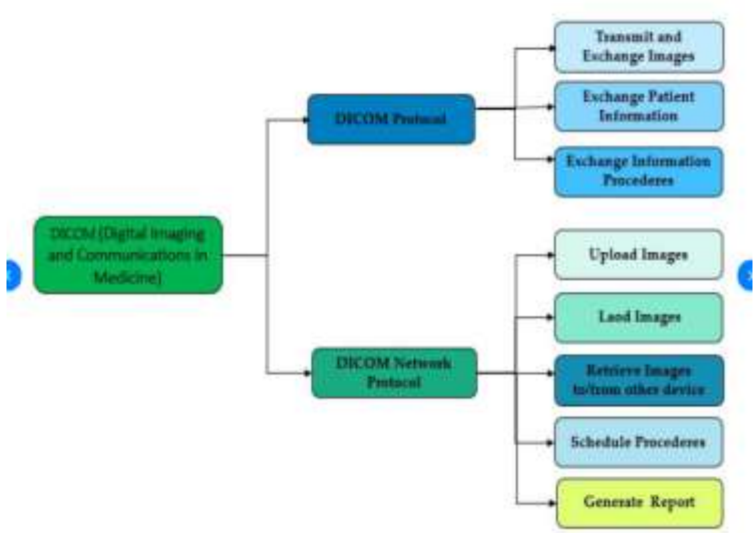
For objects exchanged using the PS3.18 Web Services, the presentation address shall be encoded as the absolute URL of the endpoint of the base of the resource or service, sufficient to identify the system. The presentation address is not expected to be the complete address of the resource. The scheme shall be "http", regardless of whether secure transport was actually used or not.

The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements. This header shall be present in every DICOM file.

The File Preamble is available for use as defined by Application Profiles or specific implementations. This Part of the DICOM Standard does not require any structure for this fixed size Preamble. It is not required to be structured as a DICOM Data Element with a Tag and a Length. It is intended to facilitate access to the images and other data in the DICOM file by providing compatibility with a number of commonly used computer image file formats.



Block diagram of the DICOM standard

Analyze 7.5 :

It is a file format, developed by the Mayo Clinic, for storing MRI data. An Analyze 7.5 data set consists of two files: Header file (filename .hdr) — Provides information about dimensions, identification, and processing history.

NIFTI: NIfTI (Neuroimaging Informatics Technology Initiative) is a data format for the storage of Functional Magnetic Resonance Imaging (fMRI) and other medical images. The NIfTI format is adapted from Analyze™ 7.5, developed by Biomedical Imaging Resource (BIR) at Mayo Clinic.

INTERFILE: Interfile is a file format developed for data in Nuclear Medicine (Todd-Pokropek A, Cradduck TD, Deconinck F, A file format for the exchange of nuclear medicine image data: a specification of Interfile version 3.3, Nucl Med Commun.

Table 1

Summary of file formats characteristics

Format	Header	Extension	Data types
Analyze	Fixed-length: 348 byte binary format	.img and .hdr	Unsigned integer (8-bit), signed integer (16-, 32-bit), float (32-, 64-bit), complex (64-bit)
Nifti	Fixed-length: 352 byte binary format[a] (348 byte in the case of data stored as .img and .hdr)	.nii	Signed and unsigned integer (from 8- to 64-bit), float (from 32- to 128-bit), complex (from 64- to 256-bit)
Minc	Extensible binary format	.mnc	Signed and unsigned integer (from 8- to 32-bit), float (32-, 64-bit), complex (32-, 64-bit)
Dicom	Variable length binary format	.dcm	Signed and unsigned integer, (8-, 16-bit; 32-bit only allowed for radiotherapy dose), float not supported

In formats that adopt a fixed-size header, the pixel data start at a fixed position after skipping the header length. In the case of variable length header, the starting location of the pixel data is marked by a tag or a pointer. In any case, to calculate the *pixel data size*, we have to do:

$$Rows * Columns * Pixel\ Depth * (Number\ of\ Frames)$$

where the pixel depth is expressed in bytes. The *image file size* will be given by:

Header Size + Pixel Data Size

Arithmetic and Logical operations in Images

Logical operations apply only to binary images, whereas arithmetic operations apply to multi-valued pixels. Logical operations are basic tools in binary image processing, where they are used for tasks such as masking, feature detection, and shape analysis.

Arithmetic and Logical Operations

These operations are applied on pixel-by-pixel basis. So, to add two images together, we add the value at pixel (0 , 0) in image 1 to the value at pixel (0 , 0) in image 2 and store the result in a new image at pixel (0 , 0). Then we move to the next pixel and repeat the process, continuing until all pixels have been visited.

Clearly, this can work properly only if the two images have identical dimensions.

Addition can also be used to ***combine the information of two images***, such as an image morphing, in motion pictures.

Algorithm 1: image addition

```
read input-image1 into in-array1;
read input-image2 into in- array2;
for i = 1  to no-of-rows do
for j=1 to no-of-columns do
begin
out-array (i,j) = in-array1(i,j) + in-array2(i,j);
if ( out-array (i,j) > 255 ) then
out-array (i,j) = 255;end
write out-array to out-image;
```

a *b* *c*

Figure a) noisy image b) average of five observation c) average of ten observation

Subtraction

Subtracting two 8-bit grayscale images can produce values between - 225 and +225. This necessitates the use of 16-bit signed integers in the output image unless sign is unimportant, in which case we can simply take the modulus of the result and store it using 8-bit integers:

$g(x,y) = |f_1(x,y) - f_2(x,y)|$

The main application for image subtraction is in ***change detection*** (or ***motion detection***). Subtraction can also be used in ***medical imaging to remove static background information.***

Algorithm2: image subtraction

```
read input-image1 into in-array1; read input-image2 into in- array2;for i = 1  to no-of-rows do
for j=1 to no-of-columns dobegin
out-array (i,j) = in-array1(i,j) - in-array2(i,j);
if ( out-array (i,j) < 0 ) then out-array (i,j) = 0; endwrite out-array to out-image;
```

Figure a, b) two frames of video sequence c) their difference

Multiplication and division

Multiplication and division can be used to adjust brightness of an image. Multiplication of pixel values by a number greater than one will brighten the image, and division by a factor greater than one will darken the image. Brightness adjustment is often used as a ***preprocessing step*** in image enhancement.

(a) *(b)* *(c)*

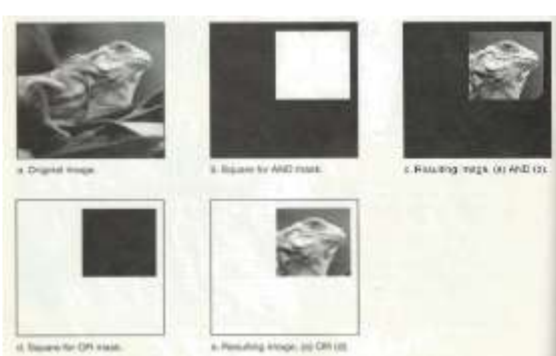
Figure a) original image b) image multiplied by 2 c) image divided by 2

Logical operations

Logical operations apply ***only to binary images***, whereas arithmetic operations apply to multi-valued pixels. Logical operations are basic tools in binary image processing, where they are used for tasks such as ***masking***, ***feature detection***, and ***shape analysis***. Logical operations on entire image are performed pixel by pixel.

	AND					OR					XOR			
Input 1	1	1	0	0		1	1	0	0		1	1	0	0
2	1	0	1	0		1	0	1	0		1	0	1	0
output	1	0	0	0		1	1	1	0		0	1	1	0

Logical AND & OR operations are useful for the ***masking and compositing*** of images. ***So, masking is a simple method to extract a region of interest from an image.***



In addition to masking, logical operation can be used in feature detection. Logical operation can be used to compare between two images, as shown below:

AND

This operation can be used to find the ***similarity*** white regions of twodifferent images (it required two images).

$$g(x,y) = a(x,y) \wedge b(x,y)$$

Exclusive OR

This operator can be used to find the differences between white regions of twodifferent images (it requires two images).

$$g(x,y) = a(x,y) \bullet b(x,y)$$

NOT

NOT operation can be performed on gray-level images, and the result of this operation is the ***negative*** of theoriginal image.

$$g(x,y) = 255 - f(x,y)$$

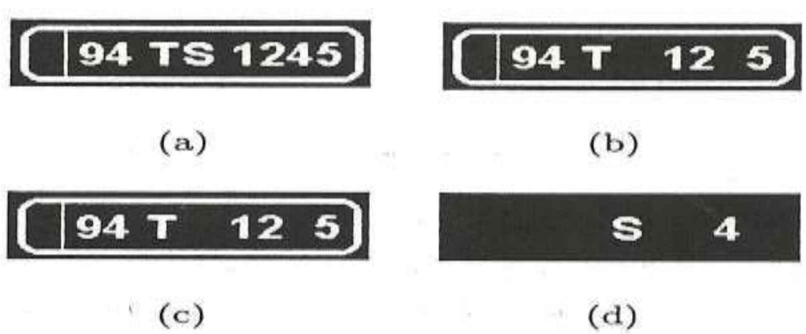


Figure a) input image a(x,y); b) input image b(x,y) ; c) a(x,y) ^ b(x,y) ; d) a(x,y) ^ ~ b(x,y)

Image Quality and Signal to Noise Ratio

The few imaging artifacts which stem from the particular properties of imaging systems. These artifacts can, of course, reduce diagnostic information content of images and are to be avoided. However, another source of reduced image quality is noise. The term noise describes all types of stochastic signals which are not related to image content. In x-ray imaging, noise is usually introduced by a lack of dose; in photography, images tend to get noisy if the ambient lighting is insufficient. The reason in both cases is simple – since image noise is a stochastic signal, its amplitude is independent of the usable signal. If the imaging signal itself is of small amplitude, the noise becomes more visible. A measure to quantify the noisiness of images is the signal-to-noise-ratio (SNR). One possible definition is given as:

$$\text{SNR} = \frac{\bar{\rho}}{\sigma\left(\rho_{\text{Uniform Area}}\right)}$$

$\bar{\rho}$ ... average pixel gray value
σ (ρUniform Area) ... standard deviation in a signal-free area of the image
From the above equation, it is obvious why an underexposed image becomes grainy. The energy impacted (in terms of visible light or dose) is proportional to the gray value ρ, whereas the noise remains the same. The reason lies in the fact that the image noise, which can have numerous origins, is not necessarily connected to the original signal. While SNR is an important measure of detector efficiency and image quality, it does not give an idea of the ability of an imaging system to resolve fine detail, and it does not necessarily give a figure of the information content of an image.
This is characterized by two more important measures, the point spread function (PSF) and the modulation transfer function (MTF), gives an idea how the SNR can be computed from an image in dependence of the dose applied. However, since the average signal is not always a useful measure, on can also define the contrast-to-noise-ratio (CNR). Here, the average gray value is replaced by the absolute value of the difference between the maximal and minimal signal:

$$\text{CNR} = \frac{\left|\rho_{\text{max}} - \rho_{\text{min}}\right|}{\sigma\left(\rho_{\text{Uniform Area}}\right)}$$

ρmax, ρmin = maximal and minimal gray value
σ (ρUniform Area) =standard deviation in a signal-free area of the image

Color image processing

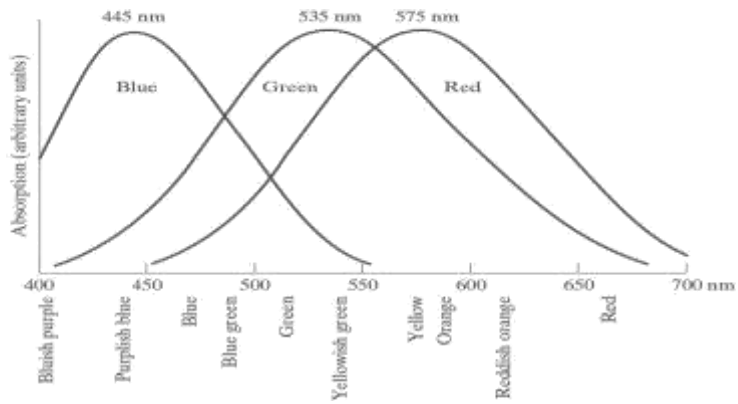
In 1666, Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging beam is split into a spectrum of colors. The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color. Three independent quantities are used to describe any particular color. The *hue* is determined by the dominant wavelength. Visible colors occur between about 400nm (violet) and 700nm (red) on the electromagnetic spectrum,

Color Fundamentals

- 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue.
- 65%: red 33%: green 2%: blue (blue cones are the most sensitive)
- The characteristics generally used to distinguish one color from another are brightness, hue, and saturation.

➢ **Brightness**: the achromatic notion of intensity.
➢ **Hue**: dominant wavelength in a mixture of light waves, represents dominant color as perceived by an observers**.**

- **Saturation**: relative purity or the amount of white light mixed with its hue.

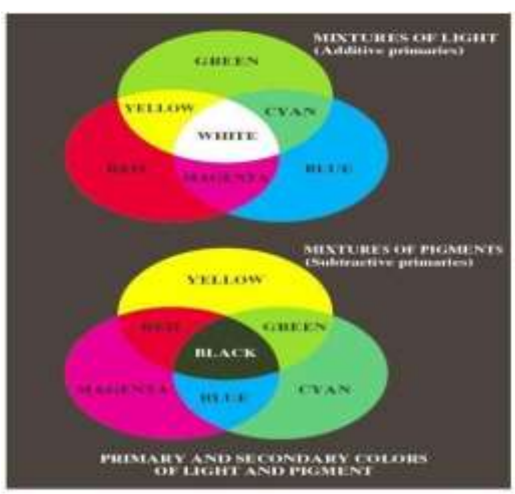


Color models:

Different color models are:

- **RGB: Color Monitor, Color Camera, Color Scanner**
- **CMY: Color Printer, Color Copier**
- **YIQ: Color TV, Y(luminance), I(In phase), Q(quadrature) – HSI, HSV**

The purpose of a color model is to facilitate the specification of colors in some standard. In essence, a color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point. Most color models are oriented either toward specific hardware or toward applications. Red, green, and blue, three primary colors. Cone cells in human eye are responsible for color vision.

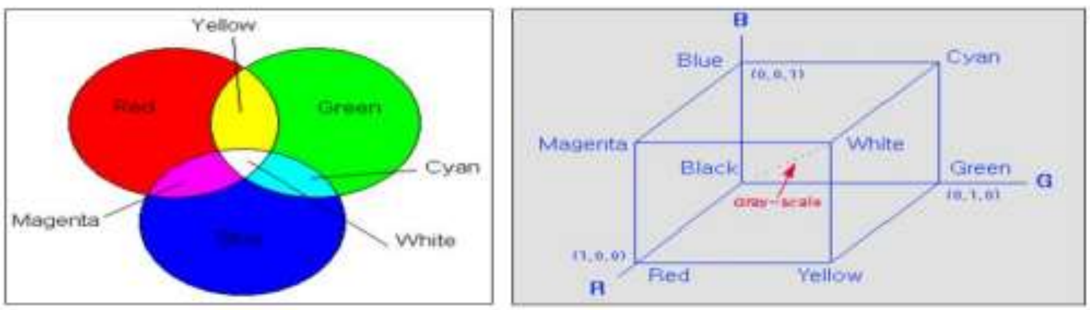


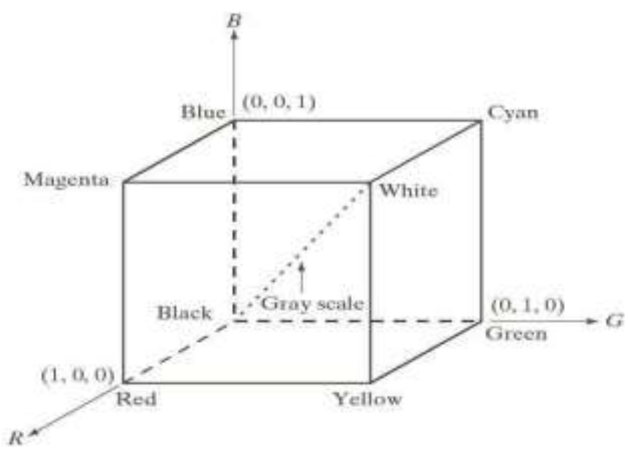
The RGB Model

In the RGB model each color appears in its primary spectral components of red, green and blue. The model is based on a Cartesian coordinate system RGB values are at 3 corners. Cyan magenta and yellow are at three other corners. Black is at the origin. White is the corner furthest from the origin. Different colors are points on or inside the cube represented by RGB vectors. Images represented in the RGB color model consist of three component images – one for each primary color. When fed into a monitor these images are combined to create a composite color image. The number of bits used to represent each pixel is referred to as the

color depth. A 24-bit image is often referred to as a full-color image as it allows16, 777,216 colors .The RGB model is used for color monitors and most video cameras.

- RGB Model





The CMY Model:

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

To produce true black, the predominant color in printing, the fourth color, black, is added, giving rise to the CMYK color model. The primary colors (R, G, B) can be added to produce the secondary colors.

- Red plus blue can generate magenta
- Green plus blue can generate cyan
- Red plus green can generate yellow

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

- CMY Model
  - Color Printer, Color Copier
  - RGB data CMY

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

YIQ models:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.620 \\ 1 & -0.272 & -0.647 \\ 1 & -1.108 & 1.705 \end{bmatrix} \times \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

The HIS Model:

Human eye distinguish one color from the other based on hue, saturation, and brightness. Hue, is a color that is evoked by a single wavelength of light in the visible spectrum, or by a relatively narrow band of wavelengths; hue represents dominant color as perceived by the observer.

Saturation (purity) refers to the relative amount of white light mixed with a hue; it is inversely proportional to the amount of white light added; to de saturate a color of given intensity in a subtractive system (such as watercolor), one can add white, black, gray.

Brightness embodies the achromatic notion of intensity.

Hue and saturation together are called chromaticity. For any particular color, the amounts of red, green, and blue needed to form it are called tristimulus values, and they are denoted as X, Y, and Z.A color is specified by its trichromatic coefficients, defined as

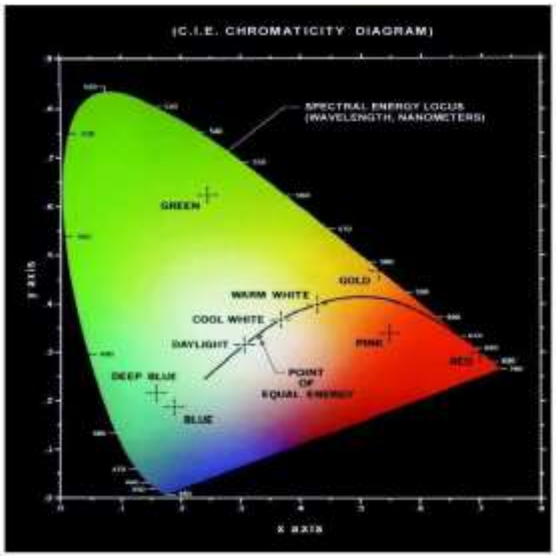
$$x = \frac{X}{X+Y+Z}$$

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

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

$$x + y + z = 1$$

CIE chromaticity diagram: It is a function of *x* (red) and *y* (green), *z* can be derived by *z* =1-*x*-*y*



CIE chromaticity diagram

Any color located on the boundary of the chromaticity chart is fully saturated; any point not on the boundary but within the diagram represents some mixture of spectrum colors The point of equal energy represents the standard white light; its saturation is zero

As a point leaves the boundary and approaches the point of equal energy, more white light is added to the color and it becomes less saturated

A straight line segment joining any two points in the diagram defines all the different color variations that can be obtained by combining these two colors additively

The HIS model uses three measures to describe colors, hue, saturation, and intensity

- Their relationship can be represented in a cylindrical coordinate system
- angle around the central vertical axis corresponds to "hue",
- the distance from the axis corresponds to "saturation",
- The distance along the axis corresponds to "value"

Conversion from RGB to HIS

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

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

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

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

Conversion from HSI to RGB

- RG sector ($0 <= H < 120°$)

$$B = I(1-S) \quad R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \quad G = 3I - (R+B)$$

- GB sector ($120° <= H < 240°$)

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

$$R = I(1-S) \quad G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \quad B = 3I - (R+G)$$

- BR sector ($240° <= H <= 360°$)

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

$$G = I(1-S) \quad B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \quad R = 3I - (G+B)$$

Pseudo color Image Processing:

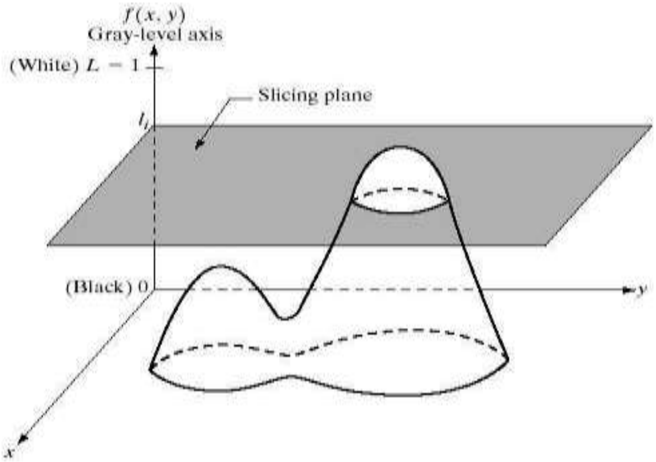
Pseudo color image processing consists of assigning colors to grey values based on a specific criterion

- The principle use of Pseudo color image processing is for human visualization
- Humans can discern between thousands of color shades and intensities, compared to only about two dozen or so shades of grey

Intensity Slicing

Intensity slicing and color coding is one of the simplest kinds of pseudo color image processing

- First we consider an image as a 3D function mapping spatial coordinates to intensities (that we can consider heights)
- Now consider placing planes at certain levels parallel to the coordinate plane
- If a value is one side of such a plane it is rendered in one color, and a different color if on the other side. Intensity slicing and color coding is one of the simplest kinds of pseudo color image processing



Intensity slicing and color coding is one of the simplest kinds of Pseudo color image processing In general intensity slicing can be summarized as:

Let [*0*, *L*-1] represent the grey scale

- Let l_0 represent black [$f(x, y) = 0$] and let l_{L-1} represent white [$f(x, y) = L-1$]
- Suppose P planes perpendicular to the intensity axis are defined at levels $l_1, l_2, \ldots, l_p$
- Assuming that $0 < P < L-1$, then the P planes partition the grey scale into $P + 1$ intervals
  $V_1, V_2, \ldots, V_{P+1}$

Grey level color assignments can then be made according to the relation:

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

Where c_k is the color associated with the k^{th} intensity level V_k defined by the partitioning planes at $l = k - 1$ and $l = k$

Full Color Image Processing

Full color image processing approaches fall into two major categories.

- Process each component image individually and form composite processed color image from the individually processed components
- Work with color pixels directly; color pixels really are vectors:

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

Note: the results of individual color component processing are not always equivalent to direct processing in color vector space. Processing is equivalent if:

(1) The process is applicable to both scalars and vectors;

(2) The operation on each component of a vector is independent of the other components.

Result for per-color-component and vector-based processing is equivalent.

Image Transforms - 2D DFT (Discrete Fourier Transform)

Discrete Fourier Transform and the Frequency Domain

Any function that periodically reports itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient, this sum is called Fourier series. Even the functions which are non-periodic but whose area under the curve if finite can also be represented in such form; this is now called Fourier transform.

A function represented in either of these forms and can be completely reconstructed via an inverse process with no loss of information.

1-D Fourier Transformation and its Inverse

If there is a single variable, continuous function f(x), then Fourier transformation F (u) may be given as

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\exp(-j2\pi ux)dx \qquad j = \sqrt{-1}$$

And the reverse process to recover f(x) from F(u) is

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u)\exp[j2\pi ux]du$$

The above equations comprise of Fourier transformation pair.

Fourier transformation of a discrete function of one variable f(x), x=0, 1, 2, m-1 is given by

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N] \quad \text{for } u=0,1,2,\ldots,N-1$$

to obtain f(x) from F(u)

$$f(x) = \sum_{u=0}^{N-1} F(u)\exp[j2\pi ux/N] \quad \text{for } x=0,1,2,\ldots,N-1$$

Now each of the m terms of F (u) is called a frequency component of transformation

Discrete Fourier Transform and its Properties

In the two-variable case the discrete Fourier transform pair is

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

for $u = 0, 1, 2, \ldots, M-1$, $v = 0, 1, 2, \ldots, N-1$, and

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

for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$.

When images are sampled in a squared array, i.e. M=N, we can write

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

for $u, v = 0, 1, 2, \ldots, N - 1$, **and**

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

2 Dimensional DFT

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

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

where u, v are frequency variables [Transfer]

x, y are spatial variable [Image]

- Fourier spectrum $= |F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$
- Phase angle $\phi[u,v] = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
- Power spectrum

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

Properties of 2D-DFT

1) Periodicity

To prove: $DFT[f(m+N, n+N)] = F(u,v)$

Solution:-

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

Put $m+N = m'$, $n+N = n'$

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

$$DFT[f(m+N, n+N)] = F(u,v)$$

Hence proved.

2) Time Shift.

To prove:-

$$DFT\{f(m-m_0, n-n_0)\} = F[u,v]$$

$$\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m-m_0, n-n_0)\, e^{-j2\pi\left[\frac{m-m_0}{M}u + \frac{n-n_0}{N}v\right]}$$

Put $m-m_0 = m'$, $n-n_0 = n'$

$$\frac{1}{MN}\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1} f(m', n')\, e^{-j2\pi\left[\frac{um'}{M}+\frac{vn'}{N}\right]}$$

$$= f(u,v)$$

$$DFT\{f(m-m_0, n-n_0\} = F[u,v]$$

Hence proved.

3) Rotational property.

To Prove,

$$f(r,\theta+\theta_0) \iff F[\omega, \phi+\theta_0]$$

Solution;

Consider Polar Coordinates

$x = r\cos\theta$; $y = r\sin\theta$, $u = \omega\cos\phi$, $v = \omega\sin\phi$.

then $f(x,y) = f(r,\theta)$

$F[u,v] = F[\omega,\phi]$

If $f(x,y)$ and $F[u,v]$ rotates by the same angle.

$$F[r,\theta+\theta_0] \iff F[\omega,\phi+\theta_0]$$

Hence proved.

4) Scaling Property

To Prove

$$DFT[f(am, bn)] = F[u/a, v/b]$$

Solution:-

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

$$\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(am,bn)\, e^{-j2\pi\left[\frac{um}{M}+\frac{vn}{N}\right]}$$

$am = m'$ $\quad bn = n'$

$m = m'/a$ $\quad n = n'/a$.

$$\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m',n')\, e^{-j2\pi\left[\frac{um'}{aM}+\frac{vn'}{bN}\right]}$$

$$F[u/a, v/b]$$

$$DFT[f(am,bn)] = F[u/a, v/b]$$

Hence Proved.

5) Frequency shift.

To prove.

$$F(k-k_0, l-l_0) = f(m,n).$$

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

↳ Inverse DFT.

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(k-k_0, l-l_0) e^{j2\pi\left[\frac{(k-k_0)m}{M} + \frac{(l-l_0)n}{N}\right]}.$$

2D DCT

- Corresponding 2D formulation

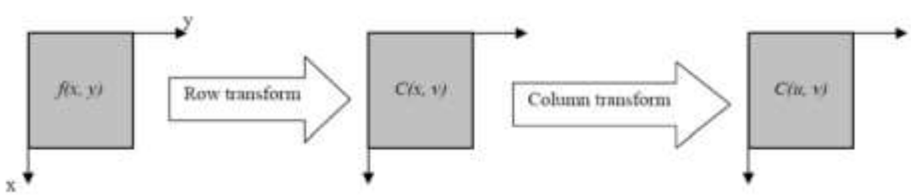
direct $$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right],$$

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

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0. \end{cases}$$

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

Separability



The inverse of a multi-dimensional DCT is just a separable product of the inverse(s) of the corresponding one-dimensional DCT , e.g. the one-dimensional inverses applied along one dimension at a time

Symmetry – Another look at the row and column operations reveals that these operations are functionally identical. Such a transformation is called a symmetric transformation. – A separable and symmetric transform can be expressed in the form – where A is a NxN symmetric transformation matrix

$$T = AfA$$

Computational efficiency – Inverse transform – DCT basis functions are orthogonal. Thus, the inverse transformation matrix of A is equal to its transpose i.e. A-1= A T. This property renders some reduction in the pre-computation complexity.

$$f = A^{-1}TA^{-1}.$$

KLT (Karhunen – Loeve Transform)

The KL Transform is also known as the Hoteling transform or the Eigen Vector transform. The KL Transform is based on the statistical properties of the image and has several important properties that make it useful for image processing particularly for image compression.

It can be also known as

1.Eigen vector transform 2.The method of Principal Components

It is based on the statistical properties of vector representation. As we know the important statistical properties like Mean and Variance

Mean is $m_x = E\{X\}$ Where E – Expected Value

Variance

The variance of a random variable X is the expected value of the squared deviation from the mean of X, $\mu = \mathrm{E}[X]$:

$$\operatorname{Var}(X) = \mathrm{E}\left[(X-\mu)^2\right].$$

This definition encompasses random variables that are generated by processes that are discrete, continuous, neither, or mixed. The variance can also be thought of as the covariance of a random variable with itself:

$$\operatorname{Var}(X) = \operatorname{Cov}(X, X).$$

The variance is also equivalent to the second cumulant of a probability distribution that generates X. The variance is typically designated as $\operatorname{Var}(X)$, or sometimes as $V(X)$ or $\mathbb{V}(X)$, or symbolically as σ^2_X or simply σ^2 (pronounced "sigma squared"). The expression for the variance can be expanded as follows:

$$\begin{aligned}\operatorname{Var}(X) &= \mathrm{E}\left[(X-\mathrm{E}[X])^2\right]\\ &= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right]\\ &= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2\\ &= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2\end{aligned}$$

Then Covariance $C_{ij} = C_{ji} = 0$

The basic function of KLT is orthogonal eigen vectors of the covariance matrix. KLT optimally decorrelates the input data.

Drawbacks of KLT

1. Basic function has to be calculated for each signal model and has no mathematical structure.

2.KLT requires $O(m^2)$ Multiply/ADD operations. But DFT require $O(log_2^m)$multiplications.

Applications

1. It is used for clustering analysis to determine new co-ordinate system for sample data. This is known as PCA(Principal Component Analysis).

2. It contains largest number of zero valued components. Therefore it can be used for compression.