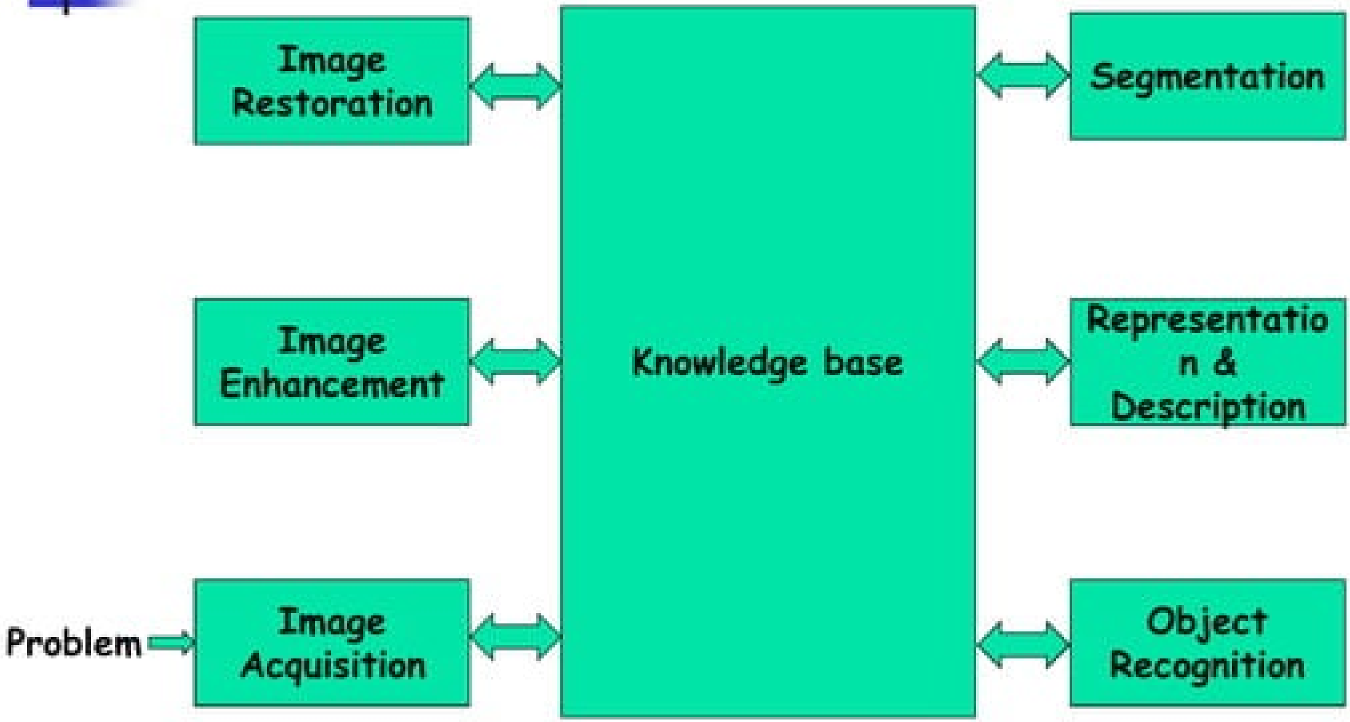
Unit 1

DIGITAL IMAGE FUNDAMENTALS AND TRANSFORMS

Elements of Digital Image Processing



Elements of Digital Image Processing

Image Acquisition: Acquiring the image of interest in digital format via imaging devices such as Charge-Coupled Devices (camera or scanner).

Image Enhancement: Bringing out the details that are obscured or simply highlighting certain features of interest in an image. Enhancement is a subjective process.

Image Restoration: Improving the quality of a degraded image based on the mathematical or probabilistic models of the degradation process. Restoration is an objective process.

Image Segmentation: Partitioning an image into its constituent parts or objects. Rugged segmentation procedures consume huge time to arrive at successful solutions to imaging problems whereas weak or erratic segmentation procedures result in total failure.

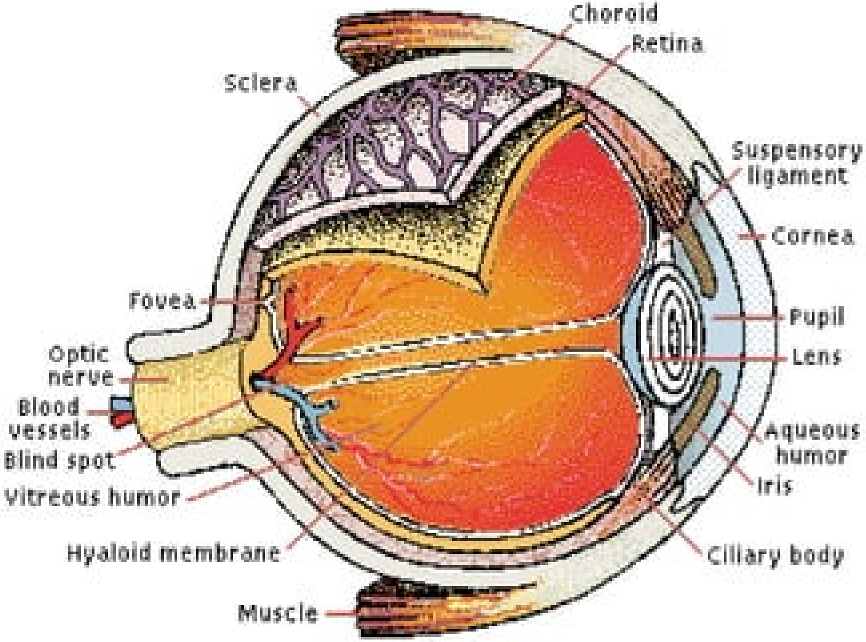
Elements of Digital Image Processing

Representation & Description: Representation - converting raw pixel data from segmentation process, normally representing boundaries of regions or all points in regions, to suitable form for computer processing. Description - extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Recognition: Assigning a label (e.g., "vehicle") to an object based on its descriptors.

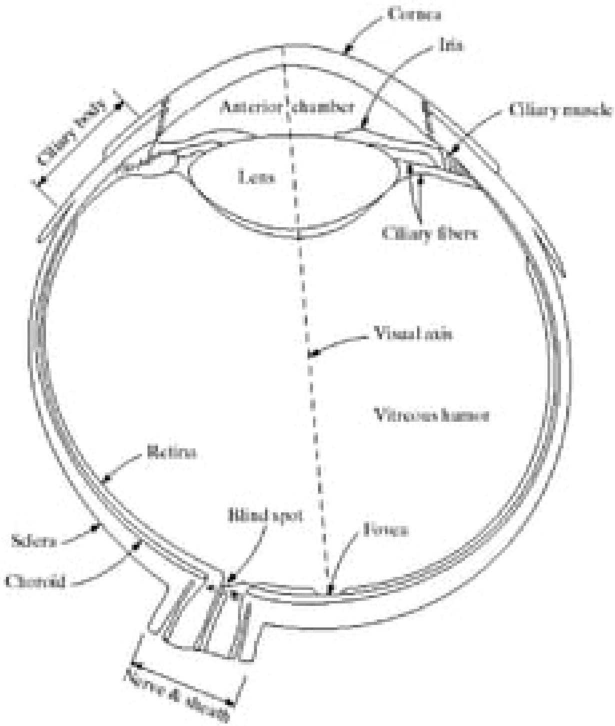
Knowledge Base: Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database. This knowledge may be simple e.g., details of regions of an image where the information of interest is known to be located, or may be quite complex, e.g., an interrelated list of all major possible defects in a materials inspection problem.

Elements of visual perception



Human Eye,
a 3D view

Elements of visual perception



Human Eye,
a 2D view

Elements of visual perception

1. A human eye, nearly a sphere with an average diameter of approximately 20 mm, is enclosed by three membranes: **cornea** and **sclera**, **choroid** and **retina**.

2. The **Cornea** is a tough & transparent tissue, covering the anterior surface of the eye.

3. The **Sclera** is an opaque membrane, enclosing the remainder of the eye globe.

4. The **Choroid** contains blood vessels to supply nutrients to the eye. It is heavily pigmented stopping external light and is divided into **ciliary body** and **iris**.

Elements of visual perception

5. Center opening of iris, known as **pupil**, is about 2-8 mm in diameter. The front of iris is filled with visible pigments and its back with black pigments.

6. The **lens**, layers of fiberous cells, is having 60% to 70% H_2O, 6% fat and rest protein. It is lightly yellowishly pigmented.

7. The **retina** is rich with **cones** and **rods** which are light receptors.

8. The **cones**, 6 to 7 millions in count are primarily located in the center of retina, known as *fovea*. They are responsible for **photopic** (*bright light*) vision-colour vision.

Elements of visual perception

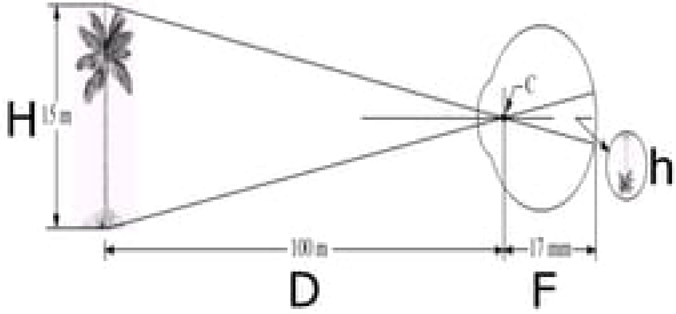
9. The **rods**, 75 to 150 millions in count, are distributed all over the retina. They are responsible for **scotopic** (*dim light*) vision-contrast.

10. An individual cone is connected to an individual optical nerve and hence accounts for perception of finer details.

11. Group of rods is connected to group of optical nerves and hence accounts for overall perception.

12. The **blind spot** in the eye is entirely deprived of the light receptors, rods and cones.

Image Formation in Human Eye



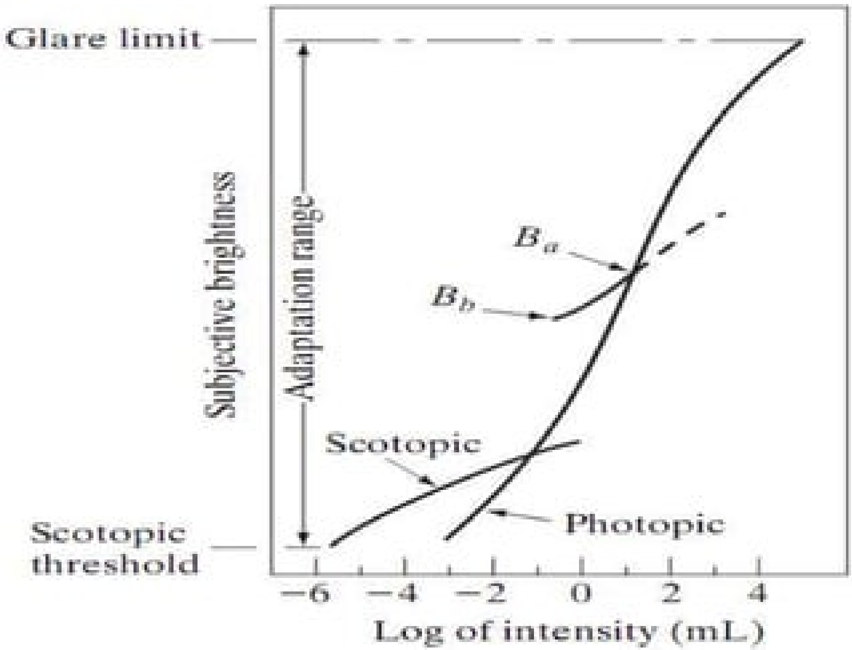
$$\frac{H}{D} = \frac{h}{F} \Rightarrow h = \left(\frac{H}{D}\right) \times F$$

The distance between the center of the lens and the retina, called the **focal length**, varies from approximately 17 mm to about 14 mm.

The height, h of an object of height, H perceived by an observer, having a focal length, F, from a distance, D is given by the principle of similar triangle.

Brightness Adaptation of Human Eye

Subjective brightness is a logarithmic function of incident light intensity.



Brightness Adaptation of Human Eye

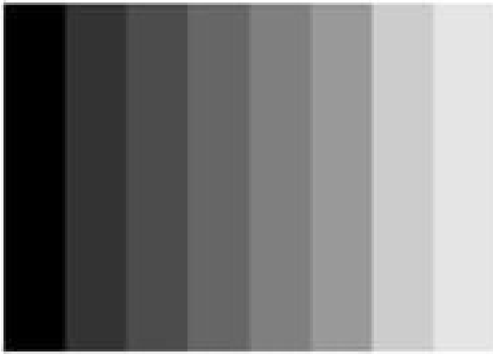
The **brightness adaptation** is a phenomenon which describes the ability of the human eye in simultaneously discriminating distinct intensity levels.

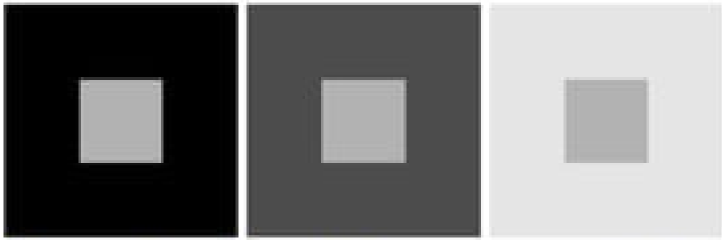
The **brightness adaptation level** is the current sensitivity level of the visual system for any given set of conditions.

The **simultaneous contrast** is a phenomenon which describes that the perceived brightness of a region in an image is not a simple function of its intensity rather it depends on the intensities of neighboring regions.

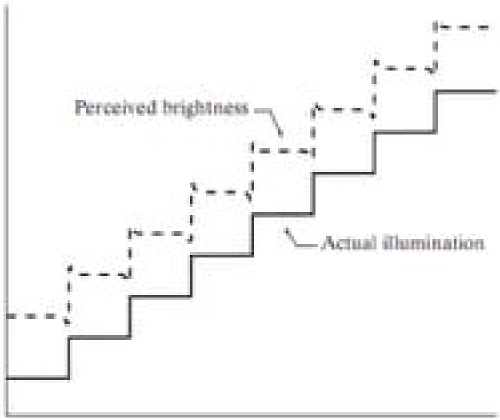
Brightness Adaptation of Human Eye

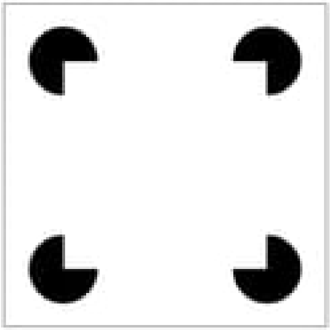
The **match bands** are the adjacently spaced rectangular stripes of constant intensities to demonstrate the phenomenon of simultaneous contrast.



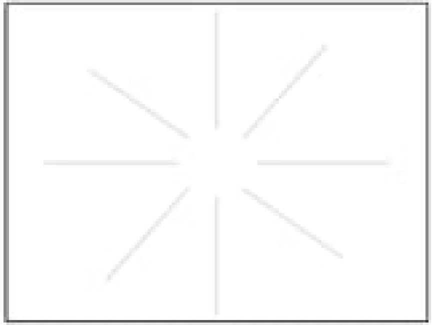


Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

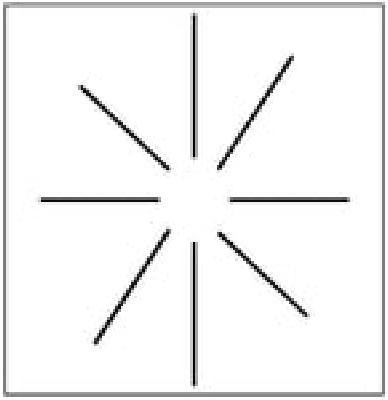




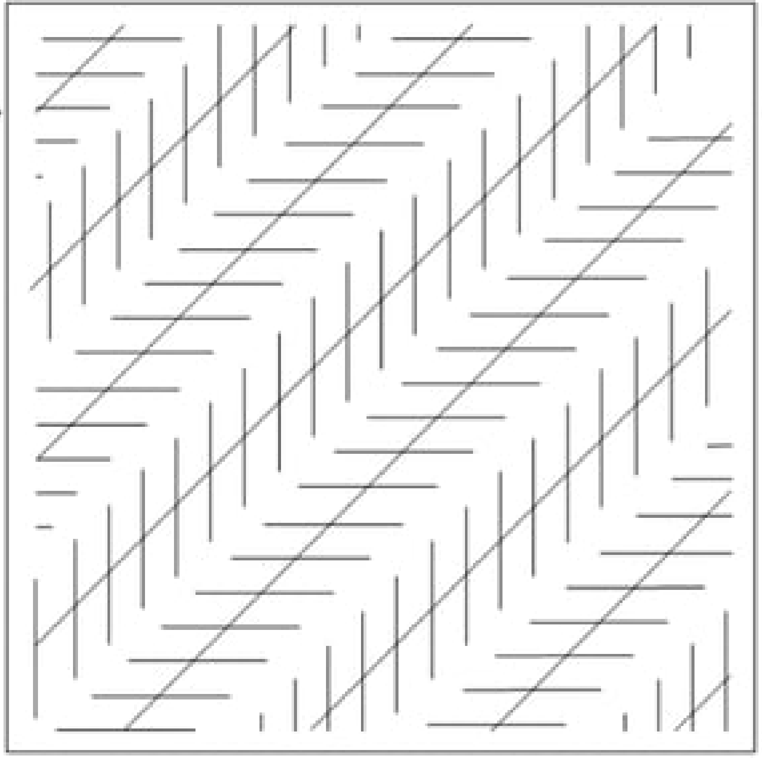
Illusion of a white square



Illusion of a white circle



Illusion of a white circle



Illusion of loss of parallelism & co-planarity

Color Fundamentals

A "color" of a light is determined by its wavelength.

Any object absorbs and reflects light energy at particular wavelengths.

The perceived color of an object is determined by the wavelength of the light reflected from it.

The object that absorbs the light energy at all wavelength "looks" black to the perceiver while the object that reflects the light energy at all wavelengths "looks" white to the perceiver.

Color Fundamentals

Achromatic light - Black and White (and their shades, gray shades).

Chromatic light - Colors (and their shades).

Three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, and brightness.

Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W).

Luminance, measured in lumens (lm), gives a measure of the amount of energy an observer perceives from a light source.

Brightness is a subjective descriptor that is practically impossible to measure.

Color Fundamentals

Cones are the sensors in the eye responsible for color vision. Approximately 65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue. Due to these absorption characteristics of the human eye, colors arc seen as variable combinations of the so-called primary colors red (R), green (G), and blue (B).

The characteristics generally used to distinguish one color from another are brightness, hue, and saturation. **Brightness** embodies the chromatic notion of intensity. **Hue** is an attribute associated with the dominant wavelength in a mixture of light waves. **Saturation** refers to the relative purity or the amount of white light mixed with a hue.

Hue and saturation taken together are called **Chromaticity**.

Color Fundamentals

The amounts of red, green, and blue needed to form any particular color are called the **tristimidus** values and are denoted, X, Y, and Z, respectively.

A color is then specified by its trichromatic coefficients, defined as

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}.$$

Color Models

A color model (also called color space or color system) is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.

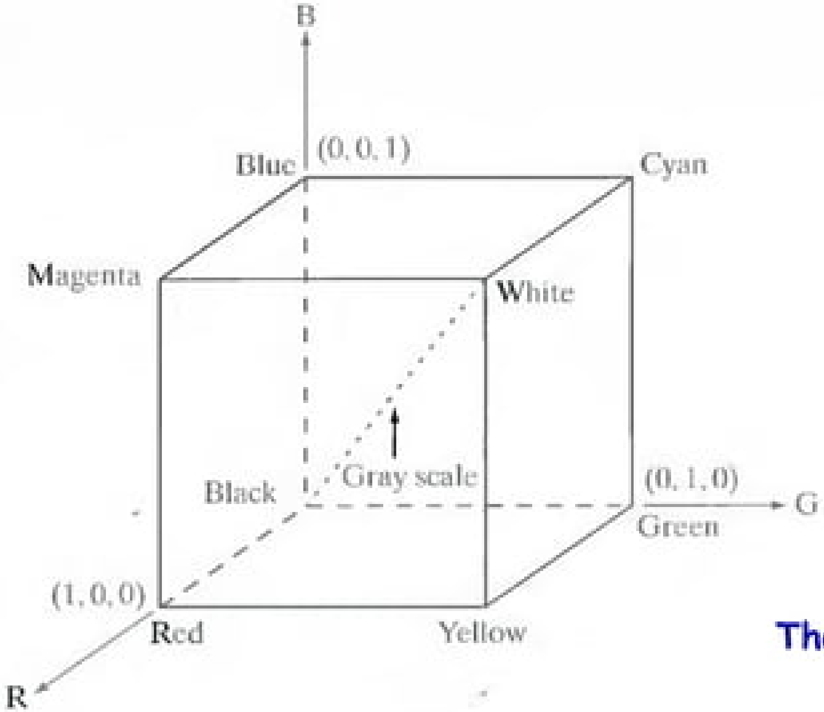
The RGB color model: In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on a **Cartesian coordinate system**. The ***color subspace*** is the **cube** in which RGB values are at three corners; cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin.

The **gray scale** (points of equal RGB values) extends from black to white along the diagonal line joining these two points.

The different **colors** are points on or inside the cube, and are defined by vectors extending from the origin.

All values of R, G, and B are assumed to be in the range [0, 1].

Color Models



The RGB color model

Color Models

Merits of RGB color model: (i) Well suited for hardware implementations and (ii) Matches nicely with the fact that the human eye is strongly perceptive to red, green, and blue primary colors.

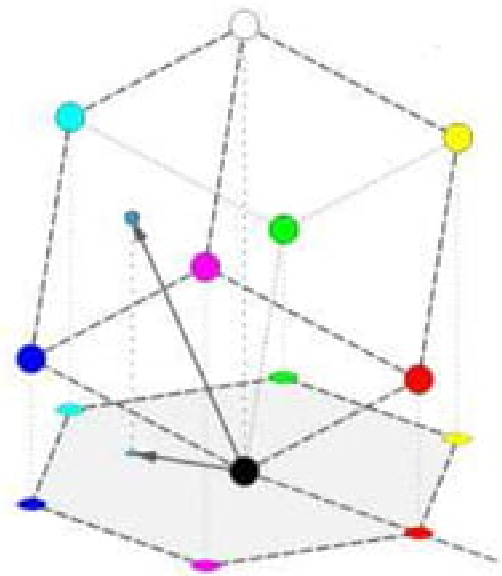
Demerits of RGB color model: Not well suited for describing colors in terms that are practical for human interpretation.

The HSI color model: A color perceived by a human eye is described by its Hue, Saturation and Intensity. **HSI** (Hue, Saturation and Intensity) color model thus decouples the intensity component from the color-carrying information (hue and saturation).

Color Models

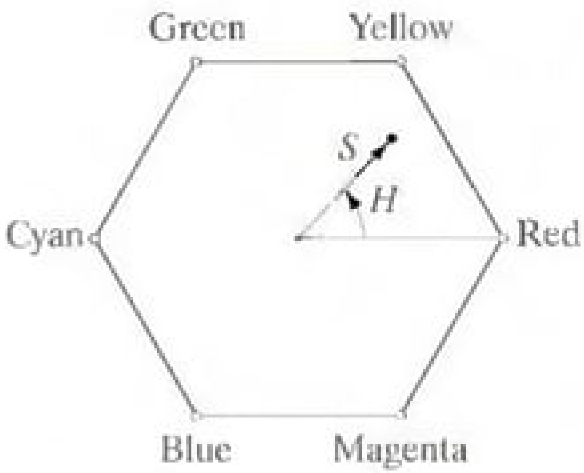
The HSI coordinate system and corresponding color subspace is obtained as follows: The RGB color cube rotated such that the cube is standing on its black vertex with the white vertex directly above and the cyan, blue, green, red, yellow and magenta vertices forming a hexagon as shown below.

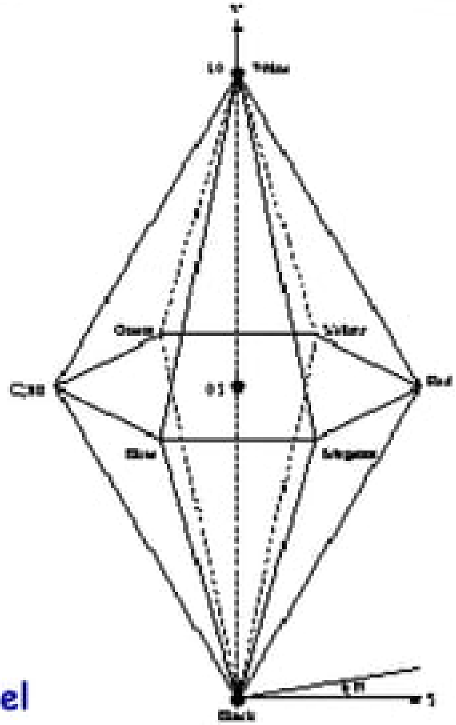
The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.



Forming the HSI color model from the RGB color model

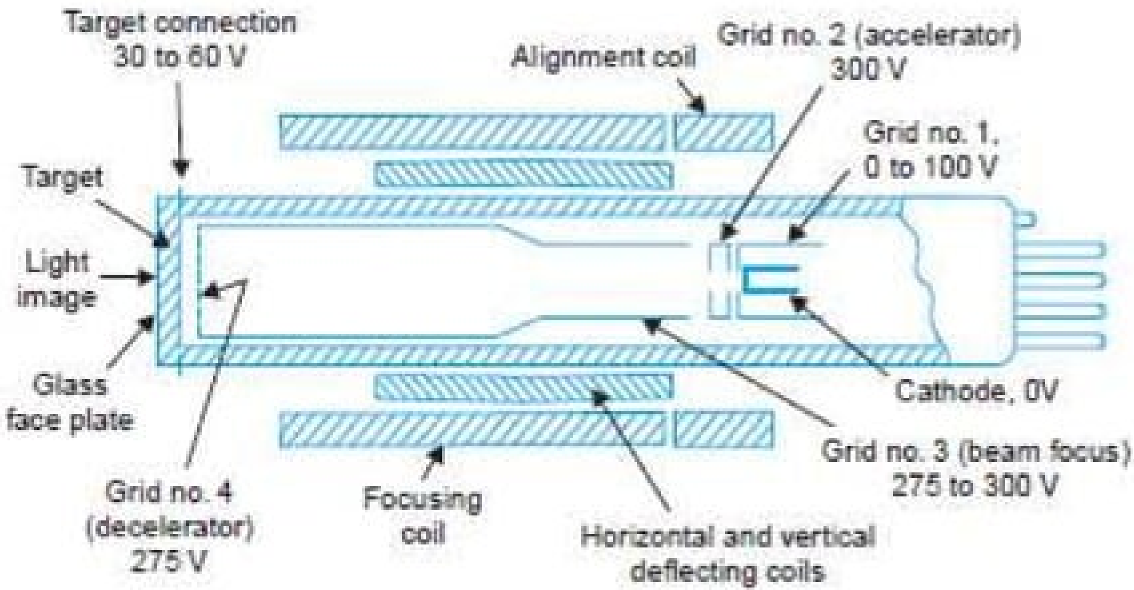
Color Models





The HSI color model

Principle of Video Camera: Vidicon



Vidicon Camera Tube – Cross Sectional View

Principle of Video Camera: Vidicon

Construction

The Vidicon came into general use in the early 50's and gained immediate popularity because of its small size and ease of operation. It functions on the principle of photoconductivity, where the resistance of the target material shows a marked decrease when exposed to light.

The target consists of a thin photo conductive layer of either selenium or anti-mony compounds. This is deposited on a transparent conducting film, coated on the inner surface of the face plate. This conductive coating is known as signal electrode or plate. Image side of the photolayer, which is in contact with the signal electrode, is connected to DC supply through the load resistance R_L.

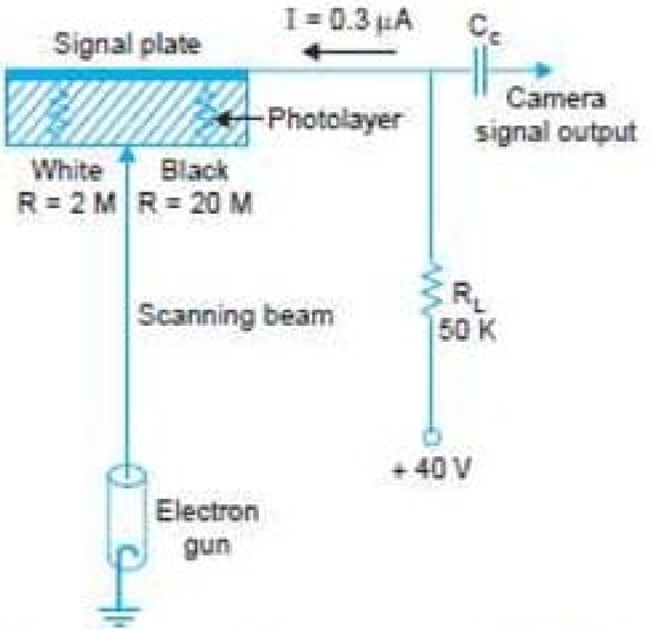
Principle of Video Camera: Vidicon

The beam that emerges from the electron gun is focused on surface of the photo conductive layer by combined action of uniform magnetic field of an external coil.

The electrostatic field of grid No 3. Grid No. 4 provides a uniform decelerating field between itself, and the photo conductive layer, so that the electron beam approaches the layer with a low velocity to prevent any secondary emission.

Deflection of the beam, for scanning the target, is obtained by vertical and horizontal deflecting coils, placed around the tube.

Principle of Video Camera: Vidicon



Circuit for output current for Vidicon Camera

Principle of Video Camera: Vidicon

Charge Image

The photolayer has a thickness of about 0.0001 cm, and behaves like an insulator with a resistance of approximately 20 MΩ when in dark.

When bright light falls on any area of the photoconductive coating, resistance across the thickness of that portion gets reduces to about 2 MΩ. Thus, with an image on the target, each point on the gun side of the photolayer assumes a certain potential with respect to the DC supply, depending on its resistance to the signal plate.

A pattern of positive potentials appears, on the gun side of the photolayer, producing a charge image, that corresponds to the incident optical image.

Principle of Video Camera: Vidicon

Another way of explaining the development of 'charge image' on the photolayer is to consider it as an array of individual target elements, each consisting of a capacitor paralleled with a light dependent resistor. One end of these target elements is connected to the signal electrode and the other end is unterminated facing the beam.

Storage Action

Each element of the photocoating is scanned at intervals equal to the frame time. This results in storage action and the net change in resistance, at any point or element on the photoconductive layer, depends on the time, which elapses between two successive scannings and the intensity of incident light. Since storage time for all points on the target plate is same, the net change in resistance of all elementary areas is proportional to light intensity variations in the scene being televised.

Principle of Video Camera: Vidicon

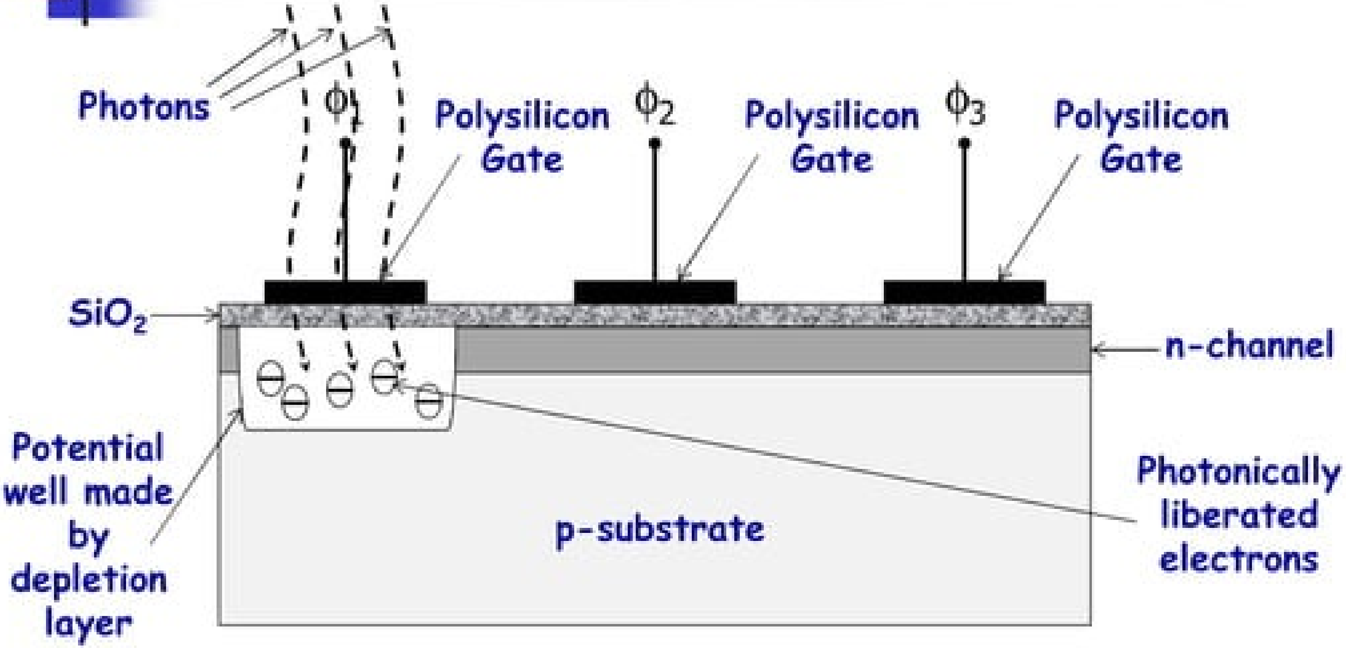
Signal Current

As the beam scans the target plate, it encounters different positive potentials on the side of the photolayer that faces the gun.

Sufficient number of electrons from the beam is then deposited on the photolayer surface to reduce the potential of each element towards the zero cathode potential. The remaining electrons, not deposited on the target, return back and are not utilized in the vidicon.

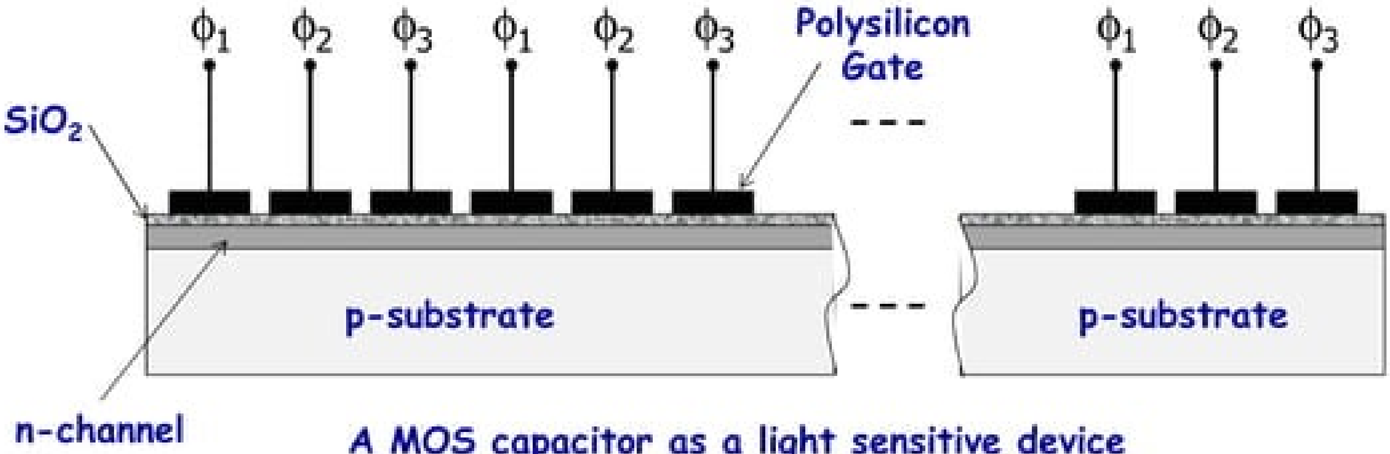
The sudden change in potential on each element while the beam scans, causes a current flow in the signal electrode circuit producing a varying voltage across the load resistance RL. The amplitude of current and the consequent output voltage across RL are directly proportional to the light intensity variations on the scene.

Principle of Still Camera: Charge Coupled Devices:



A MOS capacitor as a light sensitive device

Principle of Still Camera: Charge Coupled Devices:



A MOS capacitor as a light sensitive device

Principle of Still Camera: Charge Coupled Devices:

Charge Coupled Devices (CCD)

The operation of solid state image scanners is based on the functioning of charge coupled devices (CCDs) which is a new concept in metal-oxide-semiconductor (MOS) circuitry. The CCD may be thought of to be a shift register formed by a string of very closely spaced MOS capacitors. It can store and transfer analog charge signals—either electrons or holes—that may be introduced electrically or optically.

Construction

The chip consists of a p-type substrate, the one side of which is oxidized to form a film of silicon dioxide, which is an insulator. Then by photolithographic processes, similar to those used in miniature integrated circuits an array of metal electrodes, known as gates, are deposited on the insulator film. This results in the creation of a very large number of tiny MOS capacitors on the entire surface of the chip.

Principle of Still Camera: Charge Coupled Devices:

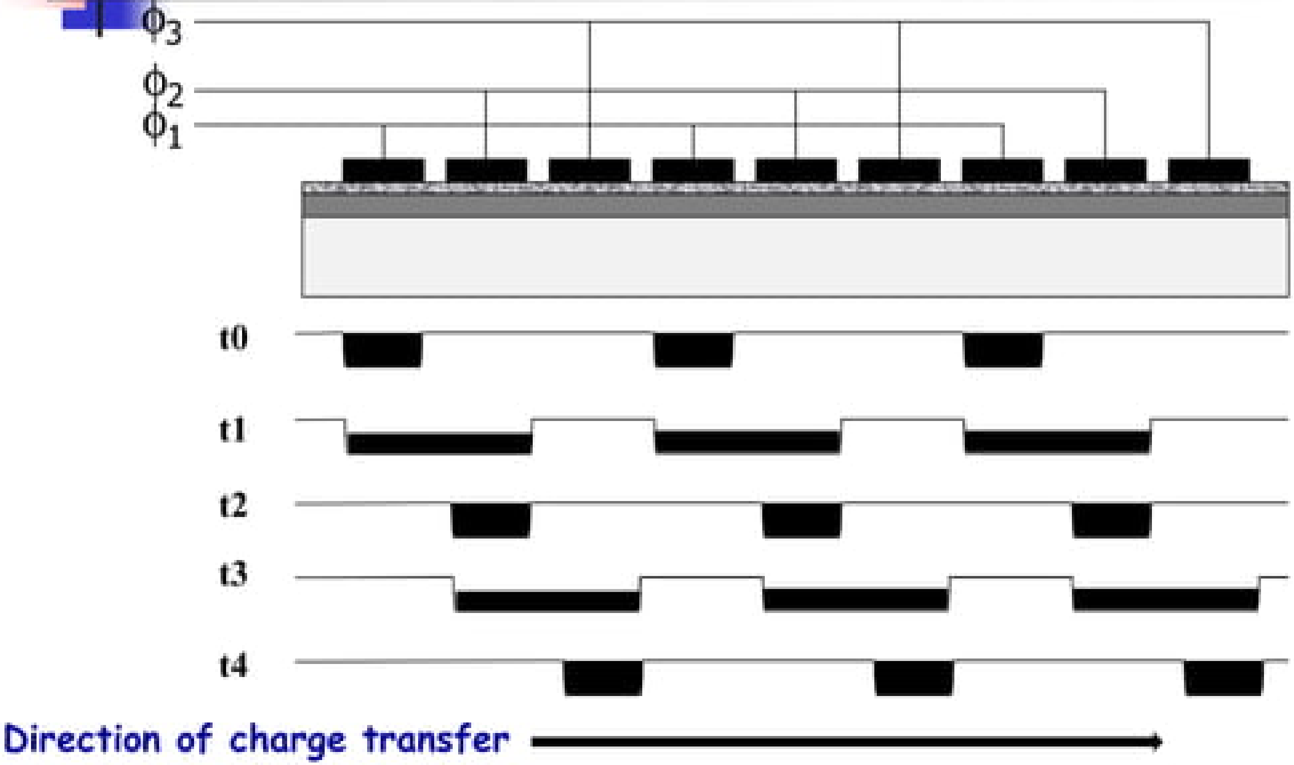
Principle of Operation

The application of small positive potentials to the gate electrodes results in the development of depletion regions just below them. These are called potential wells. The depth of each well (depletion region) varies with the magnitude of the applied potential.

The gate electrodes operate in groups of three, with every third electrode connected to a common conductor. The spots under them serve as light sensitive elements.

When any image is focused onto the silicon chip, electrons are generated within it, but very close to the surface. The number of electrons depends on the intensity of incident light. Once produced they collect in the nearby potential wells. As a result the pattern of collected charges represents the optical image.

Principle of Still Camera: Charge Coupled Devices:



Principle of Still Camera: Charge Coupled Devices:

Principle of Operation

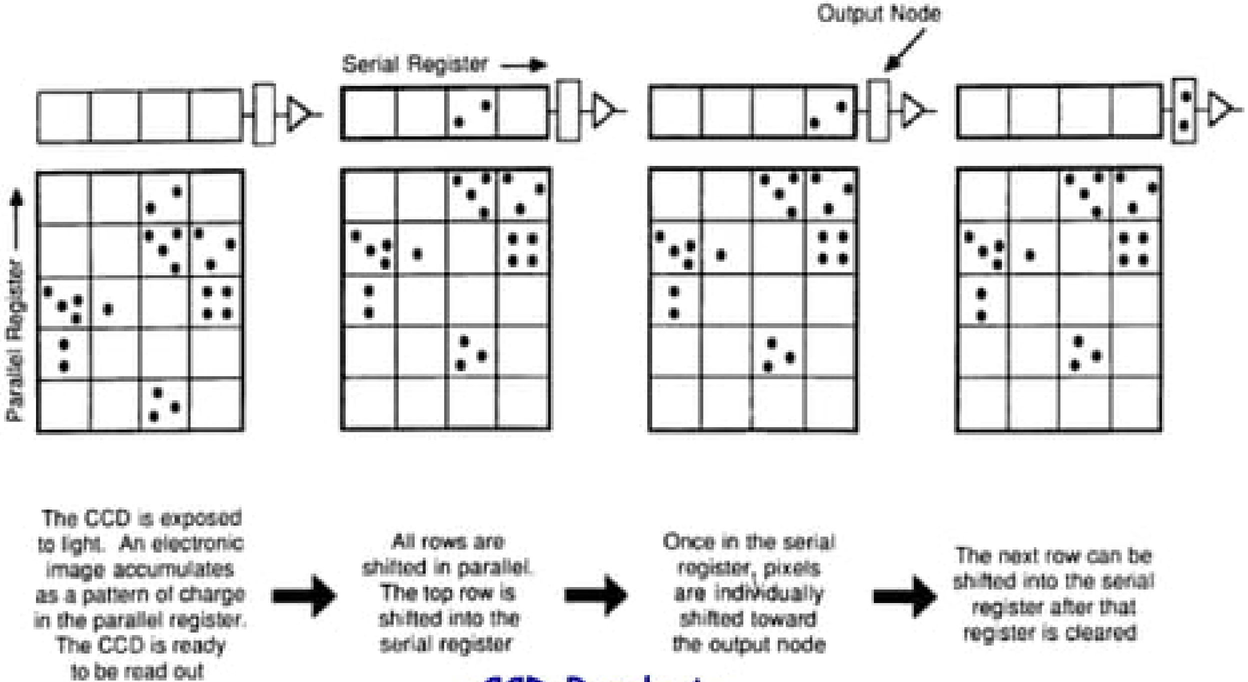
Charge Transfer

The charge of one element is transferred to another along the surface of the silicon chip by applying a more positive voltage to the adjacent electrode or gate, while reducing the voltage on it.

The manner in which the transition takes place from potential wells is illustrated in the figure. This is achieved with the influence of continuing clock pulses.

The clocking sequence continues and the charge finally reaches the end of the array where it is collected to form the signal current.

Principle of Still Camera: Charge Coupled Devices:



CCD Readout

Principle of Still Camera: Charge Coupled Devices:

Principle of Operation

CCD Readout

The two-dimensional array of potential wells is generally referred to as **parallel register**.

A one-dimensional CCD array acts as a serial register and plays an important role during the CCD readout operation.

A programmed sequence of changing gate potentials causes all charge packets stored in the parallel register to be shifted in parallel one row toward the serial register. The charge stored in the top row is shifted from the parallel register to the serial register. Once in the serial register, the charge packets are individually shifted toward the output amplifier.

(Monochrome) Image model

An (monochrome or black & white) image is a 2-D light-intensity function denoted as **f(x,y)**.

The value or amplitude, f of the function at any spatial coordinates (x,y) is the intensity of the image at that point.

As light is energy, this value is non-zero and finite i.e., $0 < f < \infty$

f(x,y) has two components: (i) **i(x,y)**, the amount of light incident on the scene being viewed and (ii) **r(x,y)**, the reflectance relating to the amount of light reflected by the objects in the scene i.e., $f(x,y) = i(x,y)\, r(x,y)$ where $0 < i < \infty$ & $0 \leq r \leq 1$

(Monochrome) Image model

For a monochrome image the intensity of the image, f at any coordinates (x,y) is termed as **gray level**, l of the image at that point, i.e.,

$L_{min} < l < L_{max} \Rightarrow 0 < l < L,$

0 → black & L → white

Intermediate values → **shades of gray** or **gray shades**

Sampling and quantization

To obtain a digital image, f(x,y) must be digitized both in space and amplitude.
-digitization of spatial coordinates - **image sampling**
-digitization of amplitude - **gray-level quantization**

The **image sampling** is viewed as partitioning an image plane into a grid with coordinates of center of each grid from an integer set $Z \times Z$.

The **(gray-level) quantization** is viewed as assigning a value from a real number set R as gray level to each grid.

Hence resulting digital image is a $M \times N$ matrix in which each matrix element represents a image element or picture element or pixel and its value represents the gray level of that pixel.

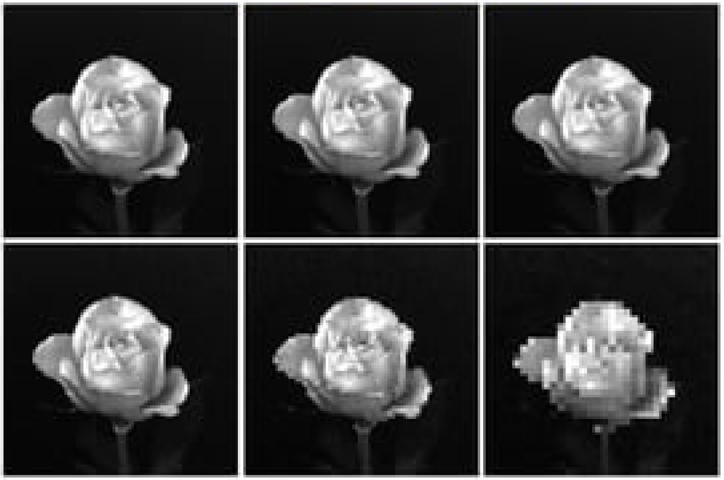
Sampling and quantization

$$\begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \bullet\bullet\bullet & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \bullet\bullet\bullet & f(1,N-1) \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \bullet\bullet\bullet & f(M-1,N-1) \end{bmatrix}$$

Sampling and quantization

The number of samples or pixels, M×N required to approximate an image is known as **spatial resolution** of the image.

The low or insufficient spatial resolution results in **pixel replication** causing a **checkerboard effect**.



Effect of spatial resolution – checkerboard effect

Sampling and quantization

The number of discrete gray levels, *G* allowed for a pixel in a digital image is known as **gray-level resolution** of the image.

The low or insufficient gray-level resolution results in **ridge-like structures** in smooth areas causing **false contouring**.

Effect of gray-level resolution – false contouring: Original 8-bit image

Effect of gray-level resolution – false contouring: Original 4-bit image

Effect of gray-level resolution – false contouring: Original 2-bit image

Effect of gray-level resolution – false contouring: Original 1-bit image, binary image

Sampling and quantization

If the quantities *M*, N and *G* are chosen to be integer powers of 2 i.e., $M=2^p$, $N=2^q$ and $G=2^r$ where p, q and r are any positive integers, then the size of the resulting digital image is $b=M \times N \times r$ bits.

Example: What is the (physical) size of an 8-bit (i.e, 256 gray-level) image of 1024×720 is $b=1024 \times 720 \times 8=5898240$ bits.
Since 8 bits are 1 byte, b=(5898240/8)=737280 bytes
Since 1024 bytes are 1 kilo byte (kB)=720 kB
(and 1024 kilo bytes are 1 mega bytes (MB))

Using different values of spatial resolution, i.e., coarse as well as fine sampling and gray-level resolution for a given image is known as **non-uniform sampling** and **quantization**.

Dithering

Dithering is a technique to simulate the display of intensities/colors that are not available in the current grayscale/color palette of the display device.

Generally a full set of intensities/colors is usually represented with a reduced number of intensities/colors.

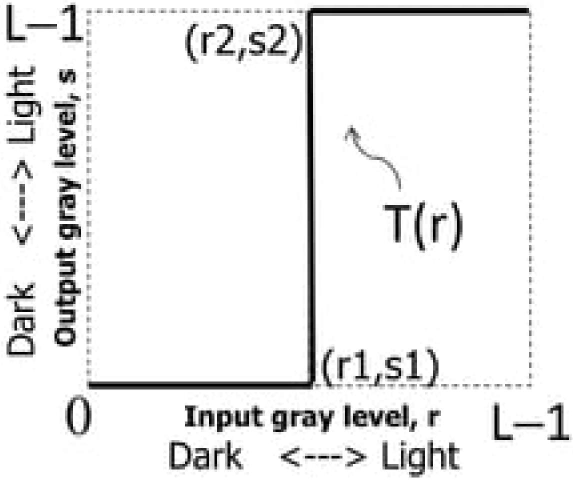
This is accomplished by arranging adjacent pixels of different intensities/colors into a pattern which simulates intensities/colors that are not available.

Dithering becomes possible because human eyes only average over an area, a property known as the **spatial integration**.

Dithering methods: Thresholding, classical half-toning, Random Dither, Patterning, Ordered Dither and Error Diffusion.

Dithering – Thresholding

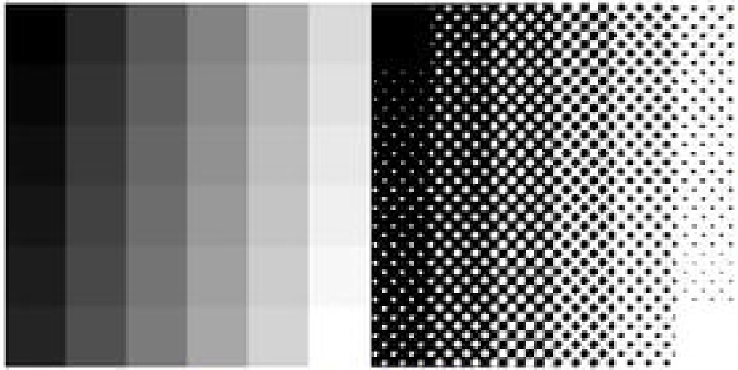
Thresholding: The threshold is chosen to be in the middle of the gray scale of the source image. The pixels in the source image darker than this threshold value are replaced with black and those lighter than it with white.



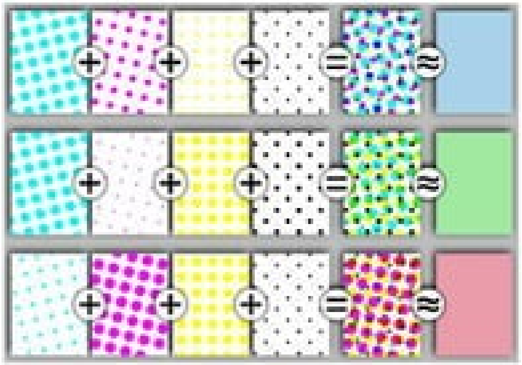
Thresholding: Function & Example

Dithering – Classical Half-toning

Classical Half-toning: Different intensities or gray levels are represented by dots of varying sizes and patterns. Half-toning is also used for printing color pictures. The general idea is the same, by varying the density of the four secondary printing colors, cyan, magenta, yellow and black (abbreviation CMYK), any particular shade can be reproduced.



Grayscale Half-toning



Color Half-toning

Dithering

Random dither: A random amount of noise is added to source image and threshold is applied.

Patterning: For each possible pixel (or group of pixels) in source image, a pattern of pixels that approximates that value is created and displayed. Remembering the concept of spatial integration, if appropriate patterns are chosen the appearance of various intensity levels can be simulated.

Ordered dither: In ordered dither, patterning is achieved with one-to-one mapping between pixels in source image and pattern pixels. This eliminates spatial distortion due to spatial enlargement and subsequent loss of spatial resolution in patterning technique.

Dithering

Error diffusion: For each possible pixel in source image, a closest available intensity/color is identified and the difference between the source image pixel value and the closest available intensity/color is calculated. This error is then distributed to some neighbors of this pixel before their closest available intensities/colors are identified.

Original (8 bits) | Threshold (1 bit) | Random dither (1 bit) | Ordered dither (1 bit) | Error diffusion (1 bit)

Image Transforms

- <u>**2D transforms:**</u>
- Generally a 2D forward transform is expressed as

$$T(u,v)=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)g(m,n,u,v)$$

where g(m,n,u,v) is called the **forward transform kernel** and a 2D inverse transform is expressed as

$$f(m,n)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F(u,v)h(m,n,u,v)$$

where h(m,n,u,v) is called the **inverse transform kernel**.

Image Transforms

- **Separable transforms:**

- A 2D transform is said to be separable if its forward and reverse kernels are expressed as product of two 1D kernels, each operating independently on each dimension i.e.

$$g(m,n,u,v) = g_1(m,u)g_2(n,v)$$

$$h(m,n,u,v) = h_1(m,u)h_2(n,v)$$

The principal advantage of separability is that the forward or inverse 2D transform can be obtained in two steps by successive applications of 1D transforms independently along each dimension.

Image Transforms

- **2D Discrete Fourier Transform (DFT):**
- The 2D Discrete Fourier Transform (DFT), F(u,v) of an image, f(m,n) of size M×N is defined as

$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi\left[\left(\frac{mu}{M}\right)+\left(\frac{nv}{N}\right)\right]}$$

for u=0,1,2,…,M-1 & v=0,1,2,…,N-1.

- The corresponding 2D Inverse Discrete Fourier Transform (IDFT), is defined as

$$f(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi\left[\left(\frac{mu}{M}\right)+\left(\frac{nv}{N}\right)\right]}$$

Image Transforms

- **2D DFT kernels:**
- The forward kernel is

$$g(m,n,u,v)=\left(\frac{1}{N}\right)e^{-j2\pi\left(\frac{mu+nv}{N}\right)}=\left(\frac{1}{\sqrt{N}}\right)e^{-j2\pi\left(\frac{mu}{N}\right)}\left(\frac{1}{\sqrt{N}}\right)e^{-j2\pi\left(\frac{nv}{N}\right)}$$

- The inverse kernel is

$$h(m,n,u,v)=\left(\frac{1}{N}\right)e^{j2\pi\left(\frac{mu+nv}{N}\right)}=\left(\frac{1}{\sqrt{N}}\right)e^{j2\pi\left(\frac{mu}{N}\right)}\left(\frac{1}{\sqrt{N}}\right)e^{j2\pi\left(\frac{nv}{N}\right)}$$

This is for the case where M=N.

Image Transforms

- **Fast Fourier Transform (FFT):**
- Due to the property of separability of 2D DFT, the FFT algorithm developed for 1D DFT is applied without any modification for 2D DFT twice successively along each dimension.

$$F(u,v)=\sum_{m=0}^{N-1}\left[\left(\frac{1}{N}\right)\sum_{n=0}^{N-1} f(m,n)e^{-j\frac{2\pi nv}{N}}\right]e^{-j\frac{2\pi mu}{N}}$$

$$F(u,v)=\sum_{m=0}^{N-1} F(m,v)\,e^{-j\frac{2\pi mu}{N}}$$

Image Transforms

- **Fast Fourier Transform (FFT):**

$$f(m,n) \xrightarrow[\text{Multiplication by } N]{\text{Row Transform}} F(m,v) \xrightarrow{\text{Column Transform}} F(u,v)$$

Image Transforms

- Other separable 2D transforms:
- 2D Discrete Cosine Transform (DCT):
- The 2D **forward Discrete Cosine Transform** (DCT) is defined as

$$F(u,v) = \alpha \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mathrm{f(m,n)} \cos\left[\frac{(2\mathrm{m}+1)\pi \mathrm{u}}{2\mathrm{N}}\right] \cos\left[\frac{(2\mathrm{n}+1)\pi \mathrm{v}}{2\mathrm{N}}\right]$$

and the 2D **inverse Discrete Cosine Transform** (IDCT) is defined as

$$f(m,n) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mathrm{F(u,v)} \cos\left[\frac{(2\mathrm{m}+1)\pi \mathrm{u}}{2\mathrm{N}}\right] \cos\left[\frac{(2\mathrm{n}+1)\pi \mathrm{v}}{2\mathrm{N}}\right]$$

where $\alpha = \sqrt{1/N} \;\; for \; u,v = 0 \;\&\; \sqrt{2/N} \;\; for \; u,v = 1,2,\ldots,N-1$

Image Transforms

- **Other separable 2D transforms:**
- **Karhunen Lowe (Hotelling) transform (Principal Component Analysis):**
- Let $x=[x_1\ x_2\ \dots\ x_n]^T$ be a population of random vectors x_i, i=1,2,…n. Then
- Let m_x be the mean vector of x, defined as
  $m_x=E\{x\}$
- Let C_x be the covariance matrix of x, defined as
  $C_x=E\{(x-m_x)\ (x-m_x)^T\}$
- Let A be a matrix whose first row is the eigenvector corresponding to the largest eignvalue of C_x and the last row is that corresponding to the smallest eignvalue of C_x.

Image Transforms

- <u>Other separable 2D transforms:</u>
- <u>Karhunen Lowe (Hotelling) transform (Principal Component Analysis):</u>
- Then the Karhunen Lowe (KL) or Hotelling transform of x is the matrix given by

  $y=A(x-m_x)$
- Mean of y is zero i.e., $m_y=0$.
- Covariance matrix C_y of y is a diagonal matrix given by

$$C_y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

where λ_i, i=1,2,...n are the eigenvalues of C_x.

Image Transforms

- <u>Other separable 2D transforms:</u>
- <u>Karhunen Lowe (Hotelling) transform (Principal Component Analysis):</u>
- Hence the components of y vectors are uncorrelated.
- λ_i, i=1,2,...n are the eigenvalues of C_y as well. Hence the eigenvectors of C_y are also same as those of C_x.
- Hence KL or Hotelling transform is useful for separating the principal components from a set of independent observations (images) of an object or a scene.

Image Transforms

- <u>Other separable 2D transforms:</u>
- <u>Singular Value Decomposition (SVD):</u>
- Any rectangular matrix, **A** of size, $m \times n$ can be expressed as

  $\mathbf{A}=\mathbf{U}\mathbf{S}\mathbf{V}^T$

  where (1) **U** is an orthogonal square matrix of size, $m \times m$ i.e., $UU^T=U^TU=I$. The columns of **U** are eigenvectors of $\mathbf{A}\mathbf{A}^T$. (2) V is an orthogonal square matrix of size, $n \times n$ i.e., $VV^T=V^TV=I$. The columns of **V** are eigenvectors of $\mathbf{A}^T\mathbf{A}$. (3) **S** is a diagonal matrix of size, $m \times n$, i.e., $s_{ij}=0$ if $i \neq j$, with the diagonal elements equal, i.e., s_{ij}, $i=j$, to the square roots of eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$.

Image Transforms

- <u>Some important features of image transforms studied:</u>
- <u>Energy Conservation & Rotation: Parseval's theorem:</u>

  The unitary transforms preserves signal energy or equivalently the length of the signal. This means that the unitary transform simply rotates the signal vector in the N-dimensional space.
- <u>Energy Compaction:</u>

  Most unitary transforms has the tendency to pack a large fraction of the signal energy into a relatively few components of the transform coefficients. The following transforms are having energy compaction in the given order - DCT, [DFT, Slant], Hadamard, KL, Haar.

Image Transforms

- **Some important features of image transforms studied:**
- **Decorrelation:**

  When the input signal is highly correlated, the transform coefficients tend to be uncorrelated. This means that the off-diagonal elements of the covariance matrix of the signal are smaller than the diagonal elements.

Unit 2

IMAGE ENHANCEMENT
TECHNIQUES

Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- The suitableness is up to each application.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another image.

Broad Classes of Image Enhancement Techniques

- Spatial Domain: (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain:
  - Techniques are based on modifying the Fourier transform of an image
- There are some enhancement techniques based on various combinations of methods from these two categories.

Good images

- For human visual
  - The visual evaluation of image quality is a highly subjective process.
  - It is hard to standardize the definition of a good image.
- A certain amount of trial and error usually is required before a particular image enhancement approach is selected.

Histogram Processing

- Histogram of a digital image with gray levels in the range [0,L-1] is a discrete function

$$h(r_k) = n_k$$

- Where
  - r_k : the k^{th} gray level
  - n_k : the number of pixels in the image having gray level r_k
  - $h(r_k)$: histogram of a digital image with gray levels r_k

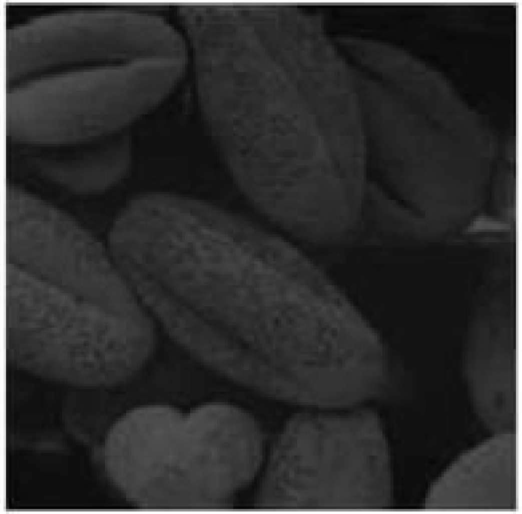
Normalized Histogram

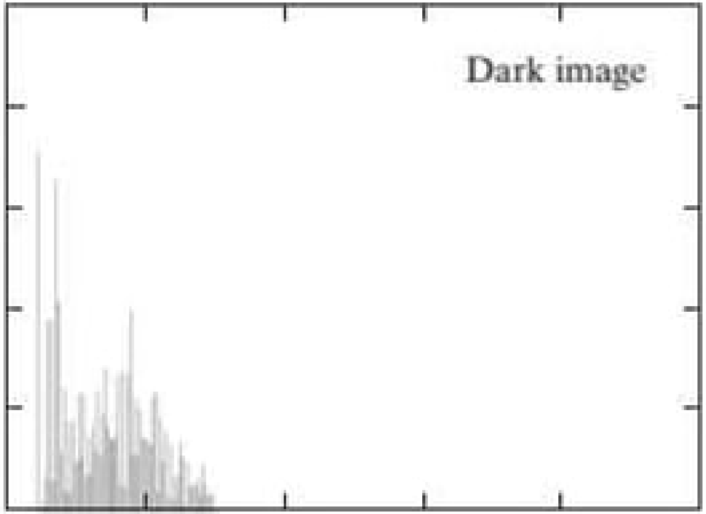
- dividing each of histogram at gray level r_k by the total number of pixels in the image, n

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

- For k = 0,1,…,L-1
- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1

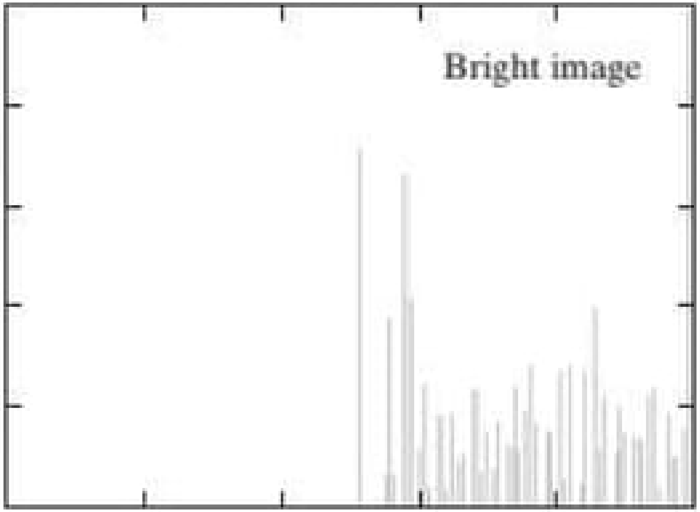
Examples of Histogram





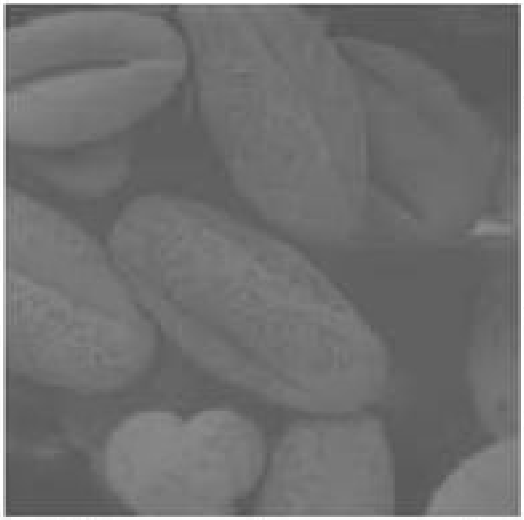
Components of histogram are concentrated on the low side of the gray scale

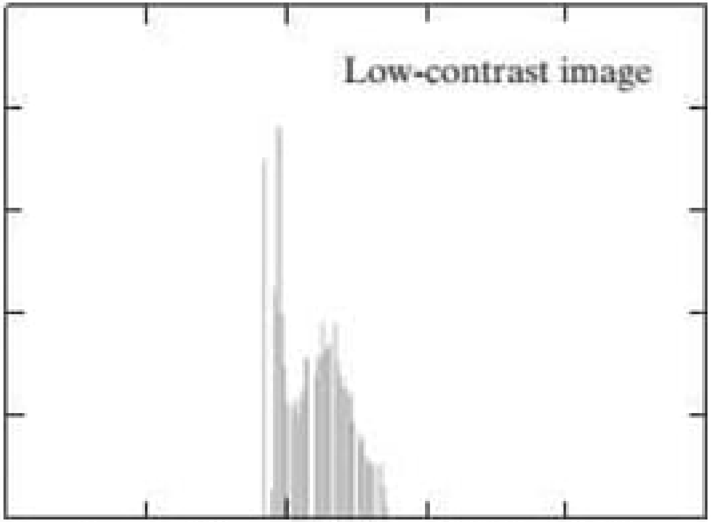
Examples of Histogram



Components of histogram are concentrated on the high side of the gray scale

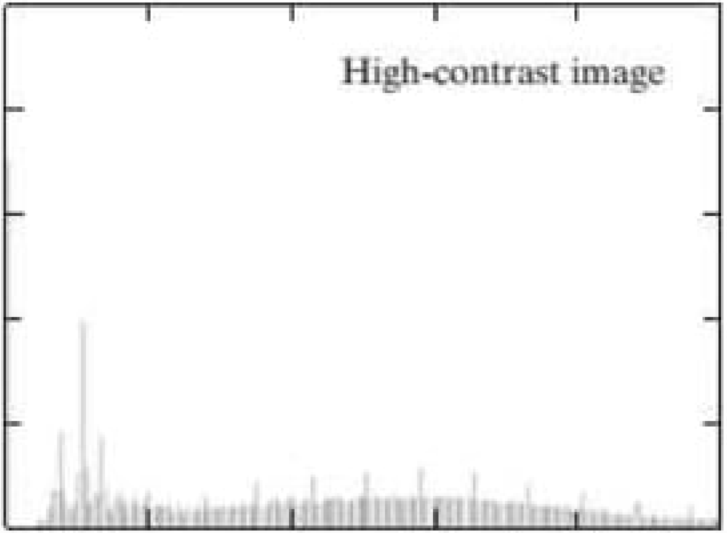
Examples of Histogram





Histogram is narrow and concentrated toward the middle of the gray scale

Examples of Histogram



Histogram covers wide range of the gray scale and the distribution is nearly uniform over the entire gray scale except at few points near the dark region of the gray scale

Histogram Equalization

- Let r represent the input gray levels in the interval [0,1] where r=0 represents black and r=1 represents white. The transformation

$$s=T(r)$$

produces a gray level, *s* in the output image for every gray level, r in the original (input) image. This transformation is to satisfy the following conditions:

a) T(r) is single-valued, monotonically increasing in the interval $0 \le r \le 1$

b) (b) $0 \le T(r) \le 1$ for $0 \le r \le 1$

Condition (a) preserves the order when r varies from black to white and (b) guarantees a mapping that is consistent with the allowed range of pixel values.

Histogram Equalization

- Single-valued function, $T(r)$ guarantees that there exists an inverse transformation

$$r = T^{-1}(s)$$

that satisfies the same set of conditions (a) and (b).

- If $p_r(r)$ represents the probability density function (PDF) of the random variable, r and $p_s(s)$ represents the probability density function (PDF) of the random variable, s, then from the basic probability theory,

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{|r=T^{-1}(s)}$$

Histogram Equalization

- Histogram equalization is to control the PDF of gray levels of an image via a transformation function so that the resulting PDF is a uniform density. This is achieved by taking the cumulative distribution function (CDF) of r as the required transformation function, T(r) i.e.,

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

where w is the dummy variable of integration.

Histogram Equalization

- With this transformation function, the PDF, $p_s(s)$ of s becomes

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

$$= \frac{d}{dr}\left[\int_0^r p_r(w)dw\right]$$

$$= p_r(r)$$

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

$$= p_r(r)\left|\frac{1}{p_r(r)}\right|$$

$$= 1 \quad \text{where } 0 \le s \le 1$$

Substitute and yield

Histogram Equalization-Discrete Form

- The probability of occurrence of gray level in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n} \quad where\ k = 0, 1, ..., L-1$$

- The discrete version of transformation

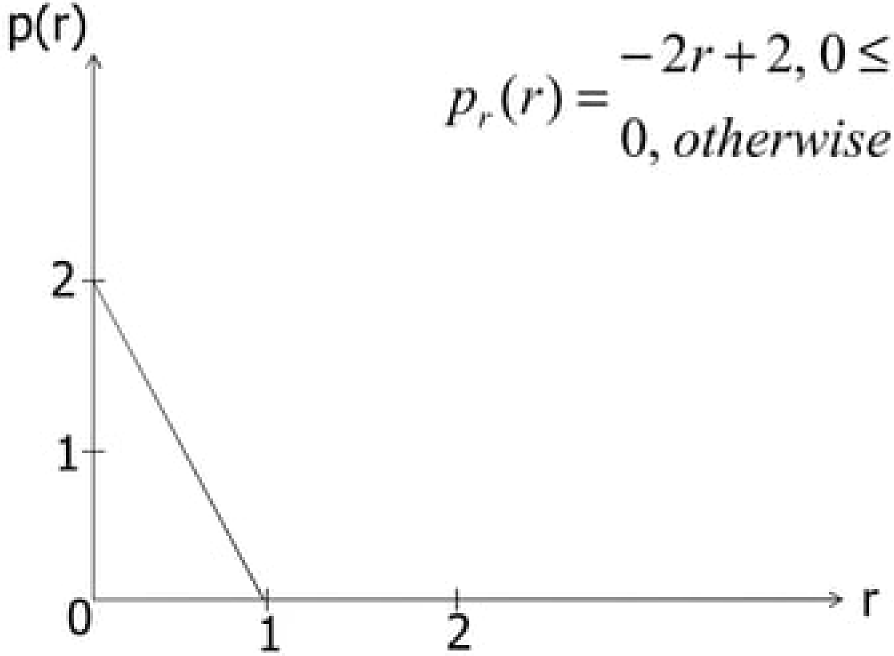
$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \quad where\ k = 0, 1, ..., L-1$$

Histogram Equalization-Discrete Form

- Thus, the histogram equalization or linearization is a method of obtaining a uniform histogram for a given image.

Histogram Equalization-Example



$$p_r(r) = \begin{cases} -2r+2, & 0 \le r \le 1 \\ 0, & otherwise \end{cases}$$

Histogram Equalization-Example

- Hence, the required transformation function is

$$s = T(r) = \int_0^r p_r(w)\, dw = \int_0^r (-2w+2)\, dw = -r^2 + 2r$$

- Solving the above equation for r, we have

$$r = T^{-1}(s) = 1 \pm \sqrt{(1-s)}$$

- Since r lies in the interval [0,1], only the function

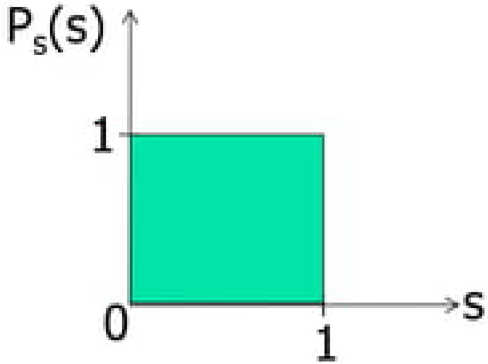
$$r = T^{-1}(s) = 1 - \sqrt{(1-s)}$$

is valid.

Histogram Equalization-Example

- Hence,

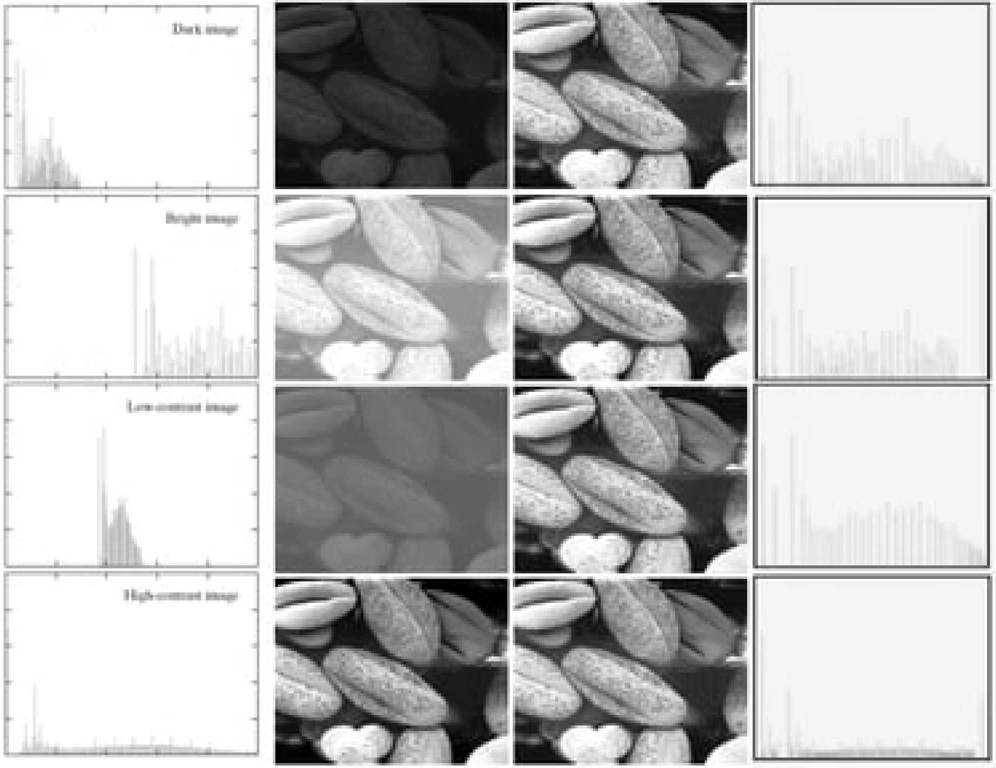
$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{|r=T^{-1}(s)=1-\sqrt{(1-s)}} = \left[(-2r+2) \frac{d}{ds}[1-\sqrt{(1-s)} \right]$$



$$= \left[(2\sqrt{(1-s)}) \left(\frac{1}{2} \right) \left[\frac{1}{\sqrt{(1-s)}} \right] \right]$$

$$= 1, \ for \ 0 \le s \le 1$$

Histogram Equalization



Histogram Specification

- Histogram specification is a method of obtaining a particular histogram shape capable of highlighting certain gray level ranges in a given image.

Histogram Specification

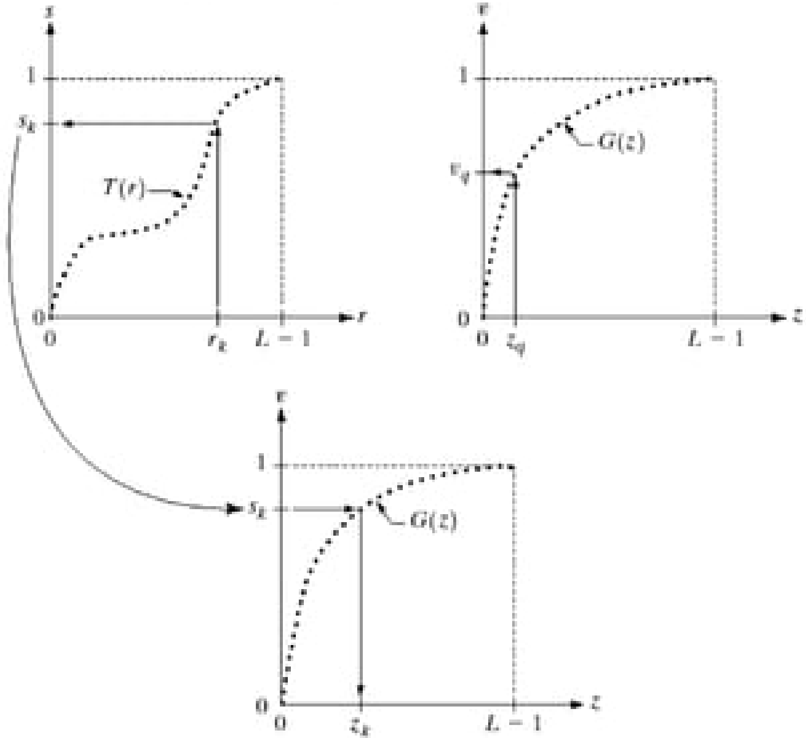
- If $p_r(r)$ and $p_s(s)$ represent the original and desired probability density functions, respectively, then the histogram specification is achieved as follows:
  1. Equalize the levels of the original image via the transformation function
  $$s = T(r) = \int_0^r p_r(w)\, dw$$
  2. Specify the desired probability density function, $p_z(z)$ and obtain the transformation function
  $$s = G(z) = \int_0^z p_z(w)\, dw$$
  3. Apply the inverse transformation $z=G^{-1}(s)$ to the levels equalized in step 1.

Histogram Specification

- The resulting image has the gray levels characterized by the specified probability density function, $p_z(z)$ i.e., has the specified histogram.

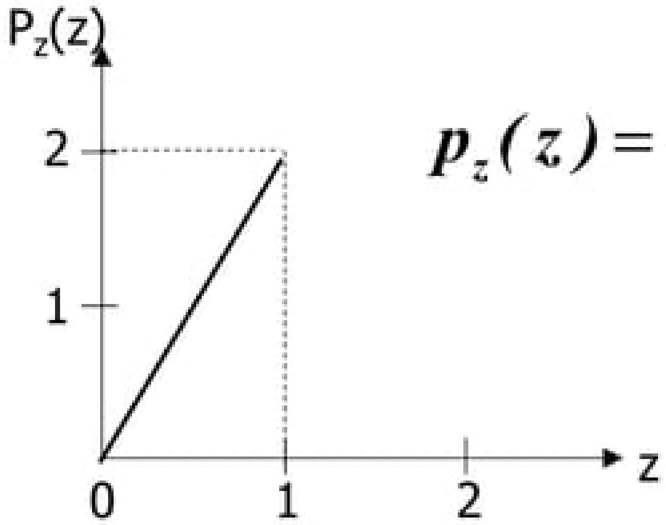
- In practice, the inverse transformation from *s* to z is not *single-valued*. This happens when there are unfilled levels in the specified histogram. These unfilled levels make the cumulative distribution function to be constant over the unfilled intervals.

Histogram Specification



Histogram Specification-Example

We would like to apply the histogram specification with the desired probability density function $p_z(z)$ as shown.

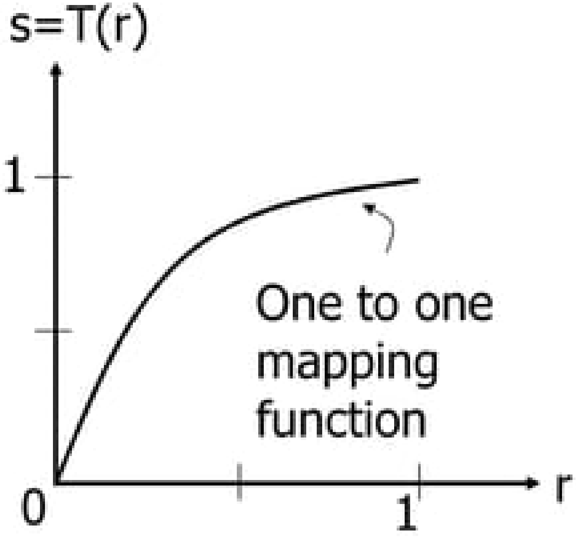


$$p_z(z) = \begin{cases} 2z & ;0 \le z \le 1 \\ 0 & ;\text{elsewhere} \end{cases}$$

$$\int_0^z p_z(w)\,dw = 1$$

Step 1

Obtain the transformation function T(r)



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

$$= \int_0^r (-2w + 2)\,dw$$

$$= -w^2 + 2w \Big|_0^r$$

$$= -r^2 + 2r$$

Step 2

Obtain the transformation function $G(z)$

$$G(z) = \int_0^z (2w)dw \quad = z^2\Big|_0^z \quad = z^2$$

Step 3

Obtain the inversed transformation function G^{-1}

$$G(z) = T(r)$$

$$z^2 = -r^2 + 2r$$

$$z = \sqrt{2r - r^2}$$

We can guarantee that $0 \le z \le 1$ when $0 \le r \le 1$

Noise Models

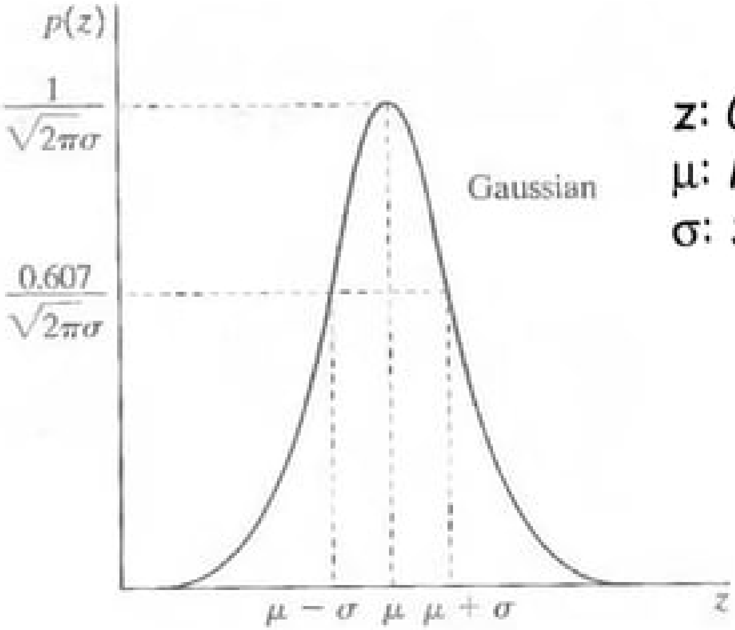
- Gaussian Noise: The Probability Density Function (PDF) of Gaussian noise is

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

  where z represents gray level, μ is the mean of average value of z, and σ is its standard deviation. The standard deviation squared, σ^2, is called the variance of z.
- Mathematically easily traceable in both spatial and frequency domains.

Noise Models

- The distribution of Gaussian noise is shown in the following figure.



z: Gray level value
μ: Mean
σ: Standard deviation

Noise Models

- Rayleigh Noise: The Probability Density Function (PDF) of Rayleigh noise is

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

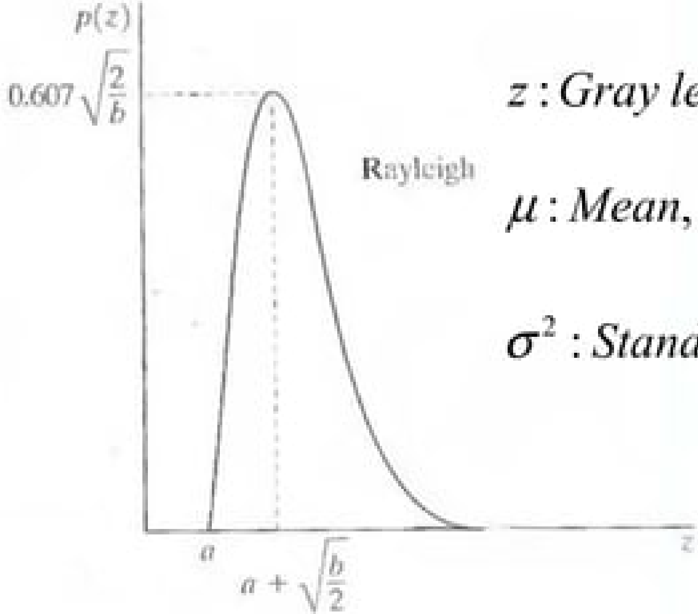
where z represents gray level and the mean and variance are given by

$$\mu = a + \sqrt{\frac{\pi b}{4}}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

Noise Models

- The distribution of Rayleigh noise is shown in the following figure.



z : *Gray level value*

μ : *Mean*, $\quad a+\sqrt{\frac{\pi b}{4}}$

σ^2 : *Standard deviation*, $\quad \frac{b(4-\pi)}{4}$

Noise Models

- Erlang (gamma) Noise: The Probability Density Function (PDF) of Eralng (gamma) noise is

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

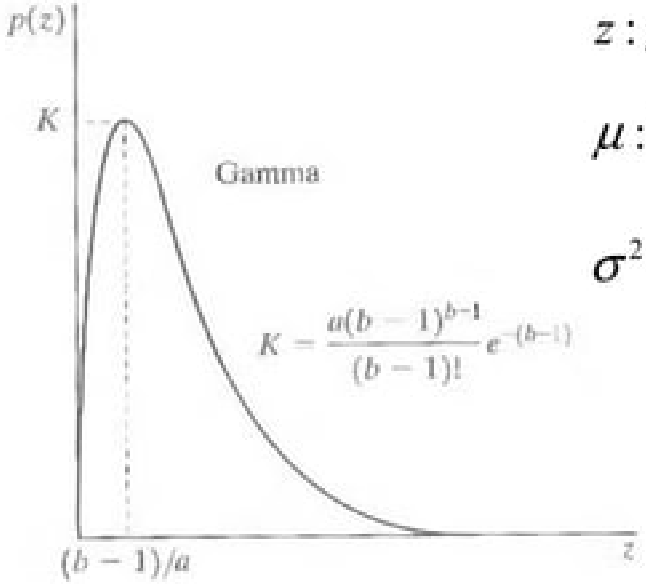
where z represents gray level and the mean and variance are given by

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}$$

Noise Models

- The distribution of Erlang (gamma) noise is shown in the following figure.



z : *gray level value*

μ : *Mean,* $\dfrac{b}{a}$

σ^2 : *Standard deviation,* $\dfrac{b}{a^2}$

Noise Models

- Exponential Noise: The Probability Density Function (PDF) of exponential noise is

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

This is Erlang with b=1.

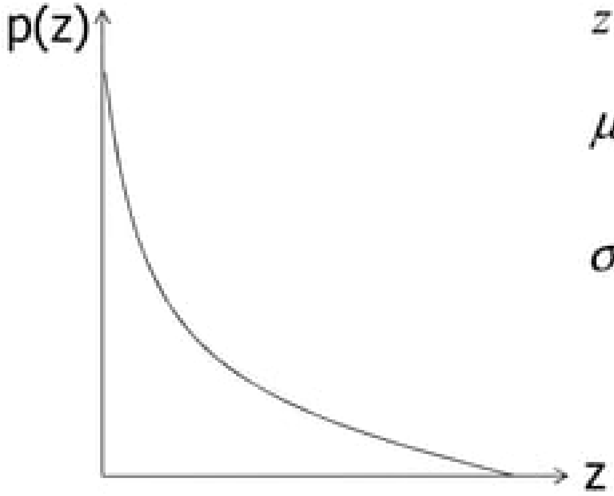
where z represents gray level and the mean and variance are given by

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}$$

Noise Models

- The distribution of exponential noise is shown in the following figure.



$z : gray\ level\ value$

$\mu : Mean,\ \ \frac{1}{a}$

$\sigma^2 : Standard\ deviation,\ \ \frac{1}{a^2}$

Noise Models

- Uniform Noise: The Probability Density Function (PDF) of uniform noise is

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

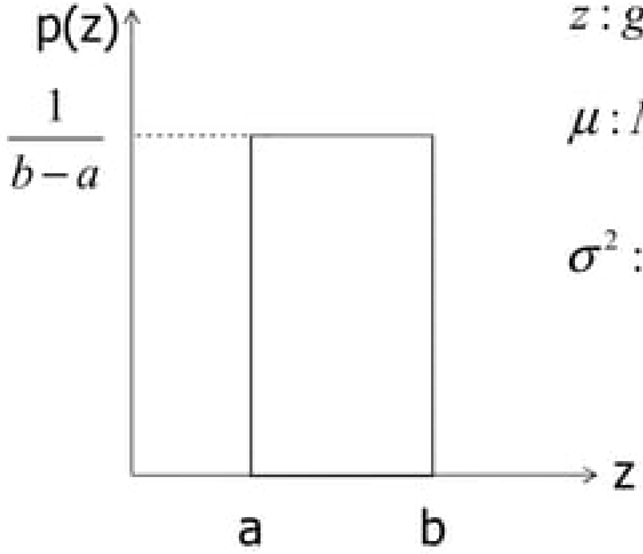
where z represents gray level and the mean and variance are given by

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

Noise Models

- The distribution of uniform noise is shown in the following figure.



z : *gray level value*

μ : *Mean*, $\dfrac{a+b}{2}$

σ^2 : *Standard deviation*, $\dfrac{(b-a)^2}{12}$

Noise Models

- Impulse (salt & pepper) Noise: The Probability Density Function (PDF) of impulse (salt & pepper) noise is

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

where z represents gray level.

Noise Models

- The distribution of impulse (salt & pepper) noise is shown in the following figure.

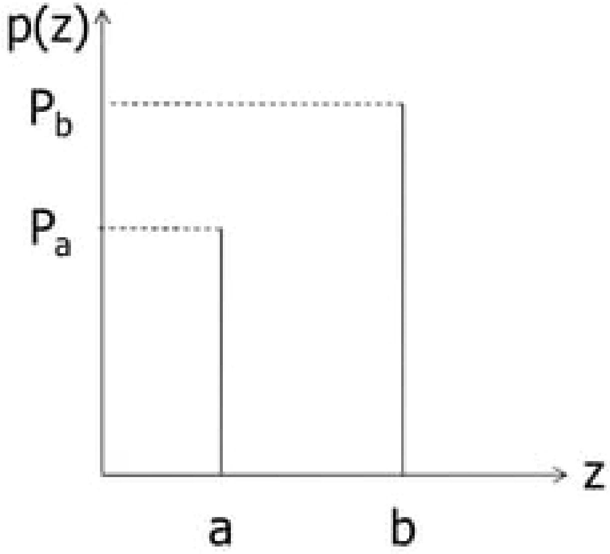


Image Averaging

- Consider a noisy image $g(x,y)$ formed by the addition of noise $\eta(x,y)$ to an original image $f(x,y)$

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

Image Averaging

- If noise has zero mean and be uncorrelated then it can be shown that if

$\overline{g}(x, y)$ = image formed by averaging K different noisy images

$$\overline{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

Image Averaging

- then

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

$\sigma^2_{\bar{g}(x,y)}, \sigma^2_{\eta(x,y)}$ = variances of $\bar{g}$ and η

if K increase, it indicates that the variability (noise) of the pixel at each location (x,y) decreases.

Image Averaging

- thus

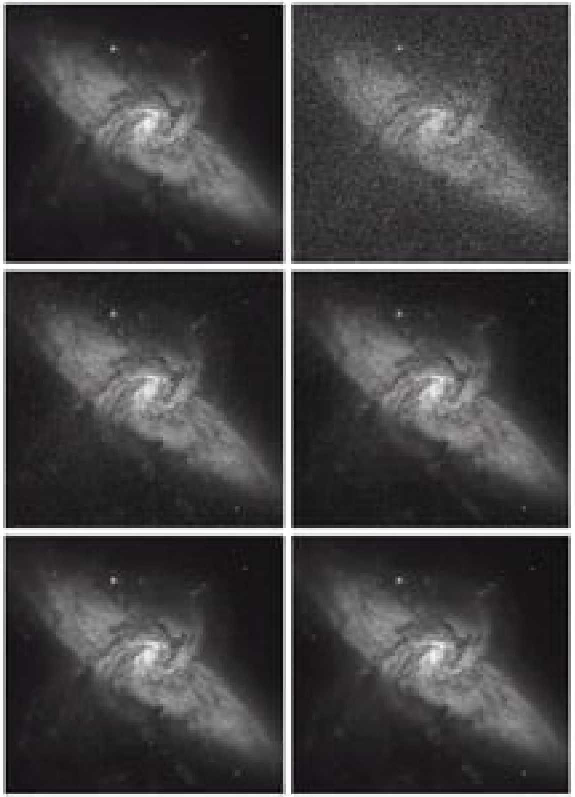
$$E\{\overline{g}(x,y)\} = f(x,y)$$

$E\{\overline{g}(x,y)\}$ = expected value of $\overline{g}$ (output after averaging)

= original image f(x,y)

Image Averaging-Example

a	b
c	d
e	f



- a) original image
- b) image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels.
- c). -f). results of averaging K = 8, 16, 64 and 128 noisy images

Enhancement by Mask Processing or Spatial Filtering

- A spatial mask is a n×n (often n being odd) matrix with matrix elements being called as the mask ***coefficients*** or ***weights***.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

These masks are called the ***spatial filters***.

Enhancement by Mask Processing or Spatial Filtering

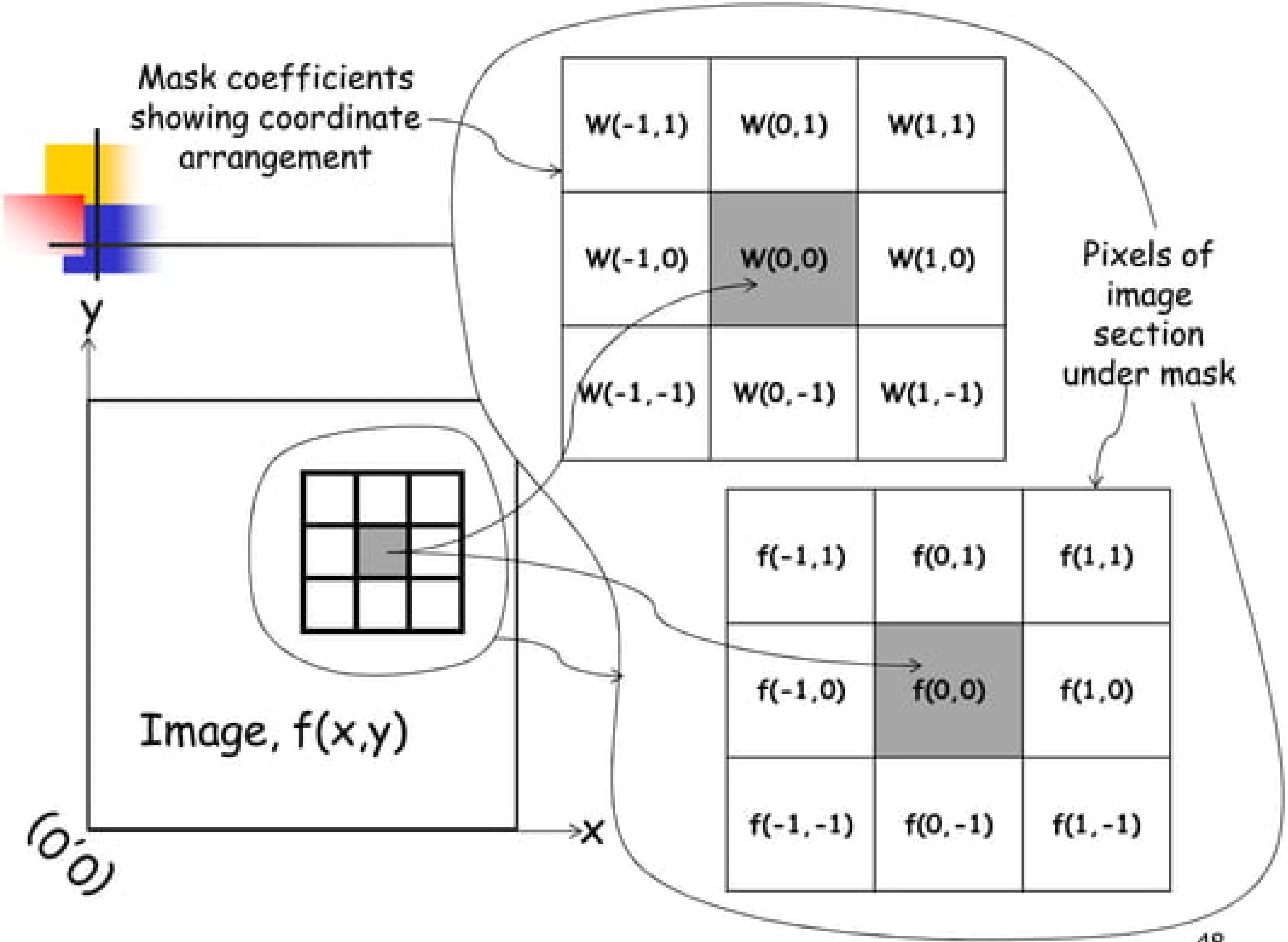
- These masks are called the ***spatial filters***. The gray level of the pixel at the center of the spatial mask is replaced by the weighted sum, R given by

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

where z_i, i=1,2,…,9 is the gray level of the pixel under the mask weight, w_i. The value, R is called the response of the spatial mask.

- The response, R is a linear relation for the linear spatial filters.

- The response, R is a non-linear relation for the non-linear spatial filters.


Mask coefficients showing coordinate arrangement
W(-1,1)
W(0,1)
W(1,1)
W(-1,0)
W(0,0)
W(1,0)
W(-1,-1)
W(0,-1)
W(1,-1)
Pixels of image section under mask
y
f(-1,1)
f(0,1)
f(1,1)
f(-1,0)
f(0,0)
f(1,0)
f(-1,-1)
f(0,-1)
f(1,-1)
Image, f(x,y)
(0,0)
x

Smoothing Spatial Filters

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

Smoothing Spatial Filters

- The followings are the *smoothing* filters:
  - Spatial Averaging or Lowpass Filter
  - Mean Filters
  - Median Filters

Lowpass (Spatial) Filter

- Lowpass spatial filter removes the sharp gray-level transitions while retaining the relatively smooth areas, hence producing blurring effect. In a lowpass spatial filter, the gray level of the pixel at the center of the spatial mask is replaced by the weighted average of the pixels under the spatial mask i.e., by the weighted average of its neighbourhood. Hence, lowpass spatial filtering is also called the ***neighbourhood averaging***. Thus, for a mask of size m×n for lowpass filtering, the response of the mask or spatial filter is given by

$$R = \frac{1}{mn}\sum_{i=1}^{mn} z_i$$

Lowpass (Spatial) Filter

- Examples

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

$3 \times 3\ mask$

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

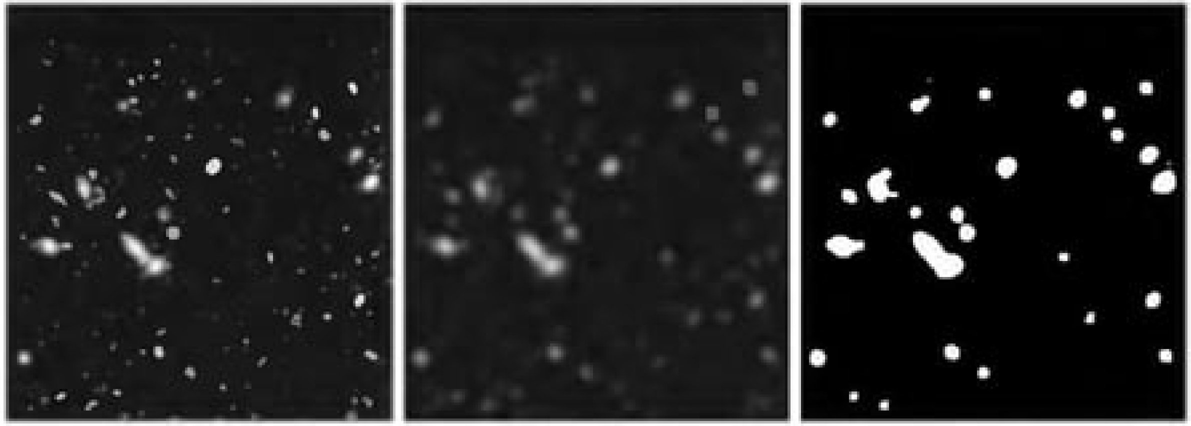
$5 \times 5\ mask$

Lowpass (spatial) Filter - Blurring

Original image

Result of lowpass (spatial) filtering - blurring

Lowpass (Spatial) Filter



1. Image from Hubble Space Telescope 2. Result of a 15×15 averaging spatial mask 3. Result of thresholding

Mean Filters

Arithmetic mean filter

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

A mean filter simply smoothes local variations in an image. Noise is reduced as a result of blurring.

Geometric mean filter

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

A geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

Mean Filters

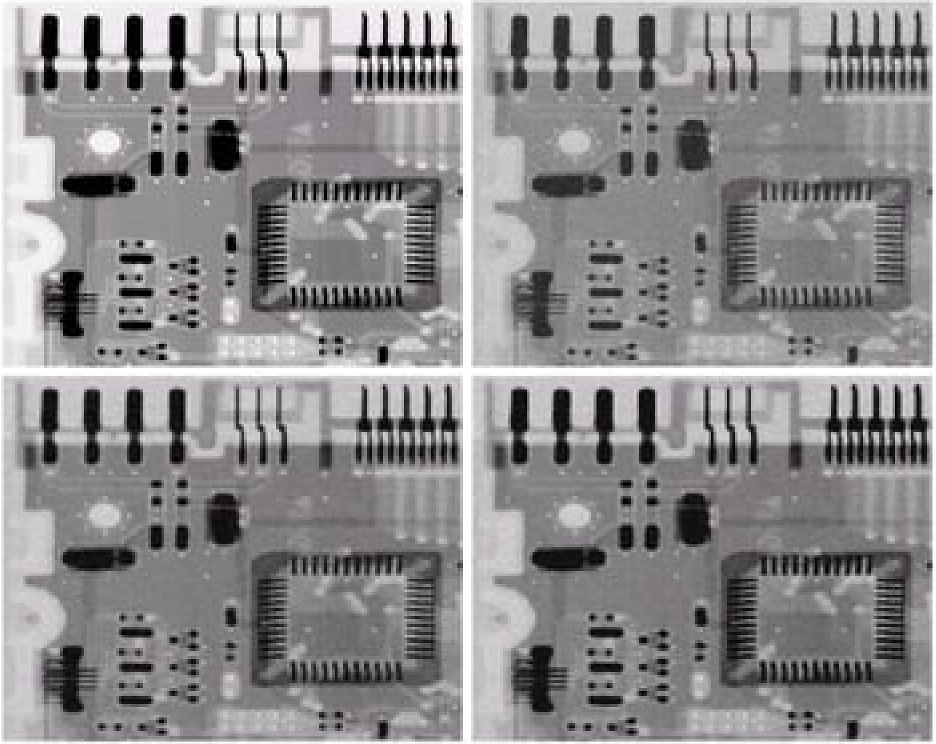


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Mean Filters

Harmonic Mean Filter

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

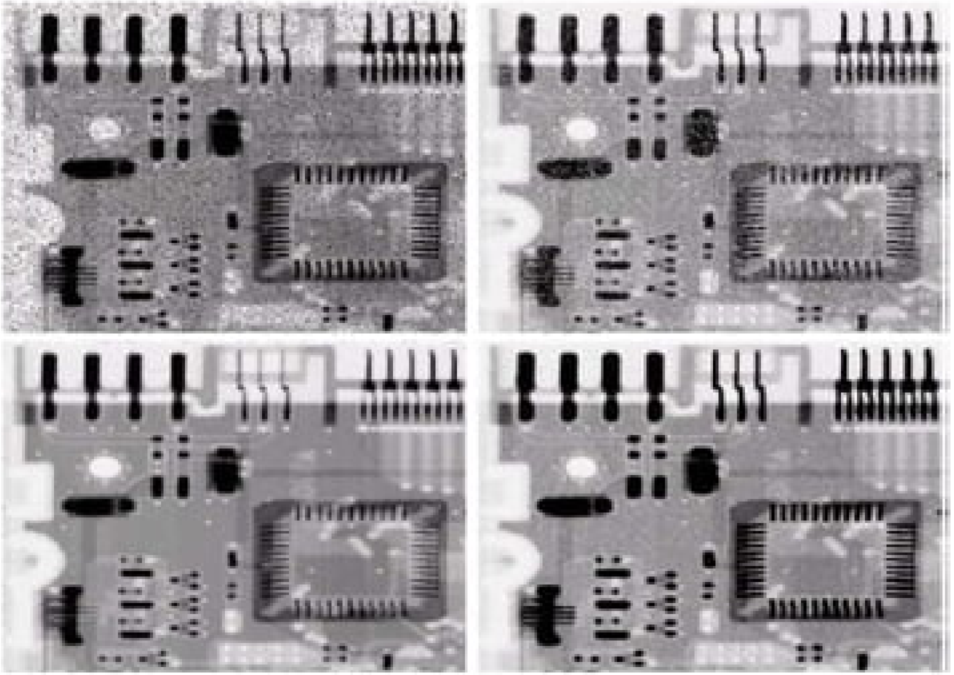
It does well also with other types of noise like Gaussian noise.

Contraharmonic Mean Filter

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

Q = order
+Q remove pepper noise
-Q remove salt noise
Q = 0 (arithmetic mean)
Q= -1 (Harmonic mean)

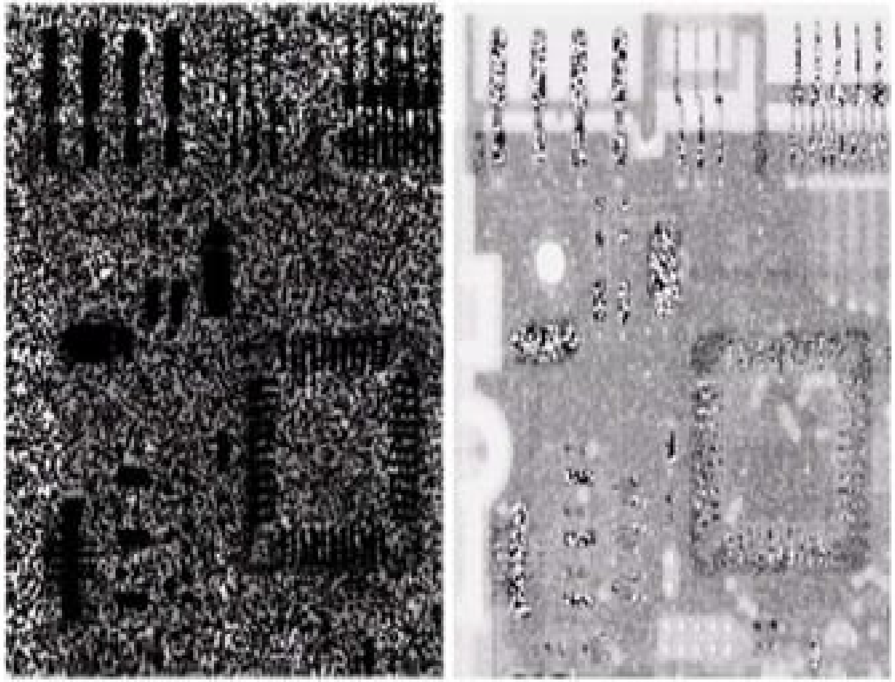
Mean Filters



a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Mean Filters



a b

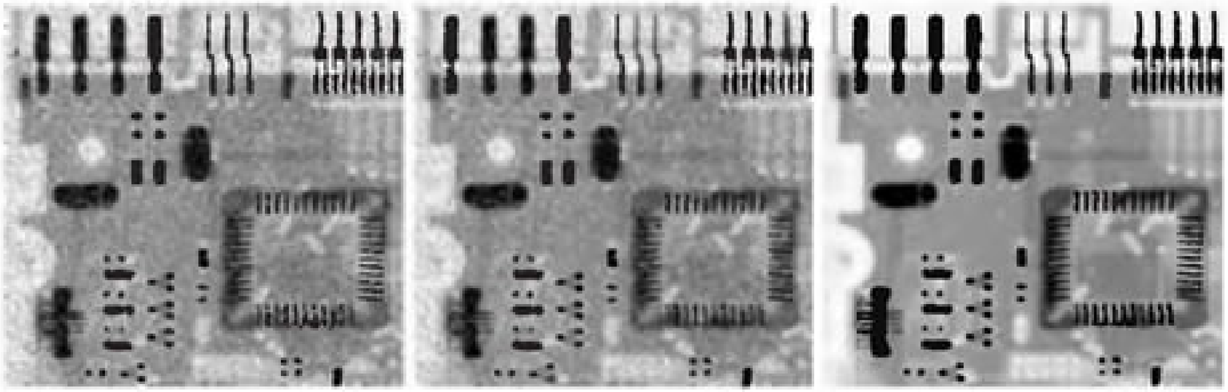
FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Median (Spatial) Filter

- A lowpass filter, if used for noise reduction, blurs edges and other sharp details. An alternate approach for noise reduction without blurring effect is the use of median spatial filters. In a median filter, the gray level of the pixel at the center of the spatial mask is replaced by the median of its neighbourhood i.e., by the median of the gray levels of the pixels under the spatial mask. Median spatial filters are very effective when the noise pattern consists of strong, spike-like components. Median filters are non-linear spatial filters. Median filters are the best-known in the category of the **order-statistics filters**.

Median (Spatial) Filter

- Example



1. X-ray image of circuit board corrupted by speckle or salt & pepper noise 2. Result of a 3×3 averaging spatial mask 3. Result of a 3×3 median filter

Directional Smoothing

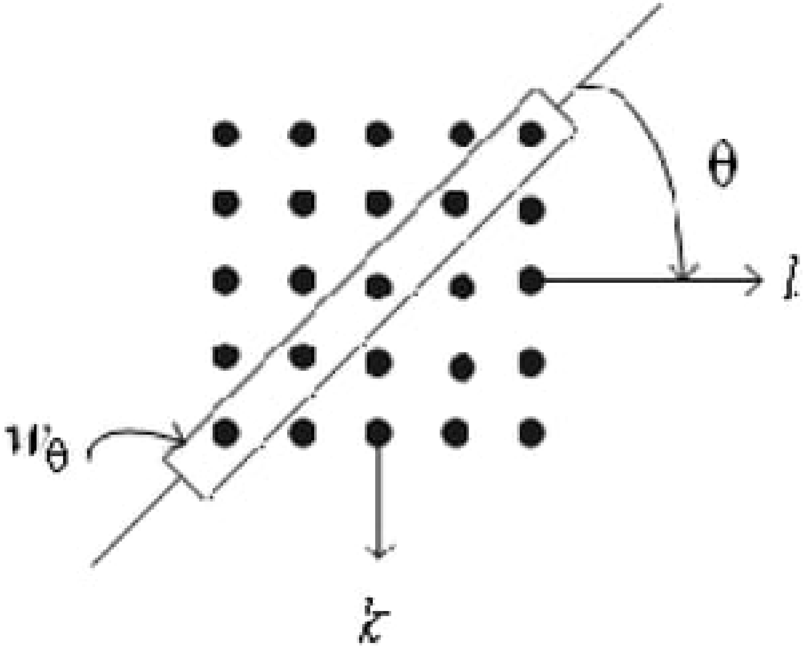
- Smoothing often results in blurring of edges.
- To protect edges from being blurred, directional averaging filters are used.
- Spatial averages are calculated in several directions as

$$v(m,n,\theta) = \frac{1}{N_\theta} \sum\sum_{(k,l \in w_\theta)} y(m-k,n-l)$$

- The direction, θ is chosen such that $|y(m,n) - v(m,n,\theta)|$ is minimum. Then it is set that

$$v(m,n) = v(m,n,\theta)$$

Directional Smoothing



Homomorphic filtering

- An image, $f(x,y)$ is expressed in terms of its illumination and reflectance components as

  $f(x,y)=i(x,y)r(x,y)$
- It is impossible to operate separately on the frequency components $i(x,y)$ and $r(x,y)$ since Fourier transform of product of two functions is not separable.
- Taking logarithm of both sides of Equ(1), we have

  $\ln[f(x,y)]=z(x,y)=\ln[i(x,y)]+\ln[r(x,y)]$
- Then taking Fourier transform, we have

  $FT[\ln[f(x,y)]]=FT[z(x,y)]=FT[\ln[i(x,y)]] + FT[\ln[r(x,y)]]$

  $\Rightarrow Z(u,v)=I(u,v) + R(u,v)$

Homomorphic filtering

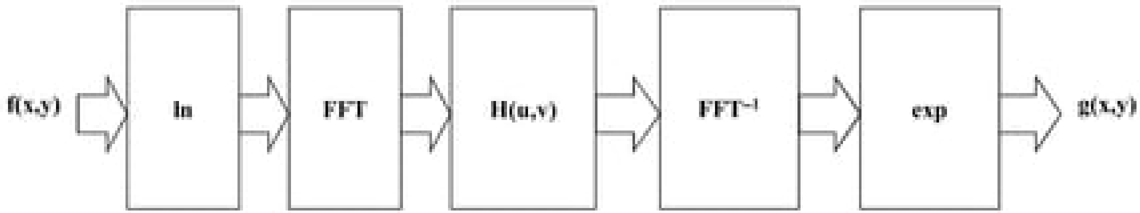
- Then processing by means of a filter with response, $H(u,v)$, we have

  $H(u,v)\ Z(u,v)=S(u,v)=H(u,v)\ I(u,v) + H(u,v)\ R(u,v)$
- Then taking inverse Fourier transform, we have
- $IFT[S(u,v)]=s(x,y)=IFT[H(u,v)I(u,v)] + IFT[H(u,v)R(u,v)]$
- $\Rightarrow s(x,y)=i'(x,y) + r'(x,y)$
- Then finally, taking exponential, we have

  $\exp[s(x,y)]=g(x,y)=\exp[i'(x,y) + r'(x,y)]=\exp[i'(x,y)]\ \exp[r'(x,y)]$

  $\Rightarrow g(x,y)=i_0(x,y)r_0(x,y)$
- where $i_0(x,y)$ and $r\text{-}_0(x,y)$ are the illumination and reflectance components of the output image.
- This approach is termed as ***homomorphic filtering***.

Homomorphic filtering

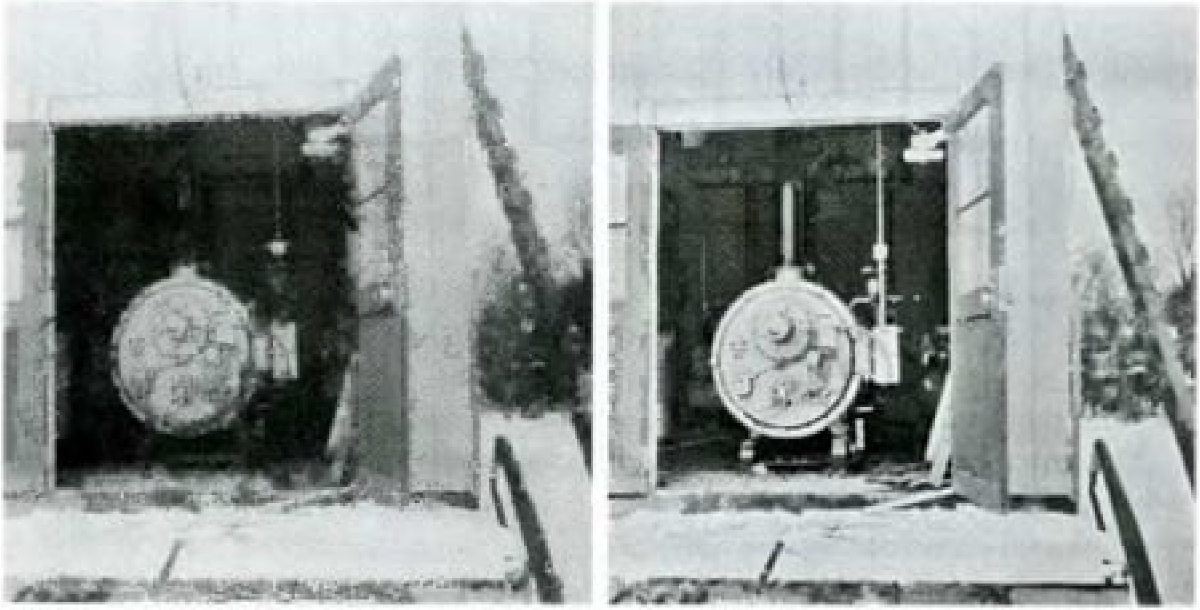


Homomorphic Filtering

Homomorphic filtering

- The homomorphic filtering is used to achieve contrast stretching and dynamic range compression simultaneously since the contrast of an image depends on the reflectance component and the dynamic range, on the illumination component.

Homomorphic filtering



Color Image Enhancement

- Assignment: Please refer to Fundamentals of Digital Image Processing by Anil K. Jain, page 262.

Unit 3

IMAGE RESTORATION

Image restoration

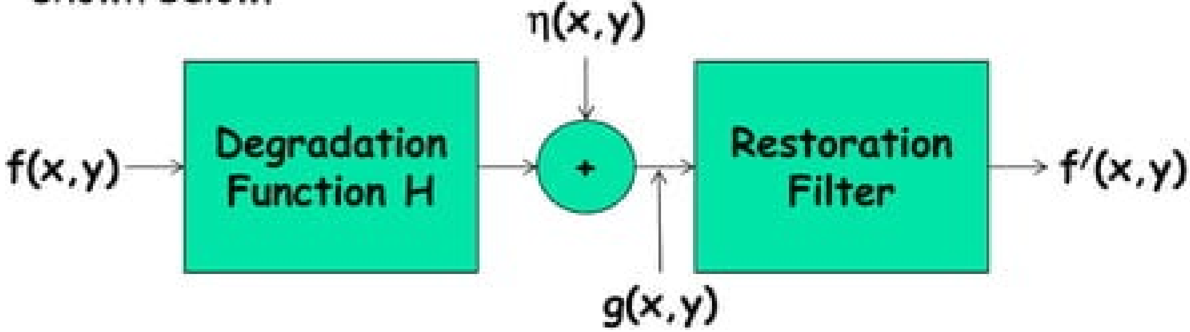
- Image restoration is to reconstruct or recover an image that has been degraded using some prior knowledge of the degradation phenomenon.

- Image restoration usually involves formulating a criterion of goodness that will yield an optimal estimate of the desired result. Thus this is an objective process.

- Enhancement techniques basically are heuristic procedures designed to manipulate an image in order to take advantage of the psychophysical aspects of the human visual system. Thus this is a subjective process.

Model of Degradation/ Restoration Process

- The (image) degradation process is modeled as a system, H together with an additive noise term, n(x,y) operating on an input image, f(x,y) to produce a degraded image, g(x,y) as shown below.



- The degradation process is mathematically expressed as

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

Model of Degradation/ Restoration Process

or in matrix form as

$$g = Hf + \eta$$

- where (i) **f**, **g** and **n** are MN×1 column matrices formed by stacking the rows of the M×N matrices formed from the extended (zero-padded) functions, $f_e(x,y)$, $g_e(x,y)$ and $n_e(x,y)$ of the original functions, f(x,y), g(x,y) and n(x,y), respectively, (ii) **H** is a MN×MN ***block circulant matrix*** formed by stacking the circulant matrix, H_j which is in turn constructed from the jth row of the extended (zero-padded) function, $h_e(x,y)$ of the original impulse or unit sample or point spread function, h(x,y) of the degradation system, H.

Model of Degradation/ Restoration Process

$$
H = \begin{bmatrix}
H_0 & H_{M-1} & H_{M-2} & \bullet & \bullet & \bullet & H_1 \\
H_1 & H_0 & H_{M-1} & \bullet & \bullet & \bullet & H_2 \\
H_2 & H_1 & H_0 & \bullet & \bullet & \bullet & H_3 \\
\bullet & \bullet & & & & & \bullet \\
\bullet & \bullet & & & & & \bullet \\
\bullet & \bullet & & & & & \bullet \\
H_{M-1} & H_{M-2} & H_{M-3} & \bullet & \bullet & \bullet & H_0
\end{bmatrix}
$$

Model of Degradation/ Restoration Process

where

$$
\mathrm{H}_j = \begin{bmatrix}
h_e(j,0) & h_e(j,N-1) & . \quad . \quad . & h_e(j,1) \\
h_e(j,1) & h_e(j,1) & . \quad . & h_e(j,2) \\
. & . & & . \\
. & . & & . \\
. & . & & . \\
h_e(j,N-1) & h_e(j,N-2) & . \quad . & h_e(j,0)
\end{bmatrix}
$$

Algebraic Approach to Image Restoration

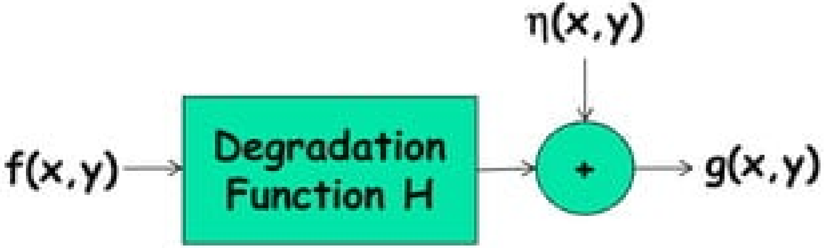
- The objective of image restoration is to estimate the original image, **f** from the degraded image, **g** using some knowledge or assumption about **H** and η.

- The objective of algebraic approach is to seek an estimate, **f**$'$, of the original image, **f** from the degraded image, **g** such that a predefined criterion function is minimized.

- There are two basic algebraic approaches: unconstrained and constrained restoration.

Unconstrained Restoration: Inverse Filtering

- The basic image degradation model



- From the basic image degradation model,

$$\eta = g - Hf \text{ -------}(1)$$

Unconstrained Restoration: Inverse Filtering

- In the absence of any knowledge about the noise, η, the objective of unconstrained restoration is to seek an estimate, $\mathbf{f}'$ of the original image, $\mathbf{f}$ from the degraded image, $\mathbf{g}$ such that $\mathbf{Hf}'$ approximates $\mathbf{g}$ and the norm of the noise term is minimized. That is, $\mathbf{f}'$ is found such that

$$\|\boldsymbol{\eta}\|^2 = \|\mathbf{g} - \mathbf{Hf}'\|^2 \text{-------} (2)$$

is minimum, where

$$\|\boldsymbol{\eta}\|^2 = \eta^T \eta, \quad \text{norm of } \eta$$

$$\|\mathbf{g} - \mathbf{Hf}'\|^2 = (\mathrm{g} - \mathrm{Hf}')^T (\mathrm{g} - \mathrm{Hf}'), \quad \text{norm of } (\mathrm{g} - \mathrm{Hf}')$$

Unconstrained Restoration: Inverse Filtering

- The minimization of Equ(2) is achieved by differentiating it with respect to $\mathbf{f}'$ and equating the result to zero.

$$\frac{\partial\left(\|\boldsymbol{\eta}\|^2\right)}{\partial\mathbf{f}'} = -2\mathbf{H}^{\mathrm{T}}\left(\mathbf{g}-\mathbf{H}\mathbf{f}'\right) = \mathbf{0} - - - - - - - -(3)$$

- Solving Equ(3) for $\mathbf{f}'$,

$$\mathbf{f}' = \mathbf{H}^{-1}\mathbf{g} - - - - - - - - - - - - - -(4)$$

- Equ(4) is the **inverse filtering**.

Unconstrained Restoration: Inverse Filtering

- The frequency domain representation of Equ(4) is

$$F^{/}(u,v)=\frac{G(u,v)}{H(u,v)}\text{-------------}(5)$$

where $F^{/}(u,v)$, $G(u,v)$ and $H(u,v)$ are the Fourier transforms of f, g and h, respectively.

- In the presence of noise, Equ(5) becomes

$$F^{/}(u,v)=\frac{G(u,v)}{H(u,v)}+\frac{N(u,v)}{H(u,v)}\text{-------------}(6)$$

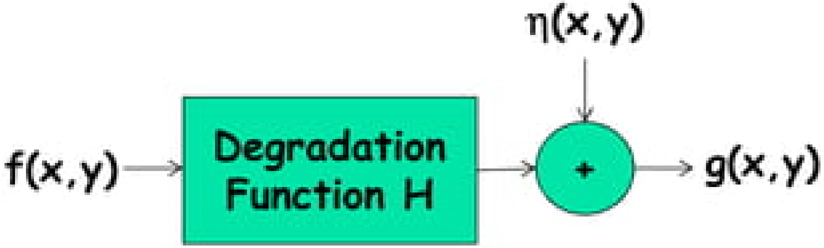
where $N(u,v)$ is the Fourier transform of the noise, η.

Unconstrained Restoration: Inverse Filtering

- The disadvantages of the inverse filtering (unconstrained restoration) are: (i) The complete knowledge about H and η is required and (ii) The restoration process is very sensitive to noise. That is, the restoration result is dominated by noise if H(u,v) is zero or small.

Constrained Restoration

- The basic image degradation/restoration model



- From the basic image degradation/restoration model,

$$\eta = g - Hf \text{ -------} (1)$$

Constrained Restoration

- The objective of constrained restoration is to seek an estimate, f' of the original image, f from the degraded image, g such that the criterion function

$$\mathbf{J}(\mathbf{f}') = \left\|\mathbf{Qf}'\right\|^2 + \alpha\left(\left\|\mathbf{g} - \mathbf{Hf}'\right\|^2 - \left\|\boldsymbol{\eta}\right\|^2\right) ------- (2)$$

is minimum, where

$$\left\|\boldsymbol{\eta}\right\|^2 = \eta^T\eta, \quad \text{norm of } \eta$$

$$\left\|\mathbf{g} - \mathbf{Hf}'\right\|^2 = \left(g - Hf'\right)^T\left(g - Hf'\right), \quad \text{norm of } (g - Hf')$$

Q is an operator on **f**

α is Lagrange's multipler.

Constrained Restoration

- The minimization of Equ(2) is achieved by differentiating it wrt $\mathbf{f}'$ and equating the result to zero.

$$\frac{\partial\left[\mathbf{J}(\mathbf{f}')\right]}{\partial \mathbf{f}'} = 2\mathbf{Q}^T\mathbf{Q}\mathbf{f}' - 2\alpha\mathbf{H}^T\left(\mathbf{g} - \mathbf{H}\mathbf{f}'\right) = \mathbf{0} ------- (3)$$

- Solving for $\mathbf{f}'$,

$$\mathbf{f}' = \left[\mathbf{H}^T\mathbf{H} + \gamma\mathbf{Q}^T\mathbf{Q}\right]^{-1}\mathbf{H}^T\mathbf{g} ------- (4)$$

where

$$\gamma = \frac{1}{\alpha}$$

Constrained Restoration

- Equ(4) yields different solutions for different choices of **Q**.

Constrained Restoration: Least Mean Square (Wiener) Filter

- Lease Mean Square or Wiener filter is obtained by defining

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{R}_{\mathrm{f}}^{-1}\mathbf{R}_{\eta} \text{-------}(5)$$

where

$\mathbf{R}_{\mathrm{f}}$ is the (auto) correlation matrix of $\mathbf{f}$.

$\mathbf{R}_{\eta}$ is the (auto) correlation matrix of $\boldsymbol{\eta}$.

- From Equ(4) and Equ(5)

$$\mathbf{f}' = \left[\mathbf{H}^{\mathrm{T}}\mathbf{H} + \gamma\, \mathbf{R}_{\mathrm{f}}^{-1}\mathbf{R}_{\eta}\right]^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{g} \text{-------}(6)$$

Constrained Restoration: Least Mean Square (Wiener) Filter

- The frequency domain representation of Equ(6) is

$$F^{/}(u,v) = \left[\frac{H^{*}(u,v)}{|H(u,v)|^{2} + \gamma\left[S_{\eta}(u,v)/S_{f}(u,v)\right]} \right] G(u,v) \text{ - - - - - - - - - - } (7)$$

where $F^{/}(u,v)$, $H(u,v)$ and $G(u,v)$ are the Fourier transforms of $f^{/}$, h and g, respectively and $S_{\eta}(u,v)$ and $S_{f}(u,v)$ are the Power Spectral Densities (PSDs) of η and f, respectively.

Constrained Restoration: Least Mean Square (Wiener) Filter

- With γ=1, Equ(7) becomes the so-called **Wiener filter**.
- With variable γ, Equ(7) becomes the so-called **parametric Wiener filter**.
- With $S_\eta(u,v)=0$ (no noise), Equ(7) becomes the **inverse filter**.
- With $H(u,v)=1$ for all (u,v) (no degradation, only noise), Equ(7) becomes the **smoothing (noise removal) filter**.

Constrained Least Squares Restoration

- Defining

$$\mathbf{Q} = \mathbf{P} \text{-------}(5)$$

where P is a Laplacian smoothing matrix, Equ(4) becomes

$$\mathbf{f}' = \left[\mathbf{H}^T\mathbf{H} + \gamma\,\mathbf{P}^T\mathbf{P}\right]^{-1}\mathbf{H}^T\mathbf{g} \text{--------}(6)$$

- The frequency domain representation of Equ(6) is

$$F'(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2}\right]G(u,v) \text{---------}(7)$$

where P(u,v) is the Fourier transform of the extended version of the 2D Laplacian operator, p(x,y) given by

Constrained Least Squares Restoration

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

- The norm of noise, η can be expressed in terms of its mean, μ_η and standard deviation, σ_η as

$$\|\boldsymbol{\eta}\|^2=(M-1)(N-1)[\sigma_\eta^2+\mu_\eta^2]-------(8)$$

where M & N are dimensions of the noise matrix.

Constrained Least Squares Restoration

- Procedure or algorithm: The procedure or algorithm for the constrained least squares restoration is as follows:

Step1: Initialize γ

Step2: Estimate $||\eta||^2$ using

$$\|\mathbf{\eta}\|^2 = (M-1)(N-1)[\sigma_\eta^2 + \mu_\eta^2]$$

Step3: Compute $F'(u,v)$ and hence f' using

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

where P(u,v) is the Fourier transform of the extended version of the 2D Laplacian operator, p(x,y) given by

Constrained Least Squares Restoration

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

Step4: Compute the residual, **r** and hence $\phi(\gamma)$ using

$$\phi(\gamma)=\|\mathbf{r}\|^2=\|\mathbf{g}-\mathbf{H}\mathbf{f}'\|^2$$

Step5: Increment γ if $\phi(\gamma)<||\eta||^2-a$

OR

Decrement γ if $\phi(\gamma)>||\eta||^2+a$

Step6: Return to Step3 and continue until the statement $\phi(\gamma)=||\eta||^2\pm a$ is true.

Pseudo Inverse Filter

- In the inverse filtering technique, it is often practically difficult to obtain the exact inverse, (1/H) of the degradation function, H, which is stable. Hence a stable version of the exact inverse of the degradation function, known as the **pseudo inverse filter**, is obtained.

Linear & Position-Invariant (LPI) Degradation

- A degradation system, H is linear if:

$$H\left[a_1 f_1(x,y) + a_2 f_2(x,y)\right] = a_1 H\left[f_1(x,y)\right] + a_2 H\left[f_2(x,y)\right]$$

- A degradation system, H is position-invariant if:

$$H\left[f(x,y)\right] = g(x,y) \Rightarrow H\left[f(x-\alpha, y-\beta)\right] = g(x-\alpha, y-\beta)$$

- A degradation system, H is LPI if it is linear and position-invariant.

Removal of blur caused by uniform linear motion

- Assignment: Please refer to Digital Image Processing by Gonzales & Woods 2nd Edition, pp.371-372.

Geometric Transformations

- Geometric transformations often are called rubber-sheet transformations, because they may be viewed as the process of "printing" an image on a sheet of rubber and then stretching this sheet according to some predefined set of rules.

- A geometric transformation consists of two basic operations:
  - (i) A spatial transformation, which defines the "rearrangement" of pixels on the image plane
  - (ii) Gray-level interpolation, which deals with the assignment of gray levels to pixels in the spatially transformed image.

Geometric Transformations

- **Spatial transformation**: an image with pixel coordinates (x,y) undergoes geometric distortion to produce an image g with coordinates (x',y'). This transformation may be expressed as

$$x_1 = r(x, y)$$
$$y_1 = s(x, y)$$

where r and s are the spatial transformations that produced the geometrically distorted image, g(x',y').

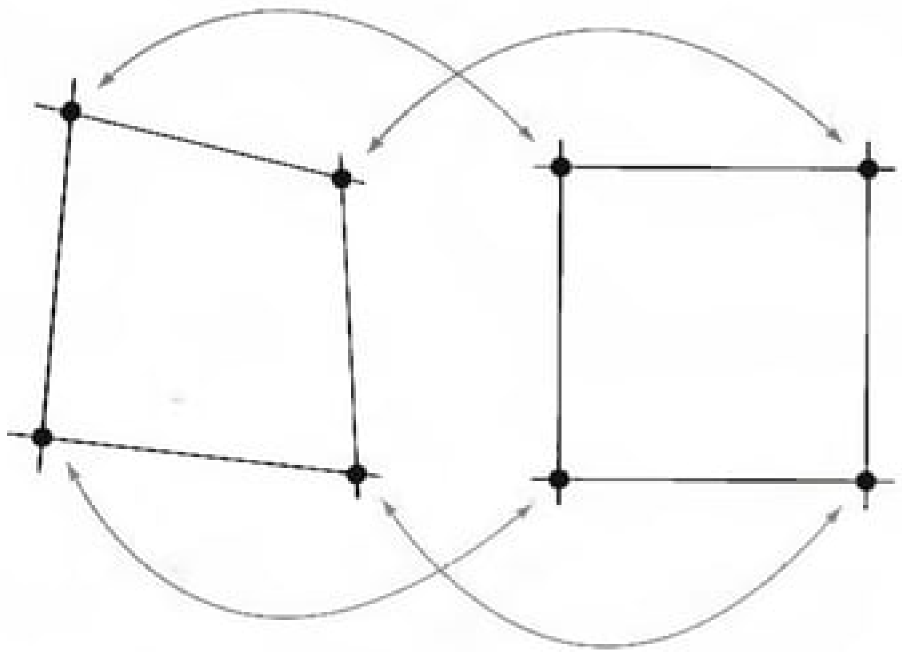
- Example: r(x,y)=x/2 and s(x,y)=y/2. This transformation is simply shrinking the image.

Geometric Transformations

- If $r(x,y)$ and $s(x,y)$ can be expressed analytically, then the original image, $f(x,y)$ can be easily recovered from the distorted image, $g(x',y')$ by applying the transformation in the reverse direction.

- If $r(x,y)$ and $s(x,y)$ cannot be expressed analytically as is the case in most practical applications, the spatial relocation of the pixels are formulated using the tiepoints which are a subset of pixels whose location in the distorted and corrected images is known precisely as shown in the following figure. A set of equations are derived for the spatial relocations of these tiepoints. The reverse transformation is achieved using them.

Geometric Transformations



Tiepoints

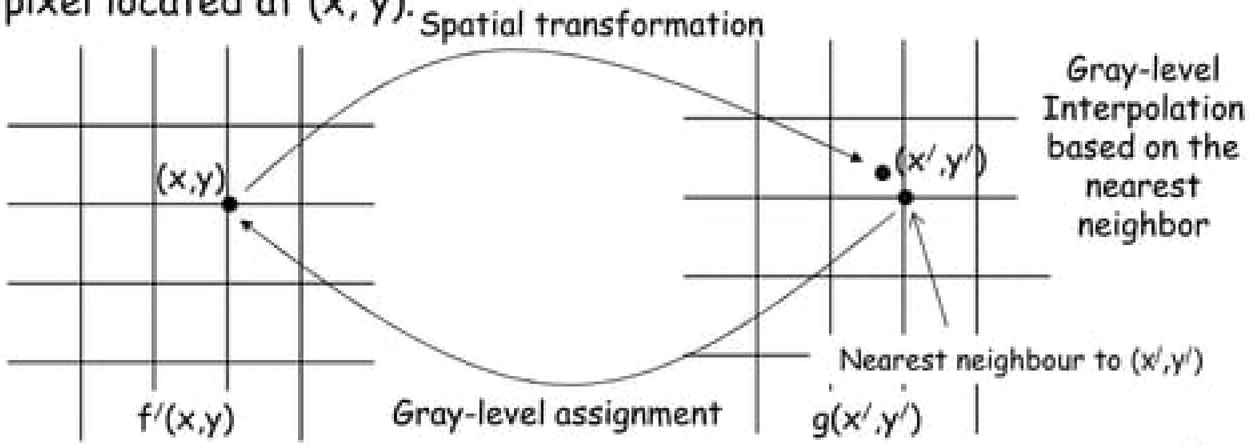
Gray-Level Interpolation

- When performing the reverse geometrical transformation to obtain the original image, $f(x,y)$ from the distorted image, $g(x',y')$. Depending on the coefficients of the equations for $r(x,y)$ and $s(x,y)$, the coordinates, (x',y') may be integers or non-integers. For a digital image, the coordinates must be integers. Hence, for the non-integer coordinates, their integer equivalents may not have any gray level values. Obtaining the gray level values for such coordinates is known as the gray-level interpolation.

- The gray-level interpolation techniques include (i) **zero-order interpolation**, (ii) **cubic convolution interpolation** and (iii) **bilinear interpolation.**

Gray-Level Interpolation

- **Zero-order interpolation**: This is based on the nearest-neighbor approach. 1) the mapping of integer (x,y) coordinates into fractional coordinates (x',y'), 2) the selection of the closest integer coordinate neighbor to (x',y') and 3) the assignment of the gray level of this nearest neighbor to the pixel located at (x, y).



Gray-Level Interpolation

- **Cubic convolution interpolation**: This fits a surface of the sin(z)/z type through a much larger number of neighbors (say, 16) in order to obtain a smooth estimate of the gray level at any desired point.
- **Bilinear interpolation**: This uses the gray levels of the four nearest neighbors usually is adequate. The gray-level value at the non-integral pairs of coordinates, denoted v(x',y'), can be interpolated from the values of its neighbors by using the relationship

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

  where the four coefficients are easily determined from the four equations in four unknowns that can be written using the four known neighbors of (x',y').
- When these coefficients have been determined, v(x',y') is computed and this value is assigned to the location in f(x,y) that yielded the spatial mapping into location (x',y').

Geometric transformation

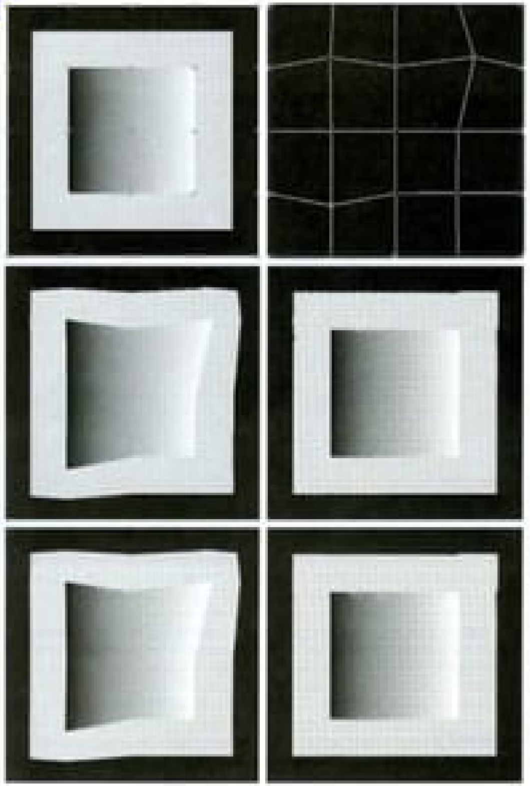


FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.

Unit 4

IMAGE SEGMENTATION

Image Segmentation

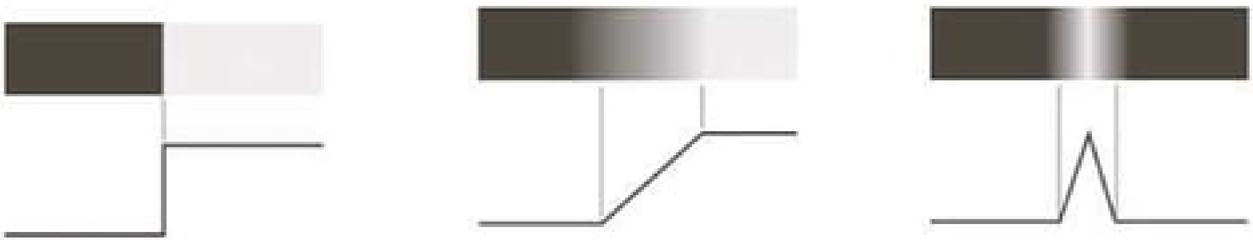
- Segmentation is to subdivide an image into its constituent regions or objects.
- Segmentation should stop when the objects of interest in an application have been isolated.
- Segmentation algorithms generally are based on one of two basic properties of intensity values: **discontinuity** and **similarity**.
  - The approach based on the discontinuity is to partition an image based on abrupt changes in intensity such as edges in the image.
  - The approach based on the similarity is to partition an image into regions that are similar according to a set of predefined criteria.

Detection of Discontinuities – Edge Detection

- There are three types of gray-level discontinuities: **points**, **lines** and **edges**.
  - Edge pixels are pixels at which the intensity of an image function abruptly changes and edges or edge segments are the set of connected edge pixels.
- Edge models: There are three basic edge models, namely the **step edge**, the **ramp edge** and the **roof edge**.
  - The **step edge** is a transition between two intensity levels occurring ideally over a distance of 1 pixel.
  - The **ramp edge** is a transition between two intensity levels occurring gradually over a distance of several pixels due to blurring. The slope of the ramp (and hence the width of the edge) is directly proportional to the degree of blurring.
  - The **roof edge** is a model of a line between two regions. The width of the edge is determined by the thickness and sharpness of the line.

Detection of Discontinuities - Edge Detection

- The following figures shows these three edge models.

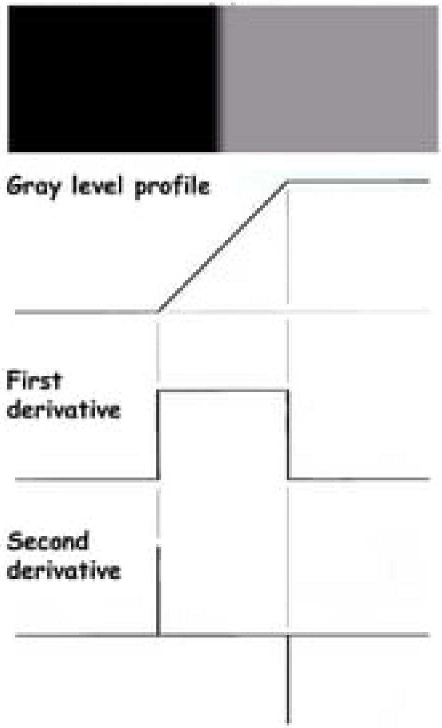


(a) (b) (c)

Models (top) and Gray-level profiles (bottom) of (a) step, (b) ramp and (c) roof edges.

Detection of Discontinuities – Edge Detection

- The common approach is to run a mask approximating either the **first order derivative (Gradient operator)** or the **second order derivative (Laplacian operator)**.
  - The magnitude of the first order derivative (Gradient) is used to determine whether a point is on the ramp.
  - The sign of the second order derivative (Laplacian) is used to determine whether an edge pixel is either on the dark (left) side of the edge or on the light (right) side of the edge.



Detection of Discontinuities – Edge Detection

- Gradient operator: The gradient of a function (image), f(x,y) is defined as the vector

$$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y} = G_x + G_y$$

where G_x is the gradient along the x-direction and G_y is the gradient along the y-direction. The magnitude, $|\nabla f|$ & phase, θ of the gradient are

$$|\nabla \mathrm{f}| = \sqrt{G_x^2 + G_y^2} \qquad \theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

- Gradient is a non-linear operator.

Detection of Discontinuities – Edge Detection

- The magnitude of the gradient is often approximated either using the difference along x- and y-directions as

$$\nabla f \approx \left[(z_5 - z_8)^2 + (z_5 - z_6)^2\right]^{1/2}$$

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6|$$

z_5	z_6
z_8	z_9

or using the cross differences along the diagonals as

$$\nabla f \approx \left[(z_5 - z_9)^2 + (z_6 - z_8)^2\right]^{1/2}$$

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8|$$

Detection of Discontinuities – Edge Detection

- The pair of 2 x 2 masks, known as the **Roberts Cross Gradient Operators**, using the cross differences along the diagonals are shown below.

1	0
0	-1

0	1
-1	0

Roberts Cross Gradient Operators

Detection of Discontinuities – Edge Detection

- Using a 2 x 2 mask has a practical difficulty as it does not have a center. Hence masks of size 3 x 3 are often preferred using either the differences along the x- and y-directions as

$$\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

or using the cross differences along the diagonals as

$$\nabla f \approx |(z_1 + z_2 + z_4) - (z_6 + z_8 + z_9)| + |(z_2 + z_3 + z_6) - (z_4 + z_7 + z_8)|$$

Detection of Discontinuities – Edge Detection

- The two pairs of3 x 3 masks, known as the **Prewitt** and **Sobel Operators**, using the differences along the x- and y-directions to detect the horizontal and vertical edges are shown below.

-1	-1	-1
0	0	0
-1	-1	-1

-1	0	-1
-1	0	-1
-1	0	-1

Prewitt Operators

-1	-2	-1
0	0	0
-1	-2	-1

-1	0	-1
-2	0	-2
-1	0	-1

Sobel Operators

Detection of Discontinuities – Edge Detection

- The two pairs of3 x 3 masks, known as the **Prewitt** and **Sobel Operators**, using the differences along the diagonals to detect the diagonal edges are shown below.

-1	-1	0
-1	0	1
0	1	1

0	1	1
-1	0	1
-1	-1	0

Prewitt Operators

-2	-1	0
-1	0	1
0	1	2

0	1	2
-1	0	1
-2	-1	0

Sobel Operators

Detection of Discontinuities – Edge Detection

- Laplacian Operator: Laplacian, for a function (image) f(x,y), is defined as

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- The Laplacian is a linear operator.

- The discrete form of the Laplacian of f(x,y), taking the 4-neighbours into account, is obtained by summing the discrete forms of partial derivatives along x- and y- directions as

$$\nabla^2 f = [4z_5 - (z_2 + z_4 + z_6 + z_8)]$$

Detection of Discontinuities – Edge Detection

or taking all the 8-neighbours into account, is obtained by summing the discrete forms of partial derivatives along x- and y-directions & along the diagonals as

$$\nabla^2 f = [8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8)]$$

- The corresponding 3 x 3 masks are

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Detection of Discontinuities – Edge Detection

- The Laplacian generally is not used in its original form for edge detection for the following reasons: (i) Second-order derivative is unacceptably sensitive to noise and (ii) The magnitude of the Laplacian produces double edges.
- The Laplacian is often used with Gaussian smoother given by

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

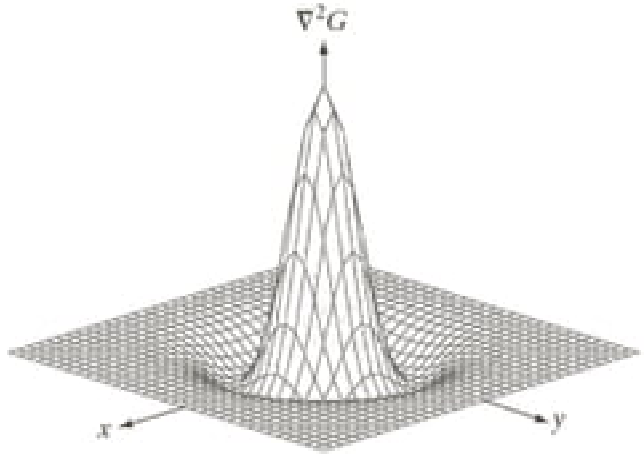
The Laplacian of h is given by

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

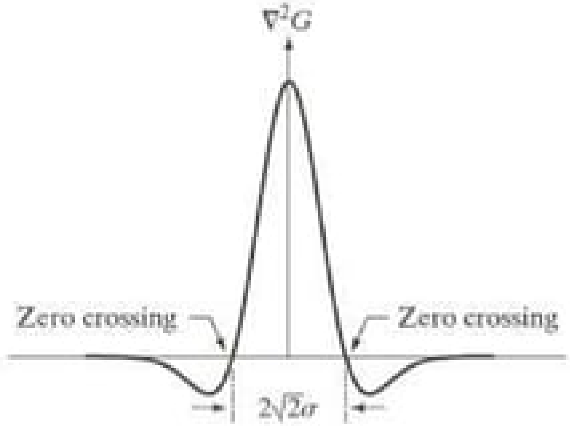
which is called the Laplacian of Gaussian (LoG).

Detection of Discontinuities - Edge Detection

- The Laplacian of Gaussian is sometimes called the Mexican Hat function because of its appearance.



LoG: 3D plot



LoG: Cross section showing zero-crossings

Detection of Discontinuities – Edge Detection

- To implement LoG, the image is convolved with the h and the result is then Laplacianed or the following mask which approximates the LoG is used.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

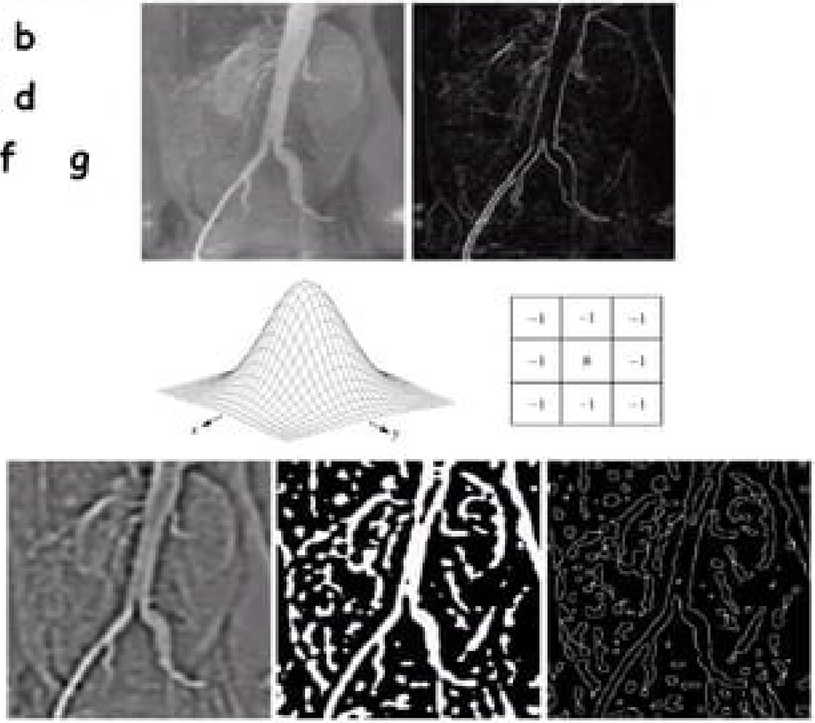
- To detect the edges the resulting image is thresholded (setting all its positive values to white and negative values to black) and the zero-crossings between these whites and blacks are found.

Detection of Discontinuities - Edge Detection - Example

a b

c d

e f g



a). Original image
b). Sobel Gradient
c). Spatial Gaussian smoothing function
d). Laplacian mask
e). LoG
f). Threshold LoG
g). Zero crossing

Edge Linking and Boundary Detection

- Edge detection algorithm are followed by linking procedures to assemble edge pixels into meaningful edges.
- Basic approaches
  - Local Processing
  - **Global Processing via the Hough Transform**
  - Global Processing via Graph-Theoretic Techniques

Edge-linking based on Hough Transformation

Reason for Hough transform:

- To find the subsets of the points that lie on straight lines in a given image, one possible solution is to first find all lines determined by every pair of points and then find all subsets of points that are close to particular lines.

- For a given image with n points, this procedure involves finding $n(n-1)/2$ lines and then performing $(n)(n(n-1))/2$ comparisons of every point to all lines. Hence the Hough transform.

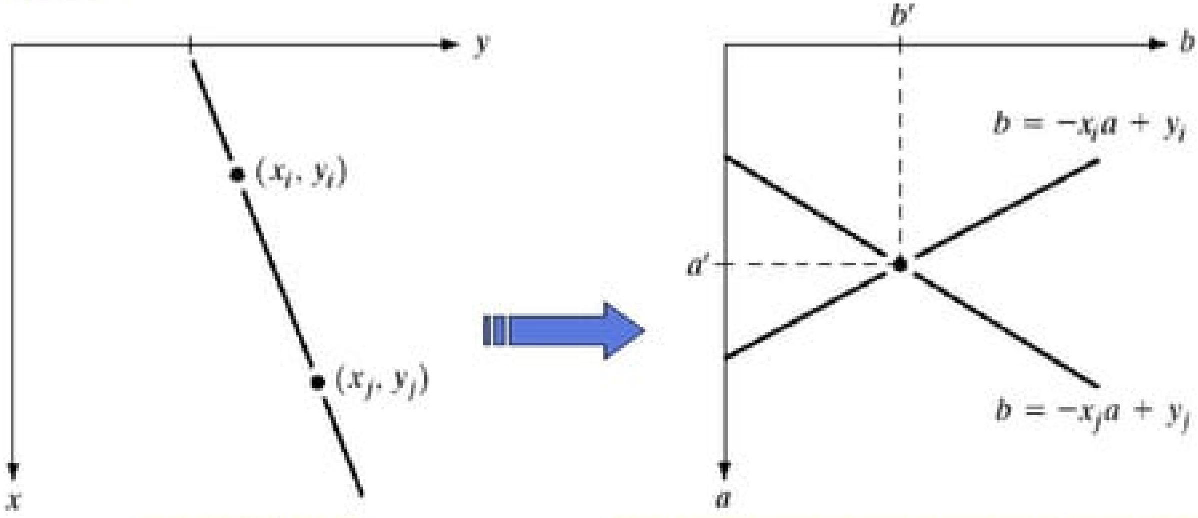
Edge-linking based on Hough Transformation

- Infinitely many lines pass through (x_i,y_i), all satisfying the equation $y_i=ax_i+b$ for varying values of a and b. But there is only a single line satisfying $b=-x_ia+y_i$ in the ab-plane (also called parameter space) for every point (x_i,y_i) in the xy-plane.

- In other words a second point (x_j, y_j) on the same line in the xy-plane intersects the line of (x_i,y_i) in the ab-plane at (a',b'), where a' is the slope and b' the intercept of the line containing both (x_i,y_i) and (x_j, y_j) in the xy-plane.

- All points (x_i ,y_i) contained on the same line must have lines in parameter space that intersect at (a',b').

Edge-linking based on Hough Transformation



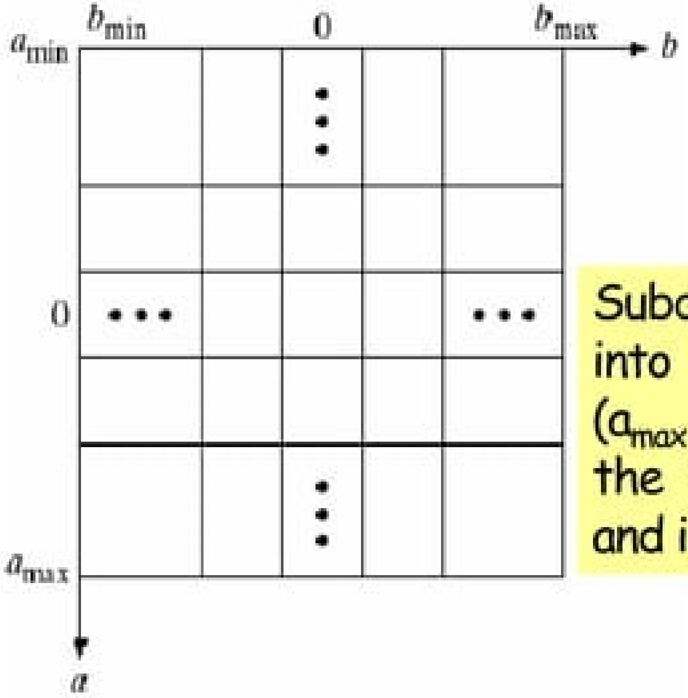
xy-plane

$y_i = ax_i + b$

ab-plane or parameter space

$b = -ax_i + y_i$

Edge-linking based on Hough Transformation



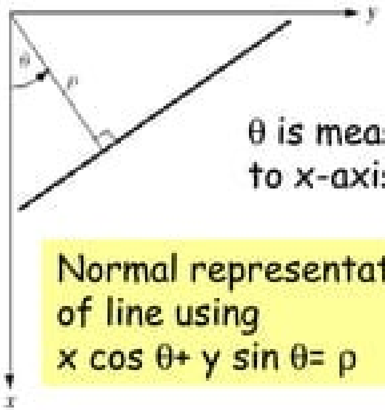
Subdivision of parameter space into accumulator cells where (a_{max}, a_{min}) and (b_{max}, b_{min}) are the expected ranges of slope and intercept values

Edge-linking based on Hough Transformation

- The parameter space is then subdivided into so-called accumulator cells where (a_{max}, a_{min}) and (b_{max}, b_{min}) are the expected ranges of slope and intercept values.
- The accumulator cell, $A(i,j)$ corresponds to the square associated with the parameter space coordinates, (a_i, b_j).
- All cells are initialized to zero.
- For every (x_k, y_k) in the xy-plane, a is chosen to be each subdivision value in the allowed range and corresponding b is calculated using $b = -ax_k + y_k$ and rounded to nearest allowed range.
- If a choice of a_p results in solution b_q then we let $A(p,q) = A(p,q)+1$.
- At the end of the procedure, value Q in $A(i,j)$ corresponds to Q points in the xy-plane lying on the line $y = a_i x + b_j$.

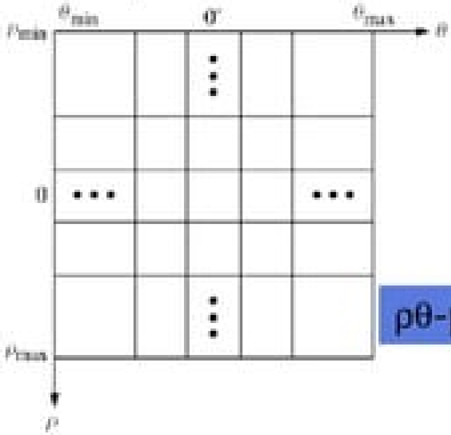
Edge-linking based on Hough Transformation

- The problem of using the equation y = ax + b is that the value of a is infinite for a vertical line.
- To avoid the problem, the equation $x \cos \theta + y \sin \theta = \rho$ is used to represent a line instead.
- Vertical line has $\theta = 90^\circ$ with ρ equals to the positive y-intercept or $\theta = -90^\circ$ with ρ equals to the negative y-intercept



θ is measured with respect to x-axis

Normal representation of line using
$x \cos \theta + y \sin \theta = \rho$



ρθ-plane

Edge-linking based on Hough Transformation

1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the ab (or $\rho\theta$)-plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Edge-linking based on Hough Transformation

- Hough is applicable to any function of the form g(**v**,**c**)=0 where **v** is a vector of coordinates and **c** is a vector of coefficients. For example, the Hough transform using circles is described below.
- Equation: $(x-c_1)^2 + (y-c_2)^2 = c_3^2$
- Three parameters (c_1, c_2, c_3)
- Cube like cells
- Accumulators of the form $A(i, j, k)$
- Increment c_1 and c_2 , solve of c_3 that satisfies the equation
- Update the accumulator corresponding to the cell associated with triplet (c_1, c_2, c_3)

Thresholding

- Thresholding may be viewed as an operation that involves tests against a function T of the form

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

where f(x, y) is the gray level of point (x, y) and p(x, y) denotes some local property of this point.

- A thresholded image g(x, y) is defined as

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

Thus, pixels labeled 1 (or any other convenient gray level) correspond to objects, whereas pixels labeled 0 (or any other gray level not assigned to objects) correspond to the background.

Thresholding

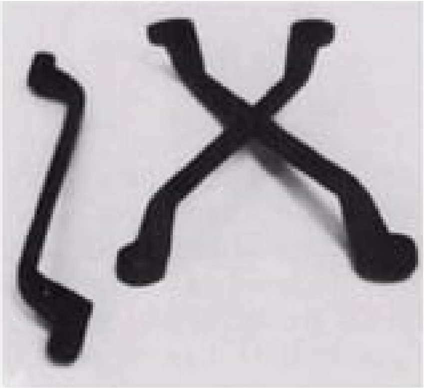
- **Global Thresholding**: the threshold, T depends only on the histogram of the image.
- **Local Thresholding**: the threshold, T at any point, (x,y) depends on the property of neighbourhood around that point.
- **Adaptive Thresholding**: the threshold, T at any point, (x,y) depends on both the neighbourhood property and the spatial coordinates x and y.
- **Multilevel Thresholding**: Here, multilevel thresholding classifies a point (x, y) as belonging to one object class if $T1 < f(x, y) < T2$, to the other object class if $f(x, y) > T2$, and to the background if $f(x, y) < T1$.

Basic Global Thresholding

- A global threshold can be obtained either by a visual inspection of the histogram of the image or automatically from the histogram via a simple algorithm.
- Simple algorithm to automatically obtain a global threshold from the histogram of the image:

1. Select an initial estimate for T.
2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values > T and G_2 consisting of pixels with gray level values $\leq$ T
3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2
4. Compute a new threshold value
5. $T = 0.5\ (\mu_1 + \mu_2)$
6. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.

Basic Global Thresholding by Visual Inspection - Example



Original Image

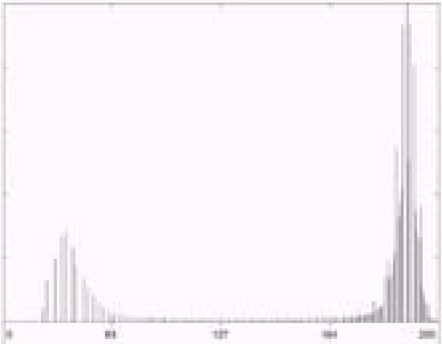
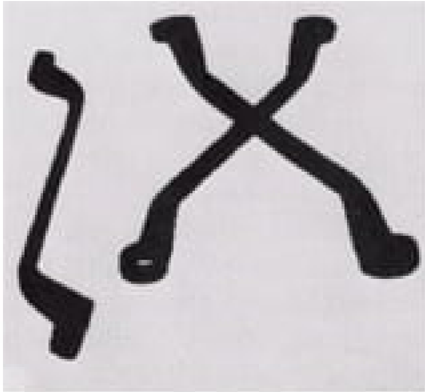


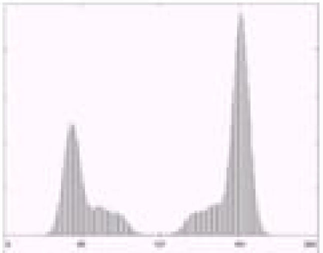
Image histogram



Result of global thresholding with T midway between the max and min gray levels

Note: Use of T midway between min and max gray levels produce binary image as shown above.

Basic Global Thresholding Automatic Approach - Example



a b
c

FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

Note: The clear valley of the histogram and the effectiveness of the segmentation between object and background.

$T_0 = 0$
3 iterations
with result T = 125

Basic Adaptive Thresholding

- In an image where a fixed global threshold does not give acceptable results, e.g., an image with poor illumination, a basic adaptive thresholding technique can be used as explained below.

1. The image is divided into smaller subimages.
2. Individual thresholds are chosen to segment each subimage.

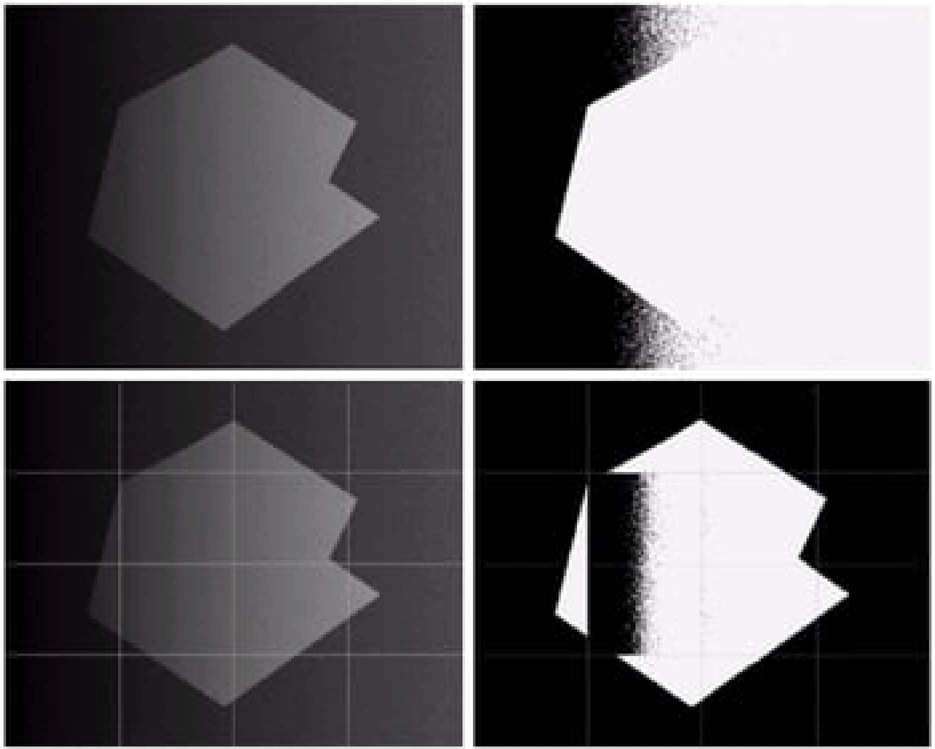
- The thresholds thus selected are "adaptive" to the pixel values in individual subimages.

- The improper subdivision can result in poor segmentation. Further subdivision of the improperly segmented subimage and subsequent adaptive thresholding can improve the process of segmentation.

Basic Adaptive Thresholding - Example

a b
c d

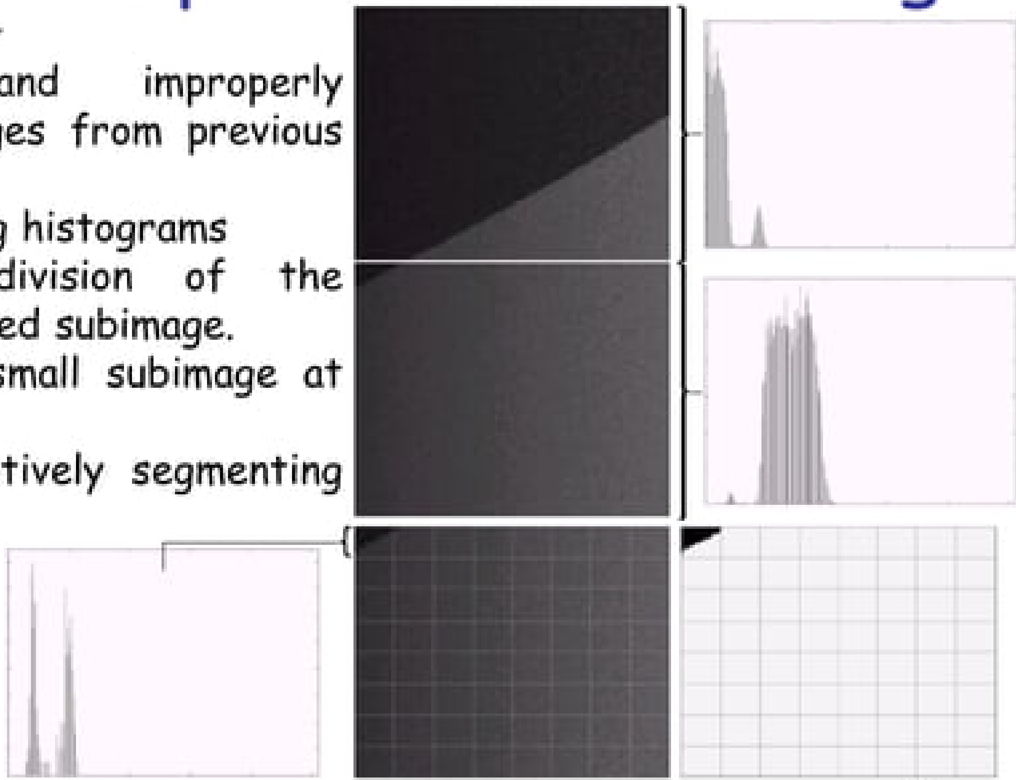
FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



Basic Adaptive Thresholding

a). Properly and improperly segmented subimages from previous example
b)-c). Corresponding histograms
d). Further subdivision of the improperly segmented subimage.
e). Histogram of small subimage at top
f). Result of adaptively segmenting d).

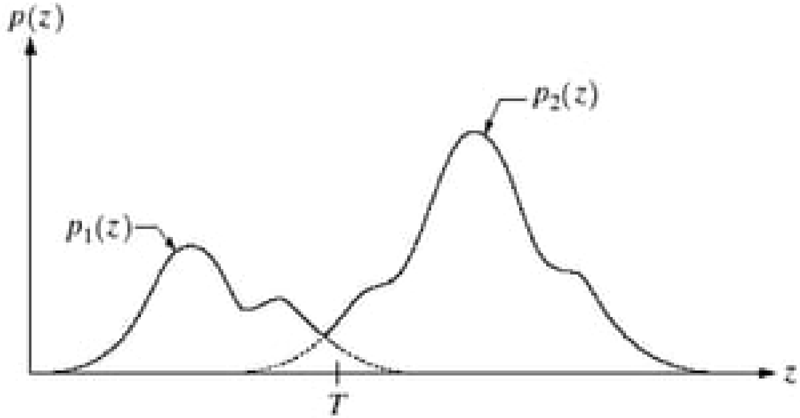
b
a
c
e d f



Optimal Global and Adaptive Thresholding

- Consider an image with only two principal gray level regions as shown below.

FIGURE 10.32 Gray-level probability density functions of two regions in an image.



- Assume that the larger of the two PDFs corresponds to the background levels while the smaller one describes the gray levels of objects in the image.

Optimal Global and Adaptive Thresholding

- The mixture probability density function describing the overall gray-level variation in the image is

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad \& \quad P_1 + P_2 = 1$$

- P_1 is the probability (a number) that a pixel is an object pixel. Similarly, P_2 is the probability that the pixel is a background pixel.

- The image is segmented by classifying all pixels with gray levels greater than a threshold T as background. All other pixels are called object pixels.

- The probability of erroneously classifying a background point as an object point is

$$E_1(T) = \int_{-\infty}^{T} p_2(z)dz$$

Optimal Global and Adaptive Thresholding

- Similarly, the probability of erroneously classifying an object point as background is

$$E_2(T) = \int_{T}^{\infty} p_1(z)\,dz$$

- Then the overall probability of error is

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

- To find the threshold value for which this error is minimal requires differentiating E(T) with respect to T (using Leibniz's rule) and equating the result to 0. The result is

$$P_1 p_1(T) = P_2 p_2(T)$$

Optimal Global and Adaptive Thresholding

- The above equation is solved for T to obtain an optimum threshold value.
- Note that if $P_1 = P_2$, then the optimum threshold is where the curves for $P_1(z)$ and $p2(z)$ intersect.
- Obtaining an analytical expression for T requires that the equations for the two PDFs are known.

Optimal Global and Adaptive Thresholding - Example

Example: use PDF = Gaussian density : $p_1(z)$ and $p_2(z)$

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

$$= \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}}$$

where

- μ_1 and σ_1^2 are the mean and variance of the Gaussian density of one object
- μ_2 and σ_2^2 are the mean and variance of the Gaussian density of the other object

Optimal Global and Adaptive Thresholding - Example

$$AT^2 + BT + C = 0 \quad \text{using } P_1 p_1(T) = P_2 p_2(T)$$

$$where \quad A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)$$

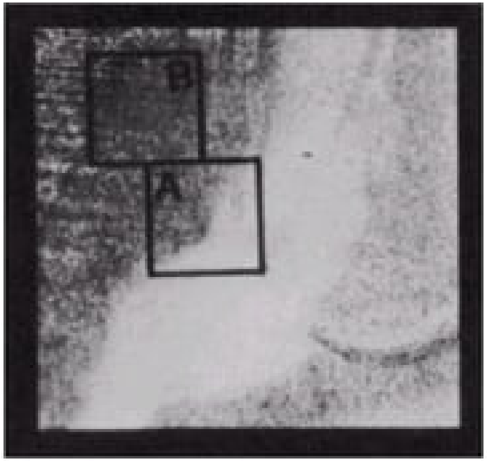
$$C = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 2\sigma_2^2 \ln(\sigma_2 P_1 / \sigma_1 P_2)$$

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

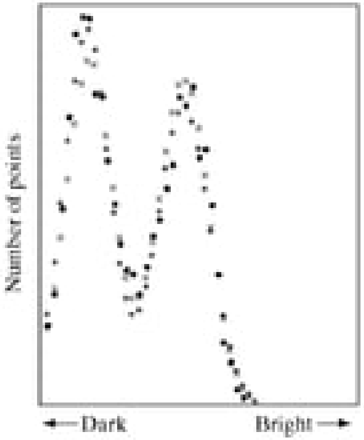
if $P_1 = P_2$ or $\sigma = 0$ then the optimal threshold is the average of the means.

Local Thresholding

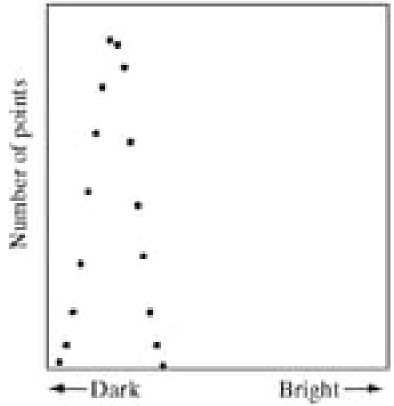
- The chances of selecting a "good" threshold are enhanced considerably if the histogram peaks are tall, narrow, symmetric, and separated by deep valleys.



Cardioangiogram

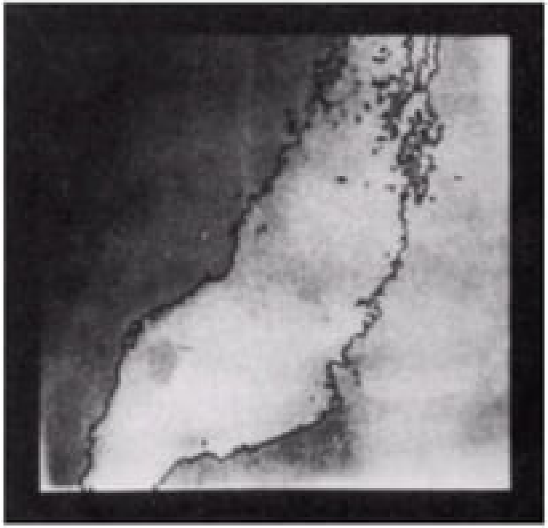


Histogram of A



Histogram of B

Local Thresholding - Example



Cardioangiogram showing superimposed boundaries.

Local Thresholding

- One approach for improving the shape of histograms is to consider only those pixels that lie on or near the edges between objects and the background.

  In this, the histograms are less dependent on the relative sizes of the object and the background.

  This however requires that the edges between the object and the background are known.
- The identification of whether the pixels are on the edges is done using the gradient.
- The identification of whether the pixels are on the left or right side of the edges is done using the Laplacian.

Local Thresholding

- These two quantities may be used to form a three-level image, as follows:

$$s(x,y)=\begin{cases}0, & \text{if } \nabla f < T \\ +, & \text{if } \nabla f \geq T \text{ and } \nabla^2 f \geq 0 \\ -, & \text{if } \nabla f \geq T \text{ and } \nabla^2 f < 0\end{cases}$$

where

all pixels that are not on an edge are labeled 0.

all pixels that are on the dark (left) side of an edge are labeled +.

all pixels that are on the light (right) side an edge are labeled -.

Local Thresholding

- A transition from - to + indicates the transition from a light background to a dark object.
- A 0 or + indicates the interior of the object.
- A transition from + to - indicates the transition from a dark object to a light background.
- Thus a horizontal or vertical scan line containing a section of an object has the following structure:

$$(\cdots)(-,+)(0\ or\ +)(+,-)(\cdots)$$

where $(\cdots)$ is any combination of +, - and 0.

Local Thresholding - Example

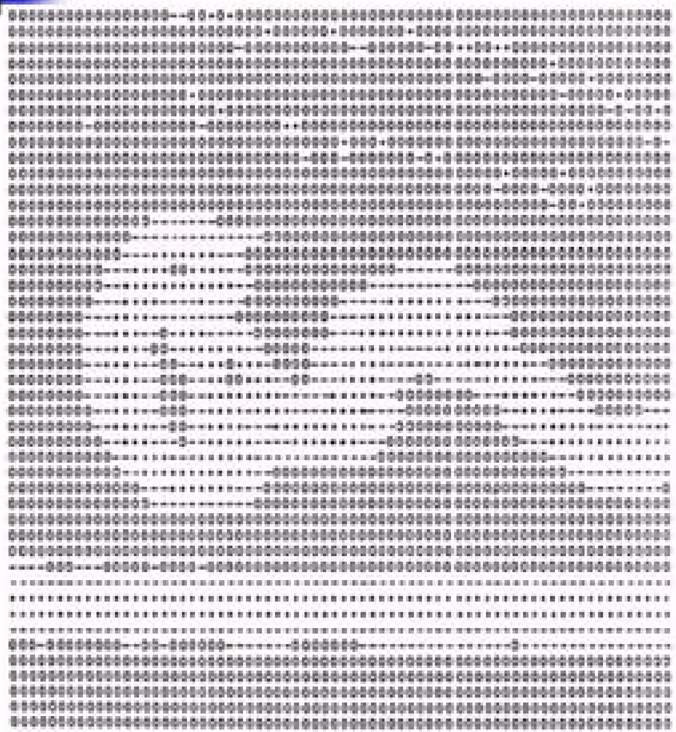
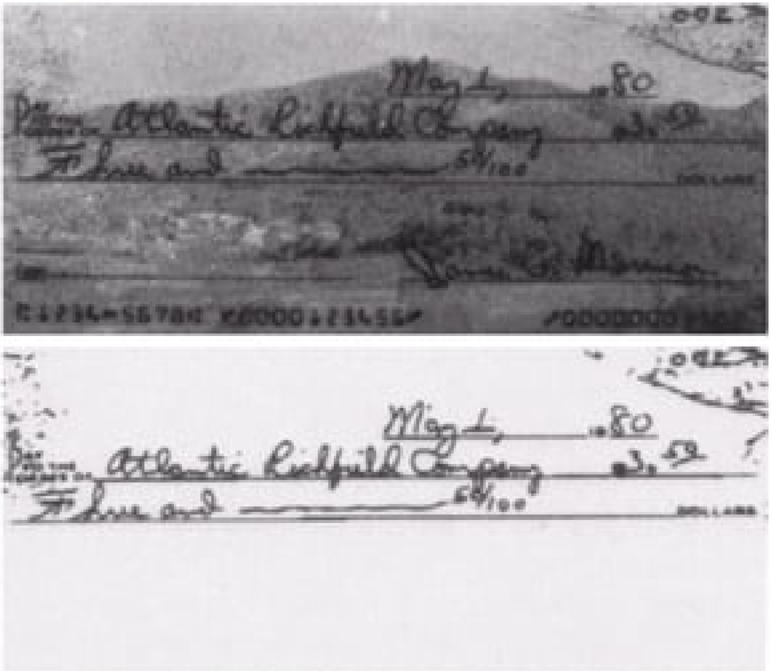


Image of a handwritten stroke coded by using the thresholding technique

Local Thresholding - Example



a

b

(a) Original image
(b) Image segmented by local thresholding

Region-Based Segmentation

- Let R represent the entire image region. We may view segmentation as a process that partitions R into n subregions, R_1, R_2, ..., R_n, such that

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

$$(b)\quad R_i \text{ is a connected region}, i = 1, 2, ..., n$$

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

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

$$(e)\quad P(R_i \cup R_j) = FALSE \;\; for\; i \neq j$$

Region-Based Segmentation

- Condition (a) indicates that the segmentation must be complete; that is, every pixel must be in a region.
- Condition (b) requires that points in a region must be connected in some predefined sense.
- Condition (c) indicates that the regions must be disjoint.
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region—for example $P(R_i)$ = TRUE if all pixels in R_i, have the same gray level.
- Condition (e) indicates that regions R_i, and R_j are different in the sense of predicate P.

Region Growing

- Region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria.
- The basic approach is to start with a set of "seed" points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (such as specific ranges of gray level or color).
- Selecting a set of one or more starting points depends on the problem under consideration.
- The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available.
- Grouping the pixels to form a region based on their similarity might result in disjoint regions. To prevent this attention must be paid to the connectivity of the pixels while grouping them.
- In addition to the criteria such as gray level, texture and colour that are local in nature, the criteria such as size of the growth, likeness of the candidate pixel to the pixels grown so far must also be considered to formulate a proper stopping rule.

Region Splitting and Merging

- An alternative approach to the region growing is to subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions

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

$$(b) \quad R_i \text{ is a connected region, } i = 1, 2, \ldots, n$$

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

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

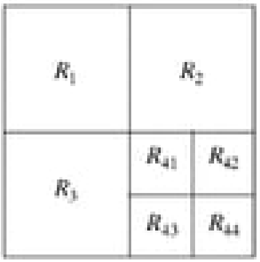
$$(e) \quad P(R_i \cup R_j) = FALSE \;\; for \; i \neq j$$

Region Splitting and Merging

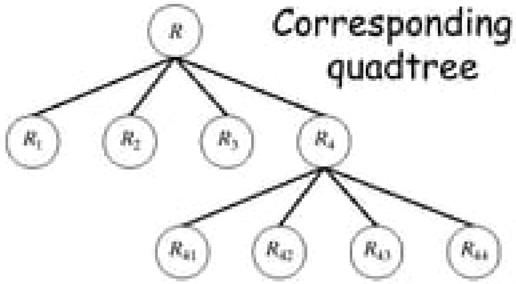
- Condition (a) indicates that the segmentation must be complete; that is, every pixel must be in a region.
- Condition (b) requires that points in a region must be connected in some predefined sense.
- Condition (c) indicates that the regions must be disjoint.
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region—for example $P(R_i)$ = TRUE if all pixels in R_i, have the same gray level.
- Condition (e) indicates that regions R_i, and R_j are different in the sense of predicate P.

Region Splitting and Merging

- The approach is to segment R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i, $P(R_i)$ = TRUE. If only splitting were used, the final partition likely would contain adjacent regions with identical properties. This drawback may be remedied by allowing merging, as well as splitting.
- The procedure is to start with the entire region. If P(R) = FALSE, divide the image into quadrants. If P is FALSE for any quadrant, subdivide that quadrant into subquadrants, and so on. This process results in a so-called **quadtree**.



Splitting image



Corresponding quadtree

Region Splitting and Merging

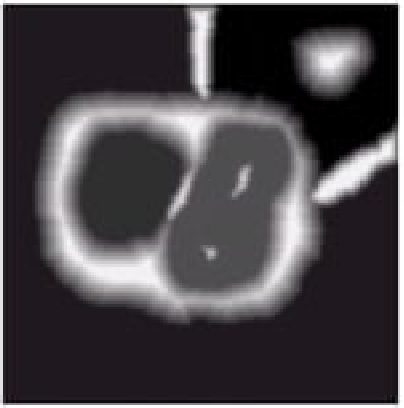
- While splitting two adjacent regions R_j and R_k are merged only if $P(R_j \cup R_k) = TRUE$.
- When no further merging or splitting is possible, the procedure is stopped.

Example for segmentation by region splitting and merging

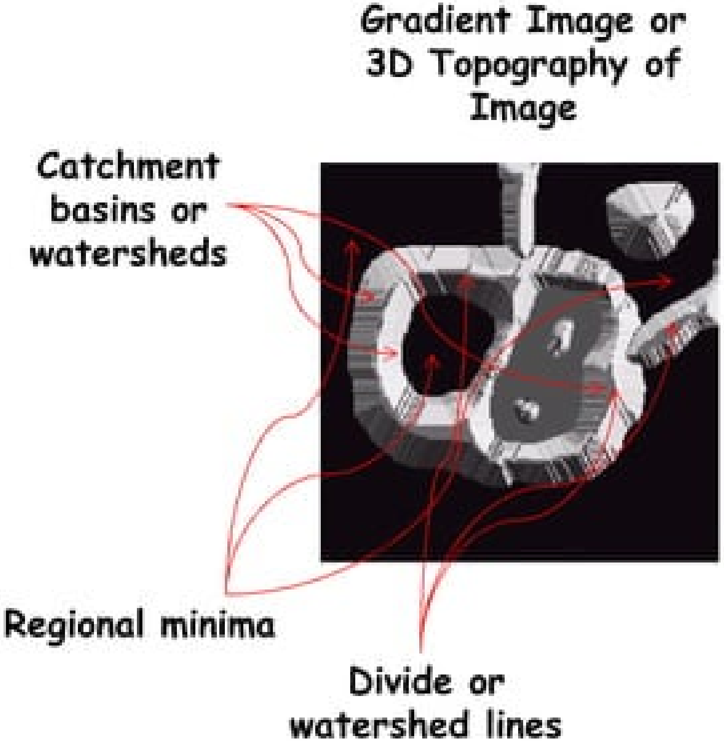
Segmentation by Morphological Watershed – Basic Concepts

- The morphological watershed is a segmentation technique applied to the gradient of the image, rather than the image itself, using the morphological tools to extract uniform objects. The gradient images are also termed as the topographic images.
- The gradient operation enhances the areas of abrupt gray-level changes such as edges and diminishes the smooth areas such as regions of objects in the image, leaving the edges look like crests and the objects (relatively smooth areas) as basins. This results in a 3D topography of the image.
- The morphological watershed segmentation algorithm basically searches the gradient image for the following three types of points:
  - Points belonging to **regional minima**
  - Points belonging to **catchment basins** or **watersheds** where a drop of water would definitely fall to a single minimum
  - Points belonging to **divide lines** or **watershed lines** where a drop of water would more likely fall to more than one minimum
- The principle objective of the segmentation algorithm is to find the third set of points i.e., the watershed lines.

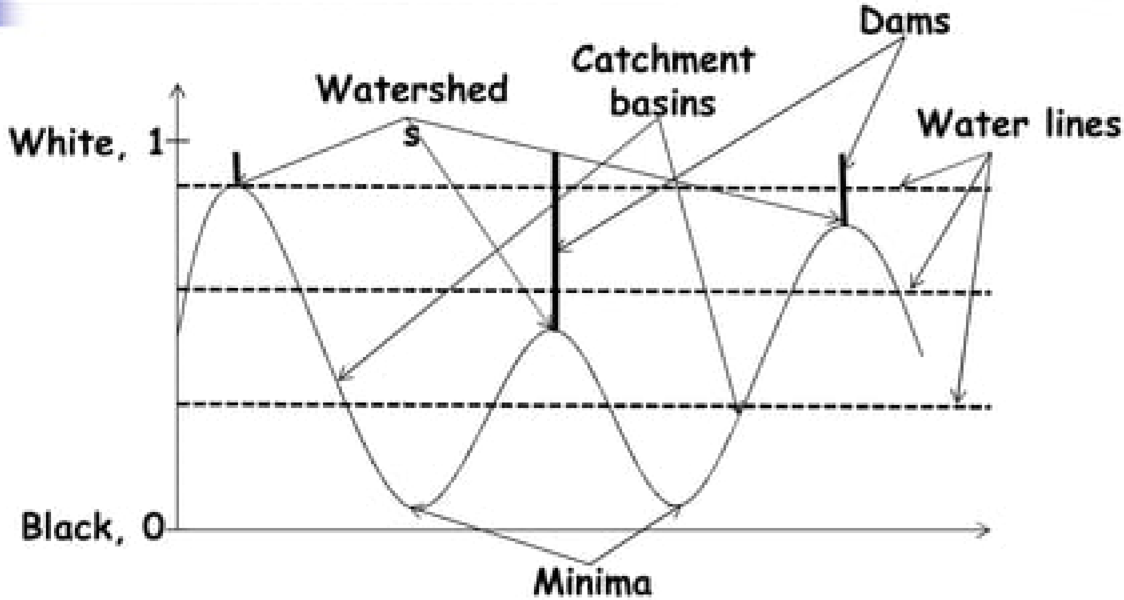
Segmentation by Morphological Watershed – Basic Concepts



Original Image



Segmentation by Morphological Watershed – Basic Concepts



One dimensional, cross-sectional view of minima, catchment basins, watersheds and dams

Segmentation by Morphological Watershed – Basic Concepts

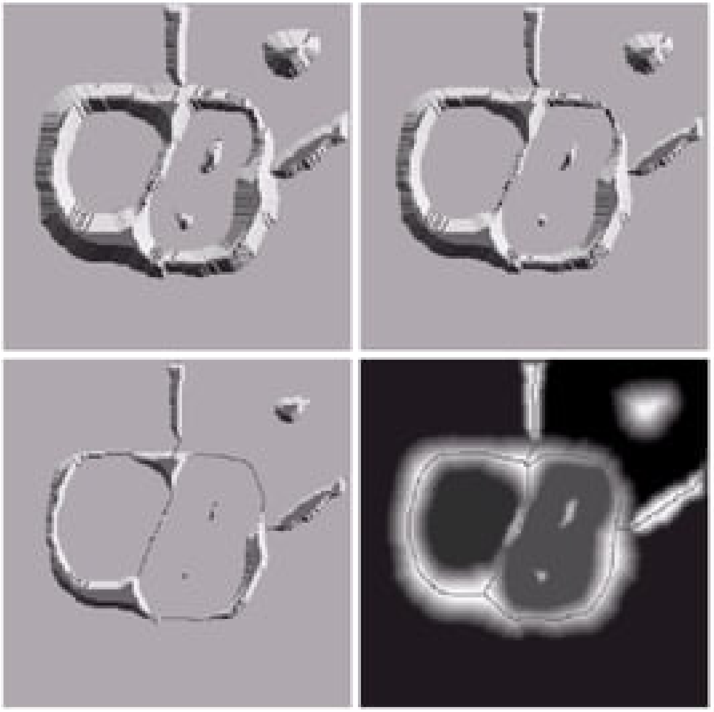
- The basic principle of the watershed segmentation algorithm is as follows.
  - Holes are pierced in all regional minima.
  - Then the topography is flooded gradually with water through these holes at a constant rate.
  - As the water-level continues to rise, it will start to overflow from one catchment basin to another.
  - Finally dams are built to prevent the water in different catchment basins from merging.
  - These dams are the watershed lines which are the desired boundaries for the segmentation purpose. These watershed lines form a connected path.
- In order to prevent the water from spilling out through the edges of the structure, the height of the dams must be greater than the highest possible 'mountain' which is determined by the highest possible gray level value in the image.

Segmentation by Morphological Watershed - Basic Concepts

a b
c d

(a) Original image
(b) Topographic view
(c)-(d) Two stages of flooding

Segmentation by Morphological Watershed – Basic Concepts



a b

c d

(a) Further flooding
(b) Water merging from two basins (a short dam being built between them)
(c) Longer dams being built
(d) Final watershed (segmentation) lines

Segmentation by Morphological Watershed – Dam Construction

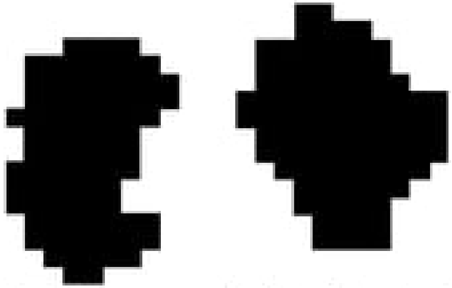


Fig (a) Two partially flooded catchment basins at stage n-1

1	1	1
1	1	1
1	1	1

Fig (c) Structuring element

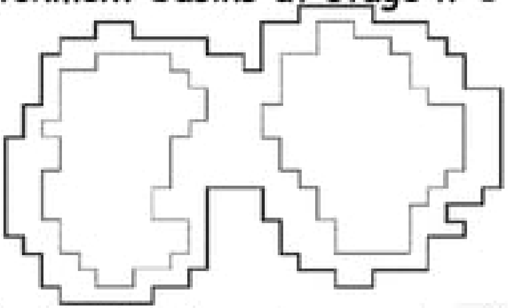
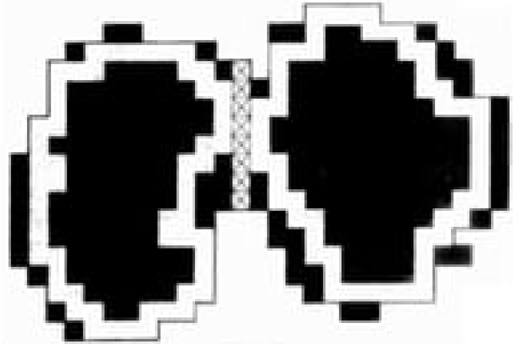


Fig (b) Flooding stage n showing merging of water from two basins



First dilation Second dilation

Dam points

Fig (d) Result of dilation and dam construction

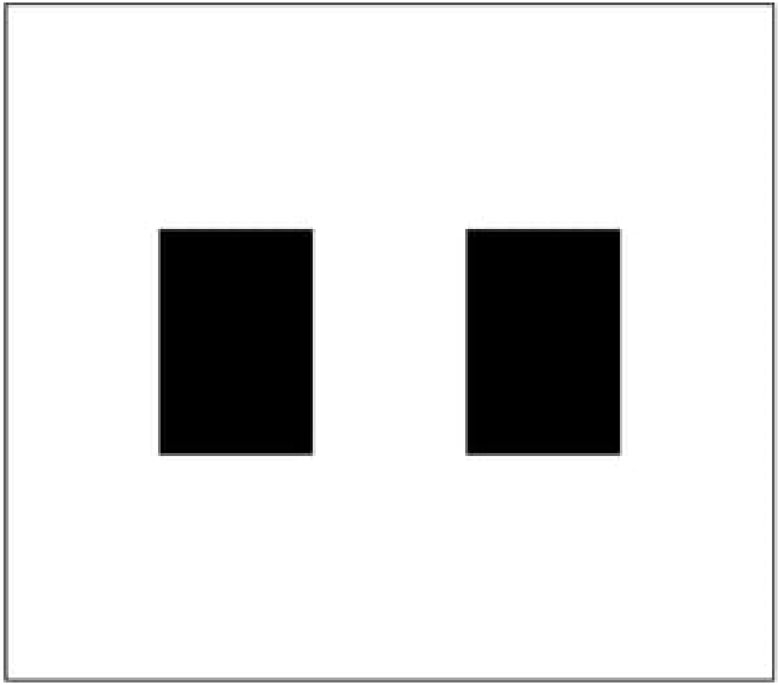
Segmentation by Morphological Watershed – Dam Construction

- *Dam Construction*: The dam is based on binary images and is constructed using morphological dilation.
  - The basics of how to construct a dam are shown in the following figures. Fig (a) shows the portions of two catchment basins at flooding stage (n-1) and Fig (b) shows the result at flooding stage n.
  - There are two connected components in Fig (a) whereas there is only one connected component in Fig (b). This single connected component in Fig (b) encompasses the two connected components in Fig (a) (shown dashed). The fact that two connected components have become a single connected component indicates that the water from two catchment basins has merged at flooding step n.
  - Let (i) M_1 and M_2 be the sets of coordinates of points in the two regional minima, (ii) $C_{n-1}(M_1)$ and $C_{n-1}(M_2)$ be the sets of coordinates of points in the catchment basins associated with these two regional minima at flooding stage n-1, (iii) $C[n-1]$ be the union of these two sets at flooding stage n-1 and (iv) q be the set of coordinates of points in the single connected component at flooding stage n.

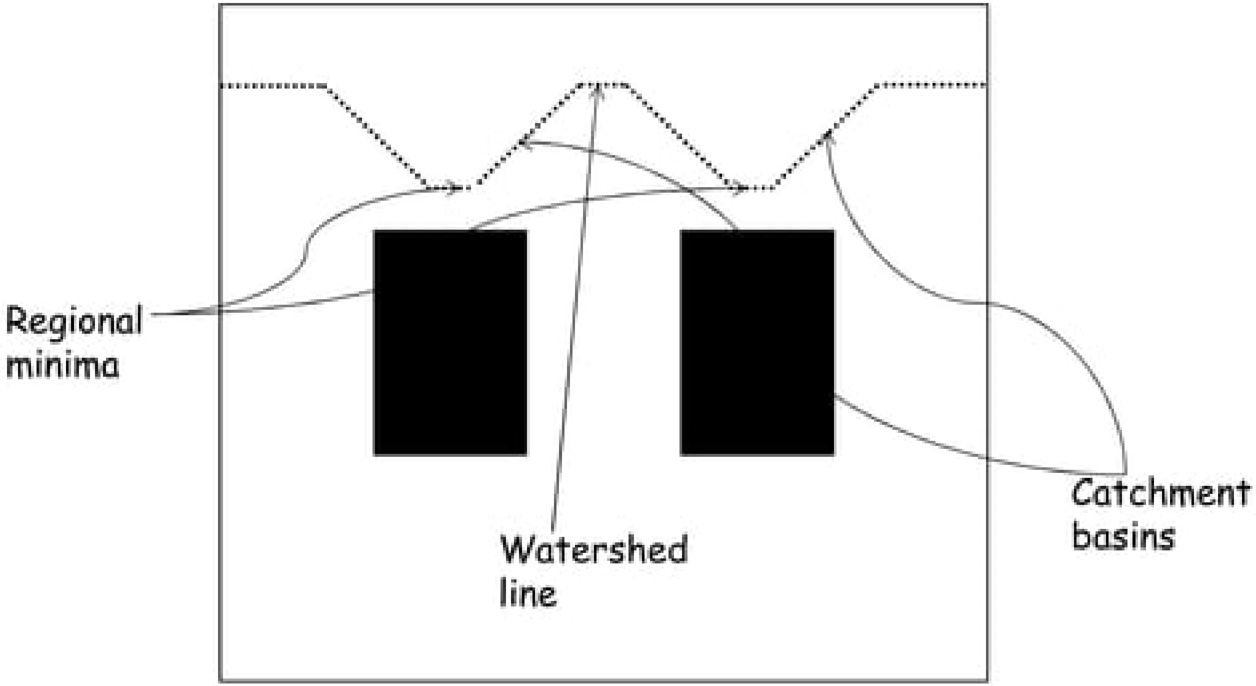
Segmentation by Morphological Watershed – Dam Construction

- The two connected components at flooding stage n-1 are now dilated using the structuring element in Fig (c) with the following two conditions: (1) **dilation is confined to q** i.e., the center of the structuring element is located only at points in q and (2) **dilation is not performed on points that would cause the sets being dilated to merge**.
- First dilation pass expands the boundaries of the two connected components as shown by white squares in Fig (d). All points satisfy the condition (1) but none satisfies the condition (2) during this pass.
- During second dilation pass, only few points satisfy the condition (1) leading to the broken boundaries as shown by black squares. Some points satisfy the condition (2) leading the construction of the dam as shown by the cross-hatched squares.
- The dam construction is completed by setting the dam points to a gray level value higher than the highest gray level value in the image, generally 1 plus the highest gray level value in the image.

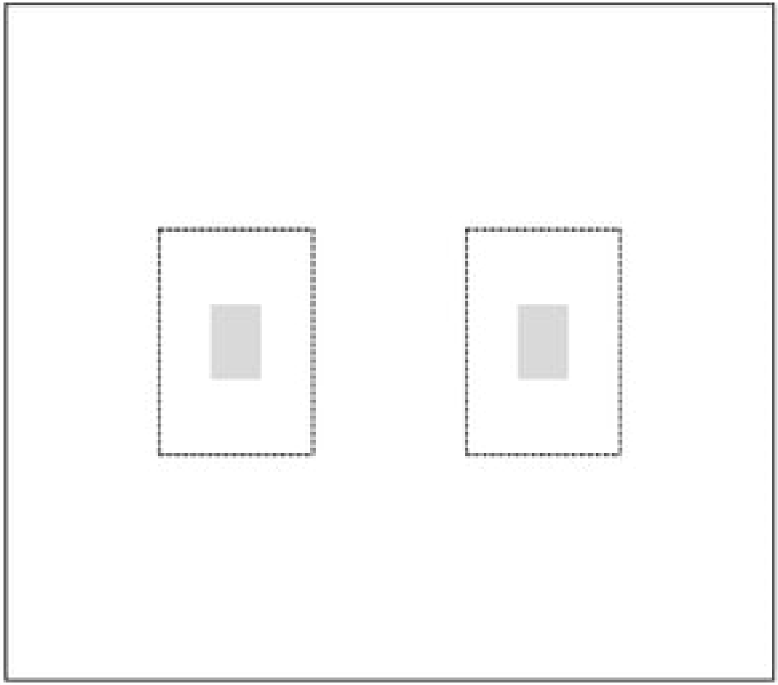
Segmentation by Morphological Watershed – Simple example



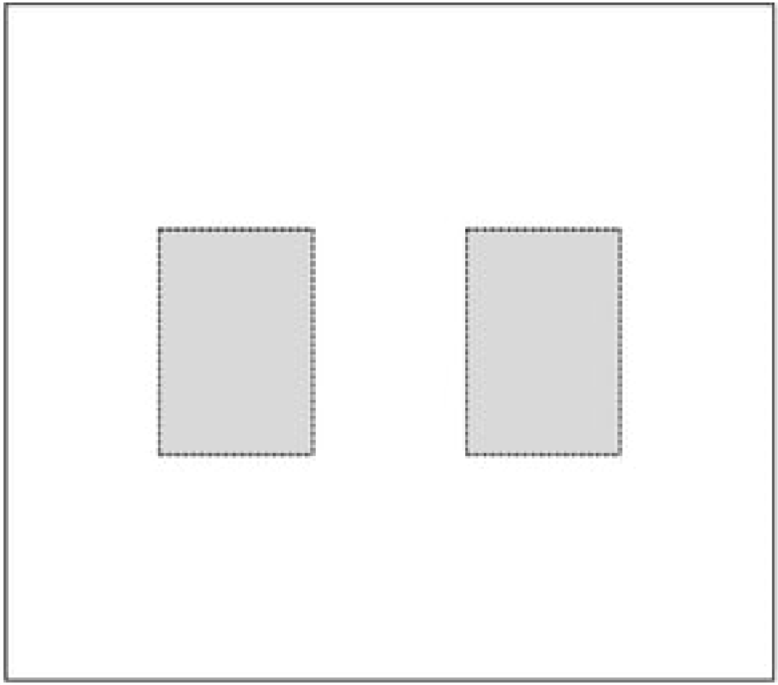
Segmentation by Morphological Watershed – Simple example



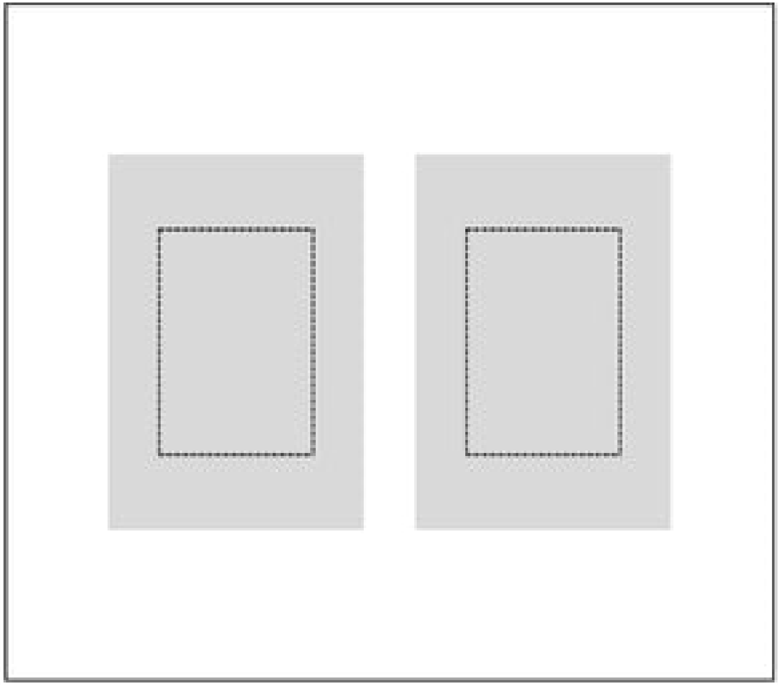
Segmentation by Morphological Watershed – Simple example



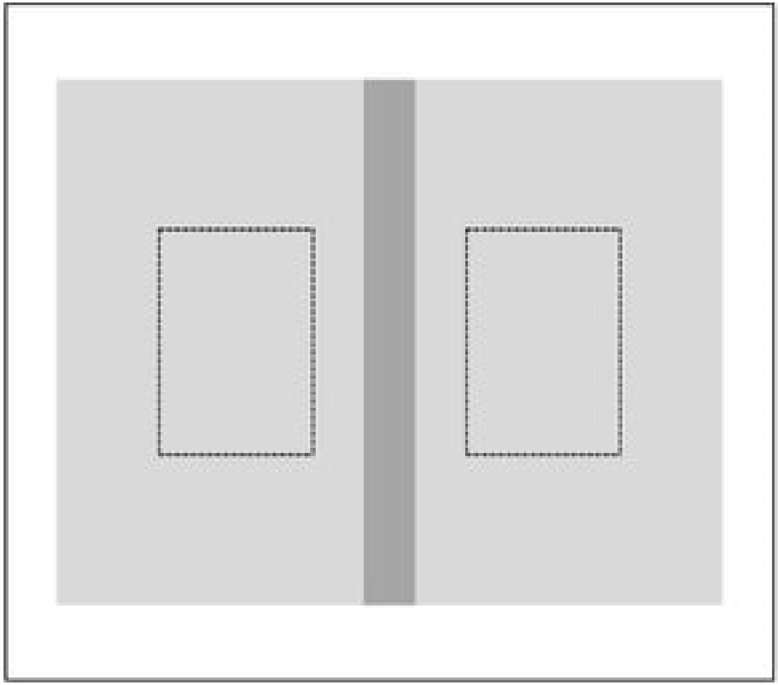
Segmentation by Morphological Watershed – Simple example



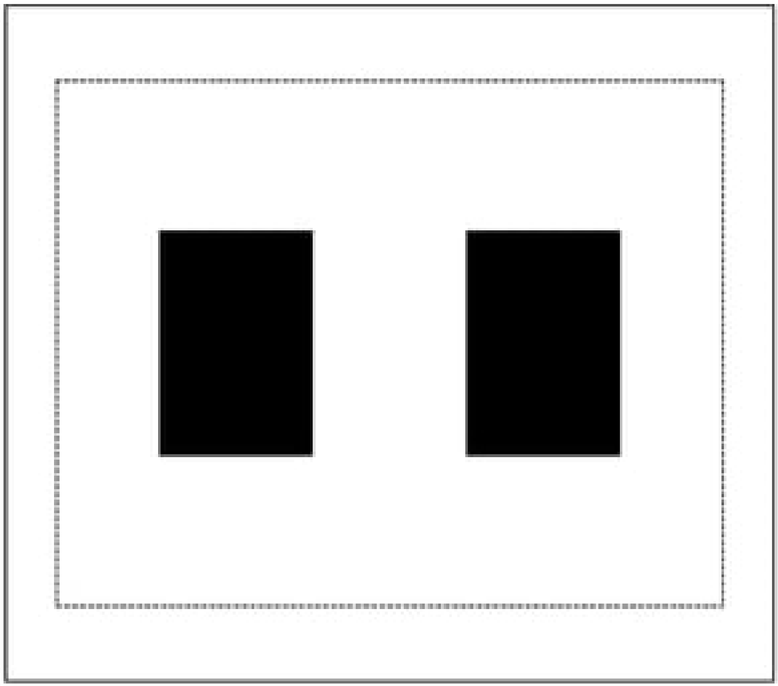
Segmentation by Morphological Watershed – Simple example



Segmentation by Morphological Watershed – Simple example



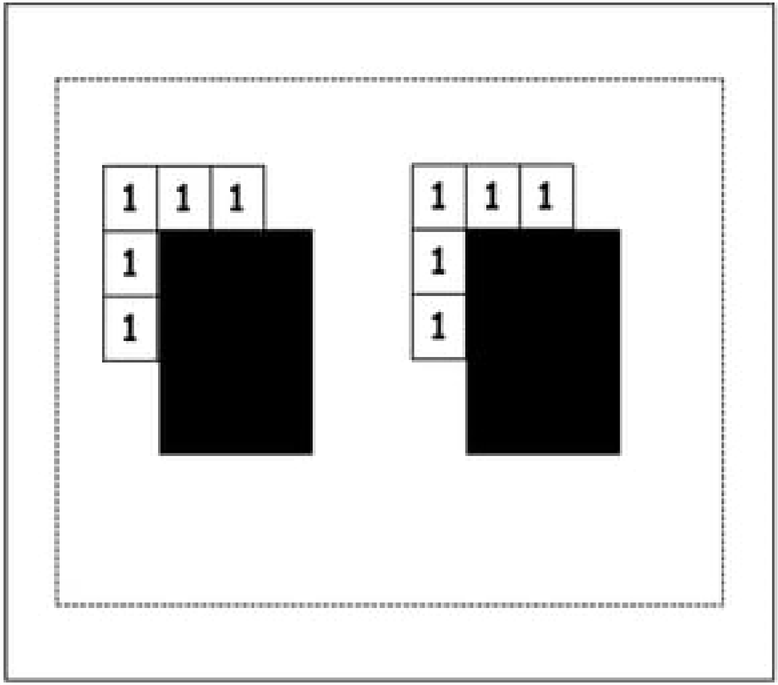
Segmentation by Morphological Watershed – Simple example



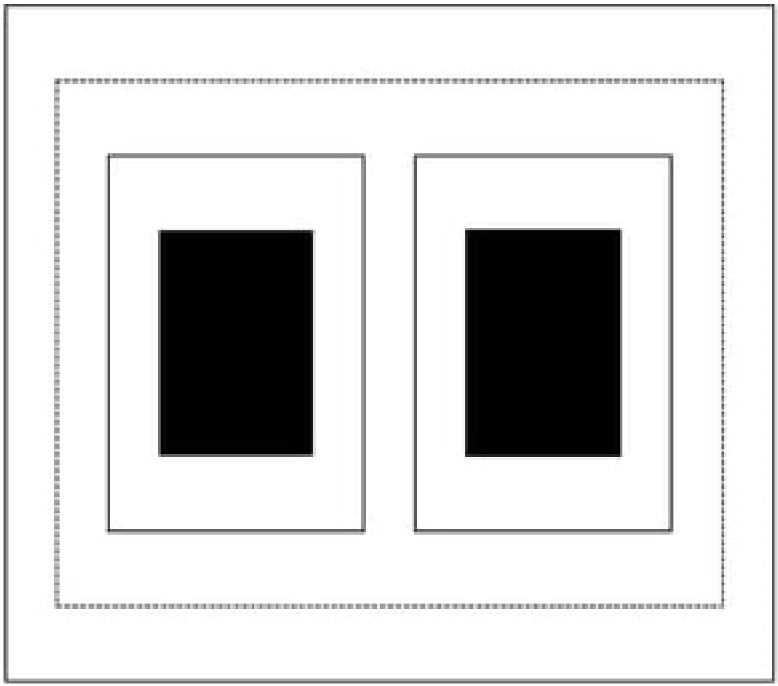
1	1	1
1	1	1
1	1	1

Structuring element

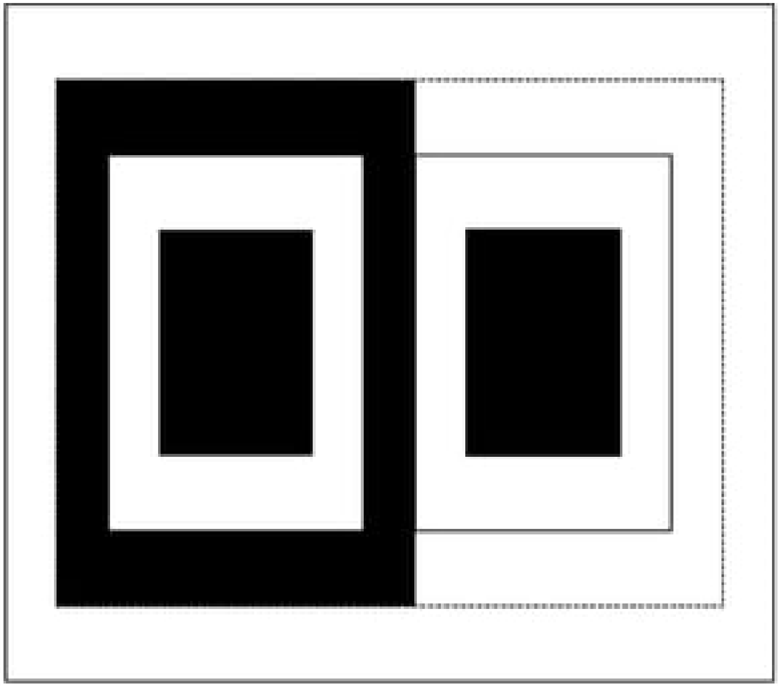
Segmentation by Morphological Watershed – Simple example



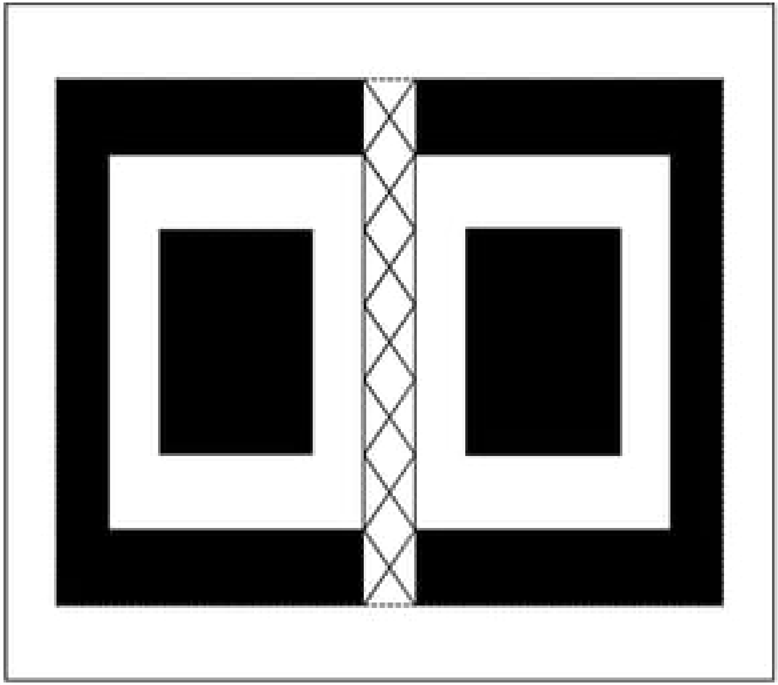
Segmentation by Morphological Watershed - Simple example



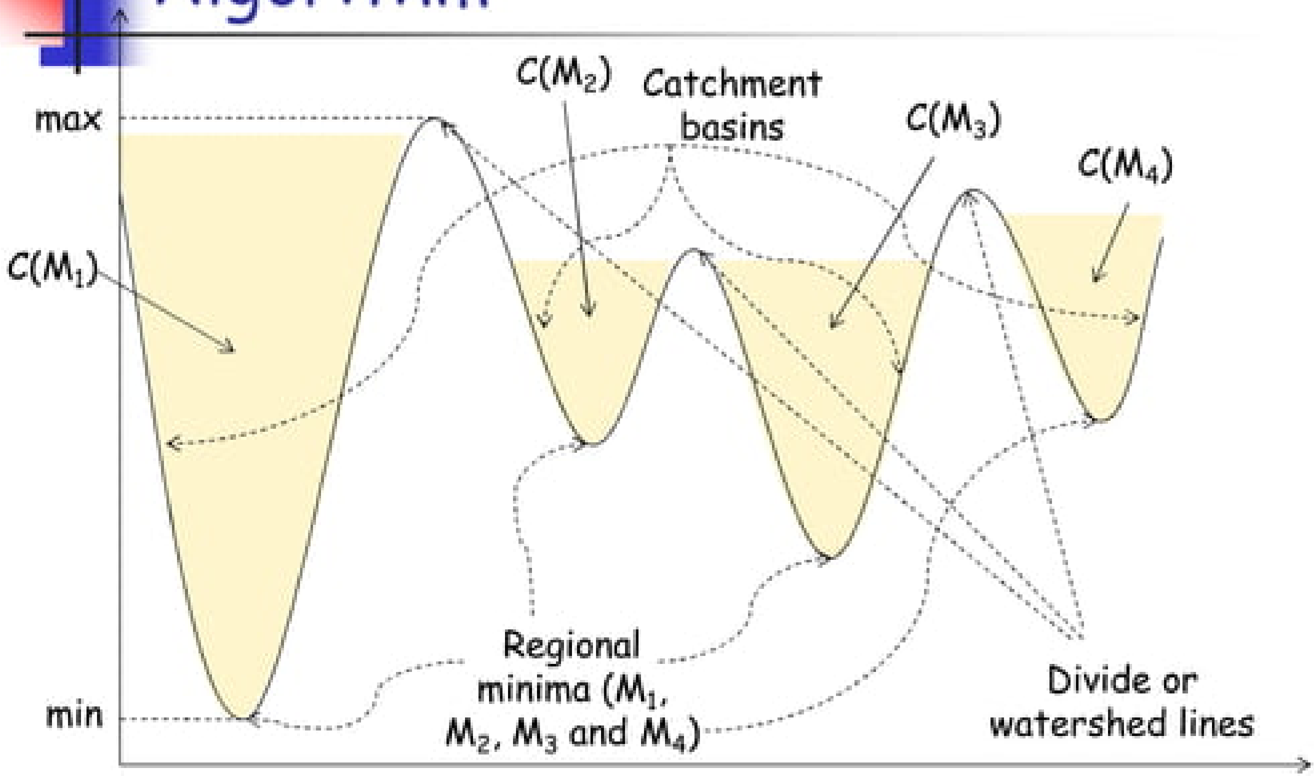
Segmentation by Morphological Watershed - Simple example



Segmentation by Morphological Watershed – Simple example

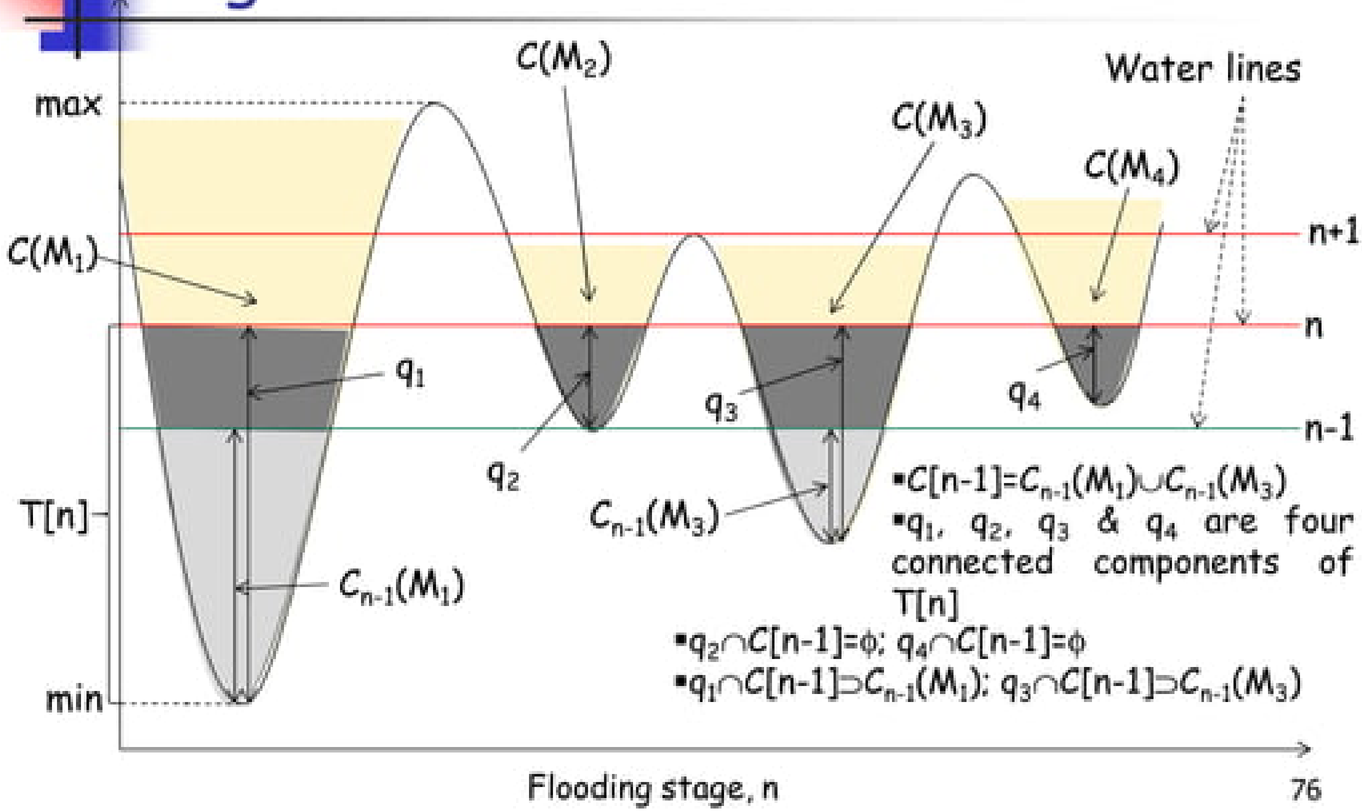


Watershed Segmentation Algorithm

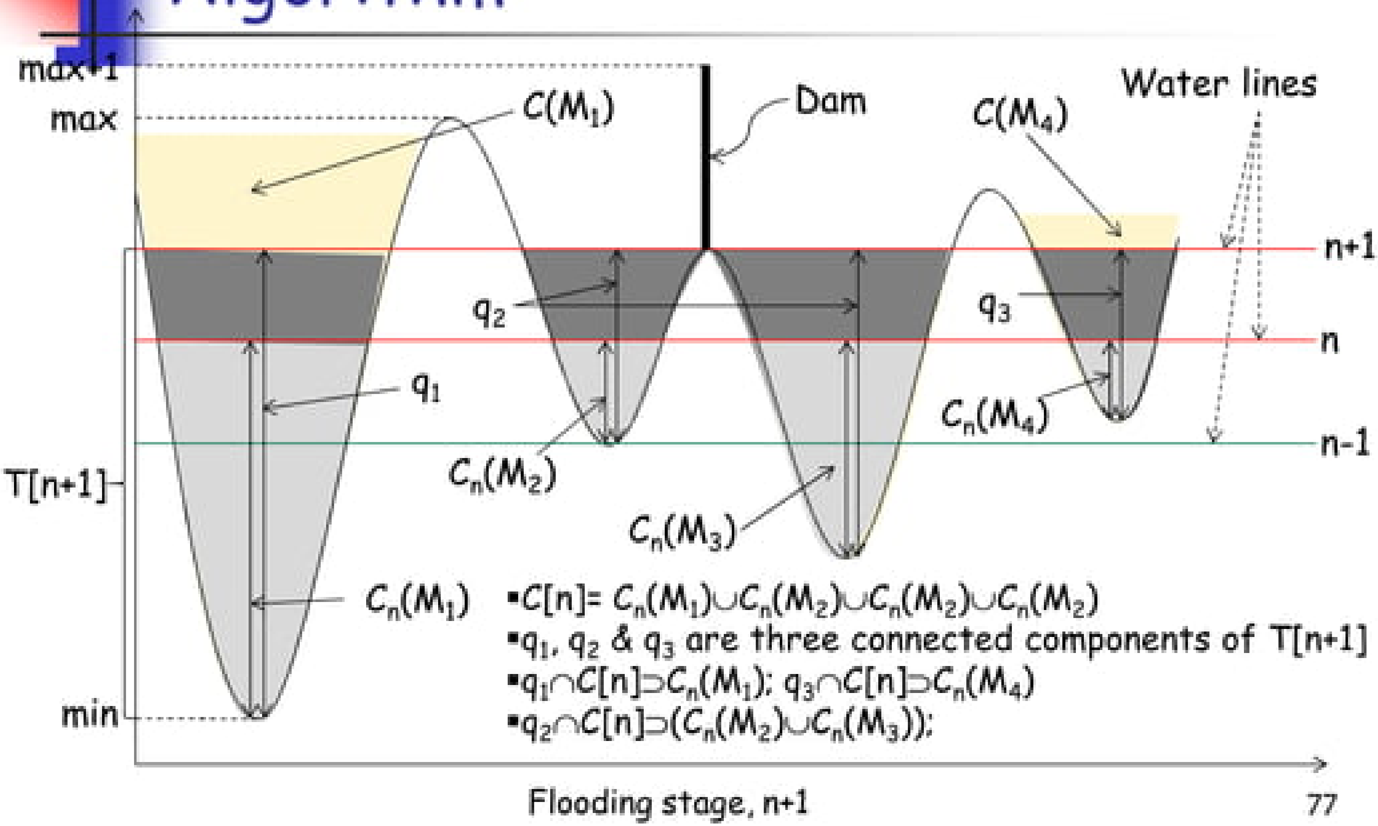


Gradient function showing regional minima, catchment basins & watershed lines

Watershed Segmentation Algorithm



Watershed Segmentation Algorithm



Watershed Segmentation Algorithm

- ***Watershed Segmentation Algorithm:***
  - Let $g(x,y)$ denote the gradient of an image (i.e., the gradient or topographic image).
  - Let $M_1, M_2, \ldots, M_R$ be the sets of coordinates of points in the regional minima of $g(x,y)$.
  - Let $C(M_i)$ be the set of coordinates of points in the catchment basin associated with the regional minimum, M_i.
  - Let $C_n(M_i)$ be the set of coordinates of points flooded in the catchment basin associated with the regional minimum M_i at stage n.
  - Let $C[n]$ be the union of flooded catchment basin portions at stage, n i.e., the union of the sets, $C_n(M_i)$, i=1,2,...,R.

$$C[n] = \bigcup_{i=1}^{R} C_n(M_i)$$

Watershed Segmentation Algorithm

- ***Watershed Segmentation Algorithm:***
  - Let T[n] be the set of coordinates of points, (s,t) for which g(s,t) < n i.e.,

    $$T[n] = \{(s,t) \mid g(s,t) < n\}$$

    Geometrically T[n] is the set of coordinates of points in g(x,y) below the plane, g(x,y)=n
  - The terms, min & max represent the minimum and maximum values of g(x,y).
  - Let Q[n] be the set of connected components in T[n] and q[n] or simply q $\in$ Q[n].

Watershed Segmentation Algorithm

- ***Watershed Segmentation Algorithm:***
  - From the above discussions, the following relations hold true:

$$(1)\, C_n(M_i) = C(M_i) \cap T[n] \Rightarrow C_n(M_i) \in T[n] \,\forall i$$

$$(2)\, C[n] = \bigcup_{i=1}^{R} C_n(M_i) \Rightarrow C[\max+1] = \bigcup_{i=1}^{R} C(M_i)$$

$$(3)\, C[n-1] \in C[n] \Rightarrow C[n] \in T[n] \Rightarrow C[n-1] \in T[n]$$

  - The relation (3) implies that each connected component of $C[n-1]$ is contained in exactly one connected component of $T[n]$.
  - The flooding stage is integrally incremented from n=min+1 to n=max+1.
  - The algorithm begins by setting $C[min+1]=T[min+1]$. The set $C[n]$ at any stage, n is recursively constructed by assuming that $C[n-1]$ has already been constructed and considering the following facts.

Watershed Segmentation Algorithm

- ***Watershed Segmentation Algorithm:***

  (1) $q \cap C[n-1]$ is empty

  (2) $q \cap C[n-1]$ contains only one connected component of $C[n-1]$

  (3) $q \cap C[n-1]$ contains more than one connected components of $C[n-1]$

  - (1) occurs when q encounters a new regional minimum and hence the connected component, q is incorporated into $C[n-1]$ to form $C[n]$.
  - (2) occurs when q lies in the catchment basin of some regional minimum and hence the connected component, q is incorporated into $C[n-1]$ to form $C[n]$.
  - (3) occurs when q encounters all or part of the edge separating two or more catchment basins causing the water from different catchment basins to merge. Now, to prevent this merging, the dam is constructed by dilating $q \cap C[n-1]$ with a 3×3 structuring element of 1's and confining the dilation to q.

Unit 5

IMAGE COMPRESSION

Image Compression

- Image compression is needed to reduce the storage requirement and to increase the transmission efficiency such as transmission rate and noise immunity.

- Image compression refers to the process of reducing the amount of data required to represent a given digital image - removal of redundant data (data redundancy).

- There are three types of data redundancies, namely, (i) the **interpixel redundancy**, (ii) the **psychovisual redundancy** and (iii) the **coding redundancy**.

Data Redundancy

- Various amount of data may be used to represent the same amount of information. In such case if two sets of data, one being large and another being small, represent the same information, then the large amount of data is said to contain data that either provide no relevant information or simply repeat the same information. This is known as data redundancy.
  - Measures of data redundancy: **Relative Data Redundancy** and **Compression Ratio**.
  - Let n_1 and n_2 be the number of information-carrying units in two data sets that represent the same information. Then the relative data redundancy, R_D is defined as

$R_D = 1 - \frac{1}{C_R}$ where $C_R = \frac{n_1}{n_2}$ is called the compression ratio.

Data Redundancy

(i) If $n_2=n_1$, then $C_R=1$ and $R_D=0$ (no redundant data in n_2)

(ii) If $n_2<<n_2$, then $C_R=\infty$ and $R_D=1$ (highly redundant data in n_2)

(iii) If $n_2>>n_2$, then $C_R=0$ and $R_D=-\infty$ (more irrelevant data in n_2)

Hence C_R and R_D lie in the open intervals, $(0,\infty)$ and $(-\infty,1)$, respectively.

- **Interpixel Redundancy**
  - If, in an image, the value of a pixel can be reasonably predicted from the value of its neighbours, then the image is said to contain interpixel redundancy.
  - The correlation statistics such as auto correlation coefficients are used to measure the interpixel redundancy.
- **Psychovisual Redundancy**
  - If an image contains certain information that is less relative important than the other information in normal visual processing, then the image is said to contain psychovisual redundancy.

Data Redundancy

- **Coding Redundancy**
  - If the gray levels of an image are coded in a way that uses more code symbols than absolutely necessary to represent each gray level, the resulting image is said to contain coding redundancy.
  - The histogram of the image is a useful tool to provide means of reducing the coding redundancy. Let r_k be a discrete random variable in the interval [0,1] representing the set of gray levels in an image. Let n_k represent the number of pixels with the gray level r_k in the image. Then the probability of occurrence of a gray level, r_k is defined as

$$p_r(r_k) = \frac{n_k}{n}, \mathrm{k} = 0,1,2,...,\mathrm{L}-1$$

    If the number of bits required to represent each gray level, r_k is $l(r_k)$ then the average code length is

$$L_{av} = \sum_{k=0}^{L-1} p_r(r_k) l(r_k)$$

Data Redundancy

- Coding Redundancy Example: An 8-level image has a gray-level distribution as shown in Table 4.1. Compute the percentage of redundancy in Code 1.

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

Table 4.1

Data Redundancy

- The average length of the code for Code 1

$$L_{1av} = \sum_{k=0}^{7} p_r(r_k) l(r_k) = 3 \sum_{k=0}^{7} p_r(r_k) = 3$$

- The average length of the code for Code 2

$$L_{2av} = \sum_{k=0}^{7} p_r(r_k) l(r_k) = \sum_{k=0}^{7} [(0.19 \times 2) + (0.25 \times 2) + (0.21 \times 2) + (0.25 \times 2) +$$
$$+ (0.16 \times 3) + (0.08 \times 4) + (0.06 \times 5) + (0.03 \times 6) + (0.02 \times 6)] = 2.7$$

- Hence the compression ratio

$$C_R = \frac{L_{1av}}{L_{2av}} = \frac{3}{2.7} = 1.11.$$

Data Redundancy

- Hence the relative data redundancy

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

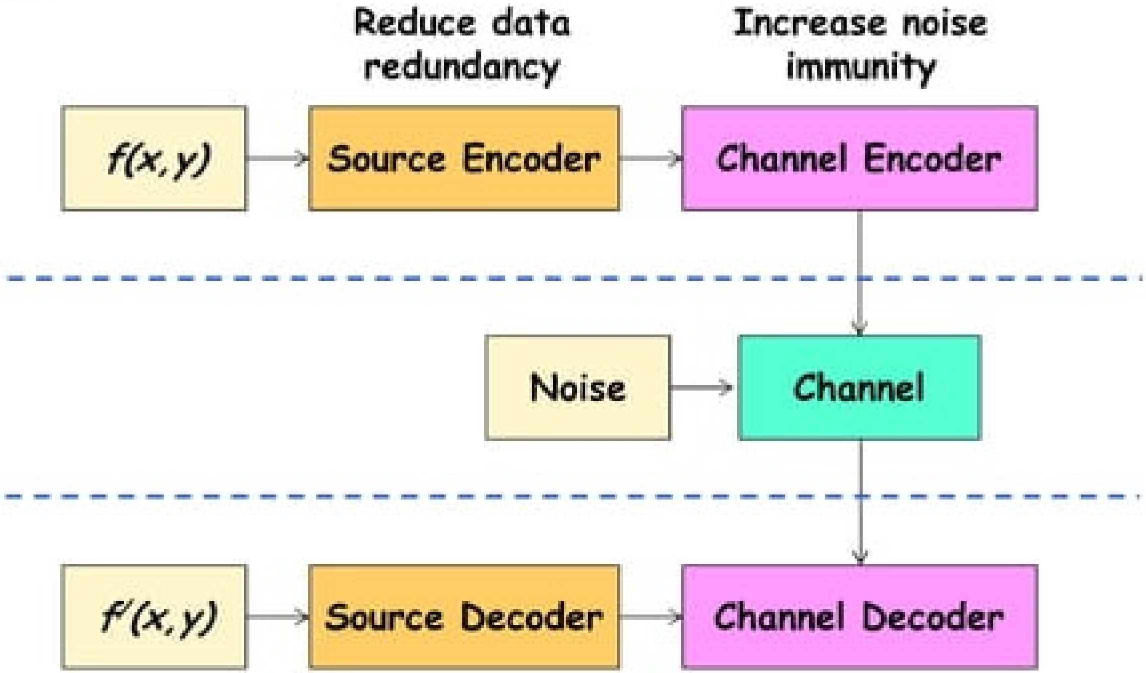
- Hence the percentage of data redundancy

$$R_D \times 100 = 9.9\%.$$

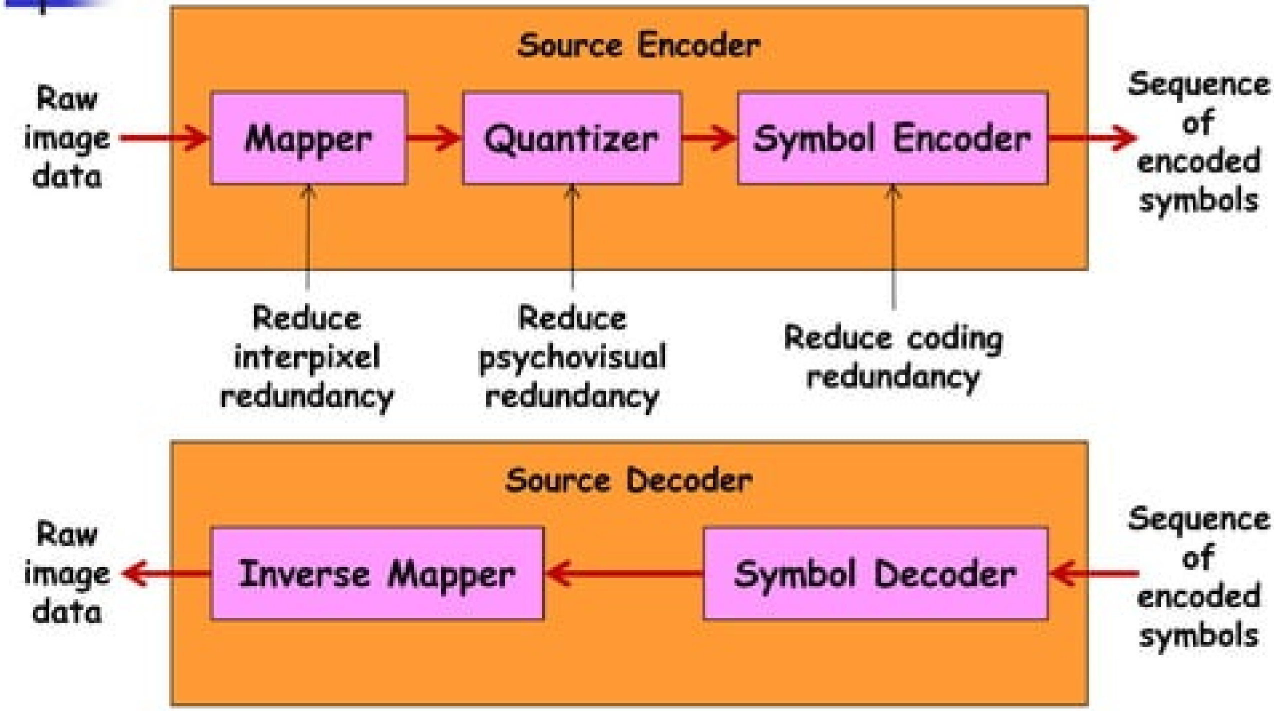
Compression Techniques

- Compression techniques are broadly classified into two: Lossless Compression & Lossy Compression.
- Lossless compression techniques: *Compression techniques where perfect (lossless) reconstruction is possible.*
  - Variable length coding
  - LZW coding
  - Bit-plane coding
  - Predictive coding-DPCM
- Lossy compression techniques: *Compression techniques where perfect (lossless) reconstruction is not possible.*
  - Transform coding
    - Wavelet coding
    - Basics of image compression standards: JPEG
    - Basics of vector quantization: MPEG

Compression Model



Source Encoder-Decoder Model



Variable Length Coding: Huffman Coding

- Each symbol is encoded with different code lengths.
- Coding redundancy is removed.
- Huffman coding is the most popular variable length coding technique. The Huffman coding involves two steps: (i) To create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. (ii) To code each reduced source starting with the smallest source and working back to the original source.

  The Huffman coding encodes the source such that the lowest the probability of the symbol is, the longest is the code length and vice versa.

Variable Length Coding: Huffman Coding

- Example: Encode the following source using the Huffman coding.

Symbol	Probability
a_1	0.1
a_2	0.4
a_3	0.06
a_4	0.1
a_5	0.04
a_6	0.3

Variable Length Coding: Huffman Coding

- Arrange source symbols in descending order of their probabilities.

Symbol	Probability
a_2	0.4
a_6	0.3
a_1	0.1
a_4	0.1
a_3	0.06
a_5	0.04

Variable Length Coding: Huffman Coding

- Create series of reduced sources by combining the lowest two probability symbols into one until a reduced source with only two symbols is reached.

Original Source		Source Reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Variable Length Coding: Huffman Coding

- Code each reduced source starting with the smallest source and working back to the original source.

Original Source			Source Reduction			
Symbol	Prob.	Code	1	2	3	4
a_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a_6	0.3	11	0.3 00	0.3 00	0.3 00	0.4 1
a_1	0.1	011	0.1 011	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

Variable Length Coding: Huffman Coding

- Some interesting properties of Huffman coding:
  - After the coding procedure is accomplished, the source symbols is encoded or decoded one at a time with a look-up table.
  - This is a ***block code*** because each source symbol is coded into a fixed sequence of code symbols.
  - This is ***instantaneous*** because each code word in a string of code symbols can be decoded without referencing to succeeding symbols.
  - This is ***uniquely decodable*** because any string of code symbols can be decoded in only one way.

Variable Length Coding: Huffman Coding

- Example: Given a string of source symbols, $a_5a_3a_4$

Look-up Table	
Symbol	Code
a_2	1
a_6	11
a_1	011
a_4	0100
a_3	01010
a_5	01011

Encoding: From the look-up table

$$a_5a_3a_4 \Rightarrow 01011 \vdots 01010 \vdots 0100$$

- Block code
- Look-up table
- One at a time

Decoding: From the look-up table

$$01011 \vdots 01010 \vdots 0100 \Rightarrow a_4a_3a_5$$

- Instantaneous
- Uniquely decodable

Variable Length Coding: Other Optimal Techniques

- Other optimal variable length coding techniques:
- Truncated Huffman code: Only the most probable k symbols of source are encoded.
- B-code: Each code word is made up of continuation bits and information bits. The continuation bit separates individual code words by alternating between 0 and 1 for successive symbols in a string. This coding is optimal if the symbol probability is

$p(a_i) = ci^{-\beta}$, for any symbol a_i, a positive constant β and

$c = \frac{1}{\sum_{i=0}^{M} i^{-\beta}}$, M being total number of symbols

Variable Length Coding: Other Optimal Techniques

- Shift codes (Binary shift and Huffman shift): Arrange source symbols in ascending order of their probabilities, Divide total number of symbols into symbol blocks of equal size, Code individual symbols in a block identically and Add special shift-up and/or shift-down symbols to identify each block.

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	100000	C10C00	111 101	00 1110
a_{14}	0.03	01101	01111	100001	C10C01	111 110	00 1111
Block 3							
a_{15}	0.03	01110	01100	100010	C10C10	111 111 000	00 00 10
a_{16}	0.02	01111	010000	100011	C10C11	111 111 001	00 00 11
a_{17}	0.02	10000	010001	100100	C11C00	111 111 010	00 00 110
a_{18}	0.02	10001	001010	100101	C11C01	111 111 011	00 00 100
a_{19}	0.02	10010	001011	100110	C11C10	111 111 100	00 00 101
a_{20}	0.02	10011	011010	100111	C11C11	111 111 101	00 00 1110
a_{21}	0.01	10100	011011	101000	C00C00C00	111 111 110	00 00 1111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

Run Length Coding

- Using the lengths of runs of 1s or 0s in a binary image: Type (1) Specifying the starting position and the length of runs of 1s in each row or Type (2) Specifying lengths of runs in each row starting with the length of runs of 1s.

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

Type 1.

Row 1: (1,3),(7,2),(12,4),(17,2),(20,3)
Row 2: (5,13) (19,4)
Row 3: (1,3) (17,6)

Type 2.

Row 1: 3,3,2,3,4,1,2,1,3
Row 2: 0,4,13,1,4
Row 3: 3,13,6

Run Length Coding

- Run lengths can further be encoded using variable length coding e.g., Huffman coding for more compression.
- Let a_k, k=0,1,2,...,M be the number of runs of 0s of length k. Then let

$$H_0 = -\sum_{k=1}^{M} a_k \log a_k \text{ and } L_0 = \sum_{k=1}^{M} k b_k$$

  be entropy associated with a_k, k=0,1,2,...M and average length of runs of 0s.
- Similarly, for the lengths of runs of 1s

$$H_1 = -\sum_{k=1}^{M} b_k \log b_k \text{ and } L_1 = \sum_{k=1}^{M} k b_k$$

  where b_k, k=0,1,2,...M be the number of runs of 1s of length k.
- Hence the approximate run length entropy of the image is defined as

$$H_{RL} = \frac{(H_0 + H_1)}{(L_0 + L_1)}$$

  which provides the average number of bits required to code the run lengths.

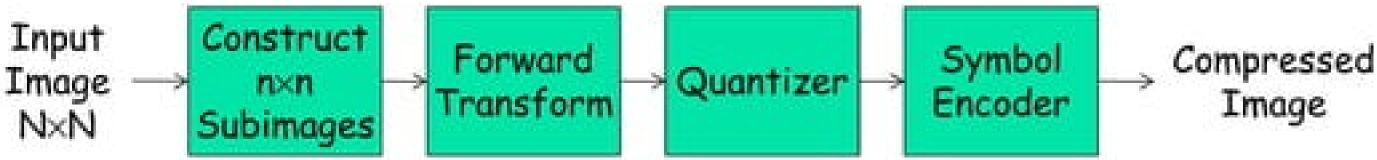
Arithmetic Coding

- Assignment: Refer to

?

Transform Coding

- In transform coding, a reversible linear transform such as Fourier transform, DCT etc is used to map the input image into a set of transform coefficients. Then these transform coefficients are quantized and coded.



Transform Coding

- Transform coding consists of decomposition of image to smaller subimages, transformation, quantization and symbol-encoding. The decoding process consists of symbol-decoding, inverse transformation and finally merging of subimages.
- Subimage decomposition and transformation decorrelate the image pixels or packs as much as information as possible into smaller number of transform coefficients.
- Quantization then selectively eliminates or more coarsely quantizes the coefficients that carry least information with very little image distortion.
- The quantized transform coefficients can then be encoded using a suitable variable length coding such as Huffman coding.
- **Transform Selection:** Transform is selected based on the following desirable characteristics: (i) Content decorrelation: packing the most amount of energy in the fewest number of coefficients (energy compaction), (ii) Content-Independent basis functions and (iii) Fast implementation or computational complexity.

Transform Coding

- **Subimage Size Selection:** Subimage size is generally selected to be a positive integer power of 2 as this simplifies the computation of transform. Typical subimage sizes are 8×8 and 16×16.
- **Bit allocation:** Quantization is to retain only a fraction of the transform coefficients. There are two basic methods:
  - (i) Zonal coding: Retaining only those transform coefficients with large variance and encoding them using a variable length code.
  - (ii) Threshold coding: Retaining only those transform coefficients with large magnitude and encoding them using a variable length code.

Transform Coding

- Zonal coding: Steps:

  (1) Calculate the variance of each coefficient,

  (2) Arrange the coefficients in the ascending order of their variances,

  (3) Retain only first K large variance coefficients and

  (4) Encode each of the retained coefficients using the variable length coding technique with no. of bits proportional to its variance.

- Threshold coding: Steps:

  (1) Arrange the coefficients in the ascending order of their magnitude,

  (2) Retain only first K large magnitude coefficients and

  (3) Encode each of the retained coefficients using the variable length coding technique.

Transform Coding

- Available transformations:
  - Karhunen-Loeve Transform (KLT)
    - Basis functions are content-dependent
    - Computationally complex
  - Discrete Fourier Transform (DFT/FFT)
    - Real and Imaginary components (Amplitude and Phase)
    - Fast Algorithms
  - Discrete Cosine Transform (DCT)
    - Real transformation
    - Fast algorithm
    - Best energy packing property
  - Walsh-Hadamard Transform (WHT)
    - Poor energy packing property
    - Simple hardware implementation, low-cost and fast

Vector Quantization

- Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. Instead of encoding each pixel, a vector representing a group of pixels is encoded.

Compression:

- Given an image, a codebook containing a set of codevectors is designed either locally or globally.
- The image is partitioned into a set of non-overlapping imagevectors.
- (A codevector or an imagevector is a set of elements representing a group of pixels, e.g., a block of 4×4 pixels.)
- For each imagevector, the codevector closest to it is found from the codebook using some distance measure e.g., Euclidian distance.
- The index of the matching codevector is found and encoded.

Vector Quantization

<u>Decompression:</u>

- The index is decoded.
- The codevector at the decoded index is retrieved.
- The image is reconstructed by combining the retrieved codevectors.

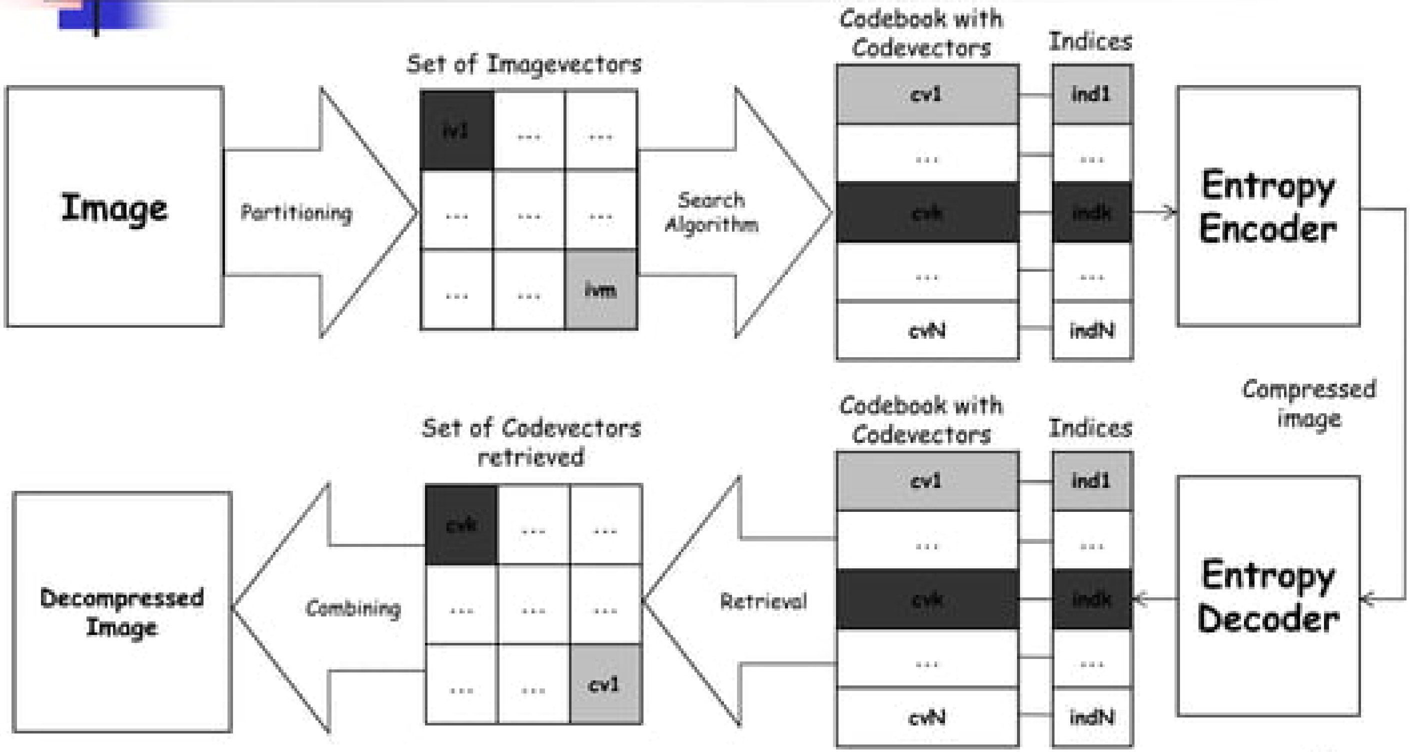
<u>Advantages of VQ:</u>

- More choices
- High compression Ratio
- High Performance

<u>Disadvantages (or difficulties) of VQ:</u>

- Computationally complex

Vector Quantization



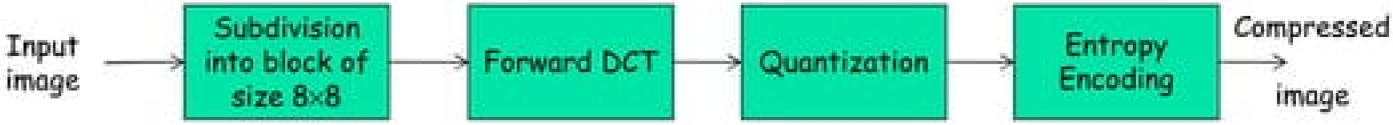
Vector Quantization

<u>Design of Codebook:</u>

- Difficult part
- Local or global
- Local Codebook:
  - One codebook for each image
  - Higher performance
  - Higher computational overhead
  - Necessity of transmission of codebook
- Global codebook:
  - One codebook for a class of images
  - Lower computational overhead
  - No need for transmission of codebook
  - Lower performance

Compression Standards: JPEG

- JPEG stands for Joint Photographic Experts Group.
- JPEG is a standard for still image compression.

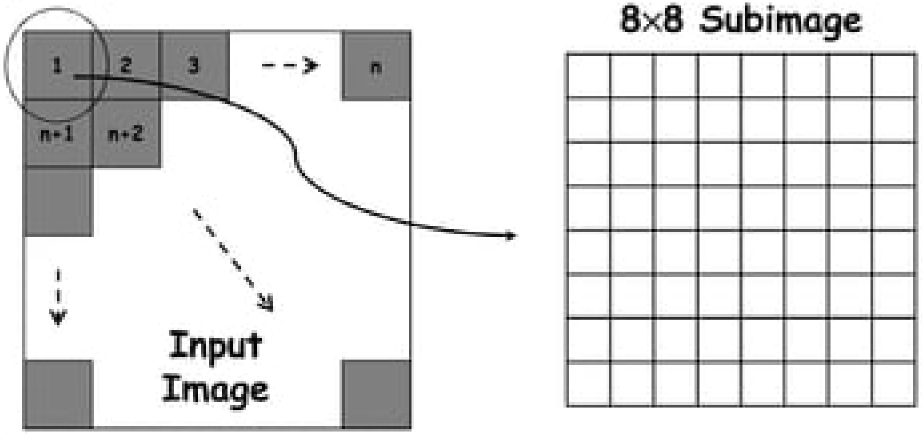


JPEG Encoding

JPEG Decoding

Compression Standards: JPEG

- The input image is divided into subimages of size 8×8 pixels.



- The pixel values in a subimage are generally positive falling in the range [0,255]. These pixel values are level-shifted to fall in the range [-128,127] by subtracting 128 from each pixel value. This reduces the dynamic rage overhead in the DCT processing.

Compression Standards: JPEG

- Then the DCT of the level-shifted subimage is taken.
- The DCT coefficient matrix is then quantized by dividing the coefficient matrix by a **quantization matrix** on an element-by-element basis and rounding the result. The quantization matrix is a predefined matrix based on the psychovisual effect.

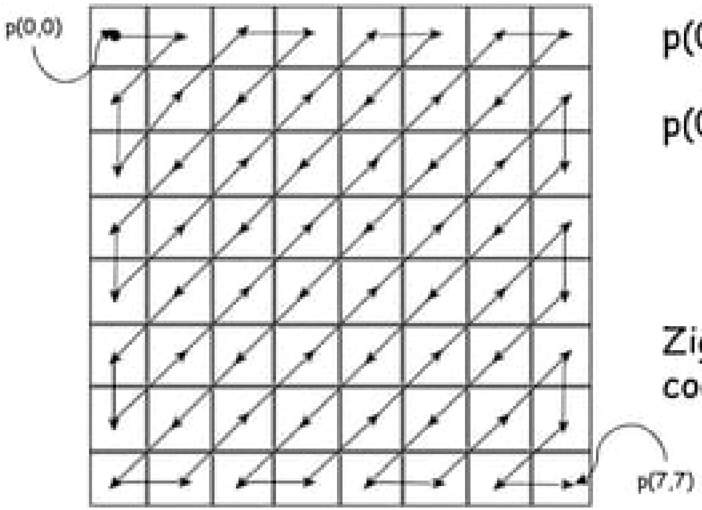
$$
\underbrace{\begin{bmatrix} p(0,0) & p(0,1) & \cdots & p(0,7) \\ p(1,0) & p(1,1) & \cdots & p(1,7) \\ \vdots & \vdots & \vdots & \vdots \\ p(7,0) & p(7,1) & \cdots & p(7,7) \end{bmatrix}}_{\text{DCT coefficient matrix}}
\div
\underbrace{\begin{bmatrix} q(0,0) & q(0,1) & \cdots & q(0,7) \\ q(1,0) & q(1,1) & \cdots & q(1,7) \\ \vdots & \vdots & \vdots & \vdots \\ q(7,0) & q(7,1) & \cdots & q(7,7) \end{bmatrix}}_{\text{Quantization matrix}}
=
\underbrace{\begin{bmatrix} p_Q(0,0) & p_Q(0,1) & \cdots & p_Q(0,7) \\ p_Q(1,0) & p_Q(1,1) & \cdots & p_Q(1,7) \\ \vdots & \vdots & \vdots & \vdots \\ p_Q(7,0) & p_Q(7,1) & \cdots & p_Q(7,7) \end{bmatrix}}_{\text{Resultant quantized coefficient matrix}}
$$

where

$$p_Q(i,j) = round\left[\frac{p(i,j)}{q(i,j)}\right]$$

Compression Standards: JPEG

- The elements of the resultant matrix after quantization are reordered in a zig-zag manner starting from the zero-frequency element at the left top-most corner of the matrix to the highest-frequency element at the right bottom-most corner of the matrx.



p(0,0) → DC coefficient

p(0,1) to p(7,7) → AC coefficients

Zig-zag ordering of quantized DCT coefficients.

Compression Standards: JPEG

- Except for the DC coefficient of the first block, the DC coefficients of the remaining blocks are DPCM-coded i.e., only the difference between the DC coefficients of successive blocks is encoded.
- Since the many *AC* coefficients are zero, the run length encoding (RLE) technique is used to code the counts of zeros efficiently. The RLE stores a **skip** and a **value**: The 'skip' is the number of zero coefficients preceding the 'value' and the 'value' is the next non-zero coefficient.
- Then finally these RLEs are encoded using a entropy coding technique e.g., using Huffman coding.
- In the decoding process, the reverse of above steps are carried out.

Compression Standards: JPEG - Example

- An 8×8 subimage is shown below. The subimage has pixel values in the range [0,255].

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Compression Standards: JPEG - Example

- The subimage is level-shifted by subtracting 128 from each pixel value. The resultant has pixel values in the range [-128,127].

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Compression Standards: JPEG - Example

- The DCT coefficients of the level-shifted subimage are calculated.

$$G(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g(x,y)\cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right]\cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)v\right]$$

where

$$0 \le u \le 8 \,\&\, 0 \le v \le 8 \qquad \alpha(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \textit{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \textit{otherwise} \end{cases}$$

$$\begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

Compression Standards: JPEG - Example

- The following quantization matrix is considered.

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Compression Standards: JPEG - Example

- The quantized coefficient matrix is obtained by dividing the coefficient matrix by the quantization matrix and rounding the result on the element-by-element basis.

$$
\begin{bmatrix}
-26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\
0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\
-3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\
-3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\quad
\begin{aligned}
& round\left(\frac{-415.38}{16}\right) \\
& = round(-26.96) \\
& = -26
\end{aligned}
$$

Compression Standards: JPEG - Example

- The quantized coefficient matrix elements are then zig-zag ordered. This forms a 1-D sequence. This contains matrix elements from zero-frequency component to highest frequency component.

DC (zero-frequency) coefficient

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

All elements except the one, which is encircled, are AC coefficients

The resulting 64-element sequence is
{-26,3,0,-3,-2,-6,2,-4,1,-3,1,1,5,1,2,-1,1,-1,2,0,0,0,0,0,-1,-1,0,...,0}

Compression Standards: JPEG - Example

EOB

{-26,3,0,-3,-2,-6,2,-4,1,-3,1,1,5,1,2,-1,1,-1,2,0,0,0,0,0,-1,-1,0,...,0}

- The 63 *AC* coefficients in the sequence is then run-length-encoded as a series of two-element sequences. The second element is a non-zero coefficient in the sequence and the first element is the number of zeros preceding it i.e.,

  {0,3},{1,-3},{0,-2},{0,-6},{0,2},{0,-4},{0,1},{0,-3},{0,1},{0,1},{0,5},{0,1}
  {0,2},{0,-1},{0,1},{0,-1},{0,2},{5,-1},{0,-1},{0,0}

  A special character indicating End Of Block (EOB) i.e., no further non-zero coefficient

- This is continued till the last non-zero coefficient in the sequence. Then the run-length-encoded sequence is ended with a special character known as the End-Of-Block (EOB).

Compression Standards: JPEG - Example

- Then the difference between the current DC coefficient and the DC coefficient of the previous block is calculated.
- The DC difference value and the RLEed *AC* coefficient values are then entropy-coded using the Huffman coding:
  - The DC difference value and the *AC* coefficient values are categorized as shown in the following table.

Range	DC Difference Category	AC Category
0	0	N/A
−1, 1	1	1
−3, −2, 2, 3	2	2
−7, …, −4, 4, …, 7	3	3
−15, …, −8, 8, …, 15	4	4
−31, …, −16, 16, …, 31	5	5
−63, …, −32, 32, …, 63	6	6
−127, …, −64, 64, …, 127	7	7
−255, …, −128, 128, …, 255	8	8
−511, …, −256, 256, …, 511	9	9
−1023, …, −512, 512, …, 1023	A	A
−2047, …, −1024, 1024, …, 2047	B	B
−4095, …, −2048, 2048, …, 4095	C	C
−8191, …, −4096, 4096, …, 8191	D	D
−16383, …, −8192, 8192, …, 16383	E	E
−32767, …, −16384, 16384, …, 32767	F	N/A

JPEG Table 1

Compression Standards: JPEG - Example

- If a DC difference falls in a category 'K', then it is encoded with 'n+K' bits with an 'n'-bit DC difference category base code and 'K' LSBs of its value if difference is positive or 'K' LSBs of its value minus 1 if negative. The base codes for the DC difference categories are shown in the following table.

Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

JPEG Table 2

- If a non-zero *AC* coefficient value falls in a category 'K', then it is encoded with 'n+K' bits with an 'n'-bit base code and 'K' LSBs of its value if positive 'K' LSBs of its value minus 1 if negative. The base code is decided by the number of zeros preceding the non-zero *AC* coefficient and its magnitude category as shown in the following table.

Compression Standards: JPEG - Example

JPEG Table 3

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	1010 (= EOB)	4	8/1	111111000	9
0/1	00	3	8/2	111111111000000	17
0/2	01	4	8/3	1111111110110110	19
0/3	100	6	8/4	1111111110110111	20
0/4	1011	8	8/5	1111111110111000	21
0/5	11010	10	8/6	1111111110111001	22
0/6	111000	12	8/7	1111111110111010	23
0/7	1111000	14	8/8	1111111110111011	24
0/8	1111110110	18	8/9	1111111110111100	25
0/9	1111111110000010	25	8/A	1111111110111101	26
0/A	1111111110000011	26	9/1	111111001	10
1/1	1100	5	9/2	1111111110111110	18
1/2	111001	8	9/3	1111111110111111	19
1/3	1111001	10	9/4	1111111111000000	20
1/4	111110110	13	9/5	1111111111000001	21
1/5	11111110110	16	9/6	1111111111000010	22
1/6	1111111110000100	22	9/7	1111111111000011	23
1/7	1111111110000101	23	9/8	1111111111000100	24
1/8	1111111110000110	24	9/9	1111111111000101	25
1/9	1111111110000111	25	9/A	1111111111000110	26
1/A	1111111110001000	26	A/1	111111010	10
2/1	11011	6	A/2	1111111111000111	18
2/2	11111000	10	A/3	1111111111001000	19
2/3	1111110111	13	A/4	1111111111001001	20
2/4	1111111110001001	20	A/5	1111111111001010	21
2/5	1111111110001010	21	A/6	1111111111001011	22
2/6	1111111110001011	22	A/7	1111111111001100	23
2/7	1111111110001100	23	A/8	1111111111001101	24
2/8	1111111110001101	24	A/9	1111111111001110	25
2/9	1111111110001110	25	A/A	1111111111001111	26
2/A	1111111110001111	26	B/1	1111111001	11
3/1	111010	7	B/2	1111111111010000	18
3/2	111110111	11	B/3	1111111111010001	19
3/3	11111110111	14	B/4	1111111111010010	20
3/4	1111111110010000	20	B/5	1111111111010011	21
3/5	1111111110010001	21	B/6	1111111111010100	22
3/6	1111111110010010	22	B/7	1111111111010101	23
3/7	1111111110010011	23	B/8	1111111111010110	24
3/8	1111111110010100	24	B/9	1111111111010111	25
3/9	1111111110010101	25	B/A	1111111111011000	26
3/A	1111111110010110	26	C/1	1111111010	11
4/1	111011	7	C/2	1111111111011001	18
4/2	1111111000	12	C/3	1111111111011010	19
4/3	1111111110010111	19	C/4	1111111111011011	20
4/4	1111111110011000	20	C/5	1111111111011100	21
4/5	1111111110011001	21	C/6	1111111111011101	22
4/6	1111111110011010	22	C/7	1111111111011110	23
4/7	1111111110011011	23	C/8	1111111111011111	24
4/8	1111111110011100	24	C/9	1111111111100000	25
4/9	1111111110011101	25	C/A	1111111111100001	26
4/A	1111111110011110	26			

Compression Standards: JPEG - Example

JPEG Table 3-contd

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
5/1	1111010	8	D/1	11111111010	12
5/2	1111111001	12	D/2	1111111111100011	18
5/3	1111111110011111	19	D/3	1111111111100100	19
5/4	1111111110100000	20	D/4	1111111111100101	20
5/5	1111111110100001	21	D/5	1111111111100110	21
5/6	1111111110100010	22	D/6	1111111111100111	22
5/7	1111111110100011	23	D/7	1111111111101000	23
5/8	1111111110100100	24	D/8	1111111111101001	24
5/9	1111111110100101	25	D/9	1111111111101010	25
5/A	1111111110100110	26	D/A	1111111111101011	26
6/1	1111011	8	E/1	111111110110	13
6/2	11111111000	13	E/2	1111111111101100	18
6/3	1111111110100111	19	E/3	1111111111101101	19
6/4	1111111110101000	20	E/4	1111111111101110	20
6/5	1111111110101001	21	E/5	1111111111101111	21
6/6	1111111110101010	22	E/6	1111111111110000	22
6/7	1111111110101011	23	E/7	1111111111110001	23
6/8	1111111110101100	24	E/8	1111111111110010	24
6/9	1111111110101101	25	E/9	1111111111110011	25
6/A	1111111110101110	26	E/A	1111111111110100	26
7/1	11111001	9	**F/0**	**111111110111**	**12**
7/2	11111111001	13	F/1	1111111111110101	17
7/3	1111111110101111	19	F/2	1111111111110110	18
7/4	1111111110110000	20	F/3	1111111111110111	19
7/5	1111111110110001	21	F/4	1111111111111000	20
7/6	1111111110110010	22	F/5	1111111111111001	21
7/7	1111111110110011	23	F/6	1111111111111010	22
7/8	1111111110110100	24	F/7	1111111111111011	23
7/9	1111111110110101	25	F/8	1111111111111100	24
7/A	1111111110110110	26	F/9	1111111111111101	25
			F/A	1111111111111110	26

Compression Standards: JPEG - Example

- E.g., the current DC coefficient is DC_n = -26 and let the DC coefficient of previous block be DC_{n-1} = -17. The DC difference is [-26-(-17) = -9].

  The DC difference category for -9 is 4 (JPEG Table 1).

  Hence this coded with 'n+4' bits with a category base code '101' (JPEG Table 2) and 4 LSBs of -9 minus 1.

  The binary of -9 is the one's complement of binary of 9 plus 1 i.e., $(1001)'+1 \rightarrow 0111$; 4 LSBs minus 1 are '0110'.

  Hence the current DC coefficient, -26 is encoded as '1010110'.

Compression Standards: JPEG - Example

- E.g., the first RLEed *AC* coefficient is AC_1 = {0,-3}.

  The magnitude category for -3 is 2 (JPEG Table 1).

  Hence this is coded with 'n+2' bits with a Run/Category base code '01' and 2 LSBs of -3 minus 1.

  The binary of -3 is $(11)'+1 \rightarrow 01$; 2 LSBs minus 1 are '00'.

  Hence the first RLEed *AC* coefficient {0,-3} is encoded as '0100'.
- The complete Huffman code for the block

  {0,-9},{0,3},{1,-3},{0,-2},{0,-6},{0,2},{0,-4},{0,1},{0,-3},{0,1},{0,1}
  {0,5},{0,1},{0,2},{0,-1},{0,1},{0,-1},{0,2},{5,-1},{0,-1},{0,0}

  is

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 11110100 000 1010

Compression Standards: MPEG

- MPEG stands for Moving Pictures Expert Group. MPEG is a standard for the compression of audio/video files.
- Video images are created from still frames of images run at a rate of at least 15 frames per second.
- Video compression is the compression of still frames of images that have 'relative' motion information and are occurring at the rate of at least 15 frames per second.

Terminology in MPEG Compression:

- Types of frames used in MPEG compression - **I** frames (intraframes), **P** frames (predicative frames), and **B** frames (bi-directional frames).
- **I** frames are encoded without reference to any other frames i.e., using just the information in the frame itself, in the same way still images are encoded. This is called **intracoding**. There are generally two or more (often three to six) **I** frames each second, and particularly complex frames are encoded as **I** frames.

Compression Standards: MPEG

- P frames are encoded with reference to a previous frame I or P frame. This is called the **forward prediction**.
- B frames are encoded with reference to both the previous and next I and/ or P frames. This is called the **forward and backward prediction**.
- Use of forward and backward prediction makes a high compression rate possible, because it records only the changes from one frame to the next.
- An I frame plus the following B and P frames before the next I frame together define a **Group of Pictures** (GOP). The size of the GOP can be set to 8, 12, or 16 to optimize encoding to suit different movies and display formats.
- Generally the frames are divided into **macroblocks** of size 16×16 pixels.

Compression Standards: MPEG

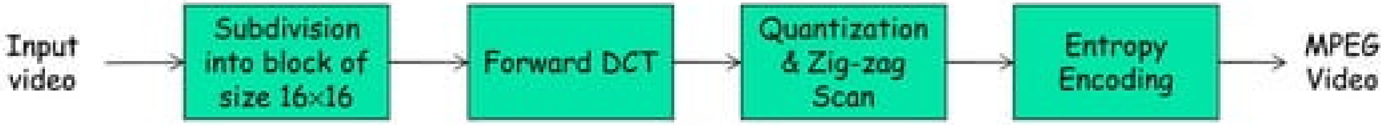
Steps in MPEG compression:

- MPEG video compression is done in two phases: The first phase analyses the video file to decide which frames are to be compressed as **I** frames, which as **P** frames and which as **B** frames. The size of *GOP* and bit rates are also decided. The second phase compresses the video file into series of **I**, **P** and **B** frames.
- The frames are divided into blocks of size 16×16 pixels called macroblocks.
- These macroblocks are in RGB format. Each macroblock is transformed into YUV format. The YUV format consists of a luminance component (Y) and two chrominance components (U and V). The Y component is generated for each of the 16×16 pixels blocked into 8×8 luminance values. The U and V components are generated only for each group of 4 pixels. Thus, for a macroblock of 16×16 pixels, there are four 8×8 luminance (Y) blocks and two 8×8 chrominance (U and V) blocks.

Compression Standards: MPEG

Steps in MPEG compression:

- The macroblocks in an I frame are encoded and decoded in the same way as a still image is encoded in a JPEG compression scheme.



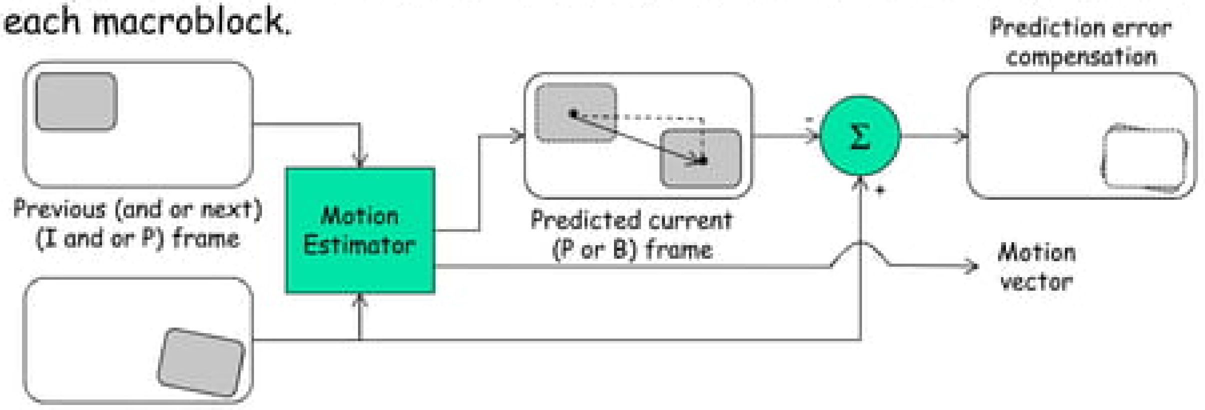
I frame encoding

I frame decoding

Compression Standards: MPEG

<u>Steps in MPEG compression:</u>

- For a **P** frame or a **B** frame, the encoder determines how a macroblock has moved from the previous frame to the current and or from the current frame to the next and then records a corresponding **motion vector** (how much and in what direction the block has moved) and a **prediction error compensation** (how much the block might have "tilted" during the move) for each macroblock.

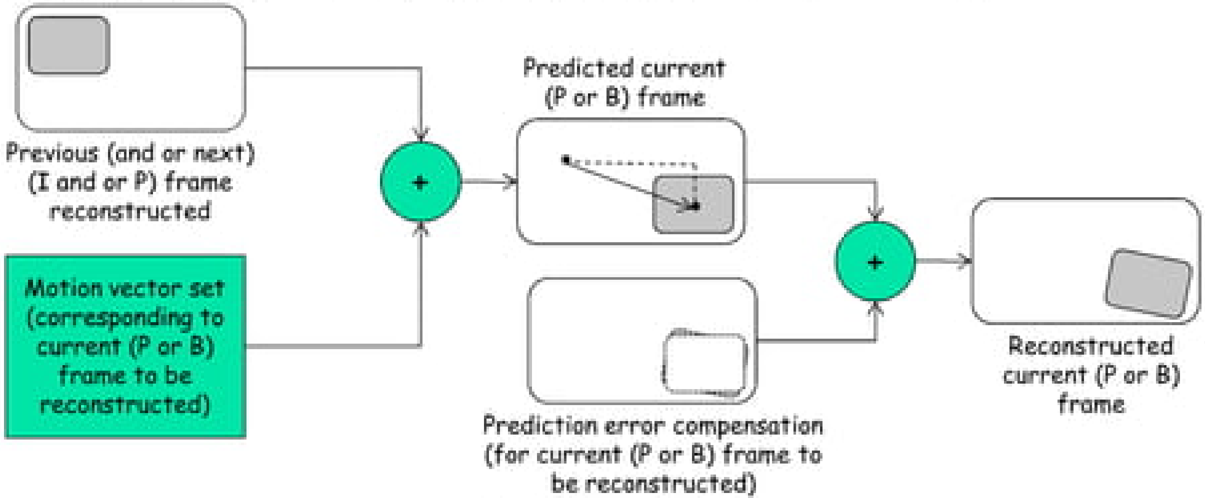


P or B frame encoding

Compression Standards: MPEG

Steps in MPEG compression:

- In decoding a P frame or a B frame is reconstructed by applying the corresponding motion vectors to the previous (and or next) referred frame and adding the prediction error compensation to the result.



P or B frame decoding

Compression Standards: MPEG

Steps in MPEG compression:

- Decoding of a P frame requires only the previous frame. However decoding of a B frame requires both the previous and the next frames for the forward and backward predictions, respectively. This requires that the coding frame sequence be different from the transmitted frame sequence as shown below.

GOP

$I_1 \quad B_1 \quad B_2 \quad P_1 \quad B_3 \quad B_4 \quad P_2 \quad B_5 \quad B_6 \quad I_2$ Coding Frame Sequence

$I_1 \quad P_1 \quad B_1 \quad B_2 \quad P_2 \quad B_3 \quad B_4 \quad I_2 \quad B_5 \quad B_6$ Transmitted Frame Sequence

The decoder has to reorder the reconstructed frames. For this purpose the frames are sequentially numbered in ascending order.

Compression Standards: MPEG - Versions

- MPEG-1 was released in 1991. It was designed for audio/video played mainly from CD-ROMS and hard disks. Its maximum data rate of 1.5 Mbit/s would be too slow for network applications. An MPEG-1 frame is generally 320 X 240 pixels.
- MPEG-2 was issued in 1994. It was intended as a coding standard for television and HDTV with a data rate of ranging from 4 to 80 Mbit/s. MPEG-2 supports interlaced video standards. An MPEG-2 frame is generally 720 X 480 pixels. Other sizes are possible depending on the target audience.
- MPEG-3 was originally intended for HDTV. However, MPEG-2 turned out to be sufficient for HDTV, so MPEG-3 was never really used.
- MPEG-4 is a graphics and video compression algorithm standard that is based on MPEG-1 and MPEG-2 and Apple QuickTime technology. MPEG-4 files are smaller than JPEG or QuickTime files. They can also mix video with text, graphics and 2-D and 3-D animation layers. MPEG-4 was standardized in October 1998.