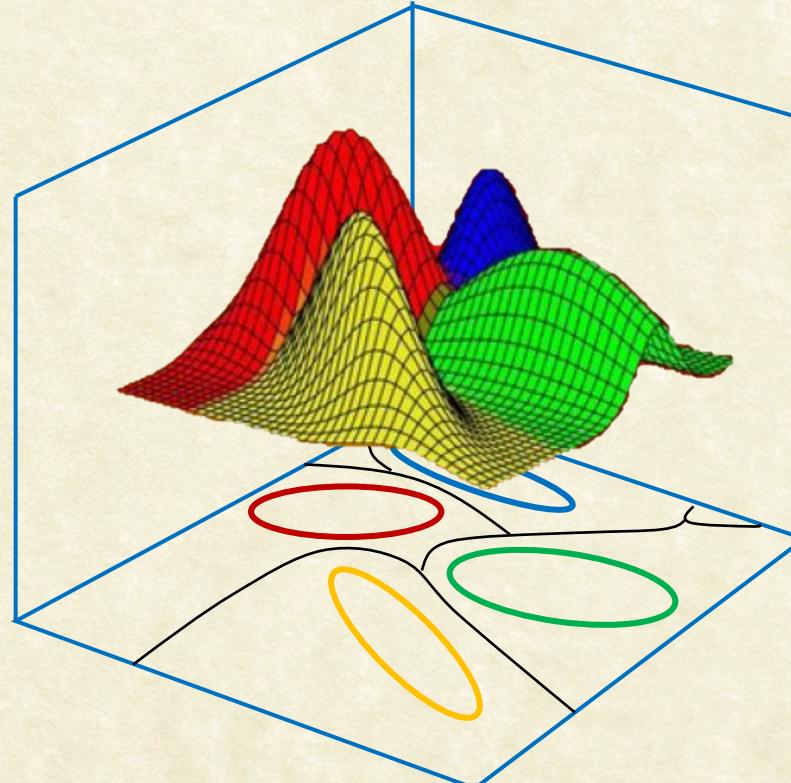




# CS7.404: Digital Image Processing

## Monsoon 2023: Spatial Domain Processing



Anoop M. Namboodiri

Biometrics and Secure ID Lab, CVIT,  
IIIT Hyderabad

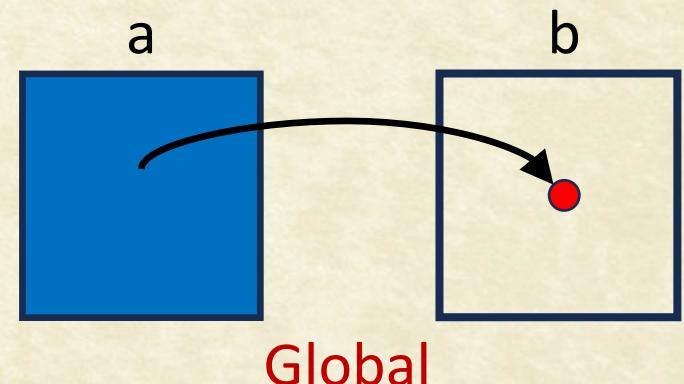
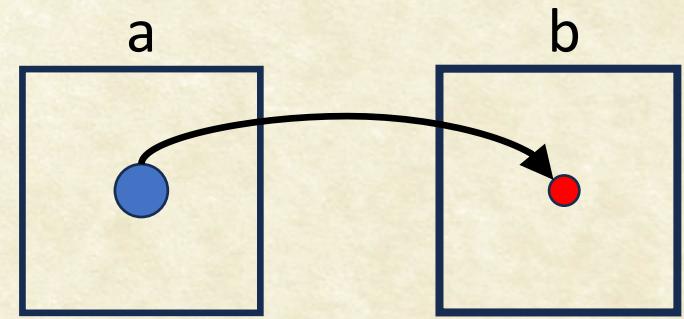
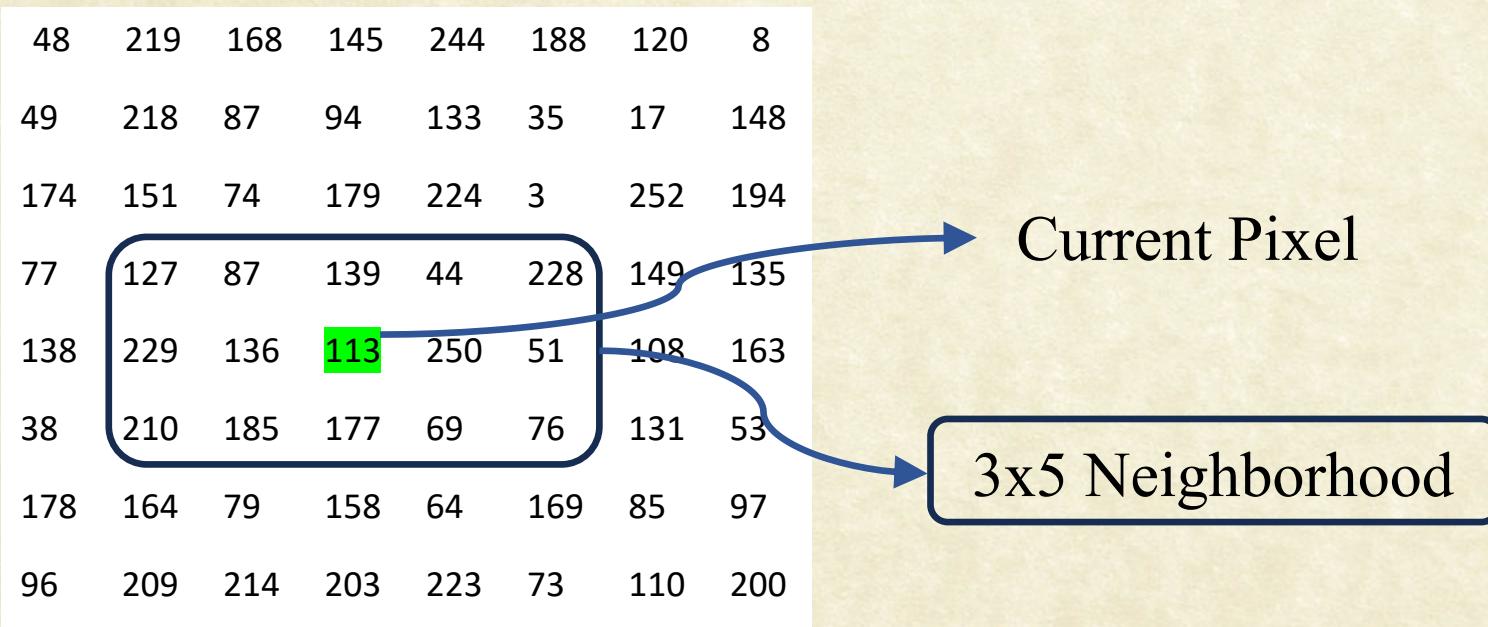


# Spatial Domain Processing

## ► Manipulating Pixels Directly in Spatial Domain

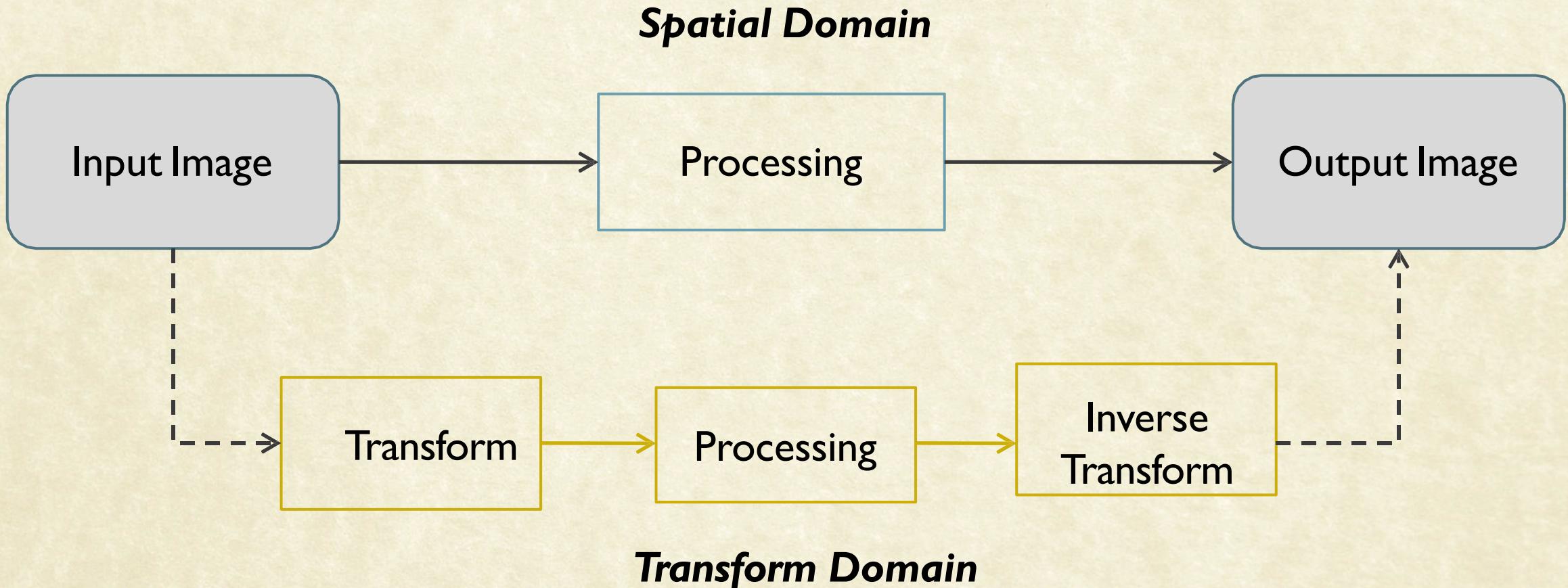
3 approaches

1. Point to Point
2. Neighbourhood to Point
3. Global to Point





# Spatial vs. Transform Domain Processing





# Spatial vs. Transform Domain Processing



Bandhani / Bandhej



Tie & Dye



# Spatial vs. Transform Domain Processing

Transform (Tie)



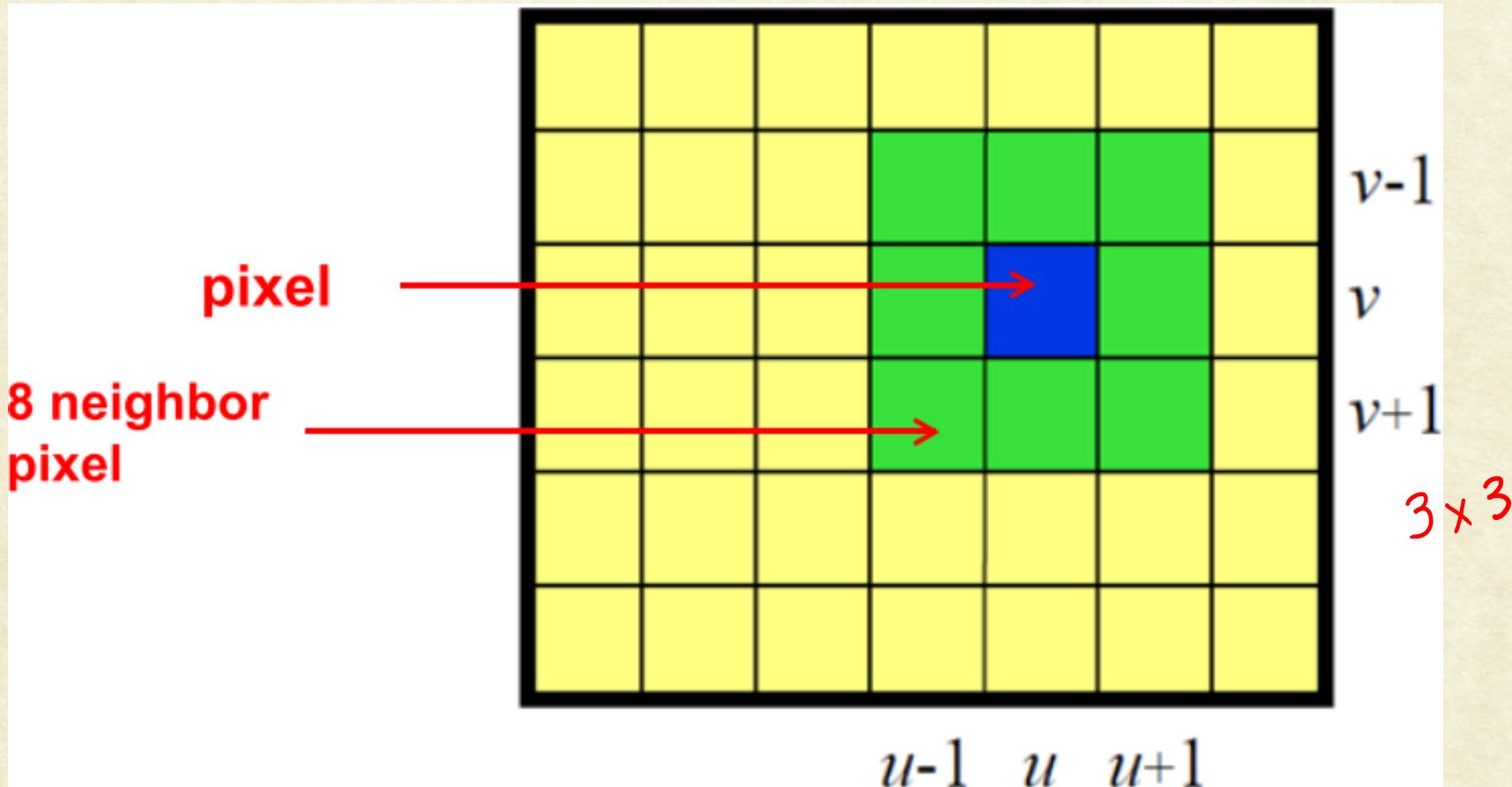
Process (Dye)

Inverse Transform (Untie)



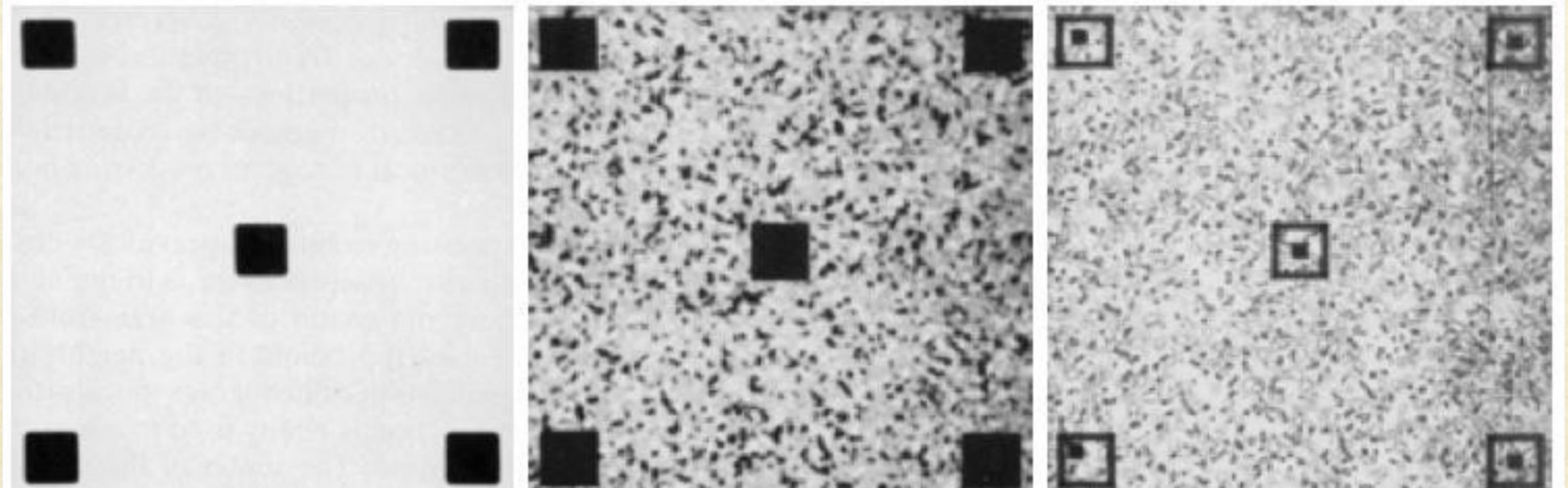


# Pixel Neighborhood





# Local Histogram Processing



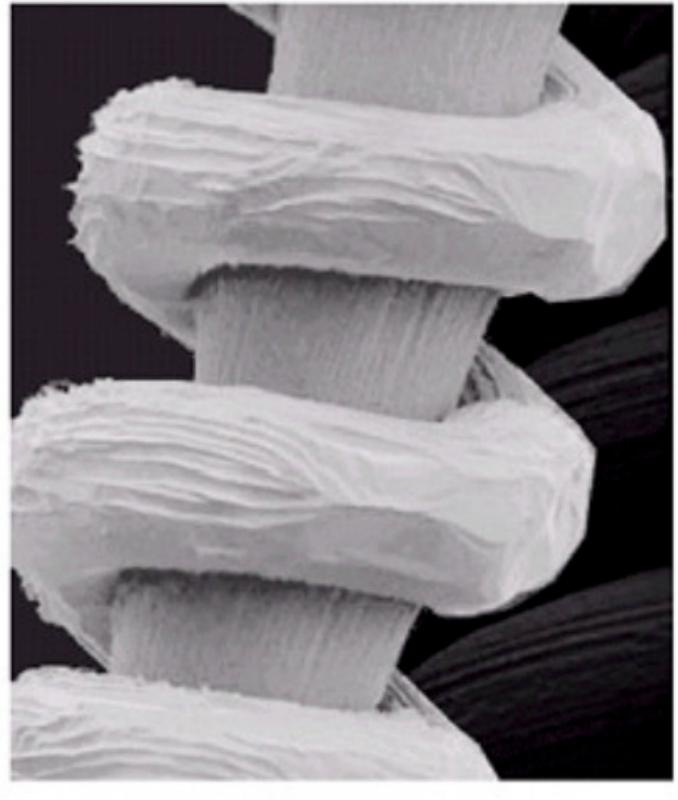
Global Hist Eq

Local Hist Eq

Image Courtesy: Gonzalez and Woods



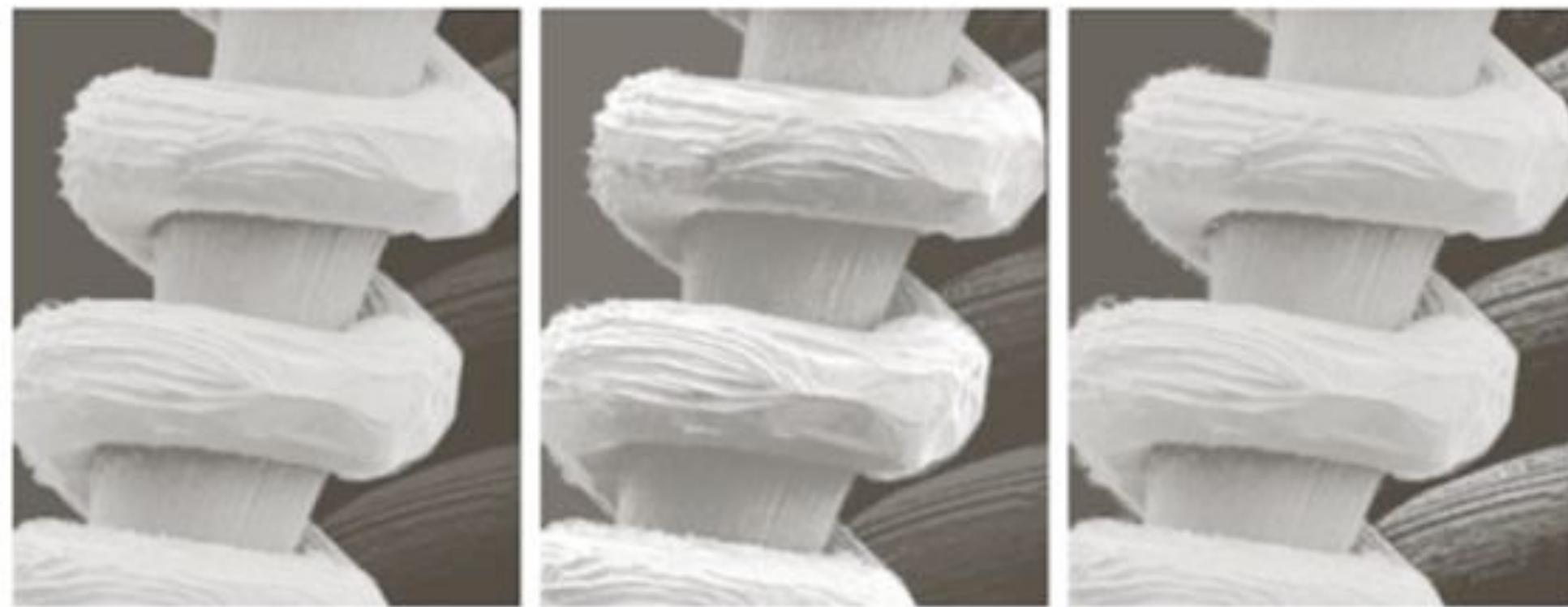
# Conditional Image Enhancement



- Objective for given image: Enhance dark areas while leaving light areas unchanged

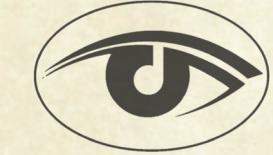


# Image Enhancement Using Histogram Statistics



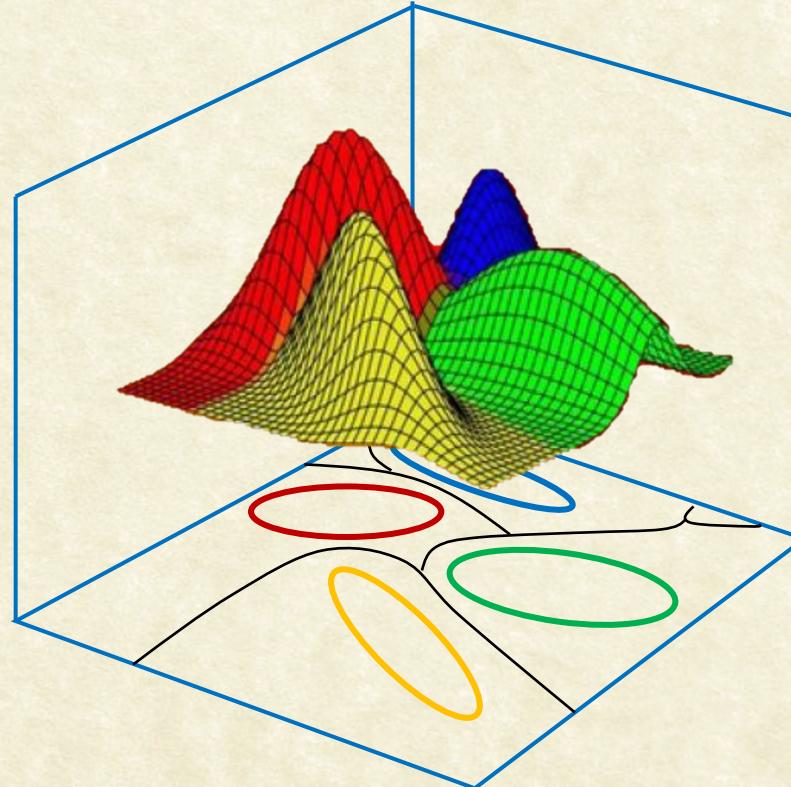
a b c

**FIGURE 3.27** (a) SEM image of a tungsten filament magnified approximately  $130\times$ . (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)



# CS7.404: Digital Image Processing

## Monsoon 2023: Spatial Filters



Anoop M. Namboodiri

Biometrics and Secure ID Lab, CVIT,  
IIIT Hyderabad

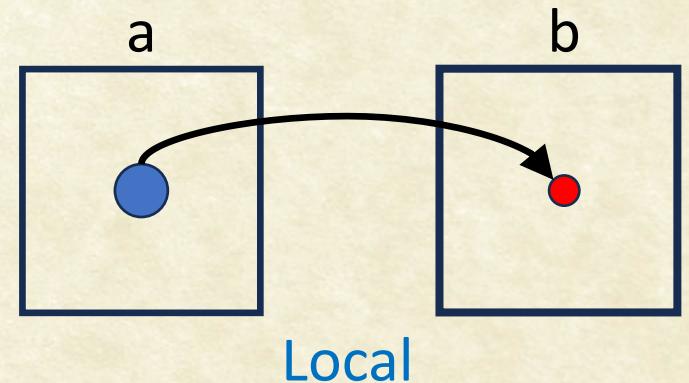


# Spatial Domain Processing

## ► Manipulating Pixels Directly in Spatial Domain

3 approaches

1. Point to Point
2. Neighbourhood to Point

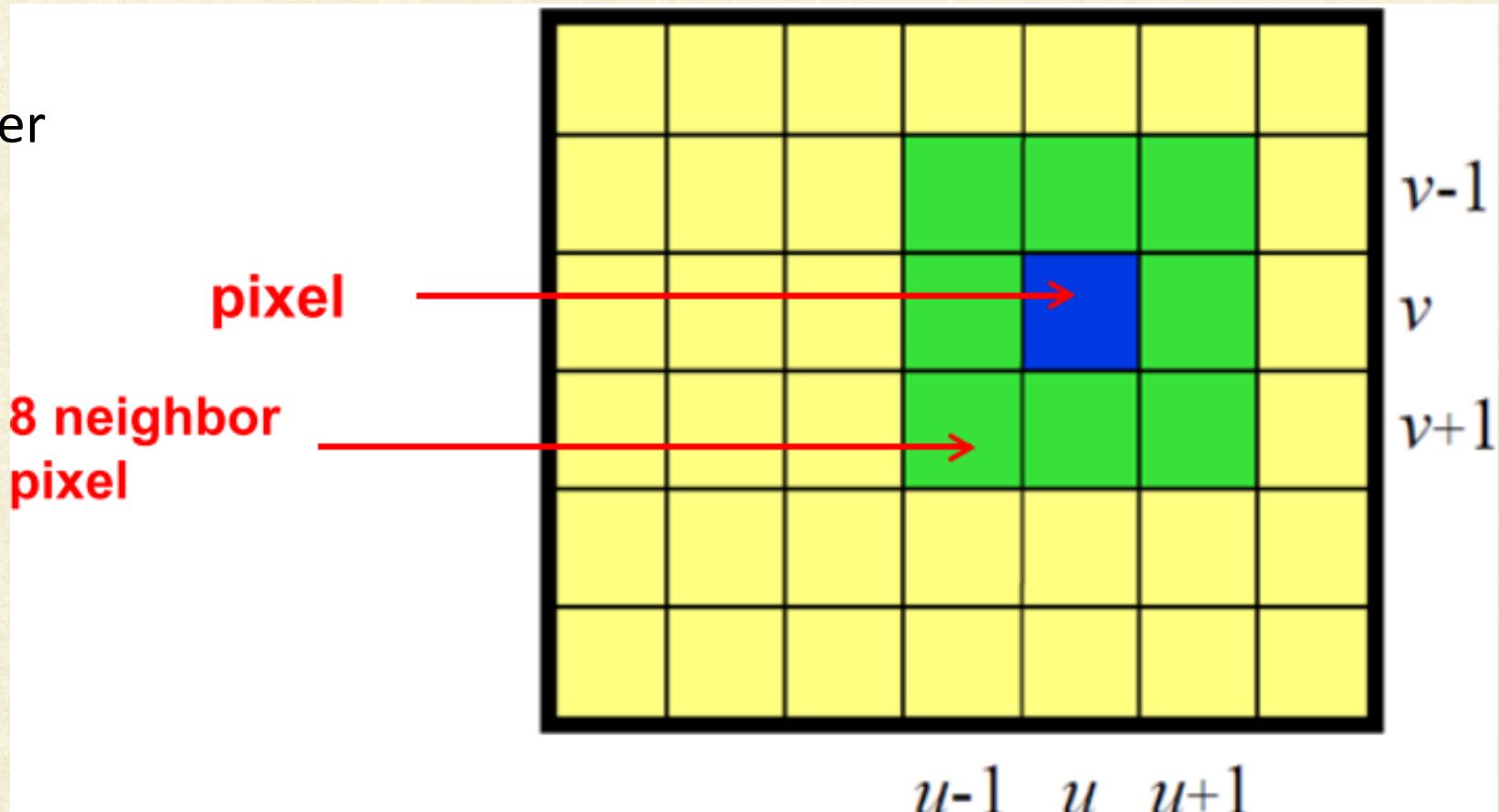




# Spatial Domain Filtering

- 2 components of a spatial filter
  - neighborhood
  - a predefined operation

- Filter aka
  - Kernel
  - Spatial Mask
  - Template
  - Window





# Mean/Average Filter (Smoothing)

M = 3

For each valid location [x,y] in S

$a \leftarrow$  Average of intensities in a  $M \times M$  neighborhood centered on [x,y]

$D[x,y] = \text{round}(a)$

120	190	140	150	200
17	21	30	8	27
89	123	150	73	56
10	178	140	150	18
190	14	76	69	87

$$x = \begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

		98		



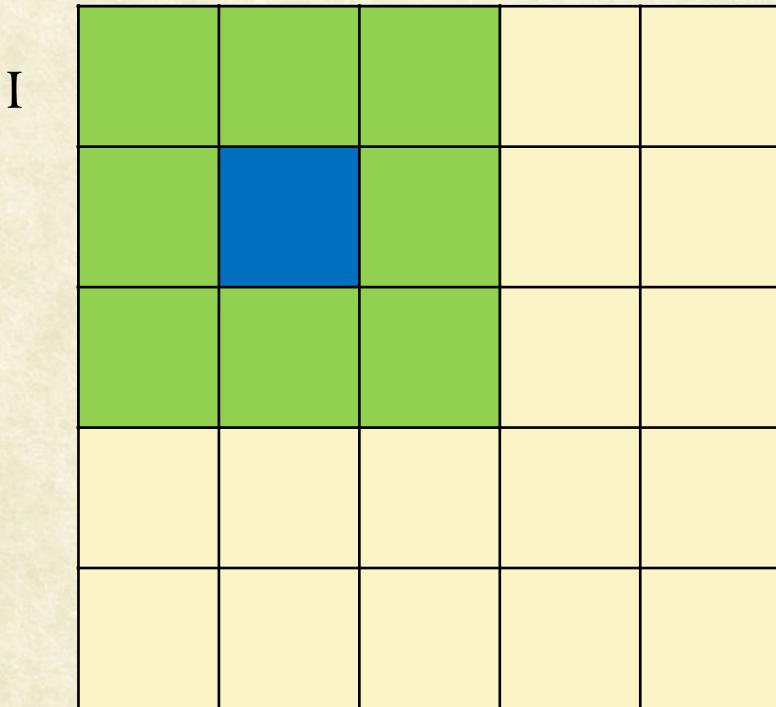
$1/9 *$

1	1	1
1	1	1
1	1	1





# Mean/Average Filter



Note: Coefficients sum to 1

H → 3 × 3

Weight Mask /  
Kernel /  
Filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$$I'(u, v) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^1 \sum_{i=-1}^1 I(u + i, v + j)$$

$$I'(u, v) \leftarrow \sum_{j=-1}^1 \sum_{i=-1}^1 I(u + i, v + j) \cdot H(i, j)$$



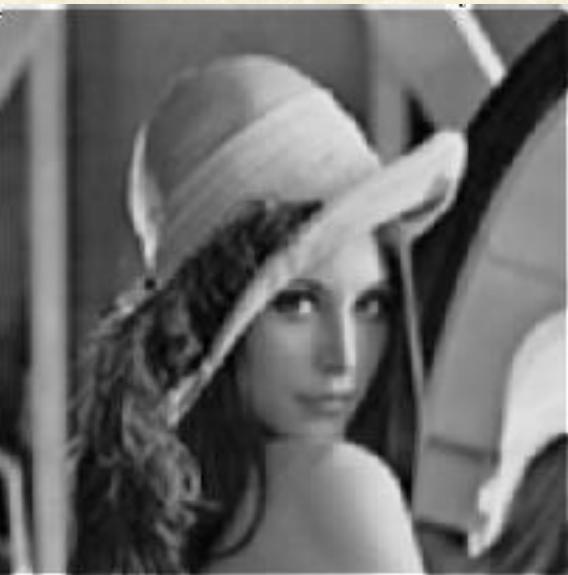
# Effect of Mask Size

Original Image

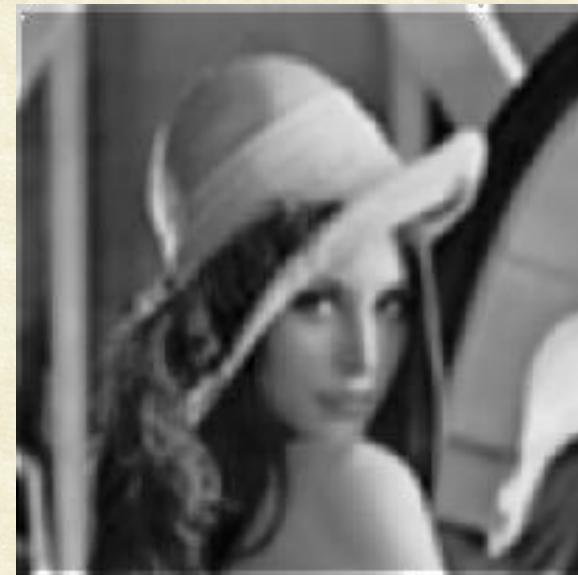
[3x3]



[5x5]



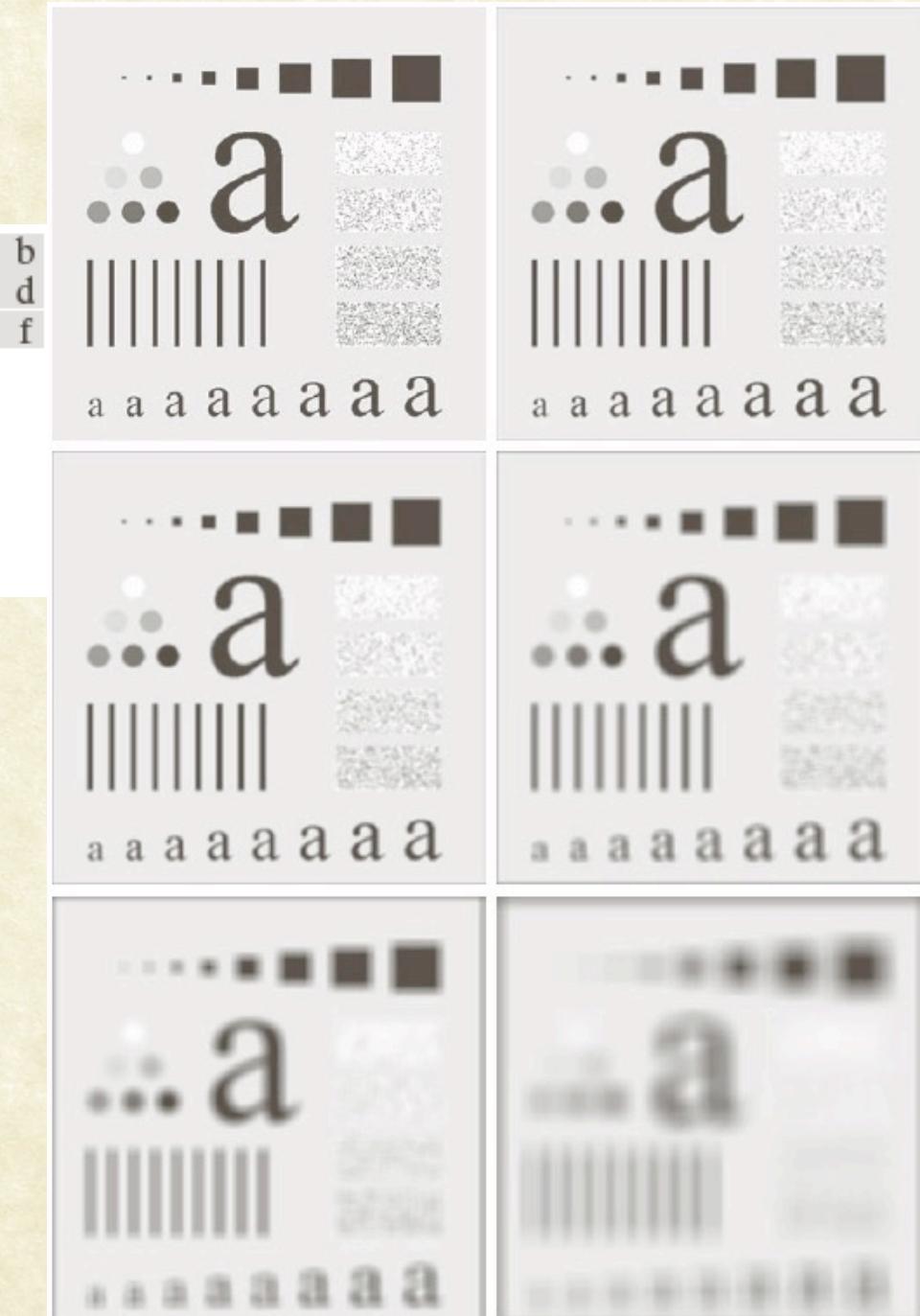
[7x7]





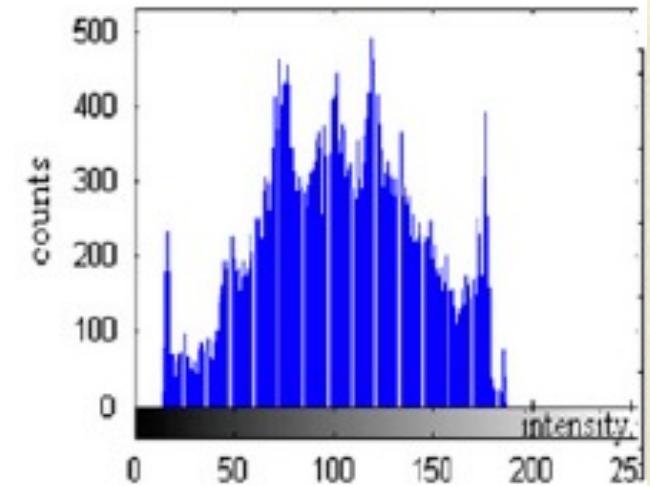
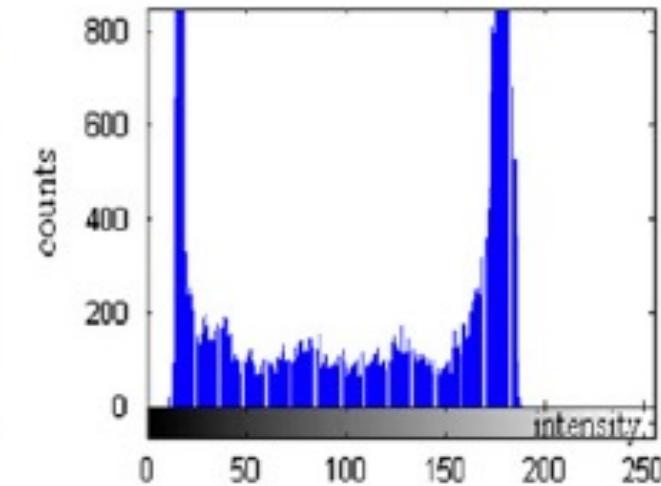
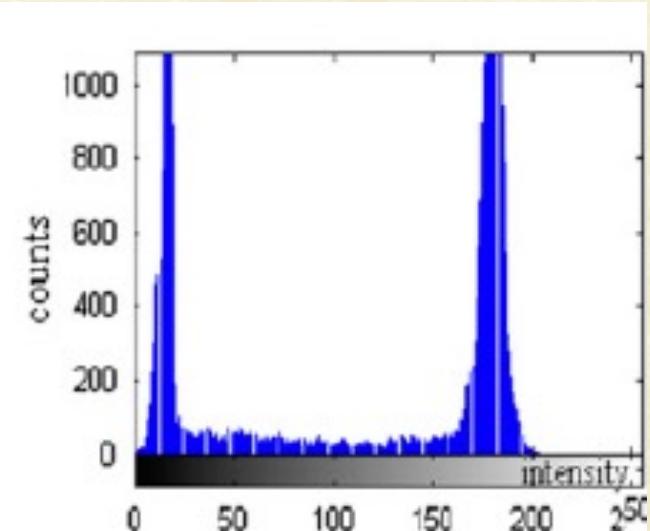
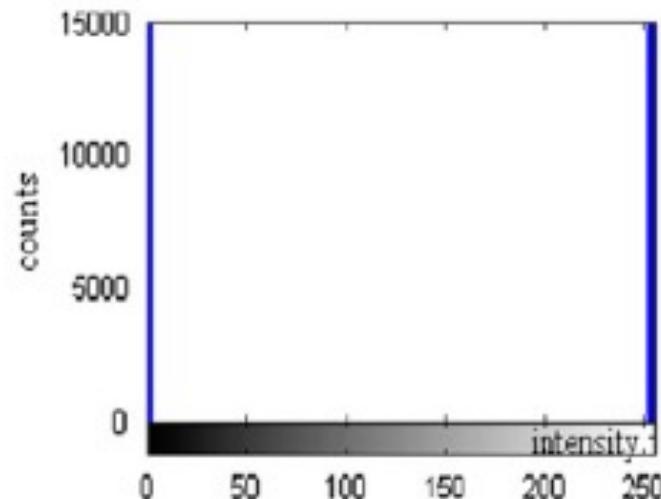
# Square averaging filter

**FIGURE 3.33** (a) Original image, of size  $500 \times 500$  pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes  $m = 3, 5, 9, 15$ , and  $35$ , respectively. The black squares at the top are of sizes  $3, 5, 9, 15, 25, 35, 45$ , and  $55$  pixels; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size  $50 \times 120$  pixels.





# Averaging – a histogram perspective





# Repeated Averaging Using Same Filter



Before



After



After repeated  
averaging

NOTE: Can get the effect of larger filters by smoothing repeatedly with smaller filters



# Weighted Averaging

$\frac{1}{9} \times$				

Standard average

Weighted Average

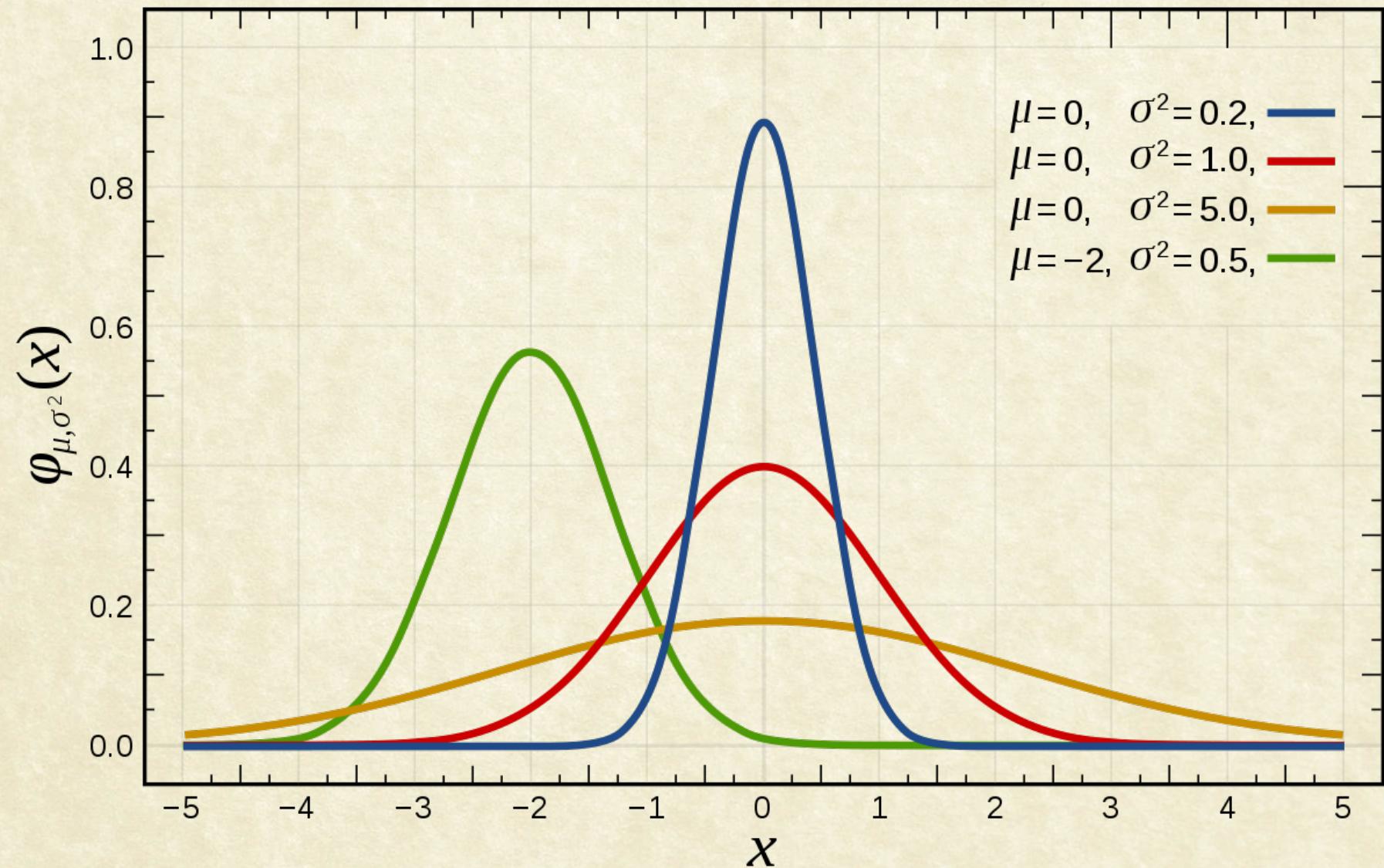
	2	
2	4	2
	2	

$$I'(u, v) \leftarrow \frac{\sum_{j=-a}^a \sum_{i=-b}^b I(u + i, v + j) \cdot H(i, j)}{\sum_{j=-a}^a \sum_{i=-b}^b H(i, j)}$$



# Gaussian Function (1-D)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

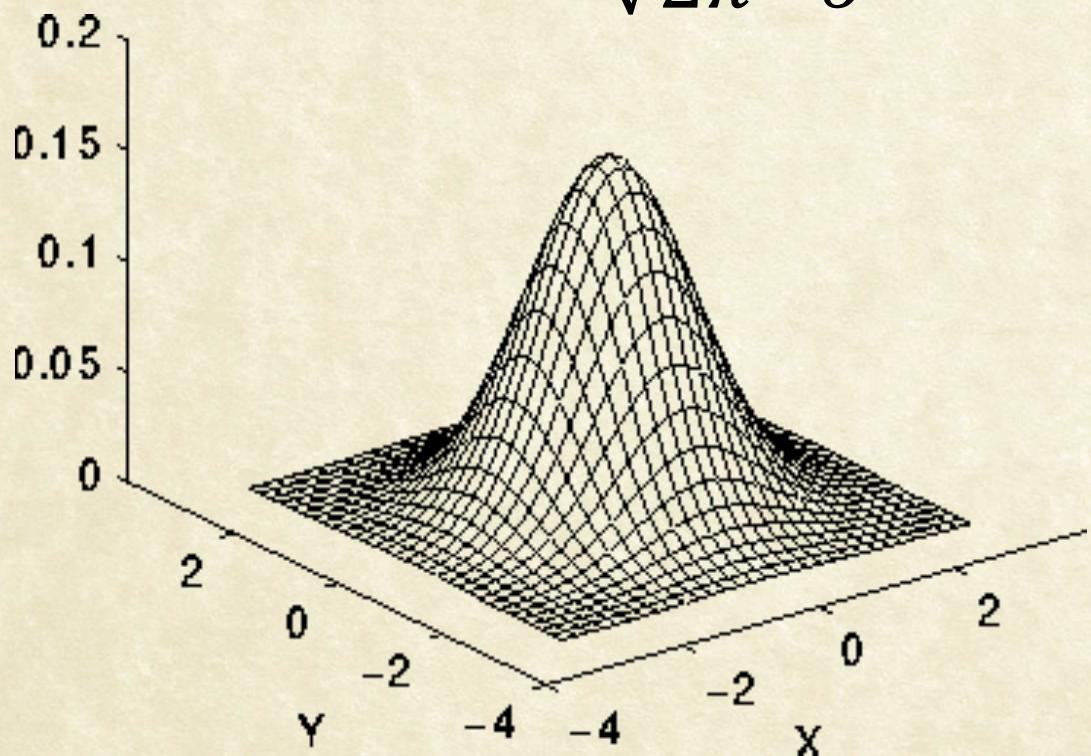




# Gaussian Smoothing

- ▶ Mask weights are samples of a zero-mean 2-D Gaussian

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



$$\frac{1}{265}$$

1	4	6	4	1
4	16	26	16	4
6	26	43	26	6
4	16	26	16	4
1	4	6	4	1

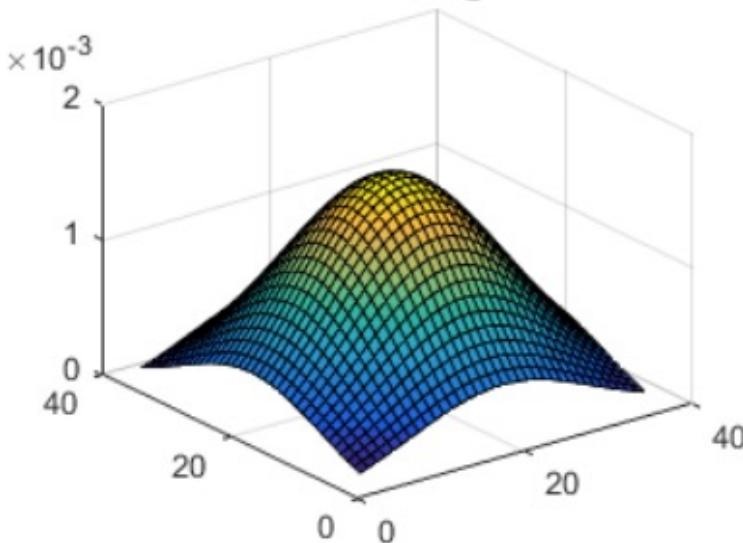
5×5 Gaussian filter,  $\sigma = 1$



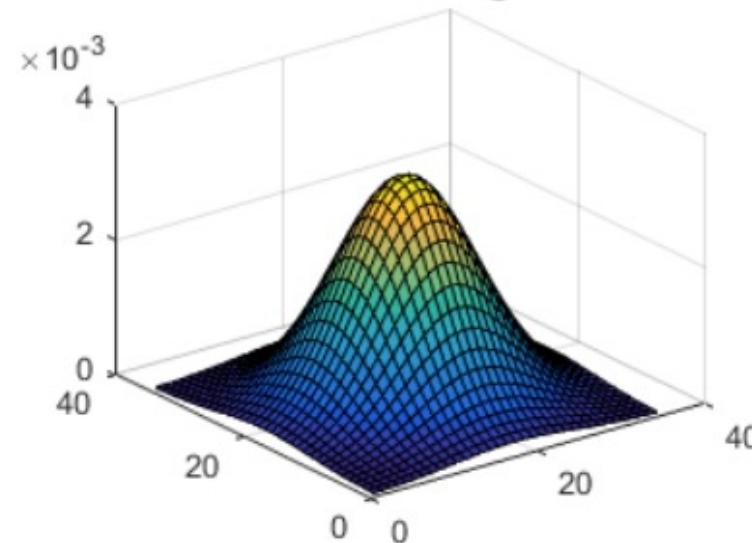
# Gaussian Smoothing – Effect of sigma

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

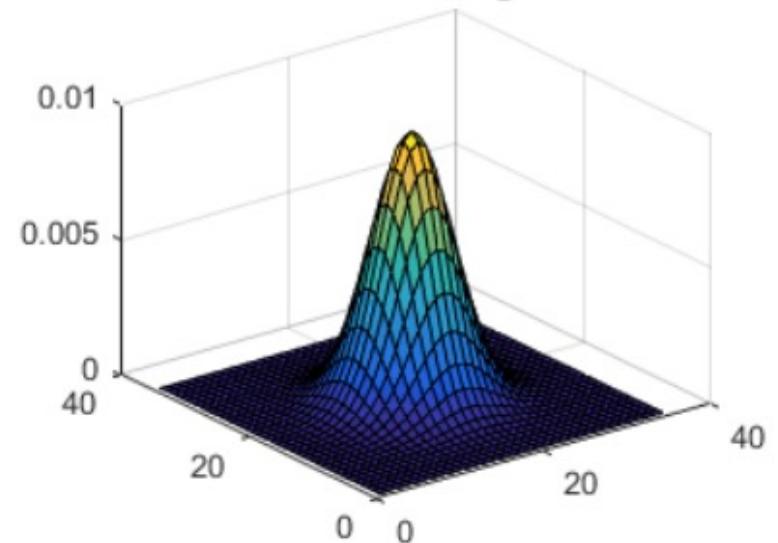
filter size = 35, sigma = 11



filter size = 35, sigma = 7



filter size = 35, sigma = 2





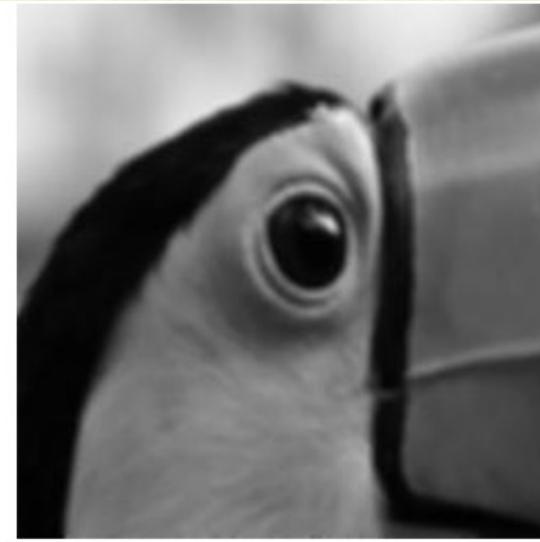
# Gaussian Smoothing – Effect of sigma



Original Image  
(Sigma 0)

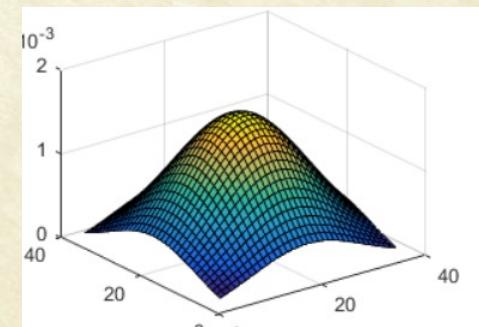
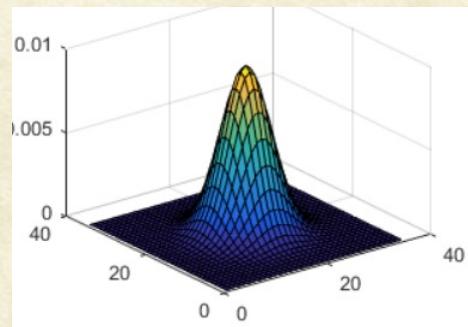


Gaussian Blur  
(Sigma 0.7)



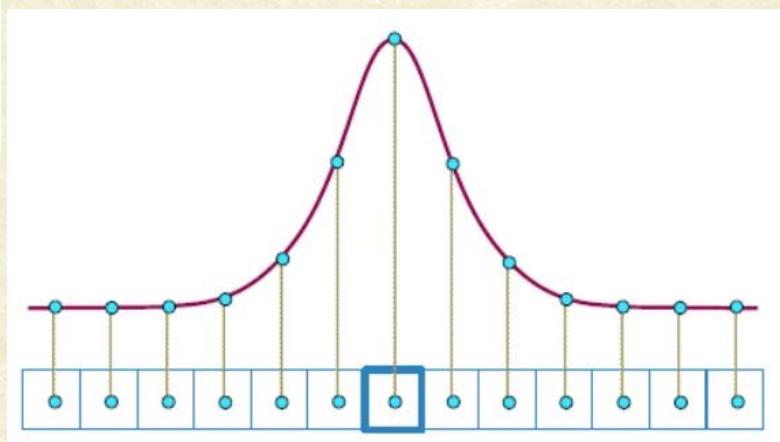
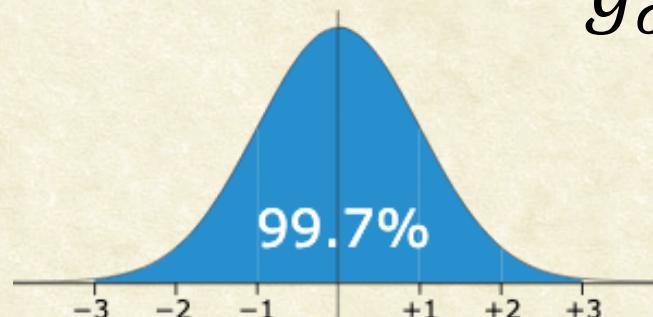
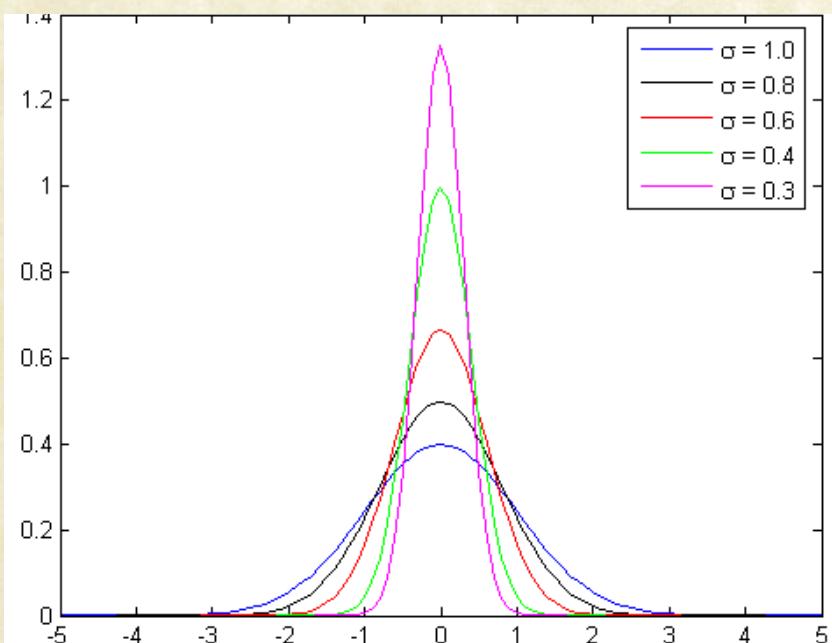
Gaussian Blur  
(Sigma 2.8)

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





# Computing Gaussian filter coefficients

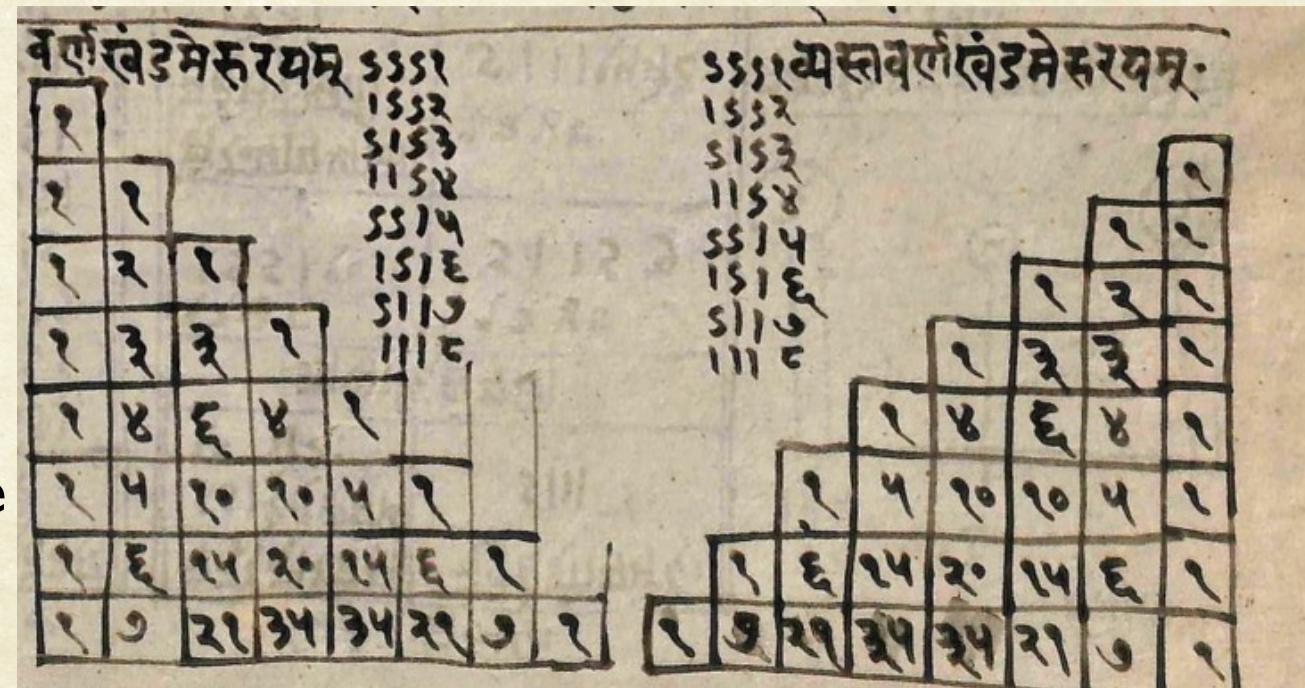


$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\left(\frac{x^2}{2\sigma^2}\right)}$$



# Computing Gaussian filter coefficients

Index N	Coefficients							Sum = $2^N$						
0		1						1						
1			1	1				2						
2				1	2	1		4						
3					1	3	3	8						
4						1	4	16						
5							1	32						
6								64						
7								128						
8								256						
9								512						
10								1024						
11								2048						
12	1	12	66	220	495	792	924	792	495	220	66	12	1	4096



Meru Prastaara, derived from Pingala's formulae  
(2 BCE), Manuscript from Raghunath  
Temple Library, Jammu



# Computing Gaussian filter coefficients

Index N

Coefficients

Sum=2<sup>N</sup>

0	1	1									
1	1	1									
2	1	2	1								
3	1	3	3	1							
4	1	4	6	4	1						
5	1	5	10	10	5	1					
6	1	6	15	20	15	6	1				
7	1	7	21	35	35	21	7	1			
8	1	8	28	56	70	56	28	8	1		
9	1	9	36	84	126	126	84	36	9	1	
10	1	10	45	120	210	252	210	120	45	10	1

1

2

4

8

16

32

64

128

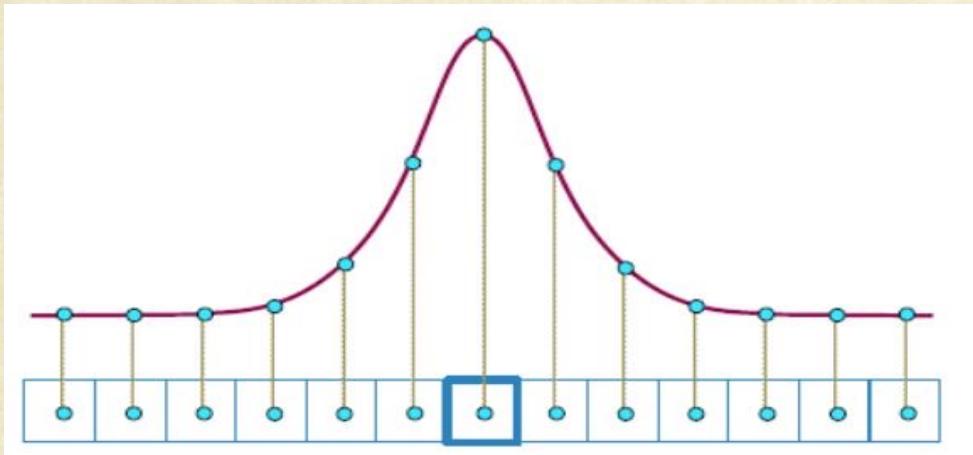
256

512

1024

e.g., s = 7×7

$$\frac{1}{64} \times \begin{matrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{matrix}$$



$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\left(\frac{x^2}{2\sigma^2}\right)}$$



# Computing Gaussian filter coefficients

Index N

Coefficients

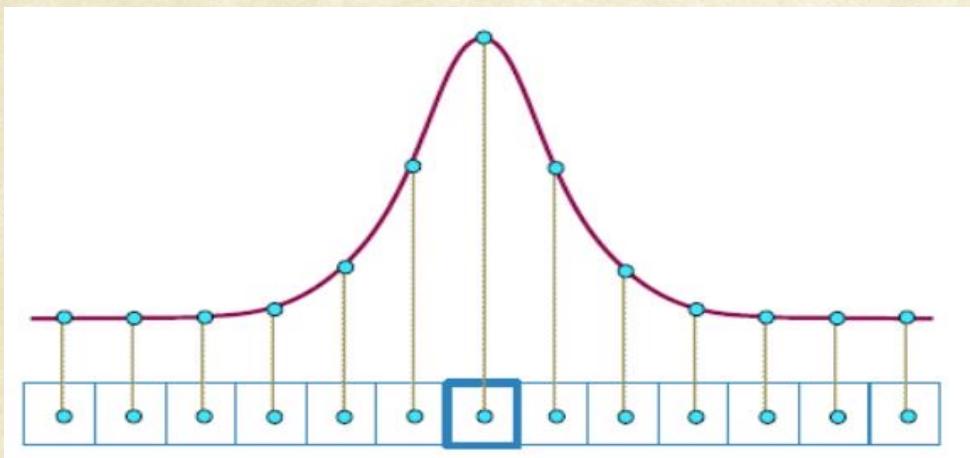
Sum=2<sup>N</sup>

0	1	1									
1	1	1									
2	1	2	1								
3	1	3	3	1							
4	1	4	6	4	1						
5	1	5	10	10	5	1					
6	1	6	15	20	15	6	1				
7	1	7	21	35	35	21	7	1			
8	1	8	28	56	70	56	28	8	1		
9	1	9	36	84	126	126	84	36	9	1	
10	1	10	45	120	210	252	210	120	45	10	1

e.g., s = 7×7

1	6	15	20	15	6	1
6	36	90	120	90	36	6
15	90	225	300	225	90	15
20	120	300	400	300	120	20
15	90	225	300	225	90	15
6	36	90	120	90	36	6
1	6	15	20	15	6	1

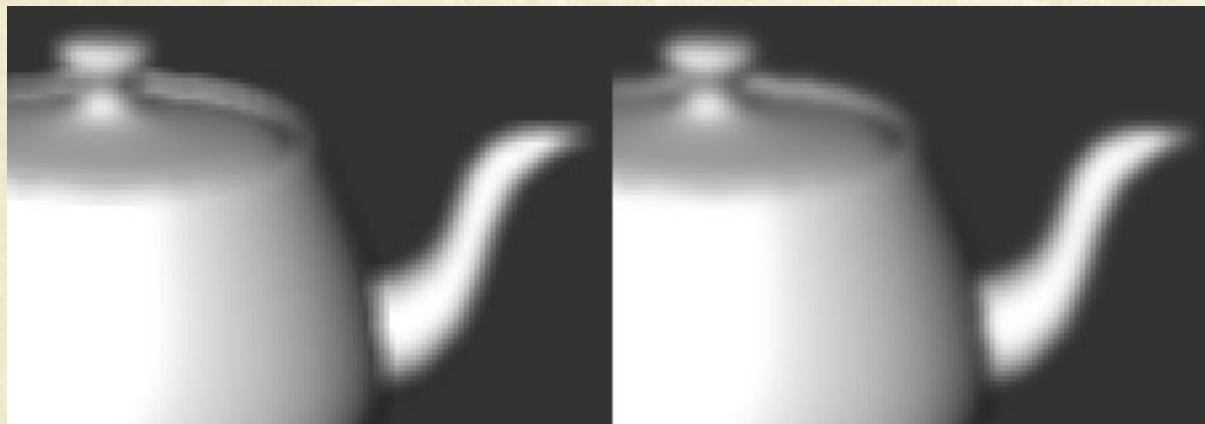
$$\frac{1}{4096} \times$$



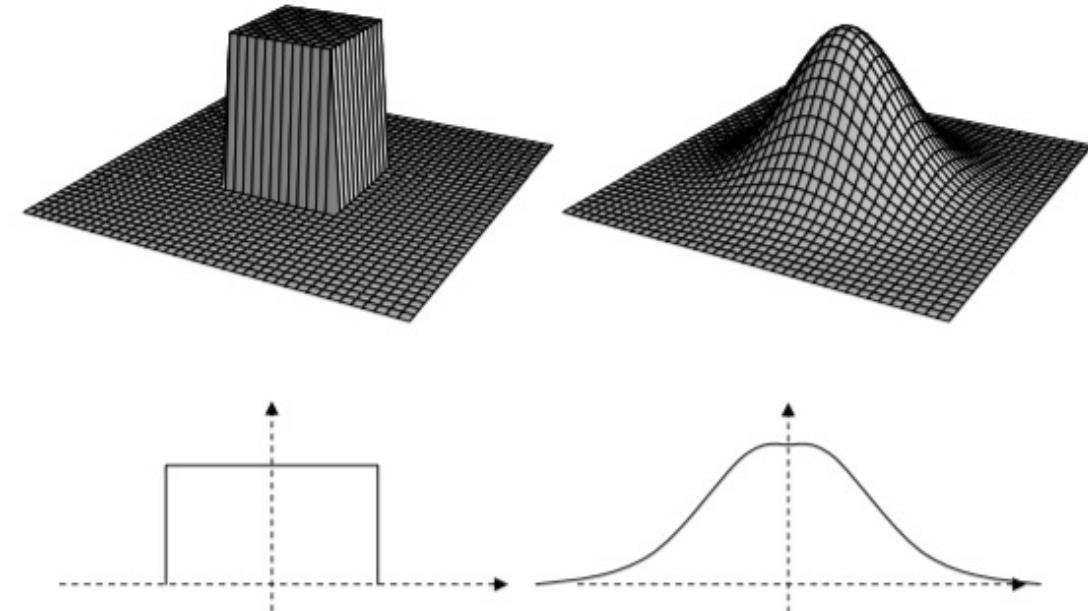
$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\left(\frac{x^2}{2\sigma^2}\right)}$$



# Averaging vs. Gaussian Filter



Smoothened intensity transitions

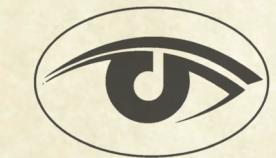


0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

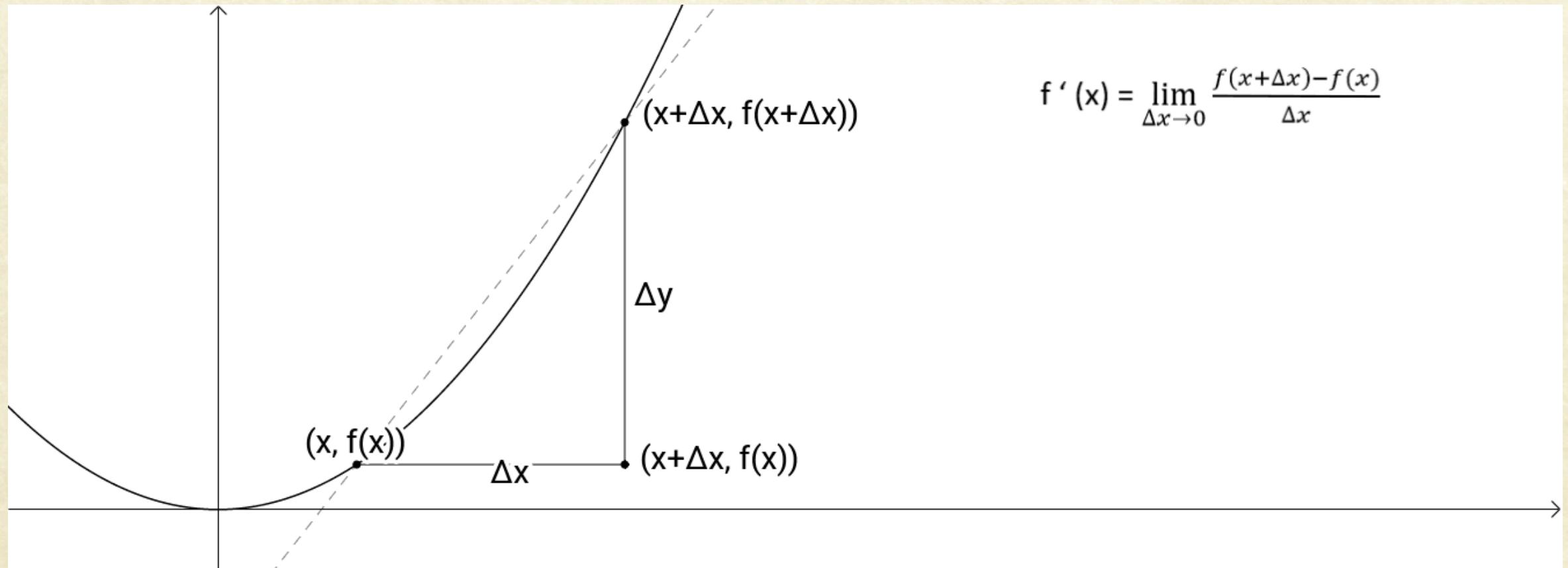
(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

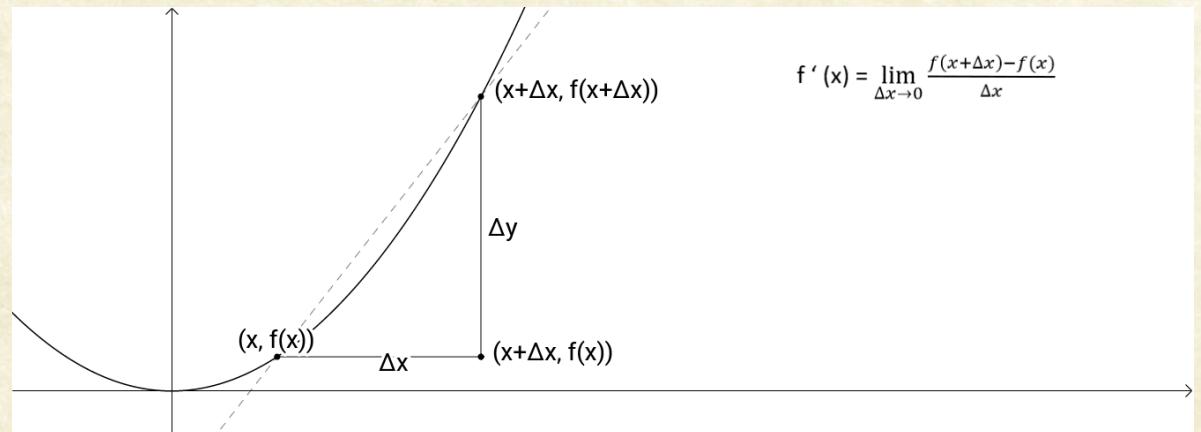
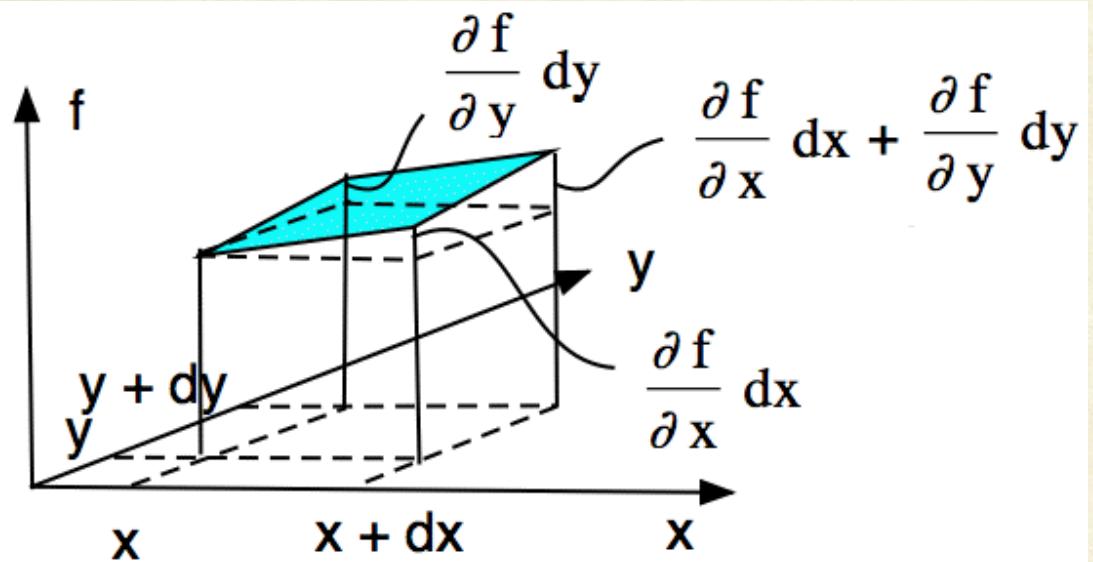


# Recap: Derivatives





# Recap: Derivatives

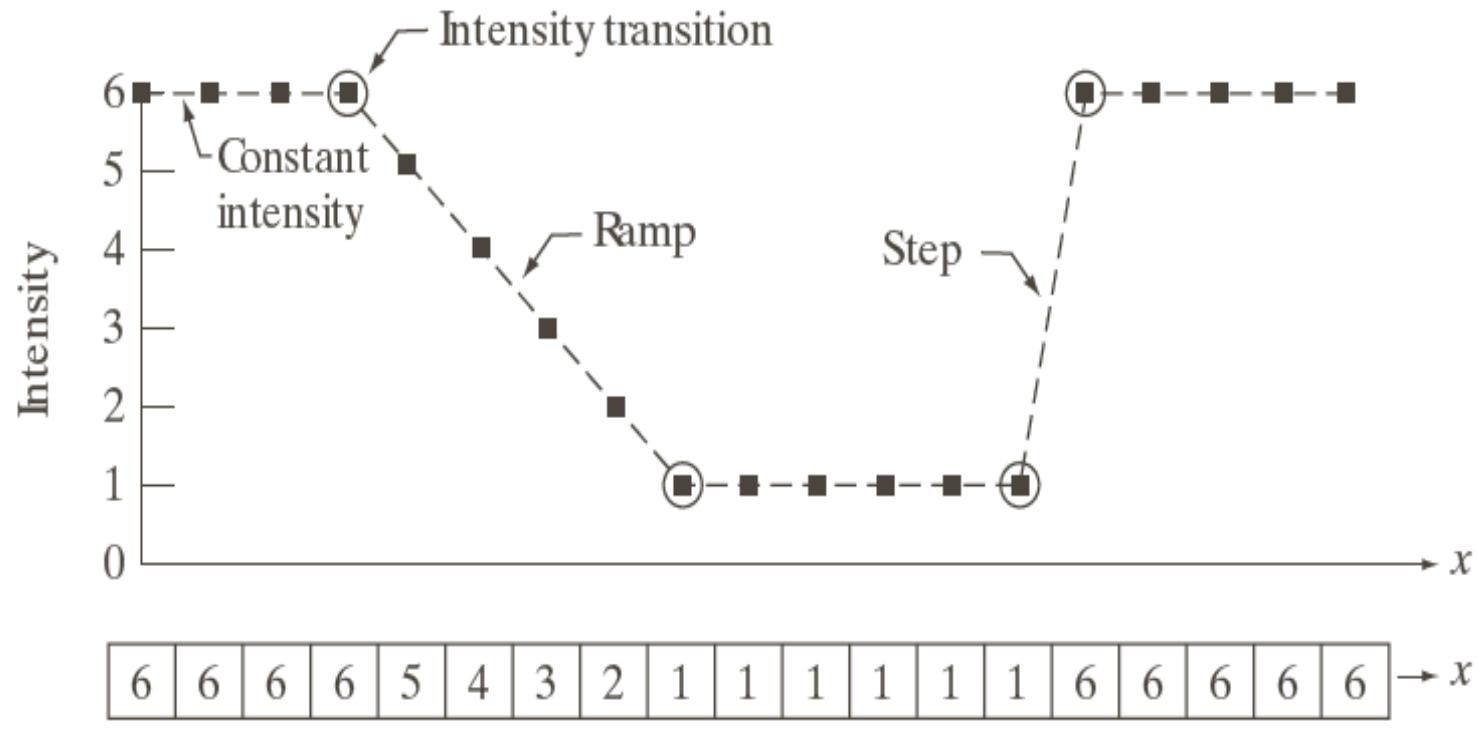


$$\frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$\frac{\partial f(x, y)}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$



## First Derivative (Digital approximation)



$$\frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$\frac{\partial f(x, y)}{\partial x} \sim f[x + 1, y] - f[x, y]$$

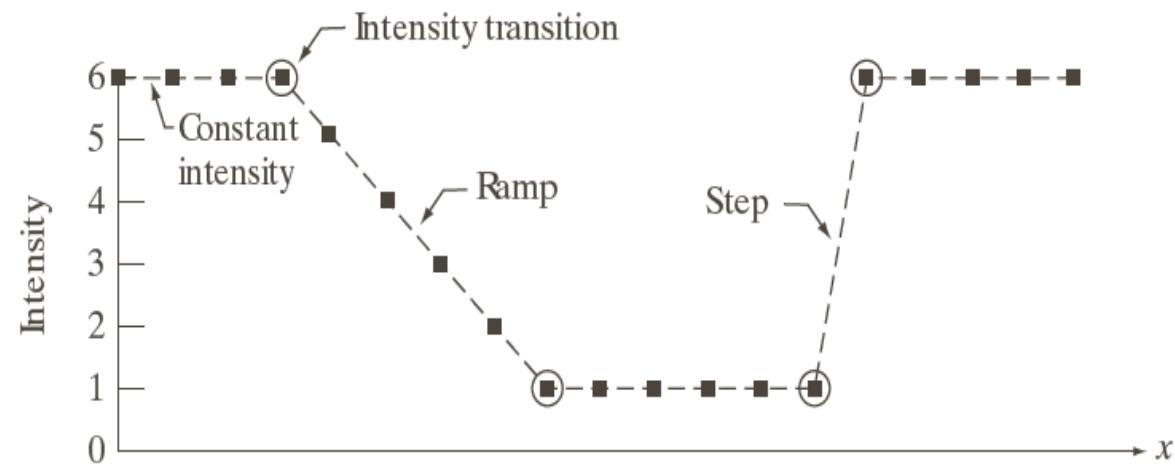


## Second Derivative (Digital approximation)

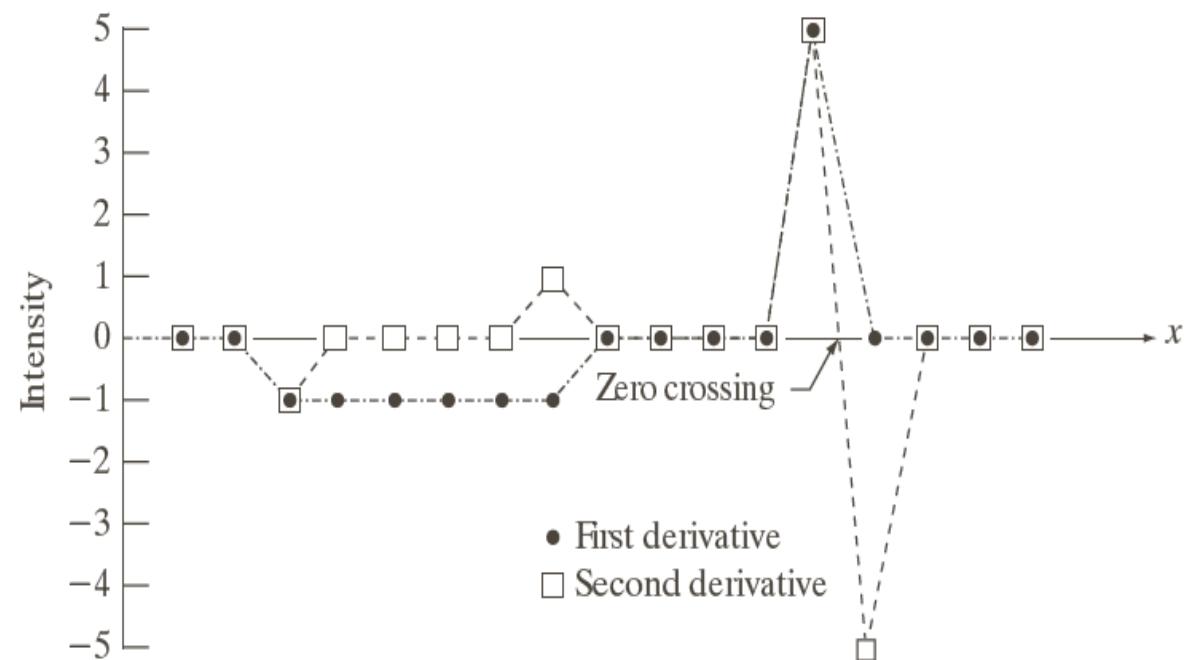
$$\frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

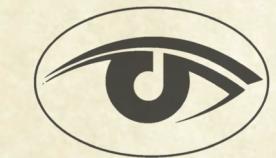
$$\frac{\partial f(x, y)}{\partial x} \sim f[x + 1, y] - f[x, y]$$

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



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6	$\rightarrow x$
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	5	0	0	0	0	
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	5	-5	0	0	0	





## Alt: Derivative as Symmetric Difference

$$\frac{\partial f(x, y)}{\partial x} \sim f[x + 1, y] - f[x, y]$$

$$\frac{\partial f(x, y)}{\partial x} \sim \frac{f[x + h, y] - f[x - h, y]}{2h}$$



# Image Gradient and Edges

$$\frac{f(x+h,y) - f(x-h,y)}{2h}$$

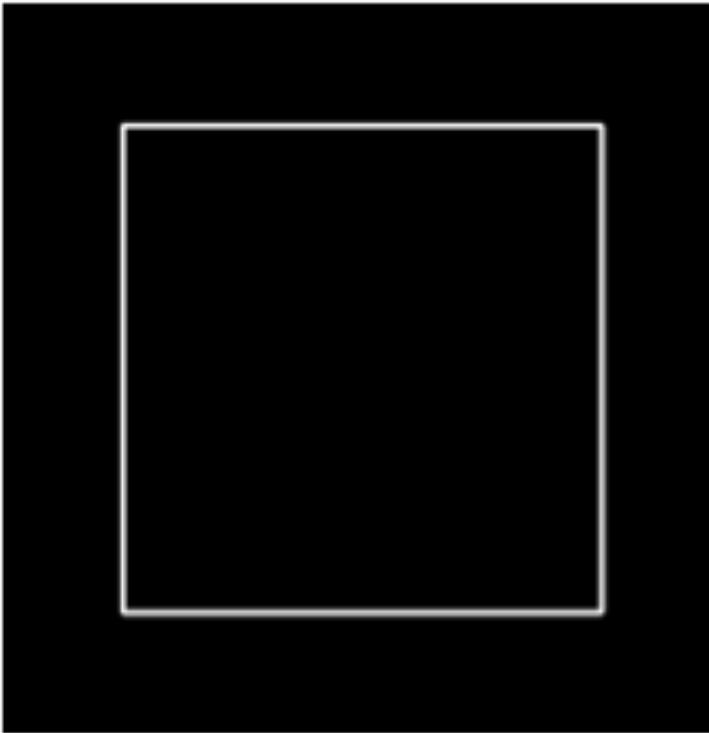
-1	0	1
----	---	---

x-derivative

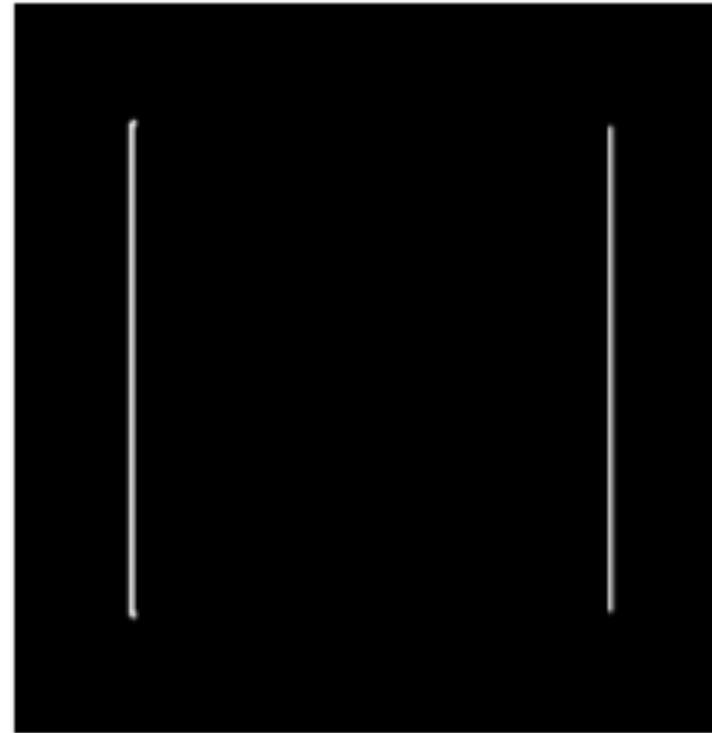
$$\frac{f(x,y+h) - f(x,y-h)}{2h}$$

-1
0
1

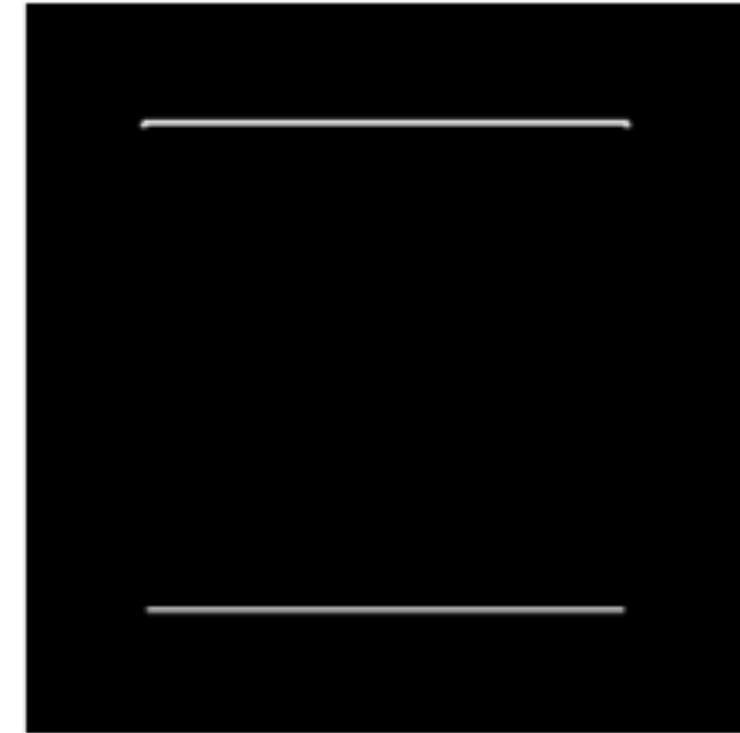
y-derivative



**Image**



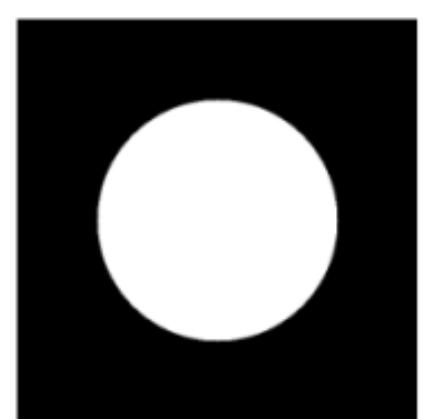
**Gradient in x**



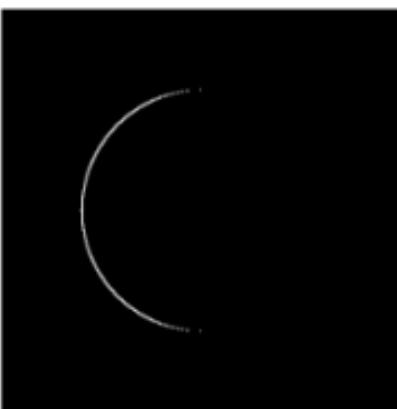
**Gradient in y**



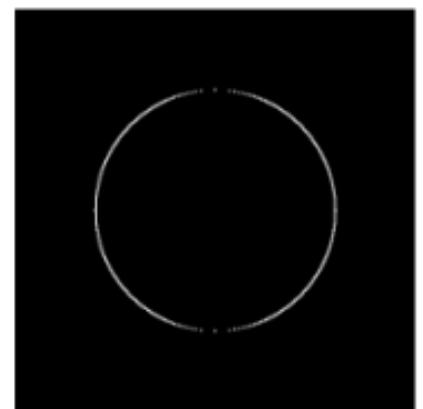
# Edge 'Image'



Input image



8u



64f → 8u

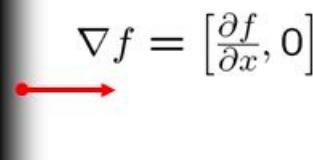


# Image Gradient

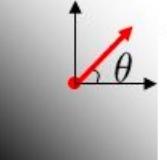
The gradient of an image:

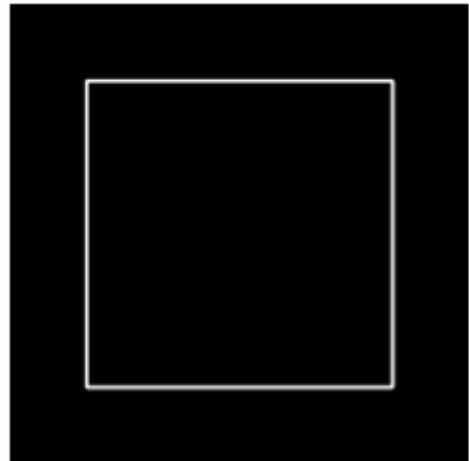
$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

The gradient points in the direction of most rapid change in intensity

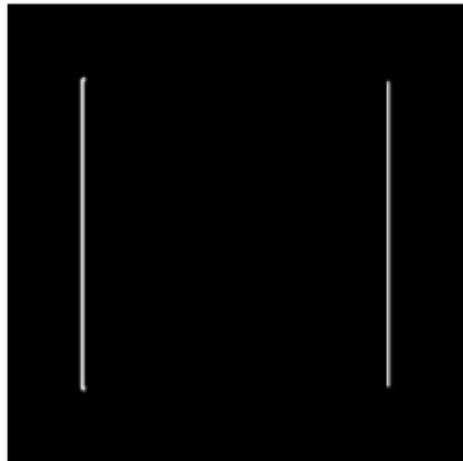

$$\nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right]$$


$$\nabla f = \left[ 0, \frac{\partial f}{\partial y} \right]$$


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



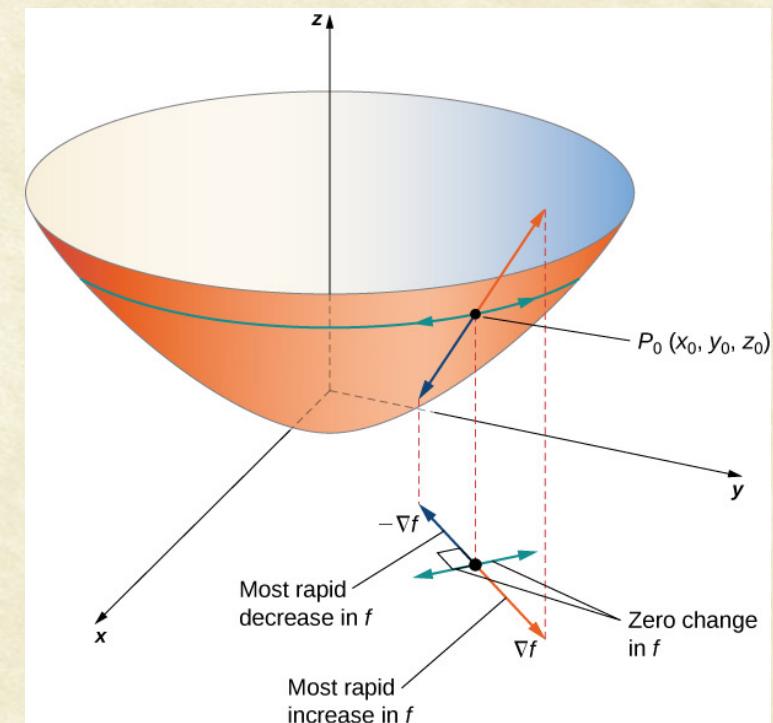
Image



Gradient in x



Gradient in y





Dr. Prewitt

<https://nihrecord.nih.gov/sites/recordNIH/files/pdf/1984/NIH-Record-1984-03-13.pdf>

# Prewitt Edge Filter

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

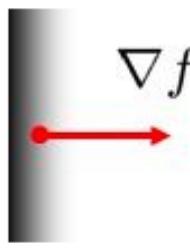
$G_x$

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

$G_y$



# Edge is perpendicular to gradient

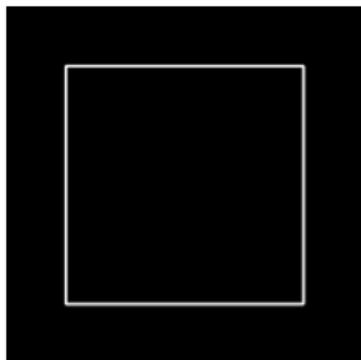

$$\nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right]$$


$$\nabla f = \left[ 0, \frac{\partial f}{\partial y} \right]$$

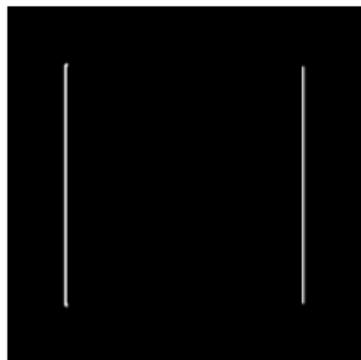
-1	0	+1
-1	0	+1
-1	0	+1

 $G_x$ 

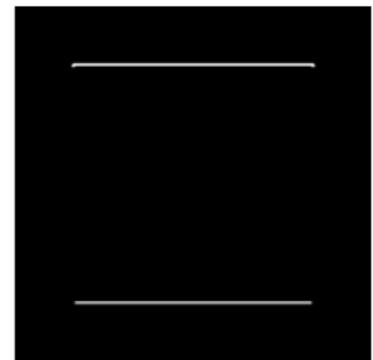
+1	+1	+1
0	0	0
-1	-1	-1

 $G_y$ 

Image



Gradient in x



Gradient in y



# Edge is perpendicular to gradient

$$\nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right]$$

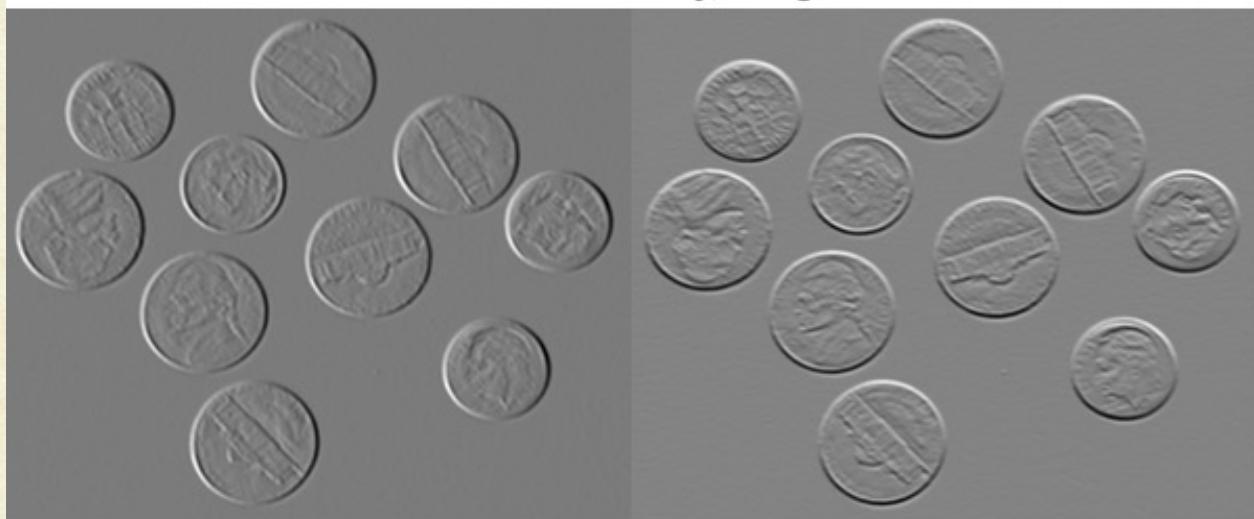
$$\nabla f = \left[ 0, \frac{\partial f}{\partial y} \right]$$

$$\begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$

$\mathbf{G}_x$

$$\begin{array}{|c|c|c|}\hline +1 & +1 & +1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$\mathbf{G}_y$





# 2-D Laplacian Filter

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

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

0	1	0
1	-4	1
0	1	0

(By me) Refer this video for more info [https://youtu.be/E1E\\_XcYxvCY](https://youtu.be/E1E_XcYxvCY)



# Edge Masks: Sobel, Laplacian

Original



Laplacian



0	-1	0
-1	4	-1
0	-1	0

Note: Coefficients sum to 0

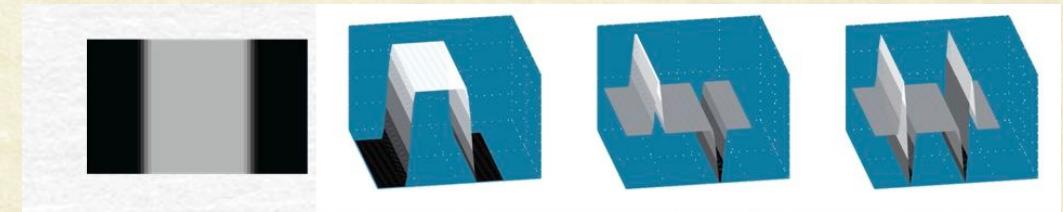
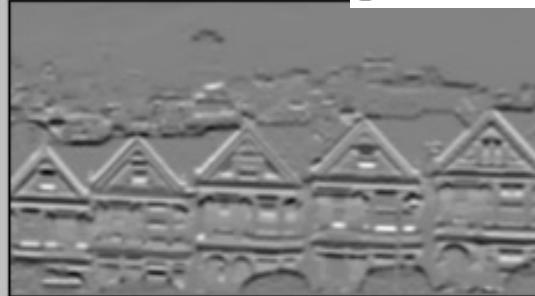
Sobel X

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



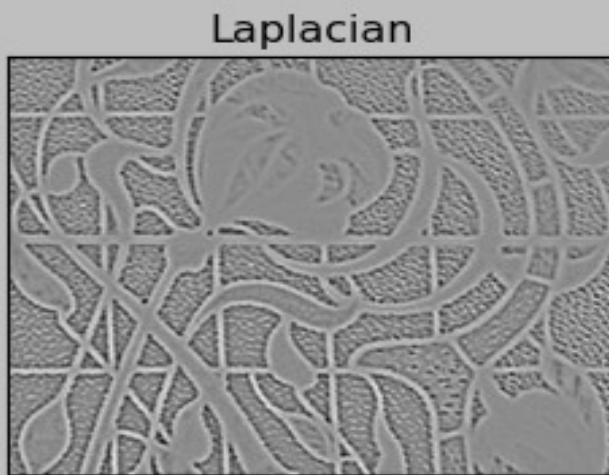
Sobel Y

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





# Edge Masks: Sobel, Laplacian



0	-1	0
-1	4	-1
0	-1	0



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

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

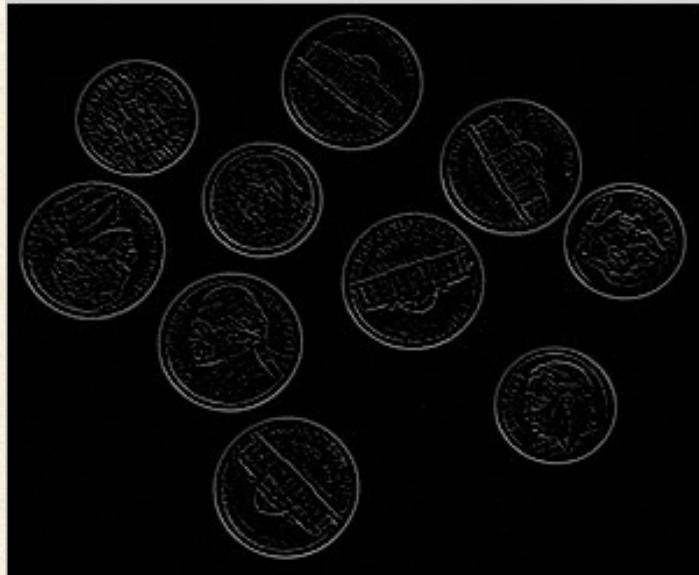


# Image Sharpening

$$I(u, v)$$

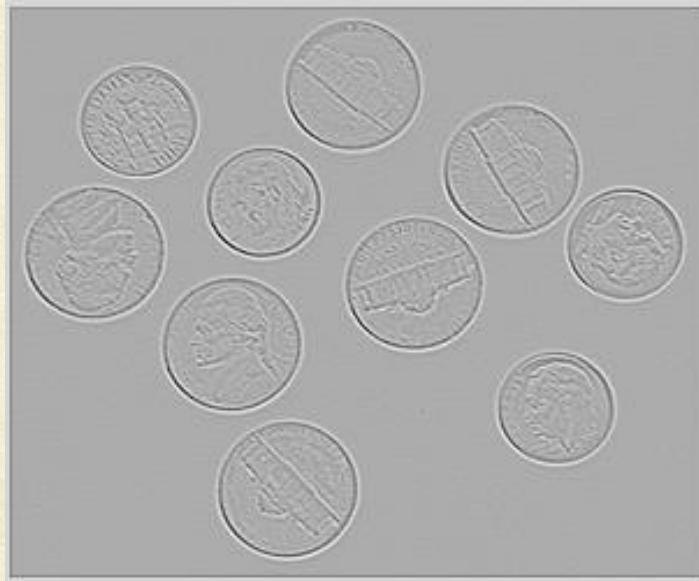


$$\nabla^2 I(u, v)$$

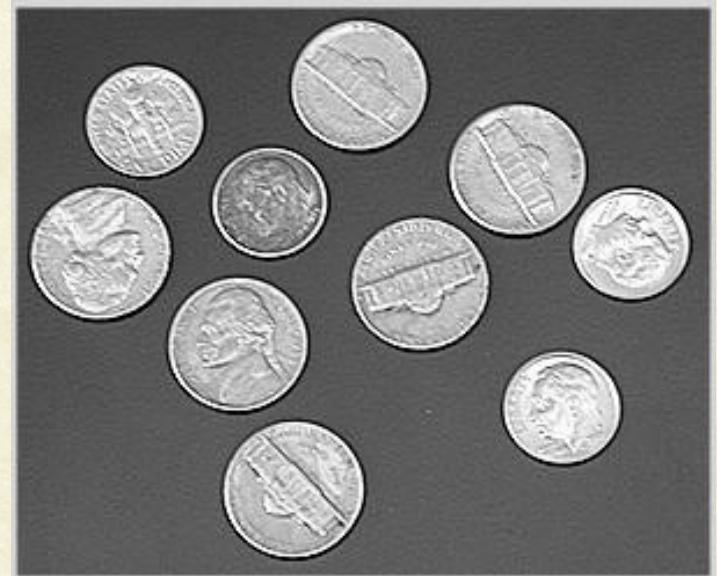


$$\nabla^2 I(u, v) + 128$$

(For visualization)



$$I'(u, v) = I(u, v) + \nabla^2 I(u, v)$$





# Sharpening (Unsharp Masking)

$$I(u, v)$$



$$\nabla I(u, v)$$



$$I'(u, v) = I(u, v) + \nabla^2 I(u, v)$$

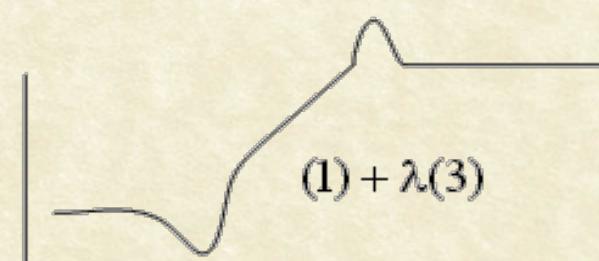
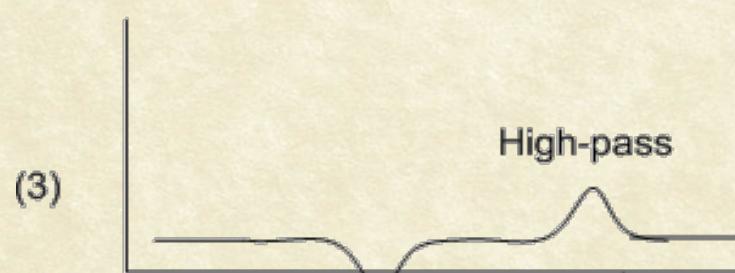
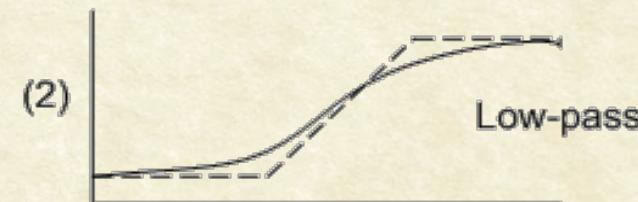
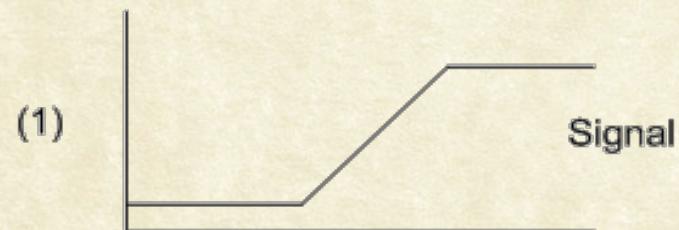


Image Courtesy:NASA



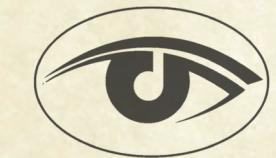
# Highboost Filtering

- What does blurring take away?



- Let's add it back:





# Unsharp Masking vs Highboost Filtering





# USM vs. HBF as Spatial Filters

A=1	A=2																		
$w = 9A - 1$	$w = 17$																		
<table border="1"><tbody><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>w</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></tbody></table>	-1	-1	-1	-1	w	-1	-1	-1	-1	<table border="1"><tbody><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>17</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></tbody></table>	-1	-1	-1	-1	17	-1	-1	-1	-1
-1	-1	-1																	
-1	w	-1																	
-1	-1	-1																	
-1	-1	-1																	
-1	17	-1																	
-1	-1	-1																	

- ▶ If A=1, we get unsharp masking.  $I'(u, v) = I(u, v) + \nabla^2 I(u, v)$
- ▶ If A>1, original image is added back to detail image (HBF).



Questions?