UNIT I

DIGITAL IMAGE FUNDAMENTALS

Elements of digital image processing systems, Vidicon and Digital Camera working principles, Elements of visual perception, brightness, contrast, hue, saturation, mach band effect, Color image fundamentals - RGB, HSI models, Image sampling, Quantization, dither, Two-dimensional mathematical preliminaries, 2D transforms - DFT, DCT, KLT, SVD.

1.1 INTRODUCTION OF DIGITAL IMAGE PROCESSING SYSTEMS

Background

Interest comes from two primary backgrounds

i. Improvement of pictorial information for human perception

How can an image/video are made more aesthetically pleasing

How can an image/video are enhanced to facilitate extraction of useful information

ii. Processing of data for autonomous machine perception

One of the earliest applications was improving digitized newspaper pictures sent by transatlantic cable (early 1920s) Bartlane cable picture transmission reduced the time to send transatlantic images from more than a week to less than 3 hours

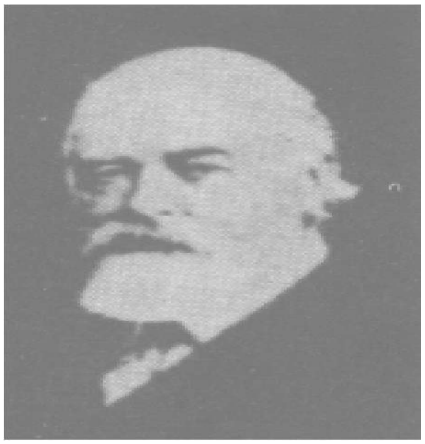
Bartlane transmitted image

Specialized printing equipment coded pictures for transmission, Received and printed on a telegraph printer fitted with typefaces to simulate a halftone pattern

Initial problems are Poor visual quality related to printing process and the distribution of brightness levels. Image produced in 1921 from a coded tape by a telegraph printer with special type faces

Improved Bartlane transmitted image

Replace the printing process. New technique used photographic reproduction made from a perforated tape at the telegraph receiving terminal. The Improvements are tonal quality and resolution



Digital picture made in 1922 from a tape punched after the signals had crossed the Atlantic twice.

15 level Bartlane image

Early images were transmitted using 5 distinct brightness levels. The process was improved in 1929 to 15 levels. A system for developing a film plate (as opposed to printing) from the coded picture tape improved the reproduction process considerably. Cable picture of Generals Pershing and Foch, transmitted in 1929 by 15-tone equipment from London to New York.

Growth in image processing

Made possible by the advent of large-scale digital computers. Often motivated by requirements of the space program

Pre-Apollo and Apollo moon missions: typical requirement was to correct various types of image distortion inherent in on-board television cameras

Mariner Mars flyby missions, etc.

Image processing now used to solve many problems

Commonly require methods capable of enhancing pictorial information for human interpretation and analysis

Example image processing applications

- ✓ Medical field: X-ray (or other biomedical) image enhancement
- ✓ Aerial and satellite image enhancement: agriculture, weather and military.
- ✓ Industrial applications: computer-based product inspection
- ✓ Law enforcement: fingerprint processing, surveillance camera processing
- ✓ Defense applications: recognizing an enemy tank in foliage, guiding a missile in flight.
- ✓ Science: enhancing an electron microscope image for readability

Digital image

Monochrome image (or simply image) refers to a 2-dimensional light intensity function f(x,y) with x and y denote spatial coordinates and the value of f(x,y) at (x,y) is proportional to the brightness (or gray level) of the image at that point.

A digital image is an image f(x,y) that has been discretized both in spatial coordinates and in brightness. The elements of such an array are referred to as image elements picture elements (pixels or pels).

1.2. ELEMENTS OF DIGITAL IMAGE PROCESSING SYSTEMS

Fig shows the components of digital image processing systems.

The first is the physical device which is use to sense the object. The second device is the digitizer which is used to convert the output of the physical sensing device into digital form.

Specialized image processing hardware usually consists of digitizer plus ALU that performs some primitive operations like arithmetic and logical.

The computer in an image processing system is a general purpose computer and can range from PC to a supercomputer.

Software for image processing system consists of specialized modules that can perform specific tasks.

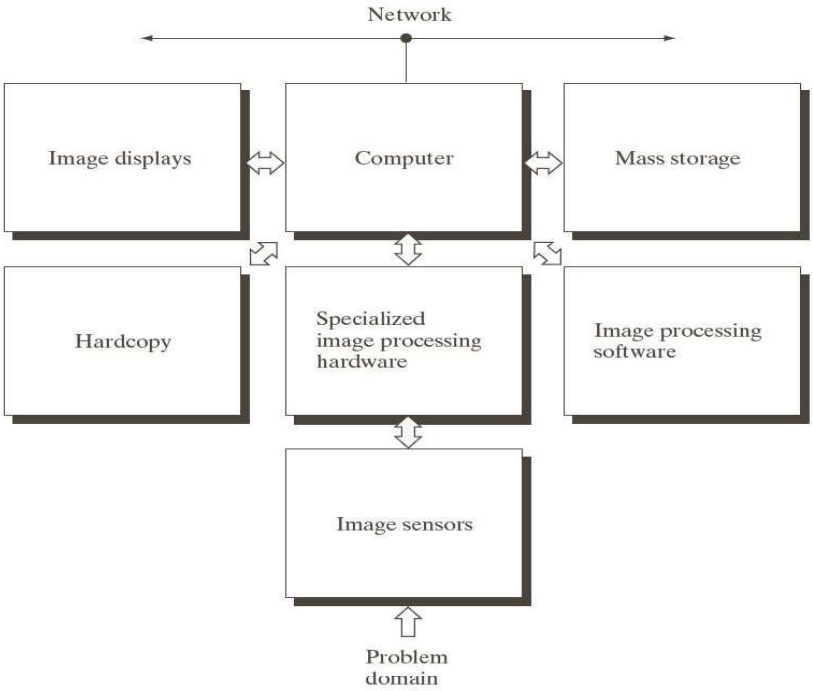


Fig.1.1 components of digital image processing systems

Mass storage capability is needed if the image is not compressed. There are three principal categories. (1)short term storage for use during processing, example: computer memory, frame buffers. (2) on line storage for relatively fast recall, example; magnetic disk or optical media(3)archival storage characterized by in frequent access, example: magnetic tapes and optical disks.

Image displays are color TV monitors. Hard copy devices for recording images include laser printers, film cameras, heat sensitive devices, inkjet printers and digital units such as optical and CD ROM disks.

Networking is useful for transmitting images. It includes optical fiber and other broad band technologies.

1.3 ELEMENTS OF VISUAL PERCEPTION

The following figure shows the anatomy of the human eye in cross section

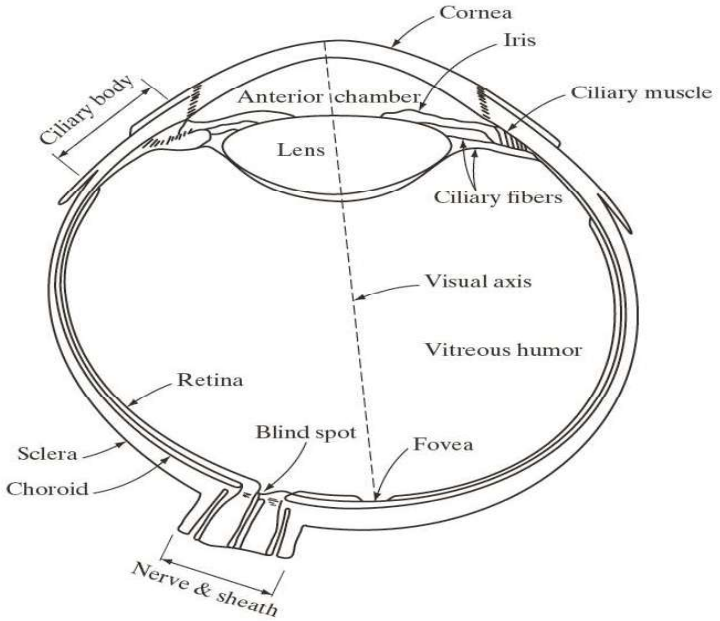


Fig.1.2 Structure of Human eye

Eye characteristics

- nearly spherical
- approximately 20 mm in diameter
- three membranes
  - Cornea (transparent)&sclera(opaque) outer cover
  - Choroid contains a network of blood vessels, heavily pigmented to reduce amount of extraneous light entering the eye. Also contains the iris diaphragm (2-8 mm to allow variable amount of light into the eye)
  - Retina is the inner most membrane, objects are imaged on the surface
- There are two types of receptors in the retina
  - The rods are long slender receptors

  - The cones are generally shorter and thicker in structure
- The rods and cones are not distributed evenly around the retina.
- Rods and cones operate differently
  - Rods are more sensitive to light than cones.
  - At low levels of illumination the rods provide a visual response called scotopic vision
  - Cones respond to higher levels of illumination; their response is called photopic vision

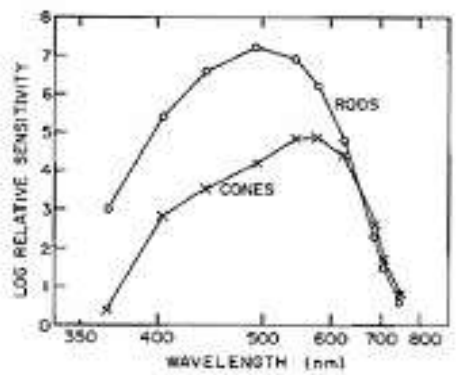


Fig1.3 Sensitivity of rodes and cones

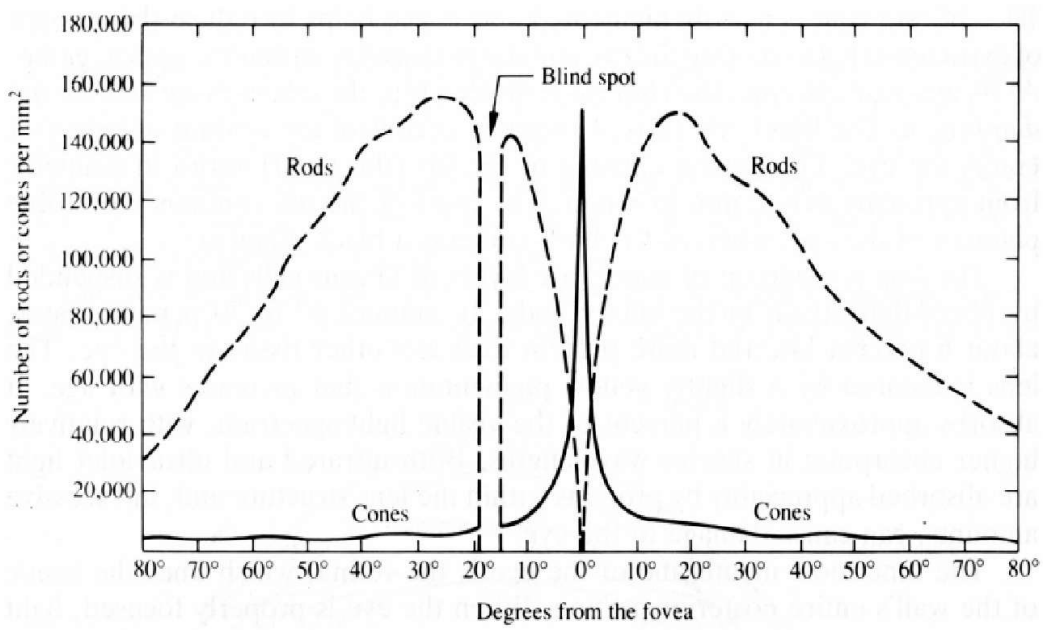


Fig.1.4 Distribution of rodes and cones in retina

- There are three basic types of cones in the retina
- These cones have different absorption characteristics as a function of wavelength with peak absorptions in the red, green, and blue regions of the optical spectrum. Most of the cones are at the fovea. Rods are spread just about everywhere except the fovea
- There is a relatively low sensitivity to blue light. There is a lot of overlap
- Distribution is radially symmetric about the fovea
- Absence of receptors in the blind spot (emergence of the optic nerve)
- Receptor density measured in degrees from the fovea (the angle formed between the visual axis and a line extending from the center of the lens to the retina)
- Fovea is a circular indentation in the retina (approximately 1.5 mm in diameter) – density of cones in that area is approximately 150,000/mm^2

1.3.1 IMAGE FORMATION IN THE EYE

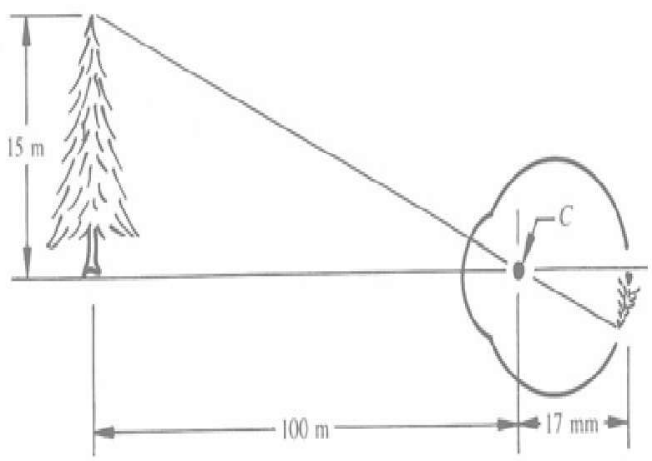


Fig.1.5 image formation in the eye

- Variable thickness lens is thick for close focus, thin for distant focus
- Distance of focal center of the lens to the retina (14-17 mm)
- Image of a 15m tree at 100m
  15/100 = X/17 or approximately 2.55 mm
- Image is almost entirely on the fovea

1.3.2 Brightness adaptation and discrimination

The range of light intensity levels to which the human visual system can adapt is enormous from scotopic threshold to the glare limit. Subjective brightness is a logarithmic function of the light intensity incident on the eye. The visual system cannot operate over long range simultaneously; rather it accomplishes this large variation by changes in its overall

sensitivity. This phenomenon is known as brightness adaptation. For any given set of conditions the current sensitivity level of the visual system is called the brightness adaptation level. Fig shows the plot of light intensity versus subjective brightness.

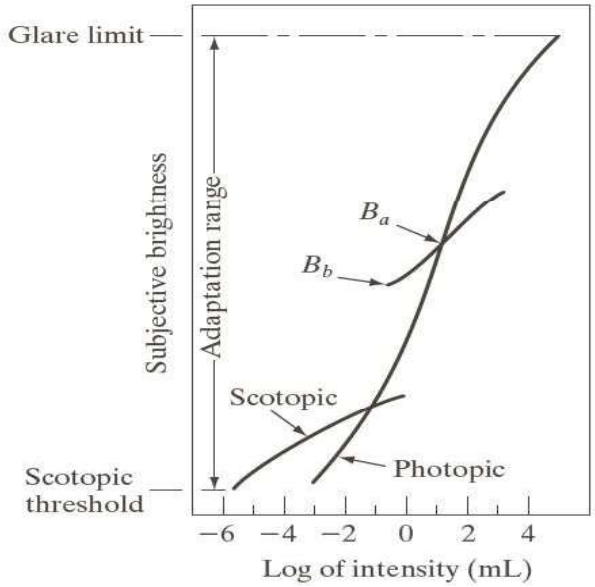


Fig1.6 Range of subjective brightness sensation showing a particular adaptation level

1.3.3 contrast sensitivity

- The response of the eye to changes in the intensity of illumination is nonlinear
- Consider a patch of light of intensity i+dI surrounded by a background intensity I as shown in the following figure
- Over a wide range of intensities, it is found that the ratio dI/I, called the Weber fraction, is nearly constant at a value of about 0.02.

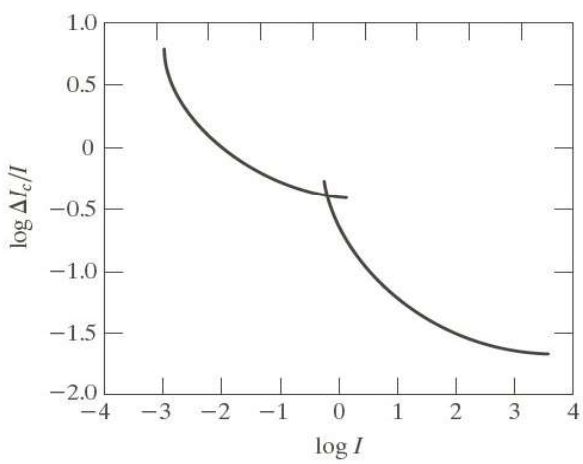


Fig1.7 Weber ratio as a function of intensity

- This does not hold at very low or very high intensities
- Furthermore, contrast sensitivity is dependent on the intensity of the surround.

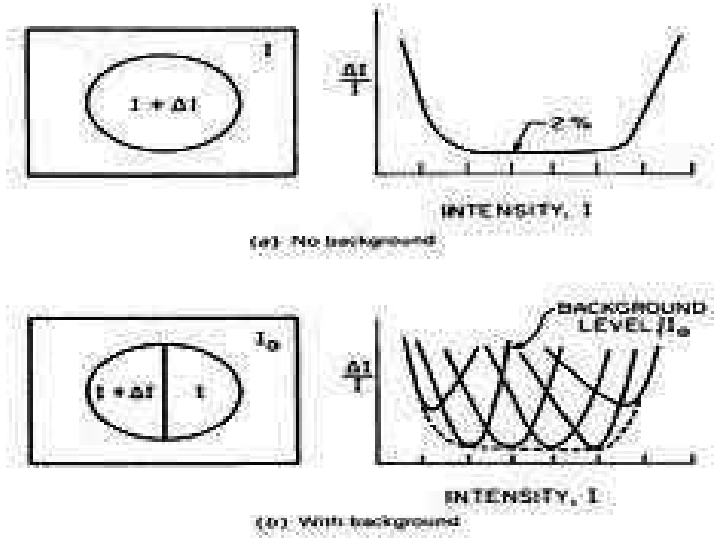


Fig.1.8 Contrast sensitivity measurement

1.3.4 Logarithmic response of cones and rods

- The response of the cones and rods to light is nonlinear. In fact many image processing systems assume that the eye's response is logarithmic instead of linear with respect to intensity.

- To test the hypothesis that the response of the cones and rods are logarithmic, we examine the following two cases:

- If the intensity response of the receptors to intensity is linear, then the derivative of the response with respect to intensity should be a constant. This is not the case as seen in the next figure.
- To show that the response to intensity is logarithmic, we take the logarithm of the intensity response and then take the derivative with respect to intensity. This derivative is nearly a constant proving that intensity response of cones and rods can be modeled as a logarithmic response.

- Another way to see this is the following, note that the differential of the logarithm of intensity is d(log(I)) = dI/I. Figure 2.3-1 shows the plot of dI/I for the intensity response of the human visual system.

- Since this plot is nearly constant in the middle frequencies, we again conclude that the intensity response of cones and rods can be modeled as a logarithmic response.

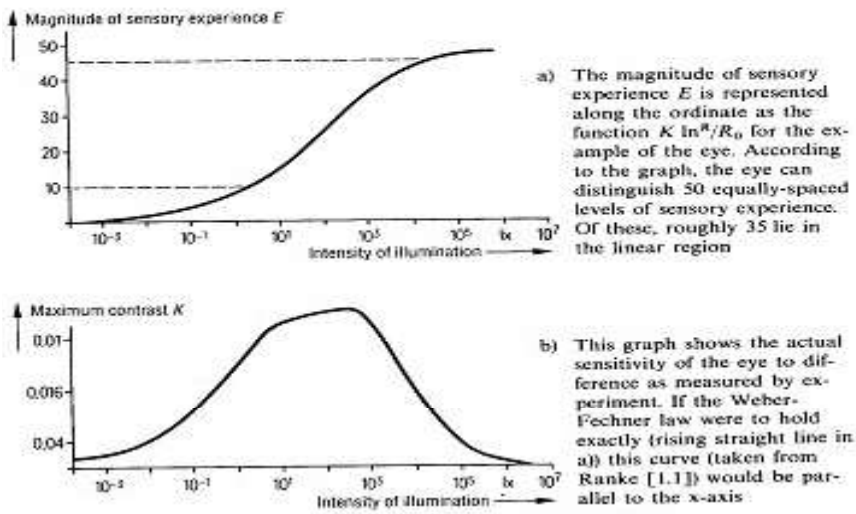


Fig1.9 Magnitude of Weber law function

1.3.5 Simultaneous contrast

- The small squares in each image are the same intensity.
- Because the different background intensities, the small squares do not appear equally bright.
- Perceiving the two squares on different backgrounds as different, even though they are in fact identical, is called the simultaneous contrast effect.
- Psychophysically, we say this effect is caused by the difference in the backgrounds.

Fig.1.10 Simultaneous contrast

1.3.6 Lateral inhibition

- Record signal from nerve fiber of receptor A.
- Illumination of receptor A alone causes a large response.
- Add illumination to three nearby receptors at B causes the response at A to decrease.
- Increasing the illumination of B further decreases A's response.
- Thus, illumination of the neighboring receptors inhibited the firing of receptor A.
- This inhibition is called lateral inhibition because it is transmitted laterally, across the retina, in a structure called the lateral plexus.
- A neural signal is assumed to be generated by a weighted contribution of many spatially adjacent rods and cones.
- Some receptors exert an inhibitory influence on the neural response.
- The weighting values are, in effect, the impulse response of the human visual system beyond the retina.

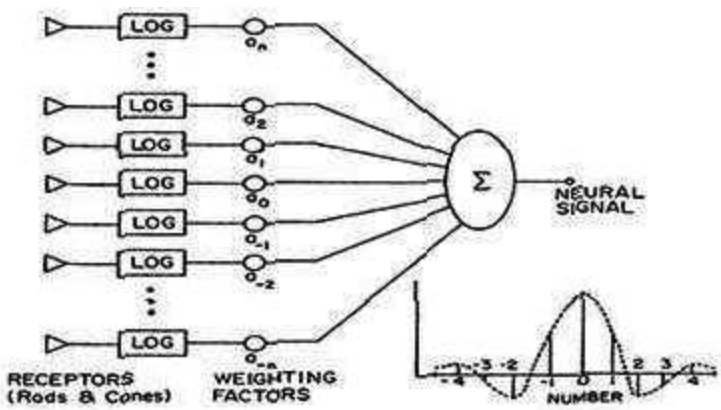


Fig.1.11 Lateral inhibition

1.3.7 Mach band effect

- Another effect that can be explained by the lateral inhibition.
- The Mach band effect is illustrated in the figure below.
- The intensity is uniform over the width of each bar.
- However, the visual appearance is that each strip is darker at its right side than its left.

The Mach band effect is illustrated in the figure below.

- A bright bar appears at position B and a dark bar appears at D.

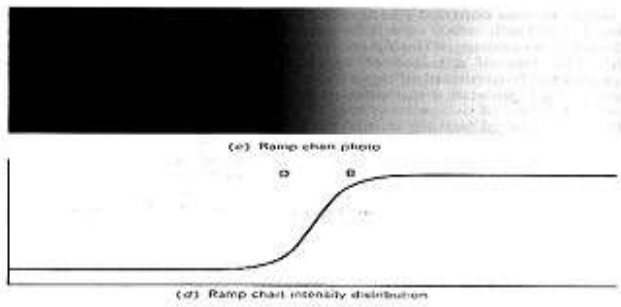


Fig.1.12 Example for match band effect

1.3.8 Modulation transfer function (mtf) experiment

- An observer is shown two sine wave grating transparencies, a reference grating of constant contrast and spatial frequency, and a variable-contrast test grating whose spatial frequency is set at some value different from that of the reference.
- Contrast is defined as the ratio

  (max-min)/(max+min)

  where max and min are the maximum and minimum of the grating intensity, respectively.
- The contrast of the test grating is varied until the brightness of the bright and dark regions of the two transparencies appear identical.
- In this manner it is possible to develop a plot of the MTF of the human visual system.
- Note that the response is nearly linear for an exponential sine wave grating.

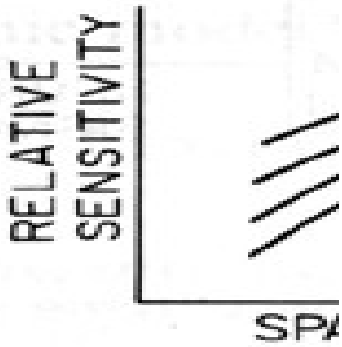


Fig.1.13 Plot for MTF

1.3.9 Monochrome vision model

- The logarithmic/linear system eye model provides a reasonable prediction of visual response over a wide range of intensities.
- However, at high spatial frequencies and at very low or very high intensities, observed responses depart from responses predicted by the model.

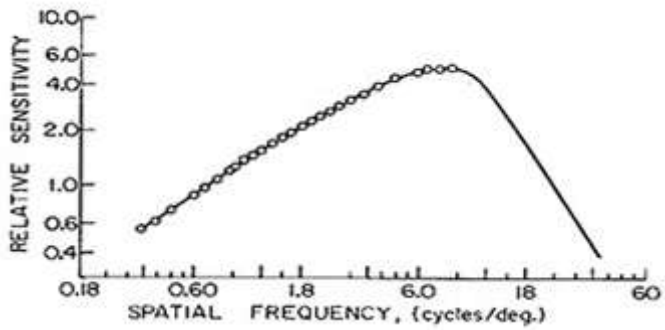
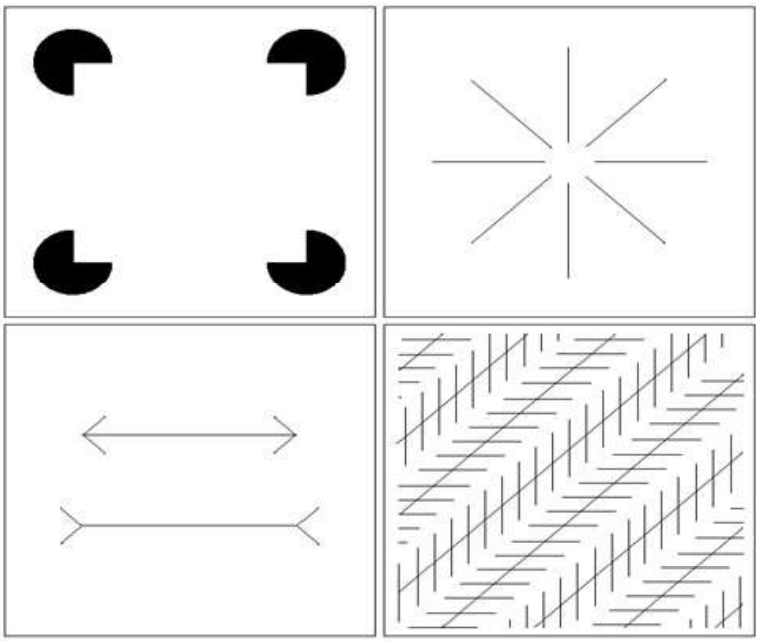


Fig.1.14 MTF measurement with exponential sine grating

1.3.10 optical illusion

This is one in which the eye fills in nonexisting information or wrongly perceives geometrical properties of object.



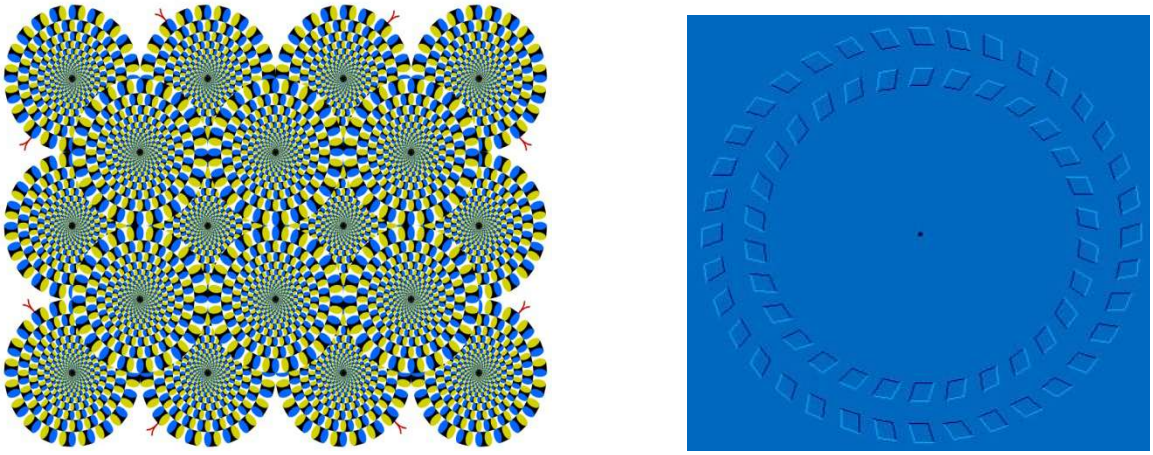


Fig.1.15 Examples for optical illusion

1.4 LIGHT

- Light exhibits some properties that make it appear to consist of particles; at other times, it behaves like a wave.
- Light is electromagnetic energy that radiates from a source of energy (or a source of light) in the form of waves
- Visible light is in the 400 nm – 700 nm range of electromagnetic spectrum

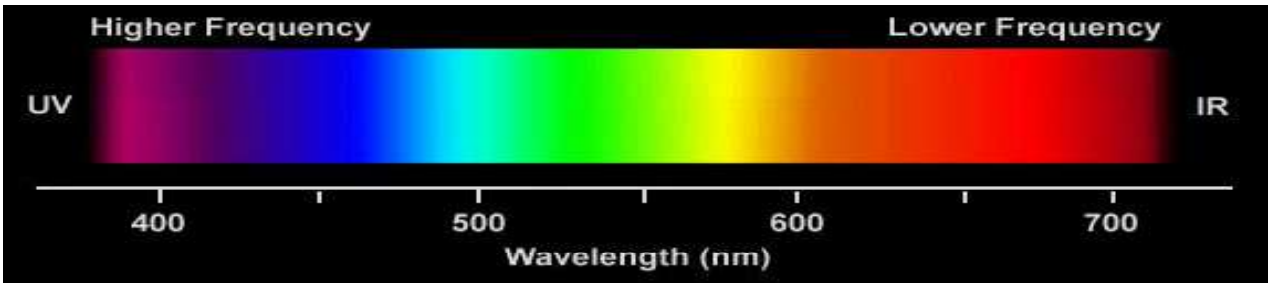


Fig.1.16 Electromagnetic spectrum

1.4.1 Intensity of light

- The strength of the radiation from a light source is measured using the unit called the candela, or candle power. The total energy from the light source, including heat and all electromagnetic radiation, is called radiance and is usually expressed in watts.
- *Luminance* is a measure of the light strength that is actually perceived by the human eye. Radiance is a measure of the total output of the source; luminance measures just the portion that is perceived.
  Brightness is a subjective, psychological measure of perceived intensity. Brightness is

practically impossible to measure objectively. It is relative. For example, a burning candle in a darkened room will appear bright to the viewer; it will not appear bright in full sunshine.

- The strength of light diminishes in inverse square proportion to its distance from its source. This effect accounts for the need for high intensity projectors for showing multimedia productions on a screen to an audience. Human light perception is sensitive but not linear.

1.5 COLOR IMAGE FUNDAMENTALS

The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

Three independent quantities are used to describe any particular color. The *hue* is determined by the dominant wavelength. Visible colors occur between about 400nm (violet) and 700nm (red) on the electromagnetic spectrum, as shown in figure

Fig.1.17 The visible spectrum.

The *saturation* is determined by the excitation purity, and depends on the amount of white light mixed with the hue. A pure hue is fully saturated, i.e. no white light mixed in. Hue and saturation together determine the *chromaticity* for a given color. Finally, the *intensity* is determined by the actual amount of light, with lighter corresponding to more intense colors.

Achromatic light has no color - its only attribute is quantity or intensity. Gray level is a measure of intensity. The *intensity* is determined by the energy, and is therefore a physical quantity. On the other hand, *brightness* or *luminance* is determined by the perception of the color, and is therefore psychological. Given equally intense blue and green, the blue is perceived as much darker than the green. Note also that our perception of intensity is nonlinear, with changes of normalized intensity from 0.1 to 0.11 and from 0.5 to 0.55 being perceived as equal changes in brightness.

Color depends primarily on the reflectance properties of an object. We see those rays that are reflected, while others are absorbed. However, we also must consider the color of the light source, and the nature of human visual system. For example, an object that reflects both red and green will appear green when there is green but no red light illuminating it, and conversely it will appear red in the absence of green light. In pure white light, it will appear yellow (= red + green).

1.5.1 Tristimulus theory of color perception

As discussed in 1.4, the human retina has 3 kinds of cones. The response of each type of cone as a function of the wavelength of the incident light is shown in figure 2. The peaks for each curve are at 440nm (blue), 545nm (green) and 580nm (red). Note that the last two actually peak in the yellow part of the spectrum.

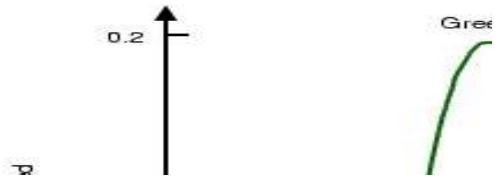


Fig.1.18 Spectral response curves for each cone type. The peaks for each curve are at 440nm (blue), 545nm (green) and 580nm (red).

1.5.2 CIE primaries

The tristimulus theory of color perception seems to imply that any color can be obtained from a mix of the three primaries, red, green and blue, but although nearly all visible colors can be matched in this way, some cannot. However, if one of the primaries is added to one of these unmatchable colors, it can be matched by a mixture of the other two, and so the color may be considered to have a negative weighting of that particular primary.

In 1931, the *Commission Internationale de l'Éclairage (CIE)* defined three standard primaries, called *X*, *Y* and *Z*, that can be added to form all visible colors. The primary *Y* was chosen so that its color matching function exactly matches the luminous-efficiency function for the human eye, given by the sum of the three curves in figure 2

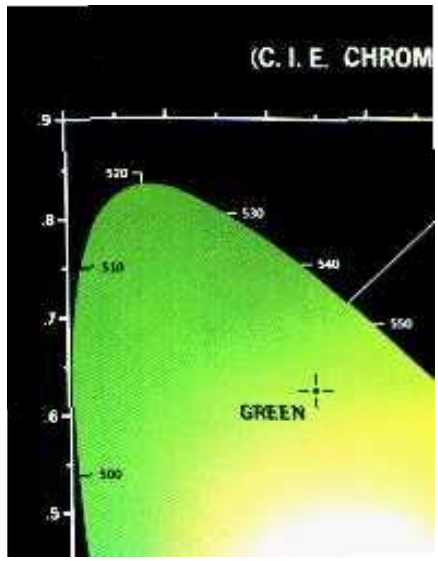


Fig.1.19 The CIE Chromaticity Diagram showing all visible colors. x and y are the normalised amounts of the X and Y primaries present, and hence z = 1 - x - y gives the amount of the Z primary required.

The CIE Chromaticity Diagram (see figure 3) shows all visible colors. The x and y axis give the normalized amounts of the X and Y primaries for a particular color, and hence z = 1 - x - y gives the amount of the Z primary required. Chromaticity depends on dominant wavelength and saturation, and is independent of luminous energy. Colors with the same chromaticity, but different luminance all map to the same point within this region.

The pure colors of the spectrum lie on the curved part of the boundary, and a standard white light has color defined to be near (but not at) the point of equal energy x = y = z = $^1/_3$. Complementary colors, i.e. colors that add to give white, lie on the endpoints of a line through this point. As illustrated in figure 4, all the colors along any line in the chromaticity diagram may be obtained by mixing the colors on the end points of the line. Furthermore, all colors within a triangle may be formed by mixing the colors at the vertices. This property illustrates graphically the fact that all visible colors cannot be obtained by a mix of R, G and B (or any other three visible) primaries alone, since the diagram is not triangular!

1.5.3 Tri-chromatic coefficients (x,y,z)

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

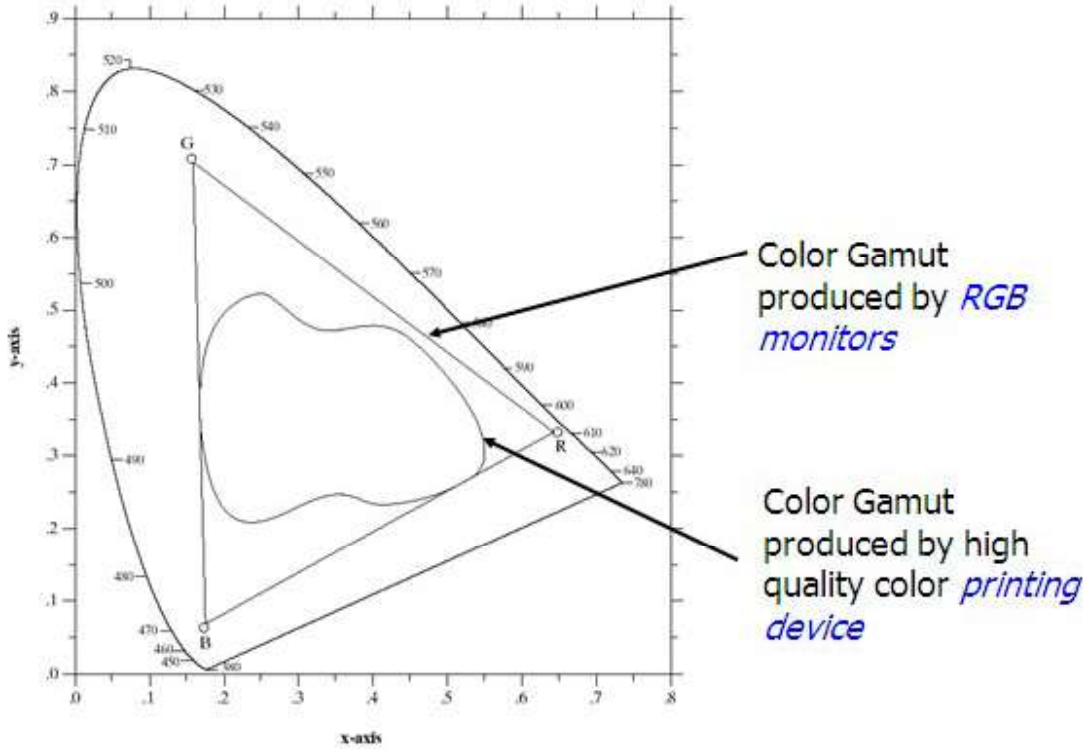


Fig.1.20 Mixing colors on the chromaticity diagram.

1.5.4 COLOR MODELS

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

- **RGB** (red, green, blue): Monitor video camera.
- **CMY** (cyan, magenta, yellow), **CMYK** (CMY, black) model for color printing.
- **HSI** model, which corresponds closely with the way humans describe and interpret color.

1.5.4.1 The RGB Model

In the RGB model, an image consists of three independent image planes, one in each of the primary colors: red, green and blue. (The standard wavelengths for the three primaries are as shown in figure 1). Specifying a particular color is by specifying the amount of each of the primary components present. Figure 5 shows the geometry of the RGB color model for specifying colors using a Cartesian coordinate system. The grayscale spectrum, i.e. those colors made from equal amounts of each primary, lies on the line joining the black and white vertices.

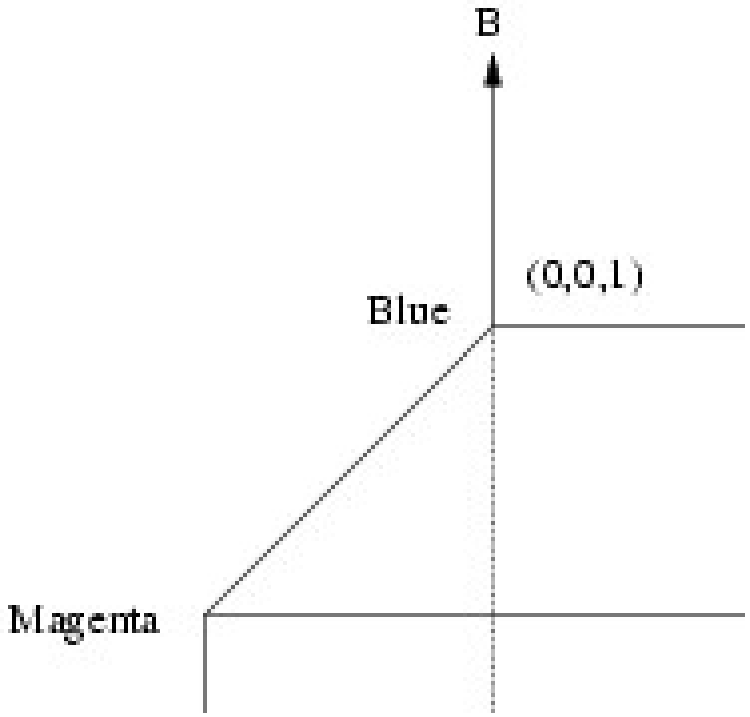


Fig.1.21 The RGB color cube. The grayscale spectrum lies on the line joining the black and white vertices.

This is an *additive* model, i.e. the colors present in the light add to form new colors, and is appropriate for the mixing of colored light for example. The image on the left of figure 7 shows the additive mixing of red, green and blue primaries to form the three secondary colors yellow (red + green), cyan (blue + green) and magenta (red + blue), and white ((red + green + blue). The RGB model is used for color monitors and most video cameras.

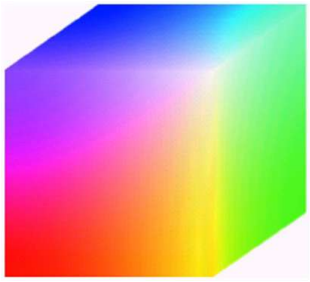


Fig.1.22 RGB 24 bit color cube

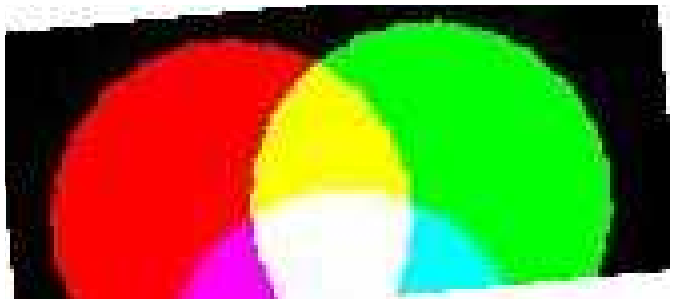


Fig.1.23 The figure on the left shows the additive mixing of red, green and blue primaries to form the three secondary colors yellow (red + green), cyan (blue + green) and magenta (red + blue), and white ((red + green + blue). The figure on the right shows the three subtractive primaries, and their pairwise combinations to form red, green and blue, and finally black by subtracting all three primaries from white.

1.5.4.2 The CMY Model

The CMY (cyan-magenta-yellow) model is a *subtractive* model appropriate to absorption of colors, for example due to pigments in paints. Whereas the RGB model asks what is added to black to get a particular color, 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 colors (see the image on the right of figure 7).

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 relationship between the RGB and CMY models is given by:

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

The CMY model is used by printing devices and filters.

Why does blue paint plus yellow paint give green?

As all schoolchildren know, the way to make green paint is to mix blue paint with yellow. But how does this work? If blue paint absorbs all but blue light, and yellow absorbs blue only, when combined no light should be reflected and black paint result.

However, what actually happens is that imperfections in the paint are exploited. In practice, blue paint reflects not only blue, but also some green. Since the yellow paint also reflects green (since yellow = green + red), some green is reflected by both pigments, and all other colors are absorbed, resulting in green paint.

1.5.4.3 The HSI Model

As mentioned above, color may be specified by the three quantities hue, saturation and intensity. This is the HSI model, and the entire space of colors that may be specified in this way is shown in figure 8.

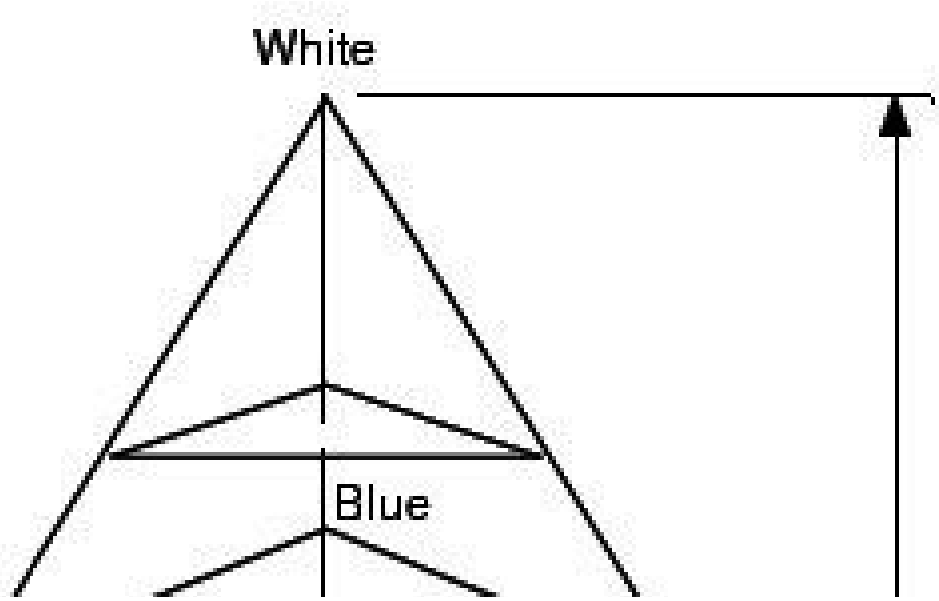


Fig.1.24 The HSI model, showing the HSI solid on the left, and the HSI triangle on the right, formed by taking a horizontal slice through the HSI solid at a particular intensity. Hue is measured from red, and saturation is given by distance from the axis. Colors on the surface of the solid are fully saturated, i.e. pure colors, and the grayscale spectrum is on the axis of the solid. For these colors, hue is undefined.

Converting colors from RGB to HSI model:

$$H = \begin{cases} \theta & \text{if B} \le \text{G} \\ 360 - \theta & \text{if B} > \text{G} \end{cases}$$

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}} \right\}$$

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

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

Converting colors from HSI to RGB model:

This is done in three sectors. RG, GB and BR sectors.

<u>RG sector:</u> $0^\circ \le H < 120^\circ$

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

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

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

<u>GB Sector:</u> $120^\circ \le H < 240^\circ$

$$H = H - 120^\circ$$

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

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

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

<u>BR Sector:</u> $240^\circ \le H < 360^\circ$

$$H = H - 240^\circ$$

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

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

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

1.6 SAMPLING AND QUANTIZATION

To be suitable for computer processing an image, f(x,y) must be digitized both spatially and in amplitude. Digitizing the spatial coordinates is called *image sampling.* Amplitude digitization is called gray-level quantization.

The one dimensional function shown in fig.9(b) is a plot of amplitude values of the continuous image along the line segment AB in fig.9(a). the random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in fig.9(c).

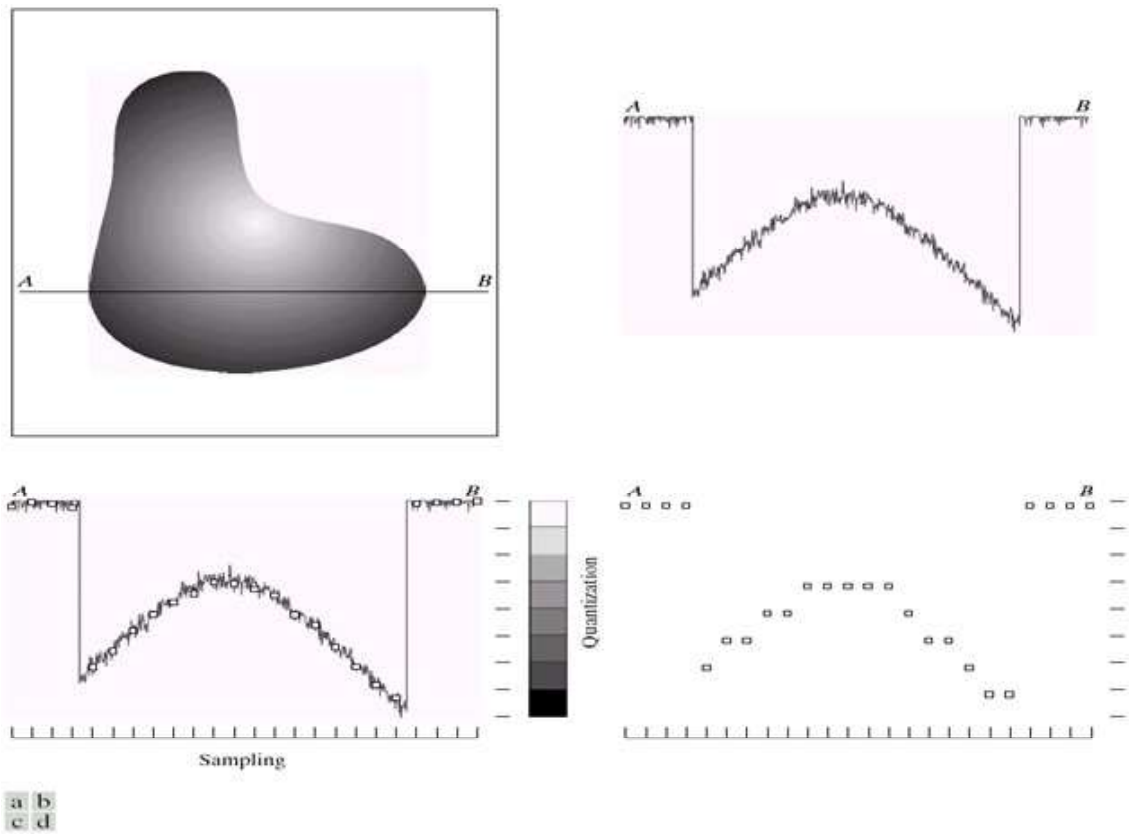


Fig.1.25 Generating a digital image. (a) continuous image (b)A scan line from Ato B in the continuous image (c) sampling and quantization (d) Digital scan line.

The location of each sample is given by vertical tick mark in the bottom part of the figure. The samples are shown as white square box superimposed on the function. The set of these discrete locations gives the sampled function. In order to form a digital function, the gray values must be quantized into digital values. The right side of fig.9(c) shows the gray level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific

value assigned to each of the eight gray levels. The continuous values are quantized simply by assigning one of the eight gray levels to each sample. The digital sample resulting from sampling and quantization is shown in fig.9 (d). Starting from the top of the image and carrying out this procedure line by line produces a two dimensional image.

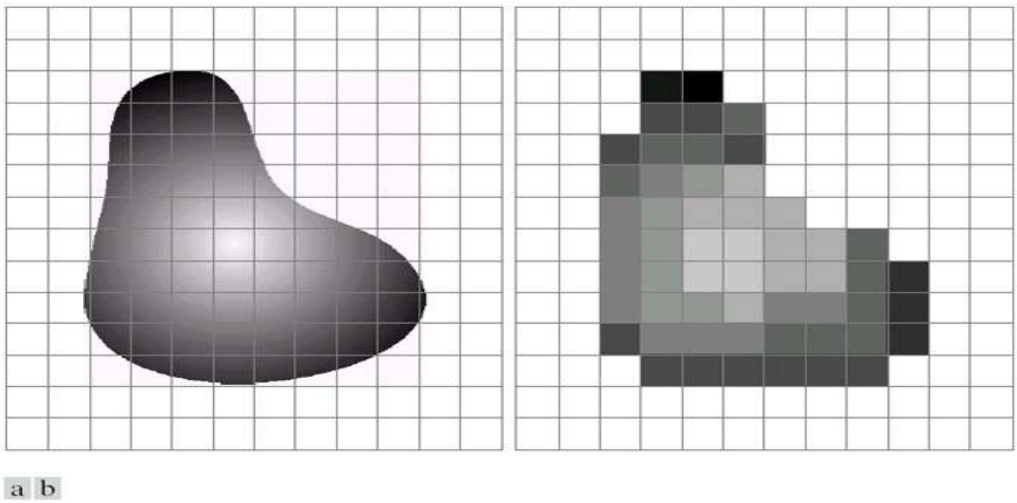


Fig.1.26 (a) Continuous image. (b) Result of sampling and quantization

1.6.1 Representing digital images

The result of sampling and quantization is a matrix of real numbers. Assume that f(x, y) is sampled so that the resulting digital image has M rows and N columns. The values of the coordinates (x,y) now become discrete quantities. The values of the coordinates at the origin are (x, y) = (0, 0). Fig.11 shows the coordinate convention.

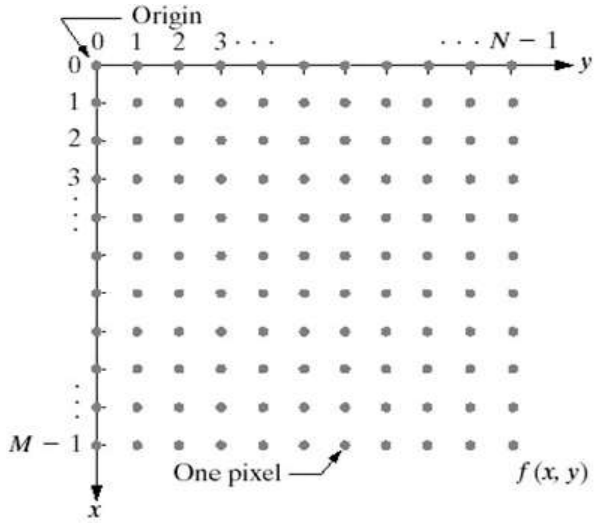


Fig.1.27 Coordinate convention to represent digital images

The complete digital image in matrix form can be represented as

$$f(x,y) \approx \begin{matrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ . & & & \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{matrix}$$

Each element of this matrix array is called an image element, picture element, pixel or pel.
Common practice is to let N and M be powers of two; $N=2^n$ and $M=2^k$ And L = 2^m where L denotes the number of gray levels.
The range of values spanned by the gray scale is called dynamic range. It is given by the interval [0,L-1]
The assumption here is that gray levels are equally space in the interval [0,L] The number of bits, b, necessary to store the image is then

$$b = N*M*m$$

or if N = M

$$b = N^2 m$$

For example, a 128x128 image with 64 gray levels would require 98,304 bits of storage

Spatial and Gray level resolution:

Spatial resolution: is the smallest discernible detail in an Image.
The more pixels in a fixed range, the higher the resolution
Gray-level resolution: refers to the smallest discernible change in gray level.
The more bits, the higher the resolution

Aliasing and Moiré patterns:

The Shannon's sampling theorem tells that, if a function is sampled at a rate equal to or greater than twice its higher frequency it is possible to recover completely the original functions from the samples. If the function is undersampled, then a phenomenon called aliasing corrupts the sampled image, the corruption in the form of additional frequency components being introduced into the sampling function. These are called aliased frequencies. The effect of aliased frequencies can be seen under right conditions in the form of Moiré patterns.

Zooming and Shrinking of digital images:

- Both applied to digital image
- Zooming
  - Creation of new pixel locations
  - Assignment of gray levels to those new locations

- Pixel replication, when increasing the size of an image an integer times
- Nearest neighbor interpolation: the new pixel is assigned the gray value of the closest pixel in the original image.
- Bilinear interpolation: Here the new pixel is assigned the value using the four nearest neighbors.

- Shrinking: Row column deletion

1.7 VIDICON CAMERA WORKING PRINCIPLES

A vidicon tube is a video camera tube design in which the target material is a photoconductor. The Vidicon was developed in the 1950s at RCA by P. K. Weimer, S. V. Forgue and R. R. Goodrich as a simple alternative to the structurally and electrically complex Image Orthicon.[citation needed] While the initial photoconductor used was selenium, other targets–including silicon diode arrays–have been used.
Principle: Photoconductivity

Schematic of vidicon tube.

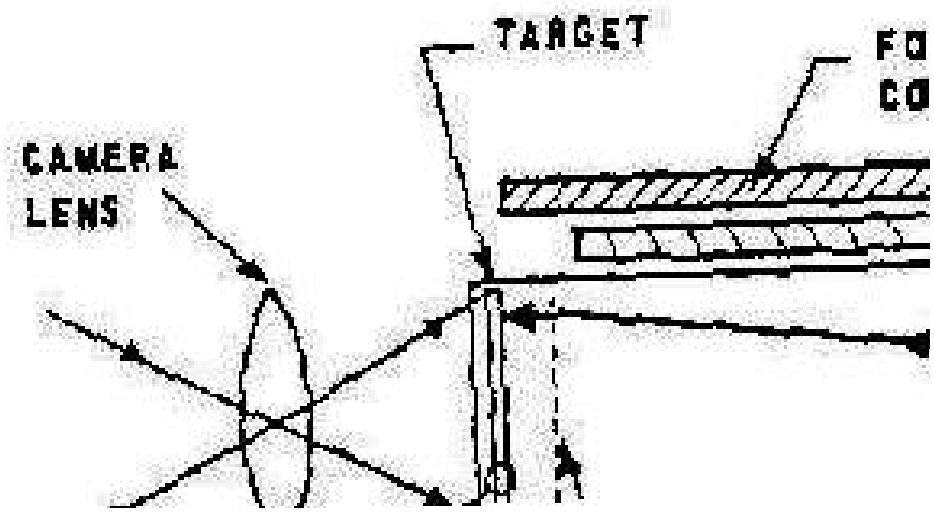


Fig.1.28 Vidicon tube

The vidicon is a storage-type camera tube in which a charge-density pattern is formed by the imaged scene radiation on a photoconductive surface which is then scanned by a beam of low-velocity electrons. The fluctuating voltage coupled out to a video amplifier can be used to reproduce the scene being imaged. The electrical charge produced by an image will remain in the face plate until it is scanned or until the charge dissipates. Pyroelectric photocathodes can be used to produce a vidicon sensitive over a broad portion of the infrared spectrum.

Prior to the design and construction of the Galileo probe to Jupiter in the late 1970s to early 1980s, NASA used Vidicon cameras on most of their unmanned deep space probes equipped with the remote sensing ability.

1.8 DIGITAL CAMERA WORKING PRINCIPLE

Analogous cameras have several advantages but cheaper and cheaper digital video cameras are in use. There are two types of digital cameras:

- CCD (Charged Coupled Device) sensors
- CID (Charge Injection Device) sensors

CCD sensors are dominant (due to price reasons)

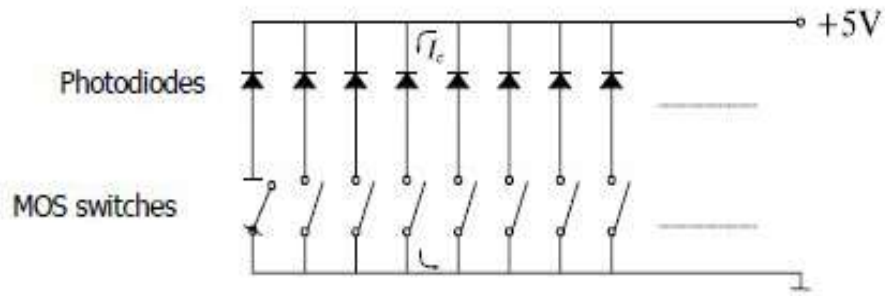


Fig.1.29 Photodiode field

The main part of this camera is photodiode field. Photodiode field is made on the single chip. Diodes works as condensators in “light integrating regime”. Opening of MOS switches causes flow of electricity that is proportional to the charge integrated on diodes. This electricity is proportional to luminance within given period of integration. This is similar (with little bit higher inertia) to the vidicon. There are problems with reading data from this sensor. There are three reading strategy

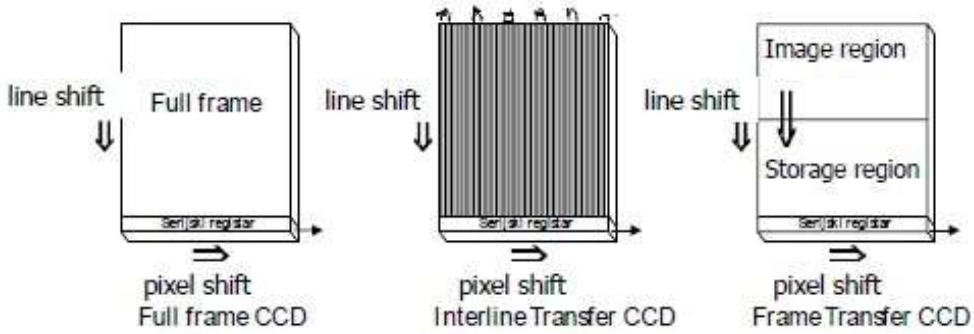


Fig.1.30 Reading strategy

Reading strategies for CCD

Interline transfer
Each even row is masked. After integration interval integrated pixels are moved to masked row and information is read from these rows while new acquisition can begin from non-masked rows.
Advantage:
- Fastness.

Drawback:
- Low percentage of used diode field for acquisition.

In the case of color images acquisition there are color filters that allow only single basic color for one pixel and combination of images from diodes with various filters form the color image.

Full frame transfer
Assume that photodiode filed is matrix with single row masked. This is row not used for light integration. Formed image is moved row by row to the masked row and from masked row is read using shift register.
Advantages:
- Simple hardware and large percentage of the photodiode field used for image acquisition.

Drawbacks:
- Slowness

1.9 TWO-DIMENSIONAL MATHEMATICAL PRELIMINARIES, 2D TRANSFORMS - DFT, DCT, KLT, SVD.

1.9.1. UNITARY TRANSFORMS

One dimensional signals

For a one dimensional sequence $\{f(x), 0 \le x \le N-1\}$ represented as a vector $\underline{f} = [f(0)\, f(1) \ldots f(N-1)]^T$ of size NxN, a transformation may be written as

$$\underline{g} = \underline{T} \cdot \underline{f} \Rightarrow g(u) = \sum_{x=0}^{N-1} T(u,x) f(x), 0 \le u \le N-1$$

where $g(u)$ is the transform (or transformation) of $f(x)$, and $T(u,x)$ is the so called **forward transformation kernel**. Similarly, the inverse transform is the relation

$$f(x) = \sum_{u=0}^{N-1} I(x,u) g(u),\ 0 \le x \le N-1$$

or written in a matrix form

$$\underline{f} = \underline{I} \cdot \underline{g} = \underline{T}^{-1} \cdot \underline{g}$$

where $I(x,u)$ is the so called **inverse transformation kernel**.
If

$$\underline{I} = \underline{T}^{-1} = \underline{T}^{*T}$$

the matrix T is called **unitary**, and the transformation is called unitary as well. It can be proven that the columns (or rows) of an NxN unitary matrix are orthonormal and therefore, form a complete set of **basis vectors** in the N dimensional vector space.
In that case

$$\underline{f} = \underline{T}^{*^T} \cdot \underline{g} \Rightarrow f(x) = \sum_{u=0}^{N-1} T^*(u,x) g(u)$$

The columns of $\underline{T}^{*^T}$, that is, the vectors $\underline{T}_u^* = \left[T^*(u,0)\, T^*(u,1) \ldots T^*(u,N-1)\right]^T$ are called the **basis vectors** of $\underline{T}$.

Two dimensional signals (images)

As a one dimensional signal can be represented by an orthonormal set of **basis vectors**, an image can also be expanded in terms of a discrete set of **basis arrays** called basis images through a **two dimensional (image) transform**.
For an $N \times N$ image f(x,y) the forward and inverse transforms are given below

$$g(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} T(u,v,x,y) f(x,y)$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} I(x,y,u,v) g(u,v)$$

where, again, $T(u,v,x,y)$ and $I(x,y,u,v)$ are called the **forward and inverse transformation kernels**, respectively.
The forward kernel is said to be **separable** if

$$T(u,v,x,y) = T_1(u,x) T_2(v,y)$$

It is said to be **symmetric** if T_1 is functionally equal to T_2 such that

$$T(u,v,x,y) = T_1(u,x) T_1(v,y)$$

The same comments are valid for the inverse kernel.
If the kernel $T(u,v,x,y)$ of an image transform is separable and symmetric, then the transform $g(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} T(u,v,x,y) f(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} T_1(u,x) T_1(v,y) f(x,y)$ can be written in matrix form as follows

$$\underline{g} = \underline{T}_1 \cdot \underline{f} \cdot \underline{T}_1^T$$

where $\underline{f}$ is the original image of size NxN, and $\underline{T}_1$ is an NxN transformation matrix with elements $t_{ij} = T_1(i,j)$. If, in addition, $\underline{T}_1$ is a unitary matrix then the transform is called **separable unitary** and the original image is recovered through the relationship

$$\underline{f} = \underline{T}_1^{*^T} \cdot \underline{g} \cdot \underline{T}_1^*$$

1.9.1.2 Fundamental properties of unitary transforms

The property of energy preservation
In the unitary transformation

$$\underline{g} = \underline{T} \cdot \underline{f}$$

it is easily proven (try the proof by using the relation $\underline{T}^{-1} = \underline{T}^{*^T}$) that

$$\|\underline{g}\|^2 = \|\underline{f}\|^2$$

Thus, a unitary transformation preserves the signal energy. This property is called energy preservation property.

This means that every unitary transformation is simply a rotation of the vector f in the N dimensional vector space.

For the 2-D case the energy preservation property is written as

$$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}|f(x,y)|^2 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}|g(u,v)|^2$$

The property of energy compaction

Most unitary transforms pack a large fraction of the energy of the image into relatively few of the transform coefficients. This means that relatively few of the transform coefficients have significant values and these are the coefficients that are close to the origin (small index coefficients).

This property is very useful for compression purposes.

1.9.2. THE TWO DIMENSIONAL FOURIER TRANSFORM

1.9.2.1 Continuous space and continuous frequency

The Fourier transform is extended to a function $f(x,y)$ of two variables. If $f(x,y)$ is continuous and integrable and $F(u,v)$ is integrable, the following Fourier transform pair exists:

$$F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)e^{-j2\pi(ux+vy)}dxdy$$

$$f(x,y) = \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\int F(u,v)e^{j2\pi(ux+vy)}dudv$$

In general $F(u,v)$ is a complex-valued function of two real frequency variables u, v and hence, it can be written as:

$$F(u,v) = R(u,v) + jI(u,v)$$

The amplitude spectrum, phase spectrum and power spectrum, respectively, are defined as follows.

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

1.9.2.2 Discrete space and continuous frequency

For the case of a discrete sequence $f(x,y)$ of infinite duration we can define the 2-D discrete space Fourier transform pair as follows

$$F(u,v) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f(x,y) e^{-j(xu+vy)}$$

$$f(x,y) = \frac{1}{(2\pi)^2} \int_{u=-\pi}^{\pi} \int_{v=-\pi}^{\pi} F(u,v) e^{j(xu+vy)} du dv$$

$F(u,v)$ is again a complex-valued function of two real frequency variables u, v and it is periodic with a period $2\pi \times 2\pi$, that is to say $F(u,v) = F(u+2\pi, v) = F(u, v+2\pi)$

The Fourier transform of $f(x,y)$ is said to converge uniformly when $F(u,v)$ is finite and

$$\lim_{N_1 \to \infty} \lim_{N_2 \to \infty} \sum_{x=-N_1}^{N_1} \sum_{y=-N_2}^{N_2} f(x,y) e^{-j(xu+vy)} = F(u,v) \text{ for all } u, v.$$

When the Fourier transform of $f(x,y)$ converges uniformly, $F(u,v)$ is an analytic function and is infinitely differentiable with respect to u and v.

1.9.2.3 Discrete space and discrete frequency: The two dimensional Discrete Fourier Transform (2-D DFT)

If $f(x,y)$ is an $M \times N$ array, such as that obtained by sampling a continuous function of two dimensions at dimensions M and N on a rectangular grid, then its two dimensional Discrete Fourier transform (DFT) is the array given by

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

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

and the inverse DFT (IDFT) is

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

When images are sampled in a square array, $M = N$ and

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

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

It is straightforward to prove that the two dimensional Discrete Fourier Transform is separable, symmetric and unitary.

1.9.2.4 Properties of the 2-D DFT

Most of them are straightforward extensions of the properties of the 1-D Fourier Transform. Advise any introductory book on Image Processing.

- **The importance of the phase in 2-D DFT. Image reconstruction from amplitude or phase only.**

The Fourier transform of a sequence is, in general, complex-valued, and the unique representation of a sequence in the Fourier transform domain requires both the phase and the magnitude of the Fourier transform. In various contexts it is often desirable to reconstruct a

signal from only partial domain information. Consider a 2-D sequence $f(x,y)$ with Fourier transform $F(u,v) = \Im\{f(x,y)\}$ so that

$$F(u,v) = \Im\{f(x,y\} = |F(u,v)|e^{j\phi_f(u,v)}$$

It has been observed that a straightforward signal synthesis from the Fourier transform phase $\phi_f(u,v)$ alone, often captures most of the intelligibility of the original image $f(x,y)$. A straightforward synthesis from the Fourier transform magnitude $|F(u,v)|$ alone, however, does not generally capture the original signal's intelligibility. The above observation is valid for a large number of signals (or images). To illustrate this, we can synthesise the phase-only signal $f_p(x,y)$ and the magnitude-only signal $f_m(x,y)$ by

$$f_p(x,y) = \Im^{-1}\left[1e^{j\phi_f(u,v)}\right]$$
$$f_m(x,y) = \Im^{-1}\left[|F(u,v)|e^{j0}\right]$$

and observe the two results. An experiment which more dramatically illustrates the observation that phase-only signal synthesis captures more of the signal intelligibility than magnitude-only synthesis, can be performed as follows.

Consider two images $f(x,y)$ and $g(x,y)$. From these two images, we synthesise two other images $f_1(x,y)$ and $g_1(x,y)$ by mixing the amplitudes and phases of the original images as follows:

$$f_1(x,y) = \Im^{-1}\left[|G(u,v)|e^{j\phi_f(u,v)}\right]$$
$$g_1(x,y) = \Im^{-1}\left[|F(u,v)|e^{j\phi_g(u,v)}\right]$$

In this experiment $f_1(x,y)$ captures the intelligibility of $f(x,y)$, while $g_1(x,y)$ captures the intelligibility of $g(x,y)$.

- **Periodicity of the Fourier transform**

The discrete Fourier transform (and its inverse) are *periodic* with period N.

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

Although F(u,v) repeats itself infinitely for many values of *u* and *v*, only *N* values of each variable are required to obtain f(x,y) from F(u,v) i.e. Only one period of the transform is necessary to specify F(u,v) in the frequency domain. Similar comments may be made for f(x,y) in the spatial domain

- **Conjugate symmetry of the Fourier Transform**

If *f(x,y)* is real (true for all of our cases), the Fourier transform exhibits *conjugate symmetry*

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

or, the more interesting

|F(*u,v*)| = |F(-*u*,-*v*)|

where F*(*u,v*) is the complex conjugate of F(*u,v*)

- **Distributive & Scaling**

The Fourier transform (and its inverse) are distributive over addition but not over multiplication So,

$$\Im\{f_1(x,y)+f_2(x,y)\}=\Im\{f_1(x,y)\}+\Im\{f_2(x,y)\}$$

$$\Im\{f_1(x,y)\times f_2(x,y)\}\neq\Im\{f_1(x,y)\}\times\Im\{f_2(x,y)\}$$

For two scalars *a* and *b*,

$$af(x,y)\Leftrightarrow aF(u,v)$$

$$f(ax,by)\Leftrightarrow\frac{1}{|ab|}F(u/a,v/b)$$

- **Average Value**

A widely used expression for the average value of a 2-D discrete function is:

$$\overline{f}(x,y)=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)$$

From the definition of F(u,v), for u=v=0,

$$F(0,0)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)$$

Therefore,

$$\overline{f}(x,y)=\frac{1}{N}F(0,0)$$

- **The Laplacian**

The Laplacian of a two variable function f(x,y) is given as:

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

From the definition of the 2-D Fourier transform

$$\Im\{\nabla^2 f(x,y)\}\Leftrightarrow-(2\pi)^2(u^2+v^2)F(u,v)$$

The Laplacian operator is useful for outlining edges in an image

- **Convolution**

The convolution of two functions f(x) and g(x) is denoted f(x)*g(x) and is given by:

$$f(x)*g(x)=\int_{-\infty}^{+\infty} f(\alpha)g(x-\alpha)d\alpha$$

1.9.3. THE DISCRETE COSINE TRANSFORM (DCT)

1.9.3.1 One dimensional signals

This is a transform that is similar to the Fourier transform in the sense that the new independent variable represents again frequency. The DCT is defined below.

$$C(u)=a(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right],\ u=0,1,\ldots,N-1$$

with $a(u)$ a parameter that is defined below.

$$a(u)=\begin{cases}\sqrt{1/N} & u=0\\ \sqrt{2/N} & u=1,\ldots,N-1\end{cases}$$

The inverse DCT (IDCT) is defined below.

$$f(x)=\sum_{u=0}^{N-1} a(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

1.9.3.2 Two dimensional signals (images)

For 2-D signals it is defined as

$$C(u,v)=a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} a(u)a(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$a(u)$ is defined as above and $u,v=0,1,\ldots,N-1$

1.9.3.3 Properties of the DCT transform

- The DCT is a real transform. This property makes it attractive in comparison to the Fourier transform.
- The DCT has excellent energy compaction properties. For that reason it is widely used in image compression standards (as for example JPEG standards).
- There are fast algorithms to compute the DCT, similar to the FFT for computing the DFT.

1.9.4. KARHUNEN-LOEVE (KLT) or HOTELLING TRANSFORM

The Karhunen-Loeve Transform or KLT was originally introduced as a series expansion for continuous random processes by Karhunen and Loeve. For discrete signals Hotelling first studied what was called a method of principal components, which is the discrete equivalent of the KL

series expansion. Consequently, the KL transform is also called the Hotelling transform or the method of principal components. The term KLT is the most widely used.

-
- **The case of many realisations of a signal or image**

The concepts of **eigenvalue** and **eigevector** are necessary to understand the KL transform.

If $\underline{C}$ is a matrix of dimension $n \times n$, then a scalar λ is called an eigenvalue of $\underline{C}$ if there is a nonzero vector $\underline{e}$ in R^n such that $\underline{C}\underline{e} = \lambda \underline{e}$ The vector $\underline{e}$ is called an eigenvector of the matrix $\underline{C}$ corresponding to the eigenvalue λ.

Consider a population of random vectors of the form

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The **mean vector** of the population is defined as

$$\underline{m}_x = E\{\underline{x}\}$$

The operator E refers to the expected value of the population calculated theoretically using the probability density functions (pdf) of the elements x_i and the joint probability density functions between the elements x_i and x_j.

The covariance matrix of the population is defined as

$$\underline{C}_x = E\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\}$$

Because $\underline{x}$ is n-dimensional, $\underline{C}_x$ and $(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T$ are matrices of order $n \times n$. The element c_{ii} of $\underline{C}_x$ is the variance of x_i, and the element c_{ij} of $\underline{C}_x$ is the covariance between the elements x_i and x_j. If the elements x_i and x_j are uncorrelated, their covariance is zero and, therefore,

$$c_{ij} = c_{ji} = 0 .$$

For M vectors from a random population, where M is large enough, the mean vector and covariance matrix can be approximately calculated from the vectors by using the following relationships where all the expected values are approximated by summations

$$\underline{m}_x = \frac{1}{M} \sum_{k=1}^{M} \underline{x}_k$$

$$\underline{C}_x = \frac{1}{M} \sum_{k=1}^{M} \underline{x}_k \underline{x}_k^T - \underline{m}_x \underline{m}_x^T$$

Very easily it can be seen that $\underline{C}_x$ is real and symmetric. In that case a set of n orthonormal eigenvectors always exists. Let $\underline{e}_i$ and λ_i, $i = 1,2,\ldots,n$, be this set of eigenvectors and corresponding eigenvalues of $\underline{C}_x$, arranged in descending order so that $\lambda_i \geq \lambda_{i+1}$ for $i = 1,2,\ldots,n-1$. Let $\underline{A}$ be a matrix whose rows are formed from the eigenvectors of $\underline{C}_x$, ordered

so that the first row of $\underline{A}$ is the eigenvector corresponding to the largest eigenvalue, and the last row the eigenvector corresponding to the smallest eigenvalue.

Suppose that $\underline{A}$ is a transformation matrix that maps the vectors $\underline{x}'s$ into vectors $\underline{y}'s$ by using the following transformation

$$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$$

The above transform is called the **Karhunen-Loeve** or **Hotelling** transform. The mean of the $\underline{y}$ vectors resulting from the above transformation is zero

$$\underline{m}_y = \underline{0}$$

the covariance matrix is

$$\underline{C}_y = \underline{A}\underline{C}_x\underline{A}^T$$

and $\underline{C}_y$ is a diagonal matrix whose elements along the main diagonal are the eigenvalues of $\underline{C}_x$

$$\underline{C}_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

The off-diagonal elements of the covariance matrix are 0, so the elements of the $\underline{y}$ vectors are uncorrelated.

Let's try to reconstruct any of the original vectors $\underline{x}$ from its corresponding $\underline{y}$. Because the rows of $\underline{A}$ are orthonormal vectors, then $\underline{A}^{-1} = \underline{A}^T$, and any vector $\underline{x}$ can by recovered from its corresponding vector $\underline{y}$ by using the relation

$$\underline{x} = \underline{A}^T\underline{y} + \underline{m}_x$$

Suppose that instead of using all the eigenvectors of $\underline{C}_x$ we form matrix $\underline{A}_K$ from the K eigenvectors corresponding to the K largest eigen values, yielding a transformation matrix of order $K \times n$. The $\underline{y}$ vectors would then be K dimensional, and the reconstruction of any of the original vectors would be approximated by the following relationship

$$\hat{\underline{x}} = \underline{A}_K^T\underline{y} + \underline{m}_x$$

The mean square error between the perfect reconstruction $\underline{x}$ and the approximate reconstruction $\hat{\underline{x}}$ is given by the expression

$$e_{ms} = \sum_{j=1}^{n}\lambda_j - \sum_{j=1}^{K}\lambda_j = \sum_{j=K+1}^{n}\lambda_j .$$

By using $\underline{A}_K$ instead of $\underline{A}$ for the KL transform we achieve compression of the available data.

- **The case of <u>one</u> realisation of a signal or image**

The derivation of the KLT for the case of one image realisation assumes that the two dimensional signal (image) is **ergodic**. This assumption allows us to calculate the statistics of the image using only one realisation. Usually we divide the image into blocks and we apply the KLT in each block. This is reasonable because the 2-D field is likely to be ergodic within a small block since the nature of the signal changes within the whole image. Let's suppose that $\underline{f}$ is a

vector obtained by lexicographic ordering of the pixels $f(x,y)$ within a block of size $M \times M$ (placing the rows of the block sequentially).
The mean vector of the random field inside the block is a scalar that is estimated by the approximate relationship

$$m_f = \frac{1}{M^2}\sum_{k=1}^{M^2} \underline{f}(k)$$

and the covariance matrix of the 2-D random field inside the block is $\underline{C}_f$ where

$$c_{ii} = \frac{1}{M^2}\sum_{k=1}^{M^2} \underline{f}(k)\underline{f}(k) - m_f^2$$

and

$$c_{ij} = c_{i-j} = \frac{1}{M^2}\sum_{k=1}^{M^2} \underline{f}(k)\underline{f}(k+i-j) - m_f^2$$

After knowing how to calculate the matrix $\underline{C}_f$, the KLT for the case of a single realisation is the same as described above.

1.9.4.1 Properties of the Karhunen-Loeve transform

Despite its favourable theoretical properties, the KLT is not used in practice for the following reasons.

- Its basis functions depend on the covariance matrix of the image, and hence they have to recomputed and transmitted for every image.
- Perfect decorrelation is not possible, since images can rarely be modelled as realisations of ergodic fields.
- There are no fast computational algorithms for its implementation.

1.9.5 SINGULAR VALUE DECOMPOSITION:

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix, with many useful applications in signal processing and statistics. Formally, the singular value decomposition of an m×n real or complex matrix M is a factorization of the form

$$M = U\Sigma V^*,$$

where U is an m×m real or complex unitary matrix, Σ is an m×n rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V* (the conjugate transpose of V) is an n×n real or complex unitary matrix. The diagonal entries $\Sigma_{i,i}$ of Σ are known as the singular values of M. The m columns of U and the n columns of V are called the left-singular vectors and right-singular vectors of M, respectively. The singular value decomposition and the eigen decomposition are closely related. Namely:

- The left-singular vectors of M are eigenvectors of MM*.
- The right-singular vectors of M are eigenvectors of M*M.

- The non-zero-singular values of M (found on the diagonal entries of Σ) are the square roots of the non-zero eigen values of both M*M and MM*.

Applications which employ the SVD include computing the pseudo-inverse, least squares fitting of data, matrix approximation, and determining the rank, range and null space of a matrix.

Applications of SVD:

The singular value decomposition can be used for computing the pseudo-inverse of a matrix. Indeed, the pseudo-inverse of the matrix M with singular value decomposition

$$M = U\Sigma V^* \text{ is}$$

$$M^+ = V\Sigma^+ U^*,$$

where Σ^+ is the pseudo-inverse of Σ, which is formed by replacing every nonzero diagonal entry by its reciprocal and transposing the resulting matrix. The pseudo-inverse is one way to solve linear least squares problems.

A set of homogeneous linear equations can be written as $\mathbf{A}\mathbf{x} = \mathbf{0}$ for a matrix $\mathbf{A}$ and vector $\mathbf{x}$. A typical situation is that $\mathbf{A}$ is known and a non-zero $\mathbf{x}$ is to be determined which satisfies the equation. Such an $\mathbf{x}$ belongs to $\mathbf{A}$'s null space and is sometimes called a (right) null vector of $\mathbf{A}$. $\mathbf{x}$ can be characterized as a right-singular vector corresponding to a singular value of $\mathbf{A}$ that is zero. This observation means that if $\mathbf{A}$ is a square matrix and has no vanishing singular value, the equation has no non-zero $\mathbf{x}$ as a solution. It also means that if there are several vanishing singular values, any linear combination of the corresponding right-singular vectors is a valid solution. Analogously to the definition of a (right) null vector, a non-zero $\mathbf{x}$ satisfying $\mathbf{x}^* \mathbf{A} = \mathbf{0}$, with $\mathbf{x}^*$ denoting the conjugate transpose of $\mathbf{x}$, is called a left null vector of $\mathbf{A}$.

A total least squares problem refers to determining the vector $\mathbf{x}$ which minimizes the 2-norm of a vector $\mathbf{A}\mathbf{x}$ under the constraint $\|\mathbf{x}\| = 1$. The solution turns out to be the right-singular vector of $\mathbf{A}$ corresponding to the smallest singular value.

Another application of the SVD is that it provides an explicit representation of the range and null space of a matrix M. The right-singular vectors corresponding to vanishing singular values of M span the null space of M. E.g., the null space is spanned by the last two columns of V in the above example. The left-singular vectors corresponding to the non-zero singular values of M span the range of M. As a consequence, the rank of M equals the number of non-zero singular values which is the same as the number of non-zero diagonal elements in Σ.

In numerical linear algebra the singular values can be used to determine the effective rank of a matrix, as rounding error may lead to small but non-zero singular values in a rank deficient matrix.

1.10 DITHER:

In digital processing and waveform analysis dither is often used in digital audio and video processing, where it is applied to bit-depth transitions; it is utilized in many different fields where digital processing and analysis are used especially for waveform analysis. These uses include systems using digital signal processing, such as digital audio, digital video, digital photography, seismology, RADAR, weather forecasting systems and many more.

The premise is that quantization and re-quantization of digital data yields error. If that error is repeating and correlated to the signal, the error that results is repeating, cyclical, and mathematically determinable. In some fields, especially where the receptor is sensitive to such artifacts, cyclical errors yield undesirable artifacts. In these fields dither results in less determinable artifacts. The field of audio is a primary example of this, when the human ear functions much like a Fourier transform, wherein it hears individual frequencies). The ear is therefore very sensitive to distortion, or additional frequency content that "colors" the sound differently, but far less sensitive to random noise at all frequencies.

Summary

The material in this chapter provides background information which is required for the further chapters. It gives the idea of the elements required for the digital image processing systems. Elements of visual perception give the basic idea of the capabilities of the eye in perceiving information. The discussion of light and electromagnetic spectrum provides information about the formation of images. This chapter provides the basics behind the color image fundamentals and the various models.
The Image sampling ideas introduced are the foundation for digitizing phenomena's. The ideas of processing the image in frequency domain are provided in two-dimensional mathematical preliminaries and in 2D transforms like DFT, DCT, KLT, and SVD. Finally the working concepts of image acquisition systems like Vidicon and Digital Camera are also discussed.

References

The major diagrams of this chapter are brought from the book by Rafael C. Gonzalez and Richard E. Woods. The contents of this chapter are collected from Princeton University, NPTEL, IIT, New Mexican Tech.

UNIT II

IMAGE ENHANCEMENT

Histogram equalization and specification techniques, Noise distributions, Spatial averaging, Directional Smoothing, Median, Geometric mean, Harmonic mean, Contraharmonic mean filters, Homomorphic filtering, Color image enhancement.

2.1 Spatial domain methods

Suppose we have a digital image which can be represented by a two dimensional random field $f(x,y)$.

An image processing operator in the spatial domain may be expressed as a mathematical function $T[\cdot]$ applied to the image $f(x,y)$ to produce a new image $g(x,y)=T[f(x,y)]$ as follows.

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

The operator T applied on $f(x,y)$ may be defined over:

(i) A single pixel (x,y). In this case T is a grey level transformation (or mapping) function.

(ii) Some neighbourhood of (x,y).

(iii) T may operate to a set of input images instead of a single image.

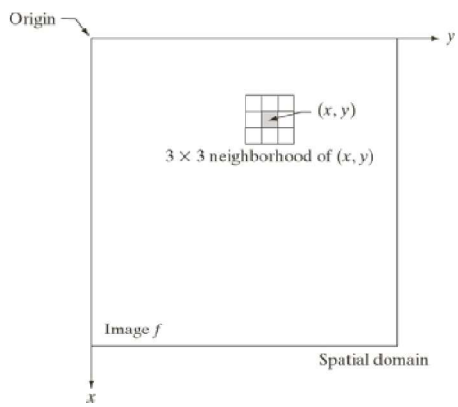


Fig 2.1 3x3 neighbourhood about a point (x,y) in an image

Spatial filtering

Generally involves operations over the entire image. Operations take place involving pixels within a neighborhood of a point of interest (x,y). Also involves a predefined operation called a *spatial filter*. The *spatial filter* is also commonly referred to as Spatial mask, Kernel, Template and window

Point Processing

The smallest neighborhood of a pixel is 1x1 in size. Here, *g* depends only on the value of *f* at

(x,y) *T* becomes an *intensity transformation function* of s=T(r)

Here *s* and *r* represent the intensity of *g* and *f* at any point (x,y). Also called a *gray-level* or *mapping function.*

Some Basic Intensity Transformation Functions

Here, T is a transformation that maps a pixel value r into a pixel value s. Since we are concerned with digital data, the transformation can generally be implemented with a simple lookup table. Three basic types of transformations

– Linear (negative and identity transformations)

– Logarithmic (log and inverse-log transformations)

– Power-law (n^{th} power and n^{th} root transformations)

Image Negatives

The negative of an image with intensity levels in the range [0,L-1] can be described by: $s = L - 1 - r$

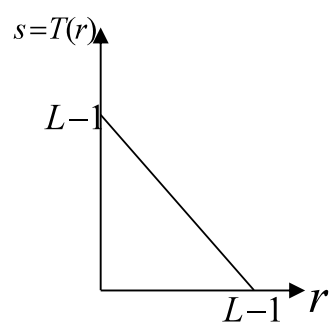


Fig:2.2 Image Negative

Log Transformations

General form: s= c log(1 □r) c is a constant and r≥0 Maps a narrow range of low intensity values in input to a wider output range. The opposite is true for high intensity input values. Compresses the dynamic range of images with large variations in pixel values.

Power-Law (Gamma) Transformations:

Basic form $S=cr^{\gamma}$

Where *c* and γ are positive constants. Power- law curves with fractional values of γ map a narrow range of dark input values to a wider range of output values. The opposite is true for higher values of input levels exists a family of possible transformation curves by varying γ

Power-Law Transformation Curves

Gamma Correction

Many devices used for image capture, display and printing respond according to a power law The exponent in the power-law equation is referred to as *gamma.* The process of correcting for the power-law response is referred to as *gamma correction*

Example:

CRT devices have an intensity-to-voltage response that is a power function (exponents typically range from 1.8-2.5). Gamma correction in this case could be achieved by applying the transformation $s=r^{1/2.5}=r^{0.4}$

Piecewise-Linear Transformations

Piecewise functions can be arbitrarily complex. A disadvantage is that their specification requires significant user input

Example functions:Contrast stretching, Intensity-level slicing, Bit-plane slicing

Contrast Stretching

Contrast stretching expands the range of intensity levels in an image so it spans a given (full) intensity range Control points (r_1,s_1) and (r_2,s_2) control the shape of the transform T(r) $r_1=r_2$, $s_1=0$ and s_2=L-1 yields a *thresholding function*

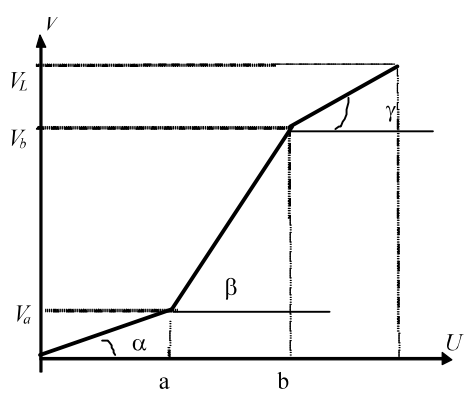


Fig.2.3 Contrast Stretching

Intensity-level Slicing

Used to highlight a specific range of intensities in an image that might be of interest. Two common approaches. Set all pixel values within a range of interest to one value (white) and all others to another value (black), Produces a binary image. Brighten (or darken) pixel values in a range of interest and leave all others unchanged.

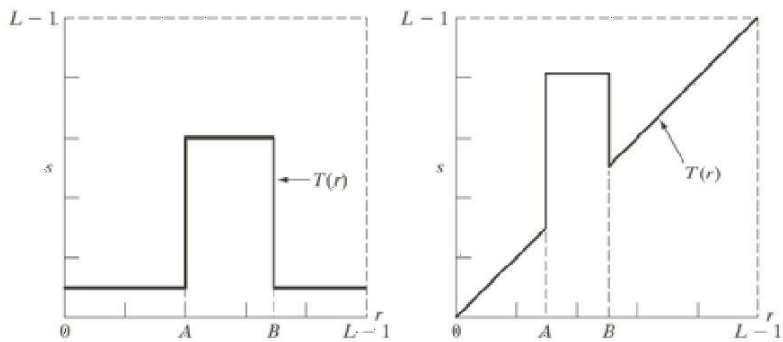


Fig.2.4 Intensity-level Slicing

2.2 Frequency domain methods

Let $g(x,y)$ be a desired image formed by the convolution of an image $f(x,y)$ and a linear, position invariant operator $h(x,y)$, that is:

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

The following frequency relationship holds:

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

We can select $H(u,v)$ so that the desired image

$$g(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$

exhibits some highlighted features of $f(x,y)$. For instance, edges in $f(x,y)$ can be accentuated by using a function $H(u,v)$ that emphasises the high frequency components of $F(u,v)$.

➢

2.3 Histogram processing. Definition of the histogram of an image.

By processing (modifying) the histogram of an image we can create a new image with specific desired properties.

Suppose we have a digital image of size $N \times N$ with grey levels in the range $[0, L-1]$. The histogram of the image is defined as the following discrete function:

$$p(r_k) = \frac{n_k}{N^2}$$

where

r_k is the kth grey level, $k = 0,1,\ldots,L-1$

n_k is the number of pixels in the image with grey level r_k
N^2 is the total number of pixels in the image
The histogram represents the frequency of occurrence of the various grey levels in the image. A plot of this function for all values of k provides a global description of the appearance of the image.

2.3.1 Global histogram equalisation

In this section we will assume that the image to be processed has a continuous intensity that lies within the interval $[0, L-1]$. Suppose we divide the image intensity with its maximum value $L-1$. Let the variable r represent the new grey levels (image intensity) in the image, where now $0 \le r \le 1$ and let $p_r(r)$ denote the probability density function (pdf) of the variable r. We now apply the following transformation function to the intensity

$$s = T(r) = \int_0^r p_r(w)dw, \qquad 0 \le r \le 1$$

(1) By observing the transformation of equation (1) we immediately see that it possesses the following properties:

(i) $0 \le s \le 1$.

(ii) $r_2 > r_1 \Rightarrow T(r_2) \ge T(r_1)$, i.e., the function $T(r)$ is increase ng with r.

(iii) $s = T(0) = \int_0^0 p_r(w)dw = 0$ and $s = T(1) = \int_0^1 p_r(w)dw = 1$. Moreover, if the original image has intensities only within a certain range $[r_{\min}, r_{\max}]$ then $s = T(r_{\min}) = \int_0^{r_{\min}} p_r(w)dw = 0$ and $s = T(r_{\max}) = \int_0^{r_{\max}} p_r(w)dw = 1$ since $p_r(r) = 0, r < r_{\min}$ and $r > r_{\max}$. Therefore, the new intensity s takes always all values within the available range [0 1].

Suppose that $P_r(r)$, $P_s(s)$ are the probability distribution functions (PDF's) of the variables r and s respectively.
Let us assume that the original intensity lies within the values r and $r + dr$ with dr a small quantity. dr can be assumed small enough so as to be able to consider the function $p_r(w)$ constant within the interval $[r, r+dr]$ and equal to $p_r(r)$. Therefore,

$$P_r[r, r+dr] = \int_r^{r+dr} p_r(w)dw \cong p_r(r) \int_r^{r+dr} dw = p_r(r)dr.$$

Now suppose that $s = T(r)$ and $s_1 = T(r+dr)$. The quantity dr can be assumed small enough so as to be able to consider that $s_1 = s + ds$ with ds small enough so as to be able to consider the function $p_s(w)$ constant within the interval $[s, s+ds]$ and equal to $p_s(s)$. Therefore,

$$P_s[s, s+ds] = \int_s^{s+ds} p_s(w)dw \cong p_s(s) \int_s^{s+ds} dw = p_s(s)ds$$

Since $s = T(r)$, $s + ds = T(r+dr)$ and the function of equation (1) is increasing with r, all and only the values within the interval $[r, r+dr]$ will be mapped within the interval $[s, s+ds]$. Therefore,

$$P_r[r, r+dr] = P_s[s, s+ds] \Rightarrow p_r(r)dr \overset{r=T^{-1}(s)}{=} p_s(s)ds \Rightarrow p_s(s) = p_r(r)\left.\frac{dr}{ds}\right|_{r=T^{-1}(s)}$$

From equation (1) we see that

$$\frac{ds}{dr} = p_r(r)$$

and hence,

$$p_s(s) = \left[p_r(r)\frac{1}{p_r(r)}\right]_{r=T^{-1}(s)} = 1,\ 0 \le s \le 1$$

From the above analysis it is obvious that the transformation of equation (1) converts the original image into a new image with uniform probability density function. This means that in the new image all intensities are present [look at property (iii) above] and with equal probabilities. The whole range of intensities from the absolute black to the absolute white is explored and the new image will definitely have higher contrast compared to the original image.

Unfortunately, in a real life scenario we must deal with digital images. The discrete form of histogram equalisation is given by the relation

$$s_k = T(r_k) = \sum_{j=0}^{k}\frac{n_j}{N^2} = \sum_{j=0}^{k} p_r(r_j),\ 0 \le r_k \le 1,\ k = 0,1,\ldots,L-1$$

(2) The quantities in equation (2) have been defined in Section 2.2. To see results of histogram equalisation look at any introductory book on Image Processing.
The improvement over the original image is quite evident after using the technique of histogram equalisation. The new histogram is not flat because of the discrete approximation of the probability density function with the histogram function. Note, however, that the grey levels of an image that has been subjected to histogram equalisation are spread out and always reach white. This process increases the dynamic range of grey levels and produces an increase in image contrast.

2.3.2 Local histogram equalisation

Global histogram equalisation is suitable for overall enhancement. It is often necessary to enhance details over small areas. The number of pixels in these areas may have negligible influence on the computation of a global transformation, so the use of this type of transformation does not necessarily guarantee the desired local enhancement. The solution is to devise transformation functions based on the grey level distribution – or other properties – in the neighbourhood of every pixel in the image. The histogram processing technique previously described is easily adaptable to local enhancement. The procedure is to define a square or rectangular neighbourhood and move the centre of this area from pixel to pixel. At each location the histogram of the points in the neighbourhood is computed and a histogram equalisation transformation function is obtained. This function is finally used to map the grey level of the pixel centred in the neighbourhood. The centre of the neighbourhood region is then moved to an adjacent pixel location and the procedure is repeated. Since only one new row or column of the neighbourhood changes during a pixel-to-pixel translation of the region, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible

quite easily. This approach has obvious advantages over repeatedly computing the histogram over all pixels in the neighbourhood region each time the region is moved one pixel location. Another approach often used to reduce computation is to utilise non overlapping regions, but this methods usually produces an undesirable checkerboard effect.

2.3.3 Histogram specification

Suppose we want to specify a particular histogram shape (not necessarily uniform) which is capable of highlighting certain grey levels in the image.
Let us suppose that:
$p_r(r)$ is the original probability density function
$p_z(z)$ is the desired probability density function
Suppose that histogram equalisation is first applied on the original image r

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

Suppose that the desired image z is available and histogram equalisation is applied as well

$$v = G(z) = \int_0^z p_z(w)dw$$

$p_s(s)$ and $p_v(v)$ are both uniform densities and they can be considered as identical. Note that the final result of histogram equalisation is independent of the density inside the integral. So in equation $v = G(z) = \int_0^z p_z(w)dw$ we can use the symbol s instead of v.

The inverse process $z = G^{-1}(s)$ will have the desired probability density function. Therefore, the process of histogram specification can be summarised in the following steps.

(i) We take the original image and equalise its intensity using the relation $s = T(r) = \int_0^r p_r(w)dw$.

(ii) From the given probability density function $p_z(z)$ we specify the probability distribution function $G(z)$.

(iii) We apply the inverse transformation function $z = G^{-1}(s) = G^{-1}[T(r)]$

2.4 IMAGE AVERAGING

Suppose that we have an image $f(x,y)$ of size $M \times N$ pixels corrupted by noise $n(x,y)$, so we obtain a noisy image as follows.

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

For the noise process $n(x,y)$ the following assumptions are made.

(i) The noise process $n(x,y)$ is erotic.

(ii) It is zero mean, i.e., $E\{n(x,y)\} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} n(x,y) = 0$

(iii) It is white, i.e., the autocorrelation function of the noise process defined as $R[k,l]=E\{n(x,y)n(x+k,y+l)\}=\frac{1}{(M-k)(N-l)}\sum_{x=0}^{M-1-k}\sum_{y=0}^{N-1-l}n(x,y)n(x+k,y+l)$ is zero, apart for the pair $[k,l]=[0,0]$.

Therefore, $R[k,l]=\frac{1}{(M-k)(N-l)}\sum_{x=0}^{M-1-k}\sum_{y=0}^{N-1-l}n(x,y)n(x+k,y+l)=\sigma^2_{n(x,y)}\delta(k,l)$ where $\sigma^2_{n(x,y)}$ is the variance of noise.Suppose now that we have L different noisy realisations of the same image $f(x,y)$ as $g_i(x,y)=f(x,y)+n_i(x,y)$, $i=0,1,\ldots,L$. Each noise process $n_i(x,y)$ satisfies the properties (i)-(iii) given above. Moreover, $\sigma^2_{n_i(x,y)}=\sigma^2$. We form the image $\bar{g}(x,y)$ by averaging these L noisy images as follows:

$$\bar{g}(x,y)=\frac{1}{L}\sum_{i=1}^{L}g_i(x,y)=\frac{1}{L}\sum_{i=1}^{L}(f(x,y)+n_i(x,y))=f(x,y)+\frac{1}{L}\sum_{i=1}^{L}n_i(x,y)$$

Therefore, the new image is again a noisy realisation of the original image $f(x,y)$ with noise $n(x,y)=\frac{1}{L}\sum_{i=1}^{L}n_i(x,y)$ .The mean value of the noise $n(x,y)$ is found below.

$$E\{n(x,y)\}=E\{\frac{1}{L}\sum_{i=1}^{L}n_i(x,y)\}=\frac{1}{L}\sum_{i=1}^{L}E\{n_i(x,y)\}=0$$

The variance of the noise $n(x,y)$ is now found below.

$$\sigma^2_{n(x,y)}=E\{n^2(x,y)\}=E\left\{\left(\frac{1}{L}\sum_{i=1}^{L}n_i(x,y)\right)^2\right\}=\frac{1}{L^2}E\left\{\left(\sum_{i=1}^{L}n_i(x,y)\right)^2\right\}$$

$$=\frac{1}{L^2}E\{(\sum_{i=1}^{L}n_i^{\,2}(x,y))\}+\frac{1}{L^2}E\{(\sum_{\substack{i=1\\ i\neq j}}^{L}\sum_{j=1}^{L}(n_i(x,y)n_j(x,y))\}=\frac{1}{L^2}\sum_{i=1}^{L}E\{n_i^{\,2}(x,y)\}+\frac{1}{L^2}\sum_{\substack{i=1\\ i\neq j}}^{L}\sum_{j=1}^{L}E\{n_i(x,y)n_j(x,y)\}$$

$$=\frac{1}{L^2}\sum_{i=1}^{L}\sigma^2+0=\frac{1}{L}\sigma^2$$

Therefore, we have shown that image averaging produces an image $\bar{g}(x,y)$, corrupted by noise with variance less than the variance of the noise of the original noisy images. Note that if $L\to\infty$ we have $\sigma^2_{n(x,y)}\to 0$, the resulting noise is negligible.

2.4.1 Spatial masks

Many image enhancement techniques are based on spatial operations performed on local neighbourhoods of input pixels. The image is usually convolved with a finite impulse response filter called spatial mask. The use of spatial masks on a digital image is called spatial filtering. Suppose that we have an image $f(x,y)$ of size N^2 and we define a neighbourhood around each pixel. For example let this neighbourhood to be a rectangular window of size 3×3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig.2.5 3x3 neighbourhood

If we replace each pixel by a weighted average of its neighbourhood pixels then the response of the linear mask for the pixel z_5 is $\sum_{i=1}^{9} w_i z_i$. We may repeat the same process for the whole image.

2.4.2 Lowpass and highpass spatial filtering

A 3×3 spatial mask operating on an image can produce (a) a smoothed version of the image (which contains the low frequencies) or (b) it can enhance the edges and suppress essentially the constant background information. The behaviour is basically dictated by the signs of the elements of the mask.
Let us suppose that the mask has the following form

a	b	c
d	1	e
f	g	h

Fig.2.6 3x3 spatial mask

To be able to estimate the effects of the above mask with relation to the sign of the coefficients a,b,c,d,e,f,g,h, we will consider the equivalent one dimensional mask

d	1	e

Fig.2.7 1D spatial mask

Let us suppose that the above mask is applied to a signal $x(n)$. The output of this operation will be a signal $y(n)$ as $y(n) = dx(n-1) + x(n) + ex(n+1) \Rightarrow Y(z) = dz^{-1}X(z) + X(z) + ezX(z) \Rightarrow$
$Y(z) = (dz^{-1} + 1 + ez)X(z) \Rightarrow \dfrac{Y(z)}{X(z)} = H(z) = dz^{-1} + 1 + ez$. This is the transfer function of a system that produces the above input-output relationship. In the frequency domain we have $H(e^{j\omega}) = d\exp(-j\omega) + 1 + e\exp(j\omega)$.
The values of this transfer function at frequencies $\omega = 0$ and $\omega = \pi$ are:

$$H(e^{j\omega})\Big|_{\omega=0} = d + 1 + e$$

$$H(e^{j\omega})\Big|_{\omega=\pi} = -d + 1 - e$$

If a lowpass filtering (smoothing) effect is required then the following condition must hold

$$H(e^{j\omega})\Big|_{\omega=0} \geq H(e^{j\omega})\Big|_{\omega=\pi} = d + e \geq 0$$

If a highpass filtering effect is required then

$$H(e^{j\omega})\Big|_{\omega=0} \leq H(e^{j\omega})\Big|_{\omega=\pi} = d + e \leq 0$$

The most popular masks for lowpass filtering are masks with all their coefficients positive and for highpass filtering, masks where the central pixel is positive and the surrounding pixels are negative or the other way round.

2.5 Popular techniques for lowpass spatial filtering

2.5.1 Uniform filtering

The most popular masks for lowpass filtering are masks with all their coefficients positive and equal to each other as for example the mask shown below. Moreover, they sum up to 1 in order to maintain the mean of the image.

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Fig.2.8 lowpass filtering spatial mask

2.5.2 Gaussian filtering

The two dimensional Gaussian mask has values that attempts to approximate the continuous function

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{\sigma^2}}$$

In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. The following shows a suitable integer-valued convolution kernel that approximates a Gaussian with a σ of 1.0.

$$\frac{1}{273}\times\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline\end{array}$$

Fig.2.9 5x5 Gaussian filtering mask

2.5.3 Median filtering

The median m of a set of values is the value that possesses the property that half the values in the set are less than m and half are greater than m. Median filtering is the operation that replaces each pixel by the median of the grey level in the neighbourhood of that pixel.

Median filters are non linear filters because for two sequences $x(n)$ and $y(n)$

$$\text{median}\{x(n)+y(n)\} \neq \text{median}\{x(n)\}+\text{median}\{y(n)\}$$

Median filters are useful for removing isolated lines or points (pixels) while preserving spatial resolutions. They perform very well on images containing binary (**salt and pepper**) noise but perform poorly when the noise is Gaussian. Their performance is also poor when the number of noise pixels in the window is greater than or half the number of pixels in the window (**why?**)

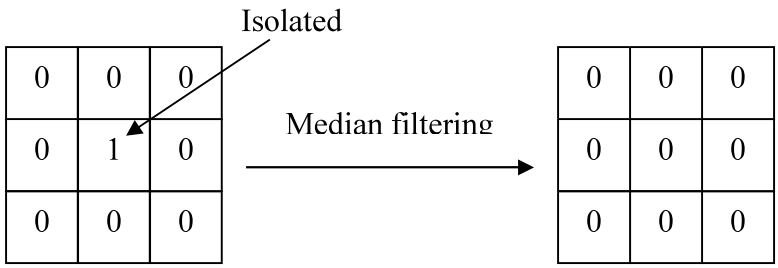


Fig.2.10 median filtering mask

2.5.4 Directional smoothing

To protect the edges from blurring while smoothing, a directional averaging filter can be useful. Spatial averages $g(x,y:\theta)$ are calculated in several selected directions (for example could be horizontal, vertical, main diagonals)

$$g(x,y:\theta)=\frac{1}{N_\theta}\sum_{(k,l)}\sum_{\in W_\theta} f(x-k,y-l)$$

and a direction θ^* is found such that $\left|f(x,y)-g(x,y:\theta^*)\right|$ is minimum. (Note that W_θ is the neighbourhood along the direction θ and N_θ is the number of pixels within this neighbourhood). Then by replacing $g(x,y:\theta)$ with $g(x,y:\theta^*)$ we get the desired result.

2.5.5 High Boost Filtering

A high pass filtered image may be computed as the difference between the original image and a lowpass filtered version of that image as follows:

(Highpass part of image)=(Original)-(Lowpass part of image)

Multiplying the original image by an amplification factor denoted by A, yields the so called high boost filter:

(Highboost image)=(A)(Original)-(Lowpass)=$(A-1)$(Original)+(Original)-(Lowpass)
=$(A-1)$(Original)+(Highpass)

The general process of subtracting a blurred image from an original as given in the first line is called ***unsharp masking***. A possible mask that implements the above procedure could be the one illustrated below.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & A-1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Fig.2.11 High Boost Filtering an unsharp masking

The high-boost filtered image looks more like the original with a degree of edge enhancement, depending on the value of A.

2.6 NOISE MODELS

- The principal sources of noise in digital images arise during image acquisition (digitization) and/or transmission.
- The performance of imaging sensors is affected by a variety of factors.
- Images are corrupted during transmission due to interference in channel

2.6.1 Spatial Properties of Noise

- With the exception of spatially periodic noise, noise is independent of spatial coordinates, and it is uncorrelated with respect to the image itself.
  We can describe that spatial noise is concerned with the statistical behavior of the gray-level values

2.6.2 Some Importance Noise

These noises are common found.

- Gaussian noise
- Rayleigh noise
- Erlang (Gamma) noise
- Exponential noise
- Uniform noise
- Impulse (salt-and-pepper) noise

2.6.2.1 Gaussian noise

The PDF of a Gaussian noise is given by

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

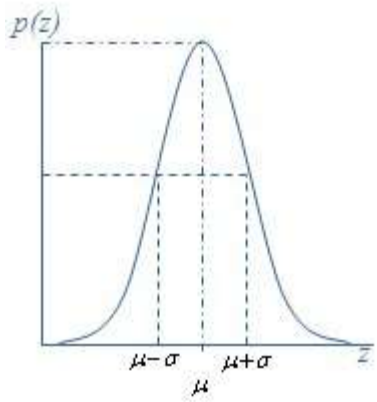


Fig.2.12 The PDF of a Gaussian noise

2.6.2.2 Rayleigh noise

The PDF of a Rayleigh noise is given by

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

The mean and variance are given

$$\mu = a + \sqrt{\pi b/4} \quad \text{and} \quad \sigma^2 = \frac{b(4-\pi)}{4}$$

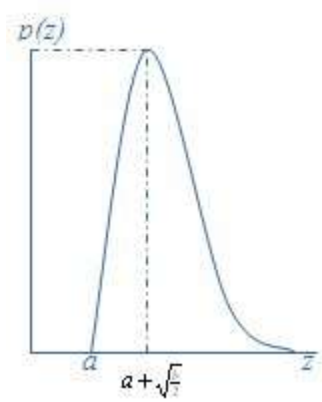


Fig.2.13 PDF of a Rayleigh noise

2.6.2.3 Erlang (Gamma) noise

The PDF of a Erlang noise is given by

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

The mean and variance are given

$$\mu = \frac{b}{a} \qquad \text{and} \qquad \sigma^2 = \frac{b}{a^2}$$

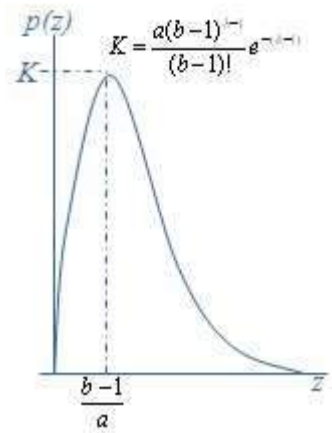


Fig.2.14 PDF of a Erlang noise

2.6.2.4 Exponential noise

The PDF of a Exponential noise is given by

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

The mean and variance are given

$$\mu = \frac{1}{a} \qquad \text{and} \qquad \sigma^2 = \frac{1}{a^2}$$

Note: It is a special case of Erlang PDF, with b=1

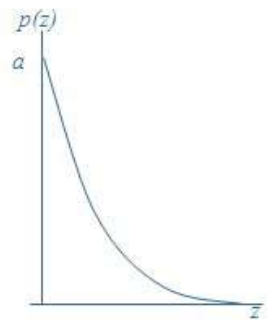


Fig.2.15 PDF of a Exponential noise

2.6.2.5 Uniform noise

The PDF of a Uniform noise is given by

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

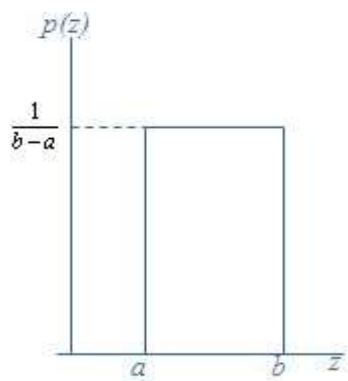


Fig.2.16 The PDF of a Uniform noise

The mean and variance are given

$$\mu = \frac{a+b}{2} \qquad \text{and} \qquad \sigma^2 = \frac{(b-a)^2}{12}$$

2.6.2.6 Impulse (salt-and-pepper) noise

The PDF of a (bipolar) impulse noise is given by

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

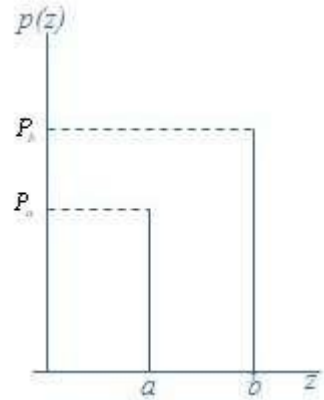


Fig.2.17 The PDF of a (bipolar) impulse noise

2.7 MEAN FILTERS

This is the simply methods to reduce noise in spatial domain.

- Arithmetic mean filter
- Geometric mean filter
- Harmonic mean filter
- Contraharmonic mean filter

Let S_{xy} represent the set of coordinates in a rectangular subimage window of size mxn, centered at point (x,y).

2.7.1 Arithmetic mean filter

Compute the average value of the corrupted image g(x,y) in the area defined by $S_{x,y}$. The value of the restored image at any point (x,y)

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

Using a convolution mask in which all coefficients have value 1/mn. Noise is reduced as a result of blurring.

2.7.2 Geometric mean filter

Compute the average value of the corrupted image g(x,y) in the area defined by $S_{x,y}$. The value of the restored image at any point (x,y), Using a geometric mean filter is given by the expression

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

2.7.3 Harmonic mean filter

Compute the average value of the corrupted image g(x,y) in the area defined by $S_{x,y}$. The value of the restored image at any point (x,y)

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

2.7.4 Contraharmonic mean filter

Compute the average value of the corrupted image g(x,y) in the area defined by $S_{x,y}$. The value of the restored image at any point (x,y) using the contraharmonic mean filter operation is given by the expression

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

Where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise.

2.8 ORDER-STATISTICS FILTERS

Order-Statistics filters are spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter

- ✓ Median filter
- ✓ Max and Min filter
- ✓ Midpoint filter
- ✓ Alpha-trimmed mean filter

Median filter

Process is replaces the value of a pixel by the median of the gray levels in region S_{xy} of that pixel:

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

Max and Min filter

Using the 100th percentile results in the so-called max filter, given by

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

This filter is useful for finding the brightest points in an image. Since pepper noise has very low values, it is reduced by this filter as a result of the max selection processing the subimage area S_{xy}.

The 0th percentile filter is min filter:

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

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

Midpoint filter

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

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

This filter works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-trimmed mean filter

Suppose that we delete the d/2 lowest and the d/2 highest gray-level values of g(s,t) in the area S_{xy}. Let $g_r(s,t)$ represent the remaining mn-d pixels. And averaging these remain pixels is denoted as:

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

Where the value of d can range from 0 to mn-1. When d=0, It is arithmetic mean filter and d=(mn-1)/2 is a median filter. It is useful for the multiple types of noise such as the combination of salt-and-pepper and Gaussian noise.

2.9 HOMOMORPHIC FILTERING

Many times, we want to remove shading effects from an image (i.e., due to uneven illumination)

- Enhance high frequencies (contrast)
- Attenuate low frequencies but preserve fine detail.(normalize brightness)

Consider the following model of image formation:
In general, the illumination component i(x,y) varies **slowly** and affects **low** frequencies mostly.

In general, the reflection component r(x,y) varies **faster** and affects **high** frequencies mostly.

<u>IDEA:</u> separate low frequencies due to i(x,y) from high frequencies due to r(x,y)

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

Low and high frequencies from **i(x,y)** and **r(x,y)** are mixed together.

$$f(x,y) = i(x,y)\ r(x,y) \quad \text{----------}> \quad F(u,v) = I(u,v) * R(u,v)$$

It is difficult to handle low/high frequencies separately,

$$F(u,v)H(u,v) = [I(u,v) * R(u,v)]H(u,v)$$

Take the ln( ) of $f(x,y) = i(x,y)\ r(x,y)$

$$ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y))$$

(1) Take $ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y))$

(2) Apply FT $F(ln(f(x,y)))=F(ln(i(x,y)))+F(ln(r(x,y)))$

$$Z(u,v) = Illum(u,v) + Refl(u,v)$$

3) Apply H(u,v)

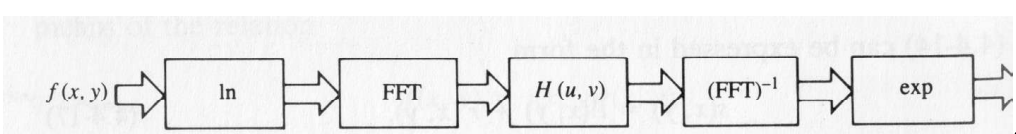
$$Z(u,v)H(u,v) = Illum(u,v)H(u,v) + Refl(u,v)H(u,v)$$

(4) Take Inverse FT $F^{-1}(Z(u,v)H(u,v))=F^{-1}(Illum(u,v)H(u,v))+F^{-1}(Refl(u,v)H(u,v))$

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

5) Take exp( ) $e^{s(x,y)} = e^{i'(x,y)}e^{r'(x,y)}$

$$g(x,y) = i_0(x,y) + r_0(x,y)$$



g(x,y)

Fig.2.18 block diagram for homomorphic filtering

Filter function:

A good deal of control is obtained over illumination and reflectance using homomorphic filtering. This control requires specification of a filter function H(u,v) that affects the low and high frequency components of the Fourier transform in different ways. If the parameters γ_H and γ_L are chosen so that $\gamma_H > 1$and $\gamma_L < 1$ the filter function tends to decrease the contribution made by low frequencies and amplify the contribution made by high frequencies. The net result is simultaneous dynamic range compression and contrast enhancement.

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[(u^2+v^2)/D_0^2\right]}\right] + \gamma_L$$

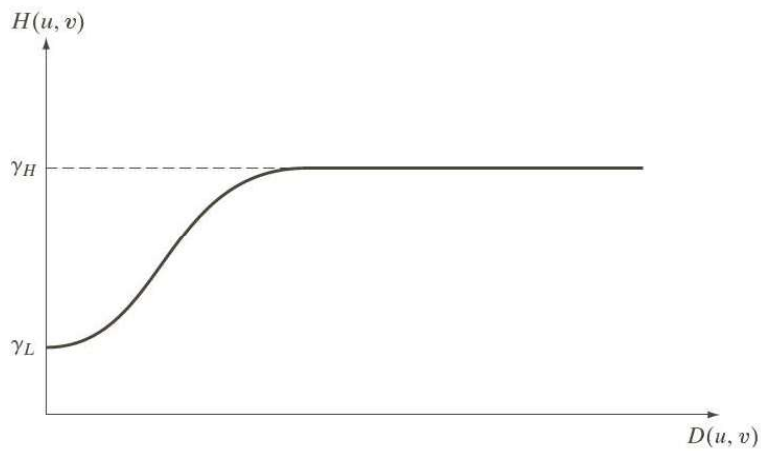


Fig2.19 Cross section of filter function

2.10 COLOR IMAGE ENHANCEMENT

In monochrome images enhancement operations are directly done on the corresponding pixel. But in color images, it has three planes, red, green and blue. If any operation is to be done on the single pixels it has to be concentrated on the three planes. Hence it is a complex process. In practical applications the color images are converted into monochrome images and any operations can be dealt.

In color image enhancement shown in fig. the original input image having R, G and B components are converted into its corresponding space coordinates, T1, T2 and T3.
These components are the individually enhanced by using any of the monochrome image enhancement techniques.

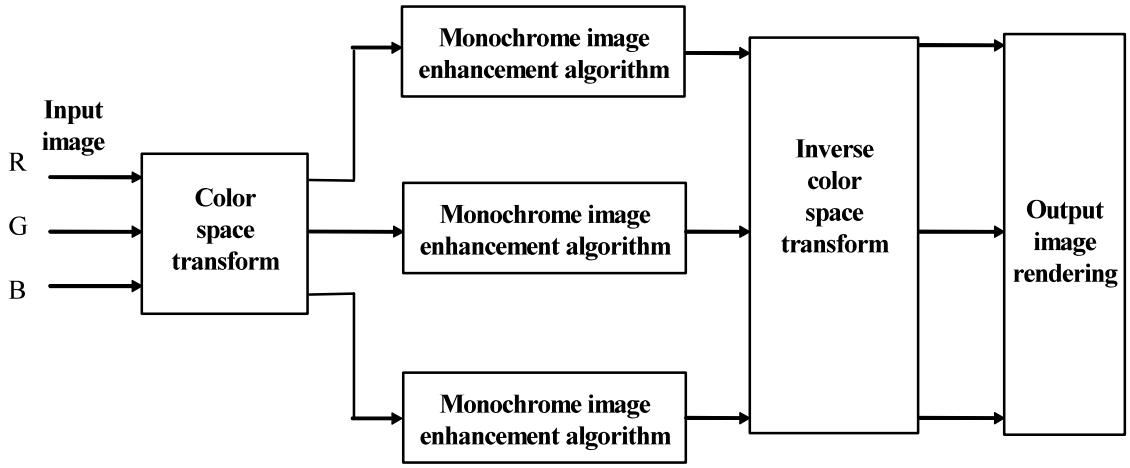


Fig.2.20 Color image enhancement

After the individual space components are enhanced they are converted into their corresponding color image coordinates to get the enhanced color image.

If smoothing(averaging) operation is to be performed then the following is used.

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

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

If sharpening is to be performed then

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

It is known from the above that all the operations are individually made on the three planes.

Summary

The material provided in this chapter is representative of spatial and frequency domain techniques used in practice for image enhancement. This chapter provides the concept of filtering with spatial masks that will be utilized in further chapters. Here we deal with various types of filters like mean filter median filter and then about different noise models. We discuss about the main type of filtering which is used in various application the homomorphic filtering. Finally we end the chapter with a brief discussion about the color image enhancement techniques.

References

The major diagrams of this chapter are brought from the book by Rafael C. Gonzalez and Richard E. Woods. The contents of this chapter are collected from Princeton University, NPTEL, IIT, New Mexican Tech.

UNIT III

IMAGE RESTORATION

Image Restoration - degradation model, Unconstrained restoration - Lagrange multiplier and Constrained restoration, Inverse filtering-removal of blur caused by uniform linear motion, Wiener filtering, Geometric transformations-spatial transformations

Preliminaries

3.1 What is image restoration?

Image Restoration refers to a class of methods that aim to remove or reduce the degradations that have occurred while the digital image was being obtained. All natural images when displayed have gone through some sort of degradation:

- during display mode
- during acquisition mode, or
- during processing mode

The degradations may be due to

- sensor noise
- blur due to camera mis focus
- relative object-camera motion
- random atmospheric turbulence
- others

In most of the existing image restoration methods we assume that the degradation process can be described using a mathematical model.

3.2 How well can we do?

Depends on how much we know about

- the original image
- the degradations

(how accurate our models are)

3.3 Image restoration and image enhancement differences

- Image restoration differs from image enhancement in that the latter is concerned more with accentuation or extraction of image features rather than restoration of degradations.
- Image restoration problems can be quantified precisely, whereas enhancement criteria are difficult to represent mathematically.

3.4 Image observation models

Typical parts of an imaging system: image formation system, a detector and a recorder. A general model for such a system could be:

$$y(i,j) = r\left[w(i,j)\right] + n(i,j)$$