Model of image degrading/restoration process:

The principal goal of restoration techniques is to improve an image in some predefined sense. Restoration attempts to recover an image that has been degraded, by using a priori knowledge of the degradation phenomenon.

ii. Thus restoration techniques are oriented towards modelling the degradation and applying the inverse process in order to recover the original image.

iii. As Fig. 1 shows, the degradation process is modelled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y).

iv. Given g(x, y), some knowledge about the degradation function H, and some knowledge about the additive noise term f(x, y), the objective of restoration is to obtain an estimate f(x, y) of the original image.

v. We want the estimate to be as close as possible to the original input image and, in general, the more we know about H and f, the closer f(x, y) will be to f(x, y).

vi. The restoration approach used mostly is based on various types of image restoration filters

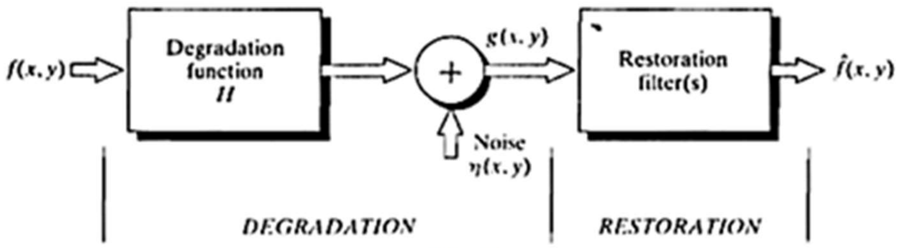


Fig. 1

vii. We know that if H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

$$g(x,y)=h(x,y)*f(x,y)+77(x,y) \quad \text{Eq.(1)}$$

Where h(x, y) is the spatial representation of the degradation function and the symbol "*" indicates convolution.

viii. Now, convolution in the spatial domain is analogous to multiplication in the frequency domain, so we may write the model in Eq. (1) in an equivalent frequency domain representation:

$$G(u,v)=H(u,v)F(u,v)+N(u,v) \ (2)$$

ix. Where the terms in capital letters are the Fourier transforms of the corresponding terms in Eq. (1). These two equations are the bases for most of the restoration material.

x. This is about the basic image restoration model.

noise models (At the end of the pdf)

Restoration in the Present of Noise

When the only degradation present in an image is noise, i.e.

$g(x,y)=f(x,y)+\eta(x,y)$ or $G(u,v)= F(u,v)+ N(u,v)$

The noise terms are unknown so subtracting them from g(x,y) or G(u,v) is not a realistic approach. In the case of periodic noise it is possible to estimate N(u,v) from the spectrumG(u,v).

So N(u,v) can be subtracted from G(u,v) to obtain an estimate of original image. Spatial filtering can be done when only additive noise is present. The following techniques can be used to reduce the noise effect:

i) Mean Filter:

ii) (a)Arithmetic Mean filter: It is the simplest mean filter. Let Sxy represents the set of coordinates in the sub image of size m*n centered at point (x,y). The arithmetic mean filter computes the average value of the corrupted image g(x,y) in the area defined by Sxy. The value of the restored image f at any point (x,y) is the arithmetic mean computed

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

using the pixels in the region defined by Sxy.

This operation can be using a convolution mask in which all coefficients have value 1/mn. A mean filter smoothes local variations in image Noise is reduced as a result of blurring. For every pixel in the image, the pixel value is replaced by the mean value of its neighboring pixels with a weight .This will resulted in a smoothing effect in the image.

(b)Geometric Mean filter:

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

$$\hat{f}(x,y)=\left(\prod_{(s,t)\in Sxy} g(s,t)\right)^{1/mn}$$

This filter is useful for flinging the darkest point in image. Also, it reduces salt noise of the min operation.

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

$$\hat{f}(x,y)=\left(\max_{(s,t)\in Sxy}\{g(s,t)\}+\min_{(s,t)\in Sxy}\{g(s,t)\}\right)\Big/2$$

It comeliness the order statistics and averaging .This filter works best for randomly distributed noise like Gaussian or uniform noise.

(d)Harmonic Mean filter: The harmonic mean filtering operation is given by the expression

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

The harmonic mean filter works well for salt noise but fails for pepper noise. It does well with Gaussian noise also.

(c) Order statistics filter: Order statistics filters are spatial filters whose response is based on ordering the pixel contained in the image area encompassed by the filter. The response of the filter at any point is determined by the ranking result.

Median filter:

It is the best order statistic filter; it replaces the value of a pixel by the median of gray levels in the Neighborhood of the pixel.

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

The original of the pixel is included in the computation of the median of the filter are quite possible because for certain types of random noise, the provide excellent noise reduction capabilities with considerably less blurring then smoothing filters of similar size. These are effective for bipolar and unipolor impulse noise.

Max and Min filter:

Using the l00th percentile of ranked set of numbers is called the max filter and is given by the equation

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

It is used for finding the brightest point in an image. Pepper noise in the image has very low values, it is reduced by max filter using the max selection process in the sublimated area sky. The 0th percentile filter is min filter

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

Linear Position-Invariant Degradations

Inverse filtering

The simplest approach to restoration is direct inverse filtering where we complete an estimate of the transform of the original image simply by dividing the transform of the degraded image G(u,v) by degradation functionH(u,v)

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

We know that

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

Therefore

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

From the above equation we observe that we cannot recover the undegraded image exactly because N(u,v) is a random function whose Fourier transform is not known.

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin.

We know that H(0,0) is equal to the average values of h(x,y). By Limiting the analysis to frequencies near the origin we reduce the probability of encountering zero values.

Minimum mean square error (Weiner) filtering.

The inverse filtering approach has poor performance. The wiener filtering approach uses the degradation function and statistical characteristics of noise into the restoration process.

The objective is to find an estimate of the uncorrupted image f such that the mean square error between them is minimized. The error measure is given by

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

Where E{.} is the expected value of the argument.

We assume that the noise and the image are uncorrelated one or the other has zero mean.

The gray levels in the estimate are a linear function of the levels in the degraded image.

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

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

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

Where H(u,v)= Degradation function

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

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

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

The power spectrum of the under graded image is rarely known. An approach used frequently when these quantities are not known or cannot be estimated then the expression used is

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

Where K is a specified constant

Color Image Processing

The use of color is important in image processing because:

- Color is a powerful descriptor that simplifies object identificationand extraction.
- Humans can discern thousands of color shades and intensities,compared to about only two dozen shades of gray.

Color image processing is divided into two major areas:

- Full-color processing: images are acquired with a full-color sensor,such as a color TV camera or color scanner.
- Pseudocolor processing: The problem is one of assigning a color toa particular monochrome intensity or range of intensities.

Color Fundamentals

Colors are seen as variable combinations of the *primary colors* of light: red (R), green (G), and blue (B). The primary colors can be mixed to produce the *secondary colors*: magenta (red+blue), cyan (green+blue), and yellow (red+green). Mixing the three primaries, or a secondary withits opposite primary color, produces white light.

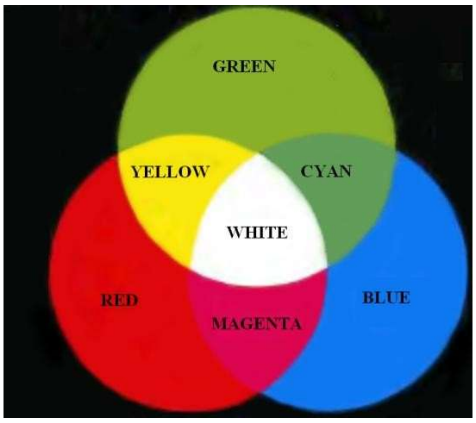


Figure 15.1 Primary and secondary colors of light

RGB colors are used for color TV, monitors, and video cameras.

However, the primary colors of pigments are cyan (C), magenta (M), andyellow (Y), and the secondary colors are red, green, and blue. A proper combination of the three pigment primaries, or a secondary with its opposite primary, produces black.

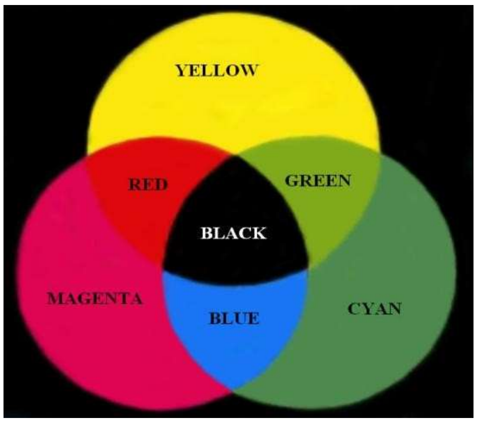


Figure 15.2 Primary and secondary colors of pigments

CMY colors are used for color printing.

Color characteristics

The characteristics used to distinguish one color from another are:

- <u>Brightness:</u> means the amount of intensity (i.e. color level).

- Hue: represents dominant color as perceived by an observer.
- Saturation: refers to the amount of white light mixed with a hue.

Color Models

The purpose of a color model is to facilitate the specification of colors in some standard way. A color model is a specification of a coordinate system and a subspace within that system where each color is representedby a single point. Color models most commonly used in image processing are:

- RGB model for color monitors and video cameras
- CMY and CMYK (cyan, magenta, yellow, black) models for colorprinting
- HSI (hue, saturation, intensity) model

The RGB color model

In this model, each color appears in its primary colors red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace is the cube shown in the figure below. The different colors inthis model are points on or inside the cube, and are defined by vectors extending from the origin.

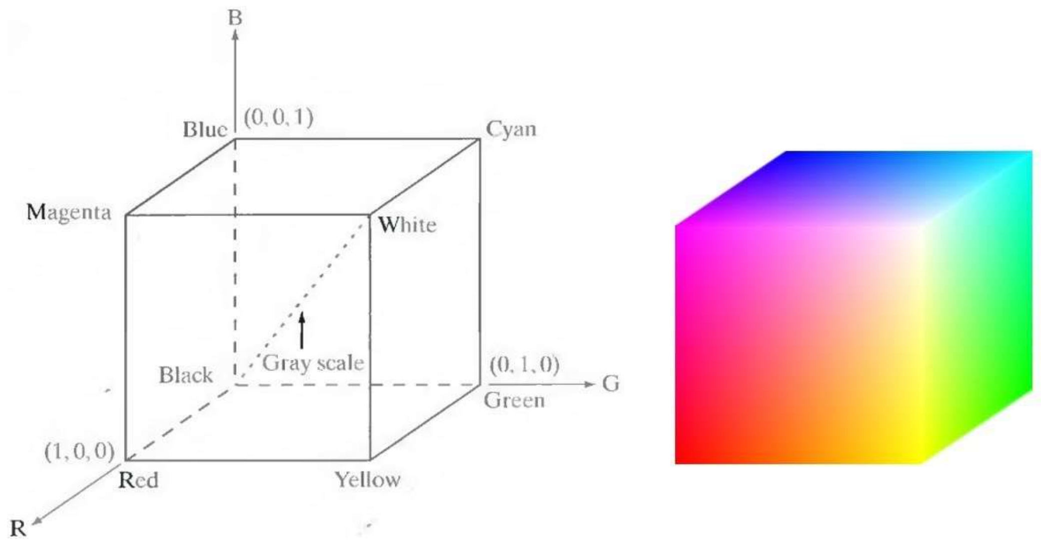


Figure 15.3 RGB color model

All color values R, G, and B have been normalized in the range [0, 1].However, we can represent each of R, G, and B from 0 to 255.

Each RGB color image consists of three component images, one for eachprimary color as shown in the figure below. These three images are combined on the screen to produce a color image.

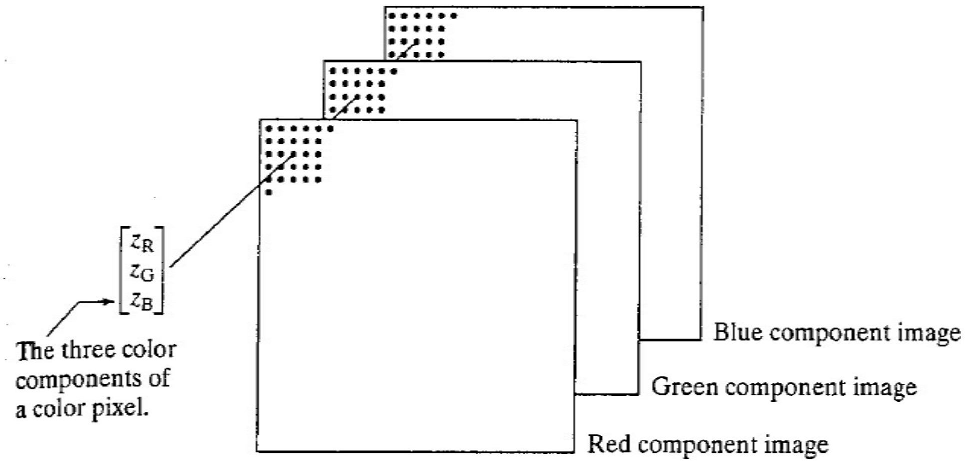
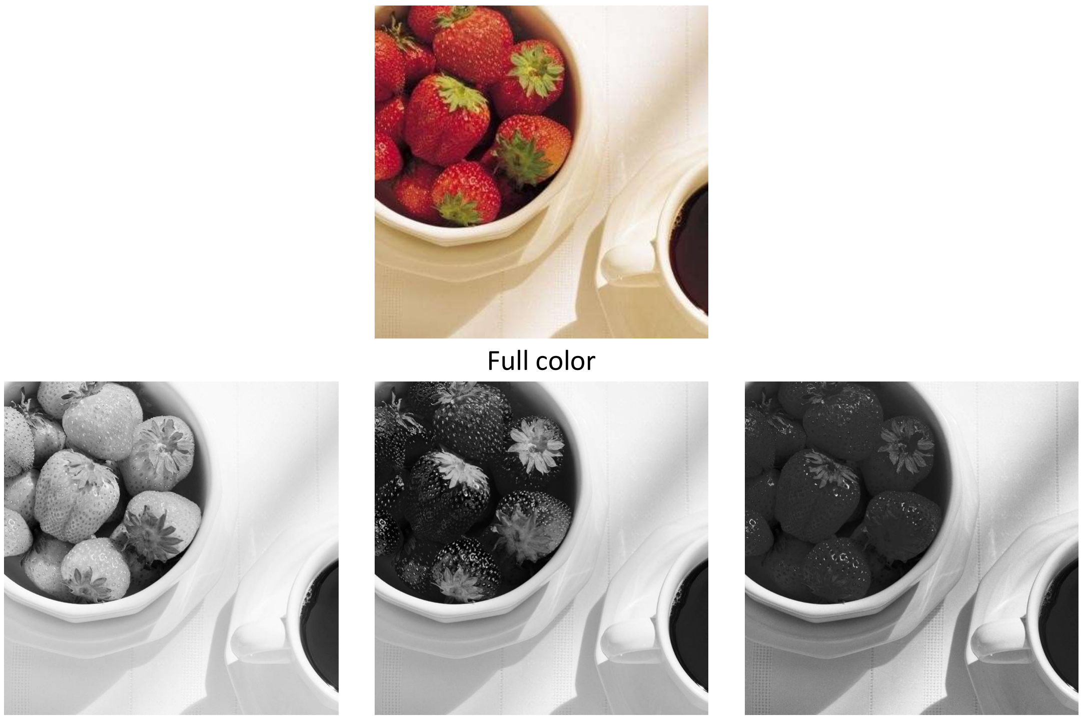


Figure 15.4 Scheme of RGB color image

The total number of bits used to represent each pixel in RGB image iscalled *pixel depth*. For example, in an RGB image if each of the red, green, and blue images is an 8-bit image, the pixel depth of the RGB image is 24-bits. The figure below shows the component images of anRGB image.



Full color

Red Green Blue

Figure 15.5 A full-color image and its RGB component images

The CMY and CMYK color model

Cyan, magenta, and yellow are the primary colors of pigments. Most printing devices such as color printers and copiers require CMY data input or perform an RGB to CMY conversion internally. This conversionis performed using the equation

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

where, all color values have been normalized to the range [0, 1].

In printing, combining equal amounts of cyan, magenta, and yellow produce muddy-looking black. In order to produce true black, a fourthcolor, black, is added, giving rise to the CMYK color model.

The figure below shows the CMYK component images of an RGB image.

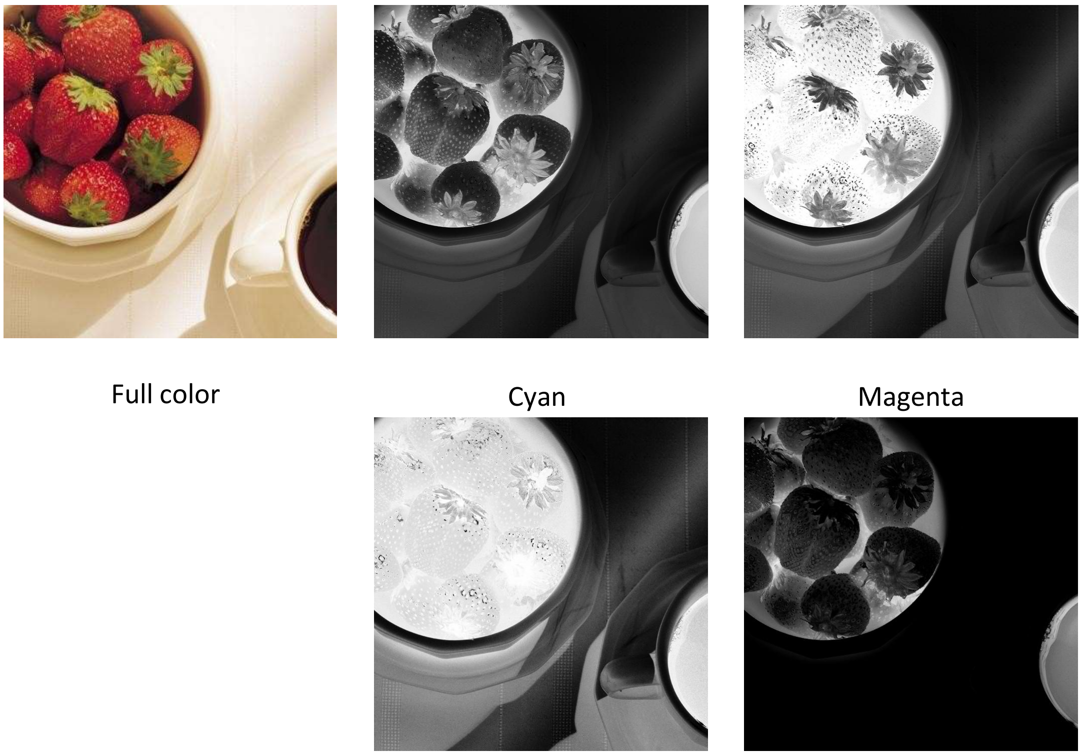


Figure 15.6 A full-color image and its CMYK component images

The HSI color model

The RGB and CMY color models are not suited for describing colors in terms of human interpretation. When we view a color object, we describeit by its hue, saturation, and brightness (intensity). Hence the HSI color model has been presented. The HSI model decouples the intensity component from the color-carrying information (hue and saturation) in a color image. As a result, this model is an ideal tool for developing color image processing algorithms.

The hue, saturation, and intensity values can be obtained from the RGBcolor cube. That is, we can convert any RGB point to a corresponding point is the HSI color model by working out the geometrical formulas.

Converting colors from RGB to HSI

The hue ***H*** is given by

$$H = \begin{Bmatrix} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{Bmatrix}$$

Where

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

The saturation ***S*** is given by

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

The intensity ***I*** is given by

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

All RGB values are normalized to the range [0,1].

Converting colors from HSI to RGB

The applicable equations depend on the value of ***H***:

If $0^\circ \leq H < 120^\circ$:

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

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

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

If $120^\circ \leq H < 240^\circ$:

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

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

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

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

If $240^\circ \leq H \leq 360^\circ$:

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

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

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

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

The next figure shows the HSI component images of an RGB image.

Full color

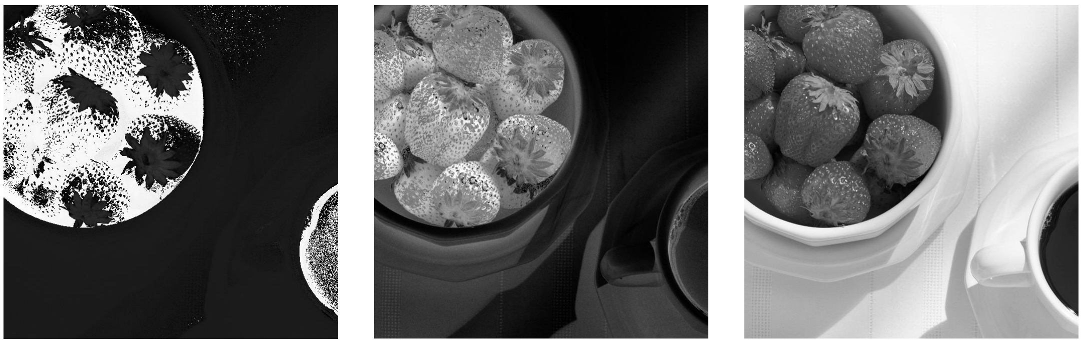


Figure 15.7 A full-color image and its HSI component images

Basics of Full-Color Image Processing

Full-color image processing approaches fall into two major categories:

- Approaches that process each component image individually and then form a composite processed color image from the individuallyprocessed components.
- Approaches that work with color pixels directly.

In full-color images, color pixels really are vectors. For example, in theRGB system, each color pixel can be expressed as

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

For an image of size $M \times N$, there are MN such vectors, $c(x, y)$, for $x = 0,1, 2,\ldots,M\text{-}1;\ y = 0,1,2,\ldots,N\text{-}1$.

In order for per color component and vector based processing to be equivalent, two conditions have to be satisfied: First the process has to be applicable to both vectors and scalars. Second, the operation on each component of a vector must be independent of the other components.

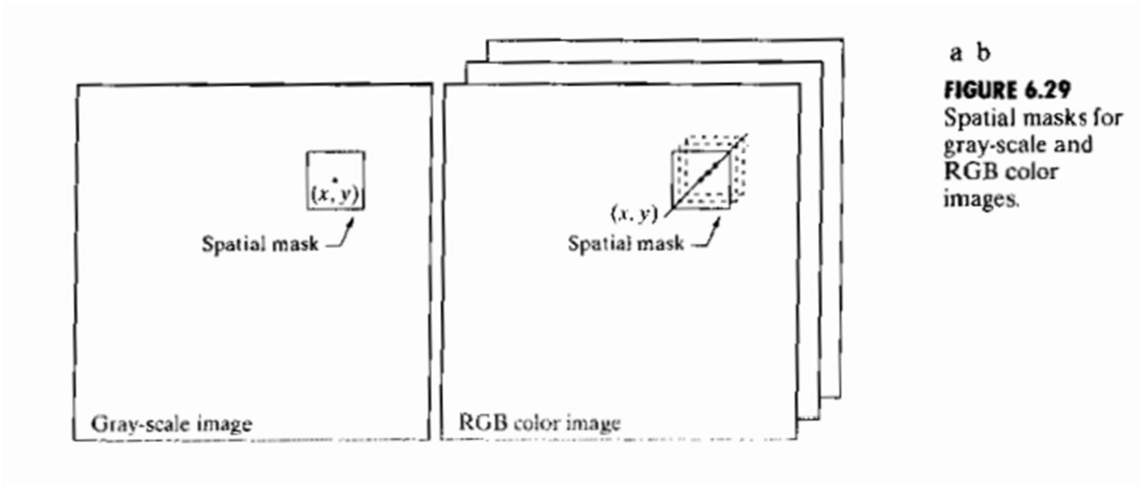


FIGURE 6.29 Spatial masks for gray-scale and RGB color images.

Fig 6.29 shows neighborhood spatial processing of gray-scale and full color images.

Suppose that the process is neighborhood averaging. In fig 6.29(a) averaging would be accomplished by summing the intensities of all the pixels in the neighbdorhood and dividing by the total number of pixels in the neighborhood.

Fig 6.29(b) averaging would be doneby summing all the vectors in the neighborhood and dividing each component by the total number of vectors in the neighborhood. But each component of the average vector is the sum of the pixels in the image corresponding to that component, which is the same as the result that would be obtained if the averaging were done on a per-color component basis and then the vector was formed

Color Transformation

As with the gray-level transformation, we model color transformationsusing the expression

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

where *f(x, y)* is a color input image, *g(x, y)* is the transformed color outputimage, and *T* is the color transform.

This color transform can also be written

$$s_i = T_i(r_1, r_2, \ldots, r_n) \qquad i = 1,2, \ldots, n$$

For example, we wish to modify the intensity of the image shown inFigure 14.8(a) using

$$(x, y) = 0.7f(x, y)$$

- In the RGB color space, three components must be transformed:

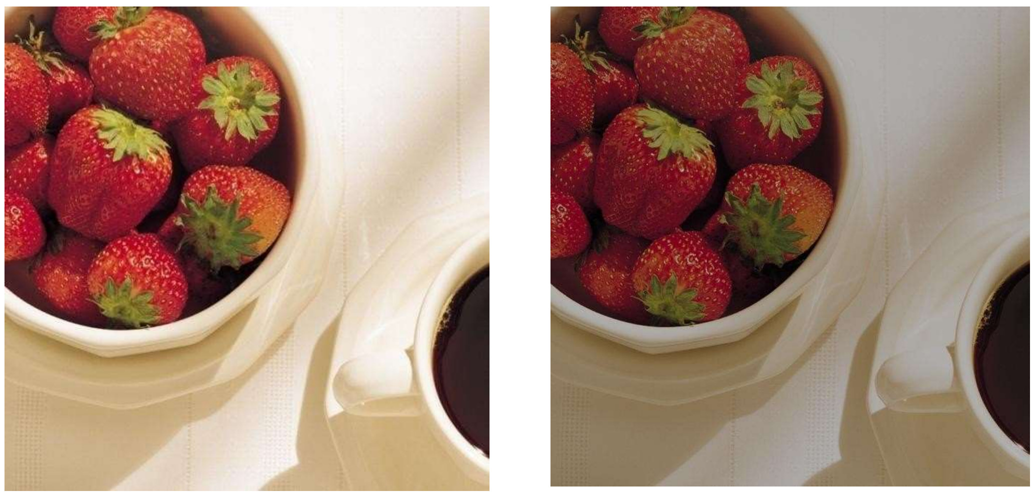
$$s_i = 0.7r_i \qquad i = 1,2,3$$

- In CMY space, also three component images must be transformed

$$s_i = 0.7r_i + 0.3 \qquad i = 1,2,3$$

- In HSI space, only intensity component r_3 is transformed

$$s_3 = 0.7r_3$$



(a) (b)
Figure 15.8 (a) Original image. (b) Result of decreasing its intensity

Pseudocolor image processing

Pseudocolor (also called false color) image processing consists of assigning colors to gray values based on specific criterion. The term pseudo or false colour is used to differentiate the process of assigning colours in monochrome images from the processes associated with true colour images.The principal use of pseudocolor is for human visualization and interpretation of gray scale events in an image or sequence of images.

Intensity Slicing

The technique of intensity (sometimes called density) slicing and colour coding is one of the simplest examples of pseudocolor image processing.

If an image is interepreted as a 3D function, the method can be viewed as one of placing planes parallel to the coordinate plane of the image; each plan then "slices: the function in the area of intersection. Figure 6.18 shows an example of using a plane at $f(x,y) = l_i$ to slice the image function into two levels.

If a different color is assigned to each side of the plane shown in fig 6.18, any pixel whose intensity level is above the plane will be coded with one color, and any pixel below the plane will be coded with the other.

Levels that lie on the plane itself may be arbitrarily assigned one of the two colors. The result is a two color image whose relative appearance can be controlled by moving the slicing plane up and down the intensity axis.

In general the technique may be summarized as follows.

Let [0, L -1] represent the gray scale, Let [0, L – 1] represent the gray scale,

let level l_0 represent black [$f(x,y) = 0$] and level l_{L-1} represent white [$f(x,y) = L -1$].

Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2 \dots l_p$. Then assuming that 0, $P < L – 1$, the P planes partition the gray scale into P + 1 intervals, $V_1, V_2 \dots V_{P+1}$.

Intensity to color assignments are made according to the relation

$$f(x, y) = c_k \qquad \text{if } f(x, y) \in V_k \qquad (6.3\text{-}1)$$

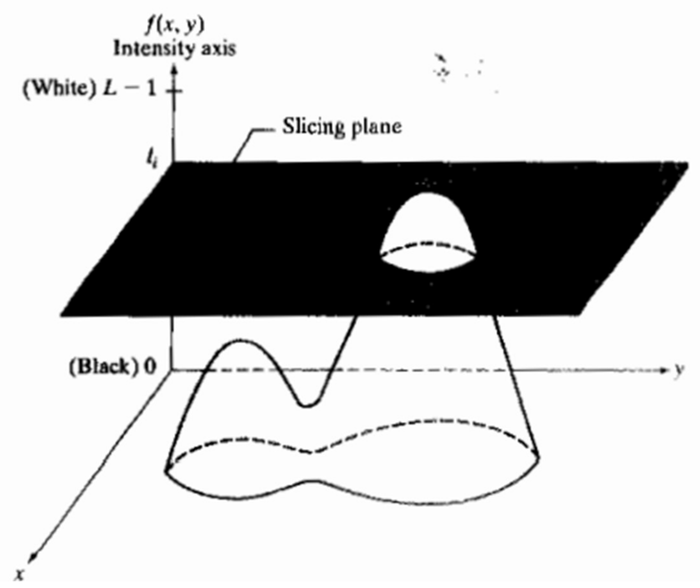
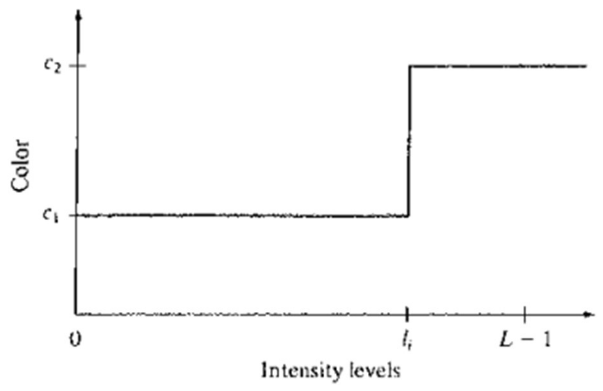


Fig 6.19 shows an alternative representation that defines the same mapping as in fig 6.18. According to the mapping functions shown in fig 6.19, any input intensity level is assigned one of two colours, depending on whether it is above or below the value of l_i.

When more levels are used, the mapping function takes on a staircase form

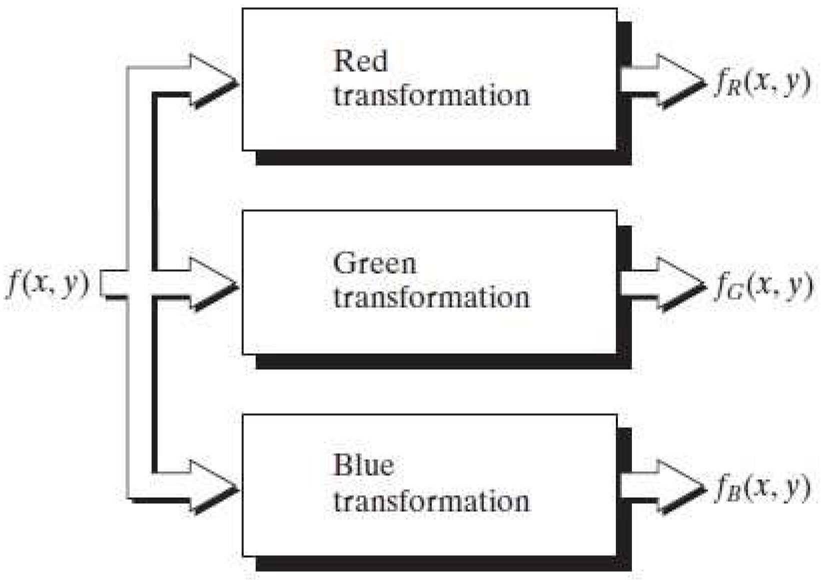
FIGURE 6.19 An alternative representation of the intensity-slicing technique.



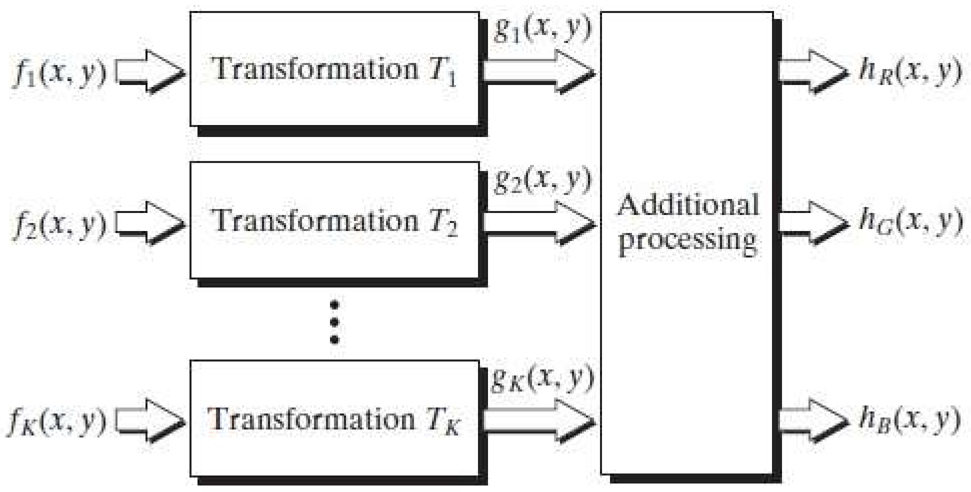
Intensity to Color Transformation

We can generalize the above technique by performing three independent transformations on the intensity of the image, resulting in three images which are the red, green, blue component images used to produce a color image. The three results are then fed separately into the red, green and blue channels of a color television monitor.

This method produces a composite image whose color content is modulated by the nature of the transformation functions.



The flexibility can be even more enhanced by using more than one monochrome images, for example, the three components of an RGB and the thermal image.



This technique is called **multispectral image processing.**

We often come across *multispectral images* in remote sensing systems, for example, Landsat Thematic Mapper (TM) Band 7 which targets the detection of hydrothermal alteration zones in bare rock surfaces.

Multispectral image processing allows us to infer the wavelengths that cannot be captured by the conventional RGB cameras or even human eyes. This is an important technique for space-based images, or document and painting analysis.

Noise models

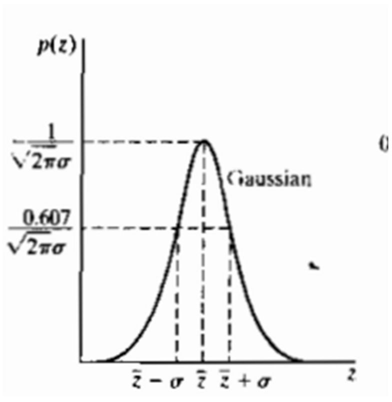
The principal source of noise in digital images arises during image acquisition and or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted during transmission principally due to interference in the channels used for transmission. Since main sources of noise presented in digital images are resulted from atmospheric disturbance and image sensor circuitry, following assumptions can be made i.e. the noise model is spatial invariant (independent of spatial location). The noise model is uncorrelated with the object function.

GAUSSIAN NOISE:

These noise models are used frequently in practices because of its tractability in both spatial and frequency domain.

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

The PDF of Gaussian random variable is



Where z represents the gray level, μ= mean of average value of z, σ= standard deviation.

Source of Gaussian Noise :

The Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination. The images acquired by image scanners exhibit this phenomenon.

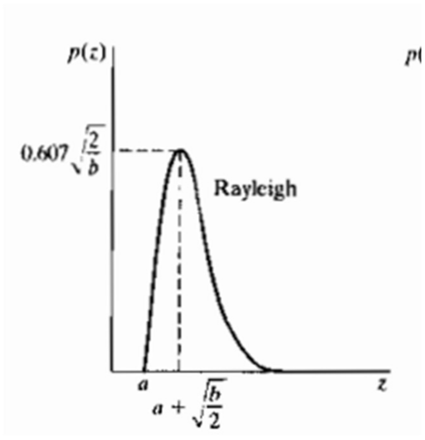
RAYLEIGH NOISE:

Unlike Gaussian distribution, the Rayleigh distribution is no symmetric. It is given by the formula

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

The mean and variance of this density is

$$m = a + \sqrt{\pi b/4}, \sigma^2 = \frac{b(4-\pi)}{4}$$



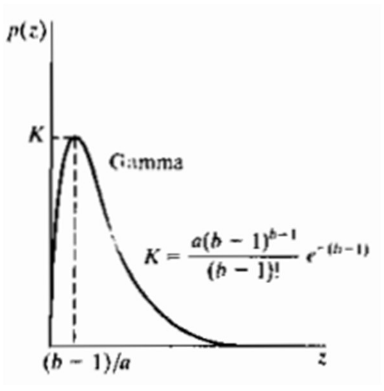
(iii) GAMMA NOISE: The PDF of Erlang (Gamma) noise is given by

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

The mean and variance of this density are given by and

$$\bar{z} = \frac{b}{a}$$

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



Its shape is similar to Rayleigh disruption. This equation is referred to as gamma density it is correct only when the denominator is the gamma function

(iv) EXPONENTIAL NOISE: Exponential distribution has an exponential shape.

The PDF of exponential noise is given as

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

Where a>0. The mean and variance of this density are given by

and

$$\bar{z} = \frac{1}{a}$$

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

Exponential pdf is a special case of Erlang pdf with b =1.

Used in laser imaging.

(v)UNIFORM NOISE: The PDF of uniform noise is given by

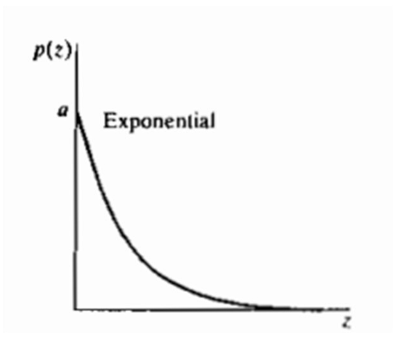
$$p_z(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean of this density function is given by

$$\bar{z} = \frac{a+b}{2}$$

and its variance by

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



$$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$$

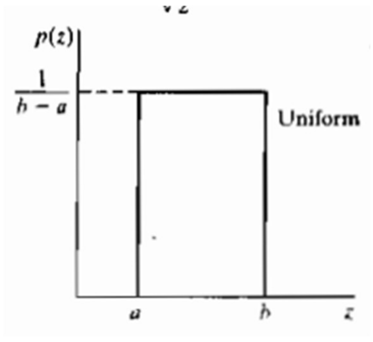


Figure 3.2.4 shows a plot of the Rayleigh density. Note the displacement from the origin and the fact that the basic shape of this density is skewed to the right. The Rayleigh density can be quite useful for approximating skewed histograms.

Impulse (salt-and-pepper) noise (bipolar) is specified as

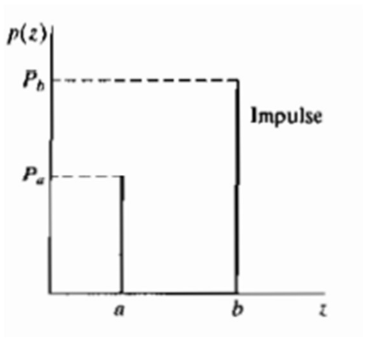


Figure shows a plot of the Rayleigh density. Note the displacement from the origin and the fact that the basic shape of this density is skewed to the right. The Rayleigh density can be quite useful for approximating skewed histograms

If b>a, intensity b will appear as a light dot on the image and a appears as a dark dot If either Pa or Pb is zero the noise is called unipolar. If neither probability is zero, and especially if they are approximately equal, impulse noise value will resemble salt and pepper granules randomly distributed over the image. For this reason, bipolar impulse noise is called salt and pepper noise.

In this case, the noise is signal dependent, and is multiplied to the image.

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

$$b > a$$