پردازش تصاویر دیجیتال

مهدی تیموری

mehditeimouri@ut.ac.ir

بسم الله الرحمن الرحیم

منابع:

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 2008

William K. Pratt, *Digital Image Processing*, 3rd ed., John Wiley & Sons, 2001

Hwei P. Hsu, *Signals and Systems*, 1st ed., McGraw Hill, 2005.

نحوه ارزیابی:

10 Homework Assignments **Policy for late submissions: Maximum grace period of 25 hours for each homework assignment with a penalty of 4% per hour.** **Policy for early submissions: Maximum bonus of 25% within the first 25 hours after the assignment is uploaded. 1% per hour is reduced from the bonus.**	**50%**
6 Exams	50%

مقدمه

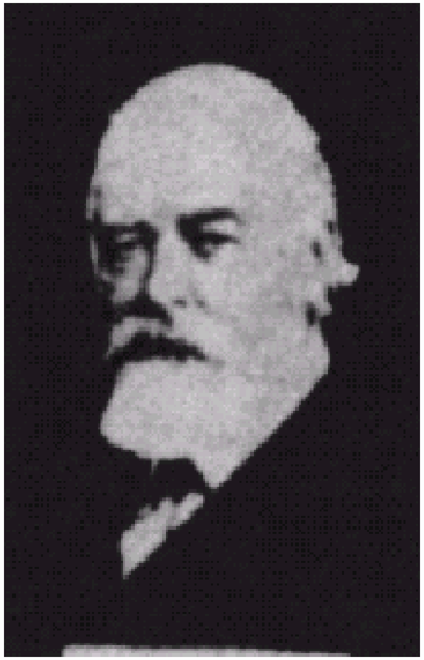
The first photograph in the world

Joseph Nicéphore Niépce, *View from the Window at Le Gras*, 1826.

A Historical Overview of DIP

Newspaper industry used Bartlane cable picture transmission system to send pictures by submarine cable between London and New York in 1920s

Early Improvement

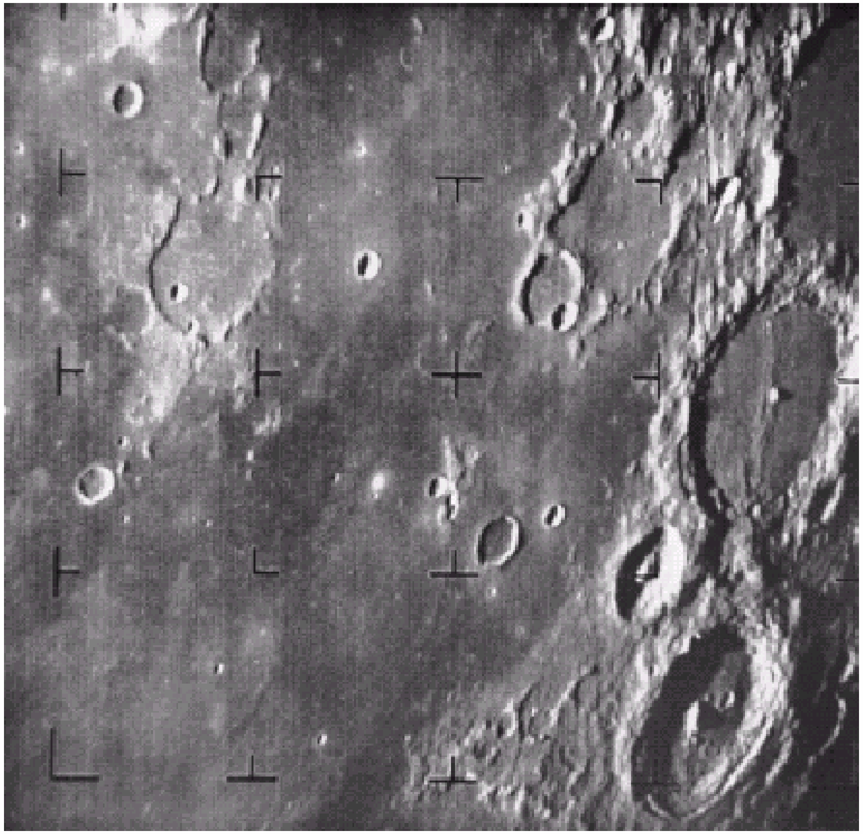


The number of distinct gray levels coded by Bartlane system was improved from 5 to 15 by the end of 1920s

The Born of Digital Image Processing

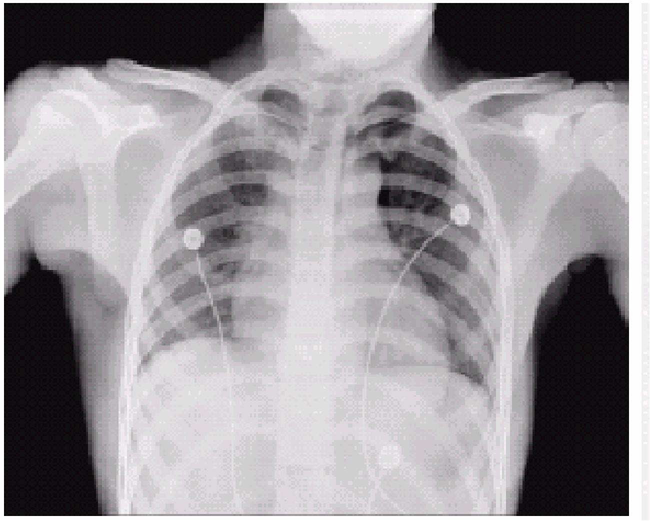
- In 1957 NIST (an agency of the United States Department of Commerce) computer pioneer Russell Kirsch asked, "What would happen if computers could look at pictures?" and helped start a revolution in information technology. Kirsch and his colleagues at NBS, who had developed the nation's first programmable computer, the Standards Eastern Automatic Computer (SEAC), created a rotating drum scanner and programming that allowed images to be fed into it. The first image scanned was a head-and-shoulders shot of Kirsch's three-month-old son Walden.

Soar Into Outer Space



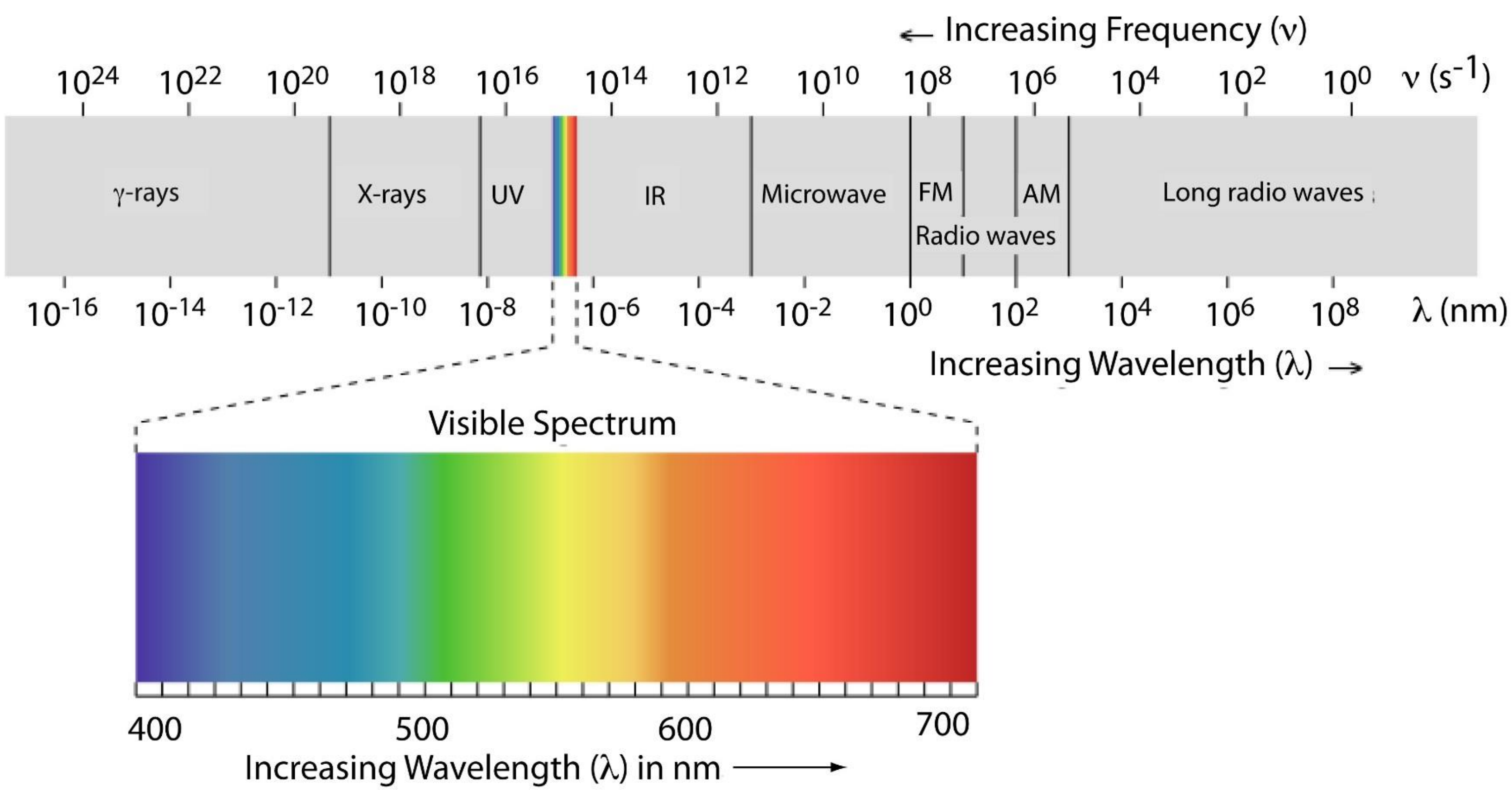
The first picture of moon by US spacecraft *Ranger 7* on July 31, 1964 at 9:09AM EDT

The Born of Computer Tomography



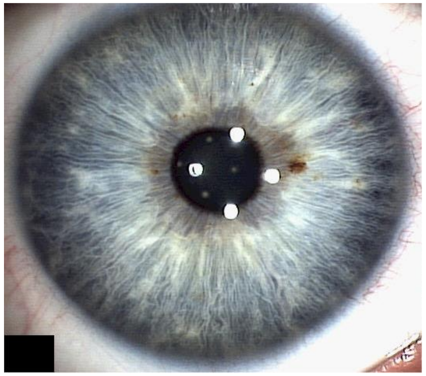
Sir Godfrey N. Housefield and Prof. Allan M. Cormack shared 1979 Nobel Prize in Medicine for the invention of CT

Electromagnetic Spectrum

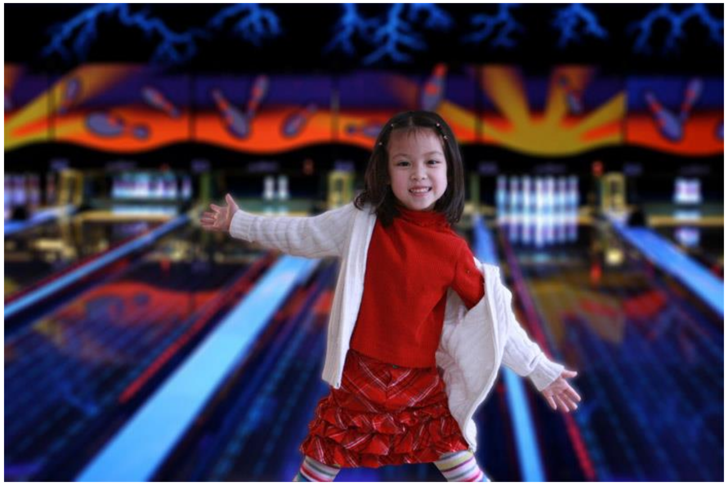


Visible (I): Photography

Visible (II): Biometrics and Forensics

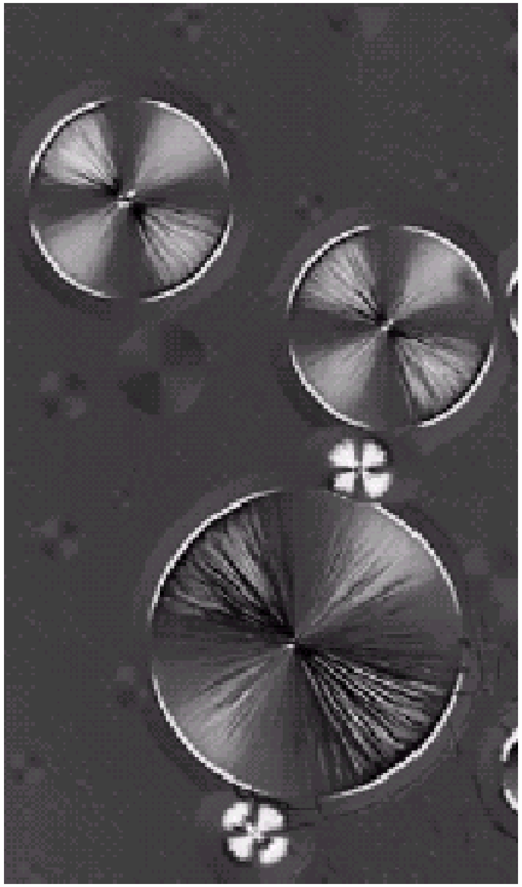


You=ID

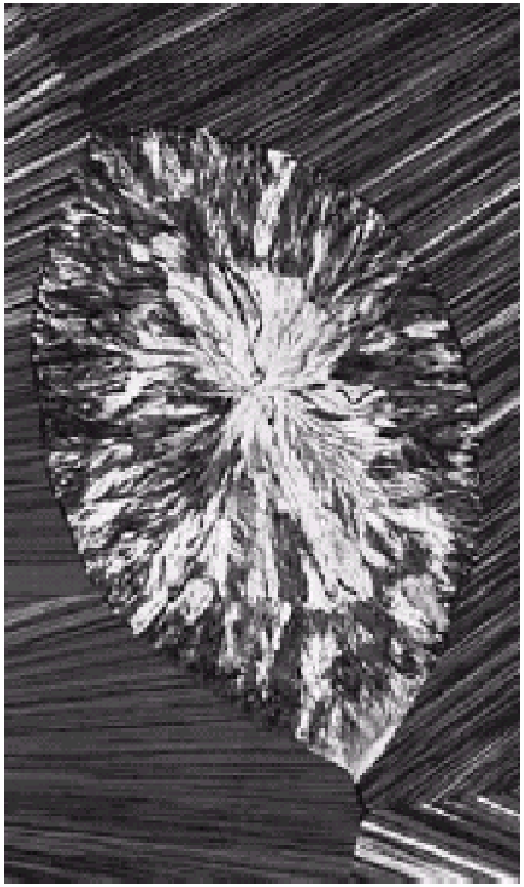


Real or PS?

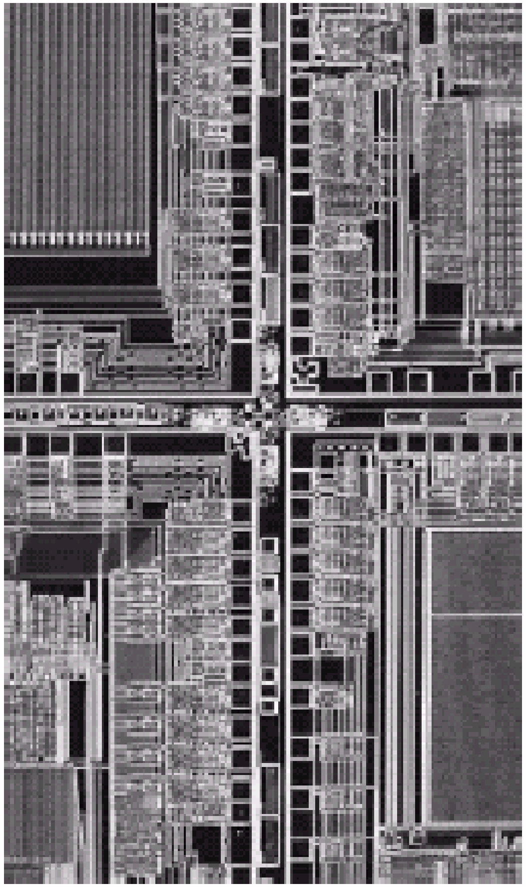
Visible (III): Light Microscopy



Taxol (250×)

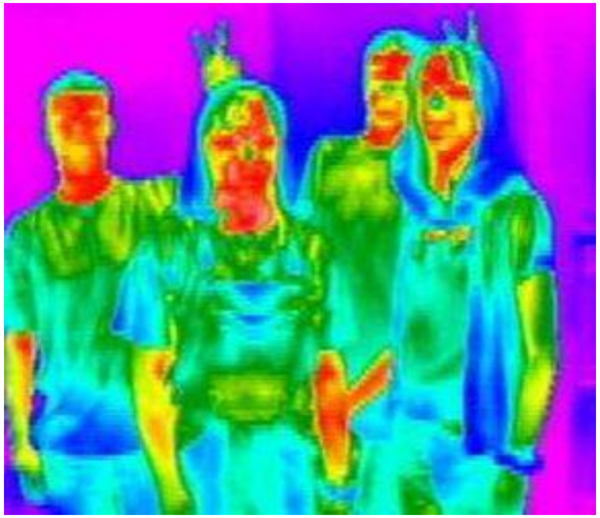


Cholesterol (40×)



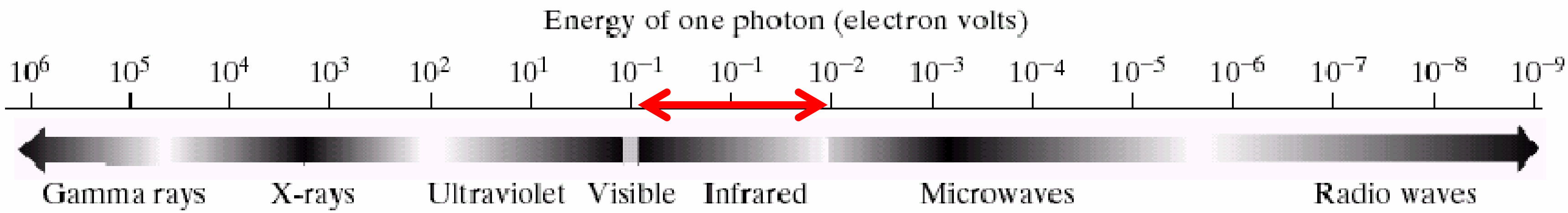
Microprocessor (60×)

Beyond Visible (I): Thermal Images



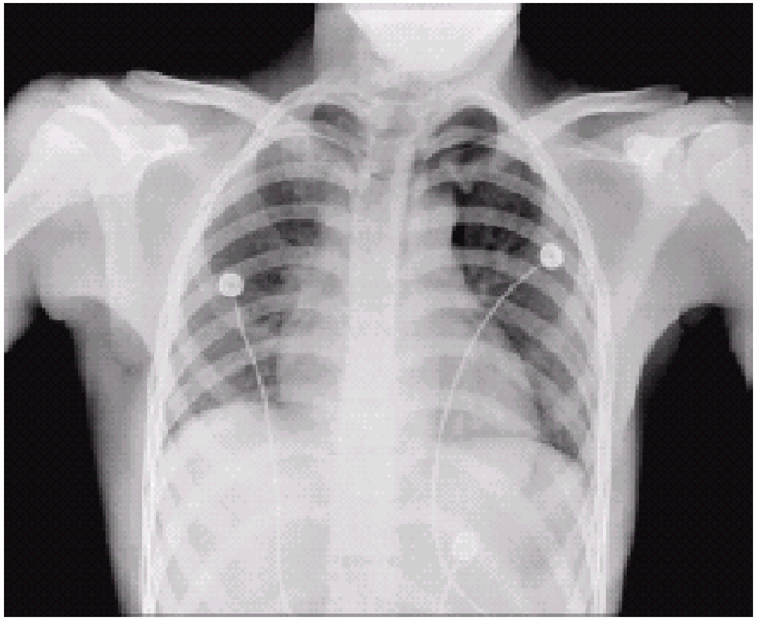
Human body disperses heat (red pixels)

Autoliv's night vision system on the BMW 7 series



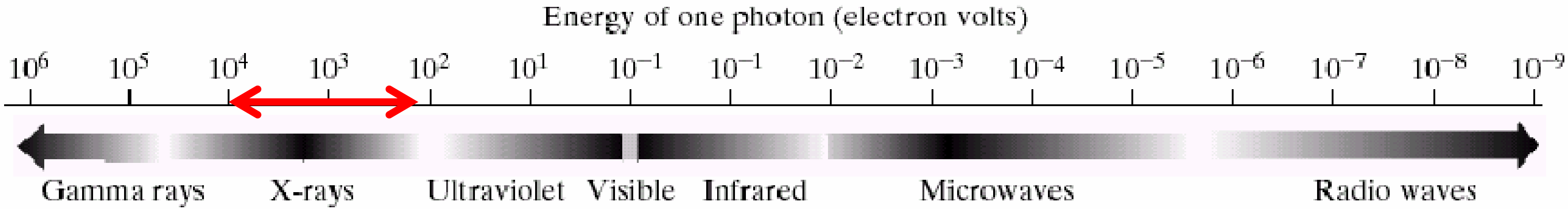
Beyond Visible (II): Medical Diagnostics

Operate in X-ray frequency



chest

head



Other Non-Electro-Magnetic Imaging Modalities

- Acoustic imaging
  - Translate “sound waves” into image signals
  - Ultrasound imaging
- Electron microscopy
  - Shine a beam of electrons through a specimen
  - Transmission electron microscopy (TEM) vs. scanning electron microscopy (SEM)
- Synthetic images in Computer Graphics
  - Computer generated (non-existent in the real world)

بخش اول– دریافت و ادراک تصویر

Sensor Array: CCD (Charge Coupled Device) Imaging

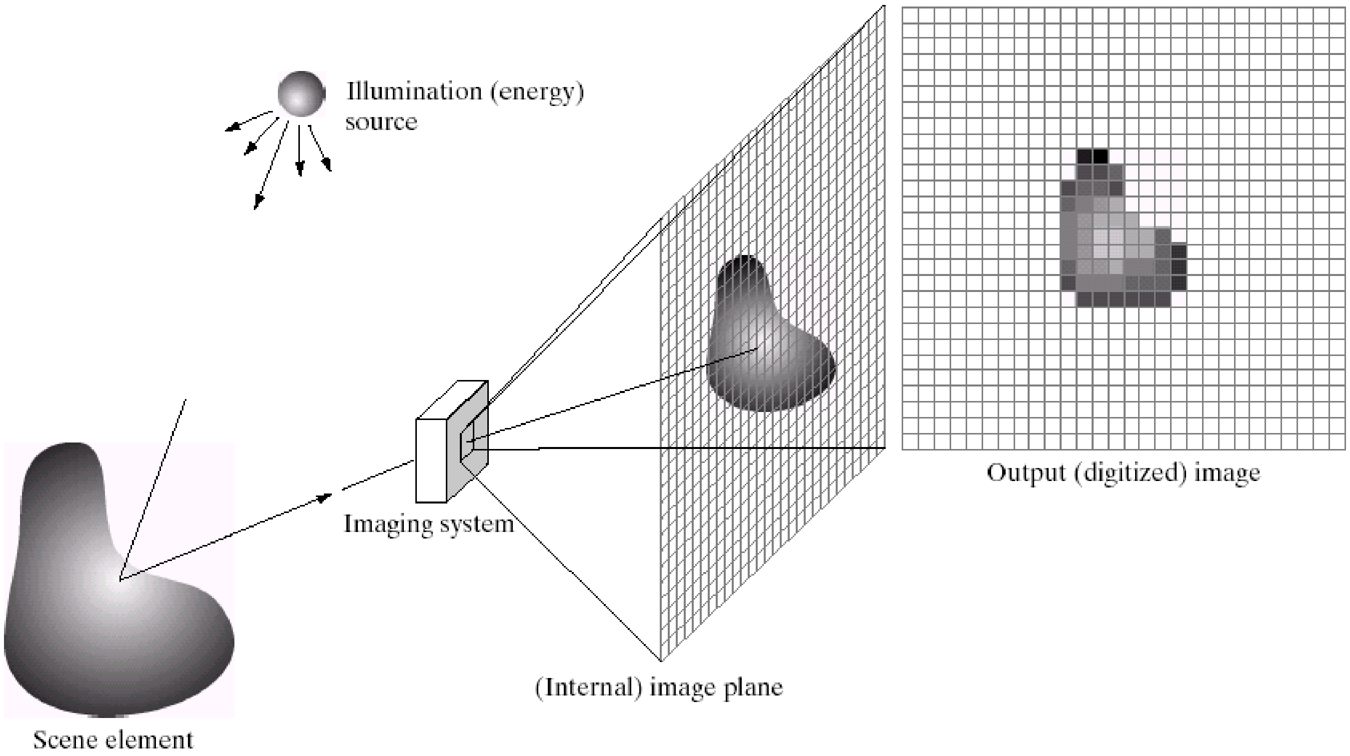


Image Formation Model

$$f(x,y)=i(x,y)r(x,y)+n(x,y)$$

$0<f(x,y)<\infty$	Intensity – proportional to energy radiated by a physical source
$0<i(x,y)<\infty$	illumination
$0<r(x,y)<1$	reflectance (“intrinsic images”)
$n(x,y)$	noise

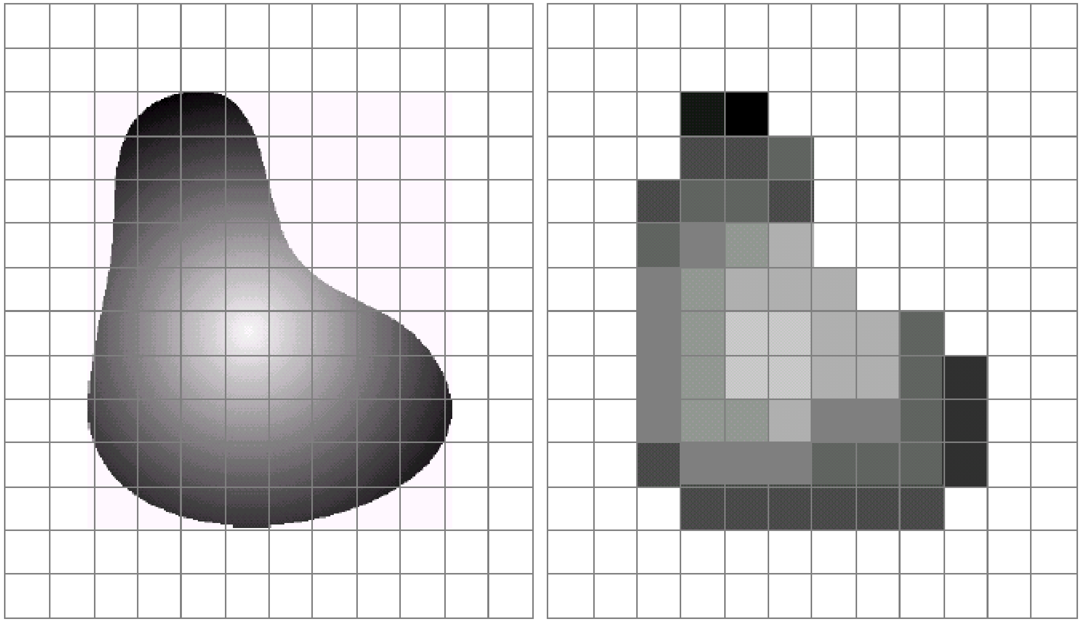
مثال: استفاده از MATLAB برای تولید نرم‌افزار ساده نقاشی

```
% Use:
%   left-click for continous drawing
%   right-click for discontinous drawing
%   left-click or keyboard for finishing

L = 10; % Drawing axes size is LxL
Res = 60; % Image size is (L*Res)x(L*Res)
I = zeros(L*Res,L*Res);
xMAP = repmat((0.5:L*Res-0.5)/Res,L*Res,1);
yMAP = repmat(flipud((0.5:L*Res-0.5)')/Res,1,L*Res);
delta = 0.5/Res;
h = figure;
curr_x = [];
while(true)
    subplot(1,2,1)
    axis equal
    title('Draw your paining');
    xlim([0 10])
    ylim([0 10])
    set(gca,'XTickLabel',{});
    set(gca,'YTickLabel',{});
    try
        [x,y,button] = ginput(1);
    catch
        return;
    end

    if button==3 % right-click
        curr_x = [];
    end
    if ~isempty(curr_x)
        plot([curr_x x],[curr_y y],'-k')
        hold on
        d = abs((curr_y-y)*(xMAP-x)-(curr_x-x)*(yMAP-y))/sqrt((curr_y-y)^2+(curr_x-x)^2);
        I(d<delta & xMAP>=(min(x,curr_x)-delta) & xMAP<=(max(x,curr_x)+delta) & yMAP>=(min(y,curr_y)-delta) & yMAP<=(max(y,curr_y)+delta)) = 1;
        subplot(1,2,2)
        imshow(I)
    else
        plot(x,y,'.k')
        hold on
    end

    curr_x = x;
    curr_y = y;
    if button~=1 && button~=3
        close(h)
        break;
    end
end
```

تمرین ۱

- شکل‌دهی به تصویر
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

2D Sampling and Quantization



a b

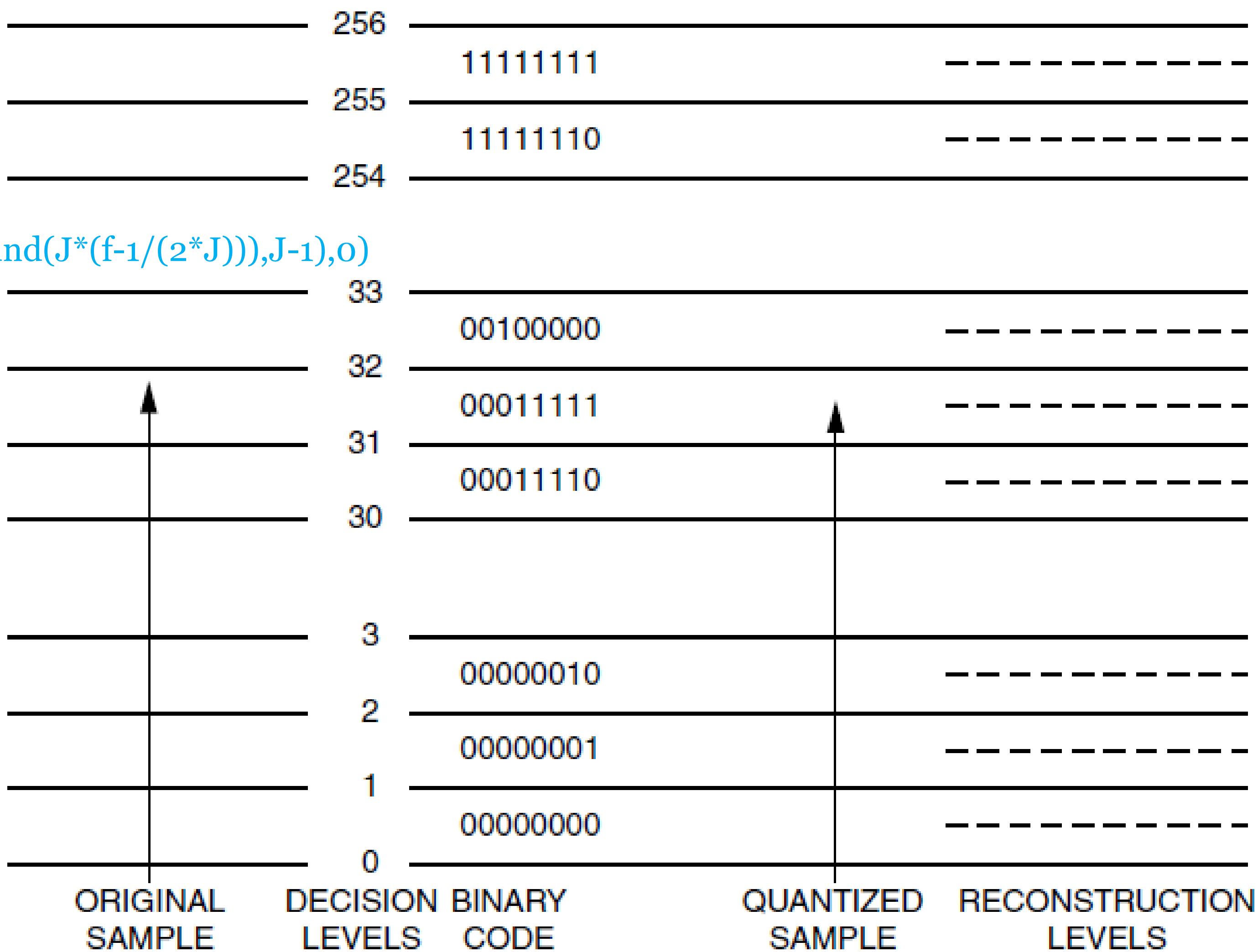
FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Quantization

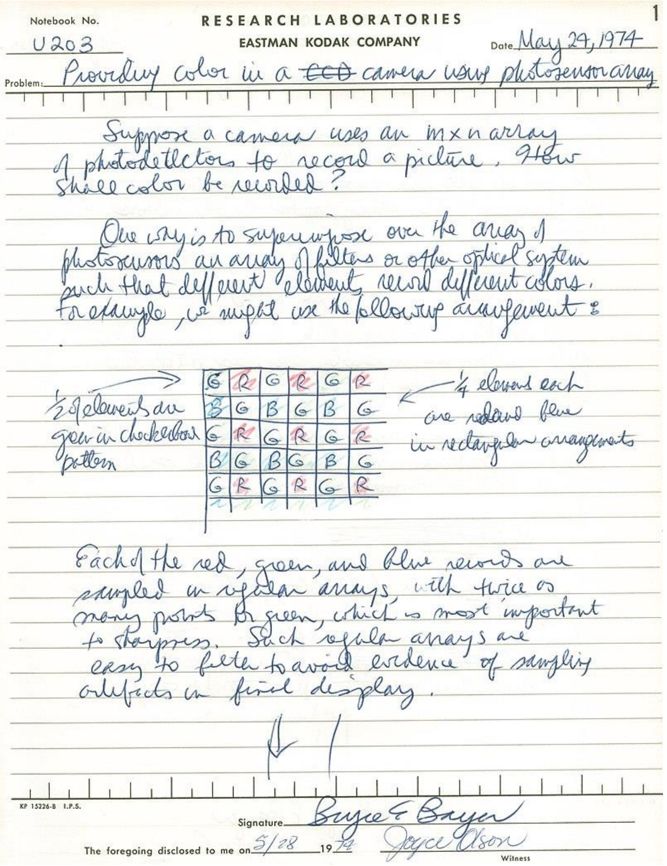
MATLAB

```
f2j = @(f,J) max(min(round(J*(f-1/(2*J))),J-1),0)
j2r = @(j,J) (j+1/2)/J
```

$$a_L = 0$$
$$a_U = 1$$
$$j = [J * (f - 1/2J)]$$
$$r_j = \frac{j + {}^1/_2}{J}$$
$$J = 256$$



Color Imaging: Bayer Pattern



Notebook No. U203

RESEARCH LABORATORIES
EASTMAN KODAK COMPANY

1

Date May 24, 1974

Problem: Providing color in a ~~CCD~~ camera using photosensor array

Suppose a camera uses an m × n array of photodetectors to record a picture. How shall color be recorded?

One way is to superimpose over the array of photosensors an array of filters or other optical system such that different elements record different colors. For example, we might use the following arrangement:

G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R

1/2 of elements are green in checkerboard pattern

1/4 elements each are red and blue in rectangular arrangements

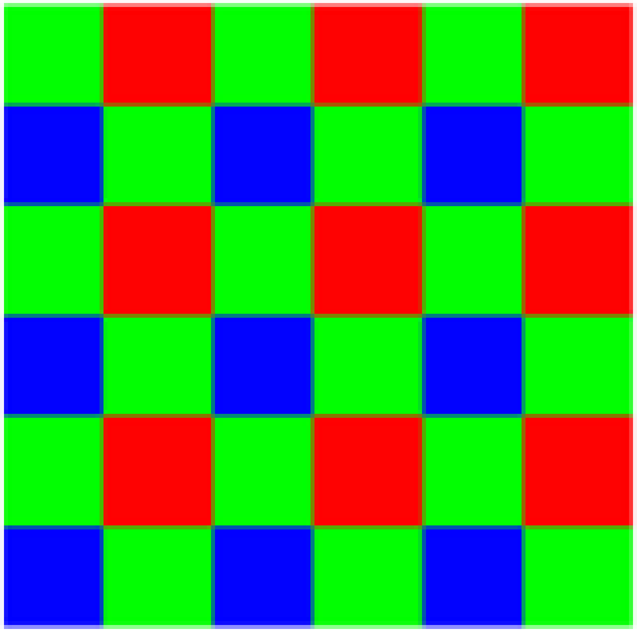
Each of the red, green, and blue records are sampled in regular arrays, with twice as many points for green, which is most important to sharpness. Such regular arrays are easy to filter to avoid evidence of sampling artifacts in final display.

KP 15226-B I.P.S.

Signature Bryce E. Bayer

The foregoing disclosed to me on 5/28 1974 Joyce Olson Witness

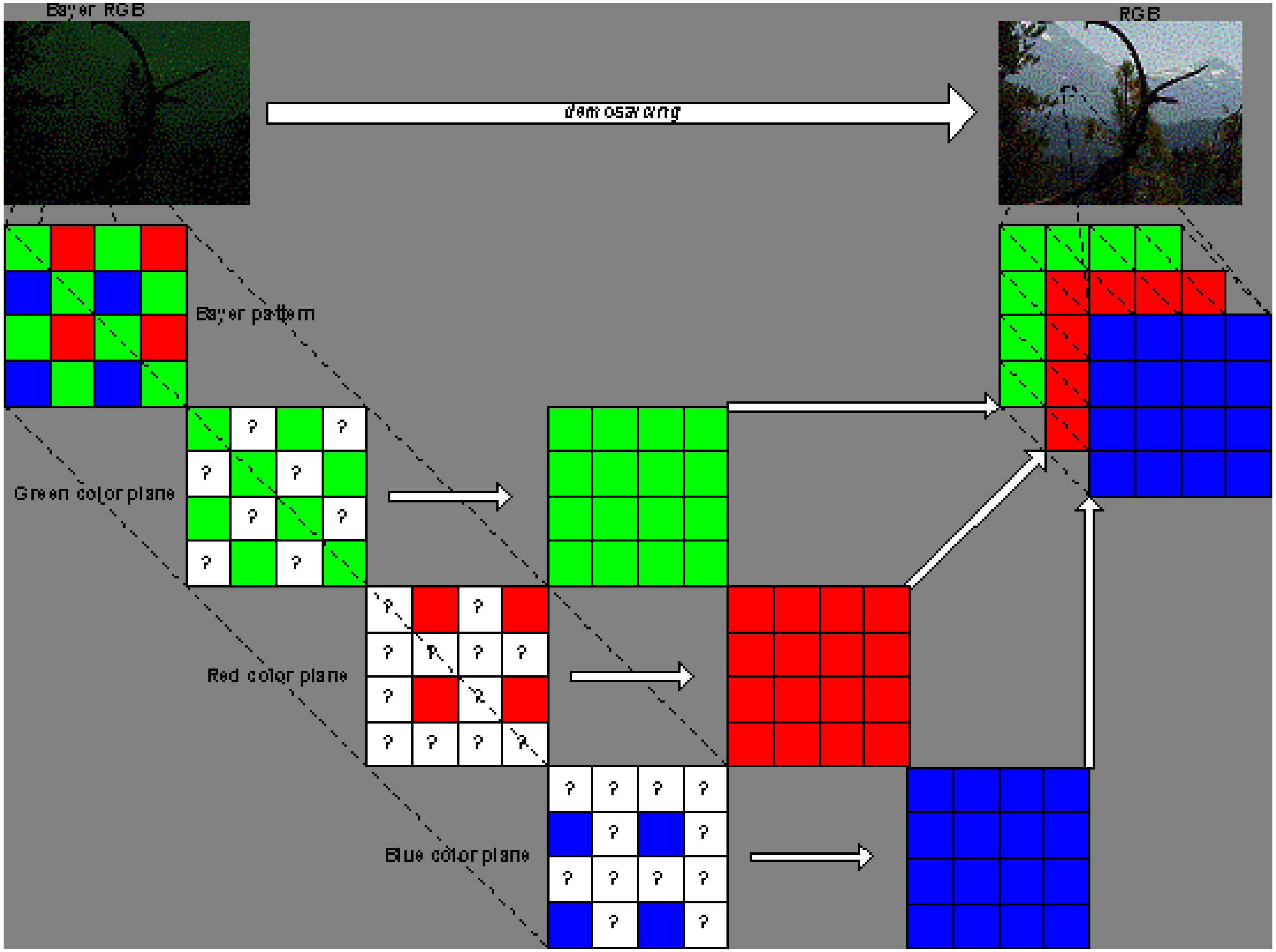
Bayer Pattern



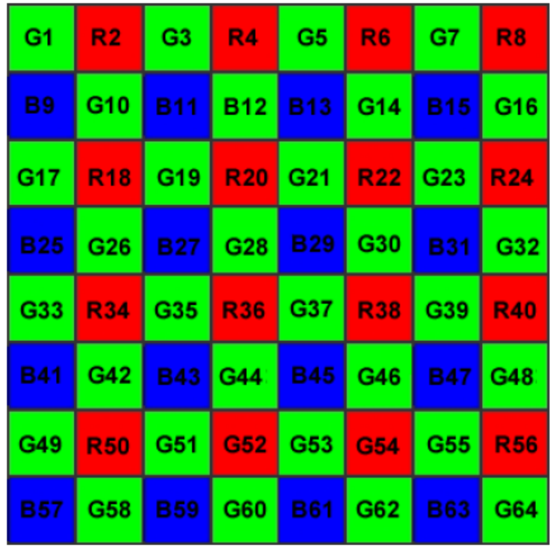
US3,971,065

Demosaicing (CFA Interpolation)

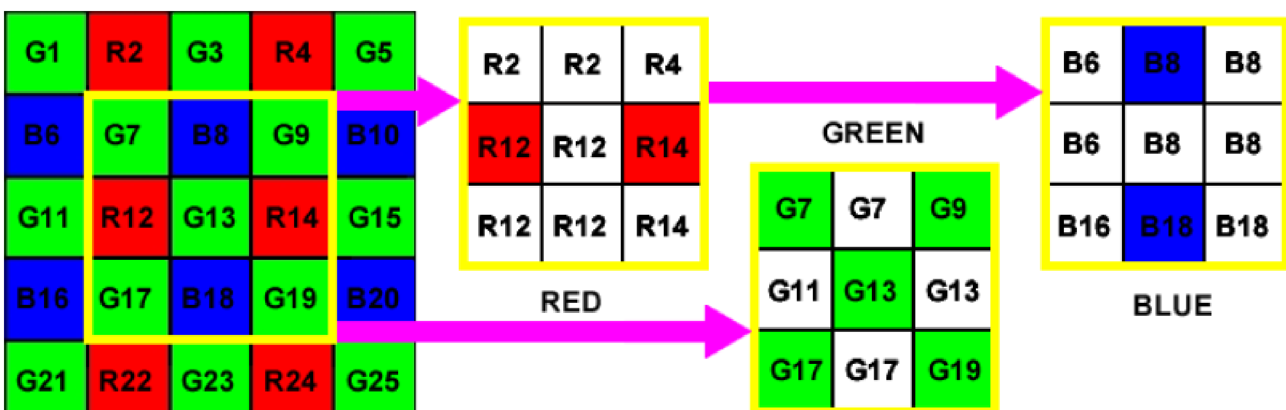
CFA: Color Filter Array



Simple Ideas



Nearest Neighbor



Simple Ideas (cont'd)

Bilinear Interpolation
Averaging the four (or less) neighboring values

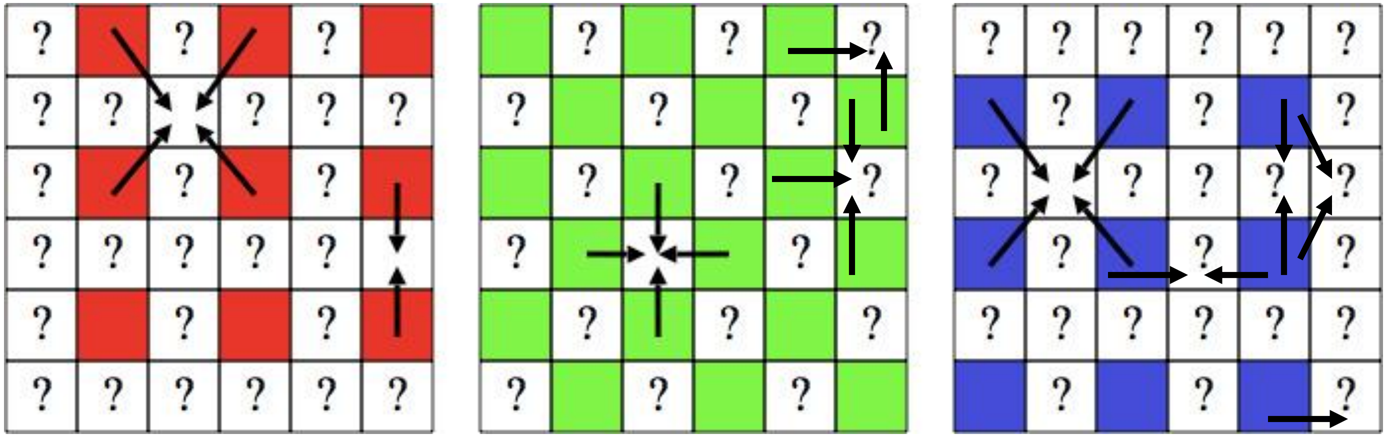
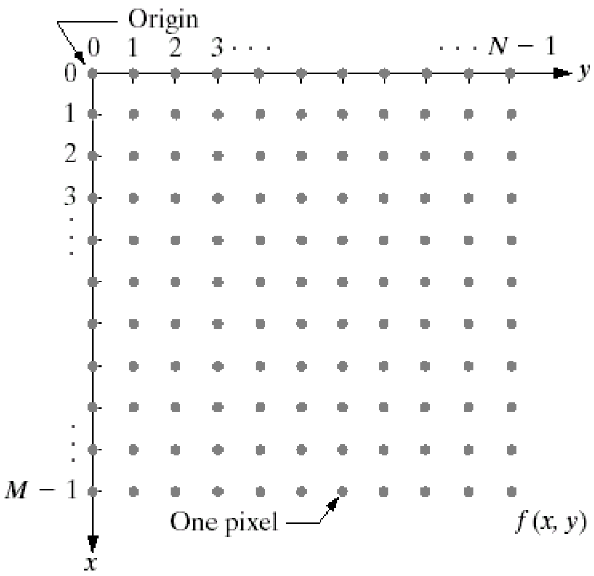


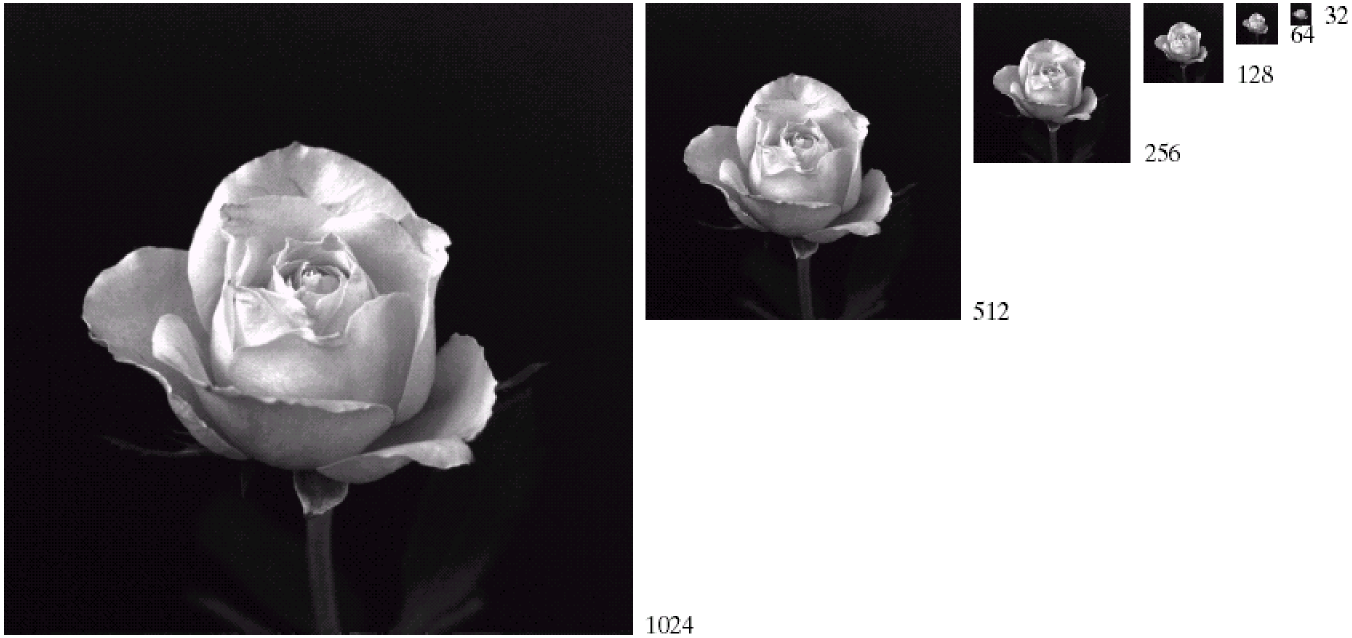
Image Represented by a Matrix



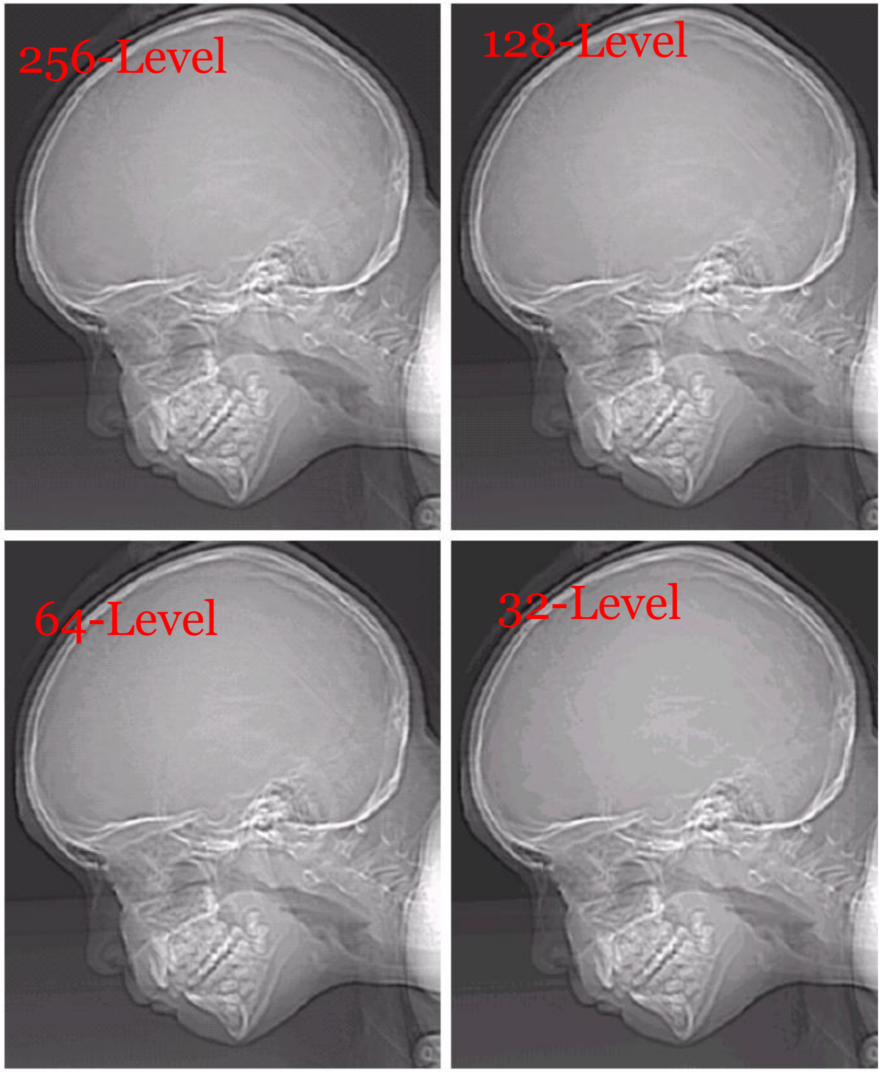
Spatial resolution

Bit-depth resolution

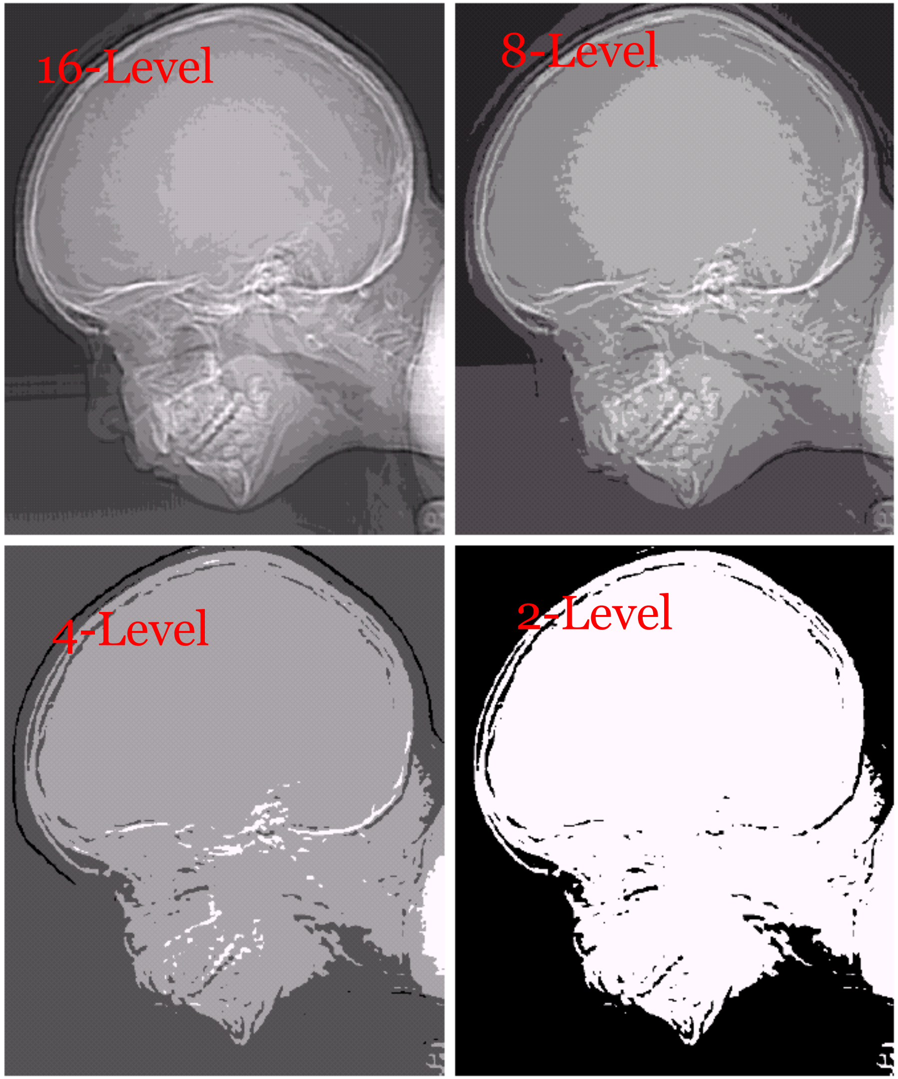
Spatial Resolution



Bit-depth Resolution



Bit-depth Resolution (cont'd)



مثال: اثر کاهش رزولوشن عمق بیت

```
I_8bit = imread('cameraman.tif');
I_8bit = double(I_8bit);
f2j = @(f,J) max(min(round(J*(f-1/(2*J))),J-1),0);
j2r = @(j,J) (j+1/2)/J;

figure
cnt = 0;
for J=[256 128 64 16 4 2]
    I1 = j2r(I_8bit,256);
    I2 = f2j(I1,J);
    cnt = cnt+1;
    subplot(2,3,cnt);
    imshow(I2,[0 J-1])
    title(sprintf('%d-Level',J))
end
```

تمرین ۲

- دریافت و ادراک تصویر
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

آزمون مستمر اول

- اصول اولیه دریافت و ادراک تصویر (۲ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.

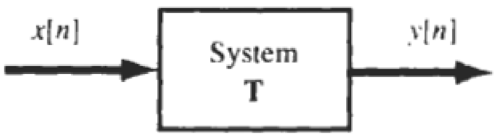
بخش دوم- مروری بر مفاهیم پایه ریاضی

سیستم‌ها و تبدیل حوزه فرکانس

Handouts
- LTI Systems and Convolution Tutorial
- Complex Numbers Tutorial

Systems

- Discrete-Time Systems



- Systems with Memory and without Memory
  - Example of System with Memory

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

- Causal and Noncausal
  - Example of Noncausal Systems
    - Note that all memoryless systems are causal, but not vice versa

$$y[n] = x[-n]$$

Linear Systems and Nonlinear Systems

- Linear Systems
  - Superposition property

$$\mathbf{T}\{\alpha_1 x_1 + \alpha_2 x_2\} = \alpha_1 y_1 + \alpha_2 y_2$$

  - Example

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

- Nonlinear Systems
  - Examples

$$y = x^2$$

$$y = \cos x$$

    - Note that for a linear system, a zero input yields a zero output

Time-Invariant and Time-Varying Systems

- Time-Invariant
  - A time shift in the input signal causes the same time shift in the output signal

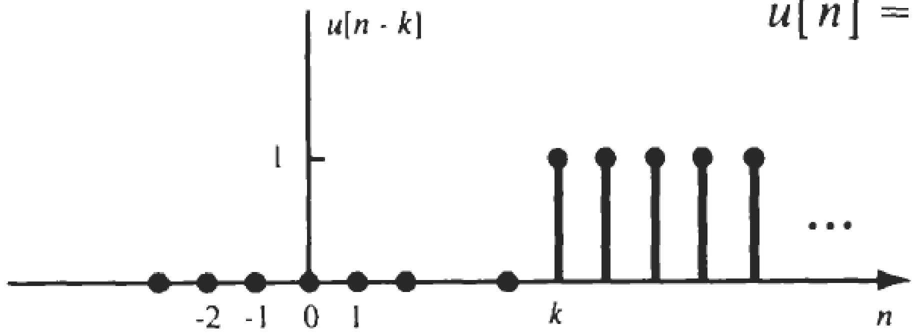
$$\mathbf{T}\{x[n-k]\} = y[n-k]$$

- Stability
  - Bounded-Input/Bounded-Output (BIBO) Stable

$$|x| \leq k_1 \Longrightarrow |y| \leq k_2$$

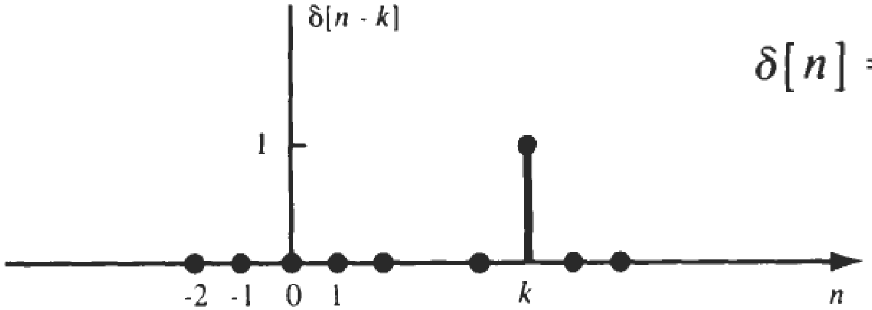
BASIC DISCRETE-TIME SIGNALS

- The Unit Step Sequence



$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

- The Unit Impulse Sequence

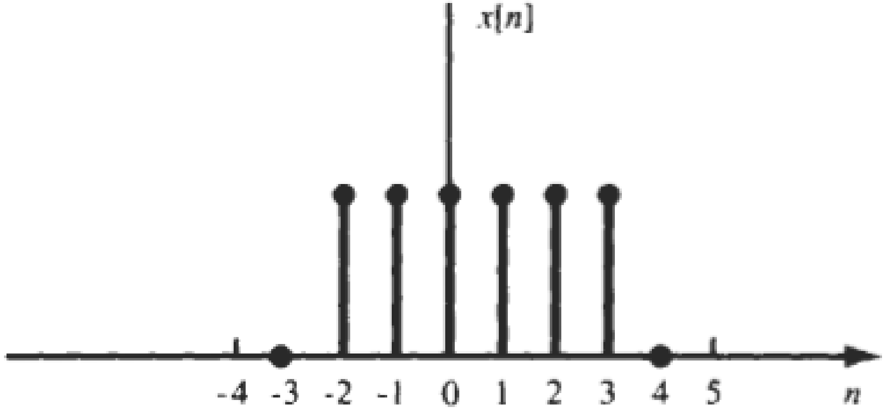


$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

BASIC DISCRETE-TIME SIGNALS

- The Unit Impulse Sequence (cont'd)

$$x[n] = u[n+2] - u[n-4]$$



BASIC DISCRETE-TIME SIGNALS (cont'd)

- Complex Exponential Sequences

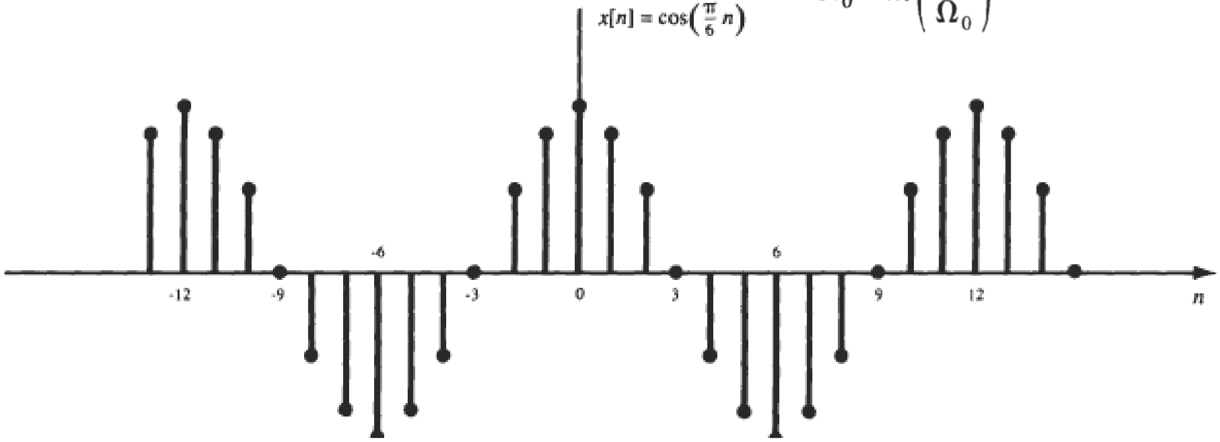
$$x[n] = e^{j\Omega_0 n} = \cos \Omega_0 n + j \sin \Omega_0 n$$

  - Condition for periodicity

$$\frac{\Omega_0}{2\pi} = \frac{m}{N} \qquad m = \text{positive integer}$$

    - Fundamental Period

$$N_0 = m\left(\frac{2\pi}{\Omega_0}\right)$$



BASIC DISCRETE-TIME SIGNALS (cont'd)

- Sinusoidal Sequences

$$A\cos(\Omega_0 n + \theta) = A\,\mathrm{Re}\{e^{j(\Omega_0 n + \theta)}\}$$

تمرین ۳

- مبانی سیگنال و سیستم

Linear Time-Invariant Systems

- Impulse Response

$$h[n] = \mathbf{T}\{\delta[n]\}$$

- Response to an Arbitrary Input

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\,\delta[n-k]$$

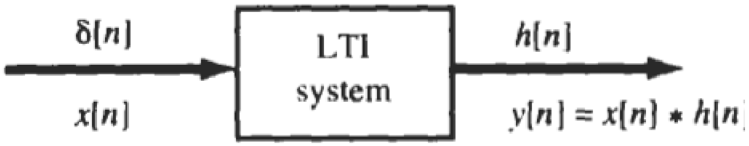
$$y[n] = \mathbf{T}\{x[n]\} = \mathbf{T}\left\{\sum_{k=-\infty}^{\infty} x[k]\,\delta[n-k]\right\}$$

$$= \sum_{k=-\infty}^{\infty} x[k]\mathbf{T}\{\delta[n-k]\} \xrightarrow{h[n-k] = \mathbf{T}\{\delta[n-k]\}} y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Linear Time-Invariant Systems (cont’d)

- Response to an Arbitrary Input (cont’d)
  - Convolution Sum

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$



Linear Time-Invariant Systems (cont'd)

- Response to an Arbitrary Input (cont'd)
  - Convolution Sum (cont'd)
    - Example

Tabular method of linear convolution

n↓	k→	–3	–2	–1	0	1	2	3	4	5	6	7		n	y(n)
	h(k) →				4	3	2	1							
	x(k) →				1	2	3	1							
0	x(0–k)	1	3	2	1									0	4
1	x(1–k)		1	3	2	1								1	11
2	x(2–k)			1	3	2	1							2	20
3	x(3–k)				1	3	2	1						3	18
4	x(4–k)					1	3	2	1					4	11
5	x(5–k)						1	3	2	1				5	5
6	x(6–k)							1	3	2	1			6	1
7	x(7–k)								1	3	2	1		7	0
.	.													.	.

MATLAB Code:
conv([1 2 3 1],[4 3 2 1])

Linear Time-Invariant Systems (cont'd)

- Properties of Convolution Sum

1. Commutative:

$$x[n] * h[n] = h[n] * x[n]$$

2. Associative:

$$\{x[n] * h_1[n]\} * h_2[n] = x[n] * \{h_1[n] * h_2[n]\}$$

3. Distributive:

$$x[n] * \{h_1[n]\} + h_2[n]\} = x[n] * h_1[n] + x[n] * h_2[n]$$

Linear Time-Invariant Systems (cont'd)

- Properties of Convolution Sum (cont'd)

$$x[n] * \delta[n] = x[n]$$

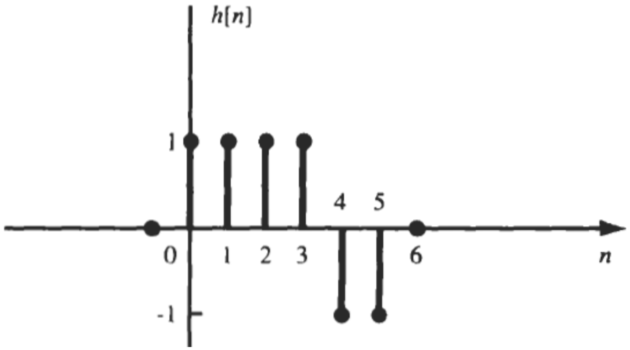
$$x[n] * \delta[n - n_0] = x[n - n_0]$$

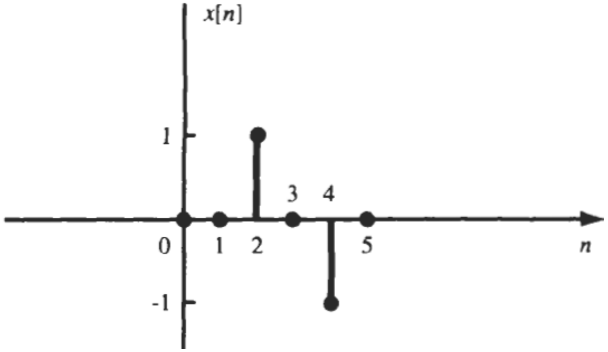
$$x[n] * u[n] = \sum_{k=-\infty}^{n} x[k]$$

$$x[n] * u[n - n_0] = \sum_{k=-\infty}^{n-n_0} x[k]$$

Linear Time-Invariant Systems (cont'd)

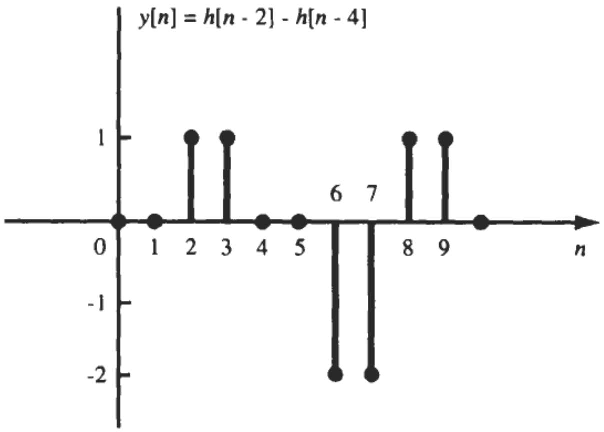
- Properties of Convolution Sum (cont'd)
  - Example





$$x[n] = \delta[n-2] - \delta[n-4]$$

$$\Rightarrow \quad y[n] = h[n-2] - h[n-4]$$



Linear Time-Invariant Systems (cont'd)

- PROPERTIES OF DISCRETE-TIME LTI SYSTEMS
  - Systems without Memory

$$h[n] = K\delta[n]$$

  - Causality

$$x[n] = 0 \qquad n < 0$$

  - Stability

$$\sum_{k=-\infty}^{\infty} |h[k]| < \infty$$

تمرین ۴

- کانولوشن

Fourier Transform

- Fourier Transform Pair

$$X(\Omega) = \mathscr{F}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\Omega n}$$

$$x[n] = \mathscr{F}^{-1}\{X(\Omega)\} = \frac{1}{2\pi}\int_{2\pi} X(\Omega)\, e^{j\Omega n}\, d\Omega$$

$$x[n] \leftrightarrow X(\Omega)$$

- Fourier Spectra
  - magnitude spectrum
  - phase spectrum

$$X(\Omega) = |X(\Omega)| e^{j\phi(\Omega)}$$

- Convergence of Fourier Transform

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty$$

Fourier Transform (cont'd)

- PROPERTIES OF THE FOURIER TRANSFOR
  - Periodicity

$$X(\Omega + 2\pi) = X(\Omega)$$

  - Linearity

$$a_1 x_1[n] + a_2 x_2[n] \leftrightarrow a_1 X_1(\Omega) + a_2 X_2(\Omega)$$

  - Time Shifting

$$x[n - n_0] \leftrightarrow e^{-j\Omega n_0} X(\Omega)$$

  - Frequency Shifting

$$e^{j\Omega_0 n} x[n] \leftrightarrow X(\Omega - \Omega_0)$$

Fourier Transform (cont'd)

- PROPERTIES OF THE FOURIER TRANSFOR (cont'd)
  - Conjugation

$$x^*[n] \leftrightarrow X^*(-\Omega)$$

  - Time Reversal

$$x[-n] \leftrightarrow X(-\Omega)$$

  - Time Scaling

$$x(at) \leftrightarrow \frac{1}{|a|} X\left(\frac{\omega}{a}\right)$$

Fourier Transform (cont’d)

- PROPERTIES OF THE FOURIER TRANSFOR (cont’d)
  - Differentiation in Frequency

$$nx[n] \leftrightarrow j\frac{dX(\Omega)}{d\Omega}$$

  - Differencing

$$x[n] - x[n-1] \leftrightarrow (1 - e^{-j\Omega})X(\Omega)$$

  - Accumulation

$$\sum_{k=-\infty}^{n} x[k] \leftrightarrow \pi X(0)\,\delta(\Omega) + \frac{1}{1 - e^{-j\Omega}}X(\Omega) \qquad |\Omega| \le \pi$$

Fourier Transform (cont'd)

- Example 1: Fourier Transform of Rectangular Pulse Sequence

$$x[n] = u[n] - u[n-N] \Rightarrow X(\Omega) = X(e^{j\Omega}) = \frac{1-e^{-j\Omega N}}{1-e^{-j\Omega}} = \frac{e^{-j\Omega N/2}(e^{j\Omega N/2} - e^{j\Omega N/2})}{e^{-j\Omega/2}(e^{j\Omega/2} - e^{-j\Omega/2})}$$

$$= e^{-j\Omega(N-1)/2}\frac{\sin(\Omega N/2)}{\sin(\Omega/2)}$$

- Example 2: Inverse Fourier Transform

$$X(\Omega) = \frac{1}{(1-ae^{-j\Omega})^2} \qquad |a| < 1 \Rightarrow a^n u[n] \leftrightarrow \frac{1}{1-ae^{-j\Omega}} \qquad |a| < 1$$

$$X(\Omega) = \frac{1}{(1-ae^{-j\Omega})^2} = \left(\frac{1}{1-ae^{-j\Omega}}\right)\left(\frac{1}{1-ae^{-j\Omega}}\right)$$

$$\Rightarrow x[n] = a^n u[n] * a^n u[n] = \sum_{k=-\infty}^{\infty} a^k u[k] a^{n-k} u[n-k]$$

$$= a^n \sum_{k=0}^{n} 1 = (n+1)a^n u[n]$$

Table of Fourier Transform Pairs

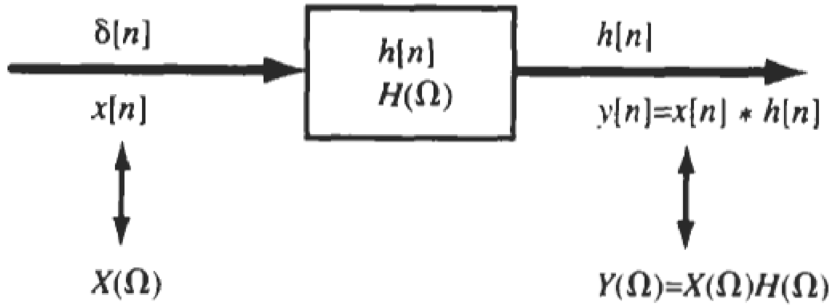
$x[n]$	$X(\Omega)$
$\delta[n]$	1
$\delta[n-n_0]$	$e^{-j\Omega n_0}$
$x[n]=1$	$2\pi\delta(\Omega), \lvert\Omega\rvert \le \pi$
$e^{j\Omega_0 n}$	$2\pi\delta(\Omega-\Omega_0), \lvert\Omega\rvert, \lvert\Omega_0\rvert \le \pi$
$\cos\Omega_0 n$	$\pi[\delta(\Omega-\Omega_0)+\delta(\Omega+\Omega_0)], \lvert\Omega\rvert, \lvert\Omega_0\rvert \le \pi$
$\sin\Omega_0 n$	$-j\pi[\delta(\Omega-\Omega_0)-\delta(\Omega+\Omega_0)], \lvert\Omega\rvert, \lvert\Omega_0\rvert \le \pi$
$u[n]$	$\pi\delta(\Omega)+\dfrac{1}{1-e^{-j\Omega}}, \lvert\Omega\rvert \le \pi$
$-u[-n-1]$	$-\pi\delta(\Omega)+\dfrac{1}{1-e^{-j\Omega}}, \lvert\Omega\rvert \le \pi$
$a^n u[n], \lvert a\rvert < 1$	$\dfrac{1}{1-ae^{-j\Omega}}$
$-a^n u[-n-1], \lvert a\rvert > 1$	$\dfrac{1}{1-ae^{-j\Omega}}$
$(n+1)a^n u[n], \lvert a\rvert < 1$	$\dfrac{1}{\left(1-ae^{-j\Omega}\right)^2}$
$a^{\lvert n\rvert}, \lvert a\rvert < 1$	$\dfrac{1-a^2}{1-2a\cos\Omega+a^2}$
$x[n]=\begin{cases} 1 & \lvert n\rvert \le N_1 \\ 0 & \lvert n\rvert > N_1 \end{cases}$	$\dfrac{\sin\left[\Omega\left(N_1+\frac{1}{2}\right)\right]}{\sin(\Omega/2)}$
$\dfrac{\sin Wn}{\pi n}, 0 < W < \pi$	$X(\Omega)=\begin{cases} 1 & 0 \le \lvert\Omega\rvert \le W \\ 0 & W < \lvert\Omega\rvert \le \pi \end{cases}$
$\sum_{k=-\infty}^{\infty}\delta[n-kN_0]$	$\Omega_0\sum_{k=-\infty}^{\infty}\delta(\Omega-k\Omega_0), \Omega_0=\dfrac{2\pi}{N_0}$

Convolution Property of Fourier Transform

- Convolution Property
  - Frequency Response

$$x_1[n] * x_2[n] \leftrightarrow X_1(\Omega)X_2(\Omega)$$

$$Y(\Omega) = X(\Omega)H(\Omega)$$



Convolution Property of Fourier Transform (cont'd)

- Example 1: A causal discrete-time LTI system

$$y[n] - \tfrac{3}{4}y[n-1] + \tfrac{1}{8}y[n-2] = x[n]$$

$$\Rightarrow \quad Y(\Omega) - \tfrac{3}{4}e^{-j\Omega}Y(\Omega) + \tfrac{1}{8}e^{-j2\Omega}Y(\Omega) = X(\Omega)$$

$$\Rightarrow \quad H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = \frac{1}{1 - \frac{3}{4}e^{-j\Omega} + \frac{1}{8}e^{-j2\Omega}} = \frac{1}{\left(1 - \frac{1}{2}e^{-j\Omega}\right)\left(1 - \frac{1}{4}e^{-j\Omega}\right)}$$

Partial-Fraction Expansion

$$\Rightarrow \quad H(\Omega) = \frac{1}{\left(1 - \frac{1}{2}e^{-j\Omega}\right)\left(1 - \frac{1}{4}e^{-j\Omega}\right)} = \frac{2}{1 - \frac{1}{2}e^{-j\Omega}} - \frac{1}{1 - \frac{1}{4}e^{-j\Omega}}$$

$$\Rightarrow \quad h[n] = \left[2\left(\tfrac{1}{2}\right)^n - \left(\tfrac{1}{4}\right)^n\right]u[n]$$

Convolution Property of Fourier Transform (cont'd)

- Example 2: A causal discrete-time FIR filter

$$h[n] = \{2, 2, -2, -2\}$$

$$\Rightarrow \quad H(\Omega) = \sum_{n=-\infty}^{\infty} h[n]\, e^{-j\Omega n} = 2 + 2e^{-j\Omega} - 2e^{-j2\Omega} - 2e^{-j3\Omega}$$

$$= 2\left(1 - e^{-j3\Omega}\right) + 2\left(e^{-j\Omega} - e^{-j2\Omega}\right)$$

$$= 2e^{-j3\Omega/2}\left(e^{j3\Omega/2} - e^{-j3\Omega/2}\right) + 2e^{-j3\Omega/2}\left(e^{j\Omega/2} - e^{-j\Omega/2}\right)$$

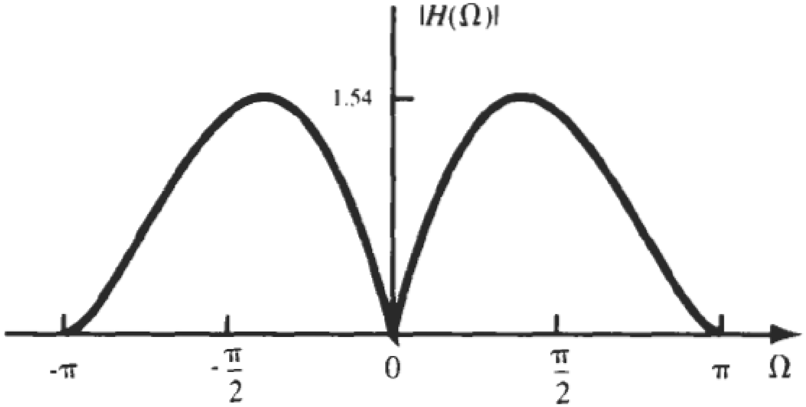
$$= j\, e^{-j3\Omega/2}\left(\sin\frac{\Omega}{2} + \sin\frac{3\Omega}{2}\right) = H_r(\Omega)\, e^{j[(\pi/2)-(3\Omega/2)]}$$

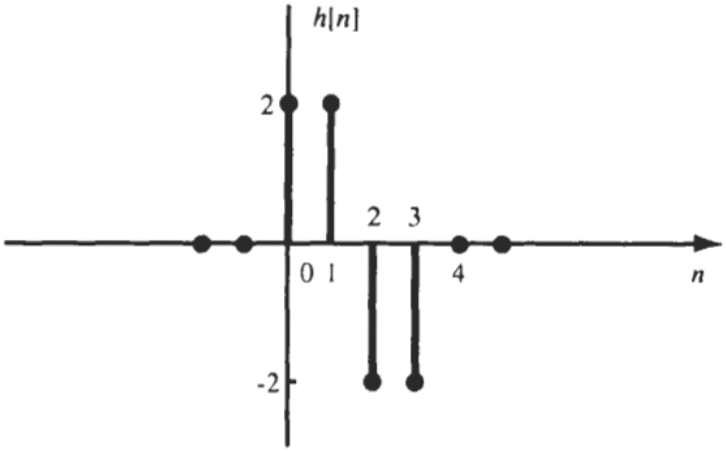
$$H_r(\Omega) = \sin\left(\frac{\Omega}{2}\right) + \sin\left(\frac{3\Omega}{2}\right)$$

$$\Rightarrow \quad |H(\Omega)| = |H_r(\Omega)| = \left|\sin\left(\frac{\Omega}{2}\right) + \sin\left(\frac{3\Omega}{2}\right)\right|$$

Convolution Property of Fourier Transform (cont'd)

- Example 2: A causal discrete-time FIR filter (cont'd)

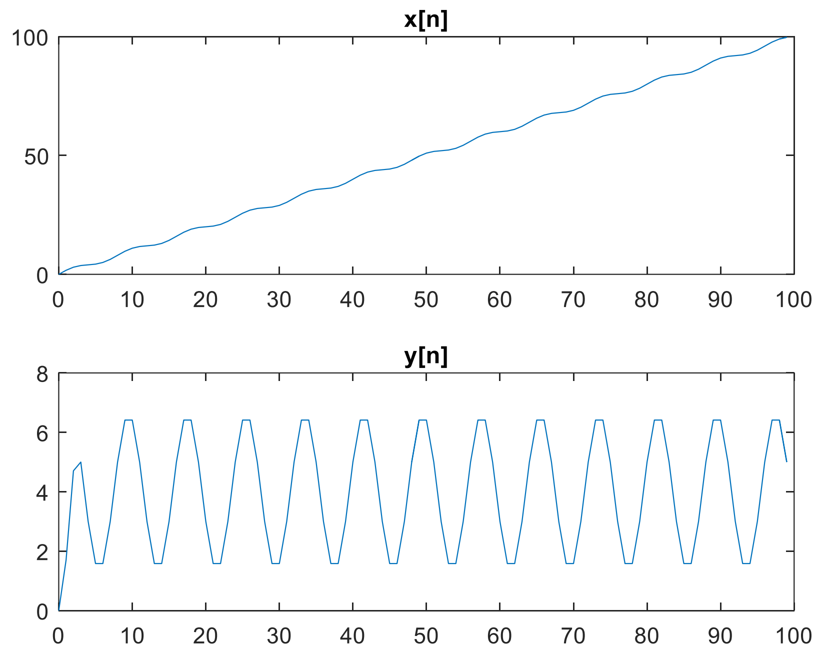




Convolution Property of Fourier Transform (cont’d)

- Example 2: A causal discrete-time FIR filter (cont’d)

```
N = 100;
n = 0:N-1;
x = n+sin(2*pi/8*n);

subplot(2,1,1)
plot(n,x)
title('x[n]')

h = [1 1 -1 -1];
y = conv(x,h);
subplot(2,1,2)
plot(n,y(1:N))
title('y[n]')
```



Discrete Fourier Transform (DFT)

- DFT for Finite-Length Sequences

$$x[n] = 0 \qquad \text{outside the range } 0 \le n \le N-1$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad k = 0, 1, \ldots, N-1$$

$$W_N = e^{-j(2\pi/N)}$$

$$X(\Omega) = \sum_{n=0}^{N-1} x[n]\, e^{-j\Omega n}$$

$$X[k] = X(\Omega)|_{\Omega = k2\pi/N} = X\left(\frac{k2\pi}{N}\right)$$

- Inverse DFT (IDFT)

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] W_N^{-kn} \qquad n = 0, 1, \ldots, N-1$$

- FFT
  - Fast Fourier Transform
    - Used in 4G and 5G Mobiles ☺

Discrete Fourier Transform (cont'd)

- Example
  - x={1,2,3,4}
  - $X_0 = \sum_{n=0}^{3} x_n e^{-i2\pi 0n/4} = \sum_{n=0}^{3} x_n$ =10
  - $X_1 = \sum_{n=0}^{3} x_n e^{-i2\pi 1n/4} = x_0 e^{-i\pi/2} + x_1 e^{-i\pi} + x_2 e^{-i3\pi/2} + x_3 e^{-i2\pi}$

    $= x_0 i - x_1 + x_2 i - x_3$ $= 1i - 2 + 3i - 4 = -2 + 2i$
  - $X_2 = \sum_{n=0}^{3} x_n e^{-i2\pi 2n/4} = \sum_{n=0}^{3} x_n e^{-i\pi n} = \sum_{n=0}^{3} (-1)^n x_n = 2 - 2i$
  - $X_3 = \sum_{n=0}^{3} x_n e^{-i2\pi 3n/4} = x_0 e^{-i3\pi/4} + x_1 e^{-i3\pi/2} + x_2 e^{-i9\pi/4} + x_3 e^{-i3\pi}$

    $= x_0(-1 - i) + x_1(-1 + i) + x_2(1 + i) + x_3(1 - i)$

    $= (1 - 1i) + (2 + 2i) - (3 - 3i) - (4 + 4i) = -4i$

$\Rightarrow$ $X = [10, -2 + 2i, 2 - 2i, -4i]$

Discrete Fourier Transform (cont'd)

- Circular Convolution

$$x_1[n] \otimes x_2[n] \leftrightarrow X_1[k]X_2[k]$$

$$x_1[n] \otimes x_2[n] = \sum_{i=0}^{N-1} x_1[i]x_2[n-i]_{\mathrm{mod}\ N}$$

- Circular vs Linear Convolution
  - To make the circular convolution of x and y (with lengths N and L, respectively) equivalent to the linear convolution, you must pad the vectors with zeros to length at least N + L - 1 before taking the DFT
  - After inverting the product of the DFTs, retain only the first N + L - 1 elements

تمرین ۵

- تبدیل فوریه و سری فوریه

آزمون مستمر دوم

- مروری بر مفاهیم پایه ریاضی (۲ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.

بخش سوم - مقابله با نویز

What is Noise?

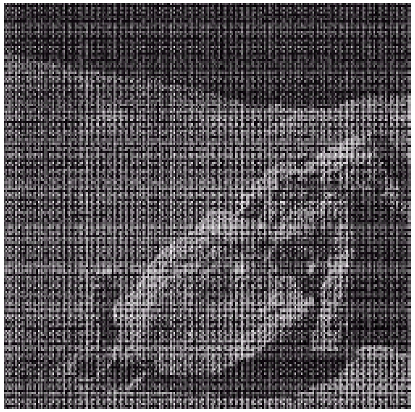
- Wiki definition: **noise** means any unwanted signal
- One person's signal is another one's noise
- Noise is not always random

Image Denoising

- Noise
  - Any unwanted signal
  - One person's signal is another one's noise
  - Noise is not always random
- Where does noise come from?
  - Sensor (e.g., thermal or electrical interference)
  - Environmental conditions (rain, snow etc.)
- Why do we want to denoise?
  - Visually unpleasant
  - Bad for analysis

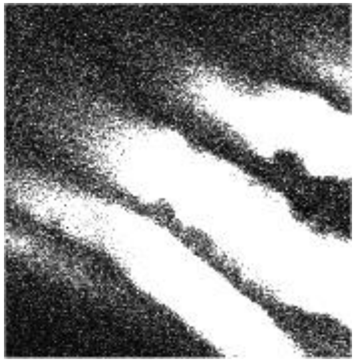
Noisy Image Examples

thermal imaging



electrical interference

ultrasound imaging

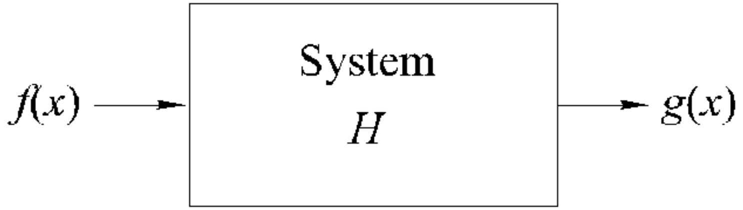


physical interference

Noise Removal Techniques

- Linear filtering
- Nonlinear filtering

Recall



Linear system

$$
\begin{aligned}
H[a_i f_i(x) + a_j f_j(x)] &= a_i H[f_i(x)] + a_j H[f_{ji}(x)] \\
&= a_i g_i(x) + a_j g_j(x)
\end{aligned}
$$

Impulse Noise (salt-pepper Noise)

- <u>Definition</u>

Each pixel in an image has the probability of p/2 (0<p<1) being contaminated by either a white dot (salt) or a black dot (pepper)

$$Y(i,j)=\begin{cases} 255 & \text{with probability of p/2} \\ 0 & \text{with probability of p/2} \\ X(i,j) & \text{with probability of 1-p} \end{cases}$$

noisy pixels

clean pixels

$$1 \le i \le H, 1 \le j \le W$$

X: noise-free image, Y: noisy image

<u>Note</u>: in some applications, noisy pixels are not simply black or white, which makes the impulse noise removal problem more difficult

Numerical Example

P=0.1

128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128

X

128	128	255	0	128	128	128	128	128	128
128	128	128	128	0	128	128	128	128	0
128	128	128	128	128	128	128	128	128	128
128	128	0	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
0	128	128	128	128	255	128	128	128	128
128	128	128	128	128	128	128	128	128	255
128	128	128	128	128	128	128	255	128	128

Y

Noise level p=0.1 means that approximately 10% of pixels are contaminated by salt or pepper noise (highlighted by red color)

>Y = imnoise(X,'salt & pepper',p)

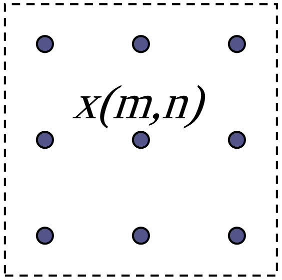
Impulse Noise Removal Problem

Noisy image Y

→ filtering algorithm → denoised image $\hat{X}$

Can we make the denoised image $\hat{X}$ as close to the noise-free image X as possible?

2D Median Filtering



W: (2T+1)-by-(2T+1) window

$$\hat{x}(m,n) = median[y(m-T,n-T),\ldots,y(m-T,n+T),\ldots,$$
$$y(m,n),\ldots,y(m+T,n-T),\ldots,y(m+T,n+T)]$$

MATLAB command: x=medfilt2(y,[2*T+1,2*T+1]);

Numerical Example

225	225	225	226	226	226	226	226
225	225	255	226	226	226	225	226
226	226	225	226	**0**	226	226	255
255	226	225	**0**	226	226	226	226
225	255	**0**	225	226	226	226	255
255	225	224	226	226	0	225	226
226	225	225	226	255	226	226	228
226	226	225	226	226	226	226	226

Y

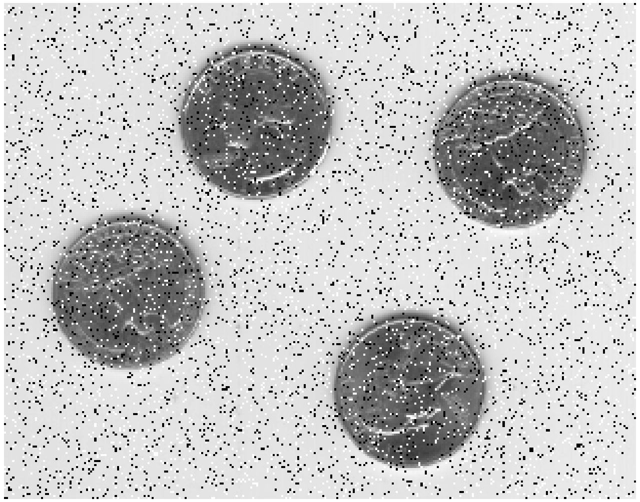
0	225	225	226	226	226	226	226
225	225	226	226	226	226	226	226
225	226	226	226	226	226	226	226
226	226	225	**225**	226	226	226	226
225	225	225	225	226	226	226	226
225	225	225	226	226	226	226	226
225	225	225	226	226	226	226	226
226	226	226	226	226	226	226	226

$\hat{X}$

Sorted: [0, 0, 0, 225, <u>**225**</u>, 225, 226, 226, 226]

Image Example

P=0.1

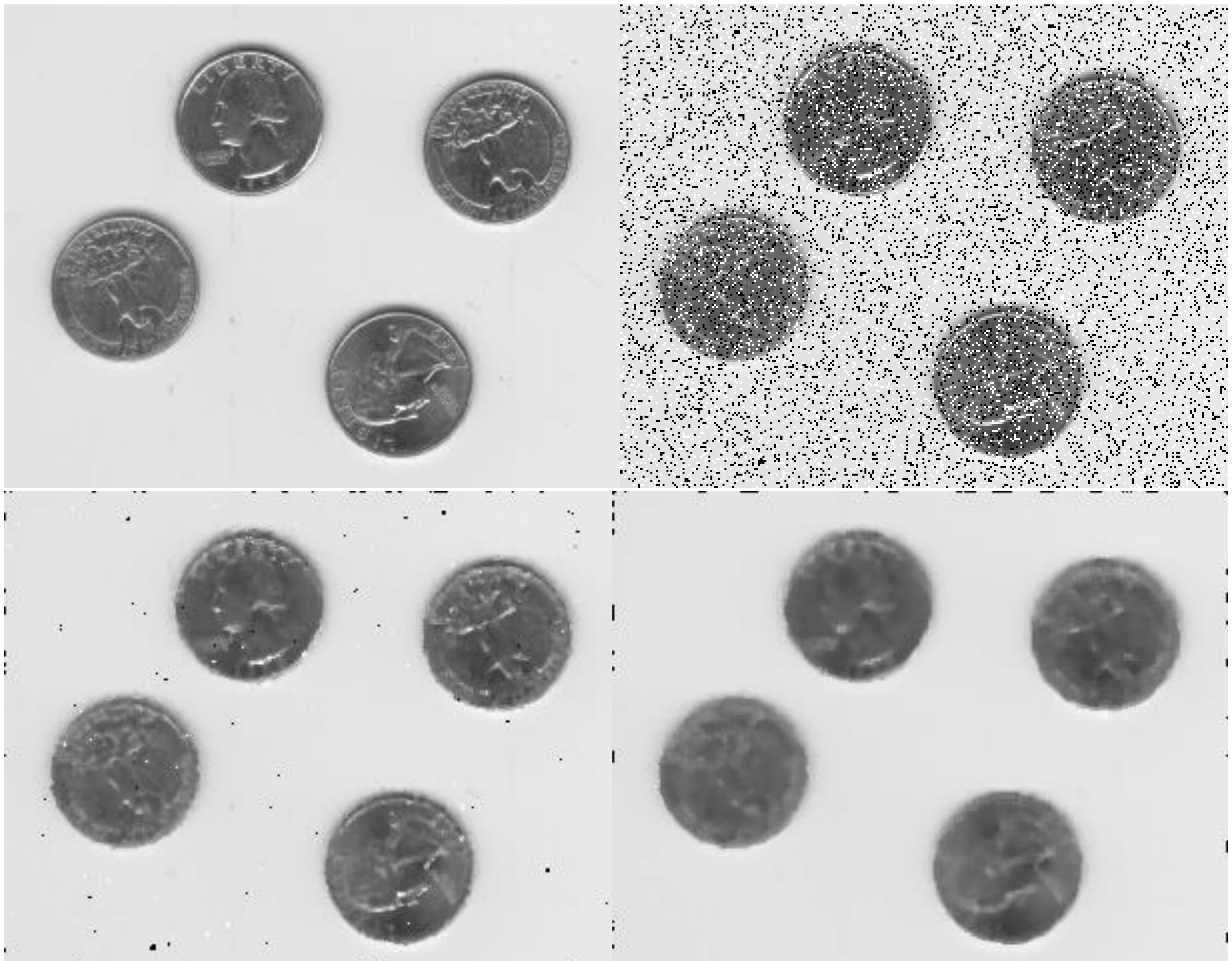


Noisy image Y

denoised image $\hat{X}$

3-by-3 window

Image Example (Cont'd)



clean

noisy (p=0.2)

3-by-3 window

5-by-5 window

Reflections

- What is good about median operation?
  - Since we know impulse noise appears as black (minimum) or white (maximum) dots, taking median effectively suppresses the noise
- What is bad about median operation?
  - It affects clean pixels as well
  - Noticeable edge blurring after median filtering

Improving Median Filtering

- Can we get rid of impulse noise without affecting clean pixels?
  - Yes, if we know where the clean pixels are or equivalently where the noisy pixels are
- How to detect noisy pixels?
  - They are black or white dots

Median Filtering with Noise Detection

Noisy image Y

↓

Median filtering

x=medfilt2(y,[2*T+1,2*T+1]);

↓

Noise detection

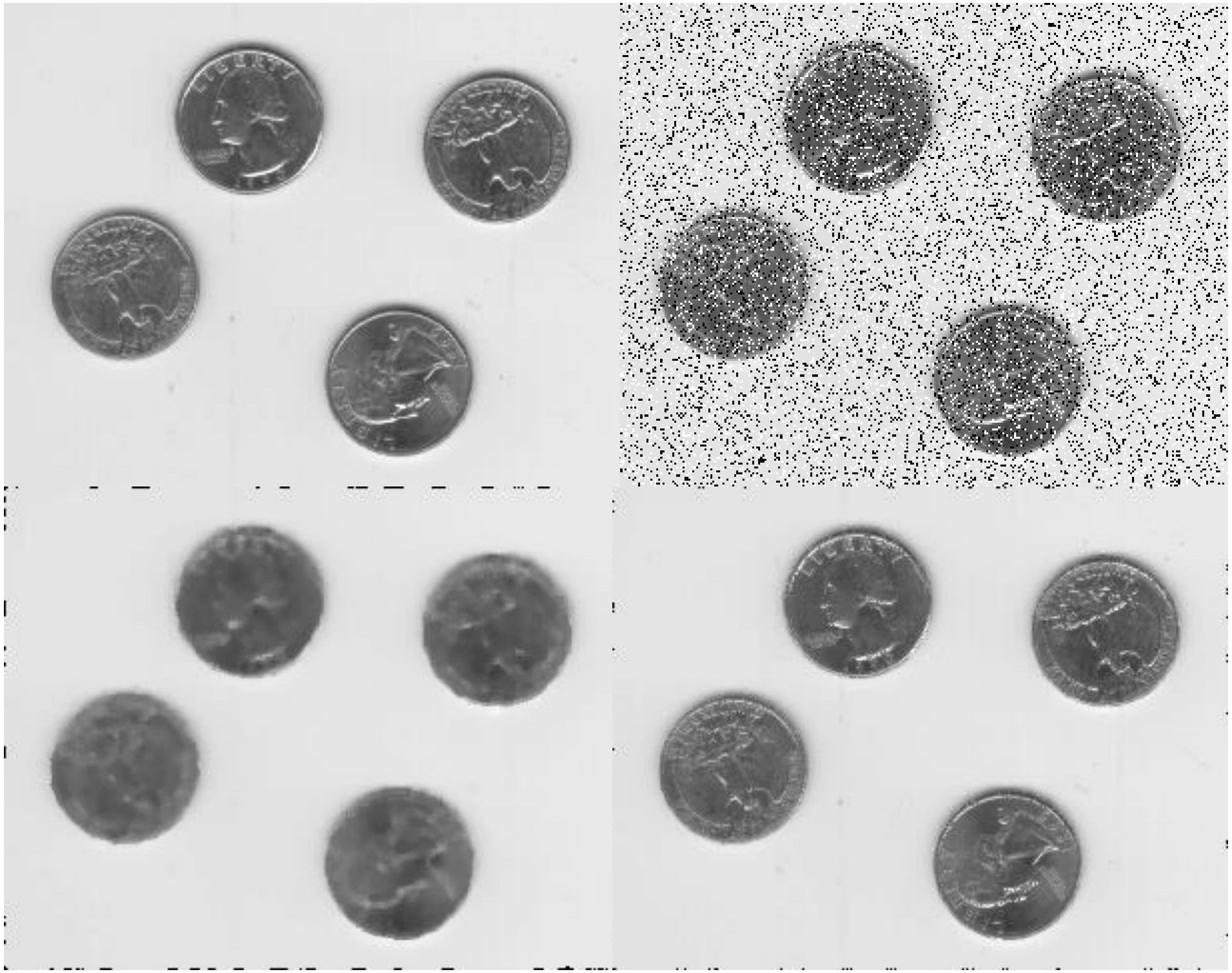
C=(y==0)|(y==255);

↓

Obtain filtering results

xx=c.*x+(1-c).*y;

Image Example



clean

noisy (p=0.2)

w/o noise detection

with noise detection

Additive White Gaussian Noise

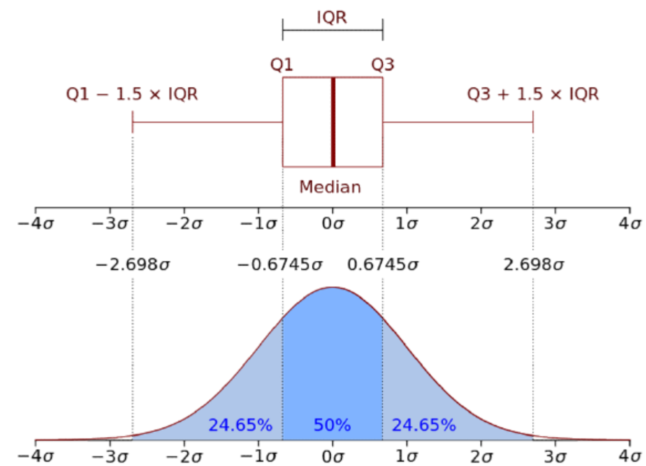
Definition

Each pixel in an image is disturbed by a Gaussian random variable
With zero mean and variance σ^2

$$Y(i,j) = X(i,j) + N(i,j),$$

$$N(i,j) \sim N(0,\sigma^2), 1 \le i \le H, 1 \le j \le W$$

X: noise-free image, Y: noisy image



Note: unlike impulse noise situation, every pixel in the image contaminated by AWGN is noisy

Numerical Example

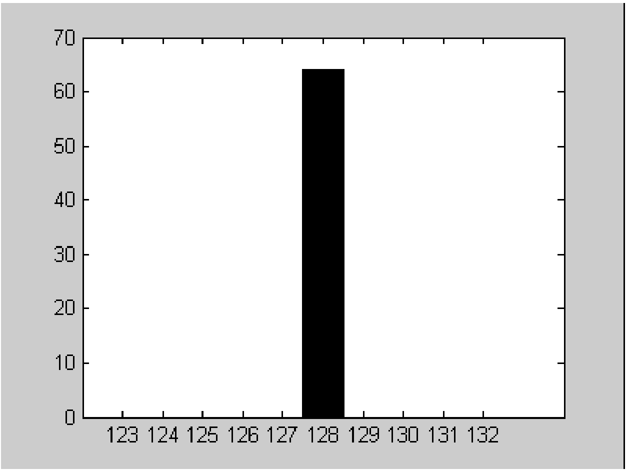
$\sigma^2 = 1$

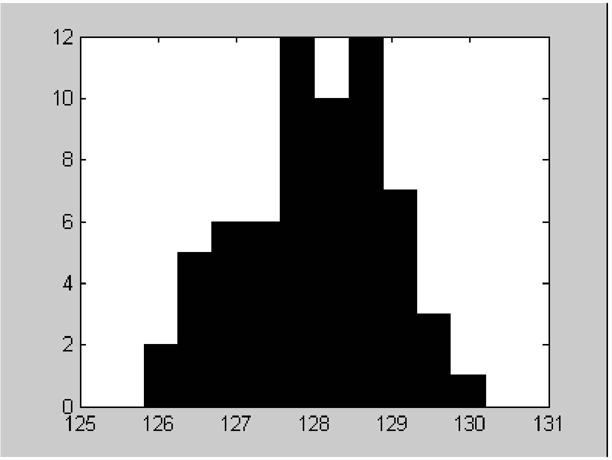
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128

X

128	128	129	127	129	126	126	128
126	128	128	129	129	128	128	127
128	128	128	129	129	127	127	128
128	129	127	126	129	129	129	128
127	127	128	127	129	127	129	128
129	130	127	129	127	129	130	128
129	128	129	128	128	128	129	129
128	128	130	129	128	127	127	126

Y





MATLAB Command

>Y = imnoise(X,’gaussian',m,v)

>Y = X+m+randn(size(X))*v;

<u>Note:</u> rand() generates random numbers uniformly distributed over [0,1]

randn() generates random numbers observing Gaussian distribution N(0,1)

Image Denoising

Noisy image Y

filtering algorithm → denoised image $\hat{X}$

Question: Why not use median filtering?
Hint: the noise type has changed.

2D Convolution: Linear Time-Invariant Filters

$$
\underset{\mathbf{I}}{\begin{bmatrix}
0 & 1 & 1 & 1_{\times 1} & 0_{\times 0} & 0_{\times 1} & 0 \\
0 & 0 & 1 & 1_{\times 0} & 1_{\times 1} & 0_{\times 0} & 0 \\
0 & 0 & 0 & 1_{\times 1} & 1_{\times 0} & 1_{\times 1} & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}
*
\underset{\mathbf{K}}{\begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 1
\end{bmatrix}}
=
\underset{\mathbf{I} * \mathbf{K}}{\begin{bmatrix}
1 & 4 & 3 & \boxed{4} & 1 \\
1 & 2 & 4 & 3 & 3 \\
1 & 2 & 3 & 4 & 1 \\
1 & 3 & 3 & 1 & 1 \\
3 & 3 & 1 & 1 & 0
\end{bmatrix}}
$$

Kernel

MATLAB function: conv2(I, K)

Two Sequential 1D Filtering

- Separable 2D filtering can be viewed as two sequential 1D filtering operations
  - One along row direction and the other along column direction
  - The order of filtering does not matter

$$h(m,n) = h^1(m) \otimes h^1(n) = h^1(n) \otimes h^1(m)$$

h^1 : 1D filter

Numerical Example

1D filter

$h^1(m)=[1,1]'$, $h^1(n)=[1,-1]$

$$h^1(m) \otimes h^1(n) = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$h^1(n) \otimes h^1(m) = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

MATLAB command:

```
>h1=[1,1]';h2=[1,-1];
>conv2(h1,h2)
>conv2(h2,h1)
```

Fourier Transform (2D case)

$$F(w_1, w_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j(w_1 m + w_2 n)}$$

spatial-domain convolution

$$f(m,n) \otimes h(m,n)$$

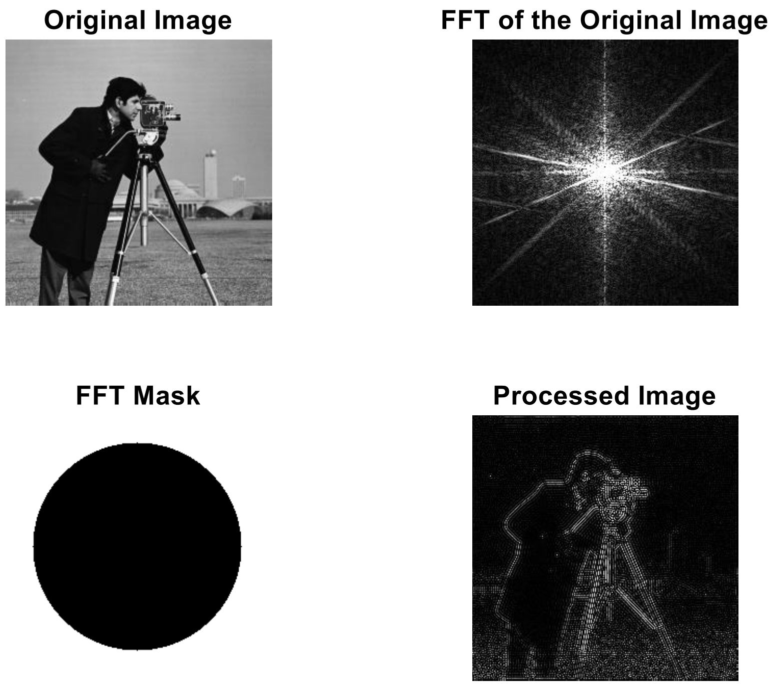
frequency-domain multiplication

$$F(w_1, w_2) H(w_1, w_2)$$

Note that the input signal is discrete while its FT is a continuous function

2D DFT

```
I = double(imread('cameraman.tif'))/255;
figure
subplot(2,2,1)
imshow(I)
title('Original Image')

subplot(2,2,2)
J = fftshift(fft2(I));
imshow(abs(J),[0 100])
title('FFT of the Original Image')

[M,N] = size(J);
xMAP = repmat(1:N,M,1);
yMAP = repmat((1:M)',1,N);
Center = [N/2 M/2];
R = 100;
Mask = zeros(M,N);
Mask(sqrt((xMAP-Center(1)).^2+(yMAP-
Center(2)).^2)>R) = 1;
subplot(2,2,3)
imshow(Mask,[0 1])
title('FFT Mask')

subplot(2,2,4)
K = J.*Mask;
I0 = real(ifft2(fftshift(K)));
imshow(I0,[0 0.1])
title('Processed Image')
```



Filter Examples

Low-pass (LP)

1D

$h^1(n)=[1,1]$

$\downarrow$

$|h^1(w)|=2\cos(w/2)$

$\downarrow$

2D

$h(n)=[1,1;1,1]$

$\downarrow$

$|h(w_1,w_2)|=4\cos(w_1/2)\cos(w_2/2)$

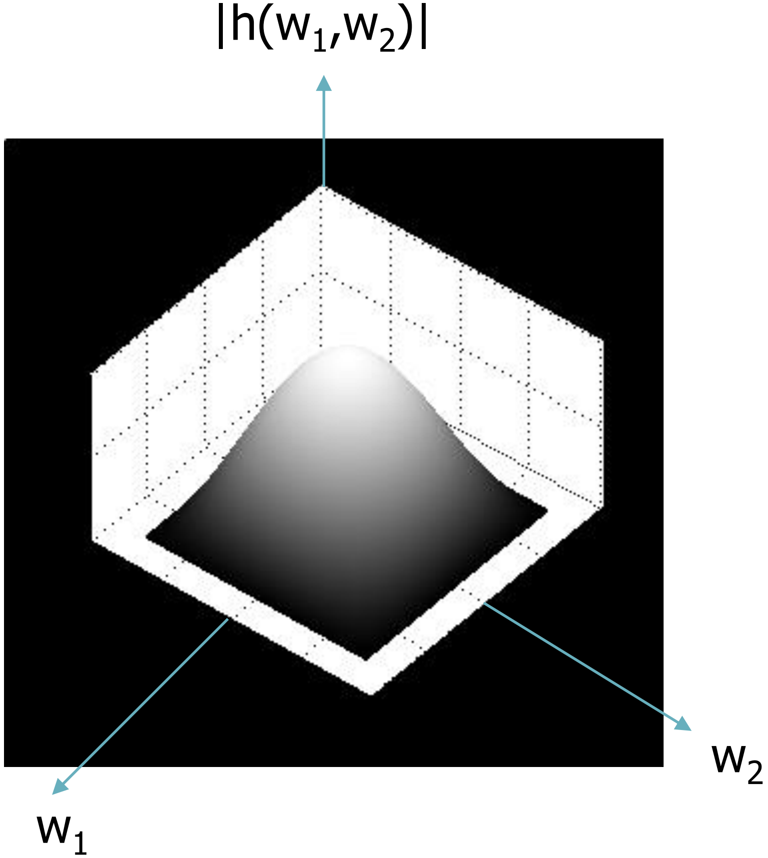
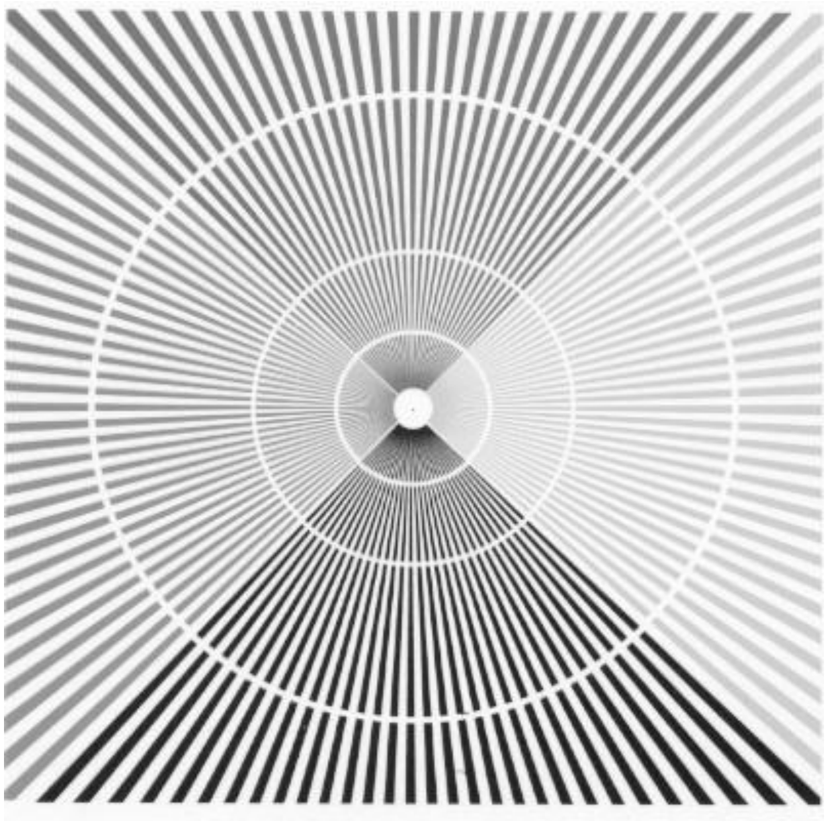
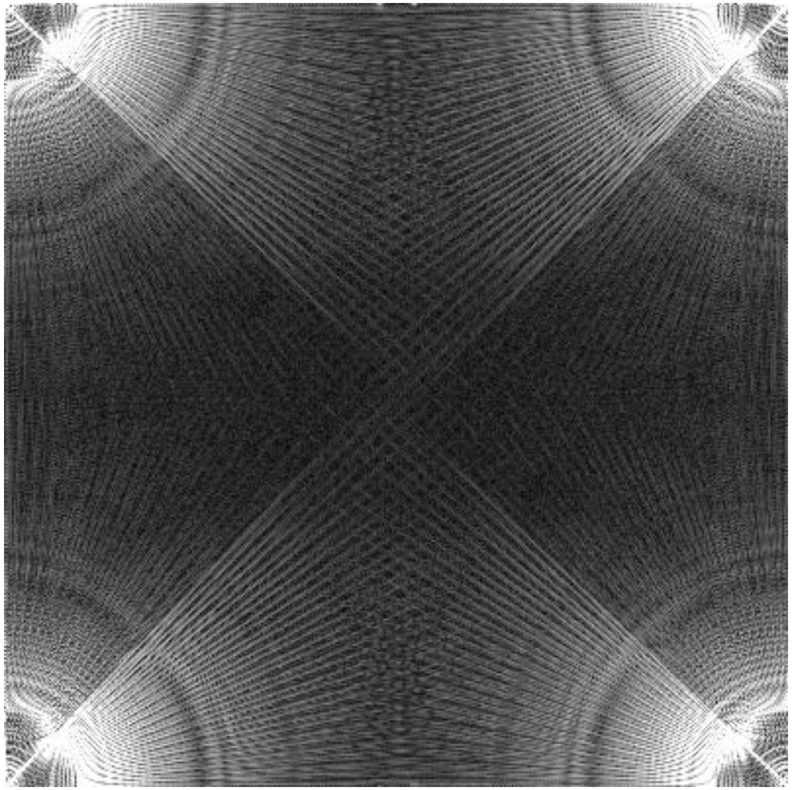


Image DFT Example

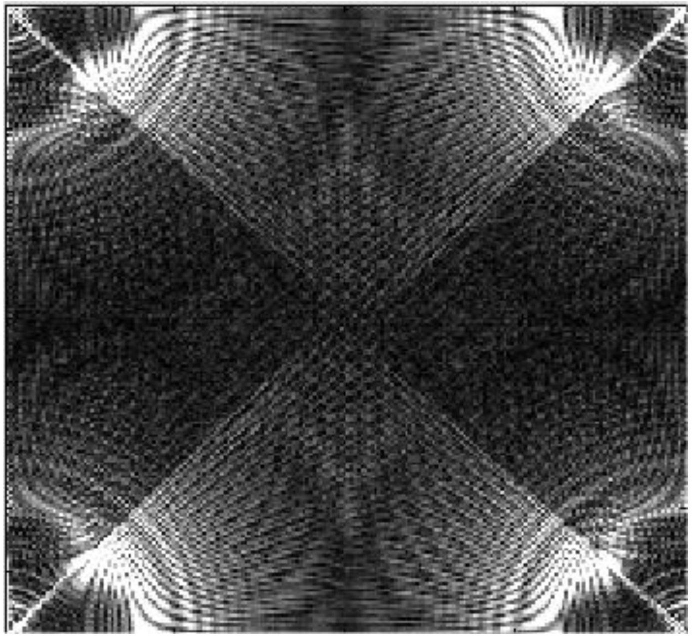


Original ray image **X**



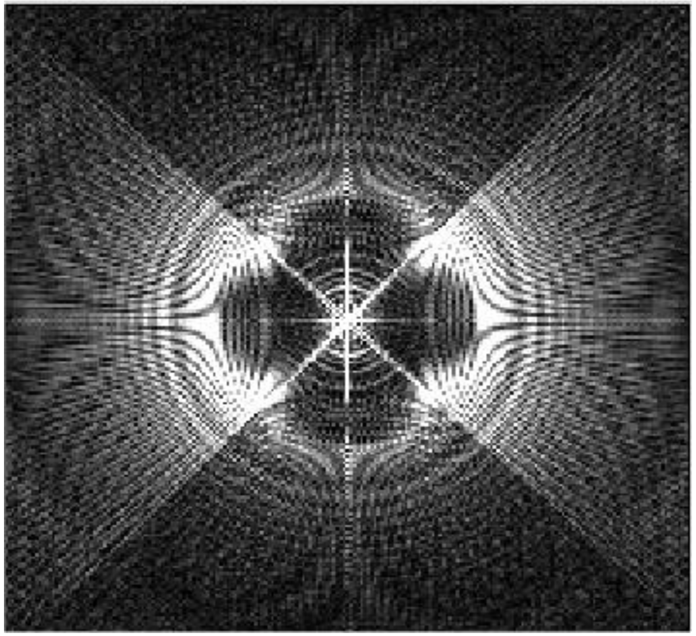
choice 1: **Y**=fft2(**X**)

Image DFT Example (Cont’d)



choice 1: **Y**=fft2(**X**)

Low-frequency at four corners



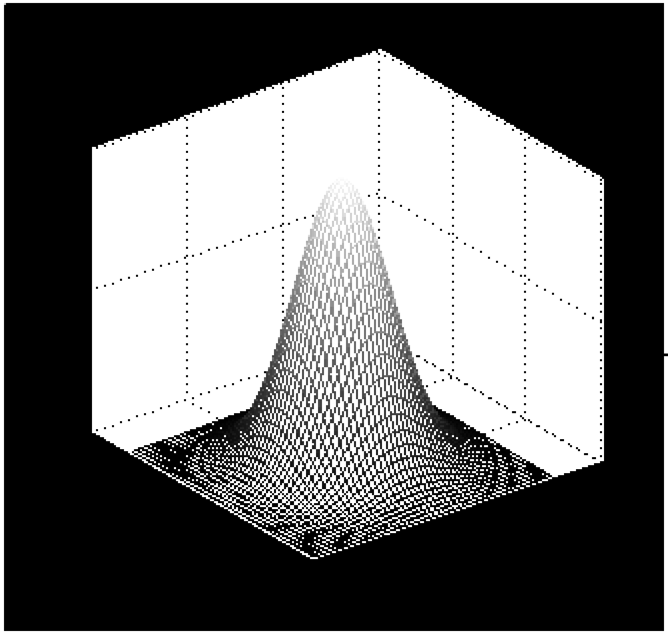
choice 2: **Y**=fftshift(fft2(**X**))

Low-frequency at the center

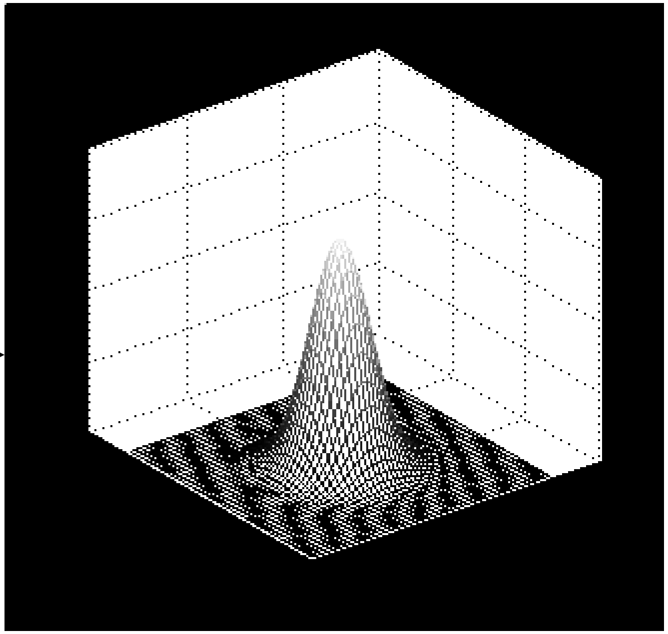
FFTSHIFT Shift zero-frequency component to center of spectrum.

Gaussian Filter

Note: Standard deviation must be at least 3 samples



FT



$$h(m,n) = \frac{1}{2\pi\sigma^2} e^{-(\frac{m^2+n^2}{2\sigma^2})}$$

$$H(w_1,w_2) = e^{-\sigma^2 \frac{w_1^2+w_2^2}{2}}$$

MATLAB code: >h=fspecial('gaussian', HSIZE,SIGMA);

Image Example

noisy

PSNR=20.2dB

(σ=25)

denoised

PSNR=24.4dB

(σ=1)

denoised

PSNR=22.8dB

(σ=1.5)

Matlab functions: imfilter, filter2

PSNR Definition

- Consider I and its noisy version K

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Hammer-Nail Analogy

Gaussian filter

salt-pepper/
impulse noise

median filter

Gaussian noise

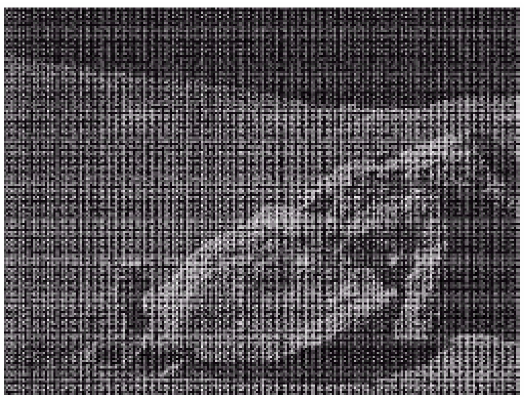
???

periodic noise

Periodic Noise

- Source
  - Electrical or electromechanical interference during image acquistion
- Characteristics
  - Spatially dependent
  - Periodic – easy to observe in frequency domain
- Processing method
  - Suppressing noise component in frequency domain

Image Example



spatial

Frequency (note the four pairs of bright dots)

Band Rejection Filter

$$H(w_1, w_2) = \begin{cases} 0 & D - \dfrac{W}{2} \le \sqrt{w_1^2 + w_2^2} \le D + \dfrac{W}{2} \\ 1 & otherwise \end{cases}$$

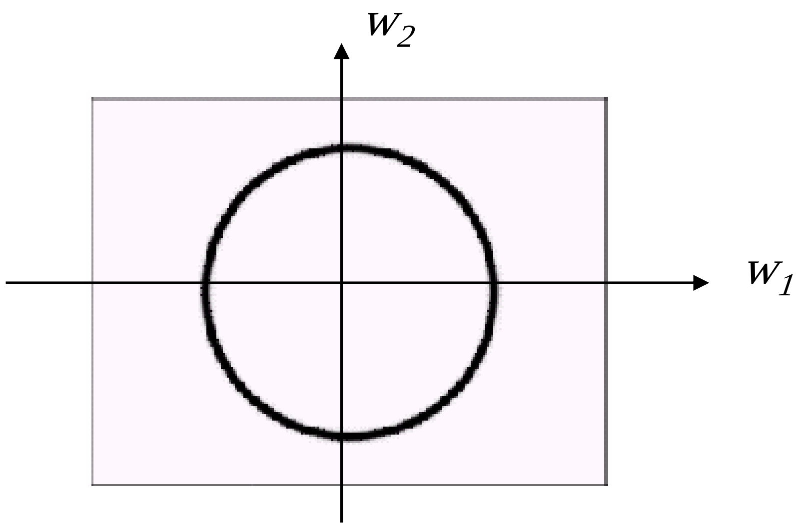
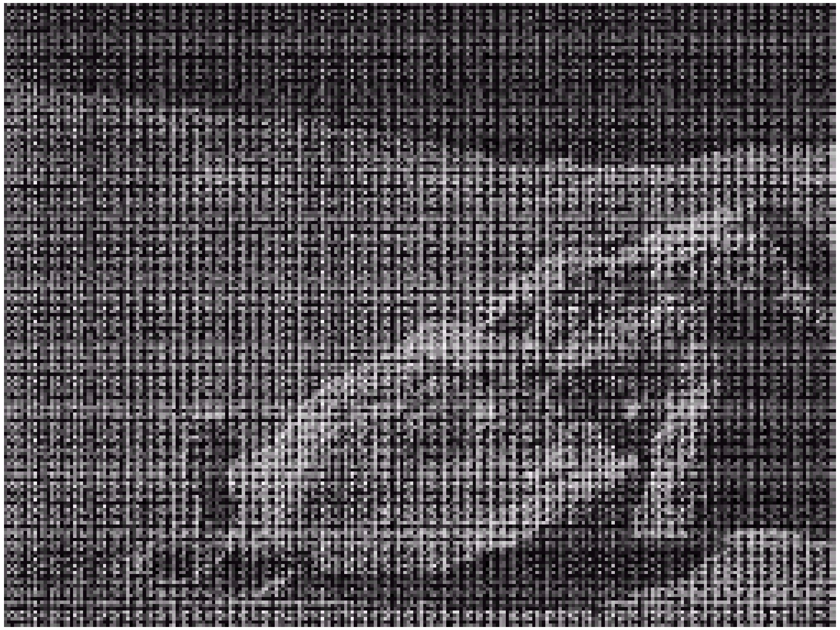
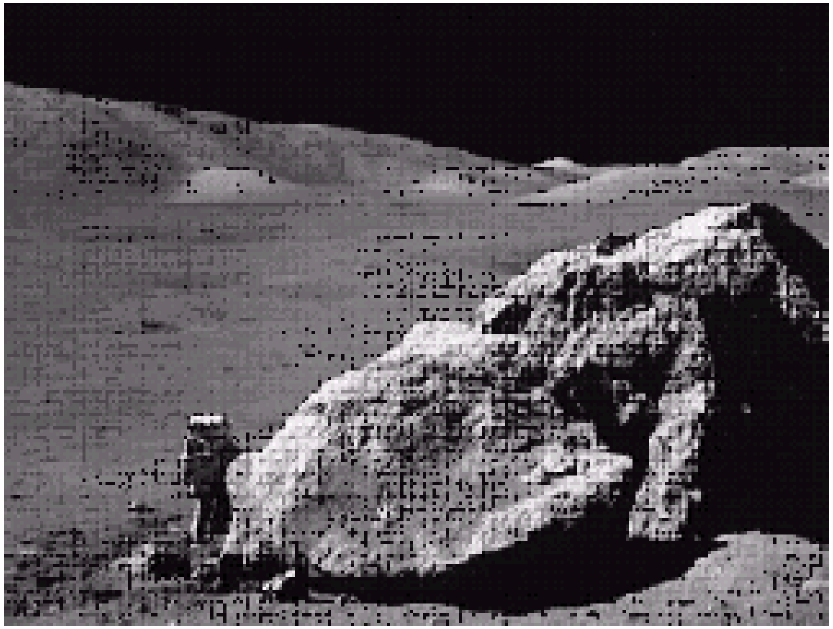


Image Example



Before filtering



After filtering

تمرین ۶

- حذف نویز گوسی
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

آزمون مستمر سوم

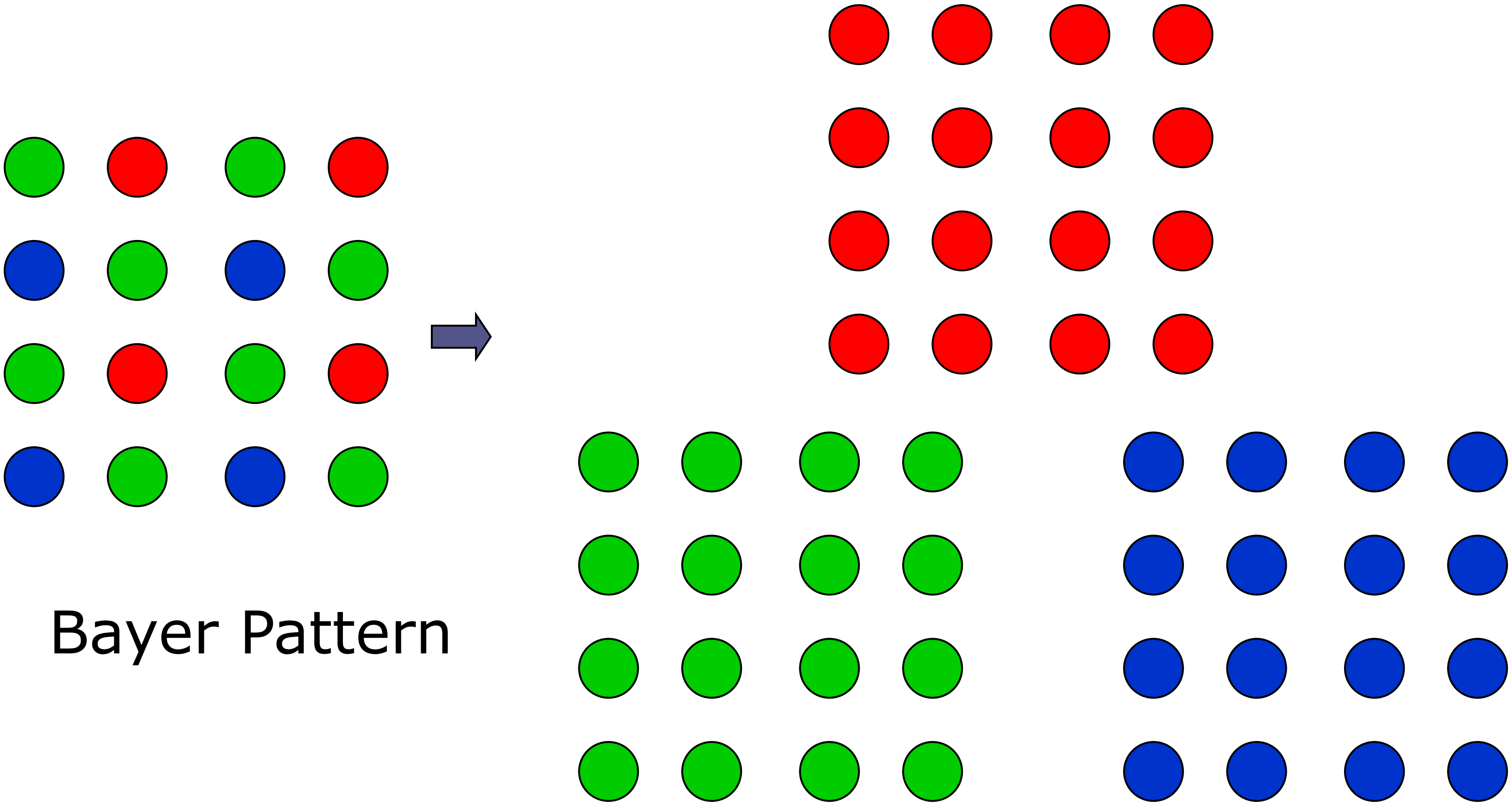
- مقابله با نویز (۱ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.

بخش چهارم- درون‌یابی تصویر و تبدیل‌های هندسی

Introduction

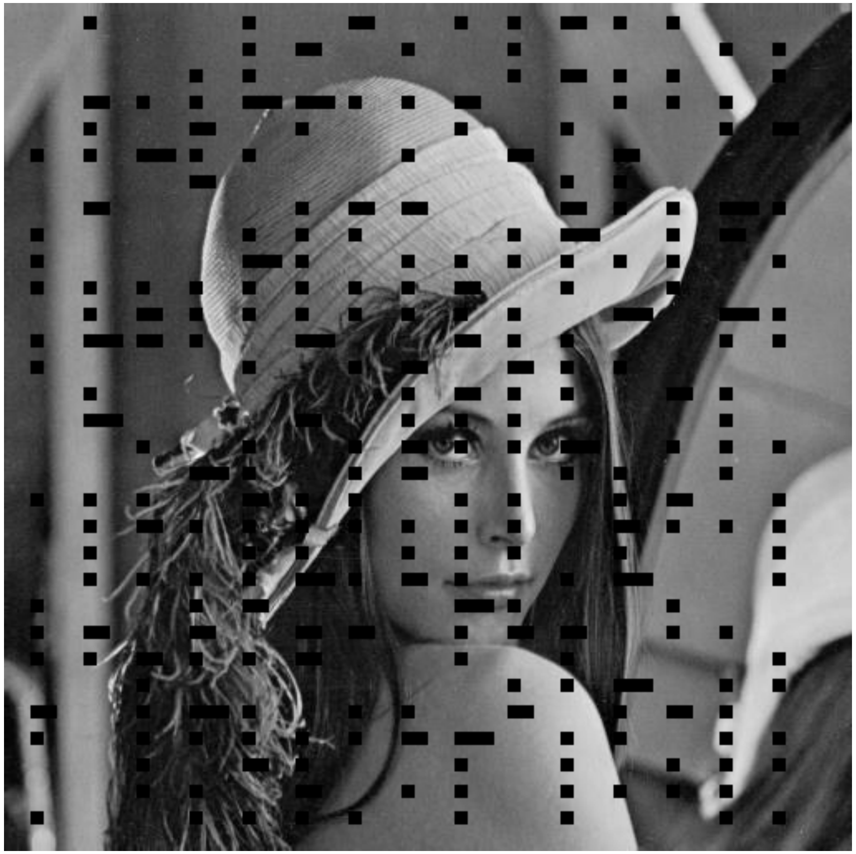
- What is image interpolation?
  - An image f(x,y) tells us the intensity values at the integral lattice locations, i.e., when x and y are both integers
  - Image interpolation refers to the "guess" of intensity values at missing locations, i.e., x and y can be arbitrary
  - Note that it is just a guess (Note that all sensors have finite sampling distance)

Image Demosaicing (Color-Filter-Array Interpolation)



Bayer Pattern

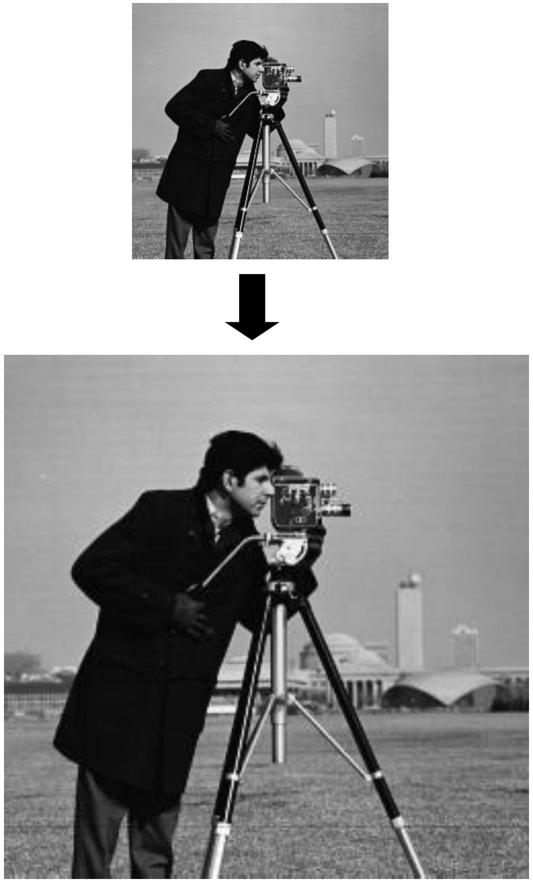
Error Concealment



damaged

interpolated

Resolution Enhancement



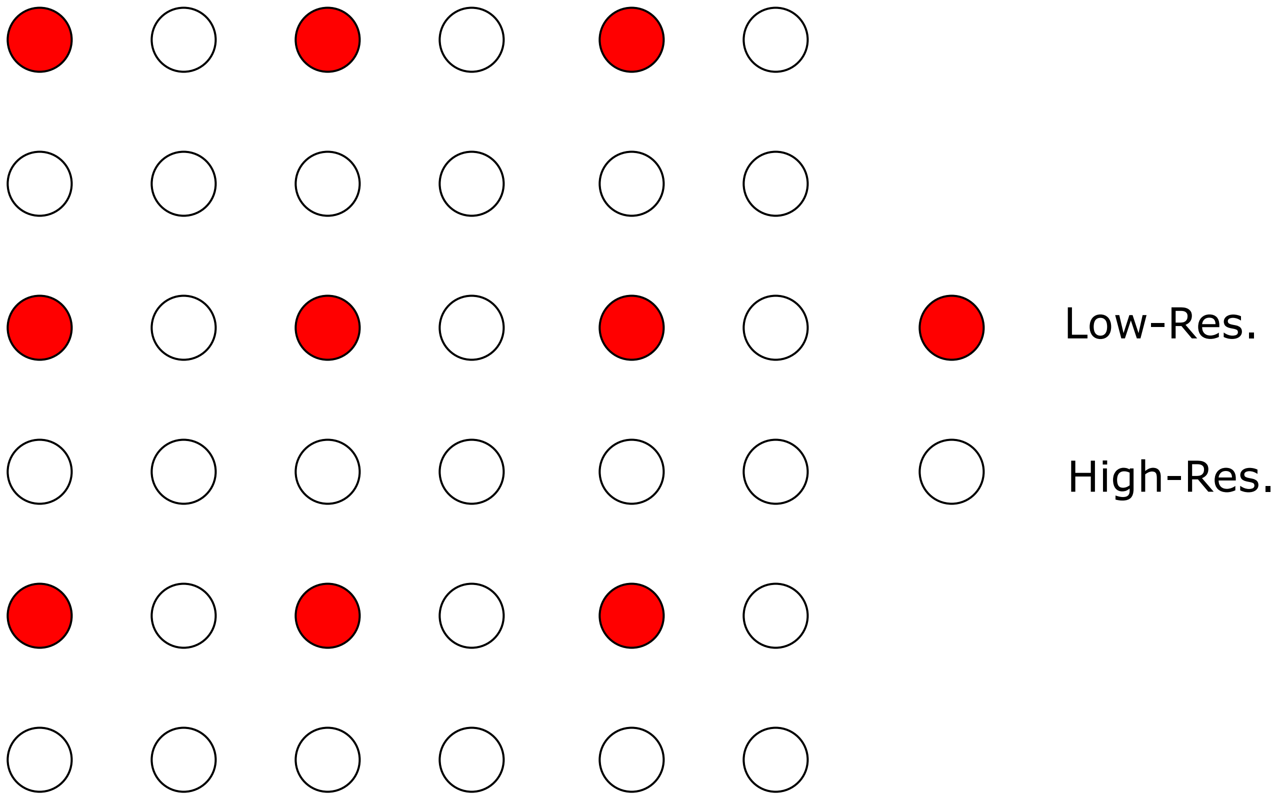
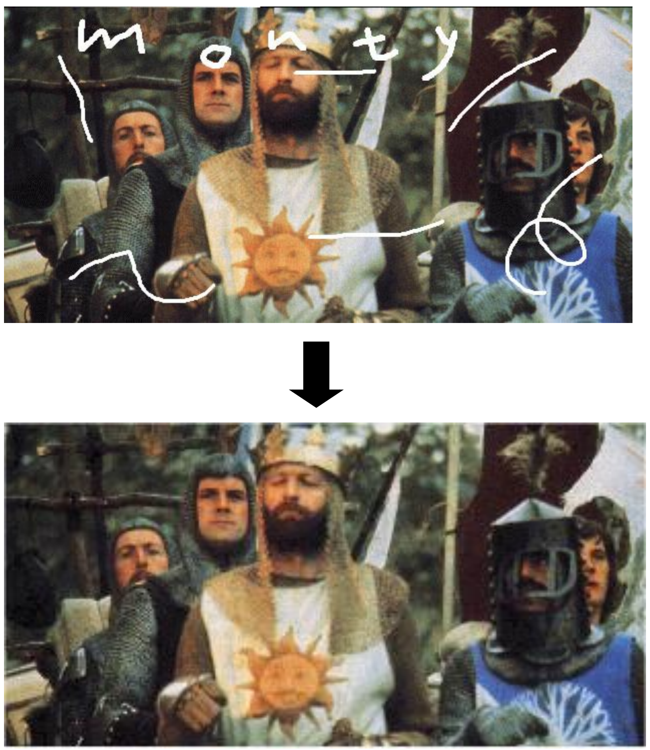
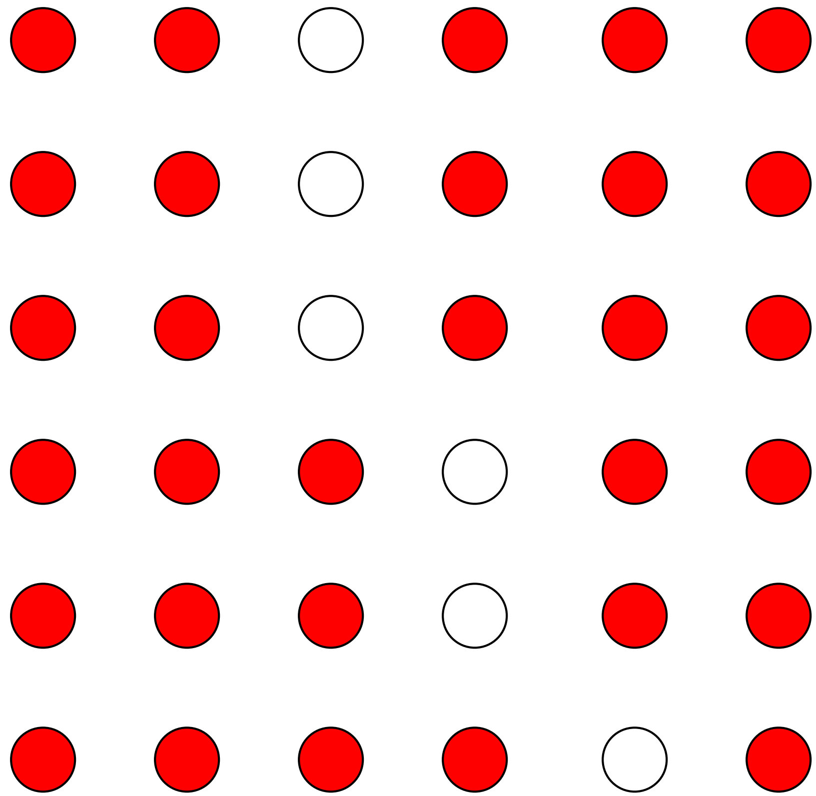


Image Inpainting





Non-damaged Damaged

Basic Methods

a	b
c	d

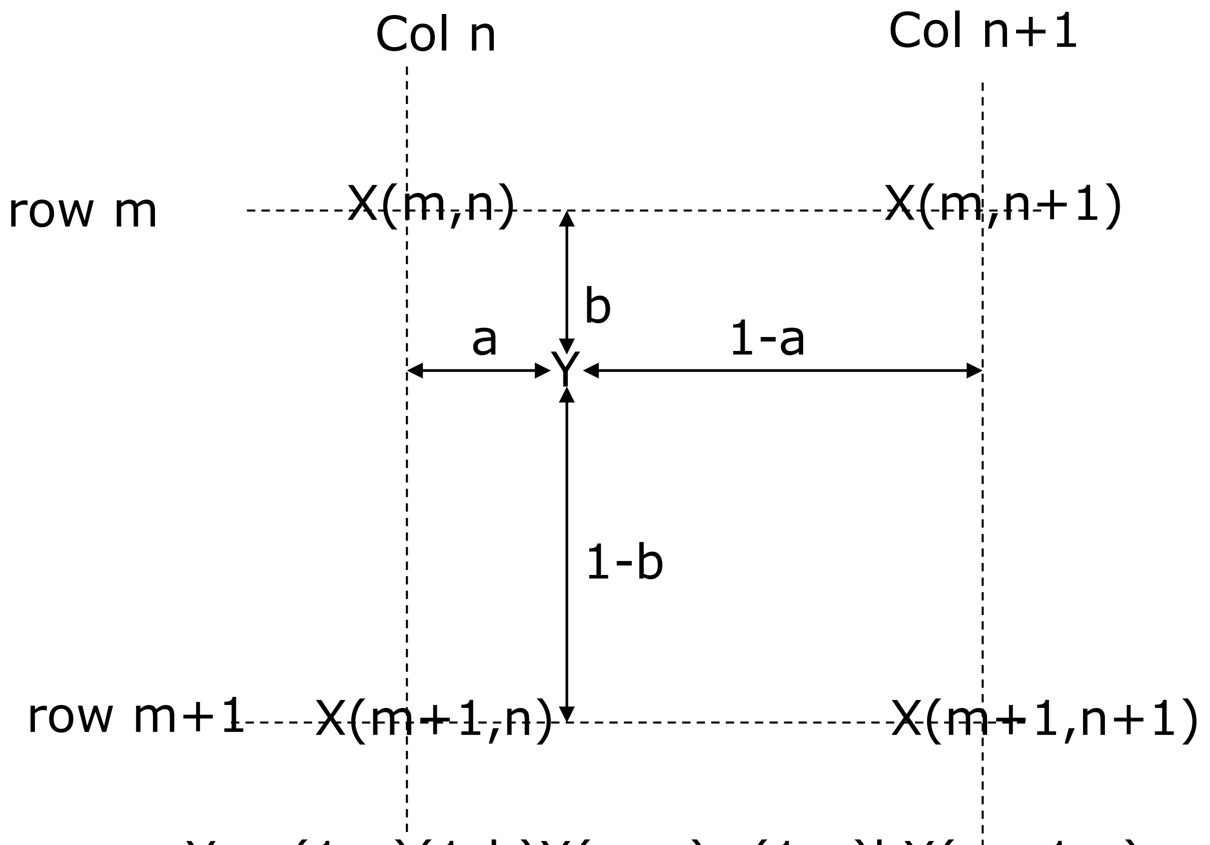
zero-order
(Nearest Neighbor)

a	a	b	b
a	a	b	b
c	c	d	d
c	c	d	d

first-order
(Bilinear Interpolation)

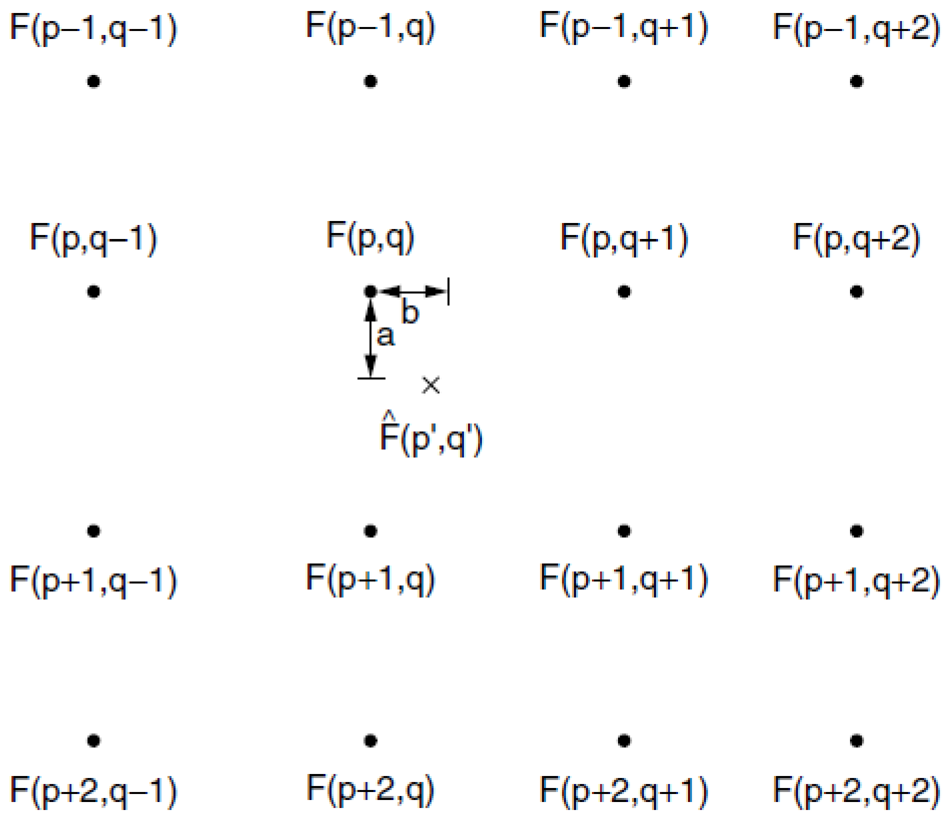
a	(a+b)/2	b
(a+c)/2	(a+b+c+d)/4	(b+d)/2
c	(c+d)/2	d

Bilinear Interpolation



Y = (1-a)(1-b)X(m,n)+(1-a)bX(m+1,n)
+a(1-b)X(m,n+1)+abX(m+1,n+1)

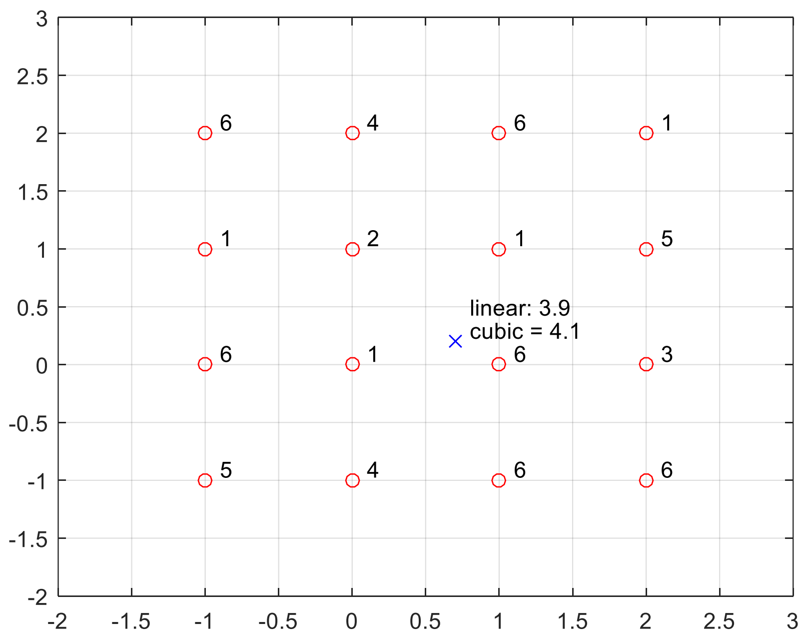
Bicubic Interpolation



$$F(p', q') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} F(p+m, q+n) R_C\{(m-a)\} R_C\{-(n-b)\}$$

Bicubic Interpolation (cont'd)

```
rng('default')
[X,Y] = meshgrid(-1:2);
V = randi([1 6],size(X));

plot(X,Y,'or');
xlim([min(X(:))-1 max(X(:))+1]);
ylim([min(Y(:))-1 max(Y(:))+1]);
for i=1:numel(X)
    text(X(i)+0.1,Y(i)+0.1,sprintf('%d',V(i)));
end
grid on

x0 = 0.7;
y0 = 0.2;
vq = interp2(X,Y,V,x0,y0,'cubic');
hold on
plot(x0,y0,'xb')
text(x0+0.1,y0+0.1,sprintf('cubic = %0.1f',vq));

vq = interp2(X,Y,V,x0,y0,'linear');
hold on
plot(x0,y0,'xb')
text(x0+0.1,y0+0.3,sprintf('linear: %0.1f',vq));
```



Bicubic Interpolation (cont'd)

- General form of cubic convolution interpolation function
  - a: tuning parameter
    - **Image-Dependent:** For example, by using local autocorrelation
    - **Image-Independent:** a=-1/2

$$R_c(x) = \begin{cases} A_1|x|^3 + B_1|x|^2 + C_1|x| + D_1 & \text{for } 0 \le |x| \le 1 \\ A_2|x|^3 + B_2|x|^2 + C_2|x| + D_2 & \text{for } 1 < |x| \le 2 \end{cases}$$

1. $R_c(x) = 1$ at $x = 0$, and $R_c(x) = 0$ at $x = 1, 2$.

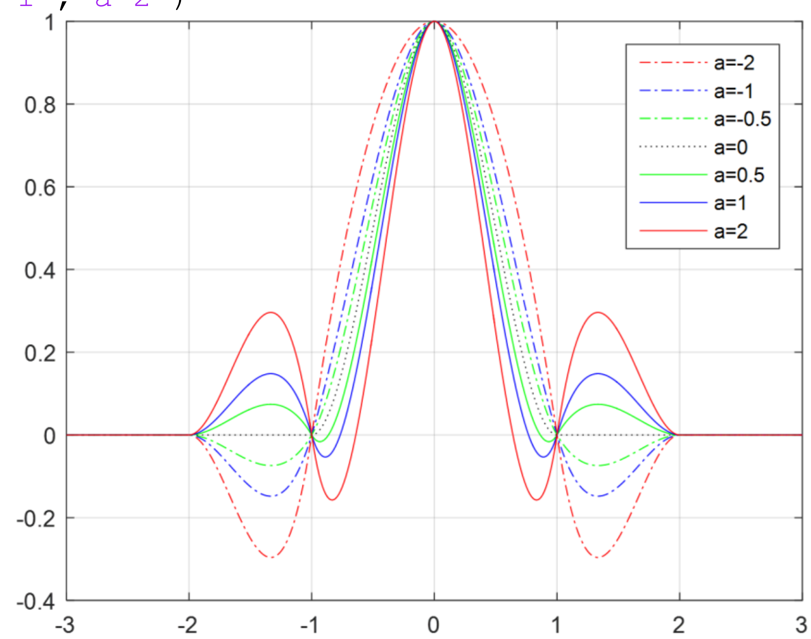
2. The first-order derivative $R'_c(x) = 0$ at $x = 0, 1, 2$.

$$\Downarrow$$

$$\boxed{R_c(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } 0 \le |x| \le 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| \le 2 \end{cases}}$$

Bicubic Interpolation (cont'd)

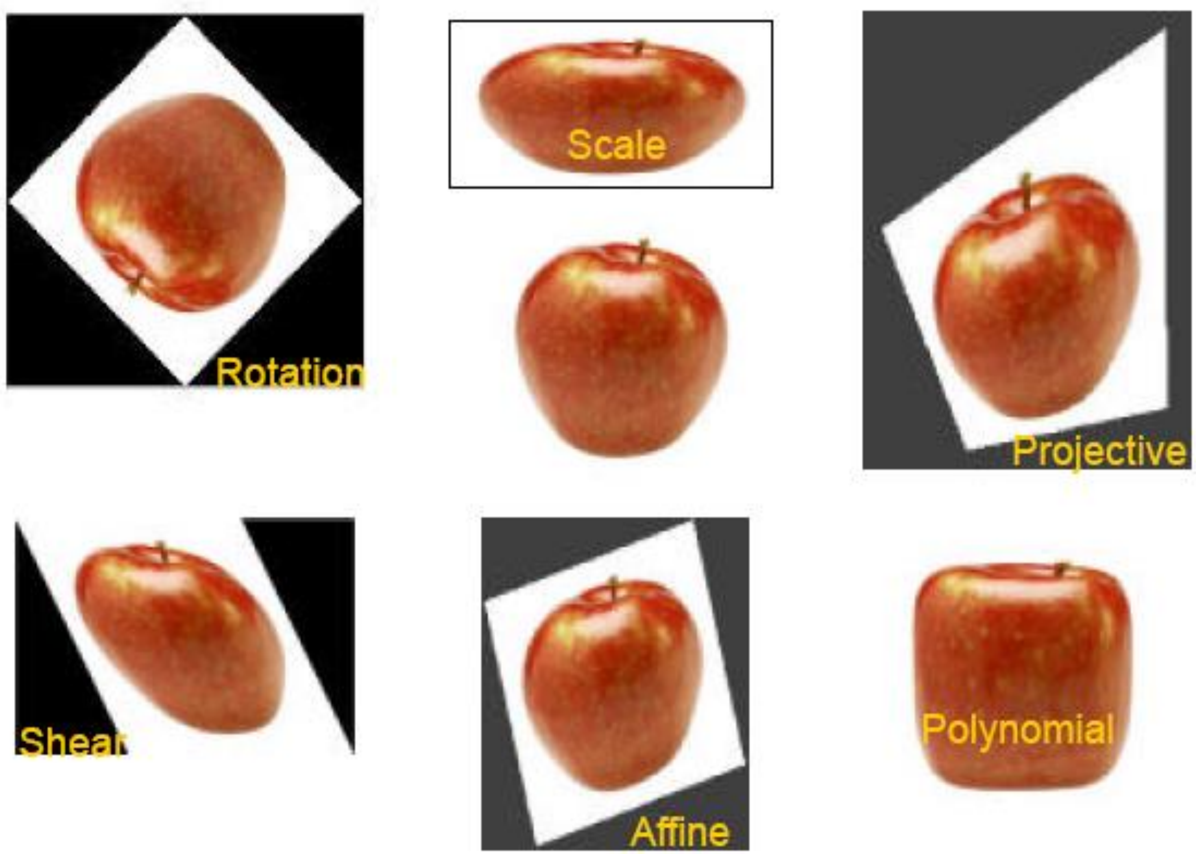
```
u = @(x) x>=0;
Rc = @(a,x) ((a+2)*abs(x).^3-(a+3)*x.^2+1).*(u(x+1)-u(x-1))+...
    (a*abs(x).^3-5*a*x.^2+8*a*abs(x)-4*a).*(u(x-1)-u(x-2)+u(x+2)-u(x+1));
x = (-3:0.01:3);
figure
linespec = {'-.r','-.b','-.g',':k','-g','-b','-r'};
cnt = 0;
for a=[-2 -1 -0.5 0 0.5 1 2]
    cnt = cnt+1;
    plot(x,Rc(a,x),linespec{cnt})
    hold on
end
grid on
legend('a=-2','a=-1','a=-0.5','a=0','a=0.5','a=1','a=2')
```



تمرین ۷

- درون‌یابی
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

Geometric Transformation



In the virtual space, you can have any kind of apple you want!

MATLAB functions: griddata, interp2, maketform, imtransform

Basic Principle

- $(x,y) \rightarrow (x',y')$ is a geometric transformation
  - We are given pixel values at (x,y) and want to interpolate the unknown values at (x',y')
  - Usually (x',y') are not integers and therefore we can use interpolation to guess values for the pixels of the transformed image.

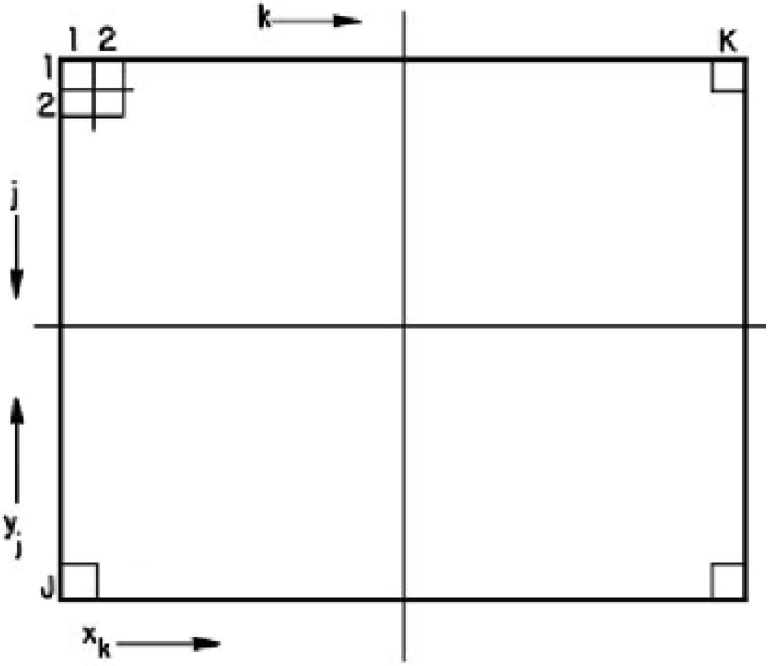
Cartesian Coordinate Representation (Example)

Transform from F to G

$$F(p,q) \quad 1 \le p \le P \quad 1 \le q \le Q \quad u_q = q - \tfrac{1}{2} \quad v_p = P + \tfrac{1}{2} - p$$

$$\Rightarrow$$

$$G(j,k) \quad 1 \le j \le J \quad 1 \le k \le K \quad x_k = k - \tfrac{1}{2} \quad y_j = J + \tfrac{1}{2} - j$$



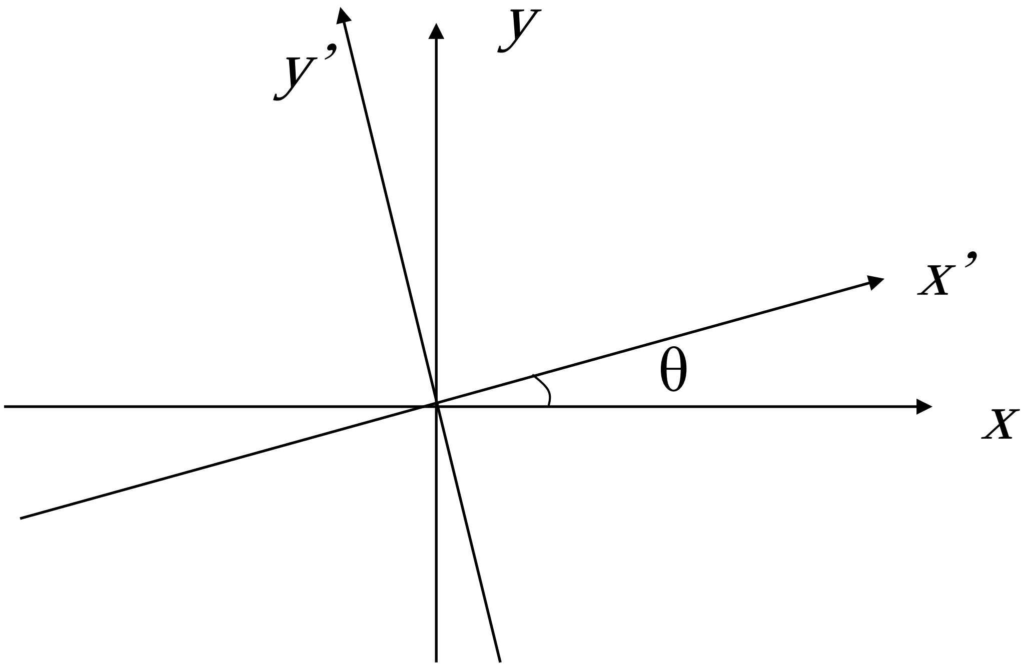
Reverse approach: For each output (transformed) point, find corresponding input and then use interpolation for that point.

Scale Example

a=2, b=1/2

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & \cdot \\ \cdot & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation

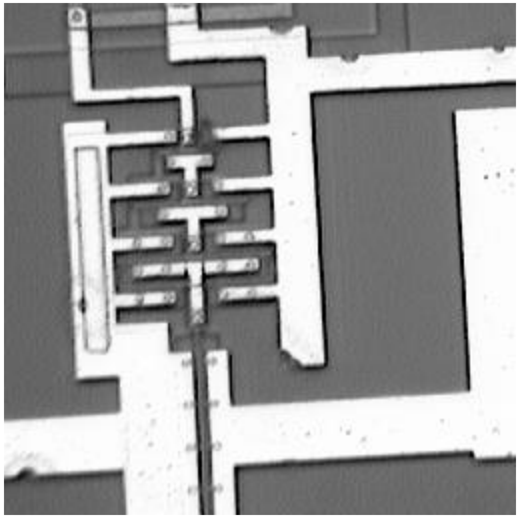


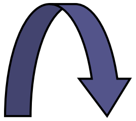
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x_k = u_q \cos\theta - v_p \sin\theta$$

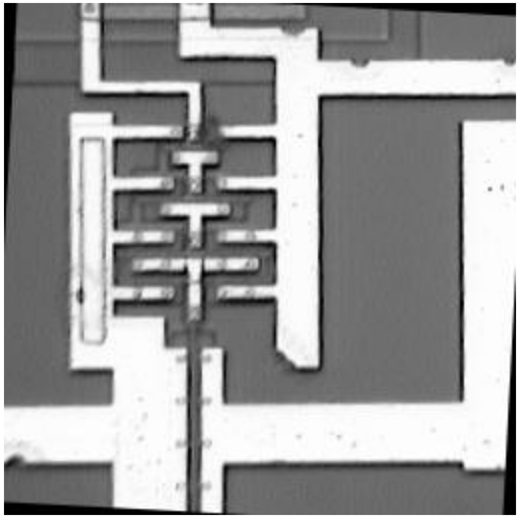
$$y_j = u_q \sin\theta + v_p \cos\theta$$

Rotation Example





$\theta=3^{o}$

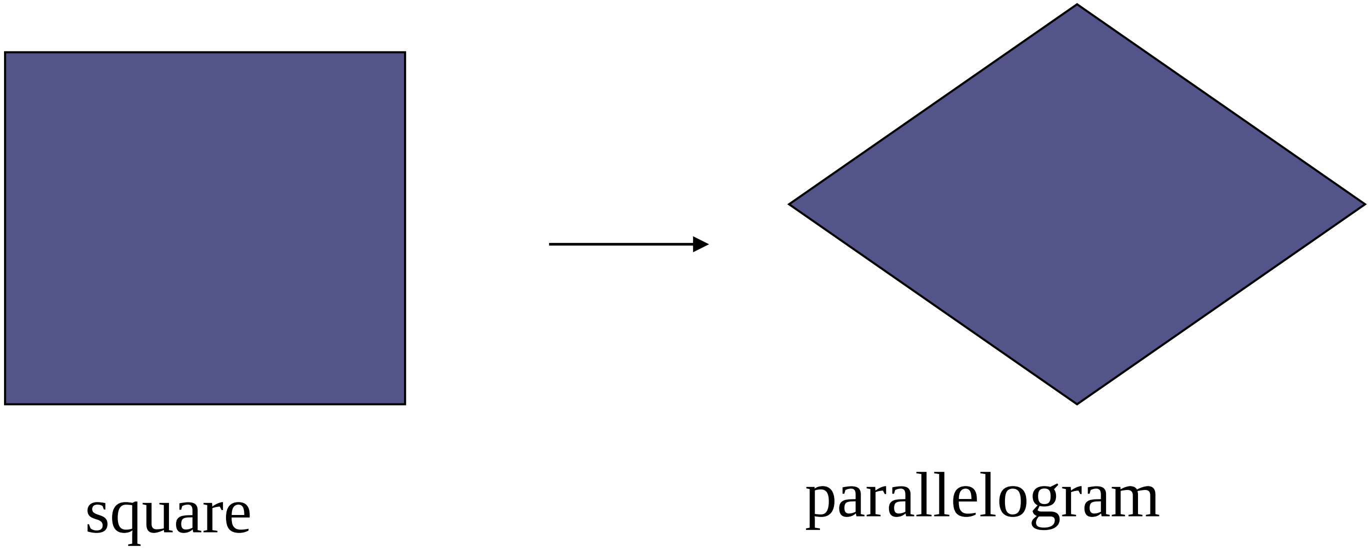


Rotation Example: MATLAB Code

```
index2coordinate = @(P,Q,pq) [pq(2)-(Q+1)/2 (P+1)/2-pq(1)];
coordinate2index = @(P,Q,uv) [(P+1)/2-uv(2) uv(1)+(Q+1)/2];

I1 = double(imread('cameraman.tif'))/255;
[P1,Q1] = size(I1);

theta = -30;
T = [cosd(theta) -sind(theta) ; sind(theta) cosd(theta)];
Tinv = T^(-1);
size2 = abs(T)*[P1 Q1]';
P2 = round(size2(1));
Q2 = round(size2(2));
I2 = zeros(P2,Q2);
for p2 = 1:P2
    for q2 = 1:Q2
        uv2 = index2coordinate(P2,Q2,[p2 q2]);
        uv1 = Tinv*uv2';
        pq1 = coordinate2index(P1,Q1,uv1);
        if all(pq1<=[P1 Q1]) && all(pq1>=[1 1])
           I2(p2,q2) = I1(round(pq1(1)),round(pq1(2)));
        end
    end
end
figure
subplot(2,1,1)
imshow(I1)
subplot(2,1,2)
imshow(I2)
```

$\theta = 30^{o}$

Affine Transform



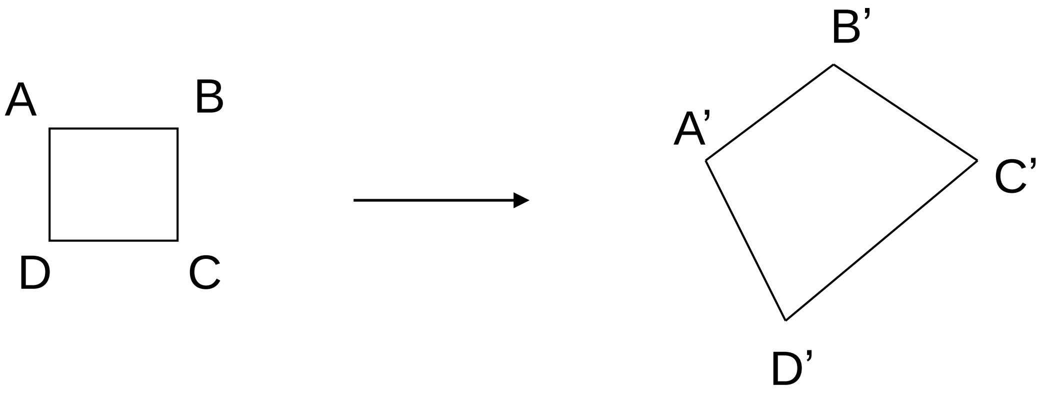
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Affine Transform Example

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} .5 & 1 \\ .5 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow$$

Projective Transform



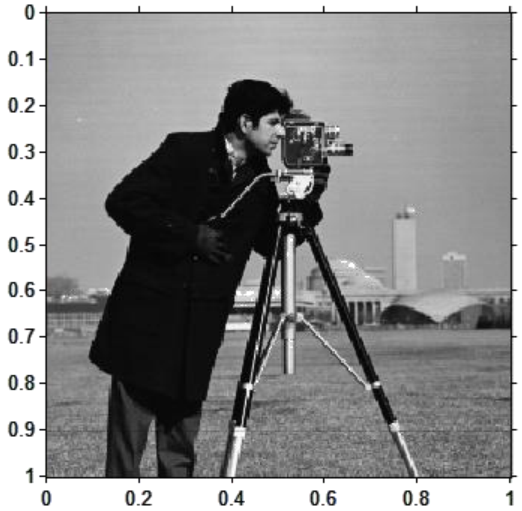
square

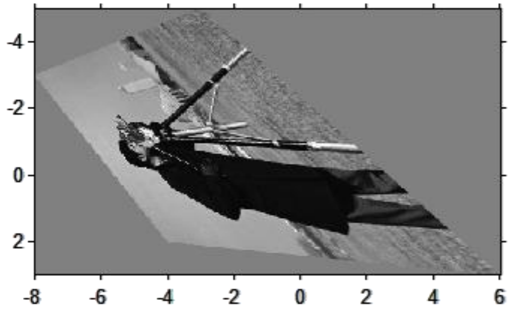
quadrilateral

$$x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1}$$

$$y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}$$

Projective Transform Example





[0 0; 1 0; 1 1; 0 1] ⟶ [-4 2; -8 -3; -3 -5; 6 3]

تمرین ۸

- تبدیل‌های هندسی
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

آزمون مستمر چهارم

- درون‌یابی و تبدیل‌های هندسی (۲ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.

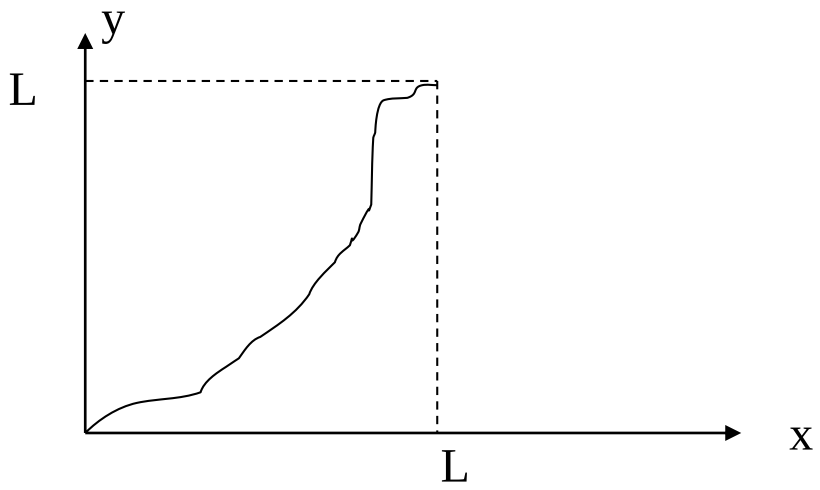
بخش پنجم - بهبود تصویر

Image is NOT Perfect Sometimes

Point Operations Overview

Point operations are **zero-memory** operations where a given gray level $x \in [0,L]$ is mapped to another gray level $y \in [0,L]$ according to a transformation

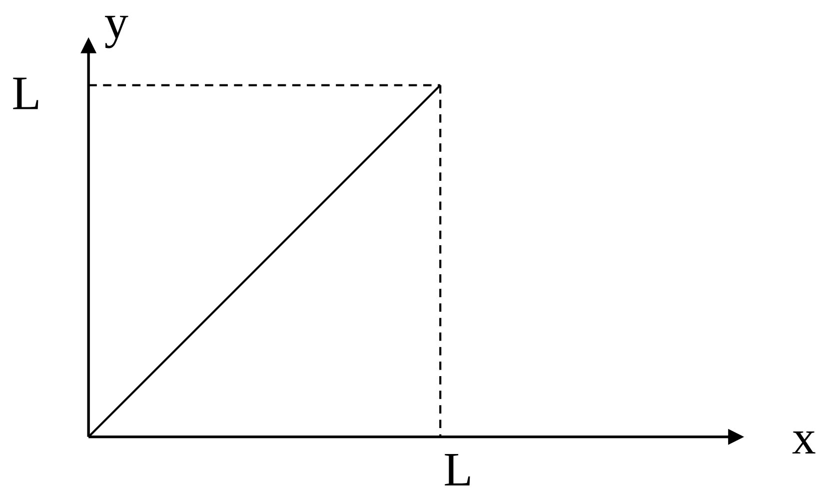
$$y = f(x)$$



L=255: for grayscale images

Lazy Man Operation

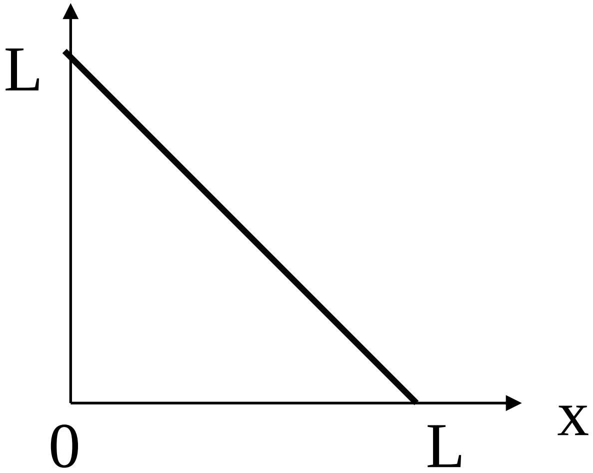
$$y = x$$



No influence on visual quality at all

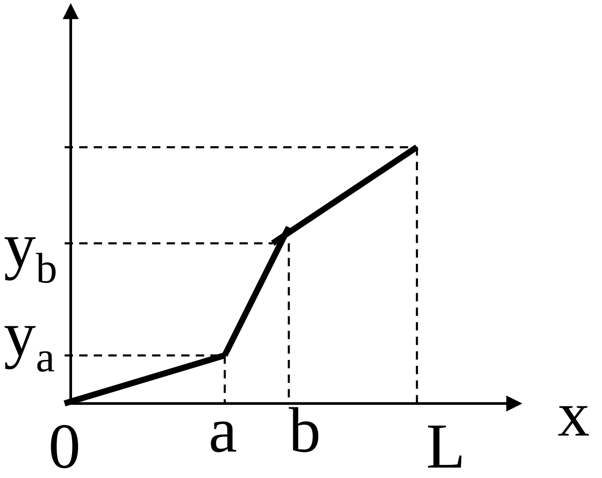
Digital Negative

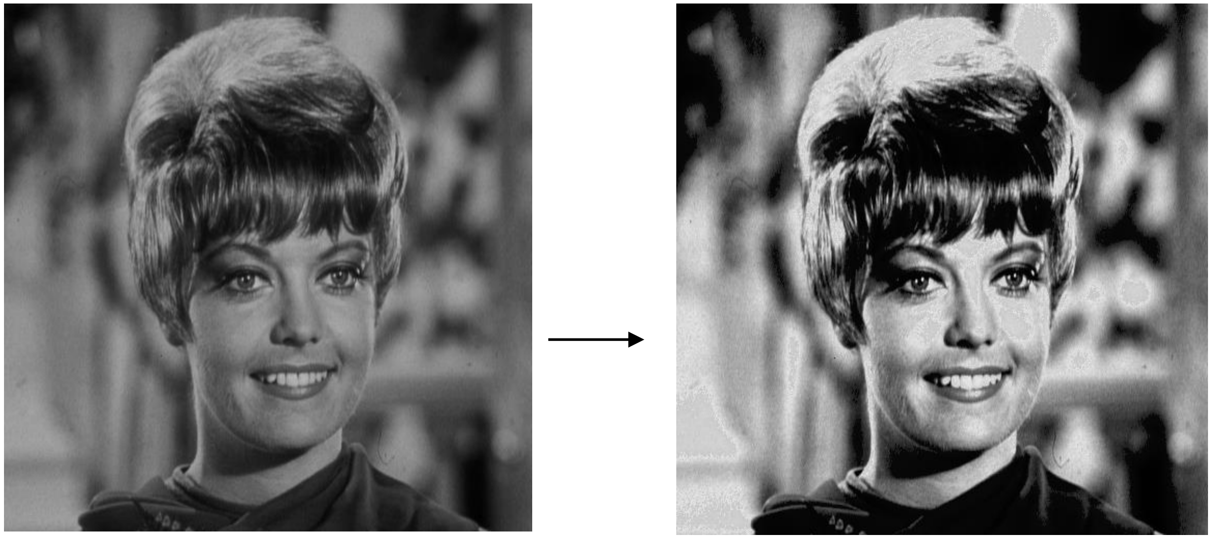
$$y = L - x$$



Contrast Stretching

$$y=\begin{cases}\alpha x & 0\le x<a\\ \beta(x-a)+y_a & a\le x<b\\ \gamma(x-b)+y_b & b\le x<L\end{cases}$$

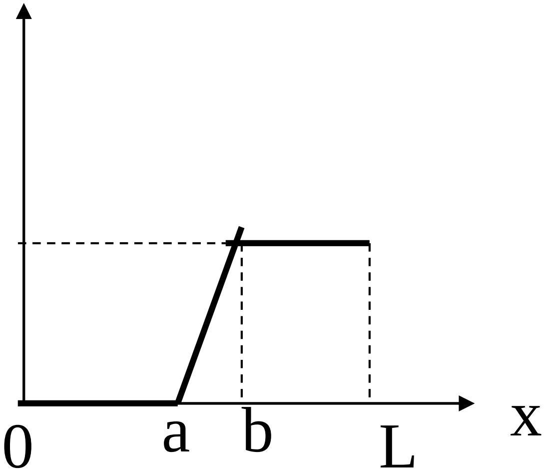




$$a=50, b=150, \alpha=0.2, \beta=2, \gamma=1, y_a=30, y_b=200$$

Clipping

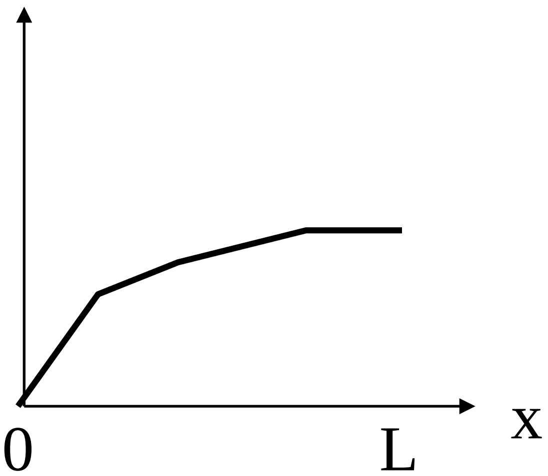
$$y = \begin{cases} 0 & 0 \le x < a \\ \beta(x-a) & a \le x < b \\ \beta(b-a) & b \le x < L \end{cases}$$



$a = 50, b = 150, \beta = 2$

Range Compression

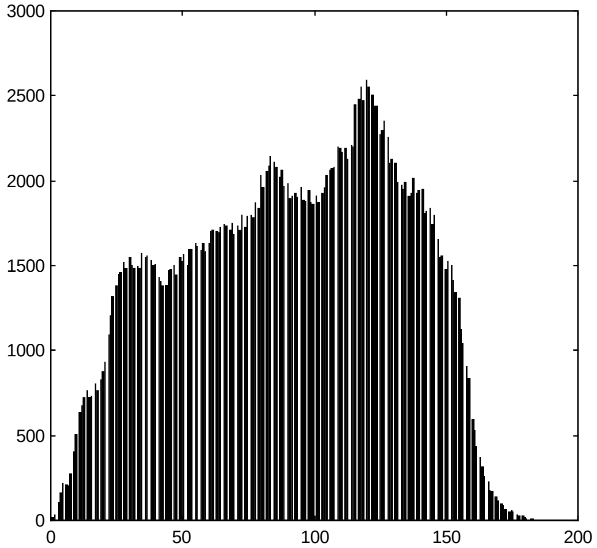
$$y = c\log_{10}(1+x)$$



c=100

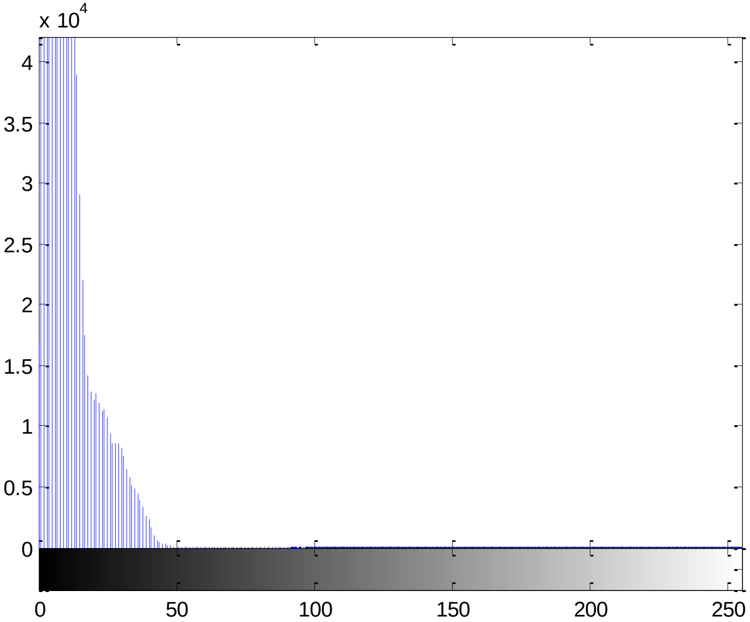
Histogram based Enhancement

- Histogram of an image represents the relative frequency of occurrence of various gray levels in the image



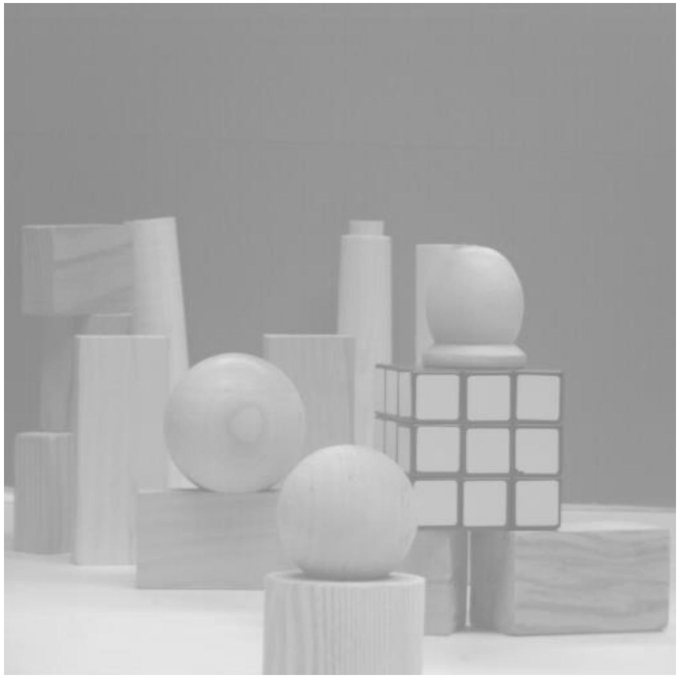
MATLAB function >imhist(x)

Why Histogram?

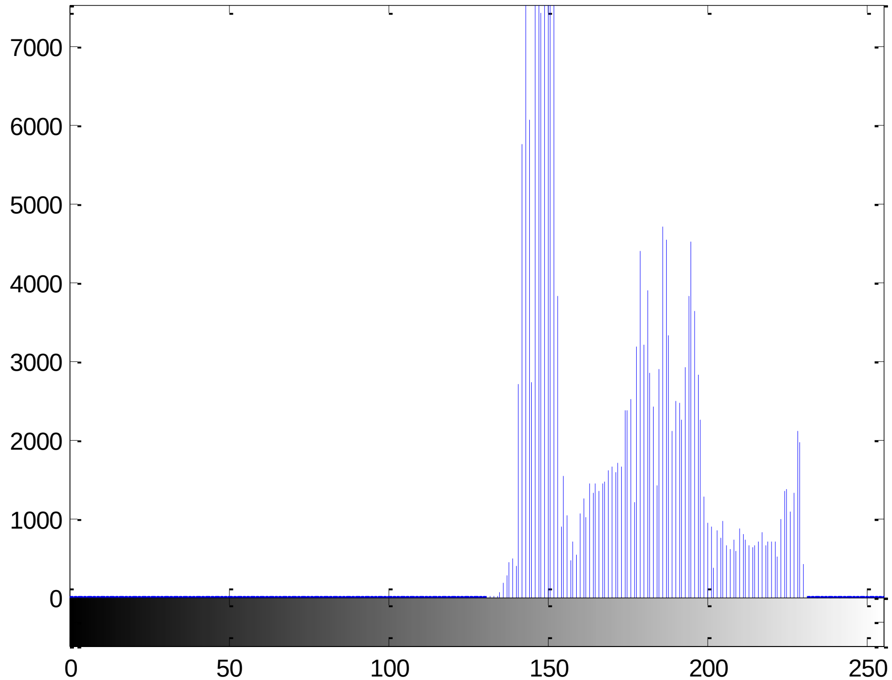


Histogram information reveals that image is under-exposed

Another Example

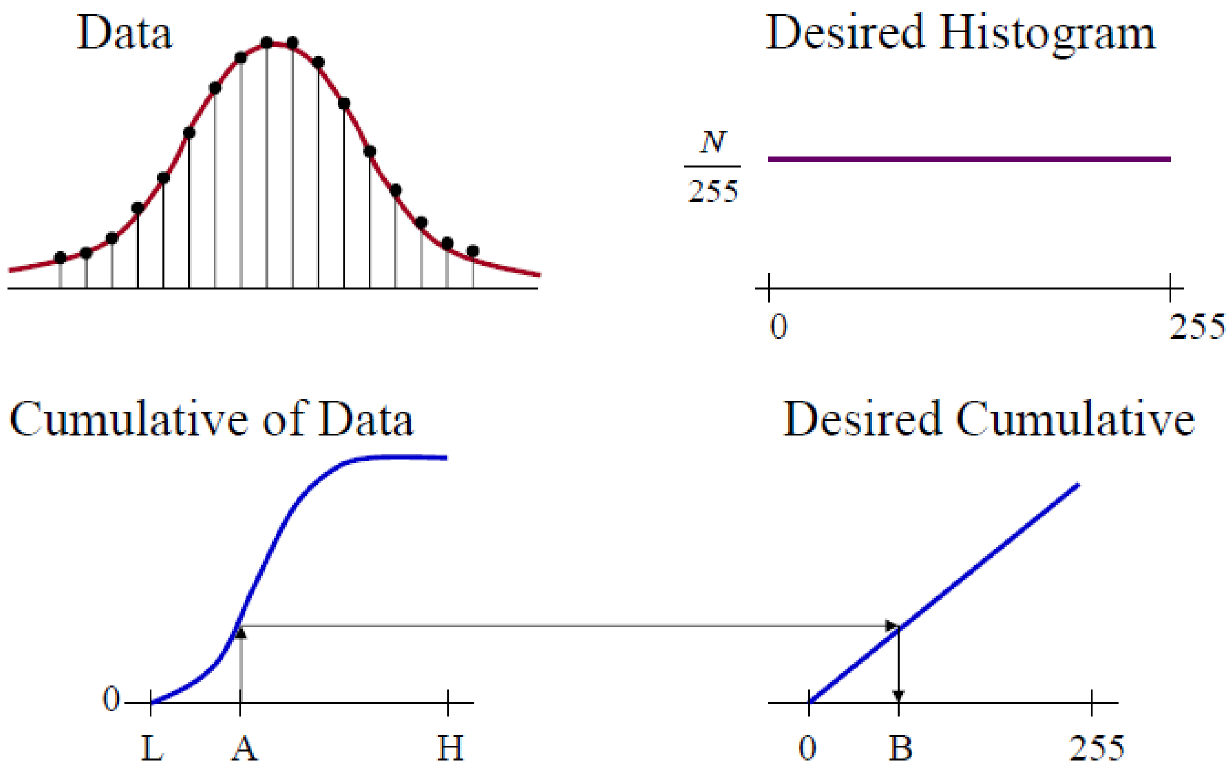


Over-exposed image



How to Adjust the Image?

- Histogram equalization



Histogram Equalization

$$\longrightarrow \boxed{y = L \cdot \sum_{t=0}^{x} h(t)} \longrightarrow \boxed{\text{Uniform Quantization}} \longrightarrow$$

Note: $\boxed{\sum_{t=0}^{L} h(t) = 1}$

$$s = \sum_{t=0}^{x} h(t)$$

cumulative probability function

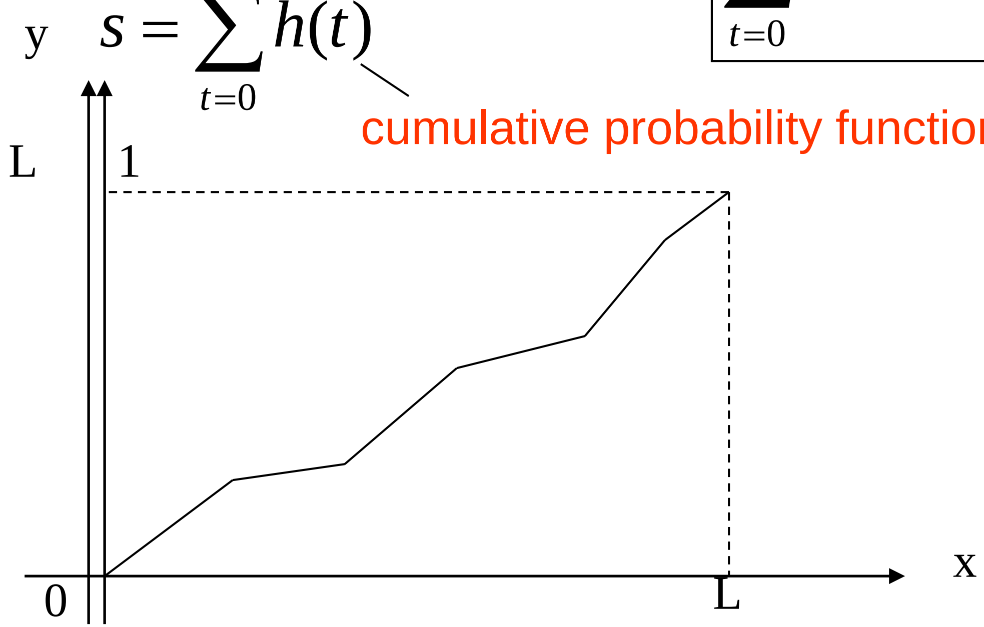


Image Example

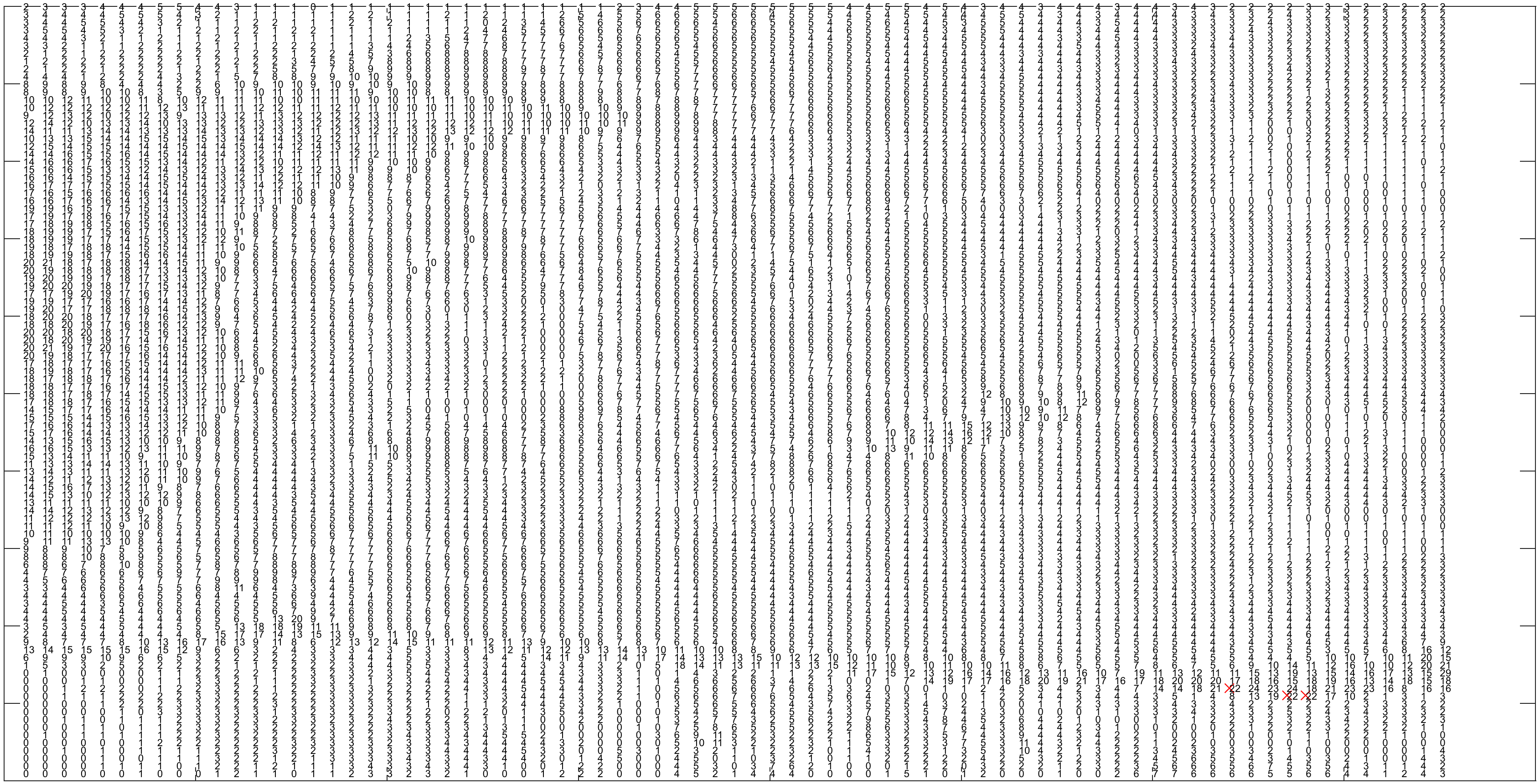
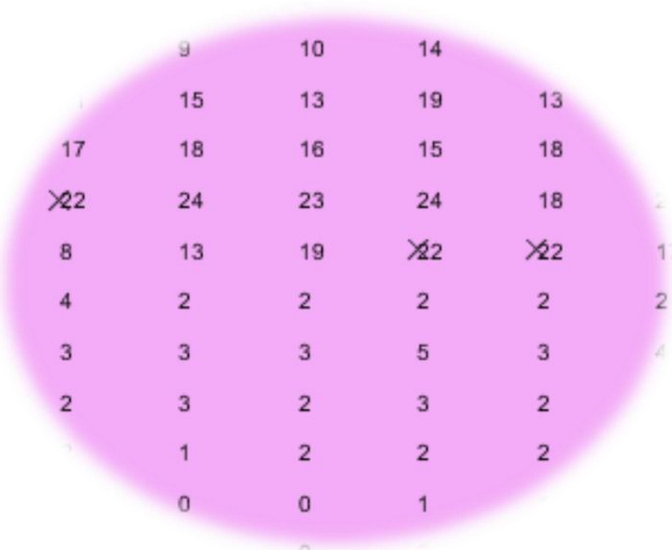


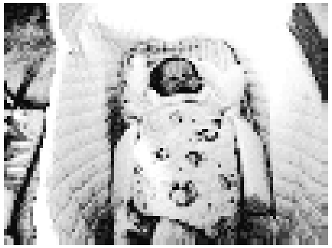
Image Example (cont'd)



I	#	f	F	J
0	386	0.0515	0.0515	6
1	607	0.0809	0.1324	16
2	812	0.1083	0.2407	30
3	1013	0.1351	0.3757	48
4	1130	0.1507	0.5264	67
5	1032	0.1376	0.6640	84
6	704	0.0939	0.7579	97
7	378	0.0504	0.8083	103
8	222	0.0296	0.8379	107
9	198	0.0264	0.8643	110
10	147	0.0196	0.8839	113
11	159	0.0212	0.9051	115
12	132	0.0176	0.9227	118
13	120	0.0160	0.9387	120
14	108	0.0144	0.9531	121
15	82	0.0109	0.9640	123
16	67	0.0089	0.9729	124
17	73	0.0097	0.9827	125
18	55	0.0073	0.9900	126
19	35	0.0047	0.9947	127
20	25	0.0033	0.9980	127
21	6	0.0008	0.9988	127
22	3	0.0004	0.9992	127
23	3	0.0004	0.9996	127
24	2	0.0003	0.9999	127
29	1	0.0001	1.0000	127

before
Bit-depth = 7

after
Bit-depth = 7



تمرین ۹

- یکنواخت‌سازی هیستوگرام
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

آزمون مستمر پنجم

- بهبود تصویر (۱.۵ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.

بخش ششم- تشخیص لبه

Computer Vision: the Grand Challenge

- Teach a computer to see is nontrivial at all
- Unlike binary images, grayscale/color images acquired by the sensor are often easy to understand by human being but difficult for a machine or a robot
- There are lots of interesting problems in the field of computer vision (image analysis)
  - Image segmentation, image understanding, face detection/recognition, object tracking …

How does Human Vision System work?

I see a human body

↓

I expect to see a human face

↓

I expect to see two eyes and a nose

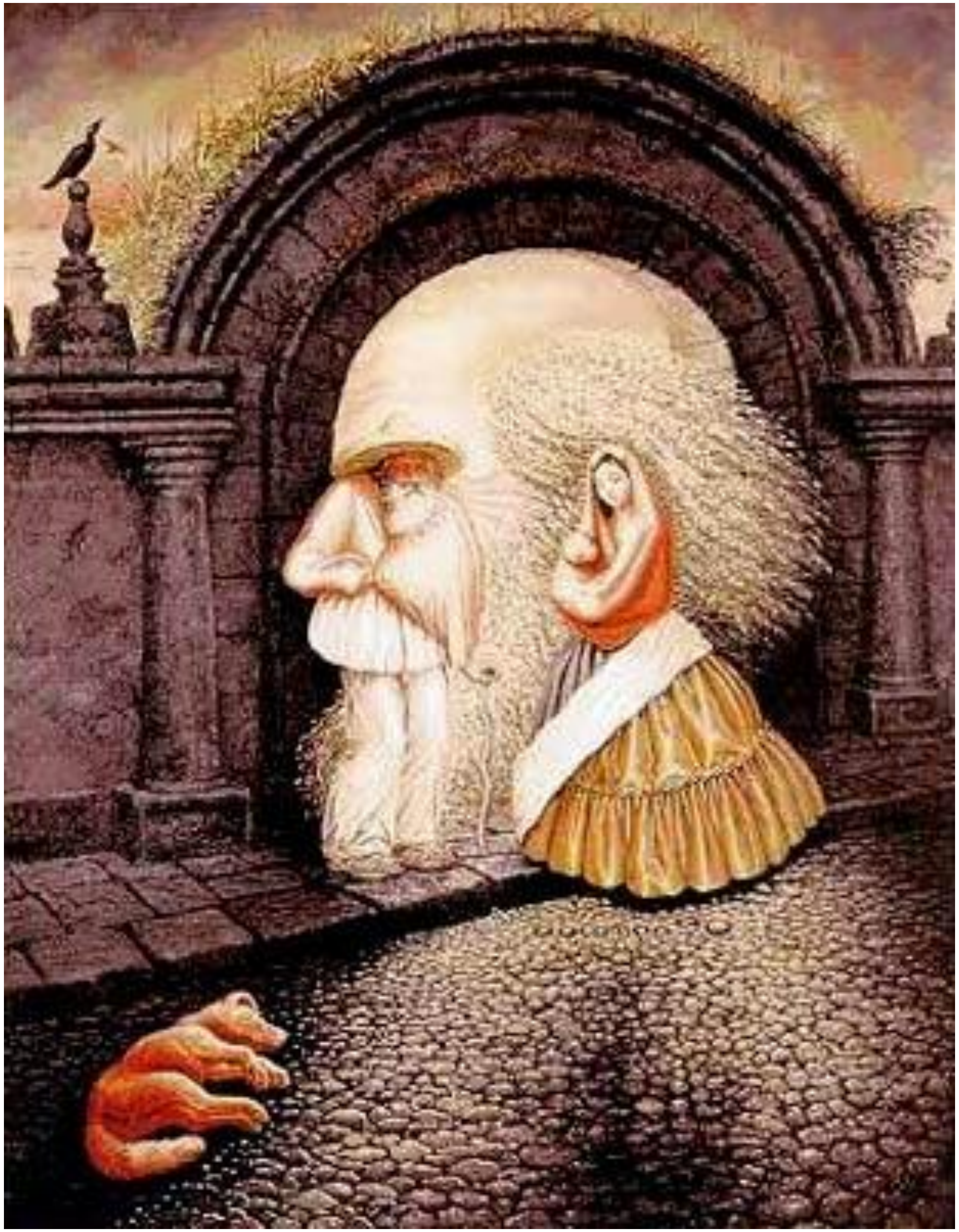
pixels

↓

components (such as edges, lines etc.)

↓

objects



Person A:

I see an old man with a fancy earring and a strange hand

Person B:

I see two people on the street and a dog lying beside

If you try really hard, you will be able to locate at least eight different faces from this image

Application: Face Detection

Edge Detection

- Why detect edge?

Edges characterize object boundaries and are useful features for **segmentation**, **registration (**align multiple scenes into a single integrated image) and **object identification** in scenes.

- What is edge (to human vision system)?

No rigorous definition exists

Intuitively, edge corresponds to singularities in the image
(i.e. where pixel value experiences abrupt change)

Gradient Operators

- Motivation: detect changes

change in the pixel value $\longrightarrow$ large gradient

image $x(m,n)$ $\longrightarrow$ Gradient operator $\longrightarrow$ $g(m,n)$ $\longrightarrow$ Thresholding $\longrightarrow$ edge map $I(m,n)$

$$I(m,n) = \begin{cases} 1 & |g(m,n)| > th \\ 0 & otherwise \end{cases}$$

MATLAB function: > help edge

Common Operators

- Gradient operator

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)}$$

Examples: 1. Roberts operator

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

g_1 g_2

Common Operators (cont'd)

2. Prewitt operator

3. Sobel operator

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

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

Examples

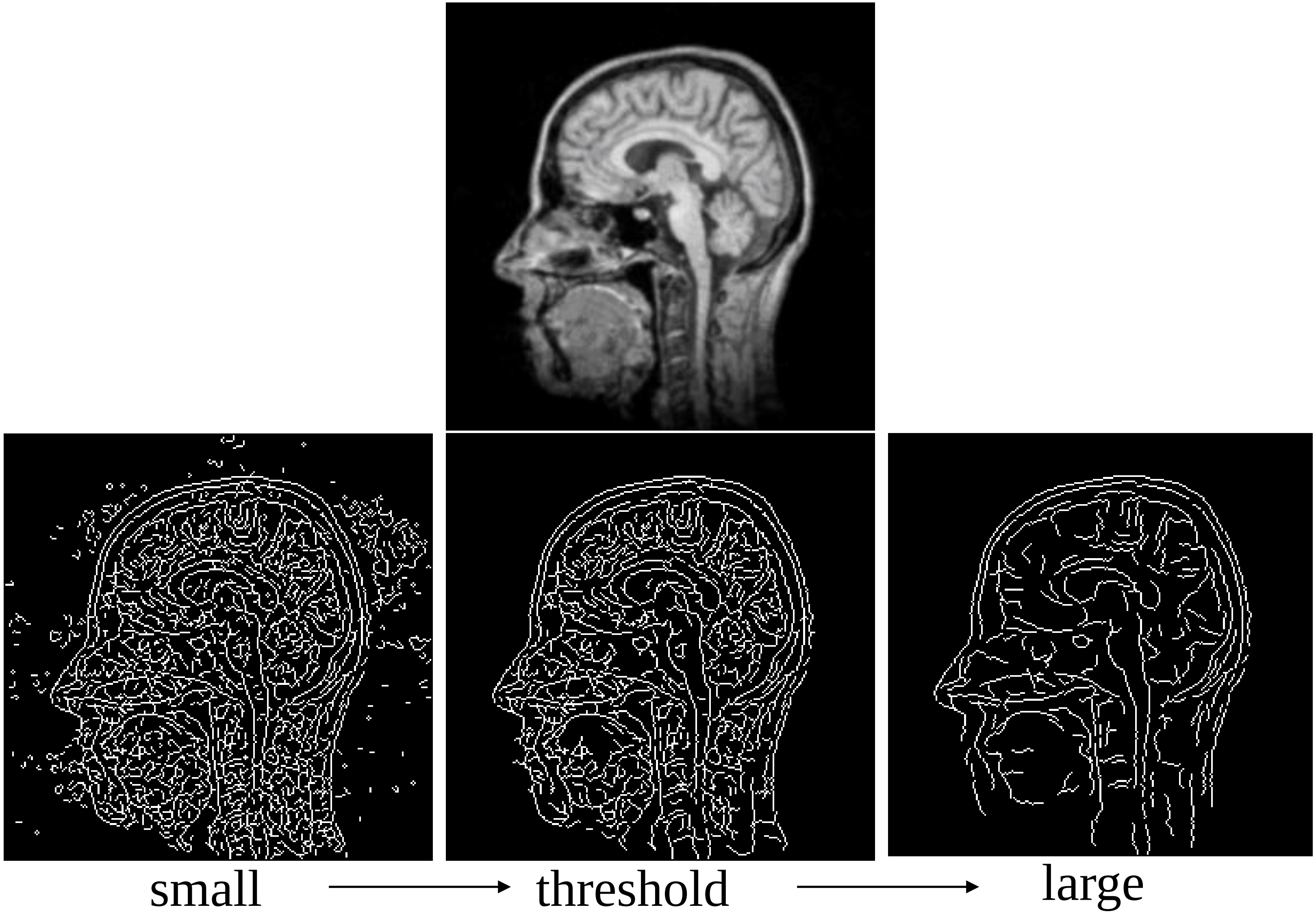
original image

horizontal edge

vertical edge

Prewitt operator (*th=48*)

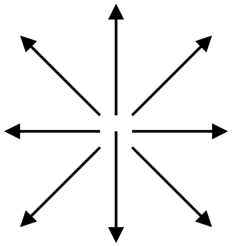
Effect of Thresholding Parameters



Compass Operators

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

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



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

$$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$g(m,n) = \max_k \{|\, g_k(m,n)\,|\}$$

Example for Compass Operators

```
I = imread('Mehdi.jpg');
I = flipud(double(mean(I,3)));
imshow(I,[0 255])

g = {[1 1 0 ; 1 0 -1 ; 0 -1 -1],...
    [1 1 1 ; 0 0 0 ; -1 -1 -1],...
    [0 1 1 ; -1 0 1 ; -1 -1 0],...
    [1 1 1 ; 0 0 0 ; -1 -1 -1]',...
    fliplr([1 1 1 ; 0 0 0 ; -1 -1 -1]'),...
    flipud([1 1 1 ; 0 0 0 ; -1 -1 -1]),...
    [0 -1 -1 ; 1 0 -1 ; 1 1 0], ...
    [-1 -1 0 ; -1 0 1 ; 0 1 1]};

J = zeros(size(I));
for i=1:length(g)
    J = max(abs(imfilter(I,g{i})),J);
end
th = 40;
J(J<th) = 0;
J(J>=th) = 1;
figure
imshow(J,[0 1])
```

Laplacian Operators

- Gradient operator: first-order derivative

sensitive to abrupt change, but not slow change

second-order derivative:

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

(Laplacian operator)

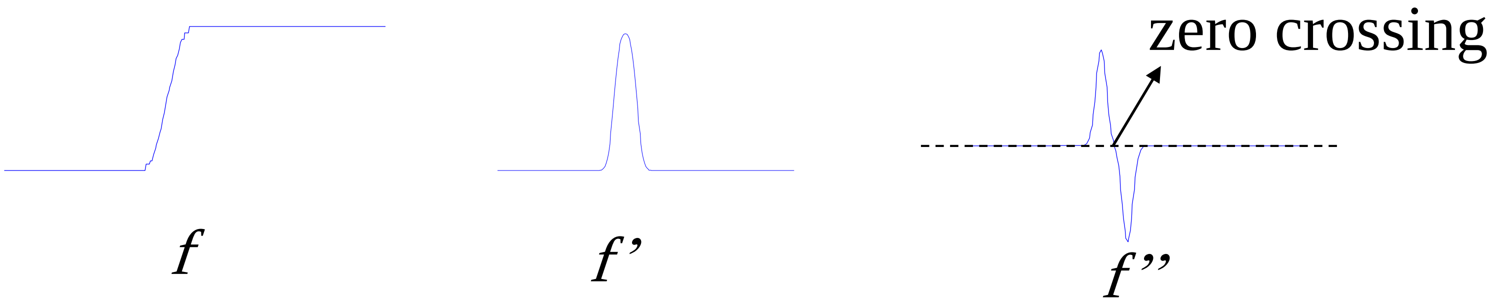
$$\frac{\partial^2 f}{\partial x^2} = 0 \longrightarrow$$ local extreme in f'

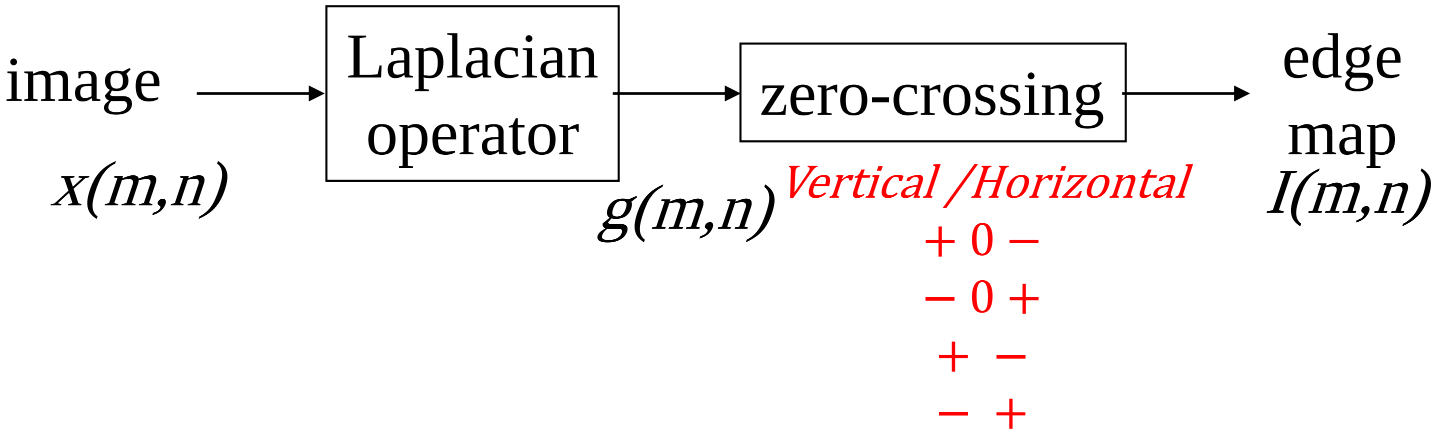
- Discrete Laplacian operator

$$\frac{1}{1+a}\begin{bmatrix} a & 1-a & a \\ 1-a & -4 & 1-a \\ a & 1-a & a \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

a=0 a=0.5

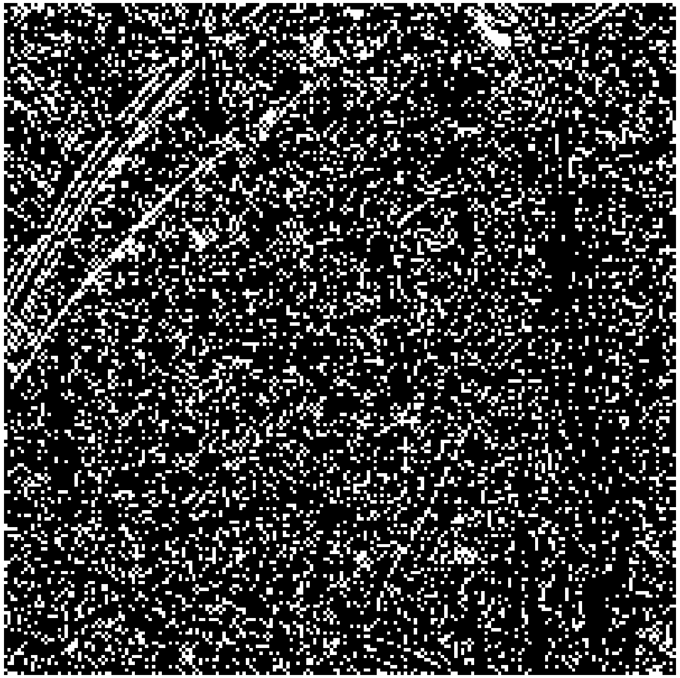
Zero Crossings





Examples

original image



zero-crossings

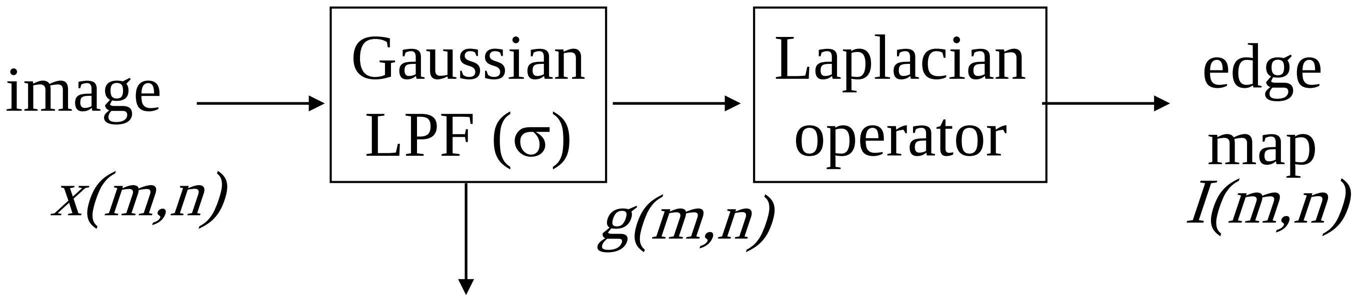
Question: why is it so sensitive to noise (many false alarms)?
Answer: a sign flip from 0.01 to -0.01 is treated the same as from 100 to -100

Ideas to Improve Robustness

- Linear filtering
  - Use a Gaussian filter to smooth out noise component → Laplacian of Gaussian
- Spatially-adaptive (Nonlinear) processing
  - Apply different detection strategies to smooth areas (low-variance) and non-smooth areas (high-variance) → Robust Laplacian edge detector
- Return single response to edges (not multiple edge pixels)
  - Hysteresis thresholding → Canny's edge detector

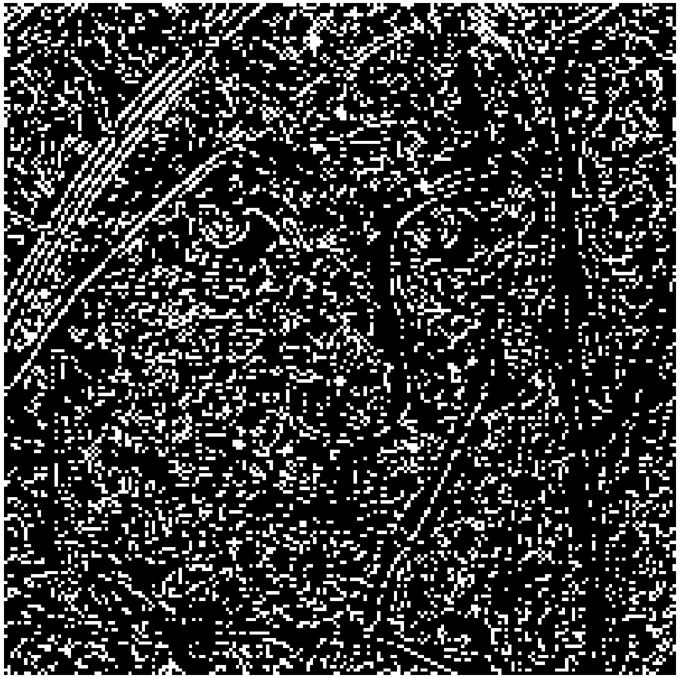
Laplacian of Gaussian

- Generalized Laplacian operator



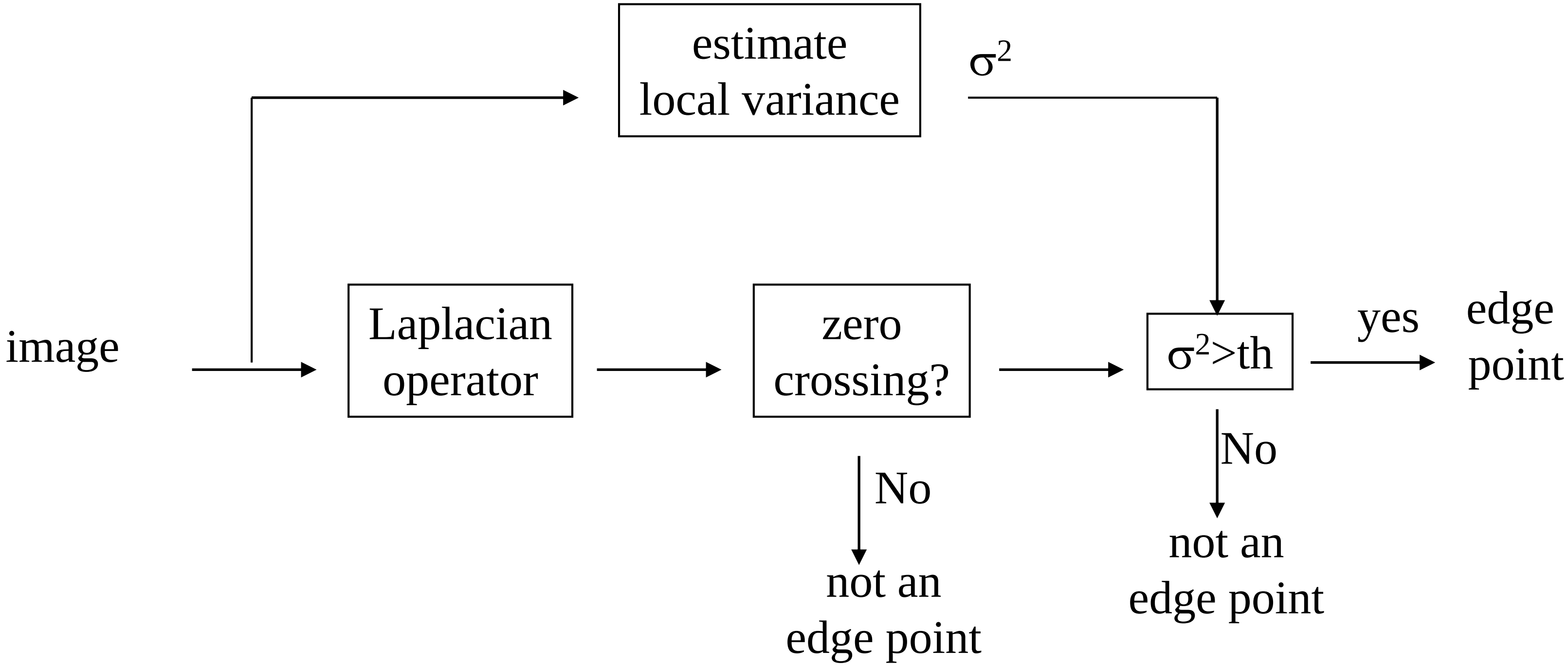
Pre-filtering: attenuate the noise sensitivity of the Laplacian

Examples



Better than Laplacian alone but still sensitive due to zero crossing

Robust Laplacian-based Edge Detector



Examples

More robust but return multiple edge pixels (poor localization)

Canny Edge Detector

- Low error rate of detection
  - Well match human perception results
- Good localization of edges
  - The distance between actual edges in an image and the edges found by a computational algorithm should be minimized
- Single response
  - The algorithm should not return multiple edges pixels when only a single one exists

Flow-chart of Canny Edge Detector
(*J. Canny'1986*)

Original image

↓

Smoothing by Gaussian convolution

↓

Differential operators along x and y axis

↓

Non-maximum suppression
finds peaks in the image gradient

↓

Hysteresis thresholding locates edge strings

↓

Edge map

Flow-chart of Canny Edge Detector (cont'd)

- The hysteresis thresholding uses a hysteresis loop to provide a more connected result. Any pixel above the upper threshold is turned white. The surround pixels are then searched recursively. If the values are greater than the lower threshold they are also turned white. The result is that there are many fewer specks of white in the resulting image.

تمرین ۱۰

- تشخیص لبه
  - به غیر از تابع imread، استفاده از سایر توابع پردازش تصویری MATLAB برای پیاده‌سازی‌ها مجاز نیست.
  - صرفاً برای مقایسه با کد نوشته‌شده شما و در صورتی که در صورت مسئله خواسته‌شده باشد، می‌توانید از توابع آماده استفاده کنید.

آزمون مستمر ششم

- تشخیص لبه (۱.۵ نمره)
  - شما مجاز به استفاده از ماشین حساب (و نه موبایل یا سایر ابزارها به عنوان ماشین حساب) هستید.