IMAGE PROCESSING - OVERVIEW

MTSC 887 – Sokratis Makrogiannis

Delaware State University

Lecture Outline

- Goal: Overview of Digital Image Processing
- Topics:
  - Scope of image processing field
  - State of the art in digital image processing
  - Outline of main image processing stages
  - Components of a system
  - Applications

What is Digital Image Processing?

- Answer: Manipulation of digital images by computer.
- Image processing focusing on two main tasks
  - Improvement of pictorial information for human interpretation and high level processing
  - Processing of imaging data for storage and transmission.

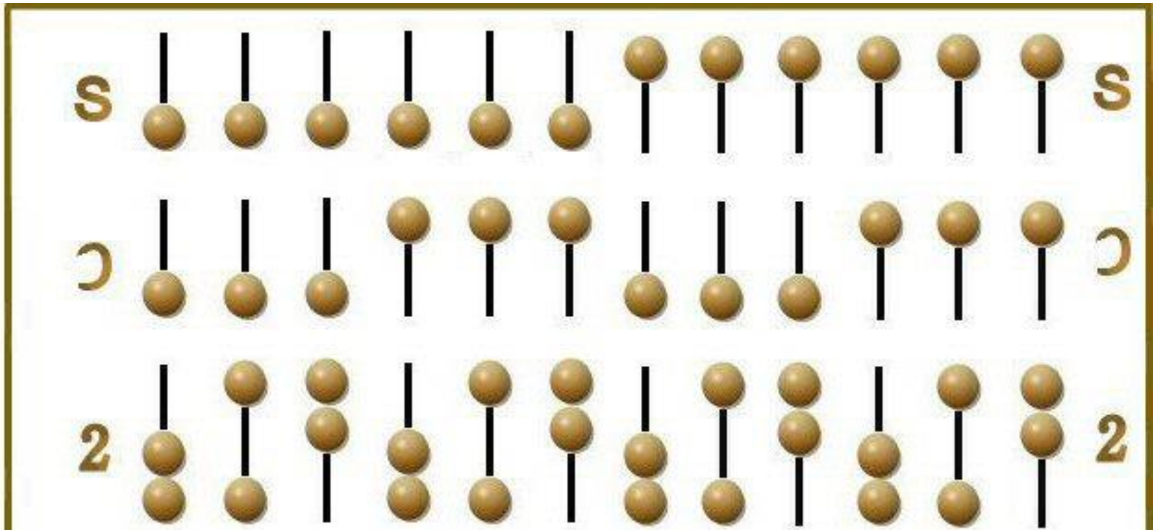
Origins of Digital Image Processing

- Telecommunications and more specifically transmission of pictorial content.
- Technological breakthroughs in electronics, digital computers and programming languages, led to the foundation of image processing.
- Early computer techniques aimed to correct distortions in images of the moon transmitted by Ranger 7 (Jet Propulsion Lab, 1964).

FIGURE 1.4 The first picture of the moon by a U.S. spacecraft. *Ranger 7* took this image on July 31, 1964 at 9 : 09 A.M. EDT, about 17 minutes before impacting the lunar surface. (Courtesy of NASA.)

Computing Machinery Origins

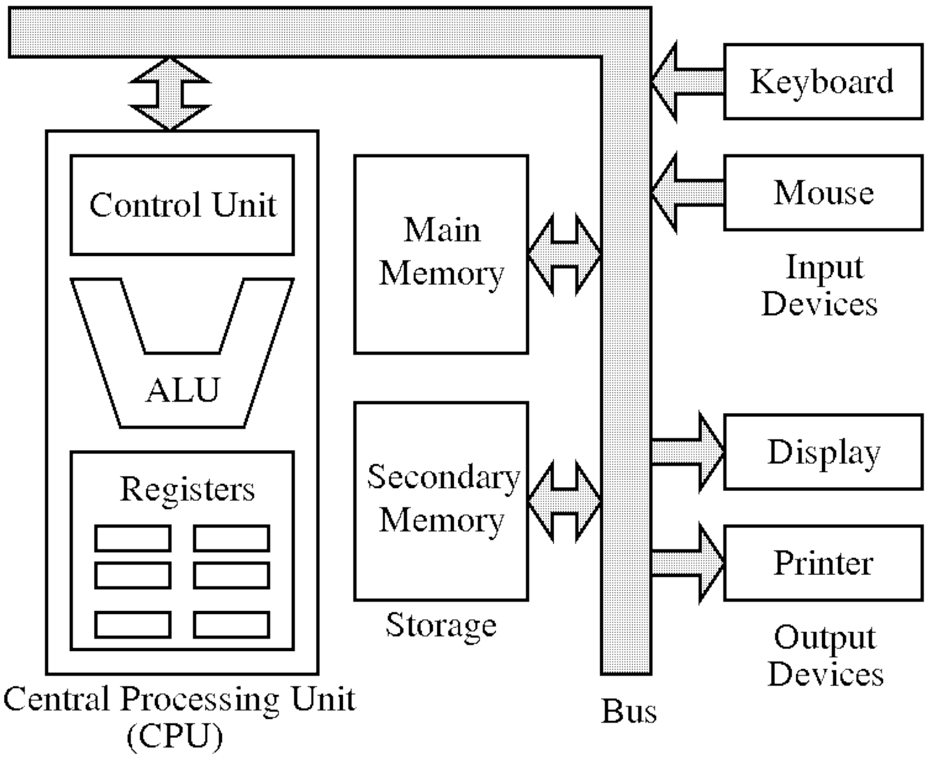
- Antiquity: Abacus

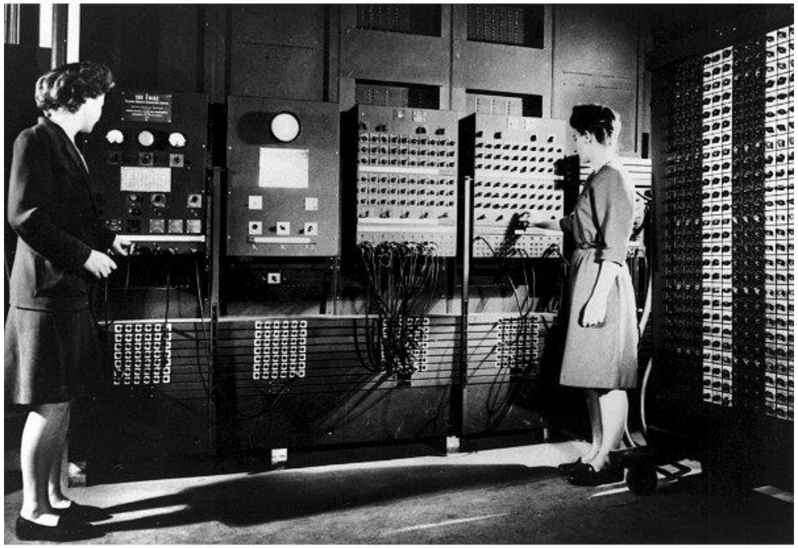


Thirds as per Menninger	As Thirds	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{4}$	$\frac{1}{3}+\frac{1}{4}$	$\frac{2}{3}+\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}+\frac{1}{3}$	$\frac{1}{2}+\frac{2}{3}$	$\frac{1}{2}+\frac{1}{4}$	$\frac{1}{2}+\frac{1}{4}+\frac{1}{3}$	$\frac{1}{2}+\frac{1}{4}+\frac{2}{3}$
	= Twelfths	0	$\frac{4}{12}$	$\frac{8}{12}$	$\frac{3}{12}$	$\frac{7}{12}$	$\frac{11}{12}$	$\frac{6}{12}$	$\frac{10}{12}$	$\frac{14}{12}$	$\frac{9}{12}$	$\frac{13}{12}$	$\frac{17}{12}$
Sixths as per Friedlein	As Sixths	0	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{6}+\frac{1}{4}$	$\frac{1}{3}+\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}+\frac{1}{6}$	$\frac{1}{2}+\frac{1}{3}$	$\frac{1}{2}+\frac{1}{4}$	$\frac{1}{2}+\frac{1}{4}+\frac{1}{6}$	$\frac{1}{2}+\frac{1}{4}+\frac{1}{3}$
	= Twelfths	0	$\frac{2}{12}$	$\frac{4}{12}$	$\frac{3}{12}$	$\frac{5}{12}$	$\frac{7}{12}$	$\frac{6}{12}$	$\frac{8}{12}$	$\frac{10}{12}$	$\frac{9}{12}$	$\frac{11}{12}$	$\frac{13}{12}$
Twelfths as per Greaves	As Twelfths	0	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{5}{12}$	$\frac{1}{2}$	$\frac{7}{12}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{5}{6}$	$\frac{11}{12}$
	= Twelfths	0	$\frac{1}{12}$	$\frac{2}{12}$	$\frac{3}{12}$	$\frac{4}{12}$	$\frac{5}{12}$	$\frac{6}{12}$	$\frac{7}{12}$	$\frac{8}{12}$	$\frac{9}{12}$	$\frac{10}{12}$	$\frac{11}{12}$

Computing Machinery Origins

- Modern times: Von Neumann architecture





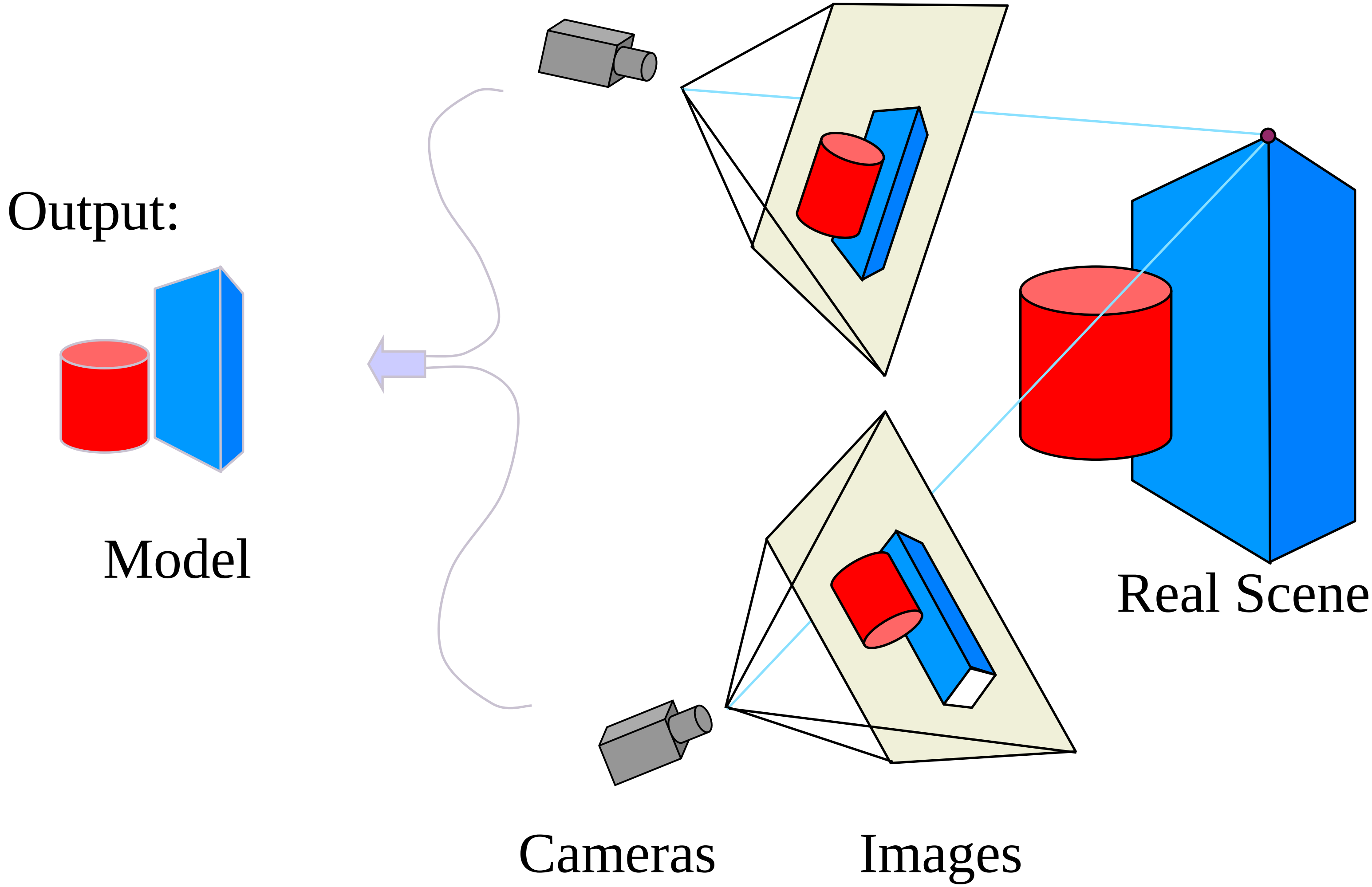
Related Areas

- Image Processing
- Computer Vision
- Computer Graphics

Image Processing

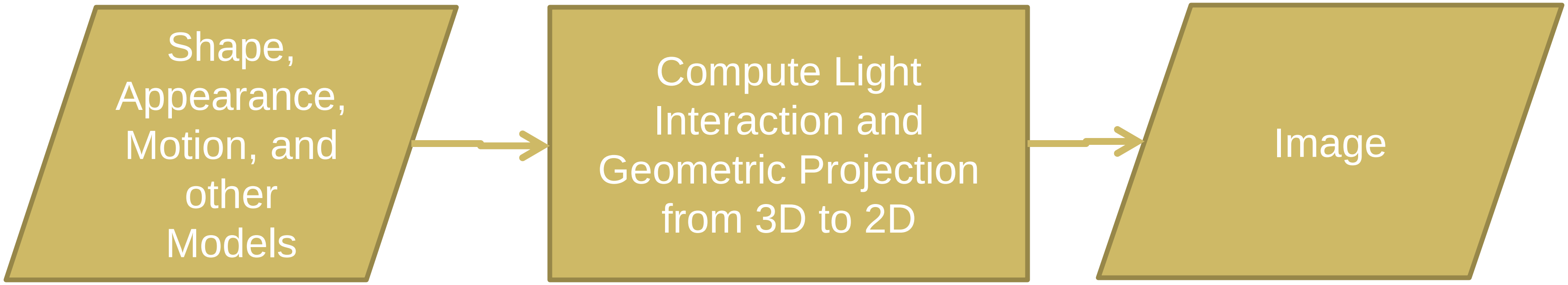
Computer Vision

Computer Vision

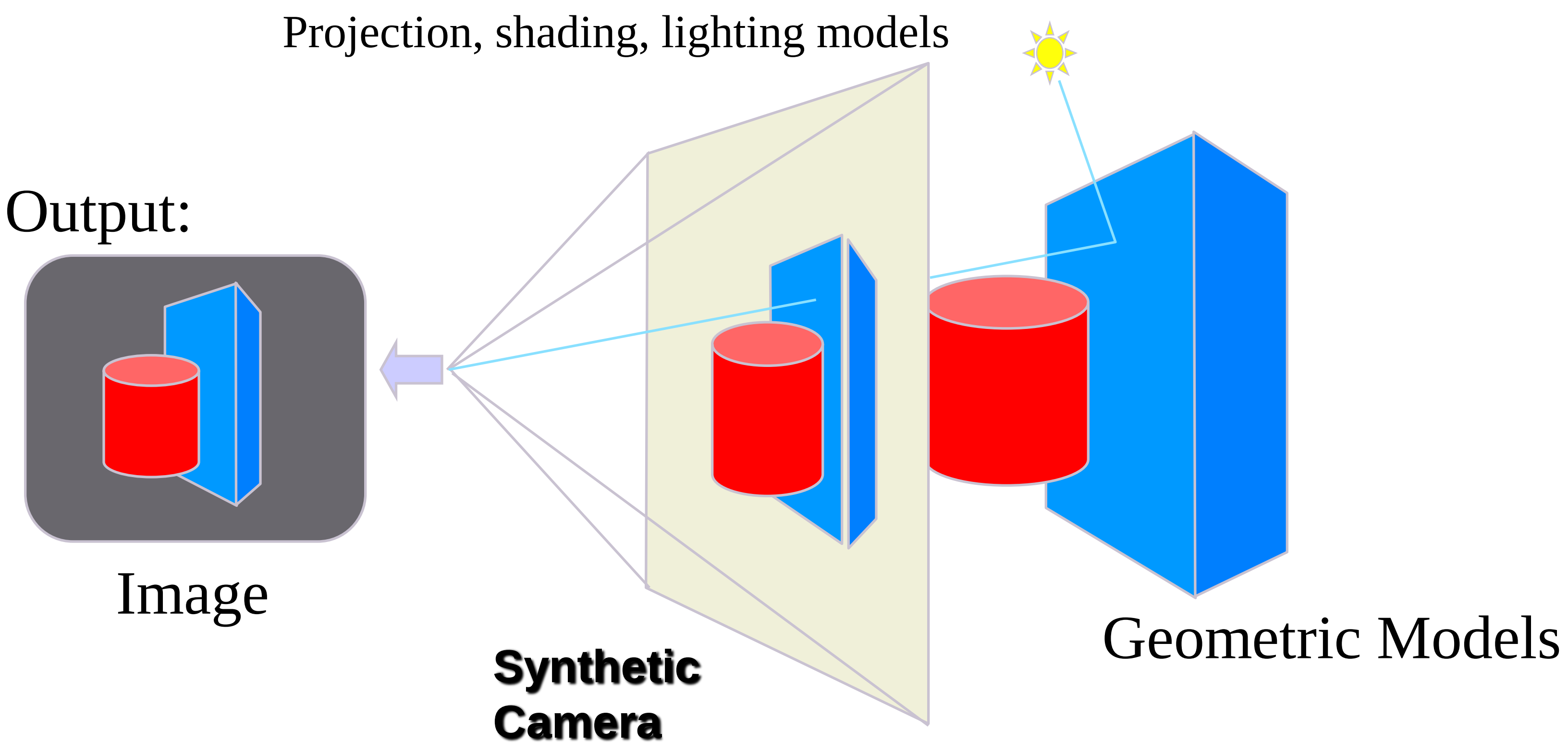


*: Source: CVL, UNR-Reno

Computer Graphics



Computer Graphics



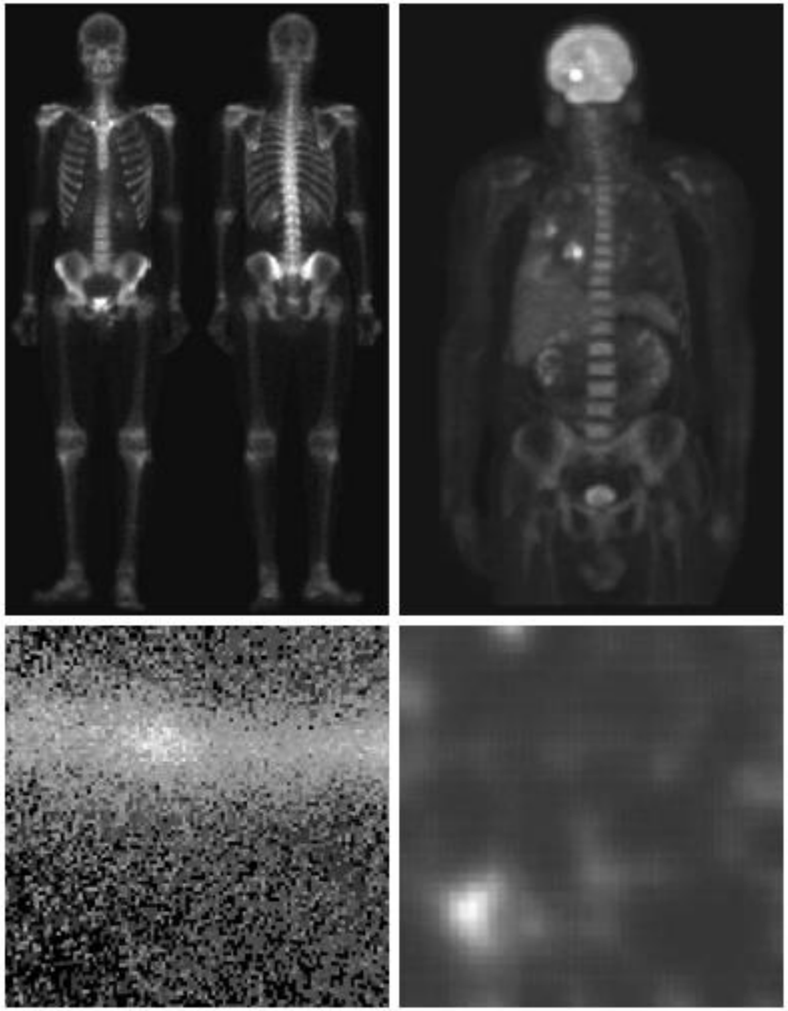
*: Source: CVL, UNR-Reno

Image Processing Application Fields

- The range of applications of digital image processing is very broad.
- We can categorize them by the imaging source.
- Imaging data is mostly sensed in the electromagnetic energy spectrum.
- Other sources include acoustic, ultrasonic and electronic.
- In the electromagnetic energy spectrum the following types are acquired
  - X-ray
  - Gamma-ray
  - Untraviolet band
  - Visible and Infrared bands
  - Microwave band
  - Radio band
  - Others (acoustic, ultrasonic, electron microscopy)

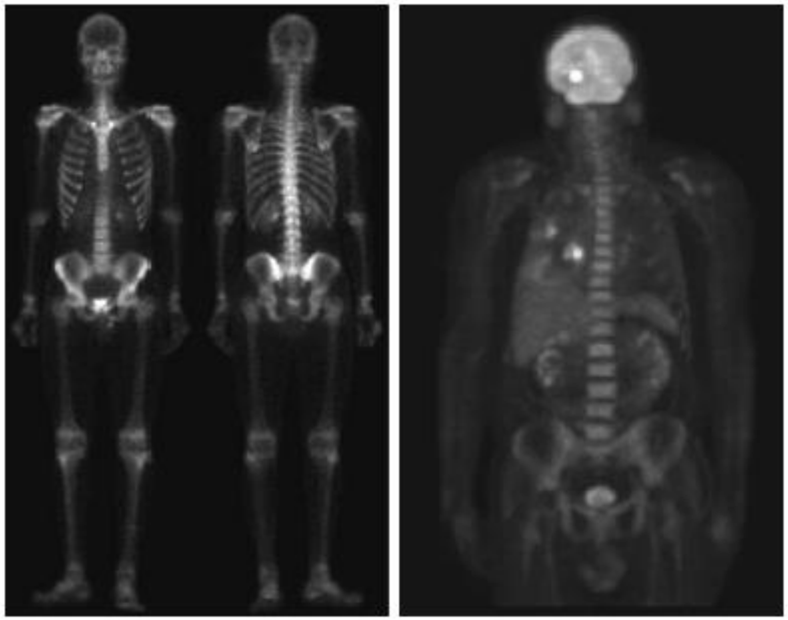
Gamma-Ray Imaging

- High energy band in EM spectrum.
- Applications in nuclear medicine and astronomical observations.



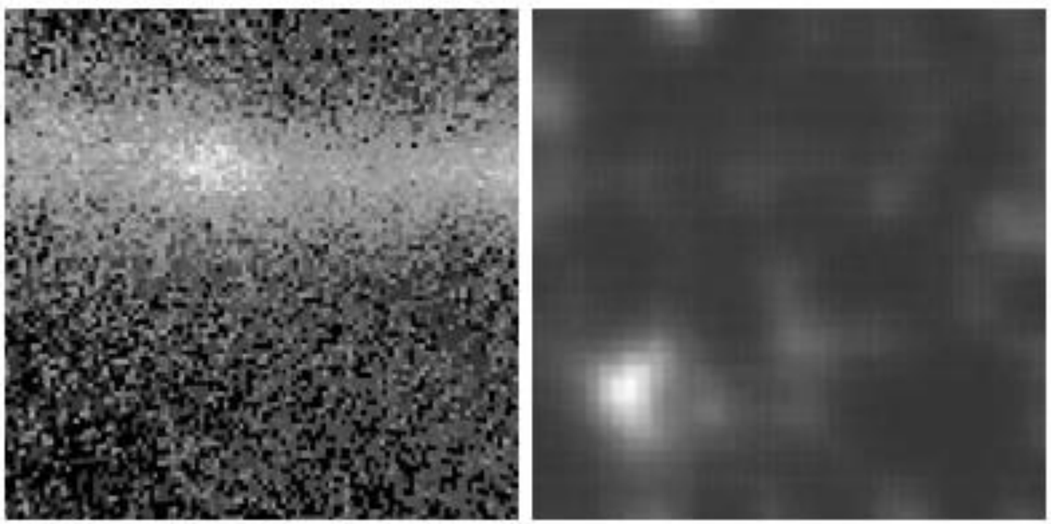
Gamma-Ray Imaging in Nuclear Medicine

- Administer radioactive agent in patient's body.
- Agent emits gamma rays as it decays.
- Used to locate sites of pathology (tumors for example) in bone, lungs and other tissues.
- Examples: Single Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET) imaging.



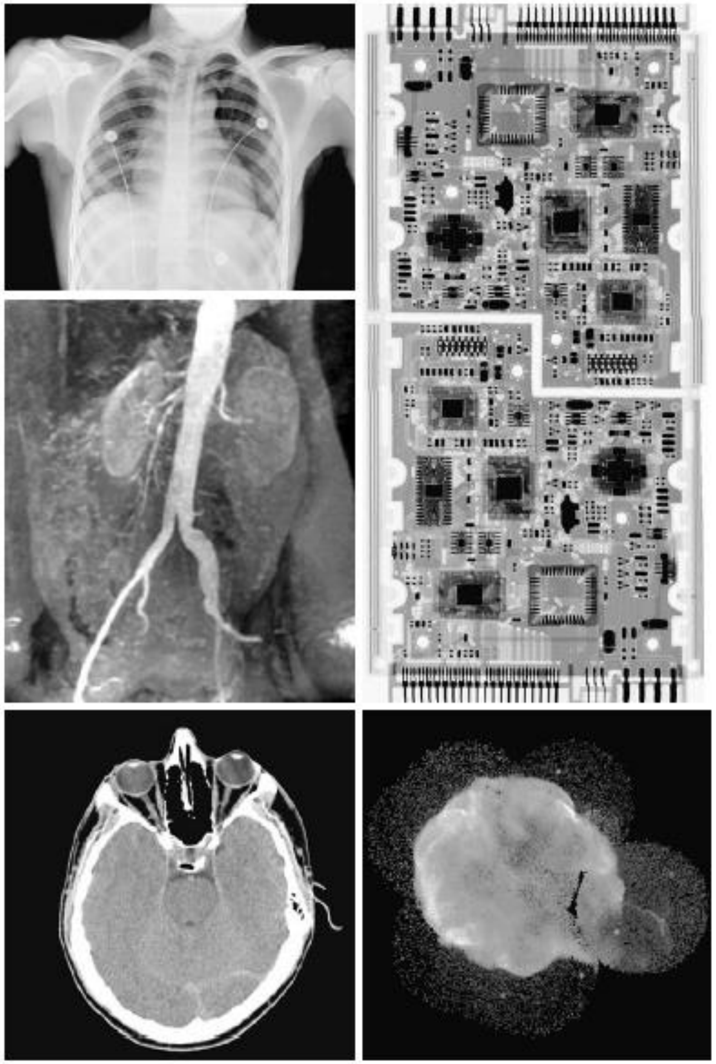
Gamma-Ray Imaging in Astronomy

- Astronomical observations: sense gamma-ray band produced by natural radiation of imaged scene.



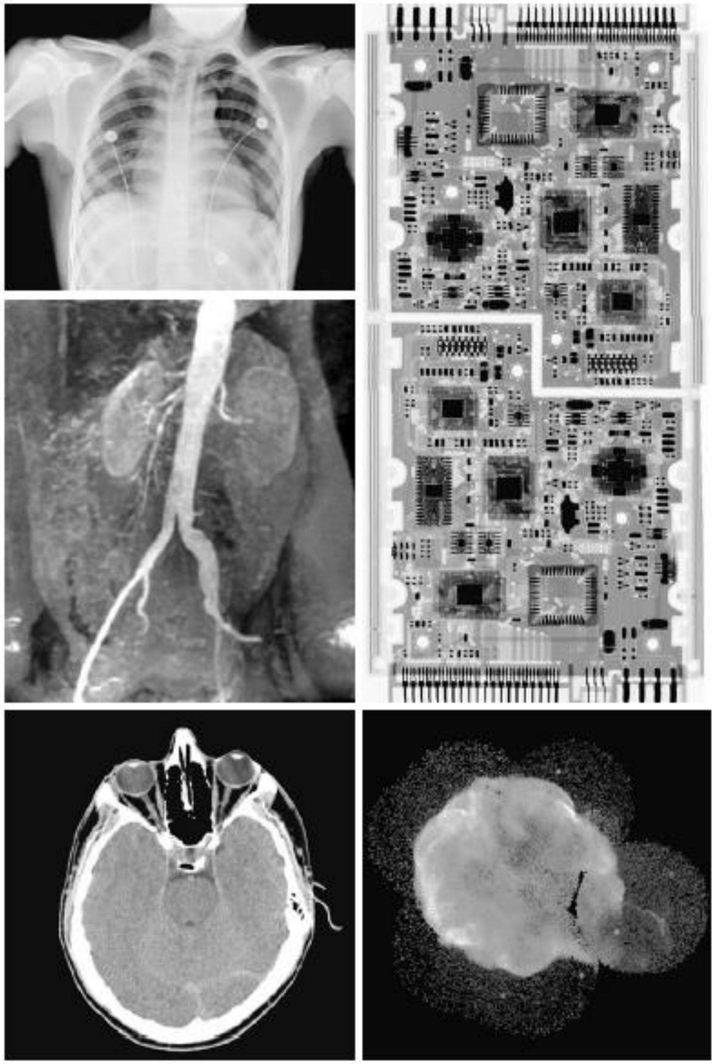
X-Ray Imaging

- X-rays are generated in a vacuum tube with anode and cathode.
- Cathode is heated, electrons are released to anode.
- When electron strikes a nucleus, x-ray radiation is emitted.
- Energy is controlled by voltage applied across anode and by current applied to filament of cathode.
- X-rays pass through patient, some absorbed by tissues, others falling on the film.



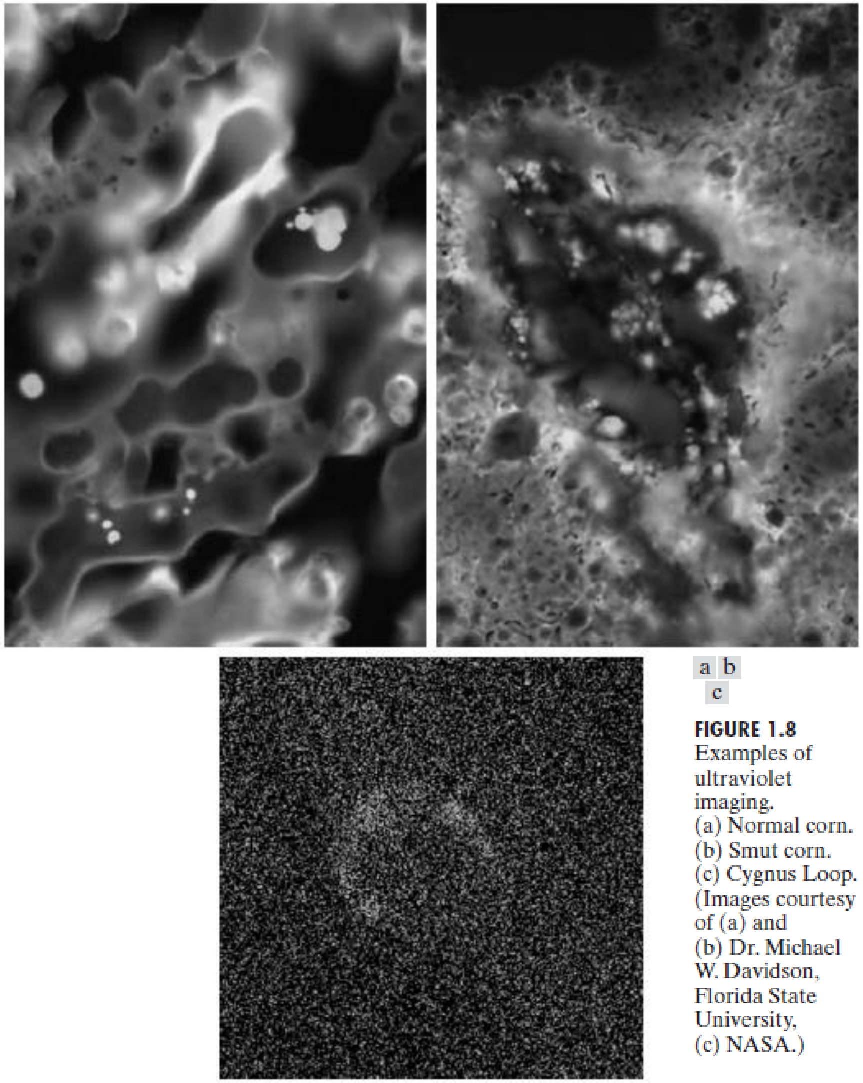
Digital X-Ray Imaging

- Two methods
  1. Digitize x-ray films, or
  2. Use phosphor screens or other devices to convert x-rays to light, passed to light sensitive system.



Ultraviolet Imaging

- Fluorescence microscopy
  - Use ultraviolet source to excite fluorescent material.
  - Lights is emitted in red light wavelengths.

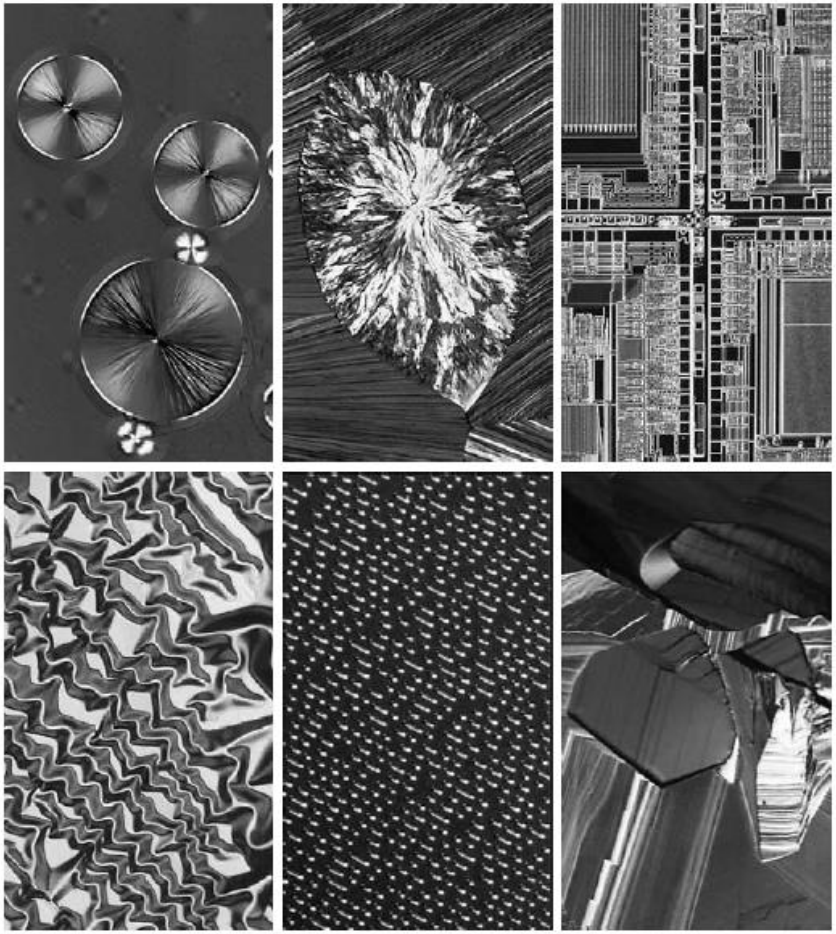


a b
c

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.)

Light Microscopy Imaging

- Light microscopes use visible light to detect small objects.
- Further categories of visible light microscopy are optical and fluorescence microscopy.
- Applications
  - pharmaceutical
  - microinspection
  - material characterization.

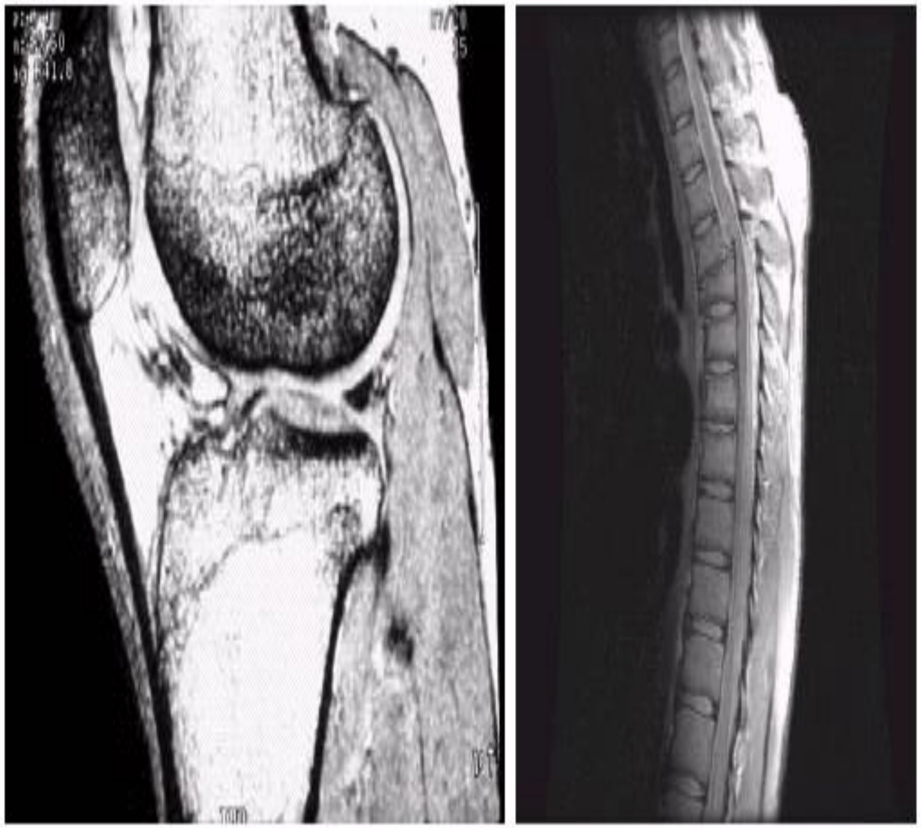


a b c
d e f

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.)

Magnetic Resonance Imaging (MRI)

- Patient under a powerful static magnet field B_0.
- Radio Frequency (RF) Coil passes radio waves through body.
- Acquire response magnetic dipoles (mostly protons) in body and reconstruct image.



Imaging Besides the EM Spectrum

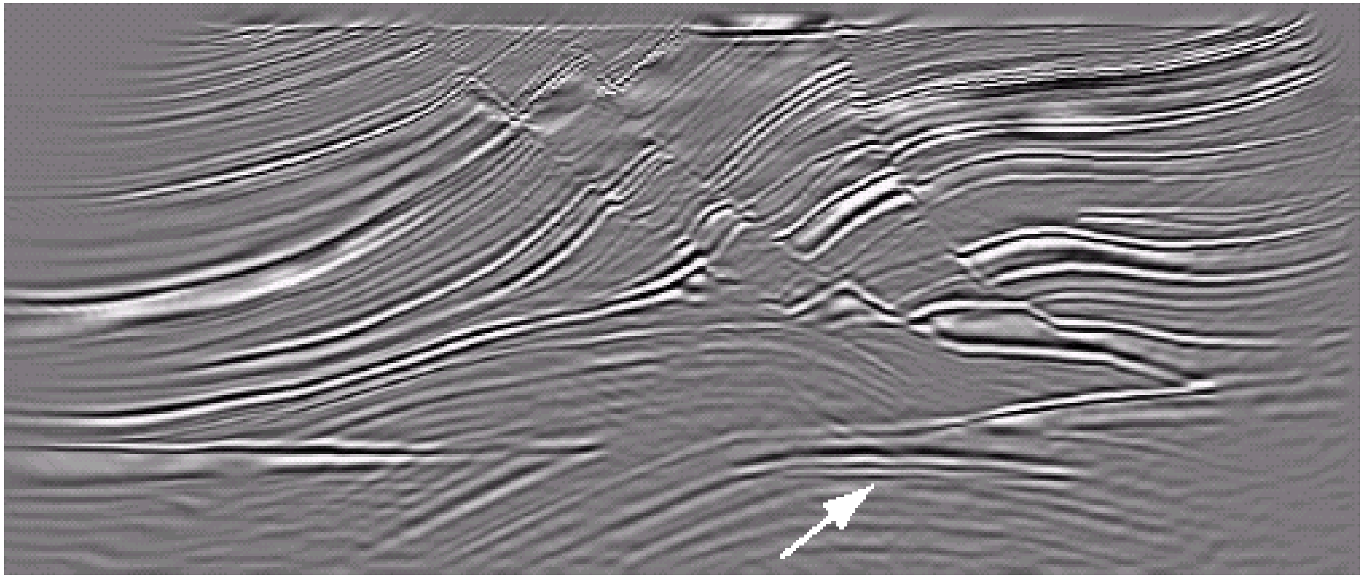
There exist imaging techniques that acquire information outside the EM spectrum.

Examples are acoustic imaging, electron microscopy, and synthetic images.

Acoustic

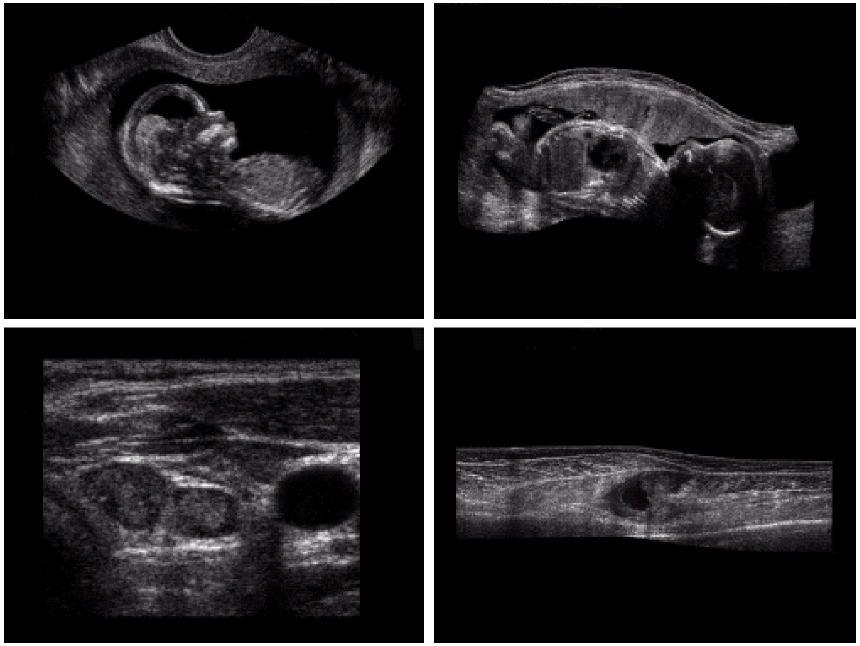
- Imaging acquired in lower frequencies (hundreds of Hz) for geological applications. Higher frequencies (millions of Hz) are sensed for ultrasound imaging.
- Applications: mineral and oil exploration and geology.

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.)



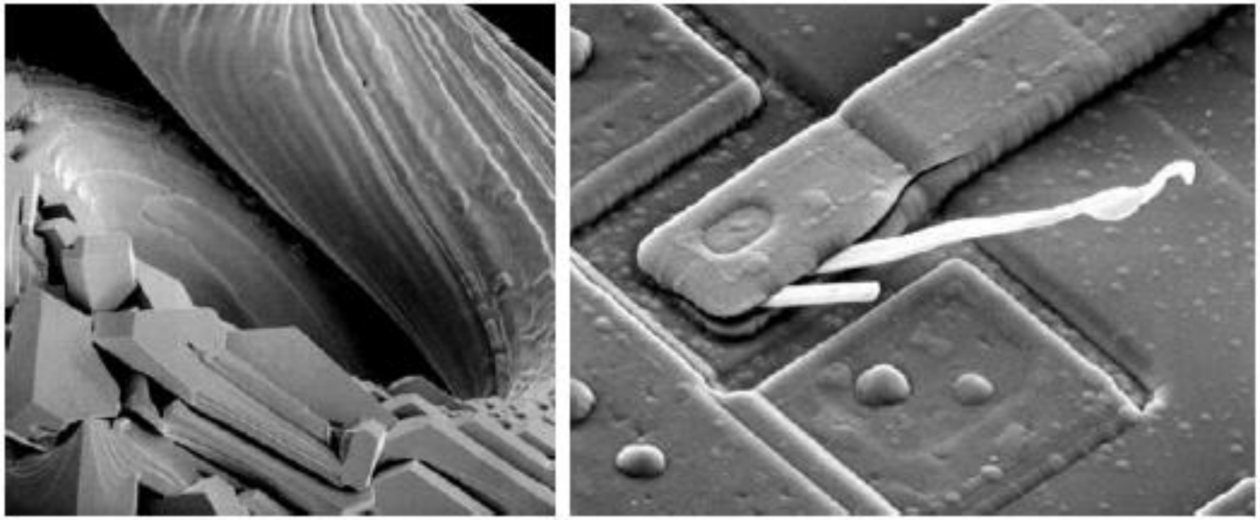
Ultrasound Imaging

- Sound waves propagate mechanical energy causing periodic vibration of particles in a continuous elastic medium.
- Unltrasound imaging system components
    1. piezoelectric crystal-based transducer (transmitter and receiver)
    2. control panel with pulse generation and control
    3. computer processing a.nd display system.



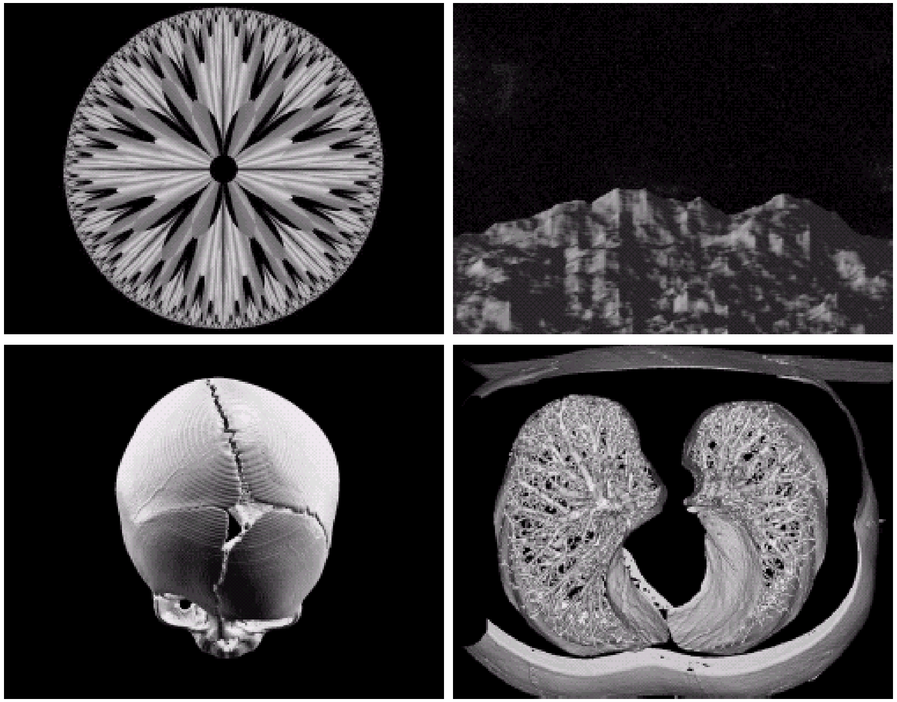
Electron Microscopy

- This modality operates similarly to optical microscopy, except for using a focused beam of electrons instead of light to image a specimen.



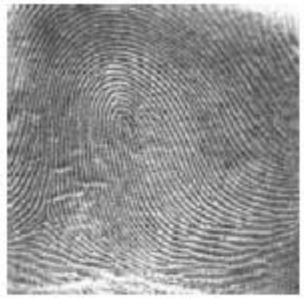
Synthetic Imaging

- Computer-generated images. Examples:
  - Fractals
  - 3-D renderings using computer graphics.



Imaging in the Visible Spectrum

- Ubiquitous in modern world.
- Used for everyday activities, also supported by technological advances.



Multiple Imaging Modalities

- It is frequently useful to acquire multiple images of the same scene that sense different wavebands, or physical properties.
- Information of multiple modalities is combined before further analysis.

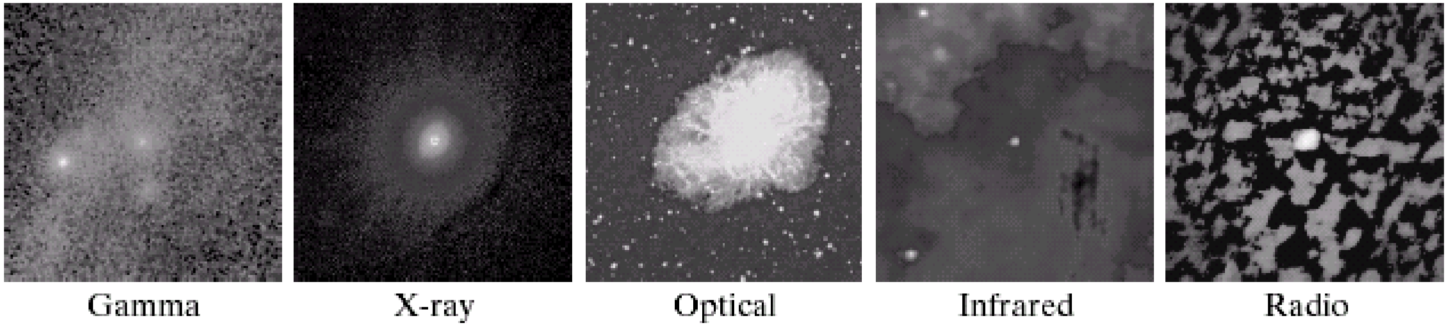


FIGURE 1.18 Images of the Crab Pulsar (in the center of images) covering the electromagnetic spectrum. (Courtesy of NASA.)

Components of Image Processing System

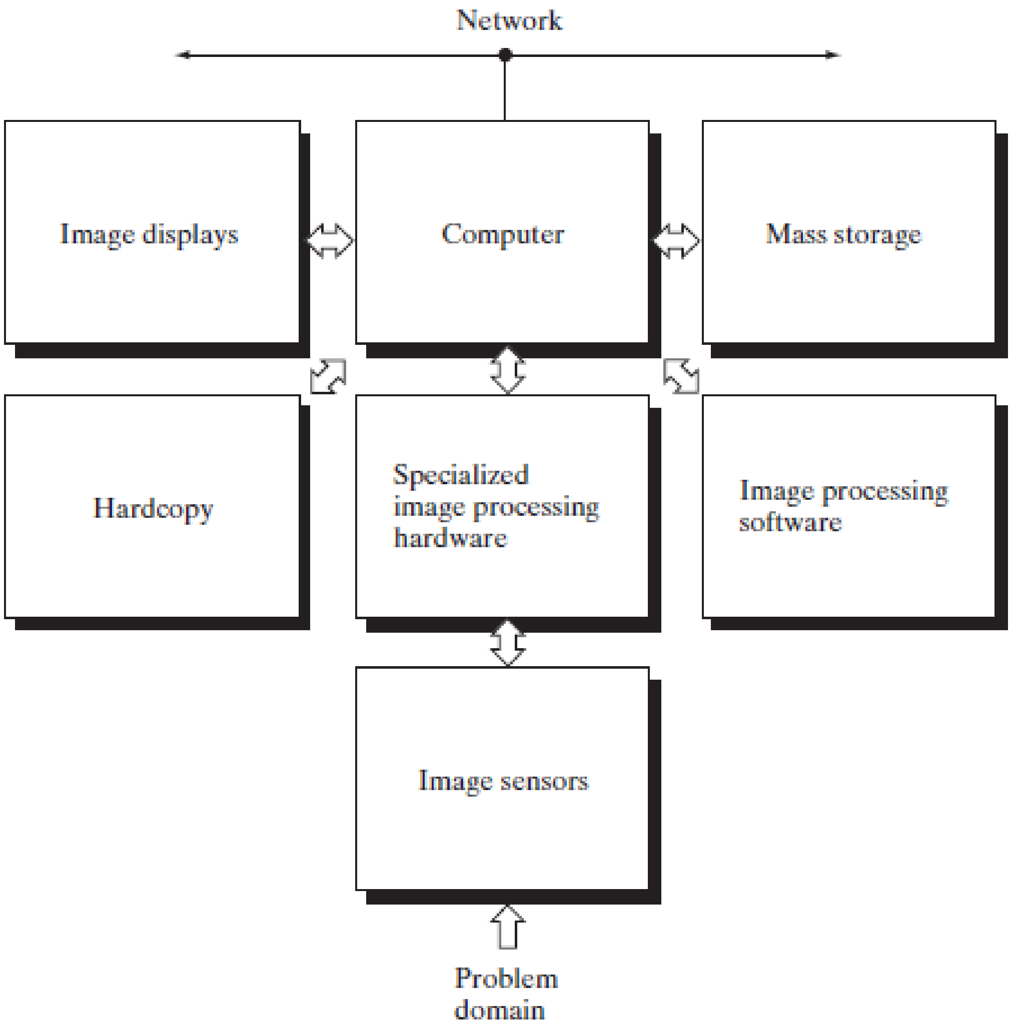


FIGURE 1.24 Components of a general-purpose image processing system.

Image Processing Tasks

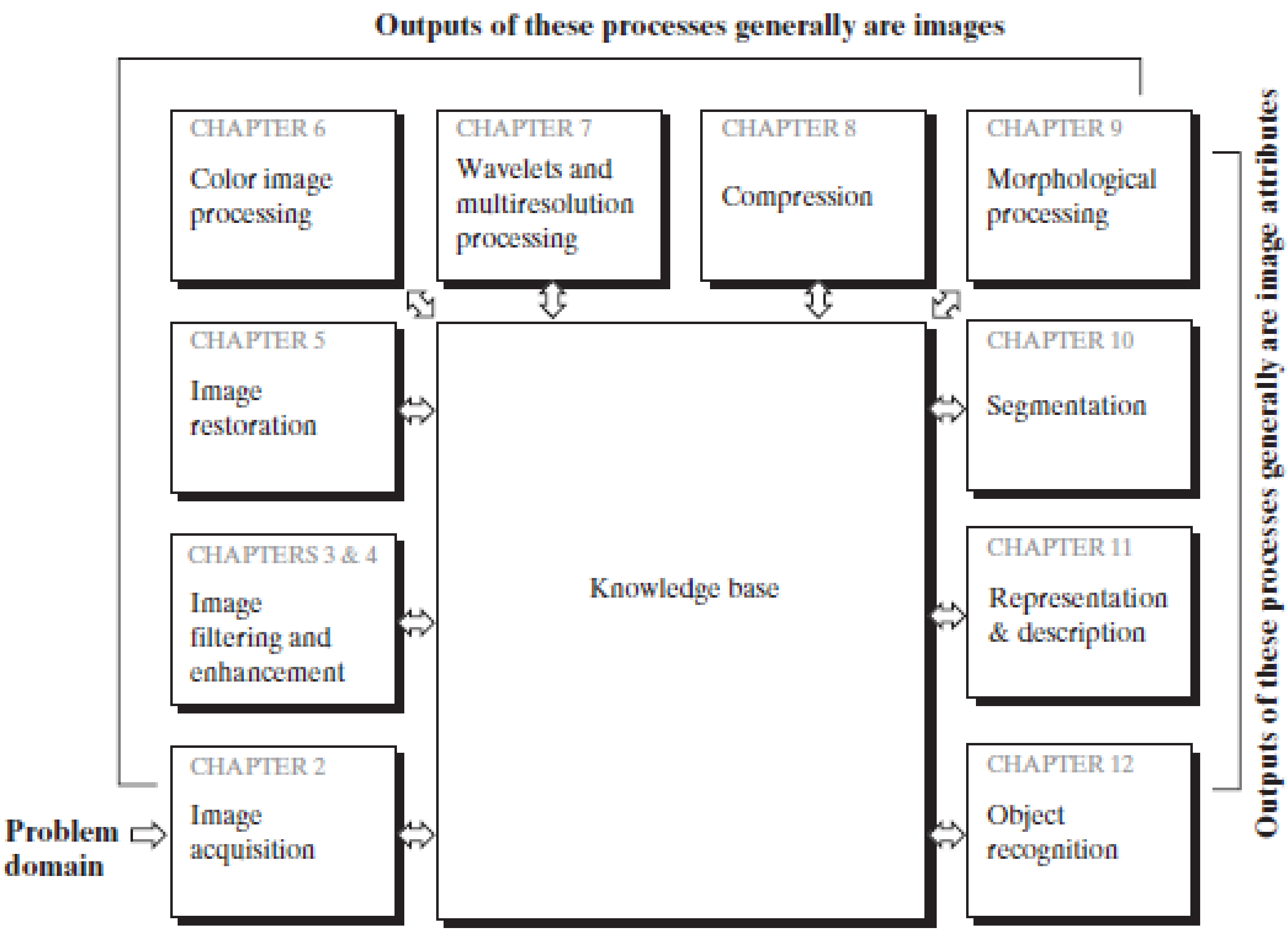
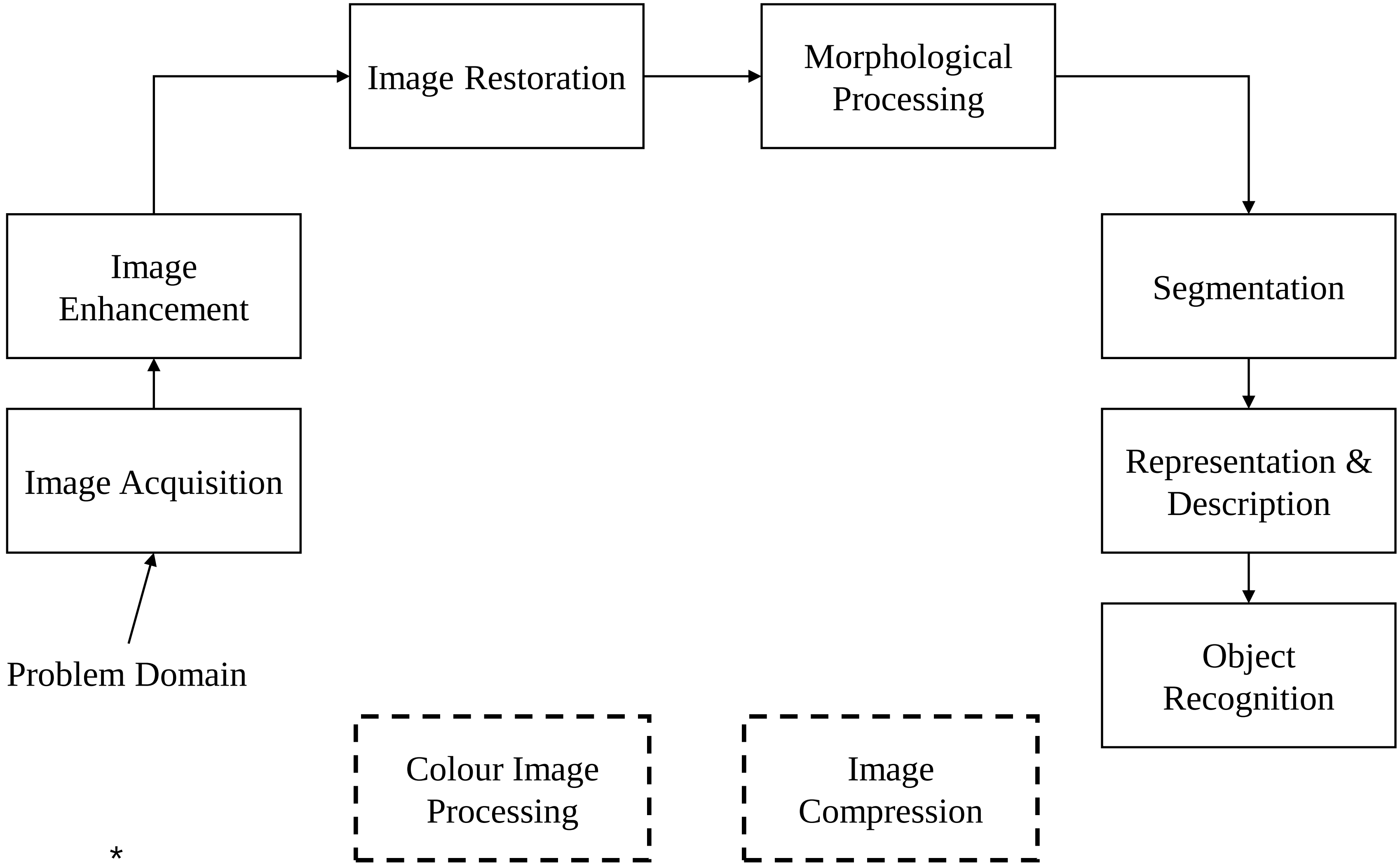


FIGURE 1.23 Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed.

Key Stages in Digital Image Processing



*

Key Stages in Digital Image Processing: Image Aquisition

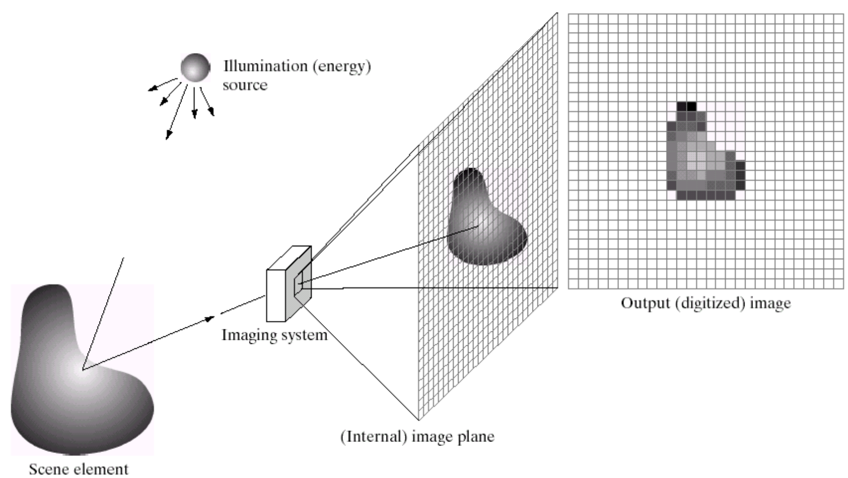
Image Restoration

Morphological Processing

Image Enhancement

Image Acquisition

Problem Domain



Segmentation

Representation & Description

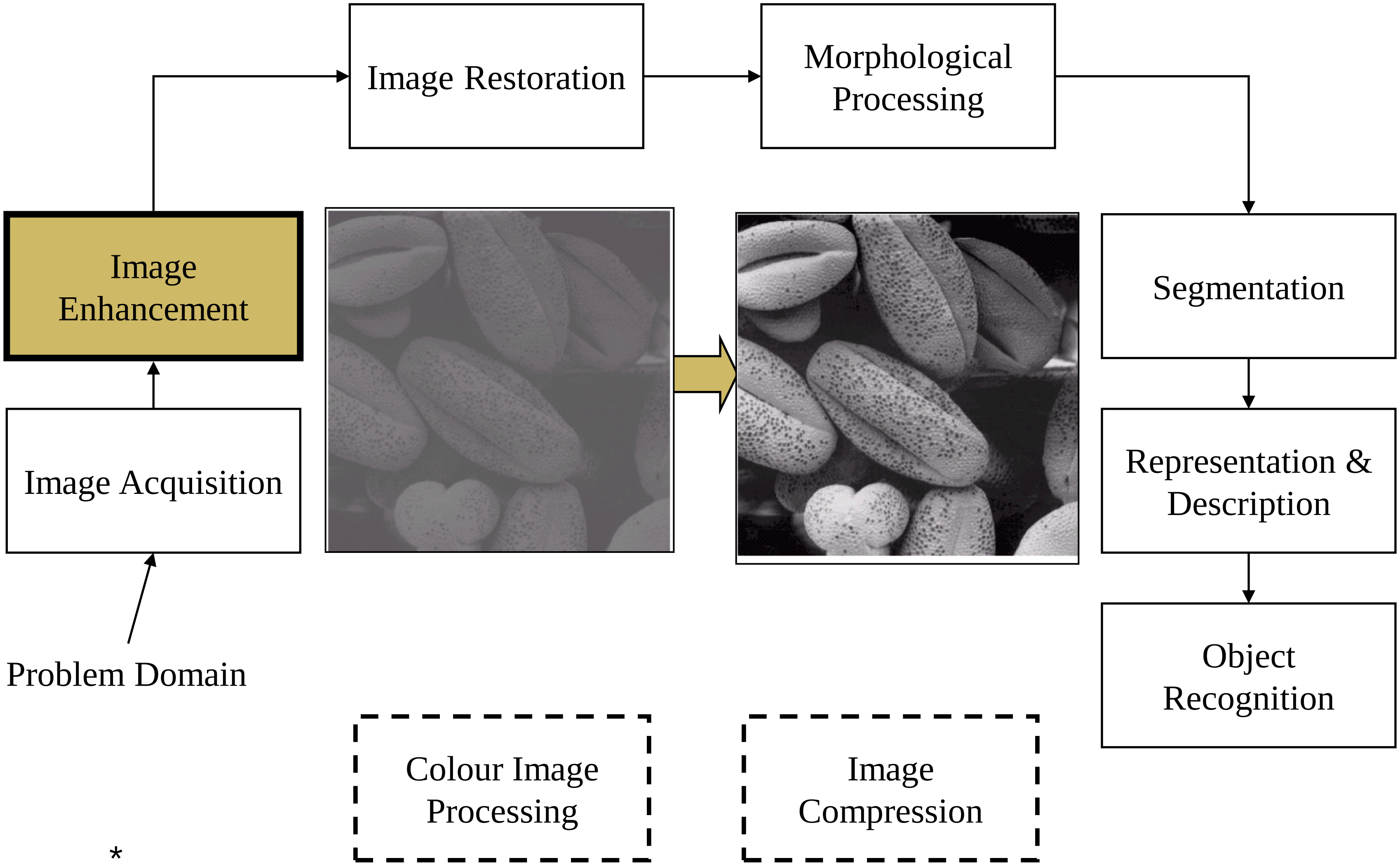
Object Recognition

Colour Image Processing

Image Compression

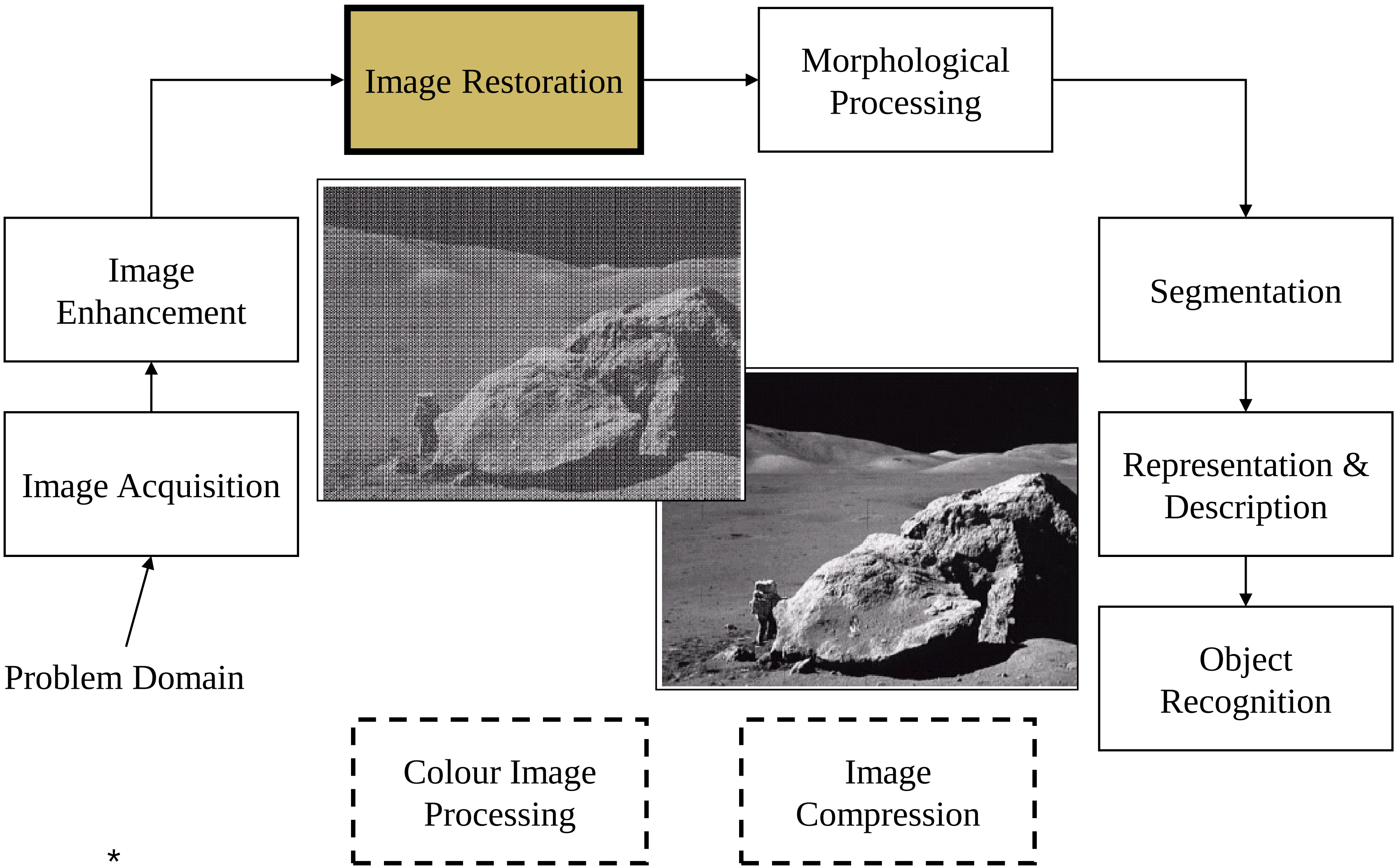
*

Key Stages in Digital Image Processing: Image Enhancement



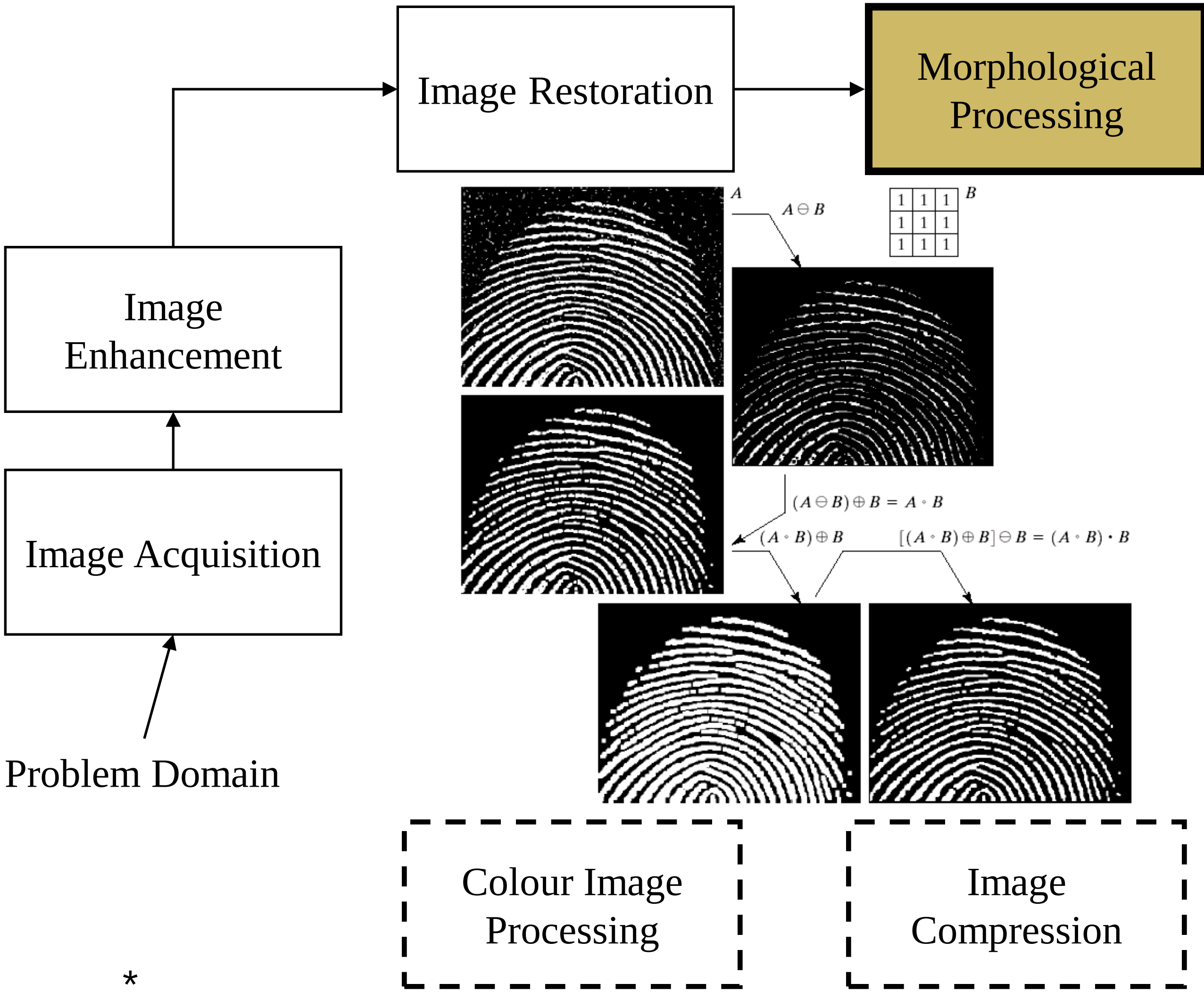
*

Key Stages in Digital Image Processing: Image Restoration



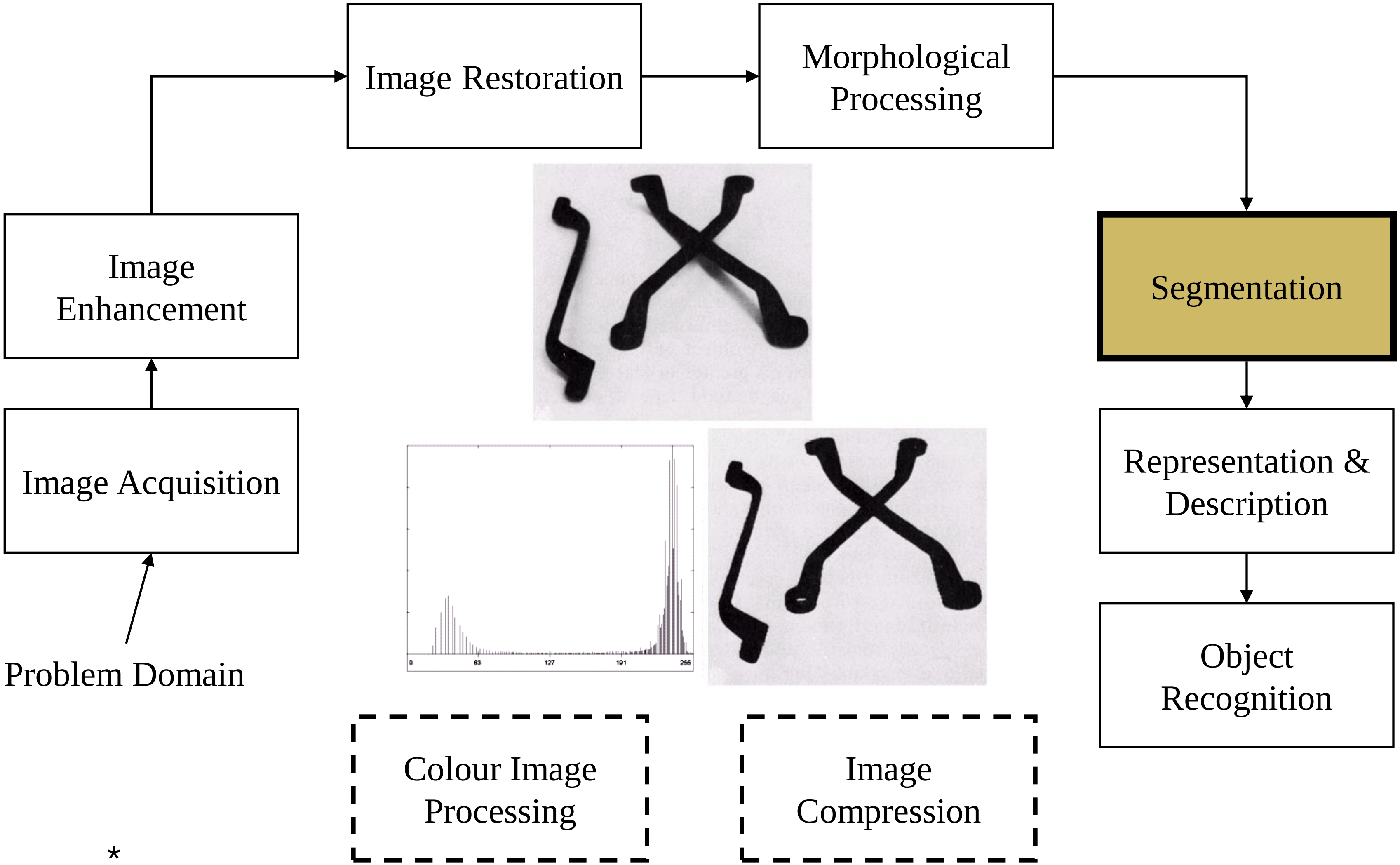
*

Key Stages in Digital Image Processing: Morphological Processing



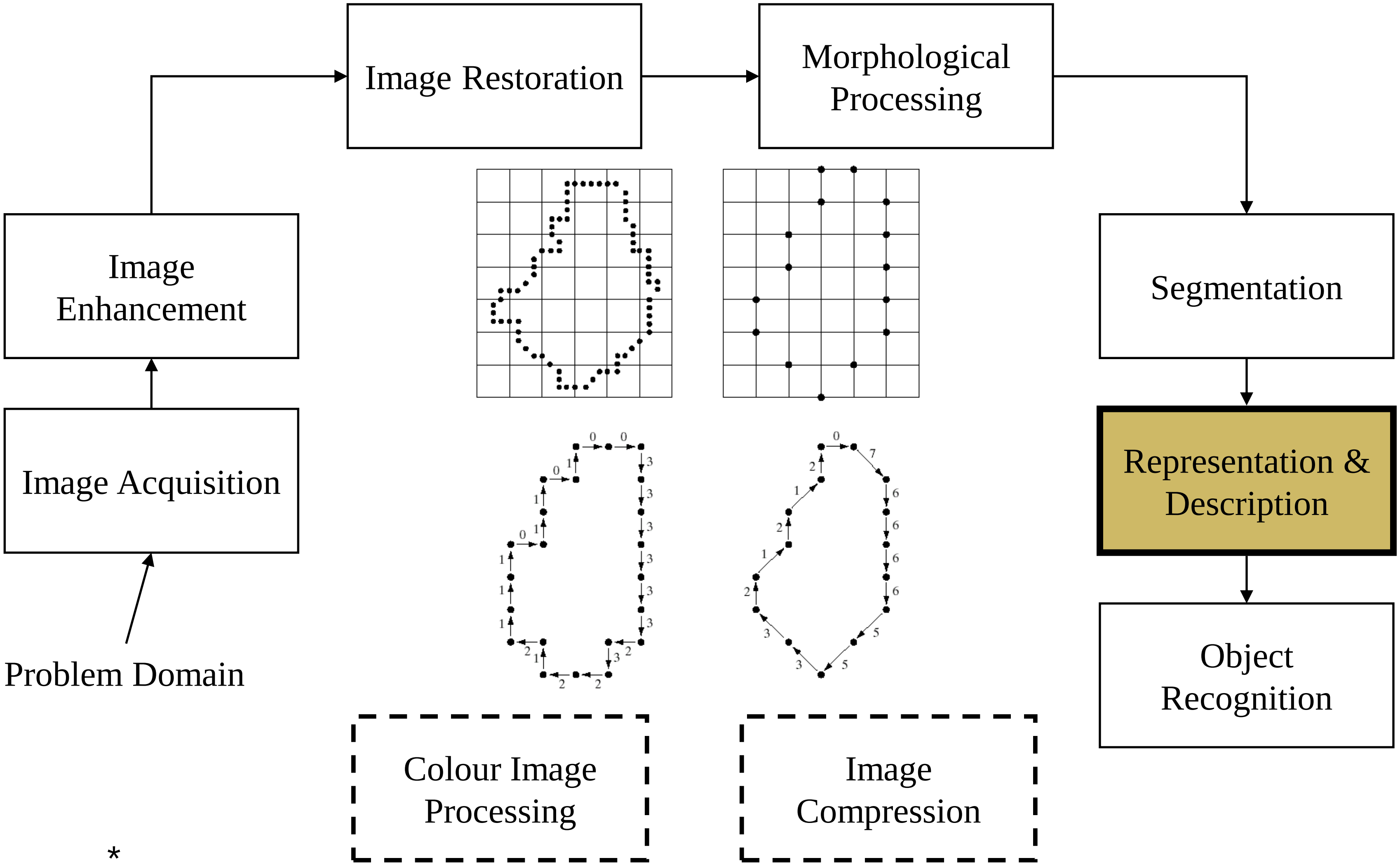
*

Key Stages in Digital Image Processing: Segmentation



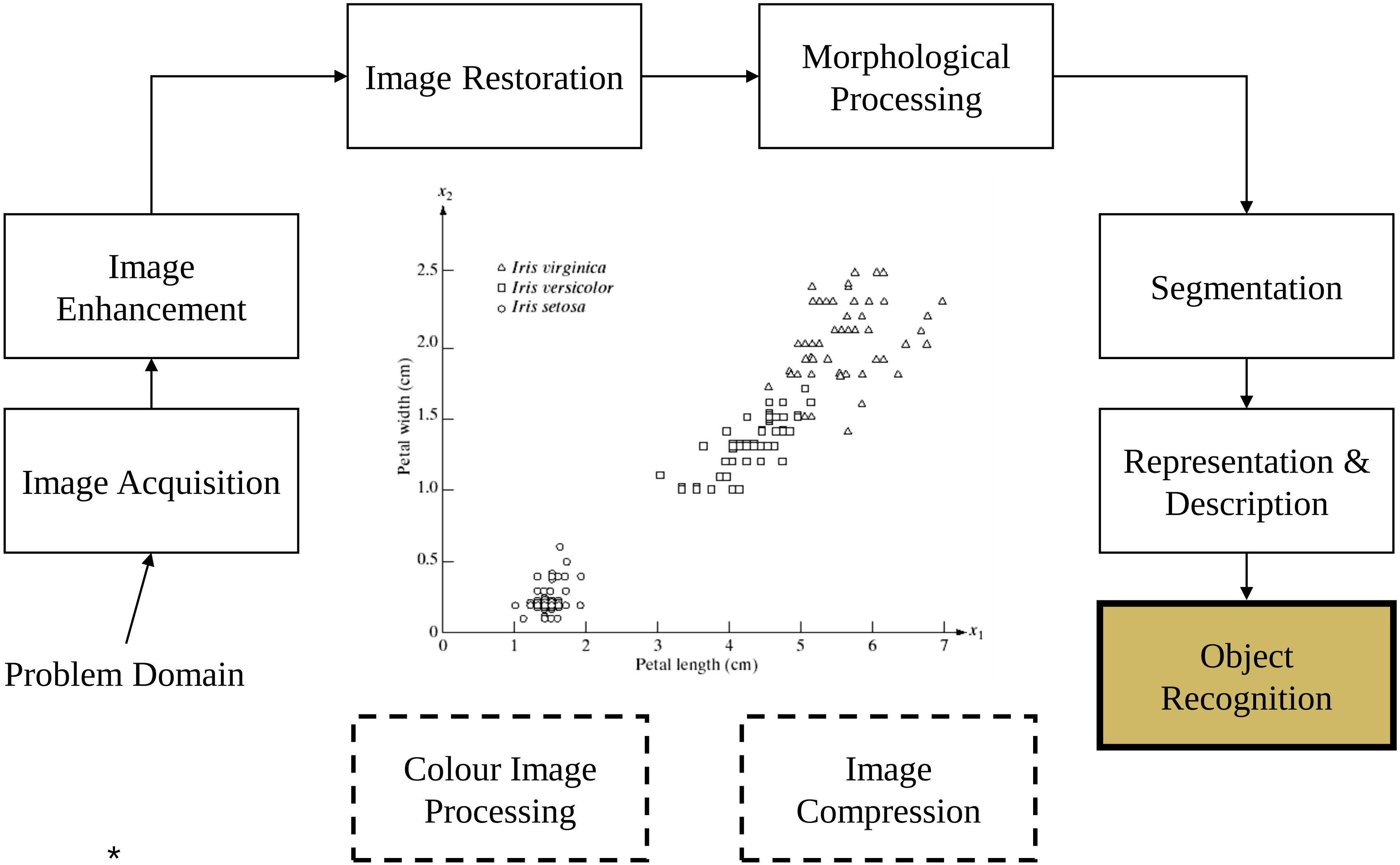
*

Key Stages in Digital Image Processing: Representation & Description



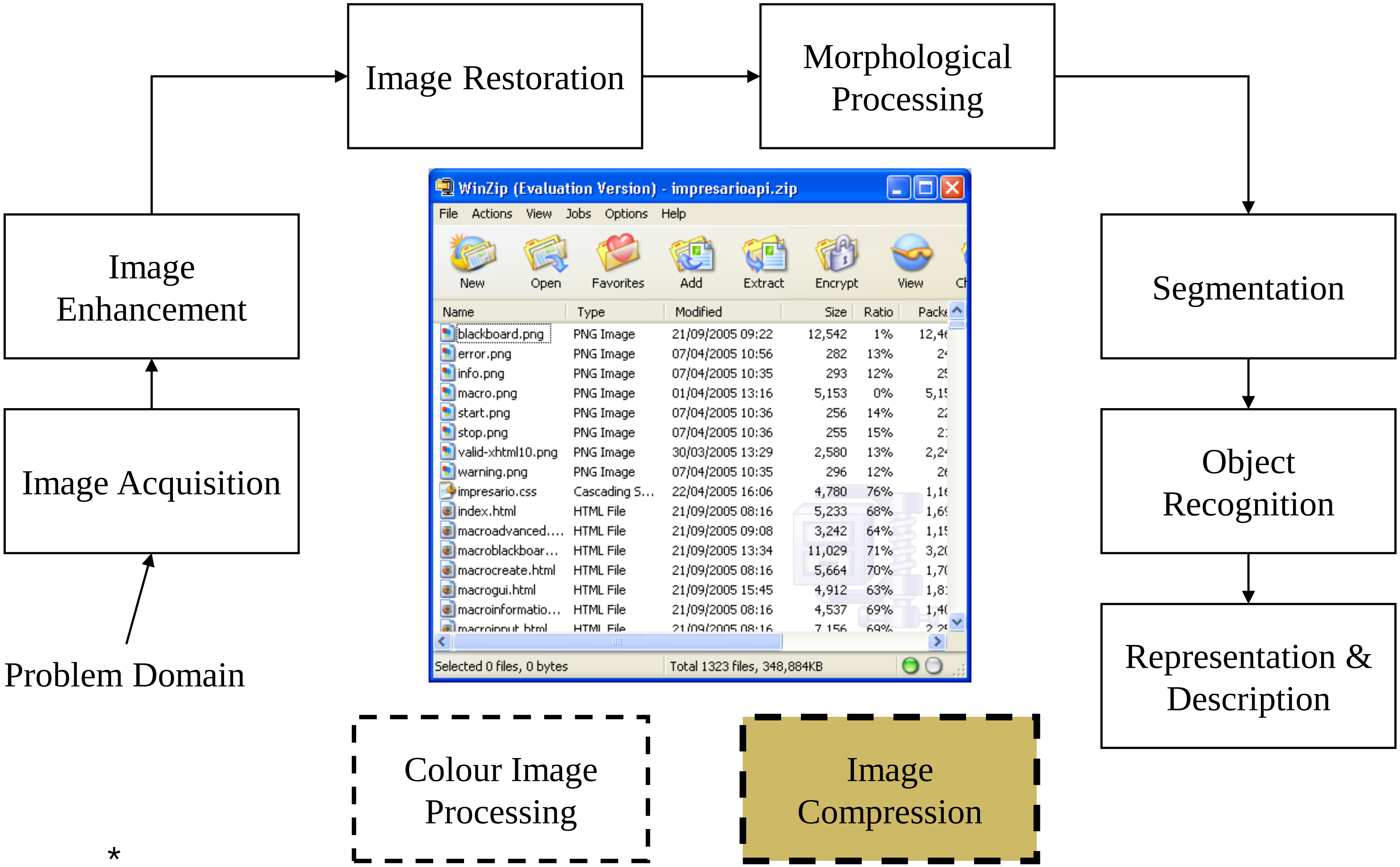
*

Key Stages in Digital Image Processing: Object Recognition



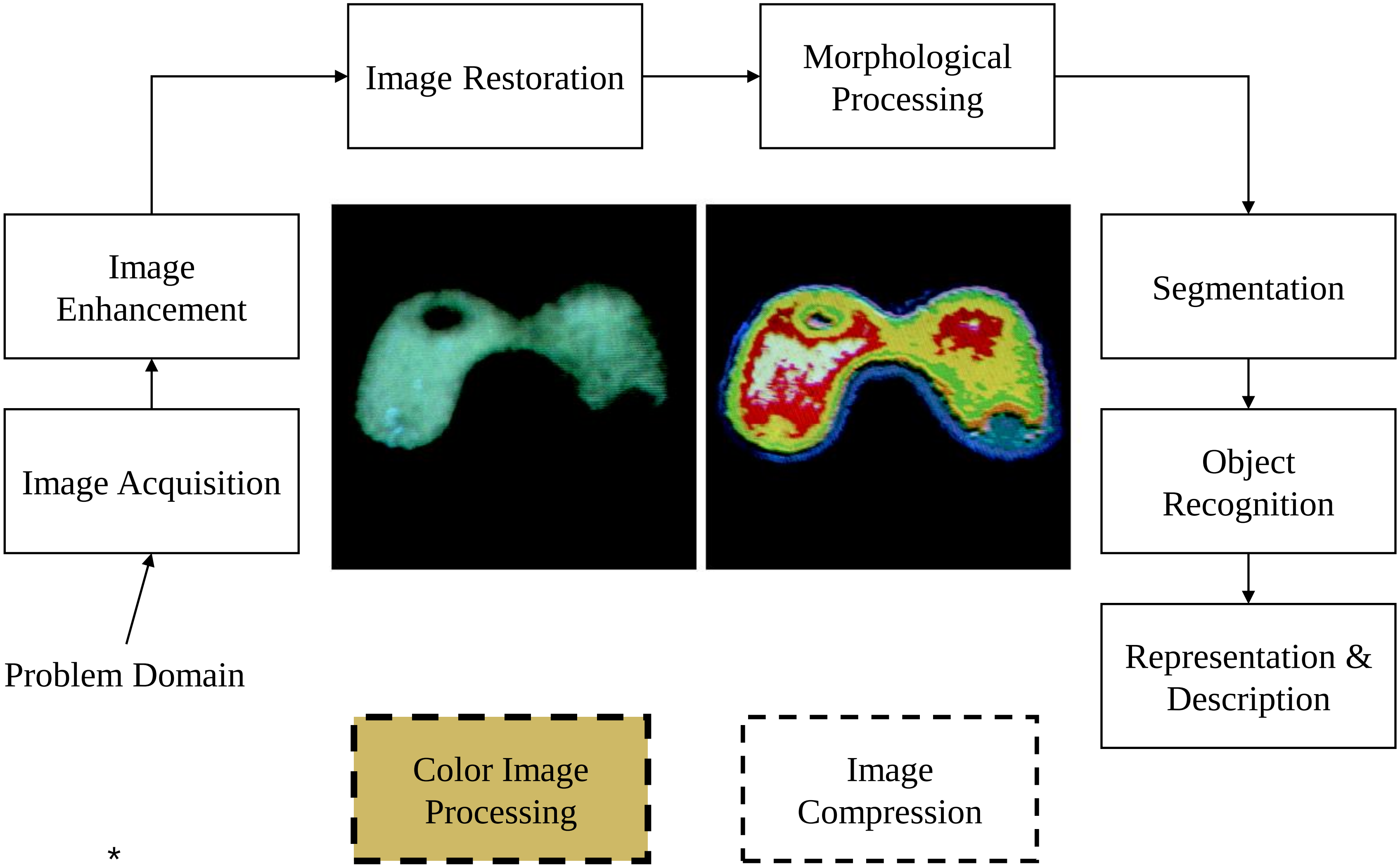
*

Key Stages in Digital Image Processing: Image Compression



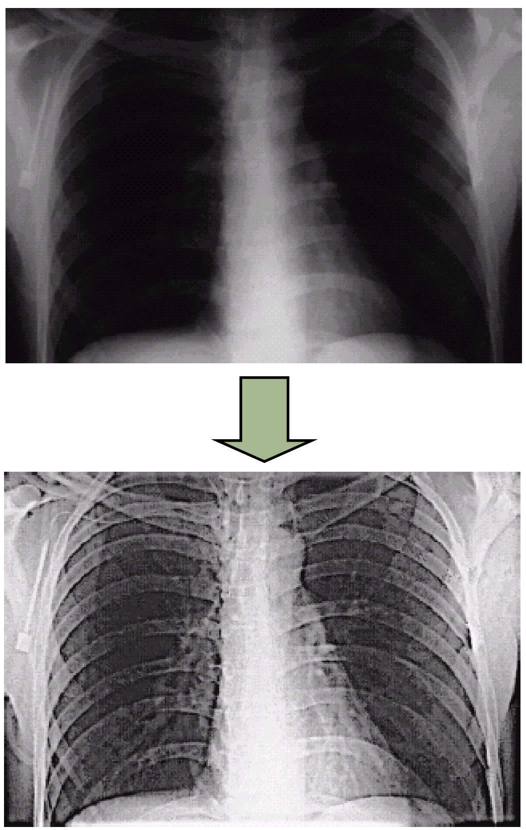
*

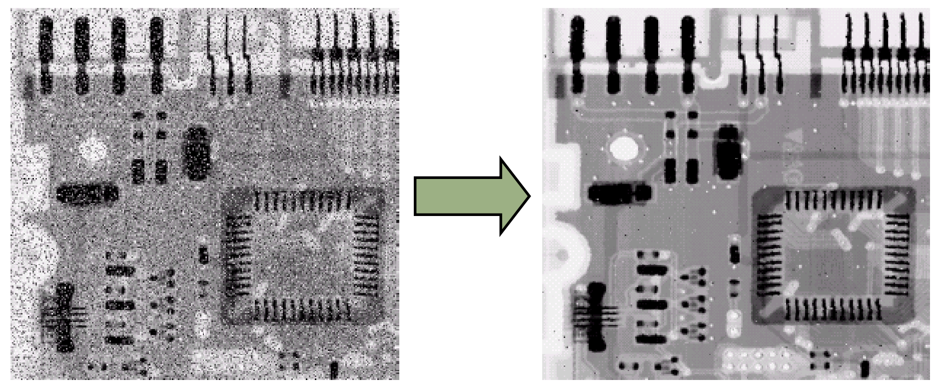
Key Stages in Digital Image Processing: Color Image Processing



Applications: Image Enhancement

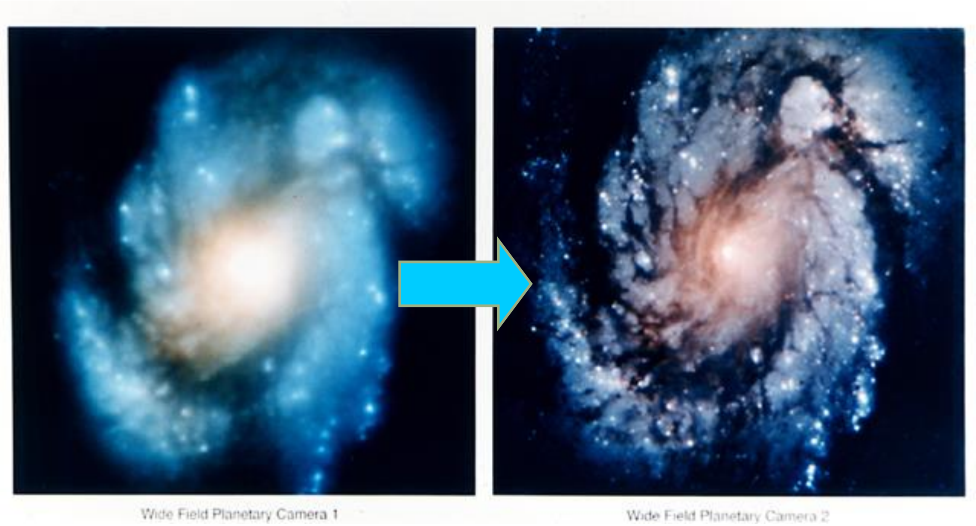
- One of the most common uses of DIP techniques: improve quality, remove noise etc.





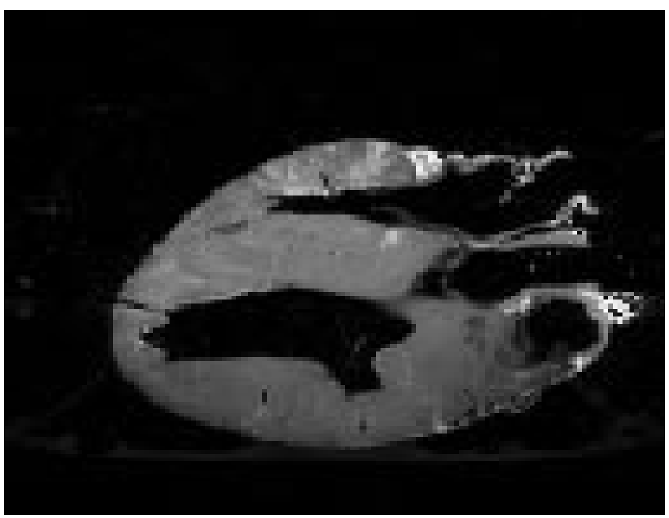
Applications: Space

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

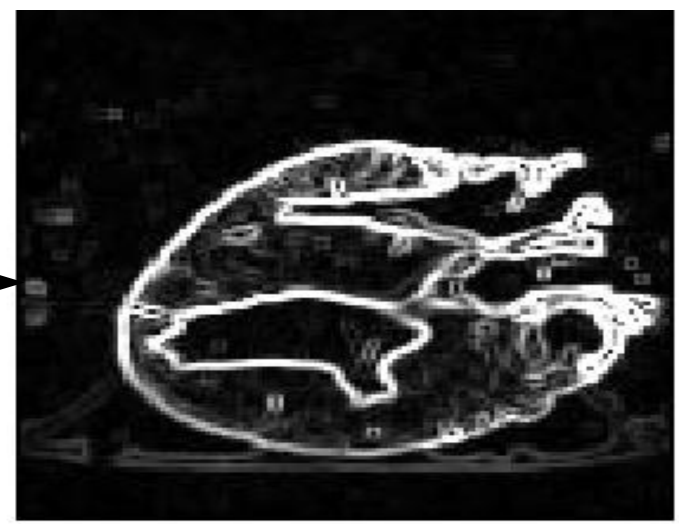


Applications: Medicine

- Take slice from MRI scan of dog heart, and find boundaries between types of tissue.
  - Image with gray levels representing tissue density.
  - Use a suitable filter to highlight edges.



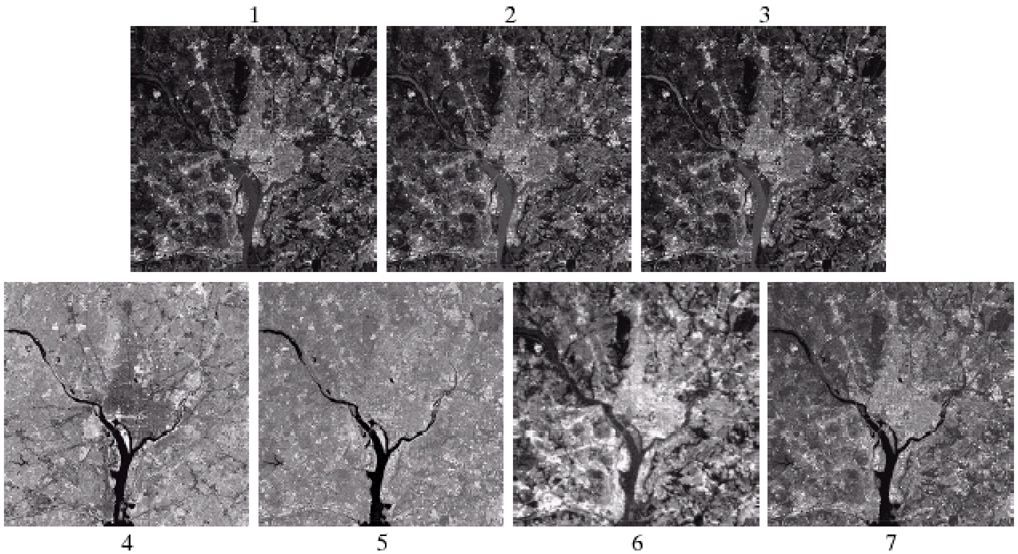
Original MRI Image of a Dog Heart

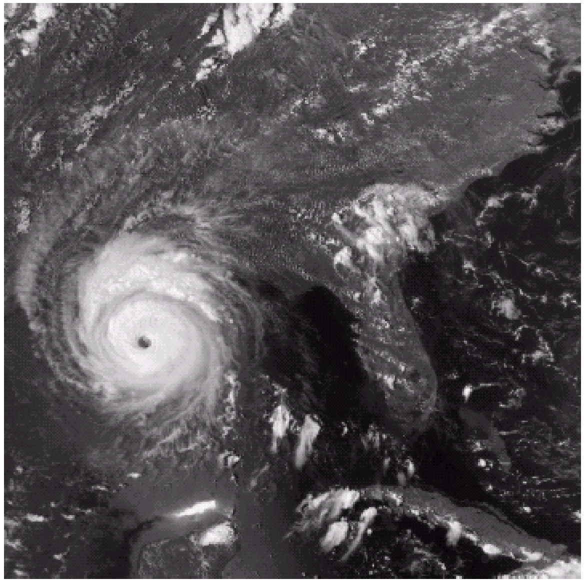


Edge Detection Image

Applications: GIS

- Geographic Information Systems
  - Digital image processing techniques are used extensively to manipulate satellite imagery.
  - Terrain classification
  - Meteorology



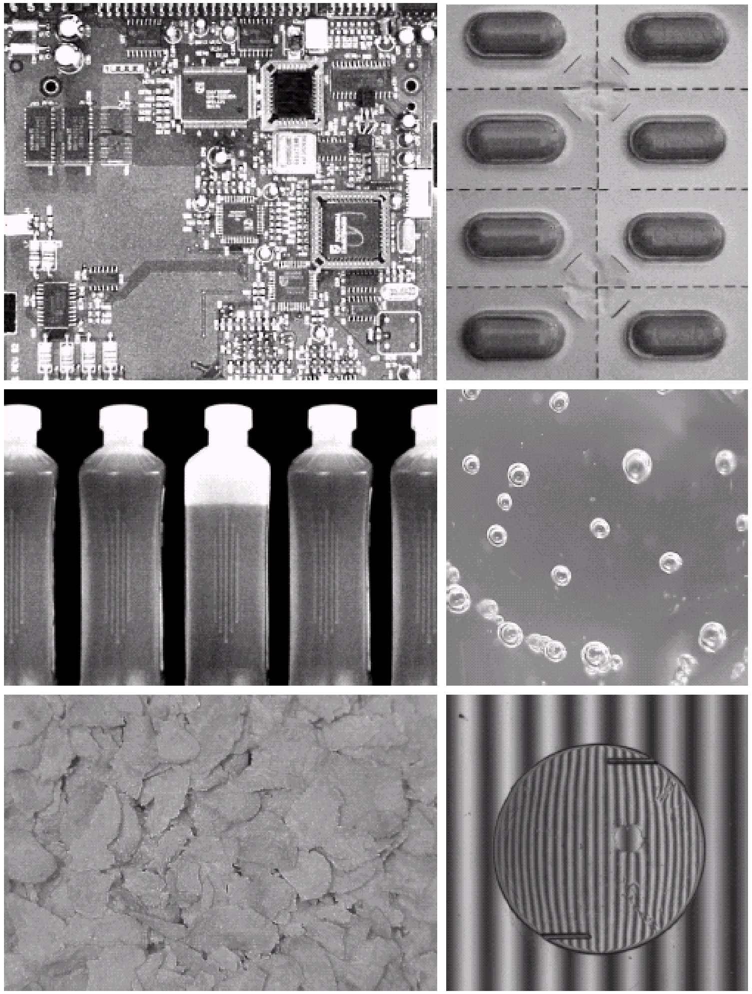


Applications: Industrial Inspection

- Human operators are expensive, slow and unreliable.

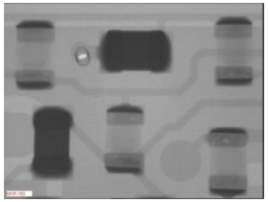
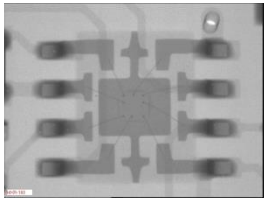
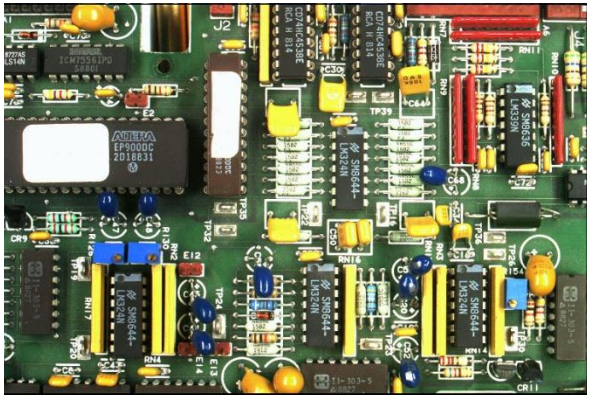
- Make machines do the job instead.

- Industrial vision systems are used in all kinds of industries.



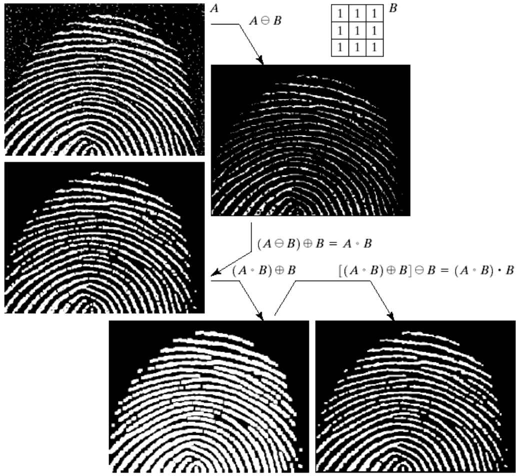
Applications: PCB Inspection

- Printed Circuit Board (PCB) inspection.
  - Machine inspection is used to determine that all components are present and that all solder joints are acceptable.
  - Both conventional imaging and x-ray imaging are used.

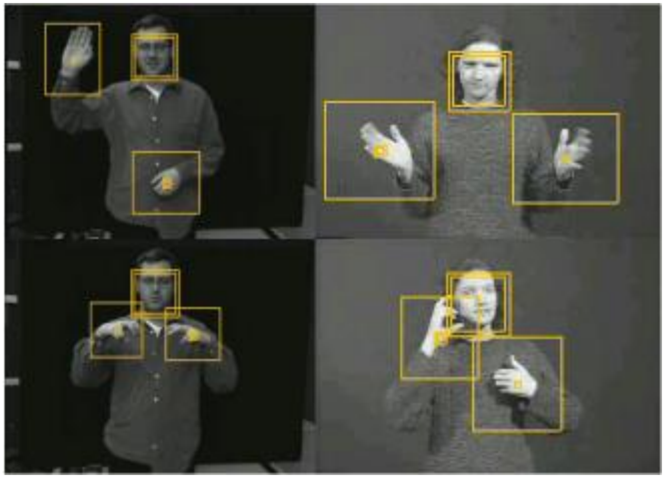

Applications: Law Enforcement

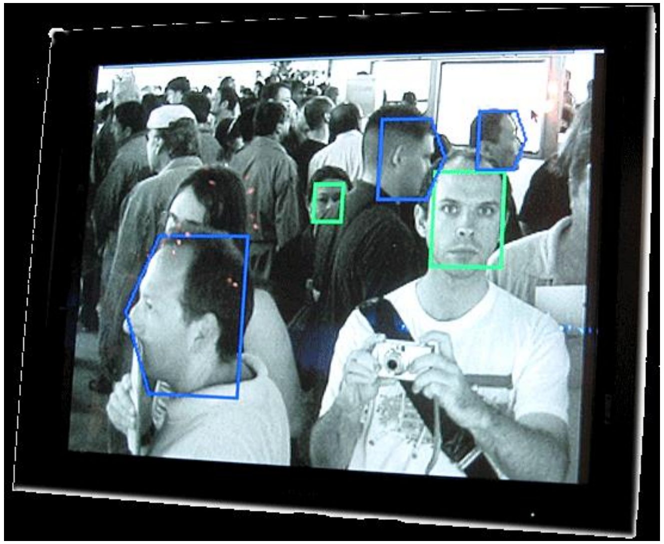
- Image processing techniques are used extensively by law enforcers.
    - Number plate recognition for speed cameras/automated toll systems.
    - Fingerprint recognition.



Examples: HCI

- Try to make Human Computer Interaction (HCI) more natural.
  - Face recognition.
  - Gesture recognition.



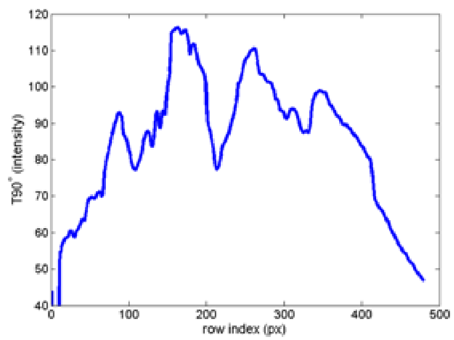


Applications: video frame stabilization

$$SSD(u)=\sum_x (I(x)-J(x-u))^2$$

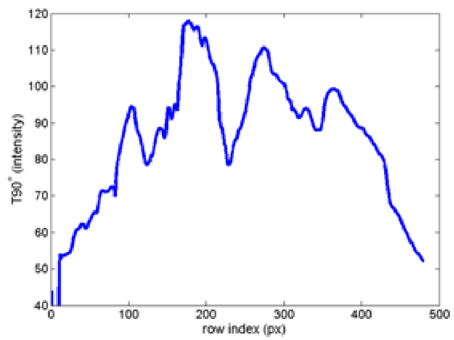
Video frame t

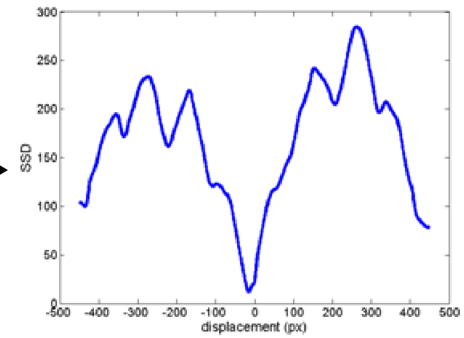
I: vertical projection



Video frame t+1

J: vertical projection





$$\Delta u_{SSD} = \arg\min_u SSD(u) - N_{PA}$$

Stabilized frame t+1

DIGITAL IMAGING FUNDAMENTALS

MTSC 887 – Sokratis Makrogiannis
Delaware State University

Lecture Outline

- Goal: Introduction to basic concepts of image processing
- Topics:
  - Human vision
  - Electromagnetic spectrum
  - Imaging sensors
  - Image sampling and quantization

Human Vision

- Eye structure

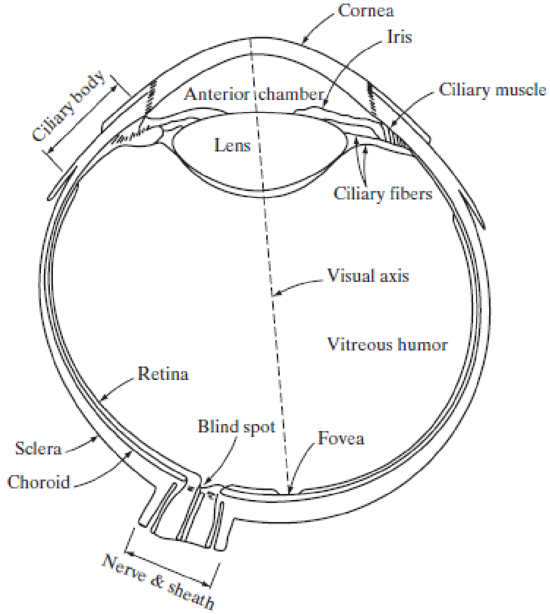


FIGURE 2.1 Simplified diagram of a cross section of the human eye.

- The lens changes shape to achieve proper focus

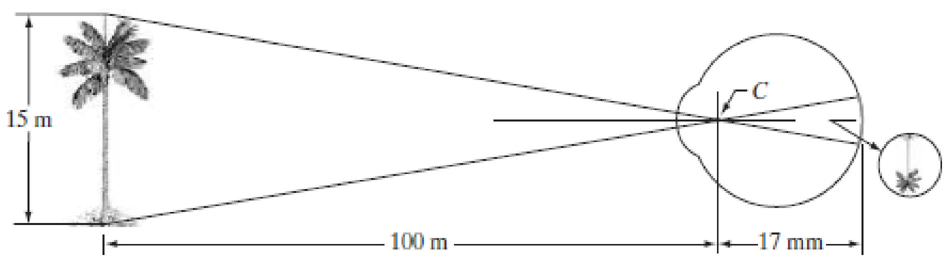


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

Light and the Electromagnetic Spectrum

- Electromagnetic waves: propagating sinusoidal waves with wavelength λ.

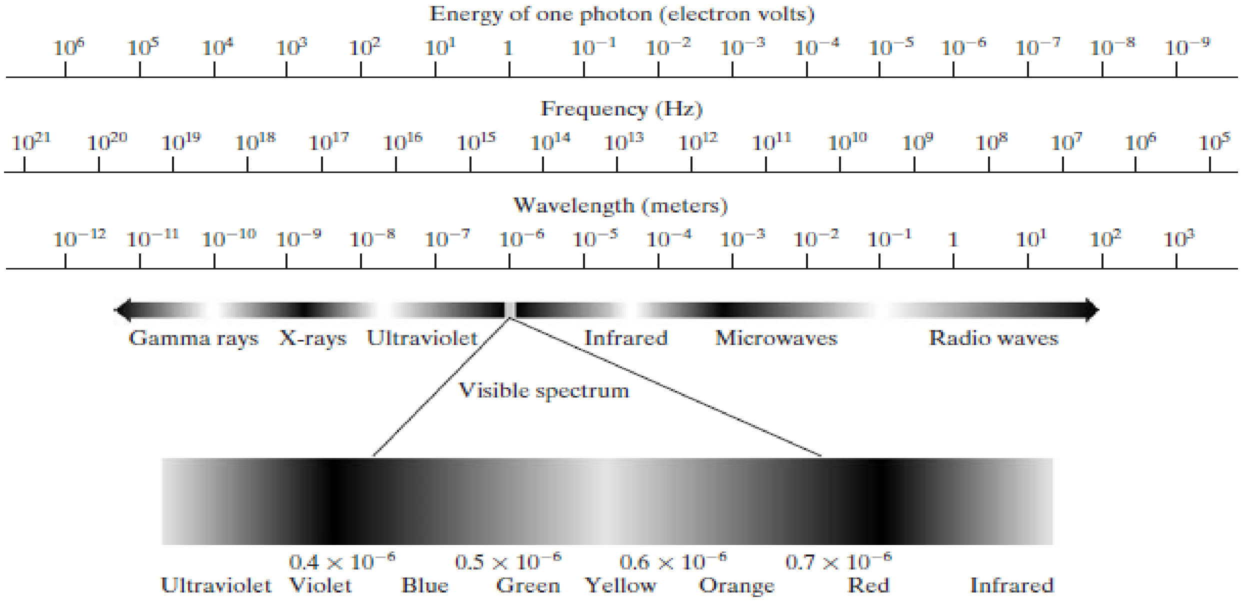


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

Light and the Electromagnetic Spectrum

Wavelength λ and frequency ν are related by the expression

$$\lambda = \frac{c}{\nu}$$

where c is the speed of light $(2.998 \times 10^8 m/s)$.
The energey E of electromagnetic waves is given by

$$E = h \cdot \nu$$

where h is Planck's constant.

Light and the Electromagnetic Spectrum

- Definitions:
- Monochromatic: light void of color.
- Chromatic light: 0.43 to 0.79nm.
- Radiance: total amount of energy flowing from source.
- Luminance: amount of energy from a light source perceived by observer.
- Brightness: subjective descriptor of light perception that is practically impossible to measure.
- Non-visible wavelengths:
- Gamma rays, X-rays, near infrared, far infrared.

Image Sensing and Acquisition

- Most of the image acquisition processes follow the model of "illumination" source and "scene".
- The "illumination" source radiates energy that is reflected or absorbed by elements of the "scene".
- Illumination sources may be a visible light source, infrared, X-ray transmitter, MRI coil, an ultrasound probe, etc.
- The scenes can be human cells, buried rock formations, a meteorite, a human brain or liver, a human face, indoor/outdoor landscapes, etc.

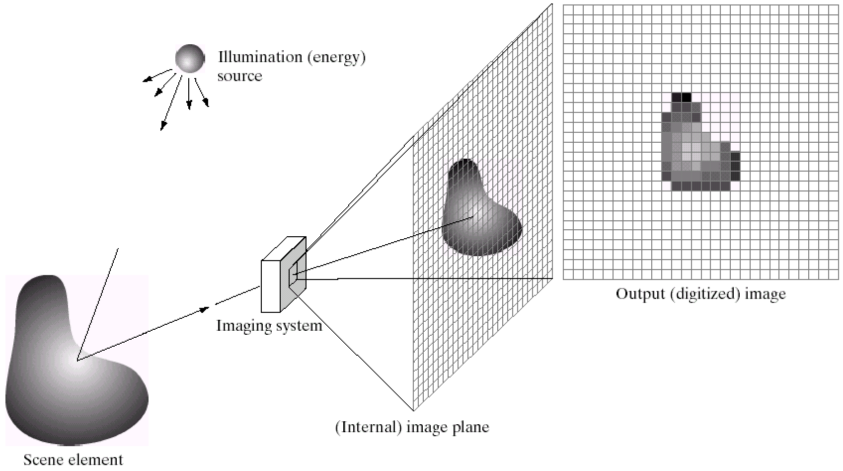


Image Sensing and Acquisition

- Incoming energy is transformed into voltage by sensors that respond to the specific type of energy that is detected.
- The voltage is then converted into digital signal by a digitization system.

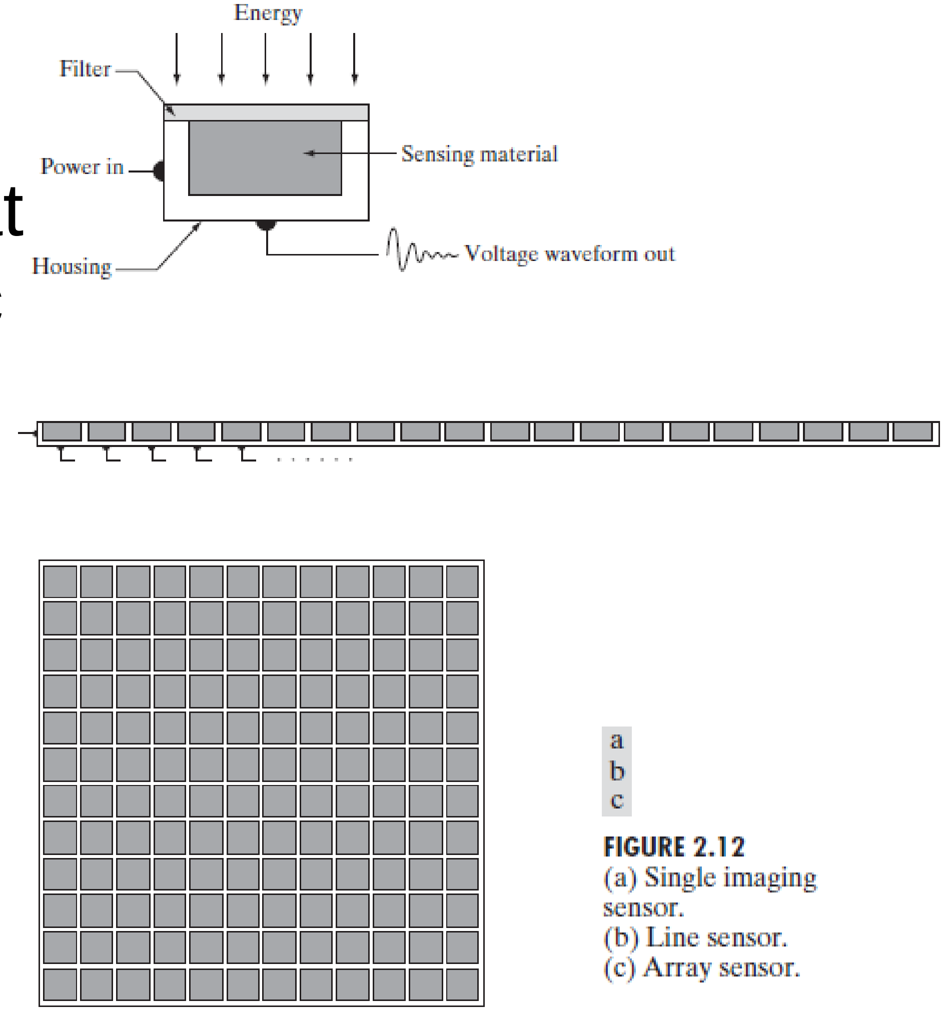
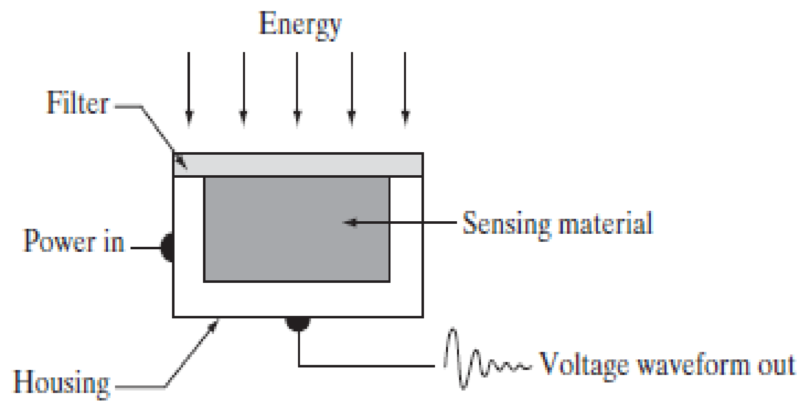
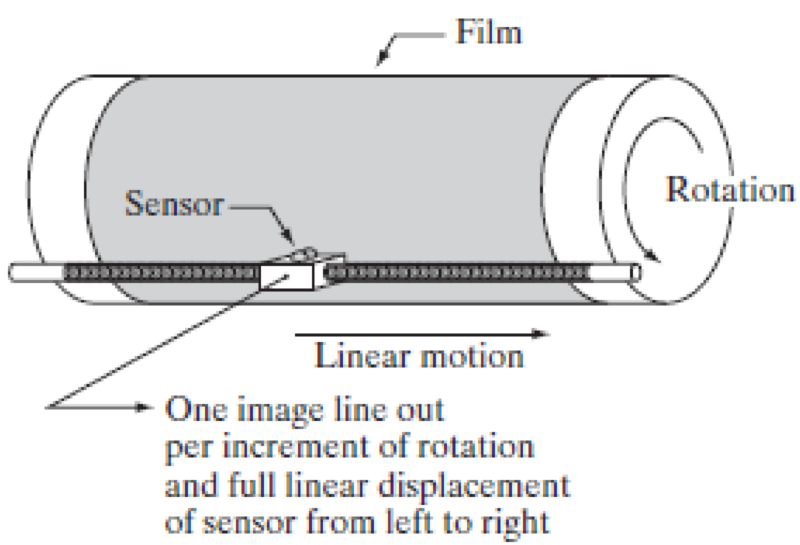


FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

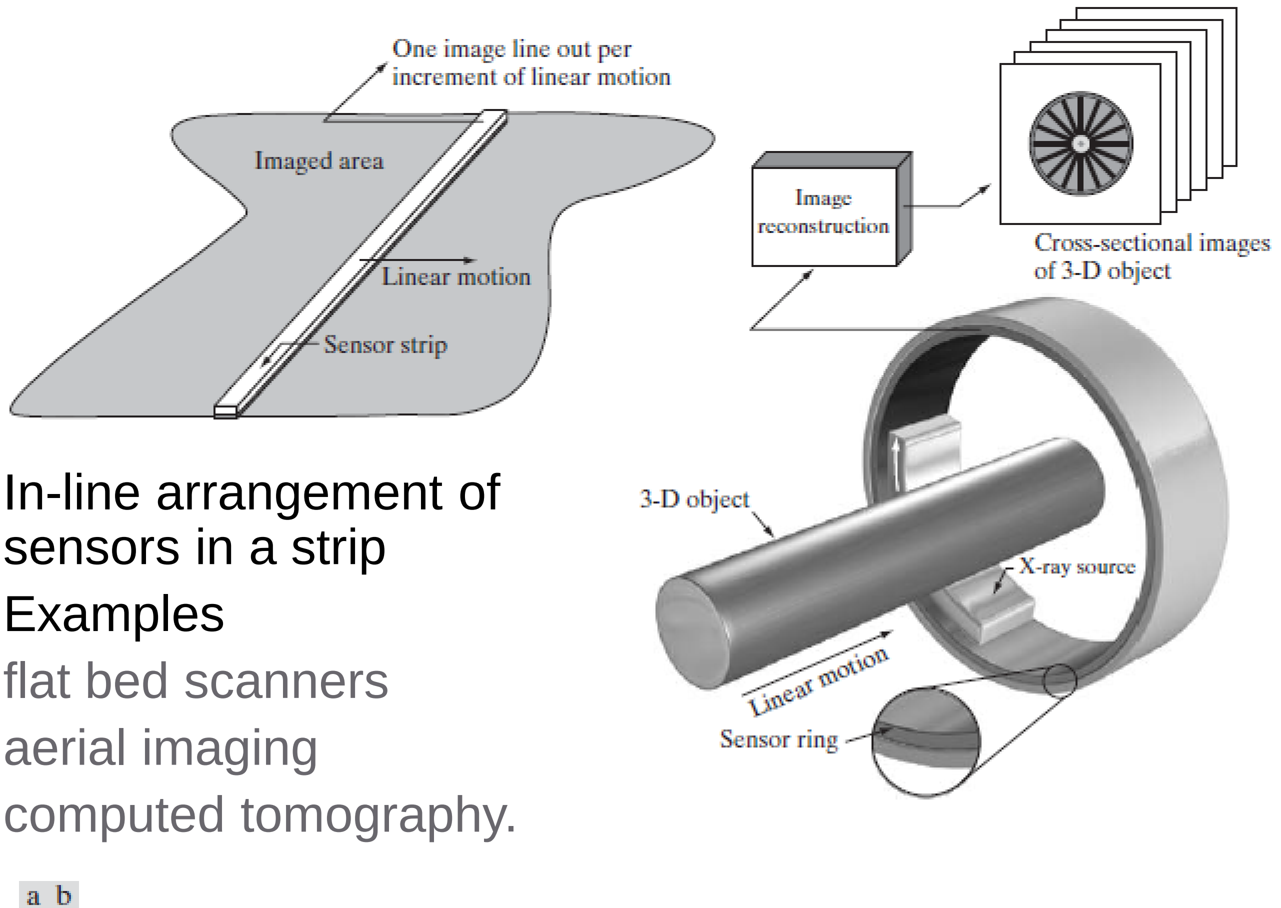
Single Sensor

- Main idea: incoming energy transformed into voltage.
- Example: photodiode, that converts light into electrical signal.
- A 2D image can be acquired using mechanical motion in two dimensions





Sensor Strips



- In-line arrangement of sensors in a strip
- Examples
- flat bed scanners
- aerial imaging
- computed tomography.

FIGURE 2.14 (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

Sensor Array

- Individual sensors are arranged in 2D arrays.
- Examples
  - Digital cameras that use CCD arrays
  - Ultrasonic devices.
- Response of CCD sensor is proportional to the integral of light energy projected onto sensor's surface.

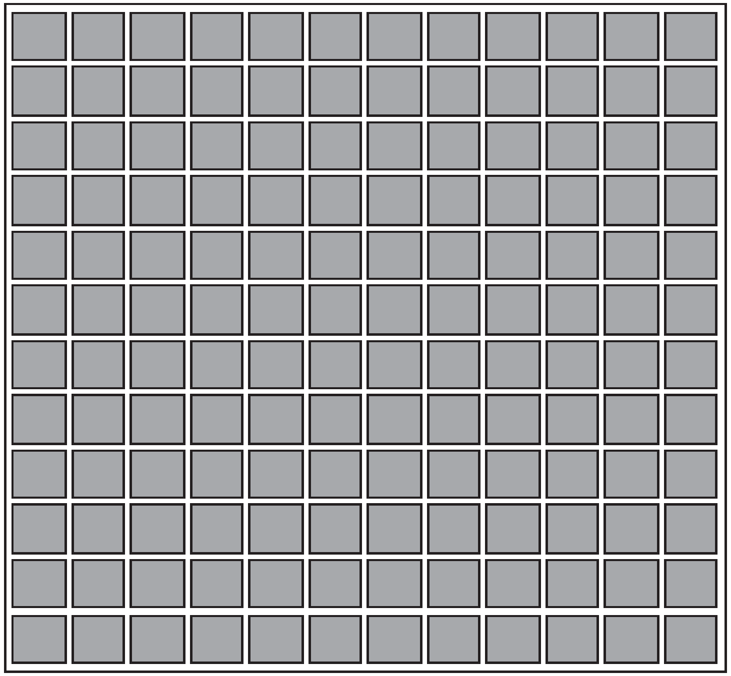


Image Formation Model

2D images are denoted by two-dimensional functions $f(x, y)$ with spatial coordinates (x, y).

Because values of f are propotional to energy detected by the sensor we have that

$$0 < f(x) < \inf .$$

The function $f(x)$ has two components

1. amount of source illumination incident on scene, denoted by $i(x, y)$, **illumination**.

2. amount of illumination reflected by objects in the scene, denoted by $r(x, y)$, **reflectance**.

Then we have that

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

with $0 < i(x, y) < \inf$ and $0 < r(x, y) < 1$.

Image Formation Model

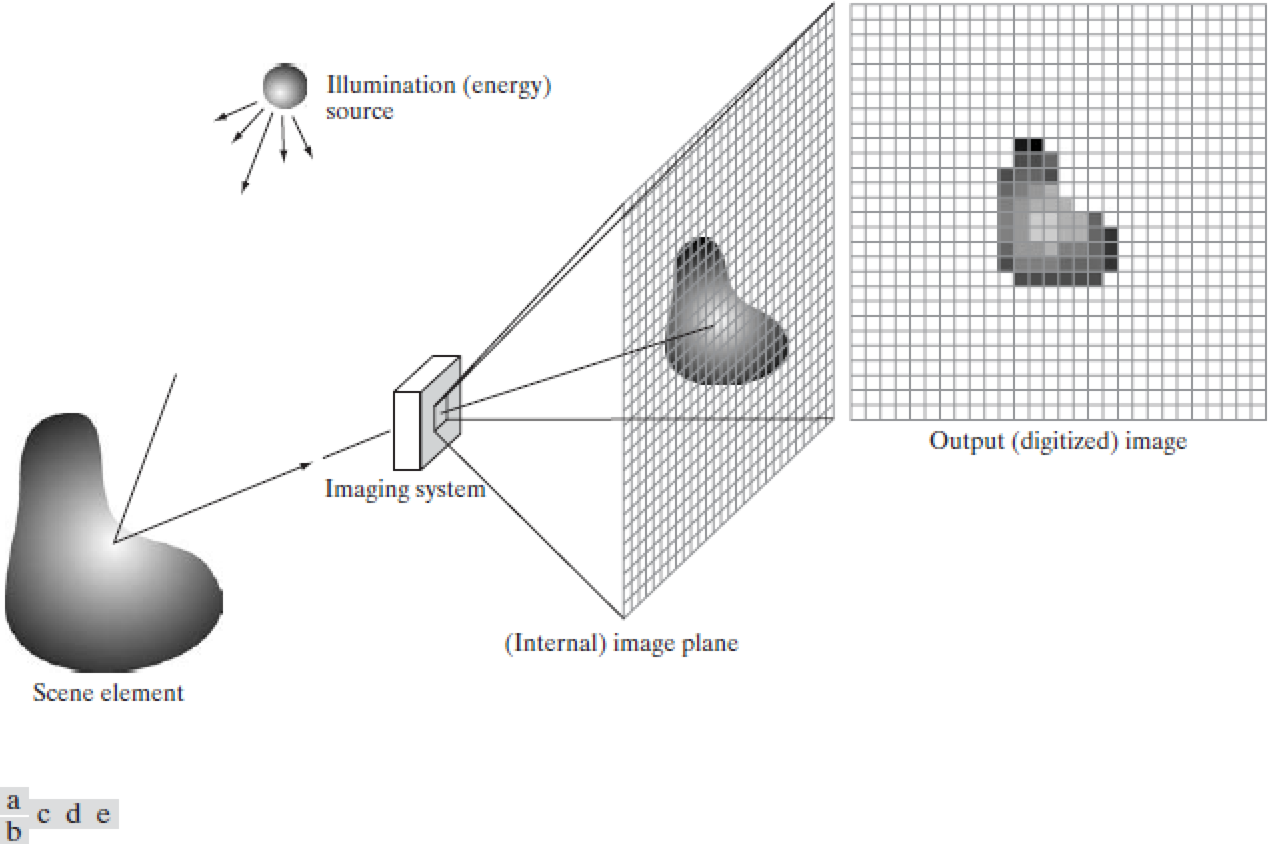
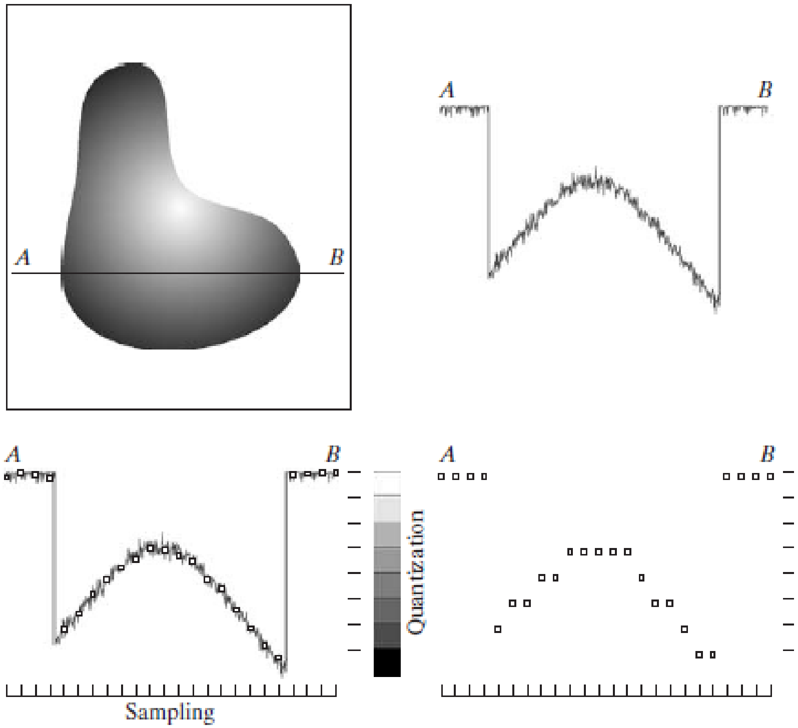


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image Sampling and Quantization

- Sensors produce an analog signal output related to the physical phenomenon being sensed.
- An acquired scene is continuous both in terms of coordinates and amplitude values.
- To create a digital image, two stages are followed
  - Sampling
  - 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.

Image Sampling and Quantization

- Sampling is the digitization of the coordinates.
- Quantization is the digitization of the amplitude.

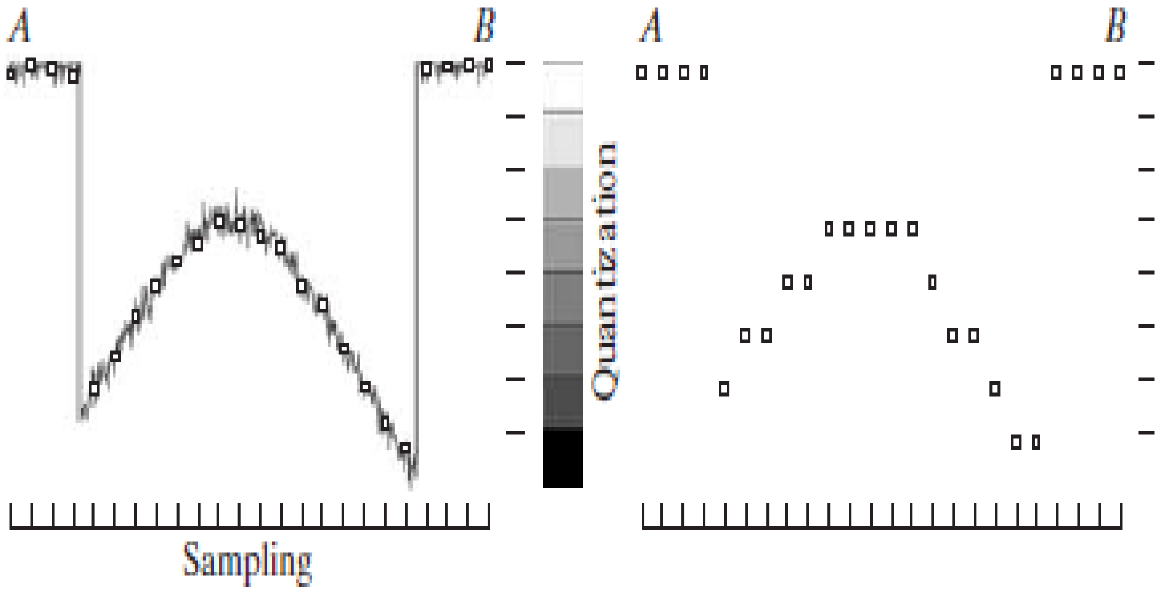


Image Sampling and Quantization

- Sampling and Quantization stages depend on sensor arrangement
- Sensors arrangements vs. digitization factors
  - Single sensor mechanical motion: number of mechanical increments.
  - Sensing strip: number of sensors in strip, number of mechanical increments.
  - Sensing array: number of sensors establishes sampling limits in both directions.

Image Sampling and Quantization

- End result depends on both sampling and quantization stages.

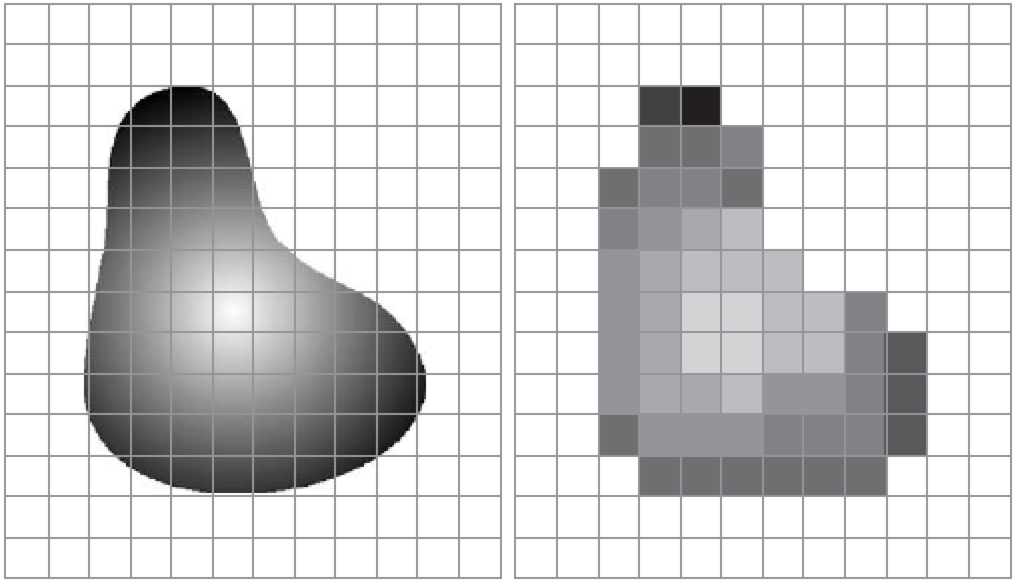


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Digital Image Representation

After acquisition is completed, a 2D digital image is represented by a 2D array.

Let the image be denoted by $f(x, y)$ and let the array consist of M rows and N columns.

The pair (x, y) denotes the discrete coordinates, or indices.

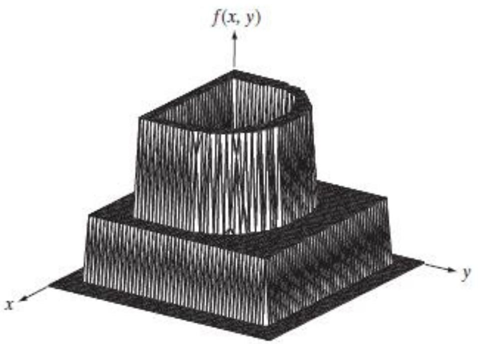
The index values are $x = 0, 1, 2, ..., M - 1$ and $y = 0, 1, 2, ..., N - 1$.

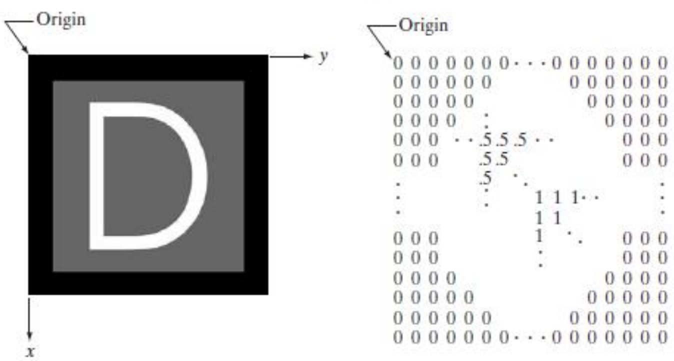
The image intensity at the origin is $f(0, 0)$.

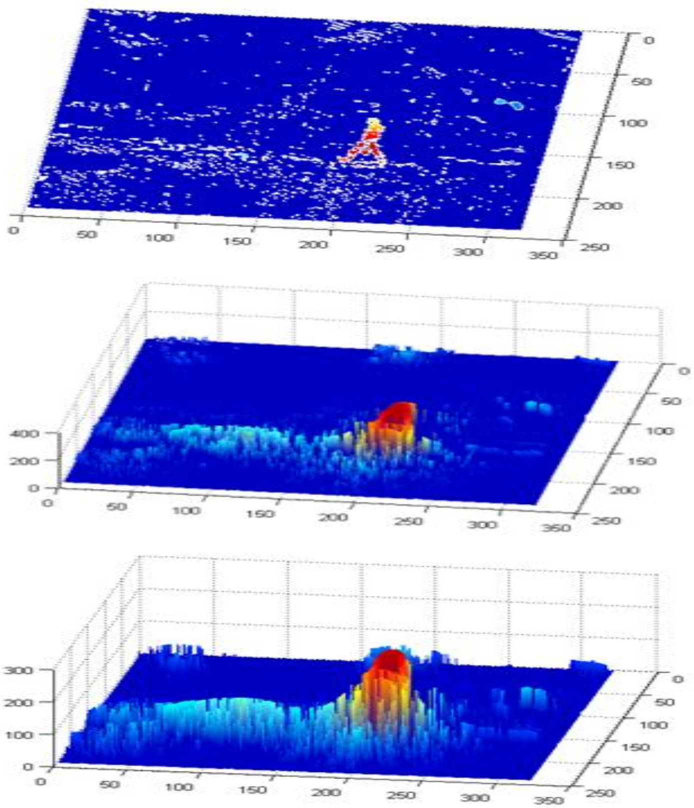
The image plane spanned by the coordinates is called the spatial domain, and x and y are called the spatial coordinates.

Digital Image Representation

- A 2D image can be displayed as
  - A surface
  - Visual intensity array
  - Numerical array







Digital Image Representation

- Numerical array representation is used for algorithm development and processing.

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

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Digital Image Representation

During digitization the number of rows M, columns N and discrete intensity levels L have to be determined.

The image dimensions M and N just need to be positive integer numbers.

The number of intensity levels needs to be a power of 2, mainly due to data storage protocols and requirements.

So $L = 2^k$ and the intensity can have values in $0, 1, ..., L - 1$.

Digital Image Representation

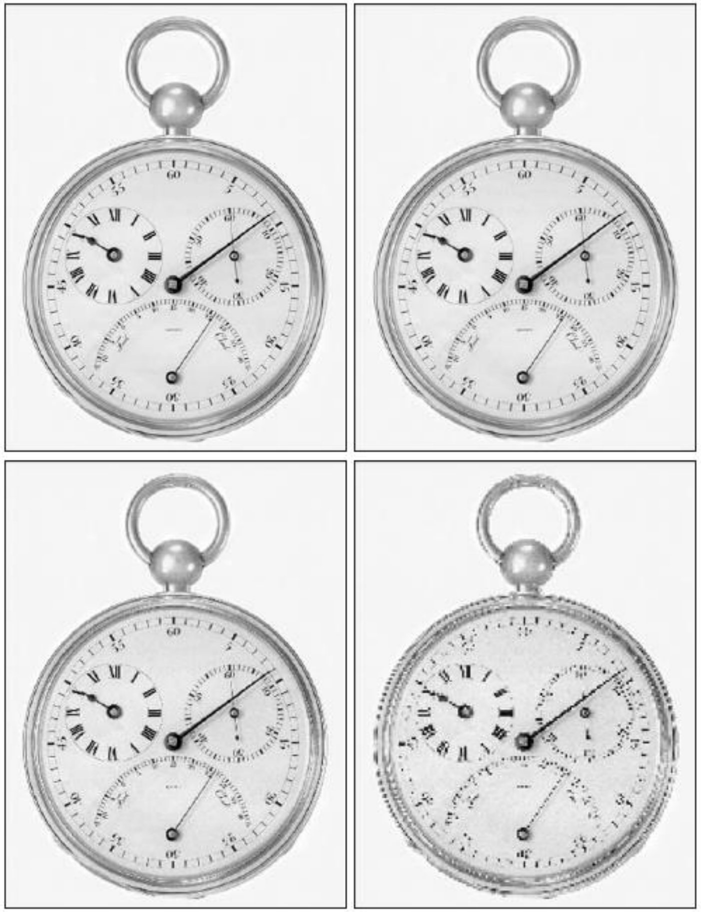
- Definitions:
  - The ratio of maximum intensity to the minimum intensity of an imaging system is called the dynamic range.
  - Upper limit is defined by saturation and minimum level is defined by noise level.
  - Image contrast is the difference in intensity between the highest intensity and the lowest intensity levels in an image.

The number b of bits required to store an image is

$$b = M \cdot N \cdot k$$

Spatial Resolution

- Spatial resolution is a measure of the smallest discernible detail in the image.
- To be meaningful it needs to be stated using units of distance.
- Examples
  - 75 dots per inch
  - 1024x1024 px, field of view 1000mmx1000mm



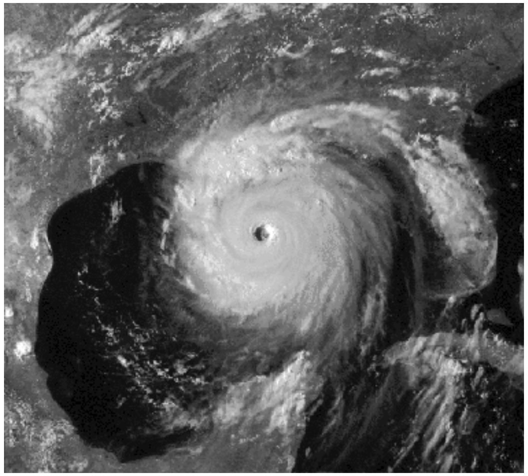
a b
c d

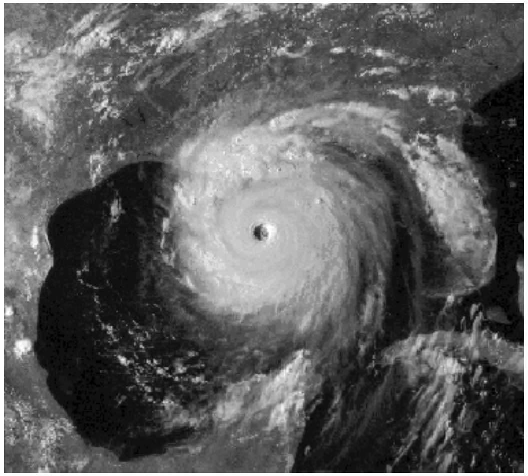
FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

Spatial Resolution

- Effect of decreasing spatial resolution on image quality.

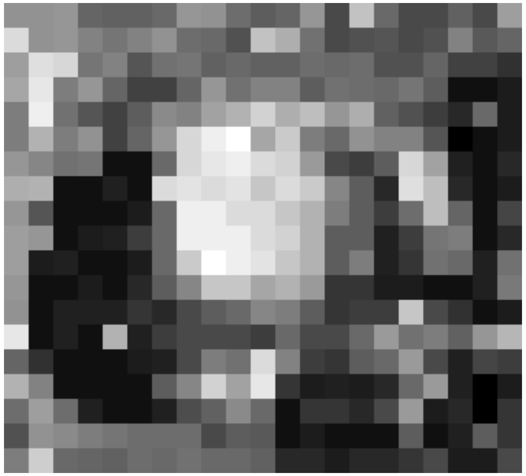
Original
scale
(1153x1281)





Rescale
factor: 1/4

Rescale
factor: 1/16



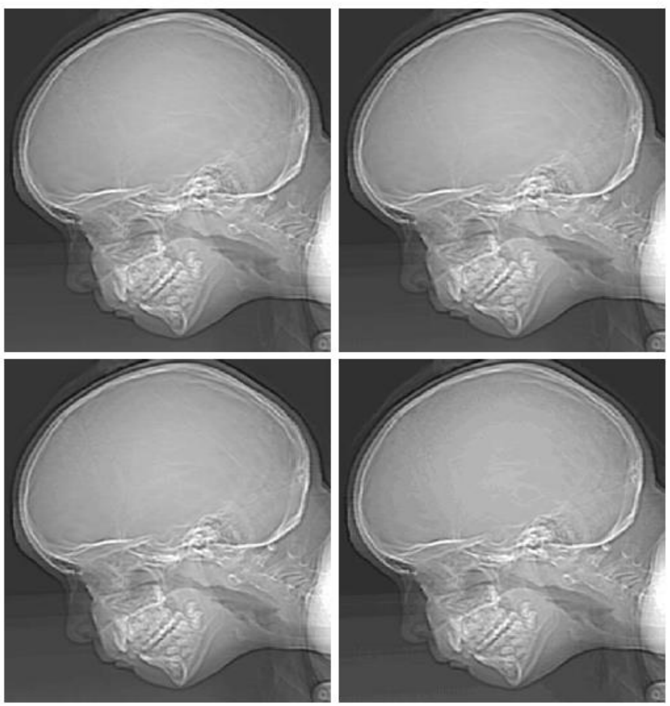
Rescale
factor: 1/64

Intensity Resolution

- Refers to the smallest discernible change in intensity level.
- Common intensity resolution for generic imagery is 8 bits.
- In medical applications on can encounter 10, 12 or 16 bit resolutions.

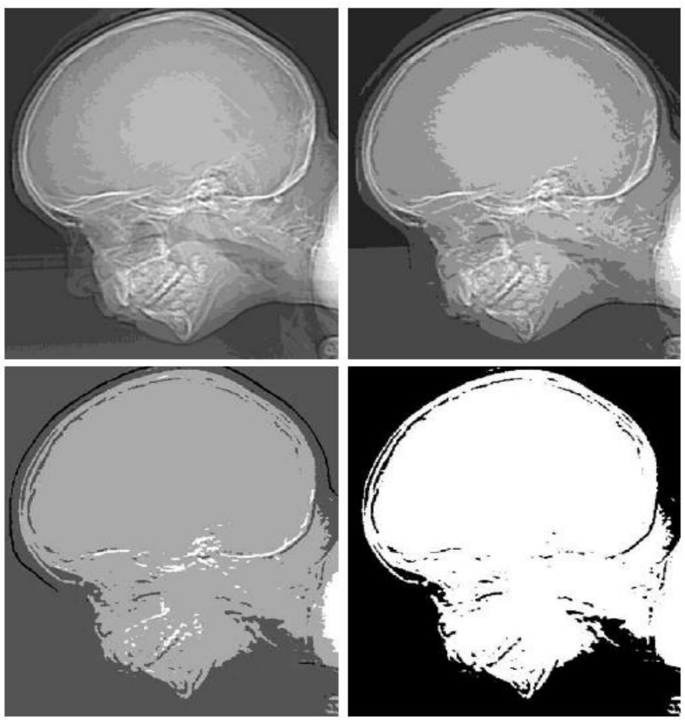
Intensity Resolution

- Effect of decreasing intensity resolution on image quality.



a b
c d

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



e f
g h

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

Image Interpolation

- Interpolation is the process of using known data to estimate values to unknown locations.
- In image processing it finds applications in image resizing, zooming, geometric transformations and image registration.
- Three popular methods of interpolation are
  - Nearest neighbor
  - Bilinear
  - Bicubic

Image Interpolation – Nearest Neighbor

- Algorithm
  - Generate output image array
  - Scan each pixel location.
    - Calculate the corresponding spatial coordinates in the original image
    - Find the nearest pixel in original image – nearest neighbor
    - Assign intensity of nearest neighbor to the output pixel

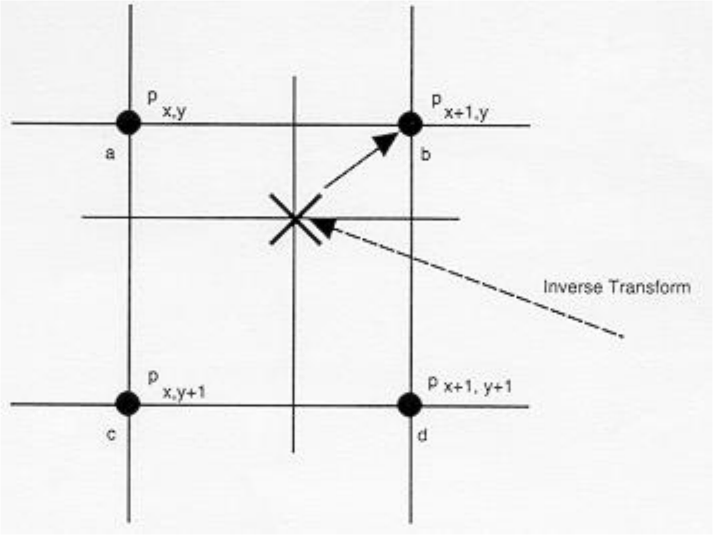
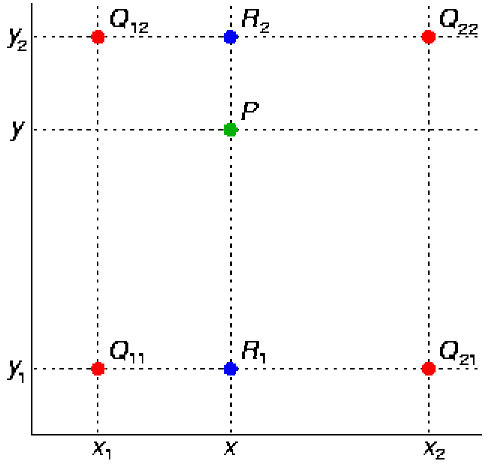


Image Interpolation – Bilinear

- Algorithm
  - Generate output image array
  - Scan each pixel location.
    - Calculate the corresponding spatial coordinates in the original image
    - Use Manhattan Distance-weighted intensity sum of the 4-neighbors.

Alternatively, the interpolant can be expressed as

$$v(x,y) = ax + by + cxy + d$$



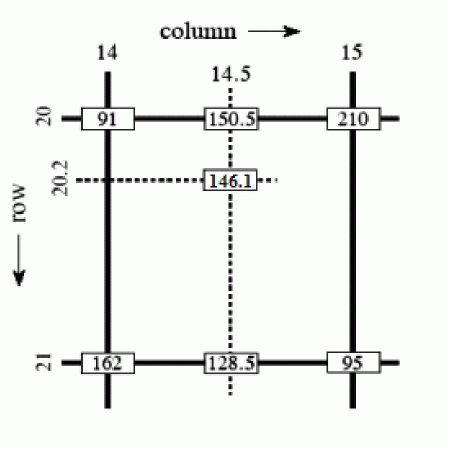


Image Interpolation – Bicubic

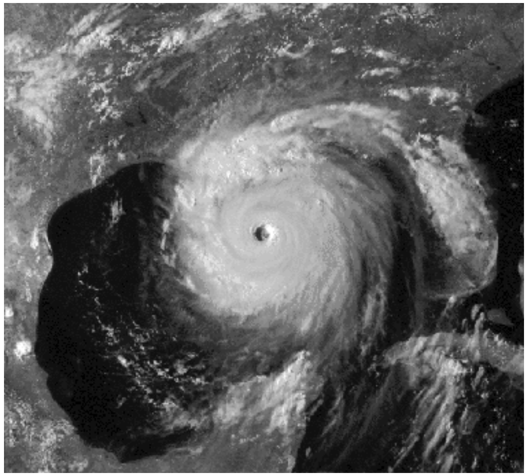
- Similar algorithm with one difference
- Uses the 16-nearest neighbors of a point *P(x,y)* to estimate intensity.

$$v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

- The coefficients can be found analytically or computationally.
- Bilinear interpolation is a special case when the upper limits of *i* and *j* are 1.

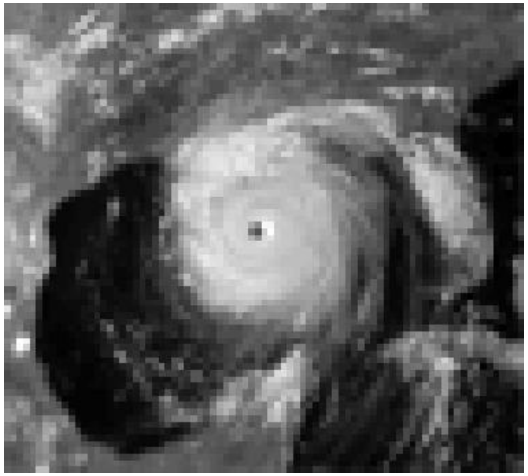
Image Interpolation Methods- Comparisons

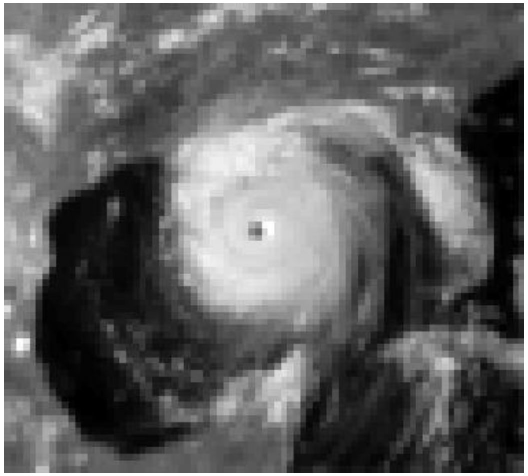
Original
scale
(1153x1281)



Nearest Neighbor,
Rescale
factor: 1/16

Bicubic,
Rescale
factor: 1/16





Bilinear,
Rescale
factor: 1/16

DIGITAL IMAGING FUNDAMENTALS

MTSC 887 – Sokratis Makrogiannis
Delaware State University

Lecture Outline

- Goal: Introduction to basic concepts of image processing
- Topics:
  - Spatial relationships between pixels
  - Mathematical tools in digital image processing

Image file formats

- Many image formats adhere to the following simple model:
  - Header
  - Data (line by line, no breaks between lines).

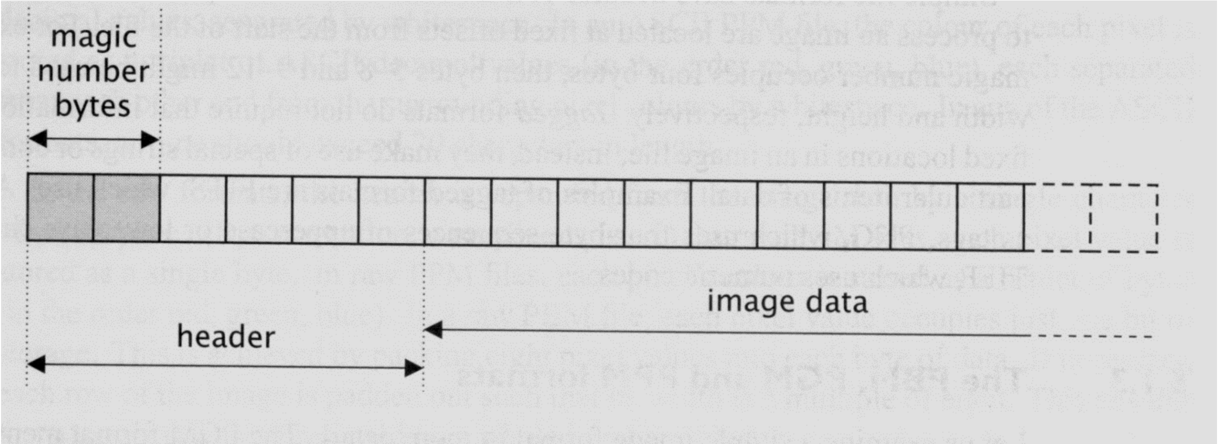
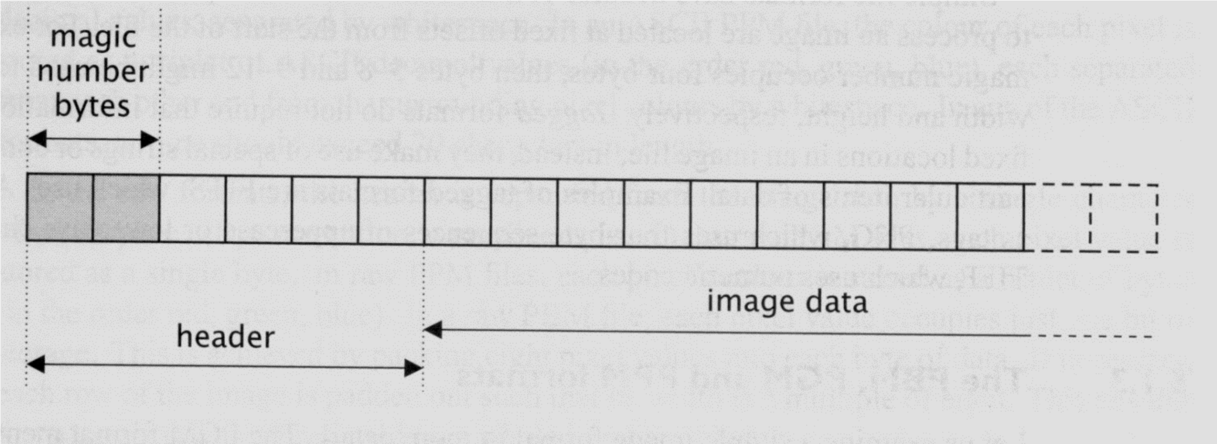


Image file formats (cont.)

- Header contains at least:
  - A signature or “magic number” (i.e., a short sequence of bytes for identifying the file format).
  - The width and height of the image.



Common image file formats

- PGM (Portable Gray Map)
- PNG (Portable Network Graphics)
- GIF (Graphic Interchange Format) –
- JPEG (Joint Photographic Experts Group)
- TIFF (Tagged Image File Format)
- FITS (Flexible Image Transport System)

PGM format

- A popular format for grayscale images (8 bits/pixel)
- Closely-related formats are:
  - PBM (Portable Bitmap), for binary images (1 bit/pixel)
  - PPM (Portable Pixelmap), for color images (24 bits/pixel)

```
P2
# a simple PGM image
7 7 255
 120 120 120 120 120 120 120
 120 120 120  33 120 120 120
 120 120 120  33 120 120 120
 120  33  33  33  33  33 120
 120 120 120  33 120 120 120
 120 120 120  33 120 120 120
 120 120 120 120 120 120 120
```

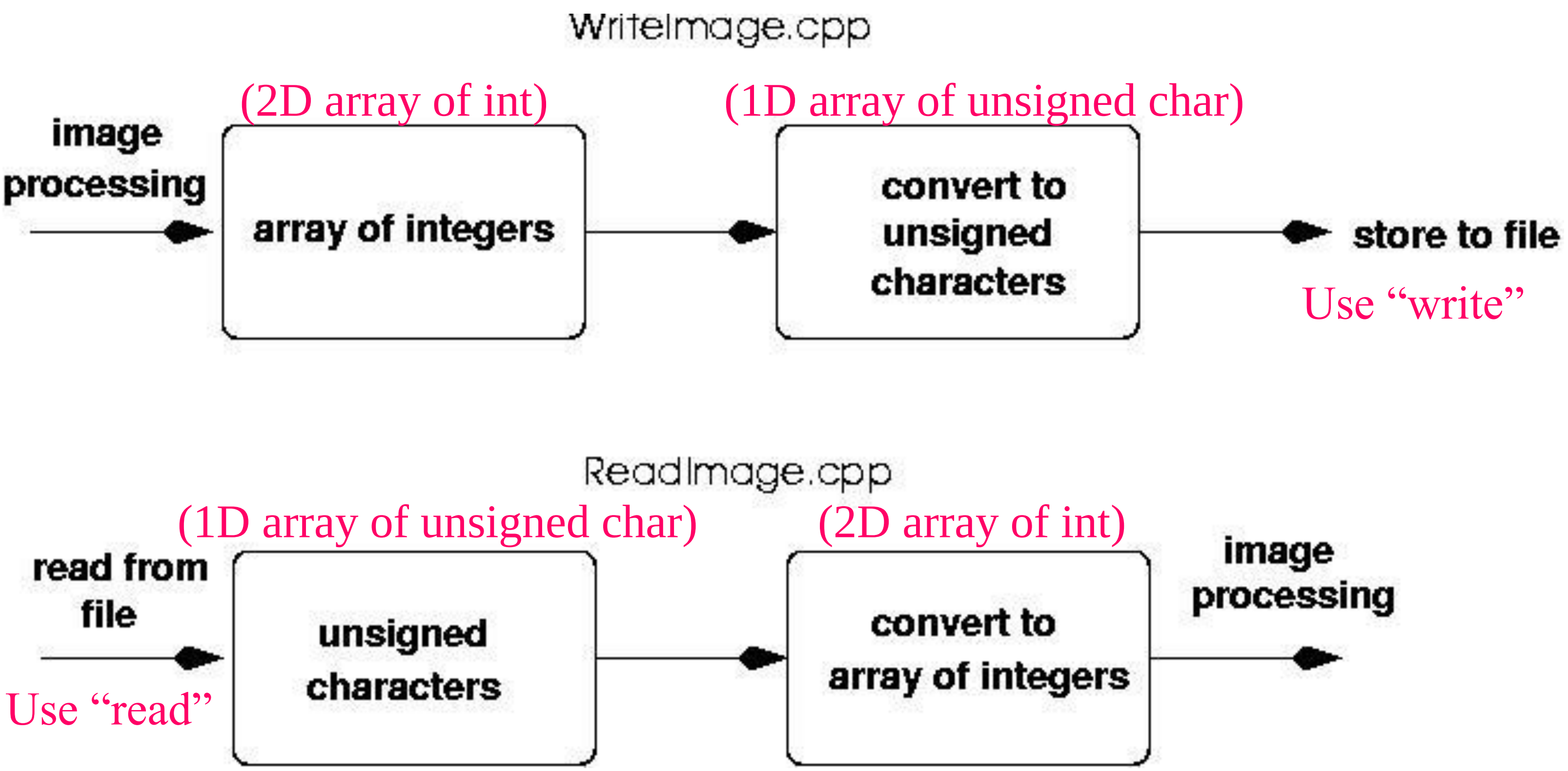
```
P5
# a simple PGM image
7 7 255
xxxxxxxxxx!xxxxxx!xxxx!!!!!xxxx!xxxxxx!xxxxxxxxxx
```

ASCII or binary (raw) storage

Signatures of the various PBM, PGM and PPM image formats.

Signature	Image type	Storage type
P1	binary	ASCII
P2	greyscale	ASCII
P3	RGB	ASCII
P4	binary	raw bytes
P5	greyscale	raw bytes
P6	RGB	raw bytes

Reading/Writing PGM images

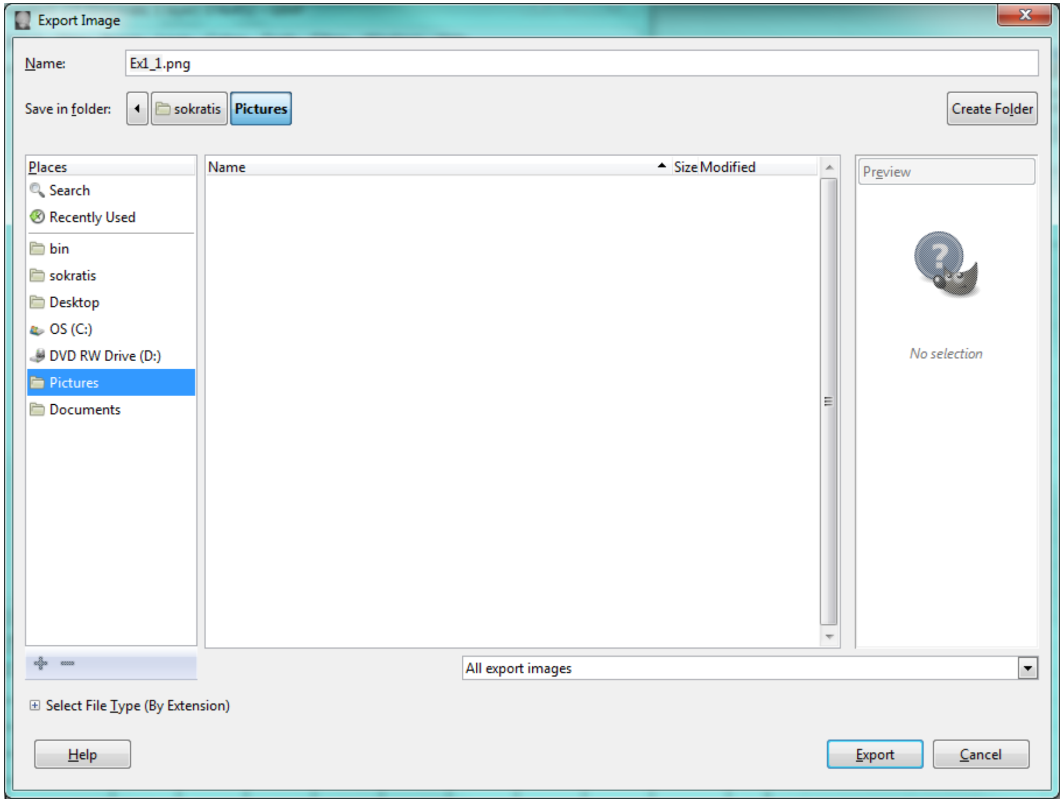


How do I “see” images on my computer?

- Linux:
  - display
  - Gimp
  - ImageJ
- Windows:
  - Gimp
  - Photoshop
  - Irfanview
  - ImageJ

How do I convert an image from one format to another?

- Use “save” or “export” option



Spatial Relationships between Pixels

- Pixel neighborhoods
- Adjacency, connectivity, regions and boundaries
- Distance measures

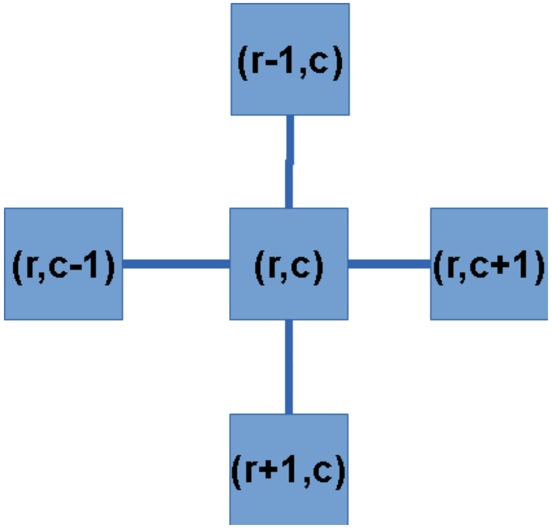
Pixel neighborhoods

Pixel $p(x, y)$:

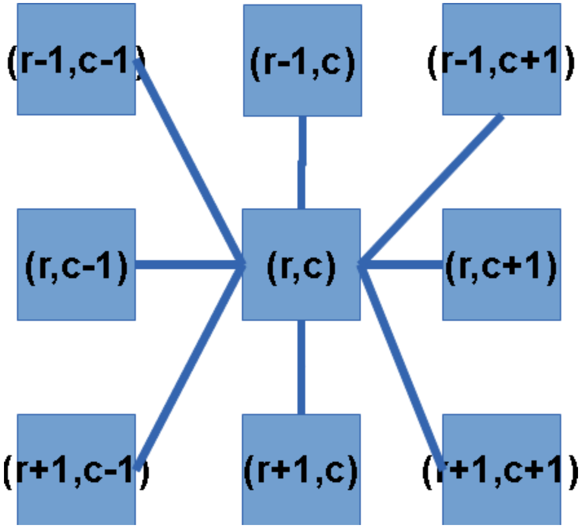
a) 4-neighbors $N_4(p) : (x+1, y), (x-1, y), (x, y+1), (x, y-1)$

b) 4-diagonal neighbors: $N_D(p) : (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$

c) 8-neighbors $N_8(p)$: $N_4(p)$ and $N_D(p)$



4-neighbors



8-neighbors

Adjacency Types

Let V be the set of intensity values used to define adjacency. For binary images usually $V = \{1\}$. Let p and q two pixels of the same image.

Adjacency types

- 4-adjacency: p and q are 4-adjacent if q is in $N_4(p)$.
- 8-adjacency: p and q are 8-adjacent if q is in $N_8(p)$.
- m-adjacency (mixed adjacency): q is in $N_D(p)$ and $N_4(p) \cap N_4(q)$ has no pixels with value V.

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

Pixels that are 8-adjacent but not 4-adjacent

Connectivity

Let V be the set of intensity values used to define adjacency. For binary images usually $V = \{1\}$. Let p and q two pixels of the same image.

- Digital path from $p(x, y)$ to $q(s, t)$: $(x, y), (x_1, y_1), \ldots, (x_k, y_k), (s, t)$, where $p_{i-1}(x_{i-1}, y_{i-1})$ and $p_i(x_i, y_i)$ are adjacent. Closed path if $(x, y) = (s, t)$
- Let S be a subset of pixels in an image. Pixels p and q are connected in S if there is a path between p and q with members in S.
- For a $p \in S$, the set of pixels that are connected to it in S is called a connected component of S. If S has one connected component, then S is a connected set.

Original image array

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

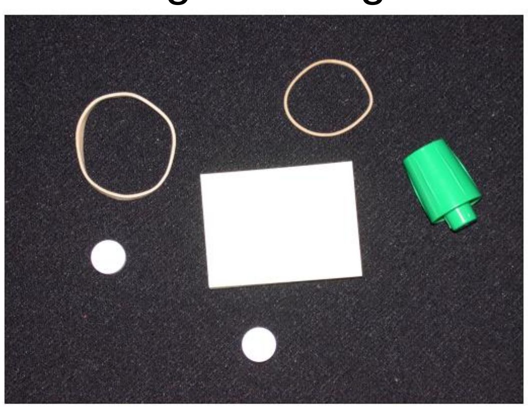
Connected components (4-adjacency)

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

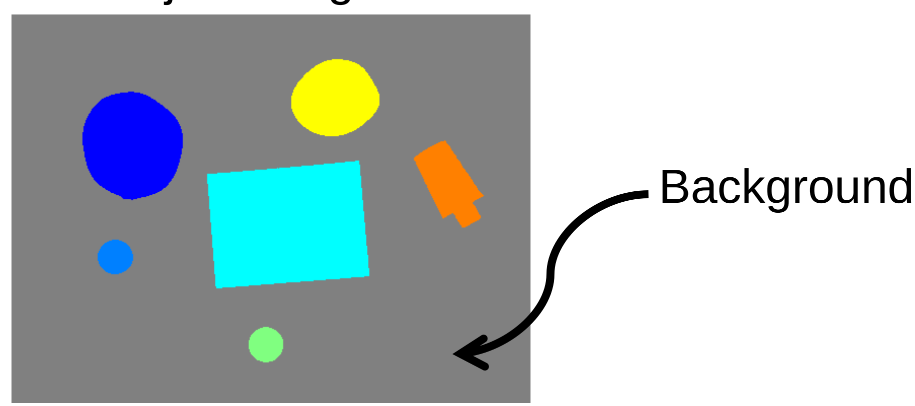
Regions

- Let R be a subset of pixels. If R is a connected set, then it is called a **region**.
- Regions R_i and R_j are adjacent if their union forms a connected set. Otherwise R_i and R_j are disjoint.
- Let k disjoint regions $R_k, k = 1, 2, ..., k$ R_u be the union of all k regions and R_u^c the complement of R_u. Then R_u: foreground and R_u^c: background.

Original Image



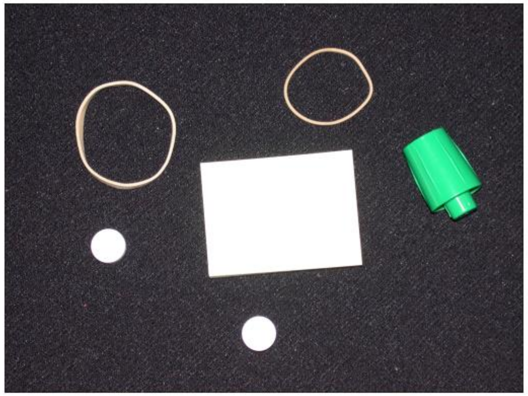
Disjoint Regions



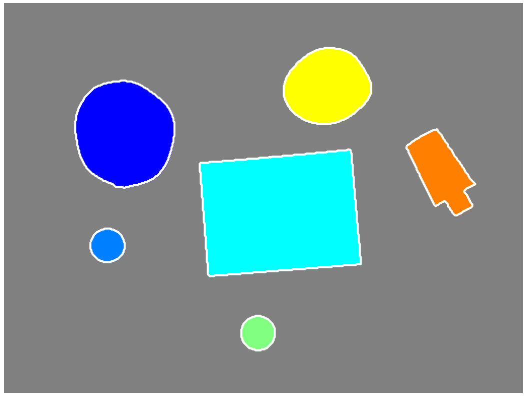
Boundaries

- Boundary of a region R is the set of points adjacent to points in the complement of R.
- The boundary is also called inner boundary.
- Outer boundary is the background's boundary.
- Many boundary following algorithms follow the outer boundary.
- Boundary is a mostly "global" concept.
- Edge is a "local" concept related to intensity level discontinuity at a point.

Original Image



Regions and Boundaries



Adjacency, Regions, Boundaries

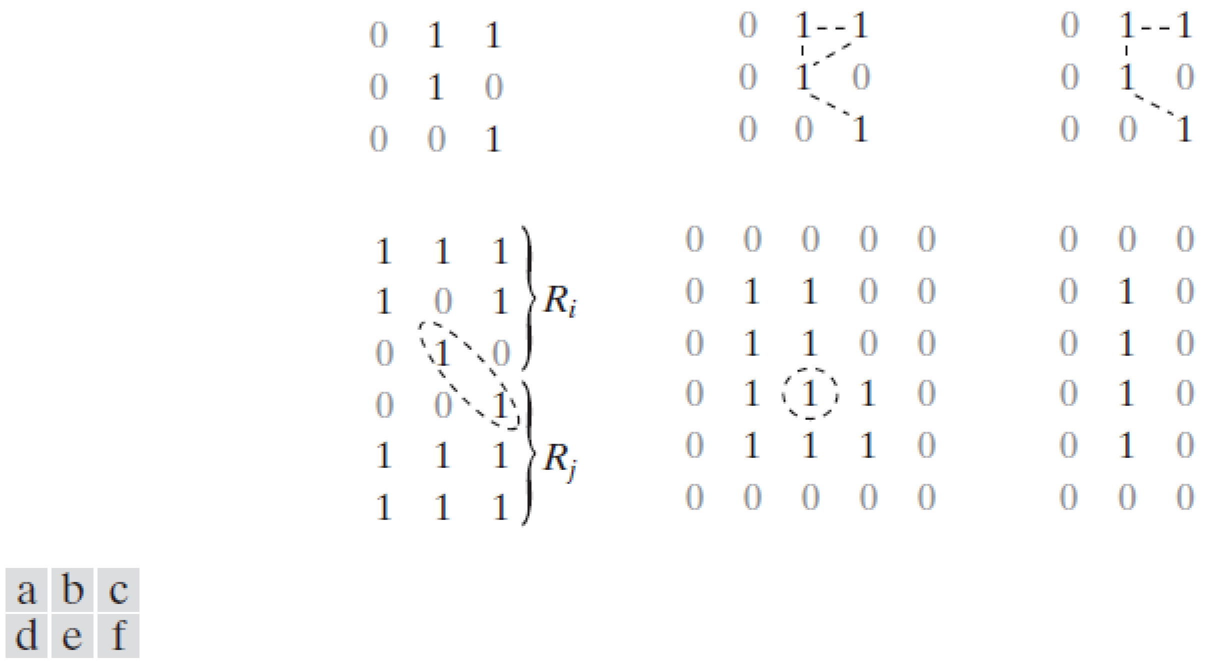


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions (of 1s) that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency 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.

Adjacency, Regions, Boundaries

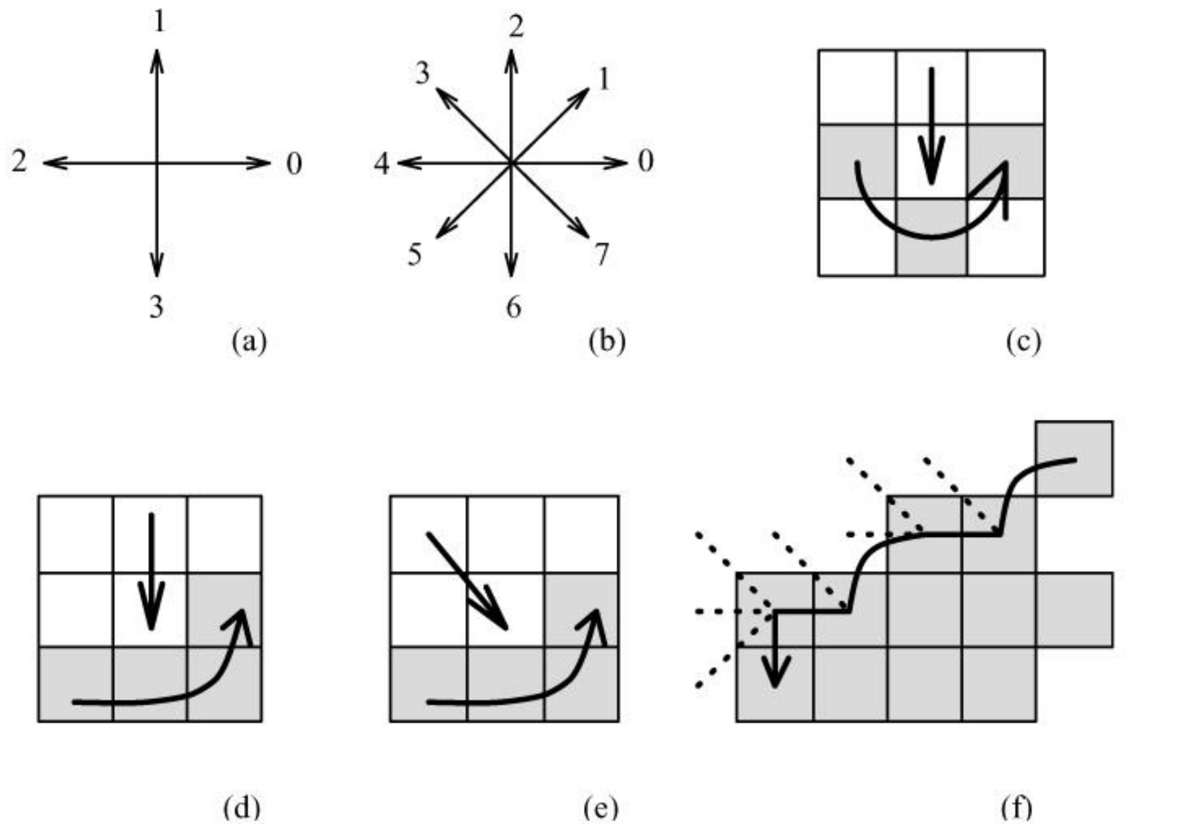


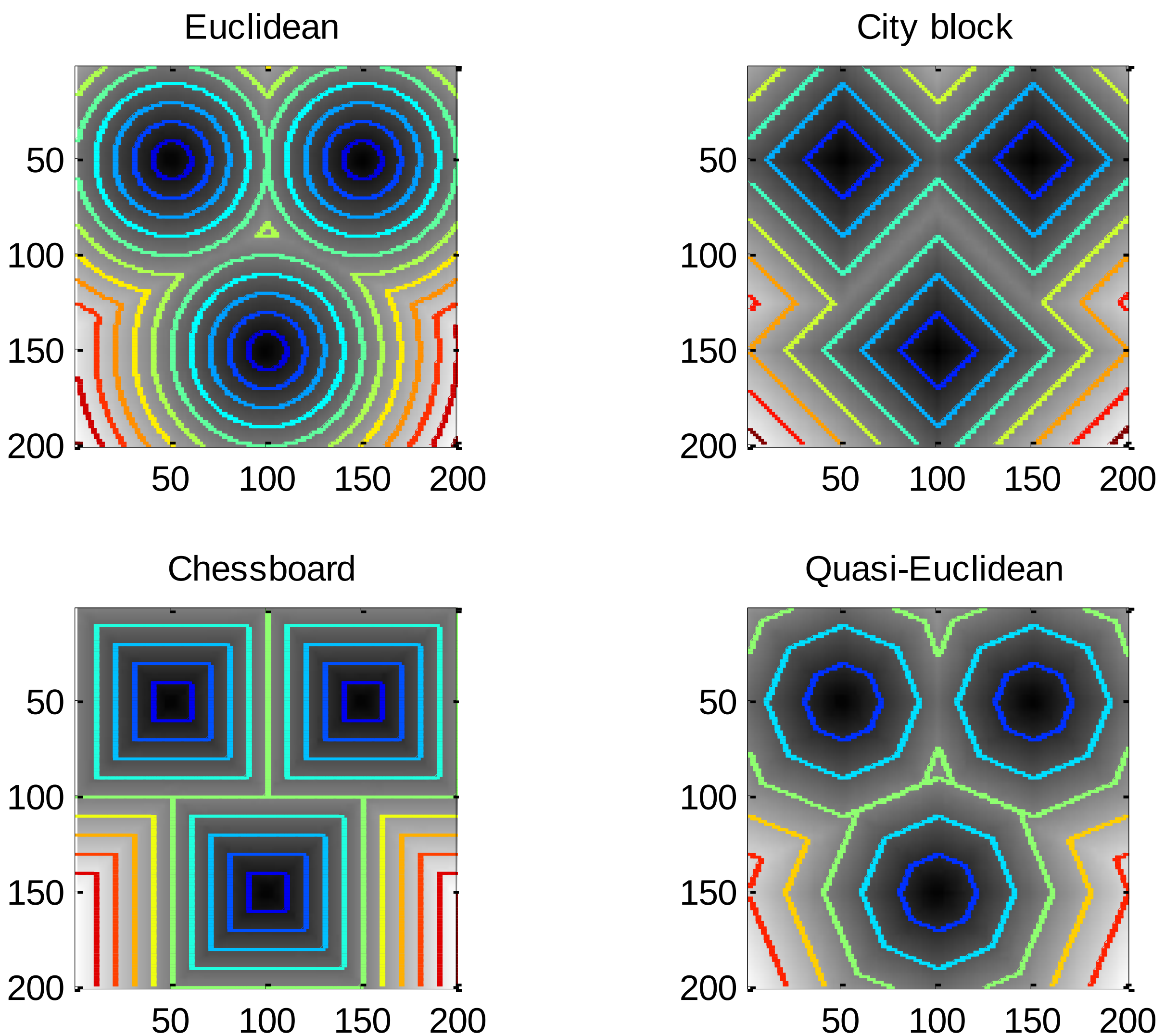
Figure 6.13: Inner boundary tracing. (a) Direction notation, 4-connectivity. (b) 8-connectivity. (c) Pixel neighborhood search sequence in 4-connectivity. (d), (e) Search sequence in 8-connectivity. (f) Boundary tracing in 8-connectivity (dotted lines show pixels tested during the border tracing). © *Cengage Learning 2015.*

Distance Measures

For pixels $p(x, y), q(s, t) and z(u, v)$ D is a distance function or metric if:
(a) $D(p, q) \geq 0$
(b) $D(p, q) = D(q, p)$
(c) $D(p, z) \leq D(p, q) + D(q, z)$

- Euclidean Distance: $D_e(p, q) = \sqrt{(x - s)^2 + (y - t))^2}$
- D_4 distance, city block distance, Manhattan distance: $D_4(p, q) = |x - s| + |y - t|$
- D_8 distance, or chessboard distance: $D_8(p, q) = max(|x - s|, |y - t|)$
- D_4 and D_8 distances do not depend on any paths between points. D_m is the shortest m-path between two points.

Distance Measures



Mathematical Tools in Digital Image Processing

- Array vs. Matrix Operations
- Linear vs. Non-linear Operations
- Arithmetic Operations
- Set and Logical Operations
- Spatial Operations
- Vector and Matrix Operations
- Probabilistic Methods

Array vs. Matrix Operations

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

Array Product

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

Matrix Product

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

Linear vs. Nonlinear Operations

- General operator

$$H = H\left[f(x,y)\right] = g(x,y)$$

H is a linear operator if

$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = a_i H\left[f_i(x,y)\right] + a_j H\left[f_j(x,y)\right] = a_i g_i(x,y) + a_j g_j(x,y),$$

where $a_i, a_j, f_i(x,y), f_j(x,y)$ arbitrary constants and images.

- The two properties are called additivity and homogeneity.

Arithmetic Operations

Pixel-wise operations

$$s(x,y) = f(x,y) + g(x,y)$$

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

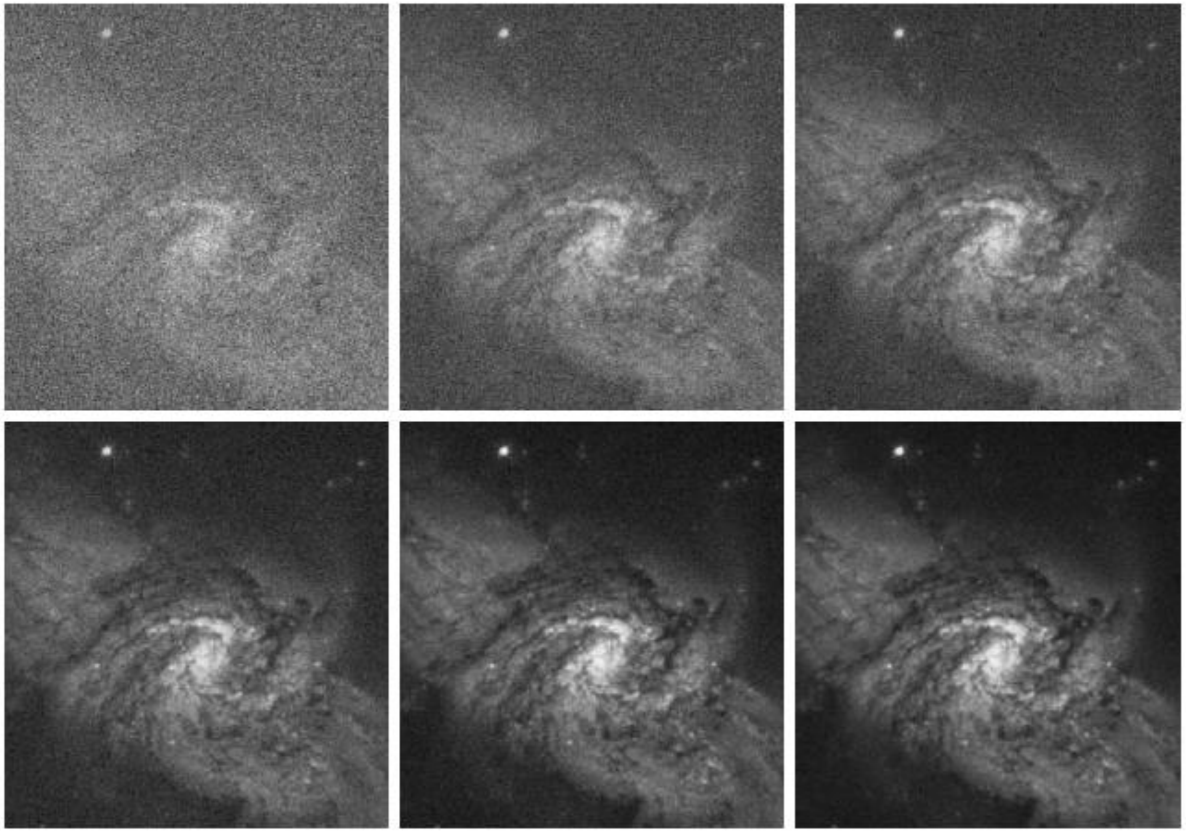
$$p(x,y) = f(x,y) \times g(x,y)$$

$$u(x,y) = f(x,y) \div g(x,y)$$

(x,y) in the image plane
$0 \leq x \leq M-1$
$0 \leq y \leq N-1$

Arithmetic Operations

- Application: Image Averaging

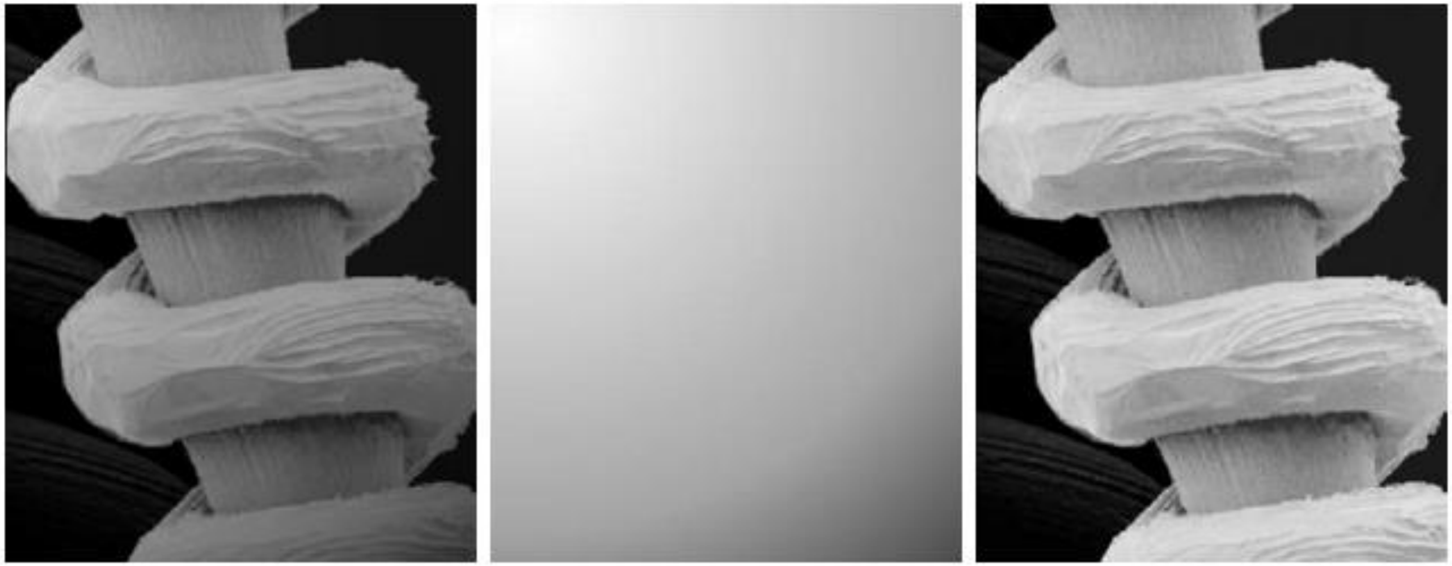


a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Arithmetic Operations

- Application: Shading Correction



a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Arithmetic Operations

- Application: ROI selection

a b c

FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

Set and Logical Operations

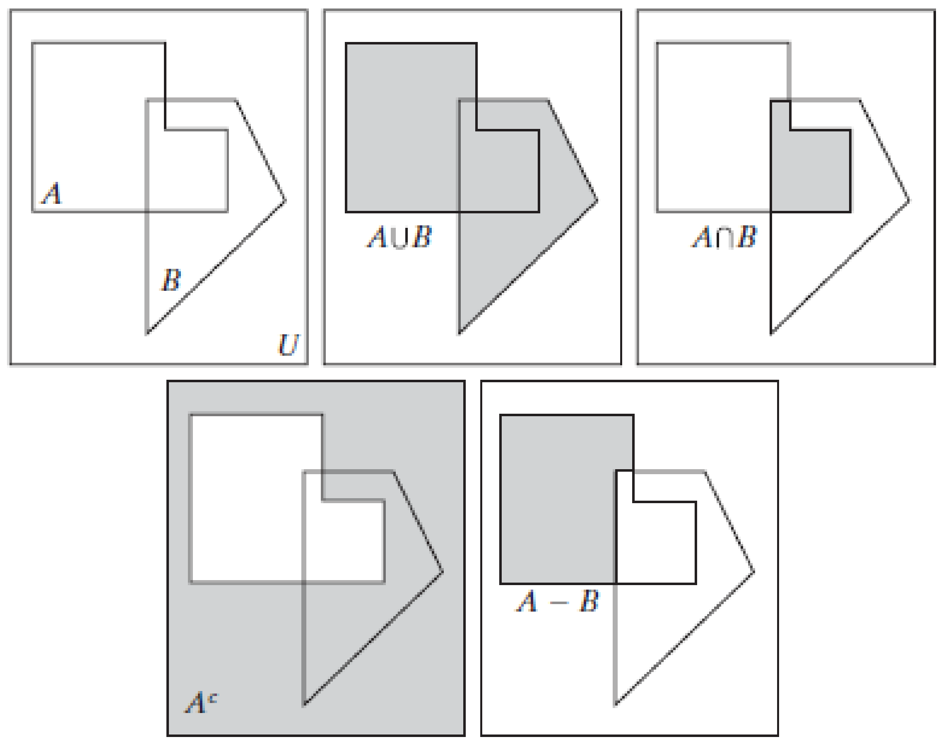
- When ordered pair $a = (a_1, a_2)$ is an element of A we write $a \in A$.
- Set-builder notation: $c = \{w | w = -d, d \in D\}$.
- We often represent pixel coordinates by ordered pairs (x, y), i.e. $R = \{(a, b) | (a, b) \in Z^2\}$
- Set A is subset of B: $A \subseteq B$.
- Union of A and B: $A \cup B$.
- Intersection of A and B: $A \cap B$.
- Disjoint sets: $A \cap B = \emptyset$.
- Universe U: set of all elements in application.
- Complement of A: $A^c = \{w | w \notin A\}$.
- Difference between A and B: $A - B = \{w | w \in A, \notin B\} = A \cap B^c$. Also $A^c = U - A$.

Set and Logical Operations

- Application: Sets of Coordinates

a b c
d e

FIGURE 2.31 (a) Two sets of coordinates, *A* and *B*, in 2-D space. (b) The union of *A* and *B*. (c) The intersection of *A* and *B*. (d) The complement of *A*. (e) The difference between *A* and *B*. In (b)–(e) the shaded areas represent the members of the set operation indicated.



Set and Logical Operations

- Application: Binary Image Regions

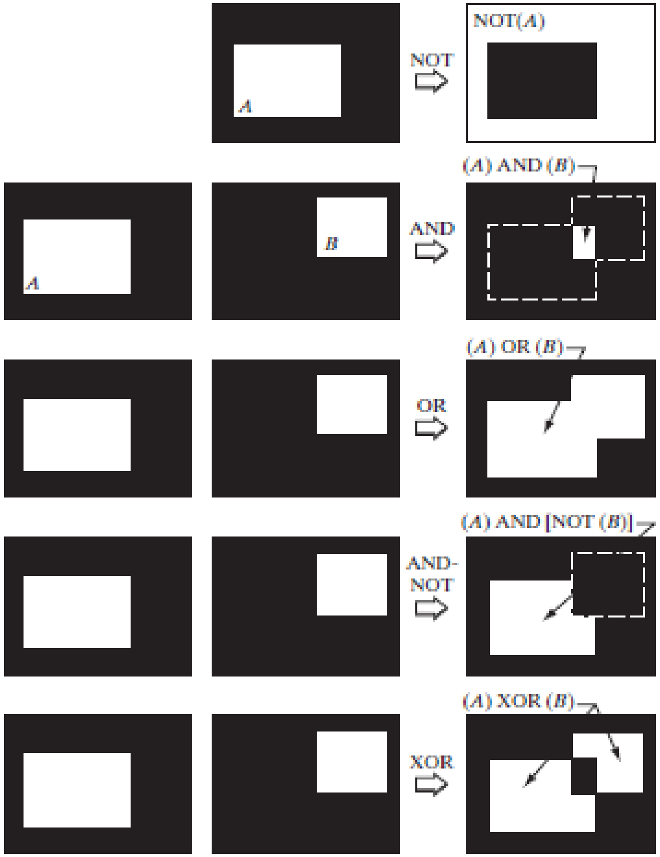


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Set and Logical Operations

- In grayscale images we can define union as a max operator and intersection as a min operator.

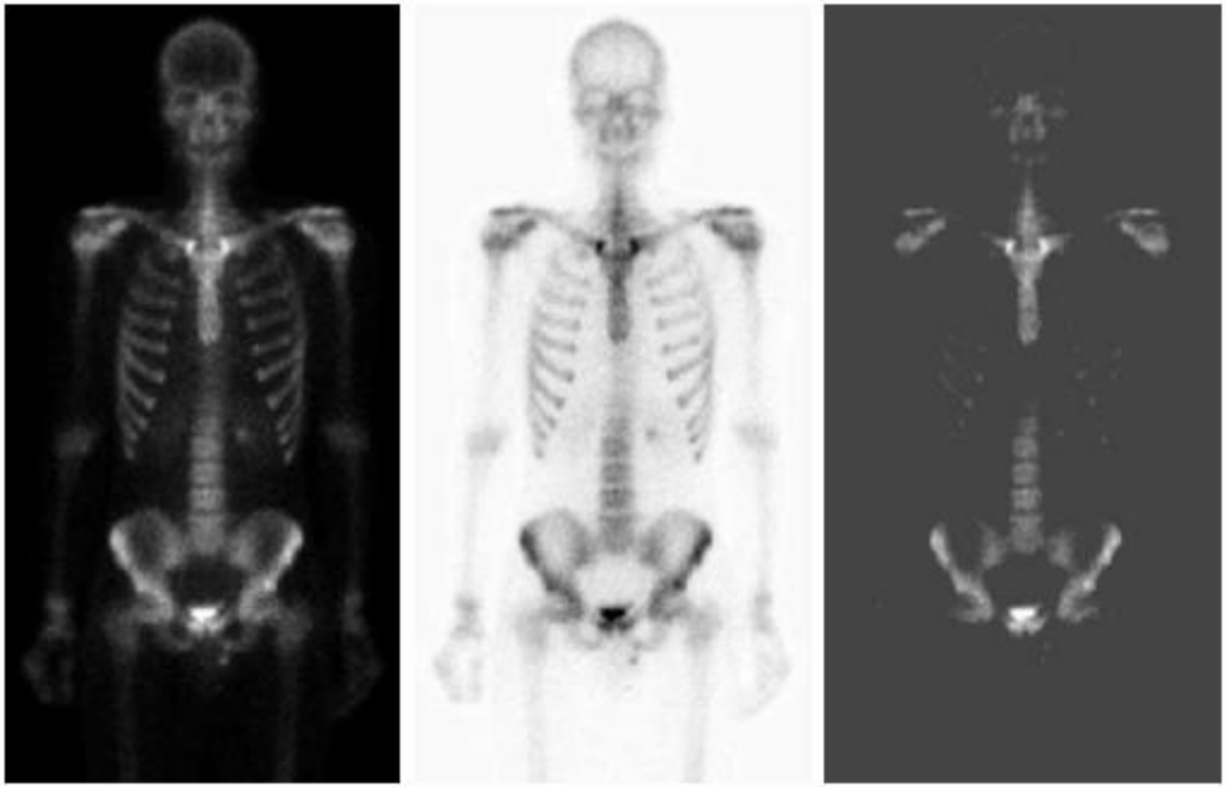
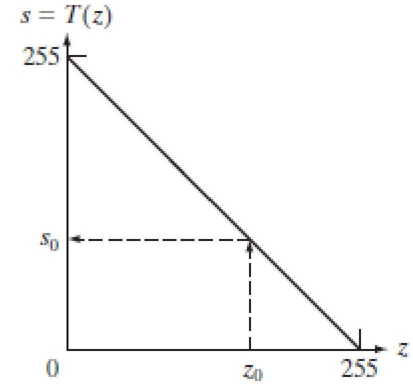


FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

Spatial Operations



- Single pixel operations alter individual pixel intensities.

  Expressed as a transformation T, $s = T(z)$,

  z: pixel intensity, s: mapped intensity.

- Neighborhood operators

  Let S_{xy} be a set of coordinates in a neighborhood centered at (x, y).

  A neighborhood operation is an operation applied to pixels in S_{xy} ad produces and output assigned to the pixel with coordinates (x, y) in the output image.

  Example: averaging filter

$$g(x, y) = \frac{1}{mn} \sum_{(r,c) \in S_{x,y}} f(r, c)$$

Spatial Operations

- Application: spatial filter - averaging

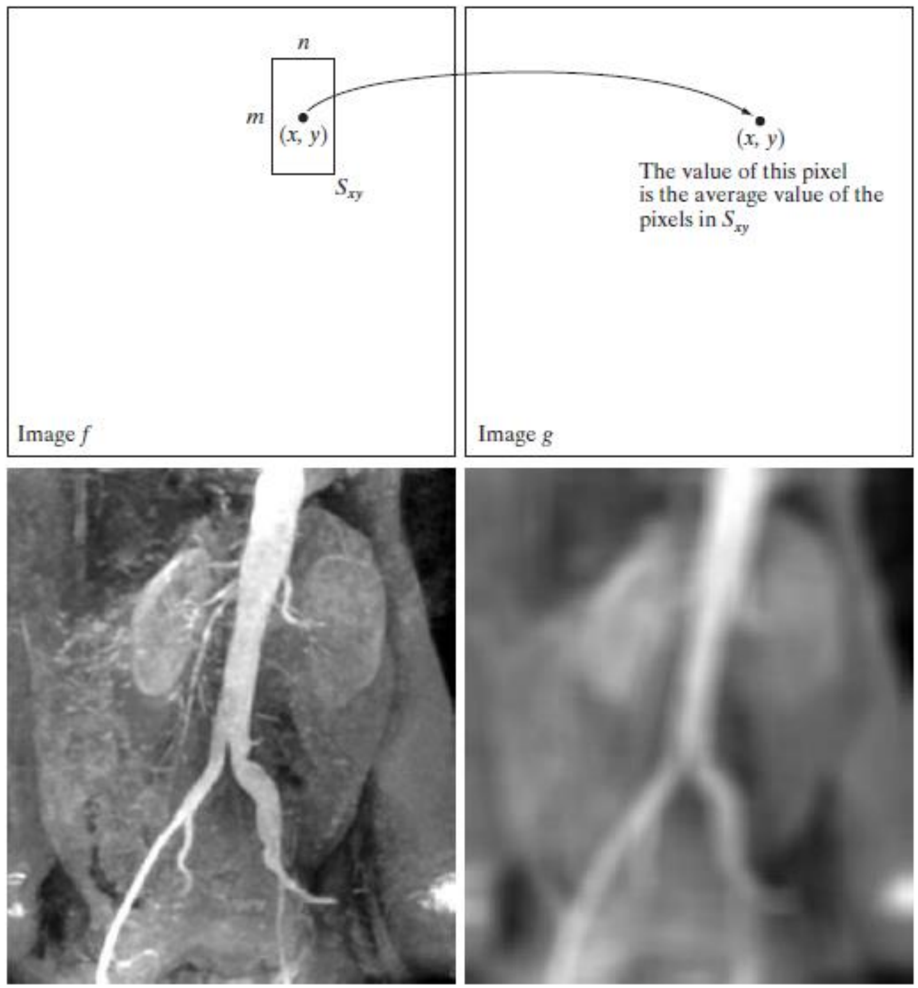


FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790×686 pixels.

Spatial Transformations

- These transformations modify the spatial relationship between pixels in the image.
- They include two operations:

1) spatial transformation of coordinates

2) intensity interpolation that assigns intensity values to transformed pixels.

Notation:

$$(x, y) = T\{(v, w)\}$$

(v, w) in original space, (x, y) in transformed space.

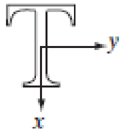
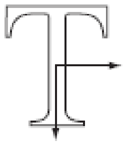
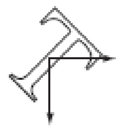
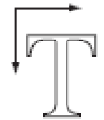
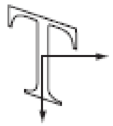
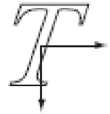
Example: $(x, y) = T\{(v, w)\} = (v/2, w/2)$, downsampling by 2.

Spatial Transformations

Affine Transform

$$[\,x \quad y \quad 1\,]$$
$$= [\,v \quad w \quad 1\,]\,T$$
$$= [\,v \quad w \quad 1\,] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

TABLE 2.2
Affine transformations based on Eq. (2.6-23).

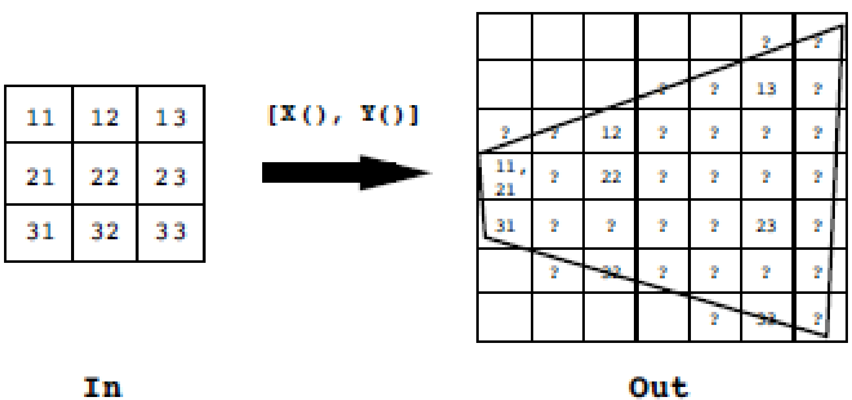
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

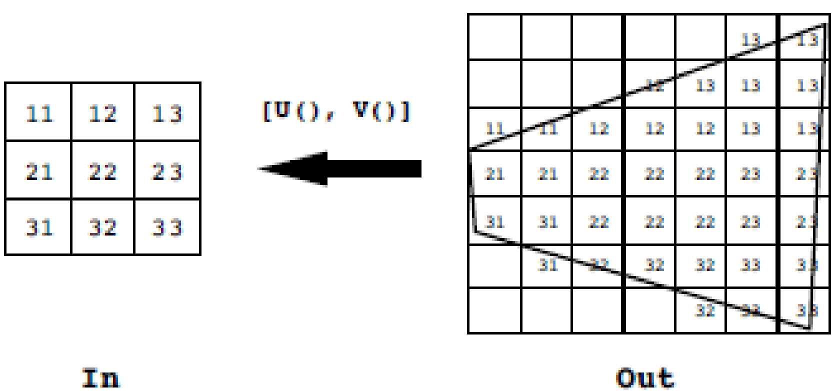
Spatial Transformations

Transformation can be implemented by **forward mapping** or **inverse mapping**.

Forward mapping leaves holes and overlaps

Inverse mapping produces complete coverage





Vectors and Matrix Operations

- Vector and matrix operations are usedin several image processing stages. Typical examples are multispectral and color imaging.
- Example: rgb pixel intensities

$$\vec{z} = \begin{bmatrix} z_R \\ z_G \\ z_B \end{bmatrix}$$

- N-dimensional Euclidean distance:

$$D(\vec{z}, \vec{a}) = [(\vec{z} - \vec{a})^T (\vec{z} - \vec{a})]^{1/2}$$

- Linear transformations:

$$\vec{w} = A(\vec{z} - \vec{a}),$$

$A[M \times N], \vec{z}[N \times 1], \vec{a}[N \times 1].$

Vectors and Matrix Operations

- We can represent an entire image by a vector with size $MN \times 1$.
- Many linear processes are modeled as

$$\vec{g} = H\vec{f} + \vec{n},$$

$\vec{g}, \vec{f}, \vec{n}$: $[MN \times 1]$ -sized vectors for processed image, input image and noise signal.

H : $[MN \times MN]$ matrix representing a linear process.

Image Transformations

- Processes are often characterized by working in the spatial or transform domain.

- A class of 2D linear transformations can be expressed as

  $T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot r(x,y,u,v)$ *Forward transform*

  $f(x,y)$: input image $r(x,y,u,v)$: transformation kernel

  (x,y): spatial variables (u,v): transform variables.

  $f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) \cdot s(x,y,u,v)$ *Inverse transform*

  $s(x,y,u,v)$: inverse transformation kernel.

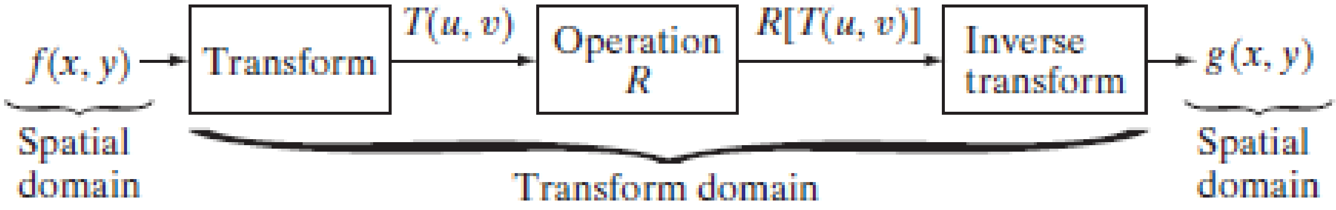


Image Transformations

Transformation kernel - separable

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

Transformation kernel - symmetric

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

Image Transformations

2D Fourier transform kernels

$$r(x, y, u, v) = e^{-j2\pi(ux/M + vy/N)},$$

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

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

Discrete Fourier Transform pair:

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

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

Image Transformations

- Fourier transform kernels are separable and symmetric.
- Separable and symmetric kernels allow 2D transform to be computed as successive 1D transformations.
- For symmetric and separable kernels and square images we can use a matrix form for transformations.

  Forward transform $T = AFA^T$

  $F : M \times M$ input image, $A : M \times M$ transform kernel, $T : M \times M$ transformed image.

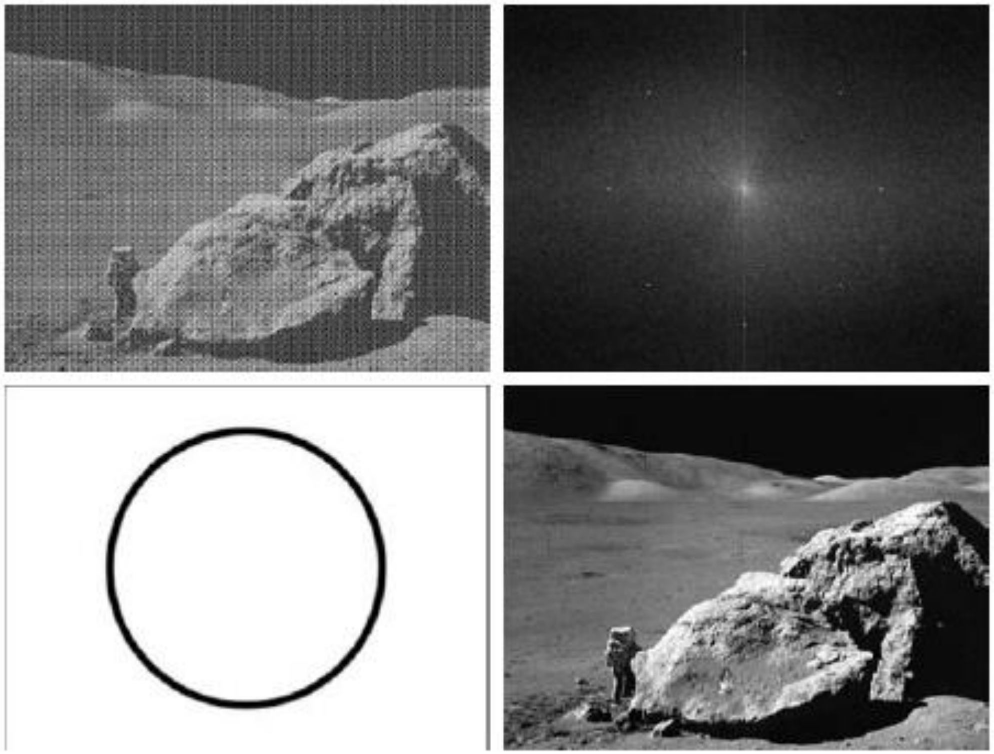
  Inverse transform $B^T TB = B^T AFAB$.

  If $B = A^{-1}$, $F = B^T TB$, else

  $\hat{F} = B^T AFAB$

Image Transformations

- Fourier domain filtering example: image restoration.



a b
c d

FIGURE 2.40 (a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

Probabilistic Methods

- We can consider image intensity z_k to be a random variable. Then

  Probability of intensity level $z_k : p(z_k) = \frac{n_k}{MN}$.

  Average intensity: $m = \sum_{k=0}^{L-1} z_k p(z_k)$.

  Intensity variance: $\sigma^2 = \sum_{k=0}^{L-1}(z_k - m)^2 p(z_k)$, measure of image contrast.

  n_{th} *moment*: $\mu_n(z) = \sum_{k=0}^{L-1}(z_k - m)^n p(z_k)$, measure of image contrast.

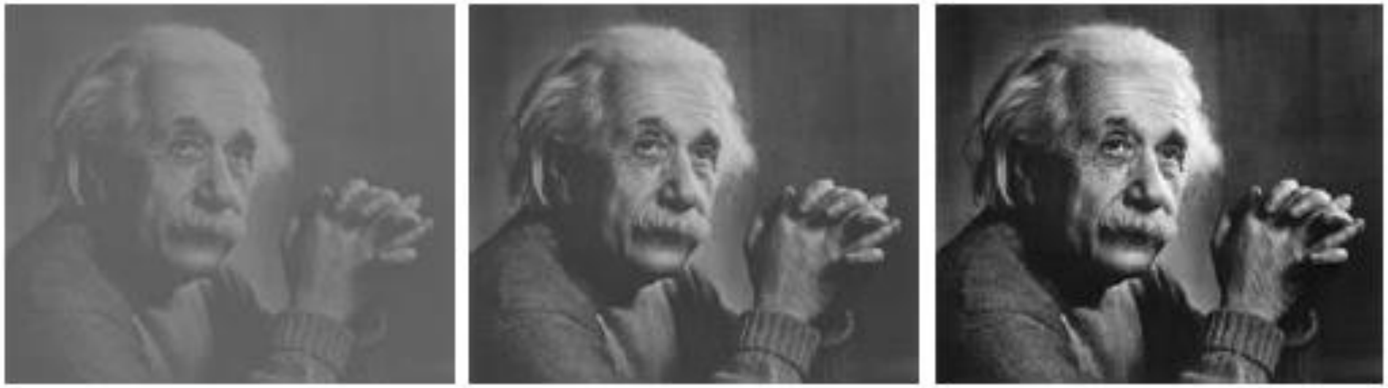
- Probabilistic approaches are used for intensity transformation, image restoration, image segmentation, texture description and object recognition to name a few examples.

- Often times, we have a number of 2D images and treat the third variable as time.

- Also, in several techniques the whole image is considered as a spatial random event. Such formulations can be handled using random fields.

Probabilistic Methods

- Example: contrast estimation by standard deviation.



a b c

FIGURE 2.41 Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Intensity Transforms and Spatial Filtering

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

February 10, 2015

Outline

Outline

Outline

Introduction

- This field of study deals with image processing in the spatial domain.
- The spatial domain processes can be expressed by $g(x, y) = T[f(x, y)]$.
- Main concepts

Introduction

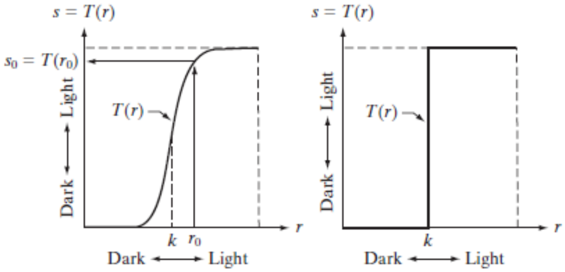
- This field of study deals with image processing in the spatial domain.
- The spatial domain processes can be expressed by $g(x, y) = T[f(x, y)]$.
- Main concepts
  Intensity Transformations are applied to a single pixel.

Introduction

- This field of study deals with image processing in the spatial domain.
- The spatial domain processes can be expressed by $g(x, y) = T[f(x, y)]$.
- Main concepts
  Intensity Transformations are applied to a single pixel.
  Spatial Filtering is applied to the neighborhood of a pixel.

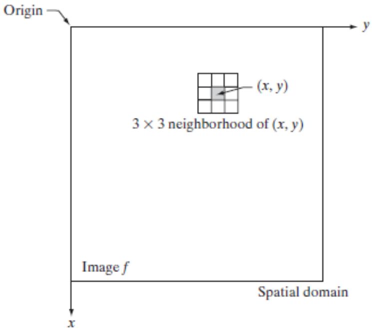
Point Processing

- It is applied to single pixels.
- Form of intensity mapping (examples: contrast stretching, thresholding)



Neighborhood processing

- Usually applied as spatial filtering (examples: averaging filter, lowpass filter).



Basic Intensity Transformation Functions

These transformations are usually of the form $s = T(r)$.

Basic Intensity Transformation Functions

These transformations are usually of the form $s = T(r)$.

- Image Negatives

Basic Intensity Transformation Functions

These transformations are usually of the form $s = T(r)$.

- Image Negatives
- Log Transformations

Basic Intensity Transformation Functions

These transformations are usually of the form $s = T(r)$.

- Image Negatives
- Log Transformations
- Piecewise-linear Transformation Functions
  - Contrast stretching
  - Intensity-level slicing
  - Bit-level slicing

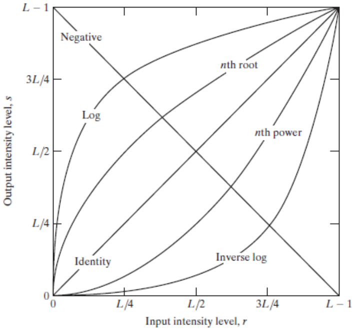
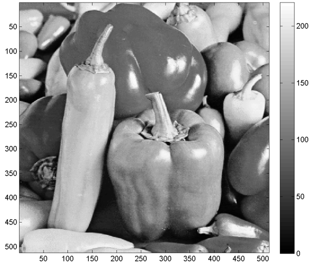
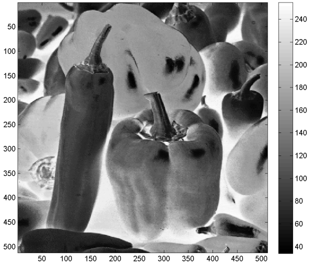


Image Negatives

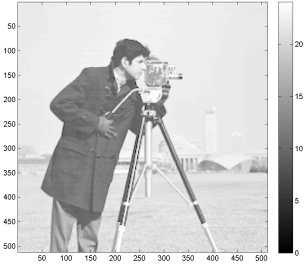
- Let an image with pixel intensities in $[0, L-1]$.
- The negative of an image calculated by $s = L - 1 - r$.





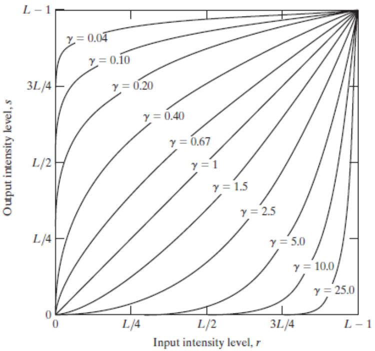
Log Transformations

- Calculated by $s = c\log(1 + r)$.

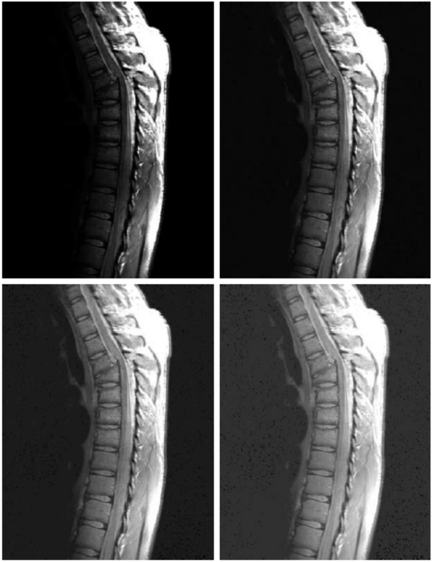


Gamma Transformations

- These transformations have the basic form $s = cr^{\gamma}$.



Gamma Transformations - Results

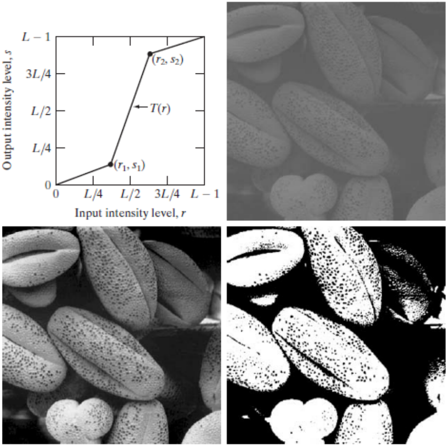


Gamma Transformations - Results

Piecewise-linear Transformation Functions

Contrast Stretching

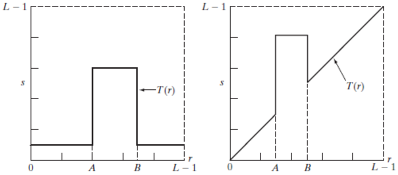
Process that expands the range of intensity levels in an image so that it spans the full range of the recording medium or display device.



Piecewise-linear Transformation Functions

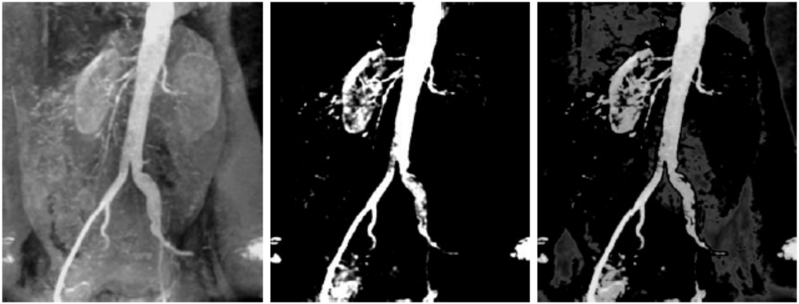
Intensity-level Slicing Approaches

1. Display in one value (white) all intensities of interest and in another (black) all other intensities.
2. Set to a fixed value the desired range of intensities but leave all other intensities unchanged.



Piecewise-linear Transformation Functions

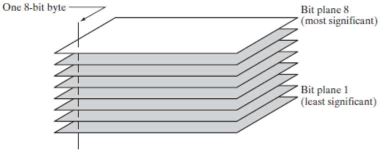
Intensity-level Slicing



Piecewise-linear Transformation Functions

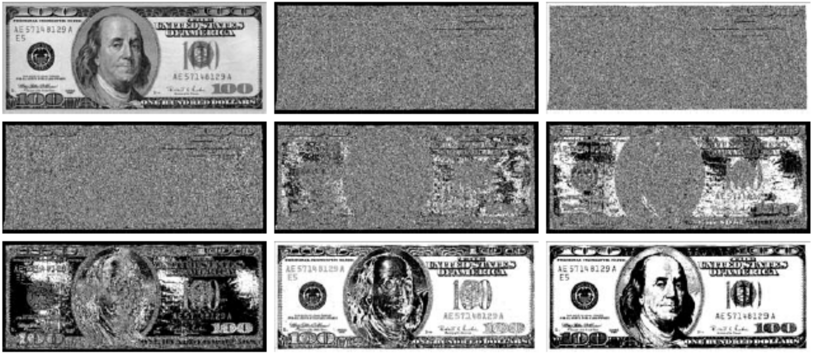
Bit-plane Slicing

- Generate black and white images that correspond to the n-th bit value.
- This process can be used to determine if the number of bits used for quantization is adequate.



Piecewise-linear Transformation Functions

Bit-plane Slicing



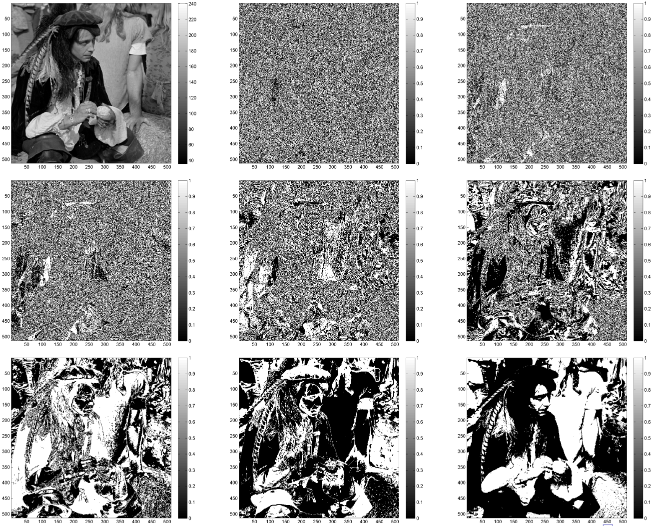
Piecewise-linear Transformation Functions

Reconstruction after bit-plane slicing

Figure: Reconstructed images using i) bit planes 8 and 7 ii) bitplanes 8, 7 and 6, and iii) bitplanes 8,7, 6, 5.

Piecewise-linear Transformation Functions

Bit-plane Slicing



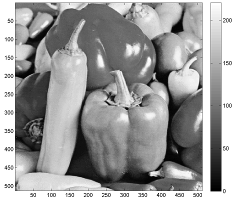
Piecewise-linear Transformation Functions

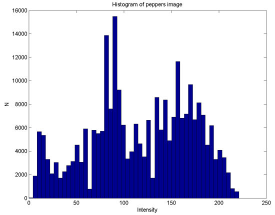
Reconstruction after bit-plane slicing

Figure: Reconstructed images using i) bit plane 8, ii) bitplanes 8 and 7, and iii) bitplanes 8,7 and 6 (left to right).

Histogram Processing

- Histogram of a digital image with levels $[0, L-1]$ is a discrete function $h(r_k) = n_k$.
  - r_k: k-th intensity value
  - n_k: number of pixels with intensity r_k.
- Normalized histogram: $p(r_k) = \dfrac{n_k}{MN}$.

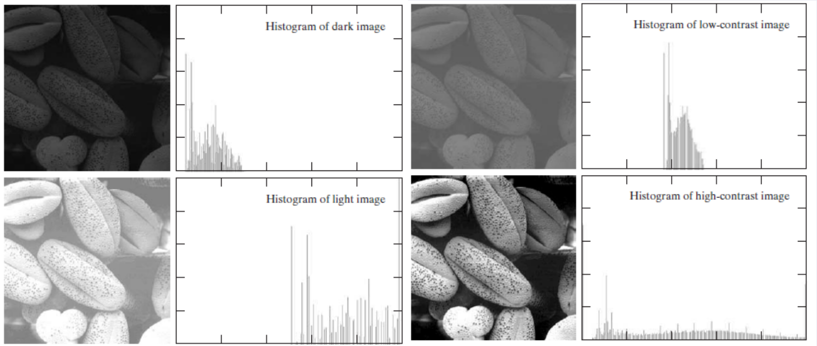




Histogram Processing

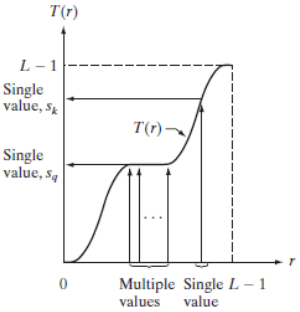
We observe that an image with intensities that occupy a large range in a uniform fashion appears to have high contrast and large variety of gray tones.

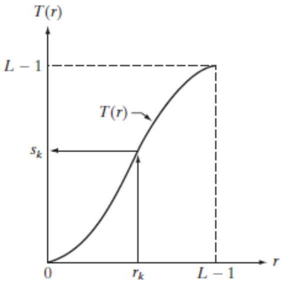
Histogram Examples



Histogram Equalization

- Let r be the image intensity and $s = T(r), 0 \leq r \leq L-1$ an intensity mapping. Our requirements are:
  1. $T(r)$ be monotonically increasing function in $[0, L-1]$.
  2. $0 \leq T(r) \leq L-1$ for $0 \leq r \leq L-1$.





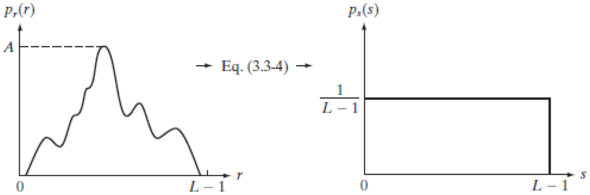
Histogram Equalization

- Intensity levels of an image can be viewed as random variables in $[0, L-1]$.
- We consider the PDF of r, $p_r(r)$ and PDF of s, $p_s(s)$.
- From probability theory it follows that if $p_r(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable over the range we are working on, then

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

Histogram Equalization

- Let's consider the mapping $s = T(R) = (L-1)\int_0^r p_r(w)dw$ (CDF of r).
- Leibniz's rule gives
  $\dfrac{ds}{dr} = \dfrac{dT(r)}{dr} = (L-1)\dfrac{d}{dr}\left[\int_0^r p_r(w)dw\right] = (L-1)p_r(r)$.
- Hence $p_s(s) = p_r(r)\left|\dfrac{1}{(L-1)p_r(r)}\right| = \dfrac{1}{L-1}$.
- We observe that $p_s(s)$ is a uniform PDF.



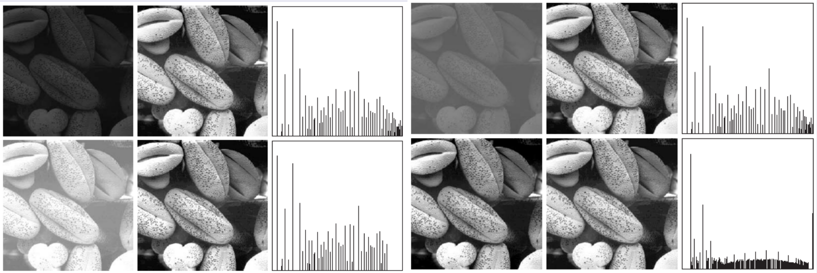
Histogram Equalization

Discrete variables

- For discrete values $p_r(r_k) = \dfrac{n_k}{M \cdot N}$, $k = 0, 1, \cdots, L-1$.
- Then $s_k = T(r_k) = (L-1)\sum_{i=0}^{k} p_r(r_i) = \dfrac{L-1}{M \cdot N}\sum_{i=0}^{k} n_i$.
- T_{r_k} is called a histogram equalization or histogram linearization transformation.
- The inverse transformation $r_k = T^{-1}(s_k)$ is used in the histogram matching technique.

Histogram Equalization

Results



Intensity Transforms and Spatial Filtering

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

February 10, 2015

Outline

Outline

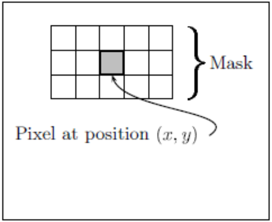
Outline

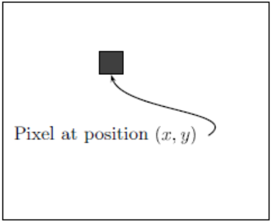
Basics of Spatial Filtering

- The name "filter" was adopted from the frequency domain filtering (such as lowpass, highpass).
- Spatial filtering may achieve similar results using masks or kernels.
- Spatial filters are more versatile than their frequency domain counterparts because they can perform nonlinear operations.

The Mechanics

- A spatial filter consists of
  1. a neighborhood and
  2. a predefined operation performed on the pixels of the neighborhood.
- Filtering creates a new pixel intensity at the center of the neighborhood as an output.
- Linear filters perform linear operations, and nonlinear filters perform nonlinear operations.



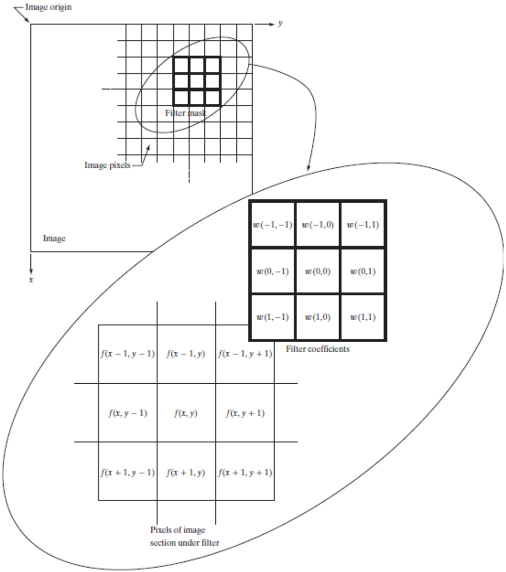


The Mechanics

Linear spatial filtering of an $M \times N$ image f with a $m \times n$ filter w is given by

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

where
x, y vary so that they span the whole image plane,
$m = 2a+1$ and $n = 2b+1$.



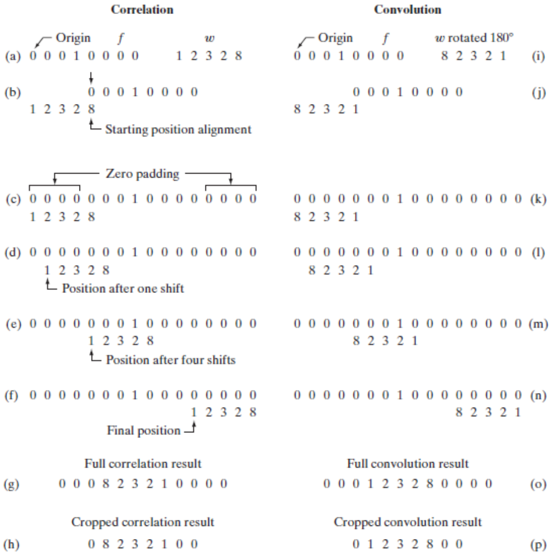
Spatial Correlation and Convolution

- Both are basic concepts in digital image processing.
- Correlation is the process of moving a filter mask over an image and computing the sum of products at each location as above.
- Convolution uses the same mechanics except that the filter is first rotated by 180°.

Spatial Correlation and Convolution

- **Correlation** of filter w with function f:
  $w(x,y) \circ f(x,y) = \sum_{-a}^{a} \sum_{-b}^{b} w(s,t) f(x+s, y+t)$.
- **Convolution** of filter w with function f:
  $w(x,y) * f(x,y) = \sum_{-a}^{a} \sum_{-b}^{b} w(s,t) f(x-s, y-t)$.

Spatial Correlation and Convolution in 1-D



Spatial Correlation and Convolution in 1-D

Signal

1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0

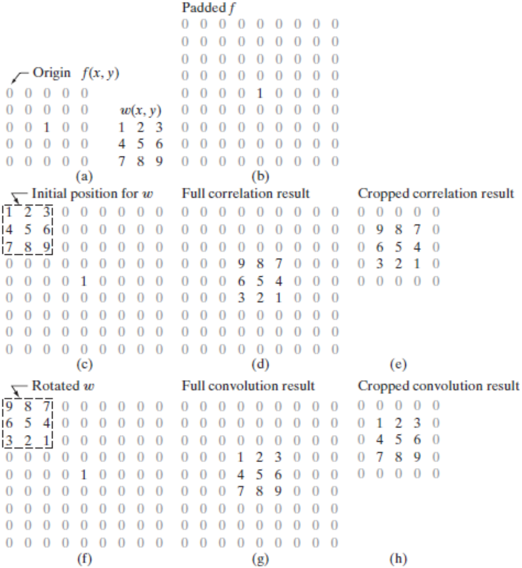
Filter kernel

0.3333	0.3333	0.3333

Output

0.7333	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	1.9333

Spatial Correlation and Convolution in 2-D



Spatial Correlation and Convolution in 2-D

Signal

170	240	10	80	150
230	50	70	140	160
40	60	130	200	220
100	120	190	210	30
110	180	250	20	90

Filter kernel

1	1	1
1	1	1
1	1	1

Correlation Result

690	770	590	610	530
790	1000	980	1160	950
600	990	1170	1350	960
610	1180	1360	1340	770
510	950	970	790	350

Vector Representation of Linear Filtering

- Another useful representation of the filtering process is

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{k=1}^{mn} w_k z_k = \vec{w}^T \vec{z}.$$

$\vec{w}$: vector with filter coefficients
$\vec{z}$: corresponding image intensities under the filter mask.

Generating Spatial Filter Masks

- We select the filter coefficients to perform specific filtering operations, using a sum of products.
- For example, to implement a 3×3 averaging filter, we perform $R = \frac{1}{9}\sum_{i=1}^{9} z_i$. This is equivalent to filtering with a kernel with $w_i = \frac{1}{9}$.

Generating Spatial Filter Masks

- In other applications the filter approximates a 2-D function for example a Gaussian $h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ with σ: standard deviation and $x, y \in \mathbb{Z}$.
- To generate the filter we sample the function in the neighborhood $h(-1,-1), h(-1,0), \ldots, h(1,0), h(1,1)$.

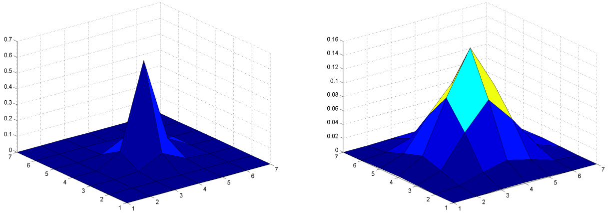


Figure: 7x7 Gaussian kernels with $\sigma = 0.5$ and $\sigma = 1.0$ (left to right).

Generating Spatial Filter Masks

- For nonlinear filters, we need to specify the kernel size and the operation. For example, maximum, minimum, or median value applied to the pixel intensities.
- Nonlinear filters can be powerful for tasks such as image enhancement and noise reduction.

Smoothing Spatial Filters

- These filters are used for blurring and noise reduction.
- Blurring may precede object extraction.
- Noise reduction can be achieved by blurring or nonlinear filtering.

Smoothing Linear Filters

- The output of such filters is the average of pixel intensities in the neighborhood of the mask. They are also called averaging or lowpass filters.
- Idea is to reduce the "sharp" transitions of intensity caused by noise.
- But blurring can smooth-out the image edges as a side-effect.

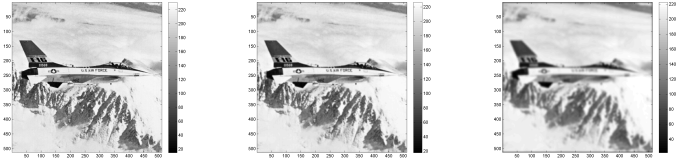


Figure: Original image, smoothed by Gaussian with $\sigma = 1$ and smoothed by Gaussian with $\sigma = 3$.

Smoothing Linear Filters

Scenarios

1. Averaging: $R = \frac{1}{9}\sum_{i=1}^{9} z_i$ also called a box filter.
2. Weighted average: $g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$.

   Example: Coefficient weights in $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ reduce the edge blurring effect.

Smoothing Linear Filters

Smoothing before segmentation

Frequently a blurring operation is followed by thresholding to identify the main objects of an image.

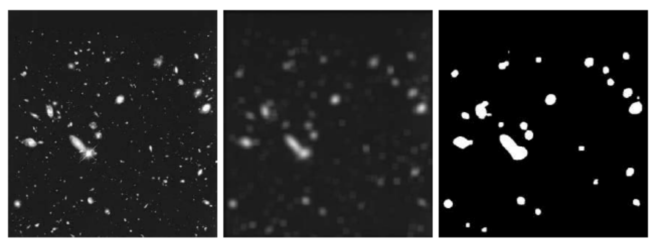


Figure: Original image, smoothed image, and segmentation after thresholding (left to right).

Order Statistics (non-linear) filters

- These filters rank the pixel intensities in the neighborhood of a mask and select a percentile p.
  - p=50%: median filter
  - p=0%: minimum filter
  - p=100%: maximum filter
- The Median filter can effectively reduce impulse noise.

Order Statistics (non-linear) filters

- The Median filter can effectively reduce impulse noise.

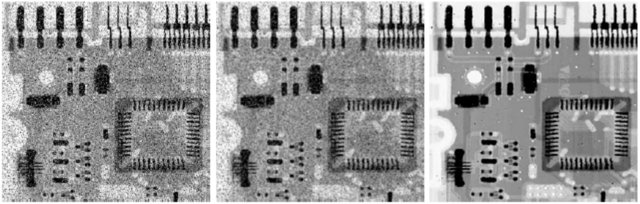


Figure: Example of an image corrupted by salt and pepper noise, result from smoothing filter, and result from median filter (left to right).

Order Statistics (non-linear) filters

- The Median filter can effectively reduce impulse noise.

Figure: Example of an image corrupted by salt and pepper noise, result from smoothing filter, and result from median filter (left to right).

Sharpening Spatial Filters

- These filters aim to highlight transitions in intensity through differentiation.
- Typical sharpening filters are based on first- and second-order derivatives.
- Derivatives of digital functions are defined in terms of differences

Sharpening Spatial Filters

Requirements for 1_{st} derivative

1. Zero in areas of constant intensity.
2. Non-zero at the onset of intensity step or ramp.
3. Non-zero along ramp.

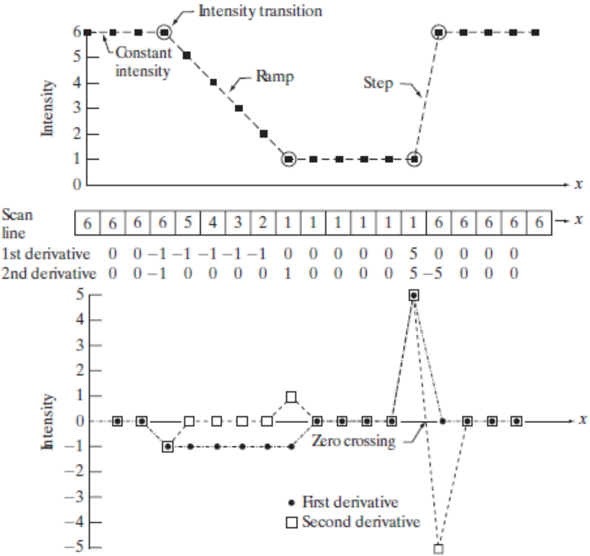
$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

Requirements for 2_{nd} derivative

1. Zero in constant areas.
2. Nonzero at onset and end of intensity step or ramp.
3. Zero along ramps of constant slope.

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

Requirements for 1_{st} and 2_{nd} derivatives



Ramp intensity profile example

Real case: CT scan cross-section

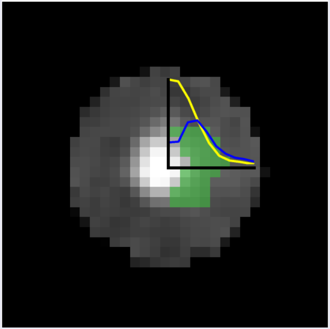


Figure: Intensity profile along the horizontal axis (in yellow) and gradient magnitude (in blue).

Using the 2_{nd} Derivative for Image Sharpening

- Here we deal with isotropic filters that are rotation invariant.
- Simplest isotropic derivative operator is the Laplacian

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

Using the 2_{nd} Derivative for Image Sharpening

- Here we deal with isotropic filters that are rotation invariant.
- Simplest isotropic derivative operator is the Laplacian

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

Using the 2_{nd} Derivative for Image Sharpening

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

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

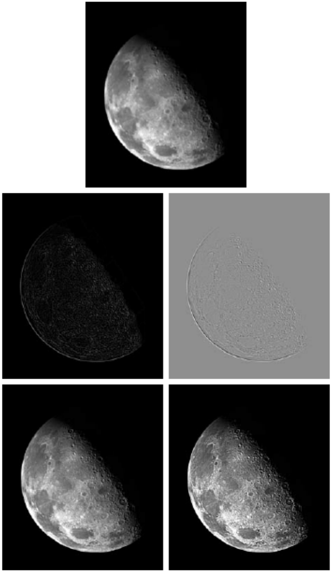
$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y).$$

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

Using the 2_{nd} Derivative for Image Sharpening

- $\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y).$
- Corresponding filter kernel: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
- Laplacian can be applied as $g(x,y) = f(x,y) + c[\nabla^2 f(x,y)].$

Using the 2_{nd} Derivative for Image Sharpening



Using the 2_{nd} Derivative for Image Sharpening

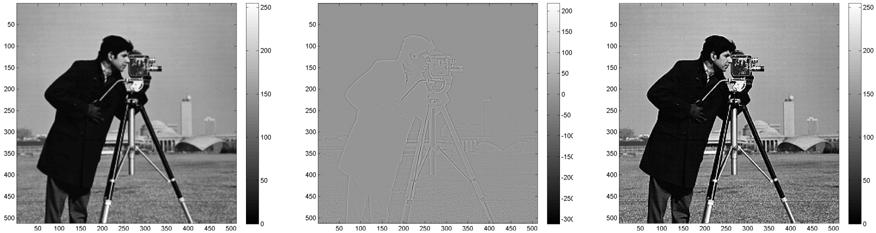
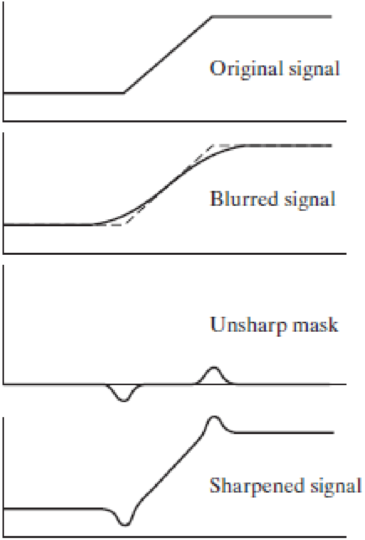


Figure: Example of an original image, result of correlation with Laplacian filter, image sharpening by subtracting the Laplacian image from the original (left to right).

Unsharp Masking and Highboost Filtering

Steps for Unsharp Masking

1. Blur the original image $f(x,y)$ to produce $\bar{f}(x,y)$.
2. Subtract blurred image from original (result is mask): $g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$.
3. Add mask to original image: $g(x,y) = f(x,y) + k \cdot g_{mask}(x,y)$.



Unsharp Masking and Highboost Filtering

- For $k = 1$, the above process is called *unsharp masking*.
- For $k > 1$, the above process is called *highboost filtering*.

Unsharp Masking

Unsharp Masking and Highboost Filtering

Figure: Example of an original image, Gaussian smoothed, it's g-mask (top row, left to right), the result of unsharp masking, and the result of highboost filtering with $k = 1.5$ (bottom row, left to right).

Filtering in Frequency Domain

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

March 24, 2015

Outline

Outline

1. Basics of Filtering in the Frequency Domain
   - Frequency Domain Interpretation
   - How to Apply Filtering in Frequency Domain
   - Frequency Domain Filtering Steps

2. Image Smoothing in the Frequency Domain
   - Ideal Lowpass Filters (ILPFs)
   - Gaussian Lowpass Filters (GLPFs)
   - Butterworth Lowpass Filters (BLPFs)

Use of Frequency Domain

- The frequency domain does not display directly the visual content, but it provides an alternate representation.

Use of Frequency Domain

- The frequency domain does not display directly the visual content, but it provides an alternate representation.
- We usually visualize the frequency magnitude (spectrum) and phase angle of DFT.

Use of Frequency Domain

- The frequency domain does not display directly the visual content, but it provides an alternate representation.
- We usually visualize the frequency magnitude (spectrum) and phase angle of DFT.
- Visualization of the spectrum can indicate characteristics of the image.

Use of Frequency Domain

- The frequency domain does not display directly the visual content, but it provides an alternate representation.
- We usually visualize the frequency magnitude (spectrum) and phase angle of DFT.
- Visualization of the spectrum can indicate characteristics of the image.
- The frequency, defined as rate of change gives a global information on the image content.

Use of Frequency Domain

- Examples
  - An image with large background areas of uniform intensities will have few high frequency components.
  - An image with sharp changes and many boundaries will produce a spectrum with high frequency coefficients.
  - Edge orientation is visible in spectrum.

Magnitude and Phase Information

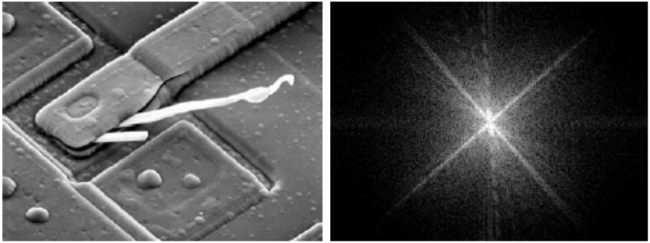


Figure: SEM image example and its frequency spectrum. Observe the lines in spectrum that correspond to orientation information and the high frequency components.

Magnitude and Phase Information

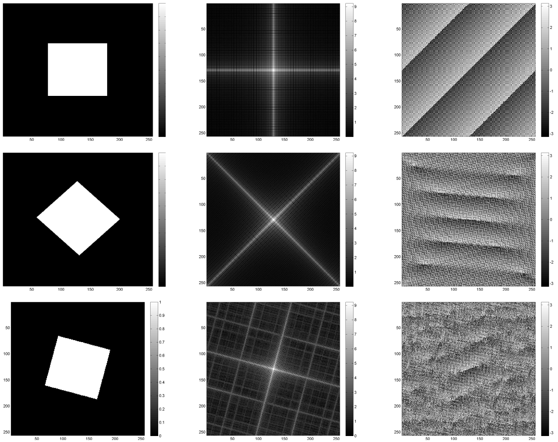


Figure: Examples of an original image, its DFT magnitude, and its DFT phase (left to right).

Introduction to Frequency Domain Filtering

- According to convolution theorem, to filter an image in frequency domain we i) compute its DFT, ii) multiply with a filter function and iii) apply the inverse DFT to the previous result.

Introduction to Frequency Domain Filtering

- According to convolution theorem, to filter an image in frequency domain we i) compute its DFT, ii) multiply with a filter function and iii) apply the inverse DFT to the previous result.
- This is also expressed as:

$$g(x,y) = \mathscr{F}^{-1}[F(u,v) \cdot H(u,v)],$$

where $f(x,y)$ is an $M \times N$ image, $F(u,v) = \mathscr{F}[f(x,y)]$ is the DFT of $f(x,y)$, $H(u,v)$ is the filter transfer function, and $g(x,y)$ is the filtered image in the spatial domain.

Introduction to Frequency Domain Filtering

- According to convolution theorem, to filter an image in frequency domain we i) compute its DFT, ii) multiply with a filter function and iii) apply the inverse DFT to the previous result.
- This is also expressed as:

$$g(x,y) = \mathscr{F}^{-1}[F(u,v) \cdot H(u,v)],$$

where $f(x,y)$ is an $M \times N$ image, $F(u,v) = \mathscr{F}[f(x,y)]$ is the DFT of $f(x,y)$, $H(u,v)$ is the filter transfer function, and $g(x,y)$ is the filtered image in the spatial domain.

- Next, we discuss how to apply filtering in the frequency domain and minimize inaccuracies in calculations.

The Filter Transfer Function

- For the filter function $H(u, v)$ we choose functions symmetric at the DC frequency.

The Filter Transfer Function

- For the filter function $H(u,v)$ we choose functions symmetric at the DC frequency.
- So, we need to center $F(u,v)$ as well.

The Filter Transfer Function

- For the filter function $H(u, v)$ we choose functions symmetric at the DC frequency.
- So, we need to center $F(u, v)$ as well.
- To do this, we use the translation property and multiply $f(x, y)$ by $(-1)^{x+y}$. Then $F(u, v)$ is centered at $(M/2, N/2)$.

Filter Design

- After centering, we design the filter.
- Usually the process involves the selection of frequencies that we want to let pass, and another range that we want to block.

Filter Design

- After centering, we design the filter.
- Usually the process involves the selection of frequencies that we want to let pass, and another range that we want to block.
- Low frequencies correspond to the main regions of approximately uniform intensities.

Filter Design

- After centering, we design the filter.
- Usually the process involves the selection of frequencies that we want to let pass, and another range that we want to block.
- Low frequencies correspond to the main regions of approximately uniform intensities.
- High frequencies correspond to sharp changes of intensity and details. Image edges and noise are such examples.

Filter Types

- There are three basic types of filters: lowpass, highpass, and bandpass.
- Lowpass filtering is used to reduce the noise but also reduces the sharpness of details.
- Highpass filtering is used to enhance edges, corners, and other details, but will also enhance the noise.
- Bandpass/bandreject filtering is used to remove systemic noise patterns.

Examples Filter Types

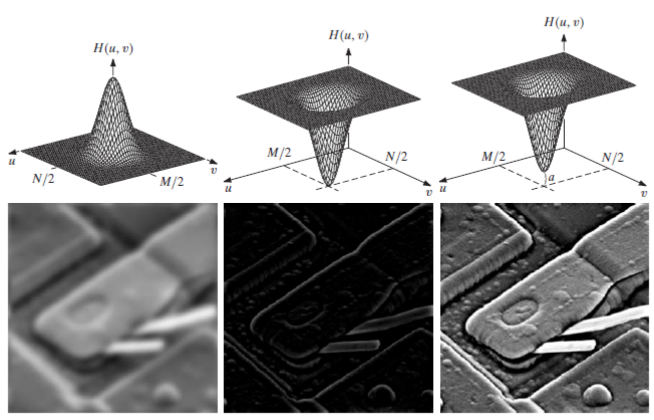


Figure: Example of a lowpass, a highpass, with 0 DC coefficient, and a highpass with non-zero DC coefficient filter transfer functions.

Image Padding

- Because of periodicity of DFT, if the image and filter are not padded, then the convolution suffers from the wraparound error.

Filter Padding

- In addition to the image, we need to pad the filter function too.

Filter Padding

- In addition to the image, we need to pad the filter function too.
- Remember that padding is applied in the spatial domain, but filter design is done in the frequency domain.

Filter Padding

- In addition to the image, we need to pad the filter function too.
- Remember that padding is applied in the spatial domain, but filter design is done in the frequency domain.
- How do we handle this?

Filter Padding Strategy and Pitfall

- First thought: create $M \times N$ filter in Fourier domain, apply IDFT to filter, pad filter, then apply DFT.

Filter Padding Strategy and Pitfall

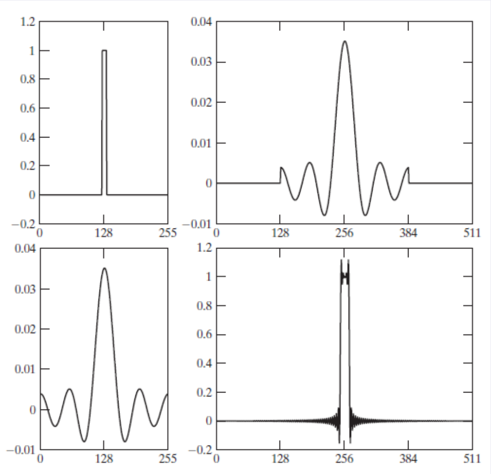
- First thought: create $M \times N$ filter in Fourier domain, apply IDFT to filter, pad filter, then apply DFT.
- The above process may introduce discontinuities at the padding points, therefore infinite number of harmonics are introduced (remember the box filter DFT).

Filter Padding Strategy and Pitfall

- First thought: create $M \times N$ filter in Fourier domain, apply IDFT to filter, pad filter, then apply DFT.
- The above process may introduce discontinuities at the padding points, therefore infinite number of harmonics are introduced (remember the box filter DFT).
- This causes ringing effect in the frequency domain.

Filter Padding Pitfall

- To summarize, padding of ideal filters in the spatial domain produces ringing effect in the frequency domain.

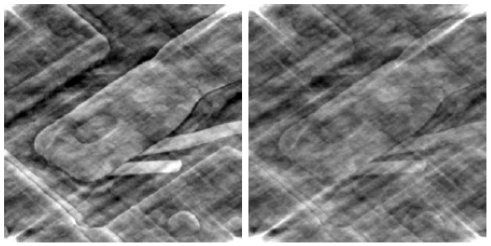


Filter Padding Solution

- We saw that padding of ideal filters in the spatial domain produces ringing effect in the frequency domain.
- To reduce ringing, we can create the filter in frequency domain with equal matrix size to the padded image. Then apply filtering in the frequency domain.

Zero-Phase Shift

- We usually modify the spectrum of an image for filtering, but changes in phase angle should be avoided.



Zero-Phase Shift

- To avoid phase shift, we choose the filter transfer function to be real.

Zero-Phase Shift

- To avoid phase shift, we choose the filter transfer function to be real.
- Let the DFT of an image $F(u, v) = R(u, v) + jI(u, v)$, and a filter $H(u, v)$.

Zero-Phase Shift

- To avoid phase shift, we choose the filter transfer function to be real.
- Let the DFT of an image $F(u,v) = R(u,v) + jI(u,v)$, and a filter $H(u,v)$.
- Filtering then is computed by $F(u,v)H(u,v) = H(u,v)R(u,v) + jH(u,v)I(u,v)$.

Zero-Phase Shift

- To avoid phase shift, we choose the filter transfer function to be real.
- Let the DFT of an image $F(u,v) = R(u,v) + jI(u,v)$, and a filter $H(u,v)$.
- Filtering then is computed by $F(u,v)H(u,v) = H(u,v)R(u,v) + jH(u,v)I(u,v)$.
- It is easy to show that the phase of $F(u,v)H(u,v)$ is equal to the phase of $F(u,v)$.

Zero-Phase Shift

- To avoid phase shift, we choose the filter transfer function to be real.
- Let the DFT of an image $F(u,v) = R(u,v) + jI(u,v)$, and a filter $H(u,v)$.
- Filtering then is computed by $F(u,v)H(u,v) = H(u,v)R(u,v) + jH(u,v)I(u,v)$.
- It is easy to show that the phase of $F(u,v)H(u,v)$ is equal to the phase of $F(u,v)$.
- These filters are called zero-phase-shift filters.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x, y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x, y)$.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x, y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x, y)$.
  2. Multiply $f(x, y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x,y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x,y)$.
  2. Multiply $f(x,y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.
  3. Compute DFT: $F(u,v) = \mathscr{F}[f(x,y)]$.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x, y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x, y)$.
  2. Multiply $f(x, y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.
  3. Compute DFT: $F(u, v) = \mathscr{F}[f(x, y)]$.
  4. Build a real and symmetric $2M \times 2N$ filter transfer function $H(u, v)$ in the frequency domain. Center function at (M, N).

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x,y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x,y)$.
  2. Multiply $f(x,y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.
  3. Compute DFT: $F(u,v) = \mathscr{F}[f(x,y)]$.
  4. Build a real and symmetric $2M \times 2N$ filter transfer function $H(u,v)$ in the frequency domain. Center function at (M,N).
  5. Apply array multiplication: $F(u,v) \cdot H(u,v)$.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x,y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x,y)$.
  2. Multiply $f(x,y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.
  3. Compute DFT: $F(u,v) = \mathscr{F}[f(x,y)]$.
  4. Build a real and symmetric $2M \times 2N$ filter transfer function $H(u,v)$ in the frequency domain. Center function at (M,N).
  5. Apply array multiplication: $F(u,v) \cdot H(u,v)$.
  6. Transform back to spatial domain: $g_p(x,y) = Re\{\mathscr{F}^{-1}[(F(u,v) \cdot H(u,v)]\}(-1)^{x+y}$.

Filtering Process in the Frequency Domain

- Using the previous results, we apply filtering in frequency domain as follows
  1. Given image $f(x, y)$ with size $M \times N$, we append zeros to image matrix to increase size to $2M \times 2N$. Let padded image be $f_p(x, y)$.
  2. Multiply $f(x, y)$ by $(-1)^{x+y}$ to shift DC frequency to the center.
  3. Compute DFT: $F(u, v) = \mathscr{F}[f(x, y)]$.
  4. Build a real and symmetric $2M \times 2N$ filter transfer function $H(u, v)$ in the frequency domain. Center function at (M, N).
  5. Apply array multiplication: $F(u, v) \cdot H(u, v)$.
  6. Transform back to spatial domain: $g_p(x, y) = Re\{\mathscr{F}^{-1}[(F(u, v) \cdot H(u, v)]\}(-1)^{x+y}$.
  7. Extract the top left quadrant of $g_p(x, y)$ to remove padding. The result is the filtered image $g(x, y)$.

Example : Filtering Process in the Frequency Domain

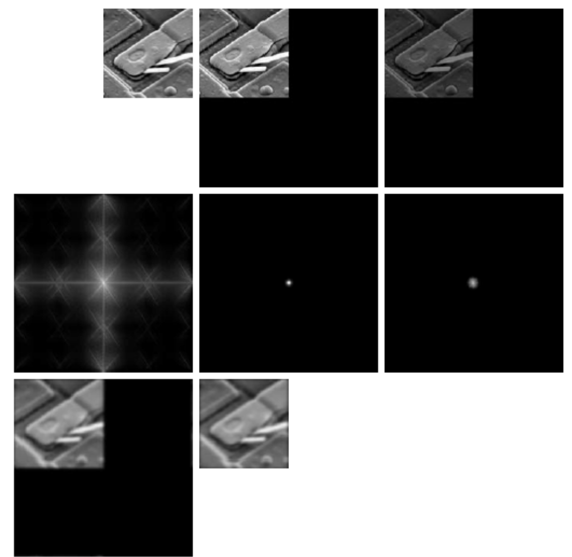


Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.
- This process is known as lowpass filtering.

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.
- This process is known as lowpass filtering.
- Widely used lowpass filter types

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.
- This process is known as lowpass filtering.
- Widely used lowpass filter types
    - Ideal Lowpass Filters

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.
- This process is known as lowpass filtering.
- Widely used lowpass filter types
  - Ideal Lowpass Filters
  - Butterworth Lowpass Filters

Image Smoothing in the Frequency Domain

- Sharp intensity changes in the image (for example noise, edges) corresond to high frequencies of the spectrum.
- Smoothing of such details corresponds to attenuation of the high frequency components.
- This process is known as lowpass filtering.
- Widely used lowpass filter types
  - Ideal Lowpass Filters
  - Butterworth Lowpass Filters
  - Gaussian Lowpass Filters

Ideal Lowpass Filters

- These filters fully pass all frequencies within a radius D_0, and fully attenuate all frequencies outside of this circle, denoted by ILPF.

Ideal Lowpass Filters

- These filters fully pass all frequencies within a radius D_0, and fully attenuate all frequencies outside of this circle, denoted by ILPF.
- D_0 is called cut-off frequency.

Ideal Lowpass Filters

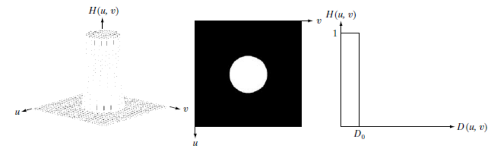
- These filters fully pass all frequencies within a radius D_0, and fully attenuate all frequencies outside of this circle, denoted by ILPF.
- D_0 is called cut-off frequency.
- The filter transfer function is

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

$D(u,v) = [(u-P/2)^2 + (v-Q/2)^2]^{1/2}$, D_0: cut-off frequency, (P,Q): padded image matrix size.

Ideal Lowpass Filters

- The term ideal is used to show full passing and full attenuation.
- It is a real and symmetric filter.



Ideal Lowpass Filter Design

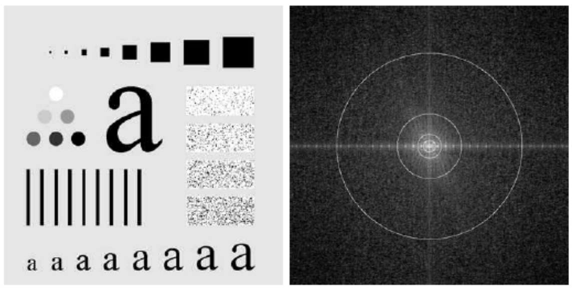
- To set D_0 we can use power spectrum measurements $P(u,v)$, computed from the squared magnitude of spectrum $P(u,v) = |\mathscr{F}[f(x,y)]|^2$.

Ideal Lowpass Filter Design

- To set D_0 we can use power spectrum measurements $P(u,v)$, computed from the squared magnitude of spectrum $P(u,v) = |\mathscr{F}[f(x,y)]|^2$.
- The total image power is $P_T = \sum_{u=0}^{P-1}\sum_{v=0}^{Q-1} P(u,v)$.

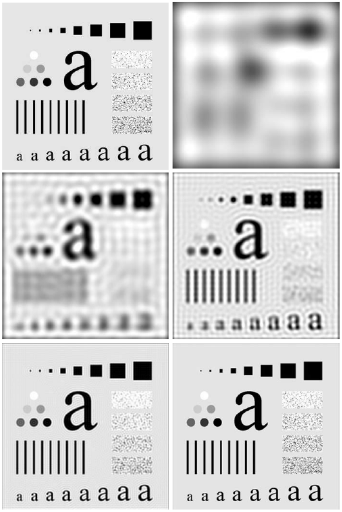
Ideal Lowpass Filter Design

- To set D_0 we can use power spectrum measurements $P(u,v)$, computed from the squared magnitude of spectrum $P(u,v) = |\mathscr{F}[f(x,y)]|^2$.
- The total image power is $P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u,v)$.
- Then, a circle with radius D_0 encloses a percent of power α equal to $\alpha = 100 \sum_u \sum_v P(u,v)/P_T$, with $D(u,v) \leq D_0$.



Ideal Lowpass Filtering Example

- As ILPF radius increases, smoothing reduces, and ringing effect reduces.
- The spatial kernel approaches an impulse.

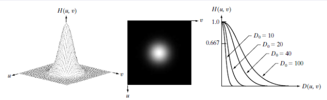


Gaussian Filters

- The Gaussian lowpass filter transfer function is defined as

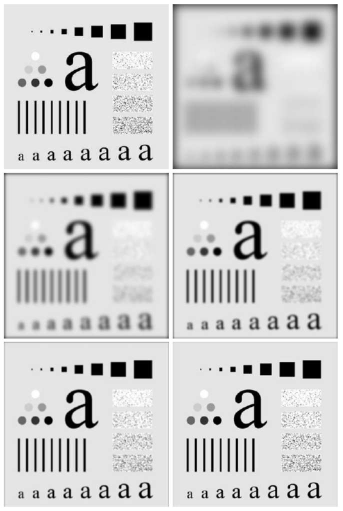
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}.$$

- We can show that the IDFT of a Gaussian is also a Gaussian, implying that a Gaussian filter does not cause ringing.



Gaussian Filtering Example

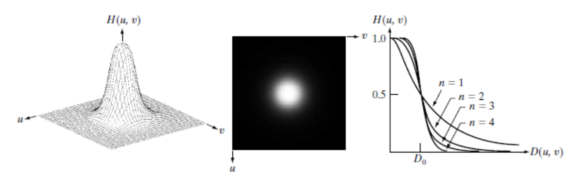
- Gaussian filtering produces no ringing effect.



Butterworth Lowpass Filters

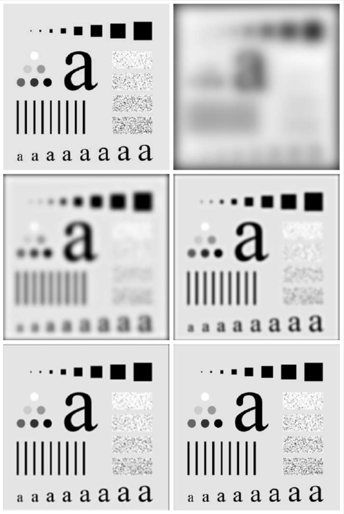
- These filters can be seen as intermediate cases between the Ideal and Gaussian lowpass filters.
- The transfer function of a BLPF with order n is defined as

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



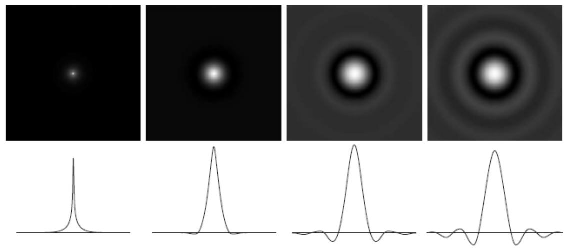
Butterworth Lowpass Filtering Example

- As BLPF radius increases, smoothing reduces.
- The spatial kernel approaches an impulse.



Butterworth Lowpass Filtering Characteristics

- In BLPF, ringing effect becomes more intense with increasing order.



Filtering in Frequency Domain

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

March 27, 2015

Outline

1. Image Sharpening in the Frequency Domain
   - Butterworth Highpass Filters (BHPF)

Outline

1. Image Sharpening in the Frequency Domain
   - Butterworth Highpass Filters (BHPF)
2. Frequency-Selective Filtering
   - Bandpass and Bandreject Filters
   - Notch Filters

Outline

1. Image Sharpening in the Frequency Domain
   - Butterworth Highpass Filters (BHPF)
2. Frequency-Selective Filtering
   - Bandpass and Bandreject Filters
   - Notch Filters
3. Implementation Topics
   - Separability of 2-D DFT
   - Using DFT for IDFT Computation

Image Sharpening in the Frequency Domain

- Fine image detail corresponds to high frequencies.

Image Sharpening in the Frequency Domain

- Fine image detail corresponds to high frequencies.
- Image sharpening can be achieved by highpass filtering.

Image Sharpening in the Frequency Domain

- Fine image detail corresponds to high frequencies.
- Image sharpening can be achieved by highpass filtering.
- Highpass filtering attenuates low frequencies and retains high frequencies.

Image Sharpening in the Frequency Domain

- Fine image detail corresponds to high frequencies.
- Image sharpening can be achieved by highpass filtering.
- Highpass filtering attenuates low frequencies and retains high frequencies.
- We can simply obtain a highpass filter $H_{HP}(u,v)$ as the complementary of a lowpass filter $H_{LP}(u,v)$:

$$H_{HP}(u,v) = 1 - H_{LP}(u,v).$$

Image Sharpening in the Frequency Domain

- Fine image detail corresponds to high frequencies.
- Image sharpening can be achieved by highpass filtering.
- Highpass filtering attenuates low frequencies and retains high frequencies.
- We can simply obtain a highpass filter $H_{HP}(u,v)$ as the complementary of a lowpass filter $H_{LP}(u,v)$:

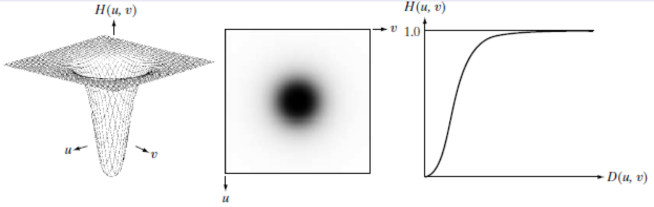
$$H_{HP}(u,v) = 1 - H_{LP}(u,v).$$

- We can design highpass filters in the frequency domain such as Ideal highpass, Gaussian highpass, Butterworth highpass, Laplacian highpass, Unsharp masking, and homomorphic filters.

Butterworth Highpass Filters

- A Butterworth highpass filter transfer function is defined as

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



Butterworth Highpass Filtering Example

- As LBPF radius increases, sharpening decreases.
- The spatial kernel approaches an impulse.

Frequency-Selective Filtering in the Frequency Domain

- Frequently, we are interested in filters that attenuate or pass a specific range of frequencies.

Frequency-Selective Filtering in the Frequency Domain

- Frequently, we are interested in filters that attenuate or pass a specific range of frequencies.
- Bandreject and bandpass filters attenuate or pass specific bands of frequencies.

Frequency-Selective Filtering in the Frequency Domain

- Frequently, we are interested in filters that attenuate or pass a specific range of frequencies.
- Bandreject and bandpass filters attenuate or pass specific bands of frequencies.
- We can design bandpass filters based on previous filter definitions.

Frequency-Selective Filtering in the Frequency Domain

- Frequently, we are interested in filters that attenuate or pass a specific range of frequencies.
- Bandreject and bandpass filters attenuate or pass specific bands of frequencies.
- We can design bandpass filters based on previous filter definitions.
- Notch filters pass or attenuate small regions of the frequency rectangle.

Ideal Bandpass Filters

- Ideal bandpass filter:

$$H(u,v) = \begin{cases} 1 & \text{if } D_0 - W/2 \leq D(u,v) \leq D_0 + W/2 \\ 0 & \text{otherwise .} \end{cases}$$

- $D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$, D_0: radial center of the band, W: width of band (or bandwidth).

Butterworth Bandpass Filters

- Butterworth bandpass filter:

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

- $D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$, D_0: radial center of the band, W: width of band (or bandwidth).

Gaussian Bandpass Filters

- Gaussian bandpass filter:

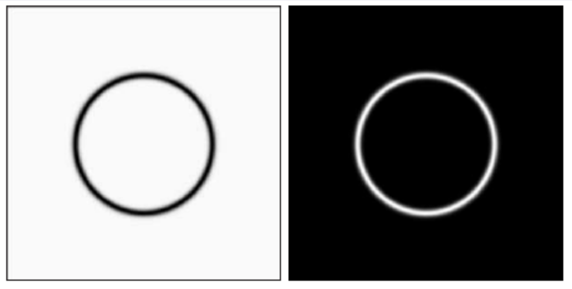
$$H(u,v) = e^{-[D(u,v)^2 - D_0^2]^2/(DW)^2}.$$

- $D(u,v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$, D_0: radial center of the band, W: width of band (or bandwidth).

Bandreject Filters

- A bandreject filter can be defined as the complementary of a bandpass filter:

$$H_{BR} = 1 - H_{BP}.$$



Notch Filters

- Notch filters pass or attenuate signal frequencies in a neighborhood of a specific frequency.

Notch Filters

- Notch filters pass or attenuate signal frequencies in a neighborhood of a specific frequency.
- To preserve zero-phase shift property, notch filters have to be symmetric.

Notch Filters

- Notch filters pass or attenuate signal frequencies in a neighborhood of a specific frequency.
- To preserve zero-phase shift property, notch filters have to be symmetric.
- Usually we design them as products of highpass filters with symmetric centers

$$H_{NR} = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

- $H_k(u,v), H_{-k}(u,v)$ are highpass filters with center frequencies at (u_k, v_k) and (u_{-k}, v_{-k}) respectively.

Butterworth Notch Reject Filters

- These are defined by:

$$H_{NR} = \prod_{k=1}^{Q} \frac{1}{1 + \left[\frac{D_{0k}}{D_k(u,v)}\right]^{2n}} \cdot \frac{1}{1 + \left[\frac{D_{0k}}{D_{-k}(u,v)}\right]^{2n}}$$

- $D_k(u,v) = [(u - P/2 - u_k)^2 + (v - Q/2 - v_k)^2]^{1/2}$
  $D_{-k}(u,v) = [(u - P/2 + u_k)^2 + (v - Q/2 + v_k)^2]^{1/2}$

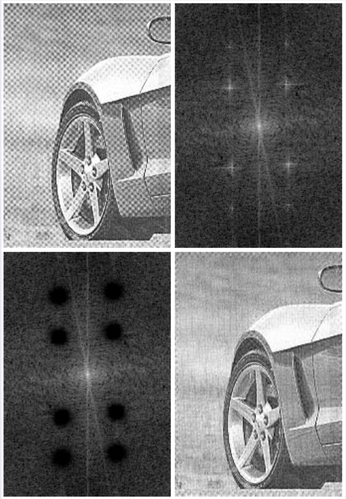
Notch Pass Filters

- A notch pass filter can be defined as the complementary of a notch reject filter

$$H_{NP} = 1 - H_{NR}.$$

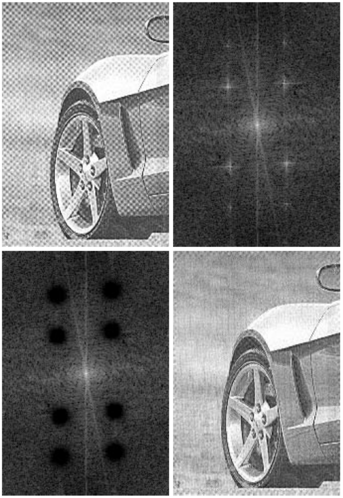
Notch Reject Filtering Example

- Image noise forming a Moiré pattern.



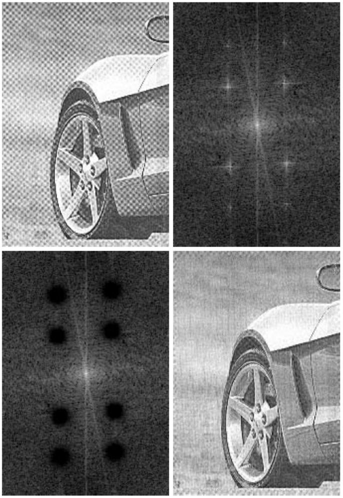
Notch Reject Filtering Example

- Image noise forming a Moiré pattern.
- Magnitude spectrum of the original image.



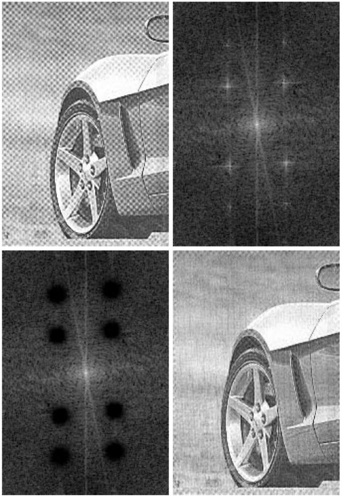
Notch Reject Filtering Example

- Image noise forming a Moiré pattern.
- Magnitude spectrum of the original image.
- Butterworth notch reject filter matching the noise pattern.



Notch Reject Filtering Example

- Image noise forming a Moiré pattern.
- Magnitude spectrum of the original image.
- Butterworth notch reject filter matching the noise pattern.
- Filtering removes noise.



Forward 2-D DFT

- Forward 2-D DFT

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

$$u = 0, 1, ..., M-1, v = 0, 1, ..., N-1.$$

Separable 2-D DFT

- DFT kernel is separable, therefore

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

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

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

Separable 2-D DFT

Separable DFT stages

1. We apply 1-D Fourier transform applied to rows yielding $F(x, v)$ that has N coefficients for each row.

Separable 2-D DFT

Separable DFT stages

1. We apply 1-D Fourier transform applied to rows yielding $F(x, v)$ that has N coefficients for each row.
2. We yield $F(u, v)$ by applying Fourier transform to the columns of $F(x, v)$.

Separable 2-D DFT

Separable DFT stages

1. We apply 1-D Fourier transform applied to rows yielding $F(x, v)$ that has N coefficients for each row.
2. We yield $F(u, v)$ by applying Fourier transform to the columns of $F(x, v)$.

- This property is used for fast implementations of DFT.

Using DFT for IDFT Computation

- Based on the 2-D IDFT definition we have that

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

Using DFT for IDFT Computation

- Based on the 2-D IDFT definition we have that

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

- Now we take the conjugate and solve for $f(x,y)$

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

$$MNf^*(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u,v) e^{-j2\pi(\frac{xu}{M} + \frac{yv}{N})}$$

$$MNf^*(x,y) = \mathscr{F}[F^*(u,v)]$$

$$f(x,y) = \frac{1}{MN} \{\mathscr{F}[F^*(u,v)]\}^*$$

Filtering in the Frequency Domain

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

February 26, 2015

Outline

Outline

Fourier Series

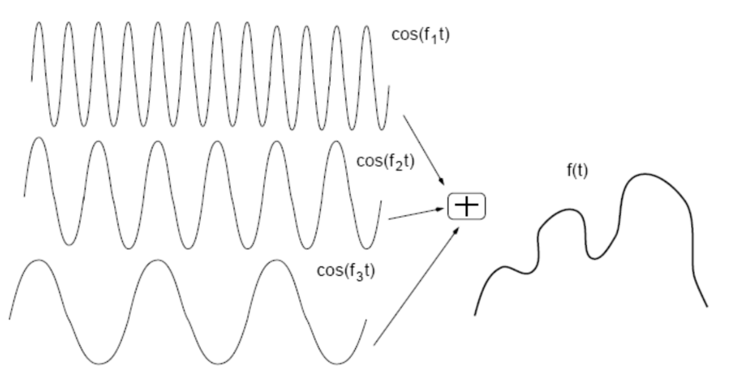
- A function $f(t)$ of a continuous variable t with period T can be expressed as the sum of sines and cosines multiplied with appropriate coefficients.
- This sum is known as Fourier expansion of $f(t)$ and is given by

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

- The coefficients c_n are given by:

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

Fourier Series



Impulses and Sifting Property

Unit Impulse

- A unit impulse $\delta(t)$ of a continuous variable t at $t = 0$ is defined as:

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

with

$$\int_{-\infty}^{\infty} \delta(t)dt = 1.$$

Impulses and Sifting Property

Sifting Property

- For a function $f(t)$ that is continuous at $t = 0$, we have that

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

- In general, for an impulse at an arbitrary point $t = t_0$ we have that

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

Impulses and Sifting Property

Discrete Case

- Impulse:

$$\delta(x) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

with

$$\sum_{x=-\infty}^{\infty} \delta(x) = 1.$$

- Sifting property:

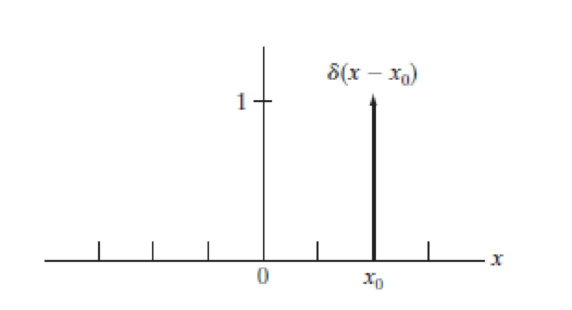
$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0)$$

- In general,

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$$

Impulses and Sifting Property

Discrete Impulse at x_0

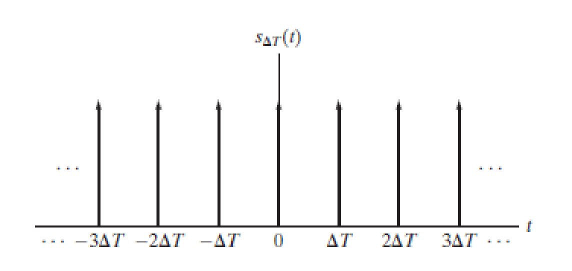


Impulses and Sifting Property

Discrete Case, Impulse Train

- Impulse train is the sum of infinitely many periodic impulses with period ΔT:

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



The Fourier Transform of Functions of One Continuous Variable

- The Fourier transform of function $f(t)$ denoted by $\mathscr{F}\{f(t)\}$ is given by
$$\mathscr{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt.$$
- $\mathscr{F}\{f(t)\} = F(\mu)$.
- Inverse Transform: $f(t) = \mathscr{F}^{-1}\{F(\mu)\} = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}d\mu$.
- We can use Euler's formula to write the forward transform as
$$F(\mu) = \int_{-\infty}^{\infty} f(t)[\cos(2\pi\mu t) - j\sin(2\pi\mu t)]dt.$$

Fourier Transform of a Pulse Function

$$
\begin{aligned}
F(\mu) &= \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt = \int_{-W/2}^{W/2} A e^{-j2\pi\mu t} dt \\
&= \frac{-A}{j2\pi\mu} [e^{-j2\pi\mu t}]_{-W/2}^{W/2} = \frac{-A}{j2\pi\mu} [e^{-j\pi\mu W} - e^{j\pi\mu W}] \\
&= \frac{A}{j2\pi\mu} [e^{j\pi\mu W} - e^{-j\pi\mu W}] = \frac{A}{\pi\mu} \sin(\pi\mu W) \\
&= AW \frac{\sin(\pi\mu W)}{\pi\mu W}
\end{aligned}
$$

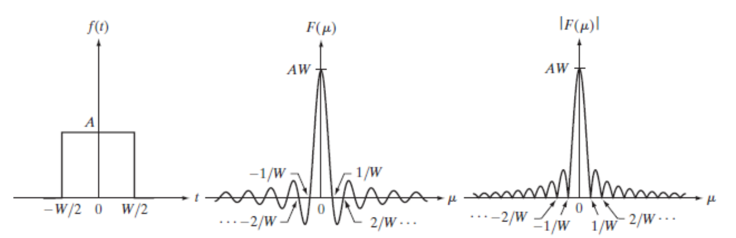
Fourier Transform of a Pulse Function

- Because the Fourier transform $F(\mu)$ of a function $f(t)$ is complex in general, often times we use its magnitude $|F(\mu)|$ for display purposes.
- The magnitude of Fourier transform $|F(\mu)|$ is called Fourier spectrum or frequency spectrum.
- For the case of a pulse function we have that

$$|F(\mu) = AW \left| \frac{\sin(\pi \mu W)}{\pi \mu W} \right|$$

Fourier Transform of a Pulse Function

Rectangular Pulse, Fourier Transform, and Fourier Spectrum



Fourier Transform of a Unit Impulse

Impulse at origin

$$
\begin{aligned}
F(\mu) &= \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi\mu t} dt \\
&= \int_{-\infty}^{\infty} e^{-j2\pi\mu t} \delta(t) dt \\
&= e^{-j2\pi\mu 0} \\
&= 1.
\end{aligned}
$$

Impulse at t_0

$$
\begin{aligned}
F(\mu) &= \int_{-\infty}^{\infty} \delta(t - t_0) e^{-j2\pi\mu t} dt \\
&= \int_{-\infty}^{\infty} e^{-j2\pi\mu t} \delta(t - t_0) dt \\
&= e^{-j2\pi\mu t_0} \\
&= \cos(2\pi\mu t_0) - j \sin(2\pi\mu t_0)
\end{aligned}
$$

Fourier Transform of an Impulse Train

Fourier series expansion

- Let an impulse train $s_{\Delta T}(t)$ with period ΔT, $s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$.
- Because it is periodic, it can be represented by Fourier series:

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{\Delta T}t}, c_n = \frac{1}{\Delta T}\int_{-\Delta T/2}^{\Delta T/2} S_{\Delta T}(t) e^{-j\frac{2\pi n}{\Delta T}t} dt$$

- For one period we have that,

$$c_n = \frac{1}{\Delta T}\int_{-\Delta T/2}^{\Delta T/2} \delta(t) e^{-j\frac{2\pi n}{\Delta T}t} dt = \frac{1}{\Delta T}.$$

- The Fourier series becomes : $s_{\Delta T}(t) = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}$.

Fourier Transform of an Impulse Train

Fourier Transform of series

- Now, we find $\mathscr{F}\{s_{\Delta T}(t)\} = S(\mu)$.
- Using the previous result we have that

$$
\begin{aligned}
S(\mu) &= \mathscr{F}\{\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi n}{\Delta T}t}\} \\
&= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \mathscr{F}\{e^{j\frac{2\pi n}{\Delta T}t}\} \\
&= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \delta(\mu - \frac{n}{\Delta T})\}.
\end{aligned}
$$

- Therefore, the Fourier transform of an impulse train is an impulse train with period reciprocal to the original one.

Convolution

- The convolution of two functions $f(t)$ and $h(t)$ is given by

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

- The Fourier transform of the convolution is

$$\begin{aligned}
\mathscr{F}\{f(t) * h(t)\} &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau \right] e^{-j2\pi\mu t}dt \\
&= \int_{-\infty}^{\infty} f(\tau) \left[\int_{-\infty}^{\infty} h(t-\tau)e^{-j2\pi\mu t}dt \right] d\tau \\
&= \int_{-\infty}^{\infty} f(\tau) \left[H(\mu)e^{-j2\pi\mu\tau} \right] d\tau \\
&= H(\mu) \int_{-\infty}^{\infty} f(\tau)e^{-j2\pi\mu\tau}d\tau \\
&= H(\mu)F(\mu)
\end{aligned}$$

Convolution

- Therefore we have that

$$\begin{aligned}\mathscr{F}\{f(t)*h(t)\} &= \mathscr{F}\{f(t)\}\mathscr{F}\{h(t)\} \\ &= H(\mu)F(\mu)\end{aligned}$$

- This result tells us the convolution in the spatial domain is equivalent to the product in the frequency domain.

Convolution Theorem

- The previous result can be symbolized as: $f(t)*h(t) \Leftrightarrow H(\mu)F(\mu)$.
- We can also show that $f(t)h(t) \Leftrightarrow H(\mu)*F(\mu)$.
- The above two expressions form the **Convolution Theorem**.

Sampling

- The analog signal acquired by imaging sensors is converted to digital by the processes of sampling and quantization.
- Let $f(t)$ be a continuous signal that we want to sample at uniform intervals ΔT.
- The sampling process can be represented by multiplication of $f(t)$ with an equally spaced impulse train. So the sampled function $\hat{f}$ is given by:

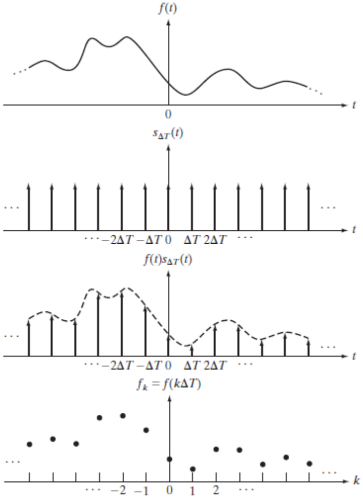
$$\hat{f}(t) = f(t)s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - n\Delta T)$$

- The value f_k at each sampling point k is given by: $f_k = \int_{-\infty}^{\infty} f(t)\delta(t - k\Delta T)dt = f(k\Delta T)$, for $k = 1, 2, \ldots$

Sampling

The Sampling Process

Continuous function, impulse train, sampled function, and sampled values (top to bottom).



Fourier Transform of Sampled Functions

- The Fourier transform of the sampled function $\hat{f}(t)$ is

$$\begin{aligned}\hat{F}(\mu) &= \mathscr{F}\{\hat{f}(t)\} \\ &= \mathscr{F}\{f(t)s_{\Delta T}\} \\ &= F(\mu) * S(\mu)\end{aligned}$$

Fourier Transform of Sampled Functions

$$
\begin{aligned}
\hat{F}(\mu) &= F(\mu) * S(\mu) \\
&= \int_{-\infty}^{\infty} F(\tau) S(\mu - \tau) d\tau \\
&= \int_{-\infty}^{\infty} F(\tau) \left[\sum_{n=-\infty}^{\infty} \frac{1}{\Delta T} \delta(\mu - \tau - \frac{n}{\Delta T}) \right] d\tau \\
&= \frac{1}{\Delta T} \int_{-\infty}^{\infty} F(\tau) \left[\sum_{n=-\infty}^{\infty} \delta(\mu - \tau - \frac{n}{\Delta T}) \right] d\tau \\
&= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} F(\tau) \delta(\mu - \tau - \frac{n}{\Delta T}) d\tau \\
&= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T})
\end{aligned}
$$

Fourier Transform of Sampled Functions

- The previous result shows that the Fourier transform of a sampled function $\hat{F}(\mu)$ is a sequence of repeated copies of $F(\mu)$.
- These copies are spaced $\frac{1}{\Delta T}$ apart.
- Because $F(\mu)$ is continuous, $\hat{F}(\mu)$ is also continuous.
- The quantity $\frac{1}{\Delta T}$ is the sampling rate.

The Sampling Theorem

- We showed that the Fourier transform of a sampled function is a periodic sequence of the Fourier transforms of the continuous function that are spaced $\frac{1}{\Delta T}$ apart.
- We can reconstruct the original signal if we can isolate one full period.
- This is guaranted when $\frac{1}{\Delta T} > 2\mu_{max}$, where μ_{max} is the maximum frequency.

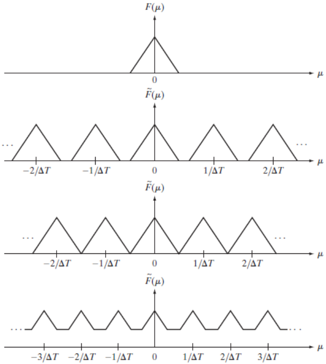
Theorem (Nyquist Theorem)

A continuous, band-limited function can be recovered completely by its samples, if the samples are acquired at a rate that exceeds twice the highest frequency coefficient of the function.

The Sampling Theorem

The Sampling Process

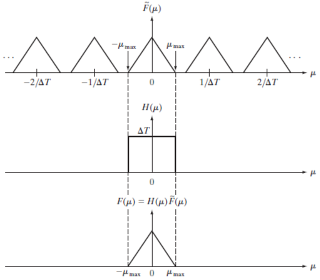
Fourier Transform of a band-limited function, transform of the corresponding sampled function with over-sampling, critical sampling, and under-sampling (top to bottom).



The Sampling Theorem

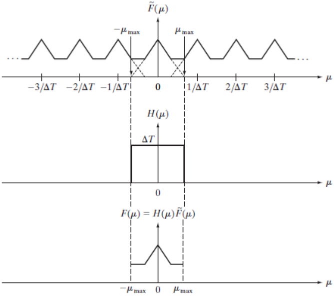
Original Signal Recovery

- Ideal lowpass filter: $H(\mu) = \begin{cases} \Delta T & -\mu_{max} \leq \mu \leq \mu_{max} \\ 0 & otherwise \end{cases}$
- Multiply: $F(\mu) = H(\mu)\tilde{F}(\mu)$
- Inverse Fourier Transform: $f(t) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu t} d\mu$



Aliasing

- The Nyquist theorem determines the sampling rate for complete signal recovery to be $\frac{1}{\Delta T} > 2\mu_{max}$.
- When the rate is smaller than required, successive periods will overlap.
- The recovered function will be corrupted by the frequency aliasing effect.



Aliasing

Reducing Aliasing Effect

- Aliasing effect is almost unavoidable, because when we limit the duration of a function, we introduce an infinite number of frequency components.
- A way to reduce this effect is by smoothing the input signal to attenuate the higher frequences, a process known as anti-aliasing. This process has to be applied before sampling.

Filtering in the Frequency Domain

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

March 3, 2015

Outline

Outline

Fourier Transform of Sampled Function

- We showed before that the Fourier transform of the sampled function $\hat{f}(t)$ is

$$\begin{aligned}\hat{F}(\mu) &= \mathscr{F}\{\hat{f}(t)\} \\ &= \mathscr{F}\{f(t)s_{\Delta T}\} \\ &= F(\mu) * S(\mu)\end{aligned}$$

Fourier Transform of Sampled Function

- We also showed that the transform $\hat{F}(\mu)$ of a sampled, and band-limited function $\hat{f}(t)$ with range $(-\infty, \infty)$ is given by

$$\begin{aligned}\hat{F}(\mu) &= F(\mu) * S(\mu) \\ &= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T}).\end{aligned}$$

- But this expression does not include $\hat{f}(t)$.
- Next, we will express $\hat{F}(\mu)$ in terms of $\hat{f}(t)$.

Obtaining the Discrete Fourier Transform

- Continuous Fourier transform of $\hat{f}(t)$

$$\begin{aligned}\hat{F}(\mu) &= \int_{-\infty}^{\infty} \hat{f}(t) e^{-j2\pi\mu t} dt \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(t)\delta(t - n\Delta T) e^{-j2\pi\mu t} dt \\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\delta(t - n\Delta T) e^{-j2\pi\mu t} dt \\ &= \sum_{n=-\infty}^{\infty} f_n e^{-j2\pi\mu n\Delta T}\end{aligned}$$

- Although f_n is discrete, $\hat{F}(\mu)$ is continuous and periodic with period $1/\Delta T$.
- We need one period to characterize $\hat{F}(\mu)$.

Obtaining the Discrete Fourier Transform

- We assume that we sample $\hat{F}(\mu)$ between 0 and $1/\Delta T$ using M points.
- The sampling frequencies would then be $\mu = \frac{m}{M\Delta T}$ for $m = 0, 1, ..., M-1$.
- From the previous result we have that

$$
\begin{aligned}
F_m &= \sum_{n=-\infty}^{\infty} f_n e^{-j2\pi\mu n\Delta T} \\
&= \sum_{n=0}^{M-1} f_n e^{-j2\pi\frac{m}{M\Delta T}n\Delta T} \\
&= \sum_{n=0}^{M-1} f_n e^{-j2\pi\frac{m}{M}n}, m = 0, 1, ..., M-1
\end{aligned}
$$

Discrete Fourier Transform Pair

- The expression

$$F_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi \frac{m}{M} n}, m = 0, 1, ..., M-1.$$

  is the Discrete Fourier Transform (DFT).
- This is a transformation from a set $\{f_n\}$ of M samples to another set $\{F_m\}$ of M samples.
- The Inverse Discrete Fourier Transform is defined as

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{j2\pi \frac{m}{M} n}, m = 0, 1, ..., M-1.$$

Discrete Fourier Transform Pair

- Using functional notation we can write the DFT pair as

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

and

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

Discrete Fourier Transform Considerations

- Forward and inverse DFT are periodic.

$$F(u) = F(u + kM), f(x) = f(x + kM)$$

- Convolution

$$f(x) * h(x) = \sum_{m=0}^{M-1} f(m)h(x - m)$$

  - The convolution is periodic, also referred to as circular convolution.
  - The convolution theorem holds for discrete variables.

Relation between the Sampling and Frequency Intervals

- Let $f(t)$ be a function sampled at M points that are ΔT units apart.
- The produced sampled signal $\{f(x)\}$ has duration $T = M \cdot \Delta T$.
- As we showed before, the spacing in the frequency domain is $\delta u = \frac{1}{M \cdot \Delta T} = \frac{1}{T}$.
- The range of the frequency spectrum is $\Omega = M \cdot \frac{1}{M \cdot \Delta T} = \frac{1}{\Delta T}$.
- Observe the inverse relationships between the sampling and frequency intervals.

2-D Impulse and Sifting Property

Impulse Definition

- A unit impulse $\delta(t,z)$ of continuous variables t and z is defined as:

$$\delta(t,z) = \begin{cases} \infty & \text{if } t = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

with

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t,z)dtdz = 1.$$

2-D Impulse and Sifting Property

Sifting Property

- For a function $f(t,z)$ that is continuous at $t=0,\ z=0$, we have that

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

- In general, for an impulse at an arbitrary point (t_0, z_0) we have that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0, z-z_0)dtdz = f(t_0, z_0)$$

Impulses and Sifting Property

Discrete Case

- Impulse:

$$\delta(x,y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{otherwise} \end{cases}$$

  with

$$\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} \delta(x,y) = 1.$$

Impulses and Sifting Property

Discrete Case

- Sifting property:

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x,y) = f(0,0)$$

- In general,

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\delta(x-x_0,y-y_0) = f(x_0,y_0)$$

2-D Continuous Fourier Transform Pair

- The Fourier transform of a continuous function $f(t,z)$ is

$$\mathscr{F}\{f(t,z)\} = F(\mu,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}dtdz.$$

- Inverse Transform:

$$f(t,z) = \mathscr{F}^{-1}\{F(\mu,\nu)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t+\nu z)}d\mu d\nu.$$

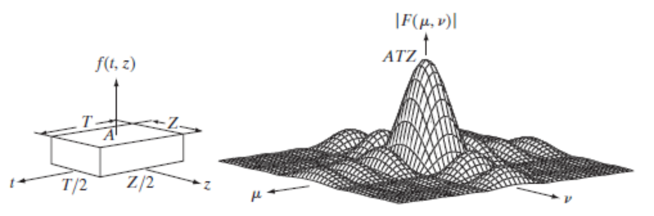
- We can use Euler's formula to write the forward transform as

$$F(\mu,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)[\cos(2\pi(\mu t+\nu z)) - j\sin(2\pi(\mu t+\nu z))]dtdz.$$

2-D Continuous Fourier Transform Pair

Fourier Transform of a 2-D Box

- Function: $F(\mu,\nu) = \int_{-T/2}^{T/2}\int_{-Z/2}^{Z/2} Ae^{-j2\pi(\mu t+\nu z)}dtdz.$
- Spectrum: $|F(\mu,\nu)| = ATZ\left|\frac{\sin(\pi\mu T)}{\pi\mu T}\right|\left|\frac{\sin(\pi\nu Z)}{\pi\nu Z}\right|.$



2-D Sampling and 2-D Sampling Theorem

- We first define the 2-D impulse train:

$$s_{\Delta T \Delta Z}(t,z) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - m\Delta T)\delta(z - n\Delta Z),$$

where ΔT and ΔZ are sampling intervals in t and z.

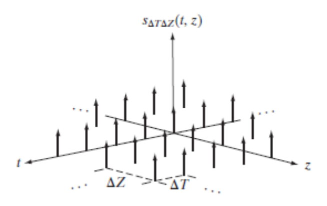


Figure: 2-D impulse train.

2-D Sampling and 2-D Sampling Theorem

- We multiply our signal with the impulse train to produce the sampled function.

$$\begin{aligned}\hat{f}(t,z) &= f(t,z)s_{\Delta T \Delta Z}(t,z)\\ &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(t,z)\delta(t-m\Delta T)\delta(z-n\Delta Z)\end{aligned}$$

- The question is how to recover $f(t,z)$ after the sampling process from $\hat{f}(t,z)$.

2-D Sampling and 2-D Sampling Theorem

Band-limited 2-D Function

- A function $f(t,z)$ is band-limited if its Fourier transform $F(\mu,\nu)$ satisfies the following:

$$|F(\mu,\nu)| = 0, \text{ for } |\mu| > \mu_{max} \text{ and } |\nu| > \nu_{max}$$

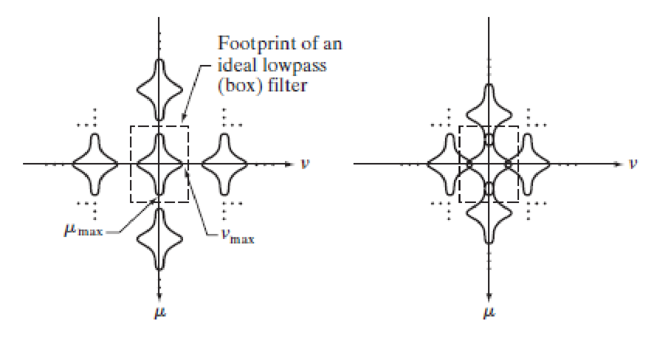
Theorem (2-D Sampling Theorem)

A continuous, band-limited function $f(t,z)$ can be recovered with no error after sampling with intervals ΔT and ΔZ if

$$\frac{1}{\Delta T} > 2\mu_{max} \text{ and } \frac{1}{\Delta Z} > 2\nu_{max}$$

2-D Sampling and 2-D Sampling Theorem

2-D Over- and Under-sampling



Aliasing in Images

- As in 1-D case, aliasing happens when the conditions of sampling theorem are violated.
- The condition of a band-limited function is violated when we limit the duration of a function in original space.
  - Some aliasing is present in all digital images as in sampled 1-D functions.
- Another source of aliasing is the sampling interval.
  - Two types: spatial aliasing, or temporal aliasing.
  - Spatial aliasing is caused by undersampling in spatial domain.
  - Temporal aliasing is caused by undersampling in temporal domain.

Aliasing in Images

Anti-aliasing Solutions

- Aliasing can be reduced by de-focusing the sensed scene before it is digitized.
- Post-digitization "anti-aliasing" filters blur the image to reduce the aliasing caused by resampling.
- Some digital cameras have true anti-aliasing filters, either in the lens, or on sensor's surface.

Aliasing in Images

Aliasing and Interpolation

- Aliasing can be caused when we resample by pixel replication, mainly during image shrinking.
- This is because we increase the sampling interval by skipping rows and columns.
- One solution is to smooth the image before interpolation.
- Other manifestations of aliasing are jaggies (blocky edges) and Moire patterns.

Aliasing in Images

Example

Figure: Original image, under- and over-sampled back to original size using pixel-replication with aliasing artifacts, use of blurring before undersampling reduces aliasing (left to right).

2-D Discrete Fourier Transform and Its Inverse

2-D Discrete Fourier Transform (DFT) Pair

- Forward 2-D DFT

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

$$u = 0,1,...,M-1, v = 0,1,...,N-1.$$

- Inverse 2-D DFT

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

$$x = 0,1,...,M-1, y = 0,1,...,M-1.$$

2-D Discrete Fourier Transform Properties

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

March 17, 2015

Outline

Sampling and frequency intervals

- Let $f(x,y)$ be a sampled image with $M \times N$ samples in dimensions t and z.
- Let ΔT, ΔZ be the sampling intervals in t and z respectively.
- Then the frequency intervals are given by

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

and

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

Periodicity

- The 2-D DFT and its inverse are infinitely periodic in u and v

$$\begin{aligned} F(u,v) &= F(u+k_1M, v) \\ &= F(u, v+k_2N) \\ &= F(u+k_1M, v+k_2N) \end{aligned}$$

$$\begin{aligned} f(x,y) &= f(x+k_1M, y) \\ &= f(x, y+k_2N) \\ &= f(x+k_1M, y+k_2N) \end{aligned}$$

Translation

Translation in spatial domain

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

Translation in frequency domain

$$f(x, y)e^{j2\pi(\frac{xu_0}{M} + \frac{yu_0}{N})} \Leftrightarrow F(u - u_0, v - v_0)$$

Rotation

- Rotating $f(x, y)$ by angle θ_0 rotates $F(u, v)$ by angle θ_0

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

where $x = r\cos\theta$, $y = r\sin\theta$ and $u = \omega\cos\phi$, $v = \omega\sin\phi$.

Using Shifting and Periodicity Properties

- For visualization and filtering purposes we may shift the spectrum by half a period.
- It follows from the translation properties that $f(x,y)e^{j2\pi(\frac{xu_0}{M}+\frac{yu_0}{N})} \Leftrightarrow F(u-u_0, v-v_0)$.

Using Shifting and Periodicity Properties

- If we set $(u_0, v_0) = (M/2, N/2)$ we have that

$$\begin{aligned} f(x,y)e^{j2\pi(\frac{xM/2}{M}+\frac{yN/2}{N})} &\Leftrightarrow F(u-M/2, v-N/2) \\ f(x,y)e^{j\pi(x+y)} &\Leftrightarrow F(u-M/2, v-N/2) \\ f(x,y)(-1)^{(x+y)} &\Leftrightarrow F(u-M/2, v-N/2) \end{aligned}$$

- In this way, the coefficients are shifted so that $F(0,0)$ appears at the $(M/2, N/2)$.

Forward 2-D DFT

- Forward 2-D DFT

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

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

Separable DFT

- DFT kernel is separable, therefore

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

Separable DFT

Separable DFT stages

1. We apply 1-D Fourier transform applied to rows yielding $F(x, v)$ that has N coefficients for each row.
2. We yield $F(u, v)$ by applying Fourier transform to the columns of $F(x, v)$.

This property is used for fast implementations of DFT.

Symmetry Basics

- Any real or complex function $w(x, y)$ can be written as the sum of an even and odd part, $w_e(x, y)$ and $w_o(x, y)$

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

- Even (or symmetric) and odd (or antisymmetric) parts are defined as
$w_e(x, y) = \frac{w(x,y)+w(-x,-y)}{2}$ and $w_o(x, y) = \frac{w(x,y)-w(-x,-y)}{2}$

Symmetry Basics

- It follows that
  $w_e(x, y) = w_e(-x, -y)$ and $w_o(x, y) = -w_o(-x, -y)$.
- Because we deal with positive indices the above definitions become
  $w_e(x, y) = w_e(M - x, M - y)$ and
  $w_o(x, y) = -w_o(N - x, N - y)$.

Symmetry Properties for DFT

- If $f(x,y)$ is real then its Fourier transform is conjugate symmetric
$$F^*(u,v) = F(-u,-v)$$
- If $f(x,y)$ is imaginary then its Fourier transform is conjugate antisymmetric
$$F^*(u,v) = -F(-u,-v)$$
- $f(x,y)$ is real and even $\Leftrightarrow$ $F(u,v)$ is real and even.
- $f(x,y)$ is real and odd $\Leftrightarrow$ $F(u,v)$ is imaginary and odd.

More Symmetry Properties

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

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

Linearity and Multiplication

Linearity

$$F[af(x,y)+bg(x,y)] = aF[f(x,y)]+bF[g(x,y)]$$

Multiplication does not hold

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

The DC Coefficient

- When we set $(u, v) = (0, 0)$, then $F(u, v)$ becomes

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

  where $\hat{f}(x,y)$ is the mean of $f(x,y)$.
- Therefore, $|F(0,0)| = MN|\hat{f}(x,y)|$.
- Because the frequency indices are zero, $F(0,0)$ is also called the DC coefficient.

The DC Coefficient

- The term $F(0,0)$ is usually the largest component of the spectrum by a large margin.
- To compress the range of Fourier coefficients, we usually apply log transform to display the Fourier spectrum.
- Another step is to shift the spectrum so that the DC coefficient appears at $(M/2, N/2)$.

Magnitude and Phase of DFT

2-D DFT in polar form

- The Fourier transform of a function is complex and can be written as
$$F(u,v) = |F(u,v)|e^{j\phi(u,v)}.$$
- The magnitude $|F(u,v)|$ is called the Fourier spectrum or frequency spectrum calculated by
$$|F(u,v)| = \left[Re(F(u,v))^2 + Im(F(u,v))^2\right]^{1/2}.$$
- The phase angle $\phi(u,v)$ is given by
$$\phi(u,v) = \arctan\left[\frac{Im(F(u,v))}{Re(F(u,v))}\right].$$

Power Spectrum

- The power spectrum $|F(u,v)|^2$ is given by $|F(u,v)|^2 = \left[Re(F(u,v))^2 + Im(F(u,v))^2\right]$.

Magnitude and Phase Visualization

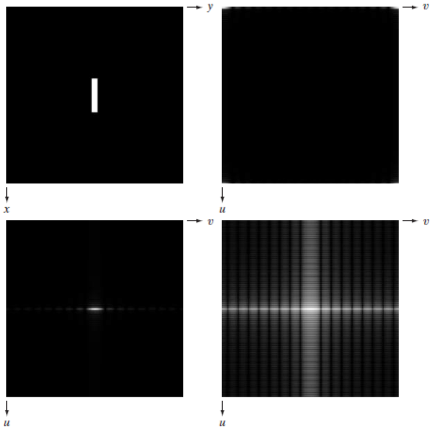


Figure: Original image, its spectrum (top row, left to right) its centered spectrum, and display of centered spectrum after log intensity transform (bottom row, left to right).

Magnitude and Phase Information

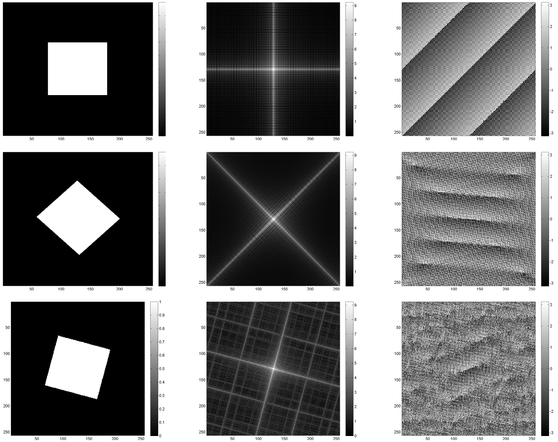


Figure: Examples of an original image, its DFT magnitude, and its DFT phase (left to right).

Magnitude and Phase Information

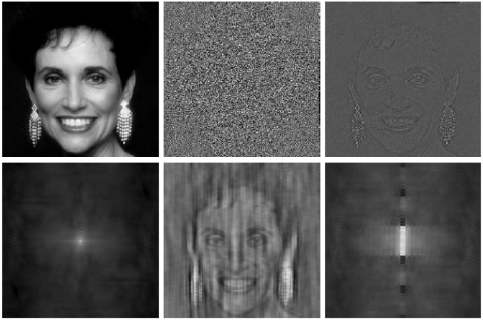


Figure: Original image, its phase, reconstructed with phase only (top row, left to right), reconstructed with magnitude only, reconstructed using phase from face image and magnitude from rectangle image, reconstructed using phase from rectangle image and magnitude from face image (bottom row, left to right).

2-D Convolution

- We can extend the findings for the 1-D DFT to 2-D.
- Circular convolution is defined by:

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

where $x = 0, \ldots, M-1$ and $y = 0, \ldots, N-1$.

2-D Convolution Theorem

- We saw that the convolution theorem is applicable to discrete variables as well.
- This can be expressed as

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v) \cdot H(u,v)$$

and

$$f(x,y) \cdot h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

2-D Convolution Theorem Application

- For an image M and a spatial filter S, we can compute $F * S$ as follows:
  1. Pad S with 0s so that it is the same size as M. Denote padded S by S'.

2-D Convolution Theorem Application

- For an image M and a spatial filter S, we can compute $F * S$ as follows:
  1. Pad S with 0s so that it is the same size as M. Denote padded S by S'.
  2. Compute DFT's of M and S:

$$\mathscr{F}(M) \text{ and } \mathscr{F}(S').$$

2-D Convolution Theorem Application

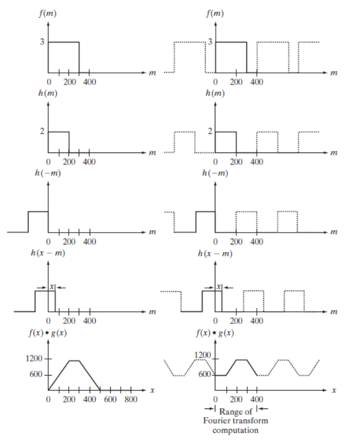
- For an image M and a spatial filter S, we can compute $F * S$ as follows:
    1. Pad S with 0s so that it is the same size as M. Denote padded S by S'.
    2. Compute DFT's of M and S:
    $$\mathscr{F}(M) \text{ and } \mathscr{F}(S').$$
    3. Multiply $\mathscr{F}(M)$ and $\mathscr{F}(S')$:
    $$\mathscr{F}(M) \cdot \mathscr{F}(S').$$

2-D Convolution Theorem Application

- For an image M and a spatial filter S, we can compute $F * S$ as follows:
  1. Pad S with 0s so that it is the same size as M. Denote padded S by S'.
  2. Compute DFT's of M and S:
  $$\mathscr{F}(M) \text{ and } \mathscr{F}(S').$$
  3. Multiply $\mathscr{F}(M)$ and $\mathscr{F}(S')$:
  $$\mathscr{F}(M) \cdot \mathscr{F}(S').$$
  4. Compute inverse DFT of $\mathscr{F}(M)$ and $\mathscr{F}(S')$:
  $$\mathscr{F}^{-1}(\mathscr{F}(M) \cdot \mathscr{F}(S')).$$

Wraparound Error

- The process we described corresponds to non-periodic functions (left column).
- But we must consider the signal periodicity assumption in DFT definition and use.
- Then the convolution produces erroneous results, caused by the **wraparound error**.



Adressing Wraparound Error

- We can avoid the wraparound error by appending 0s to both functions (zero padding) so they have length P such that

$$P \geq A + B - 1.$$

- Because many DFT implementations are designed for even matrix sizes, we can pick P as the smallest even number satisfying the above condition.

Adressing Wraparound Error: Frequency Leakage

- If the signals are not zero at the end of sampling interval, then zero padding introduces a discontinuity represented by infinite frequencies in Fourier domain.

Adressing Wraparound Error: Frequency Leakage

- If the signals are not zero at the end of sampling interval, then zero padding introduces a discontinuity represented by infinite frequencies in Fourier domain.
- This effect is called **frequency leakage** that appears as blocky effect on images.

Reducing Frequency Leakage

- Frequency leakage can be reduced by multiplying the sampled function by a function that reduces smoothly to zero at the two ends of sampling interval.

Reducing Frequency Leakage

- Frequency leakage can be reduced by multiplying the sampled function by a function that reduces smoothly to zero at the two ends of sampling interval.
- The above approach is called **windowing** or **apodizing**.

Reducing Frequency Leakage

- Frequency leakage can be reduced by multiplying the sampled function by a function that reduces smoothly to zero at the two ends of sampling interval.
- The above approach is called **windowing** or **apodizing**.
- Frequently, we use a Gaussian function for apodization.

2-D DFT Definitions

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
3) Polar representation	$F(u, v) = \lvert F(u, v) \rvert e^{j\phi(u,v)}$
4) Spectrum	$\lvert F(u, v) \rvert = \left[R^2(u, v) + I^2(u, v)\right]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)}\right]$
6) Power spectrum	$P(u, v) = \lvert F(u, v) \rvert^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

Name	Expression(s)
8) Periodicity (k_1 and k_2 are integers)	$F(u, v) = F(u + k_1 M, v) = F(u, v + k_2 N)$ $= F(u + k_1 M, v + k_2 N)$ $f(x, y) = f(x + k_1 M, y) = f(x, y + k_2 N)$ $= f(x + k_1 M, y + k_2 N)$
9) Convolution	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
10) Correlation	$f(x, y) \mathbin{\text{☆}} h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.2.

2-D DFT Pairs

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M + vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$
6) Convolution theorem†	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

Name	DFT Pairs
7) Correlation theorem†	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v)\,H(u, v)$ $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\mathrm{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0x + 2\pi v_0y) \Leftrightarrow$ $j\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$
11) Cosine	$\cos(2\pi u_0x + 2\pi v_0y) \Leftrightarrow$ $\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) *Differentiation* (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)	$\left(\frac{\partial}{\partial t}\right)^m\left(\frac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m(j2\pi\nu)^n F(\mu, \nu)$ $\frac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \frac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
13) *Gaussian*	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

†Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

Introduction to Image Restoration

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 16, 2015

Outline

Image Restoration

- Goals of both image enhancement and restoration are to improve the image quality.

Image Restoration

- Goals of both image enhancement and restoration are to improve the image quality.
- Enhancement techniques are mostly heuristic and subjective.

Image Restoration

- Goals of both image enhancement and restoration are to improve the image quality.
- Enhancement techniques are mostly heuristic and subjective.
- Restoration attempts to recover an image that has been degraded by using a priori knowledge of degradation process.

Image Restoration

- Goals of both image enhancement and restoration are to improve the image quality.
- Enhancement techniques are mostly heuristic and subjective.
- Restoration attempts to recover an image that has been degraded by using a priori knowledge of degradation process.
- Restoration techniques first model the degradation process, then apply the inverse process to recover original image.

Modeling Degradation and Restoration Processes

- Here we assume that an image $f(x,y)$ undergoes a degradation process modeled by function $h(x,y)$ followed by corruption by additive noise $n(x,y)$.

$$g(x,y) = h(x,y) * f(x,y) + n(x,y).$$

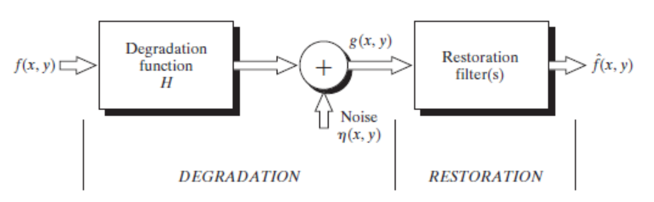
Modeling Degradation and Restoration Processes

- Here we assume that an image $f(x,y)$ undergoes a degradation process modeled by function $h(x,y)$ followed by corruption by additive noise $n(x,y)$.

$$g(x,y) = h(x,y) * f(x,y) + n(x,y).$$

- In the frequency domain this becomes

$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$



Noise Models

- Noise may occur during acquisition and transmission.

Noise Models

- Noise may occur during acquisition and transmission.
- During acquisition with a CCD camera, noise level is affected by light levels and sensor temperature.

Noise Models

- Noise may occur during acquisition and transmission.
- During acquisition with a CCD camera, noise level is affected by light levels and sensor temperature.
- Noise can corrupt a signal during transmission as well, especially if the signal is in analog form.

Spatial and Frequency Noise Properties

- It is useful to model the characteristics of noise in spatial and frequency domain.

Spatial and Frequency Noise Properties

- It is useful to model the characteristics of noise in spatial and frequency domain.
- Here we assume the noise is
  - independent of spatial coordinates (not true for periodic noise) and
  - that it is not correlated with the image.

Noise Probability Density Functions

- Most types of noise are modeled by a probability density function.

Noise Probability Density Functions

- Most types of noise are modeled by a probability density function.
- The noise models are typically chosen based on some understanding of the noise source.

Gaussian Noise

- The pdf is

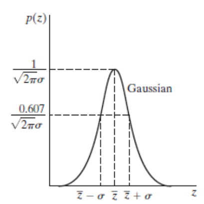
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(z-\bar{z})^2}{2\sigma^2}}.$$

Gaussian Noise

- The pdf is

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

- Gaussian noise is caused by
  - electronic circuit noise
  - sensor noise due to poor illumination and/or high temperature.



Rayleigh Noise

- The pdf is given by

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

$$\bar{z} = a + \sqrt{\frac{\pi b}{4}}, \ \sigma^2 = \frac{b(4-\pi)}{4}.$$

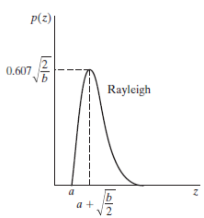
Rayleigh Noise

- The pdf is given by

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

$$\bar{z} = a + \sqrt{\frac{\pi b}{4}}, \ \sigma^2 = \frac{b(4-\pi)}{4}.$$

- Typically used to characterize noise in range imaging.



Erlang (gamma) Noise

- The pdf is given by

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

$\bar{z} = b/a$, $\sigma^2 = b/a^2$.

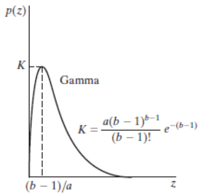
Erlang (gamma) Noise

- The pdf is given by

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

$\bar{z} = b/a$, $\sigma^2 = b/a^2$.

- Typically used to characterize noise in laser imaging.



Exponential Noise

- The pdf is given by

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

$\bar{z} = 1/a$, $\sigma^2 = 1/a^2$.

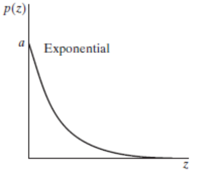
Exponential Noise

- The pdf is given by

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

$\bar{z} = 1/a$, $\sigma^2 = 1/a^2$.

- Typically used to characterize noise in laser imaging.



Uniform Noise

- The pdf is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & otherwise \end{cases}$$

$$\bar{z} = \frac{a+b}{2}/a, \ \sigma^2 = \frac{(b-a)^2}{12}.$$

Uniform Noise

- The pdf is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & otherwise \end{cases}$$

$\bar{z} = \frac{a+b}{2}/a, \; \sigma^2 = \frac{(b-a)^2}{12}.$

- Least used in practice.

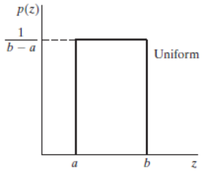
Uniform Noise

- The pdf is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & otherwise \end{cases}$$

$\bar{z} = \frac{a+b}{2}/a, \ \sigma^2 = \frac{(b-a)^2}{12}.$

- Least used in practice.
- Useful as basis for random number generators.



Impulse (Salt and Pepper) Noise

- The pdf is given by

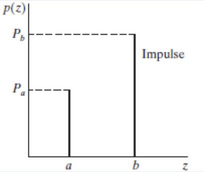
$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \end{cases}$$

Impulse (Salt and Pepper) Noise

- The pdf is given by

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \end{cases}$$

- Common when quick transients (eg, faulty switching) occur during imaging.



Noise Model Examples

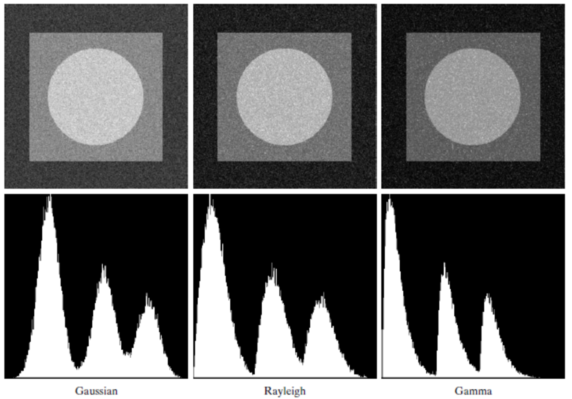


Figure: Images and histograms after adding Gaussian, Rayleigh and Gamma noise to a synthetic image.

Noise Model Examples

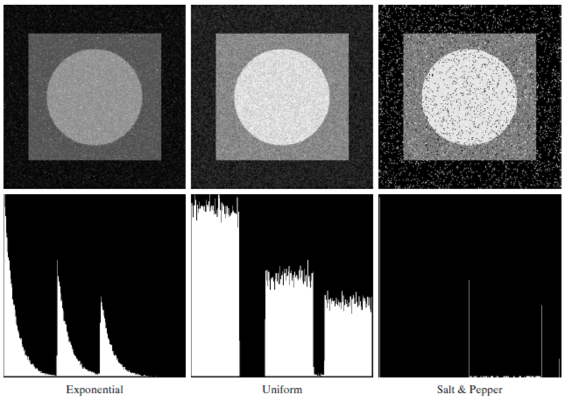


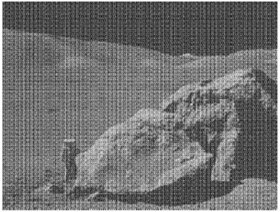
Figure: Images and histograms after adding Exponential, Uniform and Impulse to a synthetic image.

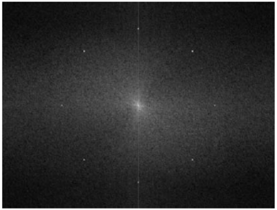
Periodic Noise

- This noise is usually caused by some electromechanical interference.

Periodic Noise

- This noise is usually caused by some electromechanical interference.
- The spectrum of this noise type will show symmetric peaks at the noise frequencies.



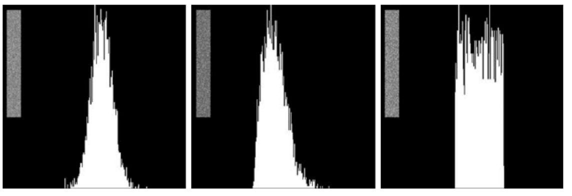


Noise Parameter Estimation

- Estimate the parameters of noise pdf from small patches of reasonably constant background intensity.

Noise Parameter Estimation

- Estimate the parameters of noise pdf from small patches of reasonably constant background intensity.
- For impulse noise, estimate probability of black/white pixels.



Restoration in the Presence of Noise only

- If noise can be estimated, it can be subtracted from the input image

Restoration in the Presence of Noise only

- If noise can be estimated, it can be subtracted from the input image
- We follow 2 steps
  1. estimate $n(x, y)$
  2. $g(x, y) = f(x, y) + n(x, y) \Rightarrow \hat{f}(x, y) = g(x, y) - \hat{n}(x, y).$

Restoration in the Presence of Noise only

- If noise can be estimated, it can be subtracted from the input image
- We follow 2 steps
  1. estimate $n(x,y)$
  2. $g(x,y) = f(x,y) + n(x,y) \Rightarrow \hat{f}(x,y) = g(x,y) - \hat{n}(x,y)$.
- It is very difficult to subtract noise when it is independent from the spatial coordinates.

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.
- Filtering is better suited for reducing additive and random noise.

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.
- Filtering is better suited for reducing additive and random noise.
- The main types of filters for denoising are

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.
- Filtering is better suited for reducing additive and random noise.
- The main types of filters for denoising are
    - Mean filters (arithmetic, geometric, harmonic, etc)

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.
- Filtering is better suited for reducing additive and random noise.
- The main types of filters for denoising are
  - Mean filters (arithmetic, geometric, harmonic, etc)
  - Order statistics filters (median, min, max)

Restoration in the Presence of Noise only

- If noise can not be estimated, filtering methods are used to suppress noise.
- Filtering is better suited for reducing additive and random noise.
- The main types of filters for denoising are
  - Mean filters (arithmetic, geometric, harmonic, etc)
  - Order statistics filters (median, min, max)
  - Frequency-domain filters.

Arithmetic Mean Filter

- The filtering operation is given by

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

Arithmetic Mean Filter

- The filtering operation is given by

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

- It applies blurring that reduces noise, but removes image detail as well.

Geometric Mean Filter

- The filtering operation is given by

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{x,y}} g(s,t) \right]^{\frac{1}{mn}}.$$

Geometric Mean Filter

- The filtering operation is given by

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{x,y}} g(s,t) \right]^{\frac{1}{mn}}.$$

- This filter applies smoothing too, and removes less image detail than the arithmetic mean.

Harmonic Mean Filter

- The filtering process is given by

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

Harmonic Mean Filter

- The filtering process is given by

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

- The harmonic filter works well for salt noise but cannot address pepper noise. It works well for Gaussian noise too.

Contraharmonic Mean Filter

- The filtering operation is given by

$$\hat{f}(x, y) = \frac{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q}}.$$

Contraharmonic Mean Filter

- The filtering operation is given by

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q}}.$$

- $Q > 0$ pepper noise.

Contraharmonic Mean Filter

- The filtering operation is given by

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q}}.$$

- $Q > 0$ pepper noise.
- $Q < 0$ salt noise.

Contraharmonic Mean Filter

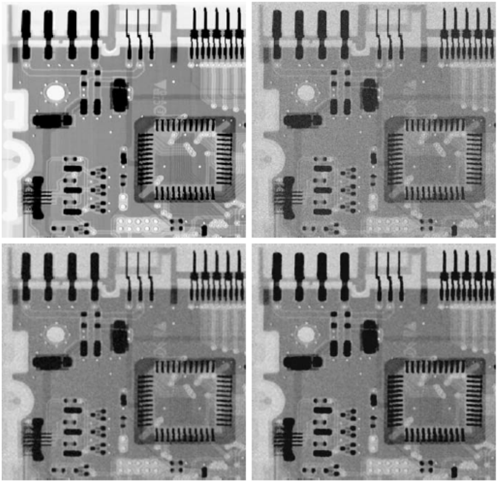
- The filtering operation is given by

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q}}.$$

- $Q > 0$ pepper noise.
- $Q < 0$ salt noise.
- $Q = 0$ becomes an arithmetic mean filter.

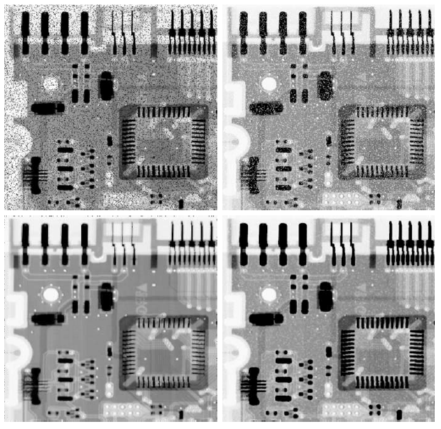
Arithmetic and Geometric Mean Filter Example

- The original image, image corrupted by Gaussian noise, filtered by arithmetic mean, filtered by geometric mean filter in clockwise order.



Contraharmonic Mean Filter Example

- Image corrupted by pepper noise, image corrupted by salt noise, pepper noise filtered by contraharmonic filter of order 1.5, salt noise filtered by contraharmonic filter with $Q = -1.5$ in clockwise order.



Order-statistic Filters

- Order-statistic filters are nonlinear spatial filters.

Order-statistic Filters

- Order-statistic filters are nonlinear spatial filters.
- Their response is based on ordering (ranking) of the pixels contained in an area covered by the filter kernel.

Median Filter

- The filtering operation is given by

$$\hat{f}(x, y) = \text{median}_{(s,t)\in S_{x,y}}\{g(s, t)\}.$$

- It produces less blurring than the arithmetic filters.

Max and Min Filters

- The max filtering operation is given by

$$\hat{f}(x,y) = \max_{(s,t)\in S_{x,y}}\{g(s,t)\}.$$

Max and Min Filters

- The max filtering operation is given by

$$\hat{f}(x, y) = \max_{(s,t) \in S_{x,y}} \{g(s, t)\}.$$

- The max filter is good for pepper noise.

Max and Min Filters

- The max filtering operation is given by

$$\hat{f}(x, y) = \max_{(s,t) \in S_{x,y}} \{g(s, t)\}.$$

- The max filter is good for pepper noise.
- The min filtering operation is given by

$$\hat{f}(x, y) = \min_{(s,t) \in S_{x,y}} \{g(s, t)\}.$$

- The min filter is good for salt noise.

Midpoint Filter

- This filtering operation is given by

$$\hat{f}(x,y) = 1/2\left[\max_{(s,t)\in S_{x,y}}\{g(s,t)\} + \min_{(s,t)\in S_{x,y}}\{g(s,t)\}\right].$$

Midpoint Filter

- This filtering operation is given by

$$\hat{f}(x,y) = 1/2\left[\max_{(s,t)\in S_{x,y}}\{g(s,t)\} + \min_{(s,t)\in S_{x,y}}\{g(s,t)\}\right].$$

- This can be seen as an averaging-order statistics hybrid.

Midpoint Filter

- This filtering operation is given by

$$\hat{f}(x, y) = 1/2 \left[\max_{(s,t) \in S_{x,y}} \{g(s, t)\} + \min_{(s,t) \in S_{x,y}} \{g(s, t)\} \right].$$

- This can be seen as an averaging-order statistics hybrid.
- It works best for Gaussian or uniform noise.

Alpha-trimmed Mean Filter

- Good for multiple types of noise such as Gaussian noise, and salt-and-pepper noise.

Alpha-trimmed Mean Filter

- Good for multiple types of noise such as Gaussian noise, and salt-and-pepper noise.
- Assume an $m \times n$ neighborhood
  1. Disregard d/2 lowest and d/2 highest values.
  2. Average the remaining values.

Alpha-trimmed Mean Filter

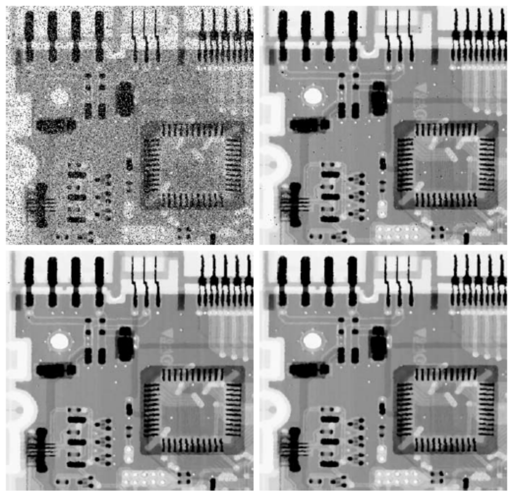
- Good for multiple types of noise such as Gaussian noise, and salt-and-pepper noise.
- Assume an $m \times n$ neighborhood
    1. Disregard d/2 lowest and d/2 highest values.
    2. Average the remaining values.
- $d = 0$, arithmetic mean

Alpha-trimmed Mean Filter

- Good for multiple types of noise such as Gaussian noise, and salt-and-pepper noise.
- Assume an $m \times n$ neighborhood
  1. Disregard d/2 lowest and d/2 highest values.
  2. Average the remaining values.
- $d = 0$, arithmetic mean
- $d = mn - 1$ median.

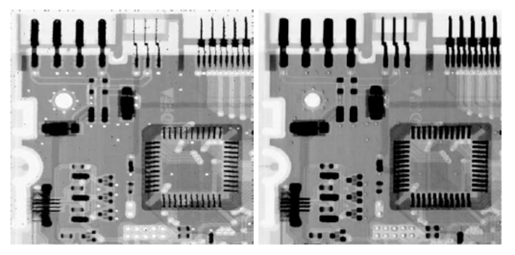
Median Filter Example

- Image corrupted by salt and pepper noise, salt and pepper noise filtered by median filter, second pass of median filter, third pass of median filter in clockwise order.



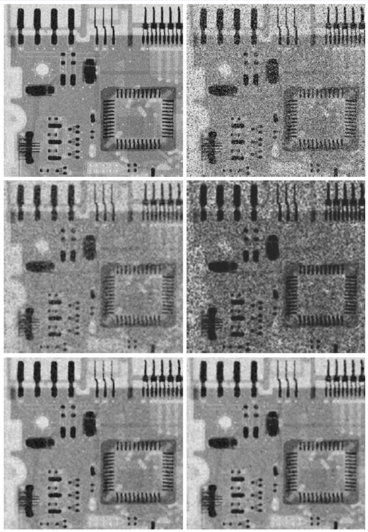
Max and Min Filter Example

- Max filter applied to pepper noise (left), and min filter applied to salt noise (right).



Mean, Median, Alpha-trimmed Mean Filter Examples

- Image corrupted by uniform noise, additional salt and pepper noise, arithmetic mean filter output, geometric mean filter output, median filter output, alpha-trimmed mean filter output (left to right and top to bottom).



Adaptive Filters

- They use non-fixed (i.e., adaptive) parameters

Adaptive Filters

- They use non-fixed (i.e., adaptive) parameters
- Typically, adaptive filters have superior performance compared to non-adaptive filters.

Adaptive Filters

- They use non-fixed (i.e., adaptive) parameters
- Typically, adaptive filters have superior performance compared to non-adaptive filters.
- But they have higher computational complexity.

Adaptive Median Filter

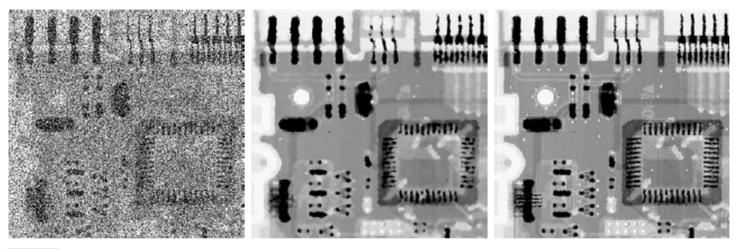
- Size of filtered region is not fixed.

Adaptive Median Filter

- Size of filtered region is not fixed.
- Its operation depends on the statistical characteristics of the pixel values inside the filtered region.

Adaptive Median Filter Example

- Image corrupted by salt and pepper noise, median filter output, adaptive median filter output (left to right).



Restoring Image Degradation

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 16, 2015

Outline

1. Image Restoration to address Degradation
   - Estimating the Degradation
   - Inverse Filtering
   - Wiener Filtering
   - Constrained Least Squares (CLS) Filtering

Degradation Models

- We assume the following degradation and noise corruption scenario

$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$

Degradation Models

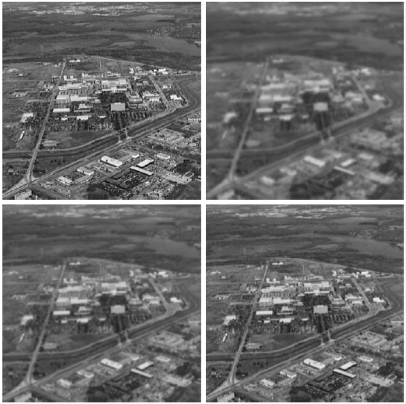
- We assume the following degradation and noise corruption scenario
$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$
- H can be modeled mathematically, for example to simulate atmospheric turbulence and motion.

Degradation Because of Atmospheric Turbulence

- The atmospheric turbulence can be modeled as

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}.$$

- From left to right and top to bottom: original image, degraded with $k = 0.0025, 0.001, 0.00025$.

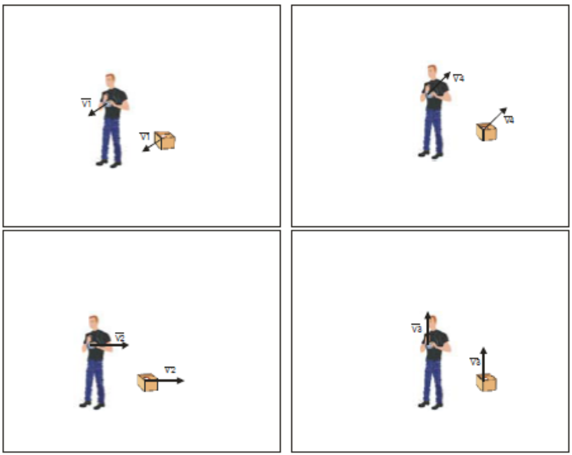


Degradation Because of Linear Motion

- Consider the case of camera/object planar motion.

Degradation Because of Linear Motion

- Consider the case of camera/object planar motion.
- The coordinates change over time, that is $x_0(t)$ and $y_0(t)$.



Degradation Because of Linear Motion

- Let T be the exposure time, which is the time interval for which the camera shutter is open.

Degradation Because of Linear Motion

- Let T be the exposure time, which is the time interval for which the camera shutter is open.
- The acquired image $g(x,y)$ is

$$g(x,y) = \int_0^T f(x - x_0(t), y - y_0(t))dt.$$

Degradation Because of Linear Motion

- Let T be the exposure time, which is the time interval for which the camera shutter is open.
- The acquired image $g(x, y)$ is

$$g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t))dt.$$

- After applying Fourier transform and using the Convolution theorem $G(u, v) = F(u, v)H(u, v)$ we can show that

$$H(u, v) = \int_0^T e^{-j2\pi(ux_0(t)+vy_0(t))}dt.$$

Degradation Because of Linear Motion

- When $x_0(t) = at/T$ and $y_0(t) = 0$

$$
\begin{aligned}
H(u,v) &= \int_0^T e^{-j2\pi(ux_0(t)+vy_0(t))dt} \\
&= \int_0^T e^{-j2\pi(uat/T)}dt \\
&= \frac{T}{\pi ua}\sin(\pi ua)e^{-j2\pi(ua)}.
\end{aligned}
$$

Degradation Because of Linear Motion

- When $x_0(t) = at/T$ and $y_0(t) = bt/T$

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin(\pi(ua+vb)) e^{-j2\pi(ua+vb)}.$$

Degradation Because of Linear Motion Example

Figure: Degradation because of linear motion with $a = b = 0.1$ and $T = 1$.

Inverse Filtering

- The degradation model is

$$G(u, v) = H(u, v)F(u, v) + N(u, v).$$

Inverse Filtering

- The degradation model is

$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$

- Straightforward solution is direct inverse filtering

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}.$$

Inverse Filtering

- The degradation model is

$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$

- Straightforward solution is direct inverse filtering

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}.$$

- This is applied as an array operation.

Inverse Filtering - Practical Issues

- Continuing the previous analysis we have

$$\frac{G(u,v)}{H(u,v)} = \frac{H(u,v)F(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)}$$
$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}.$$

Inverse Filtering - Practical Issues

- Continuing the previous analysis we have

$$\frac{G(u,v)}{H(u,v)} = \frac{H(u,v)F(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)}$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}.$$

- This approach has two problems:

Inverse Filtering - Practical Issues

- Continuing the previous analysis we have

$$\frac{G(u,v)}{H(u,v)} = \frac{H(u,v)F(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)}$$
$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}.$$

- This approach has two problems:
  - We cannot restore the image completely when we don't know $N(u,v)$.

Inverse Filtering - Practical Issues

- Continuing the previous analysis we have

$$\frac{G(u,v)}{H(u,v)} = \frac{H(u,v)F(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)}$$
$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}.$$

- This approach has two problems:
  - We cannot restore the image completely when we don't know $N(u,v)$.
  - For small values of $H(u,v)$ the noise term may dominate the estimate adding large error.

Inverse Filtering - Practical Issues

- Continuing the previous analysis we have

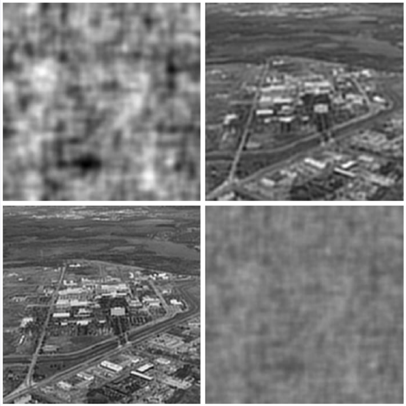
$$\frac{G(u,v)}{H(u,v)} = \frac{H(u,v)F(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)}$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}.$$

- This approach has two problems:
  - We cannot restore the image completely when we don't know $N(u,v)$.
  - For small values of $H(u,v)$ the noise term may dominate the estimate adding large error.
  - One solution for this is to restrict analysis within a radius from the DC frequency to ensure that $H(u,v)$ will have large enough magnitude.

Inverse Filtering Example

- Result of inverse filtering using full frequency range, and cut-off radius of 40, 70, and 85 (left to right and top to bottom).
- Observe that the using the full range or an excessive radius will amplify the noise term and produce unacceptable restoration.



Wiener Filtering

- In contrast to inverse filtering, Wiener utilizes statistical properties of the noise.

Wiener Filtering

- In contrast to inverse filtering, Wiener utilizes statistical properties of the noise.
- This method considers image and noise to be random variables and finds the estimate $\hat{f}$ that minimizes the mean squared error

$$e^2 = E\{(f - \hat{f})^2\}.$$

Wiener Filtering

- In contrast to inverse filtering, Wiener utilizes statistical properties of the noise.
- This method considers image and noise to be random variables and finds the estimate $\hat{f}$ that minimizes the mean squared error

$$e^2 = E\{(f - \hat{f})^2\}.$$

- Working hypotheses:
  - image and noise are uncorrelated
  - either one has zero mean
  - intensity levels in $\hat{f}$ are a linear function of intensity levels of f.

Wiener Filtering

- Wiener showed that the error function minimum in frequency domain is

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}}.$$

where
$H(u,v)$: degradation transfer function
$G(u,v)$: transformed degraded image
$S_n(u,v) = |N(u,v)|^2$: power spectrum of the noise
$S_f(u,v) = |F(u,v)|^2$: power spectrum of the ideal image.

Wiener Filtering

- In general, we do not know the power spectra of the ideal image or the noise.

Wiener Filtering

- In general, we do not know the power spectra of the ideal image or the noise.
- Then we approximate $\hat{F}(u,v)$ as

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}.$$

where
K: constant specified by user
$H(u,v)$: degradation transfer function
$G(u,v)$: transformed degraded image
$S_n(u,v) = |N(u,v)|^2$: power spectrum of the noise
$S_f(u,v) = |F(u,v)|^2$: power spectrum of the ideal image.

Error Measures from Power Spectra

- Signal to Noise Ratio, *SNR*

$$SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u,v)|^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |N(u,v)|^2}.$$

Error Measures from Power Spectra

- Signal to Noise Ratio, *SNR*

$$SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u,v)|^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |N(u,v)|^2}.$$

- Mean Square Error, *MSE*

$$MSE = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} [f(x,y) - \hat{f}(x,y)]^2.$$

Error Measures from Power Spectra

- Signal to Noise Ratio, *SNR*

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|^2}.$$

- Mean Square Error, *MSE*

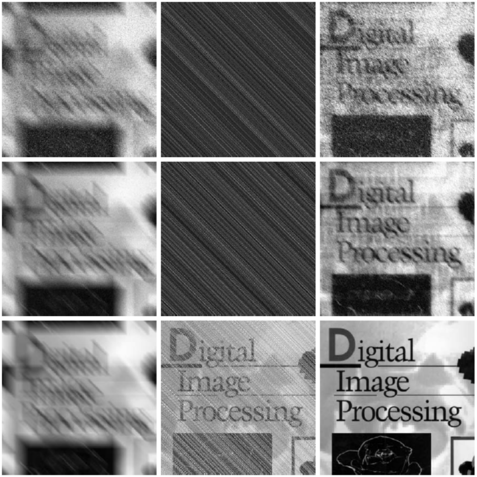
$$MSE = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}[f(x,y)-\hat{f}(x,y)]^2.$$

- Signal to Noise Ratio (spatial domain), *SNR*

$$SNR = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f(x,y)-f(\hat{x},y))^2}.$$

Wiener Filtering Example

- Left column: motion blur and additive noise corruption.
- Middle column: inverse filtering output.
- Right column: Wiener filtering output.



Constrained Least Squares Filtering Motivation

- Wiener filter
  - Requires knowledge of noise and signal power spectra.
  - Optimal averaged images.

Constrained Least Squares Filtering Motivation

- Wiener filter
  - Requires knowledge of noise and signal power spectra.
  - Optimal averaged images.
- CLS

Constrained Least Squares Filtering Motivation

- Wiener filter
  - Requires knowledge of noise and signal power spectra.
  - Optimal averaged images.
- CLS
  - Requires knowing only the mean and standard deviation of the noise.

Constrained Least Squares Filtering Motivation

- Wiener filter
  - Requires knowledge of noise and signal power spectra.
  - Optimal averaged images.
- CLS
  - Requires knowing only the mean and standard deviation of the noise.
  - Optimal for all images.

Constrained Least Squares Filtering

- Our initial assumption was that an image $f(x,y)$ undergoes a degradation process modeled by function $h(x,y)$ followed by corruption by additive noise $n(x,y)$.

$$g(x,y) = h(x,y) * f(x,y) + n(x,y).$$

Constrained Least Squares Filtering

- Our initial assumption was that an image $f(x,y)$ undergoes a degradation process modeled by function $h(x,y)$ followed by corruption by additive noise $n(x,y)$.

$$g(x,y) = h(x,y) * f(x,y) + n(x,y).$$

- By applying convolution in spatial domain we get

$$g_e(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m,n) h_e(x-m, y-n) + n_e(x,y)$$

$x = 0, 1, \cdots, M-1$
$y = 0, 1, \cdots, N-1$

Expressing Degradation using Matrix Equations

- If we consider the 1-D case the extended convolution can be expressed by

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$$

where
$\mathbf{g} : M \times 1$, $\mathbf{H} : M \times M$, $\mathbf{f} : M \times 1$, $\mathbf{n} : M \times 1$.

Expressing Degradation using Matrix Equations

- In the 2-D case the extended convolution can be expressed by

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$$

where
$\mathbf{g} : MN \times 1$, $\mathbf{H} : MN \times MN$, $\mathbf{f} : MN \times 1$, $\mathbf{n} : MN \times 1$.

Expressing Degradation using Matrix Equations

- In the 2-D case the extended convolution can be expressed by

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$$

  where
  $\mathbf{g} : MN \times 1$, $\mathbf{H} : MN \times MN$, $\mathbf{f} : MN \times 1$, $\mathbf{n} : MN \times 1$.
- Question: can we use matrix operations to estimate $\mathbf{f}$?

Expressing Degradation using Matrix Equations

- Question: can we use matrix operations to estimate $\mathbf{f}$?

Expressing Degradation using Matrix Equations

- Question: can we use matrix operations to estimate $\mathbf{f}$?
- Answer: this is not simple because of

Expressing Degradation using Matrix Equations

- Question: can we use matrix operations to estimate $\mathbf{f}$?
- Answer: this is not simple because of
    - very high dimensional vectors

Expressing Degradation using Matrix Equations

- Question: can we use matrix operations to estimate $\mathbf{f}$?
- Answer: this is not simple because of
  - very high dimensional vectors
  - need to invert $\mathbf{H}$

Expressing Degradation using Matrix Equations

- Question: can we use matrix operations to estimate $\mathbf{f}$?
- Answer: this is not simple because of
  - very high dimensional vectors
  - need to invert $\mathbf{H}$
  - solution being sensitive to noise.

Constrained Least Squares Filtering Regularization

- We can deal with sensitivity to noise by applying smoothness constraint to our solution, that is by minimizing

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\nabla^2 f(x,y)]^2$$

subject to

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\mathbf{n}\|^2.$$

Constrained Least Squares Filtering Solution

- We can show that the solution is

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma |P(u,v)|^2}.$$

where
$P(u,v)$ is the Fourier transform of Laplacian kernel

$$p(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

γ is the only parameter, and we get inverse filtering for $\gamma = 0$.

Constrained Least Squares Filtering Solution

- We can show that the solution is

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma |P(u,v)|^2}.$$

  where
  $P(u,v)$ is the Fourier transform of Laplacian kernel

$$p(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

  γ is the only parameter, and we get inverse filtering for $\gamma = 0$.
- We also note that $||n||^2$ is a monotonic function of γ.

Constrained Least Squares Filtering Algorithm

1. Define degradation model

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}.$$

Constrained Least Squares Filtering Algorithm

1. Define degradation model

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}.$$

2. Calculate residual error

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}.$$

Constrained Least Squares Filtering Algorithm

1. Define degradation model

$$\mathbf{g} = \mathbf{Hf} + \mathbf{n}.$$

2. Calculate residual error

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}.$$

3. Adjust γ and recompute $\hat{f}$.

Constrained Least Squares Filtering Algorithm

1. Define degradation model

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}.$$

2. Calculate residual error

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}.$$

3. Adjust γ and recompute $\hat{f}$.
4. If

$$\|\mathbf{r}\|^2 - \|\mathbf{n}\|^2 \leq a$$

return,
otherwise go to step 2.

Constrained Least Squares Filtering Algorithm

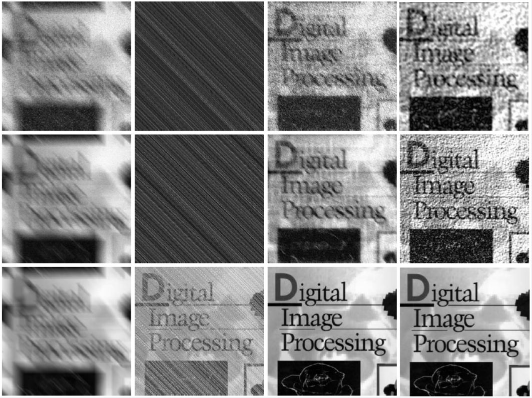
- In the previous algorithm we need to find $\mathbf{n}$.

Constrained Least Squares Filtering Algorithm

- In the previous algorithm we need to find **n**.
- We can show that $\mathbf{n} = MN[\mu_n^2 + \sigma_n^2]$. where μ_n is the noise mean, and σ_n is the noise standard deviation.

Constrained Least Squares Filtering Example

- First column: motion blur and additive noise corruption.
- Second column: inverse filtering output.
- Third column: Wiener filtering output.
- Fourth column: CLS filtering output.



Morphological Image Processing

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 19, 2015

Outline

1. Morphological Image Processing
   - Mathematical Morphology
   - Erosion and Dilation

Mathematical Morphology

- The term morphology refers to the field of Biology that studies the form and structure of animals and plants.

Mathematical Morphology

- The term morphology refers to the field of Biology that studies the form and structure of animals and plants.
- In Image Processing, the field of Mathematical Morphology or Morphology is useful for analyzing shapes.

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:
  - the set of all white pixels defines an object

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:
  - the set of all white pixels defines an object
  - pixels of the image are represented by ordered pairs of their coordinates $(x, y) \in \mathbb{Z}^2$ on the image plane.

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:
  - the set of all white pixels defines an object
  - pixels of the image are represented by ordered pairs of their coordinates $(x, y) \in \mathbb{Z}^2$ on the image plane.
- In grayscale images:

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:
  - the set of all white pixels defines an object
  - pixels of the image are represented by ordered pairs of their coordinates $(x, y) \in \mathbb{Z}^2$ on the image plane.
- In grayscale images:
  - the set of all white pixels defines an object

Mathematical Morphology

- Morphology uses set theory to represent objects in an image.
- In binary images:
  - the set of all white pixels defines an object
  - pixels of the image are represented by ordered pairs of their coordinates $(x, y) \in \mathbb{Z}^2$ on the image plane.
- In grayscale images:
  - the set of all white pixels defines an object
  - pixels of the image are represented by ordered 3-tuples of their coordinates on the image plan and the pixel intensities $(x, y, z) \in \mathbb{Z}^3$.

Reflection

- Let B be a set of pixels in a binary image and $w = (x, y)$ an element of B.

Reflection

- Let B be a set of pixels in a binary image and $w = (x, y)$ an element of B.
- The reflection of B denoted by $\hat{B}$ is defined as

$$\hat{B} = \{(-x, -y) | \forall (x, y) \in B\}.$$

Translation

- Let B be a set of pixels in a binary image and $w = (x, y)$ a coordinate point.

Translation

- Let B be a set of pixels in a binary image and $w = (x, y)$ a coordinate point.
- The translation of B denoted by $(B)_w$ is defined as

$$(B)_w = \{(a, b) + (x, y) | \forall (a, b) \in B\}.$$

Translation

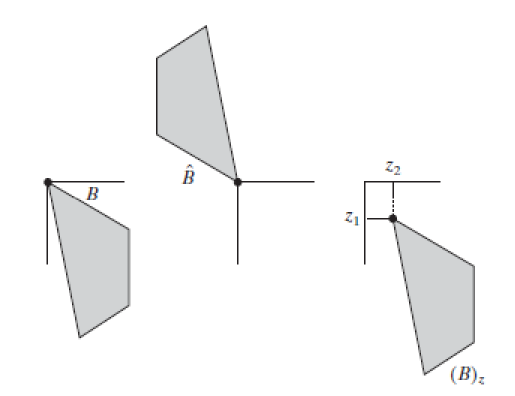
- Let B be a set of pixels in a binary image and $w = (x, y)$ a coordinate point.
- The translation of B denoted by $(B)_w$ is defined as

$$(B)_w = \{(a, b) + (x, y) | \forall (a, b) \in B\}.$$

- After translation, B has been shifted by w.

Reflection and Translation Example

- Original set of points (left), set after reflection (middle), set after translation (right).



The Structuring Element (SE)

- The structuring element also denoted by SE is a small set or subimage used to examine an image for specific properties of interest.

The Structuring Element (SE)

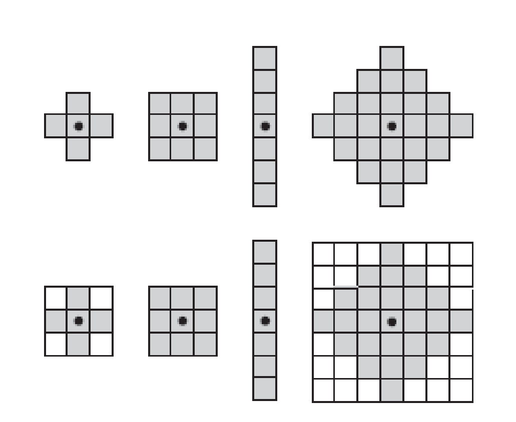
- The structuring element also denoted by SE is a small set or subimage used to examine an image for specific properties of interest.
- SEs are basic shapes such as line, cross, diamond, etc.

The Structuring Element (SE)

- The structuring element also denoted by SE is a small set or subimage used to examine an image for specific properties of interest.
- SEs are basic shapes such as line, cross, diamond, etc.
- We usually need to define the origin of an SE for morphological operations.

Structuring Element Examples

- Examples of structuring elements. The centers are denoted by the dots.



Erosion

- Let A and B be sets of pixels in $\mathbb{Z}^2$.

Erosion

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the erosion of A by B written $A \ominus B$ is defined as

$$A \ominus B = \{w|(B)_w \subseteq A\}.$$

Erosion

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the erosion of A by B written $A \ominus B$ is defined as

$$A \ominus B = \{w | (B)_w \subseteq A\}.$$

- To perform erosion we can move B over A and find all the locations it will fit. The set of all such locations forms $A \ominus B$.

Equivalent Erosion Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.

Equivalent Erosion Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the erosion of A by B written $A \ominus B$ is defined as

$$A \ominus B = \{w | (B)_w \cap A^c = \emptyset\}.$$

Equivalent Erosion Definition

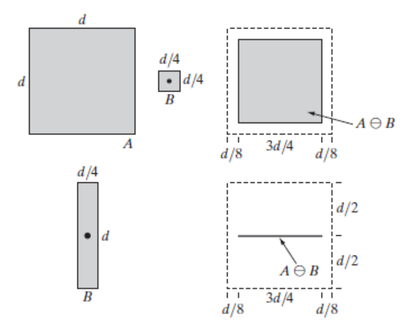
- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the erosion of A by B written $A \ominus B$ is defined as

$$A \ominus B = \{w | (B)_w \cap A^c = \emptyset\}.$$

- To perform erosion we can move B over A and find all the locations it will fit. The set of all such locations forms $A \ominus B$.

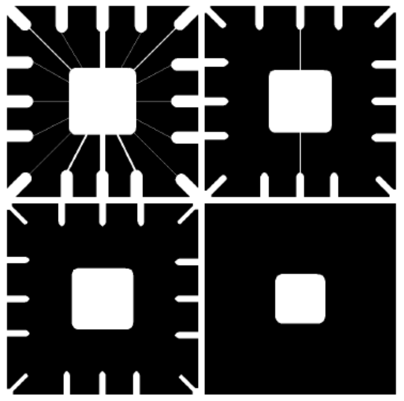
The Erosion Process

- Top row - A set A and a structuring element B (left). The result of erosion of A by B (right).
- Bottom row - A line SE and the result of erosion of A by the line SE.
- Erosion can be used for object shrinking or thinning.



Erosion Example

- Original 486×486 image, erosion output using 11×11 square SE, erosion output using 15×15 square SE, erosion output using 45×45 square SE (top to bottom and left to right).



Dilation

- Let A and B be sets of pixels in $\mathbb{Z}^2$.

Dilation

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the dilation of A by B written $A \oplus B$ is defined as

$$A \oplus B = \{(x, y) + (u, v) | \forall (x, y) \in A, \forall (u, v) \in B\}.$$

Dilation

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the dilation of A by B written $A \oplus B$ is defined as

$$A \oplus B = \{(x,y)+(u,v) | \forall (x,y) \in A, \forall (u,v) \in B\}.$$

- To perform dilation we can move B over A and replace every point $(x,y) \in A$ with a copy of B centered at (x,y). The set of all such locations forms $A \oplus B$.

Equivalent Dilation Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.

Equivalent Dilation Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the dilation of A by B written $A \oplus B$ is defined as

$$A \oplus B = \{w | (\hat{B})_w \cap A \neq \emptyset\}.$$

or

$$A \oplus B = \{w | [(\hat{B})_w \cap A] \subseteq A\}.$$

Equivalent Dilation Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the dilation of A by B written $A \oplus B$ is defined as

$$A \oplus B = \{w | (\hat{B})_w \cap A \neq \emptyset\}.$$

or

$$A \oplus B = \{w | [(\hat{B})_w \cap A] \subseteq A\}.$$

- The dilation of a A by B is the set of all translations w, such that $\hat{B}$ and A overlap by at least one element.

Equivalent Dilation Definition

- Let A and B be sets of pixels in $\mathbb{Z}^2$.
- Then the dilation of A by B written $A \oplus B$ is defined as

$$A \oplus B = \{w | (\hat{B})_w \cap A \neq \emptyset\}.$$

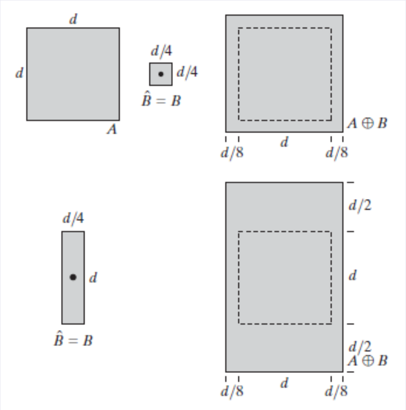
or

$$A \oplus B = \{w | [(\hat{B})_w \cap A] \subseteq A\}.$$

- The dilation of a A by B is the set of all translations w, such that $\hat{B}$ and A overlap by at least one element.
- To perform dilation we can move B over A and replace every point $(x, y) \in A$ with a copy of B centered at (x, y). The set of all such locations forms $A \oplus B$.

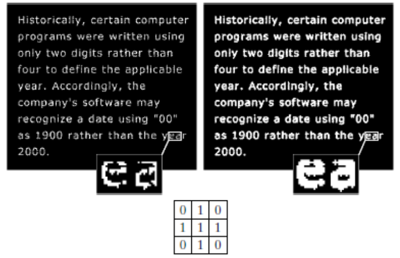
The Dilation Process

- Top row - A set A and a structuring element B (left). The result of dilation of A by B (right).
- Bottom row - A line SE and the result of erosion of A by the line SE.
- Dilation can be used for object growing or thickening.



Dilation Example

- Original image (left) and the output of dilation of the image by a diamond SE. We observe that the gaps in characters have been bridged and the dilated image is more suitable for optical character recognition.



Duality

- Erosion and dilation are duals of each other with respect to set complement and reflection:

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B}.$$

Duality

- Erosion and dilation are duals of each other with respect to set complement and reflection:

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B}.$$

- This is particularly useful when the structuring element is symmetric with respect to its origin.

Morphological Image Processing

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 21, 2015

Outline and Learning Goals

Previous class:

- Introduction to mathematical morphology
- Definitions of erosion and dilation morphological operations
- Applications of erosion and dilation

Today:

1. Morphological Image Processing
   - Opening and Closing
   - Hit-or-Miss Transformation
   - Other Morphological Algorithms

Opening and Closing

- Opening and closing are another two basic morphological operations that are built upon the erosion and dilation operations.

Opening and Closing

- Opening and closing are another two basic morphological operations that are built upon the erosion and dilation operations.
- Opening widens gaps and breaks narrow bridges between groups of pixels.

Opening and Closing

- Opening and closing are another two basic morphological operations that are built upon the erosion and dilation operations.
- Opening widens gaps and breaks narrow bridges between groups of pixels.
- Closing eliminates small holes and fills gaps.

Opening

- Opening of a set A by a structuring element B is symbolized by $A \circ B$ and given by

$$A \circ B = (A \ominus B) \oplus B.$$

Opening

- Opening of a set A by a structuring element B is symbolized by $A \circ B$ and given by

$$A \circ B = (A \ominus B) \oplus B.$$

- So A is first eroded by B and the result is dilated by B.

Geometric Interpretation of Opening

- The opening of A by B is obtained by the union of all translations of B that fit into A.

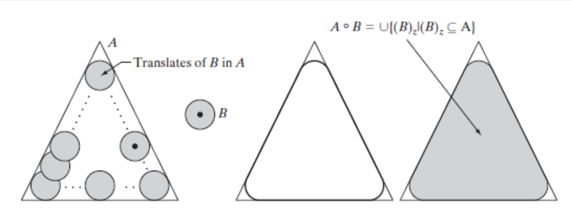
Geometric Interpretation of Opening

- The opening of A by B is obtained by the union of all translations of B that fit into A.
- So we can express opening as

$$A \circ B = \cup_z \{(B)_z | (B)_z \subseteq A\}.$$

Opening Example

- The morphological opening process can be explained as a structuring element B rolling along inner boundary of A.



Closing

- Closing of a set A by a structuring element B is symbolized by $A \bullet B$ and given by

$$A \bullet B = (A \oplus B) \ominus B.$$

Closing

- Closing of a set A by a structuring element B is symbolized by $A \bullet B$ and given by

$$A \bullet B = (A \oplus B) \ominus B.$$

- So A is first dilated by B and the result is eroded by B.

Geometric Interpretation of Closing

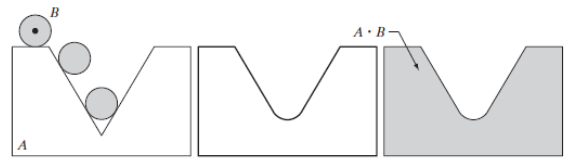
- Closing of a set A by a structuring element B can be resembled by rolling B on the outside of A's boundary and tracking the points that are reached by B.

Geometric Interpretation of Closing

- Closing of a set A by a structuring element B can be resembled by rolling B on the outside of A's boundary and tracking the points that are reached by B.
- A point w is an element of $A \bullet B$, iff $(B)_z \cap A \neq \emptyset$ for any translation of $(B)_z$ that contains w.

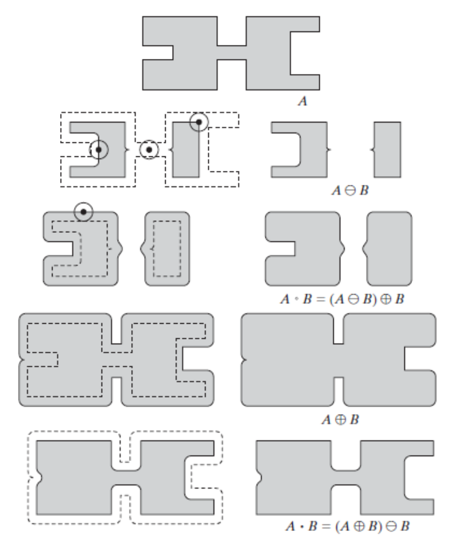
Closing Example

- The morphological closing process can be explained as a structuring element B rolling along outer boundary of A.



Opening and Closing Operations

- Morphological opening and closing of A by a circular structuring element.



Opening and Closing Example

- Morphological opening followed by closing for image enhancement.

Opening and Closing Duality

- The duality between opening and closing can be expressed by

$$(A \bullet B)^c = (A^c \circ \hat{B})$$

and

$$(A \circ B)^c = (A^c \bullet \hat{B})$$

Opening Properties

Opening satisfies the following properties

1. $A \circ B$ is a subset (subimage) of A.

Opening Properties

Opening satisfies the following properties

1. $A \circ B$ is a subset (subimage) of A.
2. If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.

Opening Properties

Opening satisfies the following properties

1. $A \circ B$ is a subset (subimage) of A.
2. If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.
3. $(A \circ B) \circ B = A \circ B$. After the first opening, any additional opening operations have no additional effect.

Closing Properties

Closing satisfies the following properties

1. A is a subset (subimage) of $A \bullet B$.

Closing Properties

Closing satisfies the following properties

1. A is a subset (subimage) of $A \bullet B$.
2. If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.

Closing Properties

Closing satisfies the following properties

1. A is a subset (subimage) of $A \bullet B$.
2. If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.
3. $(A \bullet B) \bullet B = A \bullet B$. After the first closing, any additional opening operations have no additional effect.

Hit-or-Miss Transformation

- This method is used to detect shapes.

Hit-or-Miss Transformation

- This method is used to detect shapes.
- Let A be a set consisting of sets C, D, and E, such that $A = C \cup D \cup E$.

Hit-or-Miss Transformation

- This method is used to detect shapes.
- Let A be a set consisting of sets C, D, and E, such that $A = C \cup D \cup E$.
- Let B denote the set containing D and its background W, that is $B = D \cup W$.

Hit-or-Miss Transformation

- This method is used to detect shapes.
- Let A be a set consisting of sets C, D, and E, such that $A = C \cup D \cup E$.
- Let B denote the set containing D and its background W, that is $B = D \cup W$.
- Assume that we want to find the location of D.

Hit-or-Miss Transformation

- This method is used to detect shapes.
- Let A be a set consisting of sets C, D, and E, such that $A = C \cup D \cup E$.
- Let B denote the set containing D and its background W, that is $B = D \cup W$.
- Assume that we want to find the location of D.
- The location of D is given by the intersection of the erosion of A by D with the erosion of A^c by $W - D$. This is expressed as

$$A \circledast B = (A \ominus D) \cap (A^c \ominus (W - D)).$$

Hit-or-Miss Transformation

- If we let $B_1 = D$ and $B_2 = W - D$, it follows that

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2).$$

Hit-or-Miss Transformation

- If we let $B_1 = D$ and $B_2 = W - D$, it follows that

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2).$$

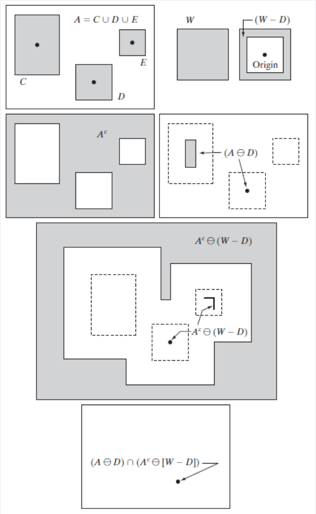
- By utilizing duality between erosion and dilation we get

$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2).$$

Hit-or-Miss Transformation Process

Find the location of D using the hit-or-miss process.

1. Erode A by D.
2. Erode A_c by $W - D$.
3. Find intersection of previous two.



Boundary Extraction

- Boundary extraction is used very frequently in image processing and analysis.

Boundary Extraction

- Boundary extraction is used very frequently in image processing and analysis.
- The boundary of a set A denoted by $\beta(A)$, is obtained by erosion by an SE element B followed by the difference between the original image and the eroded image.

Boundary Extraction

- Boundary extraction is used very frequently in image processing and analysis.
- The boundary of a set A denoted by $\beta(A)$, is obtained by erosion by an SE element B followed by the difference between the original image and the eroded image.
- This is expressed by

$$\beta(A) = A - (A \ominus B).$$

Boundary Extraction

- Boundary extraction is used very frequently in image processing and analysis.
- The boundary of a set A denoted by $\beta(A)$, is obtained by erosion by an SE element B followed by the difference between the original image and the eroded image.
- This is expressed by

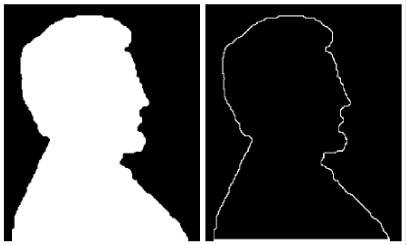
$$\beta(A) = A - (A \ominus B).$$

- The boundary thickness depends on the structuring element.

Boundary Extraction Example

Find the boundary of D.

1. Erode A by 3×3 square SE.
2. Subtract eroded image from original.



Connected Component Extraction

- Often times we want to find the connected components of a binary image.

Connected Component Extraction

- Often times we want to find the connected components of a binary image.
- Recall that a connected component is a set of pixels that are related by an adjacency type.

Connected Component Extraction

- Often times we want to find the connected components of a binary image.
- Recall that a connected component is a set of pixels that are related by an adjacency type.
- We can use morphological operations to find the connected components of a binary image.

Connected Component Extraction

- Let A be a set containing our object coordinates with one or more connected components, and B be a structuring element.

Connected Component Extraction

- Let A be a set containing our object coordinates with one or more connected components, and B be a structuring element.
- Let X_0 be a set that contains the connected components.

Connected Component Extraction

- Let A be a set containing our object coordinates with one or more connected components, and B be a structuring element.
- Let X_0 be a set that contains the connected components.
- Starting from X_0 we can find the connected component by iteratively applying

$$X_k = (X_{k-1} \oplus B) \cap A, k = 1, 2, 3, \ldots$$

Connected Component Extraction

- Let A be a set containing our object coordinates with one or more connected components, and B be a structuring element.
- Let X_0 be a set that contains the connected components.
- Starting from X_0 we can find the connected component by iteratively applying

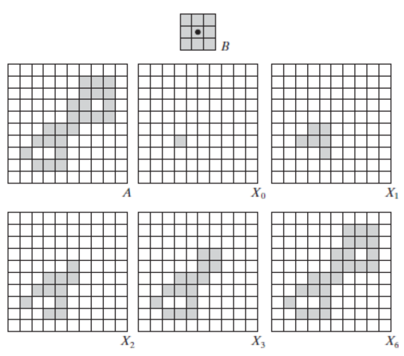
$$X_k = (X_{k-1} \oplus B) \cap A, k = 1, 2, 3, \ldots$$

- The process ends when $X_k = X_{k-1}$.

Connected Component Extraction Example

Find the connected component of A.

1. Initialize connected component subimage with one point.
2. Repeat:
   1. Dilate connected component.
   2. Find intersection of previous result with A.
3. Until connected component does not change.



Active Research Areas

- ISMM: International Symposium on Mathematical Morphology.
- Scale-space theory.
- 3D image analysis.

Image Segmentation

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 23, 2015

Outline

Previous chapter:

- Introduction to mathematical morphology
- Definitions of erosion, dilation, opening, closing morphological operators
- Morphological algorithms (hit-or-miss, boundary extraction, connected component labeling)

Today:

1. Image Segmentation
   - Image Segmentation: Background and Definitions
   - Point, Line and Edge Detection
   - Segmentation by Thresholding

Image Segmentation Introduction

- Image segmentation is a key step in image analysis and computer vision.

Image Segmentation Introduction

- Image segmentation is a key step in image analysis and computer vision.
- Segmentation divides the visual scene into smaller regions and objects.

Image Segmentation Introduction

- Image segmentation is a key step in image analysis and computer vision.
- Segmentation divides the visual scene into smaller regions and objects.
- The level of detail is driven by the application.

Image Segmentation Introduction

- Image segmentation is a key step in image analysis and computer vision.
- Segmentation divides the visual scene into smaller regions and objects.
- The level of detail is driven by the application.
- Image segmentation is a difficult task because of imaging artifacts, limits in imaging spatial and intensity resolution, and difficulty in defining the perfect segmentation.

Image Segmentation Introduction

- Sometimes we use prior knowledge to increase segmentation accuracy. One example is when the environment is controlled as in automated inspection applications (e.g., quality control of electronic components).

Image Segmentation Introduction

- Sometimes we use prior knowledge to increase segmentation accuracy. One example is when the environment is controlled as in automated inspection applications (e.g., quality control of electronic components).
- When the environment is not controlled, we select sensors that will reveal the image attributes we are looking for. For example, in remote sensing we may use multi-channel imaging sensors to identify crops, rivers, buildings and roads.

Image Segmentation Introduction

- Most of the segmentation algorithms make use of the fundamental properties of discontinuity and similarity.

Image Segmentation Introduction

- Most of the segmentation algorithms make use of the fundamental properties of discontinuity and similarity.
- Methods that detect discontinuities are line and edge detection and linking techniques.

Image Segmentation Introduction

- Most of the segmentation algorithms make use of the fundamental properties of discontinuity and similarity.
- Methods that detect discontinuities are line and edge detection and linking techniques.
- Similarity-based methods divide the image into homogeneous regions. Such methods are thresholding, region growing, region splitting and merging techniques.

Image Segmentation Introduction

- Most of the segmentation algorithms make use of the fundamental properties of discontinuity and similarity.
- Methods that detect discontinuities are line and edge detection and linking techniques.
- Similarity-based methods divide the image into homogeneous regions. Such methods are thresholding, region growing, region splitting and merging techniques.
- Nowadays it is common to develop methods combine ideas from the discontinuity-based and region-based segmentation.

Image Segmentation Definition

Let R represent the spatial region of the image. Image segmentation can be defined as the process that partitions R into n subregions $R_1, R_2, ..., R_n$ such that

1. $\cup_{i=1}^{n} R_n = R$.

Image Segmentation Definition

Let R represent the spatial region of the image. Image segmentation can be defined as the process that partitions R into n subregions $R_1, R_2, \ldots, R_n$ such that

1. $\cup_{i=1}^{n} R_n = R$.
2. R_i is a connected set, for $i = 1, 2, \ldots, n$.

Image Segmentation Definition

Let R represent the spatial region of the image. Image segmentation can be defined as the process that partitions R into n subregions $R_1, R_2, \ldots, R_n$ such that

1. $\cup_{i=1}^{n} R_n = R$.
2. R_i is a connected set, for $i = 1, 2, \ldots, n$.
3. $R_i \cap R_j = \emptyset \; \forall i, j : i \neq j$.

Image Segmentation Definition

Let R represent the spatial region of the image. Image segmentation can be defined as the process that partitions R into n subregions $R_1, R_2, ..., R_n$ such that

1. $\cup_{i=1}^{n} R_n = R$.
2. R_i is a connected set, for $i = 1, 2, ..., n$.
3. $R_i \cap R_j = \emptyset \ \forall i, j : i \neq j$.
4. $Q(R_i) = TRUE$ for $i = 1, 2, ..., n$.

Image Segmentation Definition

Let R represent the spatial region of the image. Image segmentation can be defined as the process that partitions R into n subregions $R_1, R_2, ..., R_n$ such that

1. $\cup_{i=1}^{n} R_n = R$.
2. R_i is a connected set, for $i = 1, 2, ..., n$.
3. $R_i \cap R_j = \emptyset \ \forall i, j : i \neq j$.
4. $Q(R_i) = TRUE$ for $i = 1, 2, ..., n$.
5. $Q(R_i \cup R_j) = FALSE$ for any adjacent regions R_i and R_j with $i \neq j$,

   where $Q(R_k)$ is a logical predicate defined over the points in set R_k, for example $Q(R_i)$: all pixels in R_i have the same intensity.

Image Segmentation Definition

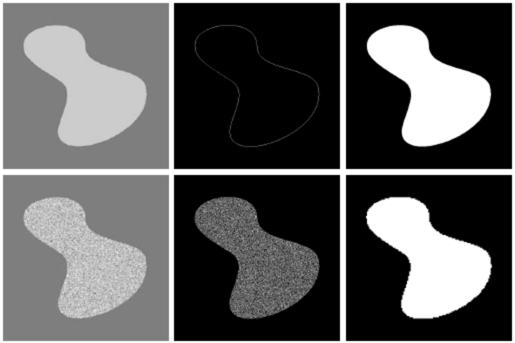
1. In edge-based segmentation we are looking for discontinuities between adjacent regions.

Image Segmentation Definition

1. In edge-based segmentation we are looking for discontinuities between adjacent regions.
2. In region-based segmentation we utilize intensity homogeneity criteria to form each region.

Image Segmentation Example

- Edge-based vs. region-based segmentation. Top: ideal case. Bottom: noisy case.



Background

- Here we detect sharp and local changes of intensity.
- Looking for three types of features: isolated points, lines and edges.
- To detect these features we first calculate first- and second-order derivatives.

After using Taylor series about x, keeping the linear terms and setting $\Delta x = 1$ we get the following approximations

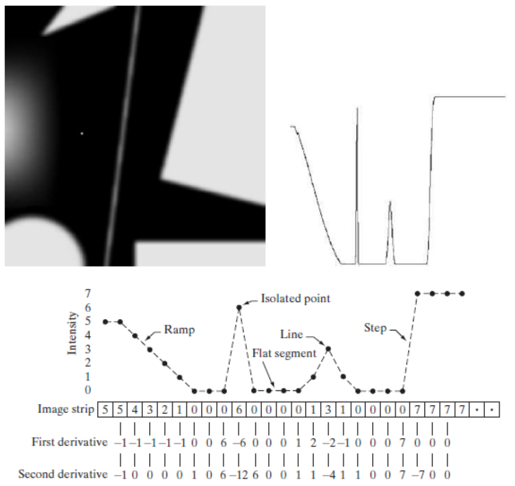
$$\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y), \quad \frac{\partial f}{\partial y} = f(x,y+1) - f(x,y),$$

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

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

First- and Second-order Derivatives of Intensity

- Let's review the intensity profile example, along with the first- and second-order derivatives.



Properties of First- and Second-order Derivatives

- First-order derivatives produce thicker edges

Properties of First- and Second-order Derivatives

- First-order derivatives produce thicker edges
- Second-order derivatives have a stronger response to isolated lines, points and noise.

Properties of First- and Second-order Derivatives

- First-order derivatives produce thicker edges
- Second-order derivatives have a stronger response to isolated lines, points and noise.
- Second-order derivatives produce a double-edge response at ramps and steps.

Properties of First- and Second-order Derivatives

- First-order derivatives produce thicker edges
- Second-order derivatives have a stronger response to isolated lines, points and noise.
- Second-order derivatives produce a double-edge response at ramps and steps.
- The sign of second-order derivative can show the intensity transition.

Point Detection

- Because the second-order derivative can detect thin details, it is suitable for point detection.

Point Detection

- Because the second-order derivative can detect thin details, it is suitable for point detection.
- We can apply a Laplacian filter followed by thresholding to detect points.

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

The discrete approximation is

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

Point Detection

- Because the second-order derivative can detect thin details, it is suitable for point detection.
- We can apply a Laplacian filter followed by thresholding to detect points.

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

The discrete approximation is

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

- We apply the Laplacian detector by spatial filtering, then apply threshold T to the filter's output $L(x,y)$ to detect points:

$$g(x,y) = \begin{cases} 1 & \text{if } |L(x,y)| \geq T \\ 0 & \text{otherwise.} \end{cases}$$

Line Detection

- Because second-order derivatives produce a stronger response and produce thinner lines than first-order derivatives, the former can be used for line detection.

Line Detection

- Because second-order derivatives produce a stronger response and produce thinner lines than first-order derivatives, the former can be used for line detection.
- We can use the positive-only values of the Laplacian to address double-edges.

Line Detection

- Because second-order derivatives produce a stronger response and produce thinner lines than first-order derivatives, the former can be used for line detection.
- We can use the positive-only values of the Laplacian to address double-edges.
- To suppress peaks caused by noise we can apply thresholding to the Laplacian output.

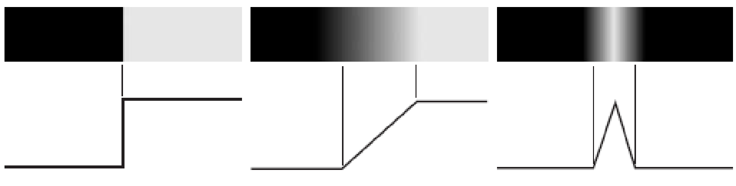
Line Detection

- Because second-order derivatives produce a stronger response and produce thinner lines than first-order derivatives, the former can be used for line detection.
- We can use the positive-only values of the Laplacian to address double-edges.
- To suppress peaks caused by noise we can apply thresholding to the Laplacian output.
- To detect specific line directions we can use masks with different orientations.

−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
Horizontal			+45°			Vertical			−45°		

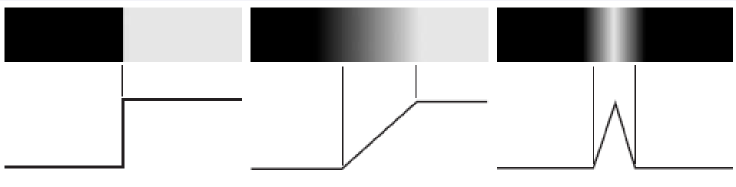
Edge Models

- Step edge (left): this is a transition between two intensity levels that is completed in 1 pixel. This is an ideal edge found in computer-generated images.



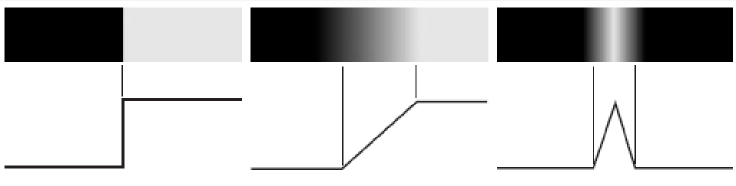
Edge Models

- Step edge (left): this is a transition between two intensity levels that is completed in 1 pixel. This is an ideal edge found in computer-generated images.
- Ramp edge (middle): this model is better approximation for blurred and noisy edges that appear in practice. The slope is reciprocal to the amount of blurring.



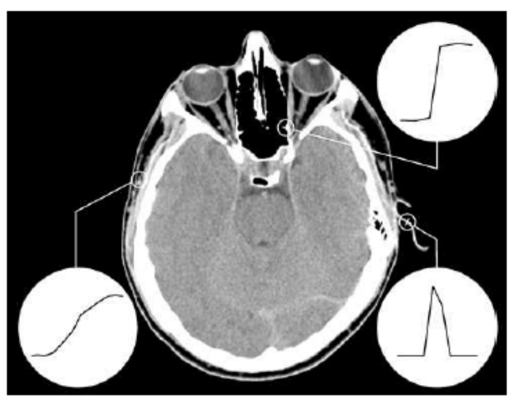
Edge Models

- Step edge (left): this is a transition between two intensity levels that is completed in 1 pixel. This is an ideal edge found in computer-generated images.
- Ramp edge (middle): this model is better approximation for blurred and noisy edges that appear in practice. The slope is reciprocal to the amount of blurring.
- Roof edge (right): these are models of lines through regions. These edges appear in digitized line drawings.



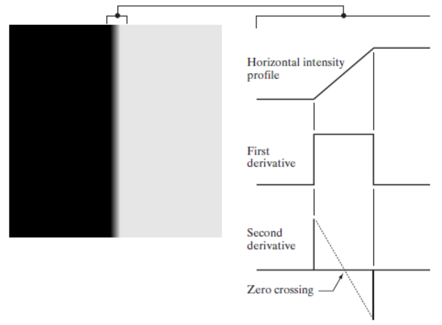
Edge Models Example

- All edge types can occur in the same image.



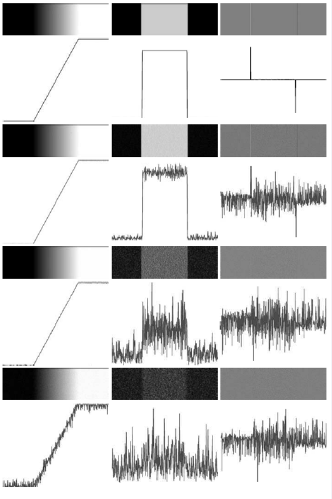
Edge Modeling Conclusions

- The first-order magnitude can be used to detect the presence of an edge.
- The sign of the second-order derivative can be used to localize the side of an edge.
- The zero-crossing of second-order derivative can localize the center of a thick edge.



Effect of Noise on Edges

- Noise can severely affect edge detection.
- This happens because both edges and noise correspond to high frequency content.



Edge Detection Steps

1. Image smoothing for noise reduction.

Edge Detection Steps

1. Image smoothing for noise reduction.
2. Detection of edge points - extract all candidates for edge points.

Edge Detection Steps

1. Image smoothing for noise reduction.
2. Detection of edge points - extract all candidates for edge points.
3. Edge localization - select from candidates the edge points.

Edge Detection using the Image Gradient

- The gradient is defined as

$$\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Edge Detection using the Image Gradient

- The gradient is defined as

$$\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

- The magnitude is given by

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

- The angle is given by

$$\alpha(x,y) = \arctan\left[\frac{g_y}{g_x}\right].$$

Gradient Operators

- We know that partial derivatives are approximated by

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

and

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

- To take diagonal differences we use Roberts operators.
- Prewitt operators compute the derivatives using 3×3 masks.
- Sobel operators use a weight of 2 at the center location.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0		0	−1
0	1		1	0

Roberts

−1	−1	−1		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

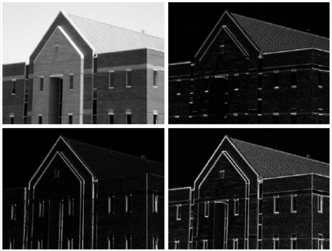
Prewitt

−1	−2	−1		−1	0	1
0	0	0		−2	0	2
1	2	1		−1	0	1

Sobel

Gradient-based Edge Detection Example

- Image smoothing using 5×5 averaging.
- Output produced by Sobel horizontal and vertical edge detectors
- Gradient magnitude is finally computed.



Thresholding

- Thresholding is a simple and fast operation that finds frequent use in image processing.

Thresholding

- Thresholding is a simple and fast operation that finds frequent use in image processing.
- Here we use thresholding to partition images into regions based on intensity values.

Thresholding Basics

- Let $f(x, y)$ the intensity function of an image, and let the image consist of light objects on a dark background. Then the intensity histogram of the image will have two main modes separated by a valley at intensity T.

Thresholding Basics

- Let $f(x,y)$ the intensity function of an image, and let the image consist of light objects on a dark background. Then the intensity histogram of the image will have two main modes separated by a valley at intensity T.
- We can segment the image by the following operation

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T. \end{cases}$$

Thresholding Basics

- Let $f(x, y)$ the intensity function of an image, and let the image consist of light objects on a dark background. Then the intensity histogram of the image will have two main modes separated by a valley at intensity T.
- We can segment the image by the following operation

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T. \end{cases}$$

- T is fixed: global thresholding.

Thresholding Basics

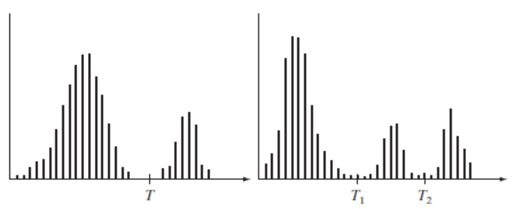
- Let $f(x, y)$ the intensity function of an image, and let the image consist of light objects on a dark background. Then the intensity histogram of the image will have two main modes separated by a valley at intensity T.
- We can segment the image by the following operation

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T. \end{cases}$$

- T is fixed: global thresholding.
- $T = T(x, y)$: variable thresholding.

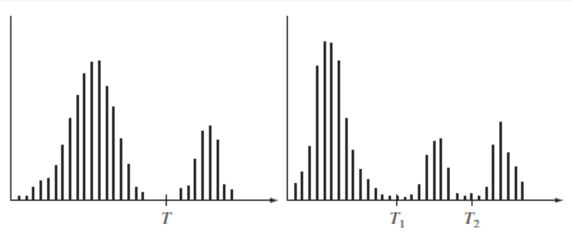
Thresholding Example

- Intensity thresholding can be visualized using the intensity histogram of the image pixels.



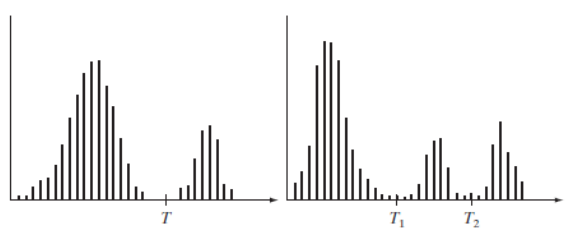
Thresholding Key Factors

- Separation between histogram peaks.



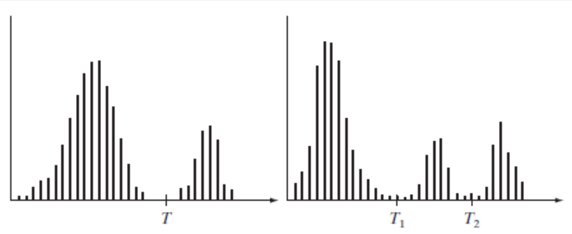
Thresholding Key Factors

- Separation between histogram peaks.
- Noise level.



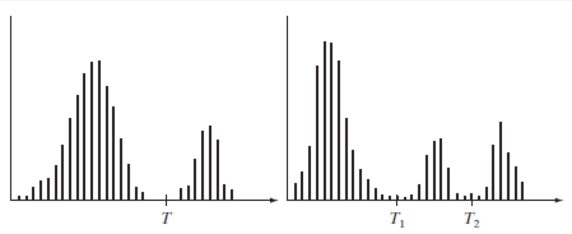
Thresholding Key Factors

- Separation between histogram peaks.
- Noise level.
- Relative sizes of objects and background.



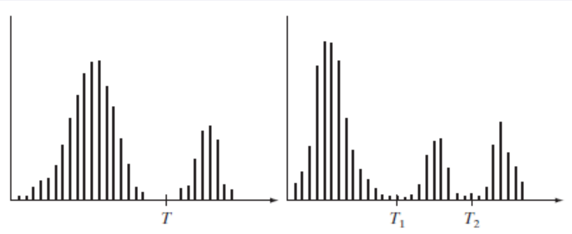
Thresholding Key Factors

- Separation between histogram peaks.
- Noise level.
- Relative sizes of objects and background.
- The uniformity of illumination source.



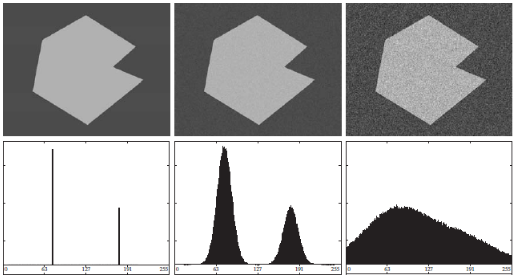
Thresholding Key Factors

- Separation between histogram peaks.
- Noise level.
- Relative sizes of objects and background.
- The uniformity of illumination source.
- The uniformity of reflectance of an object.



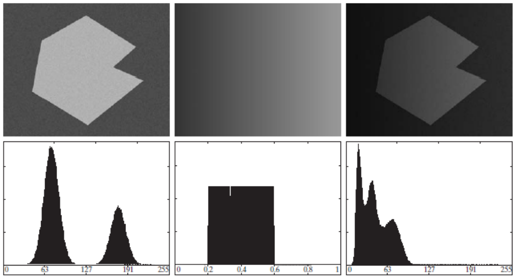
Effect of Noise on Thresholding

- Noise corruption can change histogram properties.



Effect of Illumination and Reflectance on Thresholding

- Changes in illuminant and/or reflectance can change the histogram properties.



Global Tresholding

1. Set initial threshold T.

Global Tresholding

1. Set initial threshold T.
2. Segment image into two regions G_1 and G_2, with $G_1(x, y) > T$ and $G_2(x, y) \leq T$.

Global Tresholding

1. Set initial threshold T.
2. Segment image into two regions G_1 and G_2, with $G_1(x, y) > T$ and $G_2(x, y) \leq T$.
3. Compute average intensities m_1 and m_2 over G_1 and G_2 respectively.

Global Tresholding

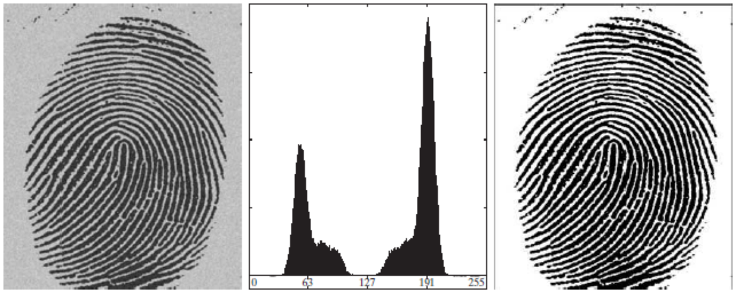
1. Set initial threshold T.
2. Segment image into two regions G_1 and G_2, with $G_1(x, y) > T$ and $G_2(x, y) \leq T$.
3. Compute average intensities m_1 and m_2 over G_1 and G_2 respectively.
4. Compute new threshold $T = 1/2(m_1 + m_2)$.

Global Tresholding

1. Set initial threshold T.
2. Segment image into two regions G_1 and G_2, with $G_1(x, y) > T$ and $G_2(x, y) \leq T$.
3. Compute average intensities m_1 and m_2 over G_1 and G_2 respectively.
4. Compute new threshold $T = 1/2(m_1 + m_2)$.
5. Repeat steps 2 to 4 until the difference between successive values of T becomes smaller than a fixed value ΔT.

Global Tresholding Example

- Fingerprint segmentation using thresholding.



Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.

Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.

Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.
3. Compute cumulative means $m(k), \forall k \in [0, L-1]$.

Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.
3. Compute cumulative means $m(k), \forall k \in [0, L-1]$.
4. Compute global intensity mean m_G and class-conditional means $m_1(k)$ and $m_2(k)$, $\forall k \in [0, L-1]$.

Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.
3. Compute cumulative means $m(k), \forall k \in [0, L-1]$.
4. Compute global intensity mean m_G and class-conditional means $m_1(k)$ and $m_2(k)$, $\forall k \in [0, L-1]$.
5. Compute between-class variance $\sigma_B^2(k), \forall k \in [0, L-1]$.

Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.
3. Compute cumulative means $m(k), \forall k \in [0, L-1]$.
4. Compute global intensity mean m_G and class-conditional means $m_1(k)$ and $m_2(k)$, $\forall k \in [0, L-1]$.
5. Compute between-class variance $\sigma_B^2(k), \forall k \in [0, L-1]$.
6. Find Otsu threshold $k*$ as the value of k for which $\sigma_B^2(k)$ is maximum. If various maxima are found, then average maxima arguments to find one value.

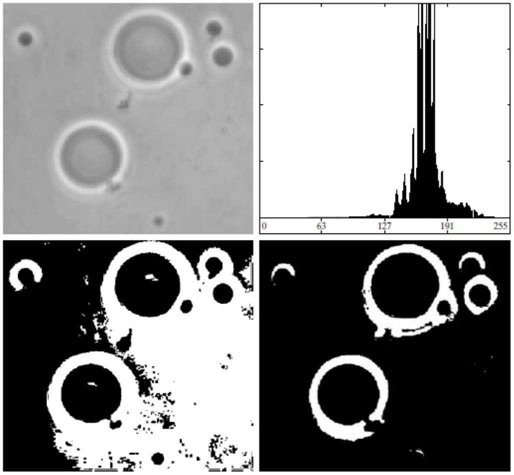
Otsu's Tresholding

This is a statistical method that seeks to maximize the between class variance. It is implemented as follows:

1. Compute normalized histogram of input image with components $p_i, \forall i \in [0, L-1]$.
2. Compute cumulative sums $P_1(k), \forall k \in [0, L-1]$.
3. Compute cumulative means $m(k), \forall k \in [0, L-1]$.
4. Compute global intensity mean m_G and class-conditional means $m_1(k)$ and $m_2(k)$, $\forall k \in [0, L-1]$.
5. Compute between-class variance $\sigma_B^2(k), \forall k \in [0, L-1]$.
6. Find Otsu threshold $k*$ as the value of k for which $\sigma_B^2(k)$ is maximum. If various maxima are found, then average maxima arguments to find one value.
7. Compute separability measure $n(k*) = \frac{\sigma_B^2(k)}{\sigma_G^2}$.

Effect of Noise Reduction on Thresholding

- Original image and its histogram.
- Segmentation produced by global thresholding (bottom left), and Otsu's thresholding (bottom right).



Otsu's Thresholding Example

- Image with Gaussian noise (left column).
- Segmentation produced without image denoising (top right).
- Segmentation produced after image denoising by a 5×5 averaging mask (bottom right).

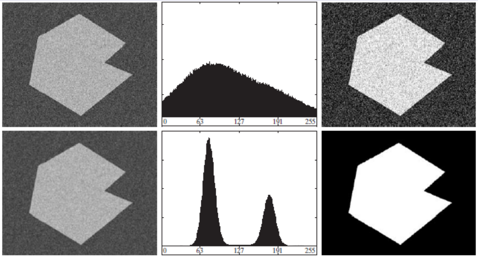


Image Segmentation

MTSC 887, Spring 2015

Sokratis Makrogiannis, Ph.D.

Department of Mathematical Sciences
Delaware State University

smakrogiannis@desu.edu

April 21, 2015

Outline

Region-based Segmentation

- We know that image segmentation aims to partition an image into regions corresponding to objects or part of objects.

Region-based Segmentation

- We know that image segmentation aims to partition an image into regions corresponding to objects or part of objects.
- So far we have discussed segmentation by boundary detection and segmentation by thresholding.

Region-based Segmentation

- We know that image segmentation aims to partition an image into regions corresponding to objects or part of objects.
- So far we have discussed segmentation by boundary detection and segmentation by thresholding.
- Here we introduce two methods that form regions in the image plane, called region growing and region splitting and merging.

Region Growing

- Region growing groups pixels or regions into larger regions using specific predicates or criteria for growth.

Region Growing

- Region growing groups pixels or regions into larger regions using specific predicates or criteria for growth.
- Region growing starts from specific points that are called seeds.

Region Growing

- Region growing groups pixels or regions into larger regions using specific predicates or criteria for growth.
- Region growing starts from specific points that are called seeds.
- The selection of criteria is a key element. Frequently used properties include color, intensity and texture similarity.

Region Growing Challenges

- Selection of seeds can affect final results. Seeds can be selected manually or using prior knowledge of the imaging data.

Region Growing Challenges

- Selection of seeds can affect final results. Seeds can be selected manually or using prior knowledge of the imaging data.
- The stopping rule is another factor. Because image segmentation is an np-complete problem, the region growing path is suboptimal and can lead to errors.

Region Growing Challenges

- Selection of seeds can affect final results. Seeds can be selected manually or using prior knowledge of the imaging data.
- The stopping rule is another factor. Because image segmentation is an np-complete problem, the region growing path is suboptimal and can lead to errors.
- To address such challenges we can incorporate prior information about the object's expected intensity, color, texture or shape.

Region Growing Algorithm

Let, $f(x, y)$: input image, $S(x, y)$: seed array (1 for seeds, 0 for background), $Q(x, y)$: predicate applied to each location (x, y). Region growing using 8-connectivity can be implemented as follows:

1. Find all connected components in $S(x, y)$ and erode connected components down to one pixel.

Region Growing Algorithm

Let, $f(x,y)$: input image, $S(x,y)$: seed array (1 for seeds, 0 for background), $Q(x,y)$: predicate applied to each location (x,y). Region growing using 8-connectivity can be implemented as follows:

1. Find all connected components in $S(x,y)$ and erode connected components down to one pixel.
2. Form an image f_Q such that

$$f_Q(x,y) = \begin{cases} 1 & \text{if } Q(x,y) \text{ is TRUE} \\ 0 & \text{otherwise.} \end{cases}$$

Region Growing Algorithm

Let, $f(x,y)$: input image, $S(x,y)$: seed array (1 for seeds, 0 for background), $Q(x,y)$: predicate applied to each location (x,y). Region growing using 8-connectivity can be implemented as follows:

1. Find all connected components in $S(x,y)$ and erode connected components down to one pixel.
2. Form an image f_Q such that

$$f_Q(x,y) = \begin{cases} 1 & \text{if } Q(x,y) \text{ is TRUE} \\ 0 & \text{otherwise.} \end{cases}$$

3. Form an image g that will include the seeds and all the points $w_i(x_i, y_i)$ for which i) $f_Q(x_i, y_i = 1$, and ii) w_i is 8-connected to a seed point.

Region Growing Algorithm

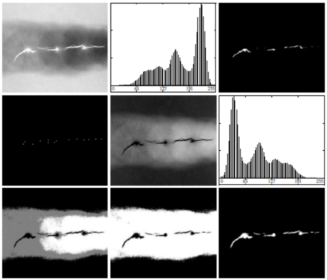
Let, $f(x,y)$: input image, $S(x,y)$: seed array (1 for seeds, 0 for background), $Q(x,y)$: predicate applied to each location (x,y). Region growing using 8-connectivity can be implemented as follows:

1. Find all connected components in $S(x,y)$ and erode connected components down to one pixel.
2. Form an image f_Q such that
$$f_Q(x,y) = \begin{cases} 1 & \text{if } Q(x,y) \text{ is TRUE} \\ 0 & \text{otherwise.} \end{cases}$$
3. Form an image g that will include the seeds and all the points $w_i(x_i,y_i)$ for which i) $f_Q(x_i,y_i = 1$, and ii) w_i is 8-connected to a seed point.
4. Assign to each connected component a unique region label and form the segmentation output.

Region Growing Example

- Region growing using threshold predicate.
- Predicate:

$$Q(x,y) = \begin{cases} \text{TRUE} & \text{if } |f(x,y) - s_i| \leq T \\ \text{FALSE} & \text{otherwise.} \end{cases}$$



Region Splitting

- These methods can be divided into two steps.

Region Splitting

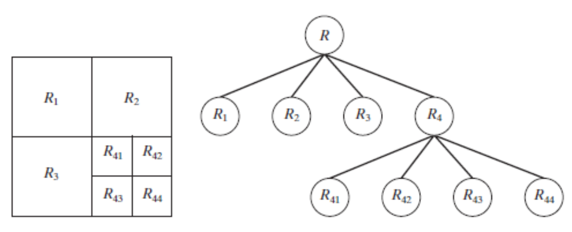
- These methods can be divided into two steps.
  1. Recursively divide the image into sub-regions for which a predicate Q is FALSE.

Region Splitting

- These methods can be divided into two steps.
  1. Recursively divide the image into sub-regions for which a predicate Q is FALSE.
  2. Merge neighboring sub-regions for which Q is TRUE.

Region Splitting

- These methods can be divided into two steps.
  1. Recursively divide the image into sub-regions for which a predicate Q is FALSE.
  2. Merge neighboring sub-regions for which Q is TRUE.
- A standard approach is to recursively divide the image into quadrants and generate hierarchical structures that are called quadtrees.



Region Splitting and Merging

* If we apply splitting only, then we may produce adjacent regions that are similar but belong to different connected components.
* To address this we can add a merging step as follows:

1. Recursively split into four quadrants any region R_i for which $Q(R_i)$ is FALSE.

* Several variations of this technique have been proposed, that may also employ graph theory.

Region Splitting and Merging

* If we apply splitting only, then we may produce adjacent regions that are similar but belong to different connected components.
* To address this we can add a merging step as follows:

1. Recursively split into four quadrants any region R_i for which $Q(R_i)$ is FALSE.
2. Recursively merge any adjacent regions R_i and R_j for which $Q(R_i \cup R_j)$ is TRUE.

* Several variations of this technique have been proposed, that may also employ graph theory.

Region Growing Example

- Region splitting and merging using a mean m_{R_i} and standard deviation σ_{R_i} - based predicate for a region R_i.
- Predicate:

$$Q(R_i) = \begin{cases} \text{TRUE} & \text{if } \sigma_{R_i} > a \text{ AND } 0 < m_{R_i} < b \\ \text{FALSE} & \text{otherwise.} \end{cases}$$