LECTURE NOTES

ON

IMAGE PROCESSING

B.Tech V Sem (IARE-R16)

By

Ms. S J Sowjanya
Assistant Professor

Ms. B Tejaswi
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

DUNDIGAL, HYDERABAD - 500 043

UNIT-I
INTRODUCTION

What is digital image processing?

Image – A two-dimensional signal that can be observed by human visual system
Digital image –Representation of images by sampling in time and space.
Digital image processing – perform digital signal processing operations on digital images

- An image may be defined as a two- dimensional function, f(x,y) where x and y are spatial (plane) coordinates, and the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point
- When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image.
- A digital image is composed of a finite number of elements, of which has a particular location and value
- These elements are referred to as picture elements, image elements and pixels.
- Pixel is the term most widely used to denote the elements of a digital image.

The notation introduced in the preceding paragraph allows us to write the complete M*N digital image in the following compact matrix form:

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

The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, pixel, or pel.

Origins of Digital Image Processing

- One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York.
- Specialized printing equipment coded pictures for cable transmission and then reconstructed them at the receiving end.
- Image was transmitted in this way and reproduced on a telegraph printer fitted with typefaces simulating a halftone pattern.
- The printing technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal from 1921.
- The improvements are tonal quality and in resolution.
- The early Bartlane systems were capable of coding images in five distinct levels of gray.
- This capability was increased to 15 levels in 1929.

Examples of fields that use DIP

- Gamma ray imaging
- X-ray Imaging (oldest source of EM radiation)
- Imaging in the visible and infrared bands
- Imaging in the microwave band
- Imaging in the radio band
- Other Imaging Modalities Acoustic images, electron microscopy and synthetic (computer – generated images

Gamma ray imaging

Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations. In nuclear medicine, the approach is to inject a patient with a radioactive isotope that emits gamma rays as it decays.

X-ray Imaging (oldest source of EM radiation)

X-rays for medical and industrial imaging are generated using an x-ray tube, which is a vacuum tube with a cathode and anode. The cathode is heated, causing free electrons to be released. These electrons flow at high speed to the positively charged anode. When the electron strike a nucleus, energy is released in the form x-ray radiation. The energy (penetrating power) of the x-rays is controlled by a current applied to the filament in the cathode.

Imaging in the visible and infrared bands

Infrared band often is used in conjunction with visual imaging. The applications ranges from light microscopy, astronomy, remote sensing industry and law enforcement.
E.g. Microscopy- the applications ranges from enhancement to measurement Remote sensing-weather observation from multispectral images from satellites Industry-check up the bottle drink with less quantity Law enforcement – biometrics

Imaging in the microwave band

Dominant application in microwave band is radar. The unique feature of imaging radar is its ability to collect data over virtually any region at any time, regardless of weather or ambient lighting conditions. Some radar waves can penetrate clouds and under certain conditions can also see through vegetation, ice and extremely dry sand. In many cases, radar is the only way to explore inaccessible regions of the earth’s surface.

Imaging in the radio band

Major applications of imaging in the radio band are in medicine and astronomy. In medicine radio waves are used in magnetic resonance imaging (MRI). This technique places a patient in a powerful magnet and passes radio waves through his or her body in short pulses. Each pulse causes a responding pulse of radio waves to be emitted by patient’s tissues. The location from which theses signals originate and their strength are determined by a computer which produces a two-dimensional picture of a section of the patient.

Other Imaging Modalities Acoustic images, electron microscopy and synthetic (computer – generated images

Imaging using sound finds application in geological exploration, industry and medicine. The most important commercial applications of image processing in geology are in mineral and oil exploration.

Ultrasound imaging is used routinely in manufacturing, the best known applications of this technique are in medicine, especially in obsterics, where unborn babies are imaged to determine the health of their development.

Fundamental steps in DIP

There are some fundamental steps but as they are fundamental, all these steps may have sub-steps. The fundamental steps are described below with a neat diagram.

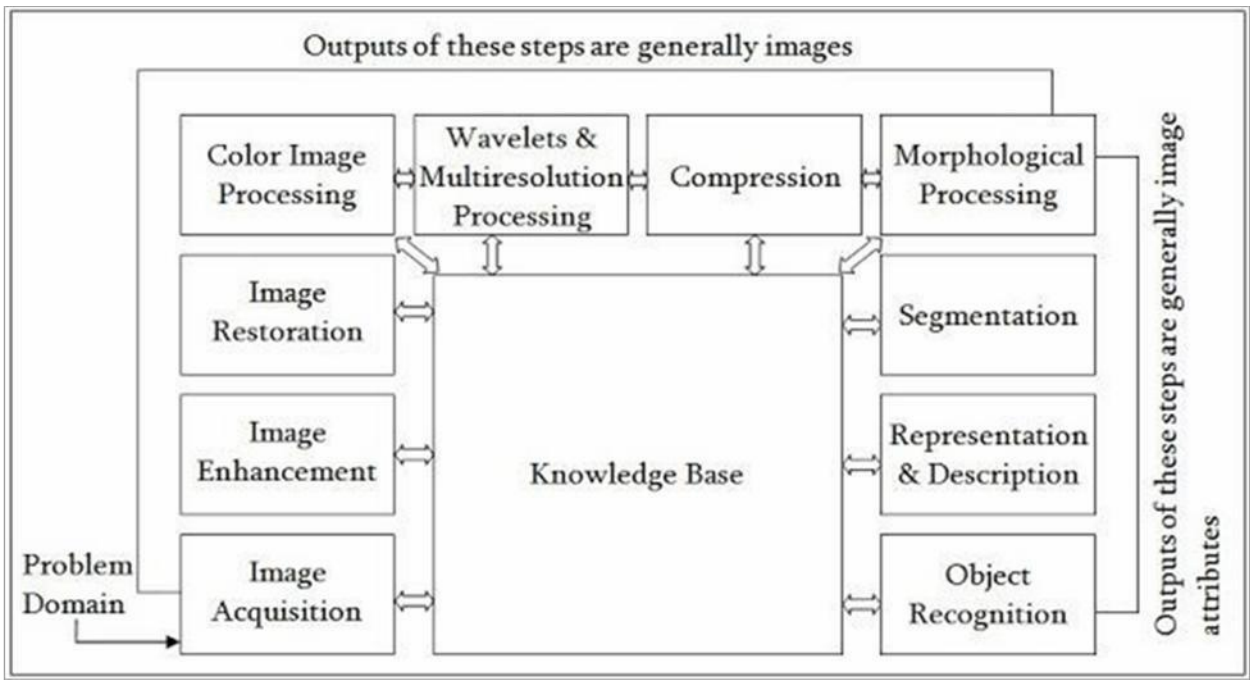


Figure **1.1**: Fundamental steps in Digital Image Processing

1. Image Acquisition:

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre- processing, such as scaling etc.

2. Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness &

contrast etc.

3. Image Restoration:
Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

4. Color Image Processing:
Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

5. Wavelets and Multi-Resolution Processing:
Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

6. Compression:
Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data.

7. Morphological Processing:
Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

8. Segmentation:
Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing.

9. Representation and Description:
Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing.

10. Object recognition:
Recognition is the process that assigns a label, such as, “vehicle” to an object based on its descriptors.

11. Knowledge Base:
Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information.

Components of image processing system

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images. Figure shows the basic components comprising a typical general- purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

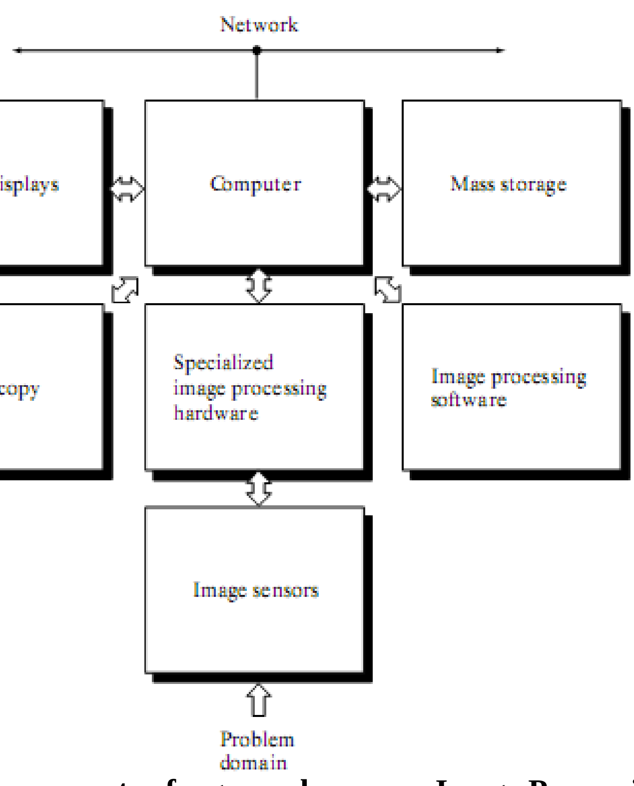


Figure 1.2: Components of a general purpose Image Processing System

- In sensing, two elements are required to acquire digital images. The first is physical device that is sensitive to the energy radiated by the object we wish to image. The second called a digitizer, is a device for converting the output of the physical sensing device into digital form.
- Specialized image processing hardware usually consists of the digitizer plus hardware that performs other primitive operations such as arithmetic and logical operations (ALU).
- The computer is an image processing system is a general purpose to supercomputer
- Software which include image processing specialized modules that perform specific tasks.
- Mass storage capability is a must in image processing applications.
- Image displays in use today are mainly color tv monitors.
- hardcopy devices for recording images include laser printers, film cameras, inkjet units and CD-ROM
- Networking for communication

Elements of visual perception

Although the digital image processing field is built on a foundation of mathematical and probabilistic formulations, human intuition and analysis play a central role in the choice of one technique versus another, and this choice often is made based on subjective, visual judgments.

1. Structure of the Human Eye:

The eye is nearly a sphere, with an average diameter of approximately 20mm.Three membranes enclose the eye:

Cornea: The cornea is a tough, transparent tissue that covers the anterior surface of the eye
Sclera: sclera is an opaque membrane that encloses the remainder of the optic globe.

Choroid: The choroid lies directly below the sclera. This membrane contains a net- work of blood vessels that serve as the major source of nutrition to the eye. The choroid coat is heavily pigmented and hence helps to reduce the amount of extraneous light entering the eye and the backscatter within the optical globe.

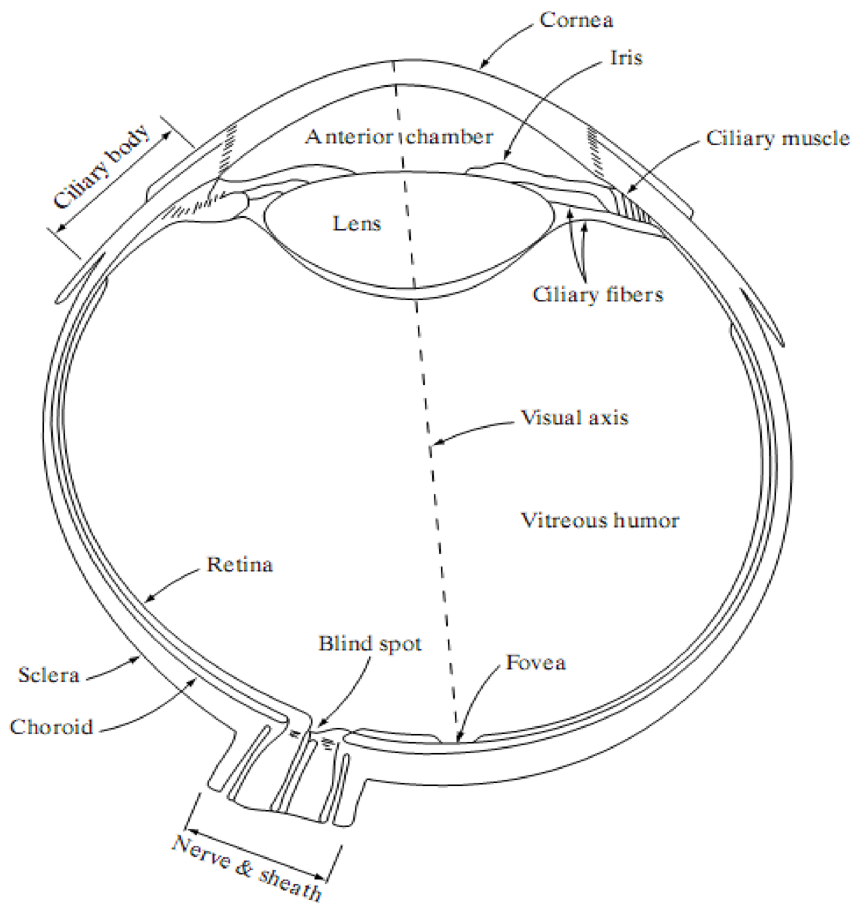


Figure 1.3:Simplified diagram of a cross section of the human eye.

The lens is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It contains 60 to 70% water, about 6% fat, and more protein than any other tissue in the eye. The innermost membrane of the eye is the retina, which lines the Inside of all entire posterior portions. When the eye is properly focused, light from an object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina.

There are two classes of receptors: cones and rods.

- The cones in each eye number between 6 and 7 million
- They are located primarily in the central portion of the retina, called the fovea, and are highly sensitive to color.

Muscles controlling the eye rotate the eyeball until the image of an object of interest falls on the fovea. Cone vision is called photopic or bright-light vision. The number of rods is much larger: Some 75 to 150 million are distributed over the retinal surface.

2. Image formation in the eye:

The principal difference between the lens of the eye and an ordinary optical lens is that the former is flexible. As illustrated in Fig. 3.1, the radius of curvature of the anterior surface of the lens is greater than the radius of its posterior surface. The shape of the lens is controlled by tension in the fibers of the ciliary body. To focus on distant objects, the controlling muscles cause the lens to be relatively flattened. Similarly, these muscles allow the lens to become thicker in order to focus on objects near the eye. The distance between the center of the lens and the retina (called the focal length) varies from approximately 17 mm to about 14 mm, as the refractive power of the lens increases from its minimum to its maximum. When the eye

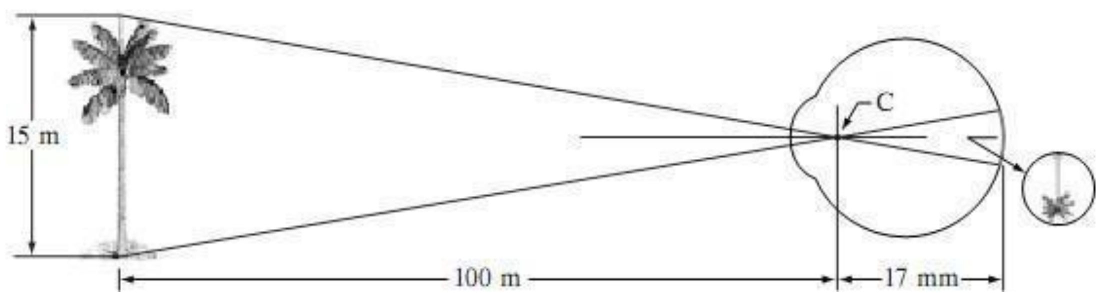


Figure 1.4: Graphical representation of the eye looking at a palm tree Point C is the optical center of the lens.

When the eye focuses on an object farther away than about 3 m, the lens exhibits its lowest refractive power. When the eye focuses on a nearby object, the lens is most strongly refractive. This information

When the eye focuses on a nearby object, the lens is most strongly refractive. For example, the observer is looking at a tree 15 m high at a distance of 100 m. If h is the height in mm of that object in the retinal image, the geometry of Fig. yields 15/100 = h/17 or h=2.55mm.

<u>Simple image formation model</u>

- The two functions combine as a product to form f(x,y): f(x,y)=i(x,y)r(x,y) where
  r(x,y)=0-----Total absorption
  r(x,y)=1----Total reflection
- The intensity of a monochrome image f at any coordinates(x,y) the *gray level(l) of the image at that point.*
- *That is,l=f(x0,y0) L lies in the range Lmin≤l ≤ Lmax where Lmin= lmin rmin and Lmax=lmax rmax*

Image Sensing and Acquisition:

The types of images in which we are interested are generated by the combination of an–illumination source and the reflection or absorption of energy from that source by the elements of the –scene being imaged. We enclose illumination and scene in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene. For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern.

Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock formations, or a human brain. We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light reflected from a planar surface. An example in the second category is when X-rays pass through a patient's body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach.

Figure 1.5 shows the three principal sensor arrangements used to transform illumination energy into digital images. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response.

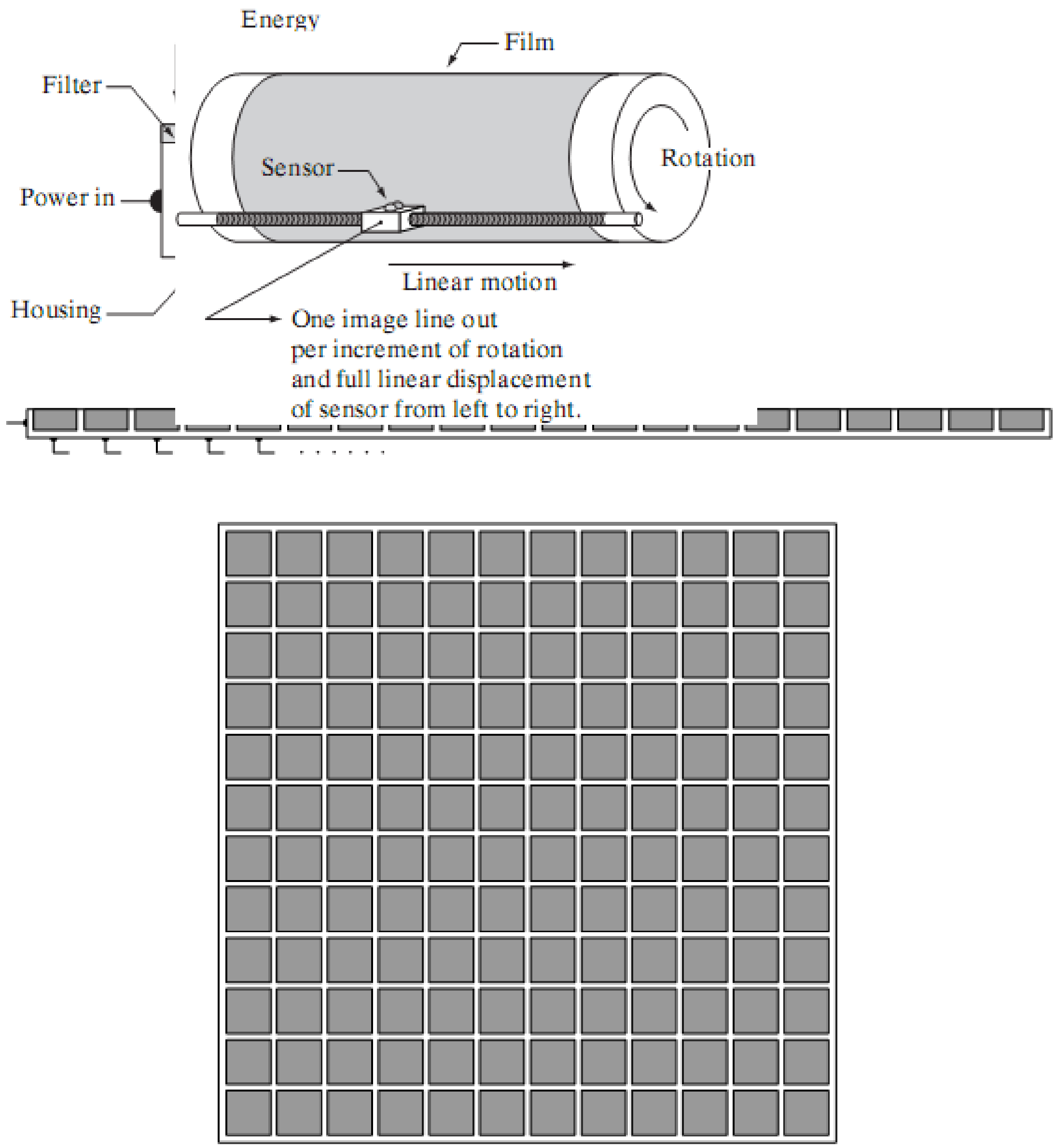


Figure 1.5: Single imaging Sensor (b) Line sensor (c) Array sensor

(1) Image Acquisition Using a Single Sensor:

Figure 1.5 (a) shows the components of a single sensor. Perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform is proportional to light. The use of a filter in front of a sensor improves selectivity. For example, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.

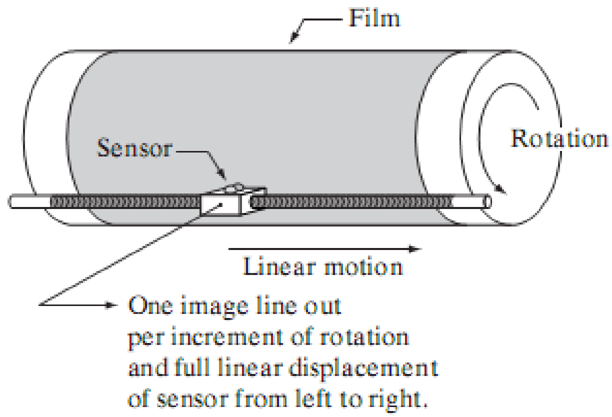


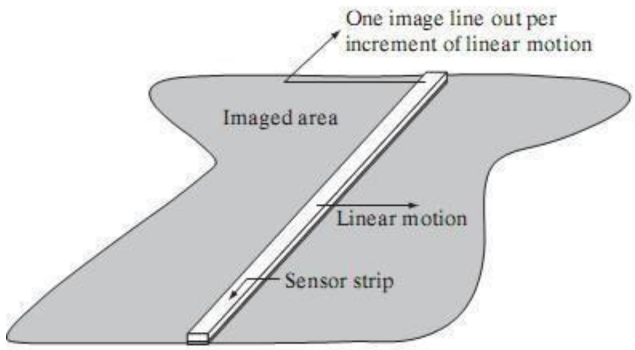
Figure 1.6: Combining a single sensor with motion to generate a 2-D image

Figure 4.2 shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Since mechanical motion can be controlled with high precision, this method is an inexpensive (but slow) way to obtain high-resolution images. Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as microdensitometers.

(2) Image Acquisition Using Sensor Strips:

A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, as Fig. 4.1 (b) shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction, as shown in Fig. 4.3 (a)

Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional (―slice‖) images of 3-D objects, as Fig. 4.3 (b) shows. A rotating X-ray source provides illumination and the portion of the sensors opposite the source collect the X-ray energy that pass through the object (the sensors obviously have to be sensitive to X-ray energy).This is the basis for medical and industrial computerized axial tomography (CAT). It is important to note that the output of the sensors must be processed by reconstruction algorithms whose objective is to transform the sensed data into meaningful cross-sectional images.



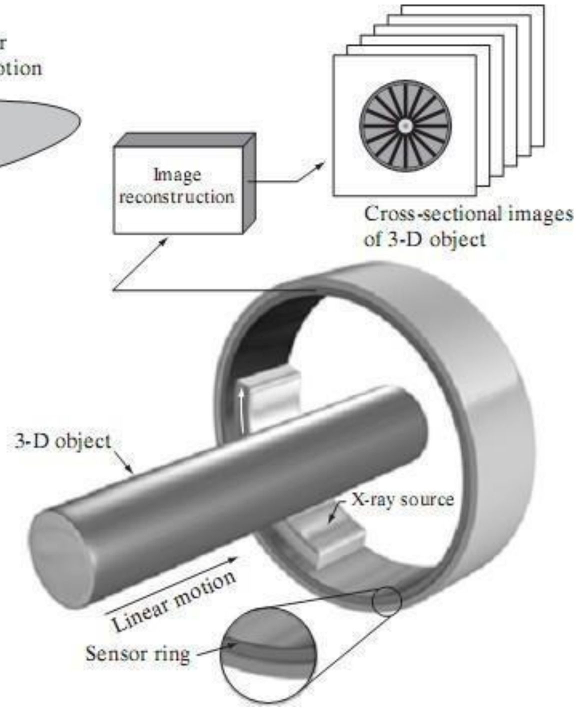


Figure 1.7: (a) Image acquisition using a linear sensor strip (b) Image acquisition using a circular sensor strip.

(3) Image Acquisition Using Sensor Arrays:

Figure 4.1 (c) shows individual sensors arranged in the form of a 2-D array. Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras. A typical sensor for these cameras is a CCD array, which can be manufactured with a broad range of sensing properties and can be packaged in rugged arrays of 4000 * 4000 elements or more. CCD sensors are used widely in digital cameras and other light sensing instruments. The response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor, a property that is used in astronomical and other applications requiring low noise images. Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or even hours. Since the sensor array shown in Fig. 4.4 (c) is two dimensional, its key advantage is that a complete image can be obtained by focusing the energy pattern onto the surface of the array. The principal manner in which array sensors are used is shown in Fig.4.4.

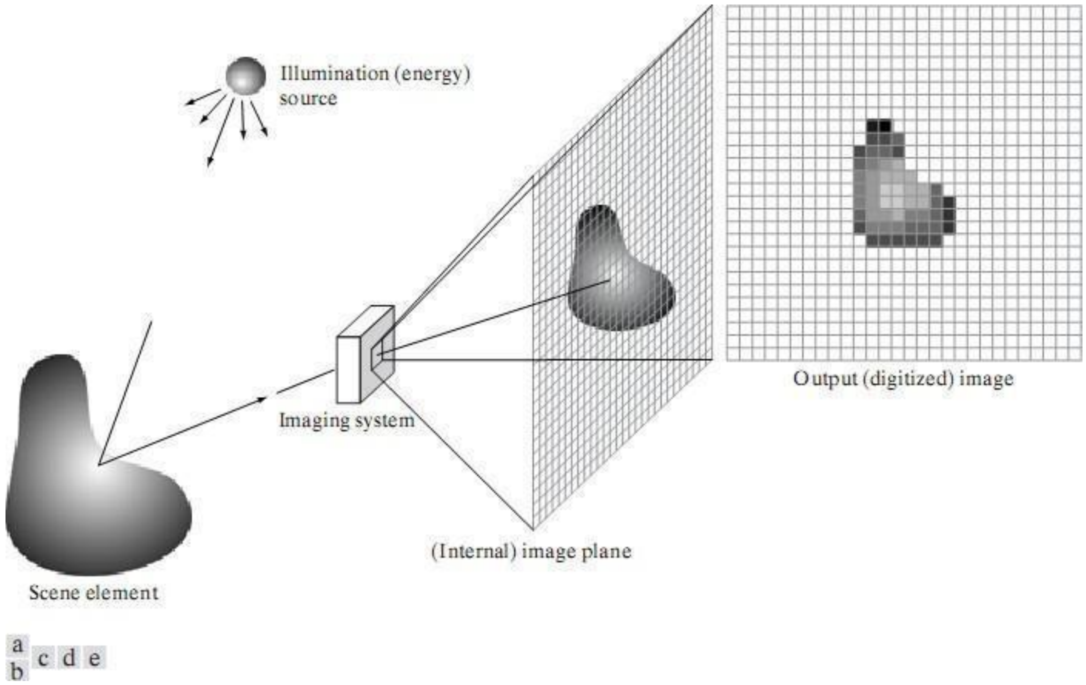


Figure 1.8: An example of the digital image acquisition process (a) Energy (―illumination‖) source (b) An element of a scene (c) Imaging system (d) Projection of the scene onto the image plane (e) Digitized image

Basic concepts in Sampling and Quantization

To create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done.

1. Sampling
2. Quantization

Since an image is continuous not just in its co-ordinates (x axis), but also in its amplitude (y axis), so the part that deals with the digitizing of co-ordinates is known as sampling and the part that deals with digitizing the amplitude is known as quantization

The one-dimensional function shown in Figure 1.9 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Figure 1.9 (a).The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Figure 1.9 (c).The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of Figure 1.9 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Figure 1.9 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image. Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single

Sensing element combined with mechanical motion, as in Fig below the output of the sensor is quantized in the manner described above. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle; there is almost no limit as to how fine we can sample an image

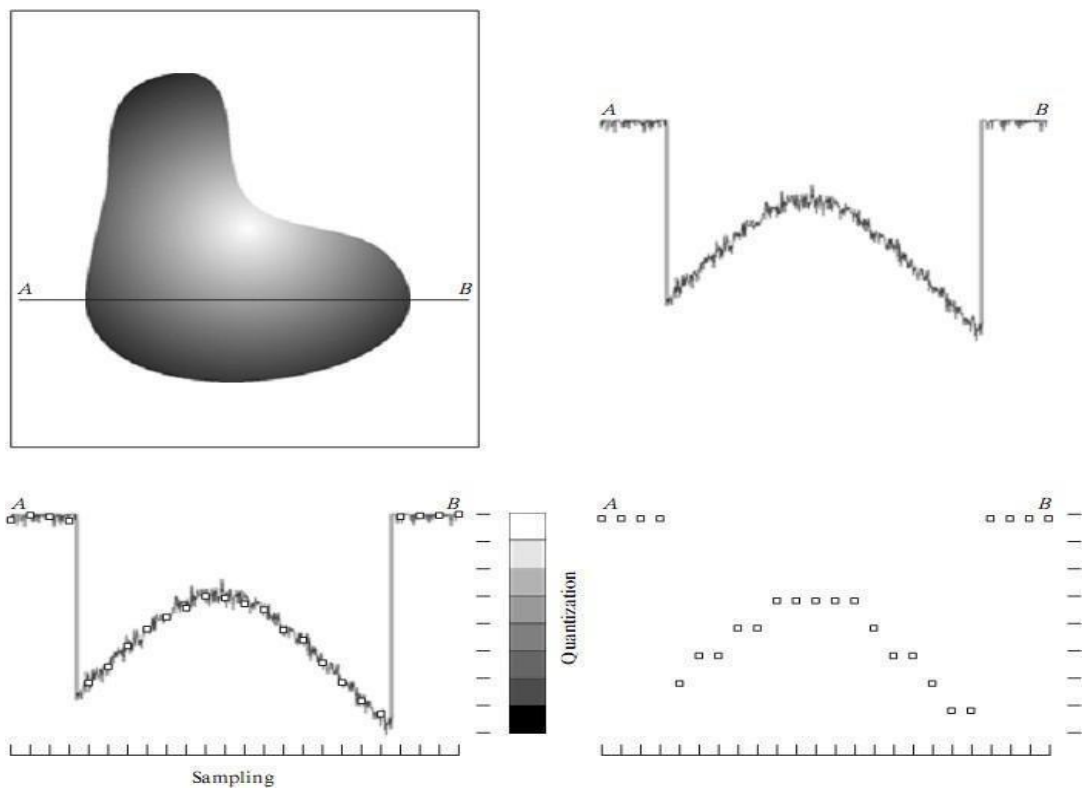


Figure 1.9: Generating a digital image (a) Continuous image (b) A scan line from A to Bin the continuous image, used to illustrate the concepts of sampling and quantization (c) Sampling and quantization. (d) Digital scan line

Representing Digital Images

We will use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting digital image has M rows and N columns. The values of the coordinates (x, y) now become discrete quantities. For notational clarity and convenience, we shall use integer values for these discrete coordinates. Thus, the values of the coordinates at the origin are (x, y) = (0, 0). The next coordinate values along the first row of the image are represented as (x, y) = (0, 1). It is important to keep in mind that the notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure shows the coordinate convention used.

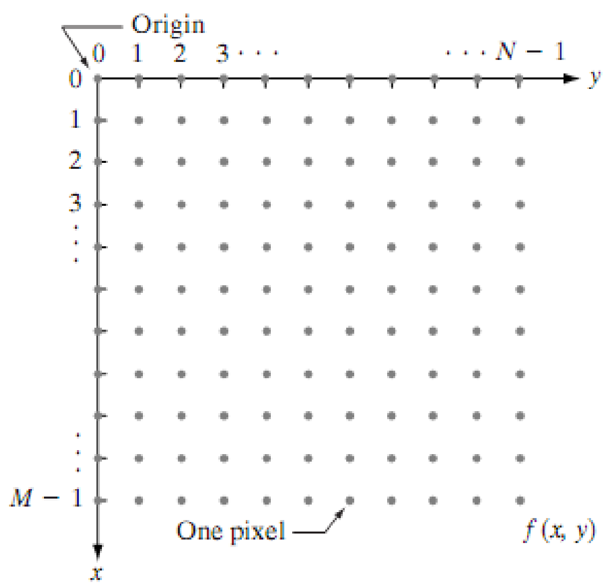


Figure 1.10: Coordinate convention used to represent digital images

The notation introduced in the preceding paragraph allows us to write the complete M*N digital image in the following compact matrix form:

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

The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, pixel, or pel.

Spatial and Gray-Level Resolution:

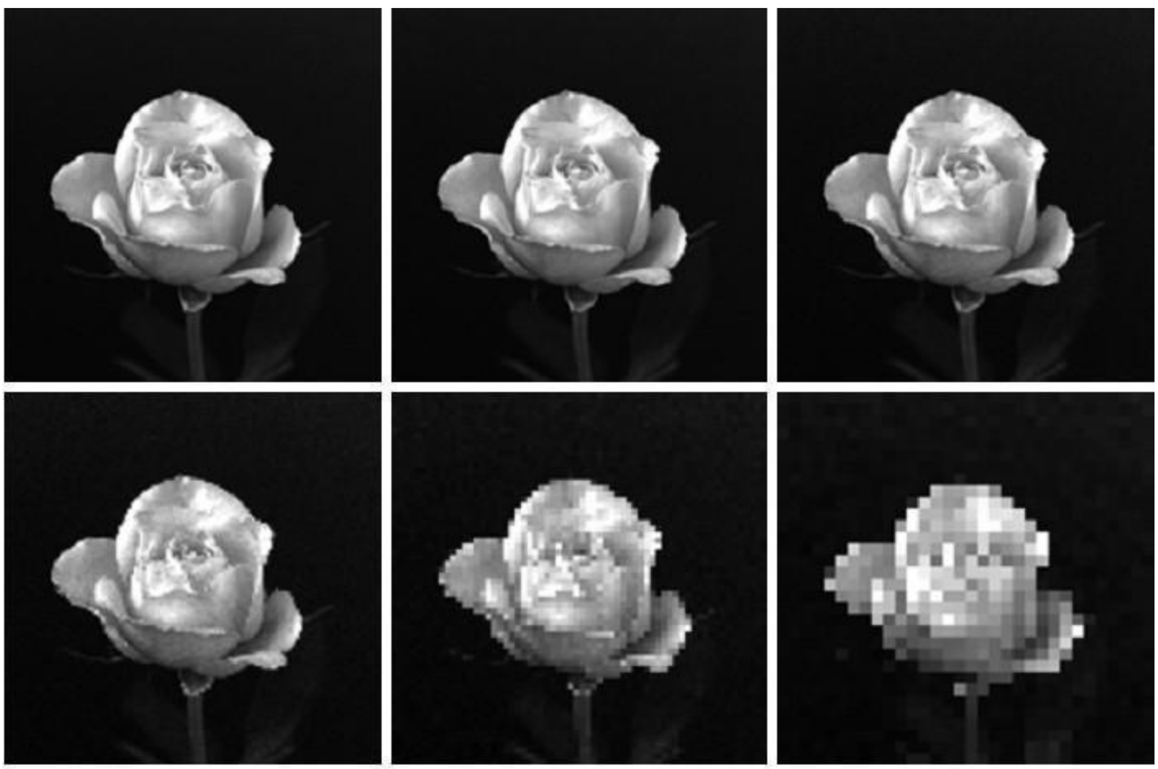
Sampling is the principal factor determining the spatial resolution of an image. Basically, spatial resolution is the smallest discernible detail in an image. Suppose that we construct a chart with vertical lines of width W, with the space between the lines also having width W.A line pair consists of one such line and its adjacent space. Thus, the width of a line pair is 2W, and there are 1/2Wline pairs per unit distance. A widely used definition of resolution is simply the smallest number of discernible line pairs per unit distance; for example, 100 line pairs per millimeter. Gray- level resolution similarly refers to the smallest discernible change in gray level. We have considerable discretion regarding the number of samples used to generate a digital image, but this is not true for the number of gray levels. Due to hardware considerations, the number of gray levels is usually an integer power of 2.

The most common number is 8 bits, with 16 bits being used in some applications where enhancement of specific gray-level ranges is necessary. Sometimes we find systems that can digitize the gray levels of an image with 10 or 12 bit of accuracy, but these are the exception rather than the rule. When an actual measure of physical resolution relating pixels and the level of detail they resolve in the original scene are not necessary, it is not uncommon to refer to an L- level digital image of size M*N as having a spatial resolution of M*N pixels and a gray-level resolution of L levels.

Figure 1.11:A 1024*1024, 8-bit image subsampled down to size 32*32 pixels the number of allowable gray levels was kept at 256.

The sub sampling was accomplished by deleting the appropriate number of rows and columns from the original image. For example, the 512*512 image was obtained by deleting every other row and column from the 1024*1024 image. The 256*256 image was generated by deleting every other row and column in the 512*512 image, and so on. The number of allowed gray levels was kept at 256. These images show the dimensional proportions between various sampling densities, but their size differences make it difficult to see the effects resulting from a reduction in the number of samples. The simplest way to compare these effects is to bring all the sub sampled images up to size 1024*1024 by row and column pixel replication. The results are shown in Figs. 1.12 (b) through (f). Figure 1.12 (a) is the same 1024*1024, 256-level image shown in Fig.1.12; it is repeated to facilitate comparisons.

Figure 1.12: 1024*1024, 8-bit image (b) 512*512 image re sampled into 1024*1024 pixels by row and column duplication (c) through (f) 256*256, 128*128, 64*64, and 32*32 images re sampled into 1024*1024 pixels



a b c
d e f

<u>Zooming and Shrinking of Digital images</u>

<u>Zooming:</u> It may be viewed as over sampling and increasing number of pixels in an image so that image appears bigger.

It requires two steps

1. Creation of new pixels
2. Assignment of gray levels to those new locations.

Zooming Methods
1. Nearest Neighbor interpolation
2. Bilinear interpolation
3. K-times zooming

Shrinking: It may be viewed as under sampling. It is performed by row-column deletion.

Basic Relationships between pixels
1. Neighborhood
2. Adjacency
3. Paths
4. Connectivity
5. Regions and boundaries

Neighbors of a Pixel:

A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by (x+1, y), (x-1, y), (x, y+1), (x, y-1). This set of pixels, called the 4-neighbors of p, is denoted by N_4 (p). Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) and are denoted by N_D (p). These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N_8 (p). As before, some of the points in N_D (p) and N_8 (p) fall outside the image if (x, y) is on the border of the image

Connectivity:

Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To establish if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal). For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are said to be connected only if they have the same value.

Let V be the set of gray-level values used to define adjacency. In a binary image, V={1} if we are referring to adjacency of pixels with value 1. In a grayscale image, the idea is the same, but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible gray-level values 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency:

(a) 4-adjacency. Two pixels p and q with values from V are 4-adjacent if q is in the set N_4(p).

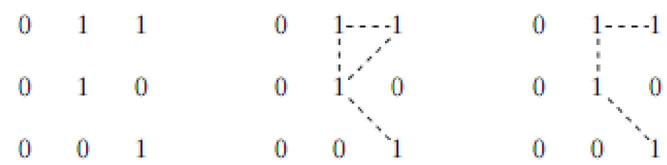
(b) 8-adjacency. Two pixels p and q with values from V are 8-adjacent if q is in the set N_8(p).

(c) m-adjacency (mixed adjacency).Two pixels p and q with values from V are m-adjacent if

(i) q is in N_4 (p), or

(ii) q is in N_D (p) and the set has no pixels whose values are from V.

Figure 1.13: (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency



a b c

Linear and Nonlinear operations

- General operator, H, that performs an output image, g(x,y),for a given input image, f(x,y)

  $H[f(x,y)]=g(x,y)$
- H is said to be linear operator if
- $H[a_i f_i(x,y)+a_j f_j(x,y)]= a_i H[f_i(x,y)]+a_j H[f_j(x,y)]$

  $=a_i g_i(x,y)+a_j g_j(x,y)$

Where ai,aj are arbitrary constants and fi(x,y),fj(x,y) are images of same size.

- For example sum is a linear operator and max is nonlinear operator

UNIT-II IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

Image Enhancement

- The principal objective of enhancement is to process an image so that the result is more suitable for a special process

- Image Enhancement Fall into two categories: Enhancement in spatial domain and Frequency domain.

- The term spatial domain refers to the Image Plane itself which is DIRECT manipulation of pixels

- Frequency domain processing techniques are based on modifying the Fourier transform of an image.

Image Enhancement in the spatial domain

- Spatial Domain=Aggregate of pixels composing an image.
- Spatial Domain Methods=Procedures that operate directly on the pixels. Denoted by: g(x,y)=T[f(x,y)]
- f(x,y) : Input Image ,
- T: Operator on Image g(x,y): Processed Image.
- T also can operate on a set of Images.

Definition of Neighborhood:

Input for Process: A neighborhood about a point (x,y). The simplest form of input is a one pixel neighborhood. s=T(r) T: Transformation Function s, r: gray level of f(x,y) and g(x,y) respectively.

Basic Gray Level Transformations:

The study of image enhancement techniques is done by discussing gray-level transformation functions. These are among the simplest of all image enhancement techniques. The values of pixels, before and after processing, will be denoted by r and s, respectively. As indicated in the previous section, these values are related by an expression of the form s=T(r), where T is a transformation that maps a pixel value r into a pixel value s. Since we are dealing with digital quantities, values of the transformation function typically are stored in a one-dimensional array and the mappings from r to s are implemented via table lookups. For an 8-bit environment, a lookup table containing the values of T will have 256 entries. As an introduction to gray-level transformations, consider Fig. 1.1, which shows three basic types of functions used frequently for image enhancement: linear (negative and identity transformations), logarithmic (log and inverse-log transformations), and power-law (nth power and nth root transformations).The identity function is the trivial case in which output intensities are identical to input intensities. It is included in the graph only for completeness.

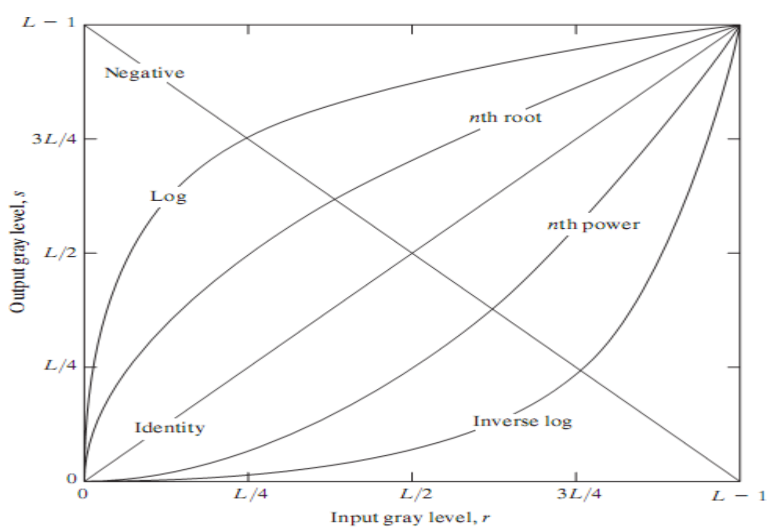
Image Negatives:

The negative of an image with gray levels in the range [0, L-1] is obtained by using the negative transformation shown in Fig.1.1, which is given by the expression

$$s = L - 1 - r.$$

Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.

Figure 2.1:Some basic gray-level transformation functions used for image enhancement



Log Transformations:

The general form of the log transformation shown in Fig.1.1 is

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

where c is a constant, and it is assumed that $r \geq 0$.The shape of the log curve in Fig. 1.1 shows that this transformation maps a narrow range of low gray-level values in the input image into a wider range of output levels. The opposite is true of higher values of input levels. We would use a transformation of this type to expand the values of dark pixels in an image while compressing the higher-level values. The opposite is true of the inverse log transformation.

Power-Law Transformations:

Power-law transformations have the basic form

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

Where c and g are positive constants. Sometimes Eq. is written as $s = c(r + \varepsilon)^{\gamma}$

Piecewise-Linear Transformation Functions:

The principal advantage of piecewise linear functions over the types of functions we have discussed above is that the form of piecewise functions can be arbitrarily complex. In fact, as we will see shortly, a practical implementation of some important transformations can be formulated only as piecewise functions. The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

Contrast stretching:

One of the simplest piecewise linear functions is a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition. The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.

Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images. There are several ways of doing level slicing, but most of them are variations of two basic themes. One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels

Bit-plane slicing

Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits.

Histogram Processing:

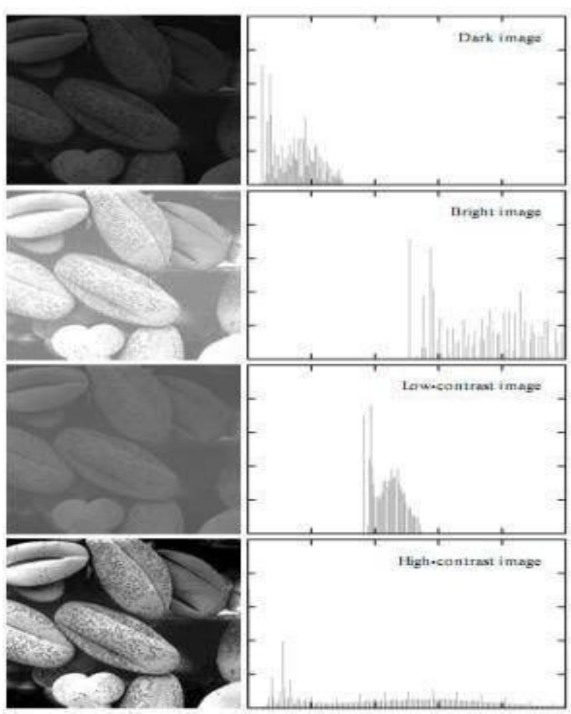
The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k) = (n_k)$, where r_k is the kth gray level and n_k is the number of pixels in the image having

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

gray level r_k. It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by for k=0,1,…… .,L-1. Loosely speaking, p (r_k) gives an estimate of the probability of occurrence of gray level r_k. Note that the sum of all components of a normalized histogram is equal to 1.

Histograms are the basis for numerous spatial domain processing techniques. Histogram manipulation can be used effectively for image enhancement. Histograms are simple to calculate in software and also lend themselves to economic hardware implementations, thus making them a popular tool for real-time image processing.

Fig. Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.



Histogram Equalization:

Consider for a moment continuous functions, and let the variable r represent the gray levels of the image to be enhanced. We assume that r has been normalized to the interval [0, 1], with r=0 representing black and r=1 representing white. Later, we consider a discrete formulation and allow pixel values to be in the interval [0, L-1]. For any r satisfying the aforementioned conditions, we focus attention on transformations of the form

$$s = T(r) \qquad 0 \leq r \leq 1$$

that produce a level s for every pixel value r in the original image. For reasons that will become obvious shortly, we assume that the transformation function T(r) satisfies the following conditions:

(a) T(r) is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$;

and

(b) (b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

Histogram Matching (Specification):

Histogram equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram. When automatic enhancement is desired, this is a good approach because the results from this technique are predictable and the method is simple to implement. In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have. The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Basics of Spatial Filtering:

Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a sub image that has the same dimensions as the neighborhood. The sub image is called a filter, mask, kernel, template, or window, with the first three terms being the most prevalent terminology. The values in a filter sub image are referred to as coefficients, rather than pixels. The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain. We use the term spatial filtering to differentiate this type of process from the more traditional frequency domain filtering.

The mechanics of spatial filtering are illustrated in Fig below the process consists simply of moving the filter mask from point to point in an image. At each point (x, y), the response of the filter at that point is calculated using a predefined relationship. The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 x 3 mask shown in Fig. the result (or response), R, of linear filtering with the filter mask at a point (x, y) in the image is

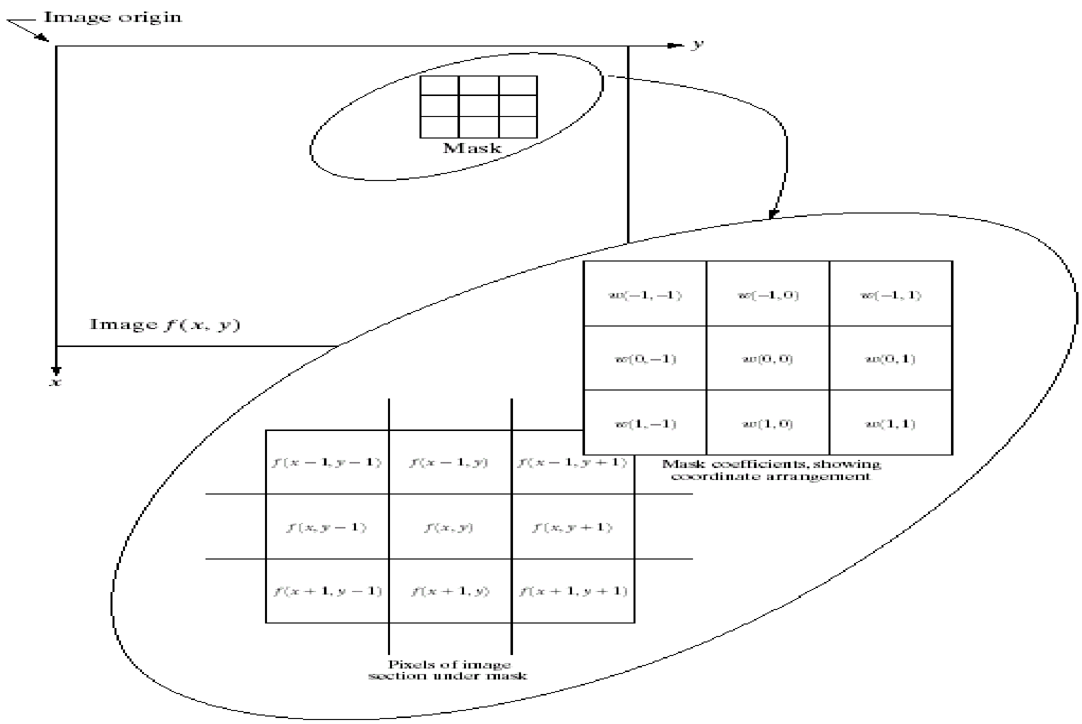


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

$$R = w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \cdots + w(0,0)f(x, y) + \cdots + w(1,0)f(x+1, y) + w(1,1)f(x+1, y+1),$$

Fig. The mechanics of spatial filtering. The magnified drawing shows a 3X3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability

Smoothing Spatial Filters:

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by non-linear filtering.

(1) Smoothing Linear Filters:

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters. The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced –sharp‖ transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges. Another application of this type of process includes the smoothing of false contours that result from using an insufficient number of gray levels.

- Averaging
- Gaussian
- Median filtering (non-linear)

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

Fig. Two 3 x 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

2. Order-Statistics Filters:

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The best-known example in this category is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of impulse noise, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image

We first sort the values of the pixel in question and its neighbors, determine their median and assign this value to that pixel. For example, in a 3 x 3 neighborhood the median is the 5th largest value, in a 5 x 5 neighborhood the 13th largest value, and so on. When several values in a neighborhood are the same, all equal values are grouped. For example, suppose that a 3 x 3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, and 100).

These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors. In fact, isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^{2}/2$ (one-half the filter area), are eliminated by an n x n median filter. In this case–eliminated means forced to the median intensity of the neighbors. Larger clusters are affected considerably less.

Sharpening spatial filters

- Unsharp masking
- High Boost filter
- Gradient (1^{st} derivative)
- Laplacian (2^{nd} derivative)

Unsharp masking

- Obtain a sharp image by subtracting a lowpass filtered (i.e., smoothed) image from the original image.

$$Highpass = Original - Lowpass$$

High Boost filter

- If **A=1**, we get unsharp masking.
- If **A>1**, part of the original image is added back to the high pass filtered image.
- One way to implement high boost filtering is using the masks below:

A>=1
w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

A=2
w = 17

-1	-1	-1
-1	17	-1
-1	-1	-1

Gradient (1^{st} derivative)

- Taking the derivative of an image results in sharpening the image.
- The derivative of an image (i.e., 2D signal) can be computed using the gradient

$$grad(f) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$f'(x) = \lim_{h->0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x) \quad (h=1)$$

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

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

Laplacian (2nd derivative)

Approximate 2nd derivatives

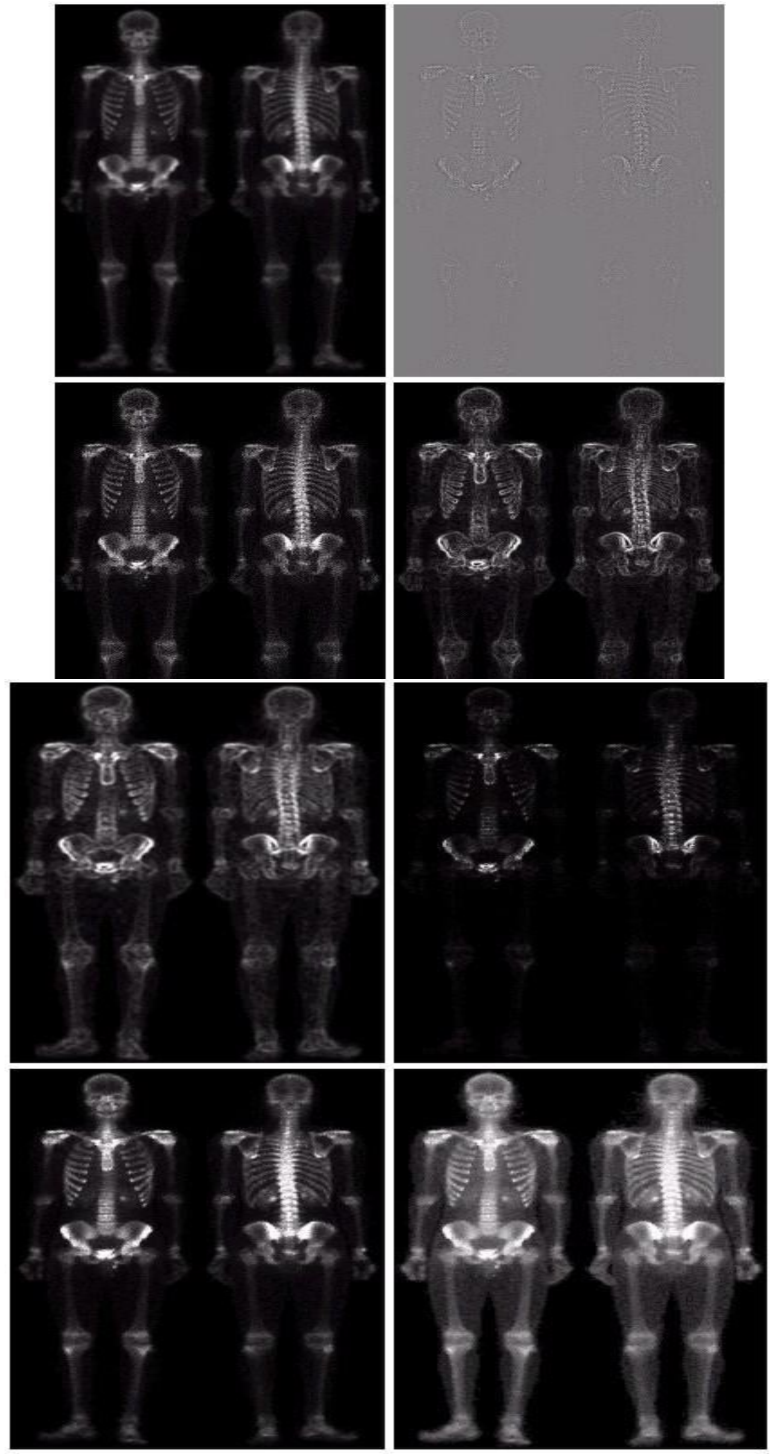
$$\nabla^2 = \nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

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

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

$$\nabla^2 f = -4f(i,j) + f(i,j+1) + f(i,j-1) + f(i+1,j) + f(i-1,j)$$

Combining spatial enhancement methods



a b
c d

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

e f
g h

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

Image Enhancement in Frequency Domain

- Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient (Fourier series).
- Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function (Fourier transform).
- The frequency domain refers to the plane of the two dimensional discrete Fourier transform of an image.
- The purpose of the Fourier transform is to represent a signal as a linear combination of sinusoidal signals of various frequencies.

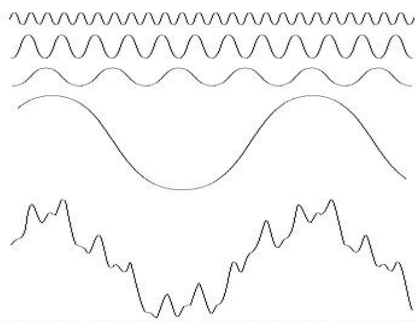


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Filtering in the frequency domain

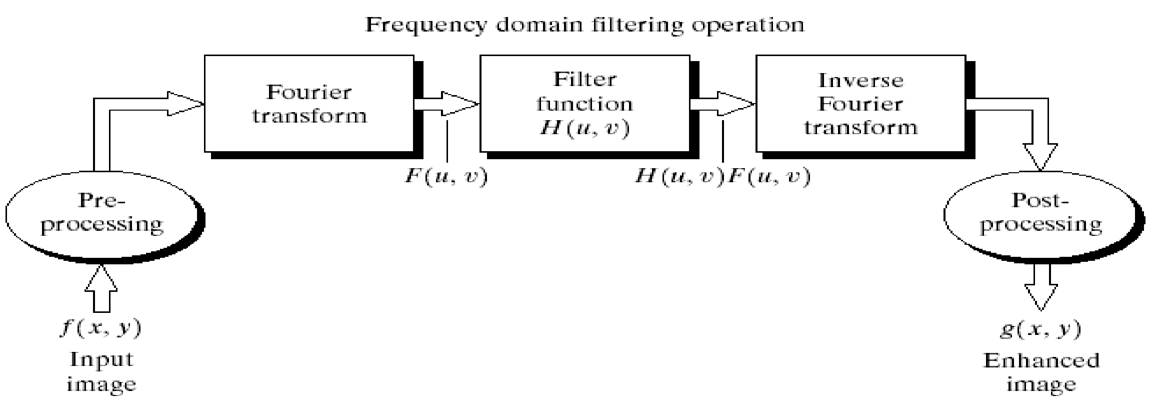


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Introduction to the Fourier Transform and the Frequency Domain

The Fourier transform simply states that that the non-periodic signals whose area under the curve is finite can also be represented into integrals of the sines and cosines after being multiplied by a certain weight.

The Fourier transform has many wide applications that include, image compression (e.g JPEG compression) , filtering and image analysis.

The formula for 2 dimensional discrete Fourier transform is given below.

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

The discrete Fourier transform is actually the sampled Fourier transform, so it contains some samples that denotes an image. In the above formula f(x,y) denotes the image , and F(u,v) denotes the discrete Fourier transform. The formula for 2 dimensional inverse discrete Fourier transform is given below.

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

Smoothing Filters in Frequency Domain

The basic model for filtering in the frequency domain G(u,v) = H(u,v)F(u,v) where $F(u,v)$: the Fourier transform of the image to be smoothed $H(u,v)$: a filter transfer function.

- Smoothing is fundamentally a lowpass operation in the frequency domain.
- There are several standard forms of lowpass filters (LPF).
  - Ideal lowpass filter
  - Butterworth lowpass filter
  - Gaussian lowpass filter

Ideal low-pass filters

- The simplest lowpass filter is a filter that "cuts off" all high-

frequency components of the Fourier transform that are at a distance greater than a specified distance D_0 from the origin of the transform.

- The transfer function of an ideal lowpass filter

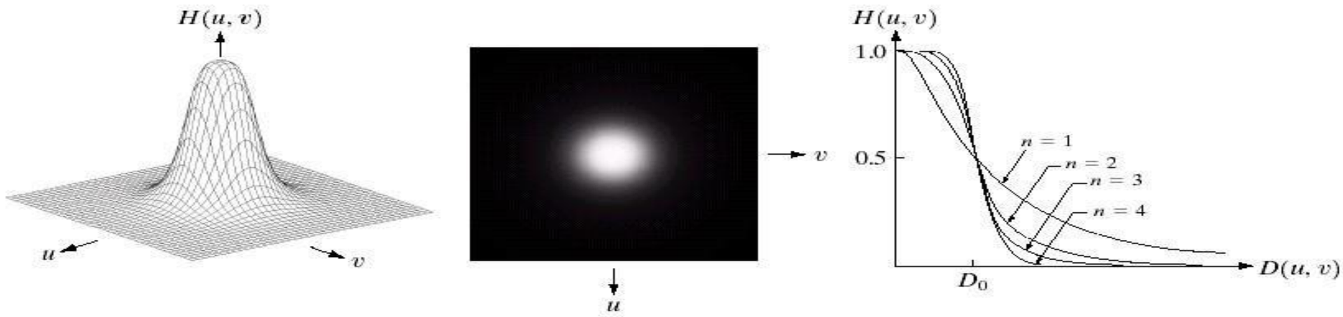
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

Where $D(u,v)$: the distance from point (u,v) to the center of ther frequency rectangle

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

Butterworth Lowpass Filters (BLPFs) with order *n*

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



a b c

FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies. For filters with smooth transfer functions, defining a cutoff frequency locus at points for which H (u, v) is down to a certain fraction of its maximum value is customary. In the case of above Eq. H (u, v) = 0.5 (down 50 percent from its maximum value of 1) when D (u, v) = Do. Another value commonly used is 1/√2 of the maximum value of H (u, v). The following simple modification yields the desired value when D (u, v) = Do:

$$H(u,v) = \frac{1}{1 + [\sqrt{2} - 1][D(u,v)/D_0]^{2n}} = \frac{1}{1 + 0.414[D(u,v)/D_0]^{2n}}.$$

Gaussian low pass filters

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

D0 is a measure of the spread of the Gaussian curve. By letting σ = Du, we can express

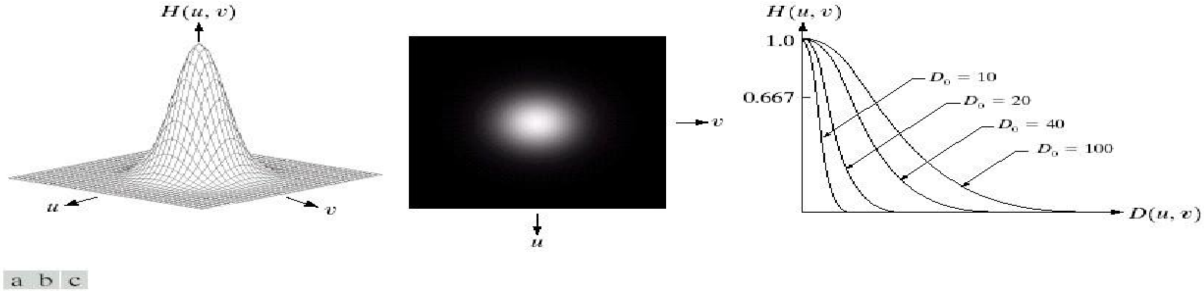


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

the filter in a more familiar form in terms of the notation:

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

Where Do is the cutoff frequency. When D (u, v) = Do, the filter is down to 0.607 of its maximum value.

Sharpening Filters in Frequency Domain

High pass Filtering:

An image can be blurred by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in gray levels are associated with high-frequency components, image sharpening can be achieved in the frequency domain by a high pass filtering process, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

Ideal filter:

2-D ideal high pass filter (IHPF) is one whose transfer function satisfies the relation

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

Where Do is the cutoff distance measured from the origin of the frequency plane. Figure shows a perspective plot and cross section of the IHPF function. This filter is the opposite of the ideal lowpass filter, because it completely attenuates all frequencies inside a circle of radius Do while passing, without attenuation, all frequencies outside the circle. As in the case of the ideal lowpass filler, the IHPF is not physically realizable

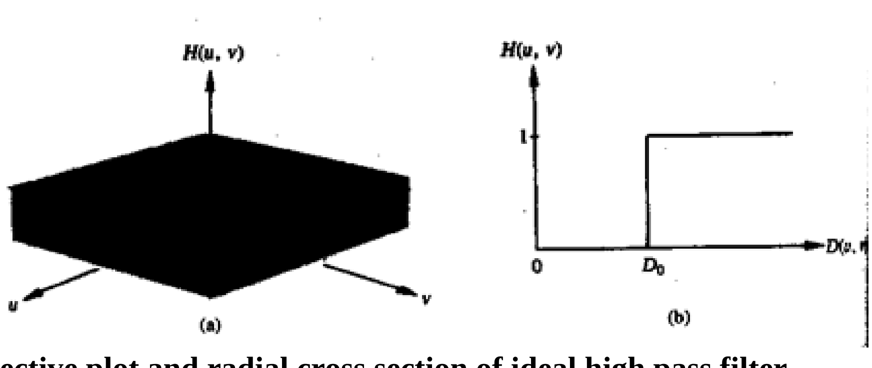


Fig. Perspective plot and radial cross section of ideal high pass filter

Butterworth filter:

The transfer function of the Butterworth high pass filter (BHPF) of order n and with cutoff frequency locus at a distance Do from the origin is defined by the relation

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

Figure below shows a perspective plot and cross section of the BHPF function. Note that when D (u, v) = Do, H (u, v) is down to ½ of its maximum value. As in the case of the Butterworth lowpass filter, common practice is to select the cutoff frequency locus at points for which H (u,
v) is down to 1/√2 of its maximum value.

$$H(u, v) = \frac{1}{1 + [\sqrt{2} - 1][D_0/D(u, v)]^{2n}}$$
$$= \frac{1}{1 + 0.414[D_0/D(u, v)]^{2n}}.$$

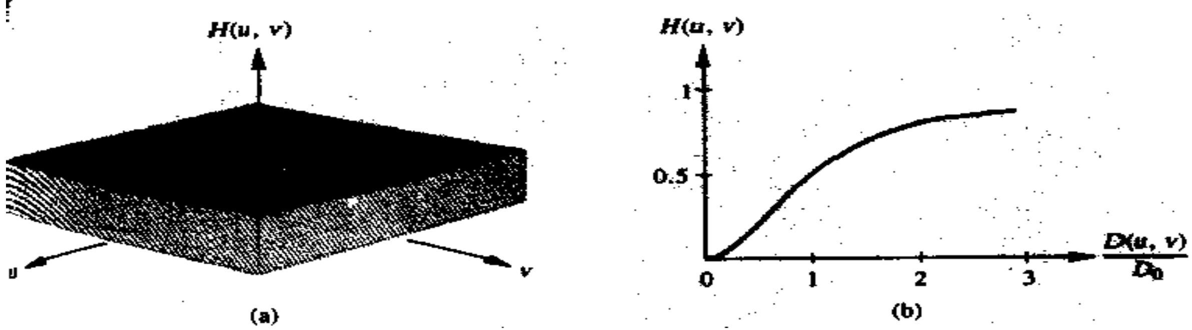


Fig. Perspective plot and radial cross section for Butterworth High Pass Filter with n = 1

Gaussian Highpass Filters:

The transfer function of the Gaussian highpass filter (GHPF) with cutoff frequency locus at a distance Do from the origin is given by

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

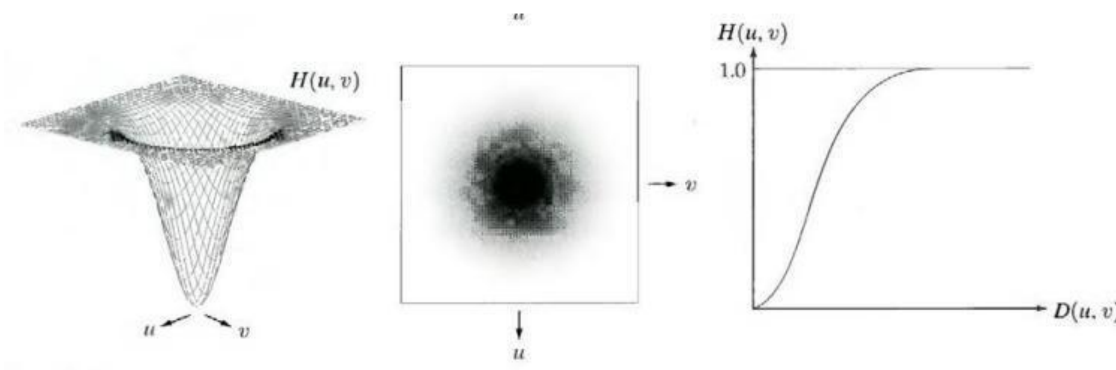


Fig. Perspective plot, image representation, and cross section of a typical Gaussian high pass filter.

Homomorphic filtering:

The illumination-reflectance model can be used to develop a frequency domain procedure for improving the appearance of an image by simultaneous gray-level range compression and contrast enhancement. An image f(x, y) can be expressed as the product of illumination and reflectance components:

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

Equation above cannot be used directly to operate separately on the frequency components of illumination and reflectance because the Fourier transform of the product of two functions is not separable; in other words,

$$\Im\{f(x, y)\} \neq \Im\{i(x, y)\}\Im\{r(x, y)\}.$$

Suppose, however, that we define

$$\begin{aligned} z(x, y) &= \ln f(x, y) \\ &= \ln i(x, y) + \ln r(x, y). \end{aligned}$$

Then

$$\begin{aligned} \Im\{z(x, y)\} &= \Im\{\ln f(x, y)\} \\ &= \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\} \end{aligned}$$

or

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Where F_i (u, v) and F_r (u, v) are the Fourier transforms of ln i(x, y) and ln r(x, y), respectively. If we process Z (u, v) by means of a filter function H (u, v) then, from

$$
\begin{aligned}
S(u, v) &= H(u, v)Z(u, v) \\
&= H(u, v)F_i(u, v) + H(u, v)F_r(u, v)
\end{aligned}
$$

Where S (u, v) is the Fourier transform of the result. In the spatial domain,

$$
\begin{aligned}
s(x, y) &= \Im^{-1}\{S(u, v)\} \\
&= \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\}.
\end{aligned}
$$

By letting

$$i'(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\}$$

and

$$r'(x, y) = \Im^{-1}\{H(u, v)F_r(u, v)\},$$

Now we have

$$s(x, y) = i'(x, y) + r'(x, y).$$

Finally, as z (x, y) was formed by taking the logarithm of the original image f (x, y), the inverse (exponential) operation yields the desired enhanced image, denoted by g(x, y); that is,

$$
\begin{aligned}
g(x, y) &= e^{s(x, y)} \\
&= e^{i'(x, y)} \cdot e^{r'(x, y)} \\
&= i_0(x, y)r_0(x, y)
\end{aligned}
$$

where

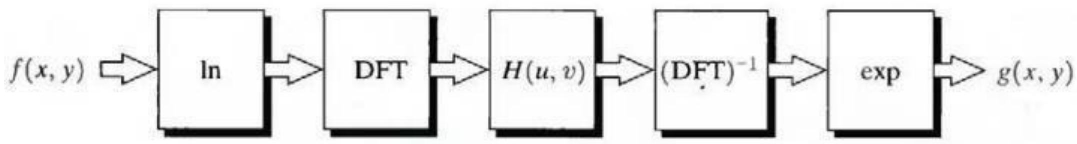
$$i_0(x, y) = e^{i'(x, y)}$$

Fig. Homomorphic filtering approach for image enhancement

And

$$r_0(x, y) = e^{r'(x, y)}$$

are the illumination and reflectance components of the output image. The enhancement approach using the foregoing concepts is summarized in Fig. 9.1. This method is based


$f(x, y)$
ln
DFT
$H(u, v)$
$(\mathrm{DFT})^{-1}$
exp
$g(x, y)$

on a special case of a class of systems known as homomorphic systems. In this particular application, the key to the approach is the separation of the illumination and reflectance components achieved. The homomorphic filter function H (u, v) can then operate on these components separately.

The illumination component of an image generally is characterized by slow spatial variations, while the reflectance component tends to vary abruptly, particularly at the junctions of dissimilar objects. These characteristics lead to associating the low frequencies of the Fourier transform of the logarithm of an image with illumination and the high frequencies with reflectance. Although these associations are rough approximations, they can be used to advantage in image enhancement.

UNIT-III IMAGE RESTORATION AND FILTERING

Image Restoration

Objective of image restoration

- To recover a distorted image to the original form based on idealized models**.**

The distortion is due to

- Image degradation in sensing environment e.g. random atmospheric turbulence
- Noisy degradation from sensor noise.
- Blurring degradation due to sensors

  camera motion or out-of-focus
- Geometric distortion

  earth photos taken by a camera in a satellite
- **Enhancement**
  - Concerning the extraction of image features
  - Difficult to quantify performance
  - Subjective; making an image "look better
- **Restoration**
  - Concerning the restoration of degradation
  - Performance can be quantified
  - Objective; recovering the original image

Image Restoration/Degradation Model

The Fig. below shows, the degradation process is modeled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y). Given g(x, y), some knowledge about the degradation function H, and some knowledge about the additive noise term η(x, y), the objective of restoration is to obtain an estimate *f*(x, y) of the original image. the estimate should be as close as possible to the original input image and, in general, the more we know about H and η, the closer *f*(x, y) will be to f(x, y).

The degraded image is given in the spatial domain by

$$\mathbf{g(x, y) = h(x, y) * f(x, y) + \eta(x, y)}$$

where h (x, y) is the spatial representation of the degradation function and, the symbol * indicates convolution. Convolution in the spatial domain is equal to multiplication in the frequency domain, hence

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

where the terms in capital letters are the Fourier transforms of the corresponding terms in above equation.

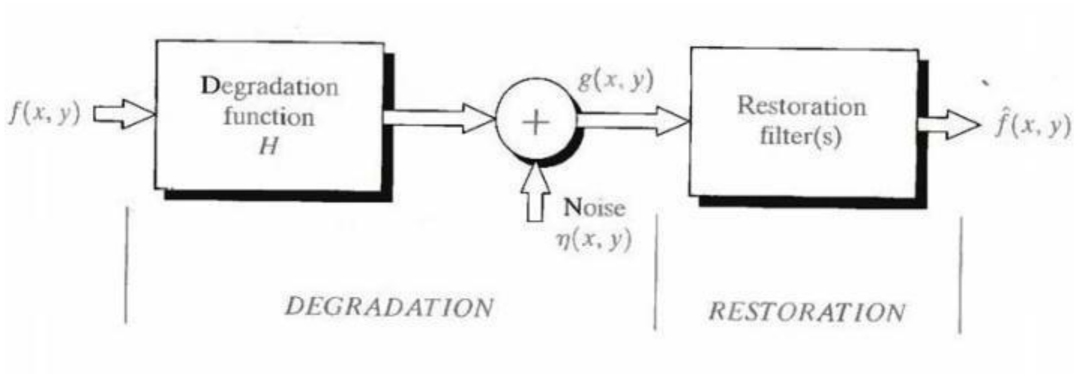


Fig: Model of Image Degradation/Restoration Process

Noise Models

The following are among the most common PDFs found in image processing applications.

Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called normal) noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best.

The PDF of a Gaussian random variable, z, is given by

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

where z represents gray level, μ is the mean of average value of z, and a σ is its standard deviation. The standard deviation squared, σ^2, is called the variance of z. A plot of this function is shown in Fig. 5.10. When z is described by Eq. (1), approximately 70% of its values will be in the range [(μ - σ), (μ +σ)], and about 95% will be in the range [(μ - 2σ), (μ + 2σ)].

Rayleigh noise

The PDF of Rayleigh noise is given by

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

The mean and variance of this density are given by

μ = a +*ƒ*~~Mb/4~~

$\sigma^2 = b(4 - \Pi)/4$

Erlang (Gamma) noise

The PDF of Erlang noise is given by

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

where the parameters are such that a > 0, b is a positive integer, and "!" indicates factorial. The mean and variance of this density are given by

$\mu = b / a$ $\sigma^2 =$ b

/ a^2

Exponential noise

The PDF of exponential noise is given by

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

The mean of this density function is given by

$\mu = 1 / a$ $\sigma^2 = 1$

$/ a^2$

This PDF is a special case of the Erlang PDF, with b = 1.

Uniform noise

The PDF of uniform noise is given by

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

The mean of this density function is given by

$\mu = a + b /2$

$\sigma^2 = (b - a)^2 / 12$

Impulse (salt-and-pepper) noise

The PDF of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

If b > a, gray-level b will appear as a light dot in the image. Conversely, level a will appear like a dark dot. If either P_a or P_b is zero, the impulse noise is called unipolar. If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image. For this reason, bipolar impulse noise also is called salt-and-pepper noise. Shot and spike noise also are terms used to refer to this type of noise.

Restoration in the presence of noise by Spatial Filtering

If the degradation present in an image is only due to noise, then,

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

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

The restoration filters used in this case are,

1. Mean filters
2. Order static filters and
3. Adaptive filters

1.Mean Filters

There are four types of mean filters. They are

(i) Arithmetic mean filter

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window of size m X n, centered at point (x, y).The arithmetic mean filtering process computes the average value of the corrupted image g(x, y) in the area defined by S_{xy}. The value of the restored image *f* at any point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by S_{xy}. In other words

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

This operation can be implemented using a convolution mask in which all coefficients have value 1/mn

(ii) Geometric mean filter

An image restored using a geometric mean filter is given by the expression

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

Here, each restored pixel is given by the product of the pixels in the subimage window, raised to the power 1/mn. A geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

(iii) Harmonic mean filter

The harmonic mean filtering operation is given by the expression

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

The harmonic mean filter works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

(iv) Contra harmonic mean filter

The contra harmonic mean filtering operation yields a restored image based on the expression

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

where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q, the filter eliminates pepper noise. For negative values of Q it eliminates salt noise. It cannot do both simultaneously. Note that the contra harmonic filter reduces to arithmetic mean filter if Q = 0, and to the harmonic mean filter if Q = -1.

2.Order-statistics Filters

There are four types of Order-Statistic filters. They are

(i) Median filters

The best-known order-statistics filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel:

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\operatorname{median}}\{g(s, t)\}.$$

The original value of the pixel is included in the computation of the median

(ii) Max and min filters

Although the median filter is by far the order-statistics filler most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100^{th} percentile results in the so-called max filter, given by

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

This filter is useful for finding the brightest points in an image The 0^{th} percentile filter is the min filter.

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

This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

(iii) Midpoint filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

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

Note that this filter combines order statistics and averaging. This filter works best for randomly distributed noise, like Gaussian or uniform noise.

(iv) Alpha - trimmed mean filter

It is a filter formed by deleting the d/2 lowest and the d/2 highest gray-level values of g(s, t) in the neighborhood S_{xy}. Let g_r (s, t) represent the remaining mn - d pixels. A filter formed by averaging these remaining pixels is called an alpha-trimmed mean filter:

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

where the value of d can range from 0 to mn - 1. When d = 0, the alpha- trimmed filter reduces to the arithmetic mean filter. If d = (mn - l)/2, the filter becomes a median filter. For other values of d, the alpha-trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

3. Adaptive Filters

Adaptive filters are filters whose behavior changes based on statistical characteristics of the image inside the filter region defined by the m X n rectangular window S_{xy}.

(i) Adaptive, local noise reduction filter:

The simplest statistical measures of a random variable are its mean and variance. These are reasonable parameters on which to base an adaptive filler because they are quantities closely related to the appearance of an image. The mean gives a measure of average gray level in the region over which the mean is computed, and the variance gives a measure of average contrast in that region.

1. If σ^2_η is zero, the filler should return simply the value of g (x, y). This is the trivial, zero-noise case in which g (x, y) is equal to f (x, y).

2. If the local variance is high relative to σ^2_η the filter should return a value close to g (x, y). A

high local variance typically is associated with edges, and these should be preserved.

3. If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in S_{xy}. This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced simply by averaging.

Adaptive local noise filter is given by,

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right].$$

(ii) Adaptive median filter:

The output of the filter is a single value used to replace the value of the pixel at (x, y), the particular point on which the window S_{xy} is centered at a given time.

Consider the following notation:

z_{min} = minimum gray level value in S_{xy}
z_{max} = maximum gray level value in S_{xy}
z_{mcd} = median of gray levels in S_{xy}
z_{xy} = gray level at coordinates (x, y)

S_{max} = maximum allowed size of S_{xy}.

The adaptive median filtering algorithm works in two levels, denoted level A and level B, as follows:

Level A: $A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$ If A1 > 0 AND A2 < 0, Go to level B Else increase the window size

If window size $\leq S_{max}$ repeat level A
Else output z_{xy}

Level B: $B1 = z_{xy} - z_{min}$

$\mathbf{B2 = z_{xy} - z_{max}}$

If B1> 0 AND B2 < 0, output zxy
Else output z_{med}

Periodic noise reduction by frequency domain filtering

- Pure sine wave
  -Appear as a pair of impulse (conjugate) in the frequency domain

$$f(x,y) = A\sin(u_0 x + v_0 y)$$

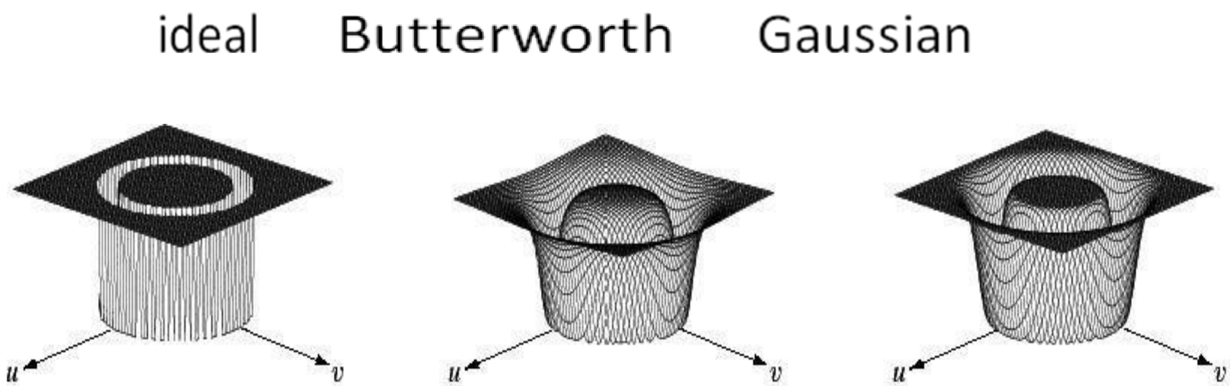
$$F(u,v) = -j\frac{A}{\ }\left[\delta\left(u - \frac{u_0}{\ }, v - \frac{v_0}{\ }\right) - \delta\left(u + \frac{u_0}{\ }, v + \frac{v_0}{\ }\right)\right]$$

$$2 \Big| \Big\lfloor \quad 2\pi \quad 2\pi \quad\quad 2\pi \quad 2\pi \Big\rfloor \Big|$$

1. Band reject filters
2. Band pass filters
3. Notch filters
4. Optimum notch filtering

1. Band reject filters

- Reject an isotropic frequency

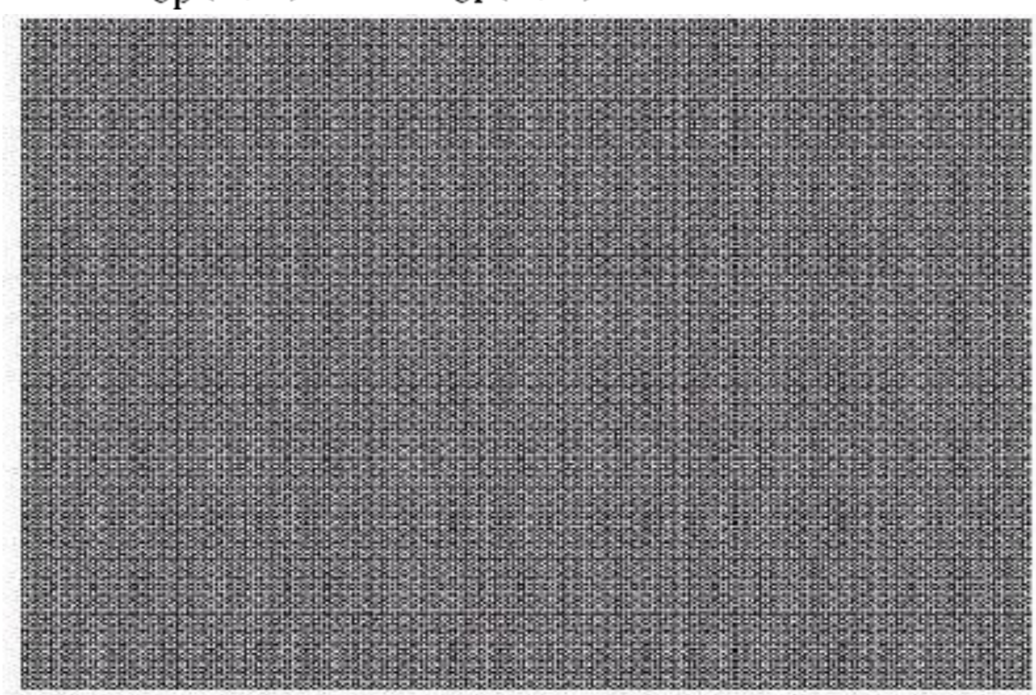


a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

2. Band pass filters

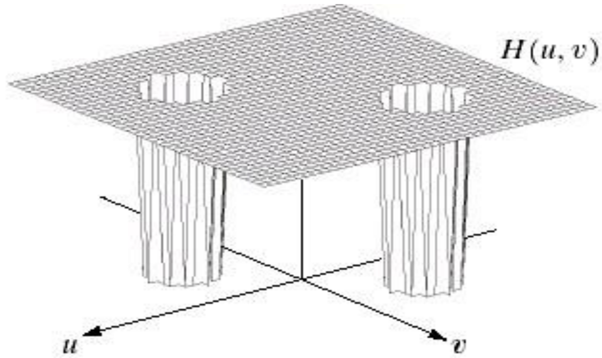
- $H_{bp}(u,v)=1- H_{br}(u,v)$

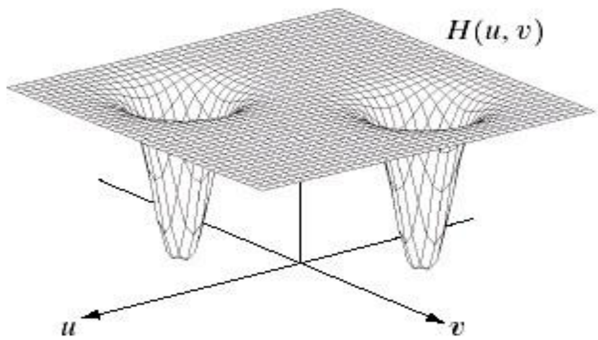


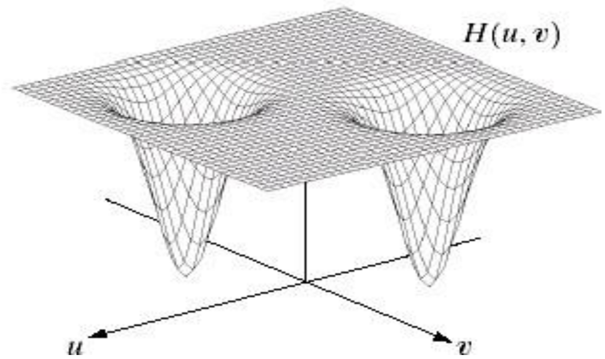
$$\Im^{-1}\{G(u,v)H_{bp}(u,v)\}$$

3. Notch filters

- Reject(or pass) frequencies in predefined neighborhoods about a center frequency







4. Optimum notch filtering

The Fourier transform of the interference noise pattern is given by the expression

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

<u>Image Filtering: Linear, Position-invariant degradations</u>

Properties of the degradation function H

- Linear system

$$H[af_1(x,y)+bf_2(x,y)]=aH[f_1(x,y)]+bH[f_2(x,y)]$$

- Position(space)-invariant system

$$H[f(x,y)]=g(x,y) \Leftrightarrow H[f(x-a, y-b)]=g(x-a, y-b)$$

- c.f. 1-D signal
  LTI (linear time-invariant system)

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

$$g(x,y)=H[f(x,y)]=H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha,y-\beta)\,d\alpha d\beta\right]$$

$$g(x,y)=H[f(x,y)]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H\left[f(\alpha,\beta)\delta(x-\alpha,y-\beta)\right]d\alpha d\beta$$

$$g(x,y) = H[f(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta) H\left[\delta(x-\alpha, y-\beta)\right] d\alpha d\beta$$

$$h(x,\alpha,y,\beta) = H\left[\delta(x-\alpha, y-\beta)\right]$$

$$H\left[\delta(x-\alpha, y-\beta)\right] = h(x-\alpha, y-\beta)$$

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

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

- Linear system theory is ready
- Non-linear, position-dependent system
  - -May be general and more accurate
  - -Difficult to solve computationally
- Image restoration: find H(u,v) and apply inverse process
  - -Image deconvolution

Estimating the degradation function

1. Estimation by Image observation
2. Estimation by experimentation
3. Estimation by modeling

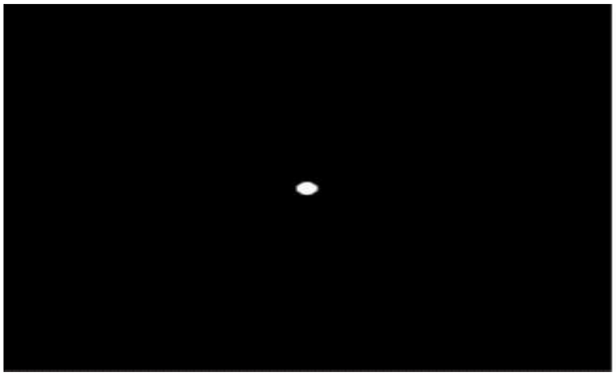
1. Estimation by Image observation

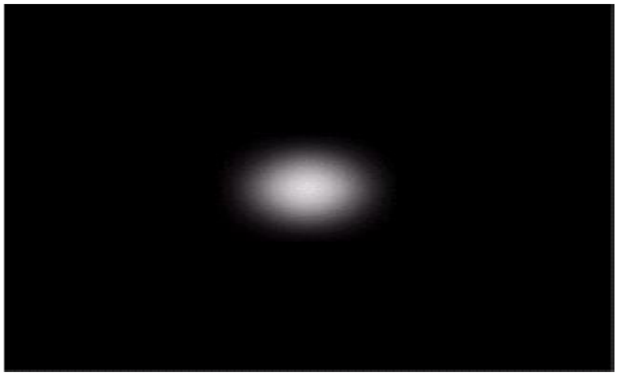
- Take a window in the image
  - -Simple structure
  - -Strong signal content
- Estimate the original image in the window

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

2. Estimation by experimentation

- If the image acquisition system is ready
- Obtain the impulse response





impulse Impulse response

3. Estimation by modeling

- Derive a mathematical model

Ex. Motion of image

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

$$G(u,v) = F(u,v)\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}dt$$

<u>Inverse Filtering</u>

The simplest approach to restoration is direct inverse filtering, where F (u, v), the transform of the original image is computed simply by dividing the transform of the degraded image, G (u, v), by the degradation function

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

The divisions are between individual elements of the functions.

But G (u, v) is given by

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

Hence

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

It tells that even if the degradation function is known the undegraded image cannot be recovered [the inverse Fourier transform of F(u, v)] exactly because N(u, v) is a random function whose Fourier transform is not known.

If the degradation has zero or very small values, then the ratio N(u, v)/H(u, v) could easily dominate the estimate F(u, v).

<u>Minimum Mean Square Error(Wiener)Filtering</u>

The inverse filtering approach makes no explicit provision for handling noise. This approach incorporates both the degradation function and statistical characteristics of noise into the restoration process. The method is founded on considering images and noise as random processes, and the objective is to find an estimate *f* of the uncorrupted image f such that the mean square error between them is minimized. This error measure is given by

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

where E{•} is the expected value of the argument. It is assumed that the noise and the image are uncorrelated; that one or the other has zero mean; and that the gray levels in the estimate are a linear function of the levels in the degraded image. Based on these conditions, the minimum of the error function is given in the frequency domain by the expression

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v)$$
$$= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$
$$= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

where we used the fact that the product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. This result is known as the Wiener filter, after N. Wiener [1942], who first proposed the concept in the year shown. The filter, which consists of the terms inside the brackets, also is commonly referred to as the minimum mean square error filter or the least square error filter. The Wiener filter does not have the same problem as the inverse filter with zeros in the degradation function, unless both H(u, v) and S_η(u, v) are zero for the same value(s) of u and v.

The terms in above equation are as follows: H

(u, v) = degradation function

H*(u, v) = complex conjugate of H (u, v)

$|H(u,v|^2$ = H*(u, v)* H (u, v)

S_η (u, v) = $|N(u,v)^2$ = power spectrum of the noise

S_f (u, v) = $|F(u,v)^2$ = power spectrum of the undegraded image.

As before, H (u, v) is the transform of the degradation function and G (u, v) is the transform of the degraded image. The restored image in the spatial domain is given by the inverse Fourier transform of the frequency-domain estimate F (u, v). Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter.

When we are dealing with spectrally white noise, the spectrum $|N(u,v|^2$ is a constant, which simplifies things considerably. However, the power spectrum of the undegraded image seldom is known. An approach used frequently when these quantities are not known or cannot be estimated is to approximate the equation as

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

where K is a specified constant.

Constrained Least Square Filtering

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2}\right] G(u,v)$$

- P(u,v) is the fourier transform of the Laplacian operator Constraint:
- $|g - H|^2 = |\eta|^2$
- R(u,v) = G(u,v) – H(u,v)
- Adjust γ from the constraint – by Newton-Raphson root-finding

- In the Fourier domain, the constrained least squares filter becomes:

$$F(k,l) = \frac{H^*(k,l)}{|H(k,l)|^2 + \lambda |Q(k,l)|^2} G(k,l)$$

- Keep always in mind to zero-pad the images properly.

a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

Geometric Mean Filter

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha} \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta\left[\frac{S_\eta(u,v)}{S_f(u,v)}\right]}\right]^{1-\alpha} G(u,v)$$

- Geometric mean filter is quite useful when implementing restoration filters because it represents a family of filters combined into a single expression.

UNIT-IV IMAGE PROCESSING

Color Image Fundamentals

The use of color is important in image processing because

•Color is a powerful descriptor that simplifies object identification and extraction.

- Humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray.

  Color image processing is divided into two major areas:

•Full color processing :images are acquired with a full color sensor, such as a color TV camera or color scanner.

•Pseudo color processing: The problem is one of assigning a color to a particular monochrome intensity or range of intensities.

- Physical phenomenon

Physical nature of color is known

- Psysio-psychological phenomenon

How human brain perceive and interpret color?

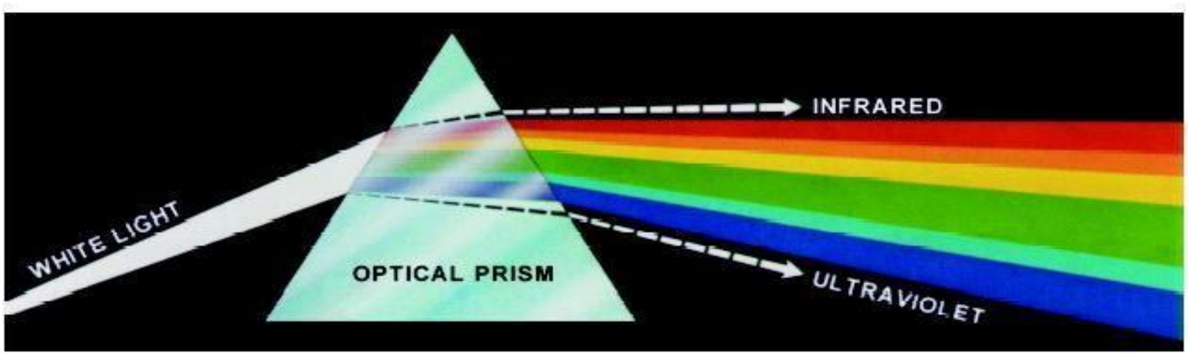


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Visible light:

Chromatic light span the electromagnetic spectrum (EM) from 400 to 700 nm

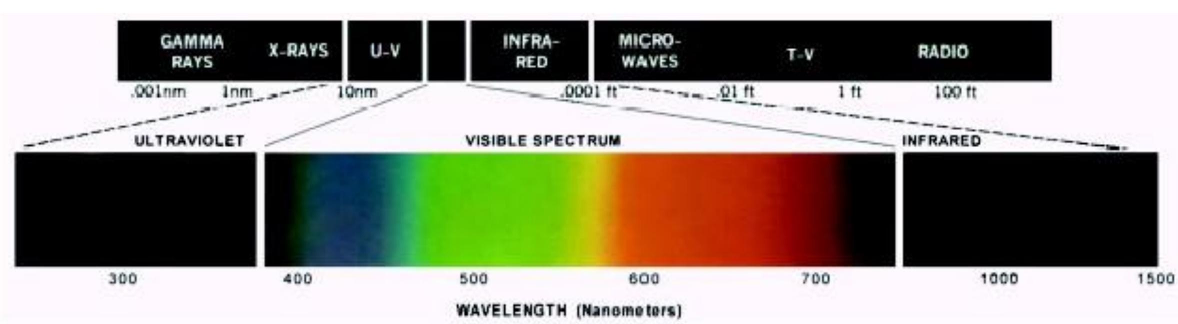


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

- The color that human perceive in an object = the light reflected from the object.

Physical quantities to describe a chromatic light source

Radiance: total amount of energy that flow from the light source, measured in watts (W)
Luminance: amount of energy an observer *perceives* from a light source, measured in lumens (lm)

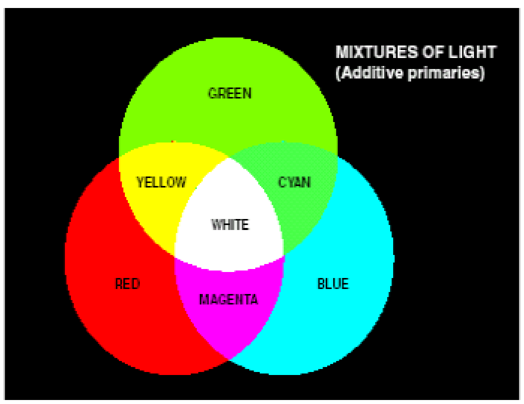
- Far infrared light: high radiance, but 0 luminance

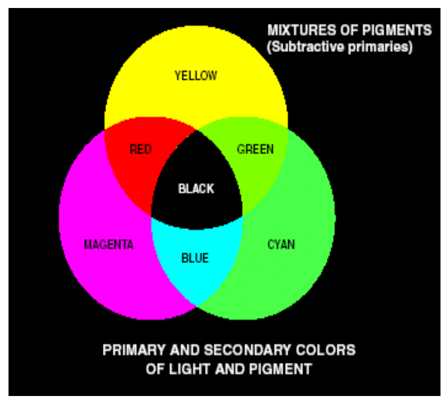
Brightness: subjective descriptor that is hard to measure, similar to the achromatic notion of intensity.
How human eyes sense light?

- 6~7M Cones are the sensors in the eye
- 3 principal sensing categories in eyes
    - Red light 65%, green light 33%, and blue light 2%

<u>Primary and secondary colors</u>

- In 1931, CIE(International Commission on Illumination) defines specific wavelength values to the primary colors
    - B = 435.8 nm, G = 546.1 nm, R = 700 nm
    - However, we know that no single color may be called red, green, or blue
- Secondary colors: G+B=Cyan, R+G=Yellow, R+B=Magenta

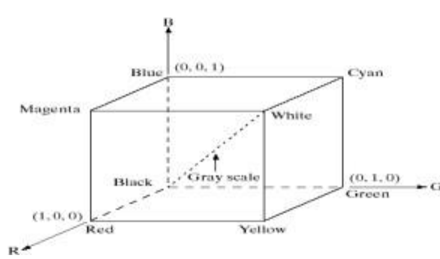




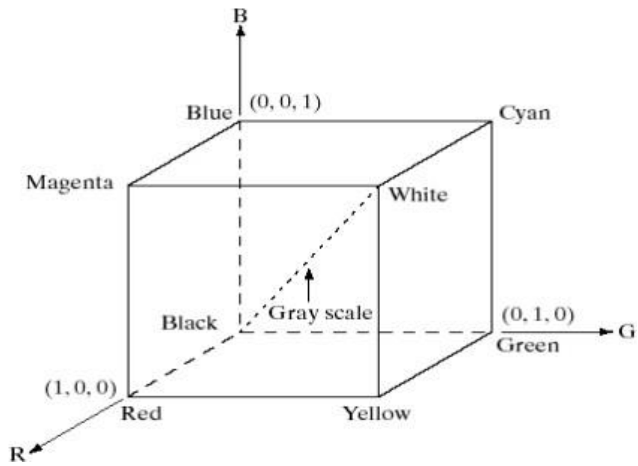
Color Models

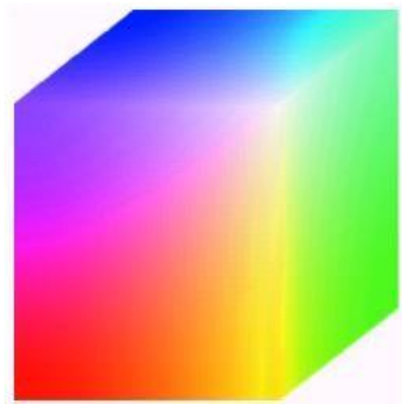
- Color model, color space, color system
    - Specify colors in a standard way
    - A coordinate system that each color is represented by a single point
- RGB model
- CYM model
- CYMK model
- HSI model
- Suitable for hardware or applications

RGB color model



- Pixel depth: the number of bits used to represent each pixel in RGB space
- Full-color image: 24-bit RGB color image
    - (R, G, B) = (8 bits, 8 bits, 8 bits)

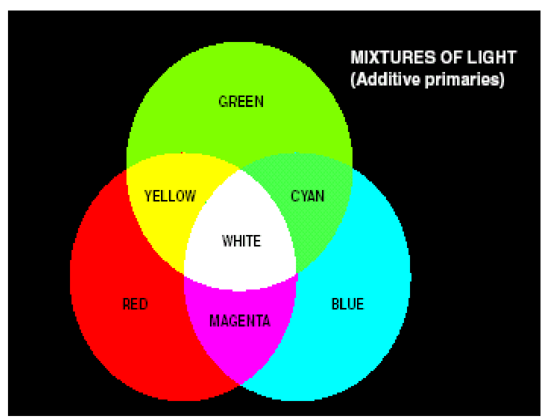




<u>CMY model (+Black = CMYK)</u>

- CMY: secondary colors of light, or primary colors of pigments
- Used to generate hardcopy output

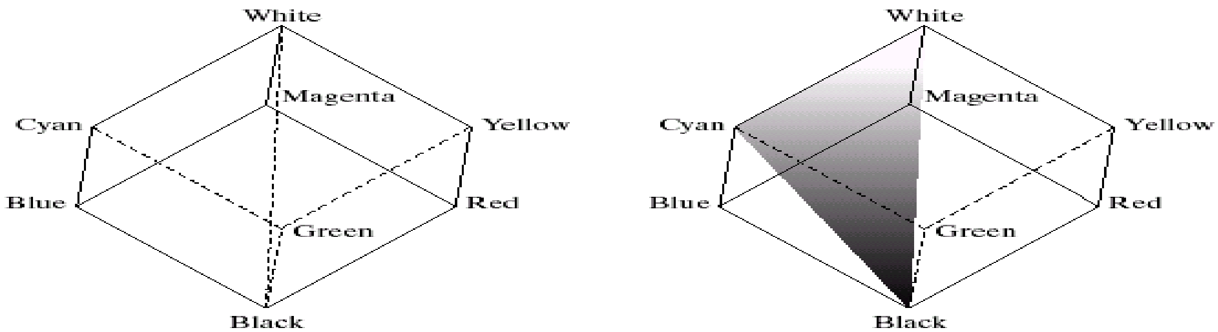
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

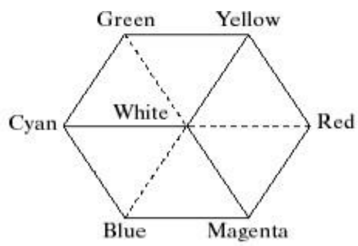


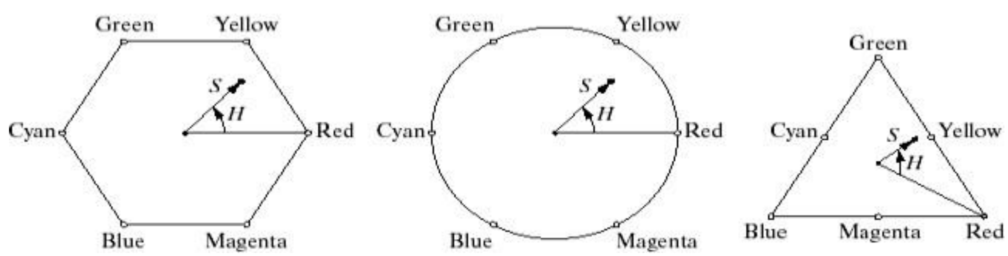
<u>HSI color model</u>

- Will you describe a color using its R, G, B components?
- Human describe a color by its hue, saturation, and brightness
    - Hue : color attribute
    - Saturation: purity of color (white->0, primary color->1)
    - Brightness: achromatic notion of intensity

RGB -> HSI model



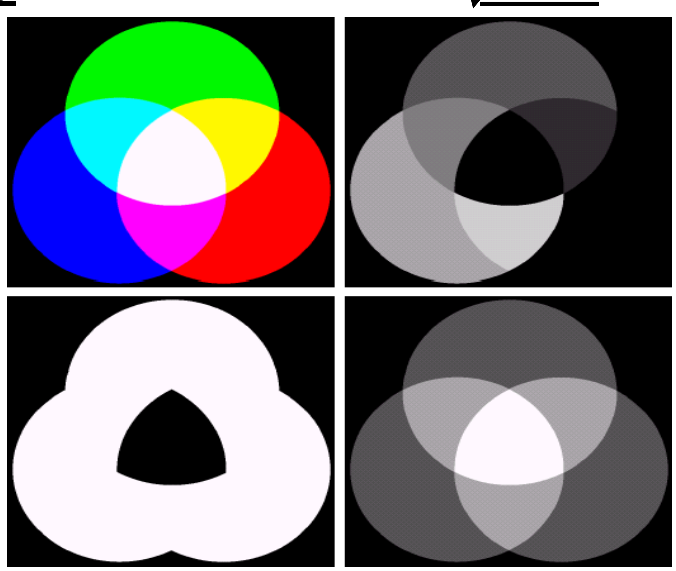




HSI component images

Hue

R,G,B



saturation

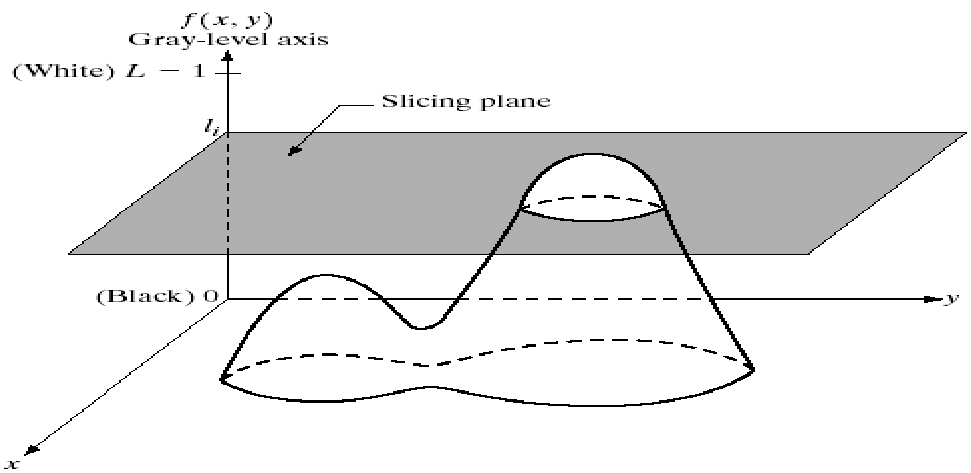
intensity

<u>Pseudo-color image processing</u>

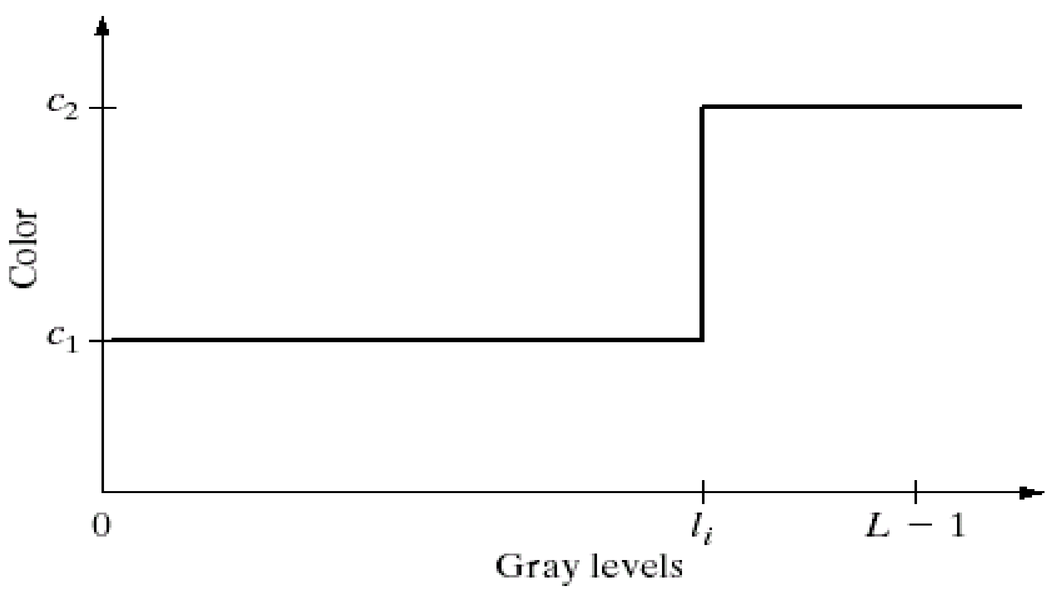
- Assign colors to gray values based on a specified criterion
- For human visualization and interpretation of gray-scale events
- Intensity slicing
- Gray level to color transformations.

1. Intensity slicing

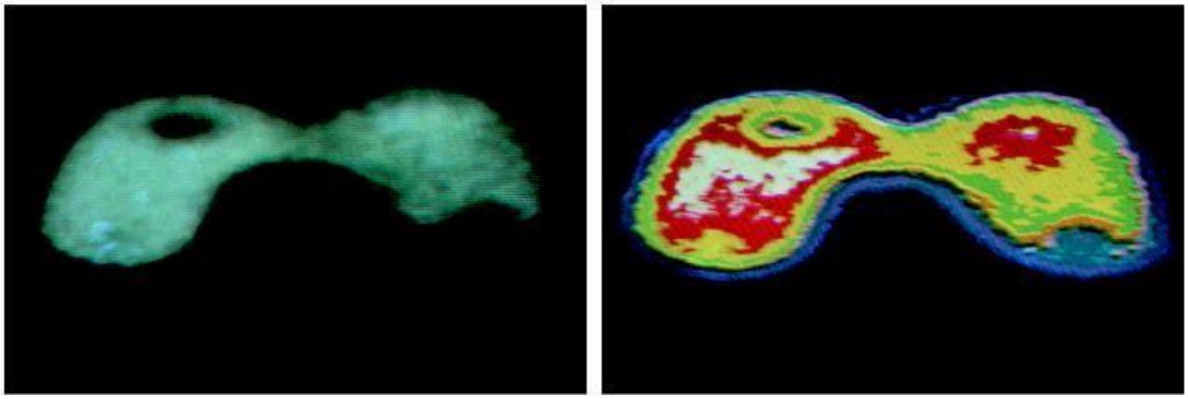
- 3-D view of intensity image.



Alternative representation of intensity slicing



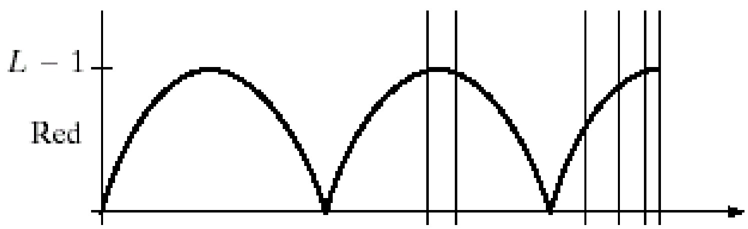
Application of Intensity slicing



Radiation test pattern → 8 color regions

2. Gray level to color transformation

- General Gray level to color transformation



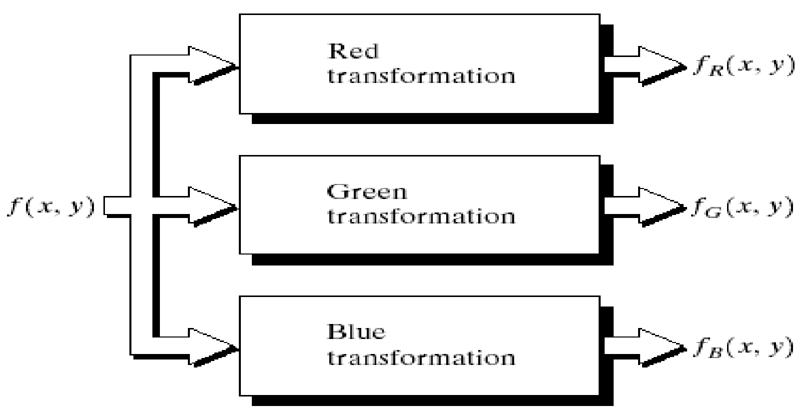


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

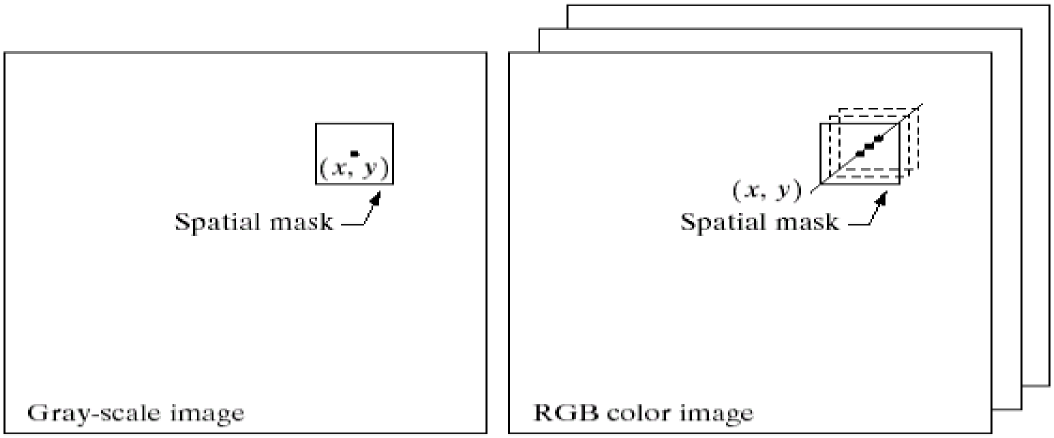
<u>Basics of Full-Color Image Processing</u>

Color pixel

- A pixel at (x,y) is a vector in the color space
    - RGB color space

$$\mathbf{c}(x, y) = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

Example: spatial mask



How to deal with color vector?

- Per-color-component processing
    - Process each color component
- Vector-based processing
    - Process the color vector of each pixel
- When can the above methods be equivalent?
    - Process can be applied to both scalars and vectors
    - Operation on each component of a vector must be independent of the other component

Color Transformations

- Similar to gray scale transformation

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

- Color transformation

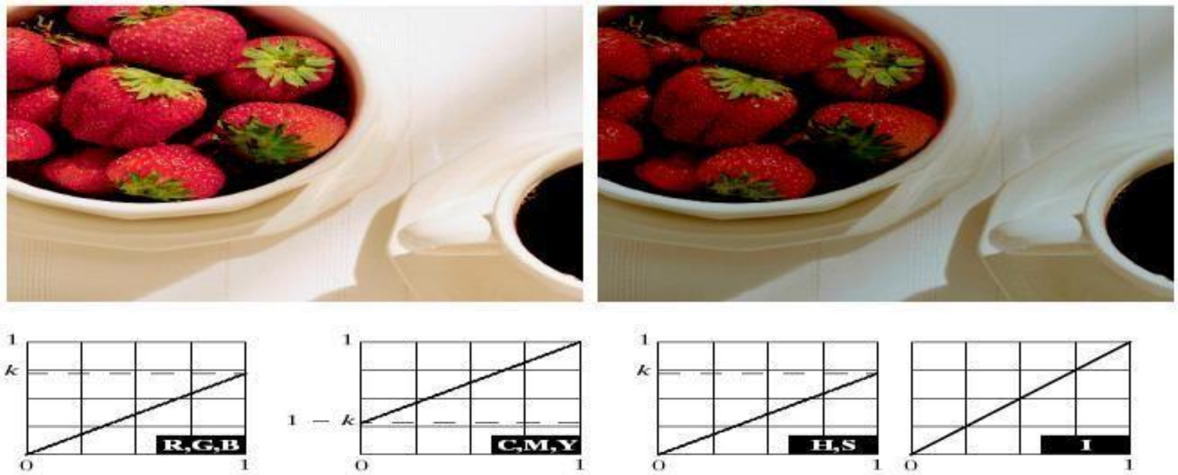
$$s_i = T_i(r_1, r_2, ..., r_n), \quad i = 1,2,..., n$$

Use which color model in color transformation?

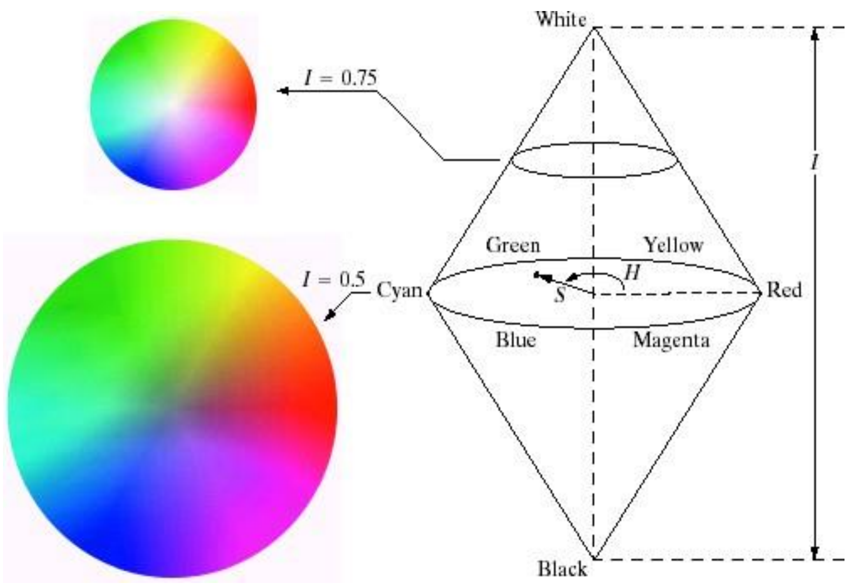
- RGB ⟺ CMY(K) ⟺ HSI
- Theoretically, any transformation can be performed in any color model
- Practically, some operations are better suited to specific color model

Example: modify intensity of a color image

- Example: *g(x,y)=k f(x,y)*, 0<k<1
- HSI color space
    - Intensity: $s_3 = k\, r_3$
    - Note: transform to HSI requires complex operations
- RGB color space
    - For each R,G,B component: $s_i = k\, r_i$
- CMY color space
    - For each C,M,Y component:
    - $s_i = k\, r_i + (1-k)$



Problem of using Hue component



Color Complement

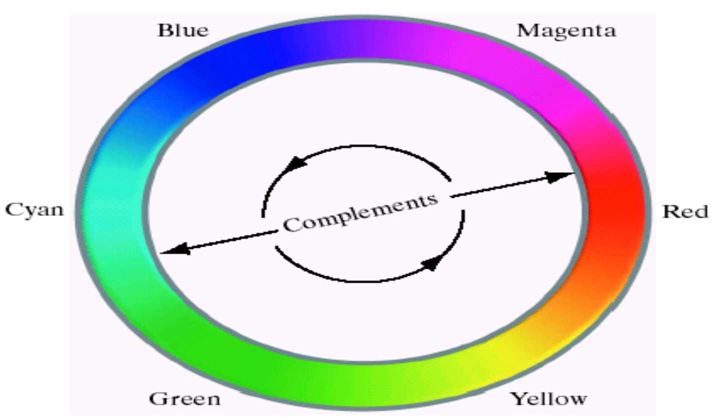
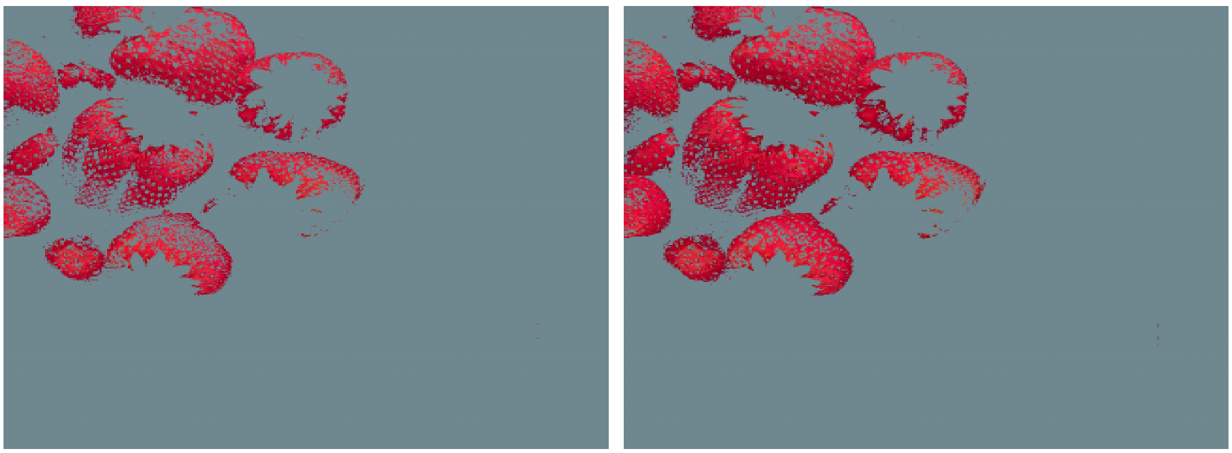


FIGURE 6.32 Complements on the color circle.

Implementation of color slicing

- High light a specific range of colors in an image.

$$s_i = \begin{cases} 0.5 & if \left[\left| r_j - a_j \right| > \dfrac{W}{2} \right]_{any 1 \le j \le n} \\ r_i & otherwise \end{cases}, \qquad i = 1,2,\ldots, n$$



a b

FIGURE 6.34 Color slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

Tone and Color Correction

- In conjunction with digital cameras, flatbed scanners, and inkjet printers, they turn a personal computer into a digital *Darkroom*

- The Colors of monitor should represent accurately any digitally scanned source images, as well as the final printed out

The Model of choice for many color management system (CMS) is

***CIE L*a*b** (also called CIELAB) model**

L*a*b Color Components:

$$L^* = 116 \cdot h\left(\frac{Y}{Y_w}\right) - 16$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_w}\right) \right]$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_w}\right) \right]$$

$$where:\ h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.7.87q + 16/116 & q \le 0.008856 \end{cases}$$

Middle-key Image

FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

High-key Image

FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Low-key Image

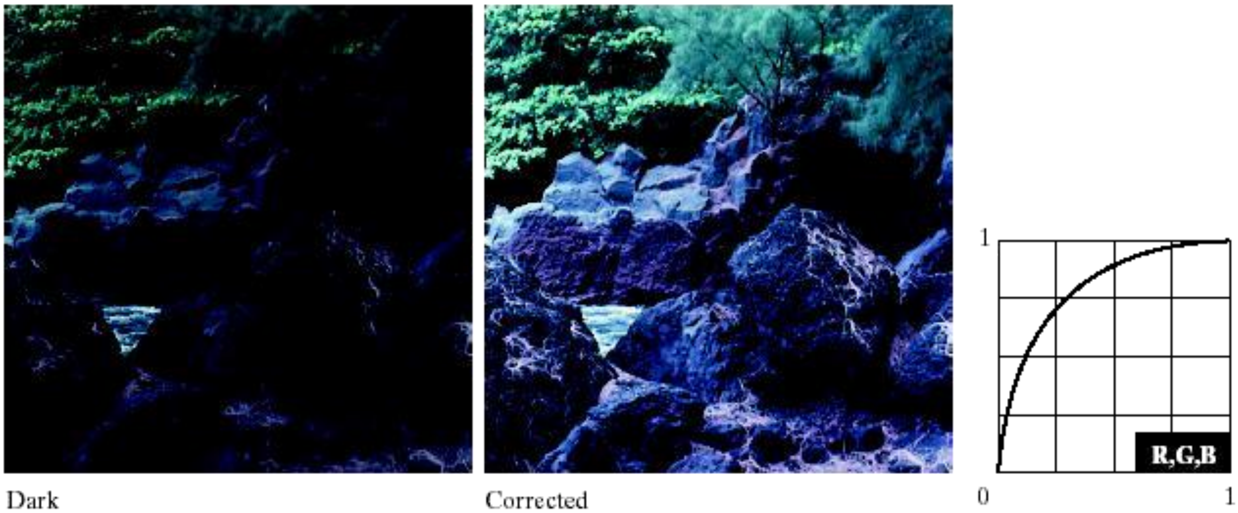
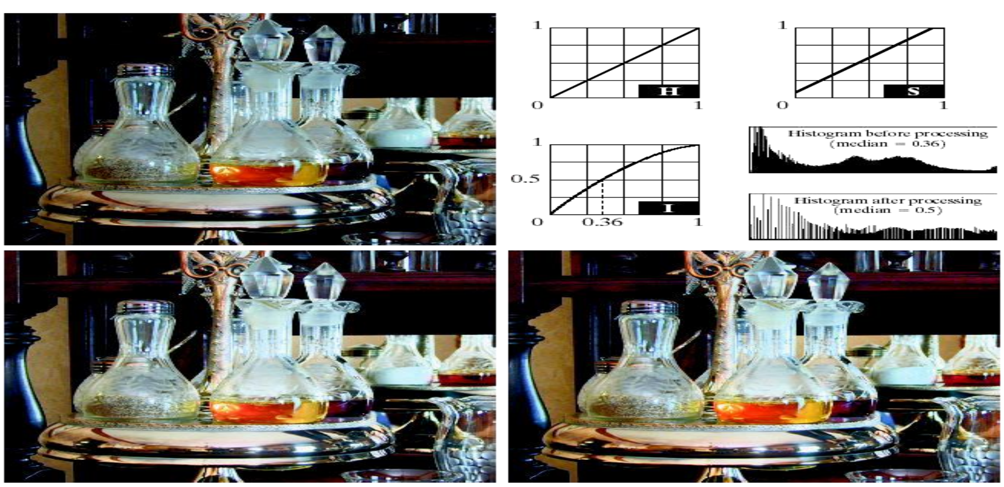


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Histogram Processing



a b
c d
FIGURE 6.37 Histogram equalization (followed by saturation adjustment) in the HSI color space.

<u>Color image Smoothing and Sharpening</u>

Averaging:

$$\bar{\mathbf{c}}(x, y) = \frac{1}{K} \sum_{(x,y)\in S_{xy}} \mathbf{c}(x, y)$$

$$\bar{\mathbf{c}}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{(x,y)\in S_{xy}} R(x, y) \\ \frac{1}{K} \sum_{(x,y)\in S_{xy}} G(x, y) \\ \frac{1}{K} \sum_{(x,y)\in S_{xy}} B(x, y) \end{bmatrix}$$

$$\left\lfloor K_{(x,y)\in S_{xy}} \right\rfloor$$

Sharpening

The Laplacian of Vector c :

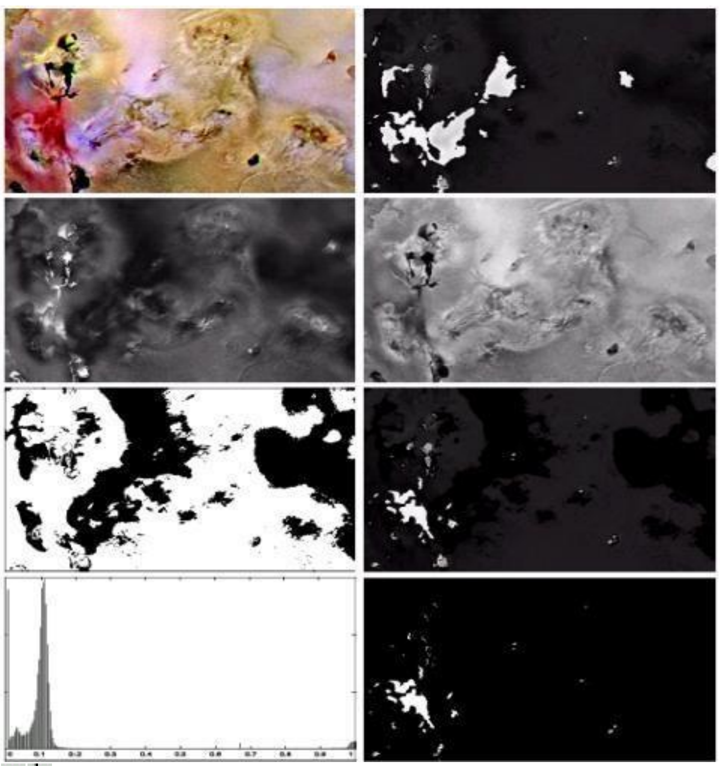
$$\nabla^2\left[\mathbf{c}(x,y)\right]=\begin{bmatrix}\nabla^2 R(x,y)\\ \nabla^2 G(x,y)\\ \nabla^2 B(x,y)\end{bmatrix}$$

<u>Color Segmentation</u>

Segmentation is a process that partitions an image into regions.

- Segmentation in HIS Color Space
- Segmentation in RGB Vector Space
- Color Edge Detection

Color Segmentation: in HIS Color Space



a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

Color Segmentation: in RGB Vector Space

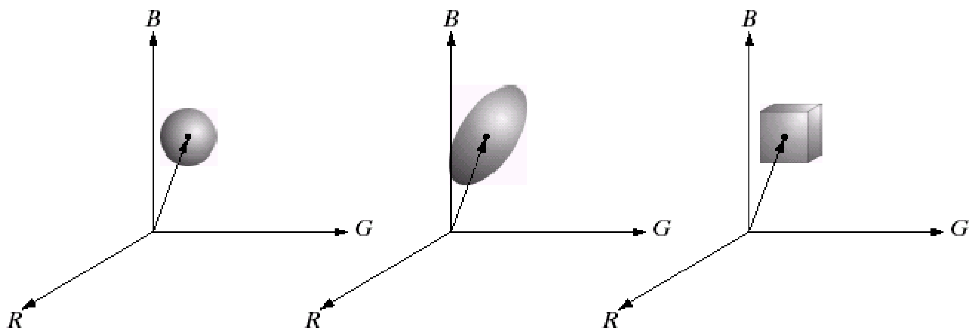
z is similar to a if the distance between them is less than a specified threshold.

Euclidian Distance

$$D(\mathbf{z},\mathbf{a}) = \|\mathbf{z}-\mathbf{a}\| = \left[(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\right]^{1/2}$$

$$= \left[(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2 \right]^{1/2}$$

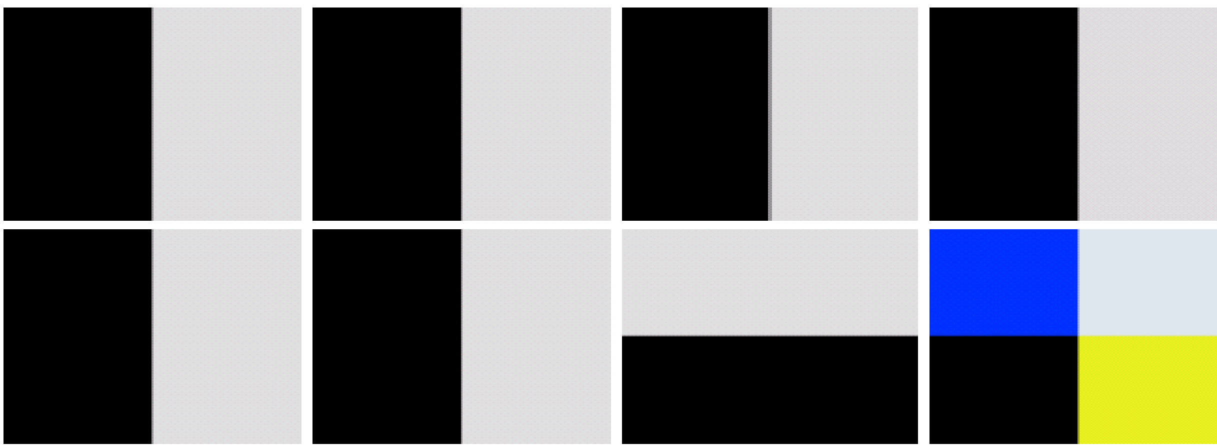
Generalized form: $$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1} (\mathbf{z}-\mathbf{a})\right]^{1/2}$$



a b c

FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.

Color Segmentation: Color Edge Detection



a b c d
e f g h

FIGURE 6.45 (a)–(c) R, G, and B component images and (d) resulting RGB color image. (f)–(g) R, G, and B component images and (h) resulting RGB color image.

$$\mathbf{u} = \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b}$$

$$\mathbf{v} = \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b}$$

$$g_{xx} = \mathbf{u}^T\mathbf{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2$$

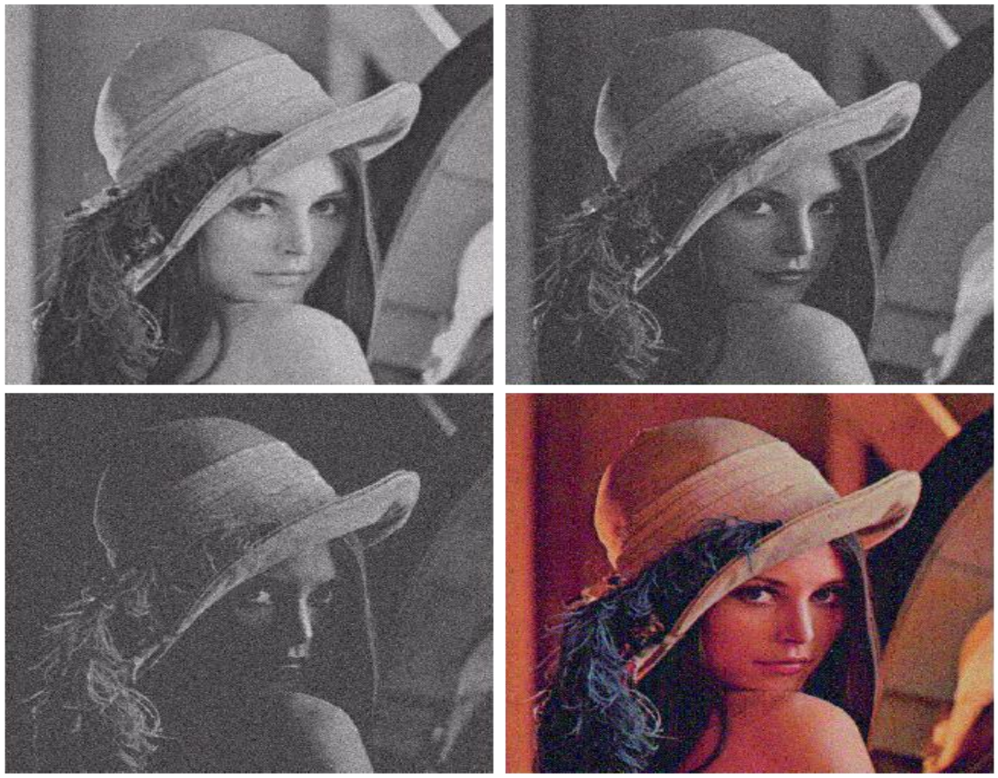
$$g_{yy} = \mathbf{v}^T\mathbf{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2$$

$$g_{xy} = \mathbf{u}^T\mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2g_{xy}}{\left(g_{xx} - g_{yy}\right)}\right]$$

$$F(\theta) = \left\{ \frac{1}{2} \left[\left(g_{xx} + g_{yy} \right) + \left(g_{xx} - g_{yy} \right) \cos 2\theta + 2 g_{xy} \sin 2\theta \right] \right\}^{1/2}$$

Noise in Color Images



a b
c d

FIGURE 6.48 (a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

a b c

FIGURE 6.49 HSI components of the noisy color image in Fig. 6.48(d). (a) Hue. (b) Saturation. (c) Intensity.

a b
c d

FIGURE 6.50 (a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.

Color Image Compression

a b
c d

FIGURE 6.51 Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a).

Wavelets and Multi-resolution Processing

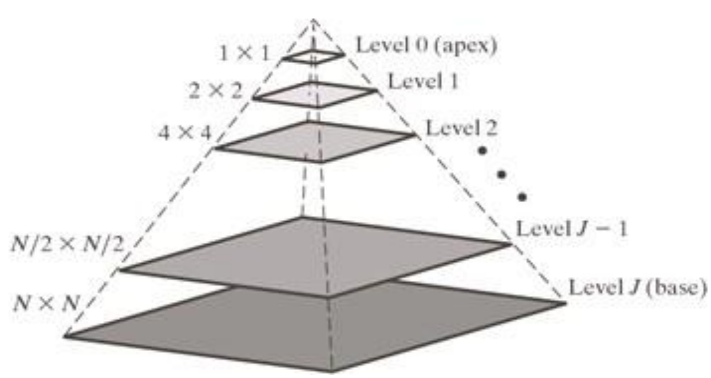
- Fourier transform has its basis functions in sinusoids
- Wavelets based on small waves of varying frequency and limited duration
- In addition to frequency, wavelets capture temporal information
  - ✓ Bound in both frequency and time domains
  - ✓ Localized wave and decays to zero instead of oscillating forever
- Form the basis of an approach to signal processing and analysis known as *multiresolution theory*
  - ✓ Concerned with the representation and analysis of images at different resolutions
  - ✓ Features that may not be prominent at one level can be easily detected at another level

Comparison with Fourier transform

- Fourier transform used to analyze signals by converting signals into a continuous series of sine and cosine functions, each with a constant frequency and amplitude, and of infinite duration
- Real world signals (images) have a finite duration and exhibit abrupt changes in frequency
- Wavelet transform converts a signal into a series of wavelets
- In theory, signals processed by wavelets can be stored more efficiently compared to Fourier transform
- Wavelets can be constructed with rough edges, to better approximate real-world signals
- Wavelets do not remove information but move it around, separating out the noise and averaging the signal
- Noise (or detail) and average are expressed as sum and difference of signal, sampled at different points.
- Objects in images are connected regions of similar texture and intensity levels
- Use high resolution to look at small objects; coarse resolution to look at large objects
- If you have both large and small objects, use different resolutions to look at them
- Images are 2D arrays of intensity values with locally varying statistics

Image Pyramids

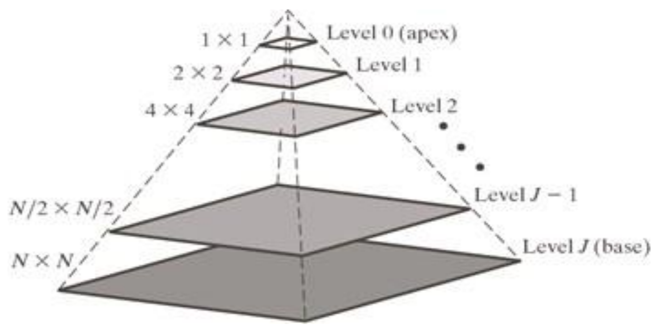
- Originally devised for machine vision and image compression.
- It is a collection of images at decreasing resolution levels.
- Base level is of size $2^J x 2^J$ or NxN.
- Level j is of size $2^j x 2^j$.



Approximation pyramid:

- At each reduced resolution level we have a filtered and downsampled image.

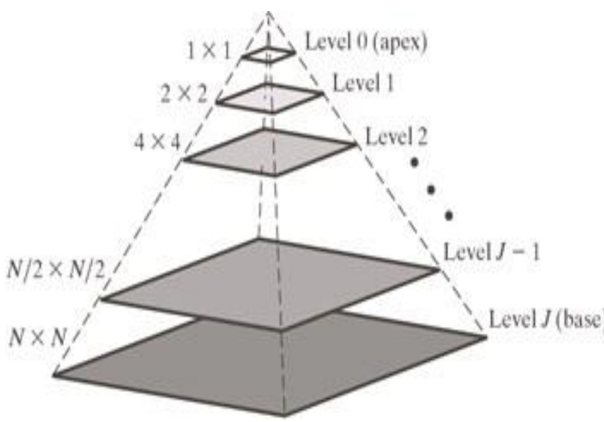
$$f_{\downarrow 2}(n) = f(2n)$$



Prediction pyramid:

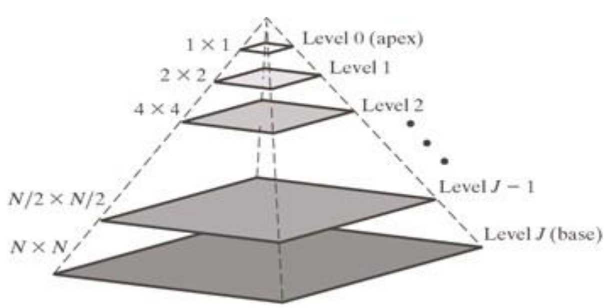
A prediction of each high resolution level is obtained by upsampling (inserting zeros) the previous low resolution level (prediction pyramid) and interpolation (filtering).

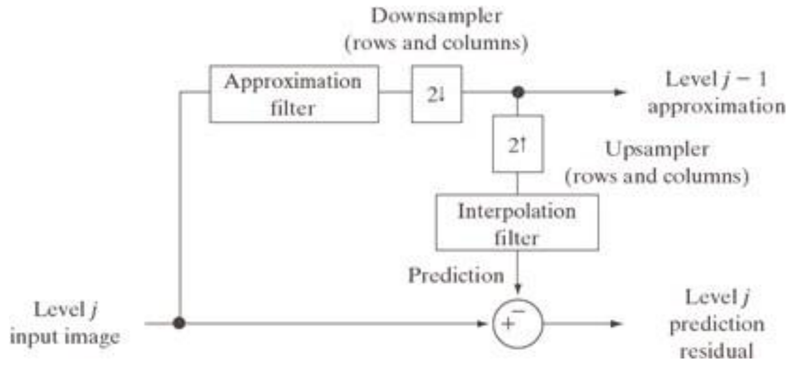
$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$



Prediction residual pyramid:

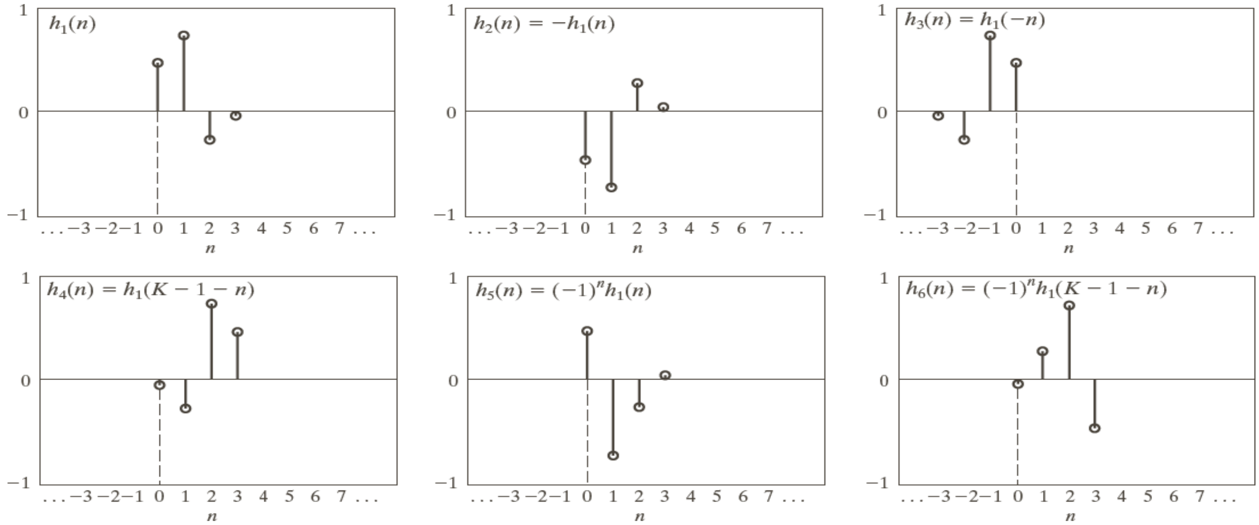
- At each resolution level, the prediction error is retained along with the lowest resolution level image.
- The original image may be reconstructed from this information.



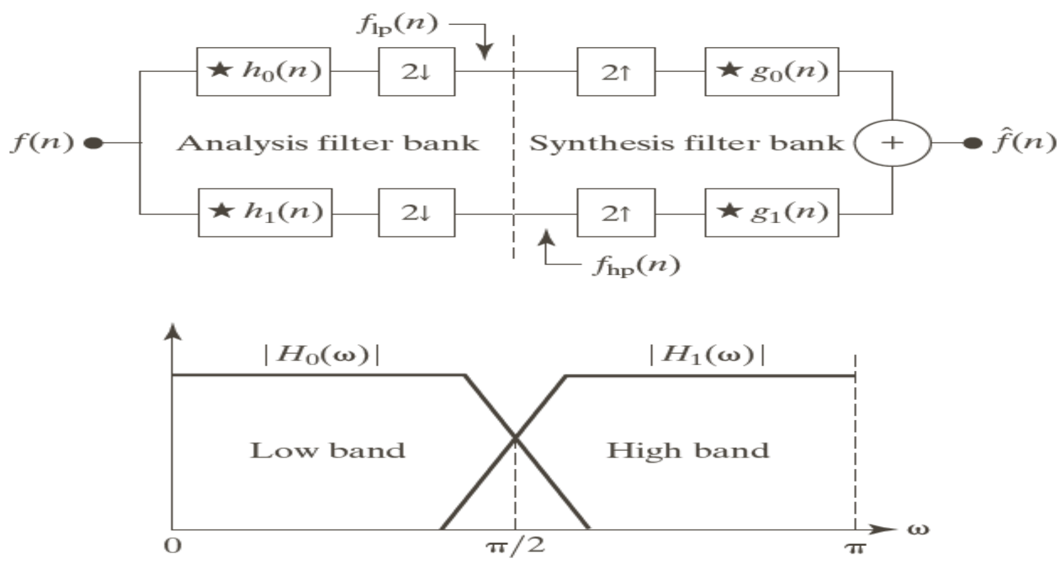


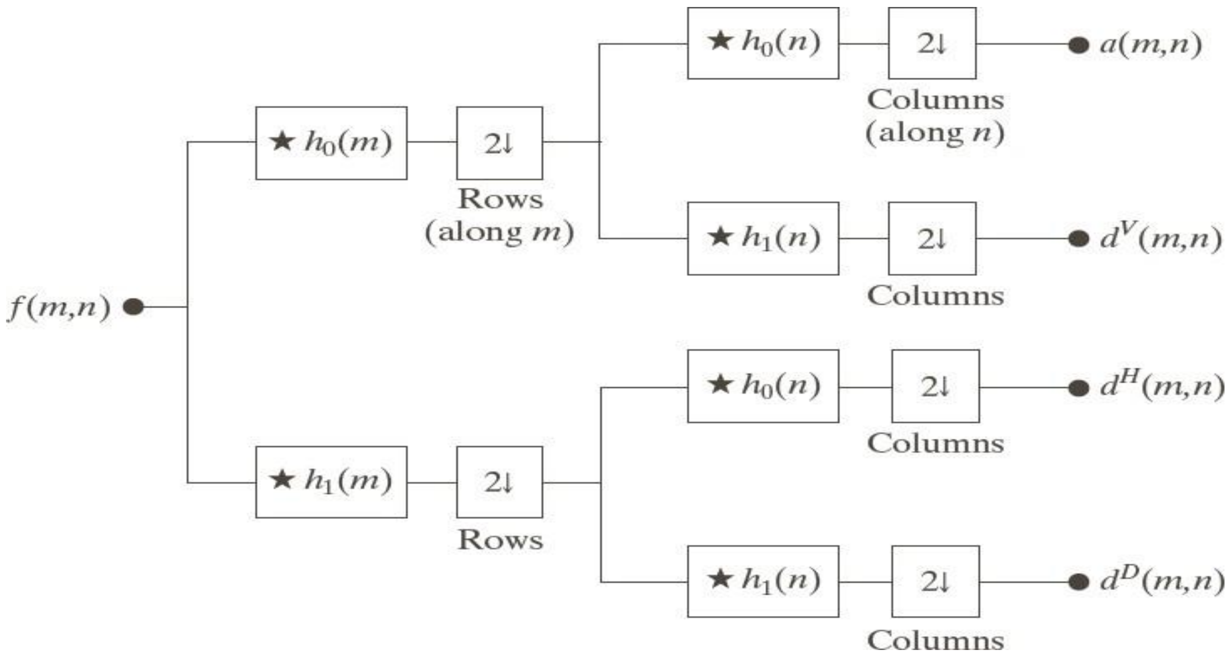
Subband Coding

- An image is decomposed to a set of bandlimited components (subbands).
- The decomposition is carried by filtering and downsampling.
- If the filters are properly selected the image may be reconstructed without error by filtering and upsampling.



A two-band subband coding





The Haar Transform

- It is due to Alfred Haar [1910].
- Its basis functions are the simplest known orthonormal wavelets.
- The Haar transform is both separable and symmetric:
- T=HFH,
- F is a NxN image and H is the NxN transformation matrix and T is the NxN transformed image.
- Matrix H contains the Haar basis functions.
- The Haar basis functions $h_k(z)$ are defined for in $0\leq z \leq 1$, for k=0,1,…, N-1, where $N=2^n$.

To generate H:

- we define the integer $k=2^p+q-1$, with $0\leq p \leq N-1$.
- if p=0, then q=0 or q=1.
- if $p\neq 0$, $1\leq q \leq 2^p$

For the above pairs of p and q, a value for k is determined and the Haar basis functions are computed

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}} z \in [0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \leq z \leq (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \leq z \leq q/2^p \\ 0 & \text{otherwise, } z \in [0,1] \end{cases}$$

- The ith row of a NxN Haar transformation matrix contains the elements of $h_k(z)$ for z=0/N, 1/N, 2/N,…, (N-1)/N.

For instance, for N=4, p,q and k have the following values:

k	p	q
0	0	0
1	0	1
2	1	1
3	1	2

and the 4x4 transformation matrix is:

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Similarly, for N=2, the 2x2 transformation matrix is:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- The rows of H_2 are the simplest filters of length 2 that may be used as analysis filters $h_0(n)$ and $h_1(n)$ of a perfect reconstruction filter bank.
- Moreover, they can be used as scaling and wavelet vectors (defined in what follows) of the simplest and oldest wavelet transform.

Multi-resolution Expansions

- **Expansion of a signal $f(x)$:**

$$f(x) = \sum_k \alpha_k \phi_k(x)$$

α_k : real-valued expansion coefficients

$\phi_k(x)$: real-valued expansion functions

$$\alpha_k = \langle \tilde{\phi}_k(x), f(x) \rangle = \int \tilde{\phi}_k^*(x) f(x) dx \qquad \tilde{\phi}_k(x)\text{: the dual function of } \phi_k(x)$$

•If $\{\phi_k(x)\}$ is an orthonormal basis for V, then $\phi_k(x) = \tilde{\phi}_k(x)$

$\phi_k(x)$

•If $\{\phi_k(x)\}$ are not orthonormal but are an orthogonal basis for V,
then the basis funcitons and their duals are called biorthogonal.

Biorthogonal: $\langle \phi_j(x), \tilde{\phi}_k(x) \rangle = \delta_{jk} = \begin{cases} 0 & , j \neq k \\ 1 & , j = k \end{cases}$

Scaling function

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k), \quad \text{for } k \in \mathbf{Z} \text{ and } \phi(x) \in L^2(\mathbf{R})$$

- The subspace spanned over k for any j : $V_j = span\left\{ \phi_{j,k}(x) \right\}$
- The scaling functions of any subspace can be built from k double-resolution copies of themselves. That is,

$$\phi(x) = \sum_n h_\phi(n)\sqrt{2}\phi(2x-n)$$

where the coefficients are called scaling function coefficients.
Requirements of scaling function:

1. The scaling function is orthogonal to its integer translates.
2. The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales.
   That is
   $$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_\infty$$
3. The only function that is common to all V_j is $f(x) = 0$
   That is $V_{-\infty} = \{0\}$
4. Any function can be represented with arbitrary precision.
   That is, $V_\infty = \left\{L^2(\mathbf{R})\right\}$

Wavelet Transforms in One Dimension

Wavelet series expansion

$$f(x) = \sum_k c_{j_0}(k)\phi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\psi_{j,k}(x)$$

$$c_{j_0}(k) = \left\langle f(x), \tilde{\phi}_{j_0,k}(x) \right\rangle = \int f(x)\tilde{\phi}_{j_0,k}(x)dx$$

$$d_j(k) = \left\langle f(x), \tilde{\psi}_{j,k}(x) \right\rangle = \int f(x)\tilde{\psi}_{j,k}(x)dx$$

Discrete Wavelet Transform

- The function $f(x)$ is a sequence of numbers

$$f(x) = \frac{1}{\sqrt{M}}\sum_k W_\phi(j_0,k)\phi_{j_0,k}(x) + \frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_k W_\psi(j,k)\psi_{j,k}(x)$$

$$W_\phi(j_0,k) = \frac{1}{\sqrt{M}}\sum_{x=0}^{M-1} f(x)\phi_{j_0,k}(x)$$

$$W_\psi(j,k) = \frac{1}{\sqrt{M}}\sum_{x=0}^{M-1} f(x)\tilde{\psi}_{j,k}(x)$$

Fast Wavelet Transform

- computationally efficient implementation of the DWT

- the relationship between the coefficients of the DWTat adjacent scales
- also called Mallat's herringbone algorithm
- resembles the twoband subband coding scheme

$$\phi(x)=\sum_n h_\phi(n)\sqrt{2}\phi(2x-n)$$

$$\phi(2^j x-k)=\sum_n h_\phi(n)\sqrt{2}\phi\left(2(2^j x-k)-n\right)=\sum_m h_\phi(m-2k)\sqrt{2}\phi\left(2^{j+1}x-m\right)$$

Similarity

$$\psi(2^j x-k)=\sum_m h_\psi(m-2k)\sqrt{2}\phi(2^{j+1}x-m)$$

Consider the DWT. Assume $\tilde{\phi}(x)=\phi(x)$ and $\tilde{\psi}(x)=\psi(x)$

$$W_\phi(j_0,k)=\frac{1}{\sqrt{M}}\sum_{x=0}^{M-1} f(x)\phi_{j_0,k}(x)$$

$$=\frac{1}{\sqrt{M}}\sum_m h_\psi(m-2k)\sqrt{2}\phi(2^{j_0}x-m)$$

$$\phi(x)=\sum_n h_\phi(n)\sqrt{2}\phi(2x-n)$$

$$W_\psi(j,k)=\frac{1}{\sqrt{M}}\sum_{x=0}^{M-1} f(x)\psi_{j,k}(x)$$

$$=\frac{1}{\sqrt{M}}\sum_x f(x)2^{j/2}\psi(2^j x-k)$$

$$=\frac{1}{\sqrt{M}}\sum_x f(x)2^{j/2}\left[\sum_m h_\psi(m-2k)\sqrt{2}\phi(2^{j+1}x-m)\right]$$

$$=\sum_m h_\psi(m-2k)\left[\frac{1}{\sqrt{M}}\sum_x f(x)2^{(j+1)/2}\phi(2^{j+1}x-m)\right]$$

$$=\sum_m h_\psi(m-2k)W_\phi(j+1,m)$$

Similarity,

$$W_\phi(j,k)=\sum_m h_\phi(m-2k)W_\phi(j+1,m)$$

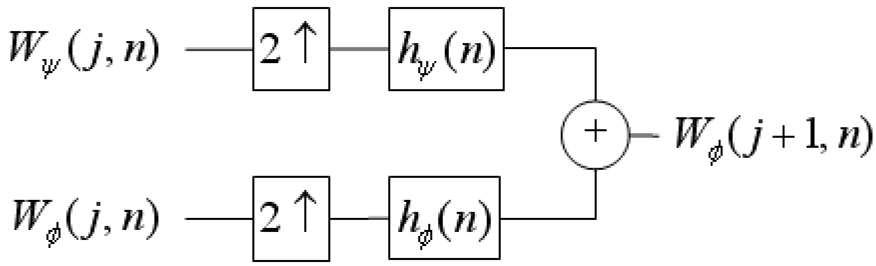


Figure: An FWT^{-1} synthesis filter bank.

Wavelet Trasnforms in Two Dimensions

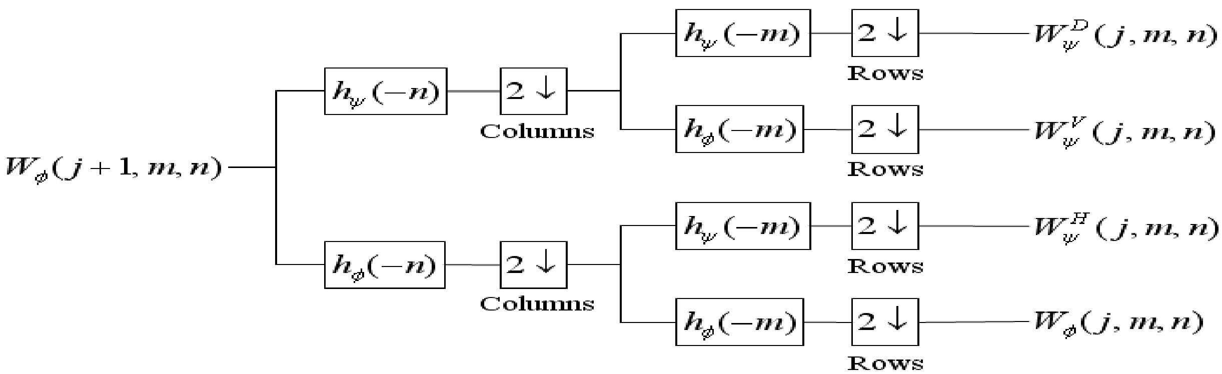
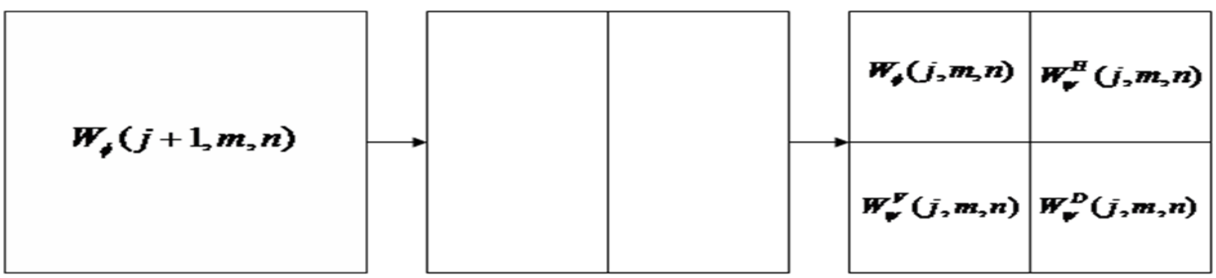


Figure : The two-dimensional FWT — the analysis filter.



two-dimensional decomposition

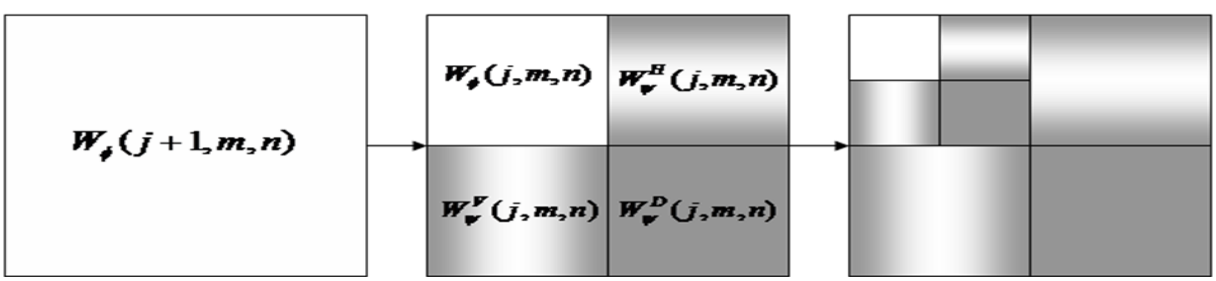


Figure : Two-scale of two-dimensional decomposition

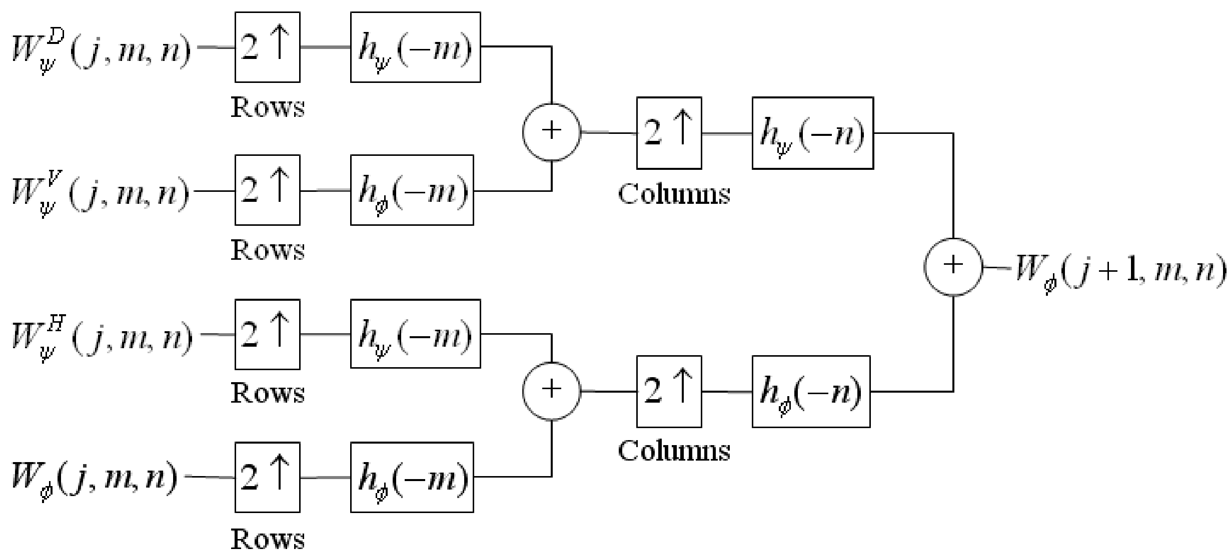
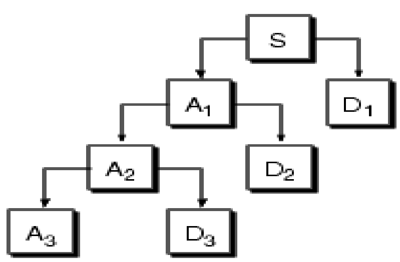


Figure : The two-dimensional FWT — the synthesis filter bank.

Wavelet Packets

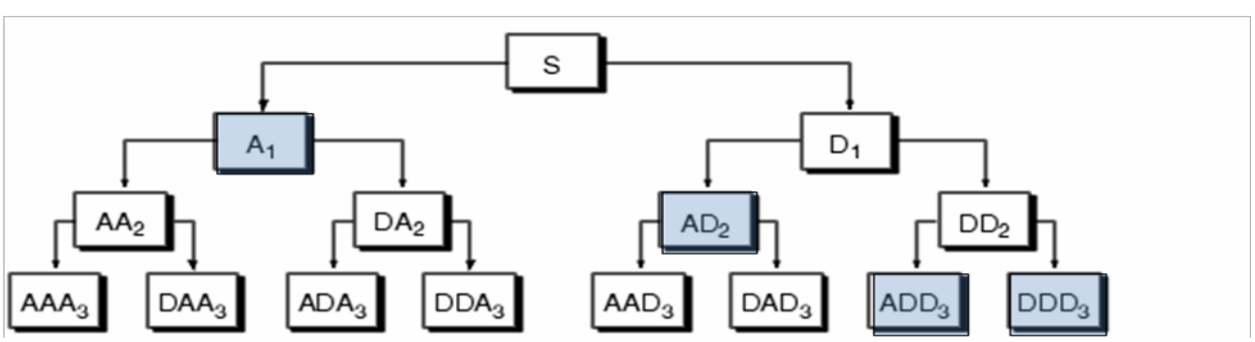
- Generalization of wavelet decomposition
- Very useful for signal analysis



$$S = A_1 + D_1$$
$$= A_2 + D_2 + D_1$$
$$= A_3 + D_3 + D_2 + D_1$$

Wavelet analysis: n+1 (at level n) different ways to reconstuct S

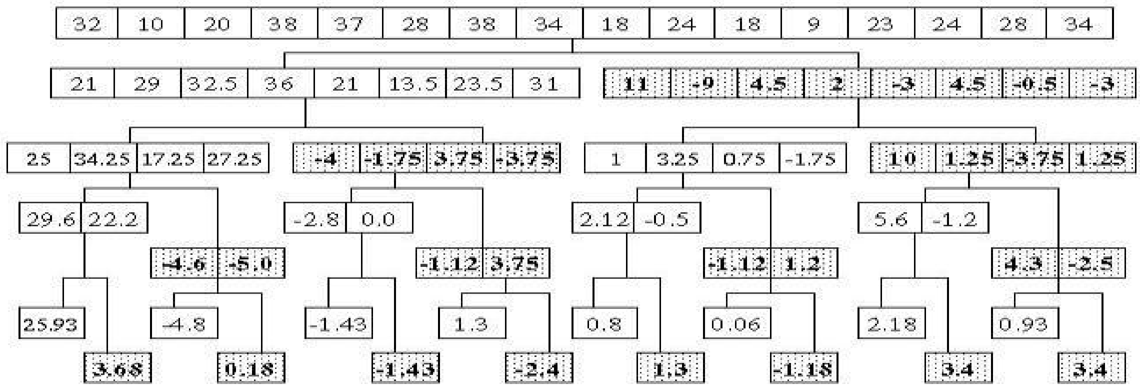
- We have a complete tree



Wavelet packets: a lot of new possibilities to reconstruct S:

i.e. $S=A_1+AD_2+ADD_3+DDD_3$

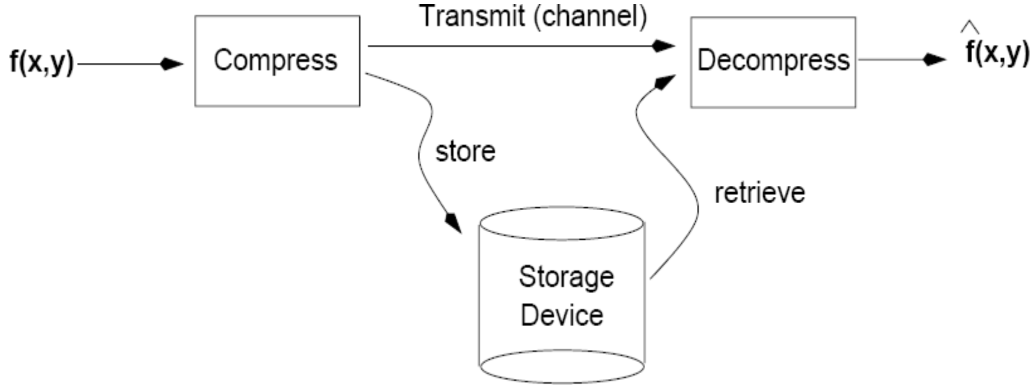
Wavelet Packet Transform example (Haar)



Wavelet Packet Tree

Image Compression

- The goal of image compression is to reduce the amount of data required to represent a digital image.



Types of Image Compression

Lossless

- Information preserving
- Low compression ratios

Lossy

- Not information preserving
- High compression ratios

Types of Data Redundancy

(1) Coding Redundancy
(2) Interpixel Redundancy
(3) Psychovisual Redundancy

- Data compression attempts to reduce one or more of these redundancy types.

(1) Coding Redundancy

Case 1: $l(r_k)$ = constant length

r_k	$p_r(r_k)$	*Code 1*	$l_1(r_k)$
$r_0 = 0$	0.19	000	3
$r_1 = 1/7$	0.25	001	3
$r_2 = 2/7$	0.21	010	3
$r_3 = 3/7$	0.16	011	3
$r_4 = 4/7$	0.08	100	3
$r_5 = 5/7$	0.06	101	3
$r_6 = 6/7$	0.03	110	3
$r_7 = 1$	0.02	111	3

Assume an image with $L = 8$

Assume $l(r_k) = 3$, $L_{avg} = \sum_{k=0}^{7} 3P(r_k) = 3 \sum_{k=0}^{7} P(r_k) = 3$ bits

Total number of bits: $3NM$

<u>Case 2:</u> $l(r_k)$ = variable length

Table 6.1 Variable-Length Coding Example variable length

r_k	$p_r(r_k)$	*Code 1*	$l_1(r_k)$	*Code 2*	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

$$L_{avg} = \sum_{k=0}^{7} l(r_k)P(r_k) = 2.7 \text{ bits}$$

Total number of bits: 2.7NM

$$C_R = \frac{3}{2.7} = 1.11 \text{ (about 10\%)}$$

$$R_D = 1 - \frac{1}{1.11} = 0.099$$

(2) Interpixel Redundancy

- Interpixel redundancy implies that pixel values are correlated (i.e., a pixel value can be reasonably predicted by its neighbors).

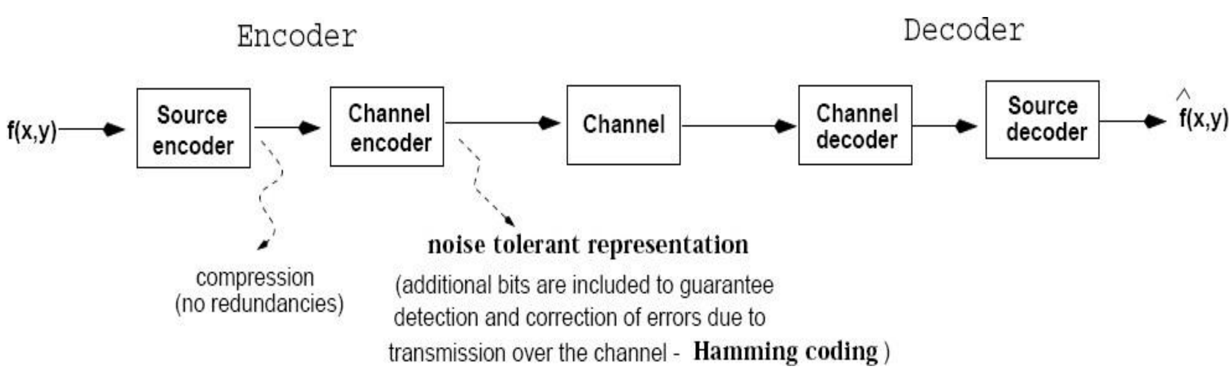
$$f(x) \, o \, g(x) = \int_{-\infty}^{\infty} f(x) g(x+a) da$$

(3) Psychovisual Redundancy

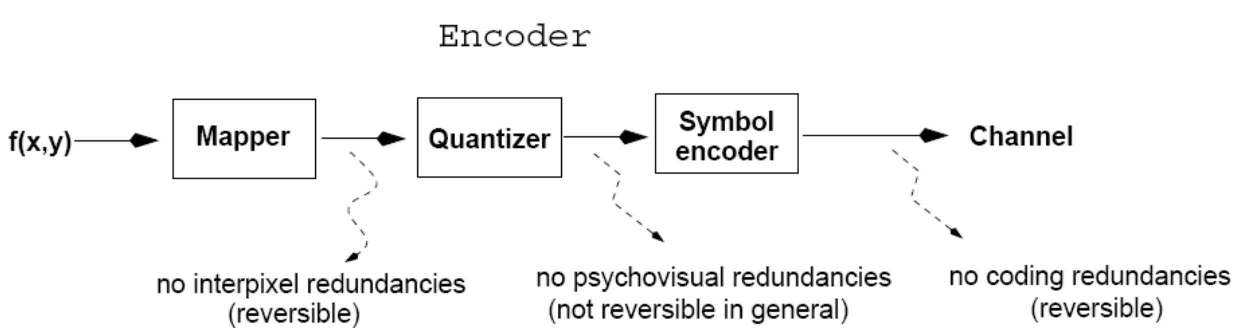
- The human eye is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.
- <u>Idea:</u> discard data that is perceptually insignificant!

Example: quantization

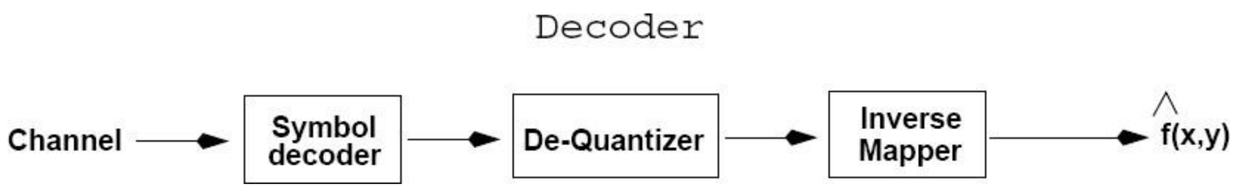
<u>Image Compression Model</u>



will focus on the Source Encoder/Decoder only

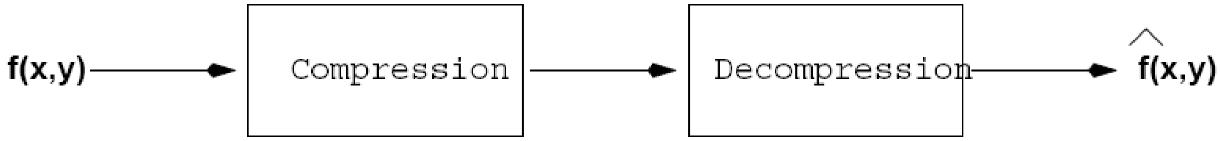


Mapper: transforms data to account for interpixel redundancies
Quantizer: quantizes the data to account for psychovisual redundancies.
Symbol encoder: encodes the data to account for coding redundancies.



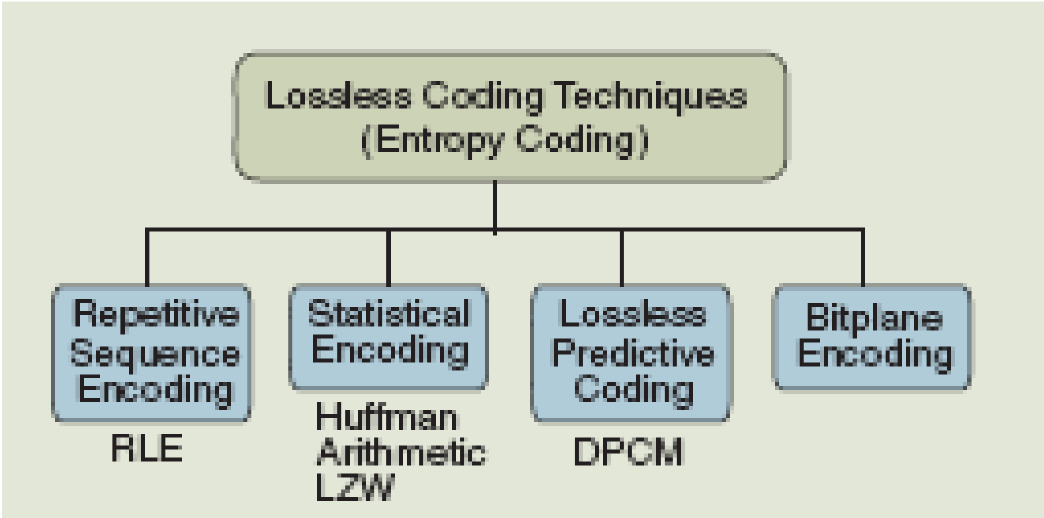
- The decoder applies the inverse steps.
- Note that quantization is irreversible in general.

<u>Error-free (Lossless)Compression</u>

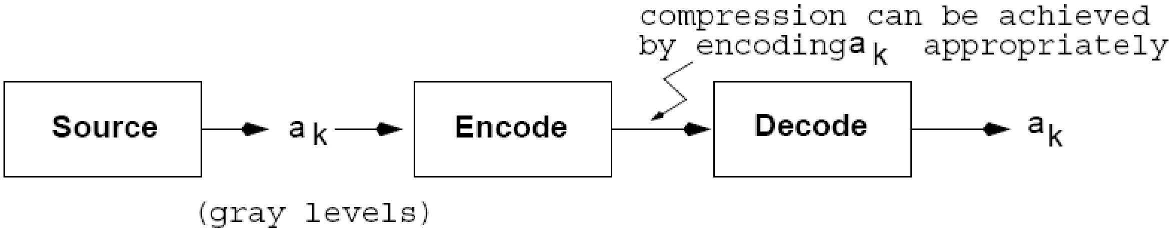


$$e(x,y) = \hat{f}(x,y) - f(x,y) = 0$$

Taxonomy of Lossless Methods



Huffman Coding (addresses coding redundancy)



- A variable-length coding technique.
- Source symbols are encoded one at a time!
    - There is a one-to-one correspondence between source symbols and code words.
- Optimal code - minimizes code word length per source symbol.

LZW Coding (addresses interpixel redundancy

- Requires no prior knowledge of symbol probabilities.
- Assigns fixed length code words to variable length symbol sequences.
    - There is no one-to-one correspondence between source symbols and code words.

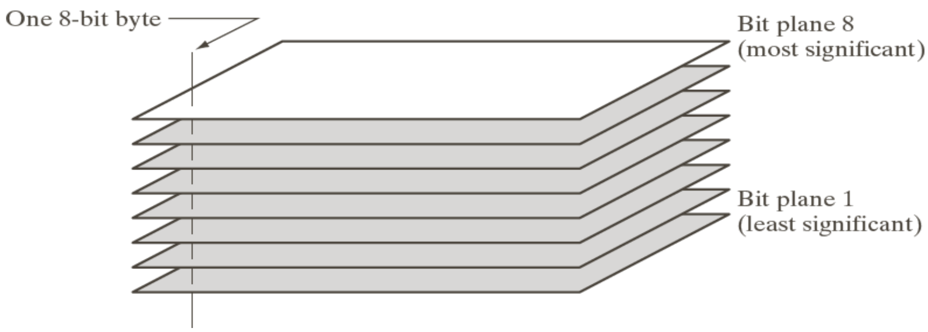
Included in GIF, TIFF and PDF file formats.

Bit-plane coding (addresses interpixel redundancy)

Process each bit plane individually.

(1) Decompose an image into a series of binary images.

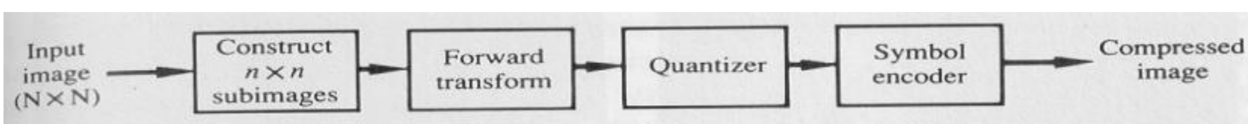
(2) Compress each binary image (e.g., using run-length coding)

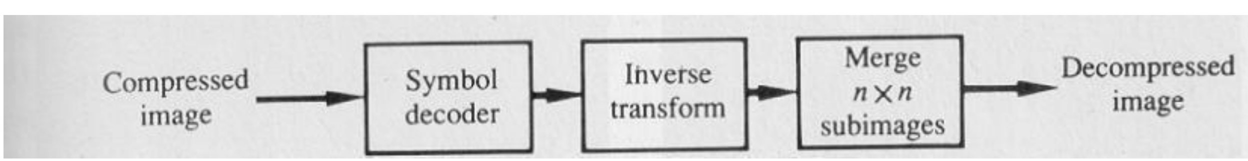


<u>Lossy Compression</u>

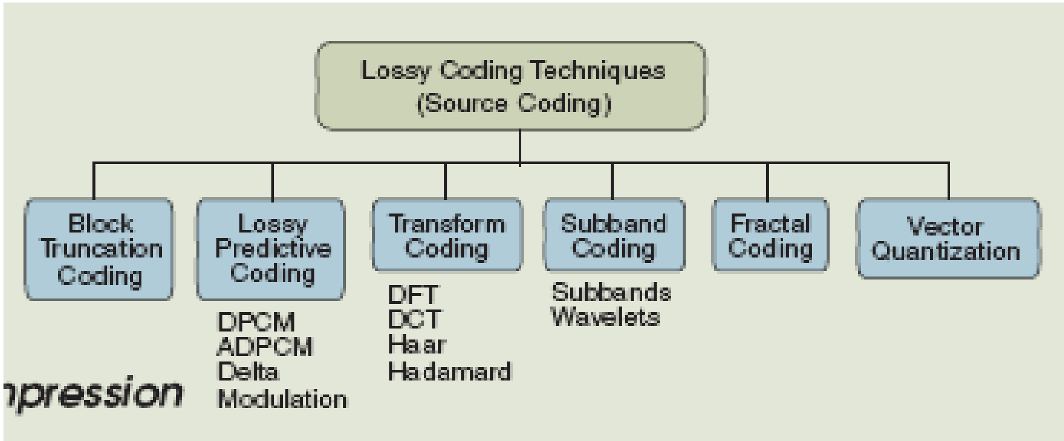
- Transform the image into some other domain to reduce interpixel redundancy.

~ $(N/n)^2$ subimages





Lossy Methods - Taxonomy


Lossy Coding Techniques
(Source Coding)
Block Truncation Coding
Lossy Predictive Coding
Transform Coding
Subband Coding
Fractal Coding
Vector Quantization
DPCM
ADPCM
Delta Modulation
DFT
DCT
Haar
Hadamard
Subbands
Wavelets
npression

UNIT-V MORPHOLOGICAL IMAGE PROCESSING

Introduction

- Morphology: a branch of biology that deals with the form and structure of animals and plants
- Morphological image processing is used to extract image components for representation and description of region shape, such as boundaries, skeletons, and the convex hull

- Identification, analysis, and description of the structure of the smallest unit of words
- Theory and technique for the analysis and processing of geometric structures
  - Based on set theory, lattice theory, topology, and random functions
  - Extract image components useful in the representation and description of region shape such as boundaries, skeletons, and convex hull
  - Input in the form of images, output in the form of attributes extracted from those images
  - Attempt to extract the *meaning* of the images

Preliminaries

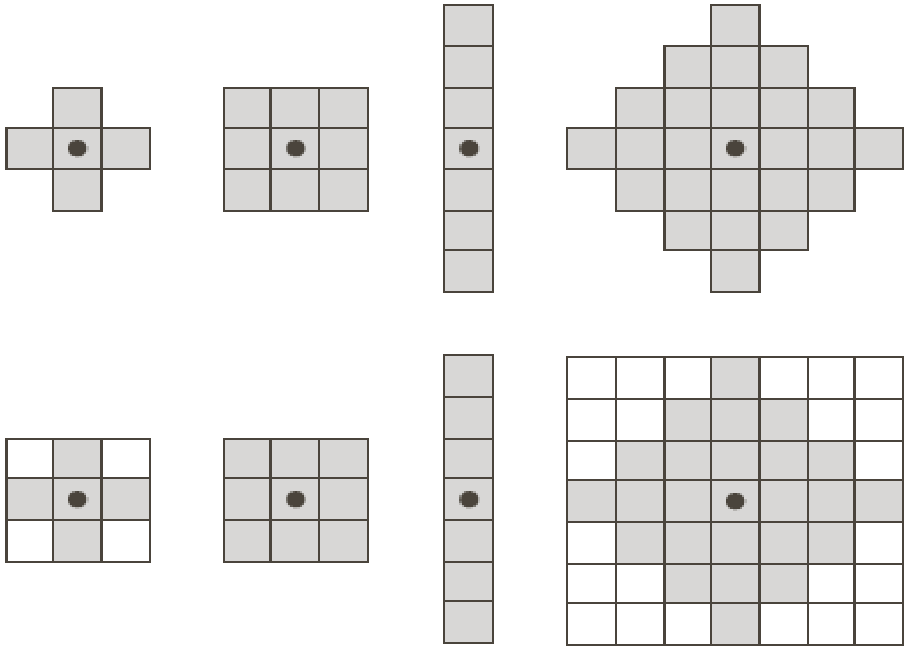
- Set theory in the context of image processing
  - Sets of pixels represent objects in the image
- Set Sets in binary images
  - Members of the 2D integer space Z^2
  - Each element of the set is a 2-tuple whose coordinates are the (x, y) coordinates of a white pixel in the image
  - Gray scale images can be represented as a set of 3-tuples in Z^3
  - Set reflection $\hat{B}$

    $\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$

    - In binary image, $\hat{B}$ is the set of points in B whose (x, y) coordinates have been replaced by $(-x, -y)$
  - Set translation
    - Translation of a set B by point $z = (z_1, z_2)$ is denoted by $(B)_z$

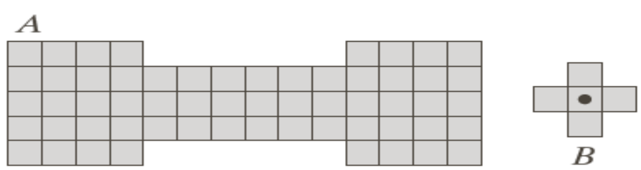
      $(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$

    - In binary image, $(B)_z$ is the set of points in B whose (x, y) coordinates have been replaced by $(x + z_1, y + z_2)$
  - Set reflection and set translation are used to formulate operations based on so-called *structuring elements*
    - Small sets or subimages used to probe an image for properties of interest
    - Figure 9.1



- Preference for ses to be rectangular arrays
- Some locations are such that it does not matter whether they are part of the se
  - Such locations are flagged by × in the se
- The origin of the se must also be specified
  - Indicated by • in Figure 9.2
  - If se is symmetric and no • is shown, the origin is assumed to be at the center of se

– Using ses in morphology



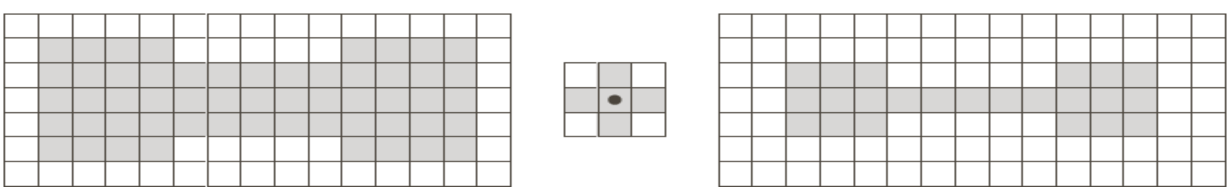
- 

Figure 9.2 – A simple set A and an se B

Convert A to a rectangular array by adding background elements

Make background border large enough to accommodate the entire se when the origin is on the border of original A

Fill in the se with the smallest number of background elements to make it a rectangular array

Operation of set A using se B

1. Create a new set by running B over A

2. Origin of B visits every element of A

3. If B is complet·ely contained in A, mark that location as a member of the new set; else it is not a member of the new set

4. Results in *eroding* the boundary of A

<u>Dilation</u>

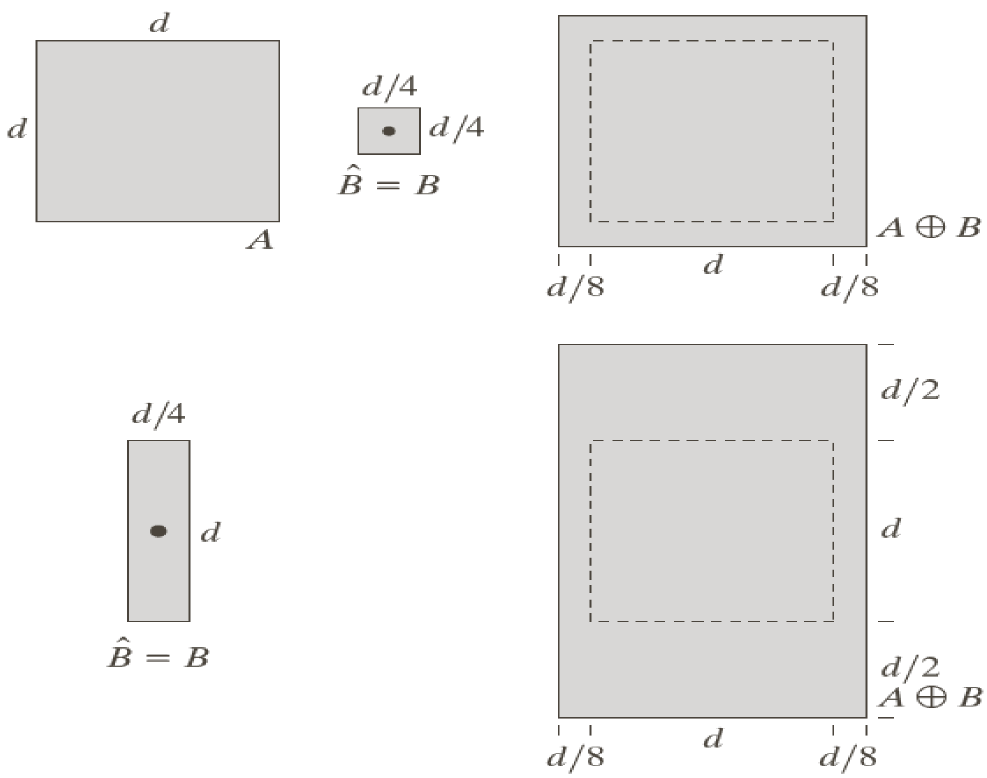
- With A and B as sets in Z^2, dilation of A by B, denoted by $A \oplus B$ is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

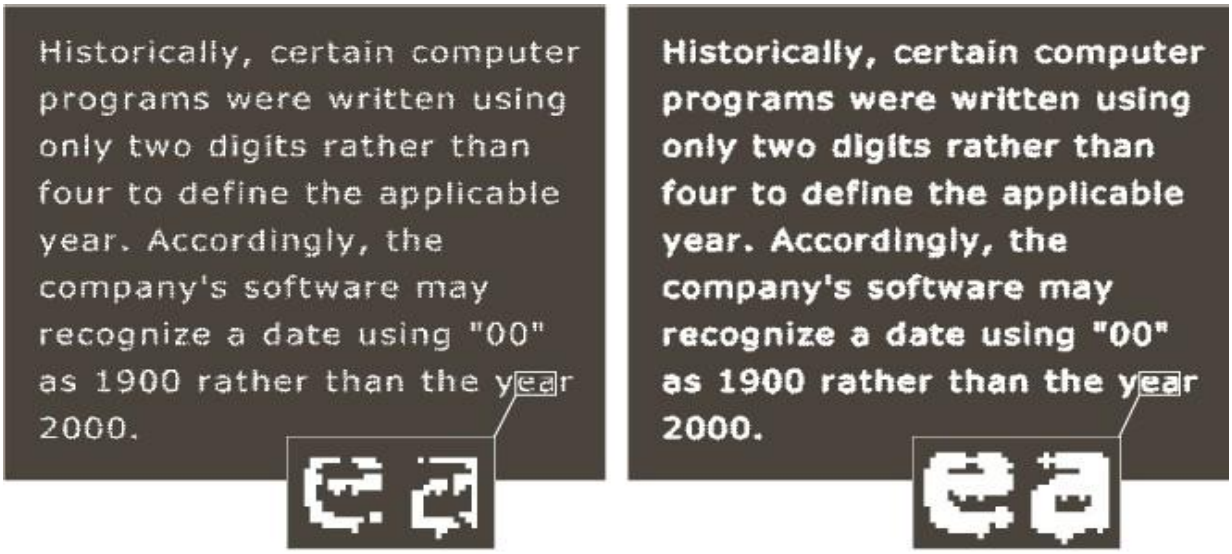
- Reflect B about the origin, and shift the reflection by z
- Dilation is the set of all displacements z such that B and A overlap by at least one element
- An equivalent formulation is

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\}$$

- Grows or thickens objects in a binary image
- Figure 9.6



– Example: Figure 9.7



0	1	0
1	1	1
0	1	0

–

- Bridging gaps in broken characters
- Lowpass filtering produces a grayscale image; morphological operation produces a binary image

- Erosion and dilation are based on set operations and therefore, are nonlinear
- Duality

-Erosion and dilation are duals of each other with respect to set complementation and reflection

$(A \text{ g } B)^c \quad = \quad A^c \oplus \hat{B}(A \oplus B)^c \qquad = \quad A^c \text{ g } \hat{B}$

-Duality property is especially useful when se is symmetric with respect to its origin so that $\hat{B} = B$

-Allows for erosion of an image by dilating its background (A^c) using the same se and complementing the results

-Proving duality

-Definition for erosion can be written as

$(A \text{ g } B)^c = \{z \mid (B)_z \subseteq A\}^C$

$(B)_z \subseteq A \Rightarrow (B)_z \cap A^c = \emptyset$

-So, the previous expression yields

$(A \text{ g } B)^c = \{z \mid (B)_z \cap A^c = \emptyset \}^C$

-The complement of the set of z's that satisfy

$(B)_z \cap A^c = \emptyset$ is the set of z's such that $(B)_z \cap A^c \neq \emptyset$

-This leads to

$$(A \text{ g } B)^c \quad = \quad \{z \mid (B)_z \cap A^c \neq \emptyset \} = \quad A^c \oplus \hat{B}$$

<u>Erosion</u>

-With A and B as sets in Z^2, erosion of A by B, denoted by $A \text{ g } B$ is defined as

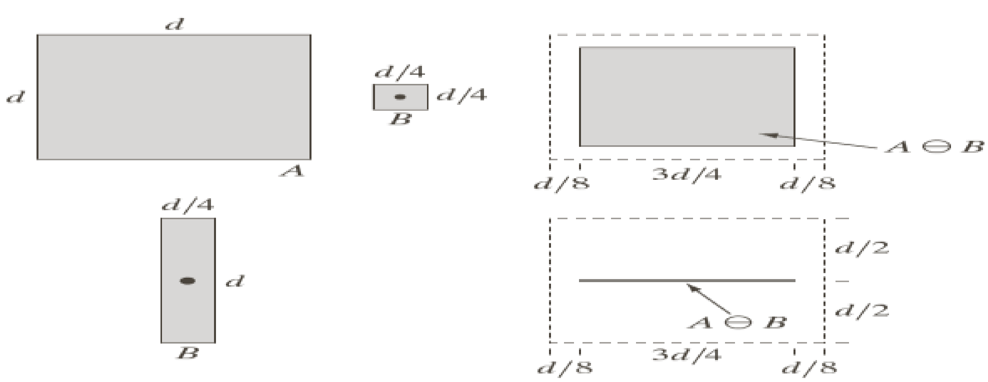
$$A \text{ g } B = \{z \mid (B)_z \subseteq A\}$$

-Set of all points z such that B, translated by z, is contained in A

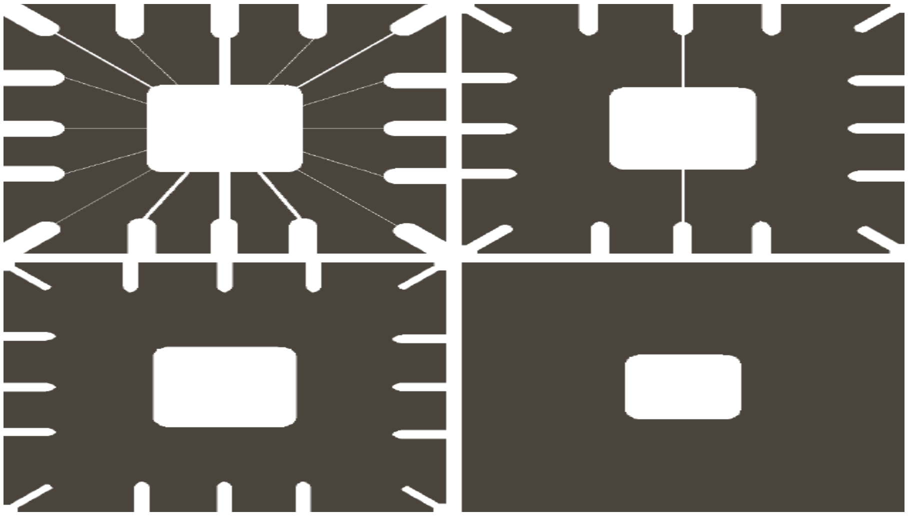
-B does not share any common elements with the background

$A \text{ g } B = \{z \mid (B)_z \cap A^c = \emptyset$

Figure 9.4



-Example: Figure 9.5



-Erosion shrinks or thins objects in a binary image

-Morphological filter in which image details smaller than the se are filtered/removed from the image

<u>Opening and closing</u>

- Opening smoothes the contours of an object, breaks narrow isthmuses, and eliminates thin protrusions
- Closing smoothes sections of contours, fusing narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour
- Opening of a set A by se B, denoted by $A \circ B$, is defined by

$$A \circ B = (A \ominus B) \oplus B$$

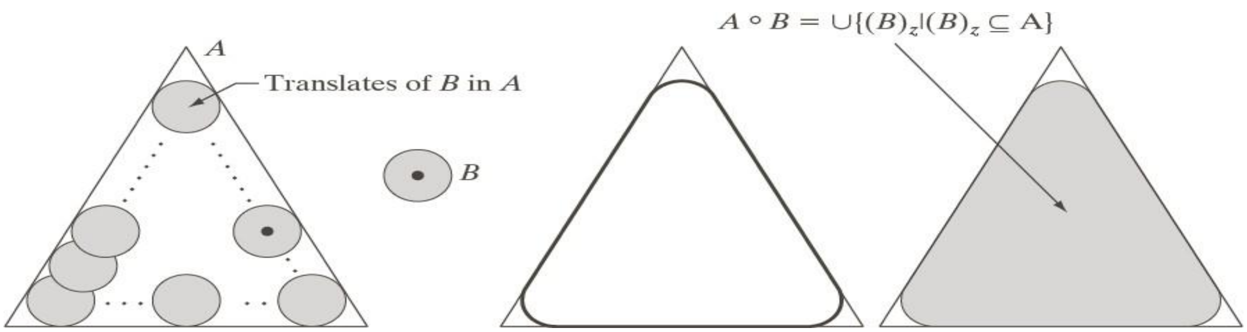
- Closing of a set A by se B, denoted by $A \bullet B$, is defined by

$$A \bullet B = (A \oplus B) \ominus B$$

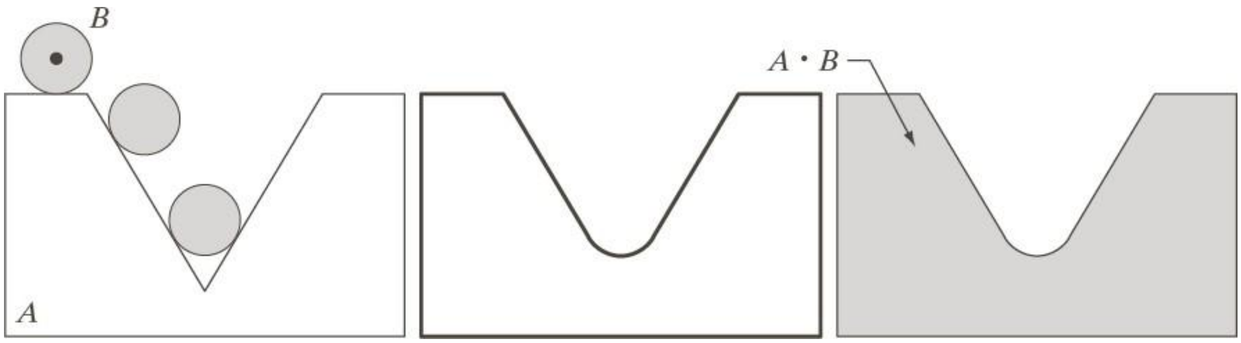
- Geometric interpretation of opening expressed as a fitting process such that

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

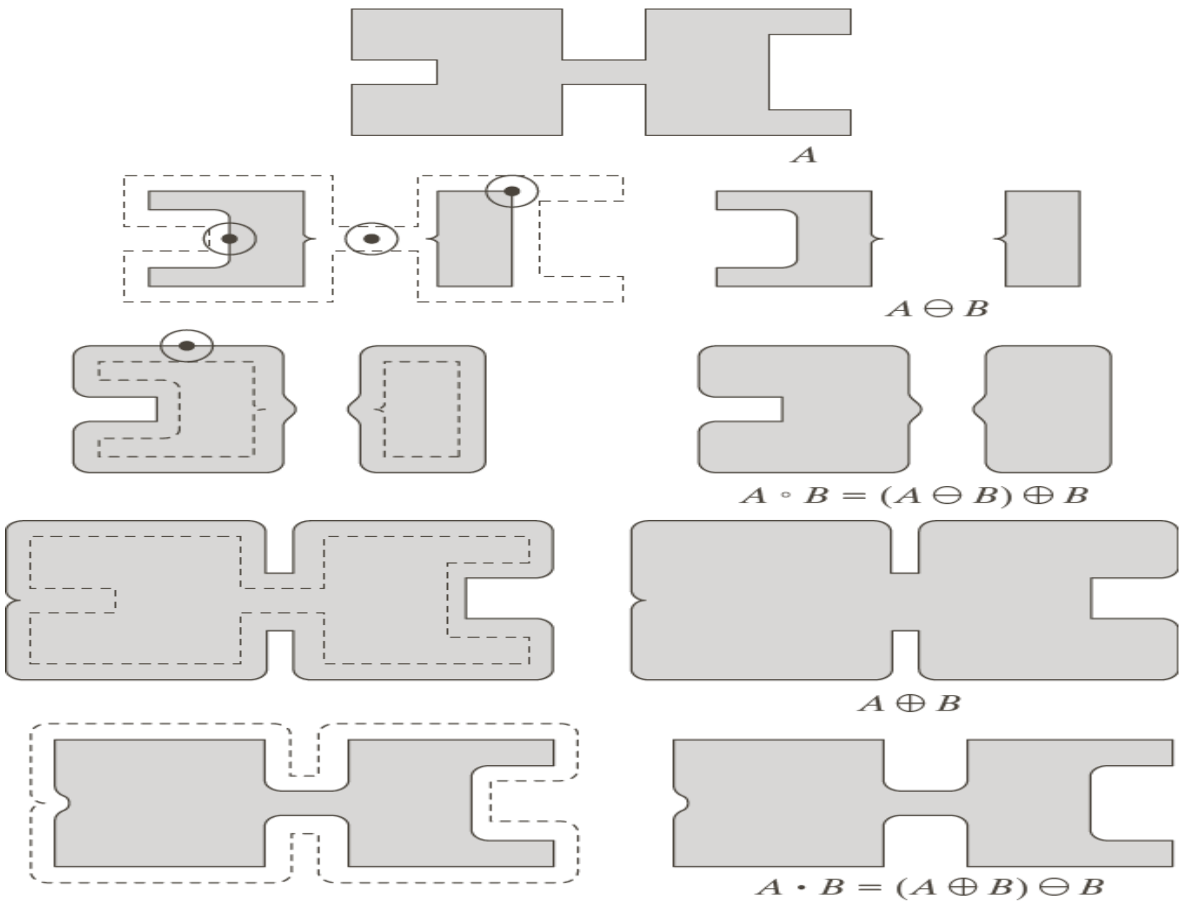
  - Union of all translates of B that fit into A
  - Figure 9.8



- Similar interpretation of closing in Figure 9.9



- Example – Figure 9.10



-Duality property

$$(A \bullet B)^c = (A^c \circ \hat{B})$$

$$(A \circ B)^c = (A^c \bullet \hat{B})$$

- Opening operation satisfies the following properties

  1. $A \circ B \subseteq A$
  2. $C \subseteq D \Rightarrow C \circ B \subseteq D \circ B$
  3. $(A \circ B) \circ B = A \circ B$

- Similarly, closing operation satisfies

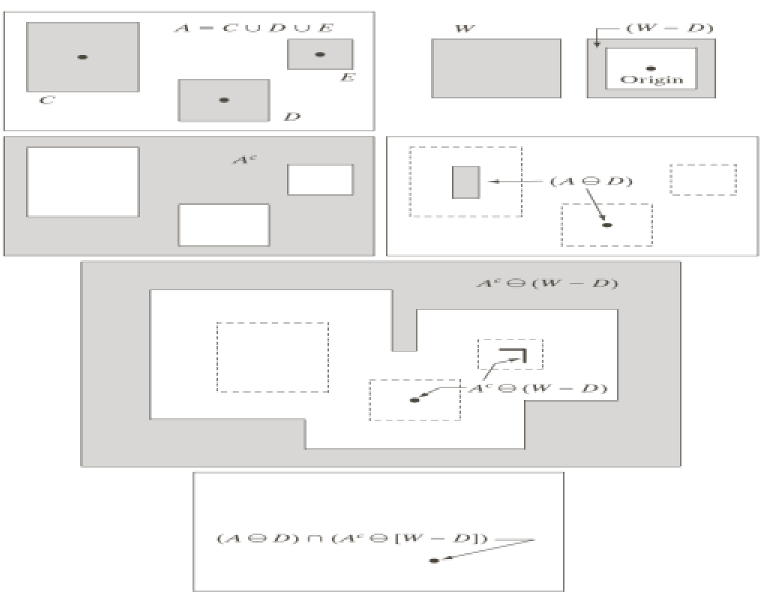
  1. $A \subseteq A \bullet B$
  2. $C \subseteq D \Rightarrow C \bullet B \subseteq D \bullet B$
  3. $(A \bullet B) \bullet B = A \bullet B$

  – In both the above cases, multiple application of opening and closing has no effect after the first application

- Example: Removing noise from fingerprints
  – Noise as random light elements on a dark background

Hit-or-miss transformation

- Basic tool for shape detection in a binary image
  – Uses the morphological erosion operator and a pair of disjoint ses
  – First se fits in the foreground of input image; second se misses it completely
  – The pair of two ses is called *composite structuring element*
- Figure 9.12



- Three disjoint shapes denoted C, D, and E
  - $A = C \cup D \cup E$
- Objective: To find the location of one of the shapes, say D

- Origin/location of each shape given by its center of gravity
- Let D be enclosed by a small window W
- *Local background* of D defined by the set difference $(W-D)$
  - Note that D and $W-D$ provide us with the two disjoint ses

$D \cap (W-D) = \emptyset$

- Compute A^c
- Compute A g D
- Compute A^c g $(W-D)$
- Set of locations where D exactly fits inside A is $(A$ g $D) \cap (A^c$ g $(W-D))$
  - The exact location of D
- If B is the set composed of D and its background, the match of B in A is given by

$$A \sim B = (A \text{ g } D) \cap [A^c \text{ g } (W-D)]$$

- The above can be generalized to the composite se being defined by $B = (B_1, B_2)$ leading to

$A \sim B = (A \text{ g } B_1) \cap (A^c \text{ g } B_2)$

- B_1 is the set formed from elements of B associated with the object; $B_1 = D$
- $B_2 = (W-D)$

- A point z in universe A belongs to the output if $(B_1)_z$ fits in A (hit) and $(B_2)_z$ misses A

<u>Some basic morphological algorithms</u>

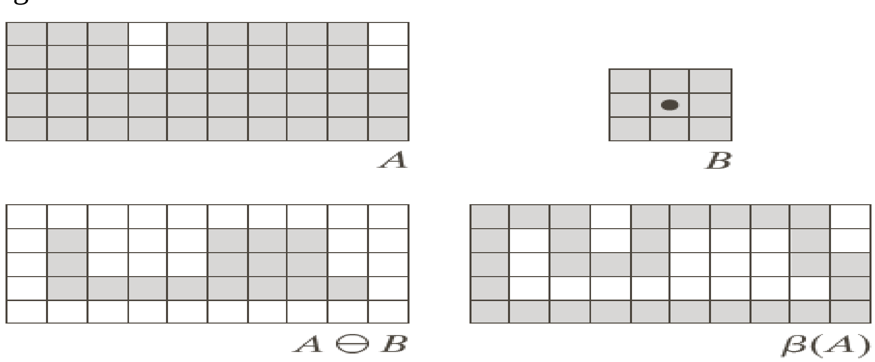
- Useful in extracting image components for representation and description of shape

Boundary extraction

- Boundary of a set A
  - Denoted by $\beta(A)$
  - Extracted by eroding A by a suitable se B and computing set difference between A and its erosion

$\beta(A) = A - (A \text{ g } B)$

- Figure 9.13



* Using a larger se will yield a thicker boundary

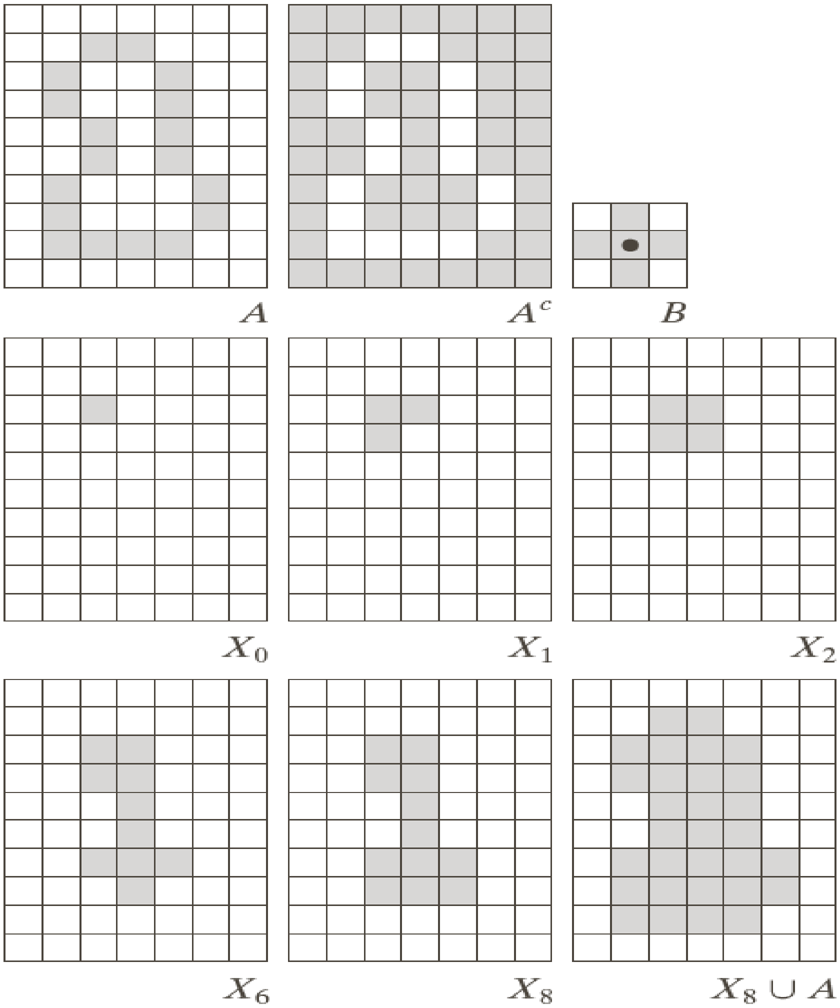
– Figure 9.14

Hole filling

- Hole
  - Background region surrounded by a connected border of foreground pixels
- Algorithm based on set dilation, complementation, and intersection
- Let A be a set whose elements are 8-connected boundaries, each boundary enclosing a background (hole)
- Given a point in each hole, we want to fill all holes
- Start by forming an array X_0 of 0s of the same size as A
  - The locations in X_0 corresponding to the given point in each hole are set to 1
- Let B be a symmetric se with 4-connected neighbors to the origin

0	1	0
1	1	1
0	1	0

- Compute $X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \ldots$
- Algorithm terminates at iteration step k if $X_k = X_{k-1}$
- X_k contains all the filled holes
- $X_k \cup A$ contains all the filled holes and their boundaries
- The intersection with A^c at each step limits the result to inside the roi
  - Also called *conditioned dilation*
- Figure 9.15



- Thresholded image of polished spheres (ball bearings)
- Eliminate reflection by hole filling
- Points inside the background selected manually

Extraction of connected components

- Let *A* be a set containing one or more connected components

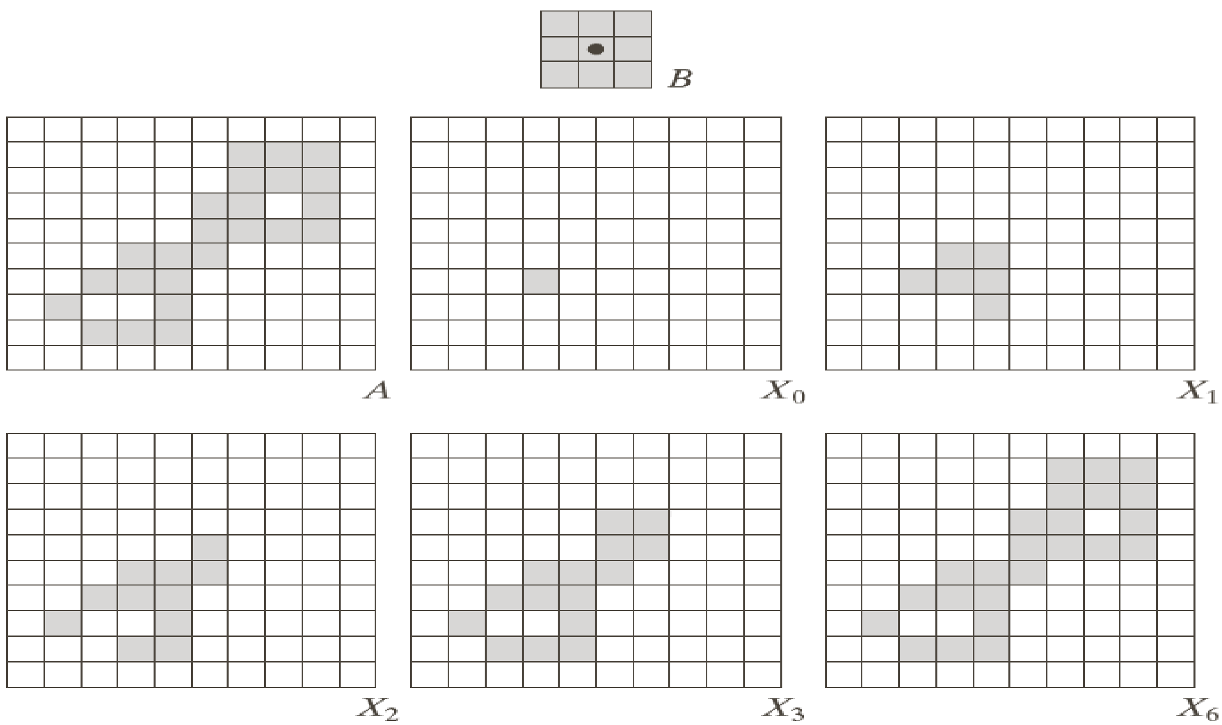
- Form an array X_0 of the same size as A
  - All elements of X_0 are 0 except for one point in each connected component set to 1
- Select a suitable se B, possibly an 8-connected neighborhood as

1	1	1
1	1	1
1	1	1

- Start with X_0 and find all connected components using the iterative procedure

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \ldots$$

- Procedure terminates when $X_k = X_{k-1}$; X_k contains all the connected components in the input image
- The only difference from the hole-filling algorithm is the intersection with A instead of A^c
  - This is because here, we are searching for foreground points while in hole filling, we looked for background points (holes)
- Figure 9.17



  - X-ray image of chicken breast with bone fragments
  - Objects of "significant size" can be selected by applying erosion to the thresholded image
  - We may apply labels to the extracted components (region labeling)

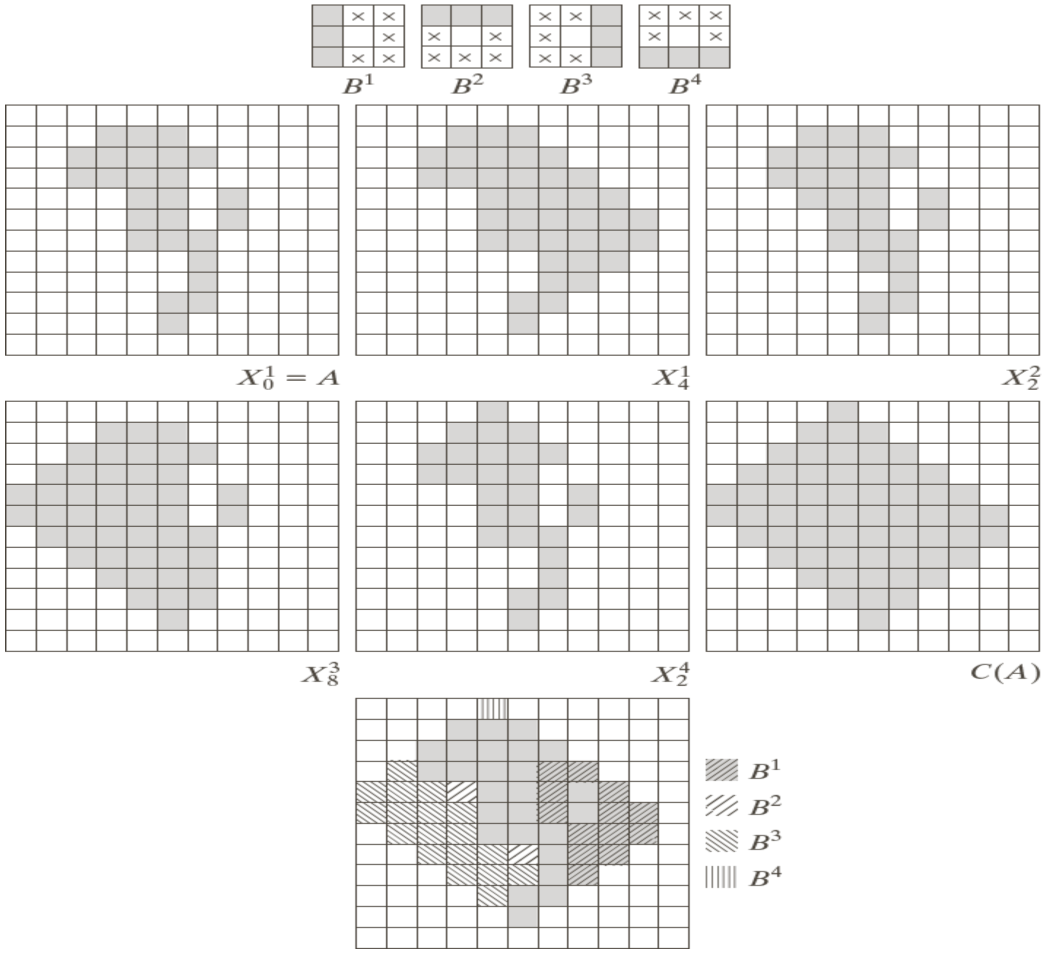
Convex hull

- Convex set A
  - Straight line segment joining any two points in A lies entirely within A
- Convex hull H of an arbitrary set of points S is the smallest convex set containing S

- Set difference $H - S$ is called the *convex deficiency* of S
- Convex hull and convex deficiency are useful to describe objects

– Algorithm to compute convex hull $C(A)$ of a set A

- Figure 9.19



- Let B^i, $i = 1, 2, 3, 4$ represent the four structuring elements in the figure
  - B^i is a clockwise rotation of B^{i-1} by 90°
- Implement the equation

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \text{ and } k = 1, 2, 3 \ldots$$

with $X_0^i = A$

- Apply hit-or-miss with B^1 till $X_k == X_{k-1}$, then, with B^2 over original A, B^3, and B_4
- Procedure converges when $X_k^i = X_{k-1}^i$ and we let $D^i = X_k^i$
- Convex hull of A is given by

$$C(A) = \bigcup_{i=1}^{4} D^i$$

– Shortcoming of the above procedure

- Convex hull can grow beyond the minimum dimensions required to guarantee convexity
- May be fixed by limiting growth to not extend past the bounding box for the original set of points

Thinning

- Transformation of a digital image into a simple topologically equivalent image
  - Remove selected foreground pixels from binary images
  - Used to tidy up the output of edge detectors by reducing all lines to single pixel thickness
- Thinning of a set A by se B is denoted by $A \otimes B$
- Defined in terms of hit-or-miss transform as

$$\begin{aligned} A \otimes B &= A - (A \sim B) \\ &= A \cap (A \sim B)^c \end{aligned}$$

- Only need to do pattern matching with se; no background operation required in hit-or-miss transform
- A more useful expression for thinning A symmetrically based on a sequence of ses

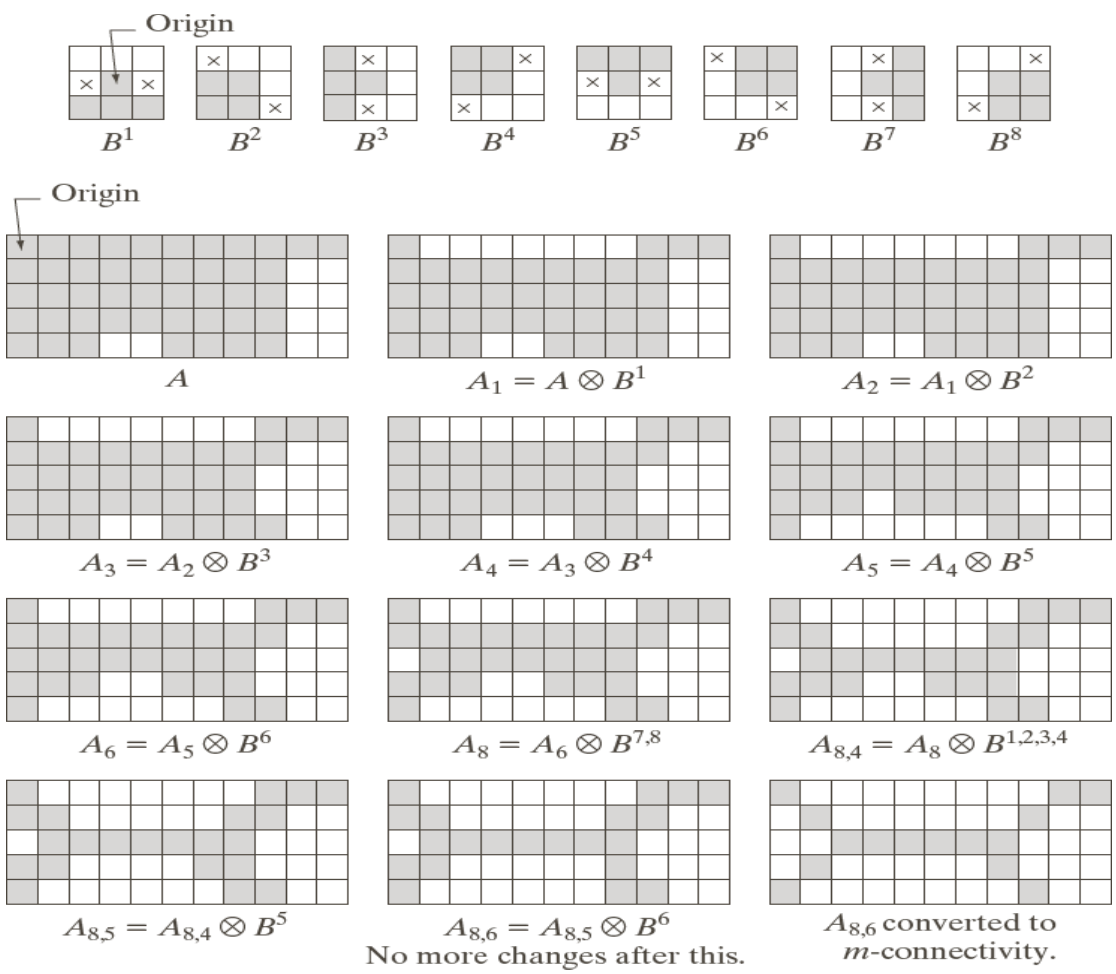
$$\{B\} = \{B^1, B^2, \ldots, B^n\}$$

where B^i is a rotated version of B^{i-1}

- Define thinning by a sequence of ses as

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$

- Figure 9.21



* Iterate over the procedure till convergence

Thickening

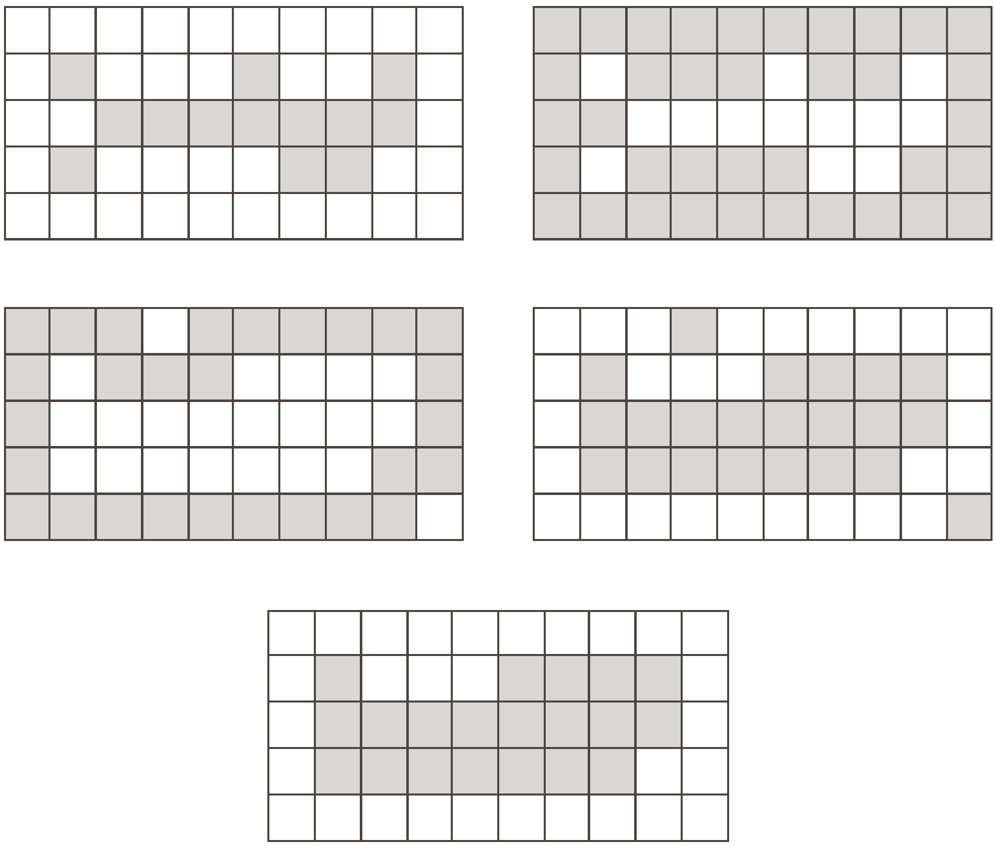
- Morphological dual of thinning defined by

$$A \circledS B = A \cup (A \sim B)$$

- ses complements of those used for thinning
- Thickening can also be defined as a sequential operation

$$A \circledS \{B\} = ((\ldots((A \circledS B^1) \circledS B^2) \ldots) \circledS B^n)$$

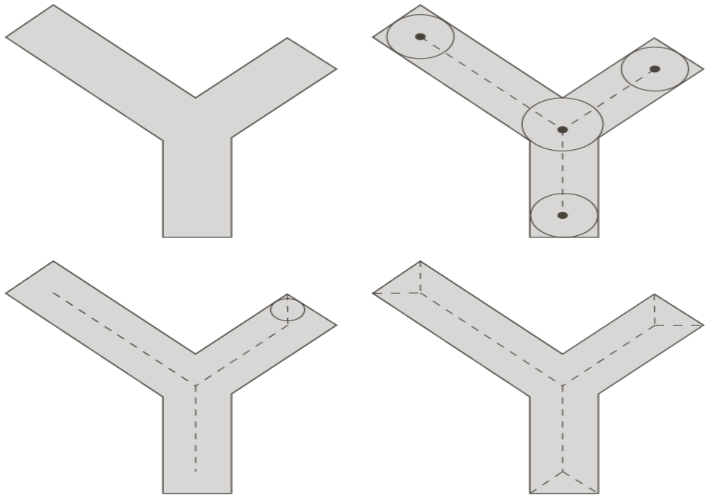
- Figure 9.22



-
- Usual practice to thin the background and take the complement
  - May result in disconnected points
  - Post-process to remove the disconnected points

Skeletons

- Figure 9.23



–

- Skeleton $S(A)$ of a set A
- Deductions
  1. If z is a point of $S(A)$ and $(D)_z$ is the largest disk centered at z and contained in A, one cannot find a larger disk (not necessarily centered at z) containing $(D)_z$ and included in A; $(D)_z$ is called a *maximum disk*
  2. Disk $(D)_z$ touches the boundary of A at two or more different places

– Skeleton can be expressed in terms of erosions and openings

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

where

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

- $A \ominus kB$ indicates k successive erosions of A

$$(A \ominus kB) = ((\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B)$$

- K is the last iterative step before A erodes to an empty set

$$K = \max\{k \mid (A \ominus kB) \neq \emptyset\}$$

- $S(A)$ can be obtained as the union of skeleton subsets $S_k(A)$
- A can be reconstructed from the subsets using the equation

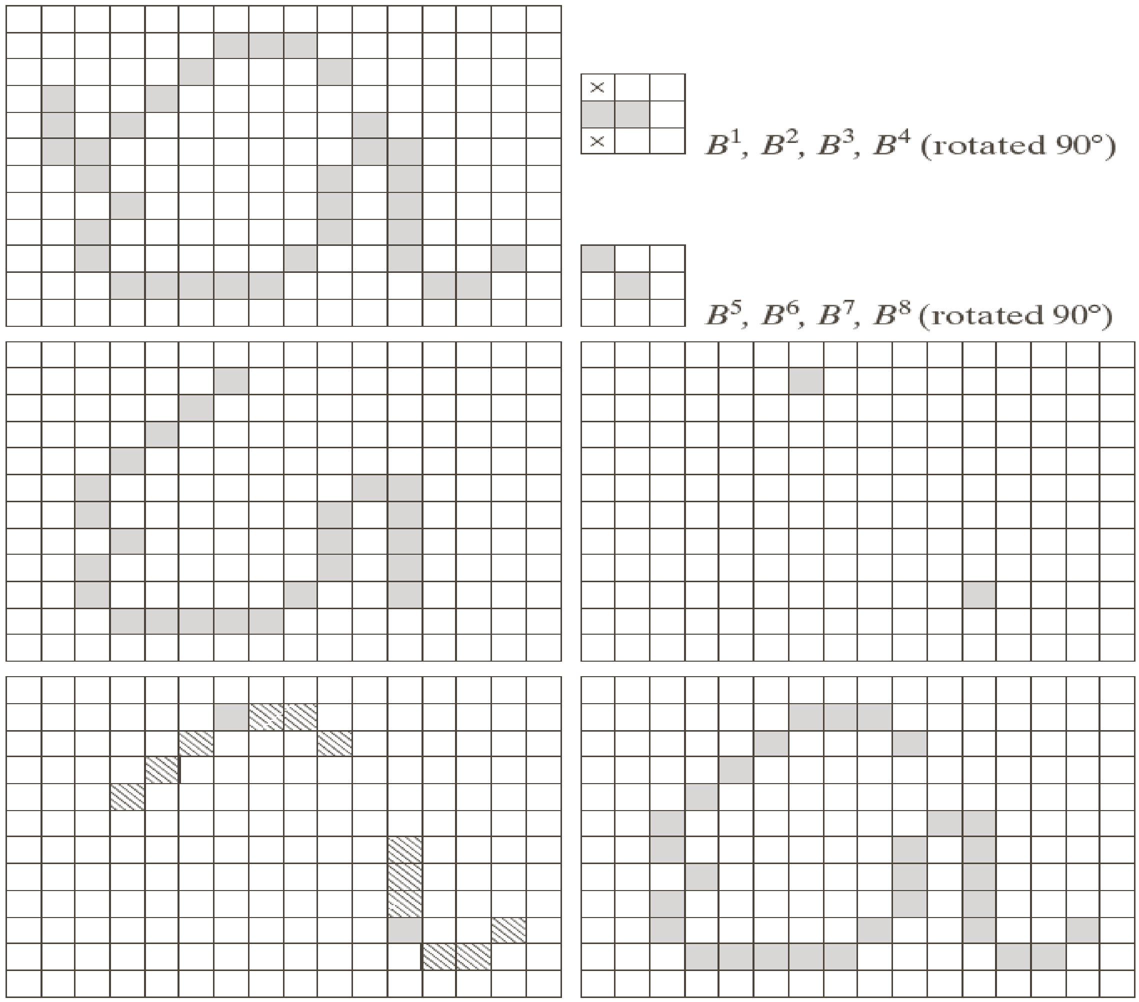
$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$$

where $(S_k(A) \oplus kB)$ denotes k successive dilations of $S_k(A)$

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B)$$

Pruning

- Complement to thinning and sketonizing algorithms to remove unwanted parasitic components
- Automatic recognition of hand-printed characters
  - Analyze the shape of the skeleton of each character
  - Skeletons characterized by “spurs” or parasitic components
  - Spurs caused during erosion by non-uniformities in the strokes
  - Assume that the length of a spur does not exceed a specific number of pixels
- Figure 9.25 – Skeleton of hand-printed “a”



- Suppress a parasitic branch by successively eliminating its end point
- Assumption: Any branch with ≤ 3 pixels will be removed
- Achieved with thinning of an input set A with a sequence of ses designed to detect only end points

$$X_1 = A \otimes \{B\}$$

- Figure 9.25d – Result of applying the above thinning three times
- Restore the character to its original form with the parasitic branches removed
- Form a set X_2 containing all end points in X_1

$$X_2 = \bigcup_{k=1} (X_1 \sim B^k)$$

- Dilate end points three times using set A as delimiter

$$X_3 = (X_2 \oplus H) \cap A$$

where H is a 3 × 3 se of 1s and intersection with A is applied after each step

* The final result comes from

$$X_4 = X_1 \cup X_3$$

Image Segmentation

- Image segmentation divides an image into regions that are connected and have some similarity within the region and some difference between adjacent regions.
- The goal is usually to find individual objects in an image.
- For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.

  -Similarity may be due to pixel intensity, color or texture.

  -Differences are sudden changes (discontinuities) in any of these, but especially sudden changes in intensity along a boundary line, which is called an edge.

Detection of Discontinuities

- There are three kinds of discontinuities of intensity: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image.The mask determines which kind of discontinuity to look for.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

FIGURE 10.1 A general 3 × 3 mask.

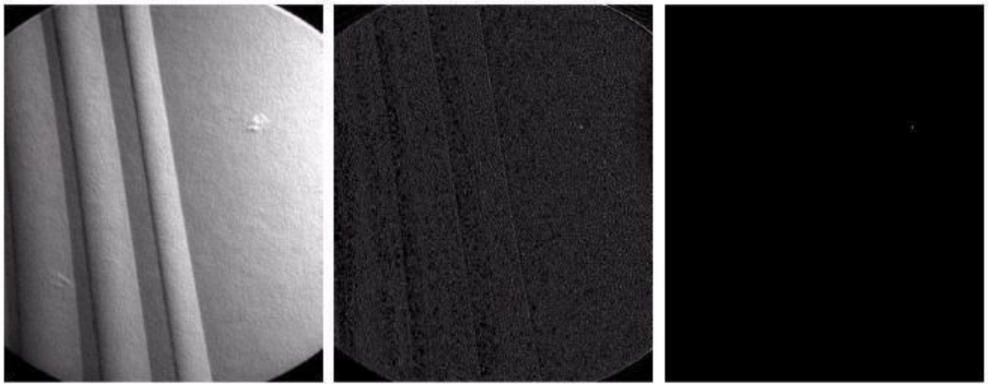
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Detection of Discontinuities point detection

$$|R| \geq T$$

where T : a nonnegative threshold

−1	−1	−1
−1	8	−1
−1	−1	−1



a
b c d

FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities Line detection

- Only slightly more common than point detection is to find a one pixel wide line in an image.
- For digital images the only three point straight lines are only horizontal, vertical, or diagonal (+ or –45°).

FIGURE 10.3 Line masks.

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

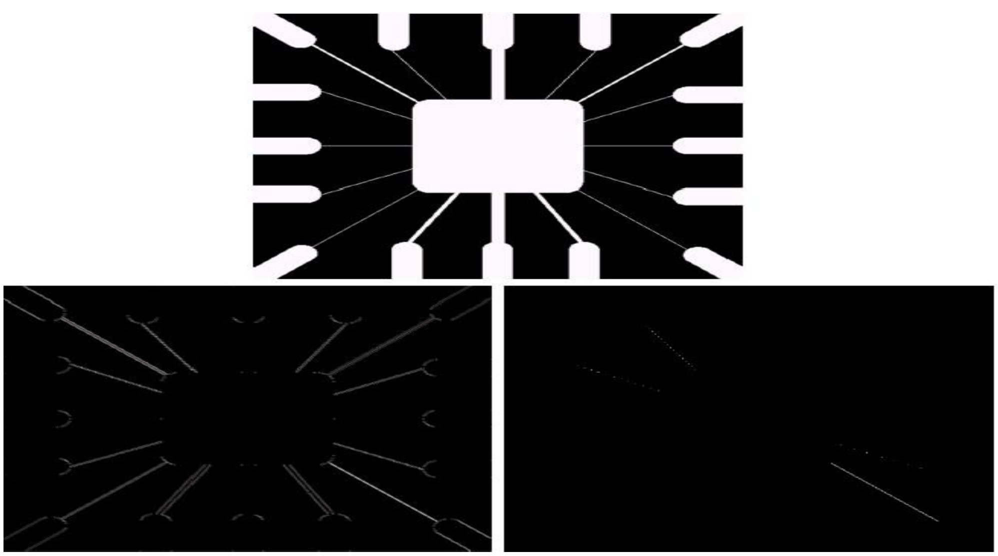
−1	−1	2
−1	2	−1
2	−1	−1

Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

2	−1	−1
−1	2	−1
−1	−1	2



a
b c

FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with −45° line detector. (c) Result of thresholding image (b).

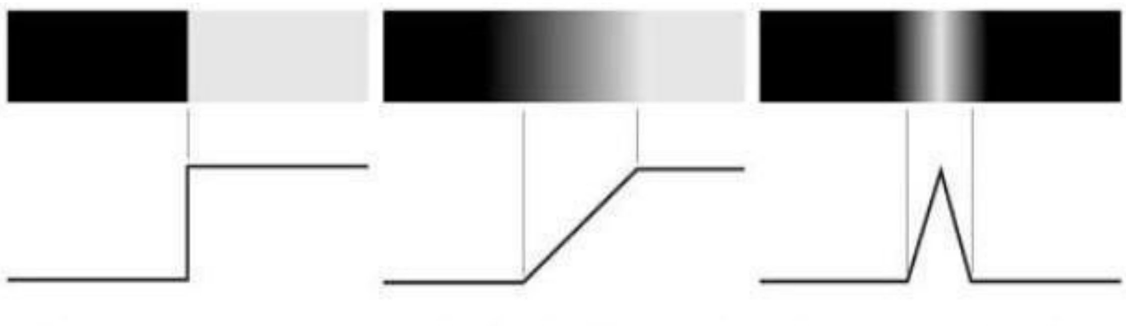
Detection of Discontinuities Edge Detection

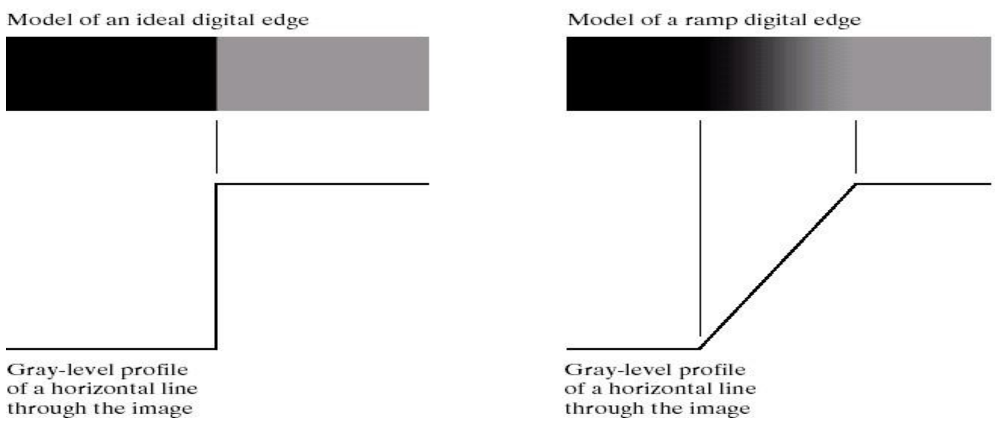
Three different edge types are observed:

1. Step edge – Transition of intensity level overν 1 pixel only in ideal, or few pixels on a more practical use
2. Ramp edge – A slow and graduate transitionν
3. Roof edge – A transition to a different intensity and back. Some kind of spread line

a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.



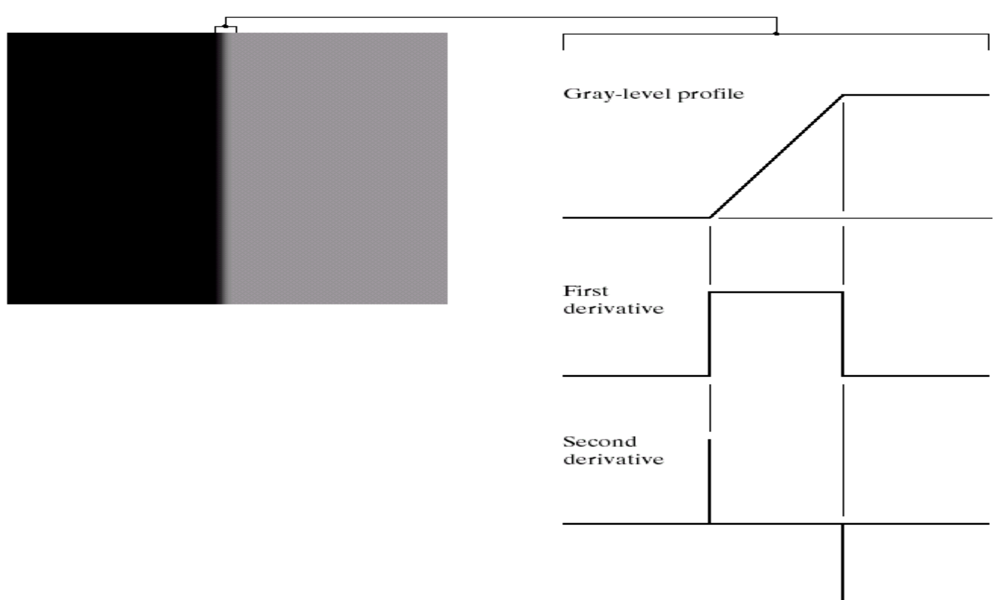


a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

a b

FIGURE 10.6 (a) Two regions separated by a vertical edge. (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.



Detection of Discontinuities Gradient Operators

First-order derivatives:

- The gradient of an image $f(x,y)$ at location (x,y) is defined as the vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- The magnitude of this vector: $\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$

- The direction of this vector: $\alpha(x, y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$

Roberts cross-gradient operators

−1	0
0	1

0	−1
1	0

Roberts

Prewitt operators

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

Sobel operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Prewitt masks for detecting diagonal edges

Sobel masks for detecting diagonal edges

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.

Detection of DiscontinuitiesGradient Operators: Example

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

Second-order derivatives: (The Laplacian)

- The Laplacian of an 2D function $f(x,y)$ is defined as

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

- Two forms in practice:

FIGURE 10.13 Laplacian masks used to implement Eqs. (10.1-14) and (10.1-15), respectively.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

- Consider the function:

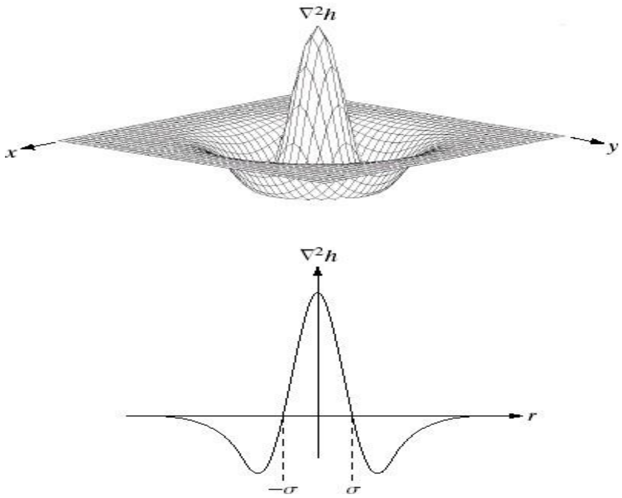
$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \quad \text{where } r^2 = x^2 + y^2$$

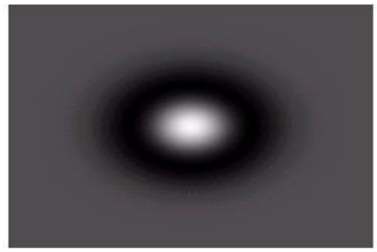
and σ: the standard deviation

- The Laplacian of h is

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

- The Laplacian of a Gaussian sometimes is called the Mexican hat function. It also can be computed by smoothing the image with the Gaussian smoothing mask, followed by application of the Laplacian mask.





0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5 × 5 mask approximation to the shape of (a).

Edge Linking and Boundary Detection

Local Processing

- Two properties of edge points are useful for edge linking:

-the strength (or magnitude) of the detected edge points

-their directions (determined from gradient directions)

- This is usually done in local neighborhoods.
- Adjacent edge points with similar magnitude and direction are linked.
- For example, an edge pixel with coordinates (x_0,y_0) in a predefined neighborhood of (x,y) is similar to the pixel at (x,y) if

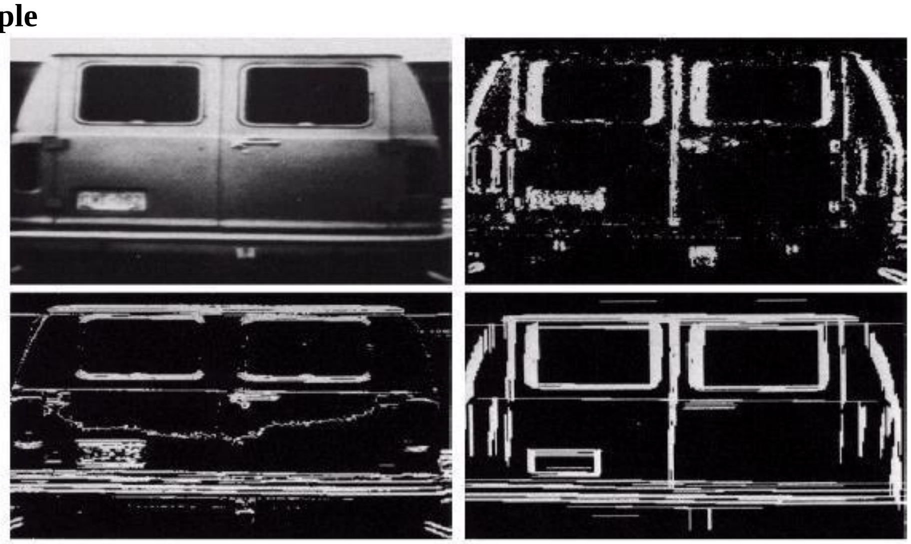
$$\left|\nabla f(x,y) - \nabla(x_0,y_0)\right| \le E, \quad E: \text{a nonnegative threshold}$$

$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| < A, \quad A: \text{a nonegative angle threshold}$$

Local Processing: Example

a b
c d

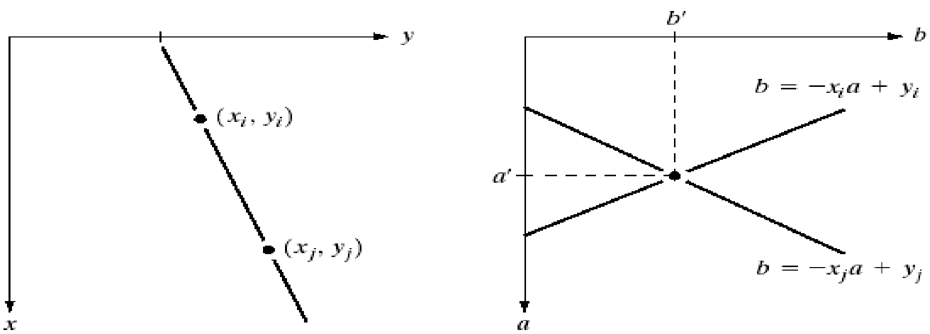
FIGURE 10.16 (a) Input image. (b) G_y component of the gradient. (c) G_x component of the gradient. (d) Result of edge linking. (Courtesy of Perceptics Corporation.)



Global Processing via the Hough Transform

- Hough transform: a way of finding edge points in an image that lie along a straight line.
- Example: *xy*-plane v.s. *ab*-plane (parameter space)

$$y_i = ax_i + b$$

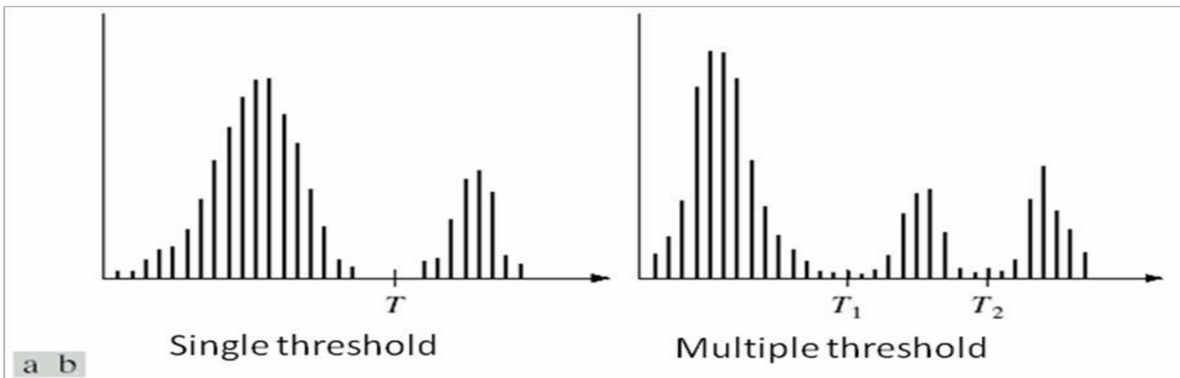


a b
FIGURE 10.17
(a) *xy*-plane.
(b) Parameter space.

<u>Thresholding</u>

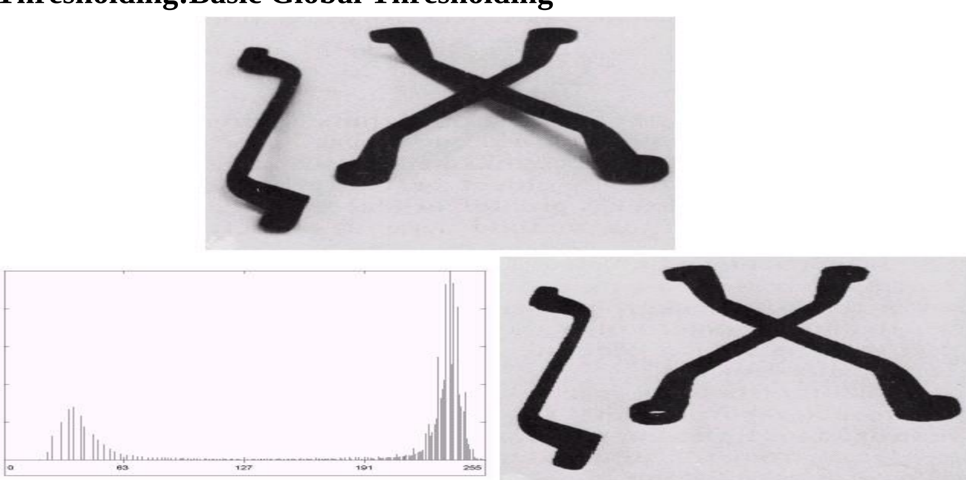
- Assumption: the range of intensity levels covered by objects of interest is different from the background.

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \le T \end{cases}$$



a b
FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

Thresholding:Basic Global Thresholding



a
b c
FIGURE 10.28
(a) Original image. (b) Image histogram.
(c) Result of global thresholding with T midway between the maximum and minimum gray levels.

Thresholding:Basic Adaptive Thresholding

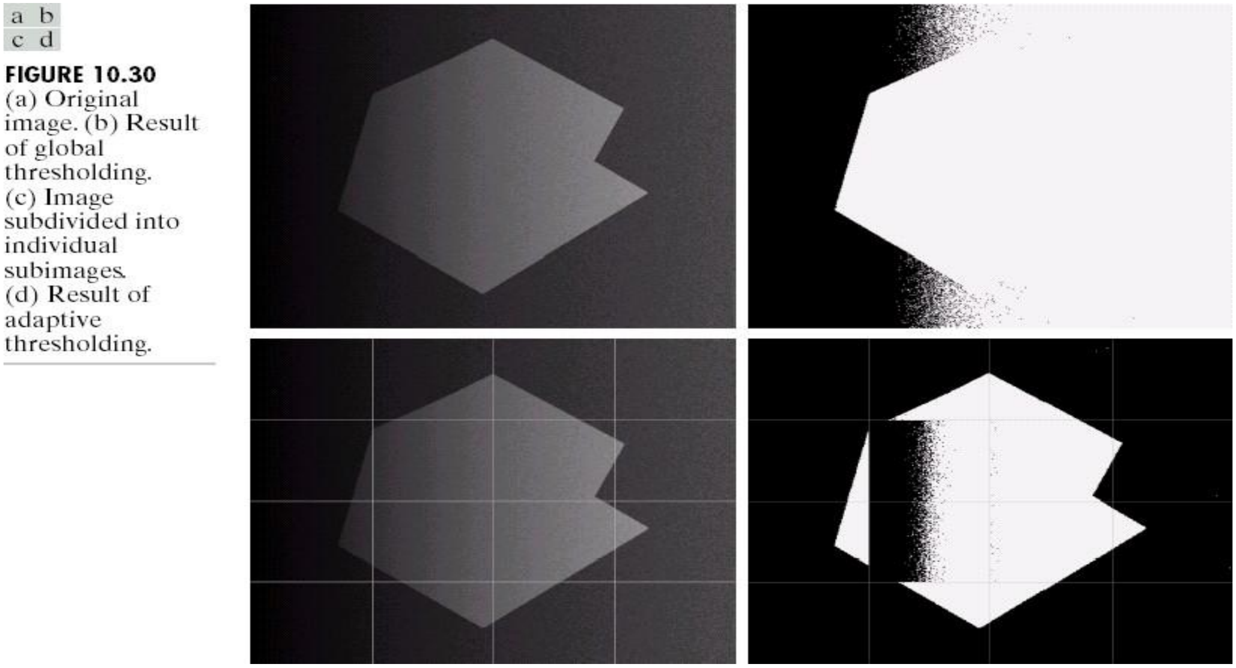


FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

Region-Based Segmentation

- Edges and thresholds sometimes do not give good results for segmentation.
- Region-based segmentation is based on the connectivity of similar pixels in a region.
    - Each region must be uniform.
    - Connectivity of the pixels within the region is very important.

- There are two main approaches to region-based segmentation: region growing and region splitting.

- Let R represent the entire image region.
- Segmentation is a process that partitions R into sub regions, $R_1, R_2, \ldots, R_n$, such that

(a) $\bigcup_{i=1}^{n} R_i = R$

(b) R_i is a connected region, $i = 1,2,..., n$

(c) $R_i \cap R_j = \phi$ for all i and $j, i \neq j$

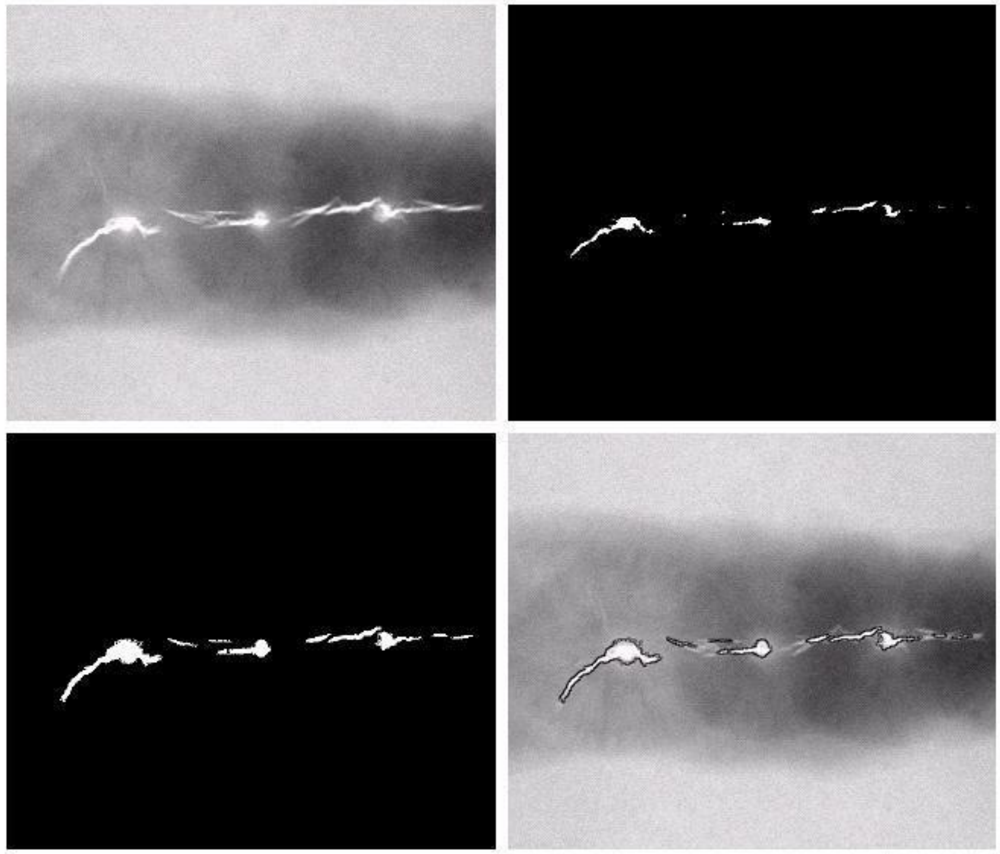
(d) $P(R_i) = \text{TRUE}$ for $i = 1,2,...,n$

(e) $P(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j

where $P(R_k)$: a logical predicate defined over the points in set R_k

For example: $P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region-Based Segmentation:Region Growing



a b
c d

FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).

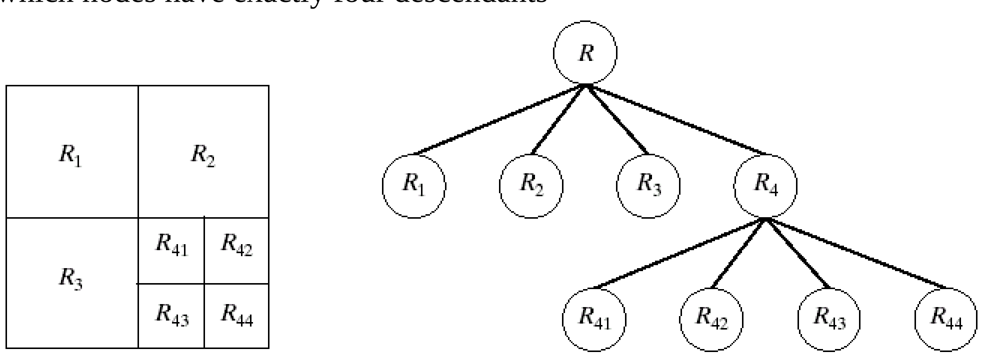
Region-Based Segmentation:Region Splitting and Merging

- Region splitting is the opposite of region growing.
  - ✓ First there is a large region (possible the entire image).
  - ✓ Then a predicate (measurement) is used to determine if the region is uniform.
  - ✓ If not, then the method requires that the region be split into two regions.
  - ✓ Then each of these two regions is independently tested by the predicate (measurement).
  - ✓ This procedure continues until all resulting regions are uniform.

Region Splitting

- The main problem with region splitting is determining where to split a region.
- One method to divide a region is to use a quadtree structure.

Quadtree: a tree in which nodes have exactly four descendants

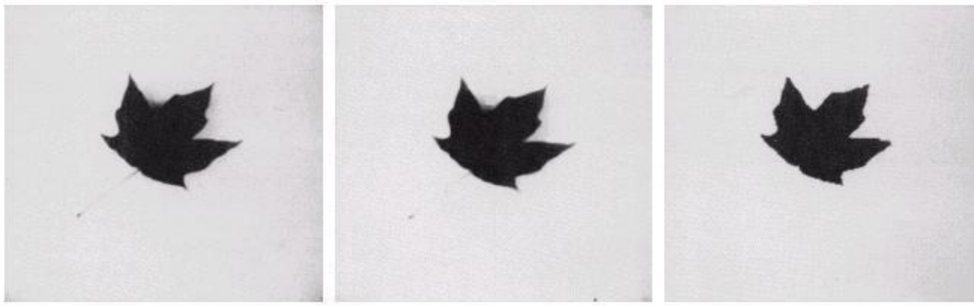


a b

FIGURE 10.42 (a) Partitioned image. (b) Corresponding quadtree.

The split and merge procedure:

- Split into four disjoint quadrants any region R_i for which $P(R_i)$ = FALSE.
- Merge any adjacent regions R_j and R_k for which $P(R_j U R_k)$ = TRUE. (the quadtree structure may not be preserved)
- Stop when no further merging or splitting is possible.



a b c

FIGURE 10.43 (a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).