Unit - IV

Segment and Animation

DATE PAGE

Prof. A. D. Sawant

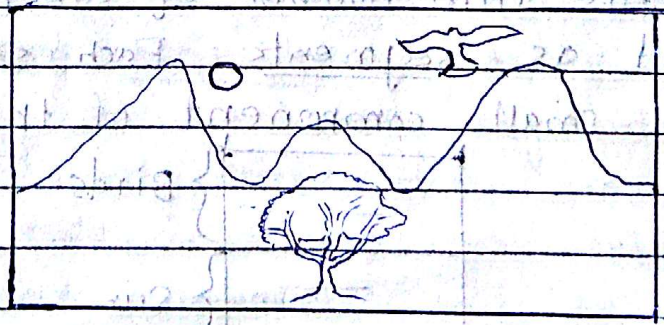
Syllabus :

Segment : Introduction, segment table, segment creation, closing, deleting and renaming, visibility
Animation : Introduction, Design of animation sequences, Animation languages, Key-frame, Morphing, Motion specification
Colour Models and applications : properties of light, CIE chromaticity diagram, RGB, HSV, CMY, YIQ, colour selection and applications.

Segment : Introduction

We have discussed how to display a perticular object or image on screen. To display the object we are storing the commands in display file, which are necessary to draw that object on screen. But in real life the pictures or scenes are not made up of single object, but it is a combination of several small sub pictures.

Suppose we want to display the scene which is shown in figure.

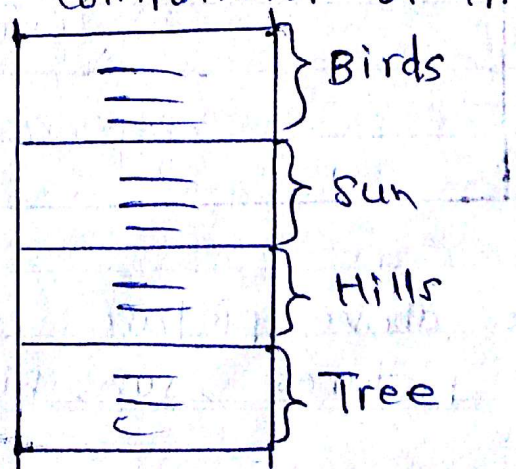


The above picture is a combination of four sub pictures, sun, Bird, Hills and Trees.

If we want to do some transformations i.e. Bird is flying, we can do this by two ways first is, we treat bird is foreground image or sub picture and othe part is background image, then move the foreground image from its original position to new position. and keep the background image as it is. Second is, move the background image and keep foreground image as it is.

But if we have to move more than one subpictures, which is time consuming process we select first method in which only bird is moving. so we have to apply some transformations on bird subpicture. only and no need to apply on rest of the pictures.

The total number of commands which are required to draw the picture are grouped into four groups. lets say total out of 20 commands 5 are for Bird, 5 are for sun, 5 are for Hills and 5 are for tree. Each group represents commands required to draw a picture. It means we are dividing the display file into number of sub parts, which is called as segments. Each segment represent small component of the overall scene.



After defining the segments we have to set the attributes of each segments. Attributes include visibility and image transformations. Visibility attribute includes whether the segment is visible or not. The segment whose visibility attribute is set ON that segment image is visible and if visibility attribute is OFF then that segment is not visible. Similarly to make changes in image we are using another attribute of a segment which is called as image transformations. We can change image size by scaling transformation, we can rotate the image or can shift the image.

We can perform different operations on segments such as create, close, delete and rename of the segments.

Segment Table :-

We have to set attributes like visibility and image transformation for each segment. we need one more attribute which is called segment name. A segment name attribute will uniquely identify a particular segment so we have give an unique name to each segment. to store the visibility need visibility attribute To display a particular segment of display file, we must know the commands of that segment this is known by determining the where segment begins and how many commands are there for that segment so segment start and segment size attributes are required.

So we need to organize this information so we are forming a table which is called as segment table.

Segment Name	Segment Start	Segment size	Visibility	Scale X	Scale Y	---
1	-	-	ON	-	-	-
2	-	-	OFF	-	-	-
3	-	-				

Table 1. Segment table.

We will use arrays to store the attributes of a segment. We can use array index as a segment name, so segment name is used as numbers. we can very easily modify the attributes of the segments using array name as segment table ~~name~~ attribute, and array subscript as segment name.

If we want to set the visibility of the attribute of segment name 2, then we can.

Visibility [segment name] = ON/OFF

Here ~~in~~ segment name will act as an index in attribute arrays.

We are storing command in display file, then these commands are interpreted by display file interpreter. with the help of segment table.

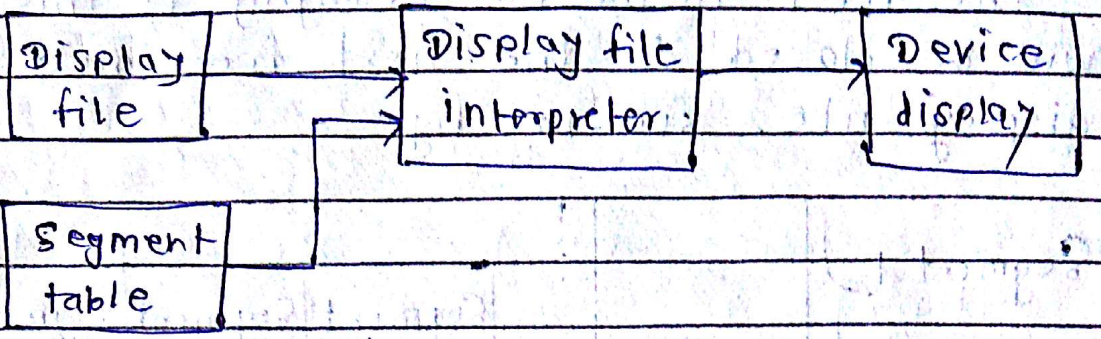


fig:- Display file interpreter.

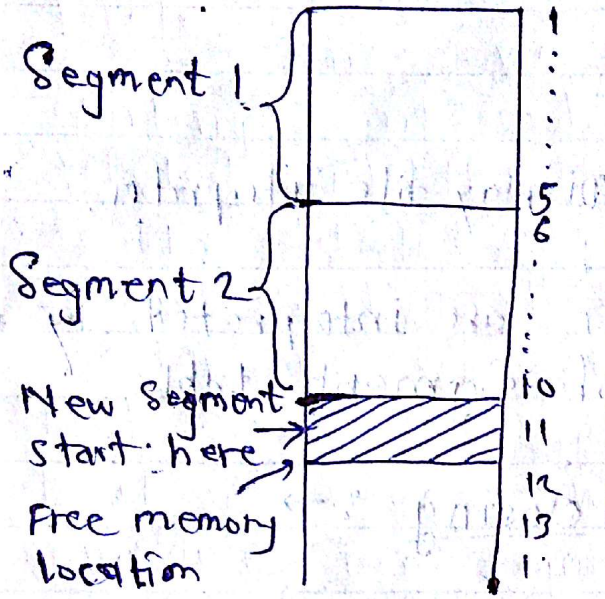
- Display file commands are interpreted by interpreter in the consultation of segment table.

Segment Creation and closing :→

- To create a new segment first we have to check whether any other segment is open or not we can not have two segments open at a time because we can't get to which segment we assign the drawing commands. If we try to open other segment then it will give us error.

If no segment is open we have to check for valid name of the segment.
If segment name is valid then we have to check is there any duplicate segment name is present?
If no we have to check newly written command whether belongs to newly created segment or not. Then we have initialize the terms in segment table to indicate that it is fresh new segment.

The first instruction belonging to this segment will be placed to the next free area of the display file. see in fig.



Display file

Segment name	Segment Start	Segment Size	Visibility
1			
2			
3	11	0	ON

New segment

Segment table

For new segment, we have to say,
Segment start [segment Name] = Next free location in display file.
The current size of the segment ~~is~~ will be zero, as there is no instruction yet into it.
and attributes are initialized to default attribute values, such as visibility ON.
If we have to check whether any other segment is open or not, we need one variable or flag. When we create new segment then we set that flag to the segment name. When no segment is opened at that time set the flag to zero. That's why we are not giving name zero to any segment. If we have completed drawing commands of segment we have to close current segment for this we are making use of that ~~segm~~ variable flag. We will move zero to that flag to close current segment.

Segment Deletion :-

If a segment is no longer needed or there is no work left of that segment, we would prefer to delete that segment to save space. But when we are deleting a segment at that time other segments should not get changed.

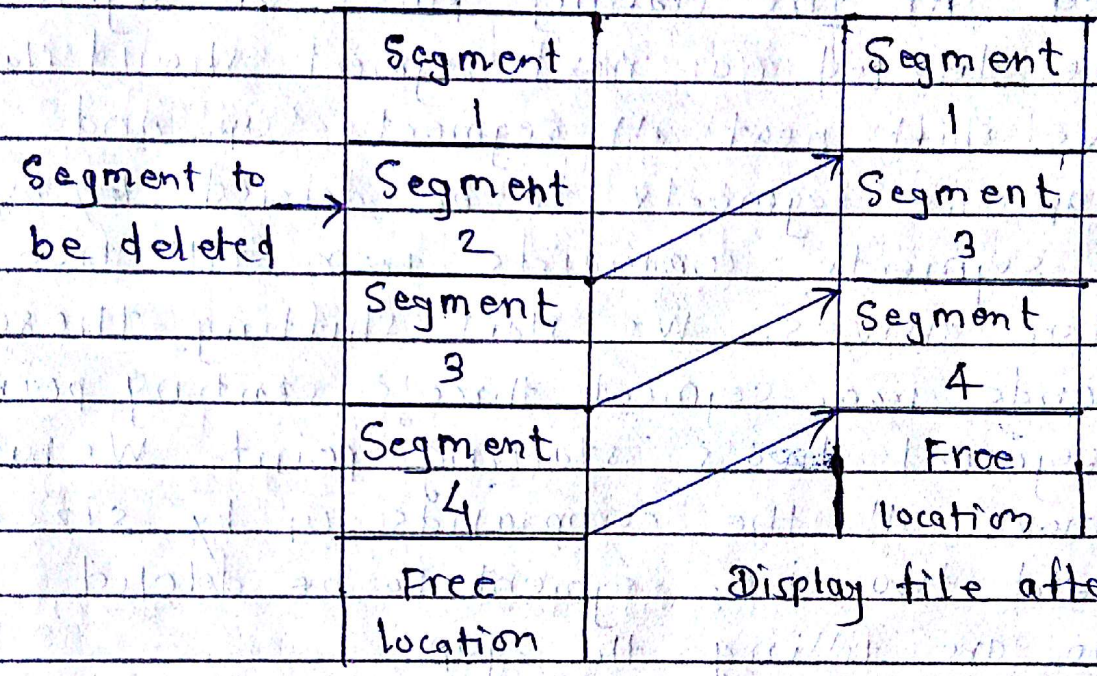


Fig: Deletion of Segment.

Let us see the example of hills, bird, sun and tree we have stored each sub picture in a separate segment so there will be four segments. Image is stored in segment 2. If we don't want to display bird's image we can set segment visibility to OFF, but if it is no longer needed then we have to delete that segment from display file to save memory at that time other segments should not get disturbed.

To delete the segment first we check segment name is valid or not, if valid next we

check, the segment size must be non-zero. then only we can delete the segment. If segment size is zero(0), there is no command in that segment; no need to delete. Then we have to know the starting point and size of segment to be deleted. At the starting point of segment to be deleted, our next segment should start. so we shift next all segments up and overlap the segments to be deleted, by next segments commands till free location arrives. We start shifting these commands from segment three's starting point to segment two's starting point. We have to move all the commands up by sizeof segment two i.e. segment to be deleted. So we are filling the gap.

Next we have to update the segment table. Below figure shows the segment table before deletion and after deletion.

Before deletion of Segment 2:

Segment Name	Segment start	Segment size	Visibility	.....
1	1	5	ON	
2	6	3	ON	
3	9	5	ON	
4	14	6	ON	

After deleting segment 2 we have to modify the starting addresses of the segments which

are after segment 2, by the size of segment 2. For segment 2 we have to we have to write segment size as zero (0) to indicate segment 2 does not contain anything. The visibility attribute of the segment which we want to delete may be kept as ON/OFF, or anything because anyway the size of segment is zero (0).

We can delete all the segments at a time we have to set size of all the segments to zero and initialize the free index of display file to the first location and set starting position to one.

Renaming a Segment :-

After deletion of Segment 2 :-

Segment Name	Segment start	Segment Size	Visibility	---...
1	1	5	ON	
2	6	0	OFF/ON	
3	9-3=6	5	ON	
4	14-3=11	6	ON	

size of segment to be deleted

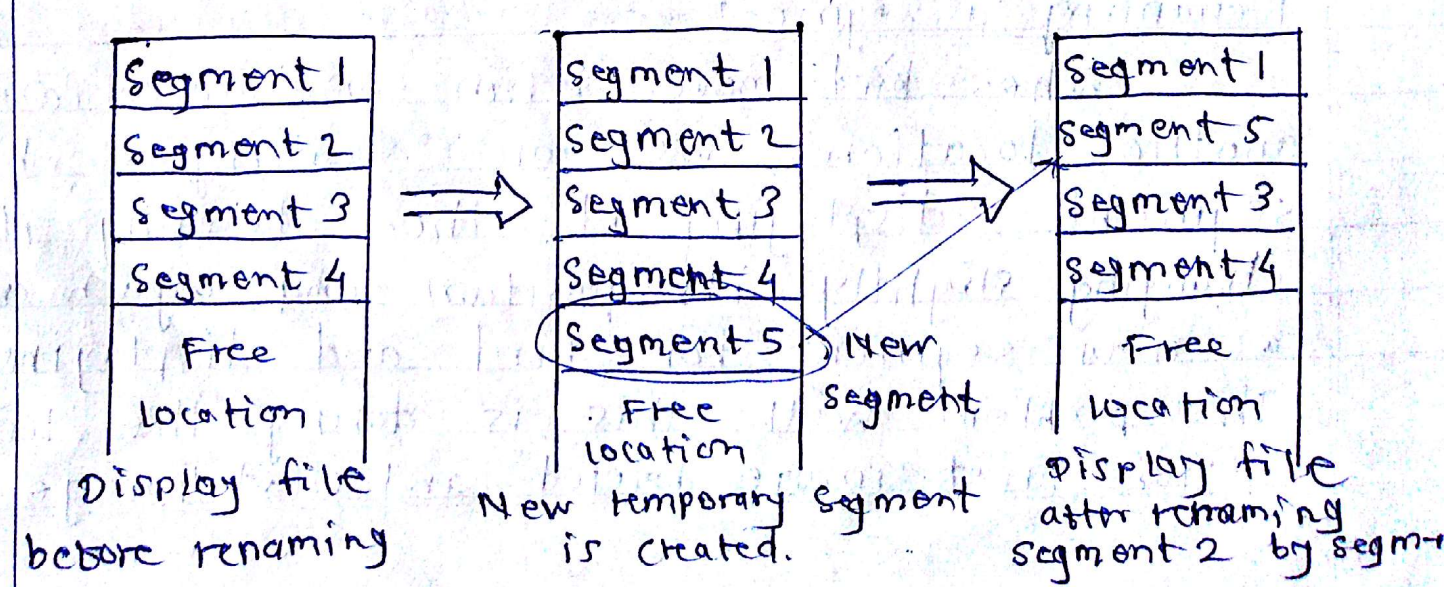
Renaming a Segment :-

When bird moves from one location to another location, we are creating a bird segment, displaying it, then deleting it, changing slightly its position and again creating a new segment for bird and displaying it. The problem with this is during the time after first image deleted and next image is

complete, only partially completed image may get displayed. So to avoid this problem we should not delete a segment till the replacement for it is completed. This means that both the segments must be present in display file at the same time.

We will build a new invisible image under some temporary name. When it is completed we can delete the original image and make replacement image visible. The idea of maintaining two images, one to show and one to modify is called as double buffering.

To do this, we have created four segments and if we want to rename the second segment by new segment then we are creating segment five, as a temporary segment with its visibility as OFF. so that it will not get displayed. Once the segment 5 is completed then we are deleting segment 2 by copying all the ~~attributes~~ commands to segment 2 from segment 5 and we delete the newly created segments.



Display file before renaming

New temporary segment is created.

Display file after renaming segment 2 by segm-

In the segment table all the attributes of segment 2 are copyied to new created segment 5 attributes and we make segment 2 size is zero to indicate that it is deleted. Here we are not copying the segment name. we also check for valid segment name and segment name must be unique

Segment table before renaming :-

Segment Name	Segment Start	Segment Size	Visibility	...
1	1	4	ON	
2	5	10	ON	
3	15	5	ON	
4	20	5	ON	

Segment table having both segments two and five

Segment Name	Segment Start	Segment Size	Visibility	...
1	1	4	ON	
2	5	10	ON	
3	15	5	ON	
4	20	5	ON	
5	-	-	OFF	

Segment table after renaming Segment 2 as 5 :

Segment Name	Segment Start	Segment Size	Visibility	...
1	1	4	ON	
2	5	0	ON/OFF	
3	15	5	ON	
4	20	5	ON	
5	5	10	ON	

Color Models and Applications :-

properties of Light :-

We perceive light or color as narrow frequency band within the electromagnetic spectrum. Other few frequency bands within this spectrum are called radio waves, microwaves, infrared waves, and x-rays low-frequency end is a red color (4.3×10^{14} hertz) and high-frequency end is violet color (7.5×10^{14} hertz) spectral colors range from the reds through orange and yellow at the low-frequency band to green, blue and violet at the high end

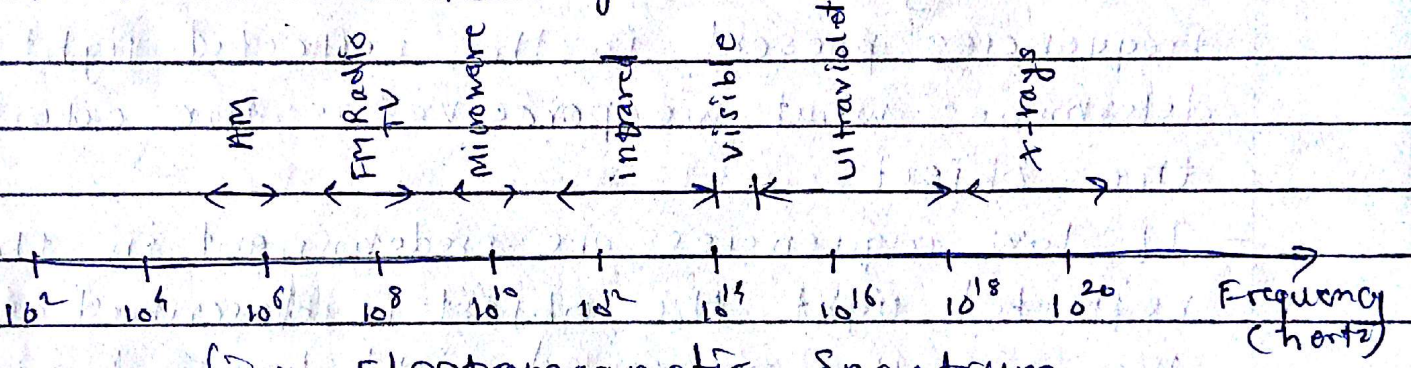


fig: Electromagnetic Spectrum

Light is an electromagnetic wave, we represent in wavelength or frequency of the monochromatic wave are inversely proportional to each other, with the proportionally as the constant.

$$c = \lambda f \qquad f = \frac{c}{\lambda} \quad \text{or} \quad \lambda = \frac{c}{f}$$

Frequency is constant for all material, but the speed of light and wavelength are material dependent dependent. In a vacuum, $c = 3 \times 10^{10}$ cm/sec. Light wavelengths are very small, so length units for designating spectral colors are ($1A^\circ = 10^{-8}$ (m) in Angstroms or nanometer ($1\,nm = 10^{-7}$ cm). Equivalent term form for nanometer is millimicron.

so light at the red end of the spectrum ~~is~~ has a wavelength of approximately 700 nm (nanometer) and wavelength of violet light at the other end of spectrum is about 400nm.

- A light source such as the sun or a light bulb emits all frequencies within the visible range to produce white light.
- When white light is incident upon an object, some frequencies are reflected and some are absorbed by the object. The combination of all frequencies present in the reflected light ~~are~~ determines what we perceive as the color of the object.
- If low frequencies are predominant in the reflected light, the object is described as red.
- We say the perceived light has a dominent frequency (or dominent wavelength) at the ~~rend~~ red end of the spectrum.
- The dominent frequency is also called as the hue or simply the color of light.

- Brightness is the perceived intensity of light. Intensity is the radiant energy emitted per unit time, per unit solid angle, and per unit projected area of the source. Radiant energy is related to the luminance of the source.

- Purity or saturation of the light describes how washed out or how "pure" the color of the light appears.

- Pastels and pale colors are described as less pure.
- The term chromaticity is used to refer collectively to the two properties describing color characteristics purity and dominant frequency.
- Energy emitted by a white-light source has a distribution over the visible frequencies as shown in figure.

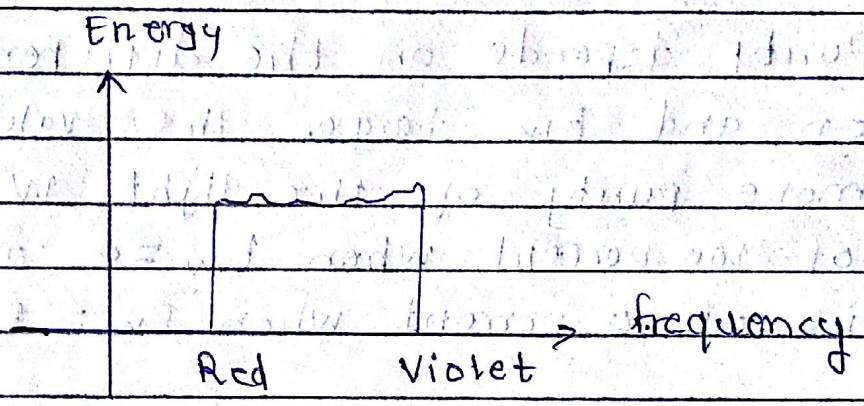


fig: Energy distribution of a white-light source.

- Each frequency component within the range contributes less or more to the total energy of light, so the color of light is described as white.
- When dominant frequency is present then we describes the color of the light is corresponding to the dominant frequency.

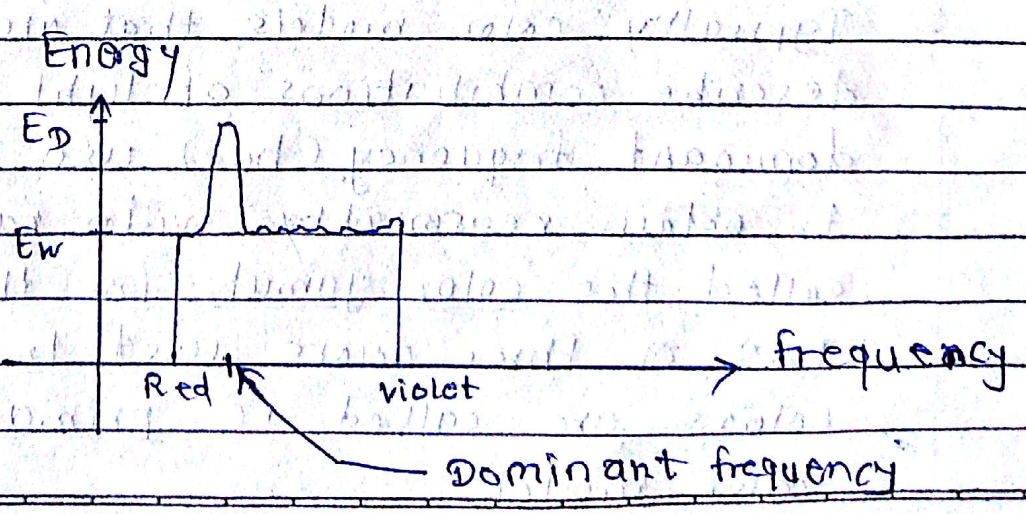


fig: Energy distribution of light with dominant frequency.

- The energy density of the dominant light component is labeled as E_D and the contributions from the other frequencies produce while light of Energy density E_w.

- The brightness of the source as the area under the curve, which gives total energy density emitted.

- Purity depends on the difference between E_D and E_w. larger the value of E_D more purity of the light. We have purity of 100 percent when $E_w = 0$ and a purity is of 0 percent when $E_w = E_D$.

- We can view color of the light formed by a combination of two or more sources. the complementary colors are ~~used to~~ combined used to produce white light such as red and cyan, green and magenta, blue and yellow

- By combining two or more colors we can form wide range of other colors.

- Typically color models that are used to describe combinations of light in terms of dominant frequency (hue) use three colors to obtain reasonably wide range of colors, called the color gamut. for that model. Two or three colors, used to produce other colors are called as primary colors color model.

CIE Chromaticity Diagram :

No finite set of light sources can be combined to display all possible colors, three standard primaries were defined in 1931 by the International Commission on Illumination, referred to as the CIE (Commission Internationale de l'Eclairage). The three standard primaries are imaginary colors They are defined mathematically with positive color-matching functions. as shown in figure

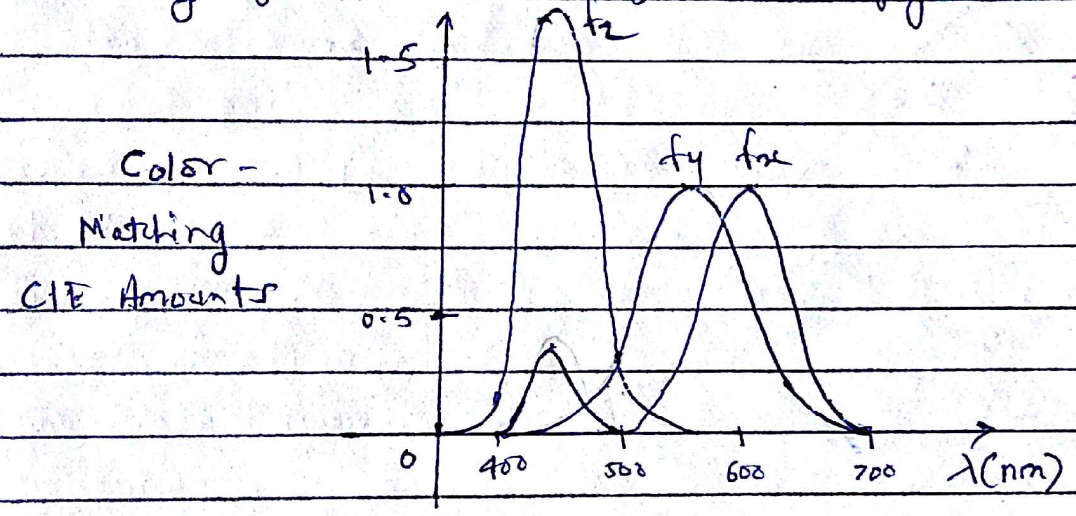


fig:- The three color-matching functions for CIE primaries.

The positive color matching functions are used to specify the amount of each primary need to describe the any spectral color. This provides an internation standard definition for all colors. CIE primaries diminate negative value color matching. The primary colors are virtual colors A, B, C. Then for a given real color, its components are.

$x = A / (A+B+C)$

$y = B / (A+B+C)$

$z = C / (A+B+C)$

since $x+y+z = 1$

if x & y are known then z can be determined as

$z = 1 - x - y$

The CIE Chromaticity Diagram :-

We plots the ~~am~~ normalized amounts x and y for colors in the visible spectrum. We obtain the tongue-shaped curve shown in figure. This curve is called the CIE Chromaticity diagram

- Points along the ~~color~~ curve are the spectral colors (pure colors).
- The line joining the red and violet spectral point referred as the purple line, is not part of spectrum
- Interior points represent all possible visible colors combinations.

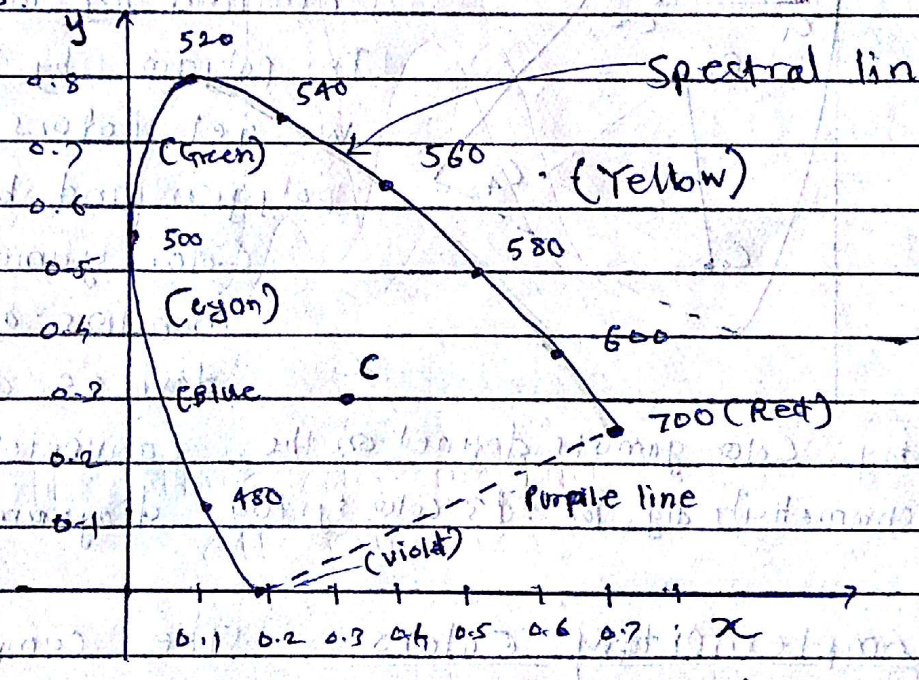


fig: CIE chromaticity diagram for the spectral colors from 400 nm to 700 nm

- Point C in the diagram corresponds to the white-light position. This point is for a white light source known as illuminant C, which is used as standard approximation for average daylight
- Luminance values are not available in the chromaticity diagram because of normalization
- This diagram is useful for

→ Comparing color gamuts for different sets of primaries.
→ Identification of complementary colors.
→ Determining purity and dominant wavelength for a given color.

* Color Gamuts : We identify color gamuts on chromaticity diagram as straight line segments or polygon regions. Colors are obtained by mixing C_1 & C_2. In polygon by mixing C_3, C_4, C_5 we get colors inside the polygon and boundary of polygon. Color gamuts for video monitors and hard-copy devices are compared conveniently on chromaticity diagram.

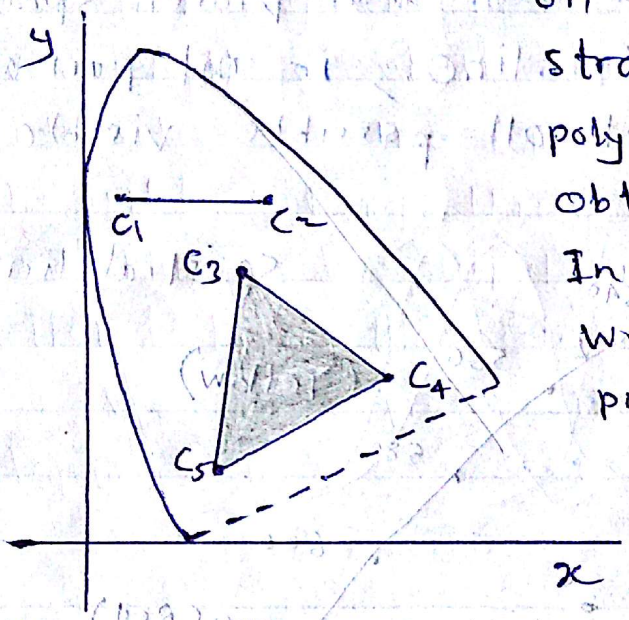


fig: Color gamuts defined on the chromaticity dig. for 2 & 3 color system. diagram.

* Complementary Colors : The complementary colors must be represented on the chromaticity diagram as two points on opposite sides of c and colinear with C. The distances of the two colors C_1 and C2 to C determine the amounts of each needed to produce the white light.

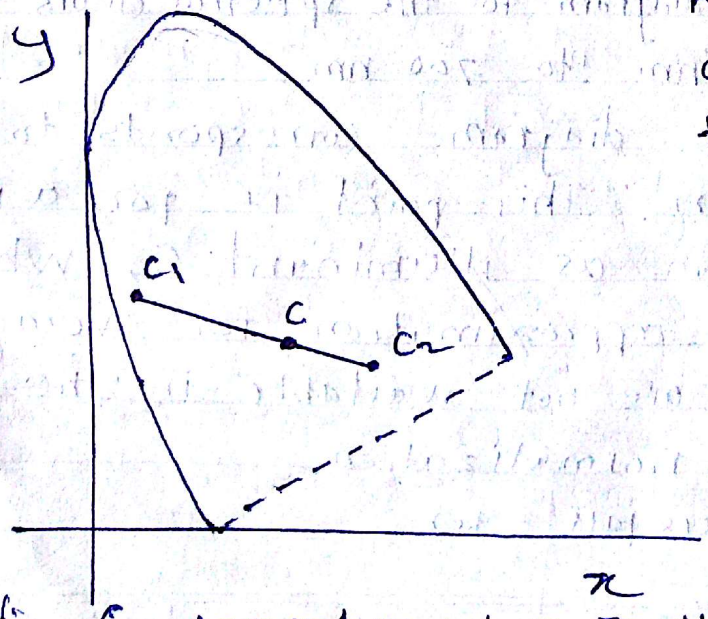


fig: Complementary colors on the

Dominent Wavelength :- To determine the dominent color we draw a straight line from C through that color point to a spectral color on the chromaticity curve. The spectral color C_s is the dominent wavelength for color C_1. The color C_1 can be represented as a combination of white light C and spectral color C_s.

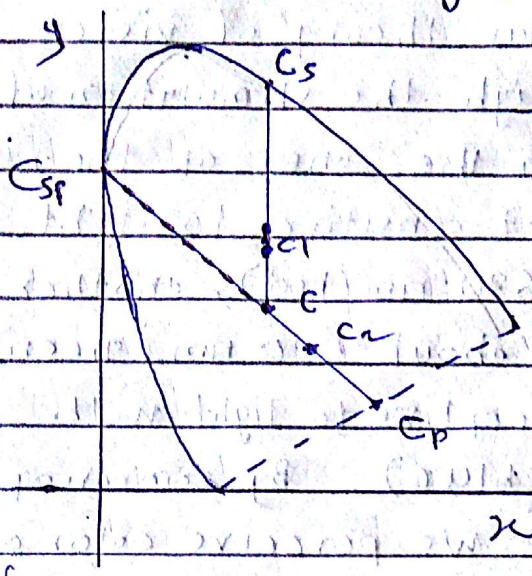


fig: Determining dominent Wavelength and purity using the chromaticity dig.

This will not work for color points both C and purple line which is not in the visible spectrum. In this case we take the complement of C_p which is C_{sp} as the dominent wavelength. C_2 is obtained by subtracting the spectral dominant wavelength from white light.

Purity :-

We determine the purity of C_1 as the relative distance of C_1 from C along the straight line joining C to C_s. If d_{c_1} denotes the distance from C to C_1 and d_{cs} is the distance from C to C_s, we represent purity as the ratio d_{c_1}/d_{cs}. Color C_1 is 25% pure, because it is situated at about one-fourth the total distance from C to C_s. At position C_s, the color point would be 100% pure.

RGB Color Model :

According to tristimulus theory of vision our eye perceive color through the stimulation of the three visual pigments in the cones of the retina. One of the pigments is most sensitive to light with a wavelength of about 630 nm (red), another has its peak sensitivity at about 530 nm (green), and third pigment is most receptive to light with wavelength of about 450 nm (blue). By comparing intensities in the light source we perceive color of the light. Using three primary colors red, green and blue color output on video monitor is displayed refered to as the RGB Color Model.

We can represent this model using unit cube defined on R, G, B axes.

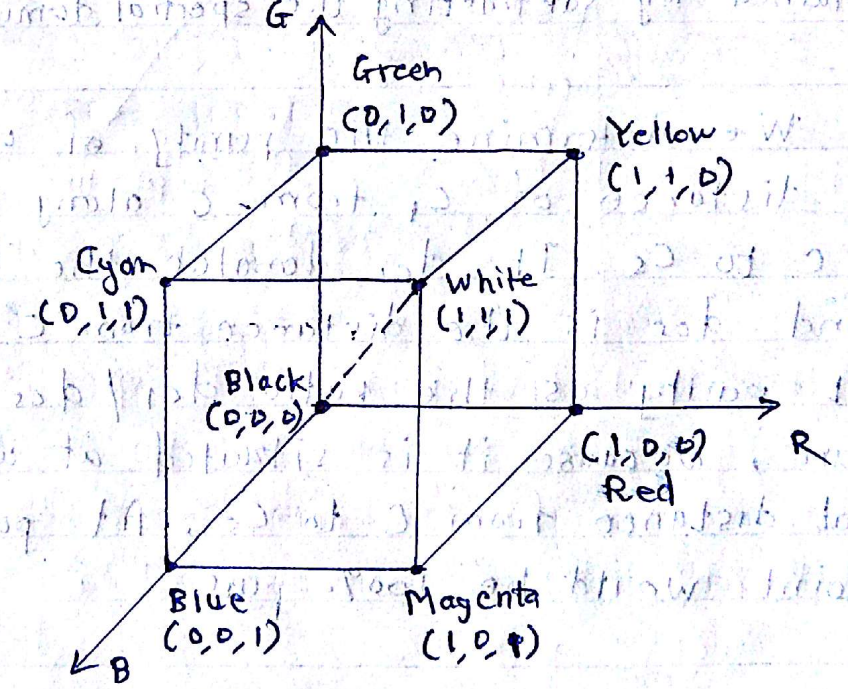


Fig: RGB Color Model, Any color within the unit cube can be described as an additive combination of the three primary colors.

The origin represents black (0, 0, 0) and diagonally opposite vertex with coordinates (1, 1, 1) i.e white. Vertices of the cube on the axes represent the primary colors and remaining vertices are complementary color points for each of the primary colors.

With XYZ color system the RGB color scheme is an additive model. Each color point within unit cube can be represented as a weighted sum of the primary colors, using unit vectors R, G, and B

$$C(\lambda) = (R, G, B) = RR + GG + BB$$

Where parameters R, G, B are assigned values in the range from 0 to 1.0. For Example the magenta vertex is obtained by adding maximum red and blue to values to produce the triple (1, 0, 1) and white at (1, 1, 1) is the sum of the maximum values for red, green, and blue. Shades of the gray are represented along the main diagonal of the cube from the origin (black) to the white vertex. Points along this diagonal have equal contribution of primary colors. Gray shade halfway between black and white is represented as (0.5, 0.5, 0.5)

Chromaticity co-ordinates for the National Television System Committee (NTSC) standard RGB phosphors are listed in Table

	NTSC standard	CIE Model	Approx. Color Monitor value
R	(0.670, 0.330)	(0.735, 0.265)	(0.628, 0.346)
G	(0.210, 0.710)	(0.274, 0.717)	(0.268, 0.588)
B	(0.140, 0.080)	(0.167, 0.009)	(0.150, 0.070)

The following figure shows the approximate color gamut for the NTSC standard RGB primaries:

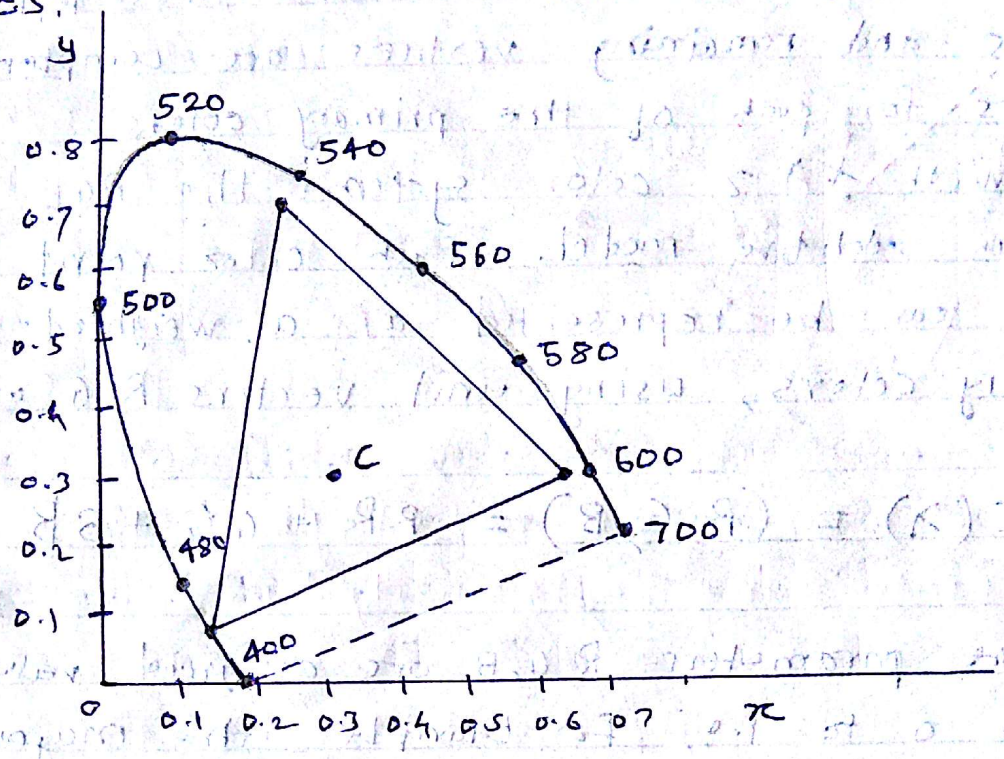


Fig:- The RGB color gamut for NTSC chromaticity coordinates. Illuminant C is at position (0.310, 0.316), with a luminance value of $Y = 100.0$

The YIQ Color Model :-

An RGB graphics Monitor requires separate signals for the red, green and blue component of an image, a television monitor uses a composite signal. NTSC color encoding for forming the composite video signal is called the YIQ color model.

The YIQ parameters :- The parameter Y is the same as the Y component in the CIE XYZ color space. The luminance (brightness) information is conveyed by the Y parameter, while chromaticity information (hue and purity)

is incorporated into the I and Q parameters.
- A combination of red, green and blue is choosen for the Y parameter to yield the standard luminosity curve. Because Y contains the luminance information, black and white television monitor use only the Y signal
- Parameter I contains orange-cyan color information that provides the flash-tone shading.
- Parameter Q carries green-magenta color information.
Black-and-white television monitor obtains grayscale information for a picture within a 6-MHz bandwidth, but luminance and chromaticity values are encoded on separate analog signals. Y value is conveyed as an amplitude modulation on carrier signal with a bandwidth of about 4.8 MHz. Chromaticity information I and Q is combined on a second carrier signal that has bandwidth of about 1.8 MHz. carries color information on this carrier.
- Amplitude modulation encoding transmits I value using about 1.3 MHz of bandwidth and phase-modulation encoding (quadrature signal) using about 0.5 MHz carries the Q value.
- Luminance values are encoded at higher precision in the NTSC signal (4.2 MHz bandwidth) than the chromaticity values (1.8 MHz bandwidth) because we can detect small brightness changes more easily compared to small color changes.
- Luminance value for an RGB color. For producing chromaticity values subtract the luminance from the red and blue components of the colors.

$$Y = 0.299\,R + 0.587\,G + 0.114\,B$$

$$I = R - Y$$

$$Q = B - Y$$

* Transformation Between RGB and YIQ Color Spaces

An RGB color is converted to a set of YIQ values using an NTSC encoder and modulates the carrier signals.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.701 & -0.587 & -0.114 \\ -0.299 & -0.587 & -0.886 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Conversely an NTSC video signal is converted to RGB color values using an NTSC encoder, which first separates the video signal into the YIQ components and then converts the YIQ values to RGB values.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 1.000 & 0.000 \\ 1.000 & -0.509 & -0.194 \\ 1.000 & 0.000 & 1.000 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

CMY Color Model:

A video monitor displays color patterns by combining light that is emitted from the screen phosphors, which is an additive process. However, hard-copy devices, such as printers and plotters, produce a picture by coating a paper with color pigments. We see color the color patterns on the paper by reflected light which is subtractive process.

The CMY Parameters:- The subtractive color model can be formed with the three primary colors cyan, Magenta, and Yellow. Cyan can be described as a combination of green and blue. When

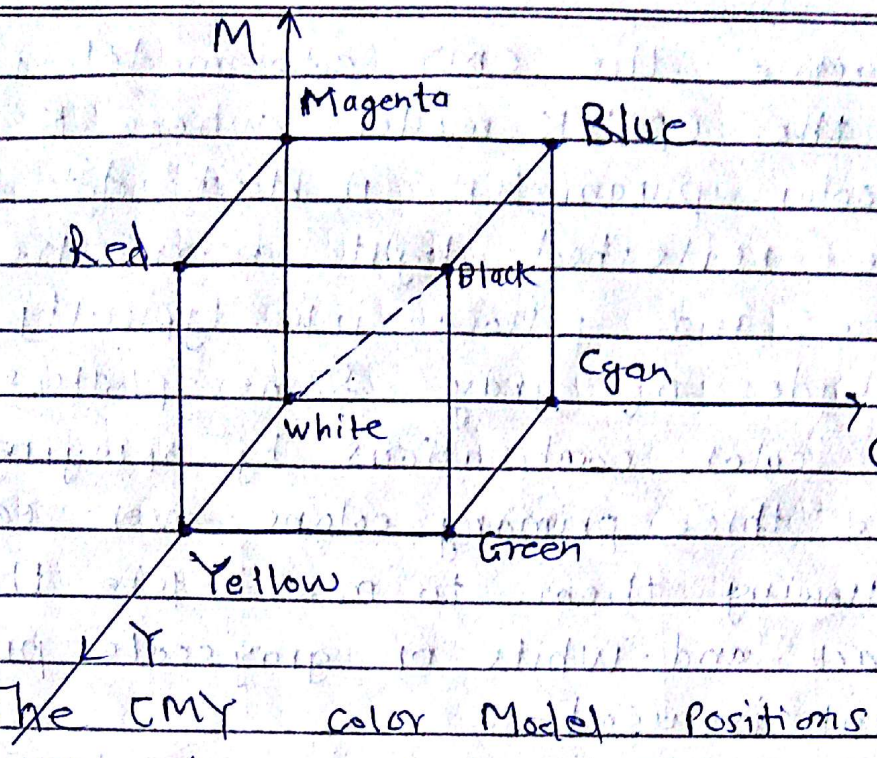


fig:- The CMY color Model. Positions within the unit code are described by subtracting the specified amounts of the primary colors from white.

White light is reflected from cyan colored ink, the reflected light contains only the green and blue components and the red component is absorbed or subtracted by the ink. Similarly Magenta ink subtract the green component and yellow ink subtract the Blue component from incident light.

In CMY model the spatial position (1, 1, 1) represent black, because all components of incident light are subtracted. Origin represents white. Equal amount of each of primary colors produce shades of gray along the main diagonal of the cube. A combination of cyan and magenta ink produces blue light because red and green components of incident light are absorbed. Similarly for others.

The CMY printing process often uses a collection of four ink dots which are arranged in a close pattern somewhat as an RGB monitor uses three phosphor dots.

In practice, the CMY color model is referred to as the CMYK model. where K is the black color parameter. A black dot is included because reflected light from the cyan, magenta, and yellow ink typically produce only shades of gray. Some plotters produce different color combinations by spraying the ink for the three primary colors over each other and allowing them to mix before they dry. For black-and-white or grayscale printing, only black ink is used.

* Transformations Between CMY and RGB color Space

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

where the white point in RGB space is represented to the unit column vector.

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

In this unit column vector represents the black point in the CMY color space.

for RGB $\longrightarrow$ CMYK

$K = \max(R, G, B)$

$C' = C - K$

$M' = M - K$

$K' = Y - K$

CMYK $\longrightarrow$ RGB.

$K = \min(R, G, B)$

$R' = R - K$

$G' = G - K$

$B' = B - K$.

HSV Color Model :-

We can give a color specification in an intuitive model by selecting a spectral color and the amount of white and black that are to be added to that color to obtain different shades, tints, and tones.

HSV parameters :-

Color parameters in this model are called hue (H), Saturation (S) and value (V). We derive this three-dimensional color space by relating the ~~cube along the~~ HSV parameters to the directions in the RGB cube.

If we imagine viewing the cube along the diagonal from the white vertex to the origin (black), we see outline of the cube as a Hexagon shape. The boundary of the hexagon represents different hues. In HSV space Saturation S is measured along a horizontal axis, and value parameter V is measured along a vertical axis through the ~~vertex~~ center of the hexcone. Here H is represented as an angle about the vertical axis ranging from 0° at red ~~60°~~ through 360°. Vertices of the hexagon are separated by 60° intervals. Yellow is at 60°, green at 120° and cyan (opposite the red point) is at $H = 180°$. Complementary colors are 180° apart.

Saturation S is used to designate the purity of a color. A pure color (spectral color) has the value S = 1.0, and decreasing S values tend toward the grayscale line S=0 at the center of the hexcone.

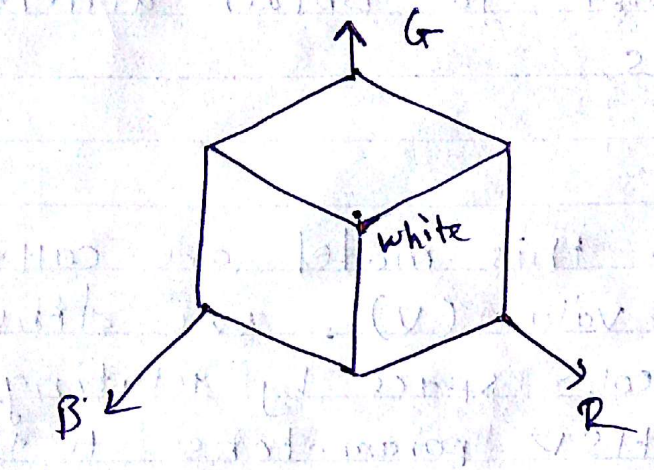


fig: RGB Color Cube.

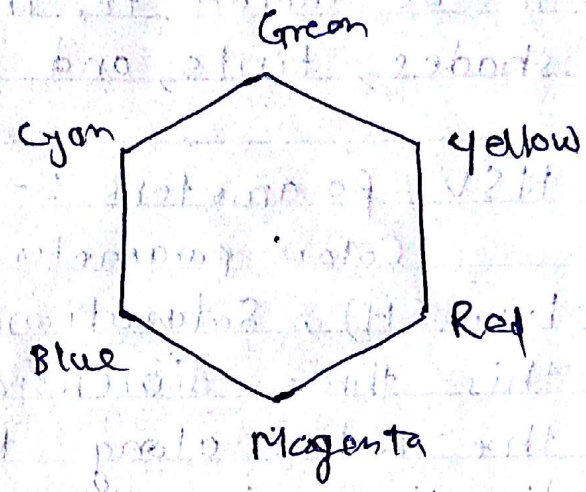


fig: Color Hexagon

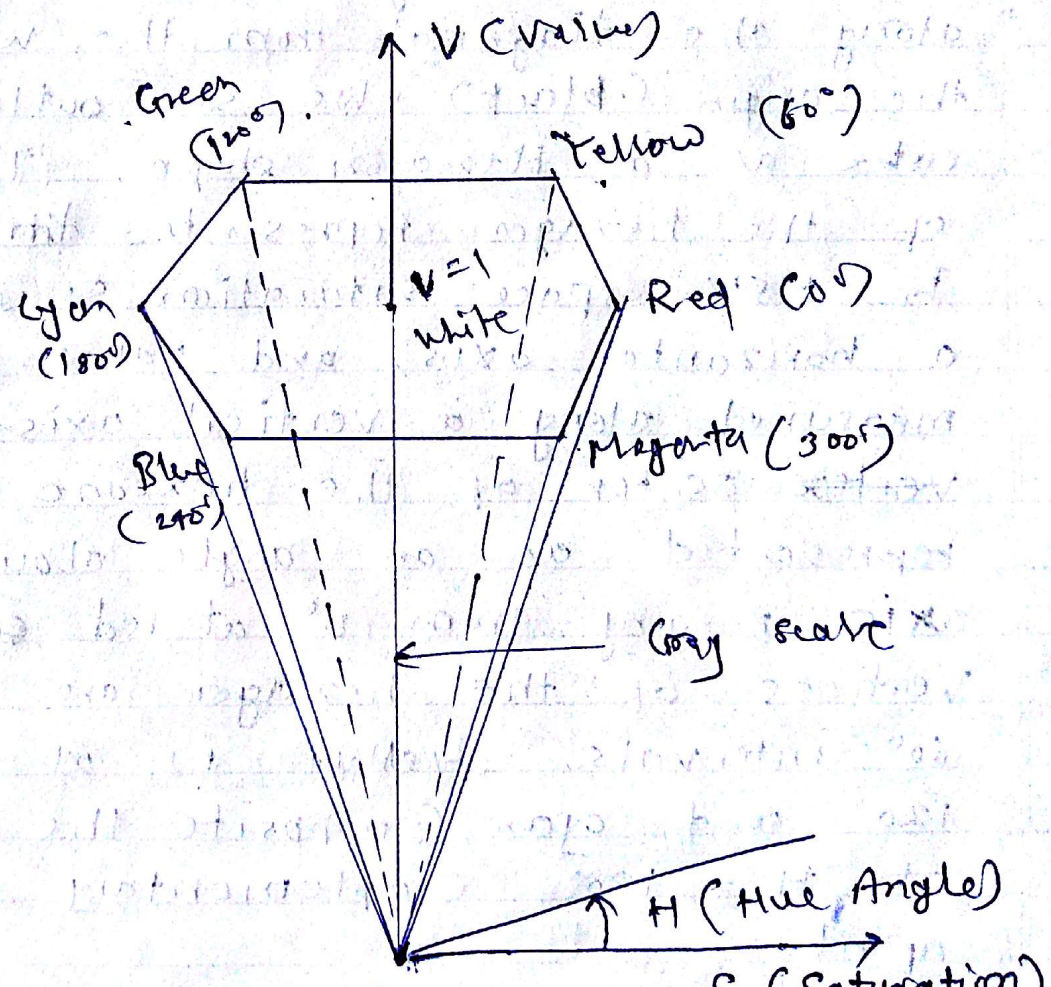


fig: HSV Hexcone

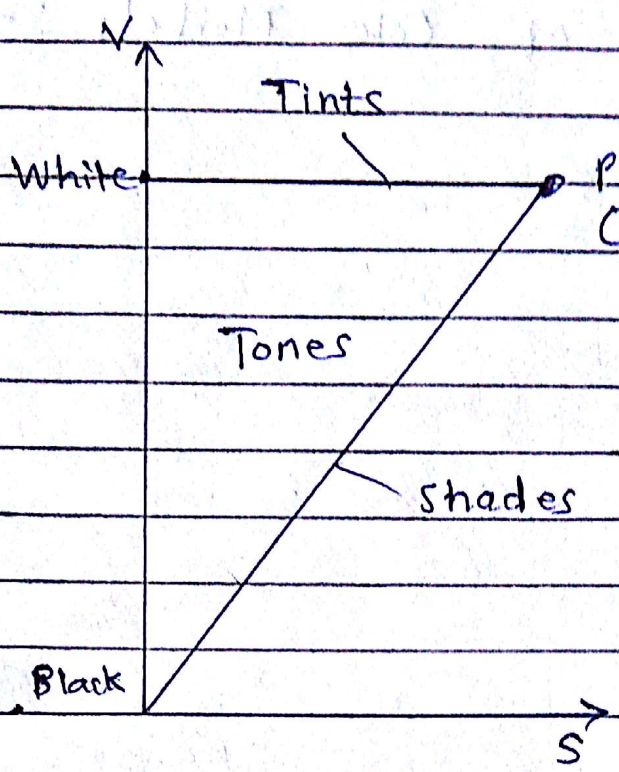


fig: Cross section of the HSV hexcone. showing regions for shades, tints and tones.

eg:

128 - different Hues
130 - different tints (saturation levels)
23 - shades are discernible with yellow color

128 X 130 X 23 = 382720 diff colors.

graphics application

128 hues, 8 saturation levels, 16 values.
total 16,384 colors.