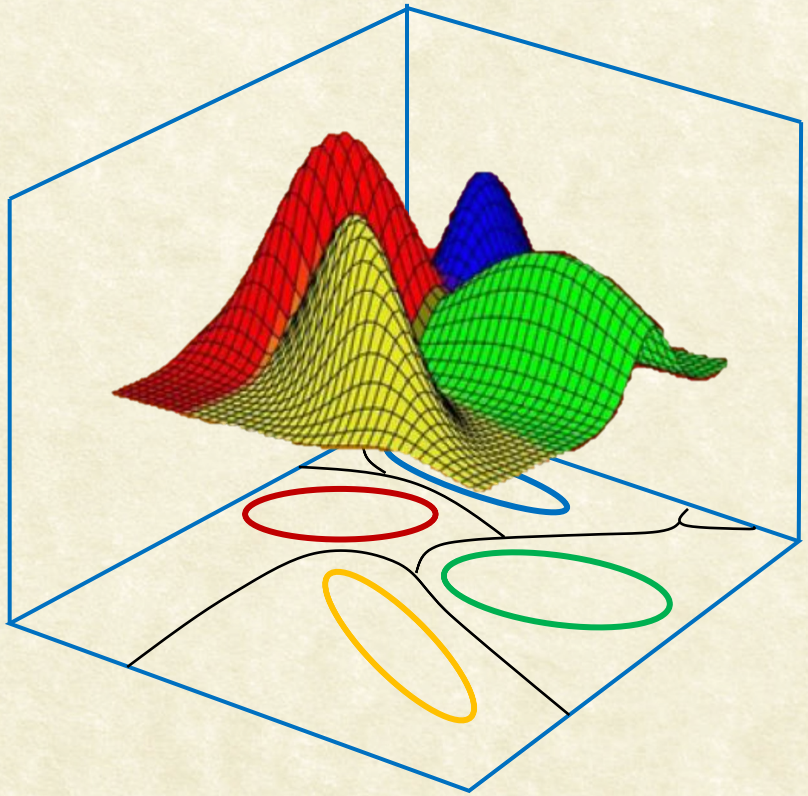
CS7.404: Digital Image Processing

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

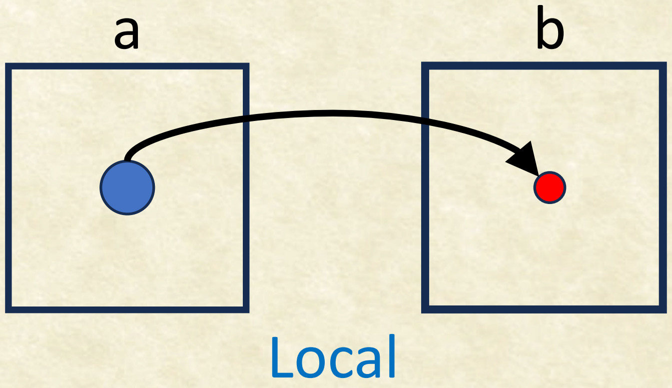
3 approaches

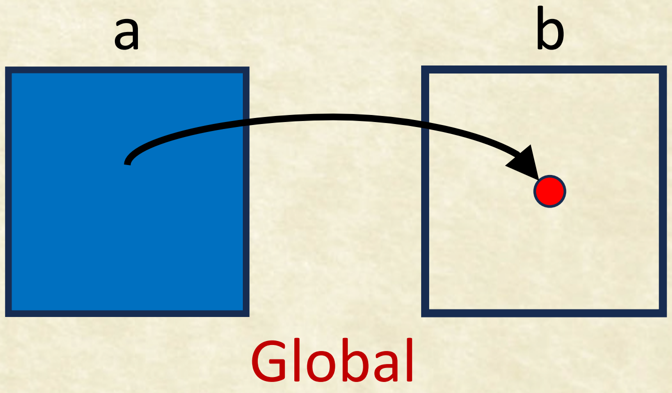
1. Point to Point
2. Neighbourhood to Point
3. Global to Point

48	219	168	145	244	188	120	8
49	218	87	94	133	35	17	148
174	151	74	179	224	3	252	194
77	127	87	139	44	228	149	135
138	229	136	113	250	51	108	163
38	210	185	177	69	76	131	53
178	164	79	158	64	169	85	97
96	209	214	203	223	73	110	200

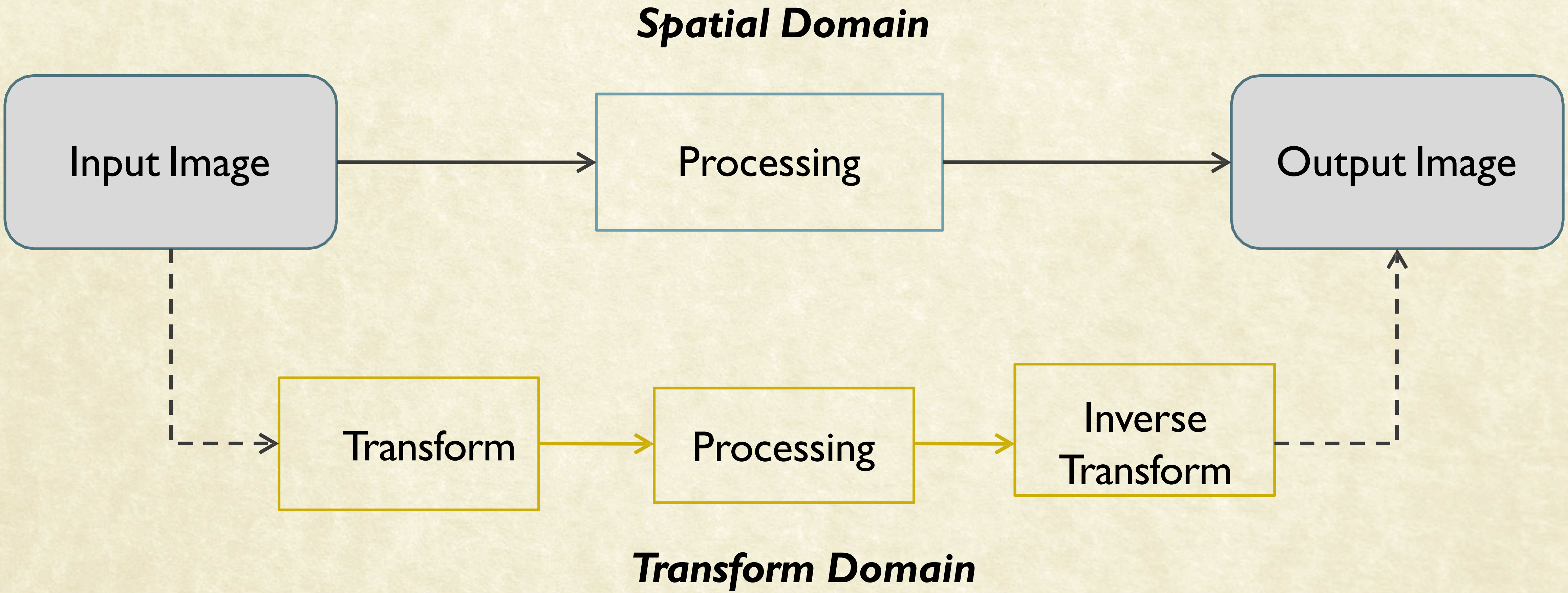
Current Pixel

3x5 Neighborhood





Spatial vs. Transform Domain Processing



Spatial vs. Transform Domain Processing

Bandhani / Bandhej

Tie & Dye

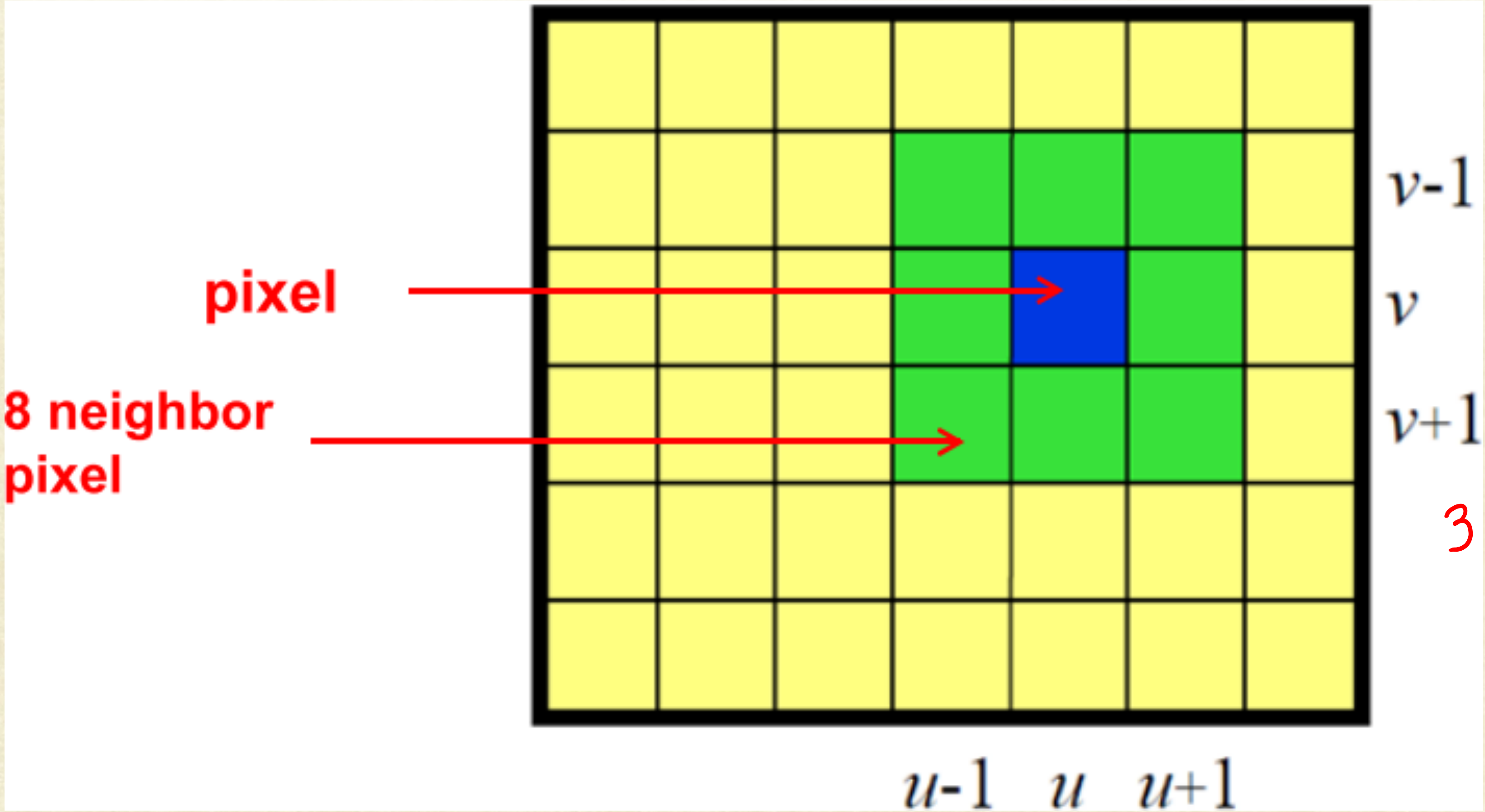
Spatial vs. Transform Domain Processing

Transform (Tie)

Process (Dye)

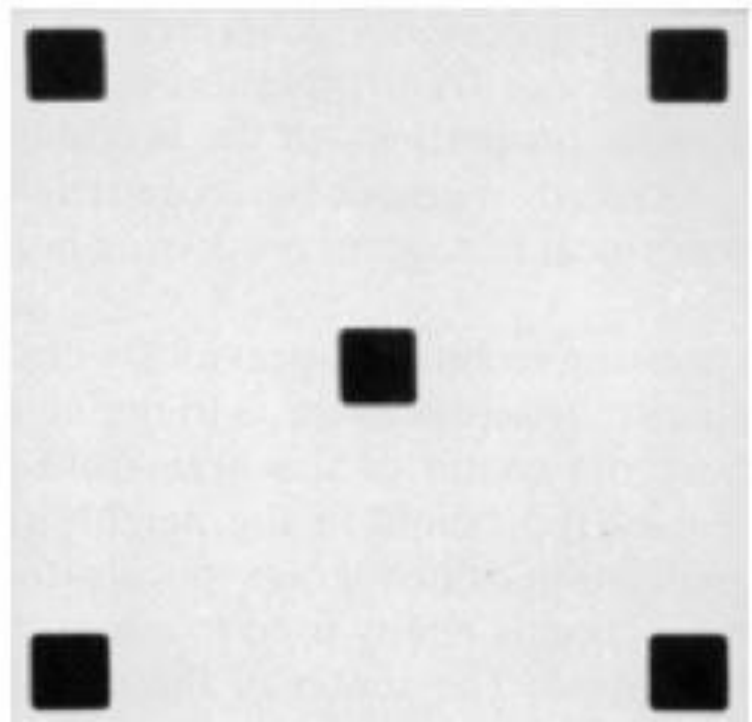
Inverse Transform (Untie)

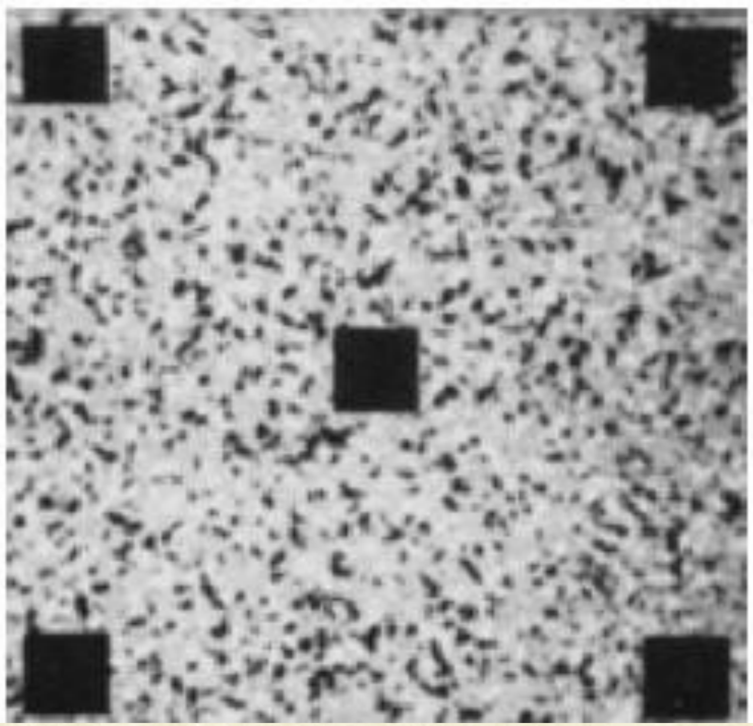
Pixel Neighborhood



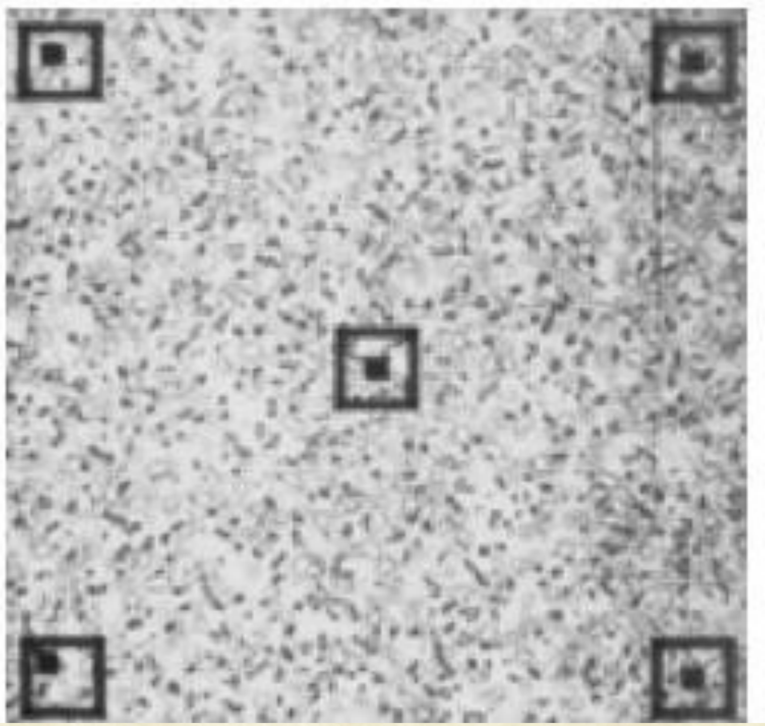
3x3

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

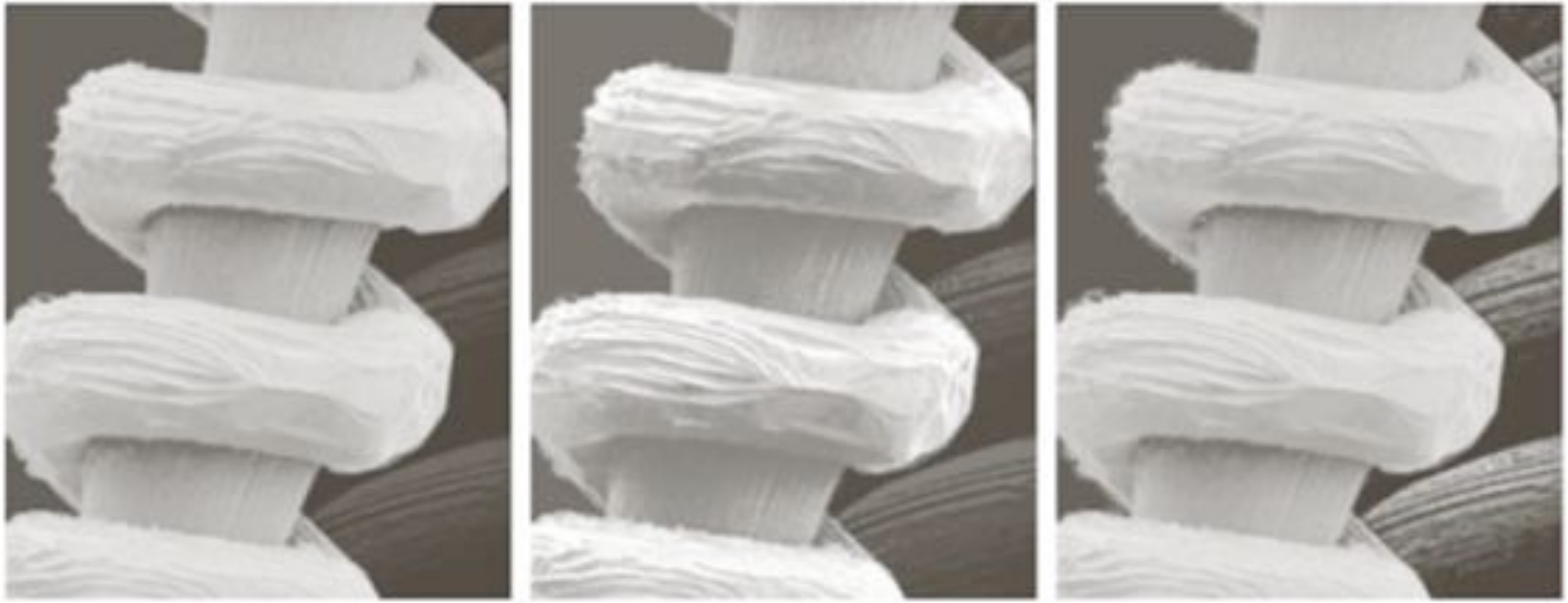
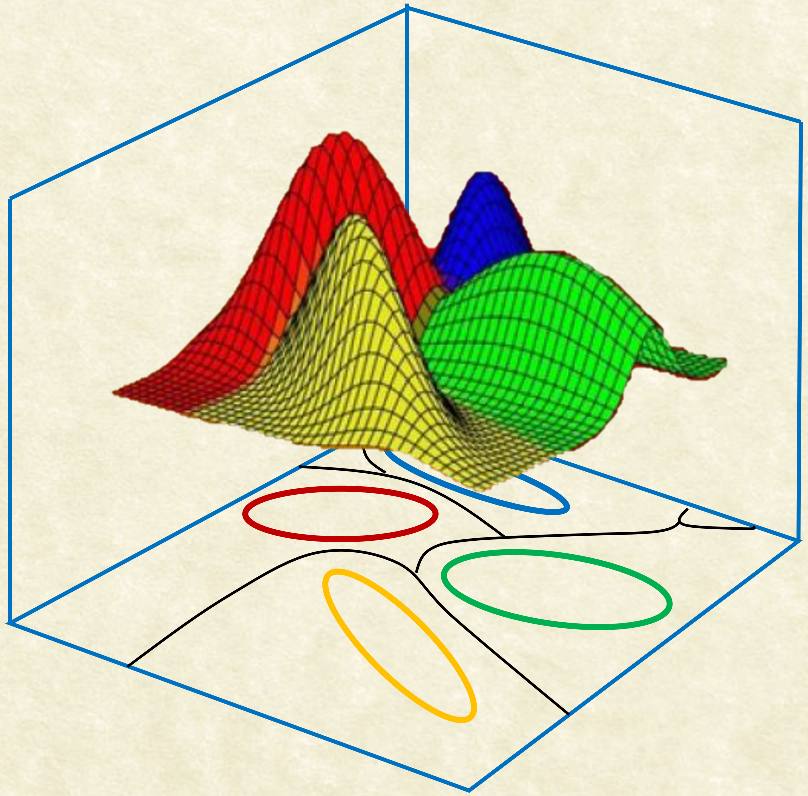


FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

CS7.404: Digital Image Processing

Monsoon 2023: Spatial Filters



Anoop M. Namboodiri

Biometrics and Secure ID Lab, CVIT,

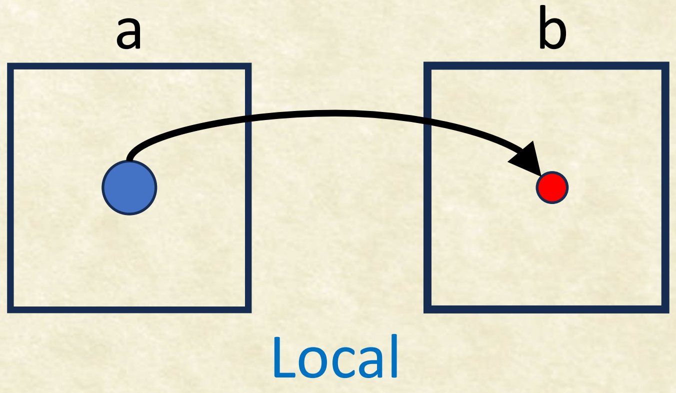
IIIT Hyderabad

Spatial Domain Processing

- **Manipulating Pixels Directly in Spatial Domain**

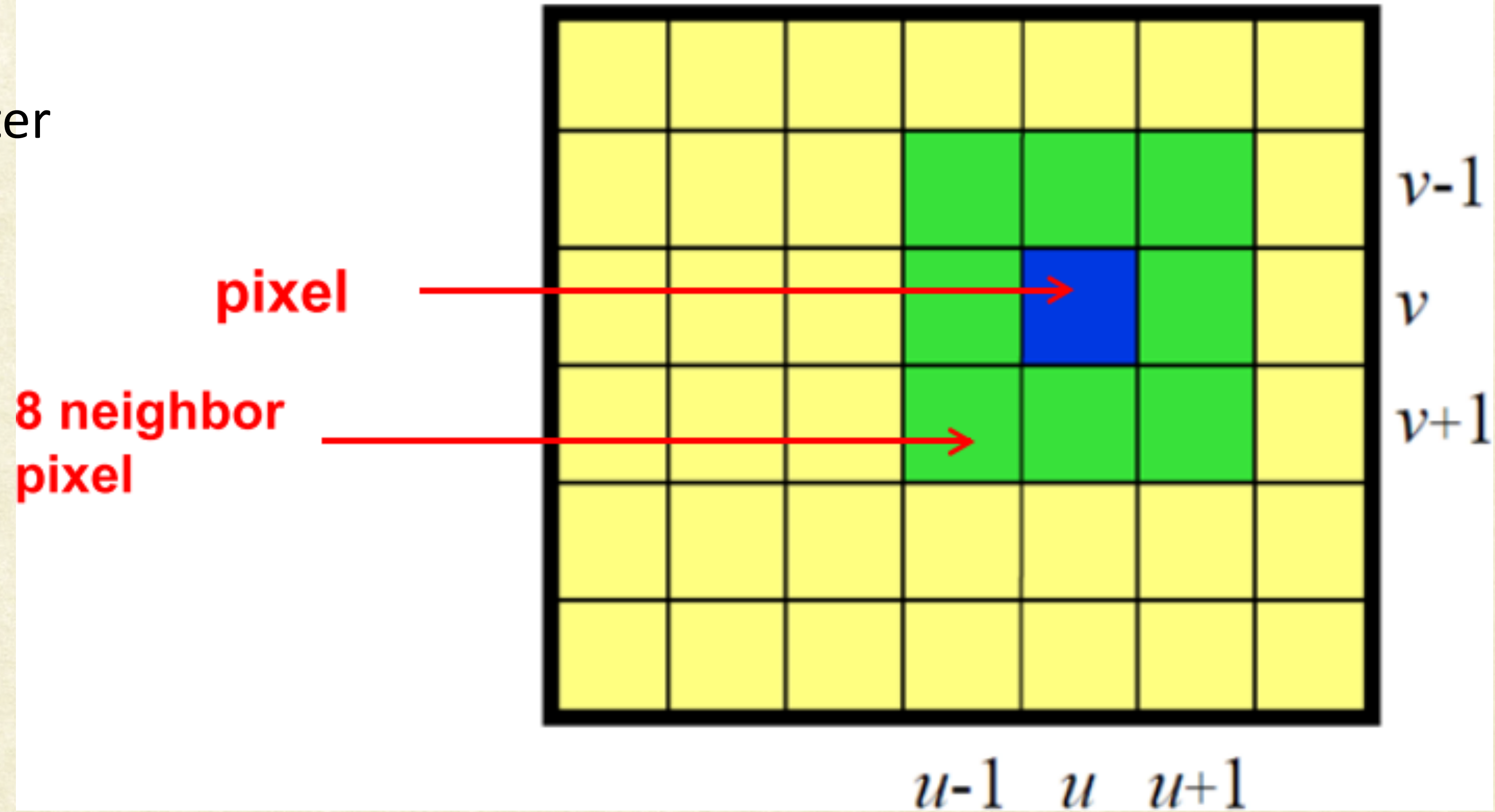
3 approaches

1. Point to Point
2. Neighbourhood to Point



Spatial Domain Filtering

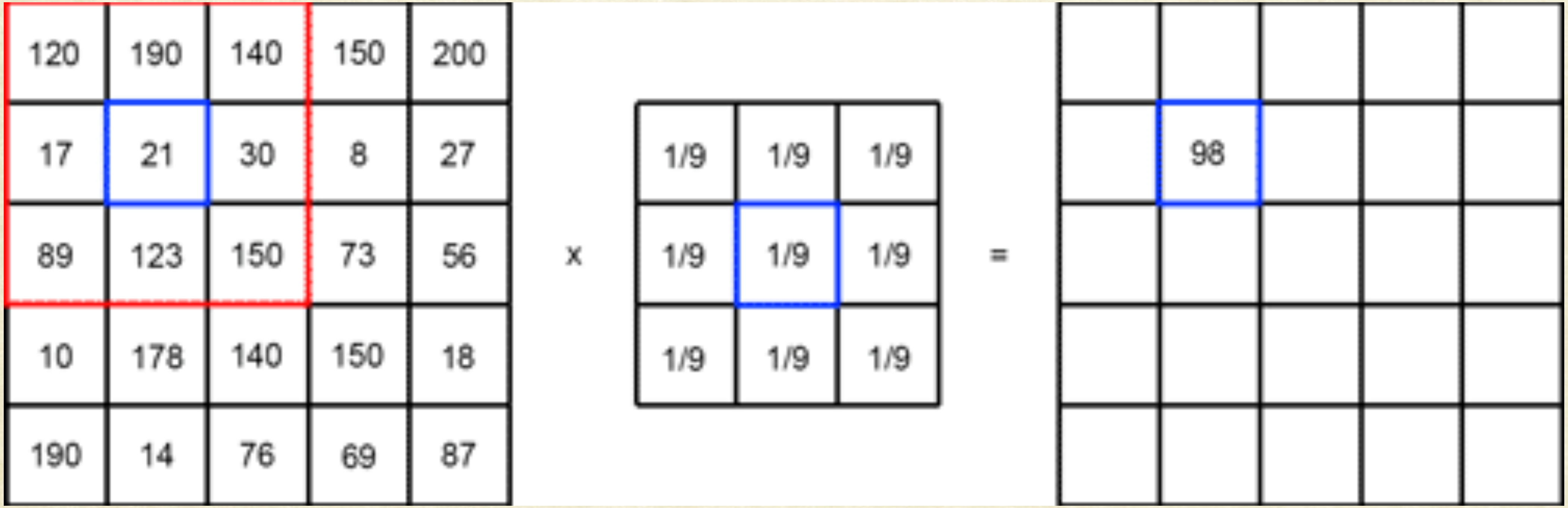
2 components of a spatial filter
- neighborhood
- a predefined operation



Filter aka
- Kernel
- Spatial Mask
- Template
- Window

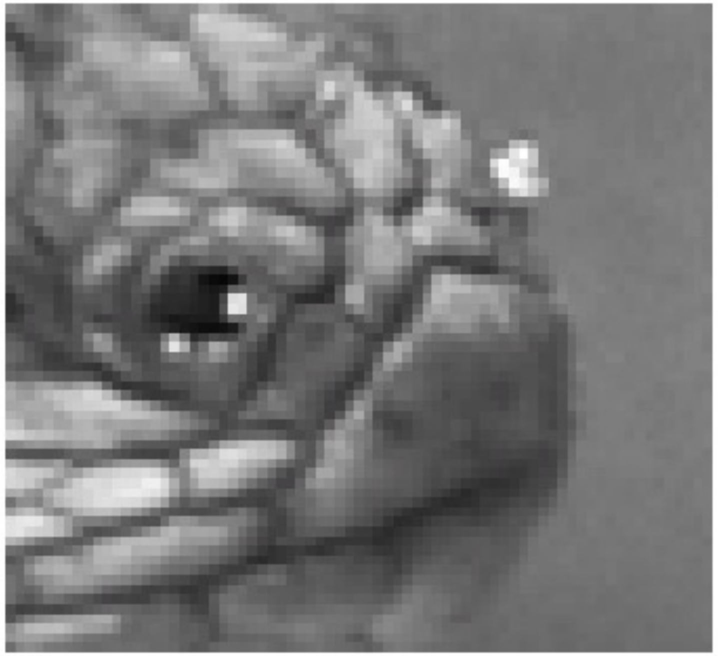
Mean/Average Filter (Smoothing)

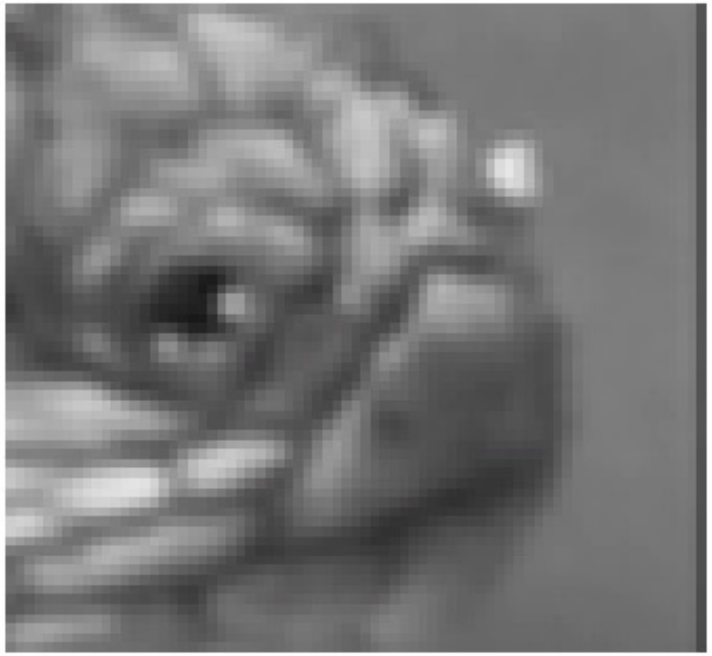
```
M = 3
For each valid location [x,y] in S
        a ← Average of intensities in a M x M neighborhood centered on [x,y]
        D[x,y] = round(a)
```



http://199.116.233.101/index.php?title=File:Animated-kernels.gif

$$\frac{1}{9} * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

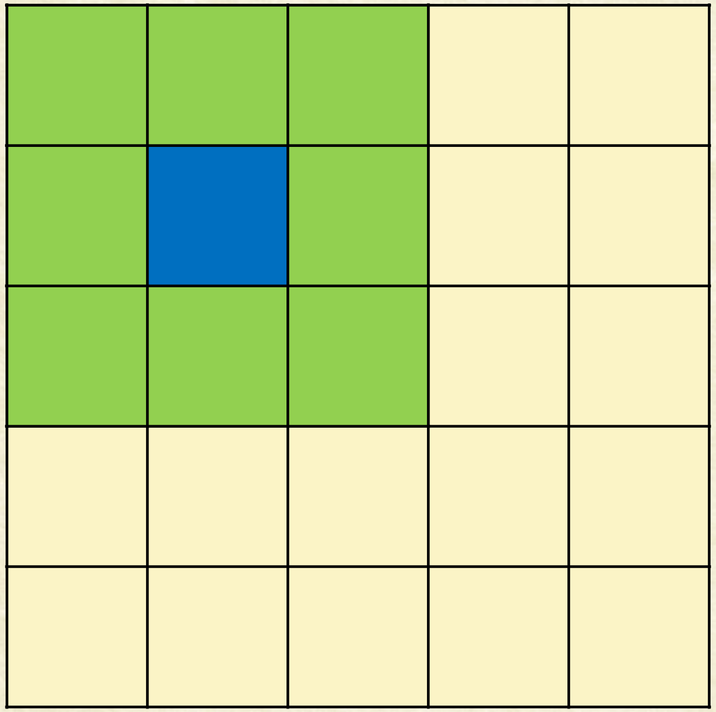




Mean/Average Filter

Note: Coefficients sum to 1

I



H → 3 x 3

Weight Mask / Kernel / Filter

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

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

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

Effect of Mask Size

Original Image

[3x3]

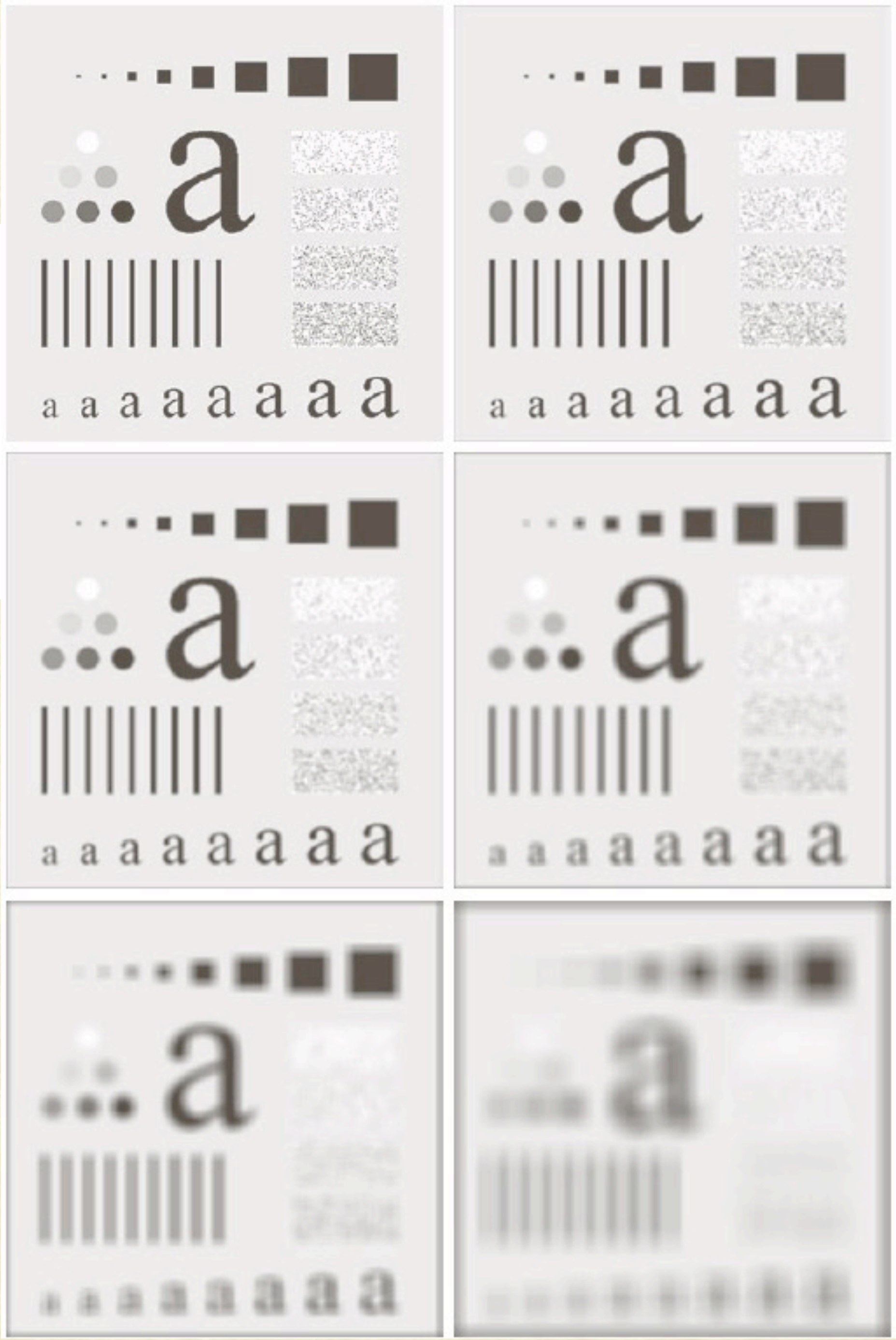
[5x5]

[7x7]

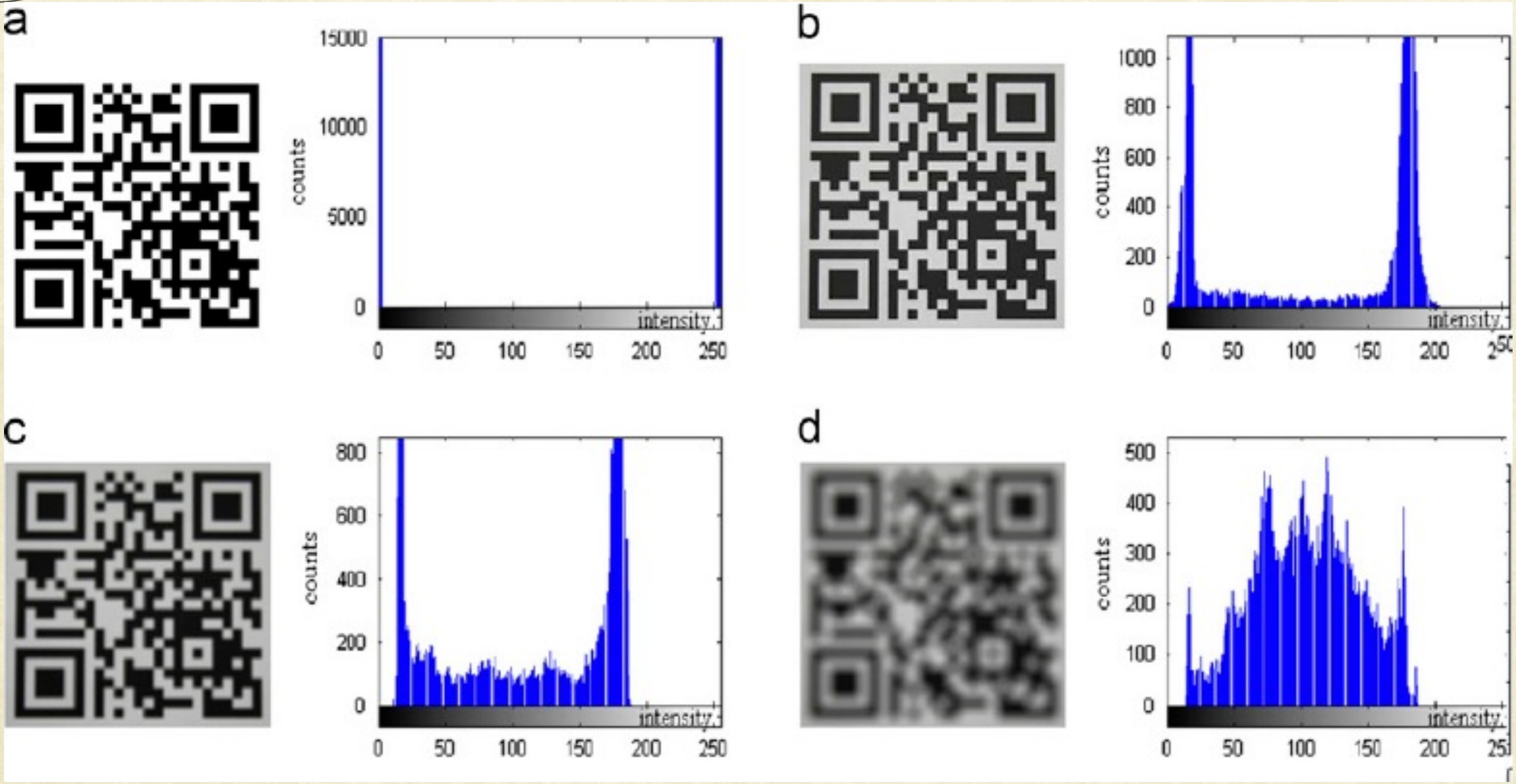
Square averaging filter

FIGURE 3.33 (a) Original image, of size 500 × 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, respectively; 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 × 120 pixels.

a b
c d
e f



Averaging – a histogram perspective



https://www.researchgate.net/figure/Histograms-of-barcode-images-with-different-levels-of-blur-a-a-synthetic-QR-code-image_fig3_224201499

Repeated Averaging Using Same Filter

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

Weighted Averaging

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Standard average

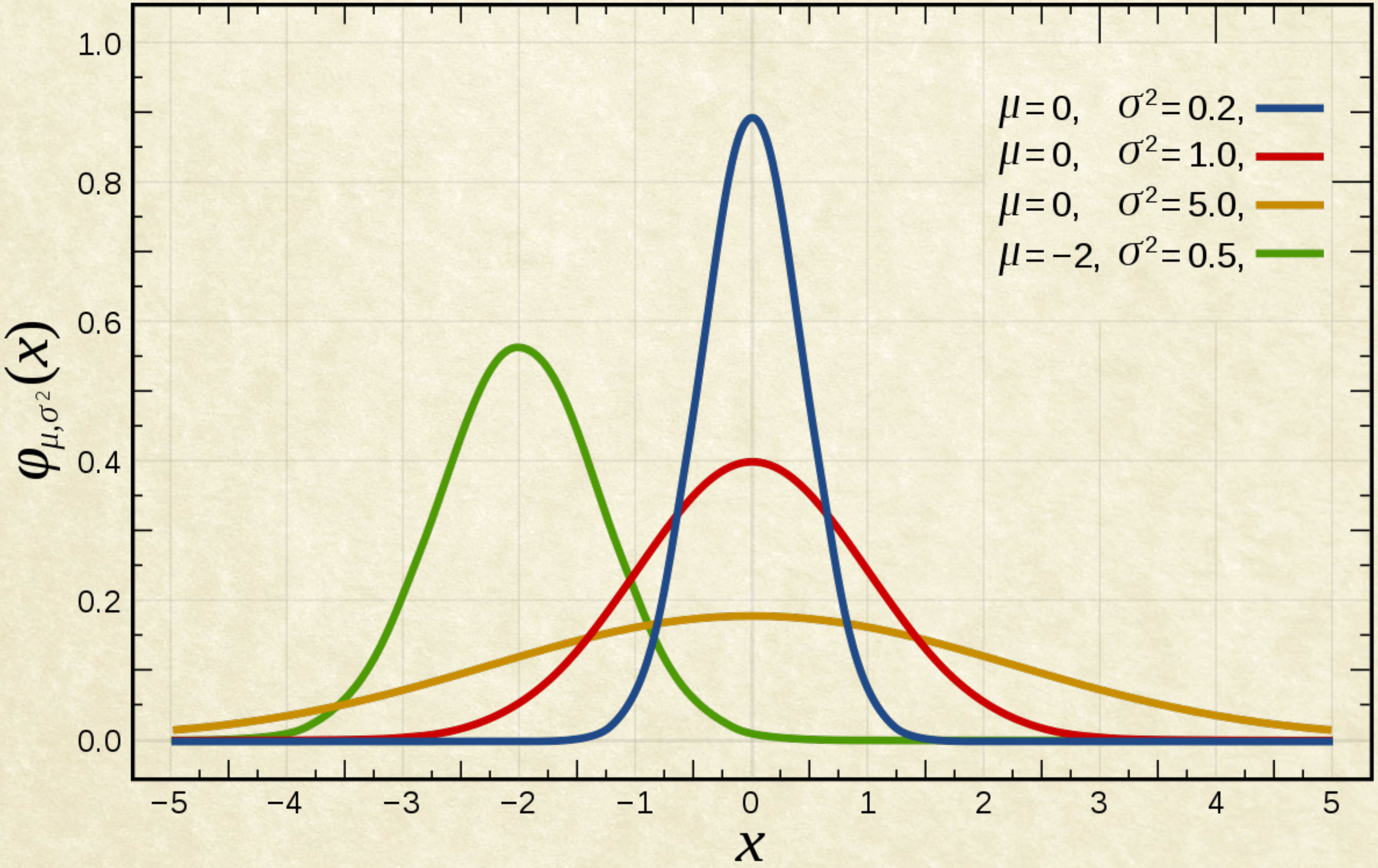
Weighted Average

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

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

Gaussian Function (1-D)

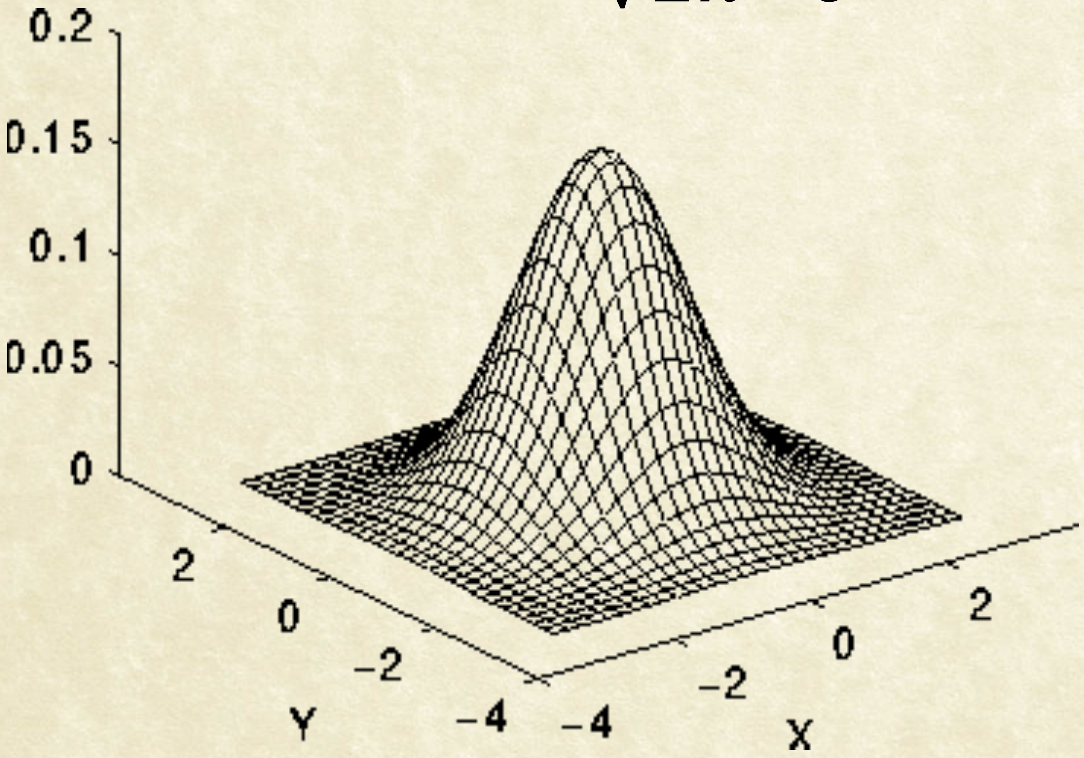
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Gaussian Smoothing

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

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



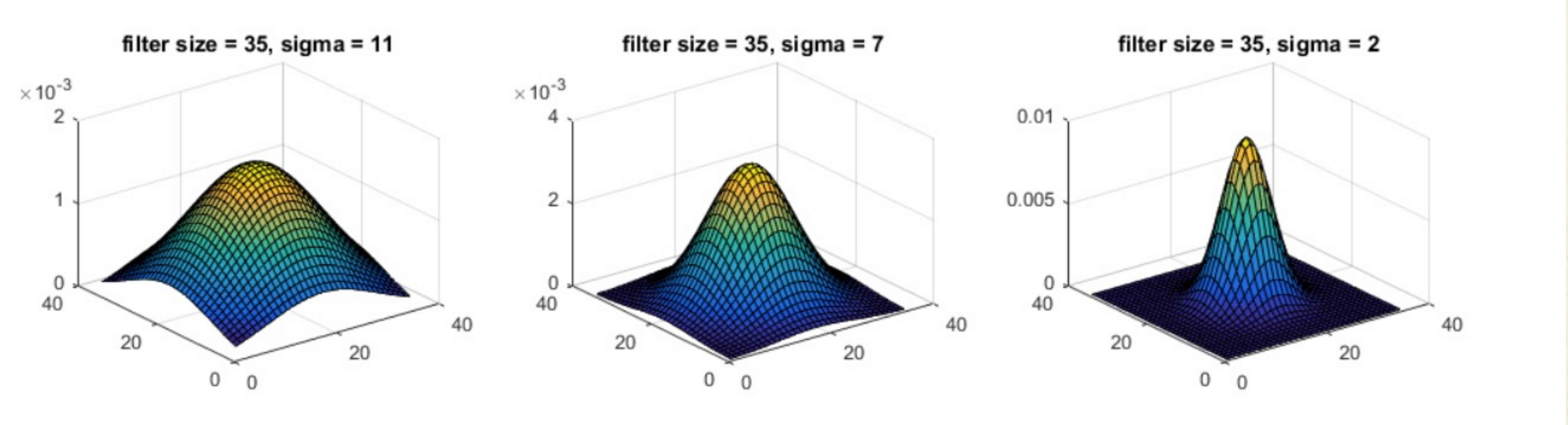
$\frac{1}{265}$

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

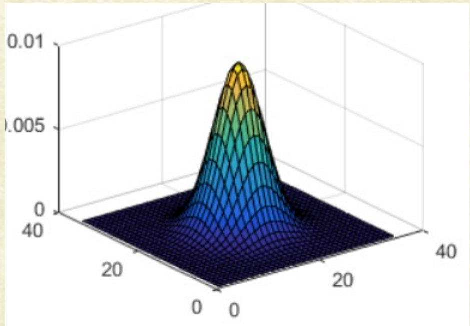
$$G(x,y) = \frac{1}{\sqrt{2\pi}\cdot\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

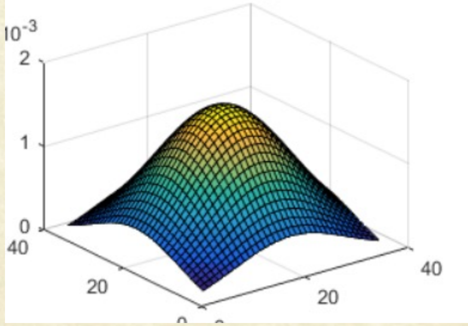


Gaussian Smoothing – Effect of sigma

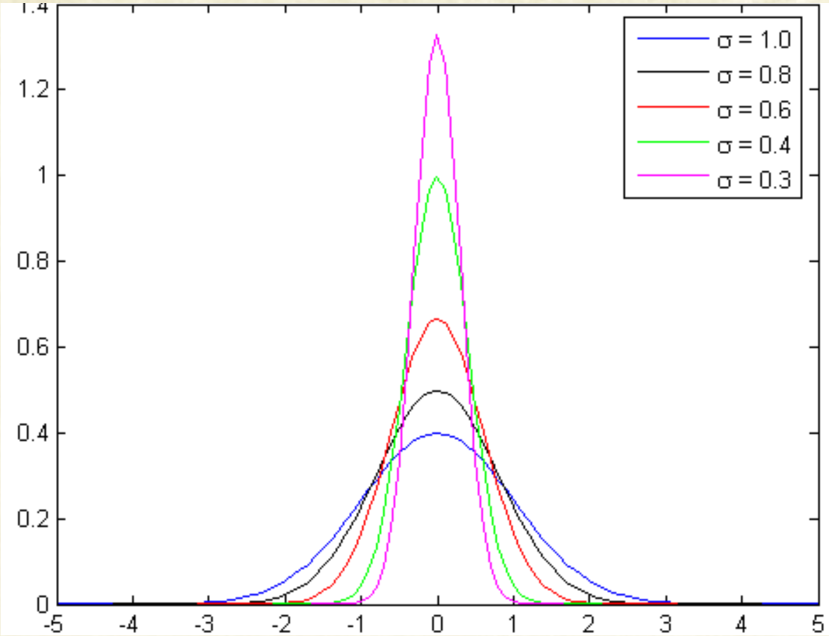
Original Image (Sigma 0) | Gaussian Blur (Sigma 0.7) | Gaussian Blur (Sigma 2.8)

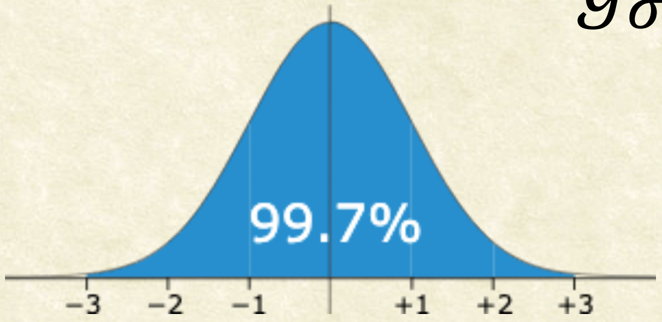
$$G(x,y) = \frac{1}{\sqrt{2\pi}\cdot\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



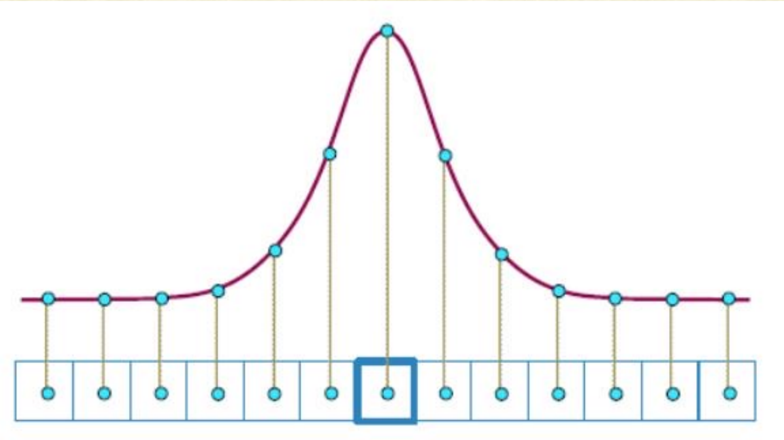


Computing Gaussian filter coefficients





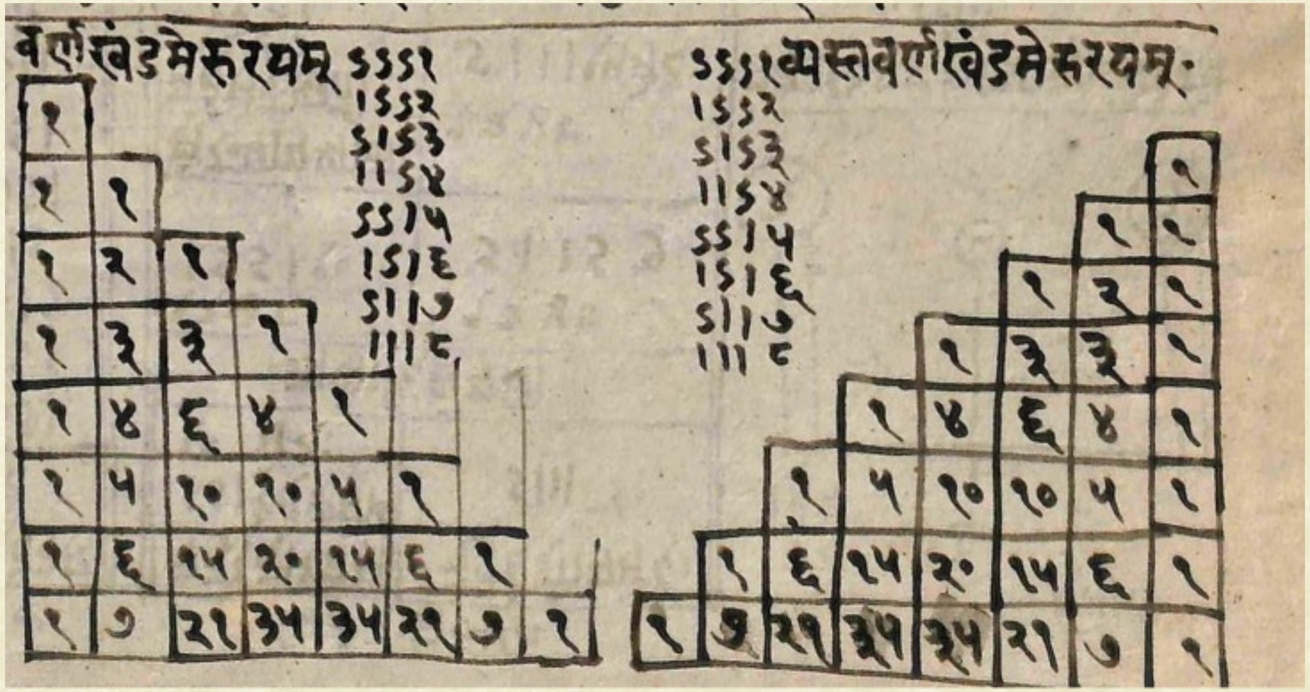
$$g_\sigma(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\left(\frac{x^2}{2\sigma^2}\right)}$$



Computing Gaussian filter coefficients

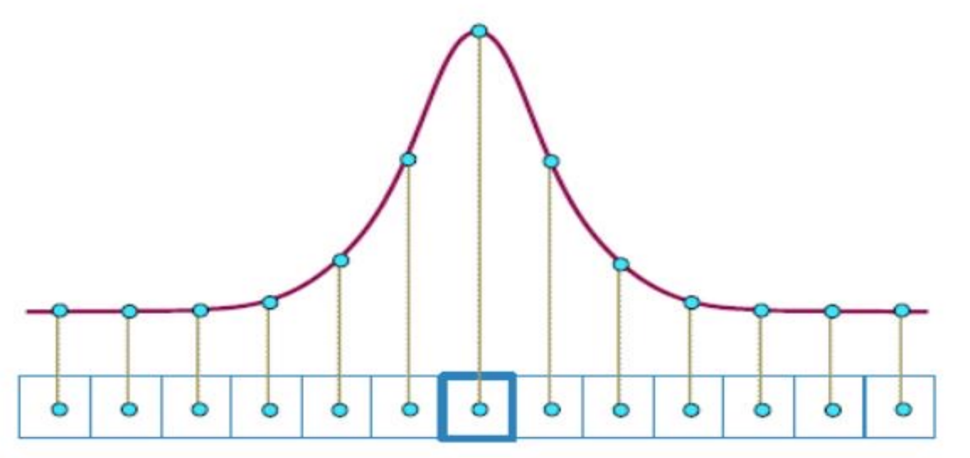
Index N	Coefficients	Sum = 2^N
0	1	1
1	1 1	2
2	1 2 1	4
3	1 3 3 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024
11	1 11 55 165 330 462 462 330 165 55 11 1	2048
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^N
0	1	1
1	1 1	2
2	1 2 1	4
3	1 3 3 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024



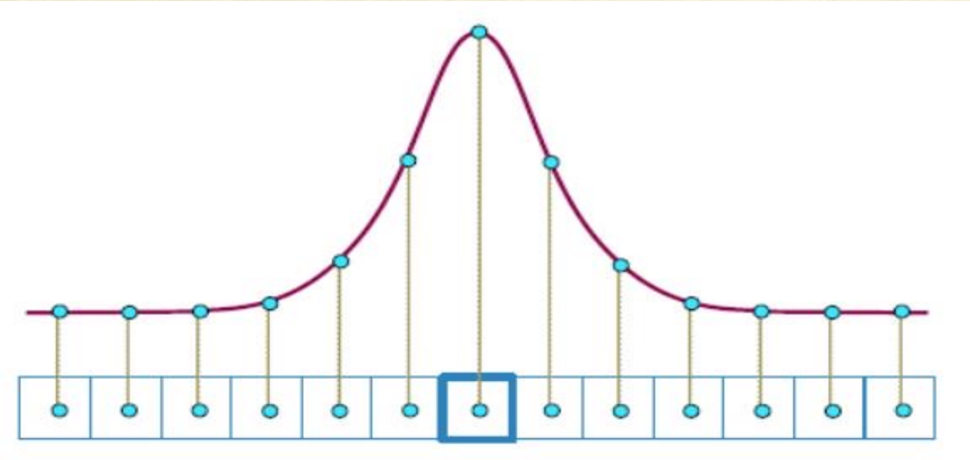
e.g., $s = 7\times7$

$$\frac{1}{64}\times \quad 1 \quad 6 \quad 15 \quad 20 \quad 15 \quad 6 \quad 1$$

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

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 1	8
4	1 4 6 4 1	16
5	1 5 10 10 5 1	32
6	1 6 15 20 15 6 1	64
7	1 7 21 35 35 21 7 1	128
8	1 8 28 56 70 56 28 8 1	256
9	1 9 36 84 126 126 84 36 9 1	512
10	1 10 45 120 210 252 210 120 45 10 1	1024



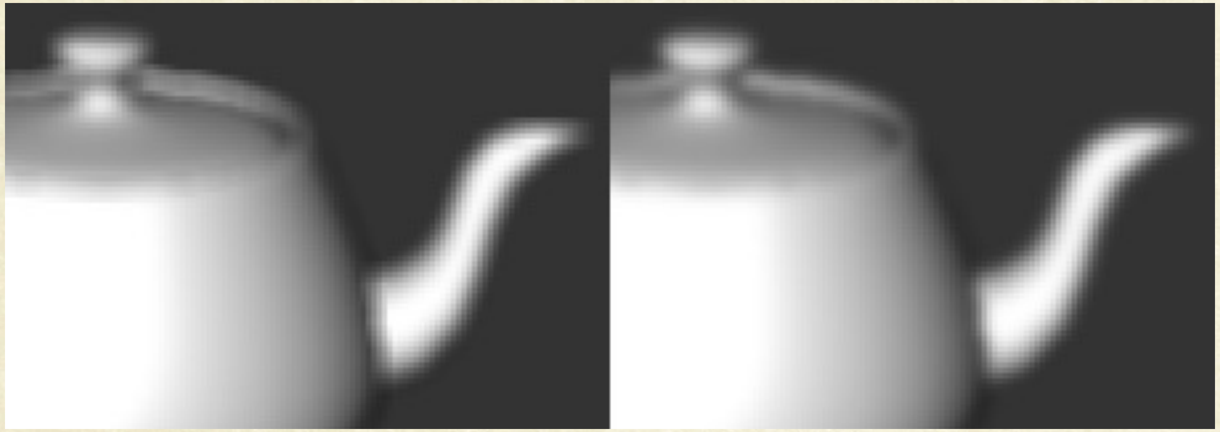
e.g., $s = 7\times 7$

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

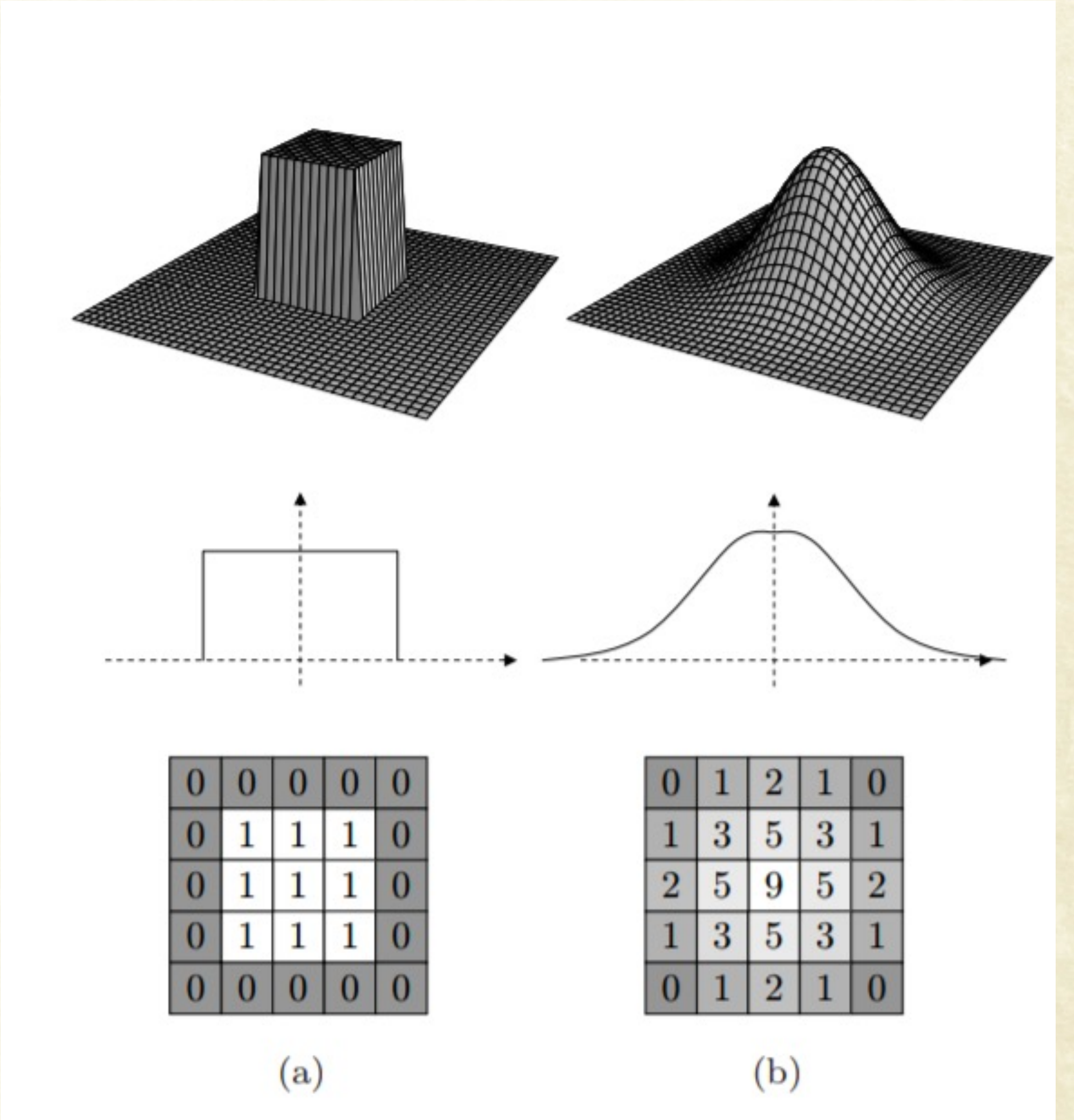
1	6	15	20	15	6	1
6	36	90	120	90	36	6
15	90	225	300	225	90	15
20	120	300	400	300	120	20
15	90	225	300	225	90	15
6	36	90	120	90	36	6
1	6	15	20	15	6	1

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

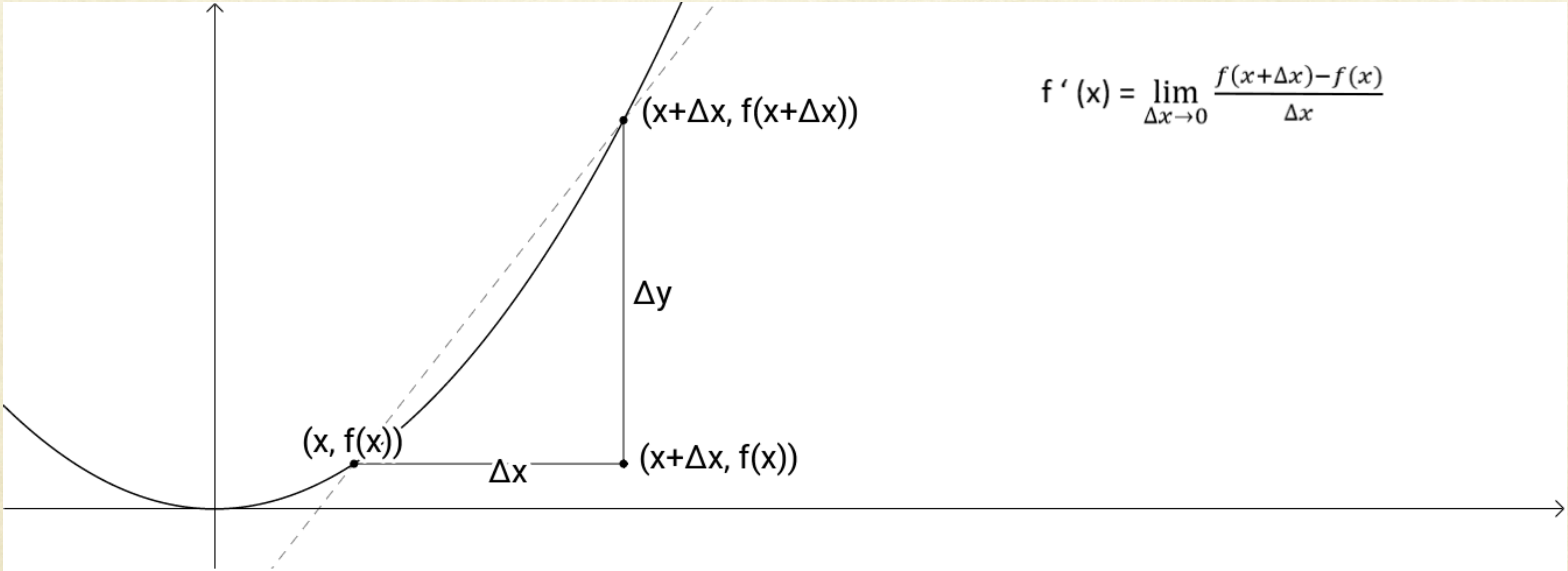
Averaging vs. Gaussian Filter



Smoother intensity transitions

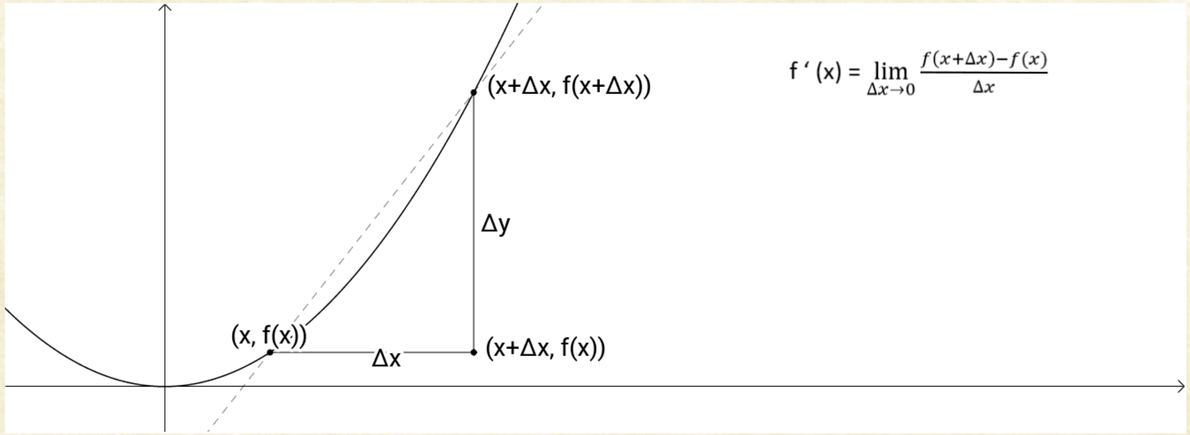


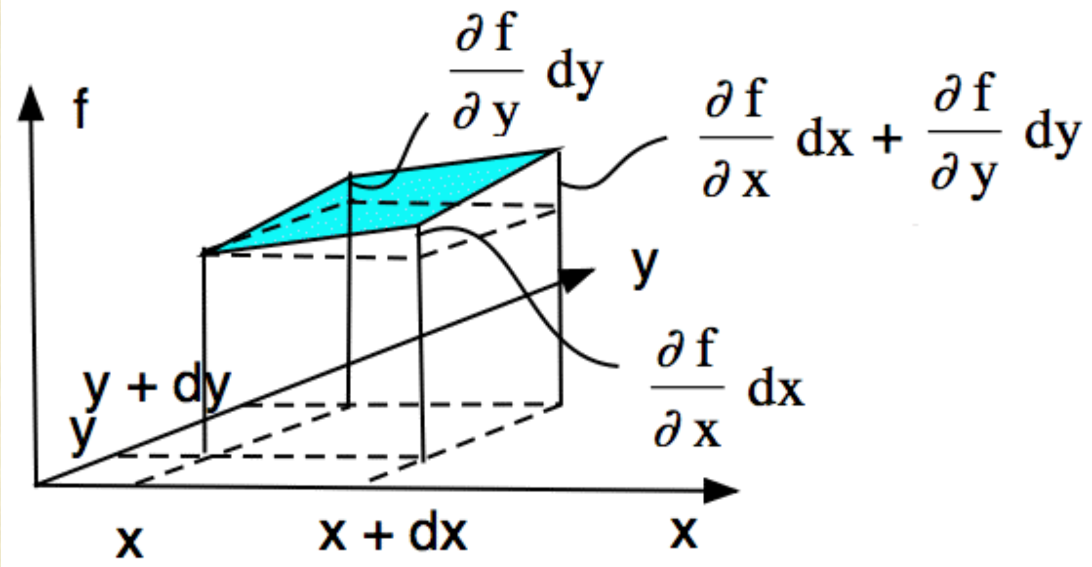
Recap: Derivatives



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

Recap: Derivatives

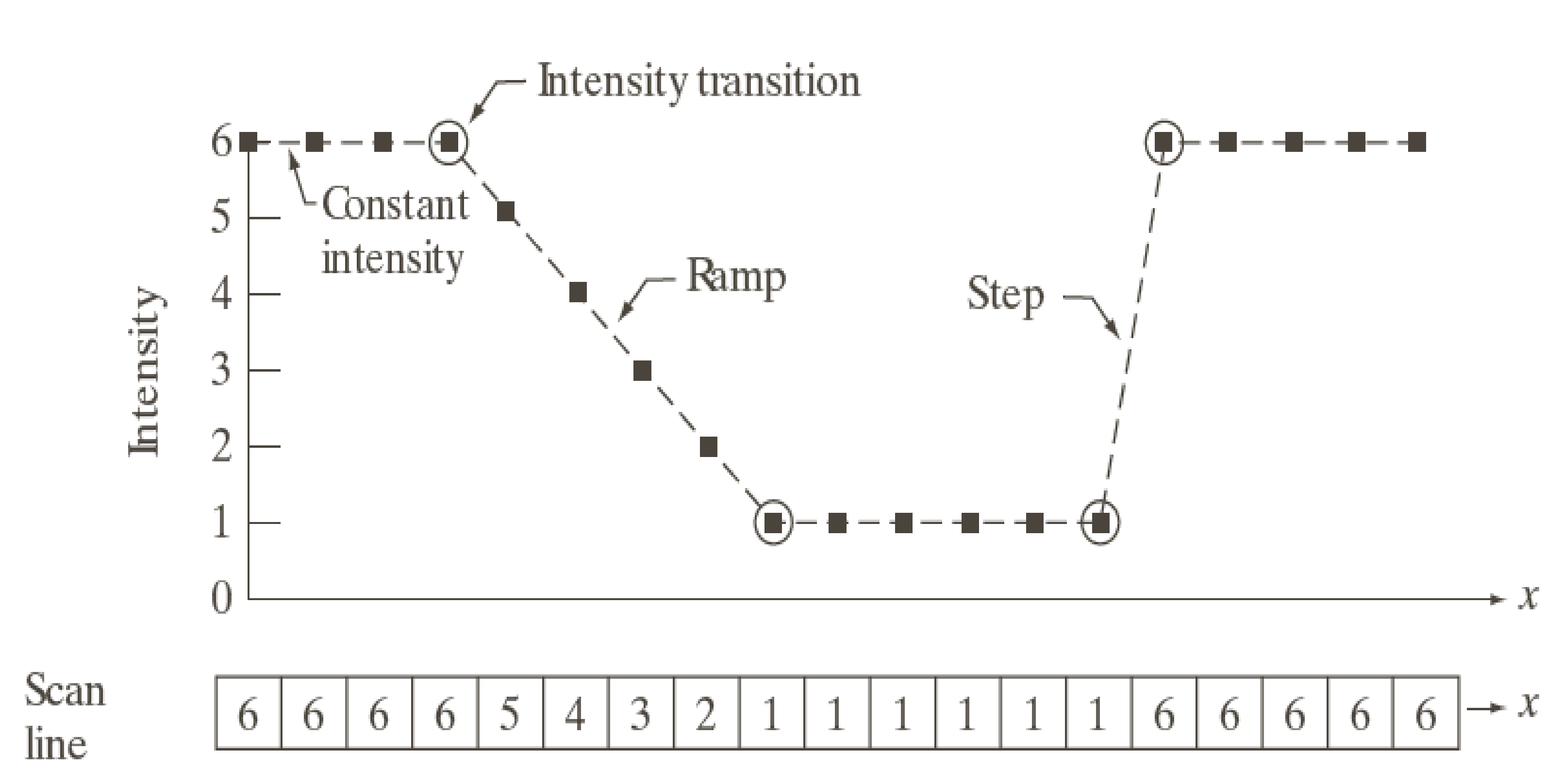




$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x\to 0} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}$$

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

First Derivative (Digital approximation)



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

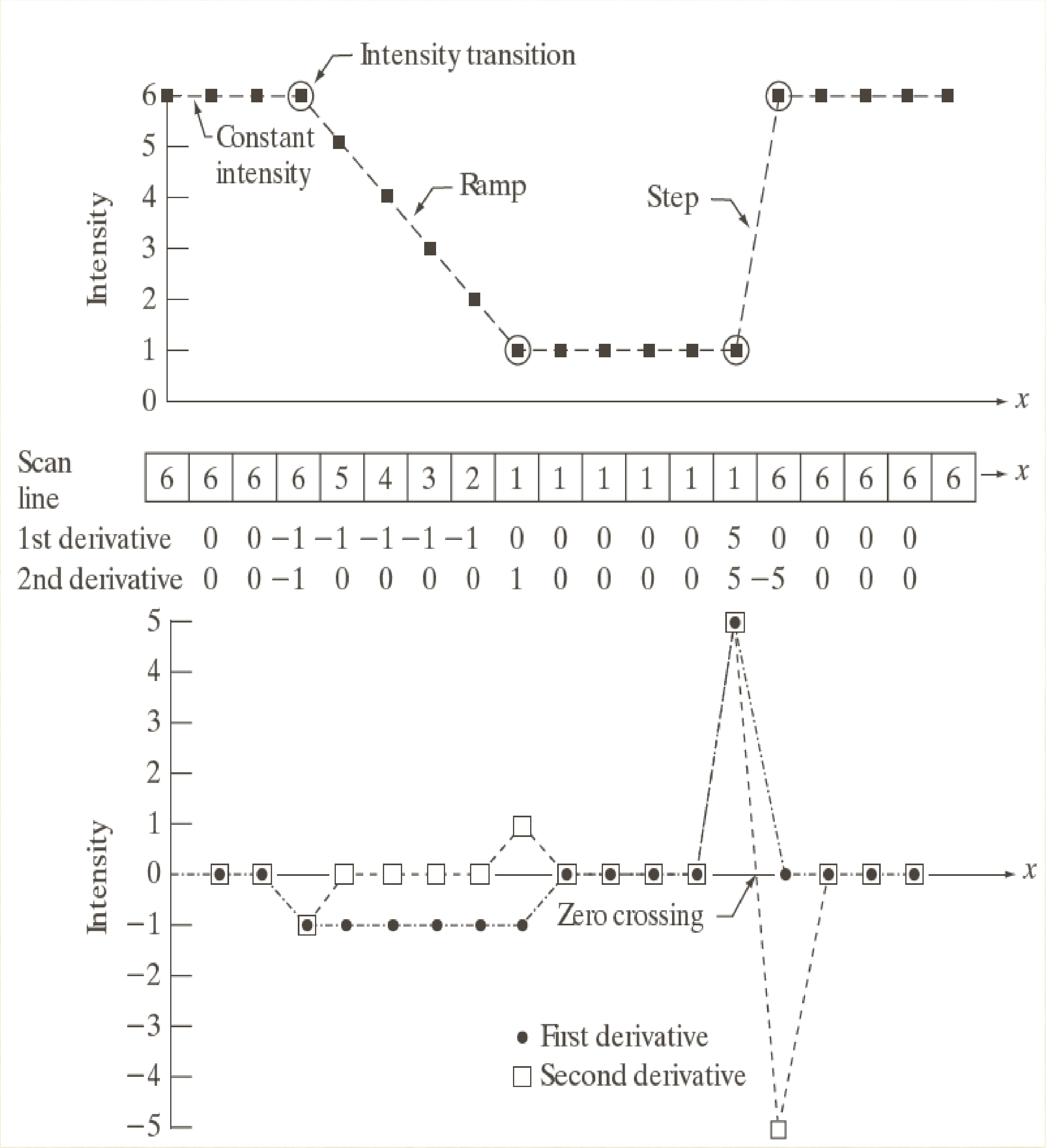
$$\frac{\partial f(x,y)}{\partial x} \sim f[x+1,y] - f[x,y]$$

Second Derivative (Digital approximation)

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

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

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



Alt: Derivative as Symmetric Difference

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

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

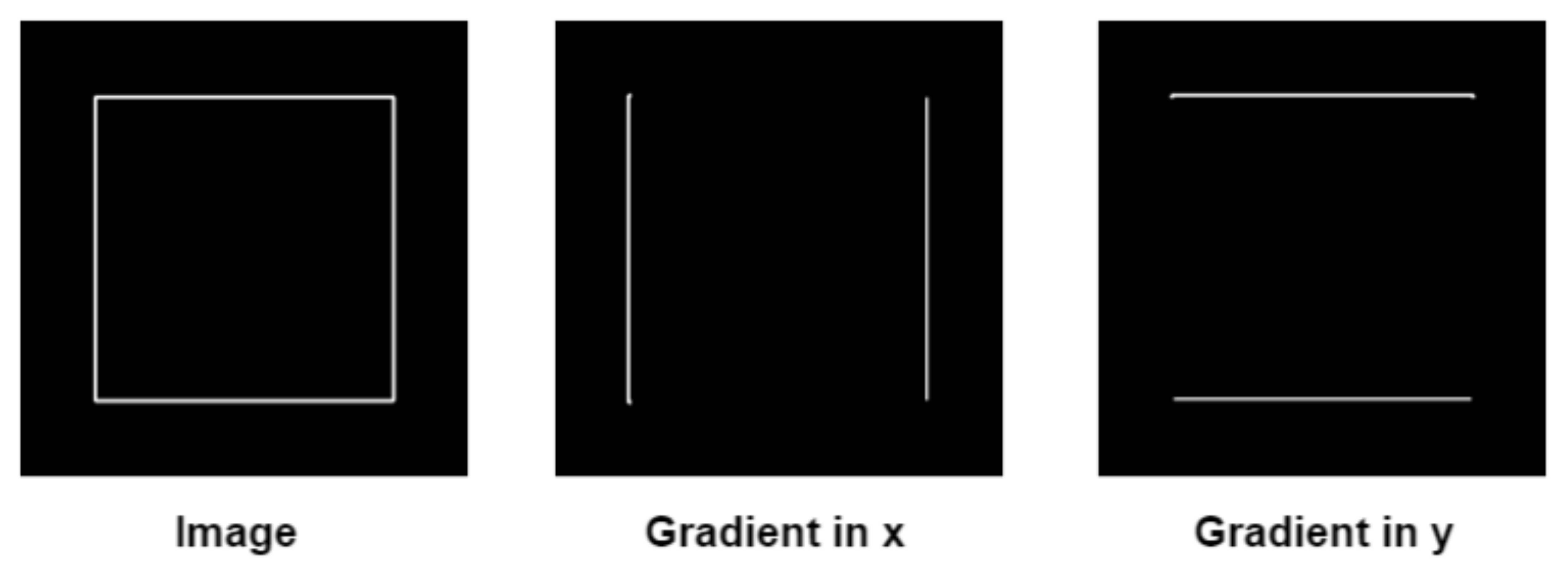
Image Gradient and Edges

$$\frac{f(x+h,y)-f(x-h,y)}{2h} \Rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

x-derivative

$$\frac{f(x,y+h)-f(x,y-h)}{2h} \Rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

y-derivative



https://i1.wp.com/theailearner.com/wp-content/uploads/2019/05/gradxy.png?resize=768%2C280&ssl=1

Edge ‘Image’

https://i0.wp.com/theailearner.com/wp-content/uploads/2019/05/Sobel64.png?w=731&ssl=1

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

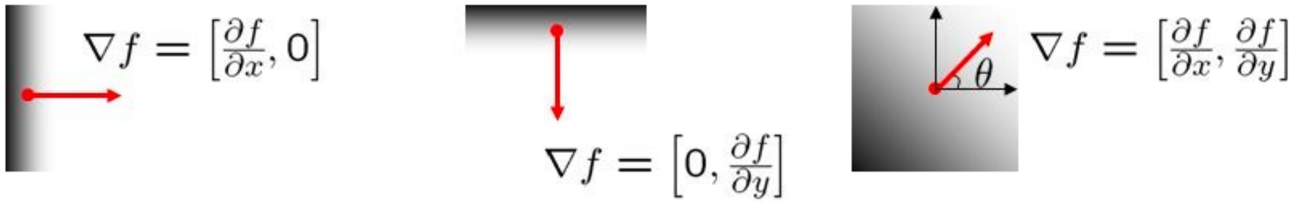
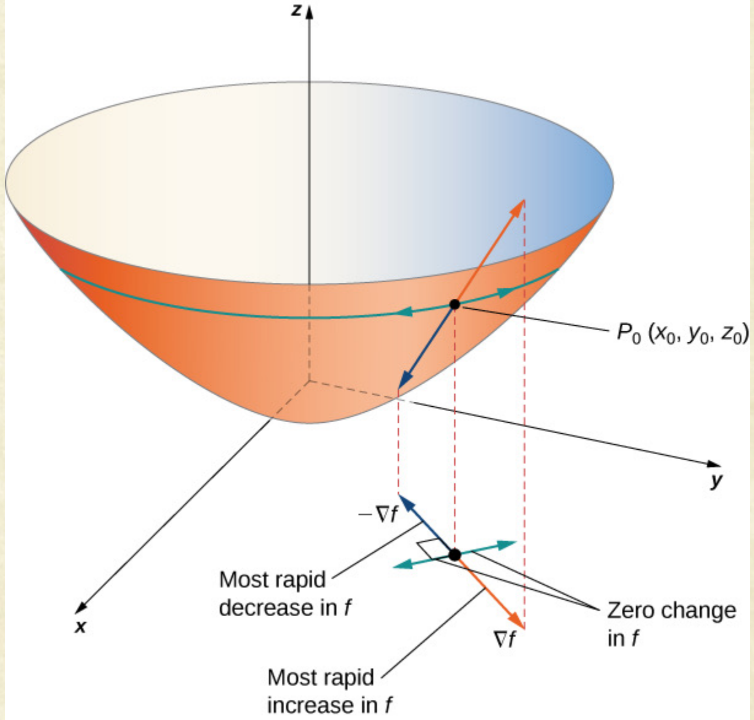


Image **Gradient in x** **Gradient in y**



Prewitt Edge Filter

Dr. Prewitt

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

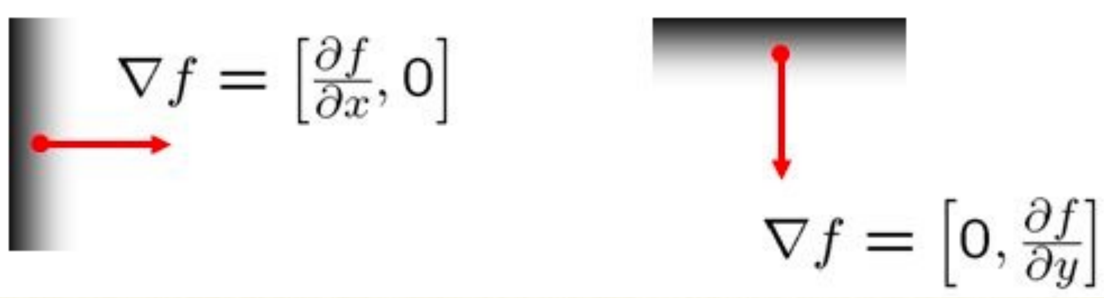
-1	0	+1
-1	0	+1
-1	0	+1

G_x

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

G_y

Edge is perpendicular to gradient

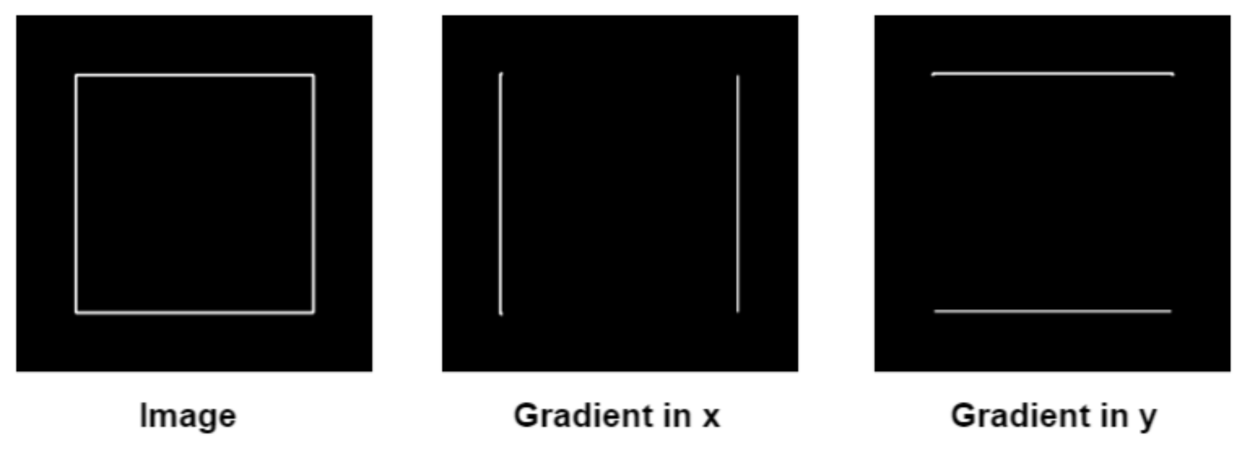


-1	0	+1
-1	0	+1
-1	0	+1

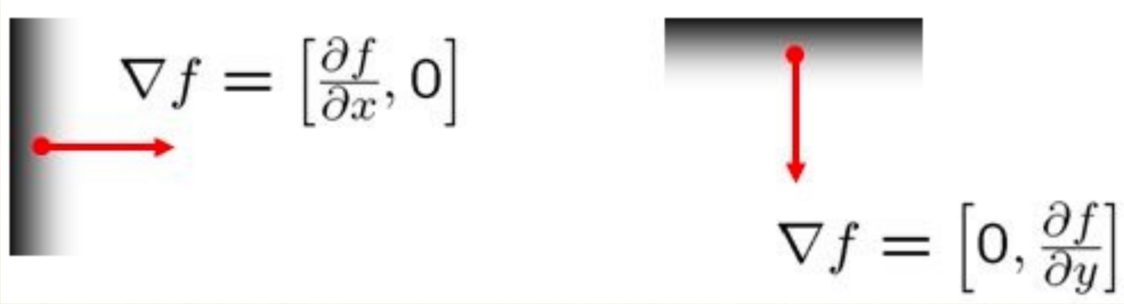
G_x

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

G_y



Edge is perpendicular to gradient

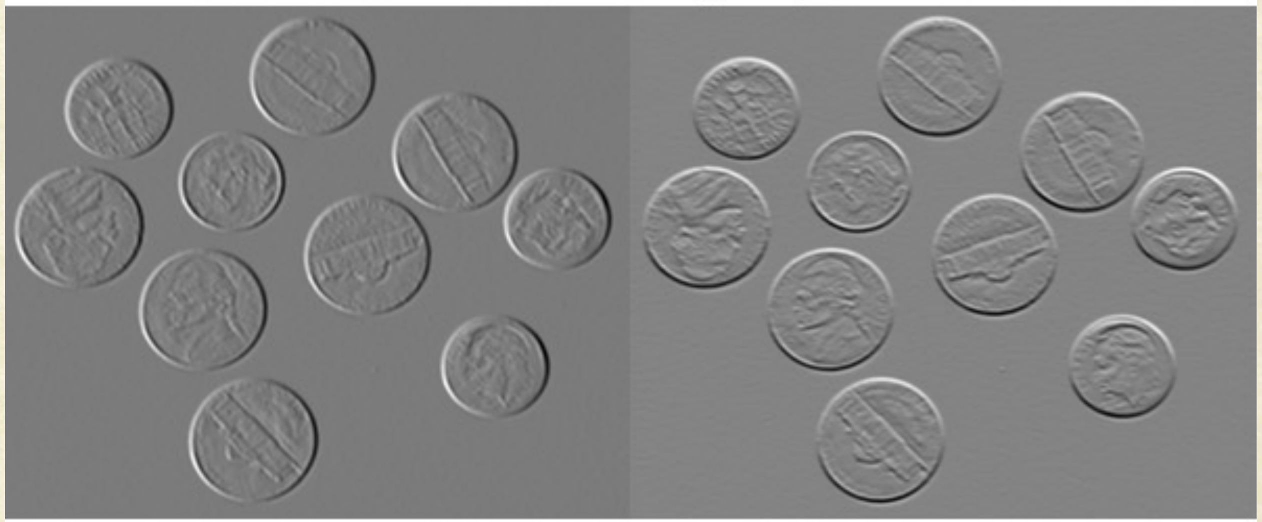


-1	0	+1
-1	0	+1
-1	0	+1

G_x

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

G_y



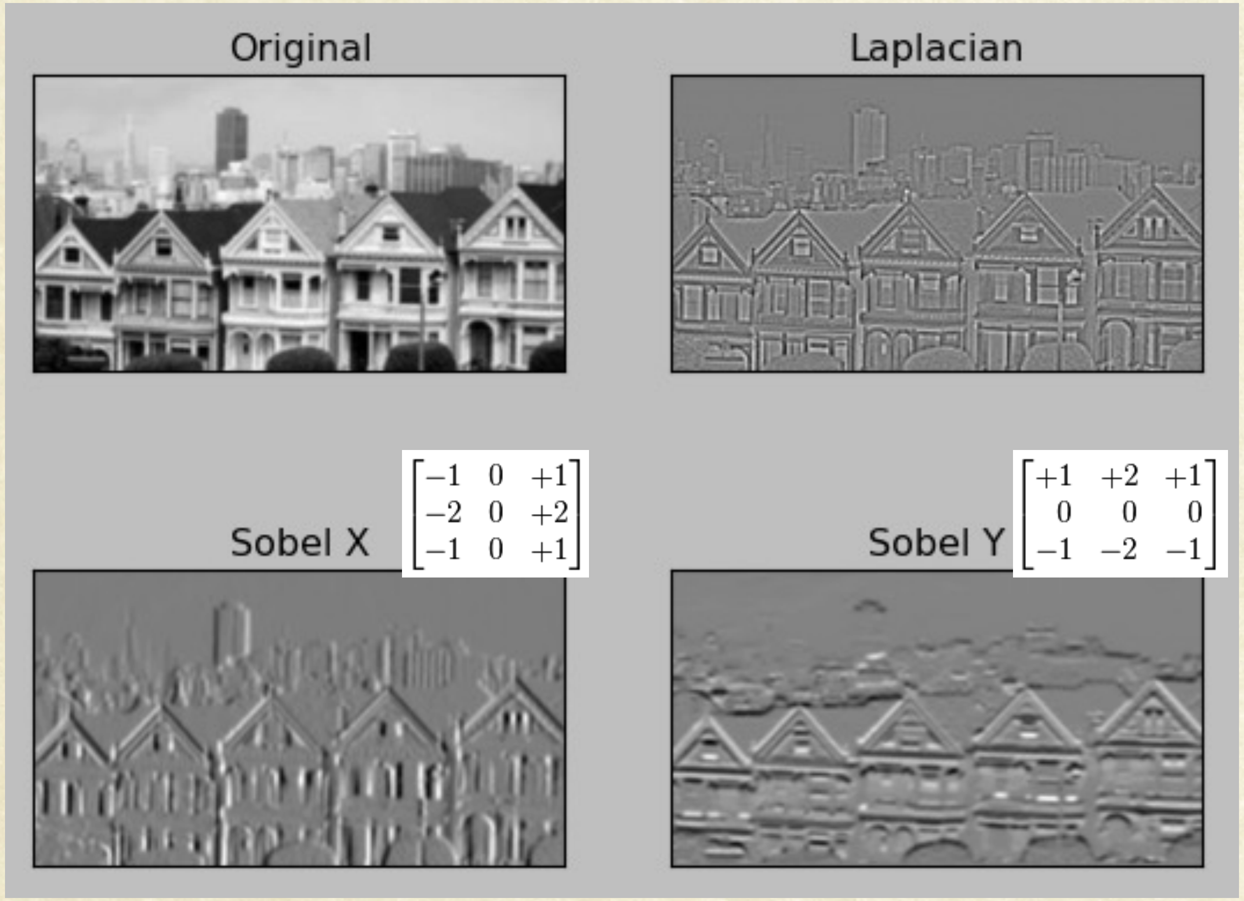
2-D Laplacian Filter

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

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

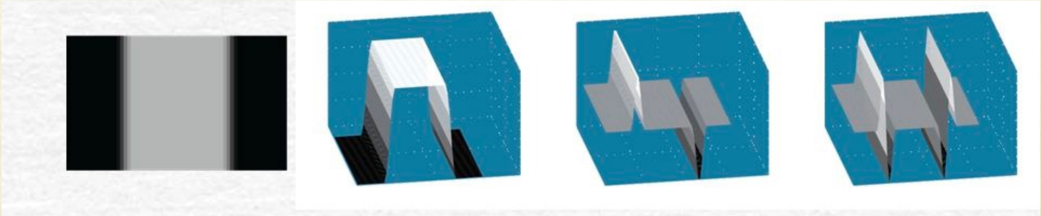
0	1	0
1	-4	1
0	1	0

Edge Masks: Sobel, Laplacian

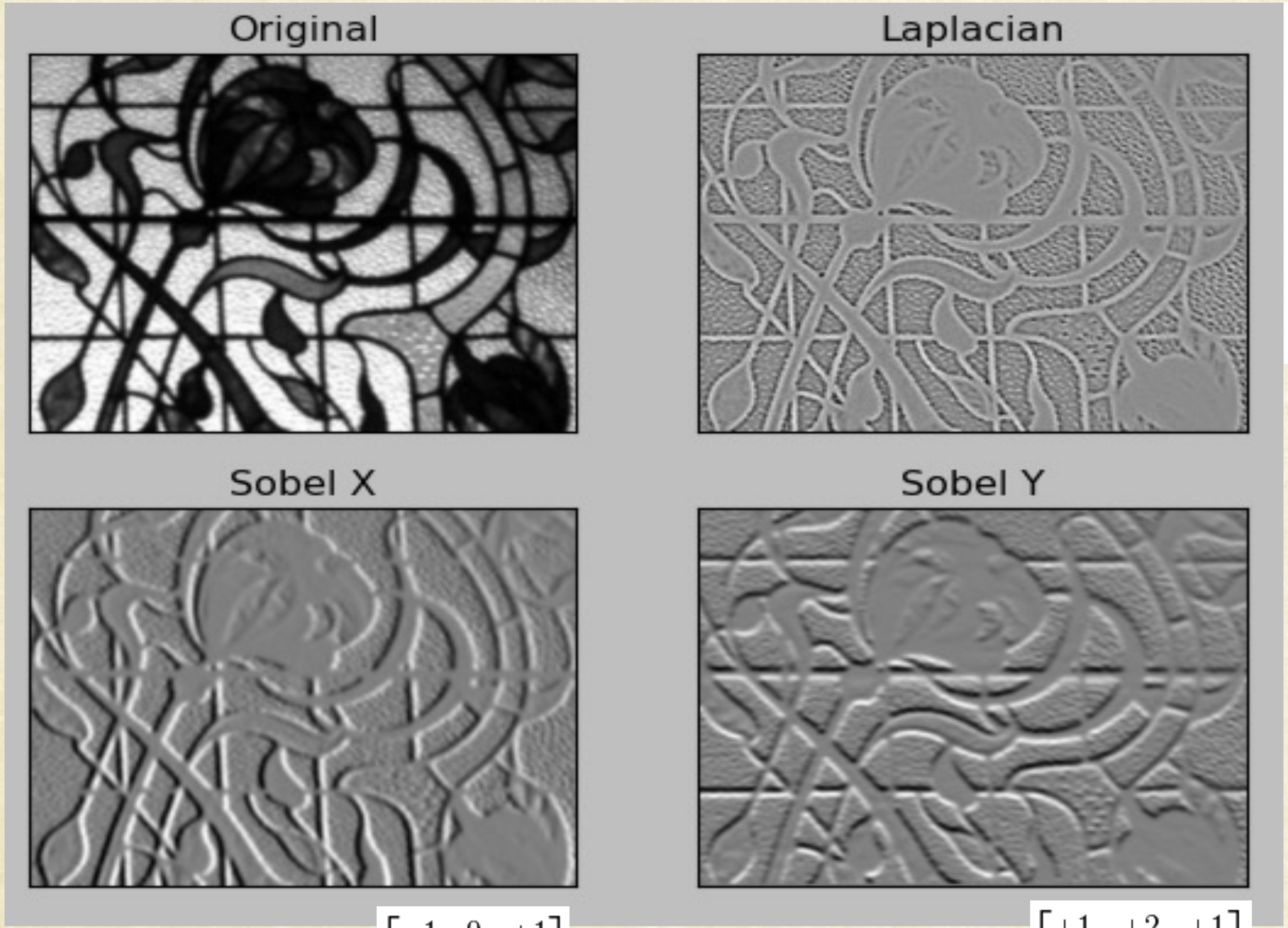


0	−1	0
−1	4	−1
0	−1	0

Note: Coefficients sum to 0



Edge Masks: Sobel, Laplacian



0	−1	0
−1	4	−1
0	−1	0

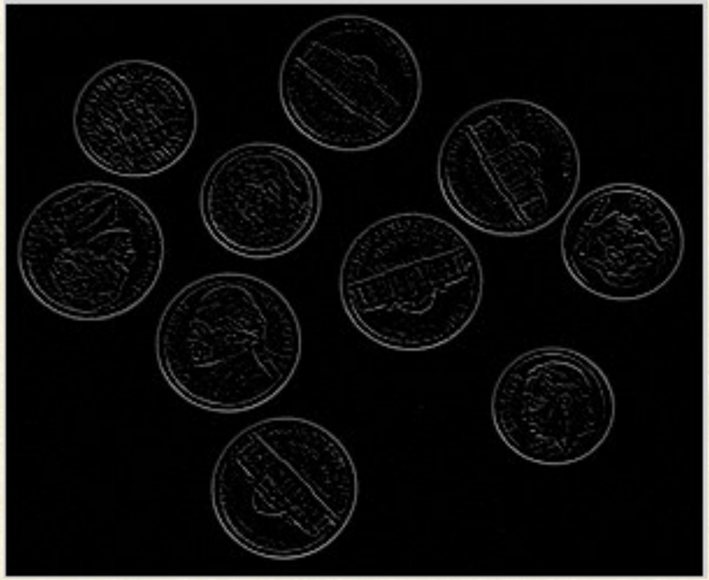
$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

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

Image Sharpening

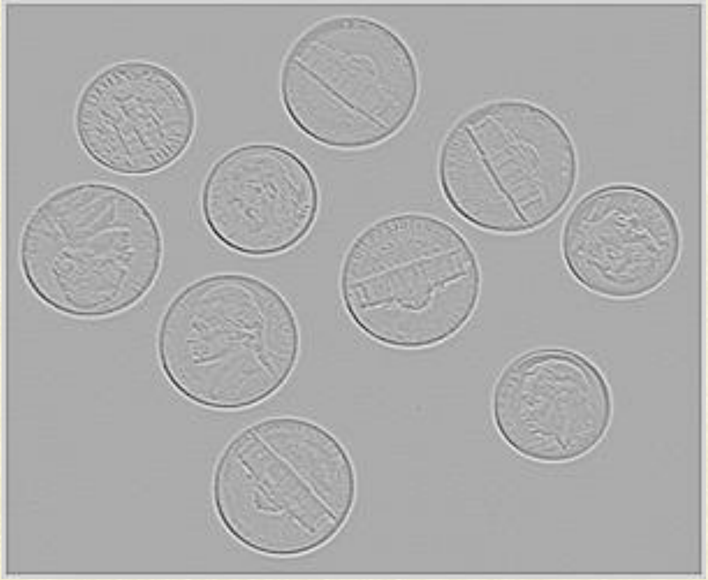
$I(u,v)$

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



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

(For visualization)



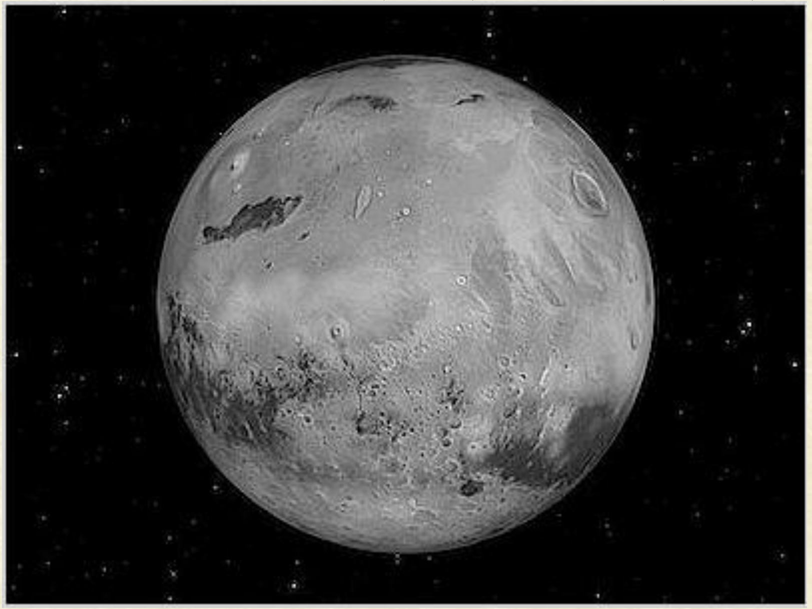
$I'(u,v) = I(u,v) + \nabla^2 I(u,v)$

Sharpening (Unsharp Masking)

$I(u,v)$

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

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



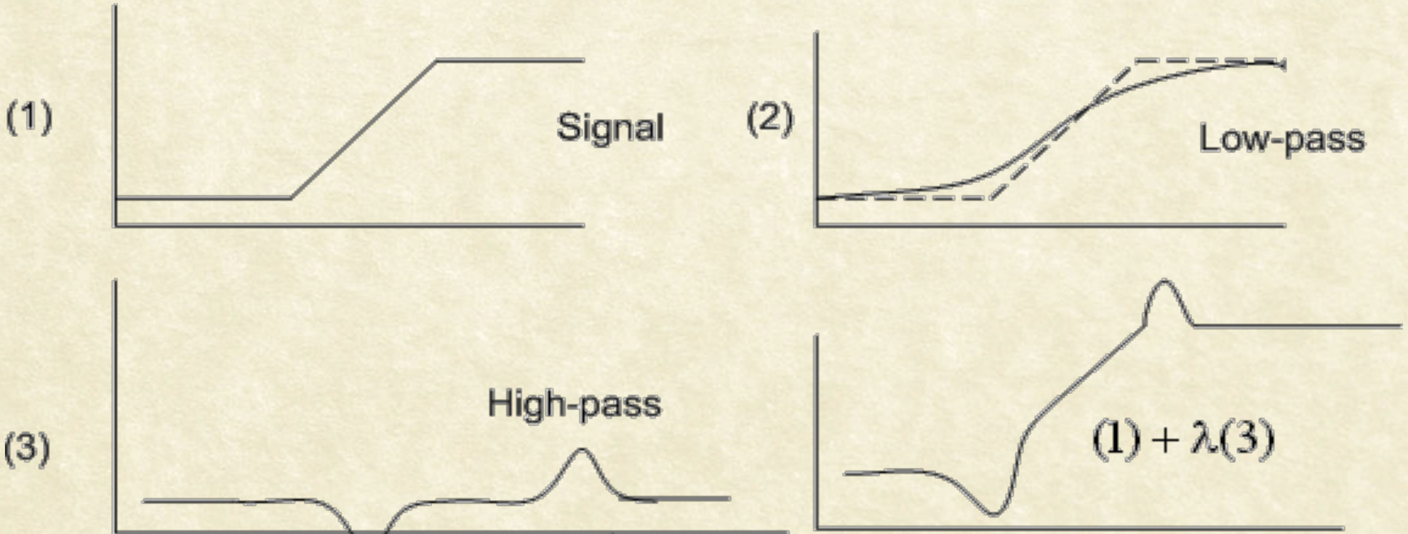
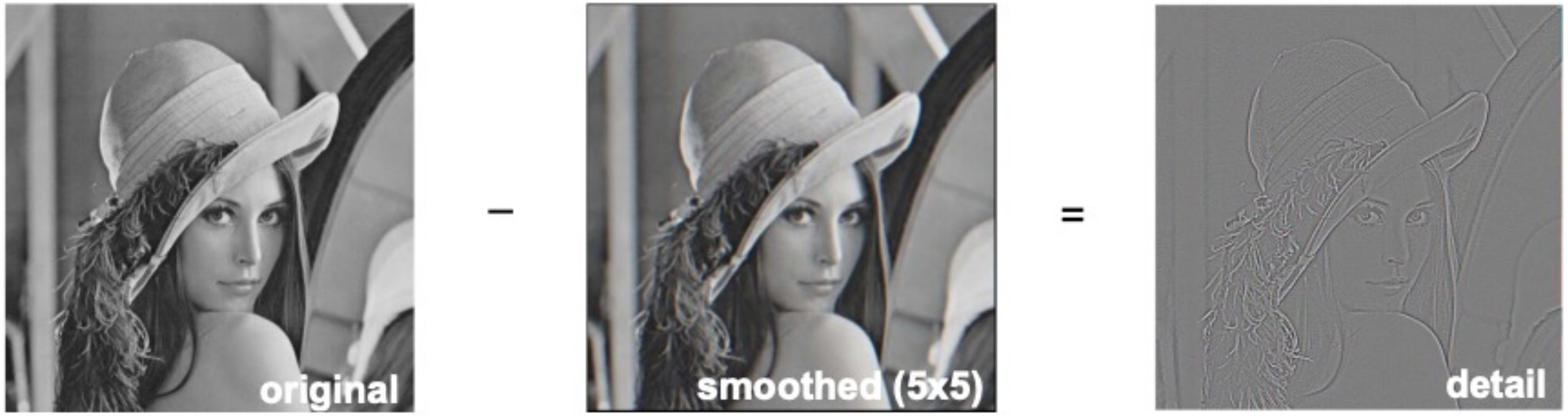


Image Courtesy:NASA

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

w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

A=2

w = 17

-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?