

CAP5415-Computer Vision

Lecture 2-Filtering

Ulas Bagci

bagci@ucf.edu

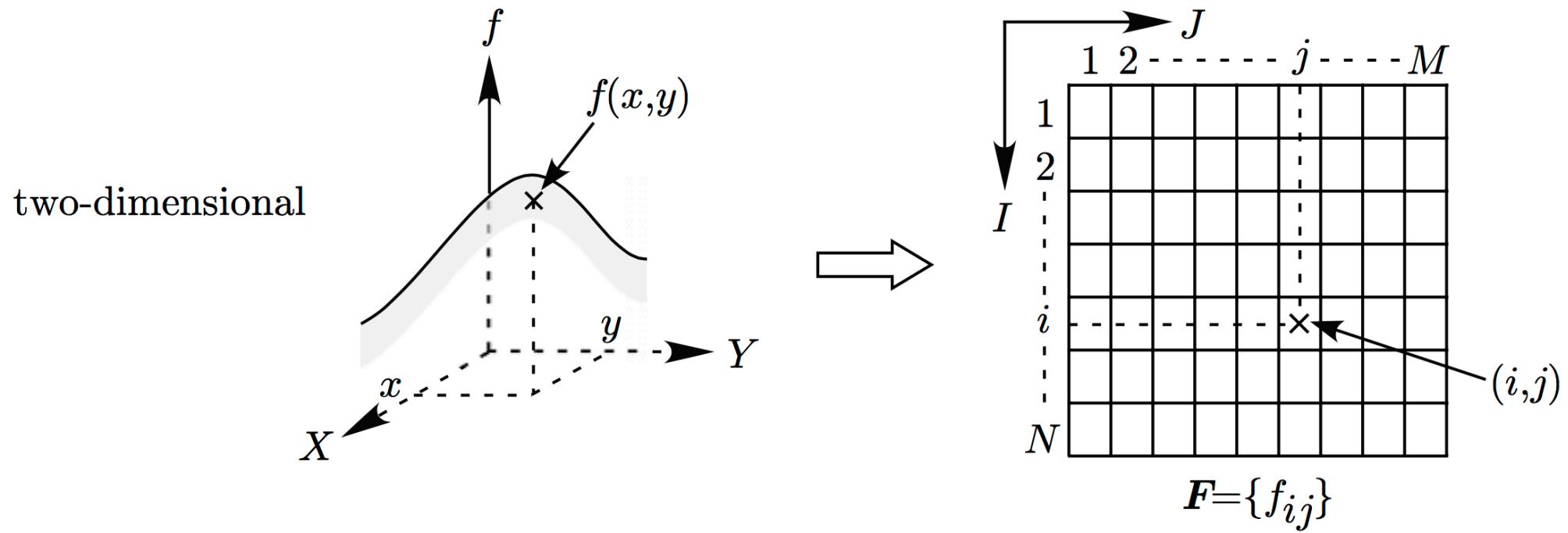
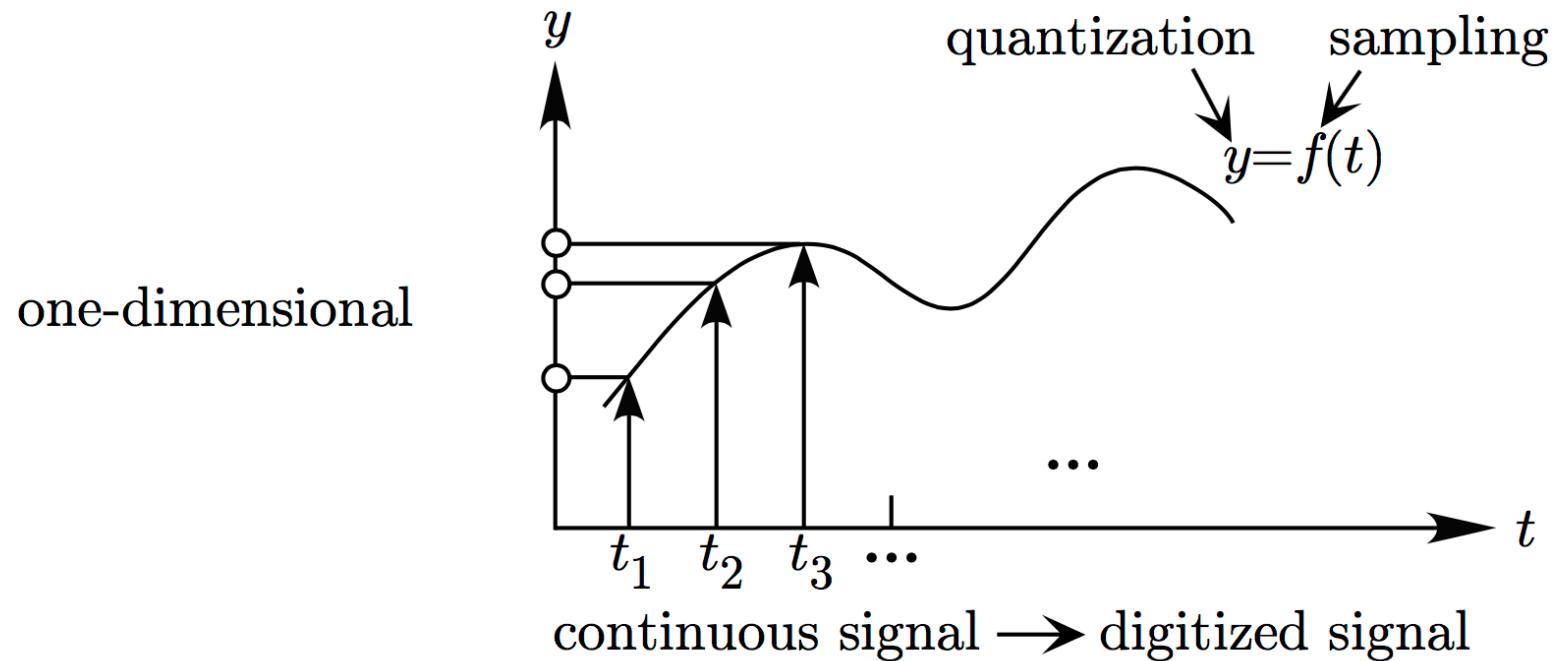
Outline

- Image as a function
- Extracting useful information from Images
 - Edges
 - Smoothing/Removing noise
 - Convolution/Correlation
 - Image Derivatives/Gradient
 - Histogram
 - Filtering (linear)
- *Read Szeliski, Chapter 3.*
- *Read Shah, Chapter 2.*
- *Read/Program CV with Python, Chapters 1 and 2.*

PICTURES

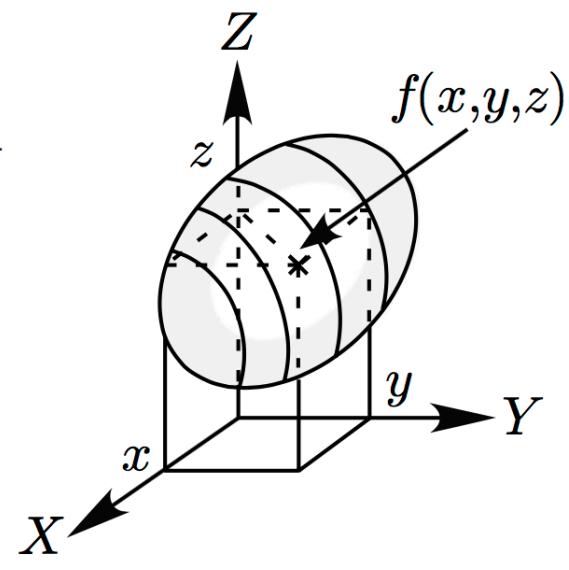
- Computers use discrete form of the pictures
- The process transforming continuous space into discrete space is called **digitization**



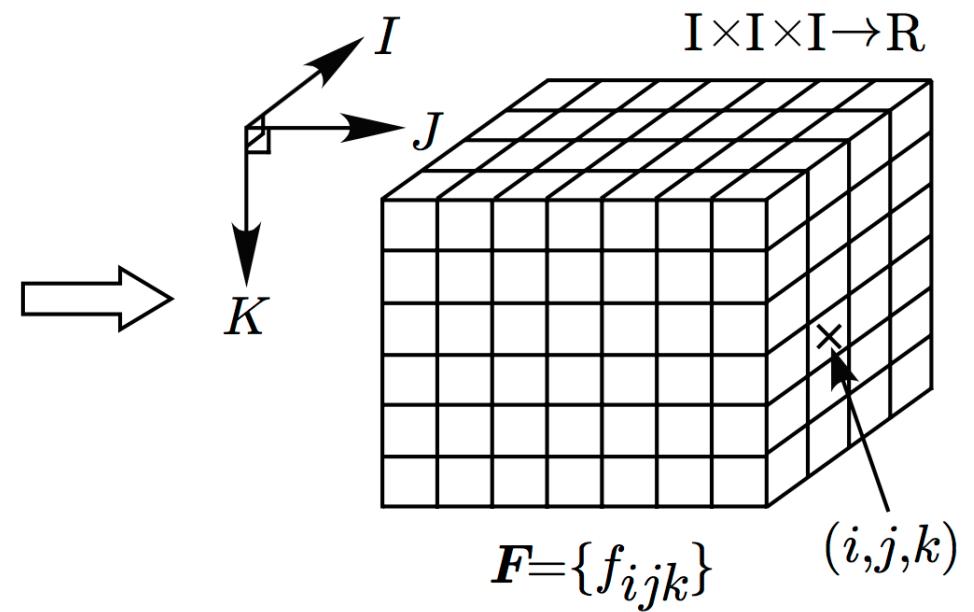


Digitization/Sampling of 3D Image

three-dimensional

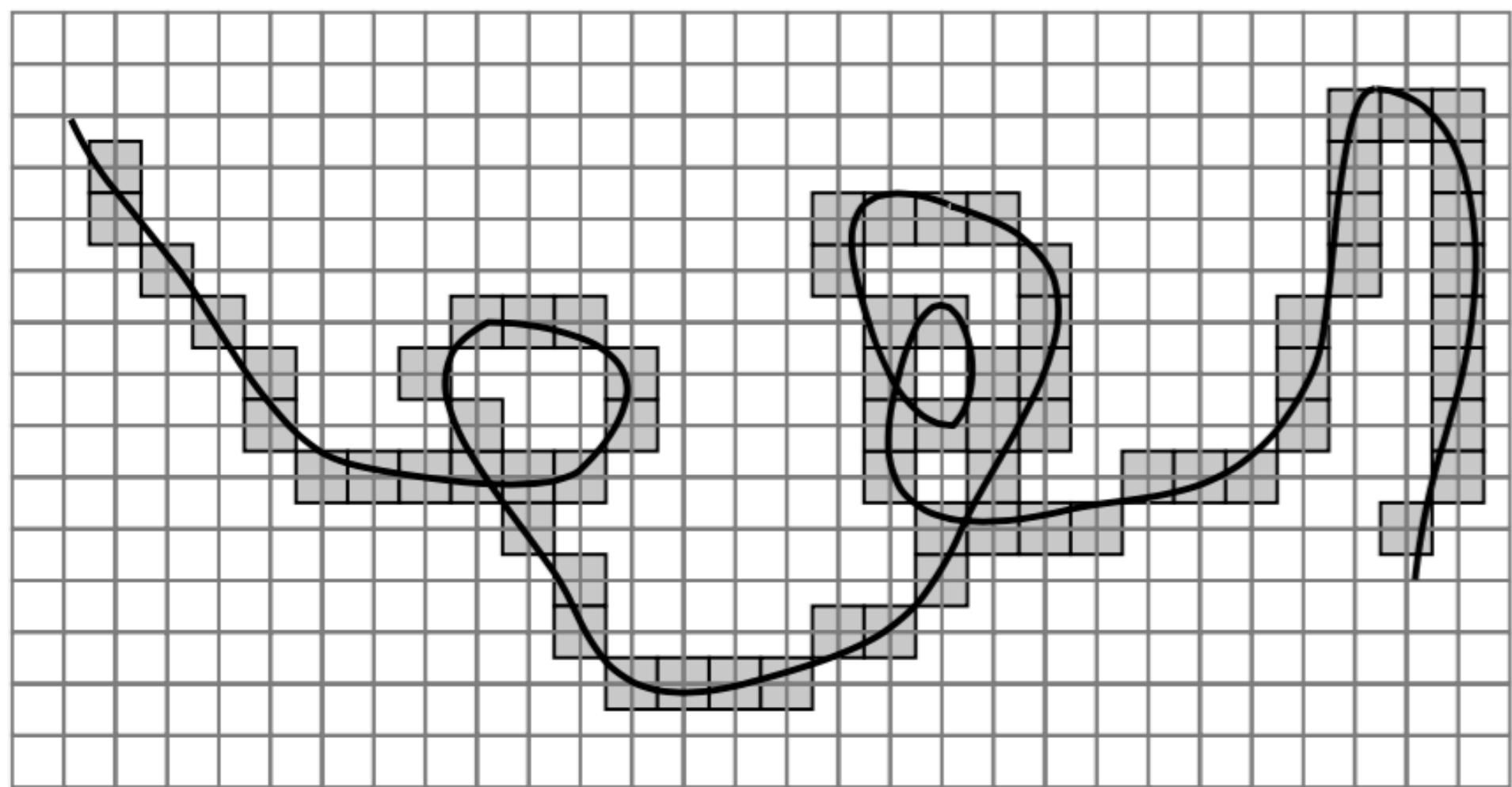


continuous image



digitized image

Digitization of an arc

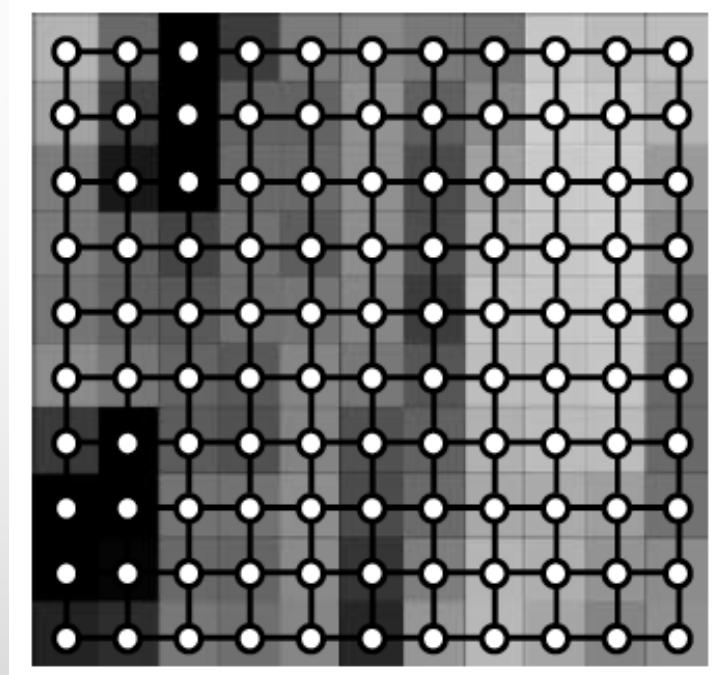
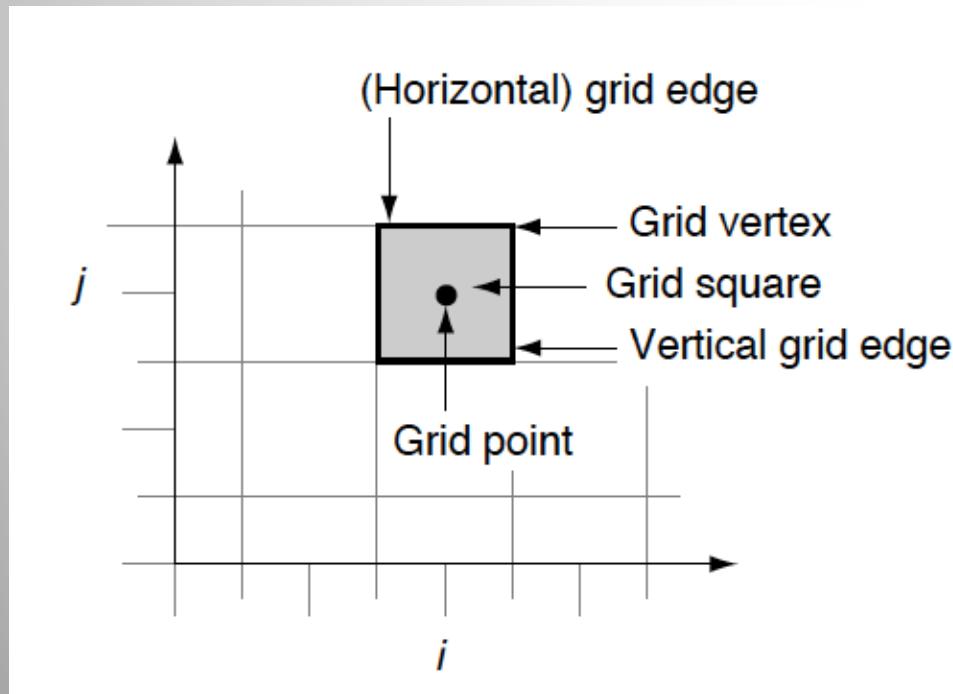


Definition

- A (2D) picture P is a function defined on a (finite) rectangular subset G of a regular planar orthogonal array. G is called (2D) **grid**, and **an element of G is called pixel**. P assigns a value of $P(p)$ to each $p \in G$

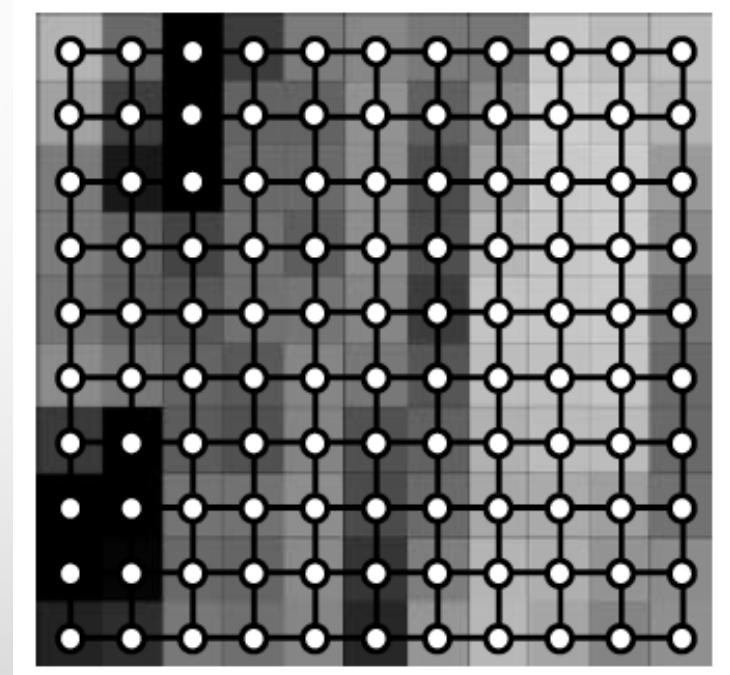
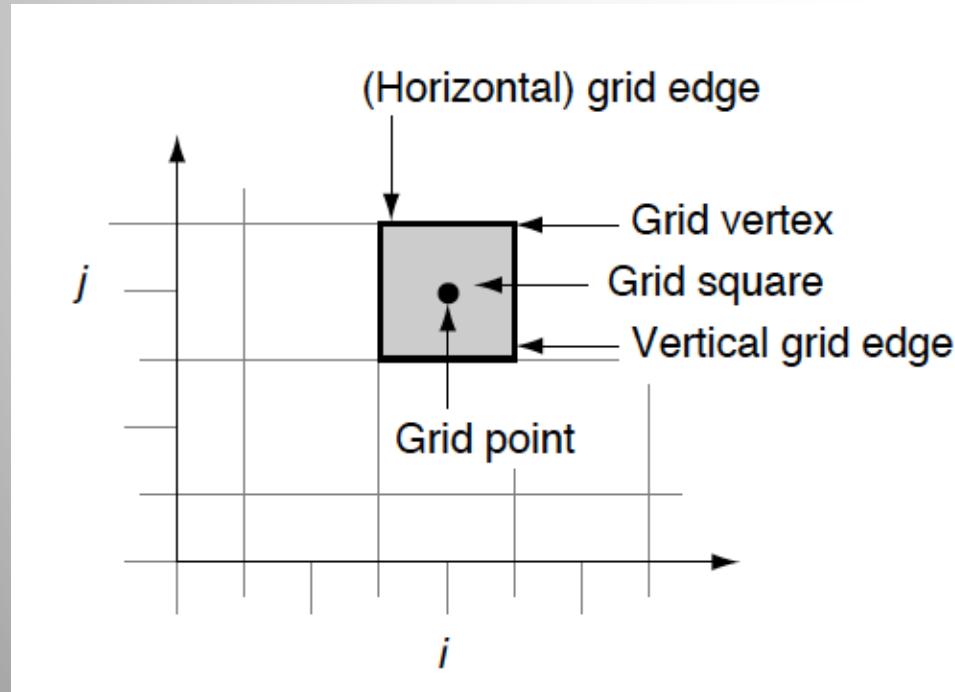
Definition

- A (2D) picture P is a function defined on a (finite) rectangular subset G of a regular planar orthogonal array. G is called (2D) **grid**, and **an element of G is called pixel**. P assigns a value of $P(p)$ to each $p \in G$



Definition

- Pictures are not only sampled, they are also quantized: they may have only a finite number of possible values (i.e., 0 to 255, 0-1, ...)

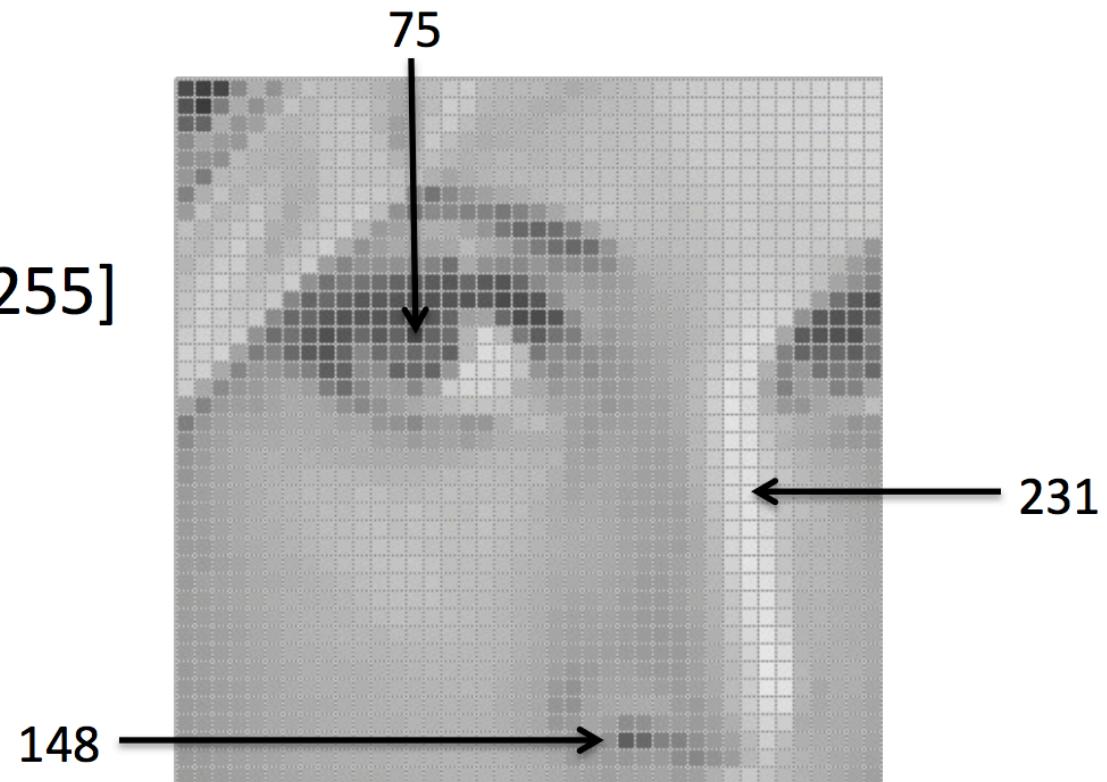


Resolution

- is a display parameter, defined in **dots per inch (DPI)** or equivalent measures of spatial pixel density, and its standard value for recent screen technologies is 72 dpi. Recent printer resolutions are in 300 dpi and/or 600 dpi.



- An image contains discrete number of pixels
 - A simple example
 - Pixel value:
 - “grayscale”
(or “intensity”): [0,255]



- An image contains discrete number of pixels

- A simple example

- Pixel value:

- “grayscale”

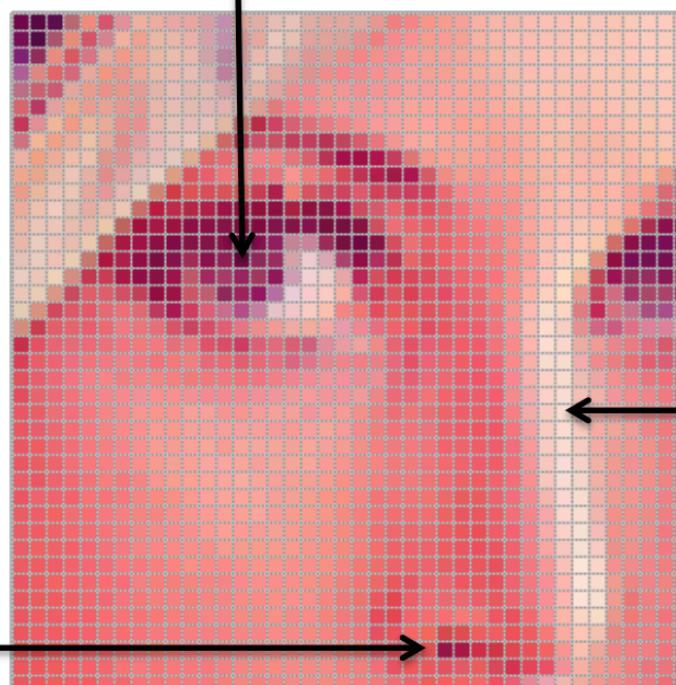
- (or “intensity”): [0,255]

- “color”

- RGB: [R, G, B]

- Lab: [L, a, b]

- HSV: [H, S, V]

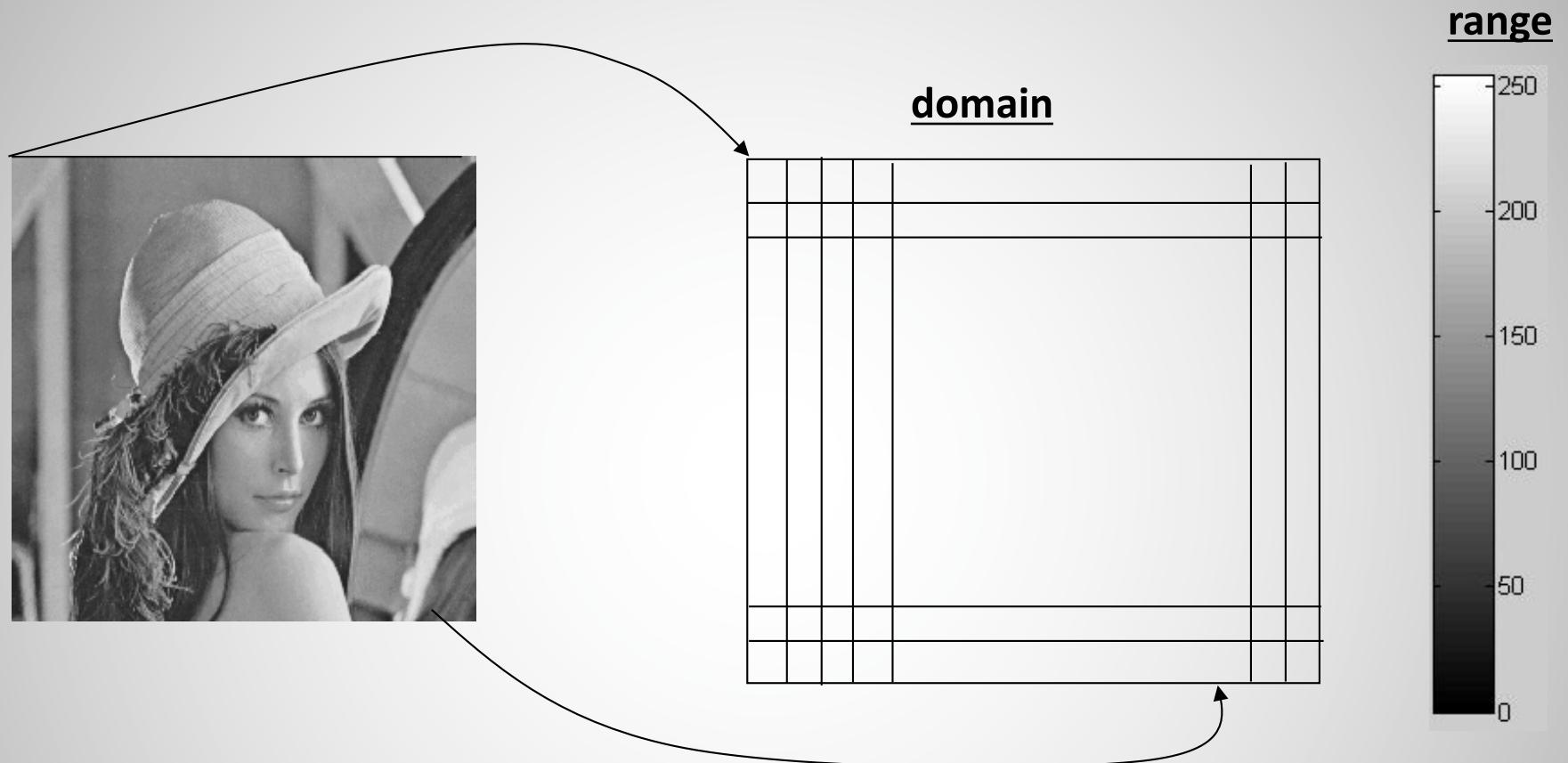


[213, 60, 67]

[90, 0, 53]

[249, 215, 203]

Source: F.F. Li



RGB Channels



Phil Noble / AP



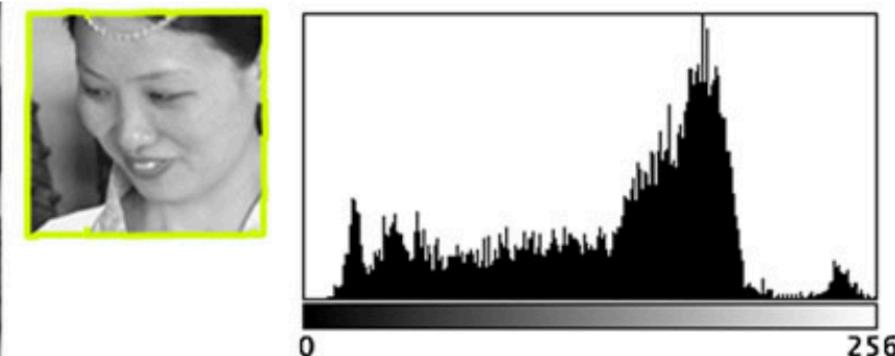
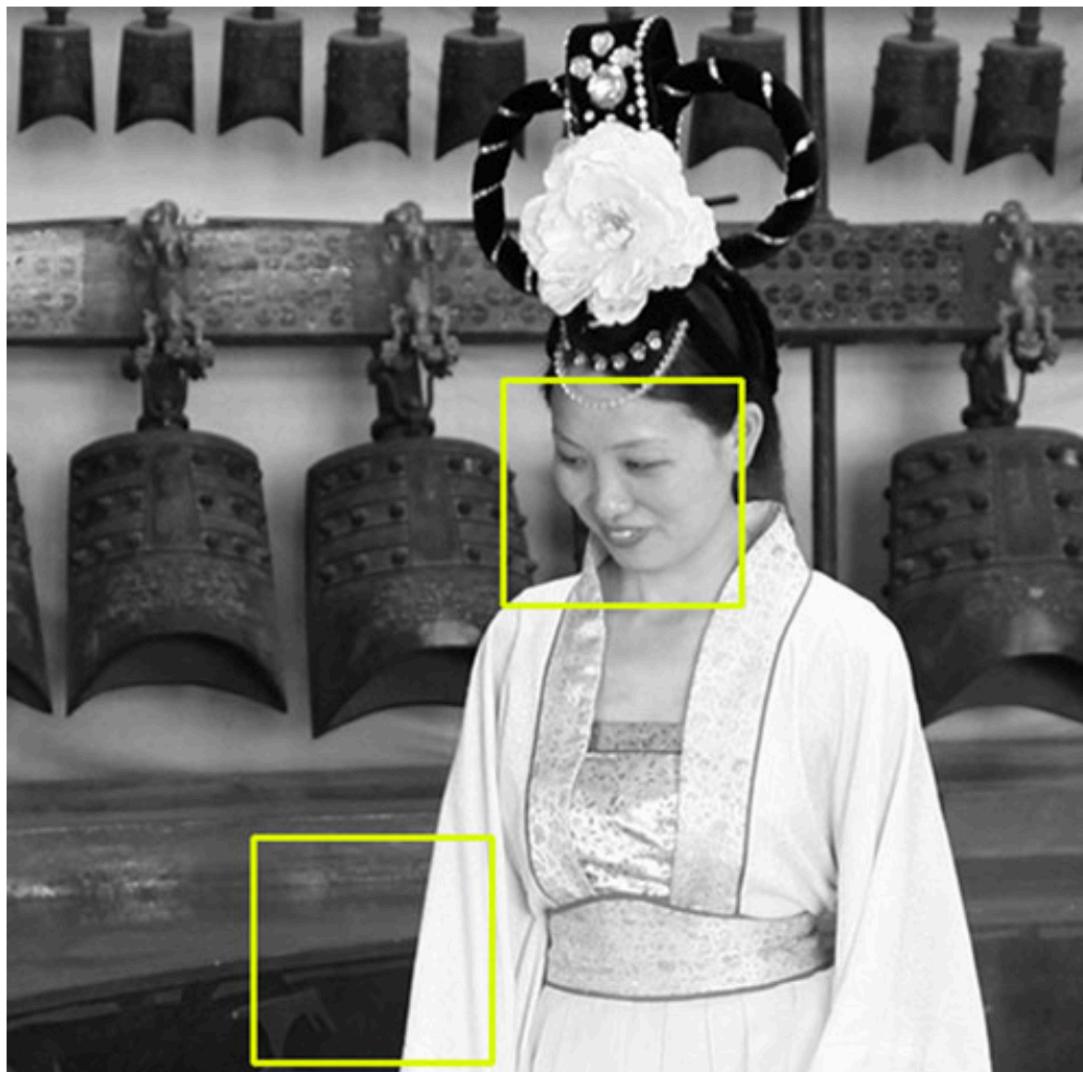
Filters

- Forming a new image whose pixel values are transformed from original pixel values
- Goal is to extract useful information from images, or transform images into another domain where we can modify/enhance image properties
 - Edges, corners, blobs, regions,....
 - Super-resolution, in-painting, denoising,..
 - Texture
 - Patterns
 - Photo editing, tone mapping, abstraction,...
- **Linear filtering:** The value of output \rightarrow is calculated by a linear combination of density values on an input image in the neighborhood

Image Histogram



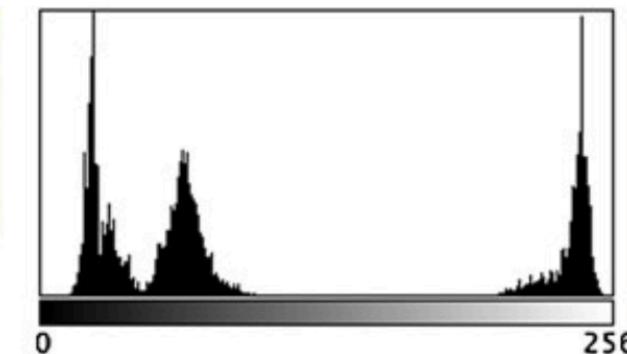
Histogram Example



Count: 10192
Mean: 133.711
StdDev: 55.391
Min: 9
Max: 255
Mode: 178 (180)



Count: 10192
Mean: 104.637
StdDev: 89.862
Min: 11
Max: 254
Mode: 23 (440)



Histogram

- Histogram of an image provides the frequency of the brightness (intensity) value in the image.
- Provides a natural bridge between images and a probabilistic description.

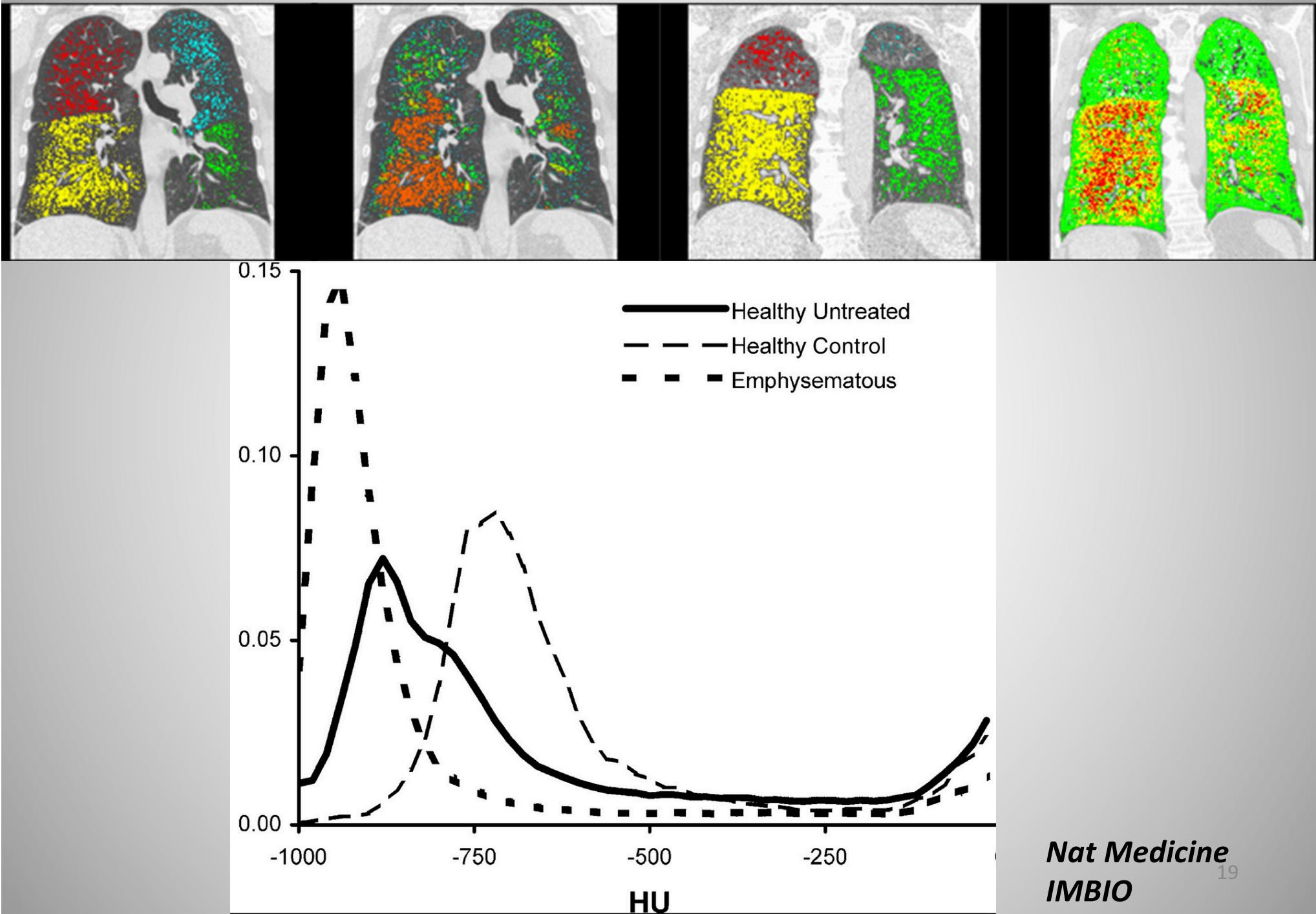
– Ex. Probability of pixel (x,y) that has a brightness (intensity) value z

Pseudo-Code for Histogram

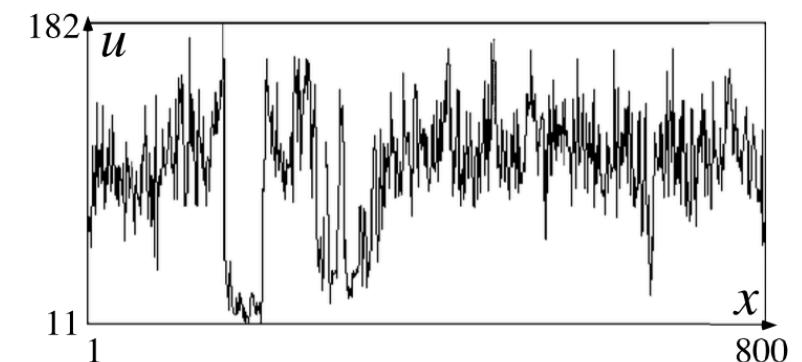
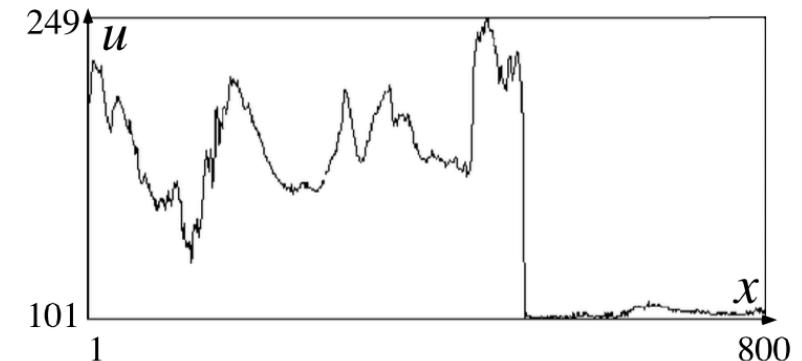
1. Create an array \mathbf{h} with zero in its elements
2. For all pixel locations (x,y) of the image A , increment $\mathbf{h}(A(x,y))$ by 1

$$h(i) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \delta(f(m, n) - i), \quad i = 0, 1, \dots, P-1,$$

$$\delta(w) = \begin{cases} 1 & w = 0, \\ 0 & \text{otherwise.} \end{cases}$$



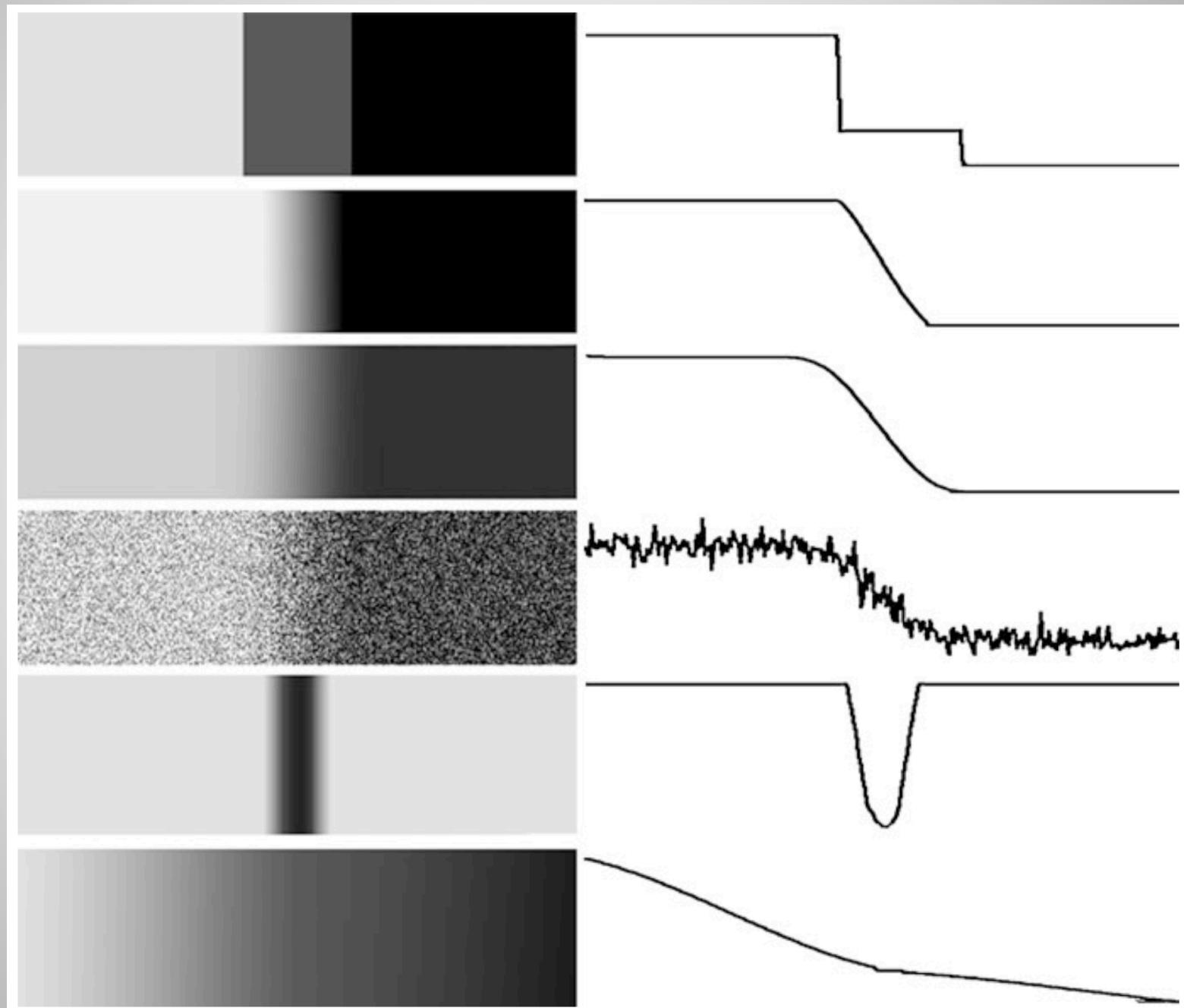
Intensity profiles for selected (two) rows



EDGES

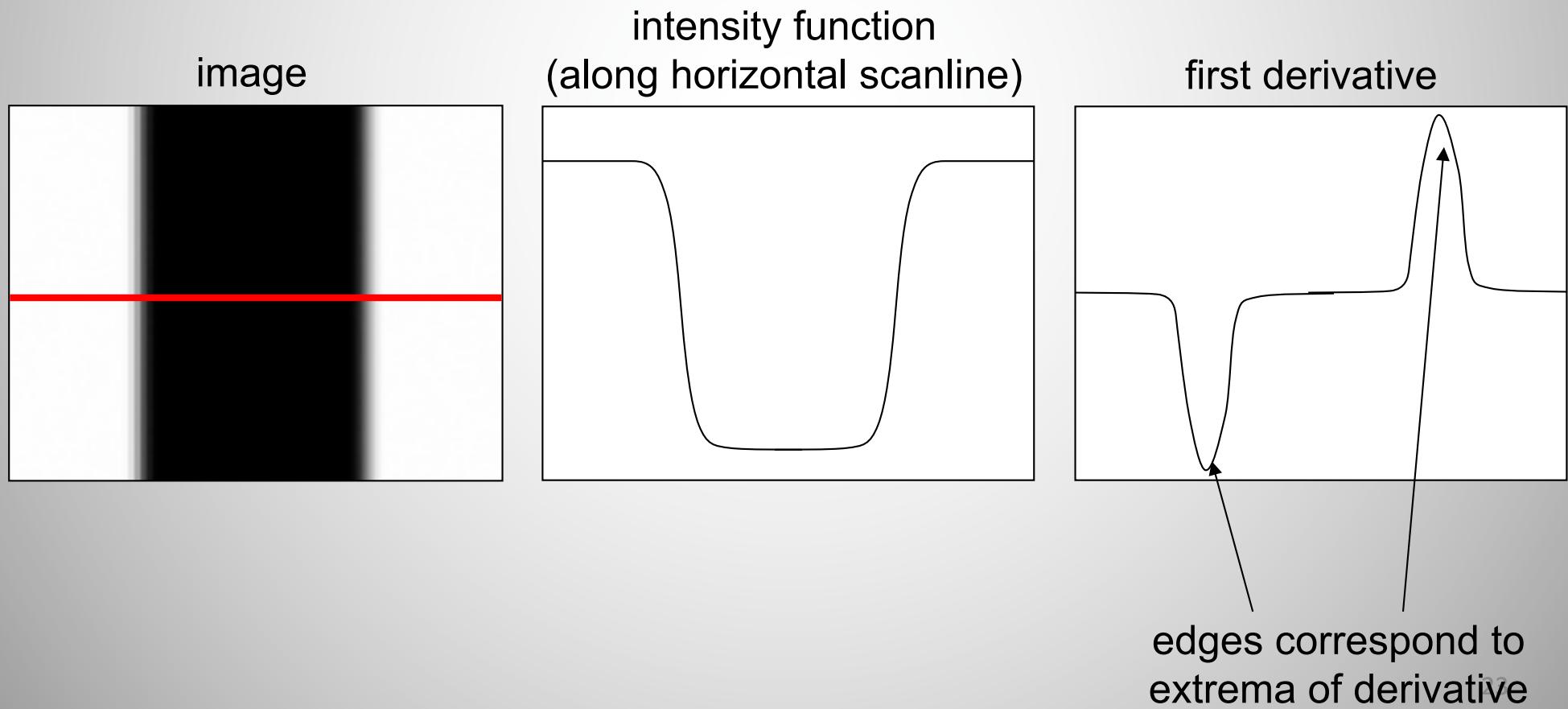
- Discontinuities in images are features that are often useful for initializing an image analysis procedure.
- Edges are important information for understanding an image; by moving “non-edge” data we also simplify the data.

Edge Models



Characterizing Edges

- An edge is a place of rapid change in the image intensity function



Derivatives and Average

- **Derivative:** rate of change
 - Speed is a rate of change of a distance, $X=V.t$
 - Acceleration is a rate of change of speed, $V=a.t$
- **Average:** mean
 - Dividing the sum of N values by N

Derivative

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

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

$$y = \sin x + e^{-x}$$

$$\frac{dy}{dx} = \cos x + (-1)e^{-x}$$

Discrete Derivative

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

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

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

Discrete Derivative / Finite Difference

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

Backward difference

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

Forward difference

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

Central difference

Example: Finite Difference

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

$$f''(x) = 0 \quad 5 \quad 10 \quad 5 \quad 15 \quad -20 \quad 5 \quad 0$$

Derivative Masks

Backward difference $[-1 \quad 1]$

Forward difference $[1 \quad -1]$

Central difference $[-1 \quad 0 \quad 1]$

Derivative in 2-D

Given function

$$f(x, y)$$

Gradient vector

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude

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

Gradient direction

$$\theta = \tan^{-1} \frac{f_x}{f_y}$$

Derivative of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad f_y \Rightarrow \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

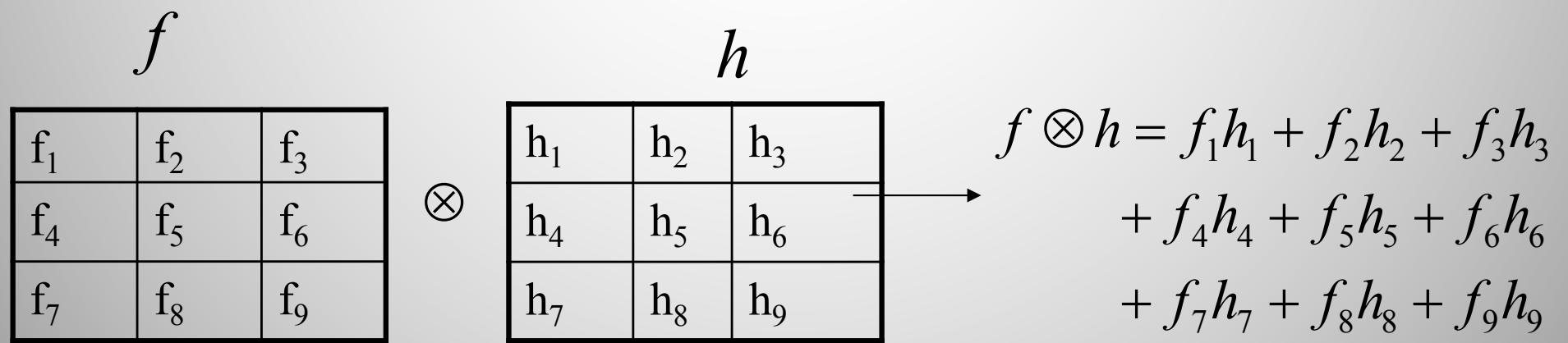
$$I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{10} & \boxed{10} & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Correlation (linear relationship)

$$f \otimes h = \sum_k \sum_l f(k, l)h(k, l)$$

f = Image

h = Kernel



Convolution

$$f * h = \sum_k \sum_l f(k, l)h(-k, -l)$$

f = Image

h = Kernel

f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

f

h_7	h_8	h_9
h_4	h_5	h_6
h_1	h_2	h_3

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

h_9	h_8	h_7
h_6	h_5	h_4
h_3	h_2	h_1

*

$Y - flip$

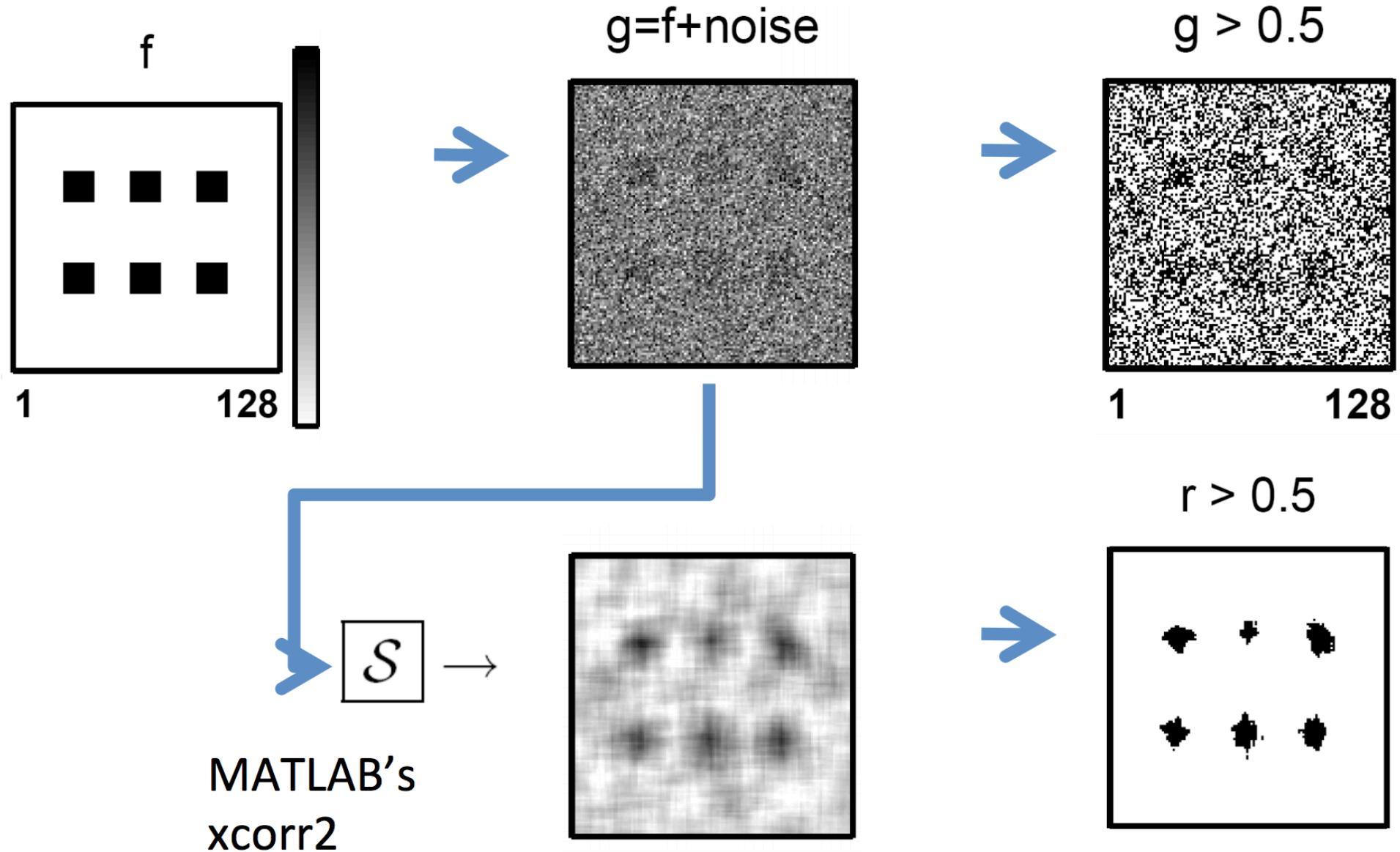
$$\begin{aligned} f * h = & f_1 h_9 + f_2 h_8 + f_3 h_7 \\ & + f_4 h_6 + f_5 h_5 + f_6 h_4 \\ & + f_7 h_3 + f_8 h_2 + f_9 h_1 \end{aligned}$$

Correlation and Convolution

- Convolution is **associative**

$$F * (G * I) = (F * G) * I$$

(Cross) correlation – example



Correlation and Convolution

- Convolution is a filtering operation, expresses the amount of overlap of one function as it is shifted over another function
- Correlation compares the similarity of two sets of data. (relatedness of the signals!)

Averages

- Mean

$$I = \frac{I_1 + I_2 + \dots + I_n}{n} = \frac{\sum_{i=1}^n I_i}{n}$$

- Weighted mean

$$I = \frac{w_1 I_1 + w_2 I_2 + \dots + w_n I_n}{n} = \frac{\sum_{i=1}^n w_i I_i}{n}$$

Laplacian-Second Order Derivative

- Will be doing in your **PA#1**

Difference of Gaussians \sim Laplacian

Gaussian

0.01	0.08	0.01
0.08	0.64	0.08
0.01	0.08	0.01

$\frac{\partial}{\partial x}$

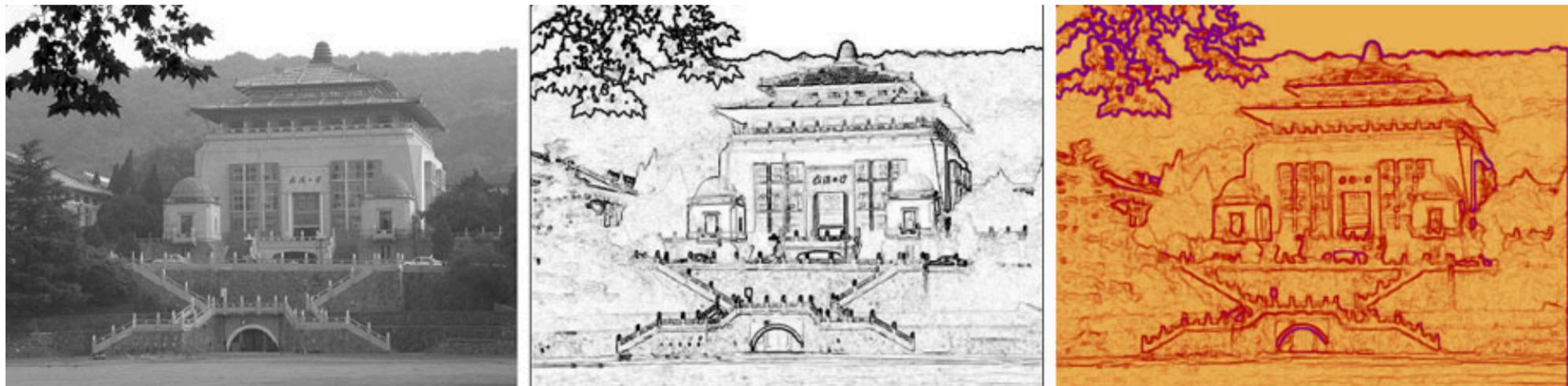
0.05	0	-0.05
0.34	0	-0.34
0.05	0	-0.05

$\frac{\partial}{\partial y}$

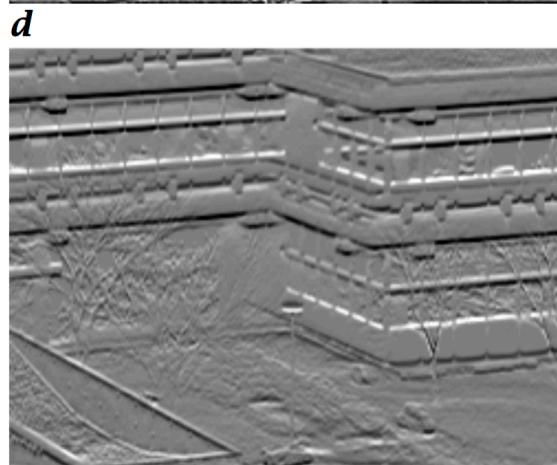
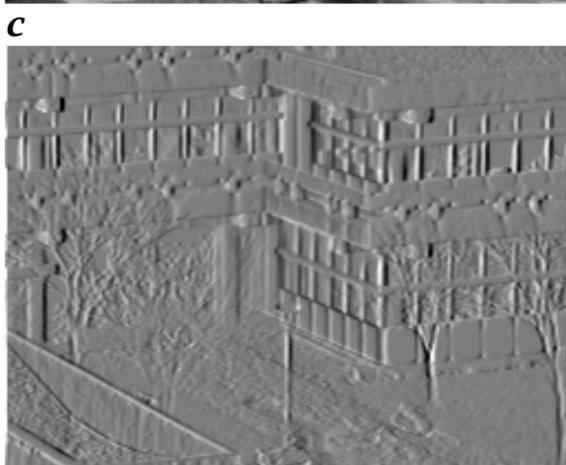
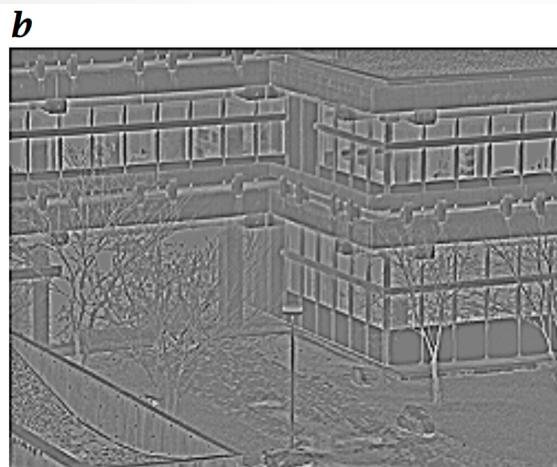
0.05	0.34	0.05
0	0	0
-0.05	-0.34	-0.05

∇^2

0.3	0.7	0.3
0.7	-4	0.7
0.3	0.7	0.3



Wuhan University.



- a. Original image
- b. Laplacian operator
- c. Horizontal derivative
- d. Vertical derivative

Noise and Denoising

- Low-pass filtering is normally thought of as the elimination of signal component with high spatial frequencies.



Salt-pepper

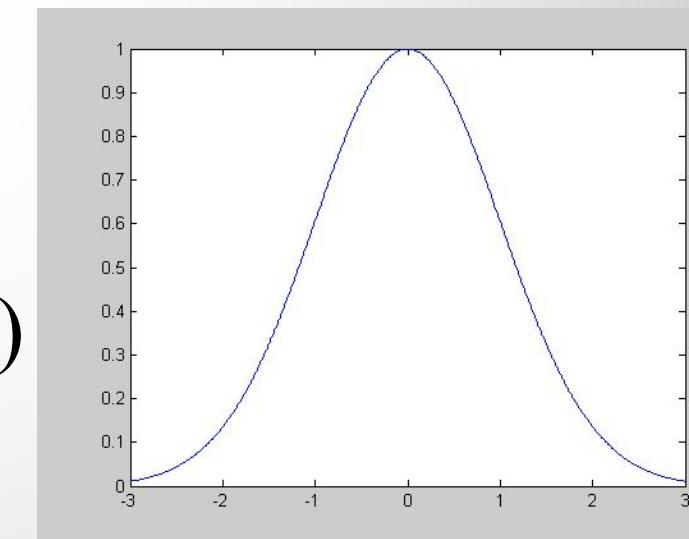
Noise

- $I_{\text{observed}} = I_{\text{original}} + n$ additive
- $I_{\text{observed}} = I_{\text{original}} * n$ multiplicative
- Light/brightness variations, camera hardware/lenses, surface reflectance
- Random in nature, occurring with some **probability**
- It has a distribution (Gaussian (white), Poisson,...)

Gaussian Noise



$g(n)$

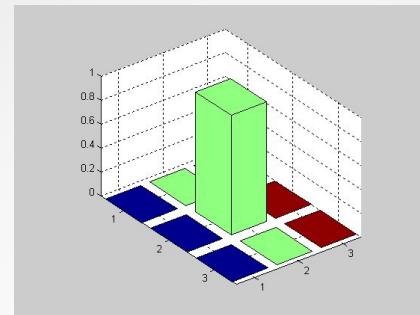


Probability Distribution
 n is a random variable

Why Gaussian Assumption?

- Most common natural model
- Smooth function, it has infinite number of derivatives
- It is Symmetric
- Fourier Transform of Gaussian is Gaussian.
- Convolution of a Gaussian with itself is a Gaussian.
- Gaussian is separable; 2D convolution can be performed by two 1-D convolutions
- There are cells in eye that perform Gaussian filtering.

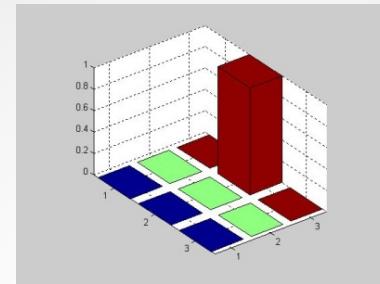
Filtering Examples - 1



$$\begin{matrix} * & \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} & = & \text{Output Image} \end{matrix}$$



Filtering Examples - 2



$$\begin{matrix} * & \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline \end{array} \end{matrix}$$



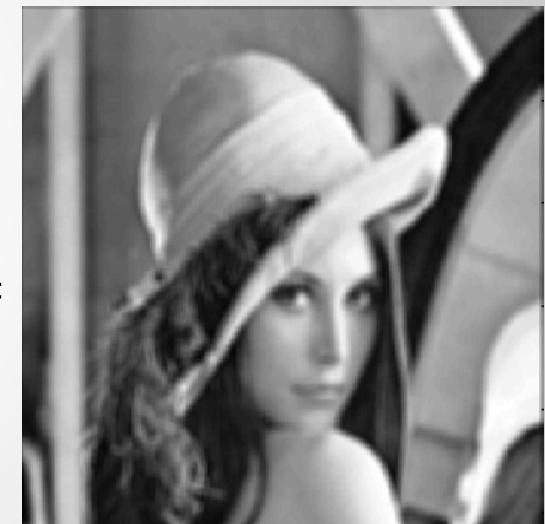
Filtering Examples - 3



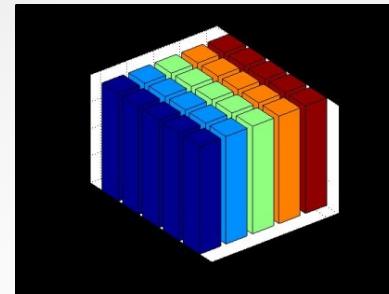
$$*\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

=



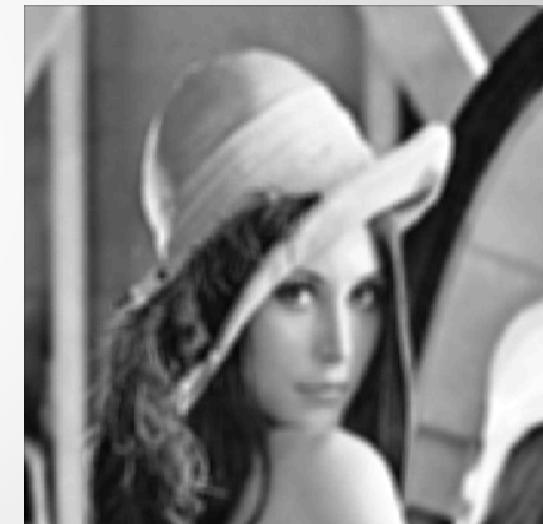
Filtering Examples - 4



$$*\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

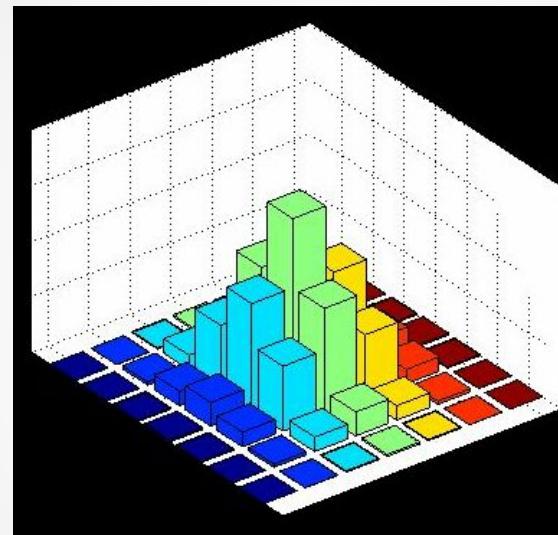
=



Filtering Examples - 5



*



=



Gaussian Smoothing

Filtering Examples - 6



Gaussian Smoothing



Smoothing by Averaging

Filtering Examples - 7



After additive
Gaussian Noise



After Averaging



After Gaussian Smoothing

Noise suppression: Image Filtering

Gaussian

0.01	0.08	0.01
0.08	0.64	0.08
0.01	0.08	0.01

- Gaussian Filtering:

$$S = (C, f) \longrightarrow S_F = (C, f_F)$$

$$f_F(\nu) = \sum_{\nu_i \in N(\nu)} w_i f(\nu_i)$$

Example: box filter

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect
(remove sharp features)

$g[\cdot, \cdot]$

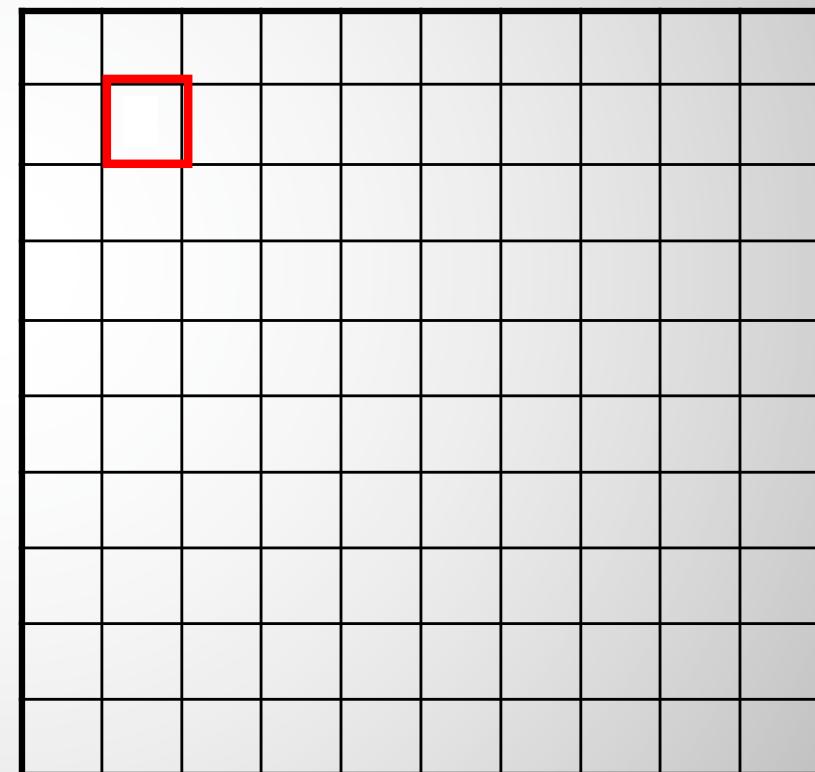
$$\frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Image filtering

$f[.,.]$

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	90	0	90	90	90	0
0	0	0	90	90	90	90	90	0
0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$h[.,.]$



$$g[\cdot, \cdot] \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Credit: S. Seitz

Image filtering

$f[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

0	10									

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Credit: S. Seitz

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Image filtering

$f[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

			0	10	20					

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

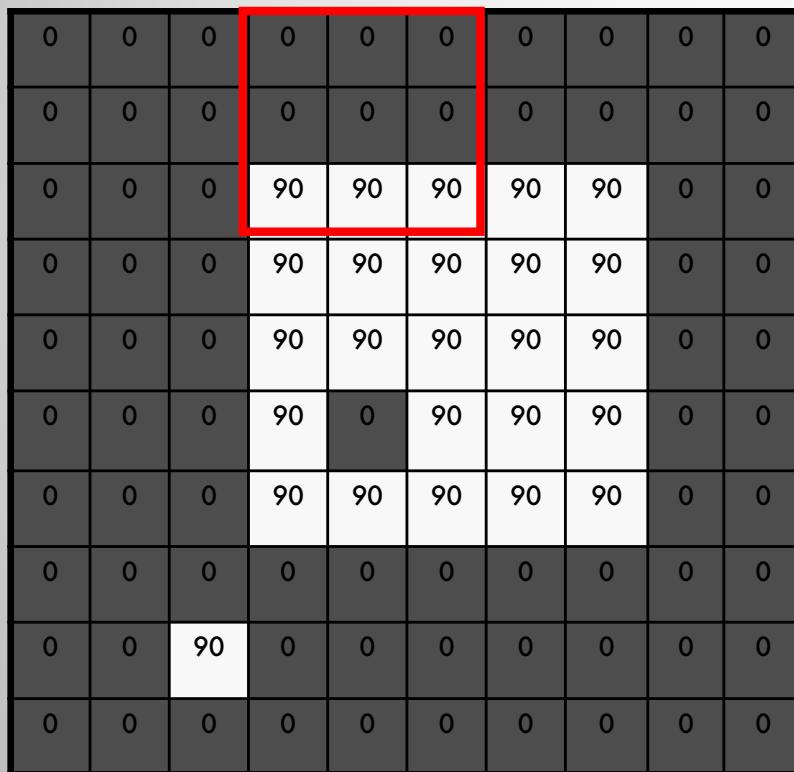
Credit: S. Seitz

$$g[\cdot, \cdot] \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

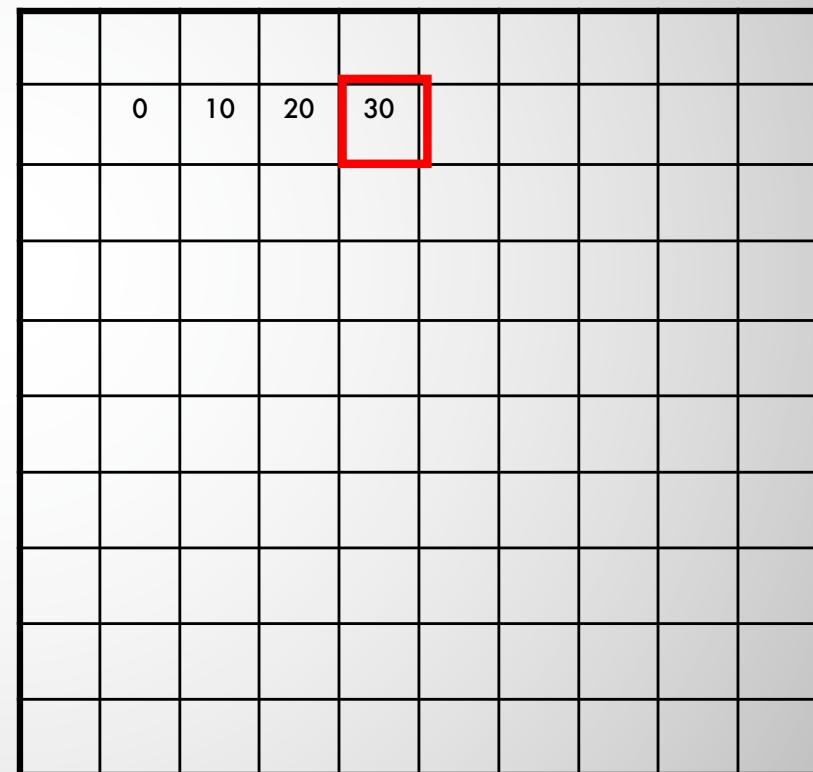
Image filtering

$$g[\cdot, \cdot] \frac{1}{9}$$

$$f[\cdot,\cdot]$$



$$h[.,.]$$



$$h[m,n] = \sum_{k,l} g[k,l] f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

 $f[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

 $h[.,.]$

	0	10	20	30	30					

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Credit: S. Seitz

$$g[\cdot, \cdot] \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Image filtering

$f[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30					

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Credit: S. Seitz

Image filtering

 $f[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

 $h[.,.]$

	0	10	20	30	30					

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Credit: S. Seitz

$$g[\cdot, \cdot] \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Image filtering

$$g[\cdot, \cdot] \quad \frac{1}{9} \begin{array}{|ccc|} \hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \hline \end{array}$$

$$f[., .]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[., .]$$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

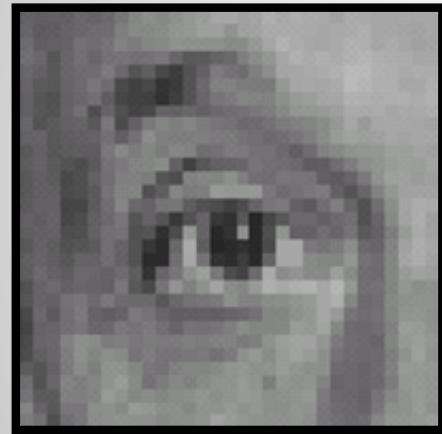
$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Credit: S. Seitz

Smoothing with Box Filter



Practice with kernels

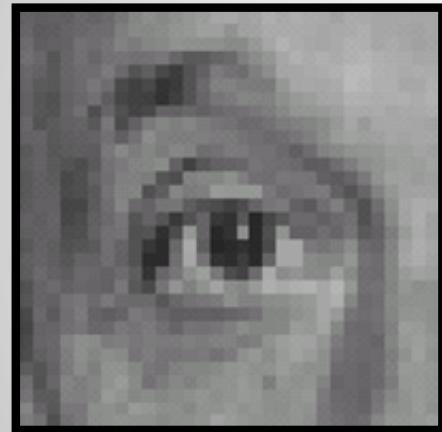


Original

0	0	0
0	1	0
0	0	0

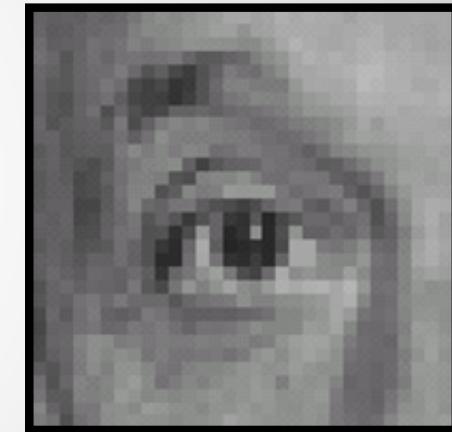
?

Practice with kernels



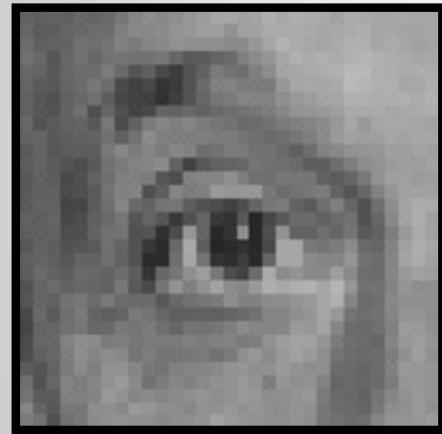
Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)

Practice with kernels

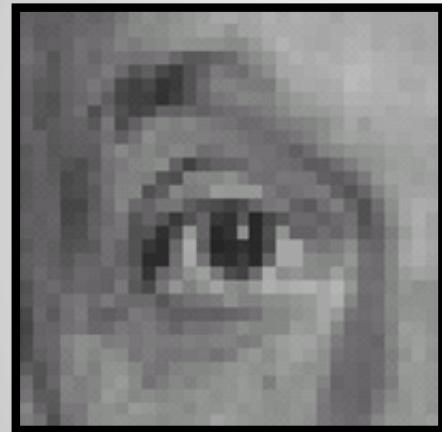


0	0	0
0	0	1
0	0	0

?

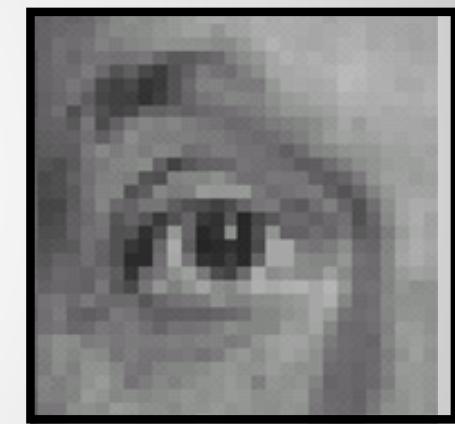
Original

Practice with kernels



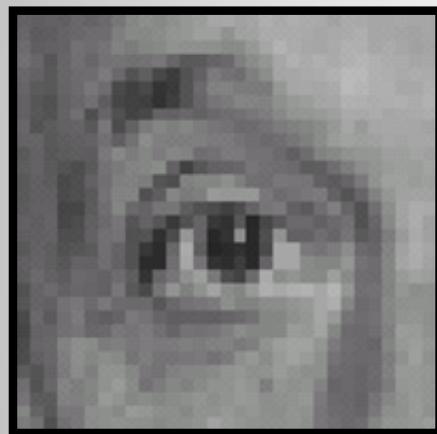
Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Practice with kernels



0	0	0
0	2	0
0	0	0

-

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

1	1	1
1	1	1
1	1	1

?

(Note that filter sums to 1)

Practice with kernels



$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

-

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

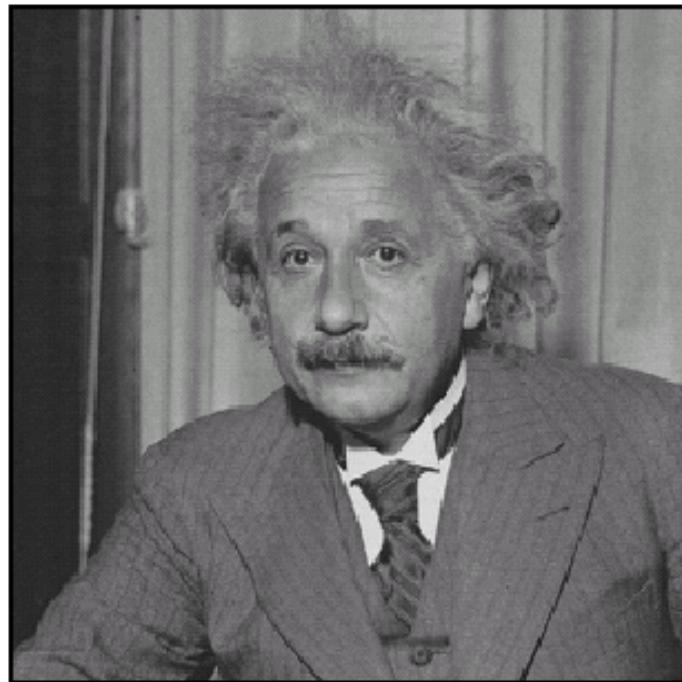


Original

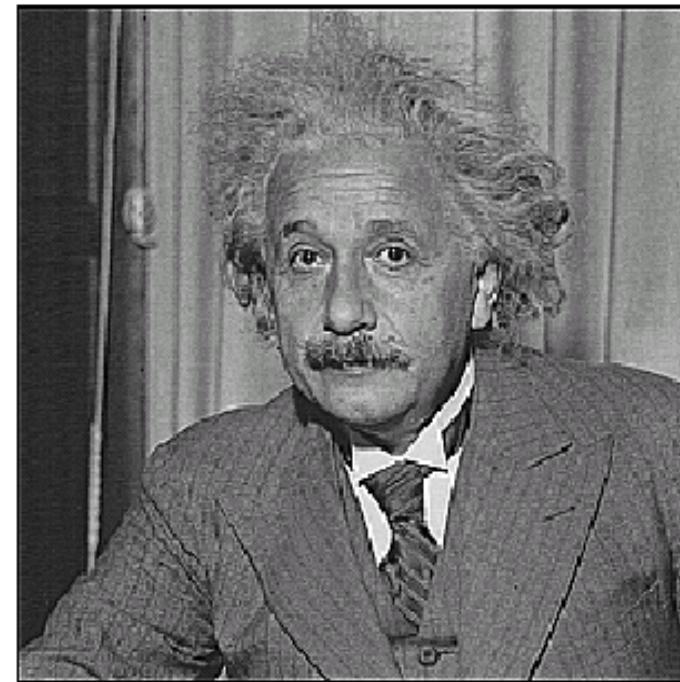
Sharpening filter

- Accentuates differences with local average

Sharpening Filter



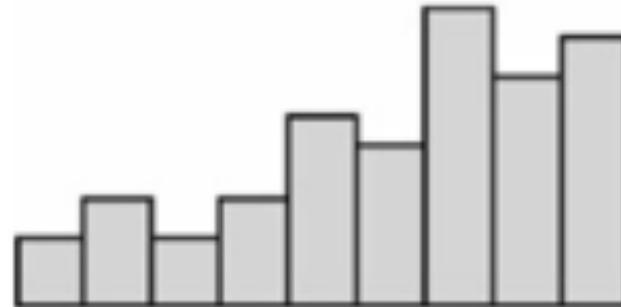
before



after

Unsharp Masking

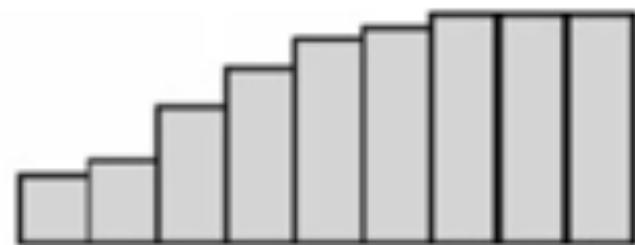
Lower Noise, Higher Contrast



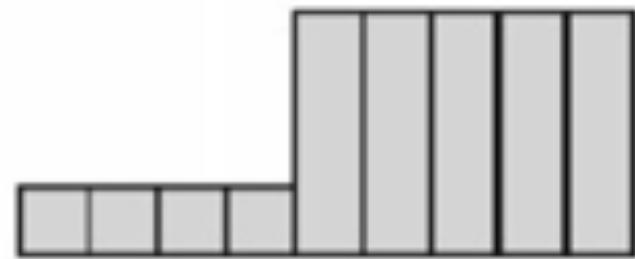
histogram

smoothness
only

smoothness
and few intensity
changes



histogram



Unsharp Masking

- Not only noise removal, but edge enhancement is necessary!

$$I = g * I + (I - g * I)$$



Smoothed image
(low pass)



Edge enhanced image
(high pass)

$$I' = g * I + (1 + \alpha)(I - g * I) \quad \alpha > 0$$

Reminder: Edges are located in high frequency of the images!

Hand X-ray Unsharp Masking ($\alpha=0.5$)

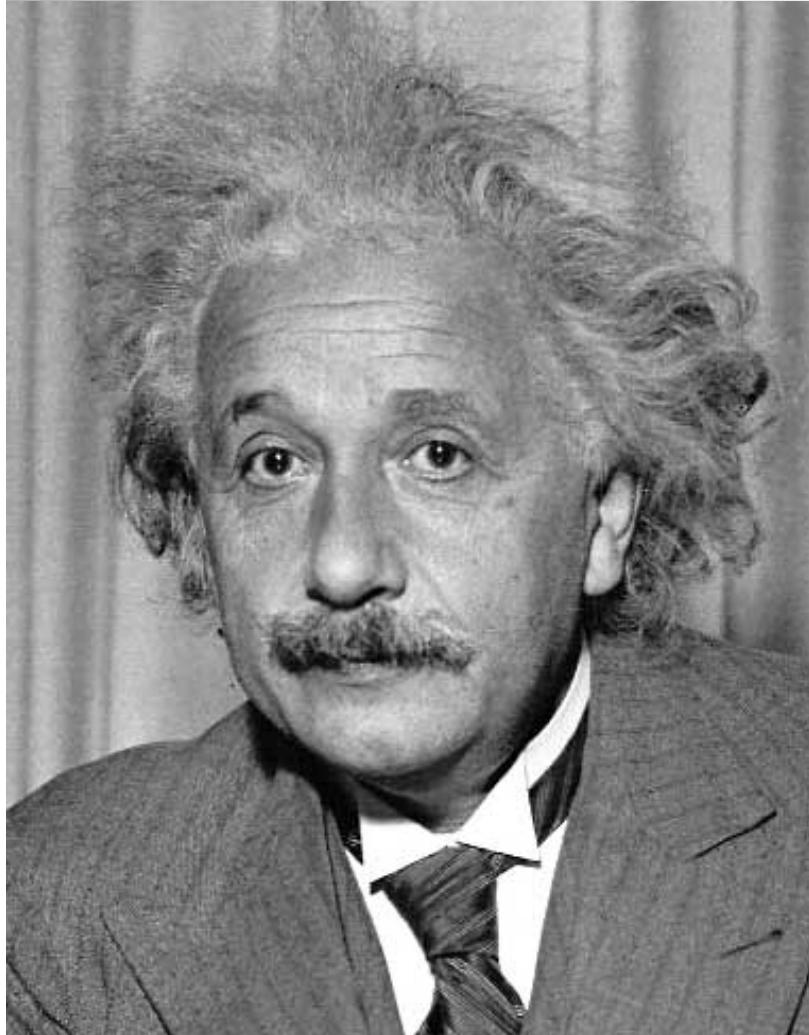


Original Image



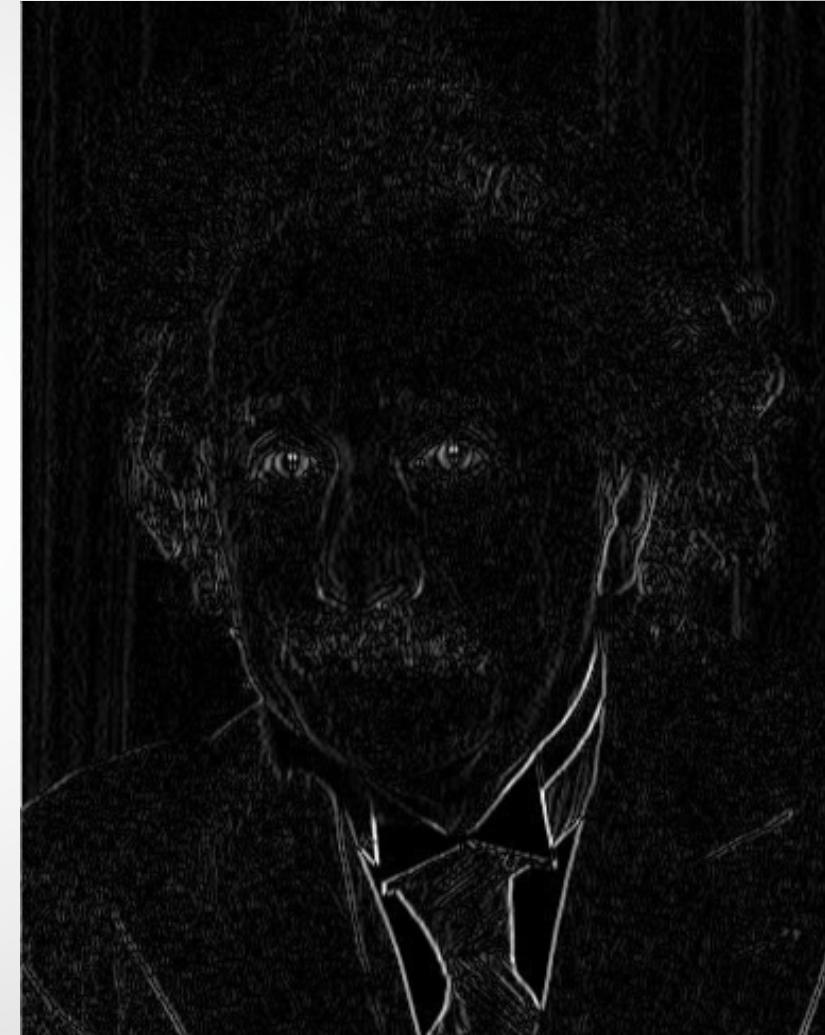
Enhanced Image

Sobel Filtering



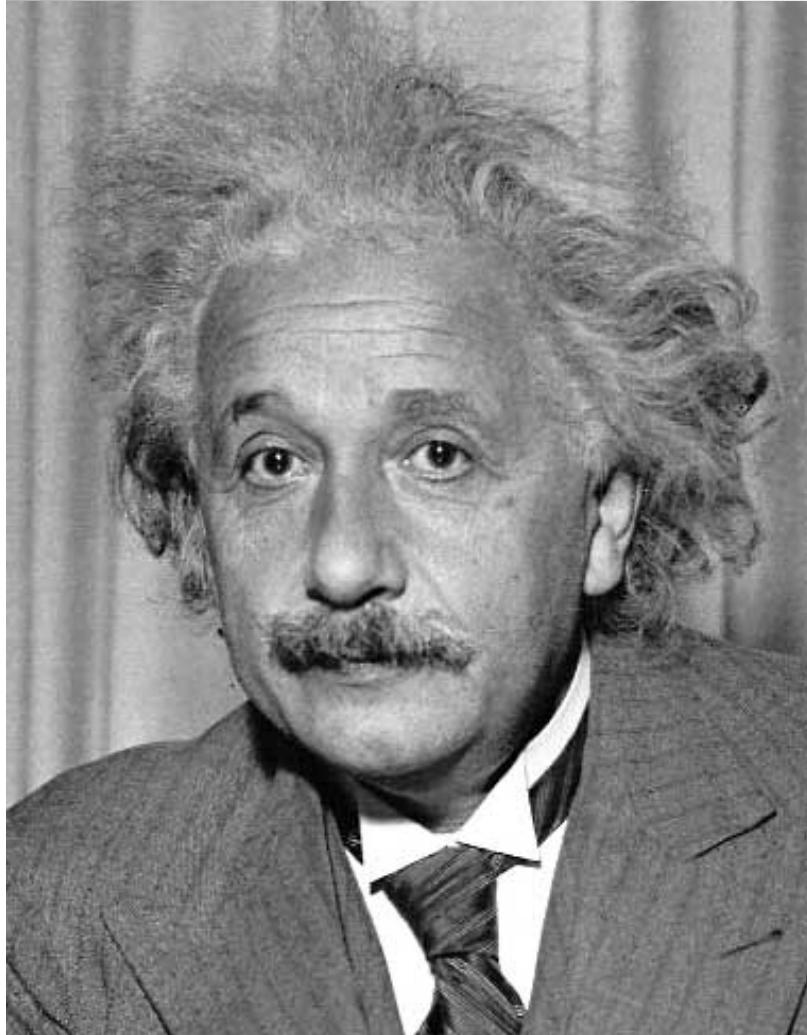
1	0	-1
2	0	-2
1	0	-1

Sobel



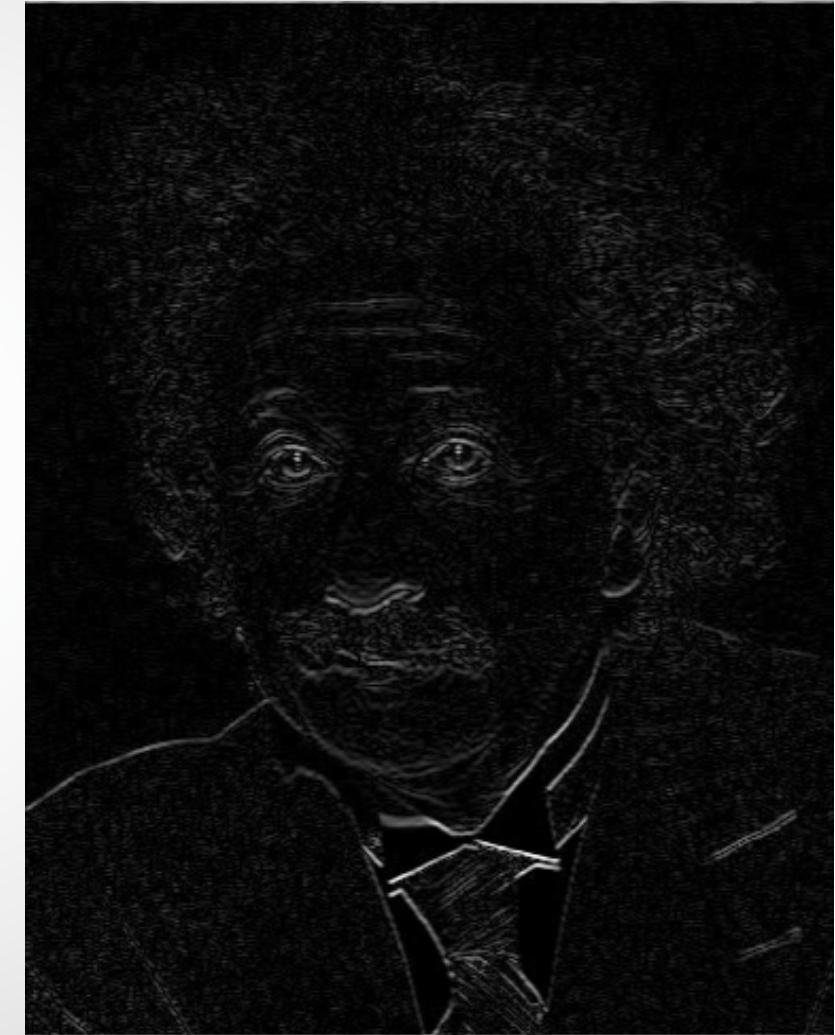
Vertical Edge
(absolute value) ⁷²

Sobel Filtering



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

Sobel



Horizontal Edge
(absolute value)

Key Properties of Linear Filters

Linearity:

$$\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$$

Shift invariance: same behavior regardless of pixel location

$$\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$$

Any linear, shift-invariant operator can be represented as a **convolution**

More properties

- Commutative: $a * b = b * a$
 - Conceptually no difference between filter and signal
 - But particular filtering implementations might break this equality
- Associative: $a * (b * c) = (a * b) * c$
 - Often apply several filters one after another: $((a * b_1) * b_2) * b_3$
 - This is equivalent to applying one filter: $a * (b_1 * b_2 * b_3)$
- Distributes over addition: $a * (b + c) = (a * b) + (a * c)$
- Scalars factor out: $ka * b = a * kb = k(a * b)$
- Identity: unit impulse $e = [0, 0, 1, 0, 0]$,
 $a * e = a$

Median Filters

- A **Median Filter** operates over a window by selecting the median intensity in the window.
- PA1. What advantage does a median filter have over a mean filter?
- PA1. Is a median filter a kind of convolution?

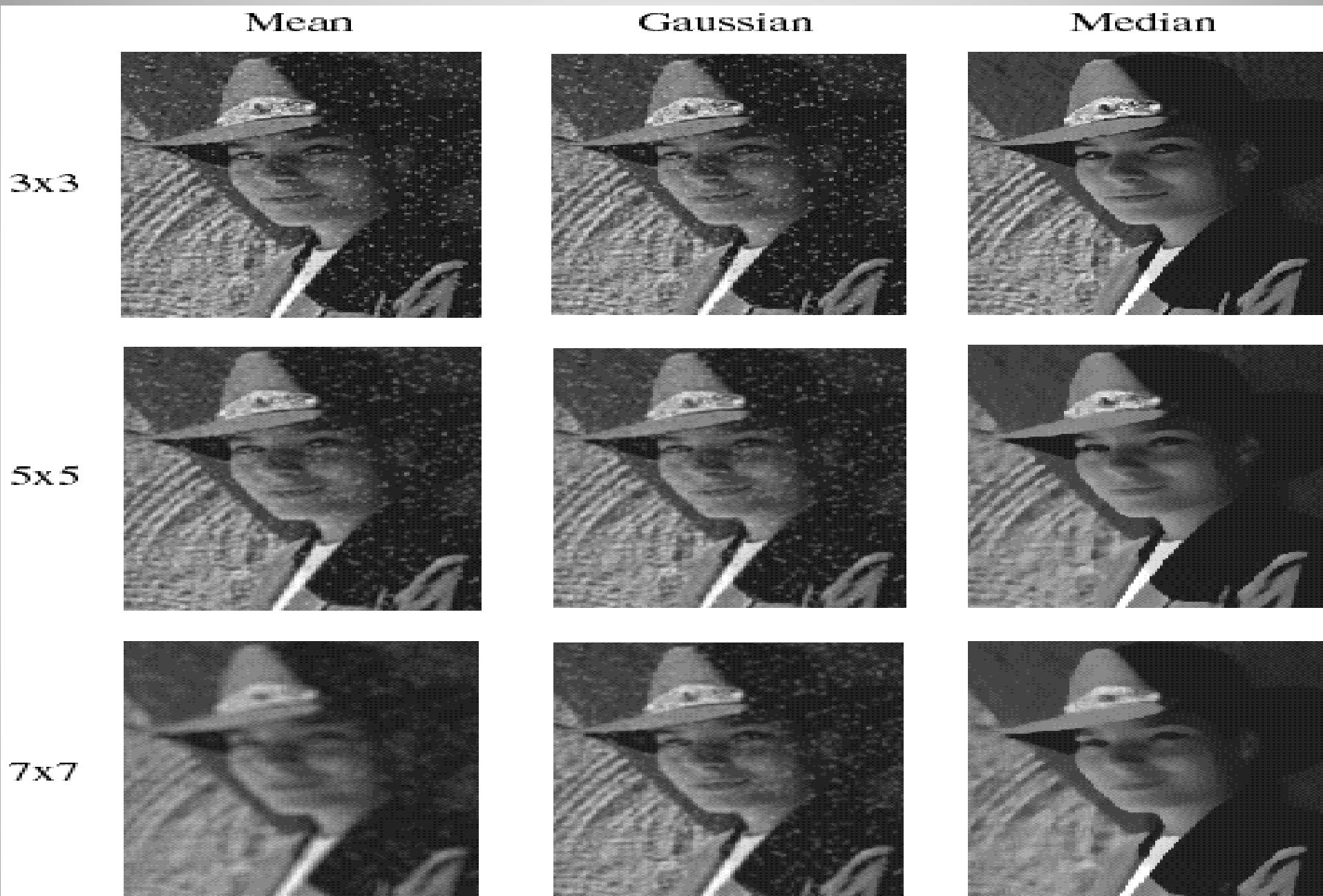
Median Filtering (Details)



[8 8 8 8 8 8 8 8 255]

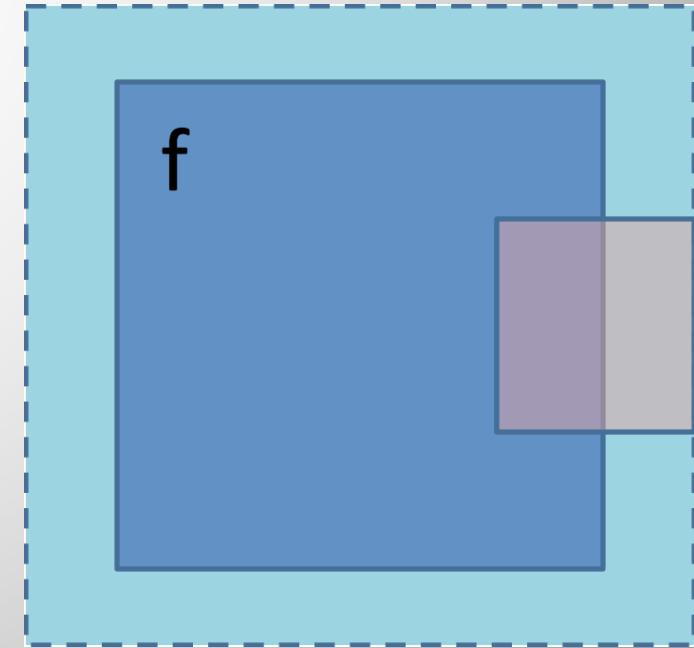


median



Practical issues

- How big should the filter be?
 - Values at edges should be near zero
 - Rule of thumb for Gaussian: set filter half-width to about 3σ
- Zero Padding and Edge Effects
 - Kernel should include edges
 - Mirroring is another way



Questions?