Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – ΜΥΕ037

Εαρινό εξάμηνο 2023-2024

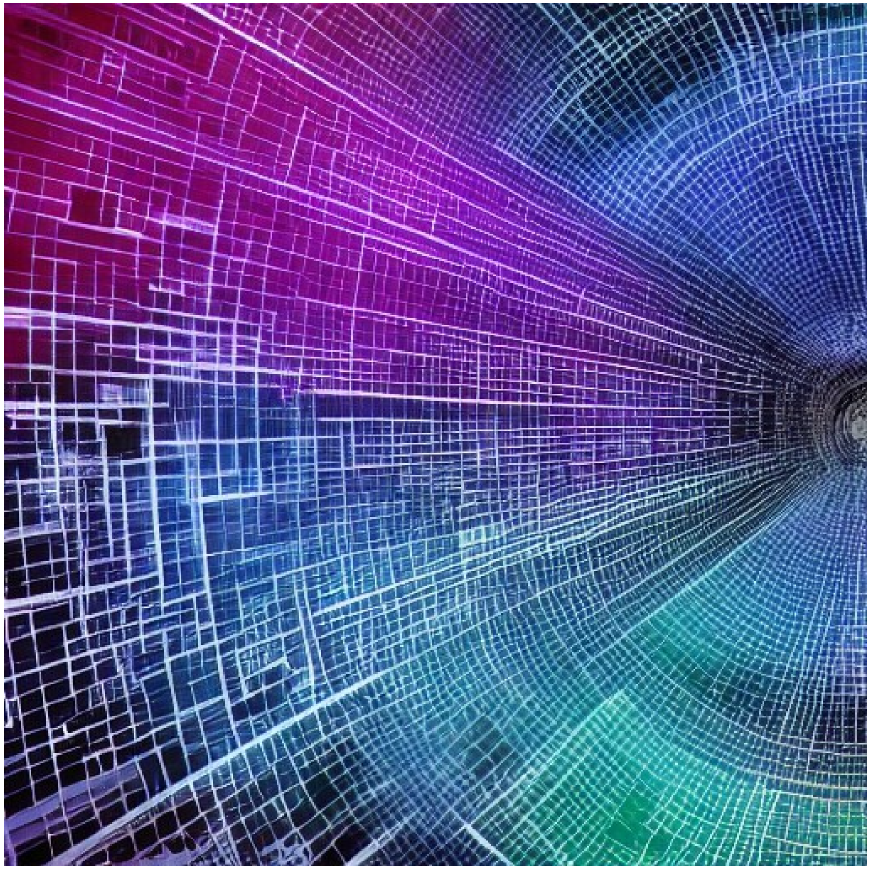
Εισαγωγή στην ΨΕΕ

Άγγελος Γιώτης

a.giotis@uoi.gr

Images taken from:
R. Gonzalez and R. Woods. Digital Image Processing, Prentice Hall, 2008.
Digital Image Processing course by Brian Mac Namee, Dublin Institute of Technology.

Introduction



“One picture is worth more than ten thousand words”

Fred R. Barnard, 1921

Grading

Prerequisites

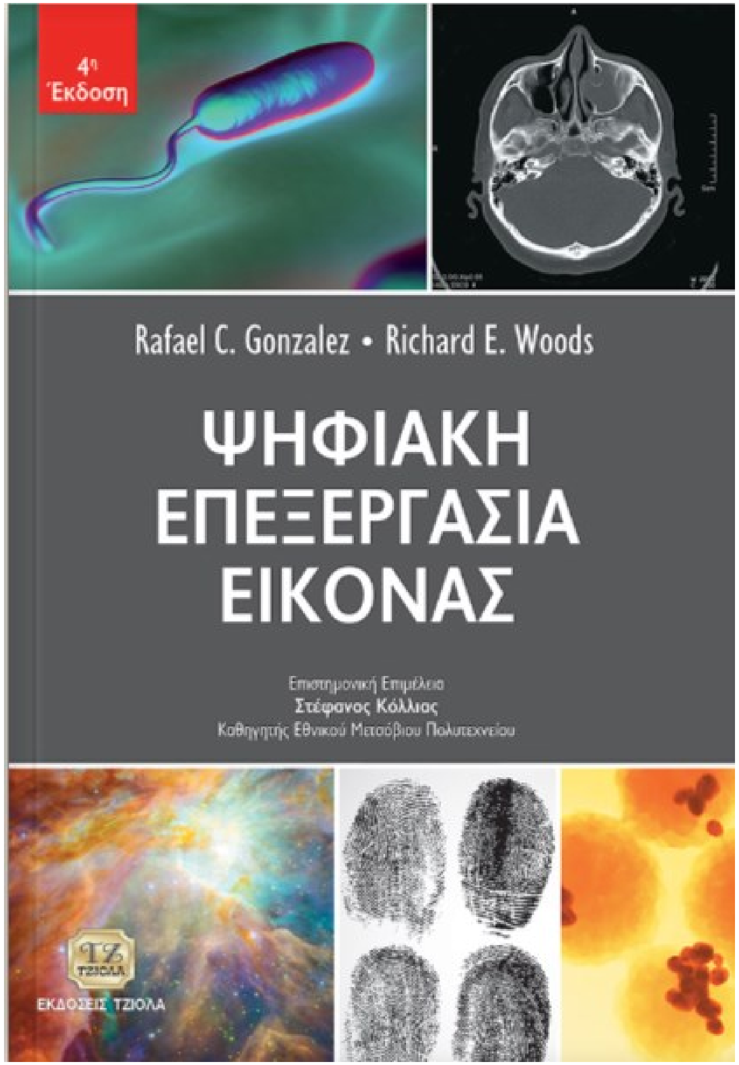
- Linear Algebra
- Signals and systems
- Python Programming skills

Course Grading

- Assignments (30%)
- Final examination (70%)

Bibliography

- R. C. Gonzalez, R. E. Woods, *Ψηφιακή Επεξεργασία Εικόνας*, Εκδόσεις Τζιόλα, 4η Έκδοση, 2018.



Bibliography (cont...)

- Ν. Παπαμάρκος, *Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας*, Εκδόσεις ΑΦΟΙ Παπαμάρκου Ο.Ε., 3η Έκδοση, 2013.

Contents

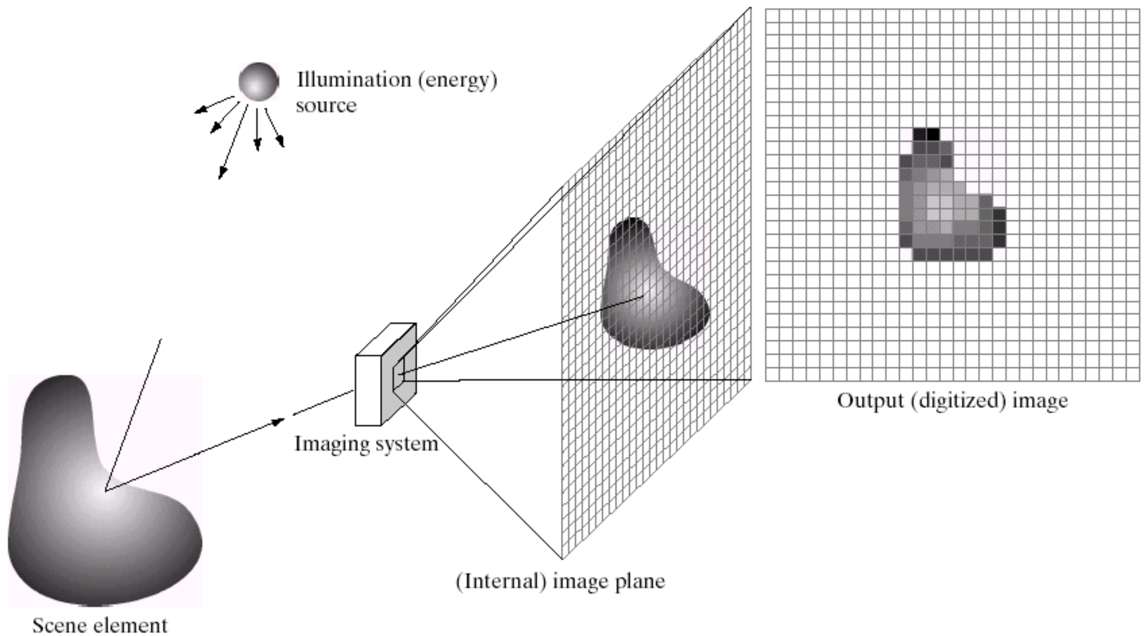
This lecture will cover:

- What is a digital image?
- What is digital image processing?
- History of digital image processing
- State of the art examples of digital image processing
- Key stages in digital image processing

What is a Digital Image?

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

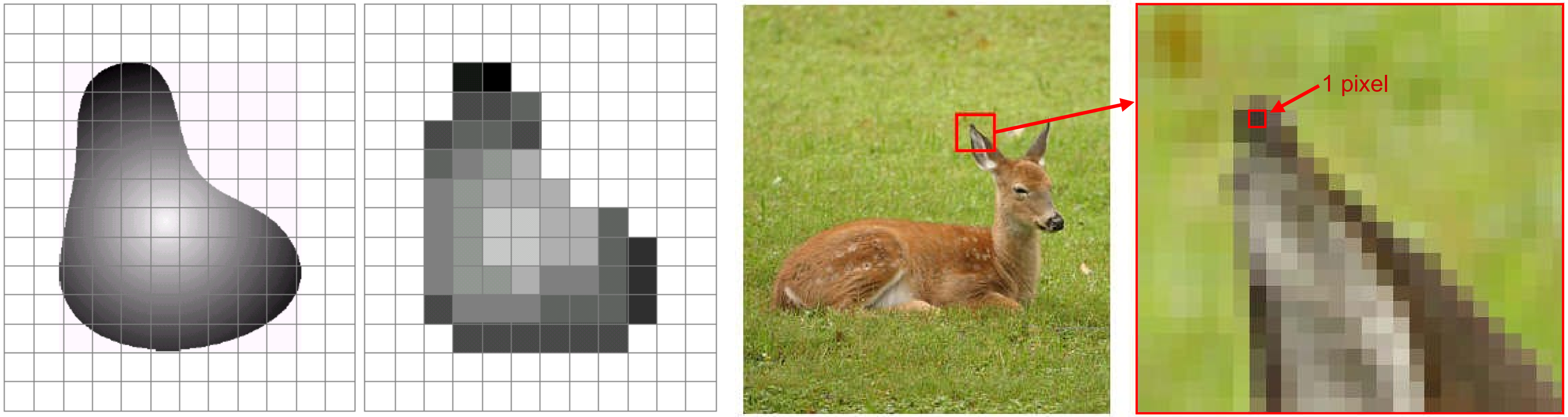
A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels



What is a Digital Image? (cont…)

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

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



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

What is a Digital Image? (cont…)

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

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

What is Digital Image Processing?

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

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

History of Digital Image Processing

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

Early 1920s: One of the first applications of digital imaging was in the news-paper industry

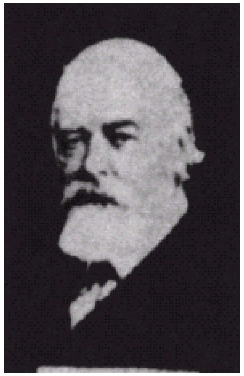
1921

- The Bartlane cable picture transmission service
- An image was transferred by submarine cable between London and New York in 3 hours
- Pictures were coded for cable transfer and reconstructed at the receiving end on a telegraph printer with halftoning

History of DIP (cont…)

Mid to late 1920s: Improvements to the Bartlane system resulted in higher quality images

- New reproduction processes based on photographic techniques
- Increased number of tones in reproduced images


1922


15 tone digital image (1929)

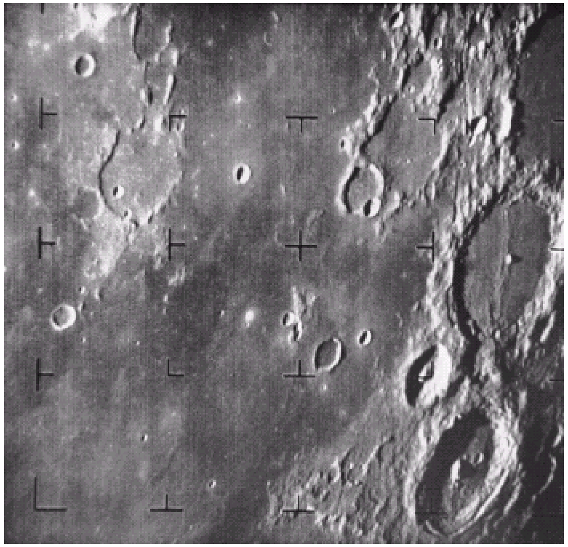
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

History of DIP (cont…)

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

1960s: Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing

- **1964:** Computers used to improve the quality of images of the moon taken by the *Ranger 7* probe
- Such techniques were used in other space missions including the Apollo landings



A picture of the moon taken 17 minutes before landing in 1964

History of DIP (cont…)

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

1970s: Digital image processing begins to be used in medical applications

- **1979:** Sir Godfrey N. Hounsfield & Prof. Allan M. Cormack share the Nobel Prize in medicine for the invention of tomography, the technology behind Computerised Axial Tomography (CAT) scans

Typical head slice CAT image

History of DIP (cont…)

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

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

Imaging modalities

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

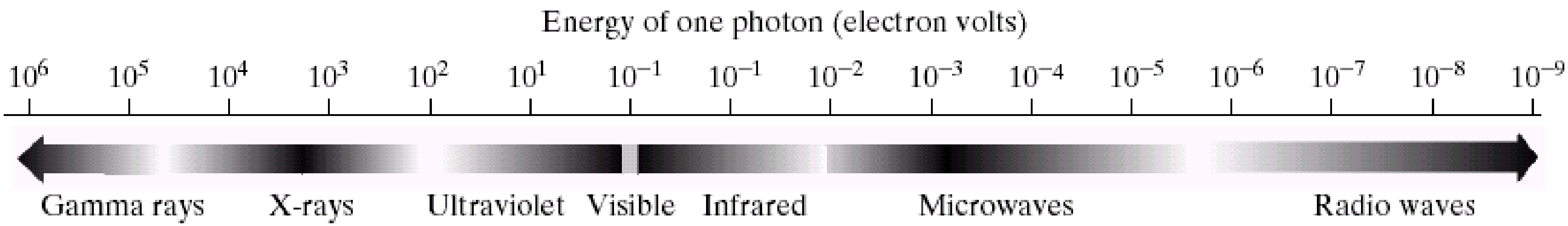
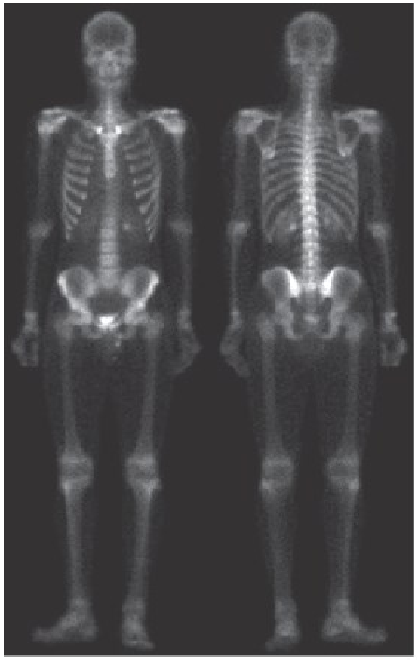


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

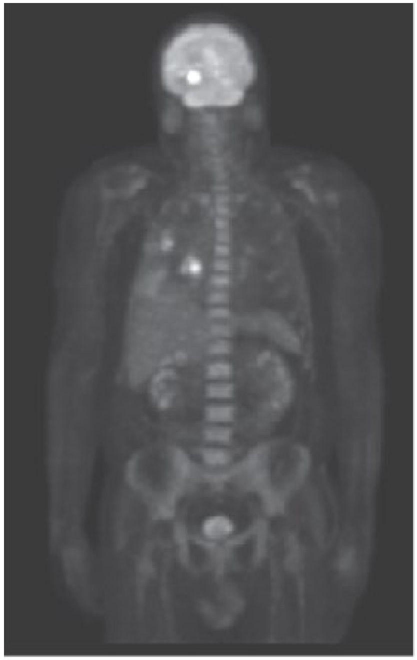
Gamma-ray imaging

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

- Bone scan

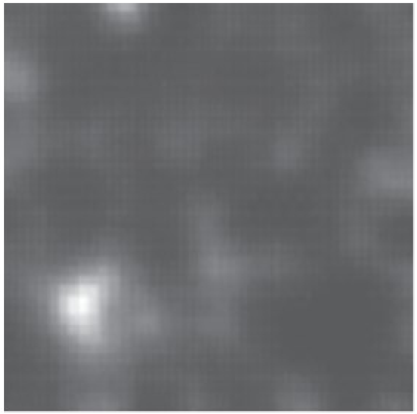


- Positron Emission Tomography (PET)



- Cygnus Loop natural radiation of the star (exploded 15k years ago)

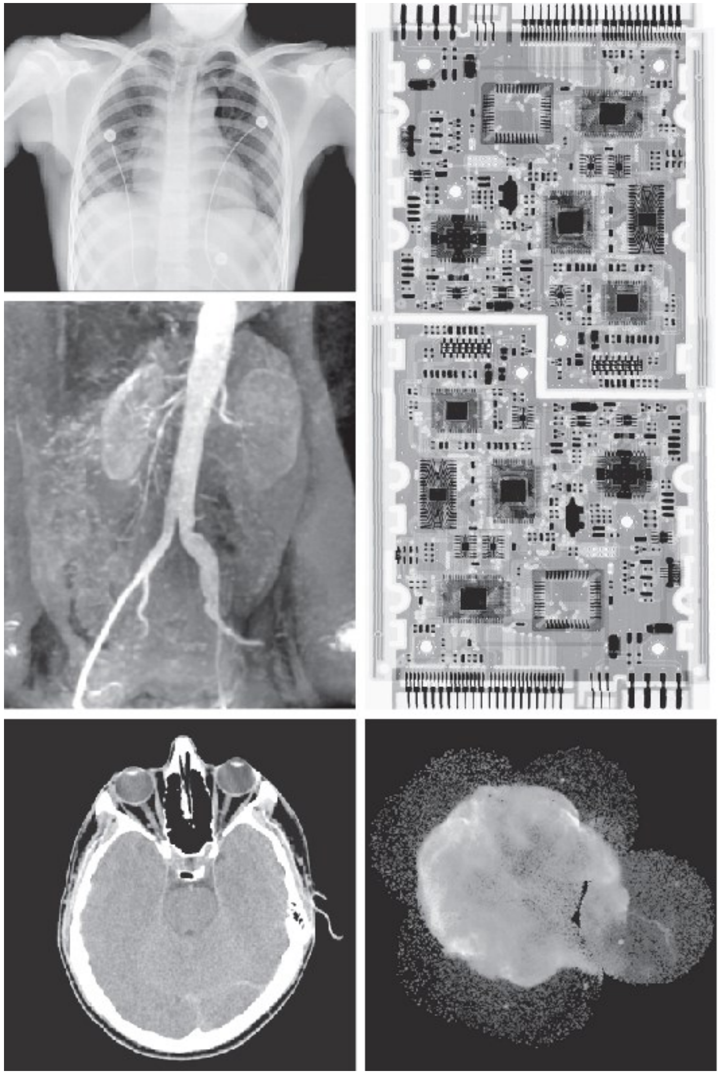
- Valve of a nuclear reactor



X-ray imaging

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

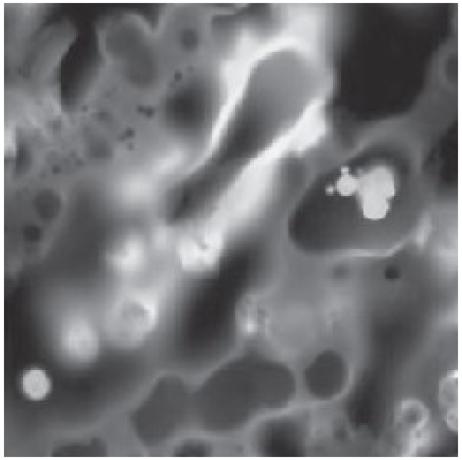
- Chest X-ray
  - Absorption of energy
- Angiography
  - Catheter with contrast medium
- Computed Axial Tomography (CAT)
- Manufacturing errors in electronic circuits
- Cygnus Loop

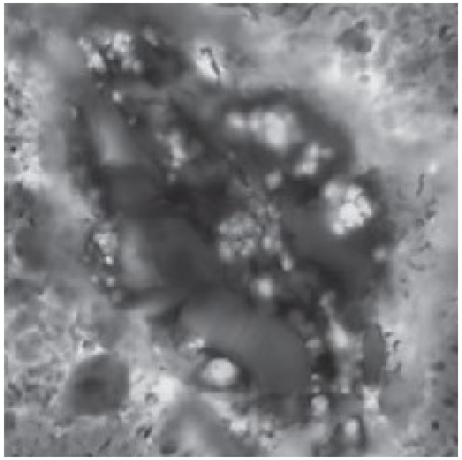


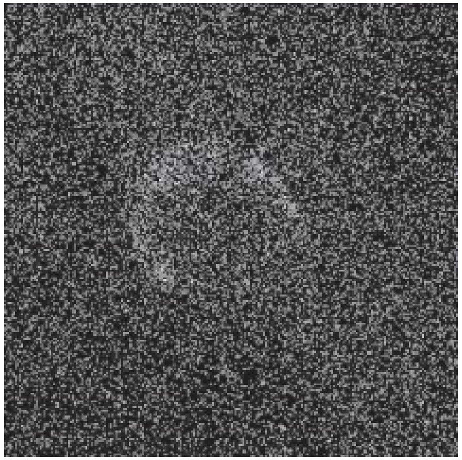
Ultraviolet imaging

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

- Fluorescence microscopy
  - Normal corn
  - Corn infected by smut disease
- Cygnus Loop



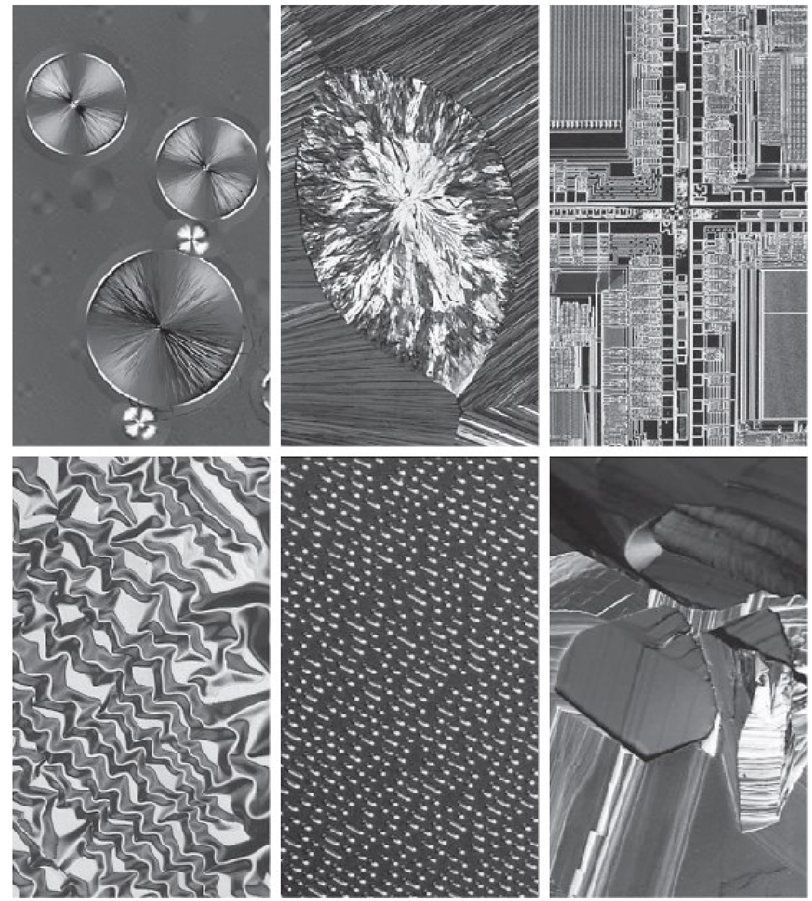




Visible and infrared imaging

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

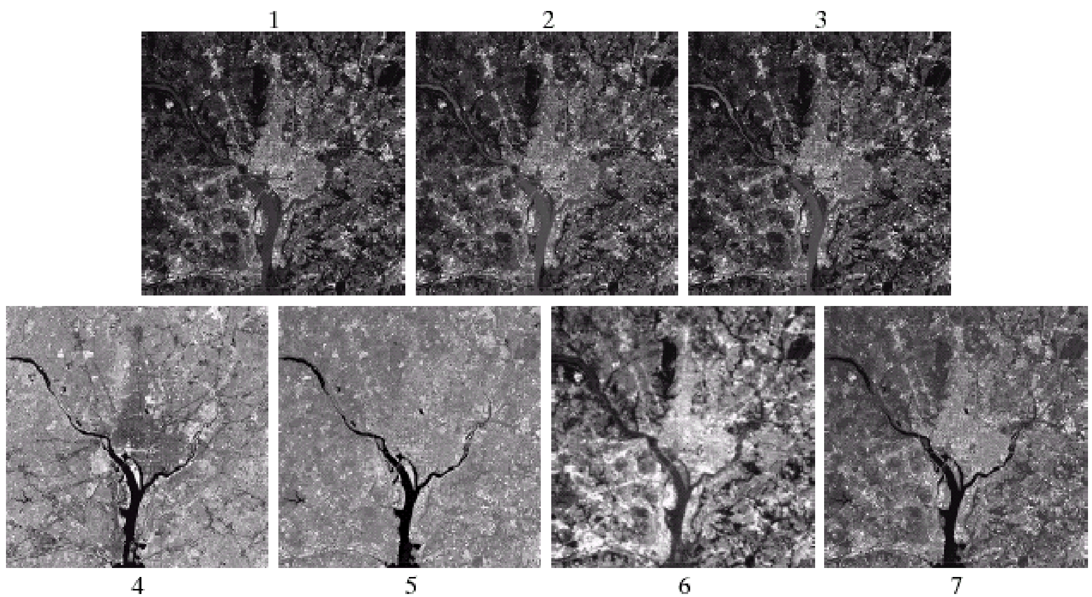
- Light microscopy
  - Taxol (anticancer agent) 250x
  - Cholesterol 40x
  - Microprocessor 60x
  - Nickel oxide thin film 600x
  - Surface of audio CD 1750x
  - Organic superconductor 450x



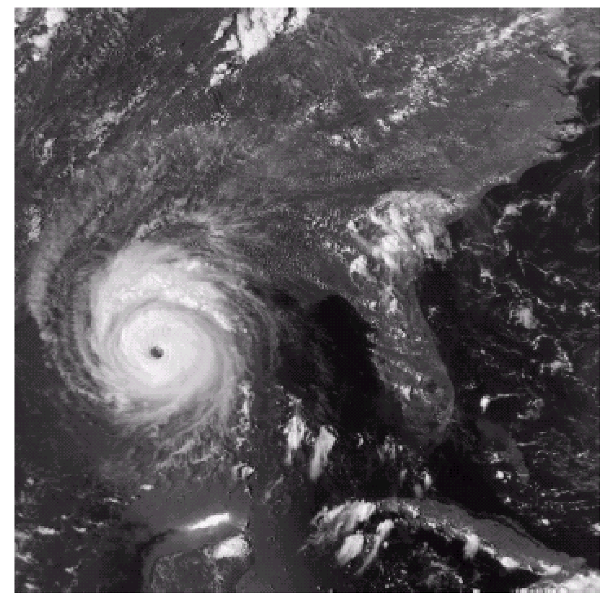
Visible and infrared imaging (cont.)

Remote sensing

- Terrain classification (LANDSAT)
- Meteorology (NOAA)



LANDSAT thematic bands of Washington DC area



Hurricane Katrina, 2005

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

Visible and infrared imaging (cont.)

- Night-time lights of the world
  - Infrared band
  - Useful for estimating the percent of total electrical energy

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

Visible and infrared imaging (cont.)

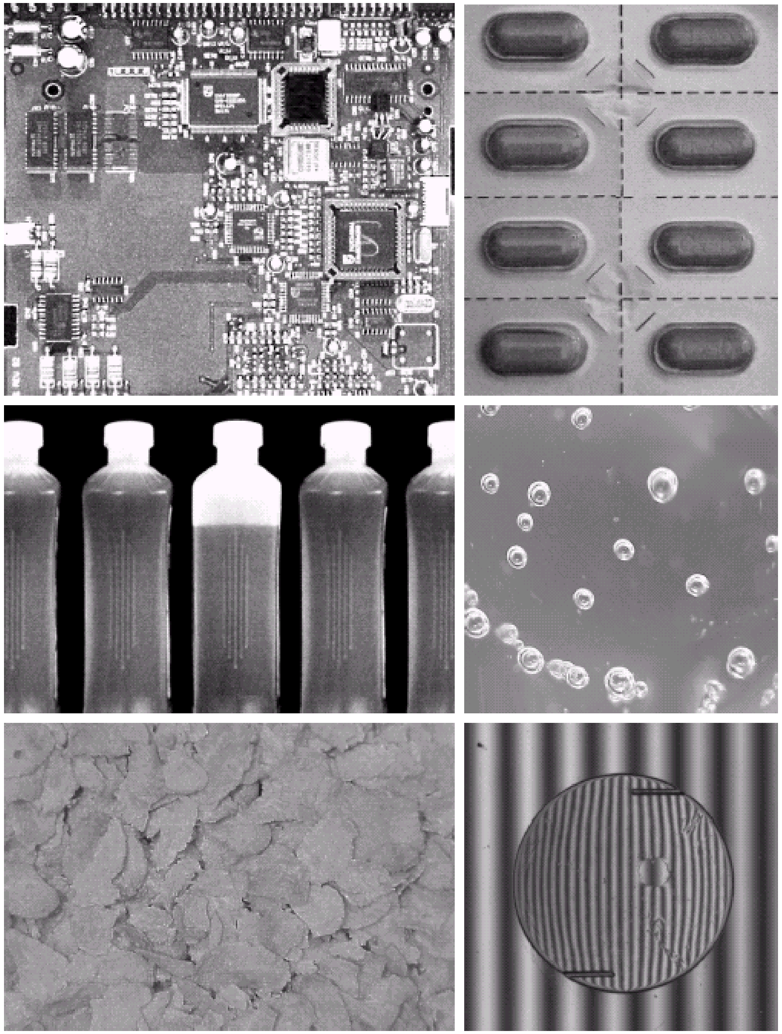
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

- Night-time lights of the world
  - Infrared band
  - Useful for estimating the percent of total electrical energy

Visible and infrared imaging (cont.)

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

- Industrial inspection
  - Circuit board controller
  - Pill container
  - Bottle filling
  - Air pockets in plastic parts
  - Burned flakes
  - Intraocular implant
    - Structured light for detecting lens deformations (damages at 1 and 5 o’clock)



Visible and infrared imaging (cont.)

- Law enforcement
  - Fingerprint for database search
  - Automated counting
  - Bill identification
  - Licence plate detection and reading

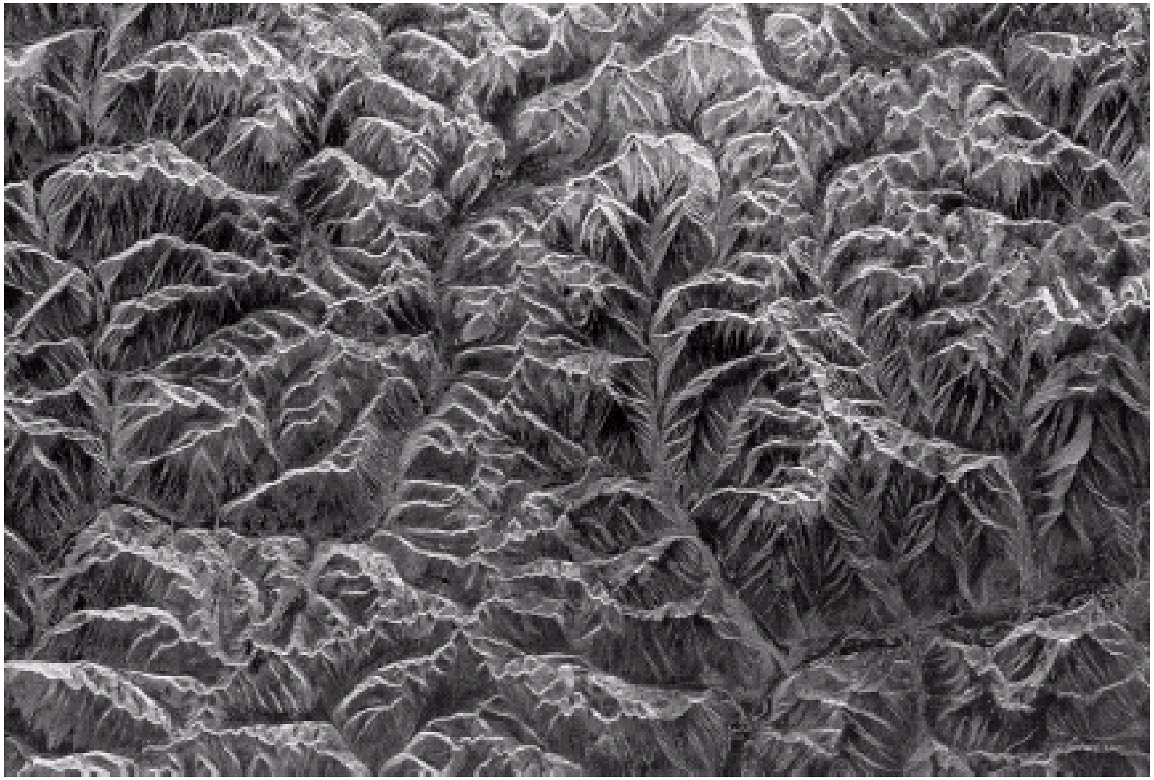
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Imaging in the microwave

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

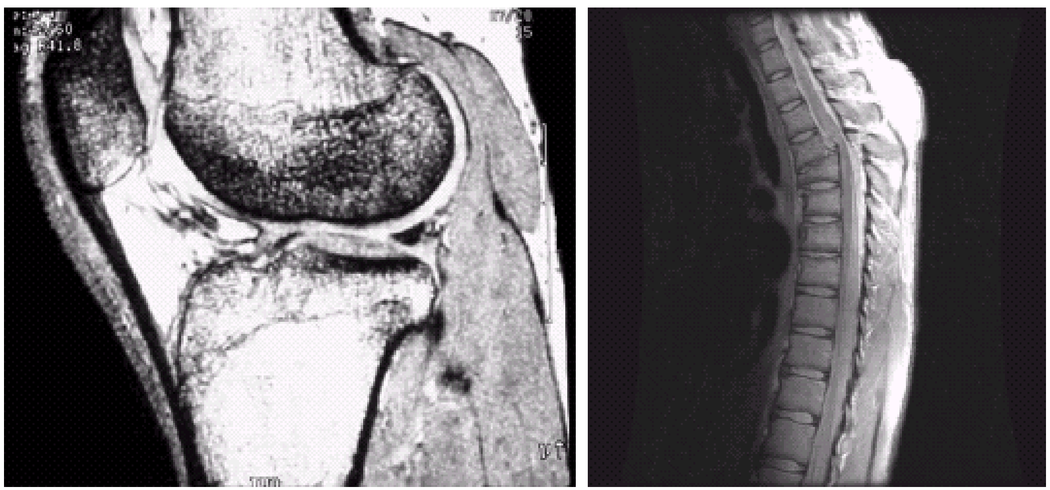
- Radar is the dominant application
  - It emits pulses and receives them back at its antenna

FIGURE 1.16 Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)



Imaging in the radio band

- Magnetic Resonance Imaging (MRI)
  - Patient placed in a magnet and radio wave pulses are emitted through the body
  - Resonance takes place with tissues (e.g. water molecules)



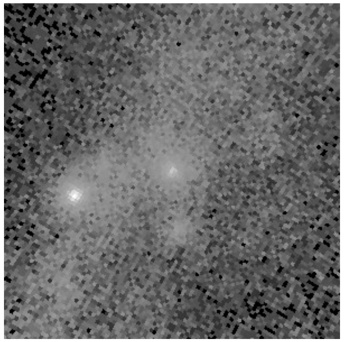
a b

FIGURE 1.17 MRI images of a human (a) knee, and (b) spine. (Image (a) courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, and (b) Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

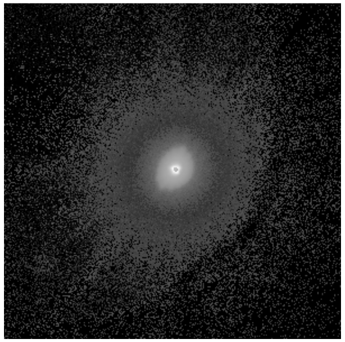
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Imaging in the radio band

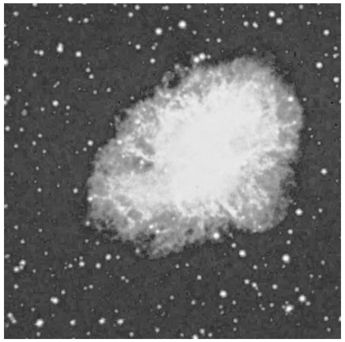
- Astronomy



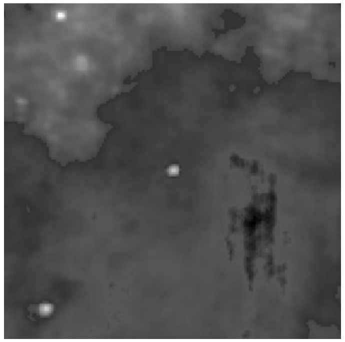
Gamma



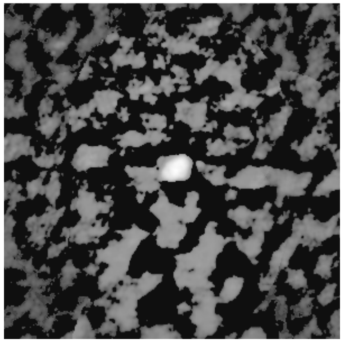
X-ray



Optical



Infrared



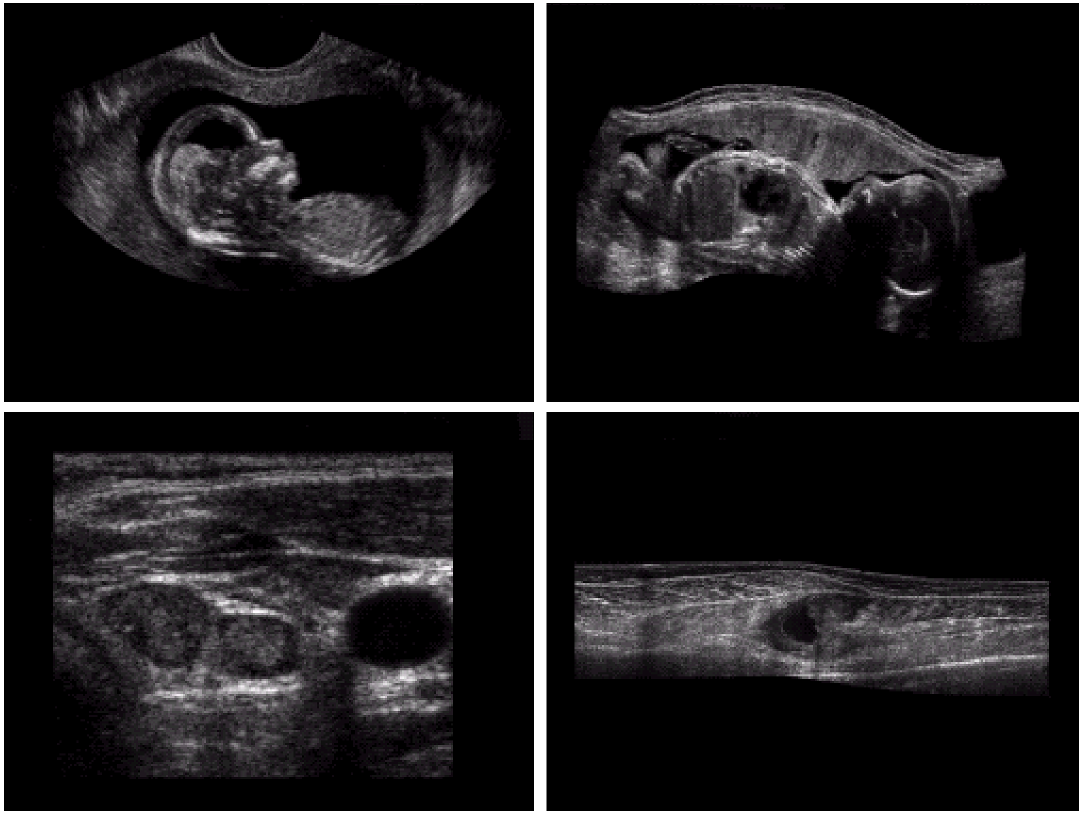
Radio

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

Other imaging modalities

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

- Ultrasound imaging



a b
c d

FIGURE 1.20 Examples of ultrasound imaging. (a) Baby. (2) Another view of baby. (c) Thyroids. (d) Muscle layers showing lesion. (Courtesy of Siemens Medical Systems, Inc., Ultrasound Group.)

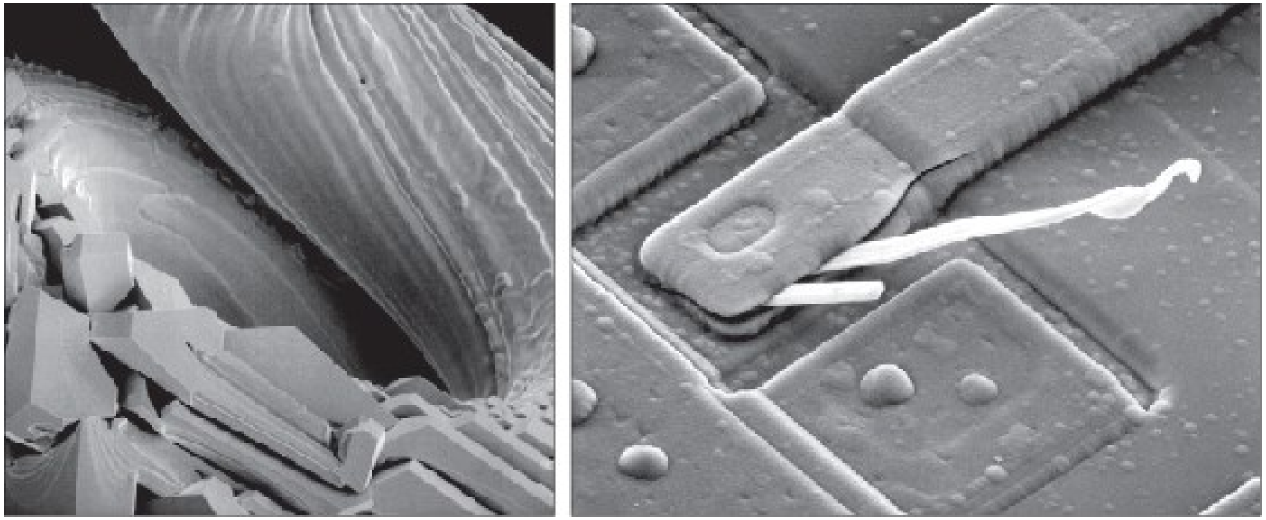
Other imaging modalities (cont.)

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

- Electron Microscopy (EM)
  - Works like a slide projector emitting a beam of electrons instead of light
  - The transmitted beam is projected on a phosphor screen
  - The interaction of the beam with the slide produces light which is recorded
    - Scanning Electron Microscopy (SEM)
    - Transmission Electron Microscopy (TEM)
- Very high magnification (10000x)

Other imaging modalities (cont.)

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

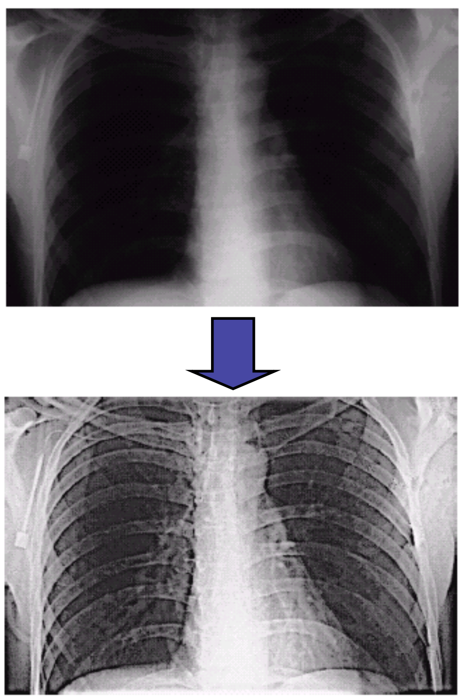


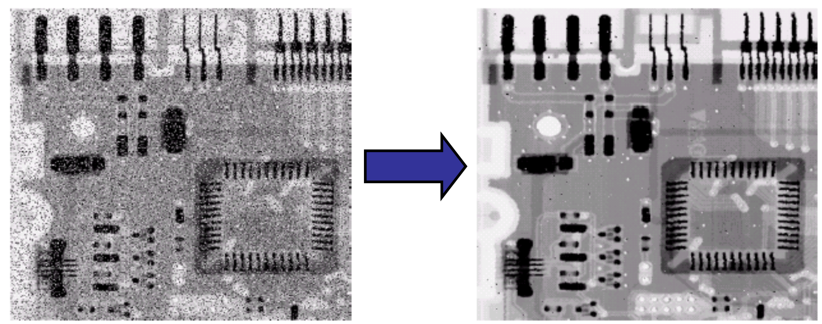
a b

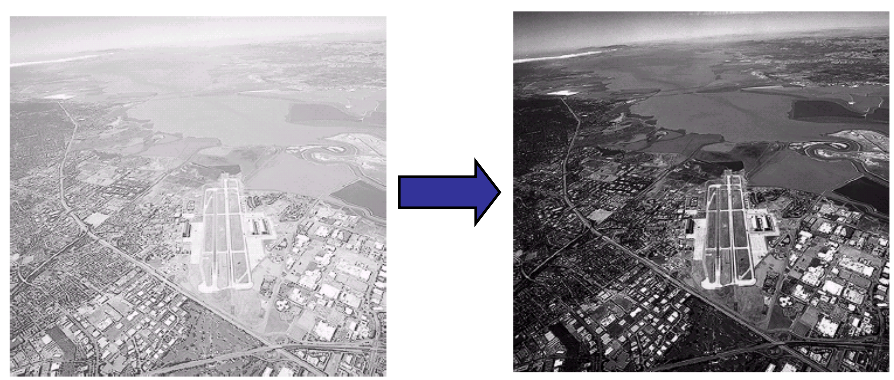
FIGURE 1.21 (a) 250 × SEM image of a tungsten filament following thermal failure (note the shattered pieces on the lower left). (b) 2500 × SEM image of a damaged integrated circuit. The white fibers are oxides resulting from thermal destruction. (Figure (a) courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene; (b) courtesy of Dr. J. M. Hudak, McMaster University, Hamilton, Ontario, Canada.)

Applications: Image Enhancement

One of the most common uses of DIP techniques: improve quality, remove noise etc



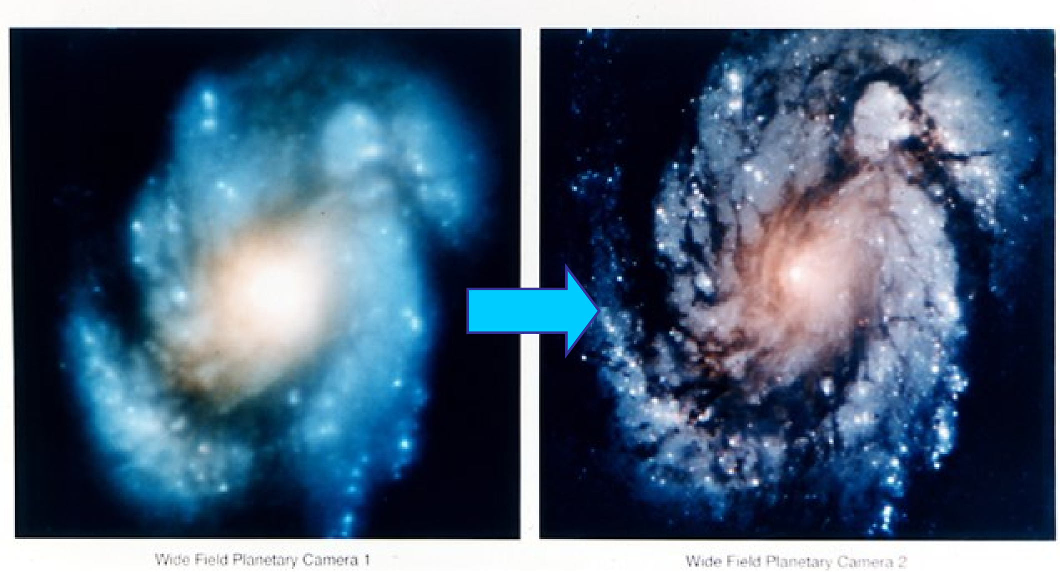




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Applications: The Hubble Telescope

- Launched in 1990 the Hubble telescope can take images of very distant objects
- However, an incorrect mirror made many of Hubble's images useless
- Image processing techniques were used to fix this

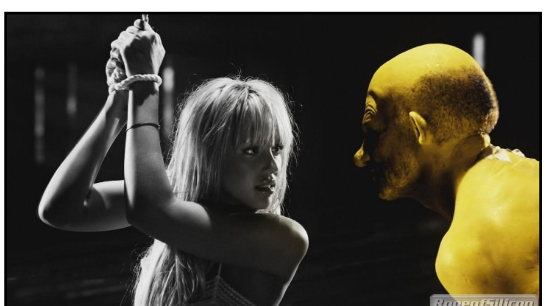


Applications: Newspaper Article Tracking

- Same colored image regions belong to the same semantic category (title)
- Same colored background indicates regions belonging to the same article

Applications: Artistic Effects

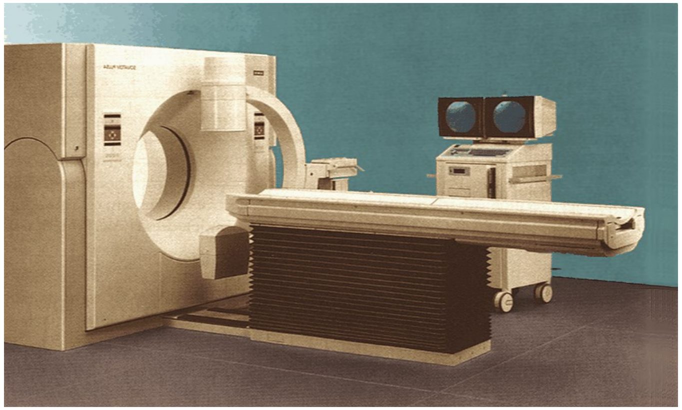
Artistic effects are used to make images more visually appealing, to add special effects and to make composite images

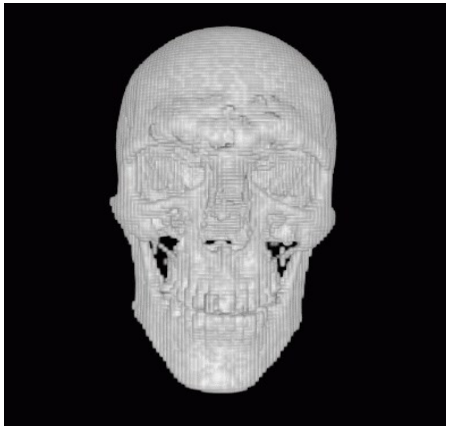


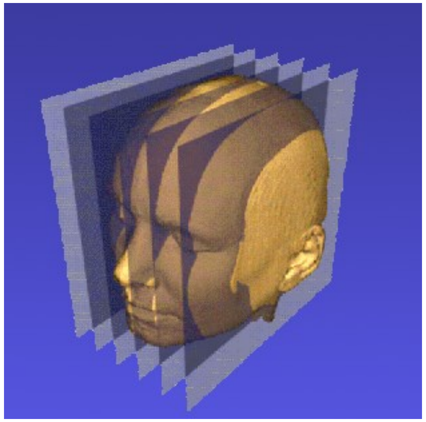
Applications: Medicine

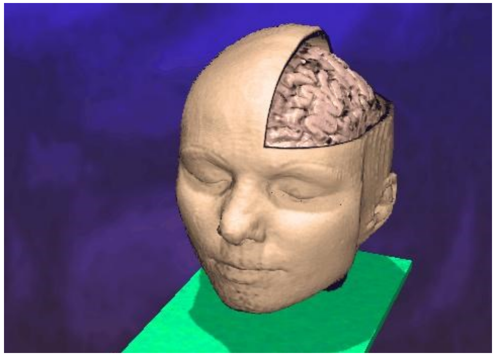
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

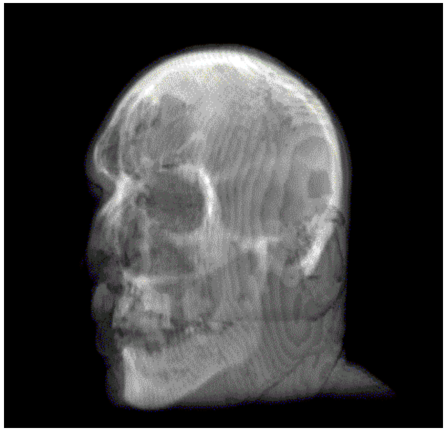
3D tomography and rendering with transparencies





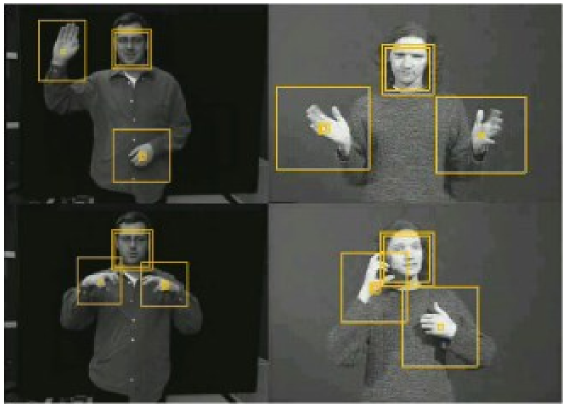


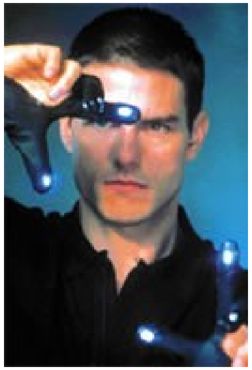




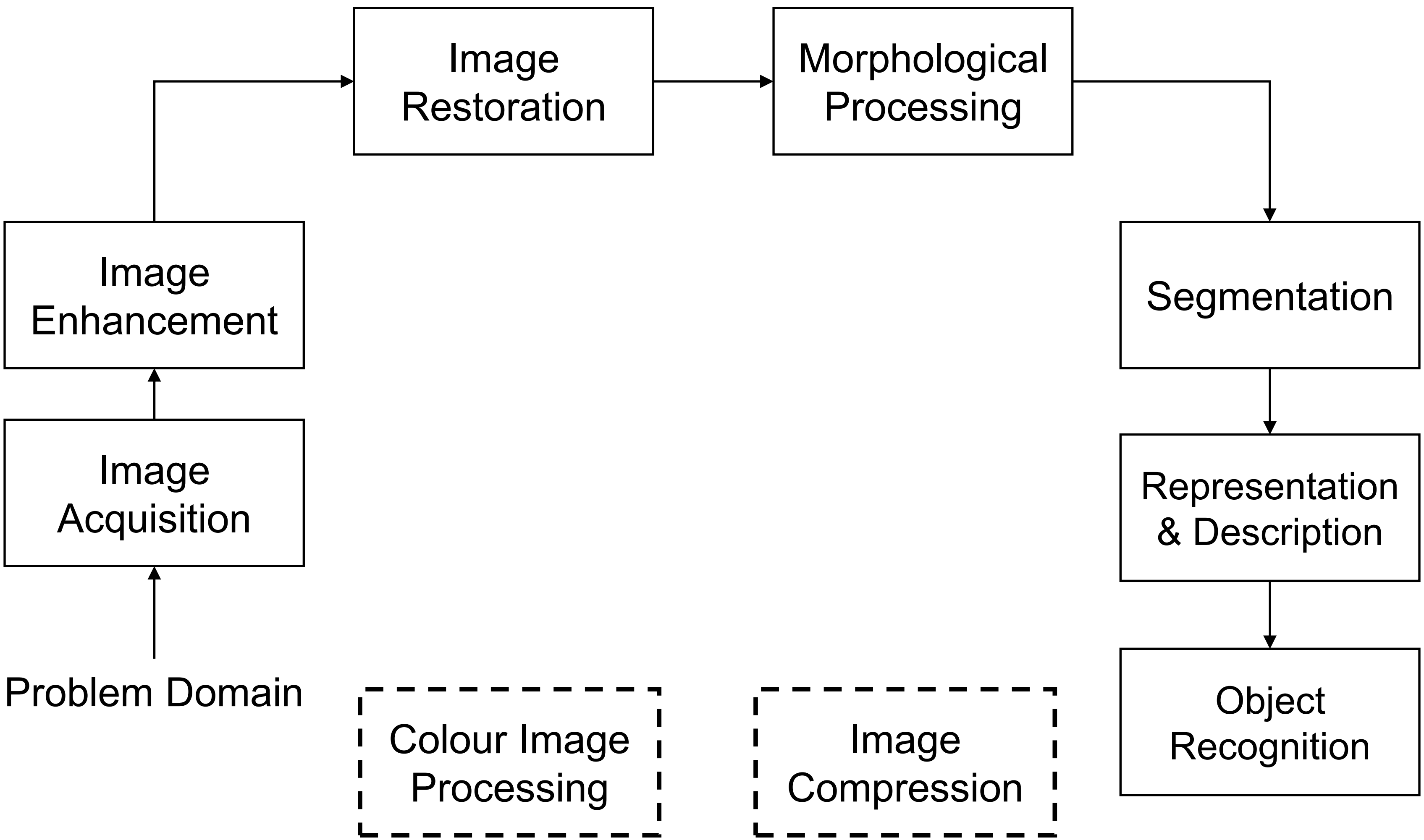
Applications: HCI

- Try to make human computer interfaces more natural
  - – Face recognition
  - – Gesture recognition
- Does anyone remember the user interface from "Minority Report" (2002)?
- These tasks were really difficult at that time





Key Stages in Digital Image Processing



Key Stages in Digital Image Processing: Image Aquisition

Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition

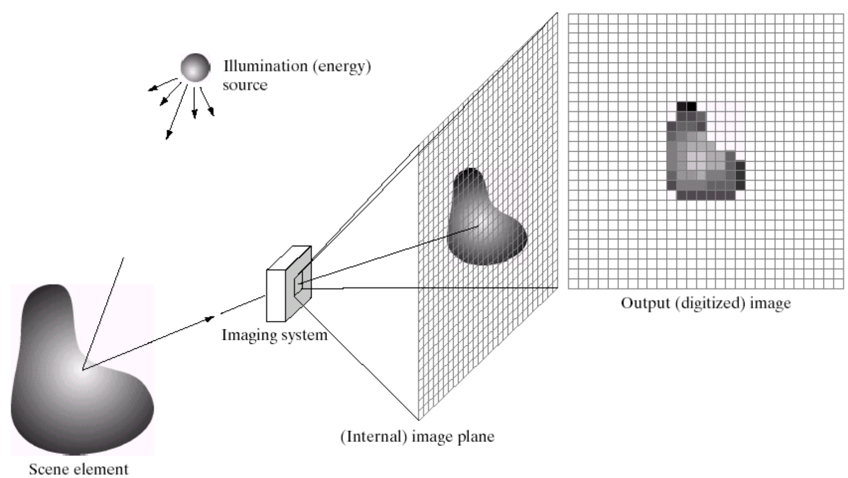
Representation & Description

Problem Domain

Object Recognition

Colour Image Processing

Image Compression



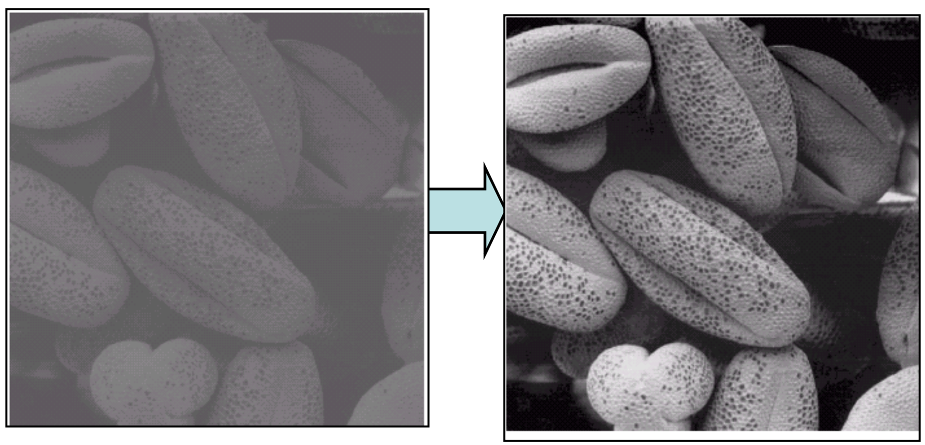
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Key Stages in Digital Image Processing: Image Enhancement

Image Restoration

Morphological Processing

Image Enhancement



Segmentation

Representation & Description

Image Acquisition

Object Recognition

Problem Domain

Colour Image Processing

Image Compression

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)

Problem Domain

Image Acquisition

Image Enhancement

Image Restoration

Morphological Processing

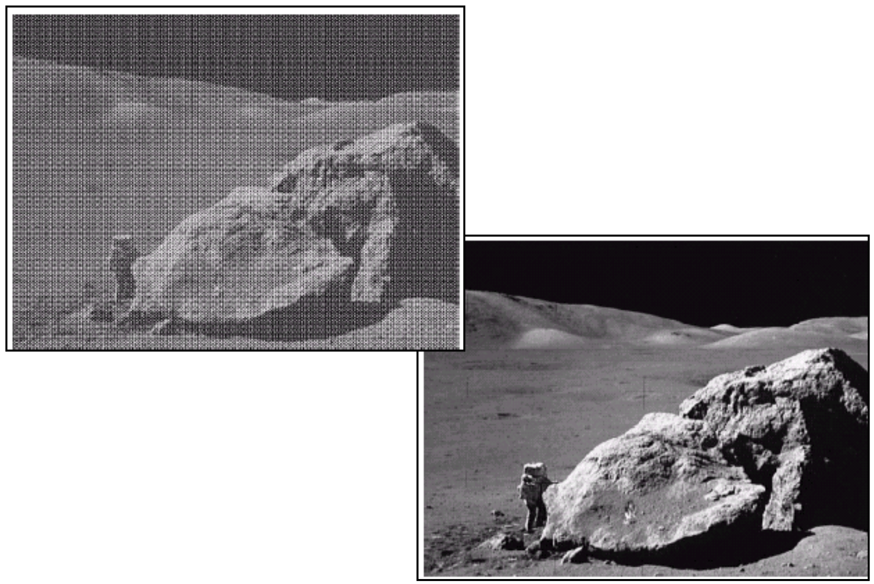
Segmentation

Representation & Description

Object Recognition

Colour Image Processing

Image Compression



Key Stages in Digital Image Processing: Morphological Processing

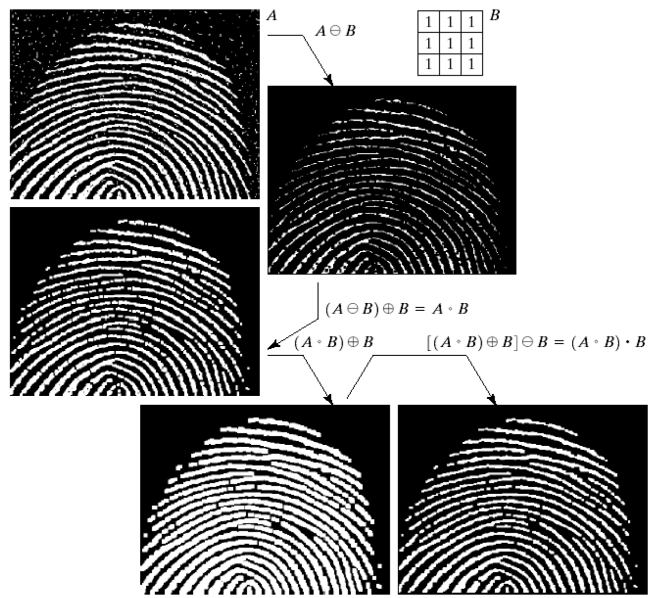
Image Restoration

Morphological Processing

Image Enhancement

Image Acquisition

Problem Domain



Segmentation

Representation & Description

Object Recognition

Colour Image Processing

Image Compression

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

Key Stages in Digital Image Processing: Segmentation

Image Acquisition

Image Enhancement

Image Restoration

Morphological Processing

Segmentation

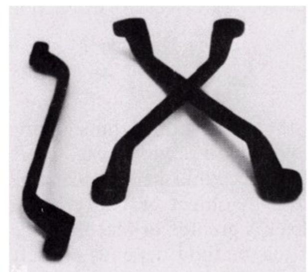
Representation & Description

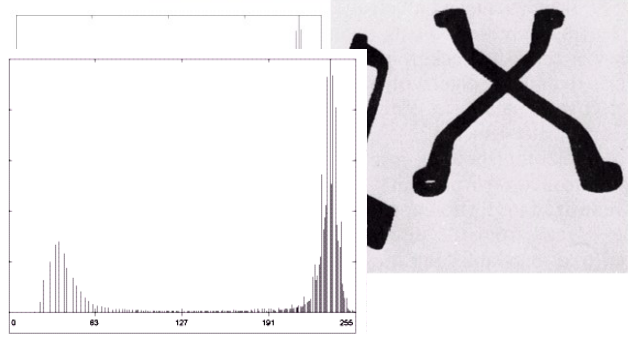
Object Recognition

Problem Domain

Colour Image Processing

Image Compression





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

Key Stages in Digital Image Processing: Representation & Description

Image Restoration

Morphological Processing

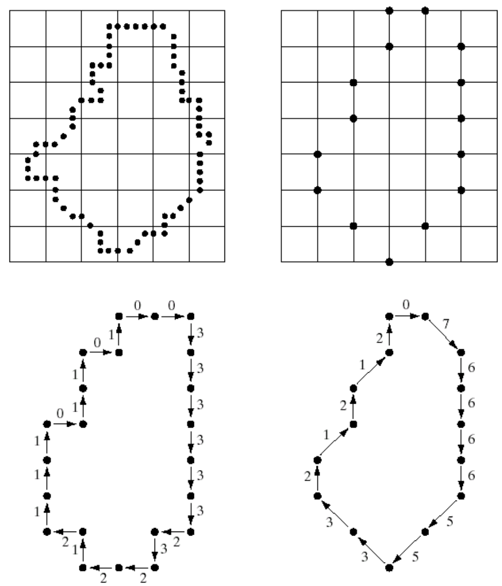
Image Enhancement

Segmentation

Image Acquisition

Representation & Description

Problem Domain



Colour Image Processing

Image Compression

Object Recognition

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

Key Stages in Digital Image Processing: Object Recognition

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

Image Restoration

Morphological Processing

Image Enhancement

Segmentation

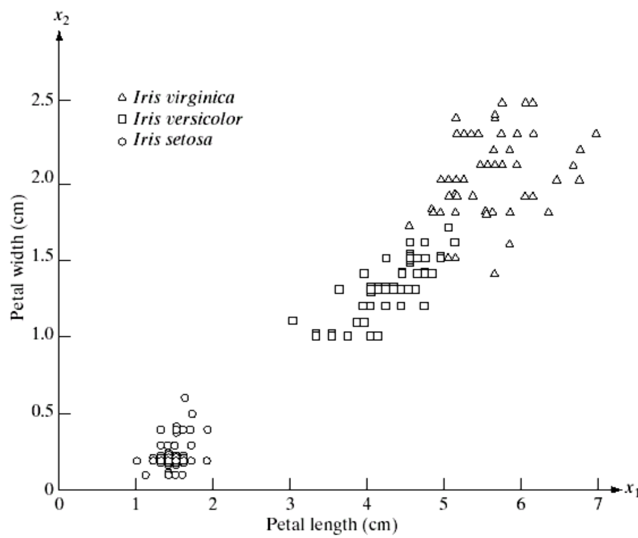


Image Acquisition

Representation & Description

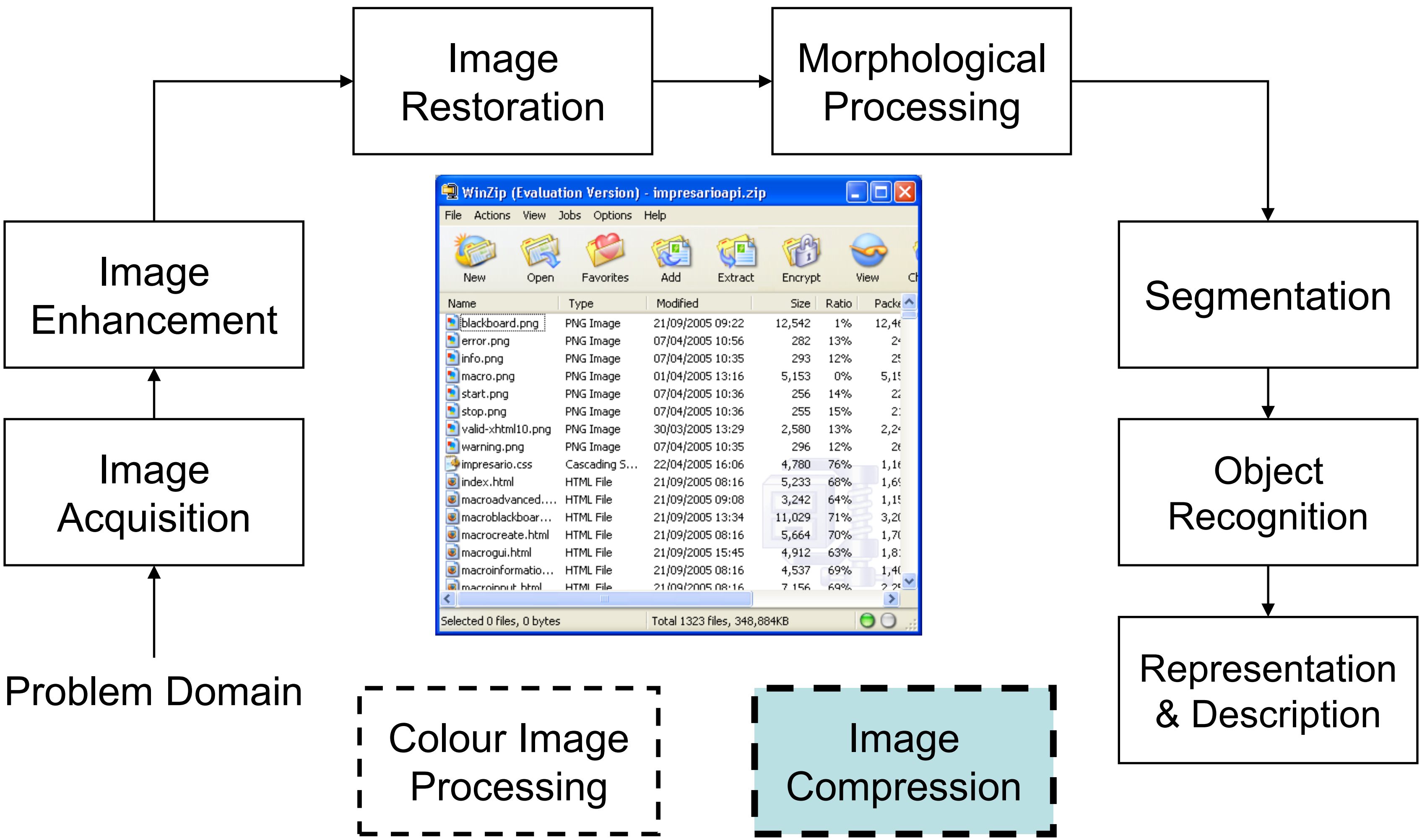
Problem Domain

Colour Image Processing

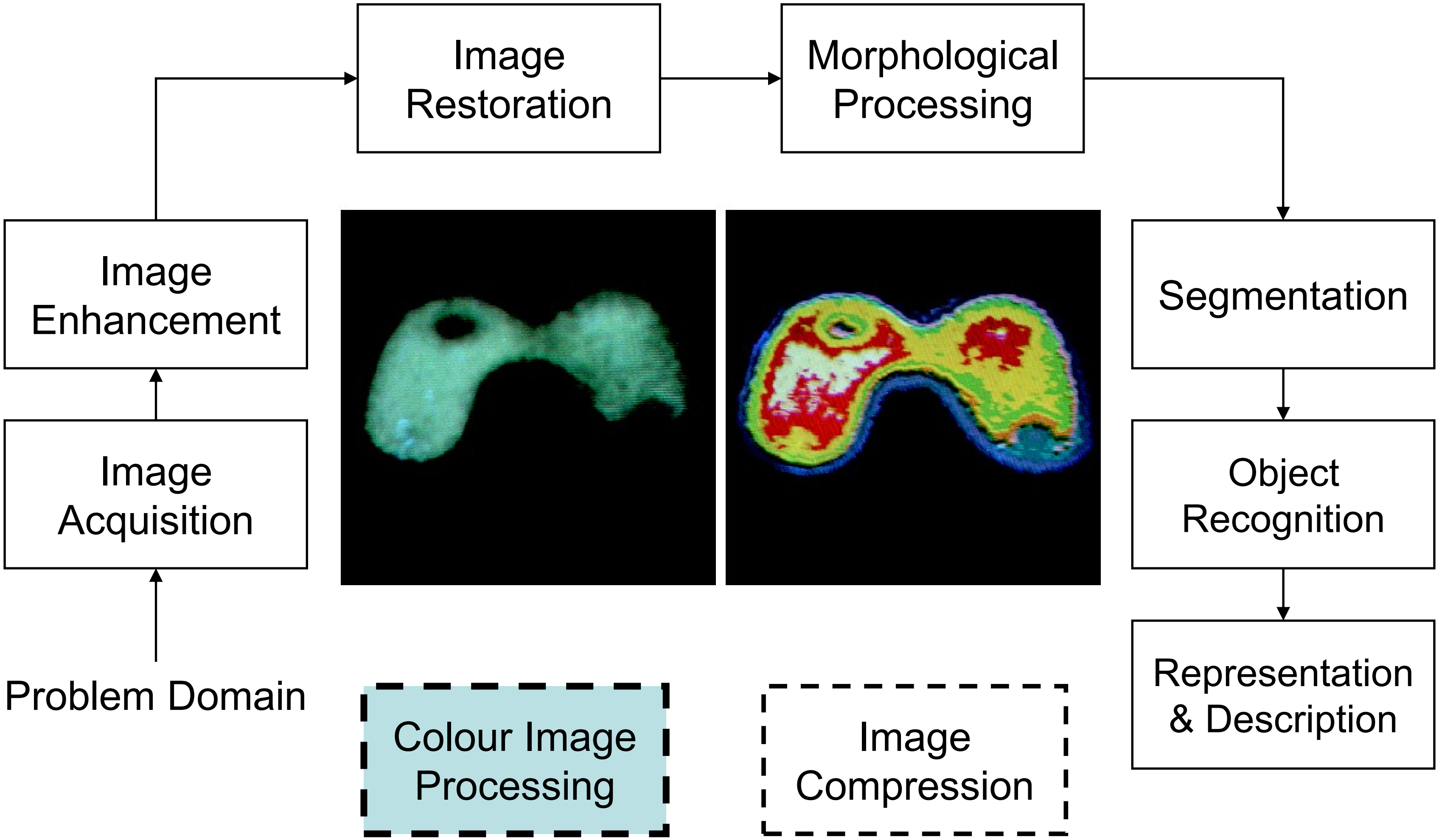
Image Compression

Object Recognition

Key Stages in Digital Image Processing: Image Compression



Key Stages in Digital Image Processing: Colour Image Processing



Summary

We have looked at:

- What is a digital image?
- What is digital image processing?
- History of digital image processing
- State of the art examples of digital image processing
- Key stages in digital image processing

Important: Acquire some experience with Python.

Until next Wednesday

Readings

- Book: Ψηφιακή επεξεργασία Εικόνας, Gonzalez - Woods
    - ***Chapter 1***

Practice

- Fundamentals in image processing (basic steps): https://github.com/zengsn/image-processing-python/blob/master/lecture1216.ipynb

- Python NumPy Tutorial (with Jupyter and Colab)
- https://cs231n.github.io/python-numpy-tutorial/

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Digital Image Fundamentals – Οι Θεμελιώδεις αρχές των ψηφιακών εικόνων

Άγγελος Γιώτης

a.giotis@uoi.gr

Images taken from:
R. Gonzalez and R. Woods. Digital Image Processing, Prentice Hall, 2008.
Digital Image Processing course by Brian Mac Namee, Dublin Institute of Technology.

Digital Image Fundamentals

“Those who wish to succeed must ask the right preliminary questions”

Aristotle

Contents

This lecture will cover:

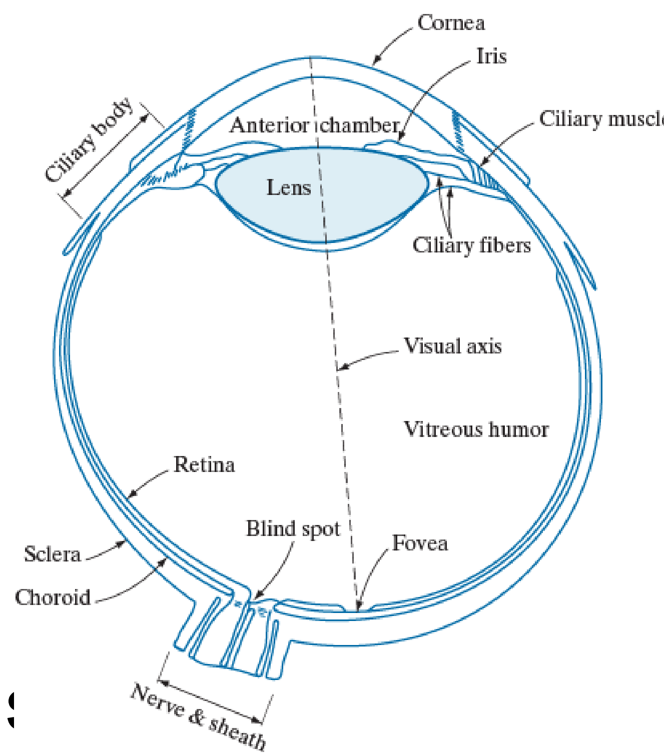
- The human visual system
- Light and the electromagnetic spectrum
- Image representation
- Image sensing and acquisition
- Sampling, quantisation and resolution

Human Visual System

- The best vision model we have!
- Knowledge of how images form in the eye can help us with processing digital images
- We will take just a whirlwind tour of the human visual system

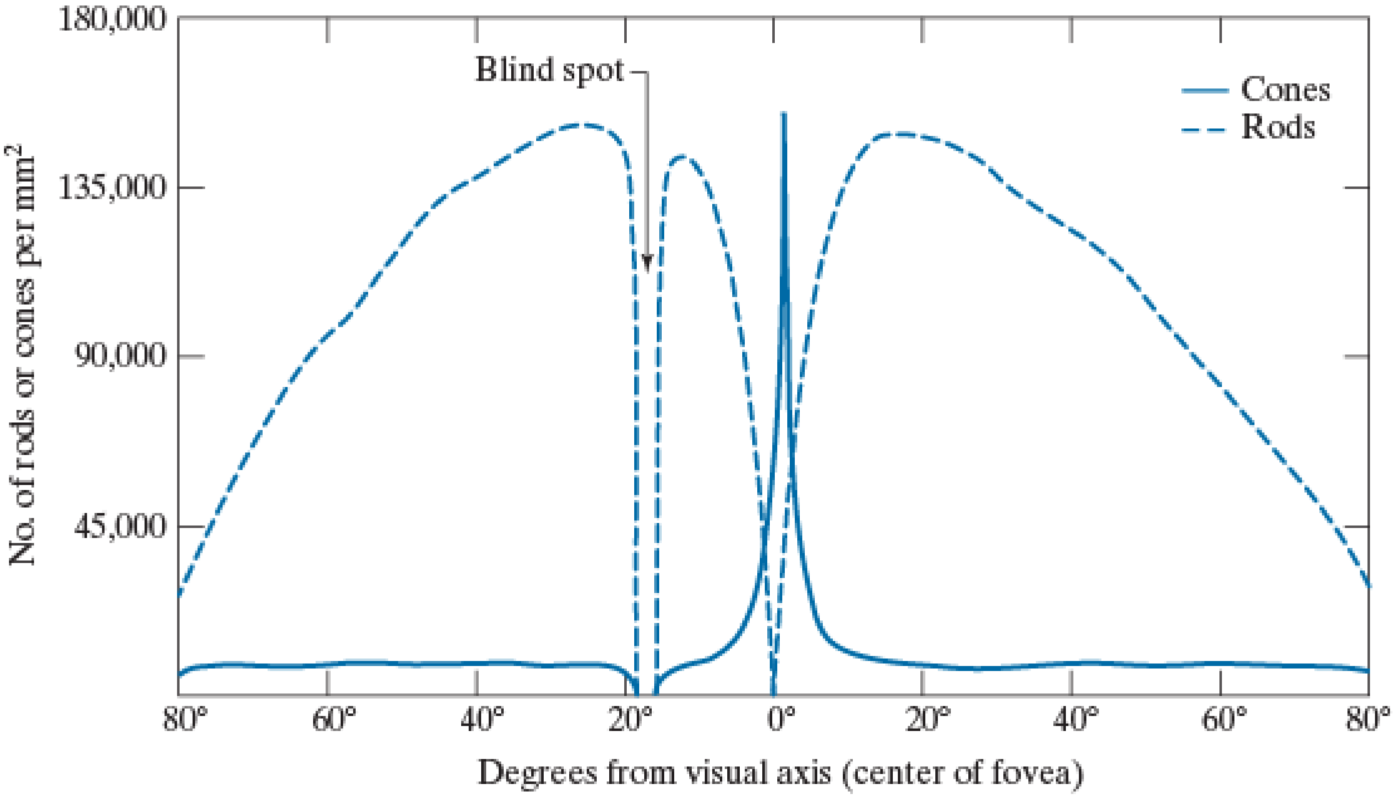
Structure Of The Human Eye

- The lens focuses light from objects onto the retina
- The retina is covered with light receptors called cones (6-7 million) and rods (75-150 million)
- Cones are concentrated around the fovea and are very sensitive to colour
- Rods are more spread out and are sensitive to low levels of illumination



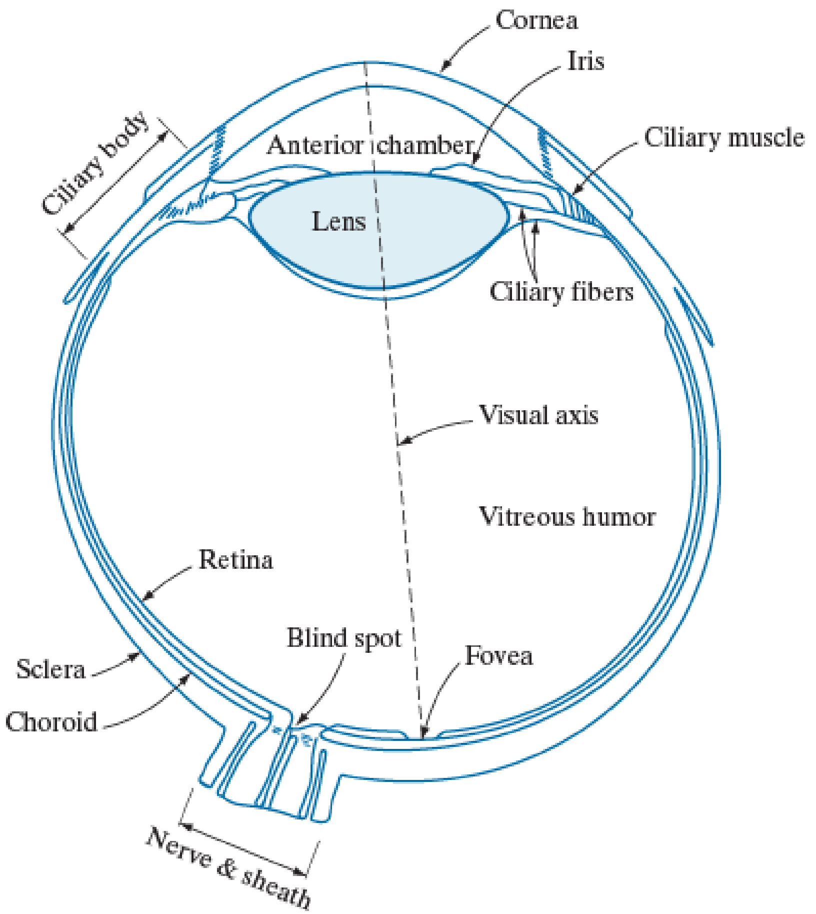
Structure Of The Human Eye (cont.)

Density of cones and rods across a section of the right eye



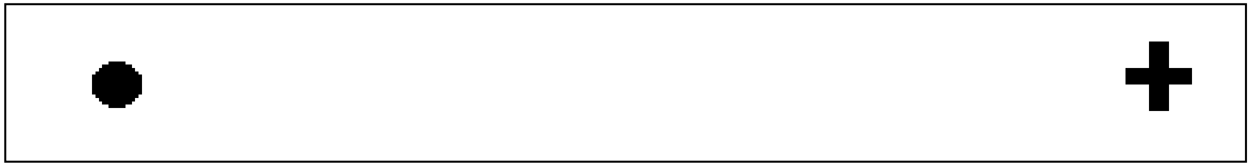
Structure Of The Human Eye (cont.)

- Each cone is connected to each own nerve end.
  - They can resolve fine details.
  - Sensitive to color (*photopic* vision)
- Many rods are connected to a single nerve end
  - Limited resolution with respect to cones
  - Not sensitive to color
  - Sensitive to low level illumination (*scotopic* vision)



Blind-Spot Experiment

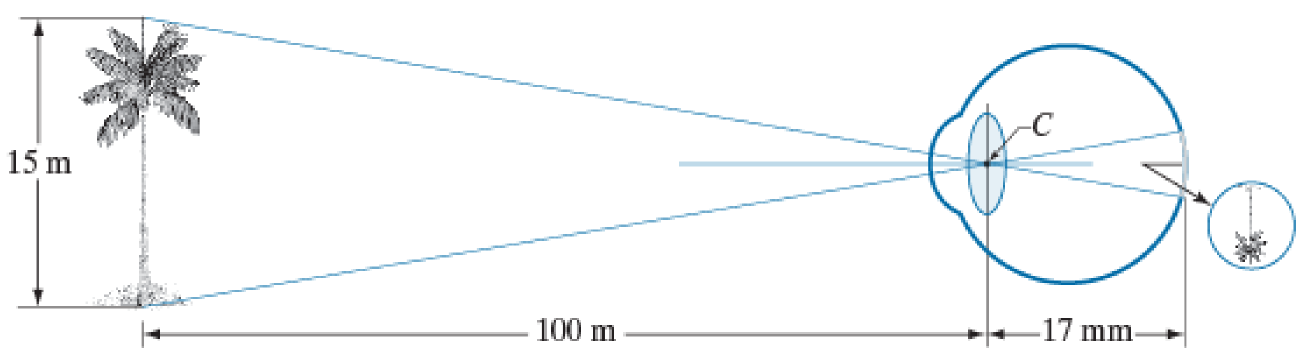
- Draw an image similar to that below on a piece of paper (the dot and cross are about 6 inches apart)



- Close your right eye and focus on the cross with your left eye
- Hold the image about 20 inches away from your face and move it slowly towards you
- The dot should disappear!

Image Formation In The Eye

- Muscles within the eye can be used to change the shape of the lens allowing us focus on objects that are near or far away (in contrast with a camera where the distance between the lens and the focal plane varies)
- An image is focused onto the retina causing rods and cones to become excited which ultimately send signals to the brain



Brightness Adaptation & Discrimination

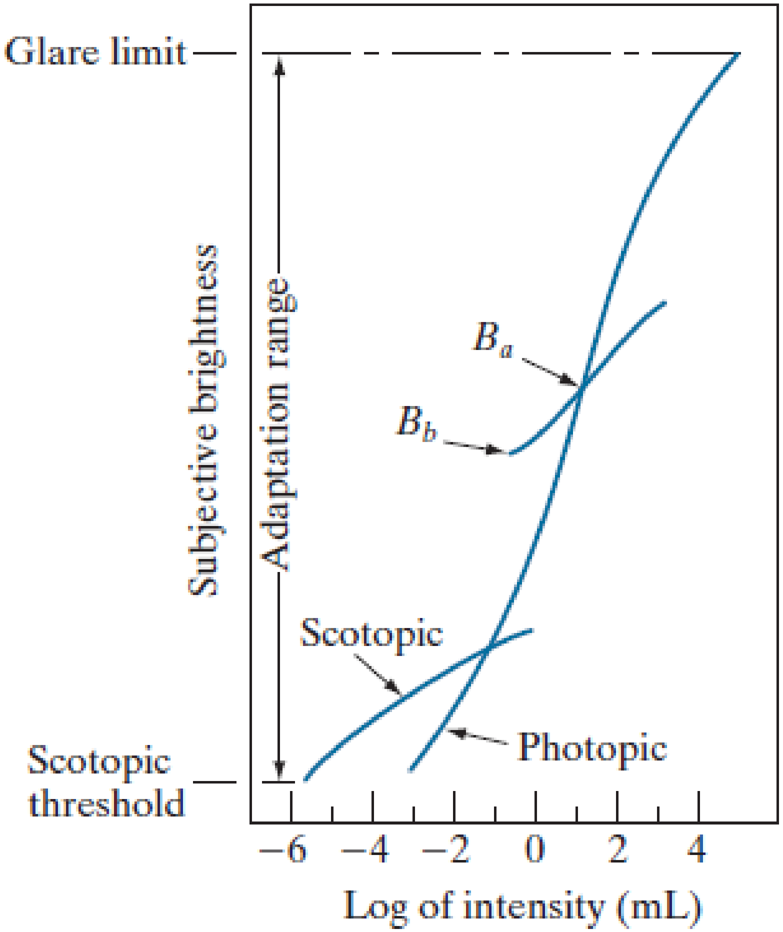
- The human visual system can perceive approximately 10^{10} different light intensity levels.
- At any time instance, we can only discriminate between a much smaller number – *brightness adaptation*.
- Similarly, the perceived intensity of a region is related to the light intensities of the regions surrounding it.

Subjective Brightness Perception

- Subjective brightness perceived by the human visual system, follows a logarithmic function relative to light intensity.
- The human visual system can adapt to a wide range of intensities, approximately 10^6 times, from scotopic to photopic vision.

Subjective Brightness Perception

- The transition from low-light (scotopic) to well-lit (photopic) vision occurs gradually over a range from 0.001 to 0.1 millilambert.
- This transition is depicted by the double branches of the adaptation curve within the specified range.

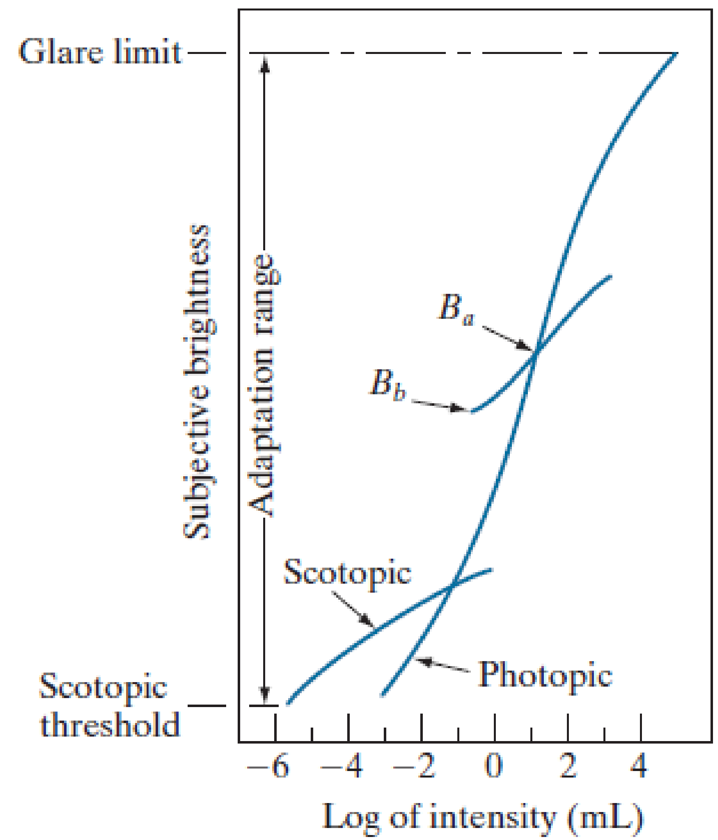


Brightness Adaptation

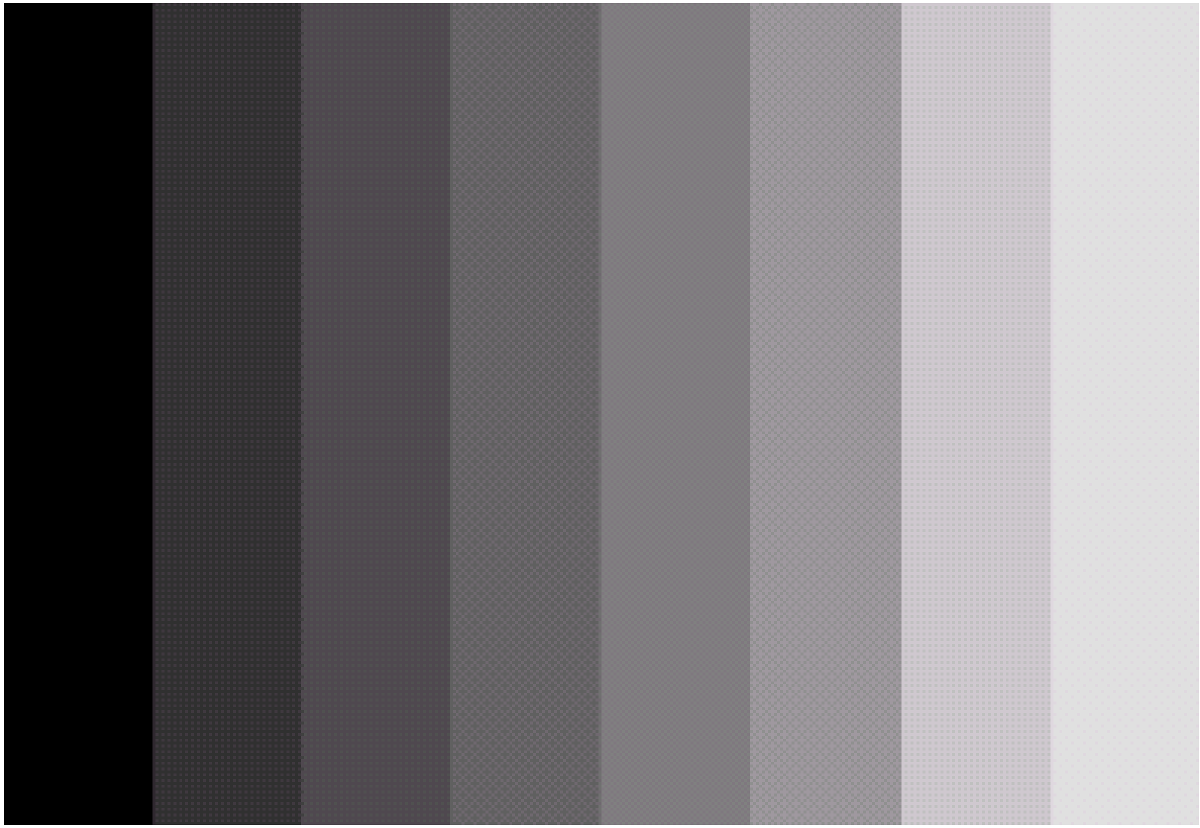
- The key point in interpreting the impressive dynamic range is that the visual system cannot operate over such a range simultaneously.
- Rather, it accomplishes this large variation by changing its overall sensitivity, a phenomenon known as *brightness adaptation*.
- The total range of distinct intensity levels the eye can discriminate simultaneously is rather small when compared with the total adaptation range.

Brightness Adaptation Level

- The current sensitivity level of the visual system under specific conditions.
- This adaptation level (brightness B_a) corresponds to a specific range of subjective brightness perceived by the eye.
- The range of perceived brightness, represented by a short intersecting curve, is limited.
- At the lower end of this range is a level B_b, below which all stimuli are perceived as indistinguishable blacks.
- The upper portion of the curve is not constrained, but extending it too far loses significance as higher intensities would elevate the adaptation level

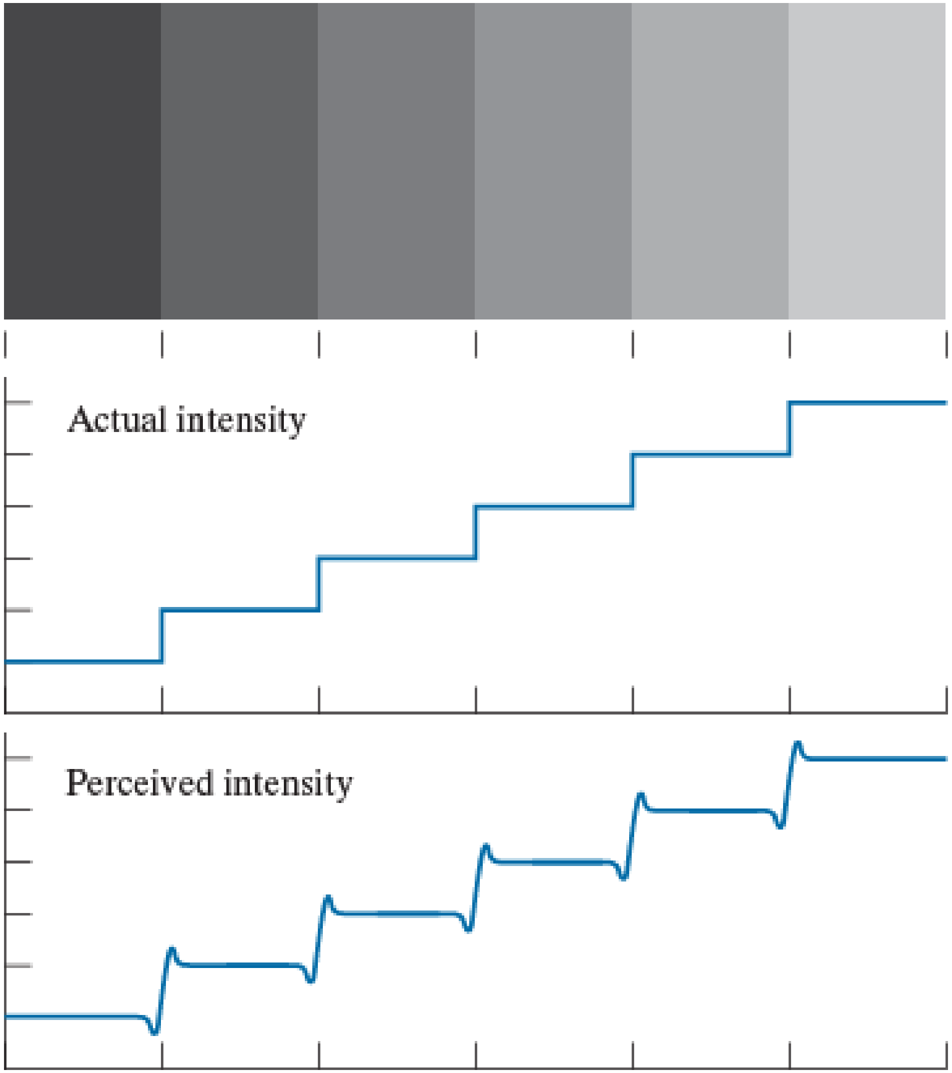


Brightness Adaptation & Discrimination (cont…)

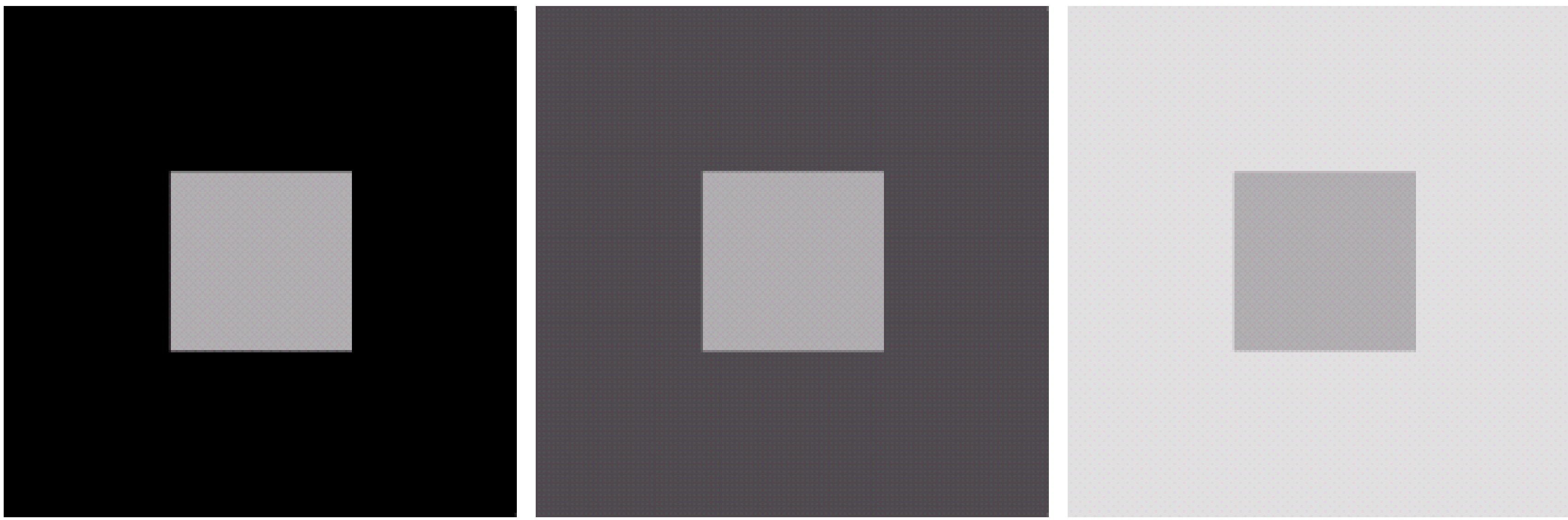


An example of Mach bands

Brightness Adaptation & Discrimination (cont…)



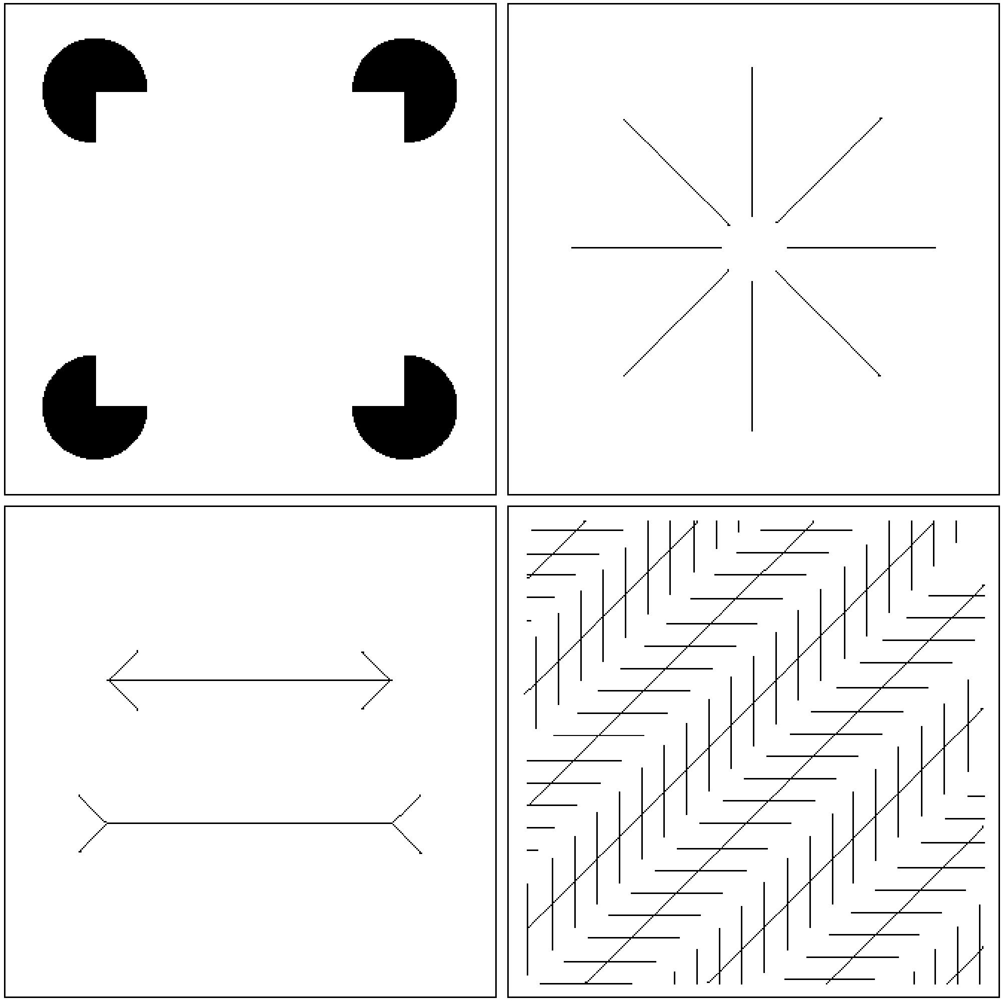
Brightness Adaptation & Discrimination (cont…)



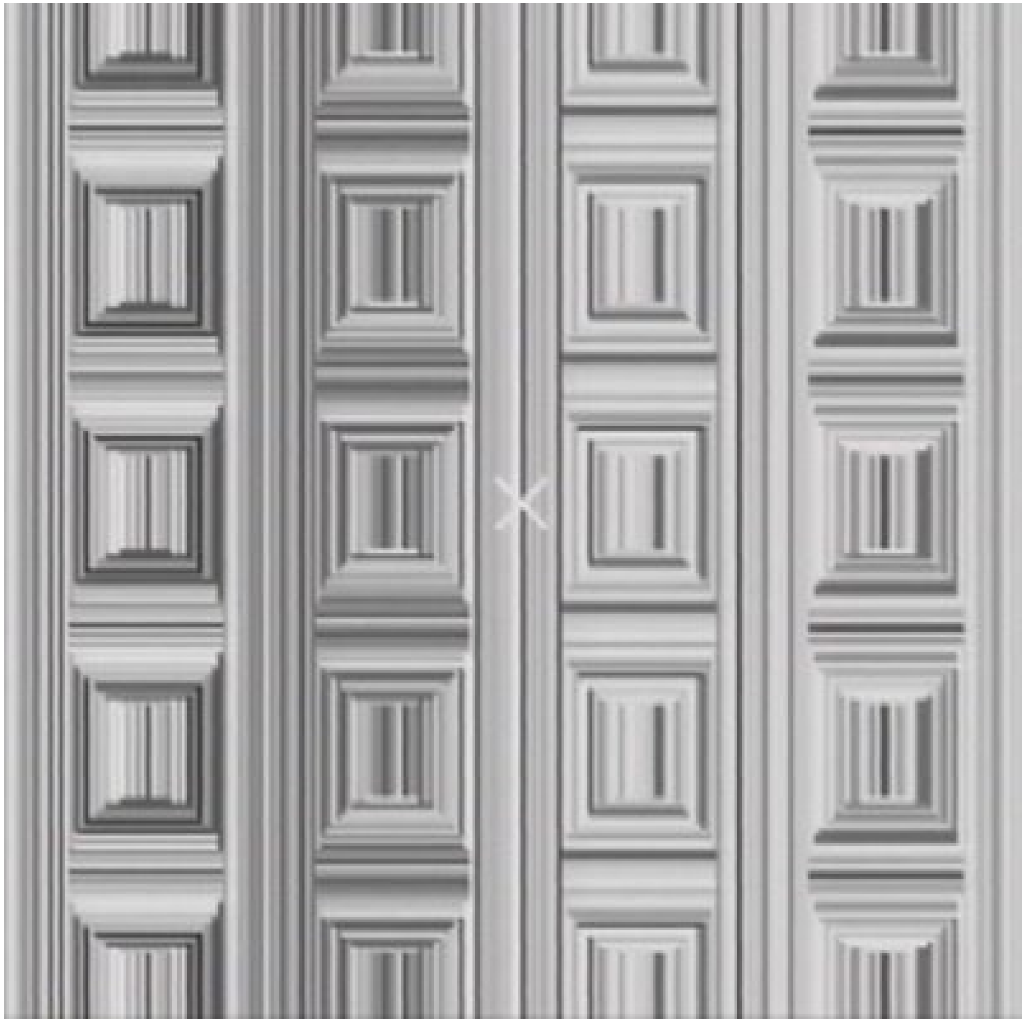
An example of *simultaneous contrast*
(square has constant brightness across shapes)

Optical Illusions

- Our visual system plays many interesting tricks on us

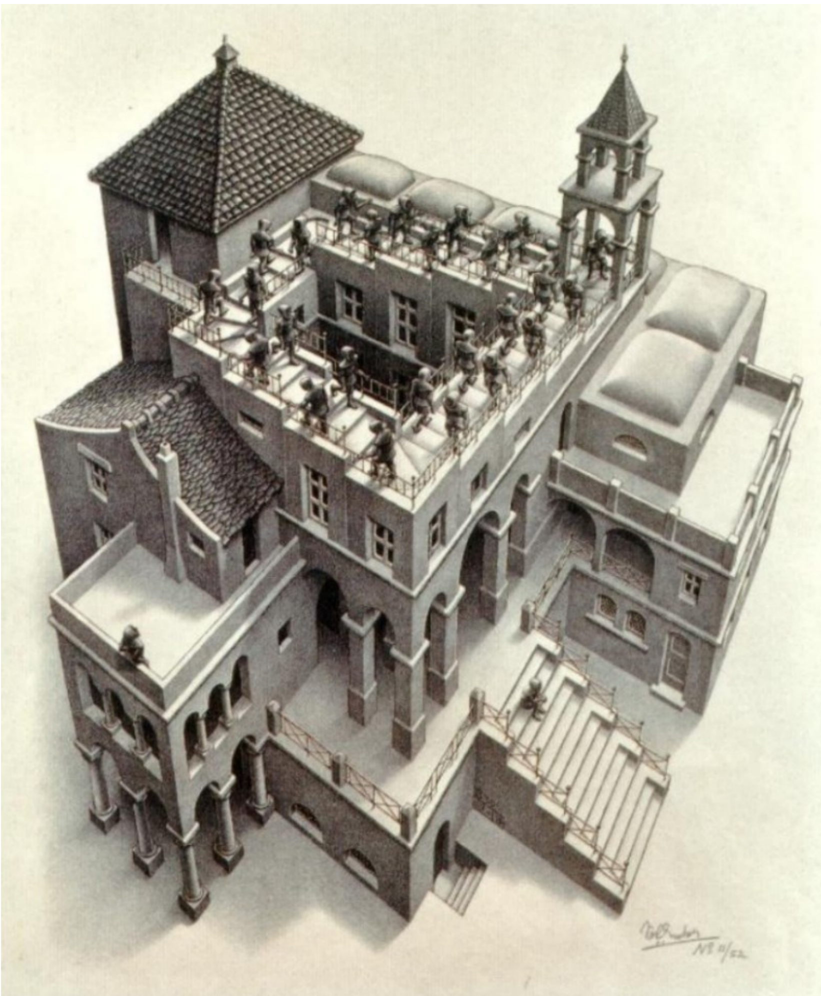


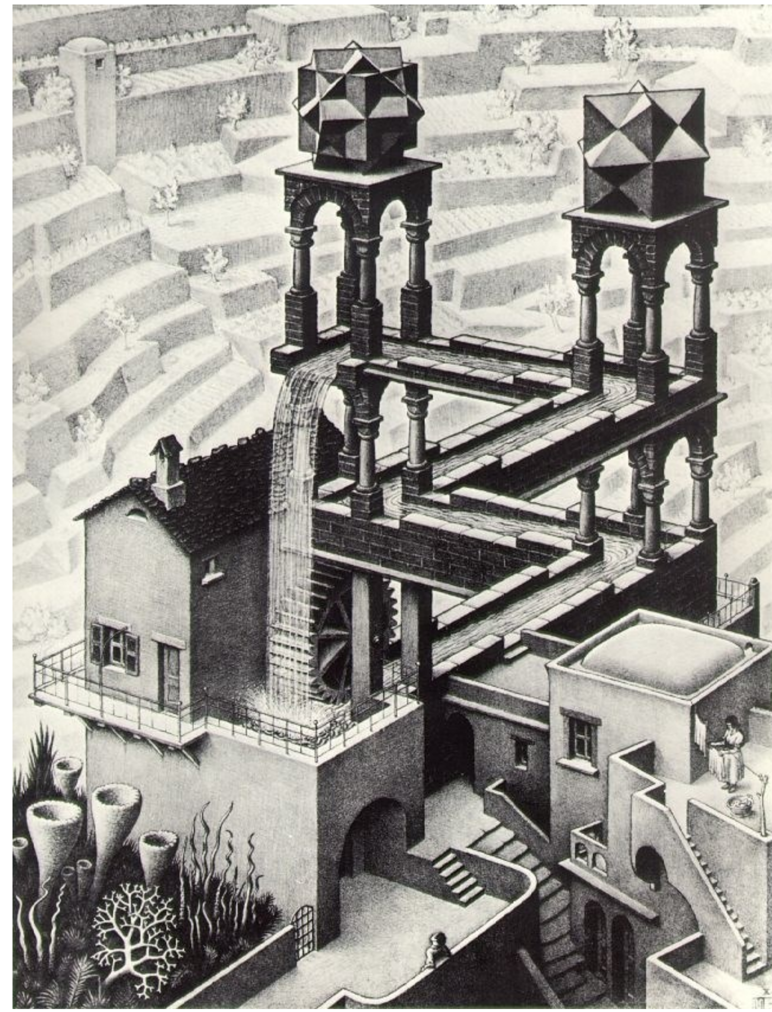
Optical Illusions (cont…)



Stare at the cross in the middle of the image and think circles

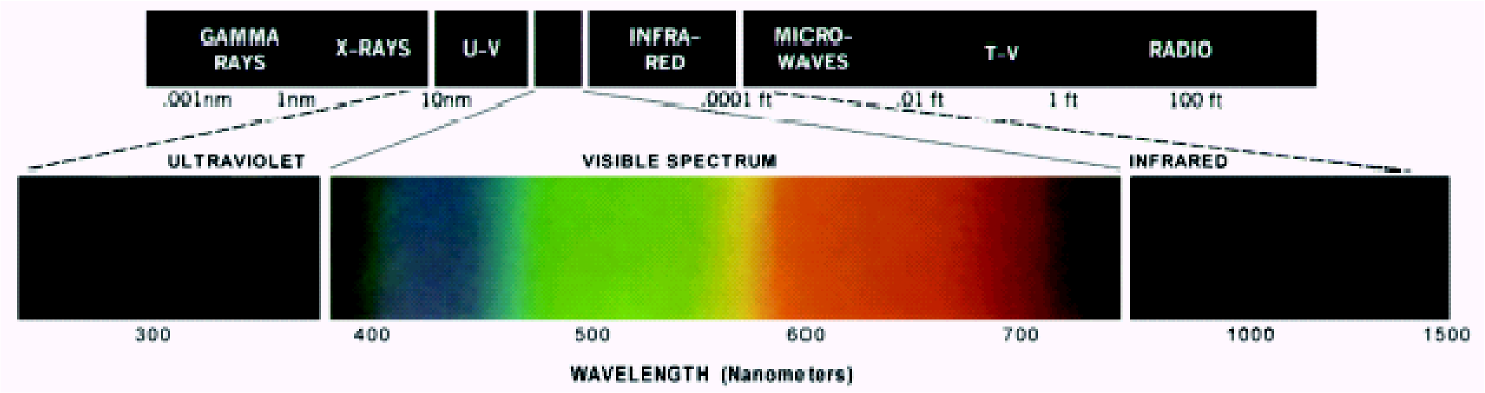
Optical Illusions (cont…)





Light And The Electromagnetic Spectrum

- Light is just a particular part of the electromagnetic spectrum that can be sensed by the human eye
- The electromagnetic spectrum is split up according to the wavelengths of different forms of energy



Reflected Light

- The colours that we perceive are determined by the nature of the light reflected from an object
- For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object

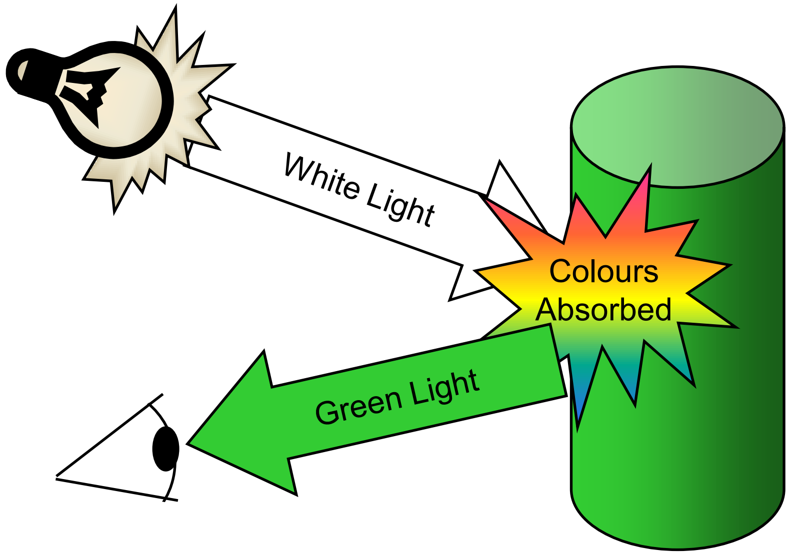


Image Acquisition

Images are typically generated by *illuminating a scene* and absorbing the energy reflected by the objects in that scene

Typical notions of illumination and scene can be way off:

- X-rays of a skeleton
- Ultrasound of an unborn baby
- Electro-microscopic images of molecules

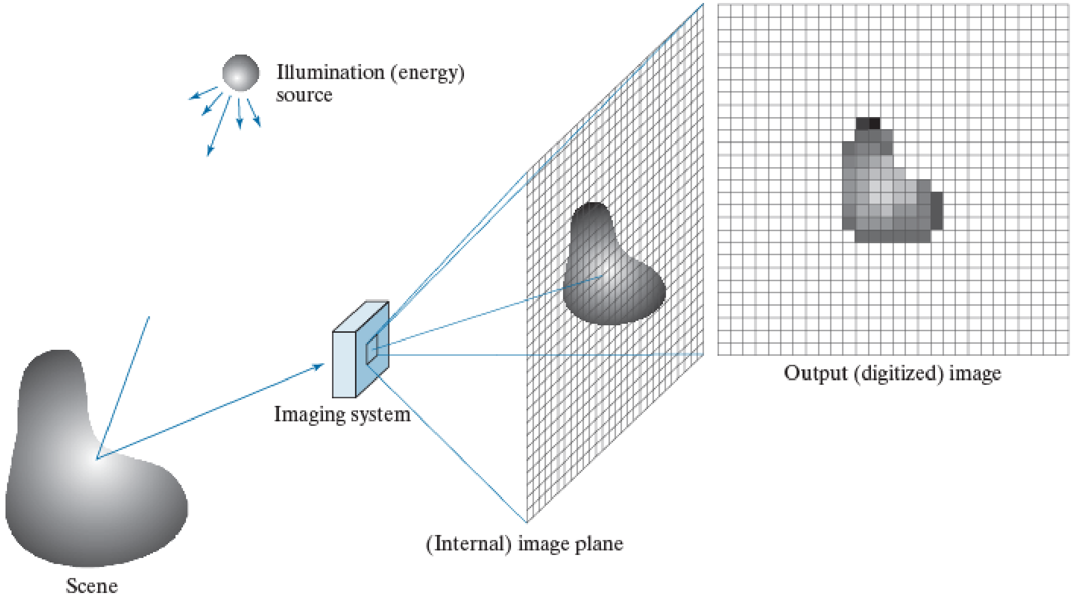
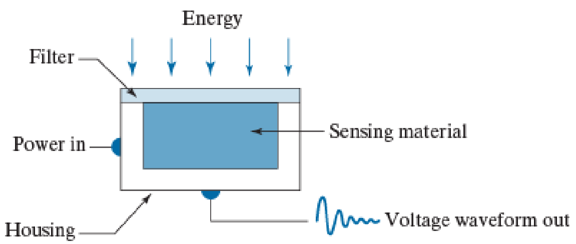
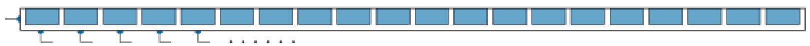


Image Sensing and Acquisition

- Sensors transform the incoming energy into voltage and the output of the sensor is digitized.
- Top: single sensing element
- Middle: Line (of image) sensors
- Bottom: 2D array (of image) sensors





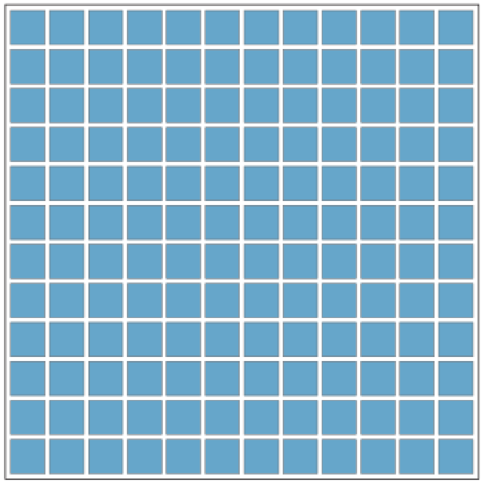


Image Sensing and Acquisition

- Using a filter in front of a sensor enhances its selectivity by favoring specific wavelengths of light, i.e., an optical green-transmission filter emphasizes light within the green band of the color spectrum.
- 2-D image generated by relative displacements in both the x- and y axis between the sensor and the area being imaged.

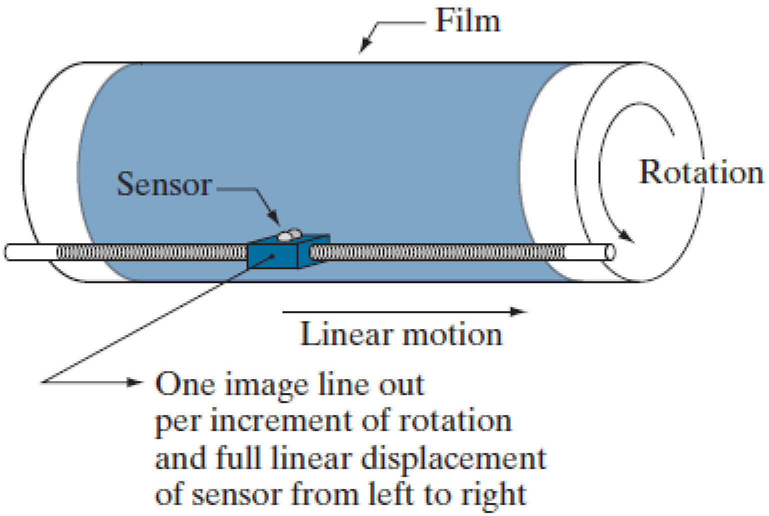


Image Sensing

Using Sensor Strips and Rings

Image acquisition using a linear sensor strip.

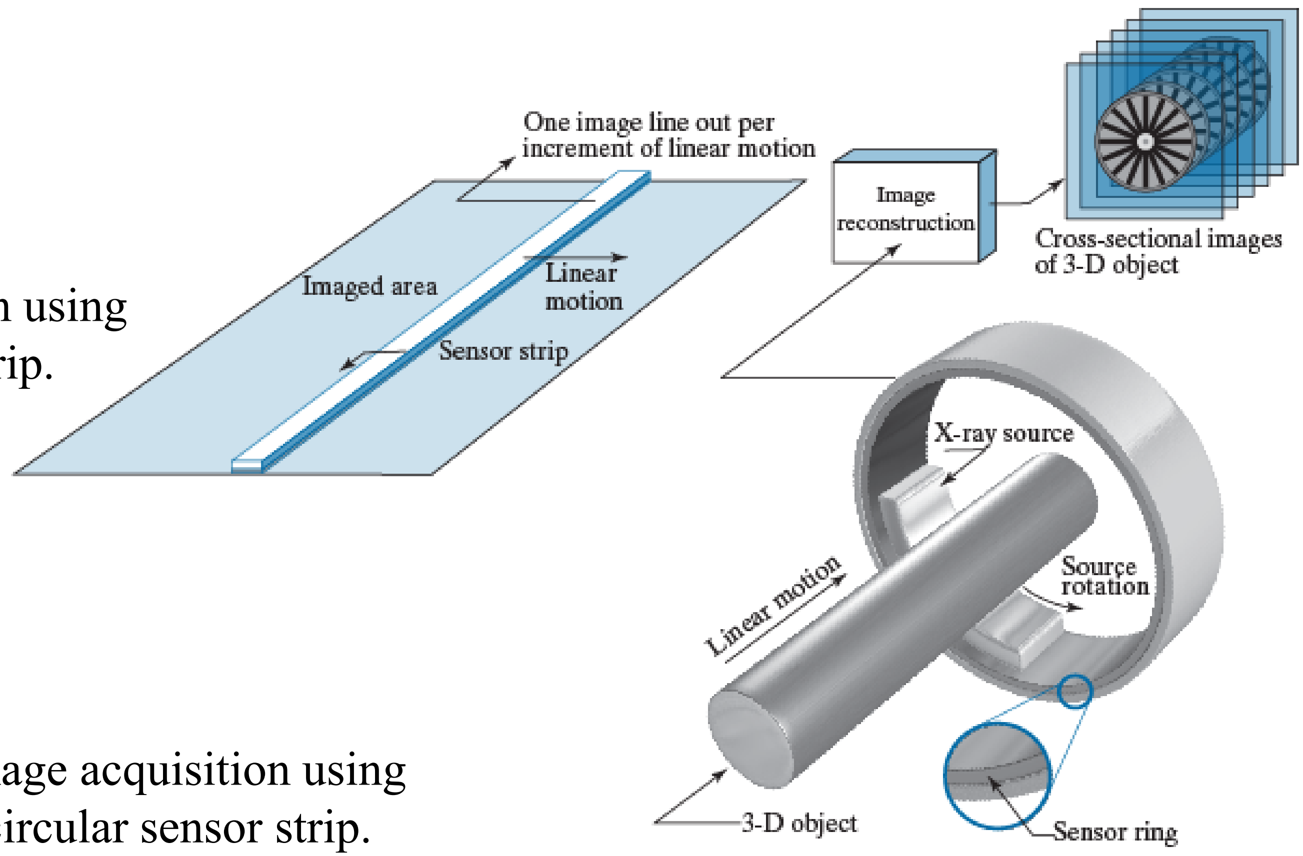
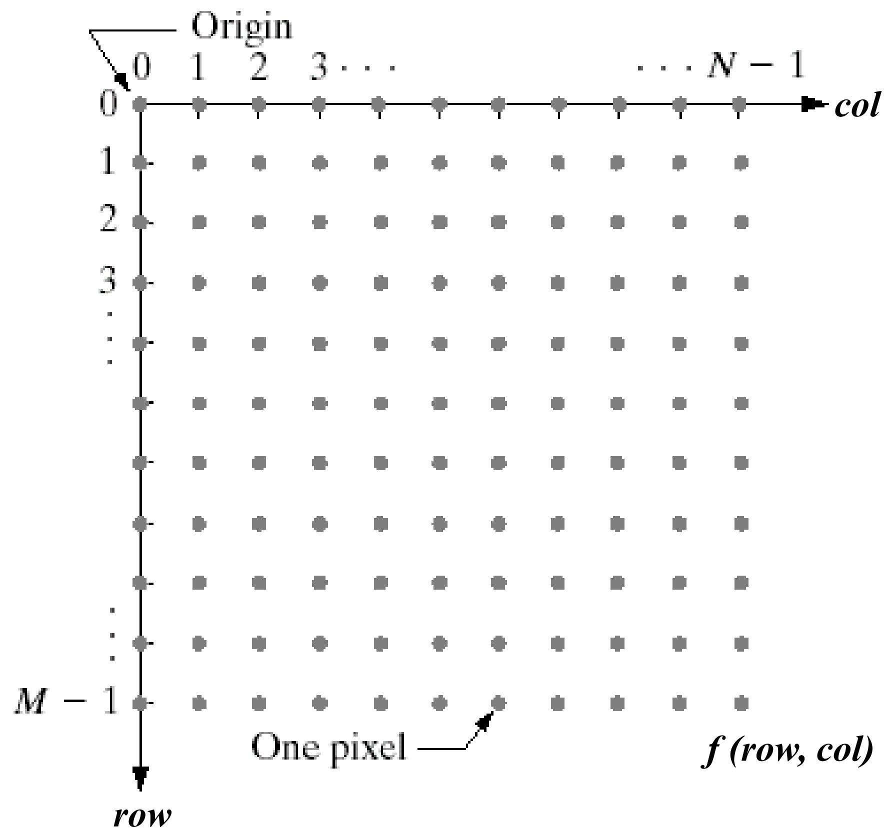


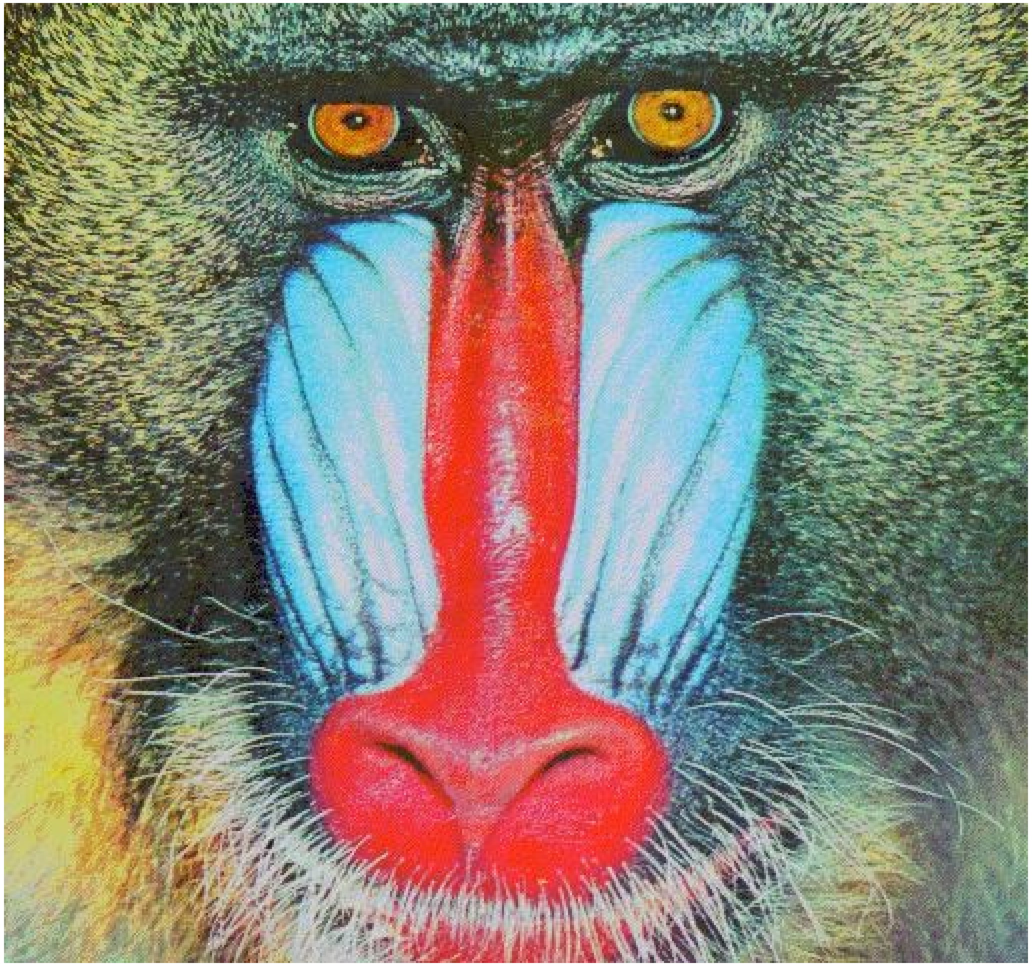
Image acquisition using a circular sensor strip.

Image Representation

- A digital image is composed of M rows and N columns of pixels each storing a value
- Pixel values are in the range 0-255 (black-white)
- Images can easily be represented as matrices



Colour images



Colour images

Image Sampling And Quantisation

- A digital sensor can only measure a limited number of **samples** at a **discrete** set of energy levels
- Quantisation is the process of converting a continuous **analogue** signal into a digital representation of this signal

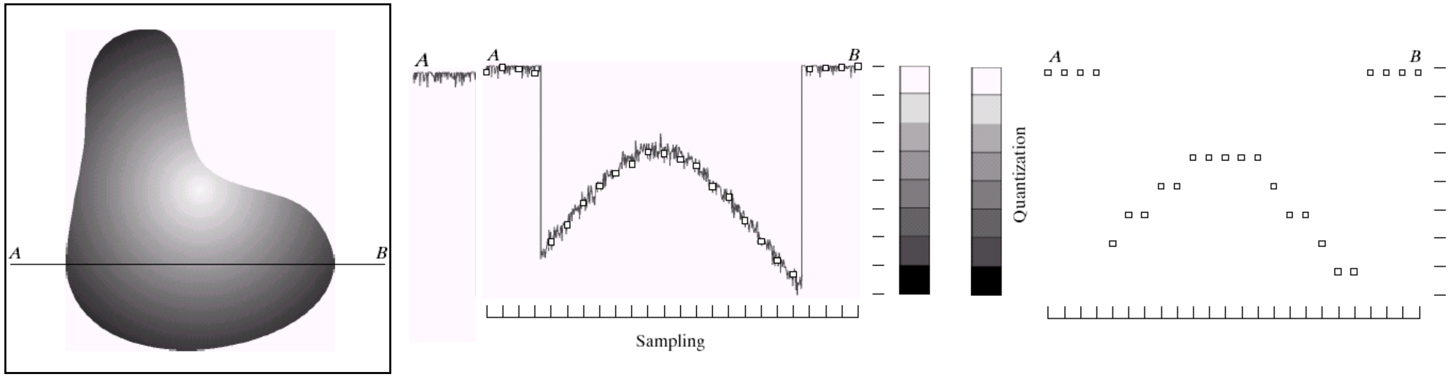


Image Sampling And Quantisation (cont…)

- Remember that a digital image is always only an **approximation** of a real world scene

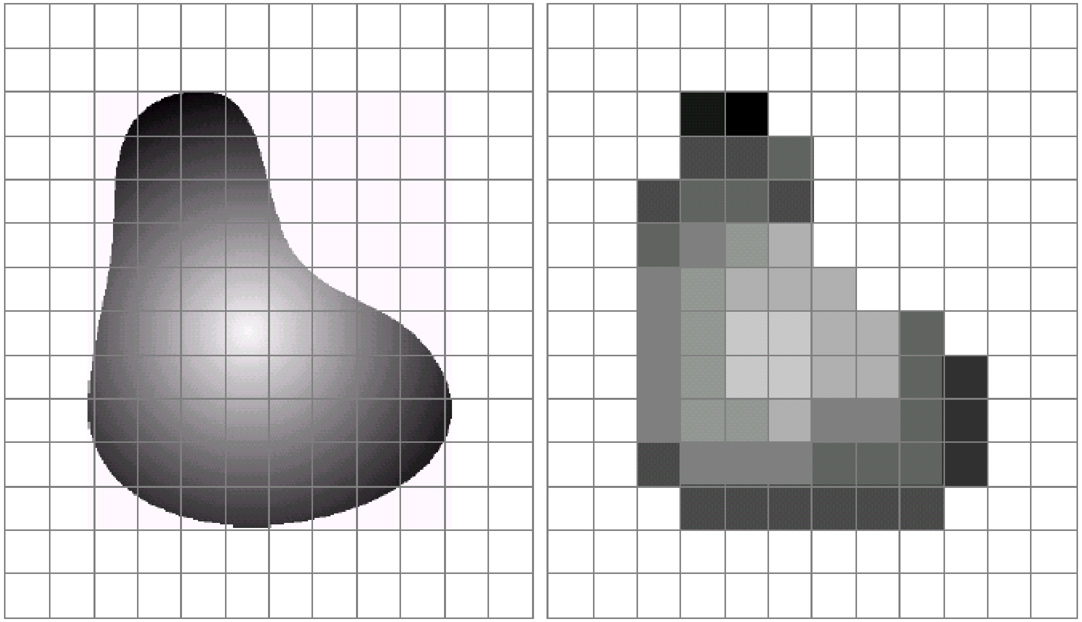
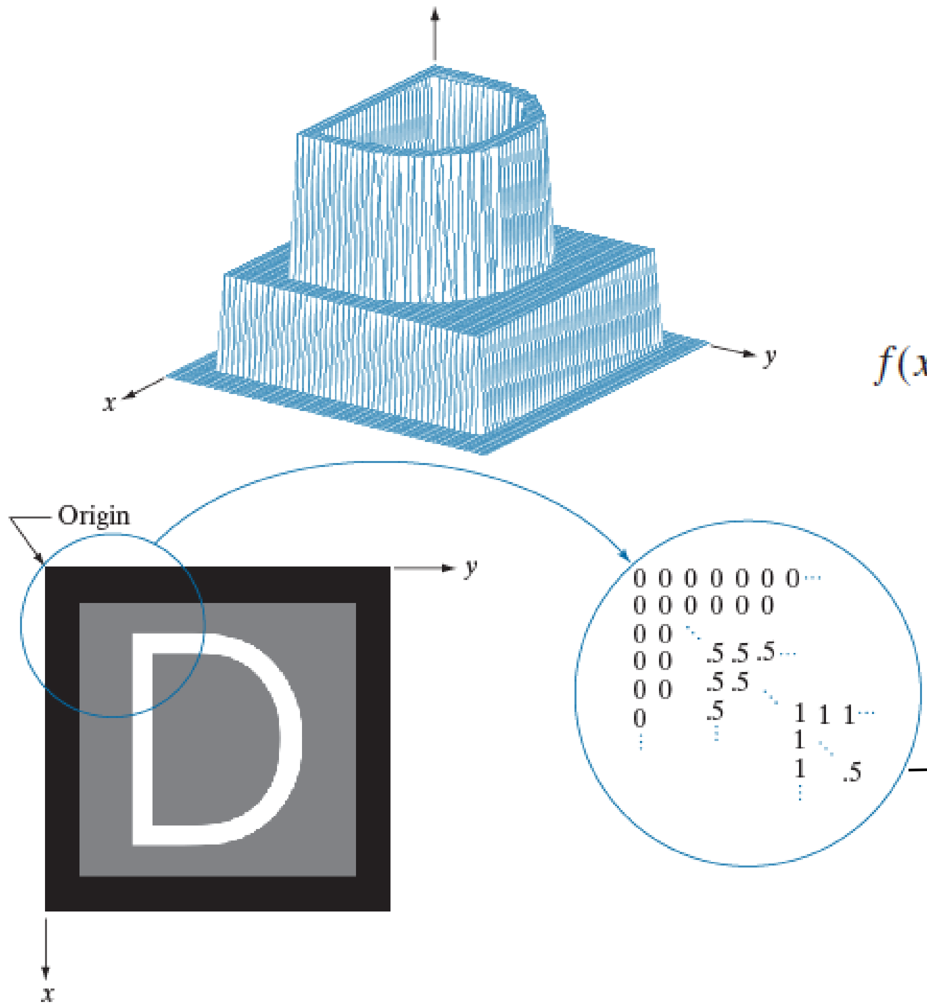


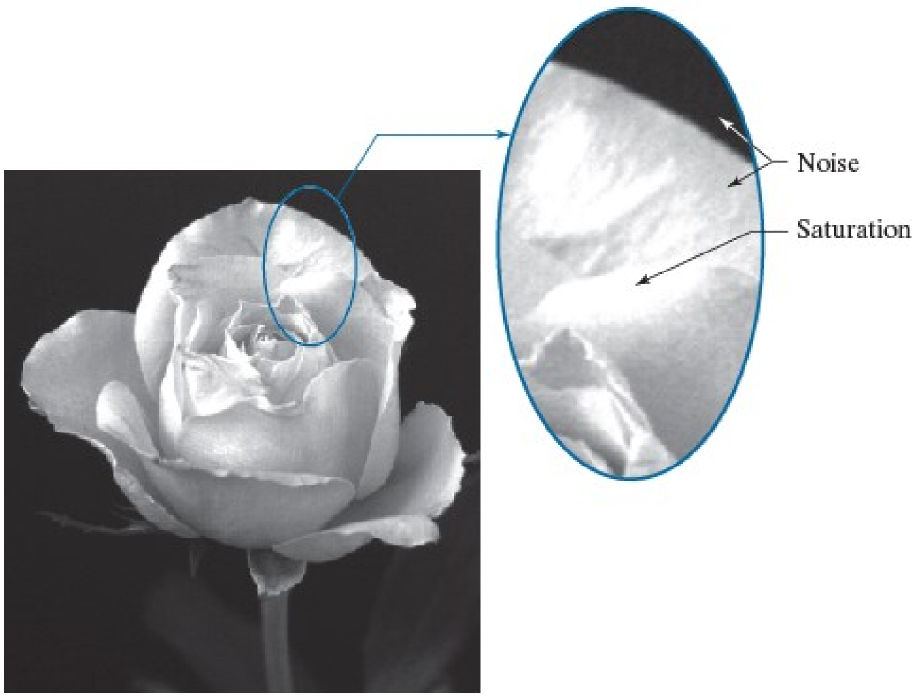
Image Representation



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

Saturation & Noise

- **Dynamic range**: The ratio of the maximum (*saturation*) to the minimum (*noise*) detectable intensity of the imaging system.
- Noise generally appear as a grainy texture pattern in the darker regions and masks the lowest detectable true intensity level



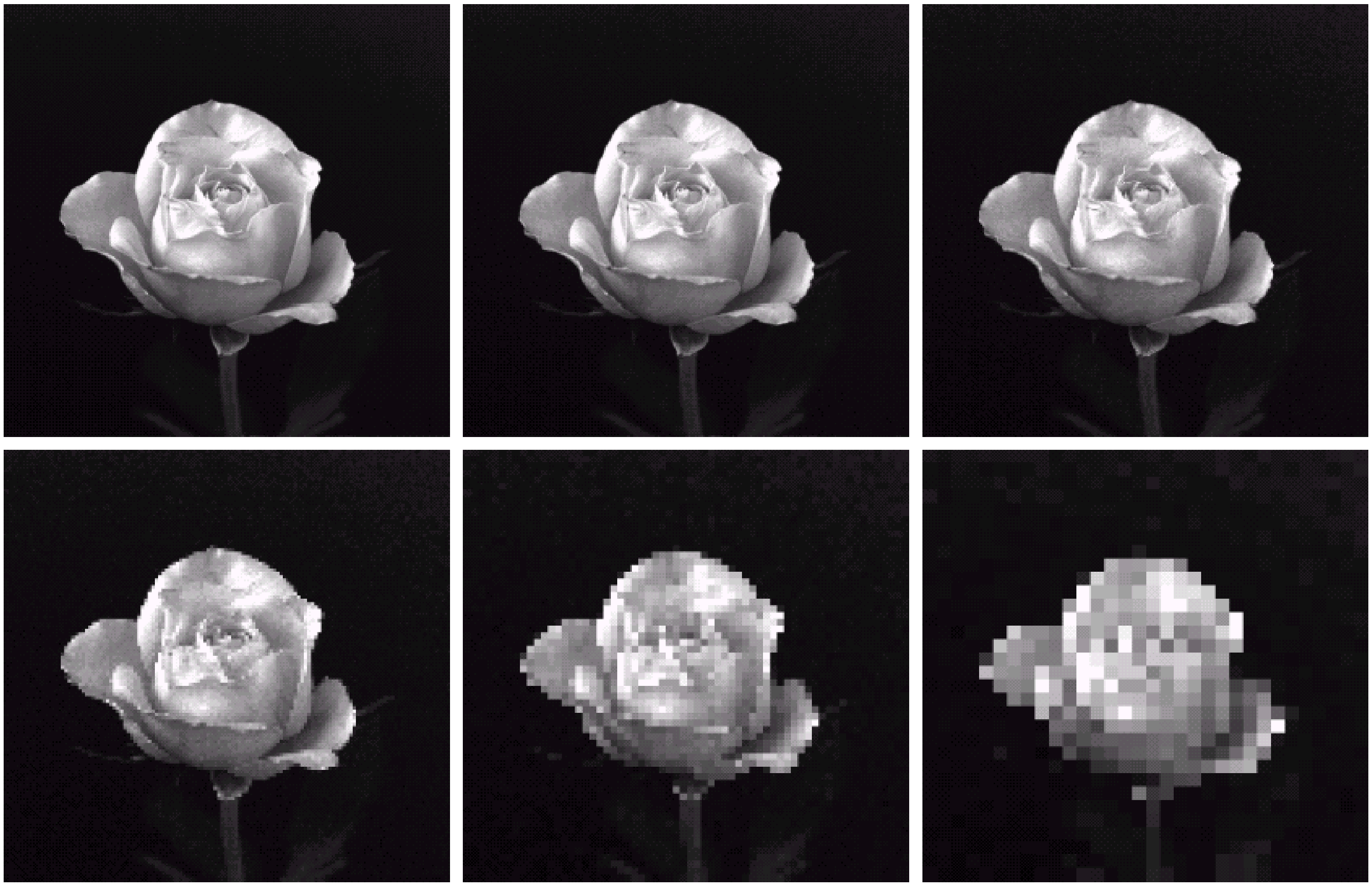
Spatial Resolution

- The spatial resolution of an image is determined by how sampling was carried out
- Spatial resolution simply refers to the smallest discernable detail in an image
  - Vision specialists will often talk about pixel size
  - Graphic designers will talk about dots per inch (DPI)

Spatial Resolution (cont…)

Spatial Resolution (cont…)

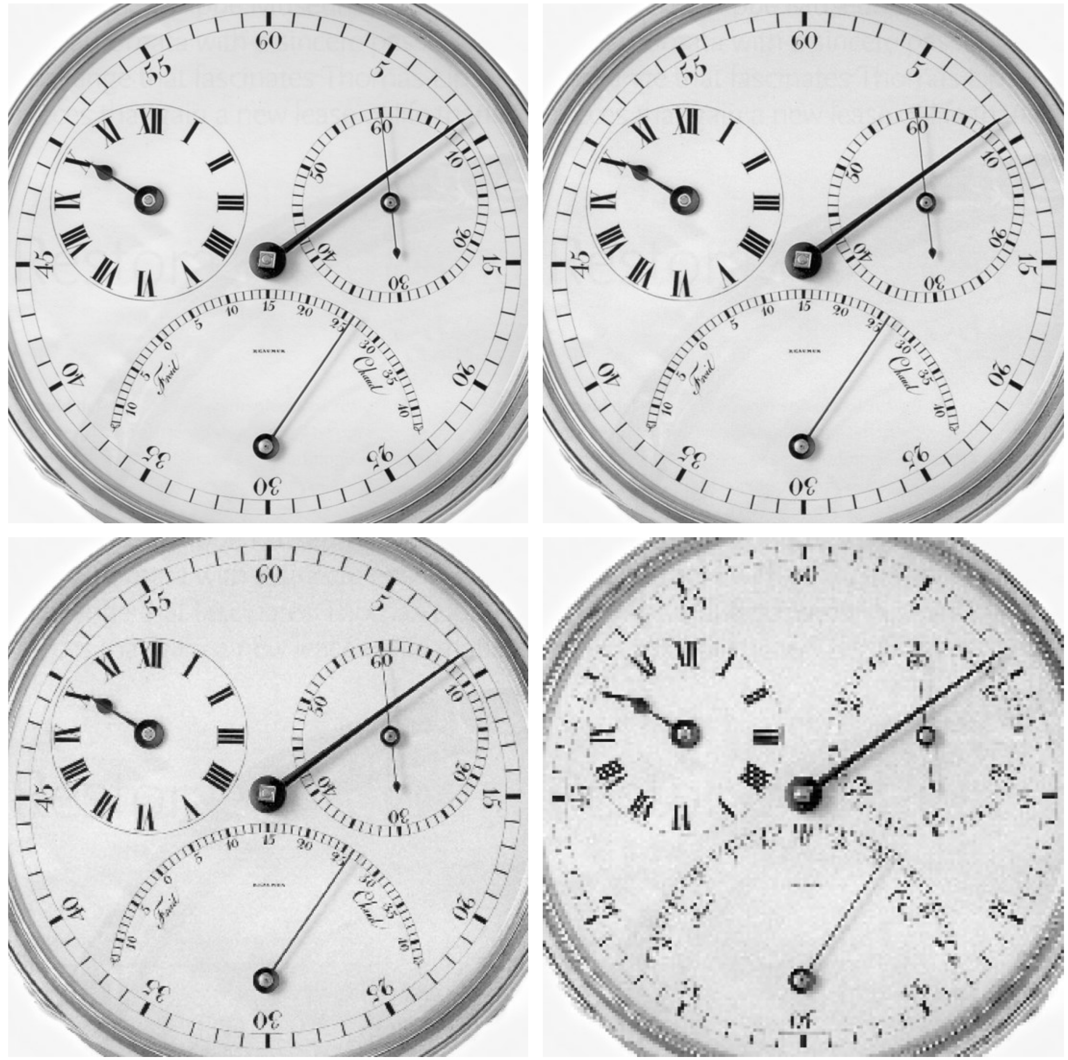
1024 * 1024 512 * 512 256 * 256



128 * 128 64 * 64 32 * 32

Spatial Resolution (cont…)

- Effects of reducing spatial resolution. The images shown are at:
  (a) 930 dpi,
  (b) 300 dpi,
  (c) 150 dpi, and
  (d) 72 dpi.



Intensity Level Resolution

- Intensity level resolution refers to the number of intensity levels used to represent the image
  - The more intensity levels used, the finer the level of detail discernable in an image
  - Intensity level resolution is usually given in terms of the number of bits used to store each intensity level

Number of Bits	Number of Intensity Levels	Examples
1	2	0, 1
2	4	00, 01, 10, 11
4	16	0000, 0101, 1111
8	256	00110011, 01010101
16	65,536	1010101010101010

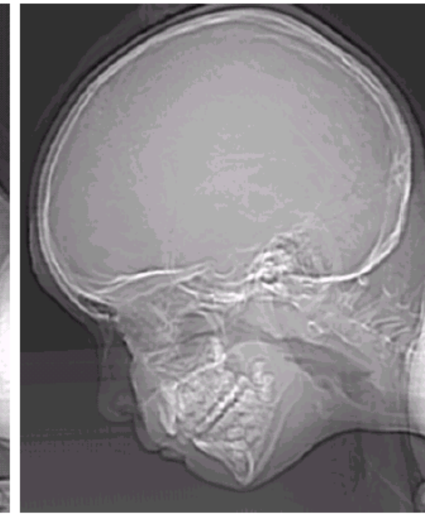
Intensity Level Resolution (cont…)

256 grey levels (8 bits per pixel)

128 grey levels (7 bpp)

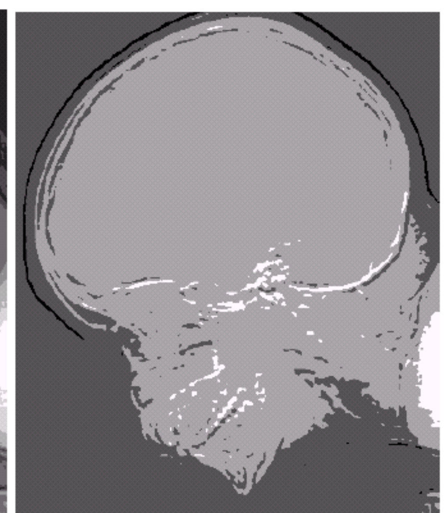
64 grey levels (6 bpp)

32 grey levels (5 bpp)

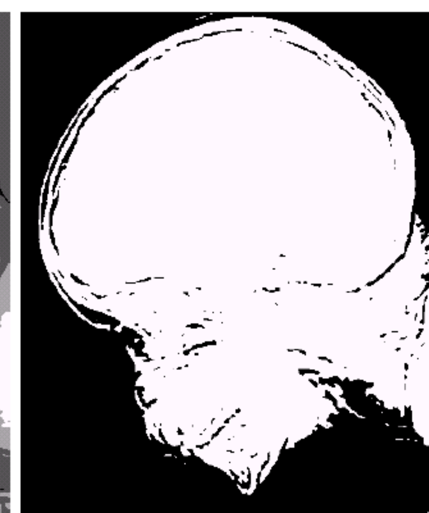


16 grey levels (4 bpp)

8 grey levels (3 bpp)



4 grey levels (2 bpp)



2 grey levels (1 bpp)

Intensity Level Resolution (cont…)

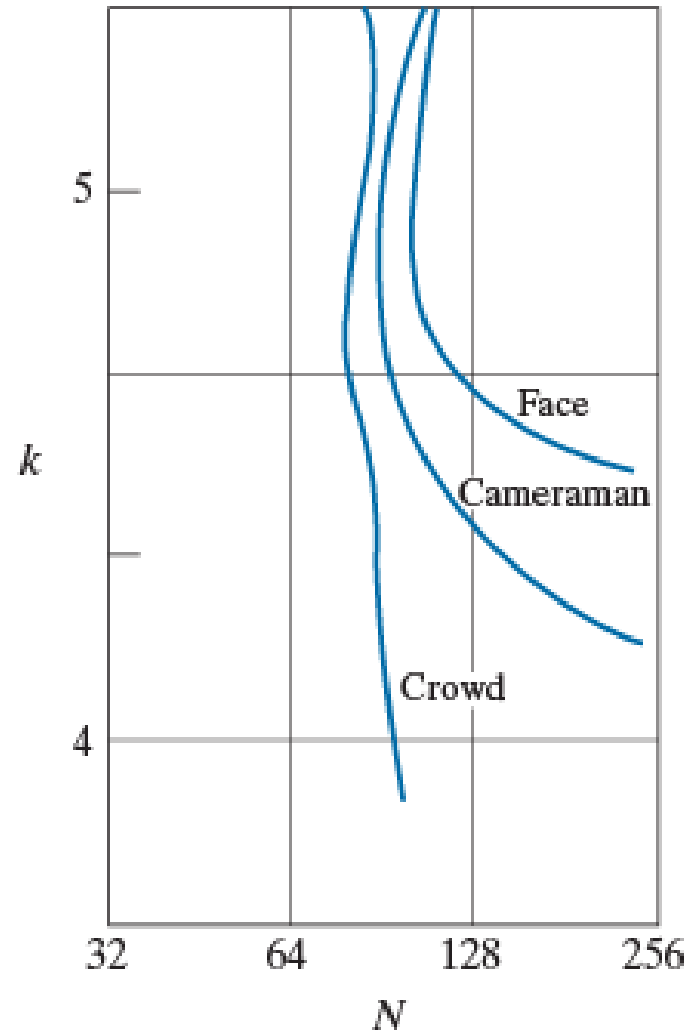
Low Detail

Medium Detail

High Detail

Intensity Level Resolution (cont…)

- $b = N^2 k$
- *Isopreference* curves represent the dependence between intensity and spatial resolutions.
  - Points lying on a curve represent images of “equal” quality as described by observers.
  - The curves become more vertical as the degree of detail increases (a lot of detail need less intensity levels).



Resolution: How Much Is Enough?

The big question with resolution is always *how much is enough*?

- This all depends on what is in the image and what you would like to do with it
- Key questions include
  - Does the image look aesthetically pleasing?
  - Can you see what you need to see within the image?

Resolution: How Much Is Enough? (cont…)

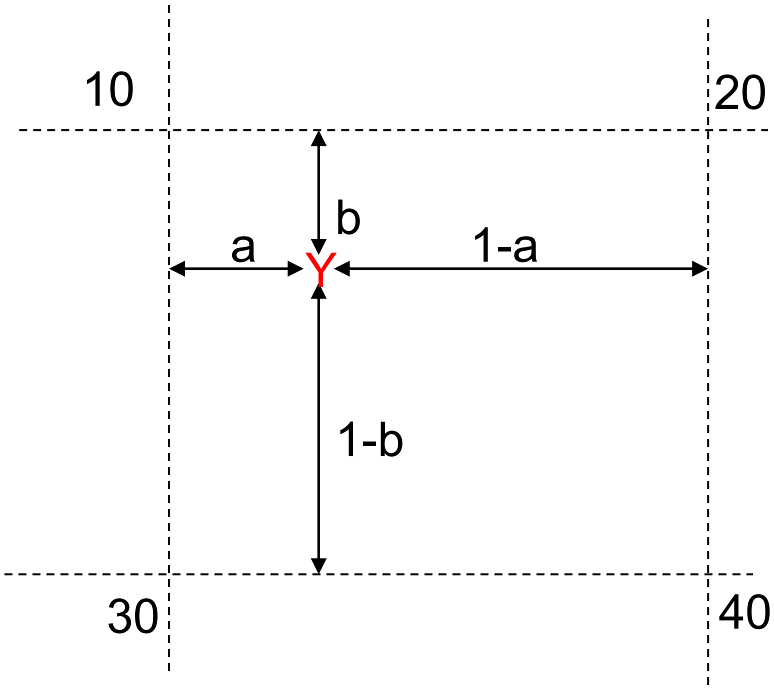
The picture on the right is fine for counting the number of cars, but not for reading the number plate

Interpolation

- The process of using known data to estimate values at unknown locations
- Basic operation for shrinking, zooming, rotation and translation
  - e.g. a 500x500 image has to be enlarged by 1.5 to 750x750 pixels
  - Create an imaginary 750x750 grid with the same pixel spacing as the original and then shrink it to 500x500
  - The 750x750 shrunk pixel spacing will be less than the spacing in the original image.
  - Pixel values have to be determined in between the original pixel locations

Interpolation (cont.)

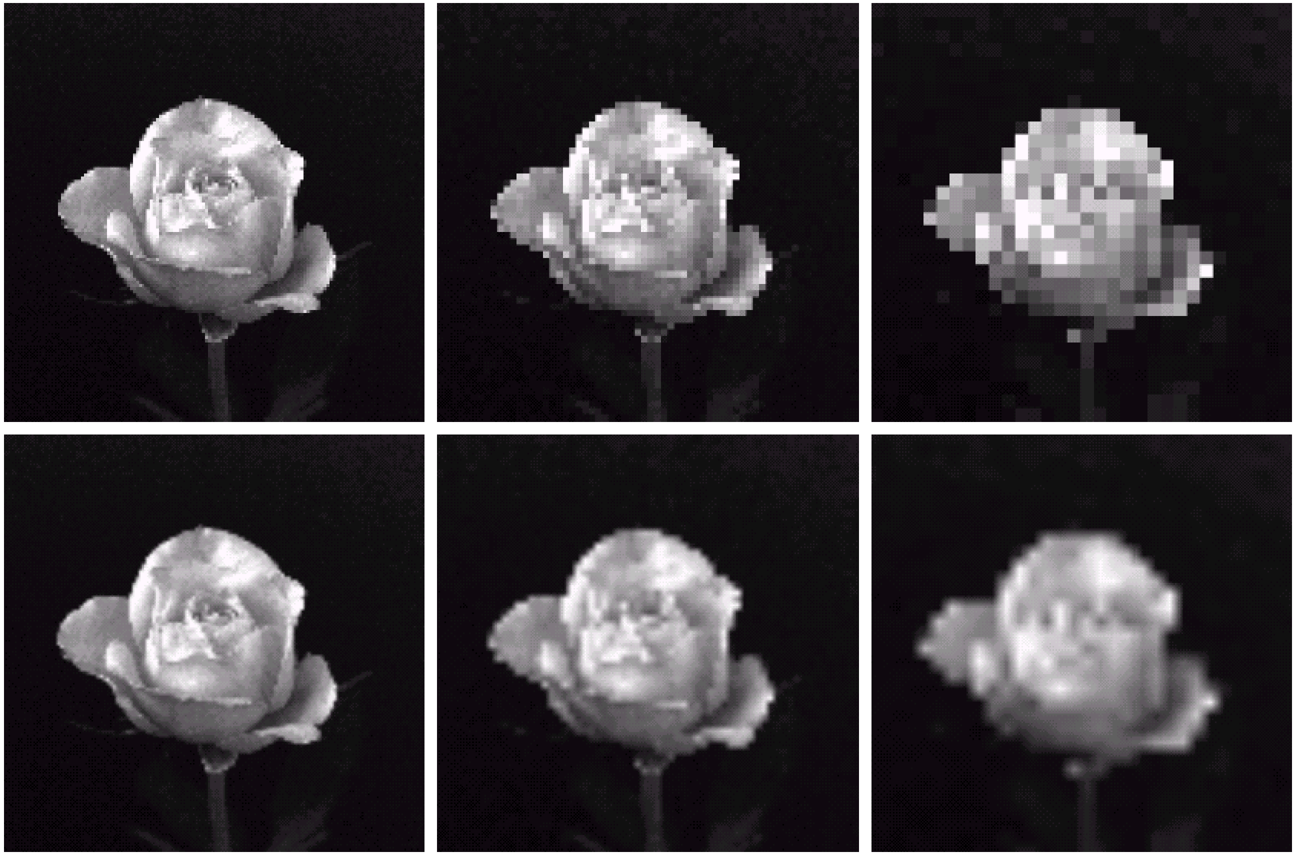
- How to determine pixel values
  - Nearest neighbour
  - Bilinear
  - Bicubic
  - 2D sinc



- Nearest Neighbour
  - Whats is the value of Y;

10

Interpolation (cont...)



a b c
d e f

FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Distances between pixels

- For pixels $p(x,y)$, $q(s,t)$ and $z(v,w)$, D is a distance function or metric if:

$$a)\, D(p,q) \geq 0 \;\; (D(p,q) = 0 \;\; iff \;\; p = q),$$
$$b)\, D(p,q) = D(q,p),$$
$$c)\, D(p,z) \leq D(p,q) + D(q,z).$$

- The Euclidean distance between p and q is defined as:

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

Distances between pixels (cont.)

- The city-block (Manhattan) or D_4 distance between p and q is defined as:

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

- Pixels having the city-block distance from a pixel (x,y) less than or equal to some value T form a diamond centered at (x,y) . For example, for $T=2$:

		2		
	2	**1**	**2**	
2	**1**	**0**	**1**	**2**
	2	**1**	**2**	
		2		

Distances between pixels (cont.)

- The chessboard or D_8 distance between p and q is defined as:

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

- Pixels having the D_8 distance from a pixel (x,y) less than or equal to some value T form a square centered at (x,y). For example, for $T=2$:

2	**2**	**2**	**2**	**2**
2	**1**	**1**	**1**	**2**
2	**1**	**0**	**1**	**2**
2	**1**	**1**	**1**	**2**
2	**2**	**2**	**2**	**2**

Mathematical operations used in digital image processing

- Arithmetic operations (e.g image subtraction pixel by pixel)
- Matrix and vector operations
- Linear (e.g. sum) and nonlinear operations (e.g. min and max)
- Set and logical operations
- Spatial and neighbourhood operations (e.g. local average)
- Geometric spatial transformations (e.g. rotation)

Image subtraction

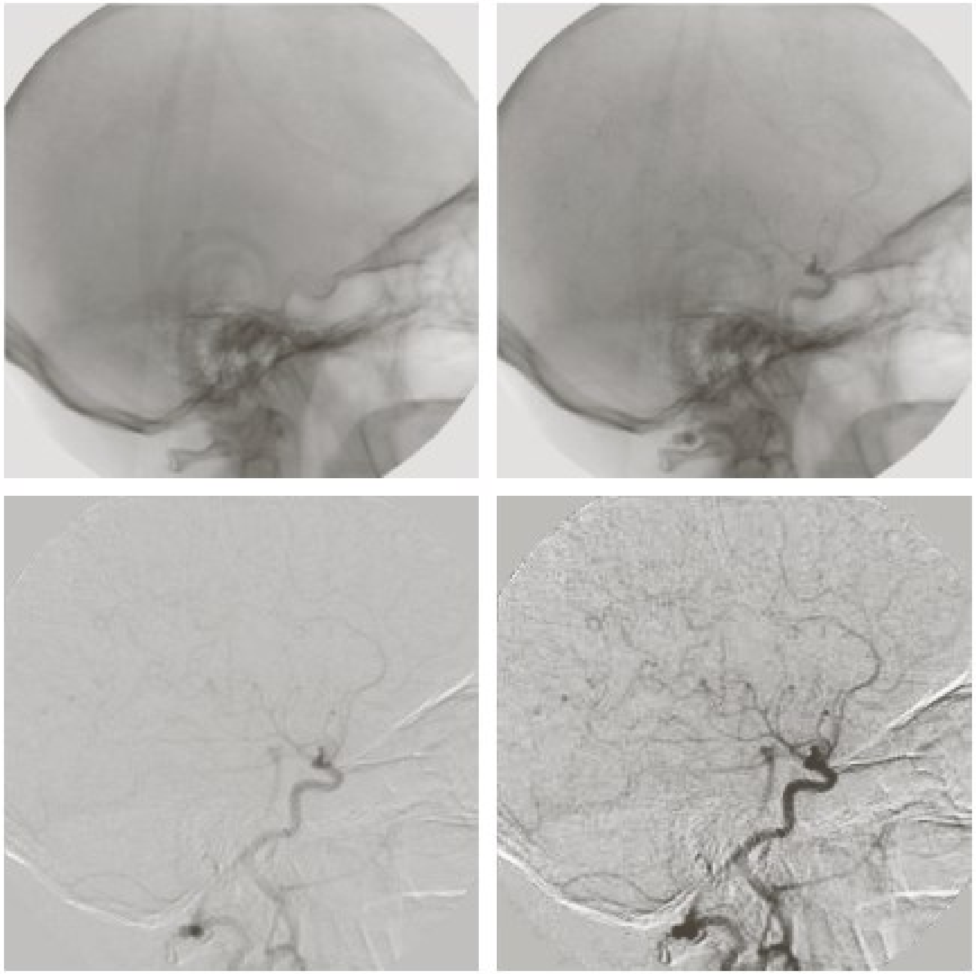


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Image multiplication

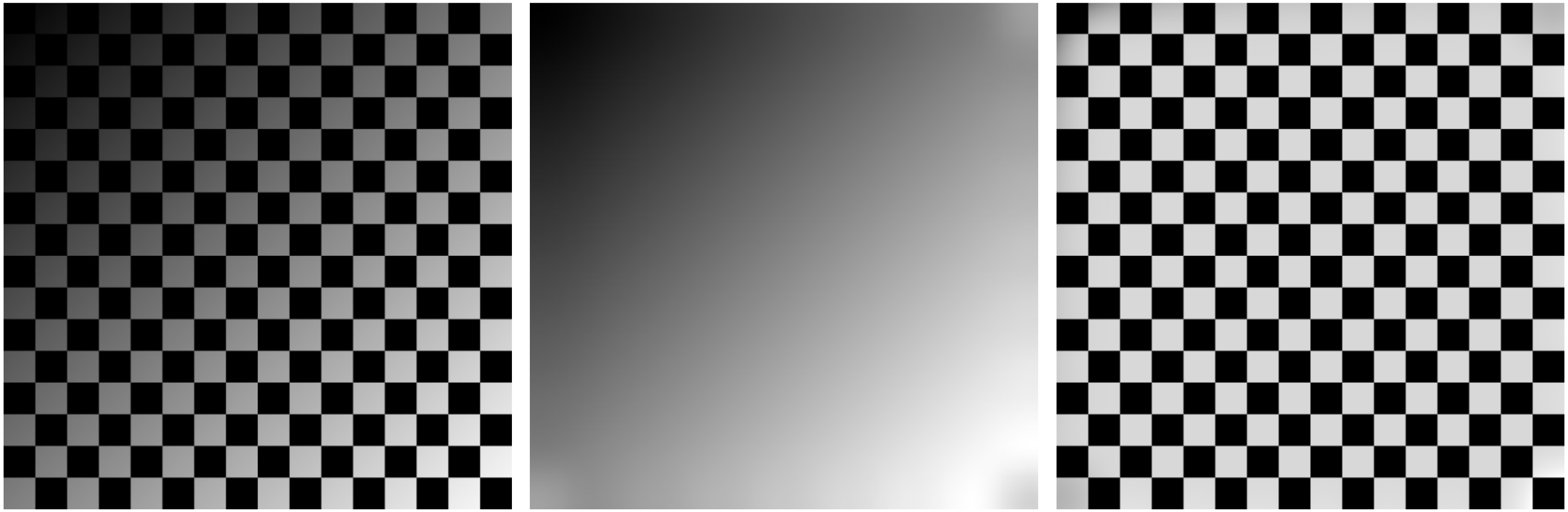


FIGURE Shading correction. (a) Shaded test pattern. (b) Estimated shading pattern. (c) Product of (a) by the reciprocal of (b). (See Section 3.5 for a discussion of how (b) was estimated.)

Image multiplication (cont.)

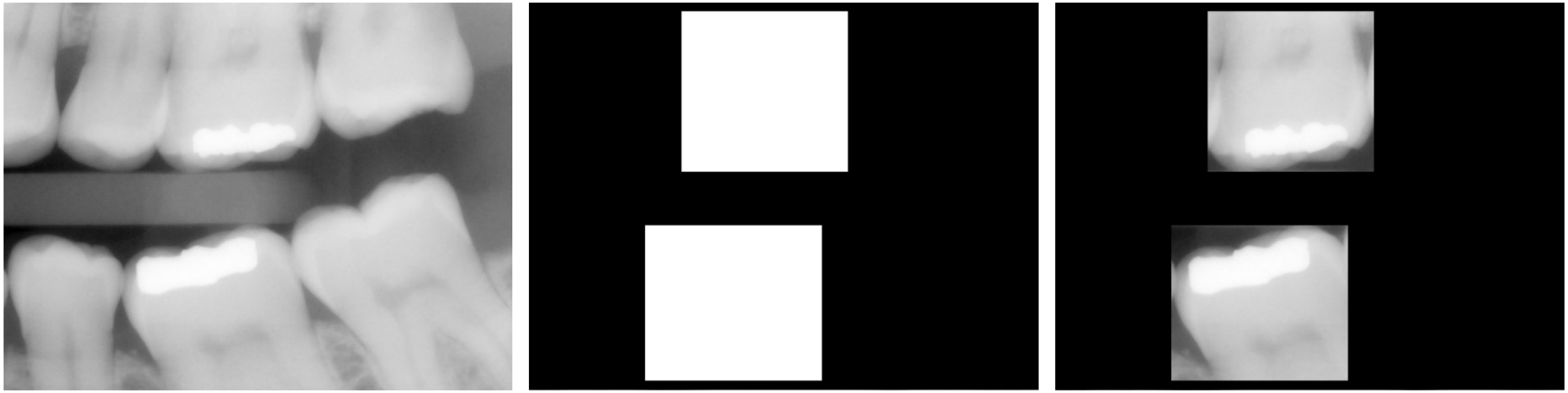
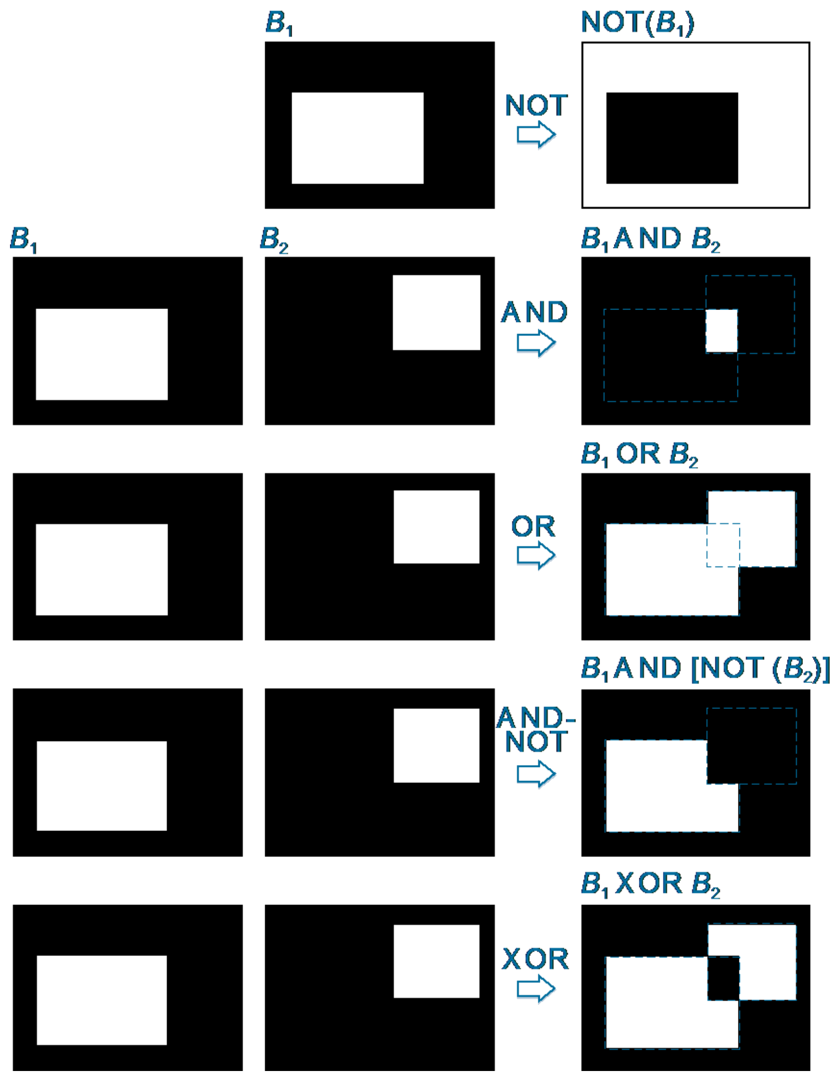
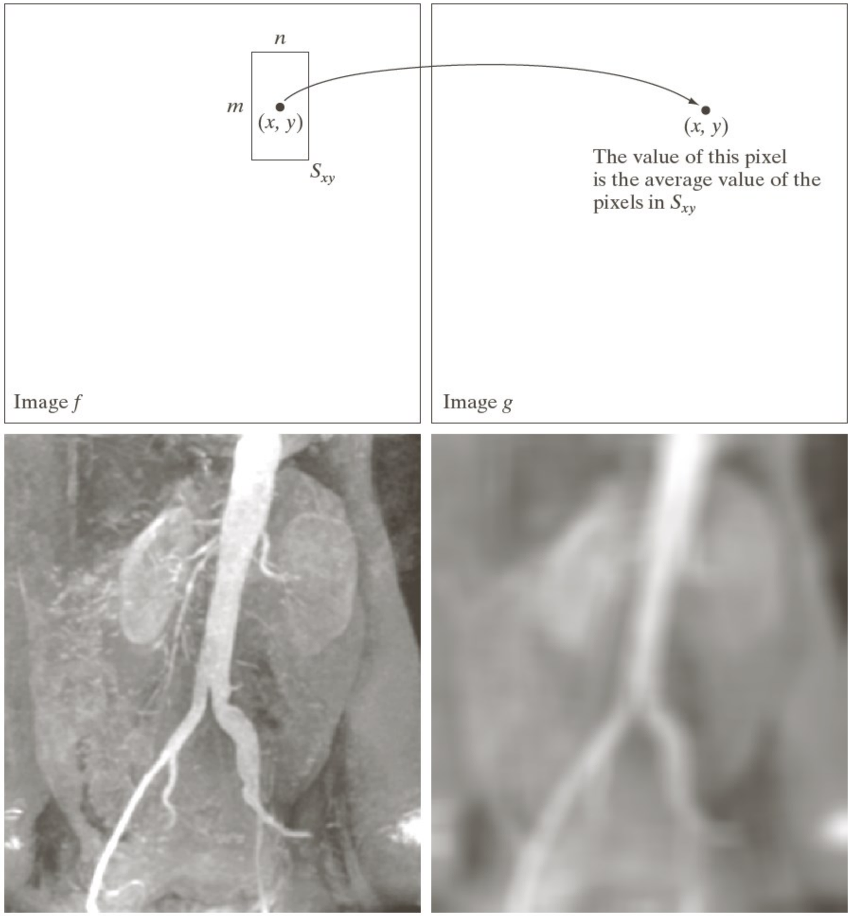


FIGURE (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

Logical operator



Neighbourhood operation



A note on arithmetic operations

- Most images are displayed at 8 bits (0-255).
- When images are saved in standard formats like TIFF or JPEG the conversion to this range is automatic.
- However, the approach used for the conversion depends on the software package.
  - The difference of two images is in the range [-255, 255] and the sum is in the range [0, 510].
  - Many packages simply set all negative values to 0 and all values exceeding 255 to 255 which is undesirable.

A note on arithmetic operations (normalization)

- An approach that guarantees that the full range is captured into a fixed number of bits is the following:
- At first, make the minimum value of the image equal to zero:

$$f_m = f - \min(f)$$

- Then perform intensity scaling to $[0, K]$

$$f_s = \frac{f_m}{\max(f_m)} K$$

Geometric spatial transformations

- A common geometric transformation is the *affine* transform

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- It can be used to express transformations such as rotate, scale and sheer an image depending on the value of the elements of $\mathbf{T}$,
- except translation, which would require that a constant 2-D vector be added to the right side of the equation.

Geometric spatial transformations

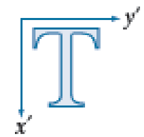
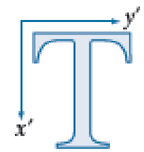
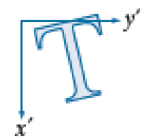
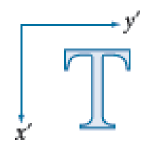
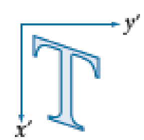
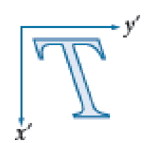
- However, it is possible to use homogeneous coordinates to express all four affine transformations using a single 3 × 3 matrix in the following general form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2\text{-}45)$$

- To avoid empty pixels we implement the inverse mapping
- Interpolation is essential

Geometric spatial transformations (cont.)

TABLE 2.3 Affine transformations based on Eq. (2-45).

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to −1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

Geometric spatial transformations (cont.)

- The effects and importance of interpolation in image transformations

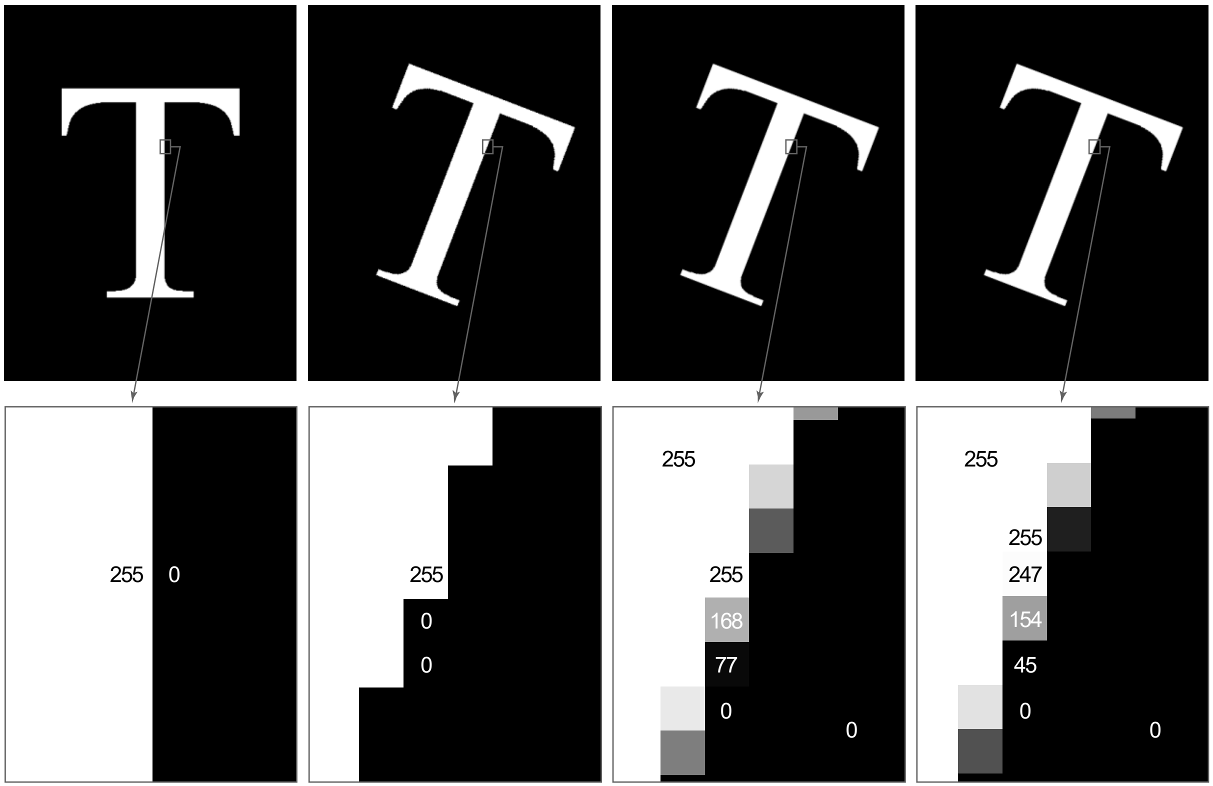


Image Registration

- Estimate the transformation parameters between two images.
- Very important application of digital image processing.
  - Single and multimodal
  - Temporal evolution and quantitative analysis (medicine, satellite images)
- A basic approach is to use control points (user defined or automatically detected) and estimate the elements of the transformation matrix by solving a linear system.

Image Registration (cont.)

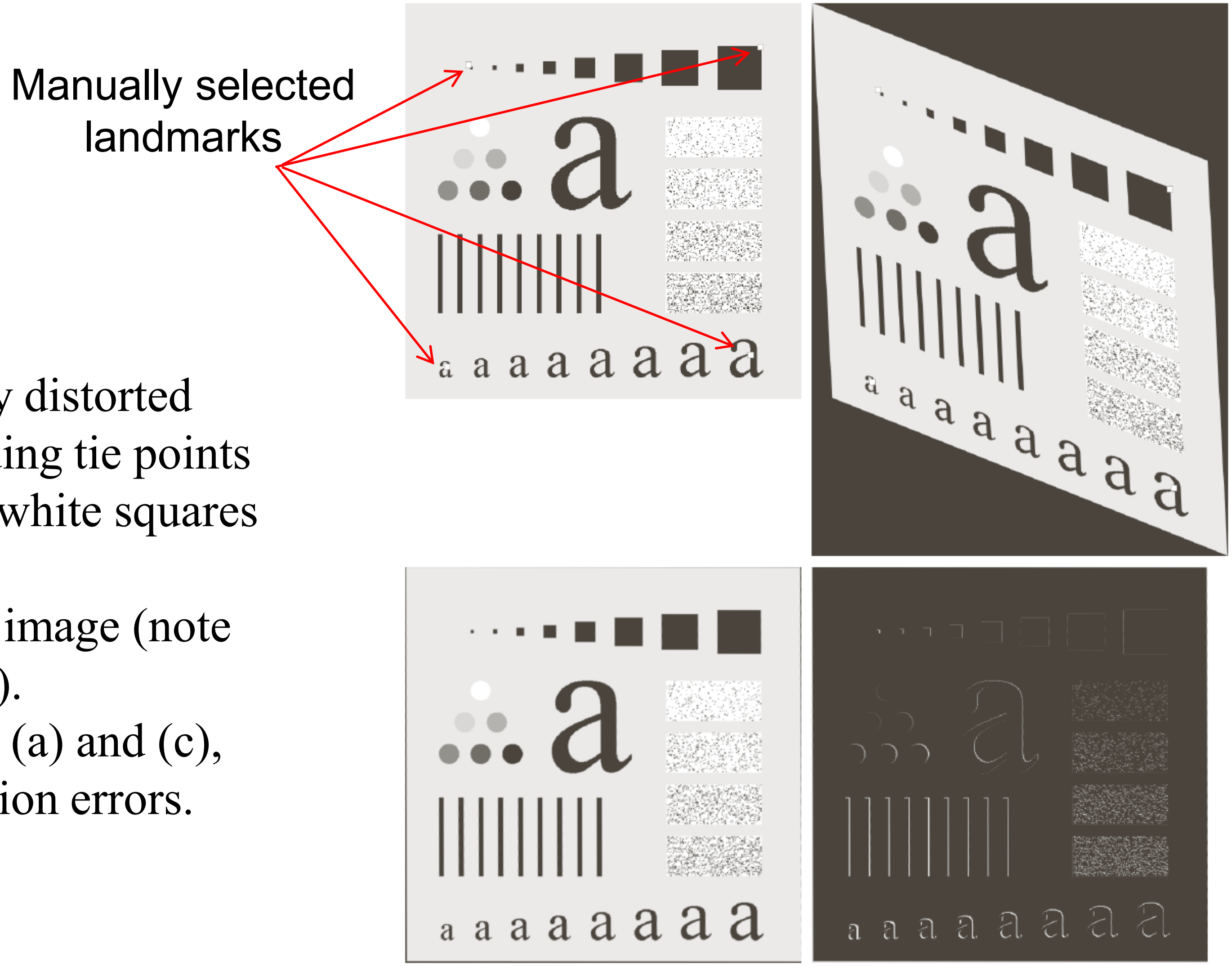


FIGURE 2.42
Image registration.
(a) Reference image.
(b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners.
(c) Registered (output) image (note the errors in the border).
(d) Difference between (a) and (c), showing more registration errors.

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Intensity Transformations (Point Processing)

Άγγελος Γιώτης

a.giotis@uoi.gr

Images taken from:
R. Gonzalez and R. Woods. Digital Image Processing, Prentice Hall, 2008.
Digital Image Processing course by Brian Mac Namee, Dublin Institute of Technology.

Intensity Transformations

"It makes all the difference whether one sees darkness through the light or brightness through the shadows"

David Lindsay
(Scottish Novelist)

Contents

Over the next few lectures we will look at image enhancement techniques working in the spatial domain:

- What is image enhancement?
- Different kinds of image enhancement
- Point processing
- Histogram processing
- Spatial filtering

What Is Image Enhancement?

Image enhancement is the process of making images more useful

The reasons for doing this include:

- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Image Enhancement Examples

Image Enhancement Examples (cont…)

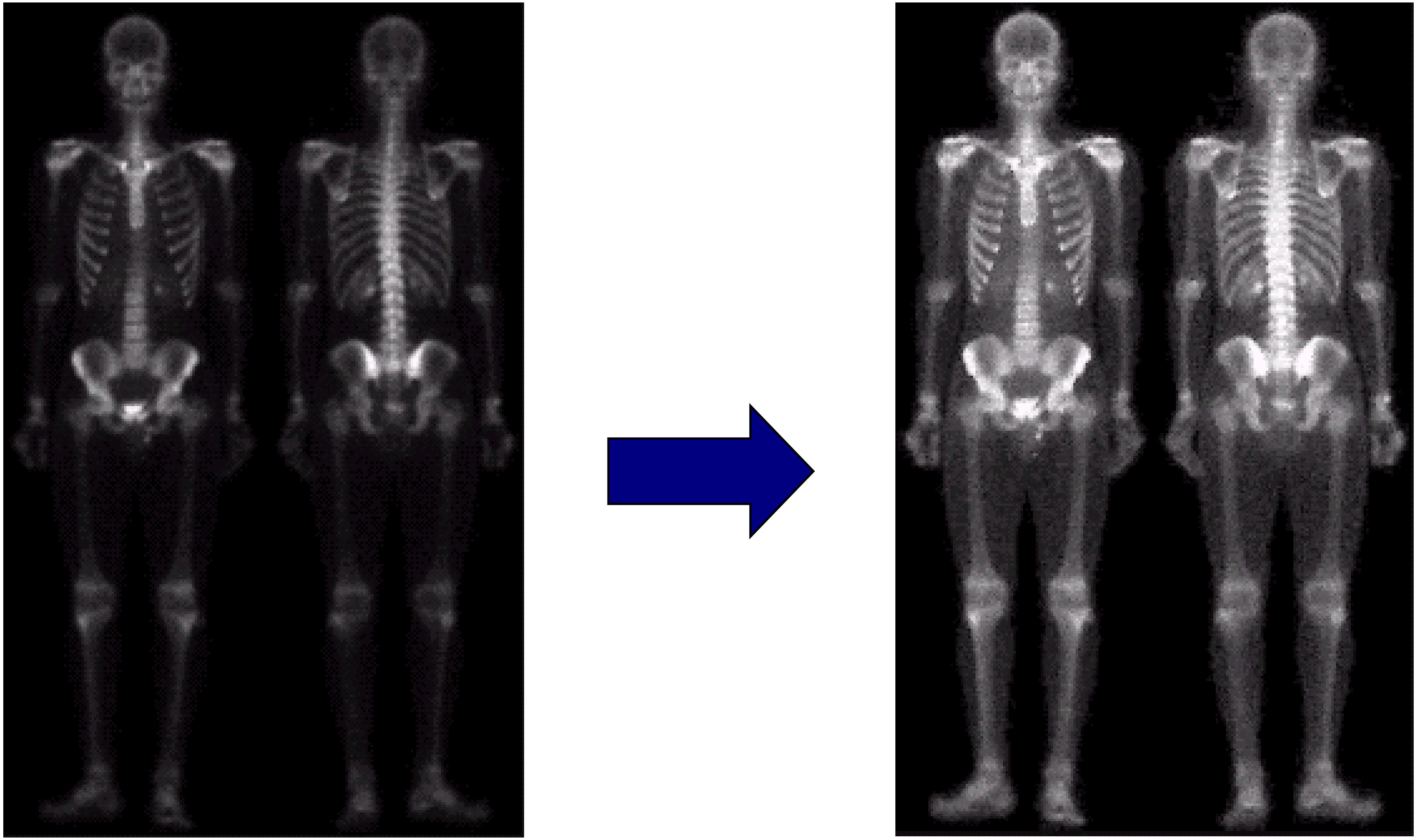


Image Enhancement Examples (cont…)

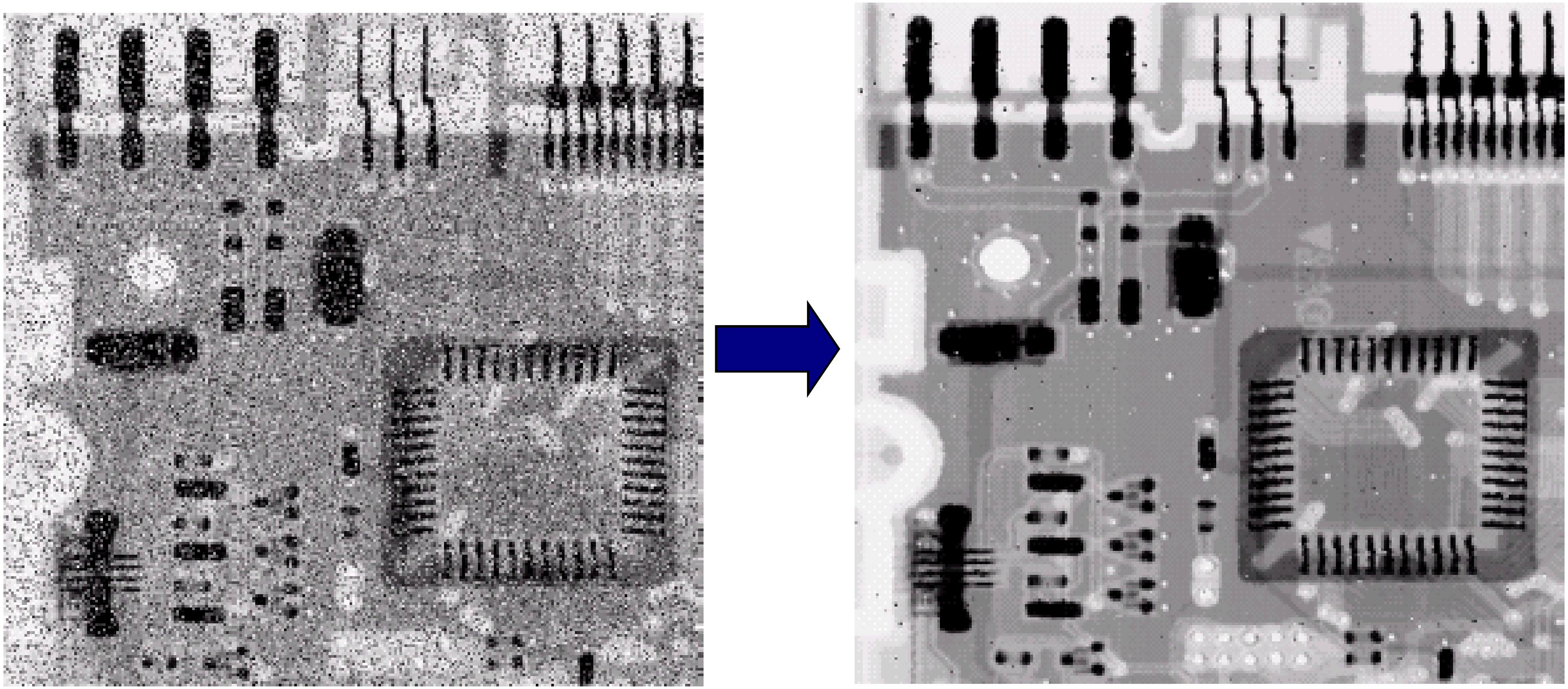


Image Enhancement Examples (cont…)

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

- **Spatial domain techniques**
  - Direct manipulation of image pixels
- Frequency domain techniques
  - Manipulation of Fourier transform or wavelet transform of an image

For the moment we will concentrate on techniques that operate in the spatial domain

Contents

In this lecture we will look at image enhancement point processing techniques:

- What is point processing?
- Negative images (highlighting)
- Thresholding (e.g., binary image)
- Logarithmic transformation (low intensity)
- Power law transforms (brightness/contrast)
- Grey level slicing (linear, log, power)
- Bit plane slicing (compression)

A Note About Grey Levels

So far when we have spoken about image grey level values we have said they are in the range [0, 255]

- Where 0 is black and 255 is white

There is no reason why we have to use this range

- The range [0,255] stems from display technologies

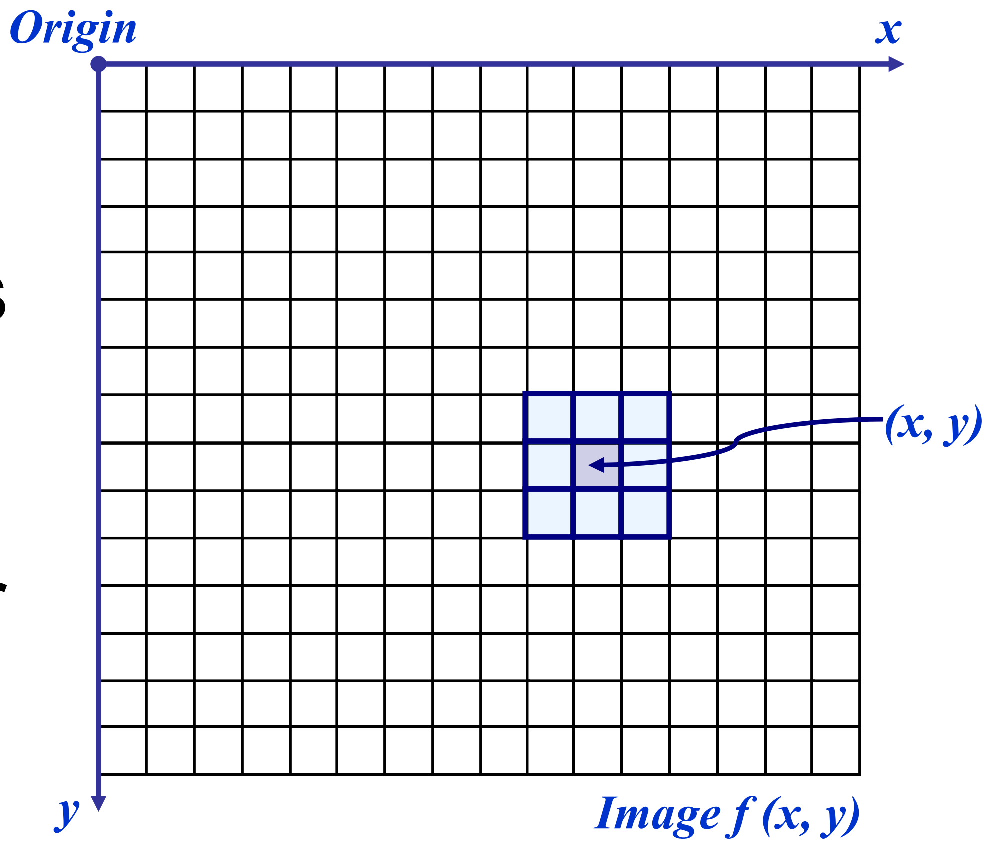
For many of the image processing operations in this lecture grey levels are assumed to be given in the range [0.0, 1.0]

Basic Spatial Domain Image Enhancement

Most spatial domain enhancement operations can be reduced to the form

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

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of (x, y)



Point Processing

The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself

In this case T is referred to as a *grey level transformation function* or a *point processing operation*

Point processing operations take the form

$$s = T(r)$$

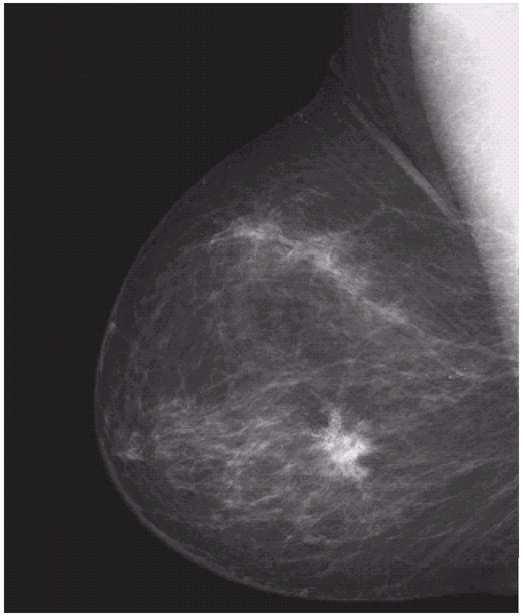
where s refers to the processed image pixel value and r refers to the original image pixel value.

Point Processing Example: Negative Images

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

- Note how much clearer the tissue is in the negative image of the mammogram below

Original Image



$s = 1.0 - r$

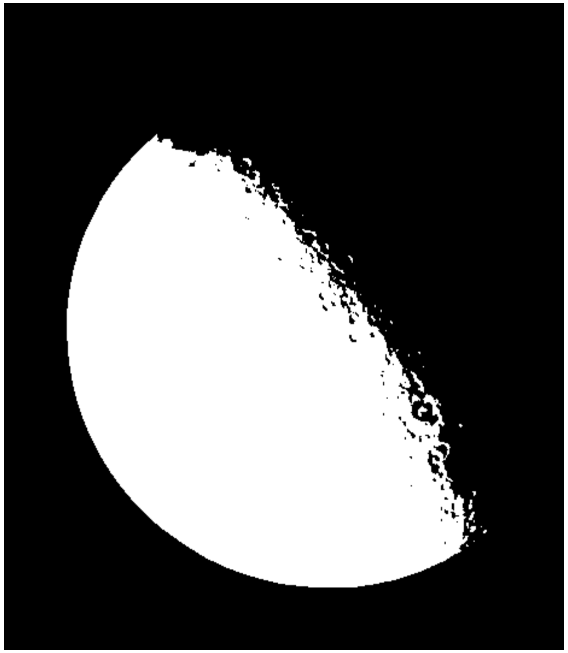
Negative Image

(MYE037)

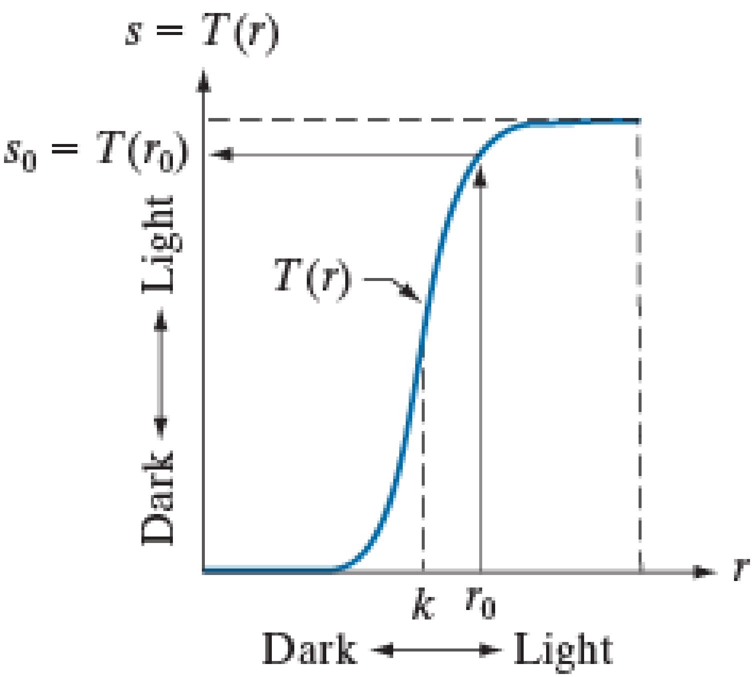
Point Processing Example: Thresholding

Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

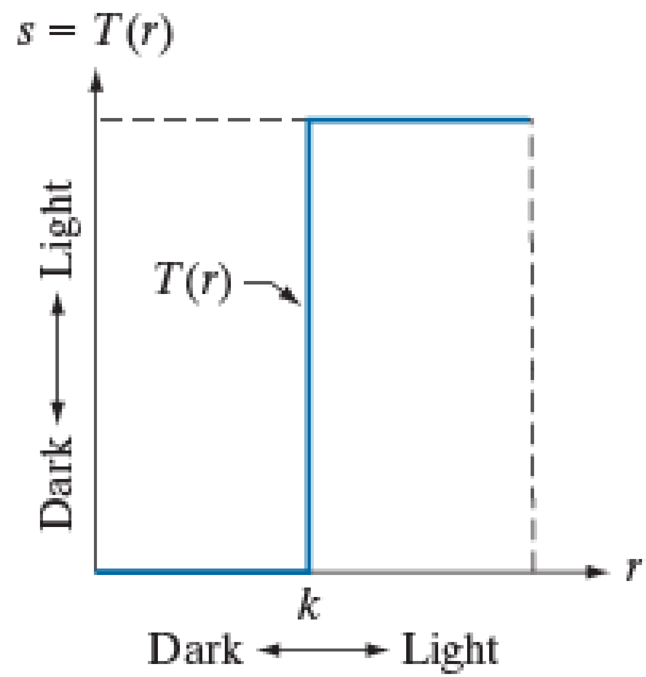
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$



Intensity Transformations



Contrast stretching



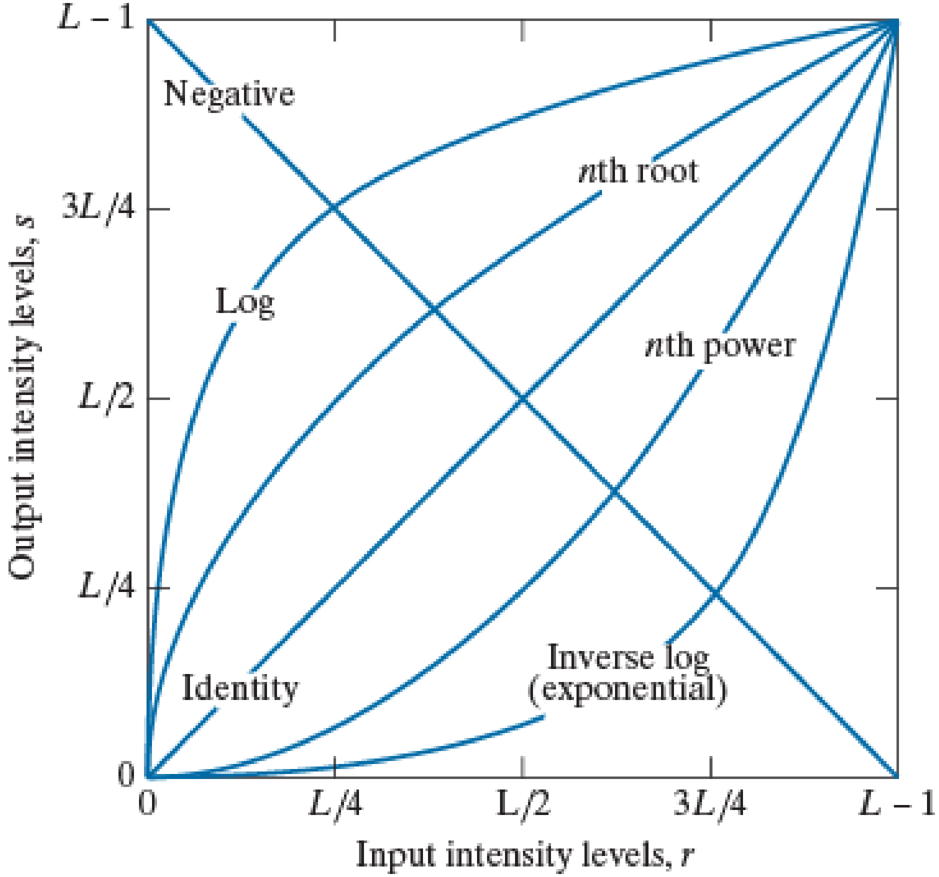
Thresholding

Basic Grey Level Transformations

There are many different kinds of grey level transformations

Three of the most common are shown here

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



Logarithmic Transformations

The general form of the log transformation is

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

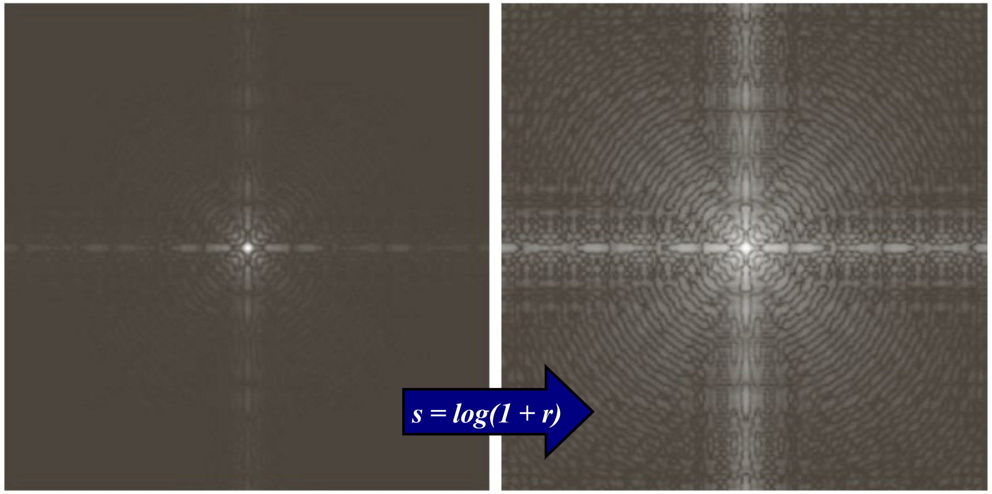
The log transformation maps a narrow range of low input grey level values into a wider range of output values

The inverse log transformation performs the opposite transformation

Logarithmic Transformations (cont…)

Log functions are particularly useful when the input grey level values may have an extremely large range of values

In the following example the Fourier transform of an image is put through a log transform to reveal more detail



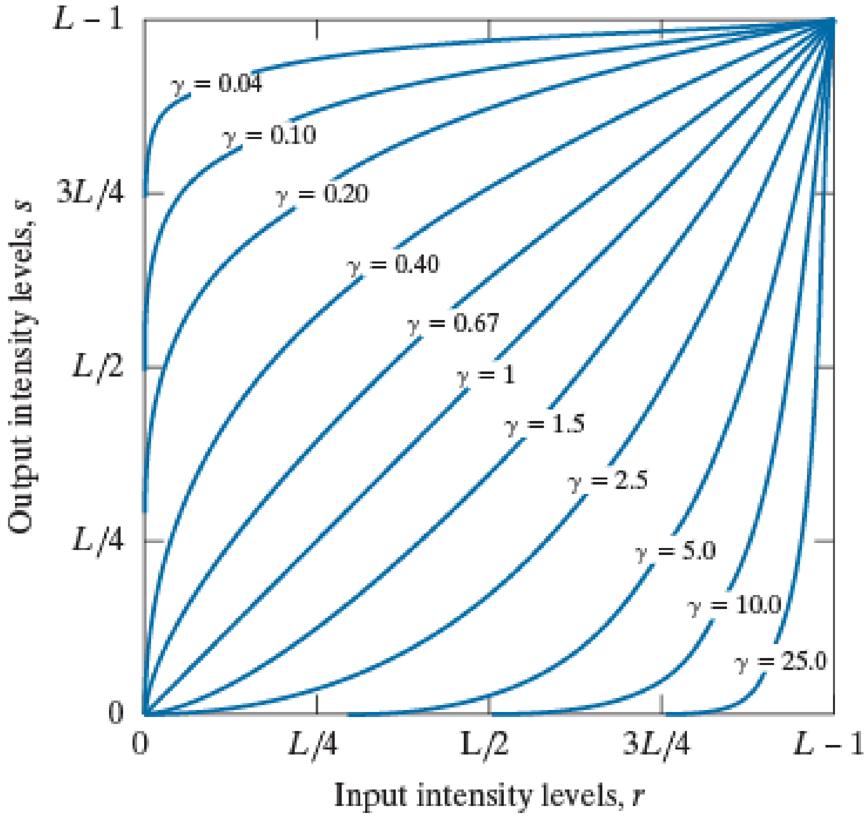
Power Law Transformations

Power law transformations have the following form

$$s = c * r^{\gamma}$$

Map a narrow range of dark input values into a wider range of output values or vice versa

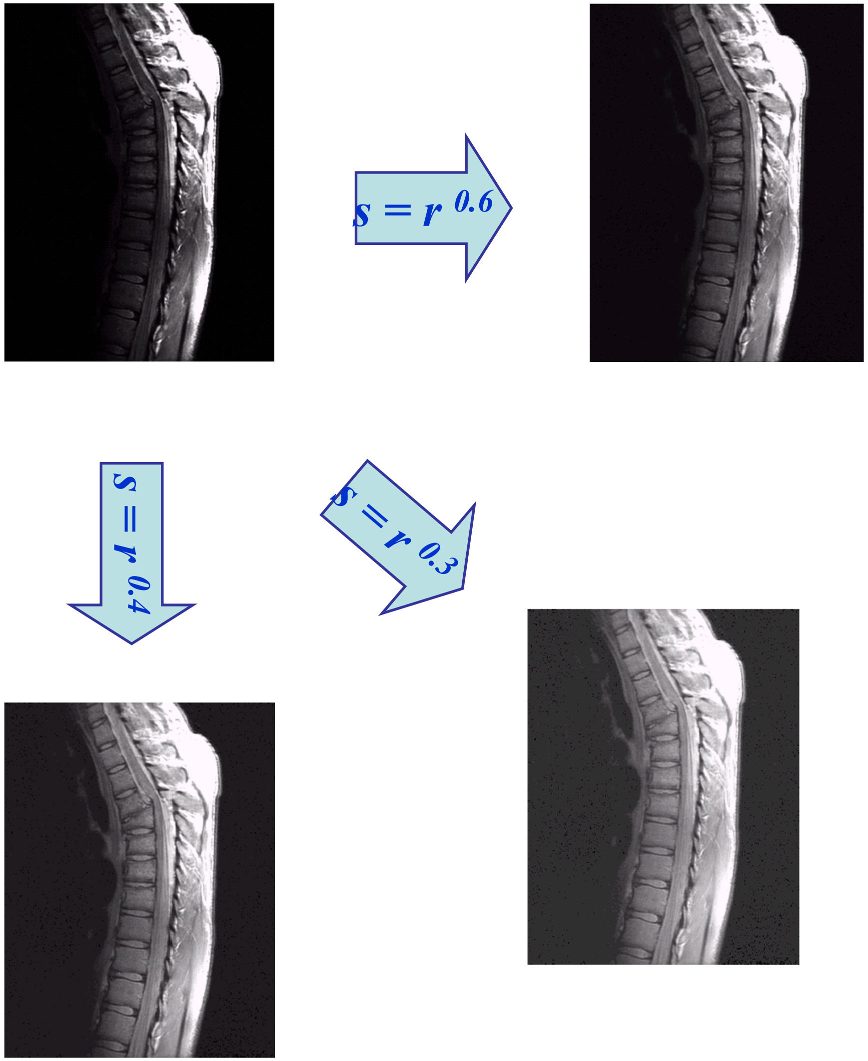
Varying γ gives a whole family of curves



Power Law Example (cont…)

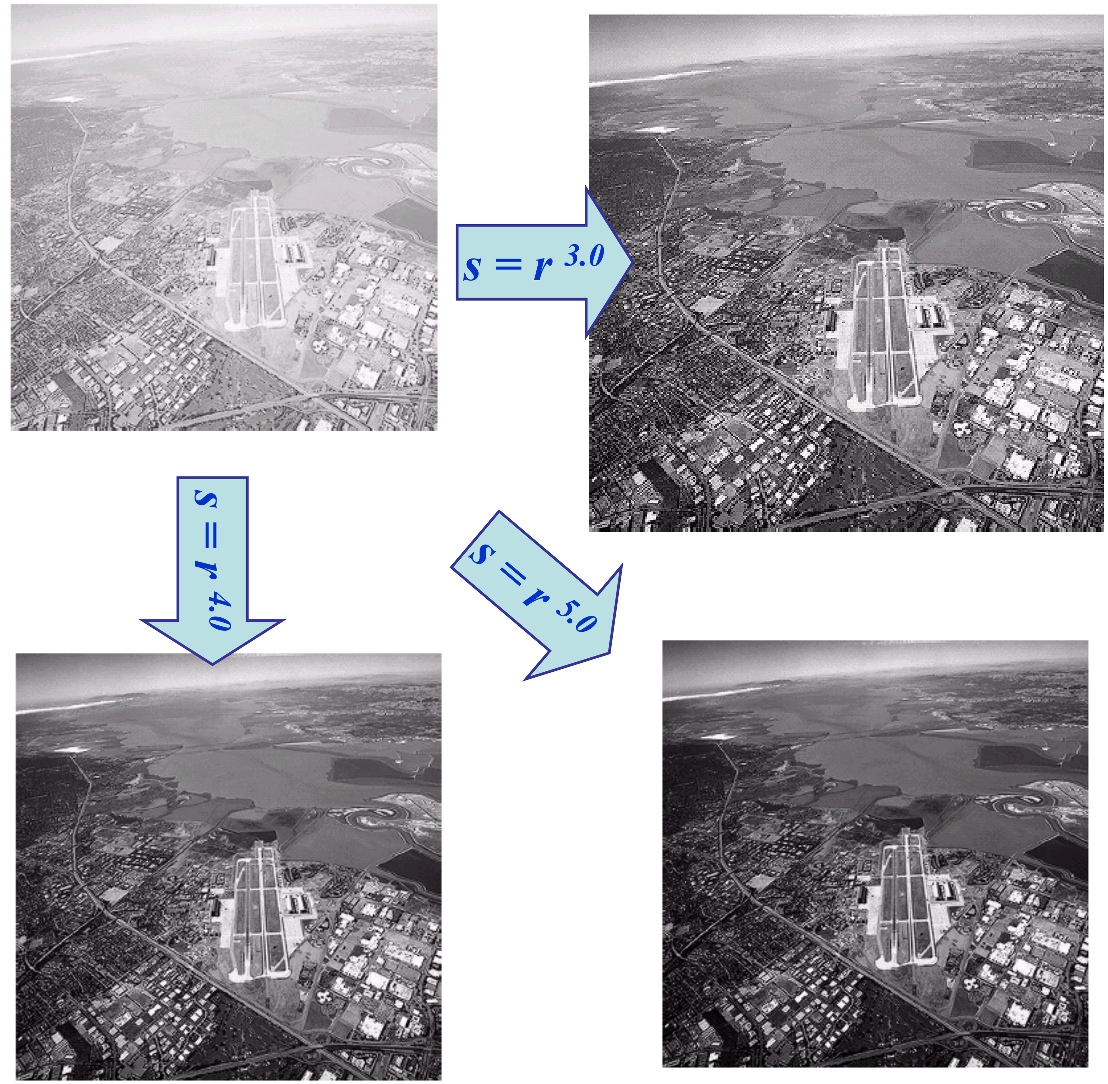
The images to the right show a magnetic resonance (MR) image of a fractured human spine

Different curves highlight different detail



Power Law Transformations (cont…)

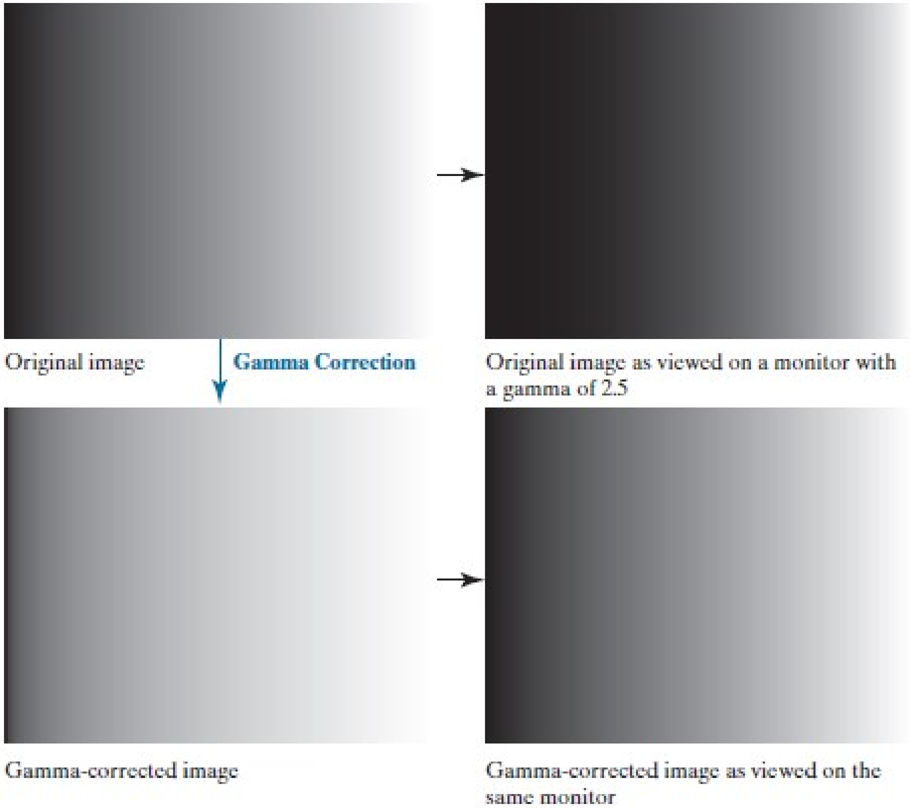
- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail



Gamma Correction

Many of you might be familiar with gamma correction of computer monitors

Problem is that display devices do not respond linearly to different intensities

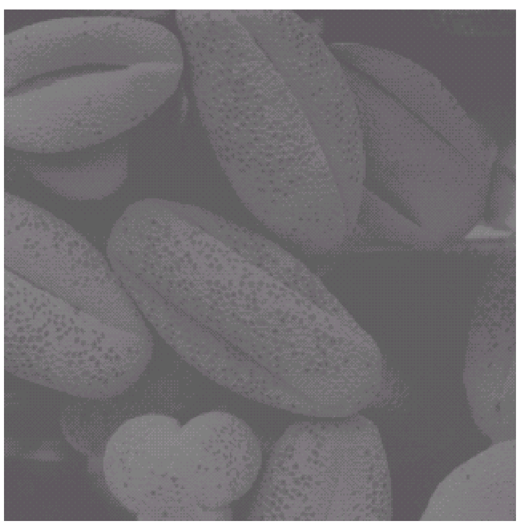


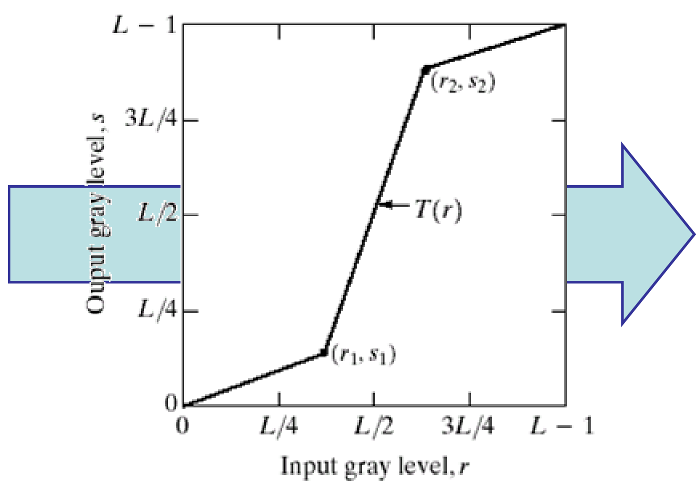
Can be corrected using a log transform

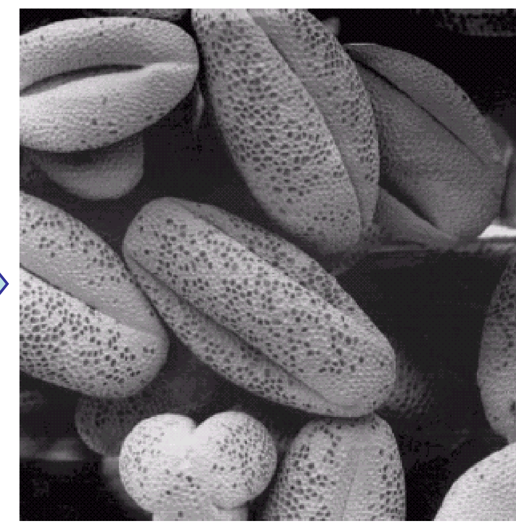
Piecewise Linear Transformation Functions

Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

The images below show a contrast stretching linear transform to add contrast to a poor quality image

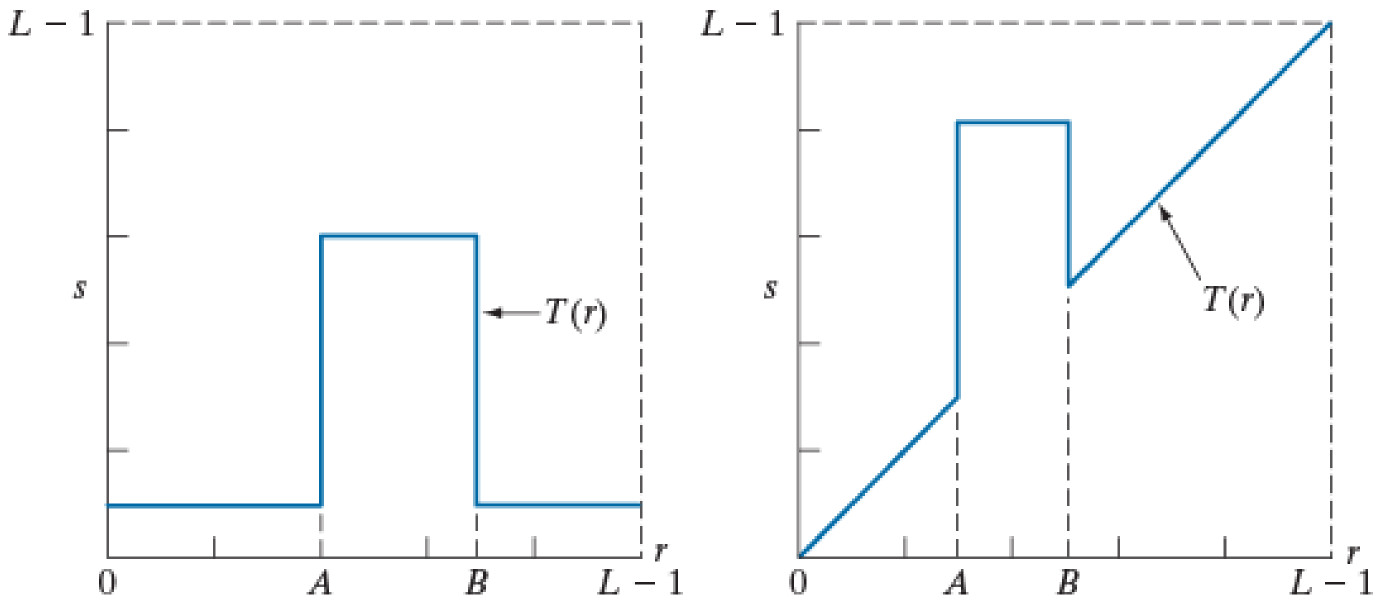




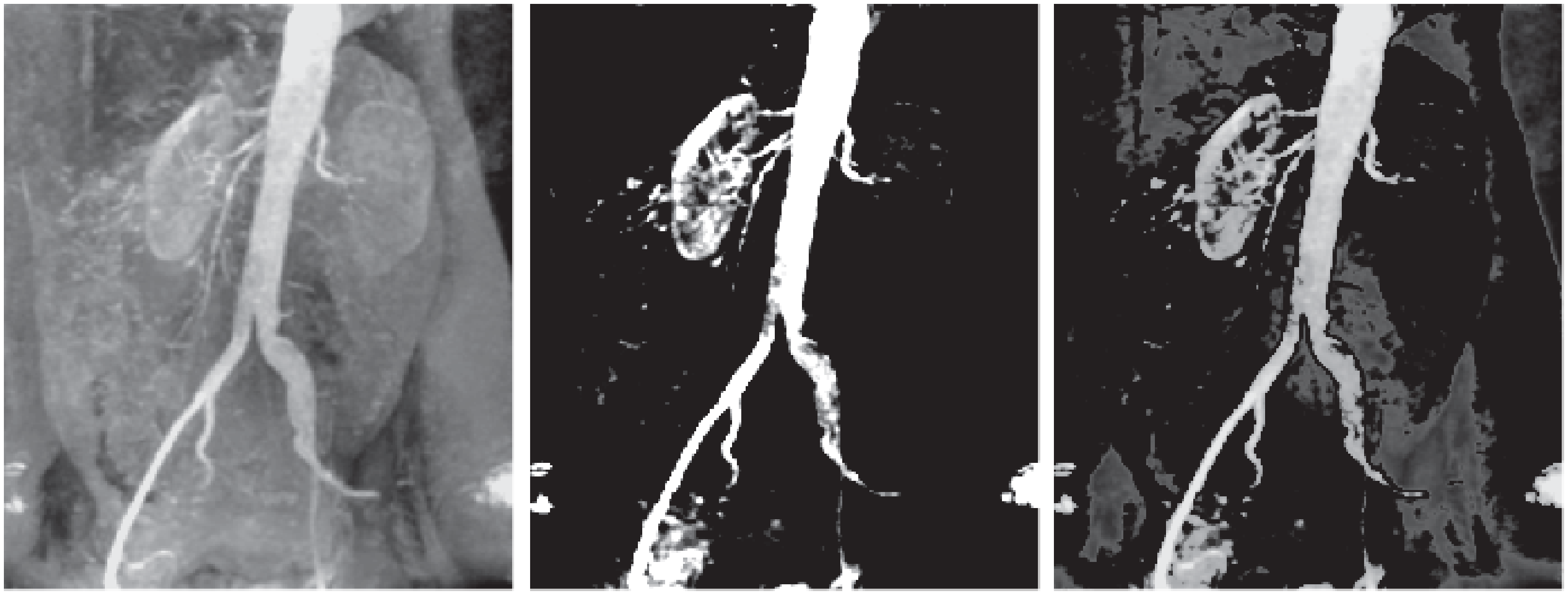


Piecewise-Linear Transformation (cont…)

(a) This transformation function highlights range [A,B] and reduces all other intensities to a lower level.
(b) This function highlights range [A,B] and leaves other intensities unchanged.



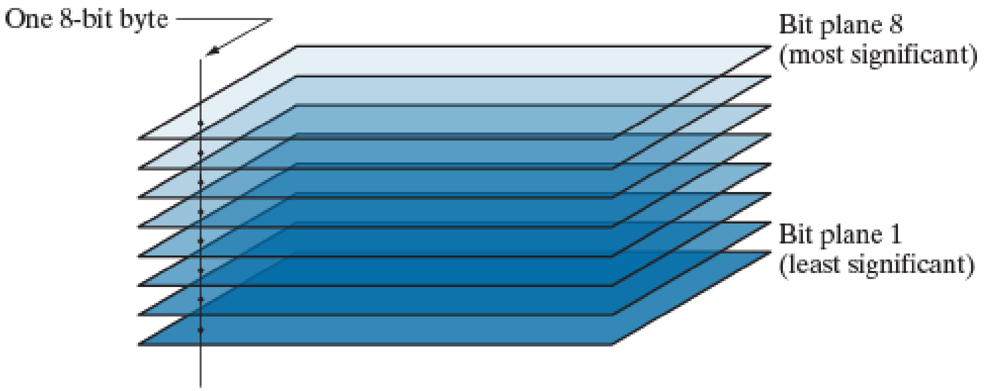
Piecewise-Linear Transformation (cont…)



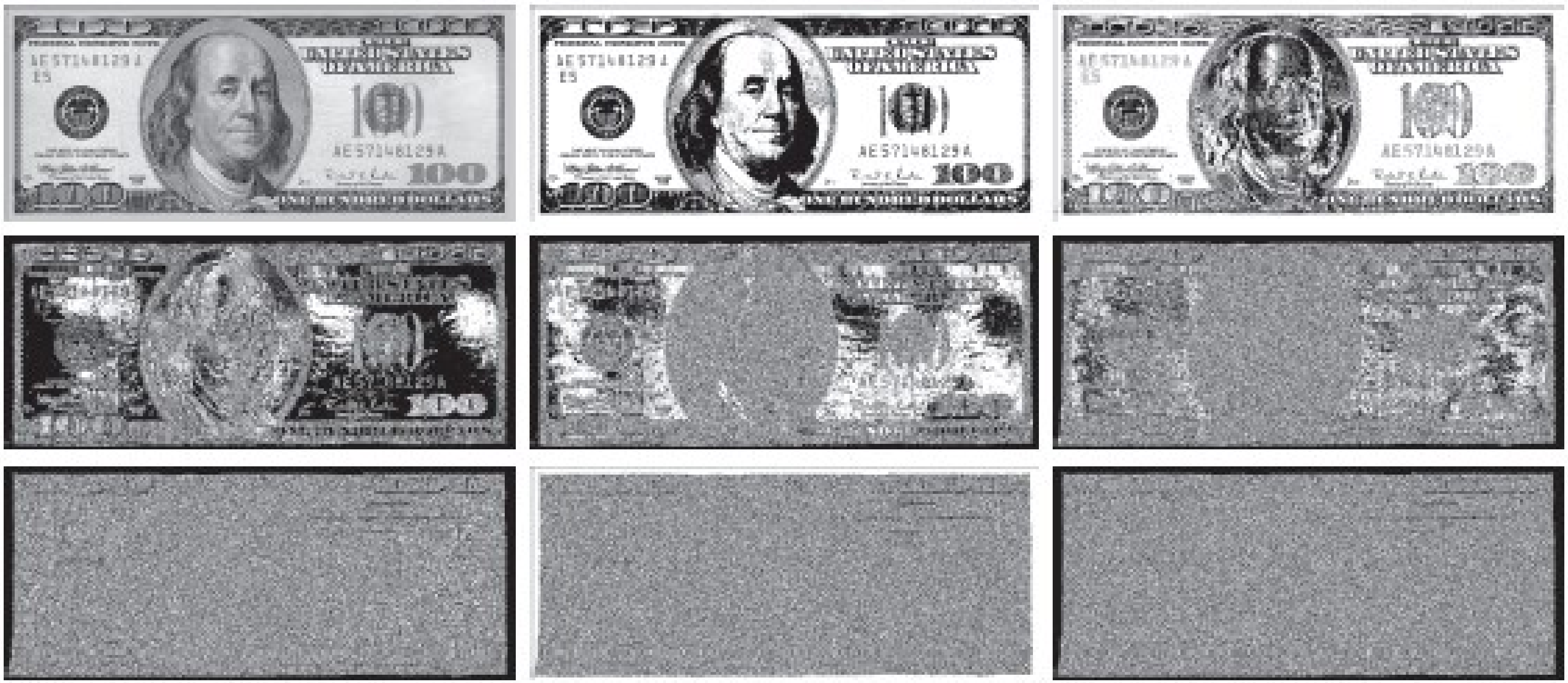
Bit Plane Slicing

Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details



Bit-Plane Slicing (cont…)



Bit-Plane Slicing (cont…)

a b c **FIGURE 3.15** Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5.

Useful for compression.

Reconstruction is obtained by:

$$I(i,j) = \sum_{n=1}^{N} 2^{n-1} I_n(i,j)$$

Average image

Let $g(x,y)$ denote a corrupted image by adding noise $\eta(x,y)$ to a noiseless image $f(x,y)$:

$$g(x, y) = f(x, y) + \eta(x, y)$$

The noise has zero mean value $E[z_i] = 0$

At every pair of coordinates $z_i=(x_i,y_i)$ the noise is uncorrelated

$$E[z_i z_j] = 0, \; E[z_i^2] = \sigma_\eta^2$$

Average image (cont...)

The noise effect is reduced by averaging a set of K noisy images. The new image is

$$\overline{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

The intensities at each pixel of the new image may be viewed as random variables.

The mean value and the standard deviation of the new image show that the effect of noise is reduced.

Average image (cont...)

$$E\left[\bar{g}(x,y)\right] = E\left[\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)\right] = \frac{1}{K}E\left[\sum_{i=1}^{K} g_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y) + \eta_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y)\right] + \frac{1}{K}E\left[\sum_{i=1}^{K} \eta_i(x,y)\right]$$

$$= \frac{1}{K}Kf(x,y) + \frac{1}{K}K0 \;= f(x,y)$$

Average image (cont...)

Similarly, the standard deviation of the new image is

$$\sigma_{\bar{g}(x,y)} = E\left[\left(\bar{g}(x,y)\right)^2\right] - \left(E\left[\bar{g}(x,y)\right]\right)^2 = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$

As K increases the variability of the pixel intensity decreases and remains close to the noiseless image values $f(x,y)$.

The images must be registered!

Average image (cont...)

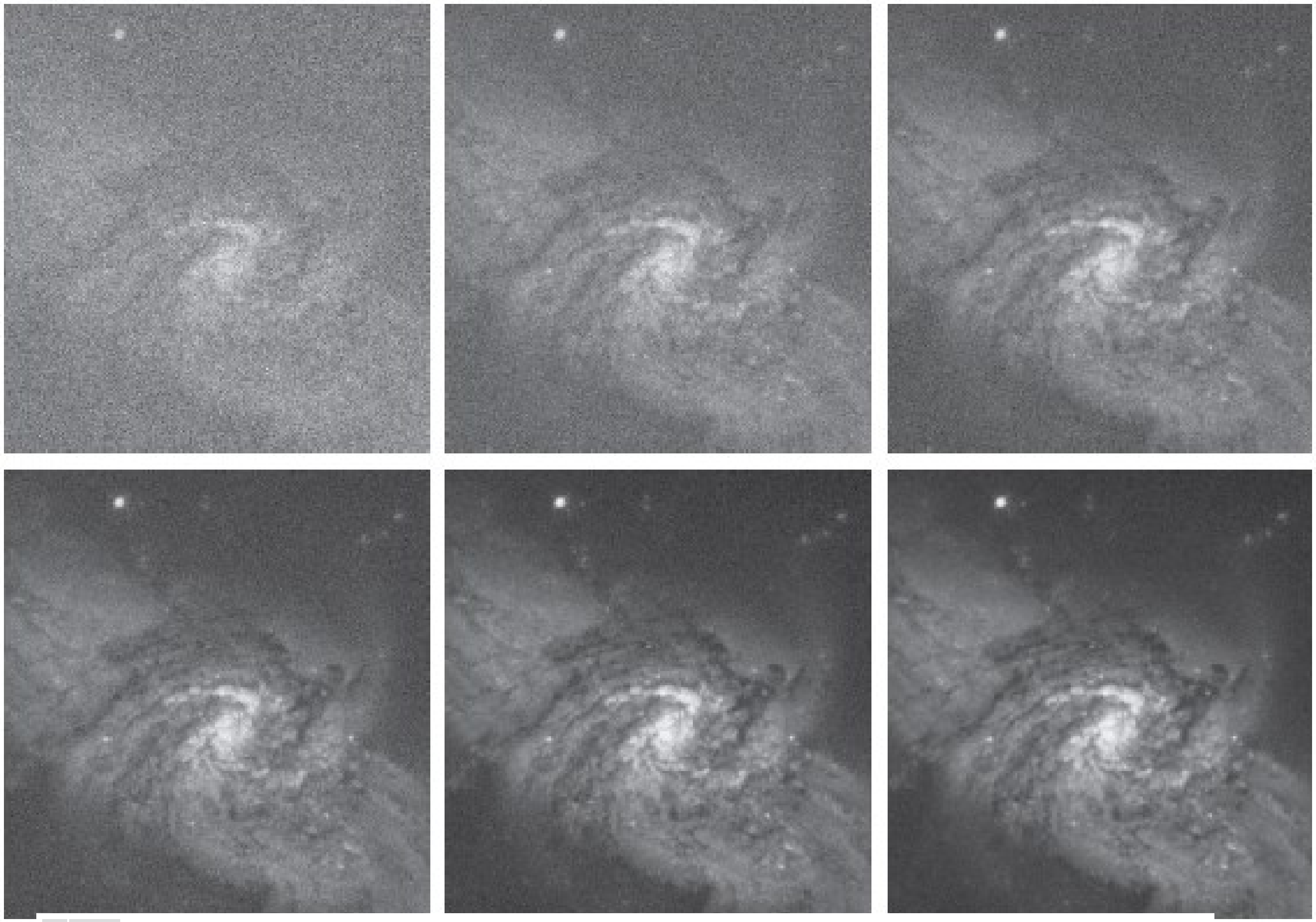


FIGURE 2.29 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)-(f) Result of averaging 5, 10, 20, 50, and 1,00 noisy images, respectively. All images are of size 566×598 pixels, and all were scaled so that their intensities would span the full [0, 255] intensity scale. (Original image courtesy of NASA.)

Summary

We have looked at different kinds of point processing image enhancement

Next time we will start to look at histogram processing methods.

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Intensity Transformations (Histogram Processing)

Άγγελος Γιώτης

a.giotis@uoi.gr

Contents

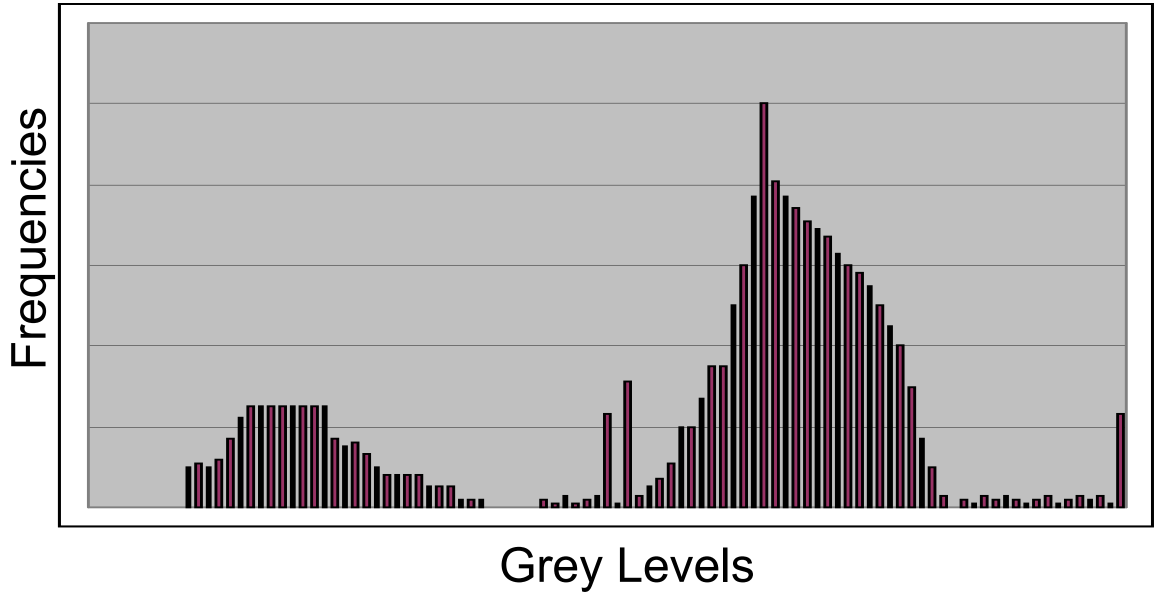
Over the next few lectures we will look at image enhancement techniques working in the spatial domain:

- Histogram processing
- Spatial filtering
- Neighbourhood operations

Image Histograms

The histogram of an image shows us the distribution of grey levels in the image

Massively useful in image processing, especially in segmentation



Histogram processing

- Let r_k, for $k = 0,1,2,\ldots,L-1$, denote the intensities of an *L*-level digital image, $f(x,y)$.

- The *unnormalized histogram* of f is defined as:

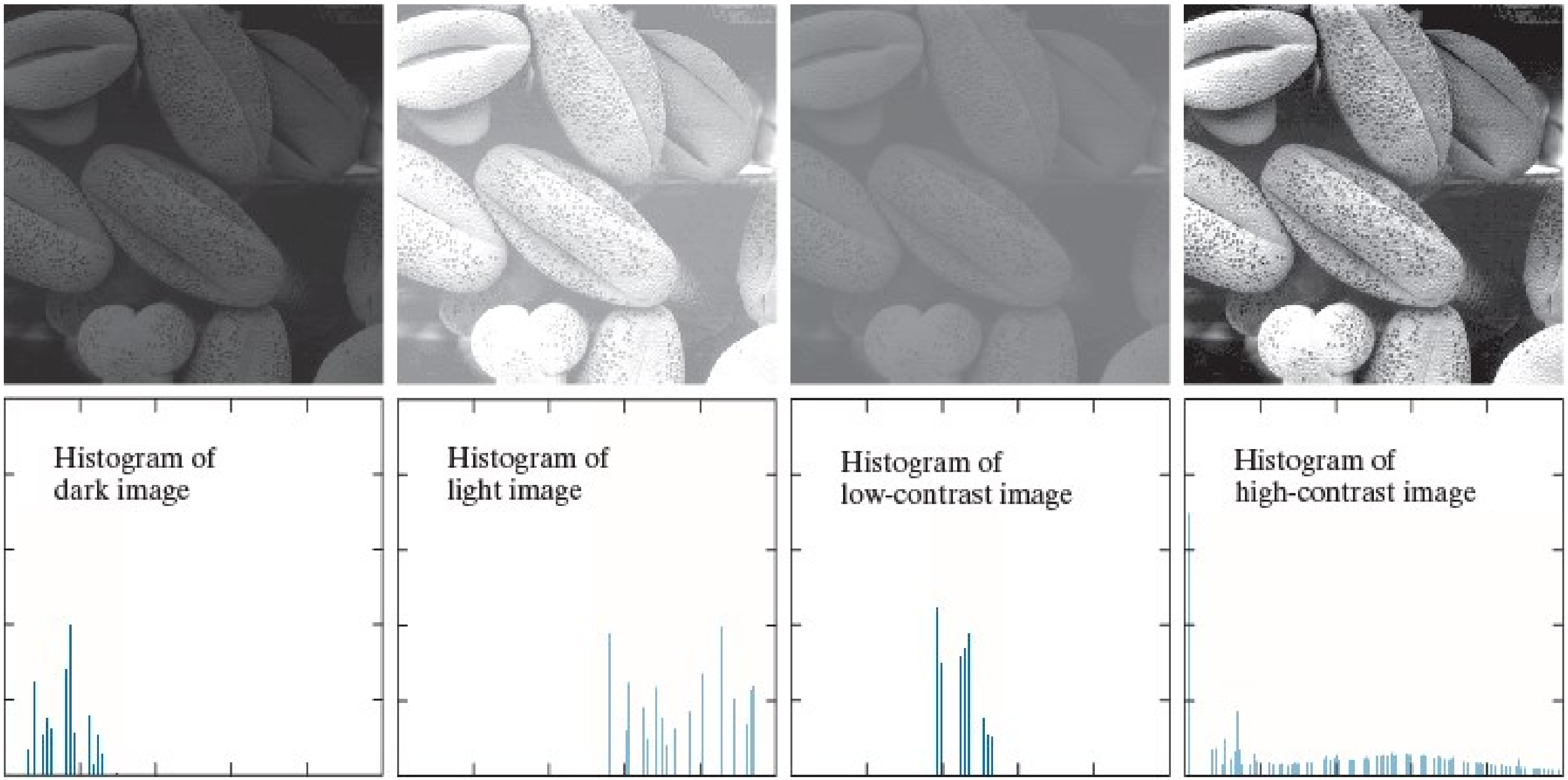
$$h(r_k) = n_k \quad \text{for } k = 0,1,2,\ldots,L-1$$

Where n_k is the number of pixels in f with intensity r_k, and the subdivisions of the intensity scale are called *histogram bins.*

- *Normalized histogram of f:*

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

Histogram Examples



Contrast Stretching

- We can fix images that have poor contrast by applying a pretty simple contrast specification
- The interesting part is how do we decide on this transformation function?

Histogram Equalisation

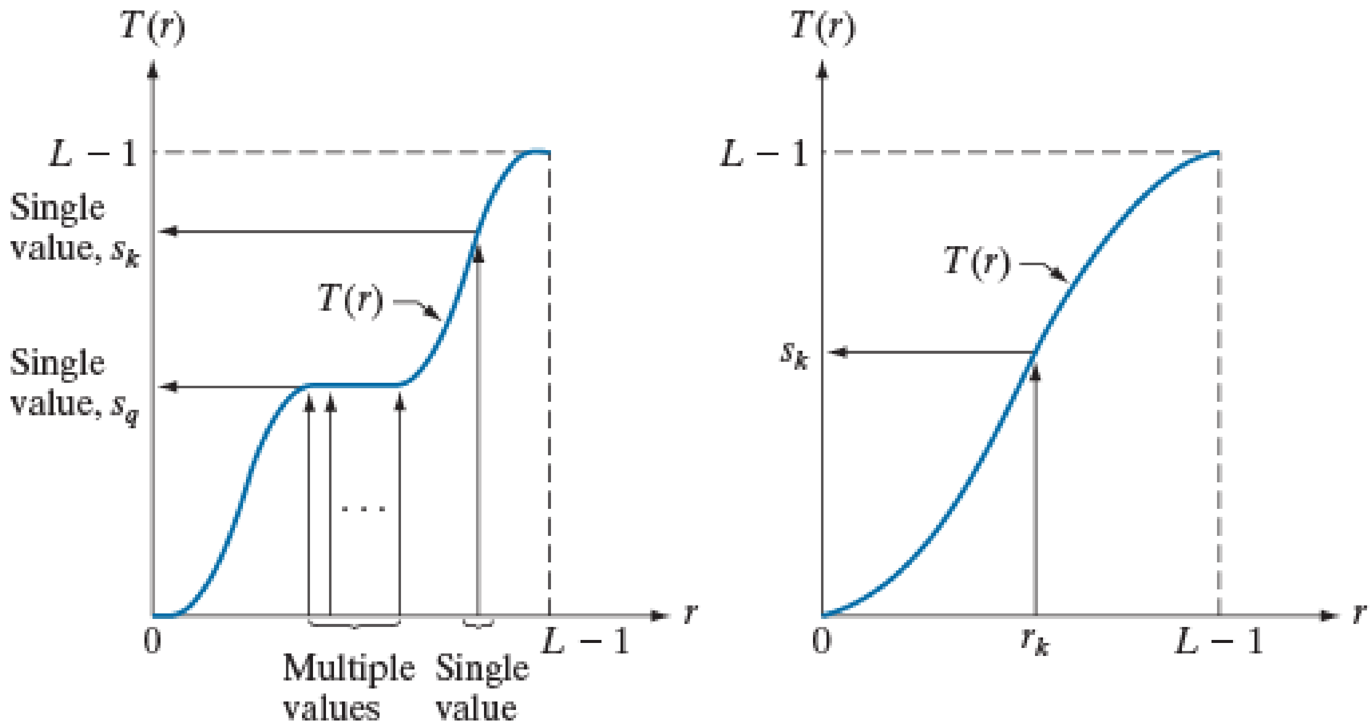
- Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images.
- At first, the continuous case will be studied:
  - r is the intensity of the image in $[0, L\text{-}1]$.
  - we focus on transformations $s=T(r)$:
    - $T(r)$ is monotonically increasing.
    - $T(r)$ must satisfy:

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

Histogram Equalisation (cont...)

- The condition for $T(r)$ to be monotonically increasing guarantees that ordering of the output intensity values will follow the ordering of the input intensity values (avoids reversal of intensities).
- If $T(r)$ is **strictly** monotonically increasing then the mapping from s back to r will be 1-1.
- The second condition ($T(r)$ in $[0,1]$) guarantees that the range of the output will be the same as the range of the input.

Histogram Equalisation (cont...)



a) We cannot perform inverse mapping (from s to r).

b) Inverse mapping is possible.

Histogram Equalisation (cont...)

- We can view intensities r and s as random variables and their histograms as probability density functions (pdf) $p_r(r)$ and $p_s(s)$.
- Fundamental result from probability theory:
  - If $p_r(r)$ and $T(r)$ are known and $s=T(r)$ is continuous and differentiable, then

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

Histogram Equalisation (cont...)

- The pdf of the output is determined by the pdf of the input and the transformation.
- This means that we can determine the histogram of the output image.
- A transformation of particular importance in image processing is the cumulative distribution function (CDF) of a random variable:

$$s = T(r) = (L-1)\int_{0}^{r} p_r(w)\,dw$$

Histogram Equalisation (cont...)

- It satisfies the first condition as the area under the curve increases as r increases.
- It satisfies the second condition as for $r=L-1$ we have $s=L-1$.
- To find $p_s(s)$ we have to compute

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\int_0^r p_r(w)\,dw = (L-1)\,p_r(r)$$

Histogram Equalisation (cont...)

Substituting this result:

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

to

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

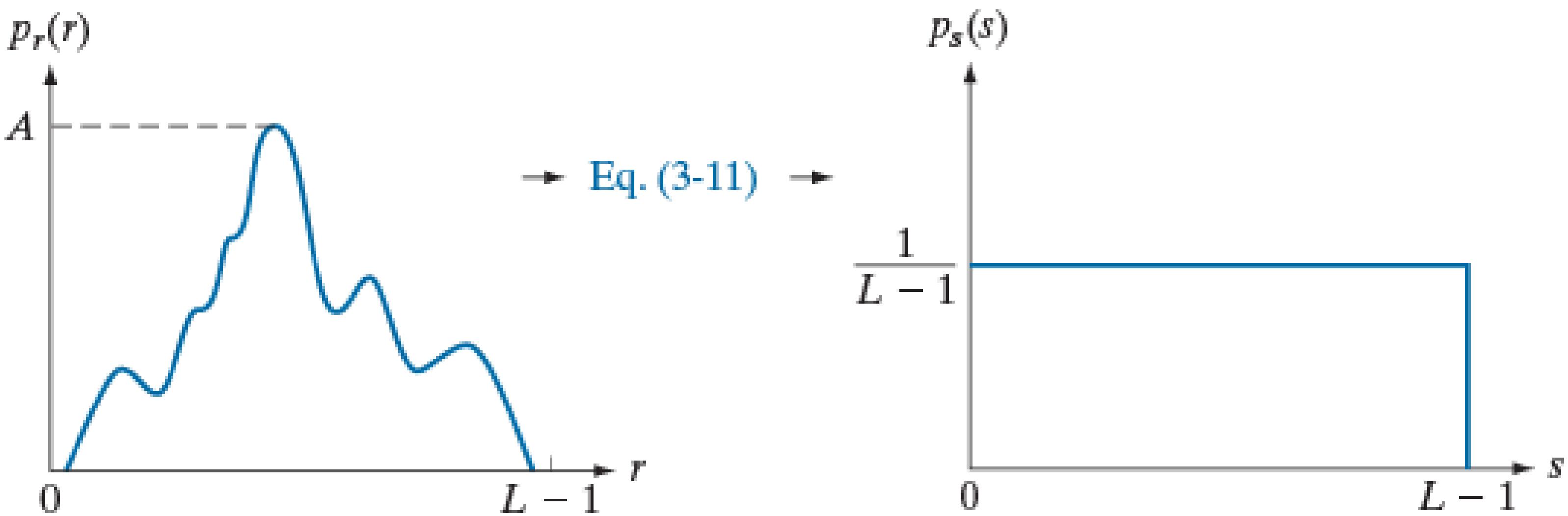
Uniform pdf

yields

$$p_s(s) = p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right| = \frac{1}{L-1}, \quad 0 \le s \le L-1$$

Histogram Equalisation (cont...)

- A continuous histogram will always result in a uniform histogram



Histogram Equalisation (cont...)

The formula for histogram equalisation in the discrete case is given

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

where

- r_k: input intensity
- s_k: processed intensity
- n_j: the frequency of intensity j
- MN: the number of image pixels.

Histogram Equalisation (cont…) Example

A 3-bit 64x64 image has the following intensities:

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

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

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

Applying histogram equalization:

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

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

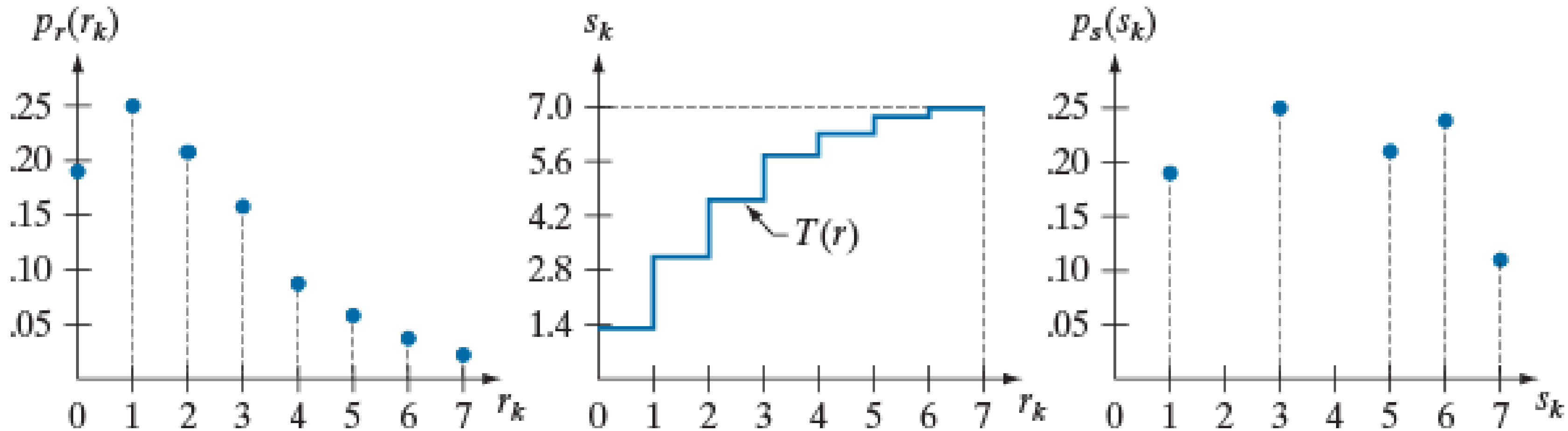
$s_2 = ??$

Histogram Equalisation (cont…) Example

Rounding to the nearest integer:

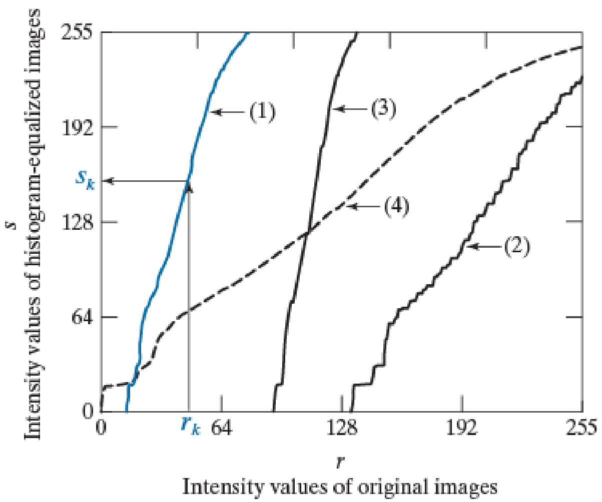
$$s_0 = 1.33 \rightarrow 1 \quad s_1 = 3.08 \rightarrow 3 \quad s_2 = 4.55 \rightarrow 5 \quad s_3 = 5.67 \rightarrow 6$$

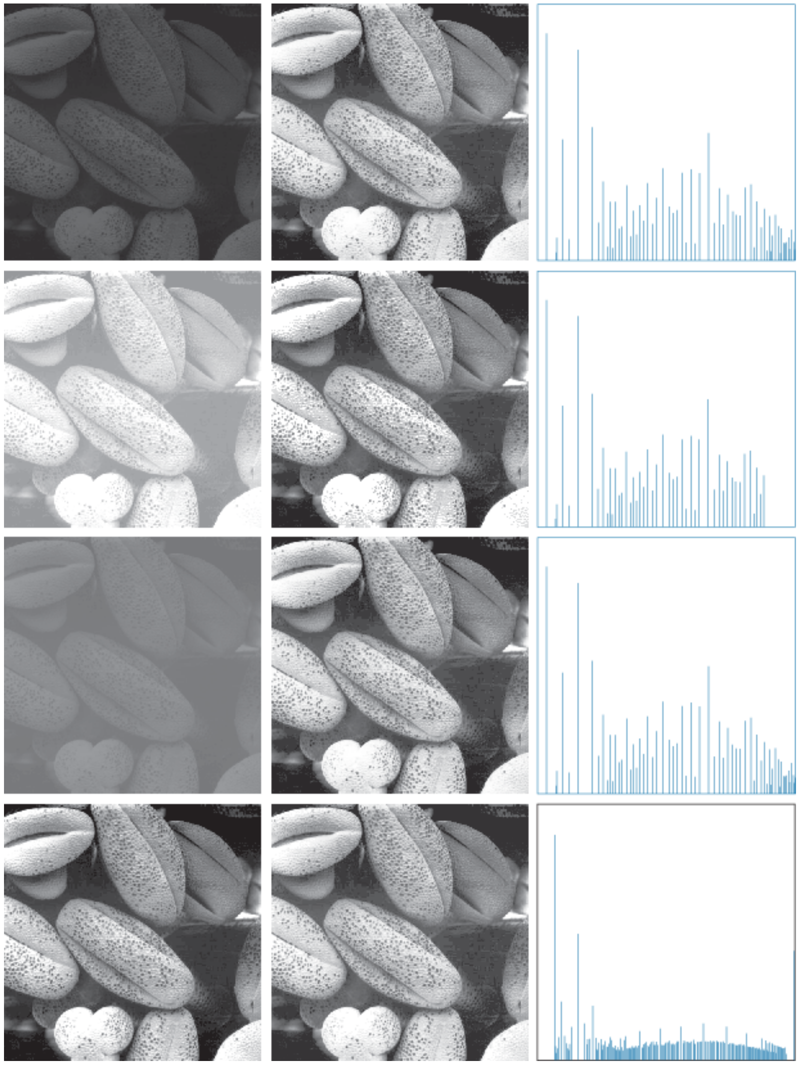
$$s_4 = 6.23 \rightarrow 6 \quad s_5 = 6.65 \rightarrow 7 \quad s_6 = 6.86 \rightarrow 7 \quad s_7 = 7.00 \rightarrow 7$$



Due to discretization, the resulting histogram, though extended, will rarely be perfectly flat.

Histogram equalization example



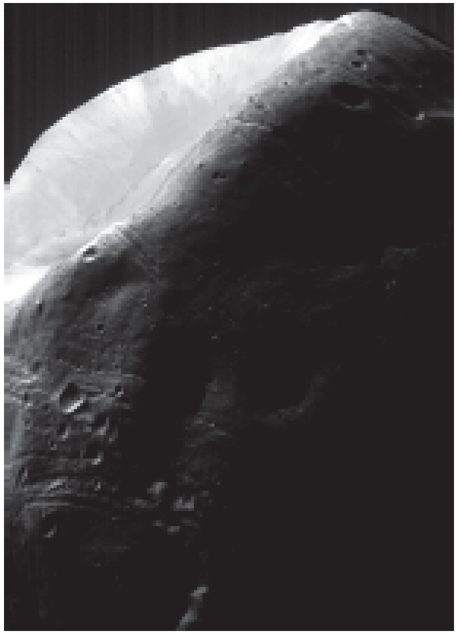


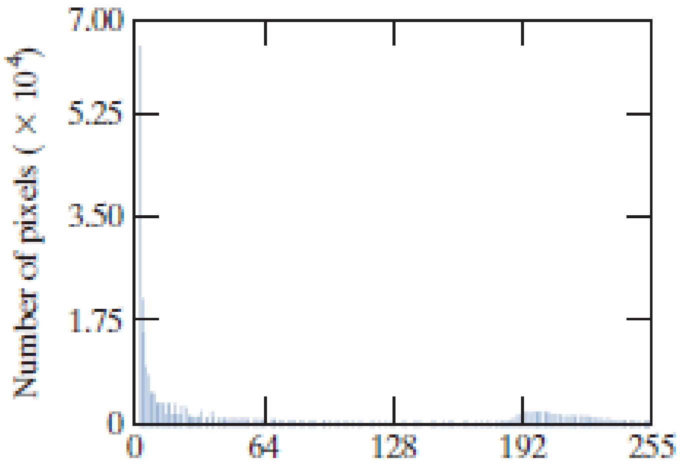
Histogram Matching (Specification)

- Histogram equalization - transformation aims to create an output image with a uniform histogram.
- Beneficial for automatic enhancement - predictable and straightforward implementation.
- However, there are scenarios where histogram equalization may not be suitable.
- Sometimes, it's necessary to **specify** the shape of the histogram for the processed image.
- The technique used to generate images with a specified histogram is known as **histogram matching** or **histogram specification**.

Histogram Specification

- Histogram equalization does not always provide the desirable results.





- Image of Phobos (Mars moon) and its histogram.
- Many values near zero in the initial histogram

Histogram specification (cont.)

- In these cases, it is more useful to specify the final histogram.
- Problem statement:
  - Given $p_r(r)$ from the image and the target histogram $p_z(z)$, estimate the transformation $z=T(r)$.
- The solution exploits histogram equalization.

Histogram specification (cont…)

- Equalize the initial histogram of the image:

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

- Equalize the target histogram:

$$s = G(z) = (L-1)\int_0^r p_z(w)\,dw$$

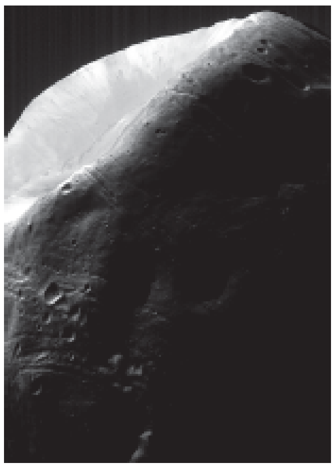
$$G(z) = T(r)$$

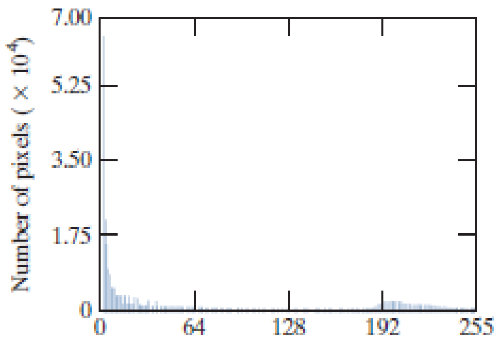
- Obtain the inverse transform: $z = G^{-1}(s) = G^{-1}(T(r))$

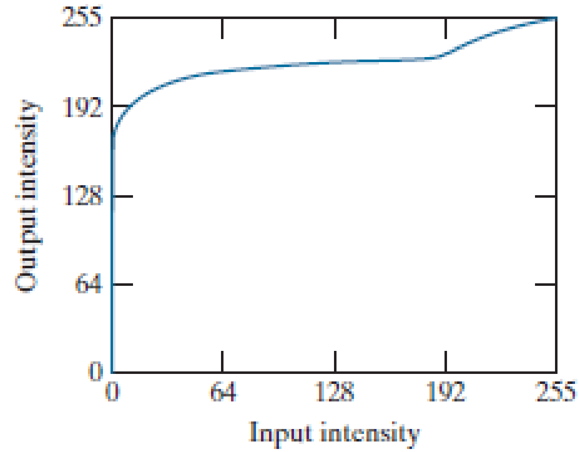
In practice, for every value of r in the image:

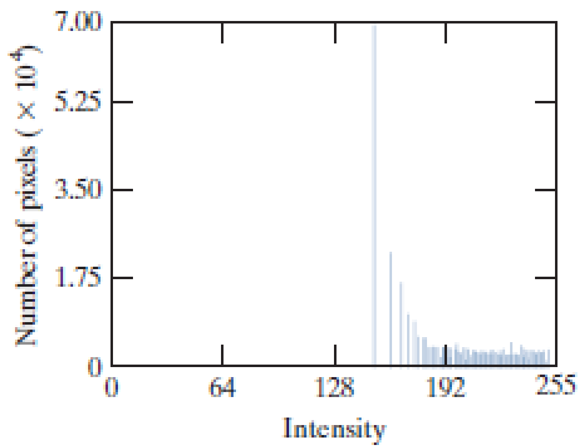
- get its equalized transformation $s=T(r)$.
- perform the inverse mapping $z=G^{-1}(s)$, where $s=G(z)$ is the equalized target histogram.

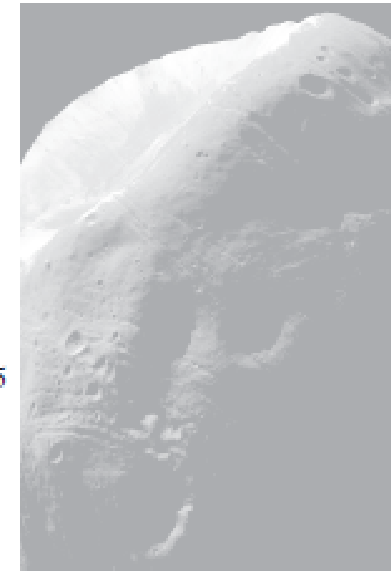
Histogram Specification (cont...)











Histogram equalization

Histogram specification (cont…)

The discrete case:

- Equalize the initial histogram of the image:

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

- Equalize the target histogram:

$$s_k = G(z_q) = (L-1)\sum_{i=0}^{q} p_z(r_i)$$

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

- Obtain the inverse transform: $z_q = G^{-1}(s_k) = G^{-1}(T(r_k))$

Histogram specification (algo…)

1. Compute the histogram, pr(r), of the input image, and use it to map the intensities in the input to those in the histogram-equalized image. Round the resulting s_k, to the integer range $[0, L - 1]$.
2. Compute all values of $G(z_q)$ for $q = 0, .., L - 1$, where $p_z(z_i)$ are the values of the specified histogram. Round the values of G to integers in $[0, L - 1]$and store them in a lookup table
3. For every s_k, use the stored values of G from Step 2 to find the corresponding value of z_q so that G(z_q) is closest to s_k. Store these mappings from s to z. When more than one value of z_q gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.
4. Form the histogram-specified image by mapping every equalized pixel with value s_k to the corresponding pixel with value z_q in the histogram-specified image, using the mappings found in Step 3.

Histogram Specification (cont...) Example

Consider again the 3-bit 64x64 image:

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

It is desired to transform this histogram to:

$$p_z(z_0) = 0.00 \quad p_z(z_1) = 0.00 \quad p_z(z_2) = 0.00 \quad p_z(z_3) = 0.15$$
$$p_z(z_4) = 0.20 \quad p_z(z_5) = 0.30 \quad p_z(z_6) = 0.20 \quad p_z(z_7) = 0.15$$

with $z_0 = 0, z_1 = 1, z_2 = 2, z_3 = 3, z_4 = 4, z_5 = 5, z_6 = 6, z_7 = 7.$

Histogram Specification (cont...) Example

The first step is to equalize the input (as before):

$$s_0 = 1,\ s_1 = 3,\ s_2 = 5,\ s_3 = 6,\ s_4 = 6,\ s_5 = 7,\ s_6 = 7,\ s_7 = 7$$

The next step is to equalize the output:

$$G(z_0) = 0 \quad G(z_1) = 0 \quad G(z_2) = 0 \quad G(z_3) = 1$$
$$G(z_4) = 2 \quad G(z_5) = 5 \quad G(z_6) = 6 \quad G(z_7) = 7$$

Notice that $G(z)$ is not strictly monotonic. We must resolve this ambiguity by choosing, e.g. the smallest value for the inverse mapping.

Histogram Specification (cont...) Example

Perform inverse mapping: find the smallest value of z_q that provides the closest $G(z_q)$ to s_k:

$s_k = T(r_i)$	$G(z_q)$
$s_0 = 1$	$G(z_0) = 0$
$s_1 = 3$	$G(z_1) = 0$
$s_2 = 5$	$G(z_2) = 0$
$s_3 = 6$	$G(z_3) = 1$
$s_4 = 6$	$G(z_4) = 2$
$s_5 = 7$	$G(z_5) = 5$
$s_6 = 7$	$G(z_6) = 6$
$s_7 = 7$	$G(z_7) = 7$

$$s_k \rightarrow z_q$$

$$1 \rightarrow 3$$

$$3 \rightarrow 4$$

$$5 \rightarrow 5$$

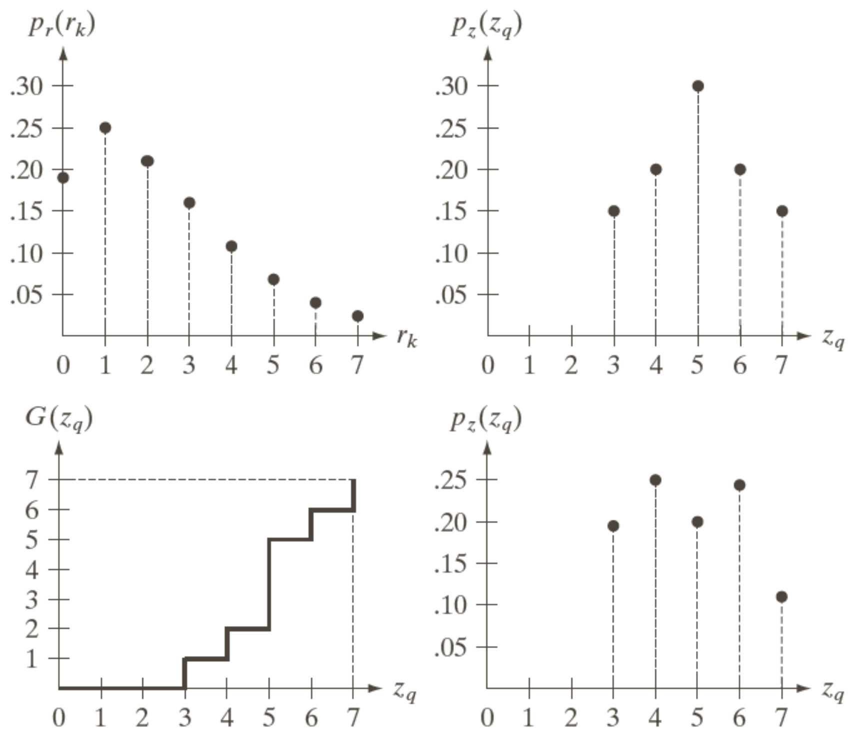
$$6 \rightarrow 6$$

$$7 \rightarrow 7$$

e.g. every pixel with value $s_0=1$ in the histogram-equalized image would have a value of 3 (z_3) in the histogram-specified image.

Histogram Specification (cont...) Example

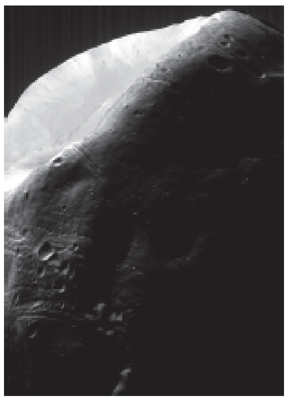
Notice that due to discretization, the resulting histogram will rarely be exactly the same as the desired histogram.

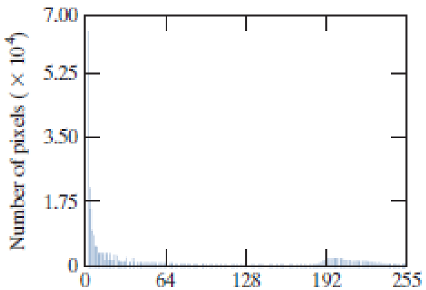


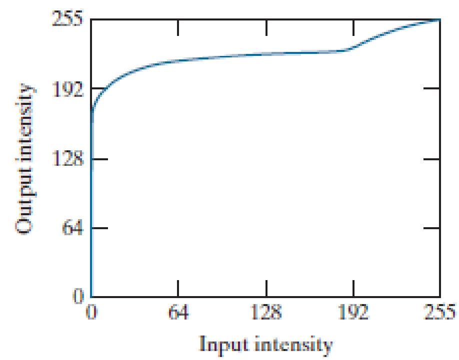
a	b
c	d

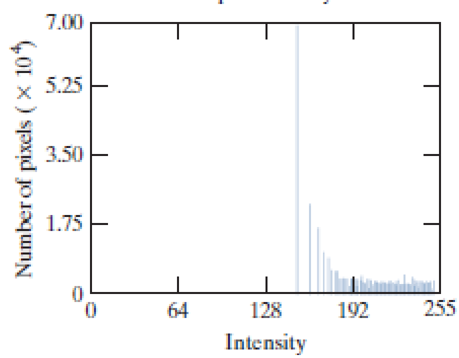
FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

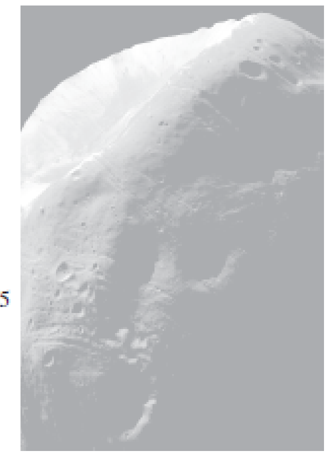
Histogram Specification (cont...)









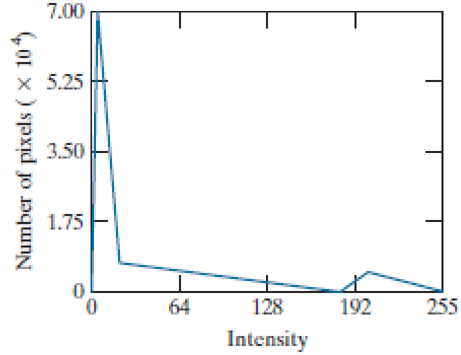


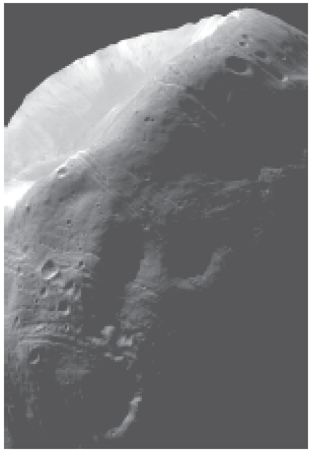
Original image

Histogram equalization

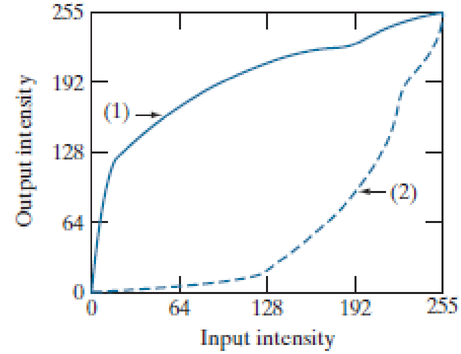
Histogram Specification (cont...)

Specified histogram

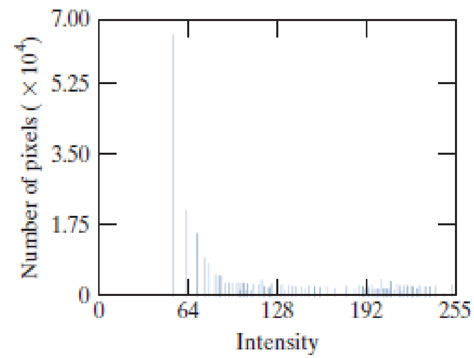




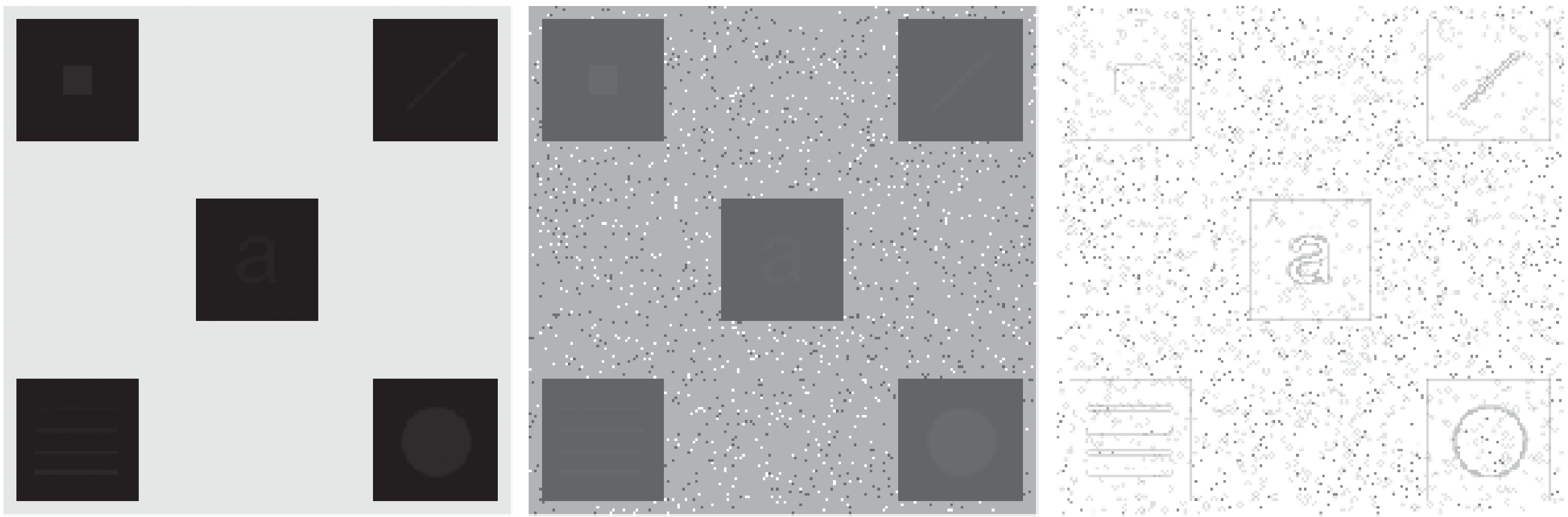
Transformation function and its inverse



Resulting histogram

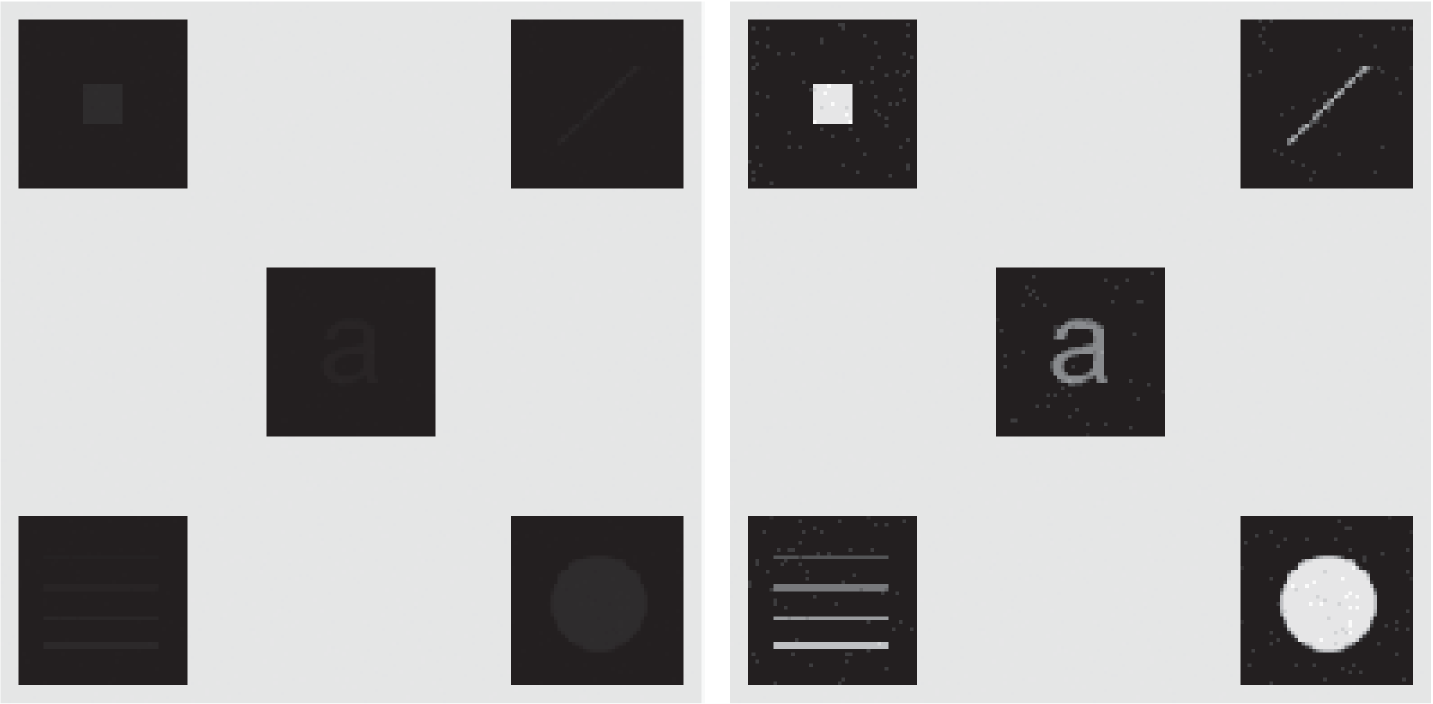


Local Histogram Processing



- Image in (a) is slightly noisy but the noise is imperceptible.
- HE enhances the noise in smooth regions (b).
- Local HE reveals structures having values close to the values of the squares and small sizes to influence HE (c).

Local Histogram Processing



Summary

We have looked at:
- Different kinds of image enhancement
- Histograms
- Histogram equalisation
- Histogram specification

Next we will start to look at spatial filtering and neighbourhood operations

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Spatial Filtering

Άγγελος Γιώτης

a.giotis@uoi.gr

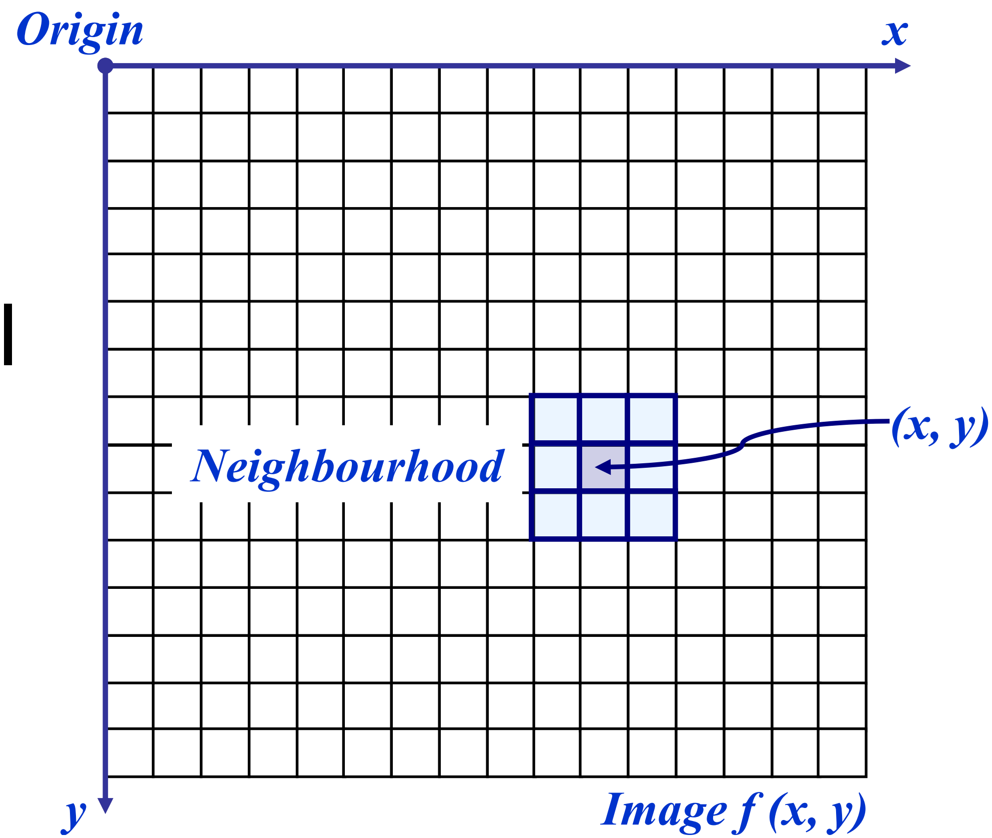
Contents

In this lecture we will look at spatial filtering techniques:

- – Neighbourhood operations
- – What is spatial filtering?
- – Smoothing operations
- – What happens at the edges?
- – Correlation and convolution
- – Sharpening filters
- – Combining filtering techniques

Neighbourhood Operations

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly a rectangle around a central pixel
- Any size rectangle and any shape filter are possible



Simple Neighbourhood Operations

Some simple neighbourhood operations include:

- **Min:** Set the pixel value to the minimum in the neighbourhood
- **Max:** Set the pixel value to the maximum in the neighbourhood
- **Median:** The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] the median is 15).

Simple Neighbourhood Operations

Some simple neighbourhood operations include:

- **Average/Mean:** Set the pixel value to the mean value over all the pixels in the neighbourhood
- Sometimes the median works better than the average

Simple Neighbourhood Operations - Examples

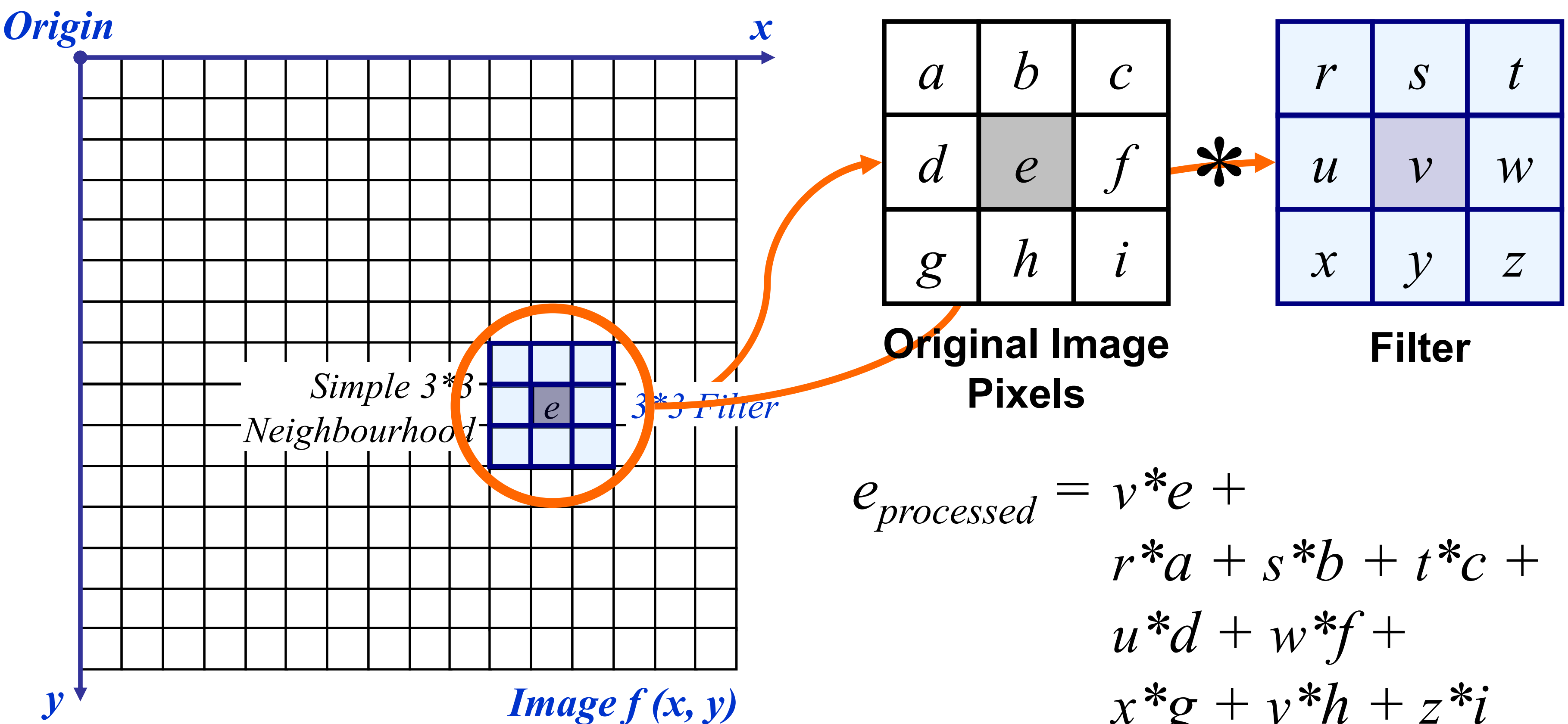
Min(1, 7, 15, 18, 24) = 1

Max(1, 7, 15, 18, 24) = 24

Mean(1 , 7, 15, 18, 24) = 13

Median(1 , 7, 15, 17, 18, 24) = 16
(even case: median = average of the two median values 15 and 17)

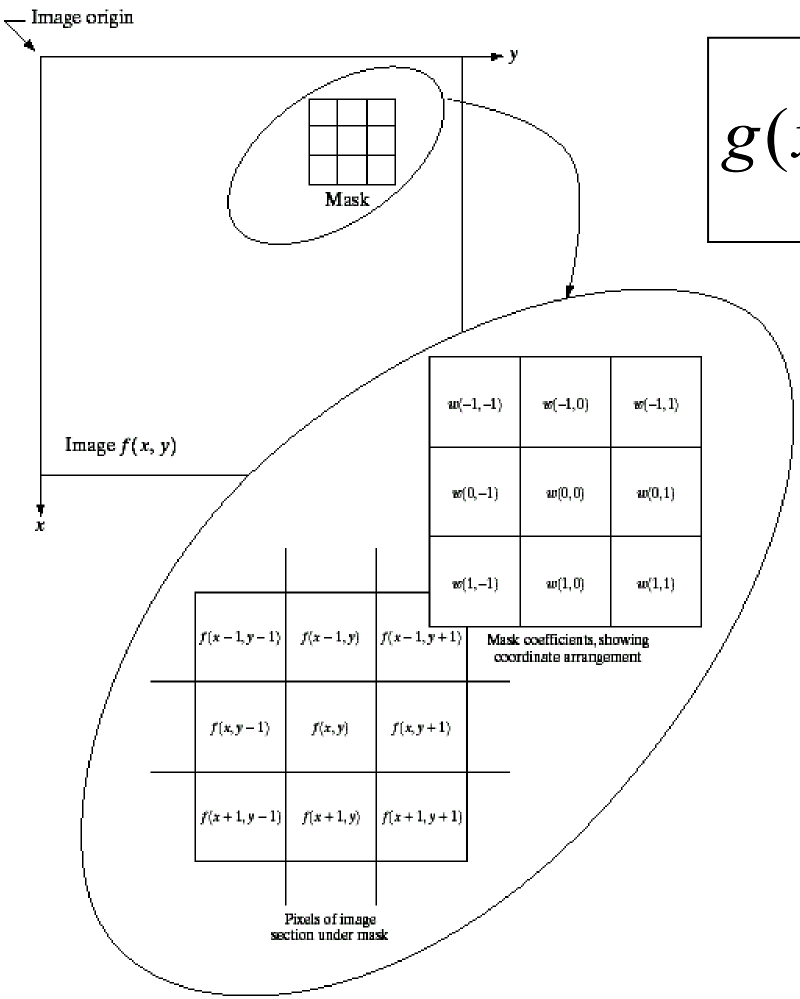
The Spatial Filtering Process



$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the filtered image

Spatial Filtering: Equation Form

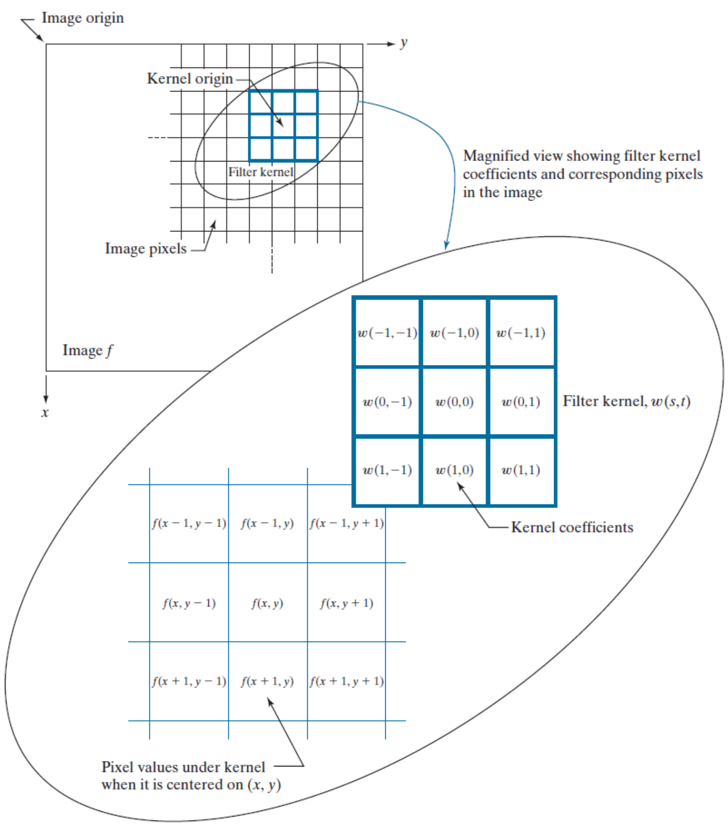


$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Filtering can be given in equation form as shown above

Notations are based on the image shown to the left

Spatial Filtering: Equation Form



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

Filtering can be given in equation form as shown above

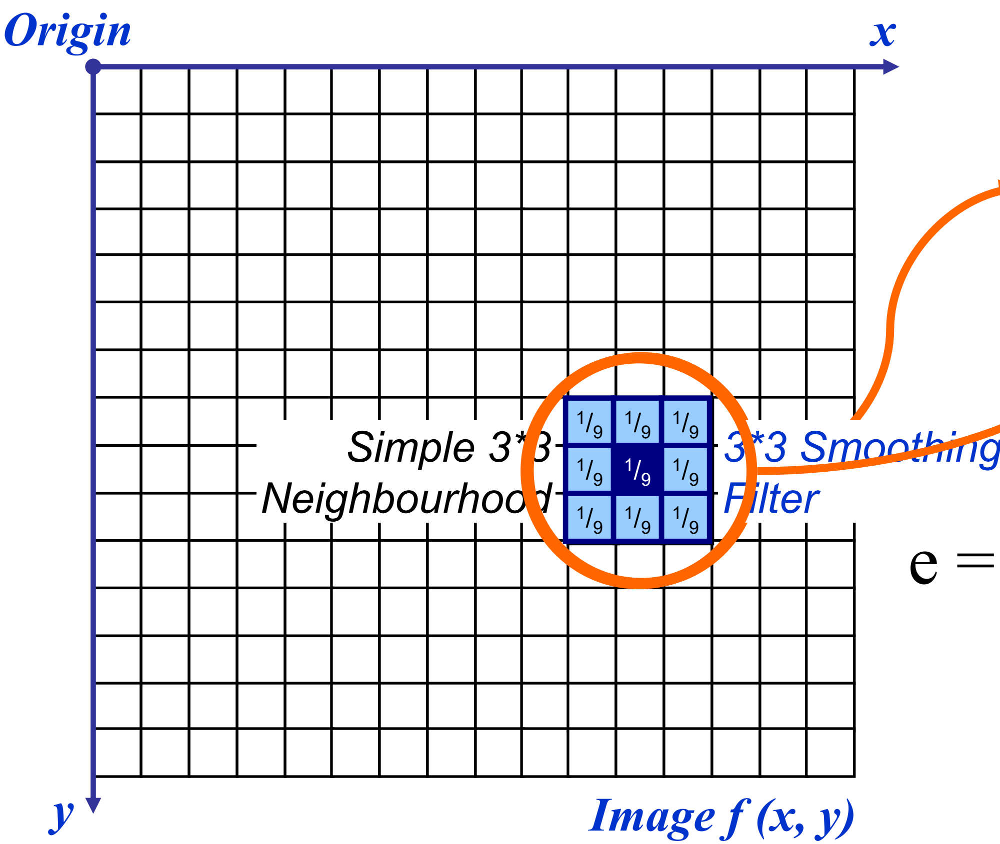
Notations are based on the image shown to the left

Smoothing Spatial Filters

- One of the simplest spatial filtering operations we can perform is a smoothing operation
  - Simply **average** all of the pixels in a neighbourhood around a central value
  - Especially useful in removing noise from images
  - Also useful for highlighting gross detail

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Smoothing Spatial Filtering



104	*100*	*108*
99	*106*	*98*
95	*90*	*85*

Original Image Pixels

$*$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

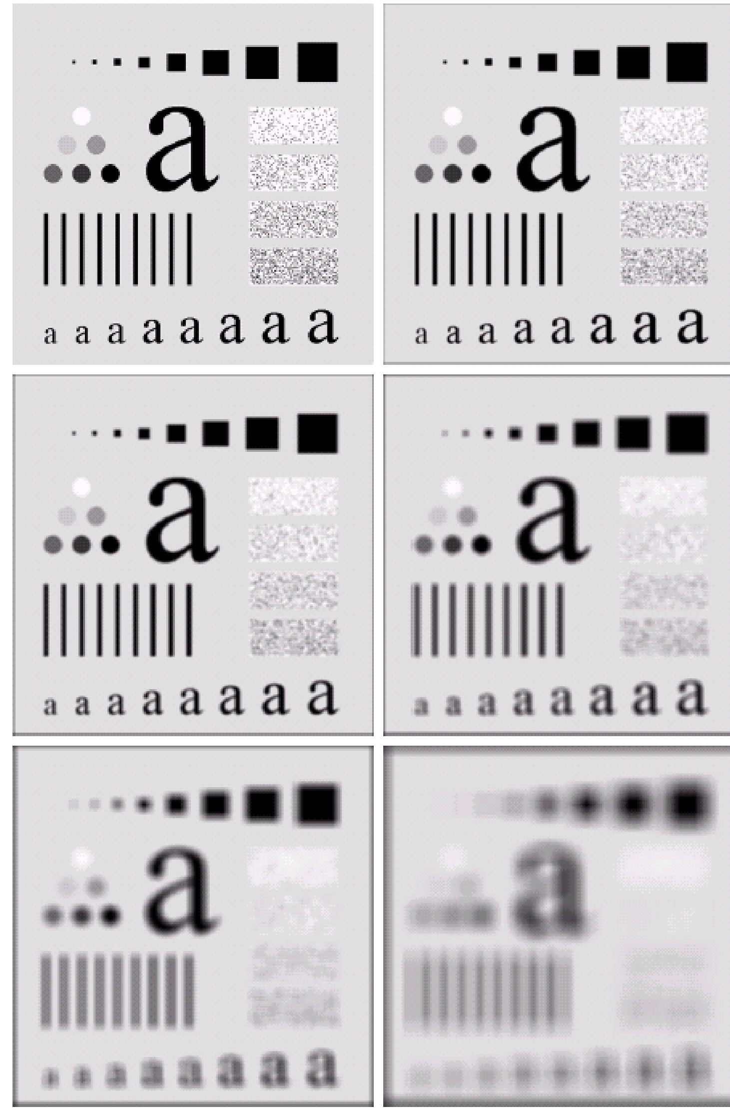
Filter

$$
\begin{aligned}
e = {} & ^1/_9*106 + \\
& ^1/_9*104 + {}^1/_9*100 + {}^1/_9*108 + \\
& ^1/_9*99 + {}^1/_9*98 + \\
& ^1/_9*95 + {}^1/_9*90 + {}^1/_9*85 \\
= {} & 98.3333
\end{aligned}
$$

The above is repeated for every pixel in the original image to generate the smoothed image.

Image Smoothing Example

- The image at the top left is an original image of size 500*500 pixels
- The subsequent images show the image after filtering with an averaging filter of increasing sizes
  - 3, 5, 9, 15 and 35
- Notice how detail begins to disappear



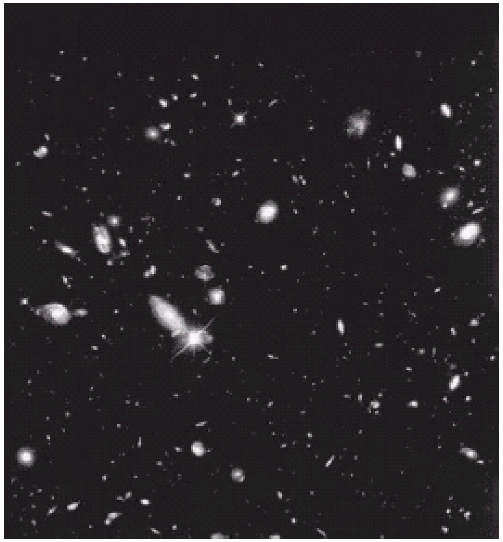
Weighted Smoothing Filters

- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a *weighted averaging*

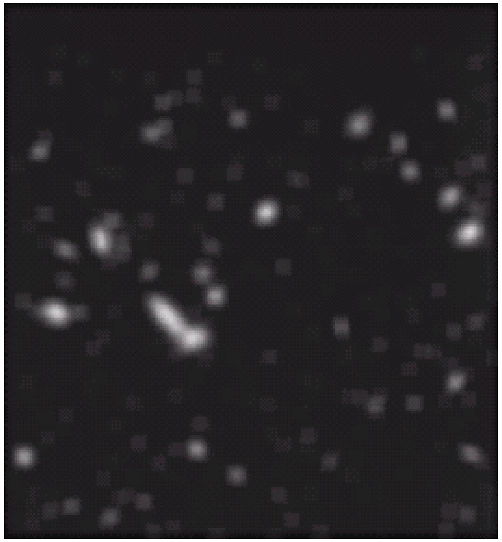
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

Another Smoothing Example

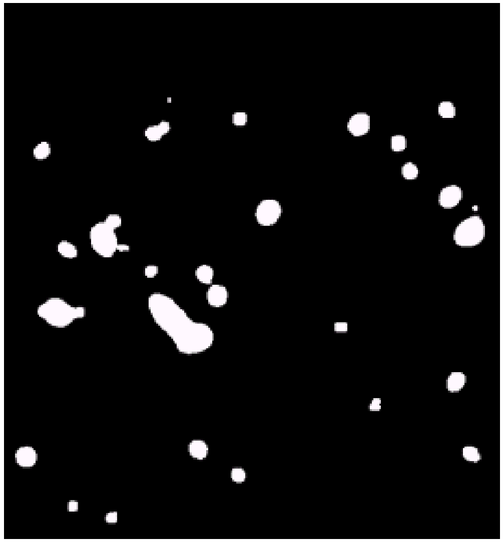
- By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image

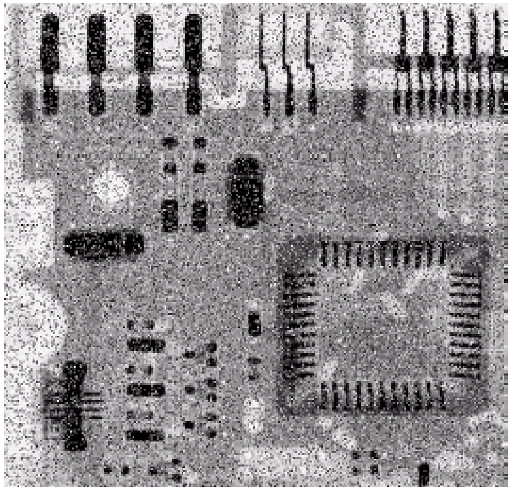


Smoothed Image



Thresholded Image

Averaging Filter vs. Median Filter Example



Original Image With Noise

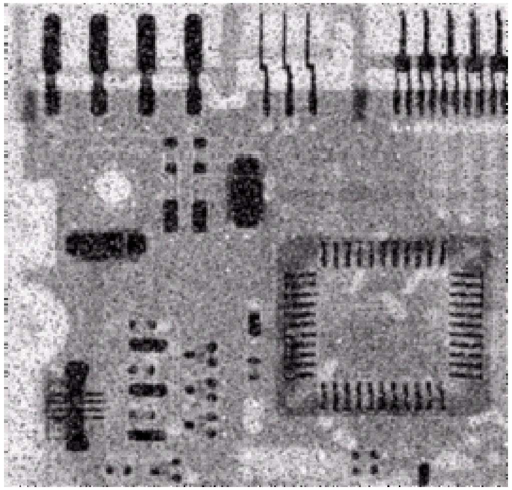


Image After Averaging Filter

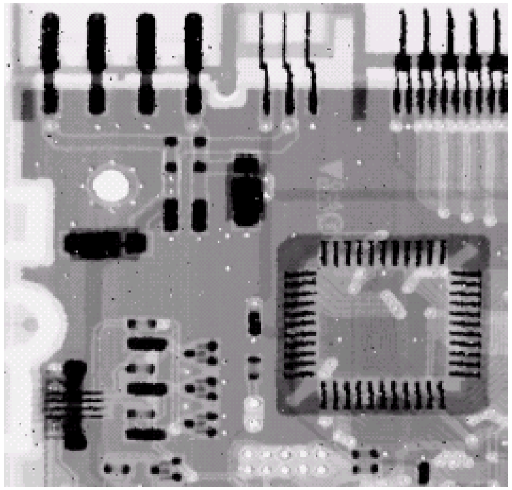


Image After Median Filter

- Filtering is often used to remove noise from images
- Sometimes a median filter works better than an averaging filter

Spatial smoothing and image approximation

- Spatial smoothing may be viewed as a process for estimating the value of a pixel from its neighbours.
- What is the value that "best" approximates the intensity of a given pixel given the intensities of its neighbours?
- We have to define "best" by establishing a criterion.

Spatial smoothing and image approximation (cont...)

A standard criterion is the the sum of squares differences.

$$E=\sum_{i=1}^{N}\left[x(i)-m\right]^2 \Leftrightarrow m=\arg\min_{m}\left\{\sum_{i=1}^{N}\left[x(i)-m\right]^2\right\}$$

$$\frac{\partial E}{\partial m}=0 \Leftrightarrow -2\sum_{i=1}^{N}\left(x(i)-m\right)=0 \Leftrightarrow \sum_{i=1}^{N}x(i)=\sum_{i=1}^{N}m$$

$$\Leftrightarrow \sum_{i=1}^{N}x(i)=Nm \Leftrightarrow m=\frac{1}{N}\sum_{i=1}^{N}x(i)$$ The average value

Spatial smoothing and image approximation (cont...)

Another criterion is the the sum of absolute differences.

$$E = \sum_{i=1}^{N} \left| x(i) - m \right| \quad \Leftrightarrow m = \arg\min_{m} \left\{ \sum_{i=1}^{N} \left| x(i) - m \right| \right\}$$

$$\frac{\partial E}{\partial m} = 0 \Leftrightarrow -\sum_{i=1}^{N} sgn\left(x(i) - m\right) = 0, \quad sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

There must be equal in quantity positive and negative values.

$$m = \text{median}\{x(i)\}$$

Spatial smoothing and image approximation (cont...)

- The median filter is non linear:

$$\text{median}\{x+y\} \neq \text{median}\{x\} + \text{median}\{y\}$$

- It works well for impulse noise (e.g. salt and pepper).
- It requires sorting of the image values.
- It preserves the edges better than an average filter in the case of impulse noise.
- It is robust to impulse noise at 50%.

Spatial smoothing and image approximation (cont...)

Example

x[n]	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**

edge

Impulse noise

x[n]	**1**	**3**	**1**	**1**	**1**	**2**	**3**	**2**	**2**	**3**

Median (N=3)

x[n]	**-**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**-**

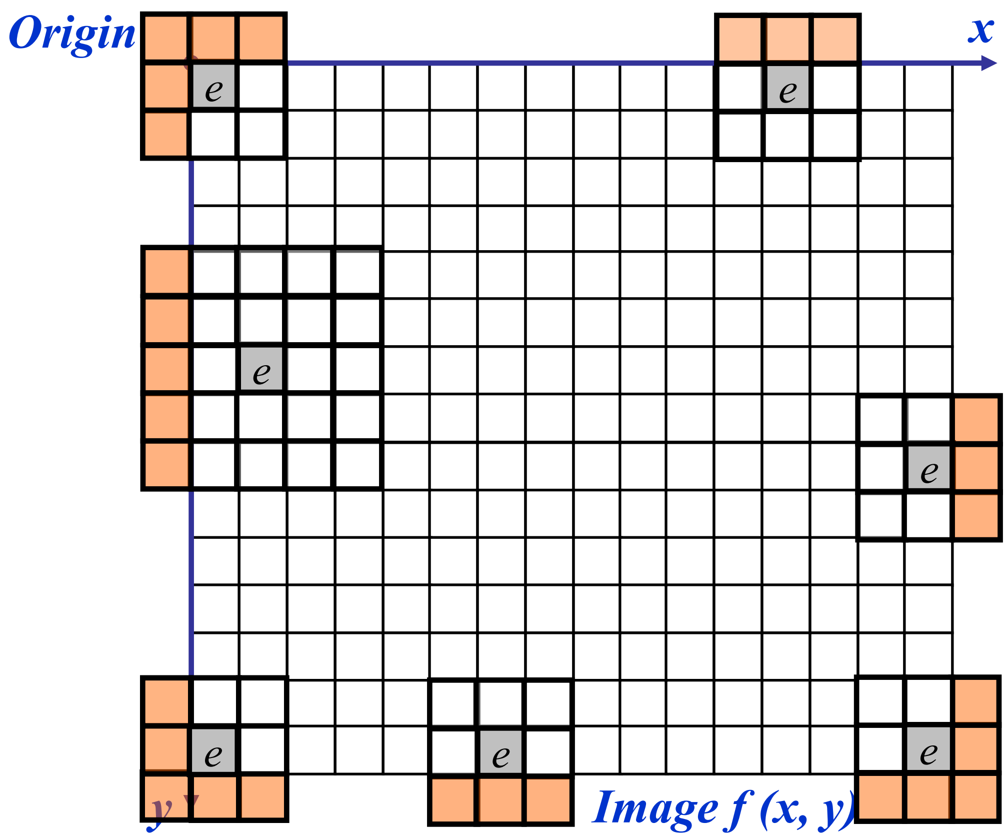
Average (N=3)

x[n]	**-**	**1.7**	**1.7**	**1**	**1.3**	**2**	**2.3**	**2.3**	**2.2**	**-**

The edge is smoothed

Strange Things Happen At The Edges!

At the edges of an image we are missing pixels to form a neighbourhood



Strange Things Happen At The Edges! (cont…)

There are a few approaches to dealing with missing edge pixels:

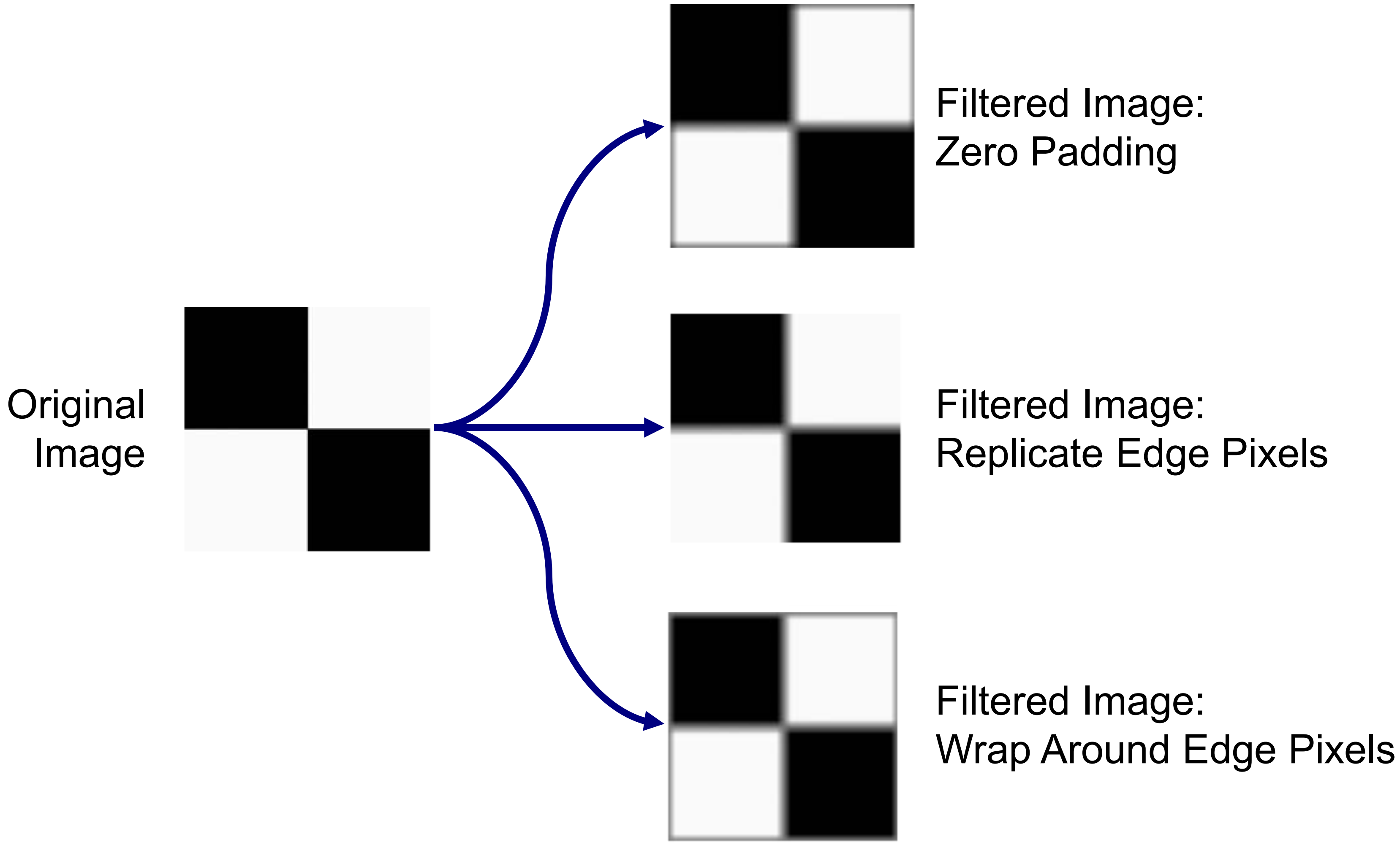
- Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing

Strange Things Happen At The Edges! (cont…)

There are a few approaches to dealing with missing edge pixels:

- Pad the image
  - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image
- Allow pixels *wrap around* the image
  - Can cause some strange image artefacts

Strange Things Happen At The Edges! (cont…)



Correlation & Convolution

- The filtering we have been talking about so far is referred to as *correlation* with the filter itself referred to as the *correlation kernel*
- *Convolution* is a similar operation, with just one subtle difference

a	b	c
d	e	e
f	g	h

Original Image Pixels

$*$

r	s	t
u	v	w
x	y	z

Filter

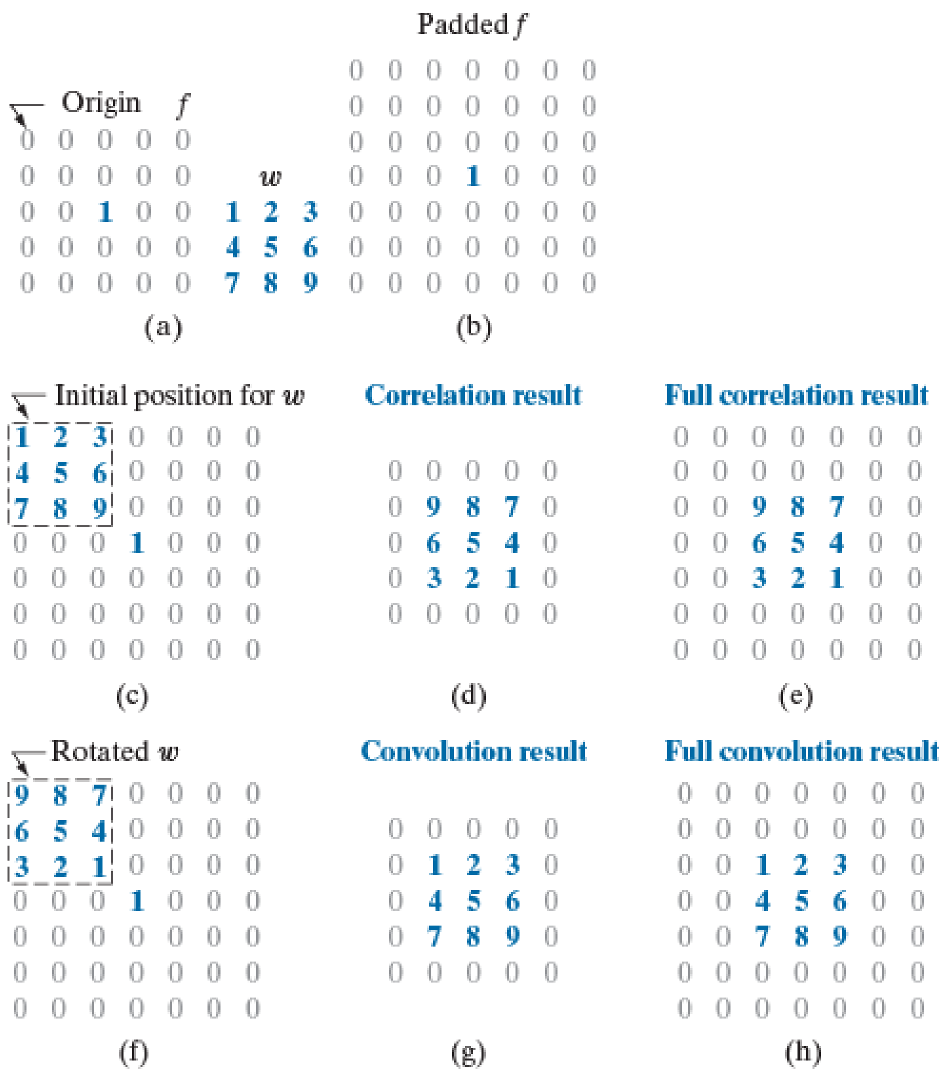
$$\begin{aligned} e_{processed} = {} & v*e + \\ & z*a + y*b + x*c + \\ & w*d + u*e + \\ & t*f + s*g + r*h \end{aligned}$$

- For symmetric filters it makes no difference.

Correlation & Convolution (cont.)

	Correlation	Convolution	
	Origin f w	Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 4 2 8	0 0 0 1 0 0 0 0 8 2 4 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 4 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 4 2 1 Starting position alignment	(j)
(c)	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Starting position	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Starting position	(k)
(d)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 1 shift	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 1 shift	(l)
(e)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 3 shifts	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 3 shifts	(m)
(f)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Final position	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Final position	(n)
	Correlation result	Convolution result	
(g)	0 8 2 4 2 1 0 0	0 1 2 4 2 8 0 0	(o)
	Extended (full) correlation result	Extended (full) convolution result	
(h)	0 0 0 8 2 4 2 1 0 0 0 0	0 0 0 1 2 4 2 8 0 0 0 0	(p)

Correlation & Convolution (cont.)



Fundamental properties of convolution and correlation

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \,☆\, g) + (f \,☆\, h)$

Separable Filters

- A 2-D function $G(x, y)$ is considered separable if it can be expressed as the product of two 1-D functions, $G_1(x)$ and $G_1(y)$, such that: $G(x, y) = G_1(x)G_2(y)$.
- For example, the 2 * 3 kernel

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

is seperable because it can be expressed as the outer product of the vectors

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Separable Filters

$$\mathbf{c}\,\mathbf{r}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = w$$

- A separable kernel of size m × n can be expressed as the outer product of two vectors, v and w of size m × 1 and n × 1, respectively: $w = \mathbf{v}\mathbf{w}^T$
- The product of a column vector and a row vector is the same as the 2-D convolution of the vectors.
- Convolving a separable kernel: $w = w_1 \star w_2$

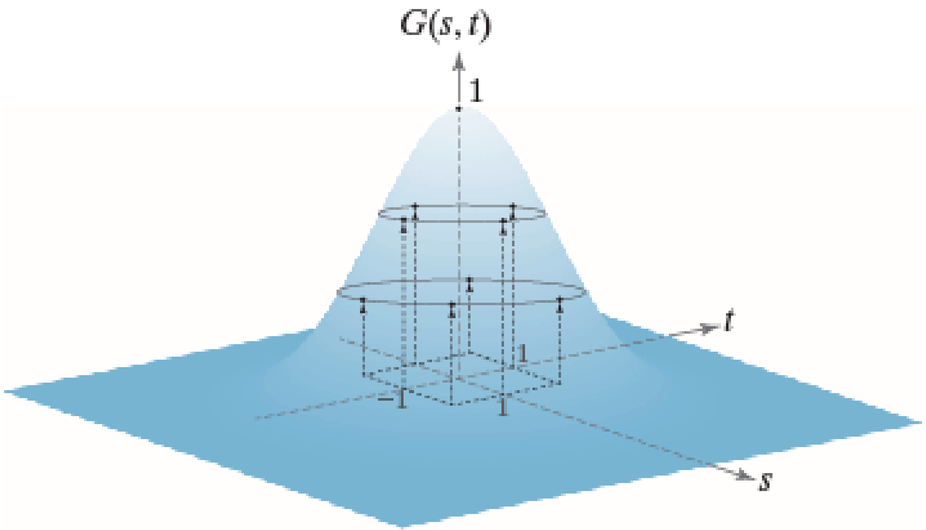
with an image is the same as convolving w_1 with f first, and then convolving the result with w_2

Gaussian filter

- Separable
  - Fast computation
- Isotropic
  - Independent of orientation
- The product of two Gaussians is also a Gaussian
- The convolution of two Gaussians is also a Gaussian

$$w(s,t) = Ke^{-\left(\frac{s^2+t^2}{2\sigma^2}\right)}$$

Gaussian filter (cont.)



$$\frac{1}{4.8976} \times$$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

A sampling size of $6\sigma \times 6\sigma$ is sufficient

Gaussian filter (cont.)

FIGURE 3.36 (a)A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.33(d). We used $K = 1$ in all cases.

Gaussian filter (cont.)

a b c

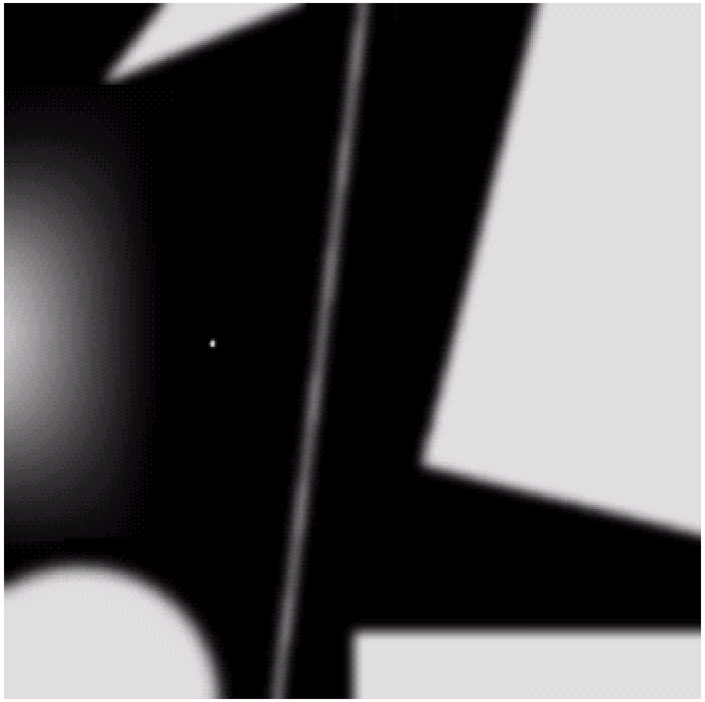
FIGURE 3.37 (a) Result of filtering Fig. 3.36(a) using a Gaussian kernels of size 43×43, with $\sigma = 7$. (b) Result of using a kernel of 85×85, with the same value of σ. (c) Difference image.

Sharpening Spatial Filters

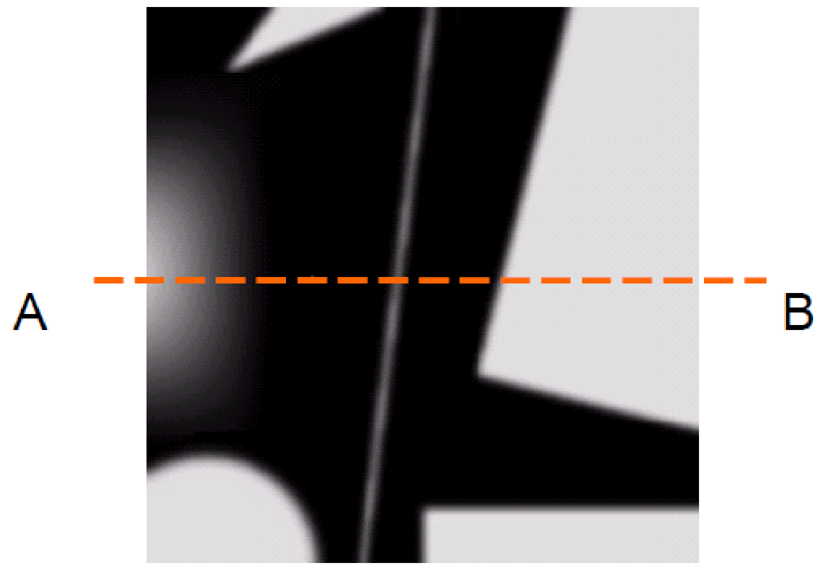
- Previously we have looked at smoothing filters which remove fine detail
- *Sharpening spatial filters* seek to highlight fine detail
  - Remove blurring from images
  - Highlight edges
- Sharpening filters are based on spatial differentiation

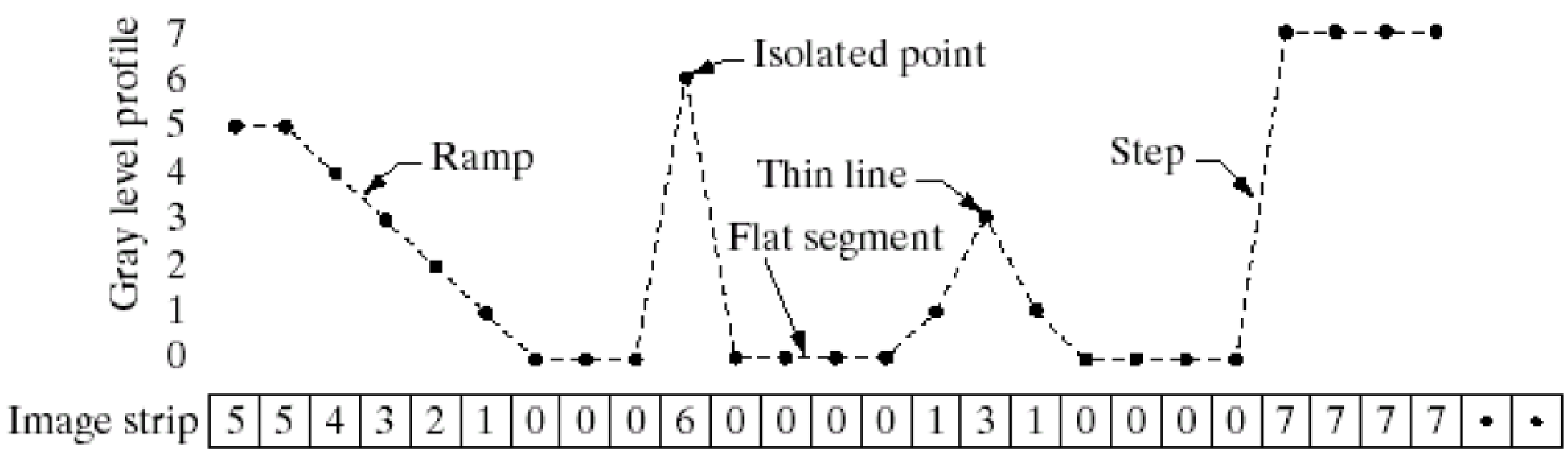
Sharpening Spatial Filters

- We want to measure the rate of change.
- We consider an example in 1D.



Sharpening Spatial Filters





Derivative Filters Requirements

- First derivative filter output
  - Zero at constant intensities
  - Non zero at the onset of a step or ramp
  - Non zero along ramps
- Second derivative filter output
  - Zero at constant intensities
  - Non zero at the onset and end of a step or ramp
  - Zero along ramps of constant slope

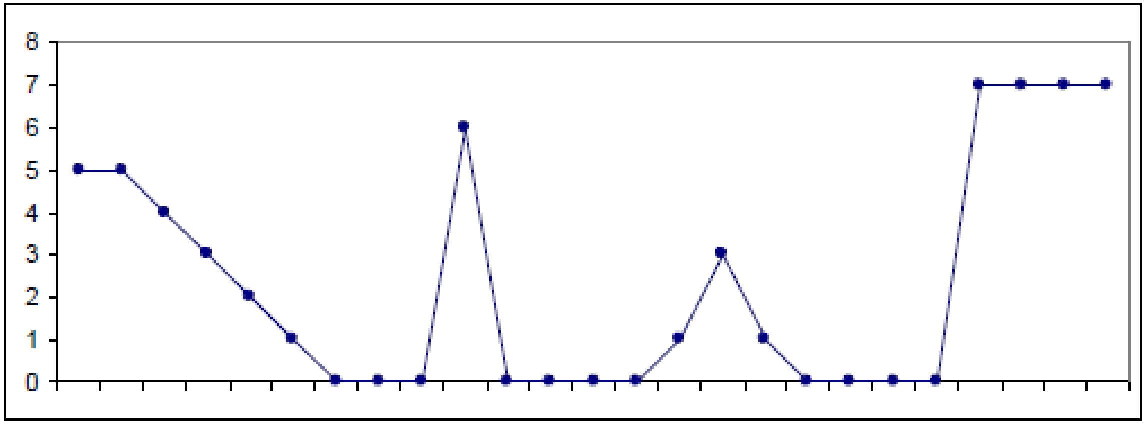
1st Derivative

- Discrete approximation of the 1st derivative

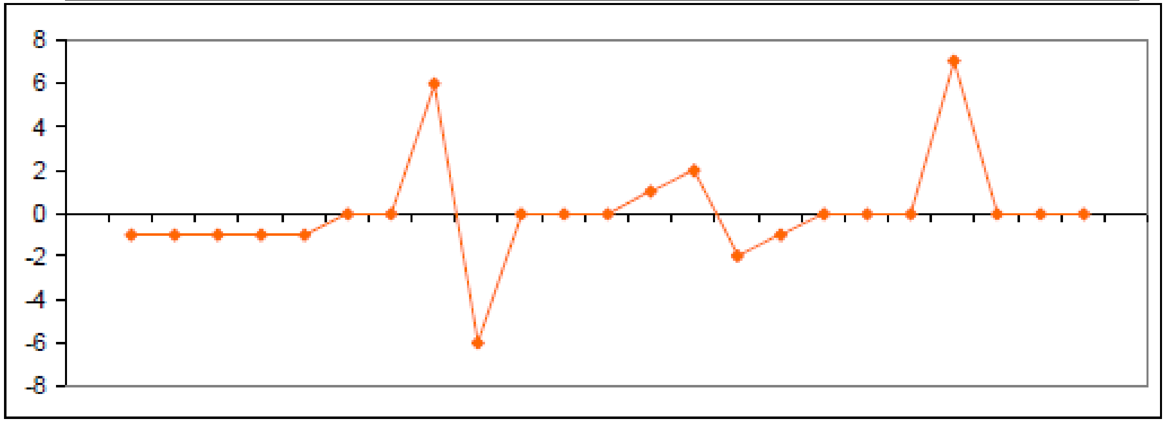
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- It is just the difference between subsequent values and measures the rate of change of the function

1st Derivative



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

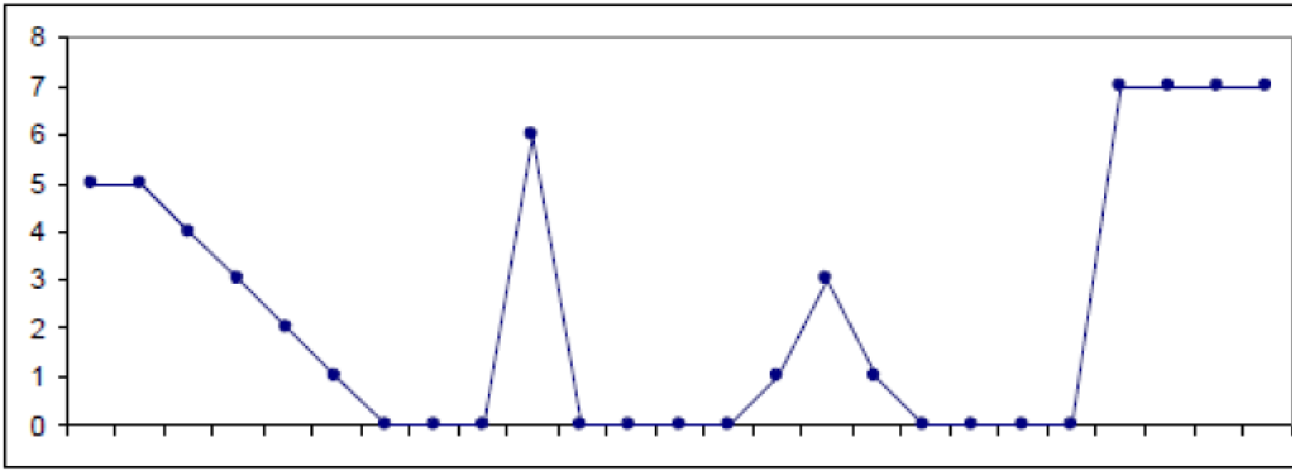


2nd Derivative

- Discrete approximation of the 2nd derivative:

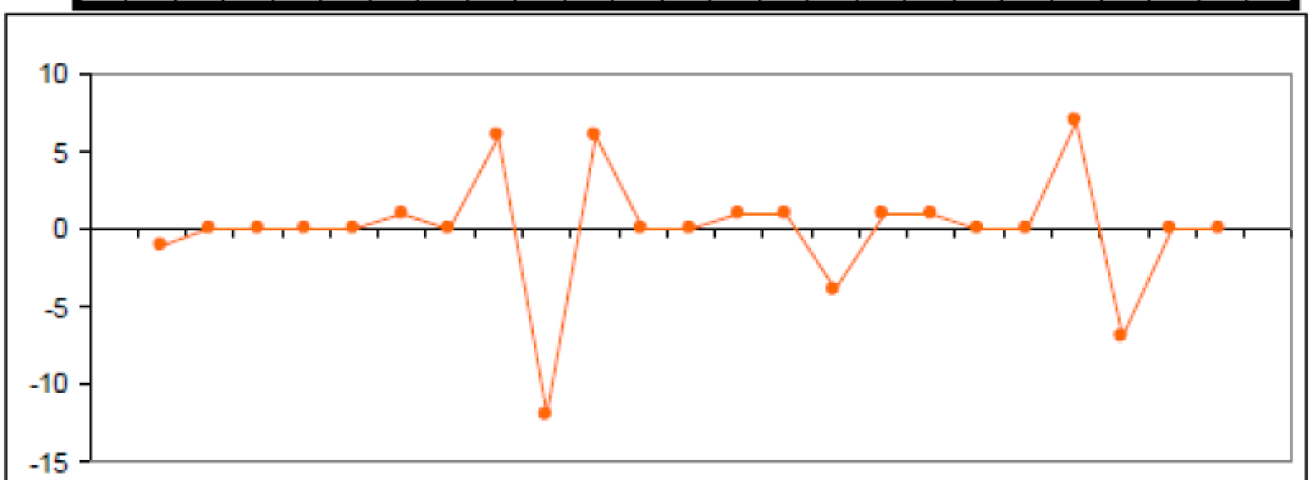
$$\frac{\partial^2 f}{\partial^2 x} = f(x-1) - 2f(x) + f(x+1)$$

2nd Derivative



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

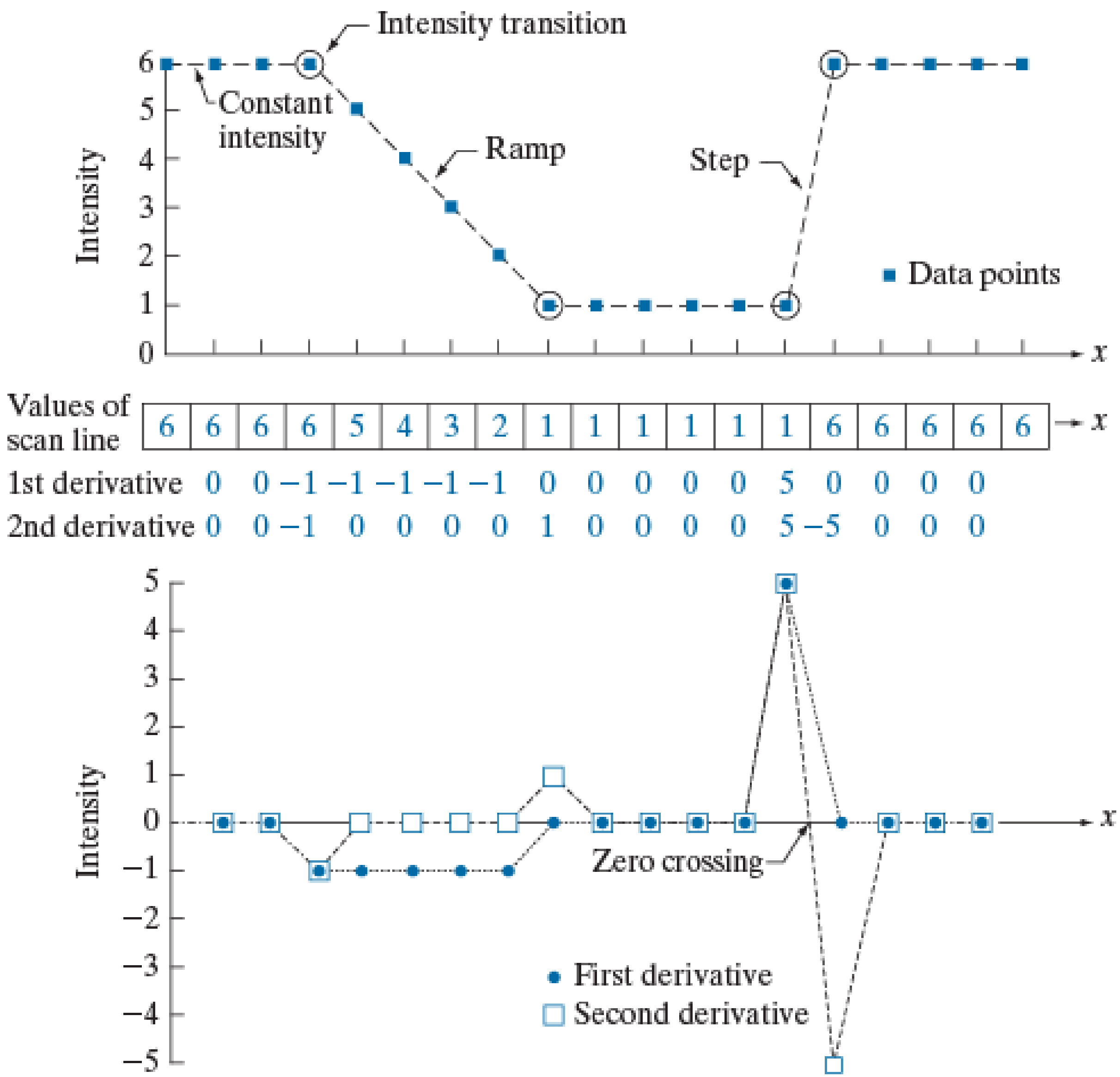
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



2nd Derivative

- Edges in images typically behave like the 'ramps' we examined. The first derivative is constant and produces thick zones at the edges.
- The second derivative gives non-zero response only at the beginning and end of the edge, while being zero in the middle.

Derivatives



Using Second Derivatives For Image Enhancement

- A common sharpening filter is the **Laplacian**
  - Isotropic
    - Rotation invariant: Rotating the image and applying the filter is the same as applying the filter and then rotating the image.
    - In other words, the Laplacian of a rotated image is the rotated Laplacian of the original image.
  - One of the simplest sharpening filters
  - We will look at a digital implementation

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

The Laplacian

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

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

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

The Laplacian (cont…)

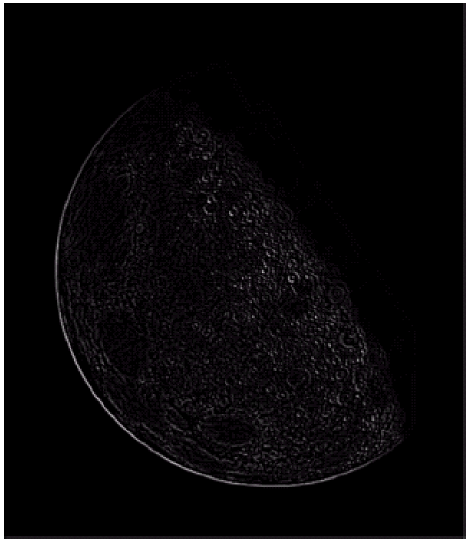
$$\nabla^2 f = -4f(x,y)$$
$$+ f(x+1,y) + f(x-1,y)$$
$$+ f(x,y+1) + f(x,y-1)$$

0	1	0
1	-4	1
0	1	0

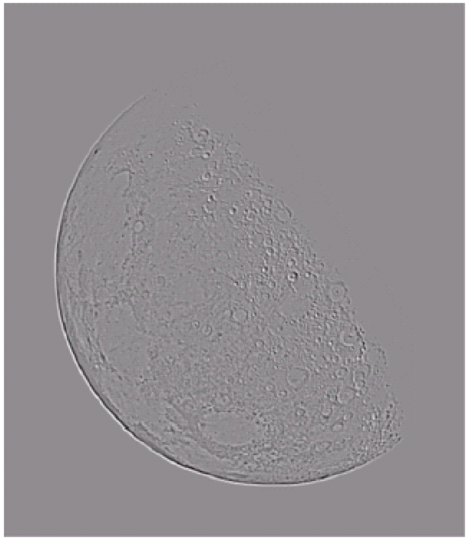
The Laplacian (cont…)

- Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities

Original Image



Laplacian Filtered Image

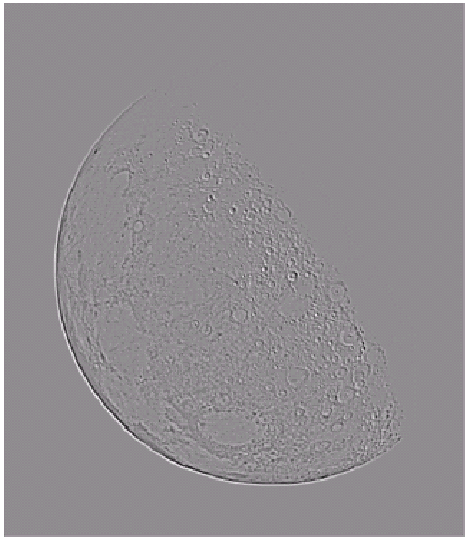


Laplacian Filtered Image Scaled for Display

The Laplacian (cont…)

- The result of a Laplacian filtering is not an enhanced image
- We have to do more work
- Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

$$g(x, y) = f(x, y) - \nabla^2 f$$

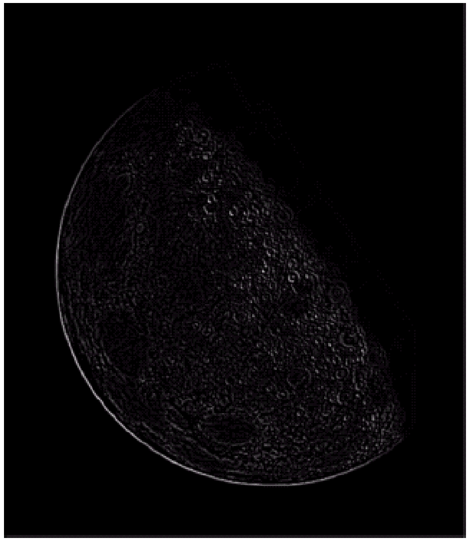


Laplacian Filtered Image Scaled for Display

Laplacian Image Enhancement

Original Image

\-



Laplacian Filtered Image

=

Sharpened Image

- In the final, sharpened image, edges and fine detail are much more obvious

Laplacian Image Enhancement

Simplified Image Enhancement

- The entire enhancement can be combined into a single filtering operation:

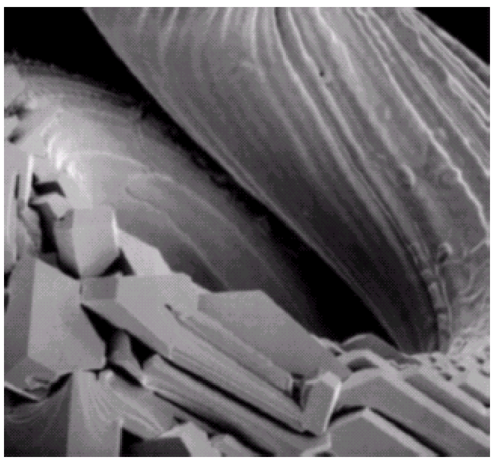
$$g(x,y) = f(x,y) - \nabla^2 f$$

$$= 5f(x,y) - f(x+1,y) - f(x-1,y)$$

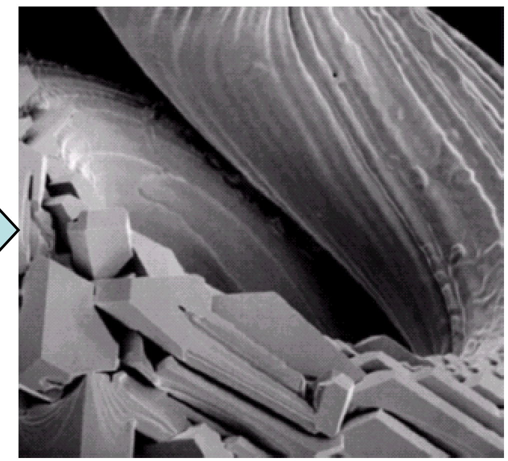
$$- f(x,y+1) - f(x,y-1)$$

Simplified Image Enhancement (cont…)

- This gives us a new filter which does the whole job in one step



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



Variants On The Simple Laplacian

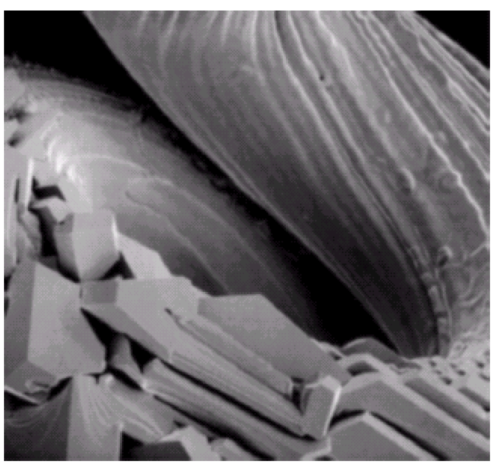
- There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

Standard Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



-1	-1	-1
-1	9	-1
-1	-1	-1

Unsharp masking

- Used by the printing industry
- Subtracts an unsharped (smooth) image from the original image $f(x,y)$.
  - Blur the image

    $b(x,y)=Blur\{f(x,y)\}$
  - Subtract the blurred image from the original (the result is called the *mask*)

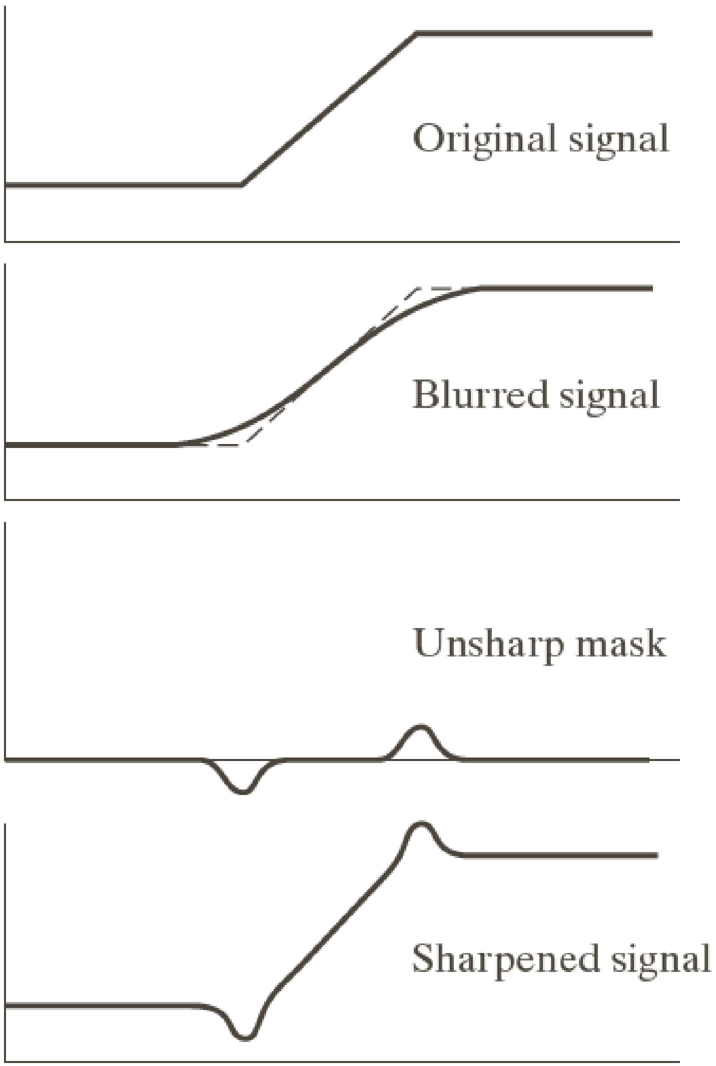
    $g_{mask}(x,y)=f(x,y)-b(x,y)$
  - Add the mask to the original

    $g(x,y)=f(x,y)+k\, g_{mask}(x,y)$, k being non negative

Unsharp masking (cont...)

Sharpening mechanism

If $k>1$, the process is referred to as **highboost filtering**



Unsharp masking (cont...)

Original image

Blurred image
(Gaussian 5x5, σ=3)

Mask

Unsharp masking (k=1)

Highboost filtering (k=4.5)

Using First Derivatives For Image Enhancement

$$\nabla f = \begin{bmatrix} G_x & G_y \end{bmatrix}^T = \begin{bmatrix} \dfrac{\partial f}{\partial x} & \dfrac{\partial f}{\partial y} \end{bmatrix}^T$$

- Although the derivatives are linear operators, the gradient magnitude is not.
- Also, the partial derivatives are not rotation invariant (isotropic).
- The magnitude of the gradient vector is isotropic.

$$||\nabla f|| = \sqrt{G_x^2 + G_y^2}$$

Using First Derivatives For Image Enhancement (cont…)

- In some applications it is more computationally efficient to approximate:

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

- This expression preserves relative changes in intensity but it is not isotropic.
- Isotropy is preserved only for a limited number of rotational increments which depend on the filter masks (e.g. 90 deg.).

Sobel Operators

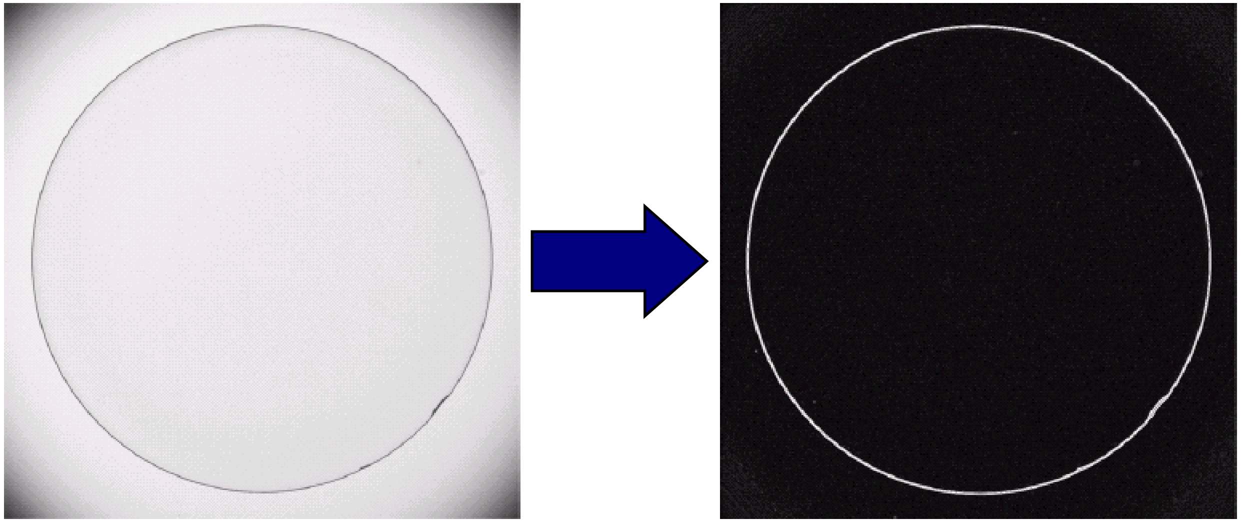
- **Sobel operators** introduce the idea of differentiating by giving more importance to the center point:

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

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

- Note that the coefficients sum to 0 to give a 0 response at areas of constant intensity.

Sobel operator Example



- Sobel gradient aids to eliminate constant or slowly varying shades of gray and assist automatic inspection.
- It also enhances small discontinuities in a flat gray filed.
- General comments
  - 1st derivatives tend to produce more thick edges
  - 2nd order derivatives have better response to detail (thin edge)
  - 2nd order derivatives produce double response to edges.

Combining Spatial Enhancement Methods

- Successful image enhancement is typically not achieved using a single operation
- Rather we combine a range of techniques in order to achieve a final result
- This example will focus on enhancing the bone scan to the right

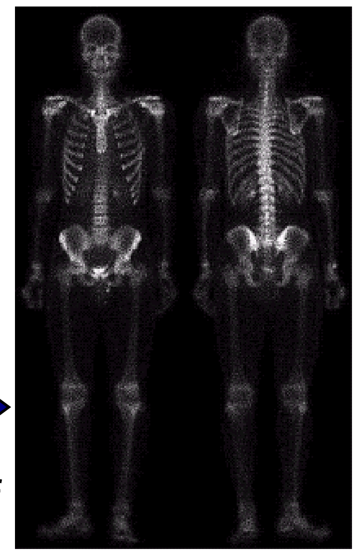
Combining Spatial Enhancement Methods (cont…)

(a)

Laplacian filter of bone scan (a)

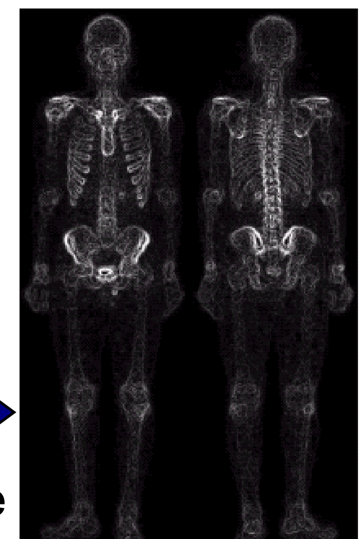
(b)

Sharpened version of bone scan achieved by subtracting (a) and (b)



(c)

Sobel filter of bone scan (a)



(d)

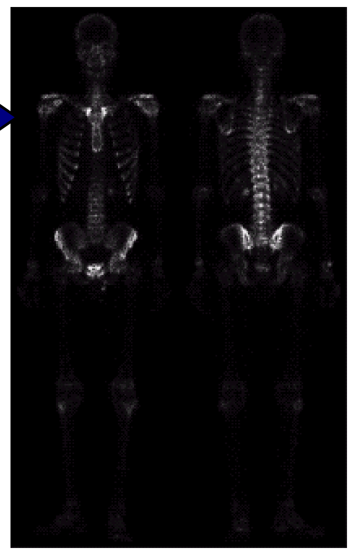
Combining Spatial Enhancement Methods (cont…)

(e)

Image (d) smoothed with a 5*5 averaging filter

The product of (c) and (e) which will be used as a mask

(f)

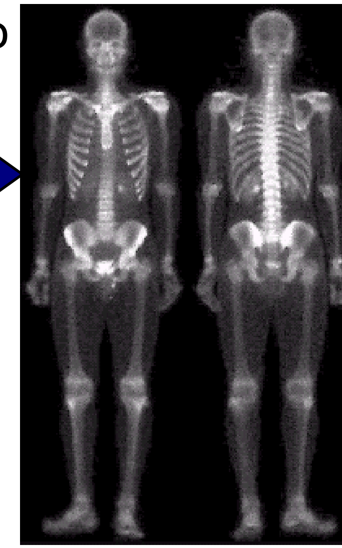


Sharpened image which is sum of (a) and (f)

(g)

Result of applying a power-law trans. to (g)

(h)



Combining Spatial Enhancement Methods (cont…)

Compare the original and final images

Summary

In this lecture we have looked at the idea of spatial filtering and in particular:

- Neighbourhood operations
- The filtering process
- Smoothing filters
- Dealing with problems at image edges when using filtering
- Correlation and convolution
- Sharpening filters
- Combining filtering techniques

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Filtering in the Frequency Domain (Fundamentals)

Άγγελος Γιώτης

a.giotis@uoi.gr

Outline

- *We will see how we can analyze the frequency content of the image. Specifically, we will first present the 1D versions for:*

- *Continuous Fourier Transform*
- *Discrete Fourier Transform (DFT)*

...and we will see how they are generalized for 2D discrete signals, such as digital images.

Filtering in the Frequency Domain

Filter: A device or material for suppressing or minimizing waves or oscillations of certain frequencies.

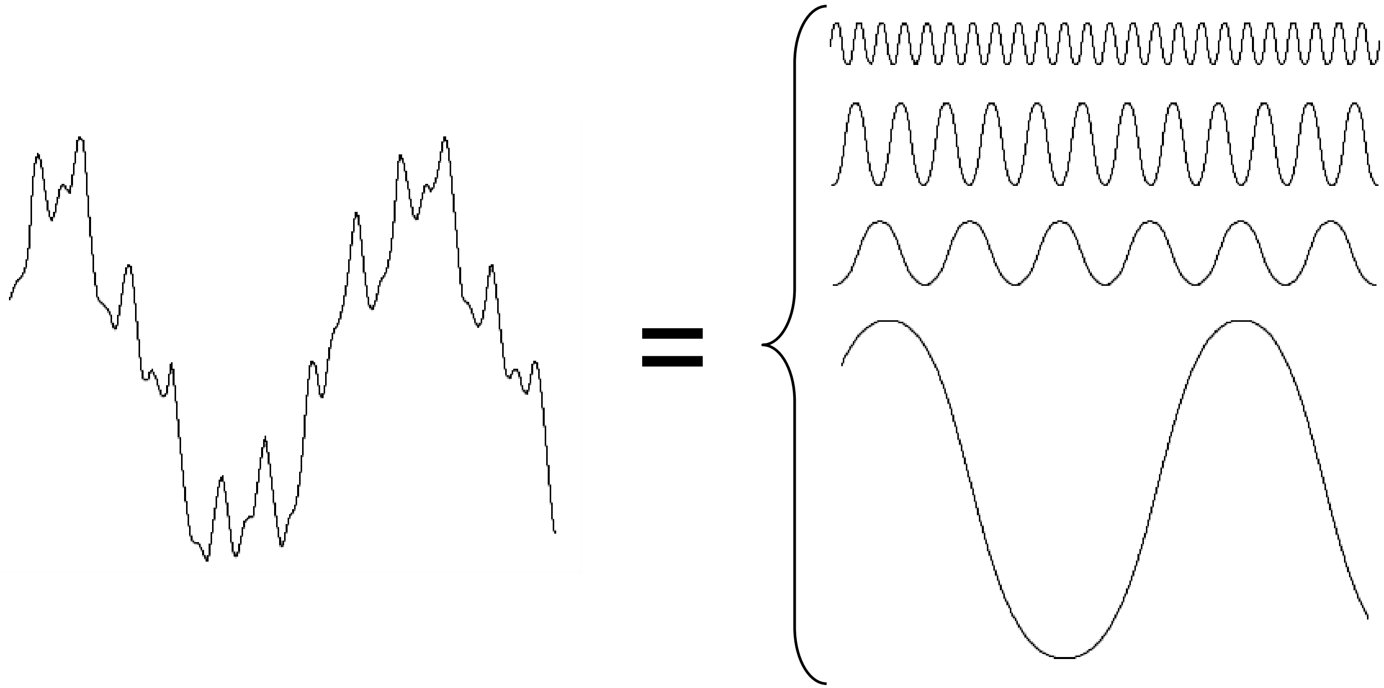
Frequency: The number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable.

Webster's New Collegiate Dictionary

Jean Baptiste Joseph Fourier

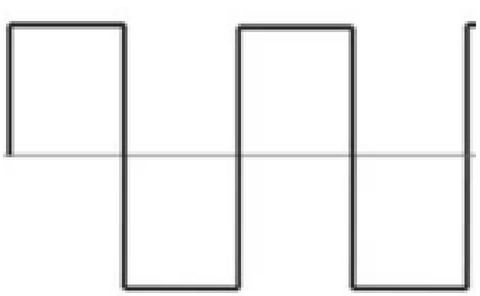
- Fourier was born in Auxerre, France in 1768.
  - Most famous for his work "*La Théorie Analytique de la Chaleur*" published in 1822.
  - Translated into English in 1878: "*The Analytic Theory of Heat*".
- Nobody paid much attention when the work was first published.
- One of the most important mathematical theories in modern engineering.

The Big Idea

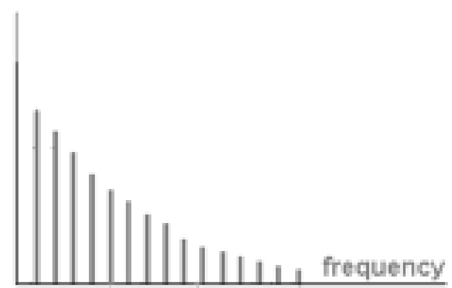


Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*

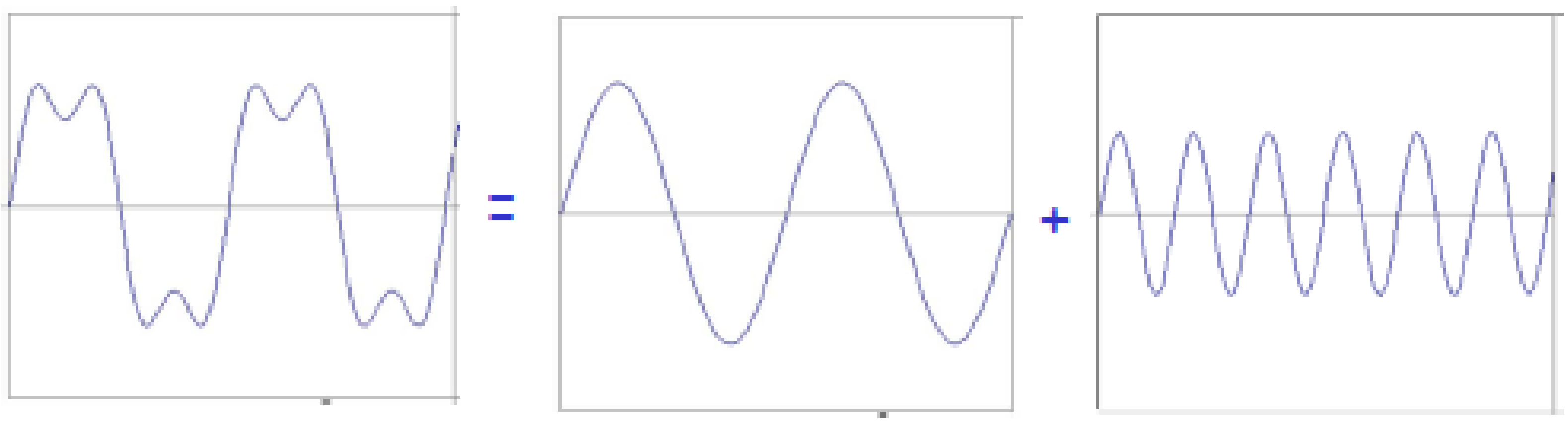
The Big Idea (cont..)



$$= \quad f(x) = \sum_{n=1,3,5,\ldots} \frac{1}{n} \sin nx$$

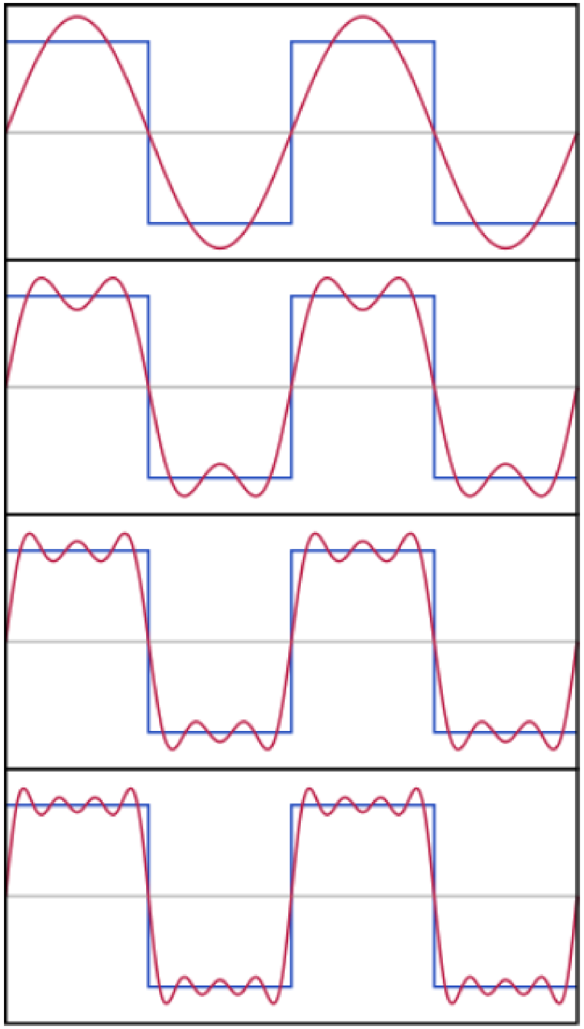


Fourier series



$$f(x) = \sin x + \frac{1}{3}\sin 3x + \ldots$$

Fourier series



Fourier series (analytic expression)

- c_n represents the contribution of the $n-th$ frequency component to the Fourier series of $f(t)$.

$$c_n = \frac{1}{\Delta T} \int_{-\Delta T/2}^{\Delta T/2} f(t)\, e^{-j2\pi \frac{n}{\Delta T} t} dt$$

$$\mathrm{f(t)} = \sum_{n=-\infty}^{+\infty} c_n e^{j2\pi \frac{n}{\Delta T} t}$$

Reminder – Euler's formula approximation using Taylor series

$$e^{j\theta} = \cos\theta + j\sin\theta \qquad f(x) = \sum_{n}^{\infty} \frac{f^{(n)}(0)}{n!} x^n$$

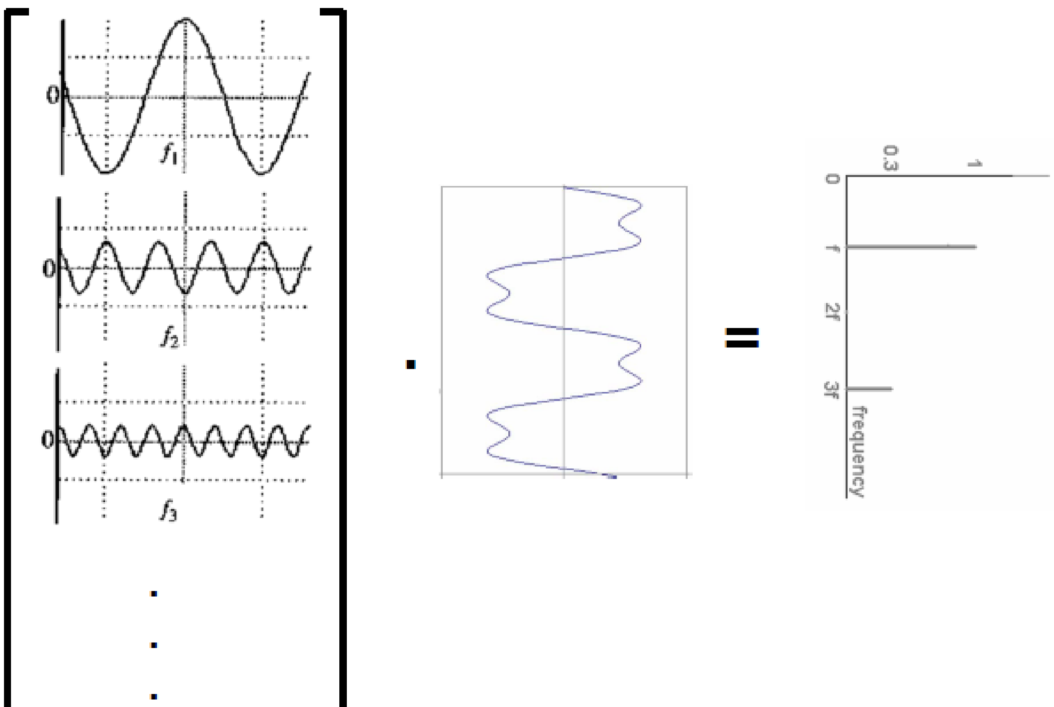
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

$$e^{jx} = 1 + jx - \frac{x^2}{2!} - j\frac{x^3}{3!} + \frac{x^4}{4!} + j\frac{x^5}{5!} - \frac{x^6}{6!} - j\frac{x^7}{7!} + \ldots$$

A different representation (change of basis)

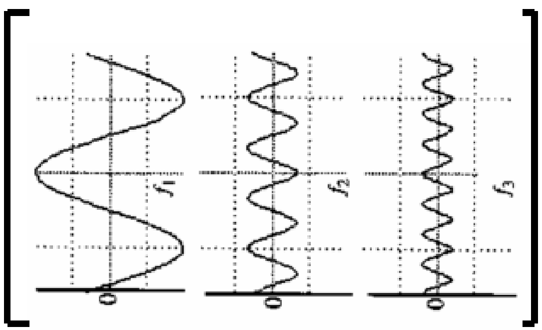
- This representation is akin to a change of basis, where we express the function in terms of a new set of basis functions.
- Trigonometric basis $\{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \ldots\}$ is the basis in which the Fourier series expresses f(x).



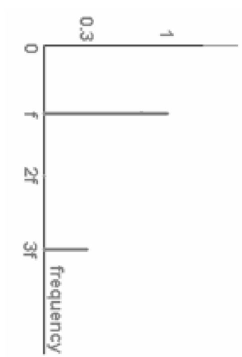
$$c_n = \frac{1}{\Delta \mathrm{T}} \int_{-\Delta \mathrm{T}/2}^{\Delta \mathrm{T}/2} f(t)\, e^{-j2\pi \frac{n}{\Delta \mathrm{T}} t}\, dt$$

A different representation (change of basis)

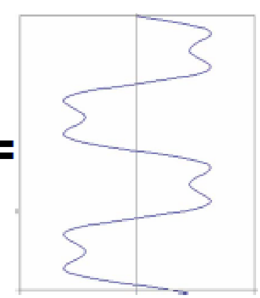
- Same holds for the reverse operation.
- Trigonometric basis changes to the exponential basis $\{e^{inx}\}_{n=-\infty}^{\infty}$



·



=



$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{j2\pi \frac{n}{\Delta T} t}$$

⋮

Transition from discrete to 1D continuous signals

- The Fourier series expansion of a periodic signal $f(t)$.

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{j\frac{2\pi}{T}nt}$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\frac{2\pi}{T}nt} dt$$

1D continuous signals

- The Fourier transform of a continuous signal $f(t)$

$$F(\mu) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi\mu t} dt$$

- The reverse Fourier transform

$$f(t) = \int_{-\infty}^{+\infty} F(\mu) e^{j2\pi\mu t} d\mu$$

1D continuous FT

- A 1-1 mapping between functions $f(t)$ and $F(\mu)$

$$f \leftrightarrow \mathrm{F}$$

$$\mathrm{F}(\mu) = F\{\mathrm{f}(\mathrm{t})\}$$

$$\mathrm{f}(\mathrm{t}) = F^{-1}\{\mathrm{F}(\mu)\}$$

1D continuous signals (cont.)

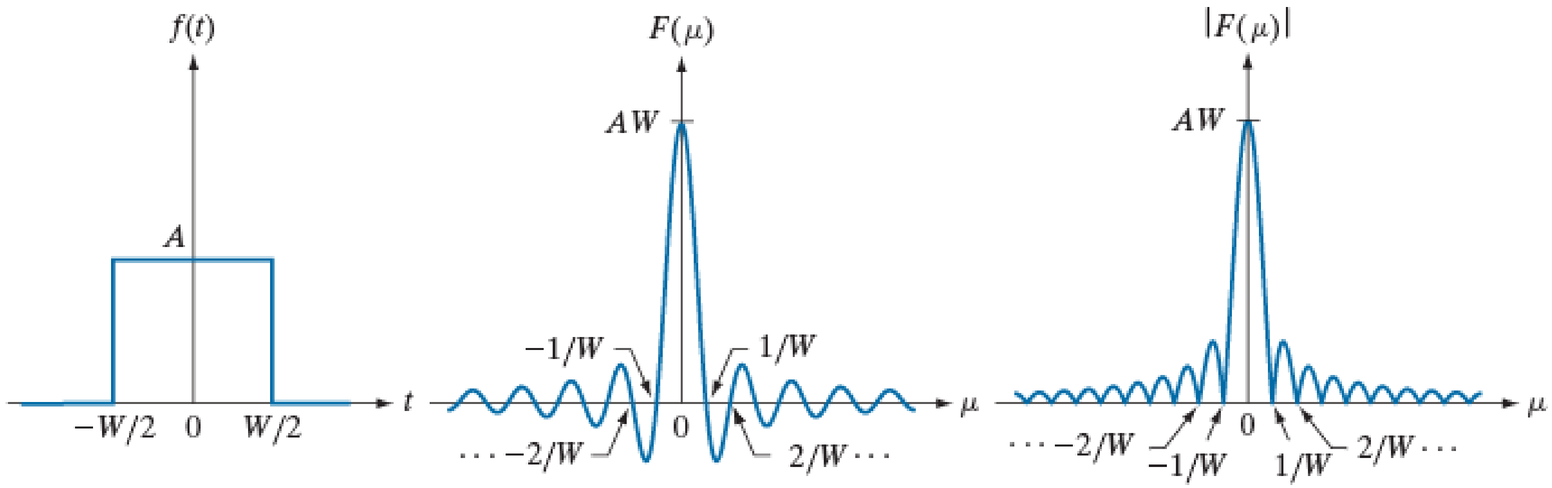


FIGURE 4.4 (a) A box function, (b) its Fourier transform, and (c) its spectrum. All functions extend to infinity in both directions. Note the inverse relationship between the width, W, of the function and the zeros of the transform.

$$F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt = \int_{-W/2}^{W/2} A e^{-j2\pi\mu t} dt = \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_{-W/2}^{W/2}$$

$$= \frac{-A}{j2\pi\mu}\left[e^{-j\pi\mu W} - e^{j\pi\mu W}\right] = \frac{A}{j2\pi\mu}\left[e^{j\pi\mu W} - e^{-j\pi\mu W}\right] = AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}$$

trigonometric identity $\sin\theta = (e^{j\theta} - e^{-j\theta})/2j.$

1D continuous signals (cont.)

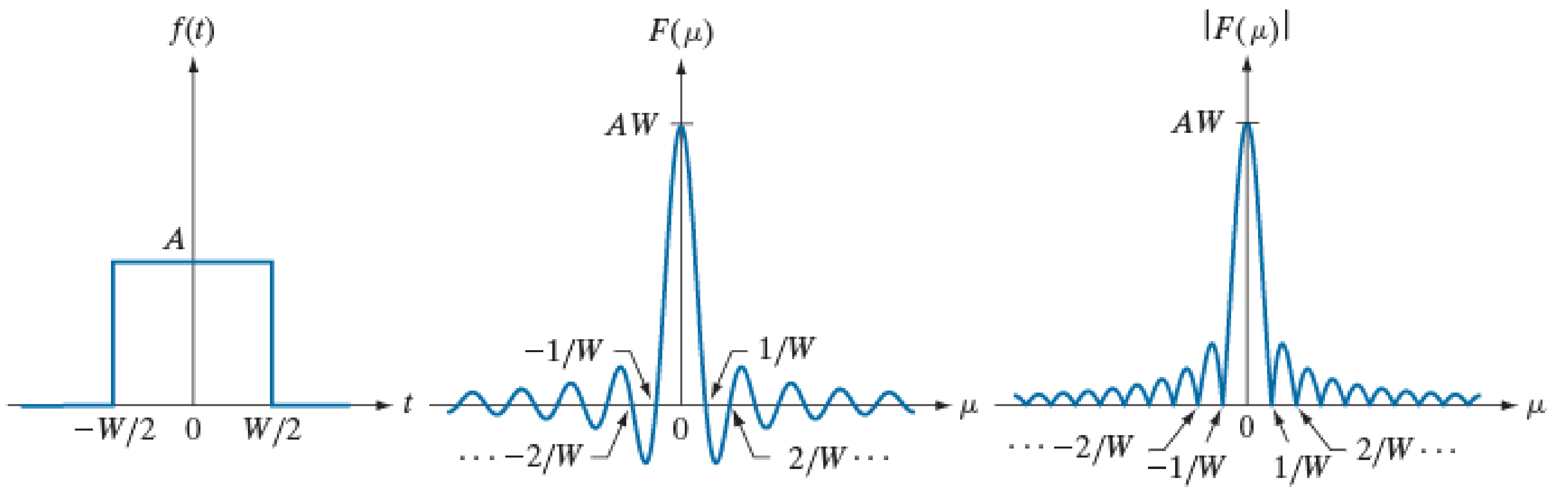


FIGURE 4.4 (a) A box function, (b) its Fourier transform, and (c) its spectrum. All functions extend to infinity in both directions. Note the inverse relationship between the width, W, of the function and the zeros of the transform.

$$f(t) = \mathrm{A}P_{W/2}(t) \leftrightarrow F(\mu) = \mathrm{A}W\frac{\sin(\pi\mu W)}{(\pi\mu W)}$$

1D continuous signals (cont.)

- Convolution property of the FT.
  - convolution in the time domain corresponds to multiplication in the frequency domain, and vice versa.

$$f(t)*h(t)=\int_{-\infty}^{+\infty} f(\tau)h(t-\tau)d\tau$$

$$f(t)*h(t) \leftrightarrow F(\mu)H(\mu)$$

$$f(t)h(t) \leftrightarrow F(\mu)*H(\mu)$$

1D continuous signals

- Dirac delta function may be considered both as continuous and discrete.
- Useful for the representation of discrete signals through sampling of continuous signals.
- The property of "selection“ refers to the fact that the delta function acts as a "filter" or "switch" that selects or "marks" a specific point in time or space.

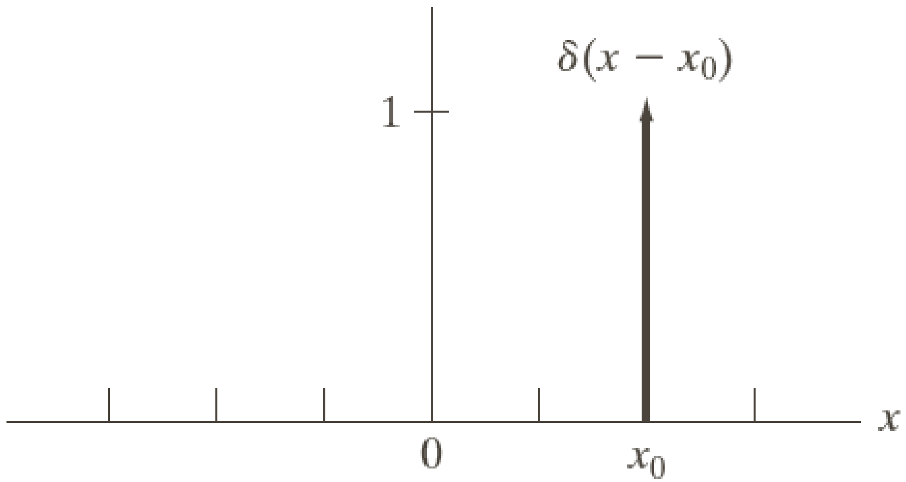
$$\delta(x - x_0) = \begin{cases} +\infty, & x = x_0 \\ 0 & \text{otherwise} \end{cases}$$

$$\int_{-\infty}^{+\infty} \delta(x) = 1$$

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

1D continuous signals

- Discrete version of “Delta”



$$\delta(x - x_0) = 0, \text{αν } x \neq x_0$$

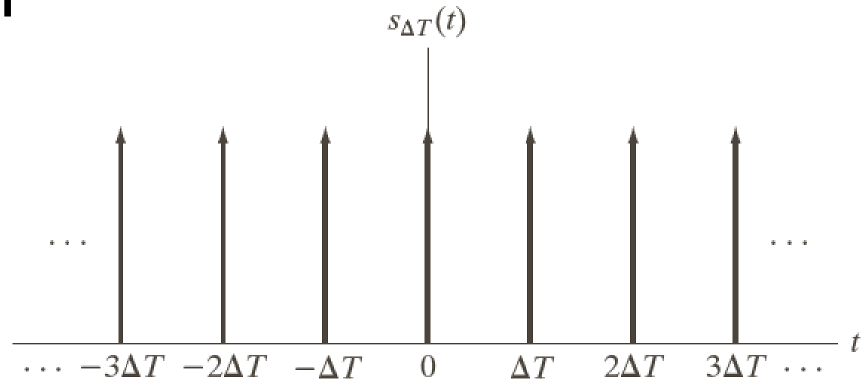
$$\delta(x - x_0) = 1, \text{αν } x = x_0$$

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

1D continuous signals (cont.)

Impulse train function

$$S_{\Delta T}(t) = \sum_{n=-\infty}^{+\infty} \delta(t - n\Delta T)$$



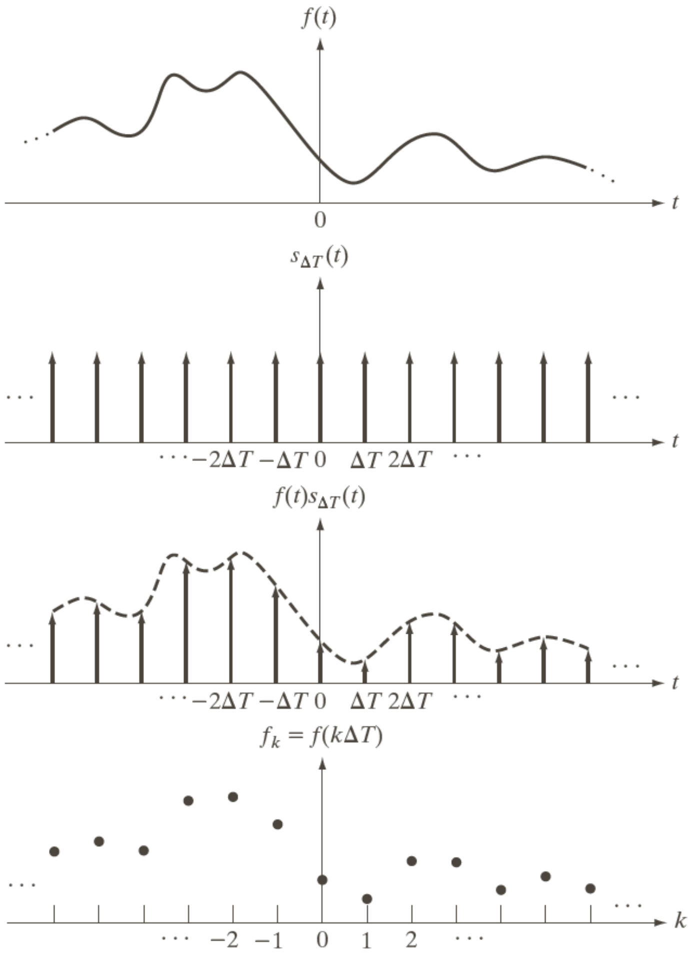
$$x[n] = x(t)S_{\Delta T}(t) = \sum_{n=-\infty}^{+\infty} x(t)\delta(t - n\Delta T) = \sum_{n=-\infty}^{+\infty} x(n\Delta T)\delta(t - n\Delta T)$$

1D continuous signals (from continuous to discrete - sampling)

$$x[n] = x(t)S_{\Delta T}(t)$$

$$= \sum_{n=-\infty}^{+\infty} x(t)\delta(t - n\Delta T)$$

$$= \sum_{n=-\infty}^{+\infty} x(n\Delta T)\delta(t - n\Delta T)$$



Continuous Fourier Transform of a discrete signal

- After expressing the sampling as

$$\widetilde{f}(t) = f(t)s_{\Delta T}(t)$$

- We compute the Fourier Transform of the discrete signal.

$$\widetilde{F}(\mu) = F(\mu) * S(\mu) = \int_{-\infty}^{\infty} F(\tau)S(\mu - \tau)d\tau$$

- It can be proven that the **Fourier transform** of the **periodic impulse train**, is:

$$S(\mu) = \Im\{s_{\Delta T}(t)\} = \Im\left\{\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}\right\} = \frac{1}{\Delta T}\Im\left\{\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}\right\} = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\delta\left(\mu - \frac{n}{\Delta T}\right)$$

Continuous Fourier Transform of a discrete signal

- We compute the Fourier Transform of the discrete signal.

$$\widetilde{F}(\mu) = F(\mu) * S(\mu) = \int_{-\infty}^{\infty} F(\tau) S(\mu - \tau) d\tau =$$

$$S(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(\mu - \frac{n}{\Delta T}\right)$$

$$= \frac{1}{\Delta \mathrm{T}} \int_{-\infty}^{\infty} F(\tau) \sum_{n=-\infty}^{\infty} \delta(\mu - \tau - \frac{n}{\Delta \mathrm{T}}) d\tau =$$

$$= \frac{1}{\Delta \mathrm{T}} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} F(\tau) \delta(\mu - \tau - \frac{n}{\Delta \mathrm{T}}) d\tau =$$

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

$$= \frac{1}{\Delta \mathrm{T}} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta \mathrm{T}})$$

1D continuous signals (cont.)

- Intermediate result
  - The Fourier transform of the impulse train.

$$\sum_{n=-\infty}^{+\infty} \delta(t - n\Delta T) \leftrightarrow \frac{1}{\Delta T} \sum_{n=-\infty}^{+\infty} \delta\left(\mu - \frac{n}{\Delta T}\right)$$

- It is also an impulse train in the frequency domain.
- Impulses are equally spaced every $1/\Delta T$.

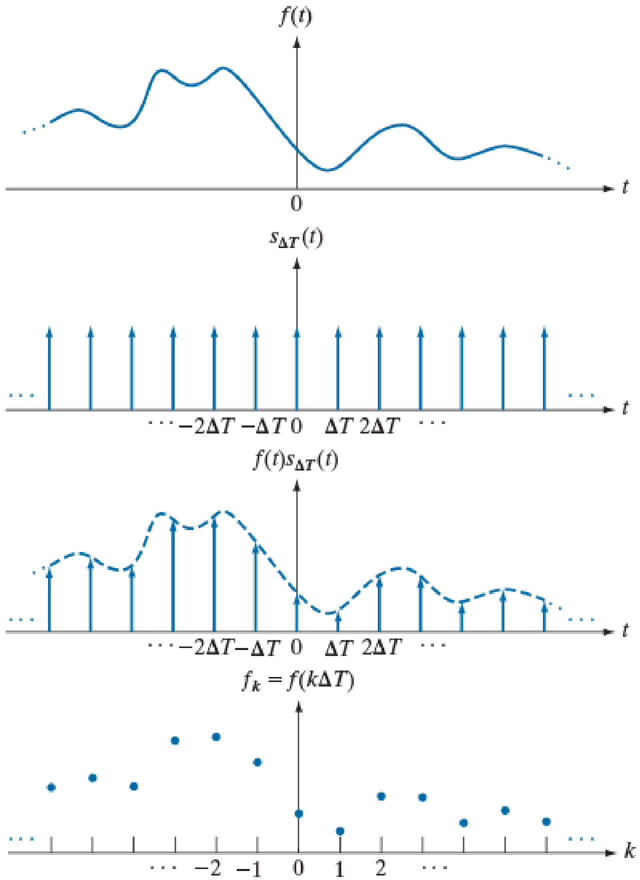
1D continuous signals (cont.)

Sampling

$$x[n] = x(t)S_{\Delta T}(t)$$

$$= \sum_{n=-\infty}^{+\infty} x(t)\delta(t - n\Delta T)$$

$$= \sum_{n=-\infty}^{+\infty} x(n\Delta T)\delta(t - n\Delta T)$$



1D continuous signals (cont.)

- Sampling
  - The spectrum of the discrete signal consists of repetitions of the spectrum of the continuous signal every $1/\Delta T$.
  - The Nyquist criterion should be satisfied.

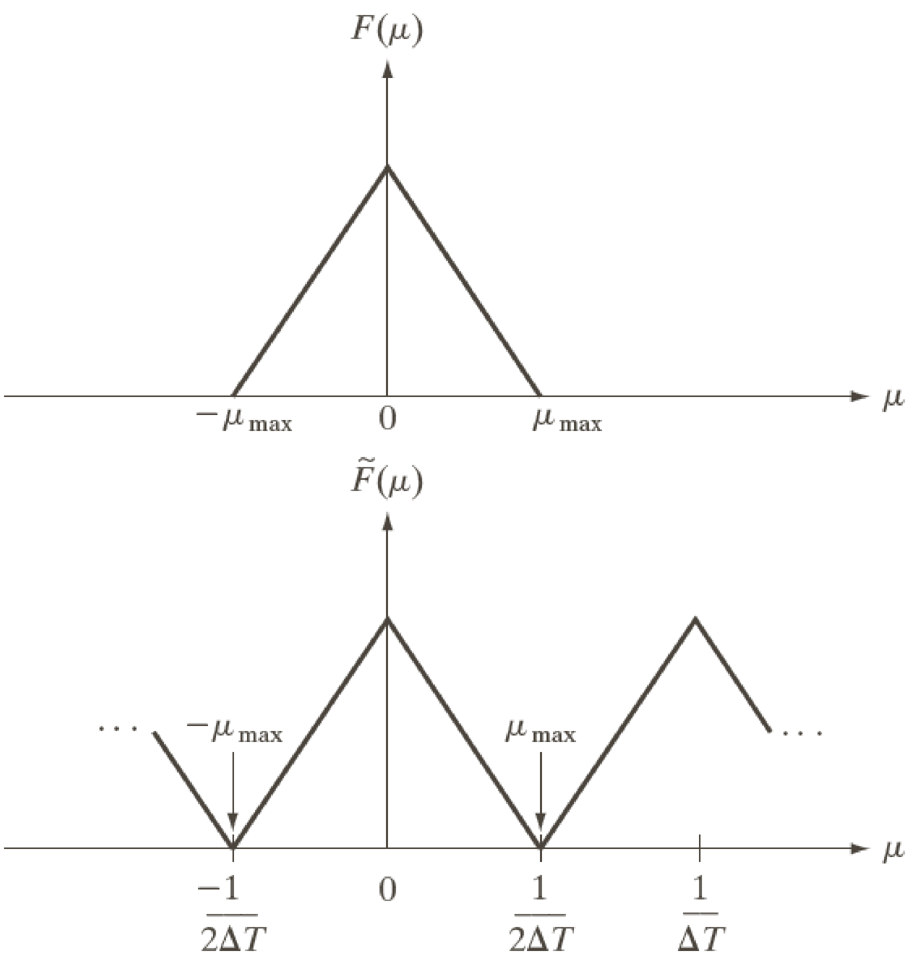
$$f(t) \leftrightarrow F(\mu)$$

$$\tilde{f}(n\Delta T) = f[n] \leftrightarrow \tilde{F}(\mu) = \frac{1}{\Delta T}\sum_{n=-\infty}^{+\infty} F\left(\mu - \frac{n}{\Delta T}\right)$$

1D continuous signals (cont.)

Nyquist theorem

$$\frac{1}{\Delta T} \geq 2\mu_{\max}$$



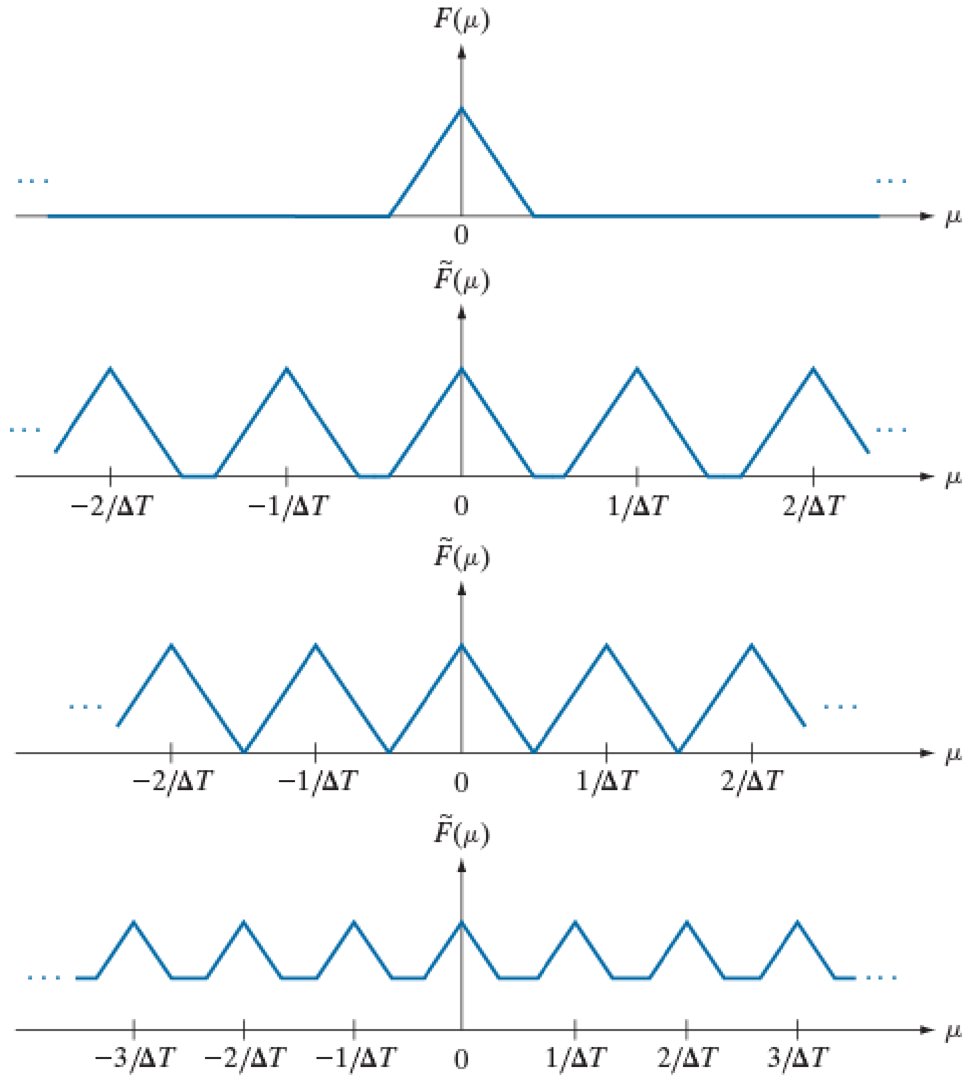
1D continuous signals (cont.)

FT of a continuous signal

Oversampling

Critical sampling with the Nyquist frequency

Undersampling
Aliasing appears



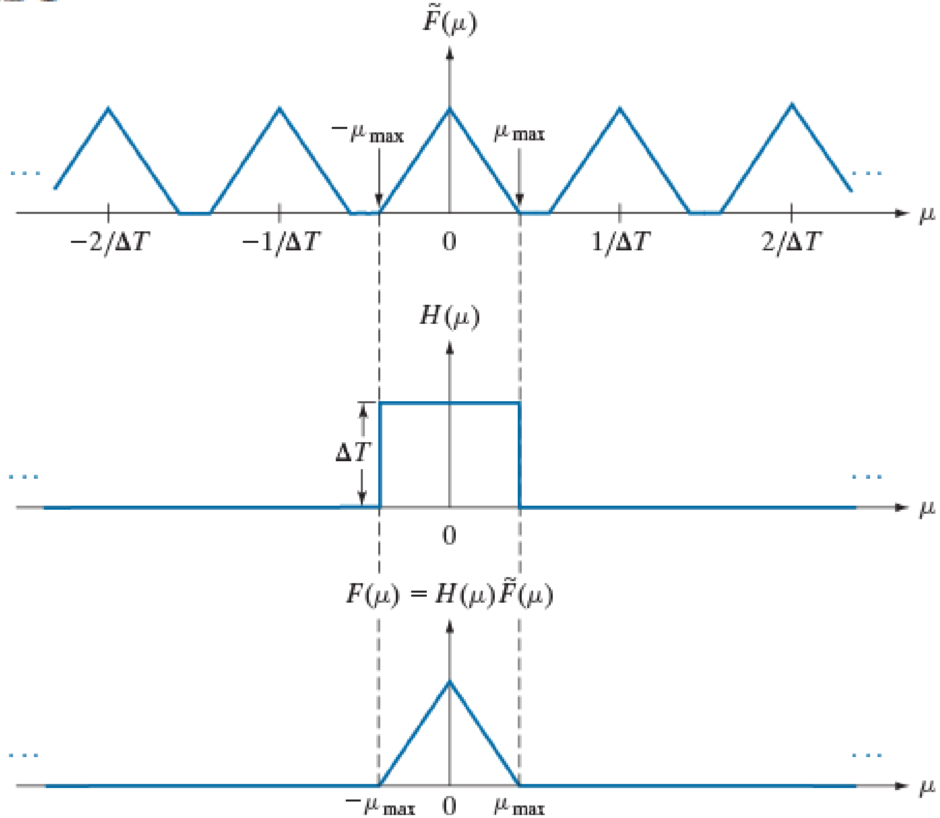
Reconstruction (correct sampling)

$$H(\mu) = \begin{cases} \Delta T & -\mu_{\max} \leq \mu \leq \mu_{\max} \\ 0 & \text{otherwise} \end{cases}$$

$$F(\mu) = \tilde{F}(\mu)H(\mu)$$

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}d\mu$$

$$f(t) = \tilde{f}(t) * T\sin c\left(\frac{t}{\Delta T}\right)$$



1D continuous signals (cont.)

- Reconstruction
  - Provided a correct sampling, the continuous signal may be perfectly reconstructed by its samples using inverse transform:

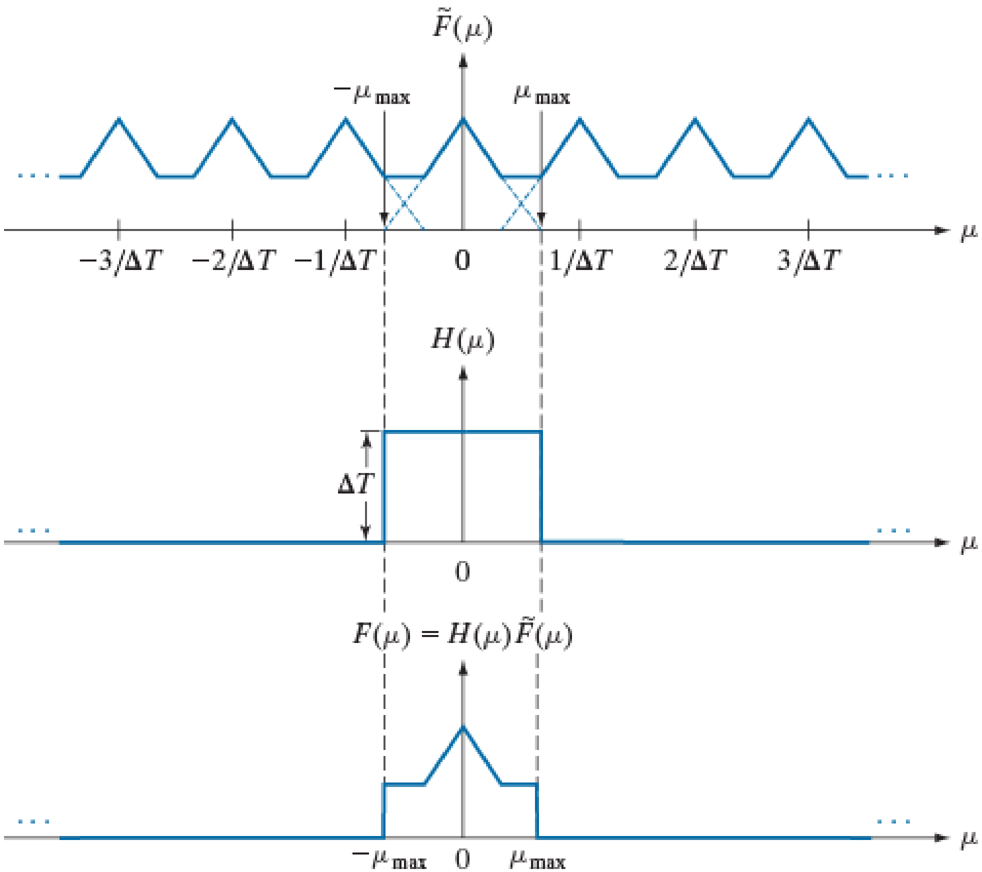
$$f(t) = \Im^{-1}\{F(\mu)\} = \Im^{-1}\left\{H(\mu)\tilde{F}(\mu)\right\}$$

$$= h(t) \star \tilde{f}(t)$$

$$f(t) = \sum_{n=-\infty}^{+\infty} f(n\Delta T)\ \mathrm{sinc}\left[\frac{(t-n\Delta T)}{n\Delta T}\right]$$

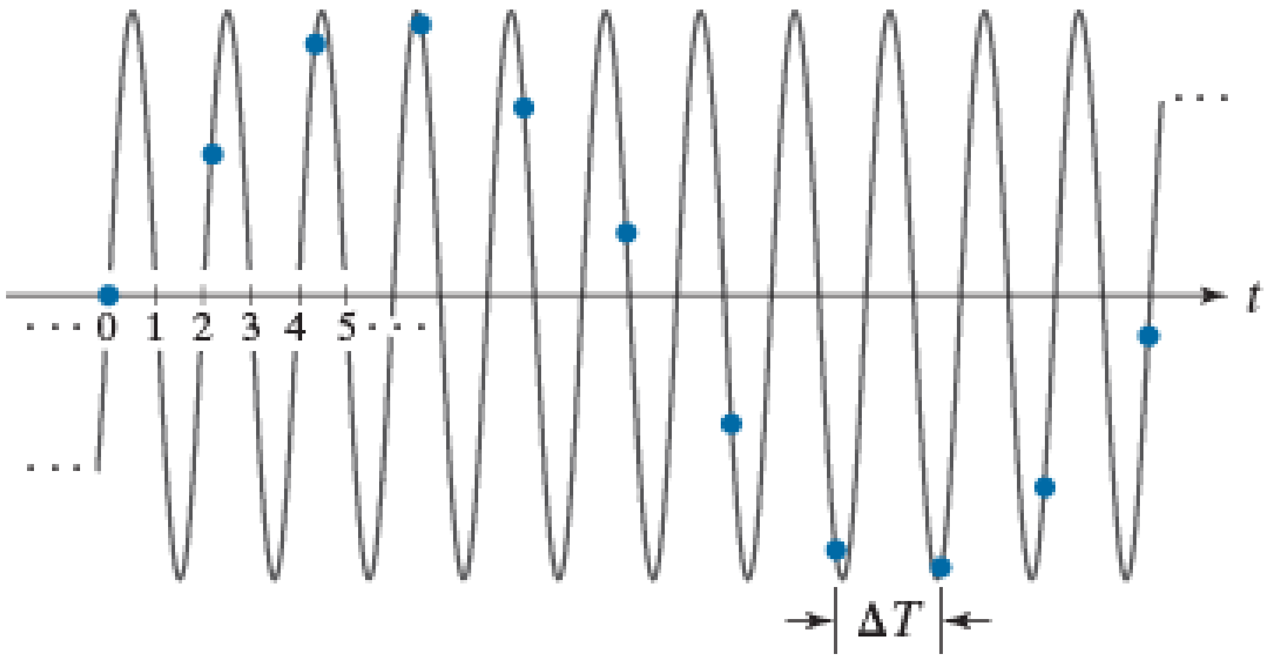
1D continuous signals (cont.)

- Aliasing, the reconstruction of the continuous signal is not correct.



1D continuous signals (cont.)

Aliased signal



The Discrete Fourier Transform

- The Fourier transform of a sampled (discrete) signal is a continuous function of the frequency.

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{+\infty} F\left(\mu - \frac{n}{\Delta T}\right)$$

- For a N-length discrete signal, taking N samples of its Fourier transform at frequencies:

$$\mu_k = \frac{k}{N\Delta T},\ k = 0, 1, .., N-1$$

provides the discrete Fourier transform (DFT) of the signal.

The Discrete Fourier Transform (cont..)

- This is possible since:

$$\widetilde{F}(\mu) = \int_{-\infty}^{\infty} \widetilde{f}(t) e^{-j2\pi\mu t} dt =$$

$$= \int_{-\infty}^{\infty} \sum_{-\infty}^{\infty} f(t)\delta(t - n\Delta\mathrm{T}) e^{-j2\pi\mu t} dt =$$

$$= \sum_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\delta(t - n\Delta\mathrm{T}) e^{-j2\pi\mu t} dt =$$

$$= \sum_{n=-\infty}^{\infty} f_n e^{-j2\pi\mu n\Delta\mathrm{T}}$$

$$\mu_k = \frac{k}{N\Delta T}, \; k = 0, 1, .., N-1$$

The Discrete Fourier Transform (cont.)

- DFT pair of signal $f[n]$ of length N.

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-j\frac{2\pi nk}{N}}, \qquad 0 \le k \le N-1$$

$$f[n] = \frac{1}{N} \sum_{n=0}^{N-1} F[k] e^{j\frac{2\pi nk}{N}}, \qquad 0 \le n \le N-1$$

The Discrete Fourier Transform (cont.)

- Property
  - The DFT of a N-length $f[n]$ signal is periodic with period N.

$$F[k+N]=F[k]$$

- This is due to the periodicity of the complex exponential:

$$F[k]=\sum_{n=0}^{N-1} f[n]e^{-j\frac{2\pi nk}{N}}$$

$$e^{-j\frac{2\pi n(k+N)}{N}}=e^{-j\frac{2\pi nk}{N}-j2\pi n}=e^{-j\frac{2\pi nk}{N}}e^{-j2\pi n}=e^{-j\frac{2\pi nk}{N}}$$

The Discrete Fourier Transform (cont.)

- DFT of a N-length $f[n]$ signal is periodic with period N.

$$F[k+N] = \sum_{n=0}^{N-1} f[n] \cdot e^{-j\frac{2\pi kn}{N}} \cdot e^{-j2\pi n}$$

$e^{-j2\pi n}$ is periodic with period N. So we have:

$$F[k+N] = F[k]$$

- Which can be simplified assuming:

$$\left(w_N\right)^n = \left(e^{-j\frac{2\pi}{N}}\right)^n \Leftrightarrow w_N^n = e^{-j\frac{2\pi n}{N}}$$

The Discrete Fourier Transform (cont.)

- DFT pair of signal $f[n]$ of length N may be expressed in matrix-vector form.

$$F[k] = \sum_{n=0}^{N-1} f[n] w_N^{nk}, \qquad 0 \le k \le N-1$$

$$f[n] = \frac{1}{N} \sum_{n=0}^{N-1} F[k] w_N^{-nk}, \qquad 0 \le n \le N-1$$

$$w_N = e^{-j\frac{2\pi}{N}}$$

The Discrete Fourier Transform (cont.)

$$\mathbf{F} = \mathbf{A}\mathbf{f} \qquad w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$$

$$\mathbf{A} = \begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \cdots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \cdots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \cdots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}$$

$$\mathbf{f} = \left[f[0], f[1], \ldots, f[N-1]\right]^T, \ \mathbf{F} = \left[F[0], F[1], \ldots, F[N-1]\right]^T$$

The Discrete Fourier Transform (cont.)

Example for N=4 (for $k = 0, e^{-j\frac{2\pi(0)n}{N}} = e^0 = 1$)

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

f[n] = [1, 2, 3, 4]

$$F = [10,*,*,*]$$

The Discrete Fourier Transform (cont.)

The inverse DFT is then expressed by:

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{F}$$

$$\mathbf{A}^{-1} = \frac{1}{N}\left(\mathbf{A}^{*}\right)^{T} = \frac{1}{N}\left(\begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \cdots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \cdots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \cdots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}^{*}\right)^{T}$$

This is derived by the complex exponential sum property.

Also, since A is symmetric: $A^{-1} = \frac{1}{\mathrm{N}} A^{*}$

Linear convolution

$$f[n]=\{\underline{1},\ 2,\ 2\},\ \ h[n]=\{\underline{1},\ -1\},\ \ N_1=3, N_2=2$$

$$g[n]=f[n]*h[n]=\sum_{m=-\infty}^{+\infty} f[m]h[n-m]$$

is of length $N=N_1+N_2-1=4$

Linear convolution (cont.)

$$f[n]=\{\underline{1},\ 2,\ 2\},\ \ h[n]=\{\underline{1},\ -1\},\ \ N_1=3, N_2=2$$

$$g[n]=f[n]*h[n]=\sum_{m=-\infty}^{+\infty} f[m]h[n-m]$$

	$f[m]$		1	2	2			$g[n]$
$n=0$	$h[0-m]$	-1	1				$\rightarrow$	1
$n=1$	$h[1-m]$		-1	1			$\rightarrow$	1
$n=2$	$h[2-m]$			-1	1		$\rightarrow$	0
$n=3$	$h[3-m]$				-1	1	$\rightarrow$	-2

$$g[n]=\{\underline{1},\ 1,\ 0,\ -2\}$$

Circular shift

- Signal $x[n]$ of length N.
- A circular shift ensures that the resulting signal will keep its length N.
- It is a shift modulo N denoted by

$$x[(n-m)_N] = x[(n-m) \bmod N]$$

- Example: $x[n]$ is of length N=8.

$$x[(-2)_N] = x[(-2)_8] = x[6]$$

$$x[(10)_N] = x[(10)_8] = x[2]$$

Circular convolution

$$f[n] = \{\underline{1},\ 2,\ 2\},\ \ h[n] = \{\underline{1},\ -1\},\ \ N_1 = 3, N_2 = 2$$

$$g[n] = f[n] \circledast h[n] = \sum_{m=-\infty}^{+\infty} f[m] h[(n-m)_N]$$

Circular shift modulo N

The result is of length $N = \max\{N_1, N_2\} = 3$

Circular convolution (cont.)

$$f[n]=\{\underline{1},\ 2,\ 2\},\ \ h[n]=\{\underline{1},\ -1\},\ \ N_1=3, N_2=2$$

$$g[n]=f[n]✪h[n]=\sum_{m=-\infty}^{+\infty} f[m]h[(n-m)_N]$$

	$f[m]$		1	2	2	$g[n]$
$n=0$	$h[(0-m)_N]$	-1	1		-1	-1
$n=1$	$h[(1-m)_N]$		-1	1		1
$n=2$	$h[(2-m)_N]$			-1	1	0

$$g[n]=\{-\underline{1},\ 1,\ 0,\}$$

DFT and convolution

$$g[n] = f[n] \circledast h[n] \leftrightarrow G[k] = F[k]H[k]$$

- The property holds for the circular convolution.
- In signal processing we are interested in linear convolution.
- Is there a similar property for the linear convolution?

DFT and convolution (cont.)

$$g[n]=f[n]\circledast h[n] \leftrightarrow G[k]=F[k]H[k]$$

- Let $f[n]$ be of length N_1 and $h[n]$ be of length N_2.
- Then $g[n]=f[n]*h[n]$ is of length N_1+N_2-1.
- If the signals are zero-padded to length $N=N_1+N_2-1$ then their circular convolution will be the same as their linear convolution:

$$\tilde{g}[n]=\tilde{f}[n]*\tilde{h}[n] \leftrightarrow \tilde{G}[k]=\tilde{F}[k]\tilde{H}[k]$$

Zero-padded signals

DFT and convolution (cont.)

$$f[n]=\{\underline{1},\ 2,\ 2\},\ \ h[n]=\{\underline{1},\ -1\},\ \ N_1=3, N_2=2$$

Zero-padding to length $N=N_1+N_2\text{-}1=4$

$$\tilde{f}[n]=\{\underline{1},\ 2,\ 2,\ 0\},\ \ \tilde{h}[n]=\{\underline{1},\ -1,\ 0,\ 0\}$$

$f[m]$				1	2	2	0	$g[n]$
$h[(n-0)_4]$	0	0	-1	1	0	0	-1	1
$h[(n-1)_4]$		0	0	-1	1	0	0	1
$h[(n-2)_4]$			0	0	-1	1	0	0
$h[(n-3)_4]$				0	0	-1	1	-2

The result is the same as the linear convolution.

DFT and convolution (cont.)

Verification using DFT

$$\tilde{\mathbf{F}} = \mathbf{A}\tilde{\mathbf{f}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ -1-j2 \\ 1 \\ -1+j2 \end{bmatrix}$$

$$\tilde{\mathbf{H}} = \mathbf{A}\tilde{\mathbf{h}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1+j \\ 2 \\ 1-j \end{bmatrix}$$

DFT and convolution (cont.)

$$\tilde{G}[k] = \tilde{F}[k]\tilde{H}[k]$$

Element-wise multiplication

$$\tilde{\mathbf{G}} = \tilde{\mathbf{F}} \times \tilde{\mathbf{H}} = \begin{bmatrix} 5 \times 0 \\ (-1 - j2) \times (1 + j) \\ 1 \times 2 \\ (-1 + j2) \times (1 - j) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 - j3 \\ 2 \\ 1 + j3 \end{bmatrix}$$

DFT and convolution (cont.)

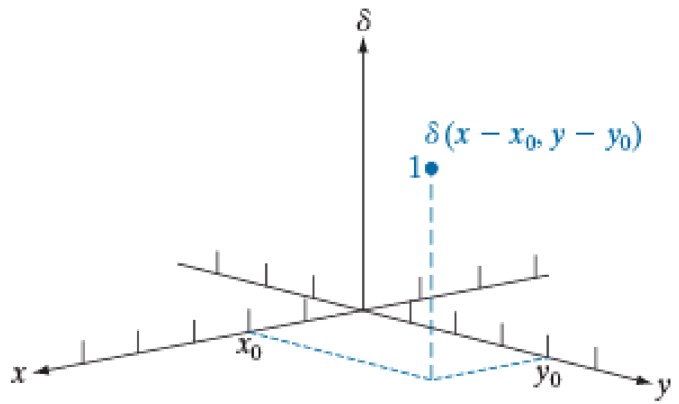
Inverse DFT of the result

$$\tilde{g} = A^{-1}\tilde{G} = \frac{1}{4}\left(A^{*}\right)^{\mathrm{T}}\tilde{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 0 \\ 1-3j \\ 2 \\ 1+3j \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -2 \end{bmatrix}$$

The same result as their linear convolution.

2D continuous signals

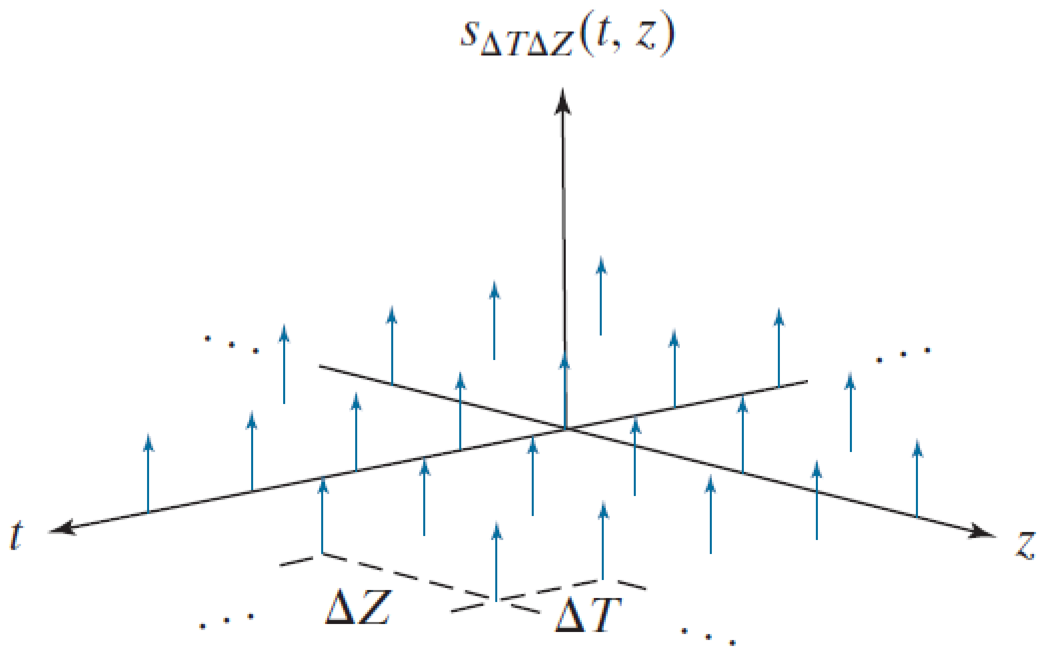
$$\delta(x-x_0, y-y_0) = \begin{cases} +\infty, & x = x_0, y = y_0 \\ 0 & \text{otherwise} \end{cases}$$



$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x-x_0, y-y_0)dydx = f(x_0, y_0) \quad \text{Selection prop.}$$

$$\delta(x-x_0, y-y_0) = \delta(x-x_0)\delta(y-y_0) \quad \text{seperability}$$

2D continuous signals (cont.)



The 2D impulse train is also separable:

$$\boldsymbol{S}_{\Delta X \Delta Y}(x, y) = \boldsymbol{S}_{\Delta X}(x)\boldsymbol{S}_{\Delta Y}(y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\Delta X, y - nDY)$$

2D continuous signals (cont.)

- The Fourier transform of a continuous 2D signal $f(x,y)$.

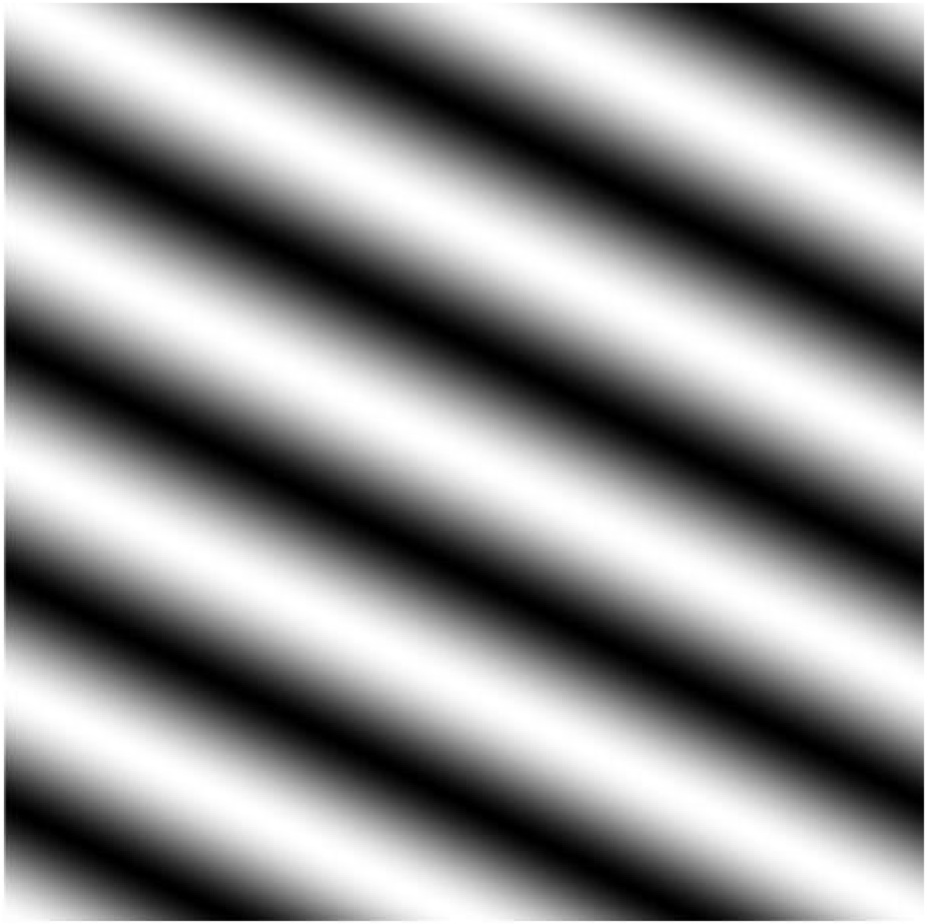
$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(\mu,\nu)e^{j2\pi(\mu x+\nu y)}d\nu d\mu$$

2D basis functions (example)

- Basis functions example of 2D continuous FT (real part)

$$e^{-j2\pi(\mu x+vy)}$$



2D basis functions (example)

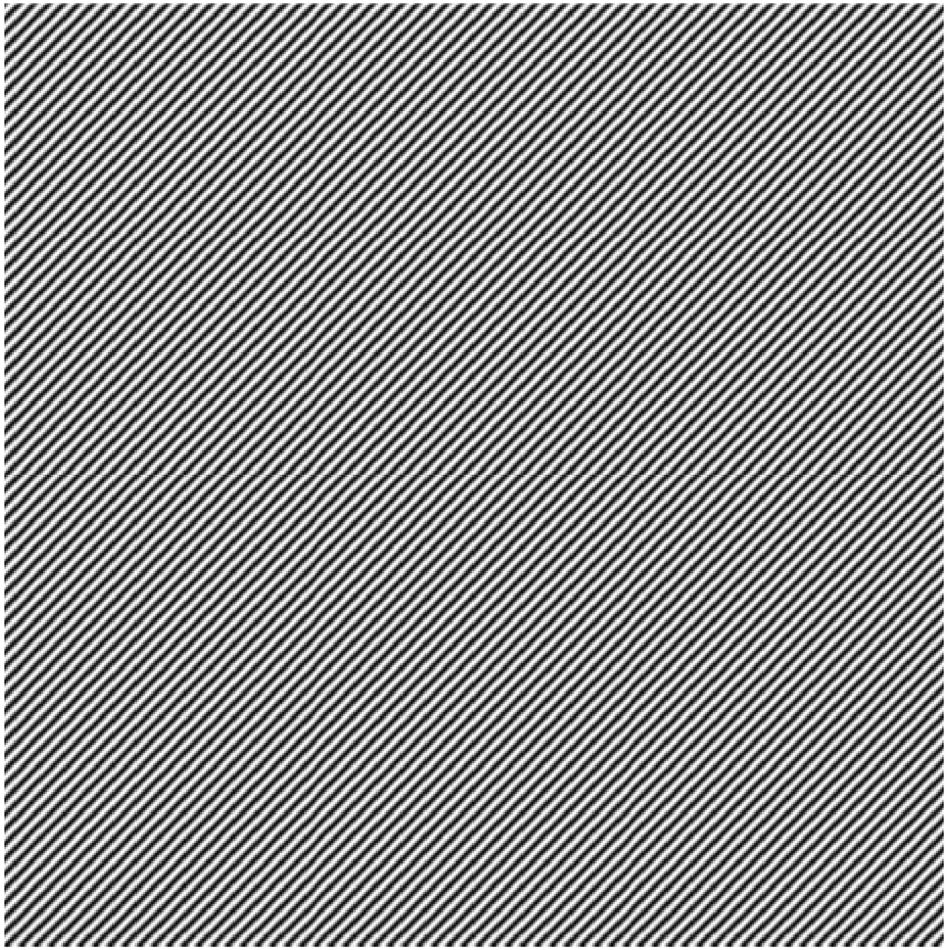
- Basis functions example of 2D continuous FT (real part)

$$e^{-j2\pi(\mu x+vy)}$$

2D basis functions (example)

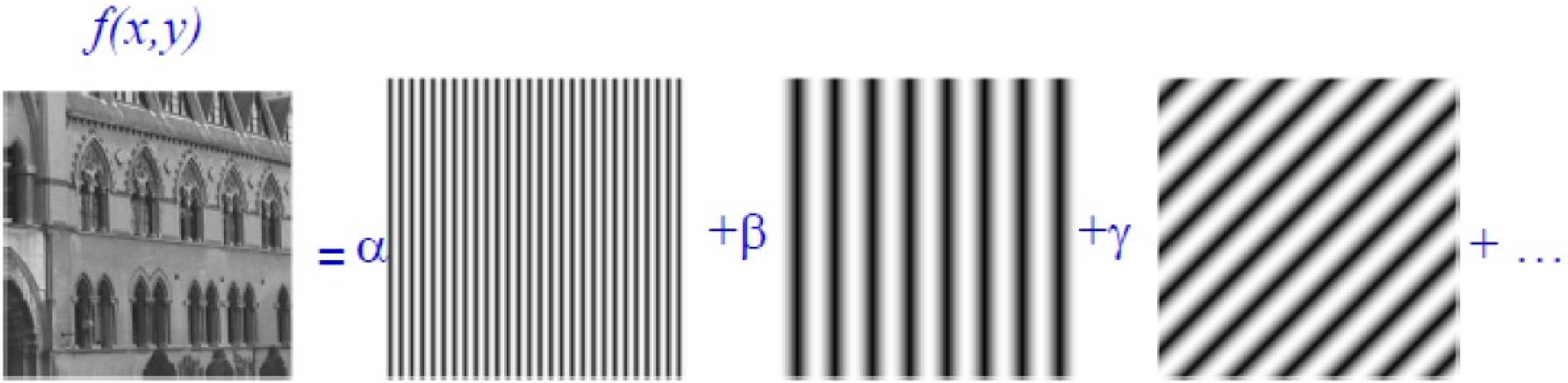
- Basis functions example of 2D continuous FT (real part)

$$e^{-j2\pi(\mu x+vy)}$$



2D basis functions (example)

- Basis functions example of 2D continuous FT (real part)



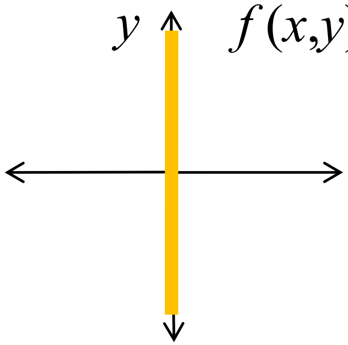
2D continuous signals (cont.)

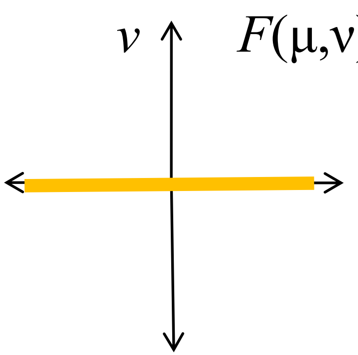
- Example: FT of $f(x,y)=\delta(x)$

$$F(\mu,v)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi(\mu x+vy)}dydx$$

$$=\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi\mu x}dx\int_{-\infty}^{+\infty}e^{-j2\pi vy}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi vy}dy \;=\delta(v)$$





2D continuous signals (cont.)

- Reminder

$$\Im\{\delta(t)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi\mu t} dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t} \delta(t)\, dt = e^{-j2\pi\mu 0}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t)\, dt = f(0)$$

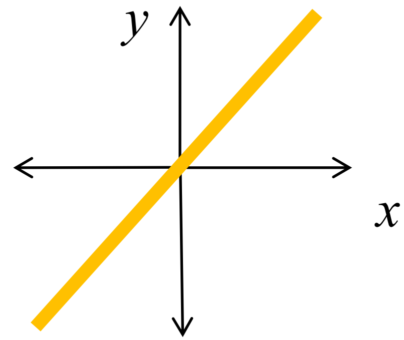
- Hence, the term that follows is ‘1’.

$$\int_{-\infty}^{+\infty} \delta(x) e^{-j2\pi\mu x} dx$$

2D continuous signals (cont.)

- Example: FT of $f(x,y)=\delta(x-y)$

$f(x,y)=\delta(x-y)$



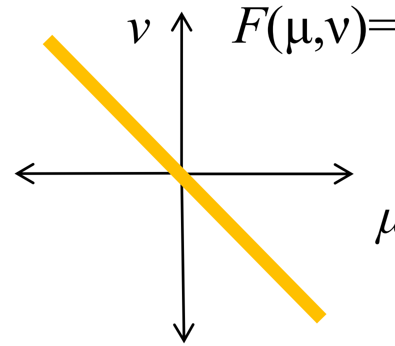
$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$=\int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi\mu x}dx\right]e^{-j2\pi\nu y}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi\mu y}e^{-j2\pi\nu y}dy=\int_{-\infty}^{+\infty}e^{-j2\pi(\mu+\nu)y}dy$$

$$=\delta(\mu+\nu)$$

$F(\mu,\nu)=\delta(\mu+\nu)$



2D continuous signals (cont.)

- Reminder

$$\Im\{\delta(t-t_0)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t-t_0)e^{-j2\pi\mu t}dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t}\,\delta(t-t_0)\,dt = e^{-j2\pi\mu t_0}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

- Hence, the term that follows is:

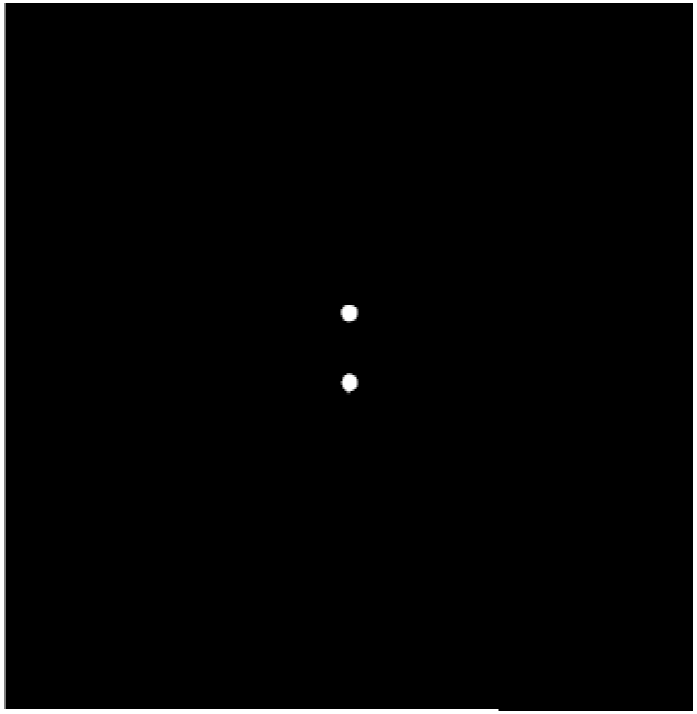
$$\left[\int_{-\infty}^{+\infty} \delta(x-y)e^{-j2\pi\mu x}dx\right] = e^{-j2\pi\mu y}$$

2D continuous signals (cont.)

- Example

$$f(x,y) = \delta(x, y-a) + \delta(x, y+a)$$

$f(x,y)$

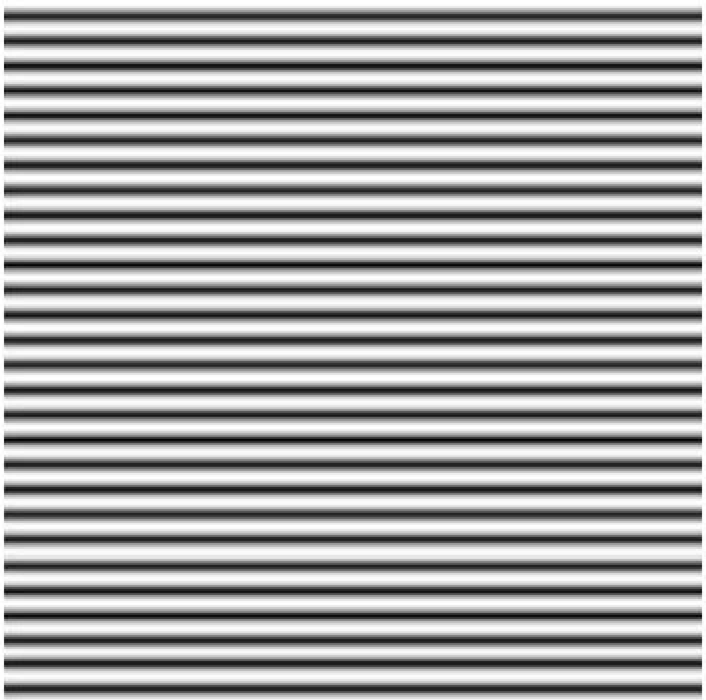


2D continuous signals (cont.)

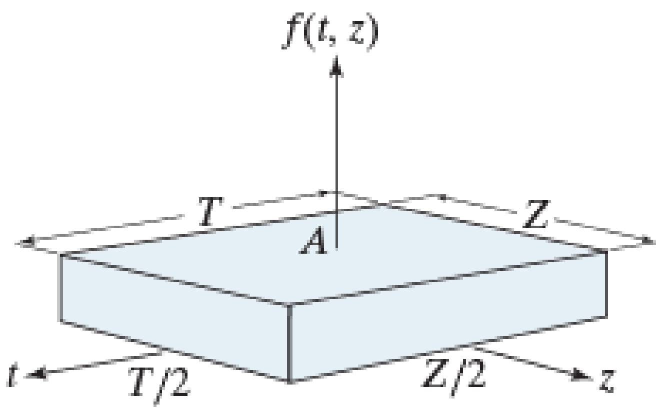
$$F(\mu, v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} [\delta(x, y-a) + \delta(x, y+a)] e^{-j2\pi(\mu+vy)} dydx$$

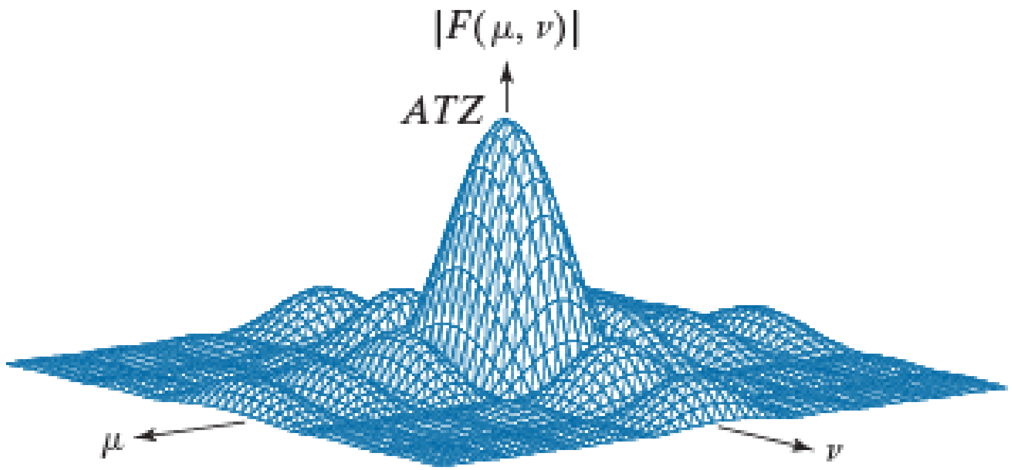
$$= e^{-j2\pi av} + e^{j2\pi av} = 2cos2\pi\alpha v$$

$F(\mu, v)$



2D continuous signals (cont.)





$$f(x,y) = \mathrm{A}\,P_{W/2,W/2}(x,y) \leftrightarrow F(\mu,\nu) = \mathrm{A}W^2 \frac{\sin(\pi\mu W)}{(\pi\mu W)} \frac{\sin(\pi\nu W)}{(\pi\nu W)}$$

2D continuous signals (cont.)

- 2D continuous convolution

$$f(x,y)*h(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-\alpha,y-\beta)h(\alpha,\beta)\,d\alpha\,d\beta$$

- We will examine the discrete convolution in more detail.
- Convolution property

$$f(x,y)*h(x,y) \leftrightarrow F(\mu,\nu)H(\mu,\nu)$$

2D continuous signals (cont.)

- 2D sampling is accomplished by

$$\boldsymbol{s}_{\Delta X \Delta Y}(x,y) = \boldsymbol{s}_{\Delta X}(x)\boldsymbol{s}_{\Delta Y}(y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(x - m\Delta X, y - nDY)$$

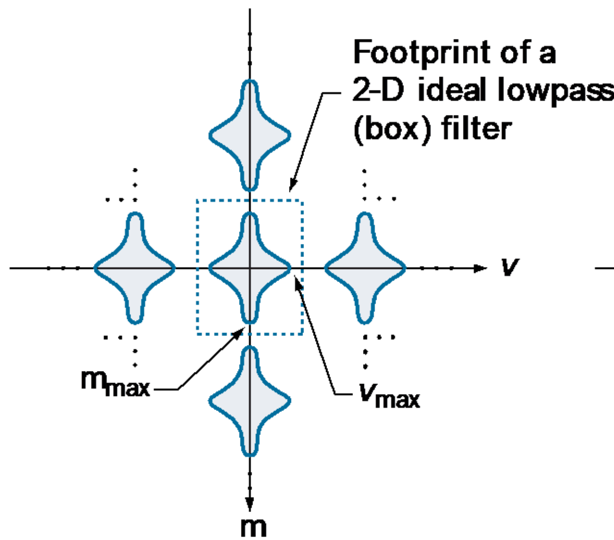
- The FT of the sampled 2D signal consists of repetitions of the spectrum of the 1D continuous signal.

$$\tilde{F}(\mu,\nu) = \frac{1}{\Delta X}\frac{1}{\Delta Y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\mathrm{F}(\mu - \frac{m}{\Delta X}, \nu - \frac{n}{\Delta X})$$

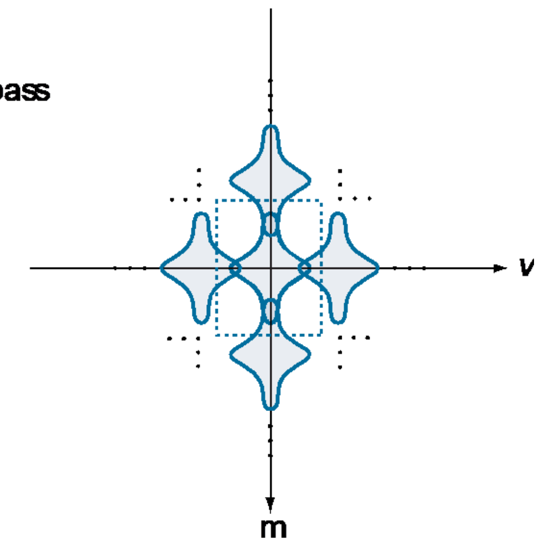
2D continuous signals (cont.)

- The Nyquist theorem involves both the horizontal and vertical frequencies.

$$\frac{1}{\Delta X} \geq 2\mu_{\max}, \quad \frac{1}{\Delta Y} \geq 2v_{\max}$$



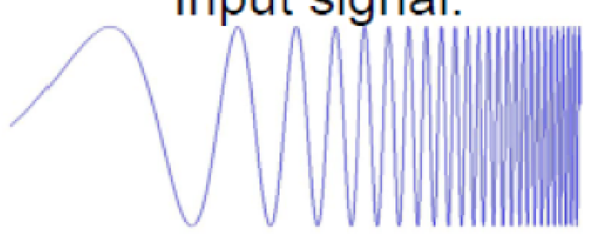
Over-sampled



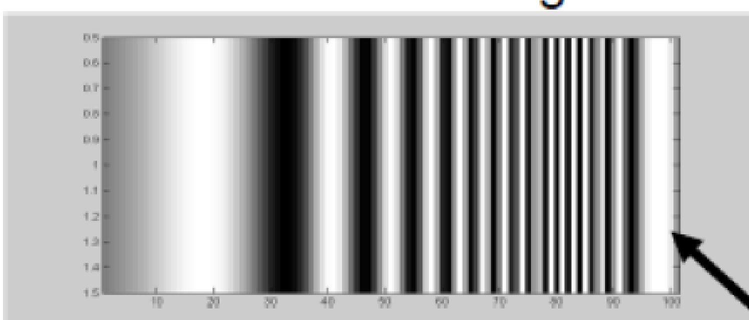
Under-sampled

Aliasing

Input signal:

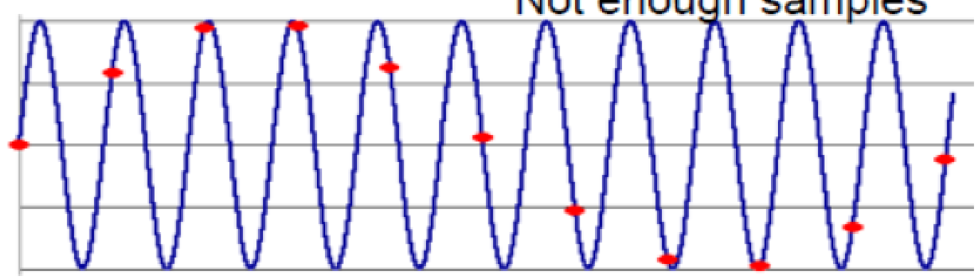


Plot as image:

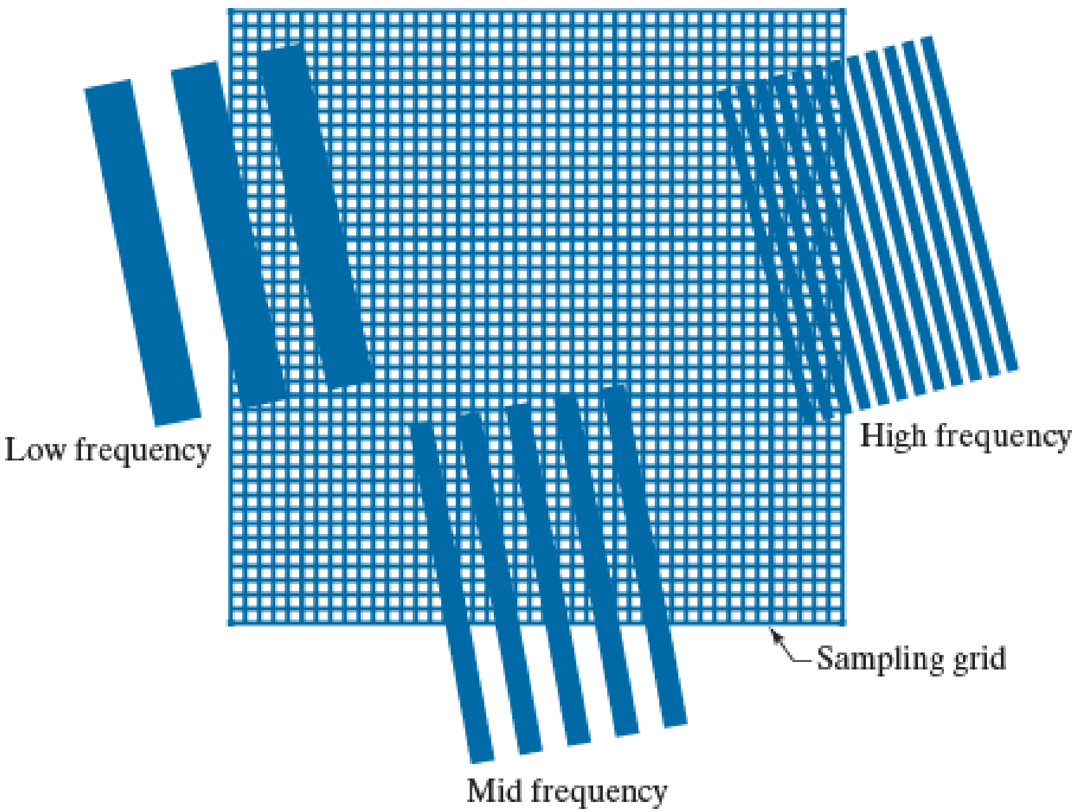


x = 0:.05:5; imagesc(sin((2.^x).*x))

Aliasing
Not enough samples



Aliasing



Aliasing and image resampling

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

a b c

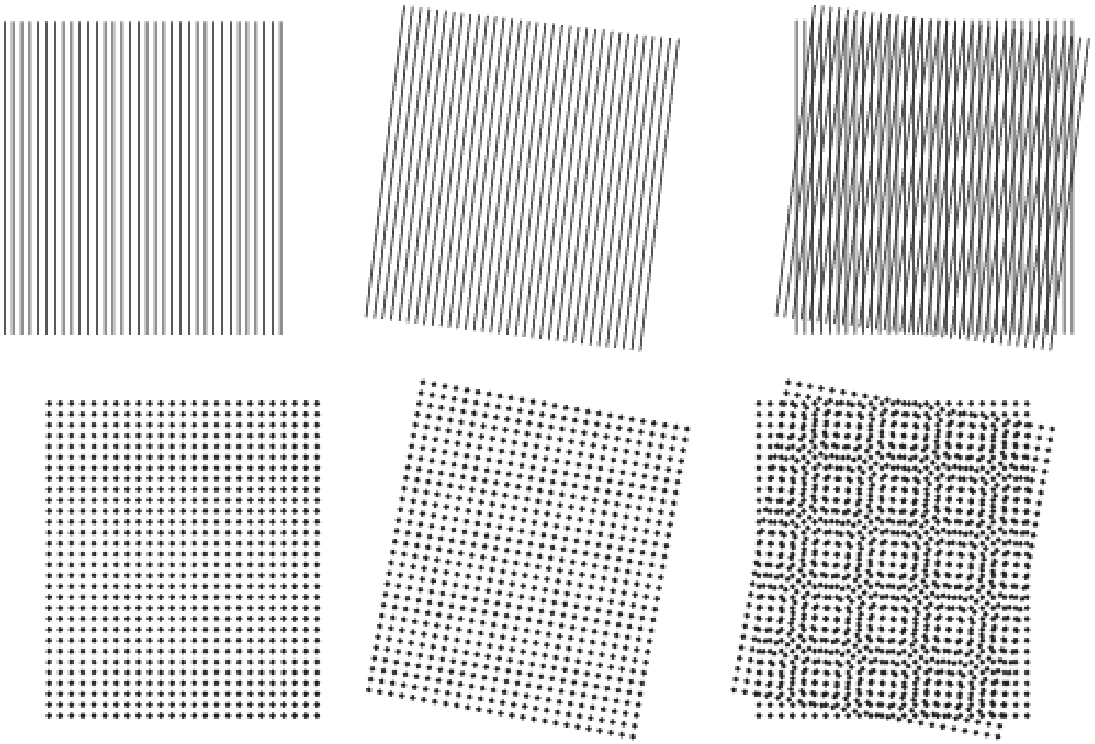
FIGURE 4.19 Illustration of aliasing on resampled natural images. (a) A digital image of size 772 × 548 pixels with visually negligible aliasing. (b) Result of resizing the image to 33% of its original size by pixel deletion and then restoring it to its original size by pixel replication. Aliasing is clearly visible. (c) Result of blurring the image in (a) with an averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

Aliasing - Moiré Patterns

- Effect of sampling a scene with periodic or nearly periodic components (e.g. overlapping grids, TV raster lines and stripped materials).
- In image processing the problem arises when scanning media prints (e.g. magazines, newspapers).
- The problem is more general than sampling artifacts.

Aliasing - Moiré Patterns (cont.)

- Superimposed grid drawings (not digitized) produce the effect of new frequencies not existing in the original components.



Aliasing - Moiré Patterns (cont.)

- In the printing industry the problem comes when scanning photographs from the superposition of:
  - The sampling lattice (usually horizontal and vertical).
  - Dot patterns on the newspaper image.

Aliasing - Moiré Patterns (cont.)

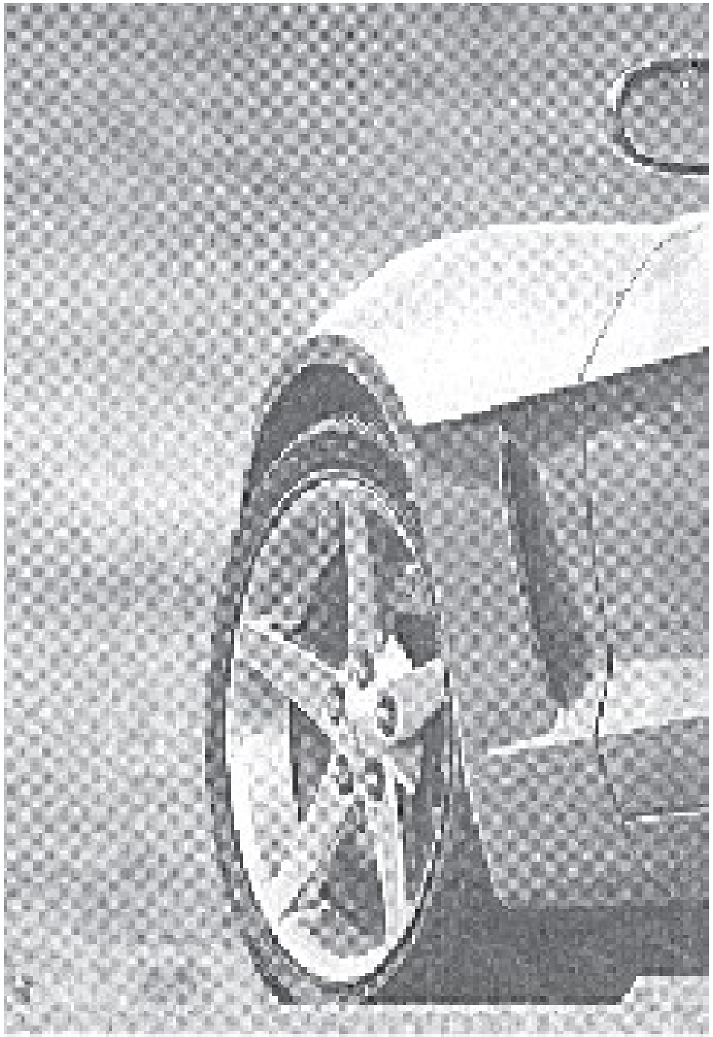
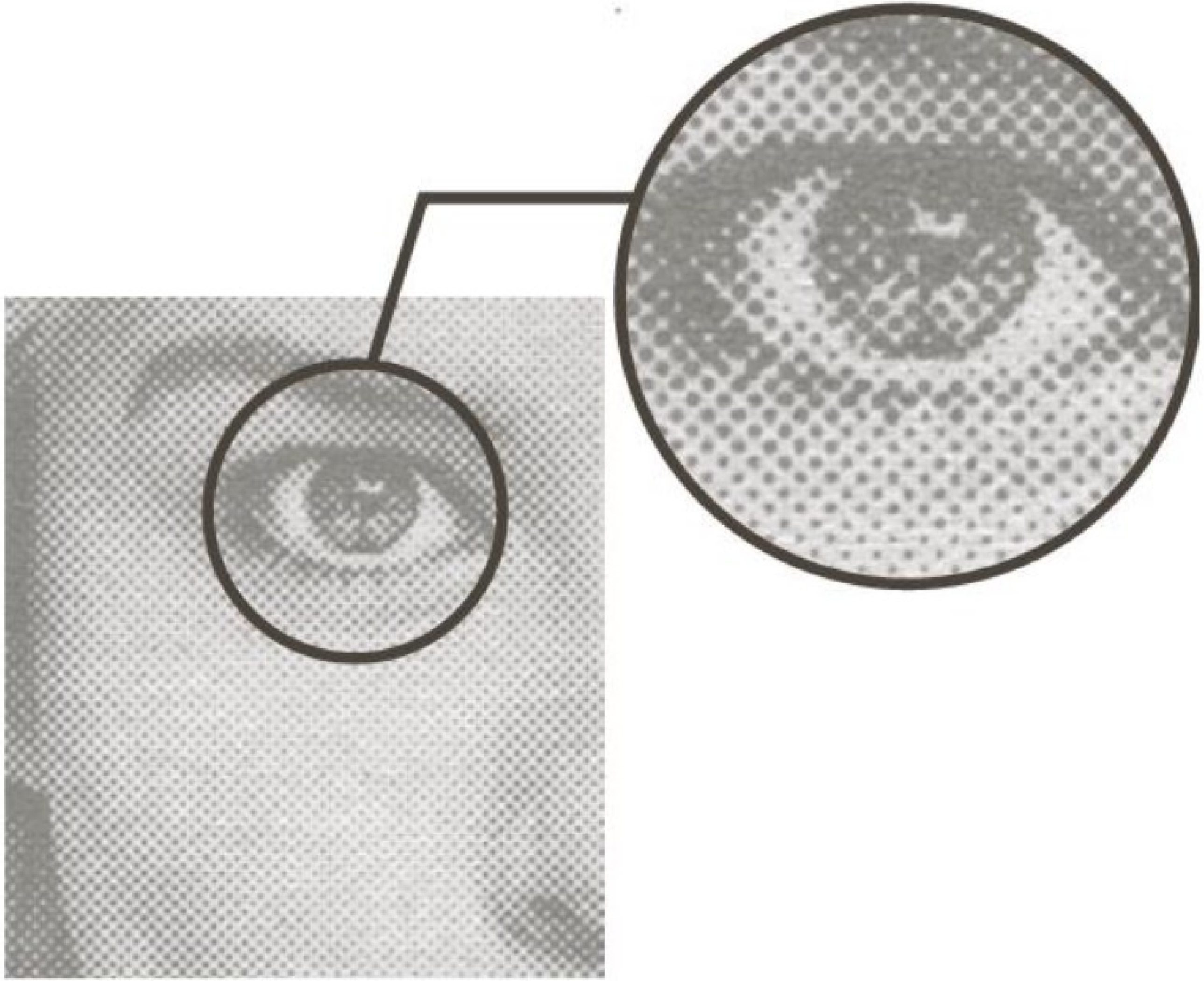


FIGURE 4.21 A newspaper image of size 246 × 168 pixels sampled at 75 dpi showing a moiré pattern. The moiré pattern in this image is the interference pattern created between the ±45° orientation of the halftone dots and the north–south orientation of the sampling grid used to digitize the image.

Aliasing - Moiré Patterns (cont.)



2D Discrete Fourier Transform (2D DFT)

- 2D DFT pair of image $f[m,n]$ of size MxN.

$$F[k,l]=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f[m,n]e^{-j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$f[m,n]=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F[k,l]e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$\begin{cases}0\le k\le M-1\\ 0\le l\le N-1\end{cases},\qquad \begin{cases}0\le m\le M-1\\ 0\le n\le N-1\end{cases}$$

2D Discrete Fourier Transform (2D DFT)

Separability of the 2D DFT:

- We can express the 2D DFT as two 1D DFTs:
- First, perform a 1D DFT along the columns and then along the rows (or vice versa).

2D Discrete Fourier Transform (2D DFT)

2D DFT can be represented in matrix form:

- Reminder for 1D: $\mathbf{F} = \mathbf{A}\mathbf{f}$ $\quad w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$

$$\mathbf{A} = \begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \cdots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \cdots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \cdots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}$$

2D Discrete Fourier Transform (2D DFT)

2D DFT can be represented in matrix form:

- In a similar fashion for 2D we employ the same matrix A:

$$F = \mathrm{A} f \mathrm{A}^{\mathrm{T}}$$

- Where now *F*, *f* are now NxN matrices:
  - Equivalent: $F = AfA$, since $A = A^T$

The 2D DFT (cont.)

- All of the properties of 1D DFT hold.
- Particularly:
  - Let $f[m,n]$ be of size M_1xN_1 and $h[m,n]$ of size M_2xN_2 .
  - If the signals are zero-padded to size (M_1+M_2-1)x(N_1+N_2-1) then their circular convolution will be the same as their linear convolution and:

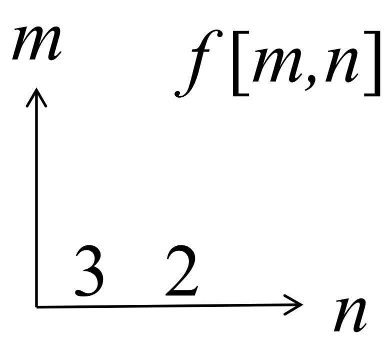
$$\tilde{g}[m,n] = \tilde{f}[m,n] * \tilde{h}[m,n] \leftrightarrow \tilde{G}[k,l] = \tilde{F}[k,l]\tilde{H}[k,l]$$

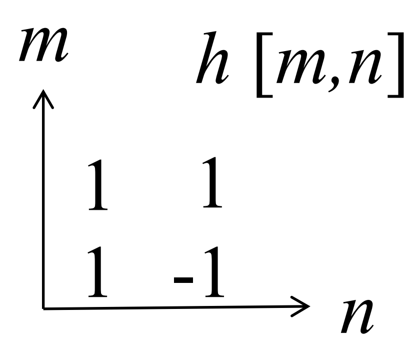
Fast Fourier Transform (FFT)

- Another reason the Fourier Transform is used in Digital Image Processing is the Fast Fourier Transform (FFT) algorithm.

- Reduces the complexity from $O(\,N^4)\ to$ $O(N^2 \log N^2)$

2D discrete convolution

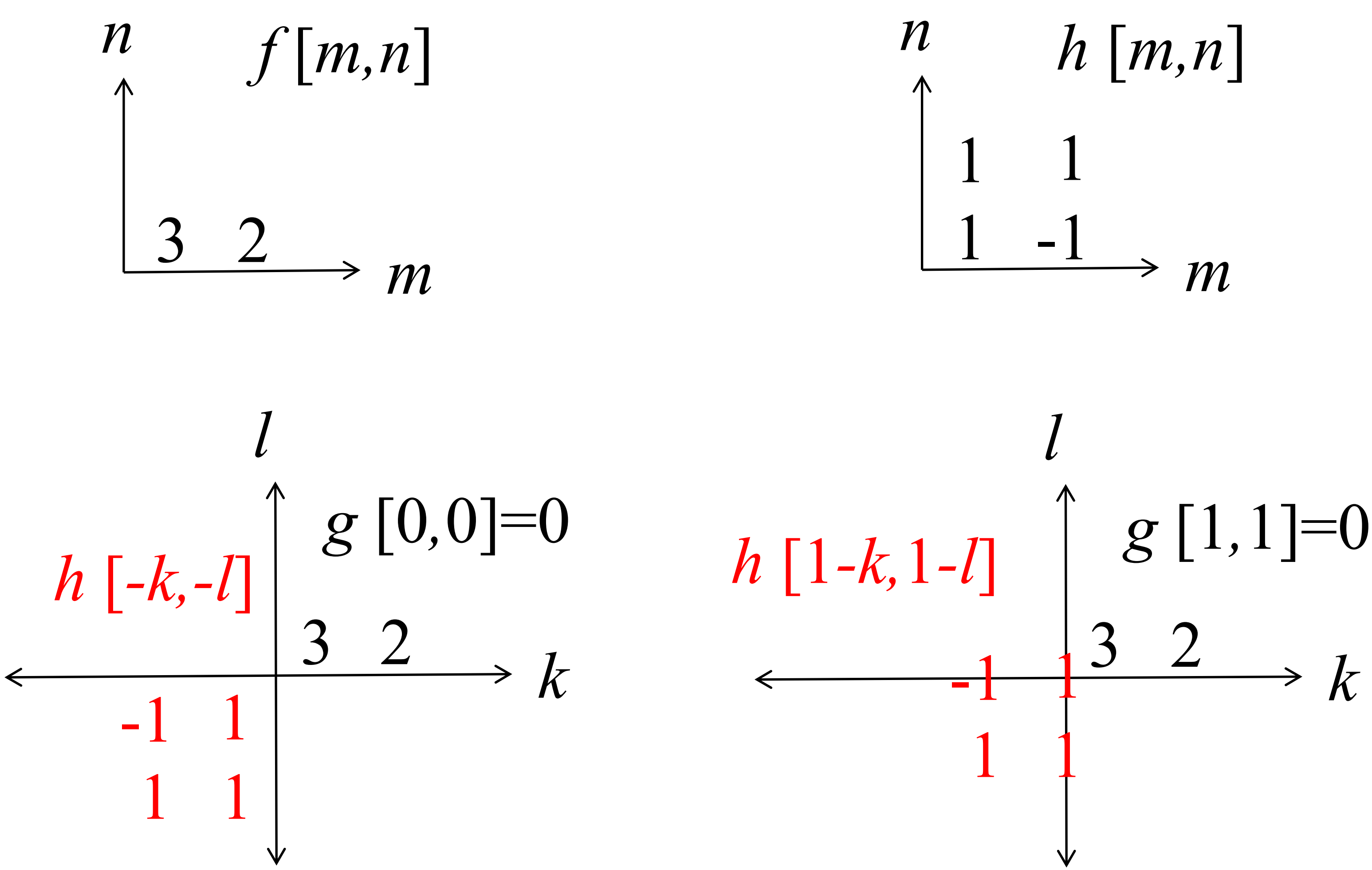




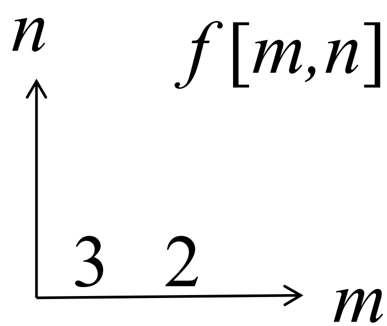
$$g[m,n] = f[m,n] * h[m,n] = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f[k,l]h[m-k,n-l]$$

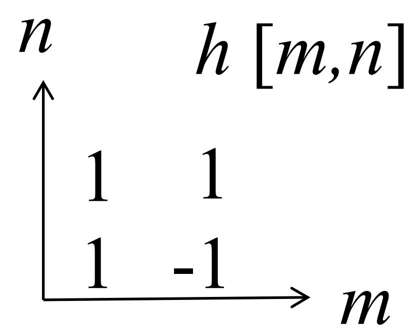
- Take the symmetric of one of the signals with respect to the origin.
- Shift it and compute the sum at every position $[m,n]$.

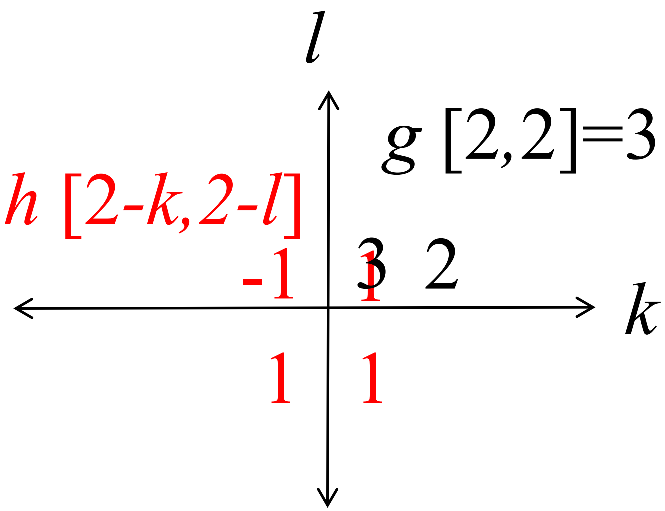
2D discrete convolution (cont.)

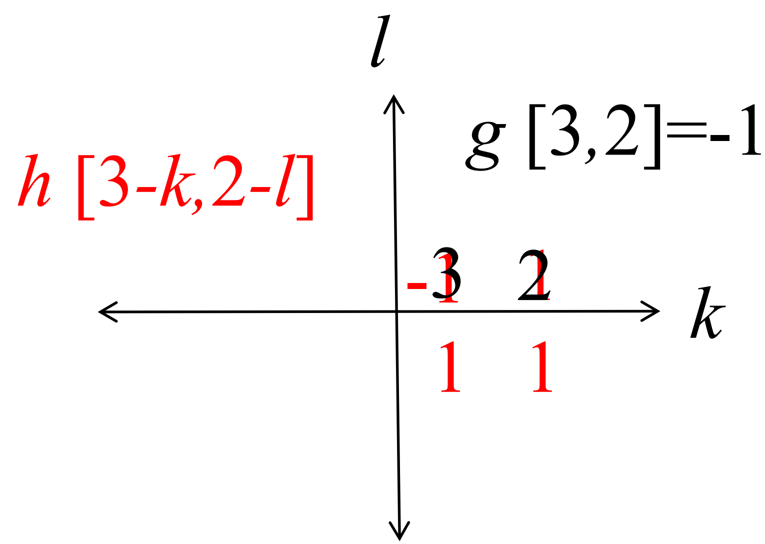


2D discrete convolution (cont.)

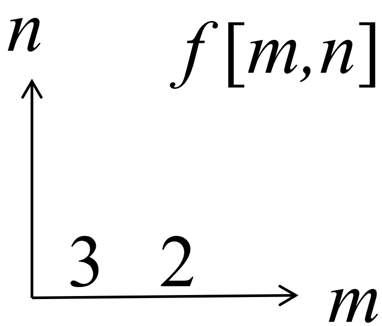


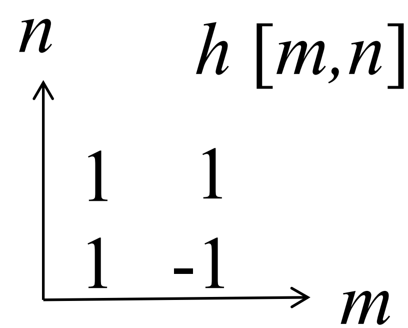


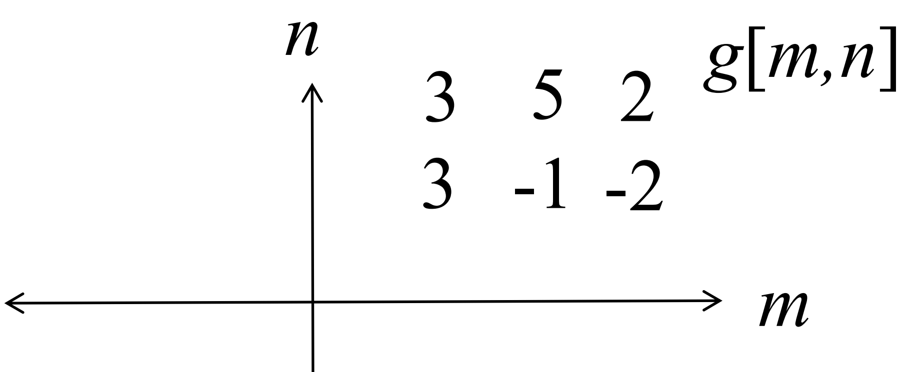




2D discrete convolution (cont.)







$M_1 + M_2 - 1 = 3$

$N_1 + N_2 - 1 = 2$

Synopsis so far

The key points of this lecture were:

- The Fourier Transform as a change of basis.
- Generalization to 2D signals.
- The convolution theorem.
- Nyquist criterion theorem.
- The Discrete Fourier Transform (DFT) and its inverse.
- Representation of the DFT as matrix multiplication.

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

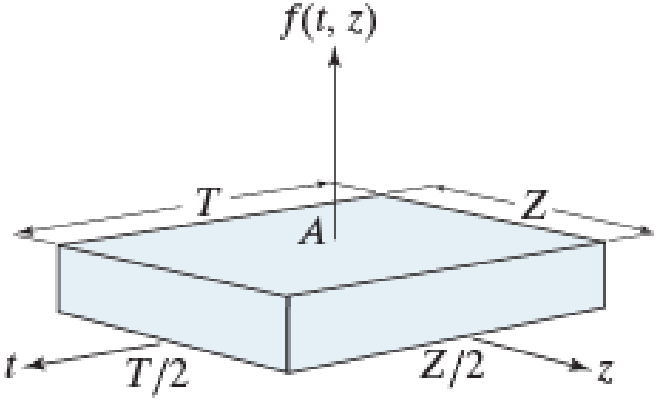
Filtering in the Frequency Domain (Application)

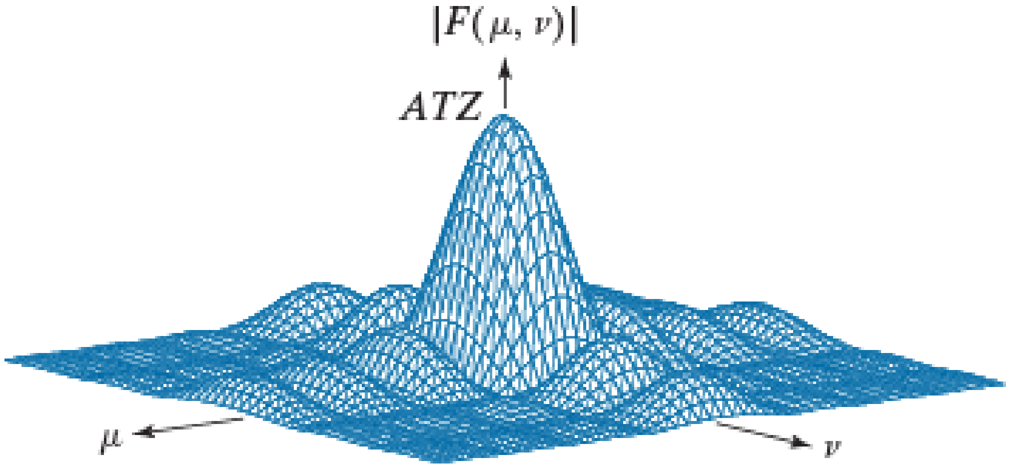
Άγγελος Γιώτης

a.giotis@uoi.gr

Filtering in the frequency domain (Previously in 2D…)

- We talked about representing a signal in the frequency domain





$$f(x,y) = \mathrm{A}\,P_{W/2,W/2}(x,y) \leftrightarrow F(\mu,\nu) = \mathrm{A}W^2 \frac{\sin(\pi\mu W)}{(\pi\mu W)} \frac{\sin(\pi\nu W)}{(\pi\nu W)}$$

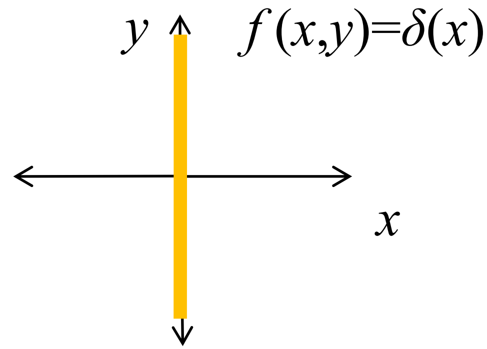
Filtering in the frequency domain (Previously in 2D…)

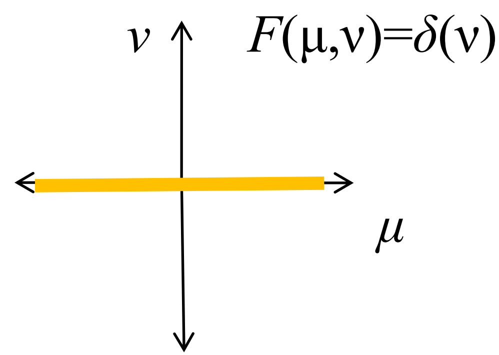
- Example: FT of $f(x,y)=\delta(x)$

$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$=\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi\mu x}dx\int_{-\infty}^{+\infty}e^{-j2\pi\nu y}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi\nu y}dy \quad =\delta(\nu)$$





Filtering in the frequency domain (Previously in 2D…)

- Reminder

$$\Im\{\delta(t)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi\mu t} dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t} \delta(t) dt = e^{-j2\pi\mu 0}$$

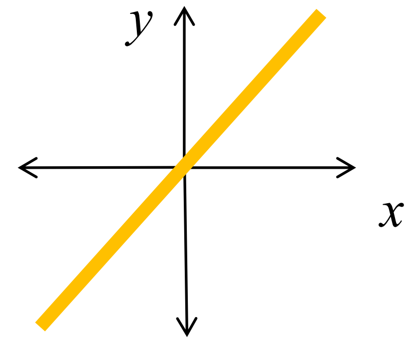
$$\int_{-\infty}^{\infty} f(t)\delta(t) dt = f(0)$$

- Hence, the term that follows is '1'.

$$\int_{-\infty}^{+\infty} \delta(x) e^{-j2\pi\mu x} dx$$

Filtering in the frequency domain (Previously in 2D…)

- Example: FT of $f(x,y)=\delta(x-y)$

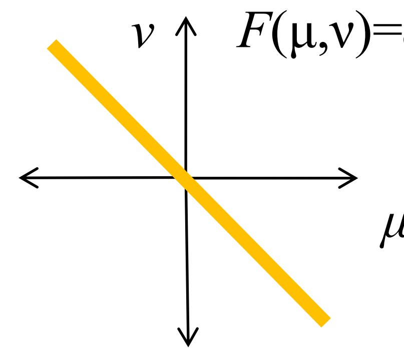


$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$=\int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi\mu x}dx\right]e^{-j2\pi\nu y}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi\mu y}e^{-j2\pi\nu y}dy=\int_{-\infty}^{+\infty}e^{-j2\pi(\mu+\nu)y}dy$$

$$=\delta(\mu+\nu)$$



Filtering in the frequency domain (Previously in 2D…)

- Reminder

$$\Im\{\delta(t-t_0)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t-t_0)e^{-j2\pi\mu t}dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t}\,\delta(t-t_0)dt = e^{-j2\pi\mu t_0}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

- Hence, the term that follows is:

$$\left[\int_{-\infty}^{+\infty} \delta(x-y)e^{-j2\pi\mu x}dx\right] = e^{-j2\pi\mu y}$$

Filtering in the frequency domain (Previously in 2D...)

- 2D continuous convolution

$$f(x,y)*h(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-\alpha,y-\beta)h(\alpha,\beta)d\alpha d\beta$$

- We will examine the discrete convolution in more detail.
- Convolution property

$$f(x,y)*h(x,y) \leftrightarrow F(\mu,\nu)H(\mu,\nu)$$

Filtering in the frequency domain (Previously in 2D…)

- 2D sampling is accomplished by

$$\boldsymbol{s}_{\Delta X \Delta Y}(x,y) = \boldsymbol{s}_{\Delta X}(x)\boldsymbol{s}_{\Delta Y}(y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \delta(x - m\Delta X, y - nDY)$$

- The FT of the sampled 2D signal consists of repetitions of the spectrum of the 1D continuous signal.

$$\tilde{F}(\mu,\nu) = \frac{1}{\Delta X}\frac{1}{\Delta Y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \mathrm{F}(\mu - \frac{m}{\Delta X}, \nu - \frac{n}{\Delta X})$$

2D Discrete Fourier Transform (2D DFT)

- 2D DFT pair of image $f[m,n]$ of size MxN.

$$F[k,l]=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]e^{-j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$f[m,n]=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F[k,l]e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$\begin{cases} 0 \le k \le M-1 \\ 0 \le l \le N-1 \end{cases}, \quad \begin{cases} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{cases}$$

2D Discrete Fourier Transform (2D DFT)

Separability of the 2D DFT:

- We can express the 2D DFT as two 1D DFTs:
- First, perform a 1D DFT along the columns and then along the rows (or vice versa).

From 1D to 2D DFT in matrix form

2D DFT can be represented in matrix form:

- Reminder for 1D: $\mathbf{F} = \mathbf{A}\mathbf{f}$ $\quad w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$

$$\mathbf{A} = \begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \cdots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \cdots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \cdots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}$$

From 1D to 2D DFT in matrix form

2D DFT can be represented in matrix form:

- Reminder 1D: $\mathbf{F} = \mathbf{A}\mathbf{f}$

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{F}$$

$$w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$$

$$\mathbf{A}^{-1} = \frac{1}{N}\left(\mathbf{A}^*\right)^T = \frac{1}{N}\left(\begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \dots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \dots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \dots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}^*\right)^T$$

2D Discrete Fourier Transform (2D DFT)

2D DFT can be represented in matrix form:

- In a similar fashion for 2D we employ the same matrix A:

$$F = \mathrm{A} f \mathrm{A}^{\mathrm{T}}$$

- Where now *F*, *f* are now NxN matrices:
    - Equivalent: $F = AfA$, since $A = A^T$

From 1D to 2D DFT in matrix form

- General case: for NxM εικόνες/signals we have:

$$F = \mathrm{A_N} f \mathrm{A_M}^{\mathrm{T}}$$

- Where now *F*, *f* are now NxM matrices:
  - Equivalent: $F = A_N f A_M$, since $A = A^T$
  - caution: when N ≠ M we multiply with different matrix on the left and right, respectively.

DFT symmetric properties

- From functional analysis any real or complex function, w(x, y), can be expressed as:

$$w(x,y) = w_e(x,y) + w_o(x,y)$$

- Where the even and odd parts are defined as:

$$w_e(x,y) \triangleq \frac{w(x,y) + w(-x,-y)}{2} \qquad w_o(x,y) \triangleq \frac{w(x,y) - w(-x,-y)}{2}$$

- Substituting gives the identity w(x, y) ≡ w(x, y) and then:

$$w_e(x,y) = w_e(-x,-y) \text{ and } w_o(x,y) = -w_o(-x,-y)$$

- *Even* functions are said to be *symmetric* and *odd* functions *antisymmetric*.

DFT symmetry property

- Symmetry Property – FT of real function *f(x,y)* is conjugate symmetric:

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

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

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

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

- For which we can infer: $F(u,v) =| F(-u,-v) |$

DFT symmetry property

- Similarly, FT of conjugate function f is conjugate anti-symmetric:

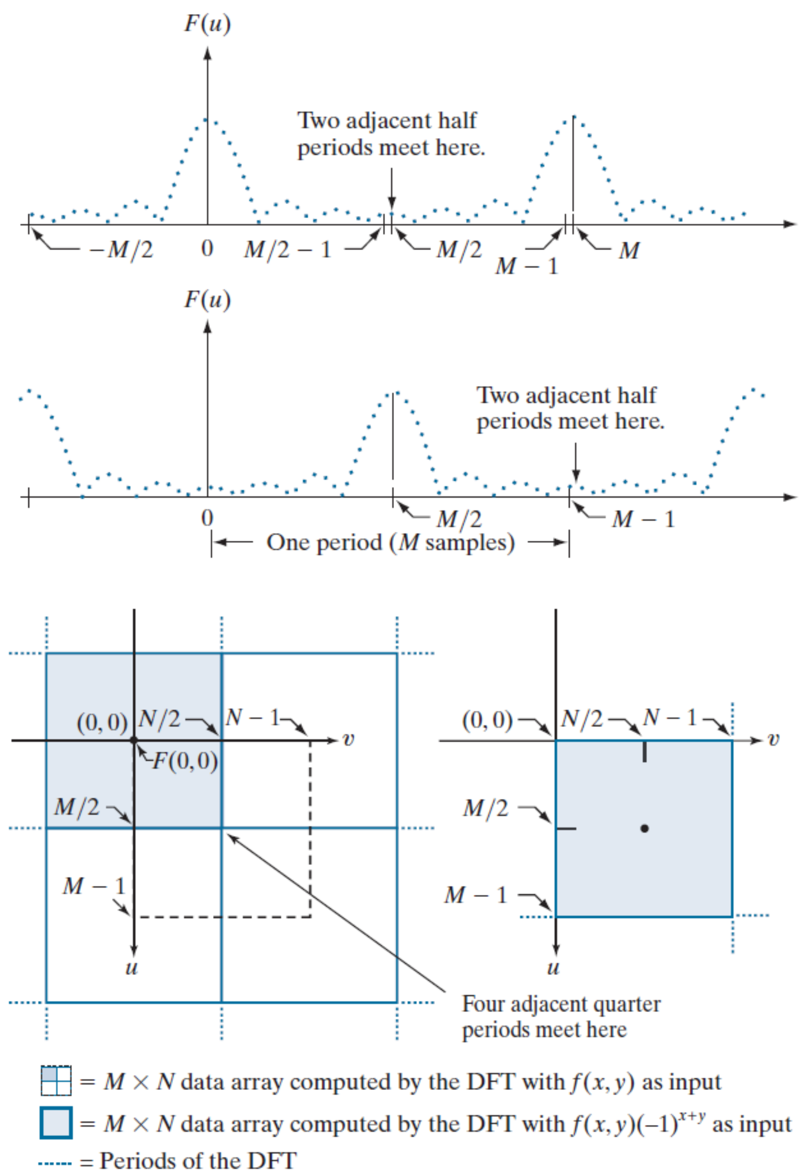
$$F^{*}(-u,-v) = -F(u,v)$$

Periodicity of the DFT

- The range of frequencies of the signal is between [$-M/2$, $M/2$].
- The DFT covers two back-to-back half periods of the signal as it covers [0, $M-1$].
- For display and computation purposes it is convenient to shift the DFT and have a complete period in [0, $M-1$].

(b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$

(d) Shifted DFT obtained by multiplying $f(x,y)$ by $(-1)^{x+y}$ before computing $F(u,v)$

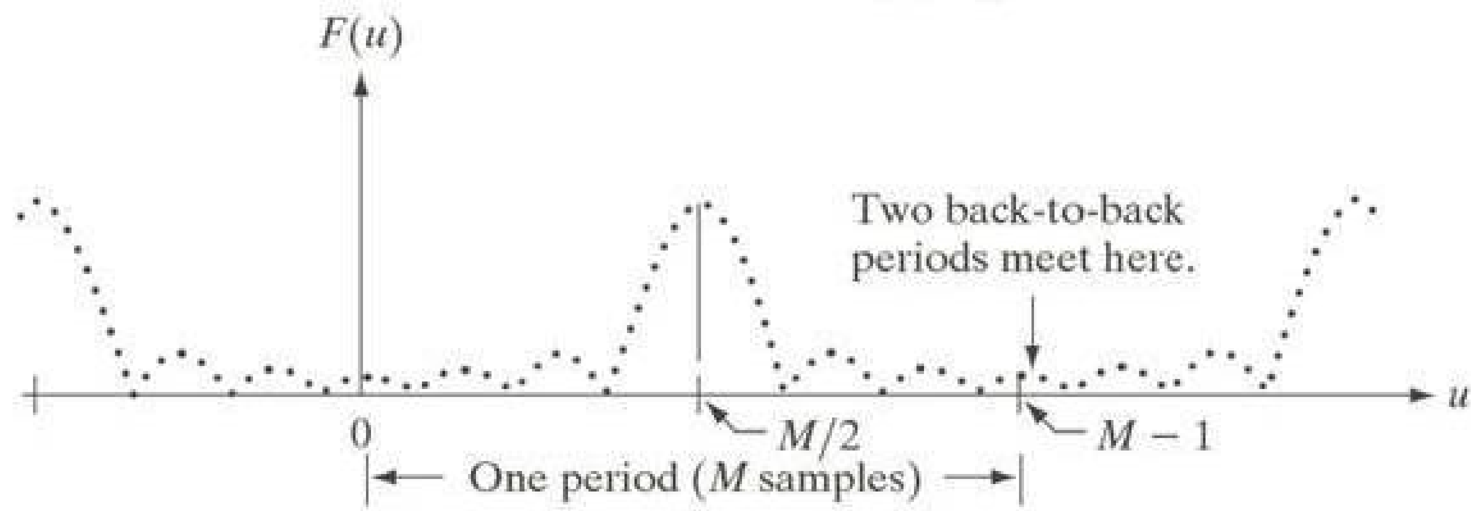


Periodicity of the DFT (cont...)

- From DFT properties: $f[n]e^{j2\pi(N_0 n/M)} \Leftrightarrow F(k-N_0)$

Letting $N_0=M/2$: $f[n](-1)^n \Leftrightarrow F(k-M/2)$

And $F(0)$ is now located at $M/2$.



- Similarly in 2D we shift F(0,0) to (M/2,N/2) using: $f(x,y)(-1)^{x+y}$

DFT properties (synopsis)

TABLE 4.3
Summary of DFT definitions and corresponding expressions.

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x,y)$	$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u,v)$	$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
3) Spectrum	$\lvert F(u,v)\rvert = \left[R^2(u,v) + I^2(u,v)\right]^{1/2} \quad R = \text{Real}(F);\ I = \text{Imag}(F)$
4) Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
5) Polar representation	$F(u,v) = \lvert F(u,v)\rvert e^{j\phi(u,v)}$
6) Power spectrum	$P(u,v) = \lvert F(u,v)\rvert^2$
7) Average value	$\bar{f} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) = \frac{1}{MN}F(0,0)$
8) Periodicity (k_1 and k_2 are integers)	$F(u,v) = F(u+k_1M,v) = F(u,v+k_2N) = F(u+k_1,v+k_2N)$ $f(x,y) = f(x+k_1M,y) = f(x,y+k_2N) = f(x+k_1M,y+k_2N)$
9) Convolution	$(f \bigstar h)(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$
10) Correlation	$(f \,☆\, h)(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f^*(m,n)h(x+m,y+n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.
12) Obtaining the IDFT using a DFT algorithm	$MNf^*(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^*(u,v)e^{-j2\pi(ux/M+vy/N)}$ This equation indicates that inputting $F^*(u,v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x,y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See

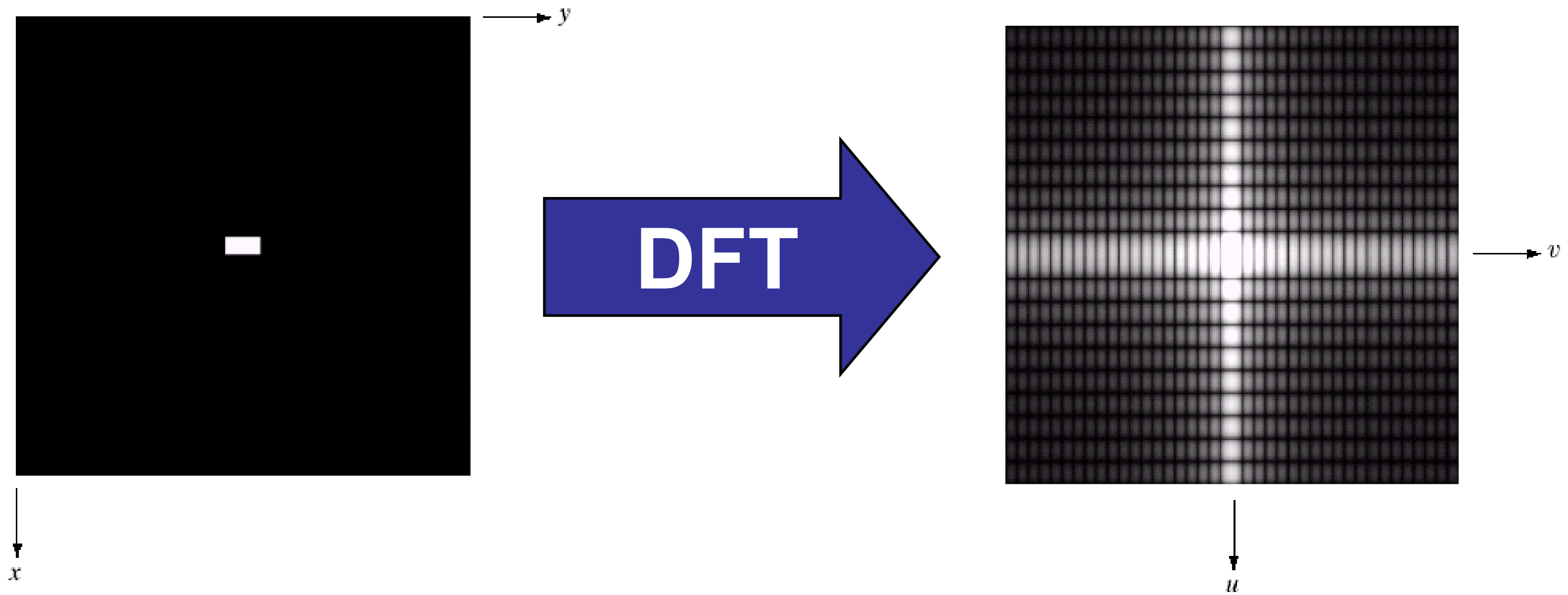
DFT properties (synopsis, cont..)

TABLE 4.4 Summmary of DFT pairs. The closed-form expressions in 12 and 13 are valid only for continuous variables. They can be used with discrete variables by sampling the continuous expressions.

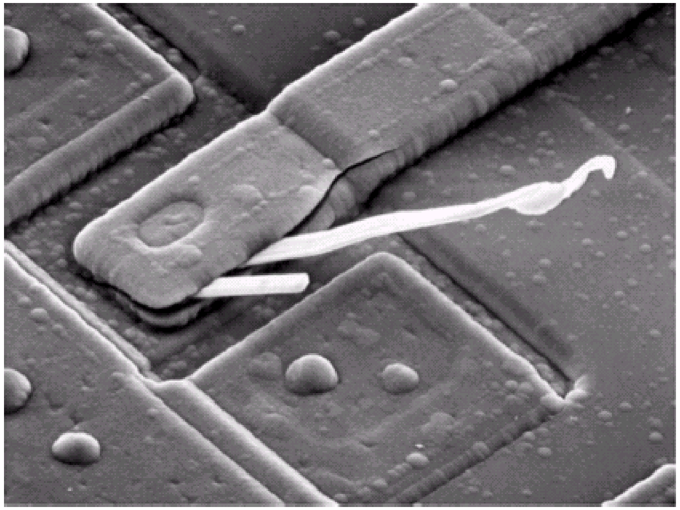
Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x,y)+bf_2(x,y) \Leftrightarrow aF_1(u,v)+bF_2(u,v)$
3) Translation (general)	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0,v-v_0)$ $f(x-x_0,y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2,N/2)$	$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2,v-N/2)$ $f(x-M/2,y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$
5) Rotation	$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$ $x=r\cos\theta$ $y=r\sin\theta$ $u=\omega\cos\varphi$ $v=\omega\sin\varphi$ $r=\sqrt{x^2+y^2}$ $\theta=\tan^{-1}(y/x)$ $\omega=\sqrt{u^2+v^2}$ $\varphi=\tan^{-1}(v/u)$
6) Convolution theorem†	$f\star h)(x,y) \Leftrightarrow (F\bullet H)(u,v)$ $(f\bullet h)(x,y) \Leftrightarrow (1/MN)[(F\star H)(u,v)]$
7) Correlation theorem†	$(f\,☆\,h)(x,y) \Leftrightarrow (F^*\bullet H)(u,v)$ $(f^*\bullet h)(x,y) \Leftrightarrow (1/MN)[(F\,☆\,H)(u,v)]$
8) Discrete unit impulse	$\delta(x,y) \Leftrightarrow 1$ $1 \Leftrightarrow MN\delta(u,v)$
9) Rectangle	$\text{rec}[a,b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0x/M+2\pi v_0y/N) \Leftrightarrow \dfrac{jMN}{2}[\delta(u+u_0,v+v_0)-\delta(u-u_0,v-v_0)]$
11) Cosine	$\cos(2\pi u_0x/M+2\pi v_0y/N) \Leftrightarrow \dfrac{1}{2}[\delta(u+u_0,v+v_0)+\delta(u-u_0,v-v_0)]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) Differentiation (the expressions on the right assume that $f(\pm\infty,\pm\infty)=0$.	$\left(\dfrac{\partial}{\partial t}\right)^m\left(\dfrac{\partial}{\partial z}\right)^n f(t,z) \Leftrightarrow (j2\pi\mu)^m(j2\pi\nu)^n F(\mu,\nu)$ $\dfrac{\partial^m f(t,z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m \mathbf{F}(\mu,\nu);$ $\dfrac{\partial^n f(t,z)}{\partial z^m} \Leftrightarrow (j2\pi\nu)^n F(\mu,\nu)$
13) Gaussian	$A2\pi\sigma^2e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

DFT & Images

The DFT of a two dimensional image can be visualised by showing the spectrum of the image component frequencies

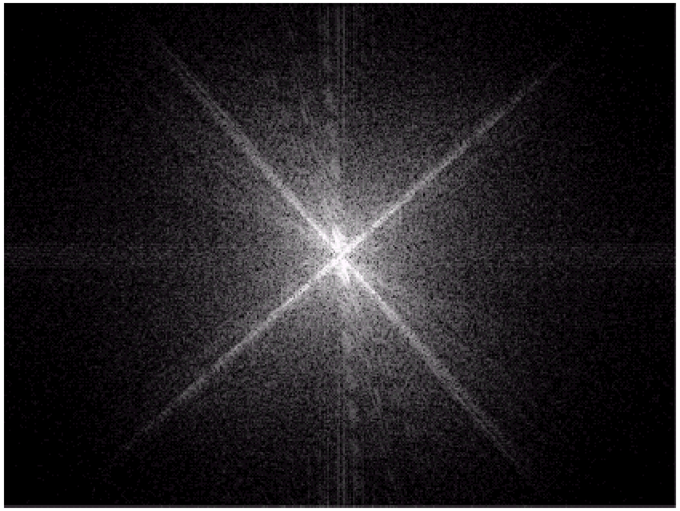


DFT & Images (cont…)



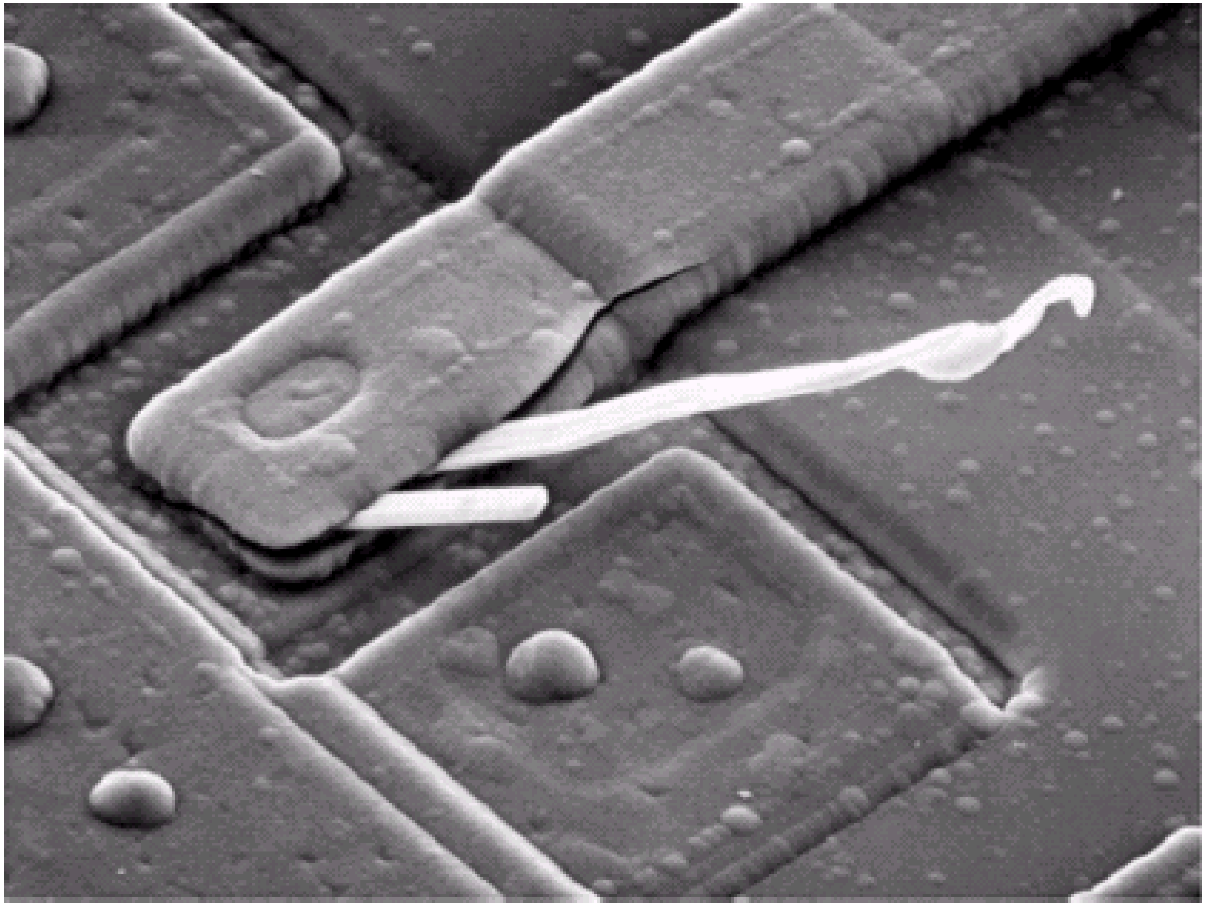
Scanning electron microscope image of an integrated circuit magnified ~2500 times

DFT

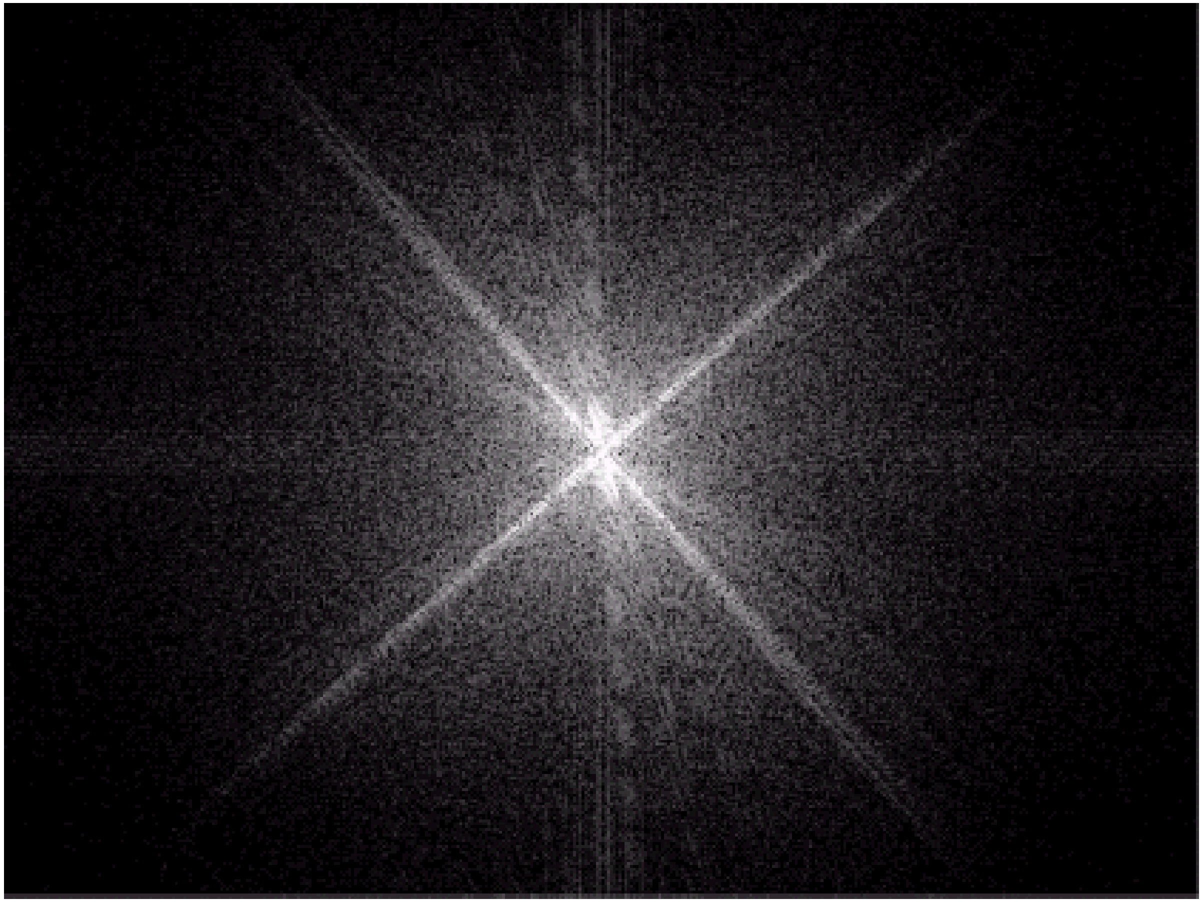


Fourier spectrum of the image

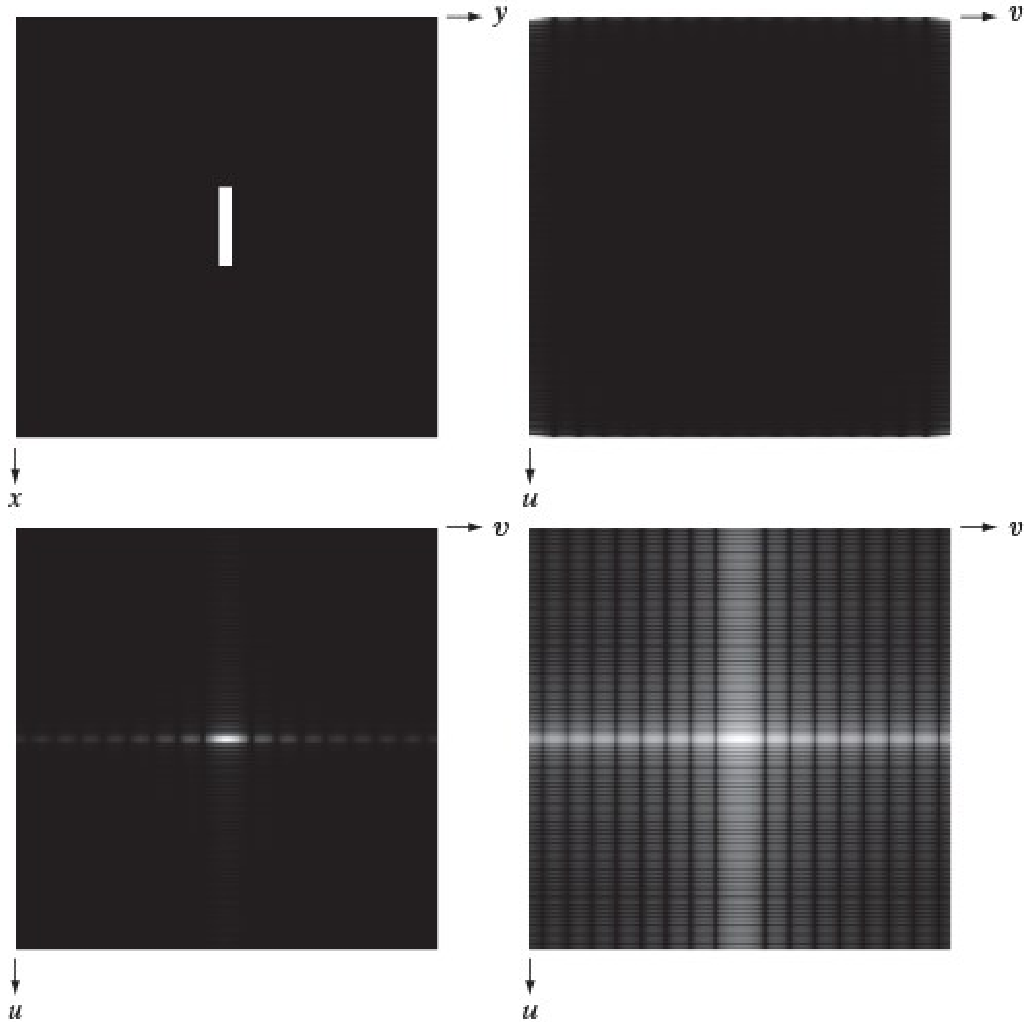
DFT & Images (cont…)



DFT & Images (cont…)



DFT & Images (cont…)



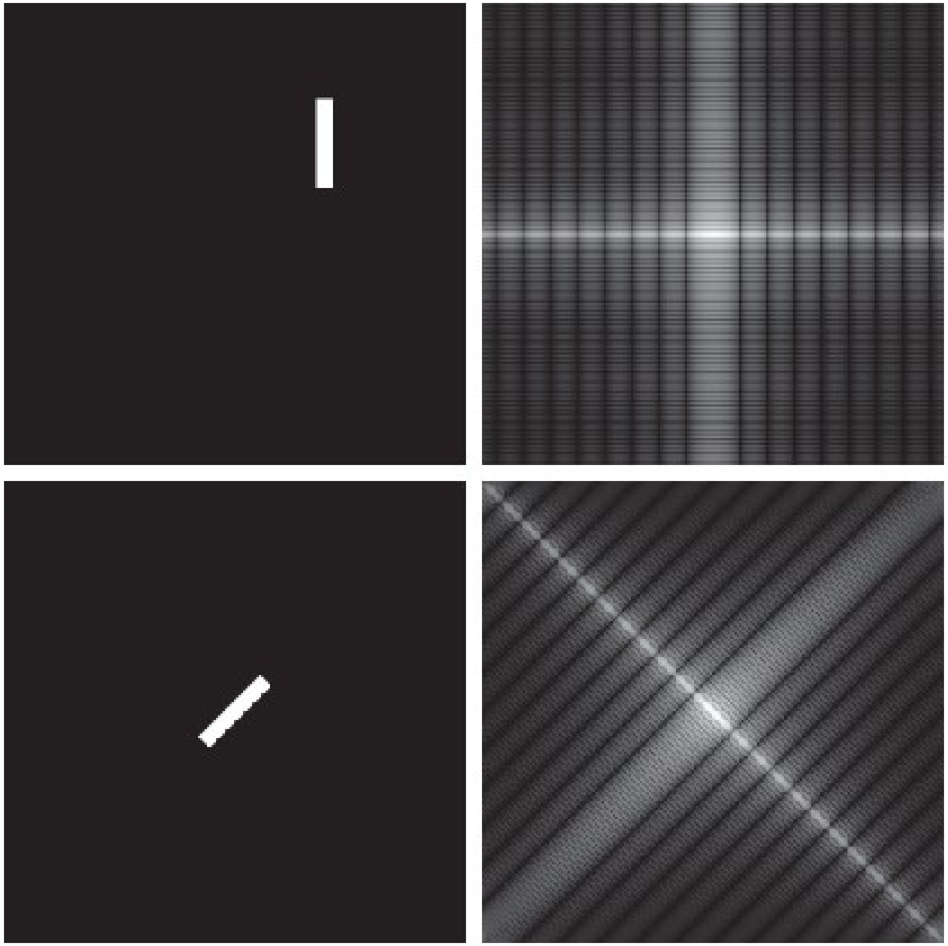
a b
c d

FIGURE 4.23
(a) Image.
(b) Spectrum, showing small, bright areas in the four corners (you have to look carefully to see them).
(c) Centered spectrum.
(d) Result after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The right-handed coordinate convention used in the book places the origin of the spatial and frequency domains at the top left (see Fig. 2.19).

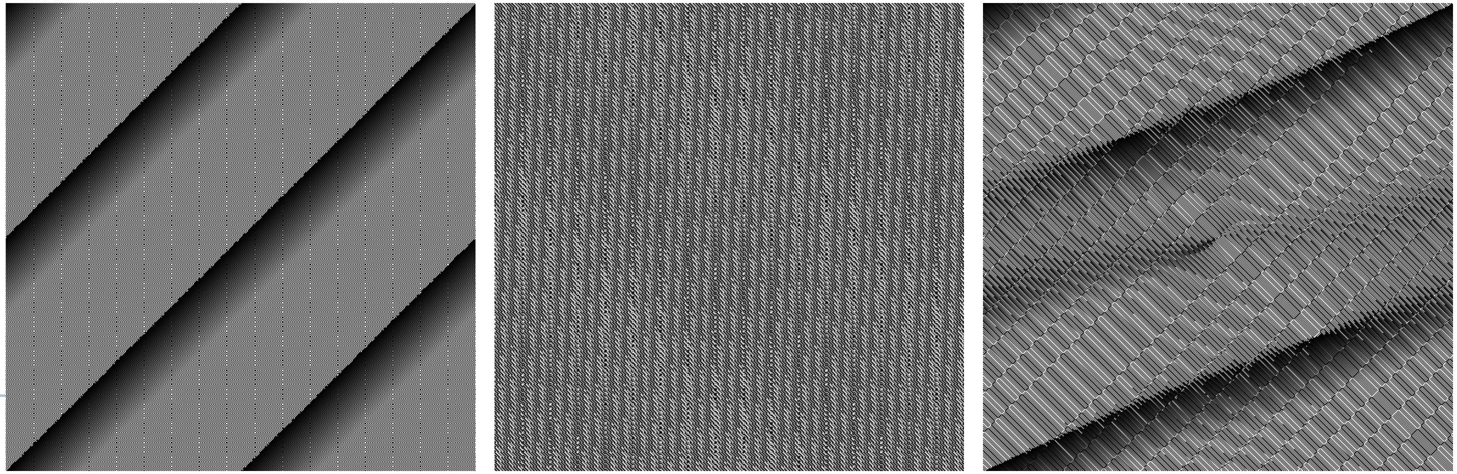
DFT & Images (cont…)

a b
c d

FIGURE 4.24
(a) The rectangle in Fig. 4.23(a) translated.
(b) Corresponding spectrum.
(c) Rotated rectangle.
(d) Corresponding spectrum.
The spectrum of the translated rectangle is identical to the spectrum of the original image in Fig. 4.23(a).



DFT & Images (cont…)



a b c

FIGURE 4.25 Phase angle images of (a) centered, (b) translated, and (c) rotated rectangles.

Although the images differ by a simple geometric transformation no intuitive information may be extracted from their phases regarding their relation.

DFT & Images (cont…)

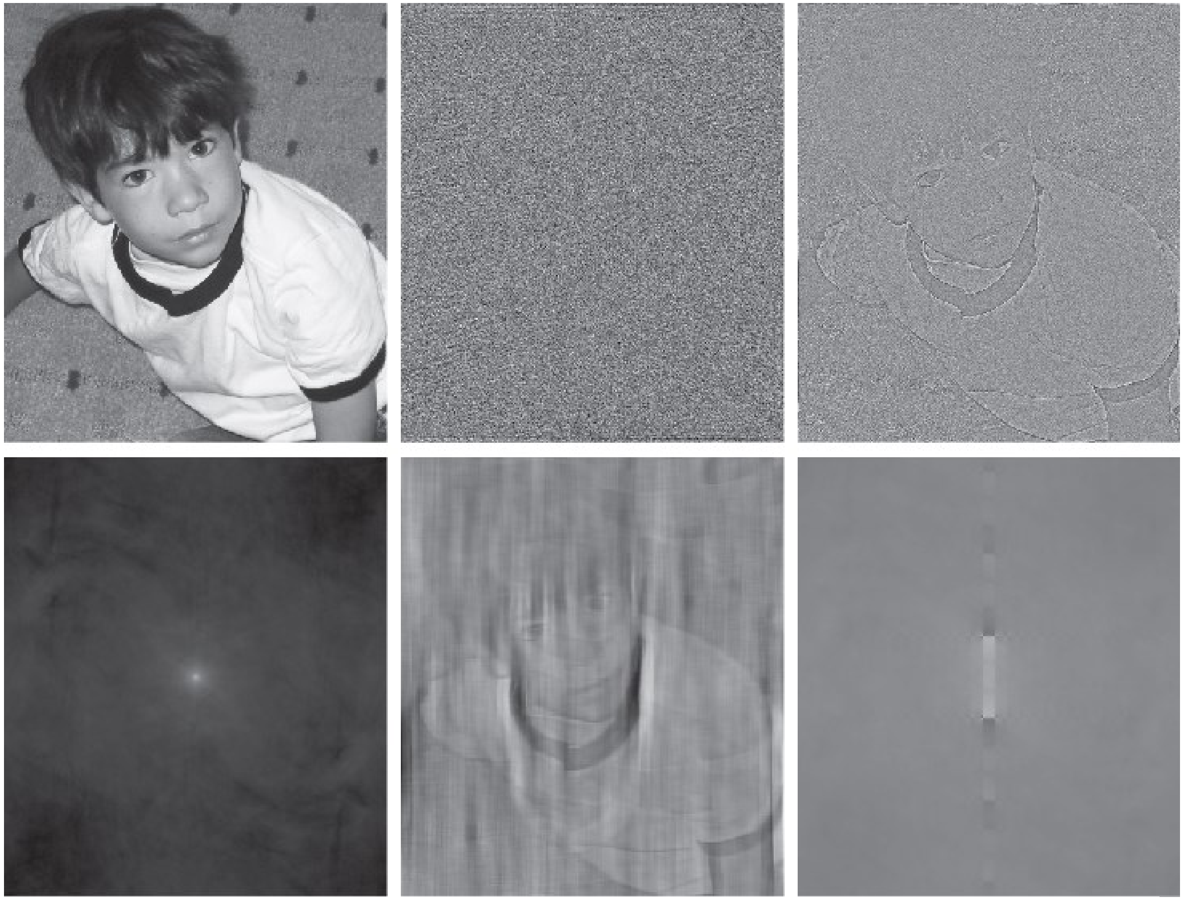
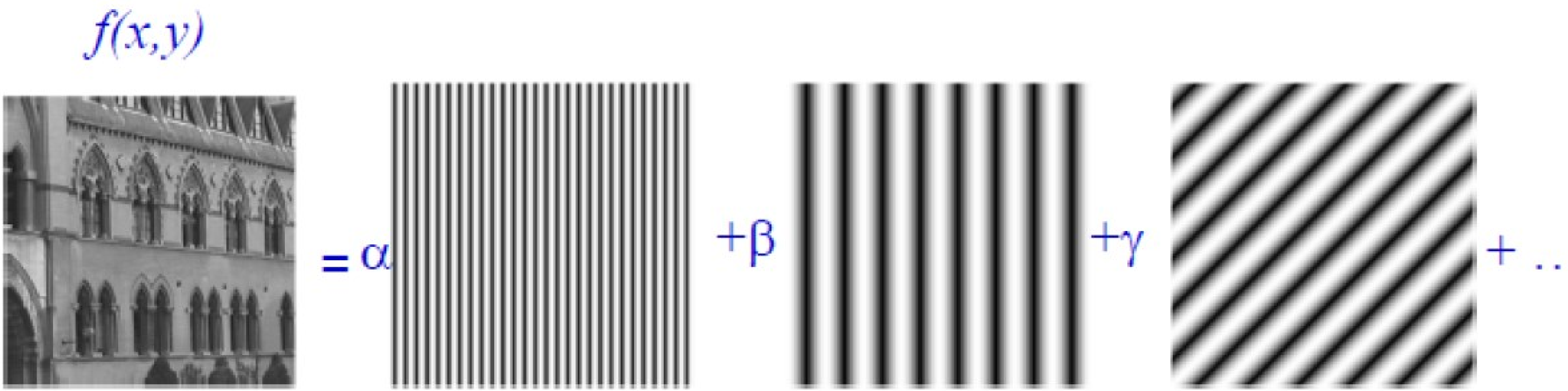


FIGURE 4.26 (a) Boy image. (b) Phase angle. (c) Boy image reconstructed using only its phase angle (all shape features are there, but the intensity information is missing because the spectrum was not used in the reconstruction). (d) Boy image reconstructed using only its spectrum. (e) Boy image reconstructed using its phase angle and the spectrum of the rectangle in Fig. 4.23(a). (f) Rectangle image reconstructed using its phase and the spectrum of the boy's image.

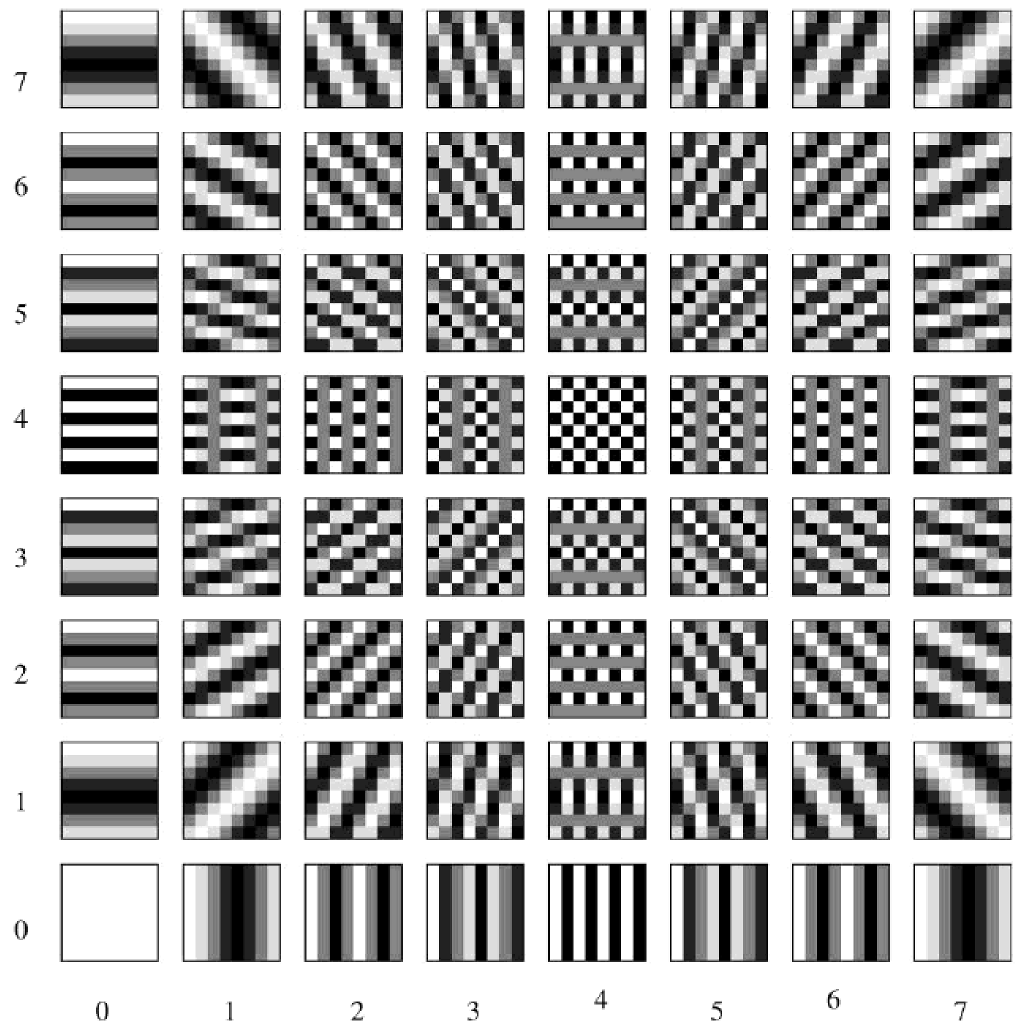
DFT in image domain (interpretation example)

- Basis functions example of 2D continuous FT (real part)



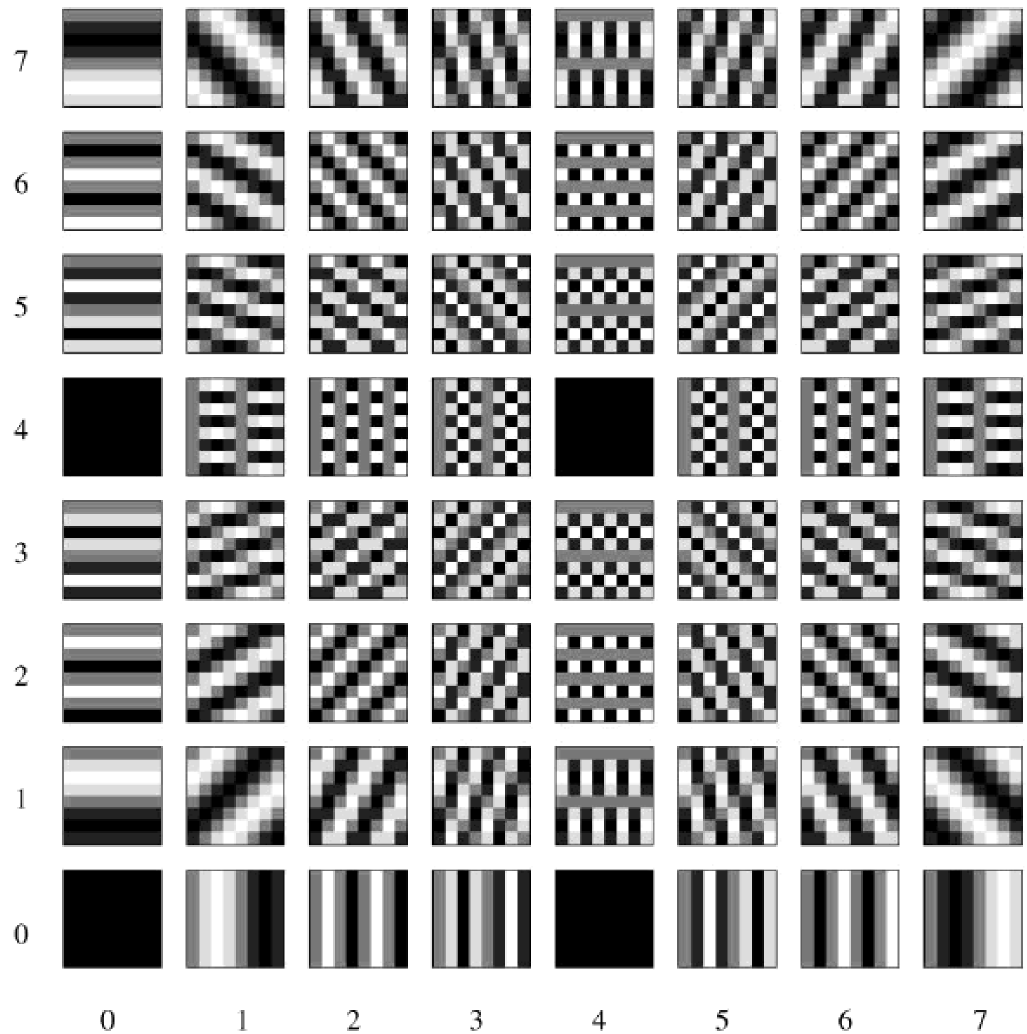
DFT in image domain (interpretation example)

- Basis functions example of 2D DFT (real part – 8x8))



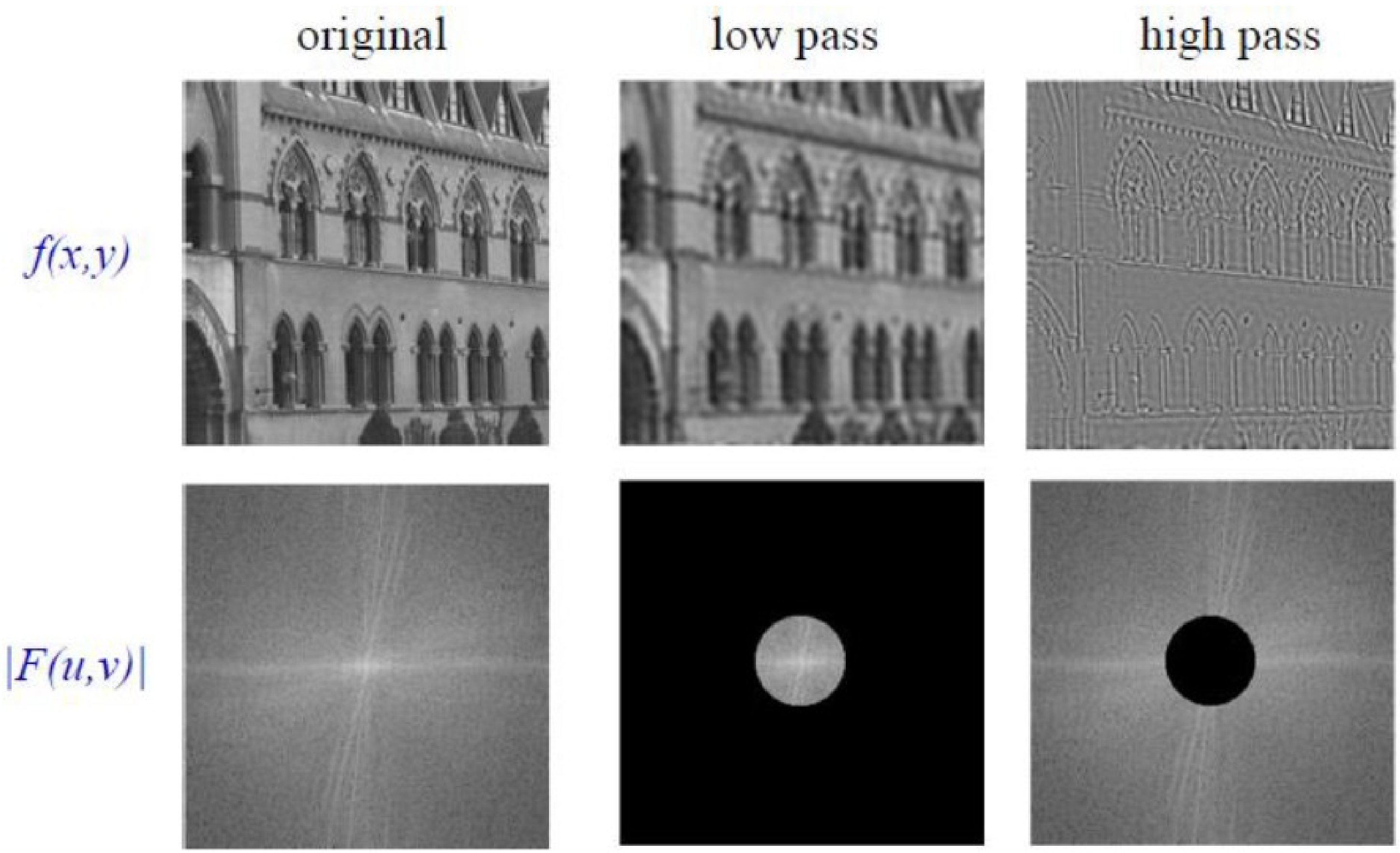
DFT in image domain (interpretation example)

- Basis functions example of 2D DFT (imaginary part – 8x8))



DFT in image domain (interpretation example)

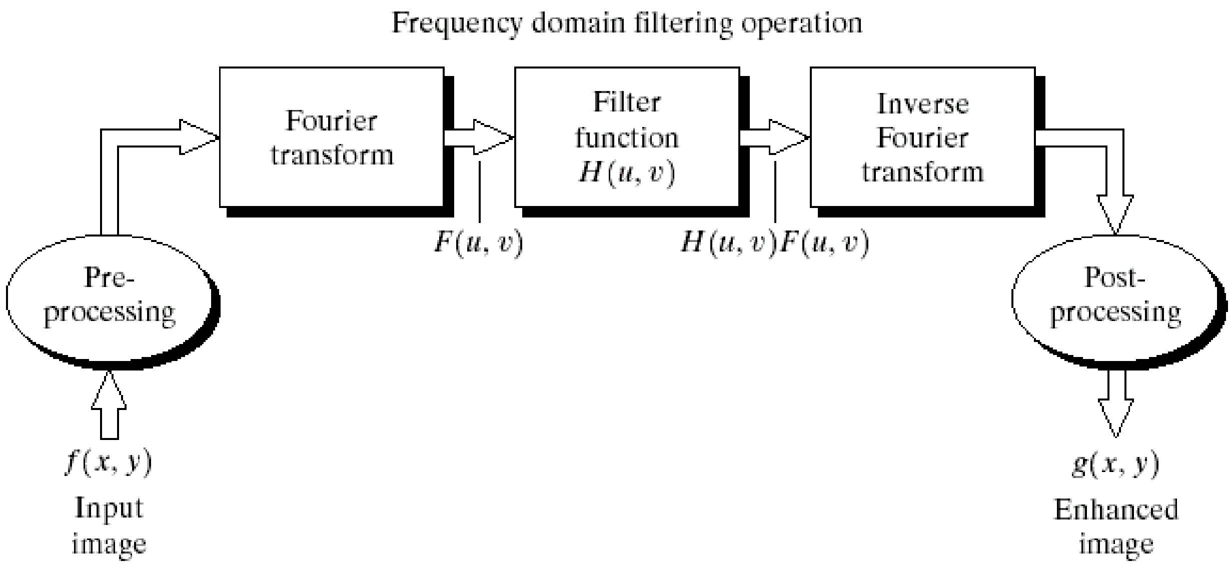
- Keeping part of all the terms is one way to construct filters in the frequency domain.



The DFT and Image Processing

To filter an image in the frequency domain:

1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result



Some Basic Frequency Domain Filters

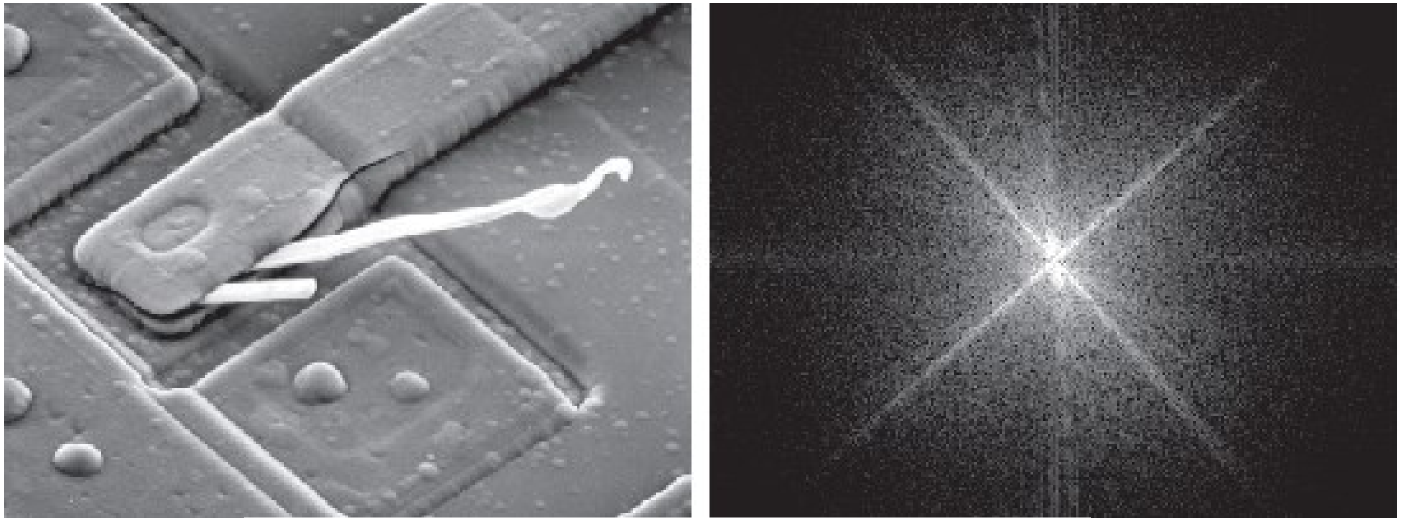
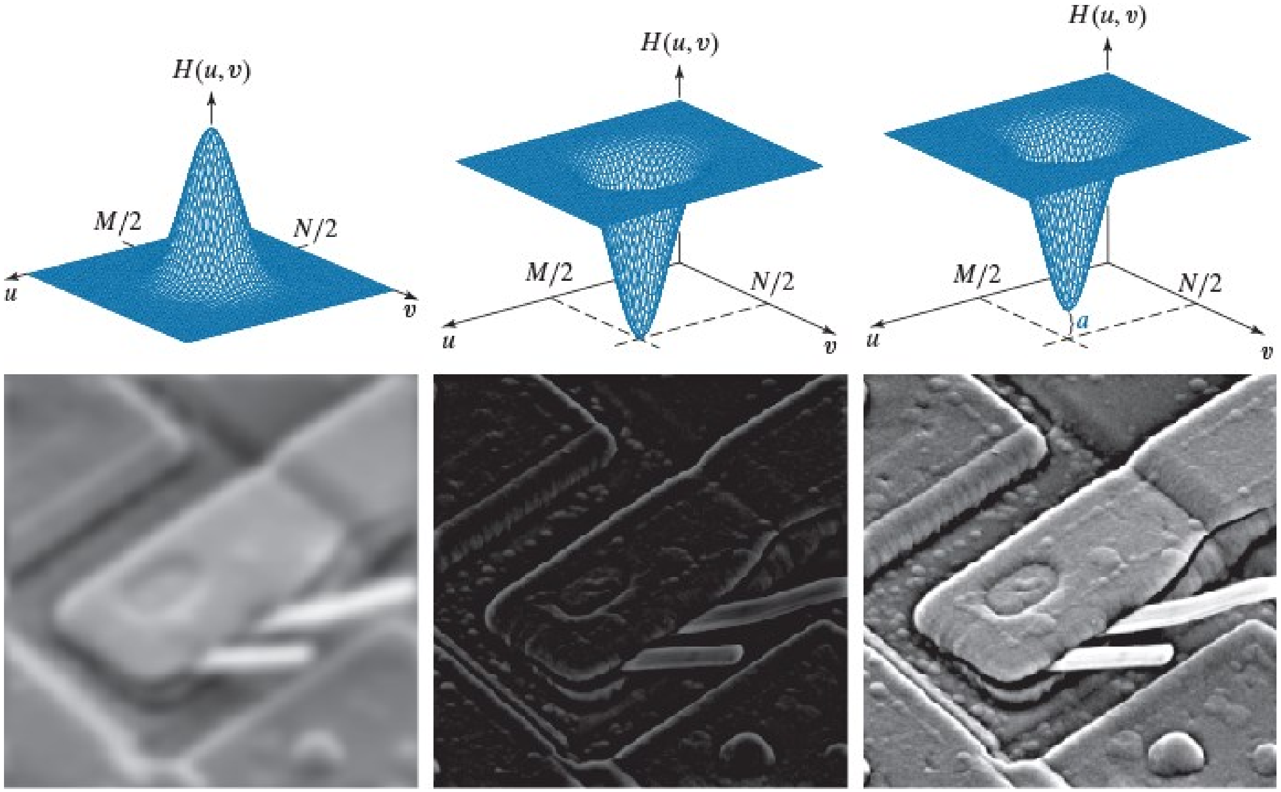


FIGURE 4.28 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

The DFT is centered after multiplication of the image by $(-1)^{m+n}$

Some Basic Frequency Domain Filters (cont.)



The importance of zero padding (cont.)

The DFT considers that the signal is periodic and produces *wraparound* errors.

a f
b g
c h
d i
e j

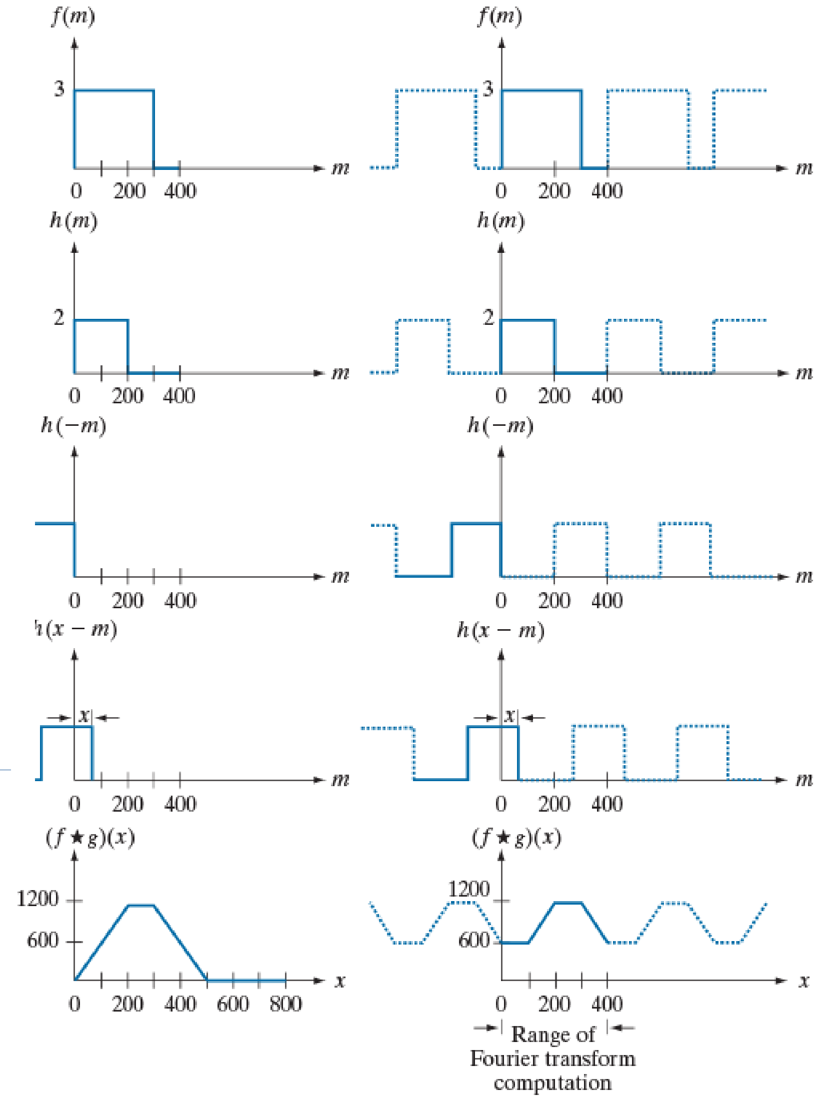


FIGURE 4.27 Left column: Spatial convolution computed with Eq. (3-44), using the approach discussed in Section 3.4. Right column: Circular convolution. The solid line in (j) is the result we would obtain using the DFT, or, equivalently, Eq. (4-48). This erroneous result can be remedied by using zero padding.

The importance of zero padding (cont.)

- All of the properties of 1D DFT hold.
- Particularly:
  - Let $f[m,n]$ be of size M_1xN_1 and $h[m,n]$ of size M_2xN_2 .
  - If the signals are zero-padded to size (M_1+M_2-1)x(N_1+N_2-1) then their circular convolution will be the same as their linear convolution and:

$$\tilde{g}[m,n] = \tilde{f}[m,n] * \tilde{h}[m,n] \leftrightarrow \tilde{G}[k,l] = \tilde{F}[k,l]\tilde{H}[k,l]$$

The importance of zero padding (cont.)

a b c

FIGURE 4.31 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with zero padding. Compare the vertical edges in (b) and (c).

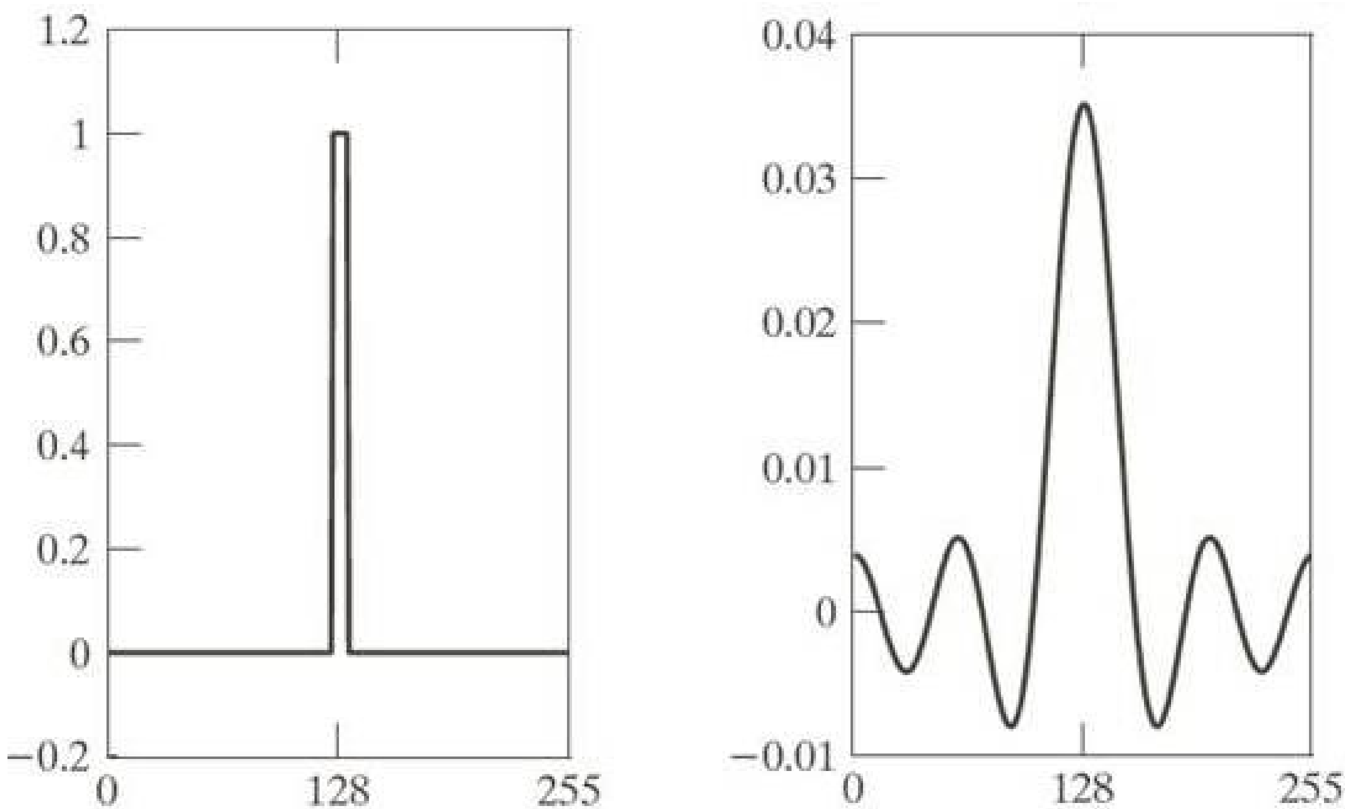
- The image and the DFT are considered to be periodic.
- The vertical edges of the middle image are not blurred if no padding is applied. Why?

The importance of zero padding

- In reality, we perform padding in the spatial domain, while the filter is defined in the frequency domain, which poses a problem. One solution is (naïve approach):
  - Compute the inverse DFT of the filter.
  - Pad the filter in the spatial domain to have the same size as the image.
  - Compute its DFT to return to the frequency domain.

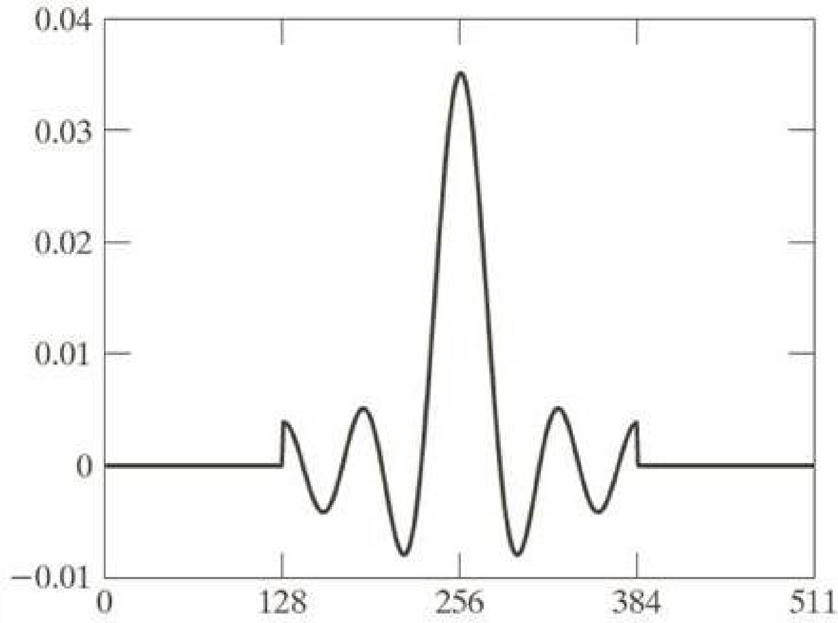
Spatial zero-padding and filters (cont.)

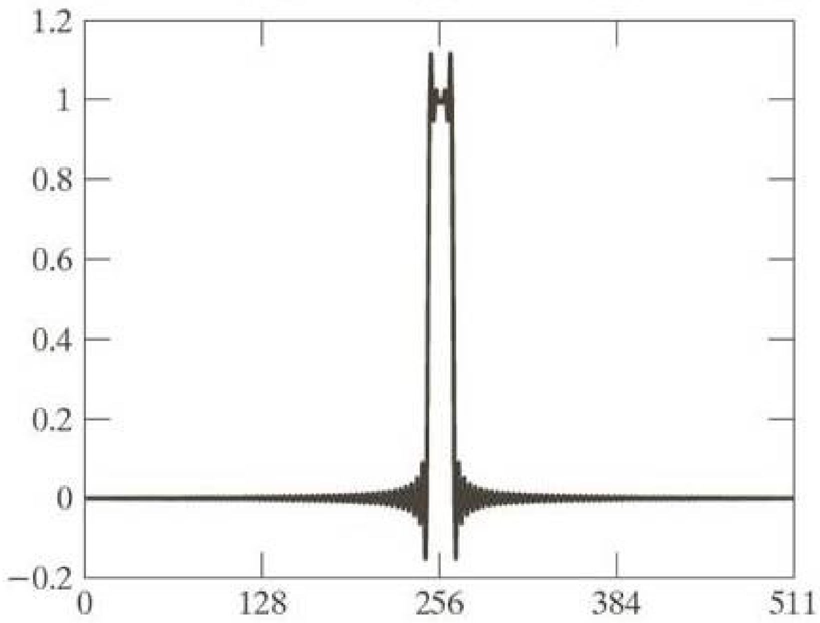
- The filter and its inverse DFT of length 256 (continuous line)



Spatial zero-padding and filters (cont.)

- Zero-padded filter and its DFT





- Spatial truncation of the filter results in ringing effects.

Spatial zero-padding and filters (cont.)

- We cannot work with an infinite number of filter components and simultaneously perform zero-padding to avoid aliasing.
- A decision on which limitation to accept is required.
- One solution to zero-pad the image and then use a filter of the same size with no zero-padding
  - Small errors due to aliasing but it is generally preferable than ringing.
- Another solution is to choose filters attenuating gradually instead of ideal filters.

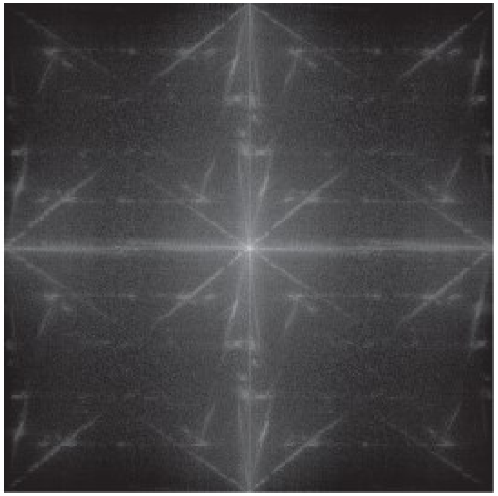
Spatial zero-padding and filters (cont.)

- Smooth regions in terms of intensity are related to low frequencies.

- *Low-pass* filters allow low frequencies to pass through while attenuating high frequencies.

Steps of filtering in the DFT domain

- What if the filter is known in the spatial domain?



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

- Apply a 3x3 Sobel filter to the 600x600 image in the frequency domain.

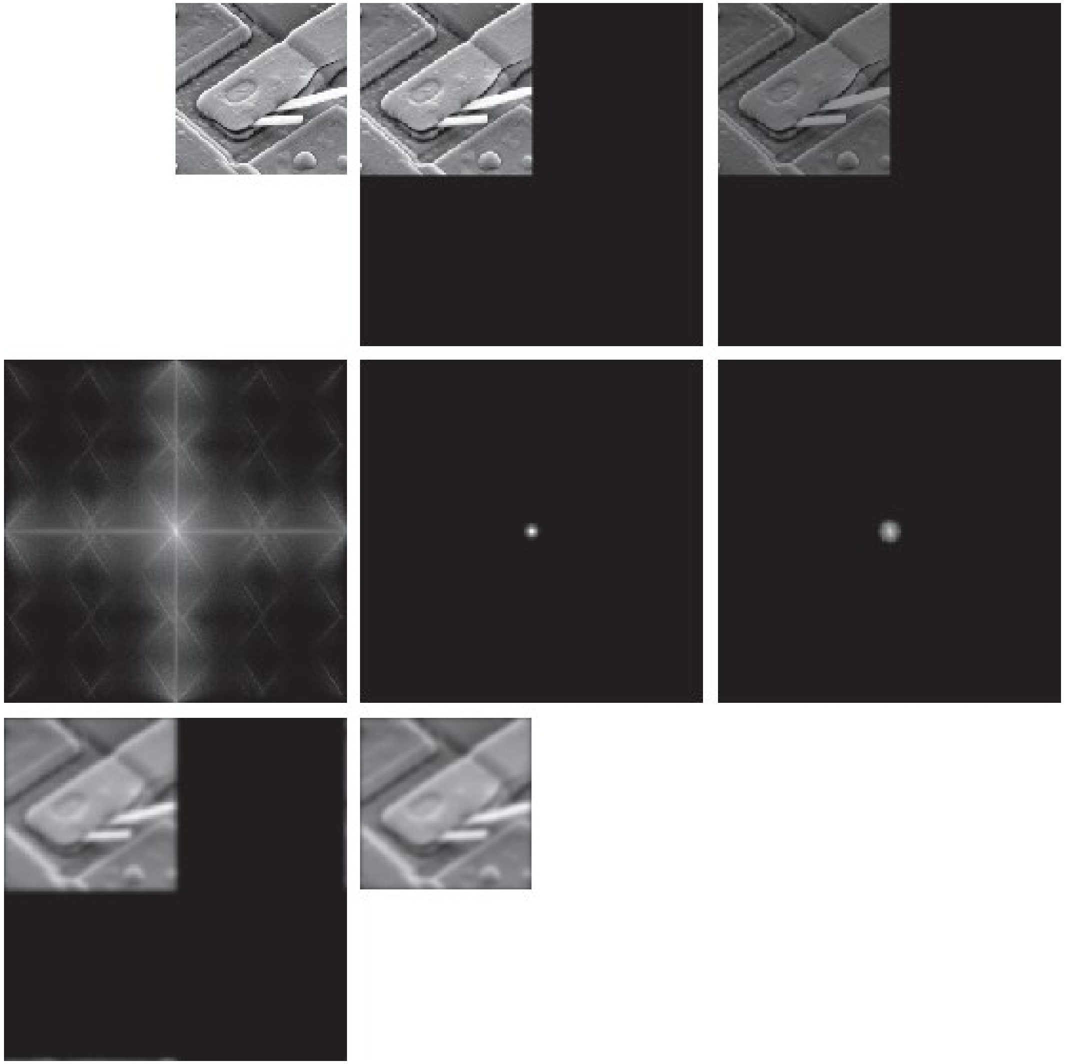
Steps of filtering in the DFT domain (cont.)

1. Pad the image and the filter to 602x602.
2. Place the filter to the center of the the 602x602 padded array.
3. Multiply the filter by $(-1)^{m+n}$ to place the center of the filter to the top left corner (0,0) of the array.
4. Compute the DFT of the filter.
5. Compute the DFT of the image.
6. Multiply the DFTs and invert.

Steps of filtering in the DFT domain (cont.)

a b c
d e f
g h

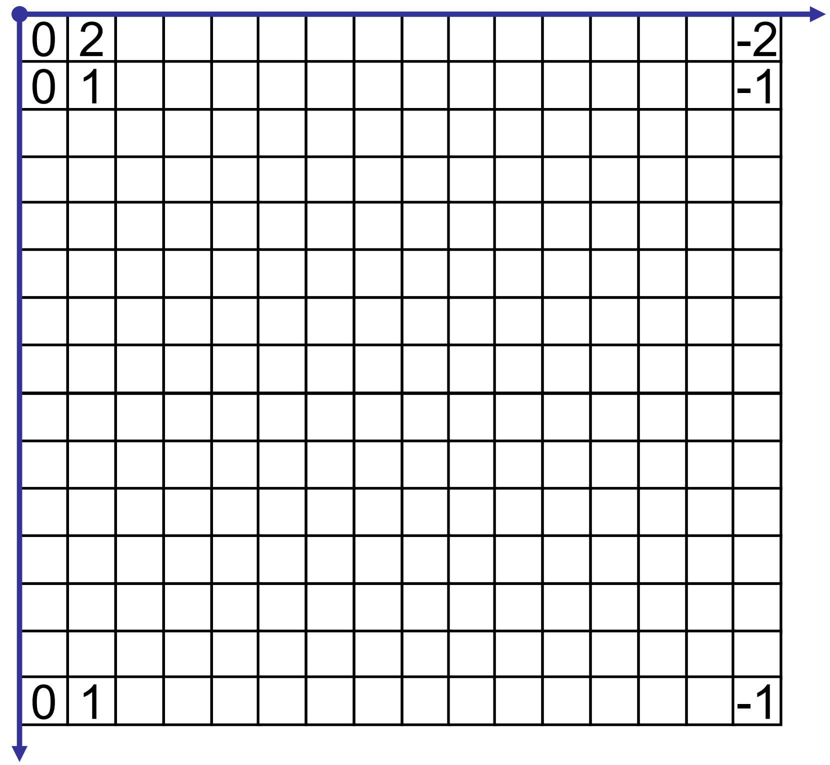
FIGURE 4.35
(a) An $M \times N$ image, f.
(b) Padded image, f_p of size $P \times Q$.
(c) Result of multiplying f_p by $(-1)^{x+y}$.
(d) Spectrum of F. (e) Centered Gaussian lowpass filter transfer function, H, of size $P \times Q$.
(f) Spectrum of the product HF.
(g) Image g_p, the real part of the IDFT of HF, multiplied by $(-1)^{x+y}$.
(h) Final result, g, obtained by extracting the first M rows and N columns of g_p.



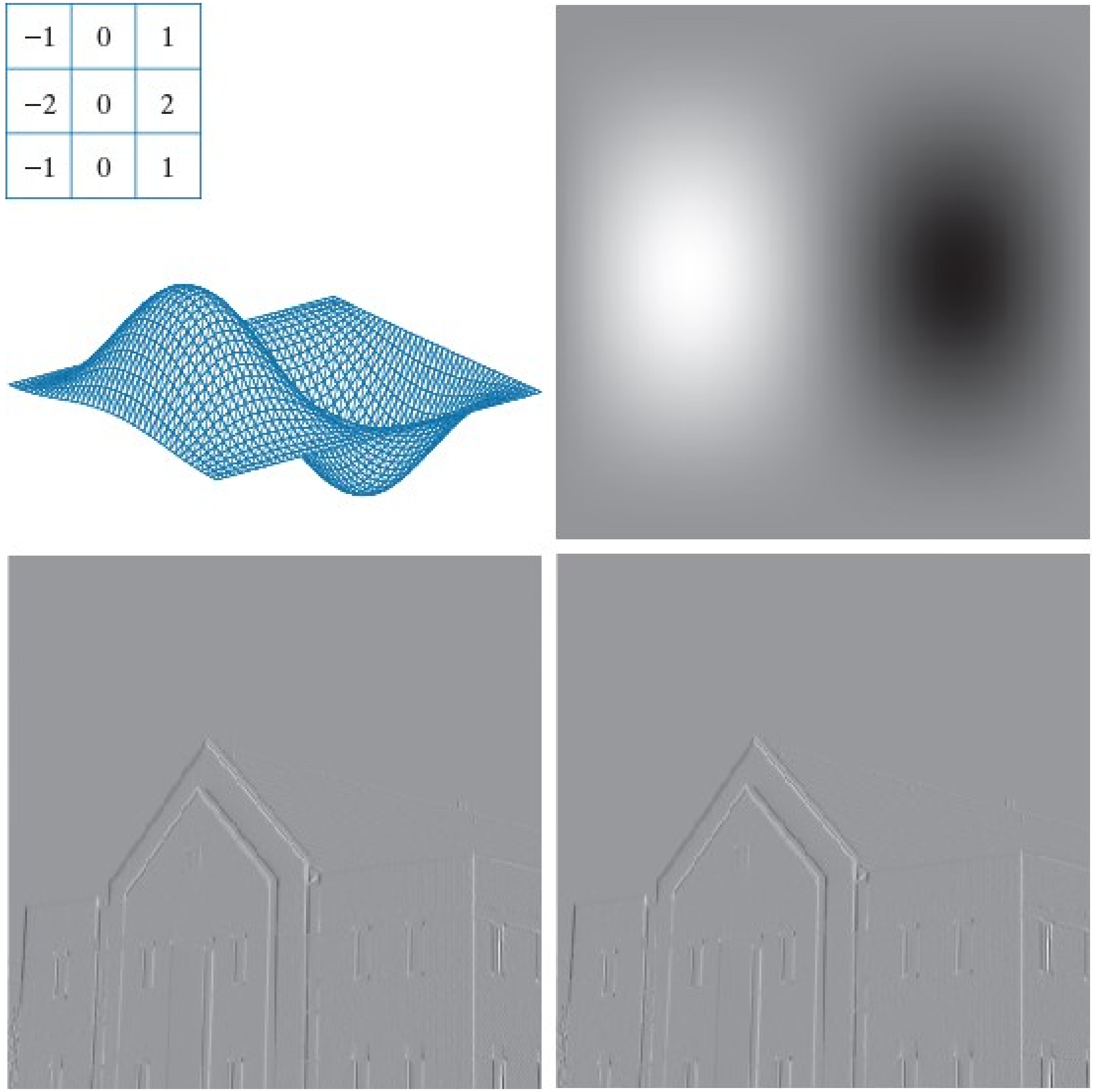
Steps of filtering in the DFT domain (cont.)

- Alternatively, pad to 602x602, repeat the filter periodically and compute the DFT.

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



The importance of zero padding (cont...)



−1	0	1
−2	0	2
−1	0	1

a b
c d

FIGURE 4.38 (a) A spatial kernel and perspective plot of its corresponding frequency domain filter transfer function. (b) Transfer function shown as an image. (c) Result of filtering Fig. 4.37(a) in the frequency domain with the transfer function in (b). (d) Result of filtering the same image in the spatial domain with the kernel in (a). The results are identical.

Smoothing Frequency Domain Filters

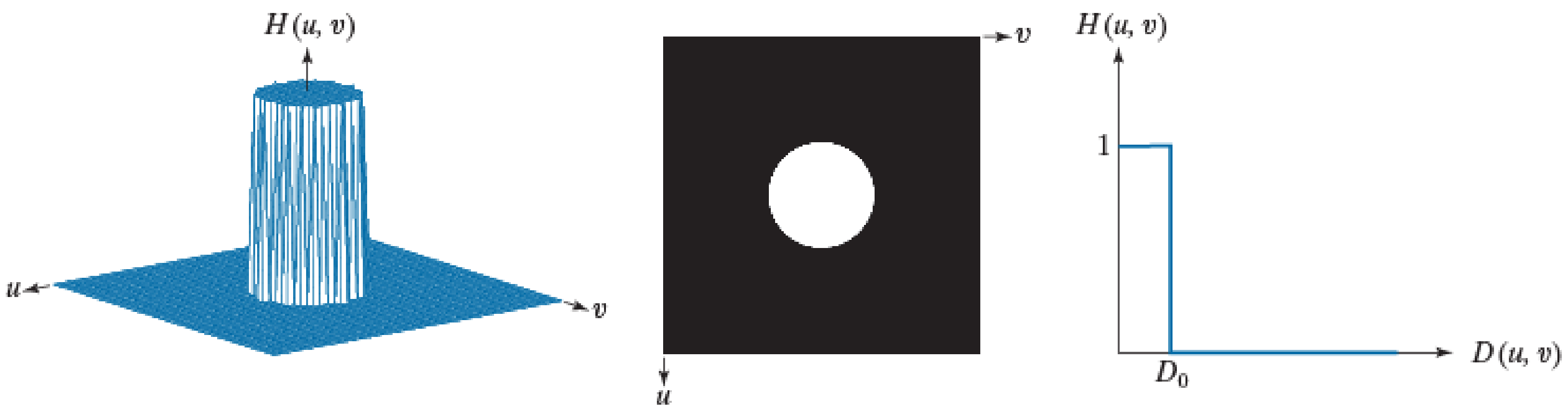
- Smoothing is achieved in the frequency domain by dropping out the high frequency components
- The basic model for filtering is:

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

- where $F(u,v)$ is the Fourier transform of the image being filtered and $H(u,v)$ is the filter transform function
- *Low pass filters* – only pass the low frequencies, drop the high ones.

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin of the transform.

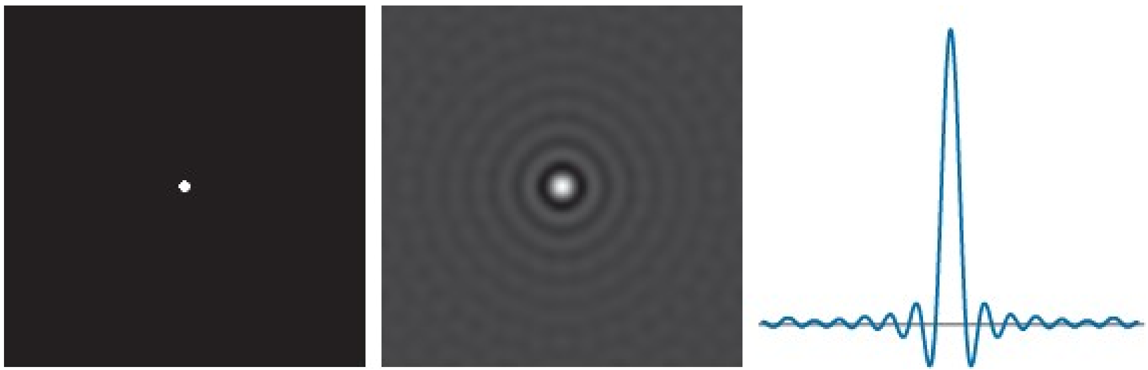


Changing the distance changes the behaviour of the filter.

Ideal Lowpass Filters (cont...)

a b c

FIGURE 4.42
(a) Frequency domain ILPF transfer function.
(b) Corresponding spatial domain kernel function.
(c) Intensity profile of a horizontal line through the center of (b).



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

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

Ideal Low Pass Filter (cont…)

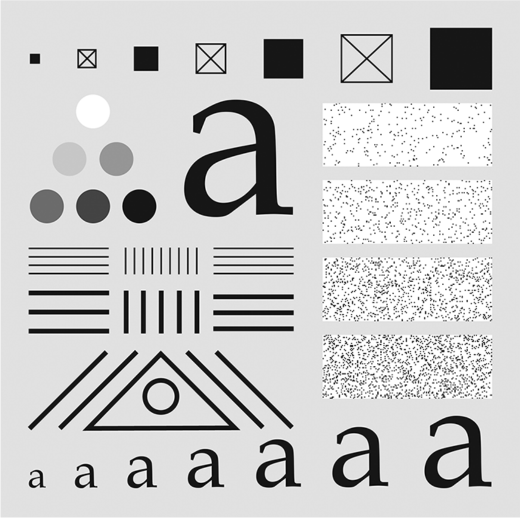
The transfer function for the ideal low pass filter can be given as:

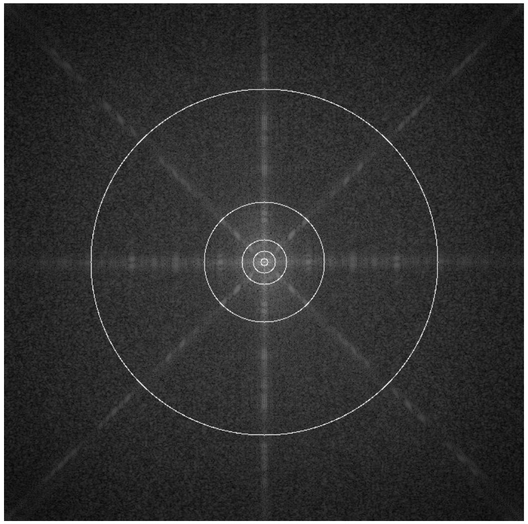
$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)>D_0\end{cases}$$

where $D(u,v)$ is given as:

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

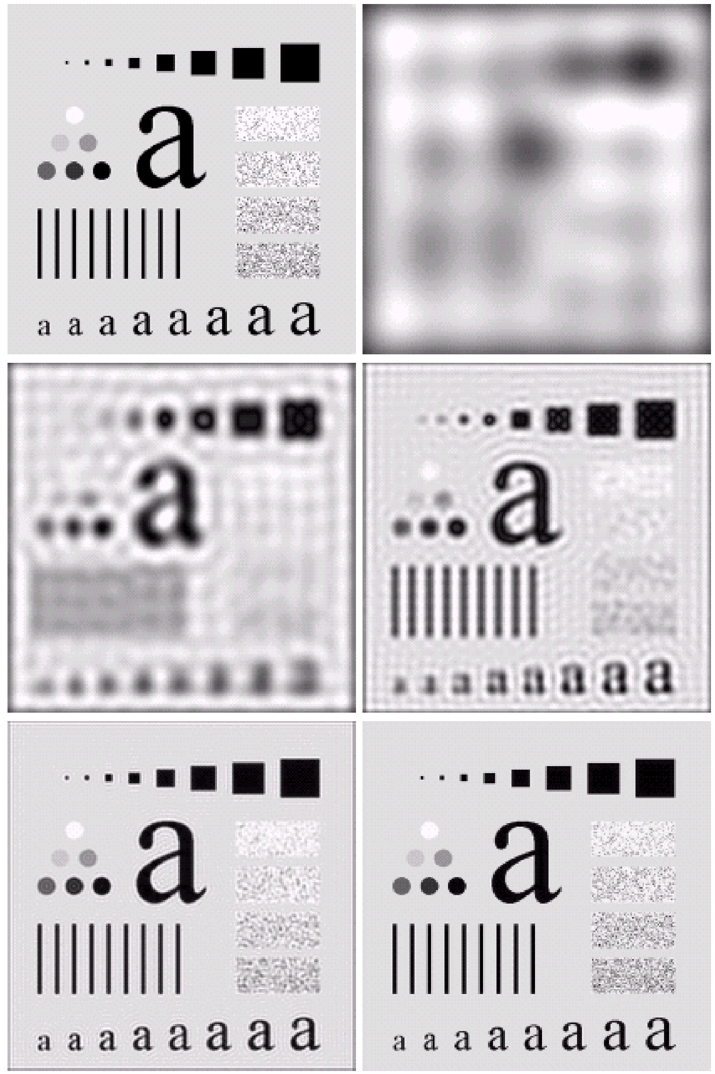
Ideal Low Pass Filter (cont…)





An image, its Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it.

Ideal Low Pass Filter (cont…)



Original image

ILPF of radius $D_0 = 5$

ILPF of radius $D_0 = 15$

ILPF of radius $D_0 = 30$

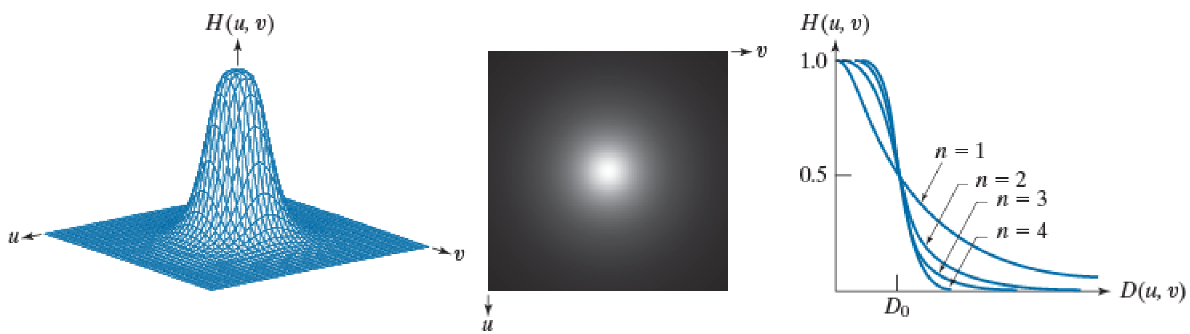
ILPF of radius $D_0 = 80$

ILPF of radius $D_0 = 230$

Butterworth Lowpass Filters

- The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$



Butterworth Lowpass Filters (cont...)

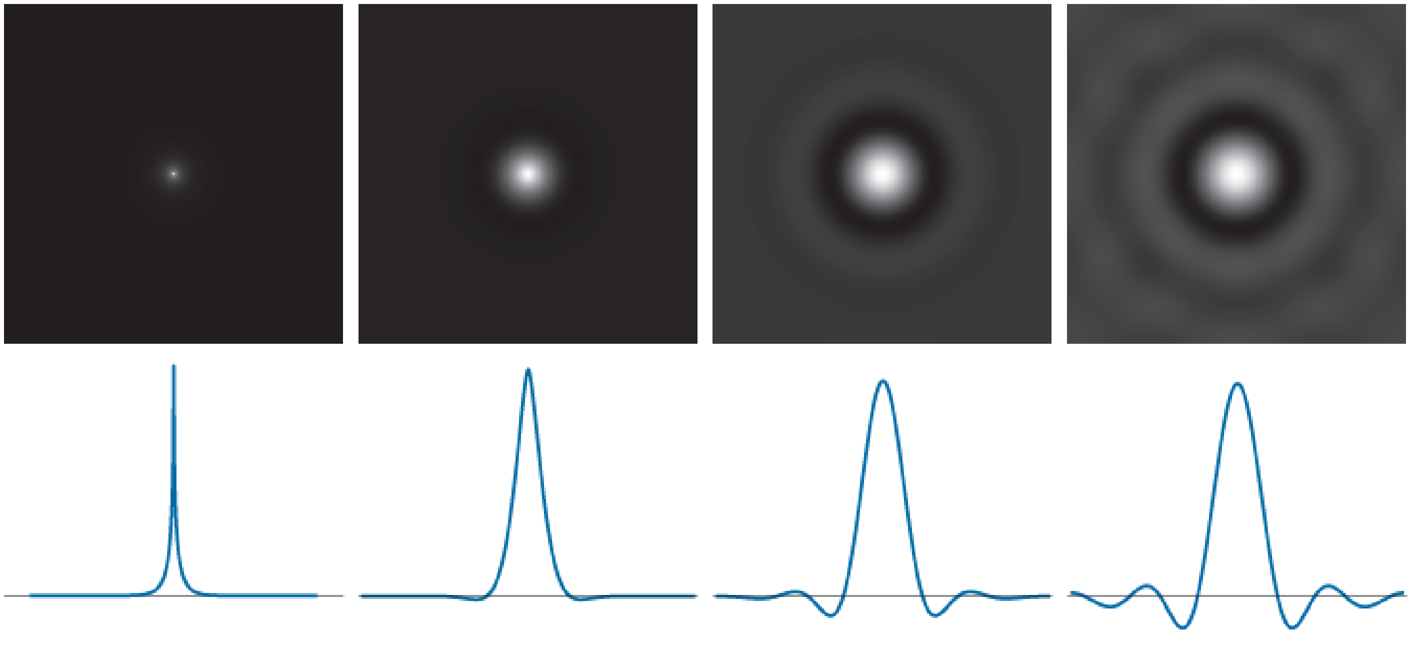
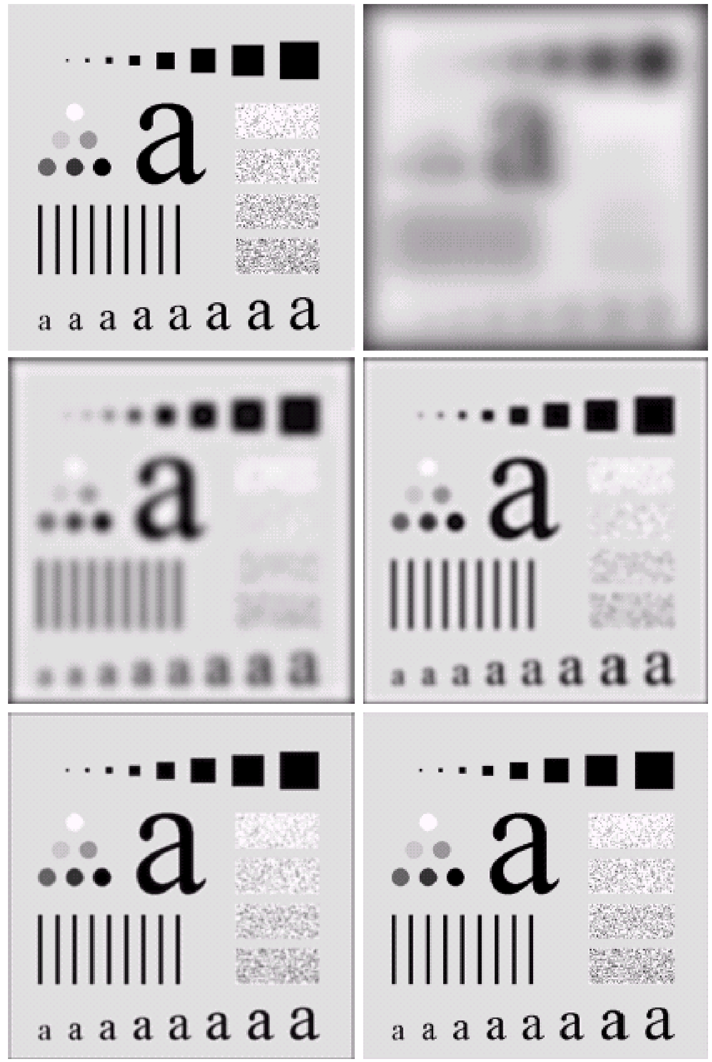


FIGURE 4.47 (a)–(d) Spatial representations (i.e., spatial kernels) corresponding to BLPF transfer functions of size 1000×1000 pixels, cut-off frequency of 5, and order 1, 2, 5, and 20, respectively. (e)–(h) Corresponding intensity profiles through the center of the filter functions.

Butterworth Lowpass Filter (cont…)

Original image

BLPF $n=2$, $D_0=5$



BLPF $n=2$, $D_0=15$

BLPF $n=2$, $D_0=30$

BLPF $n=2$, $D_0=80$

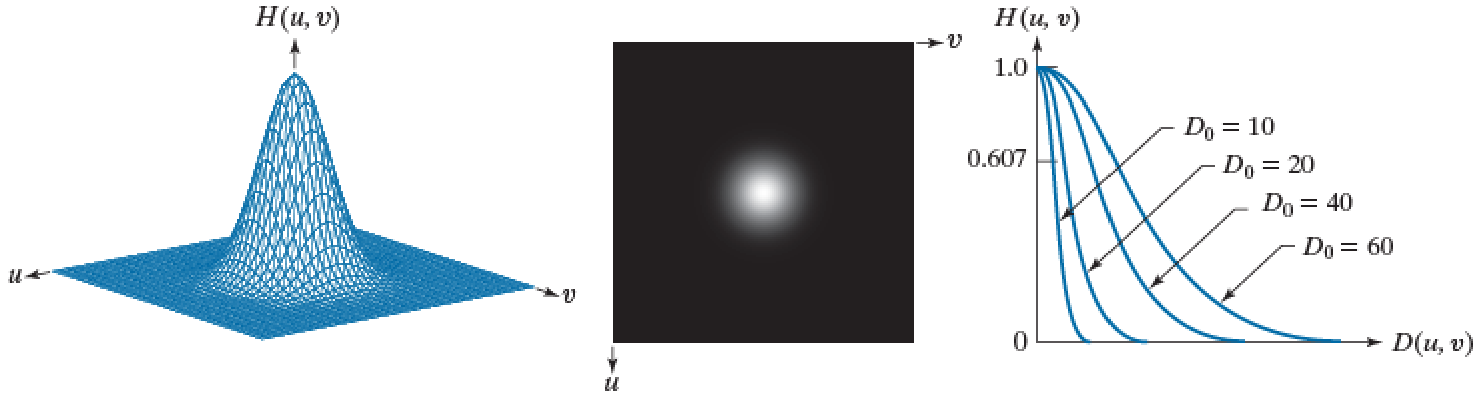
BLPF $n=2$, $D_0=230$

Less ringing than ILPF and more smoothing than the Gaussian

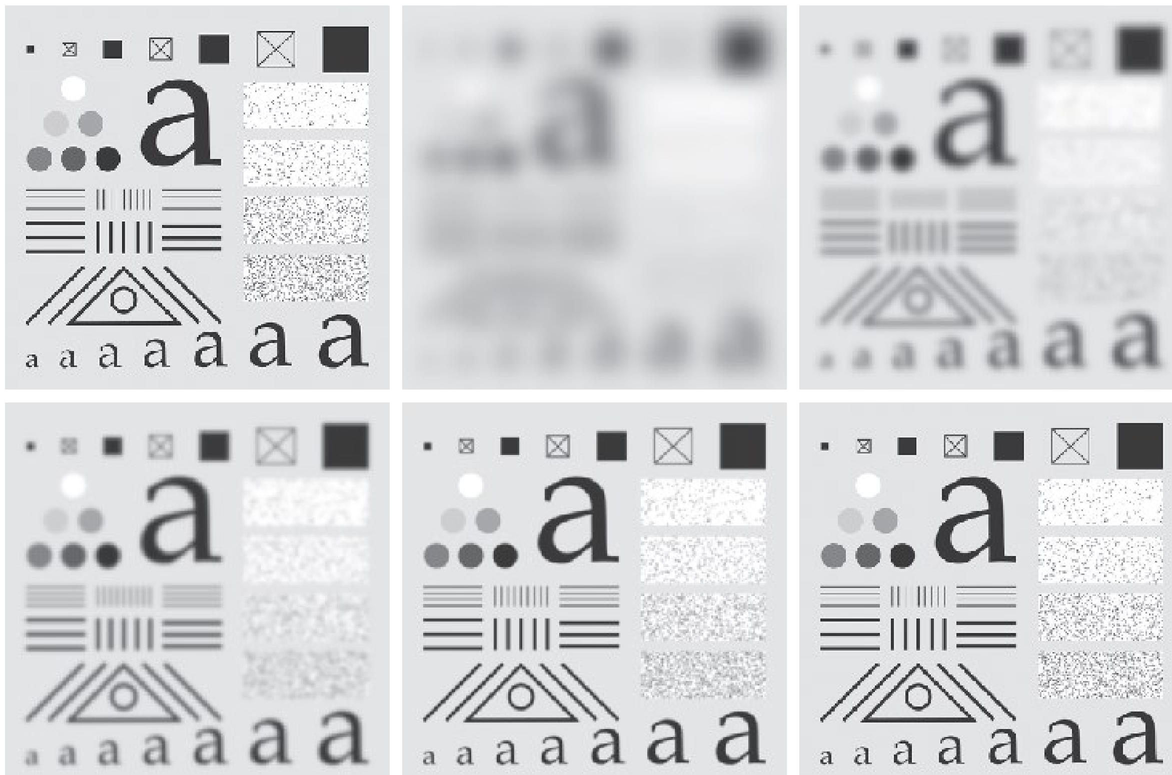
Gaussian Lowpass Filters

- The transfer function of a Gaussian lowpass filter is defined as:

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$



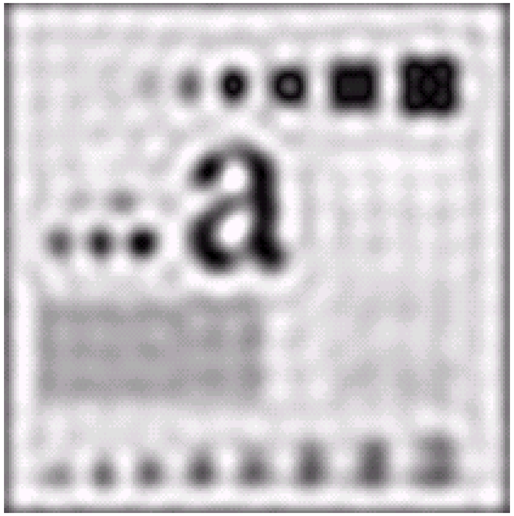
Gaussian Lowpass Filters (cont…)



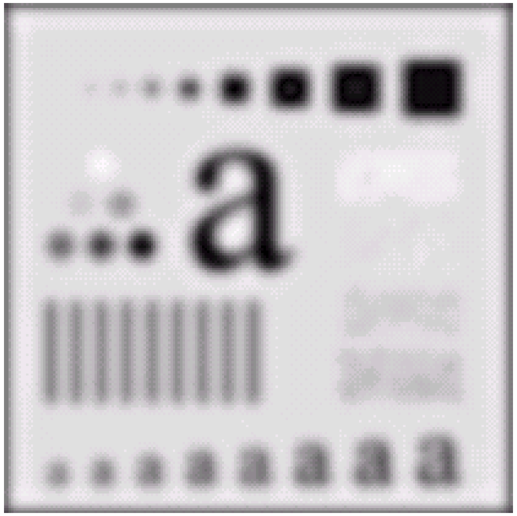
Less ringing than ILPF due to smoother transition

Lowpass Filters Compared

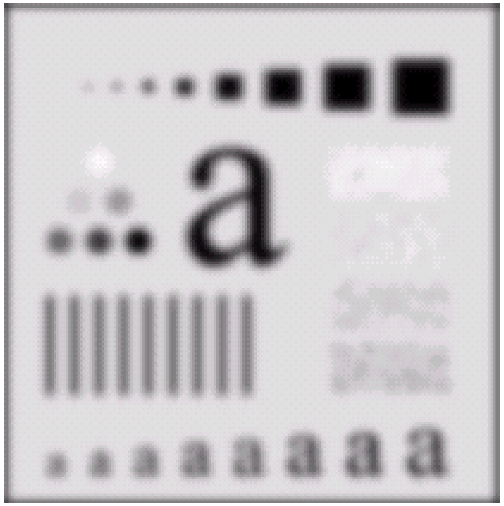
ILPF D_0=15



BLPF n=2, D_0=15

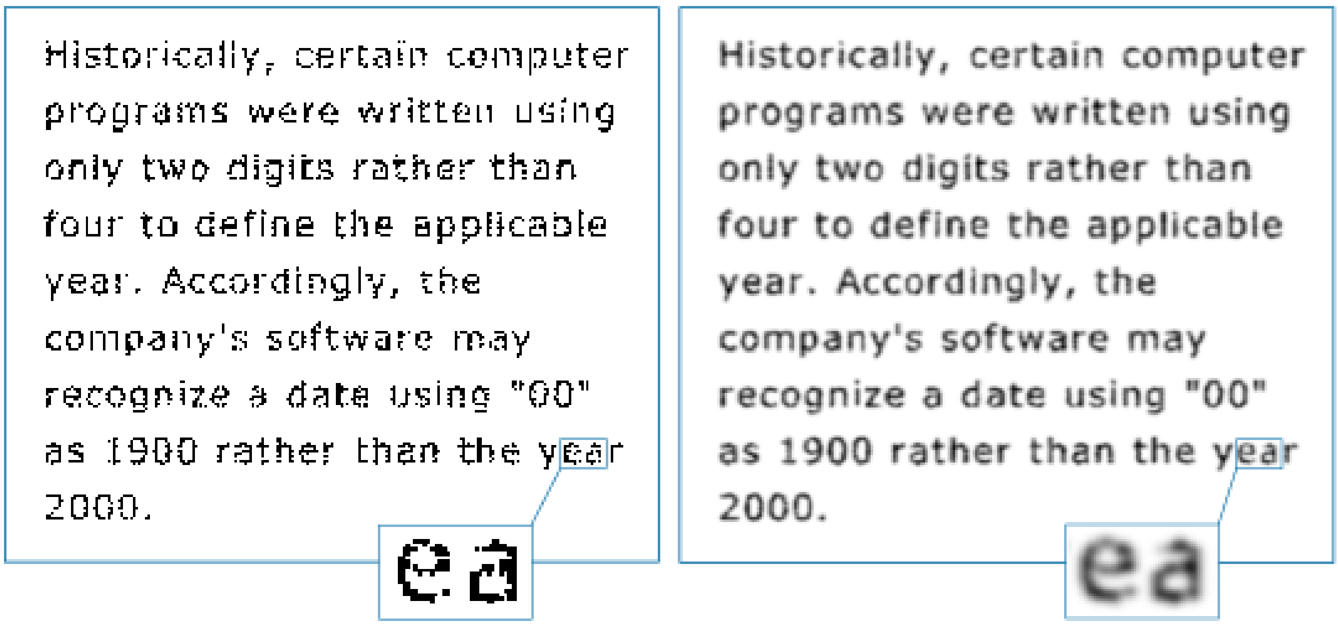


Gaussian D_0=15



Lowpass Filtering Examples

A low pass Gaussian filter is used to connect broken text



Lowpass Filtering Examples (cont…)

- Different lowpass Gaussian filters used to remove blemishes in a photograph.

Sharpening in the Frequency Domain

- Edges and fine detail in images are associated with high frequency components
- *High pass filters* – only pass the high frequencies, drop the low ones
- High pass frequencies are precisely the reverse of low pass filters, so:

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

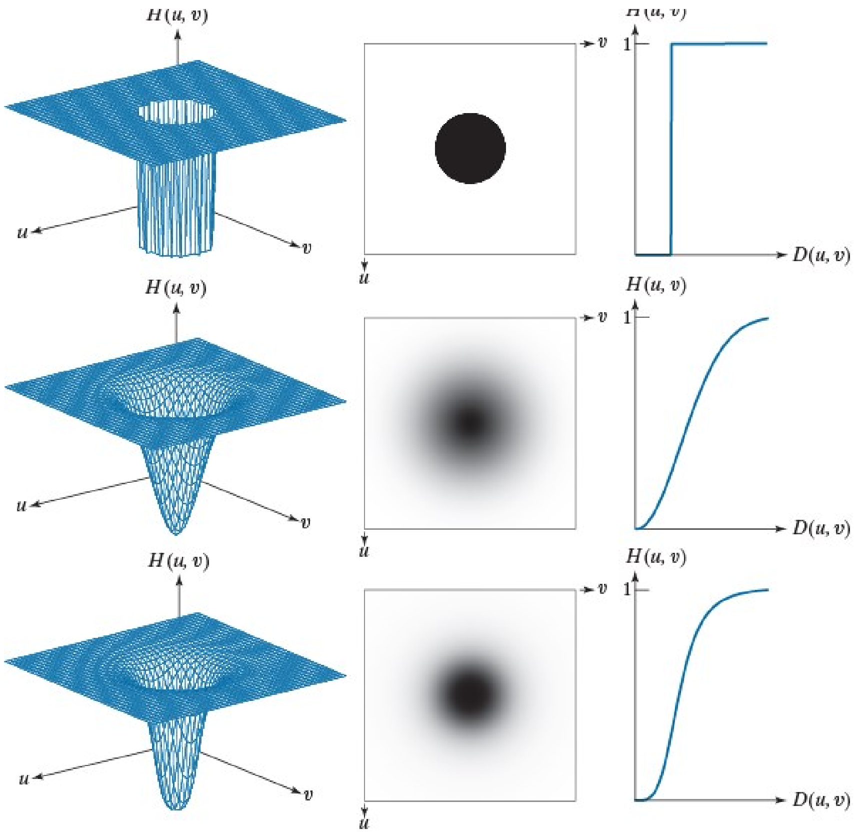
High pass filters

TABLE 4.6
Highpass filter transfer functions. D_0 is the cutoff frequency and n is the order of the Butterworth transfer function.

Ideal	Gaussian	Butterworth
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$

a b c
d e f
g h i

FIGURE 4.51 Top row: Perspective plot, image, and, radial cross section of an IHPF transfer function. Middle and bottom rows: The same sequence for GHPF and BHPF transfer functions. (The thin image borders were added for clarity. They are not part of the data.)



High pass filters (spatial domain)

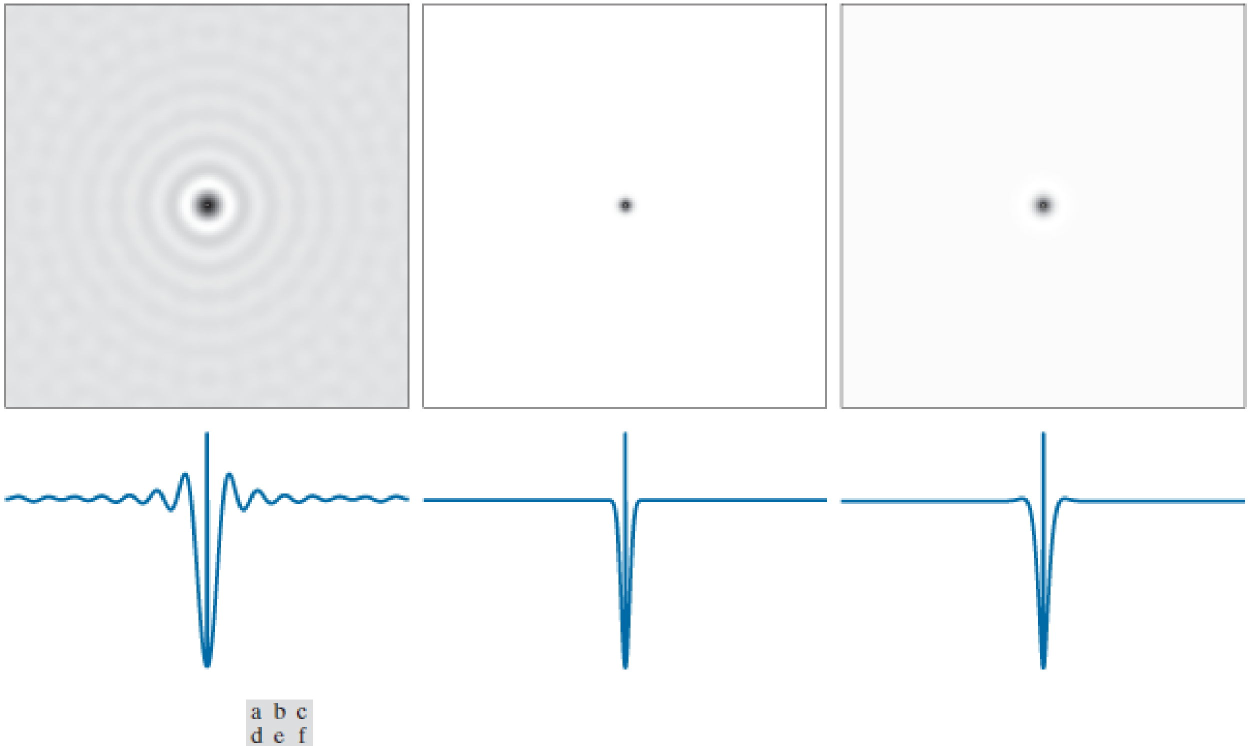
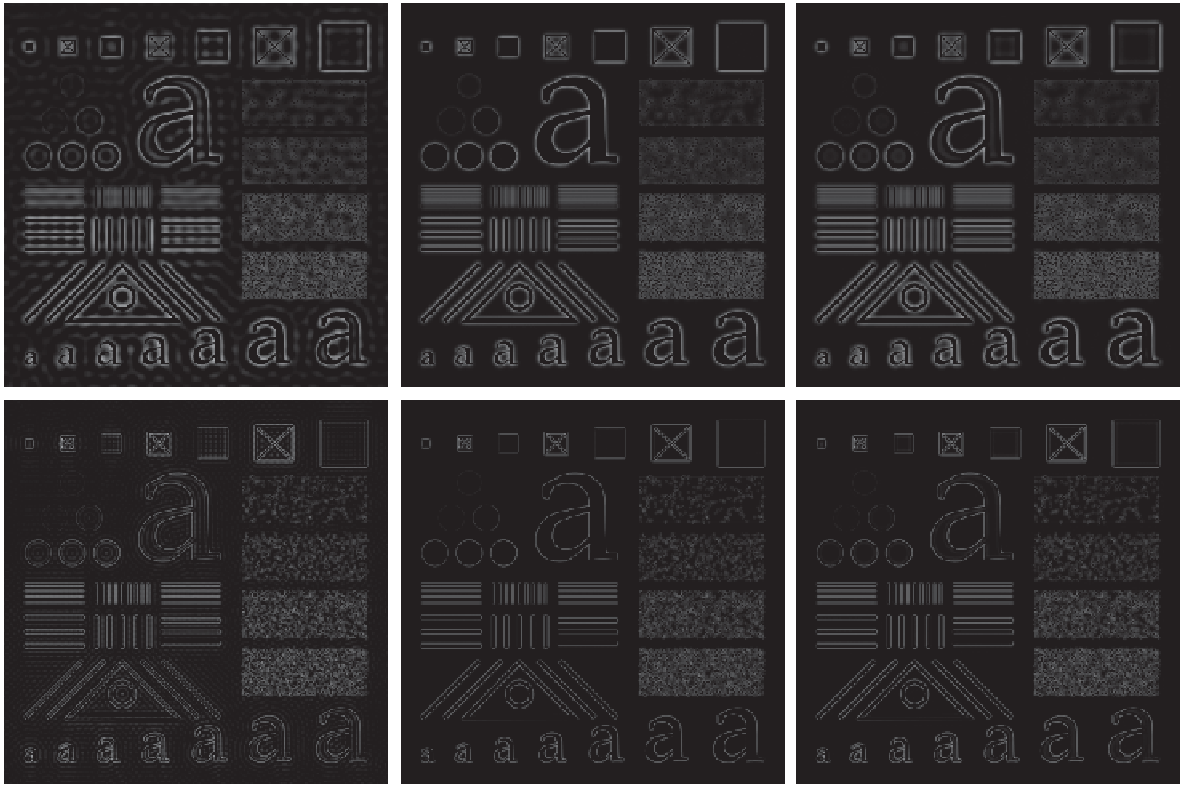


FIGURE 4.52 (a)–(c): Ideal, Gaussian, and Butterworth highpass spatial kernels obtained from IHPF, GHPF, and BHPF frequency-domain transfer functions. (The thin image borders are not part of the data.) (d)–(f): Horizontal intensity profiles through the centers of the kernels.

High pass filtering



a	b	c
d	e	f

FIGURE 4.53 Top row: The image from Fig. 4.40(a) filtered with IHPF, GHPF, and BHPF transfer functions using $D_0 = 60$ in all cases ($n = 2$ for the BHPF). Second row: Same sequence, but using $D_0 = 160$.

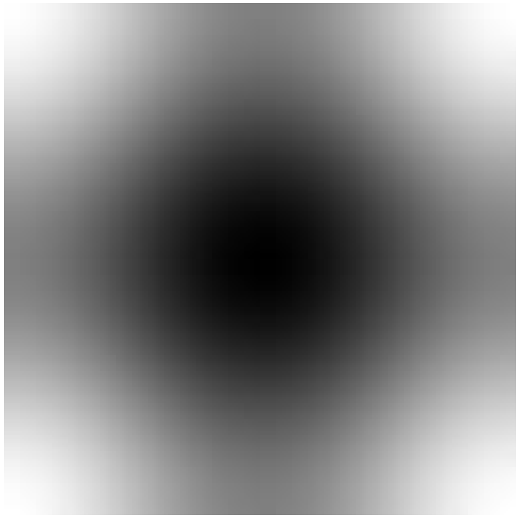
High pass filtering

High-pass filtering followed by thresholding to highlight the details

The Laplacian in the Frequency Domain

- Image enhancement operations (e.g. unsharp masking, high boost filtering) may be alternatively implemented in the frequency domain.
- Laplacian in the DFT domain:

$$H(u,v) = -4\pi^2(u^2 + v^2)$$

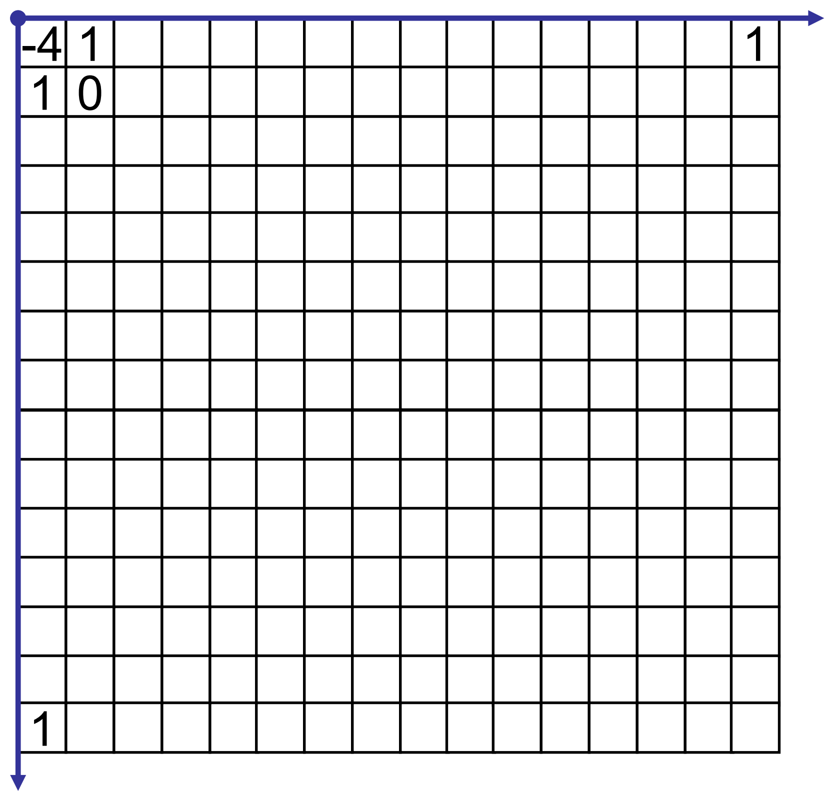


Steps of filtering in the DFT domain (cont.)

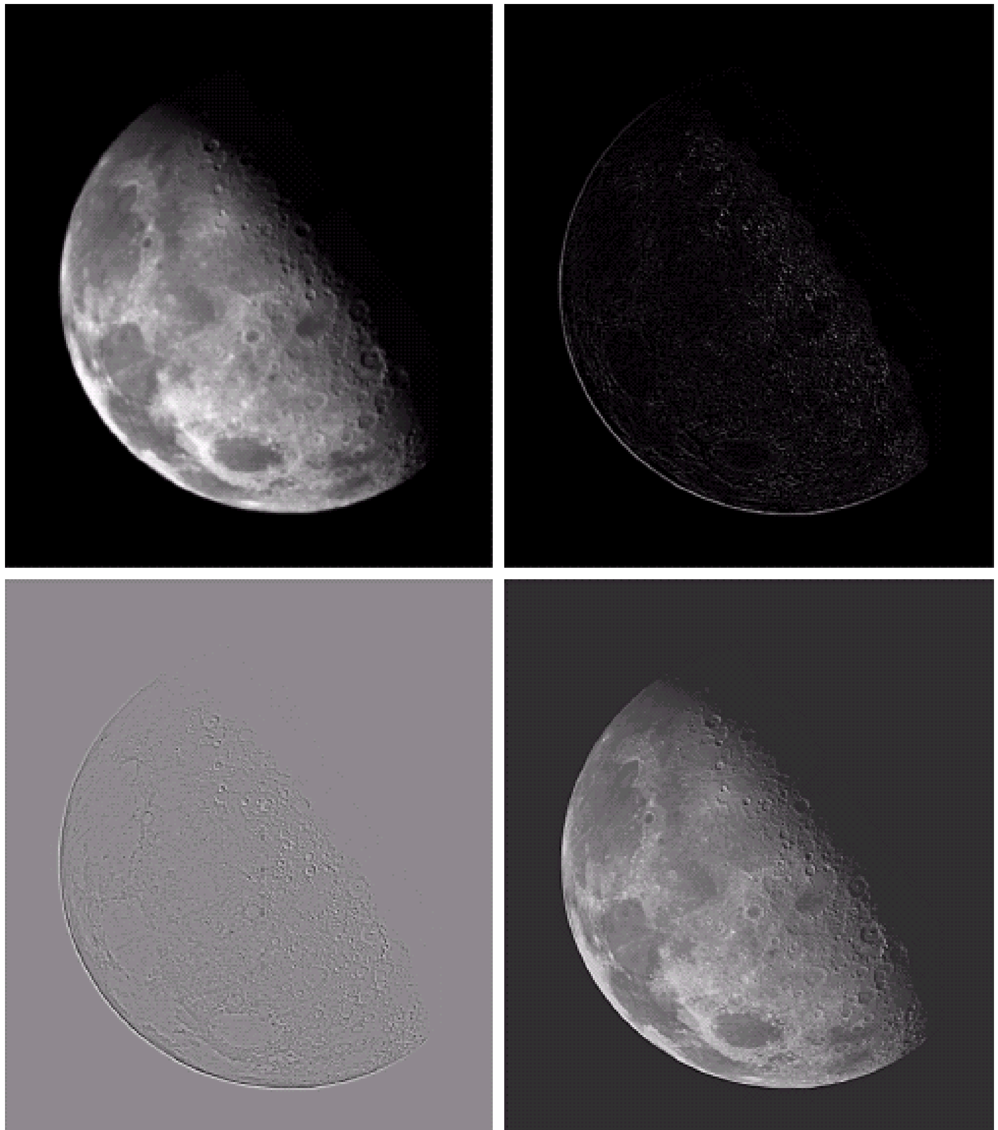
- To obtain it in the frequency domain, we zero-pad the image and repeat the content periodically.

0	1	0
1	-4	1
0	1	0

- The center of the mask is at (0,0).



Frequency Domain Laplacian Example



Original image

Laplacian filtered image

Laplacian image scaled

Enhanced image

Application – periodic noise removal

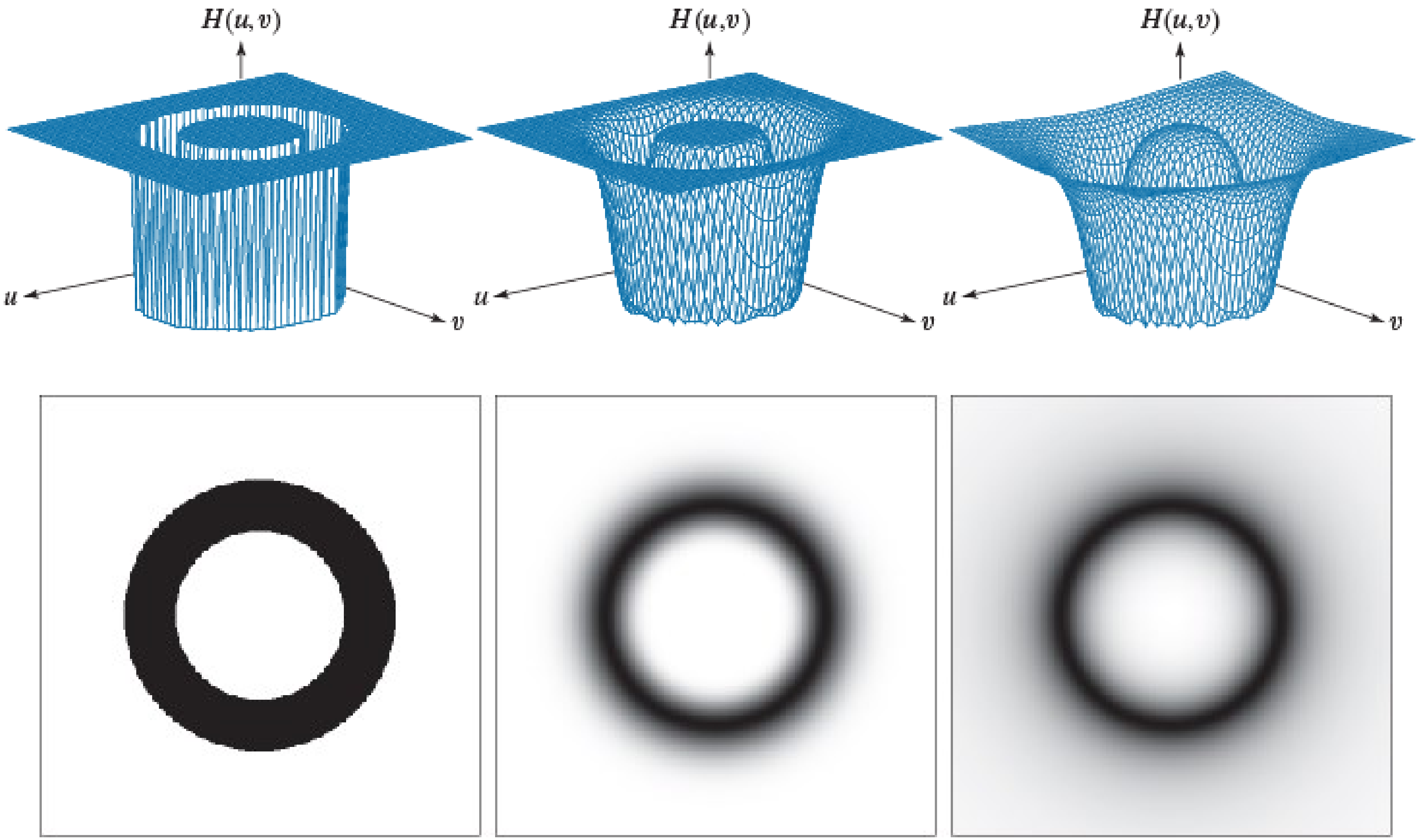
- Periodic noise or patterns often appear as a result of electrical or electromagnetic interference, Moiré patterns, etc.
- Frequency domain techniques are the most effective for addressing such types of noise.
- Some synonymous/similar techniques include:
  - Bandpass filters (ζωνοπερατά φίλτρα)
  - Notch filters (φίλτρα εγκοπής)
  - Bandreject/stop filters (ζωνοφρακτικά φίλτρα)

Band-Stop Filters

- Removal of periodic noise involves eliminating a specific range of frequencies from the image.
- This is typically achieved using bandstop filters. Conversely, bandpass filters are used to isolate specific frequency ranges.
- An ideal band reject filter is an example of a bandstop filter.

$$H(u,v)=\begin{cases}1 & if\ D(u,v)<D_0-\dfrac{W}{2}\\ 0 & if\ D_0-\dfrac{W}{2}\leq D(u,v)\leq D_0+\dfrac{W}{2}\\ 1 & if\ D(u,v)>D_0+\dfrac{W}{2}\end{cases}$$

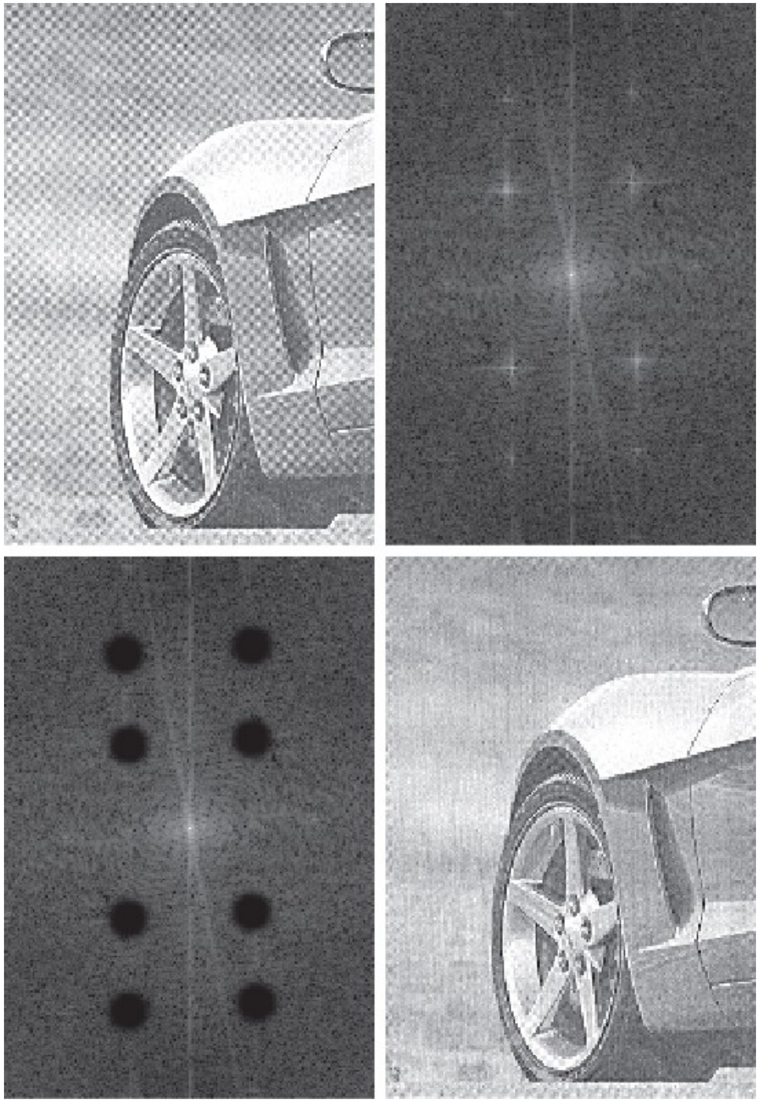
Band-Stop Filters



Band-Stop Filters (cont...)

a b
c d

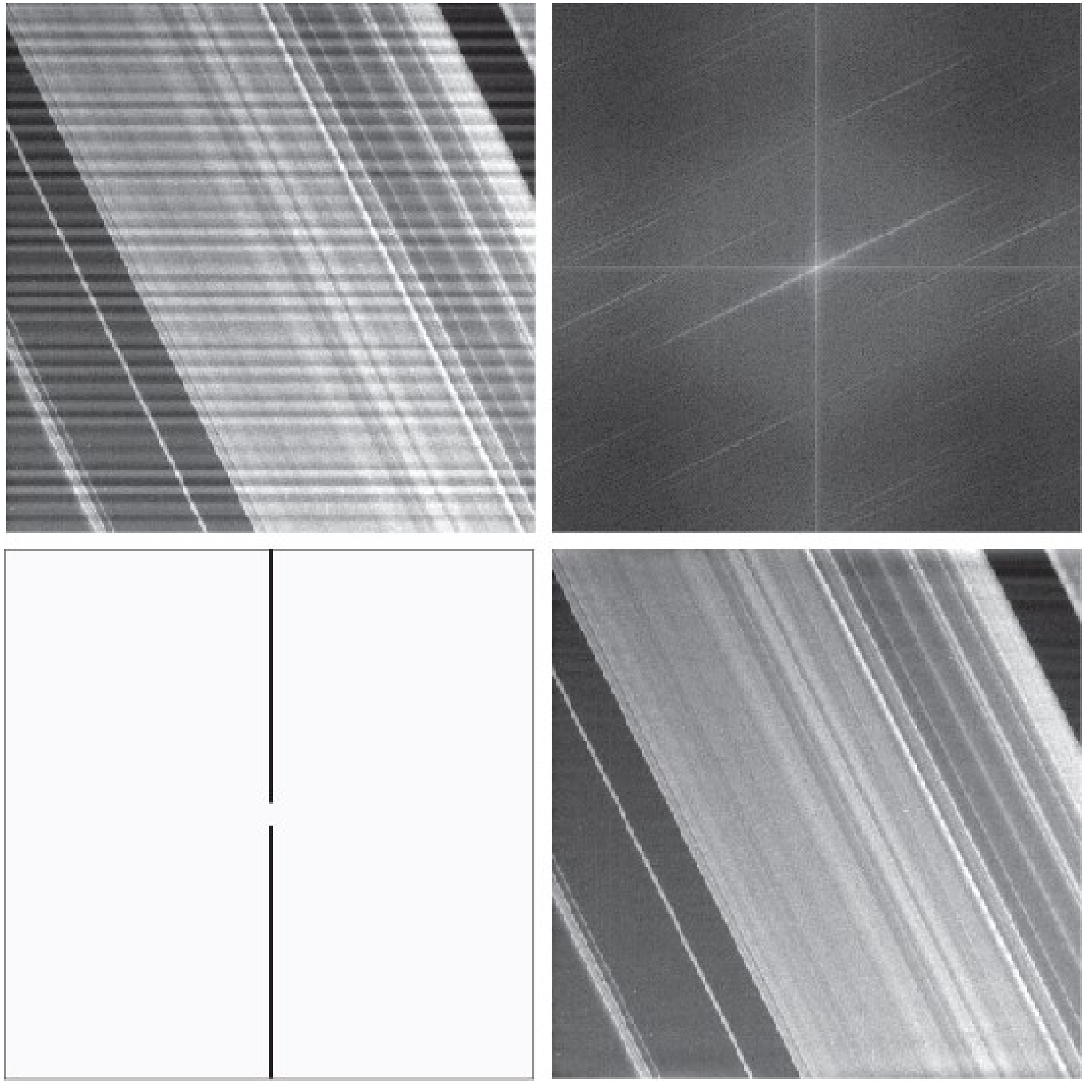
FIGURE 4.64 (a) Sampled newspaper image showing a moiré pattern. (b) Spectrum. (c) Fourier transform multiplied by a Butterworth notch reject filter transfer function. (d) Filtered image.



Band-Pass Filters (cont...)

a b
c d

FIGURE 4.65
(a) Image of Saturn rings showing nearly periodic interference.
(b) Spectrum. (The bursts of energy in the vertical axis near the origin correspond to the interference pattern).
(c) A vertical notch reject filter transfer function.
(d) Result of filtering.
(The thin black border in (c) is not part of the data.) (Original image courtesy of Dr. Robert A. West, NASA/JPL.)



Fast Fourier Transform

- The reason that Fourier based techniques have become so popular is the development of the **Fast Fourier Transform (FFT)** algorithm.
- It allows the Fourier transform to be carried out in a reasonable amount of time.
- Reduces the complexity from $O(N^4)$ to $O(N^2 \log N^2)$.

Frequency Domain Filtering & Spatial Domain Filtering

- Similar jobs can be done in the spatial and frequency domains.
- Filtering in the spatial domain can be easier to understand.
- Filtering in the frequency domain can be much faster – especially for large images.

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Image Restoration and Reconstruction (Noise Removal)

Άγγελος Γιώτης

a.giotis@uoi.gr

Image Restoration and Reconstruction

Things which we see are not by themselves what we see…

It remains completely unknown to us what the objects may be by themselves and apart from the receptivity of our senses. We know nothing but our manner of perceiving them.

Immanuel Kant

Contents

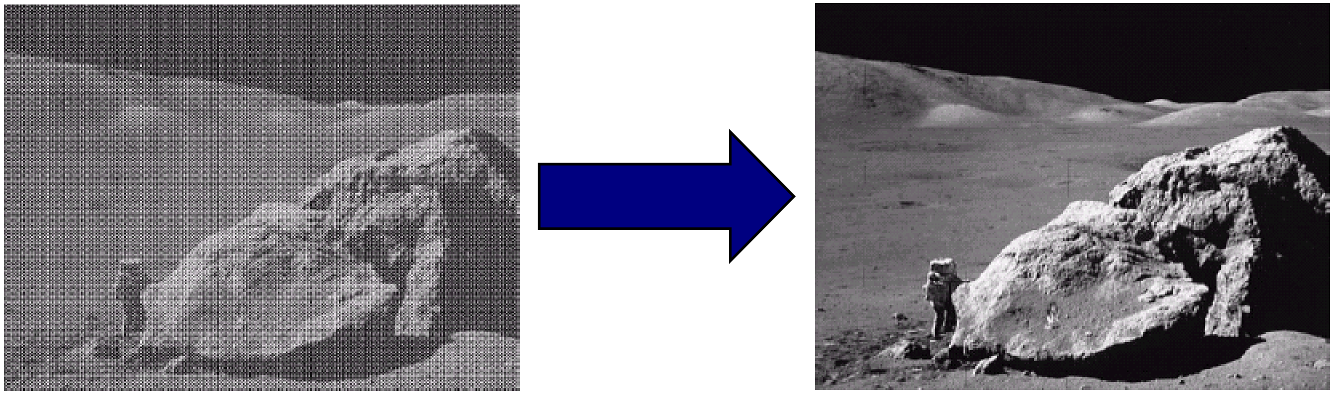
In this lecture we will look at image restoration techniques used for noise removal

- What is image restoration?
- Noise and images
- Noise models
- Noise removal using spatial domain filtering
- Noise removal using frequency domain filtering

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission

Noise (observation) Model

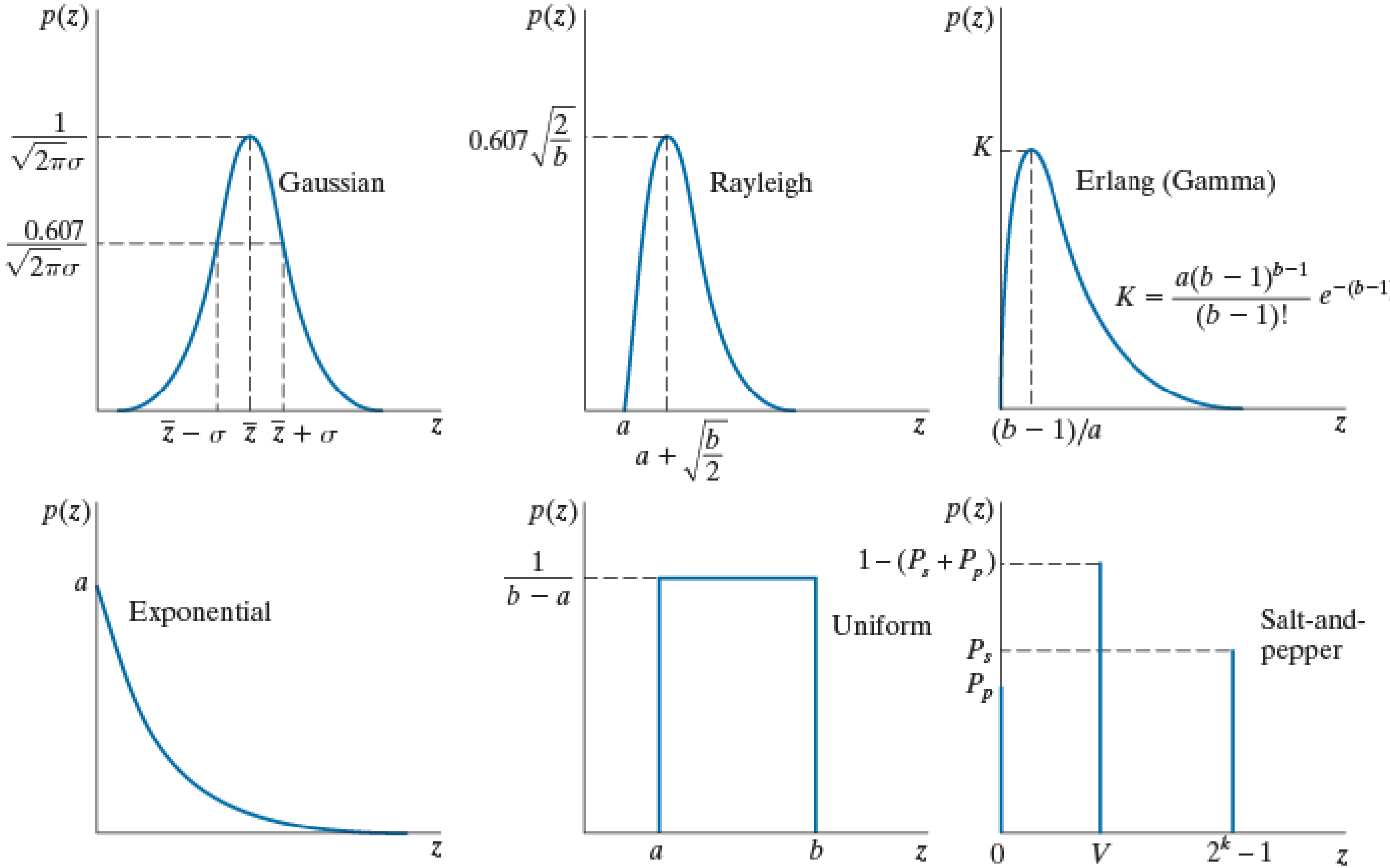
- We can model the observed image with respect to noise as:

$$g(x, y) = \mu(f(x, y), \eta(x, y))$$

where $f\ (x, y)$ is the original image pixel (or image),
$\eta(x, y)$ is the noise term (for the pixel/image) and
$g(x, y)$ is the resulting noisy pixel (or image)

- The model $\mu()$ describes the observation process: it specifies how we believe noise (and possibly other factors) affect the image.
- If we can estimate $\mu()$ we can figure out how to restore the image
- Many different ways to define η and μ

Noise Models (cont...)



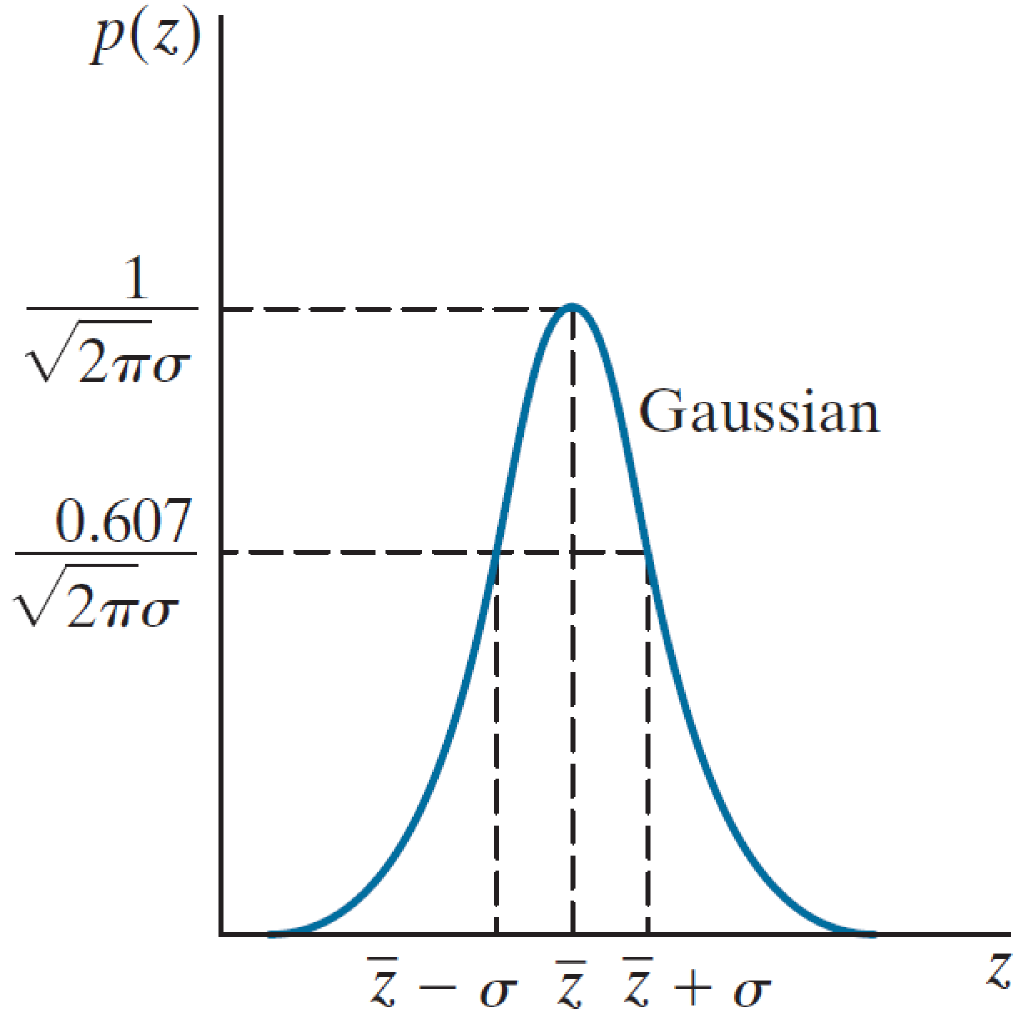
Noise Models (cont...)

There are many ways to define random variable $z = \eta(x, y)$.

The probability density function of a Gaussian z:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

$$-\infty < z < \infty$$



Noise Models (cont...)

There are many ways to define random variable $z = \eta(x,y)$.

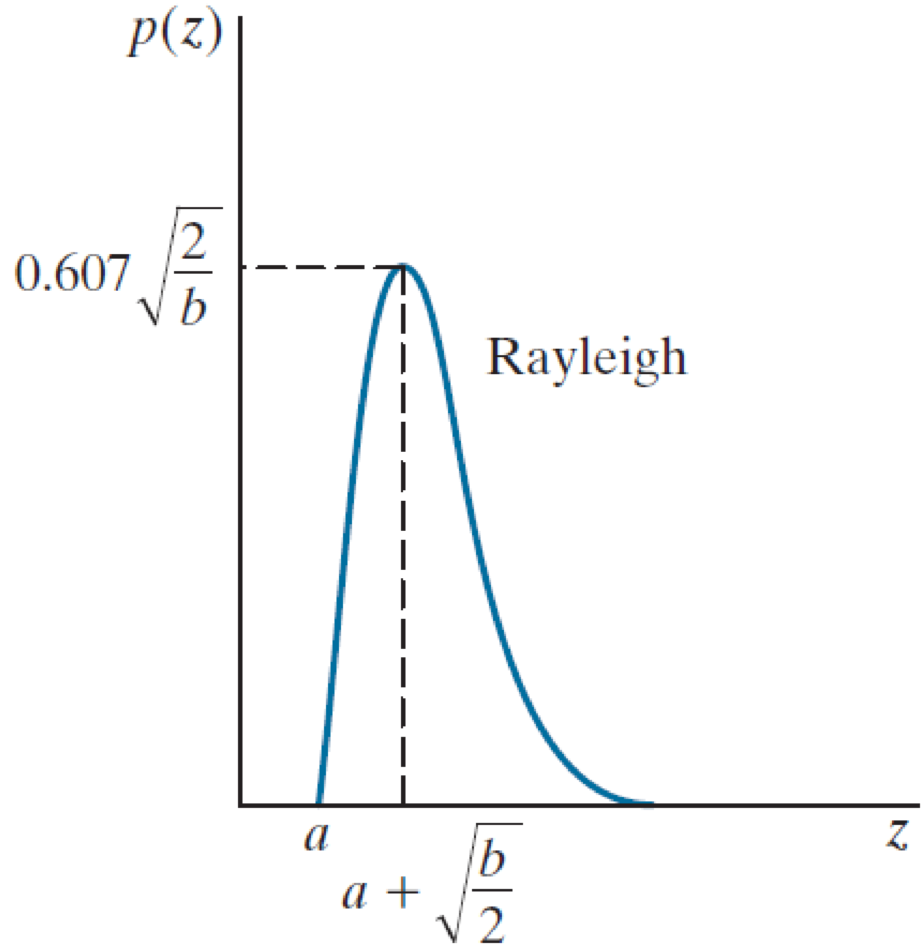
The probability density function of a Rayleigh z:

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$$

The mean and variance of z:

$$\bar{z} = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$



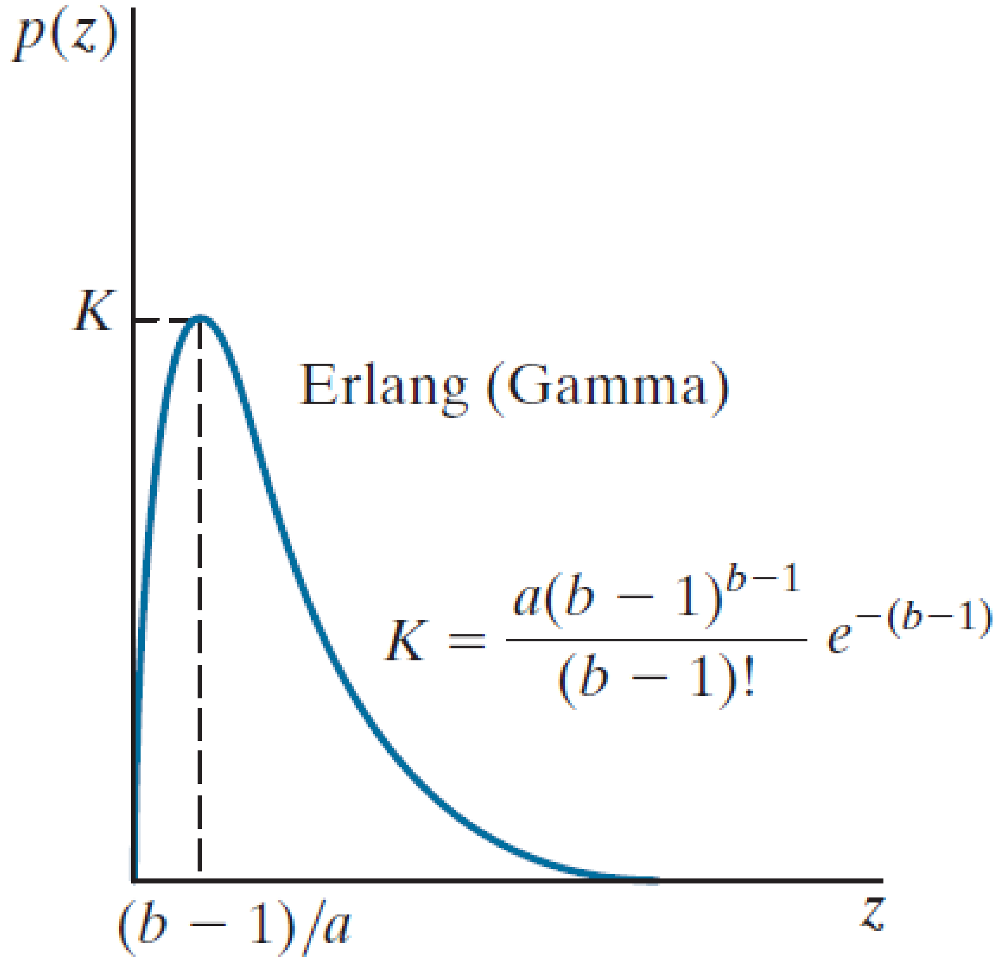
Noise Models (cont...)

- There are many ways to define random variable $z = \eta(x, y)$.
- The probability density function of a Gamma (or Erlang) z:

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

The mean and variance of z:

$$\bar{z} = \frac{b}{a} \qquad \sigma^2 = \frac{b}{a^2}$$



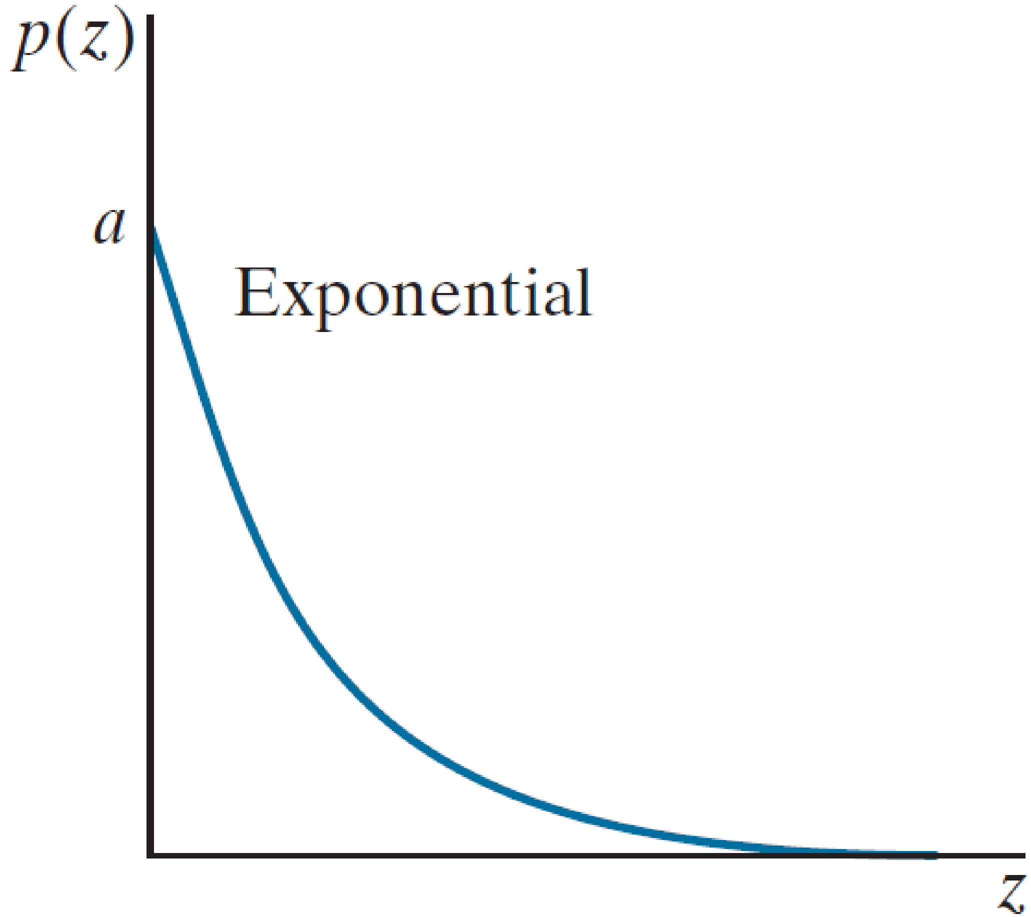
Noise Models (cont...)

- There are many ways to define random variable $z = \eta(x, y)$.
- The probability density function of an exponential z:

$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

The mean and variance of z:

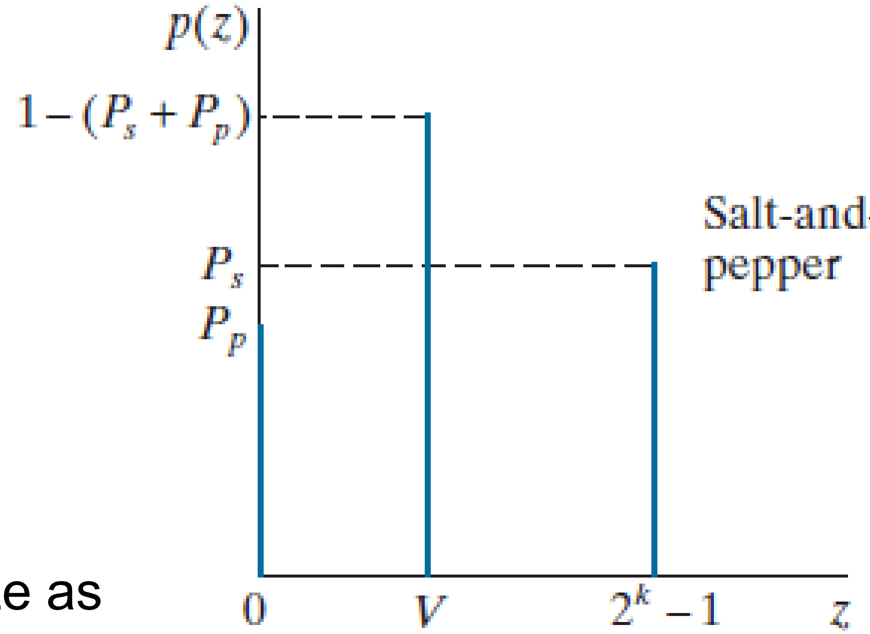
$$\bar{z} = \frac{1}{a} \qquad \sigma^2 = \frac{1}{a^2}$$



Noise Models (Salt and pepper)

- The probability density function of an impulse z (or salt-and-pepper noise):

$$p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1 - (P_s + P_p) & \text{for } z = V \end{cases}$$



Given an image *f(x, y)*, of the same size as *η(x, y)*, we corrupt it with salt-and-pepper noise by assigning a 0 to all locations in *f* where a 0 occurs in *η*. Similarly, we assign a value of $2^k - 1$ to all location in *f* where that value appears in *η*. Finally, we leave unchanged all locations in *f* where V occurs in *η*.

If neither P_s nor P_p is zero, and especially if they are equal, noise values will be white ($2^k - 1$) or black (0), and will resemble salt and pepper granules distributed randomly over the image

Standard Additive noise model

- We can consider a noisy image to be modelled as follows:

$$g(x,y) = f(x,y) + \eta(x,y)$$

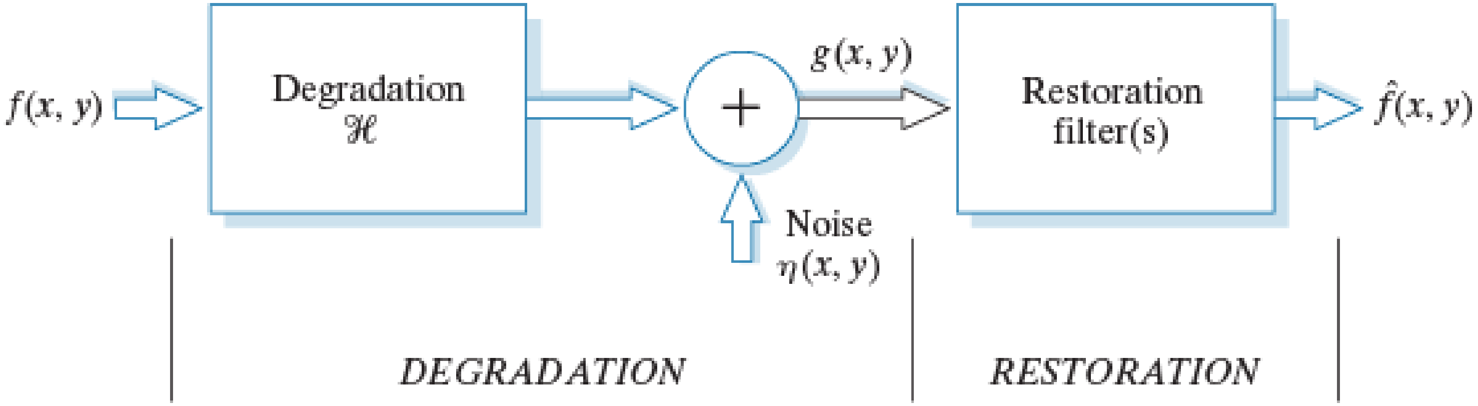
- For noise, we assume that for each point n in the image:

$E[\eta_n] = 0$ and $E[\eta_n^2] = \sigma_\eta^2$

whereas noise is **uncorrelated** between different points m, n:

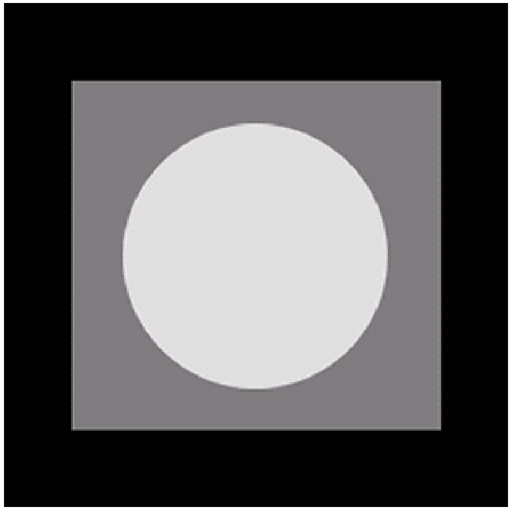
$E[\eta_m \eta_n] = 0$

- This is known as the "**white Gaussian noise**" assumption.

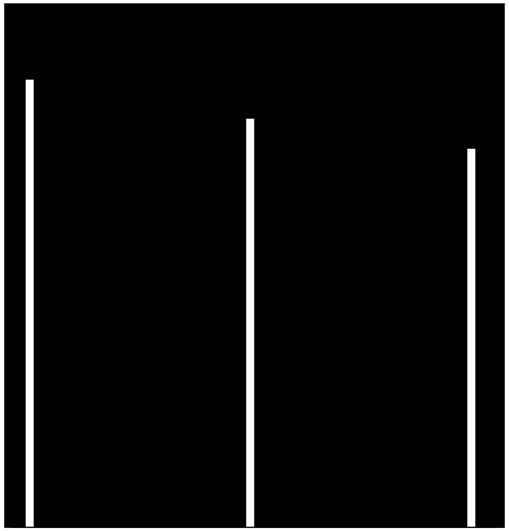


Noise Example

- The test pattern to the right is ideal for demonstrating the addition of noise
- The following slides will show the result of adding noise based on various models to this image



Image



Histogram

Noise Example (cont…)

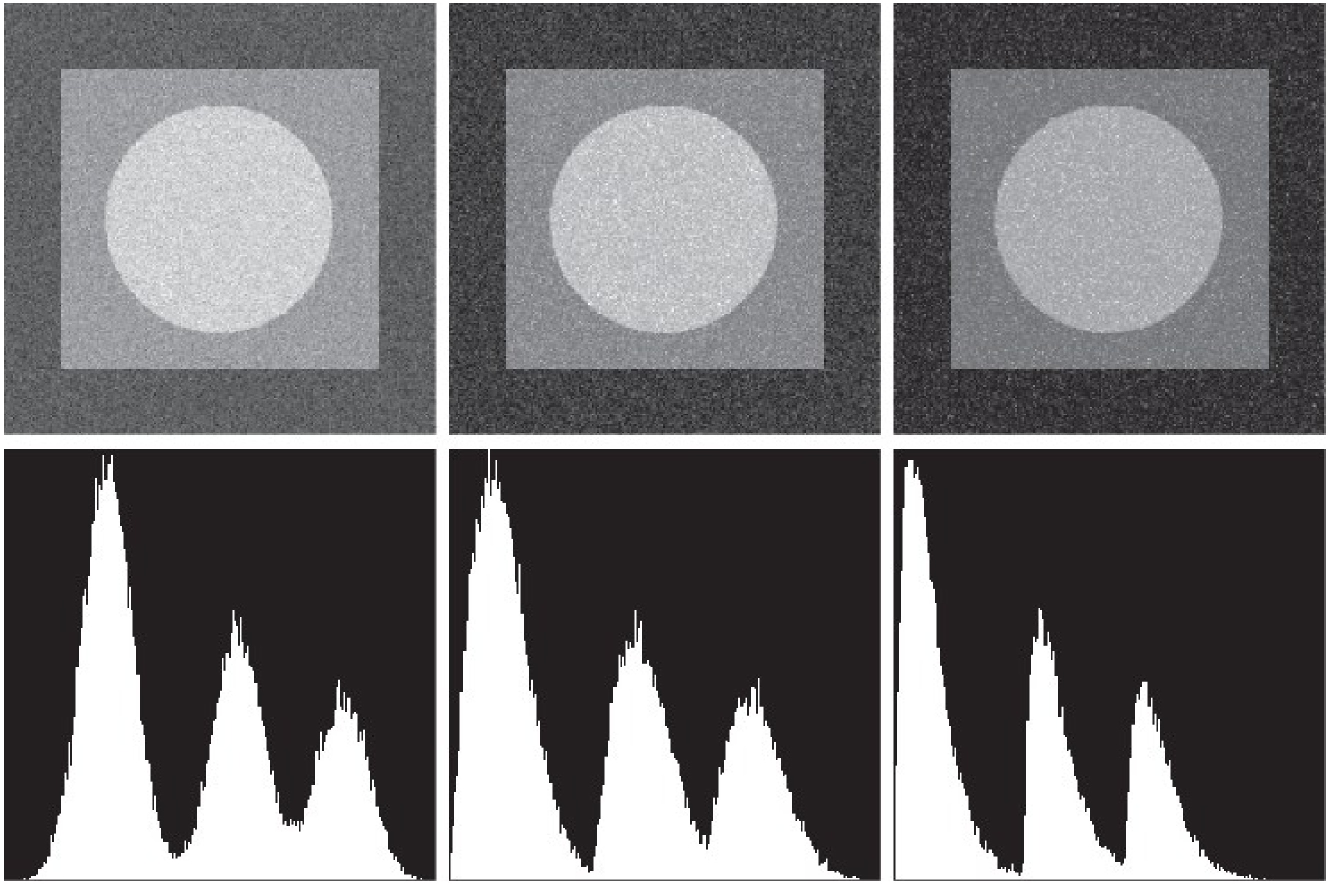


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and Erlanga noise to the image in Fig. 5.3.

Noise Example (cont…)

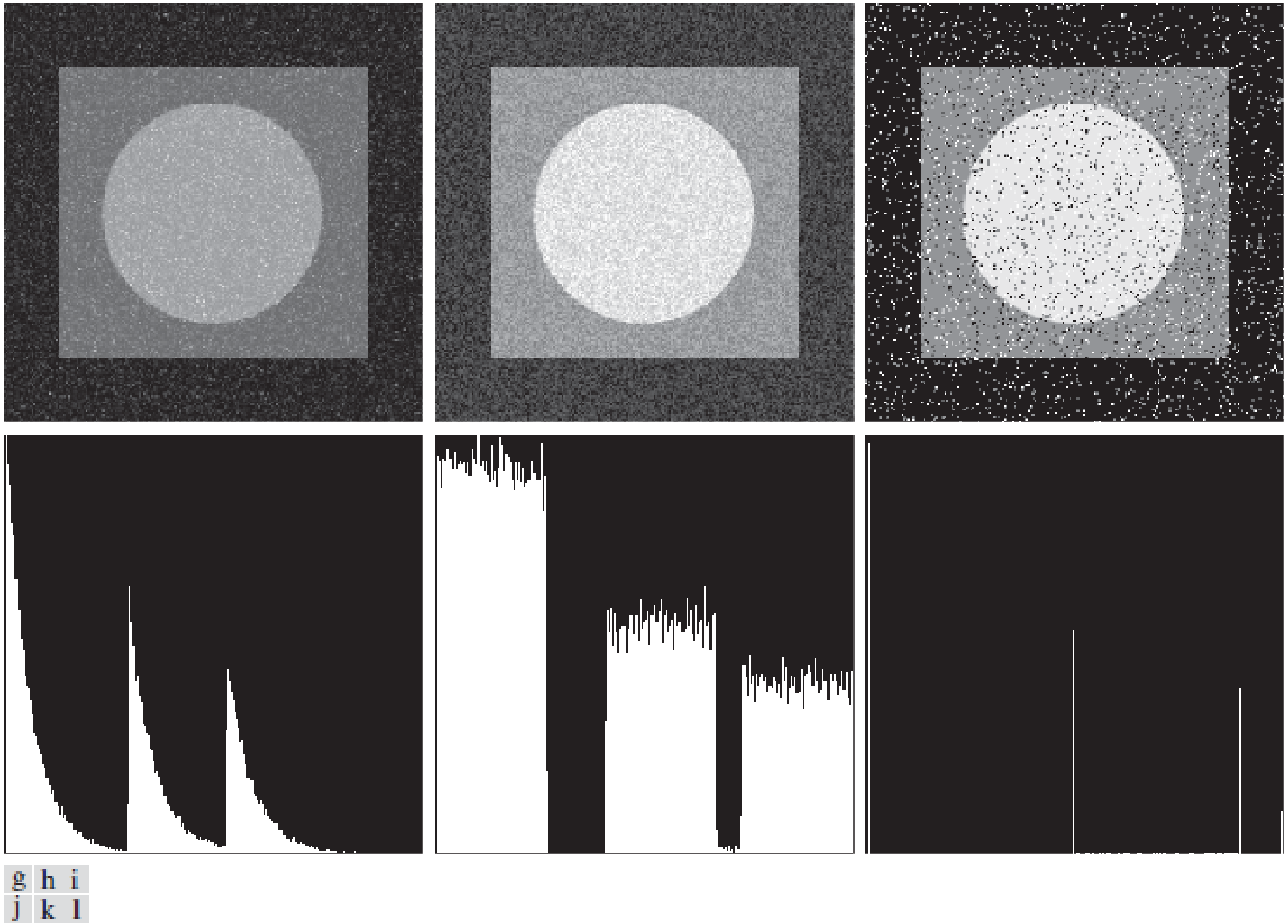


FIGURE 5.4 (*continued*) Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the image in Fig. 5.3. In the salt-and-pepper histogram, the peaks in the origin (zero intensity) and at the far end of the scale are shown displaced slightly so that they do not blend with the page background.

Restoration in the presence of noise only

- We can use **spatial filters** of different kinds to remove different kinds of noise
- The *arithmetic mean* filter is a very simple one and is calculated as follows:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

This is implemented as the simple smoothing filter

It blurs the image.

Example Restoration using combination of observations

- Suppose:

$$\hat{f}(x,y) = \bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

- Which is our estimation for the actual image
- Note that pixel intensities for each point in the 2D image can be seen as random variables
- We can demonstrate that, by calculating the expected value and the standard deviation of the pixel intensities of the estimated image, it will have less noise than that in the original image.

Example Restoration using combination of observations

- Expected value:

$$E\left[\bar{g}(x,y)\right] = E\left[\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)\right] = \frac{1}{K}E\left[\sum_{i=1}^{K} g_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y) + \eta_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y)\right] + \frac{1}{K}E\left[\sum_{i=1}^{K} \eta_i(x,y)\right]$$

$$= \frac{1}{K}Kf(x,y) + \frac{1}{K}K0 \;= f(x,y)$$

$$\mathrm{E}[f(x,y) + \cdots + f(x,y)] =$$
$$= E[Kf(x,y)] = KE[f(x,y)]$$
$$= K\, f(x,y)$$

Example Restoration using combination of observations

- Similarly, the variance of the new image is:

$$\sigma_{\hat{f}}^2 = \sigma_{\bar{g}}^2 = E[\bar{g}^2] - E[\bar{g}]^2 = \frac{1}{\mathrm{K}}\sigma_\eta^2$$

- Remember white Gaussian noise assumption: $E[\eta_n^2] = \sigma_\eta^2$ (do the rest as HW)
- As *K* increases, the variability of the brightness of each image element decreases, resulting in approaching the ideal noise-free image f(x, y).
- Note that images (f and η) must be aligned (registered)

Example Restoration using smoothing

- If we only have one observation/image available, what can we do?
- Suppose we apply a smoothing filter h. We can represent the process as a convolution with the filter:

$$\hat{f} = h * (f + \eta) = h * f + h * \eta$$

- Again, we compute the same statistical moments. That is, we calculate the **expected value**:

$$\mathrm{E}[\hat{f}] = \mathrm{E}[h * f + h * \eta] = h * E[f] = h * f$$

Example Restoration using smoothing

- And the **variance** (and standard deviation), for which we assume:

$$\hat{f} = h * f + h * \eta = \bar{f} + \bar{\eta}$$

- $$\sigma_{\hat{f}}^2 = E[\hat{f}^2] - E[\hat{f}]^2$$

$$= E\left[\left(\bar{f} + \bar{\eta}\right)^2\right] - \left(\bar{f}\right)^2 = E[(\bar{\eta})^2]$$

Example Restoration using smoothing

- Assuming that h is a mean filter, for pixel n, we have:

$$\bar{\eta}(n) = (h * \eta)(n) = \frac{1}{N} \sum_{k \in \Gamma(n)} \eta(k)$$

- Hence:

$$\mathrm{E}[\bar{\eta}(n)^2] = \mathrm{E}[\left(\frac{1}{N} \sum_{k \in \Gamma(n)} \eta(k) \right)^2]$$

Example Restoration using smoothing

- Hence:

$$E\left[\left(\frac{1}{N}\sum_{k\in\Gamma(n)}\eta(k)\right)^2\right]=\frac{1}{N^2}\sum_{k\in\Gamma(n)}E[\eta(k)^2]$$

$$+\frac{2}{N^2}\sum_{l\in\Gamma(n)}\sum_{\substack{m\in\Gamma(n)\\ m\neq l}}E[\eta(n-l)\eta(n-m)]$$

$$=\frac{1}{N^2}\sum_{k\in\Gamma(n)}E[\eta(k)^2]=\frac{1}{N^2}\sum_{k\in\Gamma(n)}\sigma_\eta^2$$

Example Restoration using smoothing

- Consequently:

$$\sigma_{\hat{f}}^2 = \frac{1}{N^2} \sum_{k \in \Gamma(n)} \sigma_\eta^2 = \frac{1}{N^2} \mathrm{N} \sigma_\eta^2 = \frac{1}{\mathrm{N}} \sigma_\eta^2$$

- Positive: The effect of noise is reduced in the estimated image.
- Negative: Unfortunately, since we are using a smoothing filter, along with the noise, we also lose potentially significant details, such as edges.

Restoration in the presence of noise only (continue..)

- We can use **spatial filters** of different kinds to remove different kinds of noise
- The *arithmetic mean* filter is a very simple one and is calculated as follows:

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

This is implemented as the simple smoothing filter

It blurs the image.

Restoration in the presence of noise only (cont.)

- There are different kinds of mean filters all of which exhibit slightly different behaviour:
  - Geometric Mean
  - Harmonic Mean
  - Contraharmonic Mean

Restoration in the presence of noise only (cont.)

Geometric Mean:

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

- Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail.

Restoration in the presence of noise only (cont.)

Harmonic Mean:

$$\hat{f}(x,y) = \frac{mn}{\displaystyle\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

- Works well for salt noise, but fails for pepper noise.
- Also does well for other kinds of noise such as Gaussian noise.

Restoration in the presence of noise only (cont.)

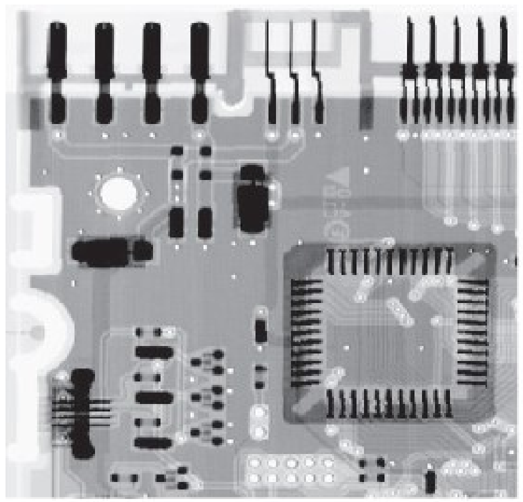
Contraharmonic Mean:

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

- Q is the order of the filter.
- Positive values of Q eliminate pepper noise.
- Negative values of Q eliminate salt noise.
- It cannot eliminate both simultaneously.

Noise Removal Examples

Original image



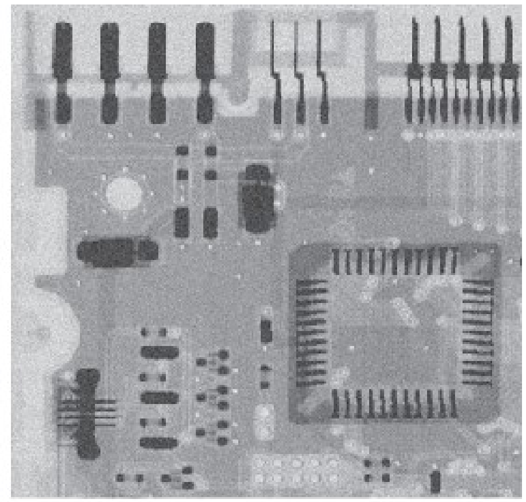
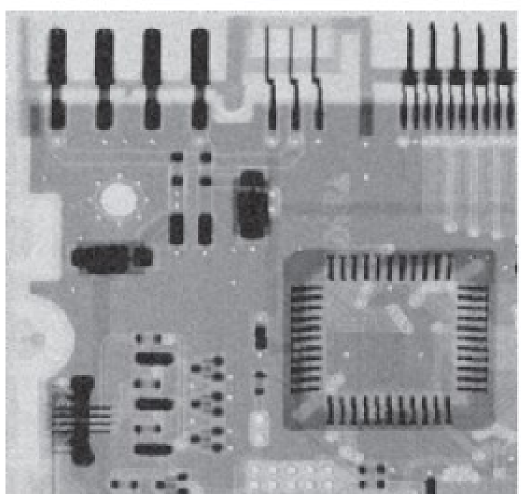
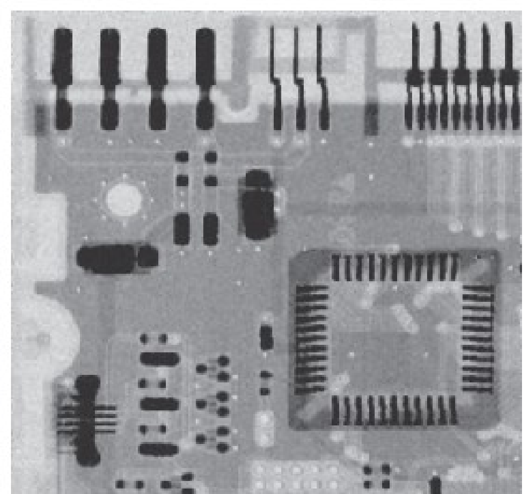


Image corrupted by Gaussian noise

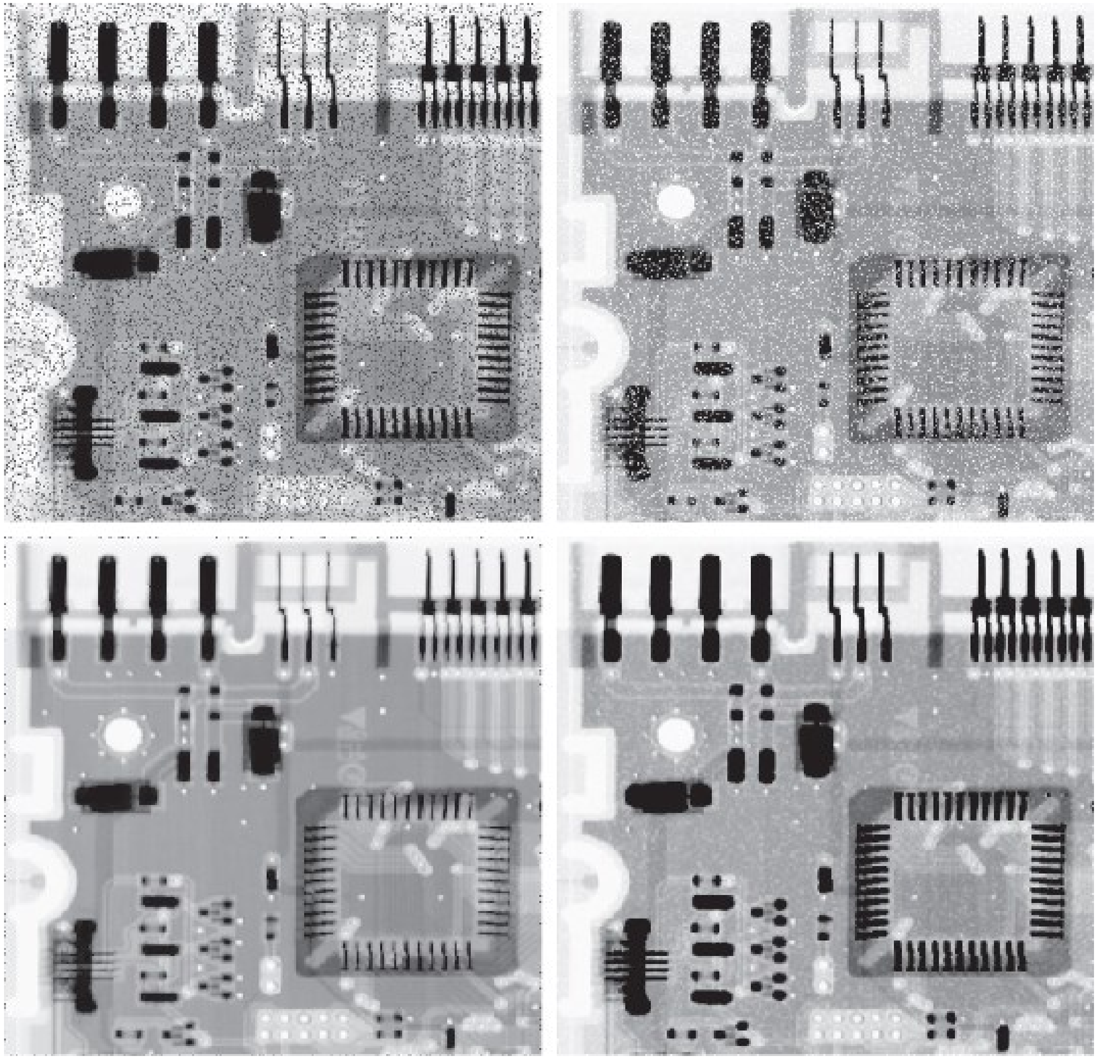
3x3 Arithmetic Mean Filter





3x3 Geometric Mean Filter (less blurring than AMF, the image is sharper)

Noise Removal Examples (cont…)

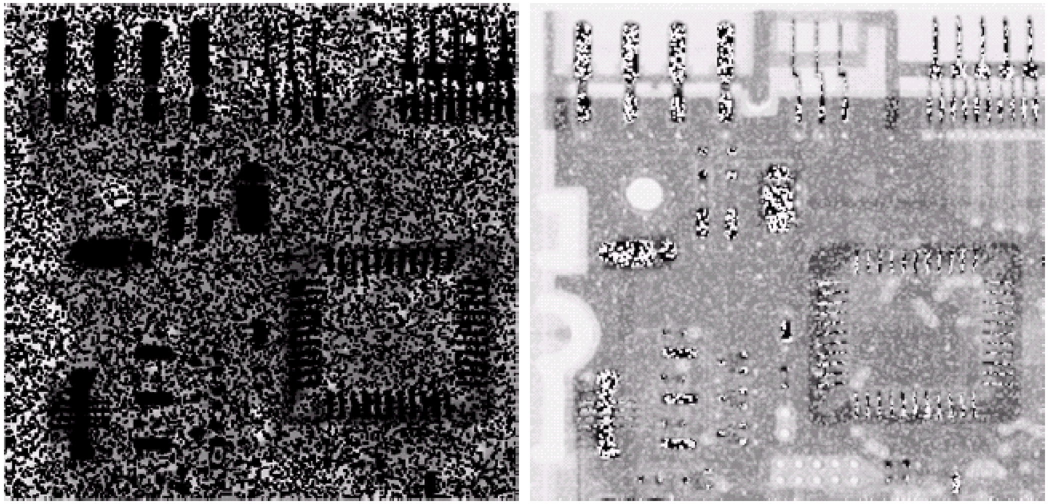


a	b
c	d

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 contra-harmonic filter $Q = 1.5$. (d) Result of filtering (b) with $Q = -1.5$.

Contraharmonic Filter: Here Be Dragons

- Choosing the wrong value for Q when using the contraharmonic filter can have drastic results



Pepper noise filtered by a 3x3 CF with Q=-1.5

Salt noise filtered by a 3x3 CF with Q=1.5

Order Statistics Filters

- Spatial filters based on ordering the pixel values that make up the neighbourhood defined by the filter support.

- Useful spatial filters include
  - Median filter
  - Max and min filter
  - Midpoint filter
  - Alpha trimmed mean filter

Median Filter

Median Filter:

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$$

- Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters.
- Particularly good when salt and pepper noise is present.

Max and Min Filter

Max Filter:

$$\hat{f}(x, y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Min Filter:

$$\hat{f}(x, y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

- Max filter is good for pepper noise and Min filter is good for salt noise.

Midpoint Filter

Midpoint Filter:

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\}+\min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

- Good for random Gaussian and uniform noise.

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

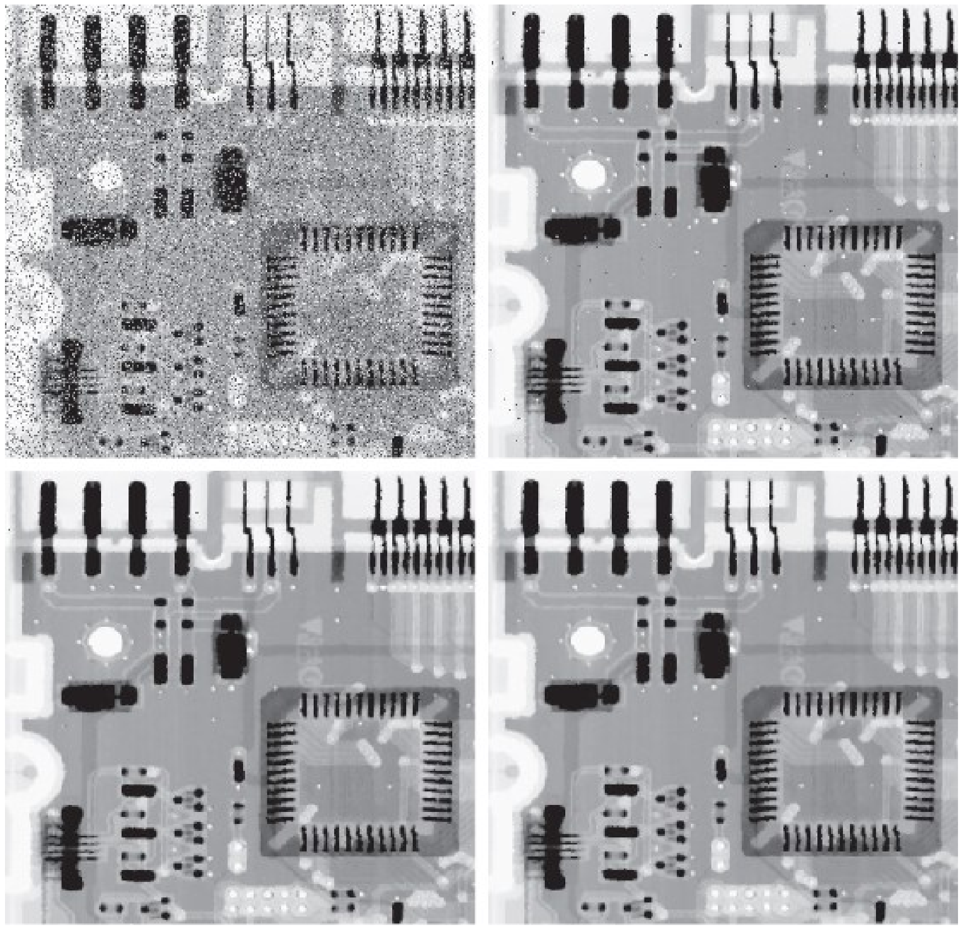
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

- We can delete the $d/2$ lowest and $d/2$ highest grey levels.
- So $g_r(s, t)$ represents the remaining $mn - d$ pixels.

Noise Removal Examples

Salt And Pepper at 0.1

1 pass with a 3x3 median



2 passes with a 3x3 median

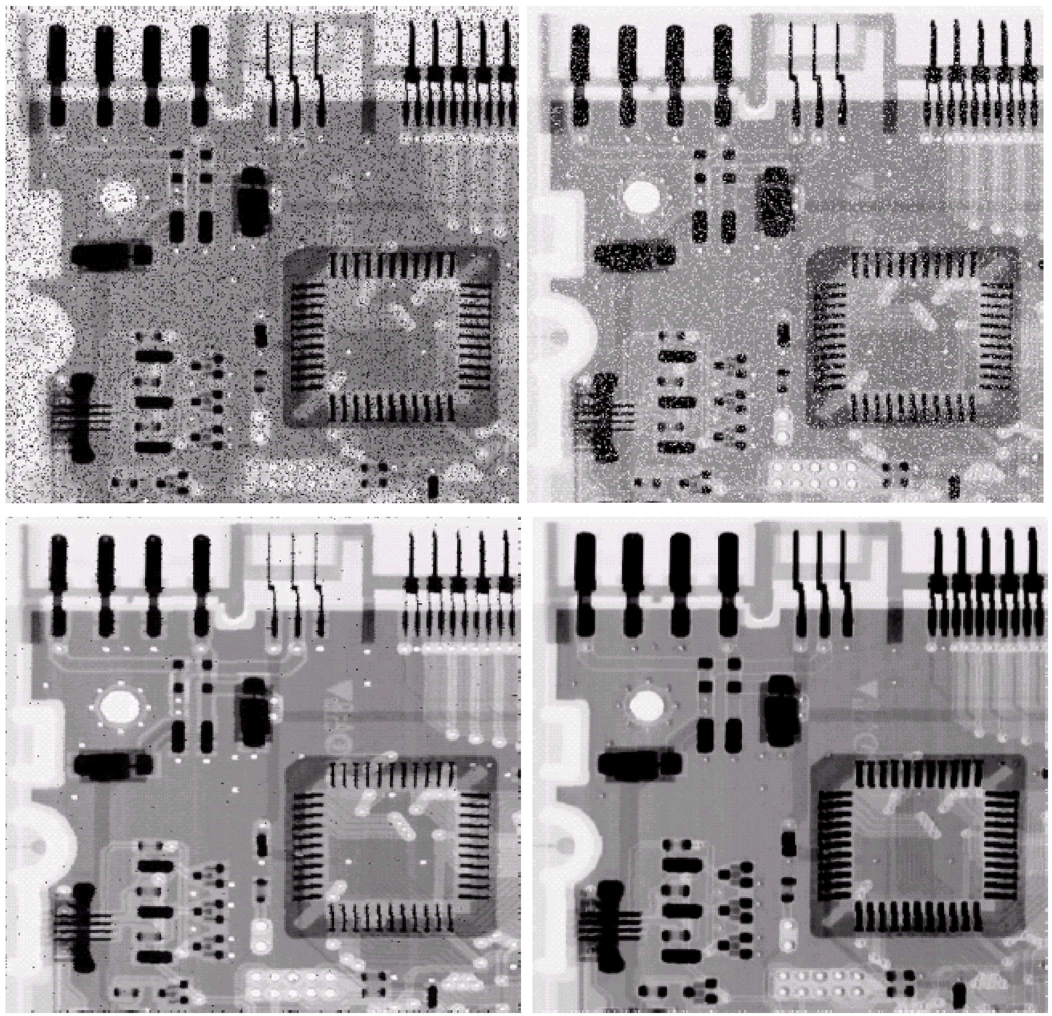
3 passes with a 3x3 median

Repeated passes remove the noise better but also blur the image

Noise Removal Examples (cont…)

Image corrupted by Pepper noise

Image corrupted by Salt noise



Filtering above with a 3x3 Max Filter

Filtering above with a 3x3 Min Filter

Noise Removal Examples (cont…)

Image corrupted by uniform noise

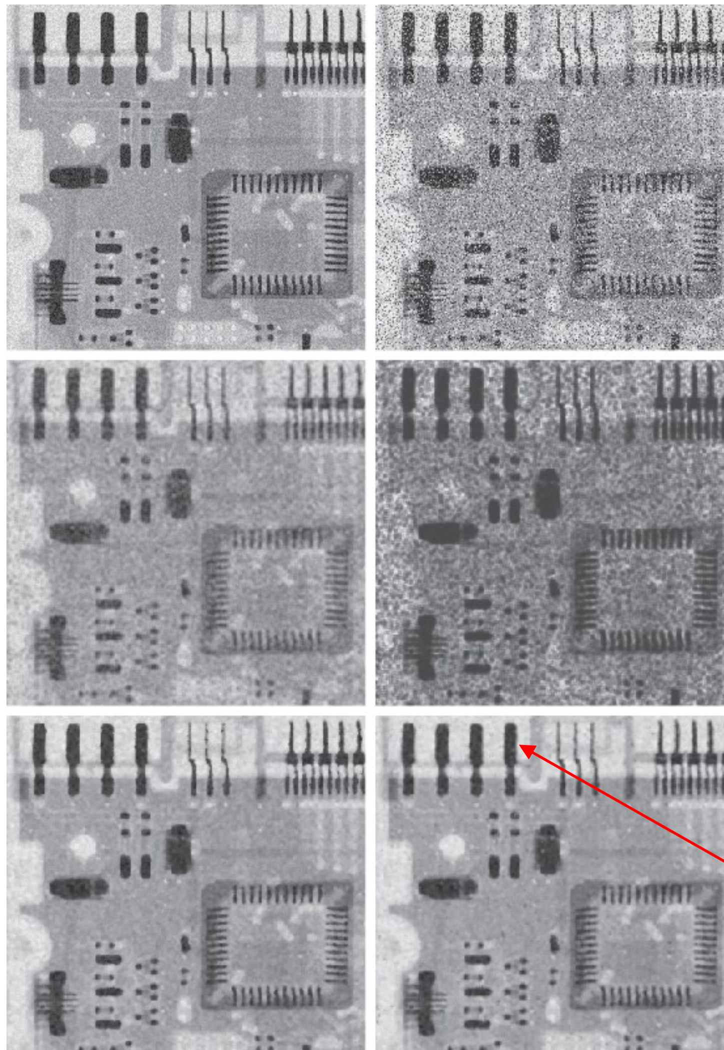
Image further corrupted by Salt and Pepper noise

Filtering by a 5x5 Arithmetic Mean Filter

Filtering by a 5x5 Geometric Mean Filter

Filtering by a 5x5 Median Filter

Filtering by a 5x5 Alpha-Trimmed Mean Filter (d=5)



More smoothing than the simple median

Adaptive Filters

- The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another.
- The behaviour of **adaptive filters** changes depending on the characteristics of the image inside the filter region.
- We will take a look at the
  - **adaptive statistical filter**
  - **adaptive median filter.**

Adaptive Statistical Filter

- Applied in a neighbourhood S_{xy} around (x, y).
- $g(x, y)$ is the degraded image pixel.
- σ_η is the noise standard deviation
- $\bar{z}_{S_{xy}}$ is the mean value in S_{xy}
- σ_{xy} is the standard deviation in S_{xy}

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_{xy}^2}\left[g(x, y) - \bar{z}_{S_{xy}}\right]$$

Adaptive Statistical Filter (cont.)

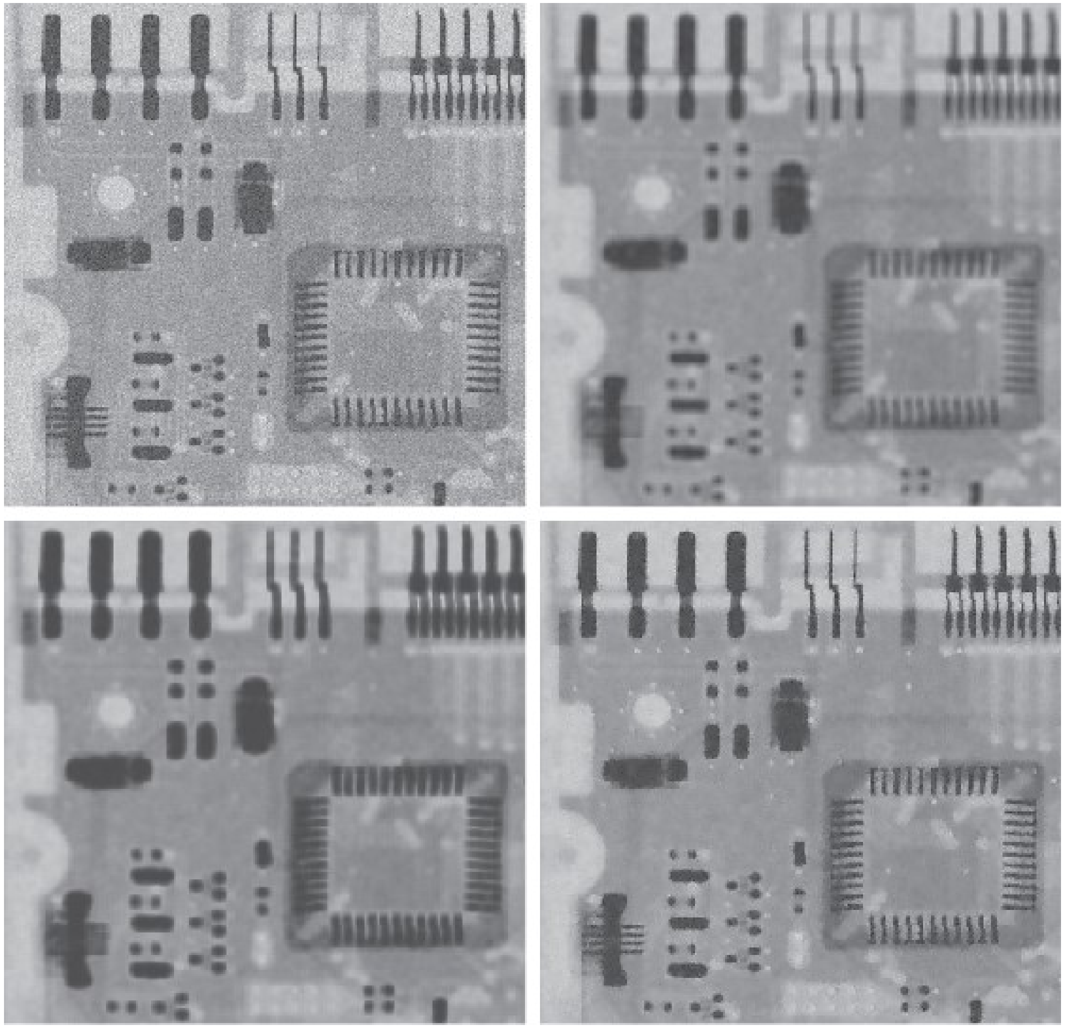
$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{xy}^2}\left[g(x,y) - \bar{z}_{S_{xy}}\right]$$

- When the local standard deviation is higher than the noise standard deviation the filter returns a value close to $g(x,y)$ (e.g. edges tht should be kept).

- When the local standard deviation is close to the noise standard deviation the filter returns a value close to the local average.

- We must estimate σ_η.

Adaptive Statistical Filter (cont.)



a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and a variance of 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise-reduction filtering. All filters used were of size 7×7.

Similar results but better contrast

Adaptive Median Filtering

- The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large.

- The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise.

Adaptive Median Filtering (cont…)

- The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:
  - Remove impulse noise
  - Provide smoothing of other noise
  - Reduce distortion (excessive thinning or thickening of object boundaries).

Adaptive Median Filtering (cont…)

- In the adaptive median filter, the filter size changes depending on the characteristics of the image.
- Notation:
  - S_{xy} = the support of the filter centered at (x, y)
  - z_{min} = minimum grey level in S_{xy}
  - z_{max} = maximum grey level in S_{xy}
  - z_{med} = median of grey levels in S_{xy}
  - z_{xy} = grey level at coordinates (x, y)
  - S_{max} =maximum allowed size of S_{xy}

Adaptive Median Filtering (cont…)

Stage A:

$A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

If $A1 > 0$ and $A2 < 0$, Go to stage B

Else increase the window size

If window size $\leq S_{max}$ repeat stage A

Else output z_{med}

Stage 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 (cont…)

Stage A:

$$A1 = z_{med} - z_{min}$$
$$A2 = z_{med} - z_{max}$$
$$\text{If } A1 > 0 \text{ and } A2 < 0, \text{ Go to stage B}$$
$$\text{Else increase the window size}$$
$$\text{If window size} \leq S_{max} \text{ repeat stage A}$$
$$\text{Else output } z_{med}$$

- Stage A determines if the output of the median filter z_{med} is an impulse or not (black or white).
- If it is not an impulse, we go to stage B.
- If it is an impulse the window size is increased until it reaches S_{max} or z_{med} is not an impulse.
- Note that there is no guarantee that z_{med} will not be an impulse. The smaller the the density of the noise is, and, the larger the support S_{max}, we expect not to have an impulse.

Adaptive Median Filtering (cont…)

Stage B:

$$B1 = z_{xy} - z_{min}$$

$$B2 = z_{xy} - z_{max}$$

If $B1 > 0$ and $B2 < 0$, output z_{xy}

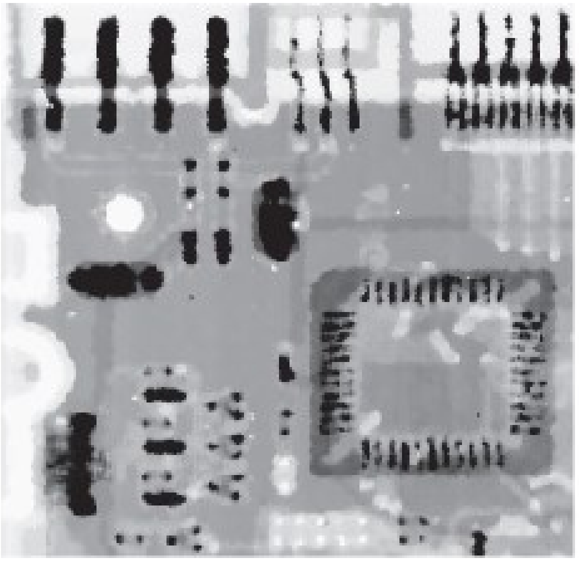
Else output z_{med}

Stage B determines if the pixel value at (x, y), that is z_{xy}, is an impulse or not (black or white).

If it is not an impulse, the algorithm outputs the unchanged pixel value z_{xy} .

If it is an impulse the algorithm outputs the median z_{med} .

Adaptive Median Filtering Example



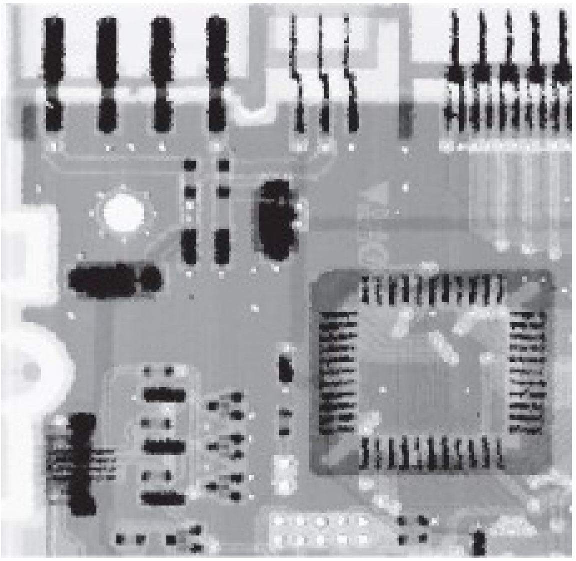


Image corrupted by salt and pepper noise with probabilities $P_a = P_b$=0.25

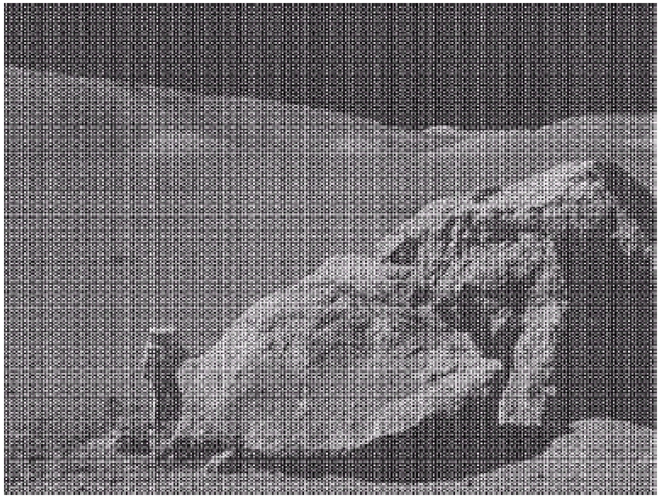
Result of filtering with a 7x7 median filter

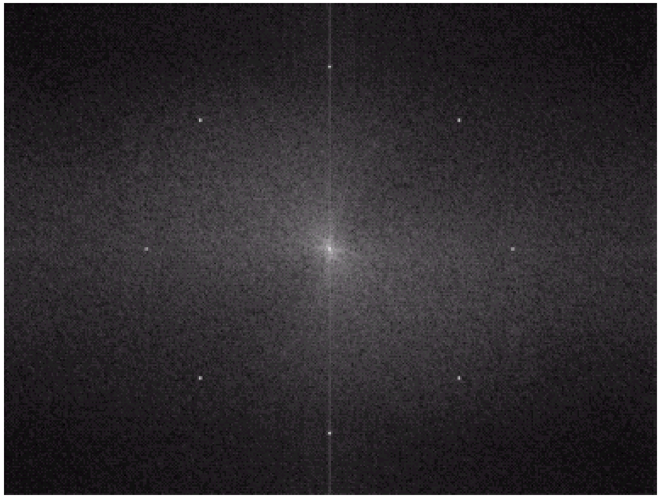
Result of adaptive median filtering with S_{max} = 7

AMF preserves sharpness and details, e.g. the connector fingers.

Periodic Noise

- Typically arises due to electrical or electromagnetic interference.
- Gives rise to regular noise patterns in an image.
- Frequency domain techniques in the Fourier domain are most effective at removing periodic noise.





Notch Filters

- Removing periodic noise form an image involves removing a particular range of frequencies from that image.
- Rejects frequencies in a predefined neighbourhood around a center frequency.

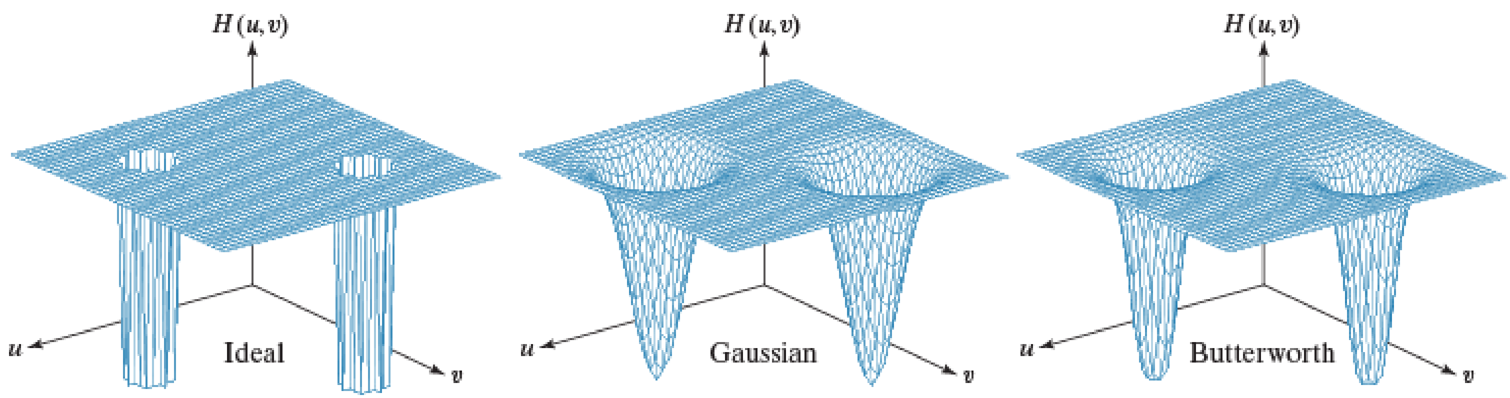


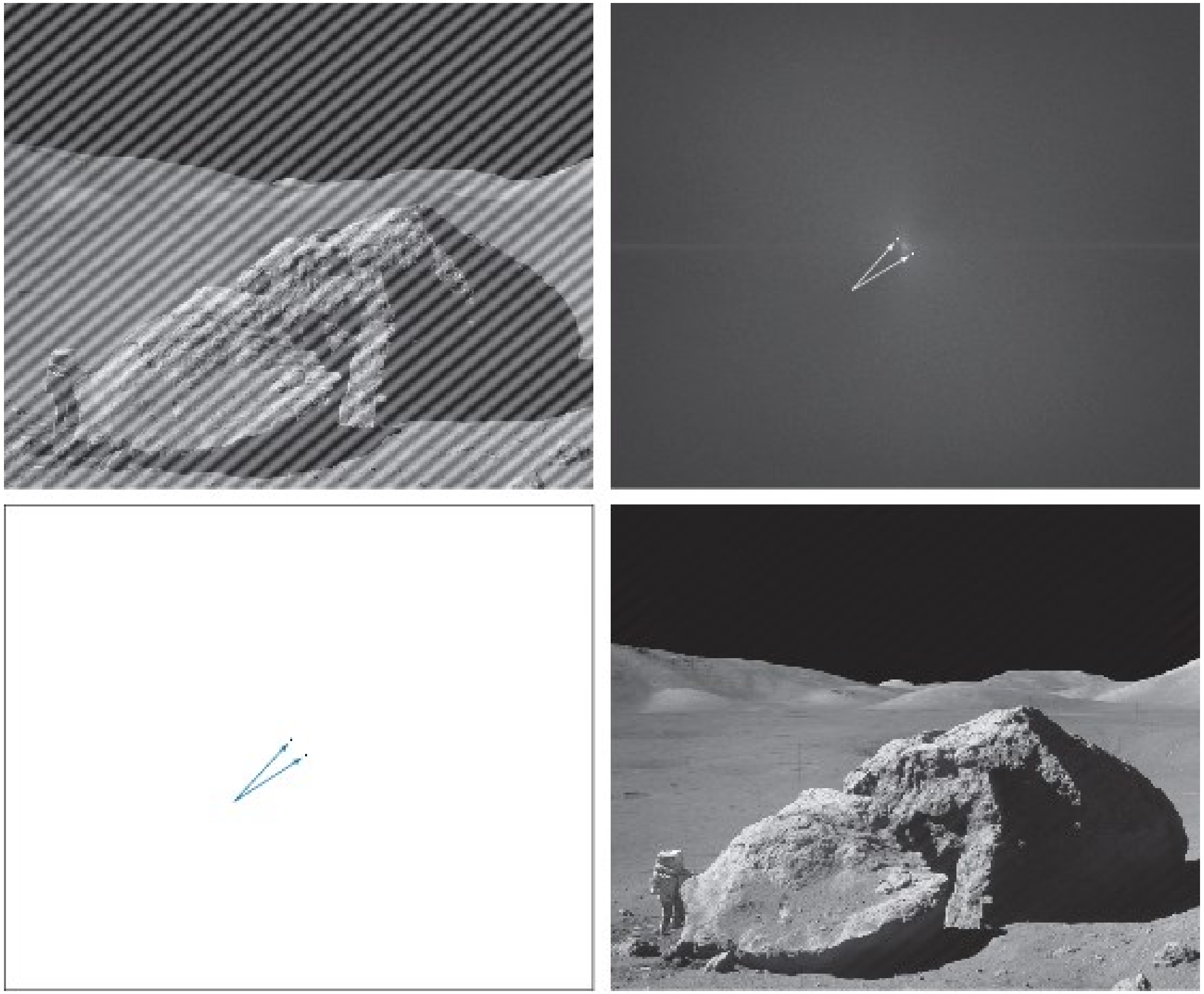
FIGURE 5.15 Perspective plots of (a) ideal, (b) Gaussian, and (c) Butterworth notch reject filter transfer functions.

Bandreject/stop filters (ζωνοφρακτικά φίλτρα)

- Removal of periodic noise involves eliminating a specific range of frequencies from the image.
- This is typically achieved using bandstop filters. Conversely, bandpass filters are used to isolate specific frequency ranges.
- An ideal band reject filter is as follows:

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)<D_0-\dfrac{W}{2}\\ 0 & \text{if } D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\ 1 & \text{if } D(u,v)>D_0+\dfrac{W}{2}\end{cases}$$

Notch Filter Example

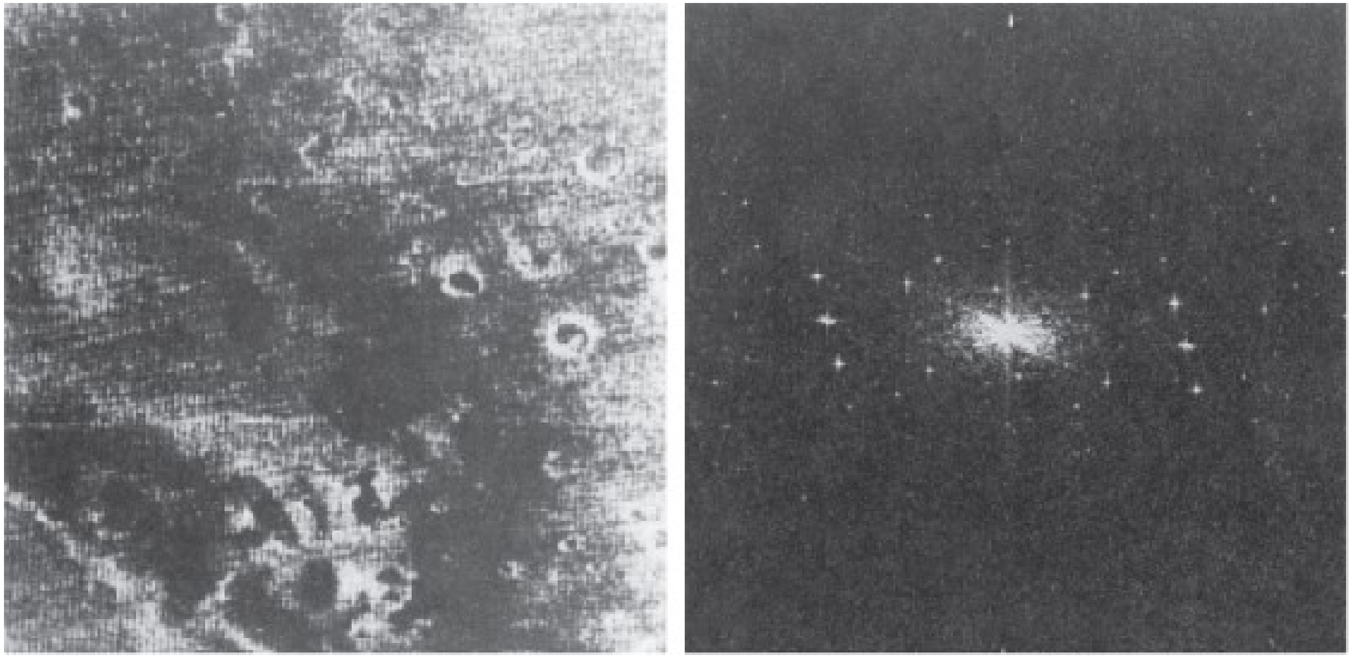


a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal interference. (b) Spectrum showing the bursts of energy caused by the interference. (The bursts were enlarged for display purposes.) (c) Notch filter (the radius of the circles is 2 pixels) used to eliminate the energy bursts. (The thin borders are not part of the data.) (d) Result of notch reject filtering. (Original image courtesy of NASA.)

Optimum Notch Filtering (Αποκατάσταση μέσω ζυγισμένου φίλτρου εγκοπής)

- Several interference components (not a single burst).
- Removing completely the star-like components may also remove image information.

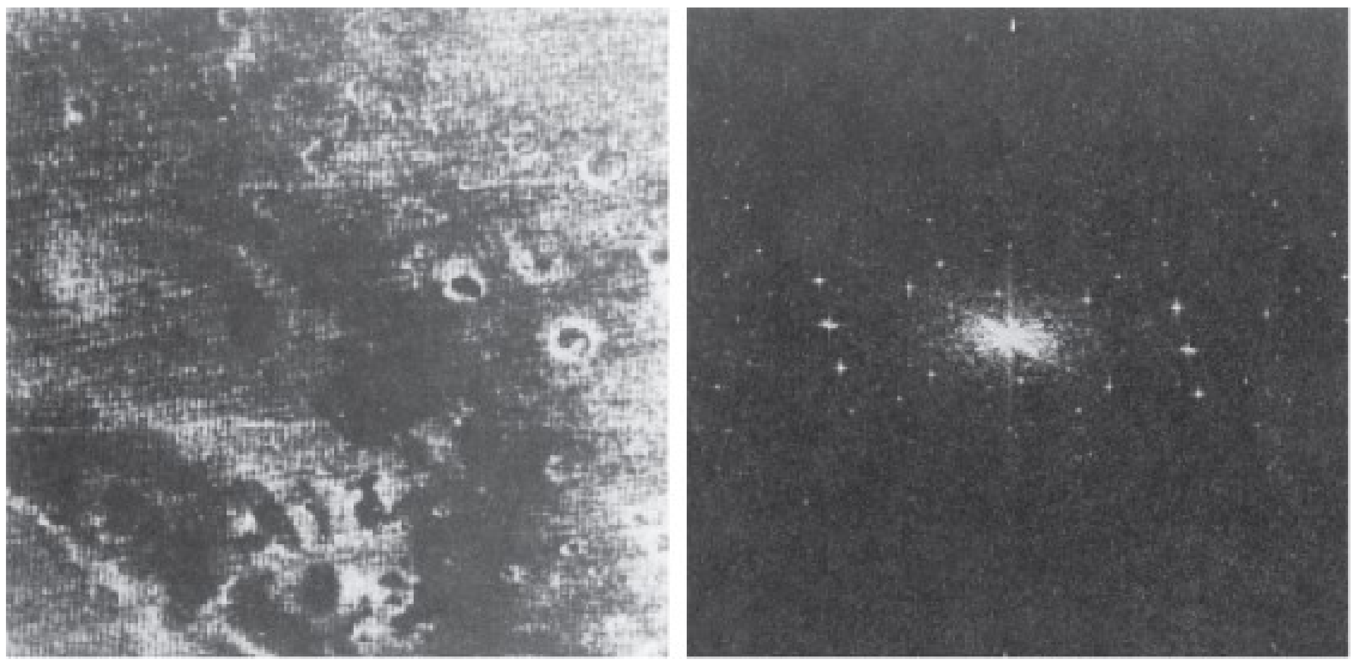


a b

FIGURE 5.20 (a) Image of the Martian terrain taken by Mariner 6. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

- Apply the notch filter to isolate the bursts.
- Remove a portion of the burst.



a b

FIGURE 5.20 (a) Image of the Martian terrain taken by Mariner 6. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

- A noise estimate in the DFT domain:

$$N(k,l) = H(k,l)G(k,l)$$

- In the spatial domain:

$$\eta(m,n) = \Im^{-1}\left\{H(k,l)G(k,l)\right\}$$

- Image estimate:

$$\hat{f}(m,n) = g(m,n) - w(m,n)\eta(m,n)$$

Optimum Notch Filtering (cont.)

$$\hat{f}(m,n) = g(m,n) - w(m,n)\eta(m,n)$$

- How we select a proper w;
- Compute the weight minimizing the variance over a local neighbourhood of the estimated image centered at (m,n):

$$\sigma^2(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \left[\hat{f}(m+k,n+l) - \bar{\hat{f}}(m,n) \right]^2$$

with
$$\bar{\hat{f}}(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \hat{f}(m+k,n+l)$$

- Substituting $\hat{f}(m,n)$ in $\sigma^2(m,n)$: yields:

Optimum Notch Filtering (cont.)

$$\sigma^2(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \Big\{ \big[g(m+k,n+l) - w(m+k,n+l)\eta(m+k,n+l) \big] - \big[\bar{g}(m,n) - \overline{w(m,n)\eta(m,n)} \big] \Big\}^2$$

- A simplification is to assume that the weight remains constant over the neighbourhood:

$$w(m+k,n+l) = w(m,n), \; -a \le k \le a, -b \le l \le b$$

$$\sigma^2(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \Big\{ \big[g(m+k,n+l) - w(m,n)\eta(m+k,n+l) \big] - \big[\bar{g}(m,n) - w(m,n)\bar{\eta}(m,n) \big] \Big\}^2$$

Optimum Notch Filtering (cont.)

- To minimize the variance:

$$\frac{\partial \sigma^2(m,n)}{\partial w(m,n)} = 0$$

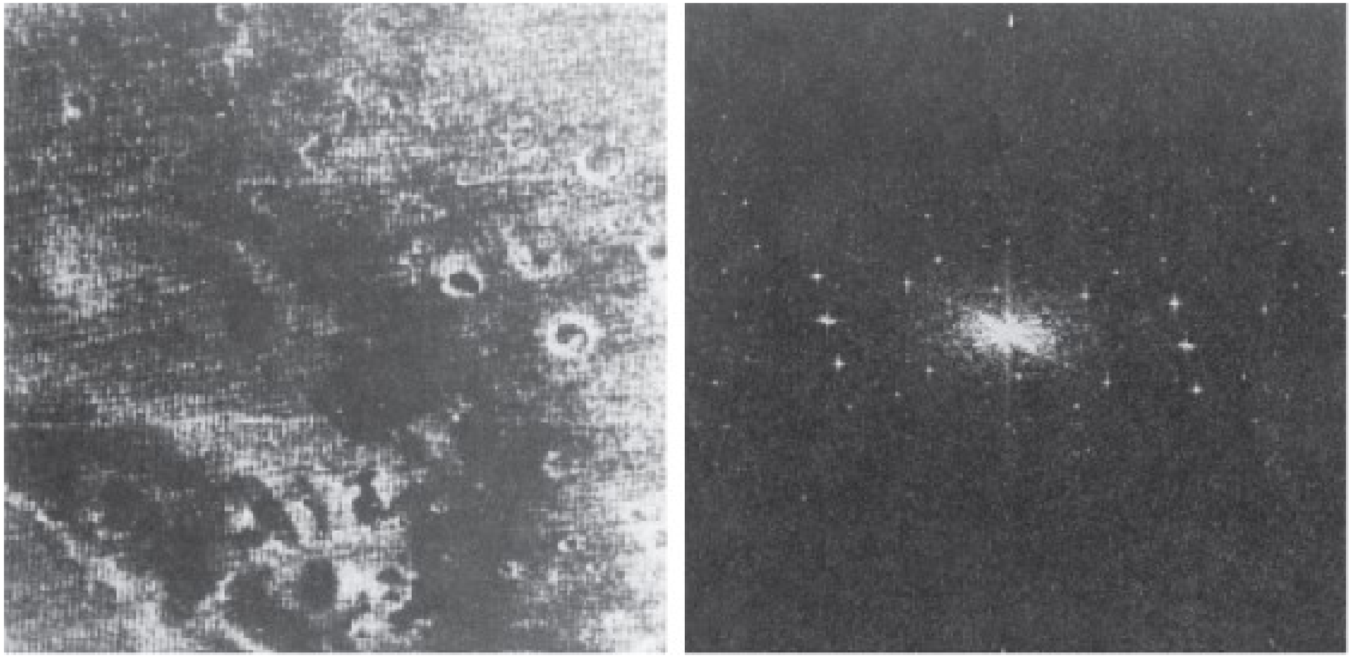
yielding the closed-form solution:

$$w(m,n) = \frac{\overline{g(m,n)\eta(m,n)} - \bar{g}(m,n)\bar{\eta}(m,n)}{\overline{\eta^2}(m,n) - \bar{\eta}^2(m,n)}$$

- More elaborated result is obtained for non-constant weight $w(m,n)$ at each pixel.

Optimum Notch Filtering (cont.)

- Apply the notch filter to isolate the bursts.
- Remove a portion of the burst.

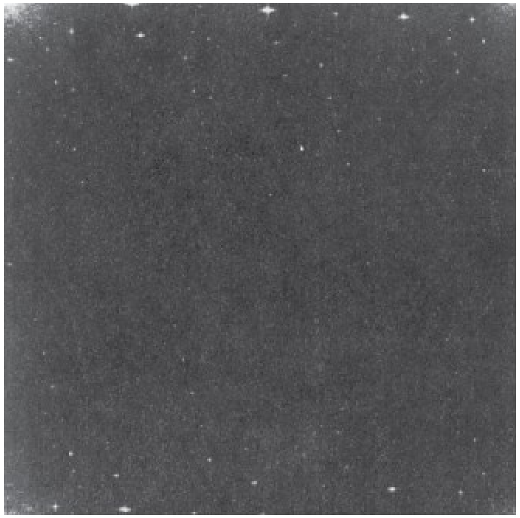


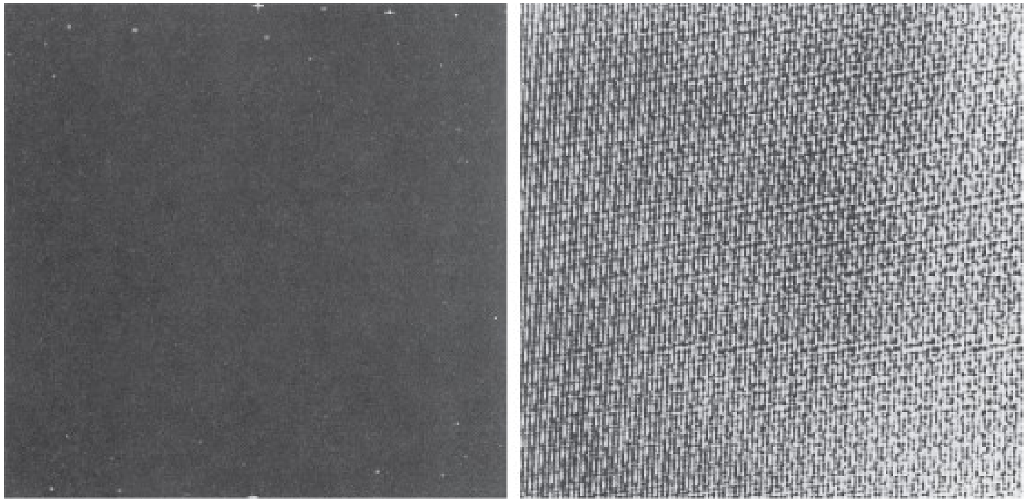
a b

FIGURE 5.20 (a) Image of the Martian terrain taken by Mariner 6. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

FIGURE 5.21 Uncentered Fourier spectrum of the image in Fig. 5.20(a). (Courtesy of NASA.)





a b

FIGURE 5.22 (a) Fourier spectrum of $N(u,v)$, and (b) corresponding spatial noise interference pattern, $\eta(x,y)$. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

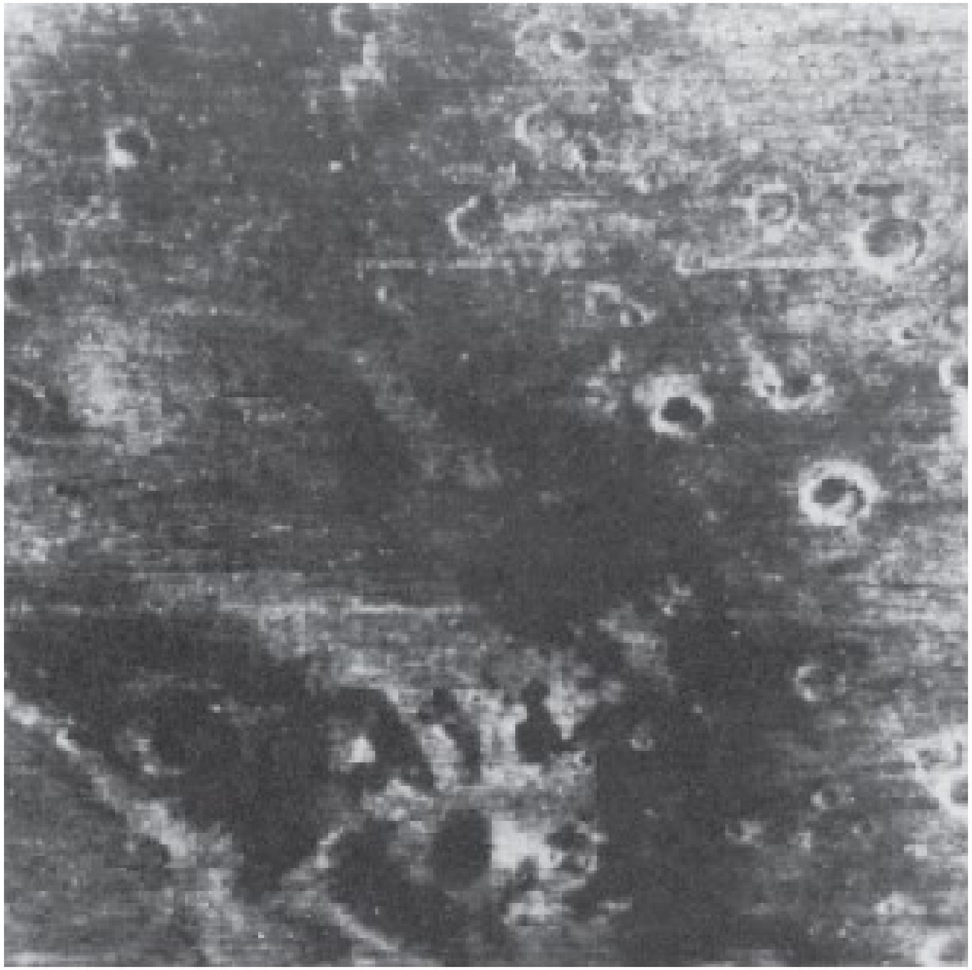


FIGURE 5.23
Restored image. (Courtesy of NASA.)

Synopsis

We examined:

- **Image restoration in the presence of noise**
- **Noise models**
- Noise removal by **filtering** in the **spatial** and **frequency** domain
- In the next lesson we will talk about **linear image restoration with no noise**

- Upcoming Project announcement – in the next few days

Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Image Restoration and Reconstruction (Linear Restoration Methods)

Άγγελος Γιώτης

a.giotis@uoi.gr

Contents

In this lecture we will look at linear image restoration techniques

- Differentiation of matrices and vectors
- Linear space invariant degradation
- Restoration in absence of noise
  - Inverse filter
  - Pseudo-inverse filter
- Restoration in presence of noise
  - Inverse filter
  - Wiener filter
  - Constrained least squares filter

Differentiation of Matrices and Vectors

Notation:

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathrm{f}(\mathbf{x})$ is a scalar function of vector $\mathbf{x}$.

$\mathbf{g}(\mathbf{x})$ is a Mx1 vector valued function of vector $\mathbf{x}$.

Differentiation of Matrices and Vectors (cont...)

Scalar derivative of a matrix.

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$$\frac{\partial \mathbf{A}}{\partial t} = \begin{pmatrix} \frac{\partial a_{11}}{\partial t} & \cdots & \frac{\partial a_{1N}}{\partial t} \\ \vdots & \ddots & \vdots \\ \frac{\partial a_{M1}}{\partial t} & \cdots & \frac{\partial a_{MN}}{\partial t} \end{pmatrix}$$

Differentiation of Matrices and Vectors (cont...)

Vector derivative of a function (gradient).

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathrm{f}(\mathbf{x})$ is a scalar function of vector $\mathbf{x}$.

$$\frac{\partial f}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial x_1} \quad \cdots \quad \frac{\partial f}{\partial x_N} \right)^T$$

Differentiation of Matrices and Vectors (cont...)

Vector derivative of a vector (Jacobian):

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{g}(\mathbf{x})$ is a Mx1 vector valued function of vector $\mathbf{x}$.

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_M}{\partial x_1} & \cdots & \frac{\partial g_M}{\partial x_N} \end{pmatrix}$$

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{b}$ is a Nx1 vector with elements b_i.

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{b}^T\mathbf{x}\right) = \mathbf{b}$$

It is the derivative of the scalar valued function $\mathbf{b}^T\mathbf{x}$ with respect to vector $\mathbf{x}$.

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{A}$ is a NxN matrix with elements a_{ij}.

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^T \mathbf{A} \mathbf{x}\right) = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{x}$$

If A is symmetric:

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^T \mathbf{A} \mathbf{x}\right) = 2\mathbf{A}\mathbf{x}$$

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{b}$ is a Mx1 vector with elements b_i.

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$$\frac{\partial}{\partial \mathbf{x}}\left\|\mathbf{A}\mathbf{x}+\mathbf{b}\right\|^2 = 2\mathbf{A}^T\left(\mathbf{A}\mathbf{x}+\mathbf{b}\right)$$

It may be proved using the previous properties.

Differentiation of Matrices and Vectors (cont...)

$$\frac{\partial}{\partial \mathbf{x}} \left\| \mathbf{Ax} + \mathbf{b} \right\|^2 = 2\mathbf{A}^T \left(\mathbf{Ax} + \mathbf{b} \right)$$

$$\boldsymbol{f}(\boldsymbol{x}) = \ \|\boldsymbol{Ax} + \boldsymbol{b}\|^2 = (\boldsymbol{Ax} + \boldsymbol{b})^T (\boldsymbol{Ax} + \boldsymbol{b})$$

$$= (\boldsymbol{Ax})^T \boldsymbol{Ax} + \boldsymbol{b}^T \boldsymbol{Ax} + (\boldsymbol{Ax})^T \boldsymbol{b} + \boldsymbol{b}^T \boldsymbol{b}$$

Since $(\boldsymbol{Ax})^T = \boldsymbol{x}^T \boldsymbol{A}^T$ and scalar quantities are equal to their transpose:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{Ax} + 2\boldsymbol{b}^T \boldsymbol{Ax} + \boldsymbol{b}^T \boldsymbol{b},$$

where $\boldsymbol{b}^T \boldsymbol{b}$ is constant with respect to $\boldsymbol{x}$, so its derivative will be zero.

Differentiation of Matrices and Vectors (cont...)

- Using the result from matrix calculus for $\boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}$ where $\boldsymbol{C} = \boldsymbol{A}^T \boldsymbol{A}$ is symmetric:

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A^T A \mathbf{x}) = 2A^T A \mathbf{x}$$

- Since $\boldsymbol{b}^T \boldsymbol{A} \boldsymbol{x}$ is a scalar, we use the result for the derivative of a linear form $\boldsymbol{c}^T \boldsymbol{x}$ (where $\boldsymbol{c}^T$ is a linear combination of the rows of $\boldsymbol{A}$, defined by $\boldsymbol{b}^T$, i.e., a row vector):

$$\frac{\partial}{\partial \mathbf{x}}(2\mathbf{b}^T A \mathbf{x}) = 2A^T \mathbf{b}.$$

- Combining the derivatives:

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A^T A \mathbf{x}) + \frac{\partial}{\partial \mathbf{x}}(2\mathbf{b}^T A \mathbf{x})$$

$$\frac{\partial f}{\partial \mathbf{x}} = 2A^T A \mathbf{x} + 2A^T \mathbf{b}$$

Yields:

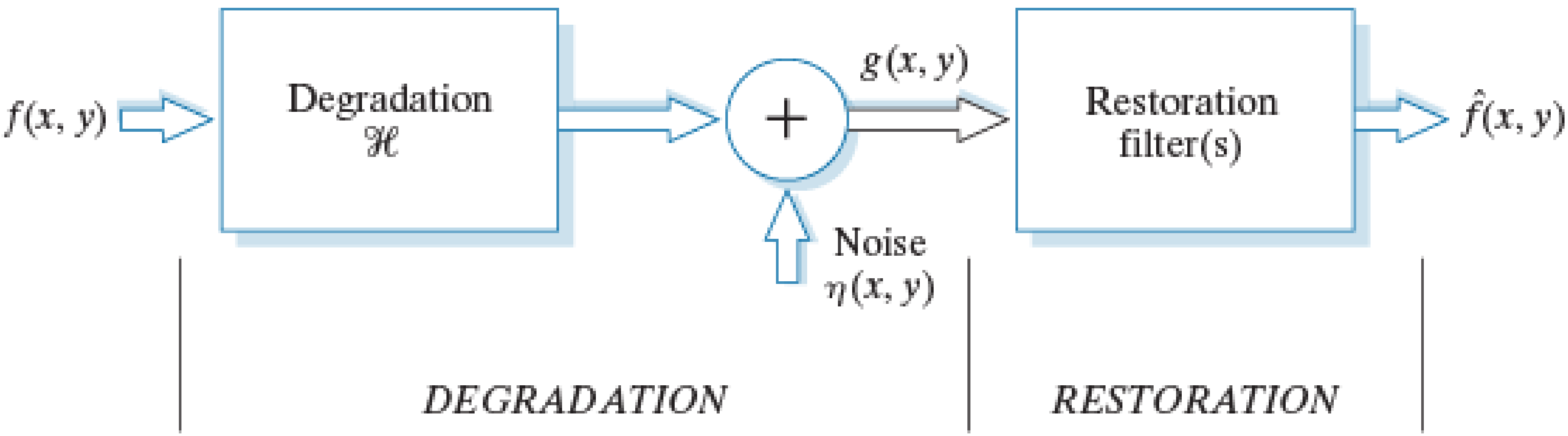
$$\frac{\partial}{\partial \mathbf{x}} \left\| \mathbf{A}\mathbf{x} + \mathbf{b} \right\|^2 = 2\mathbf{A}^T \left(\mathbf{A}\mathbf{x} + \mathbf{b} \right)$$

So far: Standard Additive noise model

- Until now: we considered a noisy image to be modelled as follows:

$$g(x,y) = f(x,y) + \eta(x,y)$$

under the "**white Gaussian noise**" assumption.



Linear, Position-Invariant Degradation

We now consider a degraded image to be modelled by:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

where $h(x, y)$ is the impulse response of the degradation function (i.e. *point spread function* blurring the image).

Linear, Position-Invariant Degradation

$$g(x,y) = h(x,y) * f(x,y) \ + \eta(x,y)$$

The convolution implies that the degradation mechanism is linear and position invariant (it depends only on image values and not on location).

An operator $\mathcal{H}$ having the input-output relationship $g(x,y) = \mathcal{H}[f(x,y)]$

is said to be *position* (or *space*) *invariant* if

$$\mathcal{H}[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta)$$

Linear, Position-Invariant Degradation

Example degraded images-observations $g(x, y)$



Linear, Position-Invariant Degradation (cont...)

In the Fourier domain:

$$G(k,l) = H(k,l)F(k,l) + N(k,l)$$

where multiplication is element-wise.

In matrix-vector form:

$$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$$

where $\mathbf{H}$ is a doubly block circulant matrix and $\mathbf{f}$, $\mathbf{g}$, and $\boldsymbol{\eta}$ are vectors (lexicographic ordering).

Linear, Position-Invariant Degradation (cont...)

$$g(x, y) = h(x, y) * f(x, y) \ + \ \eta(x, y)$$

$$G(k,l) = H(k,l)F(k,l) \ + N(k,l)$$

$$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$$

If the degradation function is unknown the problem of simultaneously recovering $f(x,y)$ and $h(x,y)$ is called *blind deconvolution*.

Estimating the point spread function

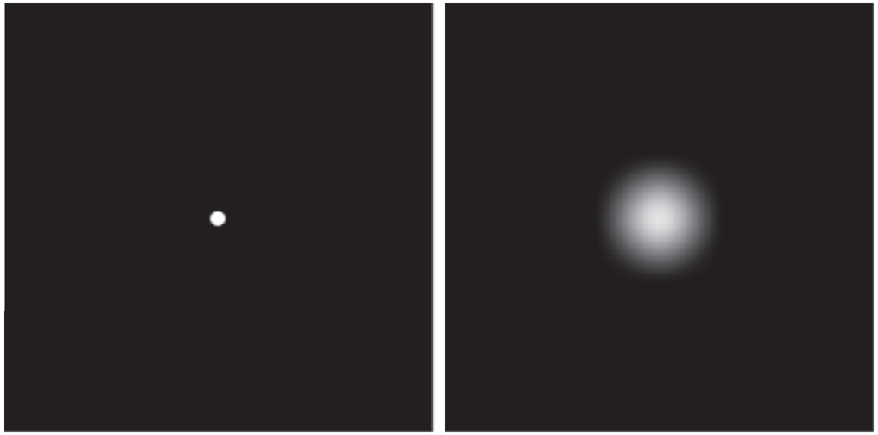
- In what follows, we consider that the degradation function is known.
- If the psf is not known, some **basic** methods to estimate it are:
  - By observation
    - Apply sharpening filters to a sub-image $g_s(m,n)$ where the signal is strong (there is almost no noise) and obtain a visually pleasant result $f_s(m,n)$. The psf may be approximated by $H_s(k,l)= G_s(k,l)/ F_s(k,l)$.
    - The task needs trial and error and may be tedious.
    - Used in special circumstances (e.g. Restoration of old photographs)

Estimating the point spread function (cont.)

- If the psf is not known:
  - By experimentation
    - If the acquisition equipment or a similar one is available an image similar to the degraded may be obtained by varying the system settings.
    - Then obtain the image of an impulse (small dot of light) using the same settings.
    - We estimate $H(k,l) = G(k,l)/A$ (for constant *A*)

a b

FIGURE 5.24 Estimating a degradation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.



Estimating the point spread function (cont.)

- If the psf is not known, some **basic** methods to estimate it are:
  - By **modeling**: **atmospheric turbulence:**

$$H(u,v) = \exp\left(-k\left(u^2+v^2\right)^{5/6}\right)$$

a b
c d

FIGURE 5.25 Modeling turbulence. (a) No visible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. All images are of size 480×480 pixels. (Original image courtesy of NASA.)

Estimating the point spread function (cont.)

- By modeling: **planar motion**
  - $x_0(t)$ and $y_0(t)$ are the time varying components of motion at each pixel.
  - The total exposure at any pixel is obtained by integrating the instantaneous exposure over the time the shutter is open.
  - Assumption: the shutter opening and closing is instantaneous.
  - If T is the duration of the exposure, the recorded image is expressed by:

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]dt$$

Estimating the point spread function (cont.)

$$
\begin{aligned}
G(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy \\
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{0}^{T}\left[\, f[x-x_0(t), y-y_0(t)]dt\,\right]e^{-j2\pi(ux+vy)}dxdy \\
&= \int_{0}^{T}\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f[x-x_0(t), y-y_0(t)]e^{-j2\pi(ux+vy)}dxdy\right]dt \\
&= \int_{0}^{T} F(u,v)e^{-j2\pi[ux_0(t)+vy_0(t)]}dt \\
&= F(u,v)\int_{0}^{T} e^{-j2\pi[ux_0(t)+vy_0(t)]}dt \Leftrightarrow \\
H(u,v) &= \int_{0}^{T} e^{-j2\pi[ux_0(t)+vy_0(t)]}dt
\end{aligned}
$$

Estimating the point spread function (cont.)

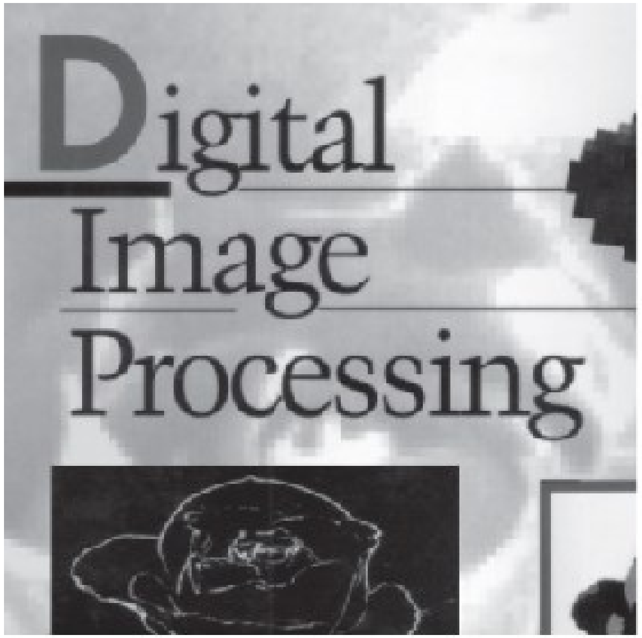
- Considering uniform linear motion:

$$x_0(t) = a\frac{t}{T},\ y_0(t) = b\frac{t}{T}$$

- The psf becomes:

$$\begin{aligned} H(u,v) &= \int_0^T e^{-j2\pi(uat+vbt)/T}\,dt \\ &= \frac{T}{\pi(ua+vb)}\sin\left[\pi(ua+vb)\right]e^{-j\pi(ua+vb)} \end{aligned}$$

Estimating the point spread function (cont.)



- Result of blurring with:

$$x_0(t) = a\frac{t}{T},\ y_0(t) = b\frac{t}{T},\ \ a = b = 0.1,\ T = 1$$

Linear Restoration

Using the imaging system

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

we want to estimate the true image from the degraded observation with **known** degradation $\mathbf{H}$.

A linear method applies an operator (a matrix) $\mathbf{P}$ to the observation $\mathbf{g}$ to estimate the unobserved noise-free image $\mathbf{f}$:

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g}$$

Restoration in Absence of Noise
The Inverse Filter

When there is no noise:

$$\mathbf{g} = \mathbf{H}\mathbf{f}$$

an obvious solution would be to use the **inverse filter**:

$$\mathbf{P} = \mathbf{H}^{-1}$$

yielding

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g} = \mathbf{H}^{-1}\mathbf{g} = \mathbf{H}^{-1}\mathbf{H}\mathbf{f} = \mathbf{f}$$

Restoration in Absence of Noise The Inverse Filter (cont...)

$$\hat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g}$$

For a NxN image, $\mathbf{H}$ is a N^2xN^2 matrix!

To tackle the problem we transform it to the Fourier domain.

$\mathbf{H}$ is doubly block circulant and therefore it may be diagonalized by the 2D DFT matrix $\mathbf{W}$:

$$\mathbf{H} = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$$

Restoration in Absence of Noise
The Inverse Filter (cont...)

$$\mathbf{H} = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$$

where

$$\mathbf{\Lambda} = diag\{H(1,1),\ldots,H(N,1),H(1,2),\ldots H(N,N)\}$$

Therefore:

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g} \Leftrightarrow \hat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g} \Leftrightarrow \hat{\mathbf{f}} = (\mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W})^{-1}\mathbf{g}$$

$$\Leftrightarrow \hat{\mathbf{f}} = \mathbf{W}^{-1}\mathbf{\Lambda}^{-1}\mathbf{W}\mathbf{g} \Leftrightarrow \mathbf{W}\hat{\mathbf{f}} = \mathbf{W}\mathbf{W}^{-1}\mathbf{\Lambda}^{-1}\mathbf{W}\mathbf{g}$$

$$\Leftrightarrow \hat{\mathbf{F}} = \mathbf{\Lambda}^{-1}\mathbf{G}$$

Restoration in Absence of Noise
The Inverse Filter (cont...)

This is the vectorized form of the DFT of the image:

$$\hat{\mathbf{F}} = \mathbf{\Lambda}^{-1}\mathbf{G} \quad \Leftrightarrow \hat{F}(k,l) = \frac{G(k,l)}{H(k,l)}$$

Take the inverse DFT and obtain $f(m,n)$.

Problem: what happens if $H(k,l)$ has zero values?

Cannot perform inverse filtering!

Restoration in Absence of Noise
The Pseudo-inverse Filter

A solution is to set:

$$\hat{F}(k,l) = \begin{cases} \dfrac{G(k,l)}{H(k,l)} & , \quad H(k,l) \neq 0 \\ 0 & , \quad H(k,l) = 0 \end{cases}$$

which is a type of pseudo-inversion.

Notice that the signal cannot be restored at locations where $H(k,l)$=0.

Restoration in Absence of Noise The Pseudo-inverse Filter (cont...)

A pseudo-inverse filter also arises by the unconstrained least squares approach.

Find the image $\mathbf{f}$, that, when it is blurred by $\mathbf{H}$, it will provide an observation as close as possible to $\mathbf{g}$, i.e. It minimizes the distance between $\mathbf{Hf}$ and $\mathbf{g}$.

Restoration in Absence of Noise
The Pseudo-inverse Filter (cont...)

This distance is expressed by the norm:

$$J(\mathbf{f}) = \left\| \mathbf{Hf} - \mathbf{g} \right\|^2$$

$$\min_{\mathbf{f}} \{J(\mathbf{f})\} \Leftrightarrow \frac{\partial J}{\partial \mathbf{f}} = 0 \Leftrightarrow \frac{\partial}{\partial \mathbf{f}}\left(\left\| \mathbf{Hf} - \mathbf{g} \right\|^2 \right) = 0$$

$$\Leftrightarrow 2\mathbf{H}^T\left(\mathbf{Hf} - \mathbf{g}\right) = 0 \Leftrightarrow 2\mathbf{H}^T\mathbf{Hf} = 2\mathbf{H}^T\mathbf{g}$$

$$\Leftrightarrow \mathbf{f} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{g}$$