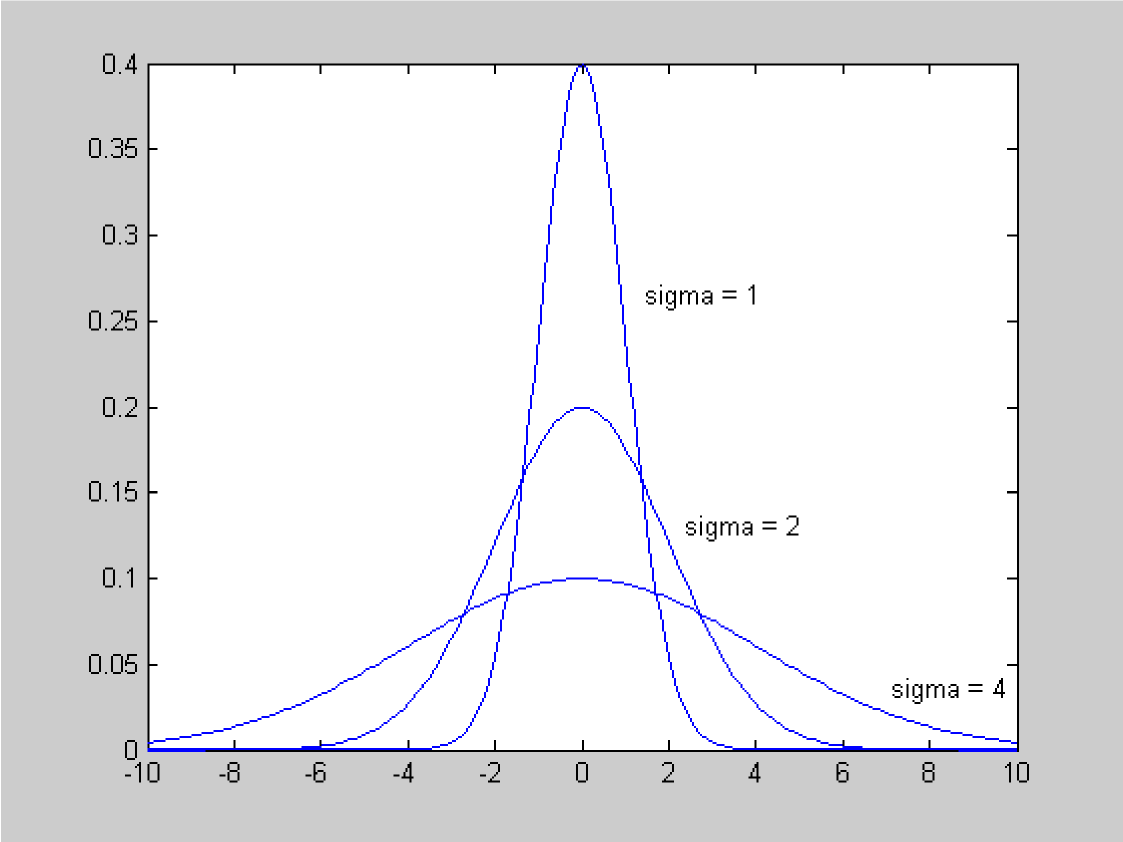
$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$$

- Non zero from -∞ to ∞
- Width is controlled by σ
- Symmetric
- Area under curve = 1

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} = 1$$

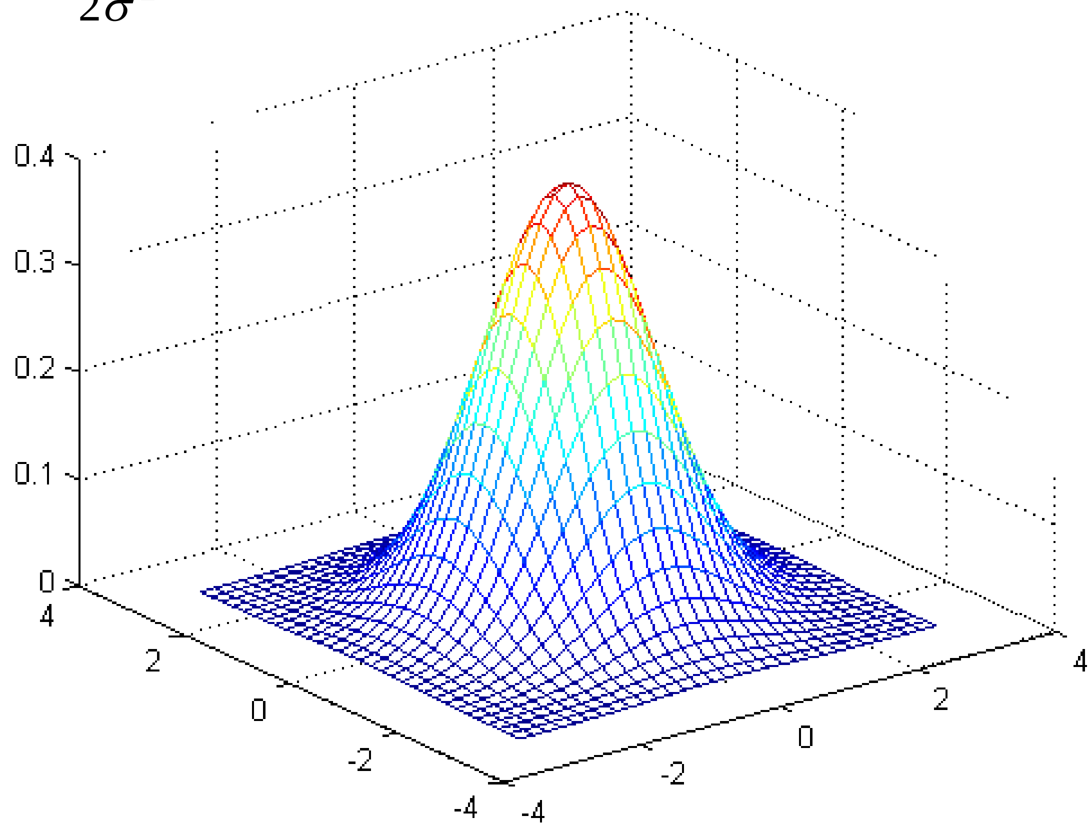


Stage1 (CONT.): Continuous Gaussian Function

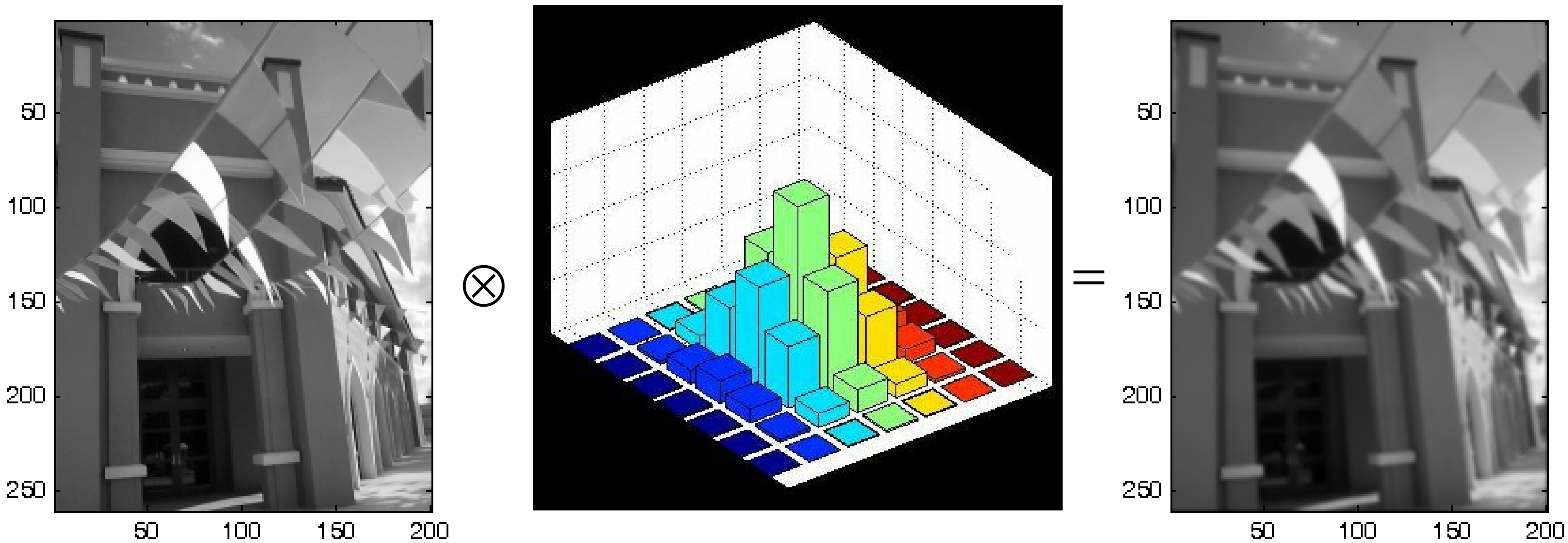


Stage1 (CONT.): 2-D Gaussian Function

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



Stage1 (CONT.): Gaussian Filter



Stage 2 Differentiation: Derivative Masks

From definition of derivatives

-1	1

-1
1

Prewitt Operator

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

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

Stage 2 CONT:

- Sobel Operator

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

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

- Robert's Operator

1	0
0	-1

0	1
-1	0

Stage 2 (CONT.):Properties of Masks

- Smoothing Masks
  - All values are +ve
  - Sum to 1
  - Output on smooth region is unchanged
  - Blurs areas of high contrast
  - Larger mask -> more smoothing
- Derivative Masks
  - opposite signs
  - Sum to zero
  - Output on smooth region is zero
  - Gives high output in areas of high contrast
  - Larger mask -> more edges detected

Stage 2 (CONT.): Application of 2-D Masks

- If f_x is derivative in x-direction, f_y is derivative in y-direction
- Gradient Magnitude

$$M = \sqrt{f_x^2 + f_y^2}$$

- Gradient Direction

$$\theta = \arctan \frac{f_y}{f_x}$$

Stage3: Detection Stage - Threshold

- Gradient Magnitude

$$M(x,y) = \sqrt{f_x^2(x,y) + f_y^2(x,y)}$$

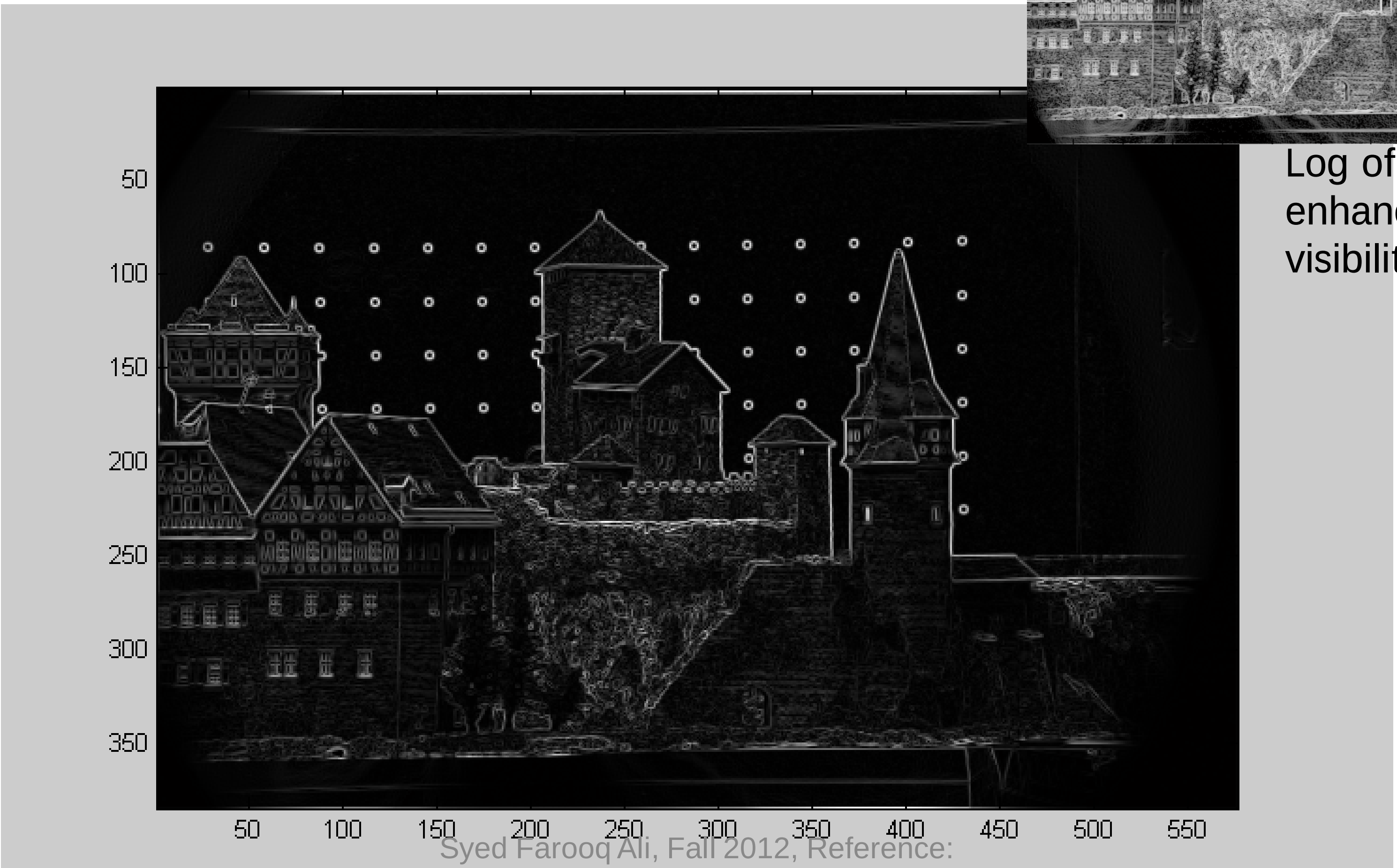
- Gradient Magnitude normalized b/w 0-100

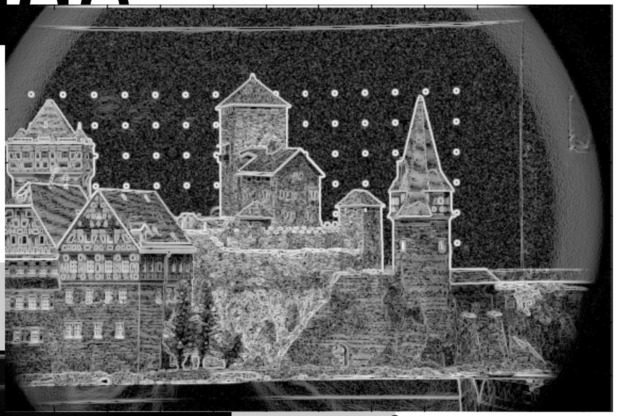
$$N(x,y) = \frac{M(x,y)}{max_{i=1,\ldots,n,j=1,\ldots,n} M(i,j)} \times 100.$$

- Application of a threshold

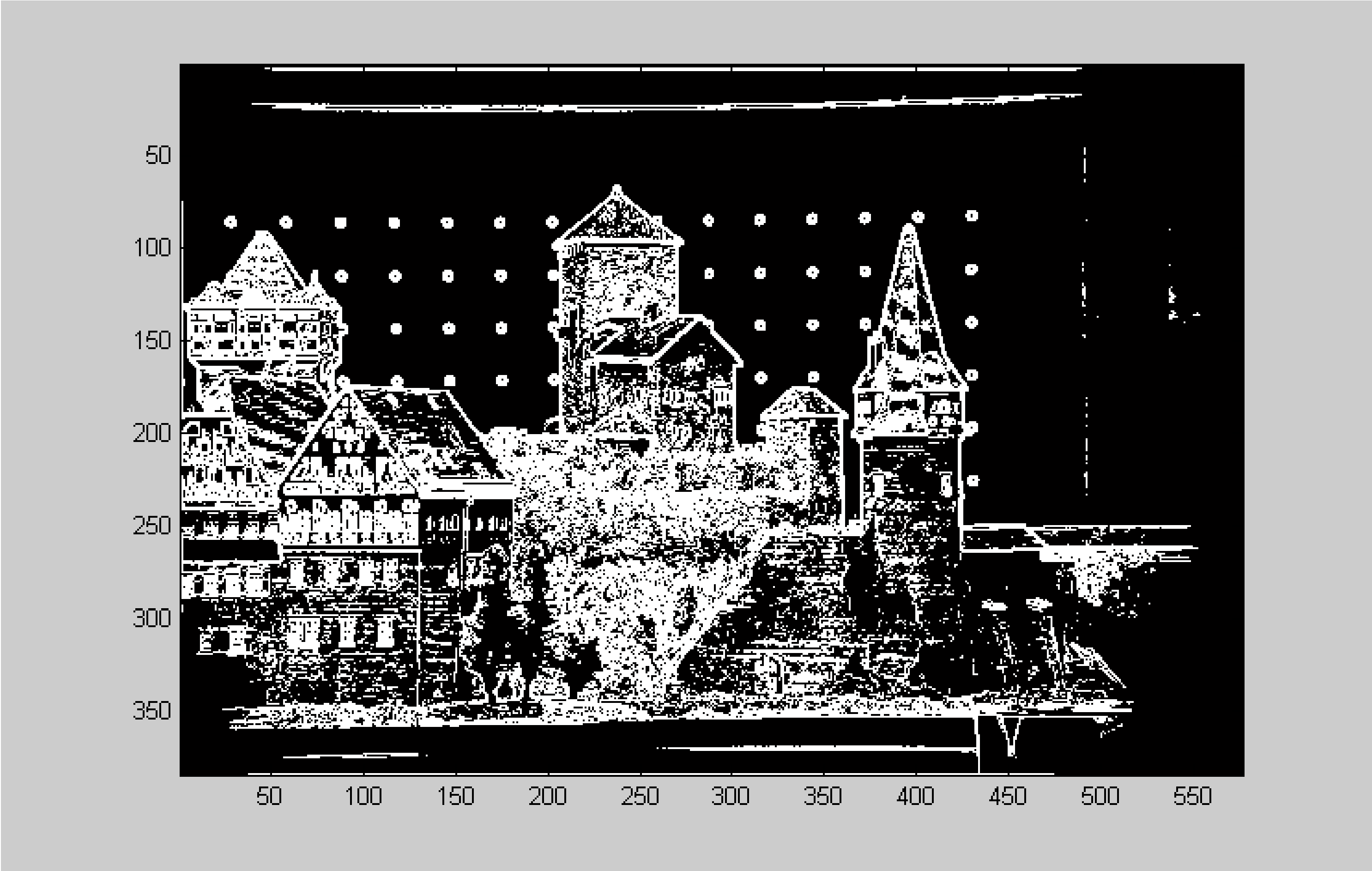
$$E(x,y) = \begin{cases} 1 & \text{if } N(x,y) > T \\ 0 & \text{otherwise} \end{cases}$$

Gradient Magnitude

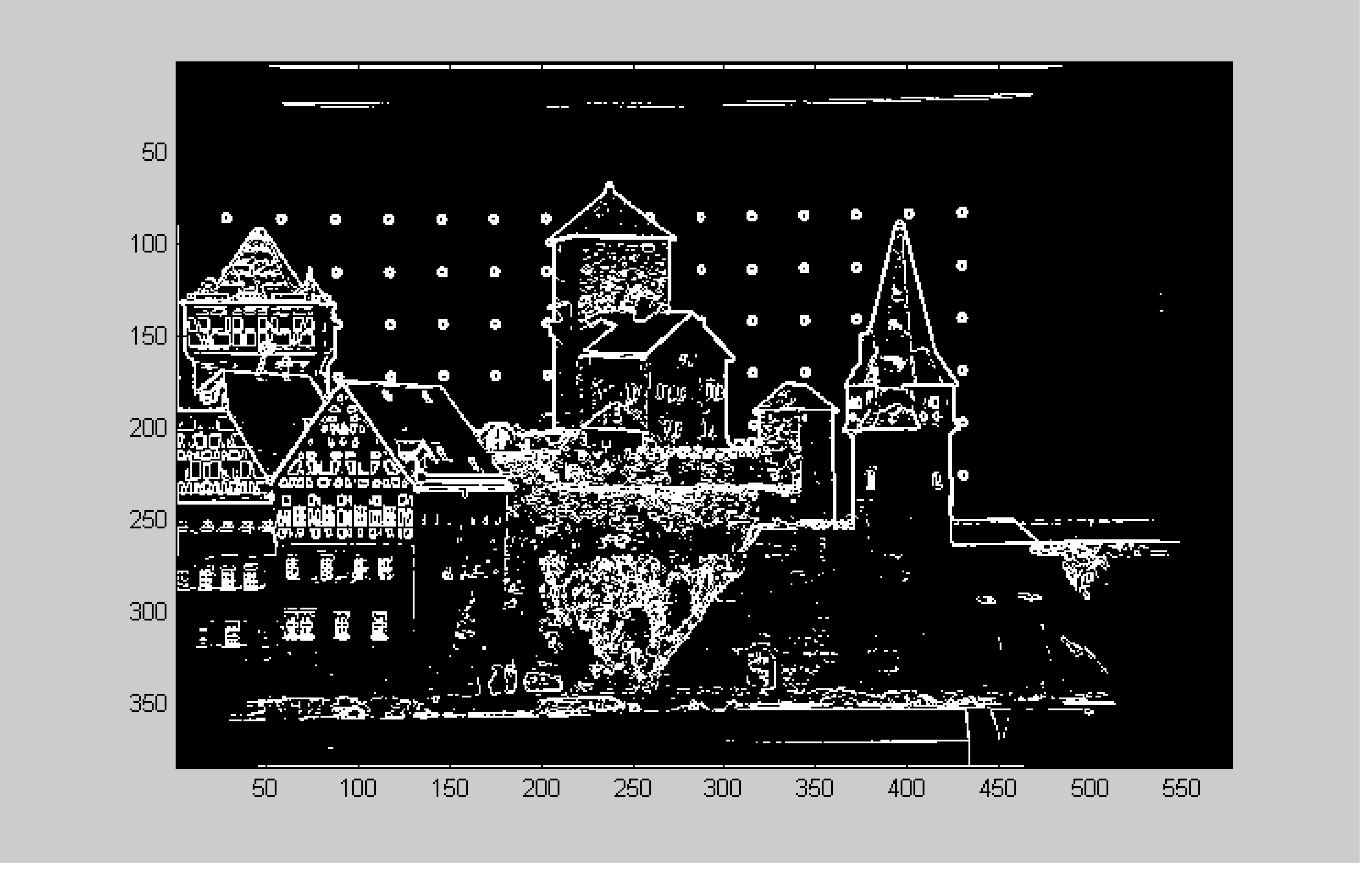




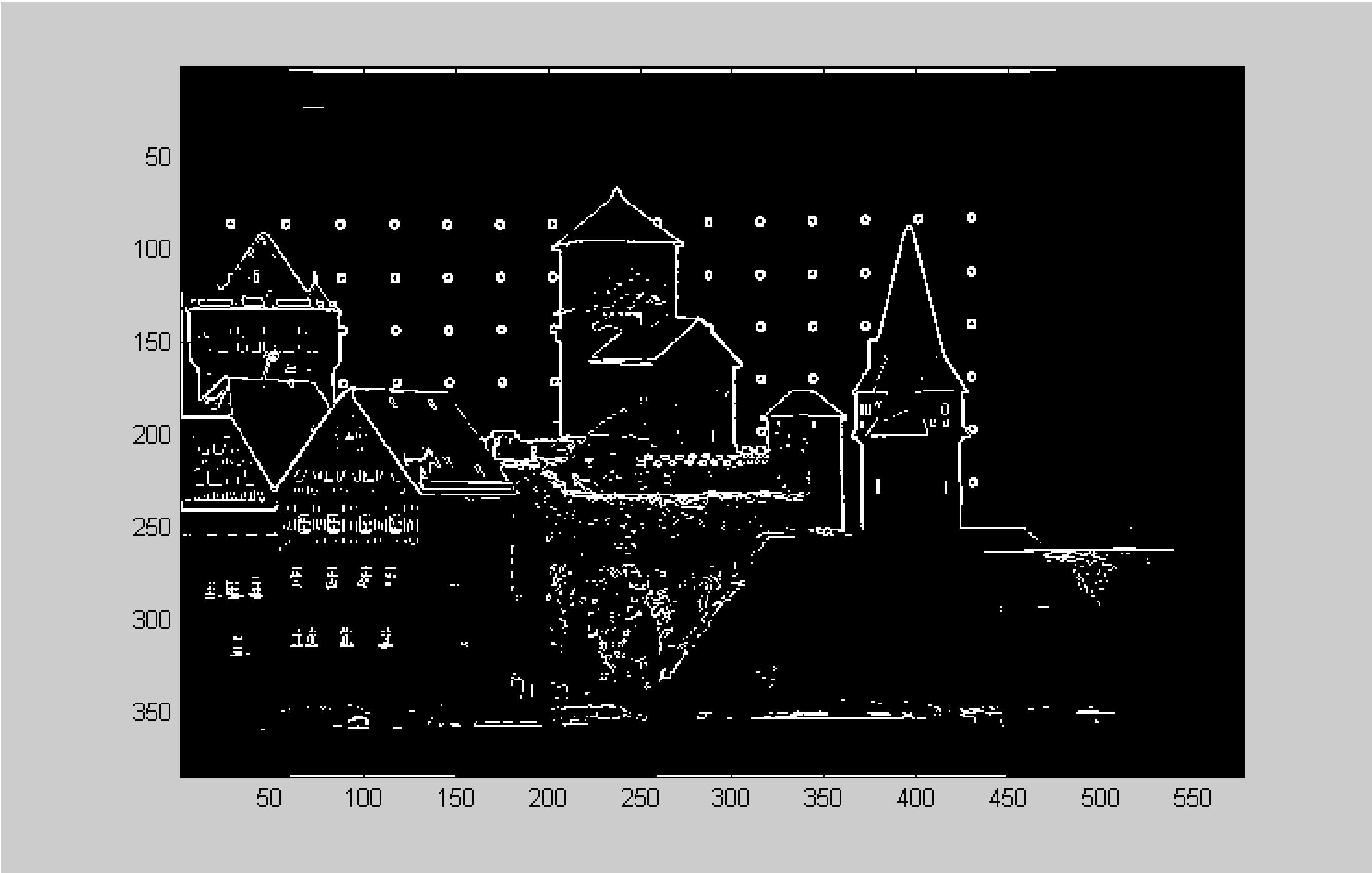
Log of M to enhance visibility


50
100
150
200
250
300
350
50
100
150
200
250
300
350
400
450
500
550

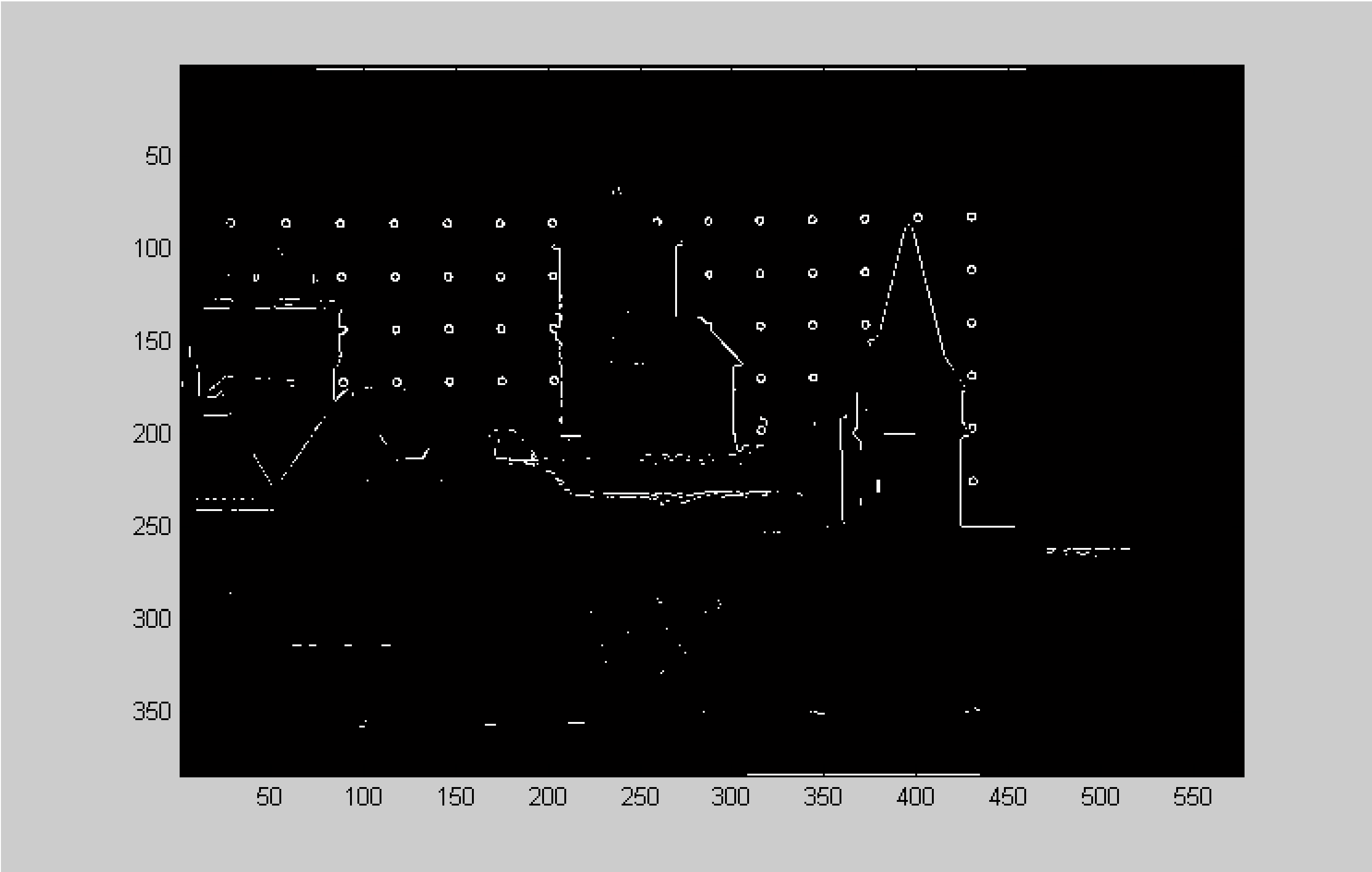
T = 10


50
100
150
200
250
300
350
50 100 150 200 250 300 350 400 450 500 550

T = 20


50
100
150
200
250
300
350
50
100
150
200
250
300
350
400
450
500
550

T = 40


50
100
150
200
250
300
350
50
100
150
200
250
300
350
400
450
500
550

T = 80

What about Gradient Direction?

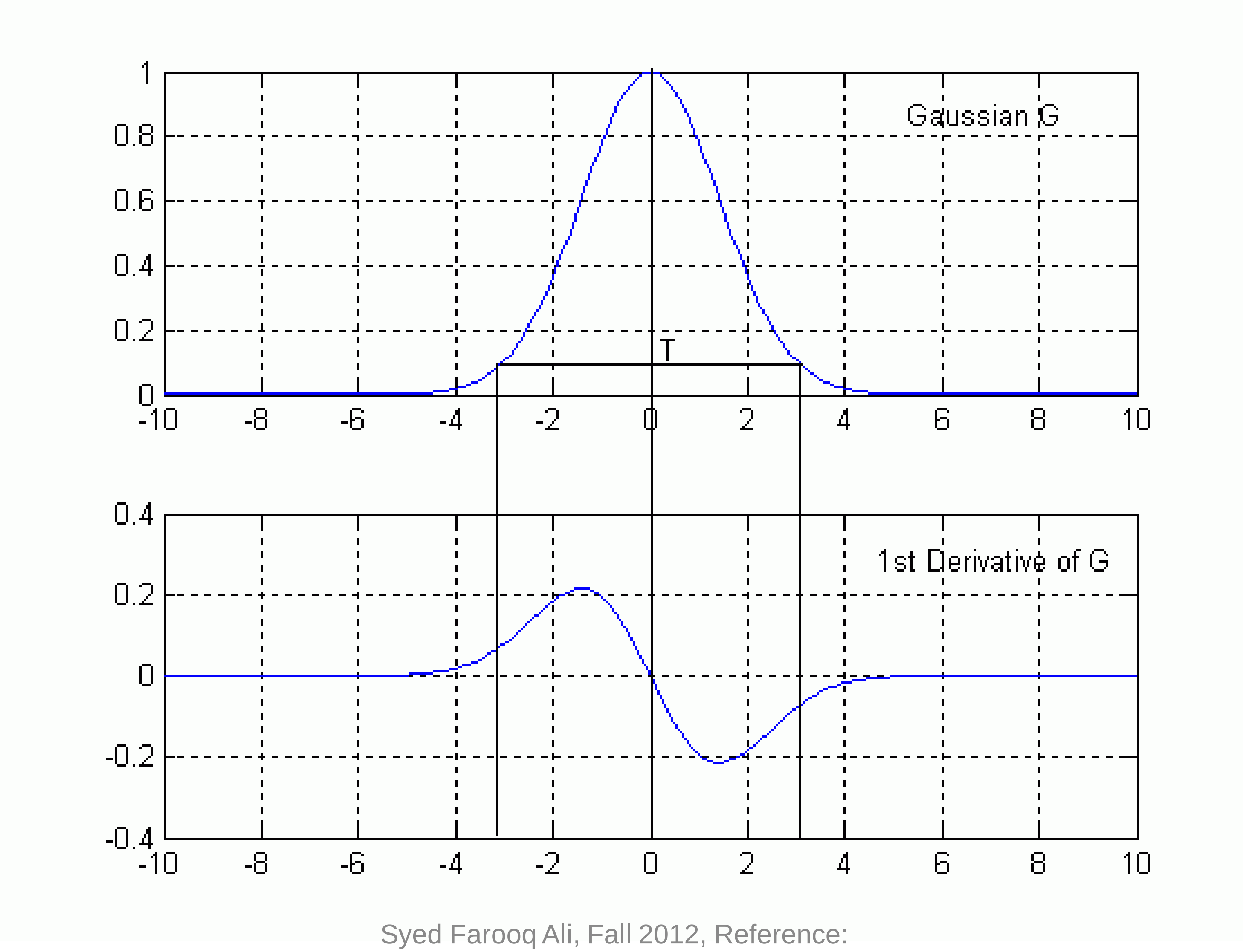
- Gradient Direction is always perpendicular to edge
- Direction of most change of gray levels
- Thick edges can be eliminated using gradient direction
- Weak edges also captured in this manner

Canny's Edge Detector

- Filtering + Derivative:
  – Uses first derivative Gaussian masks
- Detection:
  – Uses Non-Maxima Suppression
  – Uses Hysteresis Thresholding

First Derivative of Gaussian

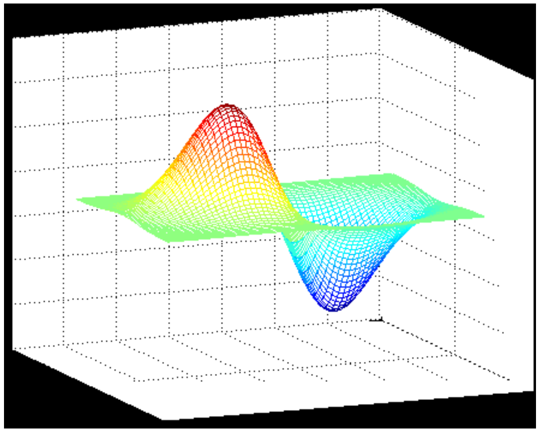
- Expression?
- Effect?
- Filtering + Derivative


Gaussian G
T
1st Derivative of G
1
0.8
0.6
0.4
0.2
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
0.4
0.2
0
-0.2
-0.4
-10
-8
-6
-4
-2
0
2
4
6
8
10

Canny Edge Operator

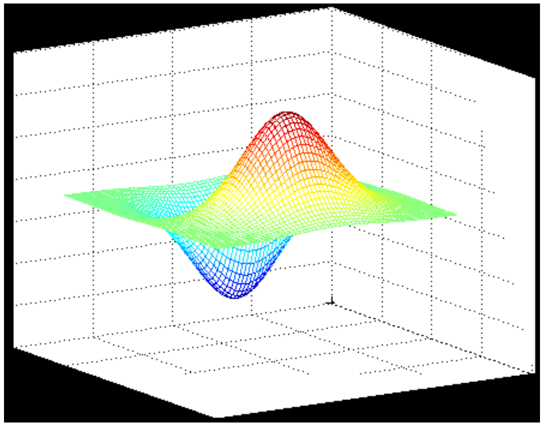
$$\Delta S = \Delta\left(G_\sigma * I\right) = \Delta G_\sigma * I$$

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



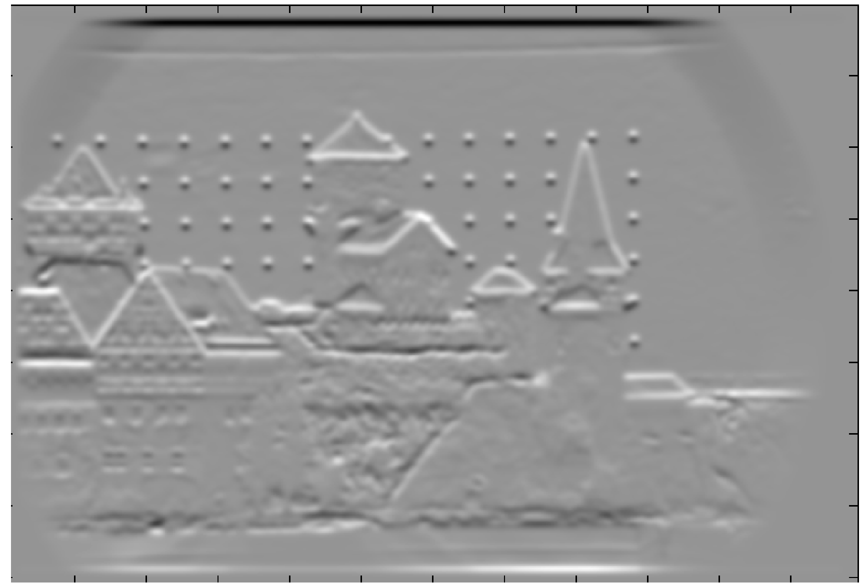
$$\Delta S = \begin{bmatrix} \dfrac{\partial G_\sigma}{\partial x} * I & \dfrac{\partial G_\sigma}{\partial y} * I \end{bmatrix}^T$$

$$f_x(x,y) = f(x,y) * \left(\frac{-x}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

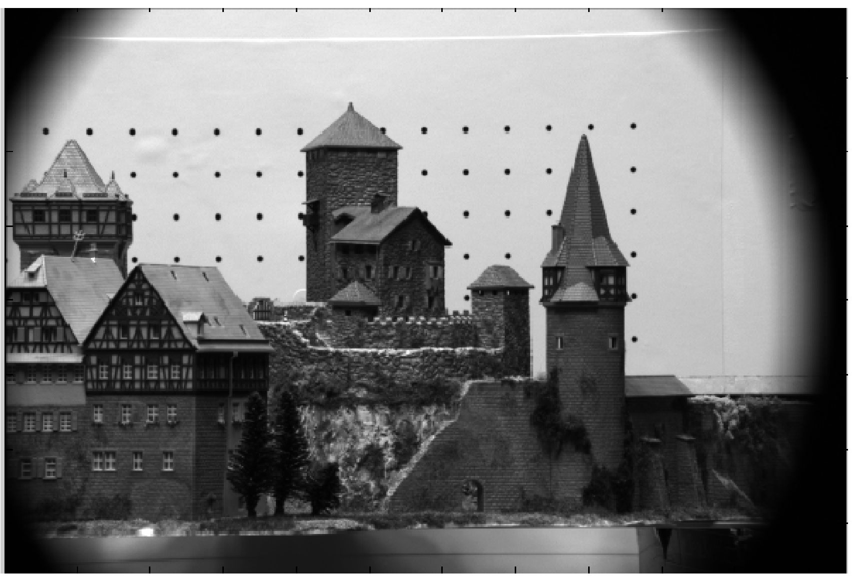


$$f_y(x,y) = f(x,y) * \left(\frac{-y}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

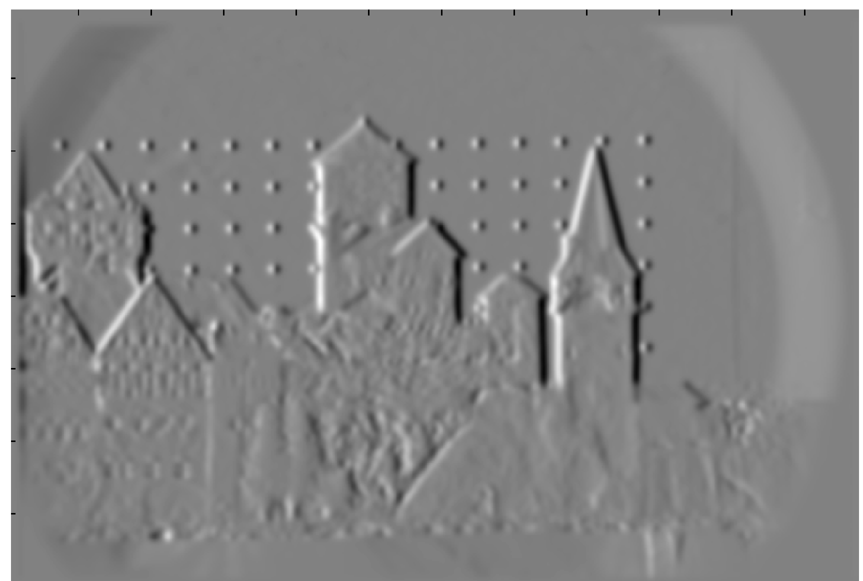
f_x



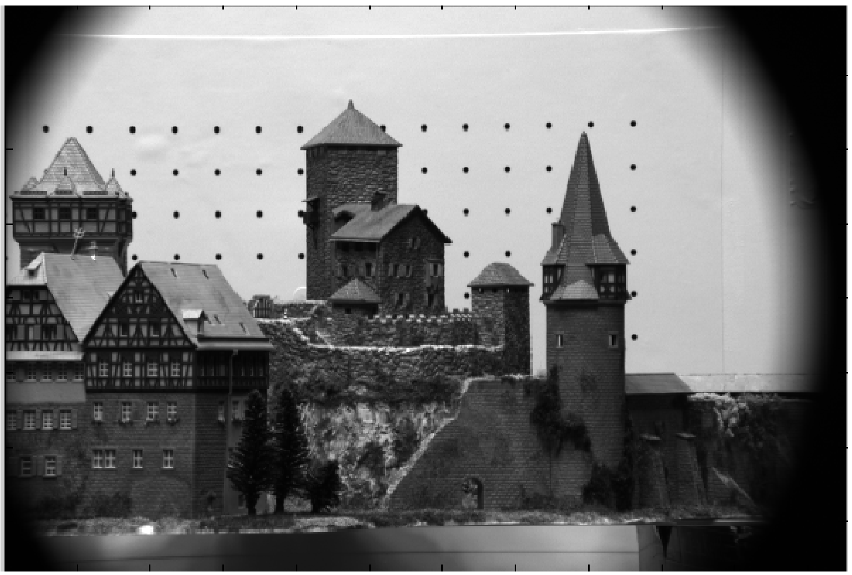
I

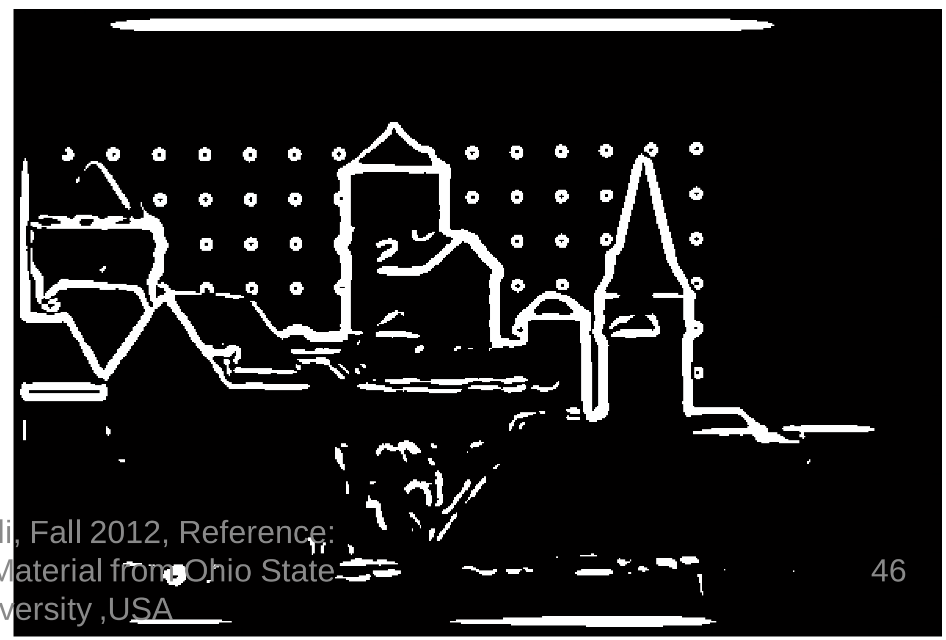


f_y



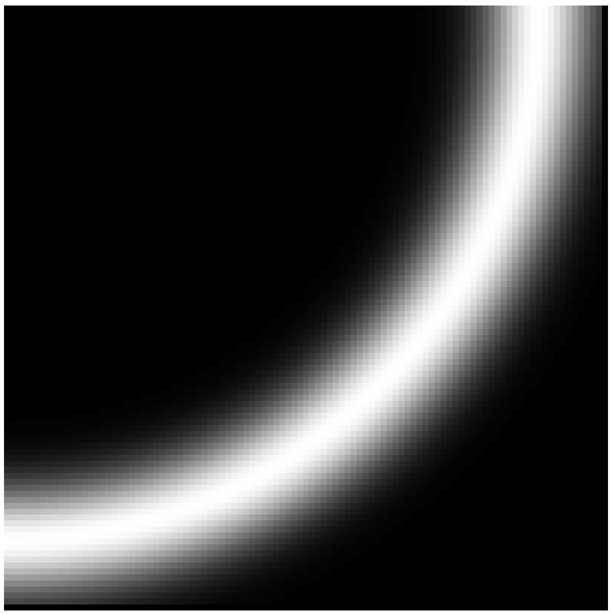
$$M = \sqrt{f_x^2 + f_y^2}$$

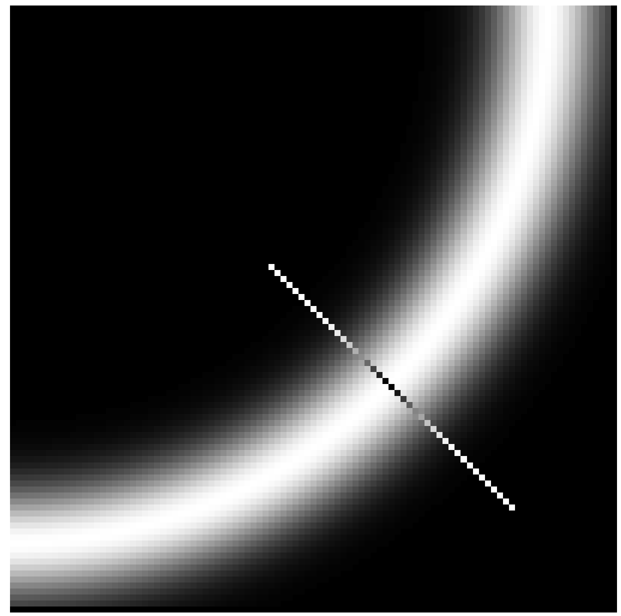




$$M \geq Threshold = 10$$

Non-Maximum Suppression





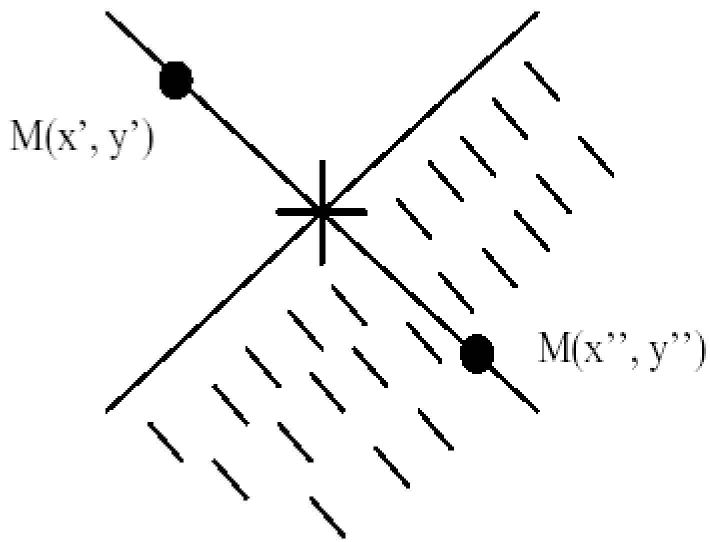
Remove all points along the gradient direction that are not maximum points

Non-Maxima Suppression

$$M(x,y) = \begin{cases} M(x,y) & \text{if } M(x,y) > M(x\prime, y\prime) \text{ and} \\ & \text{if } M(x,y) > M(x\prime\prime, y\prime\prime) \\ 0 & \text{otherwise} \end{cases}$$

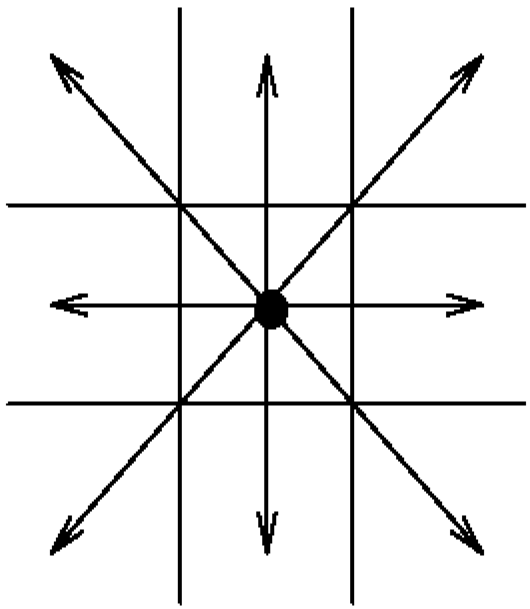
Where M(x',y') and M(x'',y'') are gradient magnitudes on both sides of edge at (x,y) in the gradient magnitude direction

Non-Maxima Supression



Quantization of Gradient Direction

$$\theta = \arctan \frac{f_y}{f_x}$$



Non-Maximum Suppression

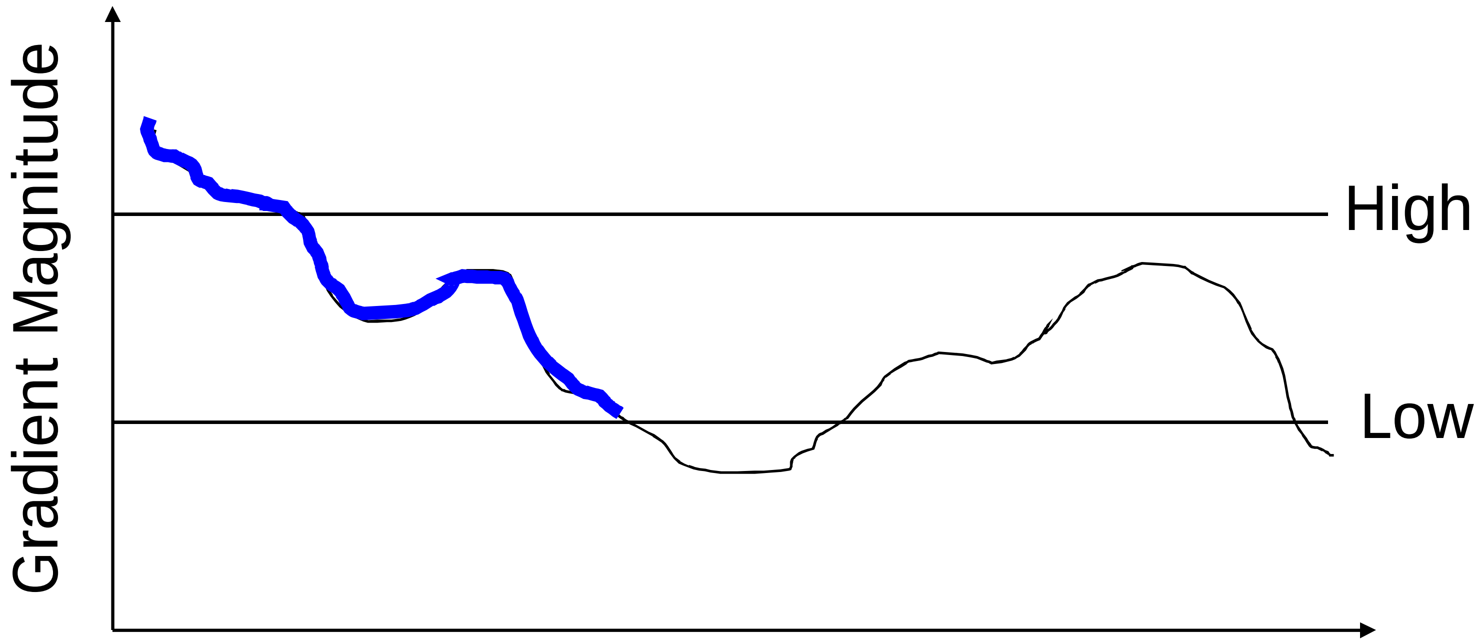
M when $\sigma = 2$

Non-maxima suppressed image

Hysteresis Thresholding

- Two thresholds, T_H and T_L
- Apply non-maxima suppression to M (gradient magnitude)
- Scan image from left to right, top to bottom
- If M(x,y) is above T_H mark it as edge
- Recursively look at neighbors; if gradient magnitude is above T_L mark it as edge

Hysteresis Thresholding



Algorithm Summary

- Compute gradient of image *f*(*x*,*y*) by convolving with first derivative of Gaussian in x and y directions

$$f_x(x,y) = f(x,y) * \left(\frac{-x}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}},$$

$$f_y(x,y) = f(x,y) * \left(\frac{-y}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

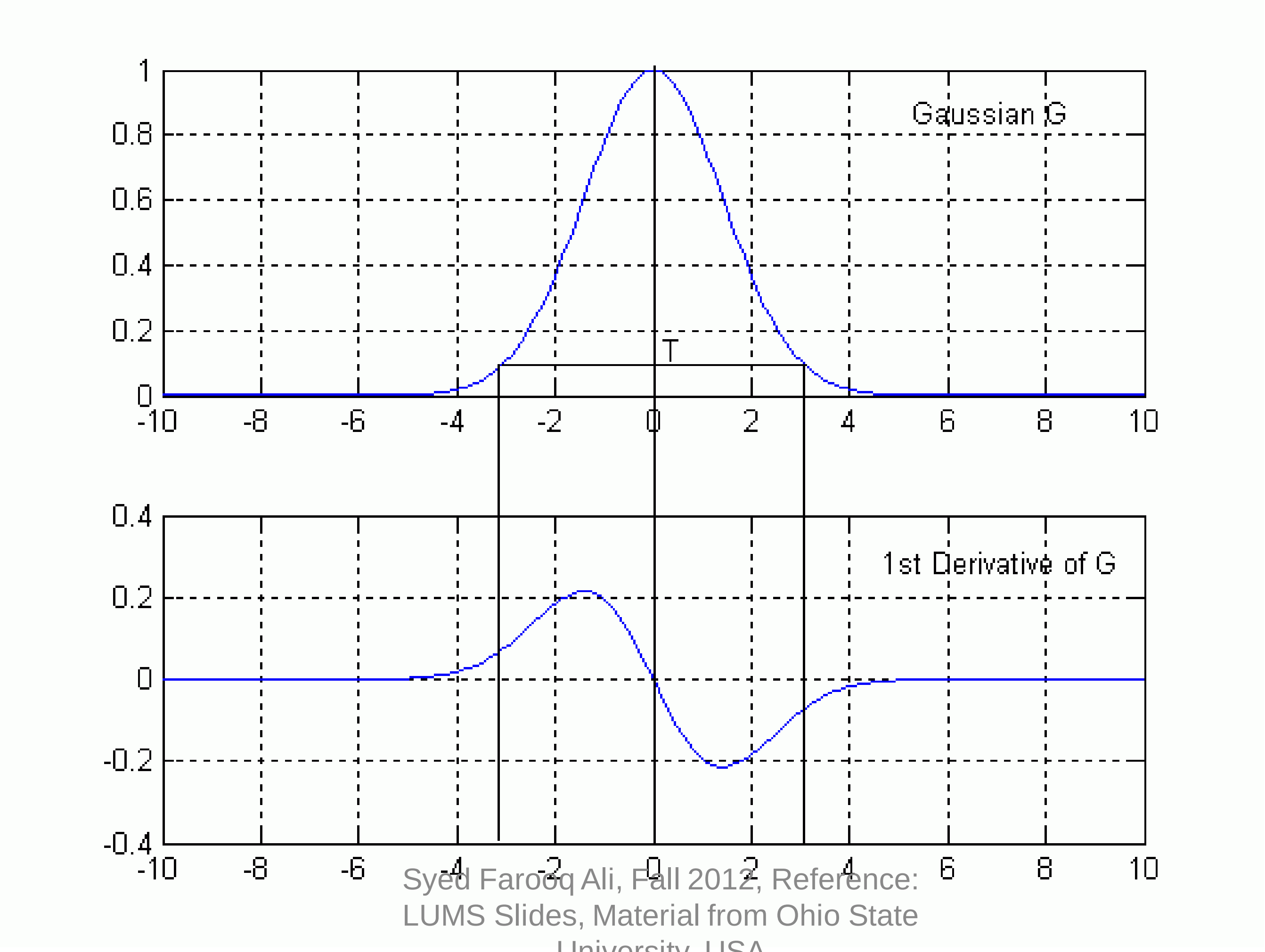
Algorithm Summary

- Compute gradient magnitude and direction at each pixel
- Perform non-maxima suppression
  - Find gradient direction at pixel
  - Quantize it in 8 directions {0, 45, 90, 135, … 315}
  - Compare current value of M with two neighbors in appropriate direction
  - If maximum, keep it, otherwise make it zero

Algorithm Summary

- Perform Hysteresis thresholding
  - Scan image from left to right, top to bottom
  - If $M(x,y)$ is above T_H mark it as edge
  - Recursively look at neighbors; if gradient magnitude is above T_L mark it as edge

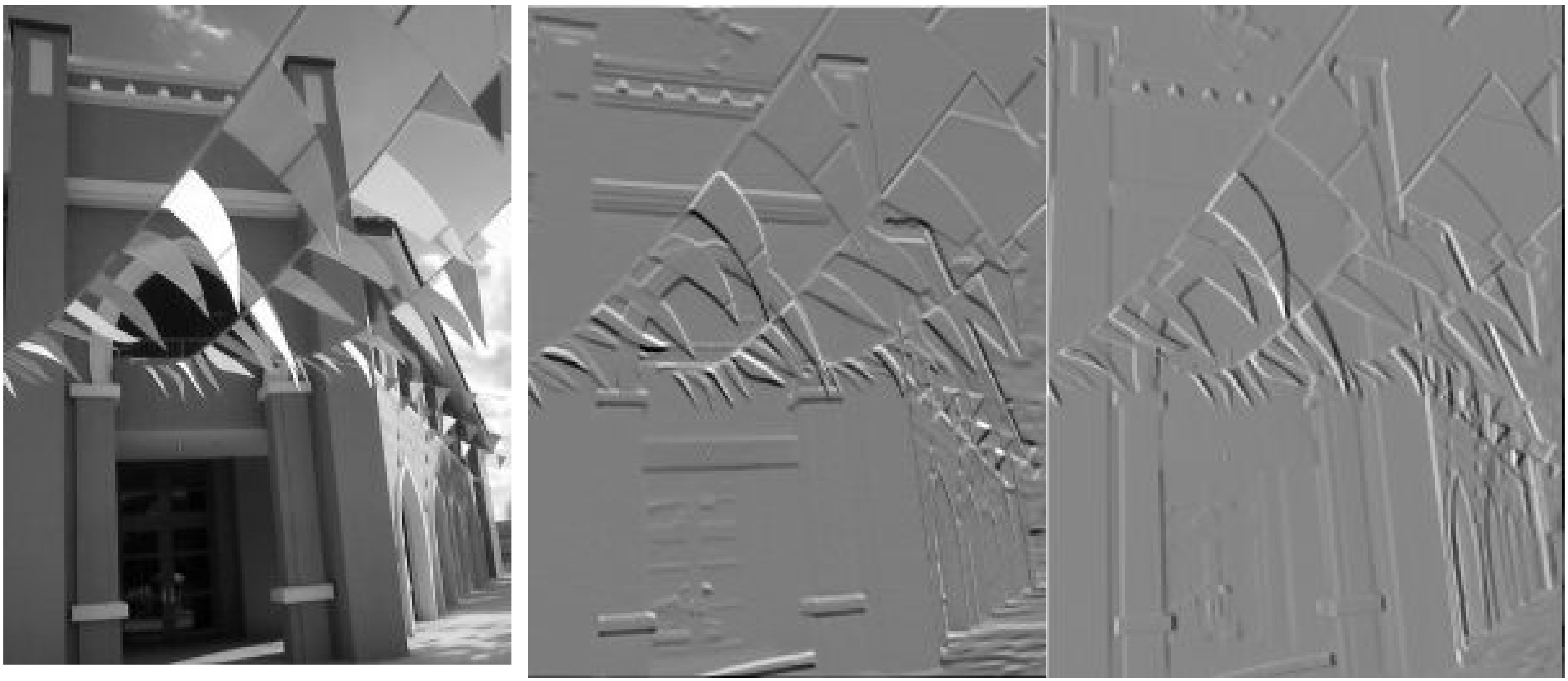
Choice of Sigma



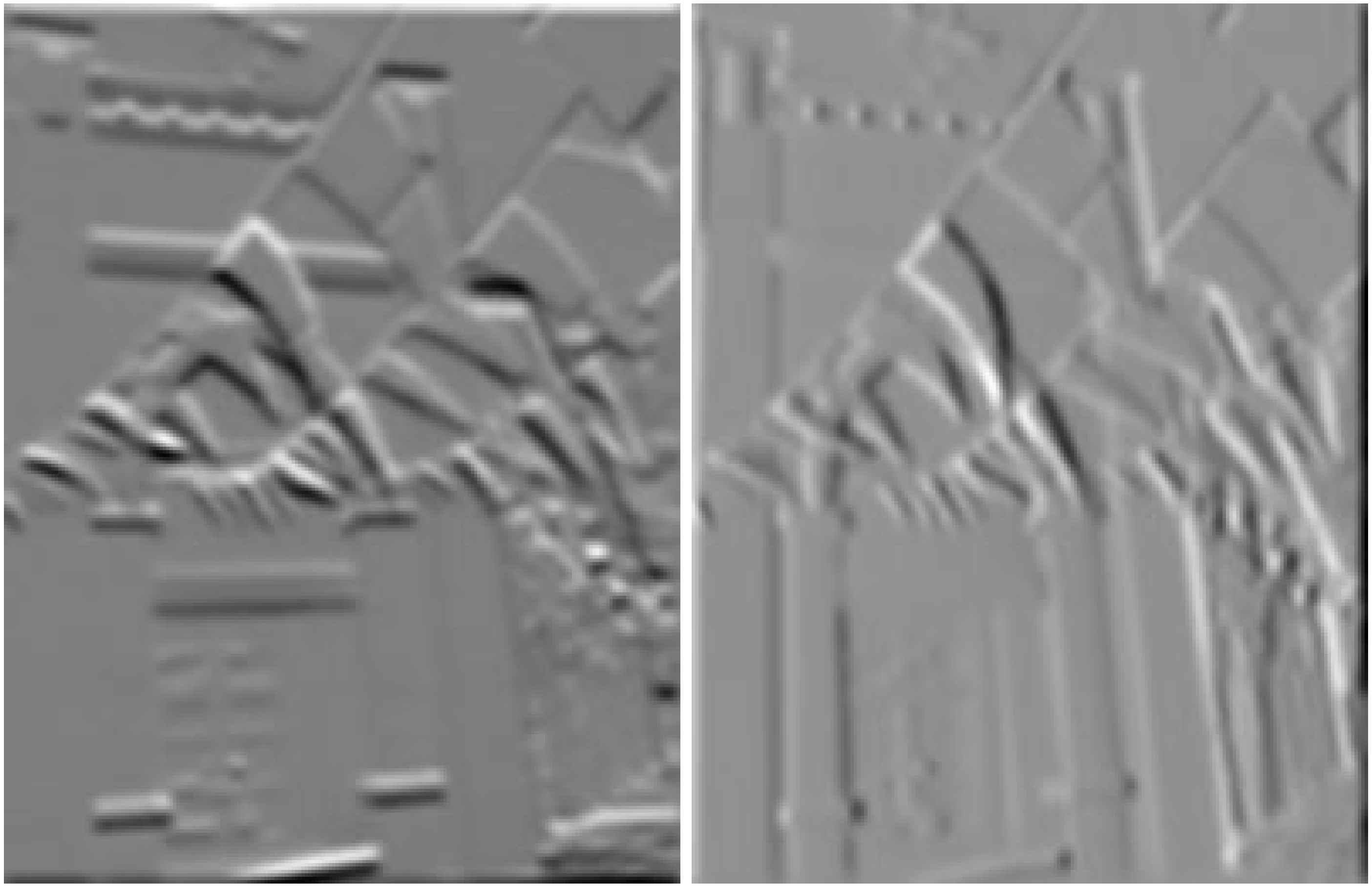
Choice of Signma

Sigma	Size of Mask
0.5	3x3
1	5x5
2	9x9
3	13x13
4	19x19

Conv. Results



sigma = 0.5



sigma = 2

Gradient Magnitude Results

sigma = 0.5

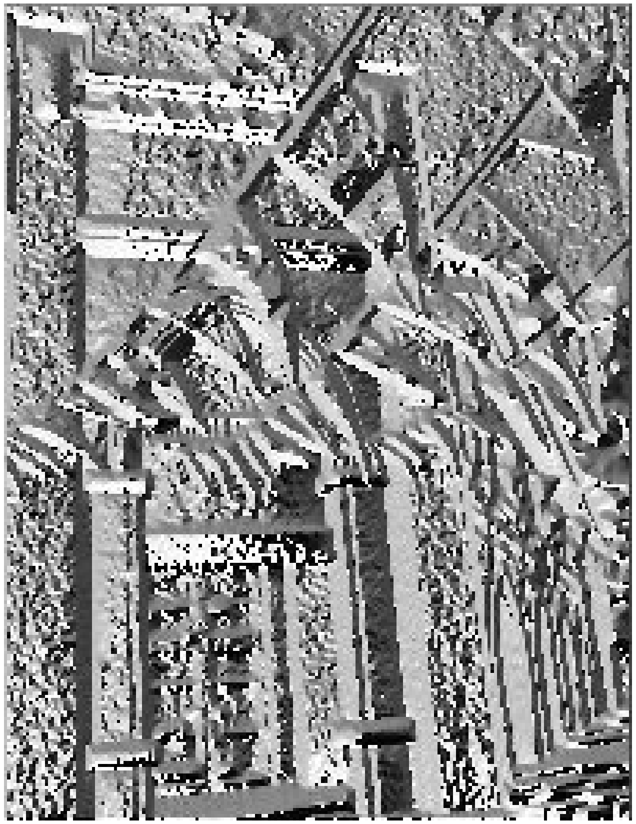
sigma = 2

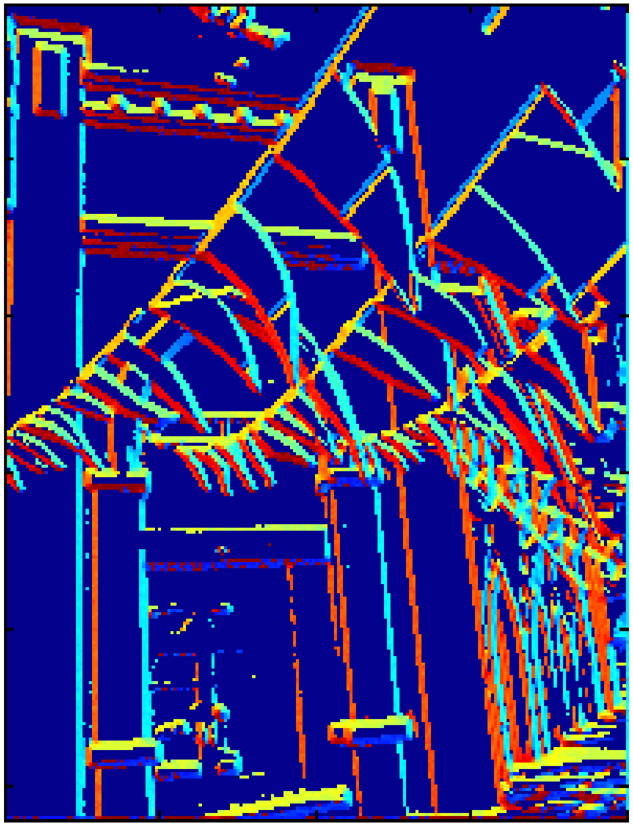
Gradient Magnitude Result

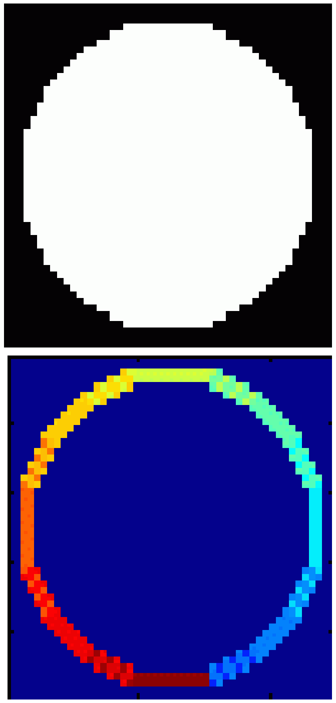
M when $\sigma = 2$

Non-maxima suppressed image

Gradient Direction Results







Raw

Quantized

Hysteresis Thresholding

Th = 50
Tl = 10

Th = 100
Tl = 10
(incr. in Th only)

Th = 50
Tl = 40
(incr. in Tl only)

Proj1: Thumb Recognition

Proj 2: Advance Features in Thumb Recognition

Proj 3: Face Recognition

Proj 4:Face Recognition

Proj 5: Face to Cartoon

Proj 6: Fall Detection

Proj 7: License Plate Recognition

Proj 8: Matching Sketch to Image

Proj 9: Watermarking

Proj 10:Image Editor