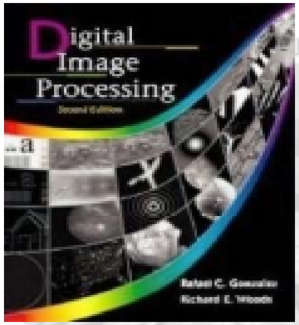


Digital Image Processing
Digital Image Processing Using Matlab

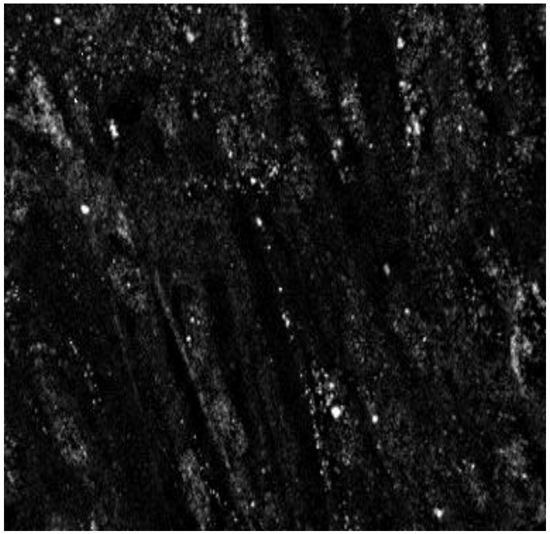
Digital Image Processing

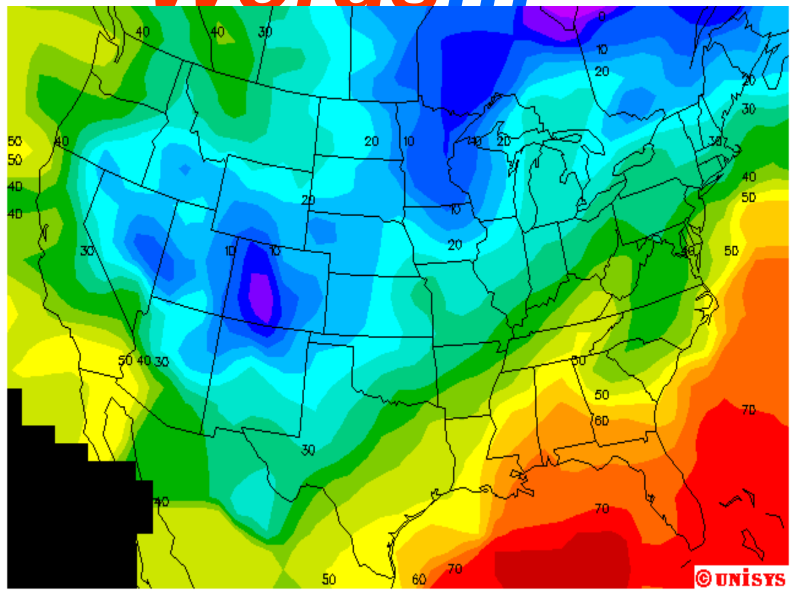
Prepared by:
Dr. Ali J. Abboud
University of Diyala
2012-2013

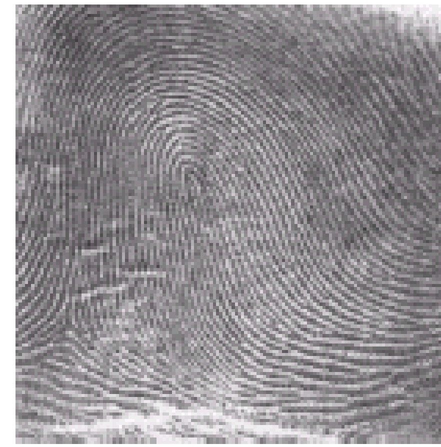
Chapter 1

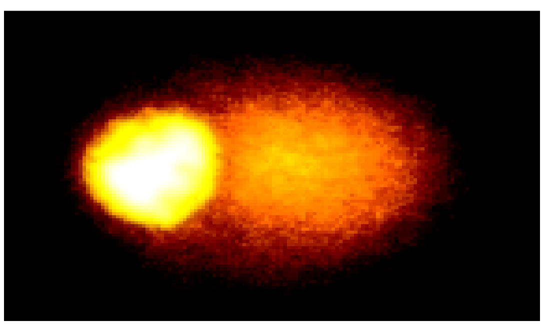
- **Define the field and its scope.**
- **Give a brief introduction about image processing history field.**
- **Explain basic terminology and notations.**
- **What will take in next chapters.**

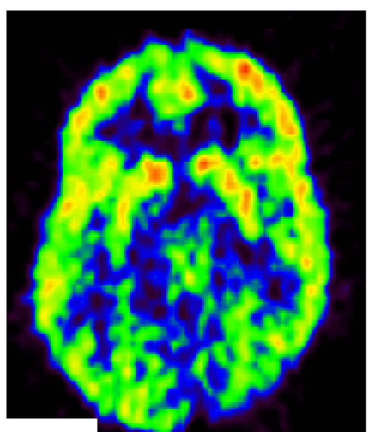
An Image is Worth more than a Thousand Words!!!

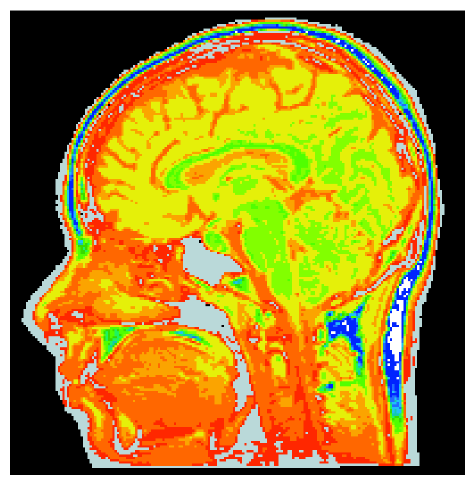


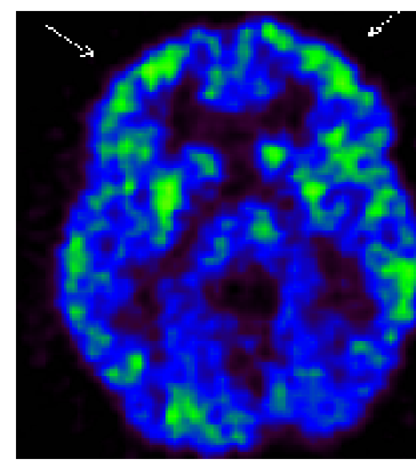












What is digital Image?

- A **digital image** differs from a photo in that the values are all **discrete**.
- Usually they take on only **integer** values.
- A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements, or more simply **pixels**.
- The pixels surrounding a given pixel constitute its **neighborhood** A neighborhood can be characterized by its shape in the same way as a matrix: we can speak of a 3x3 neighborhood, or of a 5x7 neighborhood.

What is digital Image?

48	219	168	145	244	188	120	58
49	218	87	94	133	35	17	148
174	151	74	179	224	3	252	194
77	127	87	139	44	228	149	135
138	229	136	113	250	51	108	163
38	210	185	177	69	76	131	53
178	164	79	158	64	169	85	97
96	209	214	203	223	73	110	200

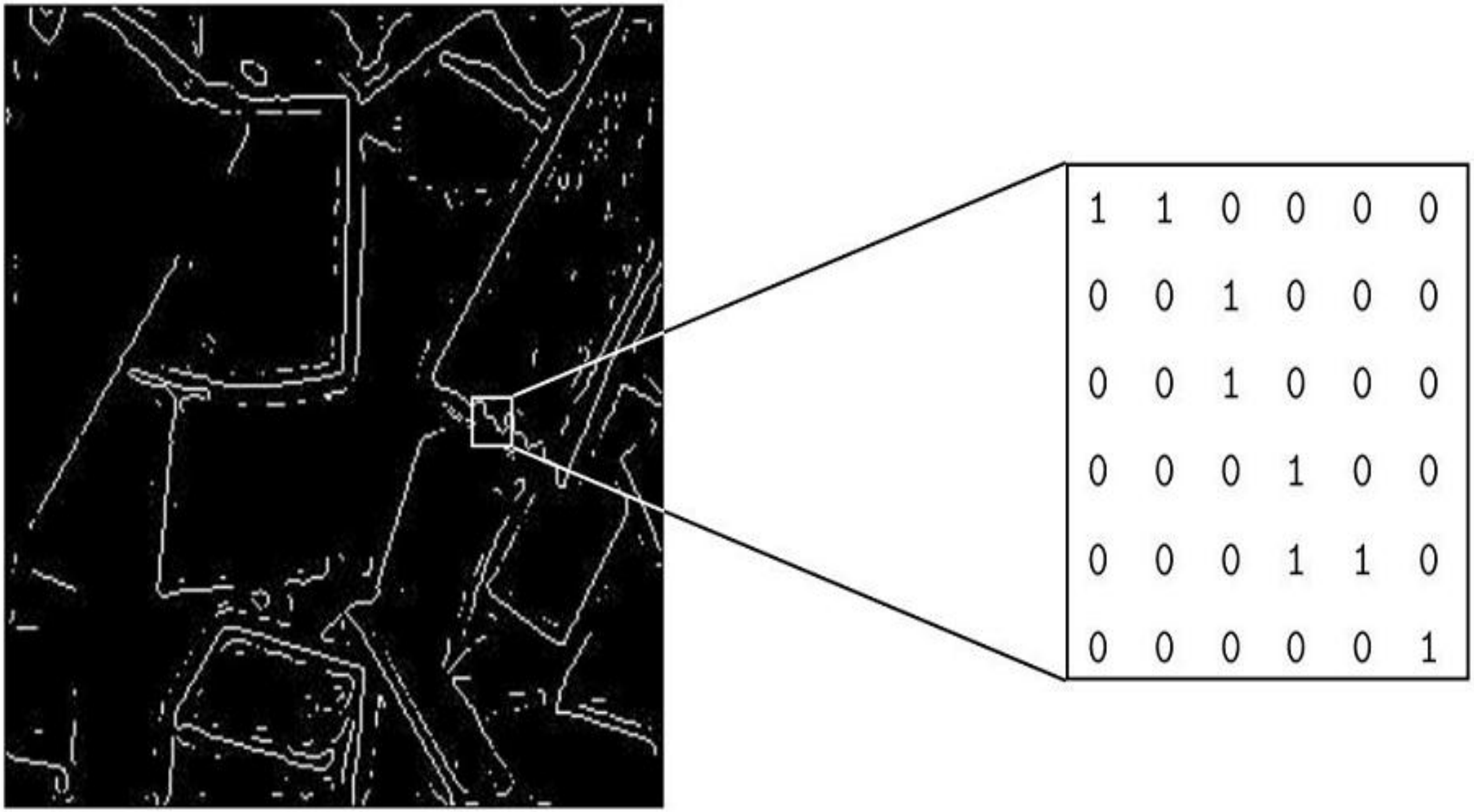
Current pixel

3×5 neighbourhood

Types of Digital Images

- **Binary:** Each pixel is just **black** or **white**. Since there are only two possible values for each pixel (0,1), we only need **one bit** per pixel.
- **Grayscale:** Each pixel is a shade of gray, normally from **0** (black) to **255** (white). This range means that each pixel can be represented by **eight bits**, or exactly **one byte**. Other greyscale ranges are used, but generally they are a power of **2**.
- **True Color**, or **RGB**: Each pixel has a particular color; that color is described by the amount of **red**, **green** and **blue** in it. If each of these components has a range 0–255, this gives a total of $\mathbf{256^3}$ different possible colors. Such an image is a “stack” of **three matrices**; representing the **red**, **green** and **blue** values for each pixel. This means that for every pixel there correspond 3 values.

Binary Images



Gray scale Images

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Color Images

49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

Red

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

Green

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

Blue

What is Digital Image Processing?

Image is nothing but a function in two dimensions plotted as a 2D display with expressing the value of the function as intensity of a gray level; we can display it as a 3D figure.

The value is called intensity.

Digital image processing refers to processing a digital image, where X and Y have finite elements (picture elements)or (pixels).

Images are not limit to the visible range or any other range of the EM spectrum.

107 108 107 106 99 ......
108 109 106 108 107
107 106 110 110 106
106 107 108 108 108
.....

$$Z = f\,(\mathrm{X}, \mathrm{Y}).$$

3dview.fig

```
I = imread('pout.tif');
figure; imshow(I);
[y x]=size(I);
figure;
mesh(1:x, 1:y, double…
        (I(end:-1:1, :)))
```

What is processing?

Low-level processing: the input is image and the output is image. Primitive operations, e.g., scaling, coloring…etc.

Mid-level processing: the input is image and the output is features, objects, regions,…etc. For recognition and classification.

High-level processing: the input is recognized objects, regions,… and the output is understanding making sense etc. This is the field of computer vision image analyses,…etc.

1920's Picture Transmission Systems

One of the first applications of digital images was in the newspaper industry. Pictures were sent by submarine cable between London and New York. Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours.

A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces.

Birth of Digital Image Processing (DIP)

- The first computers powerful enough to do meaningful image processing appeared in the early 1960s for the space program.
- DIP techniques began in the late 1960s and early 1970s to be used in medical imaging, remote Earth resources observations, and astronomy.
- The invention in the early 1970s of computerized axial tomography (CAT), also called computerized tomography (CT) for short, is one of the most important events in the application of image processing in medical diagnosis.

The first picture of the moon by the U.S. spacecraft *Ranger 7* , July 31,1964

What is, and how do we get, an Image?

- Images model variation in a detected level of activities over a spatial domain of interest as a 2-dimensional data set.
- Measured activities are either:
  - inherent in the imaged object itself

    e.g. Thermal emission, and Brain activities
  - result of interaction with to its environment

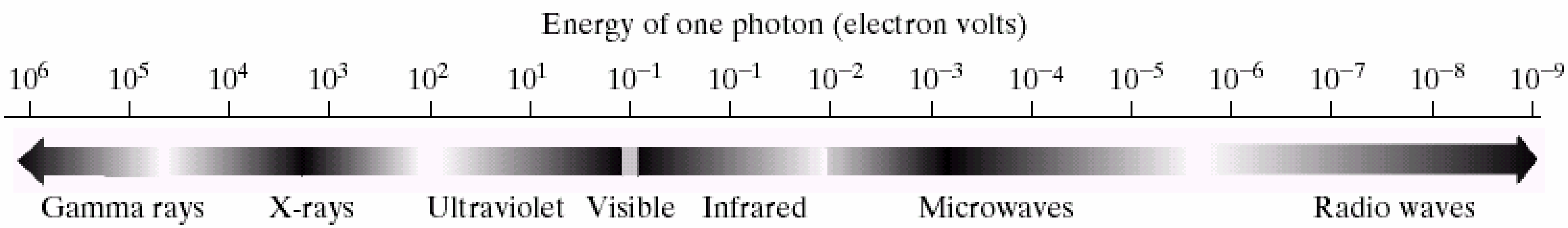
    e.g. Light reflected on the surface of objects
  - OR a combination of both

    e.g. X-rays and CAT images

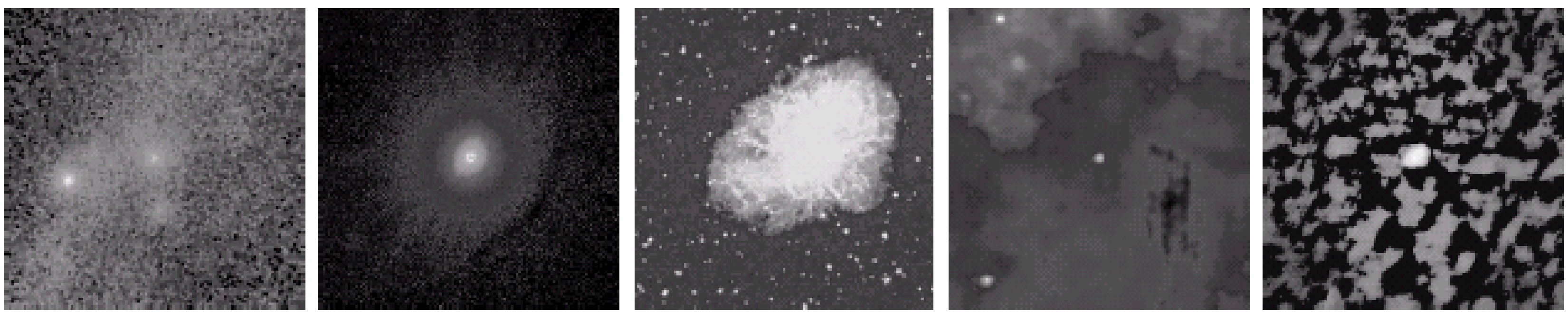
Types of Imaging Systems

- Imaging systems depend on energy sources
- Sources of energy include:
  - the electromagnetic energy (EM) spectrum,
  - Ultrasonic,
  - acoustic, and
  - electronic
- Accordingly there are different types of imaging systems and an ever growing list of applications.
- Multi-spectrum imaging is also available

The Electromagnetic Spectrum



The electromagnetic spectrum arranged according to energy per photon.



Gamma X-ray Optical Infrared Radio

Images of the Crab Pulsar (in the center of images) covering the electromagnetic spectrum. (Courtesy of NASA.)

1.3.1 Gamma-Ray Imaging.

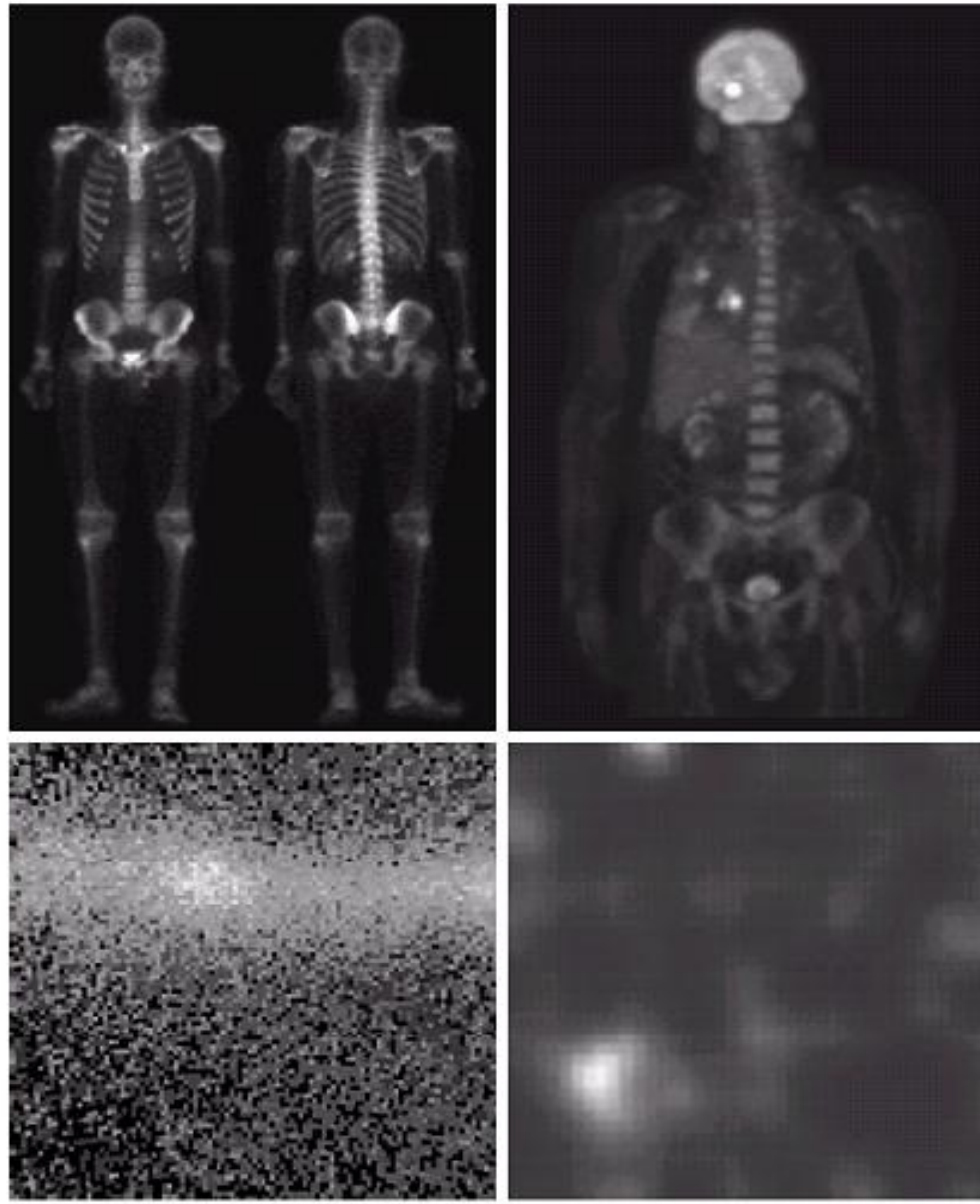
Major use in:

Nuclear medicine: injection of radio-active material (rather than external source of radiation) that decays after transmission and received by detector (Fig a, b). See the tumors in brain and lung of Fig. b.

Astronomy: Fig. c is a self radiating star that exploded 15,000 ago.

a b
c d

FIGURE 1.6 Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve. (Images courtesy of (a) G.E. Medical Systems, (b) Dr. Michael E. Casey, CTI PET Systems, (c) NASA, (d) Professors Zhong He and David K. Wehe, University of Michigan.)



1.3.2 X-Ray Imaging

Mainly used in medical imaging, but also used in astronomy.

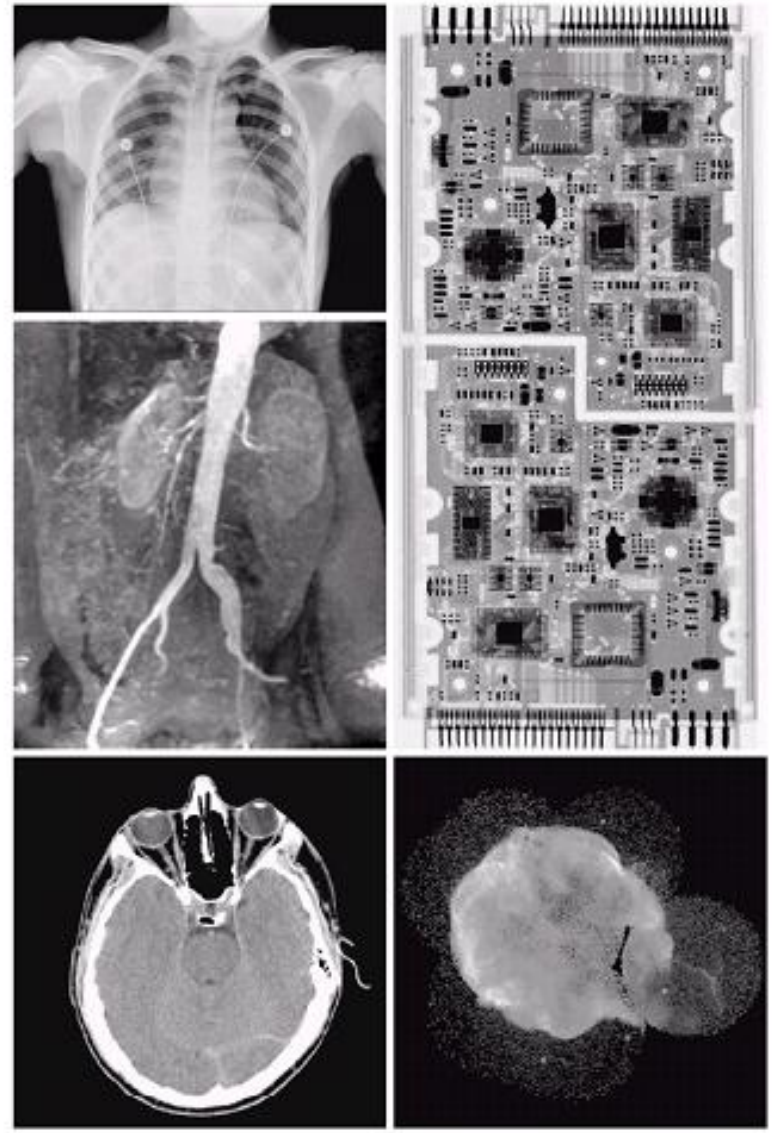
Emission is produced by heating a cathode and the patient is placed between it and the detector (which is a film). This is called analog X-ray.

The object modifies the X-Ray and, hence modulation is detected on the film.

Digital X-ray is produced by either digitizing the analog or directly by having the X-ray fall on digital device (e.g., digital mammography). See Fig. a., b.

Other X-Ray uses is CT scans, in which the object is sliced and each slice is imaged, Fig. c.

Similar uses is for X-Ray exist in industry, e.g., testing circuits. Fig. d



a d
b
c e

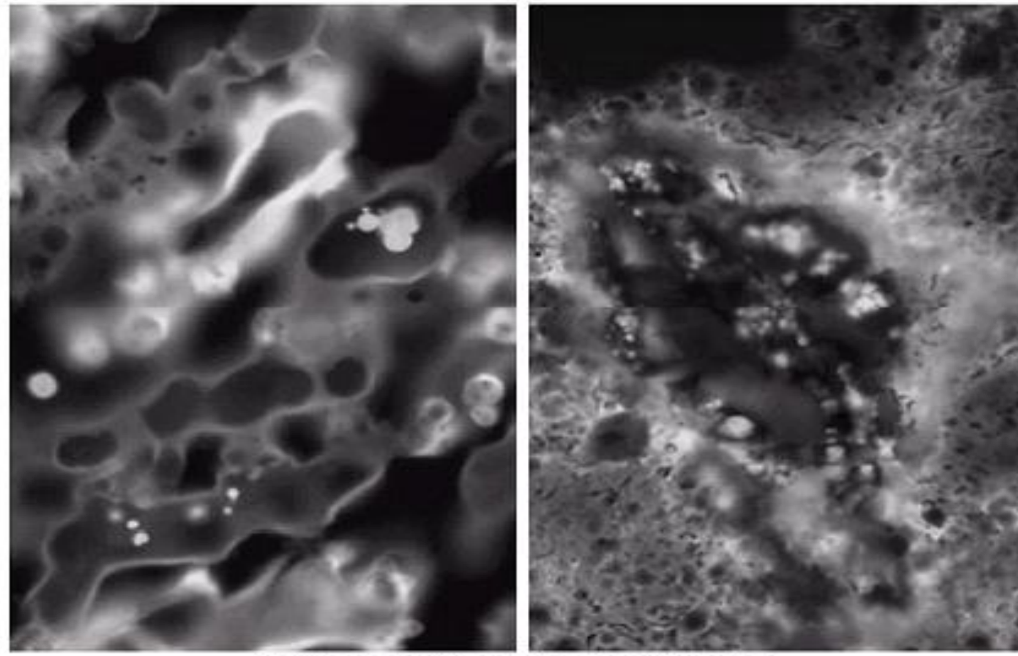
FIGURE 1.7 Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Circuit boards. (e) Cygnus Loop. (Images courtesy of (a) and (c) Dr. David R. Pickens, Dept. of Radiology & Radiological Sciences, Vanderbilt University Medical Center, (b) Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, (d) Mr. Joseph E. Pascente, Lixi, Inc., and (e) NASA.)

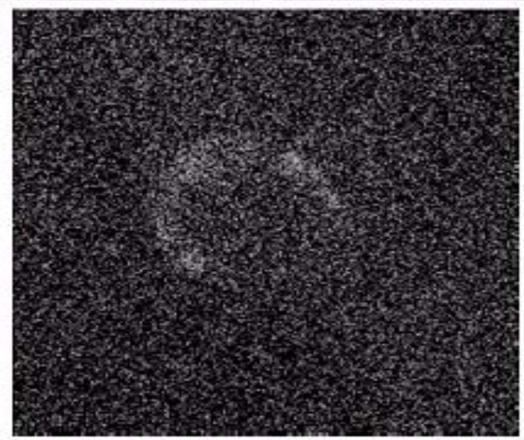
1.3.3 Imaging in the Ultraviolet Band

Appear in many applications, e.g., microscopy, lasers, biological imaging, and astronomical observations.

Fluorescence microscopy: when a photon of Ultraviolet light (not visible) collides with electron of fluorescent material it is elevated to higher energy level, and when relaxes it emits light in the visible region.

The material to be examined is either self fluorescing or, we treat it with another fluorescing material. See Fig. a., b.





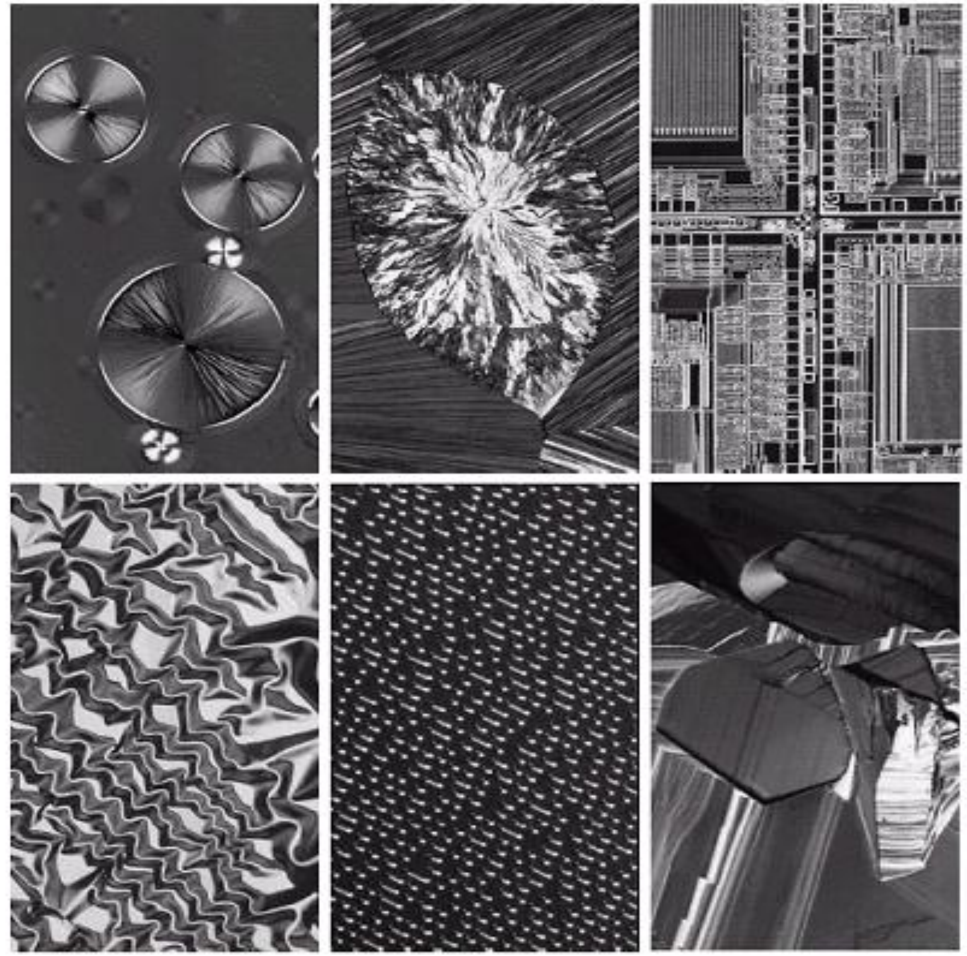
a b
c

FIGURE 1.8 Examples of ultraviolet imaging. (a) Normal corn. (b) Smut corn. (c) Cygnus Loop. (Images courtesy of (a) and (b) Dr. Michael W. Davidson, Florida State University, (c) NASA.)

1.3.4 Imaging in the Visible and Infrared Bands

Imaging in visible band is ubiquitous; frequently it is accompanied by infrared imaging.

The images in the Fig. are from light microscopes, but from different fields.



a b c
d e f

FIGURE 1.9 Examples of light microscopy images. (a) Taxol (anticancer agent), magnified 250×. (b) Cholesterol—40×. (c) Microprocessor—60×. (d) Nickel oxide thin film—600 ×. (e) Surface of audio CD—1750×. (f) Organic superconductor—450×. (Images courtesy of Dr. Michael W. Davidson, Florida State University.)

TABLE 1.1 Thematic bands in NASA's LANDSAT satellite.

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping

Remote Sensing: is another area of application for visible band; one object is imaged using different bands, all in the visible range (called thematic bands) in NASA's LANDSAT satellite.

In Fig 1.10, notice the difference between the infrared bands (4-7) and the first three; e.g., the river is so obvious in band 4 and 5.

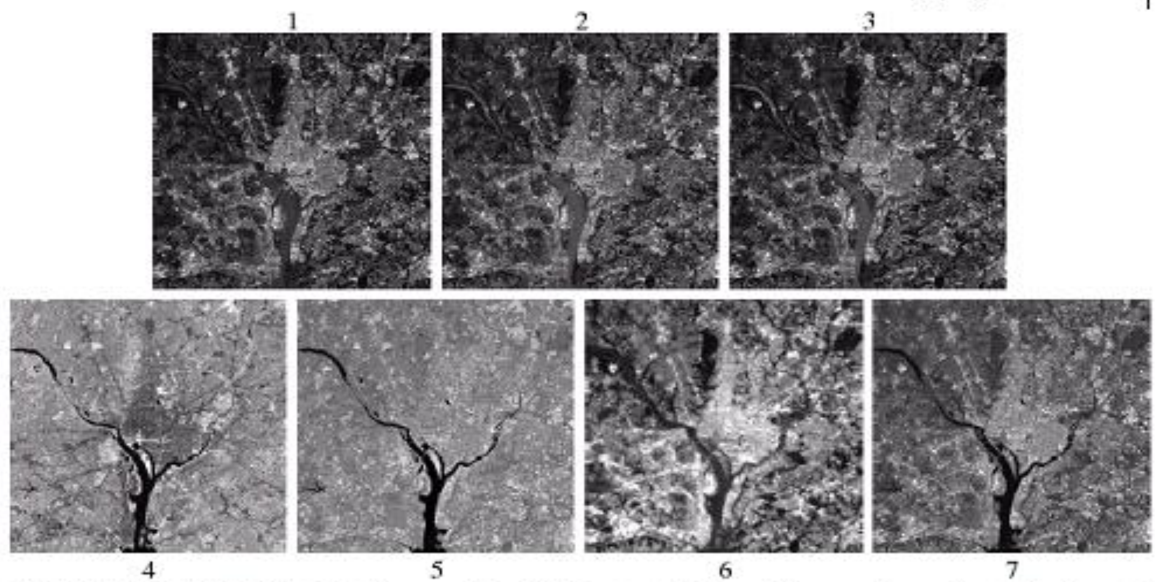


FIGURE 1.10 LANDSAT satellite images of the Washington, D.C. area. The numbers refer to the thematic bands in Table 1.1. (Images courtesy of NASA.)

Whether observation and Prediction: another application for multispectral imaging from satellites.

In the Fig., the eye of the hurricane is obvious. The image is taken in both the visible and the infrared bands.

FIGURE 1.11 Multispectral image of Hurricane Andrew taken by NOAA GEOS (Geostationary Environmental Operational Satellite) sensors. (Courtesy of NOAA.)

Fig 1.12 and 1.13 are images part of "*Nighttime Lights of the World*" dataset

This infrared imaging system has unique capability to observe faint sources of visible-near infrared emissions (this includes cities, towns, …).

It is very easy to calculate electrical energy usage by various regions in the world using this image.

Also, the difference is obvious between, e.g., US and Africa.

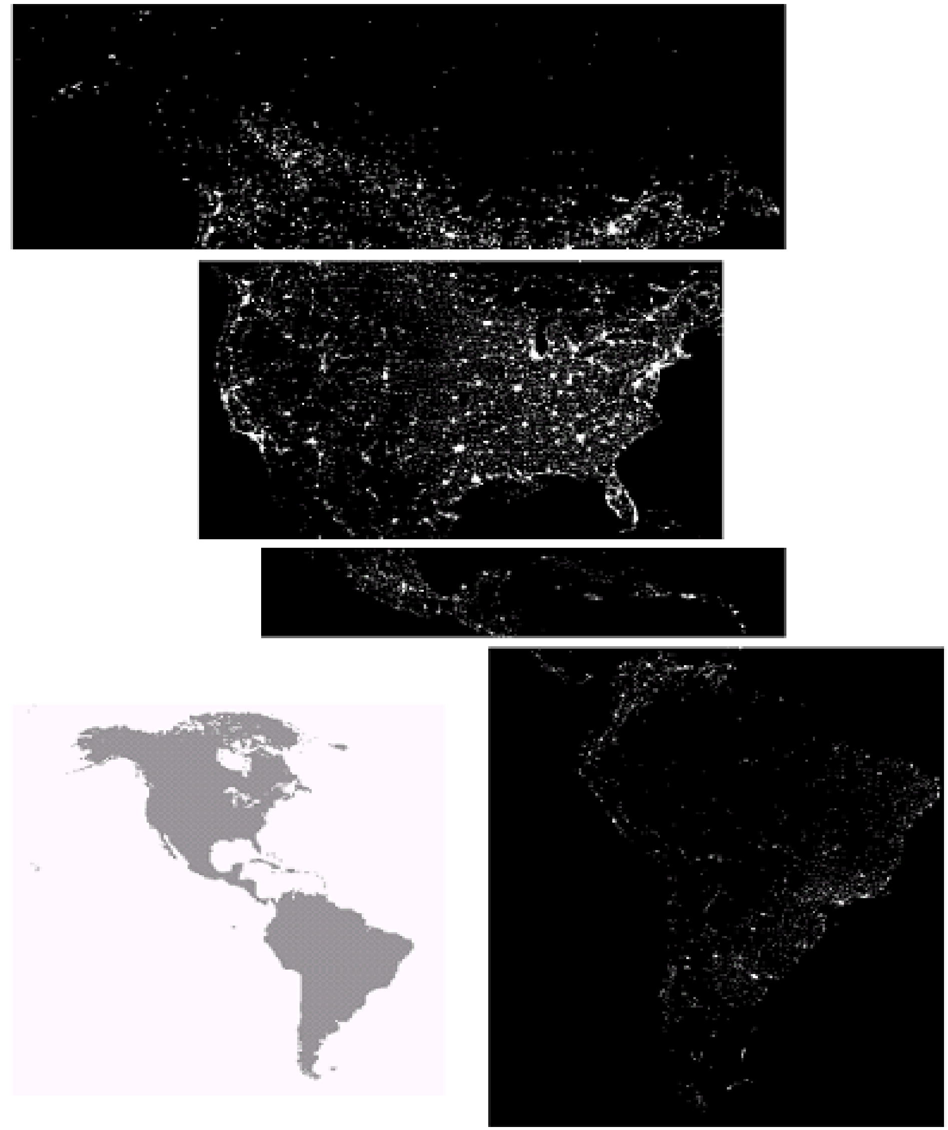


FIGURE 1.12 Infrared satellite images of the Americas. The small gray map is provided for reference. (Courtesy of NOAA.)

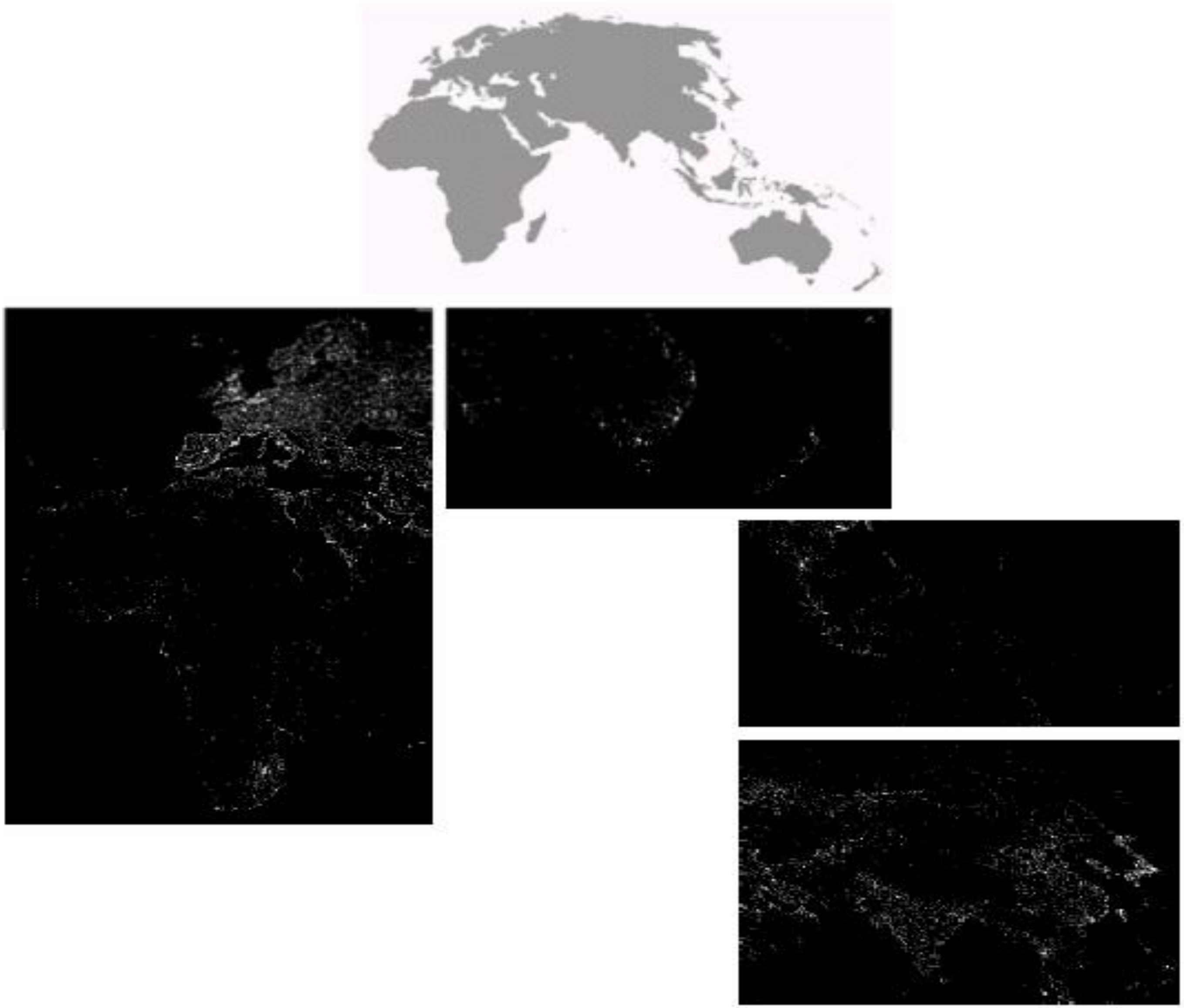


FIGURE 1.13 Infrared satellite images of the remaining populated part of the world. The small gray map is provided for reference. (Courtesy of NOAA.)

Automated visual inspection of manufactured goods:

Fig. a. the black square is a missing part.

Fig. b. no missing pills

Fig. c. There is a bottle that is not filled up.

Fig. d. unacceptable plastic product because of bubbles.

Fig. e. Some burned flakes exist, which degrades the quality.

Fig. f. detection of imperfections in lens.

a b
c d
e f

FIGURE 1.14 Some examples of manufactured goods often checked using digital image processing. (a) A circuit board controller. (b) Packaged pills. (c) Bottles. (d) Bubbles in clear-plastic product. (e) Cereal. (f) Image of intraocular implant. (Fig. (f) courtesy of Mr. Pete Sites, Perceptics Corporation.)

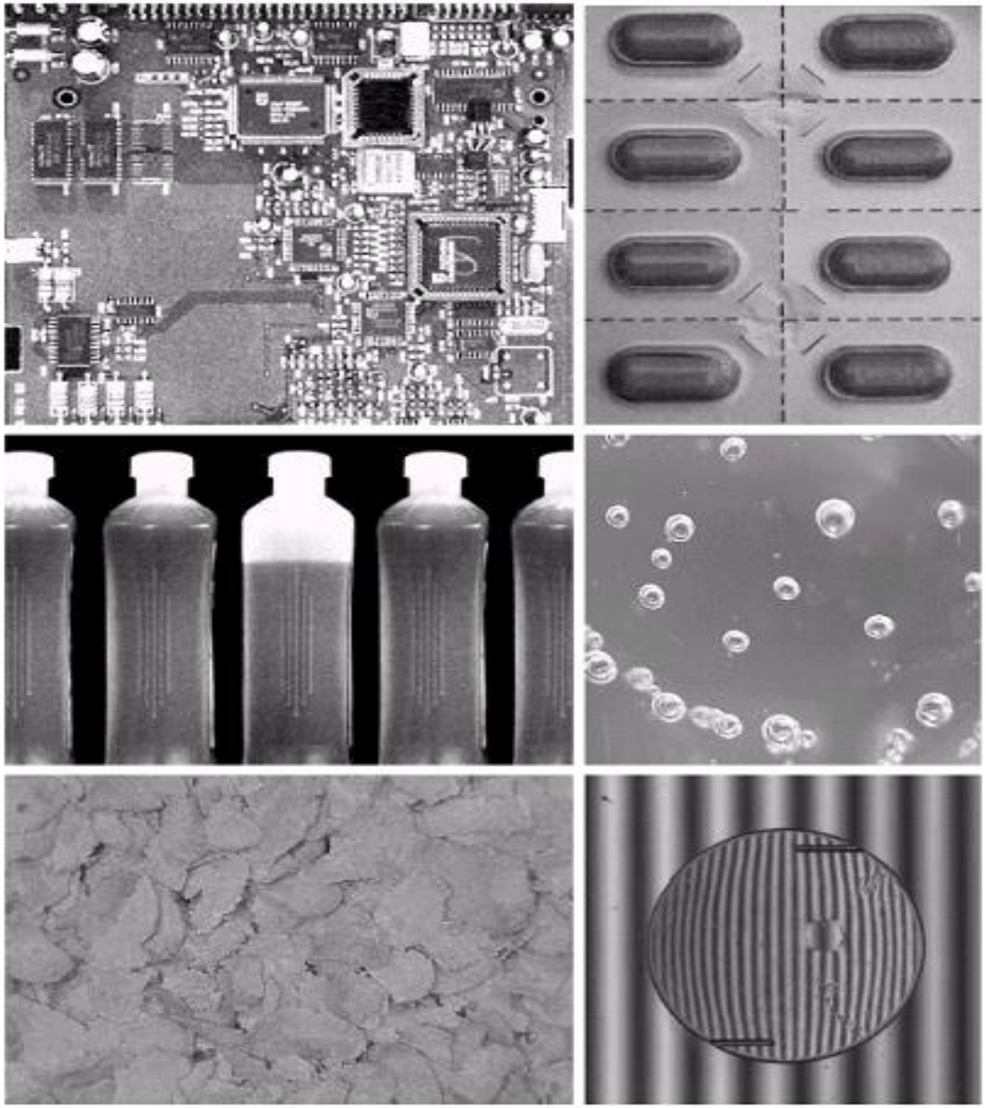


Fig. a. Finger print identification.

Fig. b. Automatic counting of bills, reading of serial numbers, …etc.

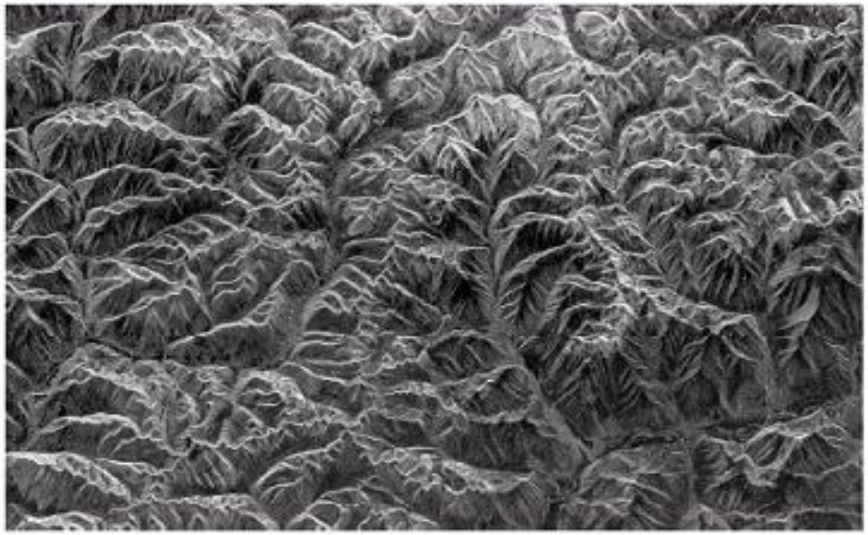
Fig. c. and d. automatic plate reading. The white rectangles are the areas detected by the system, and the black rectangles are the recognized numbers by OCR system.

a b
c
d

FIGURE 1.15 Some additional examples of imaging in the visual spectrum. (a) Thumb print. (b) Paper currency. (c) and (d). Automated license plate reading. (Figure (a) courtesy of the National Institute of Standards and Technology. Figures (c) and (d) courtesy of Dr. Juan Herrera, Perceptics Corporation.)

1.3.5 Imaging in the Microwave Band

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



Dominant application is radar. Some radar waves penetrate clouds and vegetation which makes it capable of collecting data over any region any time.

A flash camera produces microwave pulses, then reflects from the surface of the object to be detected and a snapshot image is taken.

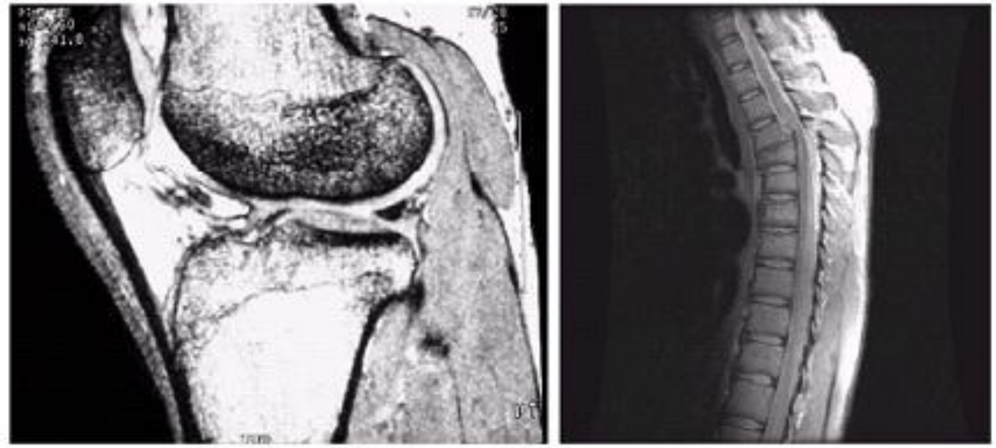
This image shows very clearly the mountains although at these heights there is a lot of clouds and other atmospheric conditions that interfere with visible light.

1.3.6 Imaging in the Radio Band

This is the other extreme end of the spectrum (as opposed to Gamma Rays).

Major application is in medical applications, e.g., Magnetic Resonance Imaging (MRI), and astronomy.

The patient is placed in a strong magnet and radio pulses are passed through his body. Each pulse results in another pulse emitted by the patient tissues. The strength and location is detected by a computer and an image is produced.



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.)

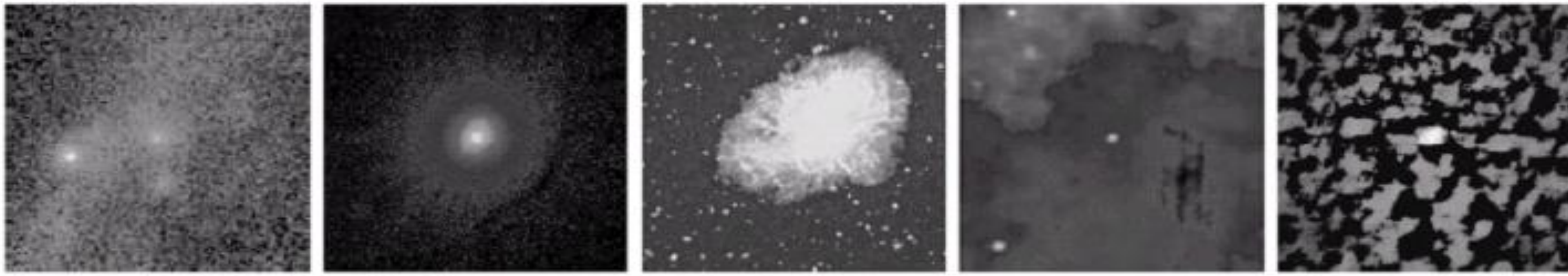


FIGURE 1.18 Images of the Crab Pulsar (in the center of images) covering the electromagnetic spectrum. (Courtesy of NASA.)

Many images in many bands for exactly the same object; totally different images!!

Which one of these is the object? The question is wrong, because all of these (and other images in other bands) are the interaction among three things:

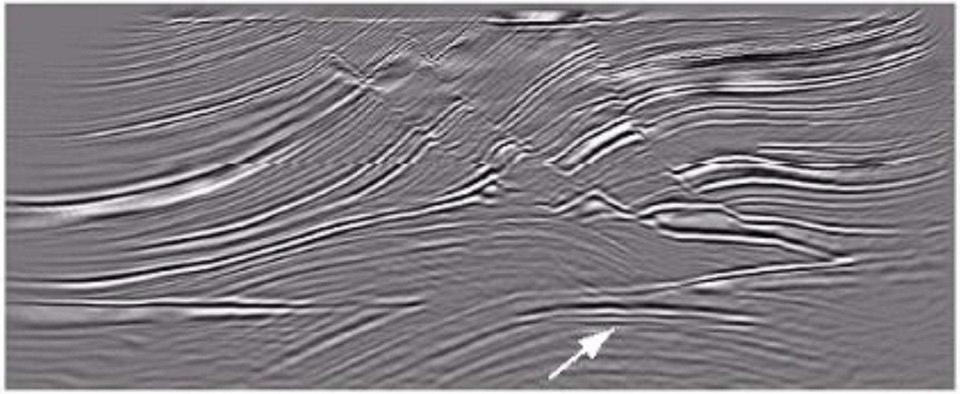
1- the wave hitting the object.
2- the quality of the object and how it reacts with the wave.
3- the receiver quality, whether it is the human eye or a special purpose camera.

No one knows the essence of anything; we cannot prove anything in science. We just observe indicators and understand in terms of these indicators. (More on this in Ch. 2).
We really do not know and cannot know but very little.
Even, we can use other modalities for the same object than the EM-based modality:

1.3.7 Examples in which Other imaging Modalities Are used

A. Acoustic Imaging.

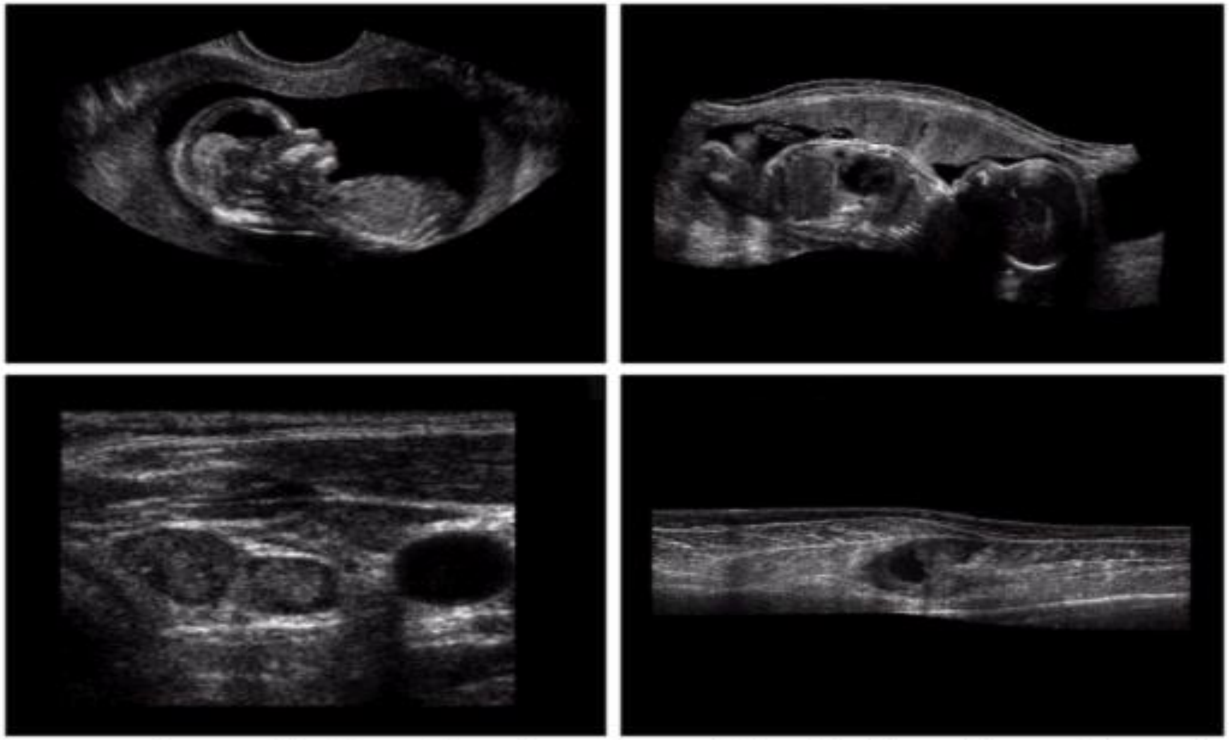
FIGURE 1.19 Cross-sectional image of a seismic model. The arrow points to a hydrocarbon (oil and/or gas) trap. (Courtesy of Dr. Curtis Ober, Sandia National Laboratories.)



Finds applications in Geology, e.g., mineral and oil exploration
In low end of the sound spectrum (hundreds of HZ)

Image acquisition over land is performed by putting a large flat steel sheet and vibrate it. The speed and frequency of returning depends on the earth below the surface.

Marine image acquisition is performed by using air guns behind the ship



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.)

Acoustic images in medical applications (specially for imaging unborn babies) use ultrasound (millions of HZ)

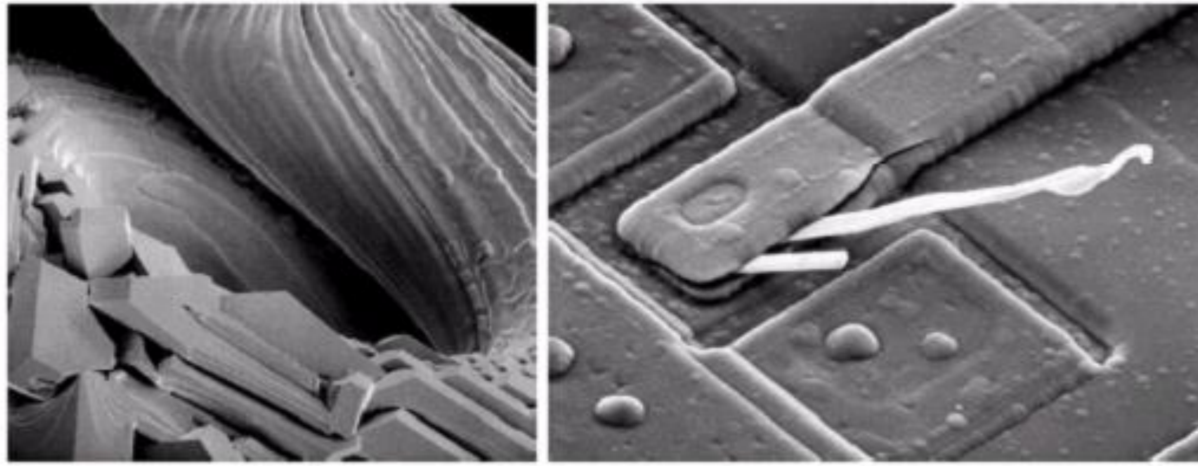
The idea is the same but with using a probe.

(of course other kinds of EM imaging is dangerous here.

B. Electron Microscopy

Electron microscopes work as optical ones except that they transmit electrons that penetrate the specimen, which absorbs and/or reflects according to its characteristics. Usually, used for inspecting components

Kinds of Electron microscopes are Transmission Electron Microscopes (TEM) and Scanning Electron Microscopes (SEM); read the book.



a b

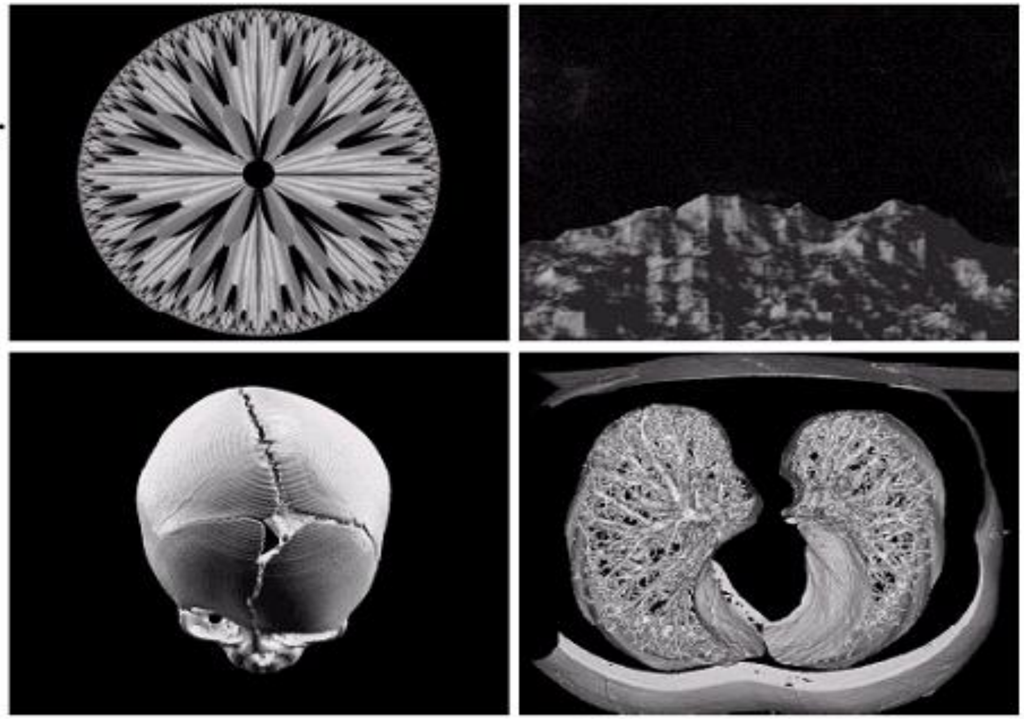
FIGURE 1.21 (a) 250× SEM image of a tungsten filament following thermal failure. (b) 2500× SEM image of damaged integrated circuit. The white fibers are oxides resulting from thermal destruction. (Figure (a) courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene;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

C. Fractals

Here, their **neither** object **nor** wave; it is synthesized by computers!!!

This is generated according to mathematical model (Fig. a., b.)

It can generate beautiful shapes and patterns.



a b
c d

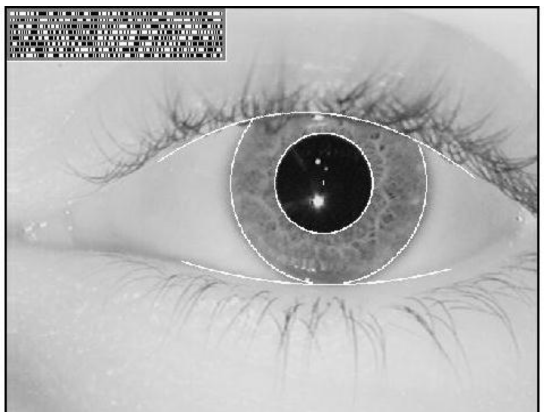
FIGURE 1.22 (a) and (b) Fractal images. (c) and (d) Images generated from 3-D computer models of the objects shown. (Figures (a) and (b) courtesy of Ms. Melissa D. Binde, Swarthmore College. (c) and (d) courtesy of NASA.)

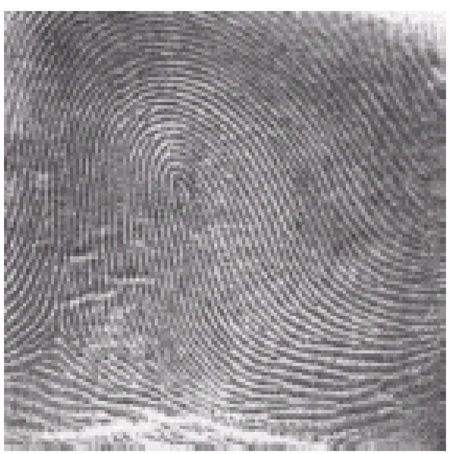
D. Model-based images

Also, their **neither** object **nor** wave; it is synthesized by computers!!! However, the model here is a model for some object, e.g., skulls, organs,..etc. (Fig. c., d.)

More advanced application is virtual reality.

Security applications





Most current Mobile Phones are equipped with digital cameras. Here we are showing image preprocessing procedure used for face recognition system for PDA developed at Buckingham University.

Digital Image Processing system components

- Digital Image Processing assumes the existence of a source of energy, a sensor devise to detect the emitted/reflected energy, a coding system for the range of measurements, and a display device.
- However, a modern DIP system requires powerful computing hardware, specialised software, large storage systems and communication devices.

Digital Image Processing system components

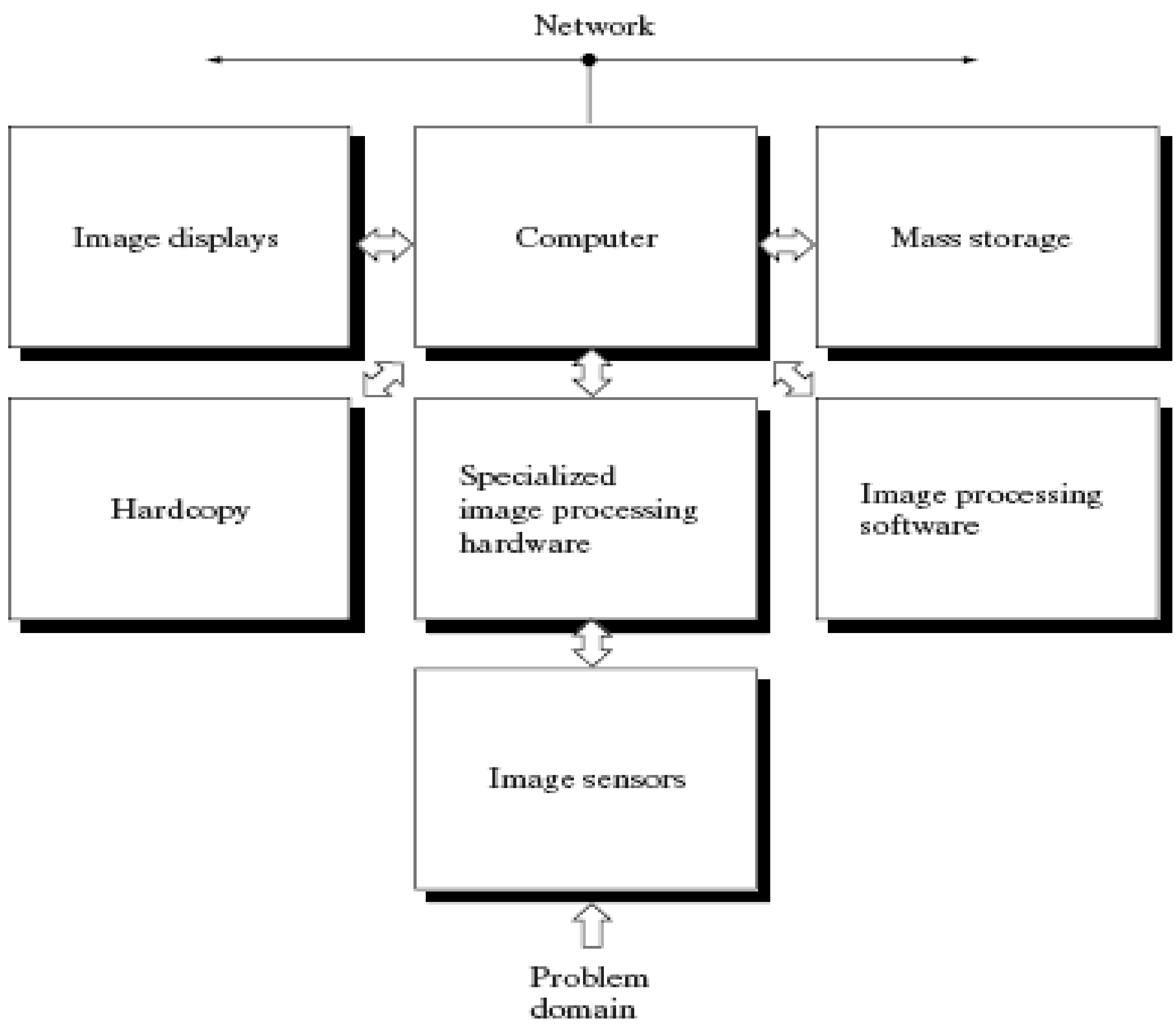
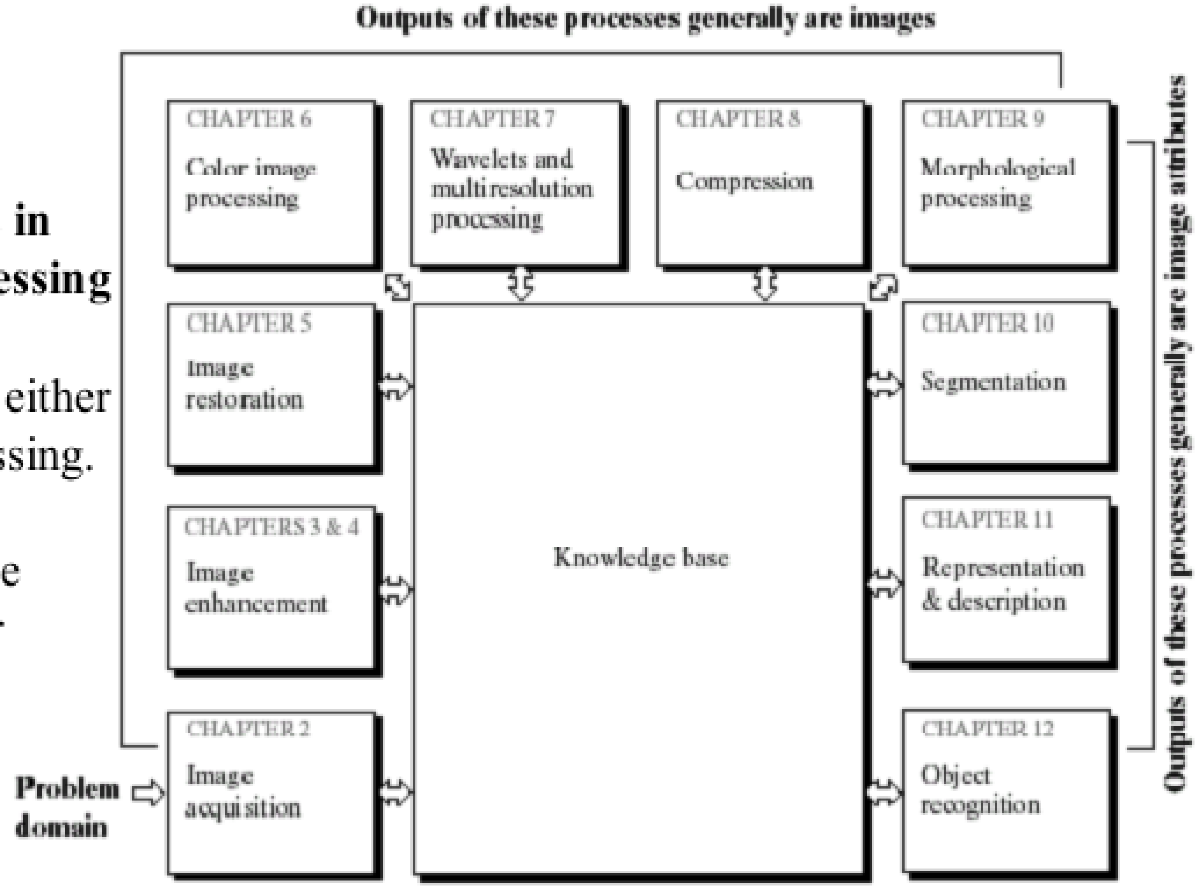


FIGURE 1.23 Fundamental steps in digital image processing.



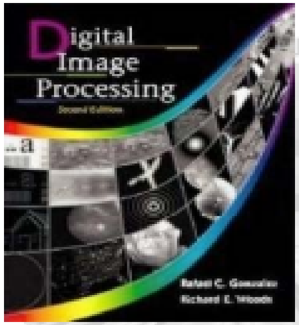
1.4 Fundamental Steps in Digital Image Processing

Each module of these is either low- or mid-level processing.

The knowledge about the problem is necessary for many of these modules.

Bilateral arrows indicate interaction between modules is done using the knowledge base.

End of Chapter 1



Digital Image Processing

Digital Image Processing Using Matlab

Digital Image Processing

Prepared by:

Dr. Ali J. Abboud

University of Diyala

2012-2013

Chapter 2
Digital Image Fundamentals

1. Elements of Visual Perception
2. Light and the Electromagnetic Spectrum
3. Image Sensing and Acquisition
4. Image Sampling and Quantization
5. Some Basic Relationships between Pixels
6. Image Histograms
7. Color Images
8. Image File Formats

1. Elements of Visual Perception

Structure of the Human Eye

Contracts & expands to control the amount of light entering the eye

Central opening of the Iris: Pupil (diameter: ~ 2 to 8 mm)

Concentric layers of fibrous cells
Absorb ~8% of the visible light spectrum

Innermost membrane of the eye
Light from object imaged on retina

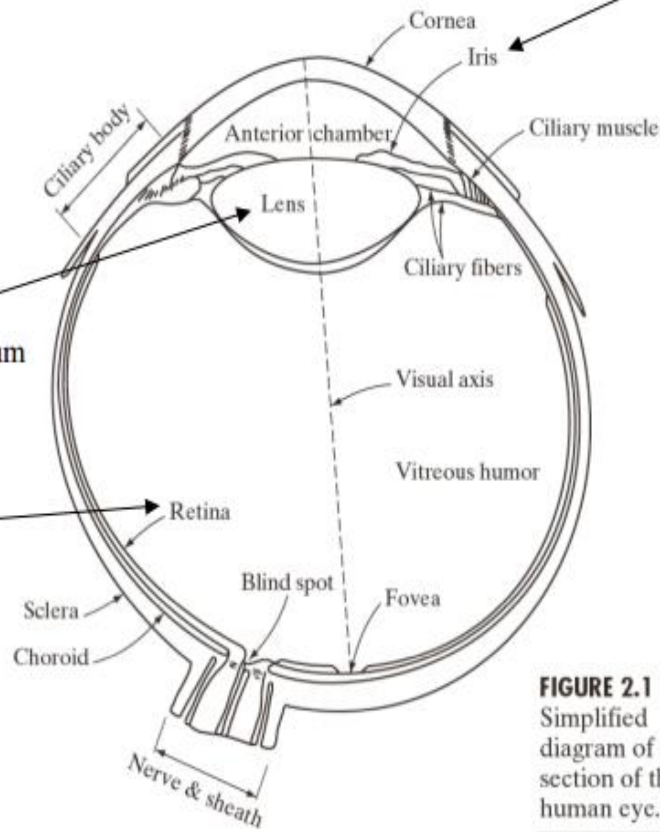


FIGURE 2.1 Simplified diagram of a cross section of the human eye.

1. Elements of Visual Perception

Structure of the Human Eye

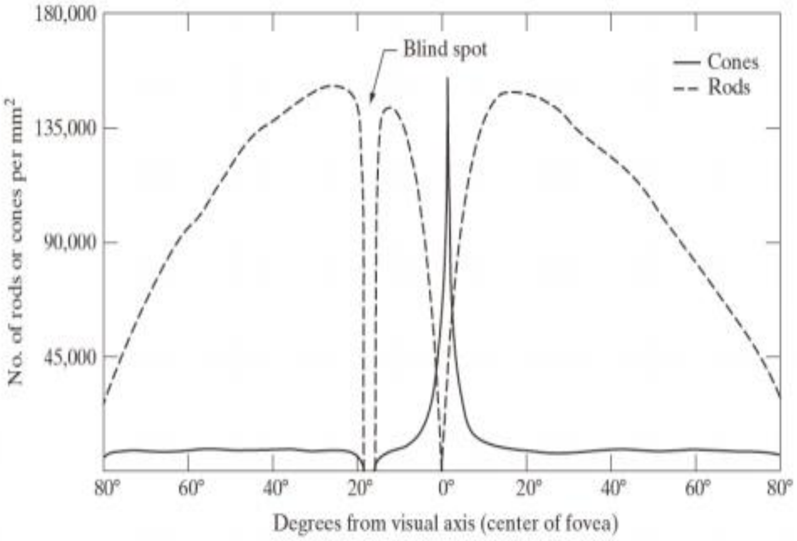
Distribution of discrete light receptors over the surface of the retina

2 classes of receptors: *cones* and *rods* :

• **Cones**: 6-7 million in each eye, mainly located in the fovea. Highly sensitive to colour, fine details.
"Photopic" or bright-light vision

• **Rods**: 75-150 million, distributed. Sensitive to low level of illumination, not involved in colour vision.
"Scotopic" or dim-light vision

FIGURE 2.2 Distribution of rods and cones in the retina.



Distribution of receptors is radially symmetric about the fovea, except the so-called "blind spot"

1. Elements of Visual Perception

Structure of the Human Eye

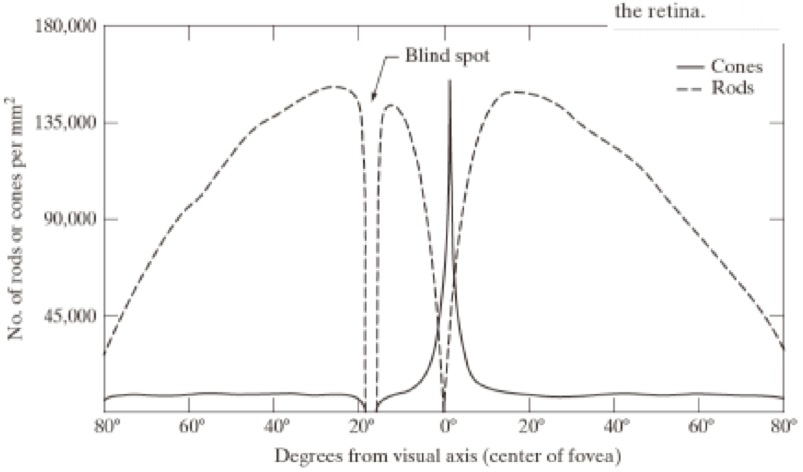


FIGURE 2.2 Distribution of rods and cones in the retina.

Approximation: fovea ≈ square sensor array of size 1.5 mm x 1.5 mm.

Density of cones in this area: 150,000 elements/mm^2

=> Number of cones in the region of highest acuity in the eye: ~337,000 elements.

Just in term of raw resolving power, a CCD can have this number of elements in a receptor array no larger than 5mm x 5mm.

=> basic ability of the eye to resolve detail is comparable to current electronic imaging sensors (but...)

1. Elements of Visual Perception

Image Formation in the Eye

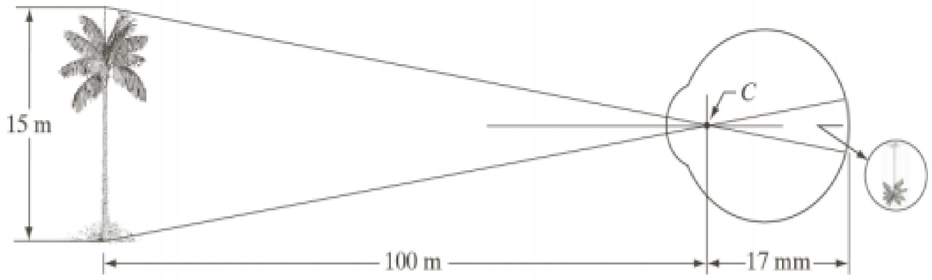


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

Photo camera: lens has *fixed focal length*. Focusing at various distances by *varying distance* between lens and imaging plane (location of film or chip)

Human eye: converse. *Distance* lens-imaging region (retina) is *fixed*. *Focal length* for proper focus obtained by *varying* the shape of the lens.

1. Elements of Visual Perception

Brightness Adaptation and Discrimination

Eye's ability to discriminate between different intensity levels

Range of light intensity levels to which the human visual system can adapt: on the order of 10^{10}

"Subjective Brightness"

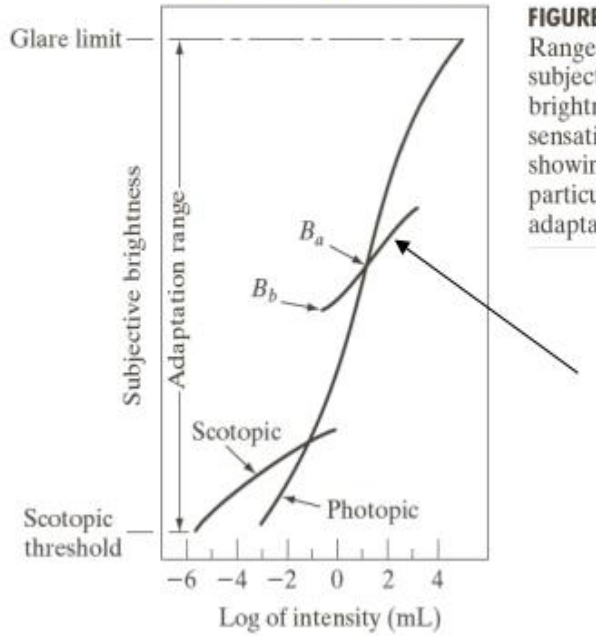


FIGURE 2.4 Range of subjective brightness sensations showing a particular adaptation level.

Range of *subjective brightness* the eye can perceive when adapted to this level B_a

1. Elements of Visual Perception

Vision carried out by rods

Poor brightness discrimination (large Weber ratio) at low levels of illumination

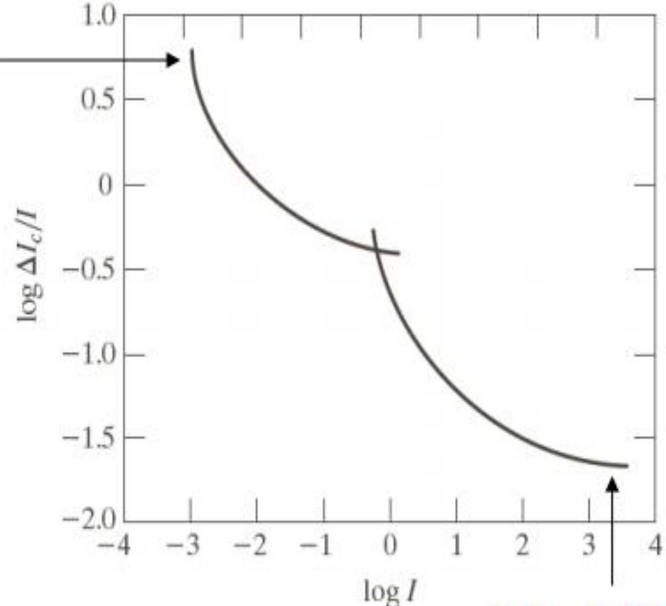


FIGURE 2.6 Typical Weber ratio as a function of intensity.

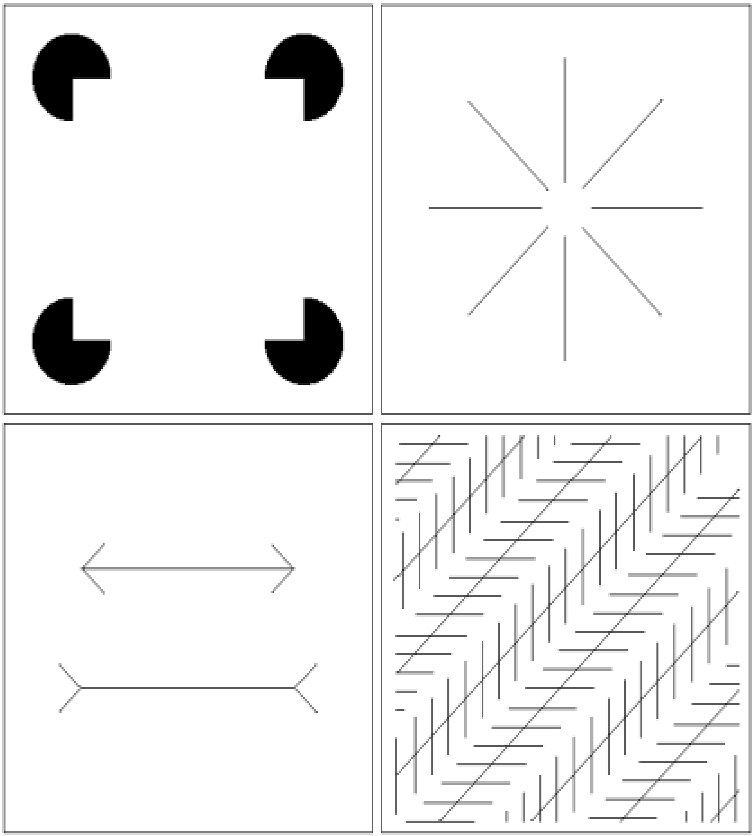
Vision is the function of cones

Better brightness discrimination as background illumination increases

1. Elements of Visual Perception

a b
c d

FIGURE 2.9 Some well-known optical illusions.



Same length? Parallel lines?

Optical illusions:

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

2. Light and the Electromagnetic Spectrum

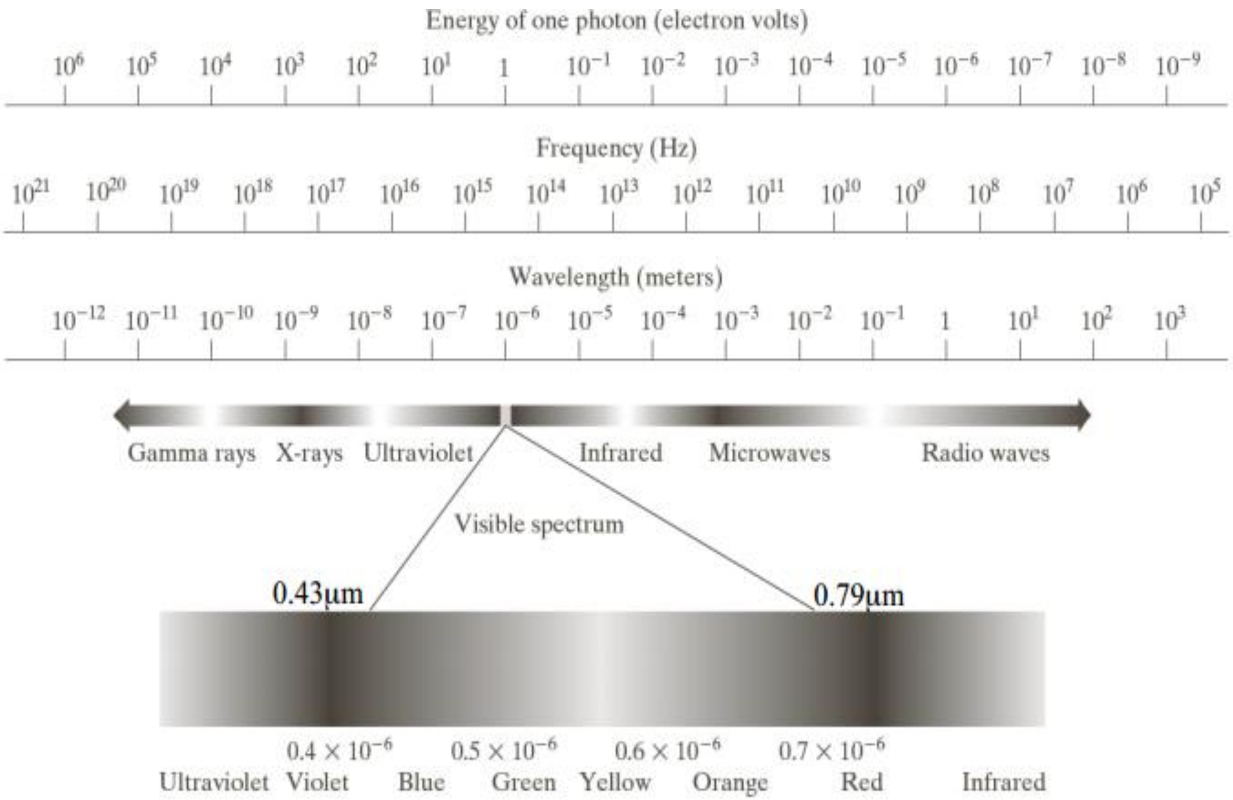
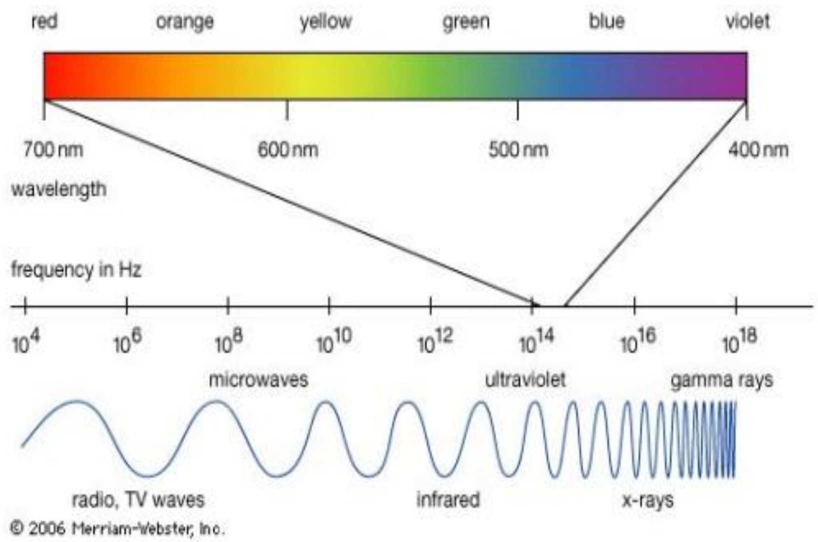


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

2. Light and the Electromagnetic Spectrum



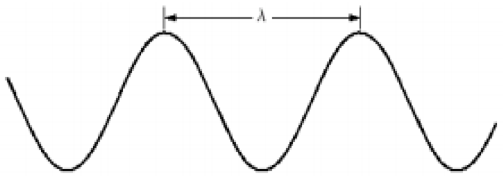
Wavelength (λ) and frequency (ν) related by: $\lambda = \frac{c}{\nu}$

c ≈ 2.998 x 10^8 m/s
λ in *microns* (µm=10^{-6} m) or
nanometers (nm=10^{-9} m)

Energy (eV): $E = h\nu$ (h: Planck's constant)

2. Light and the Electromagnetic Spectrum

FIGURE 2.11 Graphical representation of one wavelength.



• Light void of colour = ***monochromatic*** (or *achromatic*) light

=> only attribute : ***intensity*** or ***gray level***

• Range of measured values = *gray scale*

• Monochromatic images = ***gray-scale images***

Chromatic light source: frequency + radiance, luminance, brightness

• *Radiance* = total amount of energy that flows from the light source (W)

• *Luminance* (in lumens, lm) = measure of the amount of energy an observer *perceives* from a light source

• *Brightness* = subjective descriptor of light perception practically impossible to measure

3. Image Sensing and Acquisition

Transform of illumination energy into digital images:

The incoming energy is transformed into a voltage by the combination of input electrical power and sensor material.

Output voltage waveform = response of the sensor(s)

A digital quantity is obtained from each sensor by digitizing its response.

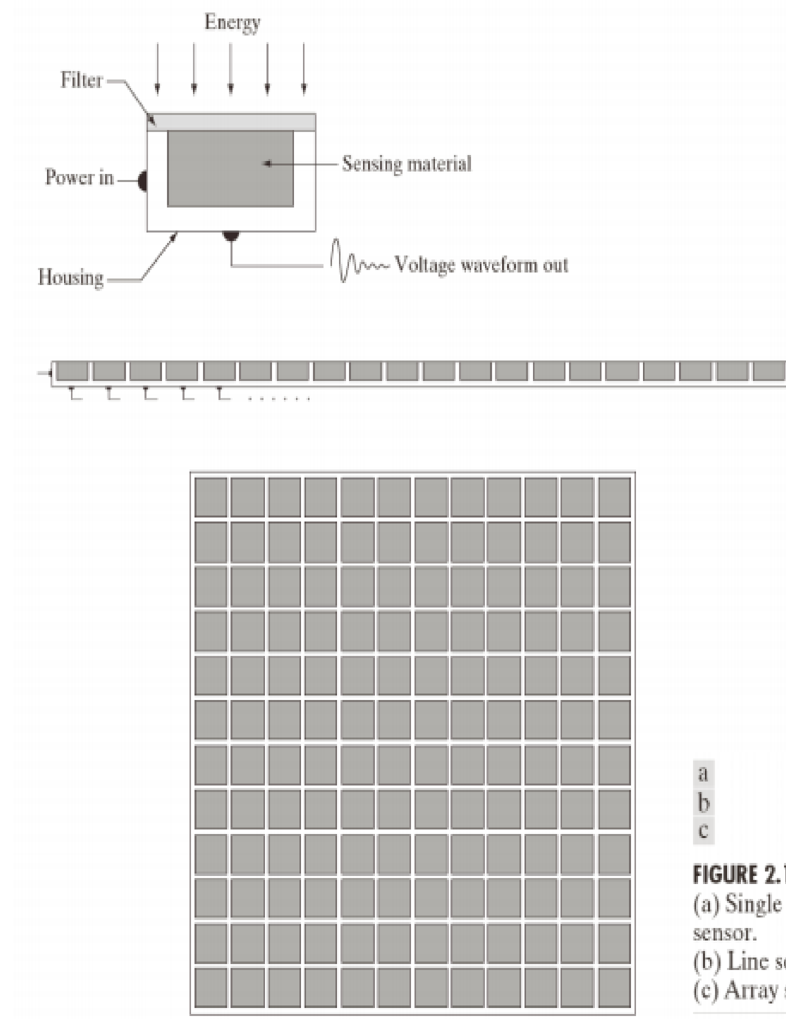
3. Image Sensing and Acquisition

Ex: Photodiode

Made of silicon

Output voltage waveform proportional to light

Filter in front: increase selectivity



a
b
c

FIGURE 2.12 (a) Single imaging sensor. (b) Line sensor. (c) Array sensor.

3. Image Sensing and Acquisition

Image acquisition using a single sensor

Arrangement for high precision scanning

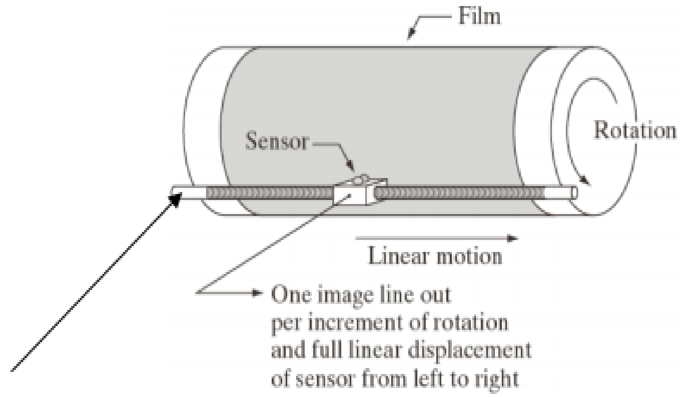


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

Lead screw

In-expensive (but slow) way to obtain high-resolution images

3. Image Sensing and Acquisition

- Illumination source reflected from a scene element
- Imaging system collects the incoming energy and focus it onto an image plane (sensor array)
- Response of each sensor proportional to the integral of the light energy projected
- Sensor output: analog signal → digitized

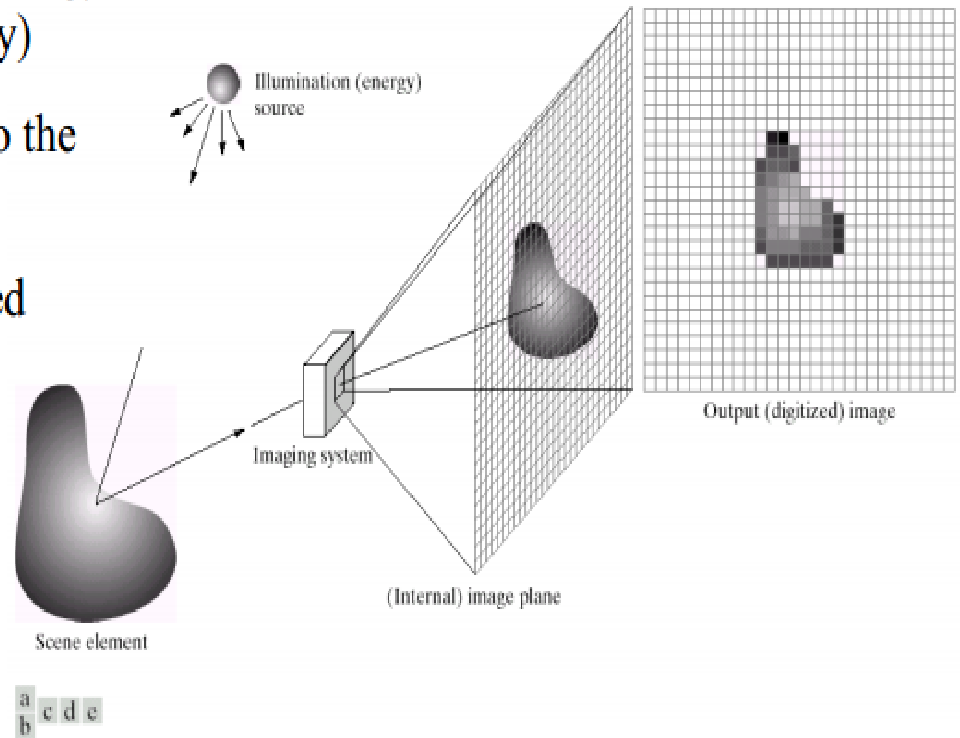


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

NB1: Motion not necessary
NB2: Predominant arrangement for digital cameras (e.g. CCD array)

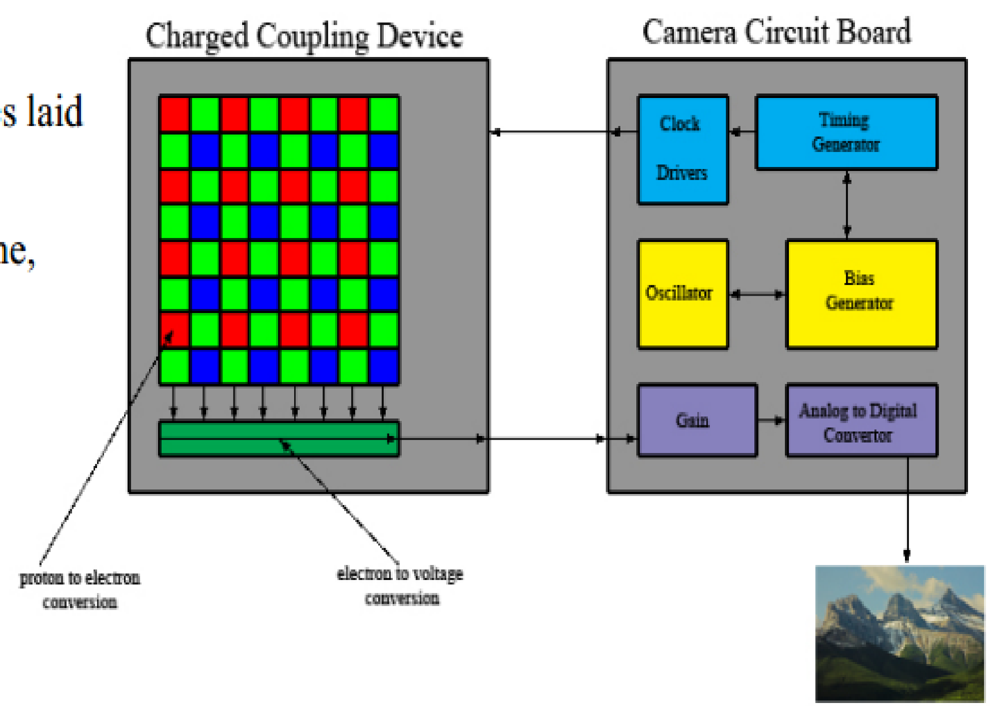
3. Image Sensing and Acquisition

CCD cameras: widely used in modern applications: private consumers, industry, astronomy…

CCD: Charge Couple Device

Rectangular grid of electron-collection sites laid over a thin silicon wafer

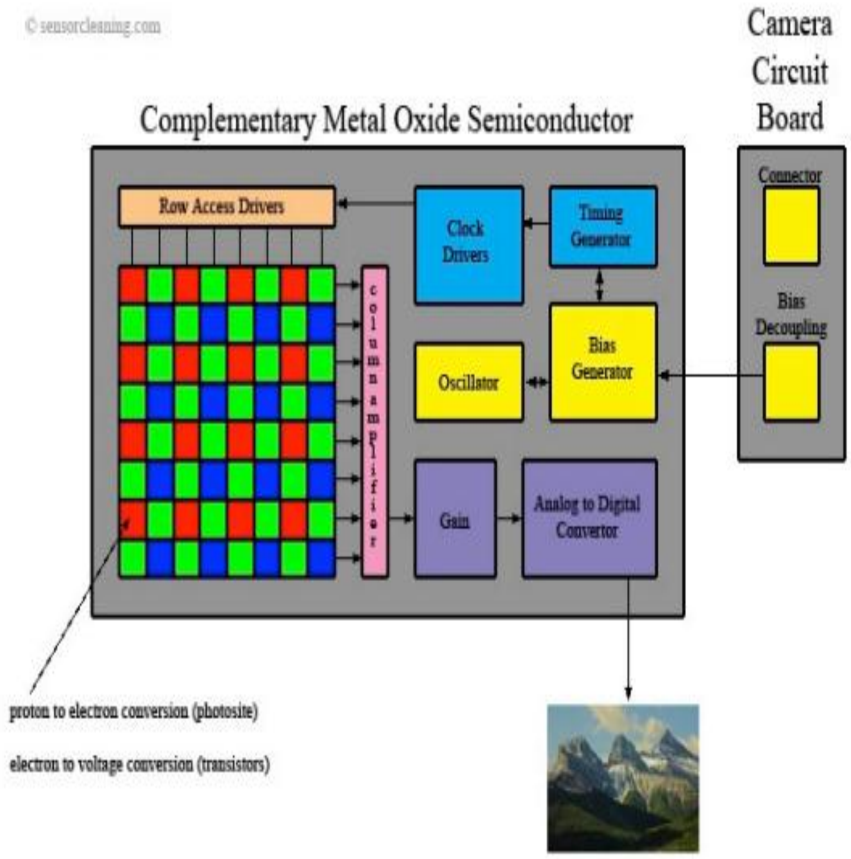
Image readout of the CCD one row at a time, each row transferred in parallel to a serial output register



3. Image Sensing and Acquisition

Alternative to CCD cameras: **CMOS** technology

CMOS: Complementary Metal-Oxyde-Semiconductor



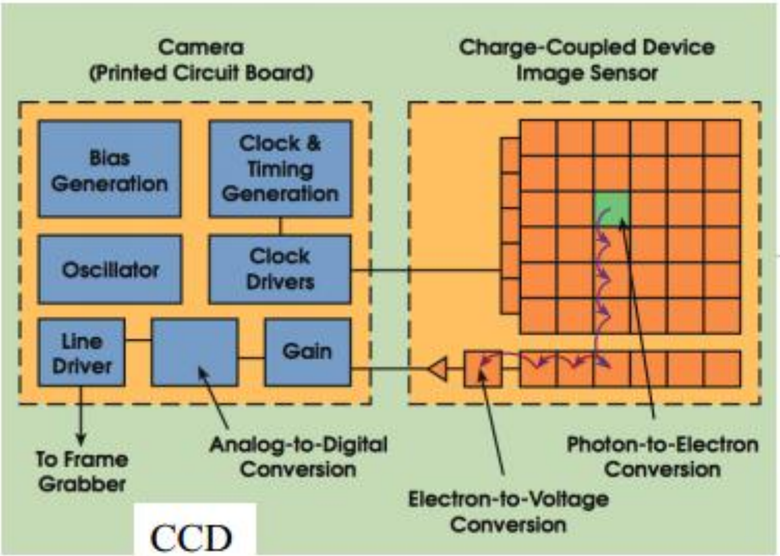
CMOS chip : active pixel sensor made using CMOS semiconductor

CMOS can potentially be implemented with fewer components, use less power and provide data faster than CCDs

CCD: more mature technology

NB: a CMOS-based camera can be significantly smaller than a comparable CCD camera

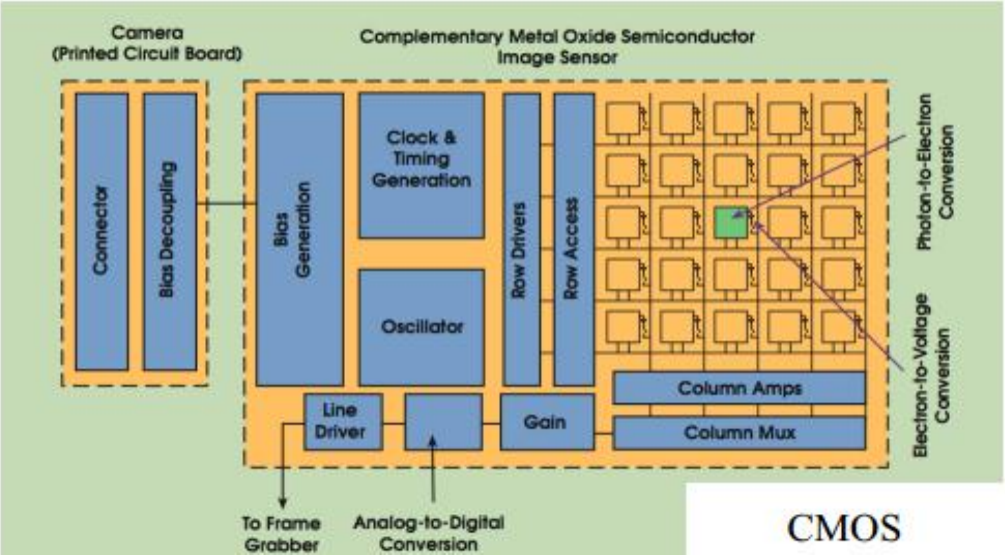
3. Image Sensing and Acquisition



CCD vs CMOS

CCD: when exposure complete, transfers each pixel's charge packet sequentially to a common output structure, which converts the charge to a voltage, buffers it and sends it off-chip.

CMOS imager: the charge-to-voltage conversion takes place in each pixel



From:

[D. Litwiller, *CCD vs. CMOS: Facts and Fiction*, Photonics Spectra, January 2001, Laurin Publishing Co. Inc.]

3. Image Sensing and Acquisition

CCD vs CMOS

• **Responsivity** *(amount of signal the sensor delivers per unit of input optical energy):* CMOS imagers marginally superior to CCDs

• **Dynamic range** *(ratio of a pixel's saturation level to its signal threshold):* CCDs have advantage by factor of 2 in comparable circumstances

• **Uniformity** *(consistency of response for different pixels under identical illumination conditions):* CMOS imagers "traditionally worse"

• **Shuttering** *(ability to start and stop exposure arbitrarily):* standard feature of virtually all consumer and industrial CCDs

[D. Litwiller, *CCD vs. CMOS: Facts and Fiction*, Photonics Spectra, January 2001, Laurin Publishing Co. Inc.]

3. Image Sensing and Acquisition

CCD vs CMOS

- **Speed**: CMOS arguably has the advantage over CCDs (all camera functions can be placed on the image sensor)
- **Windowing**: CMOS has ability to read out a portion of the image sensor (=> elevated frame or line rates for small ROI[1]). CCDs generally more limited
- **Antiblooming** *(ability to gracefully drain localized overexposure without compromising the rest of the image in the sensor)*: CMOS generally has natural blooming immunity, CCDs require specific engineering
- **Reliability**: CMOS have advantage (all circuit functions can be placed on a single integrated circuit chip)

Blooming effect

[D. Litwiller, *CCD vs. CMOS: Facts and Fiction*, Photonics Spectra, January 2001, Laurin Publishing Co. Inc.]

(1) ROI = Region of Interest

3.1. A Simple Image Formation Model

A Simple Image Formation Model

Images denoted by two-dimensional functions $f(x,y)$

Value of amplitude of f at (x,y): positive scalar quantity

Image generated by physical process: intensity values are proportional to the energy radiated by a physical source => $0 < f(x,y) < \infty$

$f(x,y)$ may be characterized by 2 components:

(1) The amount of source illumination *incident* on the scene: *illumination* $i(x,y)$

(2) The amount of illumination *reflected* by the objects of the scene: *reflectance* $r(x,y)$

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

↑ *total absorption* (at 0)

↑ *total reflectance* (at 1)

3.1. A Simple Image Formation Model

A Simple Image Formation Model

Example of typical ranges of illumination *i(x,y)* for visible light (average values):

- Sun on a clear day: ~ 90,000 lm/m^2, down to 10,000 lm/m^2 on a cloudy day
- Full moon on a clear evening: ~0.1 lm/m^2
- Typical illumination level in a commercial office: ~1000 lm/m^2

Typical values of reflectance *r(x,y)*:

- 0.01 for black velvet
- 0.65 for stainless steel
- 0.8 for flat white wall paint
- 0.9 for silver-plated metal
- 0.93 for snow

3.1. A Simple Image Formation Model

A Simple Image Formation Model

Monochrome image

Intensity l: $L_{min} \leq l \leq L_{max}$. In practice: $L_{min}=i_{min}\ r_{min}$ and $L_{max} = i_{max}\ r_{max}$

Typical limits for indoor values in the absence of additional illumination:
$L_{min} \approx 10$ and $L_{max} \approx 1000$

$[L_{min}, L_{max}]$ is called the *gray* (or *intensity*) *scale*

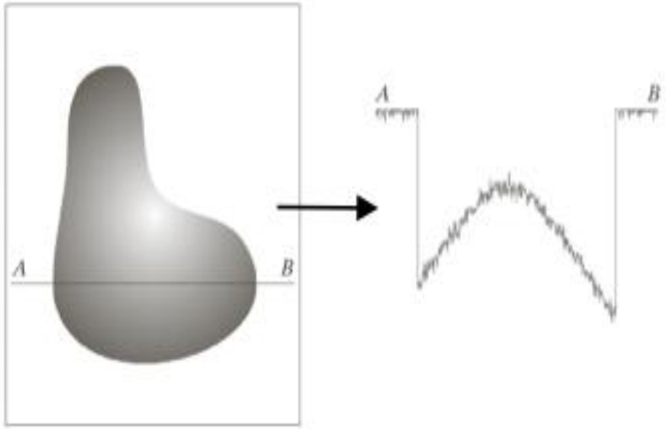
Common practice: shift to $[0, L-1]$, where $l=0$ is considered black and $l=L-1$ is considered white

4. Image Sampling and Quantization

Basic Concepts in Sampling and Quantization

Digitizing the coordinate values = Sampling

Digitizing the amplitude values = Quantization



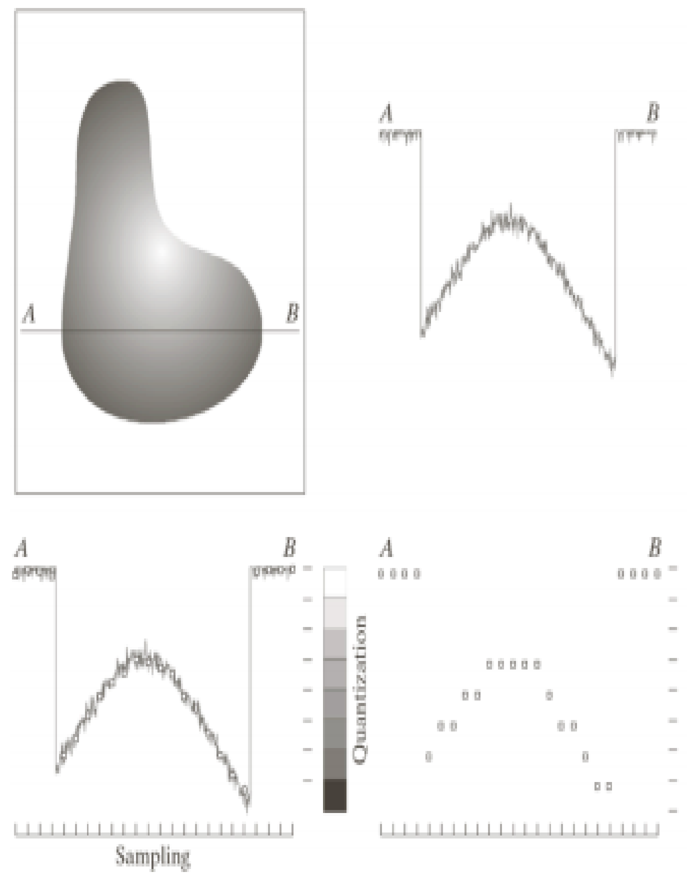
a b
c d

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

4. Image Sampling and Quantization

Method of **sampling** determined by the sensor arrangement:

- **Single sensing element combined with motion:** spatial sampling based on number of individual mechanical increments
- **Sensing strip:** the number of sensors in the strip establishes the sampling limitations in one image direction; in the other: same value taken in practice
- **Sensing array:** the number of sensors in the array establishes the limits of sampling in both directions



4. Image Sampling and Quantization

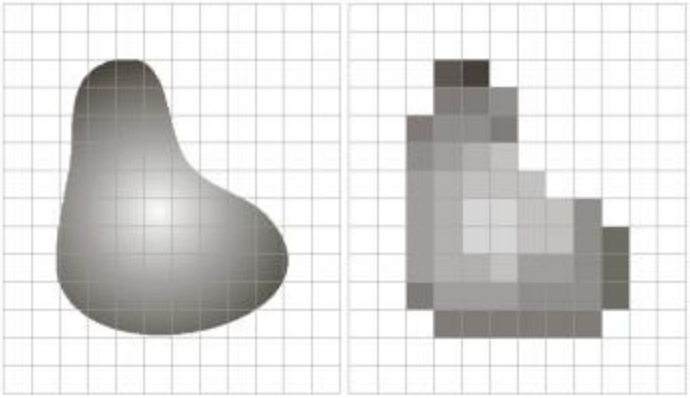


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

The *quality of a digital image* is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization.

However image content is also an important consideration in choosing these parameters

4. Image Sampling and Quantization

2 Representing Digital Images

Continuous image: function of 2 continuous variables $f(s,t)$

→ *digital image* by sampling and quantization

→ 2D array $f(x,y)$, M rows and N columns, (x,y) = discrete coordinates

$x = 0, 1, 2, \ldots, M\text{-}1$ and $y = 0, 1, 2 \ldots, N\text{-}1$

Section of the real plane spanned by the coordinates of an image = *spatial domain*

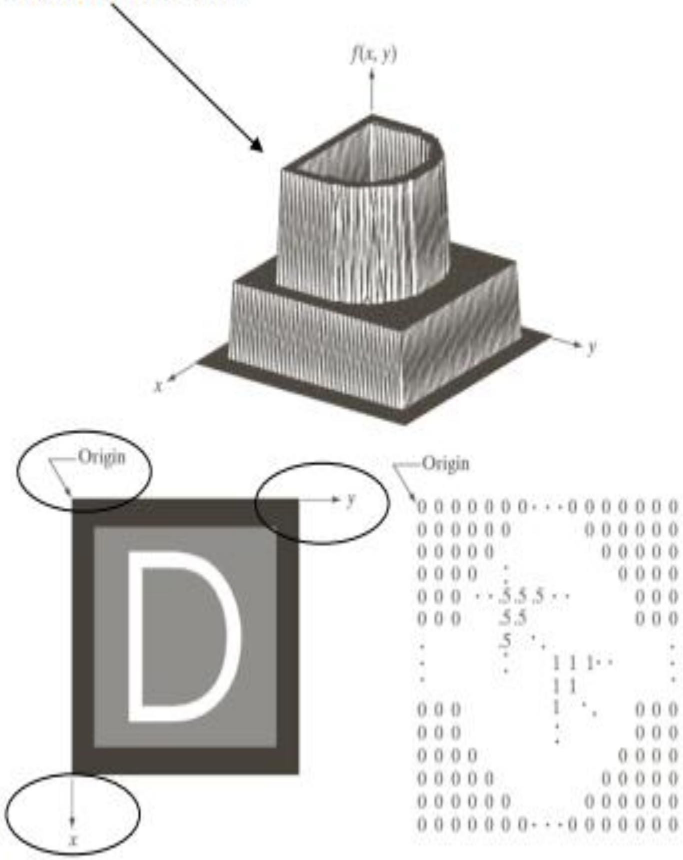
x and y are called *spatial variables* or *spatial coordinates*

4. Image Sampling and Quantization

2 Representing Digital Images

Representation useful for gray-scale images

NB: Origin and axes
→ TV + matrix



a
b c

FIGURE 2.18 (a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

F of size 600x600 here
= 360,000 numbers…
Useful for algorithms

4. Image Sampling and Quantization

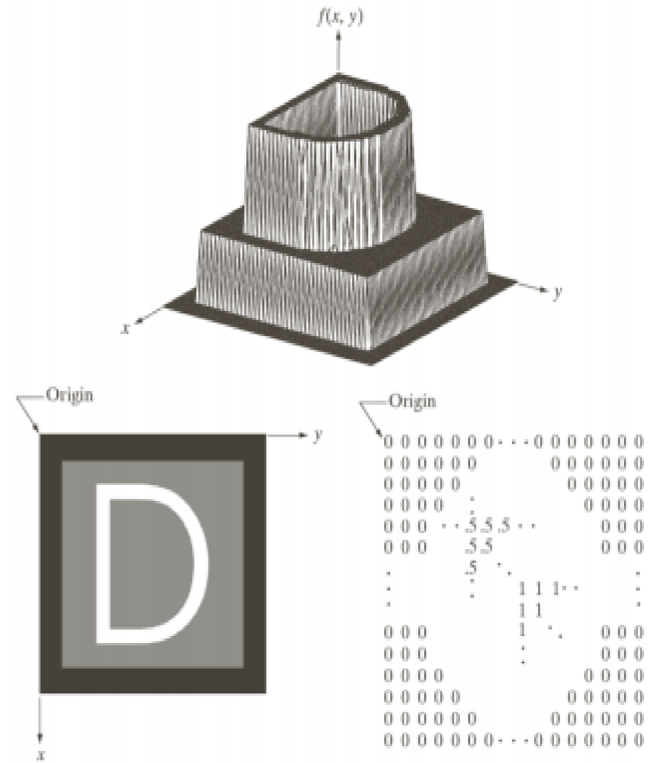
Representing Digital Images

Sampling => $(x,y) \rightarrow f(x,y) = z$
$Z^2 \rightarrow R$

Quantization => $(x,y) \rightarrow f(x,y) = z \in [0,L-1]$
$Z^2 \rightarrow Z$

The digitization process requires decisions on the values of M, N and L (number of discrete intensity levels)

No (theoretical) restrictions on M and N other than:
$M > 0$ and $N > 0$
Due to storage and hardware, typically: $L = 2^k$
Assume that discrete levels are equally spaces and integers in [0,L-1]



4. Image Sampling and Quantization

Dynamic range = ratio of maximum measurable intensity to minimum detectable intensity level in the system
Rule: upper limit determined by *saturation*, lower limit determined by *noise*

Contrast = difference in intensity between the highest and the lowest intensity levels in an image

High dynamic range => high contrast expected

Low dynamic range => dull, washed-out gray look

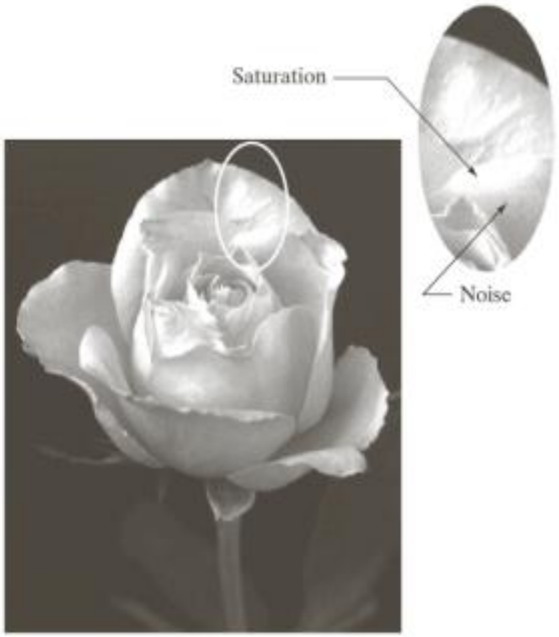


FIGURE 2.19 An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.

4. Image Sampling and Quantization

Decreasing spatial resolution reduces image quality proportionally - Checkerboard pattern.

† Images extracted from DIP, 2nd Edition, Gonzalez & Woods, PH.

4. Image Sampling and Quantization

- The checkerboard effect is not visible if a lower–resolution image is displayed in a proportionately small window.

4. Image Sampling and Quantization

Image f =

123	162	200	147	93
137	157	165	232	189
151	155	152	141	130
205	101	100	193	115
250	50	75	88	100

8 bits

f(i,j)← int(f(i,j)/2) →

61	80	100	73	46
68	78	82	116	94
75	77	76	70	65
102	50	50	96	57
125	25	37	43	50

7 bits

30	40	50	36	23
34	39	41	58	47
37	38	38	35	32
51	25	25	48	28
62	12	18	21	25

6 bits

15	20	25	18	11
17	19	20	29	23
18	19	19	17	16
25	12	12	24	14
31	6	9	10	12

5 bits

7	10	12	9	5
8	9	10	14	11
9	9	9	8	8
12	6	6	12	7
15	3	4	5	6

4 bits

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

3 bits

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

2 bits

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

1 bits

Original image f is reasonably bright, but gradually the pixels get darker as the Grey-level resolution decreases.

4. Image Sampling and Quantization

8 bits

7 bits

6 bits

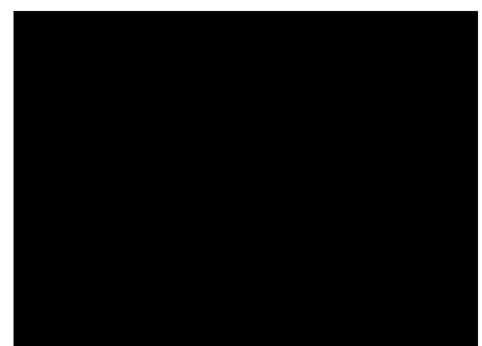
5 bits

4 bits

3 bits

2 bits

1 bit



0 bits !!!

4. Image Sampling and Quantization

Number b of bits required to store an image:
B = M x N x k
M = N => b = N^2 k

Image with 2^k intensity levels => "*k-bit image*" (ex: 256 → 8-bit image)

TABLE 2.1
Number of storage bits for various values of N and k.

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

L = Number of intensity levels

4. Image Sampling and Quantization

3 Spatial and Intensity Resolution

Intuitively, *spatial resolution* = measure of the smallest discernible detail in an image

Quantitatively (most common measures): line pairs per unit distance or dots (pixels) per unit distance (printing and publishing industry). In the US: dots per inch (dpi)
e.g. newspapers: 75 dpi, magazines: 133 dpi, glossy brochures: 175 dpi, DIP book: 2400 dpi

Key point: to be meaningful, measures of spatial resolution must be stated *w.r.t. spatial units*

Intensity resolution = smallest discernible change in intensity level
Most common: 8bit. 16bit when needed. 32 bits rare. Exceptions: 10 or 12 bits

4. Image Sampling and Quantization

Effects of Sampling

Original image: 3692 x 2812 pixels
72 dpi image: 213 x 162 array
Smaller images zoomed back to the original size

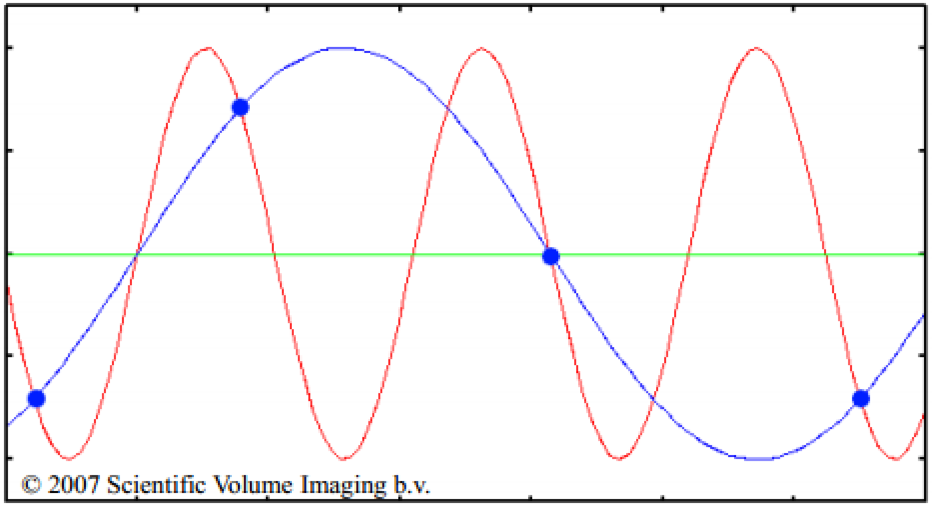
a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

4. Image Sampling and Quantization

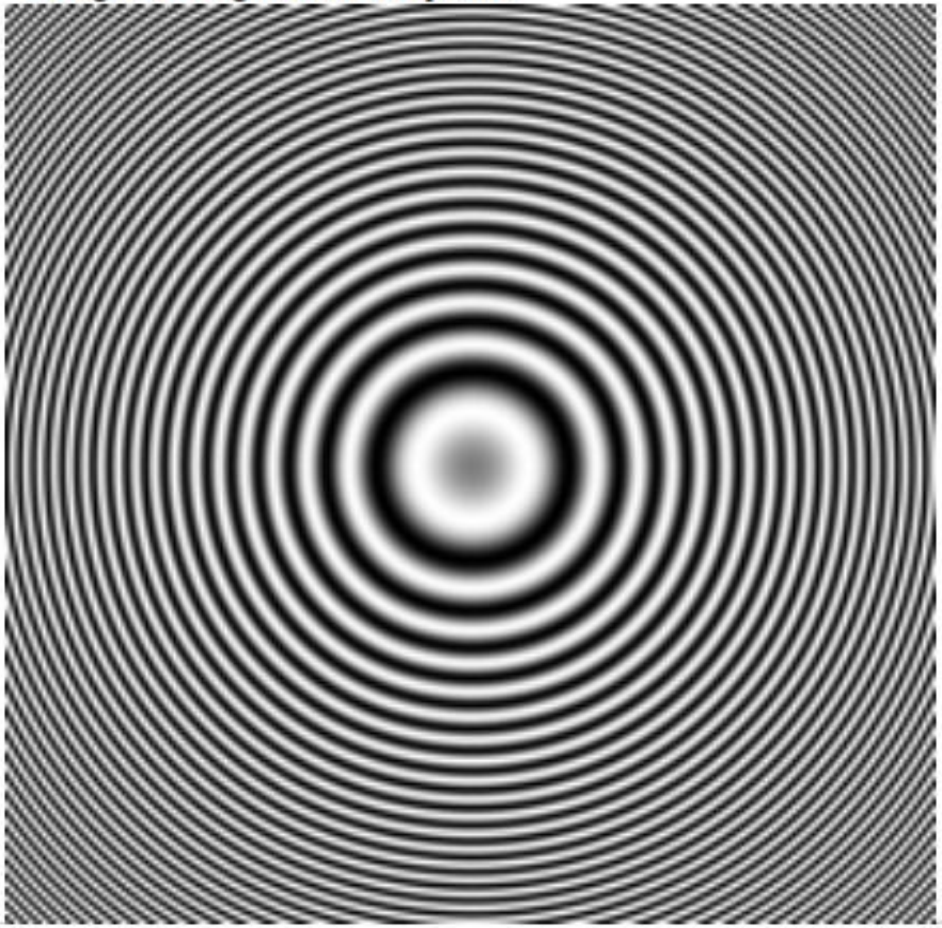
Aliasing Effect

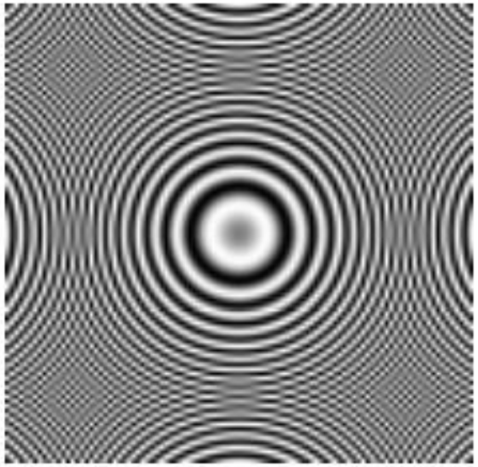
Example in 1 dimension



4. Image Sampling and Quantization

Original image: 200x200 pixels





Sampled image:
100x100 pixels

4. Image Sampling and Quantization

Aliasing Effect

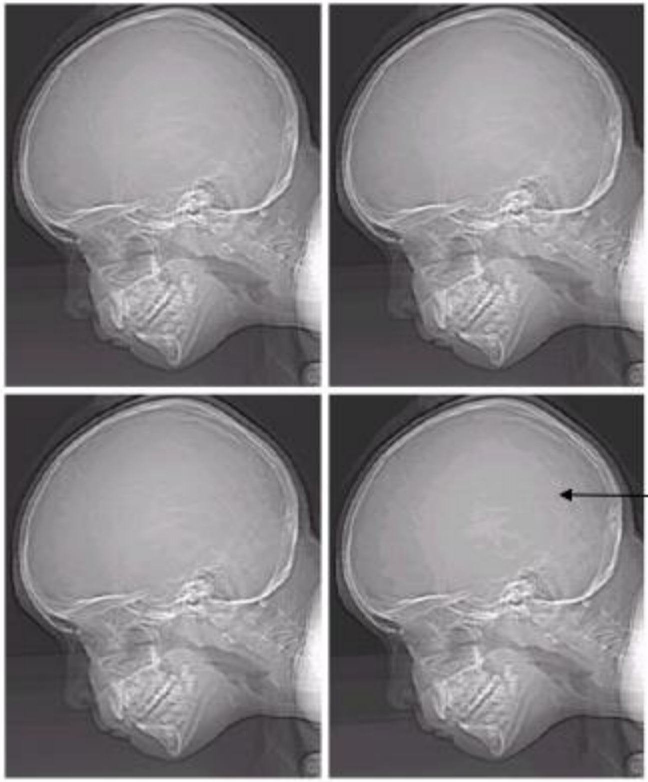
Original image: 622x756 pixels

205x250 pixels

"Moiré pattern"

4. Image Sampling and Quantization

Effects of Quantization

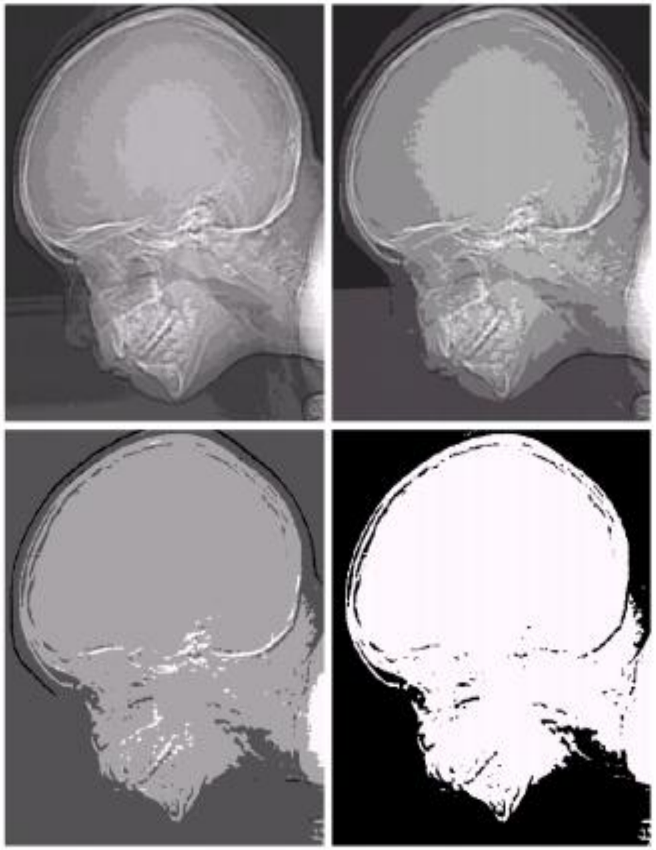


a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

4. Image Sampling and Quantization

Effects of Quantization



e f
g h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

5. Some Basic Relationships Between Pixels

5 Some Basic Relationships between Pixels

Given an image *f(x,y)* and pixels p or q

2.5.1 Neighbours of a pixel

- A pixel p at *(x,y)* has 4 horizontal and vertical *neighbours*, whose coordinates are:

  (x+1,y), (x-1,y), (x,y+1), (x,y-1) → set $N_4(p)$ (*4-neighbours* of p)

  NB: each is a unit distance from p, and some of these locations lie outside the image (borders)

- The 4 *diagonal neighbours* of p have coordinates:

  (x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1) → set $N_D(p)$

- $N_4(p) \cup N_D(p) = N_8(p)$: the set *of 8-neighbours* of p

0	1	1
0	1	0
0	0	1

5. Some Basic Relationships Between Pixels

Adjacency, Connectivity, Regions and Boundaries

Let V be a set of intensity values used to define adjacency

4-adjacency: p and q with values in V are 4-adjacent if $q \in N_4(p)$

8-adjacency: p and q with values in V are 8-adjacent if $q \in N_8(p)$

m-adjacency (mixed adjacency): p and q with values in V are m-adjacent if

$q \in N_4(p)$, *or*

$q \in N_D(p)$ *and* $N_4(p) \cap N_4(q)$ has no pixel with values from V

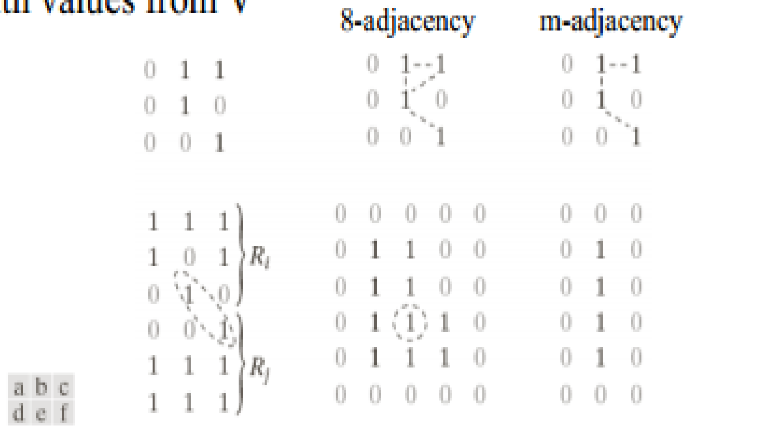


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

5. Some Basic Relationships Between Pixels

(*Digital*) *path* (or *curve*) from p (x,y) to q (s,t): sequence of *distinct* pixels with coordinates:

$(x_0,y_0), (x_1, y_1), \ldots, (x_n,y_n)$, where $(x_0,y_0) = (x,y)$, $(x_n,y_n) = (s,t)$ and

for i from 1 to n, (x_i, y_i) and (x_{i-1}, y_{i-1}) are *adjacent*.

n = *length* of the path

$(x_0,y_0) = (x_n,y_n)$ => *closed* path

4-,8-, or m-paths depending on the type of adjacency specified (cf figure)

Let S be a subset of pixels in an image

- P and q are *connected* in S if path exists between them consisting of pixels in S only
- For any p in S, set of pixels connected to it in S : *connected component* of S.
- If only one: S is a *connected set*

- R is a *region* of the image if R is a *connected set*
- Ri and Rj *adjacent* if Ri U Rj = connected set
- Regions not adjacent are *disjoint*

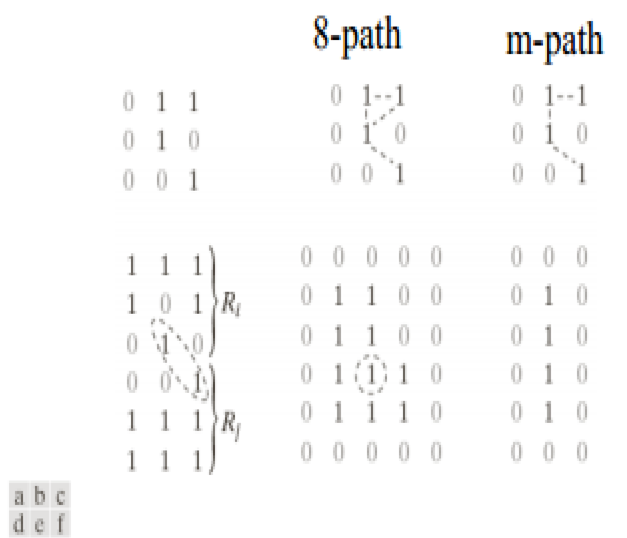


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjacency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

5. Some Basic Relationships Between Pixels

Distance Measures

For pixels p, q and r, with coord *(x,y), (s,t)* and *(v,w)* resp., D is a *distance function* or *metric* if:

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

$$D(p,q) = D(q,p)$$

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

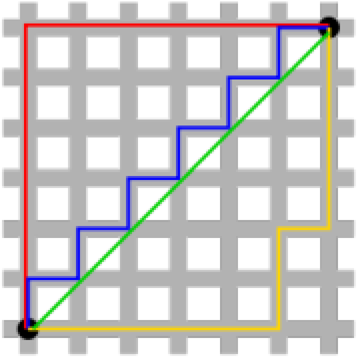
Euclidian distance between p and q:

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

D4 distance (city-block distance, or *Manhattan distance*):

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

Manhattan distance



http://en.wikipedia.org/wiki/Taxicab_geometry

D8 distance (*chessboard distance*, or Tchebychev distance):

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

6. Image Histogram

- The distribution of gray levels in an image convey some useful information on the image content.

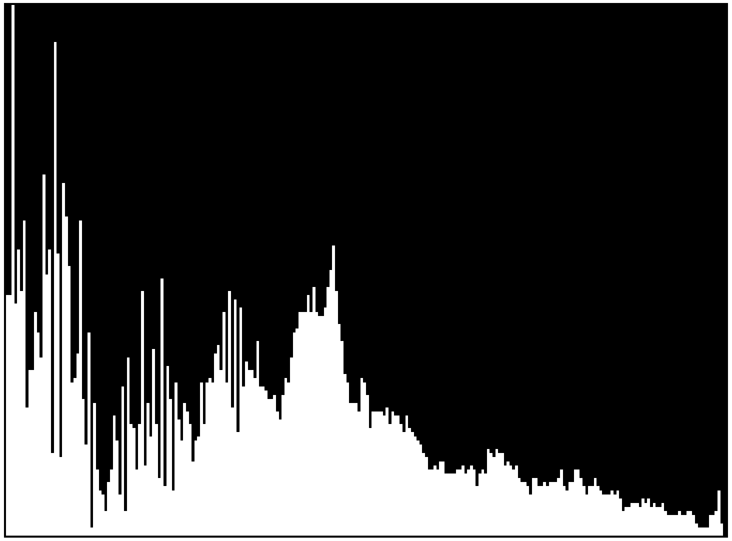
- For any image f of size mxn and Gray Level resolution k, the histogram of h is a discrete function defined on the set $\{0, 1, \ldots, 2^k-1\}$ of gray values such that h(i) is th number of pixels in the image f which have the gray value i.

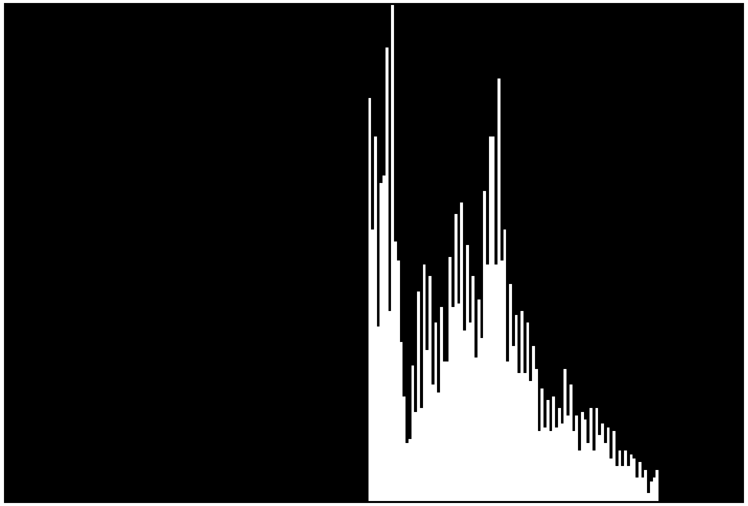
- It is customary to "normalise" a histogram by dividing h(i) by the total number of pixels in the image, i.e. use the probability distribution:

$$p(i) = h(i)/mn.$$

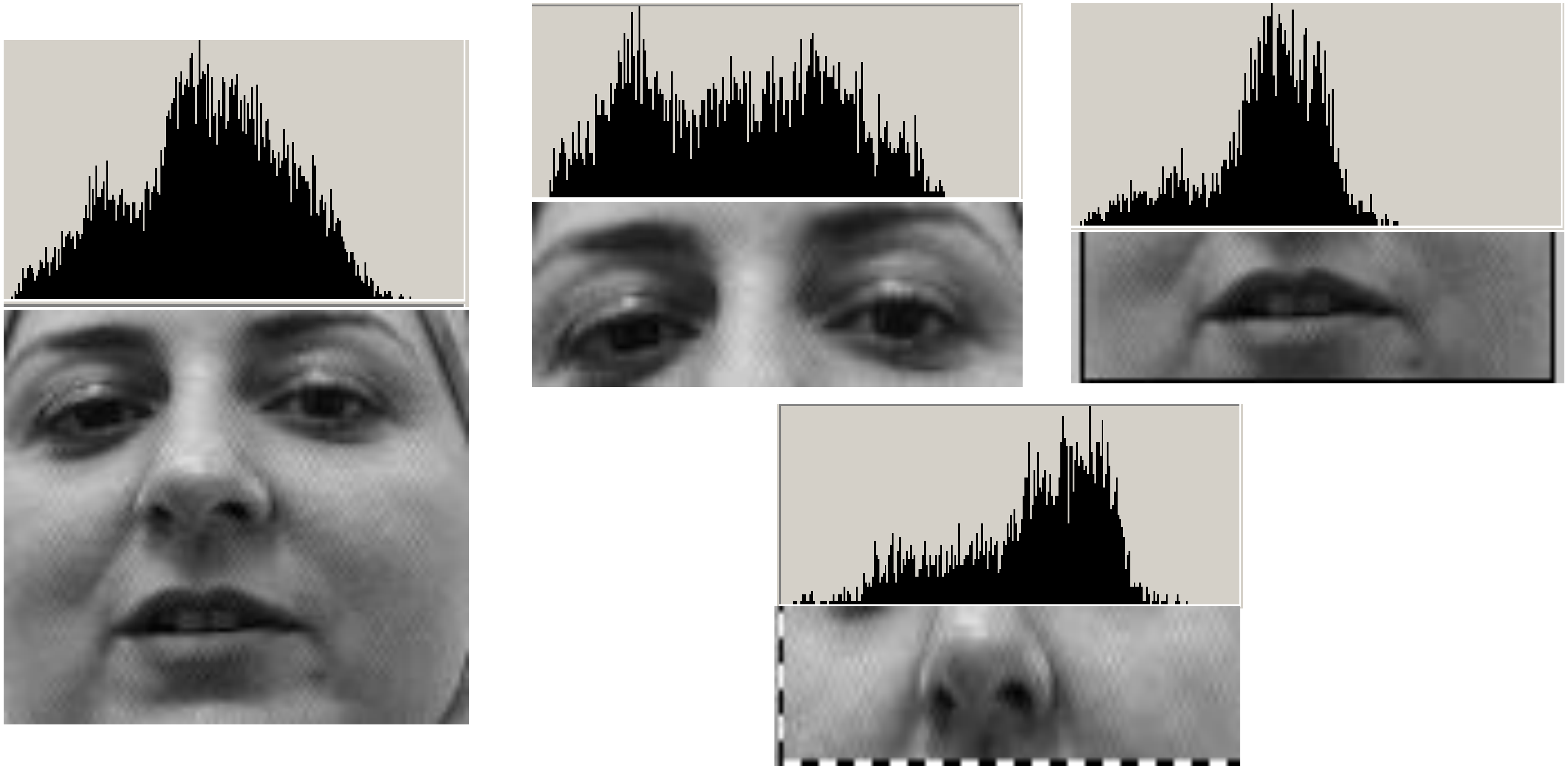
- Hstograms are used in numerous processing operations.

6. Histograms - Examples



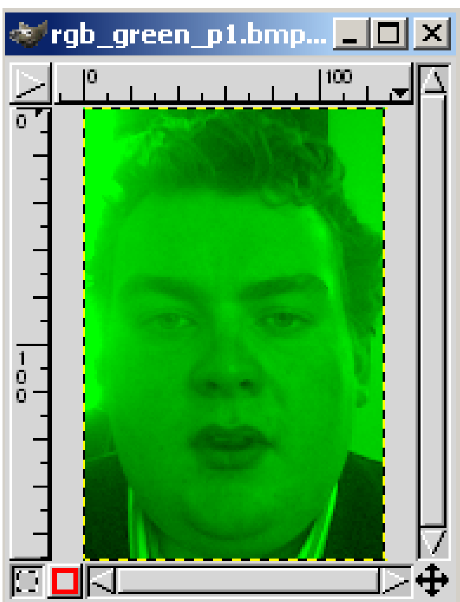


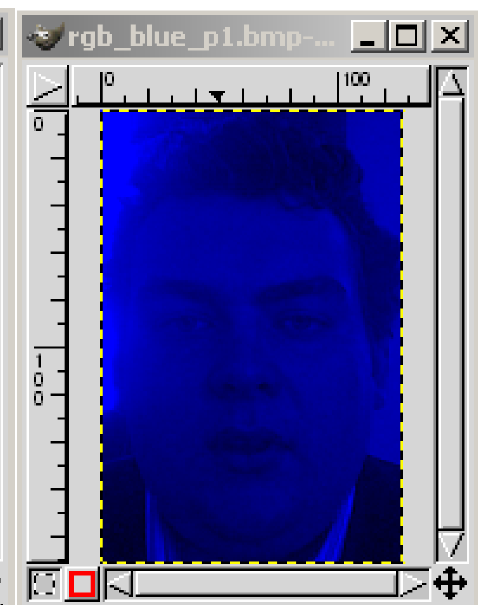
6. Local Vs. Global Histograms – Image Features



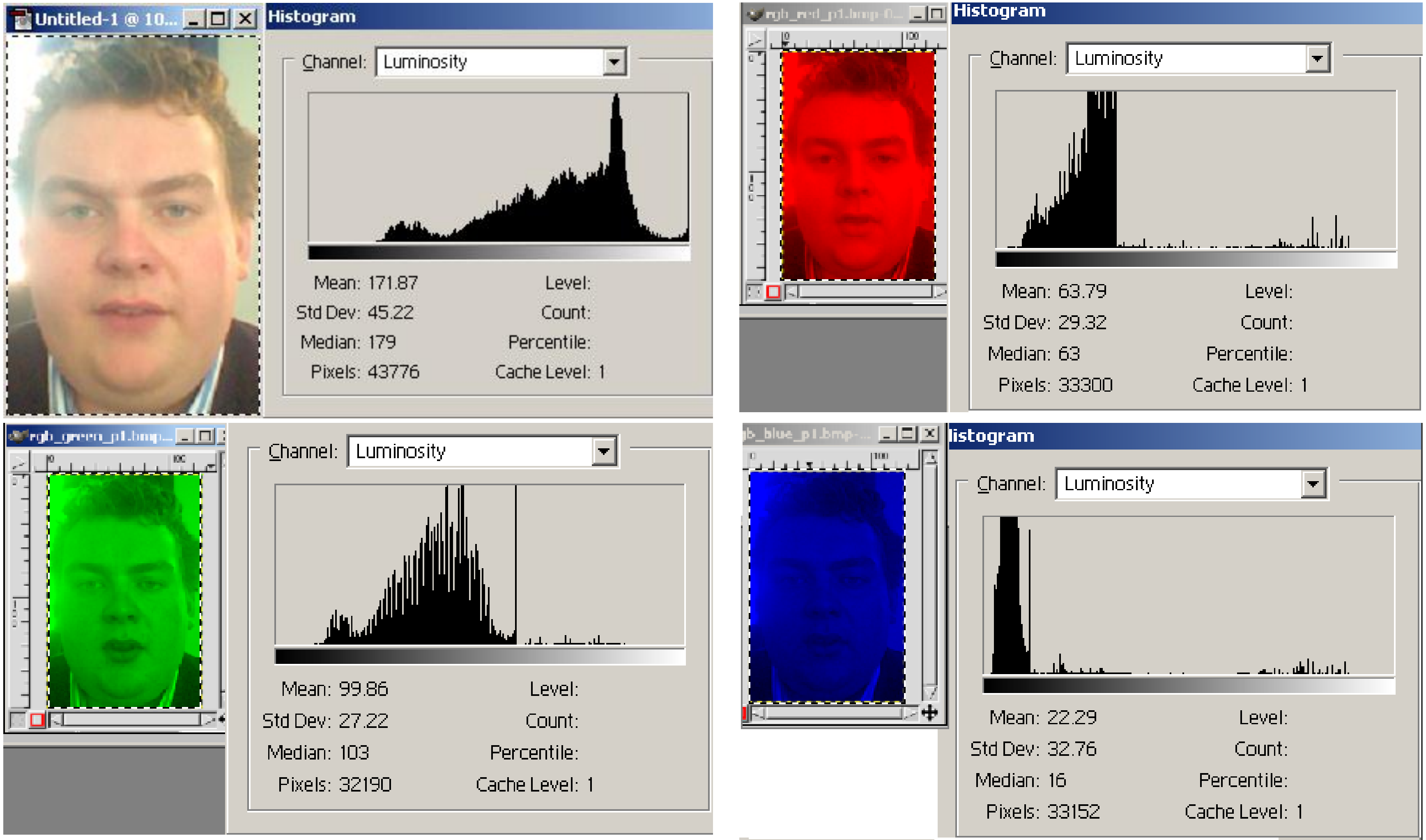
- Histograms for parts of an image provide useful tools for feature analysis.
- Local Histograms provide more information on image content than the global histogram.

7. Color Images





7. Color Images



7. Color Images

- Light reflected on an object and detected by a sensor is an *additive (linear) combination* of different wavelengths (i.e. Colours).

- Red, Green, and Blue are the *primary colors*. Other colors are a linear combination of R, G and B. **i.e** light color space is 3 dimentional with {R, G, B} as its base and every other colour can be expressed as:

$$a*R + b*G + d*B,$$

$$\text{where } 0 \leq a,b,c \leq 1 \quad \text{and} \quad a + b + c = 1.$$

- *RGB perfectly interpret of human vision*

8. Image files Format

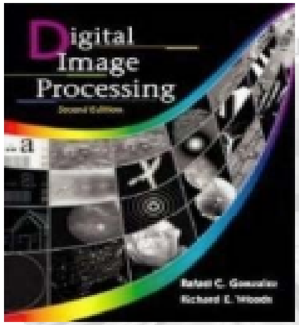
Image files consists of two parts:

- A *header* found at the start of the file and consisting of parameters regarding:
  - ✓ Number of rows (height)
  - ✓ Number of columns (width)
  - ✓ Number of bands (i.e. colors)
  - ✓ Number of bits per pixel (bpp)
  - ✓ File type
- Image *data* which lists all pixel values (vectors) on the first row, followed by 2nd row, and so on.

- Common image file formats are:
  - BIN, RAW
  - PPM,PBM,PGM
  - BMP
  - JPEG
  - TIFF
  - GIF
  - RAS
  - SGI
  - PNG
  - PICT, FPX
  - EPS
  - VIP

End of Chapter 2

Digital Image Processing
Digital Image Processing Using Matlab



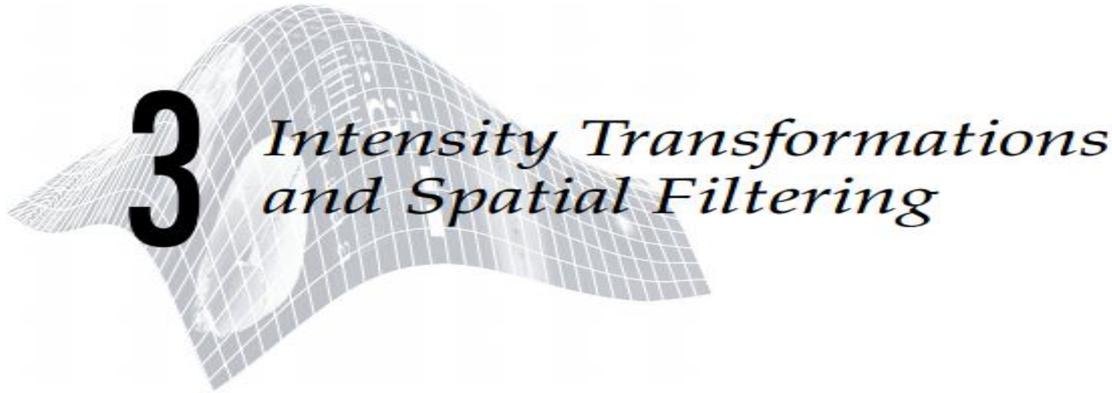
Digital Image Processing

Prepared by:

Dr. Ali J. Abboud

University of Diyala

2012-2013



Key Features of Chapter 3:

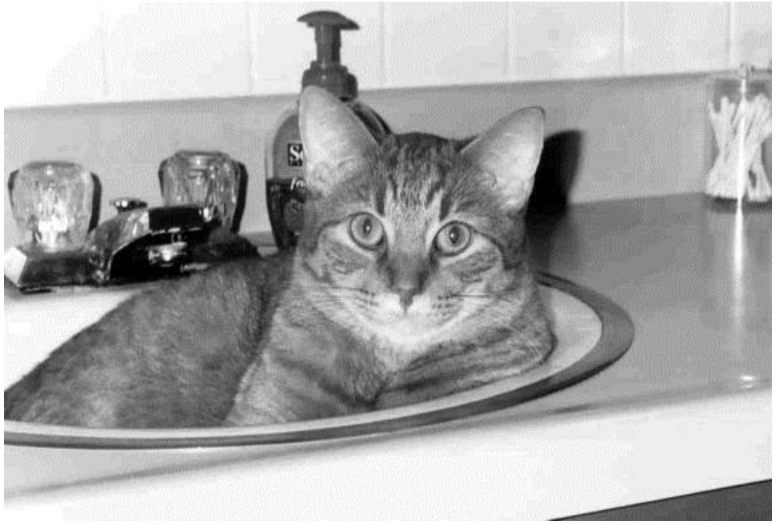
- ***Image Enhancement*.**
- ***Basic Gray Level Transformations*.**
- ***Histogram Processing*.**
- ***Smoothing Spatial Filters*.**
- ***Sharping Spatial Filters.***
- ***Combining Different Spatial enhancement Techniques.***

Introduction

- Image Applications require a variety of techniques that can be divided into two main categories:
  - Image Processing, and
  - Image Analysis
- Image processing techniques include:
  - Image Enhancement
  - Image Restoration
  - Image Compression (for storage or transmission)
- Image Analysis tasks include:
  - Feature Detection and Recognition
  - Image Classification
  - Image Indexing
- Image analysis do rely on Image pre-processing steps.

Image Enhancement - Examples

Poor contrast image

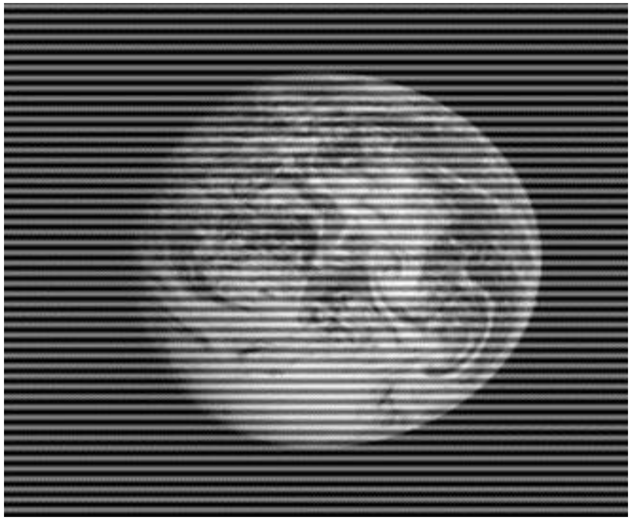


Enhanced image

Blurred image

Sharpened image

Image Restoration - Examples



Distorted image

Restored image

Geometrically distorted image

Restored image

Image Enhancement – Aims & Objectives

- ✓ ***Image enhancement*** aims to process an image so that the output image is more suitable than the original.
- ✓ Suitability is a application specific and enhancement is often a trial & error process.
- ✓ It either helps solve some computer imaging problems, or is used as an end in itself to improve image quality.
- ✓ Enhancement methods are either used as a preprocessing step to other imaging tasks, or as post-processing to create a more visually desirable image.
- ✓ Enhancement includes *improving contrast*, *sharpening*, *highlighting,* or *smoothing some features* for display and/or for further analysis.

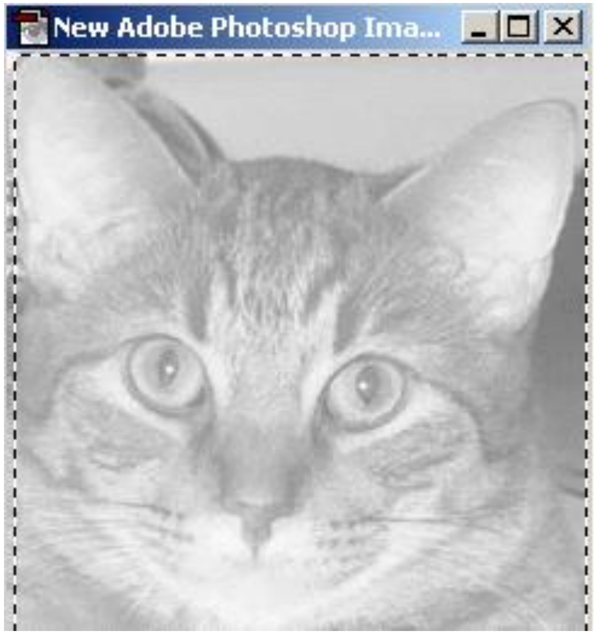
Image Enhancement in the spatial domain

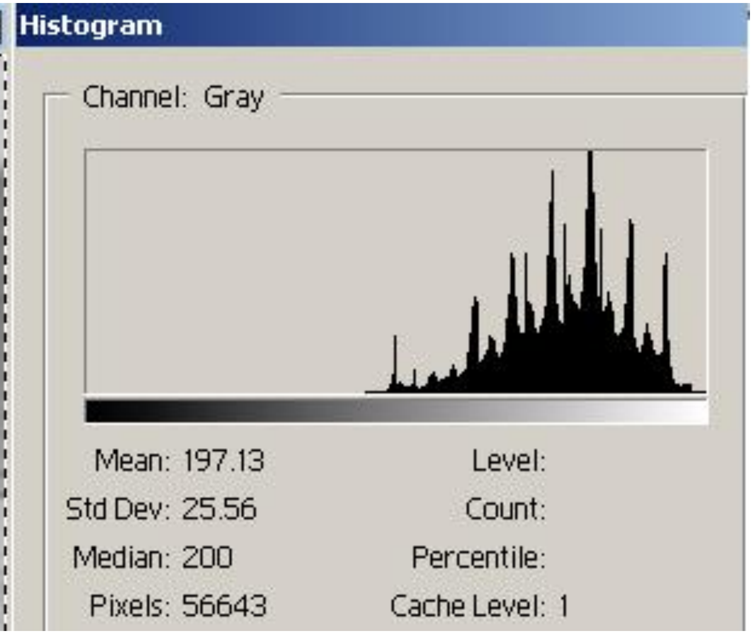
- Success of enhancement may be evaluated subjectively by viewers or automatically according to defined criterion.
- Image enhancement methods are classified as:
  - Enhancement in the Spatial Domain – using image transforms that manipulate the image by changing its pixel values or move them around.
  - Enhancement in the Frequency domain using image operators that manipulate the frequency information in the subbands which in turn have noticeable spatial effects.
- In this chapter, we are concerned with Spatial domain based enhancement.

Dynamic Range,Contrast and Brightness concepts

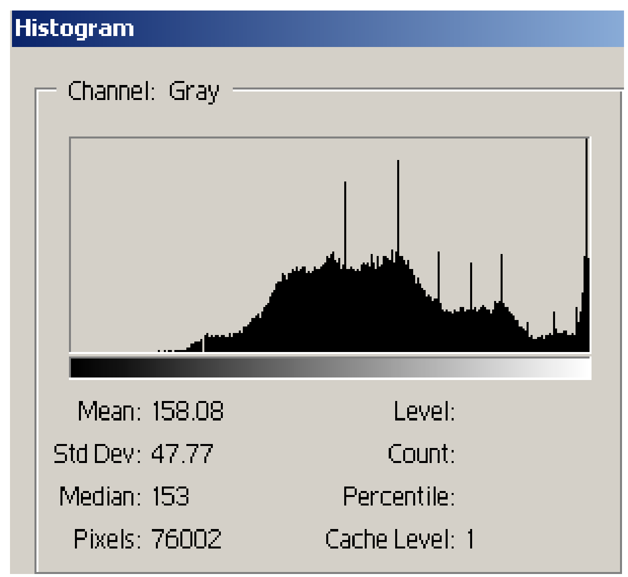
- ✓ The dynamic range of an image is the exact subset of grey values {0,1,…,L-1} that are present in the image. (In most cases L=255).
- ✓ Image histogram can be used to determine its dynamic range
- ✓ When the dynamic range contains significant proportion of the grey scale, then the image is said to have a high dynamic range and the image will have a good contrast.
- ✓ Low–contrast images can result from
  - ✓ *poor illumination*
  - ✓ *Lack of dynamic range in the imaging sensor*
  - ✓ *Wrong setting of lens aperture at the image capturing stage.*
- ✓ The most common enhancing procedures to deal with these problems are *Gray Level transform*

Examples





Poor contrast

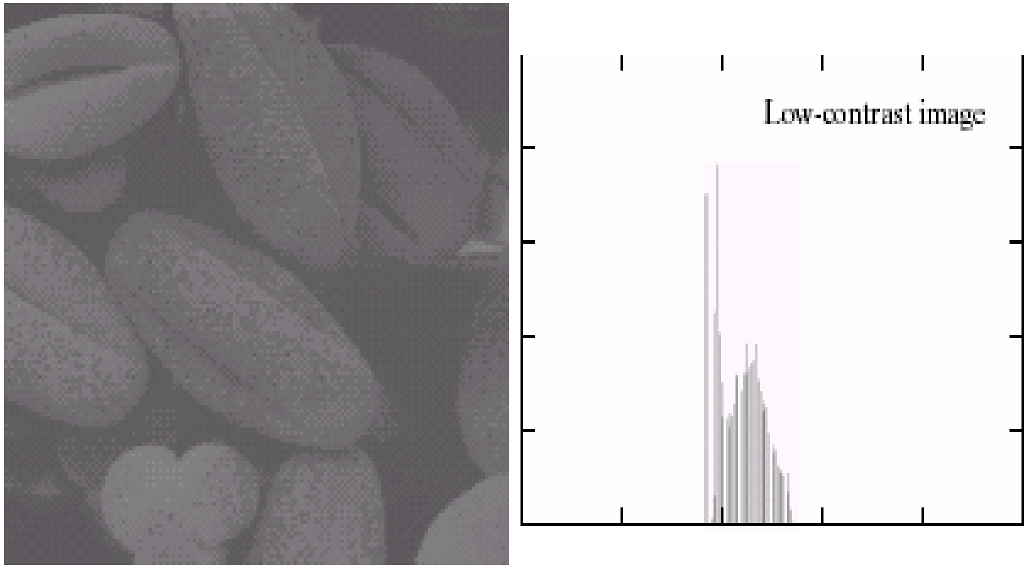


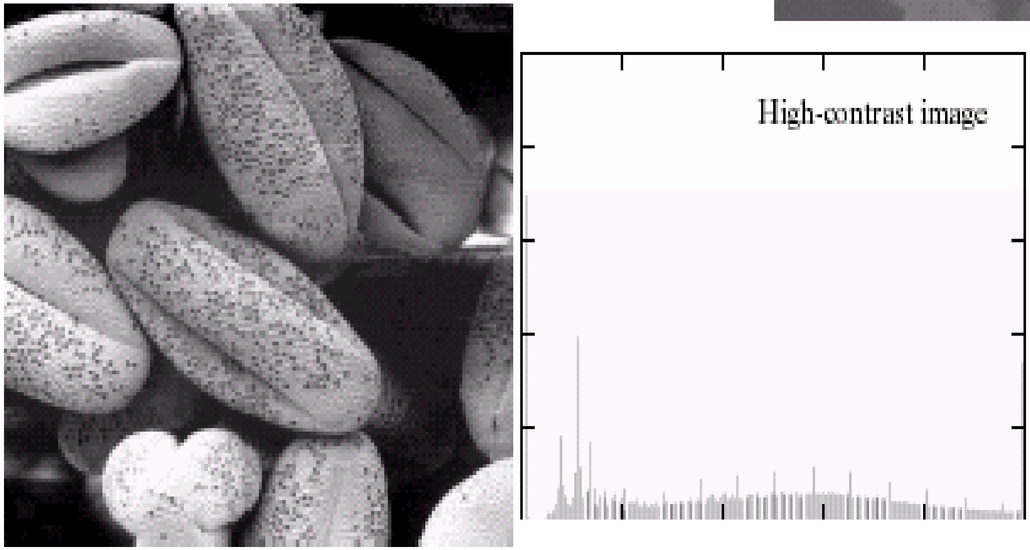
Reasonable but not perfect contrast

Image Histogram & Image contrast

Histograms also hold information about image contrast.

In low contrast images, histogram components are crammed in a narrow central part of the gray scale.

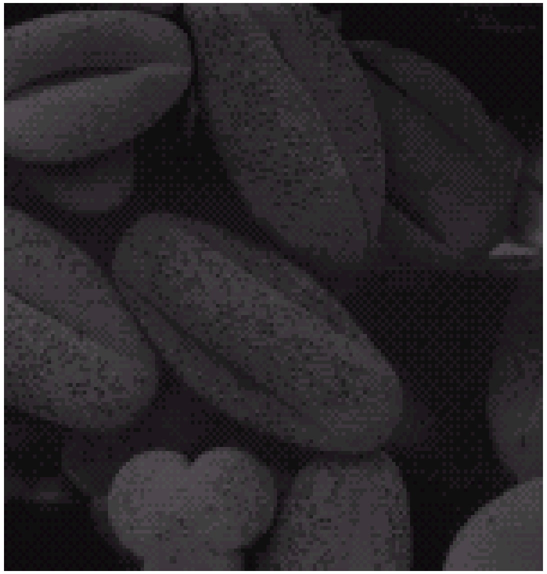


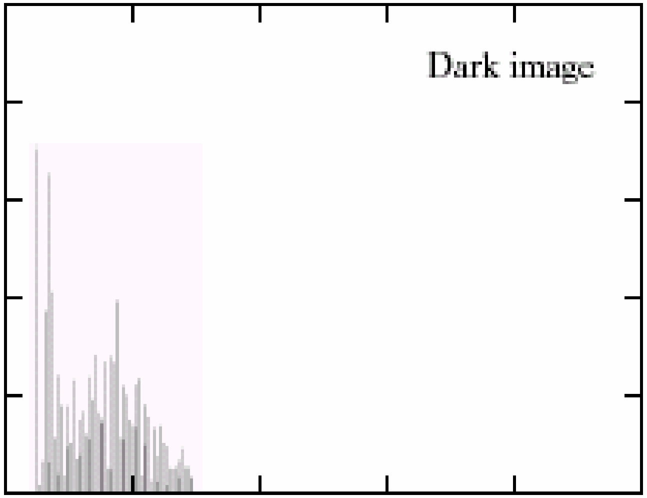


In high contrast images, the histogram occupy the entire gray scale (i.e. has high dynamic range) in a near uniform distribution.

Image Histogram & Image brightness

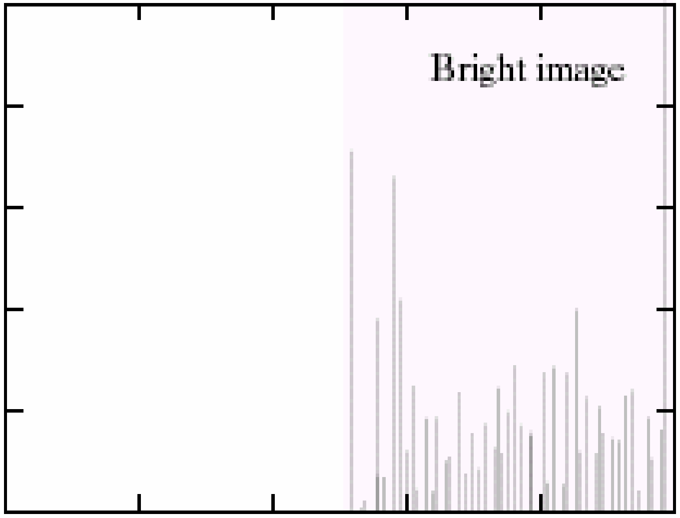
Image histogram holds information on brightness level





In dark images, histogram components are concentrated on the *low (dark) side* of the gray scale.

In bright images, histogram components are biased toward the *high (bright) side* of the gray scale.



MATLAB Implementation of Spatial domain transforms

- In MATLAB Grey level transforms can be implemented by a double nested loop using the transform formula.
- Filtering in the spatial domain can also be implemented by (triple) nested loops but one has to append the image on the boundaries of the image using some agreed scheme (by adding zeros, duplicating boundaries).
- MATLAB provides special functions for the most commonly used filters.

Image Operators in the Spatial domain

- An image operator in the spatial domain T applied on an image f(x,y) defines an output image:

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

  which is defined in terms of the pixel values in a neighbourhood centred at (x,y).

- Most commonly used neighbourhoods are squares or rectangles.
- The simplest form of T is when the neighbourhood consists of the pixel itself alone, i.e it depends on f(x,y) alone.

  In this case, T is a *Gray Level transform* which maps the set {0,1,…,L-1} of grey levels into itself, i.e. is a function:

  $$T: \{0,1,\ldots,L-1\} \longrightarrow \{0,1, \ldots, L-1\}.$$

- Larger size neighbourhood-based operators are referred to as *mask processing* or *filtering*.

Simple Gray Level transforms

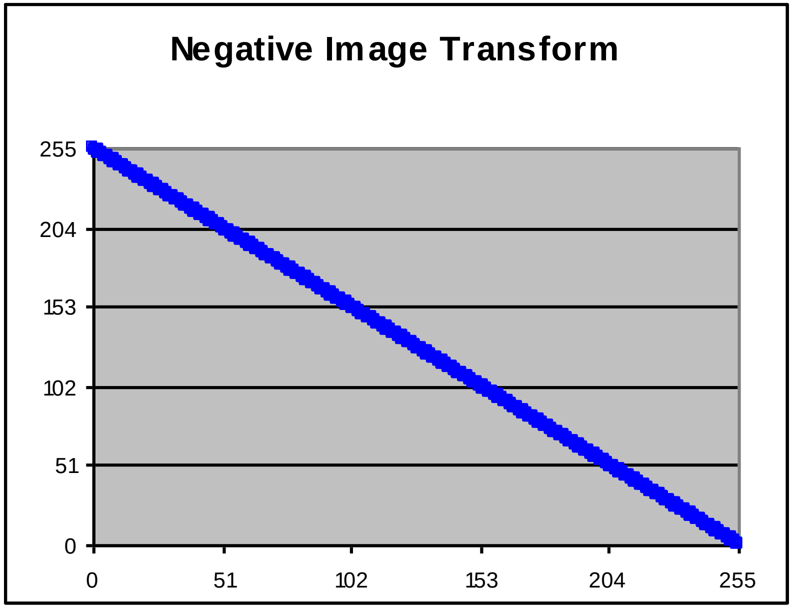
The most common type of grey level transforms are linear or piecewise (not necessarily continuous) linear functions

The Image Negative transform an image with gray level in the range {0,1,…,L-1} using the negative map:

$$T_{Neg}(i) = L - 1 - i.$$

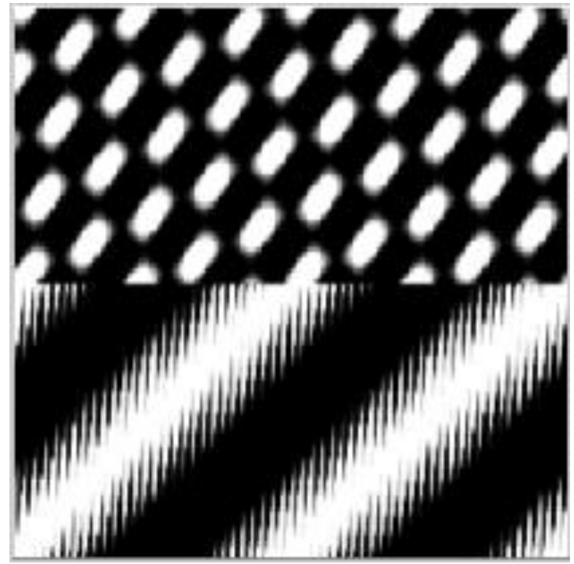
e.g. if $L = 2^8 = 256$ then

$$T_{Neg}(i) = 255 - i.$$

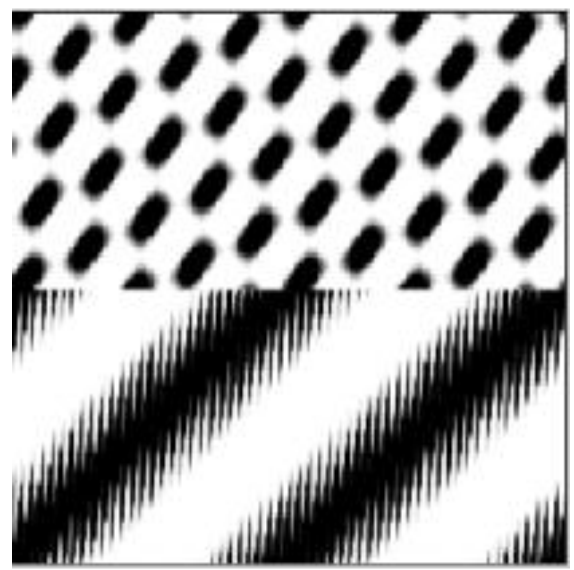


Negative transform in MATLAB

```
clear all
c=imread('Artificial.bmp');
figure; imshow(c);
[m n]=size(c);
for i=1:1:m
    for j=1:1:n
        f(i,j)=255-c(i,j);
    end
end
figure; imshow(f);
imwrite(f,'Artificial-neg.bmp')
```

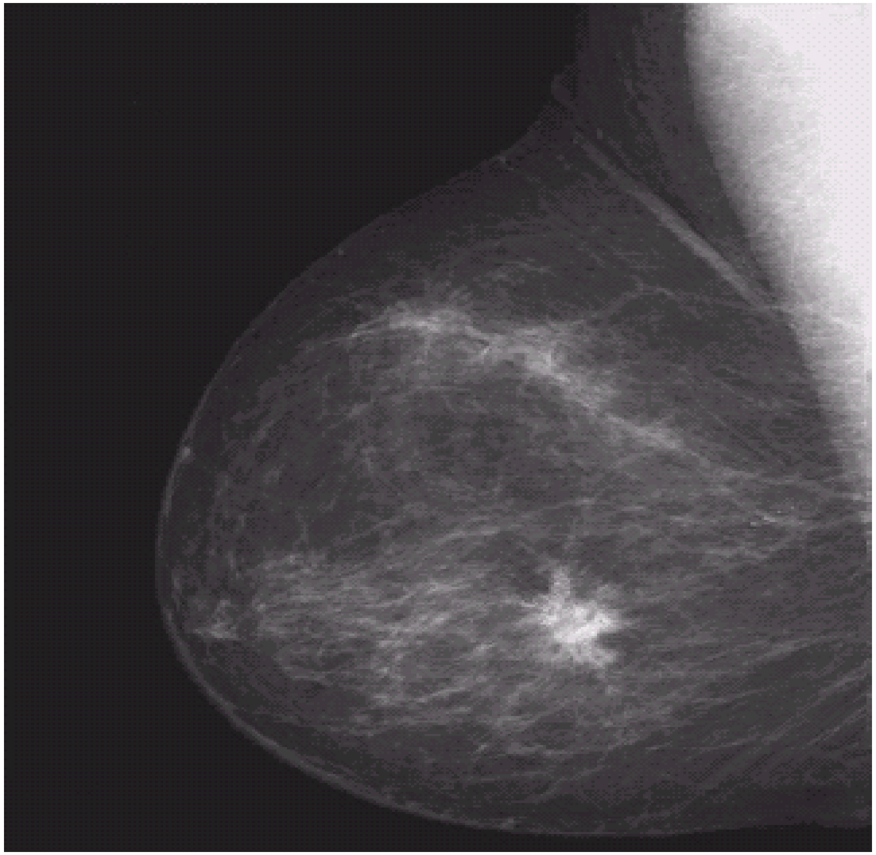


Artificial.bmp

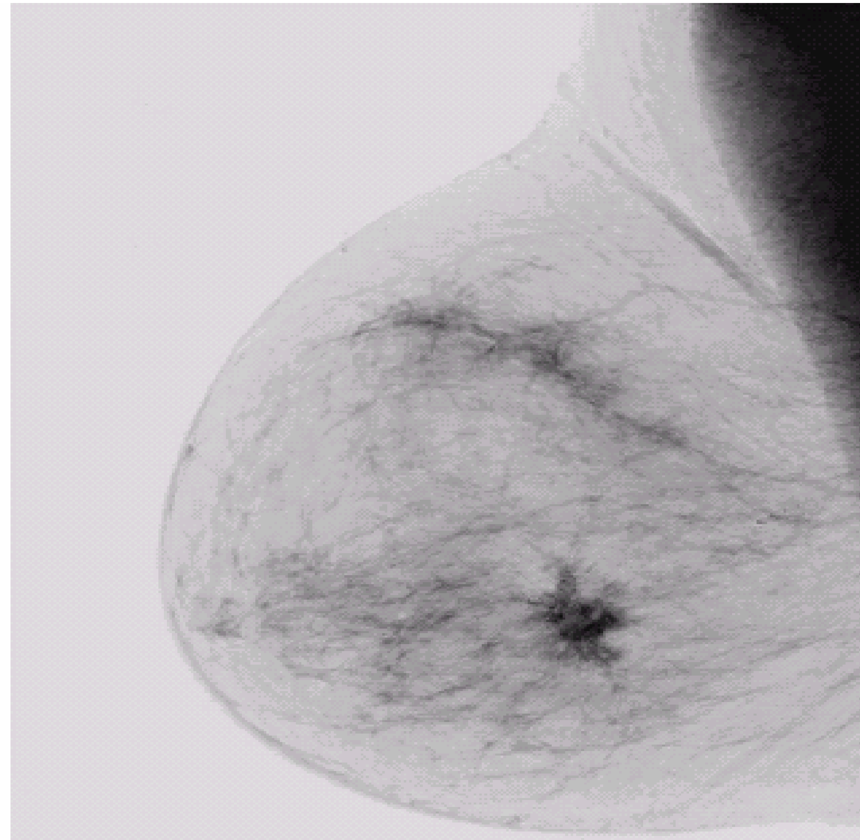


Artificial-neg.bmp

Example – Negative of an image



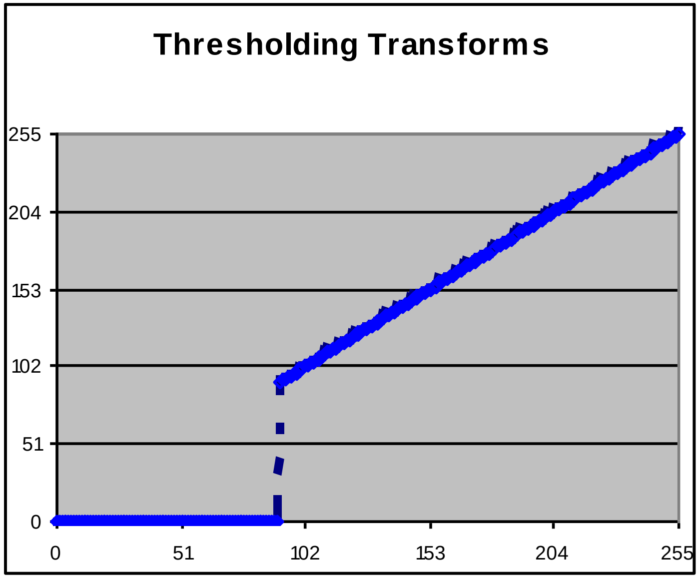
Original digital mammogram.



Negative image

Piecewise Linear Gray Level transforms

- For any $0 < t < 255$ the **threshold** transform Thr_t is defined for each i by:

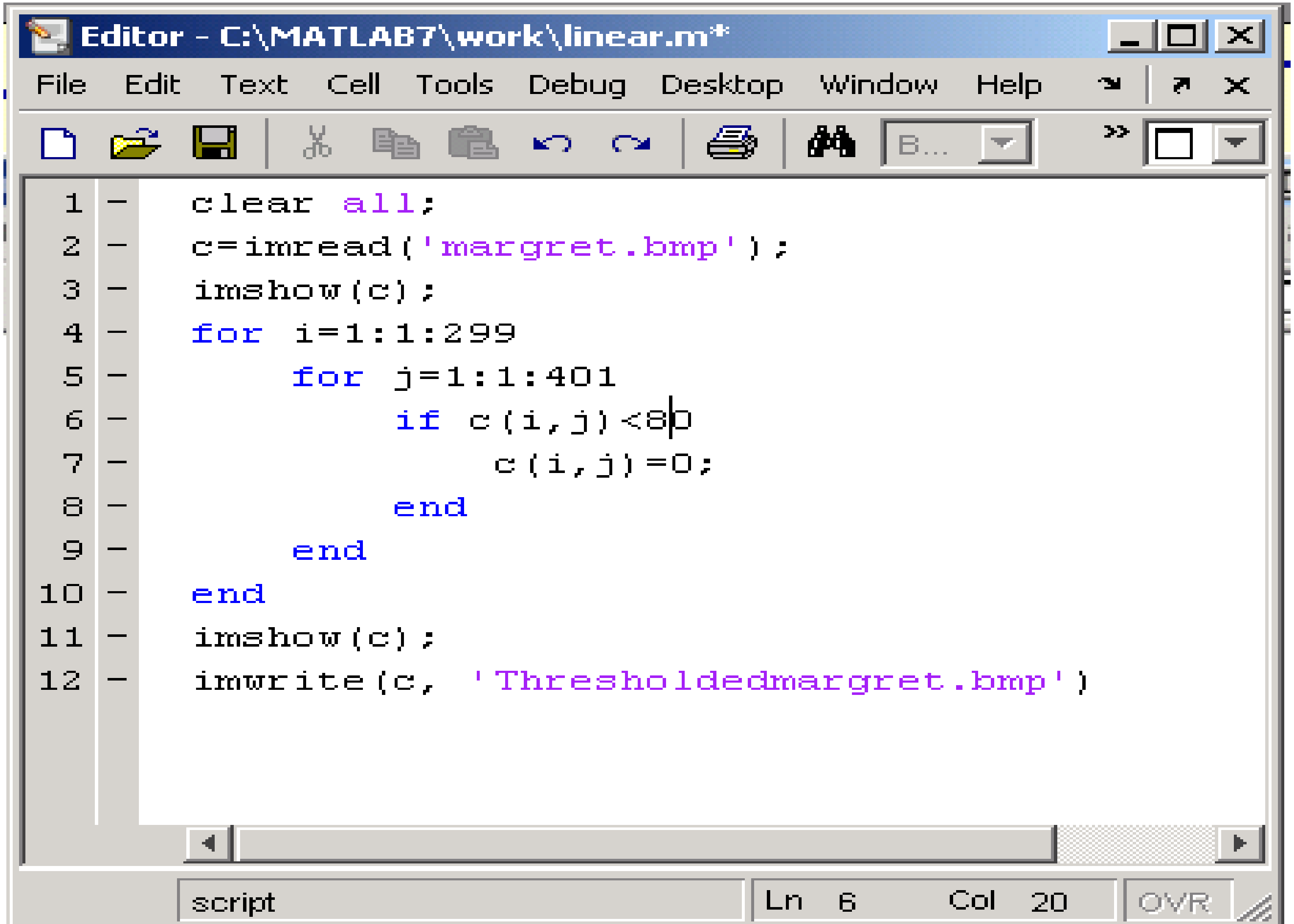


$$Thr_t(i) = \begin{cases} 0 & \text{If } i < t \\ i & \text{otherwise} \end{cases}$$

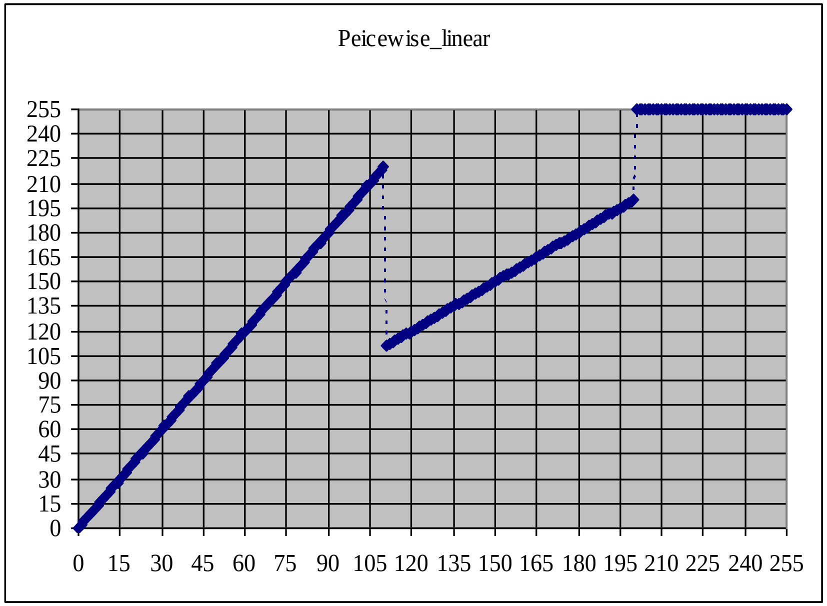
Thr_{80} →

The Threshold Transform in MATLAB

In MATLAB, Grey level transforms can be implemented by nested loop.

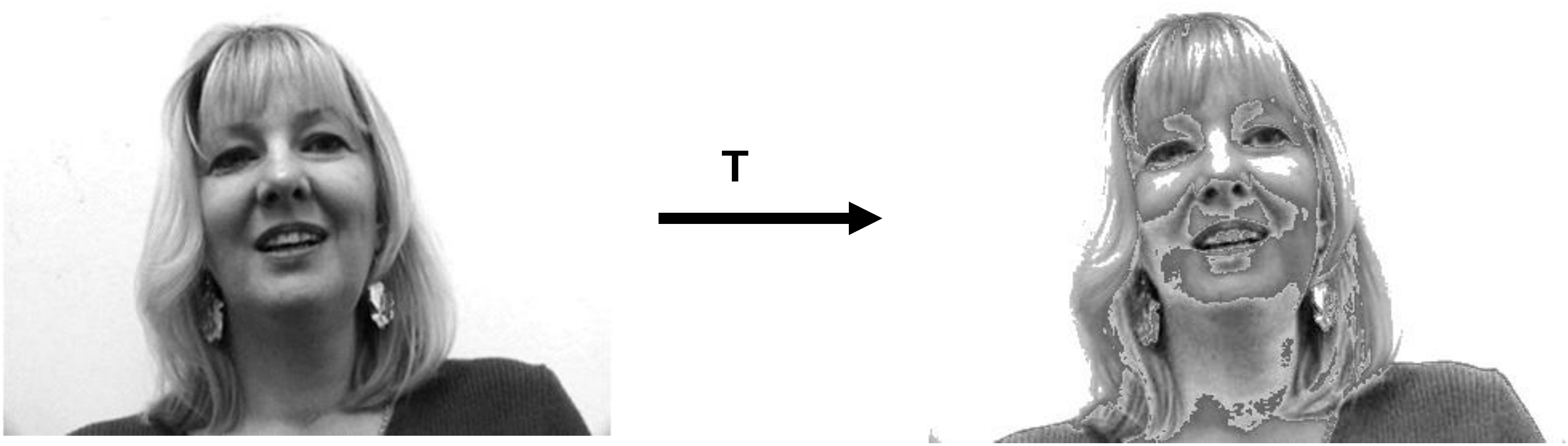


A Piecewise Linear Gray Level transform

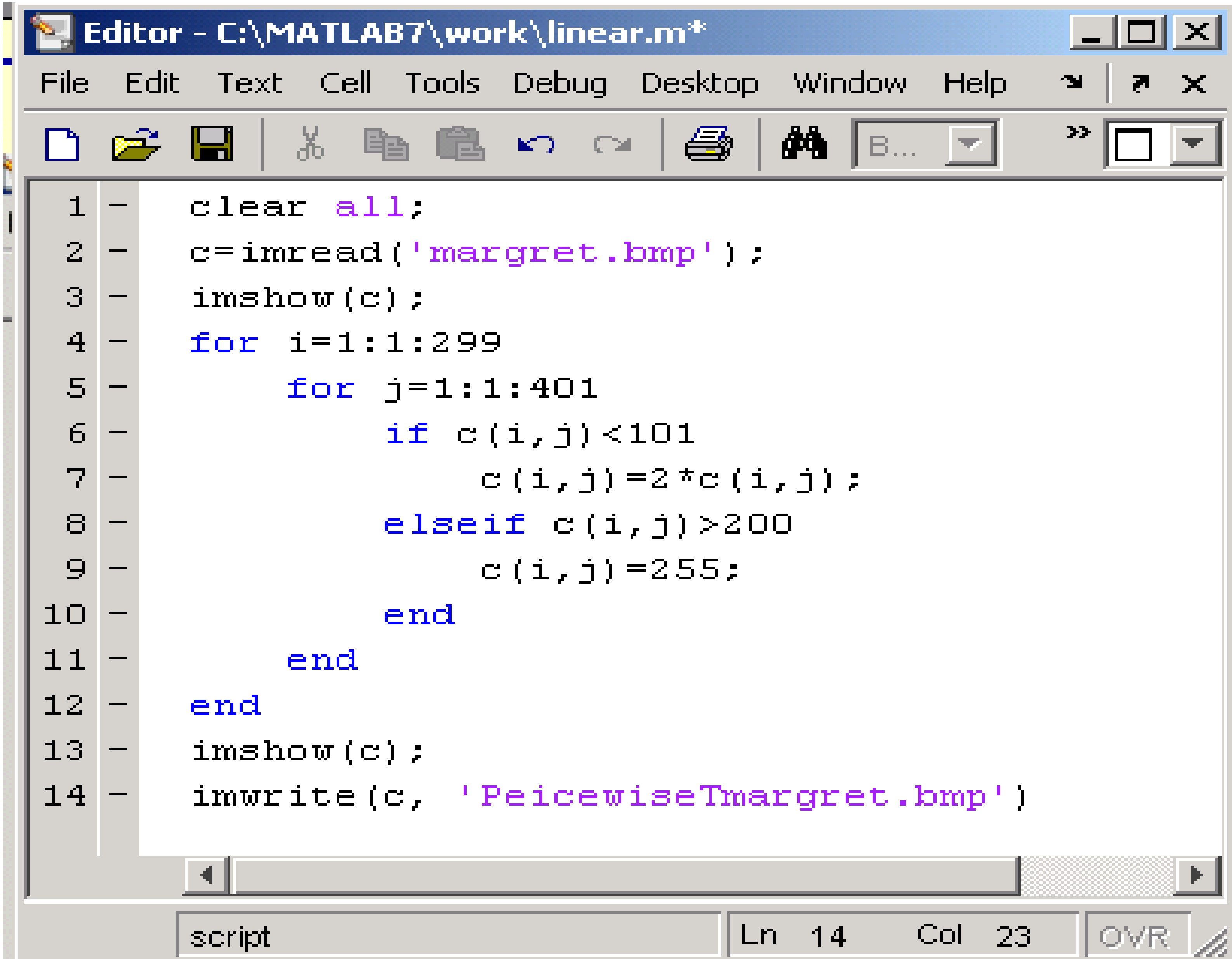


This chart represents the transform *T which* is defined for each i by:

$$T(i) = \begin{cases} 2*i & \text{If } i \le 110 \\ i & \text{If } 110 < i \le 200 \\ 255 & \text{If } i > 200. \end{cases}$$

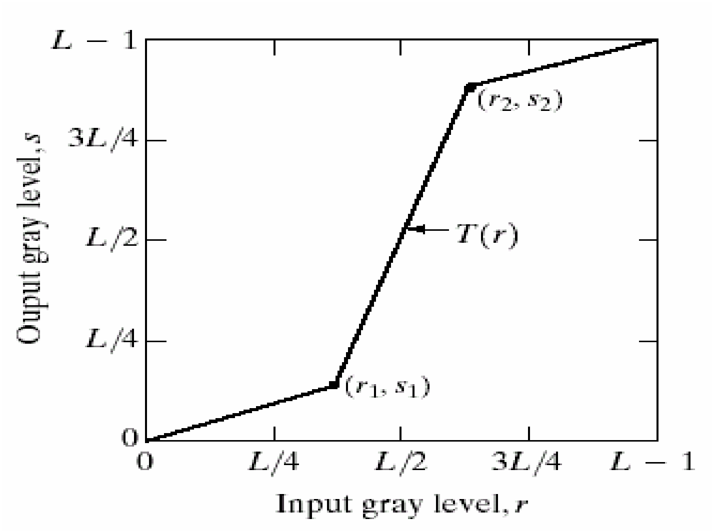


The above Piecewise Transform in MATLAB

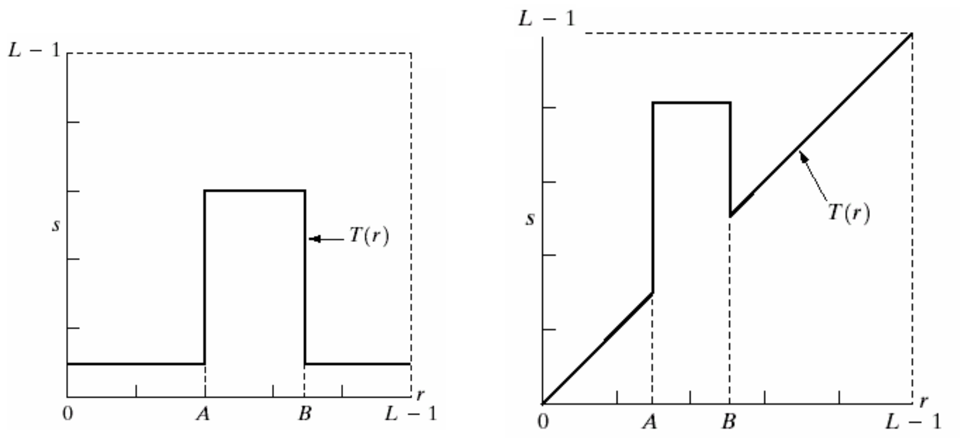


Special Grey Level transforms

- **Grey level Stretching aims to increase the dynamic range of an image. It transforms the grey levels in the range {0,1,…,L-1} by a piecewise linear function.**



- **Gray level Slicing aims to highlight a specific range [A…B] of Grey levels. It simply maps all Grey values in the chosen range to a constant. Other values are either mapped to another constant or left unchanged**



Gray level stretching - Example

Original image

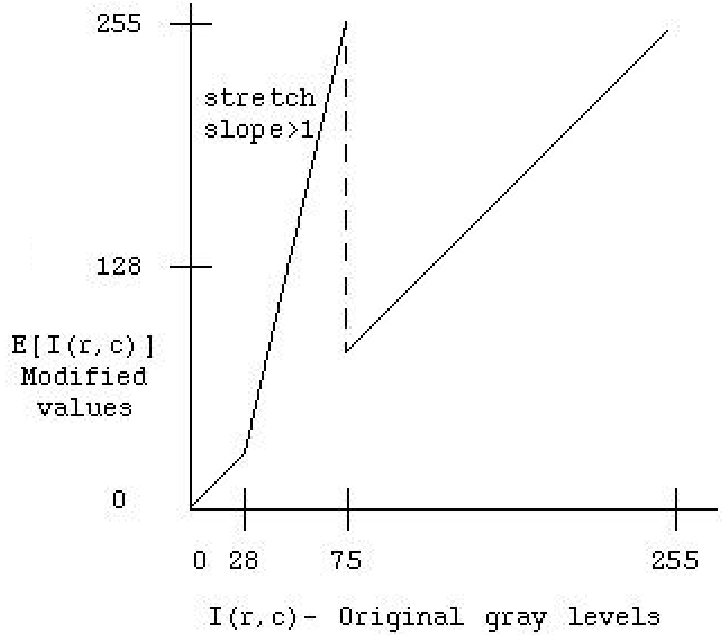


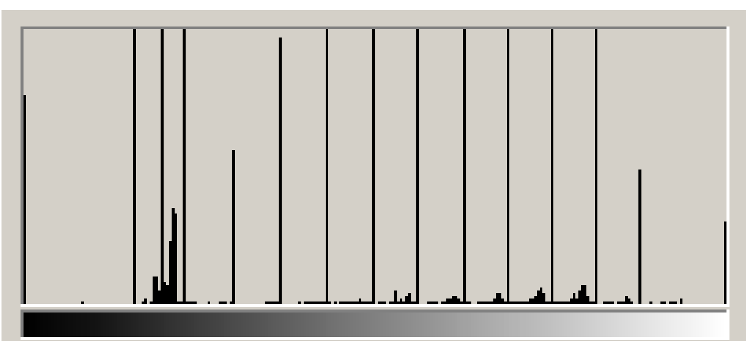
Image after modification

$$T(i) = \begin{cases} (227i - 5040)/47 & \text{if } 28 \le i \le 75 \\ i & \text{otherwise.} \end{cases}$$

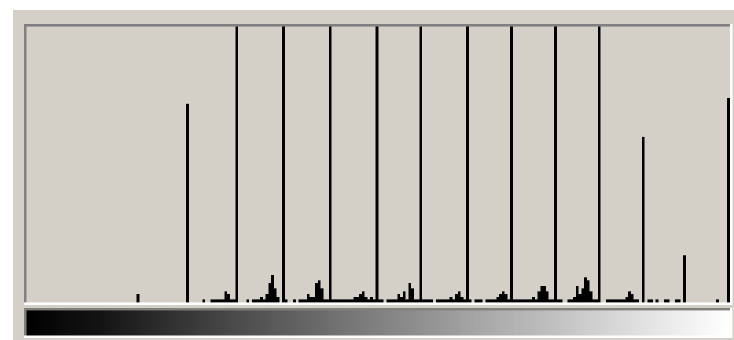
Grey-level stretching

Original image

Image after modification

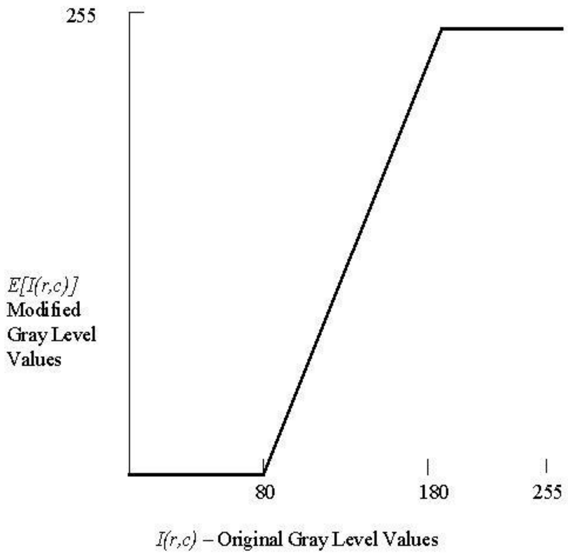


Histogram Before



Histogram After

Gray-level Stretching with Clipping at Both Ends



This operator is implemented by the function:

$$S(i) = \begin{cases} 0 & \text{if } i < 80 \\ 255(i - 80)/100 & \text{if } 80 \leq i \leq 180 \\ 255 & \text{if } i > 180 \end{cases}$$

Original image

S

Modified image with stretched gray levels

Gray-level Stretching with Clipping at Both Ends

Original image

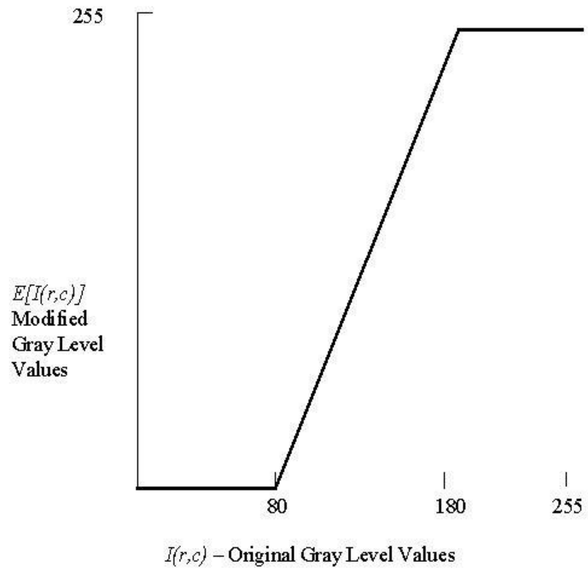
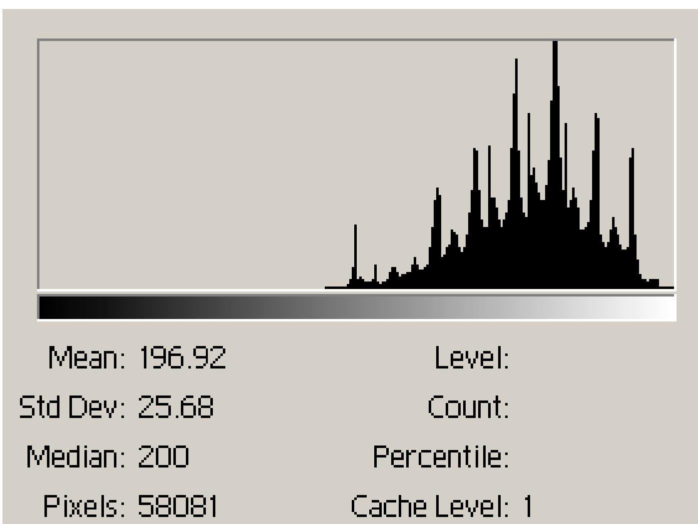
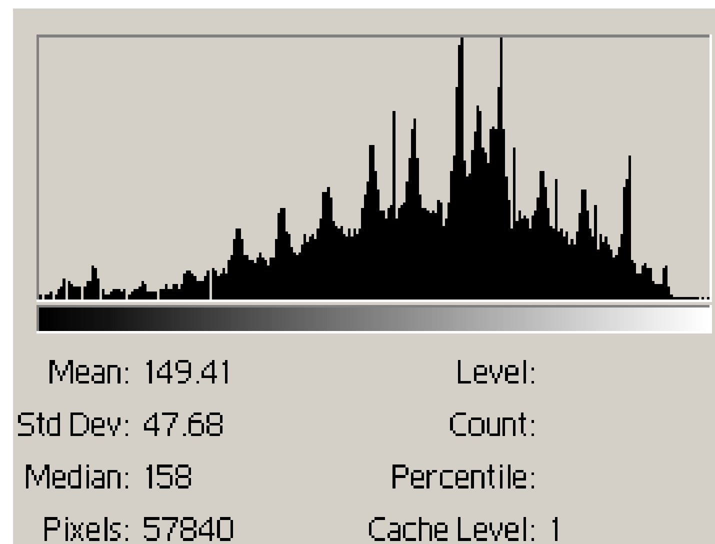


Image after gray levels stretch



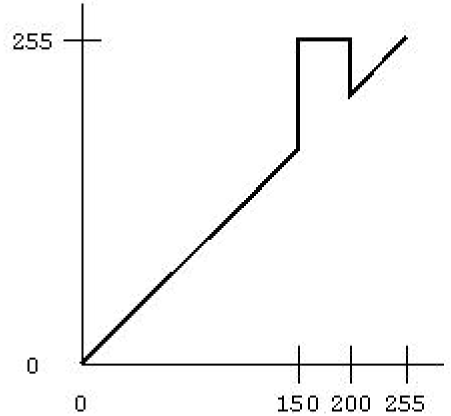
Histogram Before

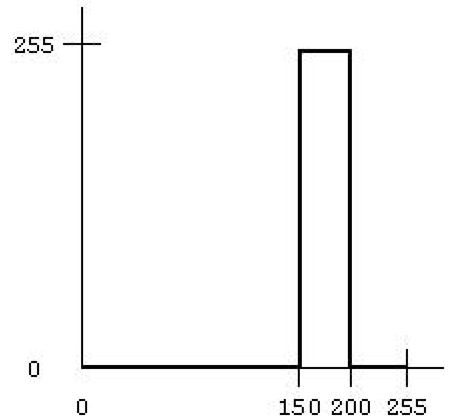


Histogram after

Gray Level Slicing - Examples

Original image





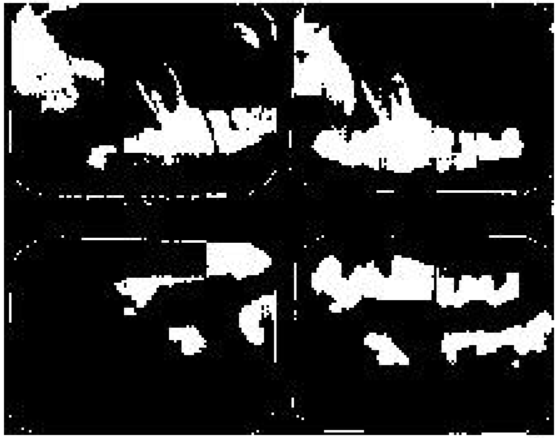


Image sliced to emphasize gray values from 150 to 200; background changed to black.

Non-linear Basic Gray Level transforms

- **The Log transform is based on a function of the form:**

  $$Log_c(i) == c\,Log(1+i)$$

  for a constant c.

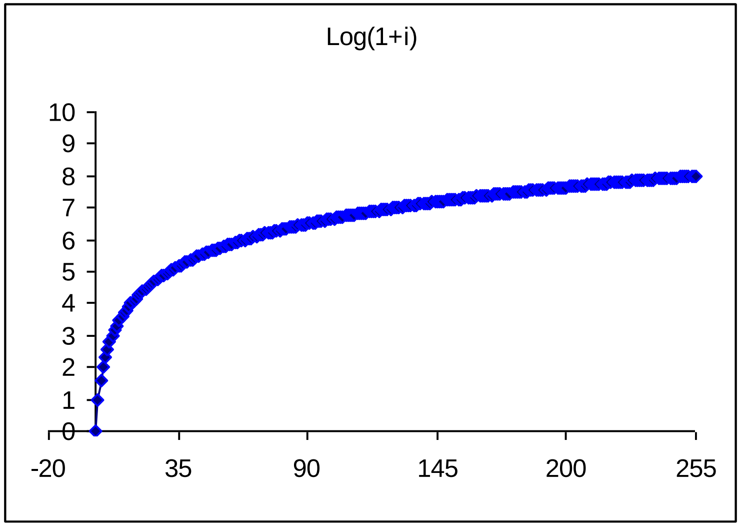
  Depending the value of c, this function darken the pixels in a non-uniform way.

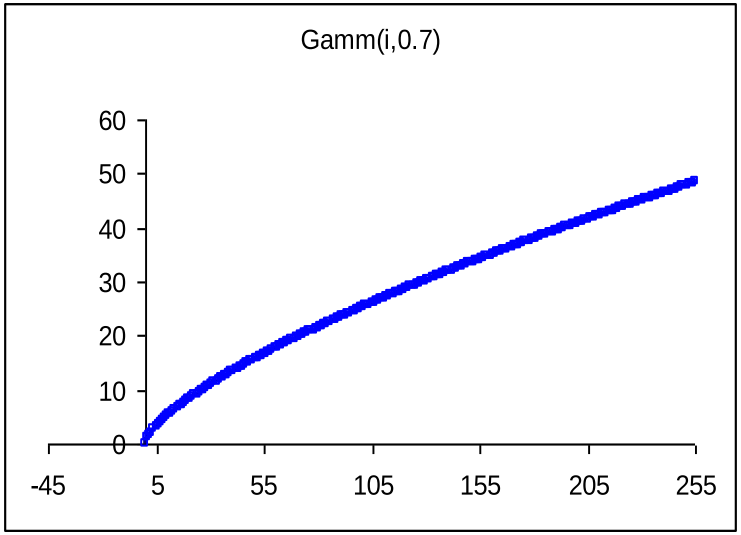
- **The Gamma transform is based on a function of the form:**

  $$Gamma_{(c,\gamma)}(i) = c * i^{\gamma}$$

  for constants c and γ.

  The effect of this transform depends on the value of c and γ.





These transforms effect different images in different ways.

The Effect of different Gamma transforms

Original MRI image

Gamma transform, c =1, γ =0.6

Gamma transform, c =1, γ =0.4

This example indicates that different γ values have different effect

Another Example

Original Aerial image

Gamma transform, c =1, γ =3.0

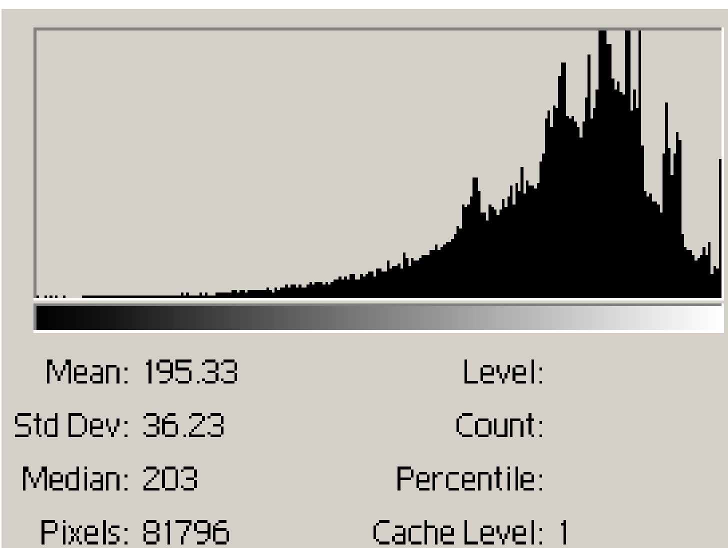
Gamma transform, c =1, γ =4.0

Gamma transform, c =1, γ =5.0

This example also link between effects and γ values. But the 2 examples, show that the characteristics of the input image results in different effects.

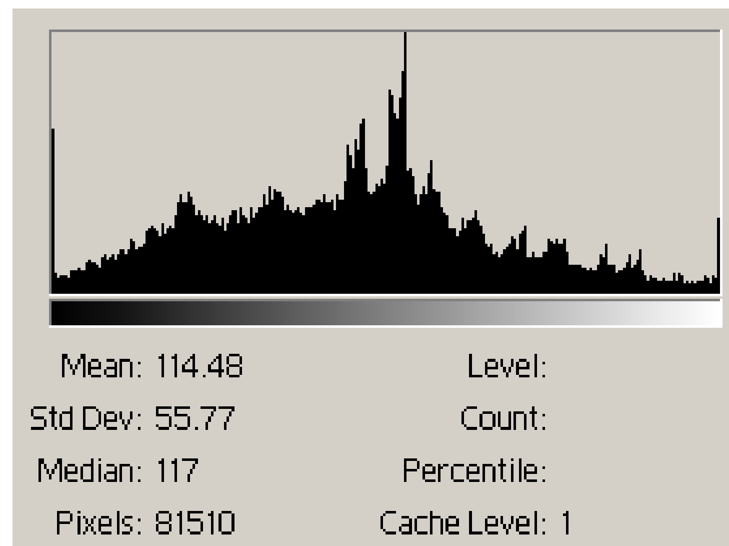
Gamma Transform - Example

Original Aerial image



Gamma Transform, c =1, γ =4.0

Transformed image



Histogram-based Gray Level Transforms

- ✓ **Image histograms provide statistical information that are useful for many image processing tasks such as enhancement, compression and segmentation.**

  Any gray-level transform, including the ones discussed in the last chapter, changes the input image histogram in way that depended on the transform parameter(s).

  Question:

  Is it possible to design gray-level transforms that manipulate image histograms in specified ways?

- ✓ **Filtering (i.e. image operators that change a pixel value in terms of a subset of a neighboring pixels) can also be used to enhance images.**

Enhancement through Histogram Manipulation

- ***Histogram manipulation*** **aims to determine a gray-level transform that produces an enhanced image that have a histogram with desired properties.**
- **The form of these gray-level transforms depend on the nature of the histogram of the input image and desired histogram.**
- **Desired properties include having a near uniformly distributed histograms or having a histogram that nearly match that of a reference (i.e. template) image.**
- **For simplicity, we normalize the gray levels r so that $0 \le r \le 1$ rather than being in the set {0,1, …, L-1}.**

  The gray level transforms are assumed to be based on an onto monotonically increasing continuous functions:

$$T : [0,1] \rightarrow [0,1]: r \mapsto s = T(r).$$

These conditions on T ensures that T has an inverse function.

Histogram Equalisation

- **Due to the randomness of light sources and sensor position among other factors we assume that gray levels in an Image is a random variable with probability density function (pdf) at gray level r being the expected proportion of pixels that have r gray value.**

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

- **This works for continuous pdf's, and for a discrete set {0,1,…,L-1} of gray levels it translates to:**

$$s = T(r) = \sum_{i=o}^{r} \frac{Freq(i)}{N}.$$

Where Freq(i) is the number of pixels in the image that have Gray value I, and N is the image size

Pseudo Code for Histogram Equalisation

Step 1: Scan the image to calculate the Freq [0..L-1], i.e. histogram

Step 2: From the Freq [ ] array compute the cumulative frequency array Cum_freq [0...L-1]:

```
{  Cum-freq[0] =Freq[0];
   for i=1 to L-1
       Cum_freq[i] =Cum_freq[i-1]+Freq[i];
}
```

Step 3: Determine the HE transformation lookup table:

```
for i=1 to L-1
    { j= round(Cum_freq[i]*(L-1)/N);
      T[i] =j;
      inv_T[j] = i;
    }
```

Step 4: Transform the image using the lookup table T.

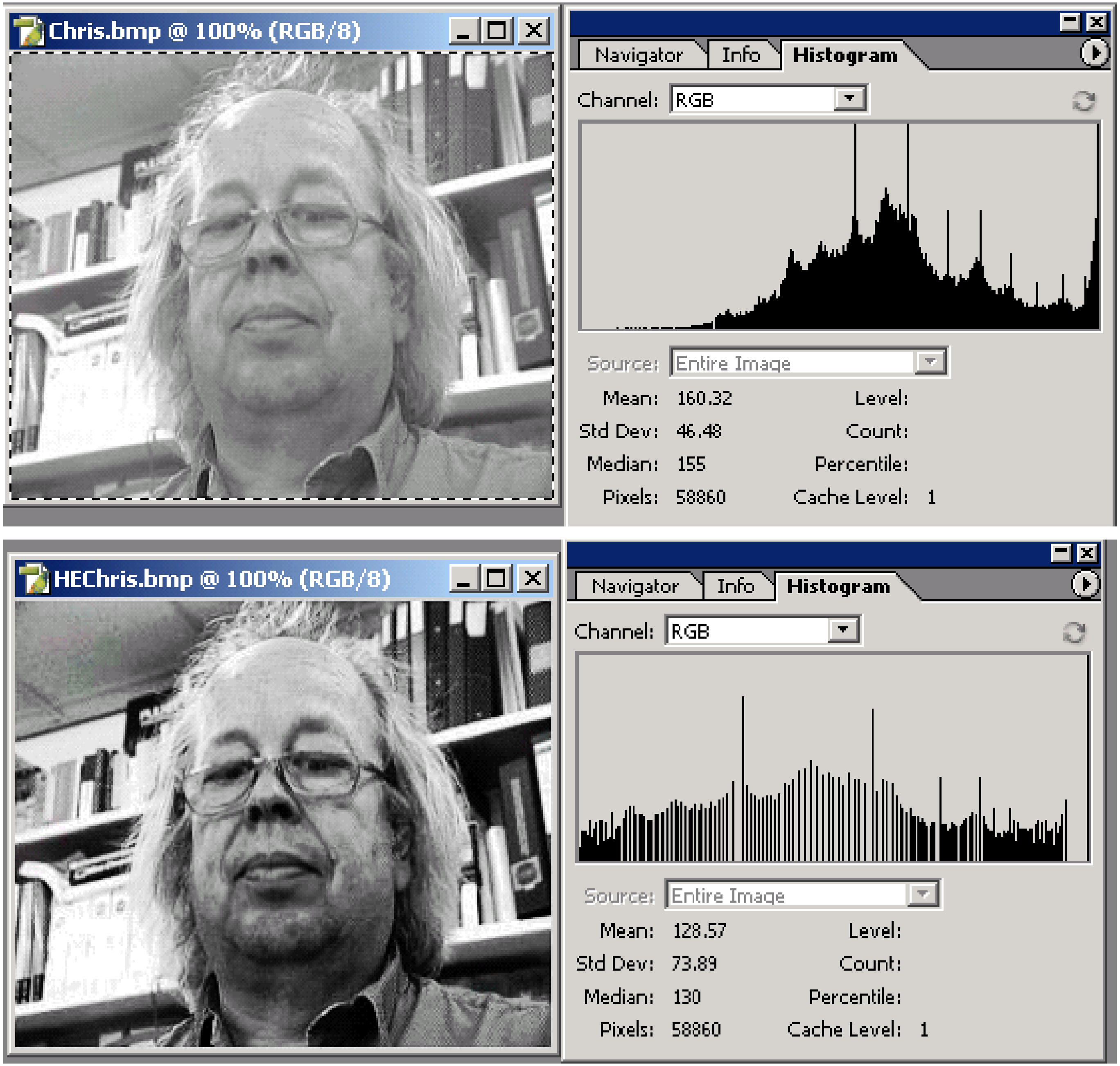
Step 5: Open a file “Inv_HE” and write inv_T entries into it.

Histogram Equalisation in MATLAB

Editor - C:\MATLAB7\work\HistEqual.m*

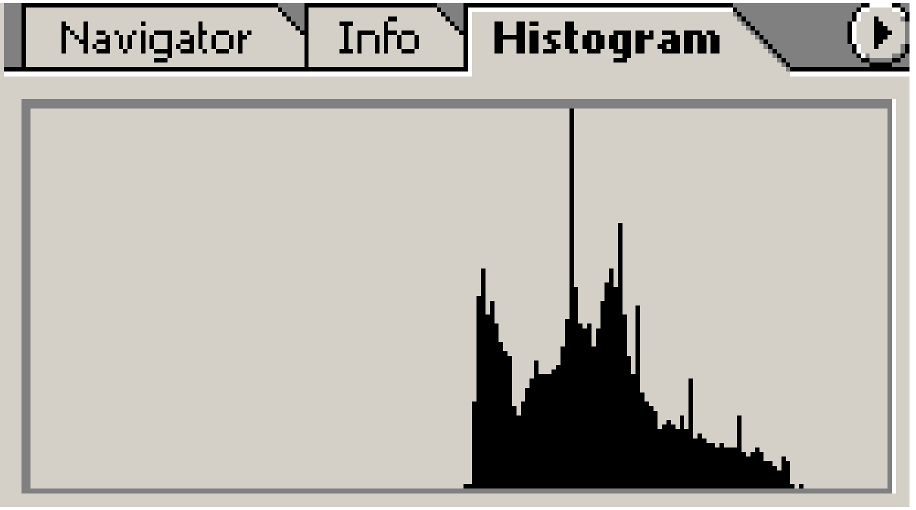
```
clear all;
c=imread('Chris.bmp');
g=zeros(1,256); % creating an array of zeros of size 256
cg=g;           %Cummulative frequency array
[m n]=size(c);
for i=1:1:m                                     % This loop
    for j=1:1:n                                 % creates the
            g(c(i,j)+1)=g(c(i,j)+1)+1;          % histogram for c
    end                                         % which is the
end                                             % frequency dist.
cg(1)=g(1);
for i=2:1:256
        for j=1:1:i
            cg(i)=cg(i)+g(j);             % Calculating the
        end                               % cummulative freq of c.
end
K=m*n;
for i=1:1:m
    for j=1:1:n
            c(i,j)=255*cg(c(i,j)+1)/K;          % HE operator.
    end
end
imshow(c);
imwrite(c,'HEChris.bmp')
```

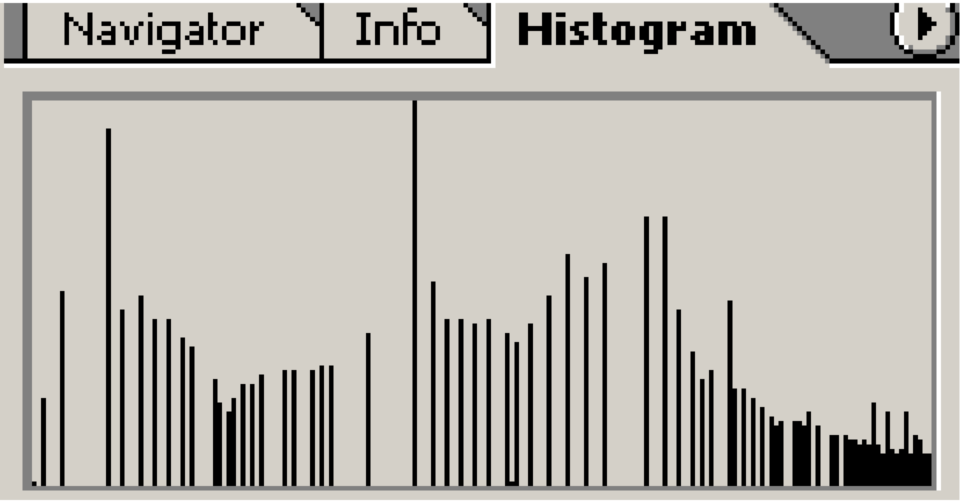
Example 1



- **Histogram is nearer to uniform than original.**
- **Improved Contrast but some added noise.**

Histograms Equalisation – Example 2

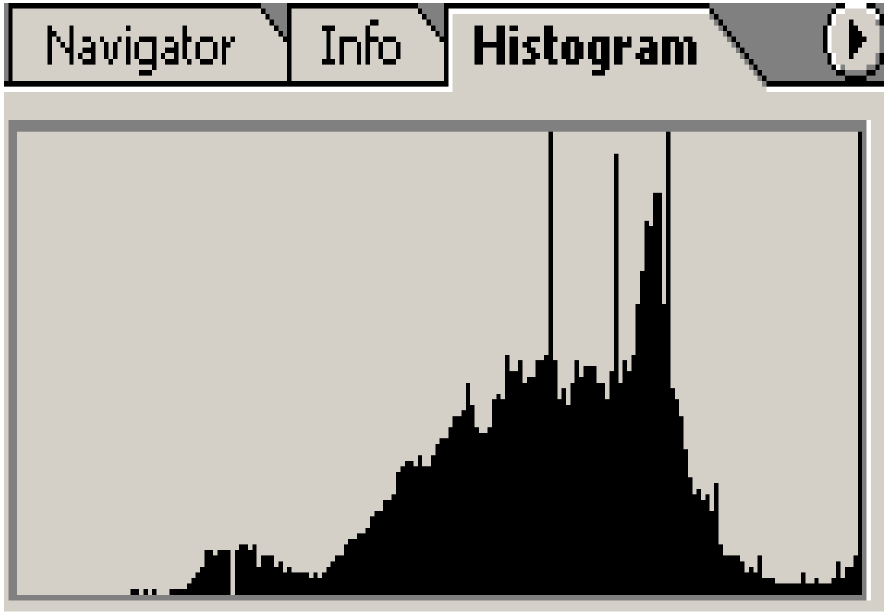




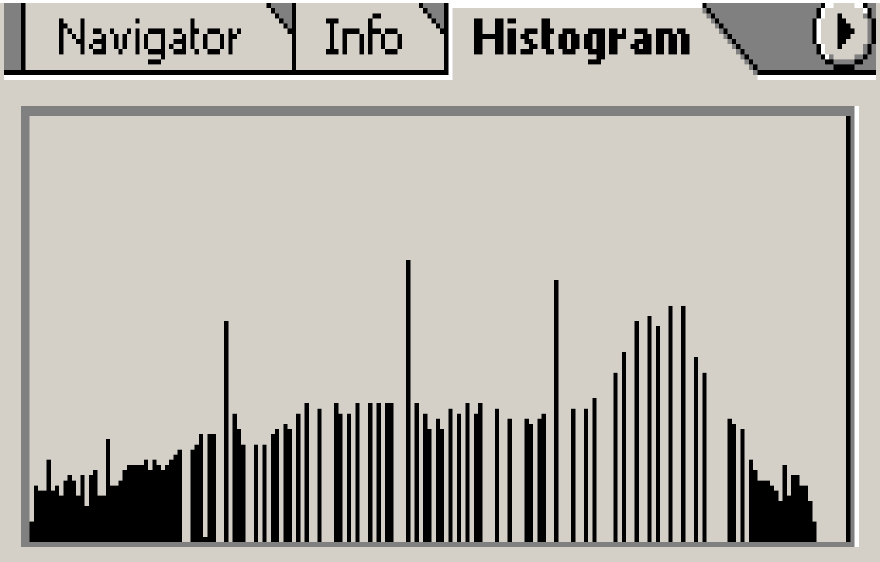
Gray levels in the output image are not very uniformly distributed.

Example 3

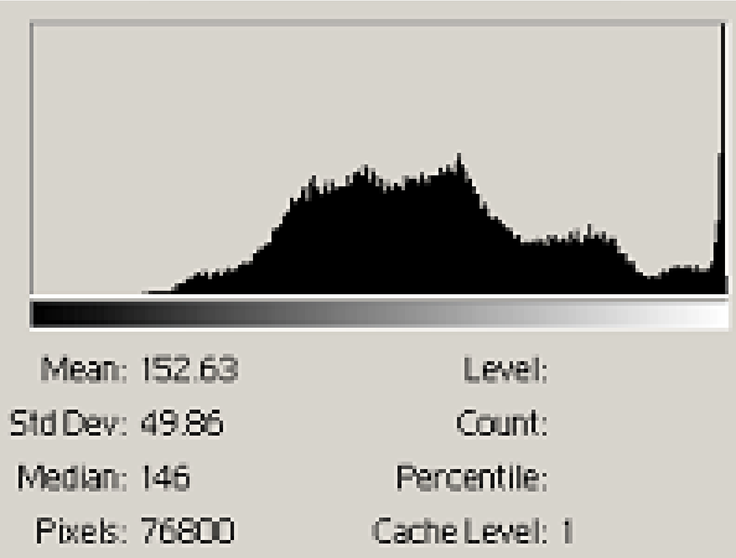
Original



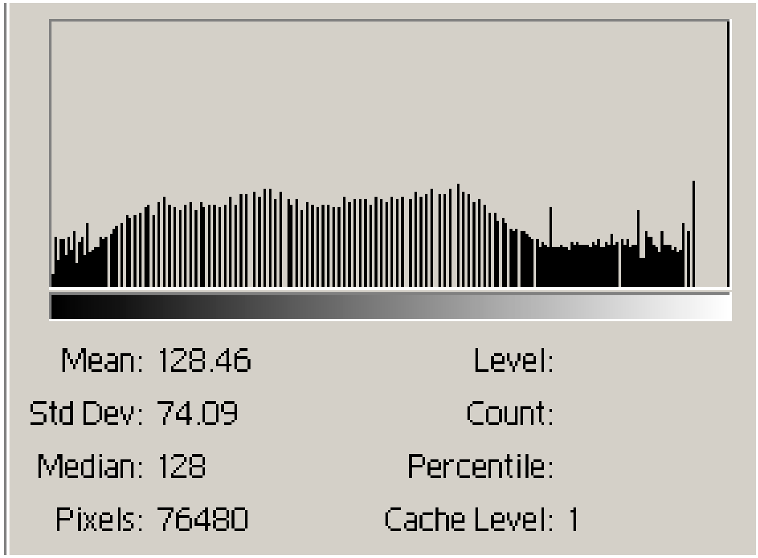
Histogram Equalised Image



Example 4



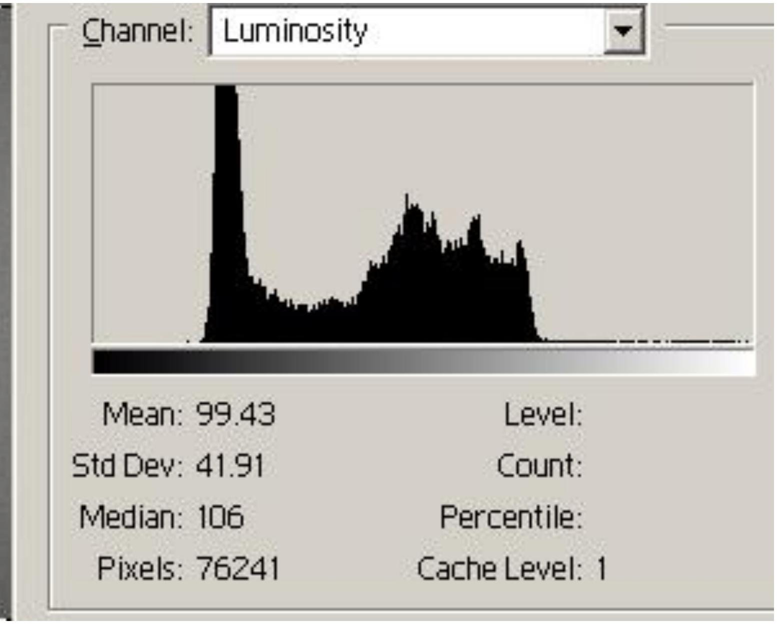
Mean: 152.63
Std Dev: 49.86
Median: 146
Pixels: 76800
Level:
Count:
Percentile:
Cache Level: 1



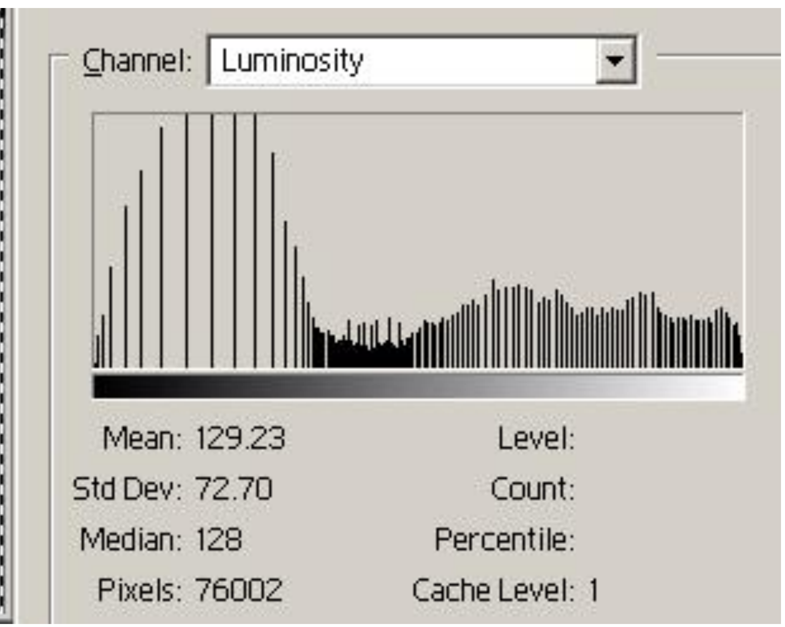
Mean: 128.46
Std Dev: 74.09
Median: 128
Pixels: 76480
Level:
Count:
Percentile:
Cache Level: 1

Example 5

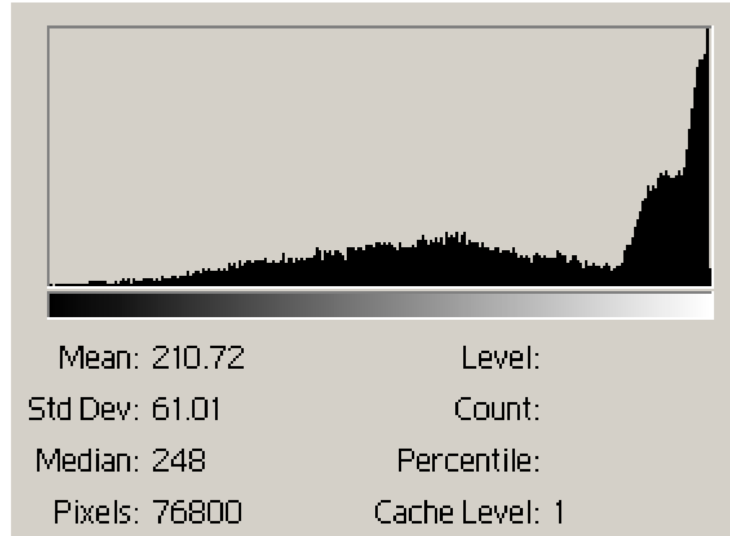
Original

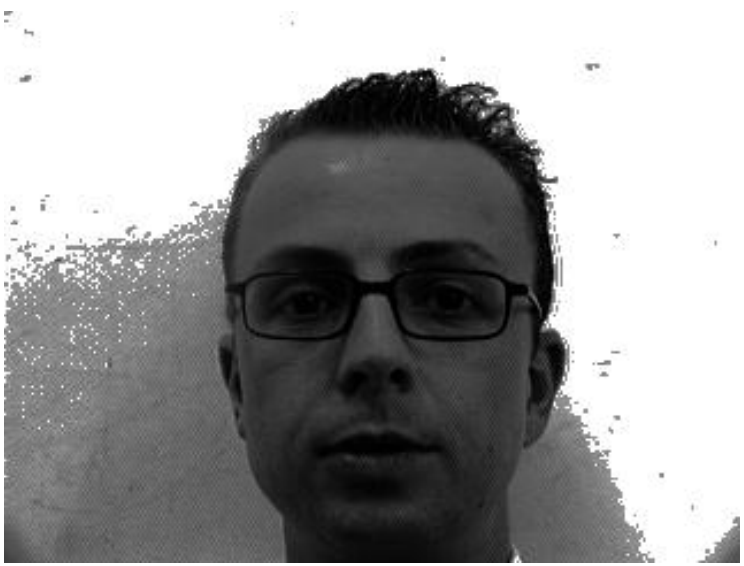


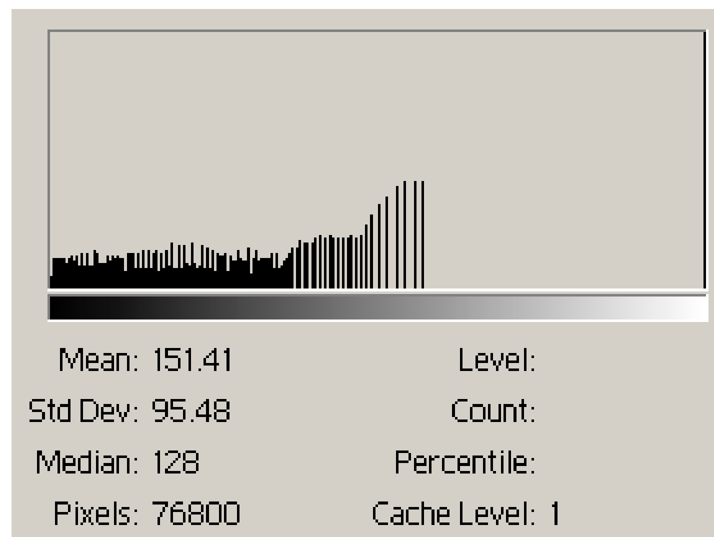
After HE operation



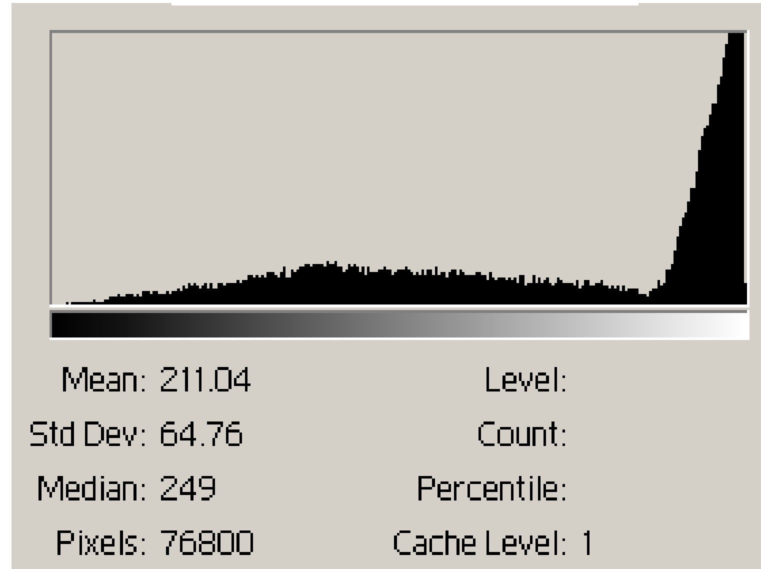
Example 6

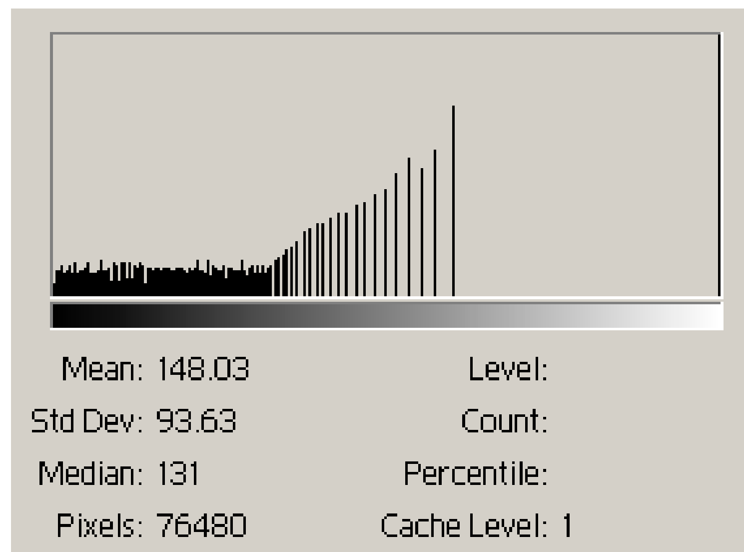

Mean: 210.72
Std Dev: 61.01
Median: 248
Pixels: 76800
Level:
Count:
Percentile:
Cache Level: 1




Mean: 151.41
Std Dev: 95.48
Median: 128
Pixels: 76800
Level:
Count:
Percentile:
Cache Level: 1

Example 7


Mean: 211.04
Std Dev: 64.76
Median: 249
Pixels: 76800
Level:
Count:
Percentile:
Cache Level: 1


Mean: 148.03
Std Dev: 93.63
Median: 131
Pixels: 76480
Level:
Count:
Percentile:
Cache Level: 1

Remarks on HE effects

- ***Histogram Equalisation*** **does improve contrast in some cases, but it may introduce noise and other undesired effect.**
- **Image regions that are dark and not clearly visible become clearer but this may happen at the expense of other regions.**
- **These undesired effect is a consequence of digitization. When digitise the continuous operations rounding leads to approximations.**
- **Images for which different regions exhibit different brightness level, may benefit from applying HE on sub-blocks of the images.**

Histogram Matching

- ***The previous examples show that the effect of HE differs from one image to another depending on global and local variation in the brightness and in the dynamic range.***
- **Applying HE in blocks may introduce boundary problems, depending on the block size.**
- ***Histogram Matching*** **is another histogram manipulation process which is useful in normalizing light variation in classification problems such as recognition.**
- **HM aims to transform an image so that its histogram nearly matches that of another given image.**
- **HM is the sequential application of a HE transform of the input image followed by the inverse of a HE transform of the given image.**

Pseudo Code for Histogram Matching

Step 1: Open the "Inv_HE" a file and read its entries into inv_T0.

This file should have been created by the HE++ algorithm for a good template image.

Step 2. Scan the input image I to calculate the Freq[0..L-1].

Step 3: From the Freq[] array compute the cumulative frequency array Cum_freq[0...L-1]:

```
{  Cum-freq[0] =Freq[0];
    for i=1 to L-1
        Cum_freq[i] =Cum_freq[i-1]+Freq[i];}
```

Step 4: Determine the HM transformation lookup table:

```
for i=1 to L-1
        j = round(Cum_freq[i]*(L-1)/N);
        HM_T[i] =Inv_T0[j];
```

Step 5: Transform the image using the lookup table HM_T.

Spatial filters classification

- ***Spatial filters can be classified by effect:***
  - ***Smoothing Filters: Aim to remove some small isolated detailed pixels by some form of averaging*** **of the pixels in the masked neighborhood. These are also called lowpass filters.**

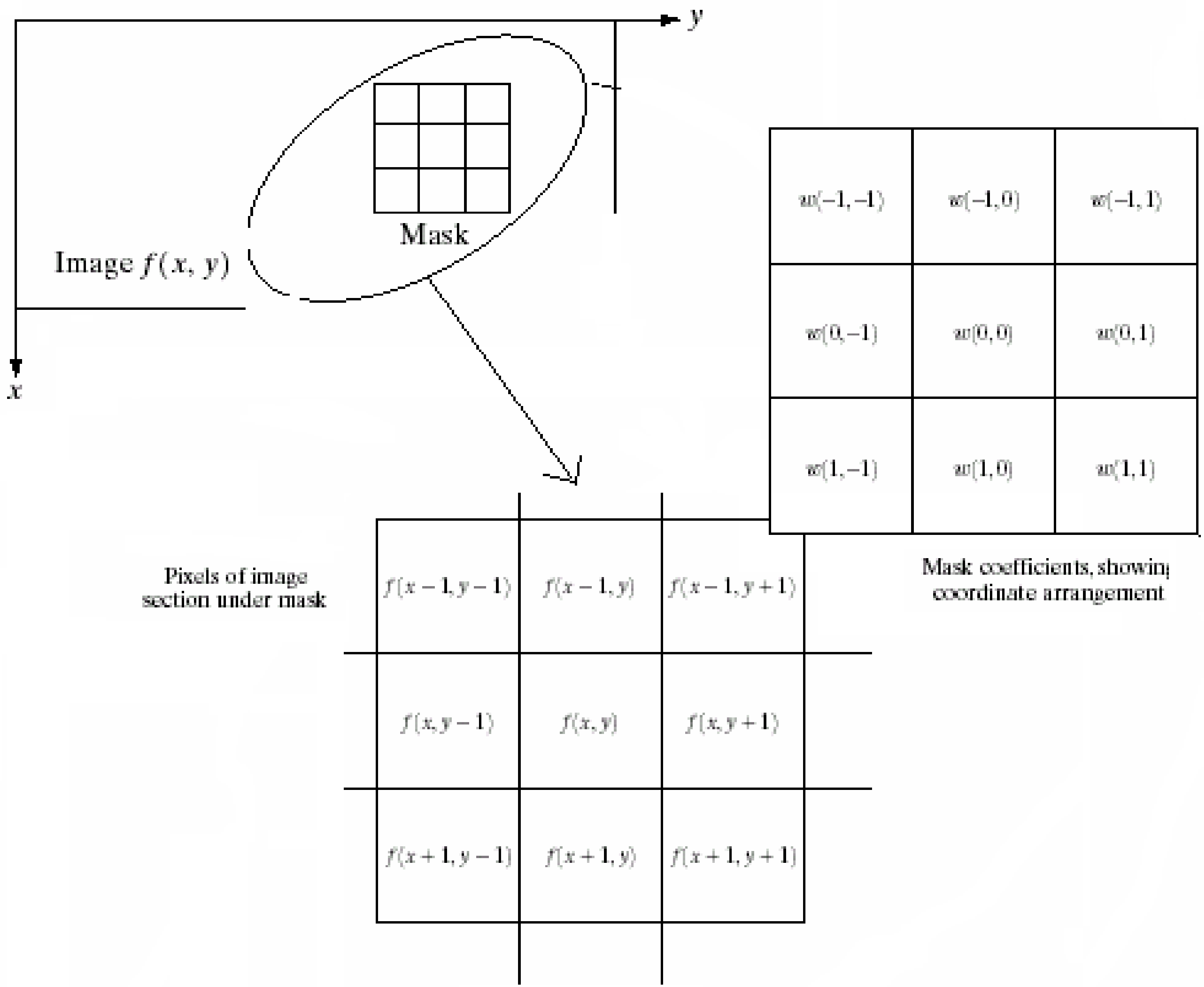
    Examples include Weighted Average, Gaussian, and order statistics filters.
  - ***Sharpening Filters: aiming at highlighting some features such as edges or boundaries of image objects.***

    Examples include the Laplacian , and Gradient filters.
- **Spatial filters are also classified in terms of mask size (e.g. 3x3, 5x5, or 7x7).**

Filtering in the spatial domain

- ***Filtering in the spatial domain refers to image operators that transform the gray value at any pixel (x,y) in terms of the pixel values in a square neighbourhood centred at (x,y) using a fixed integer matrix of the same size.***
- ***The integer matrix is called a filter, mask, kernel or a window. The operation is mainly the inner product (also known as the convolution) of the pixel neighbourhood subimage with the filter.***
- **The filtering process works by replacing each pixel value with the result of convolution at the pixel.**
- **Filtering is often used to remove noise in images that could occur as a result of less than perfect imaging devices, signal interference, or even as a result of image processing such as HE transforms.**

Spatial Filters - illustration



Spatial filters classification

- ***Spatial filters can be classified by effect:***
  - ***Smoothing Filters: Aim to remove some small isolated detailed pixels by some form of averaging*** **of the pixels in the masked neighborhood. These are also called lowpass filters.**

    Examples include Weighted Average, Gaussian, and order statistics filters.
  - ***Sharpening Filters: aiming at highlighting some features such as edges or boundaries of image objects.***

    Examples include the Laplacian , and Gradient filters.
- **Spatial filters are also classified in terms of mask size (e.g. 3x3, 5x5, or 7x7).**

A weighted average filter - Example

For example, if $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ and

the neighbourhood subimage is $\begin{bmatrix} 40 & 45 & 30 \\ 41 & 50 & 20 \\ 60 & 70 & 25 \end{bmatrix}$

then the pixel value of the output image that corresponds to the central pixel in the neighbourhood is replaced with:

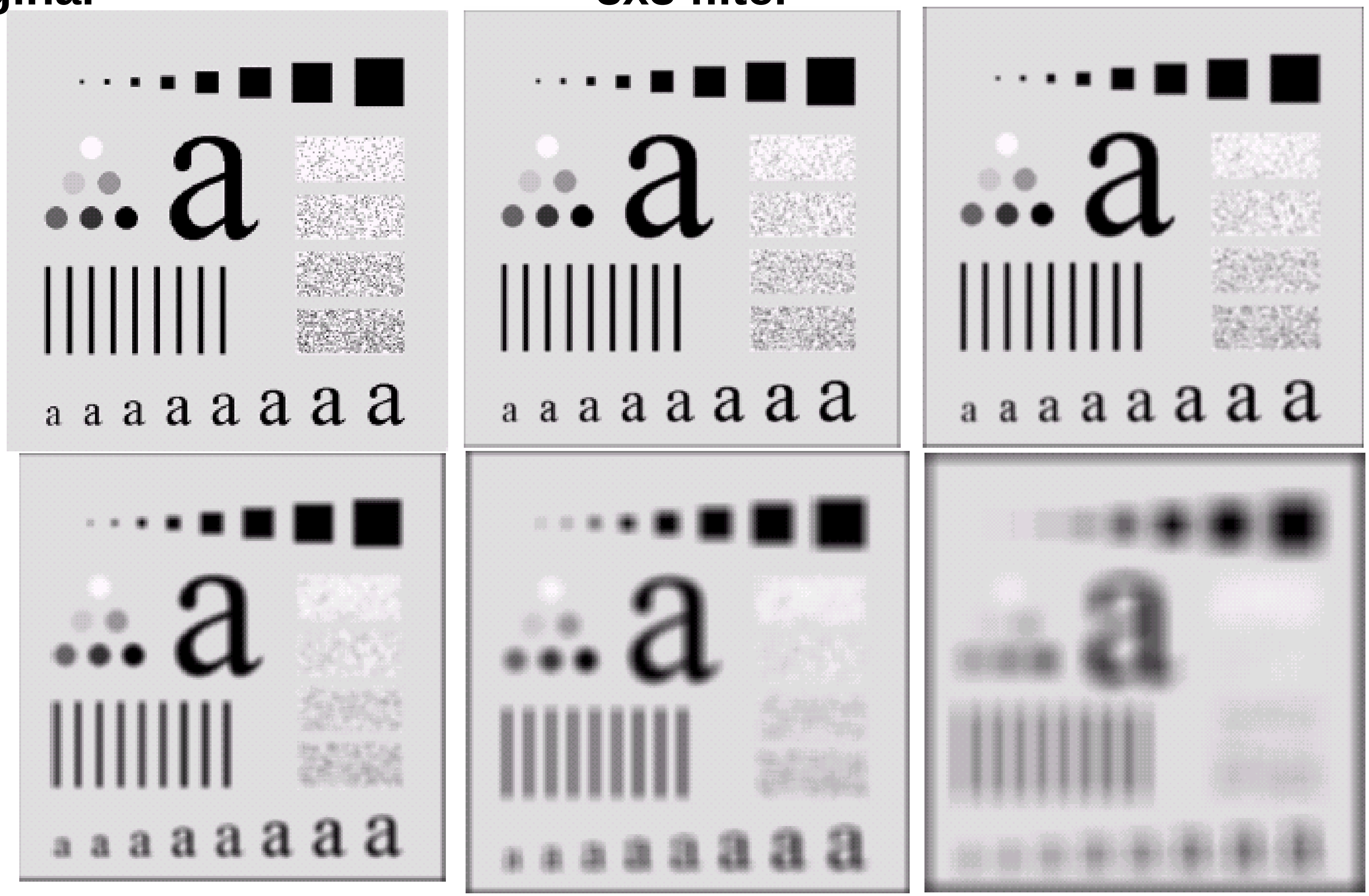
$$(40 + 2\times 45 + 30 + 2\times 41 + 4\times 50 + 2\times 20 + 60 + 2\times 70 + 25)/16$$

$$= \text{int}(44.1875) = 44.$$

Smoothing filters

- ***Smoothing Filters are particularly useful for blurring and noise reduction.***
- ***Smoothing filters work by reducing sharp transition in grey levels.***
- ***Noise reduction is accomplished by blurring with linear or non-linear filters (e.g. the order statistics filters).***
- **Beside reducing noise, smoothing filters often remove some significant features and reduce image quality.**
- **Increased filter size result in increased level of blurring and reduced image quality.**
- **Subtracting a blurred version of an image from the original may be used as a sharpening procedure.**

Effect of Averaging Linear Filters Vs filter size



The extent of burring increases the larger the filter is.

Filtering using MATLAB

- MATLAB provides easy to use **linear spatial** filter functions:
  - ***Definition:*** ***w=fspecial(‘type’, parameters)***

    is used to create a filter of the declared type with the given parameters which may gives the size or other values relating to the given type.
  - ***Applying:*** ***f = imfilter(c, w, f_mode, boundary-options, size-options)***

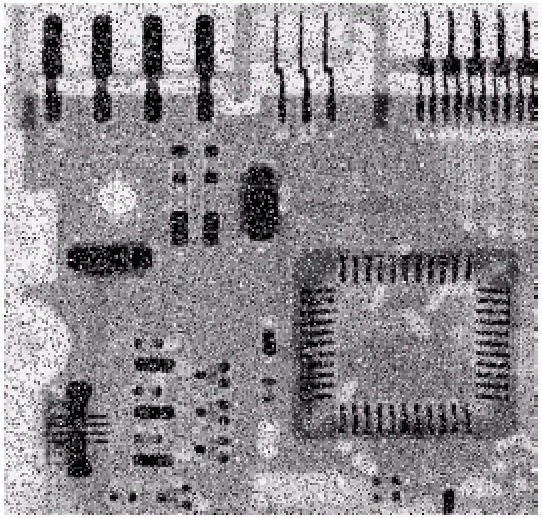
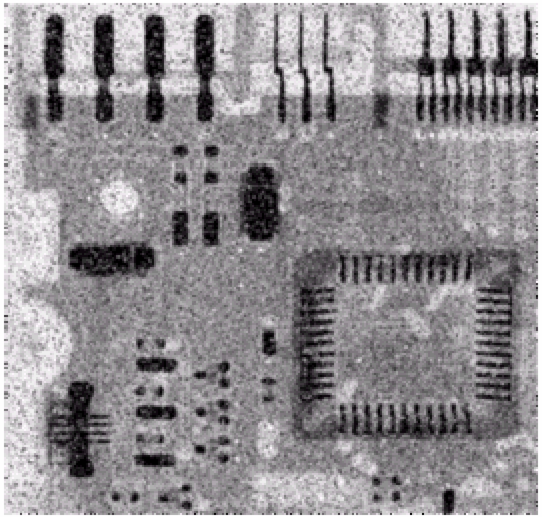
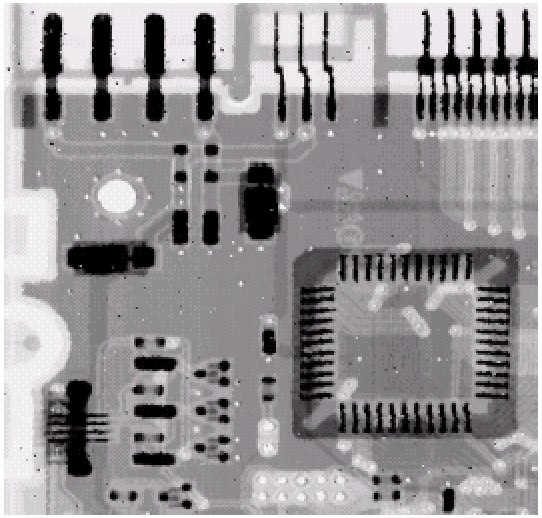
    We normally use the defaults for the last 3 parameters:

    ‘corr’ for f_mode; **0** ***for boundary-option;*** ***‘same’ for size***

```
clear all
c=imread('original.bmp');
imview(c);
w=fspecial('average'); %This is the default 3x3averaging filter.
                       % w=fspecial('average',5) stands for 5x5 filter.
F2=imfilter(c,w);
imshow(F2);
imwrite(F2,'faveoriginal.bmp')
```

Order Statistical filters

- ***These refer to non-linear filters whose response is based on ordering the pixels contained in the neighborhood. Examples include Max, Min, Median and Mode filters.***
- ***The median which replaces the value at the centre by the median pixel value in the neighbourhood, (i.e. the middle element when they are sorted.***
- ***Median filters are particularly useful in removing impulse noise, also known as salt-and-pepper.***

Noisy image **Averaging 3x3 filter** **Median 3x3 filter**

Order Statistics Filters in MATLAB

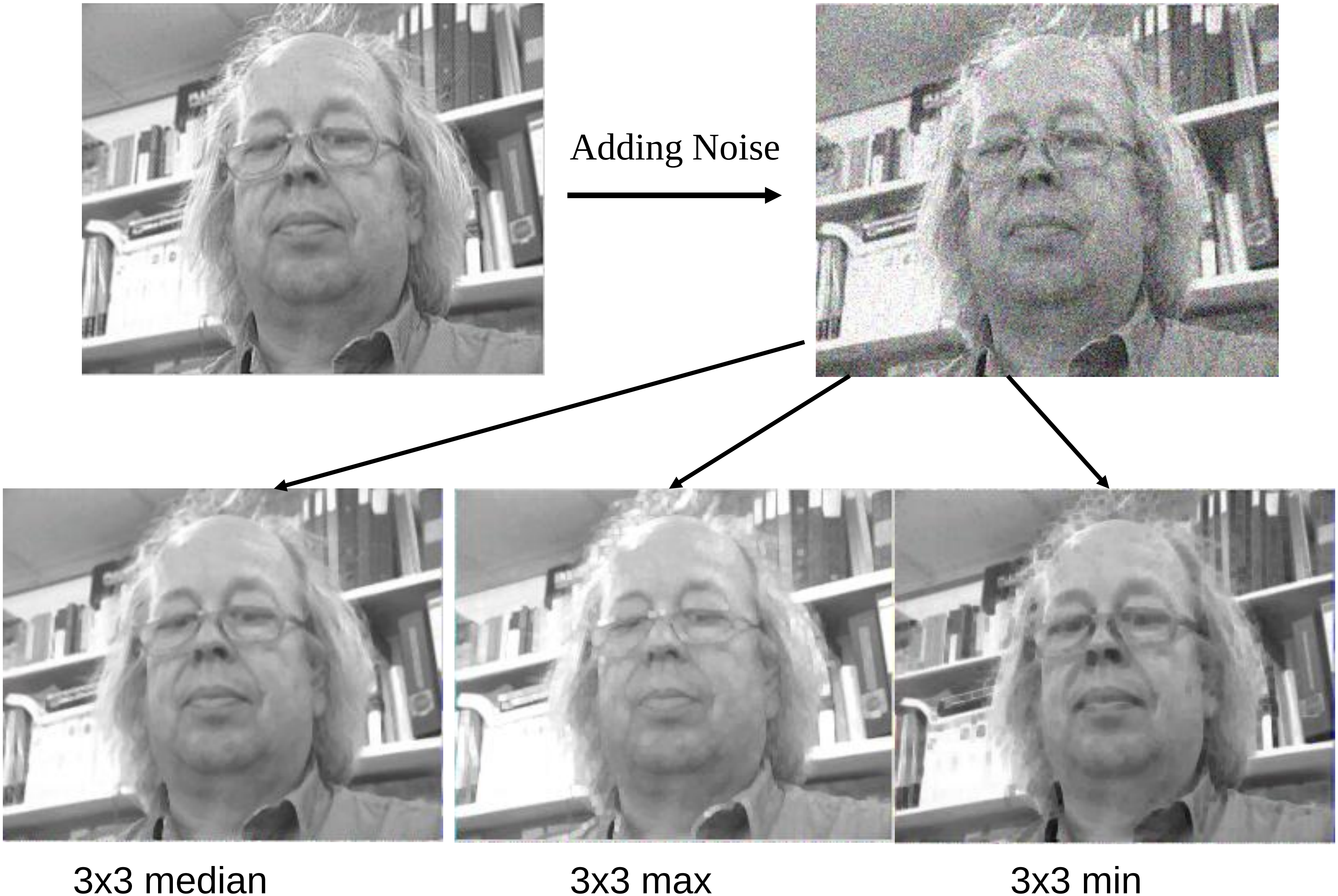
In MATLAB Order Statistics filter are applied as follows:

f = ordfilt2 (c, order, domain)

***where order is the position required when the elements of the given neighbourhood are* sorted, and domain is a matrix of 1's and 0's that specify the pixel locations in the neighbourhood that are included in the computation.**

```
clear all
c=imread('Harin.bmp');
imview(c);
f=ordfilt2(c, 1, ones(3, 3));
imshow(f);
imwrite(f, 'minHarin.bmp');
f=ordfilt2(c, 9, ones(3, 3));
imshow(f);
imwrite(f, 'maxHarin.bmp');
f=ordfilt2(c, 5, ones(3, 3));
imshow(f);
imwrite(f, 'medianHarin.bmp')
```

Example – Effect of different order statistics filters



Sharpening Spatial Filters

- **Sharpening aims to highlight fine details (e.g. edges) in an image, or enhance detail that has been blurred through errors or imperfect capturing devices.**
- **Image blurring can be achieved using averaging filters, and hence sharpening can be achieved by operators that invert averaging operators.**
- **In mathematics averaging is equivalent to the concept of integration along the gray level range:**

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

Integration inverts differentiation and as before, we need a digitized version of derivatives.

Derivatives of Digital functions of 2 variables

- **As we deal with images, which are represented by digital function of two variables, then we us the notion of partial derivatives rather than ordinary derivatives.**
- **The first order partial derivatives of the digital image f(x,y) in the x and in the y directions at (x,y) pixel are:**

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y), \text{ and } \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y).$$

- **The first derivative is 0 along flat segments (i.e. constant gray values) in the specified direction.**
- **It is non-zero at the outset and end of sudden image discontinuities (edges or noise) or along segments of continuing changes (i.e. ramps) .**

2^{nd} order Derivatives & the Laplacian operator

- **The first derivative of a digital function f(x,y) is another digital image and thus we can define 2^{nd} derivatives:**

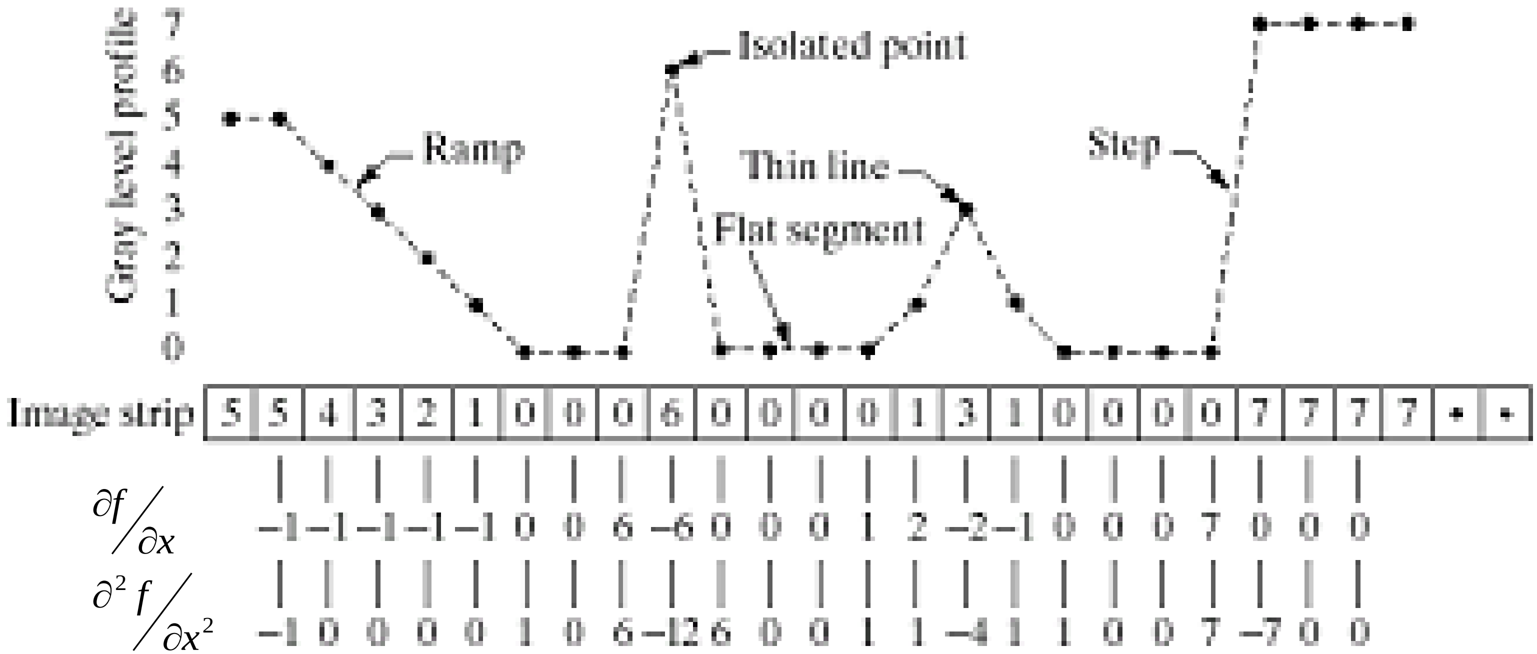
$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial\left(\partial f / \partial x\right)}{\partial x} = f(x+1, y) + f(x-1, y) - 2f(x, y), \quad \text{and}$$

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

- **Other second order partial derivatives can be defined similarly, but we will not use here.**
- **The Laplacian operator of an image f(x,y) is:**

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

Digital derivatives –Example



1. $\partial f / \partial x \neq 0$ along the entire ramp, but $\partial^2 f / \partial x^2 \neq 0$ at its ends.

   Thus $\partial f / \partial x$ detects thick edges while $\partial^2 f / \partial x^2$ detects thin edges.

2. $\partial f / \partial x$ have stronger response to gray-level step,

   but $\partial^2 f / \partial x^2$ produces a double response at gray-level steps.

The Laplacian Filter

- **The Laplacian operator, applied to an Image can be implemented by the 3x3 filter:**

$$\nabla^2(f) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

- **Image enhancement is the result of applying the difference operator:**

$$Lap(f) = f - \nabla^2(f) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Laplacian filter in MATLAB

- The Laplacian filter is a linear spatial filter, and hence we can use similar instruction as for the averaging filter.
  - ***Definition: w=fspecial('laplacian', parameters)***

    Here, parameters is a number 0≤a ≤1, and the default is 0.5.
  - ***Applying: f = imfilter(c, w)***

```
clear all
c=imread('Ayser.bmp');
imview(c);
v=fspecial('laplacian'); %One parameter is associated with Laplacian
                         % which is a number a in the range [0..1].
                         % The default is a=0.5, which used here.
f=imfilter(c,v);         % Applying the laplacian operator
imview(f);
imwrite(f,'lapAyser.bmp');
F1=c-f;                  % This is use for enhancing the input image.
imview(F1);
imwrite(F1, 'LAPAyser.bmp')
```

AverageFilter.m* | Laplacianfilter.m

Laplacian filter –Example

Original Image

Image after Laplacian filer

Note the enhanced details after applying the Laplacian filter.

Example

Laplacian Operator

Laplacian Enhanced

Image = f - ∇^2f.

Combining various enhancement filters

- **The effect of the various spatial enhancement schemes doesn’t always match the expectation, and depends on input image properties.**

  e.g. histogram equalization introduces some noise.

- **The application of any operator at any pixel does not depend on the position of the pixel, while the desired effect is often reqired in certain regions.**

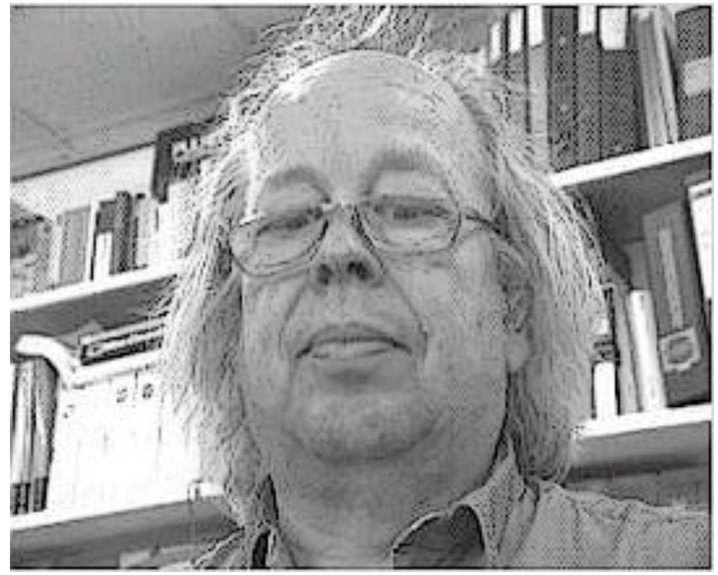
  e.g. an averaging filters blurs smooth areas as well as significant feature regions.

- **Enhancing an image is often a trial & error process.**

  In practice one may have to combine few such operations in an iterative manner .

  In what follows we try few combined operations.

Combining Laplacian and HE



LAPChris

HEChris

LAPHEChris

HELAPChris

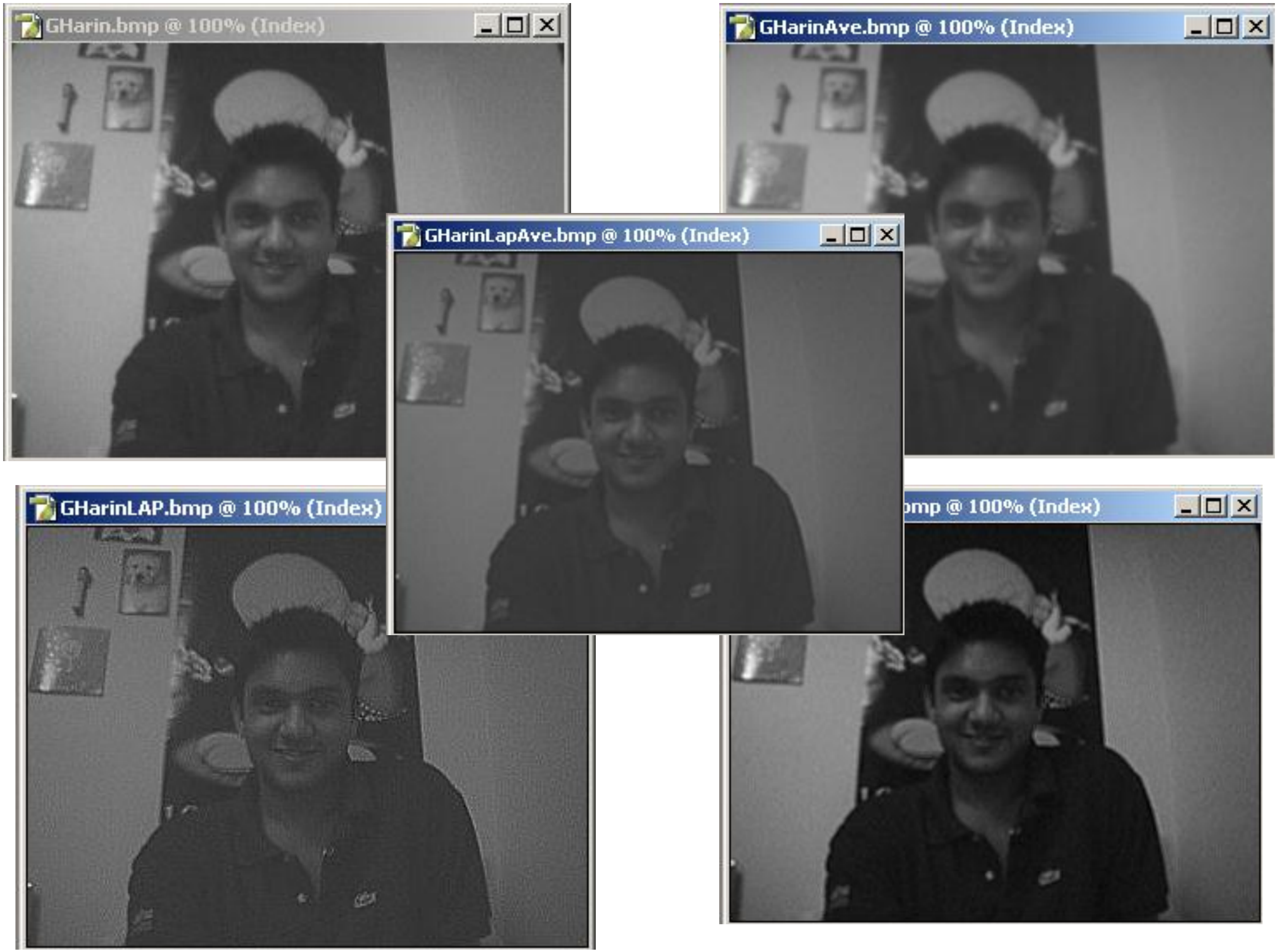
Combined Laplacian & HE

(a) **(b)**

(c) **(d)**

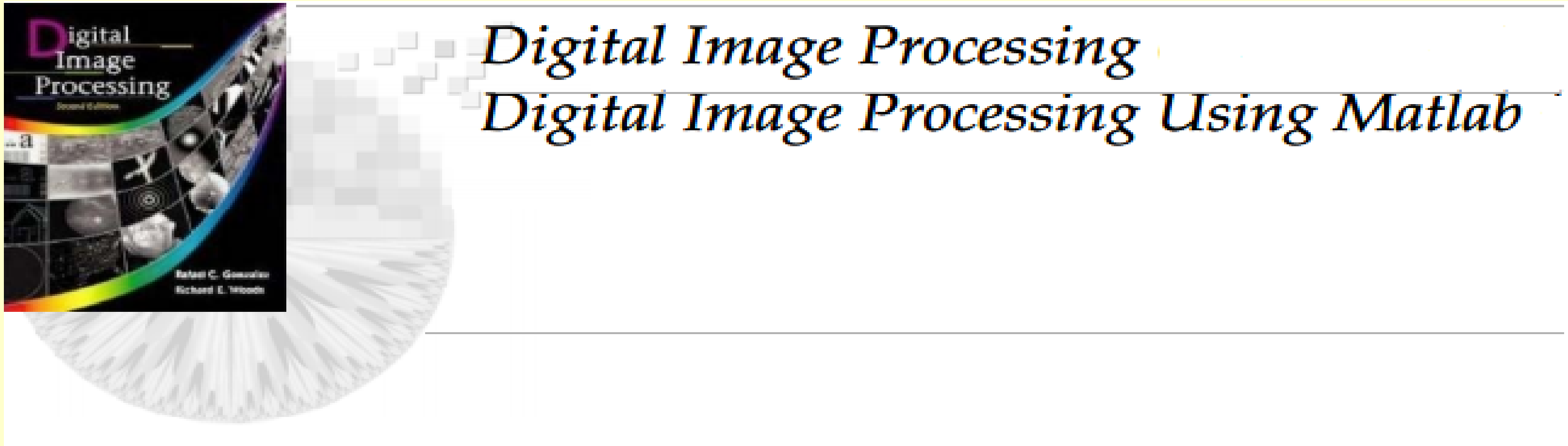
(a) Original, (b) After HE, (c) After Laplacian, (d) HE after Laplacian

Combining Average and Laplacian



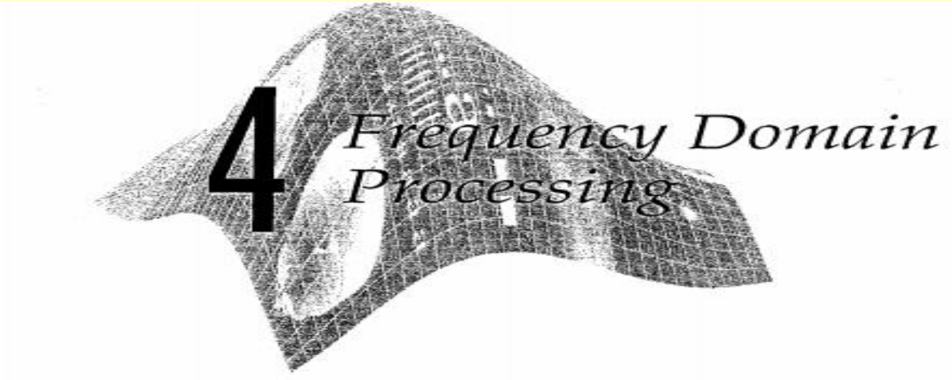
Combining Laplacian & Median Filter

End of Chapter 3



Digital Image Processing

Prepared by:
Dr. Ali J. Abboud
University of Diyala
2012-2013



Key Features of Chapter 3:

- ***Fourier Transforms*.**
- ***Mathematical background: Complex numbers*.**
- ***Fourier Spectrum*.**
- ***2-Dimensional DFT .***
- ***Filtering in the Frequency Domain.***
- ***Lowpass and Highpass Filters in the Frequency Domain .***

Introduction

- ✓ **The spatial domain refers to the representation of an image as the array of gray-level intensity.**
- ✓ **The electromagnetic spectrum consist of sinusoidel waves of different wavelengths (frequencies).**
- ✓ **The frequency content of an image refers to the rate at which the gray levels change in the image**
- ✓ **Rapidly changing brightness values correspond to high frequency terms, slowly changing brightness values correspond to low frequency terms**
- ✓ **The Fourier transform is a mathematical tool that analyses a signal (e.g. images) into its spectral components depending on its wavelength (i.e. frequency) content.**

Fourier Transforms

- ✓ **In 1822, Jean B. Fourier has shown that any function f(x) that have bounded area with the x-axis can be expressed as a linear combination of sines and/or cosines of different frequencies.**
- ✓ **This has also developed for functions of 2 variables, e.g. images.**

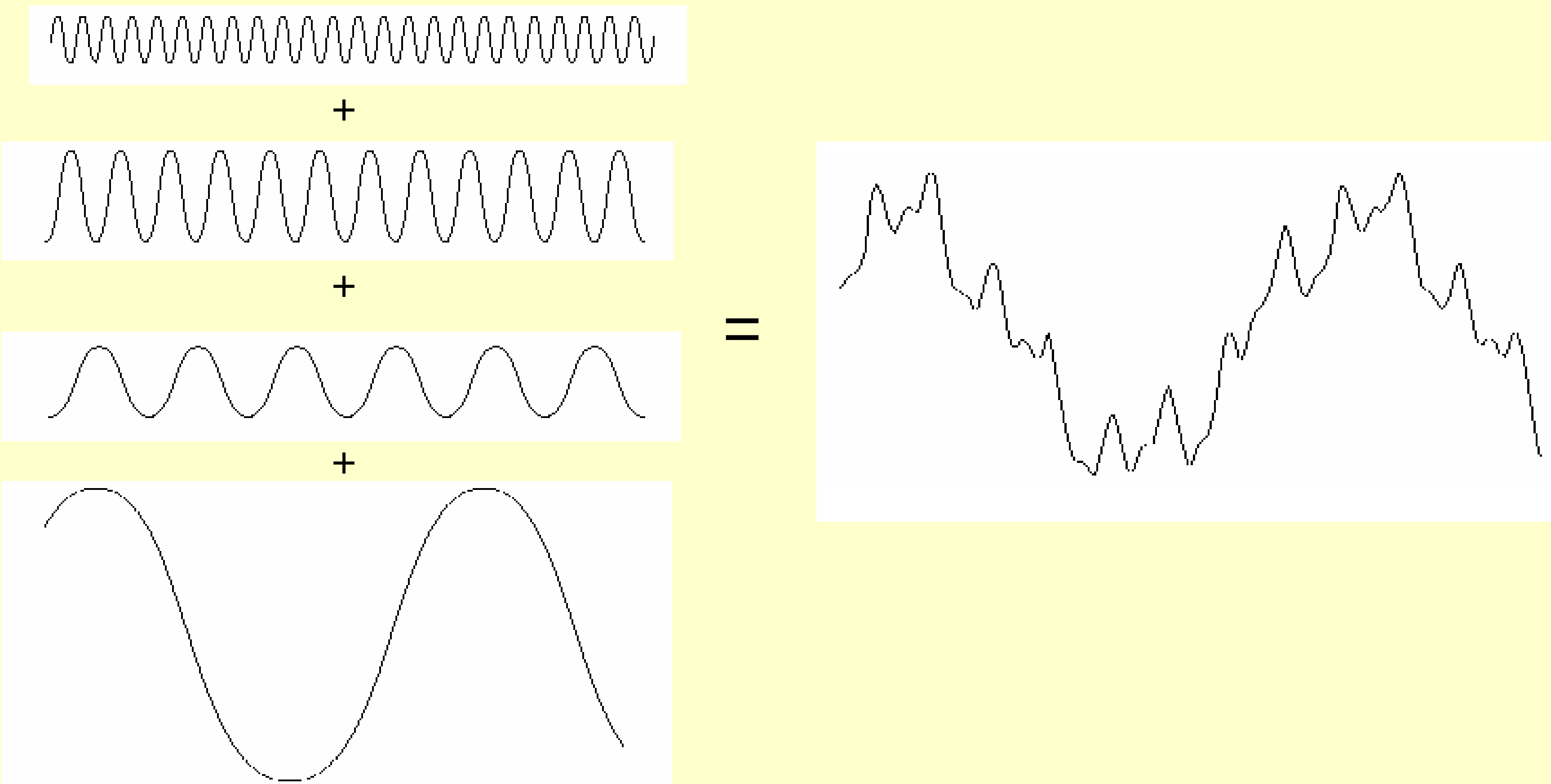
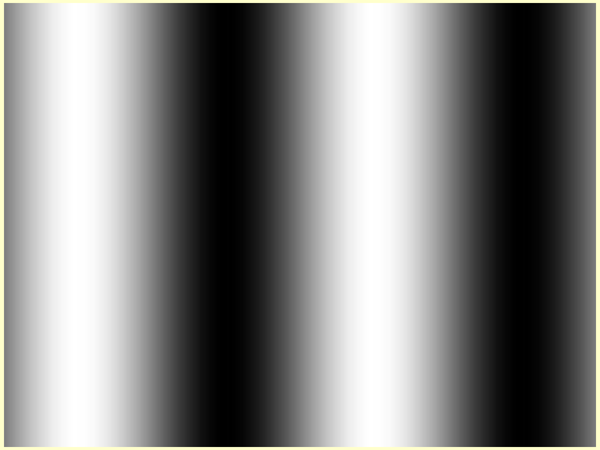
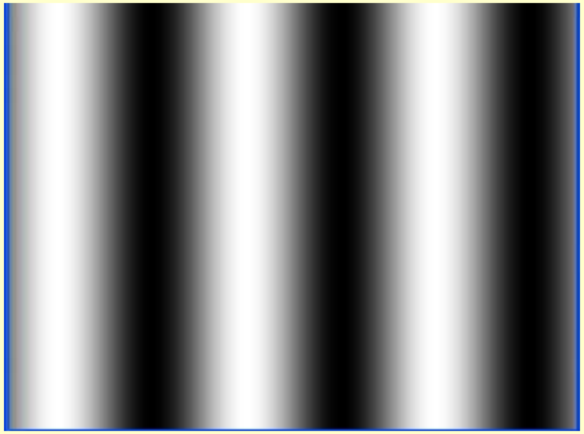


Illustration of Fourier Analysis for images

Every row is Sine wave of frequncy 1

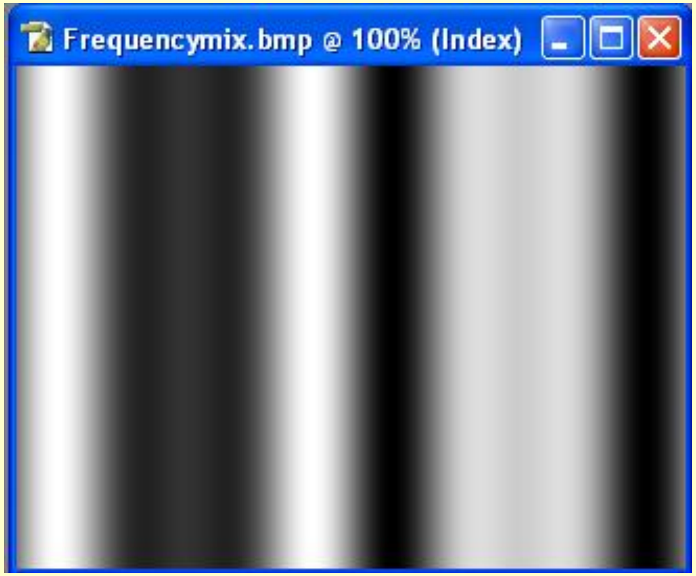


Sine wave with frequncy 2



Sine wave with frequncy 3

Combined waves frequncy 1+2+3



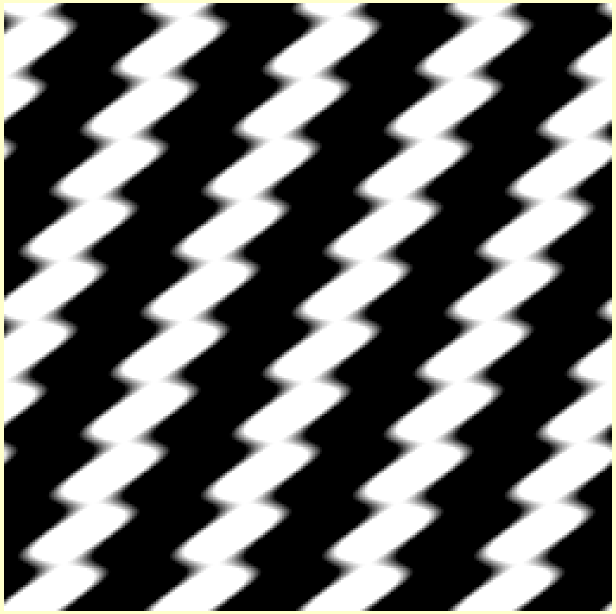
Mixed wave with frequncy 5, 2 &1

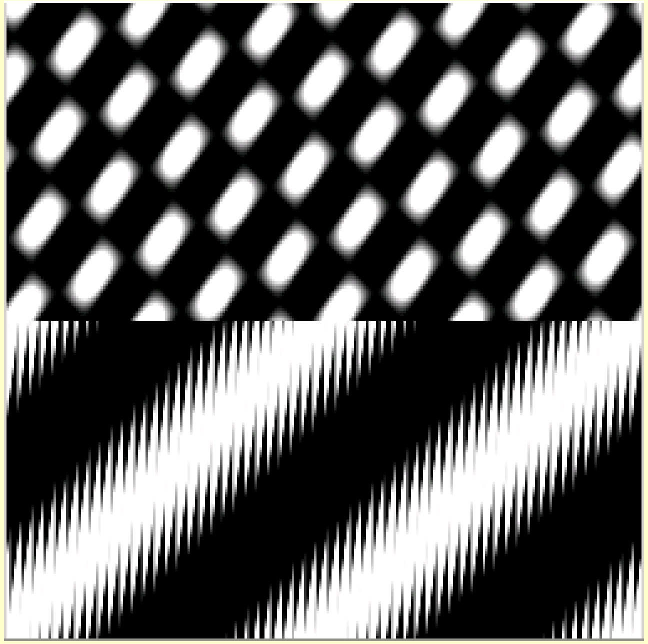
MATLAB generated images

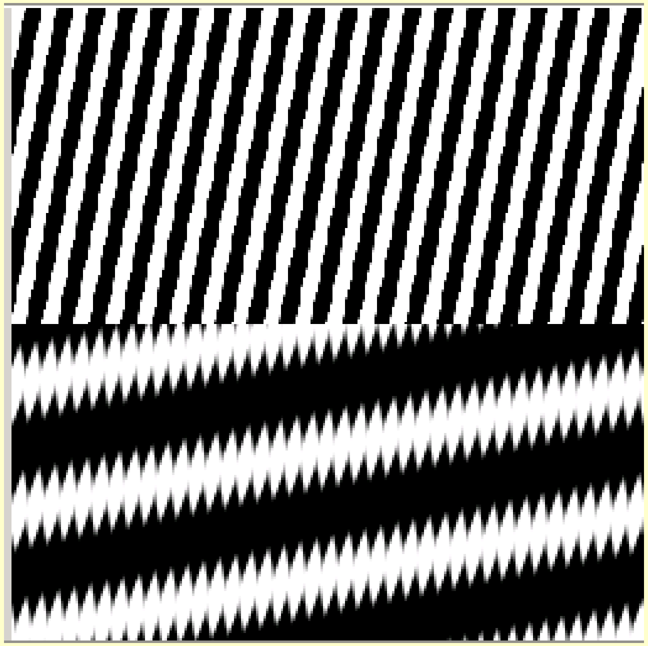
- ✓ **MATLAB can be used to generate images with patterns of any desired rate of change of brightness.**
- ✓ **For this we need to use trigonometric functions of 2 variable as indicated by the following code:**

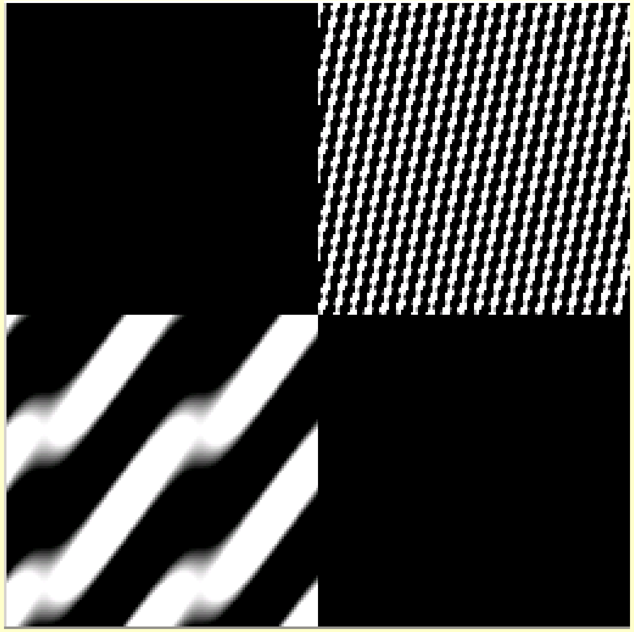
```
clear all;
A=zeros(256,256);
B=A;
for i=1:1:256
    for j=1:1:256
        A(i,j)=2*sin(pi*(i+2*j)/64);    // the 2's and 64 can be changed
        B(i,j)=cos(pi*(3*i+j)/32); // the 3 and 32 can be changed
    end
end
C=A+B;
Imshow(C);
imwrite(C, 'SinoPattern2.bmp')
```

Images generated from Sinoside function









Mathematical Background - Complex numbers

- ✓ **A complex number z is a point (a,b) in the plane. Addition and multiplication are defined as:**

  (a,b)+(c,d) =(a+c,b+d)

  (a,b)*(c,d) = (ac-bd,ad+bc).

  For example:

  (3,2)+(1,-1) = (4,1), and

  (2,1)*(2,-1) = (4-(-1),0) = (5,0).

- ✓ **A complex number z=(a,b) can be expressed as:**

  z=a+ib.

  Here i=(0,1), a is called the real part and b is the imaginary part of z.

  It is easy to show that: i^2 =-1 (i.e. i=√-1).

- ✓ **The conjugate of a complex number z=a+ib is z* = a – ib.**

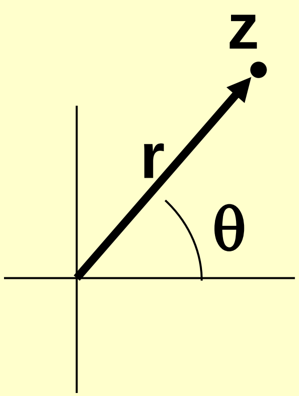
  For example, if z= 3+2i then z* = 3 – 2i.

  For any complex number z = a +ib, z z* =z*z = (a^2 + b^2).

- ✓ **The complex numbers is an "algebraically closed field".**

Complex numbers – Polar representation

✓ **A complex $z = a+ib$ can be represented as $z = r\cos\theta + i\, r\sin\theta$, where $r = \sqrt{(a^2+b^2)}$, and $\tan\theta = b/a$.**

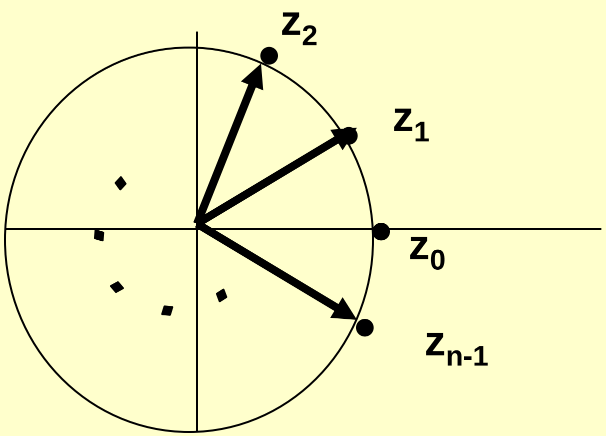


✓ **Due to properties of sin θ and cos θ , we write $z = re^{i\theta}$**

For example: **$1 + i\sqrt{3} = 2\, e^{i\pi/3}$, and** $1+i = \sqrt{2}\, e^{i\pi/4}$.

✓ **For any θ, $e^{-i\theta} = \cos\theta - i\sin\theta$.**

✓ **Roots of unity: The equation $z^n = 1$ has n complex solutions, called n-th roots of unity, namly: $z_0=1$, $z_1=e^{i2\pi/n}$, $z_2=e^{i4\pi/n}$, ... , and $z_{n-1}=e^{i2\pi(n-1)/n}$. These are equidistant points on the unit circle.**



Fourier Transform - Definition

- ✓ **The one-dim Fourier transform of a function f(x) is defined as:**

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi u x} dx.$$

and the inverse Fourier transform of F(u) is :

$$\mathrm{f(x)} = \int_{-\infty}^{\infty} F(u) e^{i2\pi x u} du.$$

- ✓ **The Discrete Fourier Transform (DFT) of f(x) is defined as:**

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-i2\pi u x / M}.$$

and the inverse Fourier transform is defined as :

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{i2\pi u x / M}.$$

- ✓ **The frequency domain of f(x), is the set { 0, 1, ..., M-1} of u values.**
- ✓ **Note that the $e^{i2\pi ux/M}$ are simply the M-th roots of unity.**

Fourier Transform - continued

- ✓ **Unlike f(x), F(u) is a complex valued function, and in terms of circular functions:**

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)[\cos(2\pi ux/M) - i\sin(2\pi ux/M)]$$

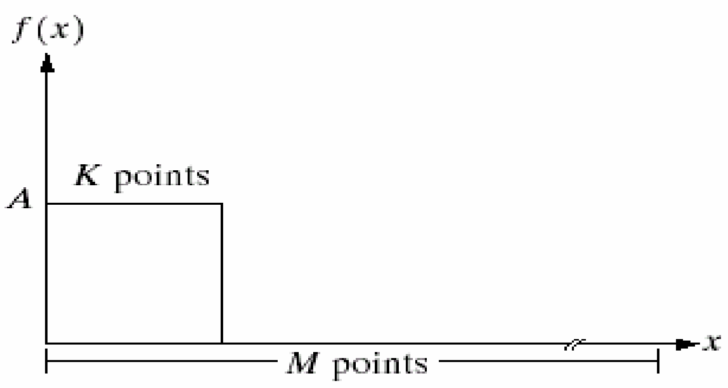
$$= (\frac{1}{M}\sum_{x=0}^{M-1} f(x)[\cos(2\pi ux/M)) - i(\frac{1}{M}\sum_{x=0}^{M-1} f(x)\sin(2\pi ux/M)),$$

i.e. $F(u) \equiv R(u) + iI(u).$

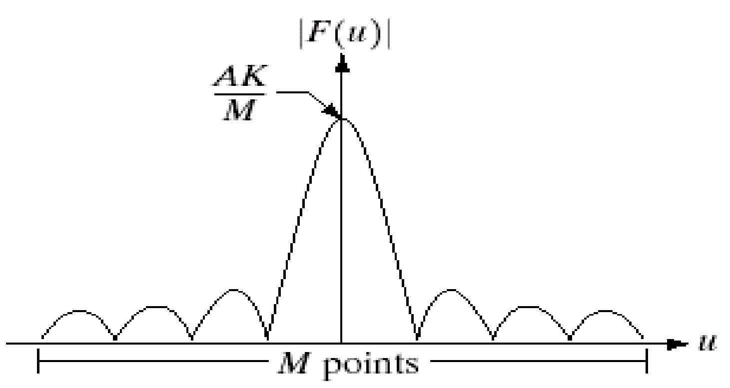
- ✓ **The modulus of F(u), $|F(u)| = [R(u)^2 + I(u)^2]^{1/2}$, is called the frequency spectrum of the transform.**
- ✓ **The phase angle of the transform is:**

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

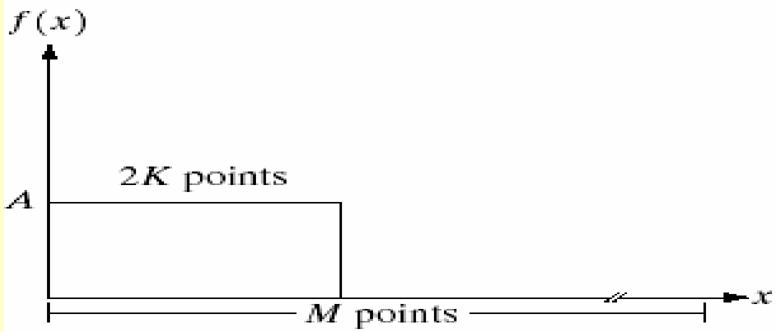
The Fourier spectrum – Examples



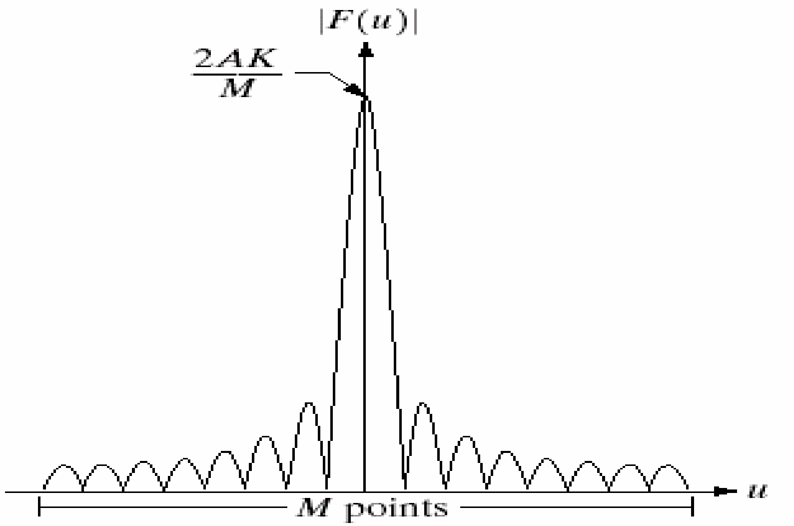
A discrete function, K=8.



Its frequency spectrum.



A discrete function, 2K=16.



Its frequency spectrum.

Fourier Transform in MATLAB

- ✓ **The MATLAB fft(x) functions provides a fast implementation of the Fourier transform for one dimensional function x=x(t).**

- ✓ **The functions X = fft(x) and x = ifft(X) implement the transform and inverse transform pair given for vectors of any length m using the given formulae. It exploits the doubling and shifting properties of sine and cosine functions.**

Example: MATLAB Help item on Fourier Transforms.

The 2-dimensional DFT

- ✓ **The DFT of a digitised function f(x,y) (i.e. an image) is defined as:**

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)}.$$

and the inverse DFT is defined as:

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i2\pi(ux/M+vy/N)}.$$

- ✓ **Note that, F(0,0) = the average value of f(x,y) and is refered to as the DC component of the spectrum.**

- ✓ **It is a common practice to multiply the image f(x,y) by $(-1)^{x+y}$. In this case, the DFT of $(f(x,y)(-1)^{x+y})$ has its origin located at the centre of the image, i.e. at (u,v)=(M/2,N/2).**

The Fourier spectrum – in 2D

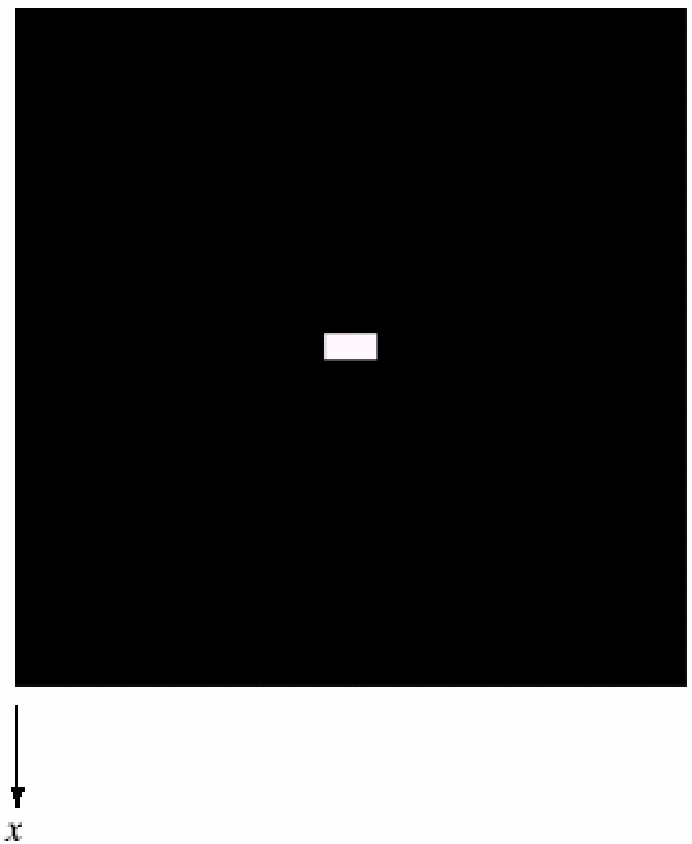
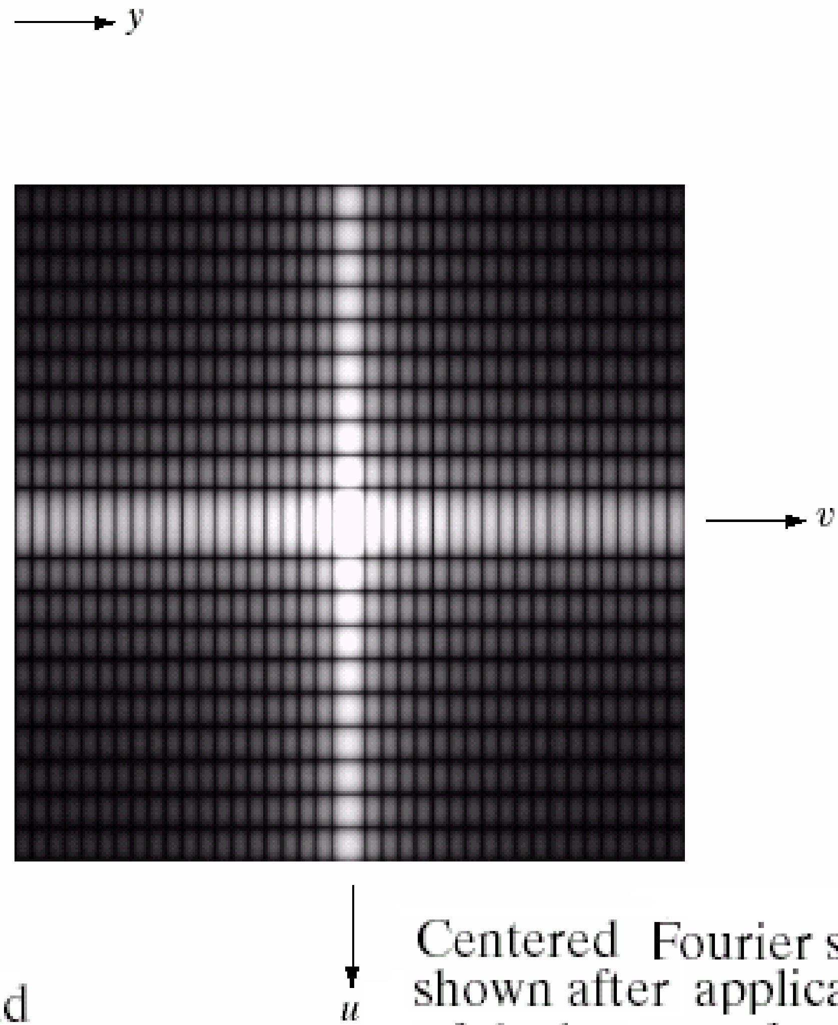
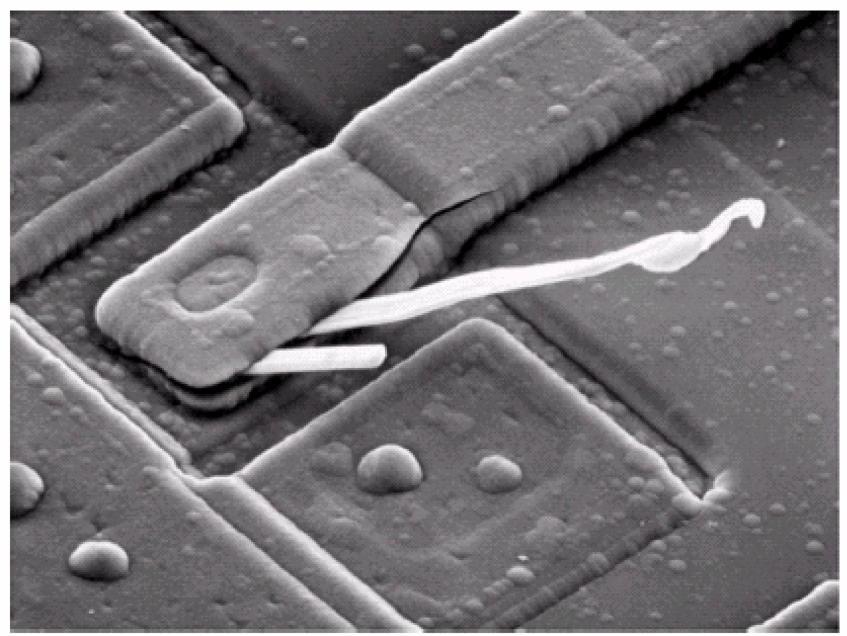


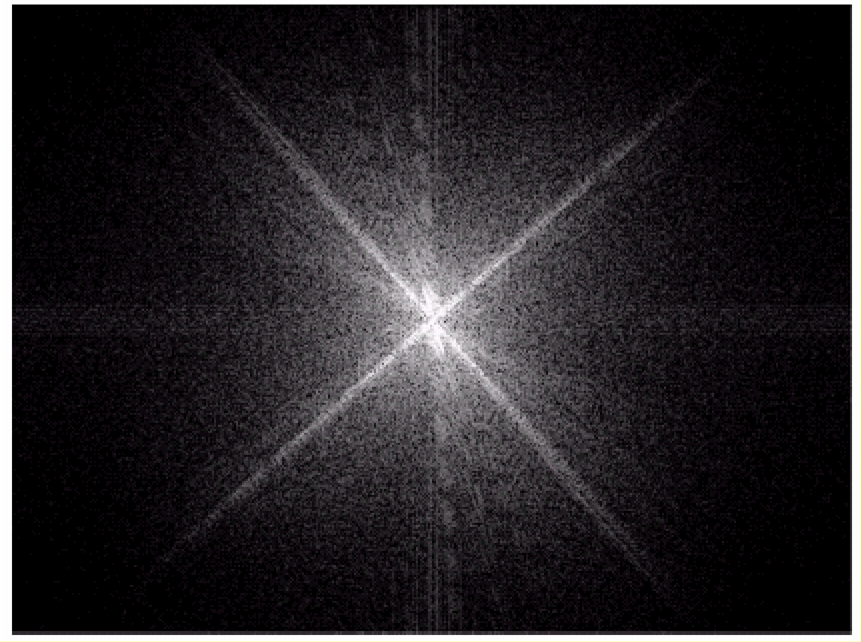
Image of a 20×40 white rectangle on a black background of size 512×512 pixels.



Centered Fourier spectrum shown after application of the log transformation

The Fourier spectrum – in 2D



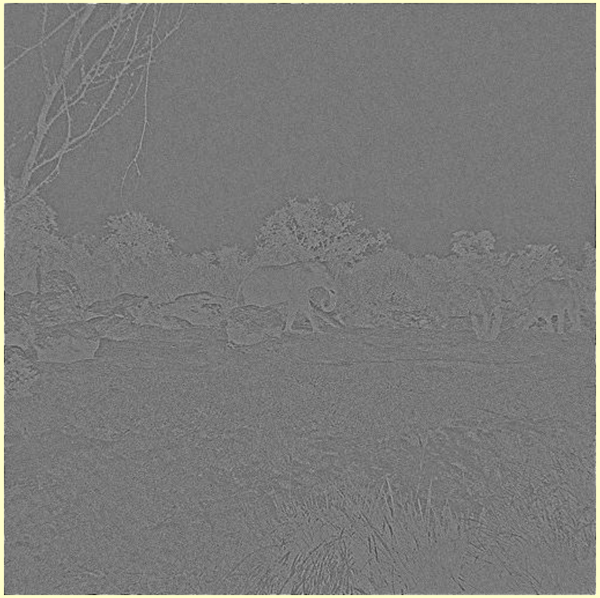


SEM image of a damaged integrated circuit. Fourier spectrum of .

- The original image contains two principal features: edges run approximately at ±45° .
- The Fourier spectrum shows prominent components in the same directions.

Phase Data Images.


a) Original image


b) Phase only image


c) Contrast enhanced version of image (b) to show detail

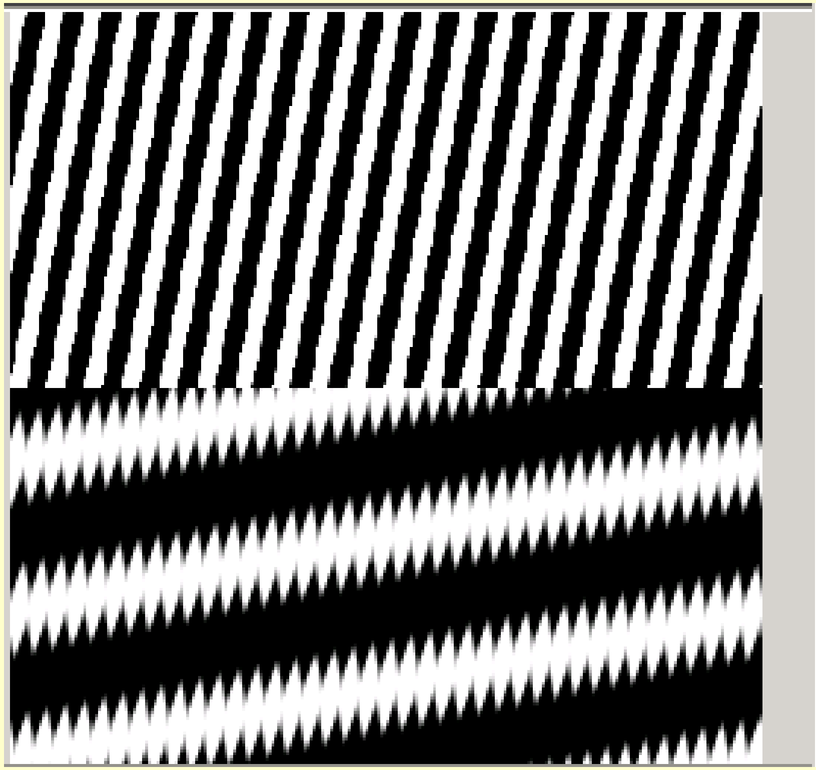
- ✓ **Phase data contains information about where objects are in the image,i.e. it holds spatial information.**
- ✓ **Fourier transforms do not provide simultaneously frequency as well as Spatial information.**

Fourier Spectrum in MATLAB

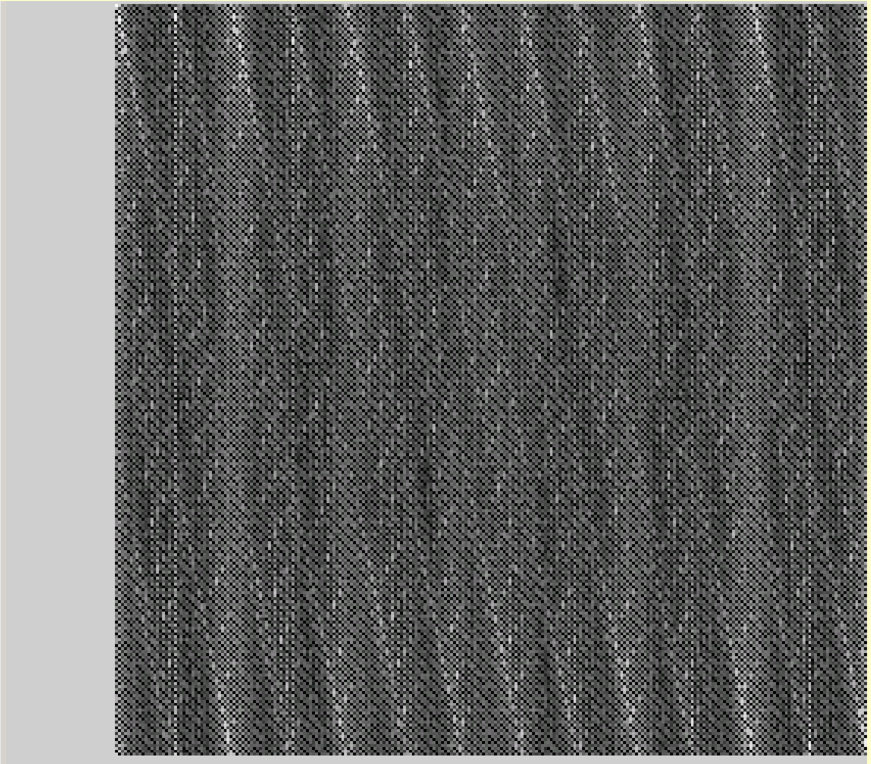
The MATLAB Fourier transform of an image c is obtained by: fft2(c).

e.g.

```
clear all
c=imread('SinoPattern.bmp');
F=fft2(c);        // fft2(C)=fft(fft(
S=abs(F);
L=log(1+double(S));  //To be able to display an image of
imShow(L, []);
```



a) Original image

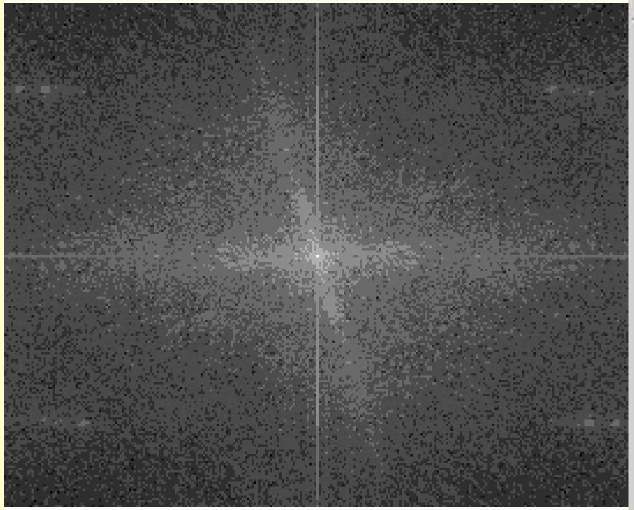


Log enhanced version of Fourier Spectrum

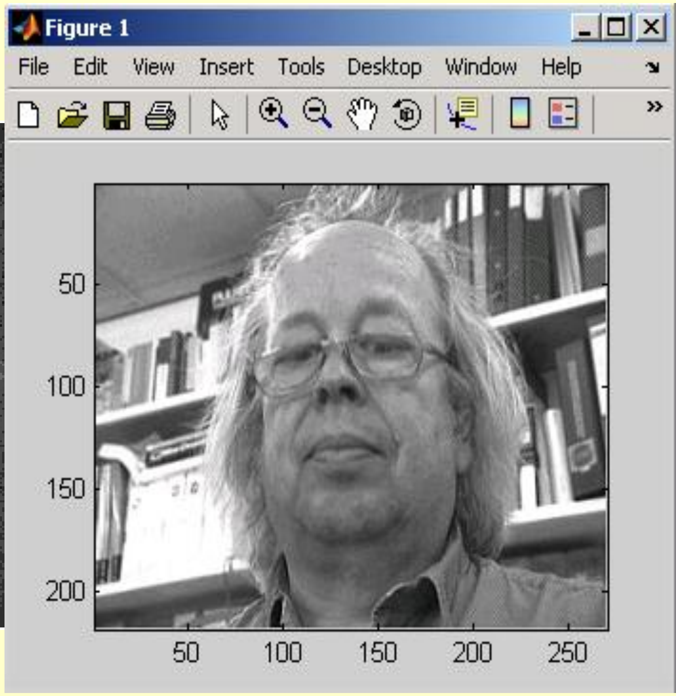
Fourier transform in MATLAB

- ✓ **Using F=fftshift(fft2(c)) in stead of F=fft2(c) in the last programme creates the centered Fourier Transform of c.**
- ✓ **The MATLAB statement ifft2(F) is used to invert the Fourier transform of the image c, where F = fft2(c).**

Original image c



Log enhanced version of Fourier Spectrum



Inverse Fourier of fft2(c)

Filtering in the Frequency Domain

- ✓ **Filtering in the frequency domain aims to enhance an image through modifying the its DFT. Thus, there is a need for an appropriate filter function H(u,v).**
- ✓ **The filtering of an image f(x,y) works in 4 steps:**
  1. **Compute the centred DFT, $F(u,v) = \Im((-1)^{x+y} f(x,y))$.**
  2. **Compute $G(u,v) = F(u,v)H(u,v)$.**
  3. **Compute the inverse DFT of G(u,v), $\Im^{-1}(G(u,v))$.**
  4. **Obtain the real part of $\Im^{-1}(G(u,v))$.**
  5. **Compute the filtered image $g(x,y) = (-1)^{x+y} R(\Im^{-1}(G(u,v)))$.**
- ✓ **Generally, the inverse DFT is a complex-valued function. However, when f(x,y) is real then the imaginary part of the inverse DFT vanishes. Therefore for images step 4, above, doesn't apply.**

Filtering in the Frequency Domain – Scheme

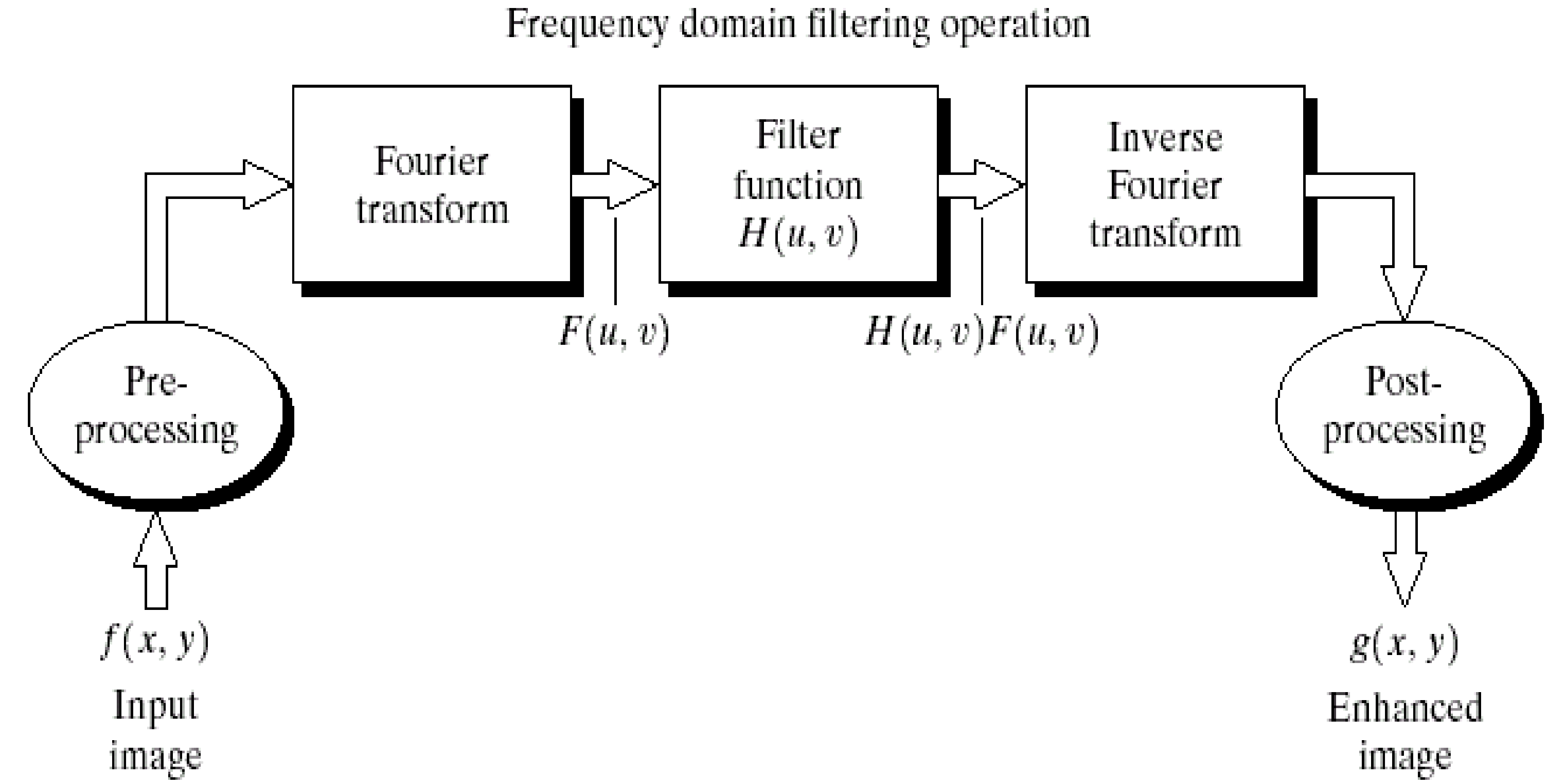


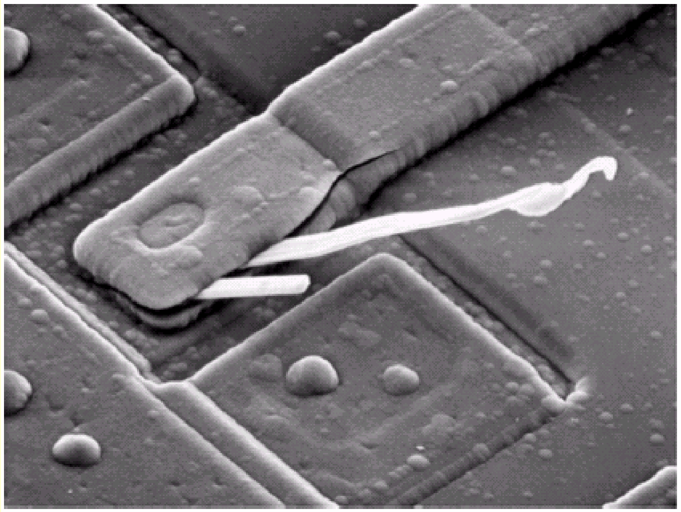
FIGURE 4.5 Basic steps for filtering in the frequency domain.

The Notch filter

- ✓ A simple filter that forces the average image value to become 0.
- ✓ The average value of an image f(x,y) is the DC component of the DFT spectrum i.e. F(0,0). The Notch filter is defined as follows:

$$H(u,v)=\begin{cases}0 & \text{if } (u,v)=(M/2,N/2)\\ 1 & \text{otherwise.}\end{cases}$$

Original image



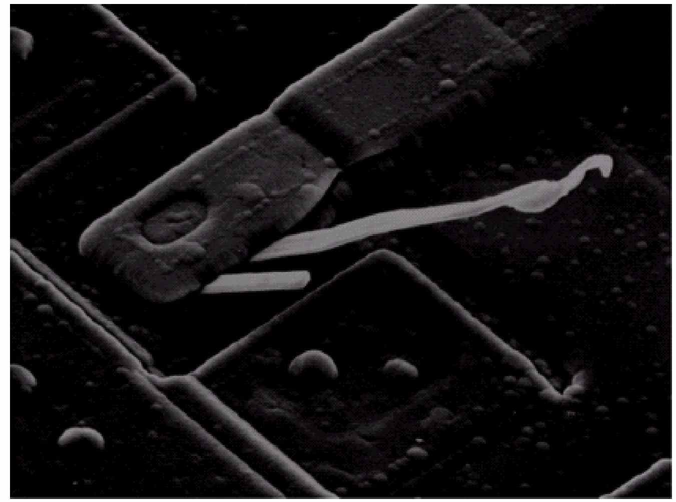


Image after Notch filter application

- ✓ Note that the edges stand out more than before filtering.
- ✓ When the average value is 0, some values of the filtered image are negative, but for display purposes pixel values are shifted.

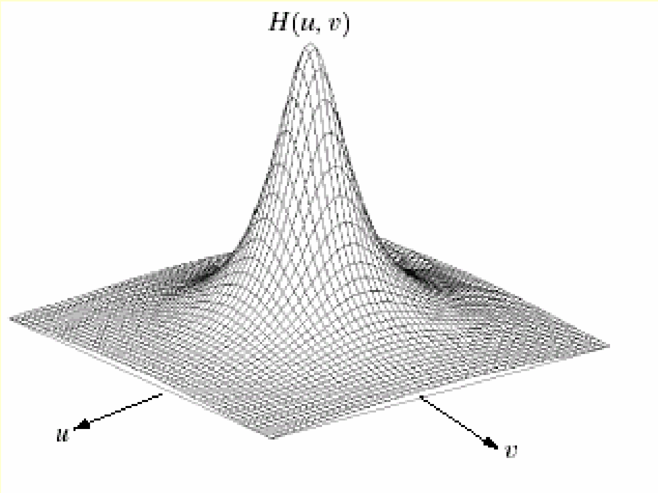
Lowpass and highpass filtering

- ✓ **Low frequencies in the DFT spectrum correspond to image values over smooth areas, while high frequencies correspond to detailed features such as edges & noise.**
- ✓ **A filter that suppresses high frequencies but passes low frequencies is called Lowpass filter, while filters that act to reduce the low frequencies but passes high ones are called Highpass filters.**
- ✓ **Examples of such filters are obtained from circular Gaussian functions of 2 variables (see next slide)**

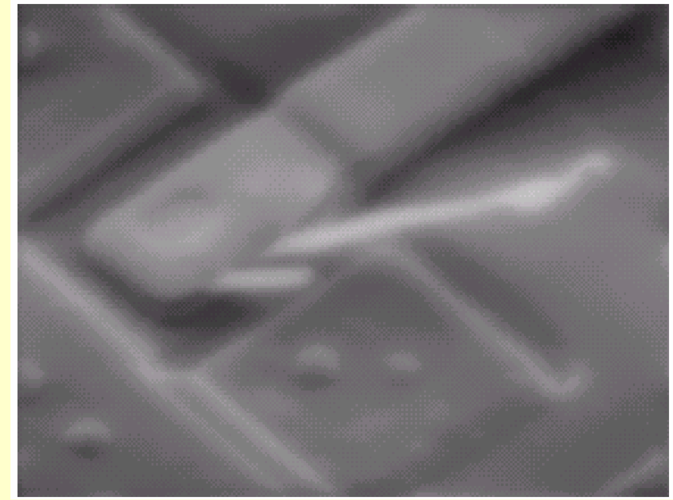
$$H(u,v) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/2\sigma^2}, \qquad \text{- Lowpass filter,}$$

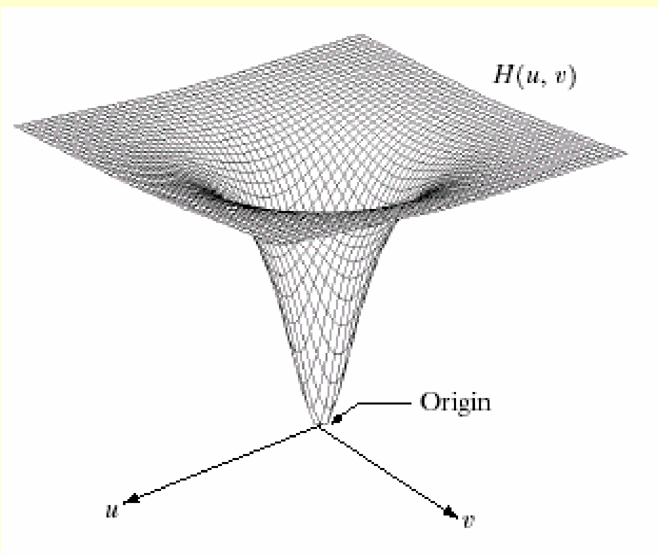
$$H(u,v) = \frac{1}{2\pi\sigma^2} (1 - e^{-(u^2+v^2)/2\sigma^2}) \qquad \text{- Highpass filter.}$$

Low-pass & High-pass filtering - Example

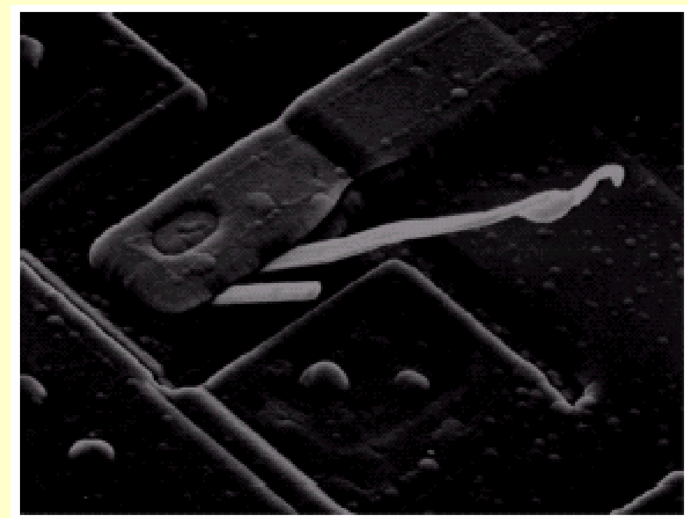


Low pass filtering





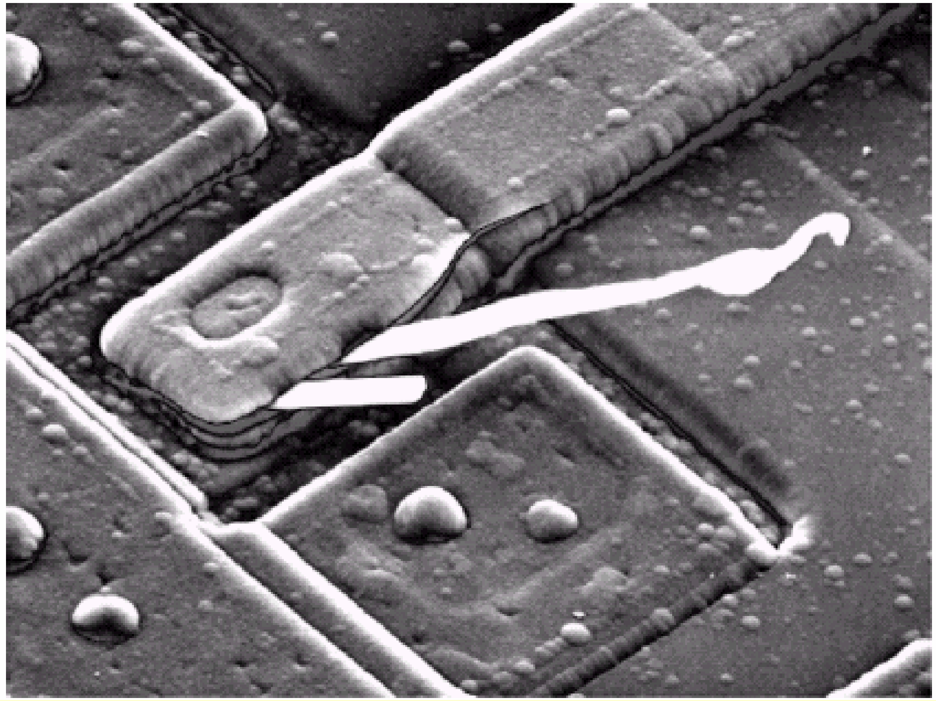
High pass filtering



Low pass filtering results in blurring effects, while High pass filtering results in sharper edges.

High-pass filtering –slight modification

- ✓ ***In the last example, the highpass filtered image has little smooth gray-level detail as a result of setting F(0,0) to 0. This can be improved by adding a constant.***
- ✓ ***Here we added 0.75/($\pi\sigma^2$) to the previous high-pass filter.***



Filtering in the Spatial and Frequency domains

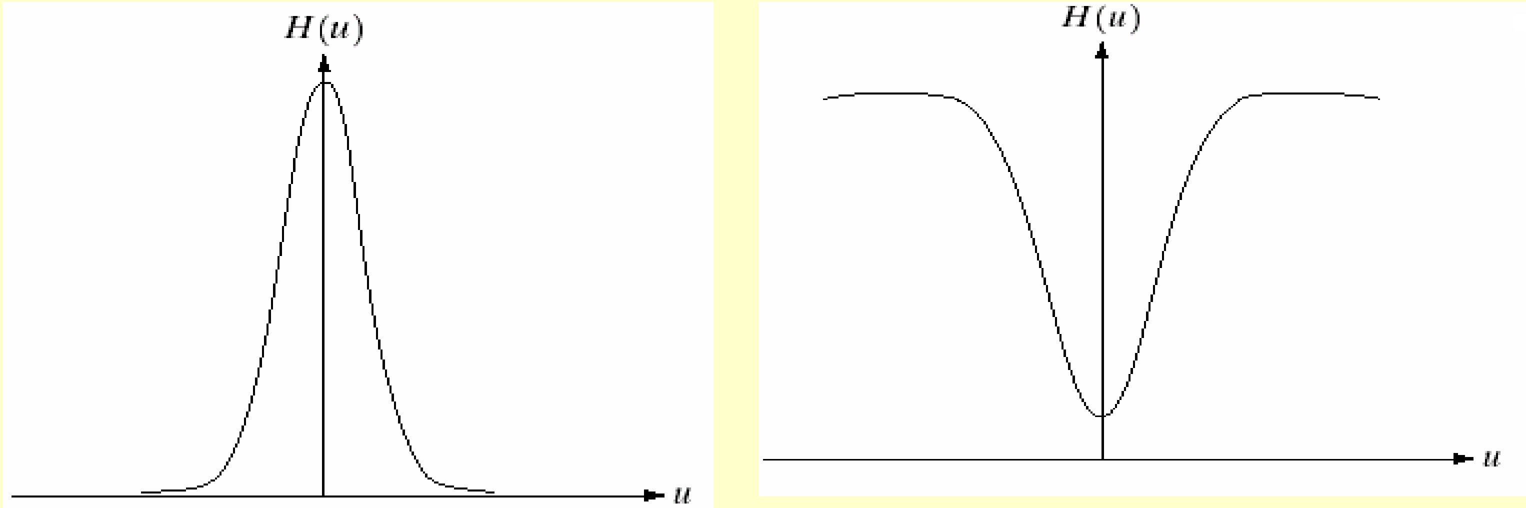
- ✓ **Spatial filters are linked to, and often obtained from, filters in the frequency domain.**
- ✓ **The *Convolution Theorem* links the spatial domain to the frequency domain.**
- ✓ **The discrete convolution of f(x,y) and h(x,y) is defined as:**

$$f(x,y)*h(x,y)=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\,h(x-m,y-n).$$

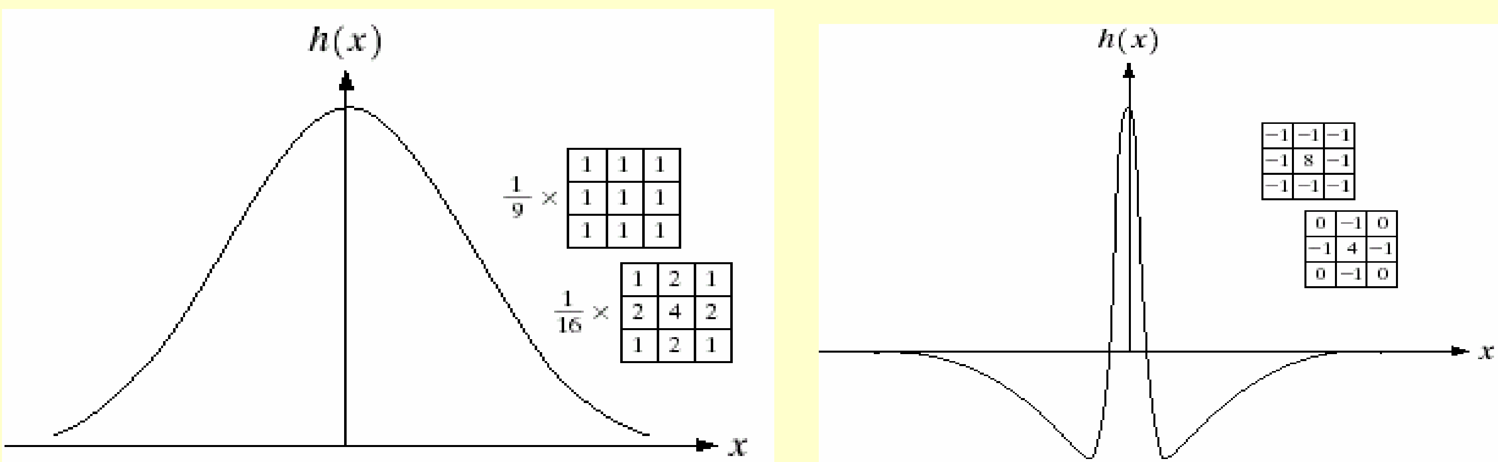
The convolution theorem states if $F(u,v)$ and $\mathrm{H(u,v)}$ are the Fourier transforms of $\mathrm{f(x,y)}$ and $\mathrm{h(x,y)}$, respectivaly, then:

$$f(x,y)*h(x,y)=\Im^{-1}(F(u,v)H(u,v)).$$

Filtering in the Spatial & Frequency -Example



Low pass and High pass Filters in the Frequency domain



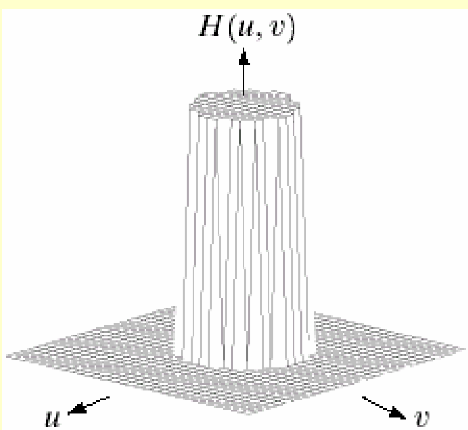
The corresponding Filters in the spatial domain.

Smoothing Frequency Domain Filters

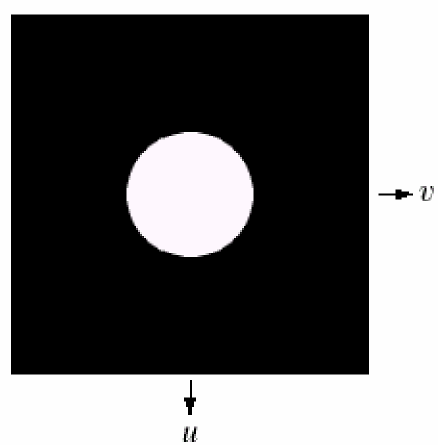
✓ **The Ideal Low-pass Filter is the simplest lowpass filter that "cuts off" all high frequency component of the DFT that are at a certain distance from the centre of the DFT.**

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

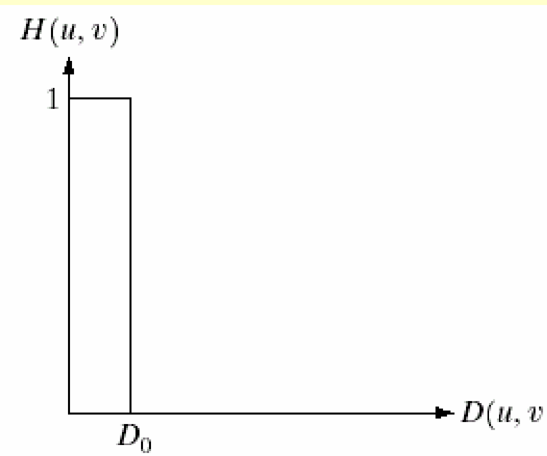
In this case D_0 is the cutoff frequency, and $D(u,v) = [(u-M/2)^2+(v- N/2)^2]^{1/2}$



The Ideal Lowpass filter

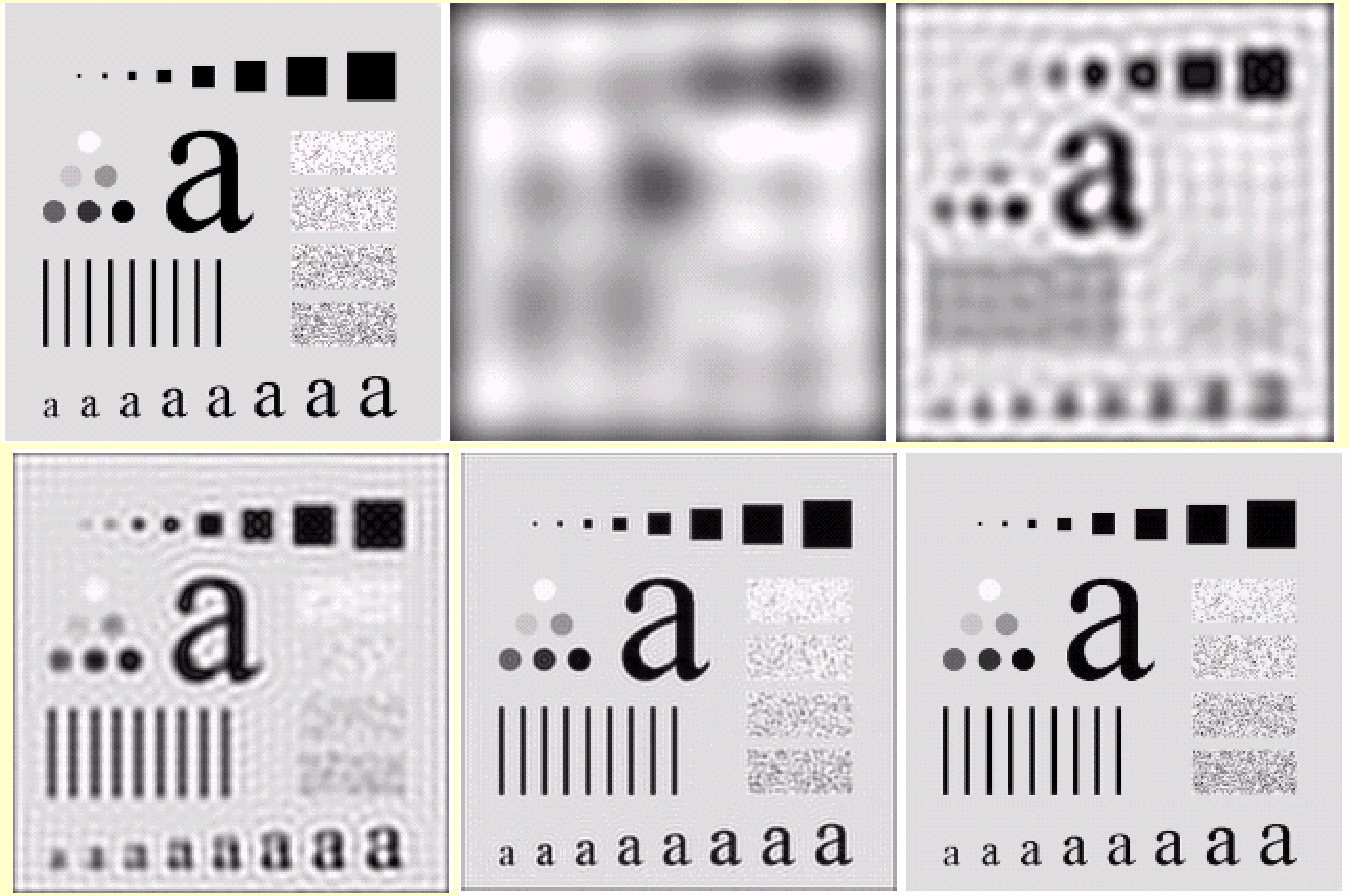


The ILPF as an image



The ILPF radial cross section

ILPF pass filter – with different cutoff levels

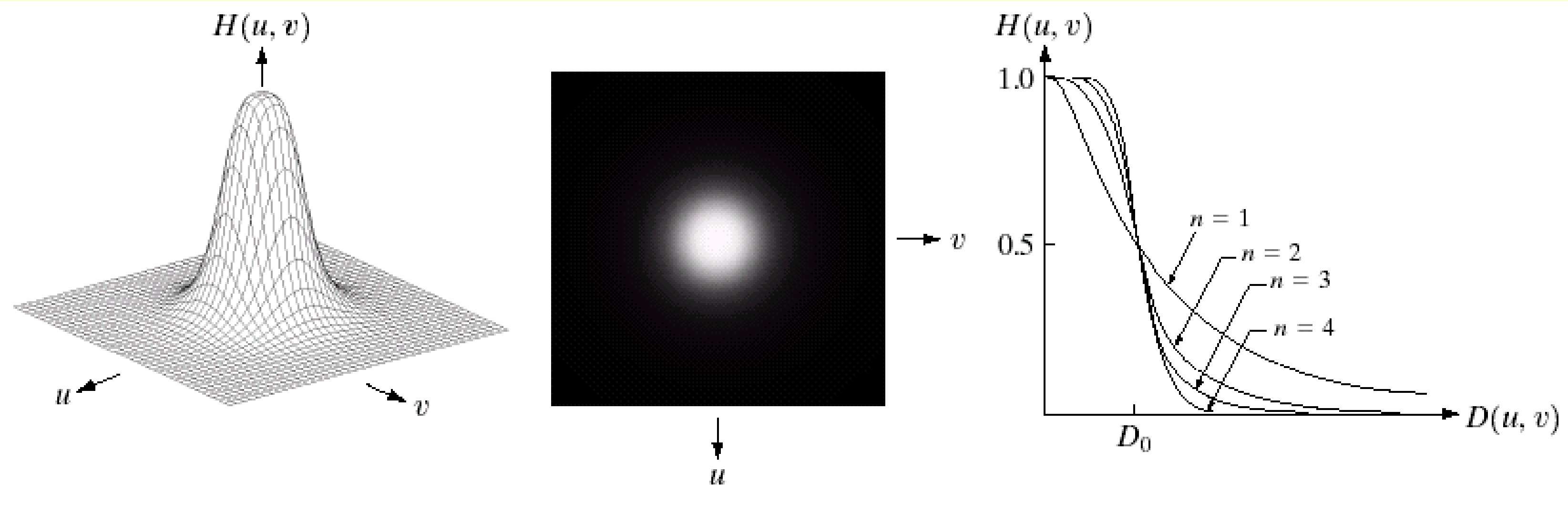


ILPF filtering: Cutoff frequencies at radii of 5, 15,30,80, and 230.

The Butterworth Lowpass Filter

- ✓ **The Butterworth Lowpass Filter (BLPF) of order n and with cutoff frequency at distance *D_0 is* defined as:**

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

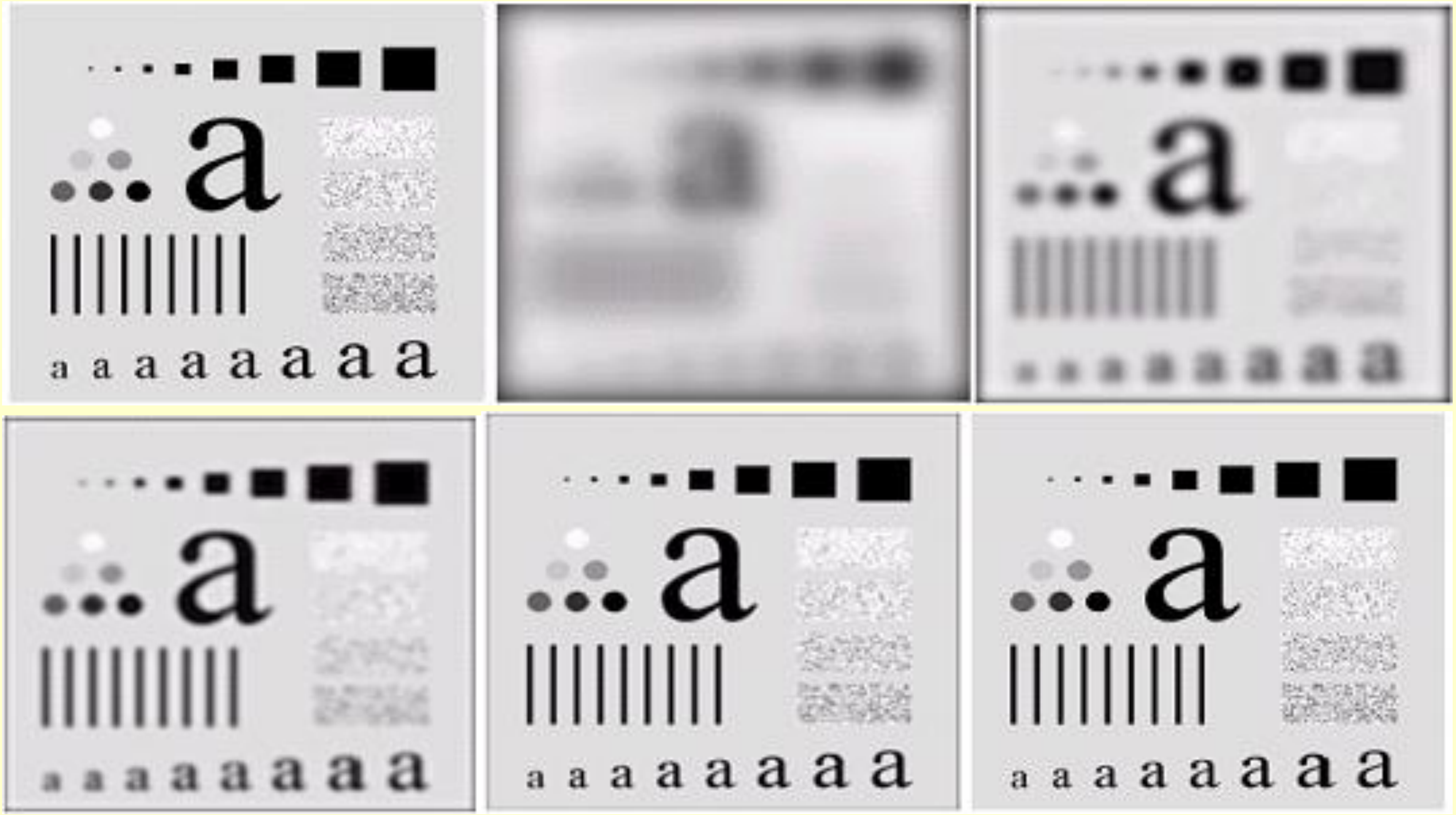


The BLPF Lowpass filter

The BLPF as an image

The BLPF radial cross section

BLPF filter – with different cutoff levels

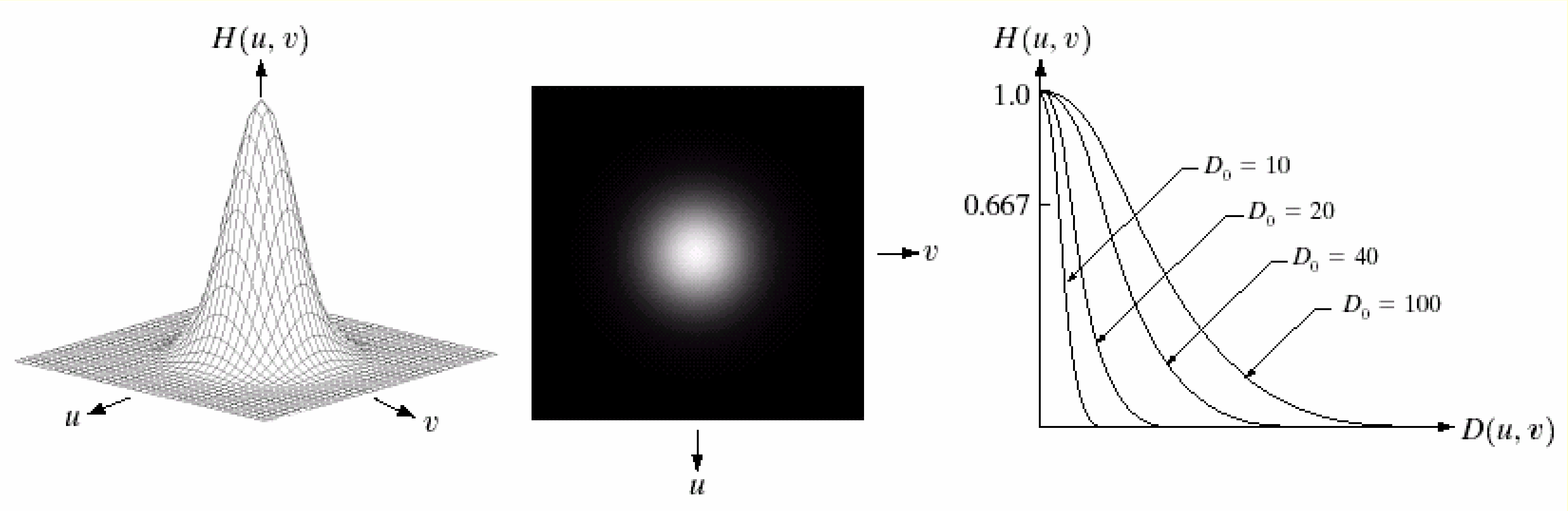


BLPF of order 2, Cutoff frequencies at radii of 5, 15,30,80, and 230.

The Gaussian Lowpass Filters

- ✓ **The Gaussian Lowpass Filter (GLPF) with cutoff frequency at distance D_0 *is* defined as:**

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} .$$

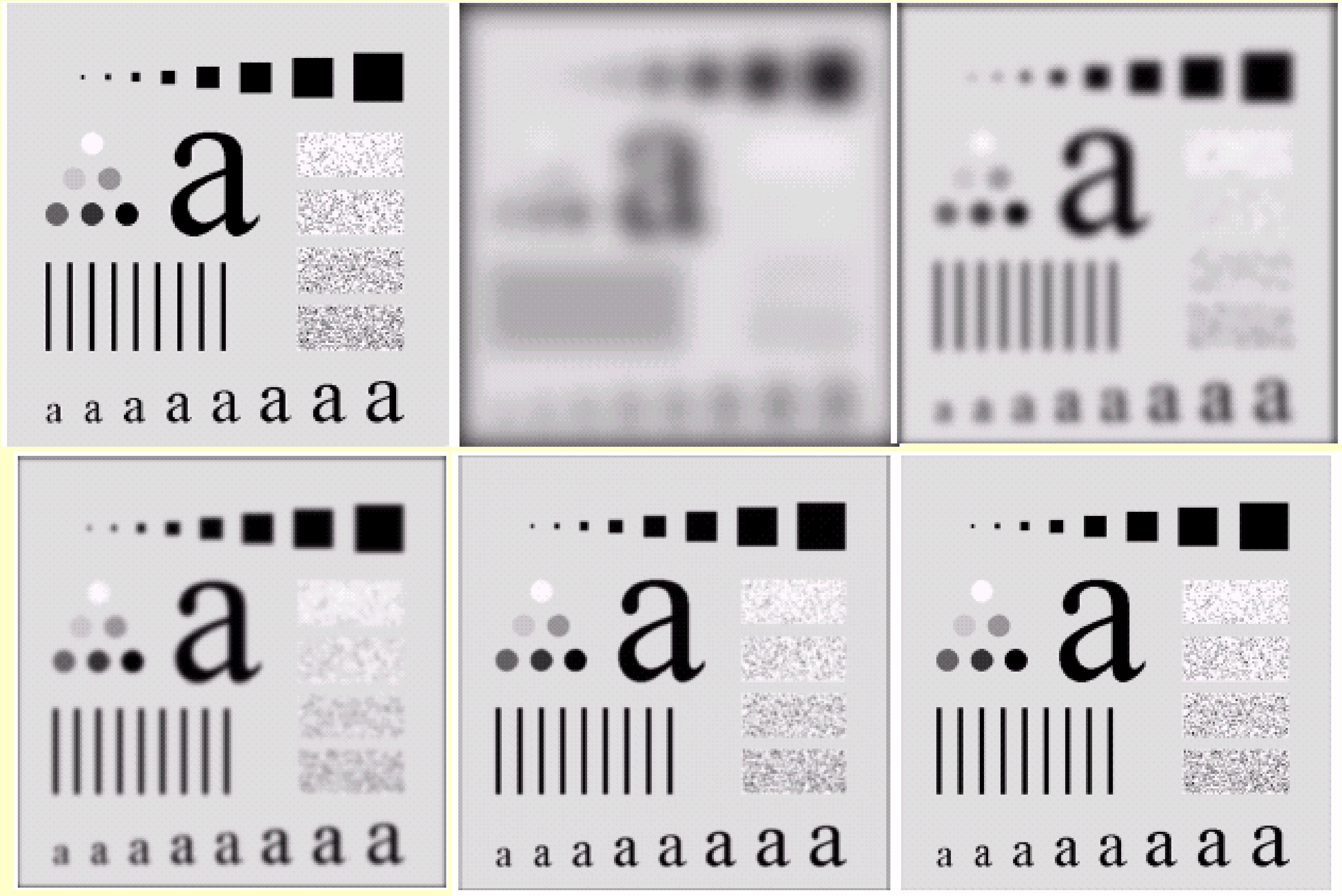


The GLPF Lowpass filter

The GLPF as an image

The GLPF radial cross section

GLPF filter – with different cutoff levels

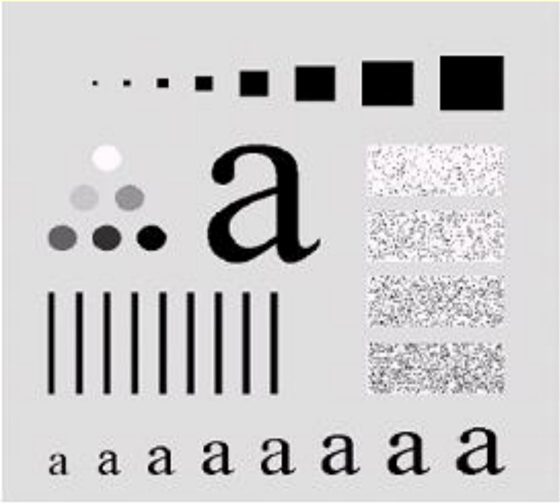


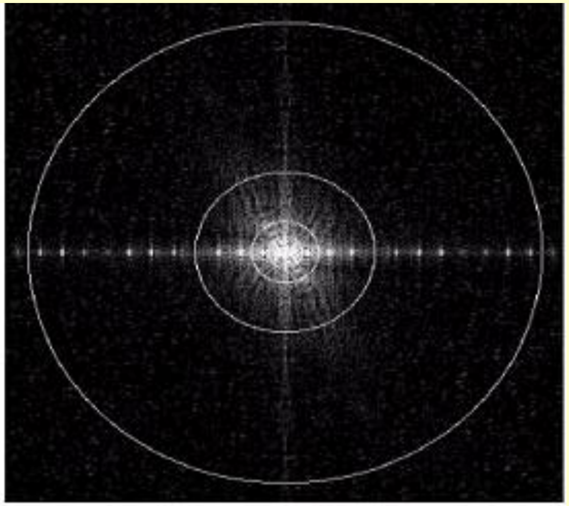
Cutoff frequencies at radii of 5, 15,30,80, and 230.

Effect of Filtering on Image Quality

- ✓ For the three types of filters, the severity of image degredation decreases as the cutoff radii increase, but the type of degredation is filter dependent.
- ✓ The cutoff radius in a lowpass filter, is the raduis of a circle centred at the origin of the Furiour Spectrum of the image.
- ✓ The power spectrum is the square of the Fourier spectrum,

  i.e. $P(u,v) = (Re(F(u,v)))^2 + (Im(F(u,v)))^2$.
- ✓ The power enclosed by the cutoff radius is the % of the sum of power values within the circle to the total image power.



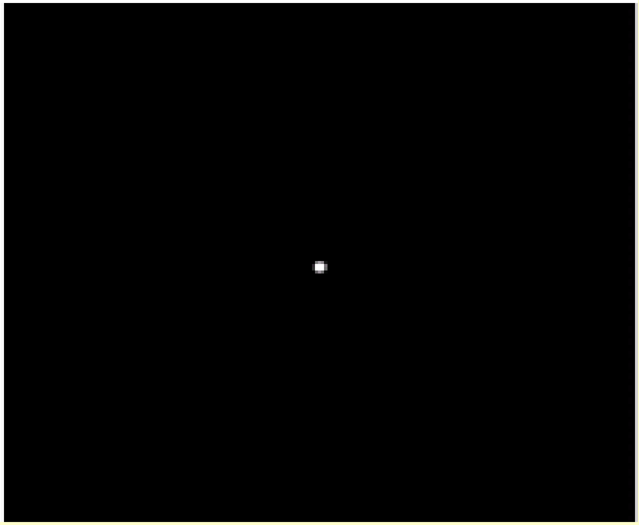


Cutoff radii of 5,15,30, 80 and 230 enclose 92%, 94.6%, 96.4%, 98%, and 99.5% of the image power.

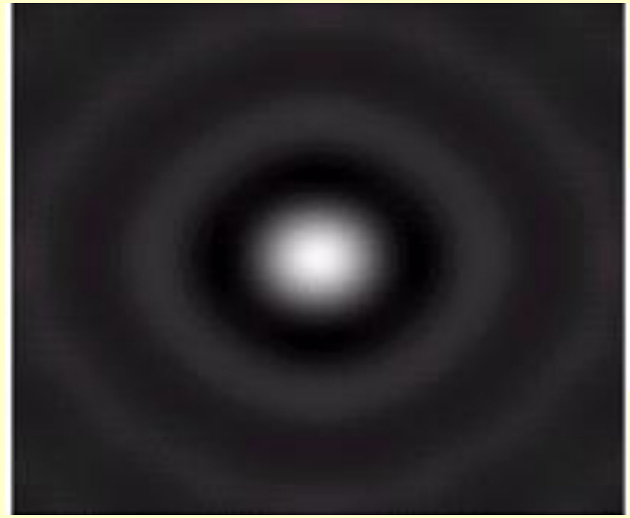
Effect of Filtering on Image Quality - coninued

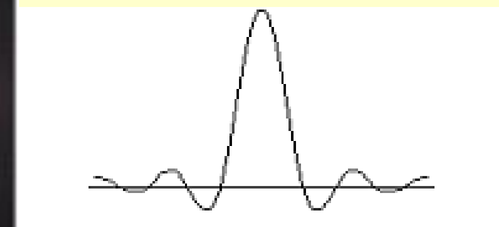
- ✓ **The three lowpass filters result in blurring effect that decreases with as the cutoff radius increases, with GLPF being the best and ILPF being the worst.**
- ✓ **ILPF filter also results in ringing effect which decreases as the cutoff raduis increases. It remains evident even for relatively large radii.**
- ✓ **The ringing effect is not present in order 1 BLPF, and is imperceptable in order 2 BLPF. However, BLPF of higher order do have visible ringing effect.**
- ✓ **GLPF does not have ringing effect.**
- ✓ **The ringing effect of lowpass filters can be explained in terms of the convolution theorem which links convolution in the spatial domain with filtering in the frequency domain. (see next slide).**

Ringing Effects of ILPF - Illustrated



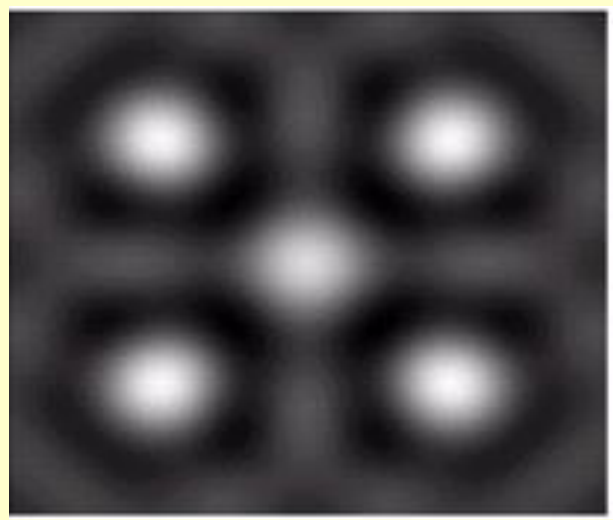
(a) Freq. domain ILPF rad=5

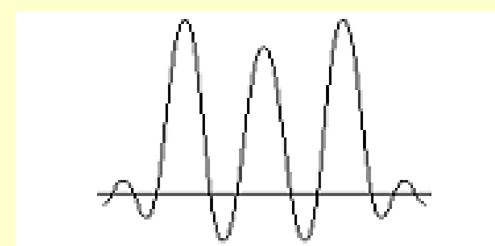




The corresponding spatial domain filter

(b) Five Impulse image





Convolution of (a) and (b) in the spatial domain

Sharpening Frequency Domain Filters

- **Edges and suden changes in Gray levels are associated with high frequencies. Thus to enhance and sharpen significant details we need to use highpass filters n the frequency domain**
- **The objectives of using highpass filters are the reverse of those for using lowpass filters.**
- **For any lowpass filter $H_{lp}(u,v)$ there is a highpass filter:**

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

- **Thus we have an Ideal highpass filter (IHPL), a Butterworth highpass filter, and a Gaussian High pass filter'**

Sharpening Frequency Domain Filters

- ✓ **Edges and suden Gray level changes are associated with high frequencies. Thus sharpening images can be acheived by highpass frequency domain filters.**
- ✓ **The objectives of using highpass filters are the reverse of those for using lowpass filters.**
- ✓ **For any lowpass filter $H_{lp}(u,v)$ there is a highpass filter:**

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

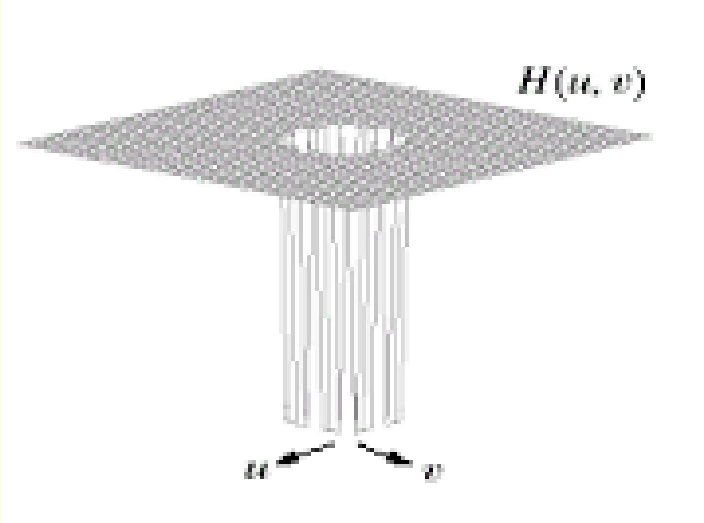
- ✓ **Thus we can define an Ideal Highpass filter, a Butterworth High frequency filter, and a Gaussian Highpass filter.**

Sharpening Frequency Domain Filters

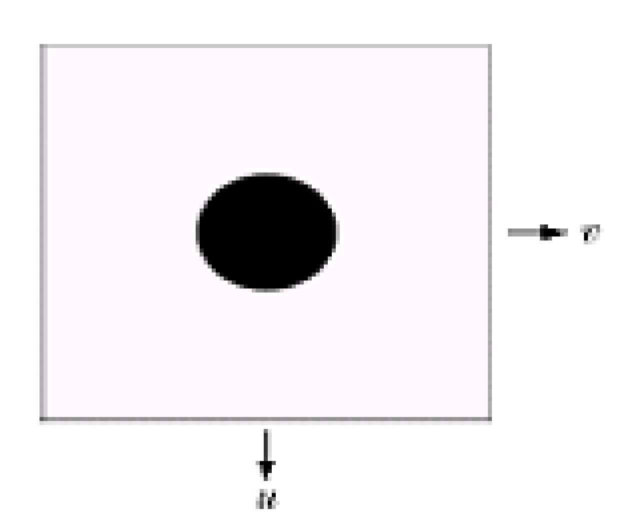
- ✓ **The Ideal Highpass Filter "cutsoff" all low frequencies of the DFT but maintain the high ones that are within a certain distance from the centre of the DFT.**

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

In this case D_0 is the cutoff frequency, and $D(u,v) = [(u-M/2)^2+(v- N/2)^2]^{1/2}$

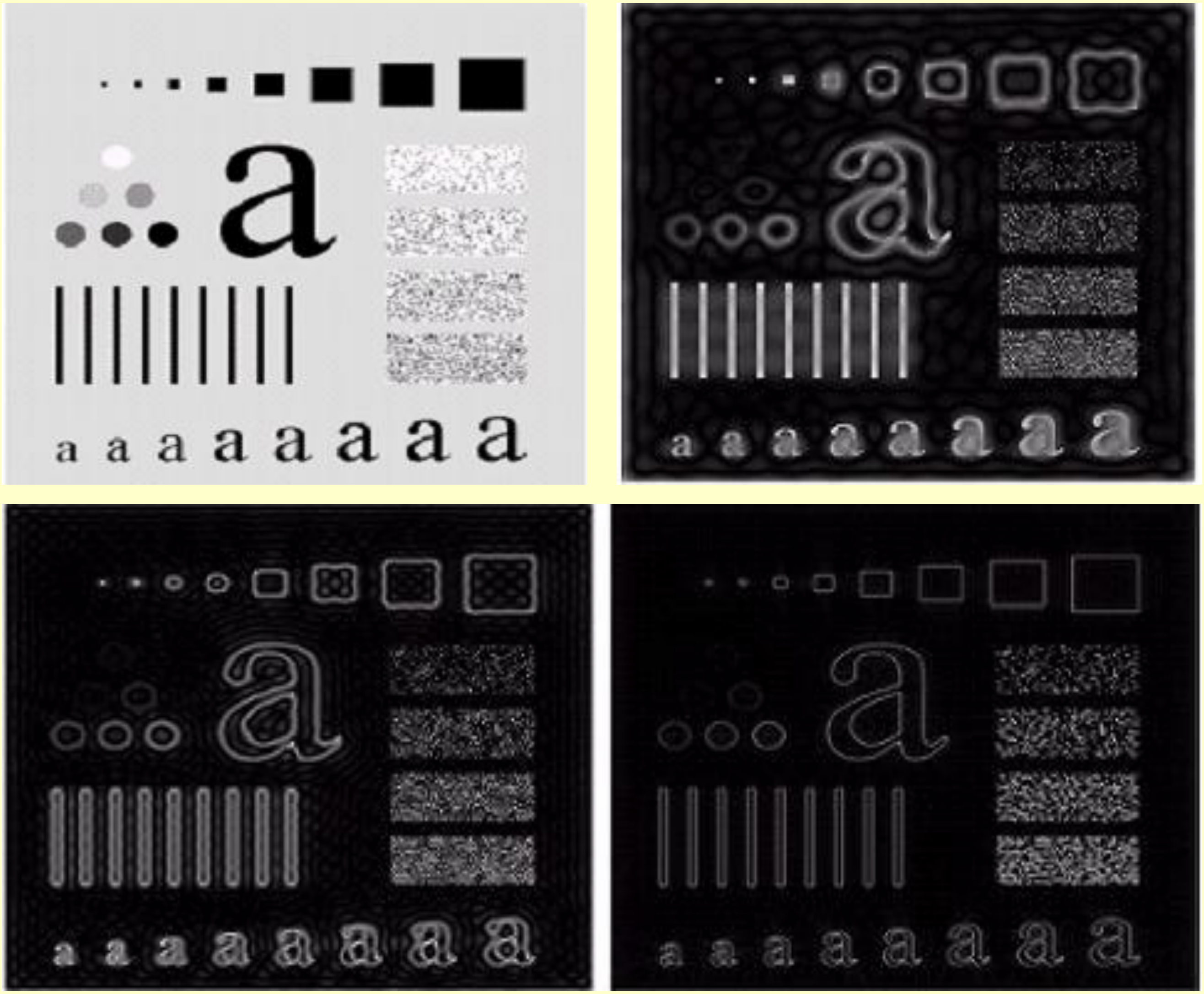


The IHPF filter



The IHPF as an image

IHPF pass filter – with different cutoff levels

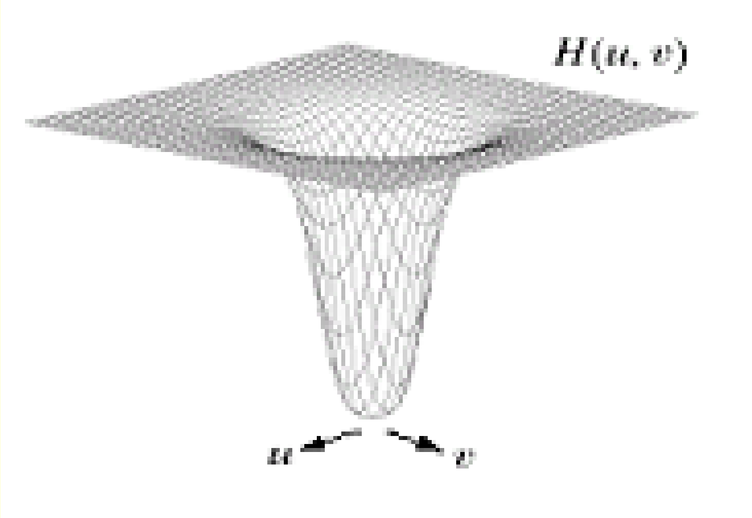


IHPF filtering: Cutoff frequencies at radii 15,30,80. Ringing is visible

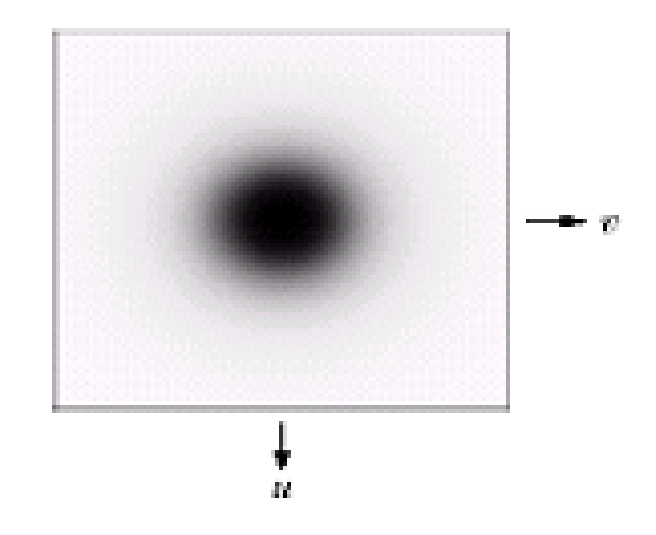
The Butterworth High pass Filter

- **The Butterworth Highpass Filter (BHPF) of order n and with cutoff frequency at distance *D_0 is* defined as:**

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

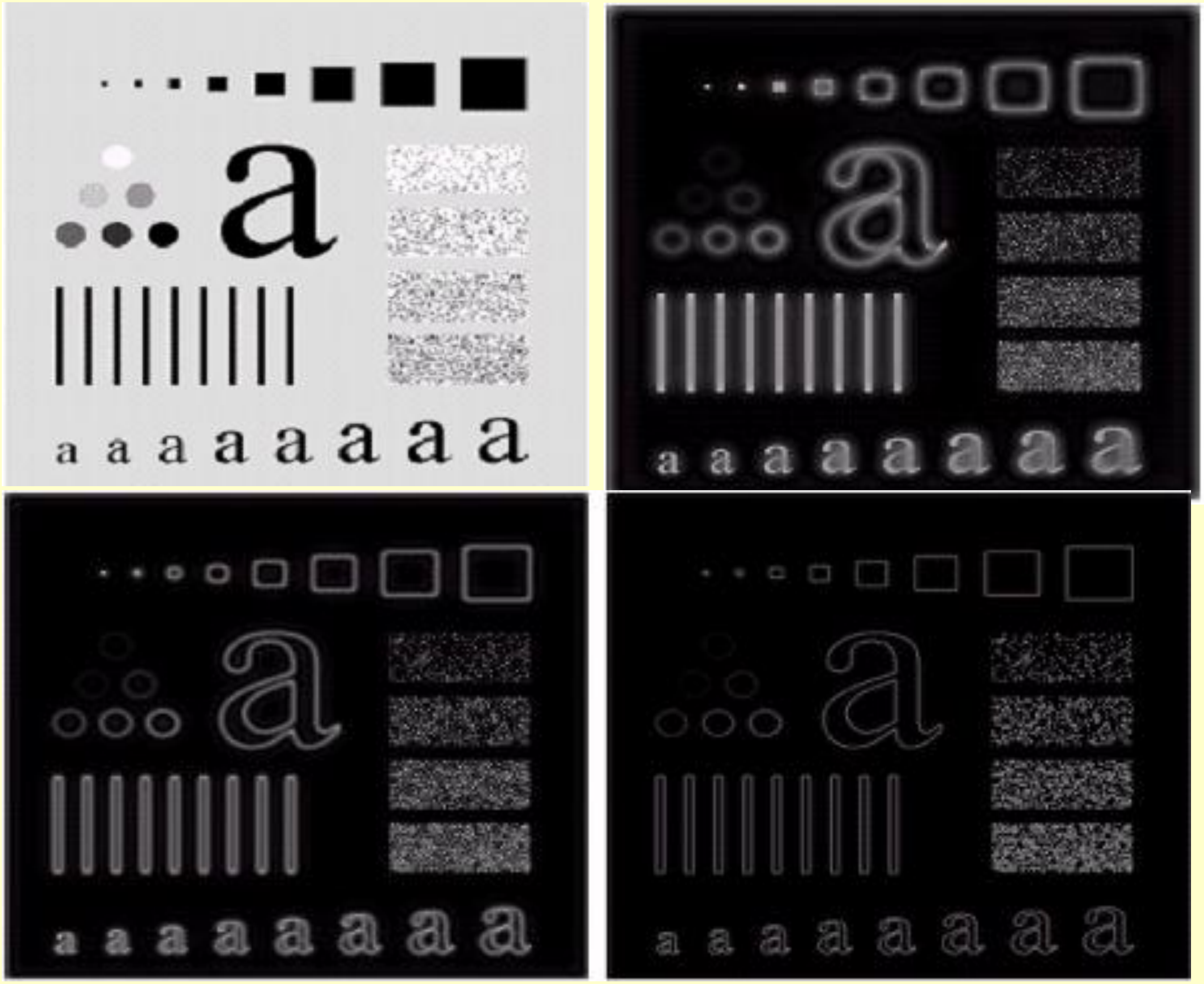


The BHPF



The BHPF as an image

BHPF filter – with different cutoff levels



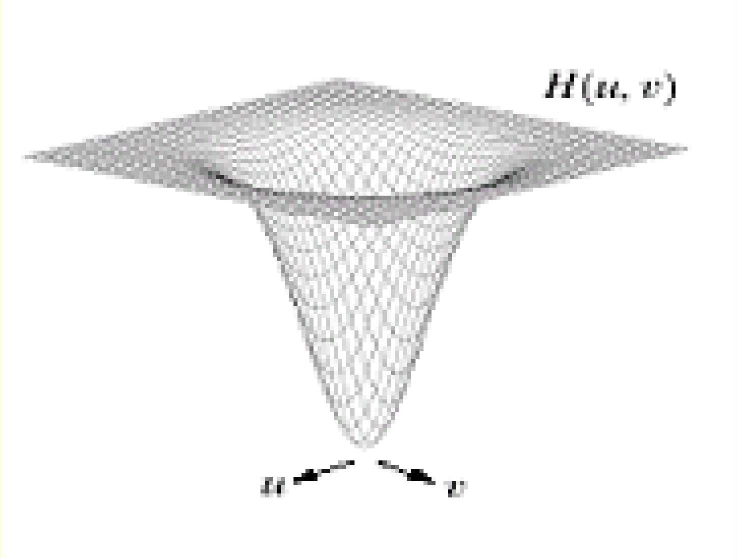
BHPF of order 2, Cutoff frequencies at radii of 15,30, and 80.

The boundaries are less distorted than with the IHPF.

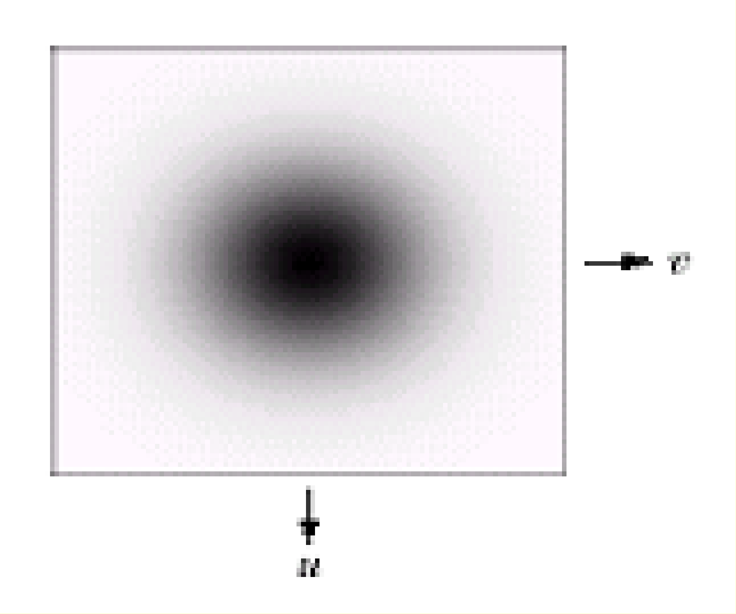
The Gaussian Highpass Filters

✓ **The Gaussian Highpass Filter (GHPF) with cutoff frequency at distance *D_0* *is* defined as:**

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \; .$$

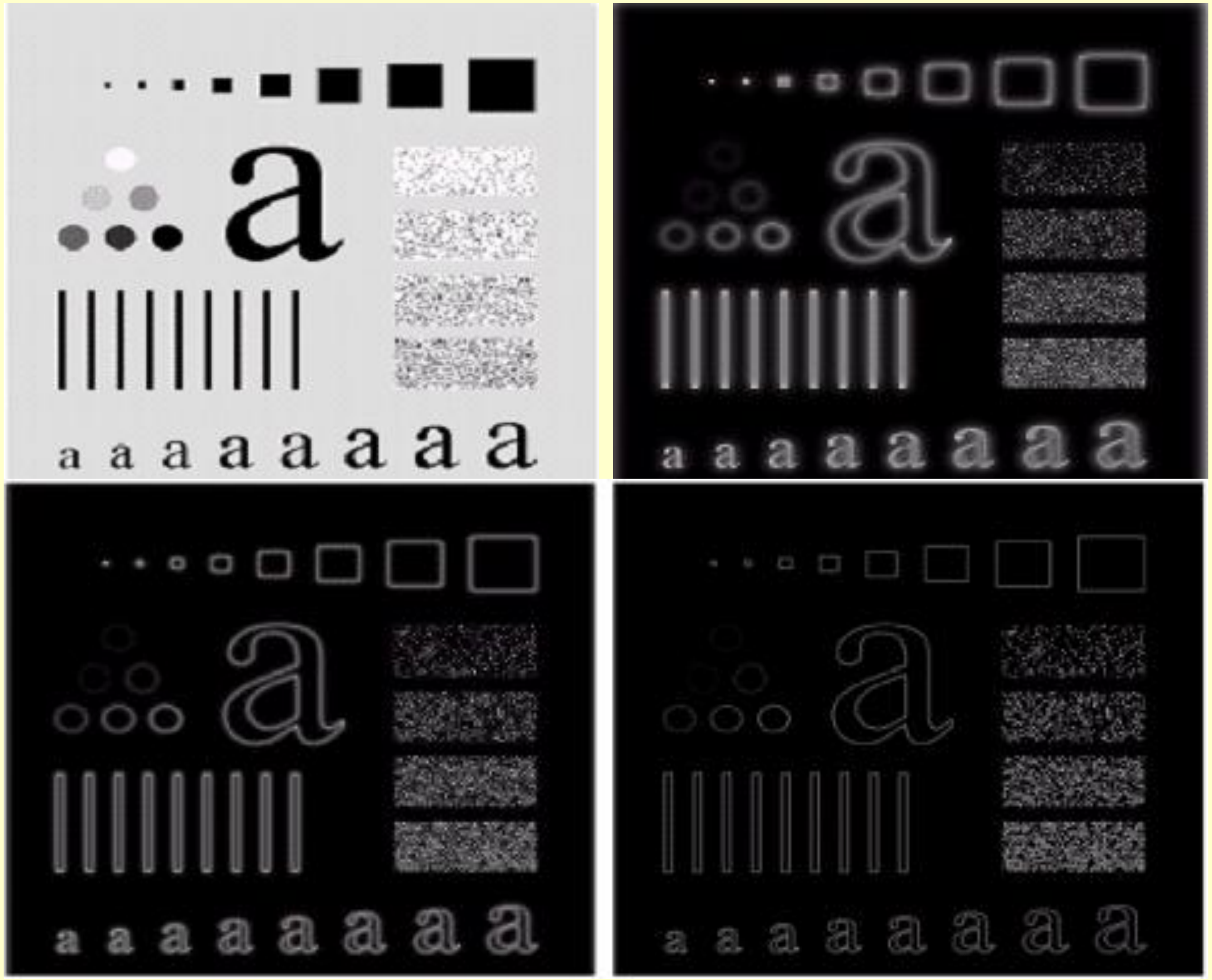


The GHPF filter



The GHPF as an image

GHPF filter – with different cutoff levels



GHPF filter : Cutoff frequencies at radii of 15,30, and 80.

The results are smoother than those obtained by the other 2 filters

Other filters

- ✓ **It is possible to construct Highpass filters as the difference of two Gaussian lowpass filters.**
- ✓ **The Laplacian can be implemented in the frequency domain as the filter:**

  $$H(u,v) = -(u^2+v^2).$$

  This follows from the fact that:

  $$\Im\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] = (iu)^2 F(u,v) + (iv)^2 F(u,v)$$

  $$= -(u^2 + v^2) F(u,v).$$

- ✓ **It is customary to use the centered version, i.e.**

  $$H(u,v) = -\left[(u-M/2)^2+(v-N/2)^2\right].$$

  The Laplacian filtered image in the spatial domain is obtained by computing the inverse DFT of H(u,v)F(u,v).

End of Chapter 5