SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIT – I – Digital Image Processing – SEC1606

I. Digital Image Fundamentals

Elements of Visual Perception; Image Sensing and Acquisition; Image Sampling and Quantization; Basic Relationships between Pixels; Monochromatic Vision Models; Colour Vision Models; Colour Fundamentals; Colour Models

1.1 Elements of Visual Perception

Although the field of digital image processing 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. Hence, developing a basic understanding of human visual perception is necessary. In particular, our interest is in the mechanics and parameters related to how images are formed and perceived by humans.

Structure of Human eye

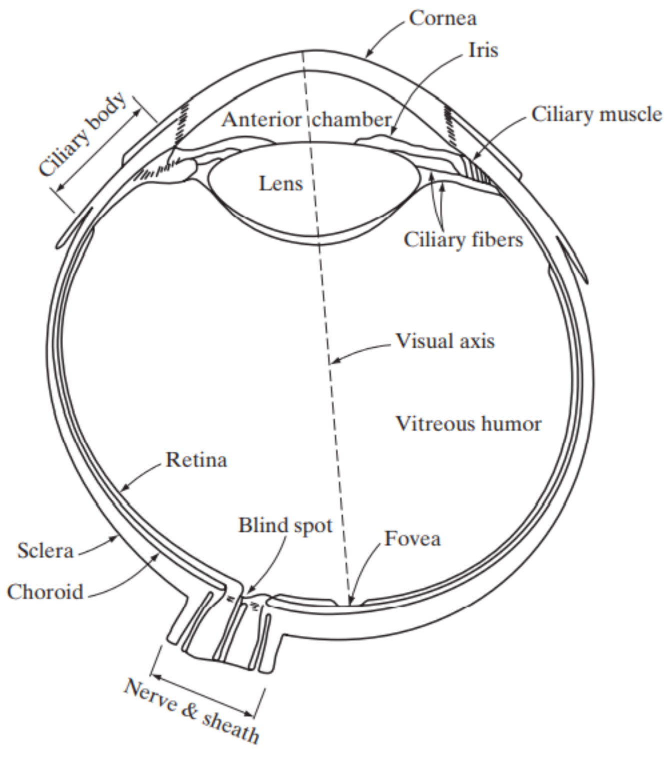


Fig. 1.1 Simplified diagram of a cross section of the human eye.

Figure 1.1 shows a simplified horizontal cross section of the human eye. The eye is nearly a sphere, with an average diameter of approximately 20 mm. Three membranes enclose the eye: the cornea and sclera outer cover; the choroid; and the retina. The cornea is a tough, transparent tissue that covers the anterior surface of the eye. Continuous with the cornea, the sclera is an opaque membrane that encloses the remainder of the optic globe. The choroid lies directly below the sclera. This membrane contains a network of blood vessels that serve as the major source of nutrition to the eye. Even superficial injury to the choroid, often not deemed serious, can lead to severe eye damage as a result of inflammation that restricts blood flow. The choroid coat is heavily pigmented and hence helps to reduce the amount of extraneous light entering the eye and the backscatter within the optic globe. At its anterior extreme, the choroid is divided into the ciliary body and the iris. The latter contracts or expands to control the amount of light that enters the eye. The central opening of the iris (the pupil) varies in diameter from approximately 2 to 8 mm. The front of the iris contains the visible pigment of the eye, whereas the back contains a black pigment. 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 lens is colored by a slightly yellow pigmentation that increases with age. In extreme cases, excessive clouding of the lens, caused by the affliction commonly referred to as cataracts, can lead to poor color discrimination and loss of clear vision. The lens absorbs approximately 8% of the visible light spectrum, with relatively higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed appreciably by proteins with in the lens structure and, in excessive amounts, can damage the eye.

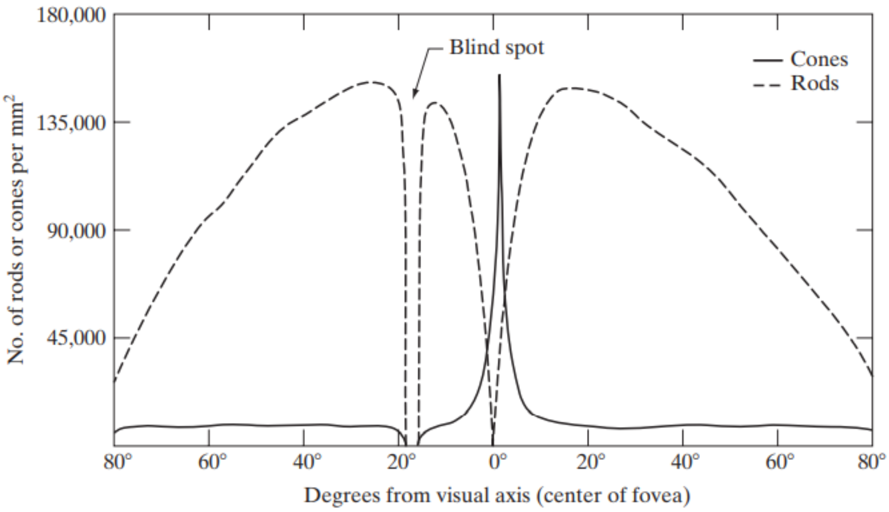


Fig. 1.2 Distribution of rods and cones in the retina.

The innermost membrane of the eye is the retina, which lines the inside of the wall's entire posterior portion. 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. Humans can resolve fine details with these cones largely because each one is connected to its own nerve end. 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. The larger area of distribution and the fact that several rods are connected to a single nerve end reduce the amount of detail discernible by these receptors. Rods serve to give a general, overall picture of the field of view. They are not involved in color vision and are sensitive to low levels of illumination. For example, objects that appear brightly colored in daylight when seen by moonlight appear as colorless forms because only the rods are stimulated. This phenomenon is known as scotopic or dim-light vision. Figure 1.2 shows the density of rods and cones for a cross section of the right eye passing through the region of emergence of the optic nerve from the eye. The absence of receptors in this area results in the so-called blind spot. Except for this region, the distribution of receptors is radially symmetric about the fovea. Receptor density is measured in degrees from the fovea (that is, in degrees off axis, as measured by the angle formed by the visual axis and a line passing through the center of the lens and intersecting the retina). Note in Fig.1.2 that cones are most dense in the center of the retina (in the center area of the fovea). Note also that rods increase in density from the center out to approximately 20° off axis and then decrease in density out to the extreme periphery of the retina. The fovea itself is a circular indentation in the retina of about 1.5 mm in diameter.

Image Formation in the Eye

In an ordinary photographic camera, the lens has a fixed focal length, and focusing at various distances is achieved by varying the distance between the lens and the imaging plane, where the film (or imaging chip in the case of a digital camera) is located. In the human eye, the converse is true; the distance between the lens and the imaging region (the retina) is fixed, and the focal length needed to achieve proper focus is obtained by varying the shape of the lens. The fibers in the ciliary body accomplish this, flattening or thickening the lens for distant or near objects, respectively. The distance between the center of the lens and the retina along the visual axis is approximately 17 mm. The range of focal lengths is approximately 14 mm to 17 mm, the latter taking place when the eye is relaxed and focused at distances greater than about 3 m. The geometry in Fig. 1.3 illustrates how to obtain the dimensions of an image formed on the retina. For example, suppose that a person is looking at a tree 15 m high at a distance of 100 m. Letting h denote the height of that object in the retinal image, the geometry of Fig. 1.3 yields or As indicated, the retinal image is focused primarily on the region of the fovea. Perception then takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that ultimately are decoded by the brain.

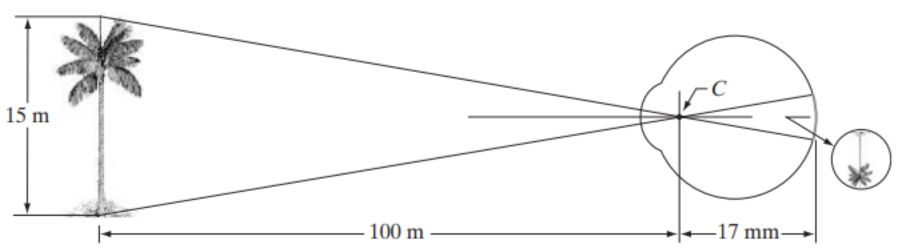


Fig. 1.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens

Brightness Adaptation and Discrimination

Because digital images are displayed as a discrete set of intensities, the eye's ability to discriminate between different intensity levels is an important consideration in presenting image processing results. The range of light intensity levels to which the human visual system can adapt is enormous—on the order of — from the scotopic threshold to the glare limit. Experimental evidence indicates that subjective brightness (intensity as perceived by the human visual system) is a logarithmic function of the light intensity incident on the eye.

1.2 Image Sensing and Acquisition

Most of the 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 system. 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. 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 photoconverter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach. Figure 1.4 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. In this section, we look at the principal modalities for image sensing and generation.

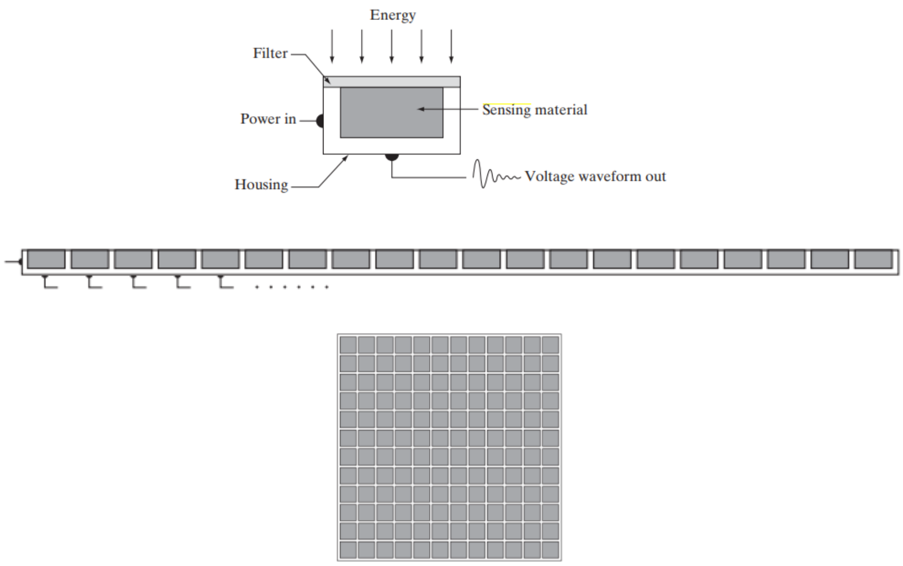


Fig. 1.4 (a) Single imaging sensor. (b) Line sensor. (c) Array sensor.

Image Acquisition Using a Single Sensor

Figure 1.4 (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. In order to generate a 2-D image using a single sensor, there has to be relative displacements in both the x- and y-directions between the sensor and the area to be imaged.

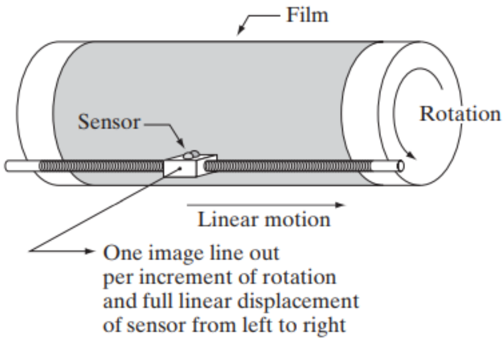


Fig. 1.5 Combining a single sensor with motion to generate a 2-D image.

Figure 1.5 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. Because 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. Another example of imaging with a single sensor places a laser source coincident with the sensor. Moving mirrors are used to control the outgoing beam in a scanning pattern and to direct the reflected laser signal onto the sensor. This arrangement can be used also to acquire images using strip and array sensors, which are discussed in the following two sections.

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. 1.4 (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. 1.6 (a). This is the type of arrangement used in most flat bed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged. One-dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project the area to be scanned onto the sensors. 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. 1.6 (b) shows. A rotating X-ray source provides illumination and the sensors opposite the source collect the X-ray energy that passes 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) imaging.

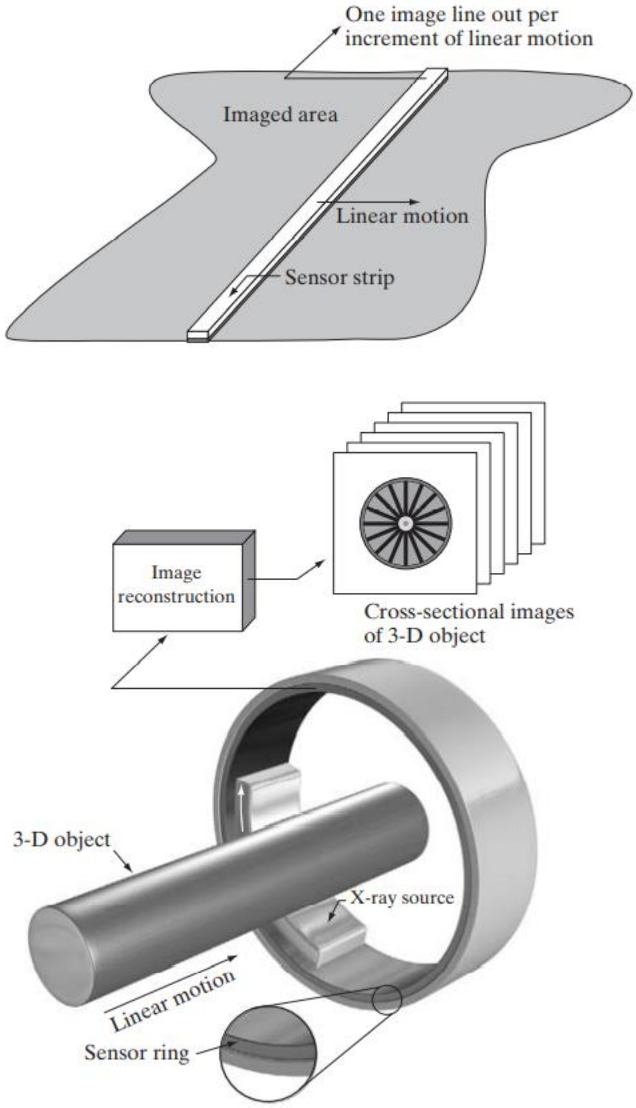


Fig. 1.6 (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

Image Acquisition Using Sensor Arrays

Figure 1.4 (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 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. Because the sensor array in Fig. 1.4 (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. Motion obviously is not necessary, as is the case with the sensor arrangements discussed in the preceding two sections. The principal manner in which array sensors are used is shown in Fig. 1.7. This figure shows the energy from an illumination source being reflected from a scene element (as mentioned at the beginning of this section, the energy also could be transmitted through the scene elements). The first function performed by the imaging system in Fig. 1.7 (c) is to collect the incoming energy and focus it onto an image plane. If the illumination is light, the front end of the imaging system is an optical lens that projects the viewed scene onto the lens focal plane, as Fig. 1.7 (d) shows. The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital and analog circuitry sweep these outputs and convert them to an analog signal, which is then digitized by another section of the imaging system. The output is a digital image, as shown diagrammatically in Fig. 1.7(e).

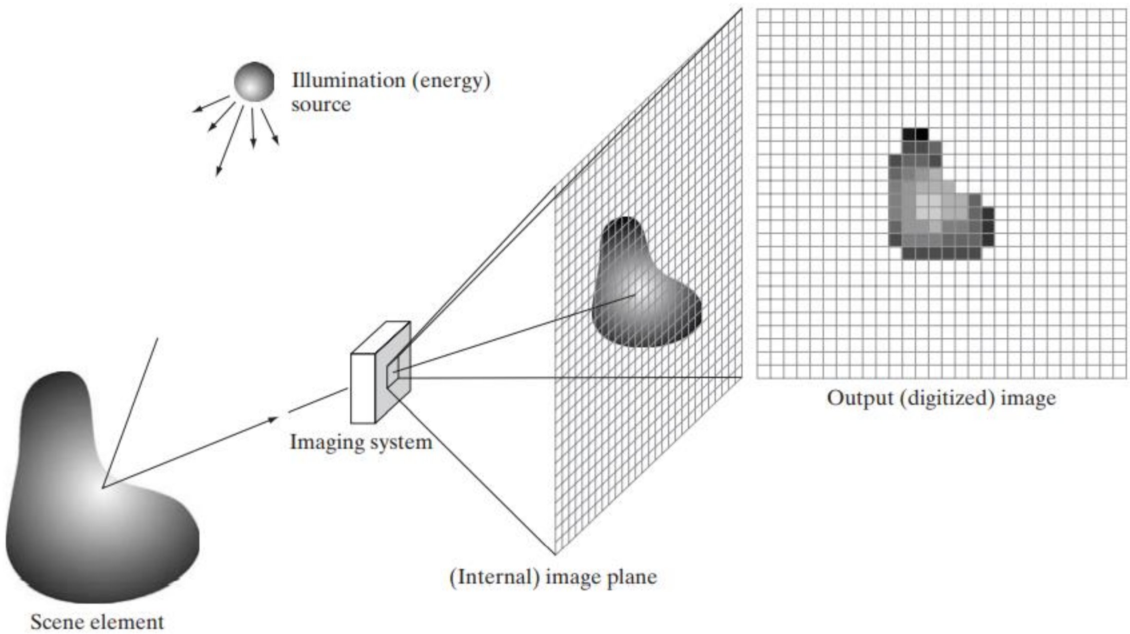


Fig. 1.7 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image

1.3 Image Sampling and Quantization

The basic idea behind sampling and quantization is illustrated in Fig. 1.8. Figure 1.8 (a) shows a continuous image f that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization. The one-dimensional function in Fig. 1.8 (b) is a plot of amplitude (intensity level) values of the continuous image along the line segment AB in Fig. 1.8(a). The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig. 1.8(c). The spatial location of each sample is indicated 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 intensity values. In order to form a digital function, the intensity values also must be converted (quantized) into discrete quantities. The right side of Fig. 1.8(c) shows the intensity scale divided into eight discrete intervals, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight intensity intervals. The continuous intensity levels are quantized by assigning one of the eight values 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 Fig. 1.8 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image. It is implied in Fig. 1.8 that, in addition to the number of discrete levels used, the accuracy achieved in quantization is highly dependent on the noise content of the sampled signal. 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.

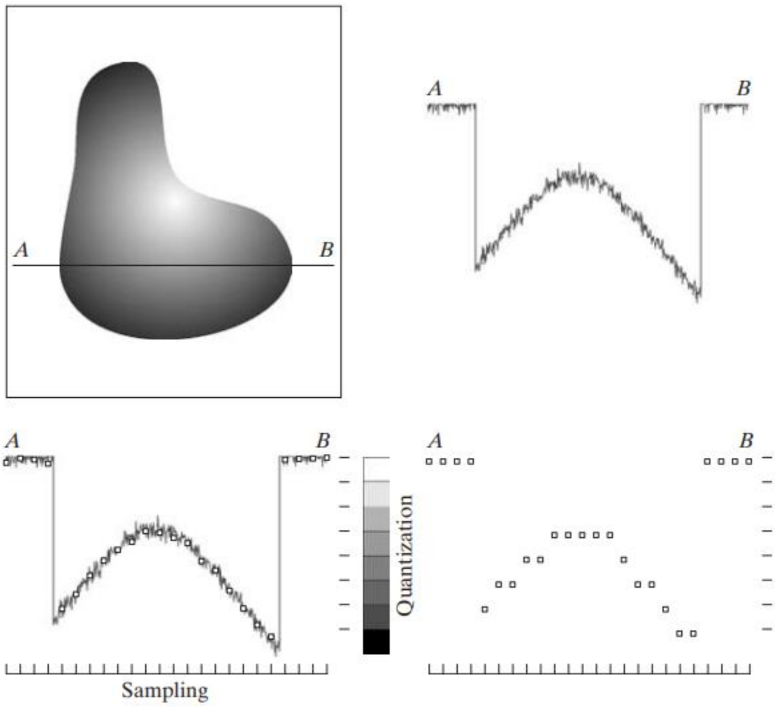


Fig. 1.8 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

When an image is generated by a single sensing element combined with mechanical motion, as in Fig. 1.5, the output of the sensor is quantized in the manner described above. However, spatial 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 using this approach. In practice, limits on sampling accuracy are determined by other factors, such as the quality of the optical components of the system. When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor outputs completes the process of generating a digital image. When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Quantization of the sensor outputs is as before. Figure 1.9 illustrates this concept. Figure 1.9(a) shows a continuous image projected onto the plane of an array sensor. Figure 1.9(b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization.

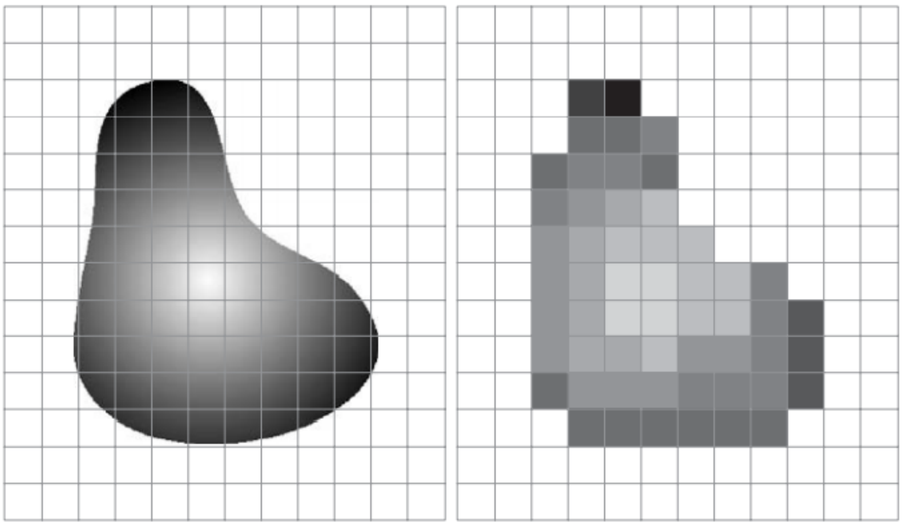


Fig. 1.9 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

1.4 Basic Relationships between Pixels

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 N4(p).

Each pixel is a unit distance from (x, y), and some of the neighbor locations 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 ND(p).

These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N8(p).

Adjacency, Connectivity, Regions, and Boundaries

Let V be the set of intensity values used to define adjacency. In a binary image, V = {1} if we are referring to adjacency of pixels with value 1. In a gray-scale 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 intensity 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

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

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

(i) q is in or

(ii) q is in ND(p) and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.

A (digital) path (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates is a sequence of distinct pixels with coordinates

(x0, y0), (x1, y1), … , (xn, yn)

where (x0, y0) = (x, y), (xn, yn) = (s, t) and pixels (xi, yi) and (xi-1, yi-1) are adjacent for 1 <= i <= n. In this case, n is the length of the path. . If (x0, y0) = (xn, yn) the path is a closed path

Let S represent a subset of pixels in an image. Two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels in S. For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S. If it only has one connected component, then set S is called a connected set.

Let R be a subset of pixels in an image. We call R a region of the image if R is a connected set. Two regions, and are said to be adjacent if their union forms a connected set. Regions that are not adjacent are said to be disjoint. We consider 4- and 8-adjacency when referring to regions.

Distance Measures

For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a distance function or metric if

i. D(p, q) >= 0 (D(p, q) = 0 iff p = q),

ii. D(p, q) = D(q, p) and

iii. D(p, z) < = D(p, q) + D(q, z).

The Euclidean distance between p and q is defined as

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

For this distance measure, the pixels having a distance less than or equal to some value r from (x, y) are the points contained in a disk of radius r centered at (x, y).

The distance (called the city-block distance) between p and q is defined as

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

In this case, the pixels having a D4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D4 distance <=2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

The pixels with D4=1 are the 4-neighbors of (x, y).

The distance (called the chessboard distance) between p and q is defined as

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

In this case, the pixels with D8 distance from (x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D8 distance <= 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

The pixels with D8=1 distance are the 8-neighbors of (D x, y).

1.5 Monochromatic Vision Models

When Light enters the eye,

- optical characteristics are represented by **LPF (Low Pass Filter)** with frequency response $\mathbf{H_l(\xi_1, \xi_2)}$
- spatial response are represented by the **relative luminous efficiency** function $V(\lambda)$, yields the luminance distribution f(x, y) via

$$f(x, y) = \int_0^\infty I(x, y, \lambda) V(\lambda)\, d\lambda$$

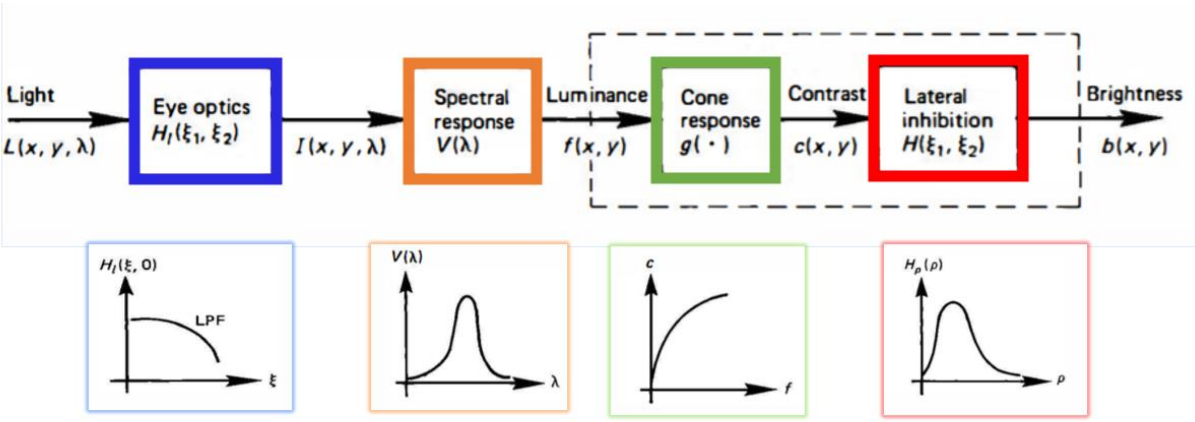


Fig. 1.10 Monochrome Vision Model

- The nonlinear response of the rods and cones, represented by the point nonlinearity **g (·)**, yields the contrast **c(x, y)**
- The lateral inhibition phenomenon is represented by a spatially invariant, isotropic, linear system whose frequency response is $\mathbf{H(\xi_1, \xi_2)}$
- Its output is the neural signal, which represents the apparent brightness **b(x, y)**
- For an optically well-corrected eye, the low-pass filter has a much slower drop-off with increasing frequency than that of the lateral inhibition mechanism
- Thus the optical effects of the eye could be ignored, and the simpler model showing the transformation between the luminance and the brightness suffices

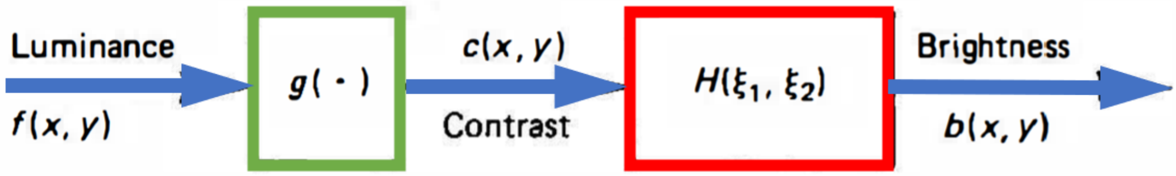


Fig. 1.11 Simplified Monochrome Vision Model

1.6 Colour Vision Models

- The color image is represented by the $\mathbf{R_N}$, $\mathbf{G_N}$, $\mathbf{B_N}$ coordinates at each pixel.
- The matrix A transforms the input into the three **cone responses** $\boldsymbol{\alpha_k}$ **(x, y, C),** k = 1, 2, 3 where (x, y) are the spatial pixel coordinates and C refers to its color
- In Fig., we have represented the **normalized cone responses**
- In analogy with the definition of tristimulus values, Tk are called the retinal cone tristimulus coordinates
- The cone responses undergo nonlinear point transformations to give three fields Tk (x, y), k = 1, 2, 3
- The 3 x 3 matrix B transforms the {f(x, y)} into {Ck(x, y)} such that C1(x, y) is the monochrome (achromatic) contrast field c(x, y), as in simplified model, and C2(x, y) and C3(x, y) represent the corresponding chromatic fields

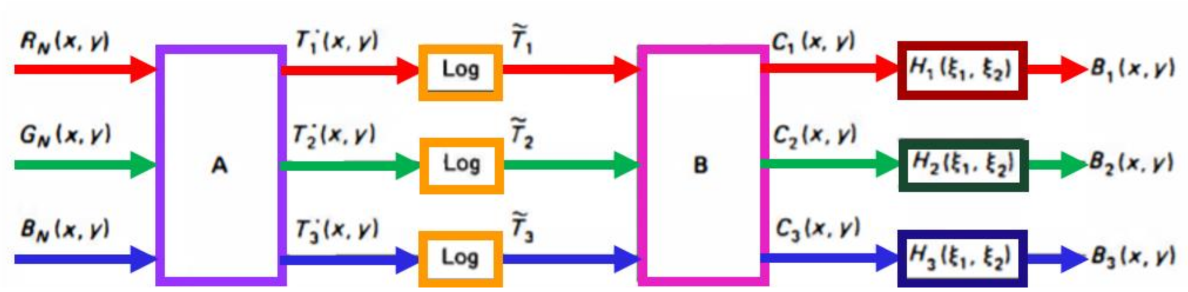


Fig. 1.12 Colour Vision Model

- The spatial filters $H_k(\xi_1 , \xi_2)$, k = 1, 2, 3, represent the frequency response of the visual system to luminance and chrominance contrast signals
- Thus $H1(\xi_1 , \xi_2)$ is the same as H (ξ 1 , ξ 2) in simplified model and is a bandpass filter that represents the lateral inhibition phenomenon

$$T_k \triangleq \frac{\alpha_k(x, y, C)}{\alpha_k(x, y, W)}, \qquad k = 1, 2, 3$$

- The visual frequency response to chrominance signals are not well established but are believed to have their passbands in the lower frequency region, as shown in figure
- The 3 x 3 matrices A and B are given as follows:

$$\mathbf{A} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.127 & 0.724 & 0.175 \\ 0.000 & 0.066 & 1.117 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 21.5 & 0.0 & 0.00 \\ -41.0 & 41.0 & 0.00 \\ -\ 6.27 & 0.0 & 6.27 \end{pmatrix}$$

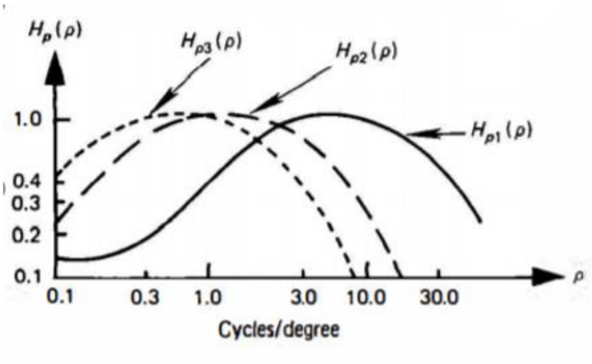


Figure 1.13 Frequency responses of the three color channels C_1, C_2, C_3 of the color vision model. Each filter is assumed to be isotropic so that $H_{pk}(\rho) \stackrel{\Delta}{=} H_k(\xi_1, \xi_2)$, $\rho = \sqrt{\xi_1^2 + \xi_2^2}$, $k = 1, 2, 3$.

- From the model , a criterion for color image fidelity can be defined
- For example, for two color images {RN , GN , BN} and {R,,V, G,,V, BN}, their subjective mean square error could be defined by

$$e_{ls} = \frac{1}{A}\sum_{k=1}^{3}\iint_{\Re}(B_k(x, y) - B_k^{\cdot}(x, y))^2\,dx\,dy$$

Where r is the region which over the image is defined (or available), A is its area. And $\{B_k(x,y)\}$ and $\{B_{kdot}(x,y)\}$ are the outputs of the model for the two colour images.

1.7 Colour Fundamentals

Color spectrum

- In 1666, Sir Isaac Newton
- When a beam of sunlight is passed through a glass prism
- The emerging beam of light consists of a continuous spectrum of colors ranging from violet to red (not white)
- Divided into six broad regions
  - Violet, blue, green, yellow, orange, and red
- Each color blends smoothly into the next

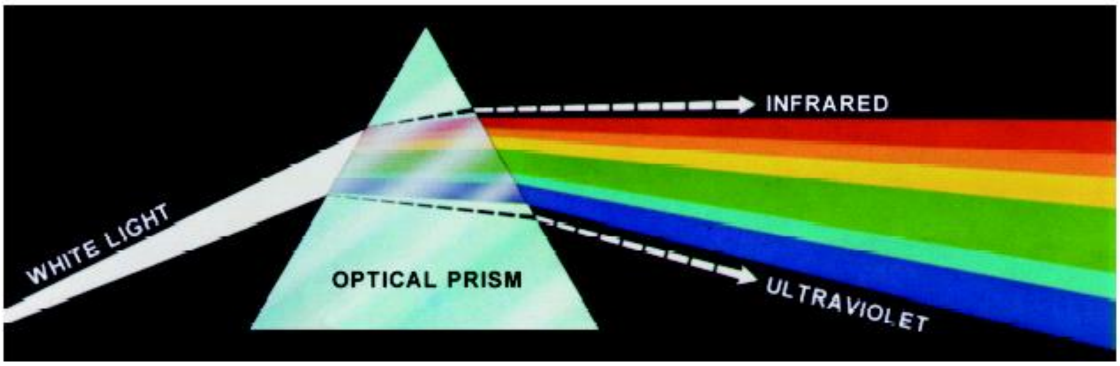


FIGURE 1.14 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

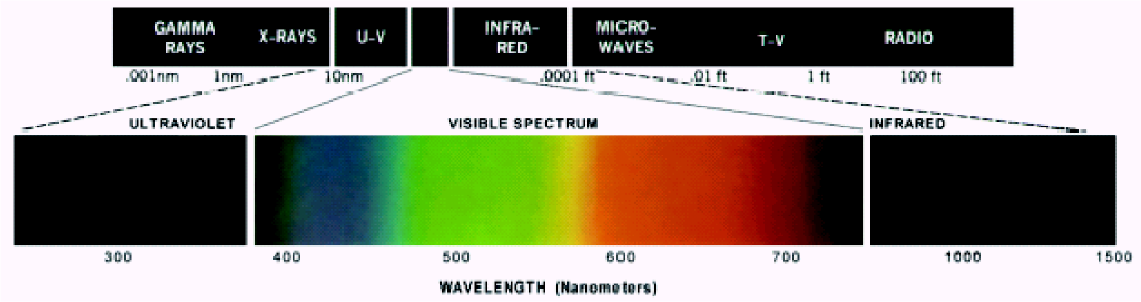


FIGURE 1.15 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Primary colors of pigments or colorants

- cyan, magenta, yellow
- A primary color of pigments is defined as one that subtracts or absorbs a primary color of light and reflects or transmits the other two

Secondary colors of pigments or colorants

- red, green, blue
- Combination of the three pigment primaries, or a secondary with its opposite primary, produces black

Characteristics of colors

- Brightness:
  - The chromatic notion of intensity
- Hue:
  - An attribute associated with the dominant wavelength in a mixture of light waves
  - Representing dominant color as perceived by an observer
- Saturation
  - Referring to relative purity or the amount of white mixed with a hue
  - Saturation is inversely proportional to the amount of white light

Hue and saturation taken together are called chromaticity

- A color may be characterized by its brightness and chromaticity
- The amounts of red, green, and blue needed to form any particular color are called the tristimulus values (Denoted X(red), Y(green), and Z(blue))

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}$$

$$x + y + z = 1 \ , \ X = \frac{x}{y}Y, Z = \frac{z}{y}Y$$

Chromaticity diagram

- Showing color composition as a function of x(R) and y(G)
- For any value of x and y, z(B) = 1 – (x + y)

- The positions of the various spectrum colors are indicated around the boundary of the tongue-shaped
- The point of equal energy = equal fractions of the three primary colors = CIE standard for white light
- Boundary : completely saturated
- Useful for color mixing
- A triangle(RGB) does not enclose the entire color region

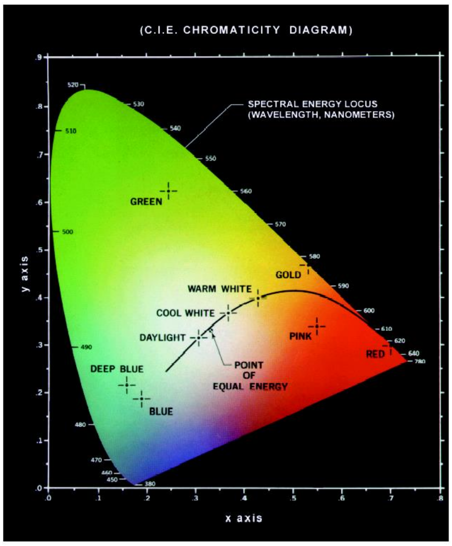


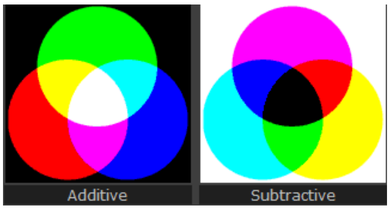
Fig. 1.17 Chromaticity diagram

1.8 Colour Models

The purpose of a color model is to facilitate the specification of colors in some standard Color models are oriented either toward hardware or applications

- Hardware-oriented
  - Color monitor or Video camera : RGB
  - Color printer : CMY
  - Color TV broadcast : YIQ (I : inphase, q : quadrature)
- Color image manipulation : HSI, HSV
- Image processing : RGB, YIQ, HIS

- Additive processes create color by adding light to a dark background (Monitors)
- Subtractive processes use pigments or dyes to selectively block white light (Printers)



Additive Color Mixing			Subtractive Color Mixing		
Red + Green	→	Yellow	Cyan + Magenta	→	Blue
Green + Blue	→	Cyan	Magenta + Yellow	→	Red
Blue + Red	→	Magenta	Yellow + Cyan	→	Green
Red + Green + Blue	→	White	Cyan + Magenta + Yellow	→	Black

Fig. 1.18 Colour Models

RGB color Model

Images represented in the RGB color model consist of three independent image planes, one for each primary color. The number of bits used to represent each pixel in RGB space is called the pixel depth. The term full-color image is used often to denote 24-bit RGB color image

- RGB model is based on a Cartesian coordinate system
- The color subspace of interest is the cube
- RGB values are at three corners
- Colors are defined by vectors extending from the origin
- For convenience, all color values have been normalized
- All values of R, G, and B are in the range [0, 1]

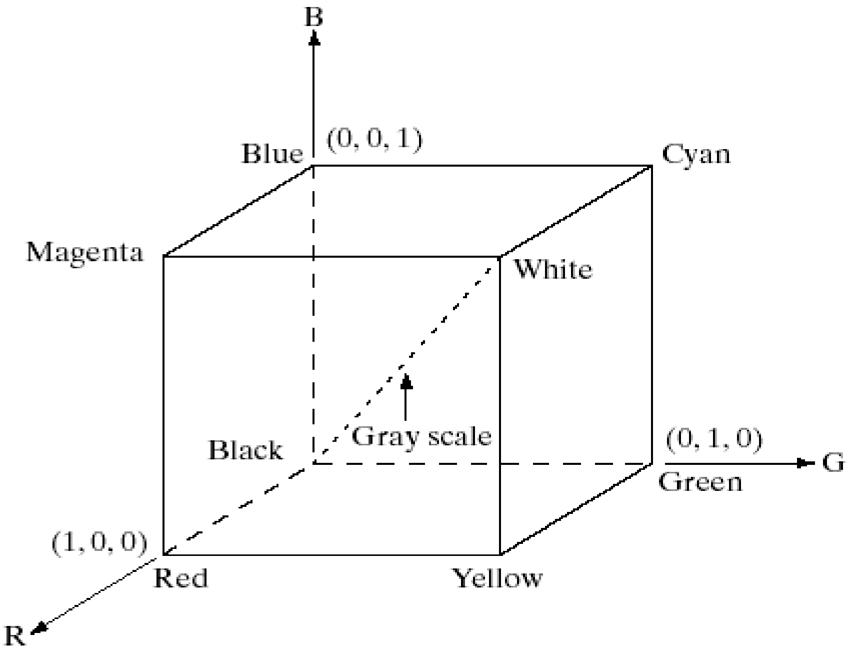


Fig. 1.19 RGB colour cube

CMY Colour Model

General purpose of CMY color model is to generate hardcopy output

- The primary colors of pigments
    - Cyan, Magenta, and Yellow
    - C = W - R, M = W - G, and Y = W - B
- Most devices that deposit colored pigments on paper require CMY data input
    - Converting RGB to CMY
    - The inverse operation from CMY to RGB is generally of no practical interest

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

HIS Colour Model

H : Hue
S : Saturation
I : Intensity

- The intensity is decoupled from the color information
- The hue and saturation are intimately related to the way in which human beings perceive color
- An ideal tool for developing image procession algorithms based on some of the color sensing properties of the human visual system

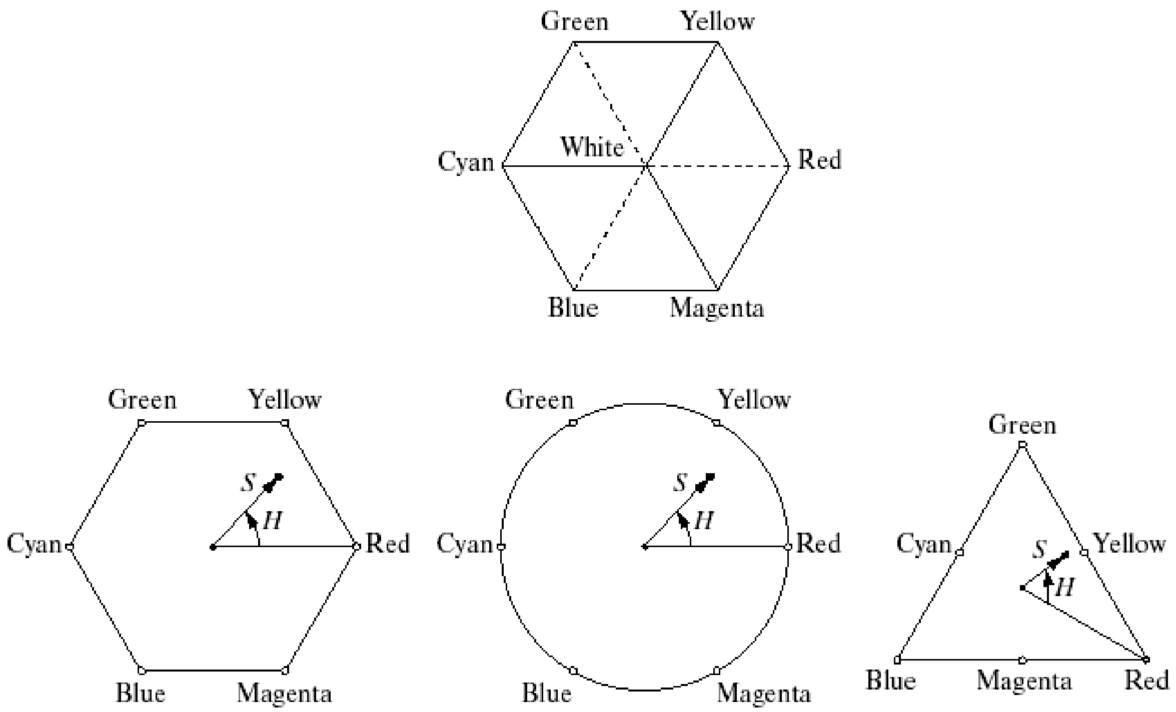


Fig. 1.20 Hue and Saturation in the HIS colour Model

Converting colors from RGB to HSI

- Hue component

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \qquad \theta = \cos^{-1}\left[\frac{\frac{1}{2}\{(R-G)+(R-B)\}}{\{(R-G)^2+(R-B)(G-B)\}^{1/2}}\right]$$

- Saturation component

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

- Intensity component

$$I = \frac{1}{3}(R+G+B)$$

RGB values have been normalized to the range [0,1]

Angle θ is measured with respect to the red axis

Hue can be normalized to the range [0, 1] by dividing by 360°

Converting colors from HSI to RGB

- Three sectors of interest, corresponding to the 120^0 intervals in the separation of primaries

RG sector

$$(0^\circ \leq H < 120^\circ)$$

$$B = I(1-S)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$

$$G = 3I - (R+B)$$

GB sector

$$(120^\circ \leq H < 240^\circ)$$

$$H = H - 120^o$$

$$R = I(1-S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$

$$B = 3I - (R+G)$$

BR sector

$$(240^\circ \leq H < 360^\circ)$$

$$H = H - 240^o$$

$$G = I(1-S)$$

$$B = I\left[1+\frac{S\cos H}{\cos(60^o - H)}\right]$$

$$R = 3I - (G+B)$$

TEXT / REFERENCE BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Education, Inc., 2004.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI Learning Private Limited, New Delhi, 2002.
3. William K. Pratt, “Digital Image Processing”, 3rd Edition, John Wiley & Sons, Inc., 2001.
4. Rafael C.Gonzalez, Richard E.Woods and Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, Inc., 2004.
5. Bernd Jähne, “Digital Image Processing”, 5th Revised and Extended Edition, Springer, 2002.

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIT – II – Digital Image Processing – SEC1606

II. Image Enhancement

Introduction; Point Processing - Image Negatives, Log transformations, Power Law Transformations, Piecewise-Linear Transformation Functions; Arithmetic/Logic Operations - Image Subtraction, Image Averaging; Histogram Processing - Histogram Equalization, Histogram Matching; Spatial filtering - Smoothing, Sharpening; Smoothing Frequency Domain Filters - Ideal Low Pass, Butterworth Low Pass, Gaussian Low Pass; Sharpening Frequency Domain Filters - Ideal High Pass, Butterworth High Pass, Gaussian High Pass.

2.1 Point Processing

Image enhancement is to process the given image such that the result is more suitable to process than the original image. It sharpens the image features such as edges, boundaries or contrast the image for better clarity. It does not increase the inherent information content of the data, but increase the dynamic range of feature chosen. The main drawback of image enhancement is quantifying the criterion for enhancement and therefore large number of image enhancement techniques is empirical and require interactive procedure to obtain satisfactory results. Point Processing is the image enhancement at any point in an image depends only on the gray level at that point. Some of the basic intensity transformation functions are

- **Linear Functions:**
    - Identity Transformation
    - Negative Transformation
- **Logarithmic Functions:**
    - Log Transformation
    - Inverse-log Transformation
- **Power-Law Functions:**
    - n^{th} power transformation
    - n^{th} root transformation
- **Piecewise Transformation function**
    - Contrast Stretching
    - Gray-level Slicing
    - Bit-plane slicing

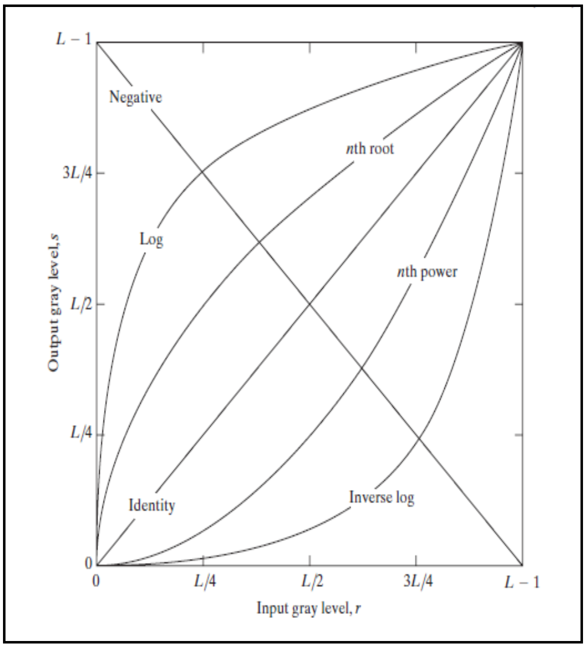


Fig. 2.1 Transformation Functions

Linear transformation

Linear transformation includes

- simple identity and
- negative transformation

- Identity transition is shown by a straight line
  - In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. Hence is called identity transformation

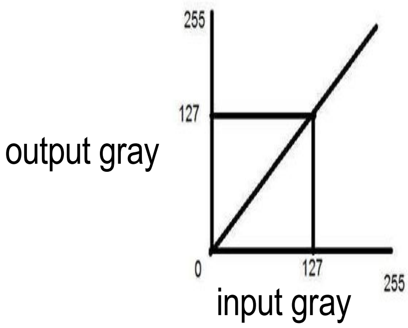
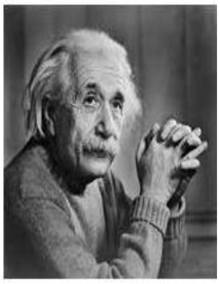


Fig. 2.2 Identity Transformation Function

Negative transformation

The negative of an image with gray level in the range **[0, L-1]** is obtained by using the negative transformation, the expression is given by **s = L-1-r**

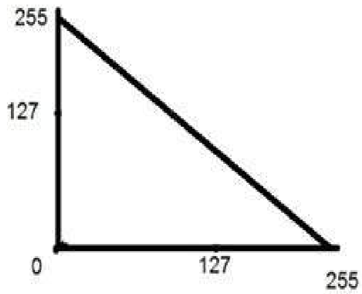
- Reversing the intensity level of an image produces the equivalent of photographic negative
- Suitable for enhancing white or gray detail embedded in dark regions of an image, when the black areas are dominant in size



Input image

Output image

Output gray level

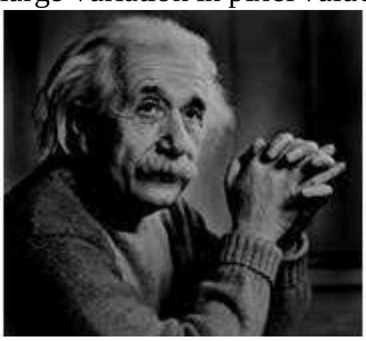


Input gray level

Fig. 2.2 Negative Transformation

Logarithmic transformations

The log transformations can be defined by $s = c \log(r + 1)$ where c is a constant and $r \geq 0$. During log transformation, the dark pixels in an image are expanded and the higher pixel values are compressed. The inverse log transform is opposite to log transform. Log transforms has the important characteristics: it compresses the dynamic range of images with large variation in pixel values.



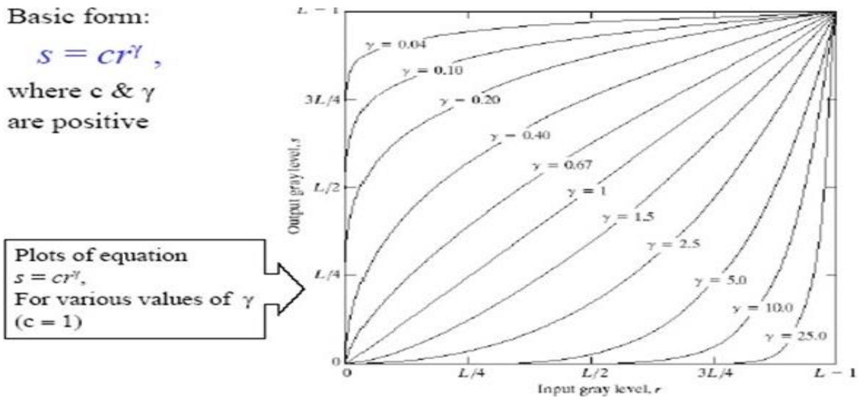
Input image

Output image

Fig. 2.3 Logarithmic Transformation

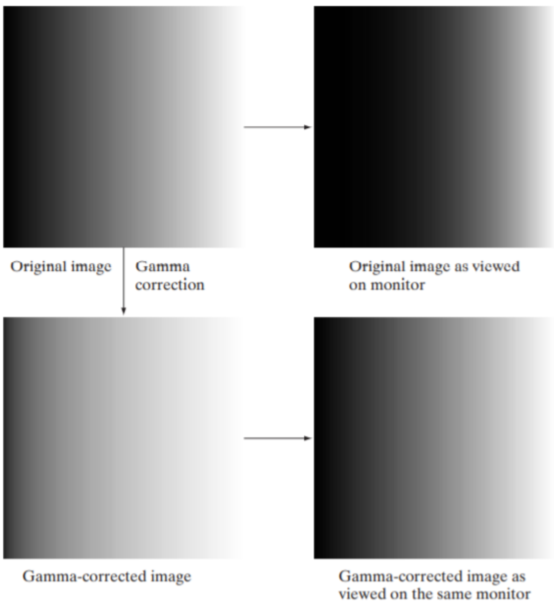
Power – Law transformations

This includes n^{th} power and n^{th} root transformations. It is given by the expression: $s=c\,r^{\gamma}$ (or) $s=c(r+\varepsilon)^{\gamma}$ where γ is called gamma, due to which this transformation is also known as gamma transformation. The exponent in the power law equation is referred to as gamma, the process used to correct this power-law response phenomenon is called gamma correction.



a b
c d

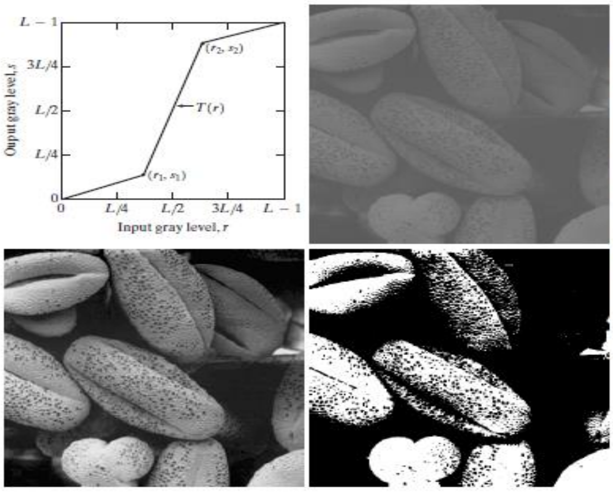
FIGURE 2.4
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).



Piecewise- Linear Transformation

One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images. Low contrast images may result from poor illumination and wrong setting of lens aperture during image acquisition.

Contrast stretching



a b
c d

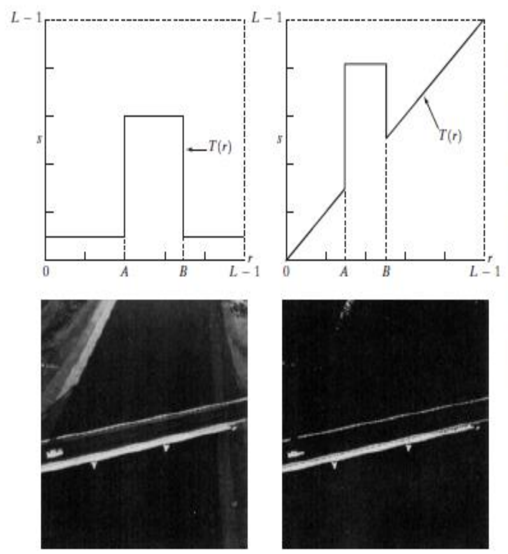
FIGURE 2.5 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Figure shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function. If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels. If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a thresholding function that creates a binary image. Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast. In general, r1 $\leq$ r2 and s1 $\leq$ s2 is assumed, so the function is always increasing. Figure (b) shows an 8-bit image with low contrast. Fig. (c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1]. Finally, Fig. (d) shows the result of using the thresholding function defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

This technique is used to highlight a specific range of gray levels. It can be implemented in several ways, but the two basic themes are:

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. This transformation, shown in Fig. (a), produces a binary image. The second approach, based on the transformation shown in Fig. (b), this brightens the desired range of gray levels but preserves gray levels unchanged Fig.(c) shows a gray scale image, and fig.(d) shows the result of using the transformation in Fig.(a).



a b
c d

FIGURE 2.6
(a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level.
(b) This transformation highlights range [A, B] but preserves all other levels.
(c) An image.
(d) Result of using the transformation in (a).

Bit-plane Slicing

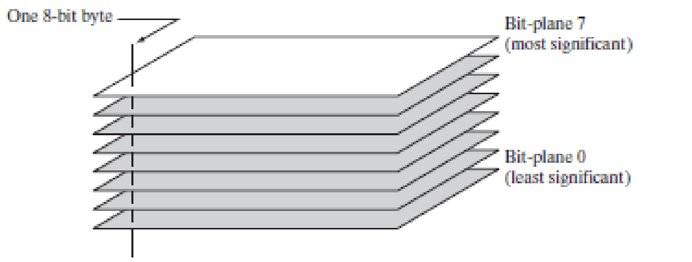


FIGURE 2.7
Bit-plane representation of an 8-bit image.

Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit. This method is useful and used in image compression. Most significant bits contain the majority of visually significant data.

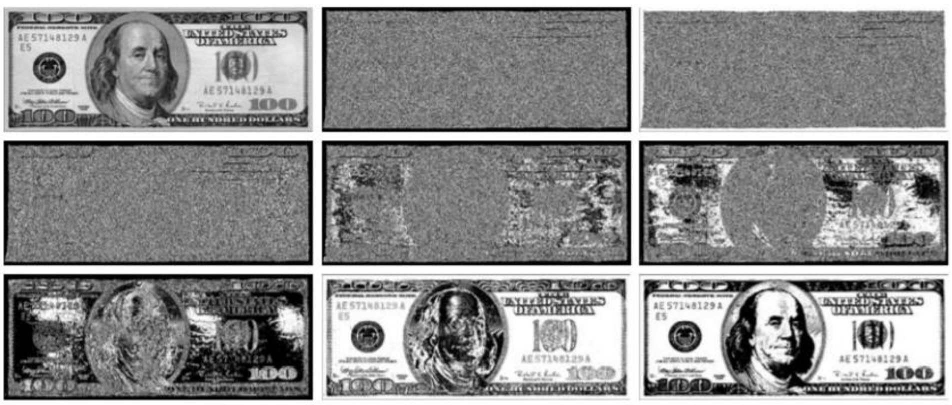


Fig. 2.8 Example for bit plane slicing

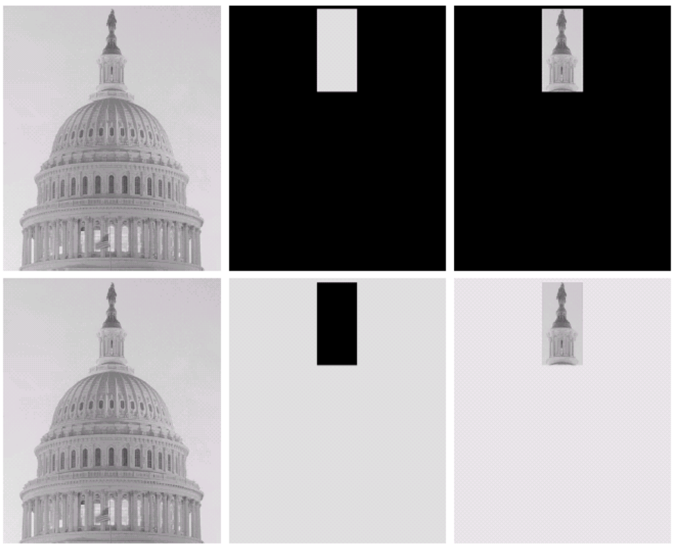
2.2 Arithmetic/Logic Operations

Image arithmetic applies one of the standard arithmetic operations or a logical operator to two or more images. The operators are applied in a pixel-by-pixel way, i.e. the value of a pixel in the output image depends only on the values of the corresponding pixels in the input images. Hence, the images must be of the same size. Although image arithmetic is the most simple form of image processing, there is a wide range of applications.

Logical operators are often used to combine two (mostly binary) images. In the case of integer images, the logical operator is normally applied in a bitwise way.

$$s(x, y) = f(x, y) + g(x, y)$$
$$d(x, y) = f(x, y) - g(x, y)$$
$$p(x, y) = f(x, y) \times g(x, y)$$
$$v(x, y) = f(x, y) \div g(x, y)$$

Arithmetic/logical operations are performed on pixel-by-pixel basis based on two or more images. When dealing with logical operations on gray-scale images, pixel values are processed as strings of binary numbers. In AND and OR image masks, light represents a binary 1 and dark represents a binary 0. Masking refers to as Region of Interest (ROI) processing.



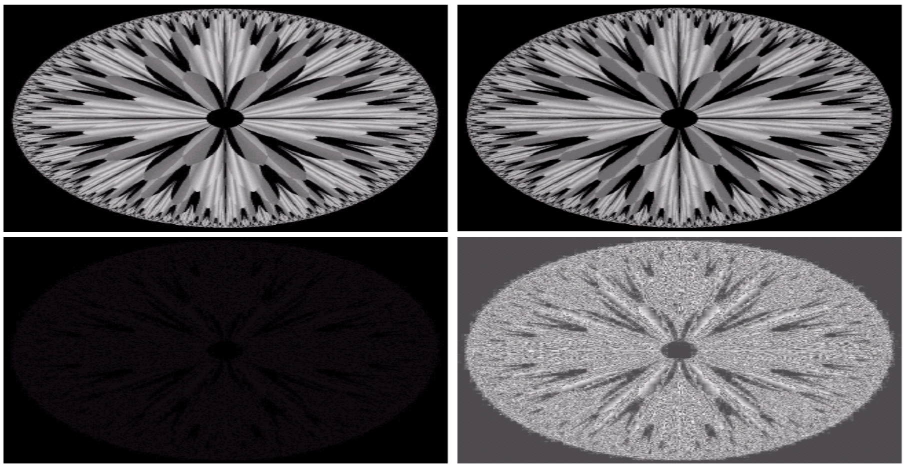
a b c
d e f

FIGURE 2.9 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

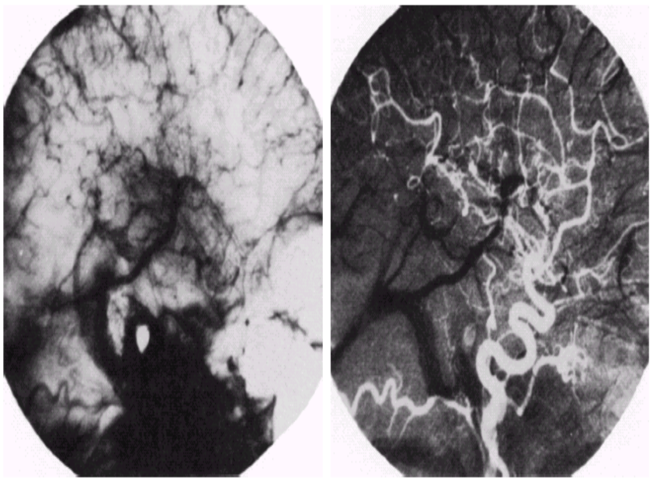
Image Subtraction

- Enhancement of differences between images
- Key usefulness of subtraction is the enhancement of differences between images.
- If the difference in the pixel value is small, then the image appears black when displayed in 8-bit display.
- To bring more detail contrast stretching can be performed

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



a b
c d
FIGURE 2.10
(a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).



a b
FIGURE 2.11
Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

Image Averaging

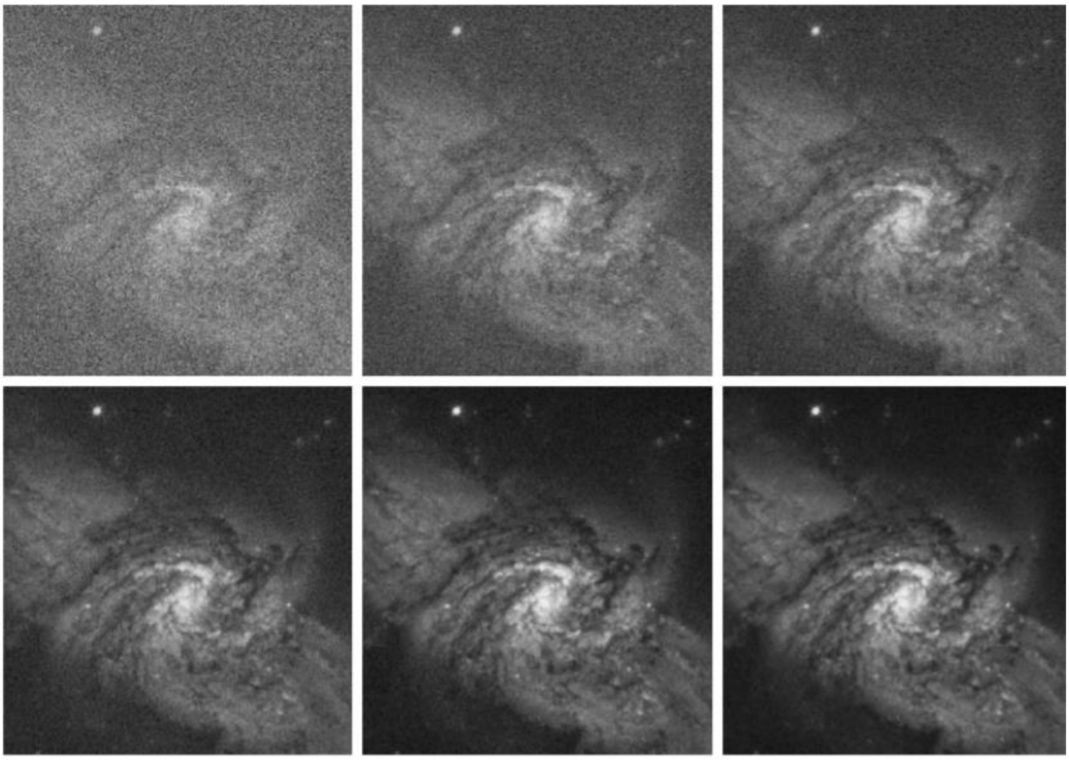
- Noisy image g(x,y) formed by the addition of noise

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

- Averaging K different noisy images $\eta(x,y)$ to an original image f(x,y)
- Objective is to reduce the noise content by adding a set of noisy images $\{g_i(x,y)\}$

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

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)} \qquad \sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$



a b c
d e f

FIGURE 2.12 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

2.4 Histogram Processing

- Histogram is a graphical representation showing a visual impression of the distribution of data
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image
- It plots the number of pixels for each value

- 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 k^{th} gray level and n_k is the number of pixels in the image having gray level r_k

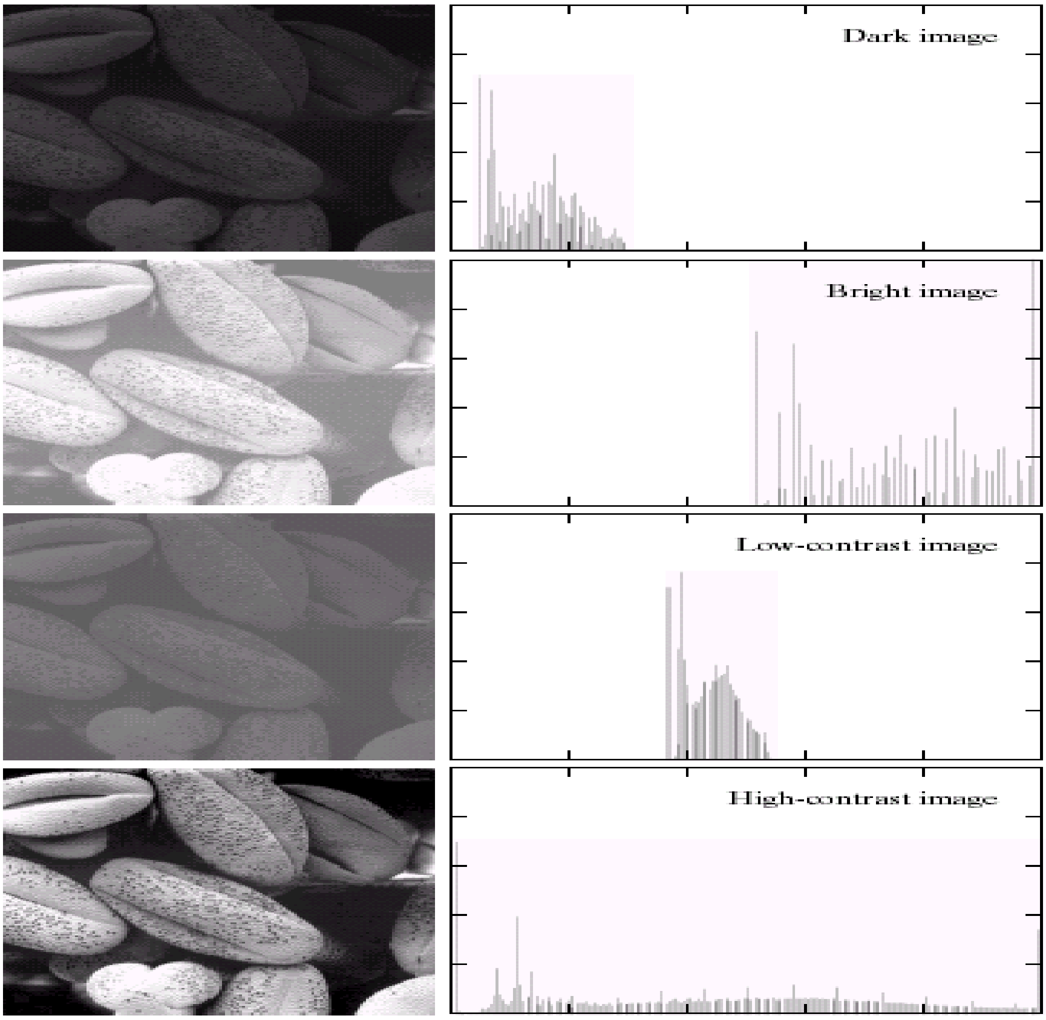


Fig. 2.13 Histogram of different types of images

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.

A normalized histogram is given by

$$p(r_k) = n_k / n \qquad \text{for } k = 0, 1, \ldots, L-1$$

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

<u>Note:</u>

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

Histogram Equalisation

The gray levels in an image is viewed as random variables in the interval[0,1]

Fundamental descriptors of a random variables is its probability density function

$p_s(s)$ and $p_s(r)$ are PDF of s and r

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

Transformation has the particular importance in image processing

$$s = T(r) = \int_0^r p_r(\omega)d\omega$$

Discrete version of transformation- histogram equalization or histogram linearization

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j)$$

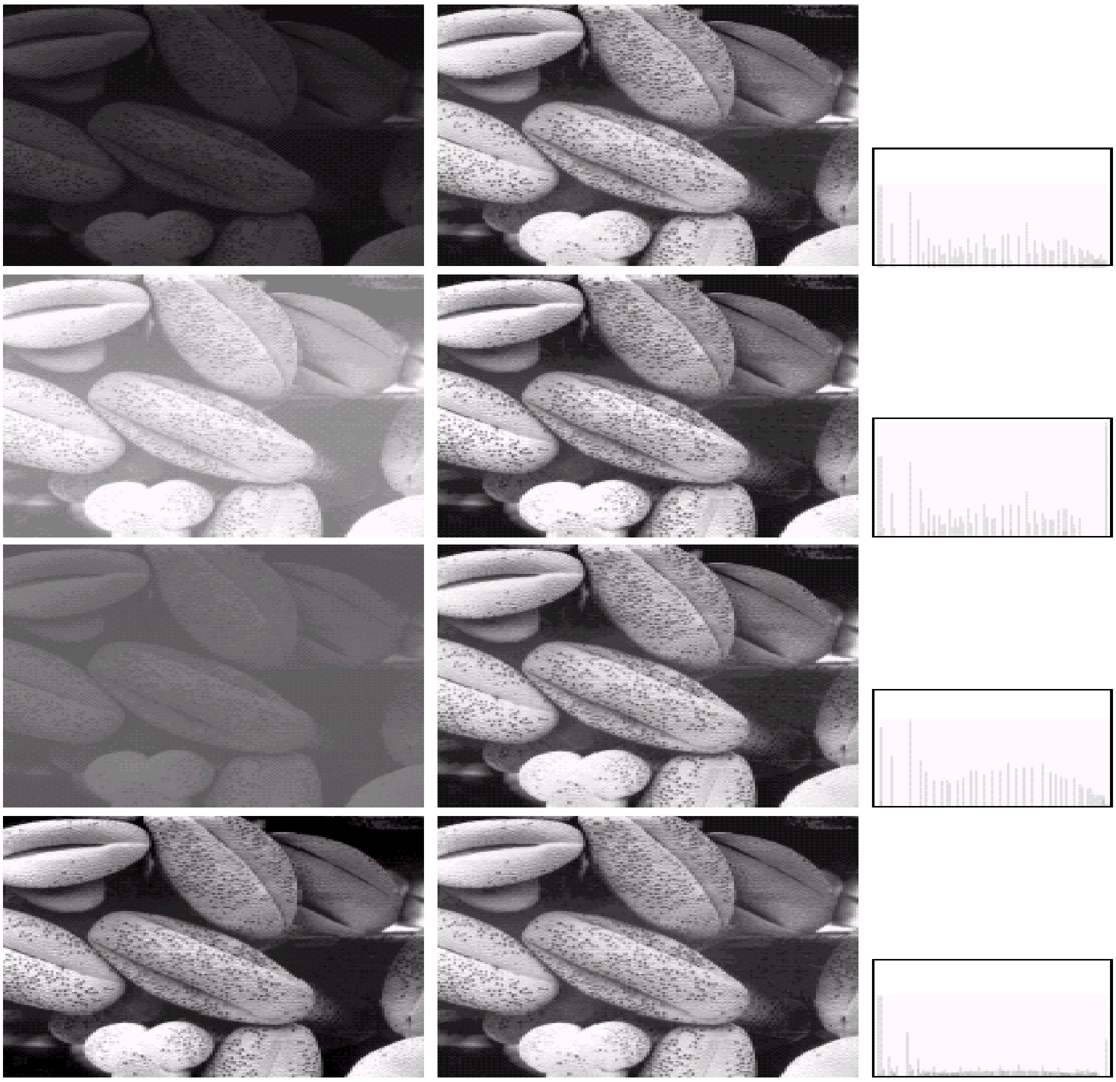


Fig. 2.14 Images after Histogram equalization

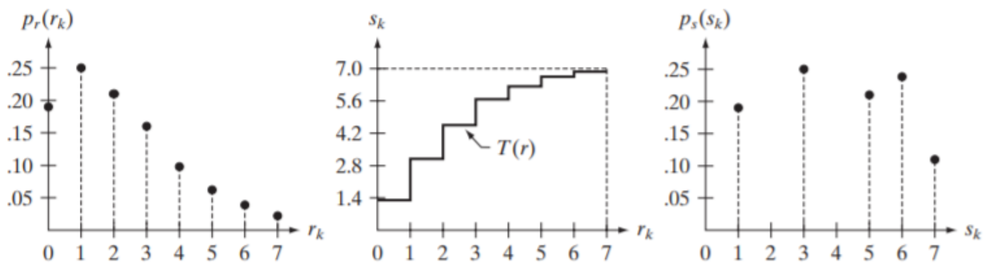
Assume the images have 64 x 64 = 4096 pixels in 8 gray levels. The following table shows the equalization process

Original Image Gray Level	No. of pixels (frequency)	Probability	Cumulative Probability	Multiply by Max. Gray Level	Rounding
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3

2	850	0.21	0.65	4.55	4
3	656	0.16	0.81	5.67	5
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	6
6	122	0.03	0.98	6.86	6
7	81	0.02	1.00	7	7

TABLE 2.1 Intensity distribution and histogram values for a 3-bit, 64 × 64 digital image.

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



a b c

FIGURE 2.15 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

- r is in the range with representing black and representing white
- For r satisfying these conditions, we focus attention on transformations (intensity mappings) of the form

$$s = T(r) \quad 0 \le r \le L-1 \tag{3.3-1}$$

that produce an output intensity level s for every pixel in the input image having intensity r. We assume that:

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

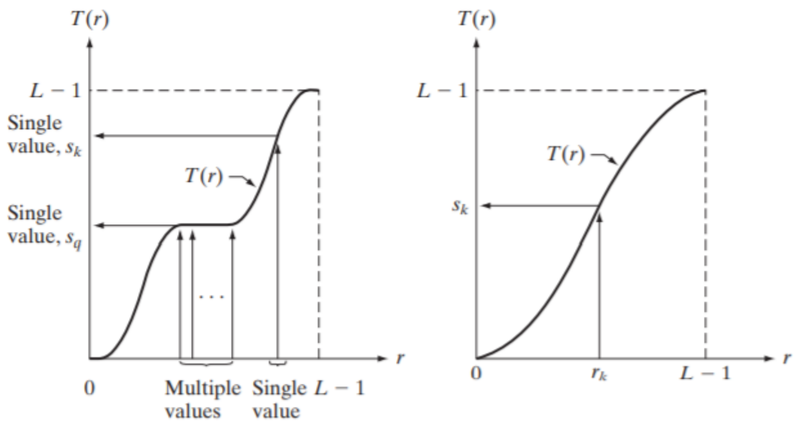
(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

In some formulations to be discussed later, we use the inverse

$$r = T^{-1}(s) \quad 0 \le s \le L-1 \tag{3.3-2}$$

in which case we change condition (a) to

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



a b

FIGURE 2.16 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Histogram Matching (Specification)

Procedure for histogram matching:

- Obtain histogram of given image
- Use the equation to pre compute a mapped level s_k for each level r_k

$$s_k = T_r(k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} n_j / n$$

- Obtain the transformation function G from the given $p_z(z)$ using the equation

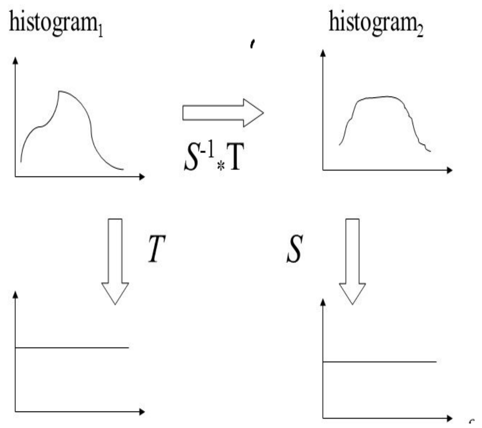
$$(G(\hat{z}) - sk) \ge 0; k = 0,1,2 \ldots L-1$$

- Precompute z_k for each value of s_k using the iterative scheme defined in connection with equation

$$(G(\hat{z}) - sk) \geq 0; k = 0,1,2....L-1$$

- For each pixel in the original image, if the value of that pixel is r_k, map this value to the corresponding level s_k; then map levels s_k into the final level z_k

Processed image that has a specified histogram is called histogram matching or histogram specification.



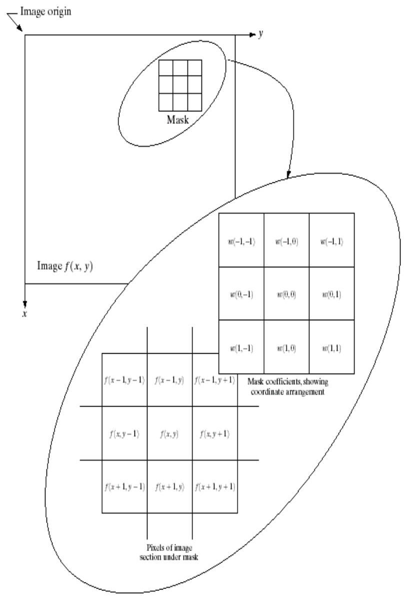
2.5 Spatial Filtering

- The output intensity value at (x,y) depends not only on the input intensity value at (x,y) but also on the specified number of neighboring intensity values around (x,y)
- Spatial masks (also called window, filter, kernel, template) are used and convolved over the entire image for local enhancement (spatial filtering)
- The size of the masks determines the number of neighboring pixels which influence the output value at (x,y)
- The values (coefficients) of the mask determine the nature and properties of enhancing technique.
- The mechanics of spatial filtering
- For an image of size *M* x *N* and a mask of size *m* x *n*
- The resulting output gray level for any coordinates x and y is given by

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

$$\text{where } a = (m-1)/2, \quad b = (n-1)/2$$

$$x = 0, 1, 2, \ldots\ldots, M-1, \quad y = 0, 1, 2, \ldots\ldots, N-1,$$



Given the 3×3 mask with coefficients: $w_1, w_2, \ldots, w_9$

The mask cover the pixels with gray levels: $z_1, z_2, \ldots, z_9$

$$z_5 \leftarrow z = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

z gives the output intensity value for the processed image (to be stored in a new array) at the location of z_5 in the input image

Mask operation near the image border

Problem arises when part of the mask is located outside the image plane; to handle the problem:

- Discard the problem pixels (e.g. $512\text{x}512_{\text{input}}$ $510\text{x}510_{\text{output}}$ if mask size is 3x3)
- Zero padding: expand the input image by padding zeros ($512\text{x}512_{\text{input}}$ $514\text{x}514_{\text{output}}$)
- Zero padding is not good - create artificial lines or edges on the border

- We normally use the gray levels of border pixels to fill up the expanded region (for 3x3 mask). For larger masks a border region equal to half of the mask size is mirrored on the expanded region.

Spatial Filtering for Smoothing

- For blurring/noise reduction;
- Smoothing/Blurring is usually used in preprocessing steps,

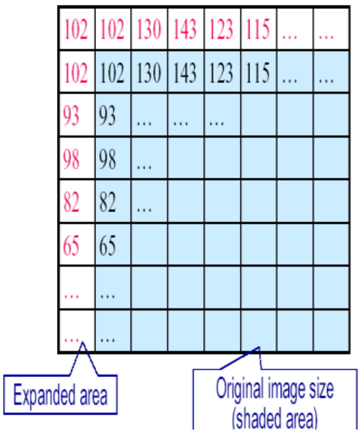
e.g., to remove small details from an image prior to object extraction, or to bridge small gaps in lines or curves

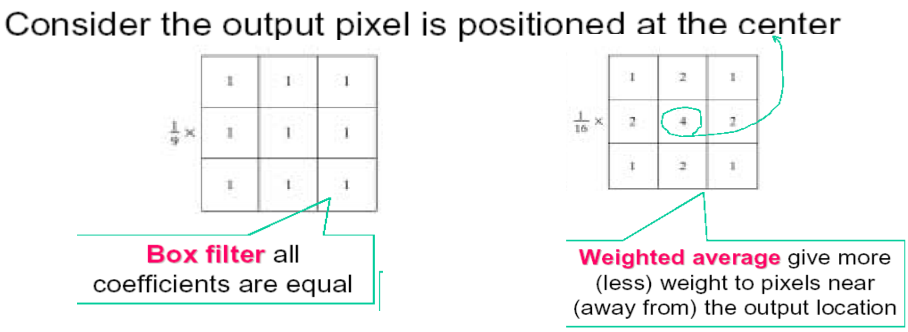
- Equivalent to Low-pass spatial filtering in frequency domain because smaller (high frequency) details are removed based on neighborhood averaging (averaging filters)

Implementation: The simplest form of the spatial filter for averaging is a square mask (assume $m\times m$ mask) with the same coefficients $1/m^2$ to preserve the gray levels (averaging).

Applications: Reduce noise; smooth false contours

Side effect: Edge blurring





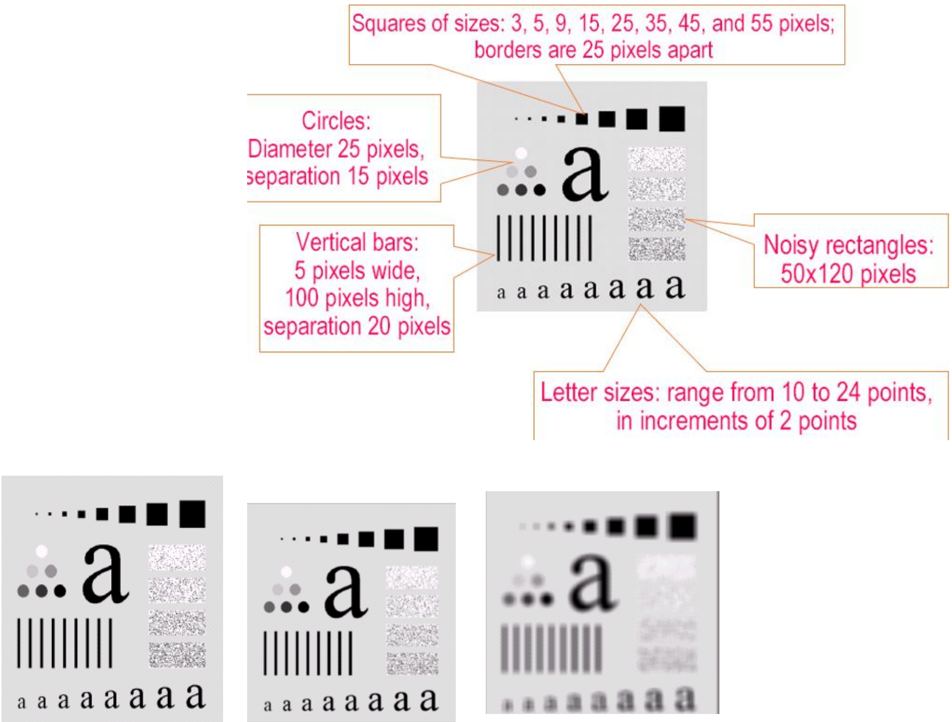


Fig. 2.17 Spatial filtering

Spatial Filtering for Sharpening

Background: to highlight fine detail in an image or to enhance blurred detail

Applications: electronic printing, medical imaging, industrial inspection, autonomous target detection (smart weapons)

Foundation:

- Blurring/smoothing is performed by spatial averaging (equivalent to integration)
- Sharpening is performed by noting only the gray level changes in the image that is the differentiation

Operation of Image Differentiation

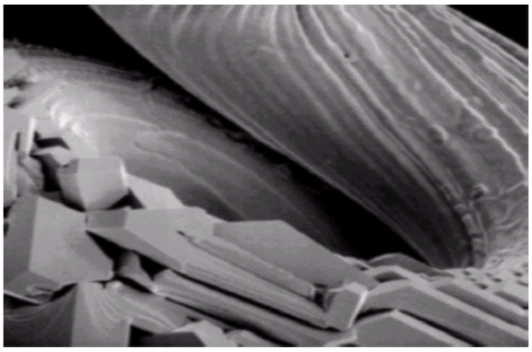
- Enhance edges and discontinuities (magnitude of output gray level >>0)
- De-emphasize areas with slowly varying gray-level values (output gray level: 0)

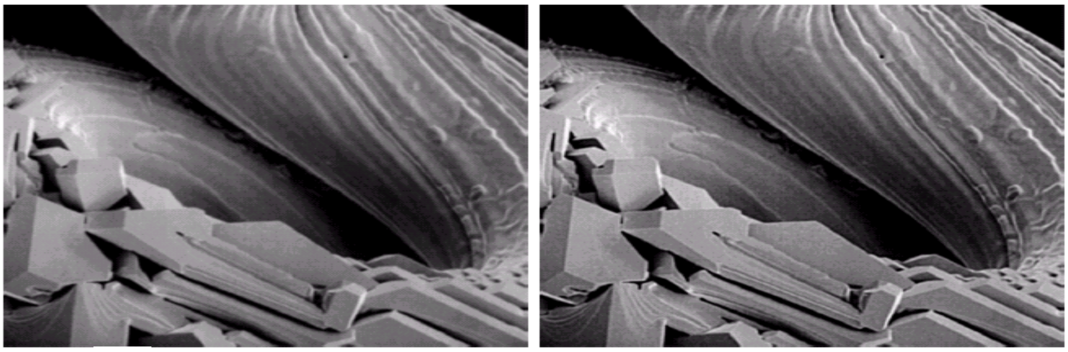
Mathematical Basis of Filtering for Image Sharpening

- First-order and second-order derivatives
- Approximation in discrete-space domain
- Implementation by mask filtering

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





a b c
d e

FIGURE 2.18 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

2.6 Frequency Domain Filters

- 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.

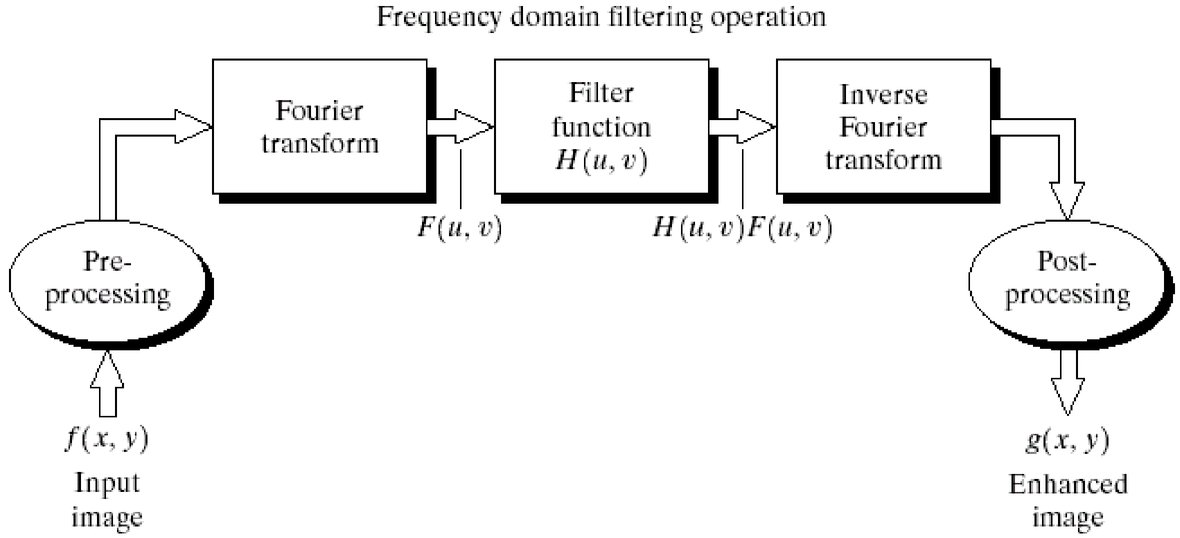


Fig. 2.19 Frequency domain operations

Frequency Domain Filters - Smoothing

Ideal Low Pass Filter

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

where D(u,v) is the distance to the center freq.

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

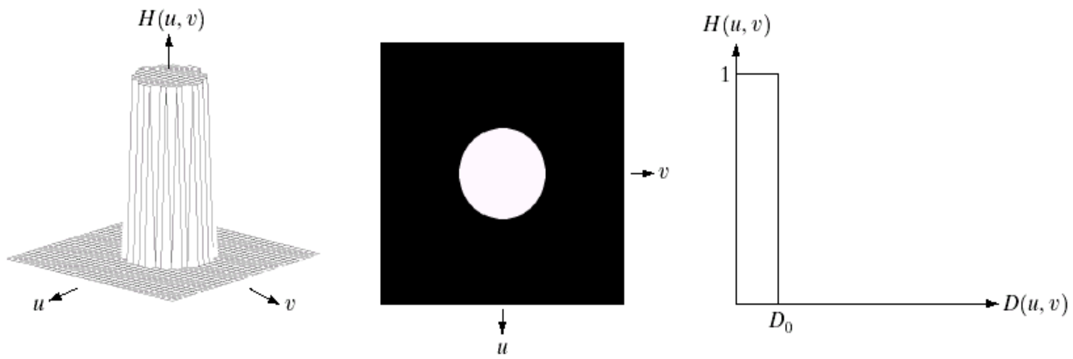


Fig. 2.20 Ideal Low Pass filter

Butterworth Low Pass Filter

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

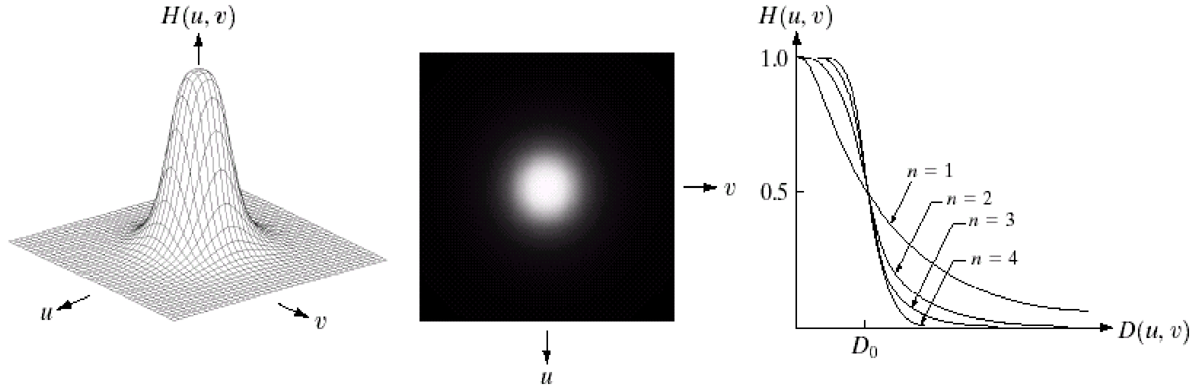


Fig. 2.21 Butterworth Low Pass filter

Gaussian Low Pass Filter

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

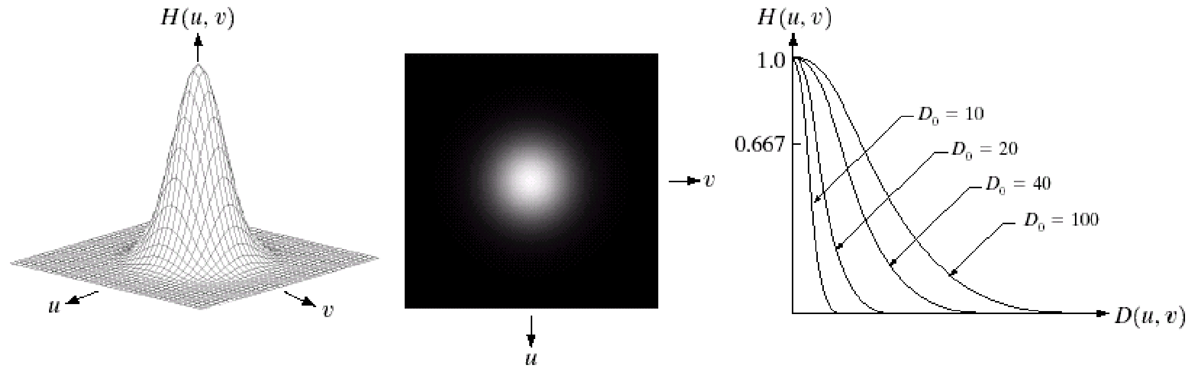


Fig. 2.22 Gaussian Low Pass filter

Frequency Domain Filters - Sharpening

- Image details corresponds to high-frequency
  - Sharpening: high-pass filters
  - $H_{hp}(u,v)=1-H_{lp}(u,v)$

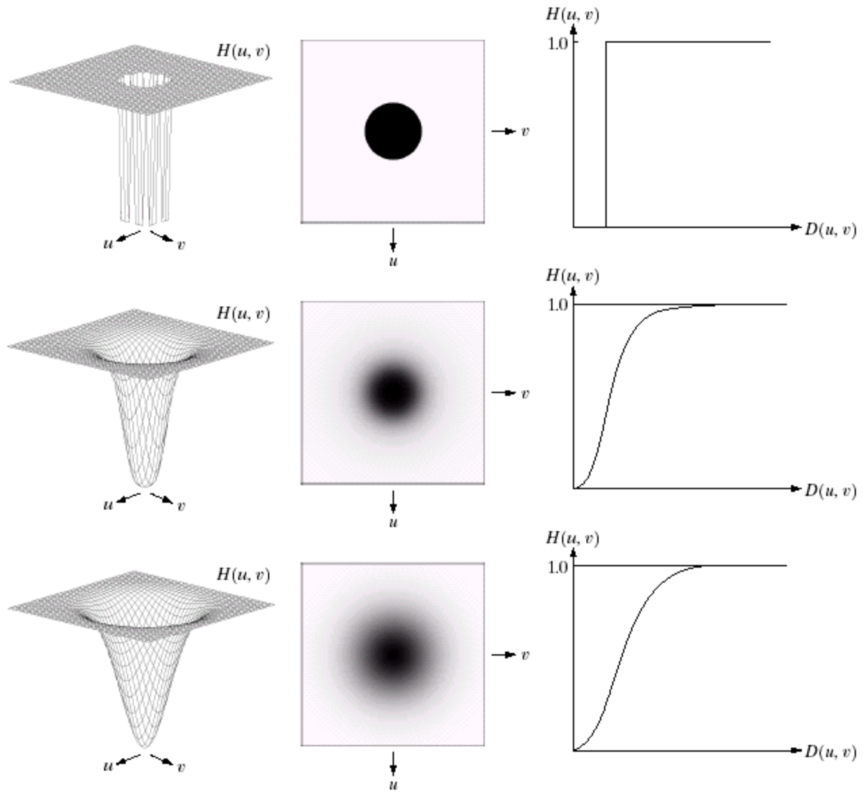


Fig. 2.23 High Pass filter a) Ideal b) Butterworth c) Gaussian

TEXT / REFERENCE BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Education, Inc., 2004.

2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI Learning Private Limited, New Delhi, 2002.

3. William K. Pratt, “Digital Image Processing”, 3rd Edition, John Wiley & Sons, Inc., 2001.

4. Rafael C.Gonzalez, Richard E.Woods and Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, Inc., 2004.

5. Bernd Jähne, “Digital Image Processing”, 5th Revised and Extended Edition, Springer, 2002.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIT – III – Digital Image Processing – SEC1606

III. Image Restoration

A Model of Image Degradation/Restoration Process; Noise Models; Inverse Filtering, Minimum Mean Square Error Filtering, Constrained Least Square Filtering; Geometric Mean Filter; Geometric Transformations - Spatial Transformations, Gray-Level Interpolation.

3.1 A Model of Image Degradation/Restoration Process

Degradation function along with some additive noise operates on f(x, y) to produce degraded image g(x, y).

- Given g(x, y), some knowledge about the degradation function H and additive noise η(x, y), objective of restoration is to obtain estimate
- f'(x, y) of the original image
- If H is linear, position invariant process then degraded image in spatial domain is given by: $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$
- h(x, y) = Spatial representation of H; * indicates convolution

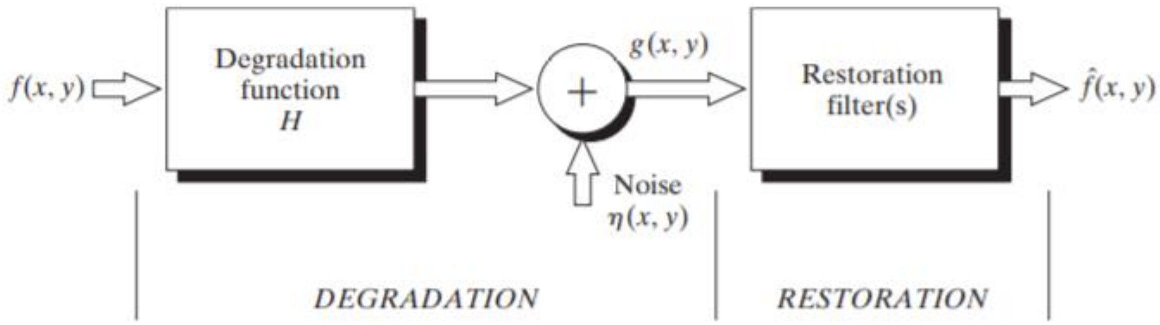


Fig. 3.1 Image degradation and restoration model

In frequency domain.

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

3.2 Noise Models

The sources of noise in digital images arise during image acquisition (digitization) and transmission. Imaging sensors can be affected by ambient conditions. Interference can be added to an image during transmission.

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

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

Where f(x, y) is the original image pixel, η(x, y) is the noise term and g(x, y) is the resulting noisy pixel. If we can estimate the noise model we can figure out how to restore the image. There are many different models for the image noise term η(x, y):

Gaussian Noise:

- Most frequently used noise model
- PDF of Gaussian random variable z is given by: $p(z) = \frac{1}{\sqrt{(2\pi)}\sigma} e^{-(z-\mu)^2/2\sigma^2}$
  - z is Gray level
  - μ is Mean of average value of z
  - σ is Standard Deviation of z
  - σ^2 is Variance of z

When z is defined by this equation then

- About 70% of its values will be in the range [(μ - σ),(μ + σ)] and
- About 95% of its values will be in the range [(μ - 2σ),(μ + 2σ)]

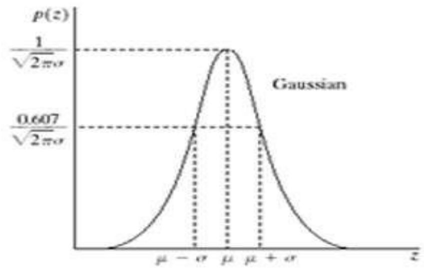


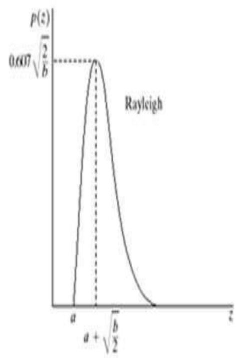
Fig. 3.2 Gaussian Noise

Rayleigh Noise

PDF of Rayleigh Noise is given by: $p(z)=\begin{cases}\frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a\\ 0 & \text{for } z \prec a\end{cases}$

- z is Gray level
- μ is Mean of average value of z
- σ^2 is Variance of z

Fig 3.3 Rayleigh Noise



Erlang (Gamma) Noise

PDF of Erlang Noise is given by: $p(z)=\begin{cases}\frac{a^b z^{b-1}}{(b-1)!}e^{-z} & \text{for } z \geq 0\\ 0 & \text{for } z \prec 0\end{cases}$

- z is Gray level
- μ is Mean of average value of z
- σ^2 is Variance of z

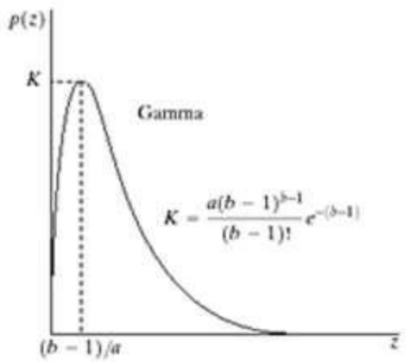


Fig 3.4 Erlang Noise

Exponential Noise

PDF of Exponential Noise is given by: $p(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$

- z is Gray level
- μ is Mean of average value of z
- σ^2 is Variance of z
- a > 0

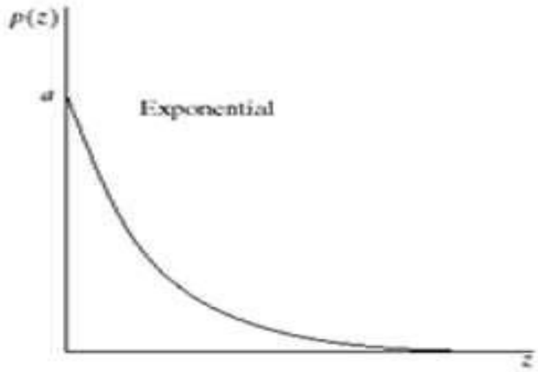


Fig 3.5 Exponential Noise

Uniform Noise

PDF of Uniform Noise is given by: $p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$

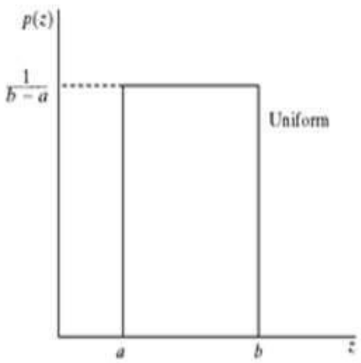


Fig 3.6 Uniform Noise

Impulse (Salt and pepper) Noise

PDF of impulse Noise is given by: $p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$

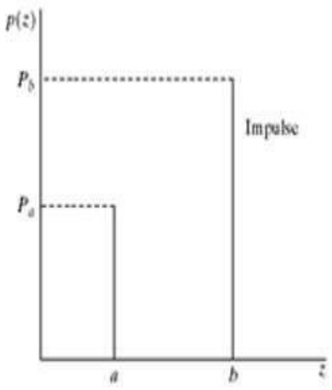


Fig 3.7 Impulse Noise

- z is Gray level
- If b > a then b is a light dot and a is a dark dot
- If either P_a or P_b =0 is Unipolar otherwise Bipolar Impulse Noise
- If Neither probability is 0 and approximately equal then noise values will resemble salt & pepper granules randomly distributed over the image.
- Also referred as Shot and Spike Noise

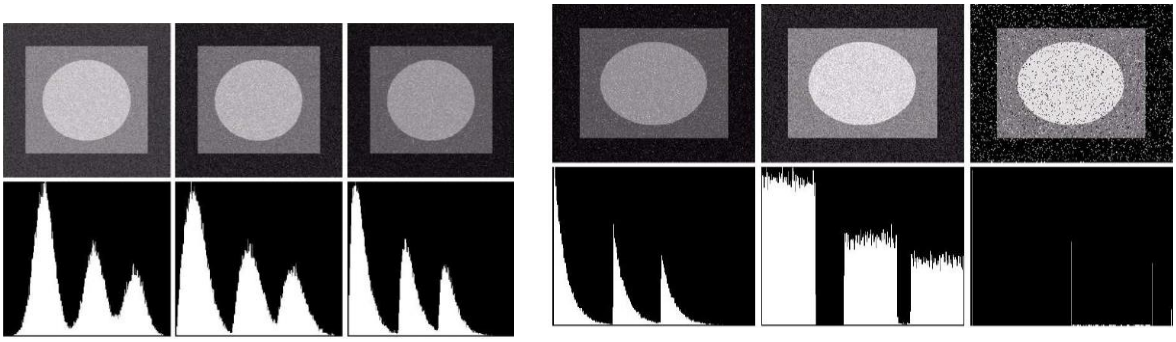


Fig. 3.8 Images and histograms resulting from adding Gaussian, Rayleigh, Gamma, exponential, uniform, and impulse noises

3.3 Inverse Filtering

- Simplest approach to restore an image
- we compute an estimate, of the transform of the original image simply by dividing the transform of the degraded image, , by the degradation function
- Compute an estimate F'(u, v) of the transform of the original image by:

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

Divisions are made between individual elements of the functions

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

- Even if we know degradation function, we can not recover the undegraded image [Inverse Fourier Transform of F(u, v)] exactly because
- N(u, v) is random function whose Fourier Transform is not known
- If degradation has ZERO or less value then N(u, v) / H(u, v) dominates the estimated F'(u, v)
- No explicit provision for handling Noise

3.4 Minimum Mean Square Error Filtering

It is an approach that incorporates both the degradation function and statistical characteristics noise into the restoration process. The method is founded on considering images and noise as random variables. The objective is to find an estimate of the uncorrupted image such that the mean square error between them is minimized.

- This error measure is given by

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

It is assumed that the noise and the image are uncorrelated; that one or the other has zero mean. The intensity 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

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

$H(u,v)$ = degradation function
$H^*(u,v)$ = complex conjugate of $H(u,v)$
$|H(u,v)|^2 = H^*(u,v)H(u,v)$
$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise
$S_f(u,v) = |F(u,v)|^2$ = power spectrum of the undegraded image

- This result is known as the Wiener filter, after N. Wiener [1942], who first proposed the concept
- 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 the entire denominator is zero for the same value(s) of u and v
- H(u,v) is the transform of the degradation function
- 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
- Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter

- A number of useful measures are based on the power spectra of noise and of the undegraded image
- One of the most important is the signal-to-noise ratio, approximated using frequency domain quantities such as

$$\text{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y)-\hat{f}(x,y)]^2} \qquad \text{SNR} = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|^2}$$

- This ratio gives a measure of the level of information bearing signal power (i.e., of the original, undegraded image) to the level of noise power
- Images with low noise tend to have a high SNR and, conversely, the same image with a higher level of noise has a lower SNR
- This ratio by itself is of limited value, but it is an important metric used in characterizing the performance of restoration algorithms
- The mean square error given in statistical form can be approximated also in terms a summation involving the original and restored images:

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

3.5 Constrained Least Square Filtering

$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$ in vector-matrix form

For example, suppose that g is of size MxN

- Then we can form the first N elements of the vector g by using the image elements in first row of g(x,y) , the next N elements from the second row, and so on
- The resulting vector will have dimensions MN x 1

- These are also the dimensions of f and η, as these vectors are formed in the same manner
- The matrix H then has dimensions MNxMN
- Its elements are given by the elements of the convolution

It would be reasonable to arrive at the conclusion that the restoration problem can now be reduced to simple matrix manipulations. Unfortunately, this is not the case. However, formulating the restoration problem in matrix form does facilitate derivation of restoration techniques. this method has its roots in a matrix formulation. Central to the method is the issue of the sensitivity of H to noise. One way to alleviate the noise sensitivity problem is to base optimality of restoration on a measure of smoothness, such as the second derivative of an image. To be meaningful, the restoration must be constrained by the parameters of the problems at hand. Thus, what is desired is to find the minimum of a criterion function, C, defined as

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

subject to the constraint

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\boldsymbol{\eta}\|^2$$

The frequency domain solution to this optimization problem is given by the expression

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

where ϒ is a parameter that must be adjusted so that the constraint is satisfied, and P(u, v) is the Fourier transform of the function. This reduces to inverse filtering if ϒ is zero.

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

By comparing the constrained least squares and Wiener results, it is noted that the former yielded slightly better results for the high- and medium-noise cases, with both filters generating essentially equal results for the low-noise case

It is not unexpected that the constrained least squares filter would outperform the Wiener filter when selecting the parameters manually for better visual results. A procedure for computing ϒ by iteration is as follows. Define a "residual" vector r as

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

$$\begin{aligned}\phi(\gamma) &= \mathbf{r}^T\mathbf{r} \\ &= \|\mathbf{r}\|^2\end{aligned}$$

is a monotonically increasing function of ϒ

What we want to do is adjust ϒ so that

$$\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2 \pm a$$

where a is an accuracy factor. In view of Eq. (5.9-6), if $\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2$, the constraint in Eq. (5.9-3) will be strictly satisfied.

Because $\phi(\gamma)$ is monotonic, finding the desired value of γ is not difficult. One approach is to

1. Specify an initial value of γ.
2. Compute $\|\mathbf{r}\|^2$.
3. Stop if Eq. (5.9-8) is satisfied; otherwise return to step 2 after increasing γ if $\|\mathbf{r}\|^2 < \|\boldsymbol{\eta}\|^2 - a$ or decreasing γ if $\|\mathbf{r}\|^2 > \|\boldsymbol{\eta}\|^2 + a$. Use the new value of γ in Eq. (5.9-4) to recompute the optimum estimate $\hat{F}(u, v)$.

Other procedures, such as a Newton–Raphson algorithm, can be used to improve the speed of convergence. This method requires knowledge of only the mean and variance of the noise. These parameters usually can be calculated from a given degraded image, so this is an important advantage. Another difference is that the Wiener filter is based on minimizing a statistical criterion and, as such, it is optimal in an average sense. This method has the notable feature that it yields an optimal result for each image to which it is applied. Of course, it is important to keep in mind that these optimality criteria, while satisfying from a theoretical point of view, are not related to the dynamics of visual perception. As a result, the choice of one algorithm over the other will almost always be determined (at least partially) by the perceived visual quality of the resulting images.

3.6 Geometric Mean Filter

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

with α and β being positive, real constants

The geometric mean filter consists of the two expressions in brackets raised to the powers α and 1- α , respectively

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

- When α=1 this filter reduces to the inverse filter
- With α=0 the filter becomes the so-called parametric Wiener filter, which reduces to the standard Wiener filter when β =1
- If α = ½ the filter becomes a product of the two quantities raised to the same power, which is the definition of the geometric mean, thus giving the filter its name
- With β =1 as α decreases below 1/2, the filter performance will tend more toward the inverse filter
- Similarly, when α increases above 1/2, the filter will behave more like the Wiener filter
- When α =1/2 and β =1 the filter also is commonly referred to as the spectrum equalization filter
- Equation is quite useful when implementing restoration filters because it represents a family of filters combined into a single expression

3.7 Geometric Transformations

- Geometric transformations modify the spatial relationship between pixels in an image
- These transformations often are called rubber-sheet transformations because they may be viewed as analogous to "printing" an image on a sheet of rubber and then stretching the sheet according to a predefined set of rules
- In terms of digital image processing, a geometric transformation consists of two basic operations:
    - a spatial transformation of coordinates and
    - intensity interpolation that assigns intensity values to the spatially transformed pixels
- The transformation of coordinates may be expressed as

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

- where (v, w) are pixel coordinates in the original image and (x, y) are the corresponding pixel coordinates in the transformed image
- One of the most commonly used spatial coordinate transformations is the affine transform, which has the general form

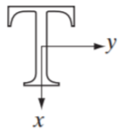
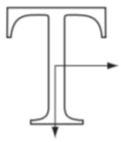
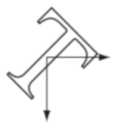
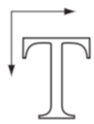
$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

This transformation can scale, rotate, translate, or shear a set of coordinate points, depending on the value chosen for the elements of matrix T.

The preceding transformations relocate pixels on an image to new locations. To complete the process, we have to assign intensity values to those locations. This task is accomplished using intensity interpolation. For an example of zooming an image and the issue of intensity assignment to new pixel locations:

- Zooming is simply scaling
- the problem of assigning intensity values to the relocated pixels resulting from the other transformations
- we consider nearest neighbor, bilinear, and bicubic interpolation techniques when working with these transformations

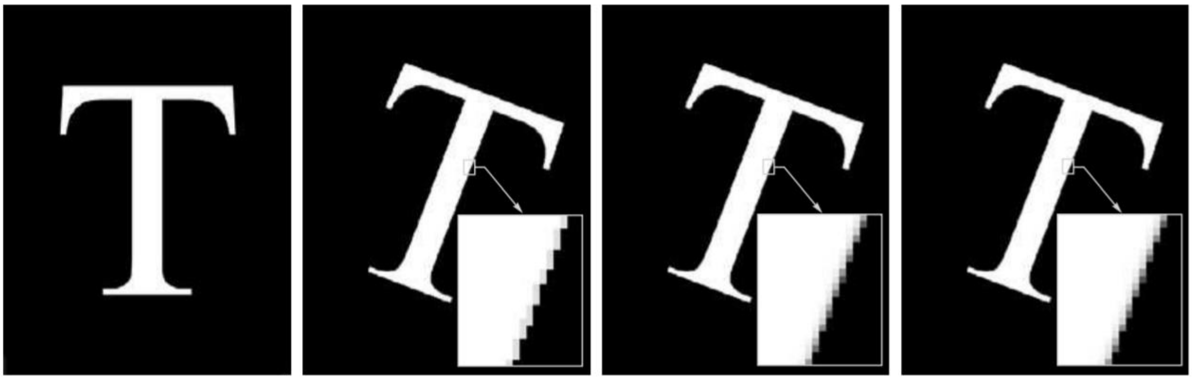
Table 3.1 Affine Transformations

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

In practice, we can use intensity interpolation in two basic ways.

- The first, called a forward mapping, consists of scanning the pixels of the input image and, at each location, , computing the spatial location, (x, y), of the corresponding pixel in the output image directly.

- A problem with the forward mapping approach is that two or more pixels in the input image can be transformed to the same location in the output image, raising the question of how to combine multiple output values into a single output pixel
- In addition, it is possible that some output locations may not be assigned a pixel at all
- The second approach, called inverse mapping, scans the output pixel locations and, at each location, (x, y), computes the corresponding location in the input image using (v, w)=T^{-1} (x, y)
- It then interpolates among the nearest input pixels to determine the intensity of the output pixel value
- Inverse mappings are more efficient to implement than forward mappings and are used in numerous commercial implementations of spatial transformations



a b c d

FIGURE 3.9 (a) A 300 dpi image of the letter T. (b) Image rotated 21° using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

TEXT / REFERENCE BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Education, Inc., 2004.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI Learning Private Limited, New Delhi, 2002.
3. William K. Pratt, “Digital Image Processing”, 3rd Edition, John Wiley & Sons, Inc., 2001.
4. Rafael C.Gonzalez, Richard E.Woods and Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, Inc., 2004.
5. Bernd Jähne, “Digital Image Processing”, 5th Revised and Extended Edition, Springer, 2002.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIT – IV – Digital Image Processing – SEC1606

IV. Morphological Image Processing & Segmentation

Morphological Image Processing - Logic Operations involving Binary Images; Dilation and Erosion; Opening and Closing; Basic Morphological Algorithms - Boundary Extraction, Region Filling, Thickening, Thinning; Image Segmentation - Detection of Discontinuities; Edge Linking; Boundary Detection; Thresholding - Global and Adaptive; Region based Segmentation.

4.1 Logic Operations involving Binary Images

Mathematical Morphology is based on the algebra of non-linear operators operating on object shape and in many respects supersedes the linear algebraic system of convolution. It performs in many tasks – pre-processing, segmentation using object shape, and object quantification – better and more quickly than the standard approach. Mathematical morphology tool is different from the usual standard algebra and calculus. Morphology tools are implemented in most advanced image analysis.

Mathematical morphology is very often used in applications where shape of objects and speed is an issue—example: analysis of microscopic images, industrial inspection, optical character recognition, and document analysis. The non-morphological approach to image processing is close to calculus, being based on the point spread function concept and linear transformations such as convolution. Mathematical morphology uses tools of non-linear algebra and operates with point sets, their connectivity and shape. Morphology operations simplify images, and quantify and preserve the main shape characteristics of objects. Morphological operations are used for the following purpose:

- Image pre-processing (noise filtering, shape simplification)
- Enhancing object structure (skeleton zing, thinning, thickening, convex hull, object marking)
- Segmenting objects from the background
- Quantitative description of objects (area, perimeter, projections, Euler-Poincare characteristics)

Mathematical morphology exploits point set properties, results of integral geometry, and topology. The real image can be modelled using point sets of any dimension; the Euclidean 2D space and its system of subsets is a natural domain for planar shape description.

Computer vision uses the digital counterpart of Euclidean space – sets of integer pairs (∈) for binary image morphology or sets of integer triples (∈) for gray-scale morphology or binary 3D morphology. Discrete grid can be defined if the neighbourhood relation between points is well defined. This representation is suitable for both rectangular and hexagonal grids. A morphological transformation is given by the relation of the image with another small point set B called structuring element. B is expressed with respect to a local origin. Structuring element is a small image-used as a moving window-- whose support delineates pixel neighbourhoods in the image plane. It can be of any shape, size, or connectivity (more than 1 piece, have holes). To apply the morphologic transformation () to the image means that the structuring element B is moved systematically across the entire image. Assume that B is positioned at some point in the image; the pixel in the image corresponding to the representative point O of the structuring element is called the current pixel. The result of the relation between the image X and the structuring element B in the current position is stored in the output image in the current image pixel position.

- **Reflection**

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

- **Translation**

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

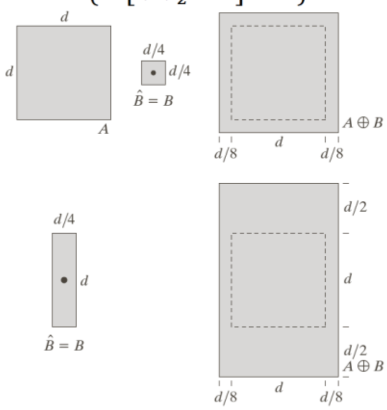
4.2 Dilation and Erosion

Dilation

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

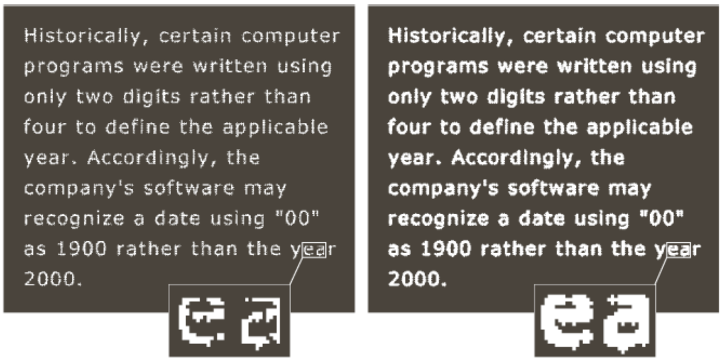
$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

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



a b c
d e

FIGURE 4.1
(a) Set A.
(b) Square structuring element (the dot denotes the origin).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference



0	1	0
1	1	1
0	1	0

a c
b

FIGURE 4.2 (a) Sample text of poor resolution with broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

Erosion

With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, defined

$$A \ominus B = \{z | (B)_Z \subseteq A\}$$

The set of all points z such that B, translated by z, is contained by A.

$$A \ominus B = \left\{ z \mid (B)_Z \cap A^c = \varnothing \right\}$$

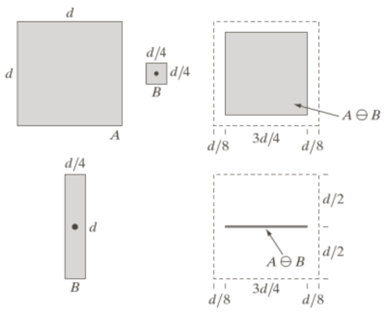
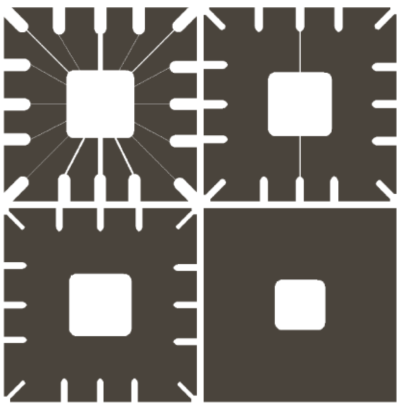


FIGURE 4.3 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.



a b
c d

FIGURE 4.4 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45, respectively. The elements of the SEs were all 1s.

4.3 Opening and Closing

Opening generally smooths the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. Closing tends to smooth sections of contours but it generates fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

The opening of set A by structuring element B,denoted $A \circ B$, is defined as

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

The closing of set A by structuring element B, denoted $A \bullet B$, is defined as

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

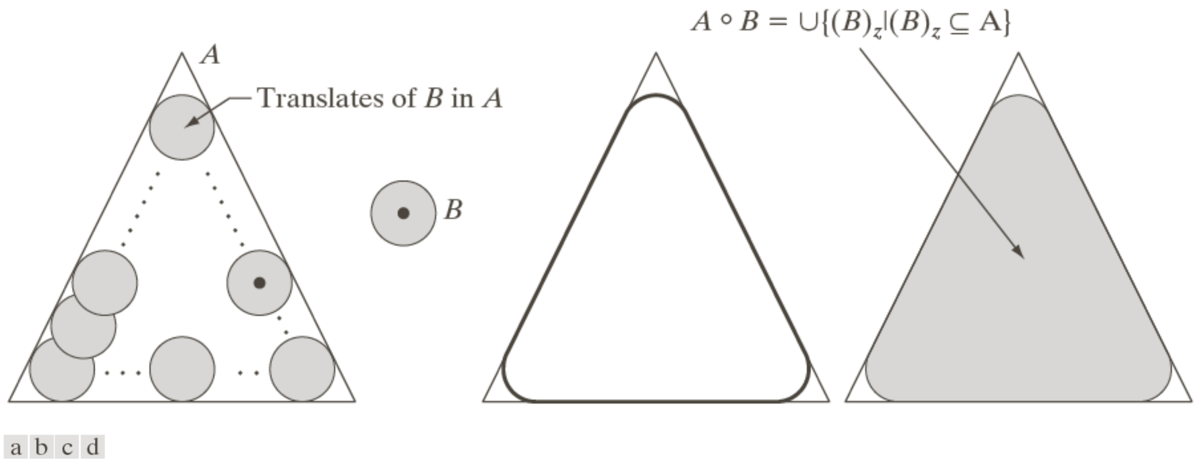


FIGURE 4.5 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

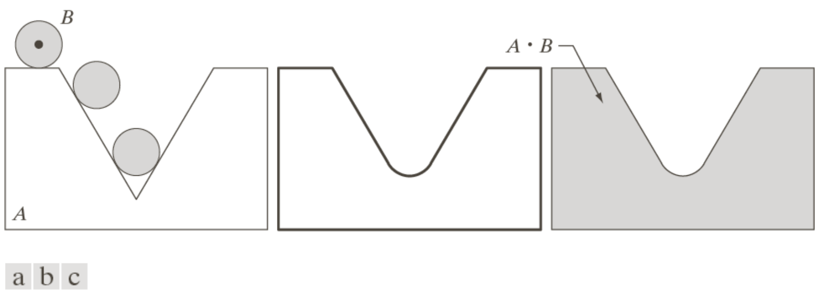
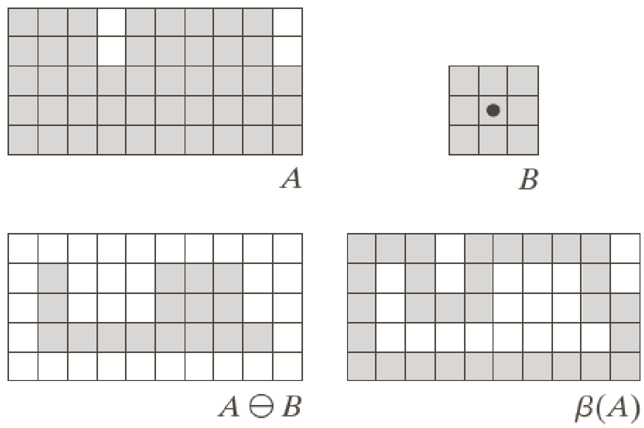


FIGURE 4.6 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

4.4 Boundary Extraction

The boundary of a set A, can be obtained by first eroding A by B and then performing the set difference between A and its erosion.

$$\beta(A) = A - \left(A - B\right)$$



a b
c d

FIGURE 4.7 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

a b

FIGURE 4.8 (a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

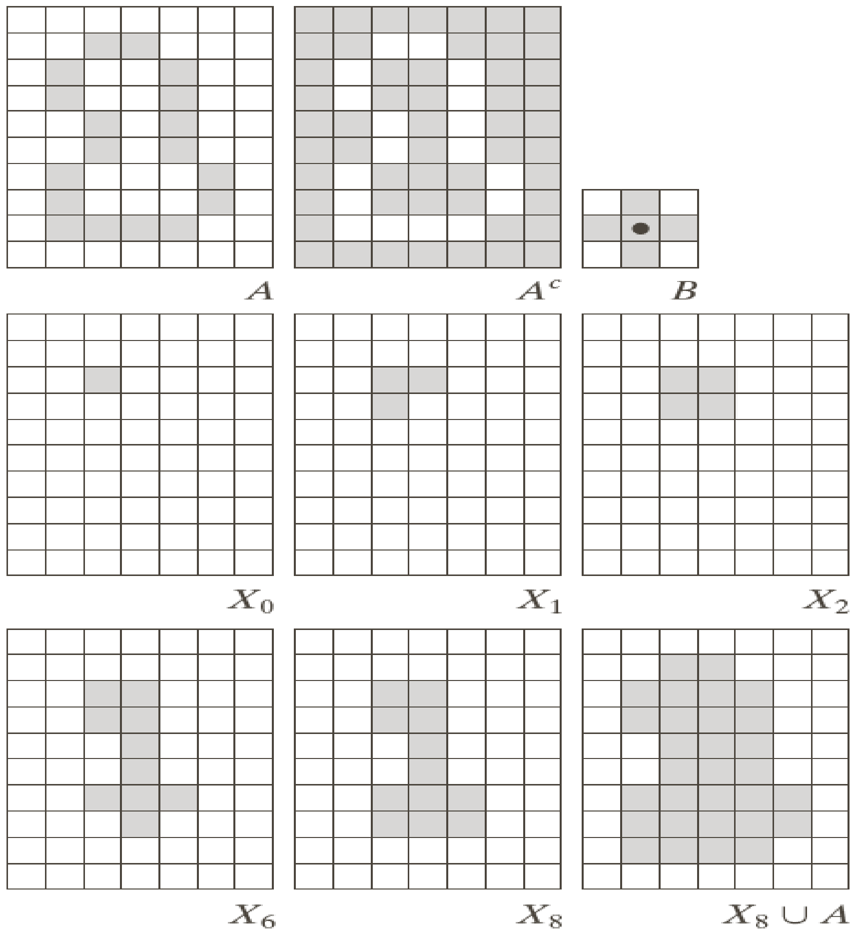
4.5 Region Filling

A hole may be defined as a background region surrounded by a connected border of foreground pixels. Let A denote a set whose elements are 8-connected boundaries, each boundary enclosing a background region (i.e., a hole). Given a point in each hole, the objective is to fill all the holes with 1s.

1. Forming an array X_0 of 0s (the same size as the array containing A), except the locations in X_0 corresponding to the given point in each hole, which we set to 1.

2. $X_k = (X_{k-1} + B) \quad A^c \quad k=1,2,3,\ldots$

Stop the iteration if $X_k = X_{k-1}$



a b c
d e f
g h i

FIGURE 4.9 Hole filling. (a) Set A (shown shaded). (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

4.6 Thickening

The thickening is defined by the expression

$A⊙B=A∪(A∗B)$

The thickening of A by a sequence of structuring element $\{B\}$
$A⊙\{B\}=((\ldots((A⊙B^1)⊙B^2)\ldots)⊙B^n)$

In practice, the usual procedure is to thin the background of the set and then complement the result.

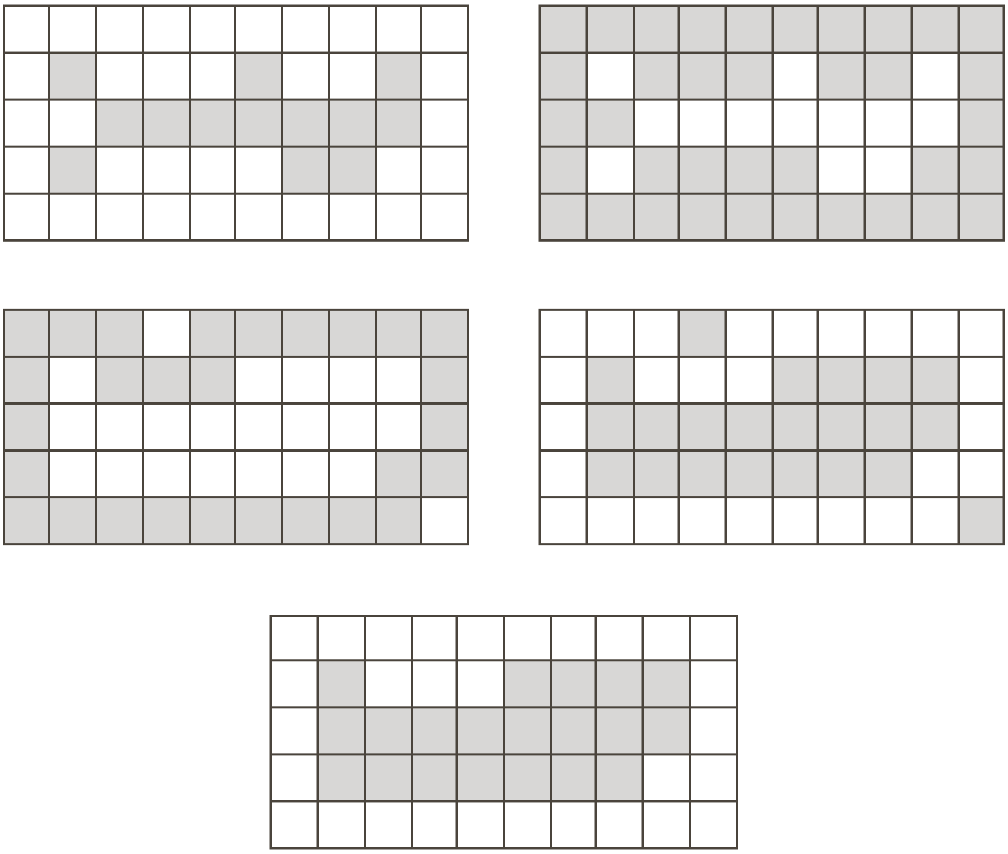


FIGURE 4.10 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

4.7 Thinning

The thinning of a set A by a structuring element B, defined

$$A \otimes B = A - (A * B)$$
$$= A \cap (A * B)^c$$

$$\{B\} = \{B^1, B^2, B^3, ..., B^n\}$$

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

The thinning of A by a sequence of structuring element $\{B\}$

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

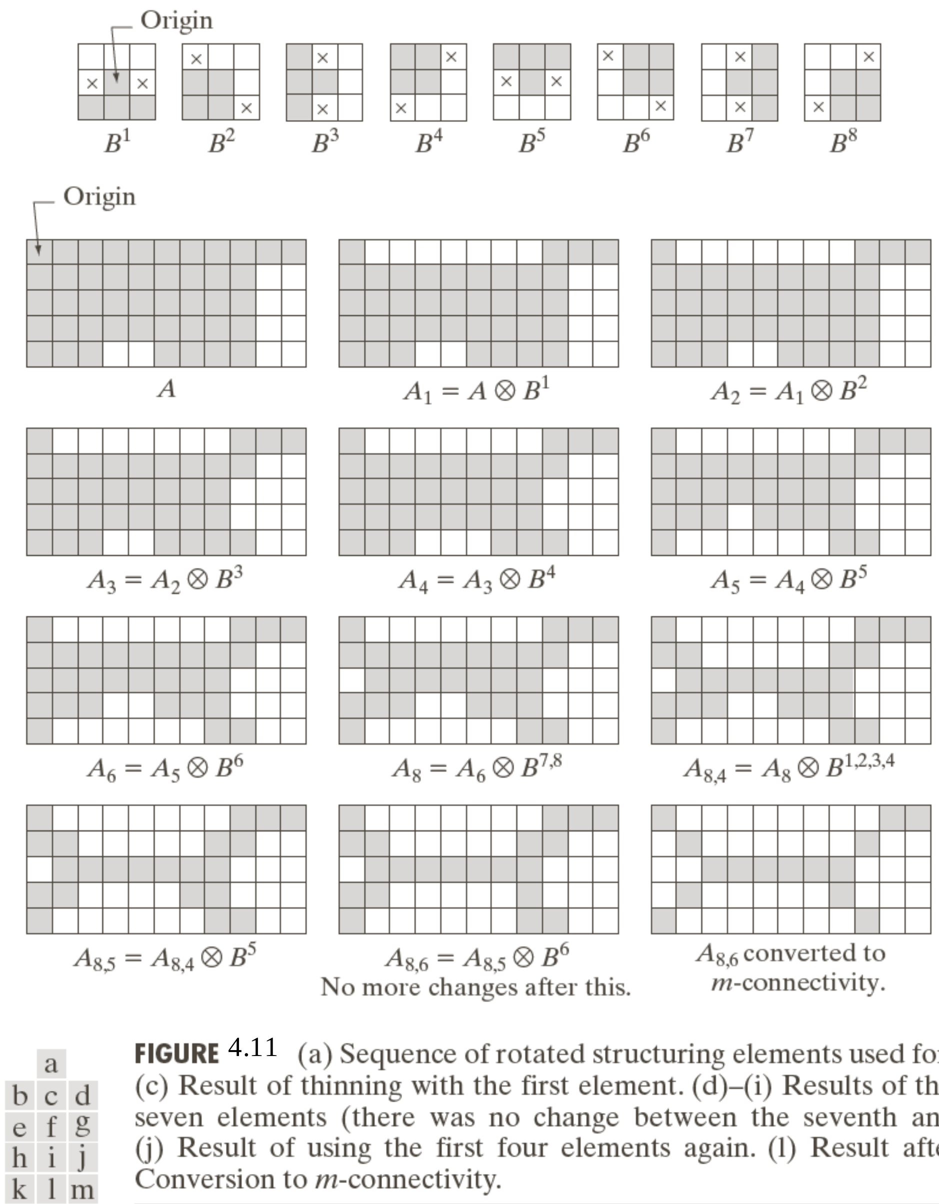


FIGURE 4.11 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to m-connectivity.

4.8 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.

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. Only slightly more common than point detection is to find one pixel wide line in an image. For digital images the only three point straight lines are only horizontal, vertical, or diagonal (+ or -45^0).

4.9 Edge Linking & Boundary Detection

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

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

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

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)

$$y_i = ax_i + b$$

- The Hough transform consists of finding all pairs of values of θ and ρ which satisfy the
  equations that pass through (x,y).
- These are accumulated in what is basically a 2-dimensional histogram.
- When plotted these pairs of θ and ρ will look like a sine wave. The process is repeated for all appropriate (x,y) locations.

4.10 Thresholding

Global – T depends only on gray level values

Local – T depends on both gray level values and local property

Dynamic or Adaptive – T depends on spatial coordinates
Different approaches possible in Graylevel threshold

- Interactive threshold

- Adaptive threshold
- Minimisation method

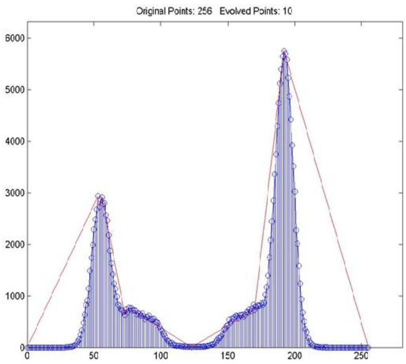


Fig. 4.11 Gray level thresholding

4.11 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.

Basic Formulation

- Let R represent the entire image region.

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

The main problem with region splitting is determining where to split a region. One method to divide a region is to use a quad tree structure. Quadtree: a tree in which nodes have exactly four descendants. 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 $P(R_j \cup R_k)$ = TRUE. (the

quadtree structure may not be preserved)

– Stop when no further merging or splitting is possible.

TEXT / REFERENCE BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Education, Inc., 2004.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI Learning Private Limited, New Delhi, 2002.
3. William K. Pratt, “Digital Image Processing”, 3rd Edition, John Wiley & Sons, Inc., 2001.
4. Rafael C.Gonzalez, Richard E.Woods and Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, Inc., 2004.
5. Bernd Jähne, “Digital Image Processing”, 5th Revised and Extended Edition, Springer, 2002.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIT – V – Digital Image Processing – SEC1606

V. Colour Image Processing & Applications

Conversion of Colour Models; Basic of Full-Colour Image Processing; Colour Transformations; Smoothing; Sharpening; Segmentation; Applications of Image Processing - Motion Analysis, Image Fusion, Image Classification

5.1 Conversion of Colour Models

Colour models:

Colour models provide a standard way to specify a particular colour, by defining a 3D coordinate system, and a subspace that contains all constructible colours within a particular model. Any colour that can be specified using a model will correspond to a single point within the subspace it defines. Each colour model is oriented towards either specific hardware (RGB, CMY, YIQ), or image processing applications (HSI).

The RGB Model

In the RGB model each colour appears in its primary spectral components of red, green and blue. The model is based on a Cartesian coordinate system RGB values are at 3 corners Cyan magenta and yellow are at three other corners Black is at the origin White is the corner furthest from the origin. Different colours are points on or inside the cube represented by RGB vectors.

Images represented in the RGB colour model consist of three component images – one for each primary colour. When fed into a monitor these images are combined to create a composite colour image. The number of bits used to represent each pixel is referred to as the colour depth. A 24-bit image is often referred to as a full-colour image as it allows16,777,216 colours. The RGB model is used for colour monitors and most video cameras.

The CMY Model

The CMY (cyan-magenta-yellow) model is a *subtractive* model appropriate to absorption of colours, for example due to pigments in paints. Whereas the RGB model asks what is added to black to get a particular colour, the CMY model asks what is subtracted from white. In this case, the primaries are cyan, magenta and yellow, with red, green and blue as secondary colours. When a surface coated with cyan pigment is illuminated by white light, no red light is reflected, and similarly for magenta and green, and yellow and blue. The CMY model is used by printing devices and filters. Equal amounts of the pigment primaries, cyan, magenta, and yellow should produce black. In practice, combining these colors for printing produces a muddy-looking black. To produce true black, the predominant color in printing, the fourth color, black, is added, giving rise to the CMYK color model.

HSI Model

The HSI model uses three measures to describe colours:

Hue: A colour attribute that describes a pure colour (pure yellow, orange or red)

Saturation: Gives a measure of how much a pure colour is diluted with white light

Intensity: Brightness is nearly impossible to measure because it is so subjective.

Intensity is the same achromatic notion that we have seen in grey level images.

Converting colors from RGB to CMY

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

Converting colors from RGB to HSI

- Hue component

$$H = \{ \begin{matrix} \theta \text{ if } B \le G \\ 360 - \theta \text{ if } B > G \end{matrix} \qquad \theta = \cos^{-1}\left[\frac{\frac{1}{2}\{(R-G)+(R-B)\}}{\{(R-G)^2+(R-B)(G-B)\}^{1/2}} \right]$$

- Saturation component

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

- Intensity component

$$I = \frac{1}{3}(R+G+B)$$

RGB values have been normalized to the range [0,1]

Angle θ is measured with respect to the red axis

Hue can be normalized to the range [0, 1] by dividing by 360°

Converting colors from HSI to RGB

- Three sectors of interest, corresponding to the 120^0 intervals in the separation of primaries

RG sector

$$(0^\circ \le H < 120^\circ)$$

$$B = I(1-S)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$

$$G = 3I - (R+B)$$

GB sector

$$(120^\circ \le H < 240^\circ)$$

$$H = H - 120^o$$

$$R = I(1-S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$

$$B = 3I - (R+G)$$

BR sector

$$(240^\circ \le H < 360^\circ)$$

$$H = H - 240^o$$

$$G = I(1 - S)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$

$$R = 3I - (G + B)$$

5.2 Basic of Full-Colour Image Processing

Colour Image Quantization

Color quantization or color image quantization is a process that reduces the number of distinct colors used in an image, usually with the intention that the new image should be as visually similar as possible to the original image. Computer algorithms to perform color quantization on bitmaps have been studied since the 1970s. Color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations, and enables efficient compression of certain types of images. The name "color quantization" is primarily used in computer graphics research literature; in applications, terms such as optimized palette generation, optimal palette generation, or decreasing color depth are used.

Histogram of Colour Image

A histogram is a graphical representation of the number of pixels in an image. In a more simple way to explain, a histogram is a bar graph, whose X-axis represents the tonal scale(black at the left and white at the right), and Y-axis represents the number of pixels in an image in a certain area of the tonal scale. For example, the graph of a luminance histogram shows the number of pixels for each brightness level(from black to white), and when there are more pixels, the peak at the certain luminance level is higher.

A color histogram of an image represents the distribution of the composition of colors in the image. It shows different types of colors appeared and the number of pixels in each type of the colors appeared. The relation between a color histogram and a luminance histogram is that a color histogram can be also expressed as “Three Color Histograms”, each of which shows the brightness distribution of each individual Red/Green/Blue color channel.

Principles of the formation of a color histogram

The formation of a color histogram is rather simple. From the definition above, we can simply count the number of pixels for each 256 scales in each of the 3 RGB channel, and plot them on 3 individual bar graphs.In general, a color histogram is based on a certain color space, such as RGB or HSV. When we compute the pixels of different colors in an image, if the color space is large, then we can first divide the color space into certain numbers of small intervals. Each of the intervals is called a bin. This process is called color quantization. Then, by counting the number of pixels in each of the bins, we get the color histogram of the image.

5.3 Colour Transformations

Color can be described by its red (R), green (G) and blue (B) coordinates (the well-known RGB system), or by some its linear transformation as XYZ, CMY, YUV, IQ, among others. The CIE adopted systems CIELAB and CIELUV, in which, to a good approximation, equal changes in the coordinates result in equal changes in perception of the color. Nevertheless, sometimes it is useful to describe the colors in an image by some type of cylindrical-like coordinate system, it means by its hue, saturation and some value representing brightness. If the RGB coordinates are in the interval from 0 to 1, each color can be represented by the point in the cube in the RGB space. Let us imagine the attitude of the cube, where the body diagonal linking ”black” vertex and ”white” vertex is vertical. Then the height of each point in the cube corresponds to the brightness of the color, the angle or azimuth corresponds to the hue and the relative distance from the vertical diagonal corresponds to the saturation of the color.

The present color models have some disadvantages in practical use. E.g. we convert an image in some image processing application into some brightness-hue-saturation model and we would like to work with individual components (coordinates) as with separate images. There is desirable regarding to the back conversion to have all combinations of the values. It means we need such model, where the range of values of saturation is identical for all hues. From this point of view, the GLHS color model [2] is probably the best from the current ones, particularly for $w_{min} = w_{mid} = w_{max} = 1/3$. The good model should satisfy some demands as:

- The brightness should be a linear combination of all three RGB components. At least, it must be continuous growing function of all of them.
- The hue differences between the basic colors (red, green and blue) should be 120° and similarly between the complement colors (yellow, purple and cyan). The hue difference between a basic color and an adjacent complement one (e.g. red and yellow) should be 60°.
- The saturation should be 1 for the colors on the surface of the RGB color cube, it means in case of one of the RGB components is 0 or 1 except black and white vertices and it is 0 in case of R=G=B.

In our opinion, the best brightness, hue and saturation system consists of the brightness as linear combination of the RGB values, the hue as actual angle in the color cube and saturation as relative distance from the body diagonal to the surface of the color cube. Such a system, called YHS, is presented in [1]. It satisfies all three demands and makes easier some color manipulations.

5.4 Colour Image Segmentation

Color image segmentation that is based on the color feature of image pixels assumes that homogeneous colors in the image correspond to separate clusters and hence meaningful objects in the image. In other words, each cluster defines a class of pixels that share similar color properties. As the segmentation results depend on the used color space, there is no single-color space that can provide acceptable results for all kinds of images. Clustering is the process of partitioning a set of objects (pattern vectors) into subsets of similar objects called clusters. Pixel clustering in three-dimensional color space on the basis of their color similarity is oneof popular approaches in the field of color image segmentation. Clustering is often seen as an unsupervised classification of pixels. Generally, the a priori knowledge about the image is not used during a clustering process. Colors, dominated in the image, create dense clusters in the color space in natural way. Region-based techniques group pixels into homogeneous regions. In this family of techniques, we can find following techniques: region growing, region splitting, region merging and others. Particularly the region growing technique, proposed for grayscale images so long ago, is constantly popular in color image processing. The region growing is a typical bottom-up technique. Neighboring pixels are merged into regions, if their attributes, for example colors, are sufficiently similar. This similarity is often represented by a homogeneity criterion. If a pixel satisfied the homogeneity criterion, then the pixel can be included to the region and then the region attributes (a mean color, an area of region etc.) are updated. The region growing process, in its classical version, is starting from chosen pixels called seeds and is continued so long as all pixels will be assigned to regions. Each of these techniques varies in homogeneity criteria and methods of seeds location. The advantages of region growing techniques result from taking into consideration two important elements: the color similarity and the pixel proximity in the image.

5.5 Applications of Image Processing

Motion analysis

Motion analysis is used in computer vision, image processing, high-speed photography and machine vision that studies methods and applications in which two or more consecutive images from an image sequences, e.g., produced by a video camera or high-speed camera, are processed to produce information based on the apparent motion in the images. In some applications, the camera is fixed relative to the scene and objects are moving around in the scene, in some applications the scene is more or less fixed and the camera is moving, and in some cases both the camera and the scene are moving.

The motion analysis processing can in the simplest case be to detect motion, i.e., find the points in the image where something is moving. More complex types of processing can be to track a specific object in the image over time, to group points that belong to the same rigid object that is moving in the scene, or to determine the magnitude and direction of the motion of every point in the image. The information that is produced is often related to a specific image in the sequence, corresponding to a specific time-point, but then depends also on the neighboring images. This means that motion analysis can produce time-dependent information about motion.

Applications of motion analysis can be found in rather diverse areas, such as surveillance, medicine, film industry, automotive crash safety, ballistic firearm studies, biological science, flame propagation, and navigation of autonomous vehicles to name a few examples.

Image fusion

The term fusion means in general an approach to extraction of information acquired in several domains. The goal of image fusion (IF) is to integrate complementary multisensor, multitemporal and/or multiview information into one new image containing information the quality of which cannot be achieved otherwise. The term quality, its meaning and measurement depend on the particular application. Image fusion has been used in many application areas. In remote sensing and in astronomy, multisensor fusion is used to achieve high spatial and spectral resolutions by combining images from two sensors, one of which has high spatial resolution and the other one high spectral resolution. Numerous fusion applications have appeared in medical imaging like simultaneous evaluation of CT, MRI, and/or PET images. Plenty of applications which use multisensor fusion of visible and infrared images have appeared in military, security, and surveillance areas. In the case of multiview fusion, a set of images of the same scene taken by the same sensor but from different viewpoints is fused to obtain an image with higher resolution than the sensor normally provides or to recover the 3D representation of the scene. The multitemporal approach recognizes two different aims. Images of the same scene are acquired at different times either to find and evaluate changes in the scene or to obtain a less degraded image of the scene. The former aim is common in medical imaging, especially in change detection of organs and tumors, and in remote sensing for monitoring land or forest exploitation. The acquisition period is usually months or years. The latter aim requires the different measurements to be much closer to each other, typically in the scale of seconds, and possibly under different conditions. The list of applications mentioned above illustrates the diversity of problems we face when fusing images. It is impossible to design a universal method applicable to all image fusion tasks. Every method should take into account not only the fusion purpose and the characteristics of individual sensors, but also particular imaging conditions, imaging geometry, noise corruption, required accuracy and application-dependent data properties. We categorize the IF methods according to the data entering the fusion and according to the fusion purpose. We distinguish the following categories.

• Multiview fusion of images from the same modality and taken at the same time but from different viewpoints.

• Multimodal fusion of images coming from different sensors (visible and infrared, CT and NMR, or panchromatic and multispectral satellite images).

• Multitemporal fusion of images taken at different times in order to detect changes between them or to synthesize realistic images of objects which were not photographed in a desired time.

• Multifocus fusion of images of a 3D scene taken repeatedly with various focal length.

• Fusion for image restoration. Fusion two or more images of the same scene and modality, each of them blurred and noisy, may lead to a deblurred and denoised image.

Multichannel deconvolution is a typical representative of this category. This approach can be extended to superresolution fusion, where input blurred images of low spatial resolution are fused to provide us a high-resolution image. In each category, the fusion consists of two basic stages: image registration, which brings the input images to spatial alignment, and combining the image functions (intensities, colors, etc) in the area of frame overlap. Image registration works usually in four steps. • Feature detection. Salient and distinctive objects (corners, line intersections, edges, contours, closed boundary regions, etc.) are manually or, preferably, automatically detected. For further processing, these features can be represented by their point representatives (distinctive points, line endings, centers of gravity), called in the literature control points. • Feature matching. In this step, the correspondence between the features detected in the sensed image and those detected in the reference image is established. Various feature descriptors and similarity measures along with spatial relationships among the features are used for that purpose. 2 • Transform model estimation. The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated. The parameters of the mapping functions are computed by means of the established feature correspondence. • Image resampling and transformation. The sensed image is transformed by means of the mapping functions. Image values in non-integer coordinates are estimated by an appropriate interpolation technique. We present a survey of traditional and up-to-date registration and fusion methods and demonstrate their performance by practical experiments from various application areas. Special attention is paid to fusion for image restoration, because this group is extremely important for producers and users of low-resolution imaging devices such as mobile phones, camcorders, web cameras, and security and surveillance cameras.

Image Classification

Image Classification is a fundamental task that attempts to comprehend an entire image as a whole. The goal is to classify the image by assigning it to a specific label. Typically, Image Classification refers to images in which only one object appears and is analyzed. In contrast, object detection involves both classification and localization tasks, and is used to analyze more realistic cases in which multiple objects may exist in an image.

Image classification - assigning pixels in the image to categories or classes of interest

Image classification is a process of mapping numbers to symbols

$f(x): x \rightarrow D; x \in R^n, D = \{c_1, c_2, \ldots, c_L\}$ Number of bands = n; Number of classes = L

f(.) is a function assigning a pixel vector x to a single class in the set of classes D

In order to classify a set of data into different classes or categories, the relationship between the data and the classes into which they are classified must be well understood

- To achieve this by computer, the computer must be trained
- Training is key to the success of classification

Computer classification of remotely sensed images involves the process of the computer program learning the relationship between the data and the information classes Important aspects of accurate classification

- Learning techniques
- Feature sets

Supervised Learning/Learning process designed to form a mapping from one set of variables (data) to another set of variables (information classes)/A teacher is involved in the learning process.

Unsupervised learning/Learning happens without a teacher/Exploration of the data space to discover the scientifc laws underlying the data distribution.

Features are attributes of the data elements based on which the elements are assigned to various classes. E.g., in satellite remote sensing, the features are measurements made by sensors in different wavelengths of the electromagnetic spectrum – visible/ infrared / microwave/texture features …

In medical diagnosis, the features may be the temperature, blood pressure, lipid profile, blood sugar, and a variety of other data collected through pathological investigations. The features may be qualitative (high, moderate, low) or quantitative. The classification may be presence of heart disease (positive) or absence of heart disease (negative).

TEXT / REFERENCE BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Education, Inc., 2004.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI Learning Private Limited, New Delhi, 2002.
3. William K. Pratt, “Digital Image Processing”, 3rd Edition, John Wiley & Sons, Inc., 2001.
4. Rafael C.Gonzalez, Richard E.Woods and Steven L. Eddins, “Digital Image Processing using Matlab”, Pearson Education, Inc., 2004.
5. Bernd Jähne, “Digital Image Processing”, 5th Revised and Extended Edition, Springer, 2002.