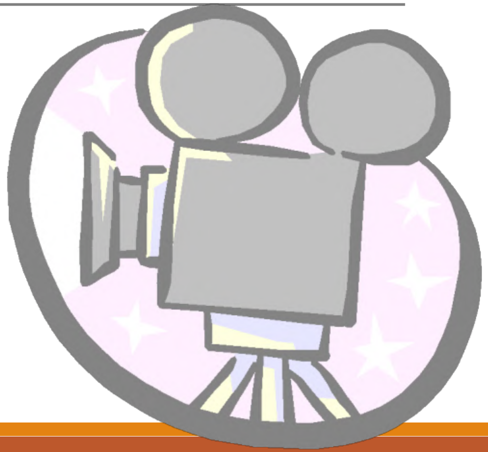
Image Processing: An Overview
CS-317/CS-341



Syllabus

CS341	Image Processing	L	T	P
		2	1	1

Digital Image Fundamental: Elements of Visual Perception- Structure of the human eye, Image formation in the eye, brightness adaptation and discrimination; light and electromagnetic spectrum, image sensing and acquisition, image sampling and quantization, basic relationships between pixels, linear and nonlinear operations.

Image Enhancement:Point processing: Contrast stretching, power-law and gamma transformation. Histogram processing: histogram equalization and matching.

Filtering and Restoration:Degradation function and Noise Models, Spatial Domain Filtering: Correlation and Convolution, Smoothing Linear and Nonlinear Filters: Mean and Median Filters, Adaptive Filtering, Sharpening Linear and Nonlinear Filters: Derivative, Laplacian, Unsharp Masking, High-boost Filtering. Frequency Domain Filtering: Filtering: Low-pass (Smoothing) & High-Pass (Sharpening) Ideal, Butterworth and Gaussian Filtering, Unsharp Masking and High-Boost Filtering, Homomorphic Filtering, Periodic Noise Reduction and Inverse Filtering & Wiener Filtering.

Image Reconstruction from Projections: Transmission tomography, reflection tomography, emission tomography, magnetic resonance imaging, and projection based image processing. Radon transform, back projection operator, projection theorem, inverse radon transform, convolution filter back projection, reconstruction from blurred noisy projections, Fourier reconstruction, fan-beam reconstruction, algebraic methods and three dimensional tomography.

Image Compression: Introduction, Error criterion- objective and subjective criterion; Lossy compression- transform domain compression, JPEG compression, block truncation compression, vector quantization compression; Lossless compression- Huffman coding, arithmetic coding, transformed coding, run-length coding, block coding, quad tree coding, and contour coding.

Suggested Readings:

1. A. K. Jain, Fundamentals of Digital Image Processing, Pearson Education India, 2015.
2. Rafael Gonzalez, Richard Woods, Digital Image Processing, Pearson Education India, 2017.
3. R. H. Vollmerhausen, R.G. Driggers, Analysis of Sampled Imaging Systems, SPIE Press, 2001.
4. B. Chanda, D. D. Majumder, Digital Image Processing and Analysis, PHI, 2011.
5. A. C. Bovik, Handbook of Image and Video Processing (Communications, Networking & Multimedia). Academic Press, 2005.
6. J. S. Lim, Two Dimensional Signal and Image Processing, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
7. D. E. Dudgeon, Russell M. Mersereau, Multidimensional Signal Processing, Prentice Hall, 1983.
8. S. G. Wilson, Digital Modulation and Coding, Pearson Education, 2003.
9. H. Maître, From Photon to Pixel: The Digital Camera Handbook, Wiley- 2017.

Topics of Course

- Digital Image Fundamental
- Image Enhancement
- Filtering and Restoration
- Image Reconstruction from Projections
- Image Compression

Three related sub-fields

- Image Processing
- Computer Graphics
- Computer Vision

Digital Image?

What is this?

1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001
1000000110011101001011100100110101001000001

ASCII Table

Dec	Hex	Char	Action (*if non-printing*)	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	(null)	32	20	Space	64	40	@	96	60	`
1	1	SOH	(start of heading)	33	21	!	65	41	A	97	61	a
2	2	STX	(start of text)	34	22	"	66	42	B	98	62	b
3	3	ETX	(end of text)	35	23	#	67	43	C	99	63	c
4	4	EOT	(end of transmission)	36	24	$	68	44	D	100	64	d
5	5	ENQ	(enquiry)	37	25	%	69	45	E	101	65	e
6	6	ACK	(acknowledge)	38	26	&	70	46	F	102	66	f
7	7	BEL	(bell)	39	27	'	71	47	G	103	67	g
8	8	BS	(backspace)	40	28	(	72	48	H	104	68	h
9	9	TAB	(horizontal tab)	41	29	)	73	49	I	105	69	i
10	A	LF	(NL line feed, new line)	42	2A	*	74	4A	J	106	6A	j
11	B	VT	(vertical tab)	43	2B	+	75	4B	K	107	6B	k
12	C	FF	(NP form feed, new page)	44	2C	,	76	4C	L	108	6C	l
13	D	CR	(carriage return)	45	2D	-	77	4D	M	109	6D	m
14	E	SO	(shift out)	46	2E	.	78	4E	N	110	6E	n
15	F	SI	(shift in)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	(data link escape)	48	30	0	80	50	P	112	70	p
17	11	DC1	(device control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2	(device control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3	(device control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4	(device control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK	(negative acknowledge)	53	35	5	85	55	U	117	75	u
22	16	SYN	(synchronous idle)	54	36	6	86	56	V	118	76	v
23	17	ETB	(end of trans. block)	55	37	7	87	57	W	119	77	w
24	18	CAN	(cancel)	56	38	8	88	58	X	120	78	x
25	19	EM	(end of medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB	(substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	(escape)	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	(file separator)	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	(group separator)	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	(record separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US	(unit separator)	63	3F	?	95	5F	_	127	7F	DEL

American standard code for information exchange

128 characters are represented by different decimal numbers

As $2^7=128$

So 7 bits are required for representing ASCII code

Decimal to Binary & Binary to Decimal Conversion

2	65	1	LSB
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
	1	1	MSB

$(65)_{10} = (1000001)_2$

Binary to decimal

1000001

$1\times2^6+0\times2^5+0\times2^4+0\times2^3+0\times2^2+0\times2^1+1\times2^0$

64+0+0+0+0+0+1

65

ANKITA

ASCII: A: 65 N: 78 K: 75 I:73 T:84 A:65

A	N	K	I	T	A
65	78	75	73	84	65

1000001 1001110 1001011 1001001 1010100 1000001

100000110011101001011100100110101001000001

What is this?

100000110011101001011100100110101001000001
100000110011101001011100100110101001000001
100000110011101001011100100110101001000001
100000110011101001011100100110101001000001
100000110011101001011100100110101001000001
100000110011101001011100100110101001000001
100000110011101001011100100110101001000001

1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001
1000001	1001110	1001011	1001001	1010100	1000001

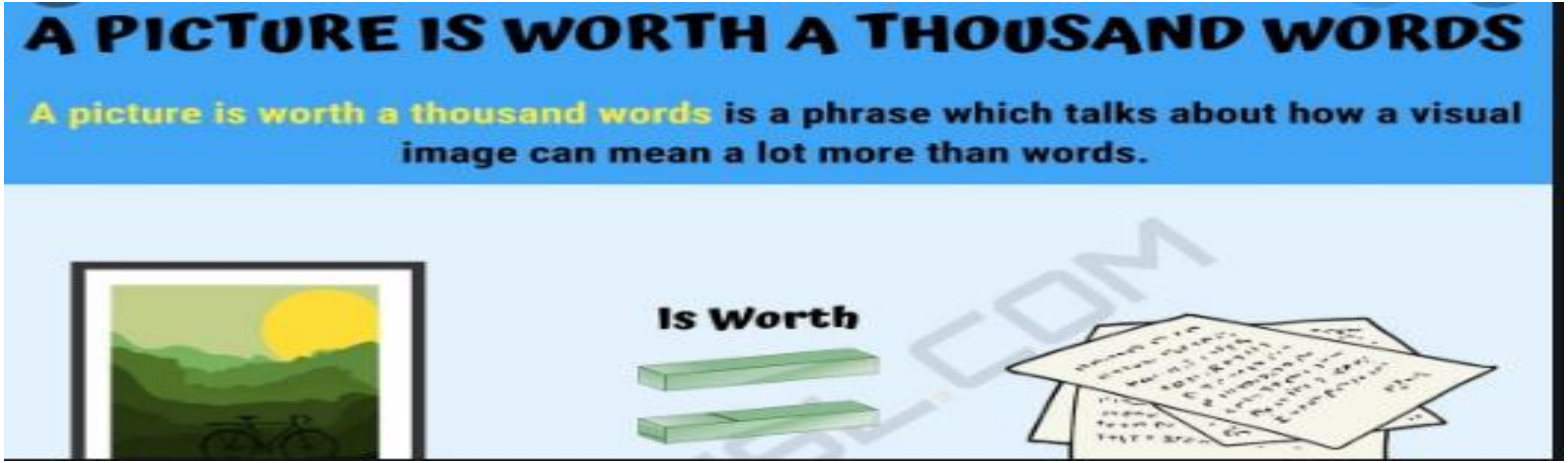
(a)

65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65
65	78	75	73	84	65

(b)

A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A
A	N	K	I	T	A

(c)


A PICTURE IS WORTH A THOUSAND WORDS
A picture is worth a thousand words is a phrase which talks about how a visual image can mean a lot more than words.
Is Worth

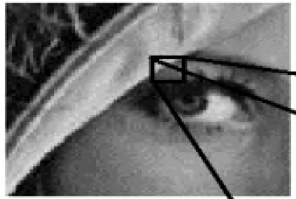
What is Digital Image?

- An image is a 2D matrix where each and every cell corresponds to a pixel value.

Or

- Digital Image is a two-dimensional function $f(x, y)$, x and y are spatial coordinates. The amplitude of f is called intensity or gray level at the point (x, y)

Matrix representation of an image



196	198	198	188	148	108
199	200	182	130	100	98
203	178	116	85	100	95
173	99	83	85	95	98
94	78	80	79	94	87
71	78	86	76	80	72

What is Digital Image Processing (DIP)?

Digital Image Processing

process digital images by means of computer, it covers low, mid, and high-level processes

Pixel

the elements of a digital image

What is DIP? (cont…)

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

Low Level Process	Mid Level Process	High Level Process
Input: Image **Output:** Image **Examples:** Noise removal, image sharpening	**Input:** Image **Output:** Attributes **Examples:** Object recognition, segmentation	**Input:** Attributes **Output:** Understanding **Examples:** Scene understanding, autonomous navigation

Topics of Course

- Image acquisition – (low-level) digital representation of the world scenes

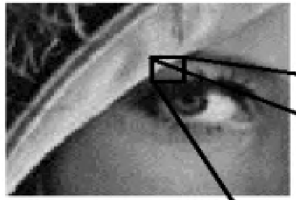
- Image processing – noise removal, smoothing, sharpening, contrast enhancement, alter the appearance of an image

- Image compression – efficiently represent image data for storage (save disk space) and communication (save network bandwidth)

What numbers?

How many numbers?

How large/small should the numbers be?



196	198	198	188	148	108
199	200	182	130	100	98
203	178	116	85	100	95
173	99	83	85	95	98
94	78	80	79	94	87
71	78	86	76	80	72

Image Processing

Image processing – noise removal, smoothing, sharpening, contrast enhancement, alter the appearance of an image

Noise removal

Image Processing

Image processing – noise removal, smoothing, sharpening, contrast enhancement, alter the appearance of an image

Sharpening

Image Processing

Image processing – noise removal, smoothing, sharpening, contrast enhancement, alter the appearance of an image

Blurring/smoothing

Image Processing

Image processing – noise removal, smoothing, sharpening, contrast enhancement, alter the appearance of an image

Contrast enhancement

Applications
&
Research Topics

Digital Image Watermarking

\+ 011111 =

Digital Image Compression

245,760 bytes

69,632 bytes

5,951 bytes

Image Fusion

Suggested Readings

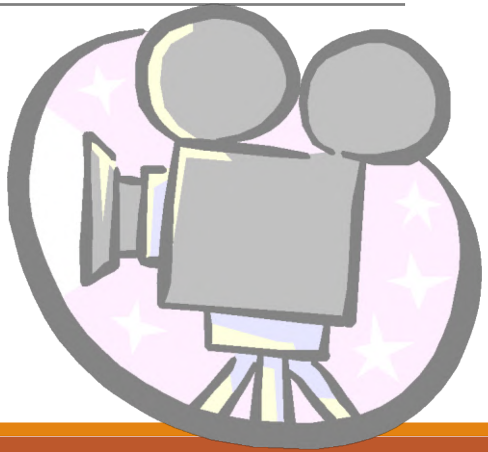
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Origin of Digital Image Processing (DIP)
- Fields that use DIP
- Fundamental steps in DIP

Origin of Digital Image Processing (DIP)

- First applications of digital images was in newspaper industry, when pictures were first sent by **submarine cable** between London and New York.

- In the early 1920s, **Bartlane cable picture transmission system**, reduces the transmission time across Atlantic from more than a week to less than three hours.

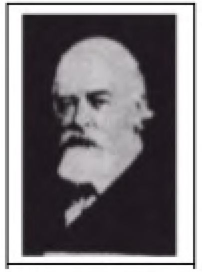
- The transmitted pictures can be reproduced using telegraph printer with special typeface.

- This method was abandoned towards the end of 1921, in favor of a technique based on photographic reproduction made from tapes

Origin of Digital Image Processing (DIP) (Cont..)

- Early **Bartlane cable picture transmission systems** were capable of coding images in five distinct levels of gray.
- This capacity was increased to 15 levels in 1929.
- Although the examples just cited involve digital images not digital image processing because computers were not involved in their creation.
- The progress in the field of DIP has been dependent on the development of digital computers and the supporting technologies that include data storage, display and transmission.

Fig. 1 Digital picture produced in 1921 from a coded tape by telegraph printer with special type faces



Digital picture made in 1922

Fig. 2 Untouched picture of Generals Pershing And Foch transmitted in 1929 between London and New York by 15 tone equipment

Origin of Digital Image Processing (DIP) (Cont..)

- The basis of modern digital computer dates back to only the 1940s with the introduction by John von Neumann of two key concepts:
  - A memory to hold a stored program and data.
  - Conditional branching
  - These two idea are the foundation of CPU.

Origin of Digital Image Processing (DIP) (Cont..)

- The key advances that led to computers powerful enough to be used in DIP are:
  - Invention of Transistor at Bell Lab in 1948.
  - High level Programming languages in 1950s to 1960s.
  - Invention of Ics at Texas in 1958s.
  - Development of OS in 1960s.
  - Development of microprocessor by Intel in the early 1970s.
  - Introduction of personal computer by IBM in 1981.
  - Large scale Ics in late 1970s
  - VLSI Ics in 1980s (Ultra large scale integration (ULSI) present)

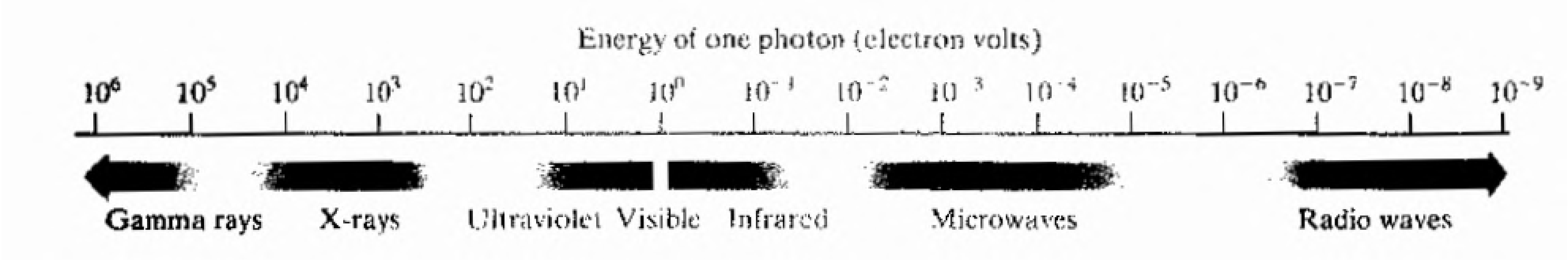
Origin of Digital Image Processing (DIP) (Cont..)

- First Computer was powerful enough to carry image processing task appeared in the early 1960s.
- First picture of moon transmitted by U.S. Spacecraft. Ranger 7 took this picture on July 31 1964.

- In late 1960s and early 1970s DIP is used in medical imaging, remote earth resources observation, and astronomy.
- From late 1960s to present, the field of image processing has grown vigorously.

Fields that use DIP

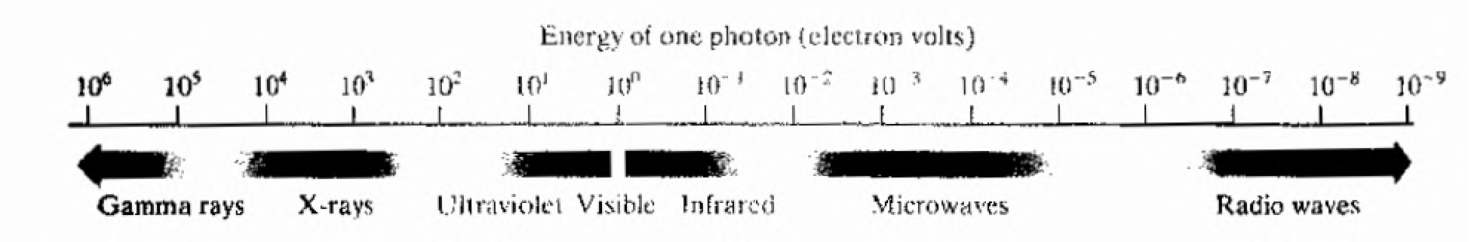
- One of the simplest ways to develop a basic understanding of the extent of image processing applications is to categorize images according to source:



Electromagnetic spectrum arranged according to energy per photon

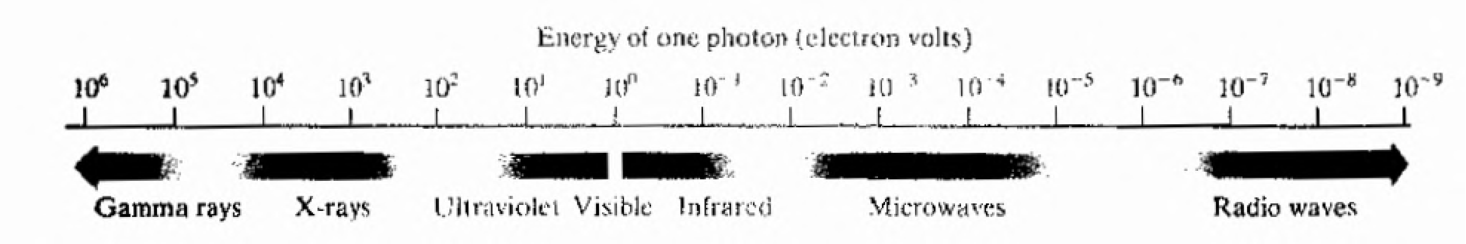
The bands are shown shaded to convey the fact that bands of electromagnetic spectrum are not distinct but rather transition smoothly from one to the other.

Gamma-Ray Imaging



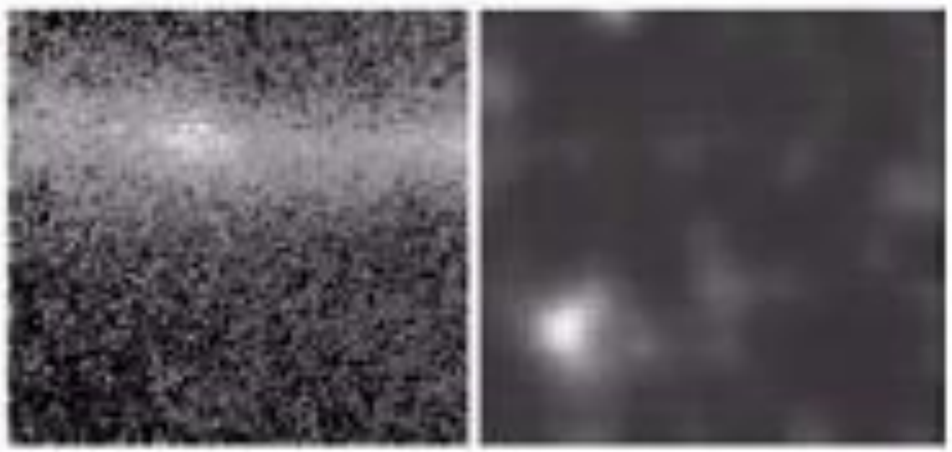
- Nuclear Medicine: Major use of imaging based on Gamma rays include nuclear medicine and astronomical observations.
  - Approach is to inject a patient with radioactive isotope that emits gamma rays as it decays
  - Images are produce from the emissions collected by gamma ray detector.
- Positron-Emission-Tomography (PET)
  - patient given radioactive isotope that emits positrons as it decays
  - When positron meets electron ,both destroyed and two gamma rays given off
  - Gamma rays are detected and using special detectors and image is constructed

Gamma-Ray Imaging



1(a) 1(b)

(a) Locate sites of bone pathology, such as infection or tumors
(b) This image shows a tumor in the brain and one in the lung

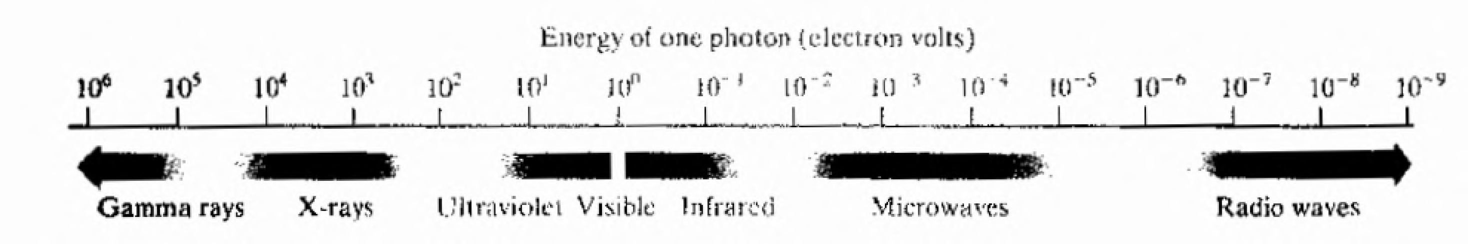


2(a) 2(b)

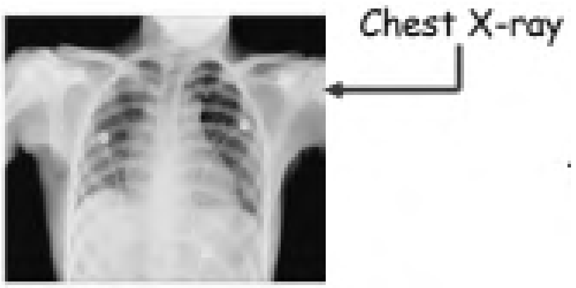
(a) Cygnus loop in the gamma ray
(b) This image shows gamma radiation from a valve in a nuclear reactor.

*star in the constellation of Cygnus exploded about 15000 years ago, generating a superheated gas cloud (known as Cygnus loop)

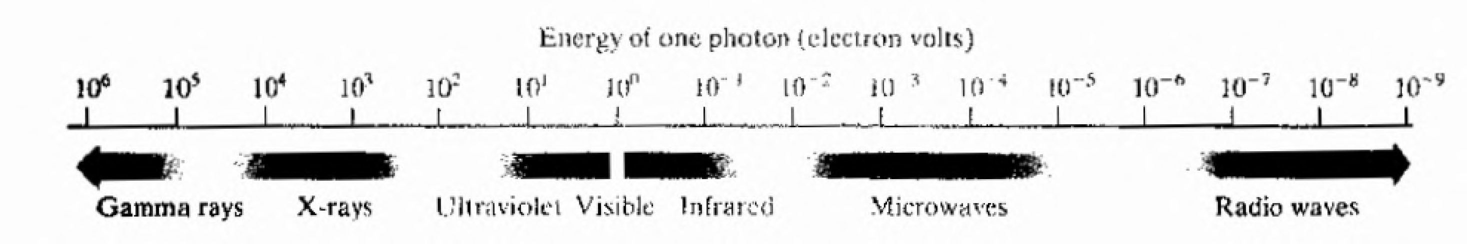
X-Ray Imaging



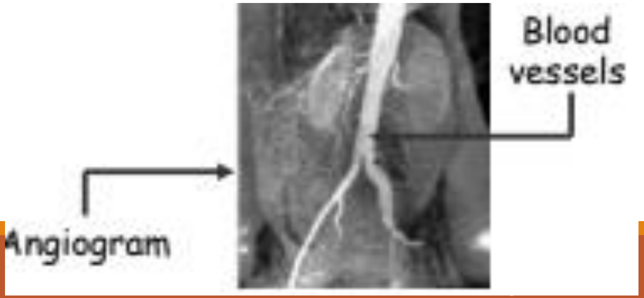
- X Ray Imaging: Oldest Sources of EM Radiation for Imaging
  - Best known for medical diagnostics
    - Patient placed between “X-ray tube” and special film sensitive to X-ray radiation
    - Electrons are emitted from X-ray tube and go through patient
    - Intensity of X-rays is modified by absorption as they go through patient
    - Intensity collected at film and image is then created

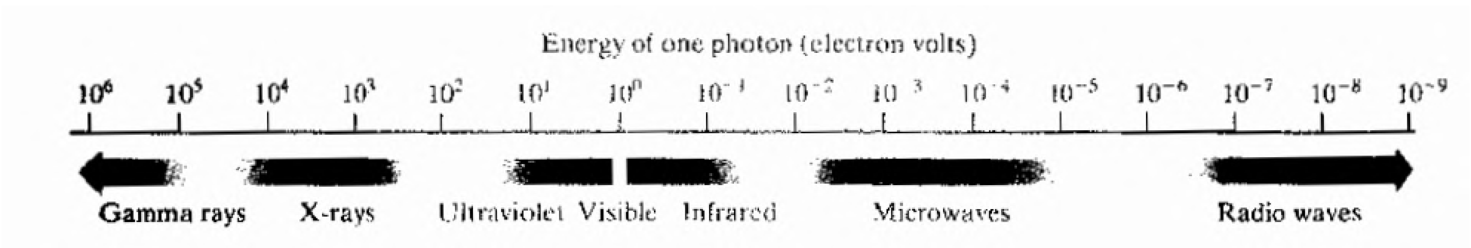


X-Ray Imaging



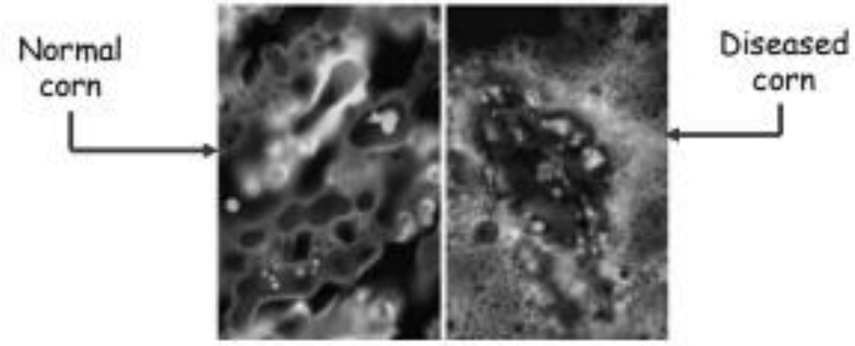
- Other Applications of X-ray Imaging
  - Computerized axial tomography (CAT scan)
  - Angiography: is another major application in an area called contrast enhancement radiography.
    - Obtain images of blood vessels (angiograms)
    - A catheter (a small, flexible, hollow tube) is inserted, for example into an artery or vein.
    - X-ray contrast medium injected via catheter at appropriate location
    - X-ray image obtained and blood vessels highlighted





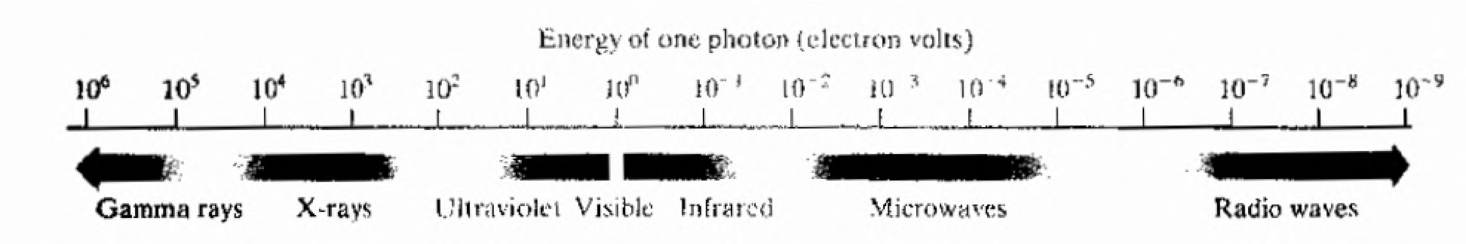
Imaging in the Ultraviolet Band

- Varied Applications
  - Industrial inspection
  - Microscopy → fluorescence microscopy one of the fastest growing fields of microscopy Lasers
  - Biological imaging
  - Astronomical observation



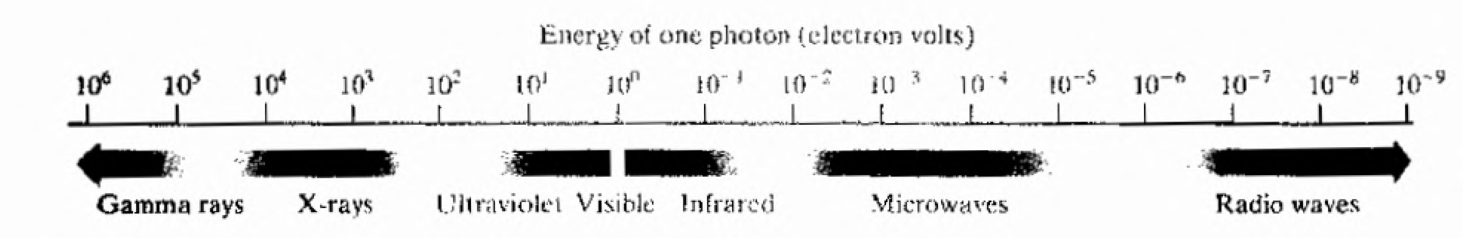
Imaging in the Visible and Infrared Bands

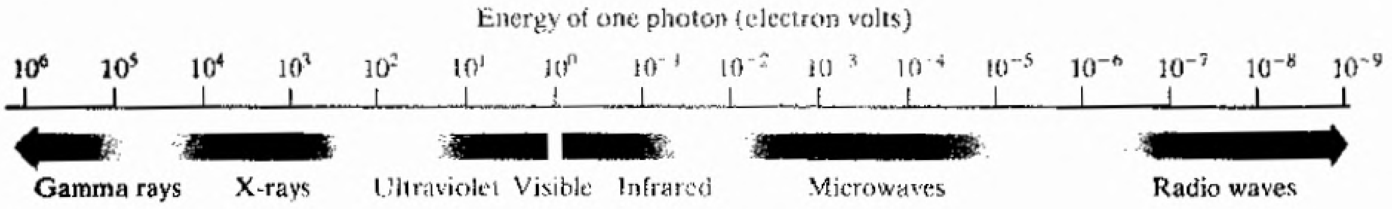
- Visual band of the spectrum is the most familiar in all our activities.
- Infrared band often used in conjunction with visual imaging.
- Various applications
  - Light microscopy, Astronomy, Industrial applications, Remote sensing



Imaging in the Visible and Infrared Bands

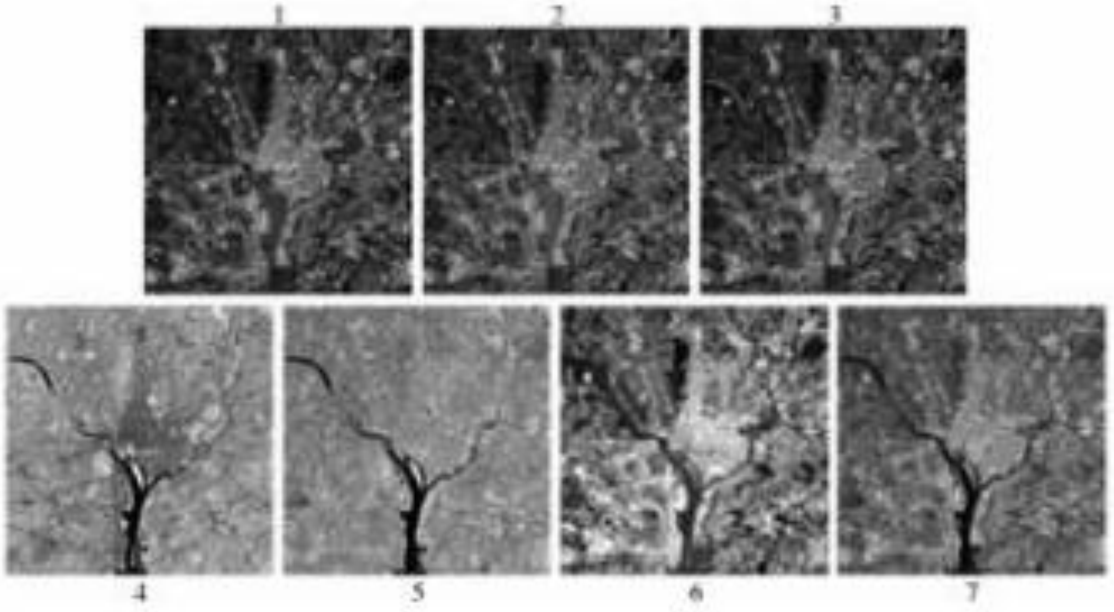
- Remote Sensing Definition: The process of obtaining data or images from a distance, as from satellites or aircraft
- Major area of visual/infrared imaging , usually covers several bands of the visual/infrared spectrum
- NASA's LANDSAT satellite
  - Primary purpose → Obtain and transmit images of earth from space for environmental monitoring purposes

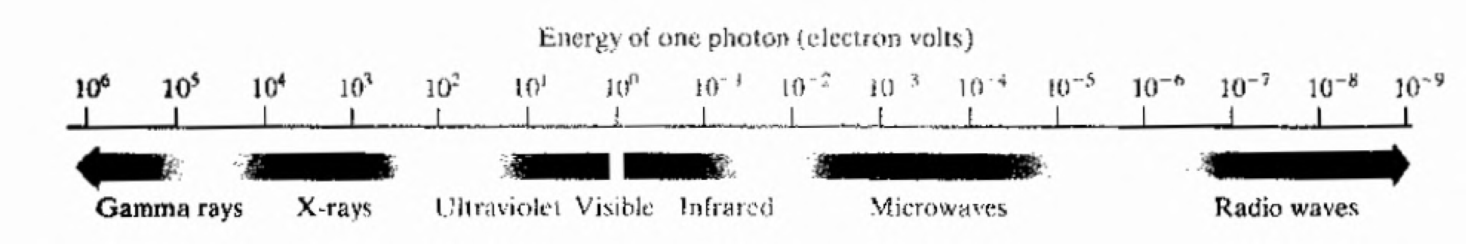




Imaging in the Visible and Infrared Bands

- ➢Images Obtained from LANDSAT
  - ➢Detect vegetation, roads, rivers, buildings etc

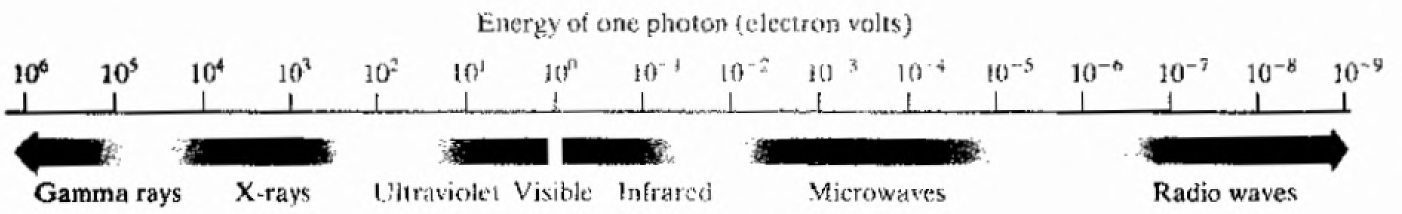




Imaging in the Microwave Band

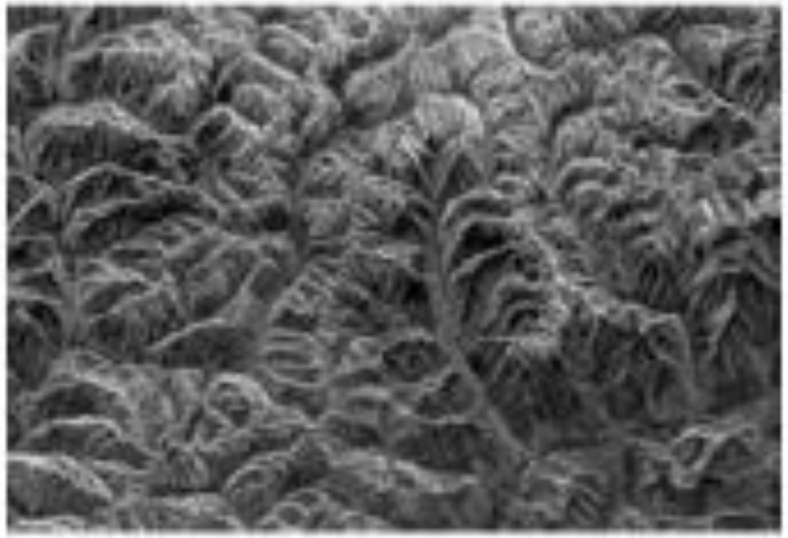
- Dominant Use is Radar
  - Ability to collect data over virtually any region, at any time, regardless of weather conditions or ambient light conditions
  - Penetrate clouds At times, can see through vegetation, ice, sand…

- Operates similar to flash camera
  - Provides its own illumination (microwave pulses) to illuminate area of interest and then "snaps" image
  - Instead of camera lens, antenna is used

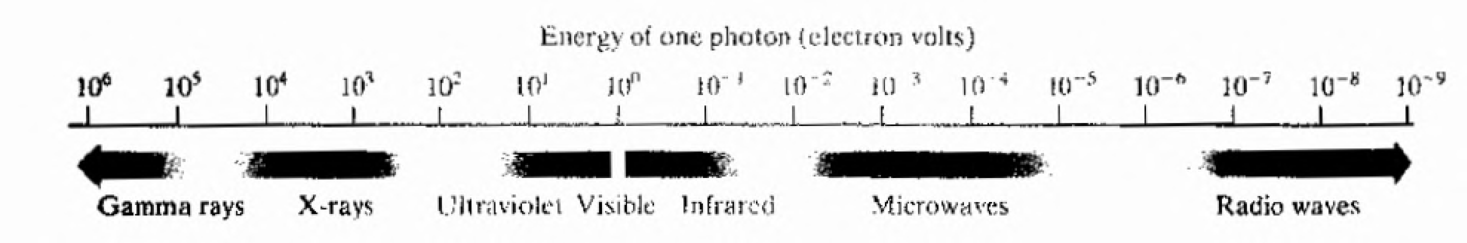


Imaging in the Microwave Band

- Image of mountainous region of Tibet obtained from space satellite



Imaging in Radio Band



- Dominant use is Medicine and Astronomy

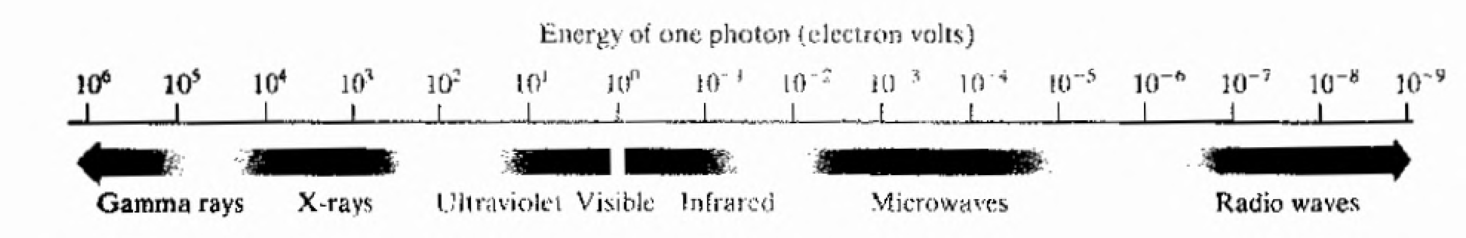
- In medicine, popular technique is magnetic resonance imagine (MRI)

- Patient placed in powerful magnet Radio waves are passed through patient's body in short pulses

- Each pulse causes another pulse to be emitted by the patients tissues

- Location and strength of the pulses is determined by computer and 2D image is created based on this information

Imaging in Radio Band



MRI image of a human knee and spine

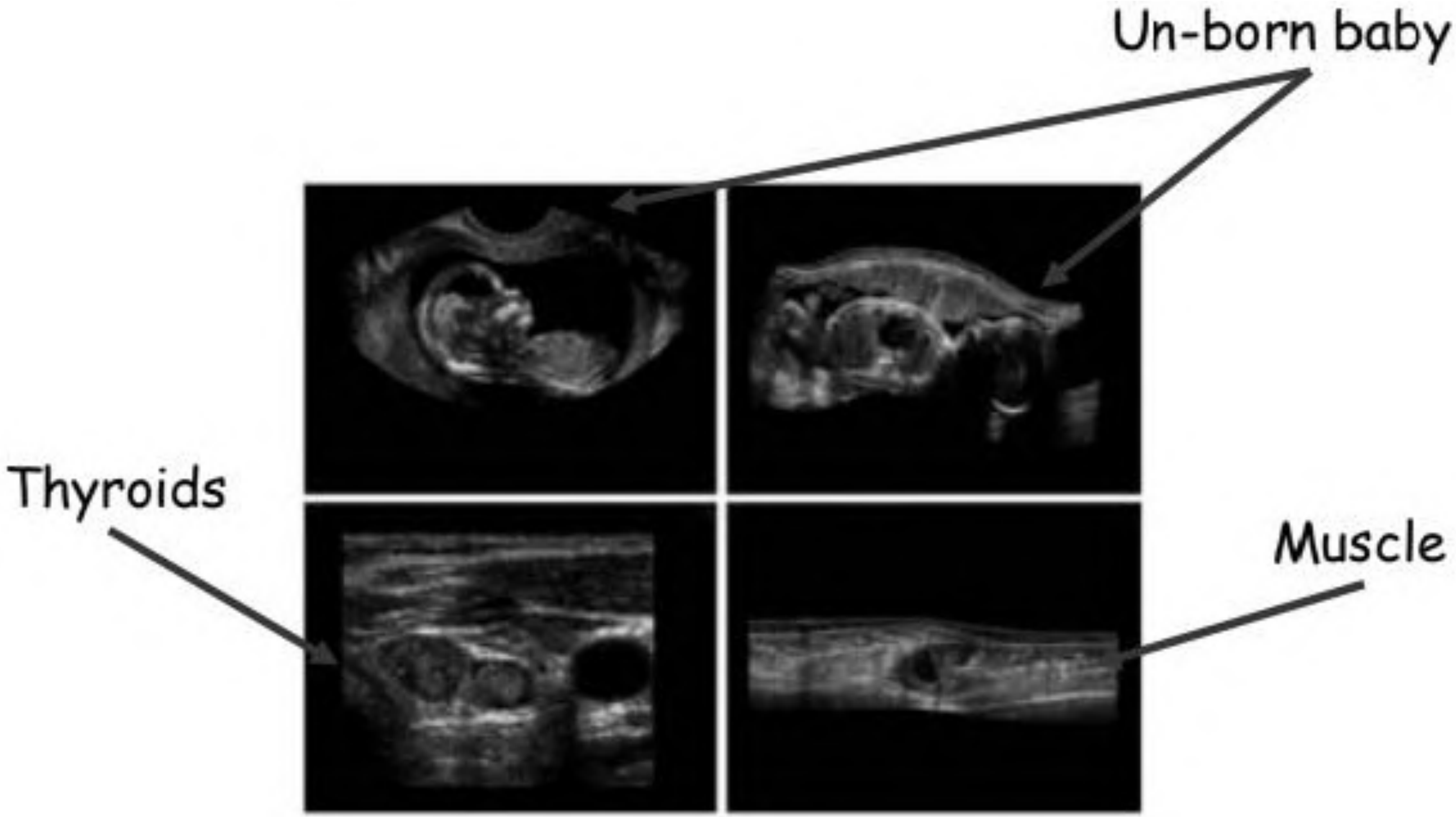
Other Imaging Modalities

- Acoustical Imaging
  - Sound waves (typically low frequency, e.g., < 100Hz) are emitted from transmitter
  - Reflections of transmitted sound recorded by receiver
  - Image constructed based on time of arrival and intensity of echoes
  - Many applications
  - Geological exploration (oil and mineral exploration)
  - Industry
  - Medicine (ultrasound)

Other Imaging Modalities

- Acoustical Imaging (cont…)
  - Popular use of acoustical imaging is ultrasound
  - Viewing of unborn babies
  - Viewing other body tissues/bones
  - Can detect certain cancers
- To construct typical ultrasound image, millions of pulses and echoes are emitted and received respectively each second
- Pulses typically 1 – 5 MHz

Other Imaging Modalities



Suggested Readings

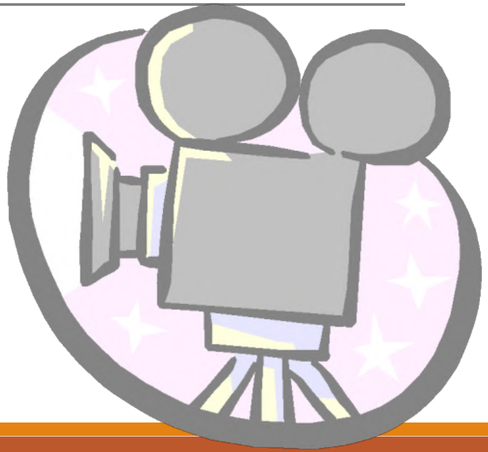
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

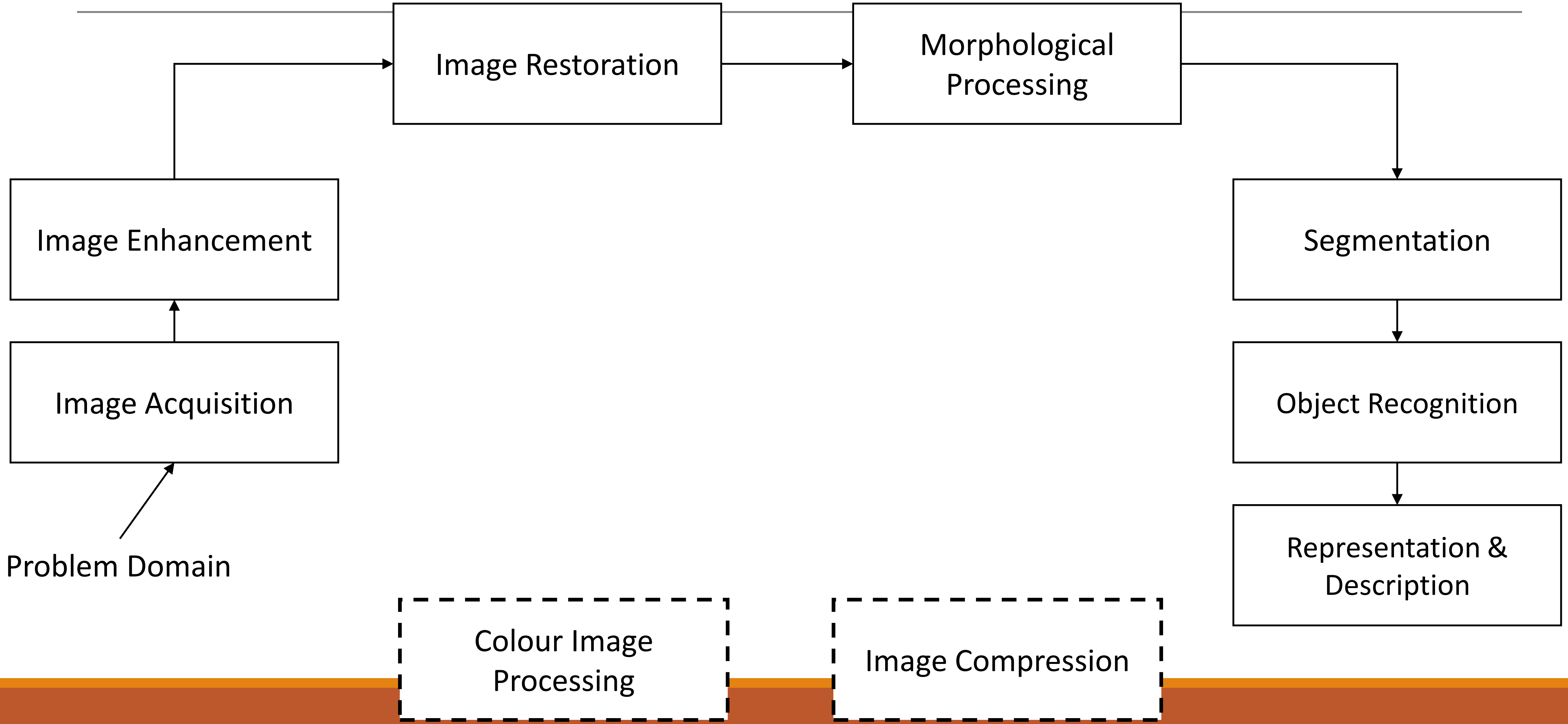
- Fundamental steps in DIP
- Elements of Visual Perception
  - Structure of Human Eye

Key Stages in Digital Image Processing

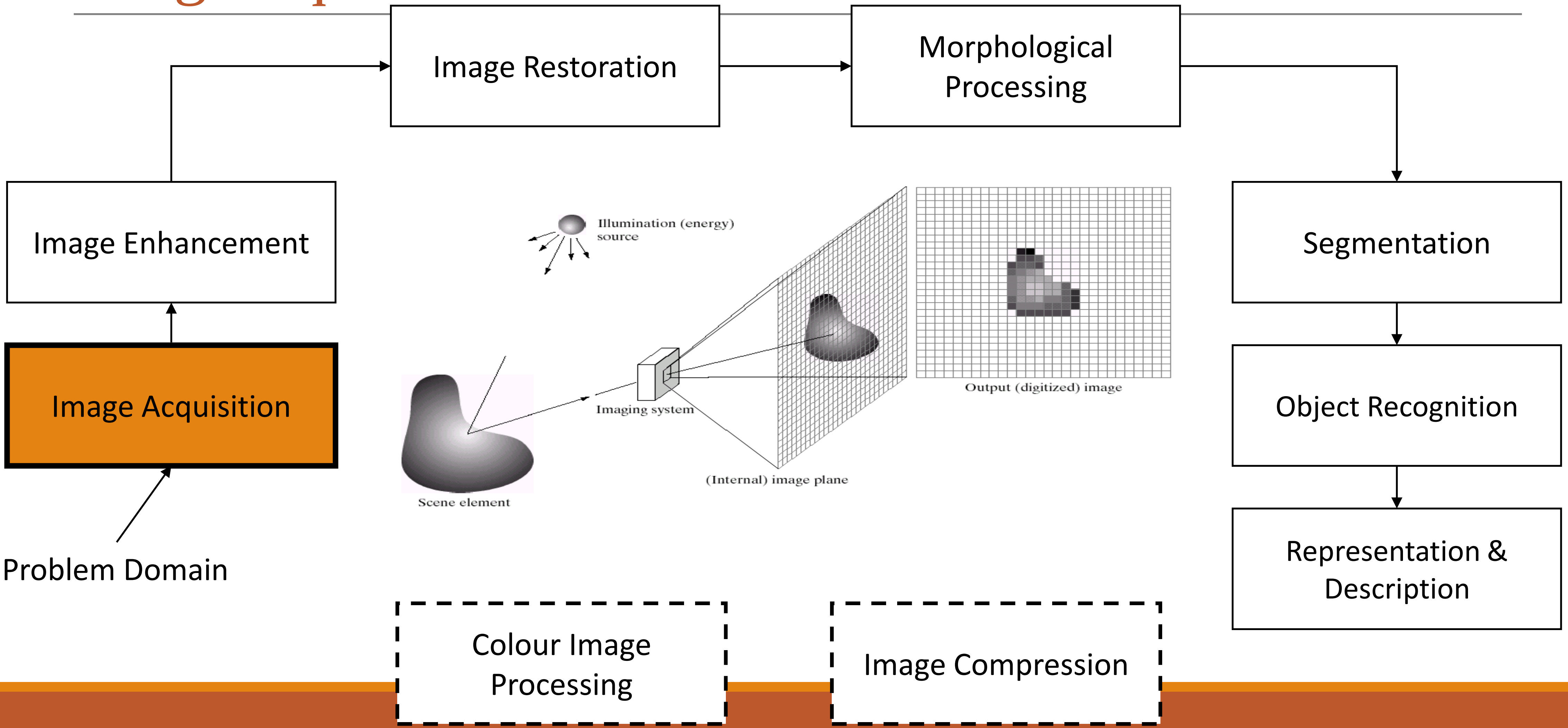
- ➢Methods Whose Input and Output are Images

- ➢ Methods Whose Inputs are Images but Outputs are Attributes, extracted from these Images

Key Stages in Digital Image Processing

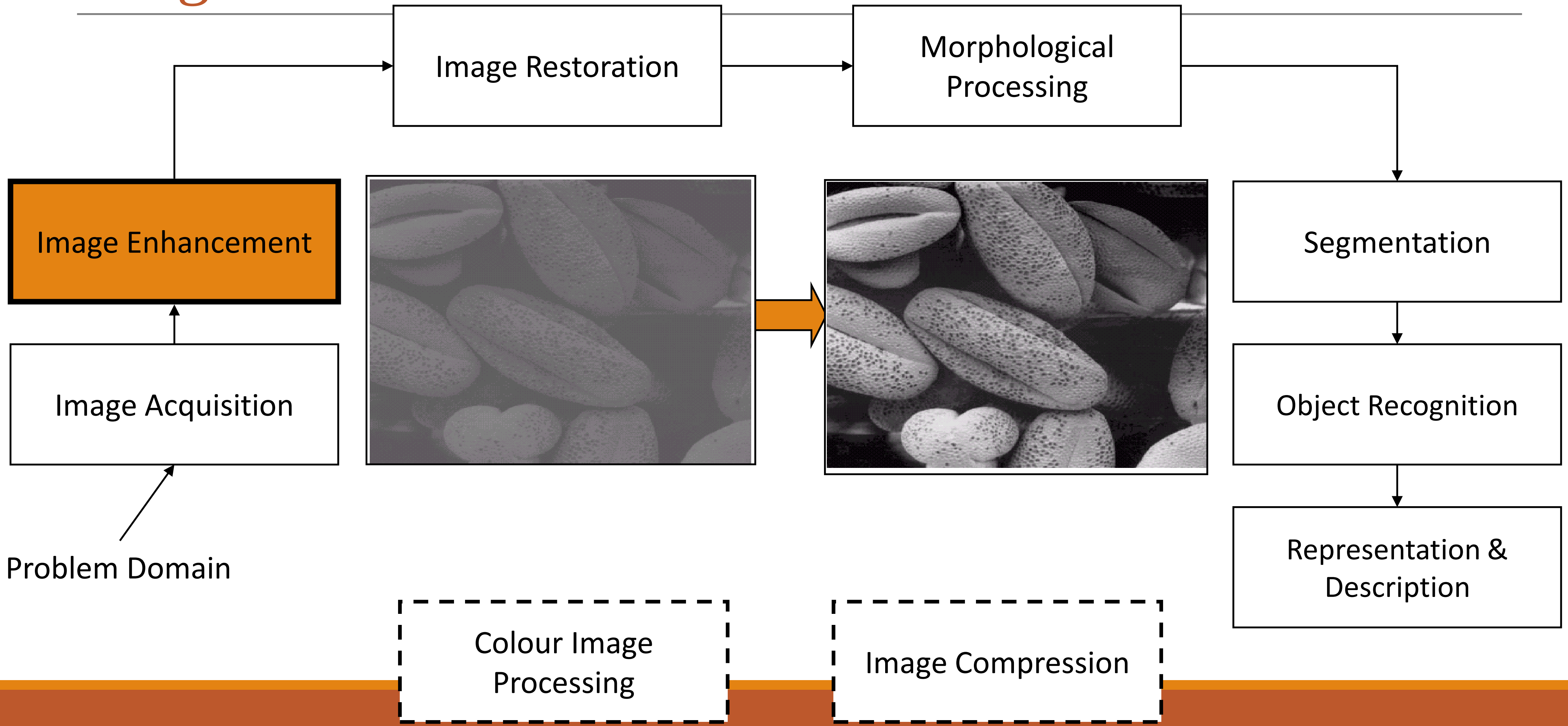


Key Stages in Digital Image Processing: Image Aquisition



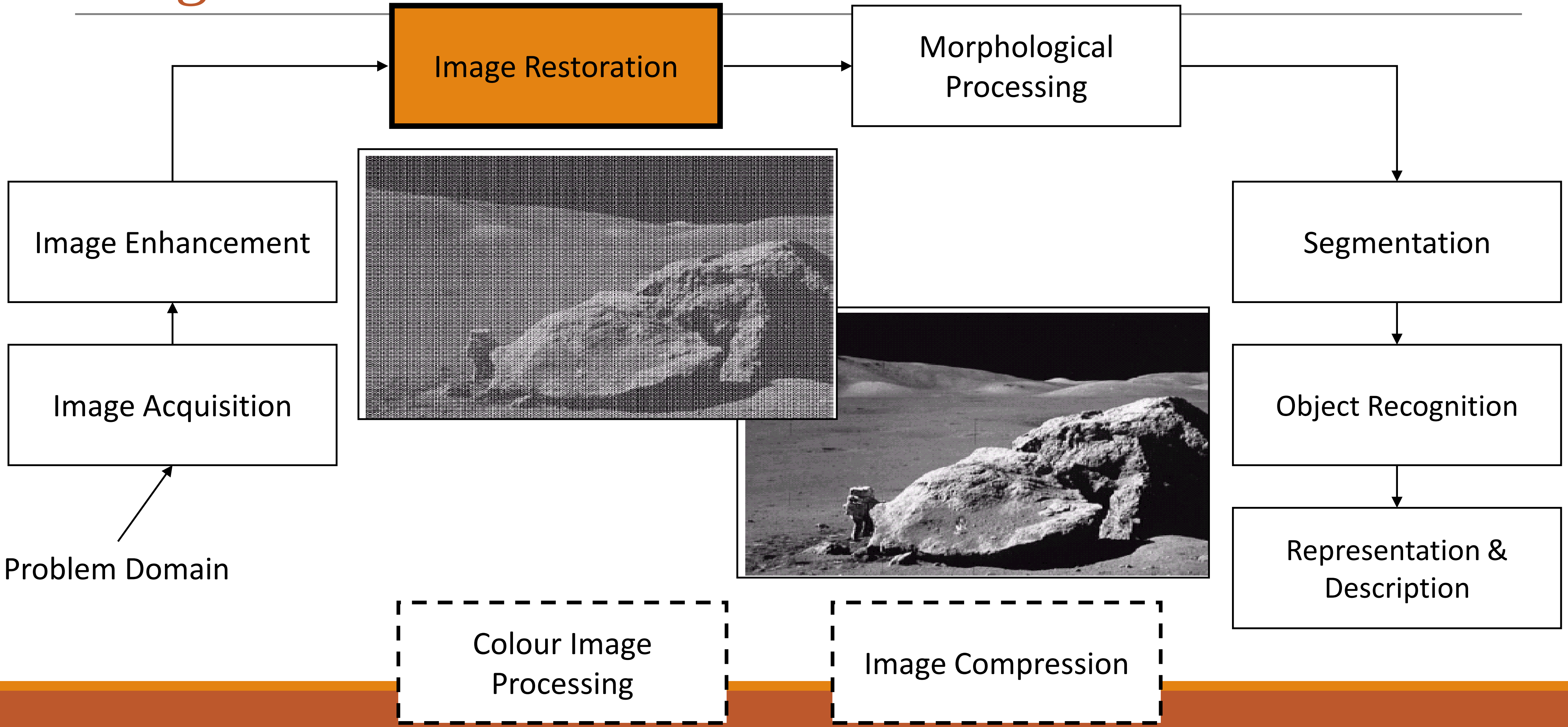
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Image Enhancement



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Image Restoration



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

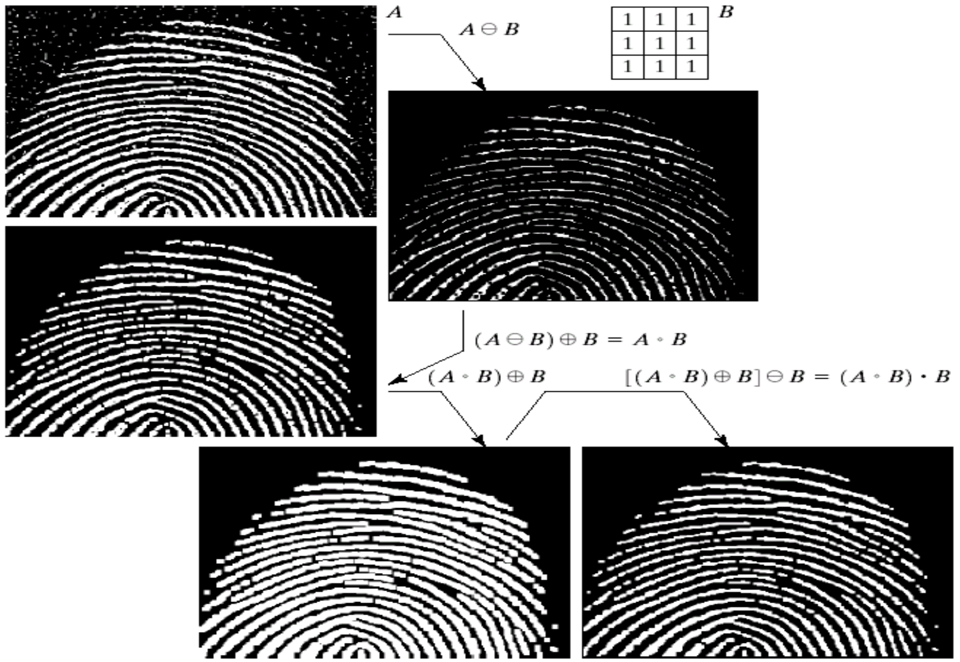
Key Stages in Digital Image Processing: Morphological Processing

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

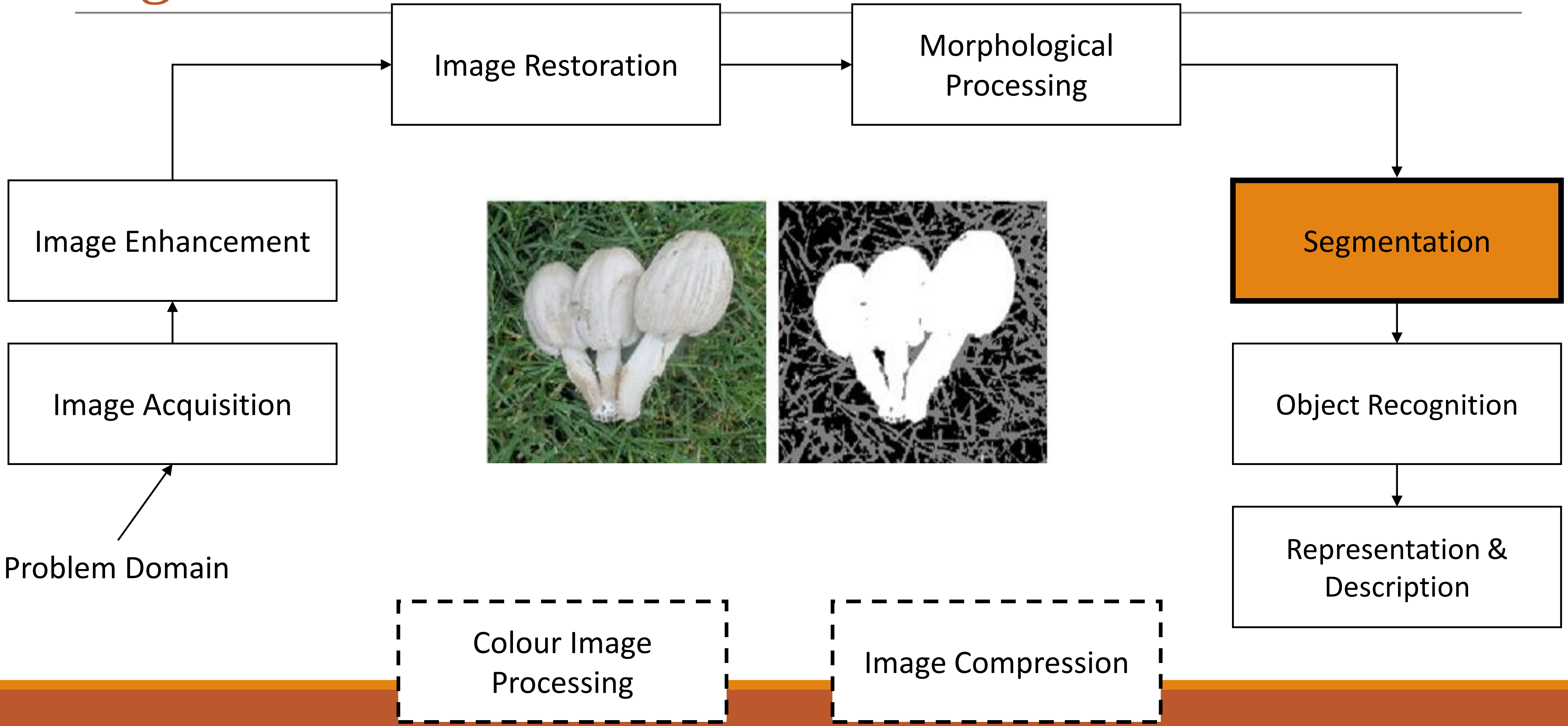
Problem Domain → Image Acquisition → Image Enhancement → Image Restoration → **Morphological Processing** → Segmentation → Object Recognition → Representation & Description

Colour Image Processing

Image Compression



Key Stages in Digital Image Processing: Segmentation



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Object Recognition

Image Restoration

Morphological Processing

Image Enhancement

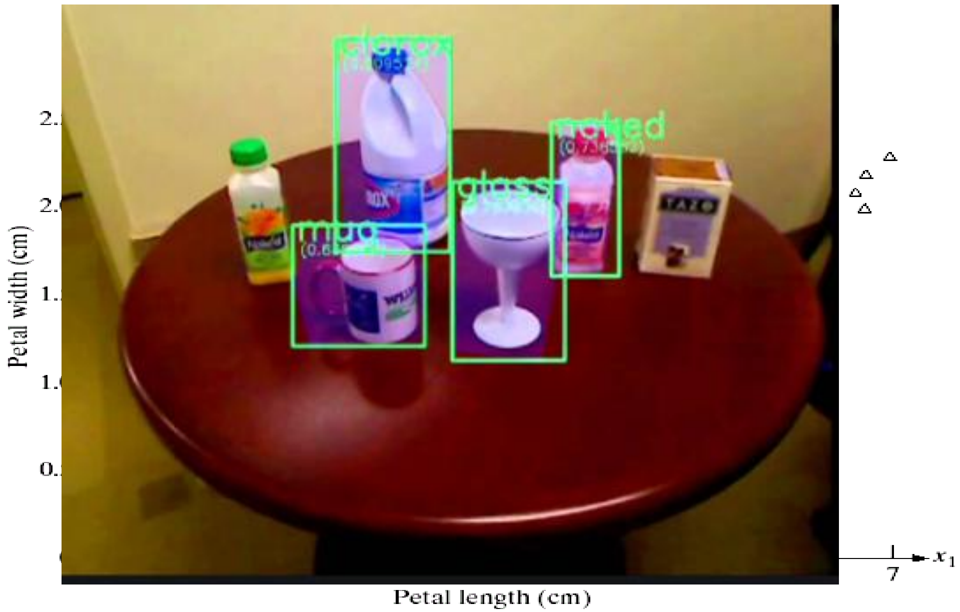
Segmentation

Image Acquisition

Object Recognition

Problem Domain

Representation & Description

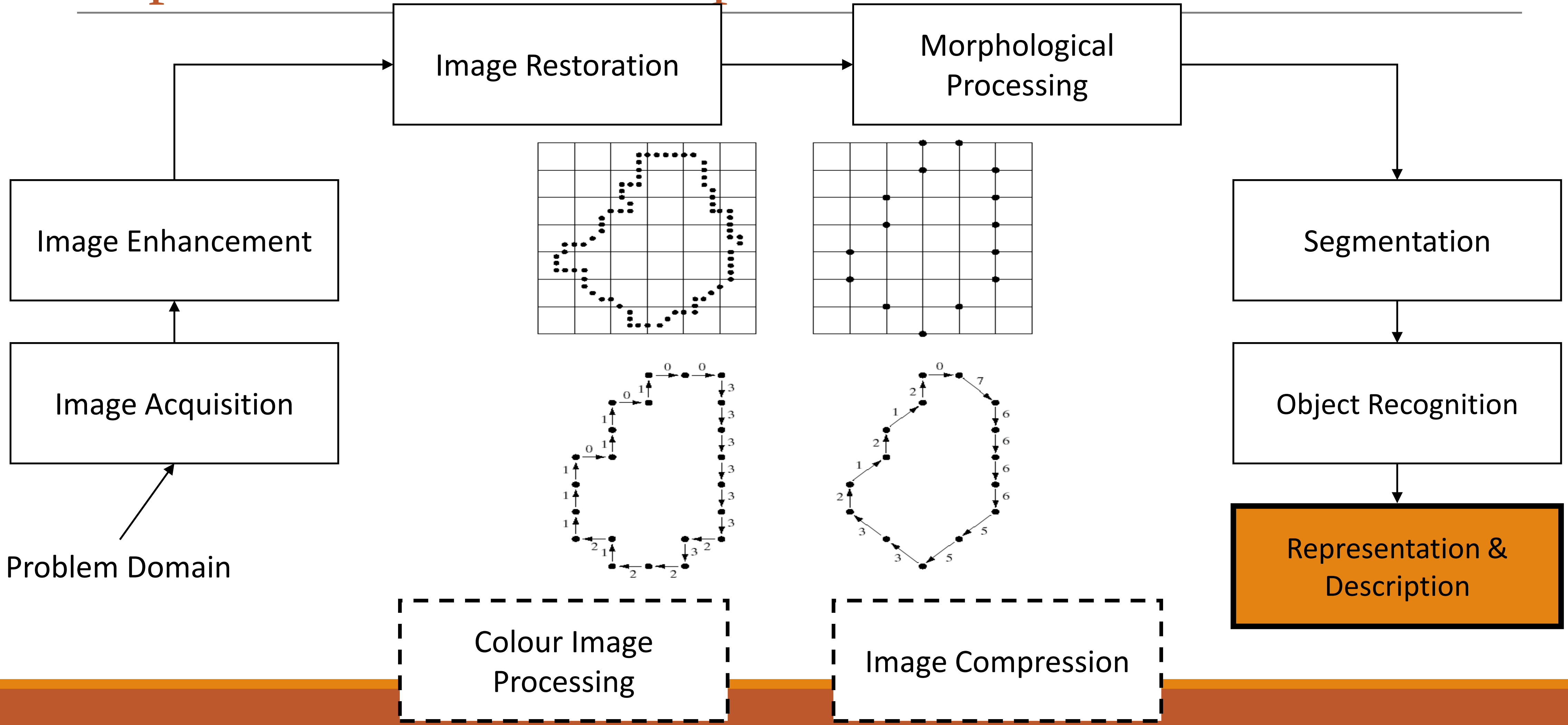


Colour Image Processing

Image Compression

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Representation & Description



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Image Compression

Image Restoration

Morphological Processing

Image Enhancement

Segmentation

62,632 bytes

5914 bytes

Image Acquisition

Object Recognition

Problem Domain

Representation & Description

Colour Image Processing

Image Compression

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

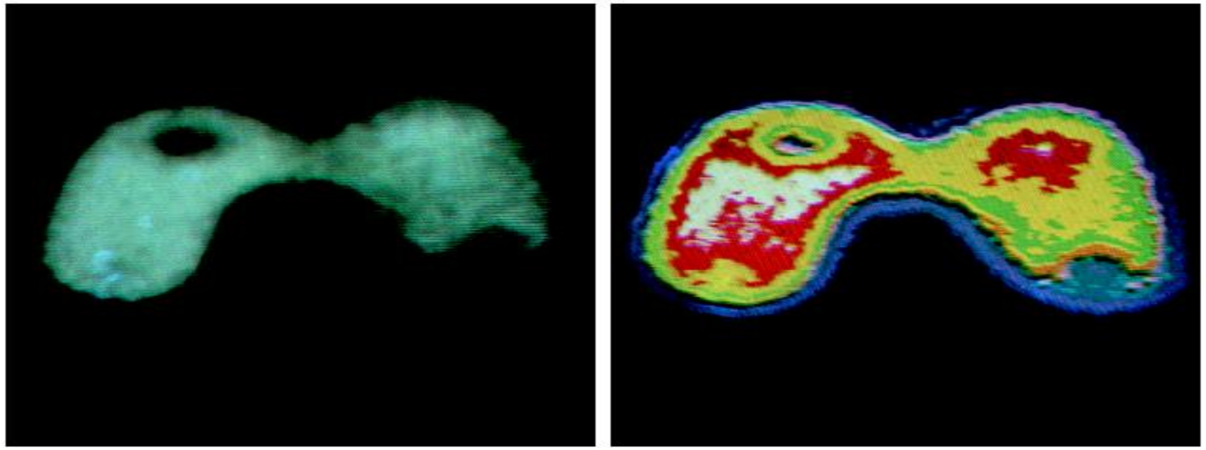
Key Stages in Digital Image Processing: Colour Image Processing

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Restoration

Morphological Processing

Image Enhancement



Segmentation

Image Acquisition

Object Recognition

Problem Domain

Representation & Description

Colour Image Processing

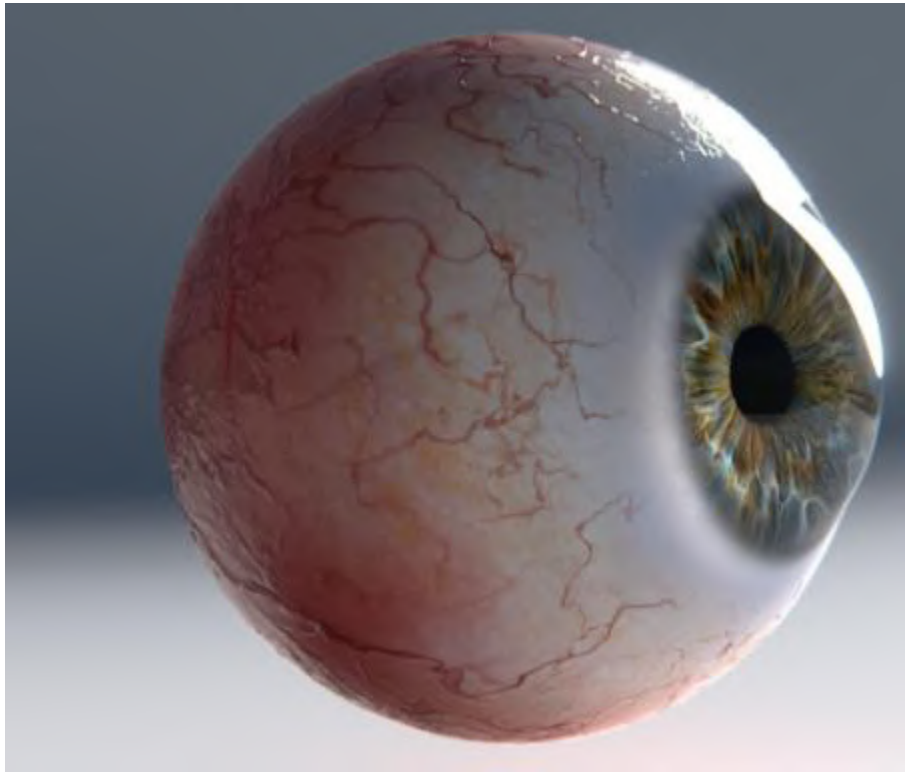
Image Compression

Structure of Human Eye

- Shape is nearly sphere
- Average diameter = 20 mm

•Consists of 3 membranes:

- Cornea and sclera : Outer Cover
- Choroid
- Retina- enclose the eye



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

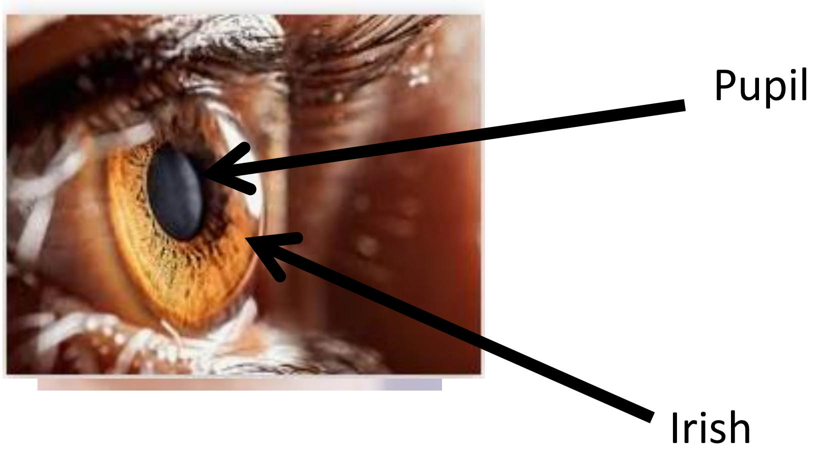
Structure of Human Eye

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Brown eye, Blue eye?

Iris have pigment, which reflects specific type of light

Black spot on the iris is called pupil, through which light enters in the eye.

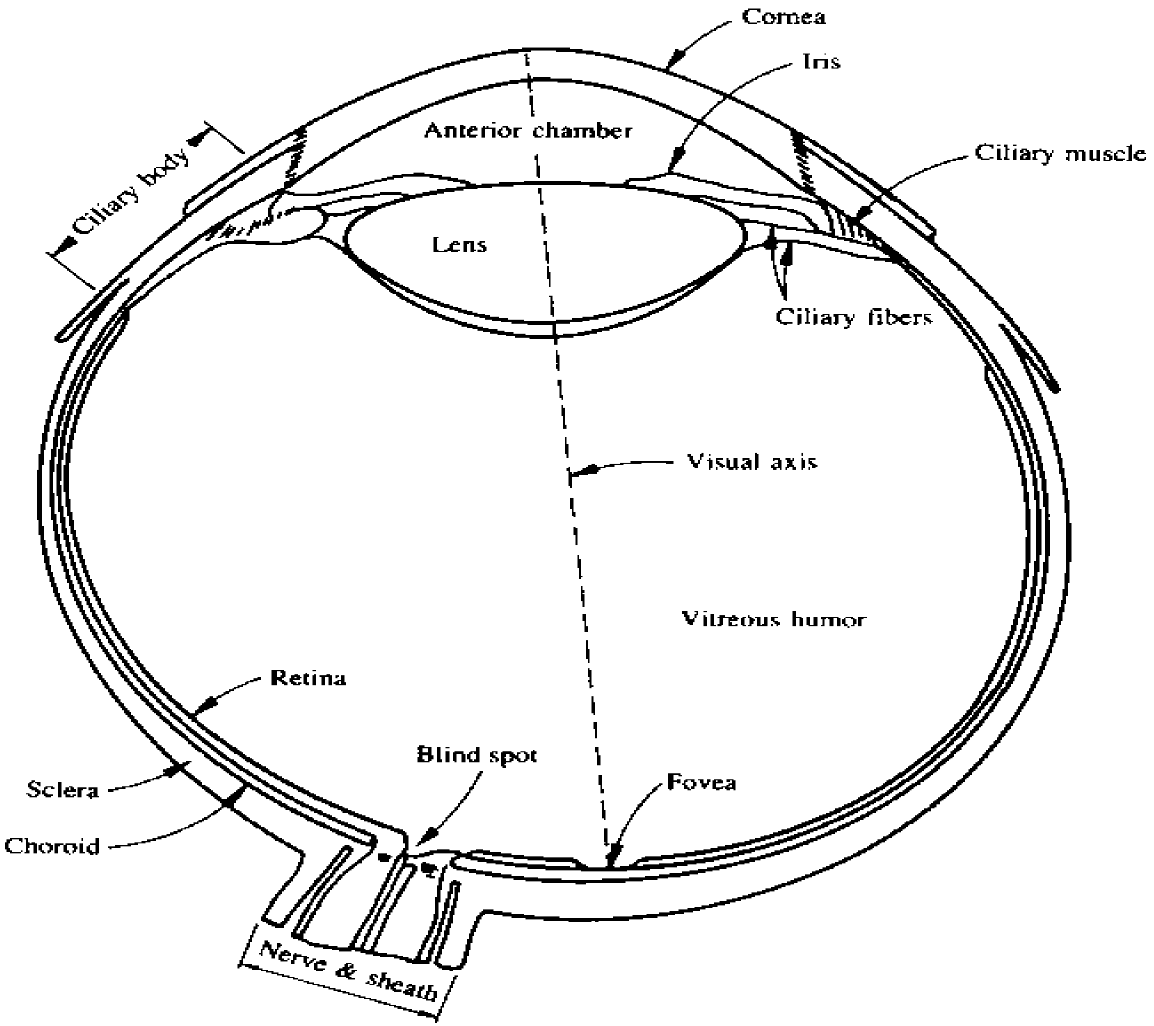


- Iris: contracts or expands to control amount of light
- Pupil: central opening of iris, 2 to 8 mm in diameter

High absorption in infrared and ultraviolet (can cause damage to eye)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

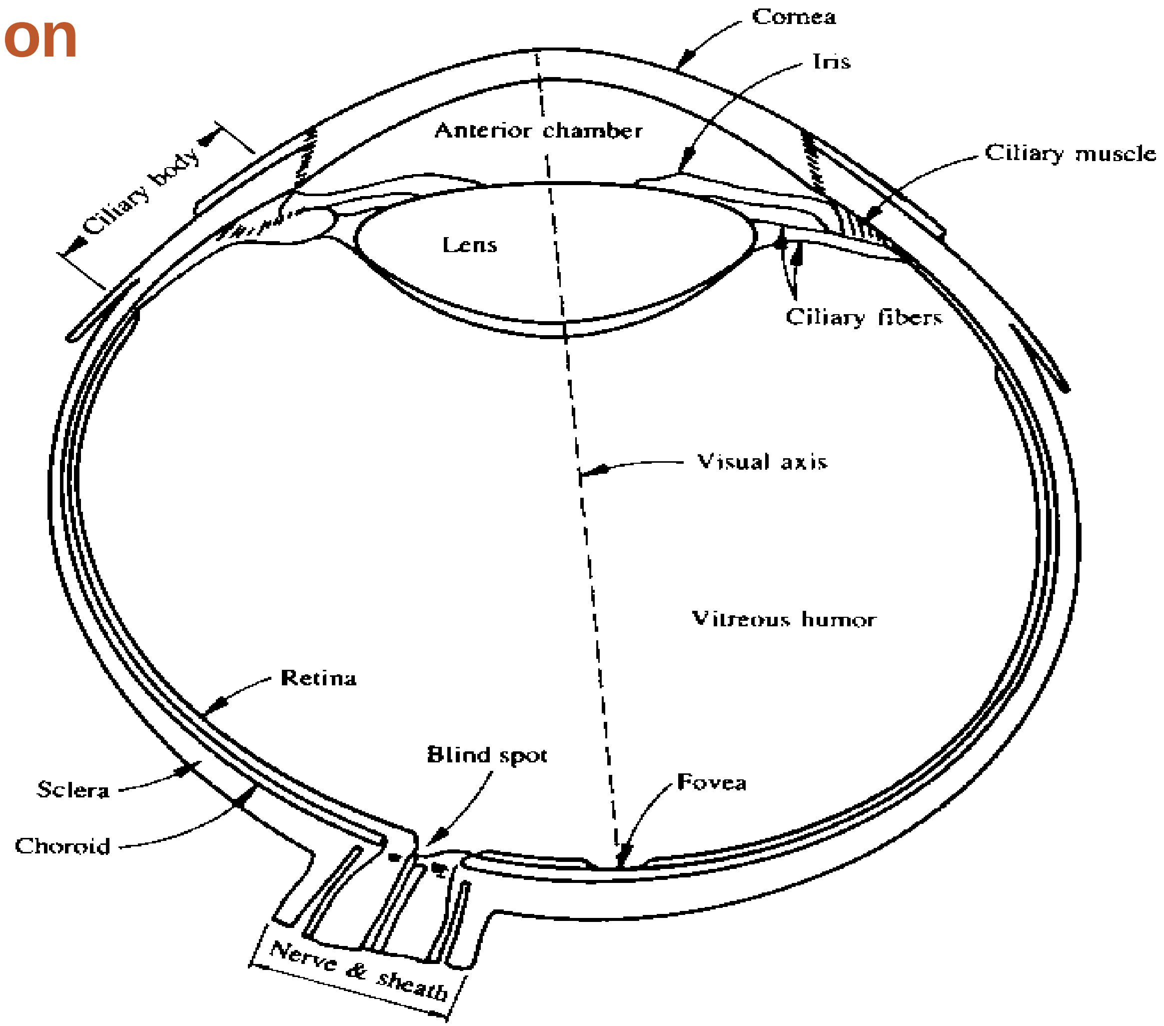
- Shape is nearly sphere
- Average diameter = 20 mm
- Consists of 3 membranes:
- Cornea and sclera : Outer Cover
- Choroid
- Retina- enclose the eye



Elements of Visual Perception

- **Cornea :**
  -tough, transparent tissue, covers the anterior surface of the eye.
- **Sclera :**
  -Opaque membrane, encloses the remainder of the optic globe.
- **Choroid :**
  -Lies below the sclera, contains network of blood vessels that serve as the major source of nutrition to the eye.

  -Choroid coat is heavily pigmented and hence helps to reduce the amount of extraneous light entering the eye and the backscatter within he optical globe.



Elements of Visual Perception

•

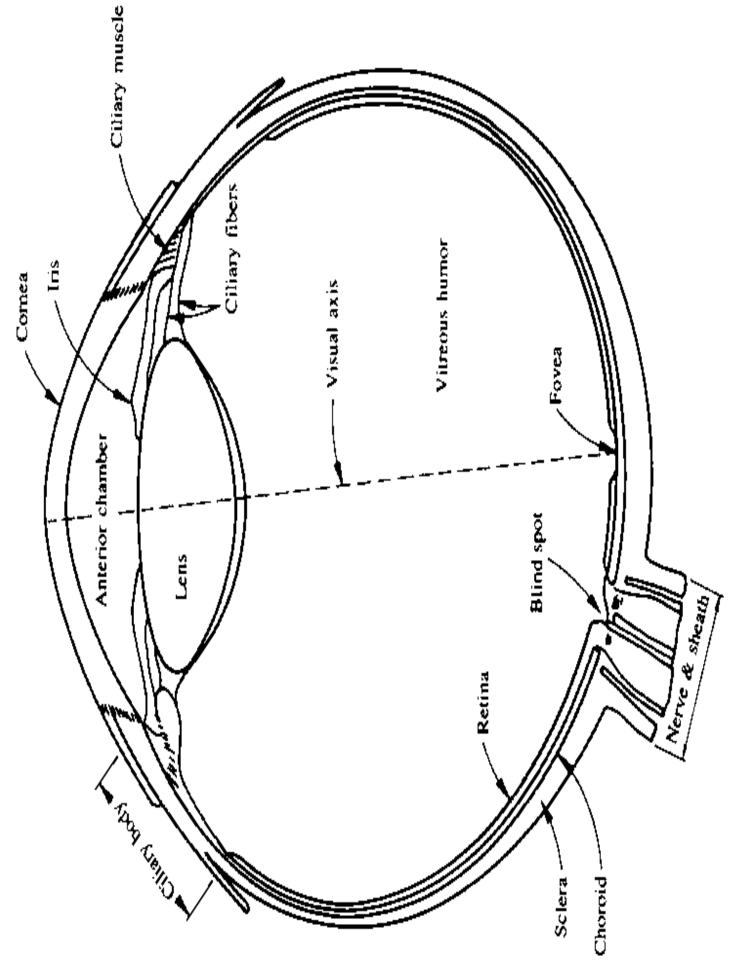
•Lens focus the image on retina

•The area between Cornea and lens is filled with water called aqueous humor.

•Maximum refraction take place at Cornea, when light enters from air to Cornea then through lens an image is formed on retina.

Retina is made of light sensitive cells which activates when light falls on it.

This light is further converted into electrical signals, which reaches to brain through optic nerves.

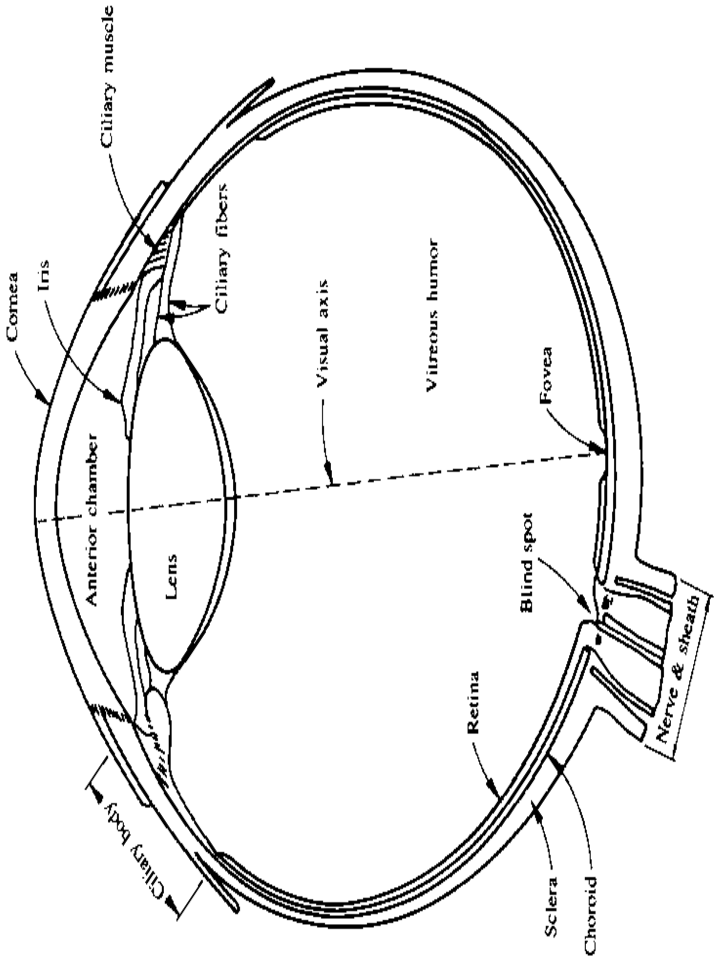


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Elements of Visual Perception

The portion between retina and lens is called Vitreous humor (jelly or glassy), it provides support to eye ball.

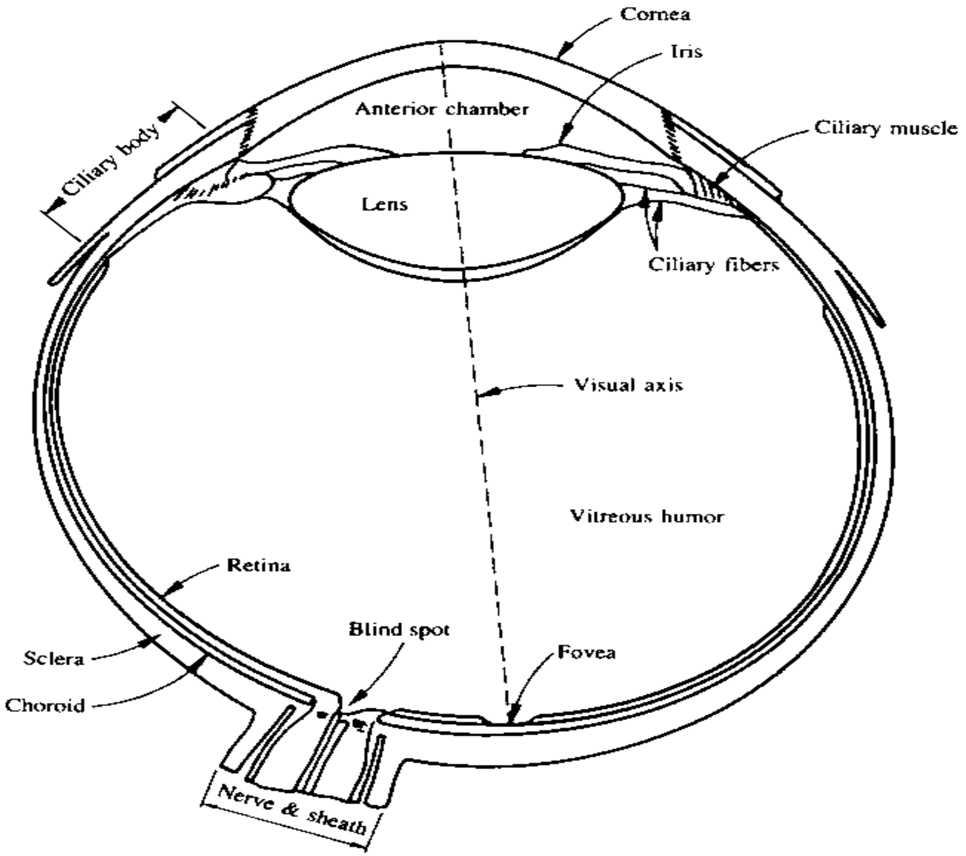
Cilliary muscle can change the shape of lens.



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Elements of Visual Perception

- **Lens:**
  - Focuses light on retina.
  - Contains 60% to 70% water.
  - Absorbs 8% of visible light.
  - High absorption in infrared and ultraviolet (can cause damage to eye).
- **Retina:**
  - The inner most layer, covers the posteriori portion of eye.
  - When eye is properly focused, light of an object is imaged on the retina.
  - Light receptors are distributed over the surface of retina.



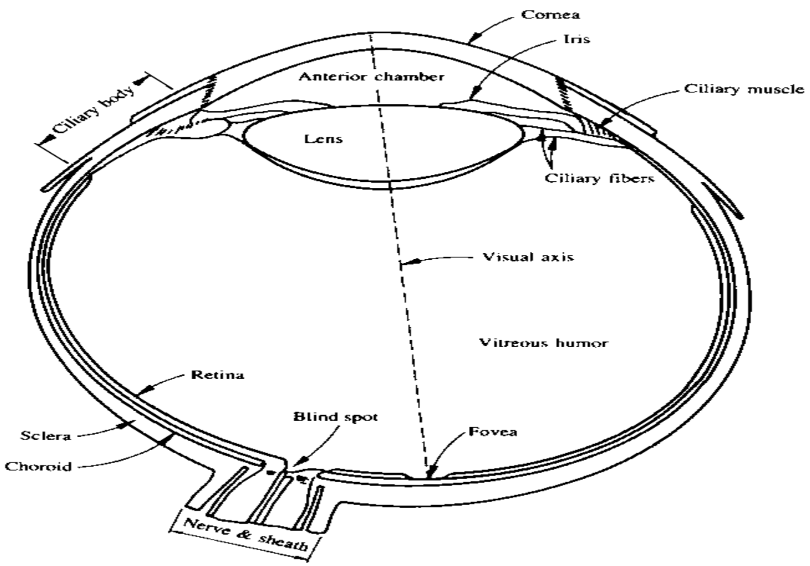
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

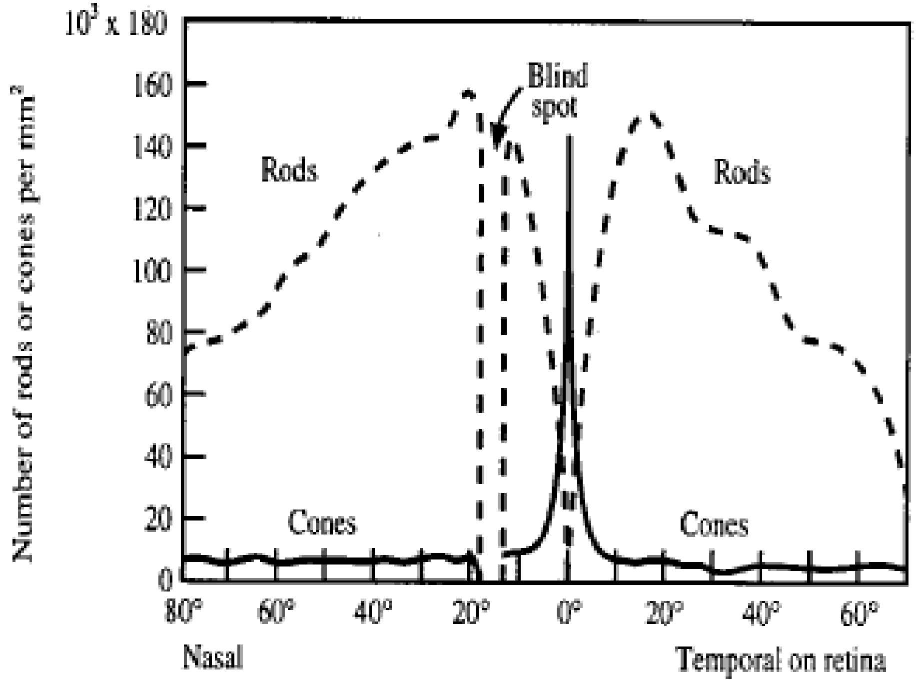
Elements of Visual Perception

Structure of the Human Eye ....

Retina: contains light receptors - Cones & Rods

- **Cones**:
  - 6 to 7 million
  - located mainly in central part of retina fovea (muscles controlling the eye rotate the eyeball until the image falls on the fovea)
  - High sensitive to color,
  - Can resolve fine details because each one is connected to its own nerve.
  - Cone vision: photopic or bright-light vision.
- **Rods:**
  - 75 to 150 million, Distributed a wide region on the retina surface.
  - Several rods are connected to a single nerve end reduce the amount of detail observation.
  - Serve to give a general, overall picture of the field of view.
  - Not involved in color vision, responsible for low level of illumination.
  - Rod vision is called scotopic or dim-light vision.

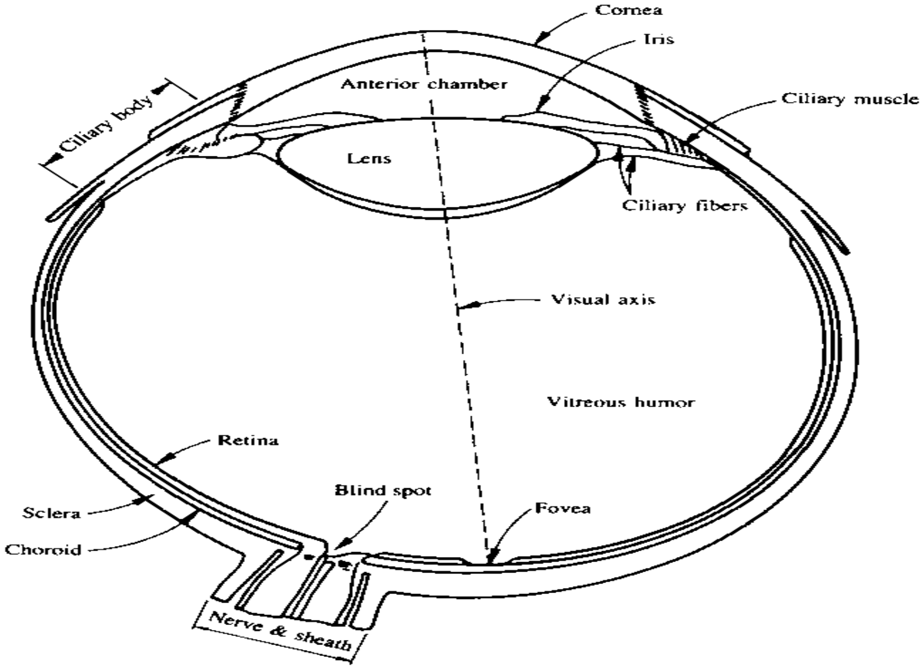


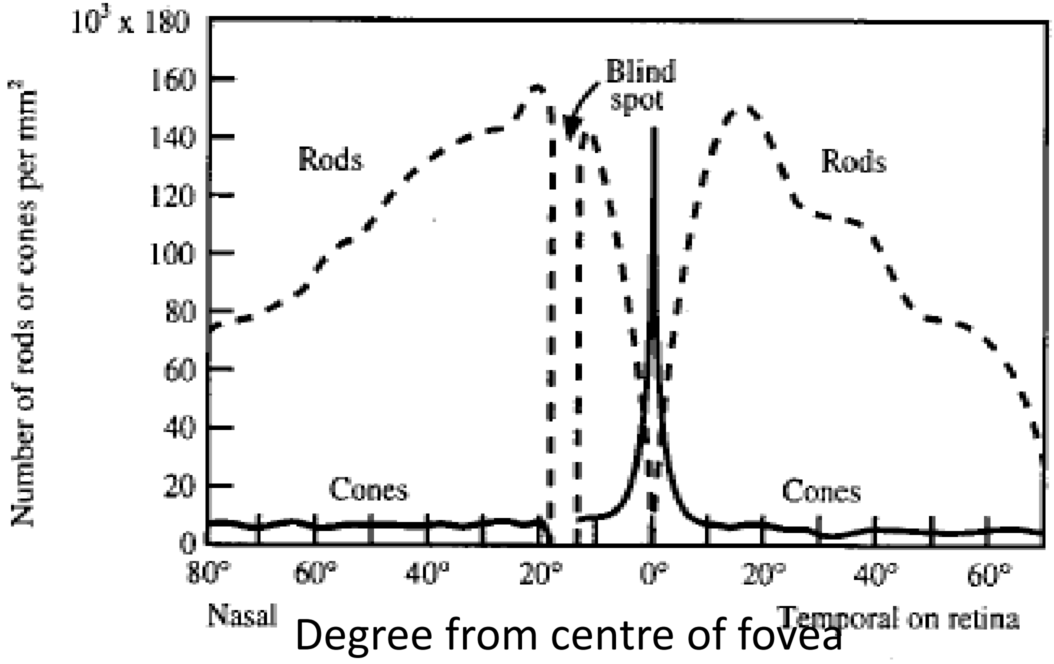


Elements of Visual Perception

- **Blind spot:** A region of retina without receptors, optic nerves go through this part.
- **Fovea:** A circular area of about 1.5 mm in diameter.

- - - Rods
___ Cones





Except for blind spot the distribution of receptors is radially symmetric about the fovea.
Receptors density is measured in degrees from the fovea.

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Suggested Readings

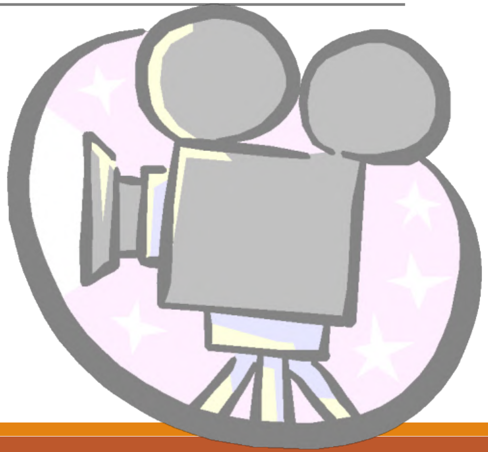
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

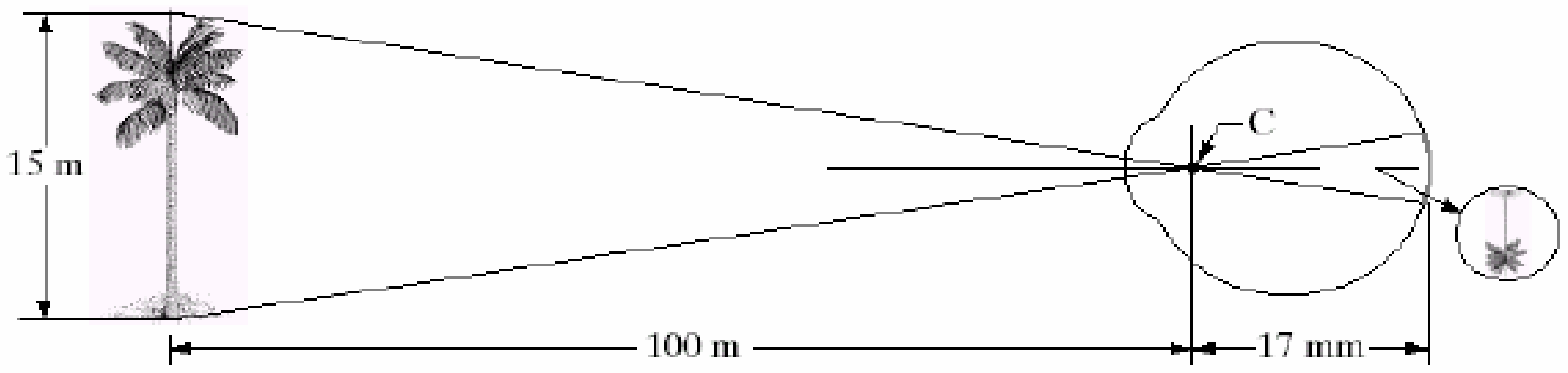
- Elements of Visual Perception
  - Image Formation in the Eye
  - Brightness Adaptation and Discrimination

Image Formation in the Eye

- Lens is flexible
- Refraction of lens is controlled by its thickness.
- Radius of curvature of the anterior surface of lens is greater than the radius of posterior surface.
- Thickness / shape is controlled by the tension of muscles (ciliary body) connected to the lens.
- ***Focus on distance objects***: lens is relatively flattened, ***refractive power is minimum.***
- ***Focus on near objects***: lens is thicker, ***refractive power is maximum.***
- The distance between the center and the retina (focal length) varies from 17 mm (minimum refractive index)~ 14 mm(maximum refractive index).
- Object farther away 3 m from eye , eye exhibit maximum focal length (17 mm) and lowest refractive power.

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.

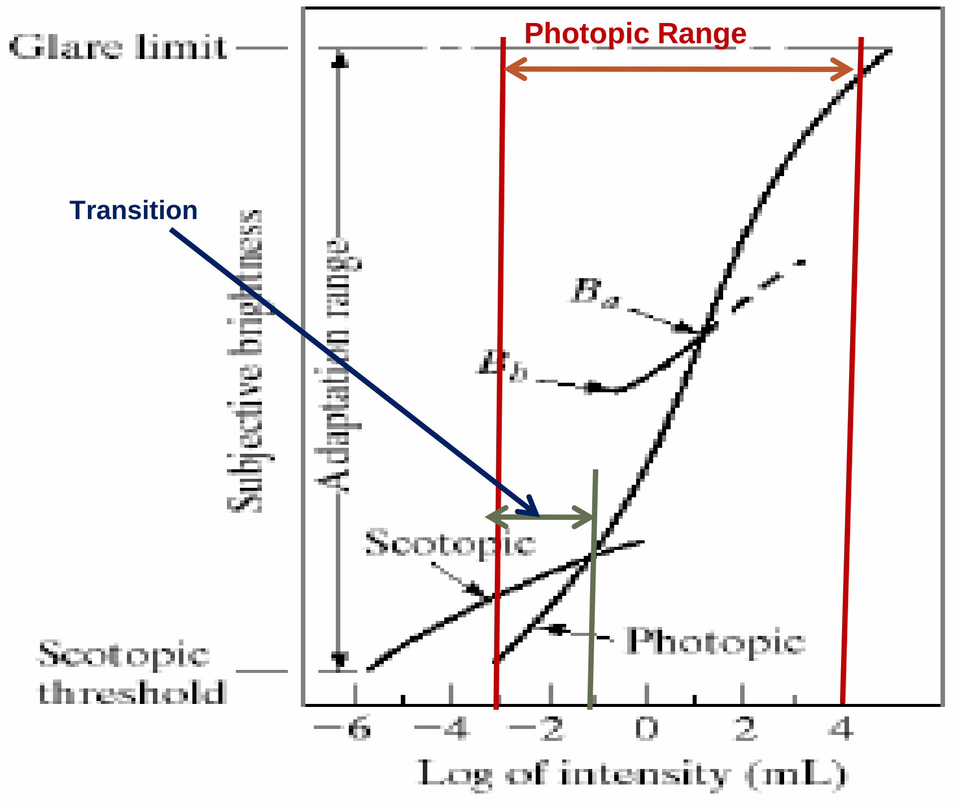


Object size 15 m at distance 100 m form an image of size 2.55 mm at retina.

- Retinal image is reflected primarily in the area of Fovea.
- Perception takes place by the relative excitation of light receptors.
- Light receptors transform the radiant energy into electrical impulses.
- Electrical impulses are decoded by the Brain.

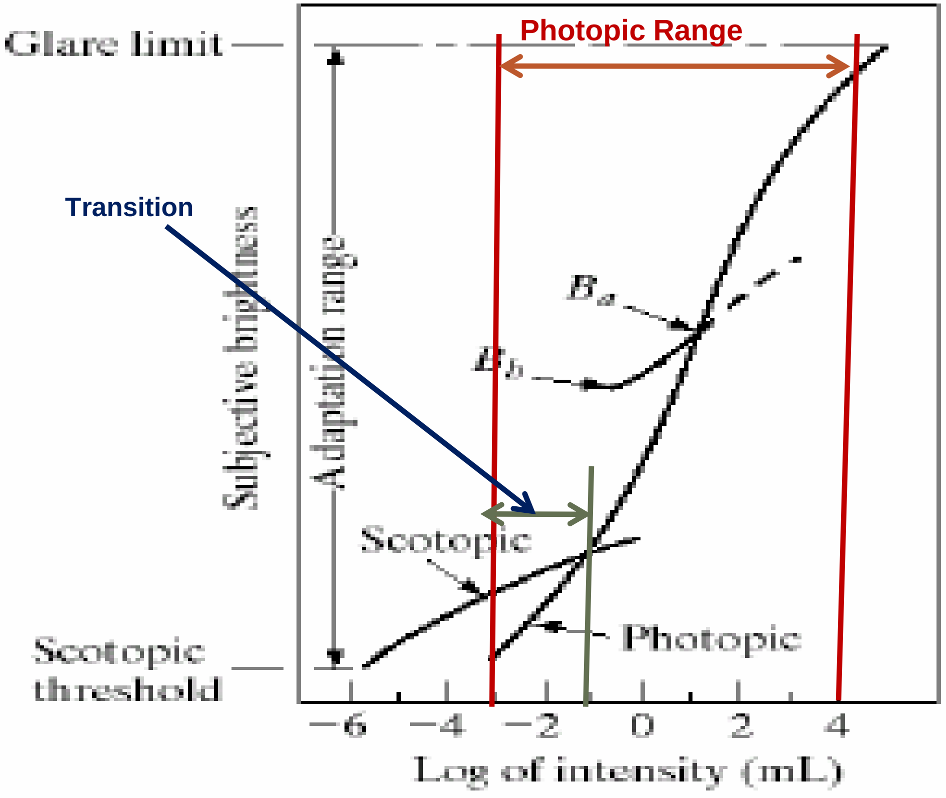
Brightness Adaptation

- The range of light intensity levels to which human visual system can adapt .
- This range is of the order of 10^{10} : ***scotopic threshold*** to the ***glare limit***.
- Experimental evidence shows that the subjective brightness (light intensity as perceived by human visual system) is a logarithmic function of light intensity incident on the eye.



Brightness Adaptation

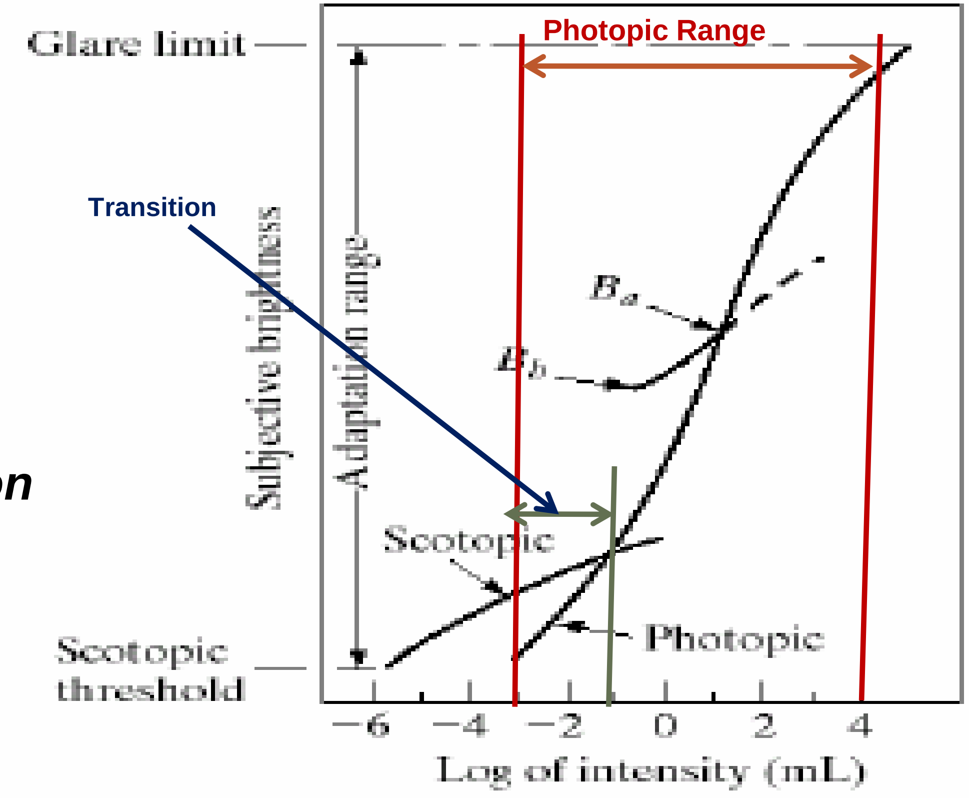
- This range is of the order of 10^{10} : ***scotopic threshold*** to the ***glare limit***.
- Glare limit: The maximum brightness under which the object is just visible. If brightness level is increased further we can not see the object.
- Scotopic threshold: The minimum brightness under which the object is just visible. If brightness level is decreased further we can not see the object.



Brightness Adaptation

Long solid curve : HVS adaption range

- Photopic range ~ 10^6
- Transition from scotopic to photopic is gardual over approximate range from 0.001 to 0.1
- millilambert (- 3 to -1 mL on the log scale)



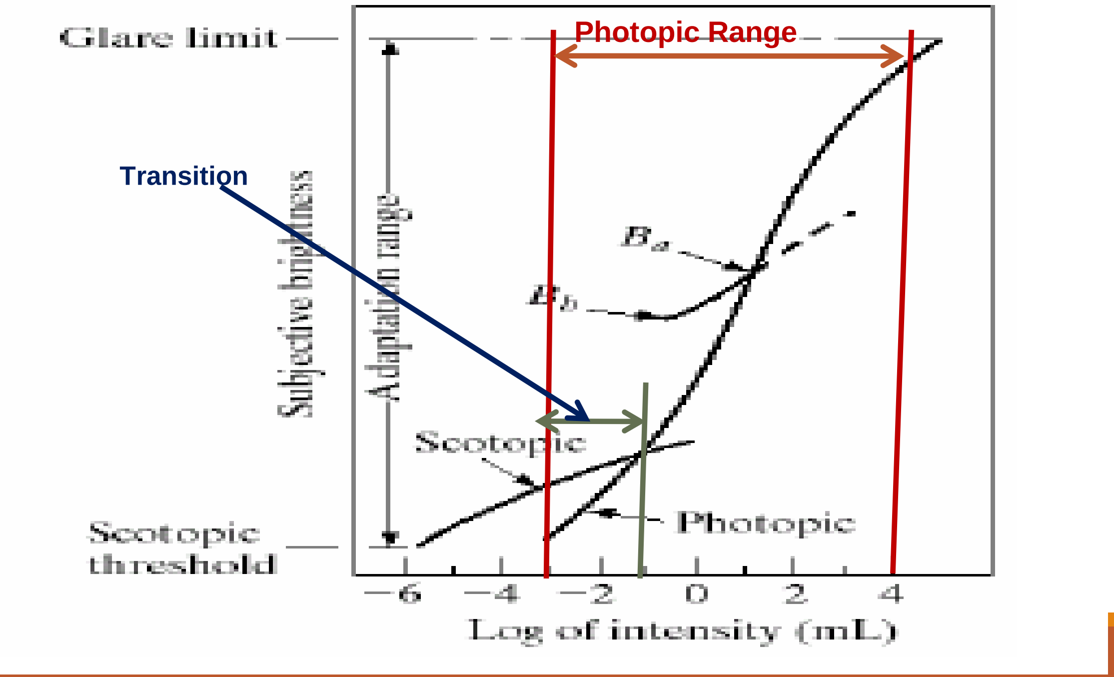
- ***Note:*** HVS can not operate over the entire range simultaneously.
- It accomplishes large variations by change in its overall ***sensitivity***, a phenomenon known as ***brightness adaptation***

Photopic : color

Scotopic : Gray

Brightness Adaptation

- Total range of intensity level simultaneously distinguishable <<< total adaption range.
- For any given set of conditions, the sensitivity level of visual system is called ***Brightness Adaption Level.*** Example B_a .
- Short intersecting curve: Subjective brightness perceived by Eye when adapted to this level.
- B_b is level below which all stimuli are perceived as indistinguisha



- The upper (- - -) is not restricted, but extended to for, loses the meaning because much higher intensity would simply raise the adaption level higher than B_a .

Brightness Discrimination

Digital Image Displayed as Discrete Set of Intensity

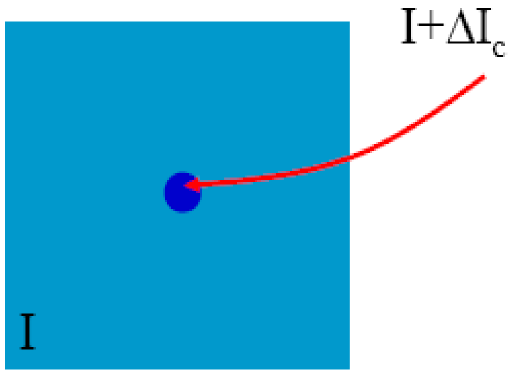
Eye Ability to discriminate between different intensity level is an important consideration in presenting image processing result.

Weber Experiments – *brightness discrimination of HVS*

- Having a subject look at flat, uniformly illuminated area large enough to occupy entire field of view

Opaque glass illuminated from behind by light source whose intensity, I, can be varied. (work as diffuser).

This field is added an increment of illumination, ΔI, in the form of a short – duration flash that appears as a circle in the centre of the uniformly illuminated field.



- If ΔI is not enough, the subject says no : Indicating that no perceivable change.
- As ΔI get stronger, the subject may give positive response of “yes,”: Indicate perceived changes.

Brightness Discrimination

Weber ratio:

The quantity $\Delta I/I$, where ΔI, is the increment of the illumination discriminable 50 % of the time with background illumination I, is called the ***Weber ratio***.

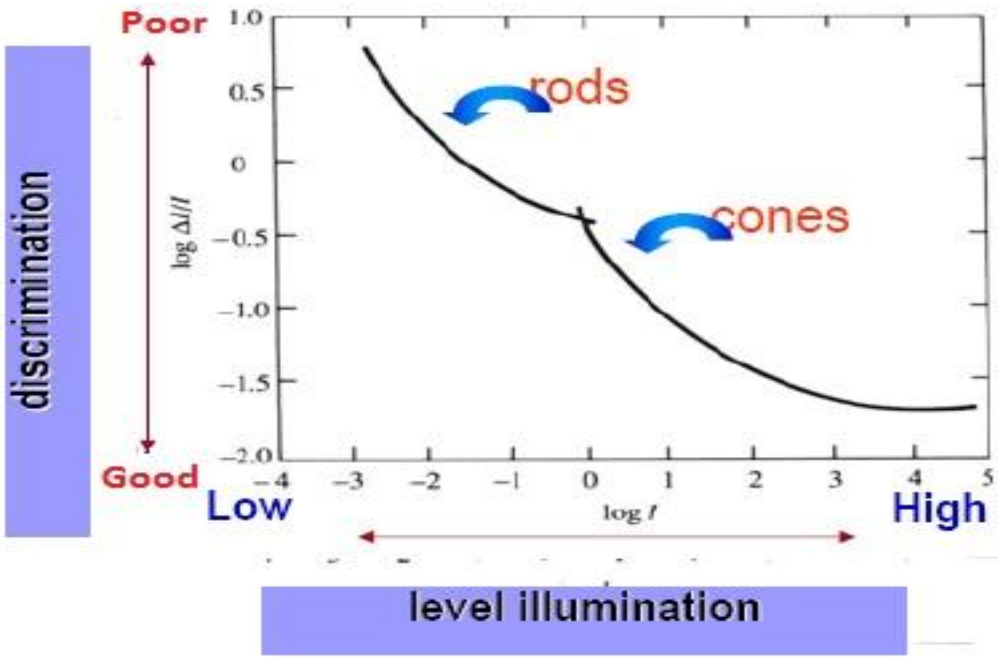
Small $\Delta I/I$ → Small percentage change in intensity is discrimanable → Good brightness discrimaination

Large $\Delta I/I$ → Large percentage change in intensity is discrimanable → Poor brightness discrimaination

Brightness Discrimination

Weber Ratio as Function of Intensity :

- Brightness discrimination is poor (the Weber ratio is large) at low levels of illumination and improves significantly (the ratio decreases) as background illumination increases.
- Hard to distinguish the discrimination when it is bright area but easier when the discrimination on a dark area.
- At low levels of illumination, vision is carried out by activity of **rods**, at high levels by **cones**.



Brightness Discrimination

Digital Image Displayed as Discrete Set of Intensity

How many different intensity level could be discriminate

Weber Experiments – *no of intensity level HVS*

- Background illumination is held constant and the intensity of other source, instead of flashing, is allowed to vary incrementally, from never being perceived to always being perceived.
- Typical observer can discriminate 1 -2 dozen different intensity changes.
- This number of different intensities a person can see at any one point in a monochrome image.
- But as eye roam about image / scene the average background changes, thus allowing a different set of incremental changes to be detected at each new adaption level.

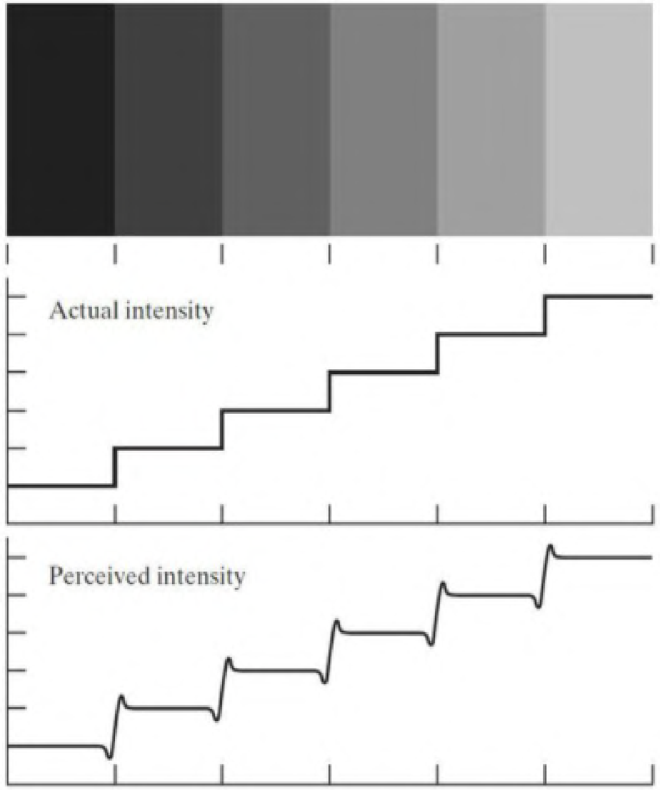
Eye is capable of much broader of over all intensity discrimination

Brightness Discrimination

- **the perceived brightness is not a simple function of intensity.**
- Visual system tends to undershoot or overshoot around the boundary of regions of different intensities.
- The intensity of the stripes is constant but we actually perceive a brightness pattern is strongly scalloped near the boundaries.

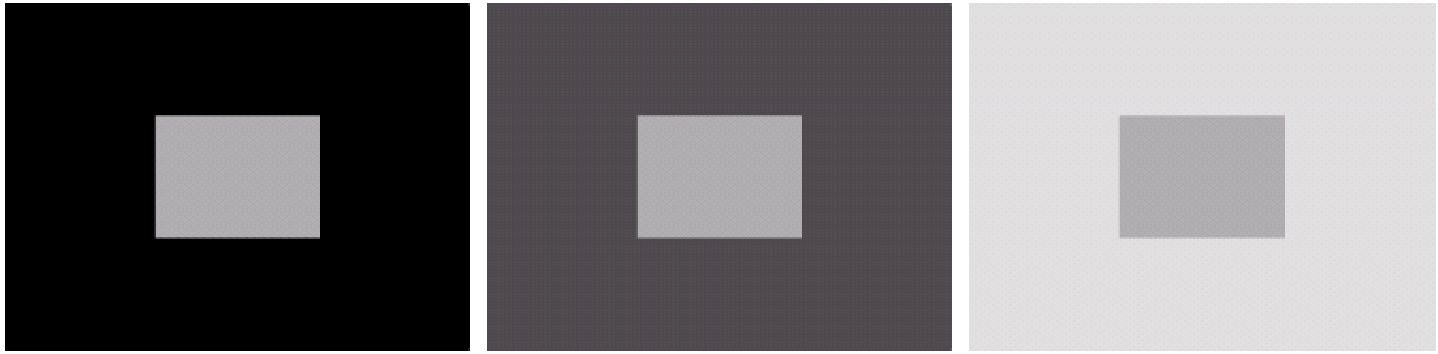
a
b
c

FIGURE 2.7 Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.



Simultaneous Contrast Effect

- Which small square is the darkest one ?



a b c

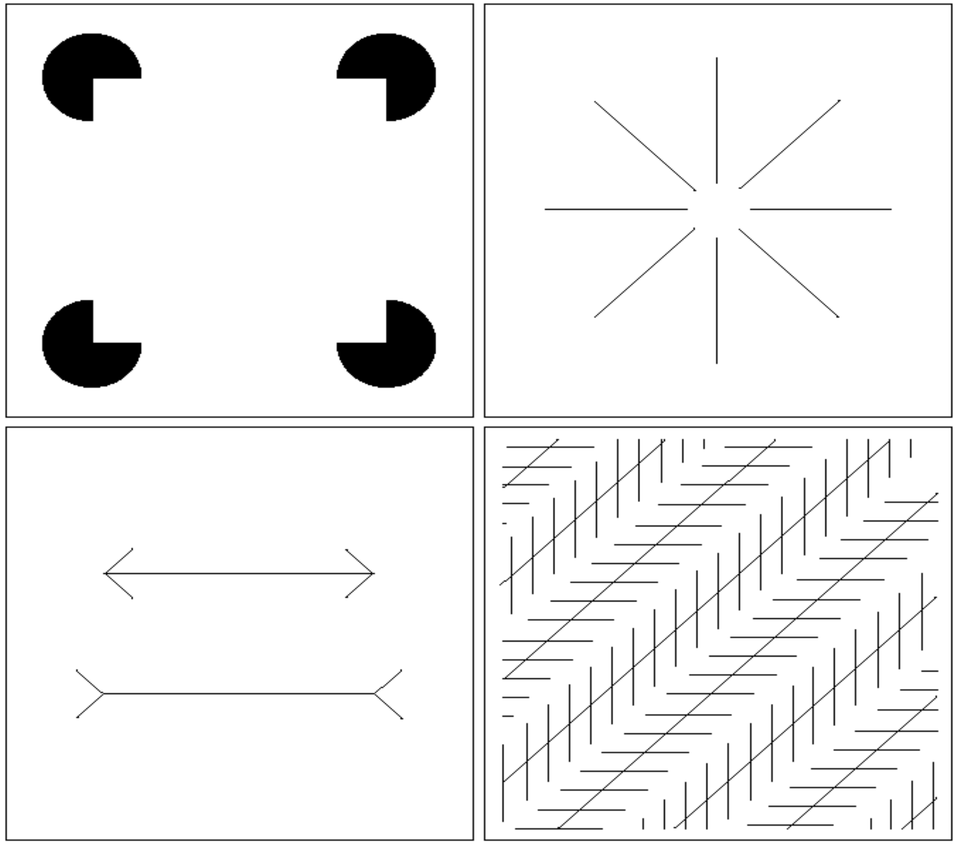
FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

- All the small squares have exactly the same intensity, but they appear to the eye progressively darker as the background becomes brighter.
- Region's perceived brightness does not depend simply on its intensity.

Optical Illusion

The eye fills in non existing information or wrongly perceives geometrical properties of objects

- Outline of square is seen, in spite of the fact that no line defining a such figure.
- Illusion of a complete circle.
- Line are same length but one appears shorter than other.
- All line are oriented at 45^o are equidistant and parallel. Crossing creates illusion that those lines are far from being parallel.



a b
c d

FIGURE 2.9 Some well-known optical illusions.

Suggested Readings

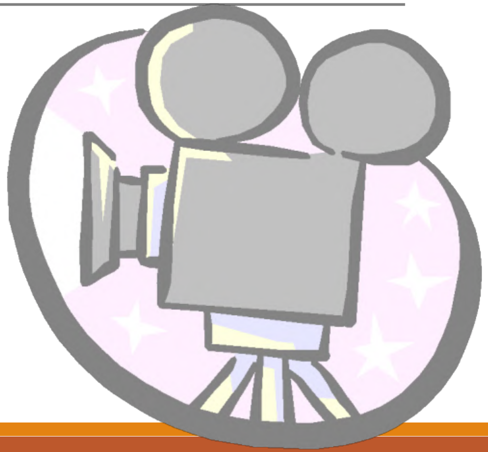
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Sensing and Acquisition
  - Single Sensor
  - Sensor Strips
  - Sensor Arrays
  - Image Formation Model

Image Sensing and Acquisition

- **The illumination source emits energy which is reflected or absorbed by the element of the scene being imaged.**
  - The wavelength of an EM wave required to see an object must be of the same size as or smaller than the object.
  - The physical properties of the sensor materials also limits the capability of imaging sensors (cf. CCD & CMOS, CCD & IRCCD).

Image Sensing and Acquisition

Radiance

Total amount of energy that flows from the light source (Watts)

Luminance

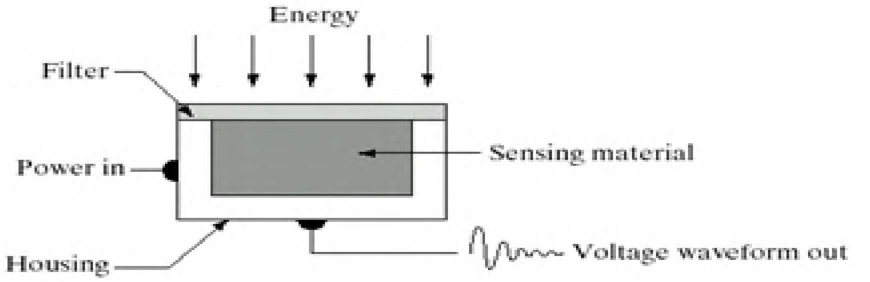
A measure of the amount of energy an observer perceives from a light source (lumens)

Brightness

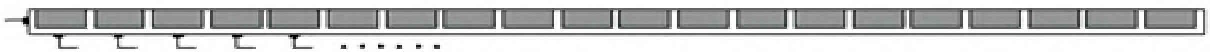
A subjective descriptor of light perception.

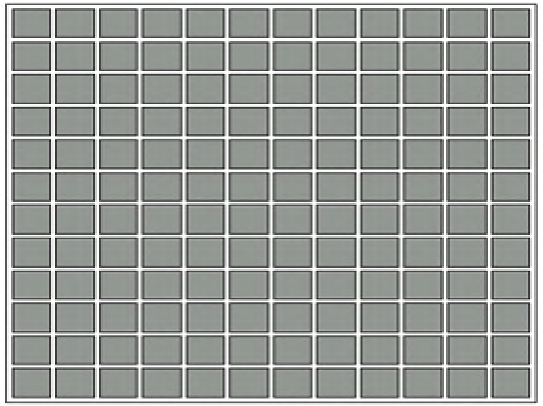
Light emitted from a far infrared source have high radiance, but almost no luminance for a sensor in visible band.

Sensor Arrangement (Spatial Sampling)



Single Imaging Sensor


Line Sensor


Array Sensor

- Incoming energy is transformed into voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected.
- O/p voltage waveform is the response of the sensor(s), and digital quantity is obtained from each sensor by digitizing its response.

Image Acquisition

- **Using Single Sensor**

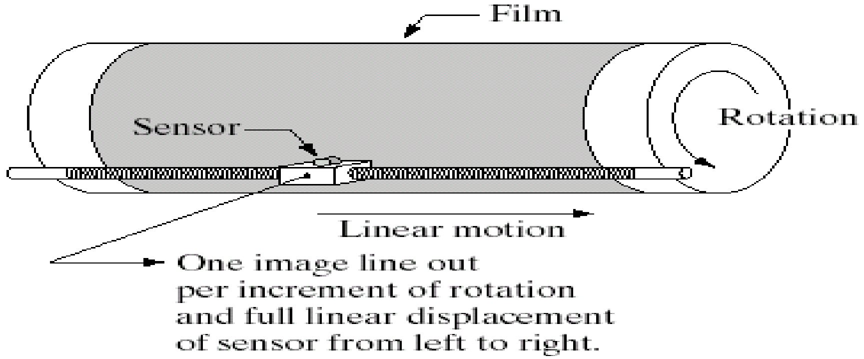


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

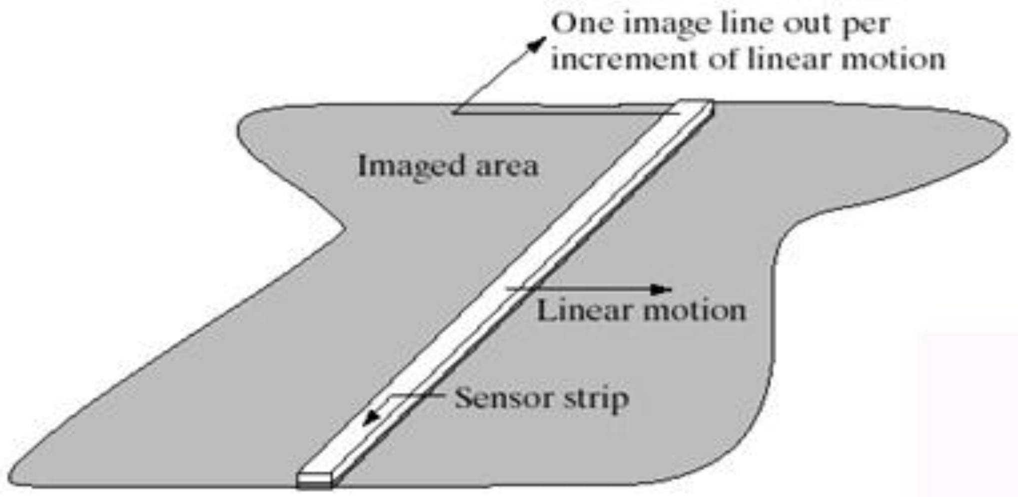
- Relative displacement in both the x- and y- directions between the sensor and the area to be imaged.

-These arrangement needs high precision scanning mechanism.

- Inexpensive but takes lot of time to obtain high-resolution images.

Image Acquisition

- **Using Line Sensor**



-In line arrangement of sensors in the form of a sensor strip.

-Strip provide imaging in one direction and motion perpendicular to the strip provides imaging in other direction.

-The imaging strip gives the one line of image at a time, and motion of the strip completes the other dimension.

-Lens or other focusing schemes are used to project the area to be scanned onto the sensor.

Used in most flat bed scanner and airborne imaging application

Image Acquisition

- **Line Sensor in Ring Configuration**

  - Used in medical and industrial imaging obtain cross-sectional images of 3-D objects.

  - Computerized Axial Tomography (CAT):

- A rotating X- ray source provides illumination and the portion of the sensors opposite the source collect the X-ray energy that pass through the object.
- Output of sensor must be processed by reconstruction algorithm to obtained the meaningful cross sectional images.
- A 3-D digital volume consisting of stacked images is generated as the object move in a direction perpendicular to the sensor ring.
- MRI, PET imaging modalities is also based on CAT principle only source, sensor, and type of images are different.

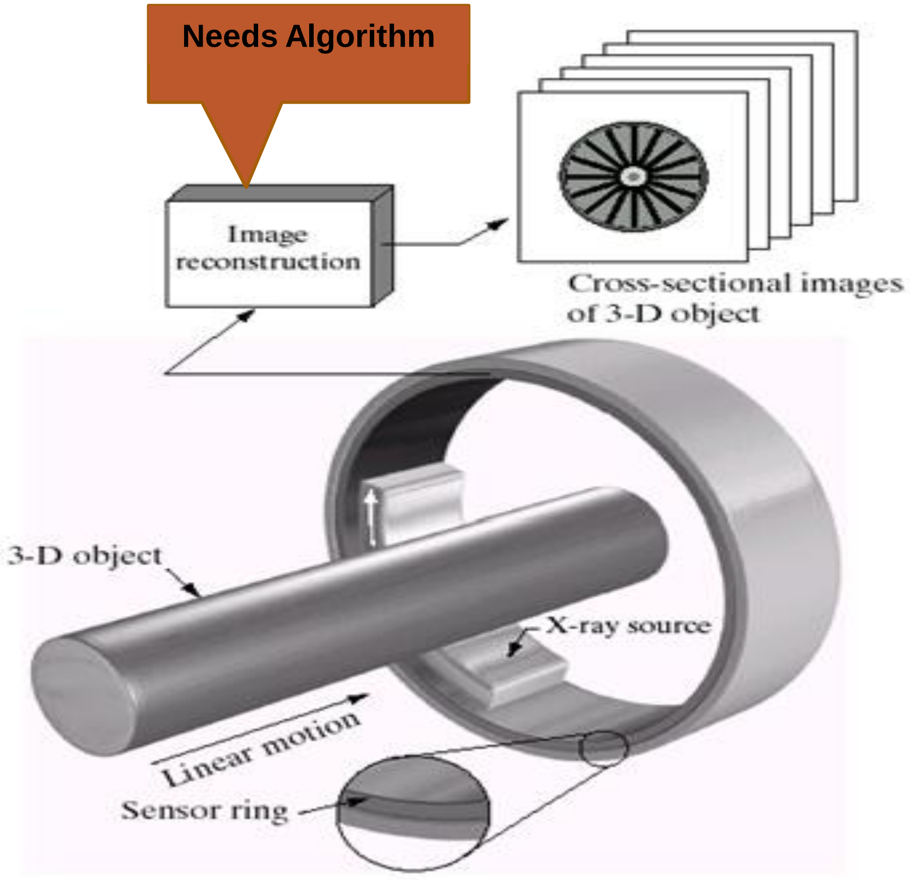


Image Acquisition

- **Using Sensor Arrays**

- Sensor are arranged in 2-D array.

- Basic principle is energy from an illumination source is reflected by a scene element is collected and focused by an imaging system onto an image plane, which same as focal plane.
- The sensor array produces the outputs proportional to the integral of light energy received at each sensor.
- Digital and analog circuitry sweep these outputs and convert them to video signal.

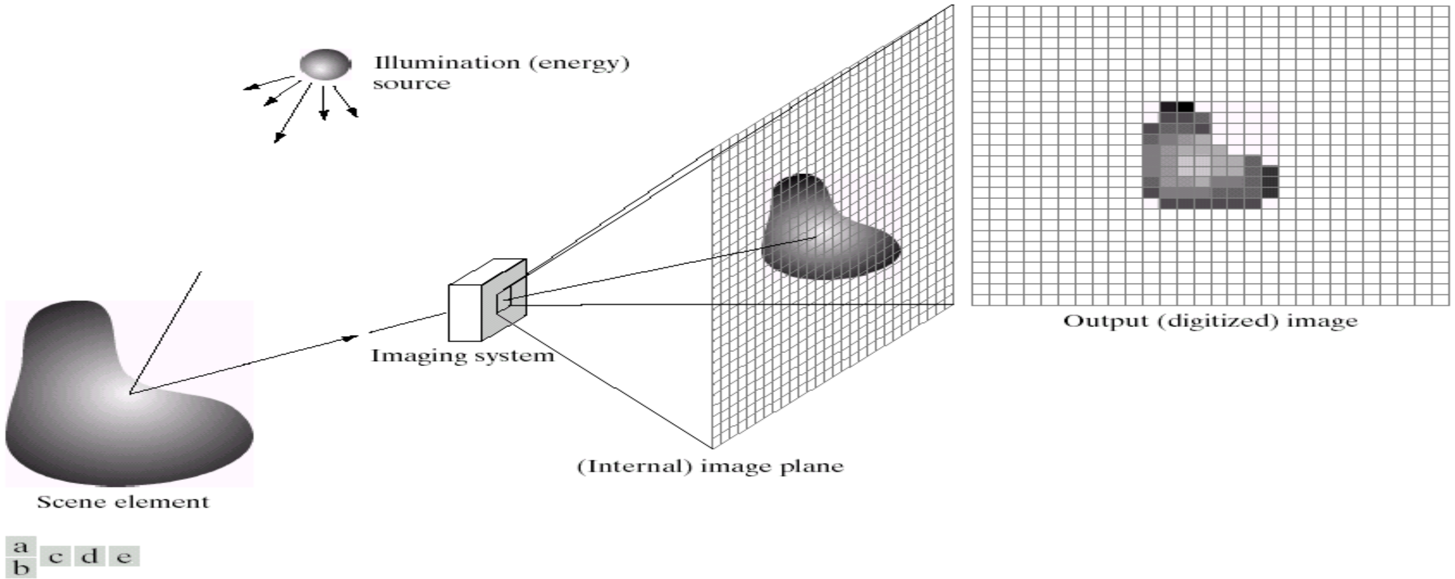
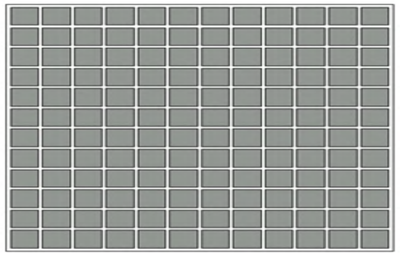


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.



Array Sensor

Image Formation Model

Light Intensity Function

- **Image** - two dimensional light intensity function, denoted by ***f(x,y)*** , where the value or amplitude of ***f*** at spatial coordinates ***(x,y)*** gives the intensity (brightness) of the image at that point.

Perception of an object : Light reflected from that object

f(x,y) can be characterized by two components

1. The amount of source light incident on the scene being viewed

   Illumination component - *i(x,y)*

2. Amount of light reflected by the objects in the scene

   Reflectance component - *r(x,y)*

$$f(x,y) = i(x,y)r(x,y)$$
$$0 \le i(x,y) < \infty$$
$$0 \le r(x,y) \le 1$$

- Physical meaning of *f* is determined by the source of the image i.e. *i(x,y).*
- *r(x,y)* is determine by the characteristics of the imaged object.

Above expression is also valid for transmission imaging, in this transmissivity is used instead of reflectivity.

Image Formation Model

Light Intensity Function

- Reflectance is bounded by ***0 (total absorption)*** and ***1 (total reflectance)***
- The nature of ***i(x,y)*** is determined by the light source and ***r(x,y)*** is determined by the characteristics of the objects in a scene.

***i(x,y)*:**
Clear Day - 9000 foot-candles
Cloudy day - 1000 foot-candles
Full moon - 0.01 foot candles
commercial office - 100 foot candles

Typical values of ***r(x,y)*:**
Black Velvet - 0.01
Stainless Steel - 0.65
Flat-White wall paint - 0.80
Silver plated metal - 0.90
Snow - 0.93

Grey Level (l) at point (x,y)
- Intensity of monochrome image ***f*** at coordinates ***(x,y)***

Image Formation Model

Light Intensity Function

- we call the intensity of a monochrome image ***f at*** coordinate ***(x,y) the gray level (l) of the image*** at that point.
- thus, ***l*** lies in the range $L_{min} < l < L_{max}$
- L_{min} *is positive and* L_{max} *is finite.*
- gray scale = [L_{min}, L_{max}]
- common practice, shift the interval to [***0, L***]
- ***0 = black , L = white***
- All intermediate values are shades of gray varying from black to white.

Suggested Readings

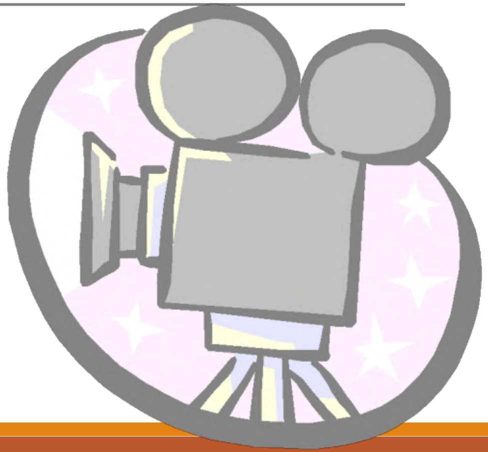
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- ➢Image Sampling and Quantization
- ➢Digital Image Representation
- ➢ Spatial and Gray -Level Resolution

Light and electromagnetic Spectrum

Frequency (ν) and **wavelength** (λ) are joined by the equation $\lambda = c/\nu$, where c is the speed of light.

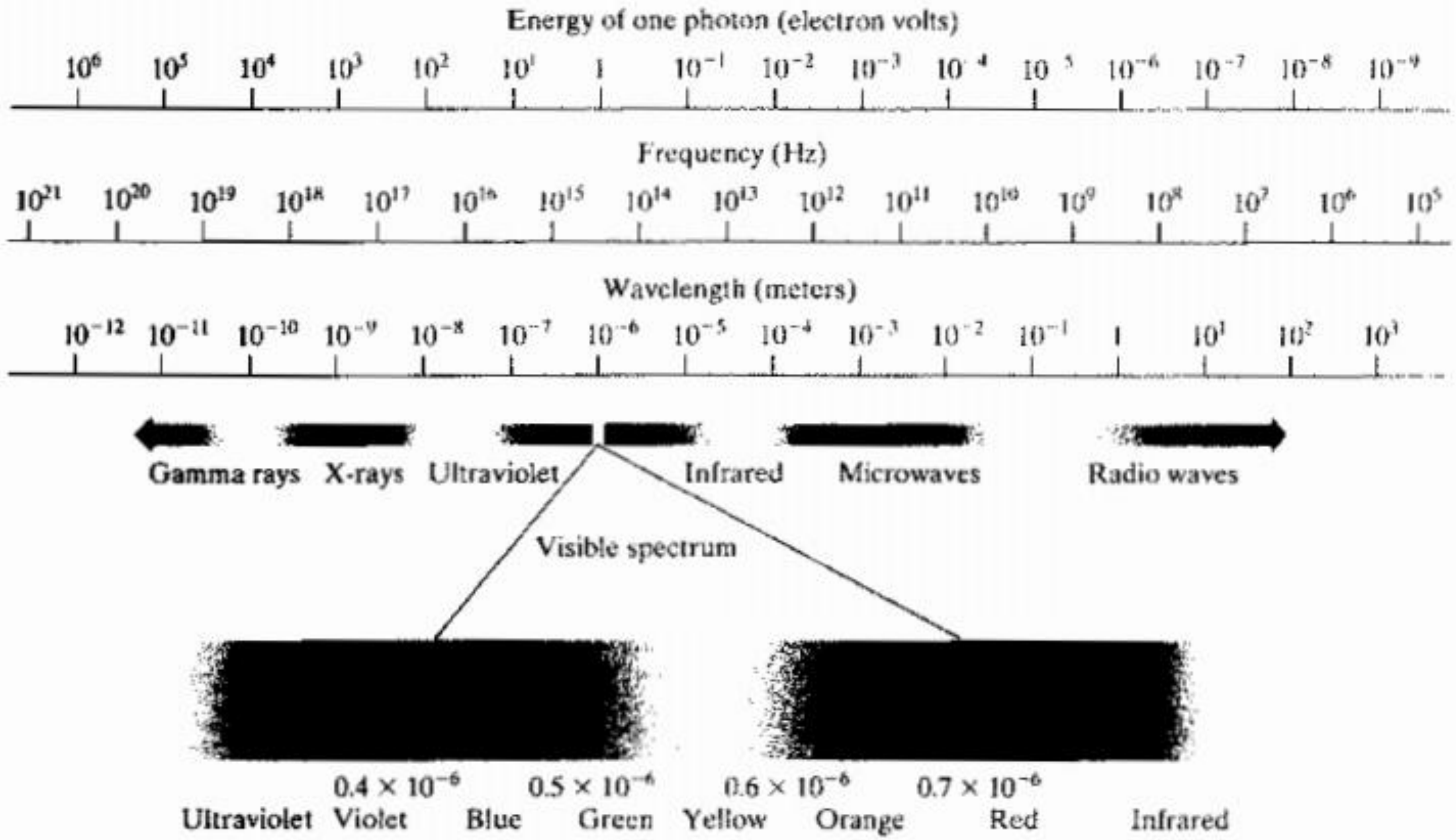


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

The energy of various Components of electromagnetic Spectrum is given by

$E=h\nu$

Where, h is Plank's constant

Image Sampling And Quantization

For computer processing, an image function needs to be digitized both ***spatially and in amplitude***.

- ***Image Sampling*** – Digitization of the spatial coordinates ***(x,y)***
- ***Gray-level quantization*** – Digitization of Amplitude
- For sampling for 2-D sensor array is needless

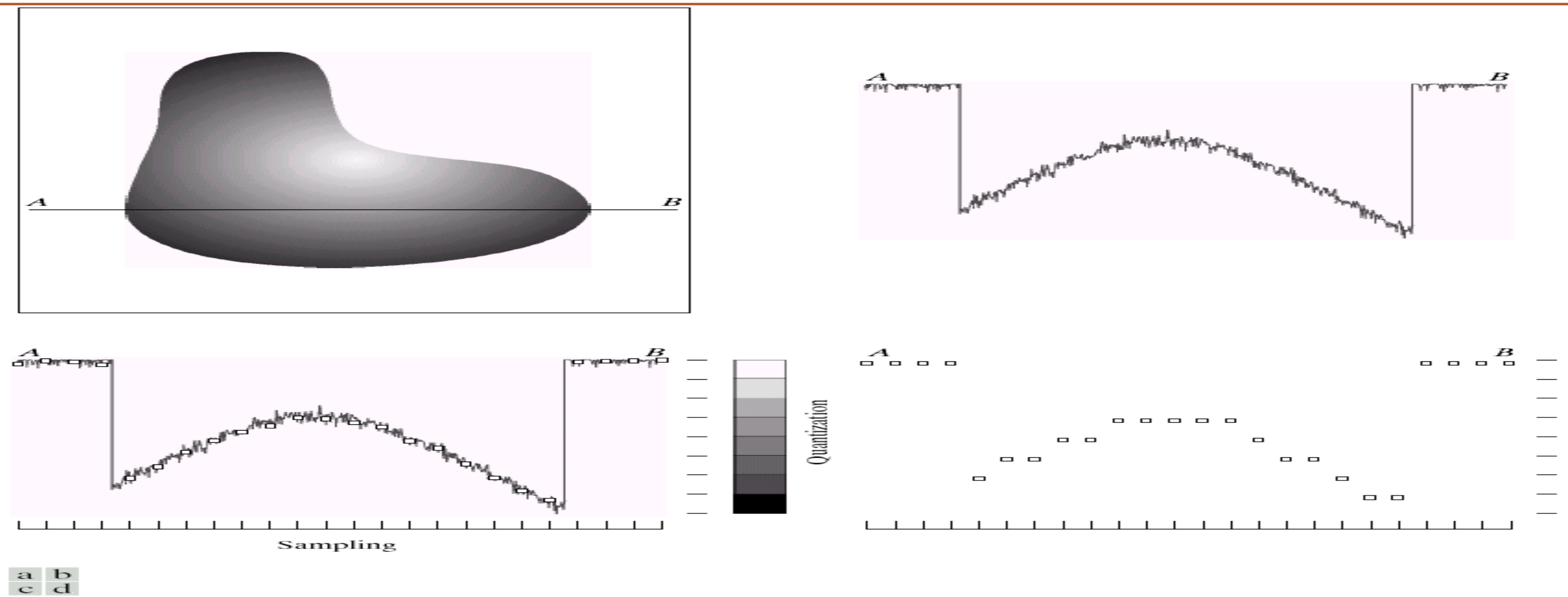
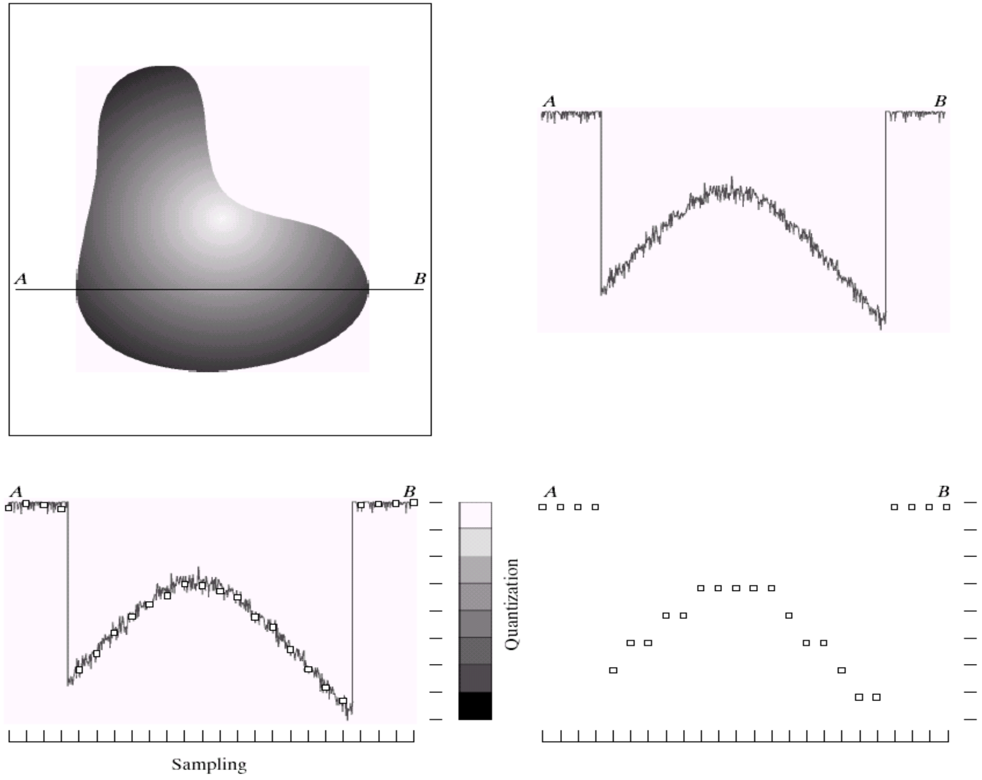


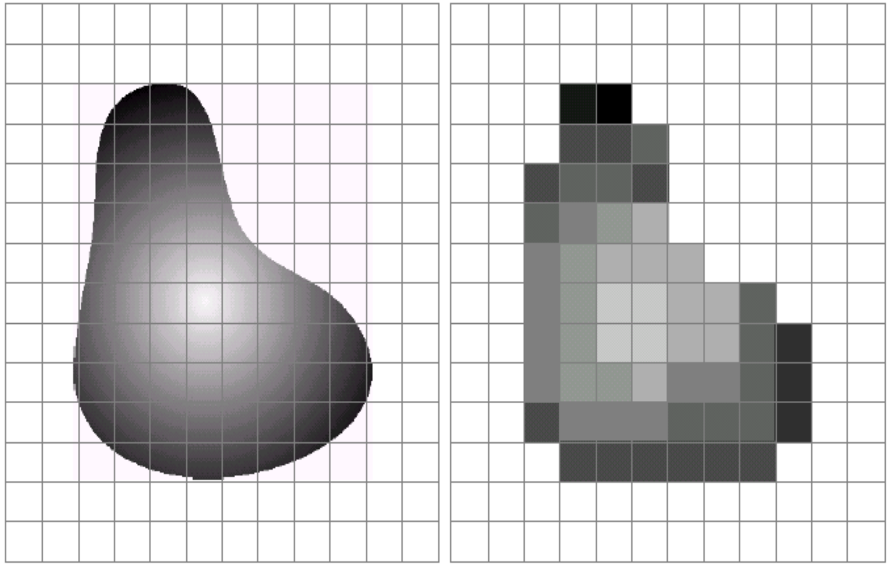
FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Image Sampling And Quantization



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



a b

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

A Simple Image Formation Model

Light Intensity Function

- **Image** - two dimensional light intensity function, denoted by ***f(x,y)*** , where the value or amplitude of ***f*** at spatial coordinates ***(x,y)*** gives the intensity (brightness) of the image at that point.

Perception of an object : Light reflected from that object

f(x,y) can be characterized by two components

1. The amount of source light incident on the scene being viewed

   Illumination component - *i(x,y)*

2. Amount of light reflected by the objects in the scene

   Reflectance component - *r(x,y)*

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

$$0 \leq i(x,y) < \infty$$

$$0 \leq r(x,y) \leq 1$$

- Physical meaning of *f* is determined by the source of the image i.e. *i(x,y).*
- *r(x,y)* is determine by the characteristics of the imaged object.

Image Formation Model

Light Intensity Function

- Reflectance is bounded by ***0 (total absorption)*** and ***1 (total reflectance)***

- The nature of ***i(x,y)*** is determined by the light source and ***r(x,y)*** is determined by the characteristics of the objects in a scene.

***i(x,y)*:**
Clear Day – 90,000 lm/m^2
Cloudy day - 10,000 lm/m^2
Full moon - 0.1 lm/m^2
commercial office - 1,000 lm/m^2

Typical values of ***r(x,y)*:**
Black Velvet - 0.01
Stainless Steel - 0.65
Flat-White wall paint - 0.80
Silver plated metal - 0.90
Snow - 0.93

Grey Level (l) at point (x,y)
- Intensity of monochrome image ***f*** at coordinates ***(x,y)***

Image Formation Model

Light Intensity Function

- we call the intensity of a monochrome image ***f at*** coordinate ***(x,y) the gray level (l) of the image*** at that point.

  thus, ***l*** lies in the range $L_{min} < l < L_{max}$

- L_{min} *is positive and* L_{max} *is finite.*

  gray scale = $[L_{min}, L_{max}]$

- common practice, shift the interval to [***0, L-1***]
- ***0 = black , L-1 = white***
- All intermediate values are shades of gray varying from black to white.

Digital Image Representation

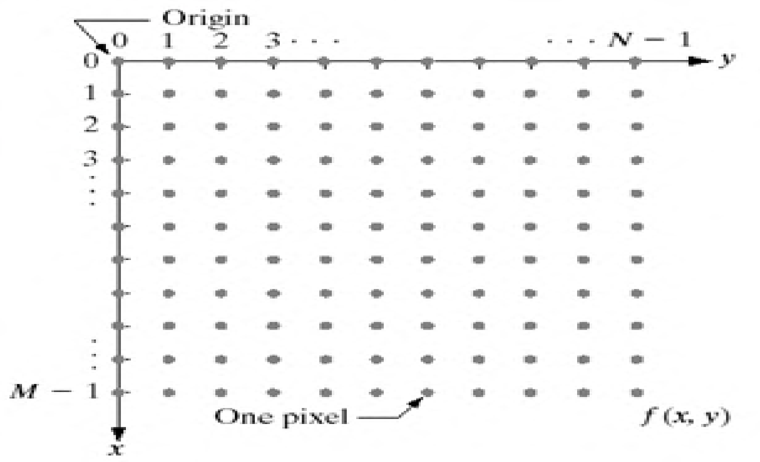
A ***digital image*** can be considered a ***matrix*** whose ***row and column indices*** identify a ***point*** in the image and the corresponding ***matrix element value*** identifies the ***gray level*** at that point.

- Definition of coordinate system
- Dynamic range of an image

  - the range of values spanned by the gray levels of existing pixels
- Spatial resolution (e.g., dpi, or 1024x1024)
- Gray-level resolution (8 (often visual), 10, 12 (e.g., thermal), or 16 bits / sample)

Digital Image Representation

Coordinate System & Matrix Representation



Continuous image *f(x,y)* – Equally spaced samples arranged as ***(N x M)*** array (matrix). ***Each element in the array is a discrete quantity***.

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

Digital Image

Image Element (pixel)

Digital Image Representation …

Need to decide the values for **N, M** and number of discrete gray levels allowed for each pixel **(G).**

In Digital Image Processing only restriction on M, N is that they are positive integer, however due to processing, storage, and sampling hardware consideration we take them is an ***integer power of 2***:

$$N = 2^n$$
$$M = 2^k$$
$$G = 2^m$$

G: No. of gray levels and it is assumed that these levels are equally spaced between ***0 and L*** =G-1 in the gray scale.

Amount of storage required to store a digitized image

$$\mathbf{b = N \times M \times k} \text{ (bits)}$$

If **M=N, b = N^2k**

Digital Image Representation …

Number of storage bits for various values of N and k

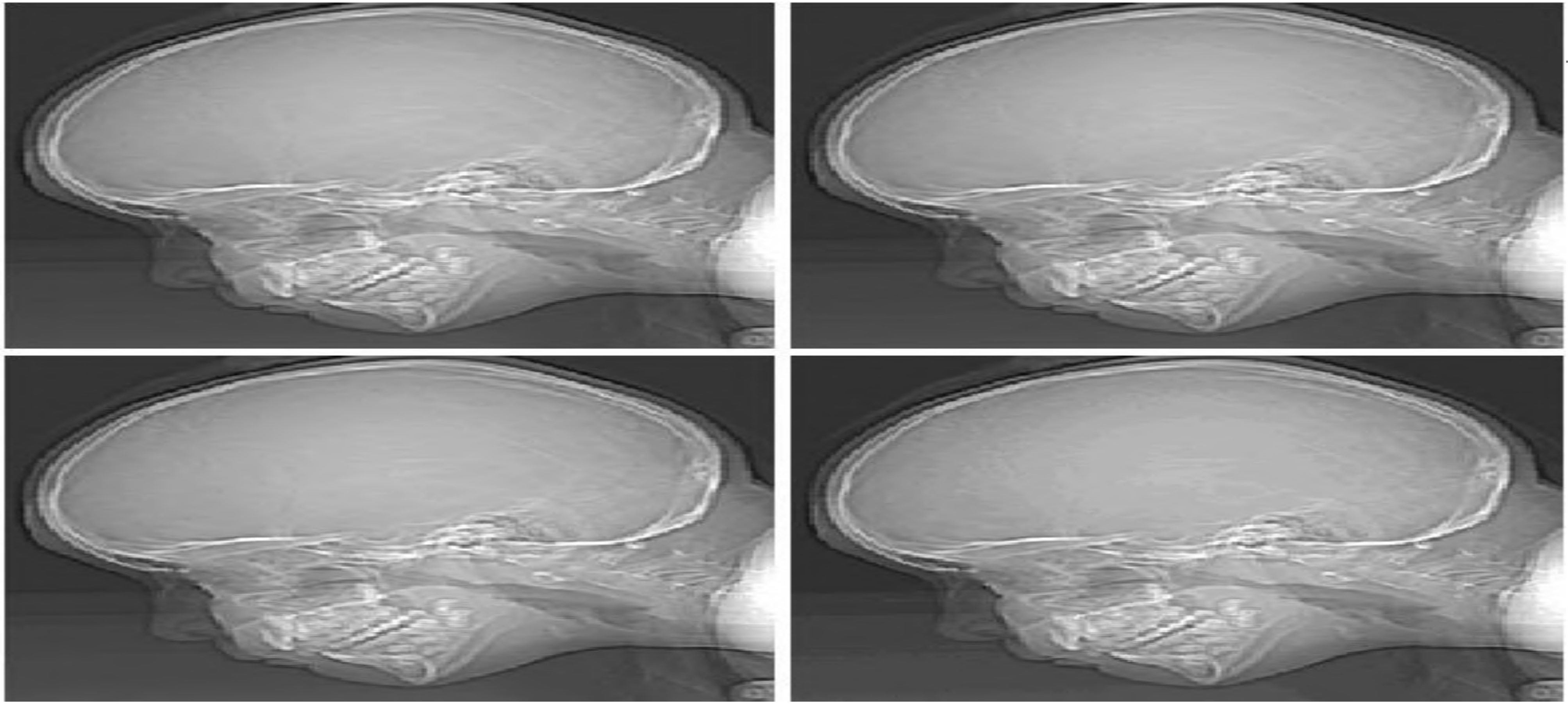
N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	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

Pixel values in highlighted region:

99	71	61	51	49	40	35	53	86	99
93	74	53	56	48	46	48	72	85	102
101	69	57	53	54	52	64	82	88	101
107	82	64	63	59	60	81	90	93	100
114	93	76	69	72	85	94	99	95	99
117	108	94	92	97	101	100	108	105	99
116	114	109	106	105	108	108	102	107	110
115	113	109	114	111	111	113	108	111	115
110	113	111	109	106	108	110	115	120	122
103	107	106	108	109	114	120	124	124	132

Resolution (Gray -Level)

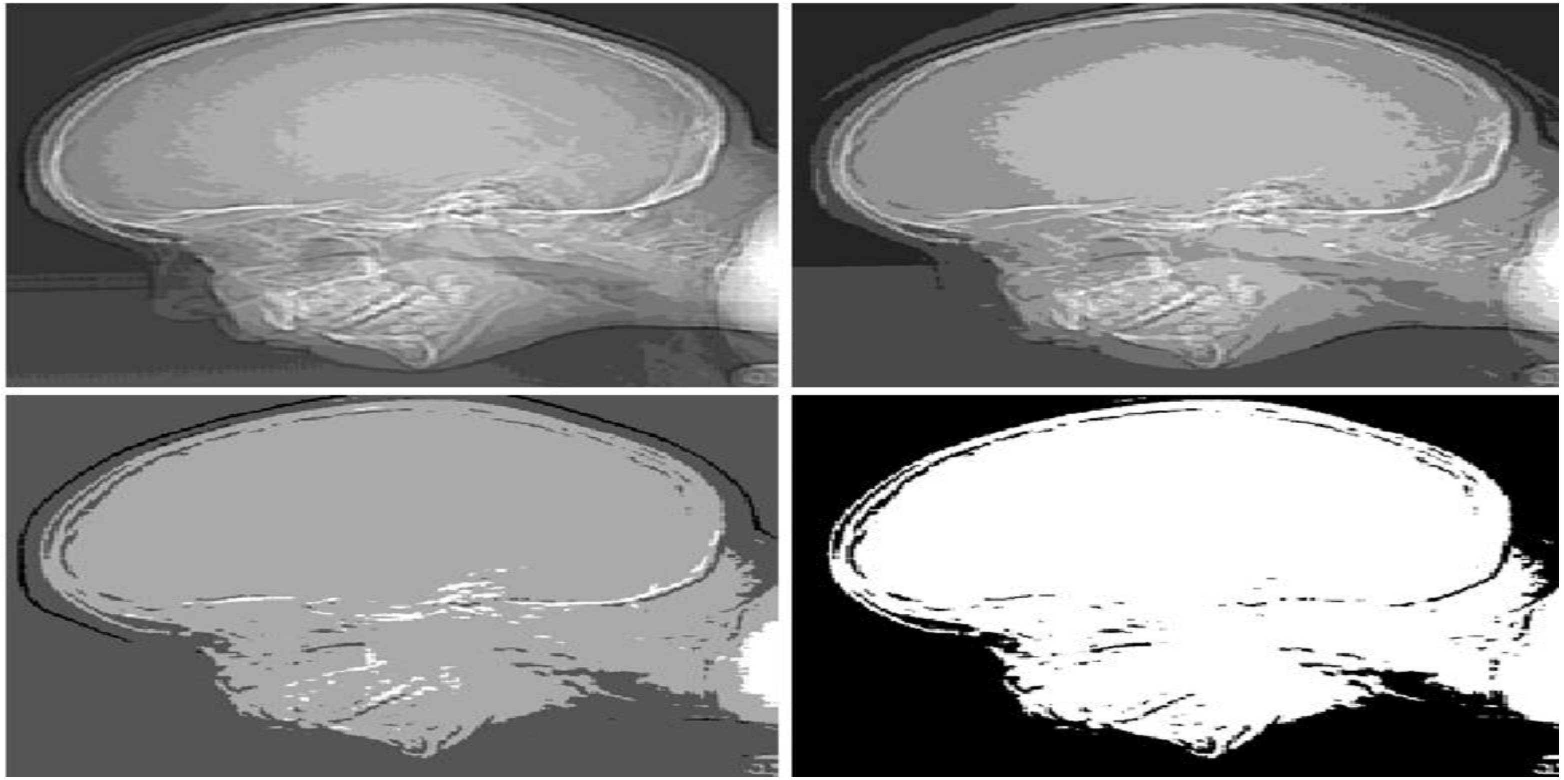
Keep the number of samples constant and reduce the number of gray levels from **256 to 32**.



(a) 452*374, 256-level image (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Resolution (Gray -Level)

Keep the number of samples constant and reduce the number of gray levels from **16 to 2**.



(e)–(h) Image displayed in 16, 8, 4 and 2 gray levels, while keeping the spatial resolution constant.

False Contouring Effect: imperceptible set of very fine ridge like structure in areas of smooth gray level.

Non uniform Sampling and Quantization

Adaptive scheme of sampling can be used to improve the appearance of an image for a fixed spatial resolution.

Here sampling process depends on the characteristics of the image.

- Fine sampling in the neighborhood of sharp gray-level transitions.
- Coarse sampling in relatively smooth regions

E.g. A face superimposed in a uniform background.

Nonuniform quantization: Considering the relative frequency of gray levels

- Frequent gray levels (within a range) – finely spaced
- Gray levels outside this range – coarsely spaced.

Resolution

Resolution (how much you can see the detail of the image) depends on sampling and gray levels.

The bigger the sampling rate (n) and the gray scale (g), the better the approximation of the digitized image from the original.

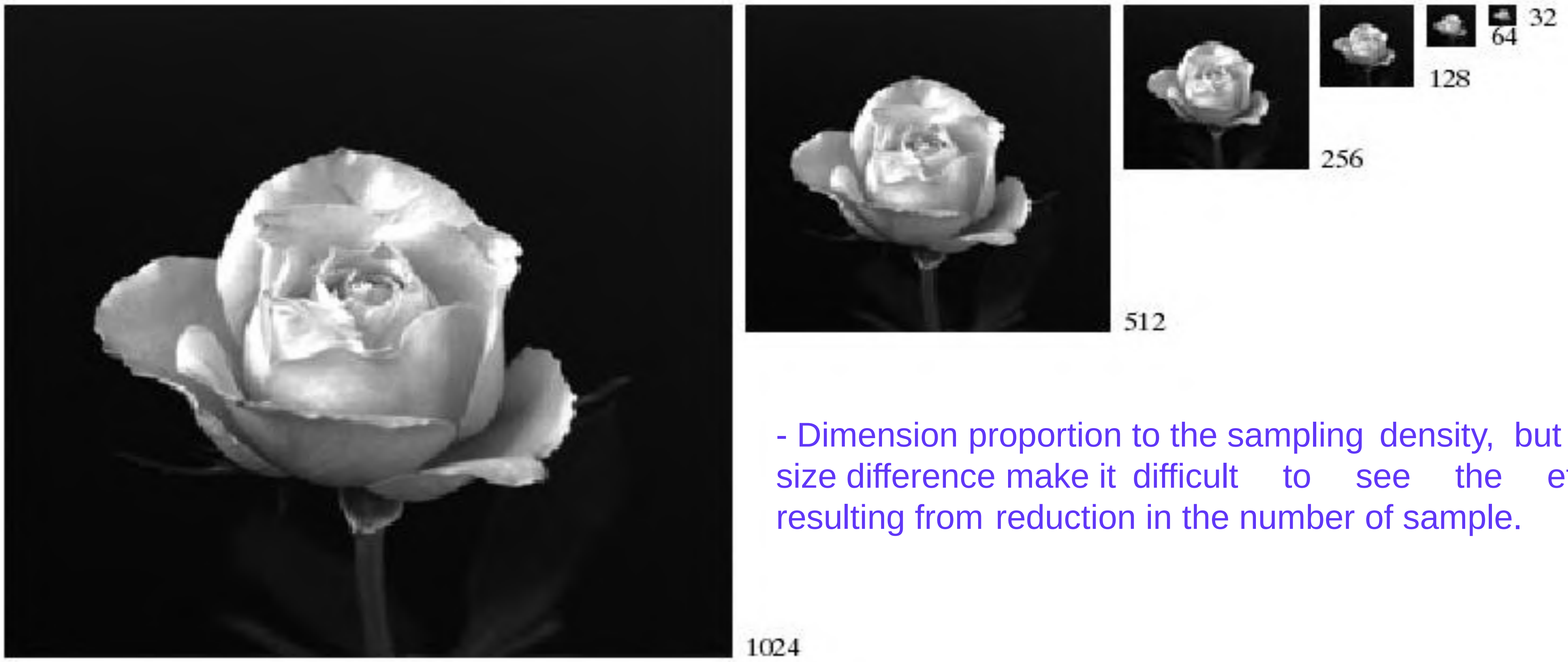
The more the quantization scale becomes, the bigger the size of the digitized image.

Resolution

Spatial Resolution: *Spatial resolution* is the smallest detectable detail in an image. Sampling is principle factor in determining the spatial resolution.

Grey level Resolution: *Gray-level resolution* similarly refers to the smallest detectable change in gray level.

Resolution (Spatial)



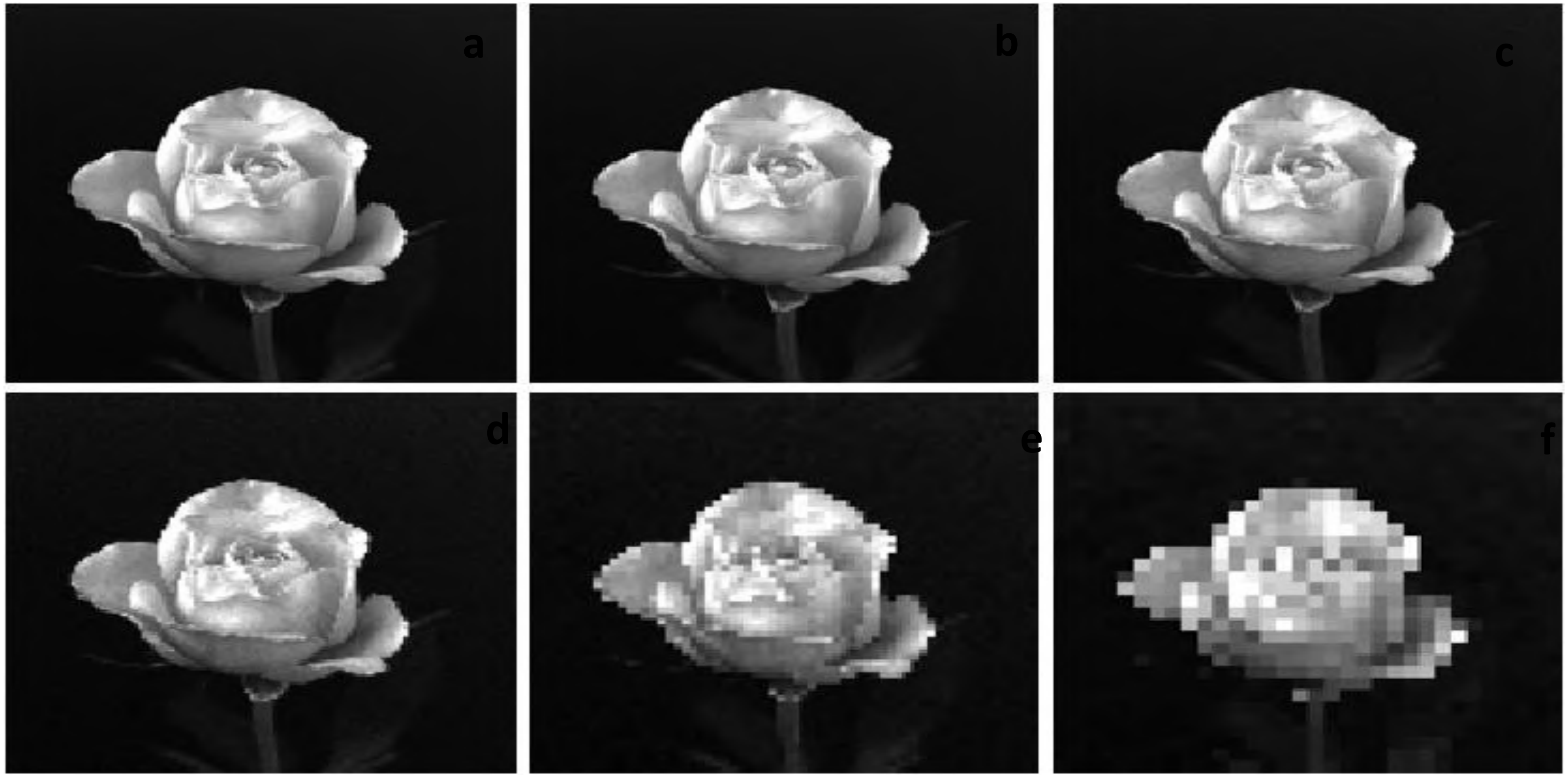
- Dimension proportion to the sampling density, but their size difference make it difficult to see the effects resulting from reduction in the number of sample.

A 1024*1024, 8-bit image sub-sampled down to size 32*32 pixels. The number of allowable gray levels was **kept at 256**. Sub-sampling is achieved by deleting every other row and column.

Resolution (Spatial)

Image re-sampled into 1024*1024 pixels by row and column duplication

Check Board Effect



(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.

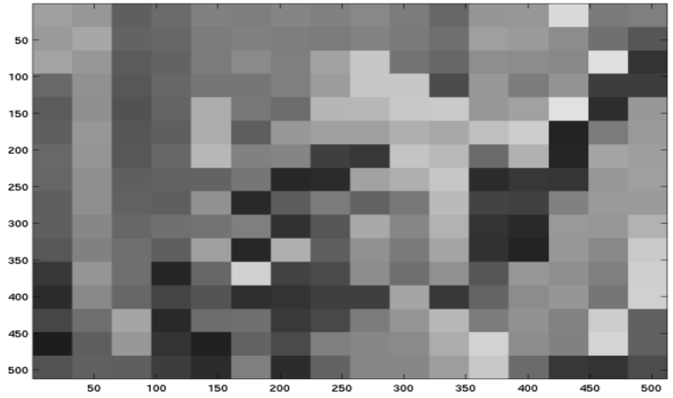
Resolution (Spatial)

Original

Reduced by 2 in each direction

Reduced by 8 in each direction

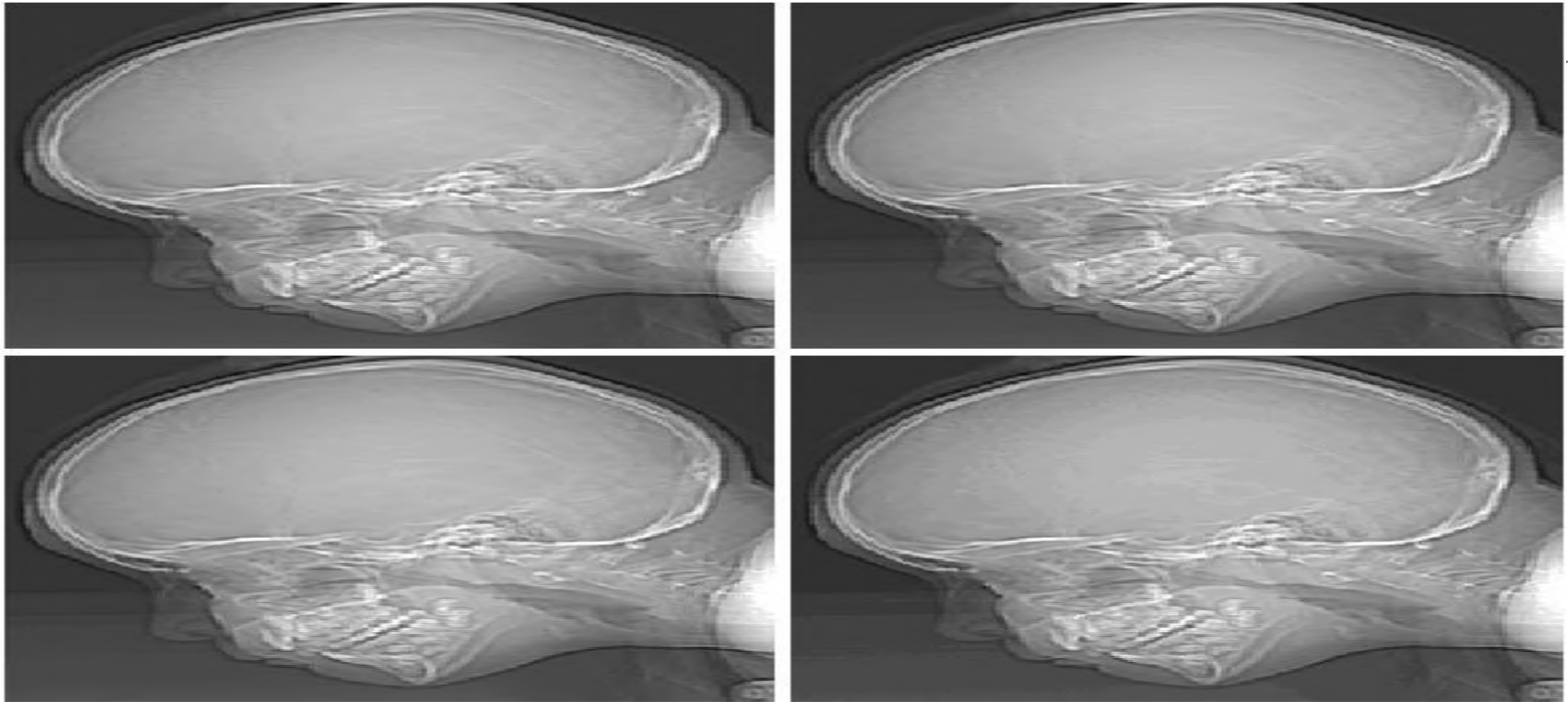
Reduced by 32 in each direction



Checkerboard Effect

Resolution (Gray -Level)

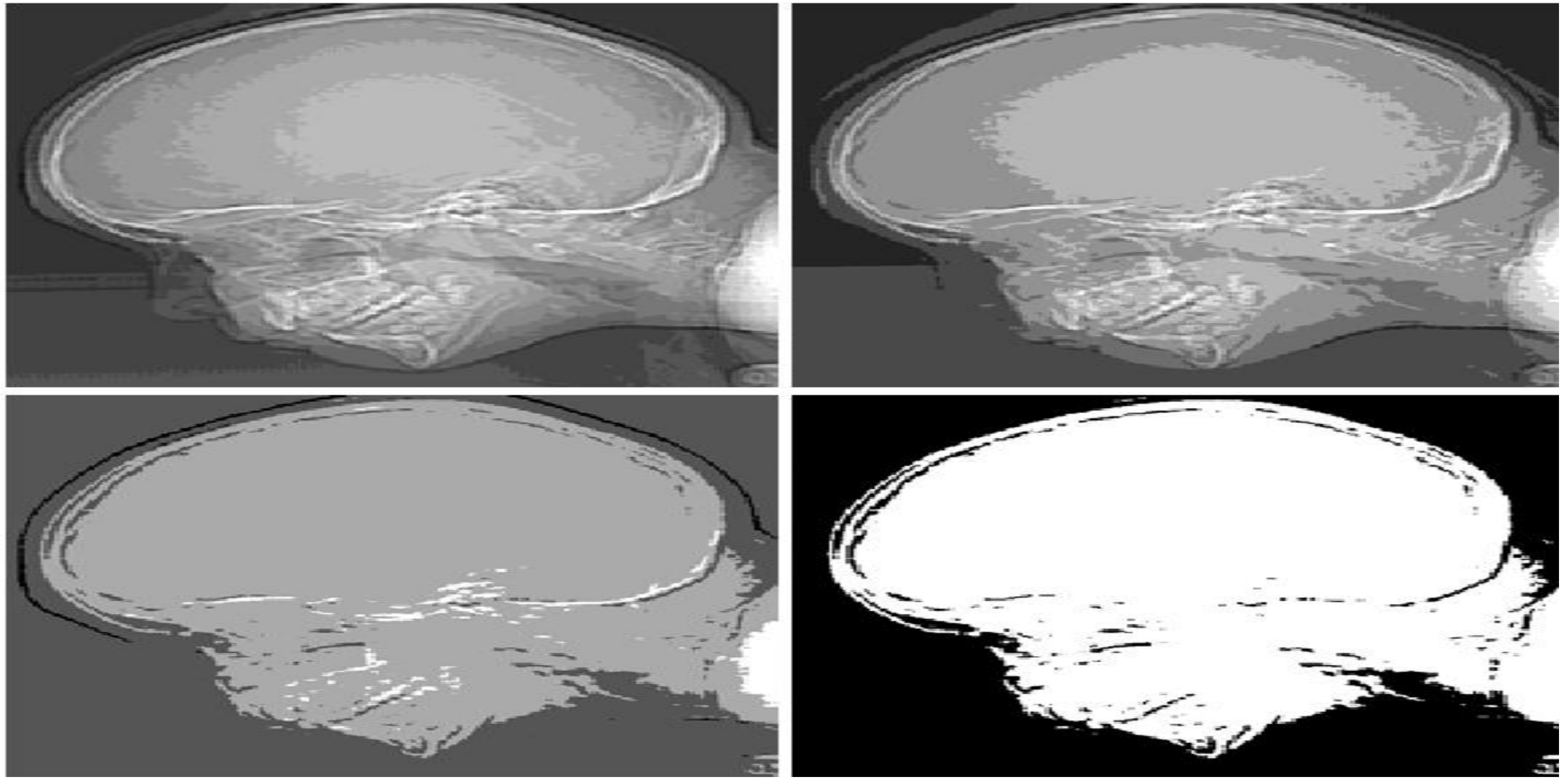
Keep the number of samples constant and reduce the number of gray levels from **256 to 32**.



(a) 452*374, 256-level image (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Resolution (Gray -Level)

Keep the number of samples constant and reduce the number of gray levels from **16 to 2**.



(e)–(h) Image displayed in 16, 8, 4 and 2 gray levels, while keeping the spatial resolution constant.

False Contouring Effect: imperceptible set of very fine ridge like structure in areas of smooth gray level.

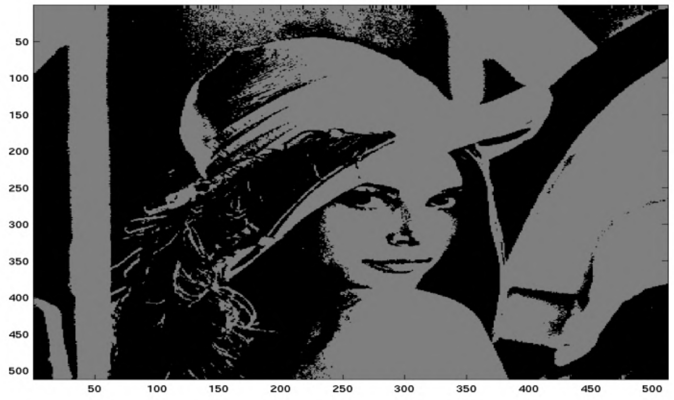
Resolution (Gray-Level)

Original (256 levels)

64 levels

4 levels

2 levels



Resolution (Gray-Level)

When the number of gray level values are reduced, very fine ridge like structures develop in the areas of gray levels.

This effect is known as ***false contouring*** and is caused by the insufficient number of gray levels in smooth areas of the image.

Suggested Readings

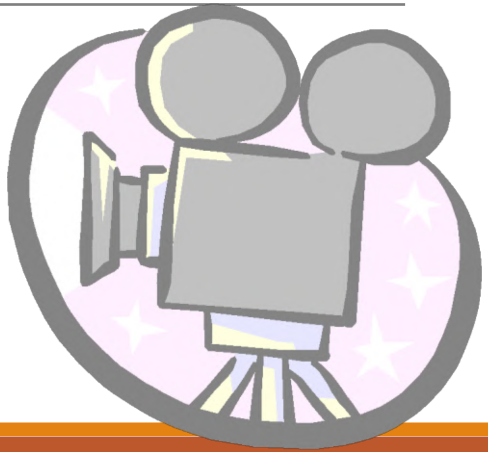
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



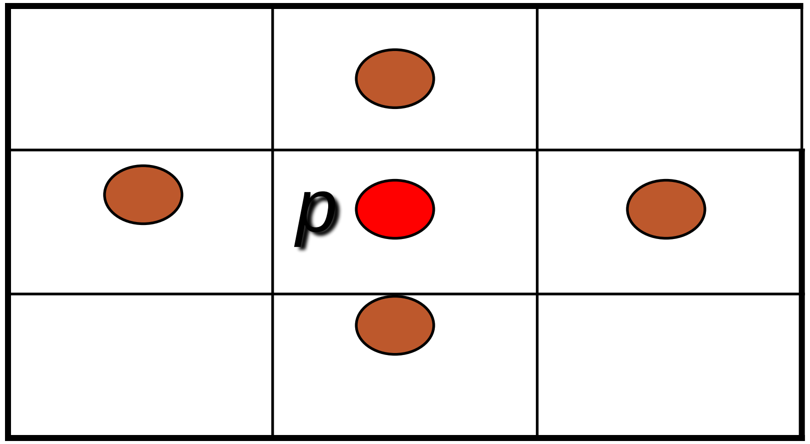
Outline

- Relationship between pixels
- Connectivity
- Distance measure

Relationship between pixels

Neighbours of a Pixel:

A pixel ***p*** at coordinates ***(x,y)*** has ***four(4) horizontal and vertical*** neighbours.



The coordinates of these neighbours are given by

(x+1,y), (x-1,y), (x,y+1) and (x,y-1)

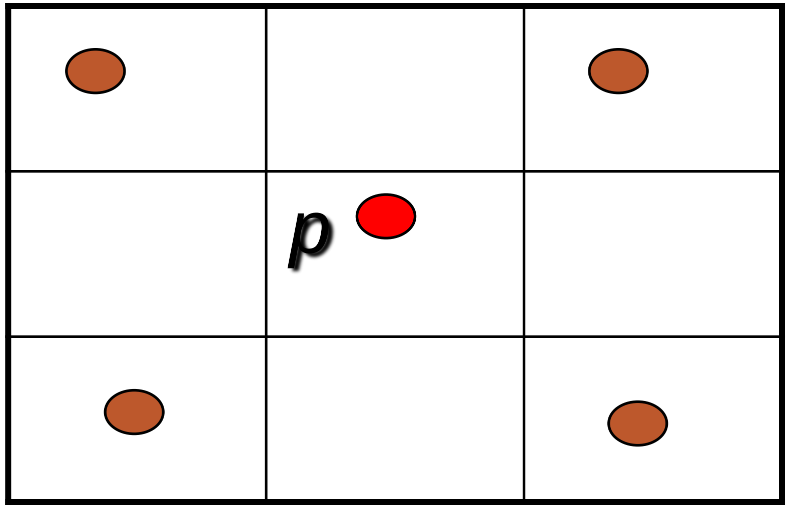
The above set of pixels is called the ***4-neighbours of p –*** $N_4(p)$

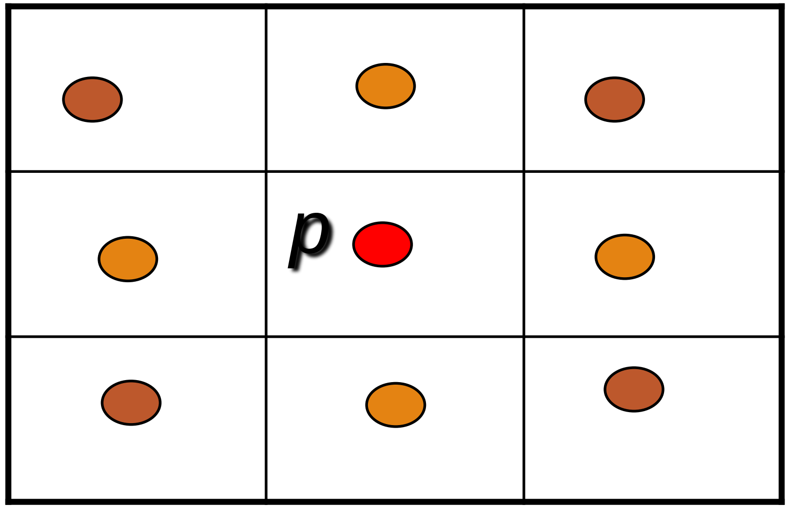
Each pixel is a unit distance from ***(x,y)***

Neighbors of a Pixel

The ***four (4) diagonal neighbors*** of ***p*** have coordinates

(x+1,y+1),(x+1,y-1),(x-1,y+1) and (x-1,y-1)





$N_8(p)$

The above set of pixels is denoted by – $N_D(p)$.

These pixels, together with 4-neighbours are called ***8-neighbours of p*** and is denoted by $N_8(p)$.

$N_8(p) = N_4(p) \cup N_D(p)$.

Connectivity

Connectivity forms the basis for establishing the ***boundaries of an objects*** and also components of regions in an image.

To establish whether two pixels are connected:

1. Whether the ***pixels are adjacent*** (e.g. are they 4-neighbours)
2. Whether their gray levels satisfy ***a specified criterion of similarity*** (e.g. equal or belongs to a set – falls within a given range of gray level)

Connectivity (Cont..)

- ***4-connectivity*** – Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$. V is the set of gray level.
- ***8 connectivity*** – Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$
- ***m-connectivity*** - Two pixels p and q with values from V are m-adjacent if:
  1. q is in $N_4(p)$ or
  2. q is in $N_D(p)$ ***and*** the intersection of ($N_4(p)$ and $N_4(q)$) is empty

Distance Measures

- For pixels p,q, and z with coordinates (x,y),(s,t) and (v,w), respectively, D is a distance functions if:

$$(a)\ D(p,q) \geq 0 \ \ (D(p,q) = 0 \text{ iff } \ p = q)$$

$$(b)\ D(p,q) = D(q,p), \text{ and}$$

$$(c)\ D(p,z) \leq D(p,q) + D(q,z)$$

Euclidean Distance between p and q is defined as :

$$D_e(p,q) = [x-s]^2 + (y-t)^2]^{1/2}$$

City-Block Distance (D_4) between p and q is defined as

$$D_4(p,q) = |x-s| + |y-t|$$

Distance Measures …

Chess Board Distance (D_8) between p and q is defined as

$$D_8(p,q) = \max(|x-s|,|y-t|)$$

D_e – Pixels having a distance less than or equal to some value ***r*** from ***(x,y)*** are the points contained in a disk of radius r centered at ***(x,y).***

D_4 – Pixels having a ***D_4*** distance from ***(x,y)*** less than or equal to some value ***r*** form a diamond centered at ***(x,y).***

D_8 – Pixels having a ***D_8*** distance from ***(x,y)*** less than or equal to some value ***r*** form a square centered at ***(x,y).***

Suggested Readings

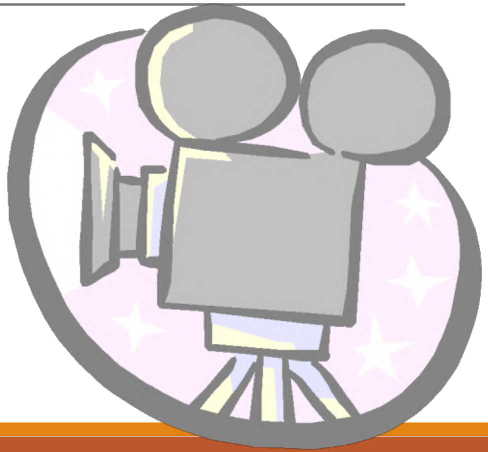
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Array vs. Matrix Operations
- Linear vs. Nonlinear operations
- Arithmetic Operations
- Logic Operations

Array vs. Matrix Operations

An array operation involving two or more images is carried out on a pixel by pixel basis. Consider the following 2 * 2 Images:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

The *array product* of these two images is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Note: Assume array operation throughout any image processing operation,.
Ex- raising to an image to a power, means each individual pixel raised to that power.

On the other hand, the matrix operation is given by:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Linear vs. Nonlinear Operations

One of the most important classification of an image processing method is whether it is linear or nonlinear. Consider a general operator, H that produces an output image, g(x, y), for a given input image, f(x, y):

$$H[f(x, y)] = g(x, y) \qquad \text{..............................(1)}$$

H is said to be linear operator if

$$\begin{aligned} H[a_i f_i(x, y) + a_j f_j(x, y)] &= a_i H[f_i(x, y)] + a_j H[f_j(x, y)] \\ &= a_i g_i(x, y) + a_j g_j(x, y) \end{aligned} \qquad \text{..............................(2)}$$

Where, a_i, $aj_,$ $f_i(x, y)$ *and* $f_j(x,y)$ are arbitrary constants and images (of the same size), respectively.

Linear vs. Nonlinear Operations

As a simple example, suppose that H is the sum operator $\sum$, that is the function of this operator is simply to sum its inputs. To test the linearity, start with the l.h.s of eq. 2

$$
\begin{aligned}
\sum\left[a_i f_i(x, y) + a_j f_j(x, y)\right] &= \sum a_i f_i(x, y) + \sum a_j f_j(x, y) \\
&= a_i \sum f_i(x, y) + a_j \sum f_j(x, y) \\
&= a_i g_i(x, y) + a_j g_j(x, y)
\end{aligned}
$$

On the other hand, consider max operation, whose function is to find maximum value of the pixels in an image.

Note: The simplest way to prove that a given function is nonlinear, is to find an example that fails the test in eq. 2.

Consider the following two images:

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

And suppose a1=1 and a2=-1, to test linearity start with L. H.S. of eq. 2

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\}$$
$$= -2$$

Working next with the R. H.S

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7$$
$$= -4$$

Arithmetic & Logic Operations

- Used extensively in Image Processing

- The arithmetic expressions between two pixels p and q are denoted as follows:

- Addition: ***p + q***
- Subtraction: ***p – q***
- Multiplication: ***p*q*** (also ***pq*** and ***p x q***)
- Division: ***p/q***

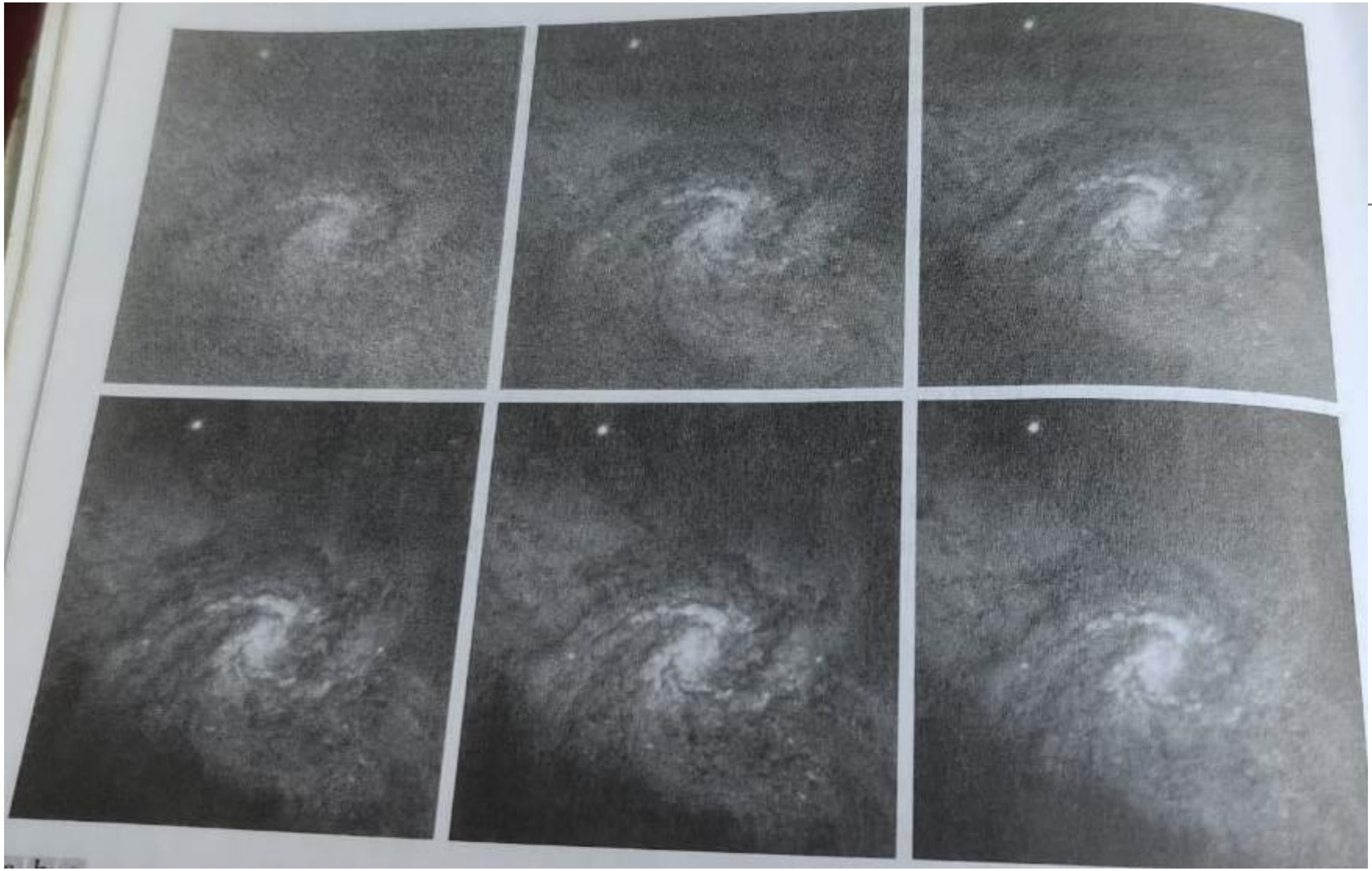
- Each operation on an entire image is carried out pixel by pixel.

- ***Image Addition*** : E.g. Image Averaging to reduce noise

- ***Image subtraction***: E.g. to remove static background information

- Image Multiplication/Division: correct gray level shading resulting from non-uniformities in illumination or in the sensor used to acquire the image.

Example

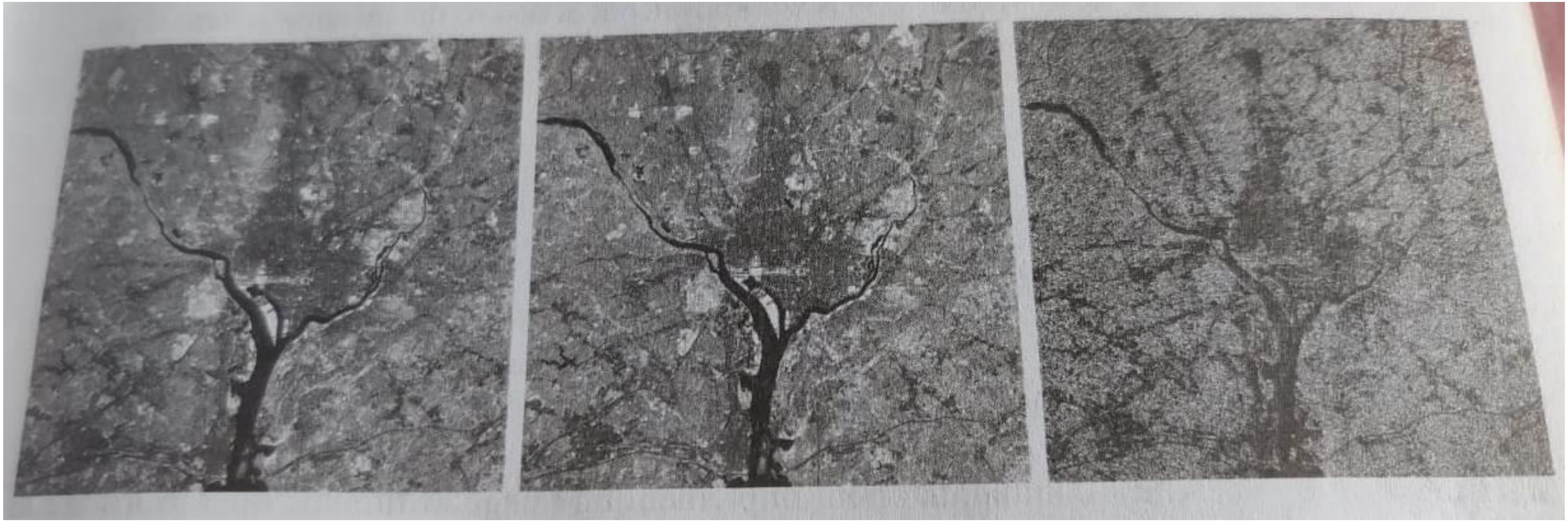


abc
def

a Image of Galaxy pair NGC 3314
Corrupted by additive Gaussian noise

b-f Results of averaging 5, 10, 20, 50, 100 noisy images respectively

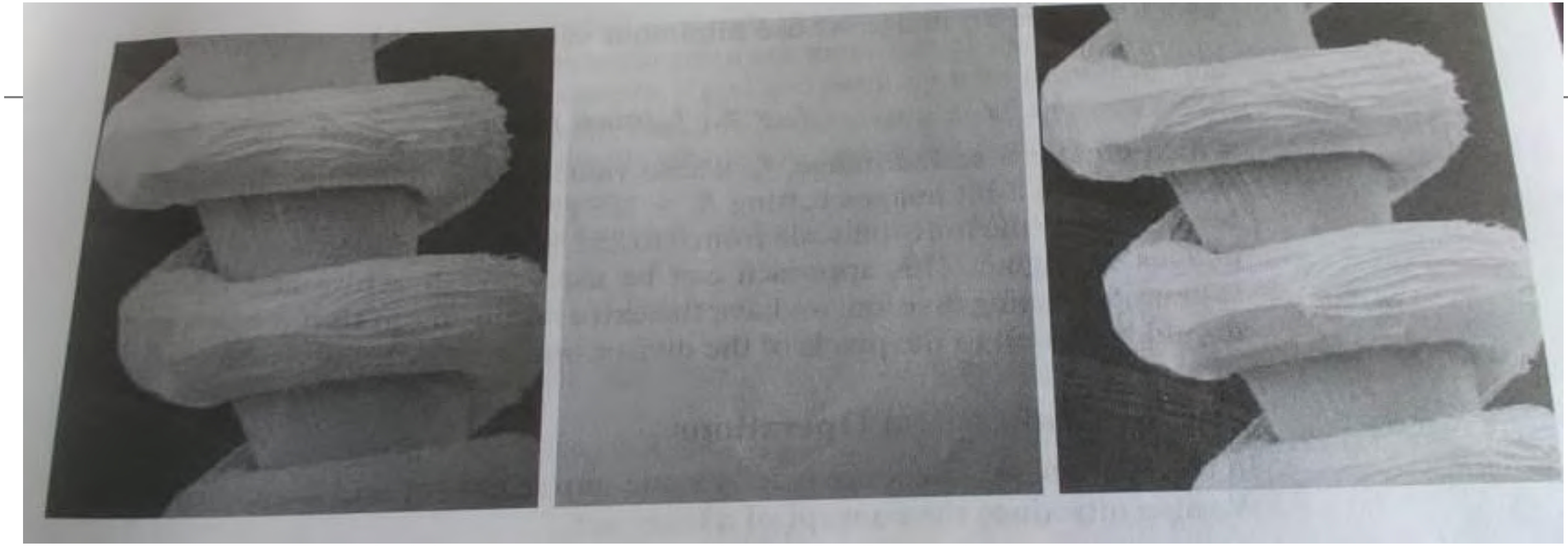
Example



(a) (b) (c)

a) Infrared image of Washington D. C. area b) image obtained by setting to zero the least significant bit of every pixel in (a)
(c) Difference of the two images, scaled to the range [0, 255 for clarity.

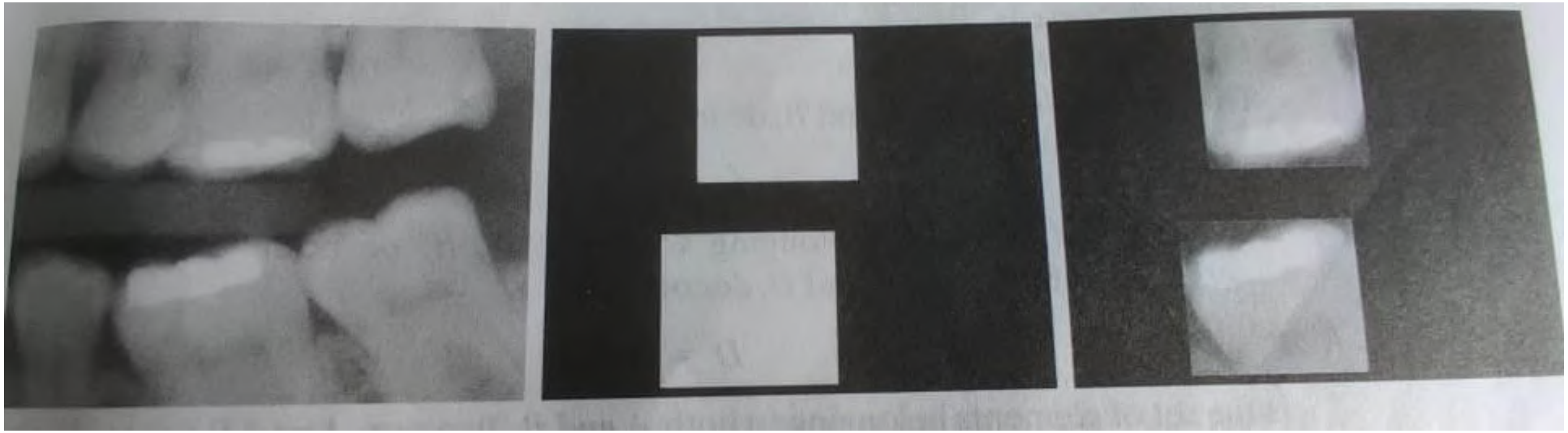
Example



(a) (b) (c)

Shading correction, (a) Shaded SEM image of a tungsten filament and support, magnified approximately 30 times
(b) The Shading pattern
(c) Product of (a) by the reciprocal of (b)

Example



(a) (b) (c)

(a) Digital Dental X-ray image.
(b) ROI mask for isolating teeth with fillings
(c) Product of (a) and (b)

Logic Operations

The principle logic operations used in Image Processing are ***AND, OR and COMPLEMENT***.

AND:	***p*** AND ***q*** (also ***p*** **.** ***q***)
OR:	***p*** OR ***q*** (also ***p*** + ***q***)
COMPLEMENT:	NOT ***q*** (also ***q***)

- Logic operations apply only to binary images, whereas arithmetic operations apply to multi-valued pixels.
- Logic operations are basic tools in binary image processing, where they are used for such tasks as masking, feature detection and shape analysis.
- Logic operations on entire images are done pixel by pixel.

Arithmetic & Logic Operations (A & L operations) – Neighborhood-oriented

- A& L operations are used ***neighbourhood oriented operations*** (in addition to pixel-by-pixel operations.

- Typically this takes the form of ***mask*** operations.
- The terms ***template, window*** and ***filter*** also are often used to denote a ***mask)***

Mask Operation: Let the value assigned to a pixel be a function of its gray level and gray level of its neighbours.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3 x 3 Sub Image

$$z = \frac{1}{9}(z_1 + z_2 + \cdots + z_9) = \frac{1}{9}\sum_{i=1}^{9} z_i$$

Replacing the value of z_5 with the average value of the pixels in 3 x 3 region centered at the pixel with value z_5

A & L operations ...

The same operation above, in generic terms, would be to replace a pixel with a weighted average of pixels in the region where weights (coefficients) can be different for different pixels.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$\mathbf{z} = (\mathbf{w_1 z_1} + \mathbf{w_2 z_2} + \cdots + \mathbf{w_9 z_9})$$

$$= \sum_{\mathbf{i=1}}^{\mathbf{9}} \mathbf{w_i z_i}$$

- Proper selection of the coefficients and application of the mask at each pixel position in an image makes possible a variety of useful image operations, such as noise reduction, region thinning, and edge detection.
- Computationally expensive. E.g applying a 3x3 mask to a 512x512 image requires nine multiplications and eight additions at each pixel location (total 2,359,296 multiplications and 2,097,152 additions)

Suggested Readings

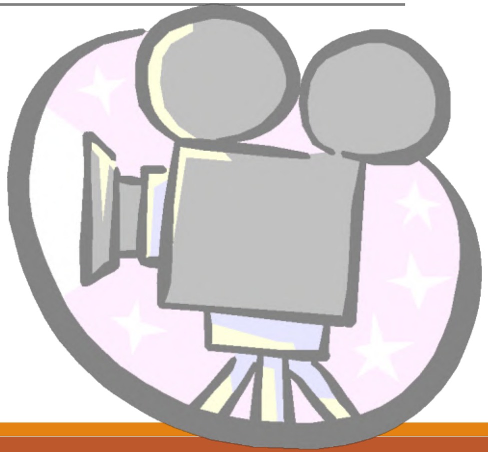
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Enhancement in spatial domain
- Some basic intensity transformation function
    - Image Negatives
    - Log Transformation
    - Power Law (Gamma) Transformation
- Piecewise Linear Transformation functions

Image Enhancement in the Spatial Domain

Principle Objective of Enhancement

Enhancement refers to accentuation, or sharpening, of image features such as edges, boundaries, or contrast to make graphic display more useful for display and analysis.

Does not increase the inherent information content in the data, only increase the dynamic range of chosen feature.

Principle Objective of Enhancement

Process an image so that the result will be more suitable than the original image for a specific application.

The suitableness is up to each application.

A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images

2 domains

Spatial Domain : (image plane)

- Techniques are based on direct manipulation of pixels in an image

Frequency Domain :

- Techniques are based on modifying the Fourier transform of an image

There are some enhancement techniques based on various combinations of methods from these two categories.

Image quality (Good or Bad?)

For human visual

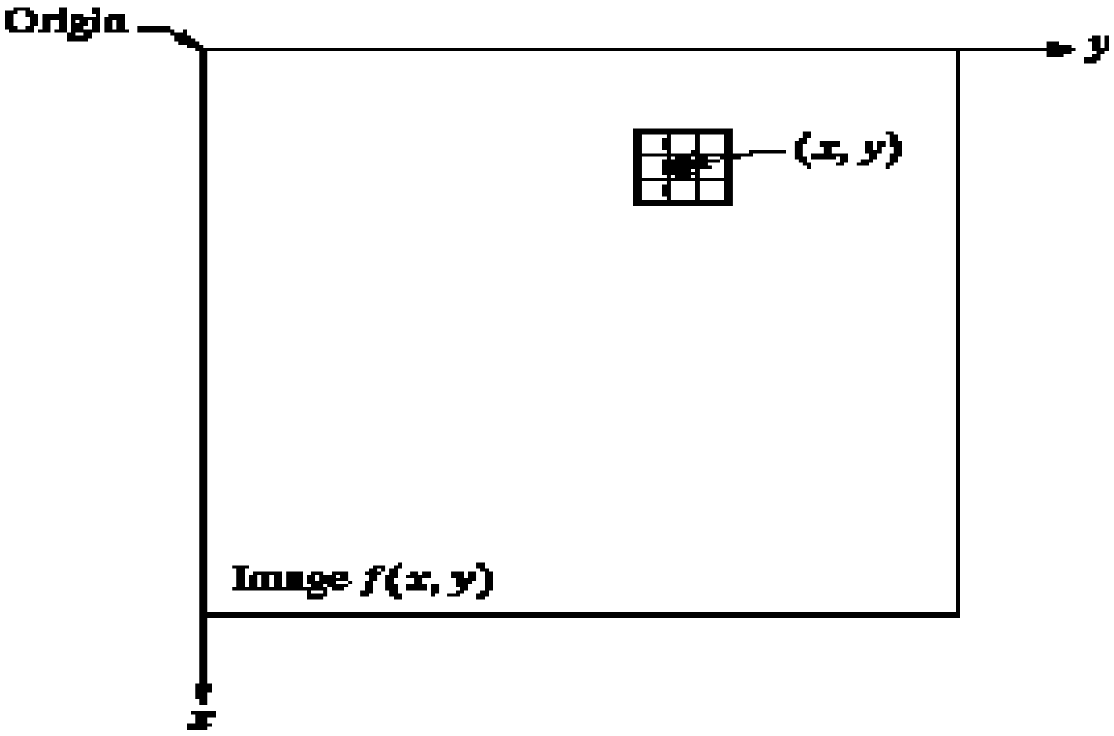
- The visual evaluation of image quality is a highly subjective process.
- It is hard to standardize the definition of a good image.

For machine perception

- The evaluation task is easier.
- A good image is one which gives the best machine recognition results.

A certain amount of trial and error usually is required before a particular image enhancement approach is selected.

Spatial Domain



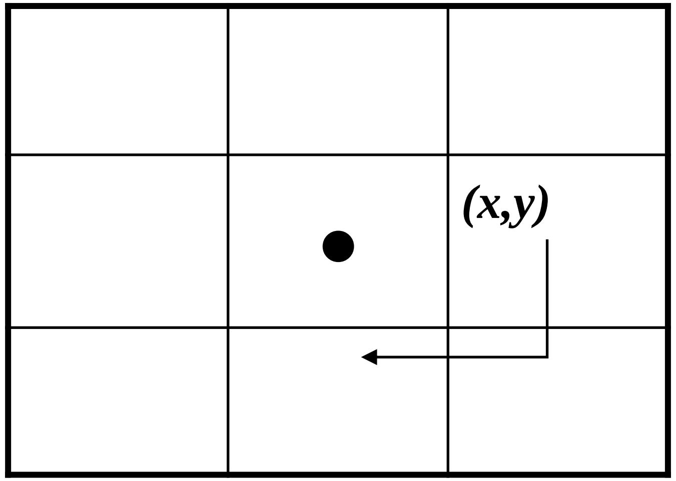
Procedures that operate directly on pixels.

$$g(x,y) = T[f(x,y)]$$

where

- $f(x,y)$ is the input image
- $g(x,y)$ is the processed image
- T is an operator on f defined over some neighborhood of (x,y)

Mask/Filter



Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) sub image area centered at (x, y).

The center of the sub image is moved from pixel to pixel starting at the top of the corner

Point Processing

Neighborhood = 1x1 pixel

g depends on only the value of f at (x,y)

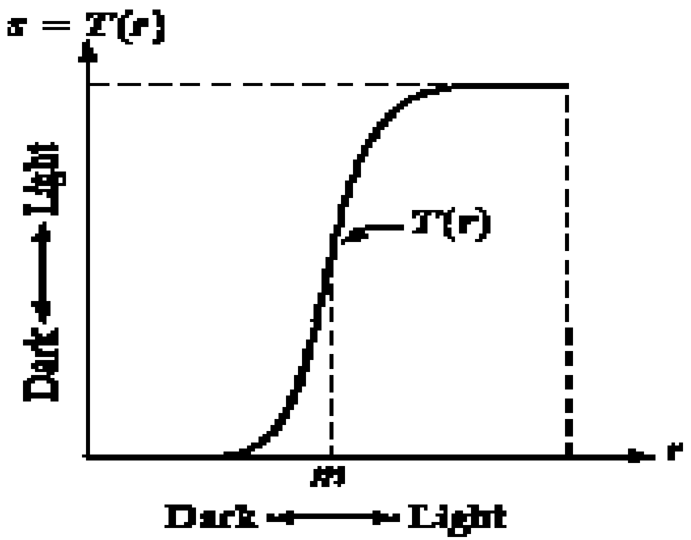
T = gray level (or intensity or mapping) transformation function

$$s = T(r)$$

Where

- r = gray level of $f(x,y)$
- s = gray level of $g(x,y)$

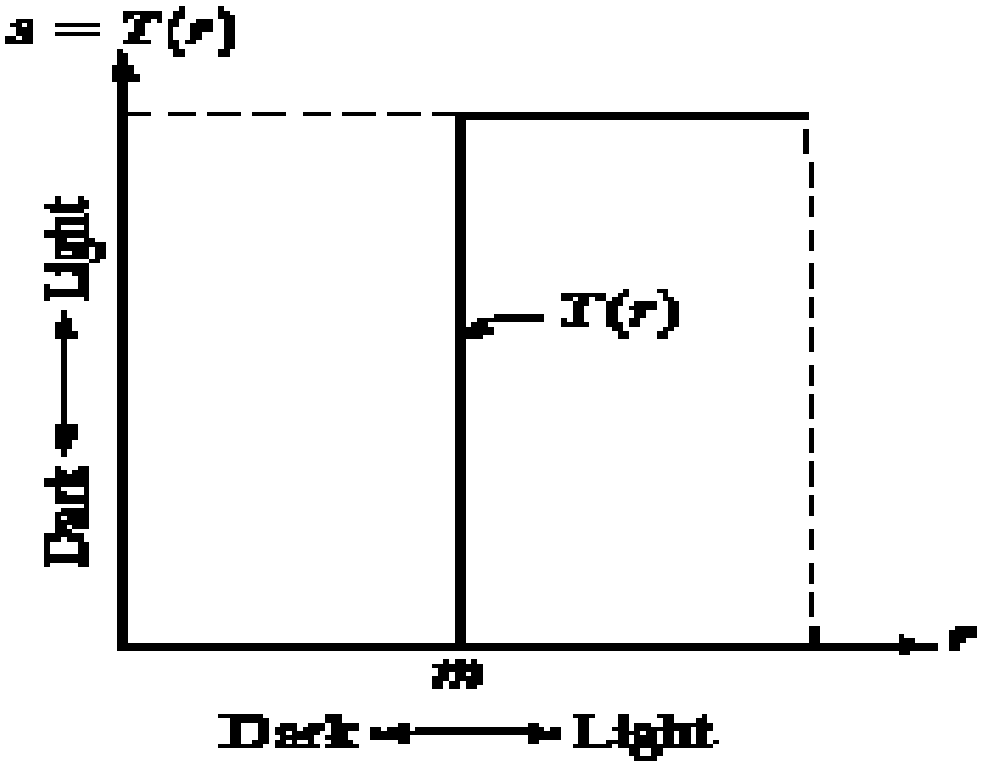
Contrast Stretching



Produce higher contrast than the original by

- darkening the intensity levels below m in the original image
- Brightening the levels above m in the original image

Thresholding



Produce a two-level (binary) image

Mask Processing or Filter

Neighborhood is bigger than 1x1 pixel

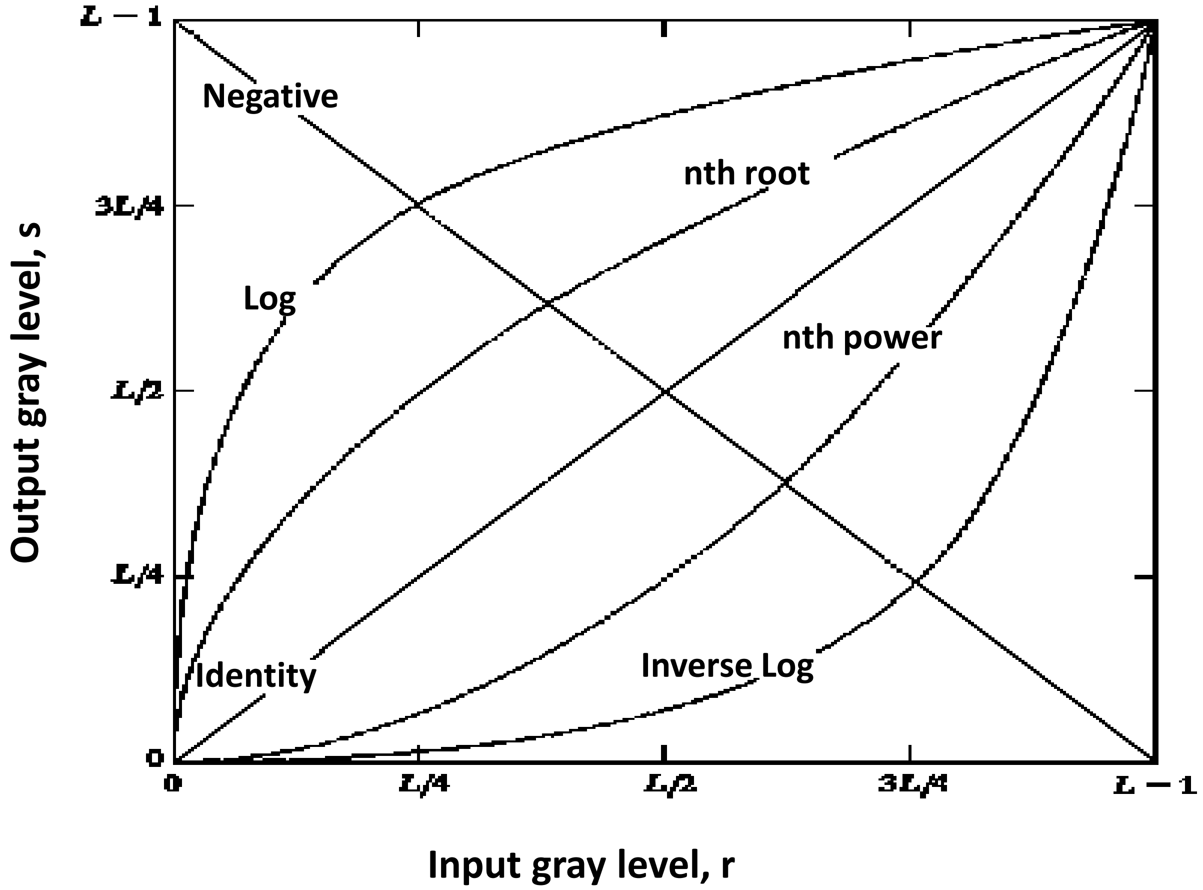
Use a function of the values of f in a predefined neighborhood of (x,y) to determine the value of g at (x,y)

The value of the mask coefficients determine the nature of the process

Used in techniques

- Image Sharpening
- Image Smoothing

3 basic gray-level transformation functions

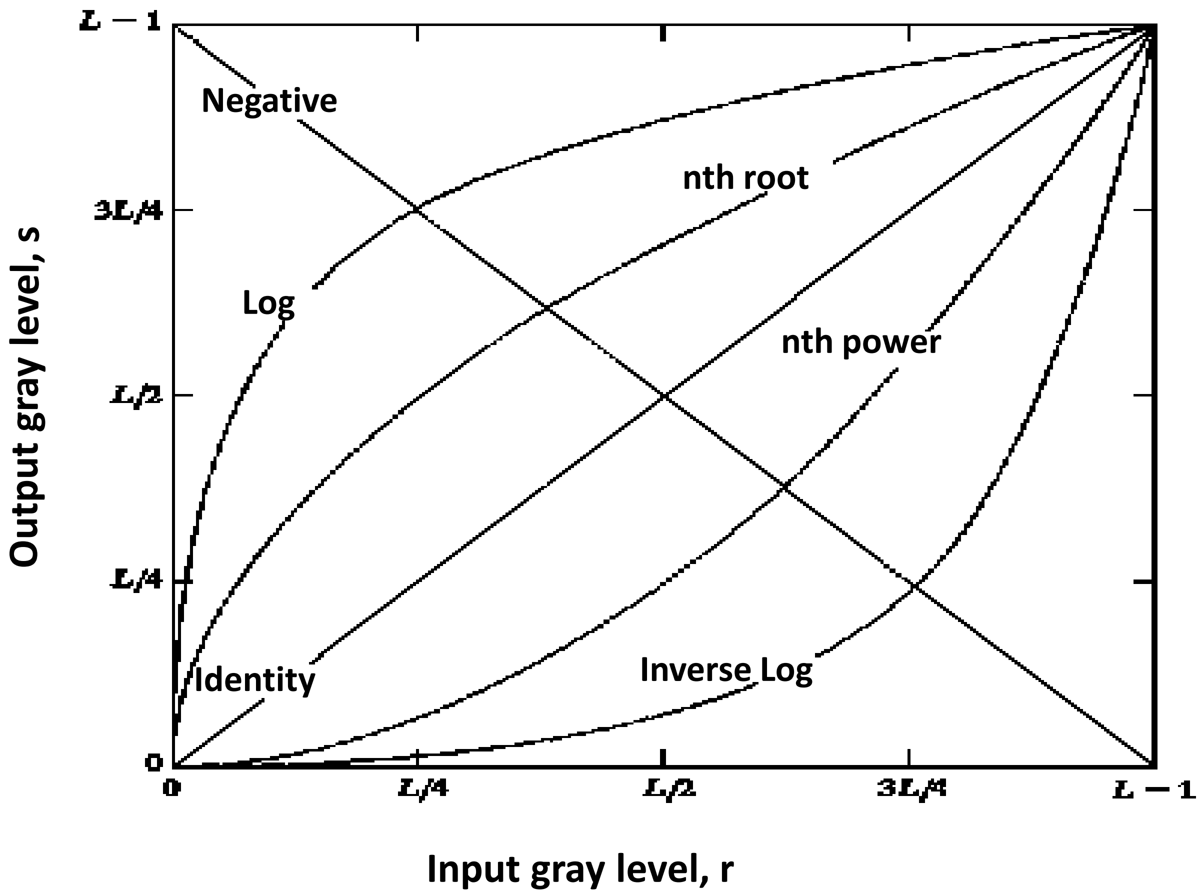


Linear function
- Negative and identity transformations

Logarithm function
- Log and inverse-log transformation

Power-law function
- n^{th} power and n^{th} root transformations

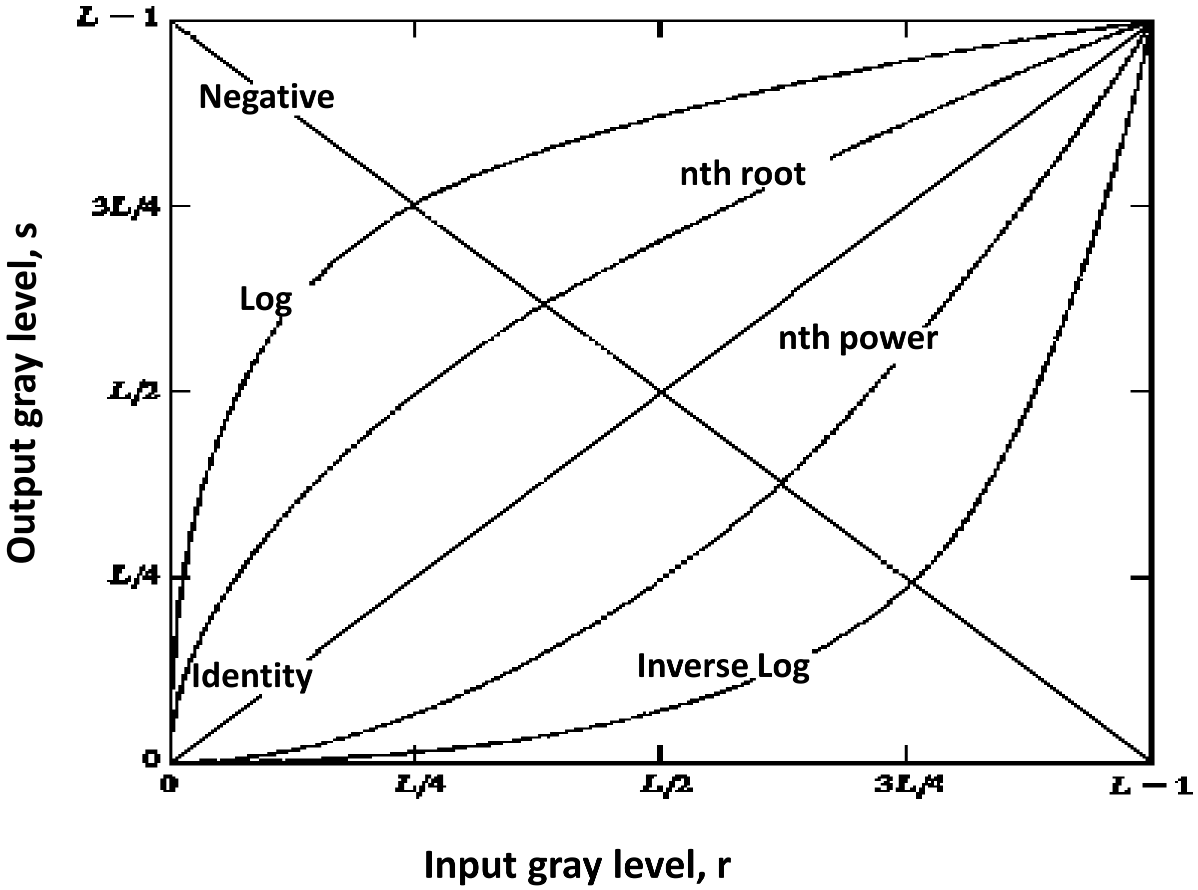
Identity function



Output intensities are identical to input intensities.

Is included in the graph only for completeness.

Image Negatives



An image with gray level in the range **[0,L-1]** where **$L = 2^n$** ; n = 1, 2…

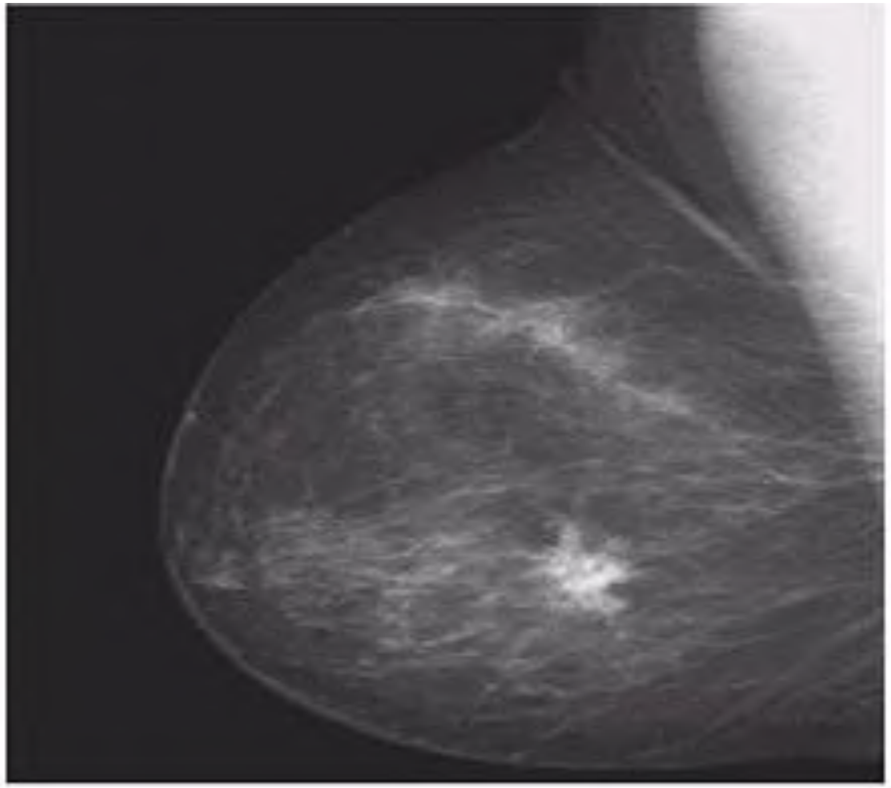
Negative transformation :

$$s = L - 1 - r$$

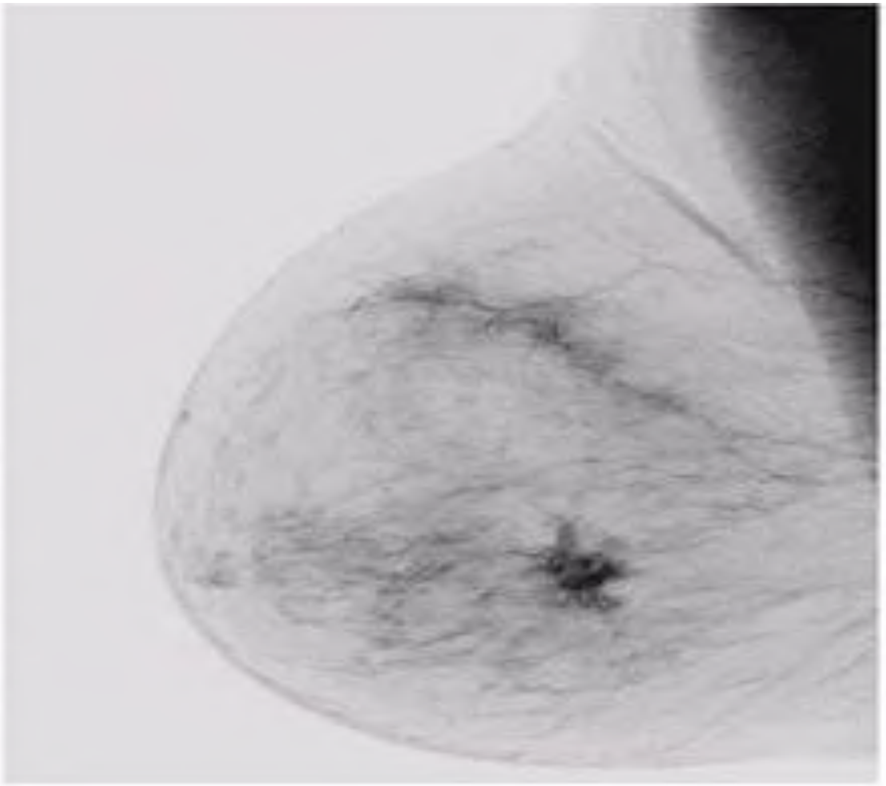
Reversing the intensity levels of an image.

Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.

Example of Negative Image

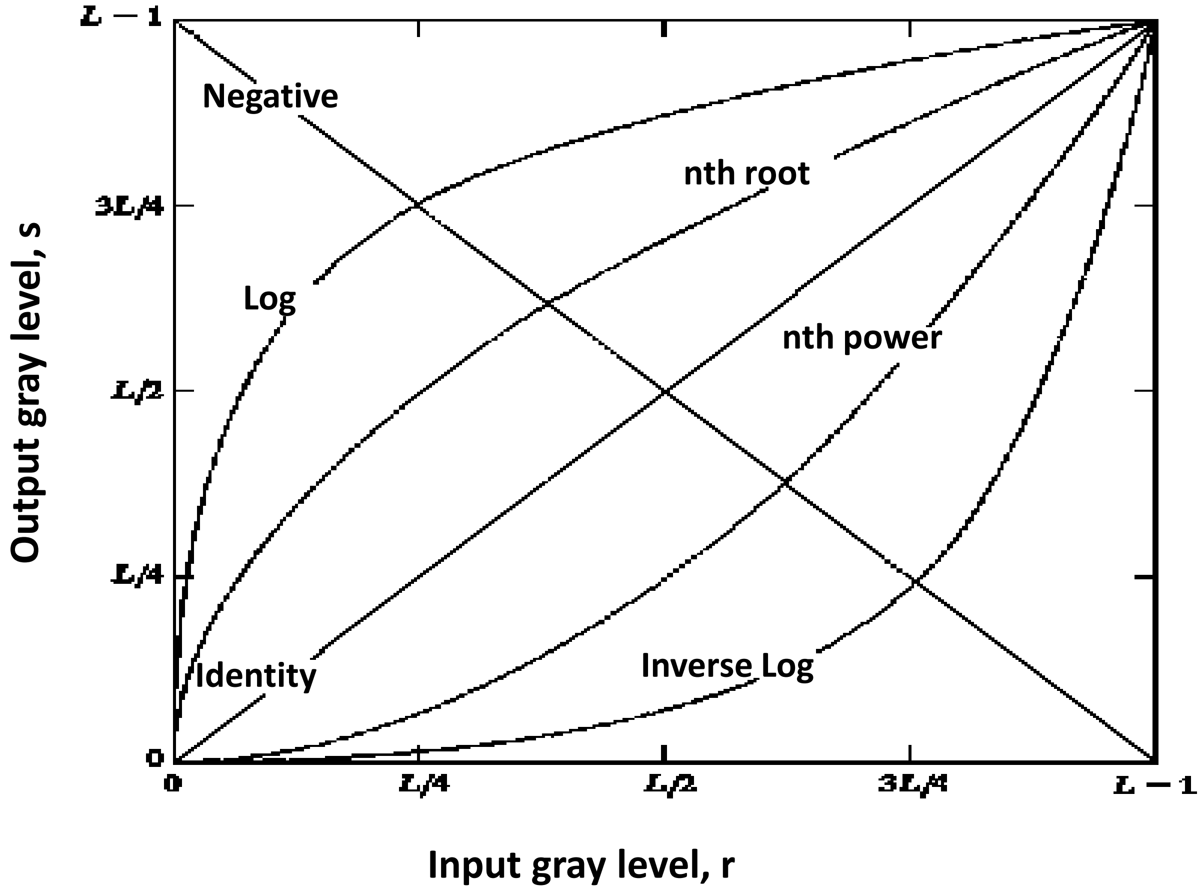


Original mammogram showing a small lesion of a breast



Negative Image : gives a better vision to analyze the image

Log Transformations



$$s = c \log(1+r)$$

c is a constant and $r \geq 0$

Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.

Used to expand the values of dark pixels in an image while compressing the higher-level values.

Log Transformations

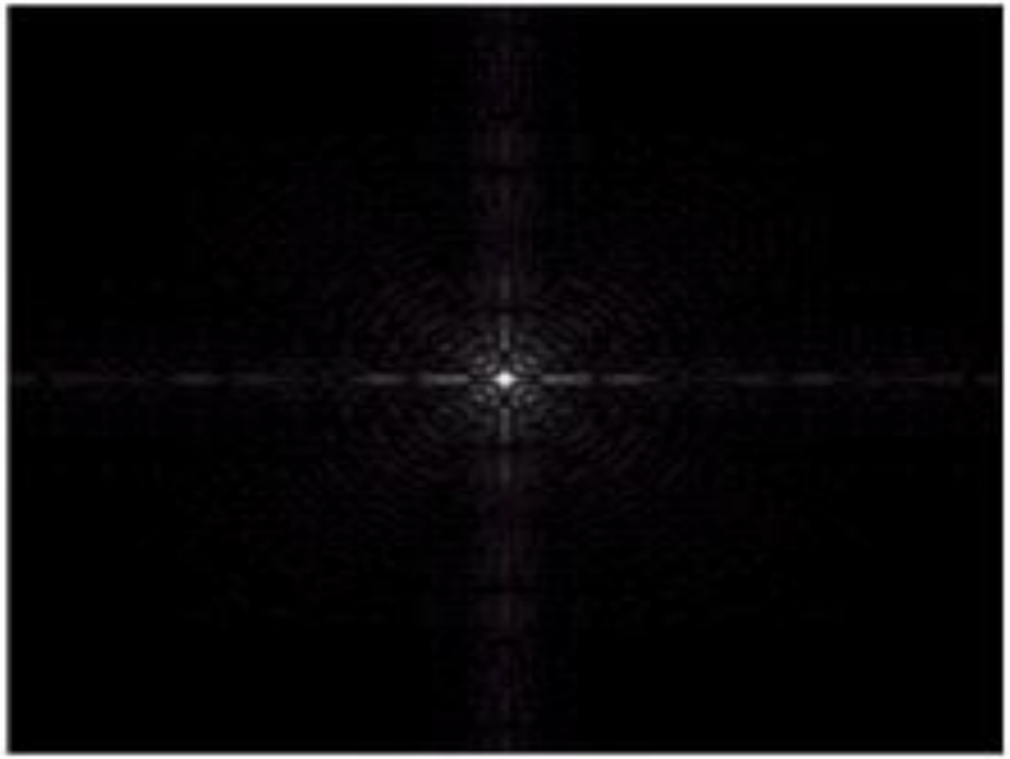
It compresses the dynamic range of images with large variations in pixel values

Example of image with dynamic range: Fourier spectrum image

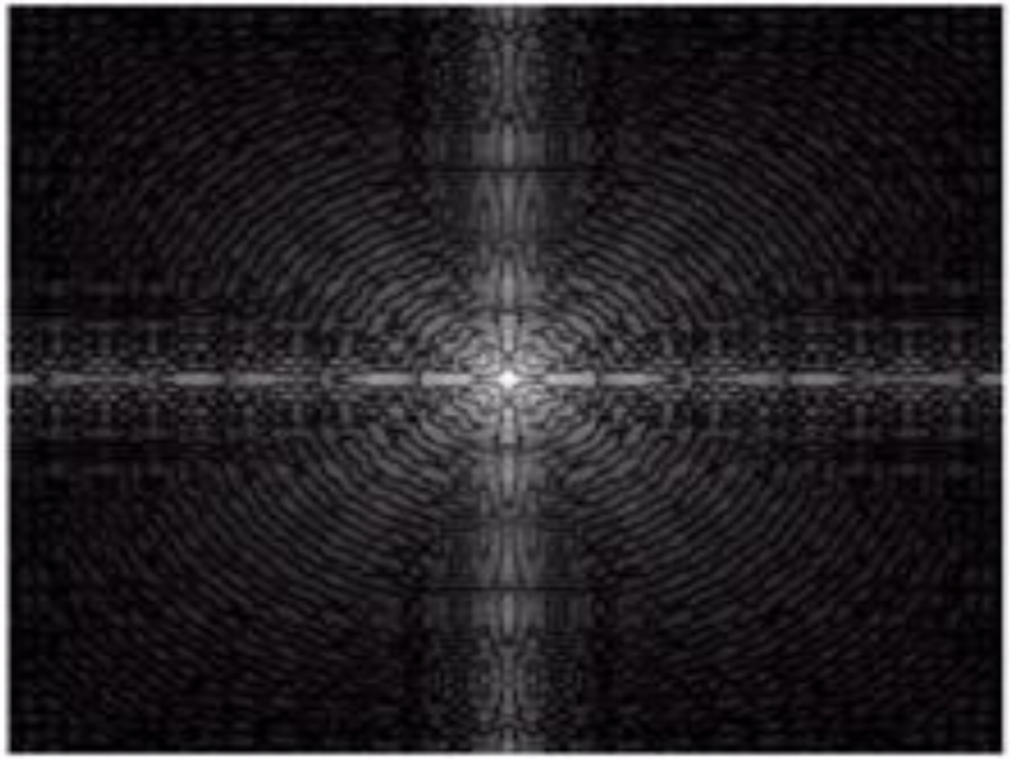
It can have intensity range from 0 to 10^6 or higher.

We can't see the significant degree of detail as it will be lost in the display.

Example of Logarithm Image



Fourier Spectrum with range = 0 to 1.5 x 10^6



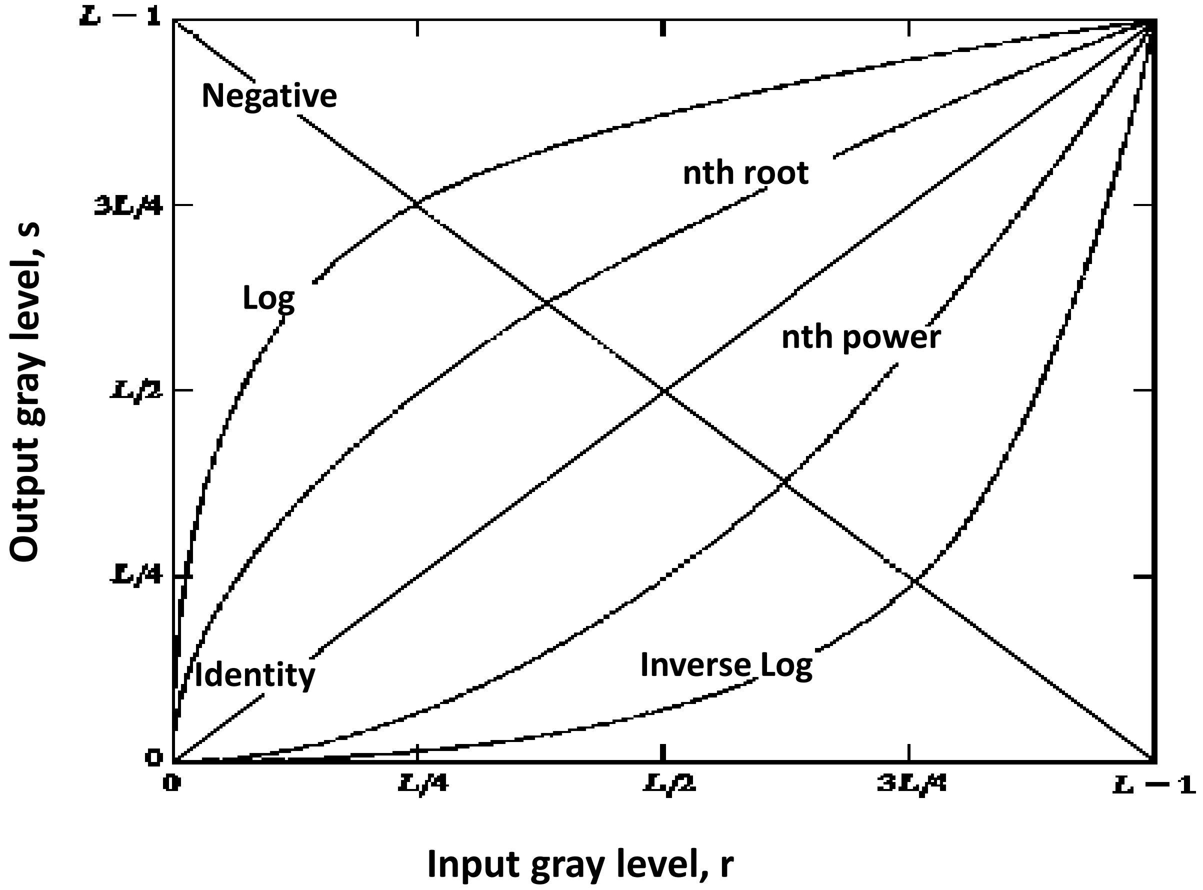
Result after apply the log transformation with c = 1, range = 0 to 6.2

Inverse Logarithm Transformations

Do opposite to the Log Transformations

Used to expand the values of high pixels in an image while compressing the darker-level values.

Power-Law Transformations



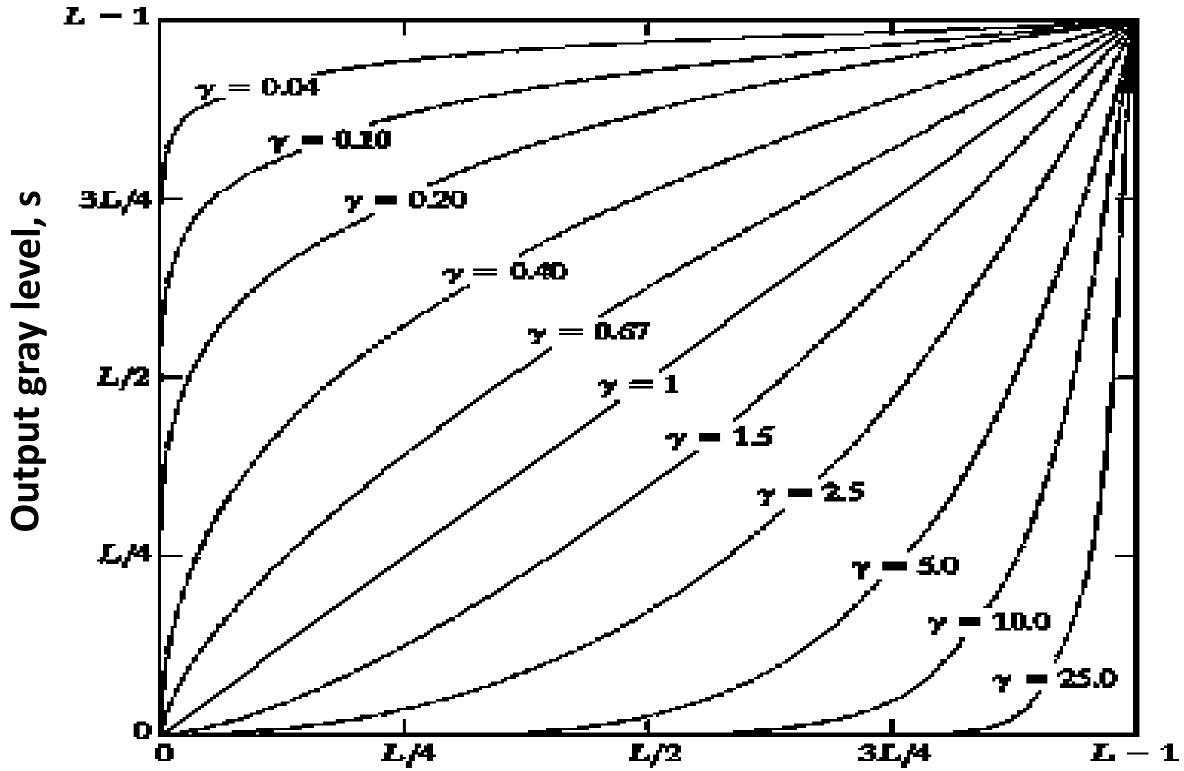
$$s = cr^{\gamma}$$

c and γ are positive constants

$\gamma > 1$ n^{th} power transformations

$\gamma < 1$ n^{th} root transformations

Power-Law Transformations



Plots of $s = cr^{\gamma}$ for various values of γ
(c = 1 in all cases)

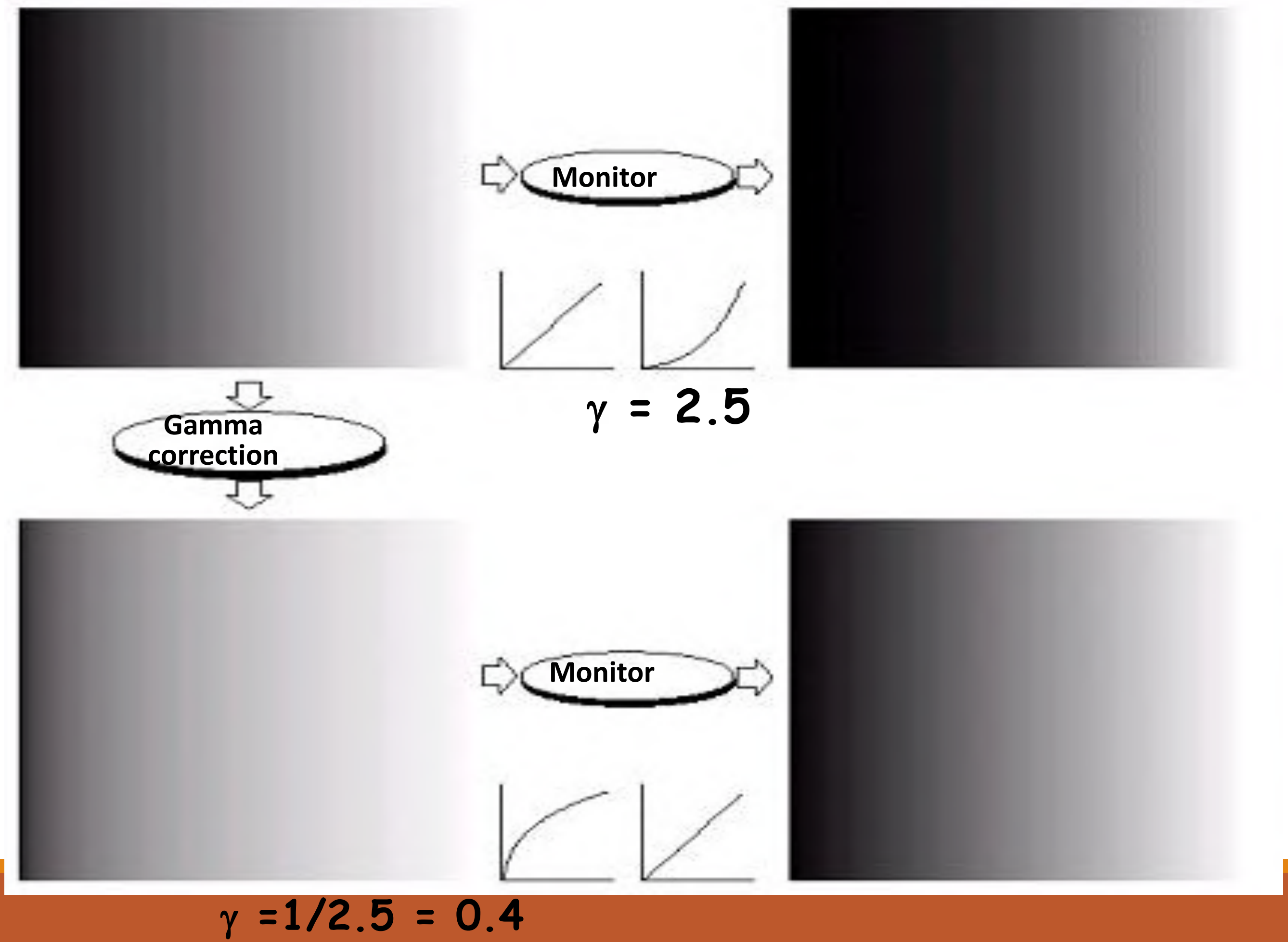
$$s = cr^{\gamma}$$

c and γ are positive constants

Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.

c = γ = 1 ⇨ Identity function

Gamma correction



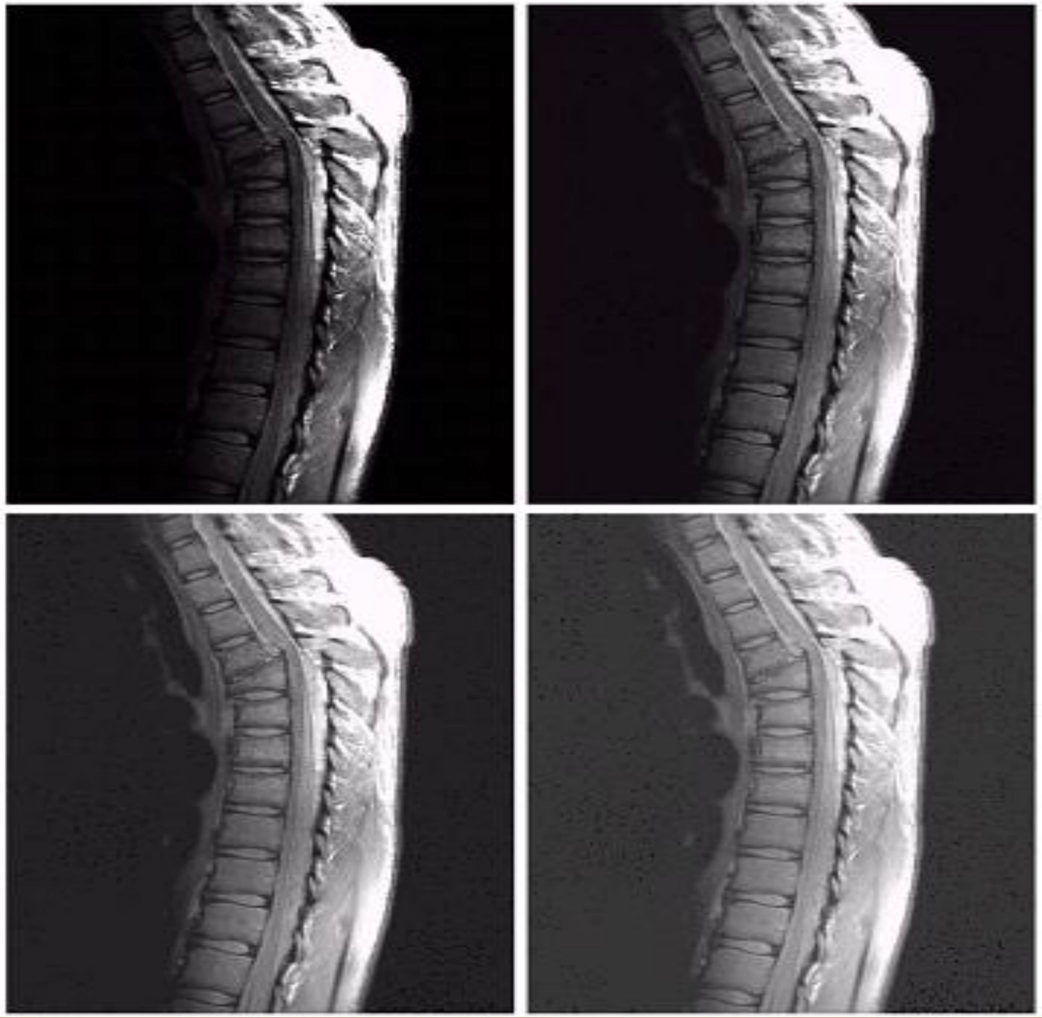
Cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with γ varying from 1.8 to 2.5

The picture will become darker.

Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = cr^{1/\gamma}$

Another example : MRI

a	b
c	d



(a) a magnetic resonance image of an upper thoracic human spine with a fracture dislocation and spinal cord impingement

- The picture is predominately dark
- An expansion of gray levels are desirable ⇨ needs $\gamma < 1$

(b) result after power-law

transformation with $\gamma = 0.6$, c=1

(c) transformation with $\gamma = 0.4$

(best result)

(d) transformation with $\gamma = 0.3$

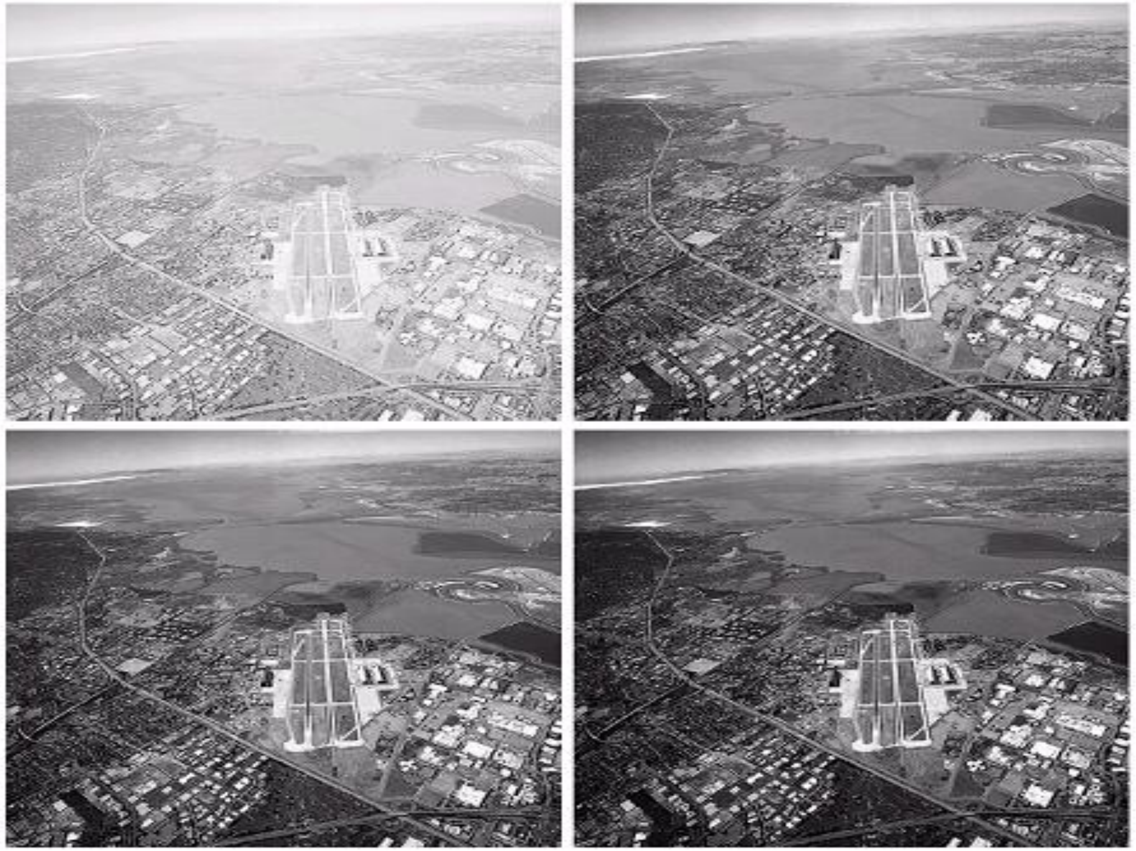
(under acceptable level)

Effect of decreasing gamma

When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight "wash-out" look, especially in the background

Another example

a	b
c	d



(a) image has a washed-out appearance, it needs a compression of gray levels ⇨ needs $\gamma > 1$

(b) result after power-law

transformation with $\gamma = 3.0$ (suitable)

(c) transformation with $\gamma = 4.0$

(suitable)

(d) transformation with $\gamma = 5.0$

(high contrast, the image has areas that are too dark, some detail is lost)

Piecewise-Linear Transformation Functions

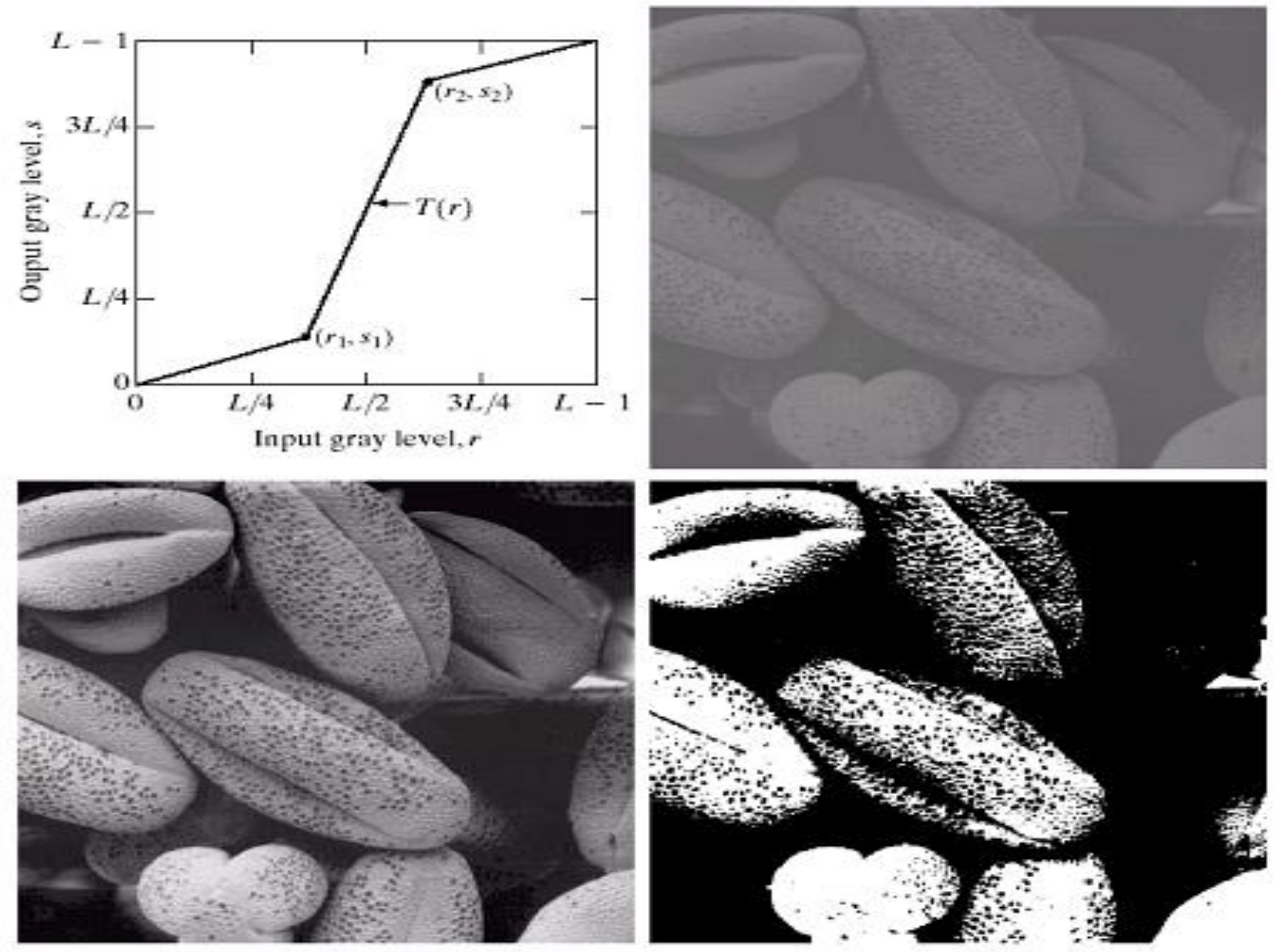
Advantage:

- The form of piecewise functions can be arbitrarily complex

Disadvantage:

- Their specification requires considerably more user input

Contrast Stretching



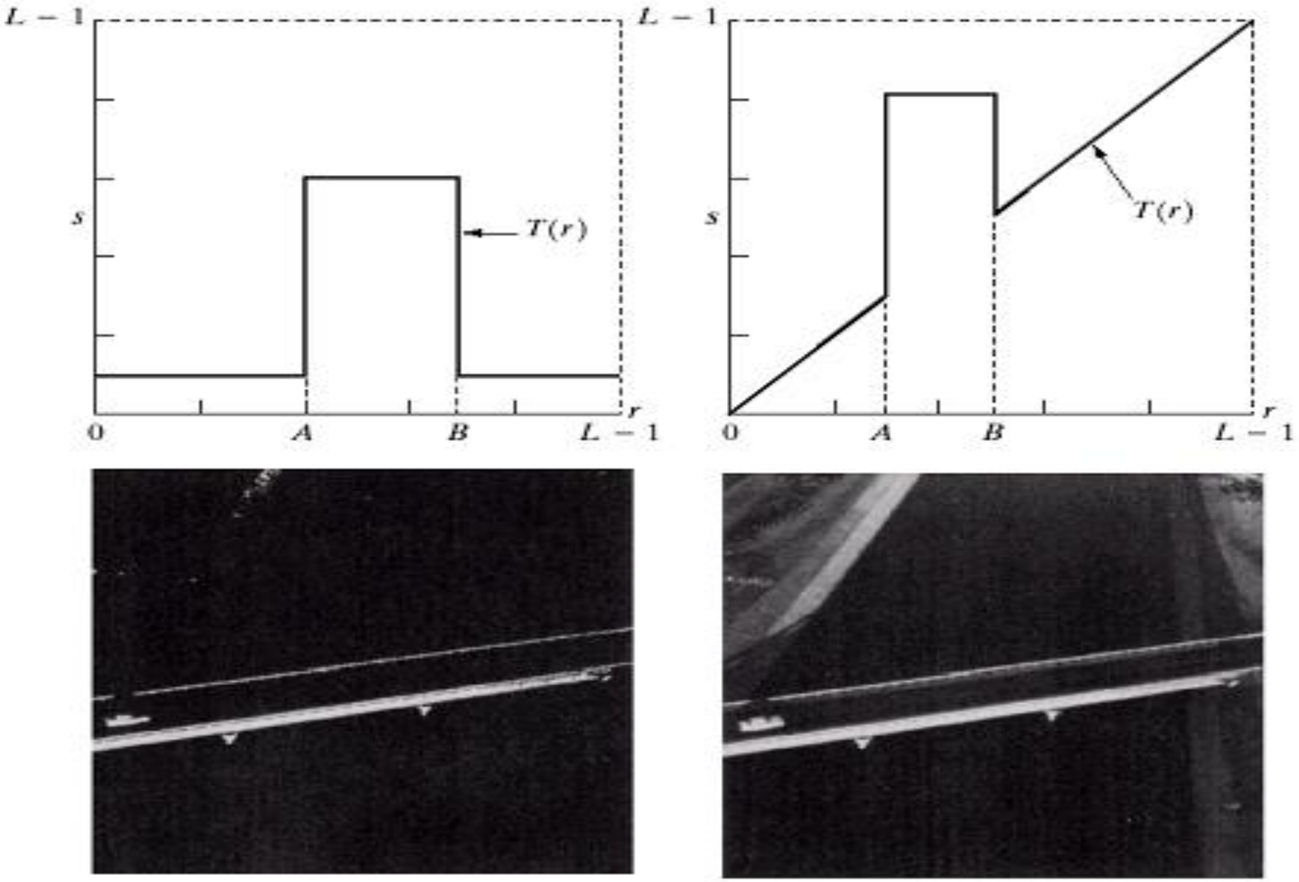
increase the dynamic range of the gray levels in the image

(b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition

(c) result of contrast stretching: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$

(d) result of thresholding

Gray-level slicing



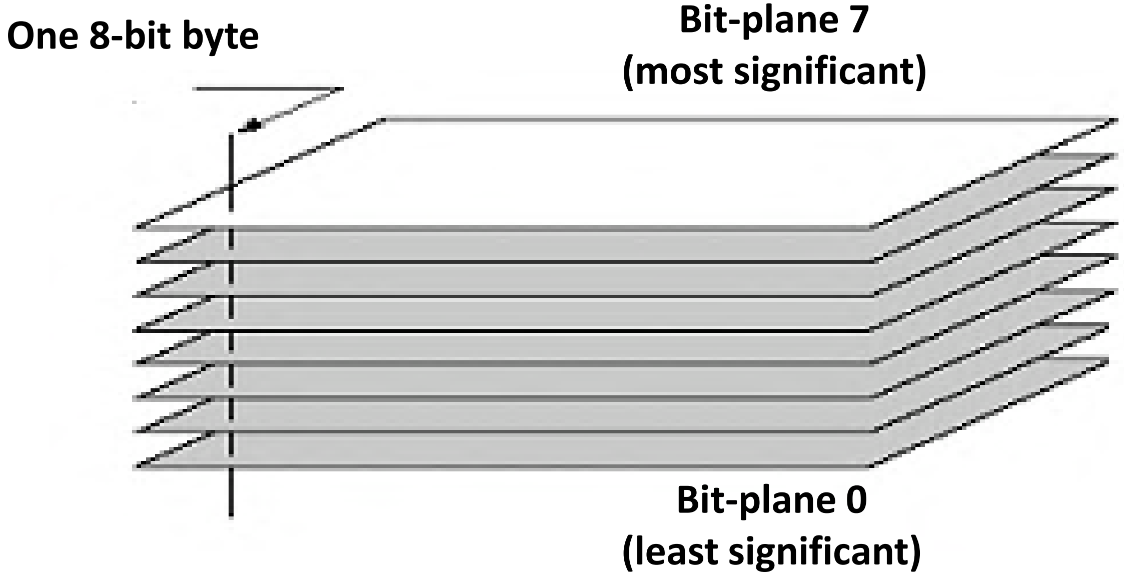
Highlighting a specific range of gray levels in an image

- Display a high value of all gray levels in the range of interest and a low value for all other gray levels

(a) transformation highlights range [A,B] of gray level and reduces all others to a constant level

(b) transformation highlights range [A,B] but preserves all other levels

Bit-plane slicing



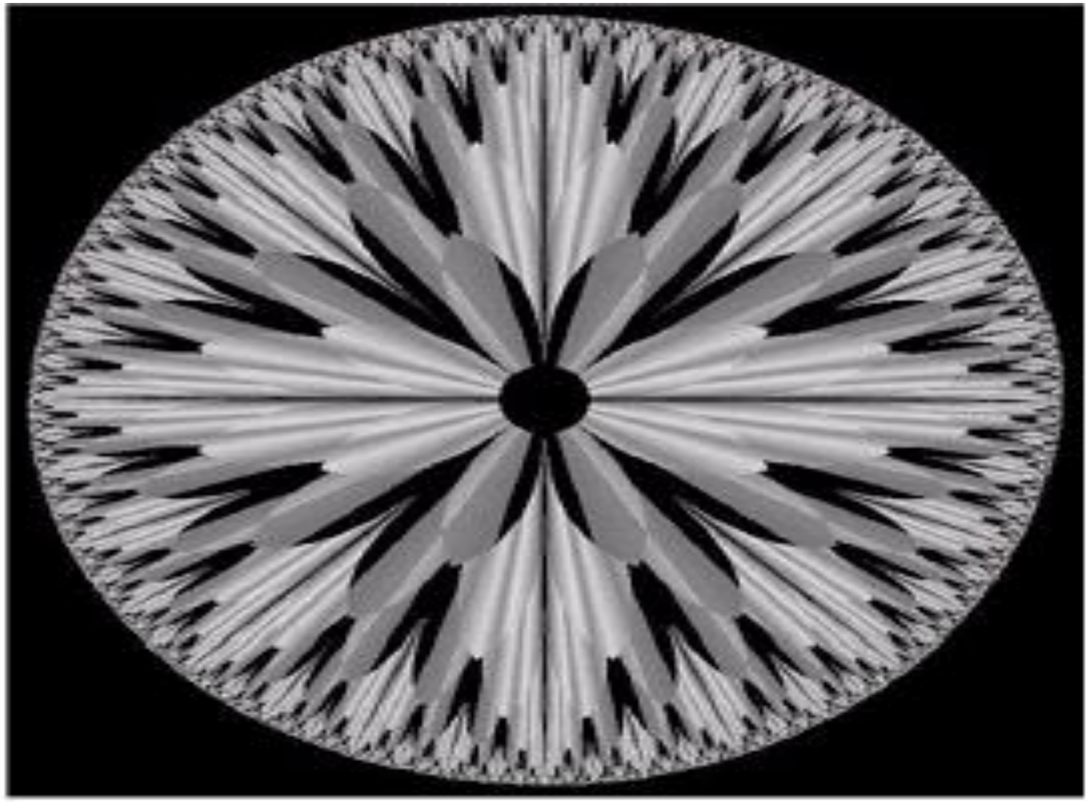
Highlighting the contribution made to total image appearance by specific bits

Suppose each pixel is represented by 8 bits

Higher-order bits contain the majority of the visually significant data

Useful for analyzing the relative importance played by each bit of the image

Example

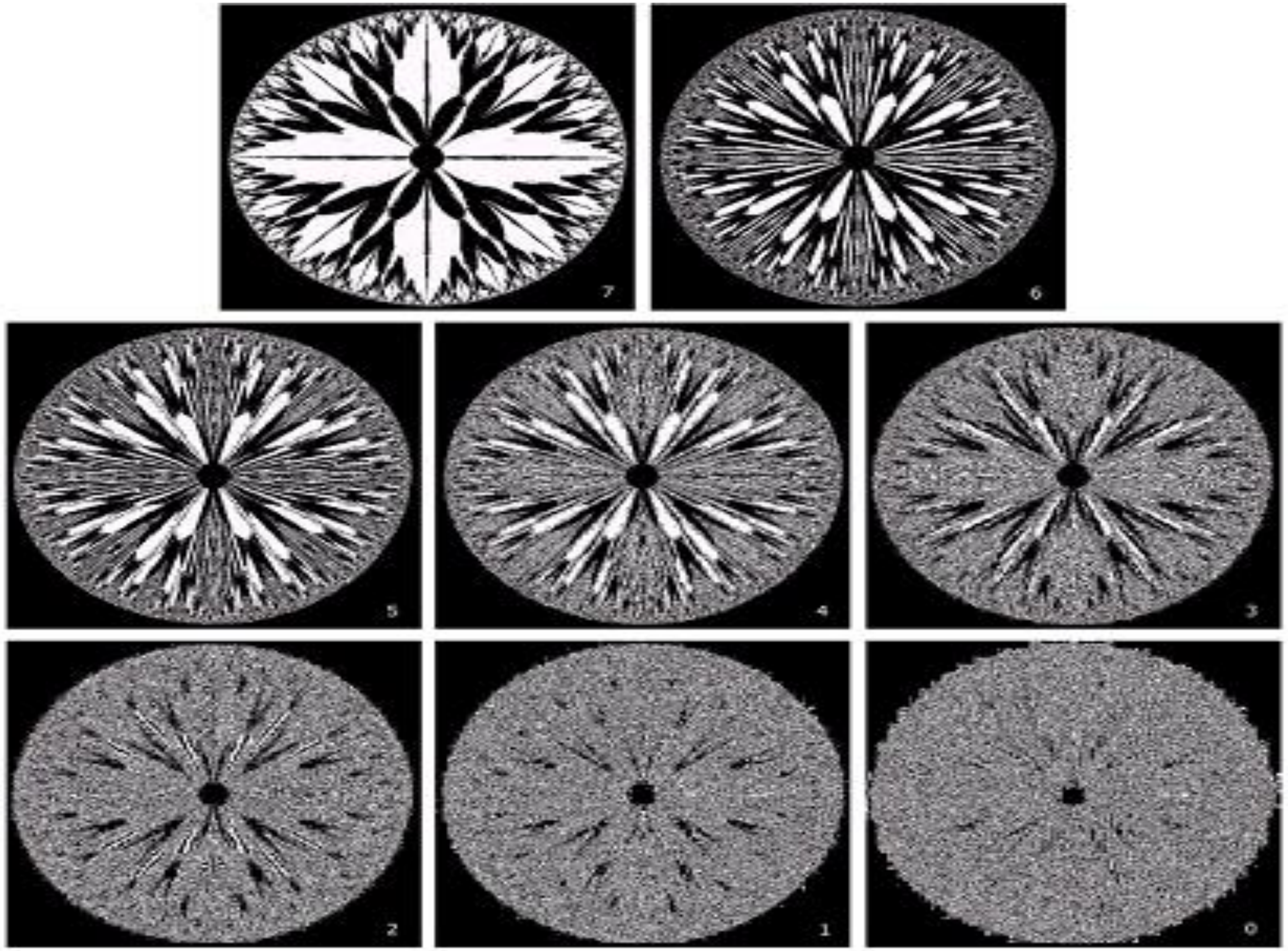


An 8-bit fractal image

The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.

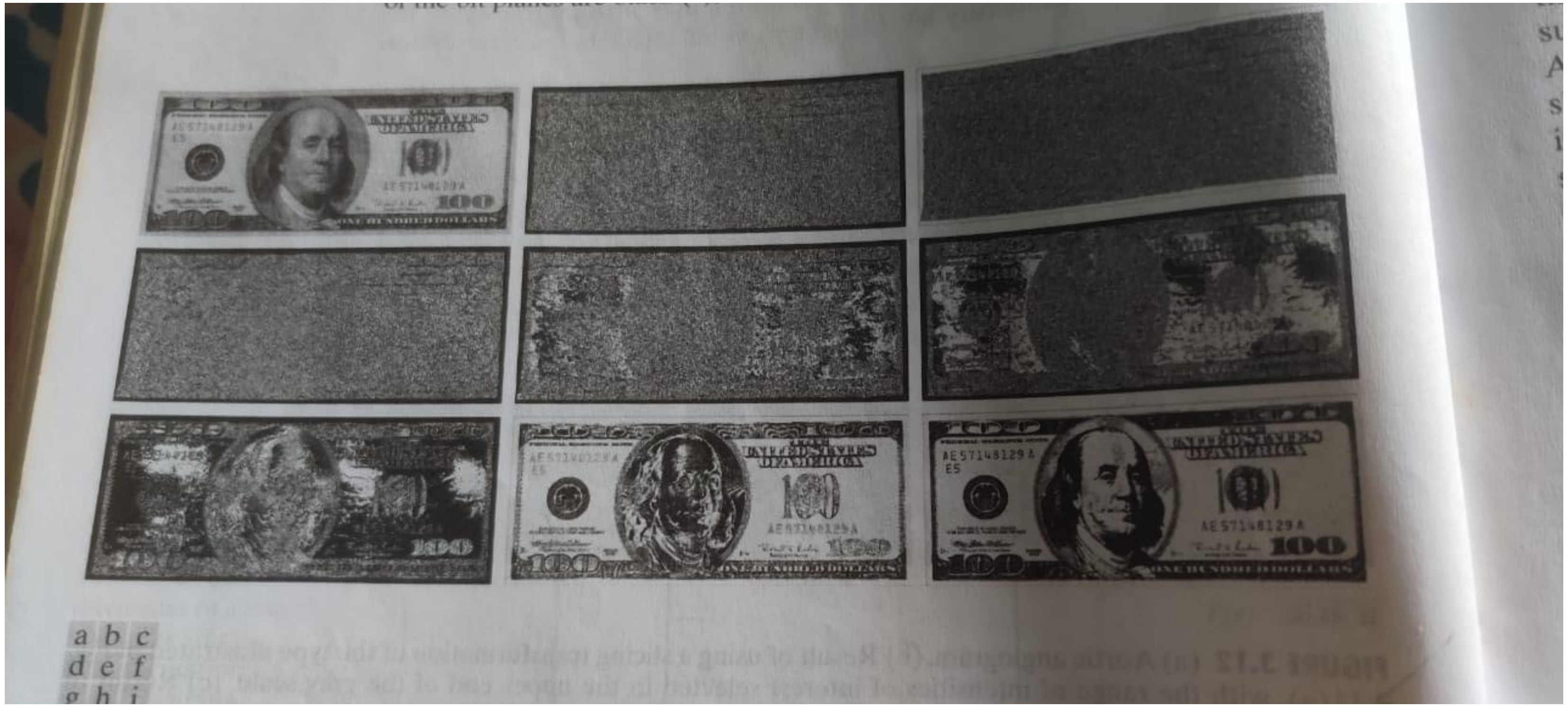
- Map all levels between 0 and 127 to 0
- Map all levels between 128 and 255 to 1

8 bit planes



Bit-plane 7	Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3
Bit-plane 2	Bit-plane 1	Bit-plane 0

8 bit planes



a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

Suggested Readings

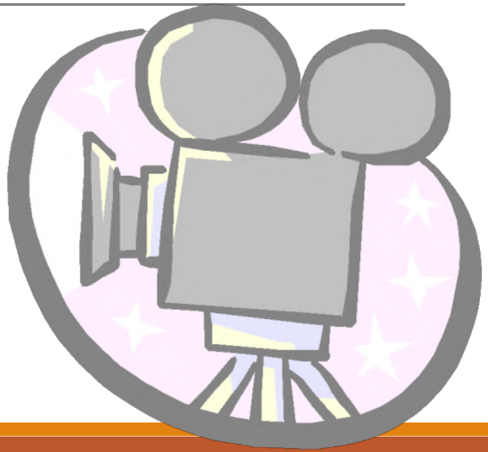
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Histogram Processing
    - Histogram Equalization
    - Histogram Matching

Histogram Processing

Histogram of a digital image with gray levels in the range [0,L-1] is a discrete function

$$h(r_k) = n_k$$

Where

- r_k : the kth gray level
- n_k : the number of pixels in the image having gray level r_k
- $h(r_k)$: histogram of a digital image with gray levels r_k

Normalized Histogram

dividing each of its component by the total number of pixels in the image, MN

$$p(r_k) = n_k / MN$$

Where, M, N are the no. of rows and columns of the image.

For k = 0,1,…,L-1

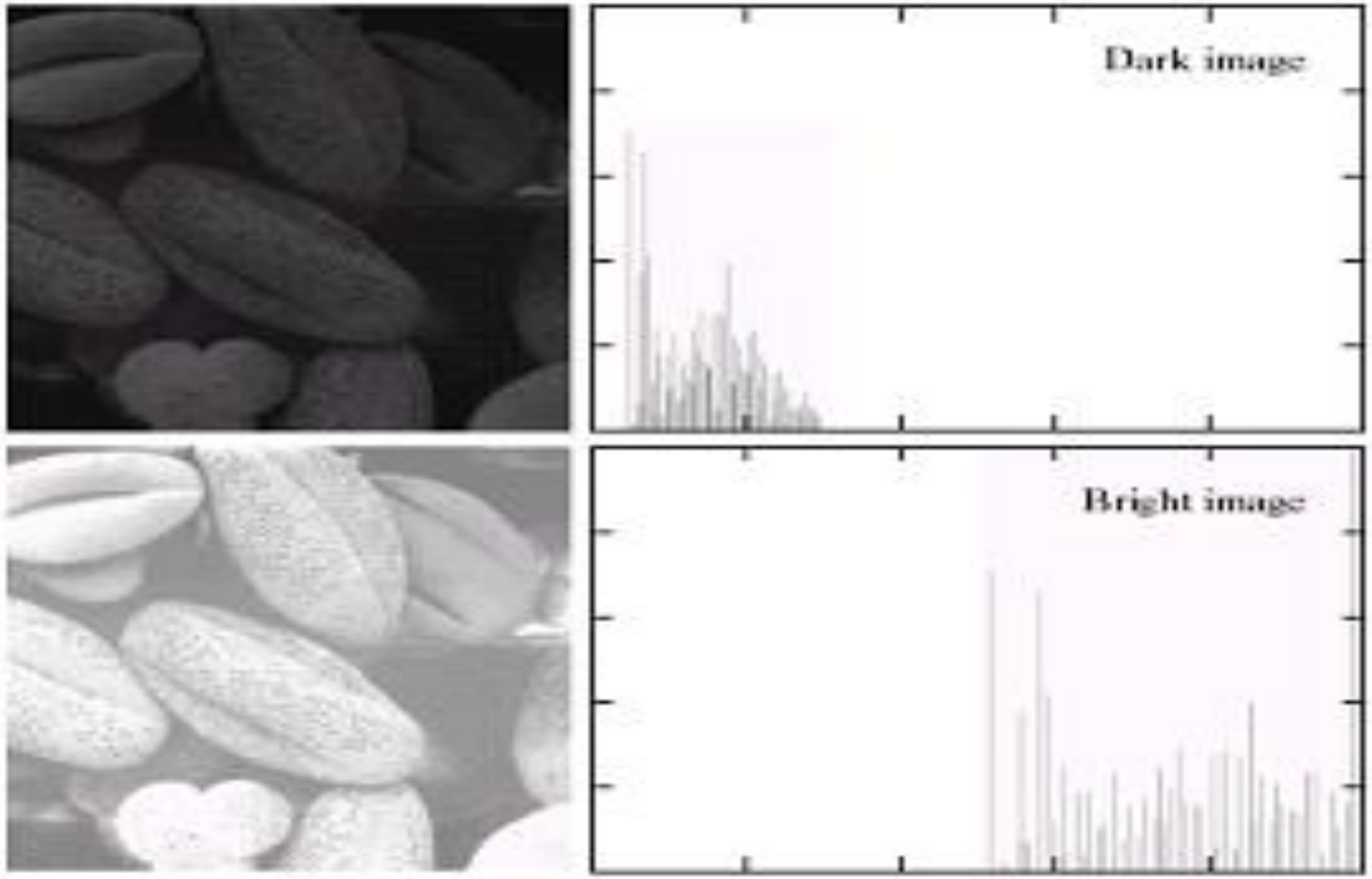
$p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k

The sum of all components of a normalized histogram is equal to 1

Histogram Processing

- Basic for numerous spatial domain processing techniques
- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation

Example



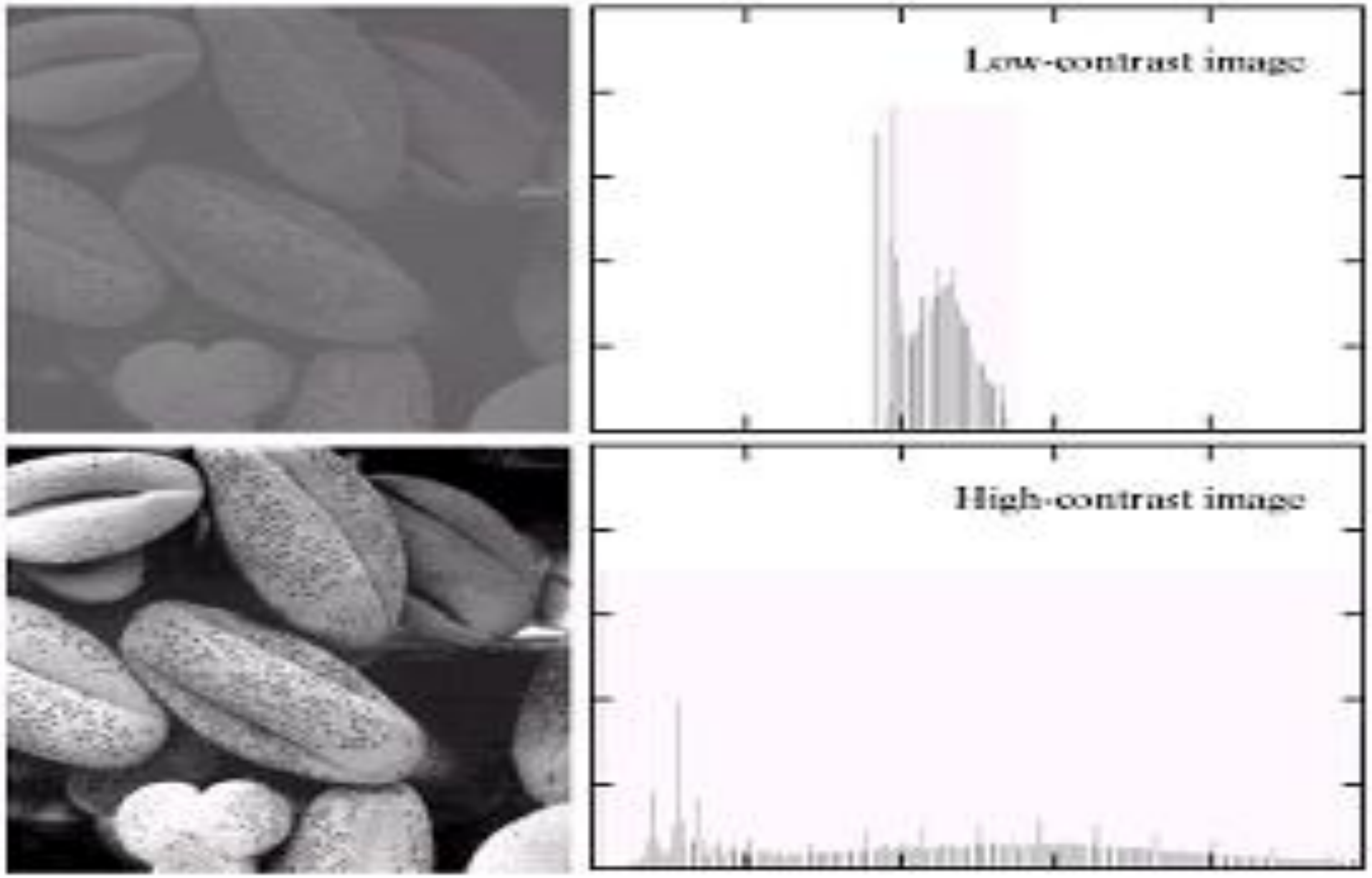
Dark image

Components of histogram are concentrated on the low side of the gray scale.

Bright image

Components of histogram are concentrated on the high side of the gray scale.

Example



Low-contrast image

histogram is narrow and centered toward the middle of the gray scale

High-contrast image

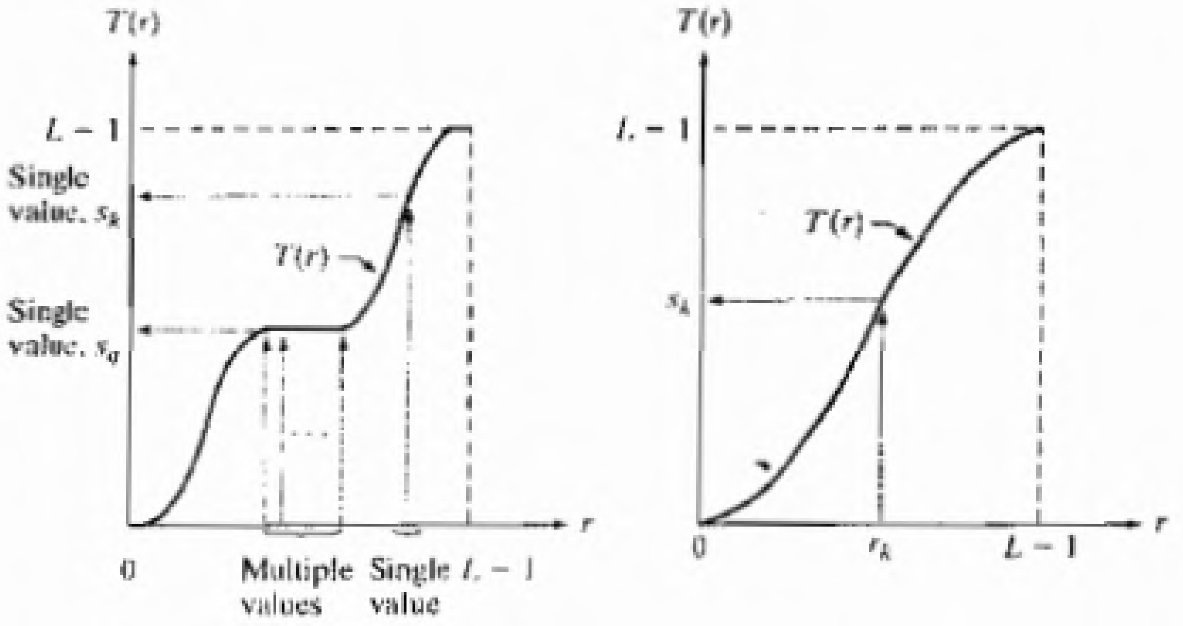
histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others

Histogram Equalization

As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range, the quality of the image will be improved.

We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally.

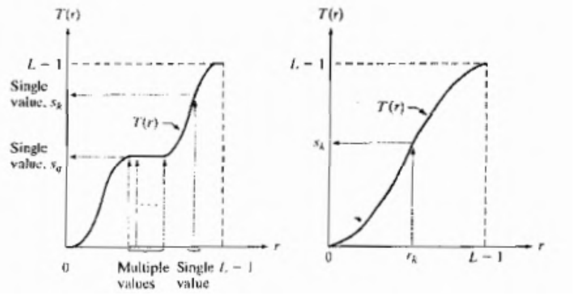
Histogram transformation



a b

FIGURE 3.17
(a) Monotonically increasing function, showing how multiple values can map to a single value.
(b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Histogram transformation



a b
FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Consider the transformation function of the form

$$s = T(r)$$

Where $0 \le r \le L-1$

T(r) satisfies

(a) T(r) is a monotonically increasing function in the interval $0 \le r \le L-1$.

(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

In some formulations to be discussed later, we use inverse

$r= T^{-1}(s)$ for $0 \le s \le L-1$

In which condition (a) is changed to (a')

(a') T(r) is strictly monotonically increasingly in the interval $0 \le r \le L- 1$

Histogram Equalization

The intensity levels in an image may be viewed as a random variables in the interval [0-L-1]. A fundamental descriptor of a random variable is its probability density function (PDF). Let $p_r(r)$ and $p_s(s)$ denote the PDFs of r and s respectively.

If $p_r(r)$ and T(r) are known anf T(r) is continuous and differentiable over the range of values of interest then the PDF of the transformed variable s can be obtained using

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \qquad \text{..................(1)}$$

Histogram Equalization

A transformation function of particular importance in image processing has the form

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

where w is a dummy variable of integration

Note: **T(r)** depends on $\mathbf{p_r(r)}$

Example

To find $p_s(s)$ corresponding to the transformation, we use eq (1) $p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$

$$\frac{ds}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right]$$

$$\frac{ds}{dr} = (L-1)\,p_r(r)$$

Substituting this result in eq 1

Example

Substituting this result in eq 1

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$p_s(s) = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right|$$

$$p_s(s) = \frac{1}{(L-1)}, 0 \le s \le L-1$$

Example

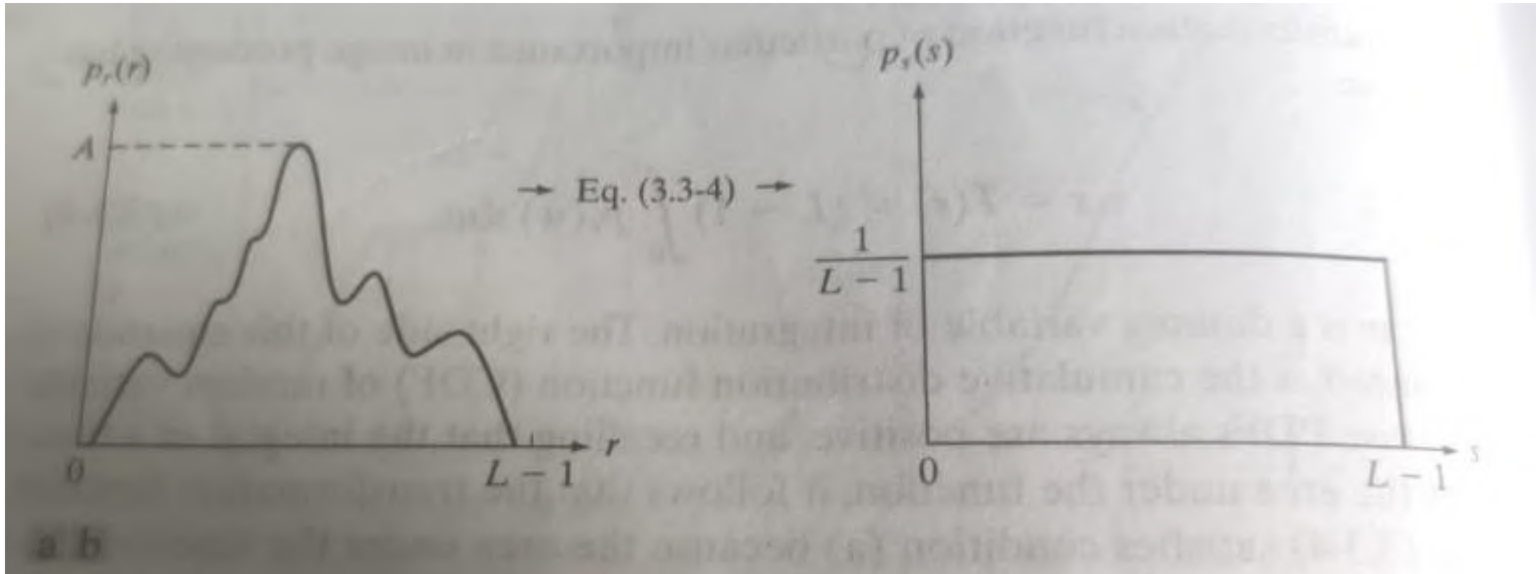


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

* Image is taken from **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

Histogram Equalization

The discrete form of the transformation is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_k), k = 0,1,2......L-1$$

$$s_k = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_k$$

Example

■ Before continuing, it will be helpful to work through a simple example. Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in Table 3.1, where the intensity levels are integers in the range $[0, L-1] = [0, 7]$.

The histogram of our hypothetical image is sketched in Fig. 3.19(a). Values of the histogram equalization transformation function are obtained using Eq. (3.3-8). For instance,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

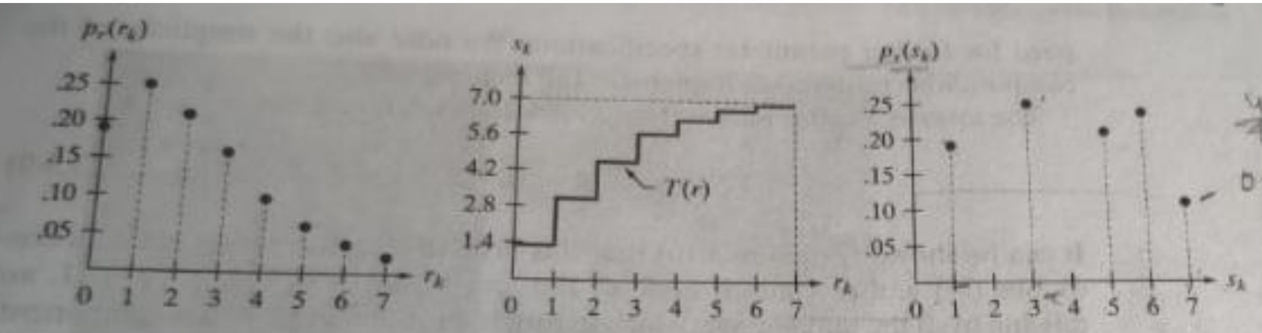
Similarly,

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Example



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

At this point, the s values still have fractions because they were generated by summing probability values, so we round them to the nearest integer:

$$s_0 = 1.33 \rightarrow 1 \qquad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \qquad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \qquad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \qquad s_7 = 7.00 \rightarrow 7$$

6 - Pixel

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

245+122+81=448 pixels with a value 7
In the histogram equalized image

Histogram Equalization

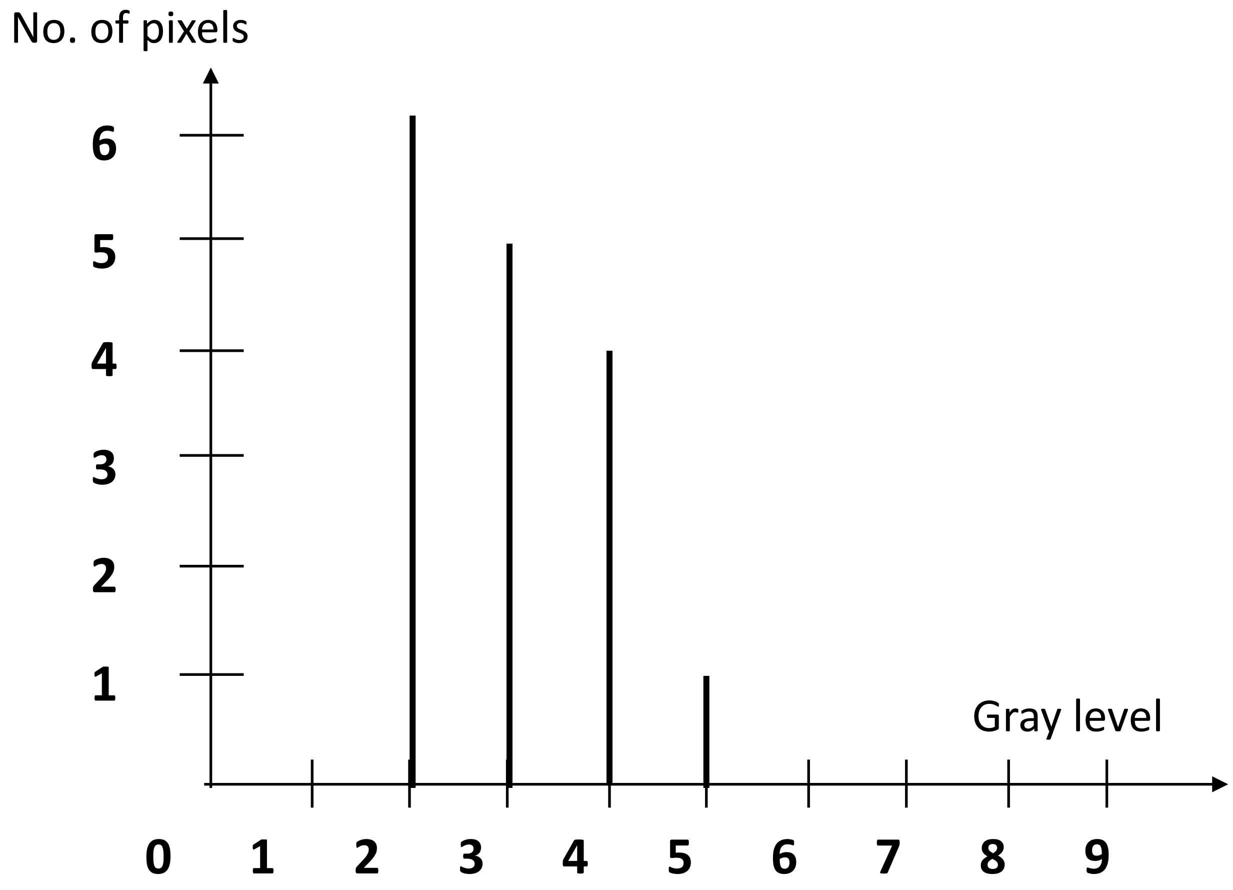
Thus, an output image is obtained by mapping each pixel with level $\boldsymbol{r_k}$ in the input image into a corresponding pixel with level $\boldsymbol{s_k}$ in the output image.

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{MN}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Example

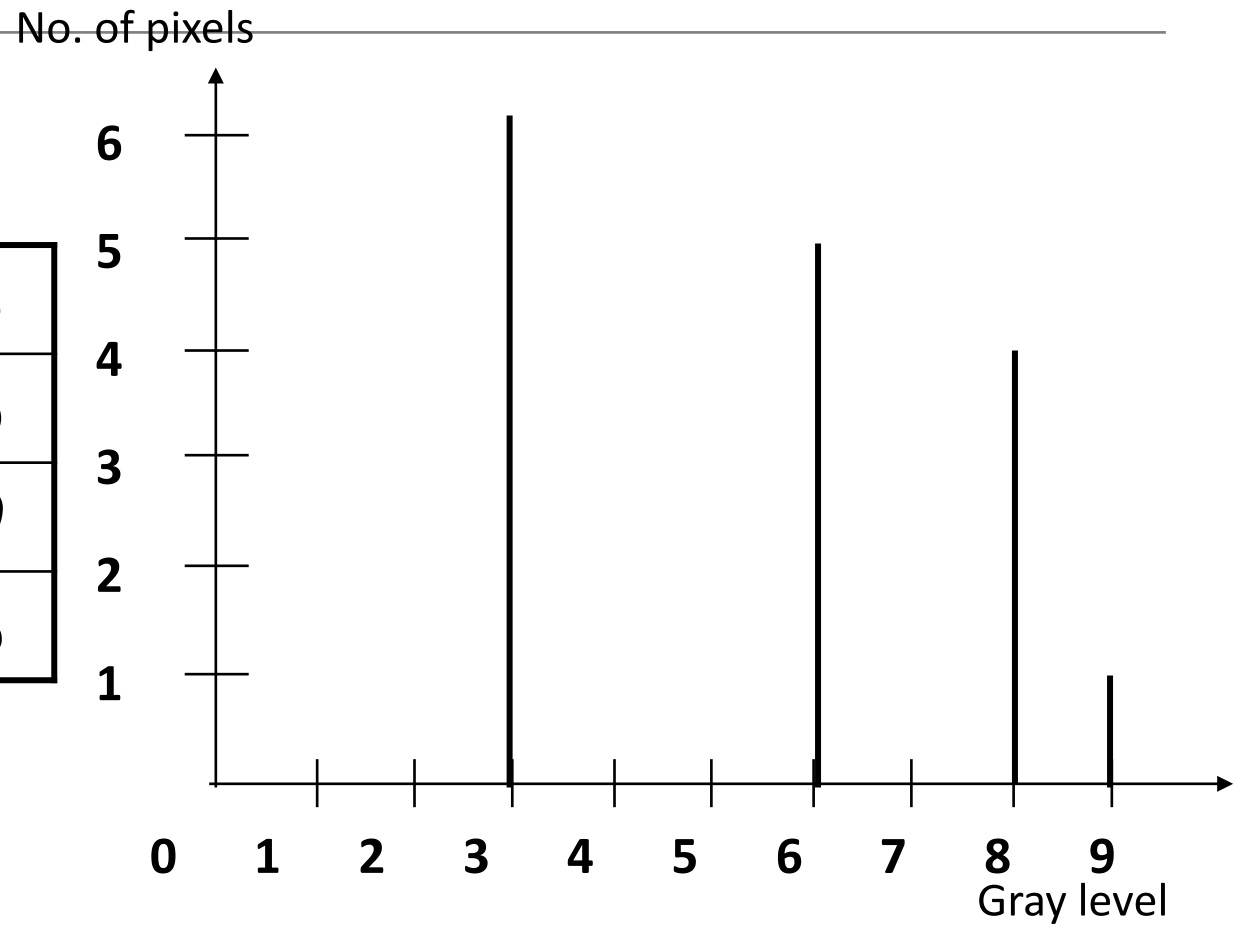
2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Input image

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

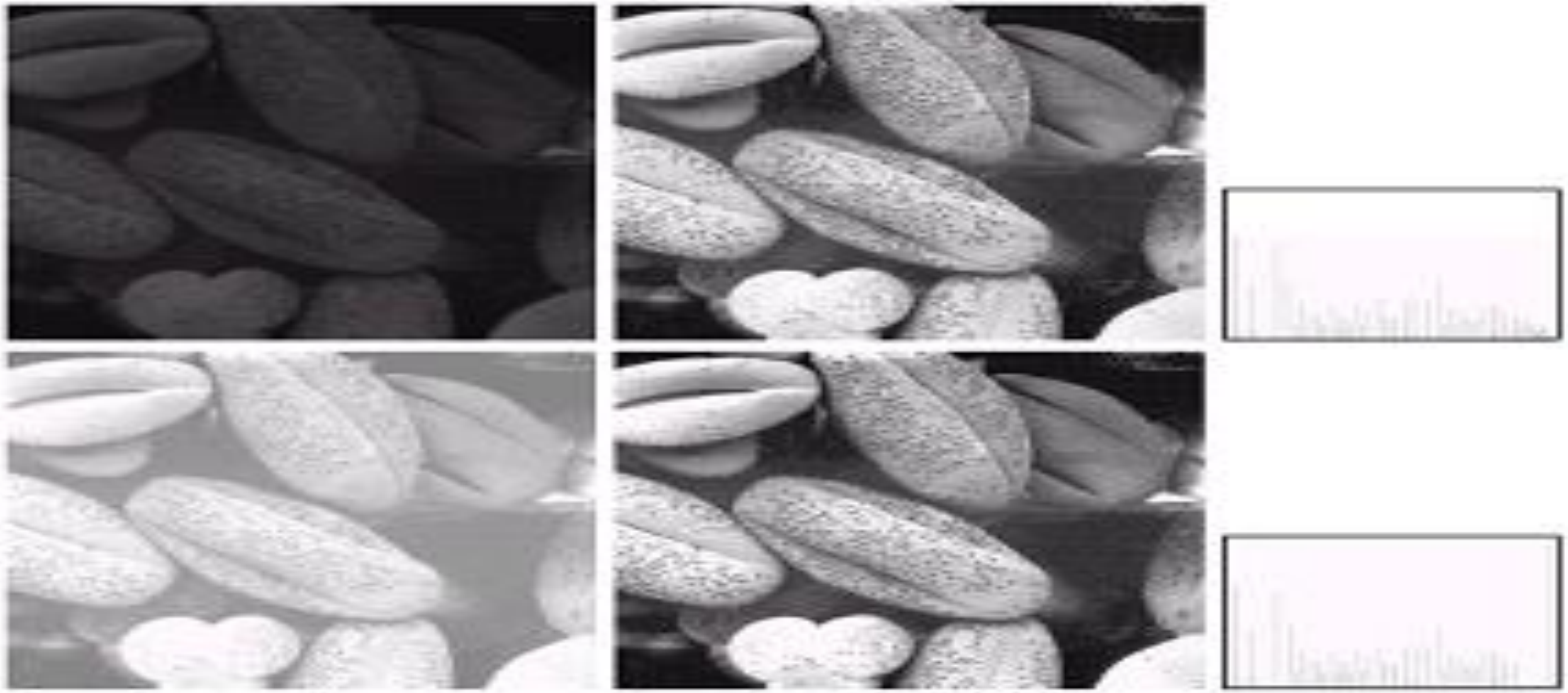
Gray scale = [0,9]



Histogram equalization

Example

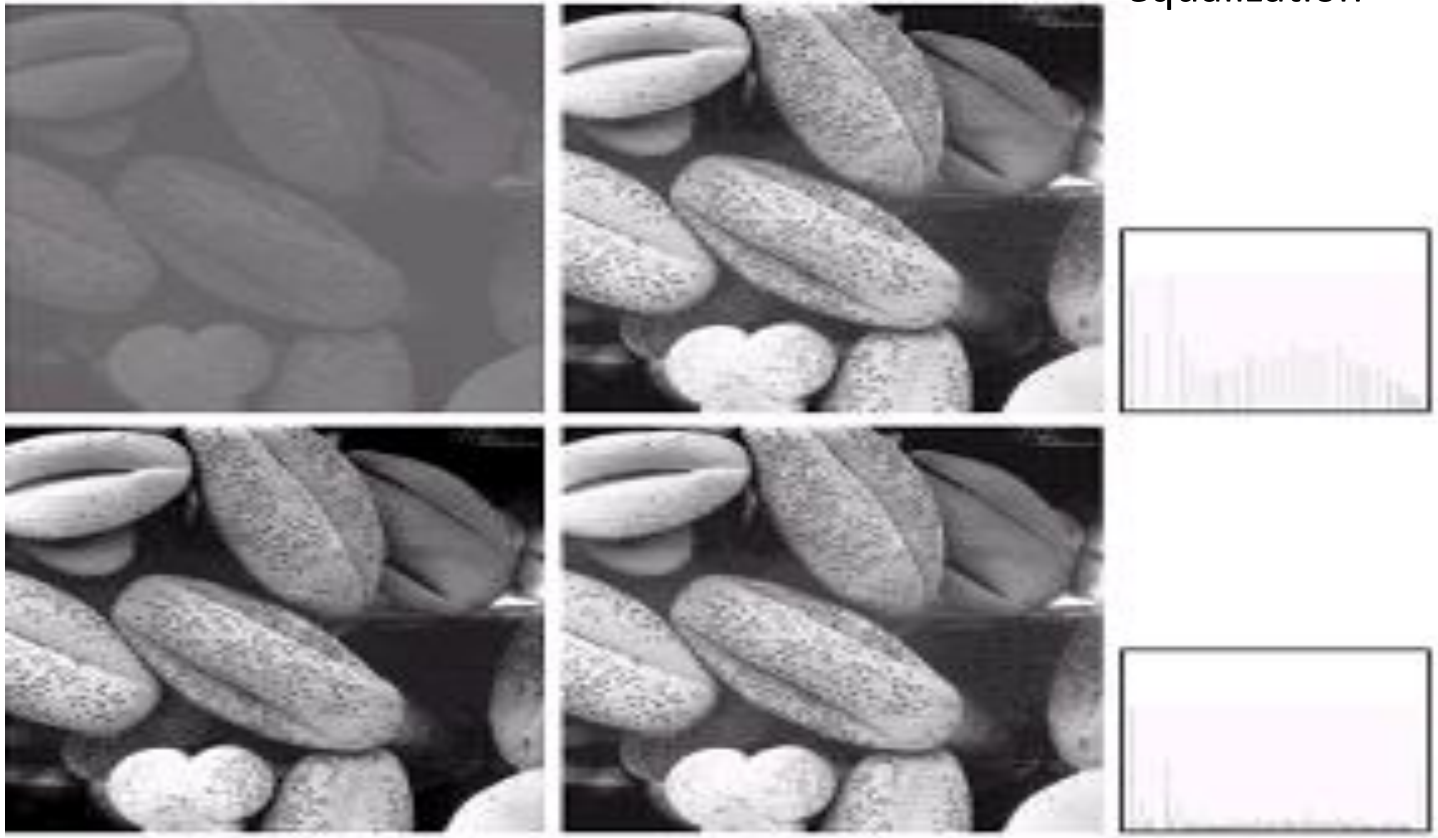
before after Histogram equalization



* Image is taken from **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

Example

before after Histogram equalization



The quality is not improved much because the original image already has a broaden gray-level scale

* Image is taken from **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

Histogram Matching (Specification)

Histogram equalization has a disadvantage which is that it can generate only one type of output image.

With Histogram Specification, we can specify the shape of the histogram that we wish the output image to have.

It doesn’t have to be a uniform histogram

Consider the continuous domain

Let $p_r(r)$ denote continuous probability density function of gray-level of input image, r

Let $p_z(z)$ denote desired (specified) continuous probability density function of gray-level of output image, z

Let s be a random variable with the property

$$s = T(r) = \int_0^r p_r(w)\,dw$$

Histogram equalization

Where w is a dummy variable of integration

Next, we define a random variable z with the property

$$G(z) = \int_0^z p_z(t)\,dt = s$$

Histogram equalization

Where t is a dummy variable of integration

thus

$$s = T(r) = G(z)$$

Therefore, z must satisfy the condition

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

Assume G^{-1} exists and satisfies the condition (a') and (b)

We can map an input gray level r to output gray level z

Procedure Conclusion

1. Obtain the transformation function T(r) by calculating the histogram equalization of the input image

$$s = T(r) = \int_0^r p_r(w)\,dw$$

2. Obtain the transformation function G(z) by calculating histogram equalization of the desired density function

$$G(z) = \int_0^z p_z(t)\,dt = s$$

Procedure Conclusion

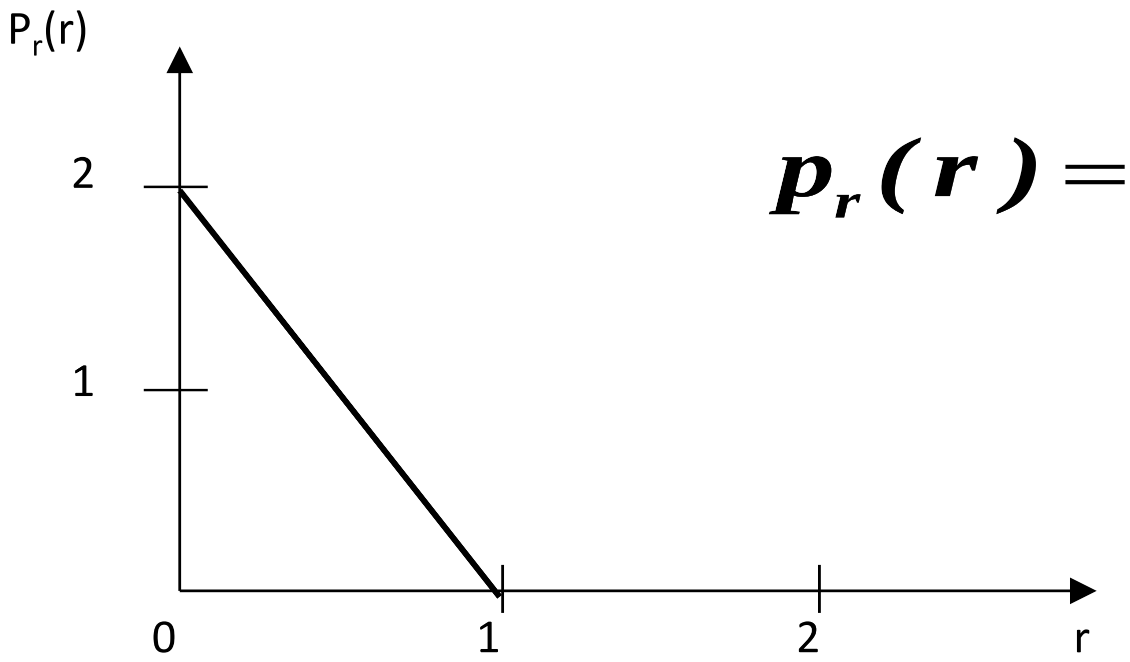
3. Obtain the inversed transformation function G^{-1}

$$z = G^{-1}(s) = G^{-1}[T(r)]$$

4. Obtain the output image by applying the processed gray-level from the inversed transformation function to all the pixels in the input image

Example

Assume an image has a gray level probability density function $p_r(r)$ as shown.

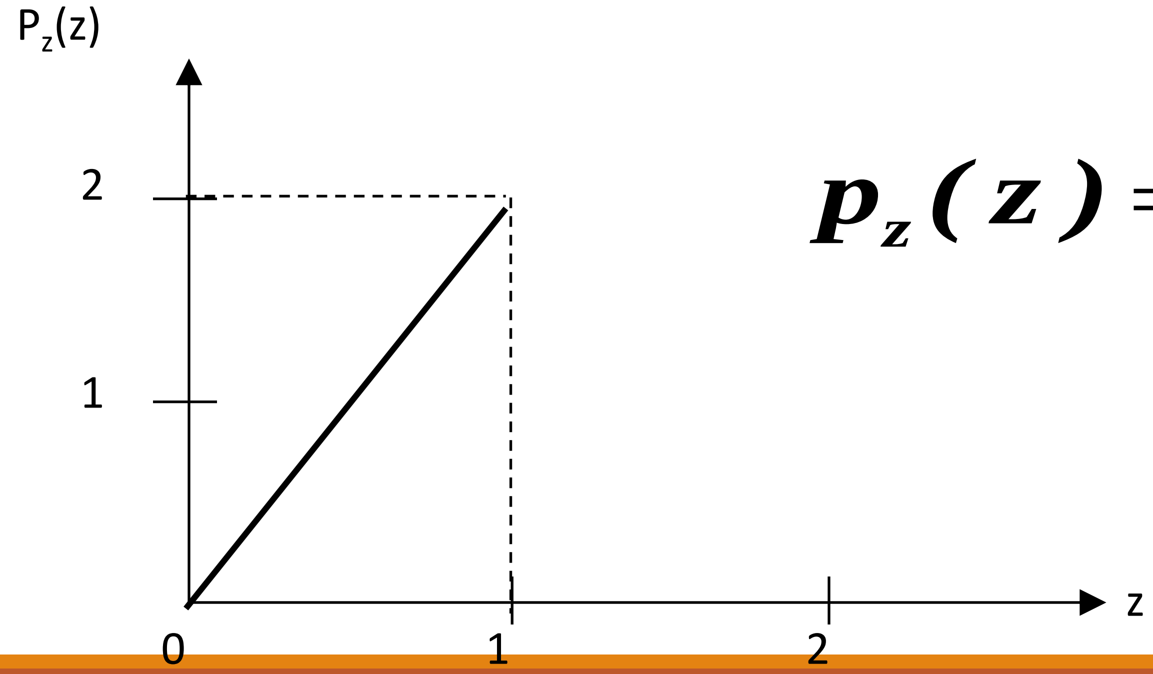


$$p_r(r) = \begin{cases} -2r + 2 & ;0 \le r \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

$$\int_0^r p_r(w)dw = 1$$

Example

We would like to apply the histogram specification with the desired probability density function $p_z(z)$ as shown.

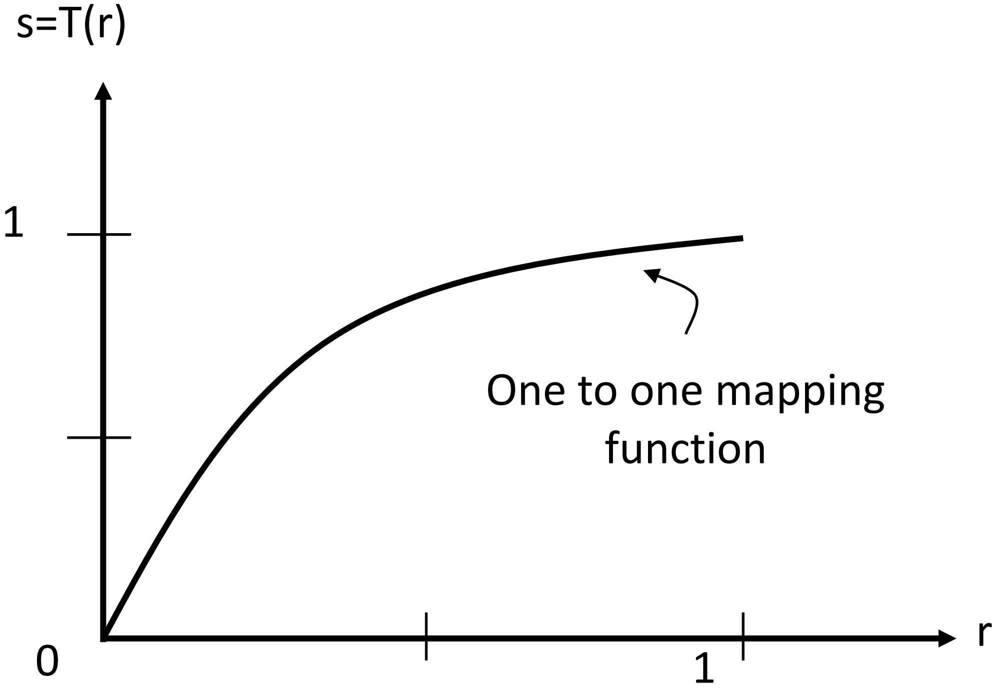


$$p_z(z) = \begin{cases} 2z & ;0 \le z \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

$$\int_0^z p_z(w)dw = 1$$

Step 1:

Obtain the transformation function T(r)



$$s = T(r) = \int_0^r p_r(w)dw$$

$$= \int_0^r (-2w + 2)dw$$

$$= -w^2 + 2w\Big|_0^r$$

$$= -r^2 + 2r$$

Step 2:

Obtain the transformation function G(z)

$$G(z) = \int_0^z (2w)dw \qquad = z^2\Big|_0^z \qquad = z^2$$

Step 3:

Obtain the inversed transformation function G^{-1}

$$G(z) = T(r)$$

$$z^2 = -r^2 + 2r$$

$$z = \sqrt{2r - r^2}$$

We can guarantee that $0 \le z \le 1$ when $0 \le r \le 1$

Discrete formulation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{MN} \qquad k = 0,1,2,\ldots, L-1$$

$$G(z_k) = (L-1)\sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0,1,2,\ldots, L-1$$

$$\boldsymbol{z_k} = \boldsymbol{G}^{-1}\left[\boldsymbol{T(r_k)}\right]$$

$$= \boldsymbol{G}^{-1}\left[\boldsymbol{s_k}\right] \qquad \boldsymbol{k} = 0,1,2,\ldots, \boldsymbol{L}-1$$

Example (Histogram specification)

Step-1 Histogram equalization of original image

Before continuing, it will be helpful to work through a simple example. Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in Table 3.1, where the intensity levels are integers in the range $[0, L-1] = [0, 7]$.

The histogram of our hypothetical image is sketched in Fig. 3.19(a). Values of the histogram equalization transformation function are obtained using Eq. (3.3-8). For instance,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

Similarly,

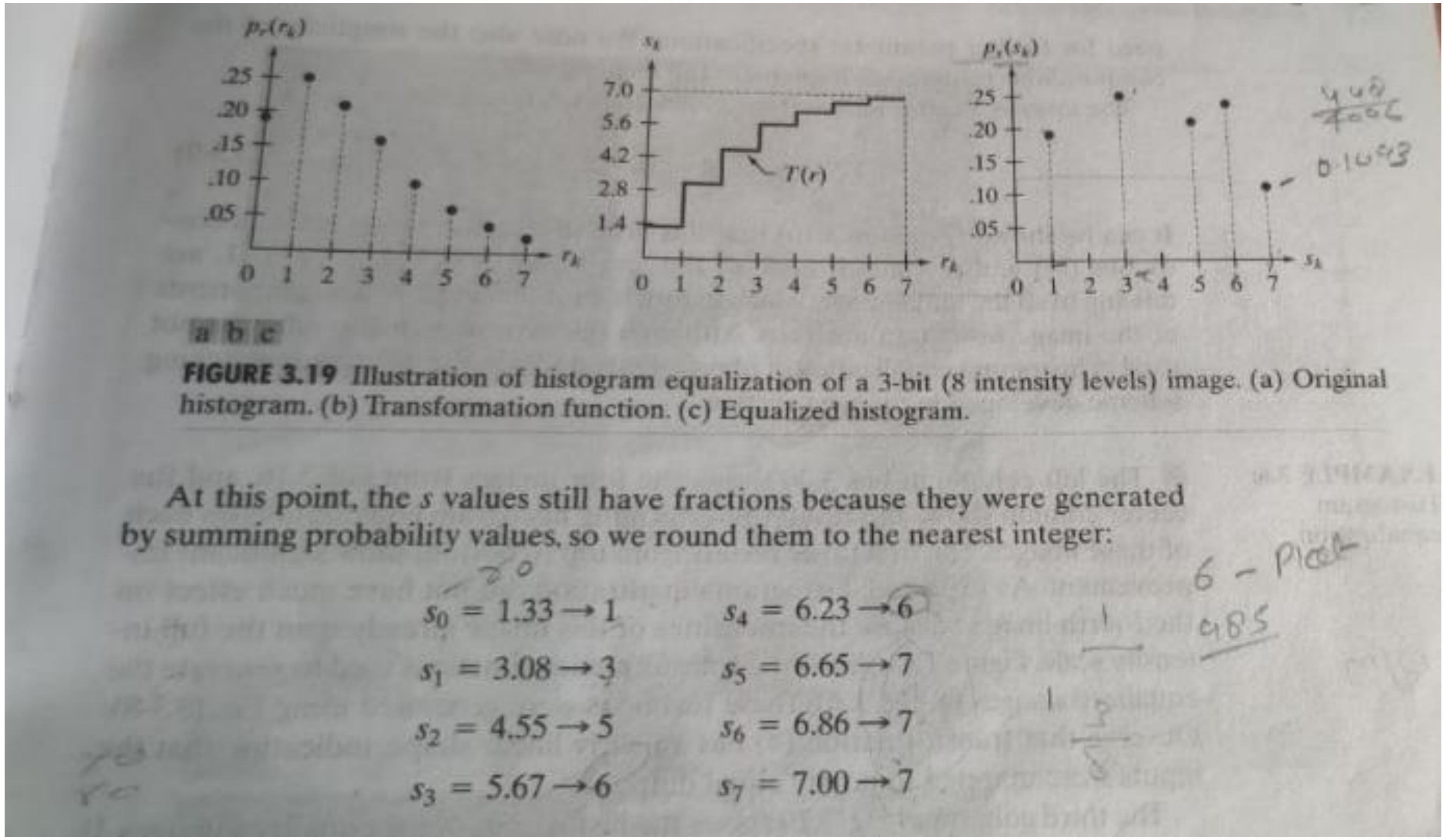
$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

* Image is taken from **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

Example



a b c

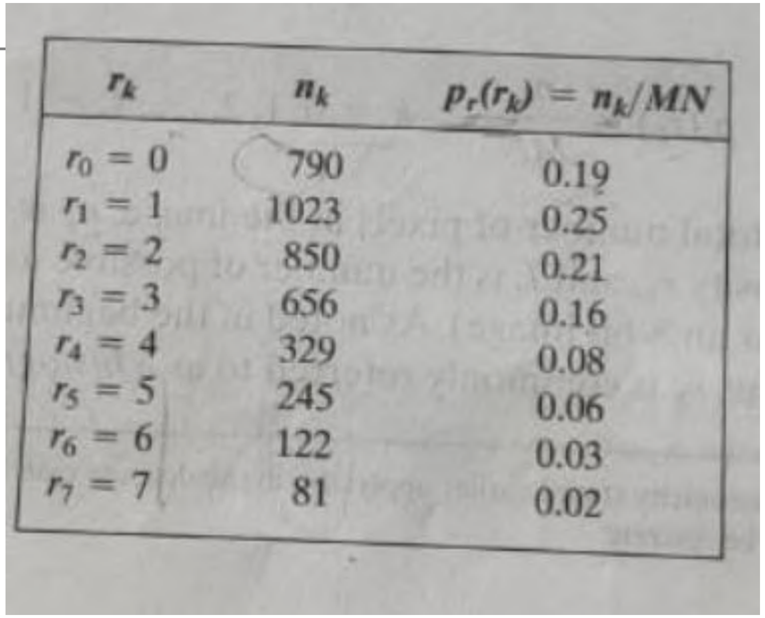
FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

At this point, the s values still have fractions because they were generated by summing probability values, so we round them to the nearest integer:

$$s_0 = 1.33 \rightarrow 1 \qquad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \qquad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \qquad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \qquad s_7 = 7.00 \rightarrow 7$$

6 - Pixel

0.85



r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

245+122+81=448 pixels with a value 7
In the histogram equalized image

Histogram Specification

Step-2 Compute all values of the transformation Function G

In the next step, we compute all the values of the transformation function, G, using Eq. (3.3-14):

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00$$

Similarly,

$$G(z_1) = 7\sum_{j=0}^{1} p_z(z_j) = 7\left[p(z_0) + p(z_1)\right] = 0.00$$

and

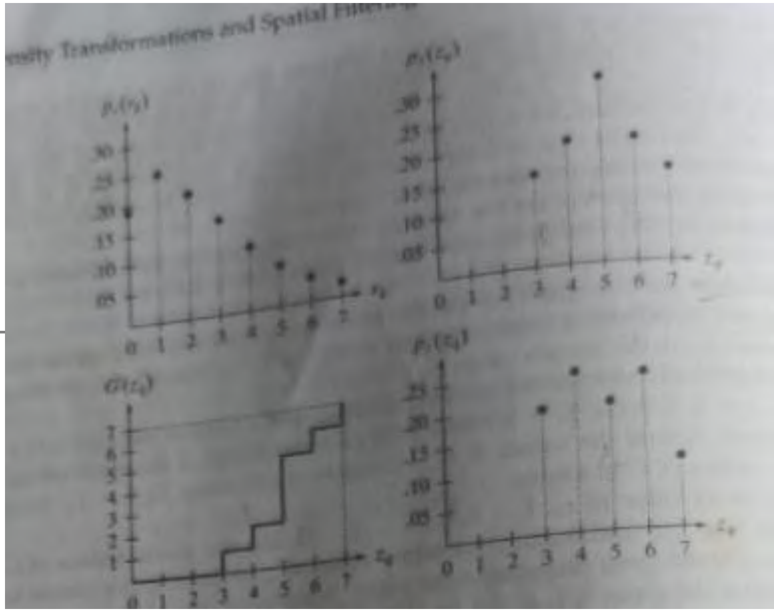
$$G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$

$$G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Step-3 Find the smallest value of z_q so that the value $F(z_q)$ is the closest to s_k.

Do this for each s_k to create the required mapping from s to z.



z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

$G(z_0) = 0.00 \rightarrow 0$ $\quad$ $G(z_4) = 2.45 \rightarrow 2$

$G(z_1) = 0.00 \rightarrow 0$ $\quad$ $G(z_5) = 4.55 \rightarrow 5$

$G(z_2) = 0.00 \rightarrow 0$ $\quad$ $G(z_6) = 5.95 \rightarrow 6$

$G(z_3) = 1.05 \rightarrow 1$ $\quad$ $G(z_7) = 7.00 \rightarrow 7$

Step-4 Finally, use the mapping shown in table to

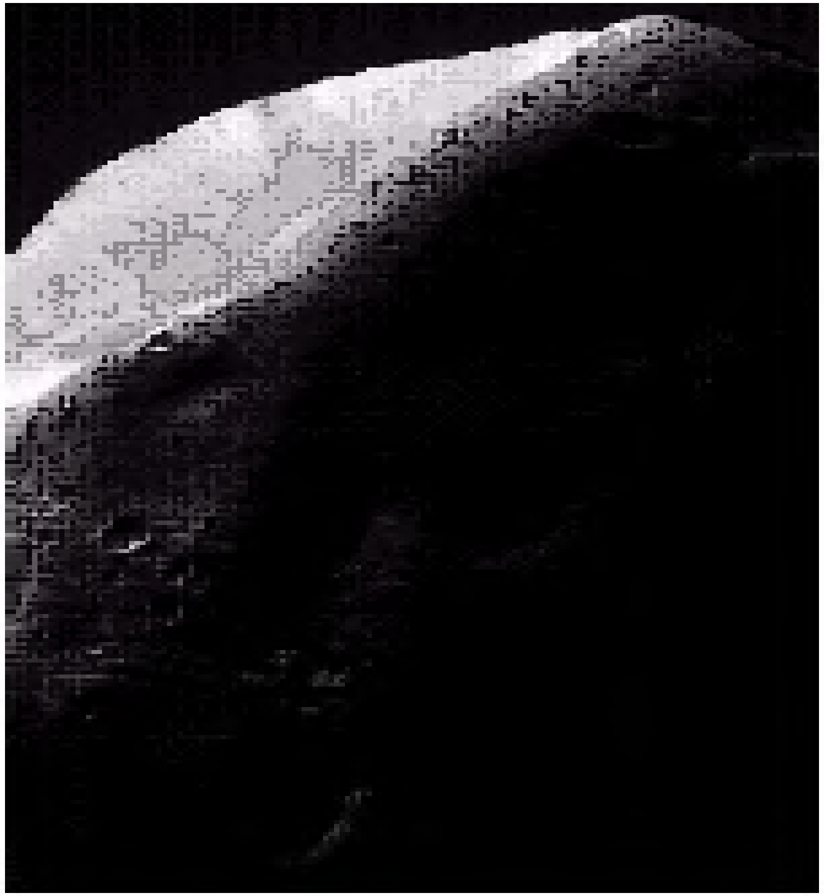
Map every pixel in the histogram equalized image

into a Corresponding pixel in the newly created

histogram Specified image.

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

Example



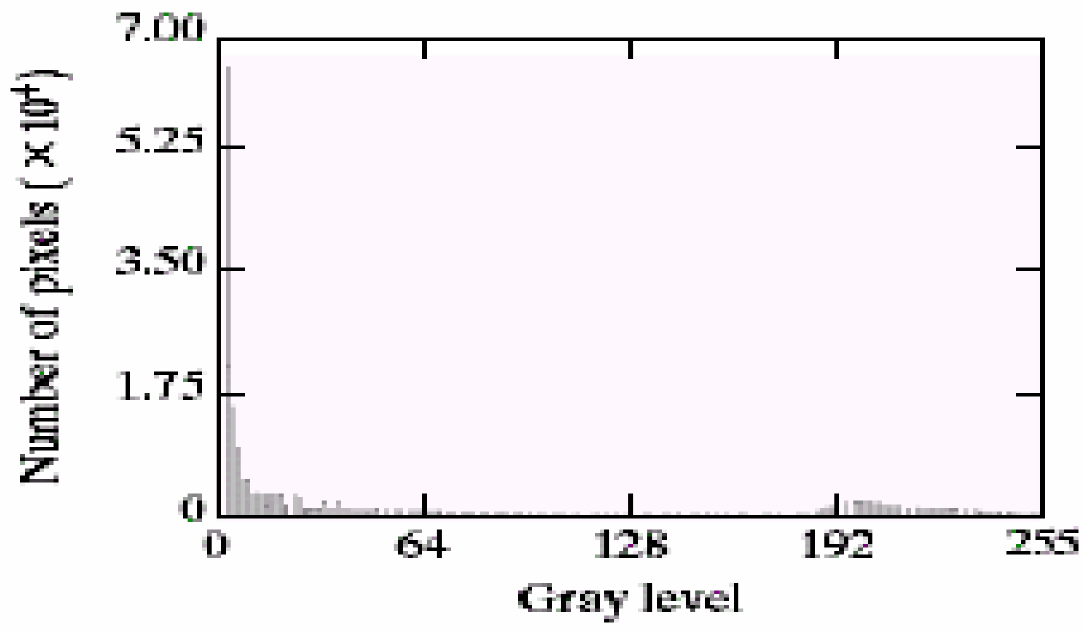
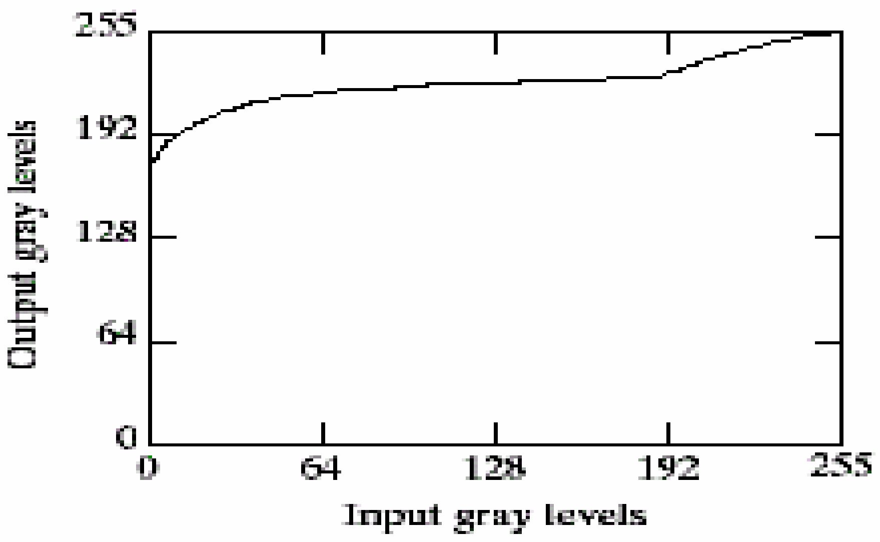


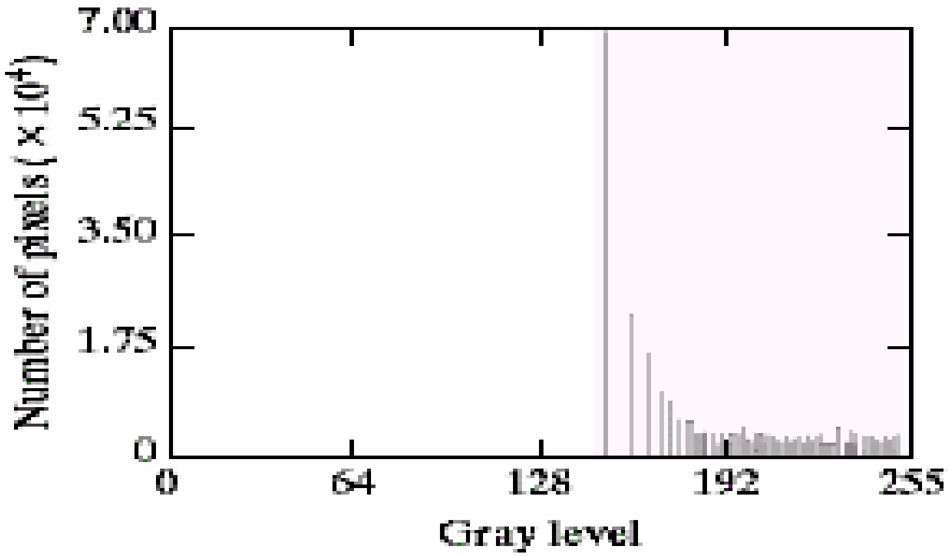
Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

Image of Mars moon

Image Equalization



Transformation function for histogram equalization

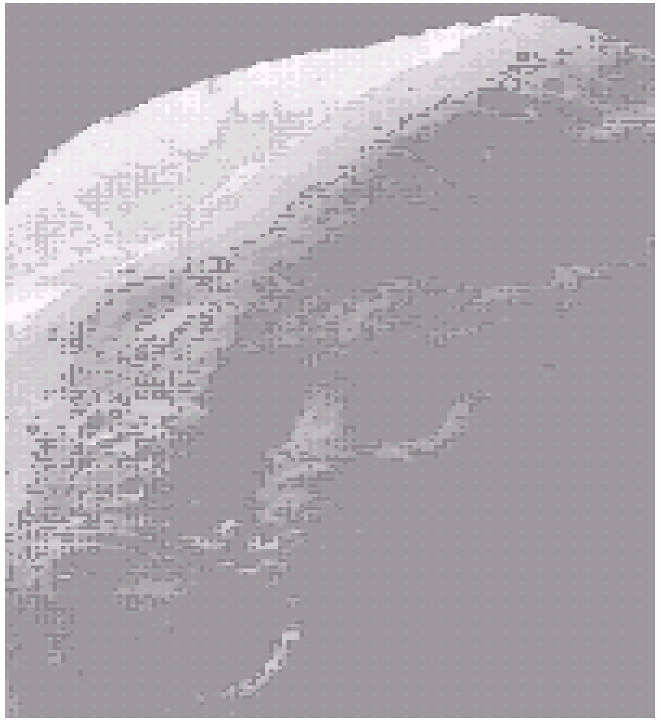


Histogram of the result image

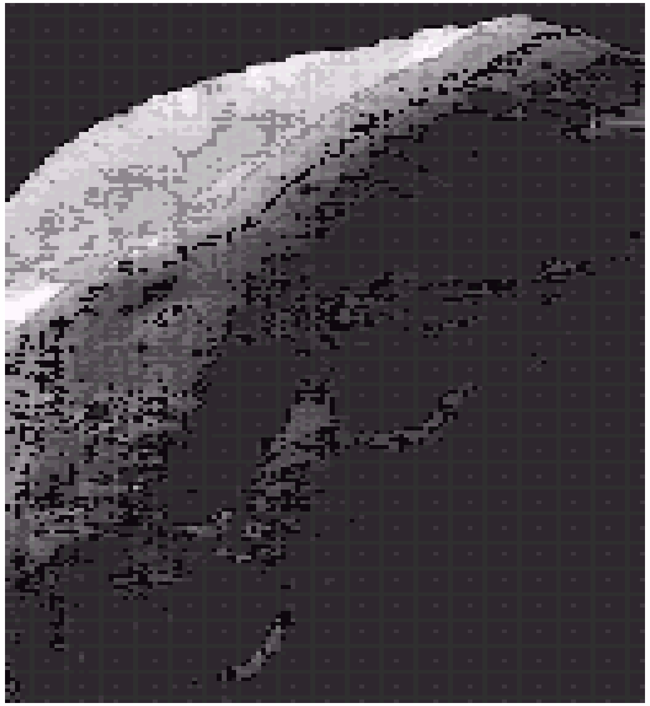
Result image after histogram equalization

The histogram equalization doesn't make the result image look better than the original image. Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image. As a consequence, the output image is light and has a washed-out appearance.

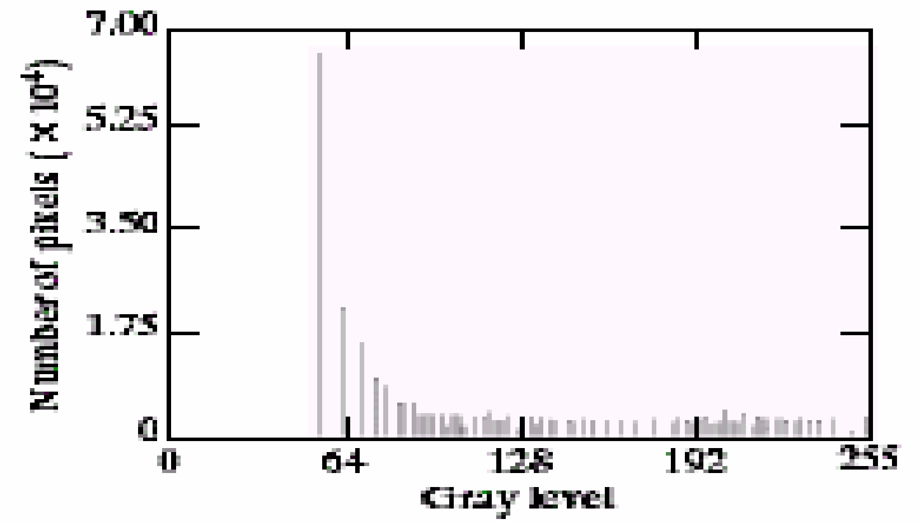
Result image and its histogram



Original image



After applied the histogram equalization



The output image's histogram

Notice that the output histogram's low end has shifted right toward the lighter region of the gray scale as desired.

Note

Histogram specification is a trial-and-error process

There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Note

Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.

Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

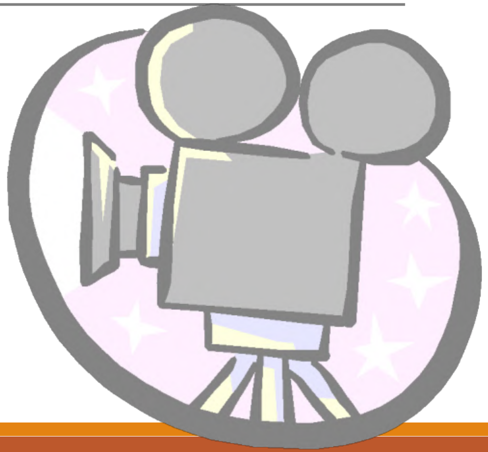
Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**
- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Histogram Processing
  - Local Histogram Processing

- Arithmetic/Logic Operations

Discrete formulation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

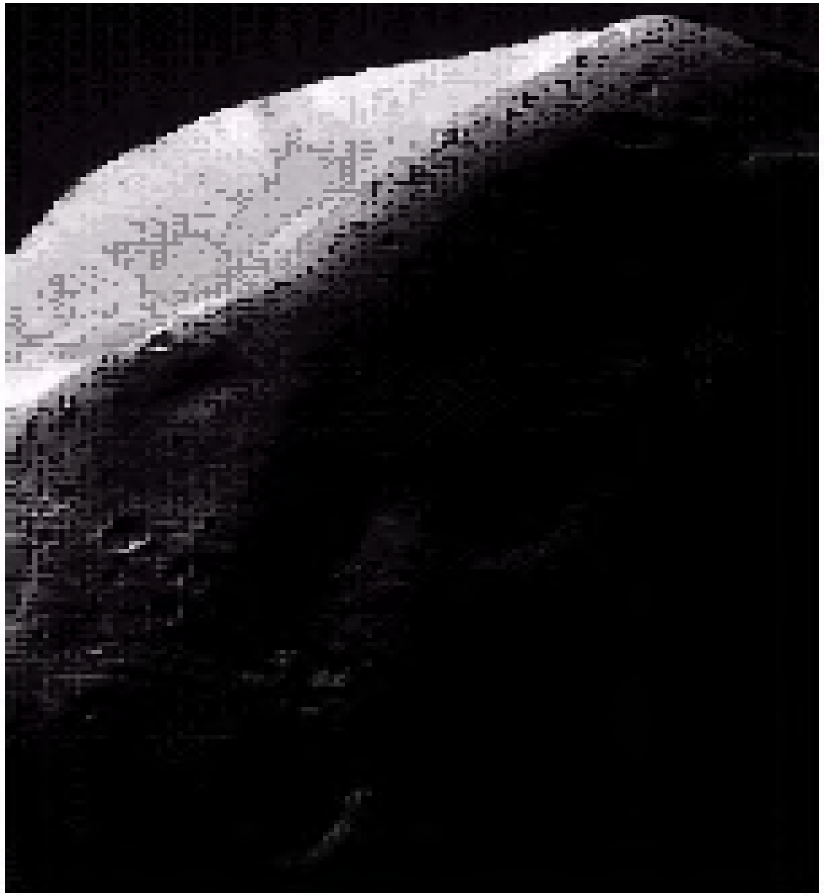
$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{MN} \qquad k = 0,1,2,\ldots, L-1$$

$$G(z_k) = (L-1)\sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0,1,2,\ldots, L-1$$

$$\boldsymbol{z_k} = \boldsymbol{G}^{-1}\left[\boldsymbol{T(r_k)}\right]$$

$$= \boldsymbol{G}^{-1}\left[\boldsymbol{s_k}\right] \qquad \boldsymbol{k} = 0,1,2,\ldots, \boldsymbol{L}-1$$

Example



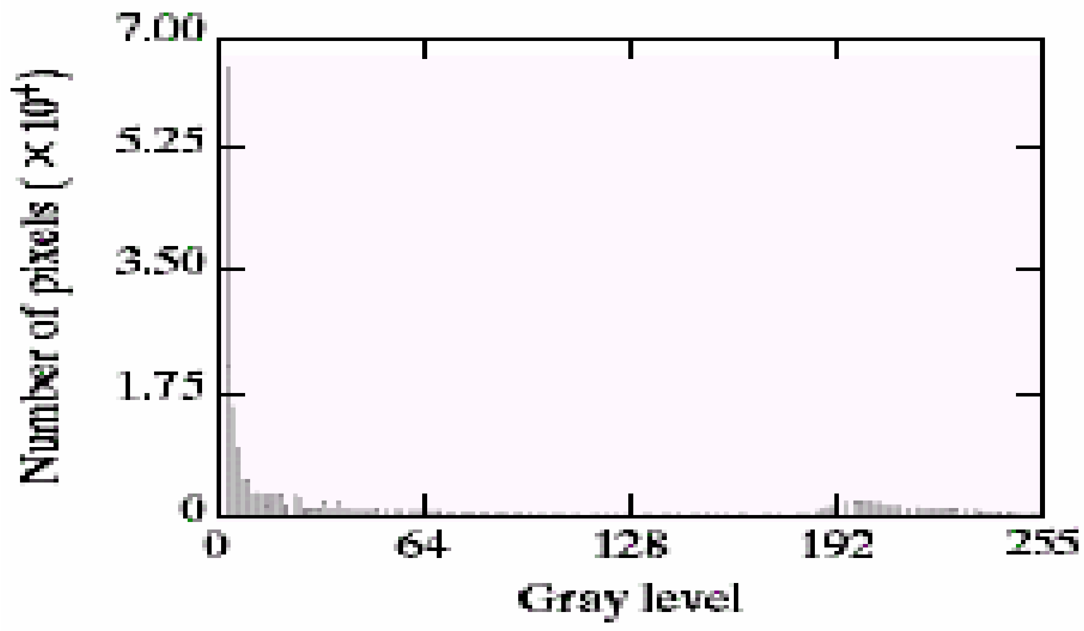
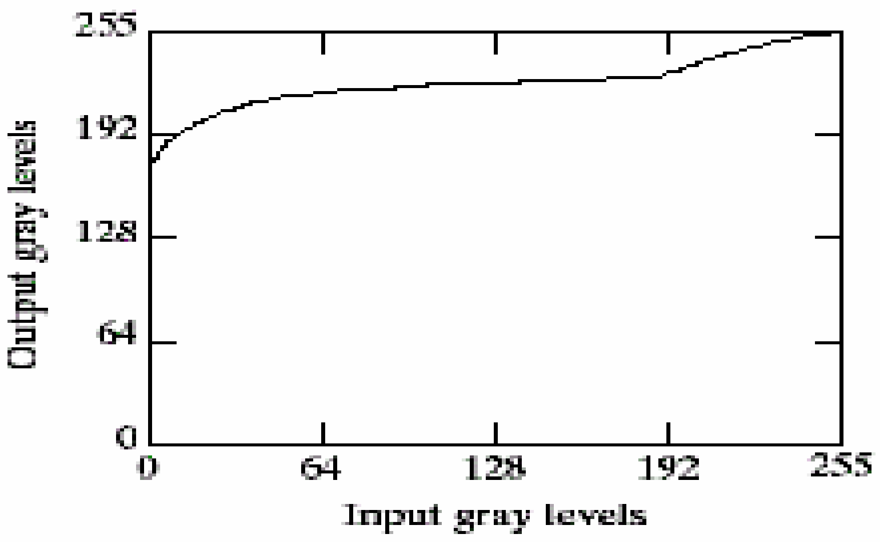


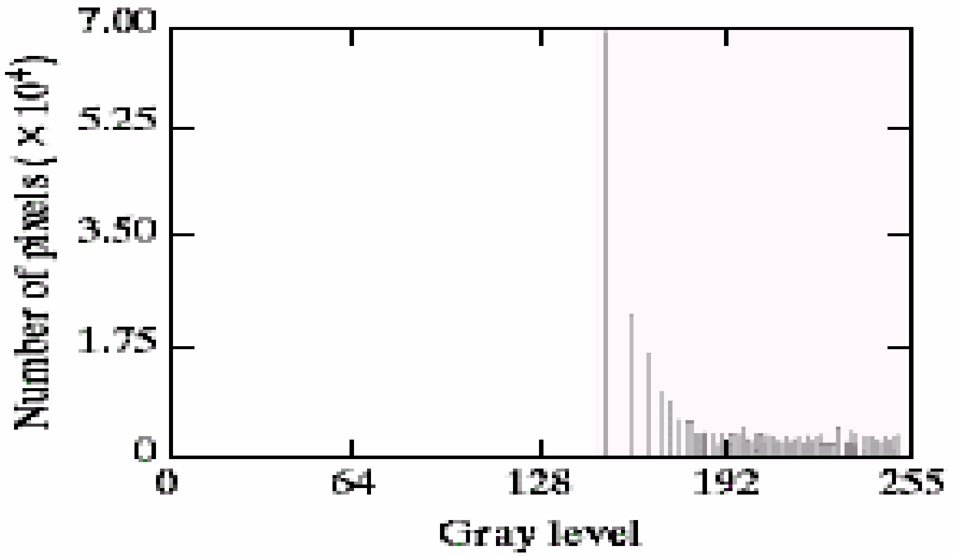
Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

Image of Mars moon

Image Equalization



Transformation function for histogram equalization



Histogram of the result image

Result image after histogram equalization

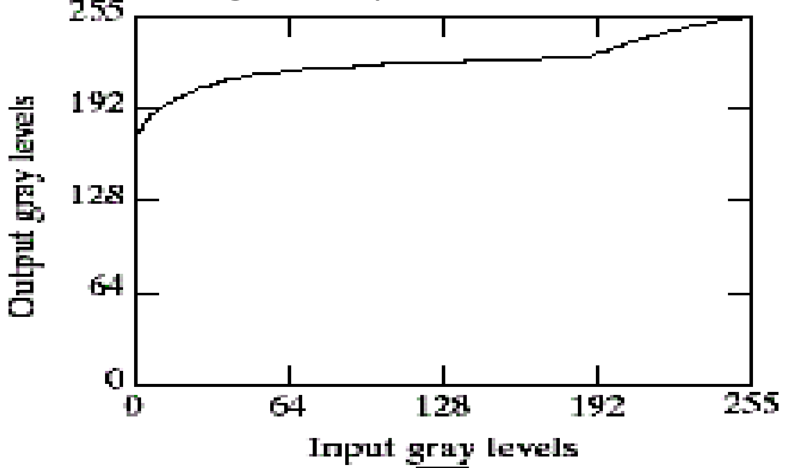
The histogram equalization doesn't make the result image look better than the original image. Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image. As a consequence, the output image is light and has a washed-out appearance.

Problem with the transformation function and solution

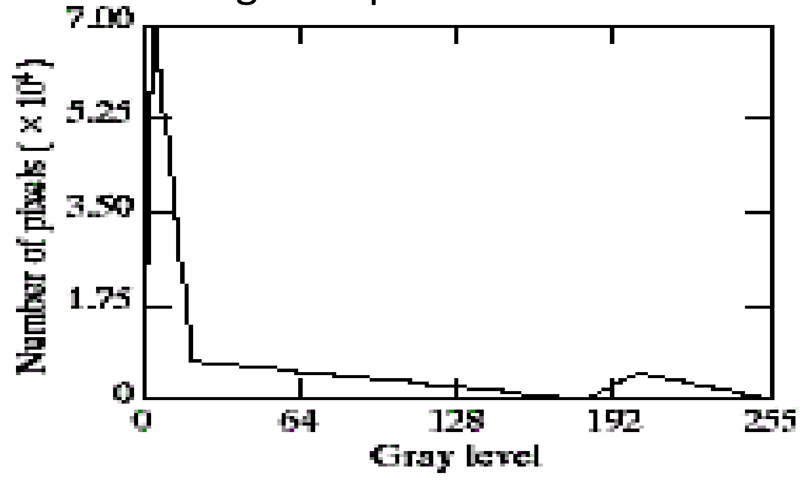
Since the problem with the transformation function of the histogram equalization was caused by a large concentration of pixels in the original image with levels near 0

a reasonable approach is to modify the histogram of that image so that it does not have this property

Histogram Equalization



Histogram Specification



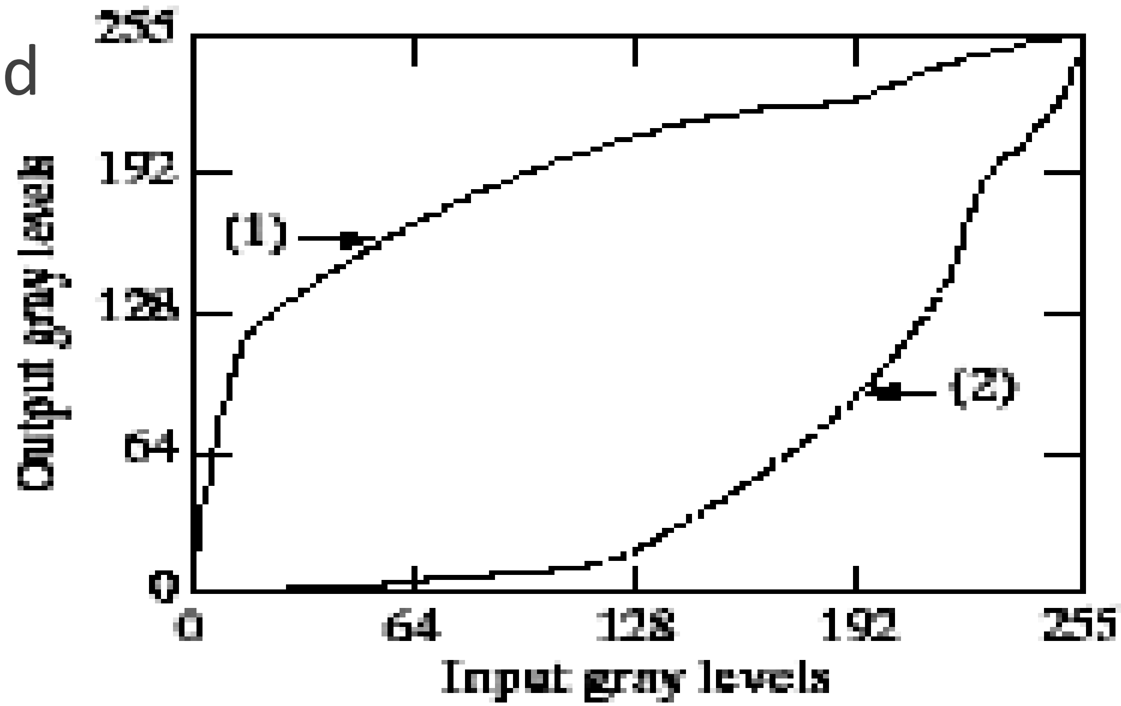
Histogram Specification

(1) the transformation function G(z) obtained from

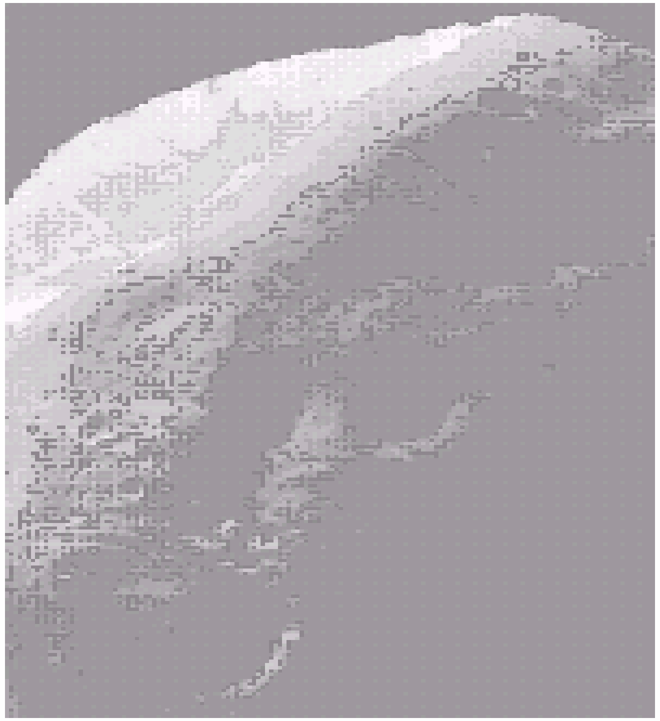
$$G(z_k) = (L-1)\sum_{i=0}^{k} p_z(z_i) = s_k$$

$$k = 0,1,2,\ldots,L-1$$

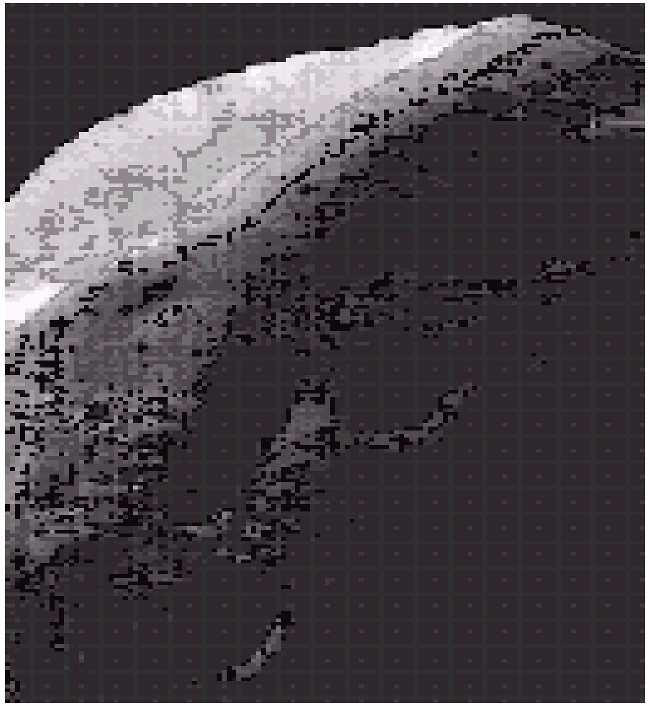
(2) the inverse transformation $G^{-1}(s)$



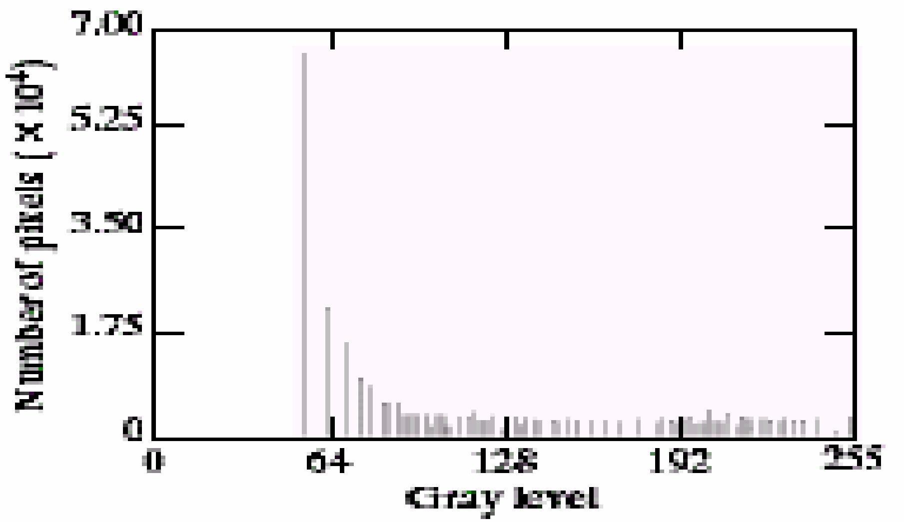
Resultant image and its histogram



Original image



Enhanced image obtained using mapping from $G^{-1}(s)$



The output image's histogram

Notice that the output histogram's low end has shifted right toward the lighter region of the gray scale as desired.

Note

Histogram specification is a trial-and-error process

There are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Note

Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.

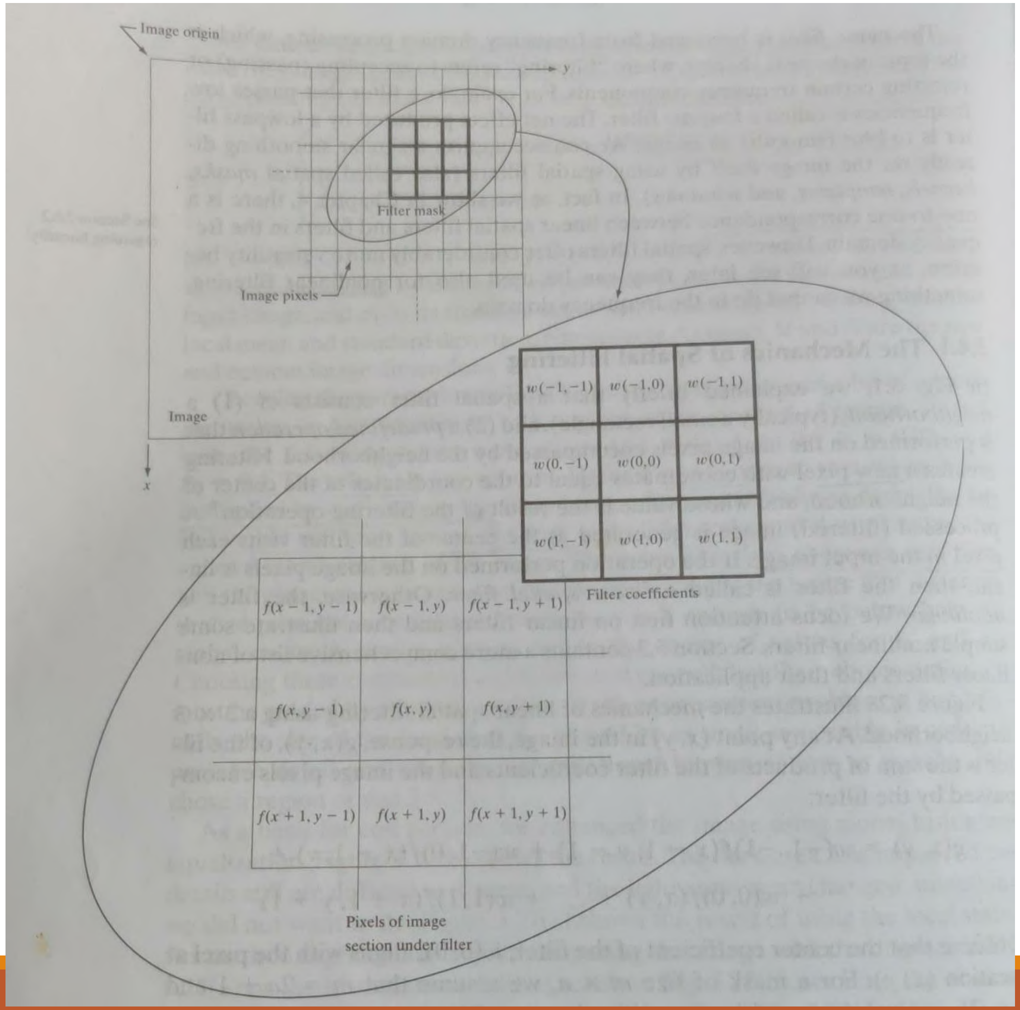
Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

Note

Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.

Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.

Local Enhancement

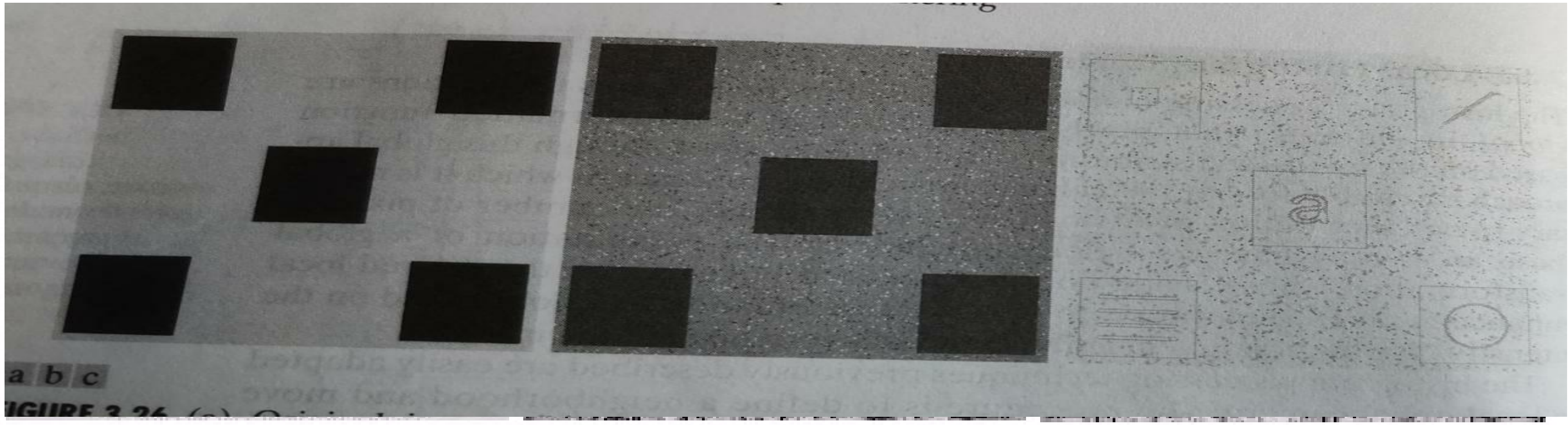


Define a square or rectangular neighborhood and move the center of this area from pixel to pixel.

At each location, the histogram of the points in the neighborhood is computed and either histogram equalization or histogram specification transformation function is obtained.

Another approach used to reduce computation is to utilize nonoverlapping regions, but it usually produces an undesirable checkerboard effect.

Local Enhancement



(a) (b) (c)

a) Original image (slightly blurred to reduce noise)
b) global histogram equalization (enhance noise & slightly increase contrast but the construction is not changed)
c) local histogram equalization using 3x3 neighborhood (reveals the small squares inside larger ones of the original image.

Explanation of the result in (c)

- Basically, the original image consists of objects inside the larger dark ones.
- Figure (c) is obtained using local histogram equalization with a neighborhood of size 3 * 3. Here we can see significant detail contained within the dark squares.
- The intensity of these objects were too close to the intensity of the large squares, and their size were too small, to influence global histogram equalization significantly enough to show this detail.
- So, when we use the local enhancement technique, it reveals the small areas.
- Note also the finer noise texture is resulted by the local processing using relatively small neighborhoods.

Enhancement using Arithmetic/Logic Operations

Arithmetic/Logic operations perform on pixel by pixel basis between two or more images

except NOT operation which perform only on a single image

Logic Operations

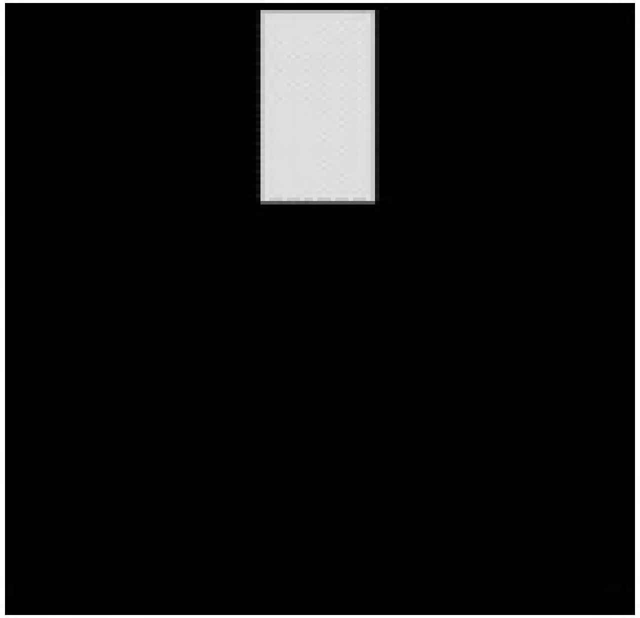
Logic operation performs on gray-level images, the pixel values are processed as binary numbers

light represents a binary 1, and dark represents a binary 0

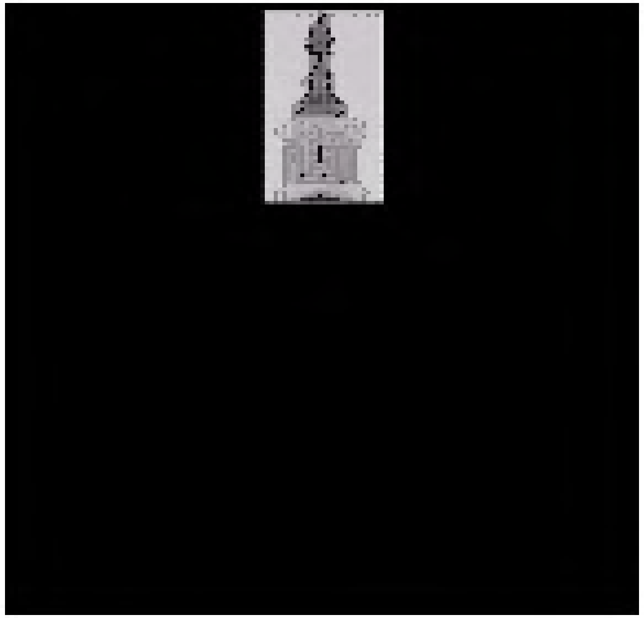
NOT operation = negative transformation

Example of AND Operation

original image



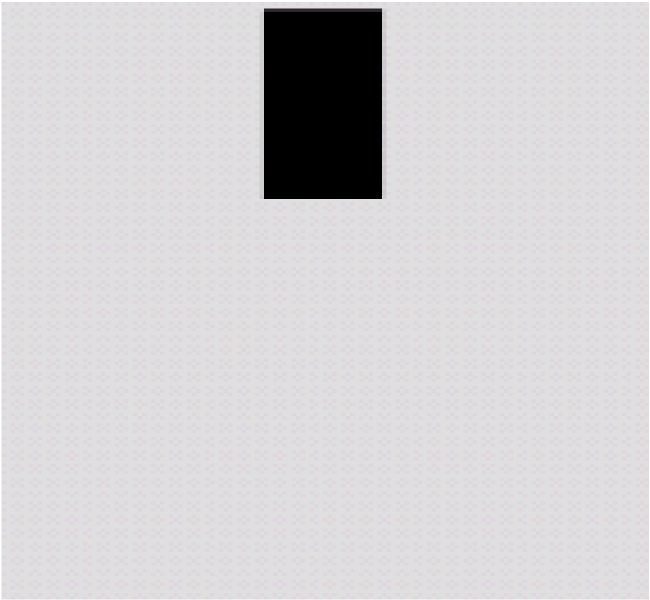
AND image
mask



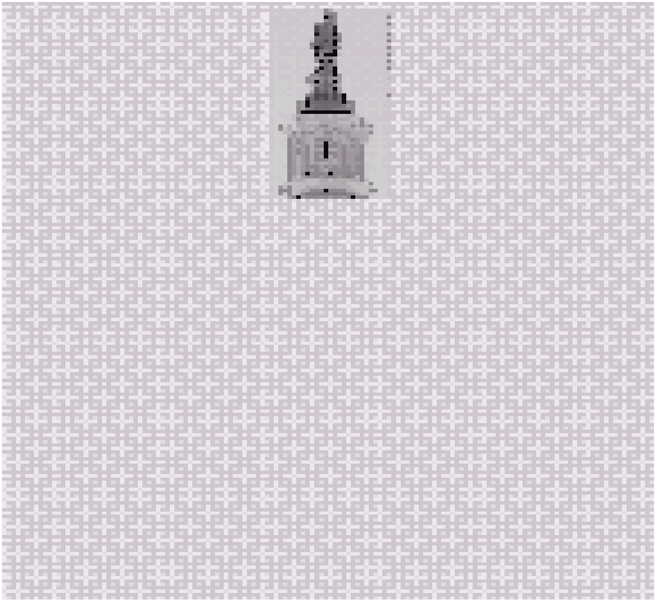
result of AND operation

Example of OR Operation

original image



OR image
mask



result of OR operation

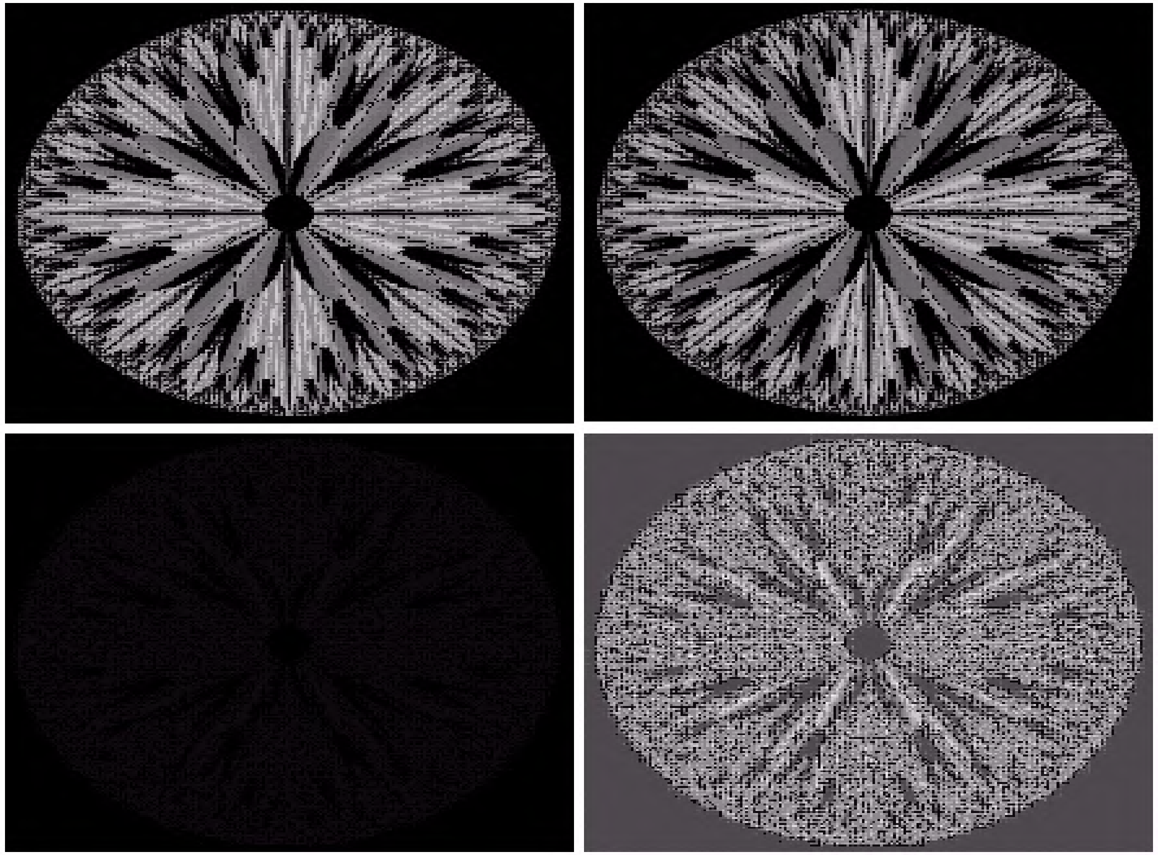
Image Subtraction

$$g(x,y) = f(x,y) - h(x,y)$$

enhancement of the differences between images

a	b
c	d

Image Subtraction



a). original fractal image

b). result of setting the four lower-order bit planes to zero

- refer to the bit-plane slicing
- the higher planes contribute significant detail
- the lower planes contribute more to fine detail
- image b). is nearly identical visually to image a), with a very slightly drop in overall contrast due to less variability of the gray-level values in the image.

c). difference between a). and b). (nearly black)

d). histogram equalization of c). (perform contrast stretching transformation)

Note

We may have to adjust the gray-scale of the subtracted image to be [0, 255] (if 8-bit is used)

- first, find the minimum gray value of the subtracted image
- second, find the maximum gray value of the subtracted image
- set the minimum value to be zero and the maximum to be 255
- while the rest are adjusted according to the interval [0, 255], by timing each value with 255/max

Subtraction is also used in segmentation of moving pictures to track the changes

- after subtract the sequenced images, what is left should be the moving elements in the image, plus noise

Suggested Readings

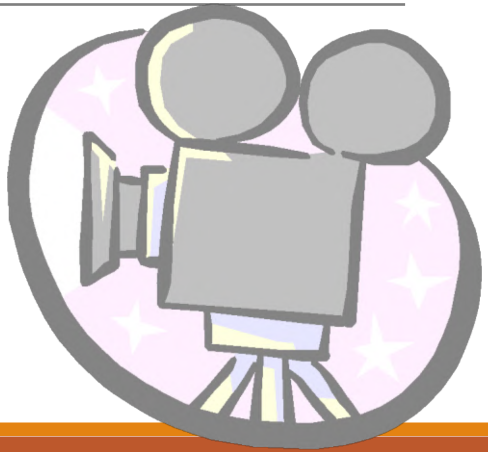
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Spatial Filtering
- Spatial Correlation and convolution
- Smoothening Filter
- Sharpening Filter

Spatial Filtering

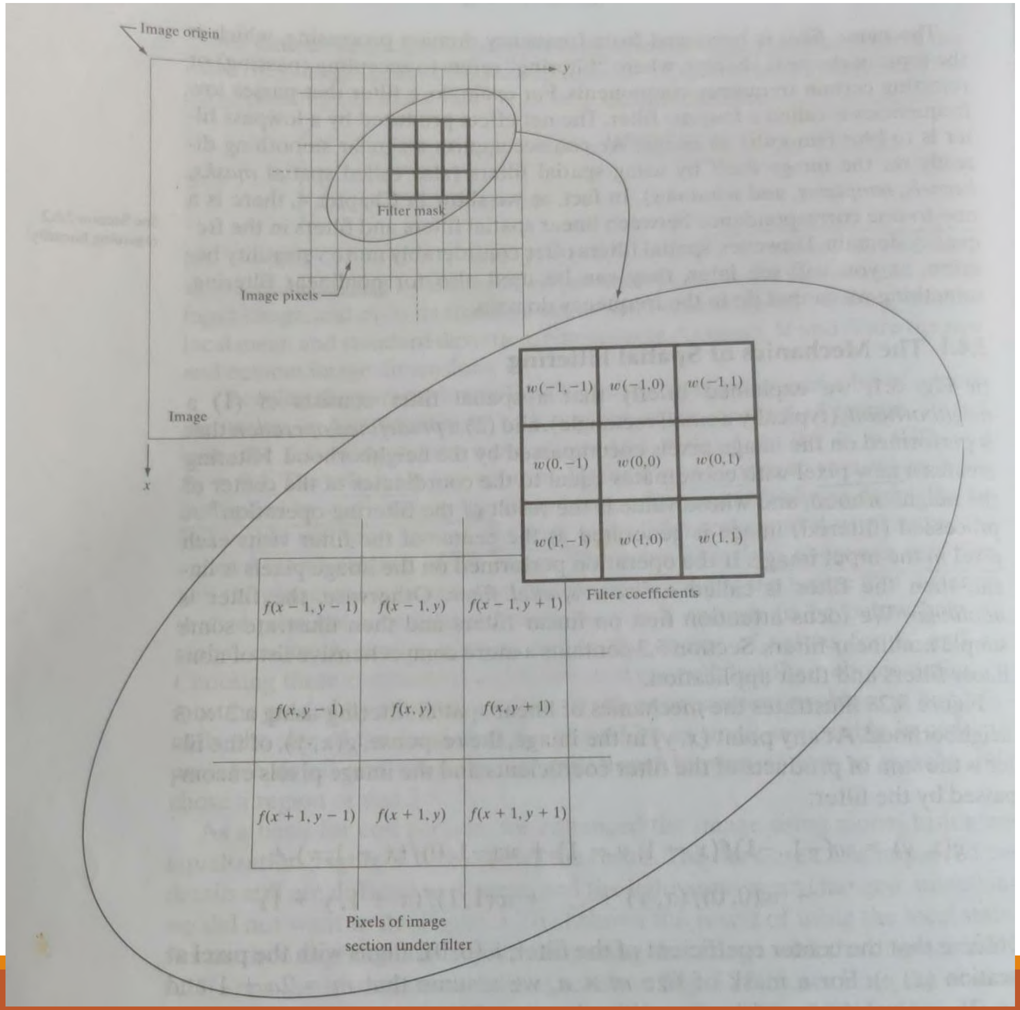
use filter (can also be called as mask/kernel/template or window)

the values in a filter subimage are referred to as coefficients, rather than pixel.

our focus will be on masks of odd sizes, e.g. 3x3, 5x5,…

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Mechanism of linear spatial filtering using 3 * 3 mask



Define a square or rectangular neighborhood and move the center of this area from pixel to pixel.

Convolution

Consider an image of size 6 x 6

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

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter /mask

stride=1

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

6 x 6 image

Dot product

3 -1

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter /mask

stride=1

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

Dot product

-1 -3

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter /mask

stride=1

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

Dot product

-3 -1

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter /mask

If stride=2

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

6 x 6 image

3 -3

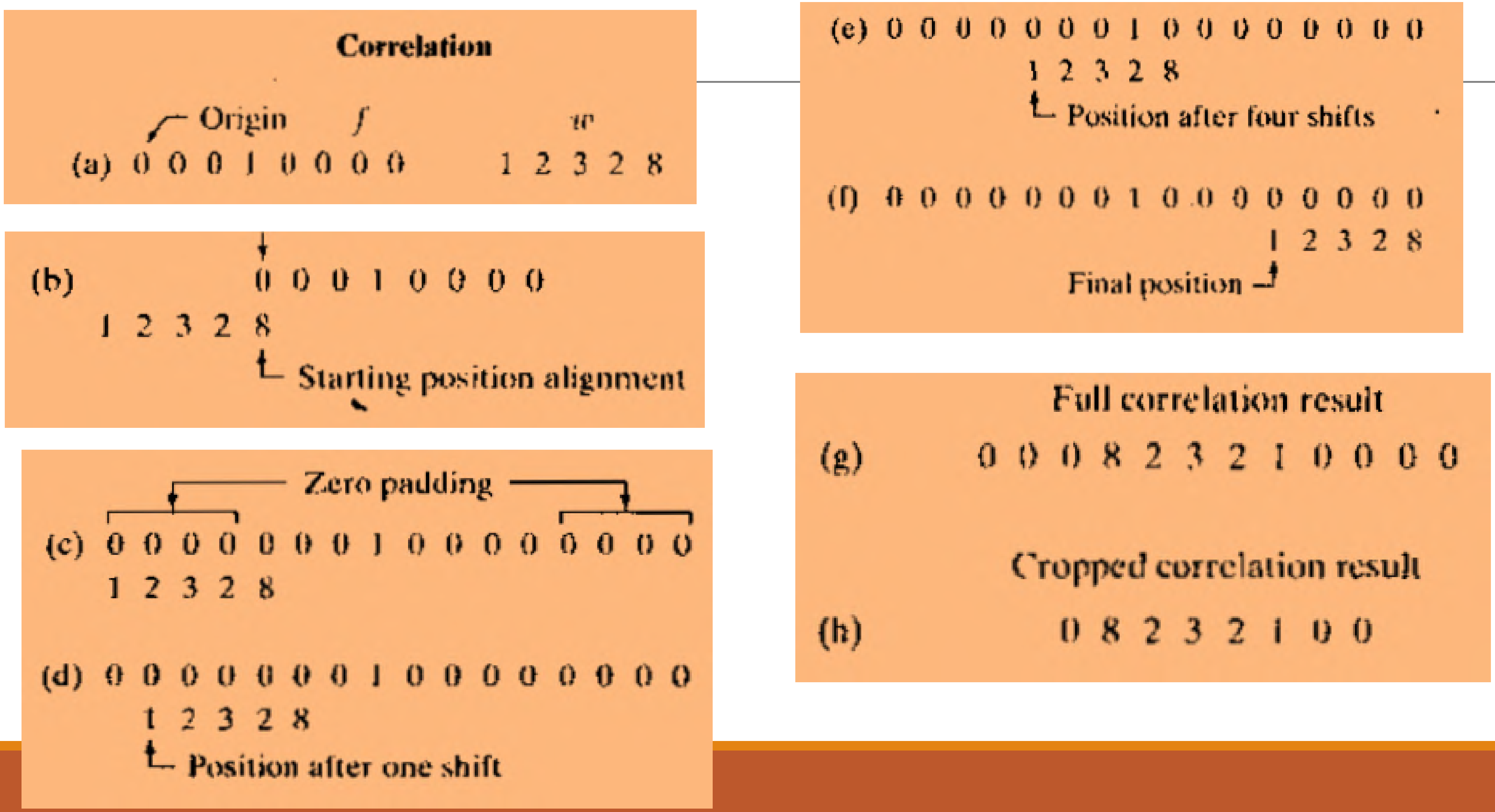
Spatial Correlation and Convolution

There are two closely related concepts that must be understood clearly when performing linear spatial filtering.

- Correlation
- Convolution

- Correlation is the process of moving a filter mask over the image and computing sum of products at each location.
- The mechanism of convolution is same, except that the filter is first rotated by 180^o.

Spatial Correlation Example: consider a 1D function f with mask w



Spatial Convoution Example: consider a 1D function f with mask w

Convolution

a

Origin f w rotated 180°

0 0 0 1 0 0 0 0 8 2 3 2 1

b

0 0 0 1 0 0 0 0

8 2 3 2 1

c

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

8 2 3 2 1

d

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

8 2 3 2 1

e

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

8 2 3 2 1

f

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

8 2 3 2 1

g

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0

h

Cropped convolution result

0 1 2 3 2 8 0 0

Correlation for 2 D function f(x,y)

Origin $f(x, y)$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} & w(x, y) & \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

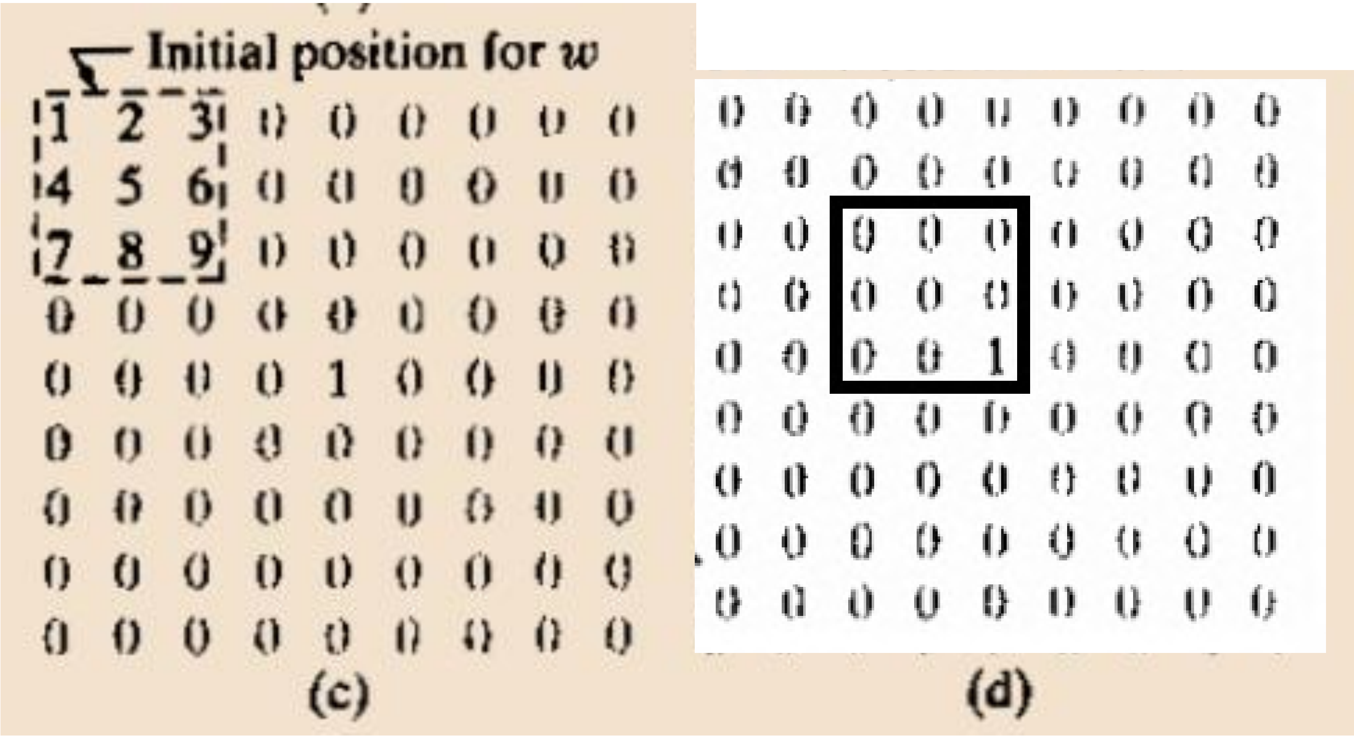
(a)

Padded f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(b)

Correlation for 2 D function f(x,y)



Correlation for 2 D function f(x,y)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Correlation for 2 D function f(x,y)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Spatial Filtering Process

simply move the filter mask from point to point in an image.

at each point (x,y), the response of the filter at that point is calculated using a predefined relationship.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$
$$= \sum_{i=i}^{mn} w_i z_i$$

Linear Filtering

Linear Filtering of an image f of size MxN filter mask of size mxn is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

where a = (m-1)/2 and b = (n-1)/2

To generate a complete filtered image this equation must be applied for x = 0, 1, 2, … , M-1 and y = 0, 1, 2, … , N-1

Smoothing Spatial Filters

used for blurring and for noise reduction

blurring is used in preprocessing steps, such as

- removal of small details from an image prior to object extraction
- bridging of small gaps in lines or curves

noise reduction can be accomplished by blurring with a linear filter and also by a nonlinear filter

Smoothing Linear Filters

output is simply the average of the pixels contained in the neighborhood of the filter mask.

called averaging filters or lowpass filters.

Smoothing Linear Filters

replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the “sharp” transitions in gray levels.

sharp transitions

- random noise in the image
- edges of objects in the image

thus, smoothing can reduce noises (desirable) and blur edges (undesirable)

3x3 Smoothing Linear Filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

box filter

weighted average

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Weighted average filter

the basic strategy behind weighting the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply **an attempt to reduce blurring in the smoothing process**.

General form : smoothing mask

filter of size mxn (m and n odd)

$$g(x,y)=\frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

summation of all coefficient of the mask

a	b
c	d
e	f

Example

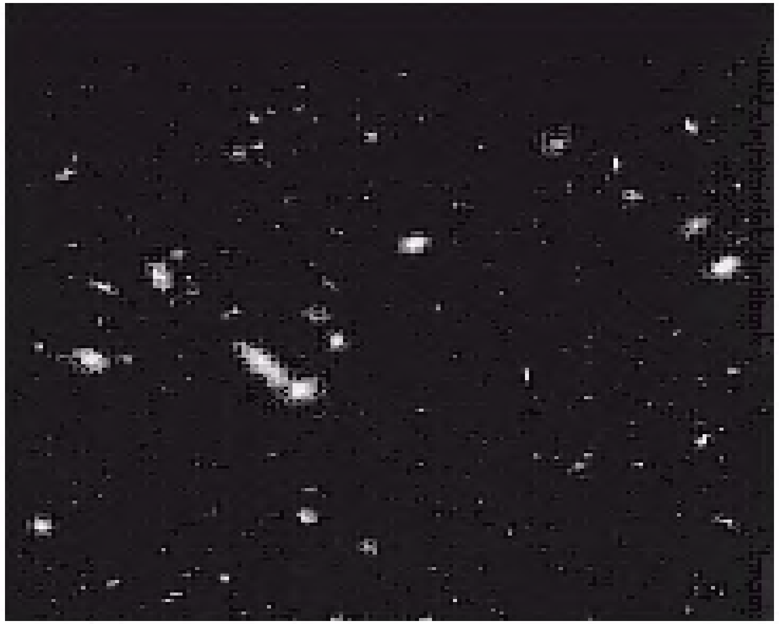
a) original image 500x500 pixel

b) - f) results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.

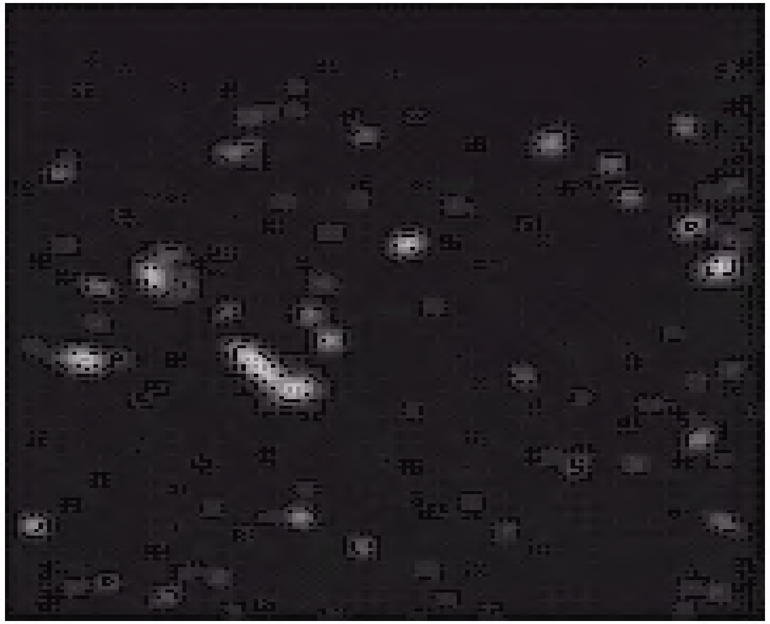
Note:

- big mask is used to eliminate small objects from an image.
- the size of the mask establishes the relative size of the objects that will be blended with the background.

Example



original image



result after smoothing with 15x15 averaging mask

result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the image.

Order-Statistics Filters (Nonlinear Filters)

the response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter

example

- median filter : $R = \text{median}\{z_k \mid k = 1,2,\ldots,n \times n\}$
- max filter : $R = \max\{z_k \mid k = 1,2,\ldots,n \times n\}$
- min filter : $R = \min\{z_k \mid k = 1,2,\ldots,n \times n\}$

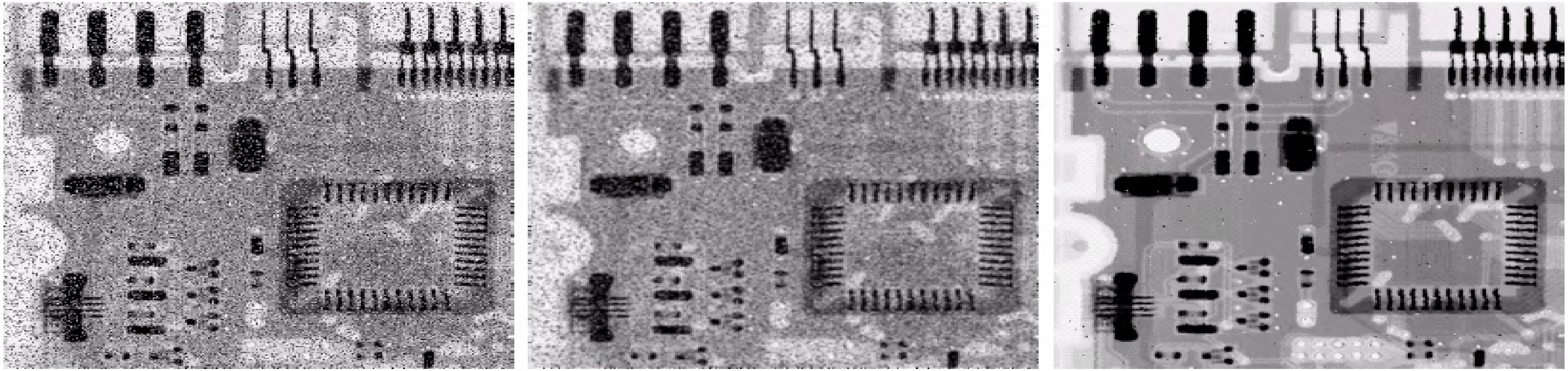
note: n x n is the size of the mask

Median Filters

replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)

quite popular because for certain types of random noise (impulse noise ⇨ salt and pepper noise) , they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.

Example : Median Filters



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Suggested Readings

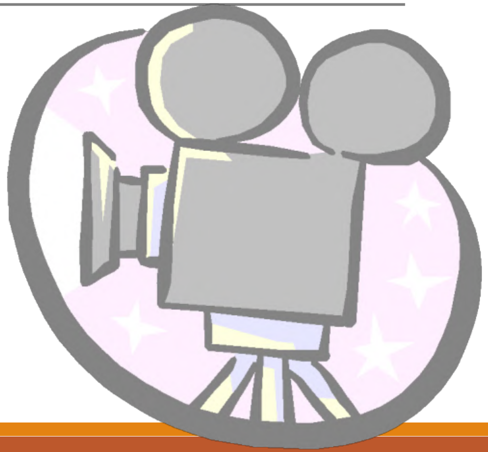
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



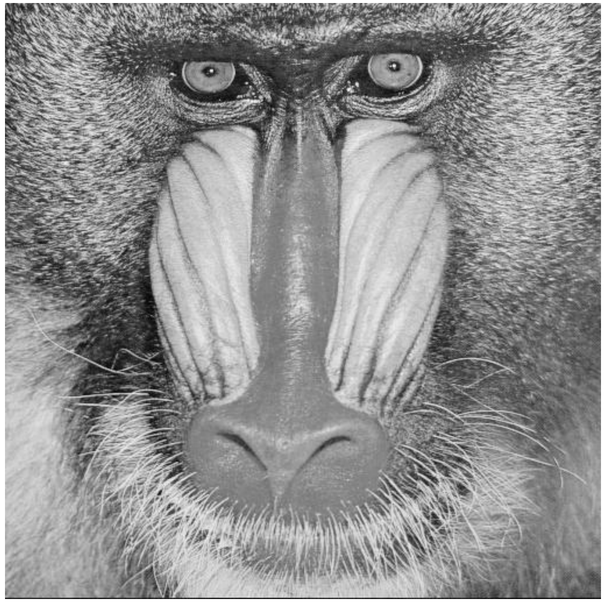
Outline

- Spatial Filtering
    - Sharpening Filter

Sharpening Spatial Filters

to highlight fine detail in an image

or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.



Blurring vs. Sharpening

as we know that blurring can be done in spatial domain by pixel averaging in a neighbors

since averaging is analogous to integration

thus, we can guess that the sharpening must be accomplished by spatial differentiation.

Derivative operator

the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

thus, image differentiation

- enhances edges and other discontinuities (noise)
- deemphasizes area with slowly varying gray-level values.

First-order derivative

a basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$or$$

$$\frac{\partial f}{\partial x} = f(x) - f(x-1)$$

Second-order derivative

similarly, we define the second-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - f(x) - f(x) + f(x-1)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

First and Second-order derivative of f(x,y)

when we consider an image function of two variables, f(x,y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

$$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator

(linear operator)

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

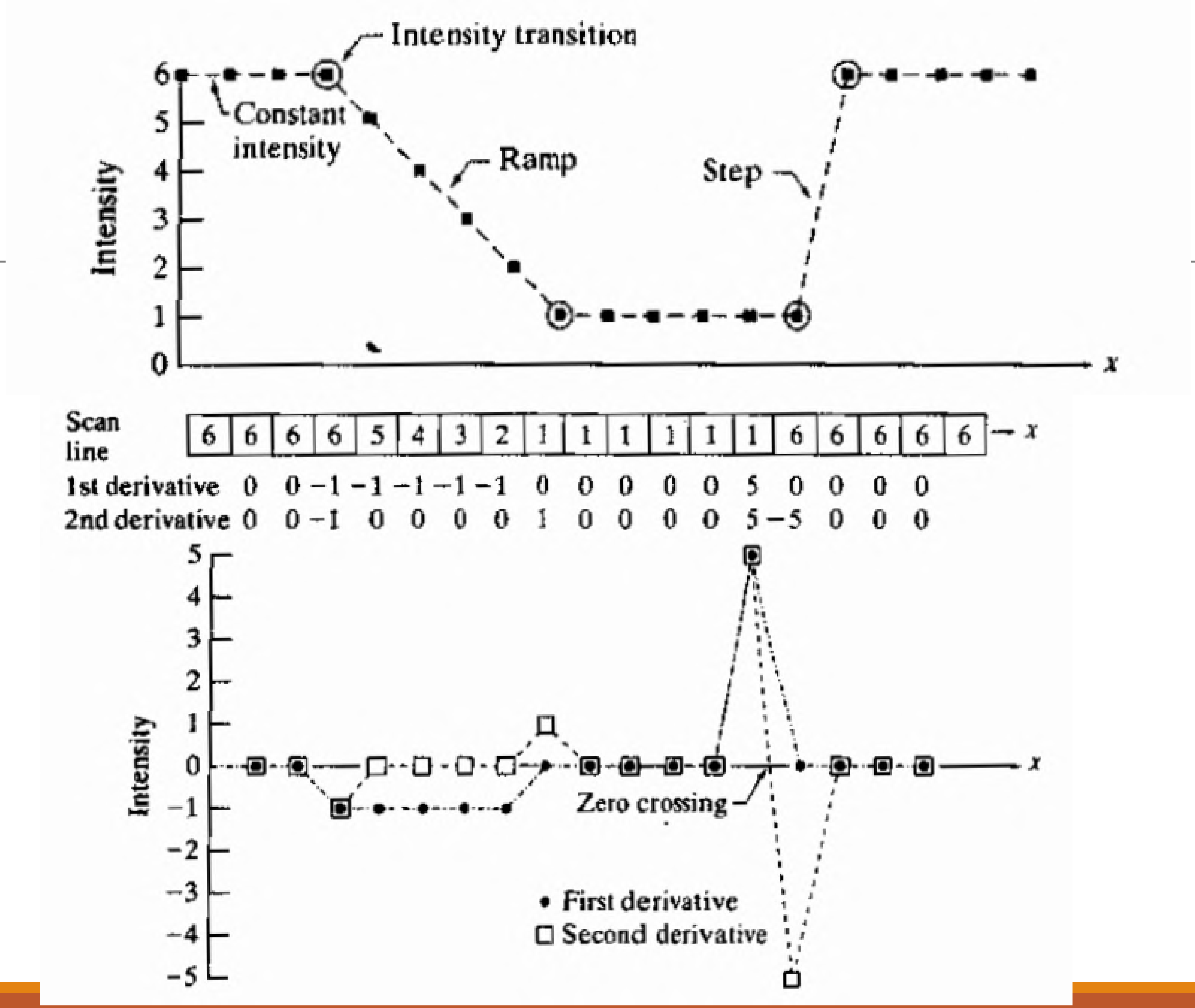
Discrete Form of Laplacian

We have,

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

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

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


Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line 6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Intensity
Zero crossing
First derivative
Second derivative
x

Result Laplacian mask

0	1	0
1	−4	1
0	1	0

Laplacian mask implemented an extension of diagonal neighbors

1	1	1
1	−8	1
1	1	1

Other implementation of Laplacian masks

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

-1	-1	-1
-1	8	-1
-1	-1	-1

give the same result, but we have to keep in mind that when combining (add / subtract) a Laplacian-filtered image with another image.

Effect of Laplacian Operator

as it is a derivative operator,

- it highlights gray-level discontinuities in an image
- it deemphasizes regions with slowly varying gray levels

tends to produce images that have

- grayish edge lines and other discontinuities, all superimposed on a dark,
- featureless background.

Unsharp masking

A process that has been used for many years by printing and publishing industry to sharpen images. The process consists of the following steps

1. Blur the original image.

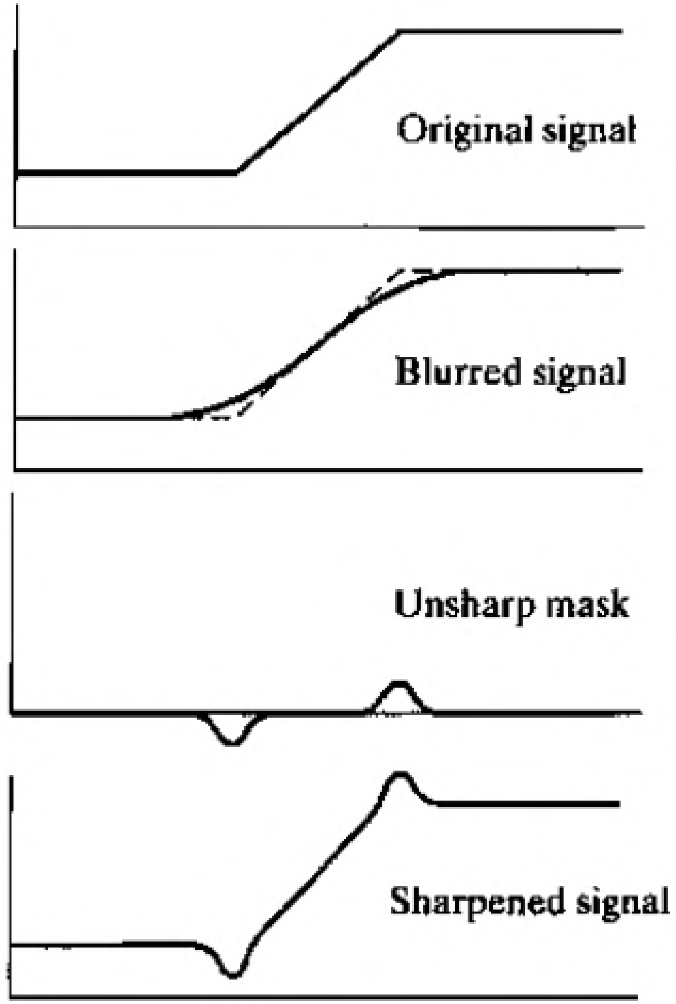
2. Subtract the blurred image from the original (resulting difference is called mask).

3. Add mask to the original.

Unsharp masking

a
b
c
d

FIGURE : 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).



Unsharp masking

Let $\bar{f}(x, y)$ denote the blurred image, the unsharp masking is expressed as:

$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

mask= original image – blurred image

to subtract a blurred version of an image produces sharpening output image.

Then we add a weighted portion of the mask back to the original image:

$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

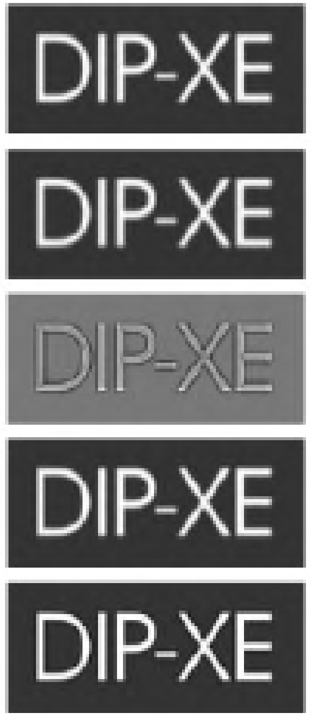
Here, we included a weight k ($k \geq 0$), for generality.

When k=1, we have **unsharp masking**.

When k>1, the Process is referred to as **highboost filtering**.

Choosing k<1 de-emphasizes the contribution of the unsharp mask.

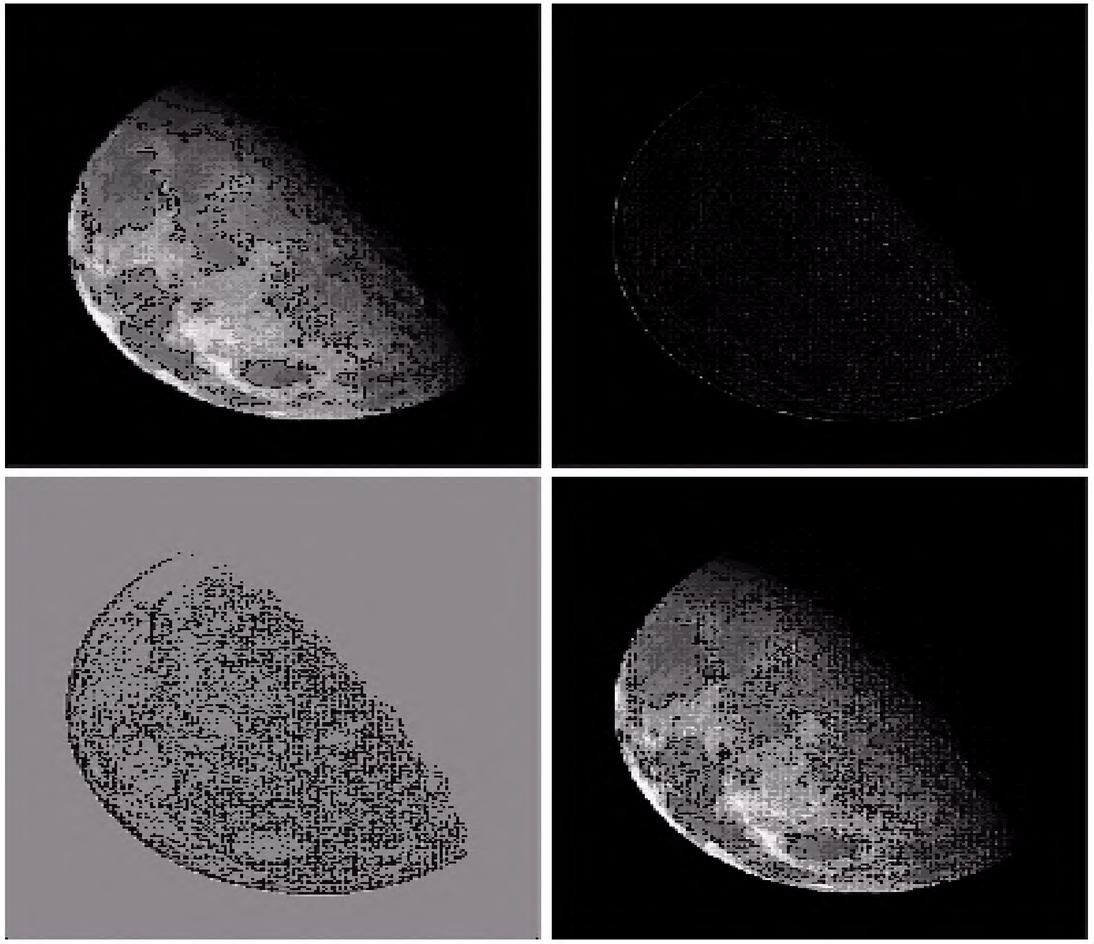
Example



a
b
c
d
e

FIGURE
(a) Original image.
(b) Result of blurring with a Gaussian filter.
(c) Unsharp mask. (d) Result of using unsharp masking.
(e) Result of using highboost filtering.

Example



a). image of the North pole of the moon

b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

c). Laplacian image scaled for display purposes [0, L-1]

d). image enhanced by addition with original image

Gradient Operator

first order derivatives are implemented using the magnitude of the gradient.

For a function f(x,y), the gradient of f at coordinates (x, y) is defined as

$$\nabla \mathrm{f} = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient Operator

$$\nabla \mathrm{f} = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

the magnitude (length) of the gradient of vector (∇f) is denoted as M(x,y) or mag(∇f)

$$|\nabla f| = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

$$|\nabla f| \approx |G_x| + |G_y|$$

the magnitude becomes nonlinear

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

simplest approximation, 2x2

$$\nabla \mathrm{f} = grad(f) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$G_x = \frac{\partial f}{\partial x} = f(x+1, y) - f(x, y) = z_8 - z_5$$

$$G_y = \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y) = z_6 - z_5$$

G_x

-1	0
1	0

G_y

-1	1
0	0

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

simplest approximation, 2x2

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$

Cross-gradient operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Another definition proposed by Roberts in the early development of image processing use cross difference, called Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

G_x

−1	0
0	1

G_y

0	−1
1	0

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$
$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \approx |G_x| + |G_y|$$

the weight value 2 is to achieve smoothing by giving more important to the center point

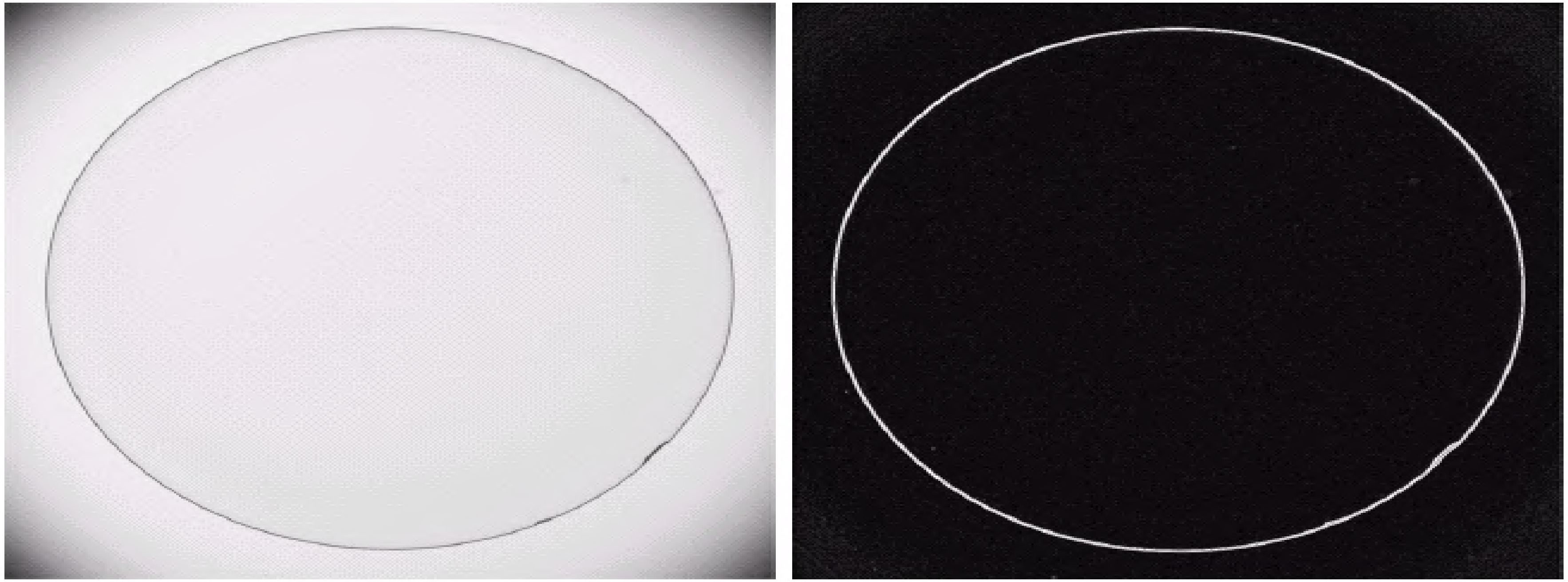
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Note

the summation of coefficients in all masks equals 0, indicating that they would give a response of 0 in an area of constant gray level.

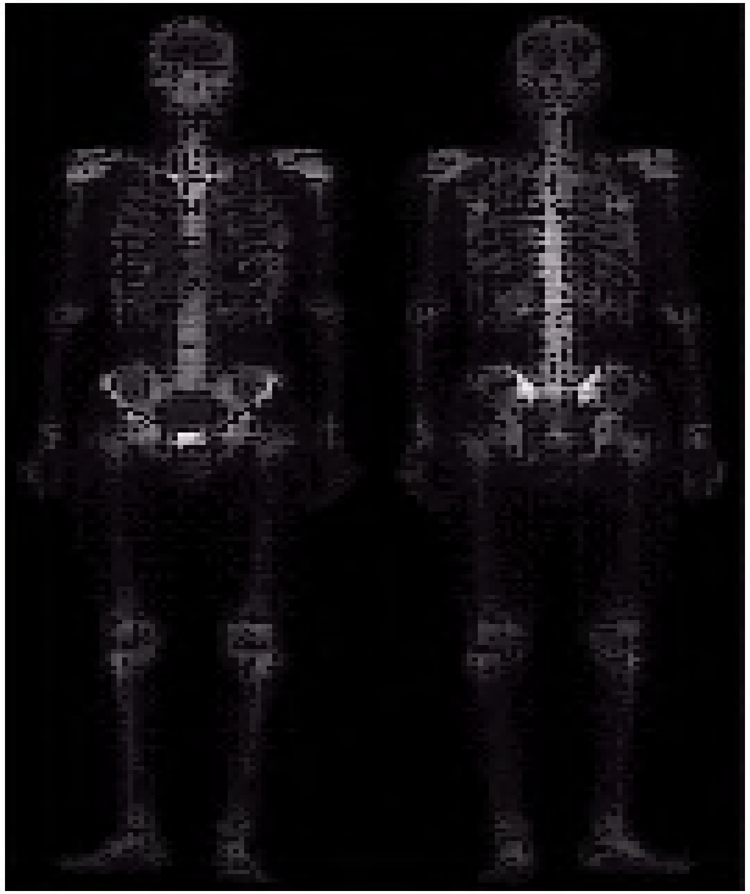
Example



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Example of Combining Spatial Enhancement Methods



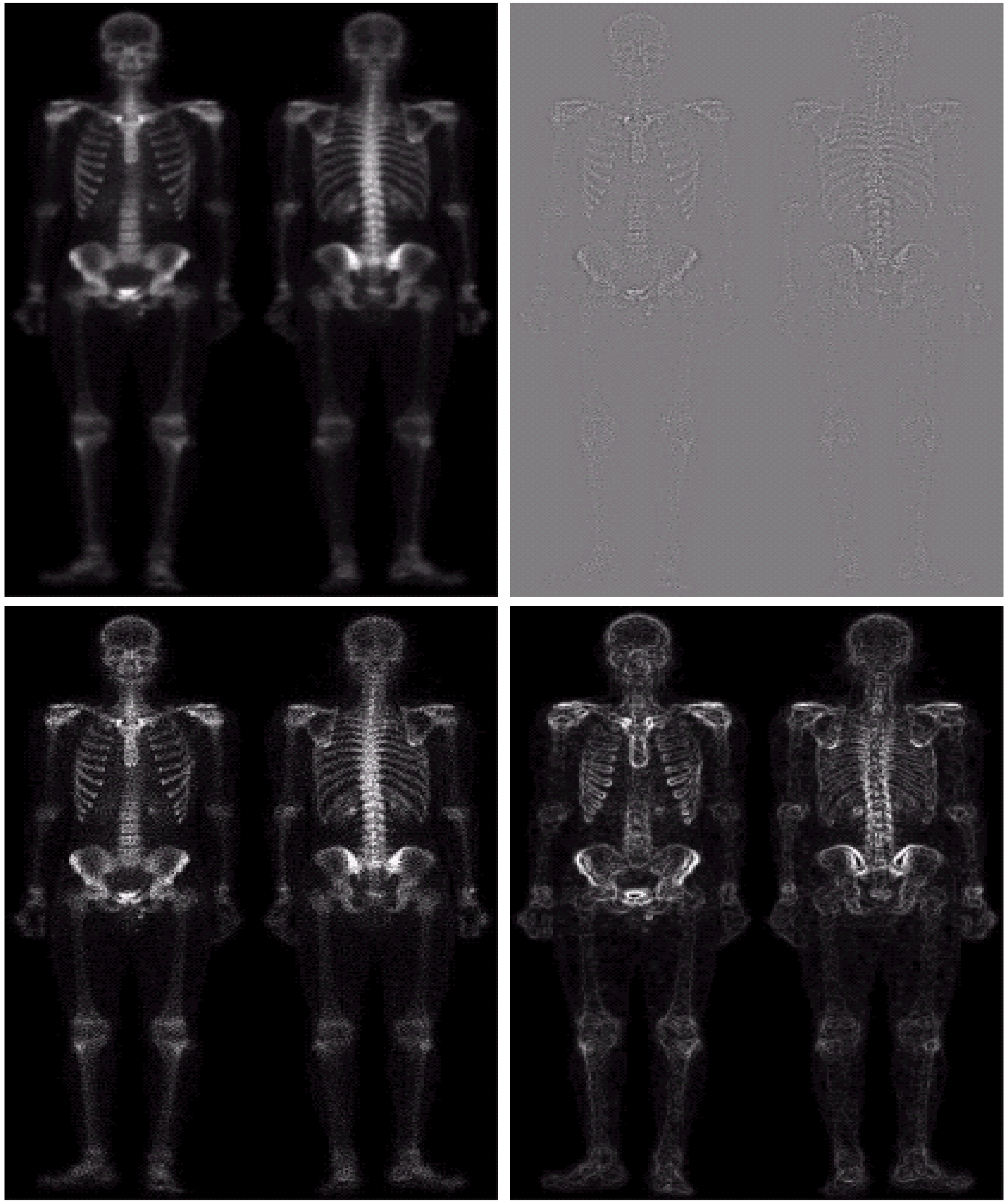
want to sharpen the original image and bring out more skeletal detail.

problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

Example of Combining Spatial Enhancement Methods

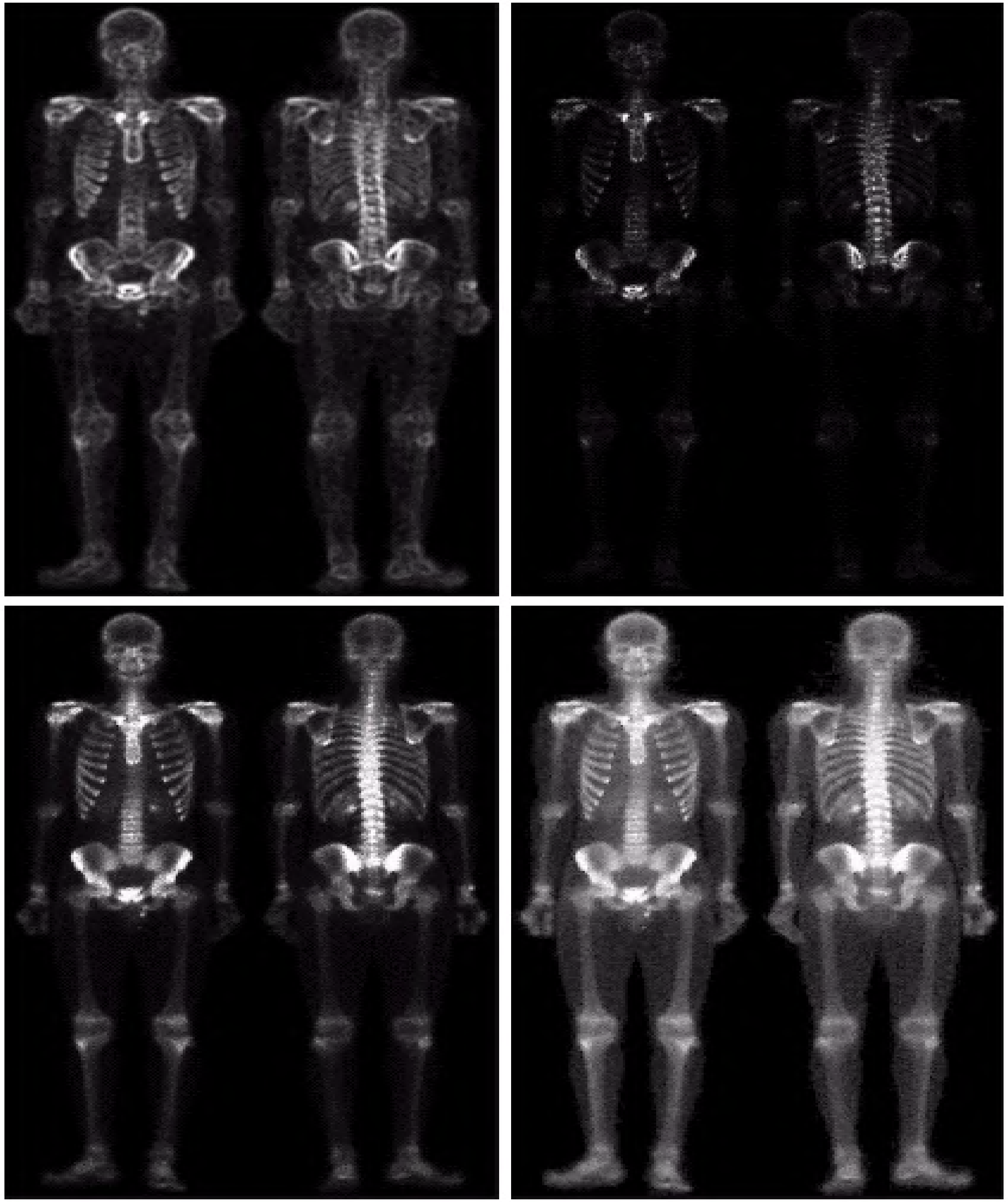
solve :

1. Laplacian to highlight fine detail
2. gradient to enhance prominent edges
3. gray-level transformation to increase the dynamic range of gray levels



a	b
c	d

FIGURE 3.46
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).



e	f
g	h

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Correct the effect of featureless background

easily by adding the original and Laplacian image.

be careful with the Laplacian filter used

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

Mask of Laplacian + addition

to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.

Mask of Laplacian + addition

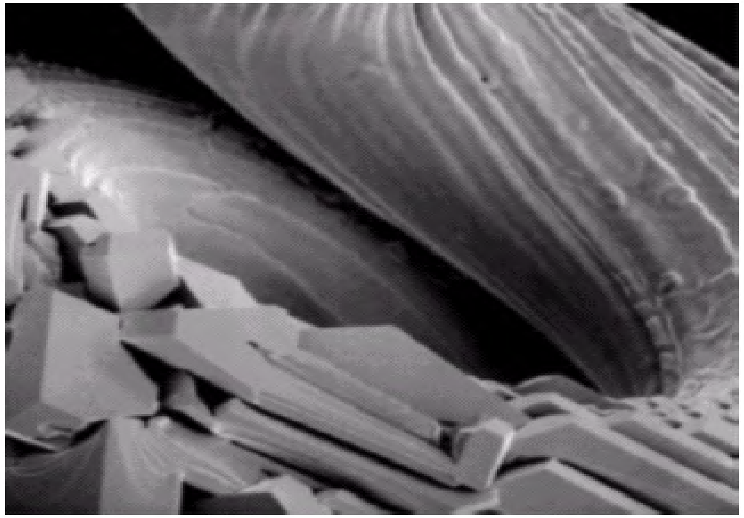
$$
\begin{aligned}
g(x,y) = {} & f(x,y) - [f(x+1,y) + f(x-1,y) \\
& + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\
= {} & 5f(x,y) - [f(x+1,y) + f(x-1,y) \\
& + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

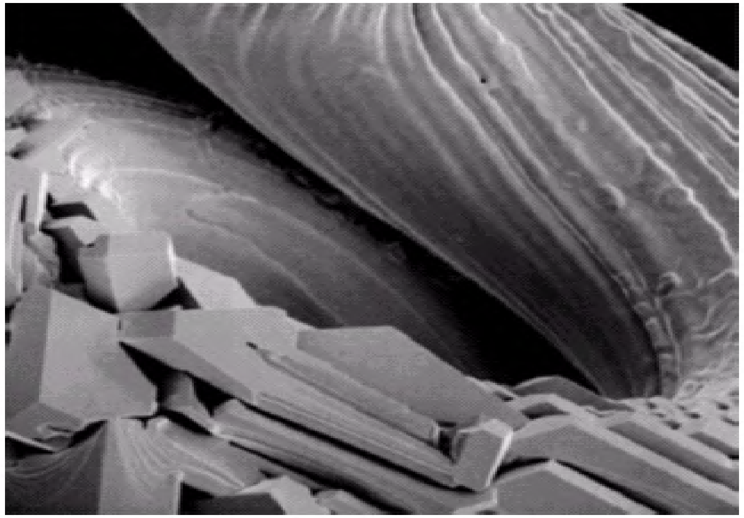
0	-1	0
-1	5	-1
0	-1	0

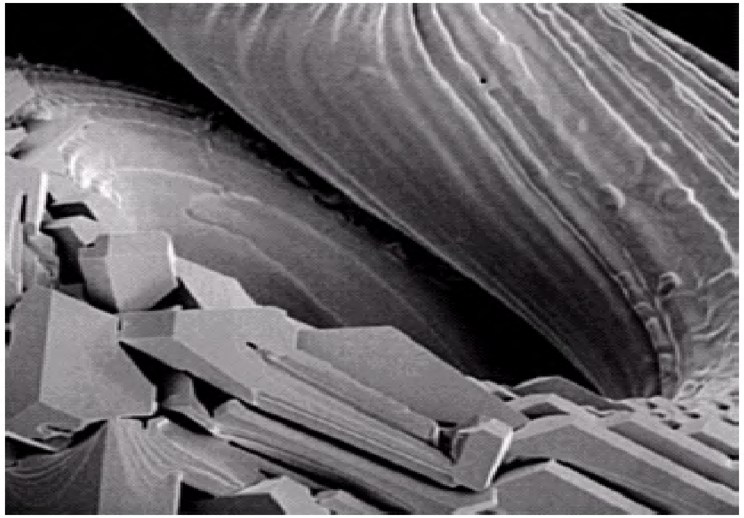
Example

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1







a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Note

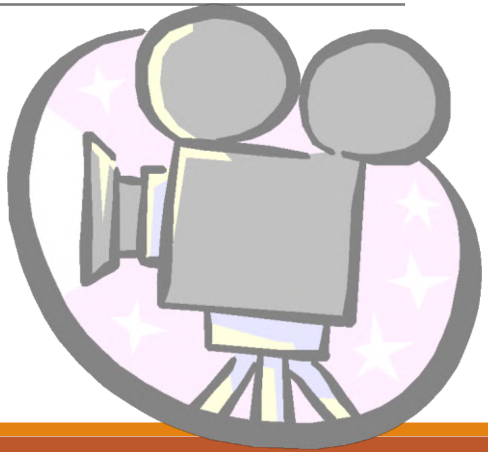
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 9 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Image Processing

CS-317/CS-341



Outline

- Image Enhancement in the Frequency Domain
    - Fourier Transformation

Background ….

- Jean Baptise Joseph (B. 1768 in town of Auxerre, France)
- Most important contribution was outlined in a memoir in 1807 and published in 1822 in his book, "La Theorie Analitique de la Chaleur (The Analytic Theory of Heat)"

Contribution

Any function that periodically repeats itself and satisfy the some mild mathematical condition, does not matter how much complicated the function is, can be expressed as the sum of sines and / or cosines of different frequencies, each multiplied by a different coefficients.

➢ **Now this sum is called as Fourier Series**

Background….

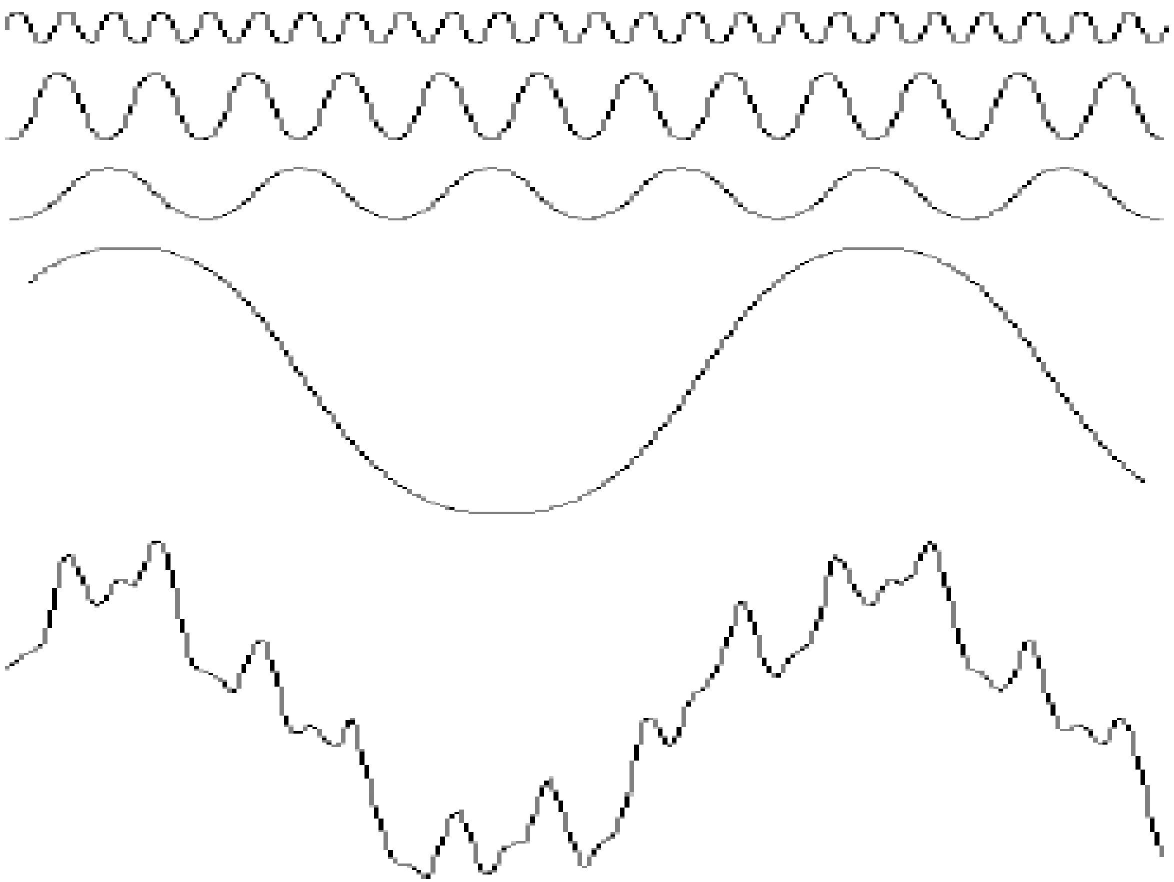
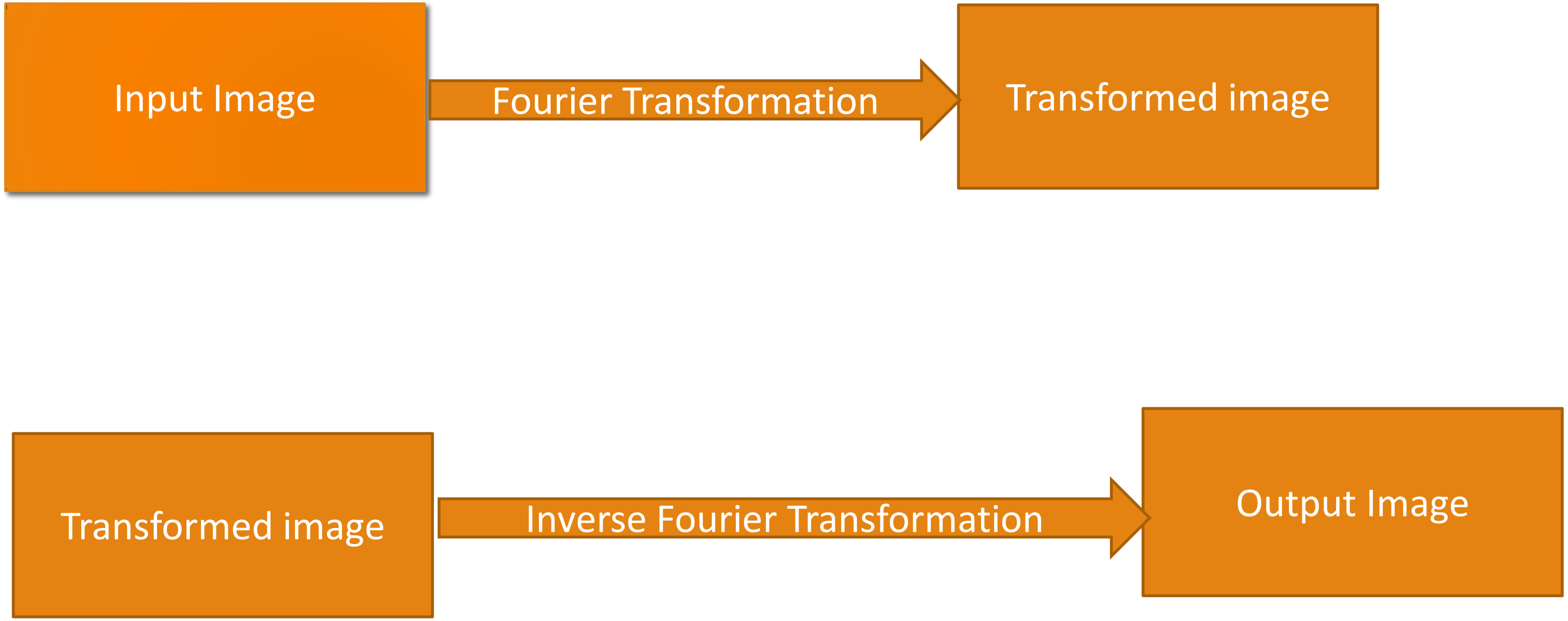


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Background….

- Even functions that are not periodic, but area under the curve is finite, can be expressed as the integral of sines and/or cosines multiplied by a weighting function.
- This representation is called as Fourier Transform. It is used very much in most of the practical applications.
- These representation share most important characteristics that a function can be perfectly reconstructed (recovered) completely via an inverse process, with no loss of information.
- For finite duration / support (e.g. image) Fourier transform is most important tool.

Fourier Transform
&
Frequency Domain


Input Image
Fourier Transformation
Transformed image
Transformed image
Inverse Fourier Transformation
Output Image

1-D Continuous Fourier Transform

Let f(x) is a 1-D function

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j2\pi ux} dx; \quad f(x) = \int_{-\infty}^{+\infty} F(u) e^{j2\pi ux} du$$

$$j = \sqrt{-1}$$

$$e^{j\theta} = \cos\theta + j\sin\theta$$

2-D Continuous Fourier Transform

Let f(x,y) is a 2-D function, the Fourier transform of f(x, y) is given as:

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

The inverse Fourier Transform of F(u,v) is

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

One Dimensional
Discrete Fourier Transform

1-D Discrete Fourier Transform

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}\; ; \quad u = 0,1,2,3,\ldots\ldots,M-1$$

$$f(x) = \sum_{x=0}^{M-1} F(u)e^{j2\pi ux/M}\; ; \quad x = 0,1,2,3,\ldots\ldots,M-1$$

- Some authors include the 1/M term in eq. f(x) instead the way shown in eq. F(u). That does not affect the proof that the two equations form the Fourier series

1-D Discrete Fourier Transform

- **Frequency Domain**

Euler's Formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\left[\cos 2\pi ux / M - j\sin 2\pi ux / M\right];\ u = 0,1,\cdots,M-1$$

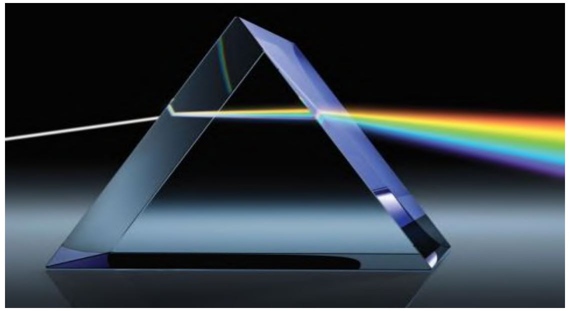
- The domain (value of u) over which the values of $F(u)$ is appropriately called the frequency domain. u determines the frequency of the component of the transform.

- Each of the M terms of $F(u)$ is called frequency component of the transform.

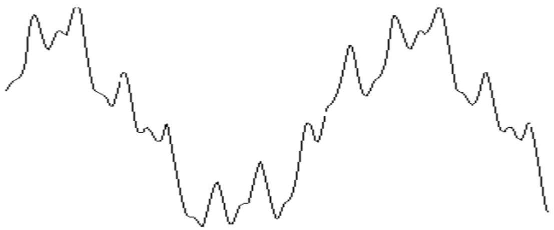
1-D Discrete Fourier Transform

- **Analogy of FT to glass prism:**

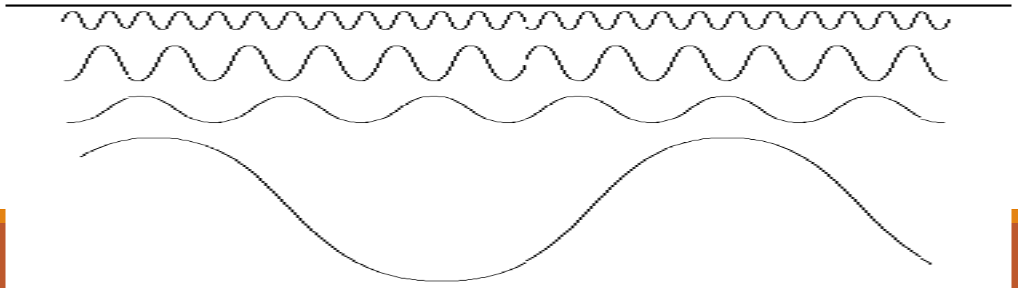
Glass prism → Physical device separates light into various color component depending on its frequency



FT as Mathematical Prism → Separates a function into various component, also based on frequency content



FT



1-D Discrete Fourier Transform

- **Frequency Domain**

Euler's Formula

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\left[\cos 2\pi ux / M - j\sin 2\pi ux / M\right]; \; u = 0,1,\cdots,M-1$$

F(u) is Complex

Properties of Fourier Transform

- **Polar Coordinate Representation of *F*(*u*):**

***F*(*u*) is Complex**

$$F(u) = |F(u)| e^{-j\phi(u)} \qquad |F(u)| = \left[R^2(u) + I^2(u) \right]^{1/2}$$

$R(u)$: Real part of $F(u)$, $I(u)$: Imaginary part of $F(u)$, $|F(u)|$: Magnitude spectrum of the Fourier Transform.

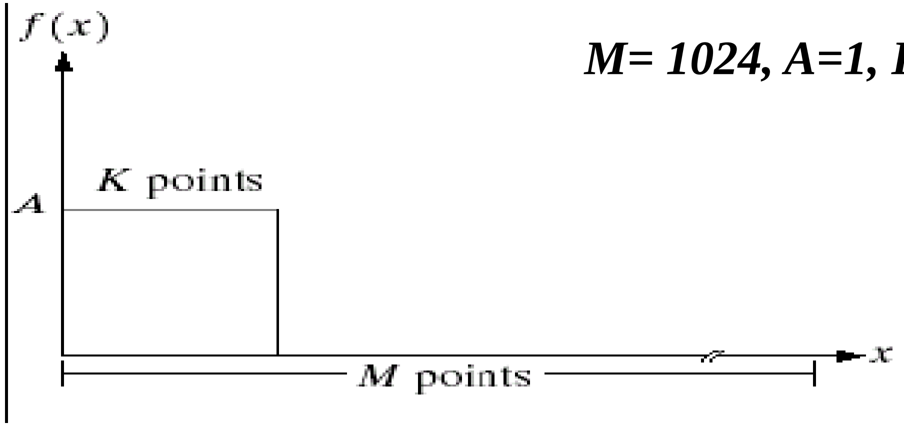
$$\phi(u) = \tan^{-1}\left[I(u)/R(u) \right]$$: phase angle or phase spectrum

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

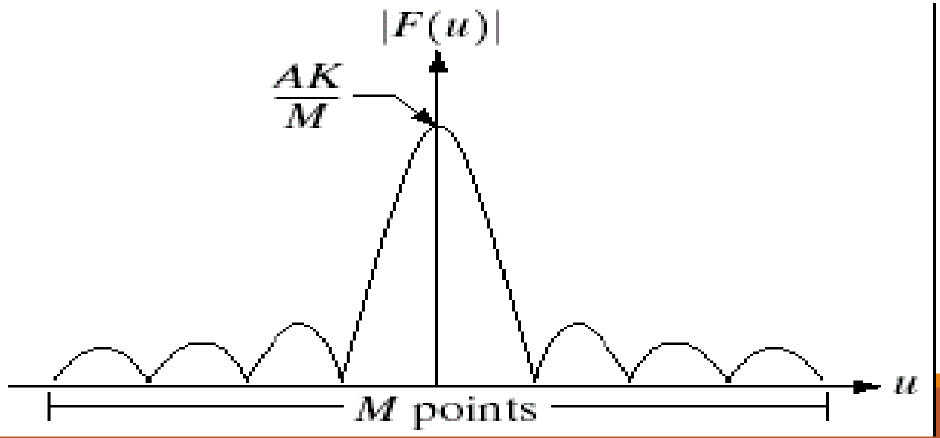
: power spectrum / power spectral density

1-D Discrete Fourier Transform

- **Example:**



M= 1024, A=1, K= 8



$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

$$= \frac{1}{M}\sum_{x=0}^{K-1} A e^{-j2\pi ux/M}$$

$$= \frac{A}{M}\sum_{x=0}^{K-1} e^{-j2\pi ux/M}$$

$$F(0) = \frac{A}{M}\sum_{x=0}^{K-1} e^{-j2\pi 0x/M}$$

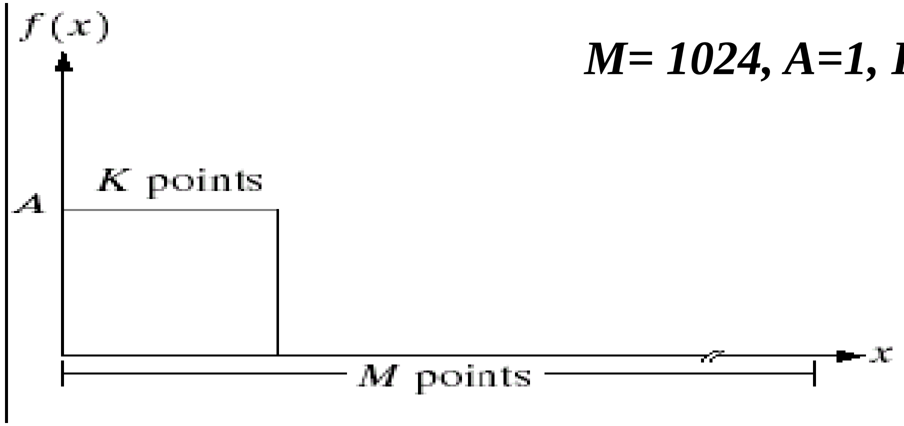
$$= \frac{A}{M}\sum_{x=0}^{K-1} 1$$

$$= \frac{AK}{M}$$

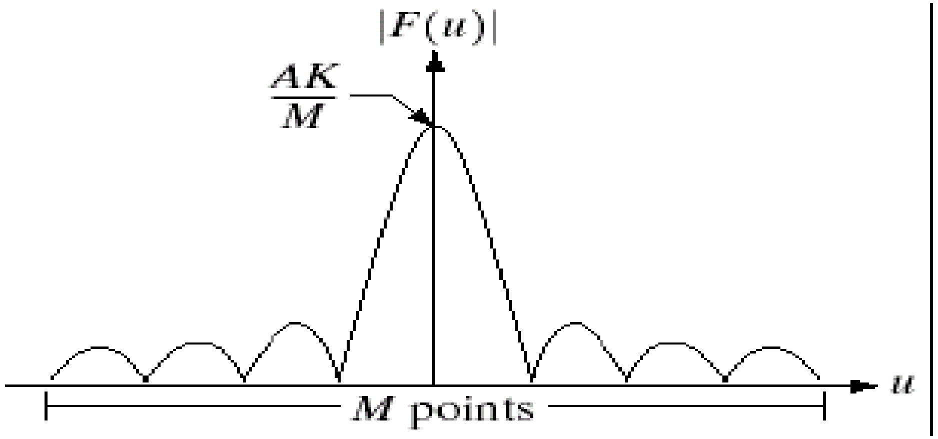
1-D Discrete Fourier Transform

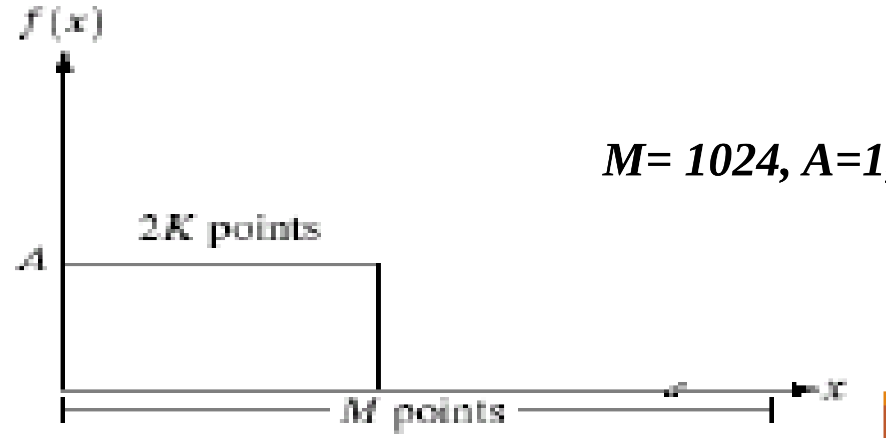
$$
\begin{aligned}
F(0) &= \frac{A}{M}\sum_{x=0}^{K-1} e^{-j2\pi 0x/M} \\
&= \frac{A}{M}\sum_{x=0}^{K-1} 1 \\
&= \frac{AK}{M}
\end{aligned}
$$

- **Example:**

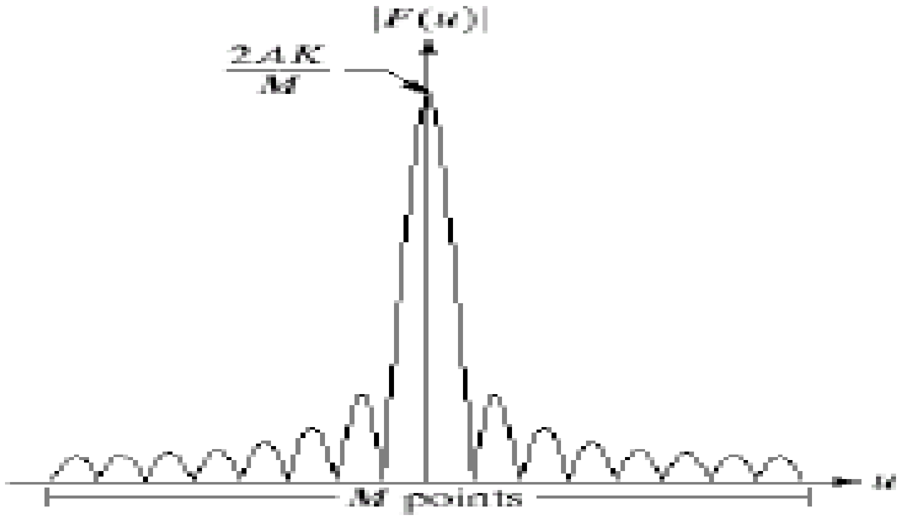


M= 1024, A=1, K= 8



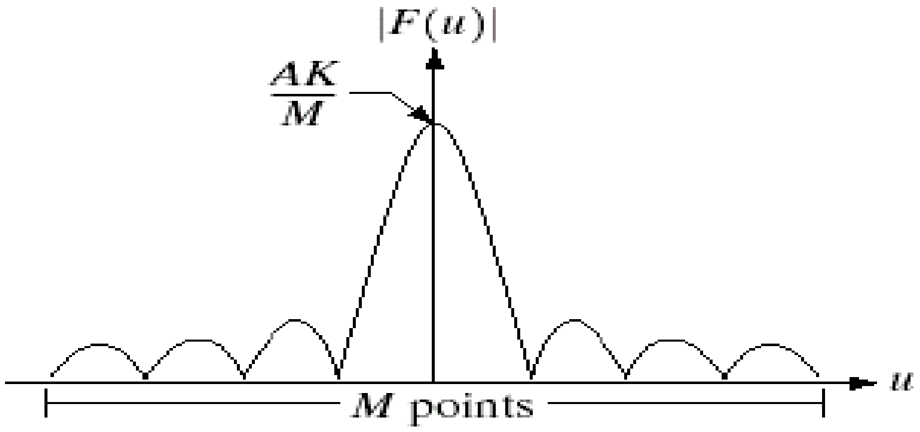


M= 1024, A=1, K= 16

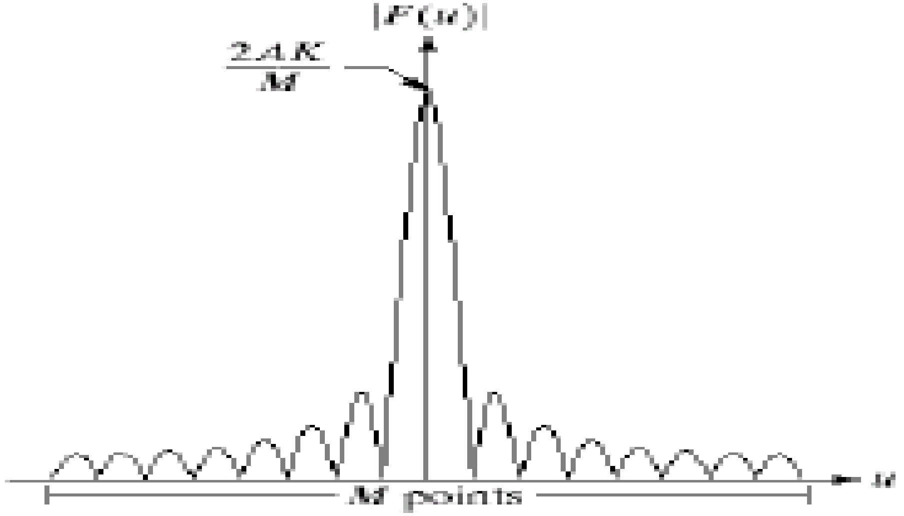


1-D Discrete Fourier Transform

- **Example:**



M= 1024, A=1, K= 8



M= 1024, A=1, K= 16

- Spectrum is centered at *u* =0, this can be accomplish by multiplying *f(x) by* $(-1)^x$ before taking transform.
- Height for K = 16 is twice of K = 8, and area under curve the curve in x - domain doubled.
- Number of zeros in the spectrum in the same interval doubled as the length function doubled.

Two Dimensional
Discrete Fourier Transform

Relationship between spatial and frequency intervals

- Δx, Δu are related as $$\Delta u = \frac{1}{M\Delta x}$$

- This relationship is useful when measurement are an issue in the image being processed.

2-D Discrete Fourier Transform

- **2D DFT and its Inverse :**

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

For $u = 0,1,2..., M\text{-}1,$ and $v = 0,1,2, ..., N\text{-}1,$ transform *or* frequency variable.

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

For $x = 0,1,2..., M\text{-}1,$ and $y = 0,1,2, ..., N\text{-}1$, spatial or image variable.

2-D Discrete Fourier Transform

- **Spectrum, Phase angle and Spectrum density:**

$$F(u,v) = |F(u,v)| e^{-j\phi(u,v)} \qquad |F(u,v)| = \left[R^2(u,v) + I^2(u,v) \right]^{1/2}$$

$R(u,v)$: Real part of **$F(u,v)$**, **$I(u,v)$:** Imaginary part of **$F(u,v)$**, **$|F(u,v)|$:** Magnitude spectrum of the Fourier Transform.

$$\phi(u,v) = \tan^{-1}\left[I(u,v)/R(u,v) \right]$$

: phase angle or phase spectrum

$$P(u,v) = |F(u,v)|^2$$

: power spectrum / power spectral density

$$= R^2(u,v) + I^2(u,v)$$

Suggested Readings

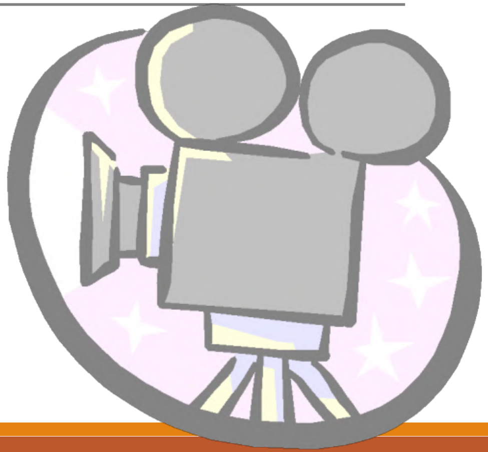
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Enhancement in the Frequency Domain
    - Relationship between sampling and frequency intervals
    - Properties of Fourier Transformation

Relationship between spatial sampling and frequency intervals

If $f(x)$ consists of M samples of a function $f(t)$ taken ΔT units apart, the duration of the record comprising the set $\{f(x)\}$, $x = 0, 1, 2, \ldots, M-1$, is

$$T = M\Delta T$$

The corresponding spacing, Δu, in the discrete frequency domain follows

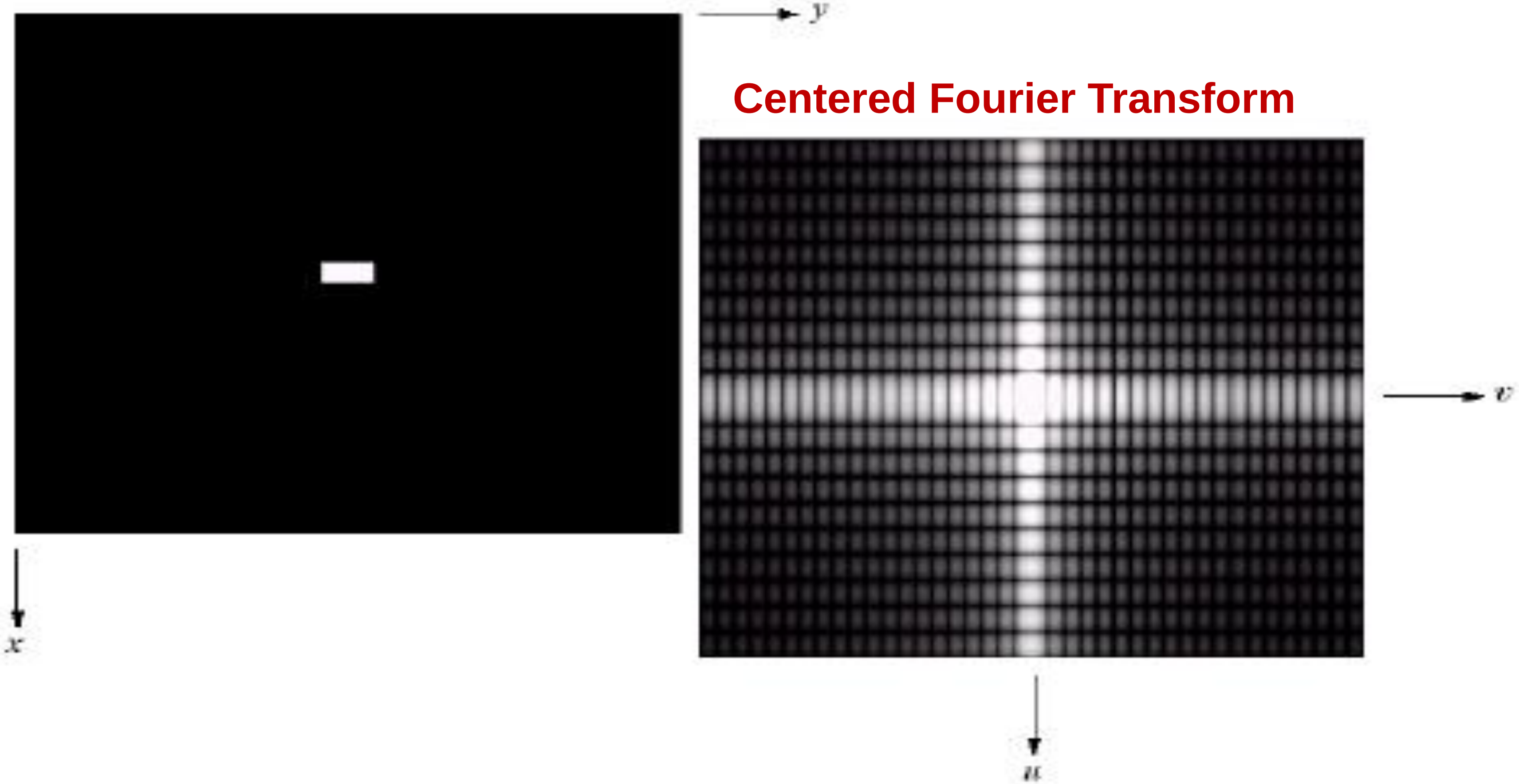
$$\Delta u = \frac{1}{M\Delta T} = \frac{1}{T}$$

The entire frequency range spanned by the M components of the DFT is

$$\Omega = M\Delta u = \frac{1}{\Delta T}$$

2-D Discrete Fourier Transform

Association Between Frequency Domain and Spatial Domain:



a b

FIGURE 4.3 (a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

Properties of Fourier Transform

- Relationships between Spatial and Frequency Intervals
- Translation and Rotation
- Periodicity
- Symmetry Properties

Relationships between Spatial and Frequency Intervals

Suppose that a continuous function f(t, z) is sampled to form a digital image, f(x, y) consisting M * N samples Taken in t and z directions respectively.

Let ΔT and ΔZ denote the separation between samples then the separation between the corresponding discrete Frequency domain variables are given by

$$\Delta u = \frac{1}{M\Delta T}$$

$$\Delta v = \frac{1}{N\Delta Z}$$

Translation and Rotation

Multiplying f(x y) by the exponential shifts the origin of the DFT to (u_0, v_0)

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

$$f(x-x_0, y-y_0) \Leftrightarrow F(u, v)e^{-j2\pi(x_0u/M+y_0v/N)}$$

Rotation

Using polar coordinates

$$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$$

Results in the following transform pair:

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$

Which indicates that rotating f(x, y) by angle θ_0 rotates F(u, v) by the same angle.

Periodicity

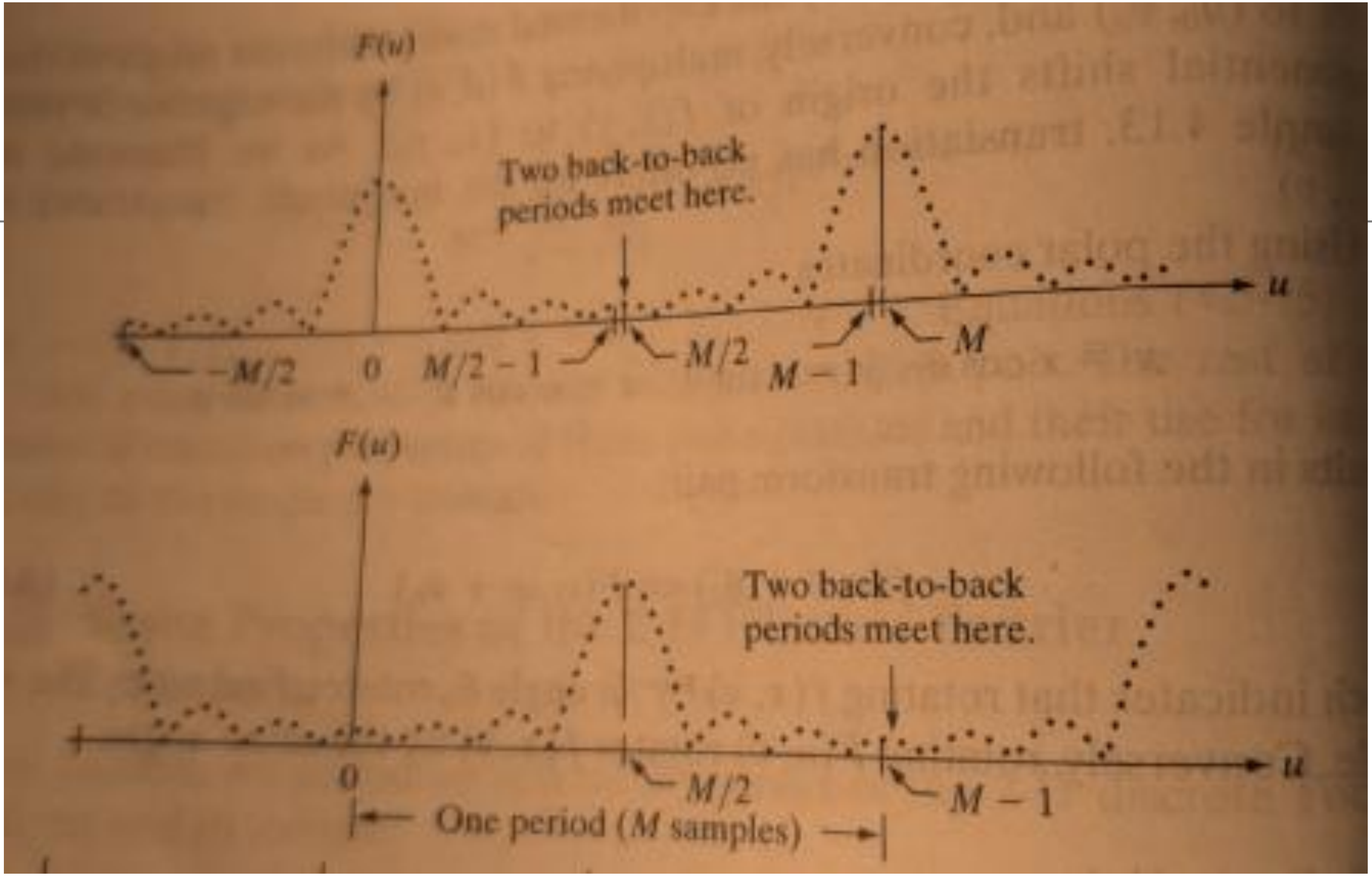
As in the 1D case, the 2-D Fourier Transform and its inverse are infinitely periodic in u and v direction.

$$F(u, v) = F(u + k_1 M, v) = F(u, v + k_2 N) = F(u + k_1 M, v + k_2 N)$$

and

$$f(x, y) = f(x + k_1 M, y) = f(x, y + k_2 N) = f(x + k_1 M, y + k_2 N)$$

K_1 and k_2 are integers.



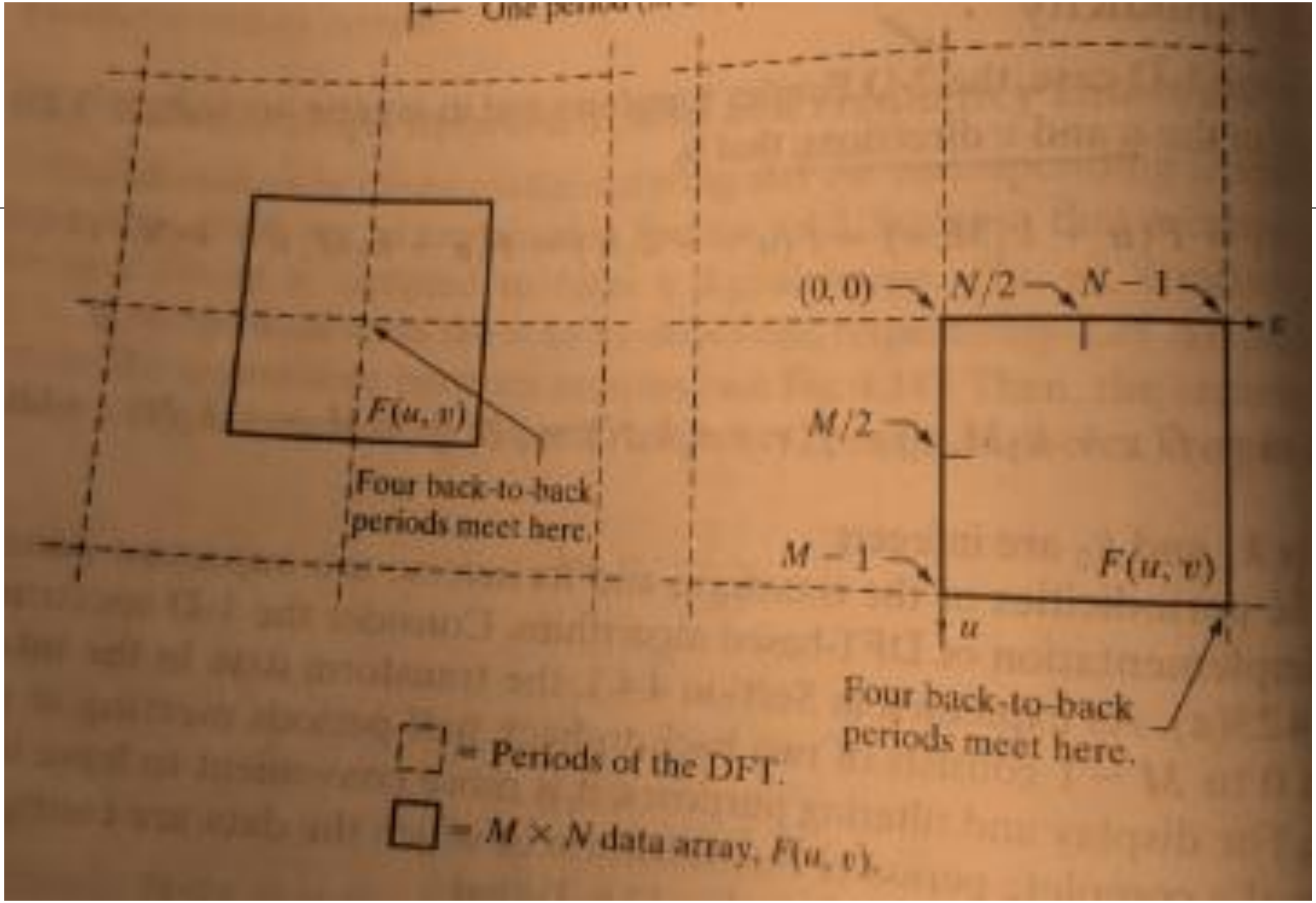
F(u)
Two back-to-back periods meet here.
−M/2
0
M/2 − 1
M/2
M − 1
M
u
F(u)
Two back-to-back periods meet here.
0
M/2
M − 1
u
One period (M samples)

$$f(x)e^{j2\pi(u_0x/M)} \Leftrightarrow F(u - u_0)$$

Let u_0=M/2 the exponential becomes $e^{j\pi x}= (-1)^x$

$$f(x)(-1)^x \Leftrightarrow F(u - M/2)$$

Multiplying f(x) by (-1)x shifts the data so that F(0) is at the center of the interval [0, M-1]



F(u, v)
Four back-to-back periods meet here.
(0, 0)
N/2
N − 1
M/2
M − 1
u
F(u, v)
Four back-to-back periods meet here.
= Periods of the DFT.
= M × N data array, F(u, v).

2-D Discrete Fourier Transform

- **Shift the origin *F*(*u*,*v*) to (*M/2, N/2*) :**

$$FT\left\{ f(x,y)(-1)^{x+y} \right\} = F\left(u - \frac{M}{2}, v - \frac{N}{2} \right)$$

- Multiplying $f(x,y)$ by $(-1)^{x+y}$ shifts the origin of $F(u, v)$, to frequency coordinate $(M/2, N/2)$.
- u, v are integer, so shifted coordinates must be an integer. This requires M and N are even number.

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

2-D Discrete Fourier Transform

- If $f(x,y)$ is real and symmetric, its Fourier transform is conjugate symmetric:

$$F(u,v) = F^*(-u,-v)$$

$$|F(u,v)| = |F(-u,-v)|$$

- Relationship between samples in the spatial domain and frequency domain

$$\Delta u = \frac{1}{M\Delta x} \qquad \Delta v = \frac{1}{N\Delta y}$$

Symmetry Property

Any real or complex function can be expressed as

$$w(x, y) = w_e(x, y) + w_o(x, y) \quad \text{.....................1}$$

where the even and odd parts are defined as

$$w_e(x, y) \triangleq \frac{w(x, y) + w(-x, -y)}{2} \quad \text{.....................1(a)}$$

and

$$w_o(x, y) \triangleq \frac{w(x, y) - w(-x, -y)}{2} \quad \text{.....................1(b)}$$

Sustituting 1(a) and 1(b) in equation 1 gives the identity w(x, y)=w(x, y)

As,

$$w_e(x, y) = w_e(-x, -y)$$

and that

$$w_o(x, y) = -w_o(-x, -y)$$

Even functions are said to be symmetric and odd functions are antisymmetric. It is convenient to think only in terms of nonnegative indices in which case the definition of evenness and oddness become:

$$w_e(x, y) = w_e(M - x, N - y)$$

and

$$w_o(x, y) = -w_o(M - x, N - y)$$

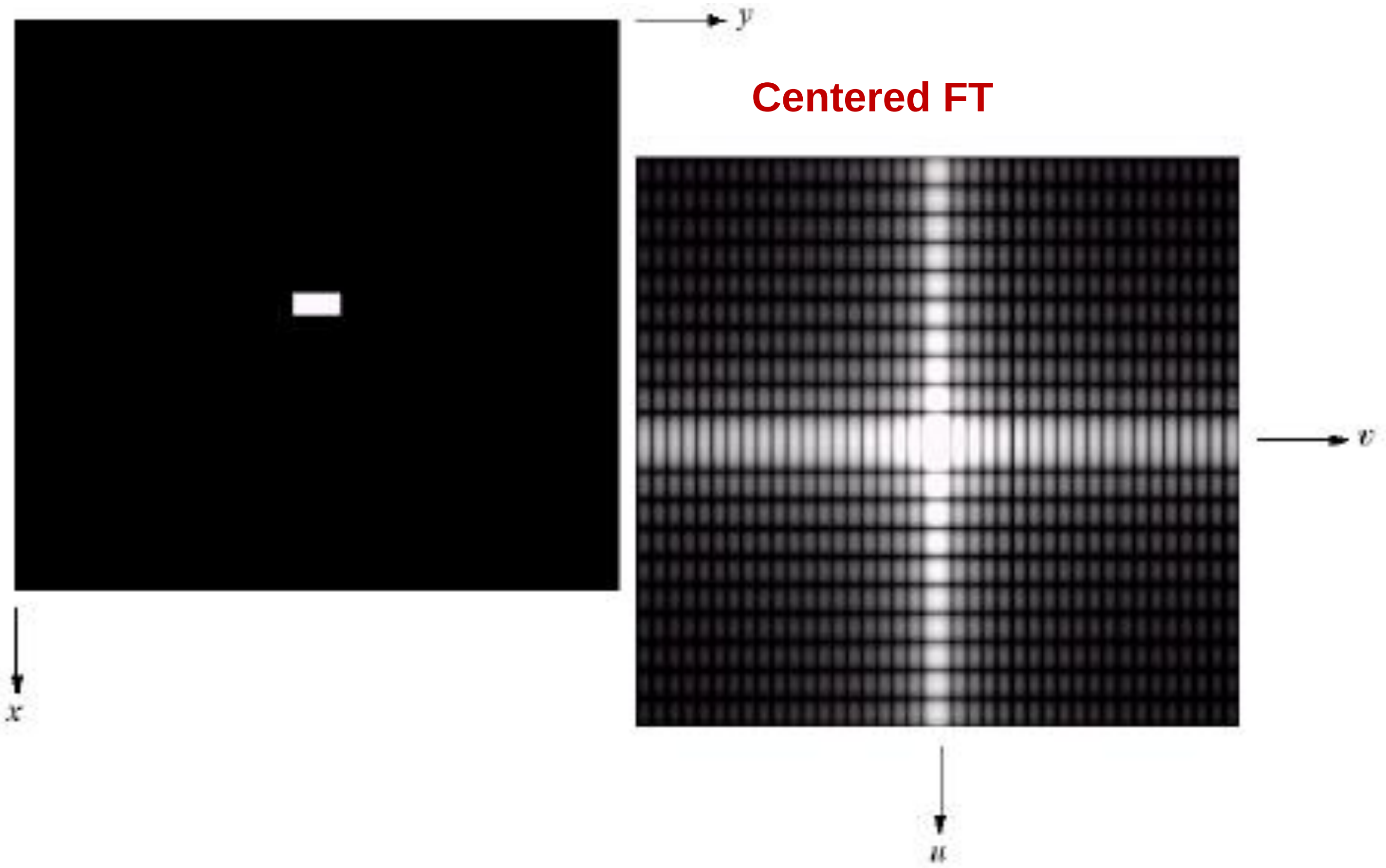
where, as usual, M and N are the number of rows and columns of a 2-D array.

We know from elementary mathematical analysis that the product of two even or two odd functions is even, and that the product of an even and an odd function is odd. In addition, the only way that a discrete function can be odd is if all its samples sum to zero. These properties lead to the important result that

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} w_e(x, y)\, w_o(x, y) = 0$$

2-D Discrete Fourier Transform

Example:



a b

FIGURE 4.3 (a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

2-D Discrete Fourier Transform

Frequency component of FT and Spatial Characteristic of an Image:

- Frequency is directly related to rate of change of intensity in the image.
- F(0,0) is the average gray level of image.
- Around origin of the FT, the low frequency corresponds to the slow varying component.
- As move further away from origin, the higher frequency being correspond to faster and faster gray level changes in the image.

Suggested Readings

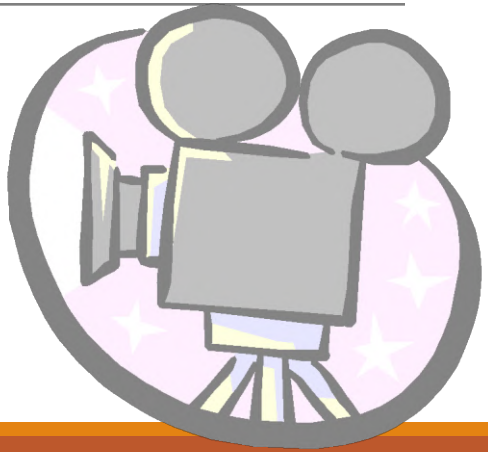
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Enhancement in the Frequency Domain
    - Smoothing Filters
        - Ideal
        - Butterworth
        - Gaussian

Basic steps for filtering in the frequency domain

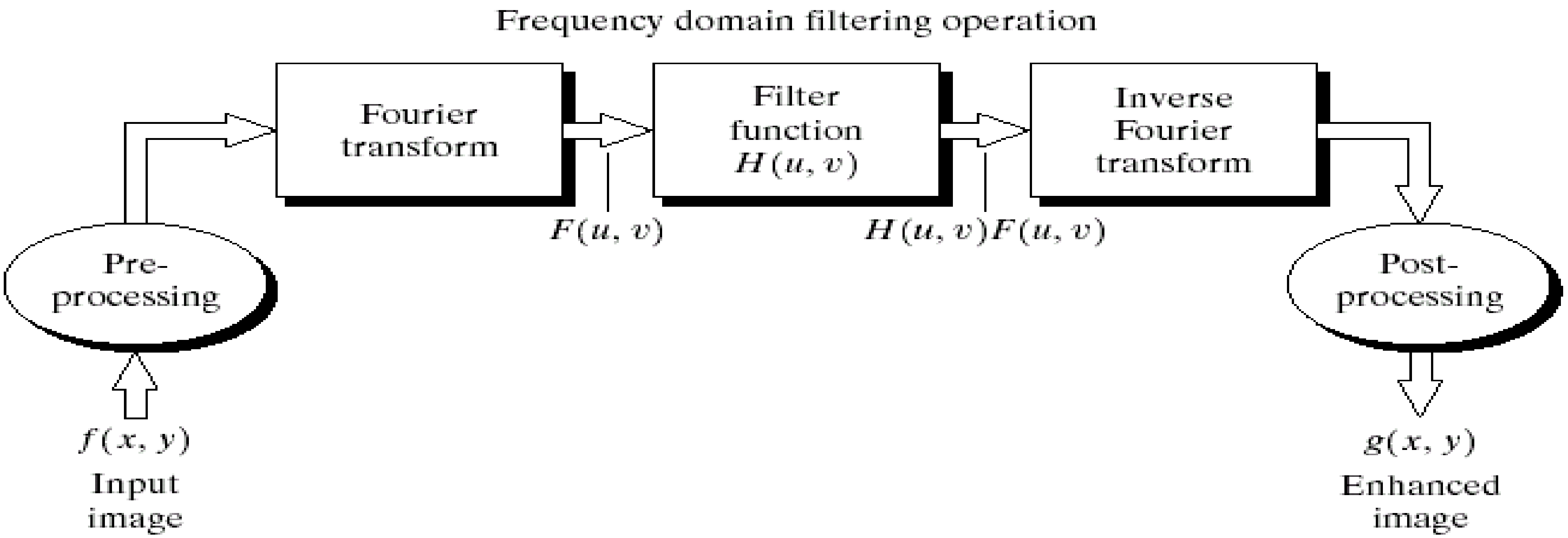


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Basics of filtering in the frequency domain

1. multiply the input image by $(-1)^{x+y}$ to center the transform to u = M/2 and v = N/2 (if M and N are even numbers, then the shifted coordinates will be integers)
2. compute F(u,v), the DFT of the image from (1)
3. multiply F(u,v) by a filter function H(u,v)
4. compute the inverse DFT of the result in (3)
5. obtain the real part of the result in (4)
6. multiply the result in (5) by $(-1)^{x+y}$ to cancel the multiplication of the input image.

Basics of filtering in the frequency domain

- **Low frequencies**

  - Smooth Areas

- **High Frequency**

  - Edge, Texture, noise

- **Low Pass Filter**

  - Pass signal of frequency around DC component(zero frequency)

- **High Pass Filter**

  - Stop signal of frequency around zero frequency.

- **Band Pass Filter**

  - Pass signal having frequency in given range.

Smoothing Frequency-domain filters

- **High Frequency Component in FT:**

edge, sharp transitions (such as noise)

Smoothing (blurring) is achieved in the frequency domain by attenuation of a specified range of high frequency components in the transform of a given image

- **Basic Filtering Model in Frequency Domain:**

$$G(u,v) = H(u,v)F(u,v)$$

$F(u,v)$: FT of image to be smoothed

$H(u,v)$: Filter Transfer Function

Smoothing Frequency-domain filters: Ideal Lowpass filter

- Cutoff all high frequency components of FT that are at a greater distance than a specified distance D_0 from the origin of the transform.

$$H(u,v) = \begin{cases} 1, & if \ D(u,v) \le D_0 \\ 0, & if \ D(u,v) > D_0 \end{cases}$$

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

Centre of frequency rectangle (M/2, N/2) is origin.

$$D(u,v) = \left[(u - M/2)^2 + (v - N/2)^2 \right]^{1/2}$$

Ideal filter: All frequencies inside the circle of radius D_0 are passed with no attenuation, while all frequencies outside of circle completely attenuated.

The point of transition between H(u,v)=1 and H(u,v)=0 is called the **cutoff frequency.**

Smoothing Frequency-domain filters: Ideal Lowpass filter

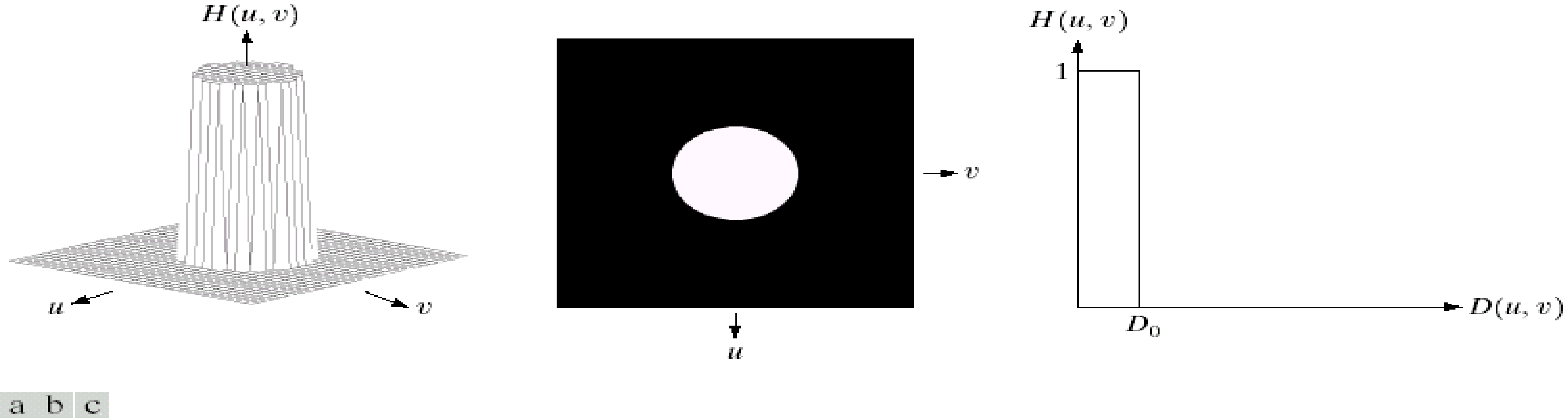


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Smoothing Frequency-domain filters: Ideal Lowpass filter

- ILPF is compared by studying the behavior as a function of the same cutoff frequency.

- Total power:

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u,v)$$

Where, $P(u, v) = |F(u, v)|^2$

- A circle of radius *r* with origin at the centre of the frequency rectangle encloses the percentage power:

$$\alpha = 100 \left[\sum_{u} \sum_{v} P(u,v) \middle/ P_T \right]$$

- Summation is taken over the values of (*u*,*v*) that lie inside the circle or on its boundary.

image power circles

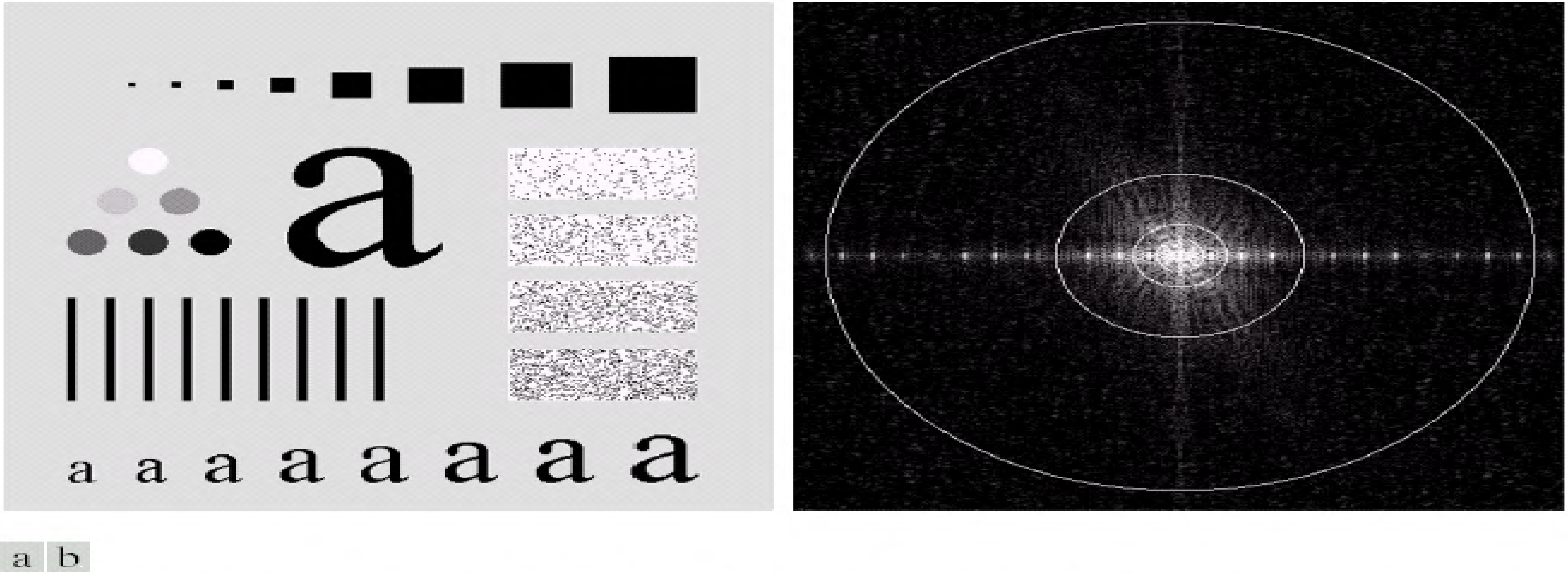


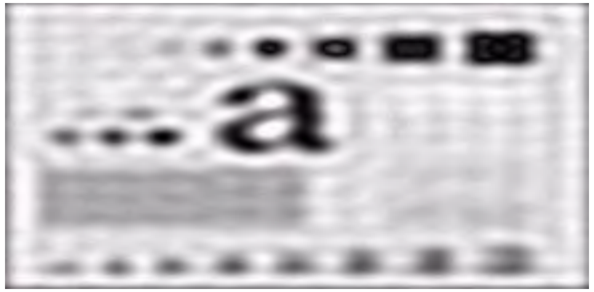
FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Result of ILPF

a	b	c
d	e	f

Original

Cutoff Freq. at radii
5, power removed 8%



Cutoff Freq. at radii
15, power removed 5.4%

Cutoff Freq. at radii
30, power removed 3.6%

Cutoff Freq. at radii
80, power removed 2%

Cutoff Freq. at radii
230, power removed 0.5%

Result of ILPF

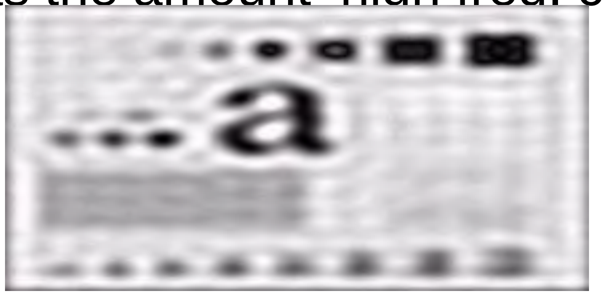
c	d	e

- (b) is useless for all practical purposes, unless objective of blurring is eliminate all fine details, except the blobs representing the largest object.

- Severe blurring in (b) indicate that most of the sharp information detail is contained in 8% of the power removed by the filter.

- As the filter radius is increases, less and less power is removed, resulting less severe blurring.

- (c) –(e) have ringing effect , even though 2% of the total power was removed, it becomes finer in texture as the amount high freq. comp. increases.



Cutoff Freq. at radii 15, power removed 5.4%

Cutoff Freq. at radii 30, power removed 3.6%

Cutoff Freq. at radii 80, power removed 2%

Result of ILPF

c	d	e

- Ringing behavior is characteristic of the ideal filter.
- 99.5% case shows very slight blurring in the noisy squares, but in most part, it is very close to original. Little edge information is contained in upper 0.5% of spectrum.

Original

Cutoff Freq. at radii 230, power removed 0.5%

- Ideal filter is not very practical, it can not be implemented in H/w, however it can be implemented on computer, and use for developing filtering concept.

Blurring and Ringing by ILPF

- Ringing behavior is characteristic of the ideal low pass filter.
- Filtering process is related to convolution process as:

$$G(u,v) = H(u,v)F(u,v)$$

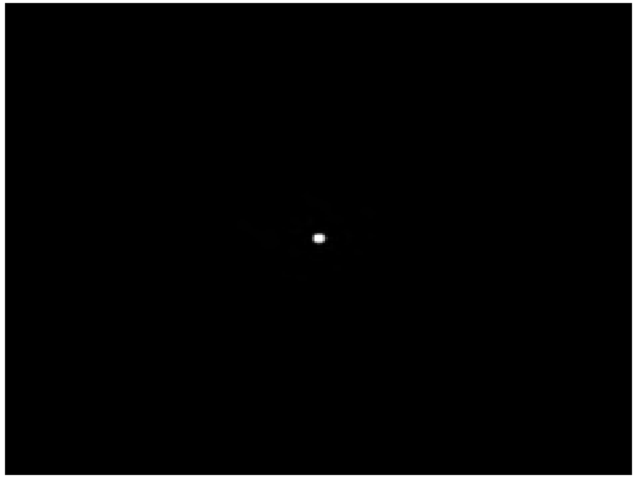
- $H(u,v)$ is filter function and F and G are FT of original image $f(x,y)$ and filtered / blurred image $g(x,y)$.
- Equivalent processes in spatial domain is convolution:

$$g(x,y) = h(x,y) \otimes f(x,y)$$

$h(x,y)$: inverse FT of filter transfer function $H(u,v)$.

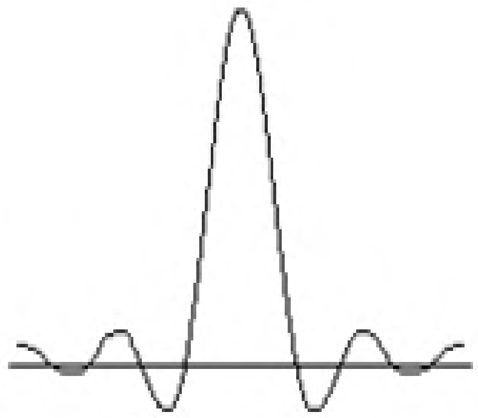
Example

a	b	c



FD ILPF of radius 5

Spatial filter corresponding (a)



Horizontal central scan of (b)

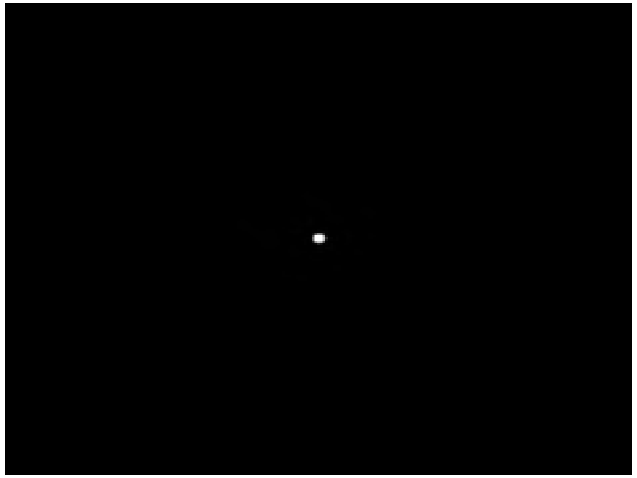
$h(x,y)$ has two component :

i. Dominant central at the origin : Responsible for blurring

ii. Concentric, circular components about the center component: Responsible for ringing

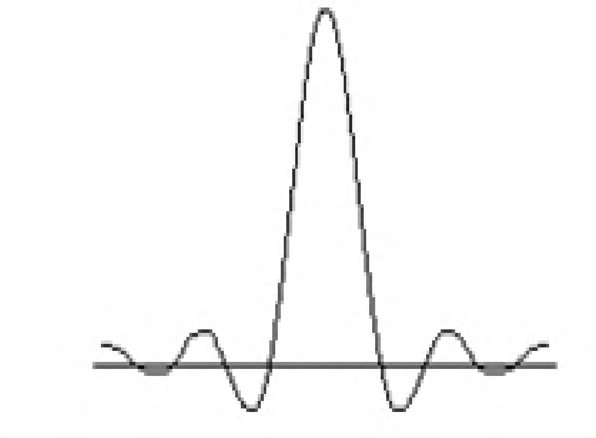
Example

a	b	c



FD ILPF of radius 5

Spatial filter corresponding (a)



Horizontal central scan of (b)

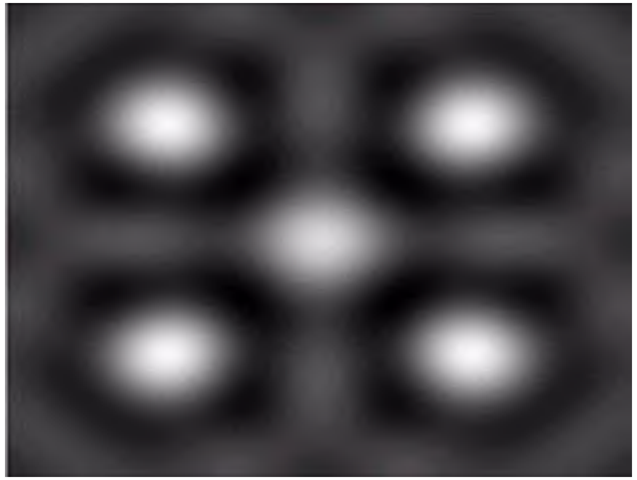
- Radius of central component is inversely proportional to the cutoff frequency of ideal filter.
- Number of circle per unit distance from the origin is inversely proportional to the cutoff frequency of ideal filter.

Example

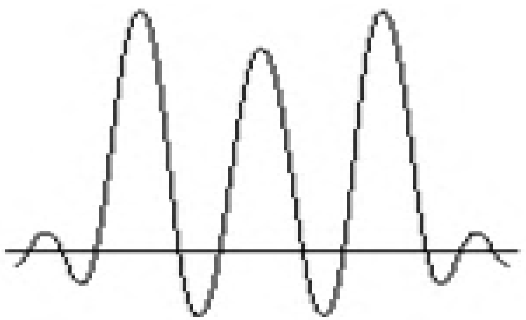
a	b	c

5 impulse in spatial domain
simulating the values of 5 pixel

Spatial filter



Convolution of (a) with (b)



Diagonal scan line of (c)

Example

- Thus reciprocal nature of $H(u,v)$ and $h(x,y)$ with convolution is responsible for blurring and ringing.

Narrow Filter In Frequency Domain $\Rightarrow$ More severe blurring and ringing

Blurring : Low frequencies are removed

Ringing : Cutoff is too sharp

Objective is to achieve blurring with little or no ringing

Butterworth Lowpass Filter: BLPF

$$H(u,v) = \frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

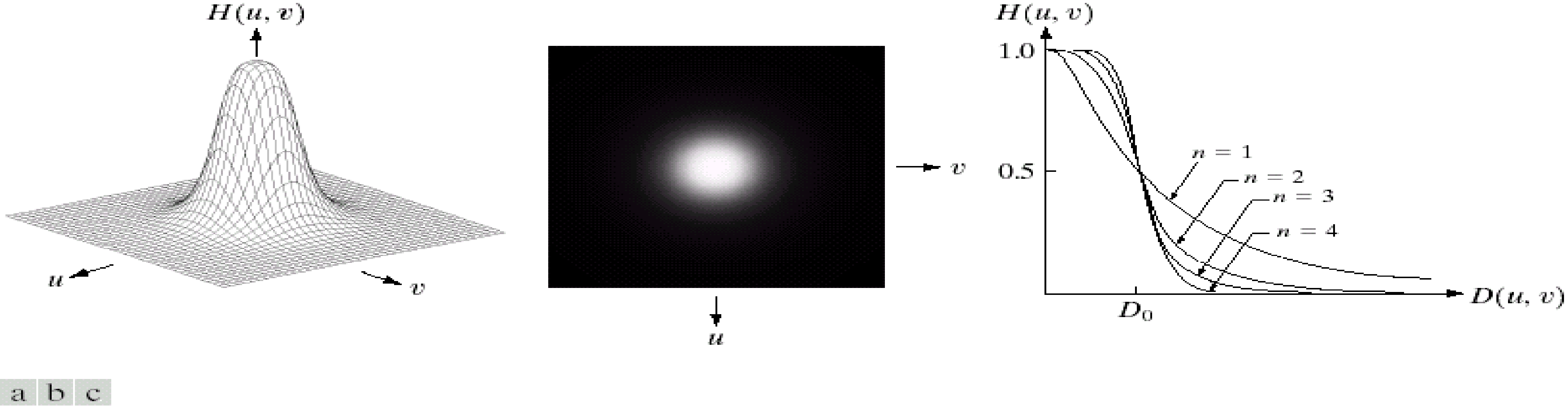


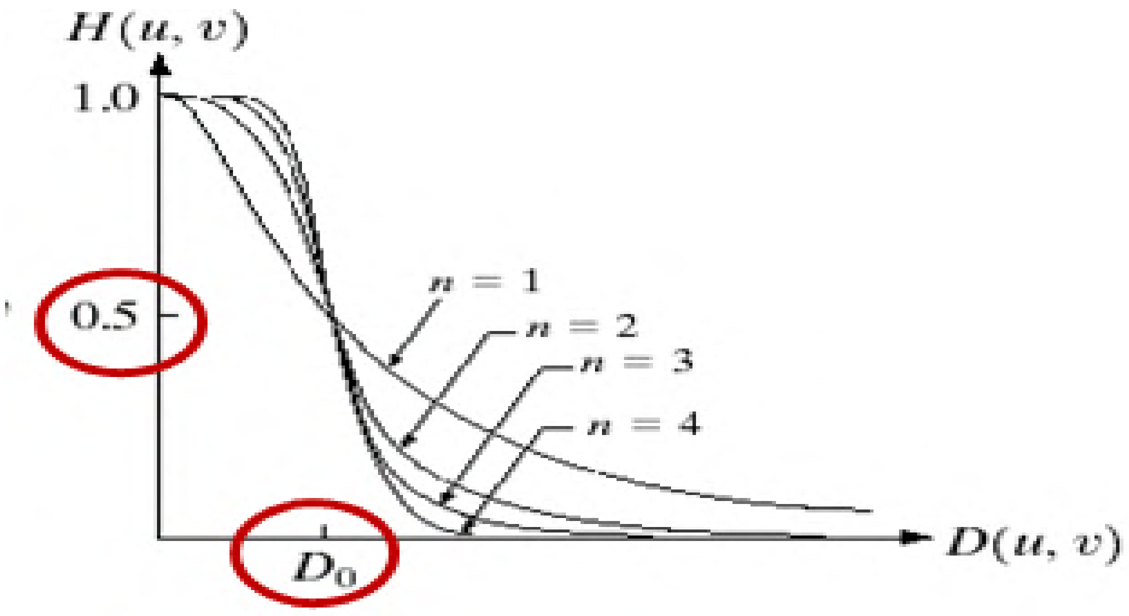
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth Lowpass Filter: BLPF

- Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies.

- For smooth transfer function, cutoff frequency locus at points for which $H(u,v)$ is down to a certain fraction of its maximum value is customary.

- $H(u,v)$ = 0.5 when $D(u,v) = D_0$



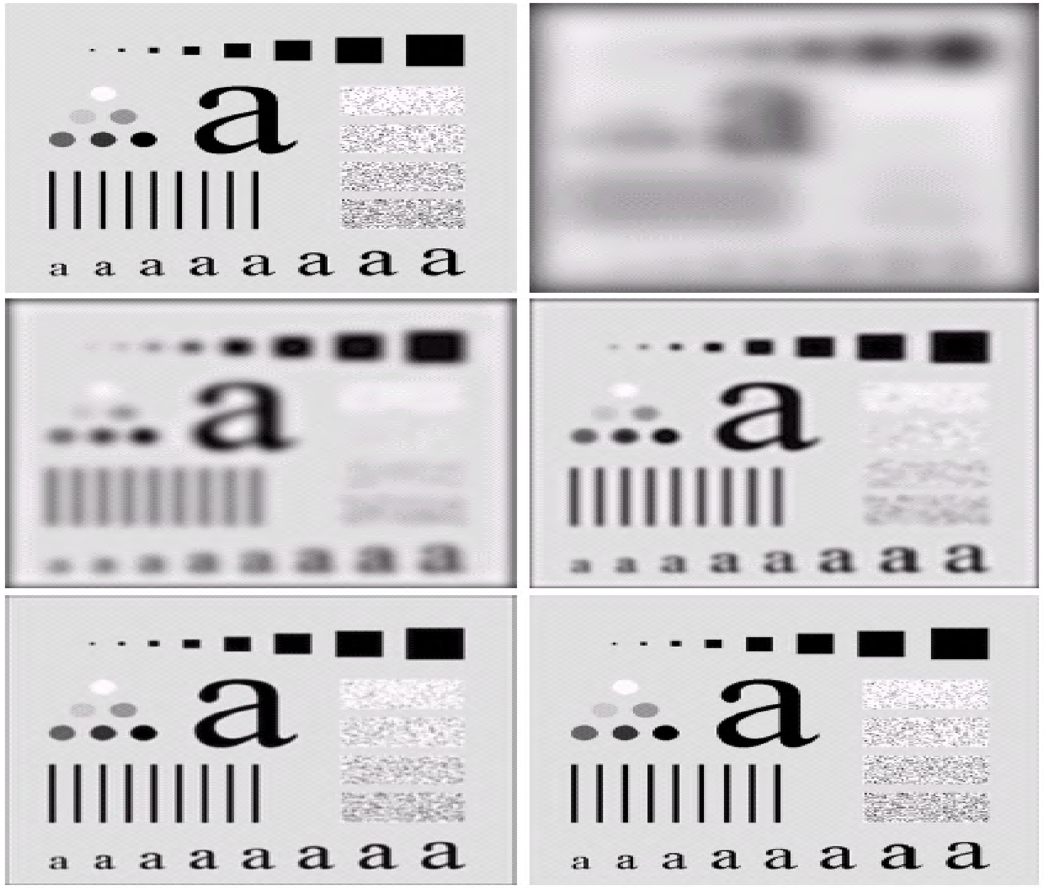
Example

ILPF

a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

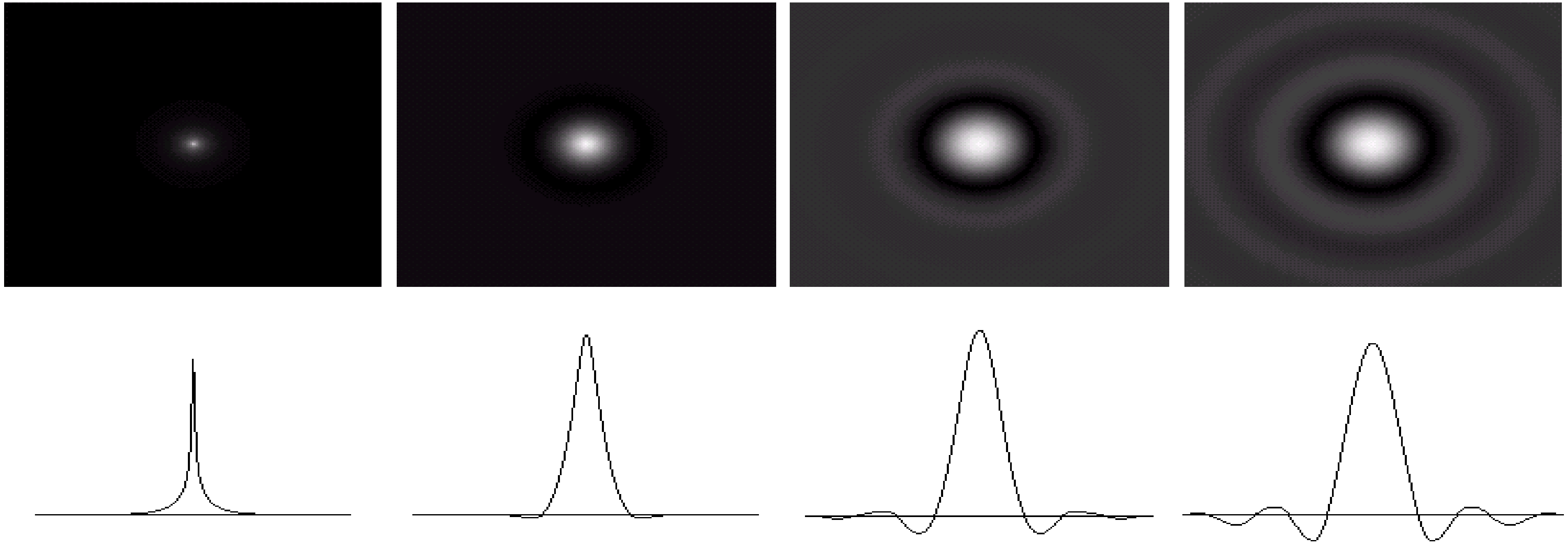
BLPF



a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Spatial representation of BLPFs



a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Spatial representation of BLPFs

- BLPF of order 1 has neither ringing nor negative value.
- Order 2 have mild ringing and small negative value but less than ILPF.
- Ringing in BLPF becomes significant for higher – order filters.
- BLPF of order 20 already have characteristics of the ILPF.

a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Gaussian Lowpass Filter: GLPF

- Filter transfer function: $$H(u,v) = e^{-D^2(u,v)/2\sigma^2};$$

σ : Measure of spread of Gaussian curve

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

- Taking $\sigma = D_0$ cutoff frequency: $$H(u,v) = e^{-D^2(u,v)/2D_0^2};$$

- Inverse FT of Gaussian low pass filter is Gaussian. Spatial GLPF will have no ringing.

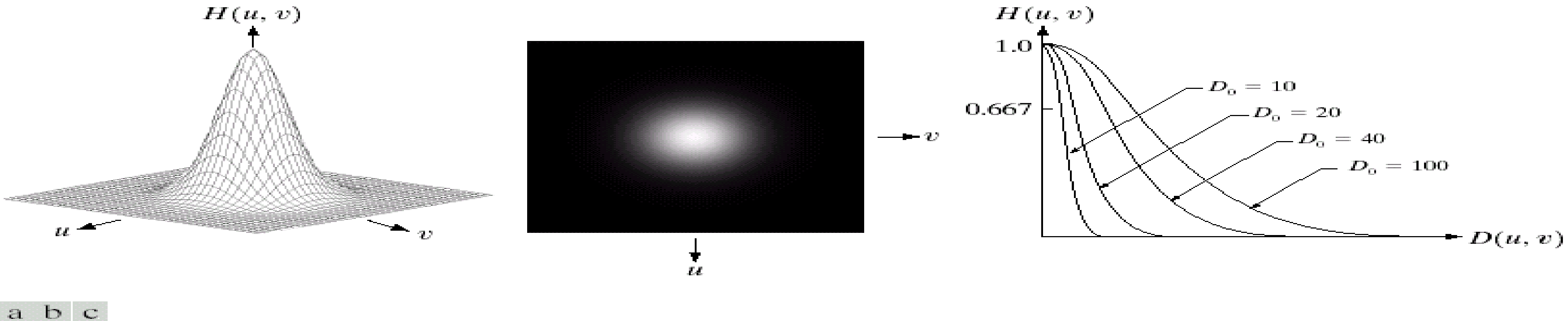


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Example

ILPF | **BLPF** | **GLPF**

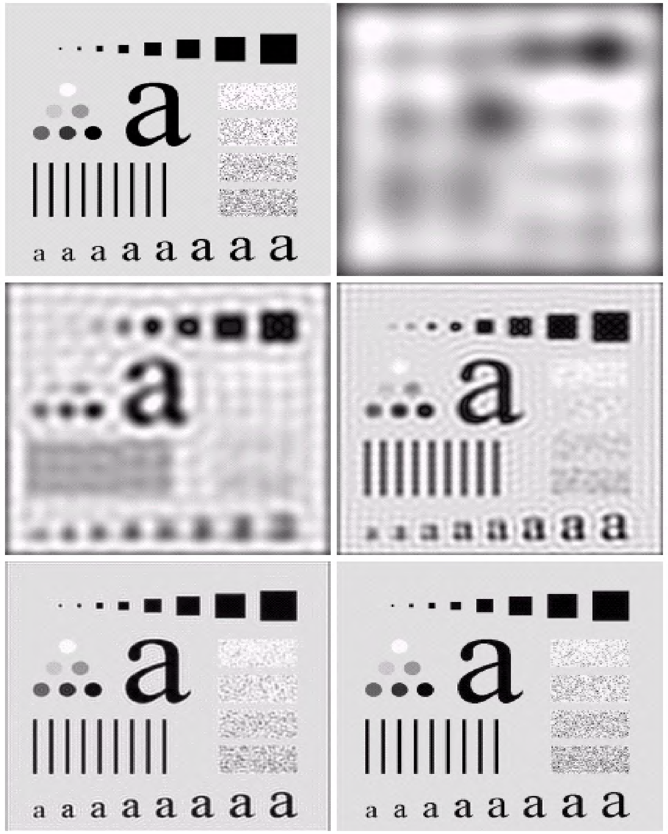


FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

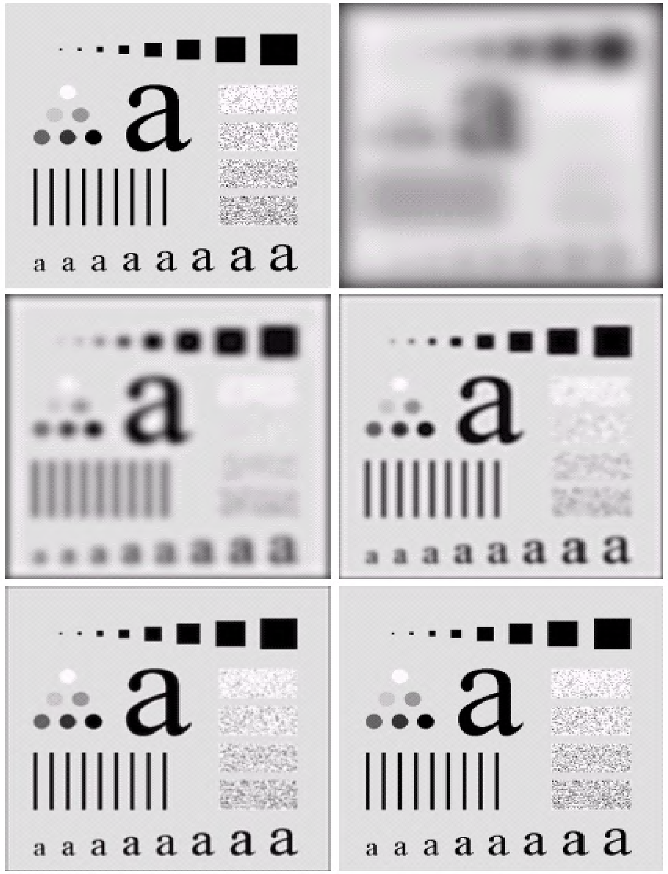


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

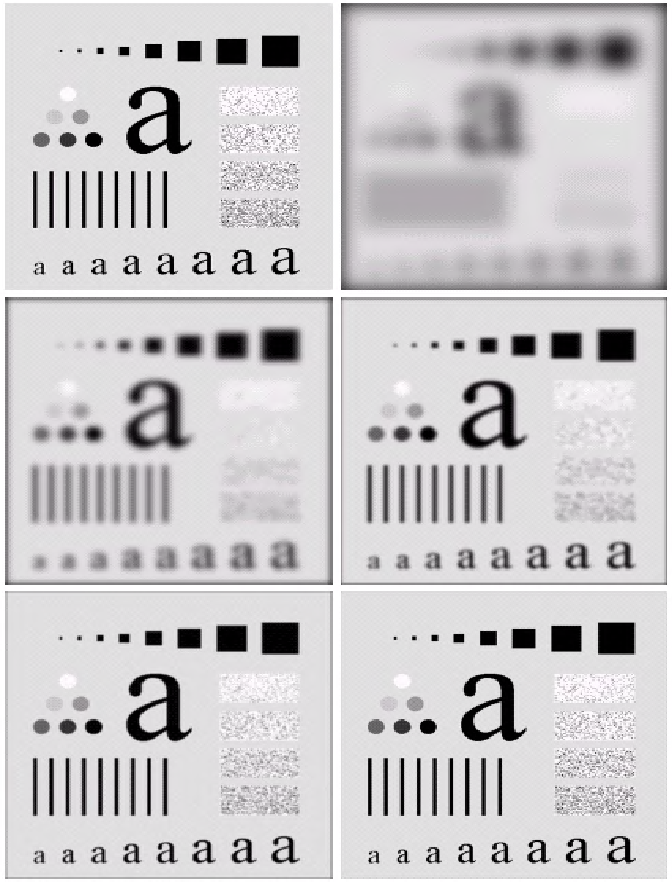
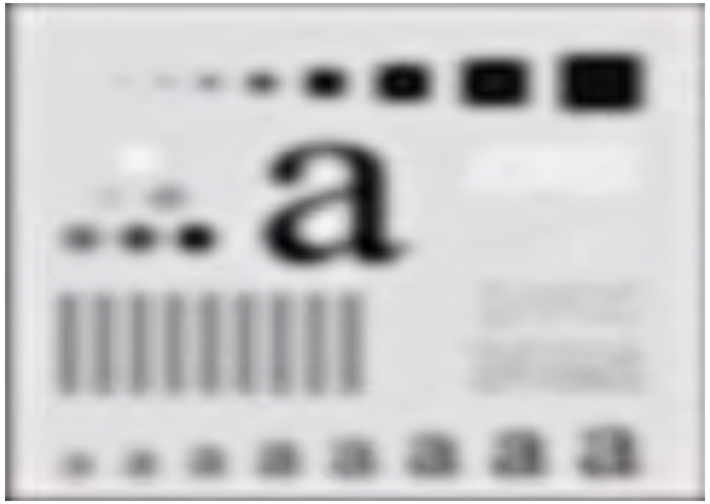


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

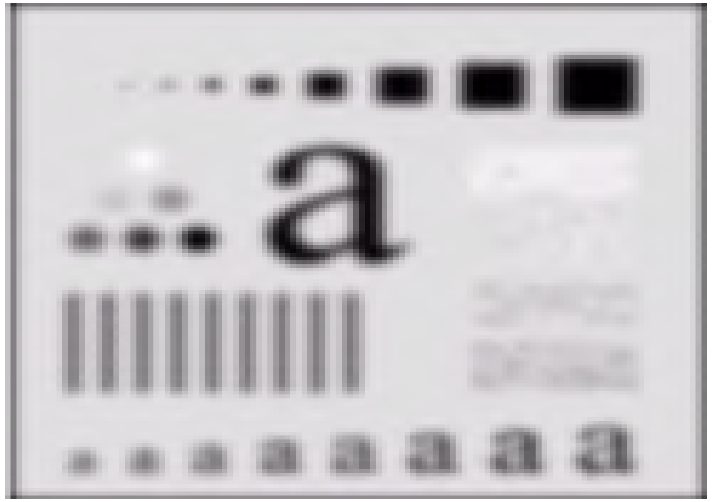
BLPF vs. GLPF

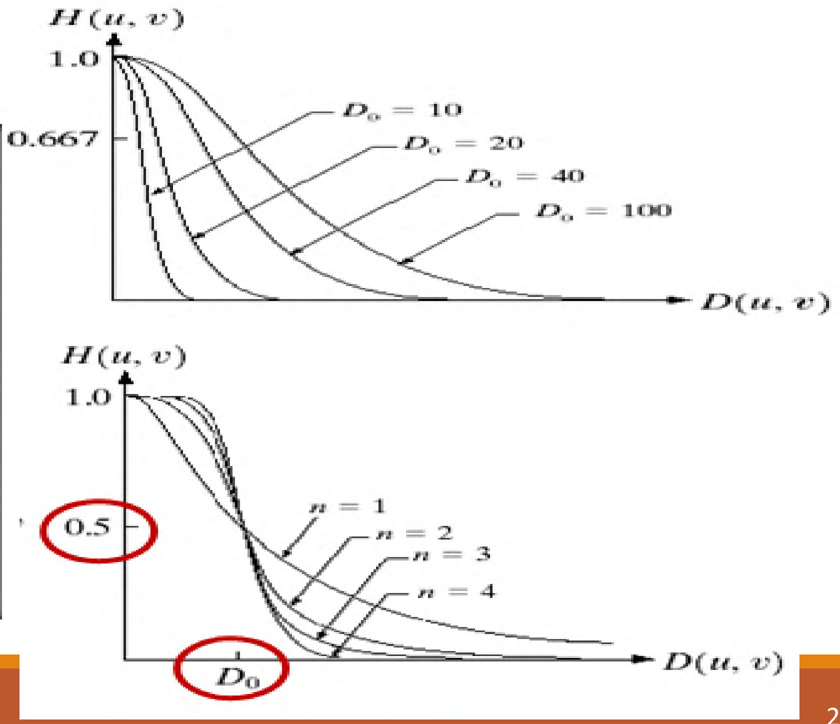
- In case of BLPF smooth transition in blurring as a function of increasing cutoff frequency.
- The GLPF does did not achieve as much smoothing as the BLPF of order 2 of same cutoff frequency.

BLPF (c)



GLPF (c)



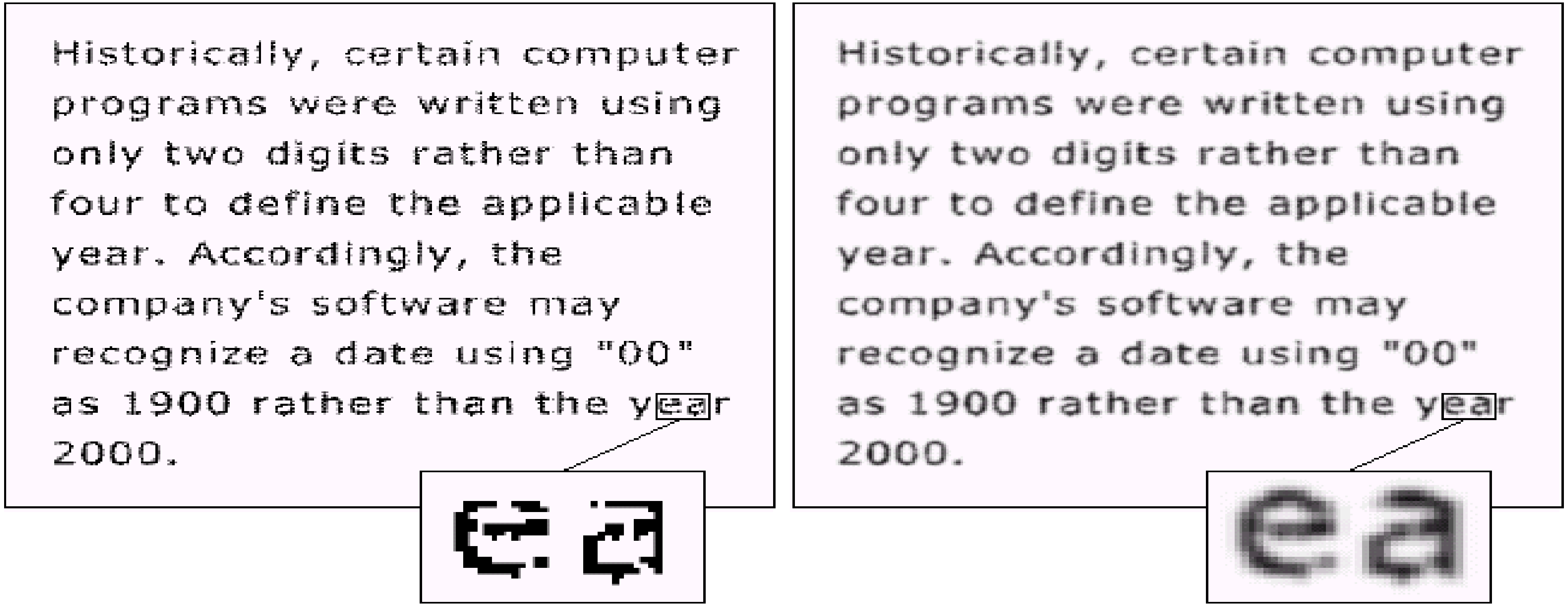


Example

Bridge small gaps in the input image by blurring

a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).



GLPF with D_0 =80

Example

Unsharp masking: Printing and Cosmetic industry

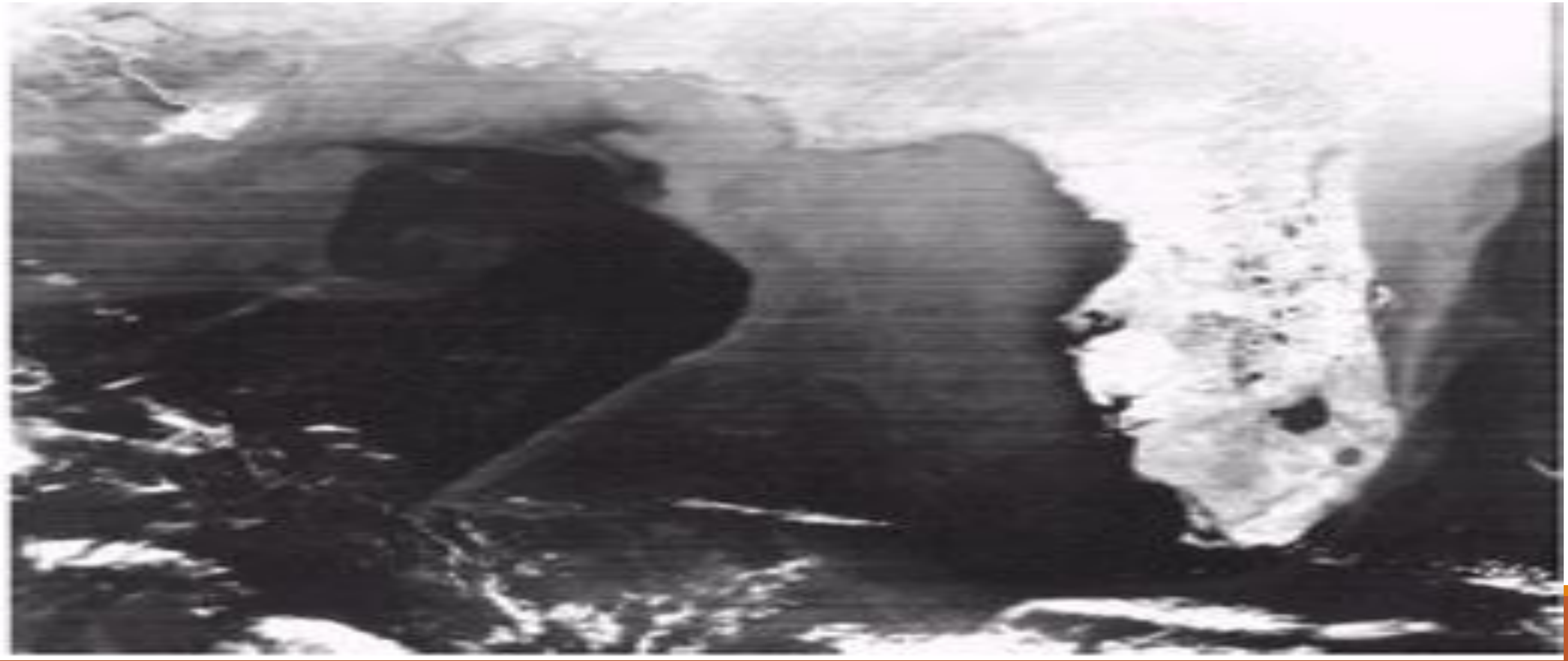
Produce a smoother, softer-looking result from a sharp original.

a b c

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

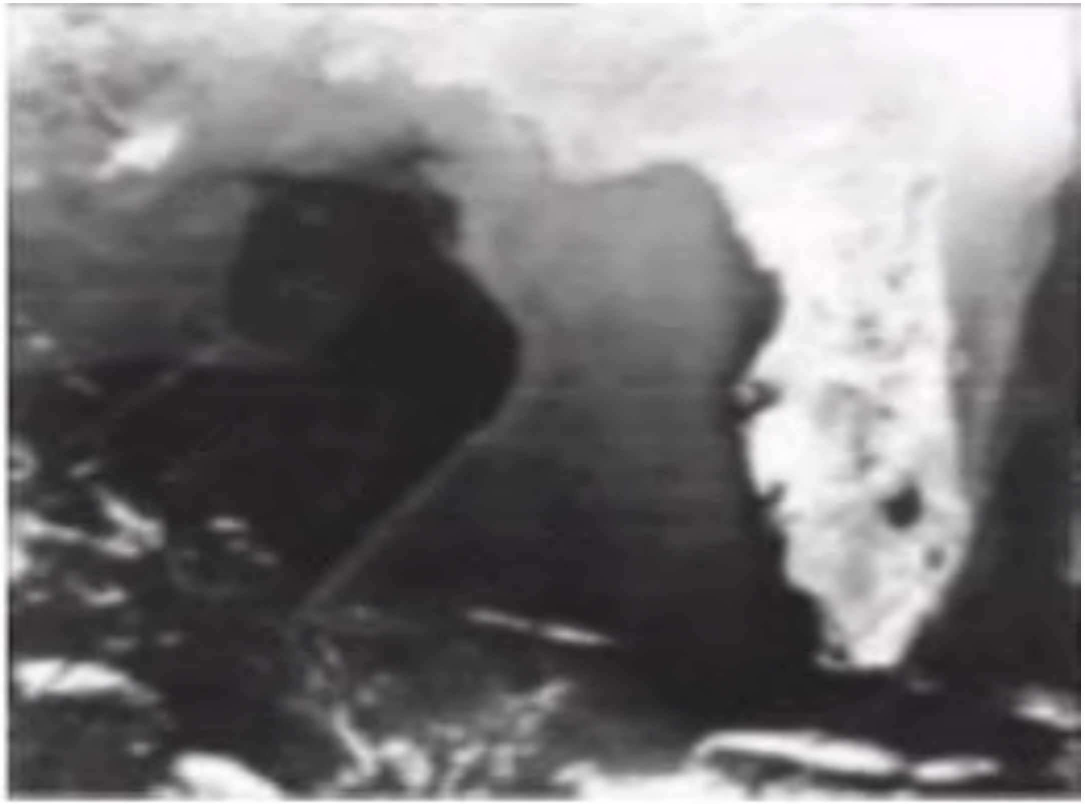
Example

- High Resolution Radiometer Image of gulf of Mexico (dark) and Florida (light)
- Prominent scan line along the direction in which scene is being scanned.



Example

- Low pass filter is crude but simple to reduce the effect of these scan line.



GLPF, $D_0 = 30$

GLPF, $D_0 = 10$

Suggested Readings

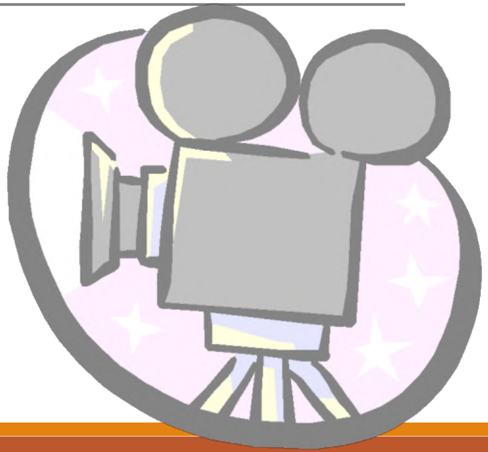
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

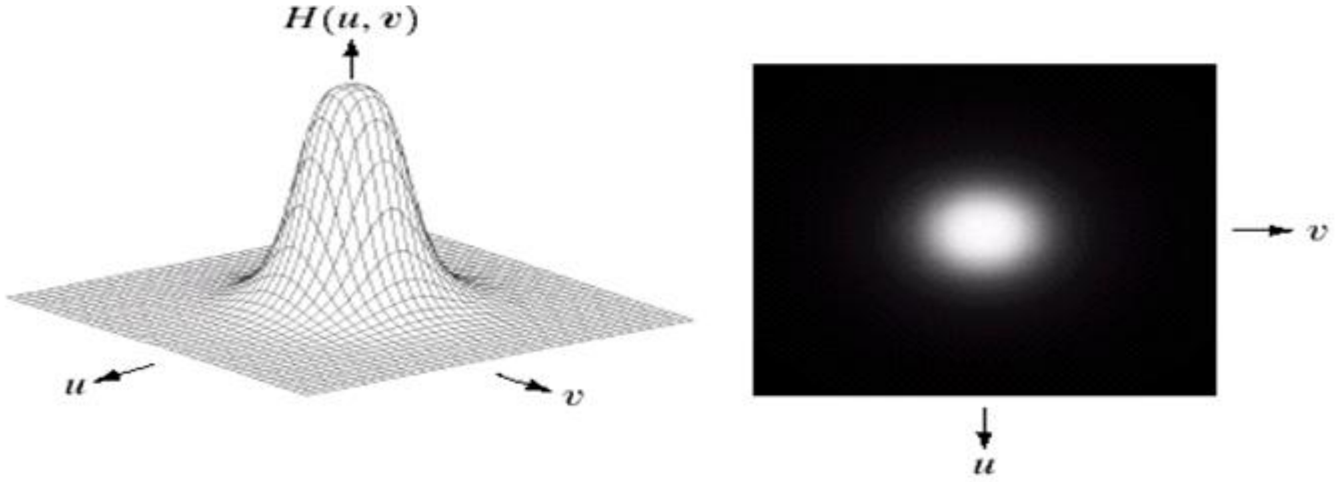
Image Processing

CS-317/CS-341



Outline

- Image Enhancement in the Frequency Domain
    - Smoothing Filters
        - Ideal
        - Butterworth
        - Gaussian
    - Sharpening Filters
        - Ideal
        - Butterworth
        - Gaussian



Smoothing Frequency-domain filters: Ideal Lowpass filter

- Cutoff all high frequency components of FT that are at a greater distance than a specified distance D_0 from the origin of the transform.

$$H(u,v) = \begin{cases} 1, & if \ D(u,v) \le D_0 \\ 0, & if \ D(u,v) > D_0 \end{cases}$$

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

Centre of frequency rectangle (M/2, N/2) is origin.

$$D(u,v) = \left[(u - M/2)^2 + (v - N/2)^2 \right]^{1/2}$$

Ideal filter: All frequencies inside the circle of radius D_0 are passed with no attenuation, while all frequencies outside of circle completely attenuated.

The point of transition between H(u,v)=1 and H(u,v)=0 is called the **cutoff frequency.**

Smoothing Frequency-domain filters: Ideal Lowpass filter

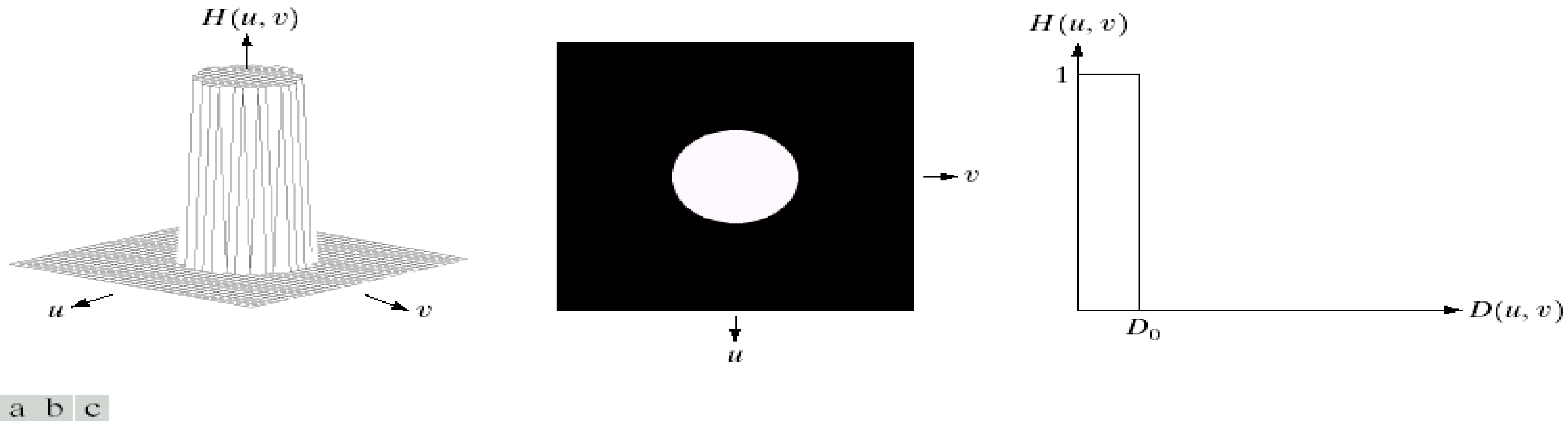


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Example

- Thus reciprocal nature of $H(u,v)$ and $h(x,y)$ with convolution is responsible for blurring and ringing.

Narrow Filter In Frequency Domain → More severe blurring and ringing

Blurring : Low frequencies are removed

Ringing : Cutoff is too sharp

Objective is to achieve blurring with little or no ringing

Butterworth Lowpass Filter: BLPF

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

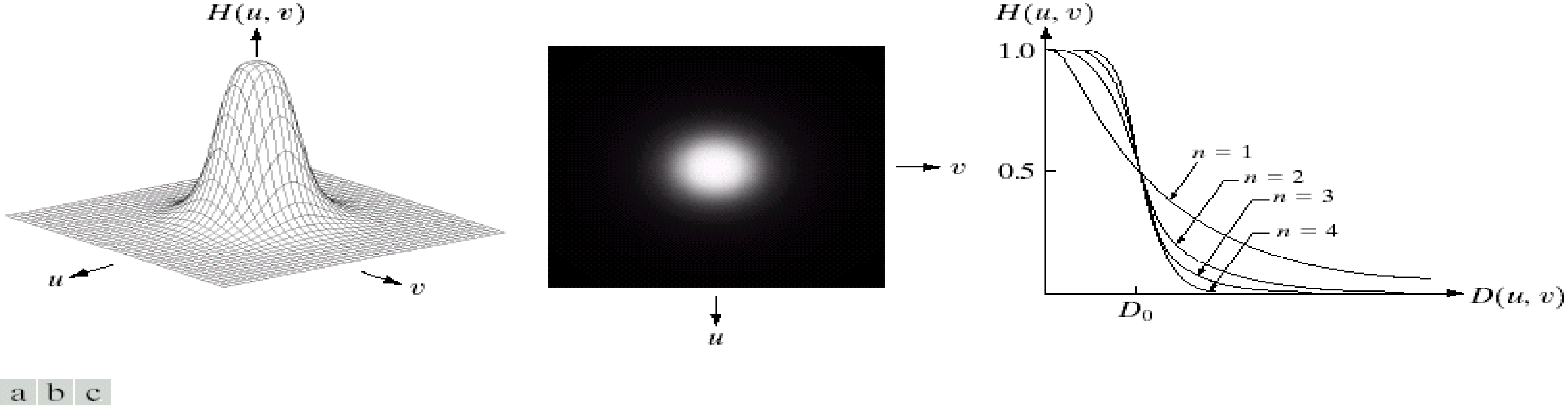


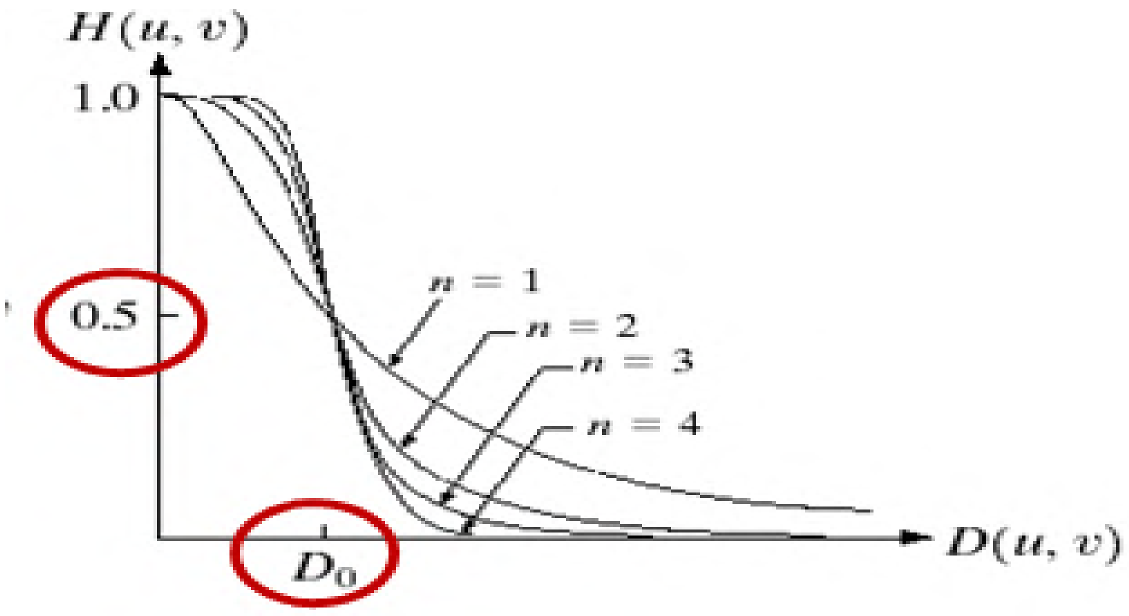
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth Lowpass Filter: BLPF

- Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies.

- For smooth transfer function, cutoff frequency locus at points for which $H(u,v)$ is down to a certain fraction of its maximum value is customary.

- $H(u,v) = 0.5$ when $D(u,v) = D_0$



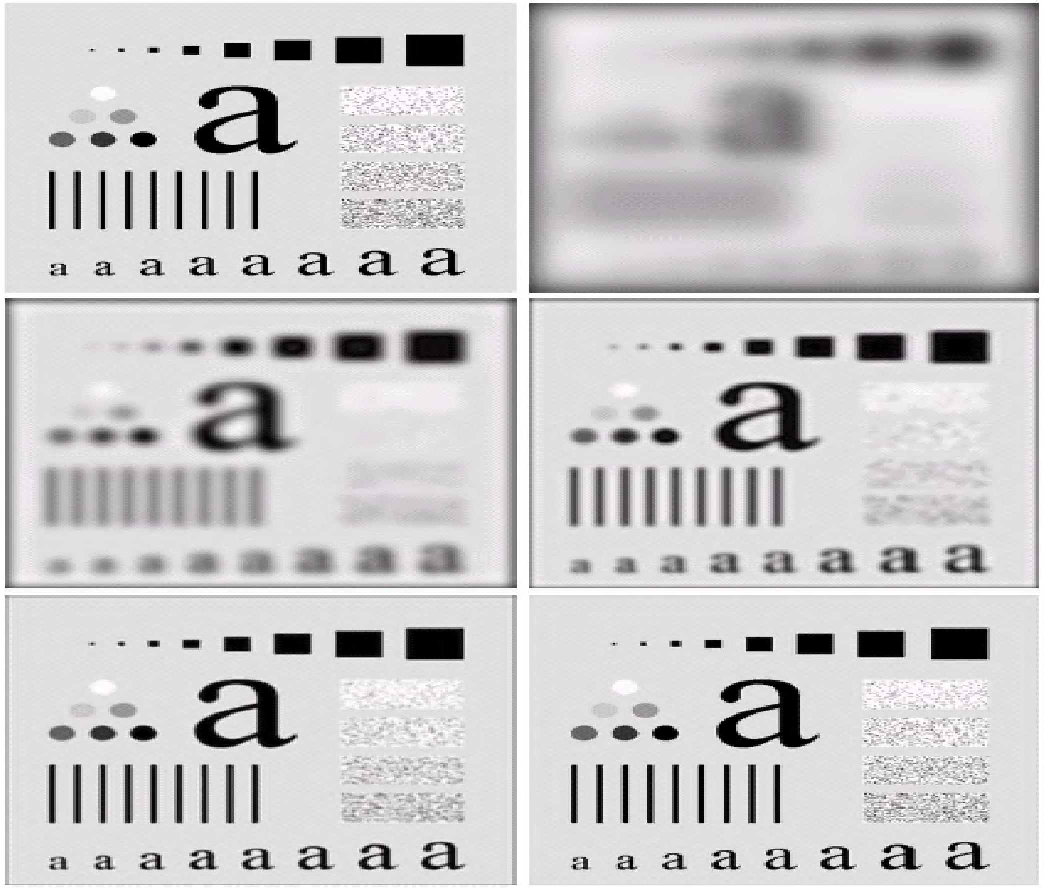
Example

ILPF

BLPF

a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.



a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Spatial representation of BLPFs

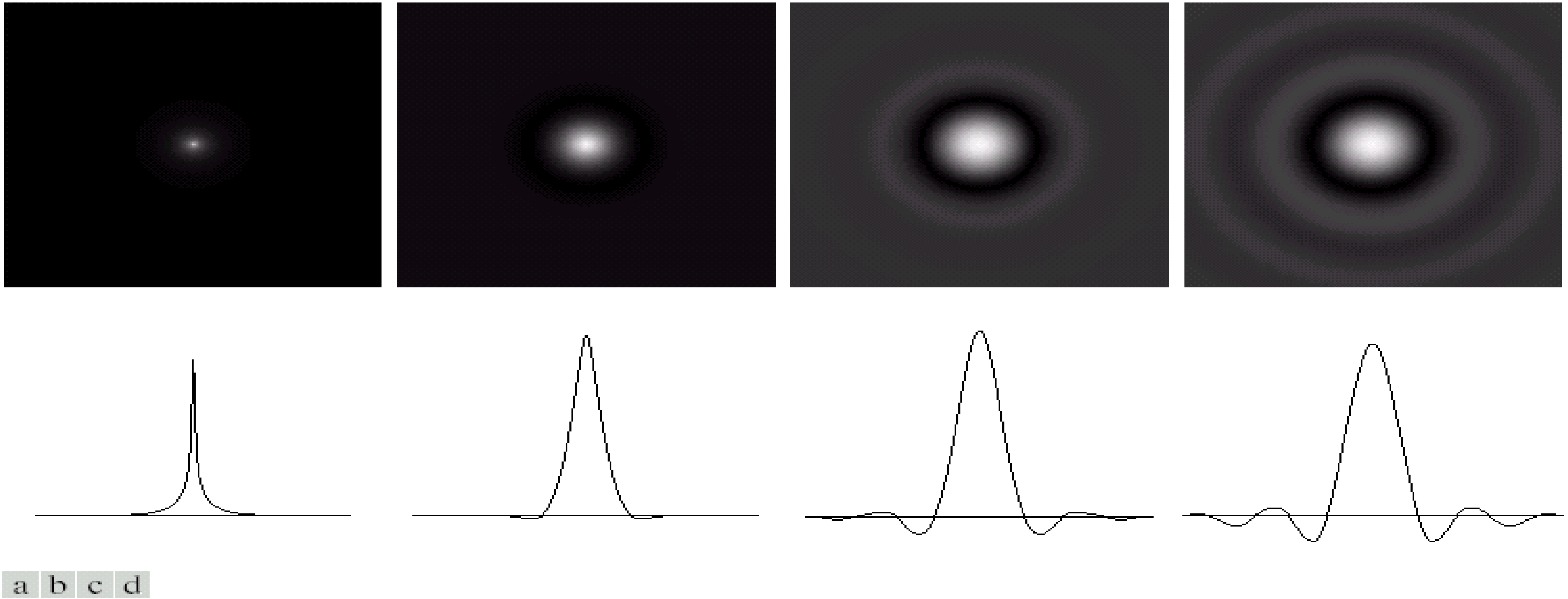


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Spatial representation of BLPFs

- BLPF of order 1 has neither ringing nor negative value.
- Order 2 have mild ringing and small negative value but less than ILPF.
- Ringing in BLPF becomes significant for higher – order filters.
- BLPF of order 20 already have characteristics of the ILPF.

a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Gaussian Lowpass Filter: GLPF

- Filter transfer function: $$H(u,v) = e^{-D^2(u,v)/2\sigma^2};$$

σ : Measure of spread of Gaussian curve

$D(u,v)$: Distance from (u, v) to the origin of frequency rectangle

- Taking $\sigma = D_0$ cutoff frequency: $$H(u,v) = e^{-D^2(u,v)/2D_0^2};$$

- Inverse FT of Gaussian low pass filter is Gaussian. Spatial GLPF will have no ringing.

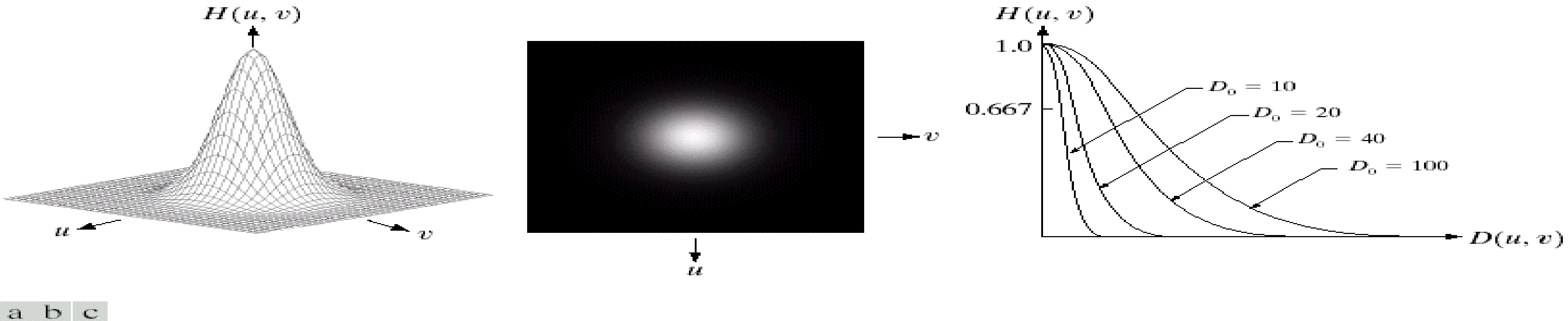


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Example

ILPF

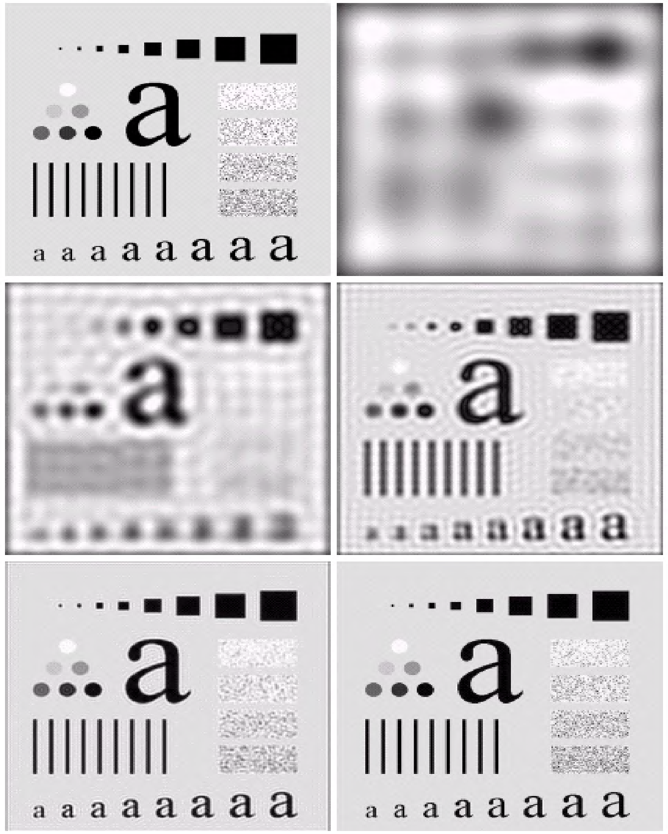


FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

BLPF

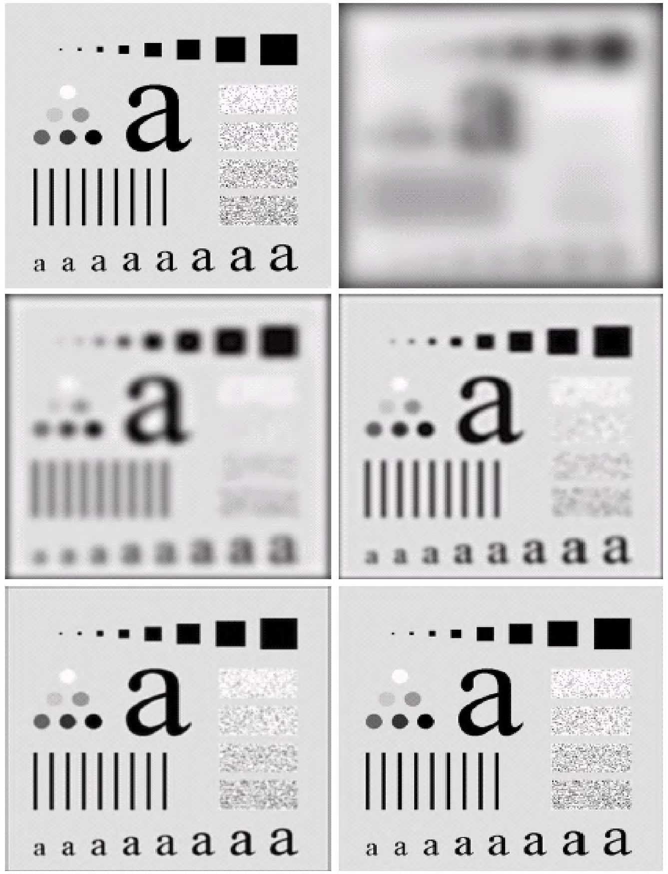


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

GLPF

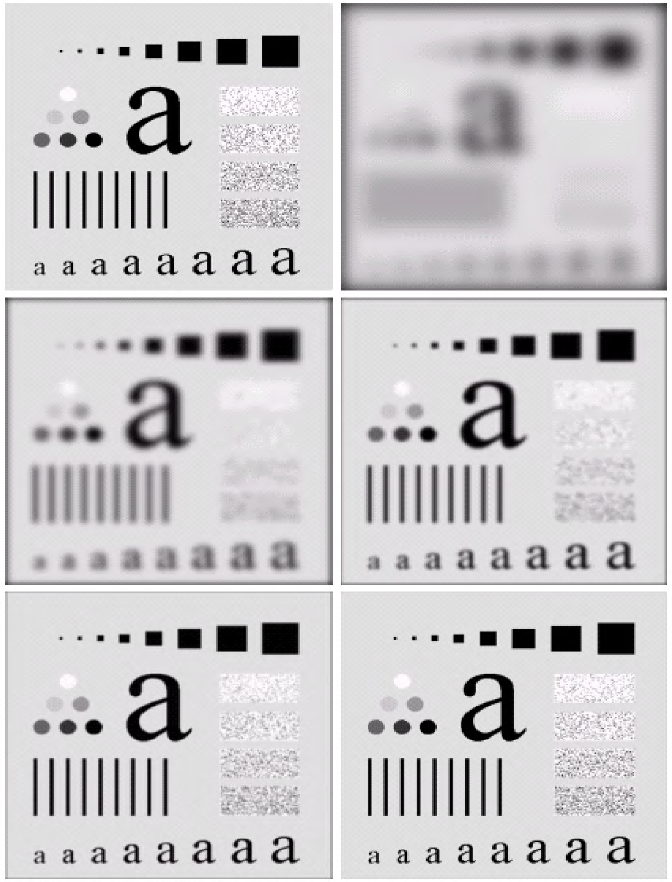
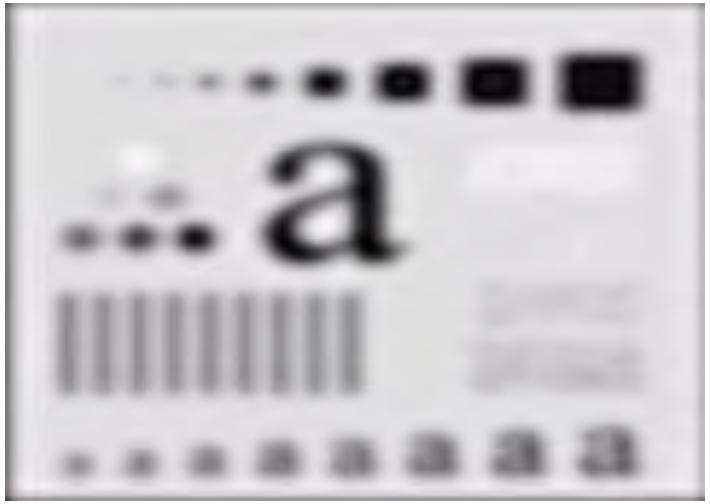


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

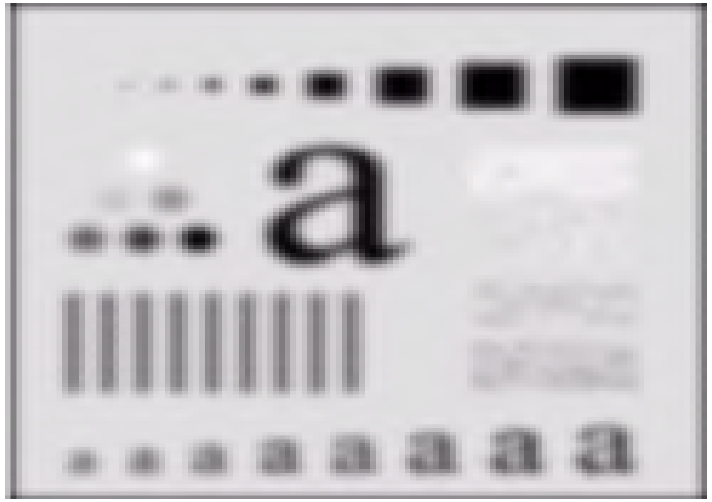
BLPF vs. GLPF

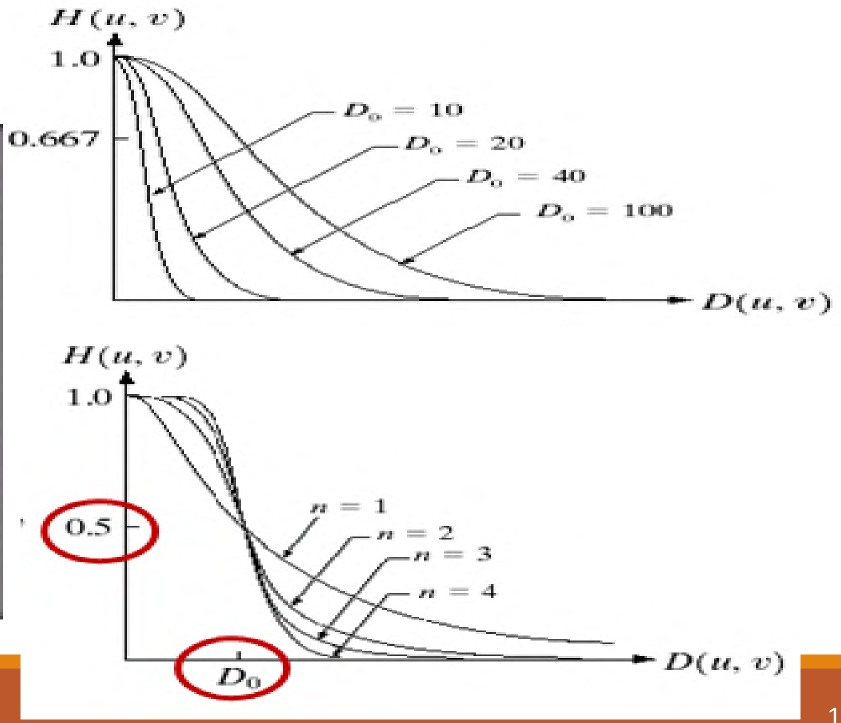
- In case of BLPF smooth transition in blurring as a function of increasing cutoff frequency.
- The GLPF does did not achieve as much smoothing as the BLPF of order 2 of same cutoff frequency.

BLPF (c)



GLPF (c)



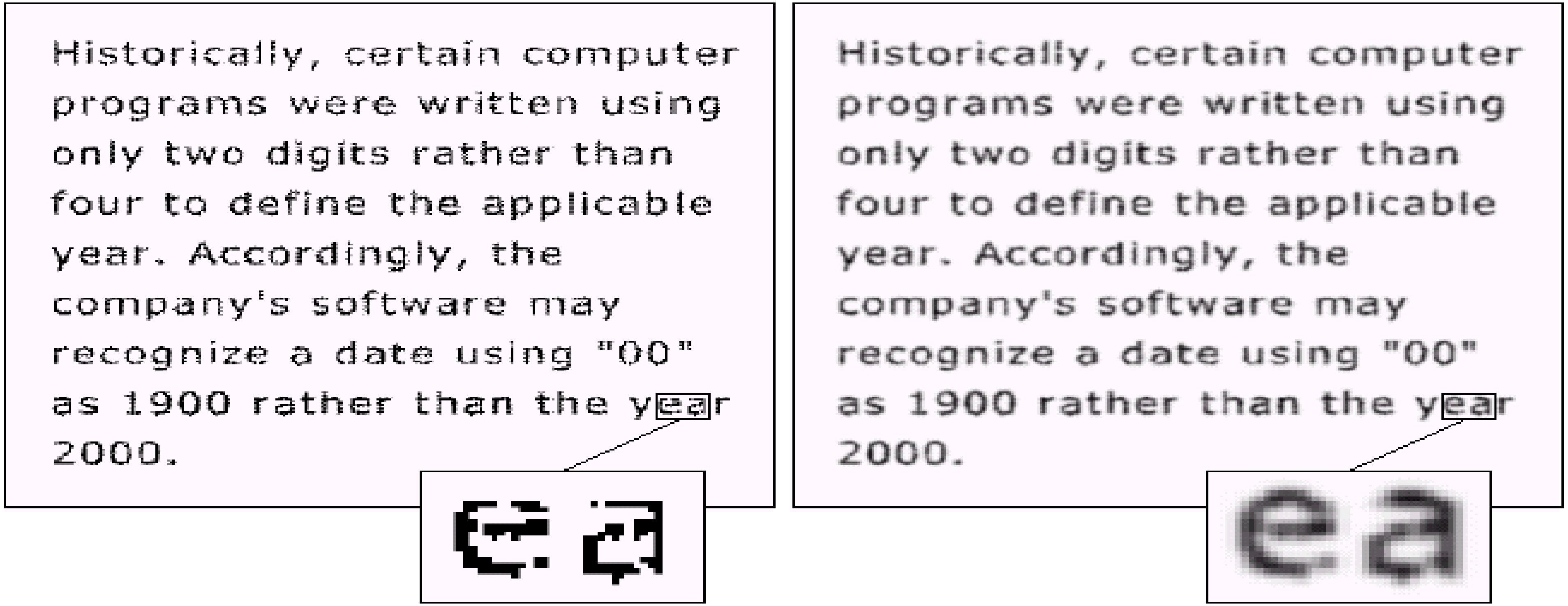


Example

Bridge small gaps in the input image by blurring

a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).



GLPF with D_0 =80

Example

Unsharp masking: Printing and Cosmetic industry

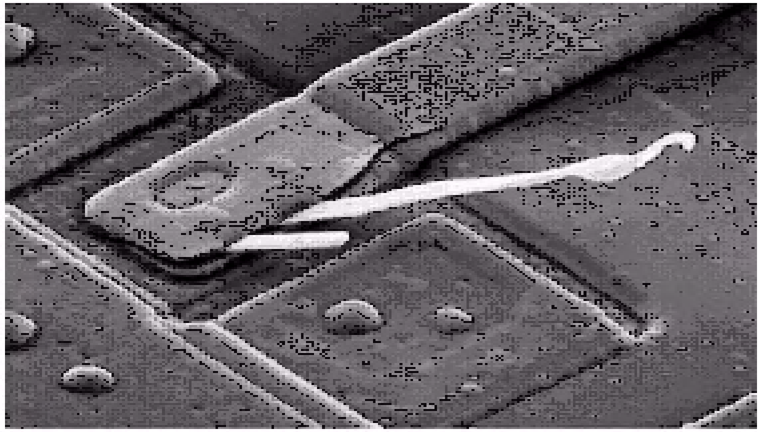
Produce a smoother, softer-looking result from a sharp original.

a b c

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

Notch filter

- this filter is to force the F(0,0) to 0,which is the average value of an image (dc component of the spectrum)
- the output has prominent edges
- in reality the average of the displayed image can't be zero as it needs to have negative gray levels. the output image needs to scale the gray level



$$H(u,v)=\begin{cases}0 & \text{if } (u,v)=(M/2,N/2)\\ 1 & \text{otherwise}\end{cases}$$

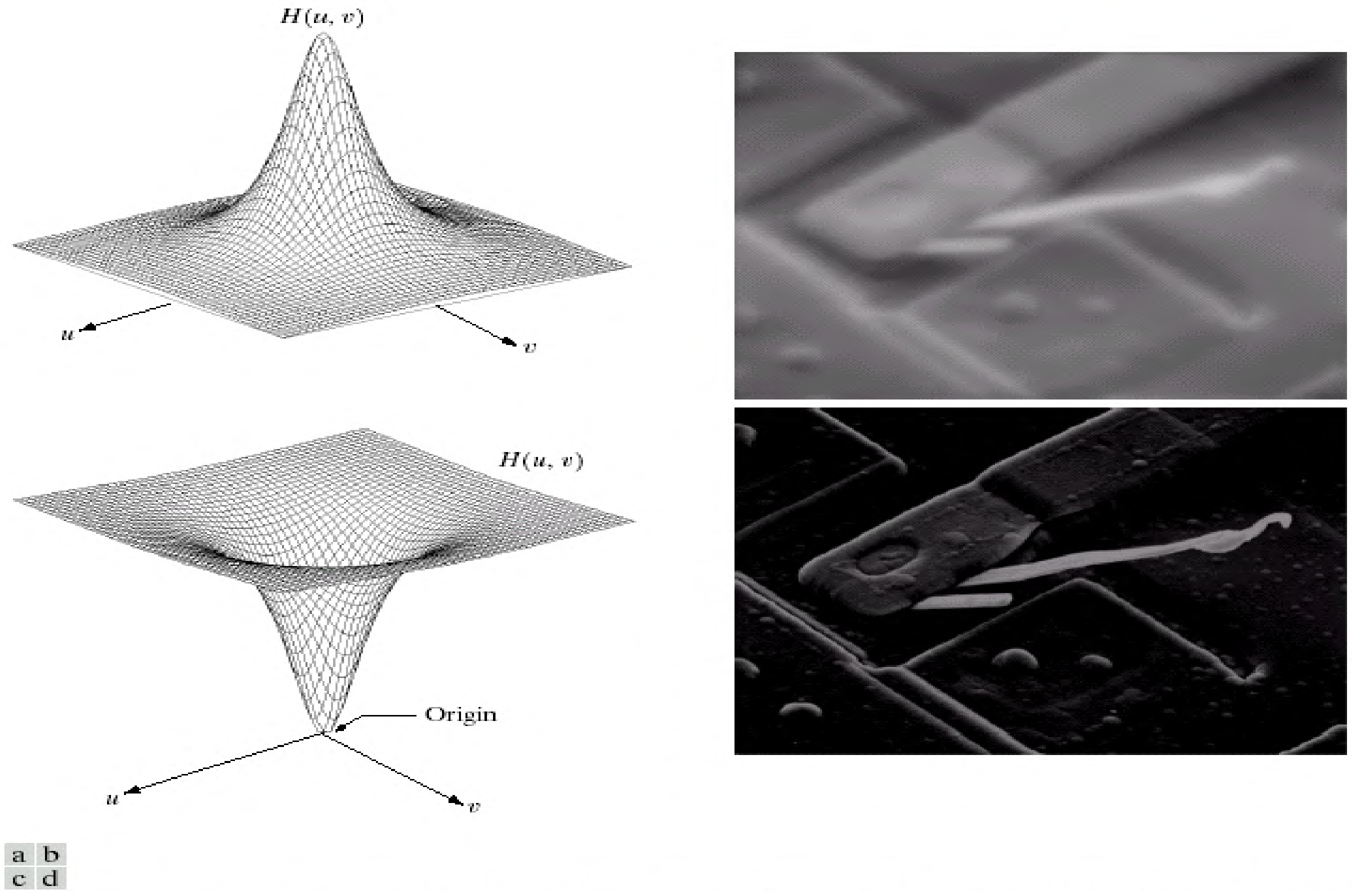


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Low pass filter

High pass filter

Comparison between filtering in spatial and frequency domains

- **Filtering in the frequency domain**

  - Significant degree of intuitiveness regarding how to specify filters.

  - More computational efficiency for a large window size.

  - FD can be viewed as a "laboratory" in which we take advantage of the correspondence between frequency content and image appearance.

  - Some enhancement method is extremely difficult / impossible to formulate in spatial domain but almost trivial in frequency domain.

  - From frequency domain we can get the spatial domain but almost trivial in frequency domain.

- **Filtering in the spatial domain**

  - We often specify small spatial mask that attempt to capture the essence of the full filter function

Sharpening Frequency Domain Filter:

Highpass Filter: Perform the reverse operation of the low pass filter.

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

$H_{lp}(u,v)$: Transfer function of the corresponding low pass filter

When low pass filter cutoff the frequencies, the high pass filter passes them and vice versa.

Sharpening Frequency Domain Filter:

Ideal highpass filter

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Butterworth highpass filter

$$H(u,v) = \frac{1}{1+[D_0/D(u,v)]^{2n}}$$

Gaussian highpass filter

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

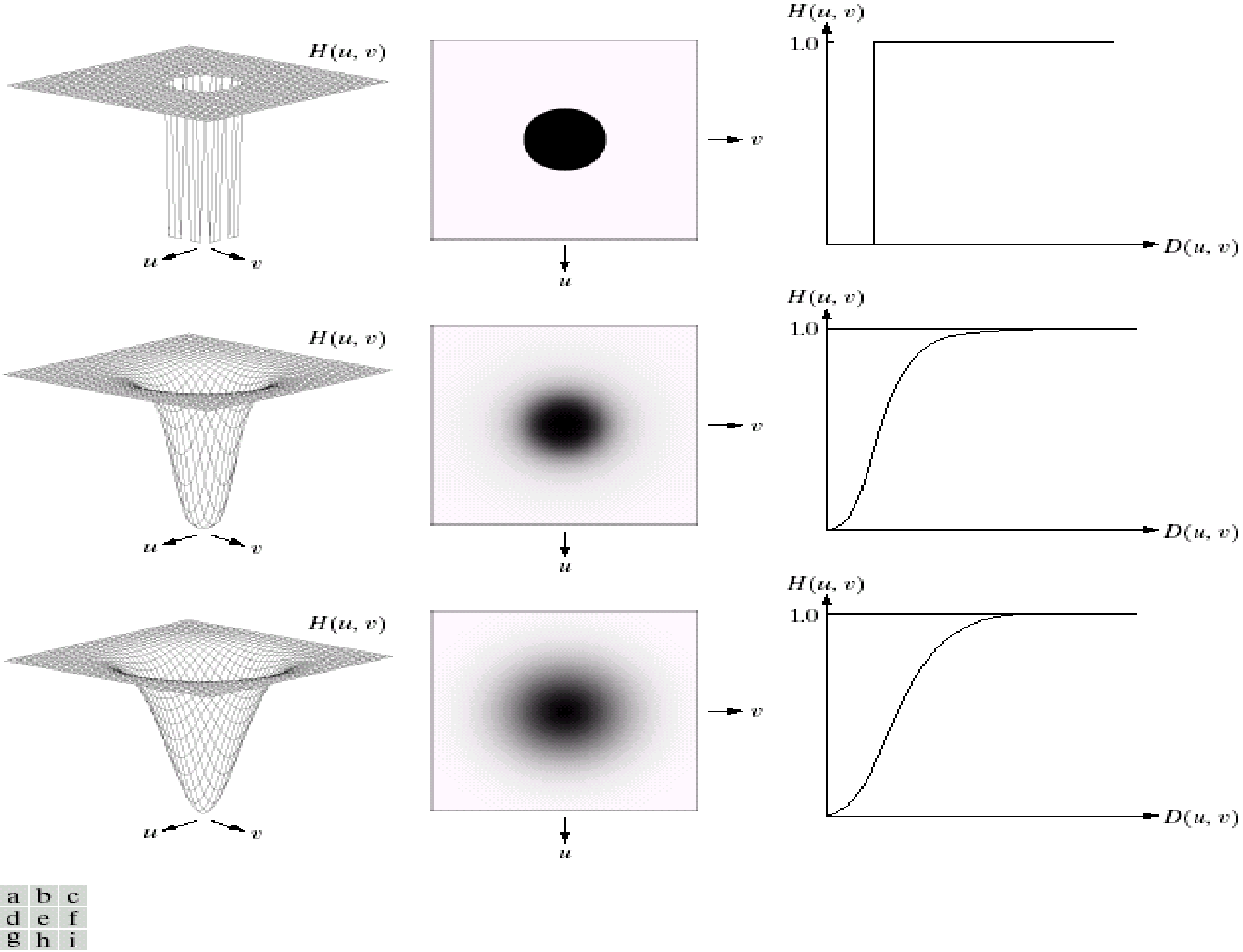


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Spatial representation of Ideal, Butterworth and Gaussian highpass filters

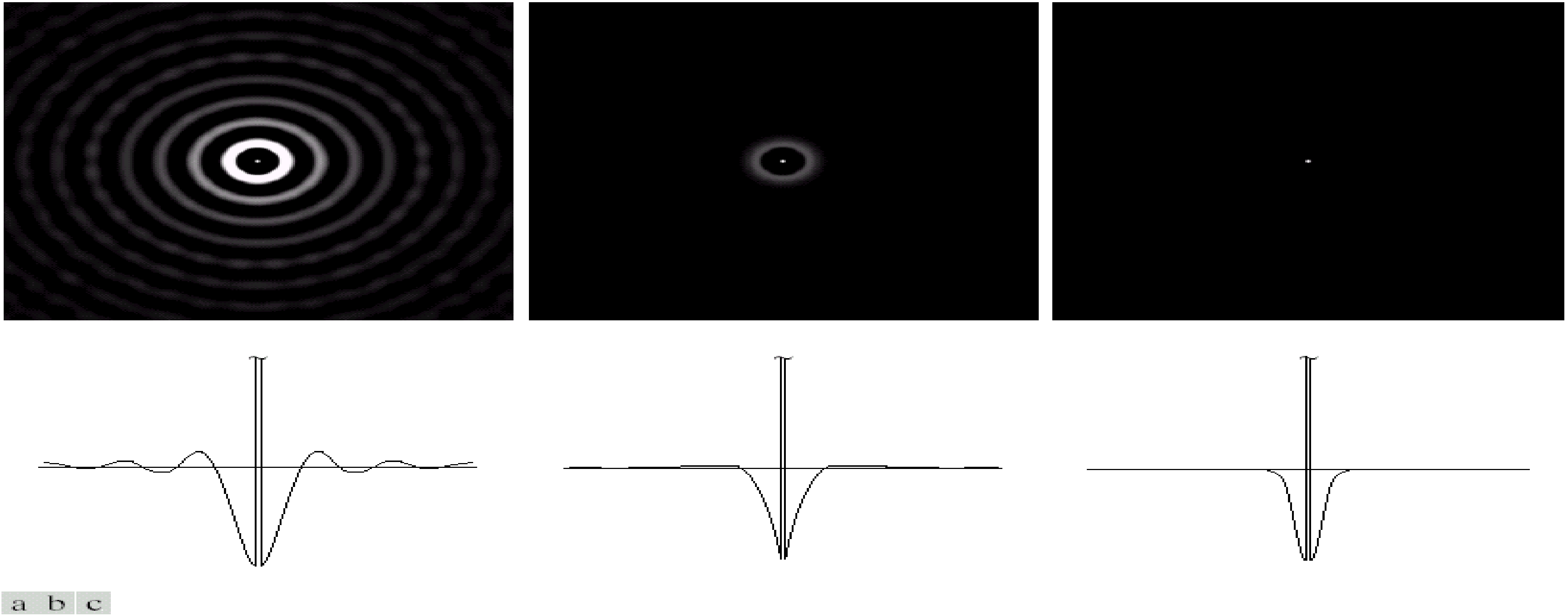
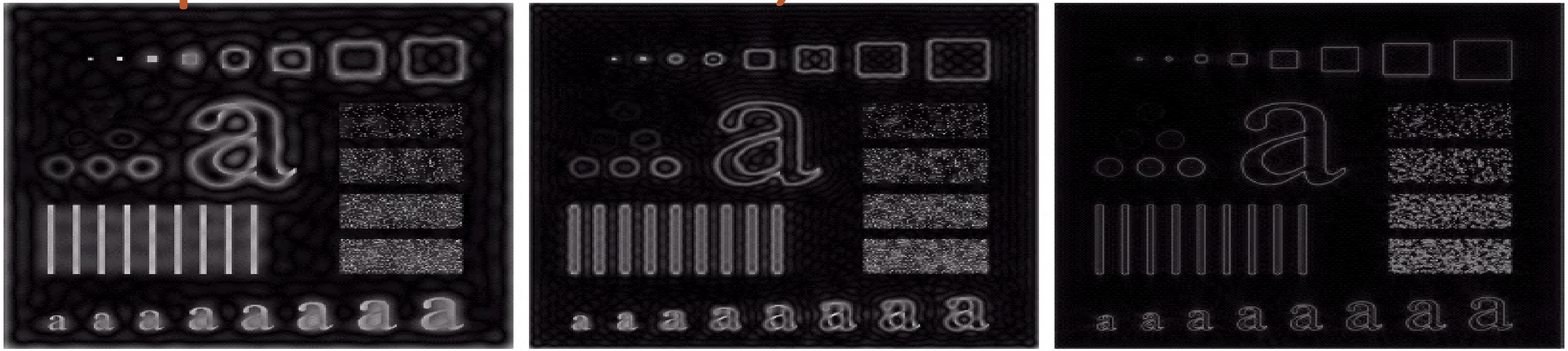


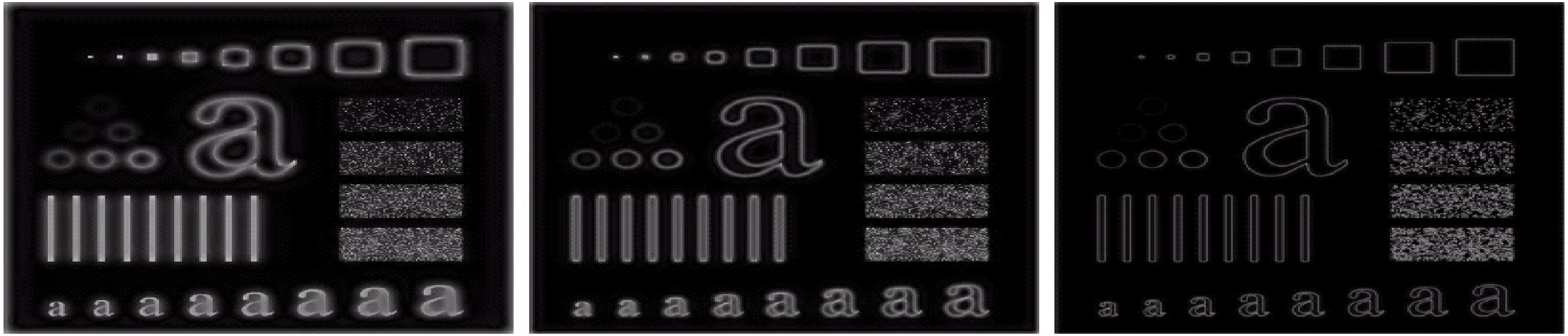
FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

Example: result of IHPF, BHPF



a b c

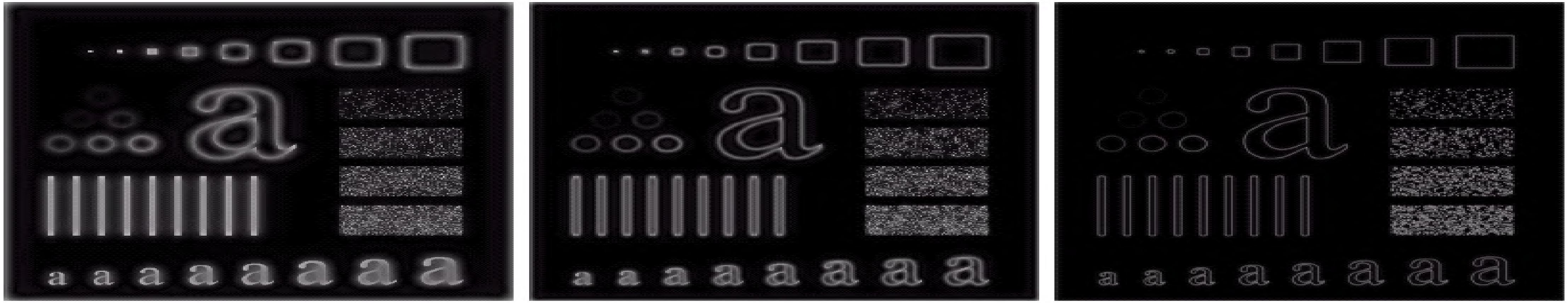
FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).



a b c

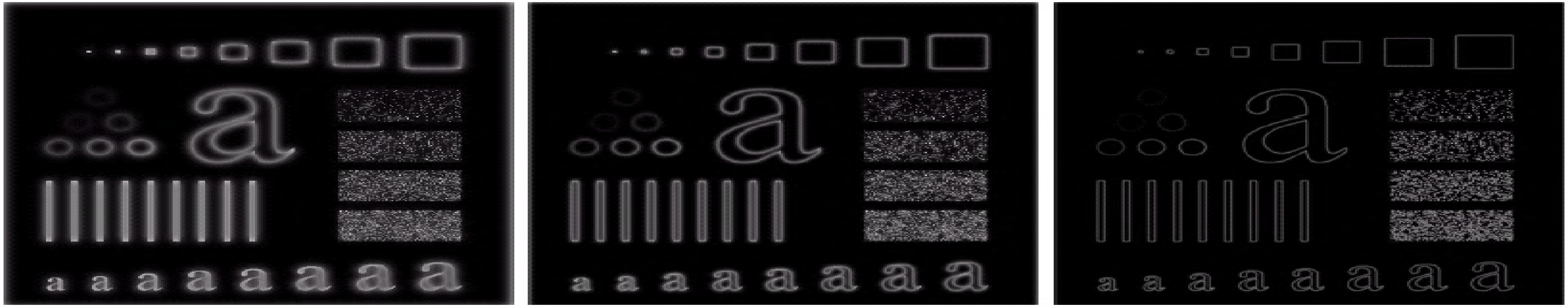
FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

Example: result of BHPF,GHPF



a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

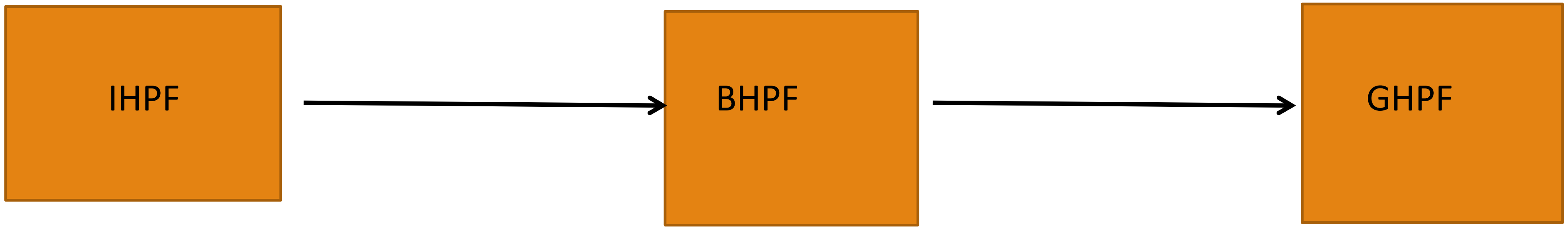


a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Example: result of IHPF, BHPF, GHPF

Increasing order of smoothness



Suggested Readings

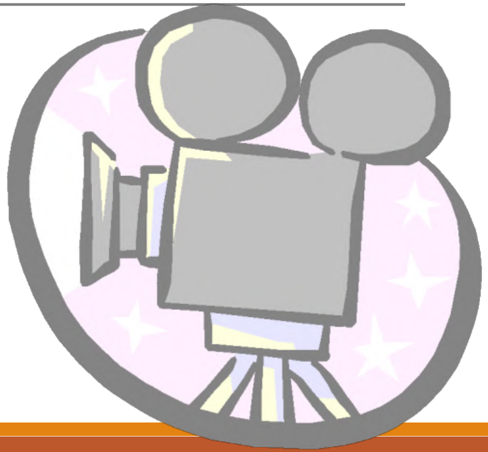
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Enhancement in the Frequency Domain
  - Laplacian Filter
  - Homomorphic Filter

Laplacian in the Frequency domain

The Laplacian in the Frequency domain can be implemented using the filter

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

or, with respect to the center of the frequency rectangle, using the filter

M N

$$H(u, v) = -4\pi^2\left[(u - M/2)^2 + (v - N/2)^2\right]$$
$$= -4\pi^2 D^2(u, v)$$

Where, D(u,v) is the distance function

Laplacian in the Frequency domain: Image Enhancement

The Laplacian image is obtained as:

$$\nabla^2 f(x, y) = \Im^{-1}\{H(u, v)F(u, v)\} \quad = IFT\{H(u,v)F(u,v)\}\}$$

Where, F(u, v) is DFT of f(x, y), Now image enhancement is achieved using the equation (in spatial domain):

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

Here, c=-1, because H(u, v) is negative.

Laplacian in the Frequency domain: Image Enhancement

In frequency domain

$$
\begin{aligned}
g(x, y) &= \Im^{-1}\left\{F(u, v) - H(u, v)F(u, v)\right\} \\
&= \Im^{-1}\left\{\left[1 - H(u, v)\right]F(u, v)\right\} \\
&= \Im^{-1}\left\{\left[1 + 4\pi^2 D^2(u, v)\right]F(u, v)\right\}
\end{aligned}
$$

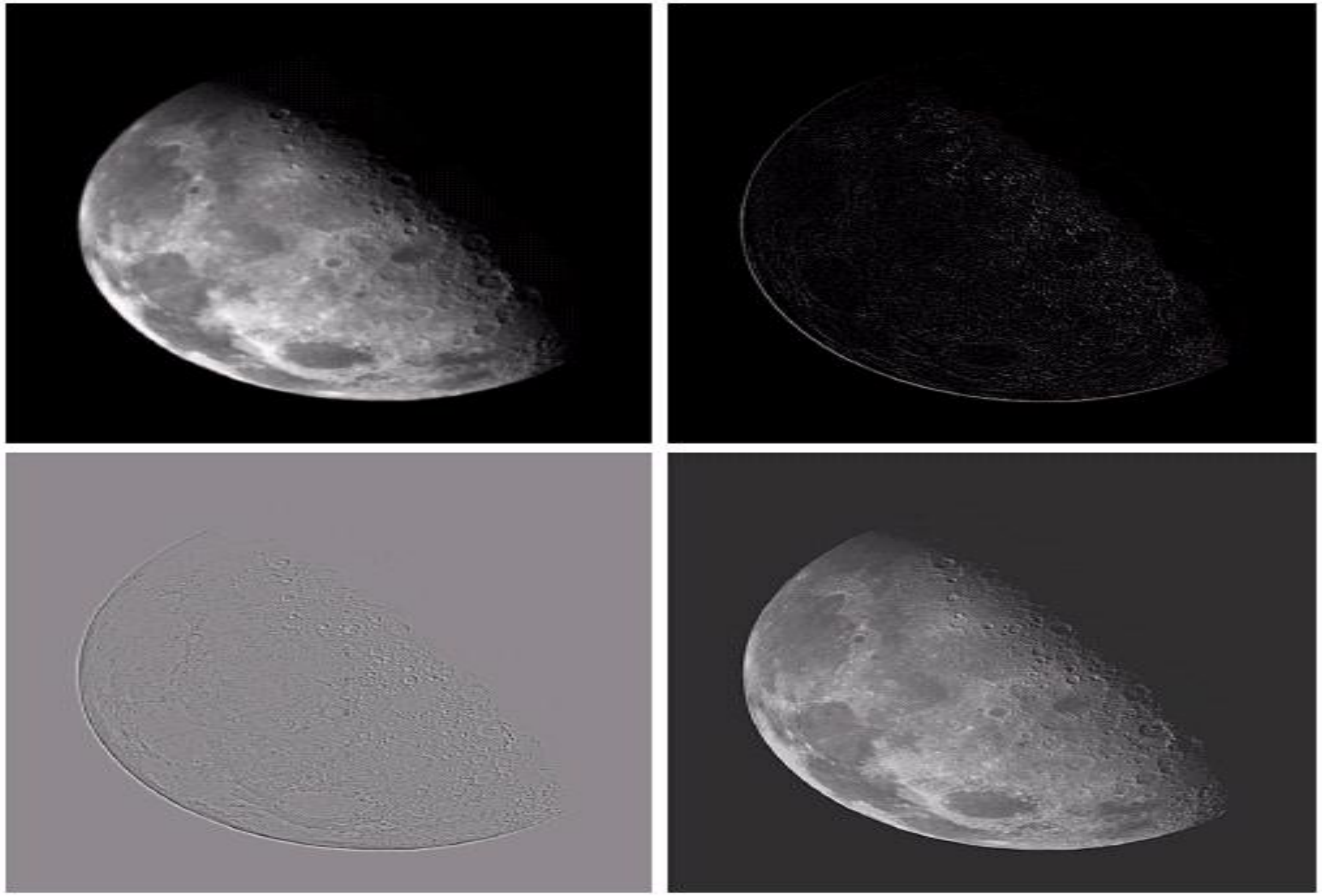
As,

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

Example: Laplacian filtered image

a	b
c	d

FIGURE 4.28
(a) Image of the North Pole of the moon.
(b) Laplacian filtered image.
(c) Laplacian image scaled.
(d) Image enhanced by using Eq. (4.4-12).
(Original image courtesy of NASA.)



Unsharp masking

The unsharp masking is expressed as:

$$g_{mask}(x, y) = f(x, y) - f_{lp}(x, y)$$

$$With,\ f_{lp} = \nabla^2 f(x, y) = \Im^{-1}\{H_{LP}(u, v)F(u, v)\}$$

Here, H_{LP}(u,v) is a low pass filter and F(u,v) is DFT of f(x, y),

Then we add a weighted portion of the mask back to the original image:

$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

Here, we included a weight k ($k \geq 0$), for generality.

When k=1, we have **unsharp masking**.

When k>1, the Process is referred to as **highboost filtering**.

$$Lowpass$$
$$g(x,y) = IFT\{(1+k*(1-H_{LP}(u,v))F(u,v)\}$$
$$Highpass,\ filter$$
$$g(x,y) = IFT\{[1+k*H_{HP}(u,v)]F(u,v)\}$$

$$General..Formula$$
$$g(x,y) = IFT\{[k_1 + k_2 * H_{HP}(u,v)]F(u,v)\}$$

Where, k_1>=0, offsets from origin
k2>=0 , contribute to High frequency filtering

Homomorphic Filtering

The illumination and reflectance model can be used to develop a frequency domain procedure for improving the appearance of an image by simultaneous intensity range compression and contrast enhancement.

An image f(x, y) can be expressed as the product of illumination i(x, y) and reflectance r(x, y) components as:

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

This equation can not be used directly to operate on the frequency components of illumination and reflectance because Fourier Transformation of a product is not the product of the transformations

$$\Im[f(x, y)] \neq \Im[i(x, y)]\,\Im[r(x, y)]$$

Homomorphic Filtering

Suppose we define,

$Z(x, y)=\ln f(x, y)$
$Z(x, y)= \ln i(x, y) +\ln r(x, y)$

Now, in frequency domain FT of z(x, y)
$FT(z(x, y)= FT(\ln i(x, y) +\ln r(x, y))$
$FT(z(x, y)= F(\ln i(x, y)) +FT(\ln r(x, y))$
$z(u, v)= F_i(u, v) +F_r(u, v)$

where $F_i(u, v)$ and $F_r(u, v)$ are the Fourier transforms of $\ln i(x, y)$ and $\ln r(x, y)$, respectively.

Now image is in Frequency domain, so we can apply filter

Homomorphic Filtering

We can filter $Z(u, v)$ using a filter $H(u, v)$ so that

$$
\begin{aligned}
S(u, v) &= H(u, v)Z(u, v) \\
&= H(u, v)F_i(u, v) + H(u, v)F_r(u, v)
\end{aligned}
$$

The filtered image in the spatial domain is

$$
\begin{aligned}
s(x, y) &= \Im^{-1}\{S(u, v)\} \\
&= \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\}
\end{aligned}
$$

Where, $\Im^{-1}$ is the inverse FT.

By defining

$$i'(x, y) = \Im^{-1}\left\{H(u, v)F_i(u, v)\right\}$$

and

$$r'(x, y) = \Im^{-1}\left\{H(u, v)F_r(u, v)\right\}$$

we can express

$$s(x, y) = i'(x, y) + r'(x, y)$$

Finally, because $z(x, y)$ was formed by taking the natural logarithm of the input image, we reverse the process by taking the exponential of the filtered result to form the output image:

$$\begin{aligned} g(x, y) &= e^{s(x,y)} \\ &= e^{i'(x,y)} e^{r'(x,y)} \\ &= i_0(x, y) r_0(x, y) \end{aligned}$$

where

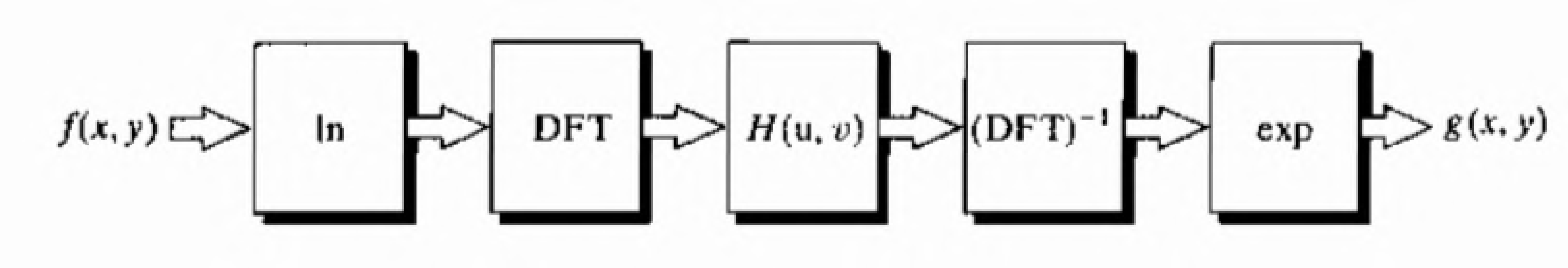
$$i_0(x, y) = e^{i'(x,y)}$$

and

$$r_0(x, y) = e^{r'(x,y)}$$

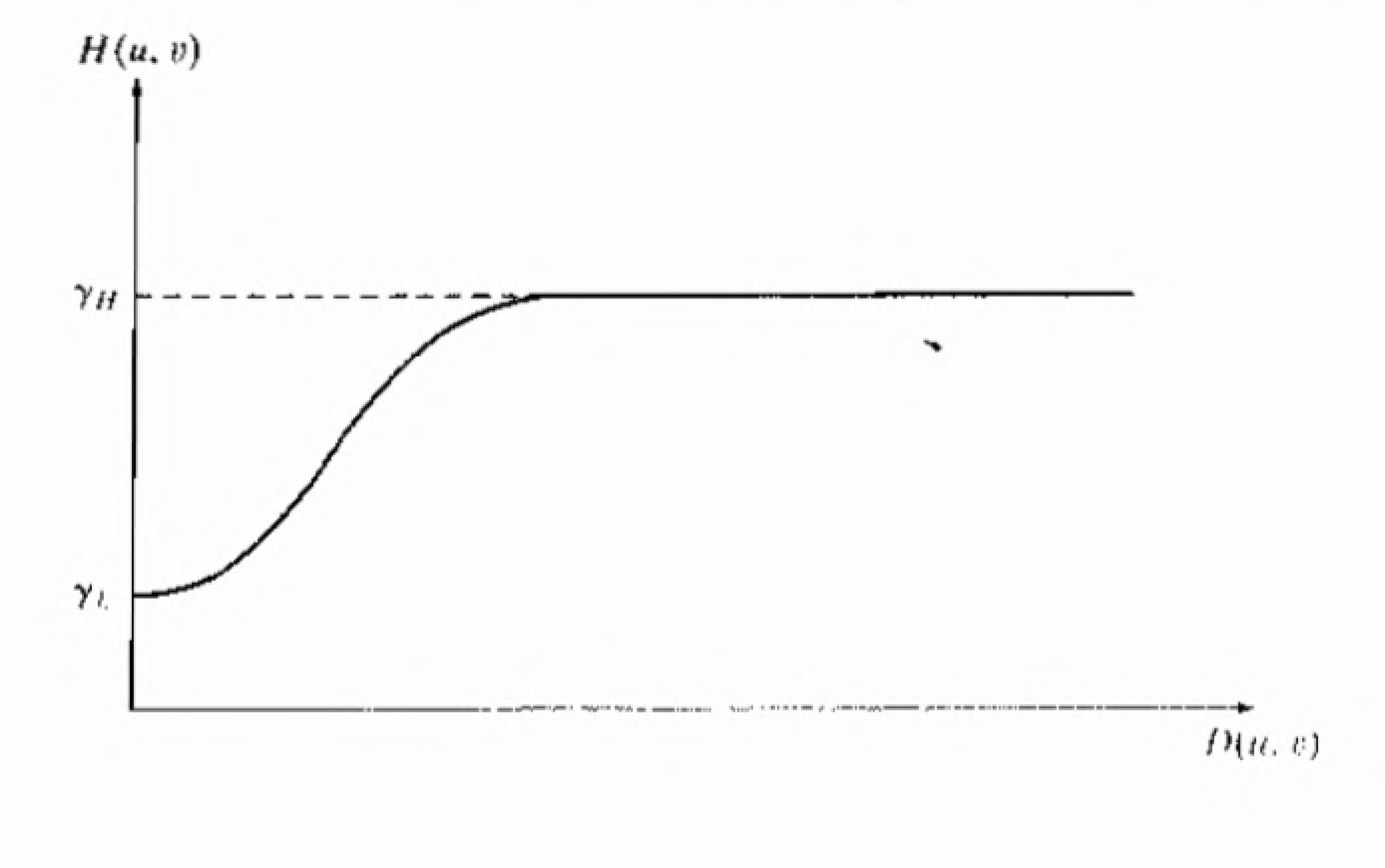
are the illumination and reflectance components of the output (processed) image.

Steps of Homomorphic Filtering



γ_L<1, it compresses low frequency components Υ_H>1, contrast enhancement

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c[D^2(u,v)/D_0^2]}\right] + \gamma_L$$



Suggested Readings

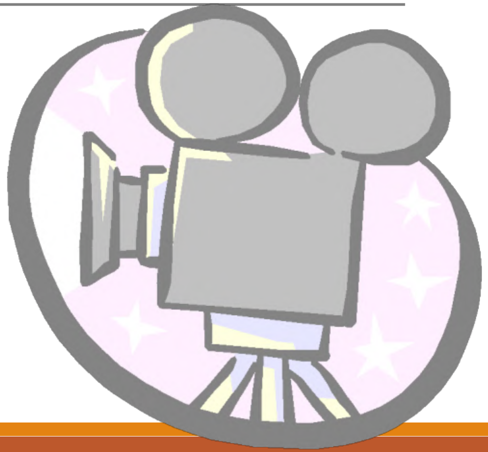
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Restoration
  - Noise Models
  - Image denoising

Image Restoration

- Restoration attempts to recover an image that has been degraded by using a priori knowledge of the degradation phenomenon.

- Restoration techniques are oriented towards modeling the degradation and applying the inverse process in order to recover the original image.

Image Restoration Example

- Contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects, it might present to viewer.

- Removal of image blur by applying a deblurring function is considered a Restoration techniques.

Model of the Image Degradation

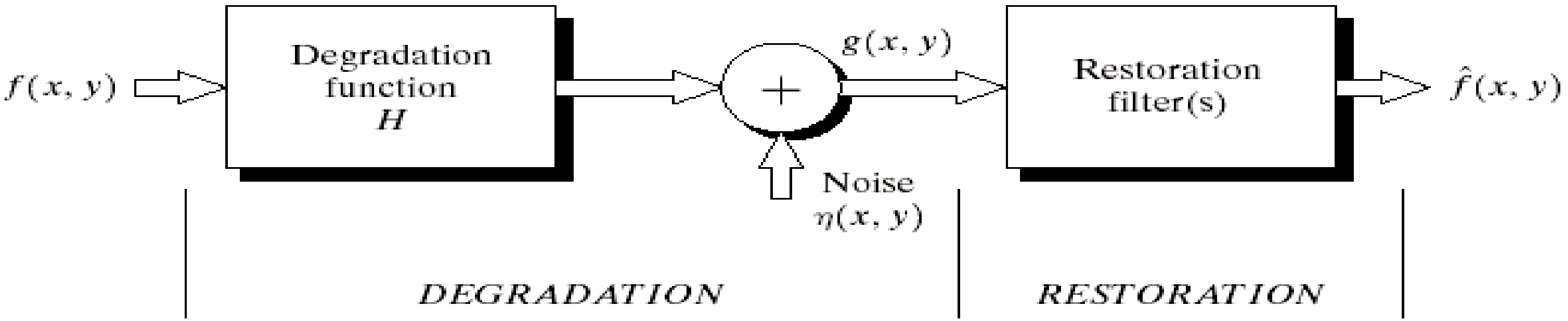


FIGURE 5.1 A model of the image degradation/ restoration process.

Model of the Image Degradation

Spatial Domain

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

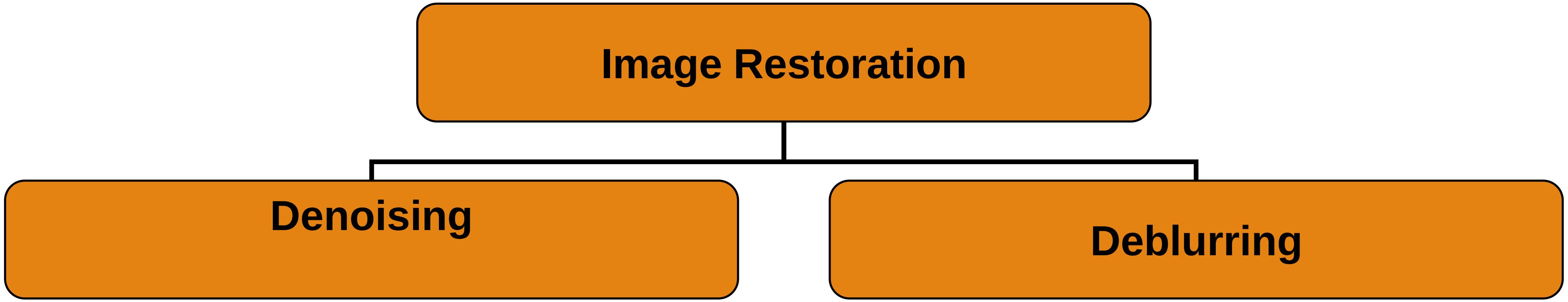
$*$ Convolution Operator
h(x, y) is the spatial degradation function,
g(x, y) is degraded image

Frequency Domain

$$G(u,v) = H(u,v)F(u,v) + \eta(u,v)$$

Objective of restoration To obtain an estimate $\hat{f}(x,y)$ of the original image, for given $g(x,y)$

Image Restoration



Denoising

$$G(u,v) = F(u,v) + \eta(u,v)$$

Deblurring

$$G(u,v) = H(u,v)F(u,v) + \eta(u,v)$$

Noise Model

- The principal sources of noise in digital images arise during image acquisition and/or transmission.

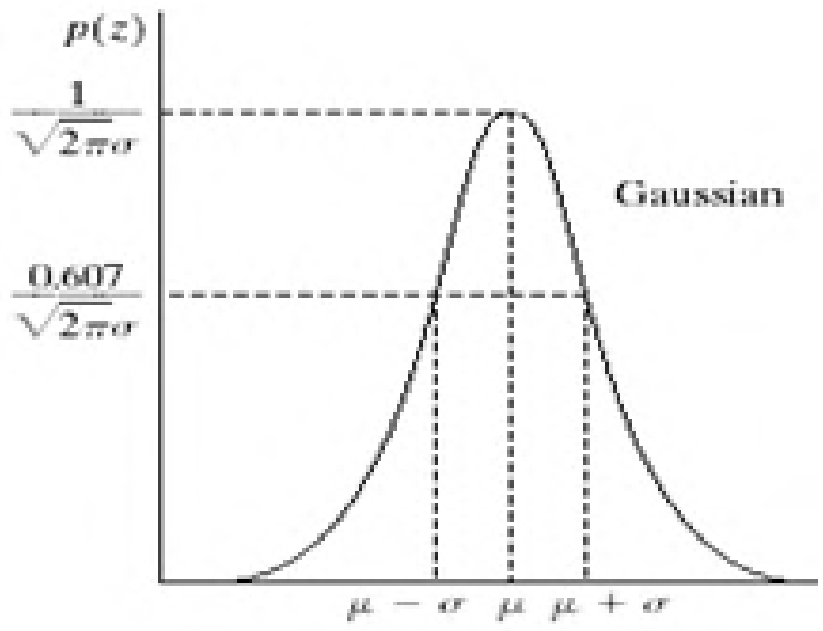
- The performance of imaging sensors is affected by a variety of factors such as environment condition during image acquisition and by the quality of sensing elements themselves.

- Images are corrupted during transmission principally due to interference in the channel used for transmission.

The spatial noise descriptor with which we shall be concerned is the statistical behavior of the intensity values in the noise component of the image restoration model .

These may be considered random variables characterized by a probability density function.

Noise Model



Gaussian noise:

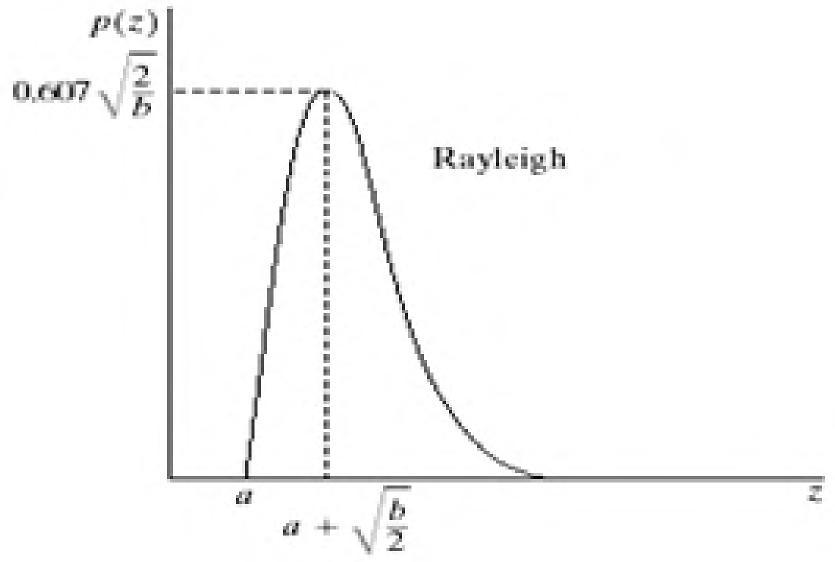
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

Z=intensity

Noise Model

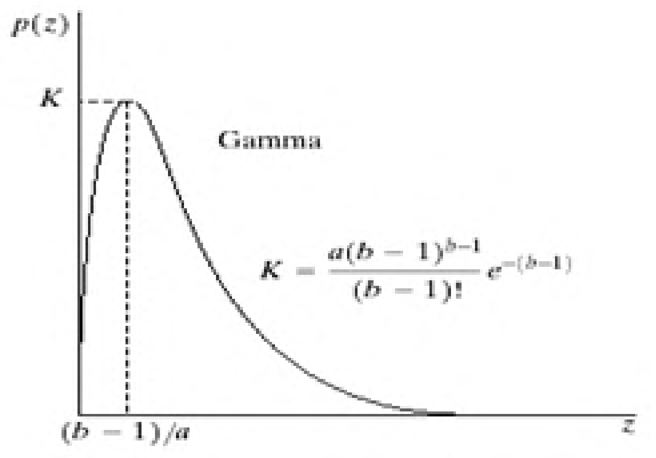
Rayleigh Noise:

$$p(z)=\begin{cases}\dfrac{2}{b}\left(z-a\right)e^{-\left(z-a\right)^2/b}, & for\ z\geq a\\ 0, & for\ z<a\end{cases}$$



$$\mu=a+\sqrt{\pi b/4},\qquad \sigma^2=\frac{b\left(4-\pi\right)}{4}$$

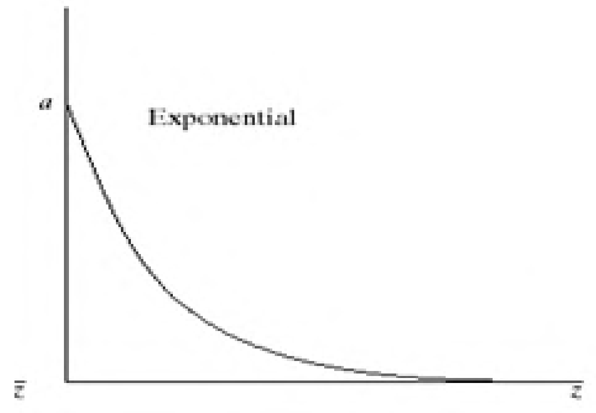
Erlang (Gamma) noise:



$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az}, & for\ z \geq 0 \\ 0, \ \ for\ z < 0 \end{cases}$$

$$\mu = \frac{b}{a}, \qquad \sigma^2 = \frac{b}{a^2}$$

Exponential Noise:

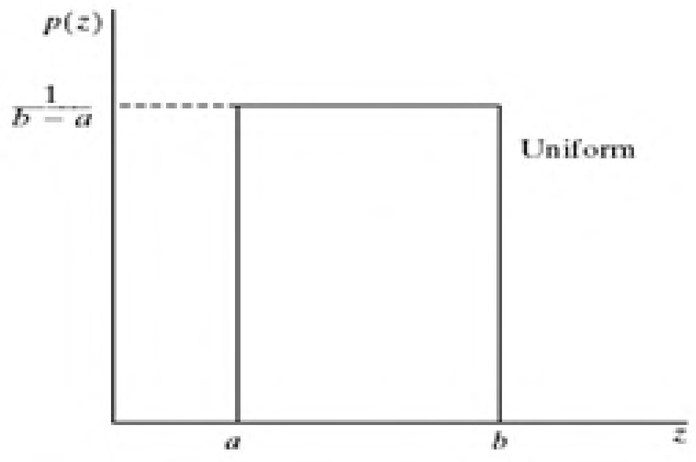


$$p(z) = \begin{cases} ae^{-az}, & for\ z \geq 0 \\ 0, \ \ for\ z < 0 \end{cases}$$

$$\mu = \frac{1}{a}, \qquad \sigma^2 = \frac{1}{a^2}$$

Special case of Erlang PDF ,with *b*=1

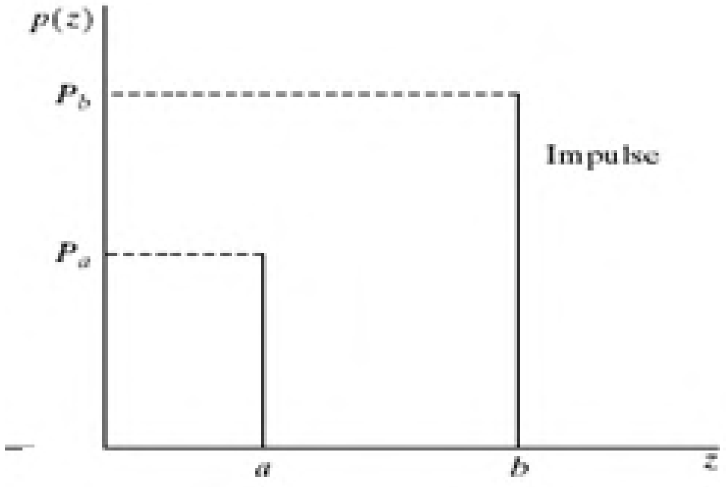
Uniform noise:



$$p(z)=\begin{cases}\dfrac{1}{(b-a)}, & if\ \ a\le\ z\le b\\ 0, & \text{Otherwise}\end{cases}$$

$$\mu=\frac{a+b}{2},\qquad \sigma^2=\frac{(b-a)^2}{12}$$

Impulse (Salt – and –pepper) Noise:



$$p(z)=\begin{cases}P_a, & for\ z=a\\ P_b, & for\ z=b\\ 0, & Otherwise\end{cases}$$

$$\mu=\frac{1}{a},\qquad \sigma^2=\frac{1}{a^2}$$

Noise Model

Impulse (Salt – and –pepper) Noise:

$$p(z) = \begin{cases} P_a, & for\ z = a \\ P_b, & for\ z = b \\ 0, & Otherwise \end{cases}$$

$$\mu = \frac{1}{a}, \qquad \sigma^2 = \frac{1}{a^2}$$

- *b*> *a*, gray level *b* will appear as light dot in the image. Conversely, level *a* will appear like dark dot.
- If either P_a or P_b is zero, the impulse noise is called unipolar.
- If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image.
- Due to this reason bipolar noise is also called salt-and- pepper noise.
- Also Shot and spike noise term is used to refer this type of noise.

Noise Model

- ***Gaussian:*** Noise due to factors such as electronic circuit noise and sensor noise due to poor illumination and / or high temperature.
- ***Rayleigh:*** Rayleigh density is helpful in characterizing noise phenomena in range imaging.
- **Exponential and Gamma** density are found application in range imaging.
- **Impulse noise** is found in satuation where quick transients, such as faulty switching, takes place during imaging.
- **Uniform density** is useful as the basis for numerous random number generator.

Estimation of Noise Parameter

Noise is estimated using histogram

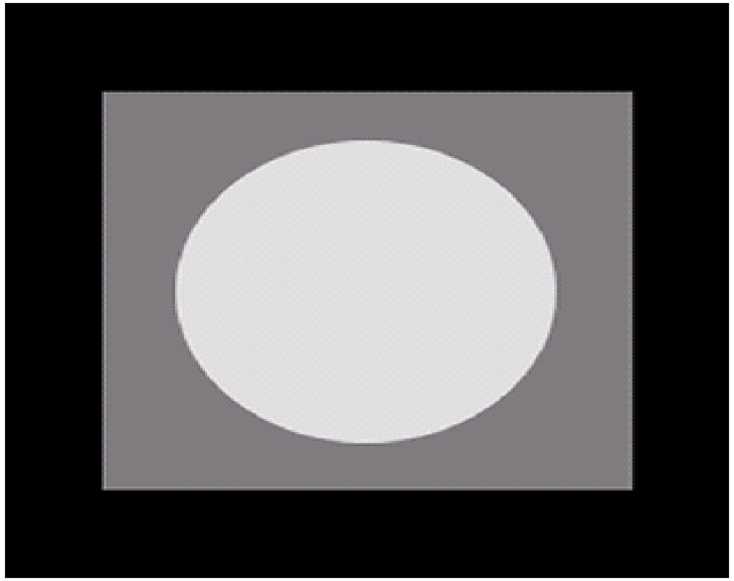


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs

$$\mu = \sum z_i p(z_i), \qquad \sigma^2 = \sum (z_i - \mu)^2 p(z_i)$$

Estimation of Noise Parameter

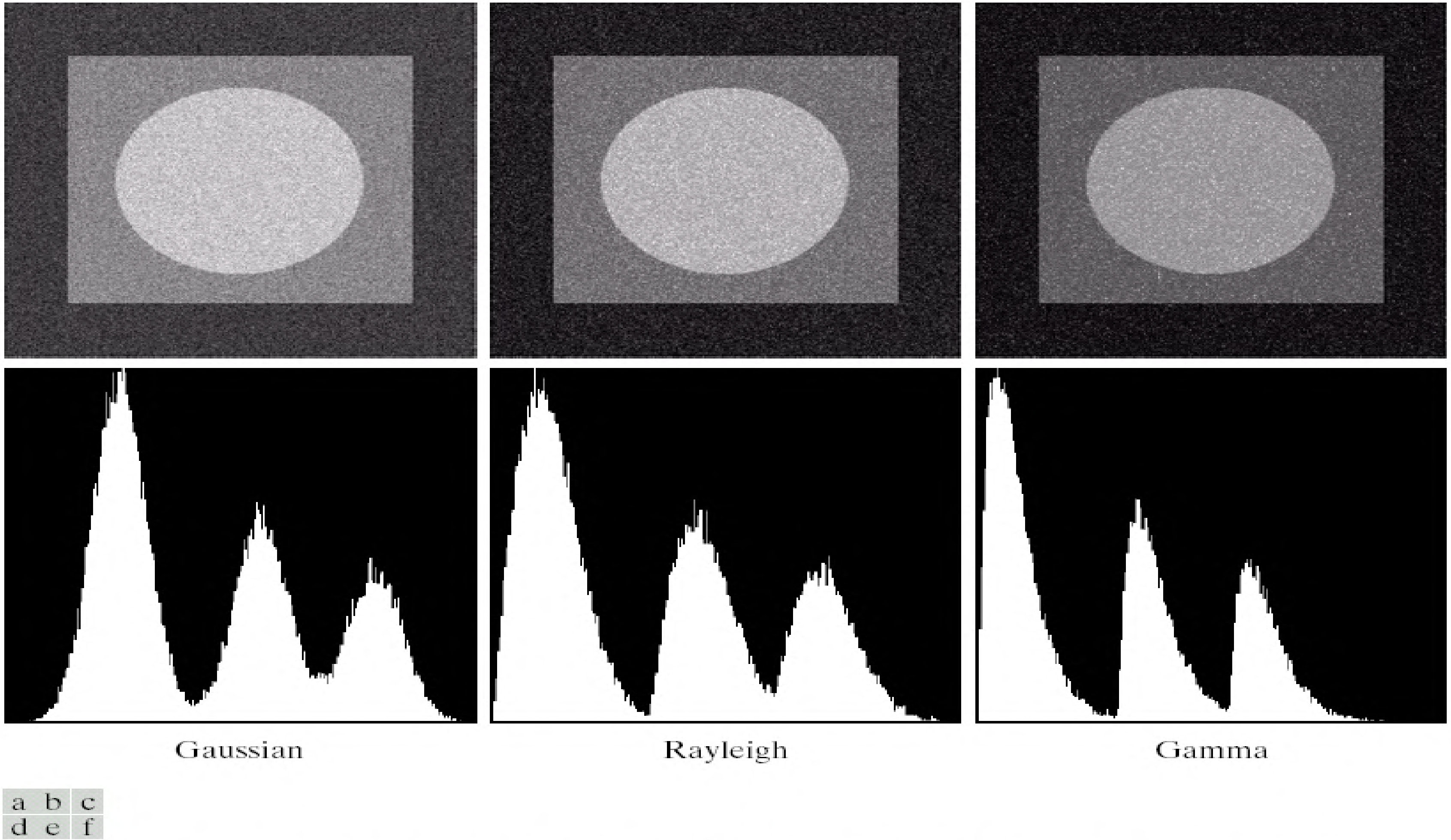


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Estimation of Noise Parameter

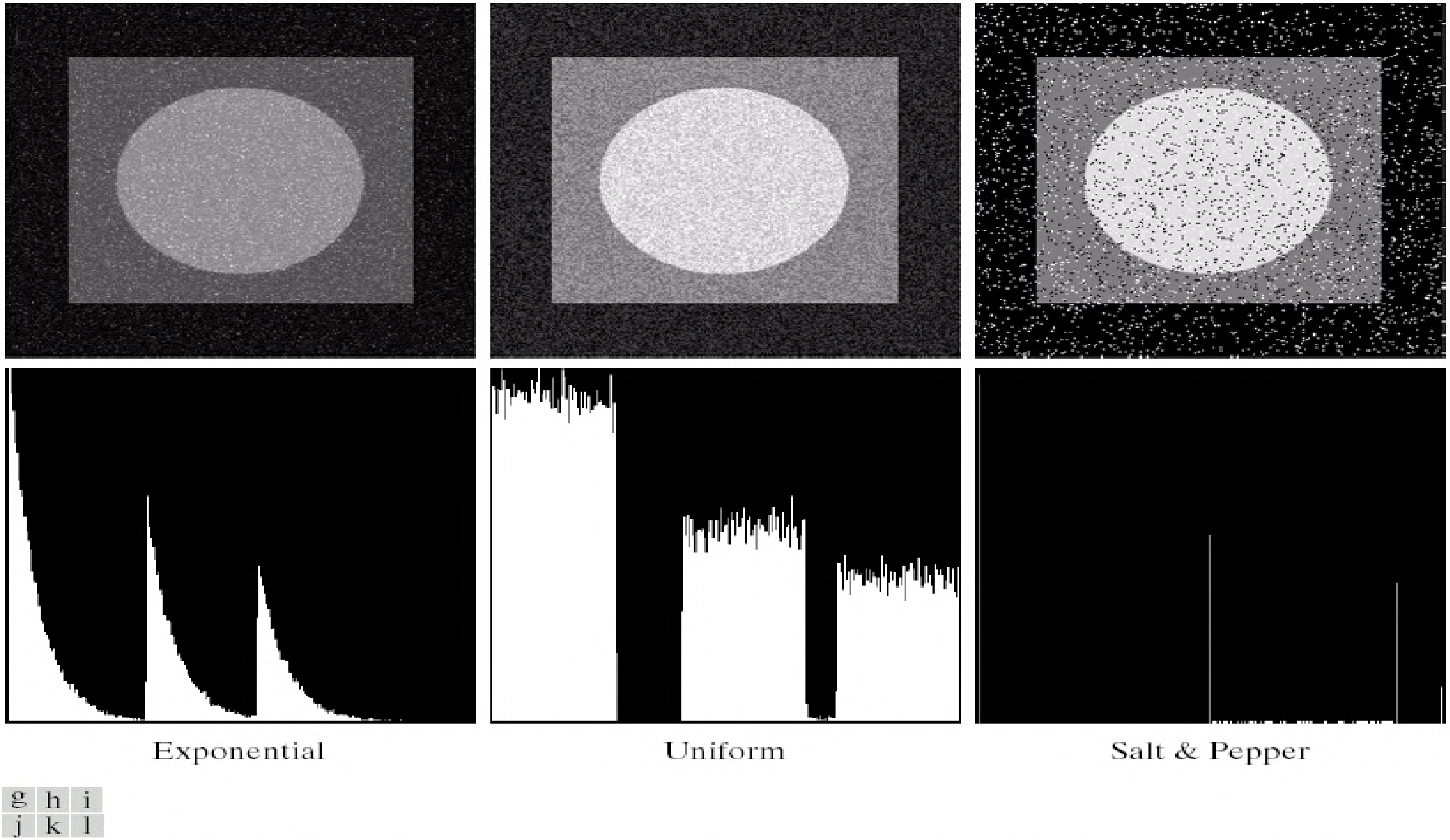


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

De-noising using Spatial Filter

Spatial filtering is method of choice in situations when only additive noise is present.

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$G(u,v) = F(u,v) + \eta(u,v)$$

Arithmetic Mean Filter

$$\hat{f}(x,y) = \frac{1}{mn}\sum_{s,t\in W_{mn}} g(s,t)$$

-Smoothes local variations in an image.

- Noise is reduced as a result of blurring

Geometric Mean Filter

$$\hat{f}(x,y) = \left[\prod_{s,t\in W_{mn}} g(s,t)\right]^{1/mn}$$

-Achieves smoothing comparable to the arithmetic mean filter.

-Lose less image detail in the process.

Harmonic Filter

$$\hat{f}(x,y) = \frac{mn}{\sum_{s,t\in W_{mn}} \frac{1}{g(s,t)}}$$

-Works well for salt noise, Gaussian noise, but fails for pepper noise

Contraharmonic Mean Filter

$$\hat{f}(x,y) = \frac{\sum_{s,t\in W_{mn}} g(s,t)^{Q+1}}{\sum_{s,t\in W_{mn}} g(s,t)^{Q}}$$

-This filter is well suited for salt-and- pepper noise. Q>0 for pepper and Q<0 for salt noise.

Q=0 , Arithmetic mean filter

Q=-1, Harmonic mean filter

Arithmetic Mean and Geometric Mean Filter

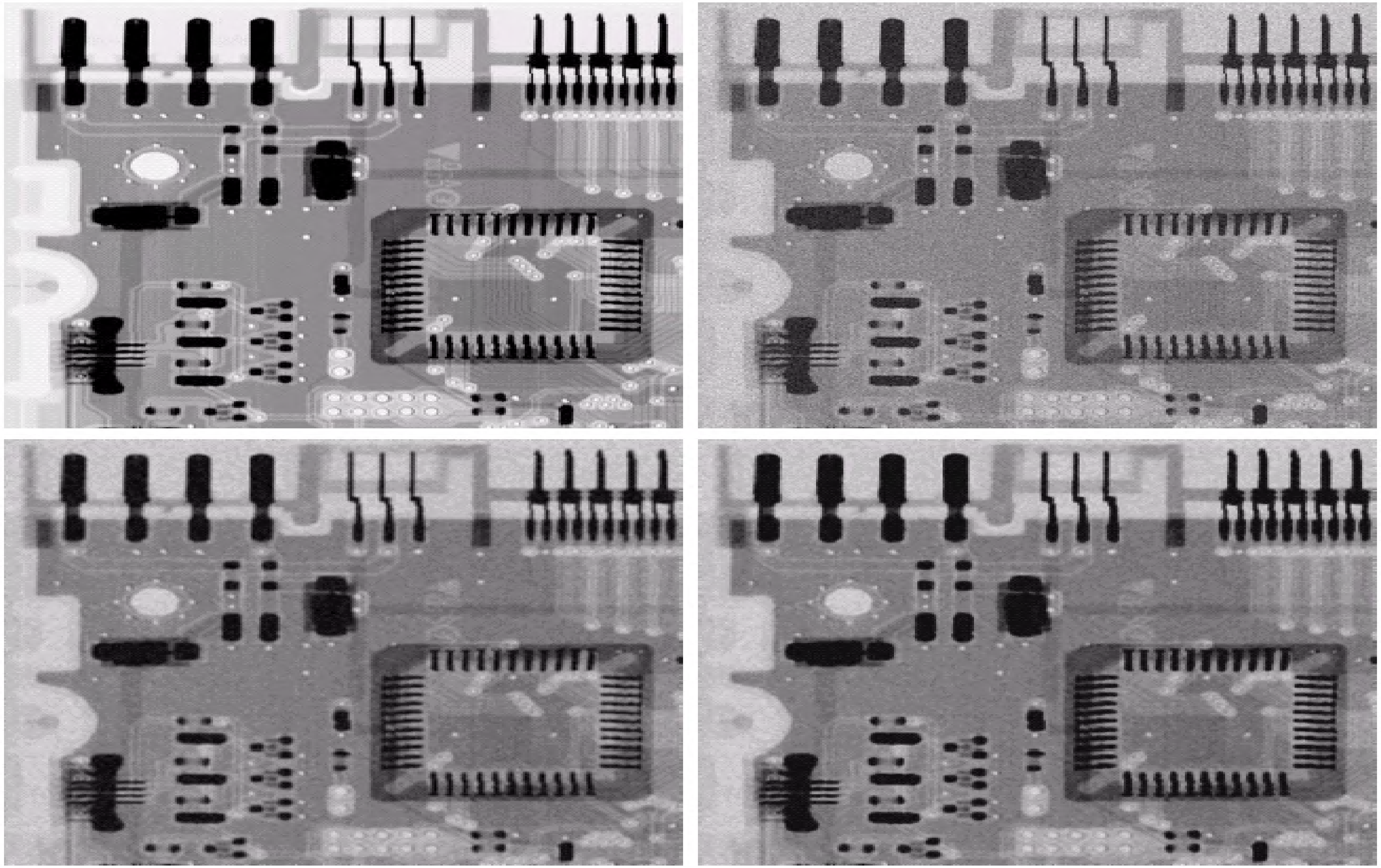
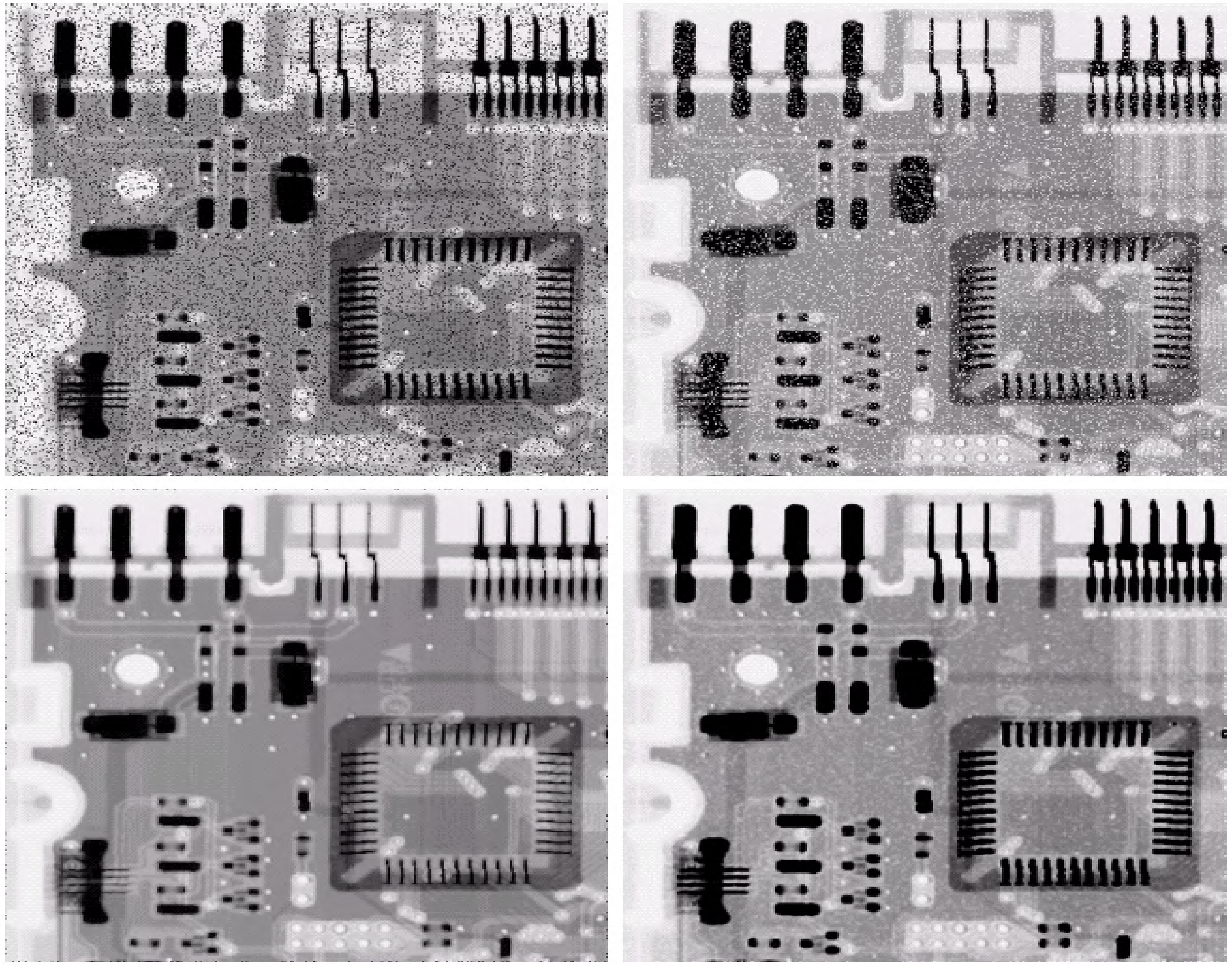


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Contraharmonic Mean Filtering

a b
c d

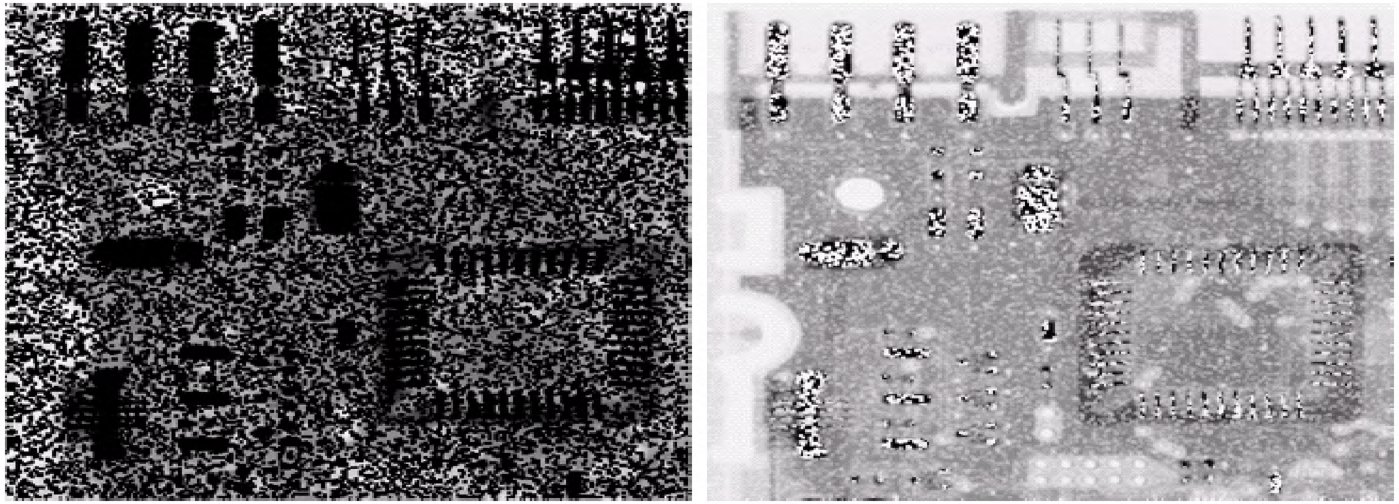
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



-Positive-order filters did better job of cleaning the background, at the expense of blurring the dark areas.

- The opposite was true of the negative filter.

Contraharmonic Mean Filtering



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

- Arithmetic mean and Geometric mean filters are well suited for random noise like Gaussian or uniform noise.

- The contraharmonic filter is well suited for impulse noise, but it has drawback that it must be known whether the noise is dark or light in order to select the proper sign for Q.

De-noising using Order -Statics Filters

Median Filter

$$\hat{f}(x,y) = \underset{s,t \in W_{mn}}{median}\{g(s,t)\}$$

-Provide excellent noise reduction capabilities, with considerably less blurring than linear smoothing filters of similar support.

Max and Min Filter

$$\hat{f}(x,y) = \max_{s,t \in W_{mn}}\{g(s,t)\}$$

- This filter is useful for finding the brightest points in an image. Useful in reducing pepper noise.

$$\hat{f}(x,y) = \min_{s,t \in W_{mn}}\{g(s,t)\}$$

- Useful for finding darkest point in an image. It reduces the salt noise.

Midpoint Filter

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{s,t \in W_{mn}}\{g(s,t)\} + \min_{s,t \in W_{mn}}\{g(s,t)\}\right]$$

- Works best for randomly distributed noise, like Gaussian or uniform noise

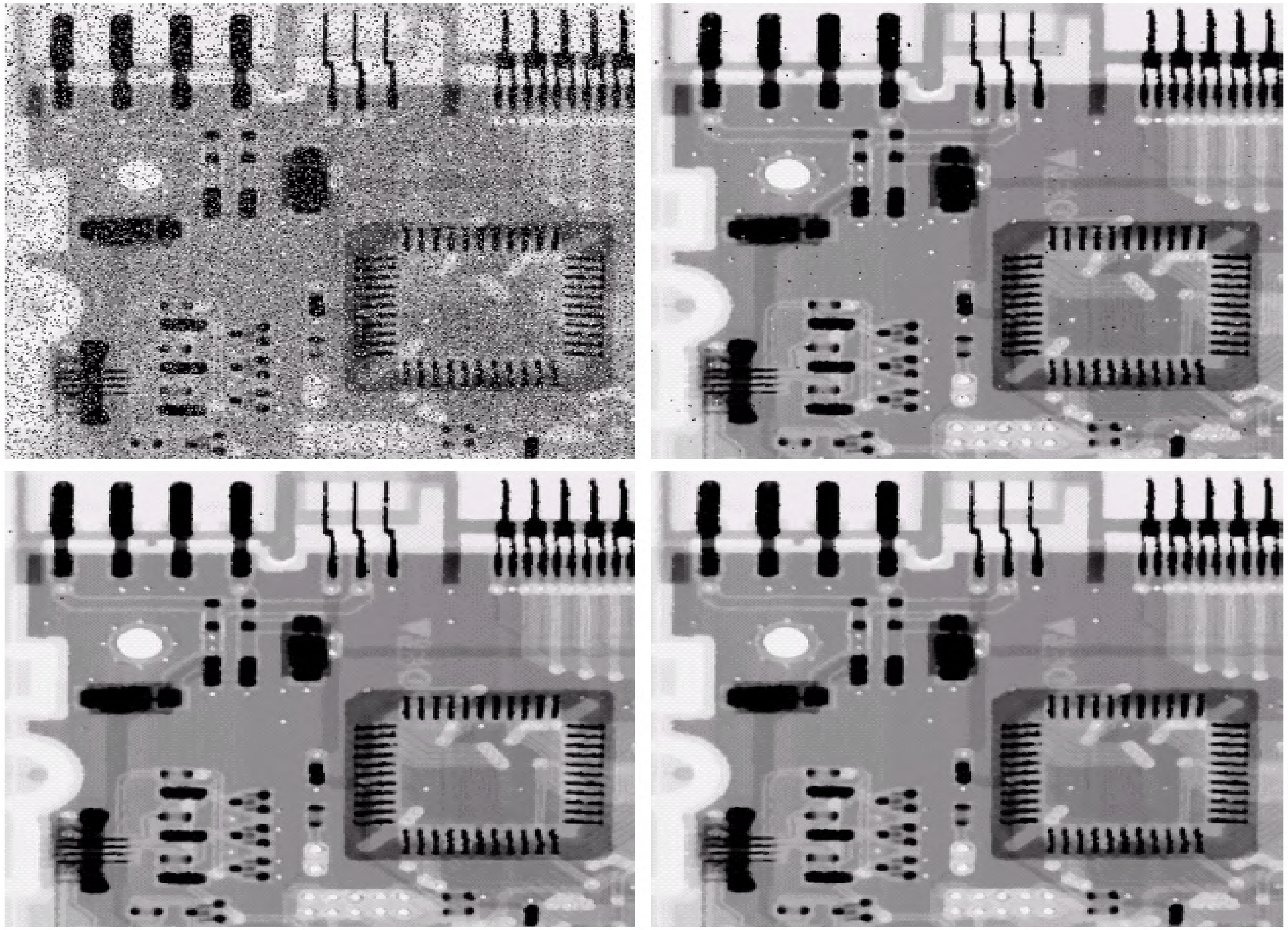
De-noising using Order -Statics Filters

Alpha-trimmed Mean Filter

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{s,t \in W_{mn}} g_r(s,t)$$

- *d* can range from *0* to *mn-1.*
- *d=0 alpha-trimmed mean filter reduces into mean filter.*
- *d= (mn-1)/2, the filter becomes median filter.*
- For other values of *d*, alpha-trimmed filter is useful in situations involving multiple type of noise, such as combination of salt-and-pepper and Gaussian noise.

Median filter for salt-and-pepper noise



a	b
c	d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Max and Min Filter

a b
c d

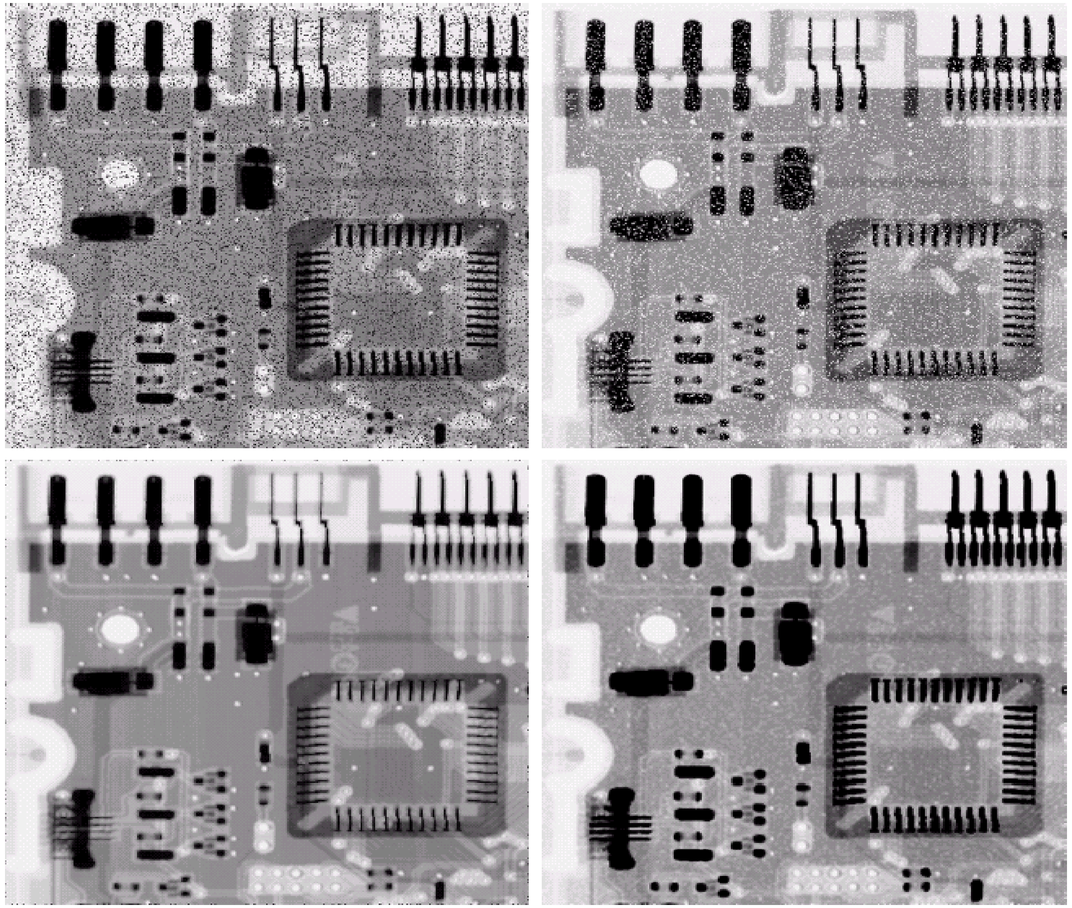
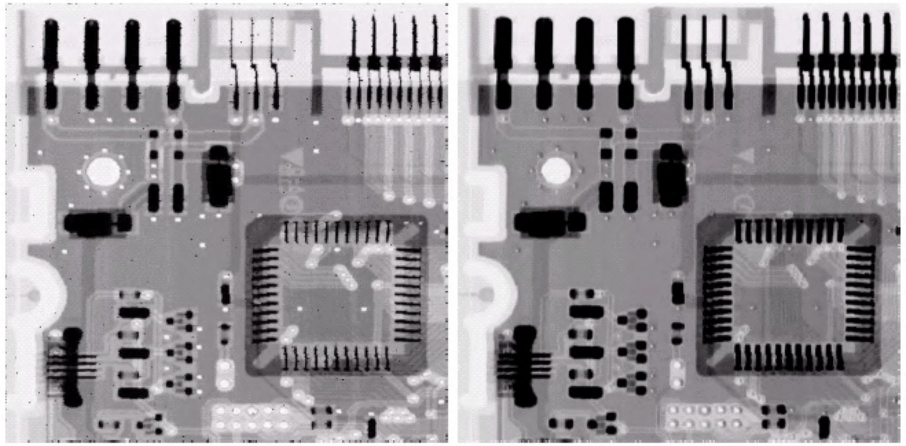


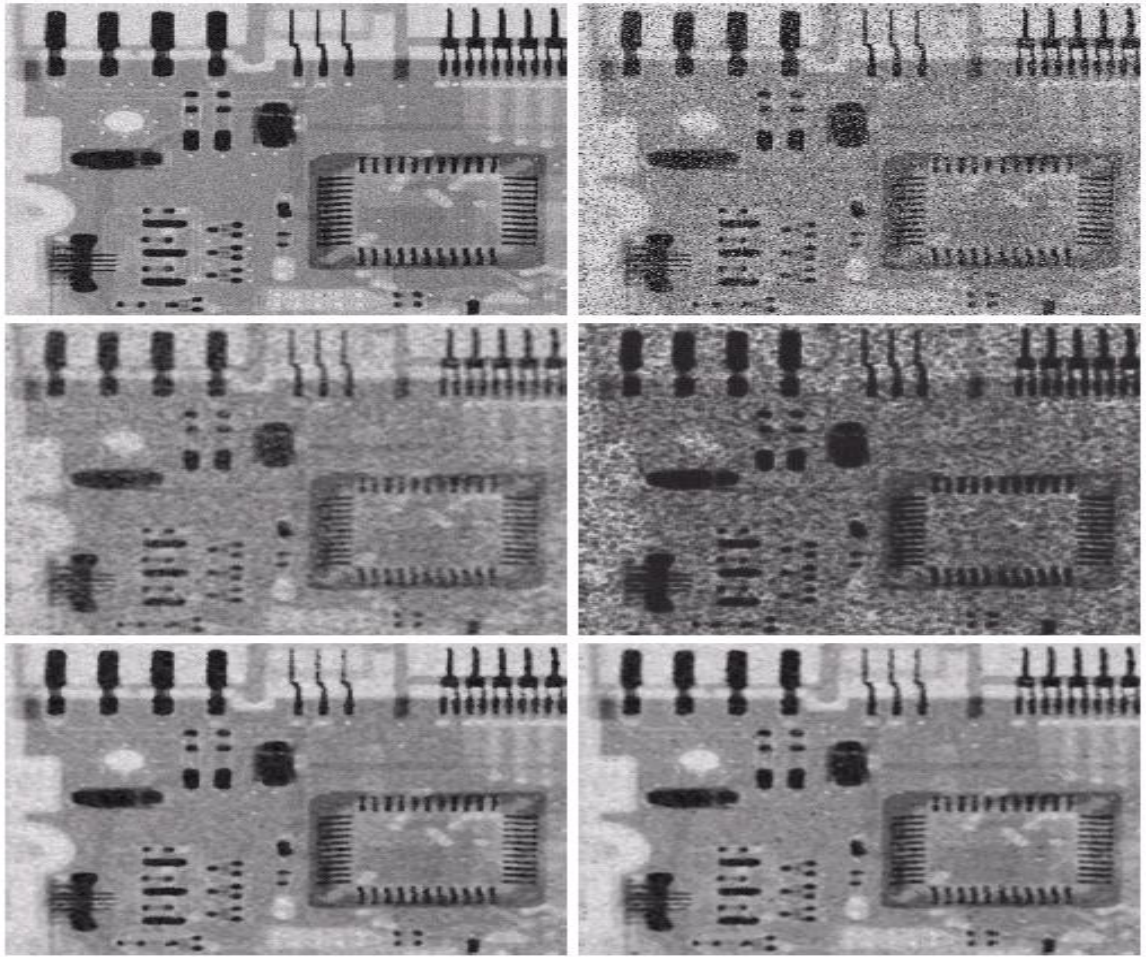
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Filter Effect on Mixed Noise



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

Suggested Readings

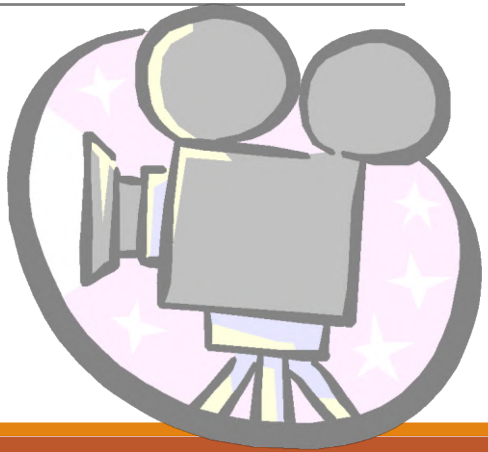
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Adaptive Filters
- Band-pass and band-reject filters
- Notch Filters

Model of the Image Degradation

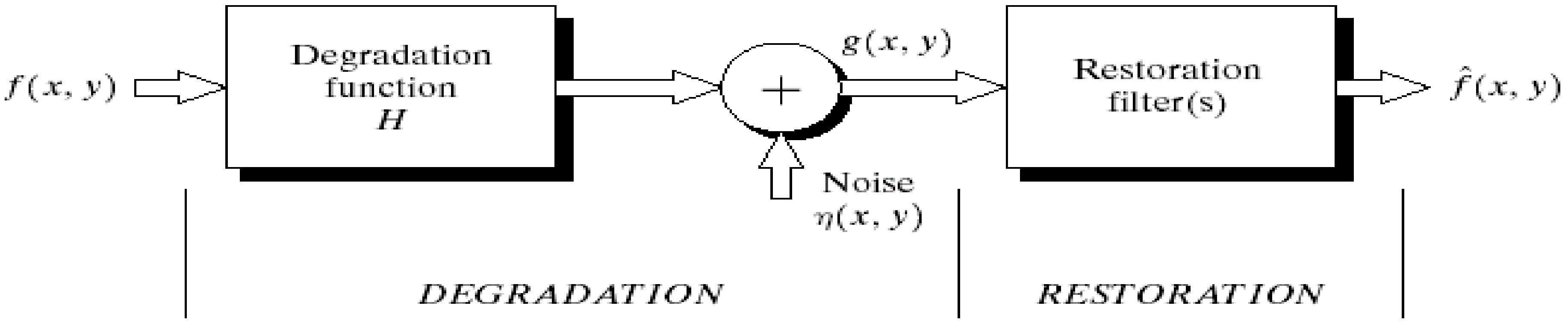


FIGURE 5.1 A model of the image degradation/ restoration process.

Adaptive Filters

- Filter behavior changes based on statistical characteristics of the image inside the filter region.

Adaptive, local noise reduction filter

- Simplest statistical measures of a random variable are its mean and variance

- Mean gives a measure of average gray level inside the filter region, and the variance gives a measure of average contrast in the filter region.

Adaptive Filter

Filter operate on a local region S_{xy}. The response of the filter at the point (x, y) is to be based on four points:

a) g(x, y), the value of the noisy image at (x, y).
b) σ_η^2 the variance of the noise corrupting f(x, y) to form g(x, y)
c) m_L, the local mean of the pixel in S_{xy}. ,
d) σ_L^2 The local variance of the pixel in S_{xy}.

- Requirement for adaptive filter

1. If noise variance, σ_η^2, is zero, the filter should return simply the value *g*(x,y). This is the trivial, zero-noise case in which *g*(x,y) is equal to *f*(x,y).

2. If local variance, σ_L^2 is high relative to noise variance σ_η^2, the filter should return a value close to *g*(x,y). High local variance typically is associated with edges, and these should be preserved.

3. If two variances are equal, we want the filter to return the arithmetic mean value of the pixels in filter region.

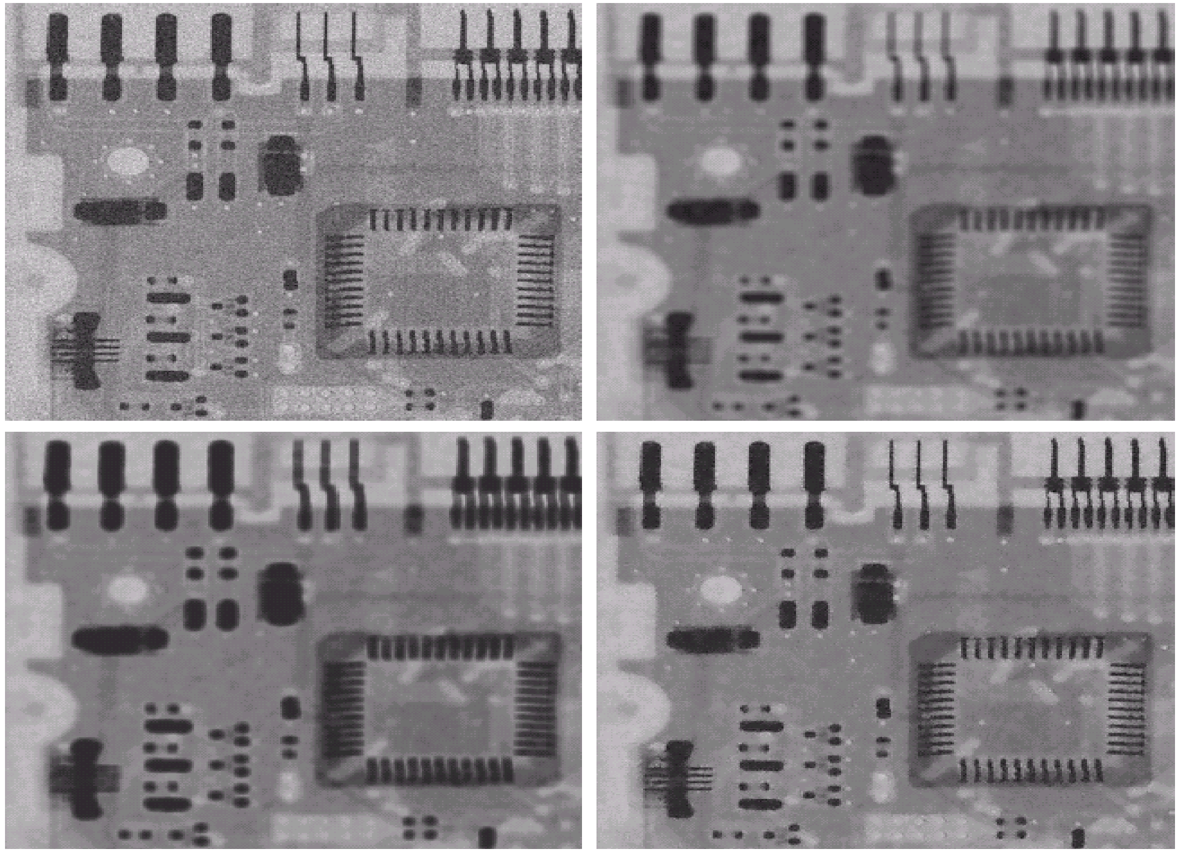
Adaptive Filters

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

- Quantity that needs to be know or estimated is the variance of the overall noise, σ_η^2

a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.



Adaptive Median Filter

- The median filter performs well as long as the spatial density of the impulse noise is not large (as a rule of thumb, P_a and P_b less than 0.2).

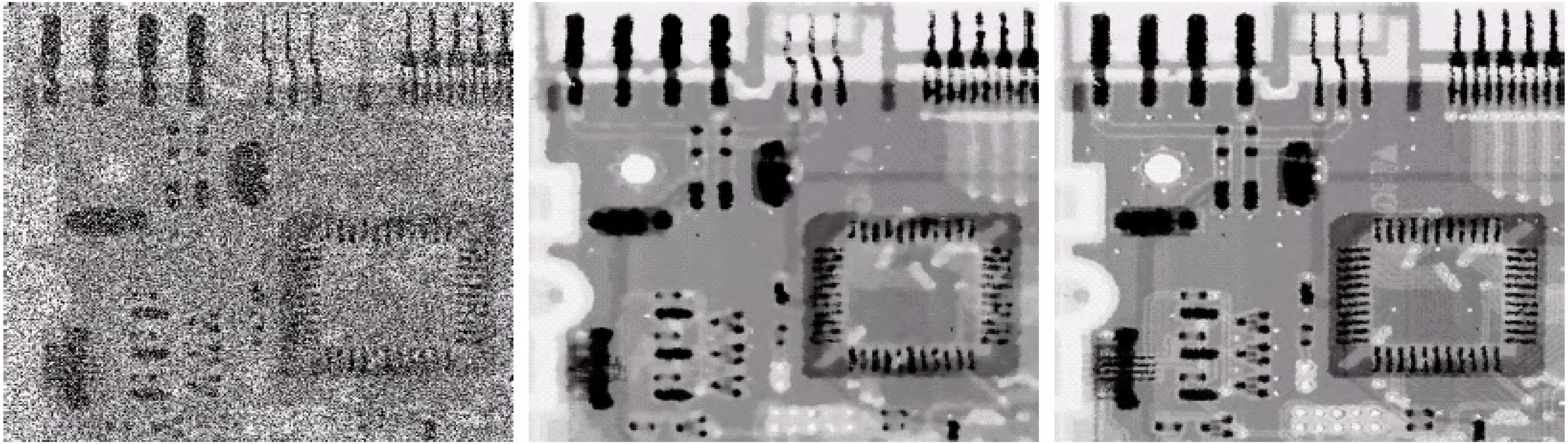
Z_{min} = minimum gray level value in filter region, W_{mn}.
Z_{max} = maximum gray level value in W_{mn}.
Z_{med} = median of gray level W_{mn}.
Z_{xy} = Gray level at coordinate (x,y).
W_{max}= maximum allowed size of W_{mn}.

Adaptive median filtering algorithm:

Level A: $A1 = Z_{med} - Z_{min}$; $A2 = Z_{med} - Z_{max}$
if A1 >0 AND A2<0, Go to level *B*
Else increase the window size
If window size <= W_{max} repeat level A
Else output z_{xy}

Level B: $B1 = Z_{xy} - Z_{min}$; $B2 = Z_{xy} - Z_{max}$
if B1 >0 AND B2<0, output z_{xy}
Else output z_{med}.

Adaptive Median Filtering



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

Periodic Noise

- Periodic noise is sinusoidal at multiple of a specific frequency and periodic in nature.

- It can be removed by Band-pass, Band reject and Notch filter.

Periodic Noise reduction by frequency domain Filtering

Band Pass filter: The objective of band pass filter is to pass limited range frequency while leaving the other.

Band Reject filter: The objective of band reject filter is to attenuate limited range frequency while leaving the other.

Periodic Noise reduction by frequency domain Filtering

Ideal Band Rejected Filter

$$H(u,v)=\begin{cases}1, & if\;\; D(u,v)<D_0-\dfrac{W}{2}\\[2mm] 0, & if\;\; D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\[2mm] 1, & if\;\; D(u,v)\ge D_0+\dfrac{W}{2}\end{cases}$$

Butterworth Band Rejected Filter

$$H(u,v)=\left\{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}\right\}^{-1}$$

Gaussian Band Rejected Filter

$$H(u,v)=1-\exp\left\{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^{2}\right\}$$

D(*u,v*) is the distance from the origin, *W* is the width of the band,
D_o is the radial center.

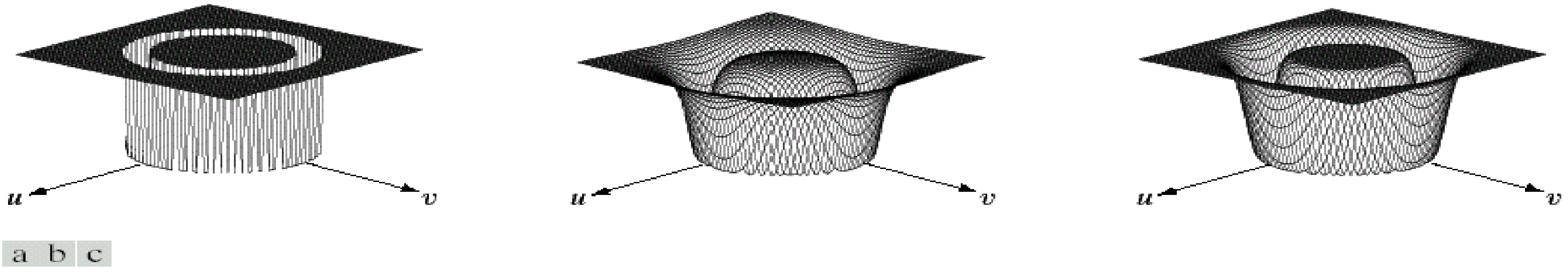
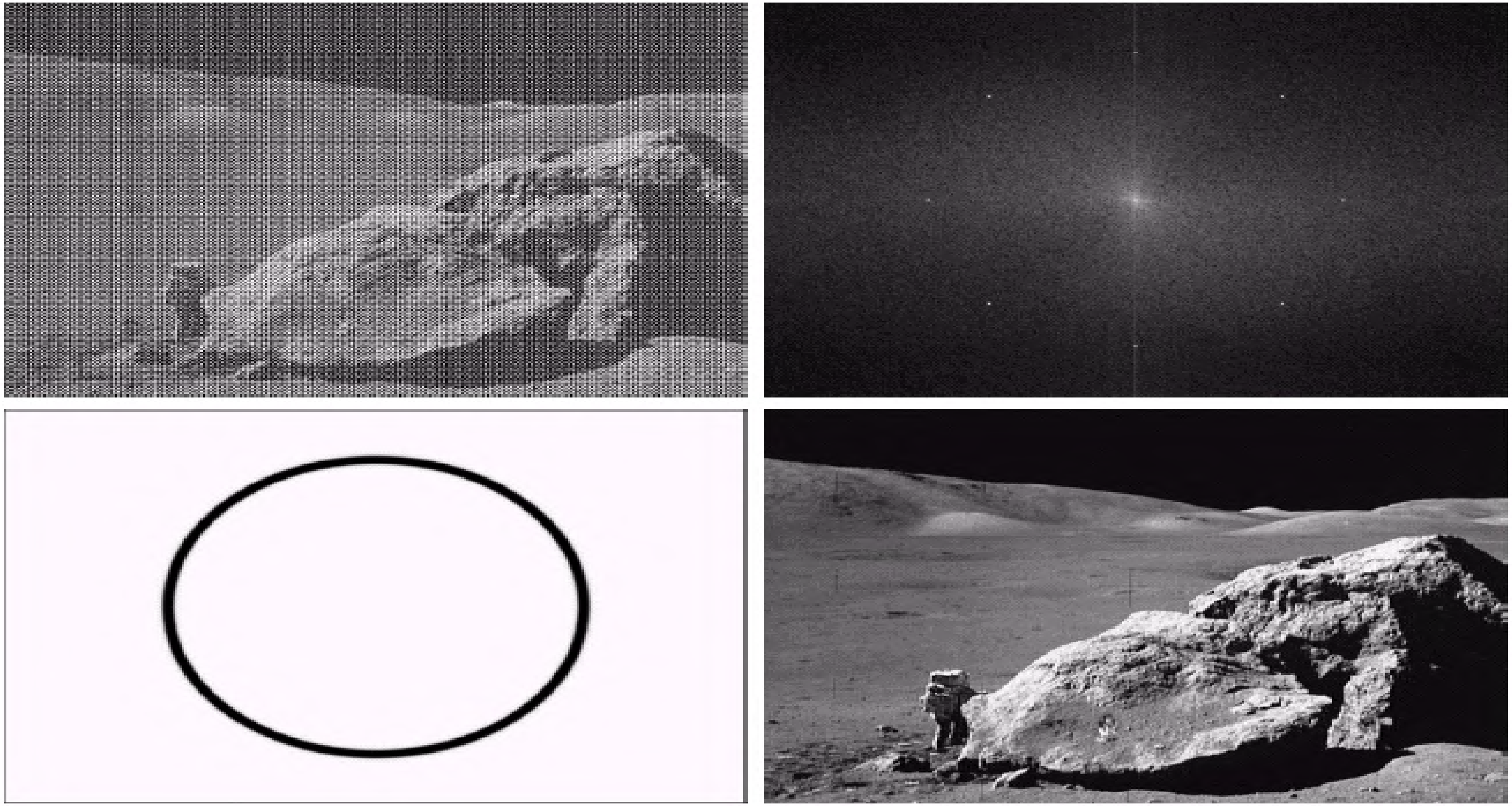


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Periodic Noise reduction by frequency domain Filtering



a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

- Image is heavily corrupted by sinusoidal noise of various frequency.
- Noise components lie on an approximate circle about the origin of the transform. Circularly symmetric band rejected filter is good choice.

Periodic Noise reduction by frequency domain Filtering

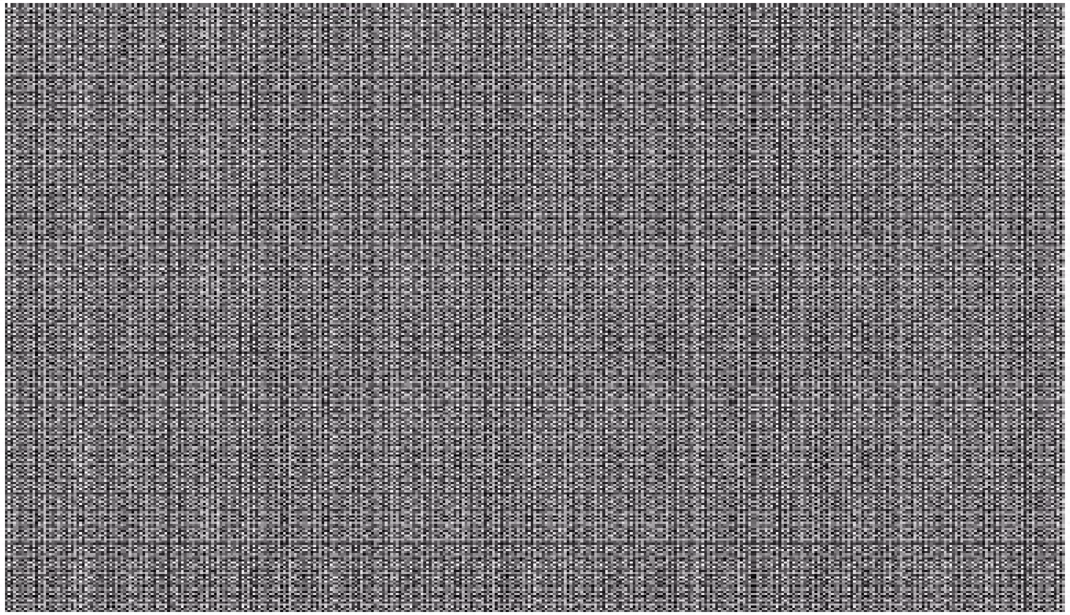
Band Pass Filter: passes frequency within a particular range

Opposite to band reject filter

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

- Band pass filtering generally removes the too much image detail.
- Band pass filtering is quite useful in isolating the effect on an image of selected frequency band.

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Notch Filters

A notch filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency. It is sepecial form of band reject filter.

Ideal Notch

$$\boxed{H(u,v)=\begin{cases}0, & if\;\; D_1(u,v)\le D_0(u,v)\;\; \text{or}\; D_2(u,v)\le D_0(u,v)\\ 1, & \text{Otherwise}\end{cases}}$$

$$D_1(u,v)=\left[\left(u-M/2-u_0\right)^2+\left(v-N/2-v_0\right)^2\right]^{1/2}$$

$$D_2(u,v)=\left[\left(u-M/2+u_0\right)^2+\left(v-N/2+v_0\right)^2\right]^{1/2}$$

Assume that center is shifted to the point (*M/2, N/2*)

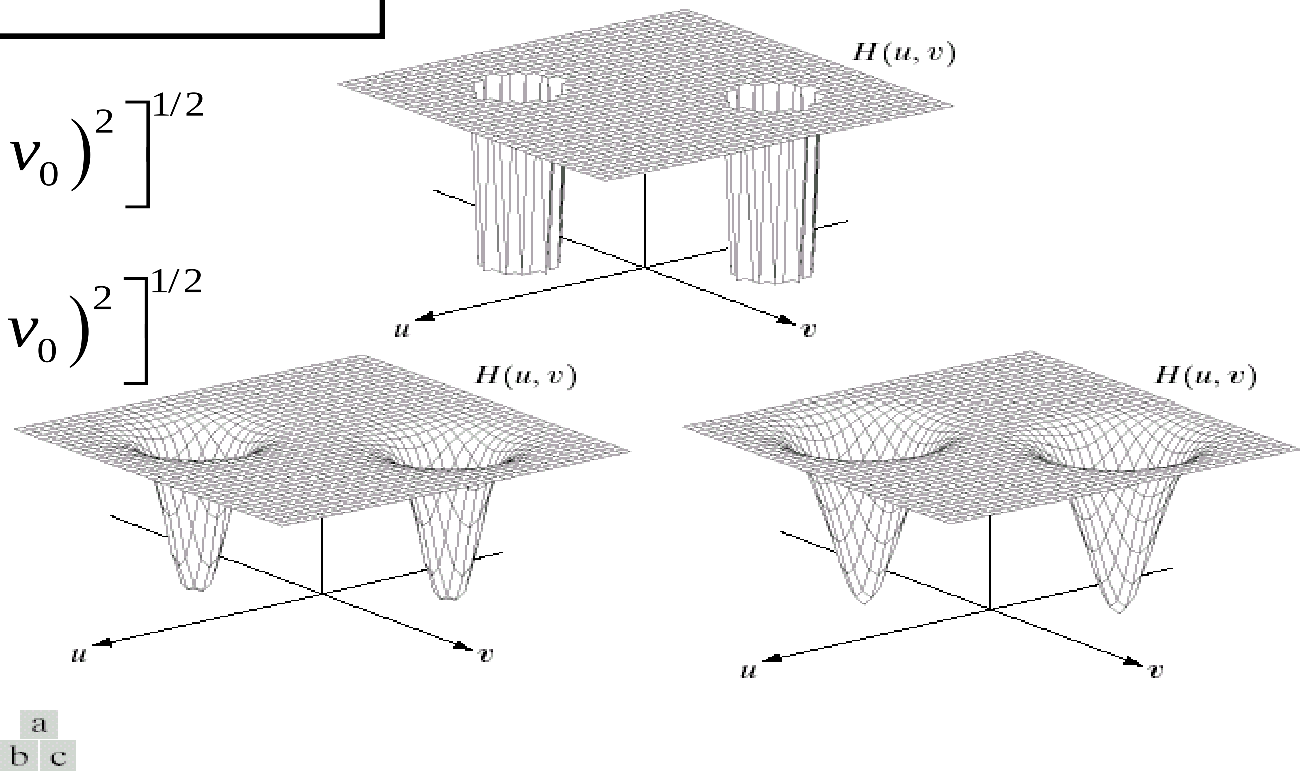


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Instead of removing the entire range of frequency,
It removes only selected frequency component

Notch Filters

Butterworth Notch Reject filter

$$H(u,v)=\left\{1+\left[\frac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^n\right\}^{-1}$$

Gaussian Notch Reject filter

$$H(u,v)=1-\exp\left\{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]\right\}$$

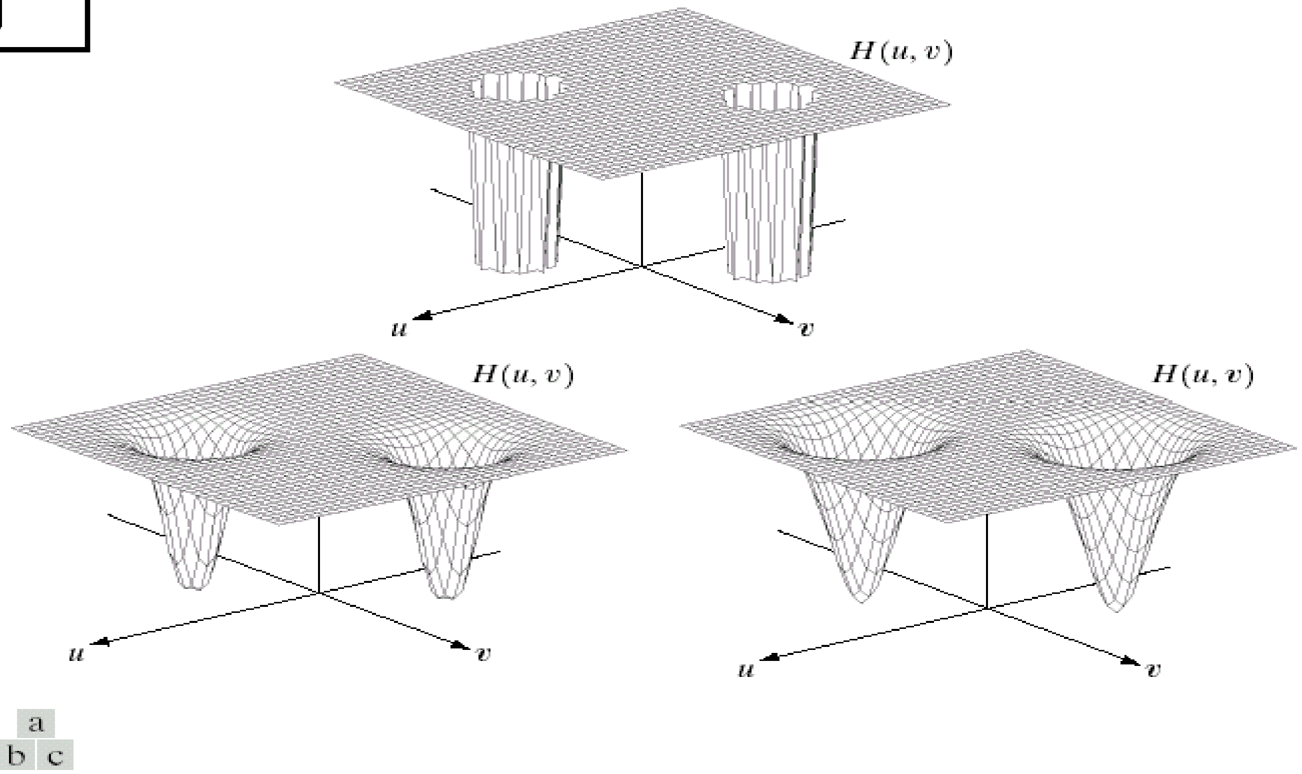
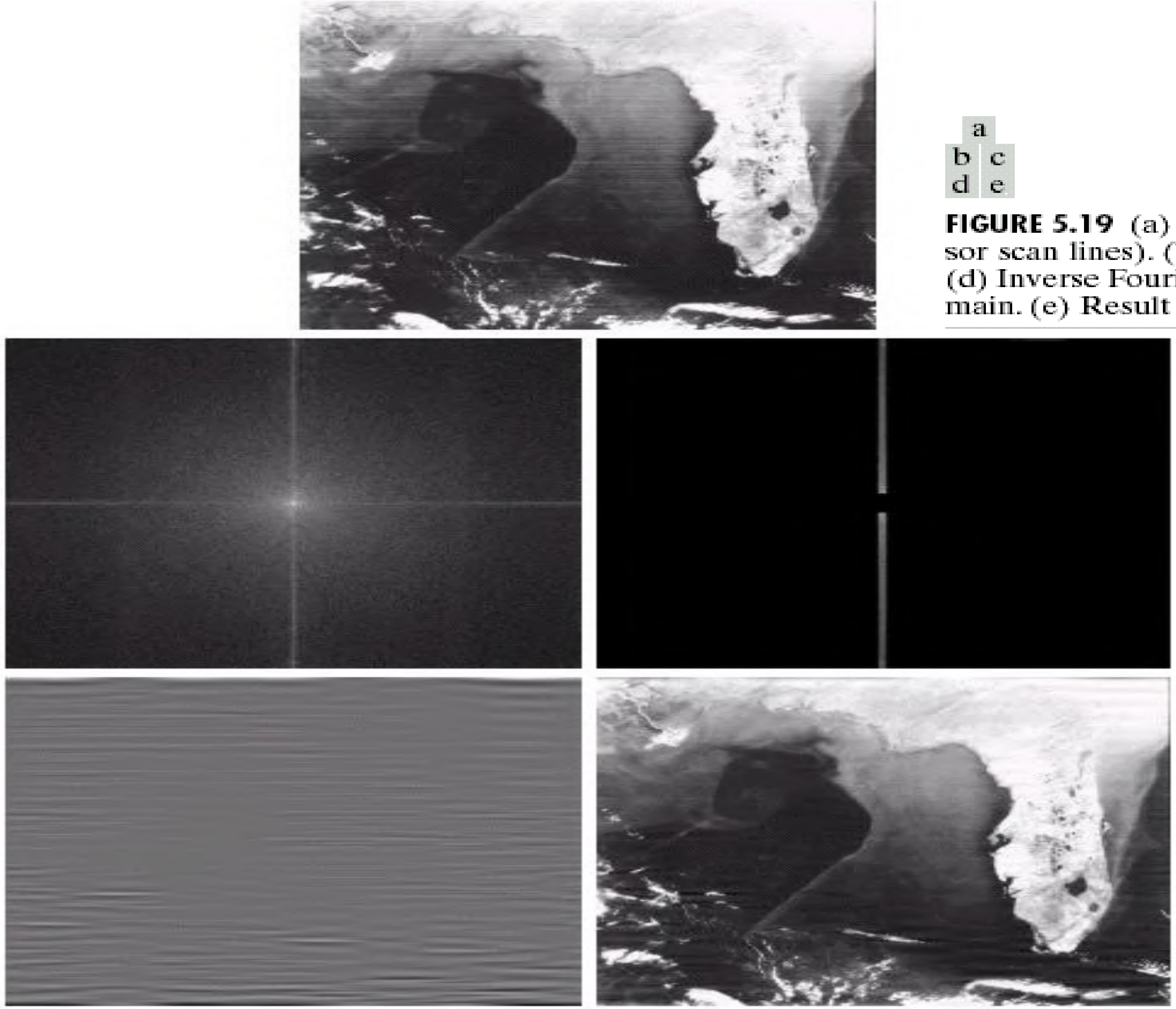


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

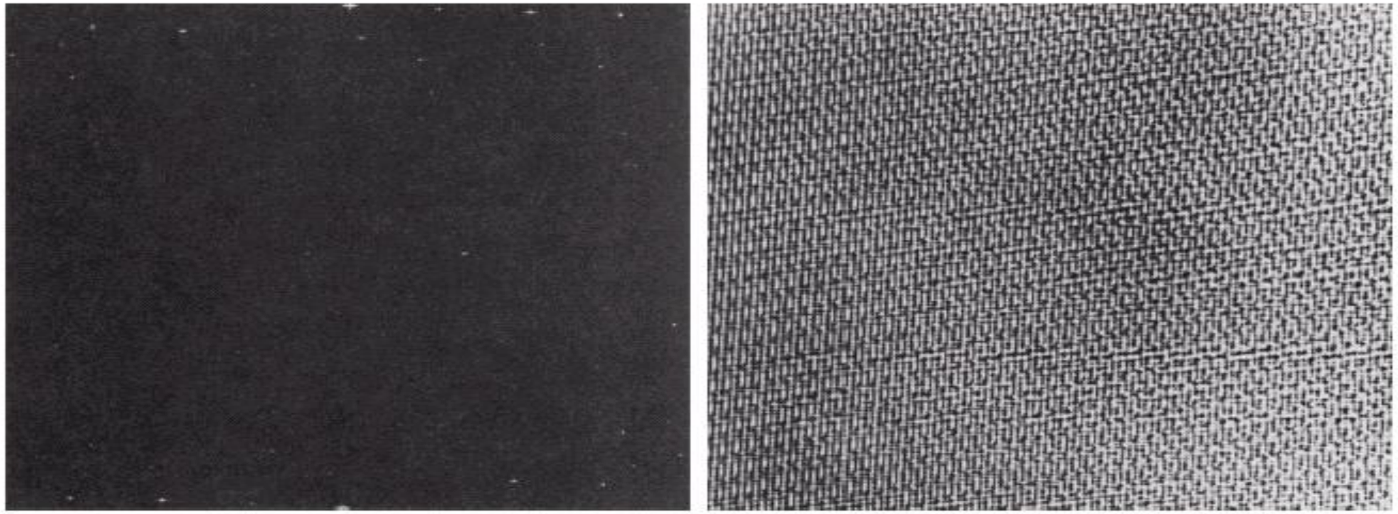
Notch Filters



a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Notch Filters



a b

FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

Suggested Readings

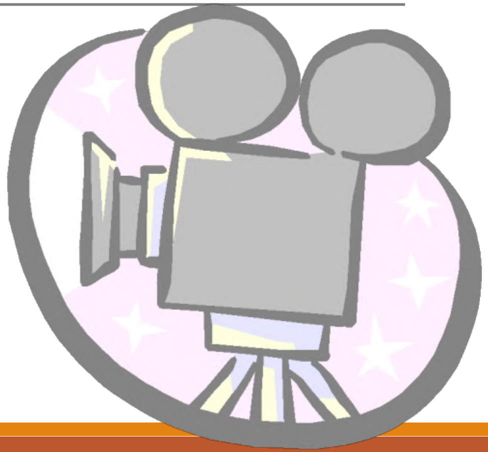
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image reconstruction from projection
  - The reconstruction problem
  - Principal of Computed Tomography
  - The Radon Transformation
  - Fourier Slice Theorem
  - Reconstruction using parallel beam Filter back projection

CT Scan

Image Reconstruction problem

Consider a single object on a uniform background (suppose that this is a cross section of 3D region of a human body).
Background represents soft, uniform tissue and the object is also uniform but with higher absorption characteristics.

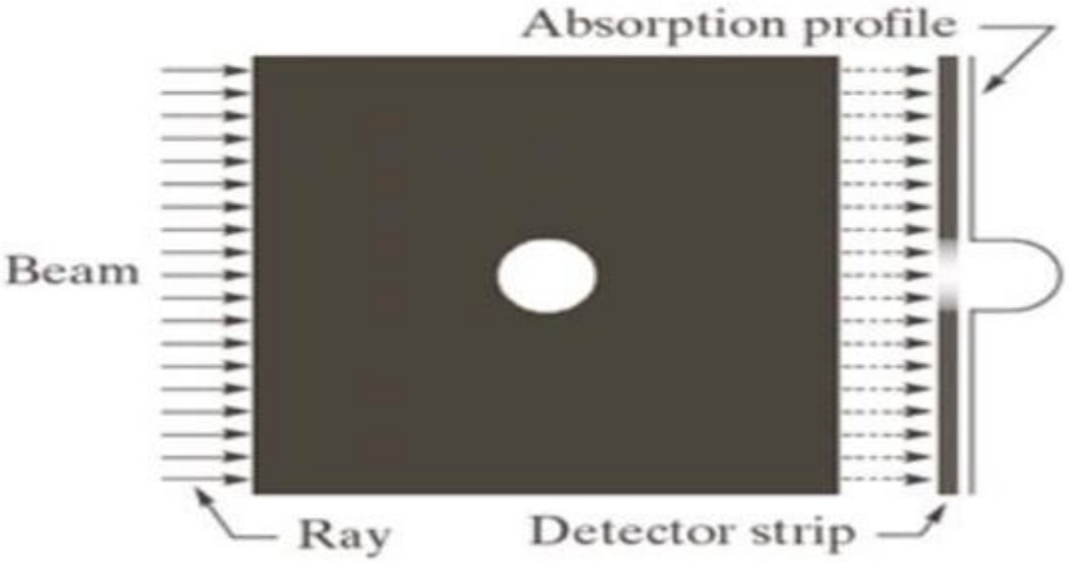


Image Reconstruction problem

A beam of X-rays is emitted and part of it is absorbed by the object.
The energy of absorption is detected by a set of detectors.
The collected information is the absorption signal.

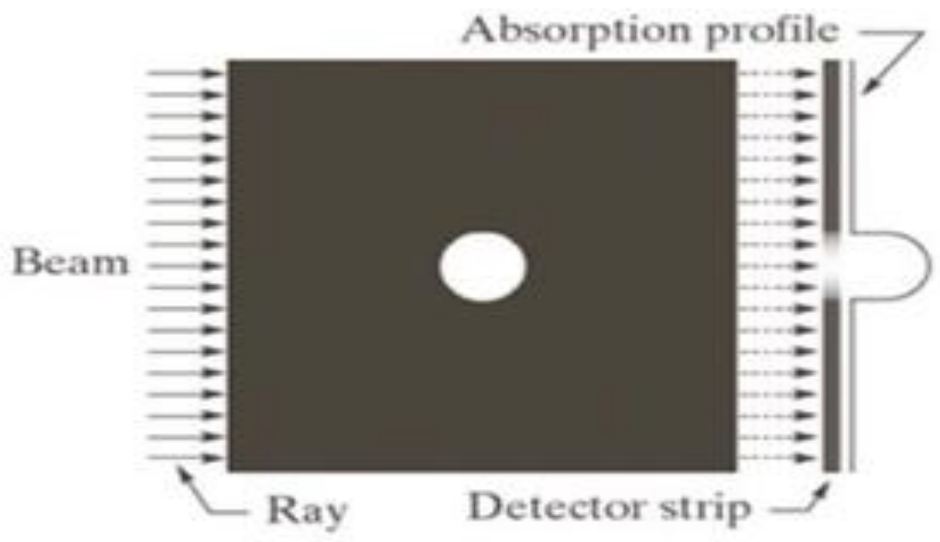


Image Reconstruction problem (Cont…)

We have no means of determining the number of objects from a single projection.

We now rotate the position of the source-detector pair and obtain another 1D signal.

We repeat the procedure and add the signals from the previous back-projections.

We can now tell that the object of interest is located at the central square.

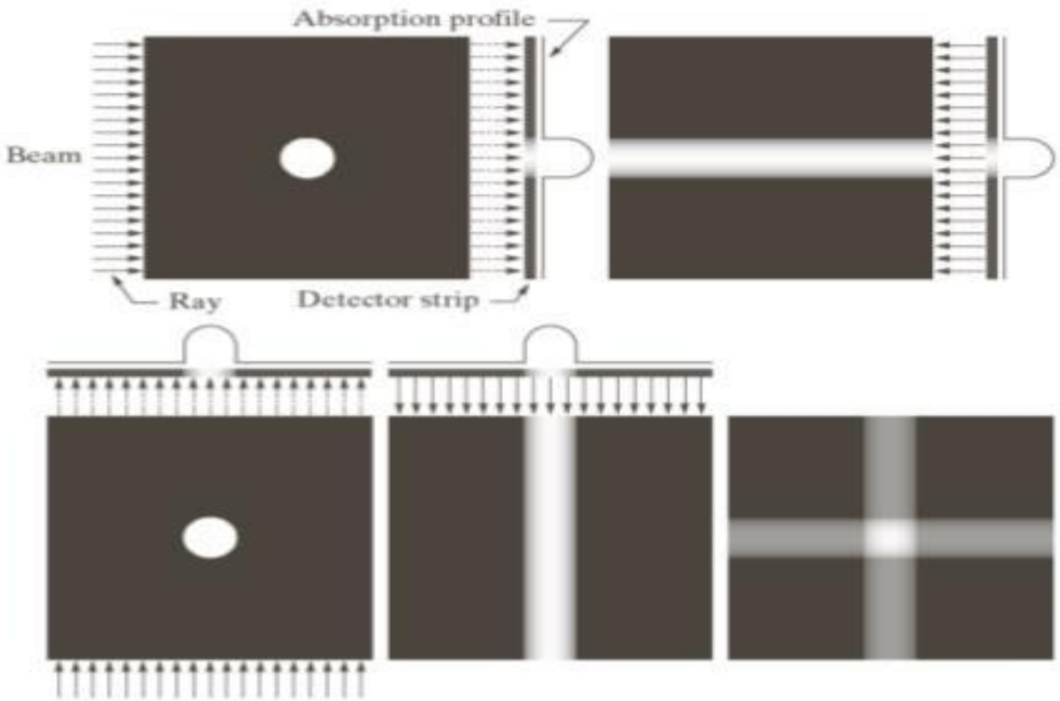
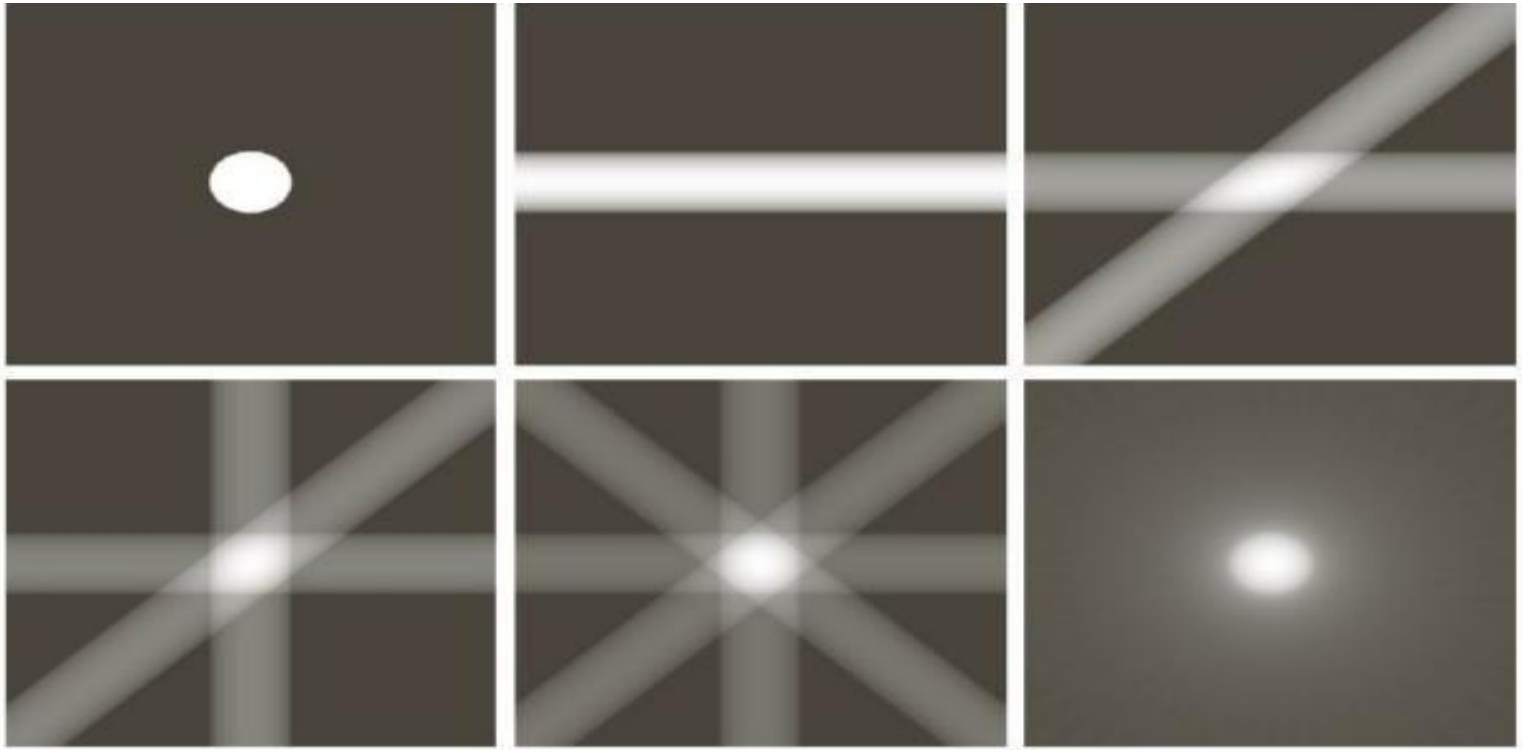


Image Reconstruction problem (Cont…)

a b c
d e f

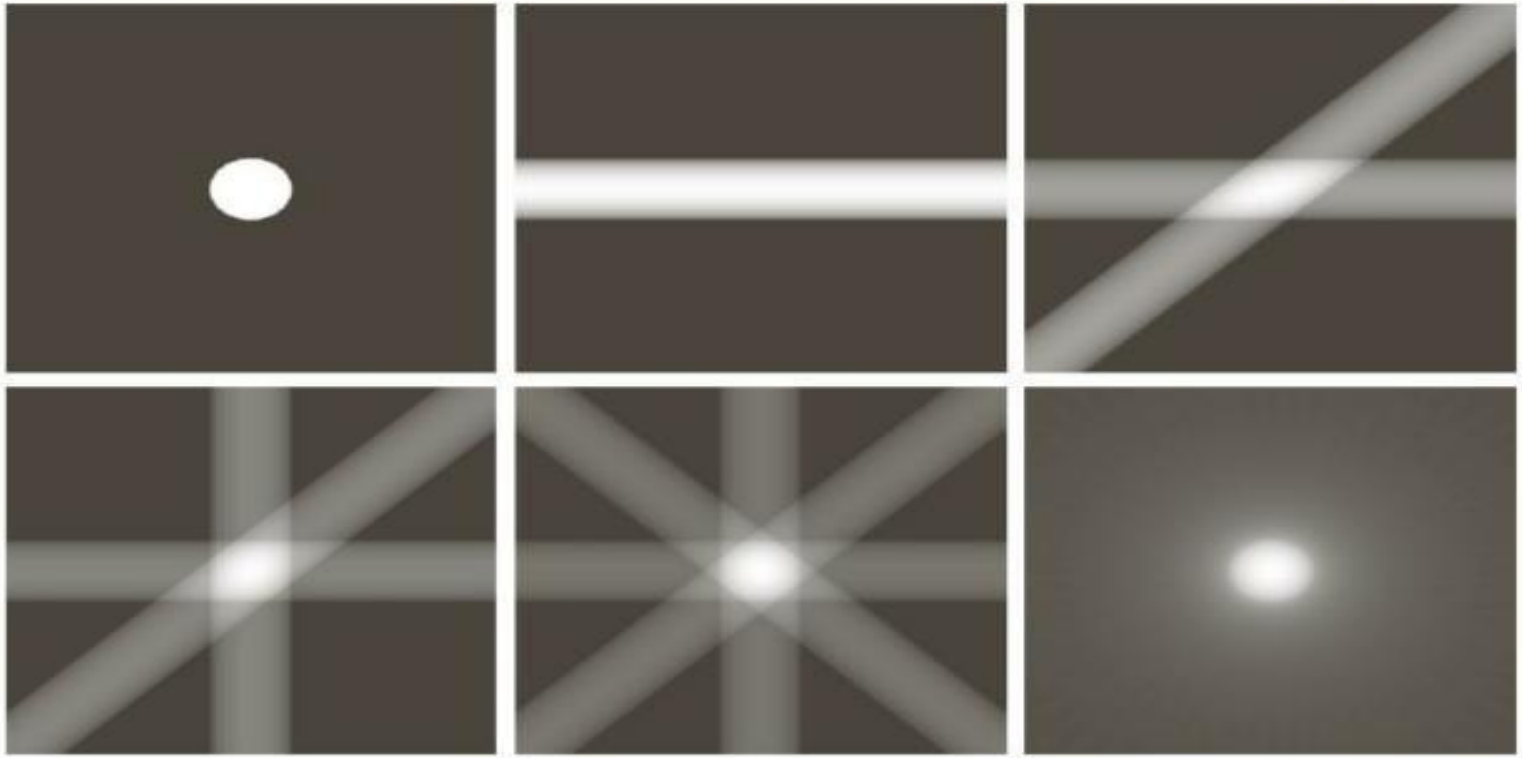
FIGURE 5.33 (a) Same as Fig. 5.32(a). (b)–(e) Reconstruction using 1, 2, 3, and 4 backprojections 45° apart. (f) Reconstruction with 32 backprojections 5.625° apart (note the blurring).



By taking more projections:

- the form of the object becomes clearer as brighter regions will dominate the result
- back-projections with few interactions with the object will fade into the background.

Image Reconstruction problem (Cont…)

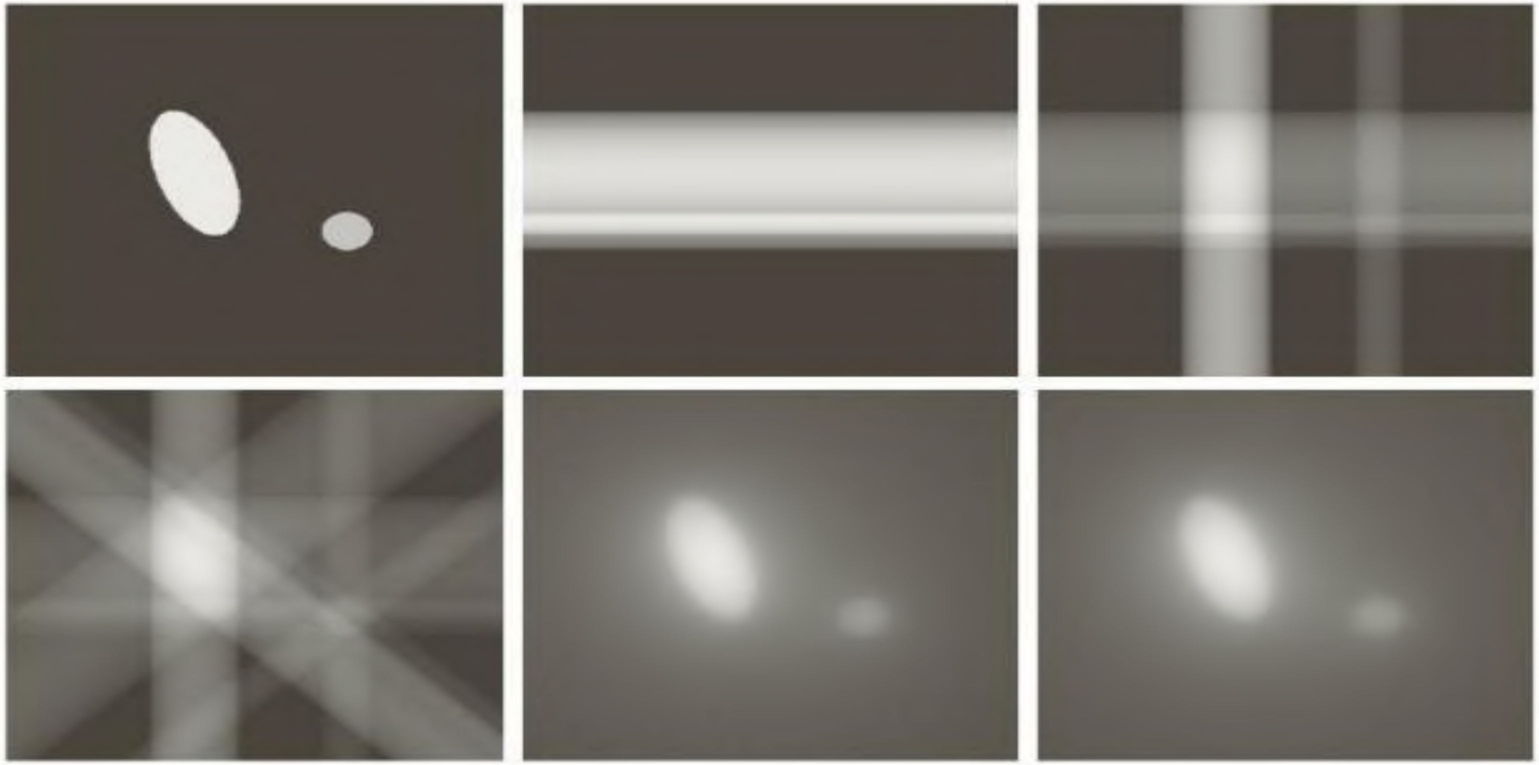


a b c
d e f

FIGURE 5.33
(a) Same as Fig. 5.32(a).
(b)–(e) Reconstruction using 1, 2, 3, and 4 backprojections 45° apart.
(f) Reconstruction with 32 backprojections 5.625° apart (note the blurring).

- The image is blurred. Important problem!
- We only consider projections from 0 to 180 degrees as projections differing 180 degrees are mirror images of each other.

Image Reconstruction problem (Cont…)



a b c
d e f

FIGURE 5.34 (a) A region with two objects. (b)–(d) Reconstruction using 1, 2, and 4 backprojections 45° apart. (e) Reconstruction with 32 backprojections 5.625° apart. (f) Reconstruction with 64 backprojections 2.8125° apart.

Principles of Computed Tomography

The goal of CT is to obtain a 3D representation of the internal structure of an object by X-raying it from many different directions.

Imagine the traditional chest X-ray obtained by different directions. The image is the 2D equivalent of a line projections.

Back-projecting the image would result in a 3D volume of the chest cavity.

Principles of Computed Tomography

CT gets the same information by generating slices through the body.

A 3D representation is then obtained by stacking the slices.

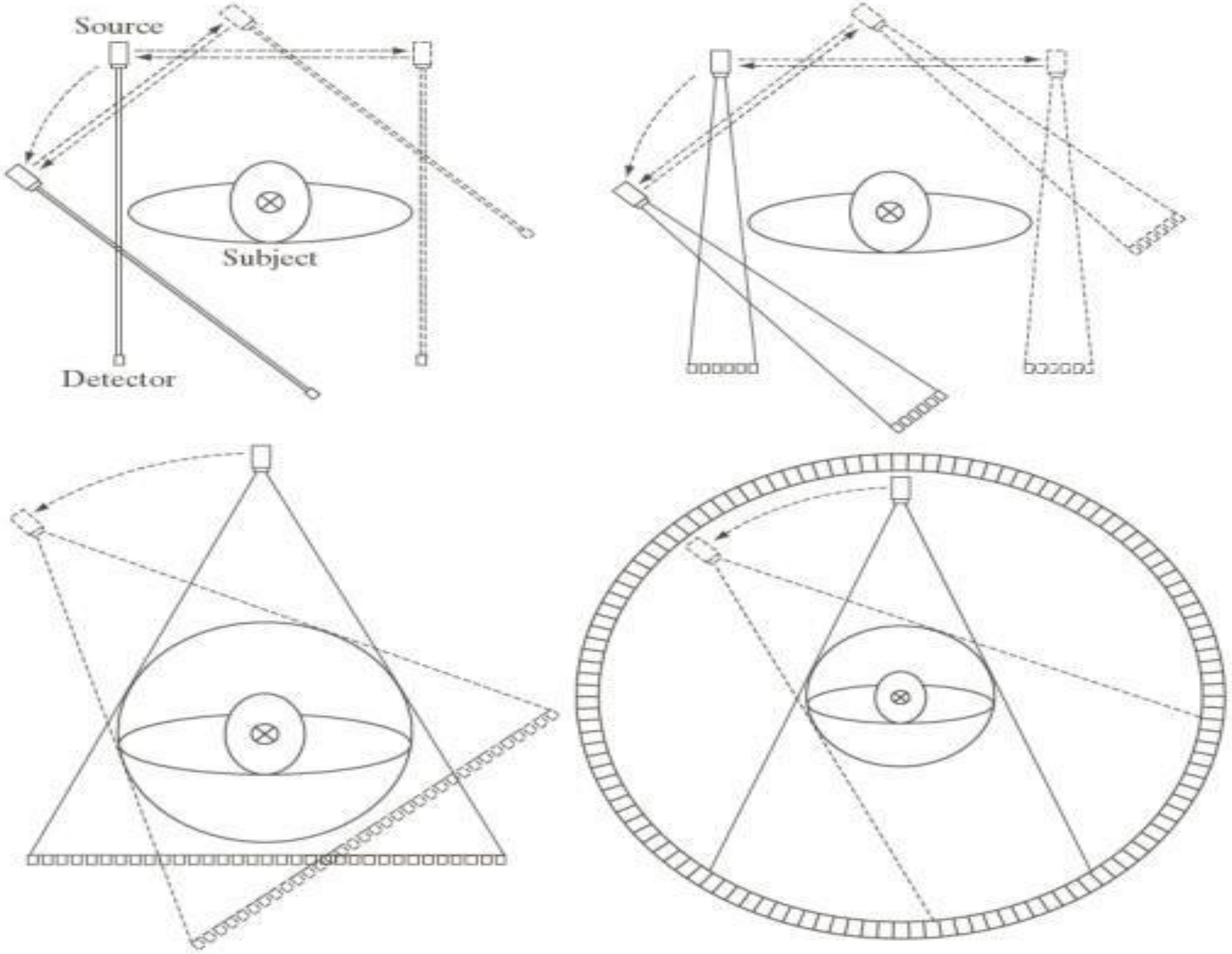
More economical due to fewer detectors.

Computational burden and dosage is reduced.

Theory developed in 1917 by J. Radon.

Application developed in 1964 by A. M. Cormack and G. N. Hounsfield independently. They shared the Nobel prize in Medicine in 1979.

Principles of Computed Tomography



a	b
c	d

FIGURE 5.35 Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.

The Radon Transform

A straight line in Cartesian coordinates may be described by its *slope-intercept* form:

$$y = ax + b$$

or by its *normal representation*:

$$x\cos\theta + y\sin\theta = \rho$$

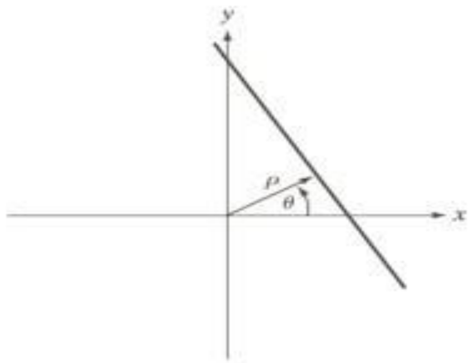
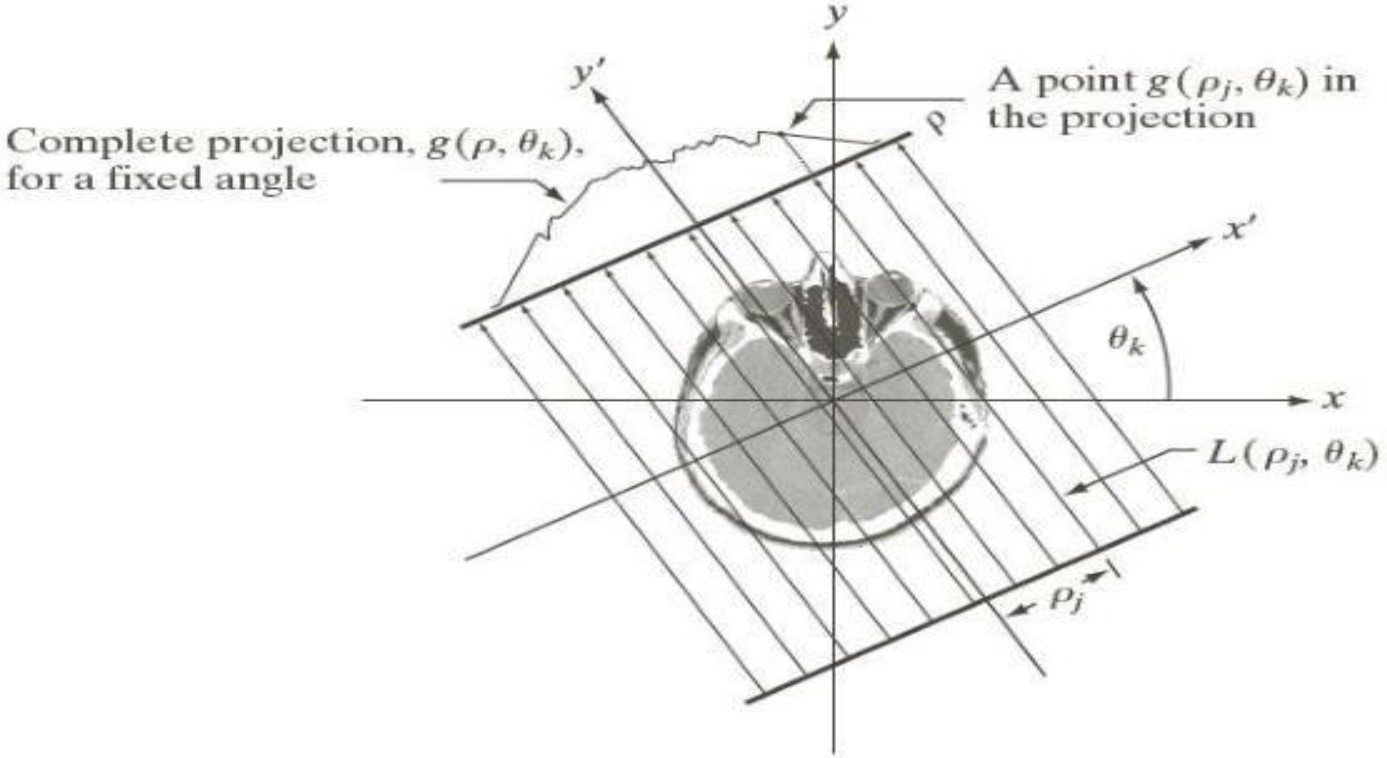


FIGURE 5.36 Normal representation of a straight line.

The Radon Transform (Cont…)

The projection of a parallel-ray beam may be modelled by a set of such lines.

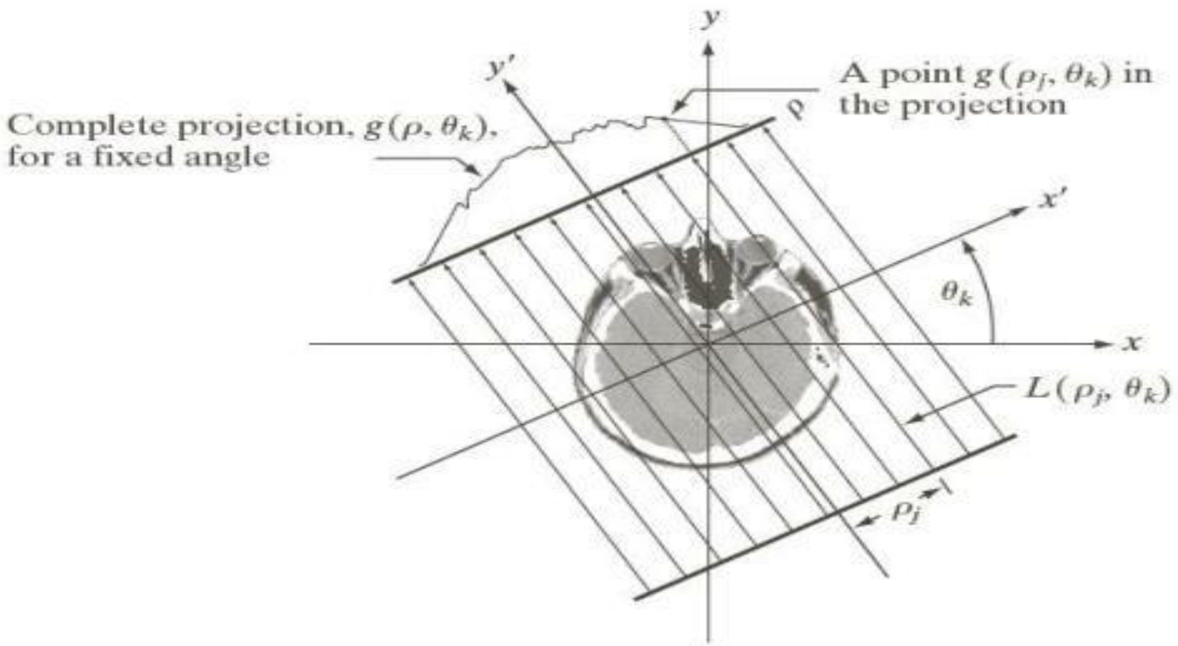
An arbitrary point (ρ_j, θ_k) in the projection signal is given by the ray-sum along the line $x\cos\theta_k + y\sin\theta_k = \rho_j$.



The Radon Transform (Cont…)

The ray-sum is a line integral:

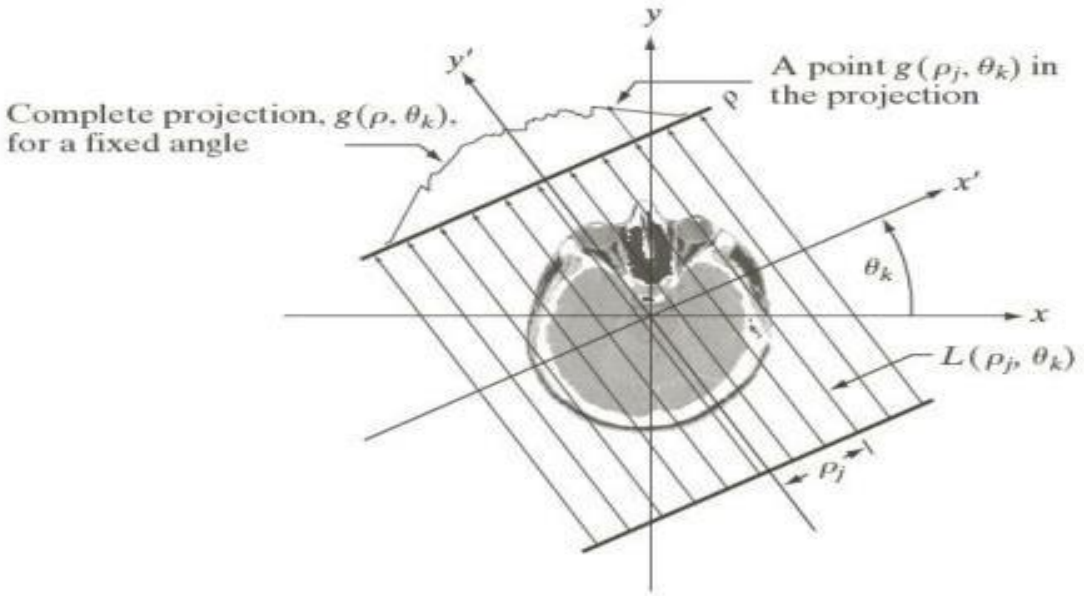
$$g(\rho_j, \theta_k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\delta(x\cos\theta_k + y\sin\theta_k - \rho_j)dxdy$$



The Radon Transform (Cont...)

For all values of ρ and θ we obtain the Radon transform:

$$g(\rho,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$



The Radon Transform (Cont…)

The representation of the Radon transform $g(\rho,\theta)$ as an image with ρ and θ as coordinates is called a *sinogram*.

It is very difficult to interpret a sinogram.

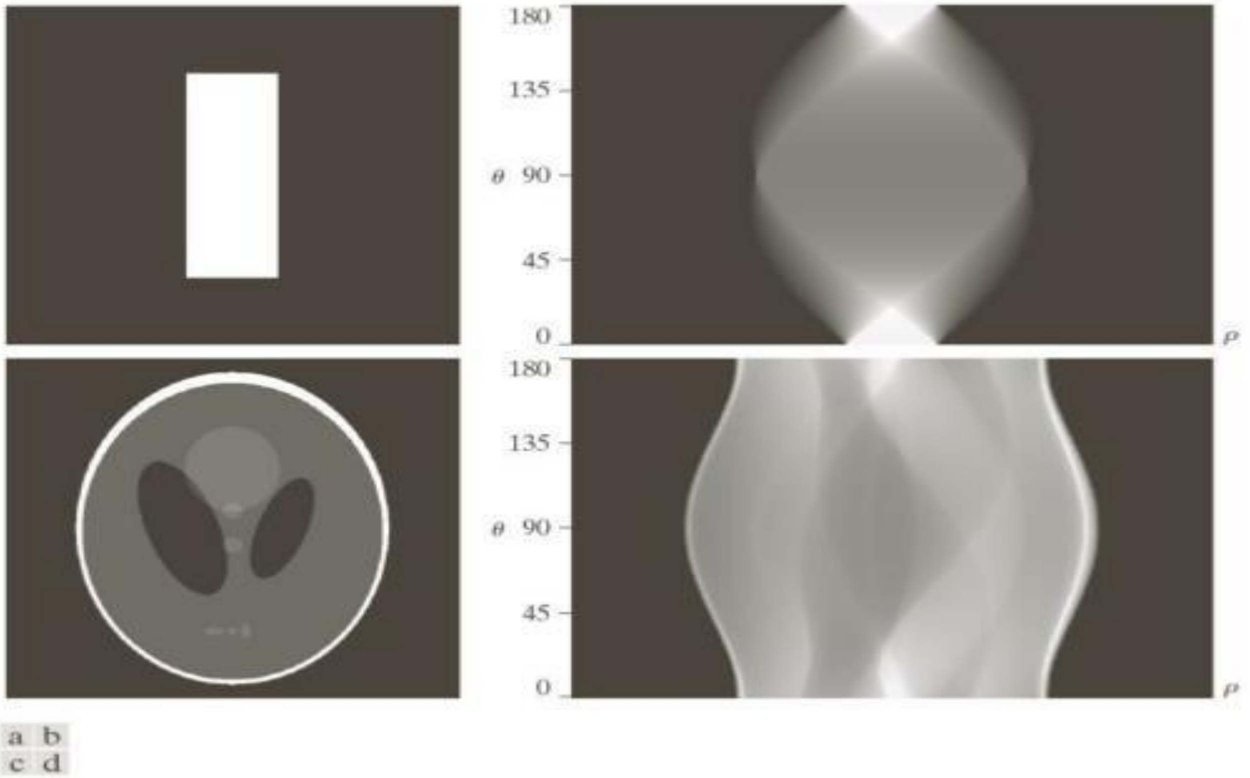


FIGURE 5.39 Two images and their sinograms (Radon transforms). Each row of a sinogram is a projection along the corresponding angle on the vertical axis. Image (c) is called the *Shepp-Logan phantom*. In its original form, the contrast of the phantom is quite low. It is shown enhanced here to facilitate viewing.

The Radon Transform (Cont…)

The objective of CT is to obtain a 3D representation of a volume from its projections.

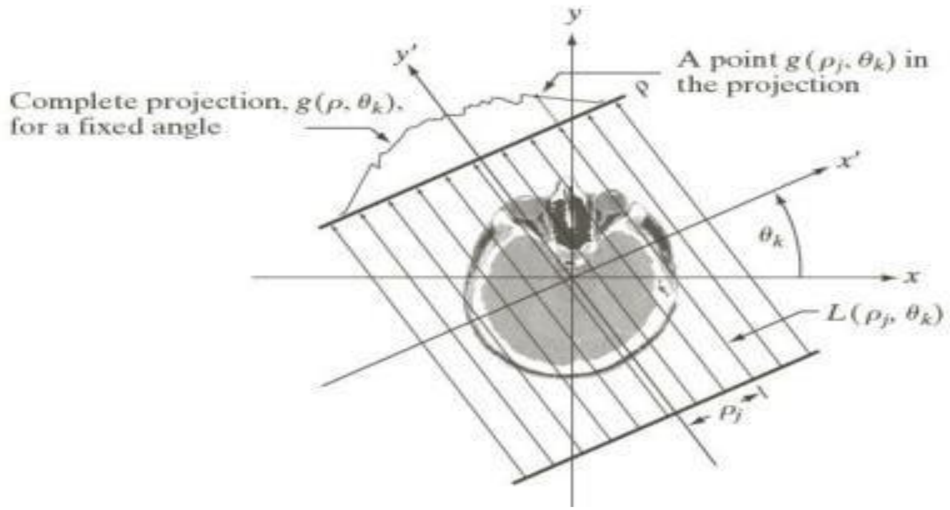
The approach is to back-project each projection and sum all the back-projections to generate a slice.

Stacking all the slices produces a 3D volume.

We will now describe the back-projection operation mathematically.

The Radon Transform (Cont…)

For a fixed rotation angle θ_k, and a fixed distance ρ_j, back-projecting the value of the projection $g(\rho_j,\theta_k)$ is equivalent to copying the value $g(\rho_j,\theta_k)$ to the image pixels belonging to the line $x\cos\theta_k + y\sin\theta_k = \rho_j$.



The Radon Transform (Cont…)

Repeating the process for all values of ρ_j, having a fixed angle θ_k results in the following expression for the image values:

$$f_{\theta_k}(x, y) = g(\rho, \theta_k) = g(x\cos\theta_k + y\sin\theta_k, \theta_k)$$

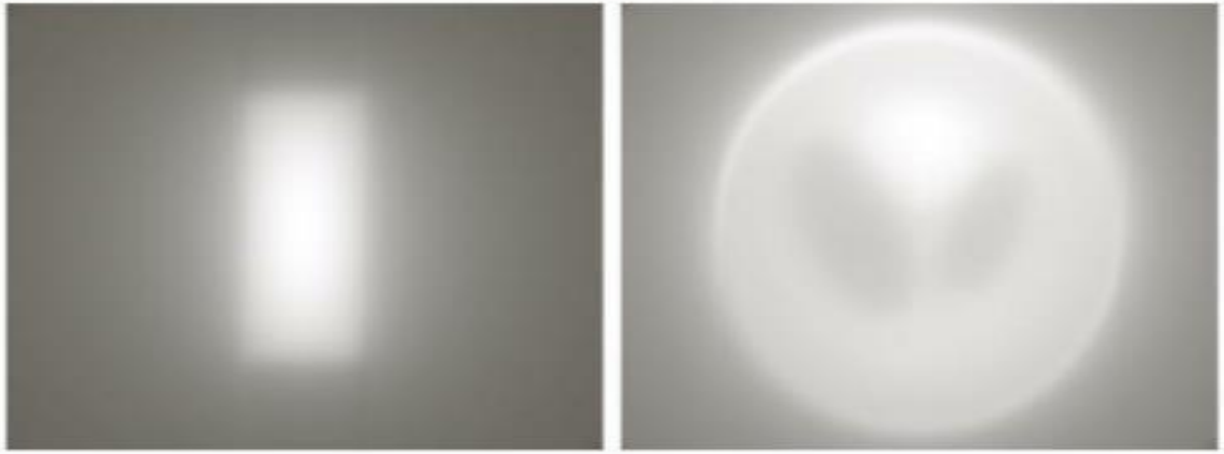
This equation holds for every angle θ:

$$f_{\theta}(x, y) = g(\rho, \theta) = g(x\cos\theta + y\sin\theta, \theta)$$

The Radon Transform (Cont…)

The final image is formed by integrating over all the back-projected images:

$$f(x, y) = \int_0^\pi f_\theta(x, y)\, d\theta$$



Back-projection provides blurred images. We will reformulate the process to eliminate blurring.

The Fourier Slice Theorem

The *Fourier-slice theorem* or the *central slice theorem* relates the 1D Fourier transform of a projection with the 2D Fourier transform of the region of the image from which the projection was obtained.

It is the basis of image reconstruction methods.

The Fourier Slice Theorem (Cont…)

Let the 1D F.T. of a projection with respect to ρ (at a given angle) be:

$$G(\omega,\theta) = \int_{-\infty}^{+\infty} g(\rho,\theta)\, e^{-j2\pi\omega\rho}\, d\rho$$

Substituting the projection $g(\rho,\theta)$ by the ray-sum:

$$G(\omega,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy\, e^{-j2\pi\omega\rho}\, d\rho$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\left[\int_{-\infty}^{+\infty} \delta(x\cos\theta + y\sin\theta - \rho)e^{-j2\pi\omega\rho}\, d\rho\right]dxdy$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi\omega(x\cos\theta + y\sin\theta)}\, dxdy$$

The Fourier Slice Theorem (Cont…)

$$G(\omega,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi\omega(x\cos\theta + y\sin\theta)}\,dxdy$$

Let now $u=\omega\cos\theta$ and $v=\omega\sin\theta$:

$$G(\omega,\theta) = \left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dxdy\right]_{u=\omega\cos\theta, v=\omega\sin\theta}$$

which is the 2D F.T. of the image $f(x,y)$ evaluated at the indicated frequencies u,v:

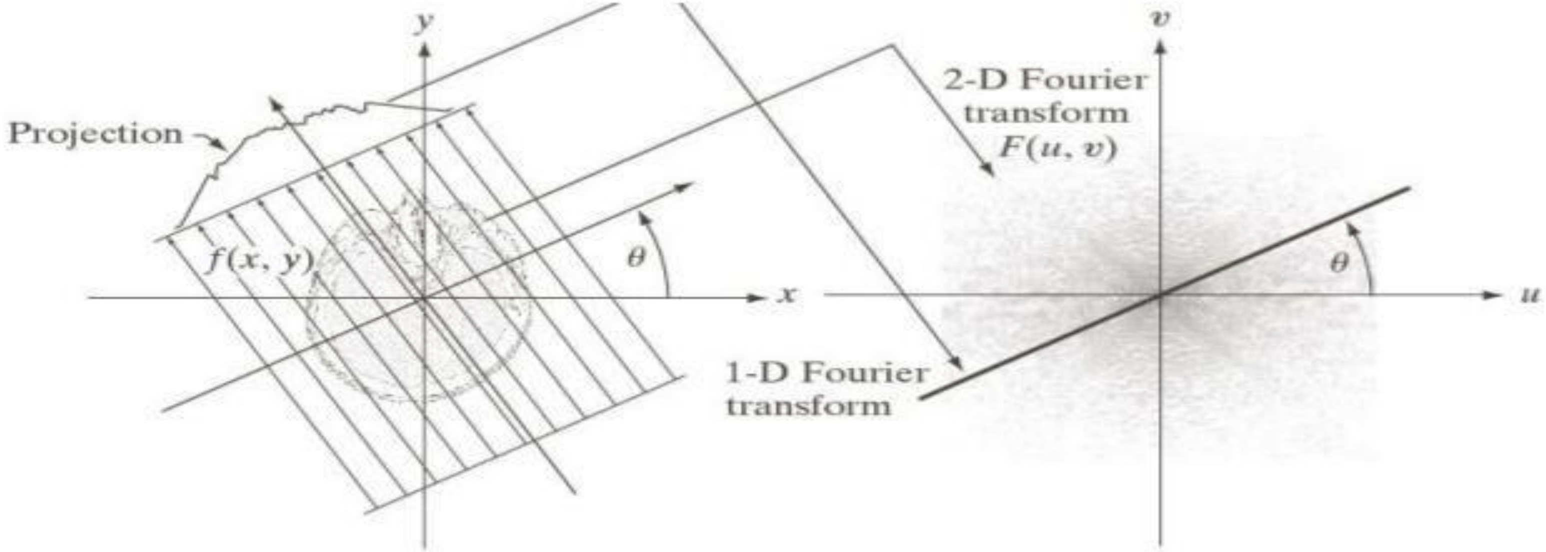
$$G(\omega,\theta) = \left[F(u,v)\right]_{u=\omega\cos\theta, v=\omega\sin\theta} = F(\omega\cos\theta, \omega\sin\theta)$$

The Fourier Slice Theorem (Cont…)

The resulting equation $G(\omega,\theta) = F(\omega\cos\theta, \omega\sin\theta)$

is known as the Fourier-slice theorem.

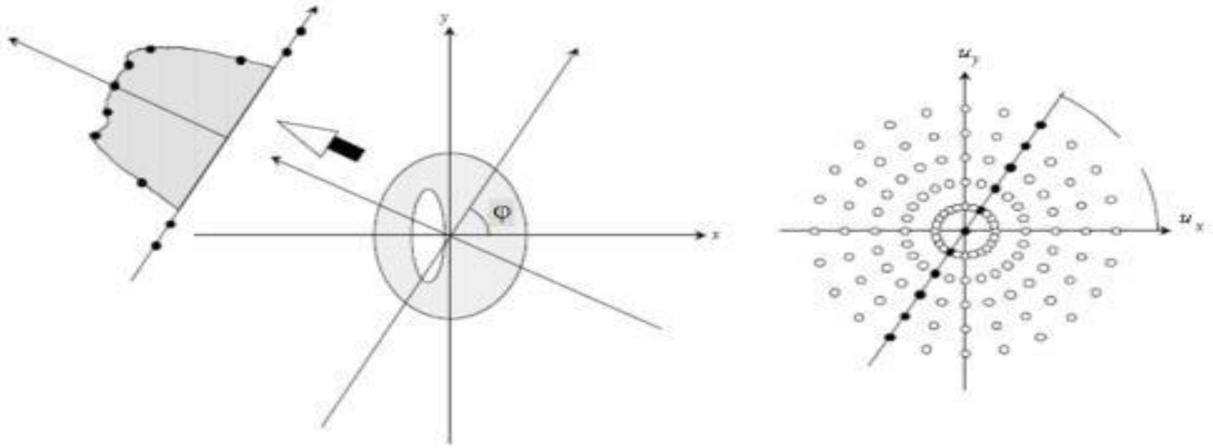
It states that the 1D F.T. of a projection (at a given angle θ) is a slice of the 2D F.T. of the image.



The Fourier Slice Theorem (Cont…)

We could obtain $f(x,y)$ by evaluating the F.T. of every projection and inverting them.

However, this procedure needs irregular interpolation which introduces inaccuracies.



Reconstruction using Parallel Beam filtered back projection

The 2D inverse Fourier transform of $F(u,v)$ is

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

Letting $u=\omega\cos\theta$ and $v=\omega\sin\theta$ then the differential

$$du\,dv = \omega\,d\omega\,d\theta$$

and

$$f(x,y) = \int_{0}^{2\pi}\int_{0}^{+\infty} F(\omega\cos\theta,\omega\sin\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}\omega\,d\omega\,d\theta$$

Reconstruction using Parallel Beam filtered back projection (Cont…)

Using the Fourier-slice theorem:

$$f(x, y) = \int_0^{2\pi} \int_0^{+\infty} G(\omega, \theta) e^{j2\pi\omega(x\cos\theta + y\sin\theta)} \omega \, d\omega \, d\theta$$

With some manipulation

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega(x\cos\theta + y\sin\theta)} \, d\omega \, d\theta$$

The term $x\cos\theta + y\sin\theta = \rho$ and is independent of ω:

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega\rho} \, d\omega \right]_{\rho = x\cos\theta + y\sin\theta} d\theta$$

Reconstruction using Parallel Beam filtered back projection (Cont…)

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega\rho} d\omega \right]_{\rho = x\cos\theta + y\sin\theta} d\theta$$

For a given angle θ, the inner expression is the 1-D Fourier transform of the projection multiplied by a ramp filter $|\omega|$.

This is equivalent in filtering the projection with a high-pass filter with Fourier Transform $|\omega|$ before back-projection.

Reconstruction using Parallel Beam filtered back projection (Cont…)

$$f(x,y)=\int_0^{\pi}\left[\int_{-\infty}^{+\infty}|\omega|G(\omega,\theta)e^{j2\pi\omega\rho}d\omega\right]_{\rho=x\cos\theta+y\sin\theta}d\theta$$

Problem: the filter $H(\omega)=|\omega|$ is not integrable in the inverse Fourier transform as it extends to infinity in both directions.

It should be truncated in the frequency domain.

The simplest approach is to multiply it by a box filter in the frequency domain.

Ringing will be noticeable.

Windows with smoother transitions are used.

Reconstruction using Parallel Beam filtered back projection (Cont…)

An M-point discrete window function used frequently is

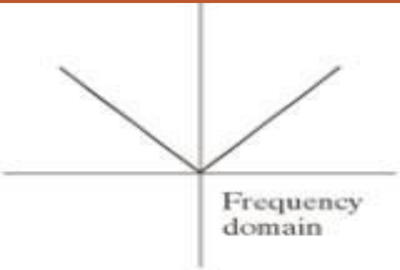
$$h(\omega) = \begin{cases} c + (c-1)\cos\dfrac{2\pi\omega}{M-1} & 0 \le \omega \le M-1 \\ 0 & \text{otherwise} \end{cases}$$

When c=0.54, it is called the *Hamming* window.
When c=0.5, it is called the *Hann* window.

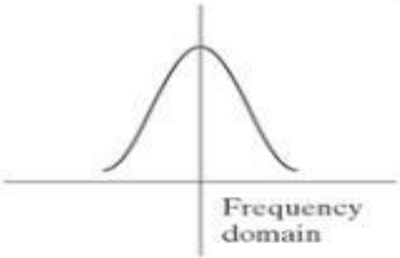
By these means ringing decreases.

Reconstruction using Parallel Beam filtered back projection (Cont…)

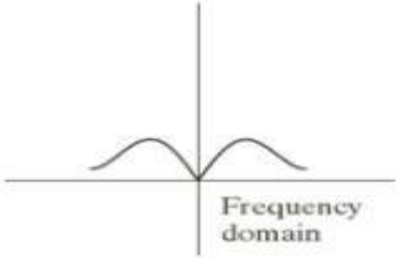
Ramp filter multiplied by a box window

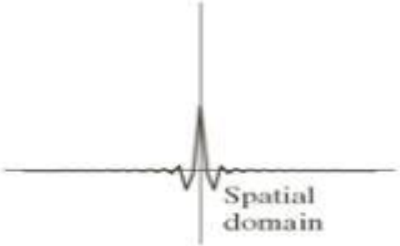


Hamming window



Ramp filter multiplied by a Hamming window





Reconstruction using Parallel Beam filtered back projection (Cont…)

The *complete* back-projection is obtained as follows:

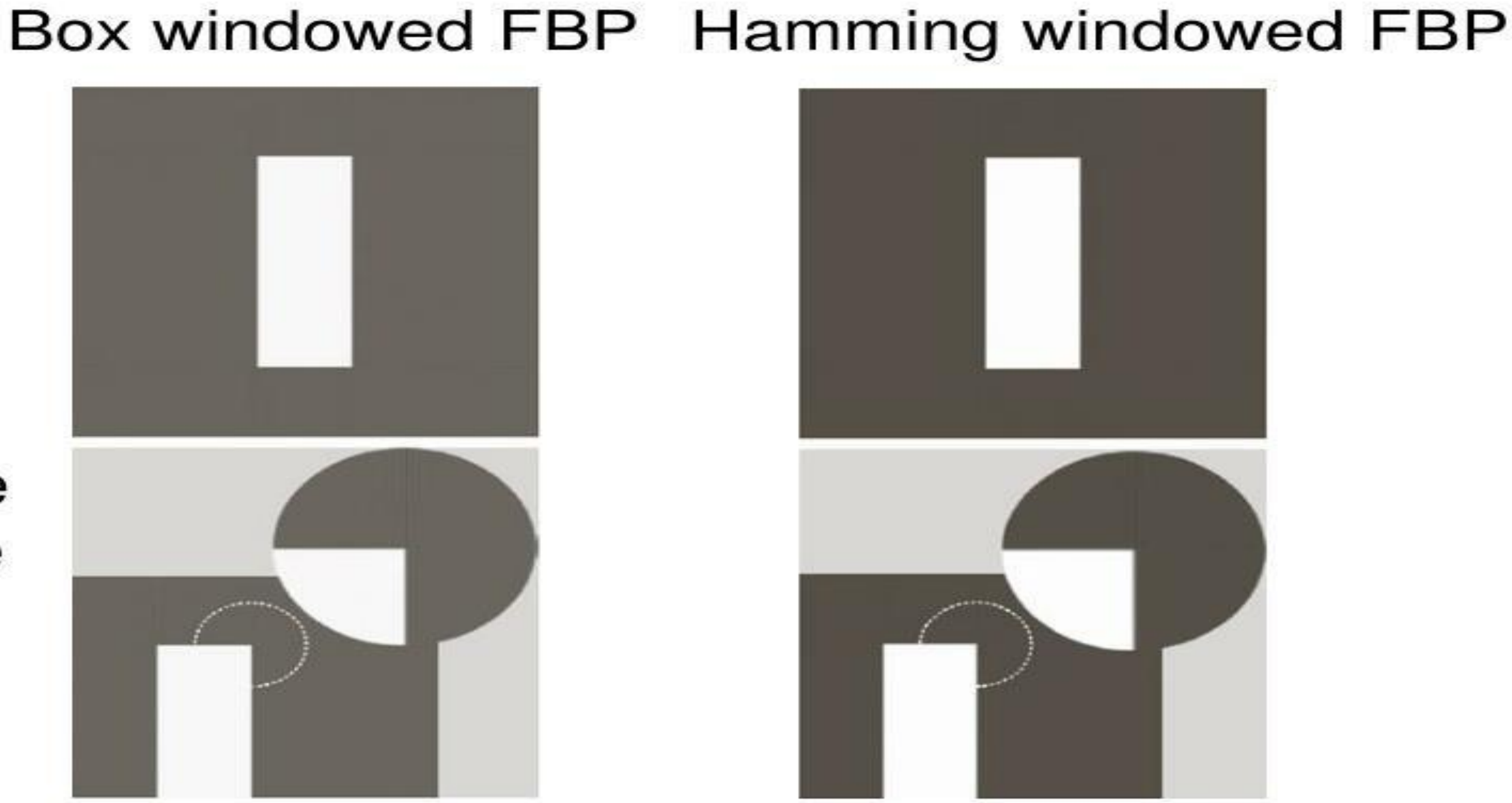
1. Compute the 1-D Fourier transform of each projection.
2. Multiply each Fourier transform by the filter function $|\omega|$ (multiplied by a suitable window, e.g. Hamming).
3. Obtain the inverse 1-D Fourier transform of each resulting filtered transform.
4. Back-project and integrate all the 1-D inverse transforms from step 3.

Reconstruction using Parallel Beam filtered back projection (Cont…)

Because of the filter function the reconstruction approach is called *filtered back-projection* (FBP).

- Sampling issues must be taken into account to prevent aliasing.
  - The number of rays per angle which determines the number of samples for each projection.
  - The number of rotation angles which determines the number of reconstructed images.

Reconstruction using Parallel Beam filtered back projection (Cont…)



Ringing is more pronounced in the Ramp FBP image

Reconstruction using Parallel Beam filtered back projection (Cont...)

Box windowed FBP

Reconstruction using Parallel Beam filtered back projection (Cont…)

Hamming windowed FBP

Reconstruction using Parallel Beam filtered back projection (Cont...)

Box windowed FBP

Hamming windowed FBP

There are no sharp transitions in the Shepp-Logan phantom and the two filters provide similar results.

Reconstruction using Parallel Beam filtered back projection (Cont…)

Back-projection Ramp FBP Hamming FBP

Notice the difference between the simple back-projection and the filtered back-projection.

Suggested Readings

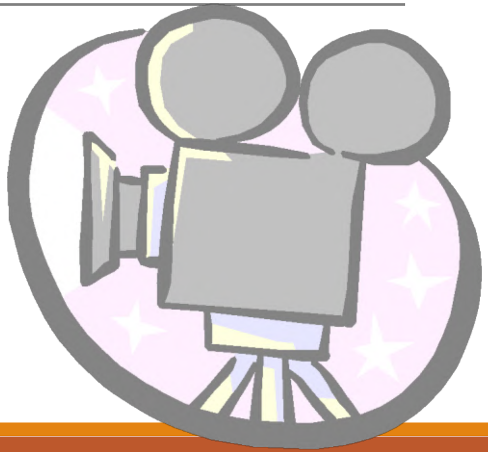
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Image Compression Fundamentals
    - Coding Redundancy
    - Interpixel Redundancy
    - Psychovisual Redundancy
    - Fidelity Criteria

Image Compression

- The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information Data Information.

- Various amount of data can be used to represent the same information

- Data might contain elements that provide no relevant information : *data redundancy*

- Data redundancy is a central issue in image compression. It is not an abstract concept but mathematically quantifiable entity

For human eyes, the image will still seems to be the same even when the Compression ratio is equal 10

Human eyes are **less sensitive** to those **high frequency** signals

Our eyes will **average fine details** within the small area and record only the overall intensity of the area, which is regarded as a **lowpass** filter.

Why do We Need Compression?

- For data STORAGE and data TRANSMISSION

Images take a lot of storage space:

1024 x 1024 x 32 bits images requires 4 MB

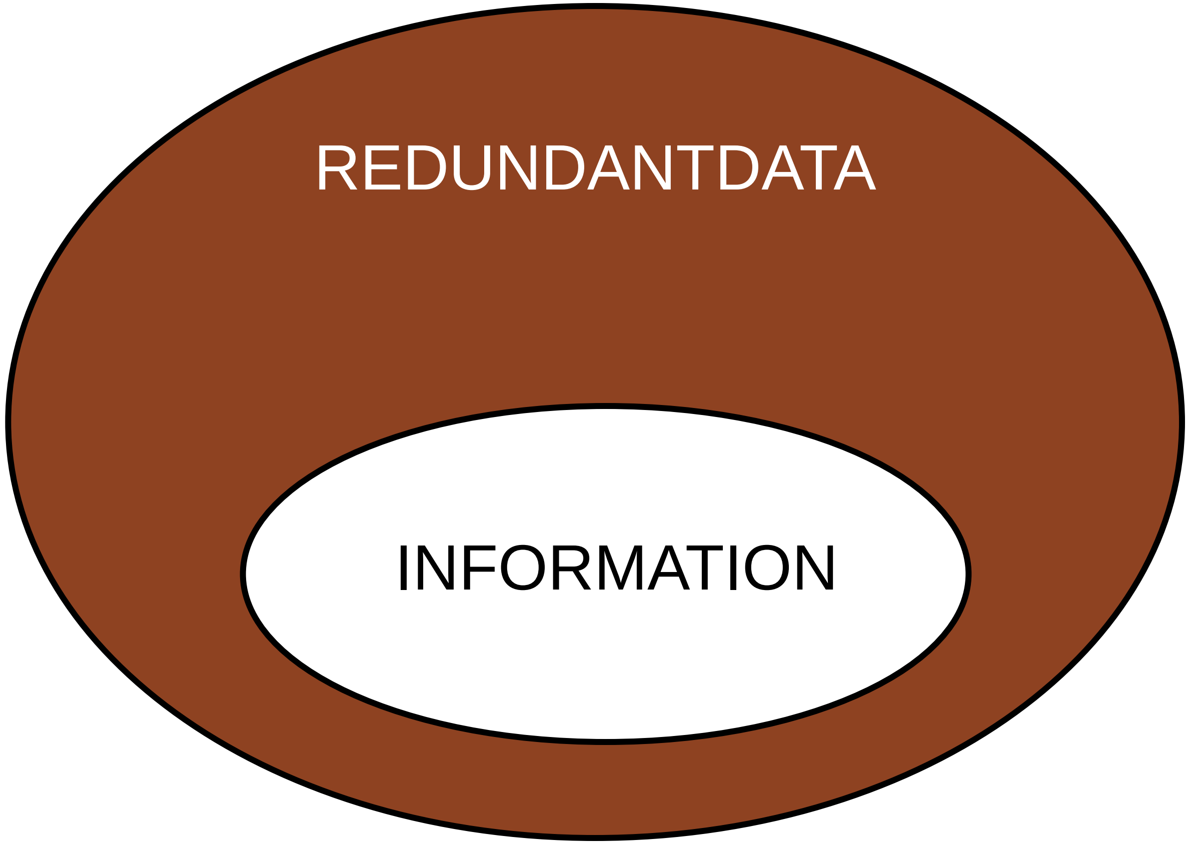
suppose you have some video that is 720 x 480 x 24 bits x 30 frames per second , 1 minute of video would require 1.54 GB

A two hour movie consists of $2.24 \times 10^{11} \approx 224$ GB of data :27 DVD of 8.5 GB are needed to store it.

Many bytes take a long time to transfer slow connections – suppose we have 56,000 bps

- 4MB will take almost 10 minutes
- 1.54 GB will take almost 66 hours

Information vs Data



DATA = INFORMATION + REDUNDANT DATA

Why Can We Compress?

Goals of compression

- Remove redundancy
- Reduce irrelevance

irrelevance or perceptual redundancy

- not all visual information is perceived by eye/brain, so throw away those that are not.

Data Redundancy

Let n_1 and n_2 denote the number of information carrying units in two data sets that represent the same information

The relative redundancy R_D is define as :

$$R_D = 1 - \frac{1}{C_R}$$

where C_R, commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2}$$

Data Redundancy

If $n_1 = n_2$, $C_R=1$ and $R_D=0$ $\Rightarrow$ *no redundancy*

If $n_1 >> n_2$, $C_R \rightarrow \infty$ and $R_D \rightarrow 1$ $\Rightarrow$ *high redundancy*

If $n_1 << n_2$, $C_R \rightarrow 0$ and $R_D \rightarrow -\infty$ $\Rightarrow$ *undesirable*

A compression ration of 10 (10:1) means that the first data set has 10 information carrying units (say, bits) for every 1 unit in the second (compressed) data set.

Data Redundancy

$$R_D = 1 - \frac{1}{C_R}$$

If $C_R = \frac{10}{1}$, then $R_D = 1 - \frac{1}{10} = 0.9$

(90% of the data in dataset 1 is redundant)

if $n_2 = n_1$, then C_R=1, R_D=0

if $n_2 \ll n_1$, then $C_R \to \infty$, $R_D \to 1$

Fundamentals

Basic data redundancies:

1. Coding redundancy
2. Inter-pixel redundancy
3. Psycho-visual redundancy

Coding redundancy

our quantized data is represented using codewords.

if the size of the codeword is larger than is necessary to represent all quantization levels, then we have coding redundancy .

An 8-bit has 256 distinct levels of intensity in an image .

But if there are only 16 different grey levels in a image than it will require only 4 bit instead of 8 bits.

Coding redundancy can also arise due to the use of fixed-length codewords.

Example:

4	8	12
15	7	11
1	6	2

(a)

0100	1000	1100
1111	0111	1011
0001	0110	0010

(b)

100	1000	1100
1111	111	1011
1	110	10

(c)

(b) Number of bits required=3×3×4
=36 bits

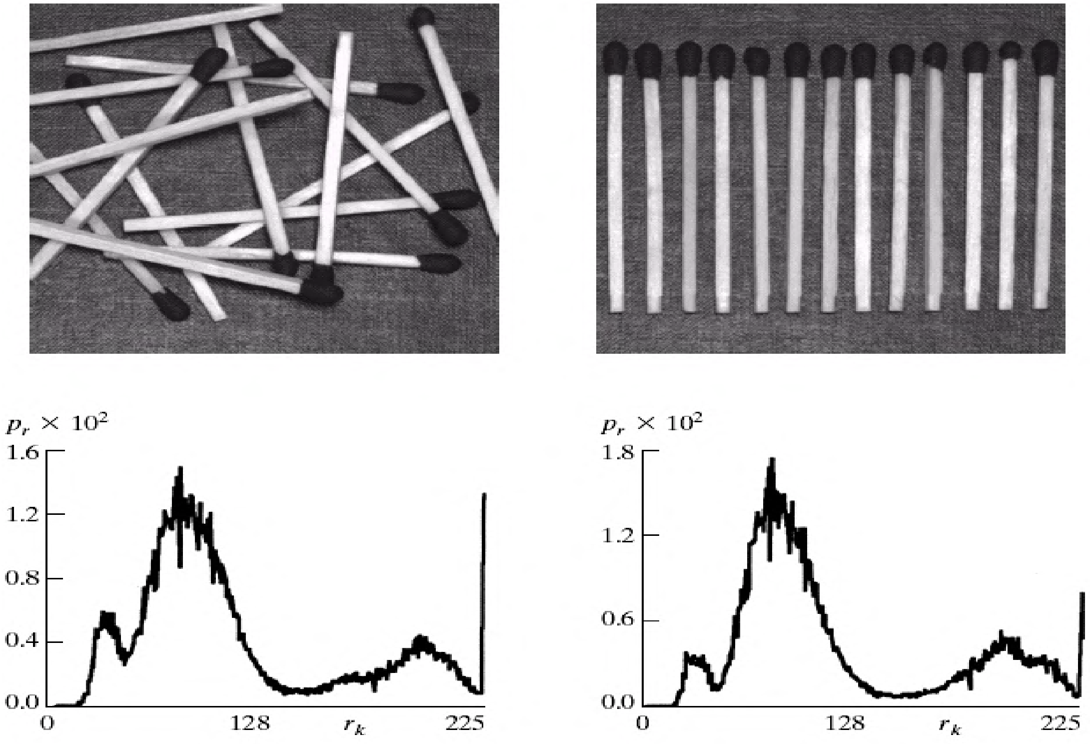
(c) Number of bits required=3×3+4×4+1+2
=9+16+1+2=29 bits

Inter-pixel Redundancy

Here the two pictures have
Approximately the same
Histogram.

We must exploit Pixel
Dependencies.

Each pixel can be estimated
From its neighbors.



Interpixel redundancy

- Interpixel redundancy implies that any pixel value can be reasonably predicted by its neighbors (i.e., correlated).

Psychovisual redundancy

The human eye does not respond with equal sensitivity to all visual information.

It is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.

<u>Idea:</u> discard data that is perceptually insignificant!

Fidelity Criteria

The general classes of criteria :

1. Objective fidelity criteria
2. Subjective fidelity criteria

Fidelity Criteria

Objective fidelity:

Level of information loss can be expressed as a function of the original and the compressed and subsequently decompressed image.

Fidelity Criteria

The error between two image of size M * N is given by:

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

So, the total error between the two images is

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]$$

The root-mean-square error averaged over the whole image is

$$e_{rms} = \frac{1}{MN}\sqrt{[\hat{f}(x,y) - f(x,y)]^2}$$

$$\hat{f}(x,y) - reconstructed$$

$$f(x,y) - original$$

Fidelity Criteria

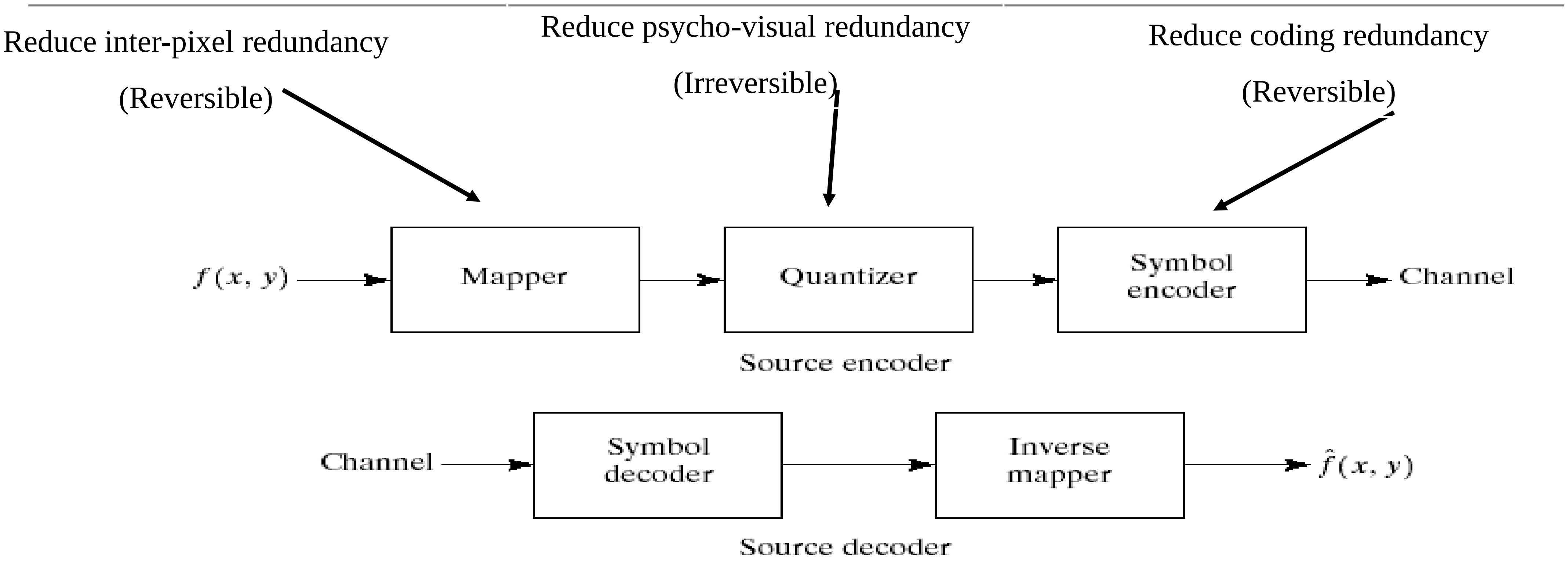
Subjective fidelity (Viewed by Human):

- By absolute rating
- By means of side-by-side comparison of $f(x, y)$ and $\hat{f}(x, y)$

TABLE 8.3
Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Image Compression model



(a) Source encoder and (b) source decoder model.

Classification

- Lossless compression
  - lossless compression for legal and medical documents, computer programs
  - exploit only code and inter-pixel redundancy
- Lossy compression
  - digital image and video where some errors or loss can be tolerated
  - exploit both code and inter-pixel redundancy and psycho-visual perception properties

Lossless compression

APPLICATIONS:

- MEDICAL IMAGING
- SATELLITE IMAGING
-

THEY PROVIDE: COMPRESSION RATIO OF 2 TO 10.

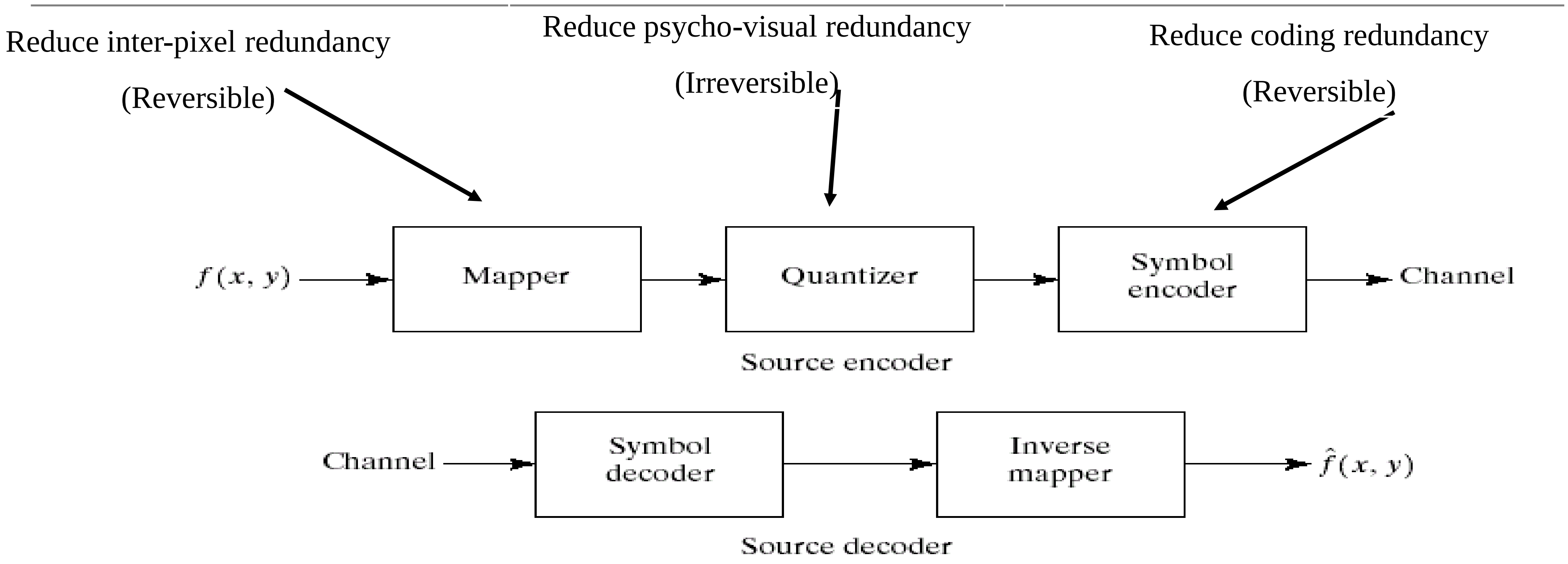
Techniques of Lossless Compression

Huffman Coding

Arithmetic Coding

Run Length Coding

Image Compression model



(a) Source encoder and (b) source decoder model.

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding (optimal code)

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

FIGURE 8.11
Huffman source reductions.

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

FIGURE 8.12 Huffman code assignment procedure.

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$

$$= 2.2 \, bits / symbol$$

$$entropy = 2.14 \, bits / symbol$$

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

FIGURE 8.12
Huffman code assignment procedure.

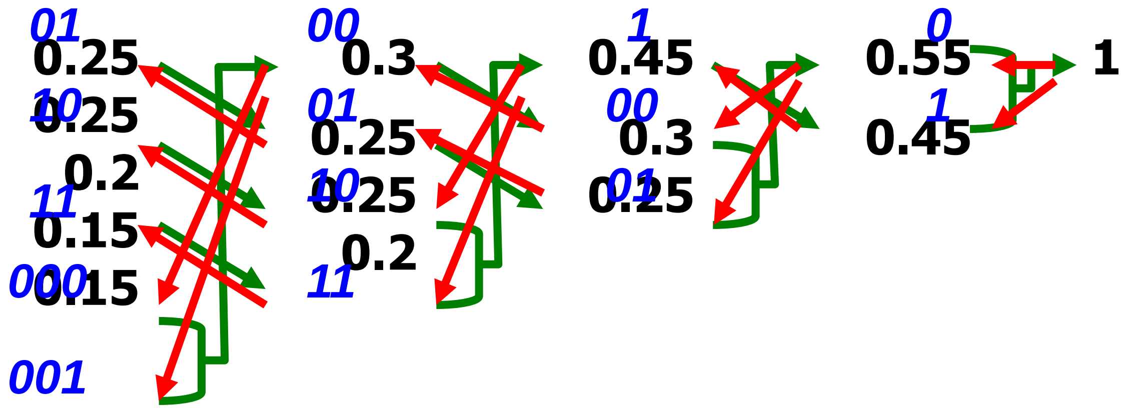
Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Example:

010100111100 = a3 a1 a2 a2 a6

Huffman Coding

Codeword length	Codeword	X	Probability
2	01	1	0.25
2	10	2	0.25
3	11	3	0.2
3	000	4	0.15
3	001	5	0.15



Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

- Variable length code whose length is inversely proportional to that character's frequency
- must satisfy non-prefix property to be uniquely decodable
- two pass algorithm
  - first pass accumulates the character frequency and generate codebook
  - second pass does compression with the codebook

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

- create codes by constructing a binary tree
  - 1. consider all characters as free nodes
  - 2. assign two free nodes with lowest frequency to a parent nodes with weights equal to sum of their frequencies
  - 3. remove the two free nodes and add the newly created parent node to the list of free nodes
  - 4. repeat step2 and 3 until there is one free node left. It becomes the root of tree

Suggested Readings

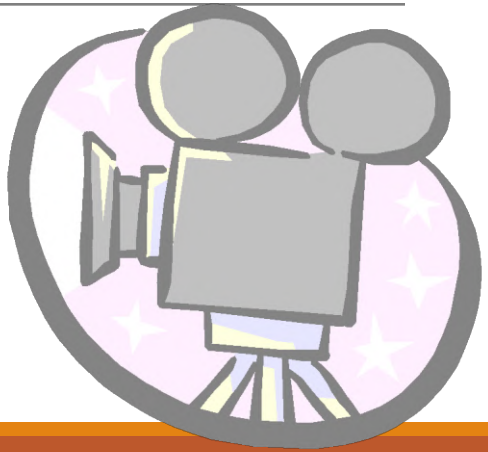
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341



Outline

- Lossless Compression
    - Huffman Coding
    - Arithmetic Coding

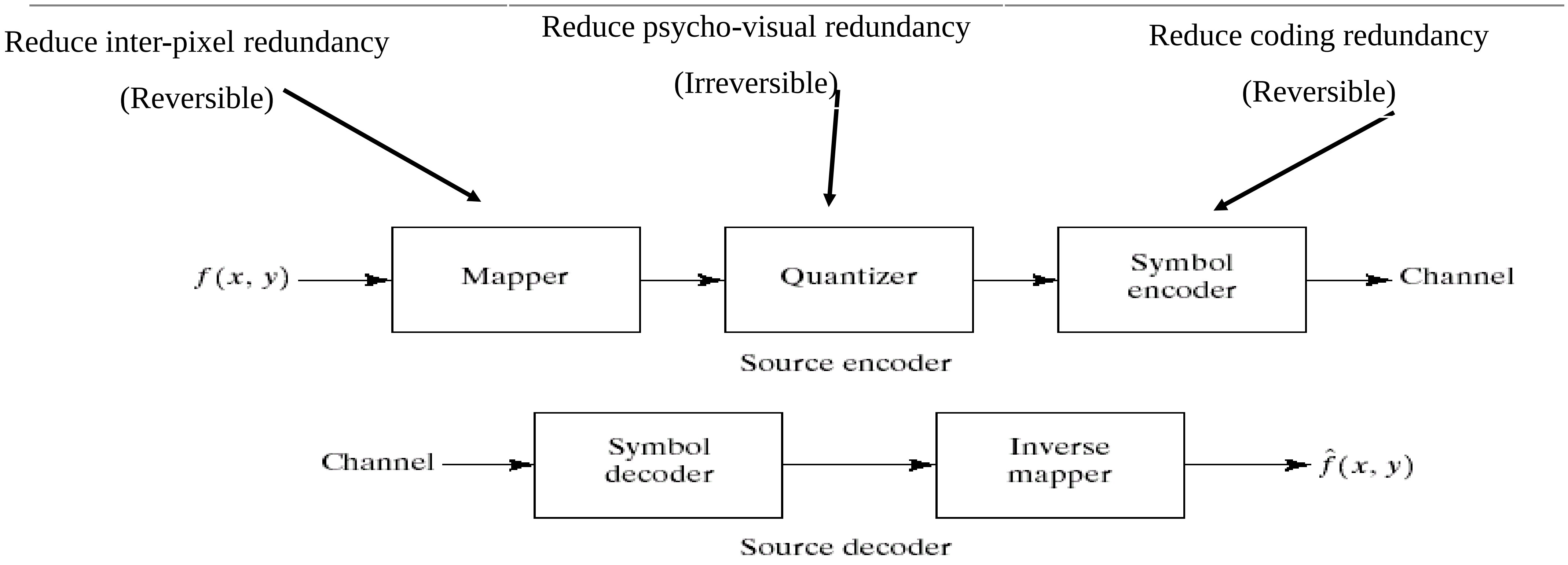
Techniques of Lossless Compression

Huffman Coding

Arithmetic Coding

Run Length Coding

Image Compression model



(a) Source encoder and (b) source decoder model.

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding (optimal code)

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

FIGURE 8.11 Huffman source reductions.

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

FIGURE 8.12
Huffman code assignment procedure.

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$

$$= 2.2\, bits\ /\ symbol$$

$$entropy = 2.14\, bits\ /\ symbol$$

Example:

a3 a1 a2 a2 a6 =010100111100

Lossless or Error-Free Compression

Variable-length Coding

Huffman decoding

FIGURE 8.12
Huffman code assignment procedure.

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Example:

010100111100 = a3 a1 a2 a2 a6

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

- Variable length code whose length is inversely proportional to that character's frequency
- must satisfy non-prefix property to be uniquely decodable
- two pass algorithm
  - first pass accumulates the character frequency and generate codebook
  - second pass does compression with the codebook

Lossless or Error-Free Compression

Variable-length Coding

Huffman coding

- create codes by constructing a binary tree
  - 1. consider all characters as free nodes
  - 2. assign two free nodes with lowest frequency to a parent nodes with weights equal to sum of their frequencies
  - 3. remove the two free nodes and add the newly created parent node to the list of free nodes
  - 4. repeat step2 and 3 until there is one free node left. It becomes the root of tree

Arithmetic (or Range) Coding

- The main weakness of Huffman coding is that it encodes source symbols one at a time.
- Arithmetic coding encodes sequences of source symbols together.
  - There is no one-to-one correspondence between source symbols and code words.
- Slower than Huffman coding but can achieve better compression.

Arithmetic Coding (cont'd)

A sequence of source symbols is assigned to a sub-interval in [0,1) which can be represented by an arithmetic code, e.g.:

	sub-interval		arithmetic code
$a_1\ a_2\ a_3\ a_3\ a_4$	→ [0.06752, 0.0688)		→ 0.06816

Start with the interval [0, 1) ; a sub-interval is chosen to represent the message which becomes smaller and smaller as the number of symbols in the message increases.

Arithmetic Coding (cont’d)

Encode message: $a_1\ a_2\ a_3\ a_3\ a_4$

Source Symbol	Probability
a_1	0.2
a_2	0.2
a_3	0.4
a_4	0.2

1) Start with interval [0, 1)

0 1

2) Subdivide [0, 1) based on the probabilities of α_i

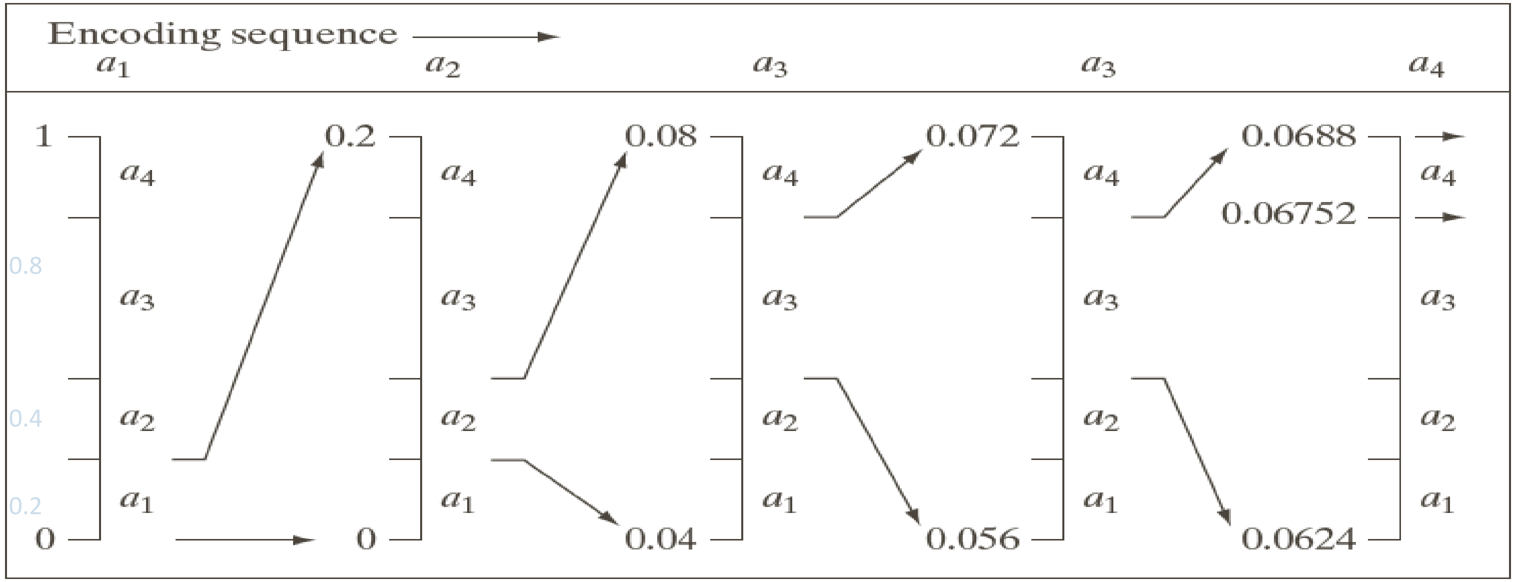
Initial Subinterval
[0.0, 0.2)
[0.2, 0.4)
[0.4, 0.8)
[0.8, 1.0)

3) Update interval by processing source symbols

Arithmetic Coding

Example: Encode the message $a_1\ a_2\ a_3\ a_3\ a_4$

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)



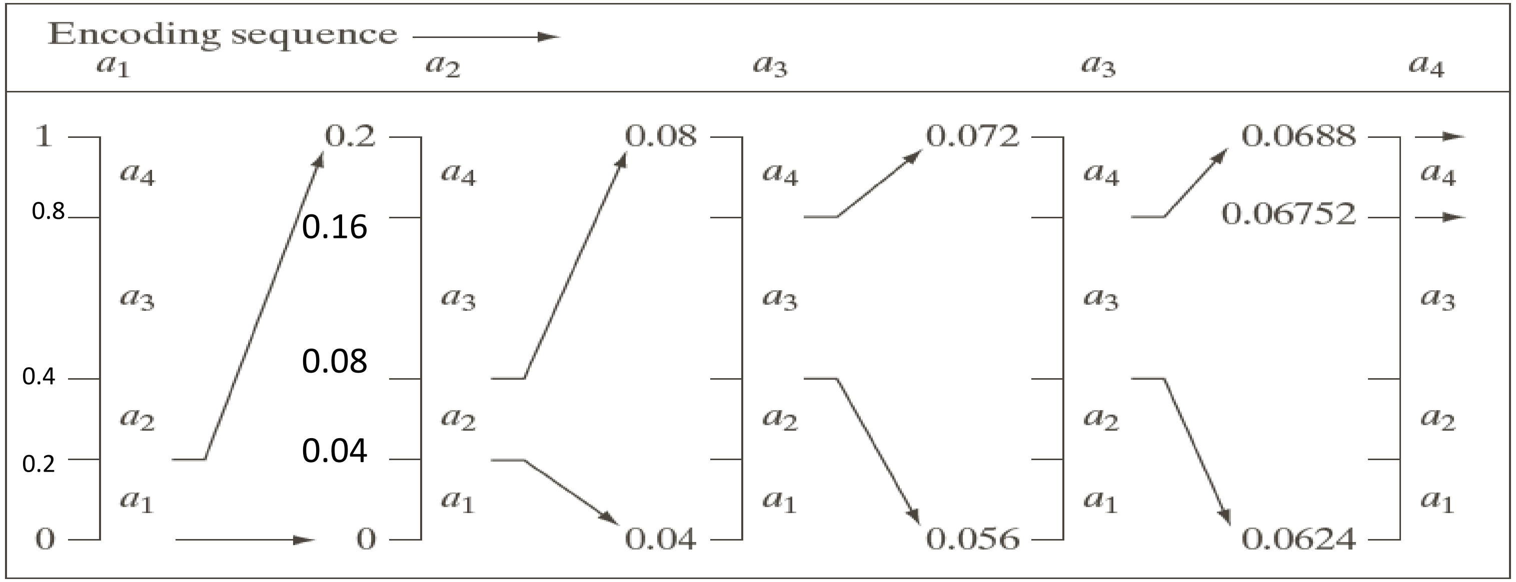
Example: Encode the message $a_1\ a_2\ a_3\ a_3\ a_4$

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

d=upper bound-lower bound

Range of symbol=lower limit+ d *(probability of Symbol)

d=0.2-0=0.20
Range of 'a1'=0+0.2*0.2=0.04



Range of 'a2'=0.04+0.2*0.2=0.08

Range of 'a3'=0.08+0.2*0.4=0.16

Range of 'a4'=0.16+0.2*0.2=0.2

Example: Encode the message $a_1\ a_2\ a_3\ a_3\ a_4$

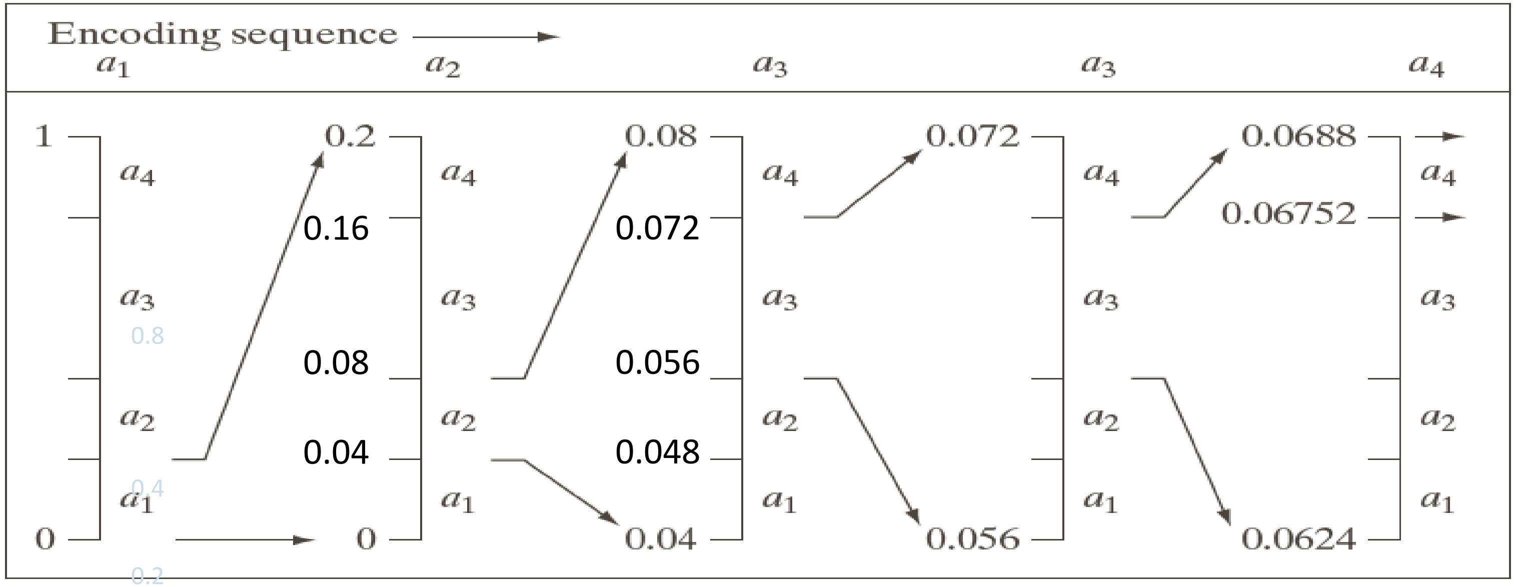
d=upper bound-lower bound

Range of symbol=lower limit+ d *(probability of Symbol)

d=0.08-0.04=0.04

Range of 'a1'=0.04+0.04*0.2=0.048

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)



Range of 'a2'=0.048+0.04*0.2=0.056

Range of 'a3'=0.056+0.04*0.4=0.072

Range of 'a4'=0.072+0.04*0.2=0.08

Example: Encode the message $a_1\ a_2\ a_3\ a_3\ a_4$

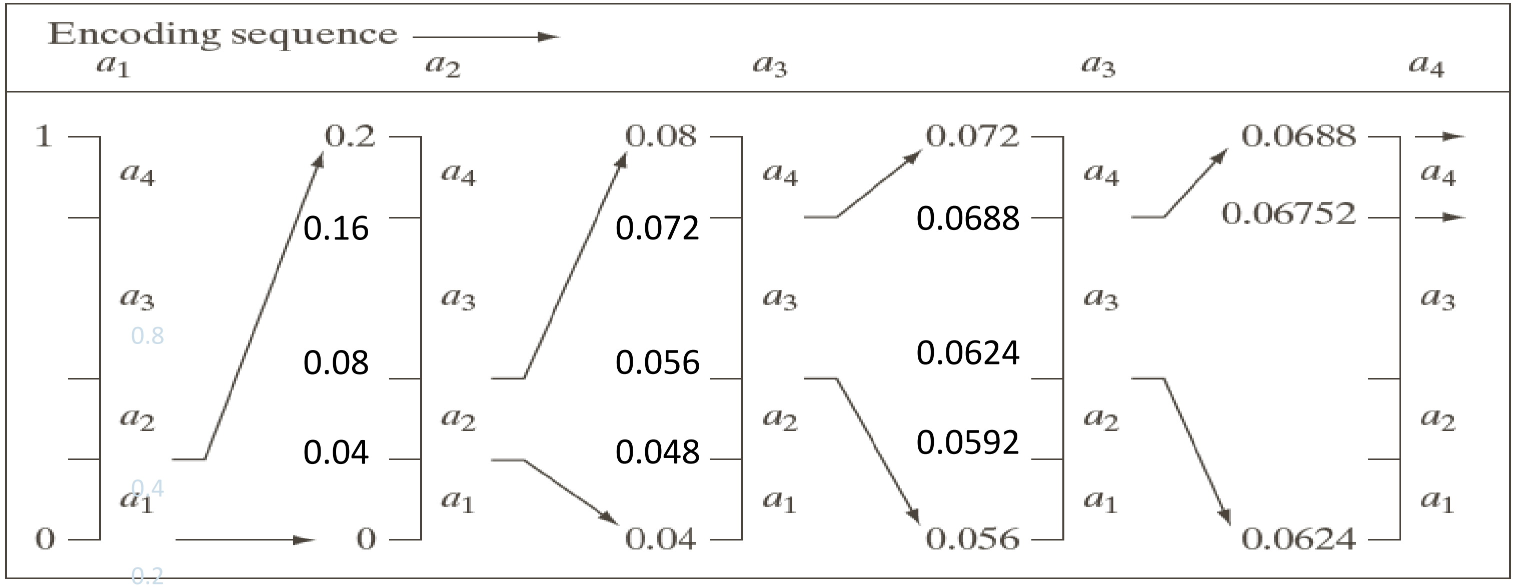
d=upper bound-lower bound

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Range of symbol=lower limit+ d *(probability of Symbol)

d=0.072-0.056=0.016

Range of ‘a1’=0.056+0.016*0.2=0.0592



Range of ‘a2’=0.0592+0.016*0.2=0.0624

Range of ‘a3’=0.0624+0.016*0.4=0.0688

Range of ‘a4’=0.0688+0.016*0.2=0.072

Example: Encode the message $a_1\ a_2\ a_3\ a_3\ a_4$

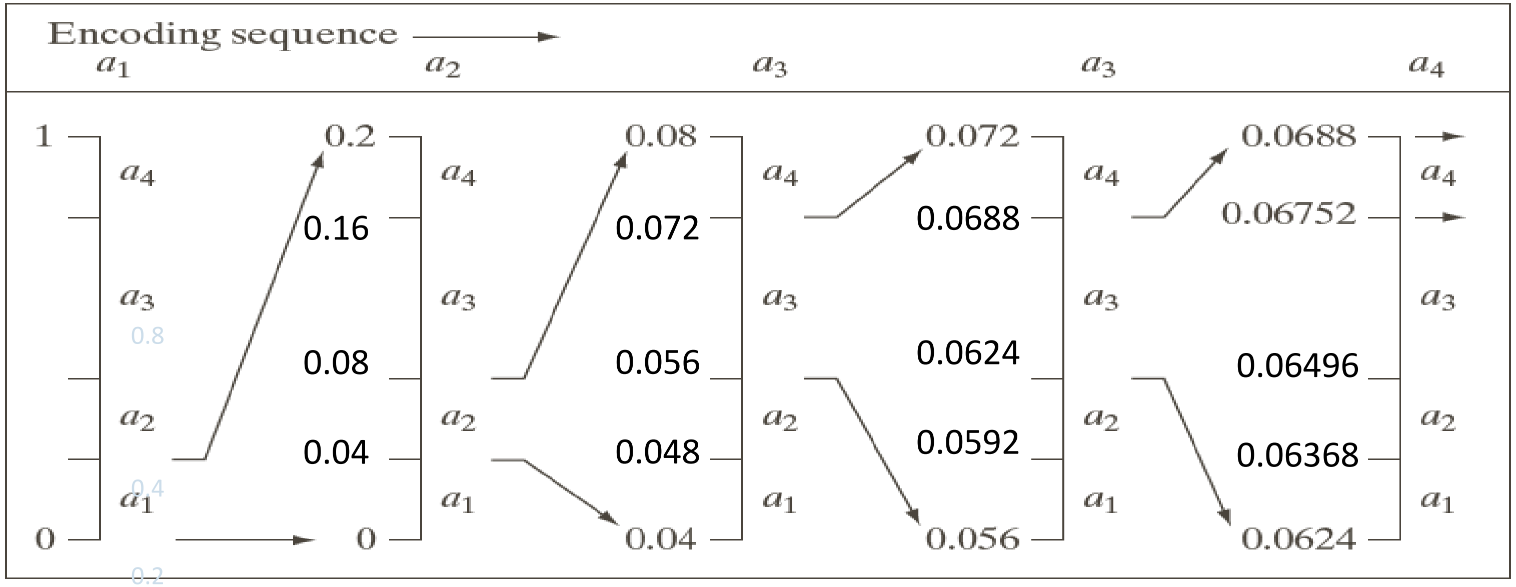
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

d=upper bound-lower bound

Range of symbol=lower limit+ d *(probability of Symbol)

d=0.0688-0.0624=0.0064

Range of 'a1'=0.0624+0.0064*0.2=0.06368



Range of 'a2'=0.06368+0.0064*0.2=0.06496

Range of 'a3'=0.06496+0.0064*0.4=0.06752

Range of 'a4'=0.06752+0.0064*0.2=0.0688

Example (cont..)

Encode $a_1\ a_2\ a_3\ a_3\ a_4$

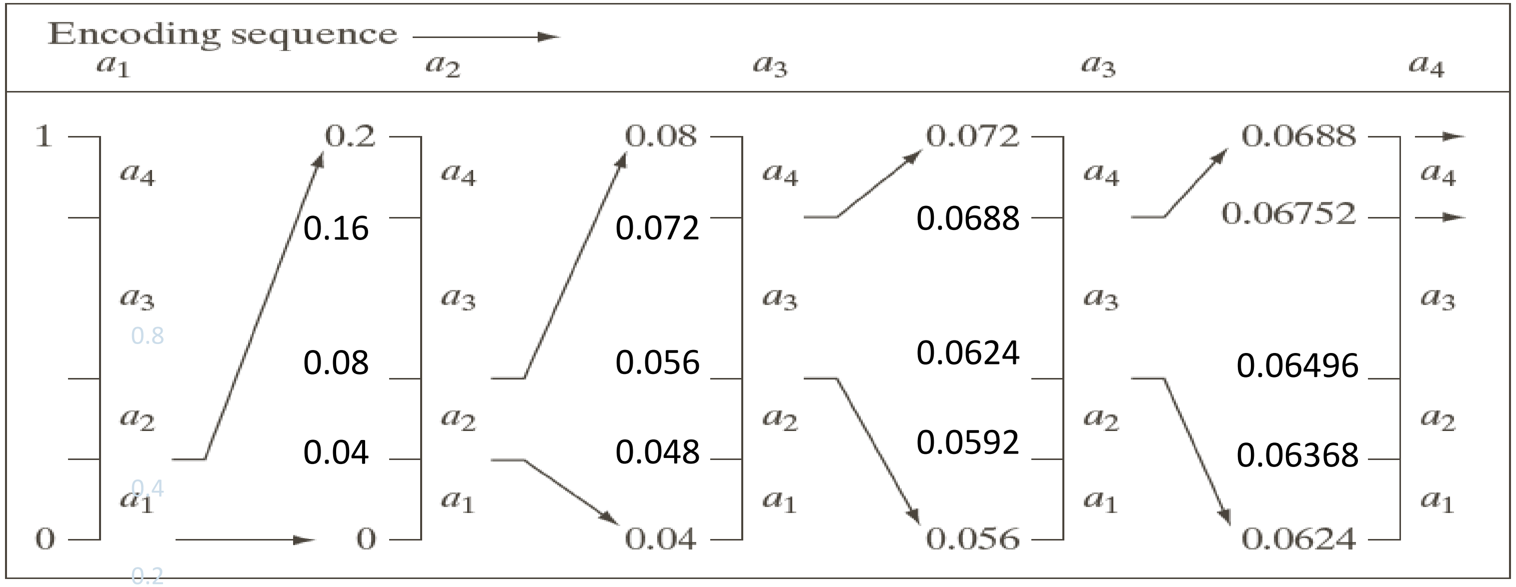
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Code lies in this range [0.06752, 0.0688)

or

Tag=(upper limit+lower linit)/2 = 0.06816

(must be inside sub-interval)

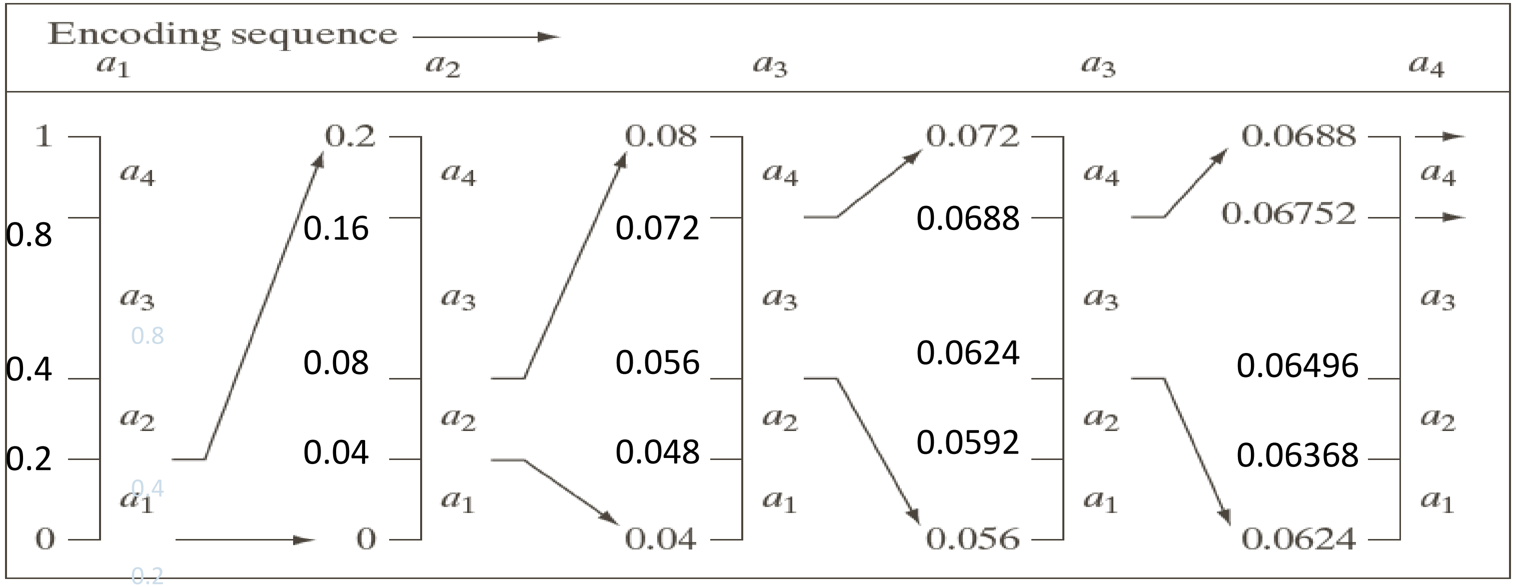


Arithmetic Decoding

Example

Decode the message 0.06816

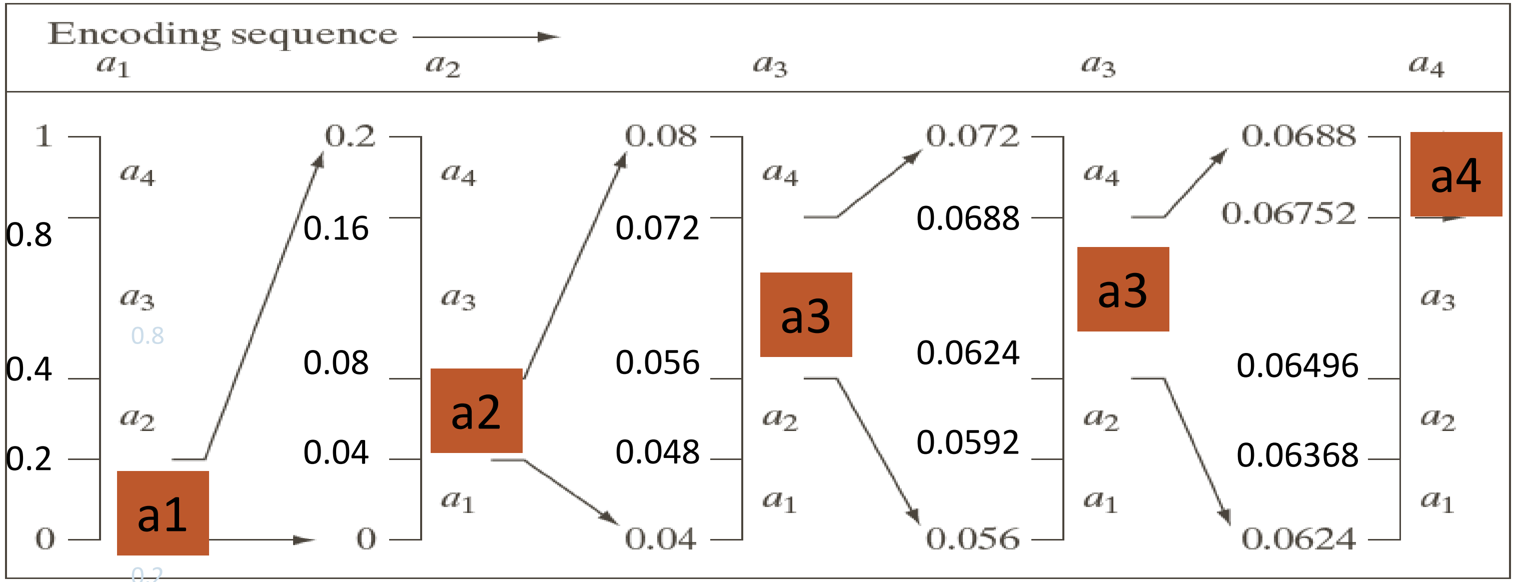
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)



Example

Decode the message 0.06816

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)



Decoded message

$a_1a_2a_3a_3a_4$

Suggested Readings

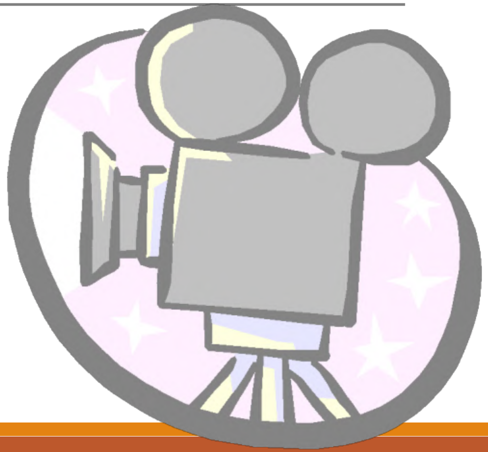
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

CS-317/CS-341

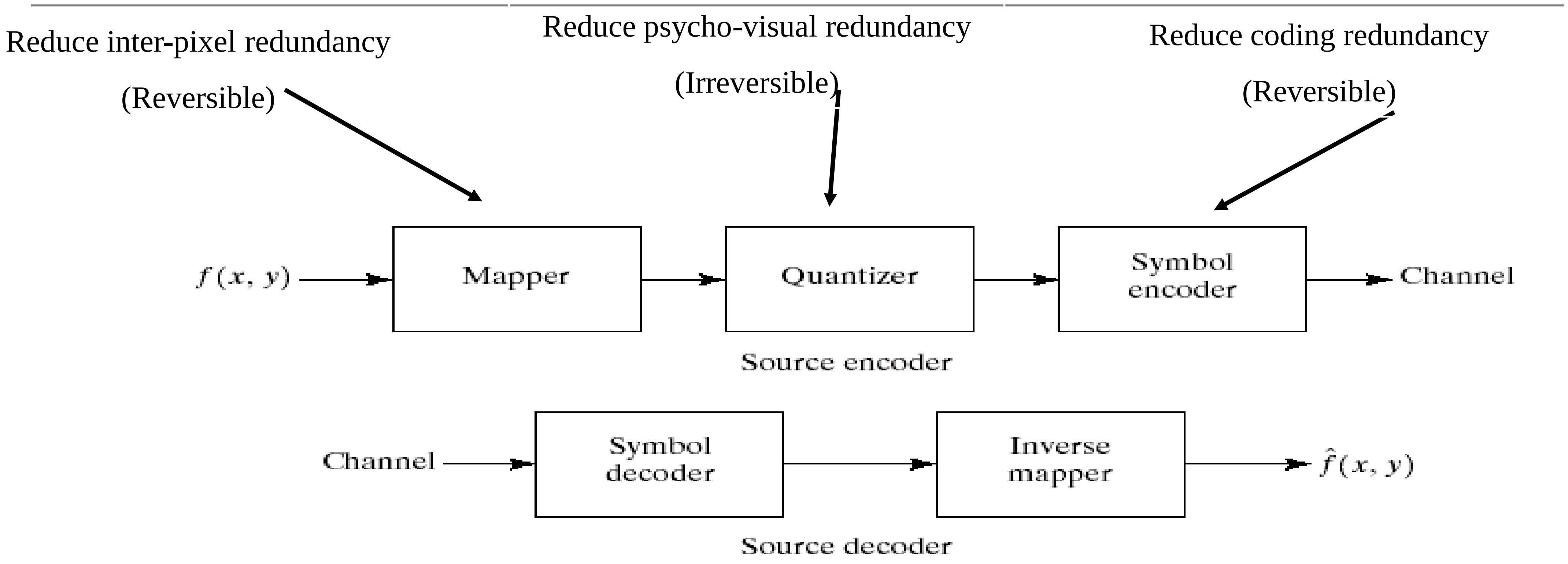


Outline

- Lossless Compression
  - Run length coding

- Lossy image compression techniques

Image Compression model



(a) Source encoder and (b) source decoder model.

Run-length encoding

- Run-length encoding is probably the simplest method of compression.

- It can be used to compress data made of any combination of symbols.

- It does not need to know the frequency of occurrence of symbols and can be very efficient if data is represented as 0s and 1s.

- The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.

    replace runs of symbols (possibly of length one) with pairs of (run-length, symbol)

    For images, the maximum run-length is the size of a row

- The method can be even more efficient if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other.

Run-length coding (RLC) (addresses interpixel redundancy)

Reduce the size of a repeating string of symbols (i.e., runs):

1 1 1 1 0 0 0 0 0 0 1 → (1,5) (0, 6) (1, 1)

a a a b b b b b b c c → (a,3) (b, 6) (c, 2)

No. of vector=3, maximum length=6, so 3 bits in binary are required, no. of bits per pixel=1

Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

3*(3+1)=12

Original size= 12*1=12

Run-length coding (RLC) (addresses interpixel redundancy)

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

Reduce the size of a repeating string of symbols (i.e., runs):

Horizontal RLC

No. of vectors : (0,5) ;(0, 3), (1, 2);(1,5);(1,5);(1,5)

No. of vector=6, maximum length=5, so 3 bits in binary are required, no. of bits per pixel=1

Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

6*(3+1)=24

Original size= 5*5*1=25, CR=25/24=1.042:1

Run-length coding (RLC) (addresses interpixel redundancy)

0	**0**	**0**	**0**	**0**
0	0	0	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Reduce the size of a repeating string of symbols (i.e., runs):

Vertical RLC

No. of vectors : (0,2) (1,3); (0, 2),(1,3);(0,2),(1,3));(0,1),(1,4);(0,1), (1,4)

No. of vector=10, maximum length=4, so 3 bits in binary are required, no. of bits per pixel=1

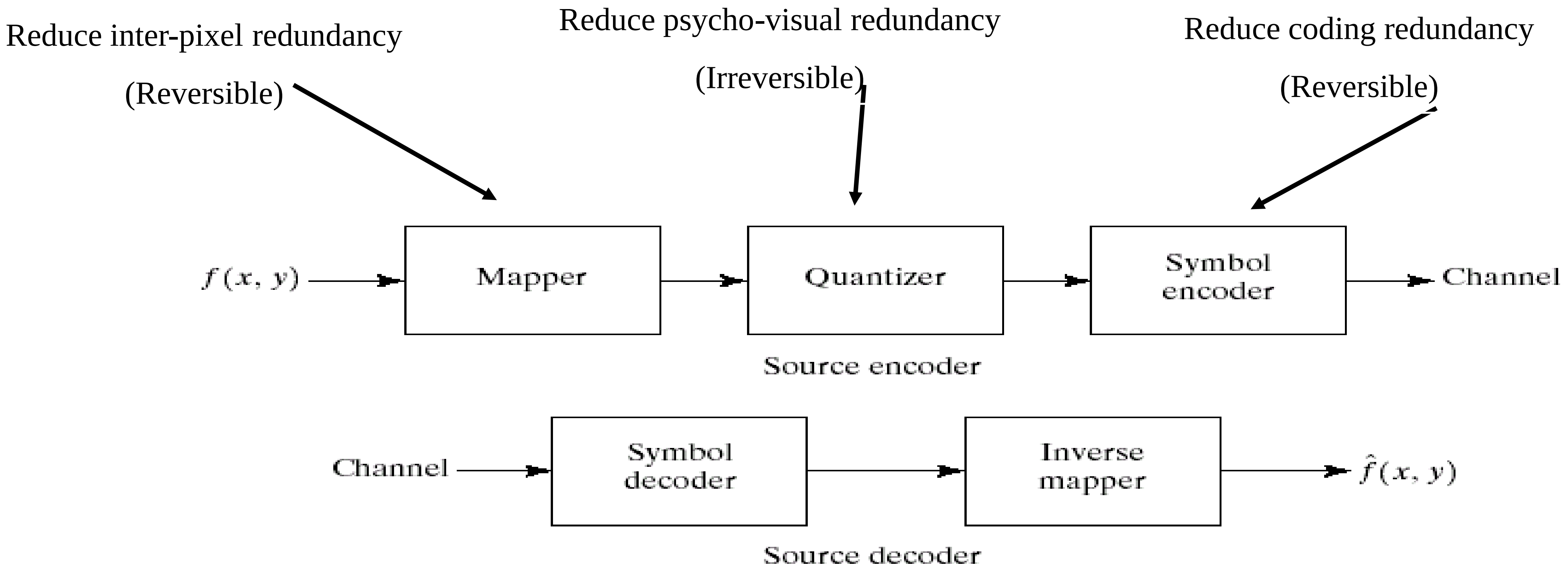
Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

10*(3+1)=40

Original size= 5*5*1=25, CR=25/40=0.625:1

Lossy Image Compression

Image Compression model



(a) Source encoder and (b) source decoder model.

Why Lossy?

In most applications related to consumer electronics, lossless compression is not necessary

- What we care is the subjective quality of the decoded image, not those intensity values

With the relaxation, it is possible to achieve a higher compression ratio (CR)

- For photographic images, CR is usually below 2 for lossless, but can reach over 10 for lossy

A Simple Experiment

Bit-plane representation

$$A=a_0+a_1 2+a_2 2^2+\ \ldots\ \ldots\ +a_7 2^7$$

Least Significant Bit
(LSB)

Most Significant Bit
(MSB)

Example

$$A=129 \rightarrow a_0 a_1 a_2 \ldots a_7=10000001$$

$$a_0 a_1 a_2 \ldots a_7=00110011 \rightarrow A=4+8+64+128=204$$

A Simple Experiment (Con't)

- How will the reduction of gray-level resolution affect the image quality?
  - Test 1: make all pixels even numbers (i.e., knock down a_0 to be zero)
  - Test 2: make all pixels multiples of 4 (i.e., knock down a_0,a_1 to be zeros)
  - Test 3: make all pixels multiples of 4 (i.e., knock down a_0,a_1,a_2 to be zeros)

Experiment Results

original

Test 1

Test 2

Test 3

How to Measure Image Quality?

Subjective

- Evaluated by human observers
- Do not require the original copy as a reference
- Reliable, accurate yet impractical

Objective

- Easy to operate (automatic)
- Often requires the original copy as the reference (measures fidelity rather than quality)

Objective Quality Measures

Mean Square Error (MSE)

original   decoded

$$MSE = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}[X(i,j) - Y(i,j)]^2$$

Peak Signal-to-Noise-Ratio (PSNR)

$$PSNR = 10\log_{10}\frac{255^2}{MSE}(dB)$$

<u>Question</u>:
Can you think of a counter-example to prove objective measure is not consistent with subjective evaluation?

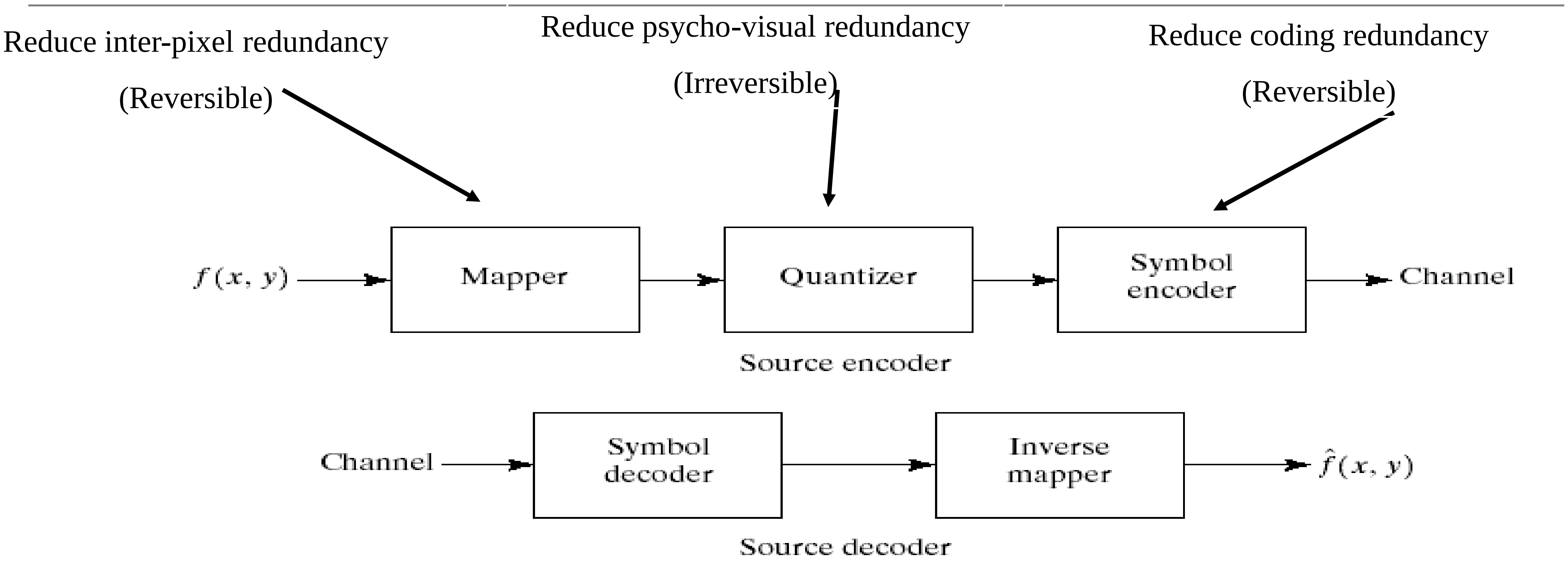
Results

Shifted (MSE=337.8)

Original *cameraman* image

By shifting the last row of the image to become the first row, we affect little on subjective quality but the measured MSE is large

Image Compression model



(a) Source encoder and (b) source decoder model.

Lossy Image Compression

Quantization basics

- Uniform Quantization

What is Quantization?

In image compression

- To limit the possible values of a pixel value or a transform coefficient to a discrete set of values by information theoretic rules

Examples

Unlike entropy, we encounter it everyday (so it is not a monster)

Continuous to discrete

- a quarter of milk, two gallons of gas, normal temperature is 98.6F, my height is 5 foot 9 inches

Discrete to discrete

- Round your tax return to integers
- The mileage of my car is about 55K.

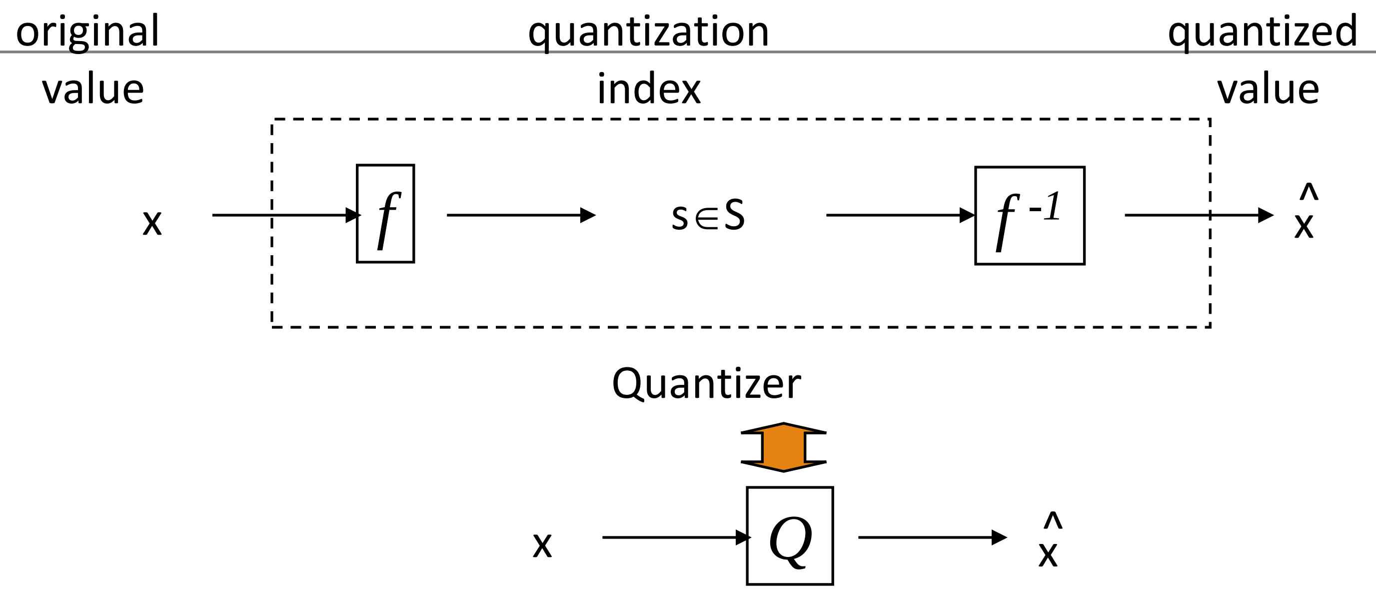
Scalar vs. Vector Quantization

We only consider the scalar quantization (SQ) in this course

- Even for a sequence of values, we will process (quantize) each sample independently

Vector quantization (VQ) is the extension of SQ into high-dimensional space

Definition of (Scalar) Quantization



***f* finds the closest (in terms of Euclidean distance) approximation of *x* from a codebook *C* (a collection of codewords) and assign its index to *s*; f^{-1} operates like a table look-up to return the corresponding codeword**

Numerical Example-3

$$Q(x) = 8 + \left\lfloor \frac{x}{16} \right\rfloor \cdot 16, x \in [0,255]$$

$$\underset{x}{\begin{matrix} 225 & 222 & 235 & 228 \\ 220 & 206 & 209 & 44 \\ 49 & 56 & 64 & 42 \\ 128 & 106 & 94 & 27 \end{matrix}} \longrightarrow \boxed{Q} \longrightarrow \underset{\hat{x}}{\begin{matrix} 232 & 216 & 232 & 232 \\ 216 & 200 & 216 & 40 \\ 56 & 56 & 72 & 40 \\ 136 & 104 & 88 & 24 \end{matrix}}$$

Notes

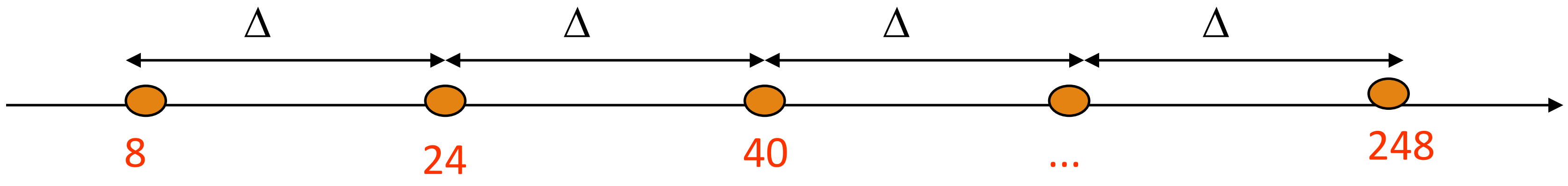
- For scalar quantization, each sample is quantized independently
- Quantization is irreversible

Uniform Quantization (UQ) for Uniform Distribution

Uniform Quantization

A scalar quantization is called uniform quantization (UQ) if all its codewords are uniformly distributed (equally-distanced)

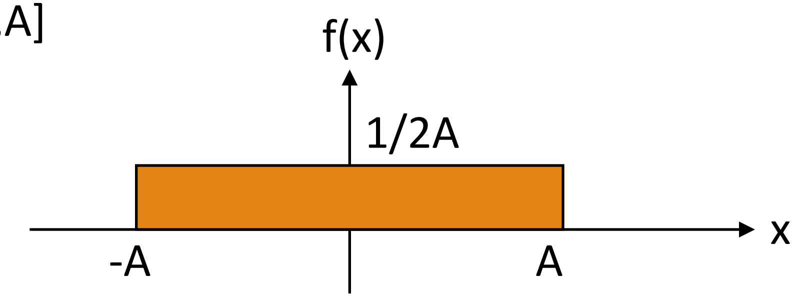
Example (quantization stepsize Δ=16)



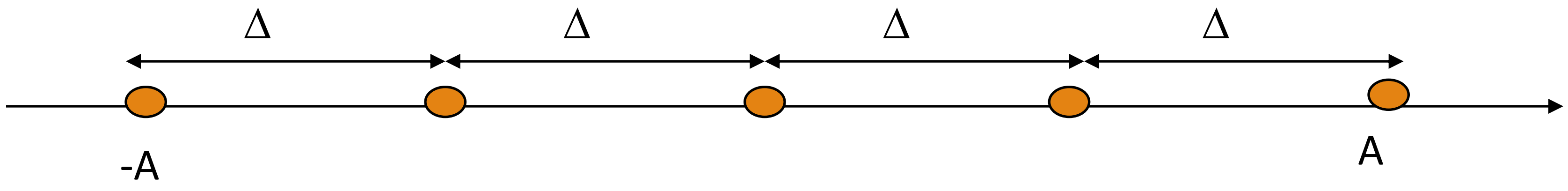
Uniform Distribution

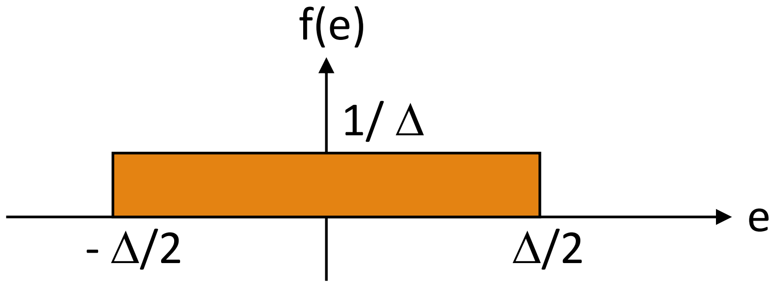
denoted by U[-A,A]

$$f(x) = \begin{cases} 1/2A & x \in [-A, A] \\ 0 & else \end{cases}$$



Quantization Noise of UQ





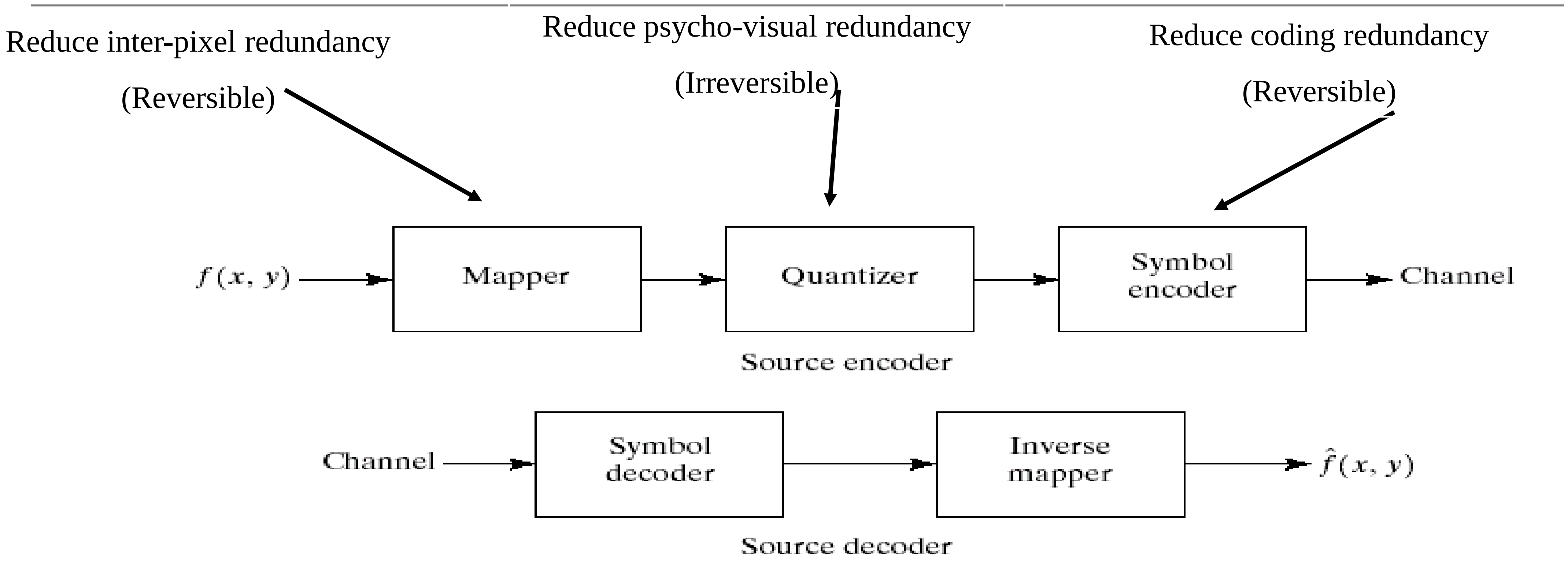
Quantization noise of UQ on uniform distribution is also uniformly distributed

Recall Variance of U[- Δ/2, Δ/2] is

$$\sigma^2 = \frac{1}{12}\Delta^2$$

Basic Transformation

Image Compression model : Transformation

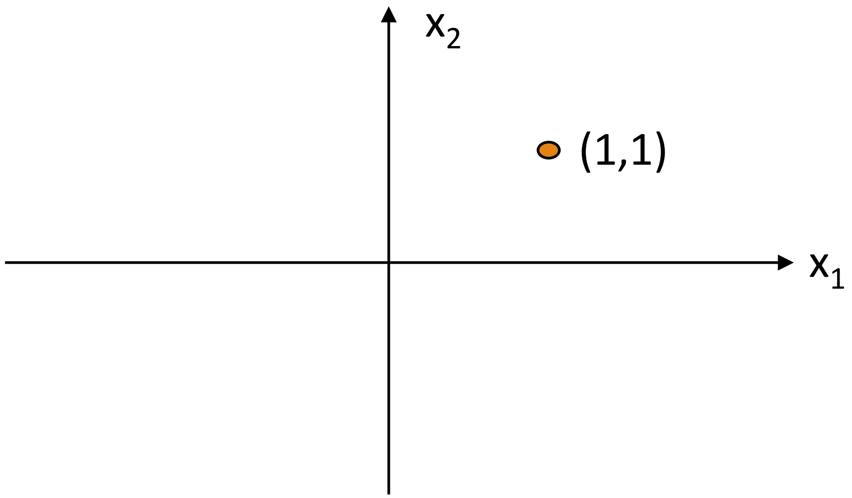


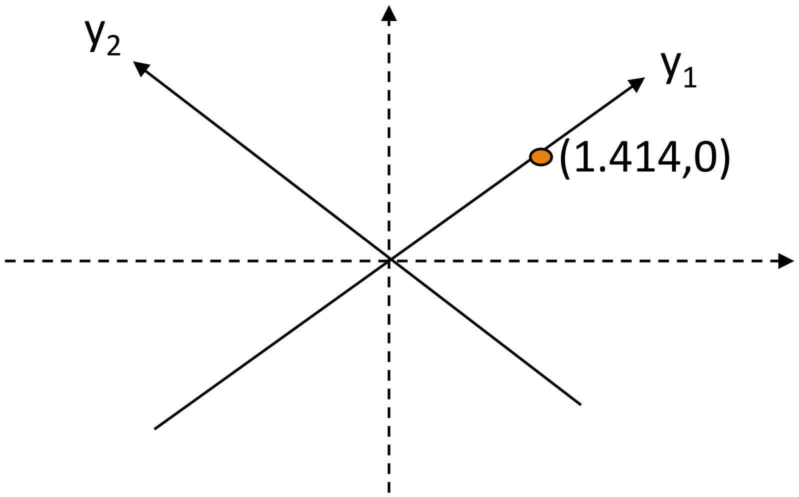
(a) Source encoder and (b) source decoder model.

Lossy Image Compression

- Lossy transform coding
  - Image Transforms (Discrete Cosine Transform)
  - Joint Photographic Expert Group (JPEG)

An Example of 1D Transform with Two Variables





$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \vec{y} = \mathbf{A}\vec{x}, \mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Transform matrix

Generalization into N Variables

forward transform $\vec{y}_{N\times 1} = \mathbf{A}_{N\times N}\vec{x}_{N\times 1}$

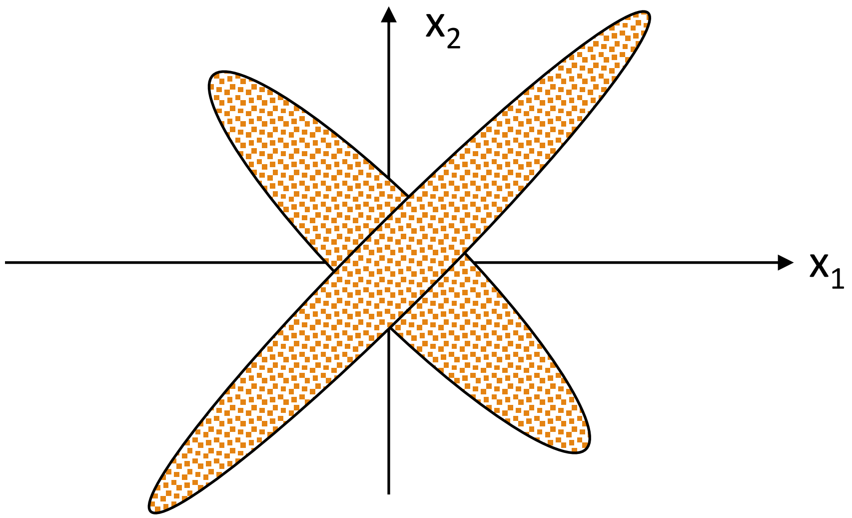
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

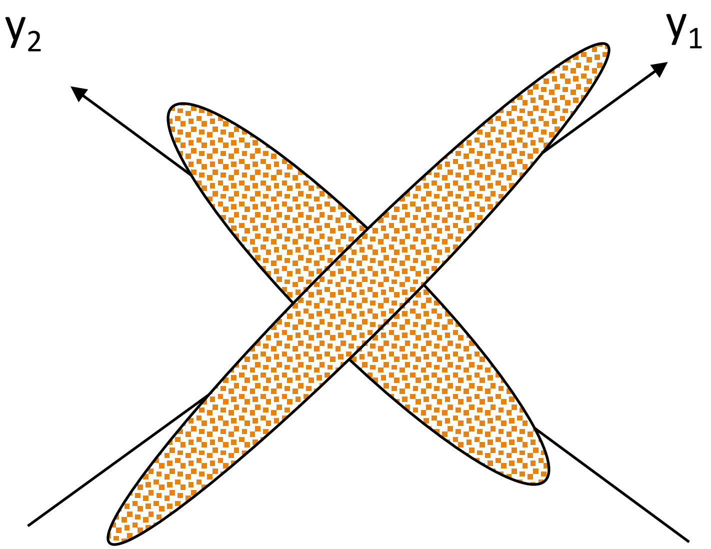
$\Updownarrow$

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i, \quad \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

basis vectors (column vectors of transform matrix)

Decorrelating Property of Transform





x_1 and x_2 are highly correlated

$$p(x_1 x_2) \neq p(x_1)p(x_2)$$

y_1 and y_2 are less correlated

$$p(y_1 y_2) \approx p(y_1)p(y_2)$$

Transform=Change of Coordinates

Intuitively speaking, transform plays the role of facilitating the source modeling

- Due to the decorrelating property of transform, it is easier to model transform coefficients (Y) instead of pixel values (X)

An appropriate choice of transform (transform matrix A) depends on the source statistics P(X)

- We will only consider the class of transforms corresponding to unitary matrices

Unitary Matrix and 1D Unitary Transform

Definition

conjugate

transpose

A matrix A is called unitary if $A^{-1} = A^{*T}$

When the transform matrix A is unitary, the defined transform $\vec{y} = A\vec{x}$ is called unitary transform

Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{A}^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \mathbf{A}^T$$

For a real matrix $\mathbf{A}$, it is unitary if $\mathbf{A}^{-1} = \mathbf{A}^T$

Inverse of Unitary Transform

For a unitary transform, its inverse is defined by

$$\vec{x} = \mathbf{A}^{-1}\vec{y} = \mathbf{A}^{*T}\vec{y}$$

Inverse Transform

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} a_{11}^* & \cdots & \cdots & a_{N1}^* \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N}^* & \cdots & \cdots & a_{NN}^* \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$\Updownarrow$

$$\vec{x} = \sum_{i=1}^{N} y_i \vec{b}_i, \vec{b}_i = [a_{1i}^*, \ldots, a_{Ni}^*]^T$$

basis vectors corresponding to inverse transform

Properties of Unitary Transform

Energy compaction: only few transform coefficients have large magnitude

- Such property is related to the decorrelating role of unitary transform

Energy conservation: unitary transform preserves the 2-norm of input vectors

- Such property essentially comes from the fact that rotating coordinates does not affect Euclidean distance

Energy Compaction Example

Hadamard matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix}$$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix} = \begin{bmatrix} 198 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

significant

insignificant

Energy Conservation

$$\vec{y} = \mathbf{A}\vec{x}$$ A is unitary $\Rightarrow$ $$\|\vec{y}\|^2 = \|\vec{x}\|^2$$

<u>Proof</u>

$$\|\vec{y}\|^2 = \sum_{i=1}^{N} |y_i|^2 = \vec{y}^{*T}\vec{y} = (\mathbf{A}\vec{x})^{*T}(\mathbf{A}\vec{x})$$

$$= \vec{x}^{*T}(\mathbf{A}^{*T}\mathbf{A})\vec{x} = \vec{x}^{*T}\vec{x} = \sum_{i=1}^{N} |x_i|^2 = \|\vec{x}\|^2$$

Numerical Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\Downarrow$$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

<u>Check:</u>

$$\|\vec{x}\|^2 = 3^2 + 4^2 = 25, \|\vec{y}\|^2 = \frac{7^2 + 1^2}{2} = 25$$

Implication of Energy Conservation

$$\vec{y} = [y_1, \ldots, y_N]^T \xrightarrow{\;\boxed{Q}\;} \hat{\vec{y}} = [\hat{y}_1, \ldots, \hat{y}_N]^T$$

$$\vec{x} = [x_1, \ldots, x_N]^T \xrightarrow{\;\boxed{T}\;} \vec{y}$$

$$\hat{\vec{y}} \xrightarrow{\;\boxed{T^{-1}}\;} \hat{\vec{x}} = [\hat{x}_1, \ldots, \hat{x}_N]^T$$

$$\vec{y} = \mathbf{A}\vec{x} \qquad \hat{\vec{y}} = \mathbf{A}\hat{\vec{x}}$$

$$\vec{y} - \hat{\vec{y}} = \mathbf{A}(\vec{x} - \hat{\vec{x}})$$

A is unitary

$$\| \vec{y} - \hat{\vec{y}} \|^2 = \| \vec{x} - \hat{\vec{x}} \|^2$$

Summary of 1D Unitary Transform

Unitary matrix: $\mathbf{A}^{-1}=\mathbf{A}^{*T}$

Unitary transform: $\vec{y} = \mathbf{A}\vec{x}$ $\mathbf{A}$ unitary

Properties of 1D unitary transform

- Energy compaction: most of transform coefficients y_i are small
- Energy conservation: quantization can be directly performed to transform coefficients

$$\|\vec{y}\|^2 = \|\vec{x}\|^2 \Rightarrow \|\vec{y} - \hat{\vec{y}}\|^2 = \|\vec{x} - \hat{\vec{x}}\|^2$$

From 1D to 2D

Do N 1D transforms in parallel

$$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix}$$

$$[\vec{y}_1 \,|\, \ldots \,|\, \vec{y}_i \,|\, \ldots \,|\, \vec{y}_N] \qquad \Updownarrow \qquad [\vec{x}_1 \,|\, \ldots \,|\, \vec{x}_i \,|\, \ldots \,|\, \vec{x}_N]$$

$$\vec{y}_i = \mathbf{A}\vec{x}_i, i = 1,2,\ldots.N$$

$$\vec{x}_i = [x_{i1},\ldots,x_{iN}]^T, \vec{y}_i = [y_{i1},\ldots,y_{iN}]^T$$

Definition of 2D Transform

2D forward transform $$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}^{T}_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D column transform}} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix} \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{N1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D row transform}}$$

2D Transform (Two Sequential 1D Transforms)

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T$$

$$\Updownarrow$$

column transform $\quad \mathbf{Y}_1 = \mathbf{A}\mathbf{X}$ (left matrix multiplication first)

row transform $\quad \mathbf{Y} = \mathbf{Y}_1\mathbf{A}^T = (\mathbf{A}\mathbf{Y}_1^T)^T$

$$\Updownarrow$$

row transform $\quad \mathbf{Y}_2 = \mathbf{X}\mathbf{A}^T = (\mathbf{A}\mathbf{X}^T)^T$ (right matrix multiplication first)

column transform $\quad \mathbf{Y} = \mathbf{A}\mathbf{Y}_2$

Conclusion:

- 2D separable transform can be decomposed into two sequential
- The ordering of 1D transforms does not matter

From Basis Vectors to Basis Images

1D transform matrix $\mathbf{A}$ consists of basis vectors (column vectors)

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

2D transform corresponds to a collection of N-by-N basis images

$$\mathbf{Y} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij} \mathbf{B}_{ij} , \mathbf{B}_{ij} = \vec{b}_i \vec{b}_j^{\,T} , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

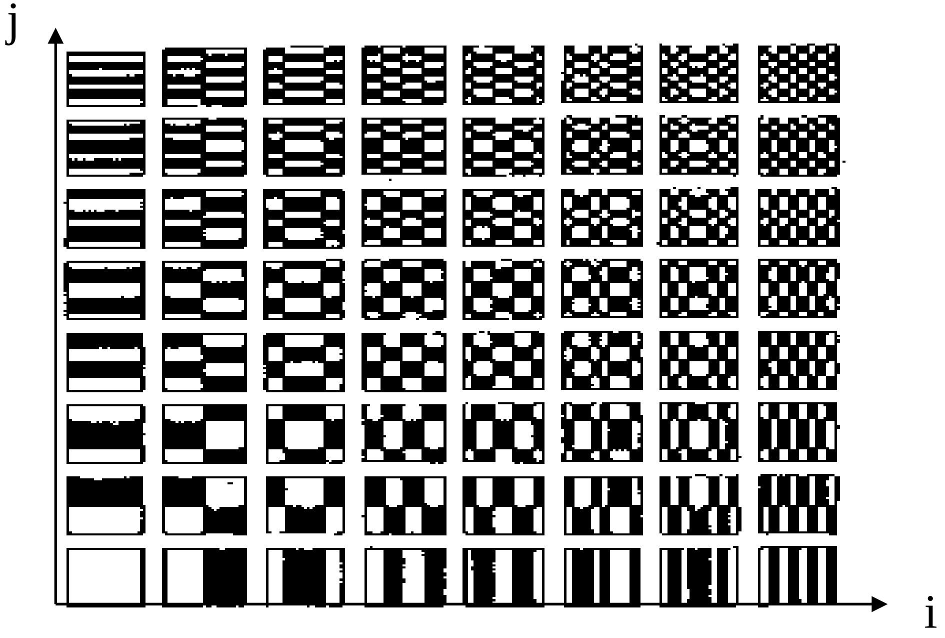
basis image

Example of Basis Images

Hadamard matrix: $$\mathbf{A}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \mathbf{A}_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{A}_n & \mathbf{A}_n \\ \mathbf{A}_n & -\mathbf{A}_n \end{bmatrix}$$

$$\mathbf{A} = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix},$$

$\vec{b}_1$ $\vec{b}_8$



B_{ij}

2D Unitary Transform

Suppose A is a unitary matrix,

forward transform

$$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}_{N\times N}^{T}$$

inverse transform

$$\mathbf{X}_{N\times N} = \mathbf{A}_{N\times N}^{*T}\mathbf{Y}_{N\times N}\mathbf{A}_{N\times N}^{*}$$

Proof

Since **A** is a unitary matrix, we have

$$\mathbf{A}^{-1} = \mathbf{A}^{*T}$$

$$\mathbf{A}^{*T}\mathbf{Y}\mathbf{A}^{*} = \mathbf{A}^{*T}(\mathbf{A}\mathbf{X}\mathbf{A}^{T})\ \mathbf{A}^{*} = \mathbf{I}\cdot\mathbf{X}\cdot\mathbf{I} = \mathbf{X}$$

Energy Compaction Property of 2D Unitary Transform

- Example

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 100 & 100 & 98 & 99 \\ 100 & 100 & 94 & 94 \\ 98 & 97 & 96 & 100 \\ 100 & 99 & 97 & 94 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T} \mathbf{Y} = \begin{bmatrix} 391.5 & 0 & 5.5 & 1 \\ 2.5 & -2 & -4.5 & 2 \\ 1 & -0.5 & 2 & -0.5 \\ 2 & 1.5 & 0 & -1.5 \end{bmatrix}$$

A coefficient is called significant if its magnitude is above a pre-selected threshold *th*

insignificant coefficients (*th*=64)

Energy Conservation Property of 2D Unitary Transform

2-norm of a matrix $\mathbf{X}$

$$\|\mathbf{X}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} |x_{ij}|^2$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T \quad A \text{ unitary} \quad \Rightarrow \quad \|\mathbf{Y}\|^2 = \|\mathbf{X}\|^2$$

<u>Example:</u>

$$\mathbf{A} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T} \mathbf{Y} = \begin{bmatrix} 5 & -1 \\ -2 & 0 \end{bmatrix}$$

$$\|\mathbf{X}\|^2 = 1^2 + 2^2 + 3^2 + 4^2 = 30 = 5^2 + 2^2 + 1^2 + 0^2 = \|\mathbf{Y}\|^2$$

You are asked to prove such property in your homework

Implication of Energy Conservation

$$\mathbf{X} \longrightarrow \boxed{\mathrm{T}} \longrightarrow \mathbf{Y} \longrightarrow \boxed{\mathrm{Q}} \longrightarrow \hat{\mathbf{Y}} \longrightarrow \boxed{\mathrm{T}^{-1}} \longrightarrow \hat{\mathbf{X}}$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^{T} \qquad \hat{\mathbf{X}} = \mathbf{A}^{*T}\hat{\mathbf{Y}}\mathbf{A}^{*}$$

$$\mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{A}(\mathbf{X} - \hat{\mathbf{X}})\mathbf{A}^{T}$$

$$\Downarrow$$

$$\| \mathbf{Y} - \hat{\mathbf{Y}} \|^2 = \| \mathbf{X} - \hat{\mathbf{X}} \|^2$$

Similar to 1D case, quantization noise in the transform domain has the same energy as that in the spatial domain

Important 2D Unitary Transforms

Discrete Fourier Transform

- Widely used in non-coding applications (frequency-domain approaches)

Discrete Cosine Transform

- Used in JPEG standard

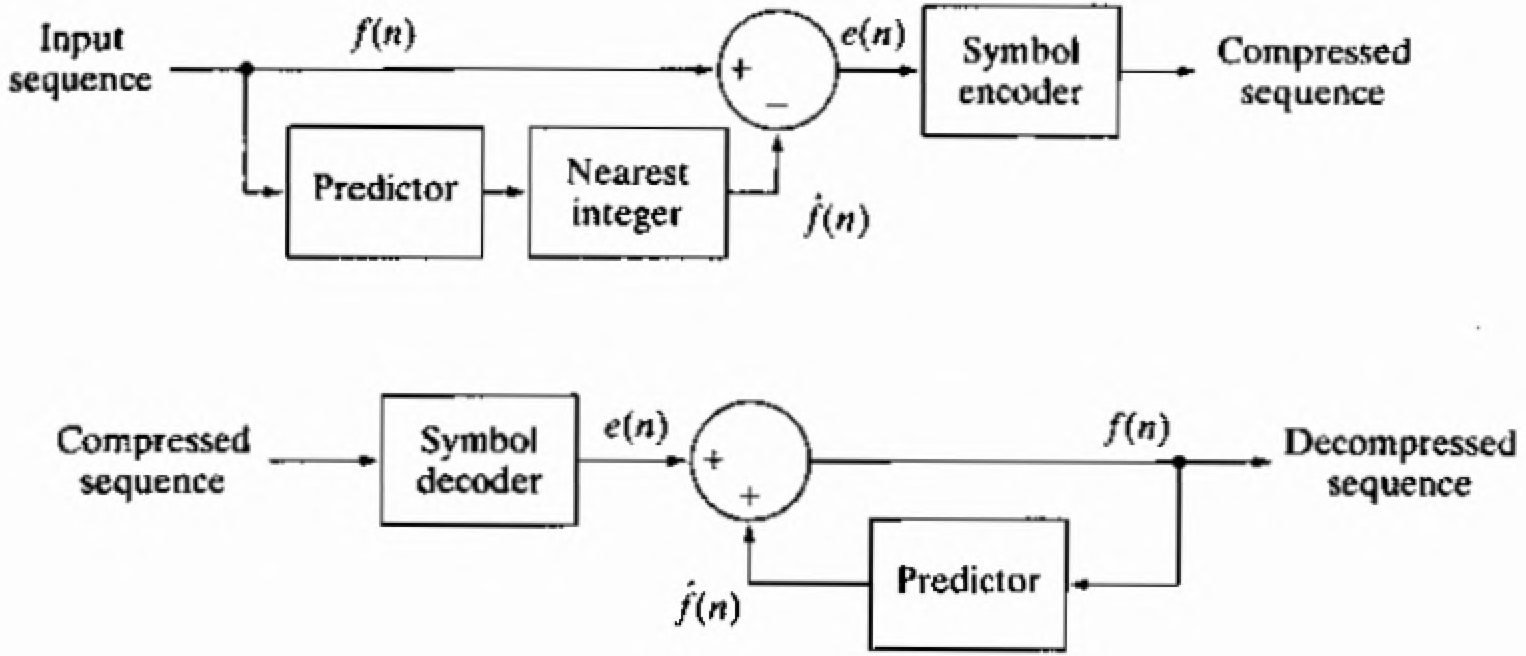
Hadamard Transform

- All entries are ± 1
- N=2: Haar Transform (simplest wavelet transform for multi-resolution analysis)

Predictive Coding

a
b

FIGURE 8.33
A lossless predictive coding model:
(a) encoder;
(b) decoder.



Suggested Readings

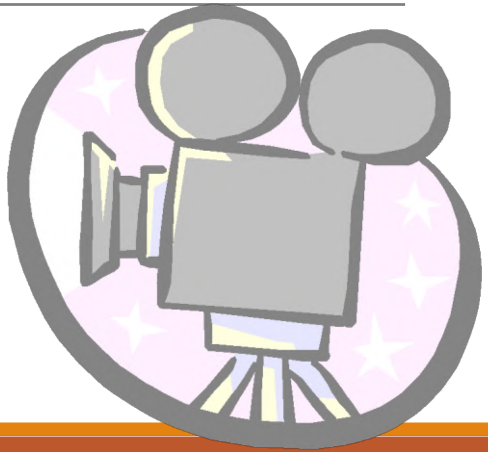
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

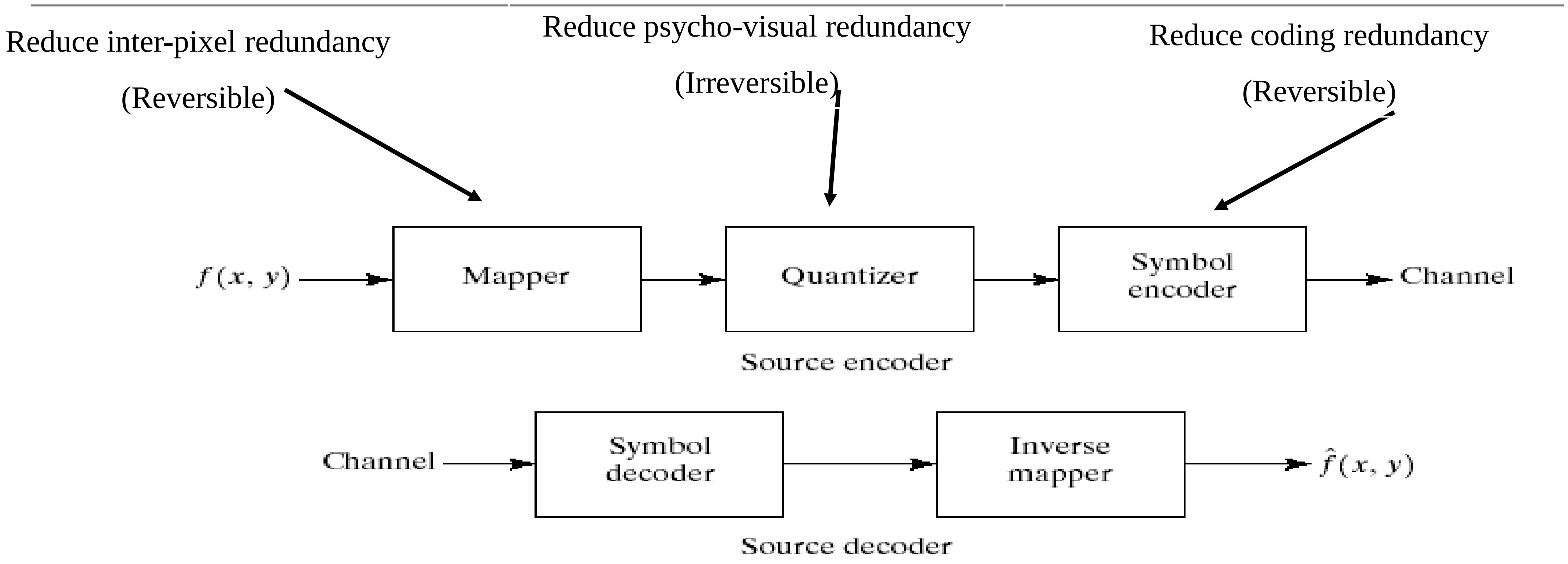
CS-317/CS-341



Outline

- Basic Transformations

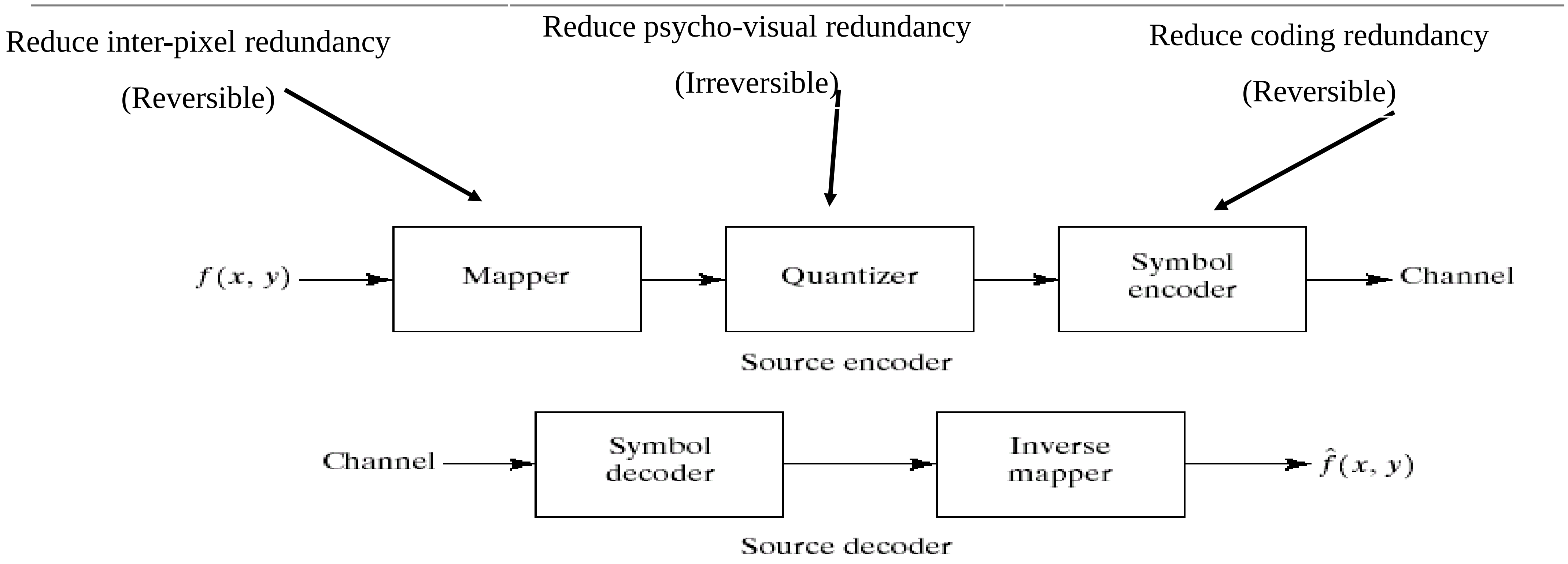
Image Compression model



(a) Source encoder and (b) source decoder model.

Basic Transformation

Image Compression model : Transformation

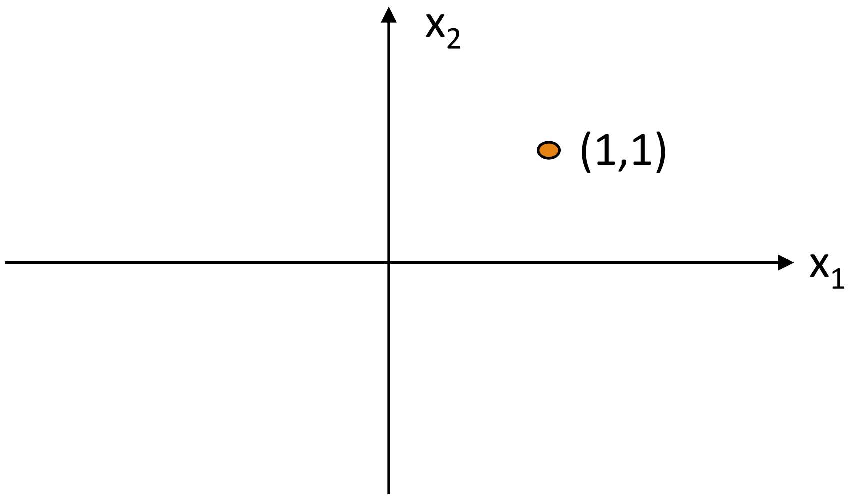


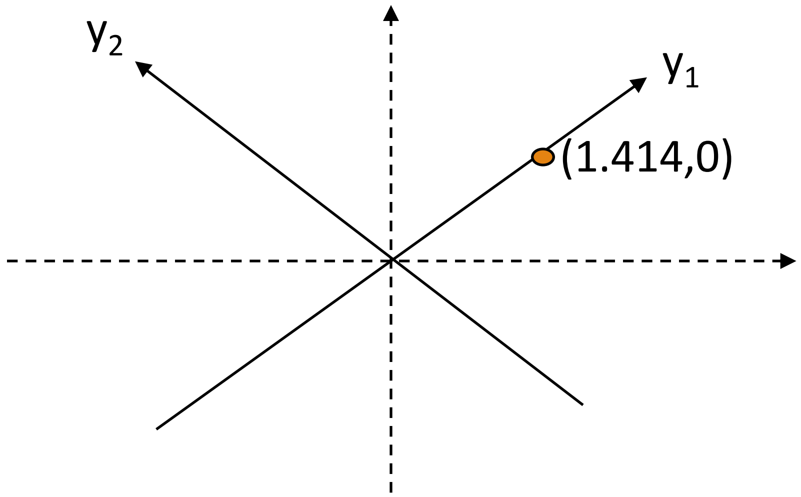
(a) Source encoder and (b) source decoder model.

Lossy Image Compression

- Lossy transform coding
  - Image Transforms (Discrete Cosine Transform)
  - Joint Photographic Expert Group (JPEG)

An Example of 1D Transform with Two Variables





$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \vec{y} = \mathbf{A}\vec{x}, \mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Transform matrix

Generalization into N Variables

forward transform $$\vec{y}_{N\times 1} = \mathbf{A}_{N\times N}\vec{x}_{N\times 1}$$

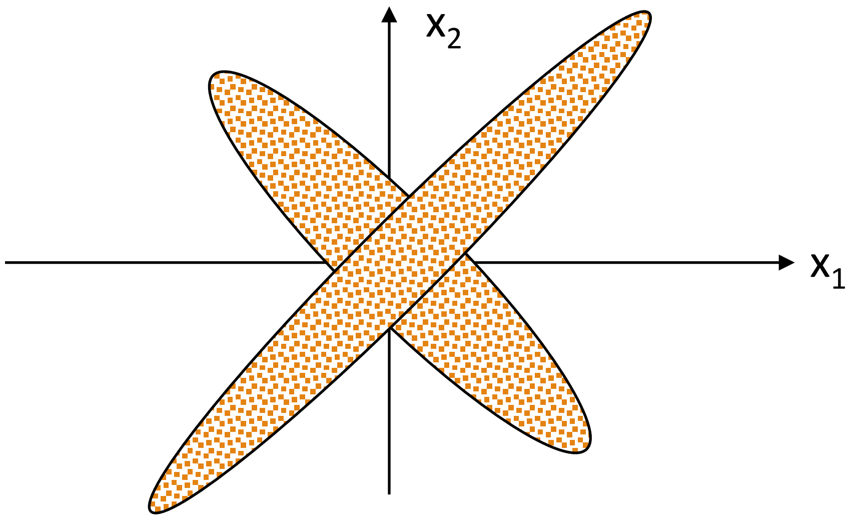
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$\Updownarrow$$

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i, \quad \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

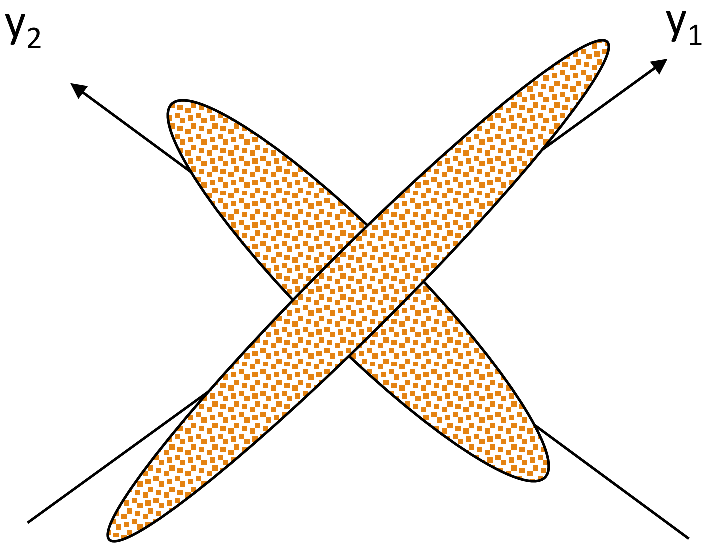
basis vectors (column vectors of transform matrix)

Decorrelating Property of Transform



x_1 and x_2 are highly correlated

$p(x_1 x_2) \neq p(x_1)p(x_2)$



y_1 and y_2 are less correlated

$p(y_1 y_2) \approx p(y_1)p(y_2)$

Transform=Change of Coordinates

Intuitively speaking, transform plays the role of facilitating the source modeling

- Due to the decorrelating property of transform, it is easier to model transform coefficients (Y) instead of pixel values (X)

An appropriate choice of transform (transform matrix A) depends on the source statistics P(X)

- We will only consider the class of transforms corresponding to unitary matrices

Unitary Matrix and 1D Unitary Transform

Definition

conjugate transpose

A matrix A is called unitary if $A^{-1}=A^{*T}$

When the transform matrix A is unitary, the defined transform $\vec{y} = A\vec{x}$ is called unitary transform

Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{A}^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \mathbf{A}^{T}$$

For a real matrix **A**, it is unitary if $\mathbf{A}^{-1}=\mathbf{A}^{T}$

Inverse of Unitary Transform

For a unitary transform, its inverse is defined by

$$\vec{x} = \mathbf{A}^{-1}\vec{y} = \mathbf{A}^{*T}\vec{y}$$

Inverse Transform

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} a_{11}^* & \cdots & \cdots & a_{N1}^* \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N}^* & \cdots & \cdots & a_{NN}^* \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$\Updownarrow$$

$$\vec{x} = \sum_{i=1}^{N} y_i \vec{b}_i, \; \vec{b}_i = [a_{1i}^*, \ldots, a_{Ni}^*]^T$$

basis vectors corresponding to inverse transform

Properties of Unitary Transform

Energy compaction: only few transform coefficients have large magnitude

- Such property is related to the decorrelating role of unitary transform

Energy conservation: unitary transform preserves the 2-norm of input vectors

- Such property essentially comes from the fact that rotating coordinates does not affect Euclidean distance

Energy Compaction Example

Hadamard matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix}$$

$$\Downarrow$$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix} = \begin{bmatrix} 198 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

significant

insignificant

Energy Conservation

$$\vec{y} = \mathbf{A}\vec{x}$$ A is unitary $\Rightarrow$ $$\|\vec{y}\|^2 = \|\vec{x}\|^2$$

<u>Proof</u>

$$\|\vec{y}\|^2 = \sum_{i=1}^{N} |y_i|^2 = \vec{y}^{*T}\vec{y} = (\mathbf{A}\vec{x})^{*T}(\mathbf{A}\vec{x})$$

$$= \vec{x}^{*T}(\mathbf{A}^{*T}\mathbf{A})\vec{x} = \vec{x}^{*T}\vec{x} = \sum_{i=1}^{N} |x_i|^2 = \|\vec{x}\|^2$$

Numerical Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\Downarrow$$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

<u>Check:</u>

$$\| \vec{x} \|^2 = 3^2 + 4^2 = 25, \| \vec{y} \|^2 = \frac{7^2 + 1^2}{2} = 25$$

Implication of Energy Conservation

$$\vec{y} = [y_1, \ldots, y_N]^T \xrightarrow{Q} \hat{\vec{y}} = [\hat{y}_1, \ldots, \hat{y}_N]^T$$

$$\vec{x} = [x_1, \ldots, x_N]^T \xrightarrow{T} \vec{y} \qquad \hat{\vec{y}} \xrightarrow{T^{-1}} \hat{\vec{x}} = [\hat{x}_1, \ldots, \hat{x}_N]^T$$

$$\vec{y} = \mathbf{A}\vec{x} \qquad \hat{\vec{y}} = \mathbf{A}\hat{\vec{x}}$$

$$\vec{y} - \hat{\vec{y}} = \mathbf{A}(\vec{x} - \hat{\vec{x}})$$

$\Downarrow$ **A** is unitary

$$\| \vec{y} - \hat{\vec{y}} \|^2 = \| \vec{x} - \hat{\vec{x}} \|^2$$

Summary of 1D Unitary Transform

Unitary matrix: $\mathbf{A}^{-1}=\mathbf{A}^{*T}$

Unitary transform: $\vec{y} = \mathbf{A}\vec{x}$ $\mathbf{A}$ unitary

Properties of 1D unitary transform

- Energy compaction: most of transform coefficients y_i are small
- Energy conservation: quantization can be directly performed to transform coefficients

$$\| \vec{y} \|^2 = \| \vec{x} \|^2 \Rightarrow \| \vec{y} - \hat{\vec{y}} \|^2 = \| \vec{x} - \hat{\vec{x}} \|^2$$

From 1D to 2D

Do N 1D transforms in parallel

$$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix}$$

$$[\vec{y}_1 \mid \ldots \mid \vec{y}_i \mid \ldots \mid \vec{y}_N] \qquad [\vec{x}_1 \mid \ldots \mid \vec{x}_i \mid \ldots \mid \vec{x}_N]$$

$$\vec{y}_i = \mathbf{A}\vec{x}_i, i = 1,2,\ldots.N$$

$$\vec{x}_i = [x_{i1},\ldots,x_{iN}]^T, \vec{y}_i = [y_{i1},\ldots,y_{iN}]^T$$

Definition of 2D Transform

2D forward transform $$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}^T_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D column transform}} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix} \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{N1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D row transform}}$$

2D Transform (Two Sequential 1D Transforms)

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T$$

column transform $\mathbf{Y}_1 = \mathbf{A}\mathbf{X}$ (left matrix multiplication first)

row transform $\mathbf{Y} = \mathbf{Y}_1\mathbf{A}^T = (\mathbf{A}\mathbf{Y}_1^T)^T$

row transform $\mathbf{Y}_2 = \mathbf{X}\mathbf{A}^T = (\mathbf{A}\mathbf{X}^T)^T$ (right matrix multiplication first)

column transform $\mathbf{Y} = \mathbf{A}\mathbf{Y}_2$

<u>Conclusion:</u>

- 2D separable transform can be decomposed into two sequential
- The ordering of 1D transforms does not matter

From Basis Vectors to Basis Images

1D transform matrix $\mathbf{A}$ consists of basis vectors (column vectors)

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

2D transform corresponds to a collection of N-by-N basis images

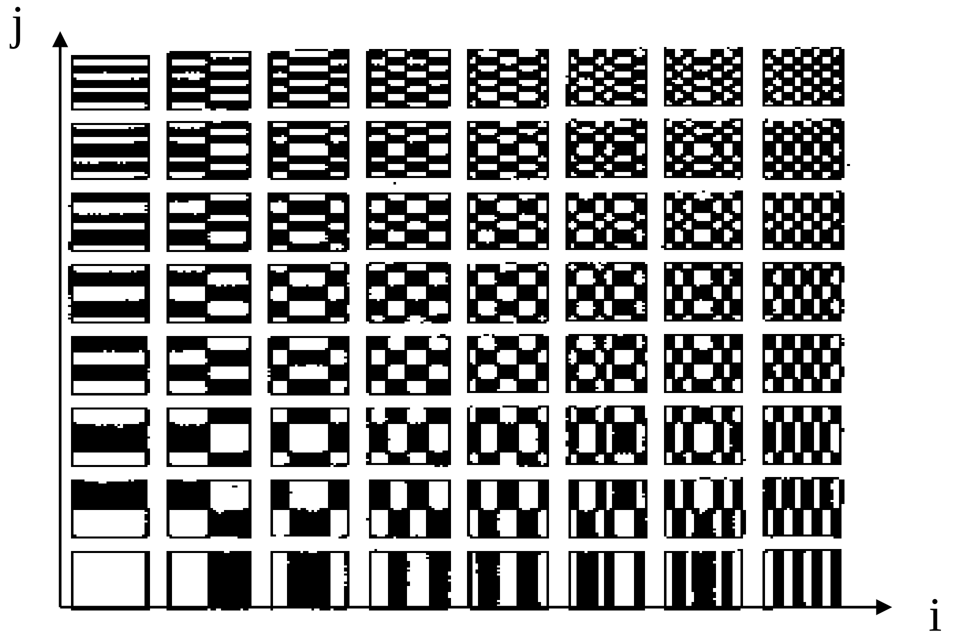
$$\mathbf{Y} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij} \mathbf{B}_{ij} , \mathbf{B}_{ij} = \vec{b}_i \vec{b}_j^{\,T} , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

basis image

Example of Basis Images

Hadamard matrix: $$\mathbf{A}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \mathbf{A}_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{A}_n & \mathbf{A}_n \\ \mathbf{A}_n & -\mathbf{A}_n \end{bmatrix}$$

$$\mathbf{A} = \frac{1}{\sqrt{8}}\begin{array}{c} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}, \\ \begin{array}{cccccccc} \uparrow & & & & & & & \uparrow \\ \vec{b}_1 & & & & & & & \vec{b}_8 \end{array} \end{array}$$



B_{ij}

2D Unitary Transform

Suppose A is a unitary matrix,

forward transform

$$\mathbf{Y}_{N\times N}=\mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}_{N\times N}^{T}$$

inverse transform

$$\mathbf{X}_{N\times N}=\mathbf{A}_{N\times N}^{*T}\mathbf{Y}_{N\times N}\mathbf{A}_{N\times N}^{*}$$

Proof

Since **A** is a unitary matrix, we have

$$\mathbf{A}^{-1}=\mathbf{A}^{*T}$$

$$\mathbf{A}^{*T}\mathbf{Y}\mathbf{A}^{*}=\mathbf{A}^{*T}(\mathbf{A}\mathbf{X}\mathbf{A}^{T})\ \mathbf{A}^{*}=\mathbf{I}\cdot\mathbf{X}\cdot\mathbf{I}=\mathbf{X}$$

Energy Compaction Property of 2D Unitary Transform

- Example

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 100 & 100 & 98 & 99 \\ 100 & 100 & 94 & 94 \\ 98 & 97 & 96 & 100 \\ 100 & 99 & 97 & 94 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{AXA}^T} \mathbf{Y} = \begin{bmatrix} 391.5 & 0 & 5.5 & 1 \\ 2.5 & -2 & -4.5 & 2 \\ 1 & -0.5 & 2 & -0.5 \\ 2 & 1.5 & 0 & -1.5 \end{bmatrix}$$

A coefficient is called significant if its magnitude is above a pre-selected threshold *th*

insignificant coefficients (*th*=64)

Energy Conservation Property of 2D Unitary Transform

2-norm of a matrix $\mathbf{X}$ $$\|\mathbf{X}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} |x_{ij}|^2$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T \quad A \text{ unitary} \quad \Longrightarrow \quad \|\mathbf{Y}\|^2 = \|\mathbf{X}\|^2$$

<u>Example:</u>

$$\mathbf{A} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T} \mathbf{Y} = \begin{bmatrix} 5 & -1 \\ -2 & 0 \end{bmatrix}$$

$$\|\mathbf{X}\|^2 = 1^2 + 2^2 + 3^2 + 4^2 = 30 = 5^2 + 2^2 + 1^2 + 0^2 = \|\mathbf{Y}\|^2$$

You are asked to prove such property in your homework

Implication of Energy Conservation

$$\mathbf{X} \longrightarrow \boxed{\mathrm{T}} \longrightarrow \mathbf{Y} \longrightarrow \boxed{\mathrm{Q}} \longrightarrow \hat{\mathbf{Y}} \longrightarrow \boxed{\mathrm{T}^{-1}} \longrightarrow \hat{\mathbf{X}}$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T \qquad \hat{\mathbf{X}} = \mathbf{A}^{*T}\hat{\mathbf{Y}}\mathbf{A}^*$$

$$\mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{A}(\mathbf{X} - \hat{\mathbf{X}})\mathbf{A}^T$$

$$\Downarrow$$

$$\| \mathbf{Y} - \hat{\mathbf{Y}} \|^2 = \| \mathbf{X} - \hat{\mathbf{X}} \|^2$$

Similar to 1D case, quantization noise in the transform domain has the same energy as that in the spatial domain

Important 2D Unitary Transforms

Discrete Fourier Transform

- Widely used in non-coding applications (frequency-domain approaches)

Discrete Cosine Transform

- Used in JPEG standard

Hadamard Transform

- All entries are ± 1
- N=2: Haar Transform (simplest wavelet transform for multi-resolution analysis)

Transform Coding

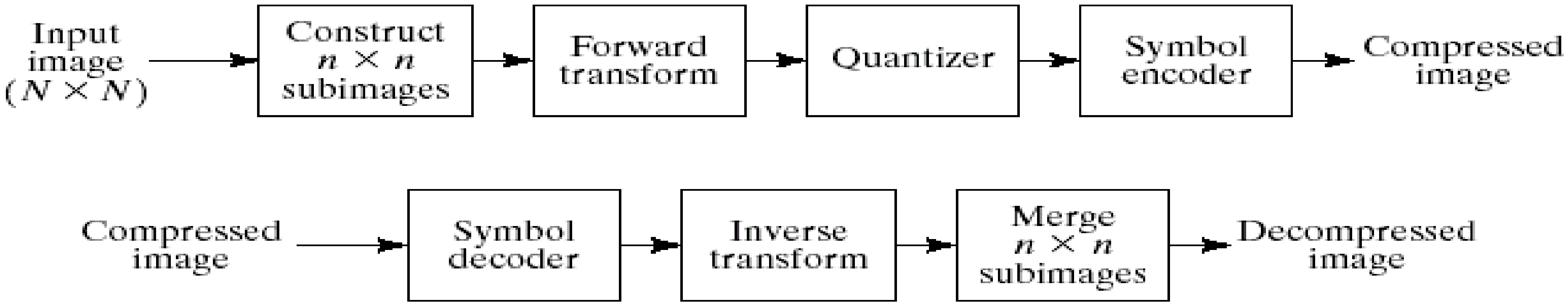
A reversible linear transform (such as Fourier Transform) is used to map the image into a set of transform coefficients

These coefficients are then quantized and coded.

The goal of transform coding is to decorrelate pixels and pack as much information into small number of transform coefficients.

Compression is achieved during quantization not during the transform step

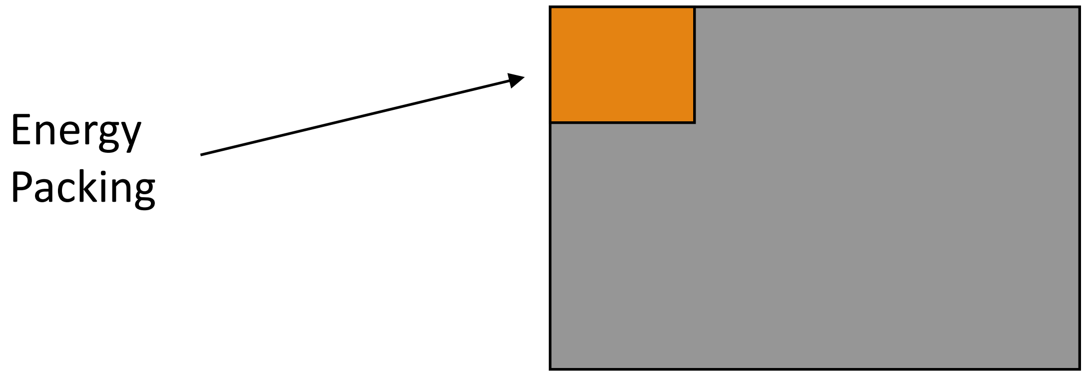
Transform Coding



2D Transforms

Energy packing

- 2D transforms pack most of the energy into small number of coefficients located at the upper left corner of the 2D array



2D Transforms

Consider an image *f(x,y)* of size *N x N*

Forward transform

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

$$u,v = 0,1,2,\ldots,N-1.$$

g(x,y,u,v) is the forward transformation kernel or basis functions

2D Transforms

Inverse transform

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

$$x,y=0,1,2,\ldots,N-1.$$

h(x,y,u,v) is the inverse transformation kernel or basis functions

Discrete Cosine Transformation (DCT)

DCT (1D)

Discrete cosine transform

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

$$\alpha(k) = \left\{ \begin{array}{ll} \sqrt{\frac{1}{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{array} \right\}.$$

The strength of the ‘u’ sinusoid is given by C(u)

- Project f onto the basis function
- All samples of f contribute the coefficient
- C(0) is the zero-frequency component – the average value!

DCT (1D)

Consider a digital image such that one row has the following samples

Index	0	1	2	3	4	5	6	7
Value	20	12	18	56	83	10	104	114

There are 8 samples so N=8

u is in [0, N-1] or [0, 7]

Must compute 8 DCT coefficients: C(0), C(1), …, C(7)

Start with C(0)

$$C(0) = \sqrt{\frac{1}{N}} \sum_{x}^{N-1} f(x)$$

DCT (1D)

$$
\begin{aligned}
C(0) &= \sqrt{\tfrac{1}{8}}\sum_{x=0}^{7} f(x)\cos\left(\frac{(2x+1)\cdot 0\pi}{2\cdot 8}\right) \\
&= \sqrt{\tfrac{1}{8}}\sum_{x=0}^{7} f(x)\cos(0) \\
&= \sqrt{\tfrac{1}{8}}\cdot\{f(0)+f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)\} \\
&= .35\cdot\{20+12+18+56+83+110+104+115\} \\
&= 183.14
\end{aligned}
$$

DCT (1D)

Repeating the computation for all u, we obtain the following coefficients

Spatial domain

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
20	12	18	56	83	110	104	114

Frequency domain

$C(0)$	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	$C(7)$
183.1	-113.0	-4.2	22.1	10.6	-1.5	4.8	-8.7

DCT (2D)

The 2D DCT is given below where the definition for alpha is the same as before

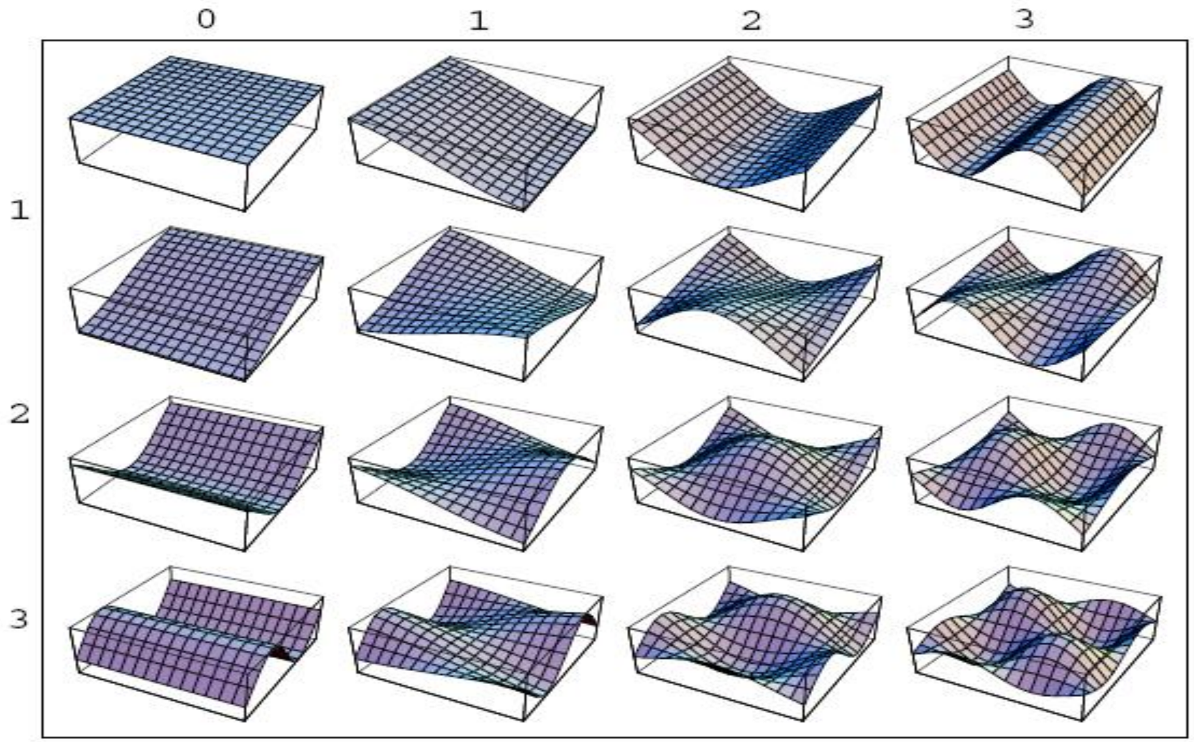
$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

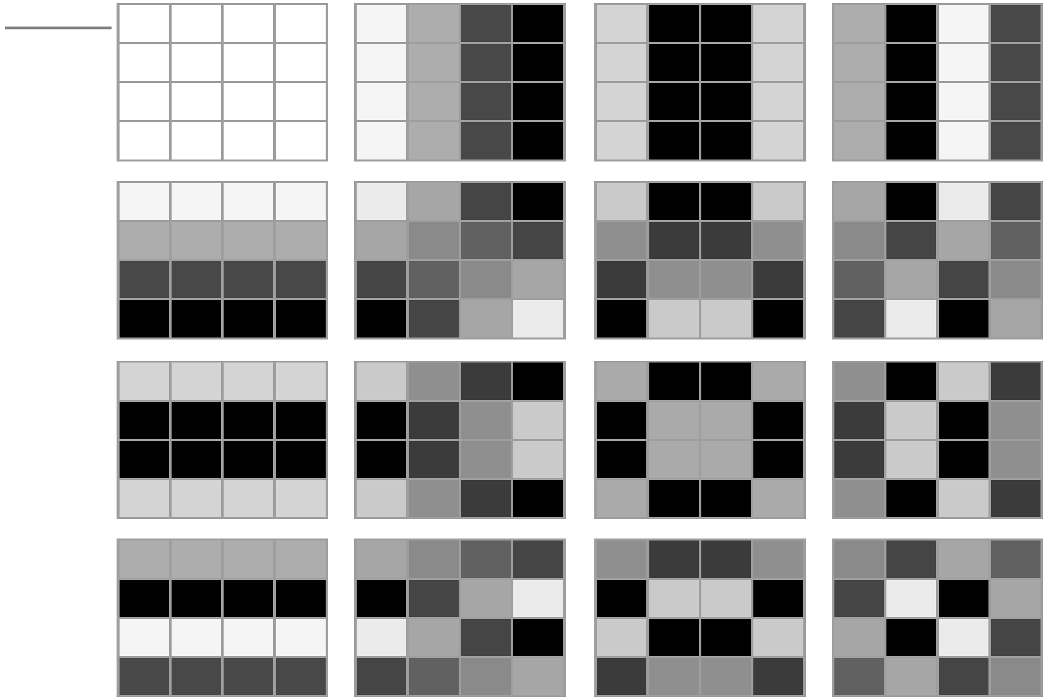
For an MxN image there are MxN coefficients

Each image sample contributes to each coefficient

Each (u,v) pair corresponds to a ‘pattern’ or ‘basis function’

DCT basis functions (patterns)





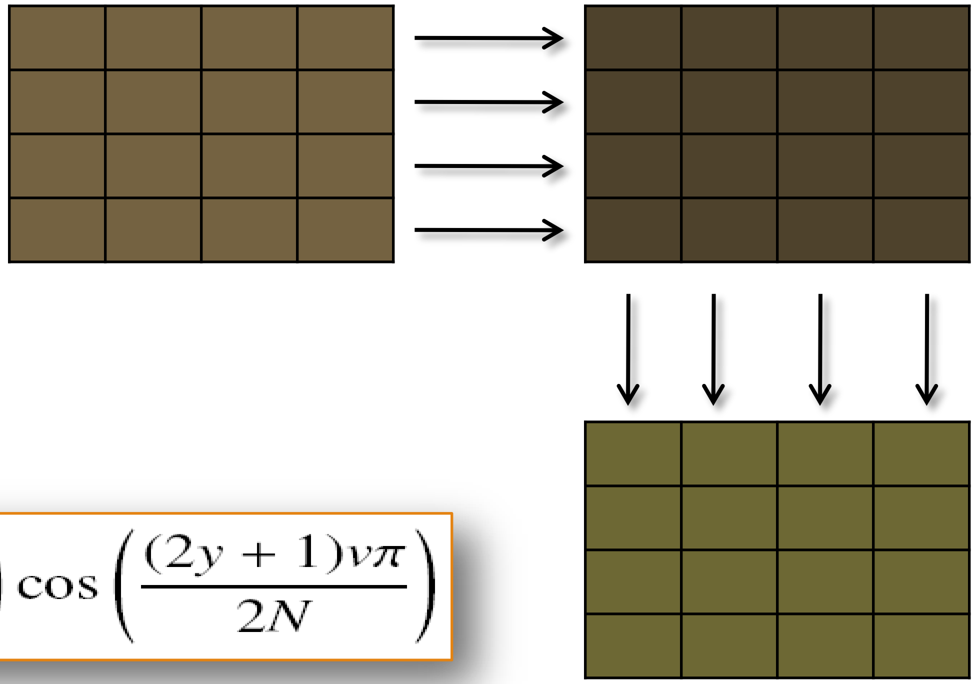
Basis functions

Basis patterns (imaged functions)

Separability

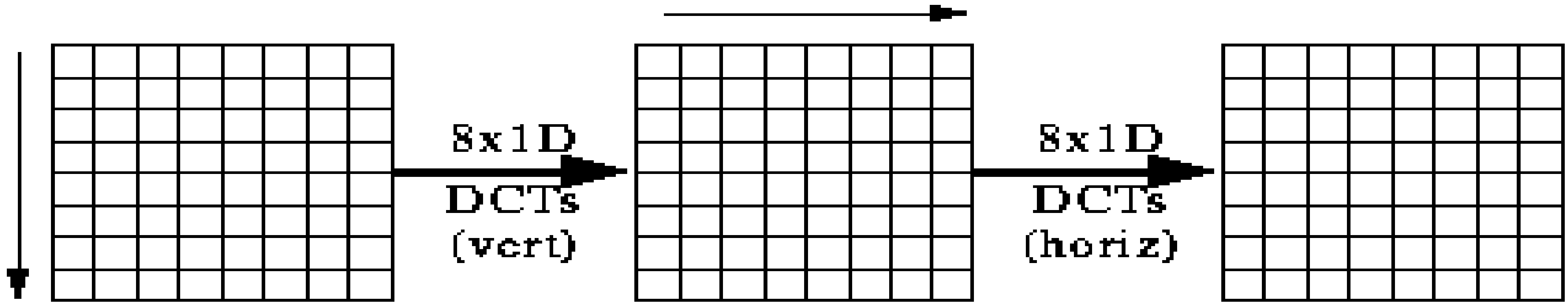
The DCT is separable

- The coefficients can be obtained by computing the 1D coefficients for each row
- Using the row-coefficients to compute the coefficients of each column (using the 1D forward transform)



$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

Separable



Invertability

The DCT is invertible

- Spatial samples can be recovered from the DCT coefficients

$$
\begin{aligned}
f(x) &= \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \\
f(x,y) &= \sum_{u=0}^{N-1} \sum_{v}^{N-1} \alpha(u)\alpha(v) C(u,v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)
\end{aligned}
$$

Summary of DCT

- The DCT provides energy compaction
  - Low frequency coefficients have larger magnitude (typically)
  - High frequency coefficients have smaller magnitude (typically)
  - Most information is compacted into the lower frequency coefficients (those coefficients at the ‘upper-left’)

- Compaction can be leveraged for compression
  - Use the DCT coefficients to store image data but discard a certain percentage of the high-frequency coefficients!
  - JPEG does this

Example: Energy Compaction

- Original Lena image

- 2D DCT

Example: Energy Compaction

- Original Lena image

(a)

- 2D DCT

(b)

75% coefficients are Discarded

Compressed

98% DC coefficients are discarded

Compressed

Suggested Readings

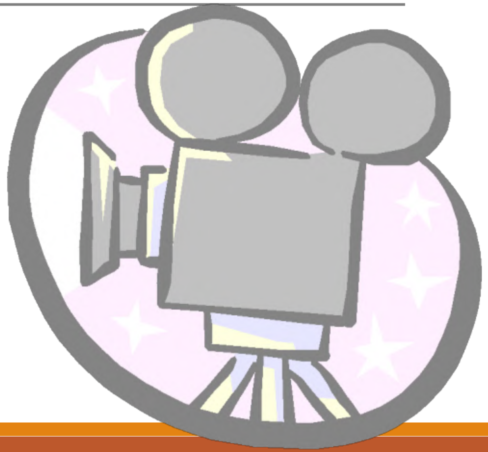
- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you

Image Processing

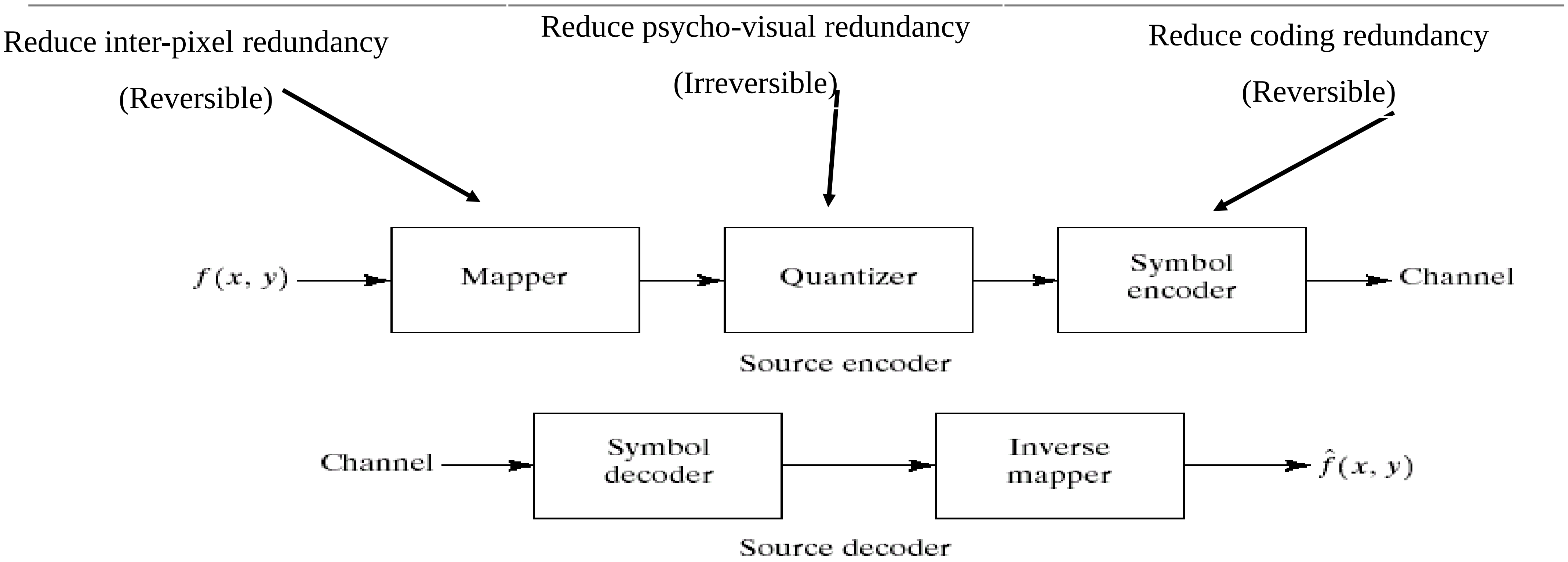
CS-317/CS-341



Outline

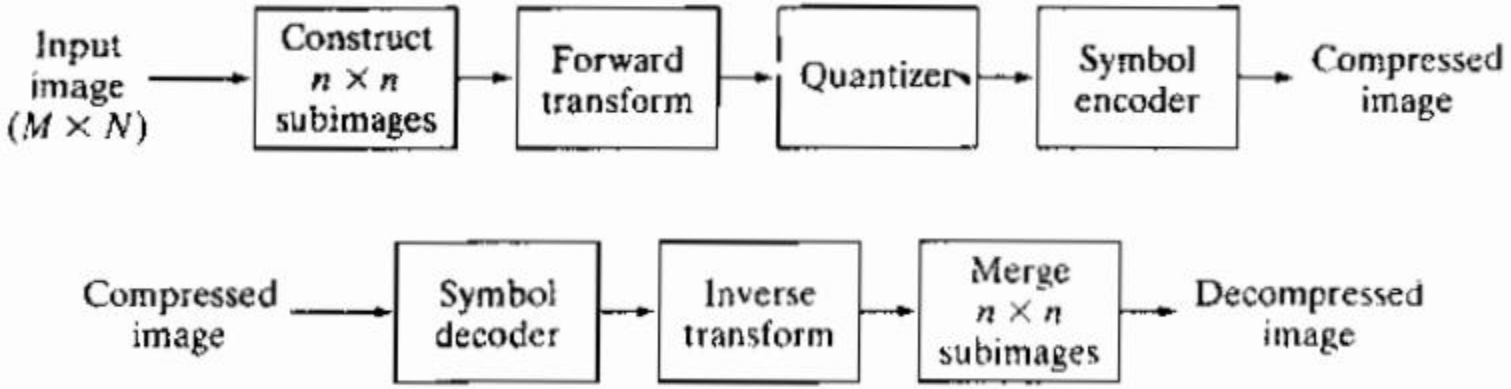
- Lossy image compression techniques
- JPEG Standard

Image Compression model



(a) Source encoder and (b) source decoder model.

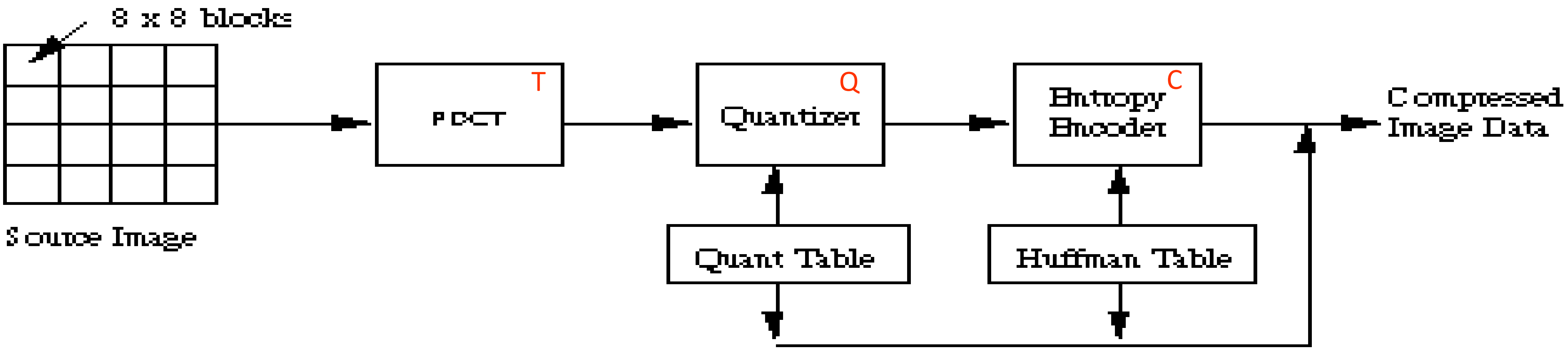
Block Transform Coding



a
b

FIGURE 8.21
A block transform coding system: (a) encoder; (b) decoder.

JPEG Coding Algorithm



Flow-chart diagram of DCT-based coding algorithm specified by Joint Photographic Expert Group (JPEG)

JPEG - Steps

1. Divide image into 8x8 subimages.

For each subimage do:

2. Shift the gray-levels in the range [-128, 127]

3. Apply DCT → 64 coefficients

    1 DC coefficient: F(0,0)

    63 AC coefficients: F(u,v)

4. Quantization

5. Coding

Transform Coding of Images

Why not transform the whole image together?
- Require a large memory to store transform matrix
- It is not a good idea for compression due to spatially varying statistics within an image

Idea of partitioning an image into blocks
- Each block is viewed as a smaller-image and processed independently
- It is not a magic, but a compromise

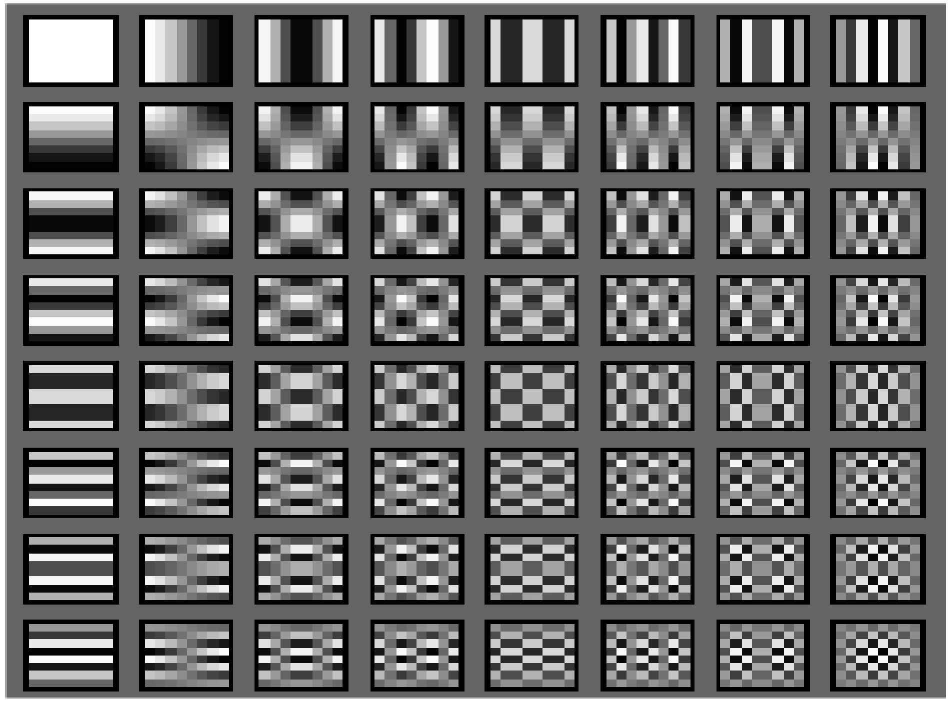
8-by-8 DCT Basis Images

$$\mathbf{A}_{8\times 8} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{18} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{81} & \cdots & \cdots & a_{88} \end{bmatrix}$$

$$a_{kl} = \begin{cases} 1/\sqrt{8}, & k = 1, 1 \le l \le 8 \\ \frac{1}{2}\cos\frac{(2l-1)(k-1)\pi}{16}, & 2 \le k \le 8, 1 \le l \le 8 \end{cases}$$

$$\mathbf{Y} = \sum_{i=1}^{8}\sum_{j=1}^{8} x_{ij}\mathbf{B}_{ij},$$

$$\mathbf{B}_{ij} = \vec{b}_i \vec{b}_j^{\,T}, \vec{b}_i = [a_{i1}, \ldots, a_{i8}]^T$$



Block Processing under MATLAB

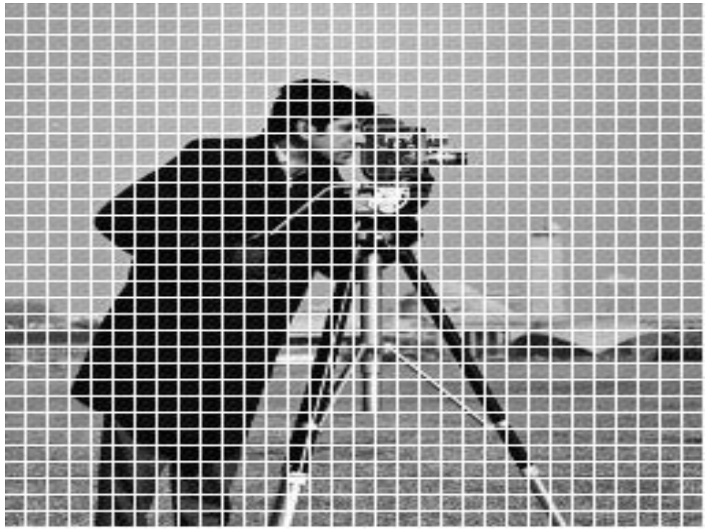
Type “help blkproc” to learn the usage of this function

- B = BLKPROC(A,[M N],FUN) processes the image A by applying the function FUN to each distinct M-by-N block of A, padding A with zeros if necessary.

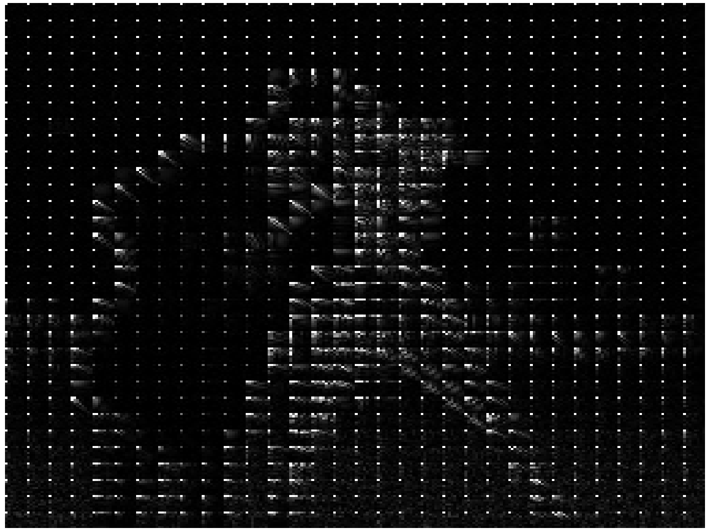
Example

```
I = imread('cameraman.tif');
    fun = @dct2;
    J = blkproc(I,[8 8],fun);
```

Block-based DCT Example



I

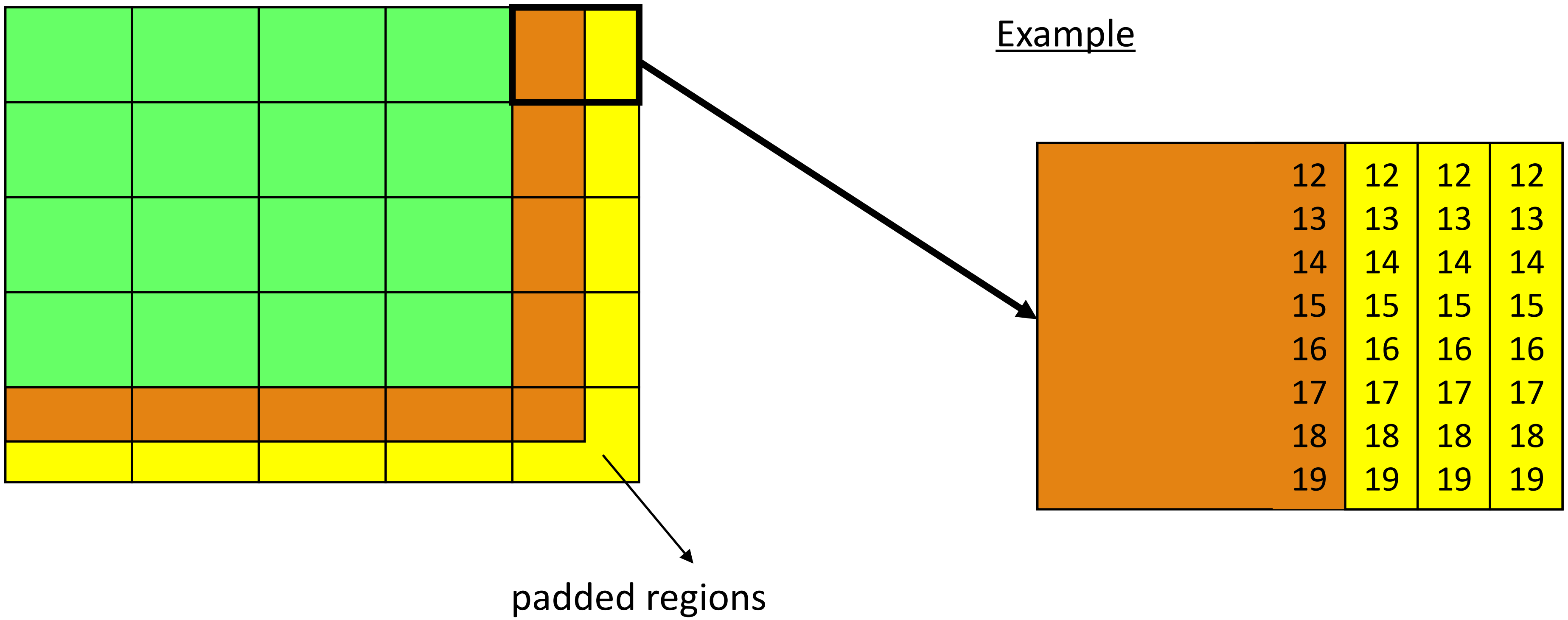


J

note that white lines are artificially added to the border of each 8-by-8 block to denote that each block is processed independently

Boundary Padding

Example



When the width/height of an image is not the multiple of 8, the boundary is artificially padded with repeated columns/rows to make them multiple of 8

Example

$$\begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

Any 8-by-8 block in an image is processed in a similar fashion

Encoding Stage I: Transform

- Step 1: DC level shifting

$$\begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix} \qquad \begin{bmatrix} 55 & 36 & -34 & 25 & 66 & 35 & 4 & 37 \\ 55 & 25 & -12 & 48 & 59 & 38 & 2 & 41 \\ 51 & 40 & 43 & 54 & 51 & 42 & 3 & 39 \\ 49 & 49 & 51 & 49 & 51 & 37 & 3 & 39 \\ 50 & 50 & 51 & 48 & 54 & 36 & 2 & 43 \\ 51 & 52 & 52 & 51 & 55 & 41 & 4 & 41 \\ 51 & 51 & 52 & 54 & 55 & 42 & 1 & 45 \\ 52 & 51 & 53 & 51 & 53 & 42 & 2 & 41 \end{bmatrix}$$

128 (DC level)

$-$

Encoding Step 1: Transform (Con’t)

- Step 2: 8-by-8 DCT

$$
\begin{bmatrix}
55 & 36 & -34 & 25 & 66 & 35 & 4 & 37 \\
55 & 25 & -12 & 48 & 59 & 38 & 2 & 41 \\
51 & 40 & 43 & 54 & 51 & 42 & 3 & 39 \\
49 & 49 & 51 & 49 & 51 & 37 & 3 & 39 \\
50 & 50 & 51 & 48 & 54 & 36 & 2 & 43 \\
51 & 52 & 52 & 51 & 55 & 41 & 4 & 41 \\
51 & 51 & 52 & 54 & 55 & 42 & 1 & 45 \\
52 & 51 & 53 & 51 & 53 & 42 & 2 & 41
\end{bmatrix}
\xrightarrow{\boxed{8\times 8 \text{ DCT}}}
\begin{bmatrix}
313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\
-38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\
-20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\
-10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\
-6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\
2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\
4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\
3 & 1 & 0 & -4 & -2 & -1 & 3 & 1
\end{bmatrix}
$$

Encoding Stage II: Quantization

<u>Q-table</u> : specifies quantization stepsize (see slide #28)

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$f : s_{ij} = \left\lfloor \frac{x_{ij}}{Q_{ij}} \right\rceil$$

$$f^{-1} : \hat{x}_{ij} = s_{ij} \cdot Q_{ij}$$

$$1 \leq i, j \leq 8$$

<u>Notes:</u>

- Q-table can be specified by customer
- Q-table is scaled up/down by a chosen quality factor
- Quantization stepsize Q_{ij} is dependent on the coordinates (i,j) within the 8-by-8 block
- Quantization stepsize Q_{ij} increases from top-left to bottom-right

Encoding Stage II: Quantization (Con't)

<u>Example</u>

$$x_{ij}$$

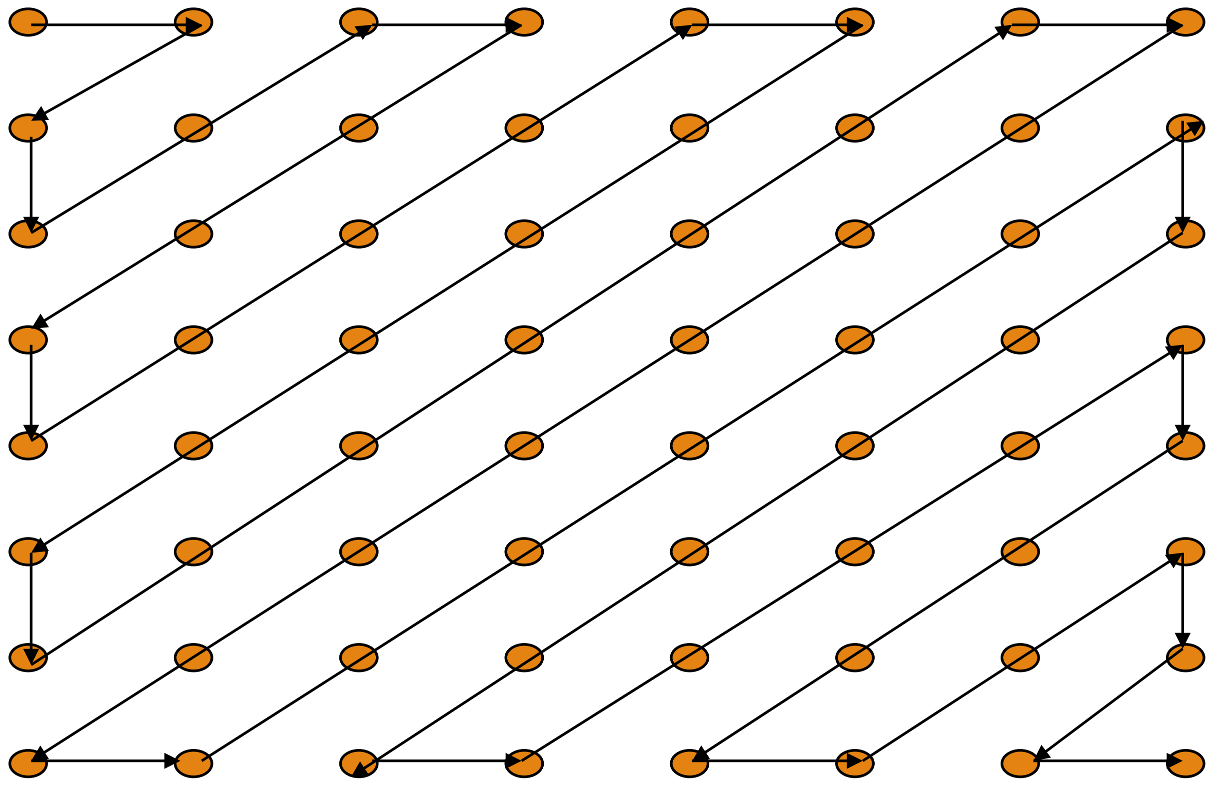
$$\begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix}$$

$$s_{ij}$$

$$\begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$x_{ij} \longrightarrow \boxed{f} \longrightarrow s_{ij}$$

Encoding Stage III: Entropy Coding



Zigzag Scan

$$\begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

zigzag scan

(20,5,-3,-1,-2,-3,1,1,-1,-1,
0,0,1,2,3,-2,1,1,0,0,0,0,0,
0,1,1,0,1,EOB)

End Of the Block:
All following coefficients are zero

Run-length Coding

(20,5,-3,-1,-2,-3,1,1,-1,-1,0,0,1,2,3,-2,1,1,0,0,0,0,0,0,1,1,0,1,EOB)

DC coefficient

AC coefficient

- DC coefficient : DPCM coding → encoded bit stream
- AC coefficient : run-length coding (run, level)

(5,-3,-1,-2,-3,1,1,-1,-1,0,0,1,2,3,-2,1,1,0,0,0,0,0,0,1,1,0,1,EOB)

↓

(0,5),(0,-3),(0,-1),(0,-2),(0,-3),(0,1),(0,1),(0,-1),(0,-1),(2,0),(0,1),
(0,2),(0,3),(0,-2),(0,1),(0,1),(6,0),(0,1),(0,1),(1,0),(0,1),EOB

↓ Huffman coding

encoded bit stream

JPEG Decoding Stage I: Entropy Decoding

encoded bit stream

↓ Huffman decoding

(0,5),(0,-3),(0,-1),(0,-2),(0,-3),(0,1),(0,1),(0,-1),(0,-1),(2,0),(0,1),
(0,2),(0,3),(0,-2),(0,1),(0,1),(6,0),(0,1),(0,1),(1,0),(0,1),EOB

encoded bit stream

↓

DPCM decoding

↓

(20,5,-3,-1,-2,-3,1,1,-1,-1,0,0,1,2,3,-2,1,1,0,0,0,0,0,0,1,1,0,1,EOB)

DC coefficient

AC coefficients

JPEG Decoding Stage II: Inverse Quantization

(20,5,-3,-1,-2,-3,1,1,-1,-1,0,0,1,2,3,-2,1,1,0,0,0,0,0,0,1,1,0,1,EOB)

$\downarrow$ zigzag

$$
\begin{bmatrix}
20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\
-3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\
-1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\xrightarrow{\boxed{f^{-1}}}
\begin{bmatrix}
320 & 55 & -30 & 16 & 72 & -80 & 51 & 0 \\
-36 & -24 & 14 & 38 & 26 & 0 & 0 & 0 \\
-14 & -13 & 16 & 24 & 40 & 0 & 0 & 0 \\
-14 & 0 & 0 & 29 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

JPEG Decoding Stage III: Inverse Transform

$$
\begin{bmatrix}
320 & 55 & -30 & 16 & 72 & -80 & 51 & 0 \\
-36 & -24 & 14 & 38 & 26 & 0 & 0 & 0 \\
-14 & -13 & 16 & 24 & 40 & 0 & 0 & 0 \\
-14 & 0 & 0 & 29 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\longrightarrow \boxed{\begin{matrix} 8\times 8 \\ \text{IDCT} \end{matrix}} \longrightarrow
\begin{bmatrix}
67 & 12 & -9 & 20 & 69 & 43 & -8 & 43 \\
58 & 25 & 15 & 30 & 65 & 40 & -4 & 47 \\
46 & 41 & 44 & 40 & 59 & 38 & 0 & 49 \\
41 & 52 & 59 & 43 & 57 & 42 & 3 & 42 \\
44 & 54 & 58 & 40 & 58 & 47 & 1 & 33 \\
53 & 50 & 53 & 46 & 63 & 41 & 0 & 45 \\
55 & 50 & 56 & 53 & 64 & 34 & -1 & 57 \\
52 & 51 & 53 & 51 & 53 & 42 & 2 & 41
\end{bmatrix}
$$

128 (DC level)

$$
\begin{bmatrix}
195 & 140 & 119 & 148 & 197 & 171 & 120 & 170 \\
186 & 153 & 143 & 158 & 193 & 168 & 124 & 175 \\
174 & 169 & 172 & 168 & 187 & 166 & 128 & 177 \\
169 & 180 & 187 & 171 & 185 & 170 & 131 & 170 \\
172 & 182 & 186 & 168 & 186 & 175 & 129 & 161 \\
181 & 178 & 181 & 174 & 191 & 169 & 128 & 173 \\
183 & 178 & 184 & 181 & 192 & 162 & 127 & 185 \\
180 & 179 & 181 & 179 & 181 & 170 & 130 & 169
\end{bmatrix}
\longleftarrow \; + \; \longleftarrow
$$

Quantization Noise

$$\begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix} \qquad \begin{bmatrix} 195 & 140 & 119 & 148 & 197 & 171 & 120 & 170 \\ 186 & 153 & 143 & 158 & 193 & 168 & 124 & 175 \\ 174 & 169 & 172 & 168 & 187 & 166 & 128 & 177 \\ 169 & 180 & 187 & 171 & 185 & 170 & 131 & 170 \\ 172 & 182 & 186 & 168 & 186 & 175 & 129 & 161 \\ 181 & 178 & 181 & 174 & 191 & 169 & 128 & 173 \\ 183 & 178 & 184 & 181 & 192 & 162 & 127 & 185 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

$\mathbf{X}$ $\qquad\qquad$ $\hat{\mathbf{X}}$

Distortion calculation: $\boxed{\text{MSE}=||\mathbf{X}-\hat{\mathbf{X}}||^2}$

Rate calculation: $\boxed{\text{Rate=length of encoded bit stream/number of pixels (bps)}}$

JPEG Examples

10 (8k bytes)

50 (21k bytes)

90 (58k bytes)

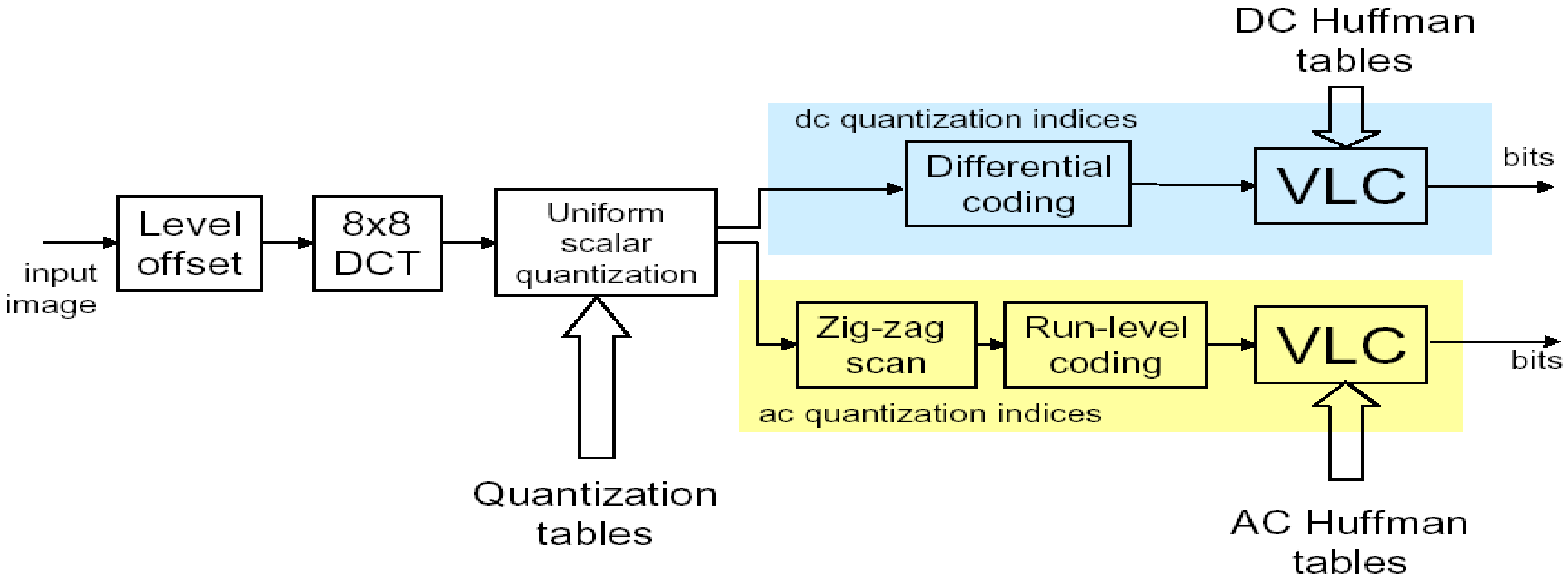
0 worst quality,

highest compression

best quality, 100

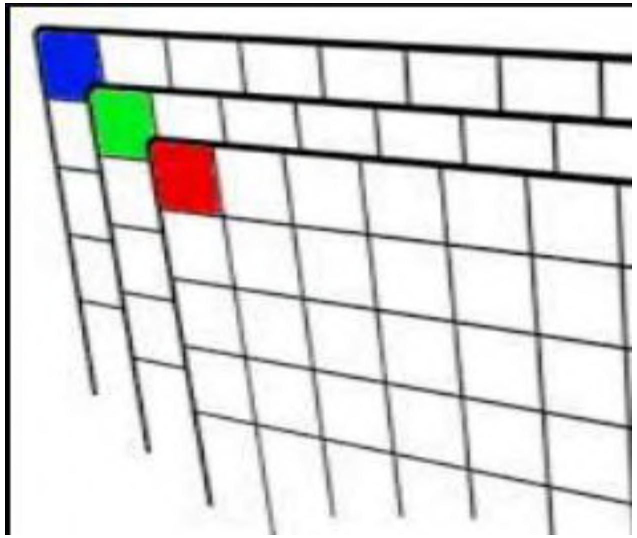
lowest compression

JPEG Coding Algorithm Summary



Color image

A **color image** has three values (or channels) per pixel and they measure the intensity and chrominance of light. The actual information stored in the digital **image** data is the brightness information in each spectral band.



Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you