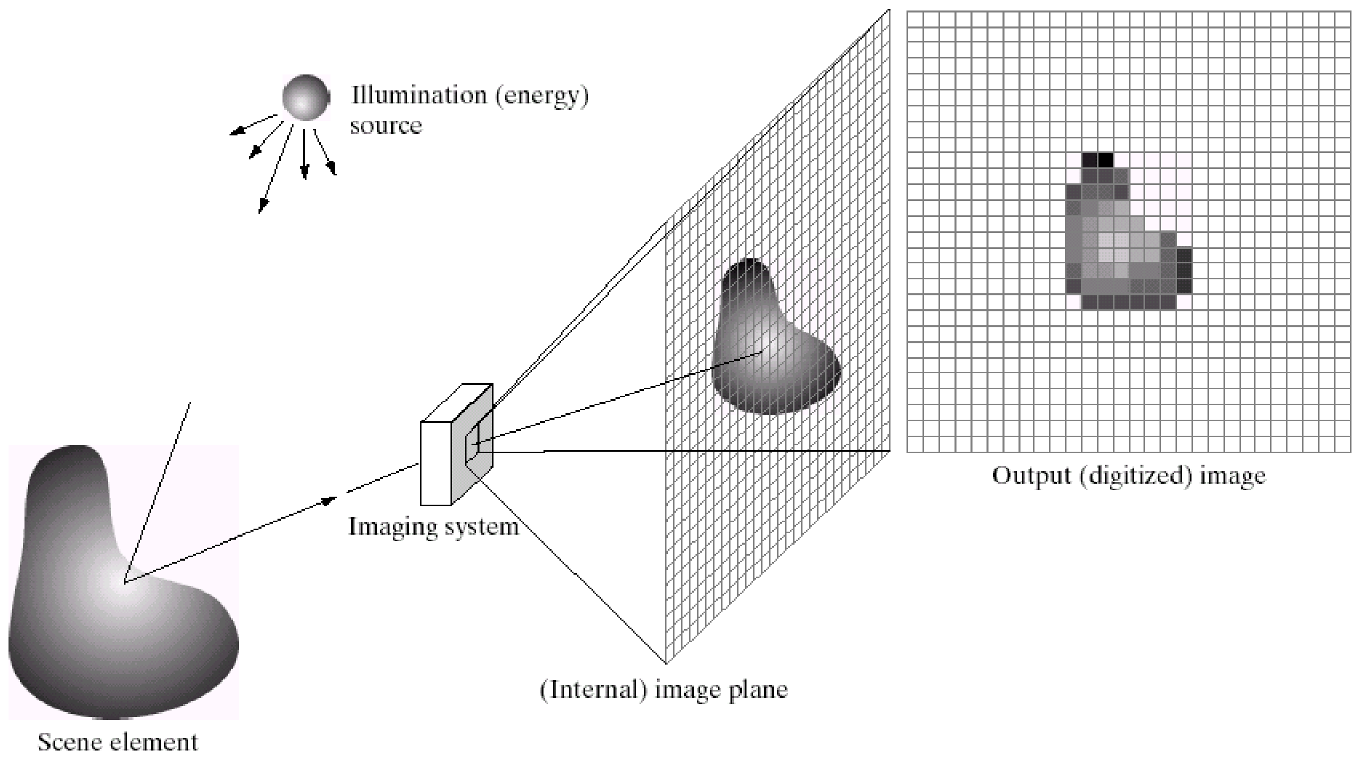
BEC007-Digital Image Processing

UNIT I
DIGITAL IMAGE FUNDAMENTAL

- Elements of digital image processing systems
- Elements of Visual perception
- Image sampling and quantization
- Matrix and Singular Value representation of discrete images.

What is a Digital Image?

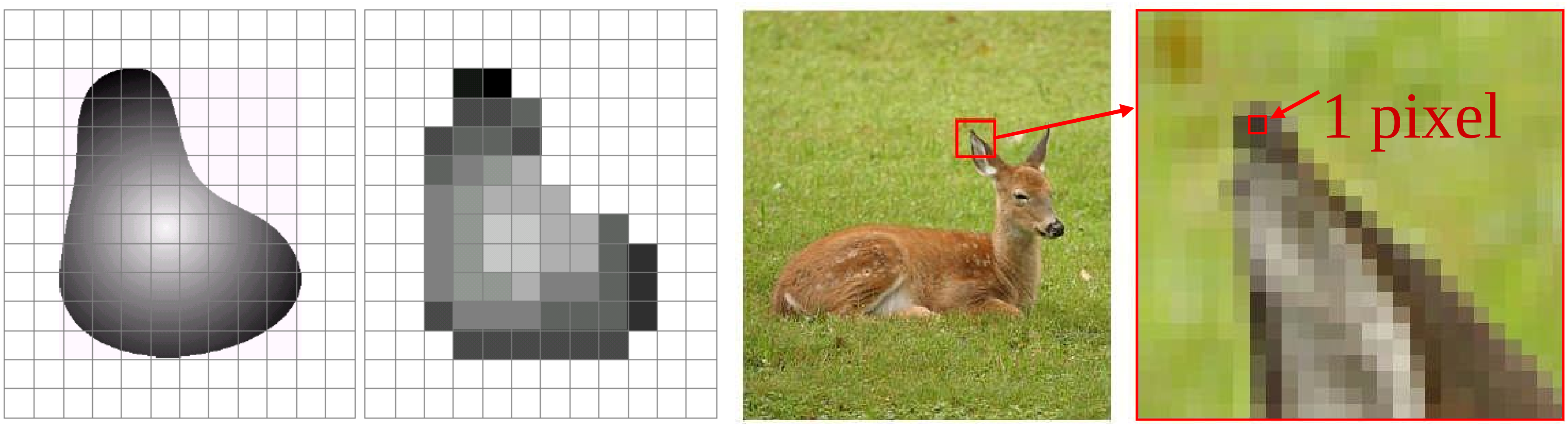
•A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels



Cont..

•Pixel values typically represent gray levels, colours, heights, opacities etc

•**Remember** *digitization* implies that a digital image is an *approximation* of a real scene



Cont..

Common image formats include:

1 sample per point (B&W or Grayscale)

3 samples per point (Red, Green, and Blue)

4 samples per point (Red, Green, Blue, and "Alpha", a.k.a. Opacity)

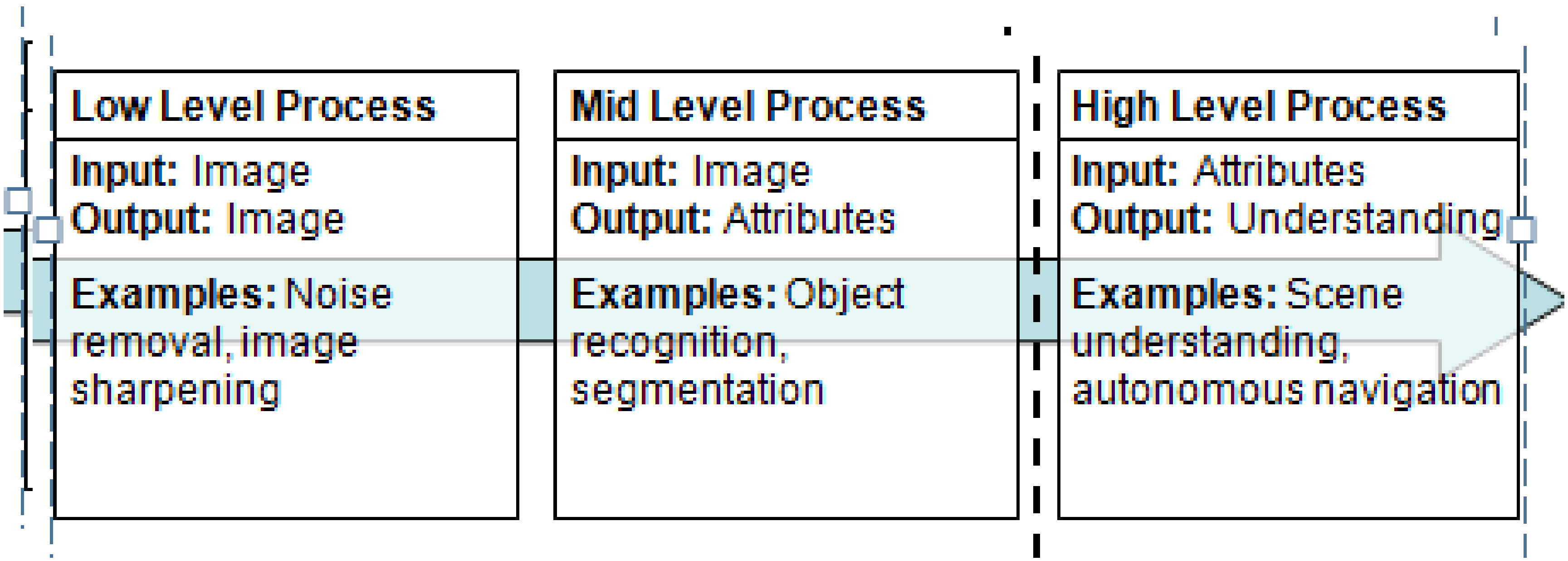
For most of this course we will focus on grey-scale images

What is Digital Image Processing?

- Digital image processing focuses on two major tasks
  - Improvement of pictorial information for human interpretation
  - Processing of image data for storage, transmission and representation for autonomous machine perception
- Some argument about where image processing ends and fields such as image analysis and computer vision start

cont…

The continuum from image processing to computer vision can be broken up into low-, mid- and high-level processes

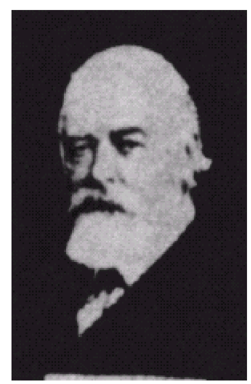


History of Digital Image Processing

- **Early 1920s:** One of the first applications of digital imaging was in the news-paper industry
  - The Bartlane cable picture transmission service
  - Images were transferred by submarine cable between London and New York
  - Pictures were coded for cable transfer and reconstructed at the receiving end on a telegraph printer

History of DIP (cont…)

- **Mid to late 1920s:** Improvements to the Bartlane system resulted in higher quality images
  - New reproduction processes based on photographic techniques
  - Increased number of tones in reproduced images

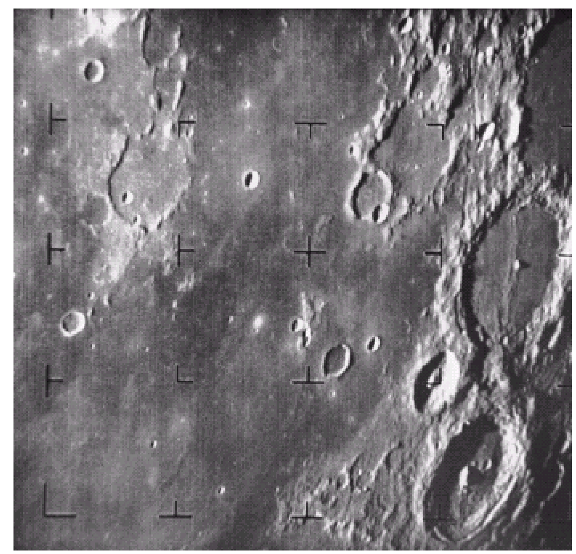


Improved digital image

Early 15 tone digital image

History of DIP (cont…)

- **1960s:** Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing
  - **1964:** Computers used to improve the quality of images of the moon taken by the *Ranger 7* probe
  - Such techniques were used in other space missions including the Apollo landings



A picture of the moon taken by the Ranger 7 probe minutes before landing

History of DIP (cont…)

- **1970s:** Digital image processing begins to be used in medical applications
  - **1979:** Sir Godfrey N. Hounsfield & Prof. Allan M. Cormack share the Nobel Prize in medicine for the invention of tomography, the technology behind Computerised Axial Tomography (CAT) scans

Typical head slice CAT image

History of DIP (cont…)

- **1980s - Today:** The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas
  - Image enhancement/restoration
  - Artistic effects
  - Medical visualisation
  - Industrial inspection
  - Law enforcement
  - Human computer interfaces

History of DIP (cont…)

•**1980s - Today:** The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas

– Image enhancement/restoration
– Artistic effects
– Medical visualisation
– Industrial inspection
– Law enforcement
– Human computer interfaces

Image Processing Fields

- **Computer Graphics:** The creation of images
- **Image Processing:** Enhancement or other manipulation of the image
- **Computer Vision:** Analysis of the image content

Image Processing Fields

Input / Output	**Image**	**Description**
Image	Image Processing	Computer Vision
Description	Computer Graphics	AI

Sometimes, Image Processing is defined as “a discipline in which both the input and output of a process are images

But, according to this classification, trivial tasks of computing the average intensity of an image would not be considered an image processing operation

Computerized Processes Types

- **Low-Level Processes:**
  - **Input** and **output** are images
  - **Tasks:** Primitive operations, such as, image processing to reduce noise, contrast enhancement and image sharpening

Computerized Processes Types

- **Mid-Level Processes:**
  - **Inputs**, generally, are images. **Outputs** are attributes extracted from those images (edges, contours, identity of individual objects)
  - **Tasks:**
    - Segmentation (partitioning an image into regions or objects)
    - Description of those objects to reduce them to a form suitable for computer processing
    - Classifications (recognition) of objects

Computerized Processes Types

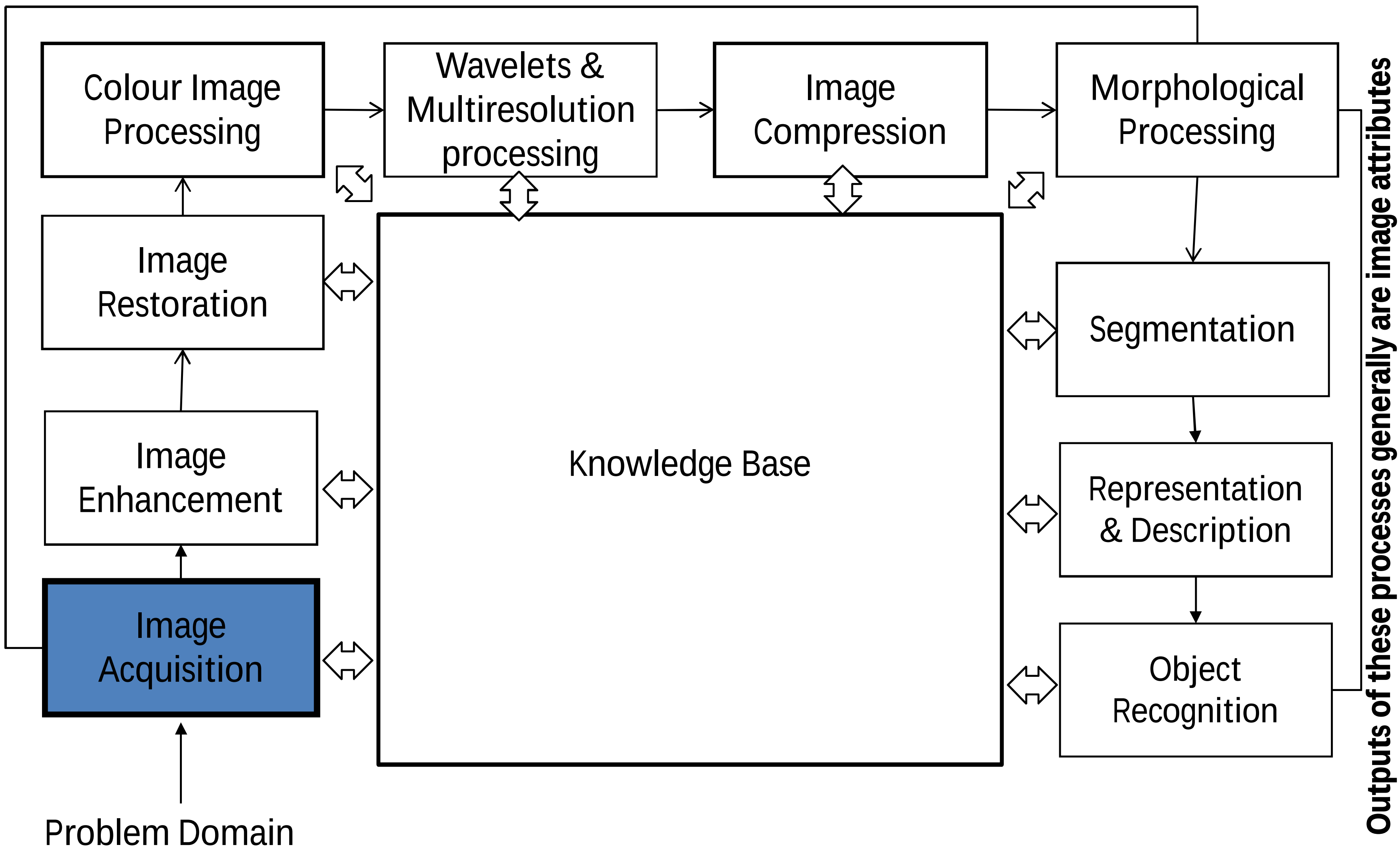
- **High-Level Processes:**
  - Image analysis and computer vision

Digital Image Definition

- An image can be defined as a two-dimensional function f(x,y)
- x,y: Spatial coordinate
- F: the amplitude of any pair of coordinate x,y, which is called the intensity or gray level of the image at that point.
- X,y and f, are all finite and discrete quantities.

Fundamental Steps in Digital Image Processing:

Outputs of these processes generally are images



Fundamental Steps in DIP: (Description)

Step 1: Image Acquisition

The image is captured by a sensor (eg. Camera), and digitized if the output of the camera or sensor is not already in digital form, using analogue-to-digital convertor

Fundamental Steps in DIP: (Description)

Step 2: Image Enhancement

The process of manipulating an image so that the result is more suitable than the original for specific applications.

The idea behind enhancement techniques is to bring out details that are hidden, or simple to highlight certain features of interest in an image.

Fundamental Steps in DIP: (Description)

Step 3: Image Restoration

- Improving the appearance of an image

- Tend to be mathematical or probabilistic models. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result.

Fundamental Steps in DIP: (Description)

Step 4: Colour Image Processing

Use the colour of the image to extract features of interest in an image

Fundamental Steps in DIP: (Description)

Step 5: Wavelets

Are the foundation of representing images in various degrees of resolution. It is used for image data compression.

Fundamental Steps in DIP: (Description)

Step 6: Compression

Techniques for reducing the storage required to save an image or the bandwidth required to transmit it.

Fundamental Steps in DIP: (Description)

Step 7: Morphological Processing

Tools for extracting image components that are useful in the representation and description of shape.

In this step, there would be a transition from processes that output images, to processes that output image attributes.

Fundamental Steps in DIP: (Description)

Step 8: Image Segmentation

Segmentation procedures partition an image into its constituent parts or objects.

Important Tip: The more accurate the segmentation, the more likely recognition is to succeed.

Fundamental Steps in DIP: (Description)

Step 9: Representation and Description

- **Representation:** Make a decision whether the data should be represented as a boundary or as a complete region. It is almost always follows the output of a segmentation stage.
  - **Boundary Representation:** Focus on external shape characteristics, such as corners and inflections (انحناءات)
  - **Region Representation:** Focus on internal properties, such as texture or skeleton (هيكلية) shape

Fundamental Steps in DIP: (Description)

Step 9: Representation and Description

- Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing (mainly recognition)

- **Description:** also called, ***feature selection***, deals with extracting attributes that result in some information of interest.

Fundamental Steps in DIP: (Description)

Step 9: Recognition and Interpretation

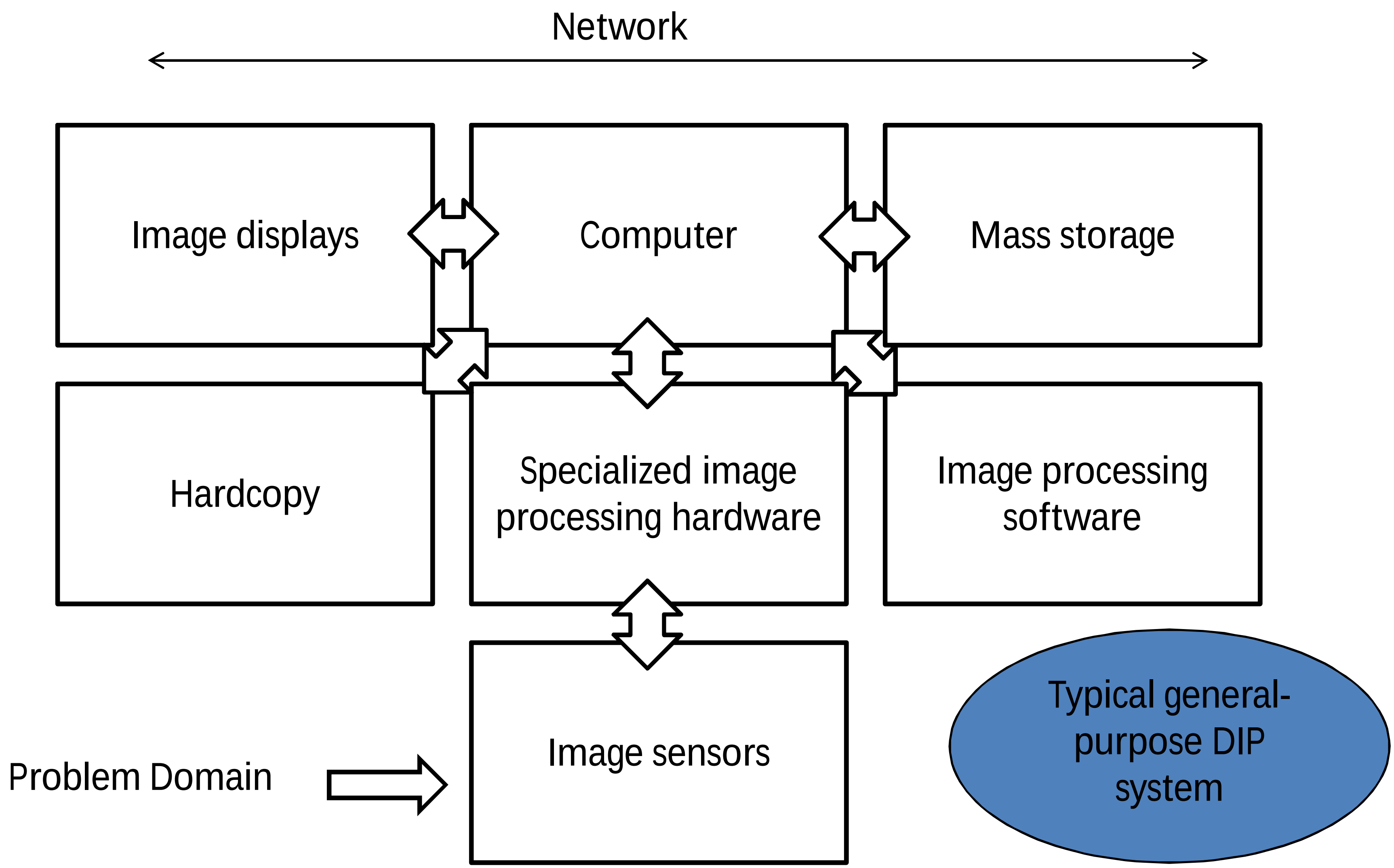
Recognition: the process that assigns label to an object based on the information provided by its description.

Fundamental Steps in DIP: (Description)

Step 10: Knowledge Base

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database.

Components of an Image Processing System



Components of an Image Processing System

1. **Image Sensors**

   Two elements are required to acquire digital images. The first is the physical device that is sensitive to the energy radiated by the object we wish to image (*Sensor*). The second, called a *digitizer*, is a device for converting the output of the physical sensing device into digital form.

Components of an Image Processing System

2. Specialized Image Processing Hardware

Usually consists of the digitizer, mentioned before, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images.

This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs that the typical main computer cannot handle.

Components of an Image Processing System

3. **Computer**

   The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance.

Components of an Image Processing System

4. Image Processing Software

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules.

Components of an Image Processing System

5. Mass Storage Capability

Mass storage capability is a must in a image processing applications. And image of sized 1024 * 1024 pixels requires one megabyte of storage space if the image is not compressed.

Digital storage for image processing applications falls into three principal categories:

1. Short-term storage for use during processing.

2. on line storage for relatively fast recall

3. Archival storage, characterized by infrequent access

Components of an Image Processing System

5. Mass Storage Capability

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers, that store one or more images and can be accessed rapidly.

The on-line storage method, allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). On-line storage generally takes the form of magnetic disks and optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data.

Components of an Image Processing System

6. Image Displays

The displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of the image and graphics display cards that are an integral part of a computer system.

Components of an Image Processing System

7. Hardcopy devices

Used for recording images, include laser printers, film cameras, heat-sensitive devices, inkjet units and digital units, such as optical and CD-Rom disks.

8. Networking

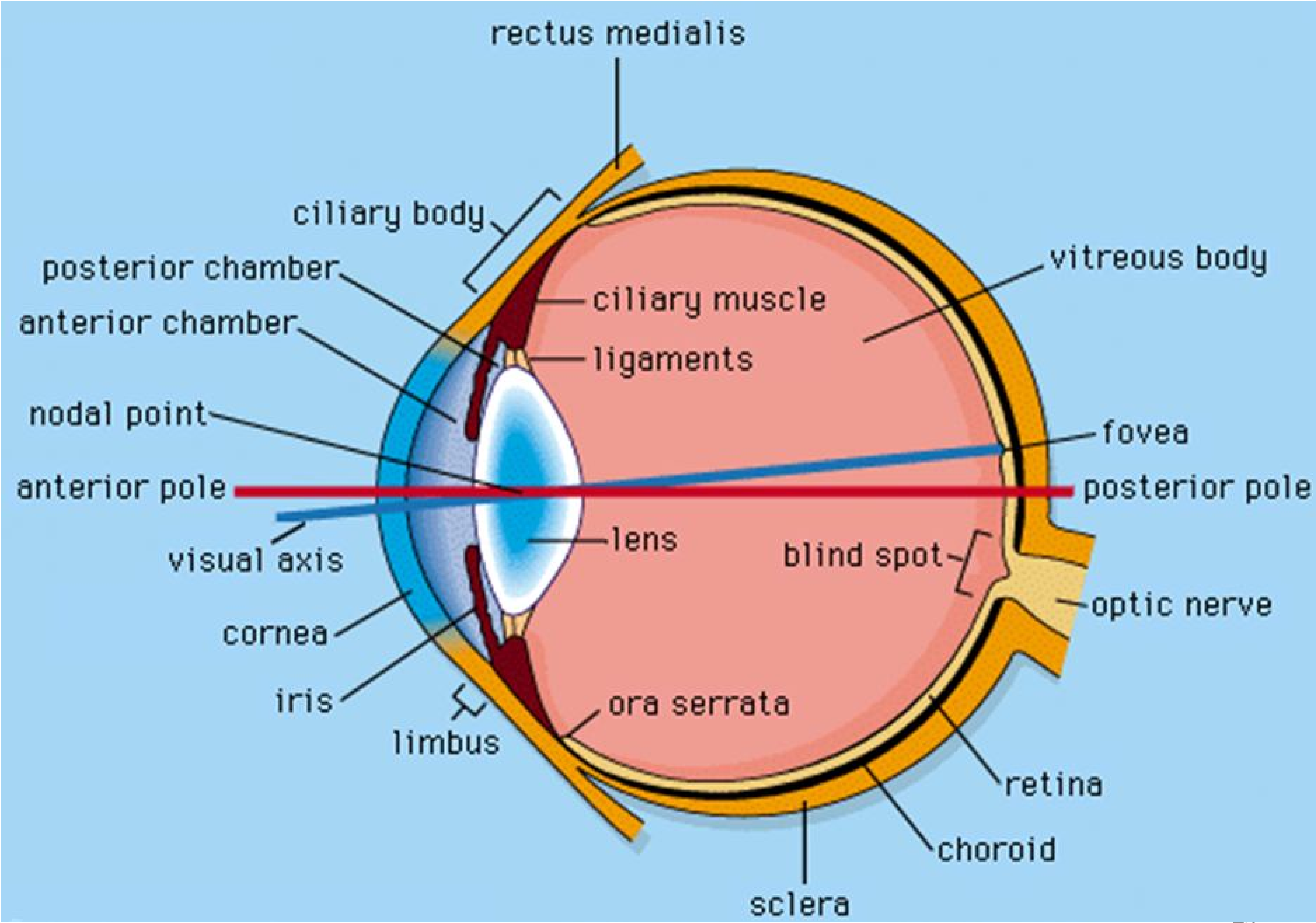
Is almost a default function in any computer system, in use today. Because of the large amount of data inherent in image processing applications the key consideration in image transmission is bandwidth.

In dedicated networks, this typically is not a problem, but communications with remote sites via the internet are not always as efficient.

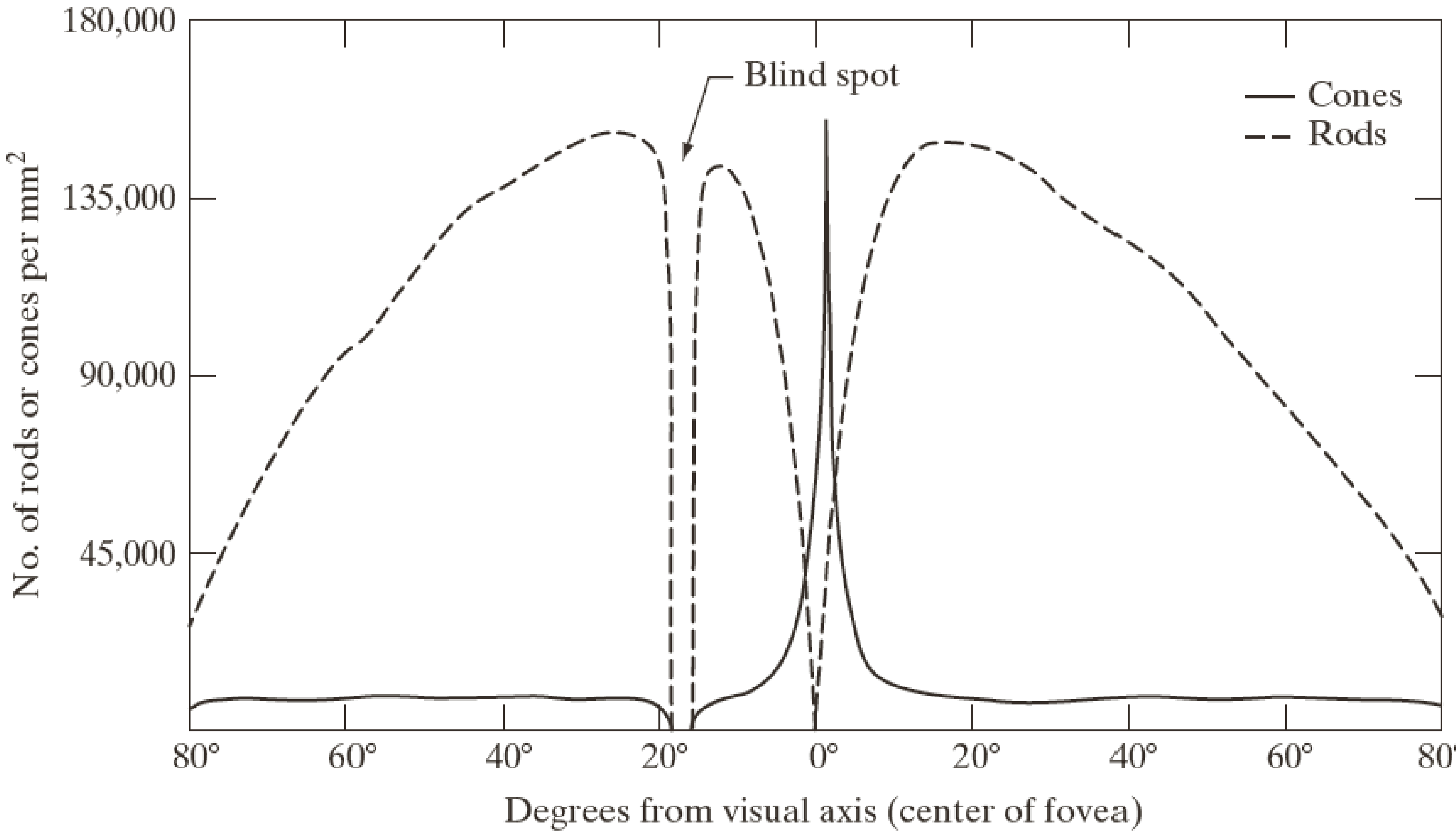
Elements of Visual perception

Structure of the human eye

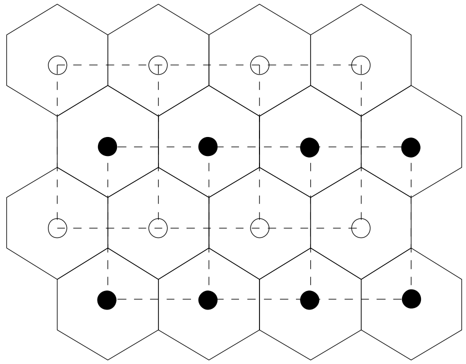
- The cornea and sclera outer cover
- The choroid
  - Ciliary body
  - Iris diaphragm
  - Lens
- The retina
  - Cones vision (photopic/bright-light vision): centered at fovea, highly sensitive to color
  - Rods (scotopic/dim-light vision): general view
  - Blind spot

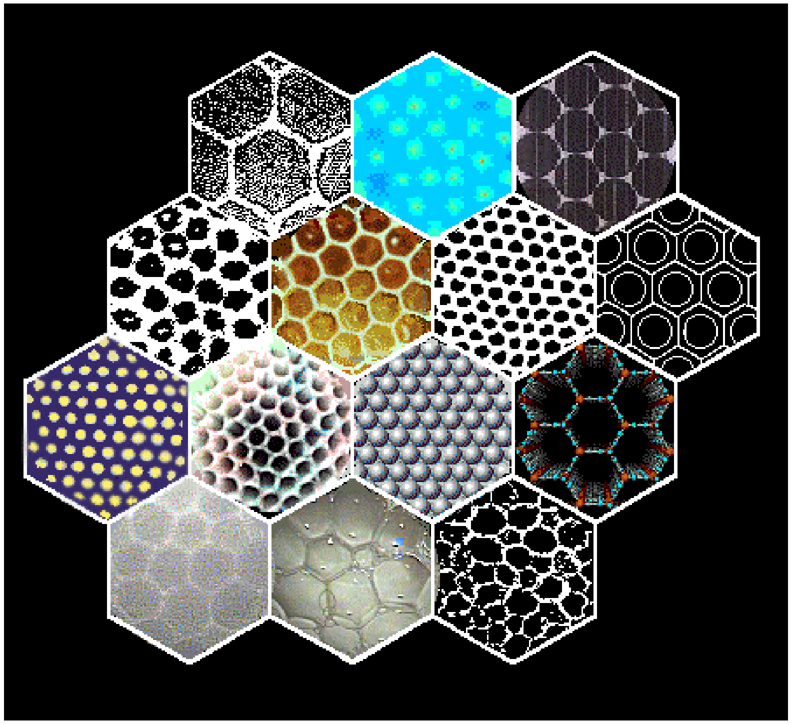

rectus medialis
ciliary body
posterior chamber
anterior chamber
nodal point
anterior pole
visual axis
cornea
iris
limbus
ciliary muscle
ligaments
lens
ora serrata
vitreous body
fovea
posterior pole
blind spot
optic nerve
retina
choroid
sclera

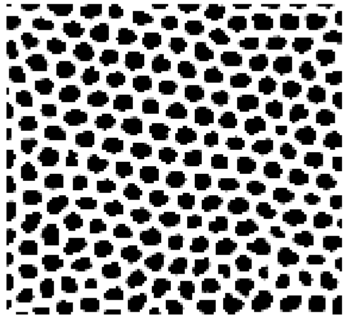
Cones vs. Rods



Hexagonal pixel



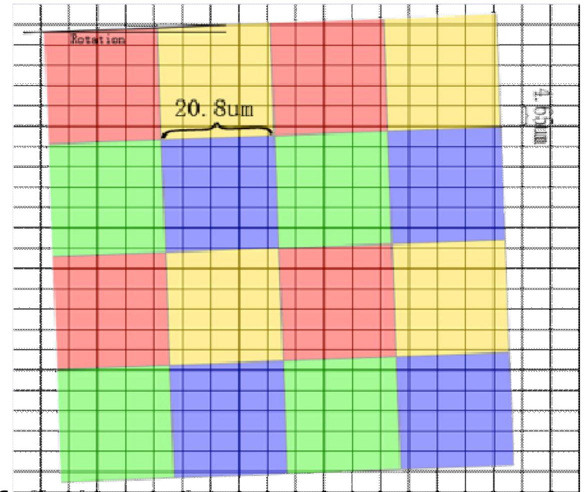


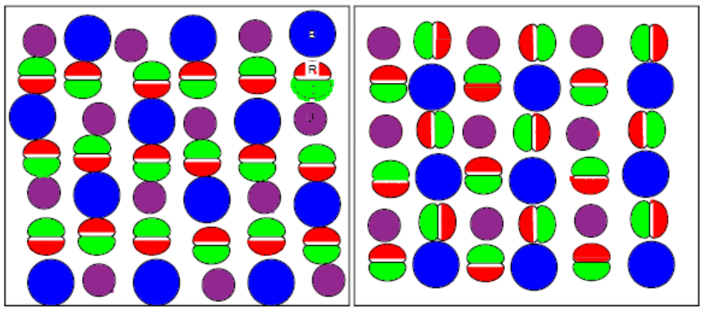


Cone distribution on the fovea (200,000 cones/mm^2)

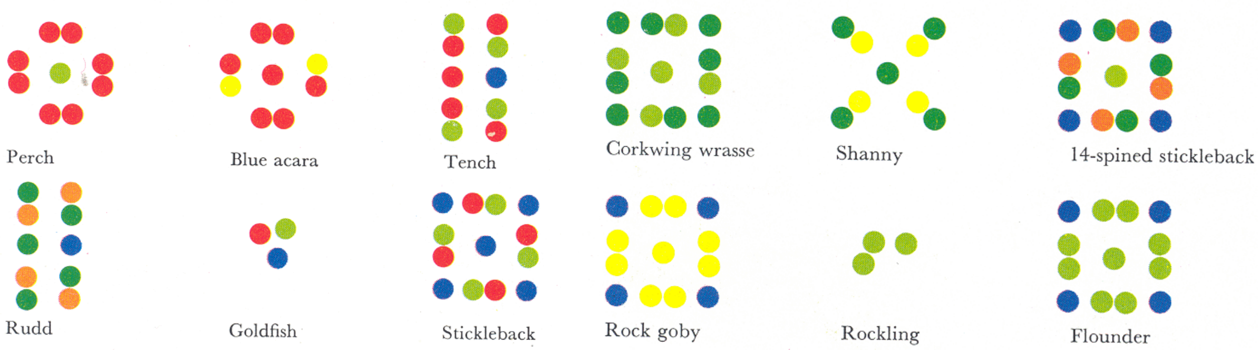
- Models human visual system more precisely
- The distance between a given pixel and its immediate neighbors is the same
- Hexagonal sampling requires 13% fewer samples than rectangular sampling
- ANN can be trained with less errors

More on the cone mosaic

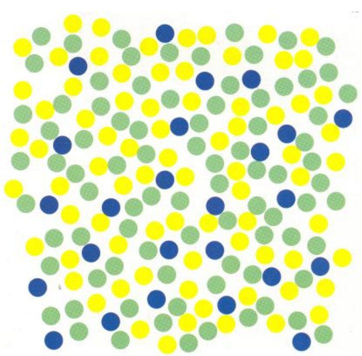




The cone mosaic of fish retina



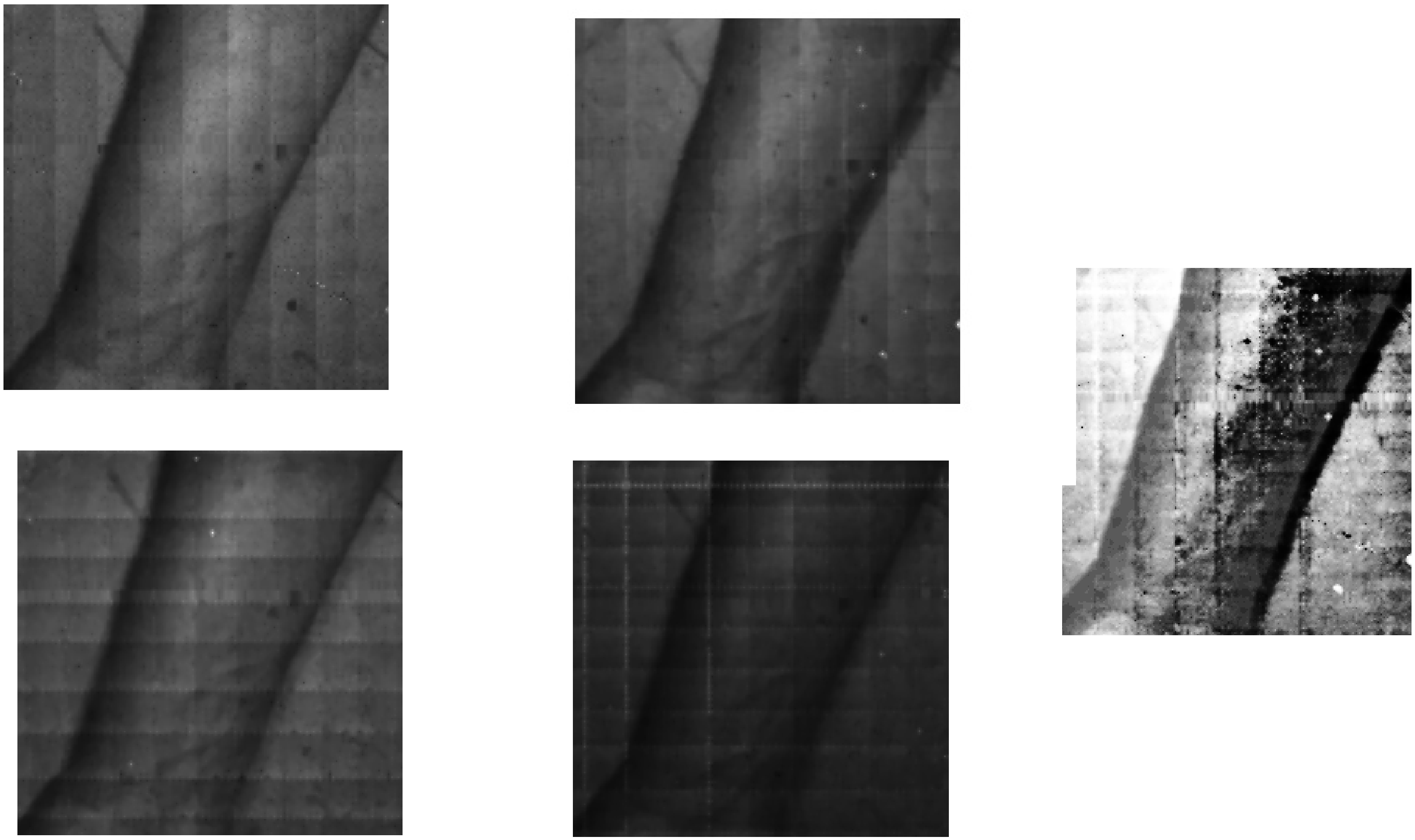
Lythgoe, *Ecology of Vision* (1979)



Human retina mosaic
-Irregularity reduces visual acuity for high-frequency signals
-Introduce random noise

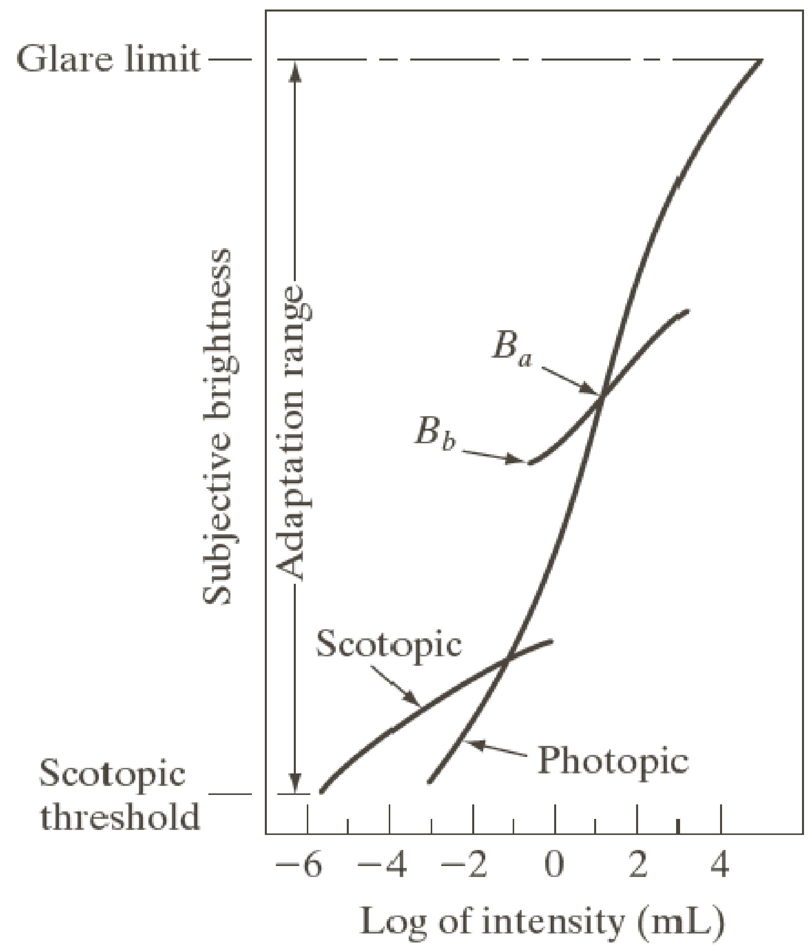
The mosaic array of most vertebrates is regular

A mosaicked multispectral camera



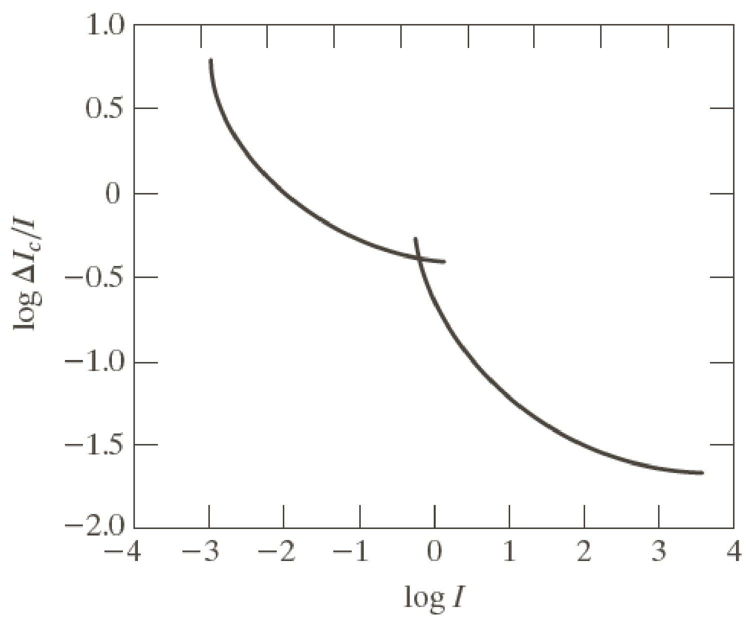
Brightness adaptation

- Dynamic range of human visual system
  - $10^{-6} \sim 10^{4}$
- Cannot accomplish this range **simultaneously**
- The current sensitivity level of the visual system is called the **brightness adaptation** level



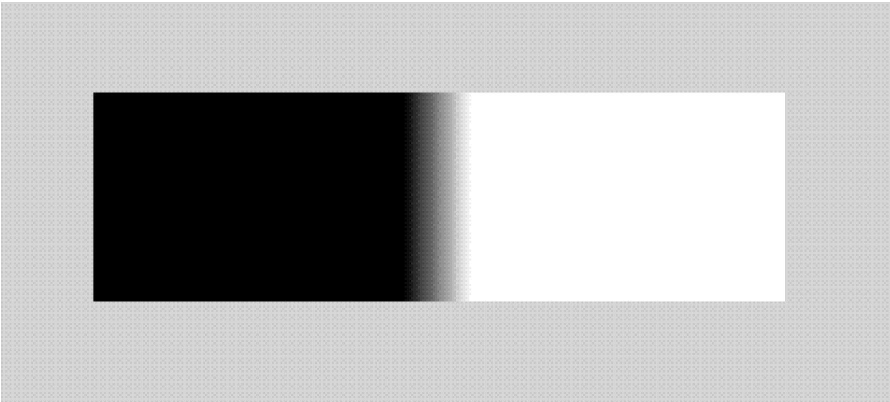
Brightness discrimination

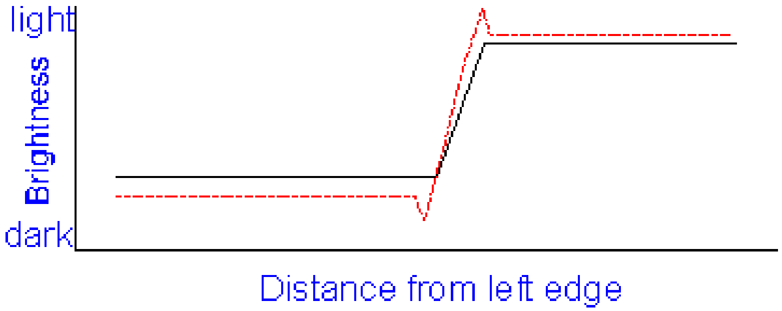
- Weber ratio (the experiment) $\Delta I_c/I$
  - I: the background illumination
  - ΔI_c : the increment of illumination
  - Small Weber ratio indicates good discrimination
  - Larger Weber ratio indicates poor discrimination



Psychovisual effects

- The perceived brightness is not a simple function of intensity
  - Mach band pattern
  - Simultaneous contrast





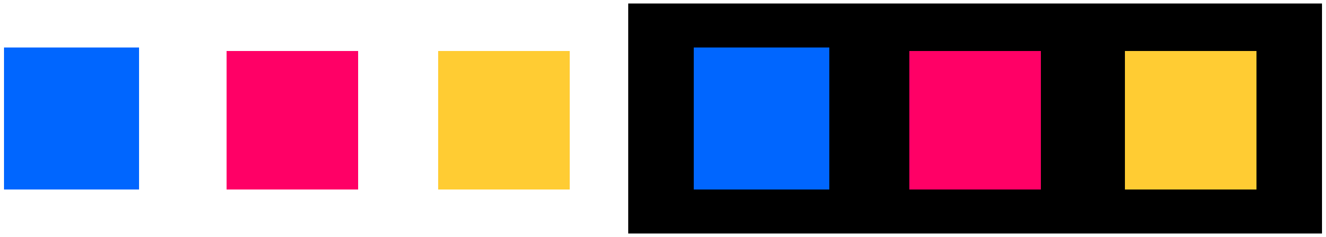
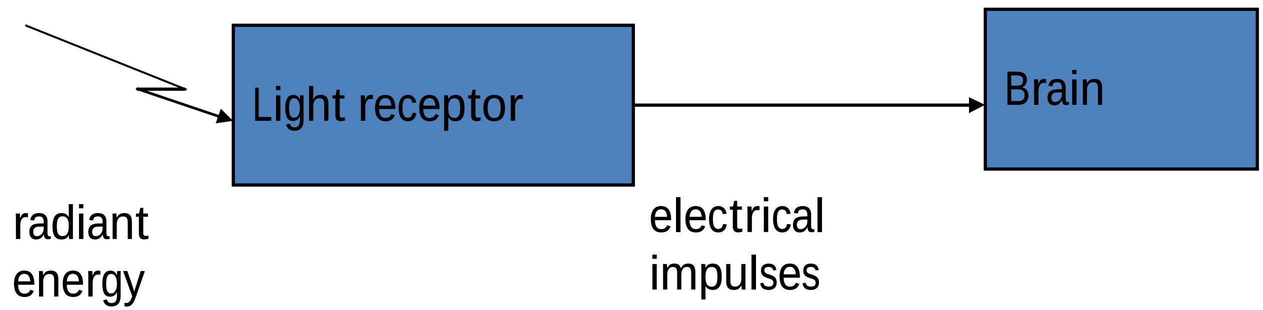


Image formation in the eye

- Flexible lens
- Controlled by the tension in the fibers of the ciliary body
  - To focus on distant objects?
  - To focus on objects near eye?
  - Near-sighted and far-sighted

Image formation in the eye



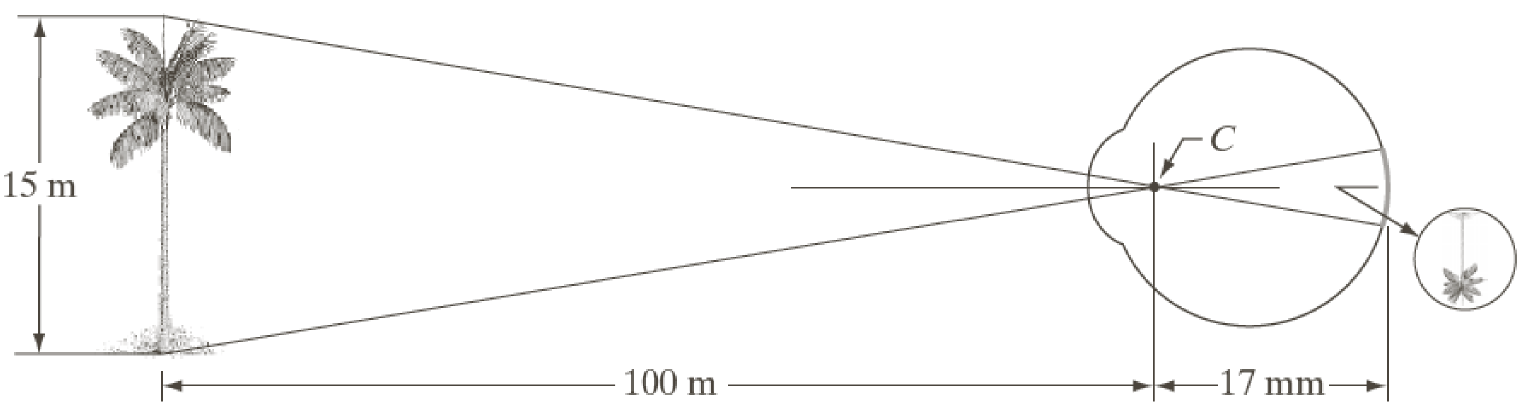


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

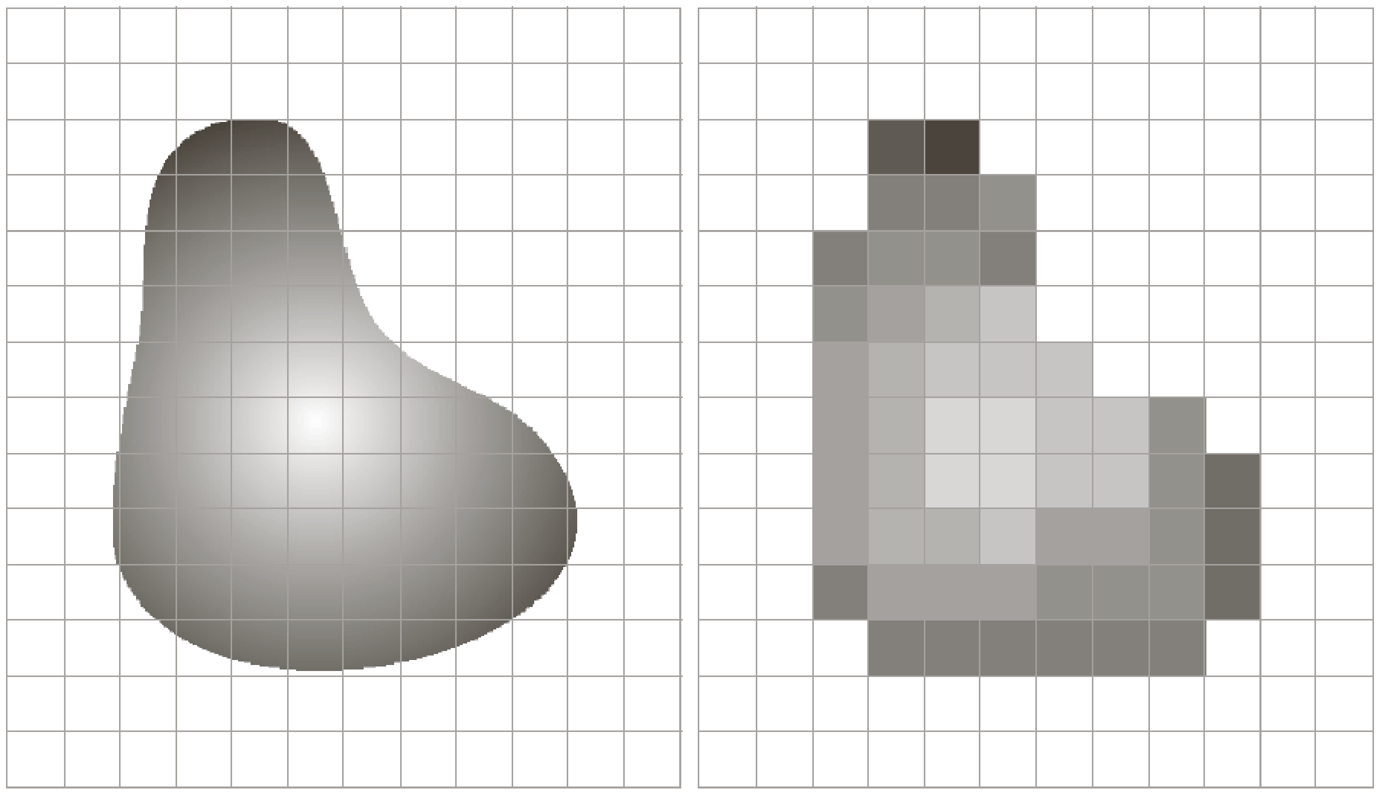
A simple image formation model

- $f(x,y)$: the intensity is called the **gray level** for monochrome image
- $f(x, y) = i(x, y).r(x, y)$
  - $0 < i(x, y) <$ inf, the illumination (lm/m^2)
  - $0 < r(x, y) < 1$, the reflectance
- Some illumination figures (lm/m^2)
  - 90,000: full sun - 0.01: black velvet
  - 10,000: cloudy day - 0.93: snow
  - 0.1: full moon
  - 1,000: commercial office

Camera exposure

- ISO number
  - Sensitivity of the film or the sensor
  - Can go as high as 1,600 and 3,200
- Shutter speed
  - How fast the shutter is opened and closed
- f/stop
  - The size of aperture
  - 1.0 ~ 32

Sampling and Quantization



a b

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

Uniform sampling

- Digitized in spatial domain ($I_{M \times N}$)
- M and N are usually integer powers of two
- Nyquist theorem and Aliasing

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)

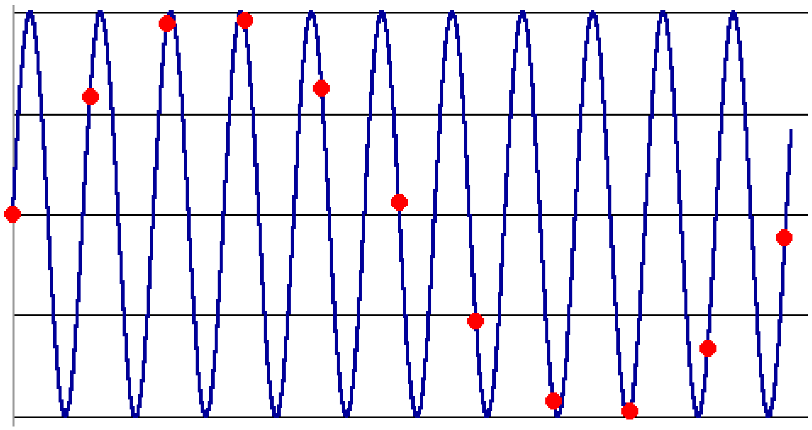
$\xrightarrow{\text{Sampled by 2}}$

(0,0)	(0,0)	(0,2)	(0,2)
(0,0)	(0,0)	(0,2)	(0,2)
(2,0)	(2,0)	(2,2)	(2,2)
(2,0)	(2,0)	(2,2)	(2,2)

- Non-uniform sampling
  - communication

More on aliasing

- Aliasing (the Moire effect)

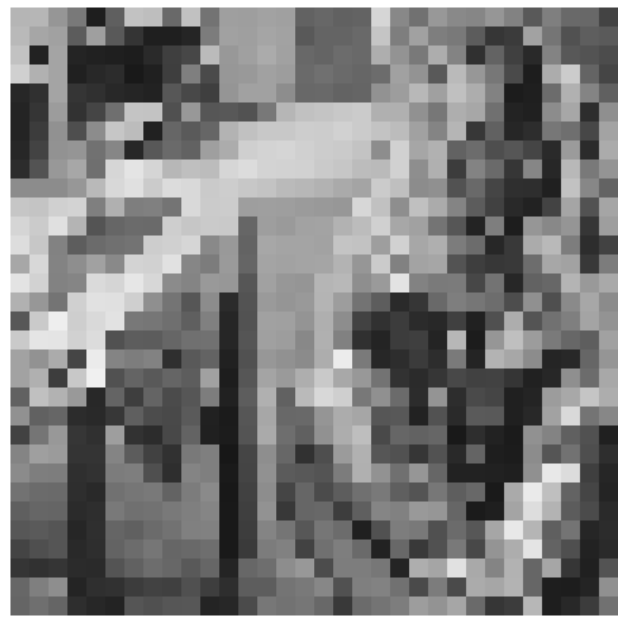



original


Sampled by 2

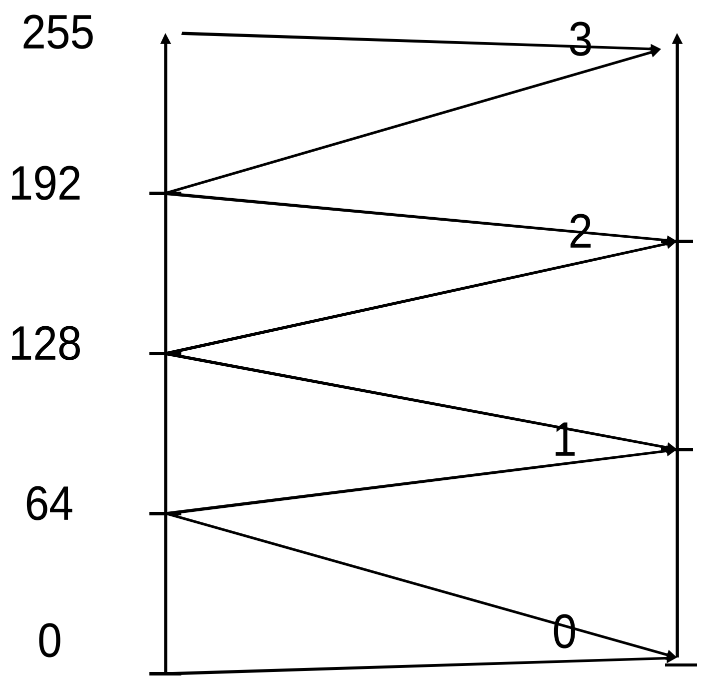

Sampled by 4


Sampled by 8


Sampled by 16

Uniform quantization

- Digitized in amplitude (or pixel value)
- PGM – 256 levels ➔ 4 levels




original


128 levels (7 bits)


16 levels (4 bits)


4 levels (2 bits)


2 levels (1 bit)

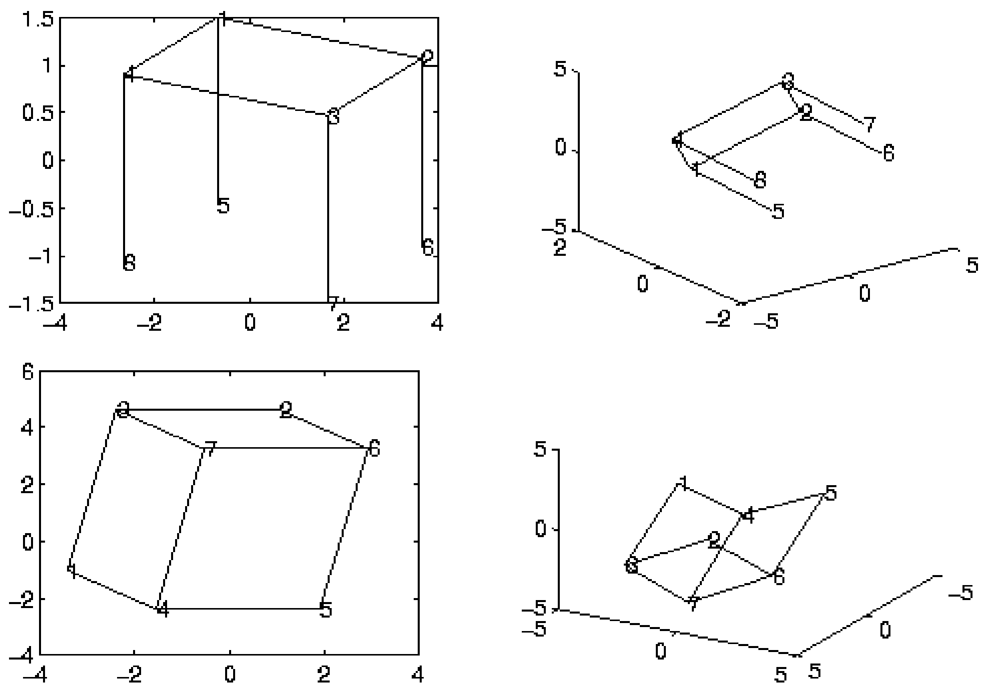
Image resolution

- Spatial resolution
  - Line pairs per unit distance
  - Dots/pixels per unit distance
    - dots per inch - dpi
- Intensity resolution
  - Smallest discernible change in intensity level

- The more samples in a fixed range, the higher the resolution
- The more bits, the higher the resolution

3D interpretation of line drawing

- Emulation approach
  - A given 3-D interpretation is considered less likely to be correct if some angles between the wires are much larger than others



Representing digital images

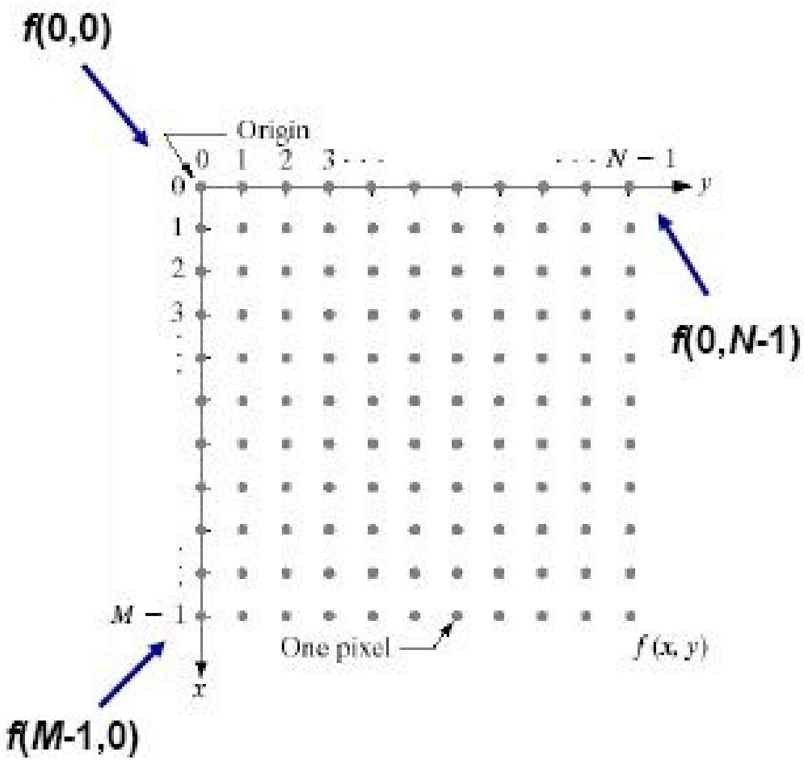


FIGURE 2.18 Coordinate convention used in this book to represent digital images.

L: # of discrete gray levels

$L = 2^k$

$b = M \times N \times k$

Every pixel has a # of bits.

Digital Image Representation Coordinate Conventions

- The result of sampling and quantization is a matrix of real numbers
- There are two principle ways to represent a digital image:
  - Assume that an image *f(x,y)* is sampled so that the resulting image has *M* rows and *N* columns. We say that the image is of size

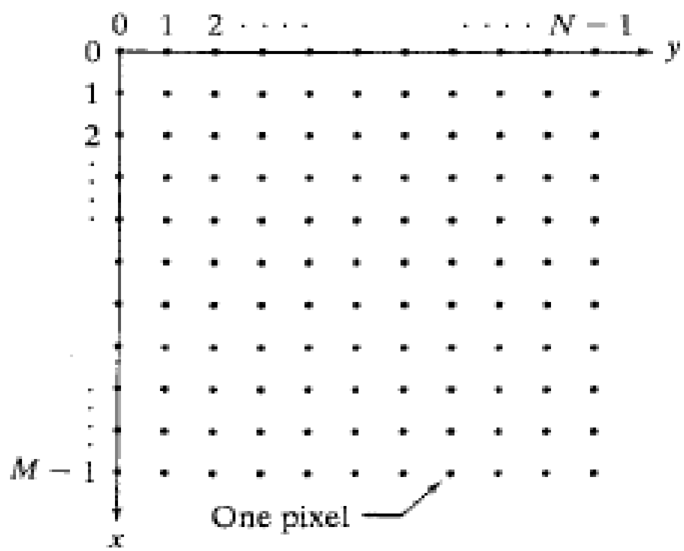
    M x *N*. The values of the coordinates (x,y) are discrete quantities. For clarity, we use integer values for these discrete coordinates.

    In many image processing books, the image origin is defined to be at (x,y) = (0,0). The next coordinate values along the first row of the image are *(x,y)* = (0,1). It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates. Note that x ranges from 0 to M-1, and y ranges from 0 to N-1. Figure (a)

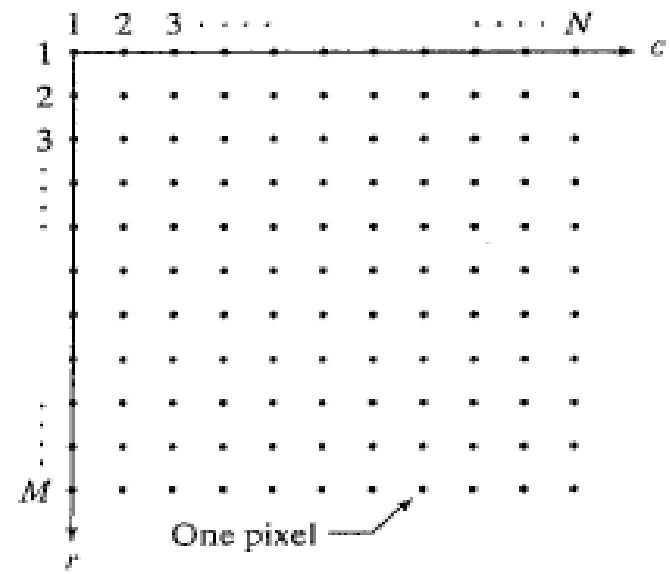
Digital Image Representation Coordinate Conventions

- The coordinate convention used in toolbox to denote arrays is different from the preceding paragraph in two minor ways.
  - Instead of using *(x,y)* the toolbox uses the notation *(r,c)* to indicate rows and columns.
  - The origin of the coordinate system is at *(r,c)* = (1,1); thus, r ranges from 1 to M and c from 1 to N, in integer increments. This coordinate convention is shown in Figure (b).

Digital Image Representation Coordinate Conventions



(A)



(B)

Digital Image Representation
Images as Matrices

- The coordination system in figure (A) and the preceding discussion lead to the following representation for a digitized image function:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

Digital Image Representation
Images as Matrices

- The right side of the equation is a digital image by definition. Each element of this array is called an *image element*, *picture element*, *pixel* or *pel*.
- A digital image can be represented naturally as a MATLAB matrix:

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

where f(1,1) = *f(0,0)*. Clearly, the two representations are identical, except for the shift in origin.

Pixels!

- Every pixel has # of bits (k)

- Q: Suppose a pixel has 1 bit, how many gray levels can it represent?
  Answer: 2 intensity levels only, black and white.
  Bit (0,1) → 0:black , 1: white

- Q: Suppose a pixel has 2 bit, how many gray levels can it represent?
  Answer: 4 gray intensity levels
  2Bit (00, 01, 10 ,11).

Now ..
if we want to represent 256 intensities of grayscale, how many bits do we need?
Answer: 8 bits → which represents: $2^8=256$

so, the gray intensities (L) that the pixel can hold, is calculated according to according to number of pixels it has (k). $L= 2^k$

Number of storage of bits:

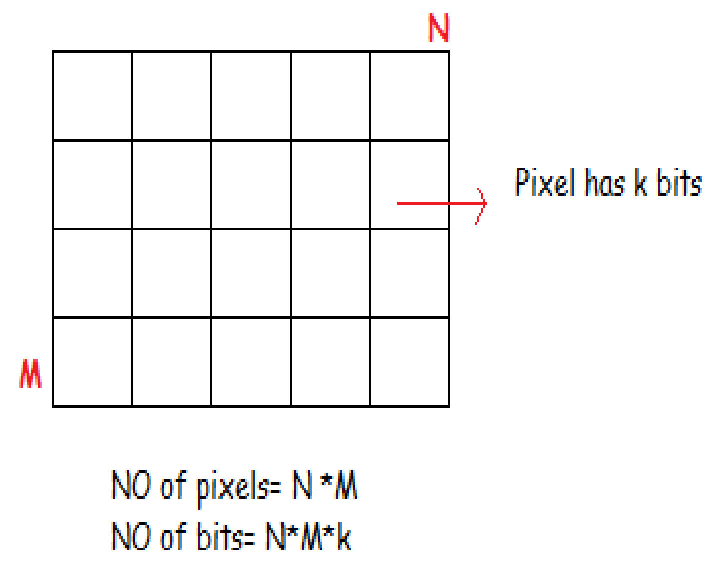
N * M: the no. of pixels in all the image.

K: no. of bits in each pixel

L: grayscale levels the pixel can represent

$L = 2^K$

all bits in image= N*N*k



Number of storage of bits:

EX: Here: N=32,   K=3,   L = 2^3 =8

# of pixels=N*N = 1024 . (because in this example: M=N)

# of bits = N*N*K = 1024*3= 3072

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

N=M in this table, which means no. of horizontal pixels= no. of vertical pixels. And thus:

# of pixels in the image= N*N

Spatial and gray-level resolution

- Sampling is the principal factor determining the spatial resolution of an image
- Basically, spatial resolution is the smallest discernible detail in an image.
- Spatial Resolution

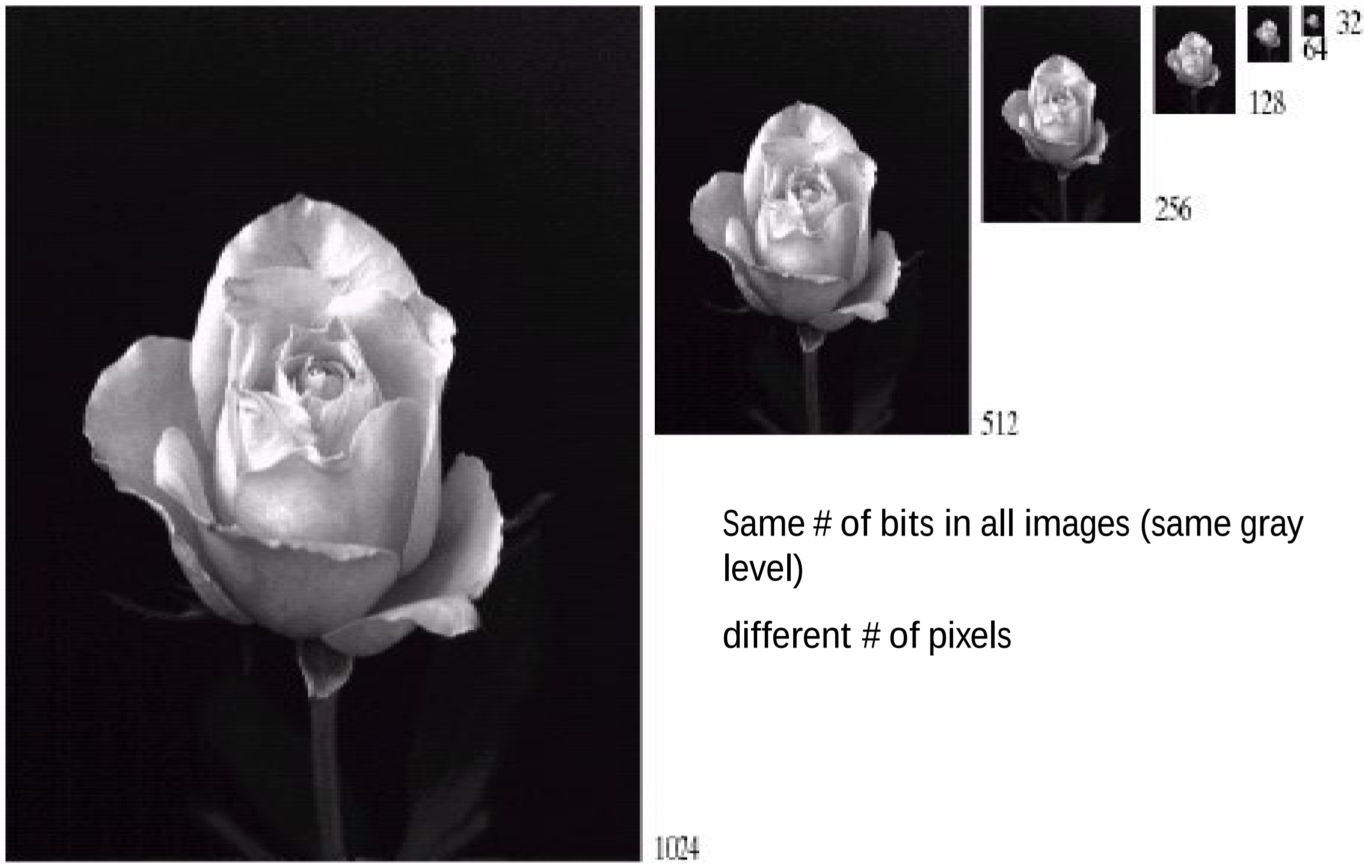
Spatial and gray-level resolution

Spatial and gray-level resolution

- Gray-level resolution refers to the smallest discernible change in gray level.

Spatial and gray-level resolution

Sub sampling



Same # of bits in all images (same gray level)

different # of pixels

FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

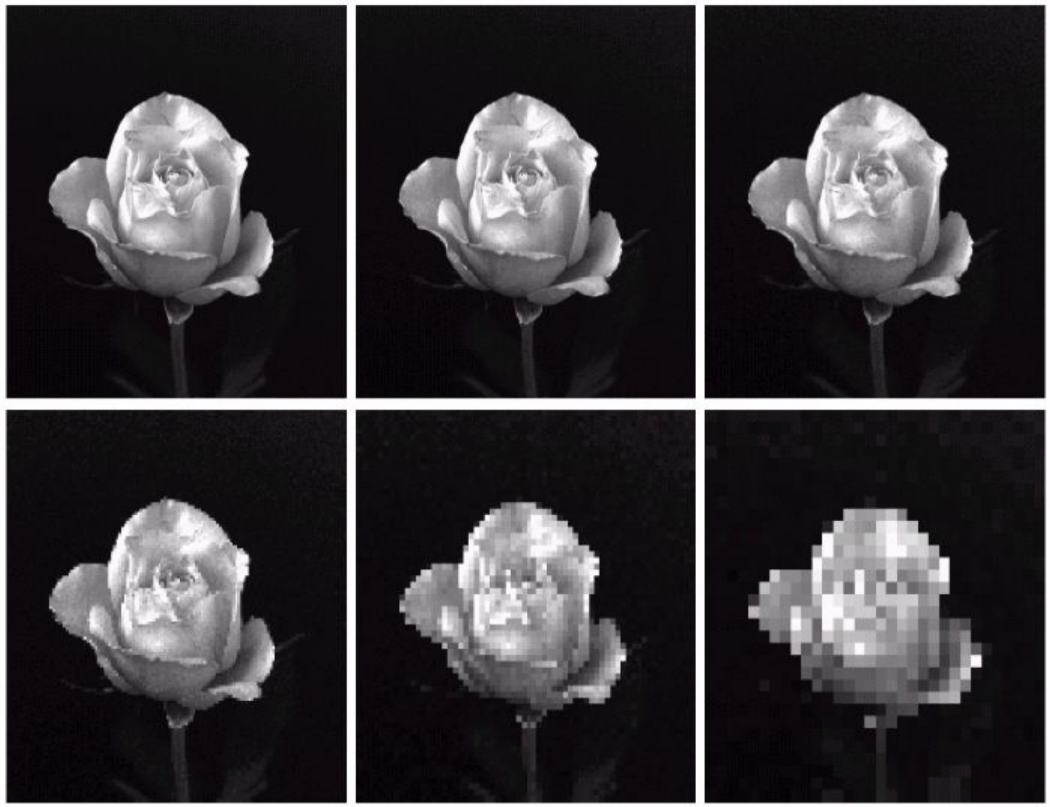
subSampling is performed by deleting rows and columns from the original image.

Spatial and gray-level resolution

Re sampling

(pixel replication)

A special case of nearest neighbor zooming.



a b c
d e f

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

Resampling is performed by row and column duplication

UNIT II
IMAGE TRANSFORMS

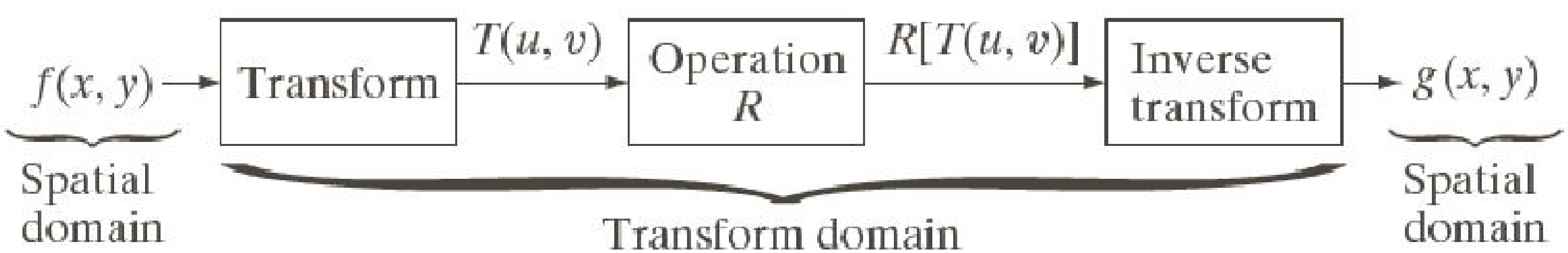
- 1D DFT
- 2D DFT
- Cosine
- Sine Hadamard
- Hear
- Slant
- KL
- SVD transform and their properties.

Image Transforms

- Many times, image processing tasks are best performed in a domain other than the *spatial domain*.
- Key steps

  (1) Transform the image

  (2) Carry the task(s) in the *transformed domain*.



Transformation Kernels

- Forward Transformation

forward transformation kernel

$$T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)r(x,y,u,v) \qquad u = 0,1,\ldots,M-1,\ \ v = 0,1,\ldots,N-1$$

- Inverse Transformation

inverse transformation kernel

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} T(u,v)s(x,y,u,v) \qquad x = 0,1,\ldots,M-1,\ \ y = 0,1,\ldots,N-1$$

Kernel Properties

- A kernel is said to be *separable* if:

$$r(x, y, u, v) = r_1(x, u) r_2(y, v)$$

- A kernel is said to be *symmetric* if:

$$r(x, y, u, v) = r_1(x, u) r_1(y, v)$$

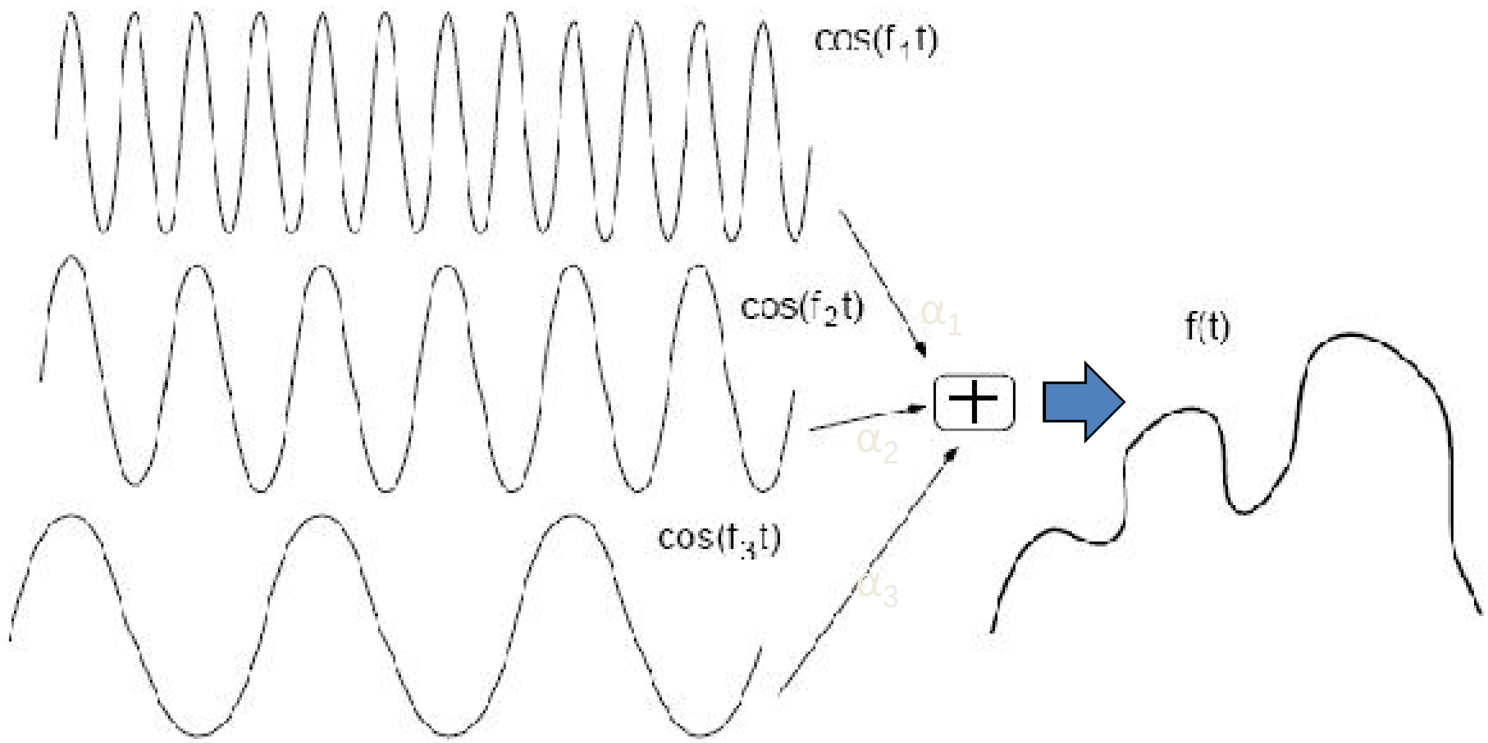
Fourier Series Theorem

- Any periodic function f(t) can be expressed as a weighted sum (infinite) of sine and cosine functions of varying frequency:

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n cos(nf_0 t) + \sum_{n=1}^{\infty} b_n sin(nf_0 t)$$

f_0 is called the “fundamental frequency”

Fourier Series (cont'd)



Continuous Fourier Transform (FT)

- Transforms a signal (i.e., function) from the **spatial** (x) domain to the **frequency** (u) domain.

Forward FT: $$\mathcal{F}(f(x)) = F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

Inverse FT: $$\mathcal{F}^{-1}(F(u)) = f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

where $$e^{\pm j\theta} = cos(\theta) \pm jsin(\theta)$$

Definitions

- **F(u)** is a complex function:

$$F(u) = R(u) + jI(u)$$

- Magnitude of FT (spectrum):

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

- Phase of FT:

$$\phi(F(u)) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right)$$

- Magnitude-Phase representation:

$$F(u) = |F(u)|e^{j\phi(u)}$$

- Power of f(x): **P(u)**=|F(u)|^2=

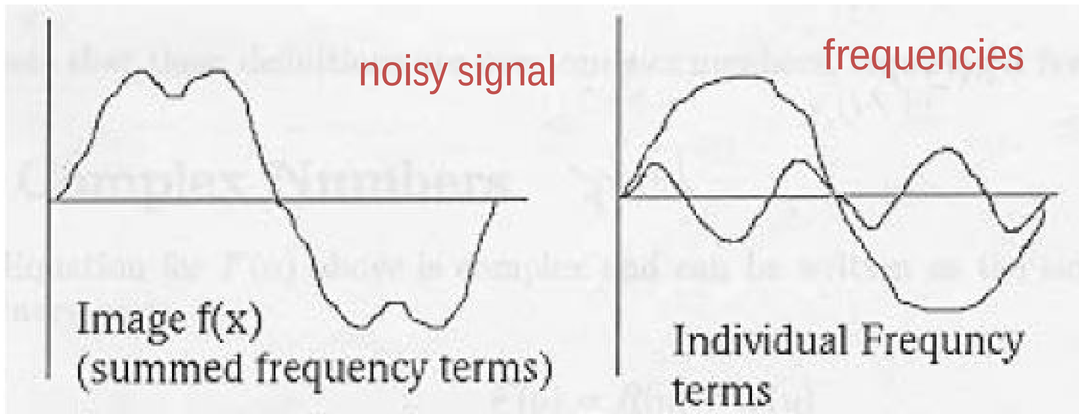
$$R^2(u) + I^2(u)$$

Why is FT Useful?

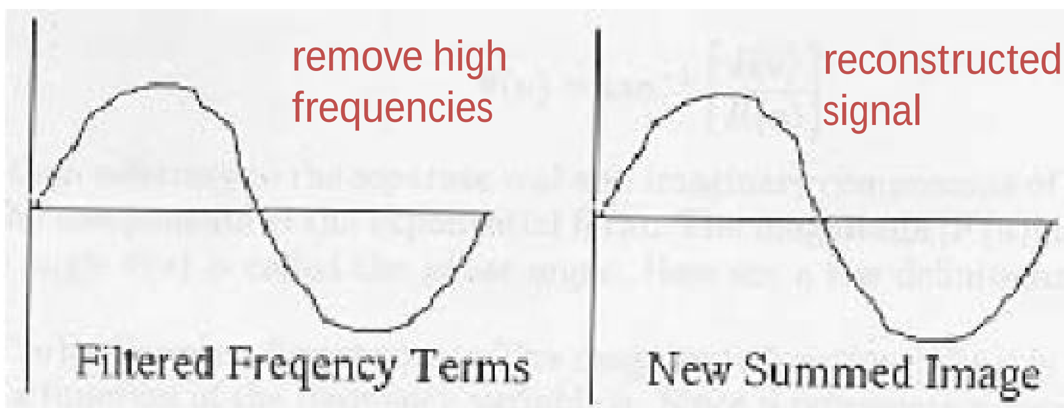
- Easier to remove undesirable frequencies in the **frequency** domain.

- Faster to perform certain operations in the **frequency** domain than in the **spatial** domain.

Example: Removing undesirable frequencies

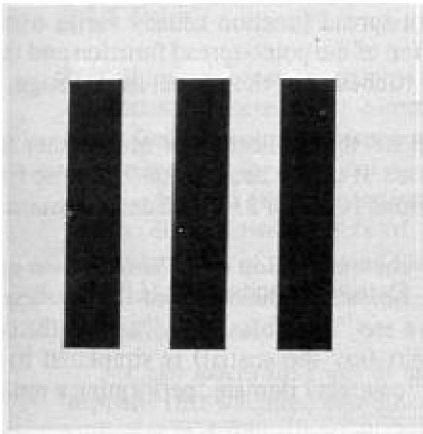


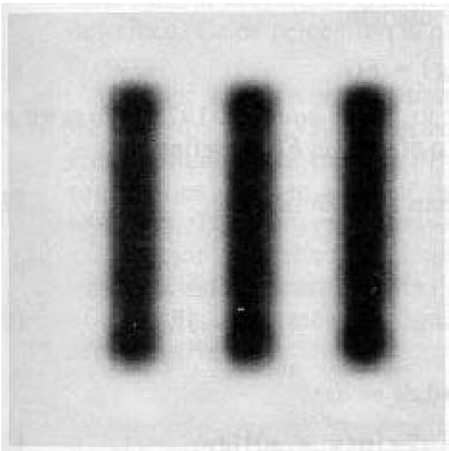
To remove certain frequencies, set their corresponding F(u) coefficients to zero!



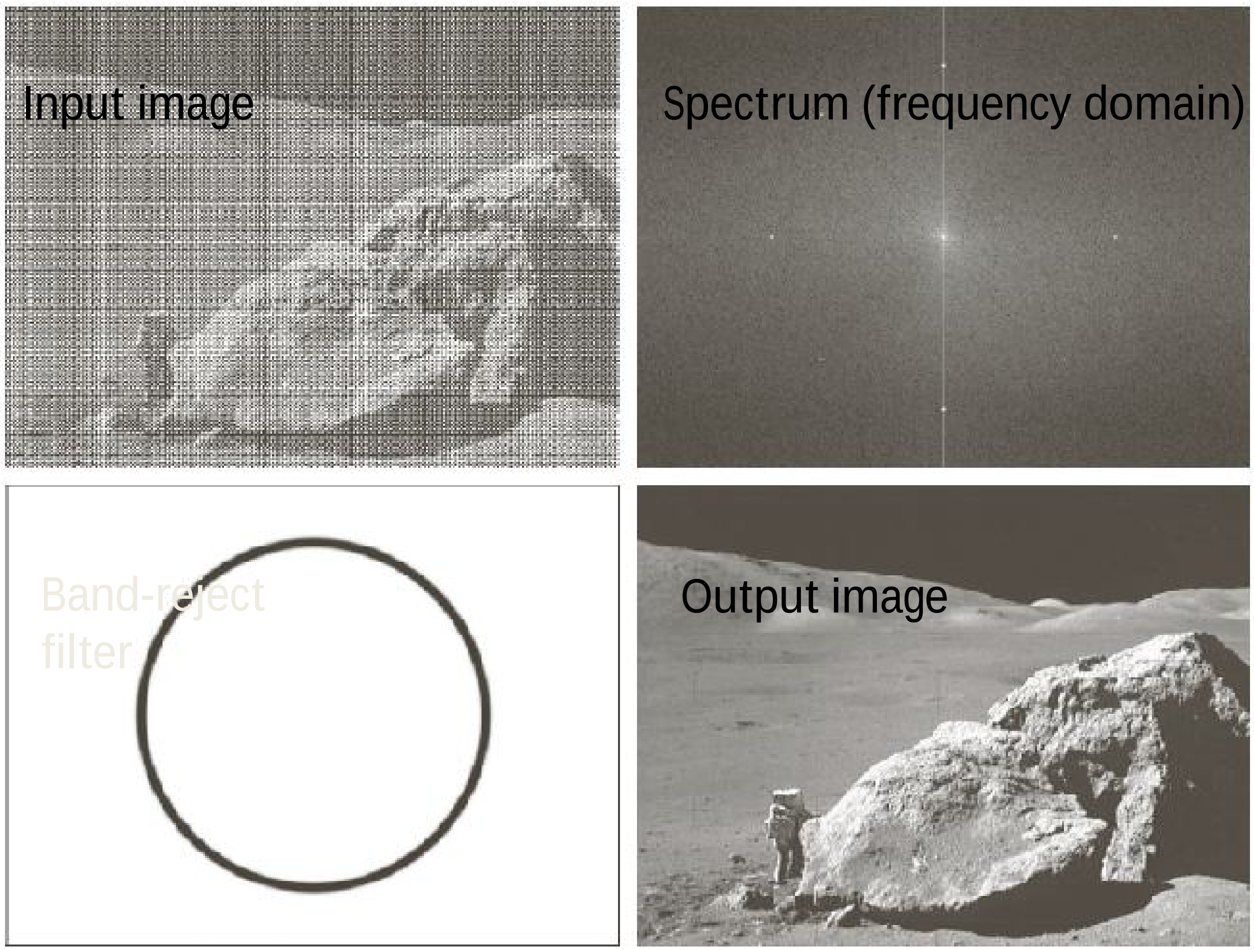
How do frequencies show up in an image?

- Low frequencies correspond to slowly varying pixel intensities (e.g., continuous surface).
- High frequencies correspond to quickly varying pixel intensities (e.g., edges)


Original Image


Low-passed

Example of noise reduction using FT



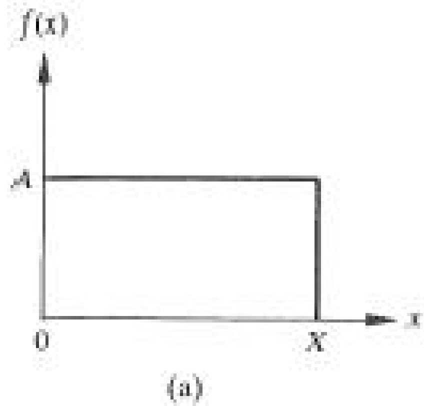
Frequency Filtering: Main Steps

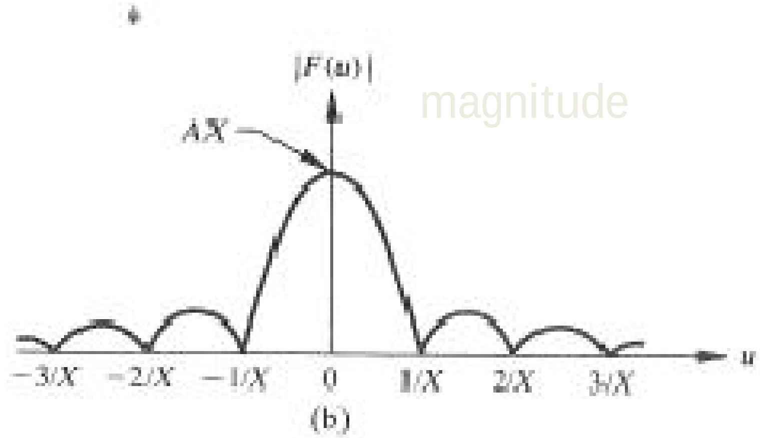
1. Take the FT of f(x): $\mathcal{F}(f(x))$

2. Remove undesired frequencies: $D(\mathcal{F}(f(x)))$

3. Convert back to a signal: $\hat{f}(x) = \mathcal{F}^{-1}(D(\mathcal{F}(f(x))))$

Example: rectangular pulse





rect(x) function

sinc(x)=sin(x)/x

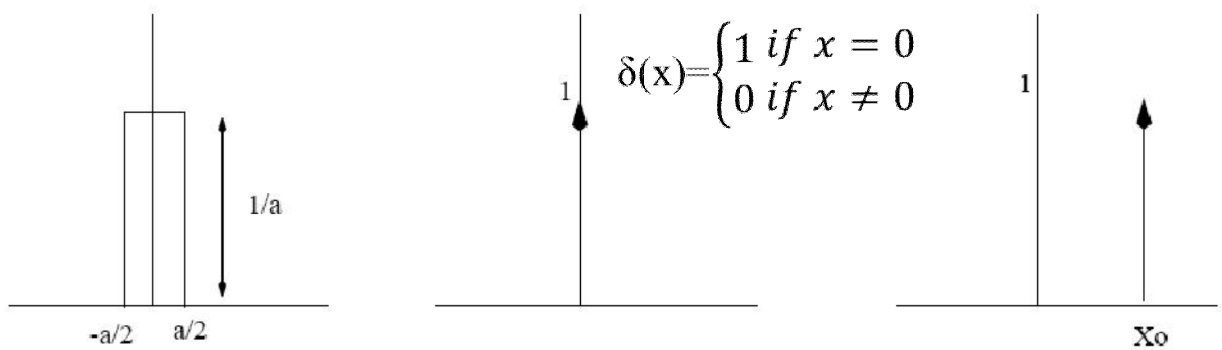
Example: impulse or “delta” function

- Definition of delta function:

$$\delta(x) = \lim_{a \to 0} \frac{1}{a}\, rect(x/a)$$

- Properties:

$$\int_{-\infty}^{\infty} \delta(x)\,dx = 1 \qquad \int_{-\infty}^{\infty} f(x)\,\delta(x - x_0)\,dx = f(x_0)$$

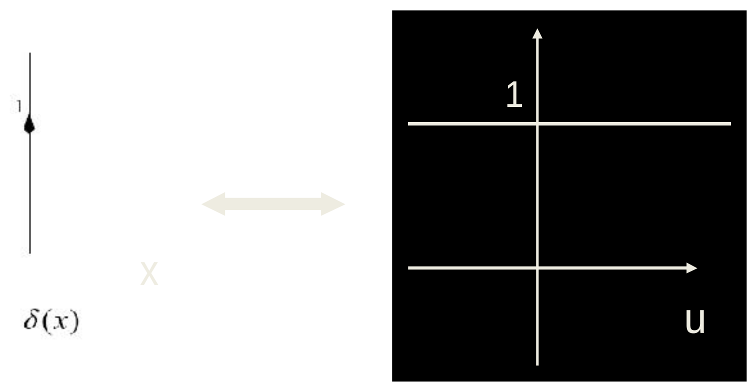


$\frac{1}{a}\, rect(x/a)$ $\quad \delta(x)$ $\quad \delta(x - x_0)$

Example: impulse or “delta” function (cont’d)

- FT of delta function:

$$F(\delta(x)) = \int_{-\infty}^{\infty} \delta(x) e^{-j2\pi ux} dx = e^0 = 1$$



Example: spatial/frequency shifts

$$f(x) \leftrightarrow F(u), \quad then$$

$$(1) \quad f(x - x_0) \leftrightarrow e^{-j2\pi u x_0} F(u)$$

$$(2) \quad f(x) e^{j2\pi u_0 x} \leftrightarrow F(u - u_0)$$

Special Cases:

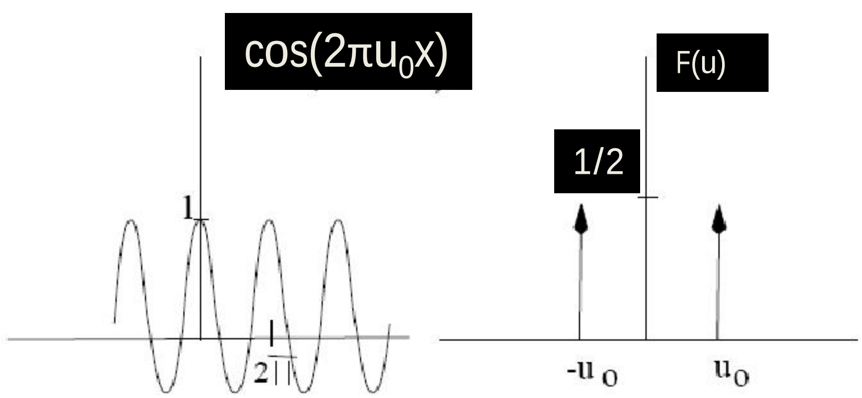
$$\delta(x - x_0) \leftrightarrow e^{-j2\pi u x_0}$$

$$e^{j2\pi u_0 x} \leftrightarrow \delta(u - u_0)$$

Example: sine and cosine functions

- FT of the cosine function

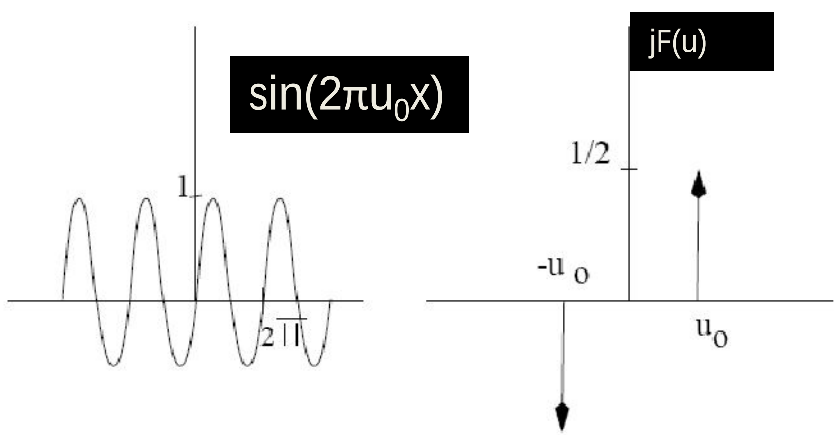
$$F(cos(2\pi u_0 x)) = \frac{1}{2}[\delta(u-u_0)+\delta(u+u_0)]$$



Example: sine and cosine functions (cont'd)

- FT of the sine function

$$\Im(\sin(2\pi u_0 x)) = \frac{1}{2j}[\delta(u-u_0) - \delta(u+u_0)]$$



Extending FT in 2D

- Forward FT

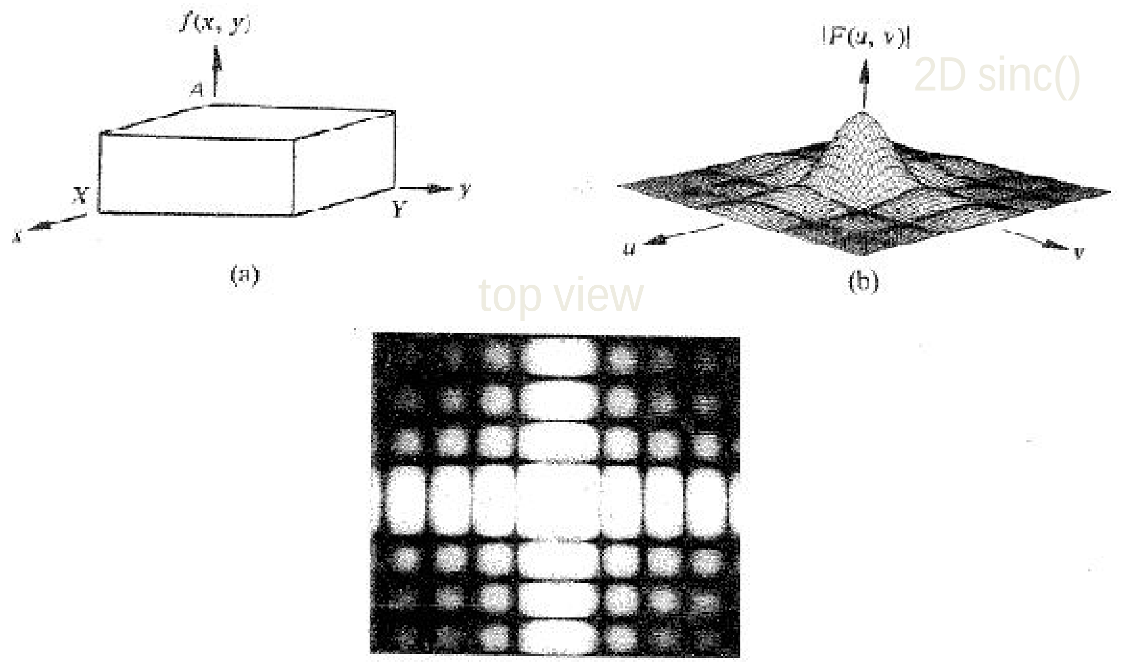
$$\mathcal{F}(f(x,y)) = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dxdy$$

- Inverse FT

$$\mathcal{F}^{-1}(F(u,v)) = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+uy)}\,dudv$$

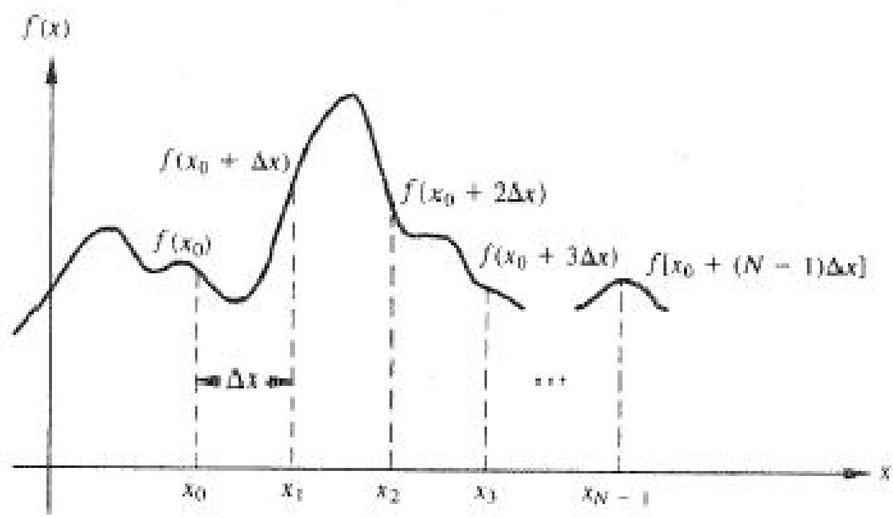
Example: 2D rectangle function

- FT of 2D rectangle function



(a) (b) (c)

Discrete Fourier Transform (DFT)



$$f(x) = f(x_0 + x\Delta x),\ x = 0, 1, \ldots, N-1$$

Discrete Fourier Transform (DFT) (cont'd)

- Forward DFT

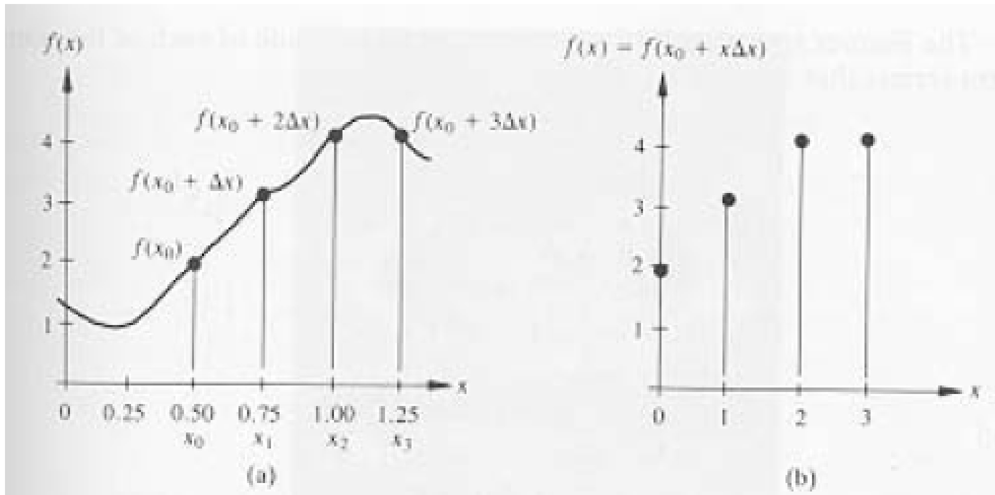
$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{\frac{-j2\pi ux}{N}}, \; u = 0, 1, \ldots, N-1$$

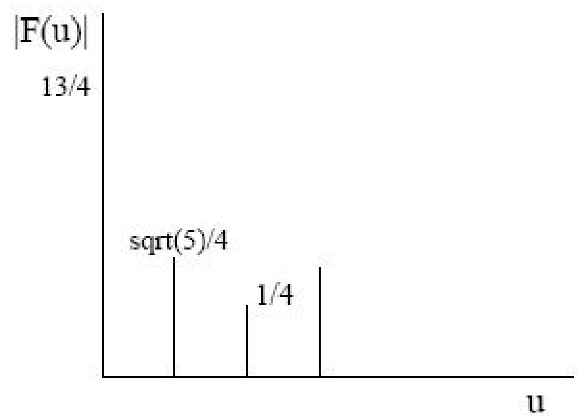
- Inverse DFT

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, \; x = 0, 1, \ldots, N-1$$

$F(u)$ is discrete: $F(u) = F(u\Delta u)$, $u = 0, 1, \ldots, N-1$, $\Delta u =$ 1/(NΔx)

Example





$N-4$

$\Delta x = 0.25$

$\Delta u = \dfrac{1}{N\Delta x} = 1$

Extending DFT to 2D

- Assume that f(x,y) is M x N.

- Forward DFT

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$(u = 0, 1, \ldots, M-1,\ v = 0, 1, \ldots, N-1)$$

- Inverse DFT:

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$(x = 0, 1, \ldots, M-1,\ y = 0, 1, \ldots, N-1)$$

Extending DFT to 2D (cont'd)

- Special case: f(x,y) is N x N.

- Forward DFT

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux+vy}{N})},$$

u,v = 0,1,2, ..., N-1

- Inverse DFT

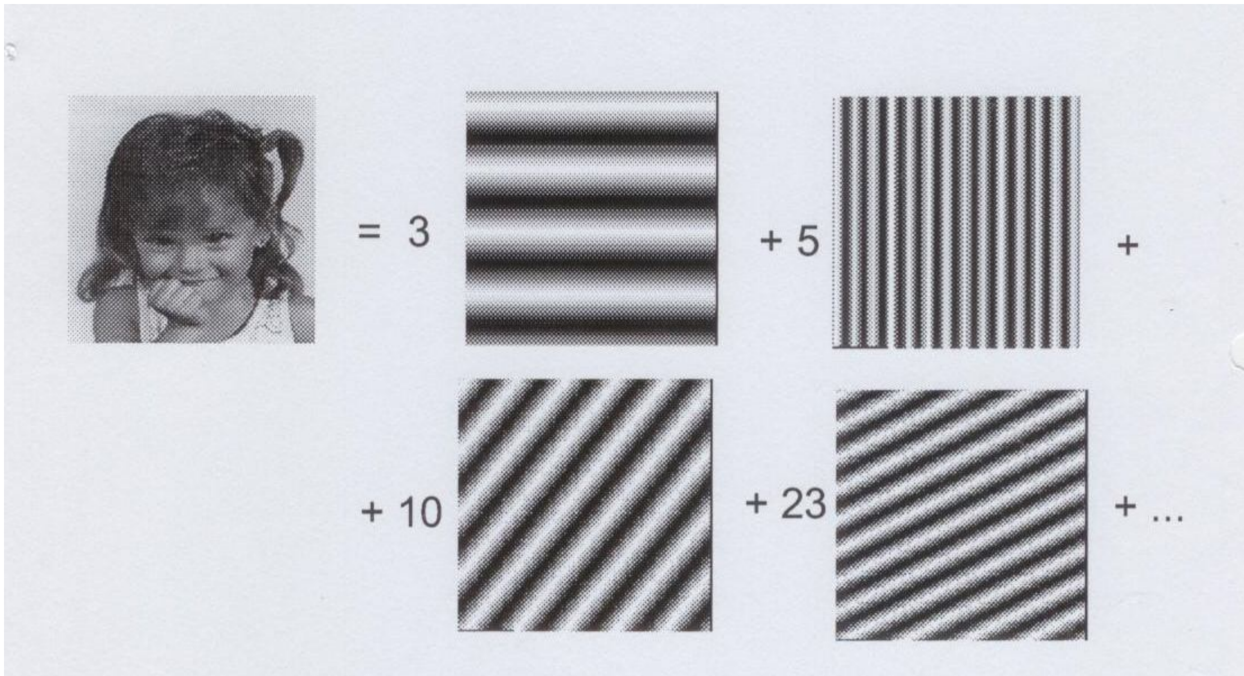
$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux+vy}{N})},$$

x,y = 0,1,2, ..., N-1

Extending DFT to 2D (cont'd)

2D cos/sin functions

Interpretation:



$$f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi\left(\frac{ux+vy}{N}\right)},$$

Visualizing DFT

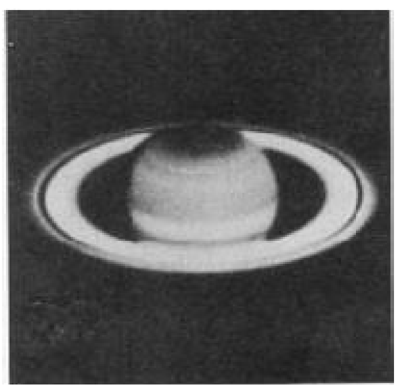
- Typically, we visualize |F(u,v)|
- The dynamic range of |F(u,v)| is typically very large

$$D(u,v) = c\ log(1 + |F(u,v)|)$$

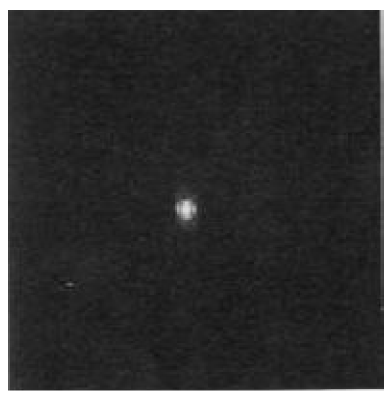
- Apply streching: (c is const)

|F(u,v)|

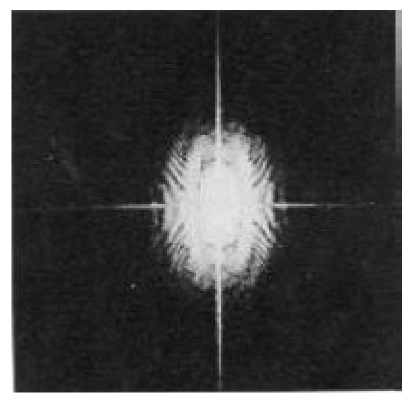
|D(u,v)|



original image



before stretching



after stretching

DFT Properties: (1) Separability

- The 2D DFT can be computed using 1D transforms **only**:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux+vy}{N})}$$

Forward DFT:

kernel is separable:

$$e^{-j2\pi(\frac{ux+vy}{N})} = e^{-j2\pi(\frac{ux}{N})}e^{-j2\pi(\frac{vy}{N})}$$

DFT Properties: (1) Separability (cont'd)

- Rewrite F(u,v) as follows:

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j2\pi(\frac{ux}{N})} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{vy}{N})}$$

- Let's set: $$\sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{vy}{N})} = F(x,v)$$

- Then: $$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j2\pi(\frac{ux}{N})} F(x,v)$$

DFT Properties: (1) Separability (cont'd)

- How can we compute F(x,v)?

$$\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{vy}{N})} = F(x,v) = N\left(\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{vy}{N})}\right)$$

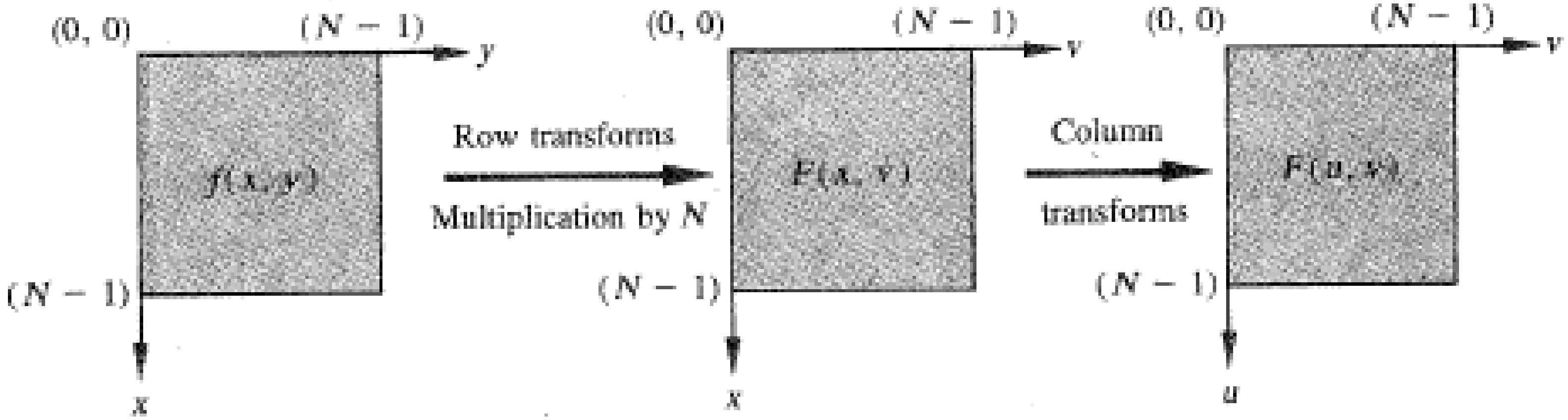
N x DFT of rows of f(x,y)

- How can we compute F(u,v)?

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} e^{-j2\pi(\frac{ux}{N})} F(x,v)$$

DFT of cols of F(x,v)

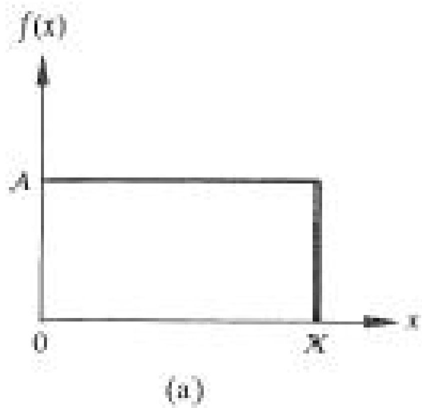
DFT Properties: (1) Separability (cont'd)

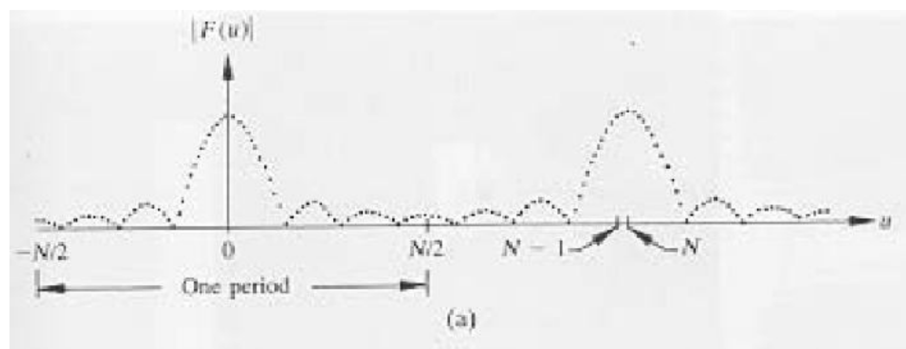


DFT Properties: (2) Periodicity

- The DFT and its inverse are periodic with period N

$$F(u,v) = F(u+N,v) = F(u,v+N) = F(u+N,v+N)$$





DFT Properties: (3) Symmetry

	Spatial Domain[†]		Frequency Domain[†]
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) - -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u. -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(\ u\ \ v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd

[†]Recall that x, y, u, and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y, and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

DFT Properties: (4) Translation

$$f(x,y) \longleftrightarrow F(u,v)$$

- Translation in spatial domain:

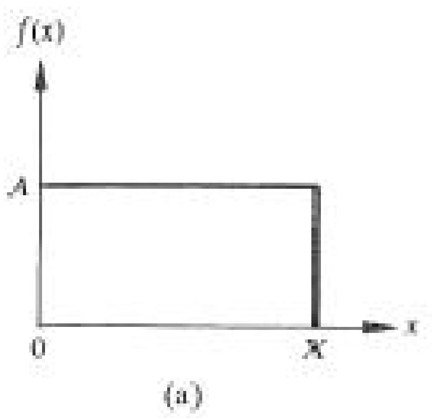
$$f(x-x_0, y-y_0) \Longleftrightarrow F(u,v)e^{-j2\pi(\frac{ux_0+vy_0}{N})}$$

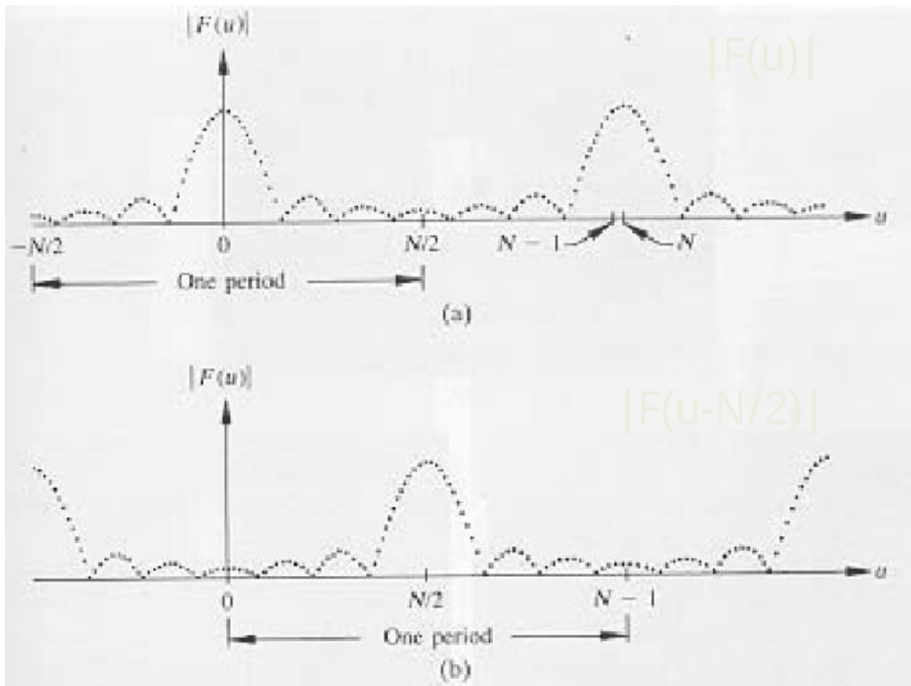
- Translation in frequency domain:

$$f(x,y)e^{j2\pi(\frac{u_0x+v_0y}{N})} \Longleftrightarrow F(u-u_0, v-v_0)$$

DFT Properties: (4) Translation (cont'd)

- To show a full period, we need to translate the origin of the transform at **u=N/2** (or at **(N/2,N/2)** in 2D)





DFT Properties: (4) Translation (cont'd)

$f(x,y)e^{j2\pi(u_0x+v_0yov}$ [redacted] $\Longleftrightarrow F(u-u_0, v-v_0)$

N

$u_0 = v_0 = N/2$

- To move F(u,v) at (N/2, N/2), take

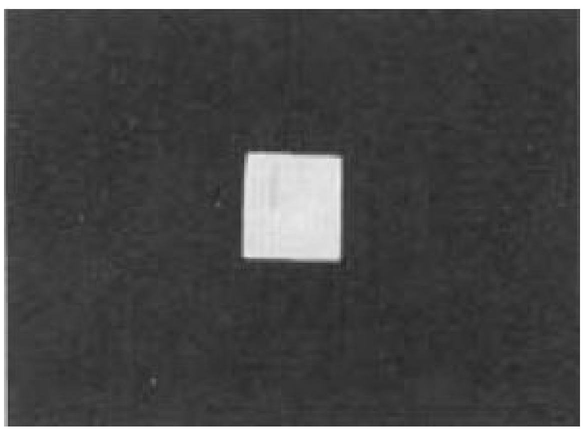
$e^{j2\pi(u_0x+v_0y}$ [redacted] $\Rightarrow$ $e^{j2\pi(\frac{\frac{N}{2}x+\frac{N}{2}y}{N})} = e^{j\pi(x+y)} = (-1)^{x+y}$

N

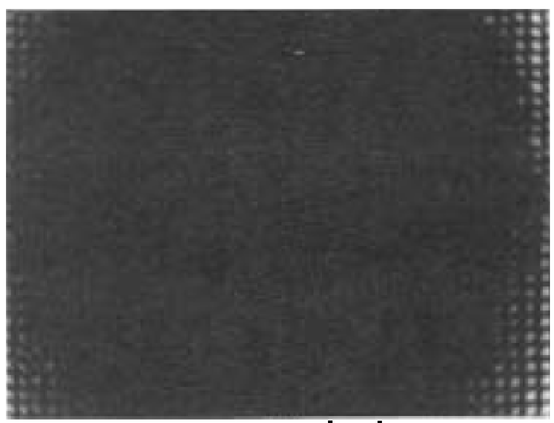
$$f(x,y)(-1)^{x+y} \Longleftrightarrow F(u-N/2,\ v-N/2)$$

DFT Properties: (4) Translation (cont'd)

$$f(x,y)(-1)^{x+y} \Longleftrightarrow F(u-N/2,\ v-N/2)$$

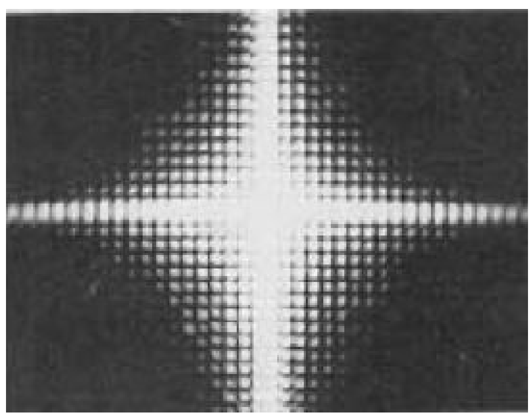


sinc



no translation

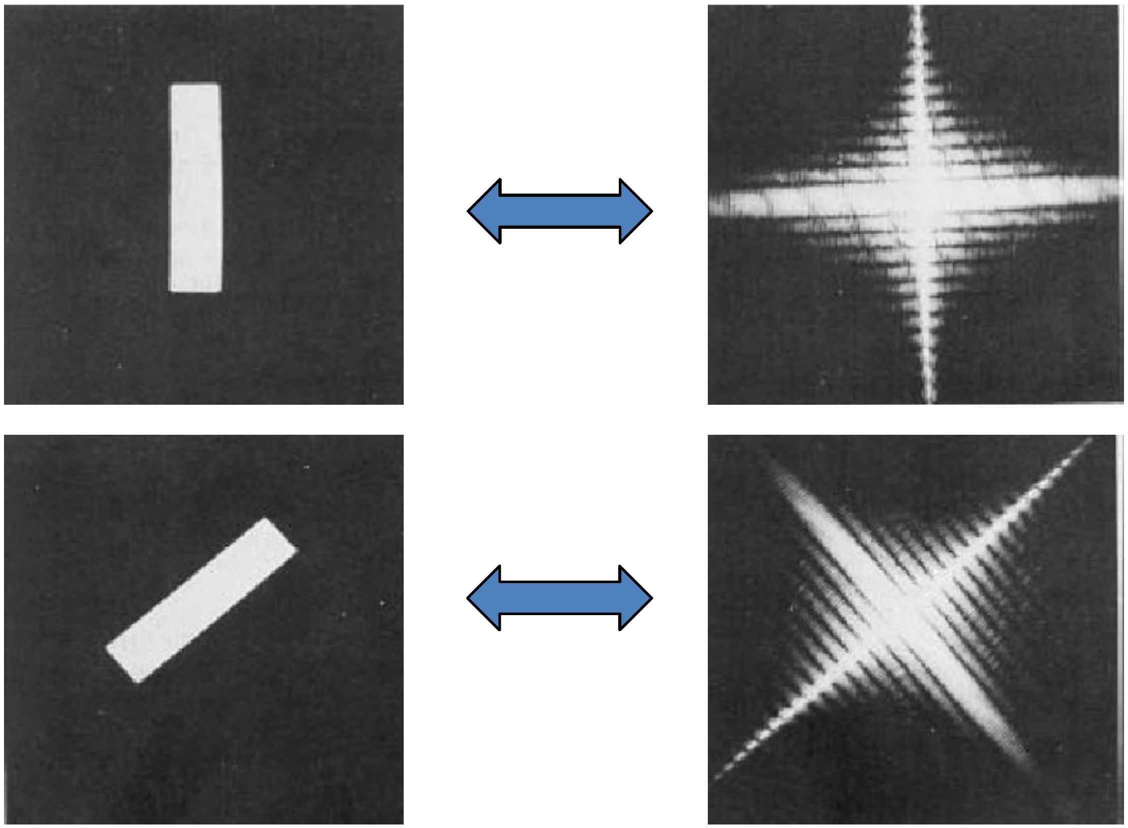
sinc



after translation

DFT Properties: (5) Rotation

- Rotating f(x,y) by θ rotates F(u,v) by θ



DFT Properties: (6) Addition/Multiplication

$$F[f(x,y)+g(x,y)]=F[f(x,y)]+F[g(x,y)]$$

but ... $$F[f(x,y)g(x,y)]\neq F[f(x,y)]F[g(x,y)]$$

DFT Properties: (8) Average value

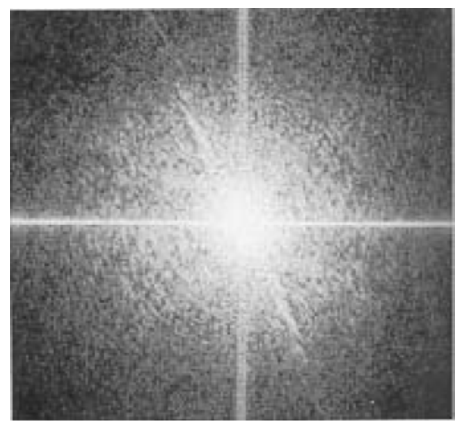
Average: $$\bar{f}(x,y) = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

F(u,v) at u=0, v=0: $$F(0,0) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

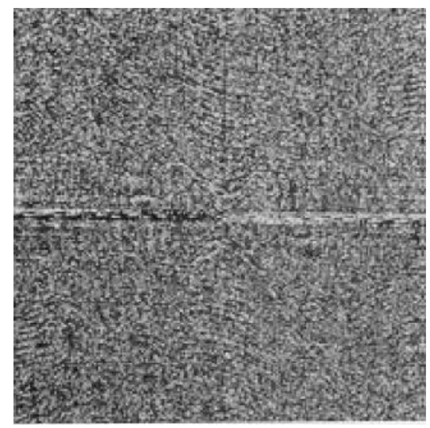
So: $$\bar{f}(x,y) = \frac{1}{N} F(0,0)$$

Magnitude and Phase of DFT

- What is more important?



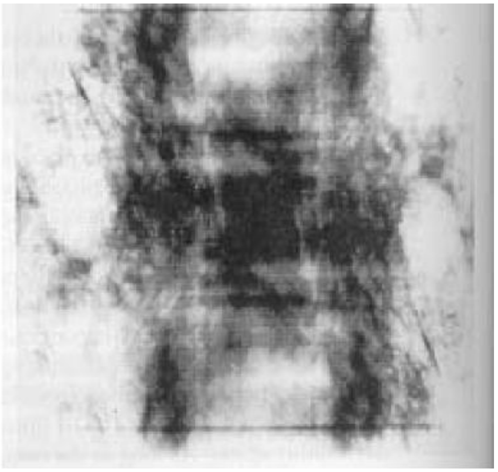
magnitude



phase

- **<u>Hint:</u>** use the inverse DFT to reconstruct the input image using only magnitude or phase information

Magnitude and Phase of DFT (cont'd)



Reconstructed image using
magnitude only
(i.e., magnitude determines the strength of each component)

Reconstructed image using
phase only
(i.e., phase determines the phase of each component)

Magnitude and Phase of DFT (cont'd)

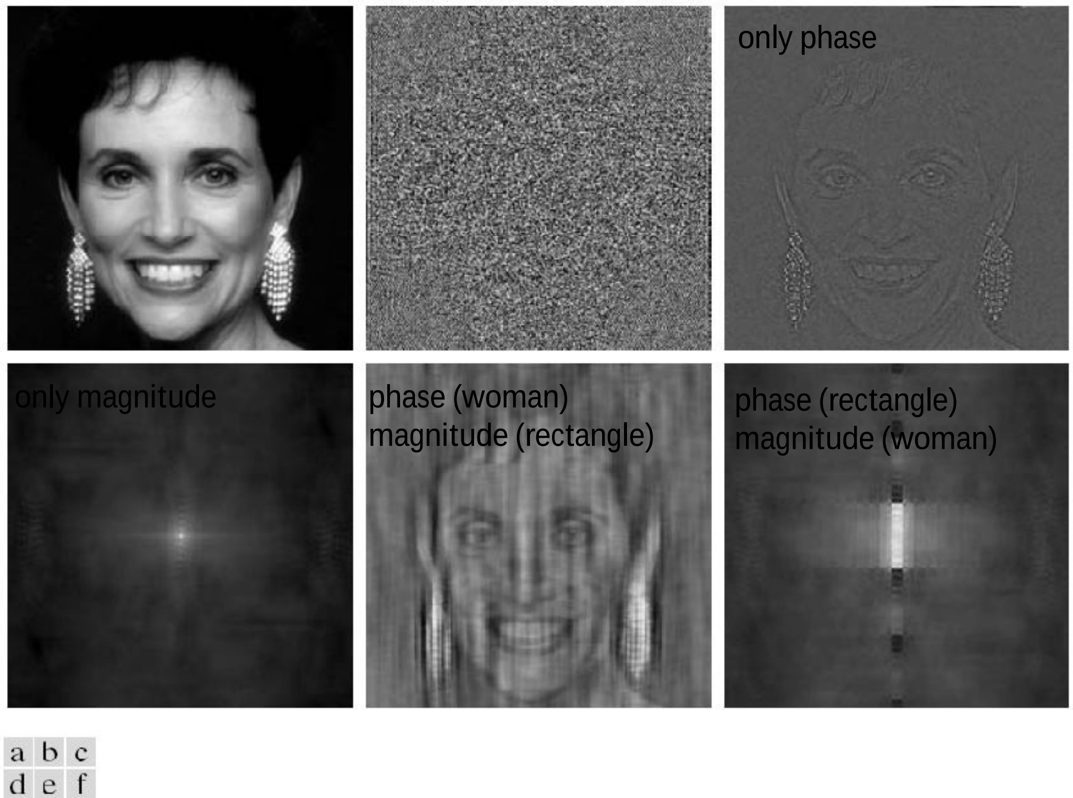


FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.