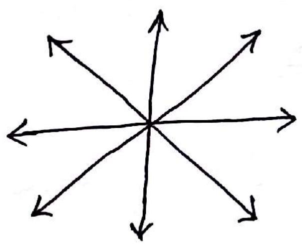
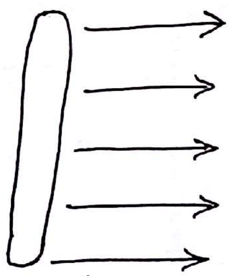
* Basic Illumination Model :-

An illumination model, also called as lighting model and sometimes referred to as a shading model, is used to calculate the intensity of light that we see at a given point on the surface of an object. A surface rendering algorithm uses an illumination model to determine the light intensity.

Basic illumination models are generally implemented for calculating light intensity ~~light~~ Intensity calculations are based on the optical properties of surfaces. optical parameter controls the amount of reflection and absorption of incident light. The light source may be a point light source or a parallel beam of light source.



(Point Light Source)



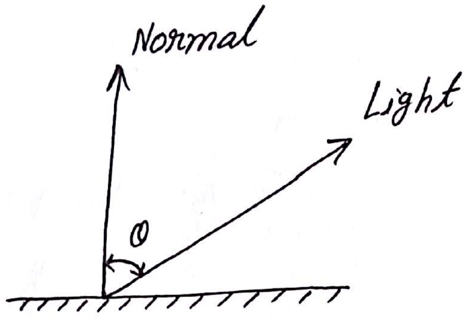
(Parallel Beam of Light Source)

Ambient Light :-

An ambient light is used to produce a uniform illumination. Ambient light has the same effect on the entire surface and has no directional effect. The amount of ambient light incident on each ~~surface~~ object is a constant for all surfaces and over all directions.

Lambert's Law :-

The diffuse reflections from the surface are scattered with equal intensity in all directions, independent of the viewing direction. Such surfaces are sometimes referred to as ideal reflectors. They are also called as Lambertian reflector, since rediated light energy from any point on the surface is governed by Lambert's Cosine law.



This law states that, the reflection of light from a surface varies as the cosine. If the incident light ~~is~~ from the source is perpedicular to the surface, that point on the surface is fully illuminated. On the otherhand, as the angle of illumination moves away from the surface normal, the brightness of the point drops off.

The expression for the brightness of an object illuminated by ambient light can be given as –

$$\boxed{I_{ambdiff} = K_a I_a}$$

where I_a is the intensity of the ambient light

K_a is the ambient reflection coefficient

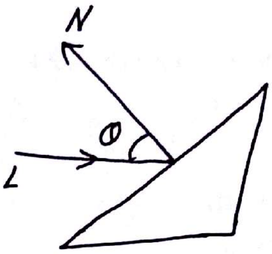
$I_{ambdiff}$ is the intensity of diffuse reflection

1) Diffuse Reflection :-

Diffuse reflections are constants over each surface, independent of the viewing direction. The amount of incident light that is diffusely ~~reflect~~ reflected with parameter K_d, called diffuse reflection coefficient or diffuse reflectivity.

For an object, the illuminated light ~~from~~ with a directional light source from an infinity distance, the intensity of the diffused reflected light from the object is given by -

$$\boxed{I_{diff} = K_d I_l (N \cdot L)}$$



where $L \rightarrow$ incident light unit vector

$N \rightarrow$ Unit normal vector to the surface

$I_l \rightarrow$ Intensity of the falling light source

$\theta \rightarrow$ Angle between the surface normal and the incident light source.

$$L \cdot N = LN \cos\theta$$

If θ is greater than $90°$, light can be assumed to come from behind the object.

Thus, the net intensity due to the ambient light and a number of directional light source is given by -

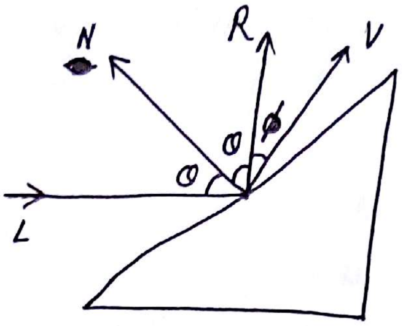
$$\boxed{I_{diff} = K_a I_a + \sum K_d I_l (N \cdot L)}$$

Both K_a and K_d, varies from 0 to 1, depend on the surface material properties.

2) Specular Reflection :-

When a polished object is kept under illumination, a bright spot can be found at a certain viewing direction. This phenomenon is known as specular reflection, which is the result of total reflection of incident light in a concentrated region around the specular reflection angle.

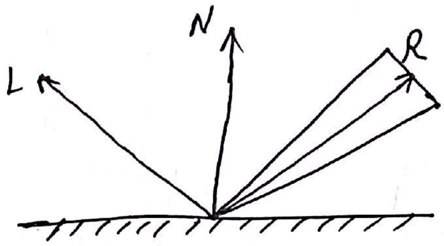
where V: viewing direction
R : Reflected light direction



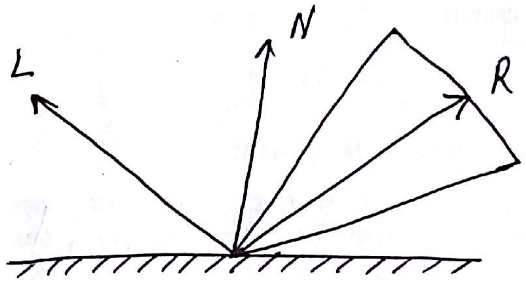
The specular-reflection angle equals to the angle of the incident light with the unit normal surface vector N. Angle φ is the viewing angle relative to the specular-reflection (R).

[Specular Reflection Model]

For an ideal reflector, incident light is reflected only in the specular reflection direction. In this case, we can see reflected light when φ=0. objects other than ideal reflectors exhibit specular reflections over a finite range of viewing positions around vector R. Shiny surfaces have narrow specular reflection range and dull surfaces have a wider reflection range.



Shiny surface
[Narrow specular Reflection]



Dull surface
[wider Reflection Range]

The intensity of specular reflection depends on the material properties of the surface and the angle of incidence.

The intensity of specular reflection is given by -

$$\boxed{I_{spec} = W(\theta)\, I_l \cos^n \phi}$$

where $W(\theta)$ is specular reflection coefficient, θ varies from 0° to 90° and at θ = 90°, $W(\theta) = 1$
n → depends on surface material
I_l → Intensity of light source

For shiny surface, n have large value and smaller values for dull surfaces. For a perfect reflector, n is infinity.

In general, intensity of specular reflection is denoted as-

$$\boxed{I_{spec} = k_s I_l (R \cdot V)^n}$$

Combined diffuse & specular reflection with multiple light source:-

For a single point light source -

$$I = I_{diff} + I_{spec}$$

$$I = K_a I_a + K_d I_l (N \cdot L) + K_s I_l (R \cdot V)^n$$

For more than one point light source -

$$I = K_a I_a + \sum_{i=1}^{n} I_{li} \left[K_d (N \cdot L_i) + K_s (R_i \cdot V)^n \right]$$

* Polygon Rendering Methods :-

In this, we consider the application of an illumination model to the rendering of standard graphics object: those formed with polygon surfaces. Each polygon can be rendered with a single intensity or the intensity can be obtained at each point of the surface using a interpolation scheme.

1) Phong shading

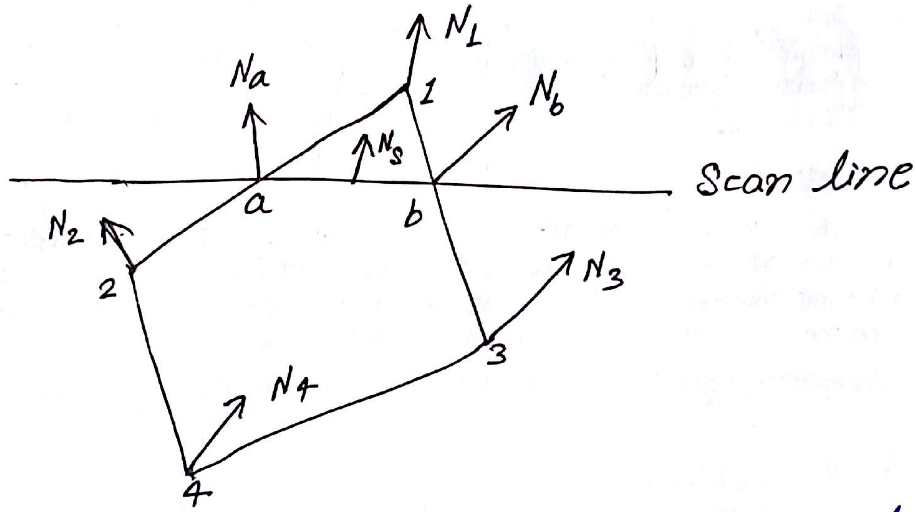
2) Gouraud shading

1) Phong Shading :-

Phong Shading is a method for rendering a polygon surface by interpolate normal vectors, and then apply the illumination model to each surface point. This method, developed by Phong Bui Tuong, is called Phong Shading or Normal-Vector interpolation Shading.

Ambient + Diffuse + Specular = Phong Reflection

In this method, determine the unit normal vector at each polygon vertex and then interpolate the vertex normals over the surface of the polygon. Apply an illumination model along each scan line to calculate intensities for the surface points.



For each scan line in the polygon, the normal vector N_a and N_b for the scan-line intersection point along the edge between vertices 1,2 and 1,3 respectively can be obtained by vertically interpolating between edge endpoints normals :

$$N_a = \frac{y_a - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y_a}{y_1 - y_2} N_2$$

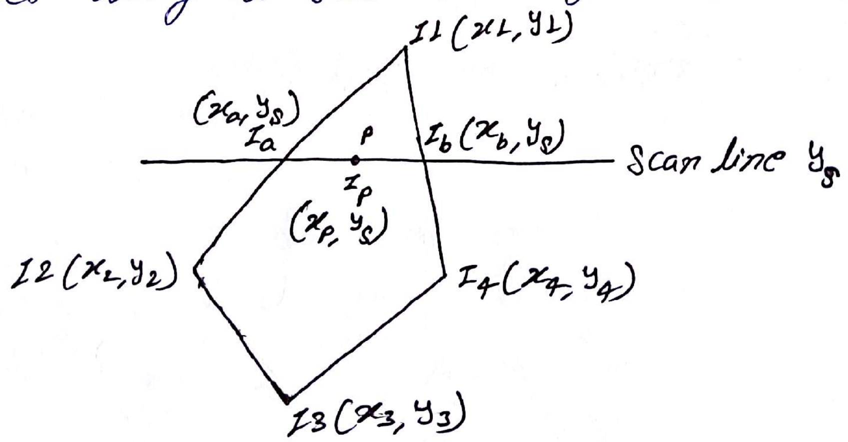
These two vectors N_a and N_b are then used to interpolate N_s. N_s, the interpolated normal vector, is then used ~~to~~ in the intensity calculation.

2) Gouraud Shading :-

Gouraud Shading is a method for linearly interpolating a colour or shade across a polygon. This method adding a curved feel to a polygon otherwise appear flat.

Imagine ~~you~~ we have a large polygon with a light near it's center. The light intensity at each vertex will be quite low because they are far from light. ~~The~~ The polygon will be rendered quite dark. This can be accomplished by applying the illumination model for each vertex of the polygon. Then linear interpolation is used to determine the color at each point inside the polygon. This can be done for each scan line.

In Gouraud shading, the intensity at each vertex of the polygon is first calculated. Then, the intensity at the edge of each scan line are calculated from the vertex intensities and the intensities along a scan line from these.



Intensity of ~~vertex~~ point a & b calculated as -

$$I_a = \left(\frac{y_s - y_2}{y_1 - y_2}\right) I_1 + \left(\frac{y_1 - y_s}{y_1 - y_2}\right) I_2$$

$$I_b = \left(\frac{y_s - y_4}{y_1 - y_4}\right) I_1 + \left(\frac{y_1 - y_s}{y_1 - y_4}\right) I_4$$

Now intensity of a & b is used to calculate intensity of point p -

$$I_p = \left(\frac{x_p - x_a}{x_b - x_a}\right) I_b + \left(\frac{x_b - x_p}{x_b - x_a}\right) I_a$$

The advantage of Gouraud Shading is that it is computationally less expensive, only requiring the evaluation of the intensity equation at the polygon vertices and then interpolation of these values. But Phong Shading require more calculation. It involve the interpolation of the surface normal and then evaluating intensity function for each pixel.

* Ray Tracing :-

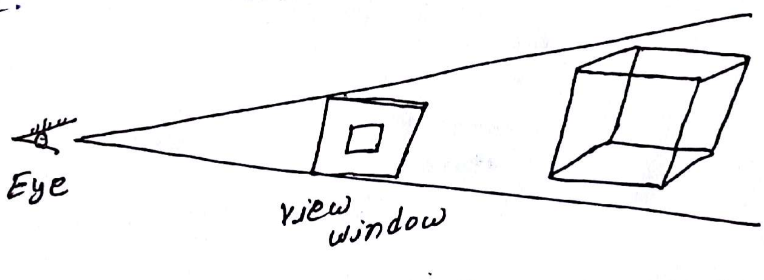
Ray tracing is a global illumination based rendering method. It traces the rays of light from the eye. The rays are tested against all objects in the scene to determine they intersect any objects or not. If ray misses all objects, then the pixel is shaded the background color.

In ray tracing, we start from the eye or camera and trace the ray through a pixel in the image plane into the scene and determine what it hits. The pixel is then set of the color values returned by the ray.

Ray Tracing Terminology :-

In computer graphics, we have a list of objects that are part of scene or world. The ray tracer drawing the objects from a given view point, this view point is called the eye or camera. In graphics, we have a view window, on which we drawn the scene.

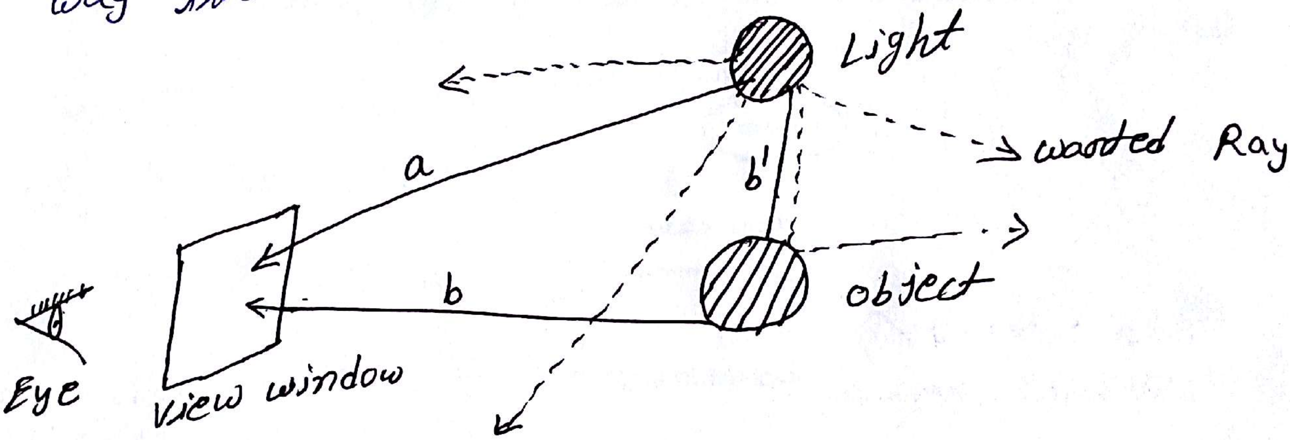
The view window is subdivided into a small window squares, where each corresponds to one pixel in the final image. If we want to create an image at resolution of 640 X 400, then break up the view window into a grid of 640 squares across and 400 squares down. The real problem then is assigning a color to each square.



Working :-

Ray tracing is to determine the color of each light ray that strikes the view window before reaching the eye. we start from the light, then for each ray we have to decide its direction, there is an infinite direction in which it can travel. Some will reach eye directly, other will bounce around some and then reach the eye, and many more will probably never hit the eye at all. In order to save wasted effort, we trace only those rays that are guaranted to hit the view window and reach the eye.

A very simple solution of this problem is, instead of tracing the rays starting at the light source, we trace them backward starting at the eye. we can just follow the ray backward by starting at the eye and passing through the point on its way into the same scene.



To determine the color of a, we follow the ray a directly towards the light source; and b will be shadowed because the ray b' towards the light source is blocked by the sphere itself. Ray a would ~~also~~ have also been shadowed if another object blocked the ray a.

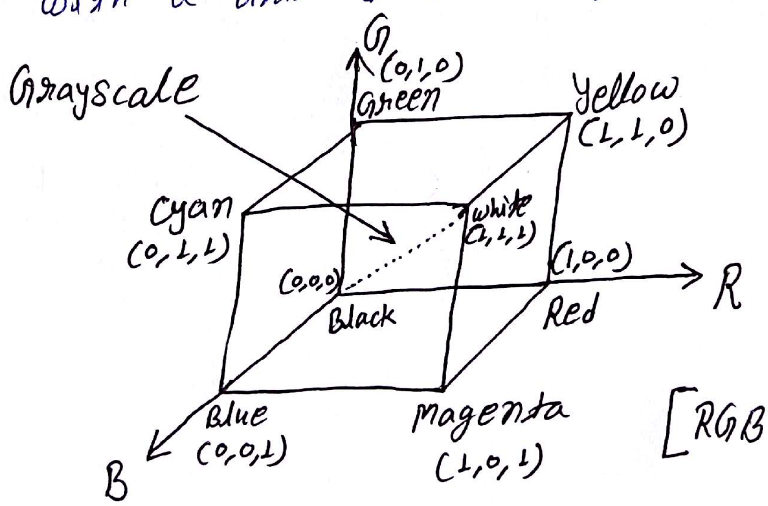
Color Models :-

A color model is a method for explaining the properties or behaviour of color. No single color model can explain all aspects of color. Basically, a color model is a specification of a 3D color coordinate system within which all colors lies in a particular color range.

Generally, we quickly give responce to blue, green and red colors because our eyes have three different color sensor, they are called blue, green and red photopigments. our eyes are not equally sensitive to all colors. It gives the greatest responce to green.

1) RGB Color Model :-

RGB Color model is also called as additive model. The three primary colors red, green and blue are used in color CRT monitor. In this model, the individual contribution of red, green and blue are added together to get the resultant color. we can represent this color model with a unit cube defined on R, G and B axis.



[RGB Color Cube]

The origin ~~represent~~ represents black (0,0,0) and the vertex with coordinate (1,1,1) represents as white. Vertices of the cube on the axes represents the primary colors, and the remaining vertices represents the complementary color for each of the primary colors.

Each color points on the cube can be represented as the triple (R, G, B), where values for R, G and B are assigned in the range from 0 to 1. The white vertex (1,1,1) is the sum of the red, green and blue vertices.

Shades of gray are represented along the main diagonal of the cube from the origin (black) to the white vertex. Each point along this diagonal has an equal contribution from each primary color, is represented as (0.5, 0.5, 0.5).

✳ YIQ Color Model :-

RGB monitors requires separate signals for the red, green and blue components of an image, a television monitor uses a single composite signal.

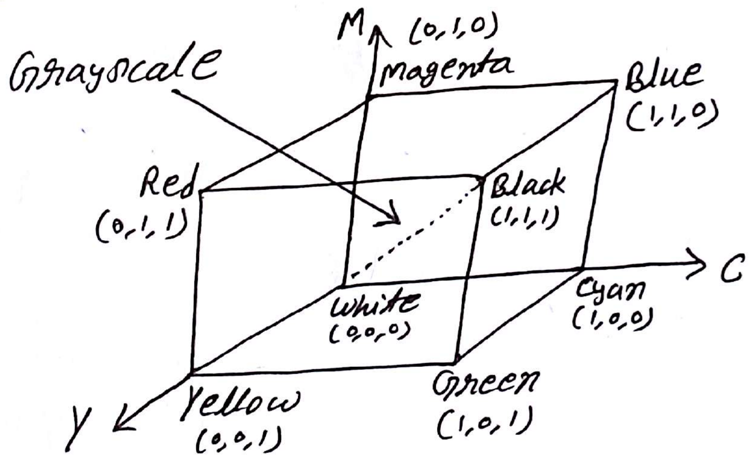
In the YIQ model, parameter Y is same as in the ~~previous~~ previous model. ~~Luminance~~ Luminance (Brightness) information is contained in the Y parameter, while chromaticity information (hue and purity) is incorporated into the I and Q parameters.

A combination of red, green and blue intensities are chosen for the Y parameter to yield the standard luminosity curve. Since Y contains the brightness information, black and white television monitors use only the Y signal. Parameter I contains orange-cyan hue information and parameter Q carries green-magenta hue information.

The advantage of YIQ model is that we can deal with the luminance (brightness) component independently.

* CMY Color Model :-

A CMY color model defined with the primary colors cyan, magenta and yellow (CMY) is useful for describing color output to hard copy devices. This color model also defined with a unit cube and C, M, Y as a axis.



In this color model, cyan is located at X-axis and it is formed by adding green and blue light. The origin of the cube represent white (0,0,0) and point (1,1,1) represents black. Equal amount of each of the primary colors produces grays along the main diagonal of the cube.

In the printing process, an RGB color monitor uses a collection of three phosphor dots but the CMY color model, generates a color point with a collection of four ink dots. one dot is used for each of the primary colors (cyan, magenta and yellow) and one dot is black.

Conversion from RGB to CMY –

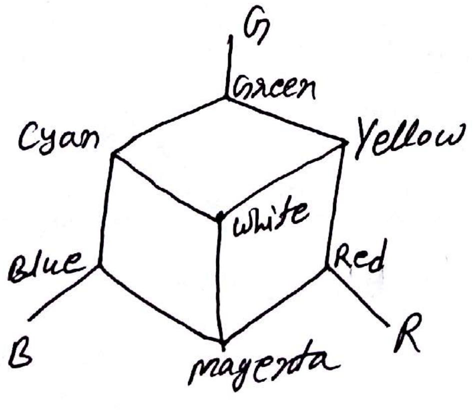
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Conversion from CMY to RGB –

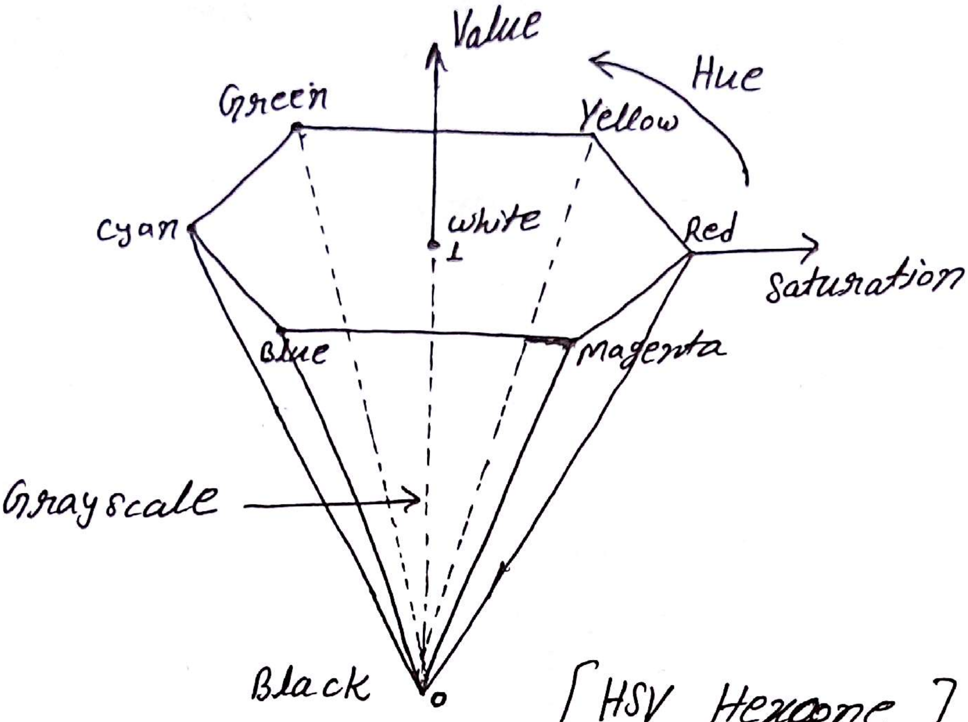
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

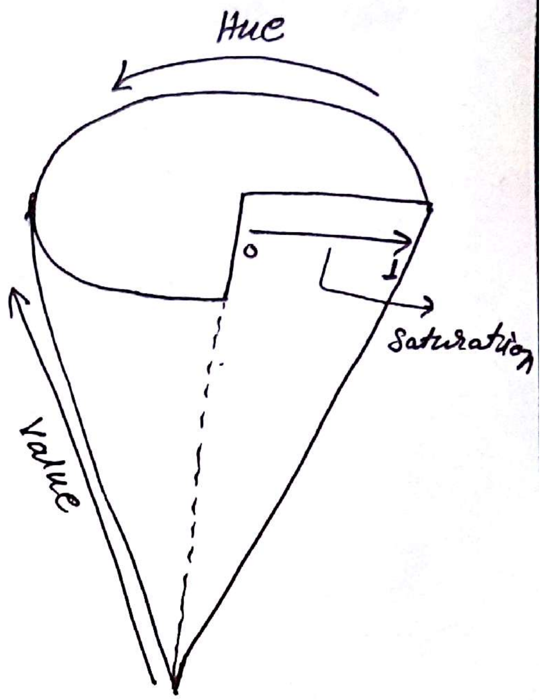
* HSV Color Model :-

In this model, color parameters are Hue (H), Saturation (S) and value (V). The three dimensional representation of the HSV color model derived from the RGB cube. If we observe the RGB cube along the diagonal from white vertex to origin (black), we found an hexagon shape.



[RGB Color Cube]



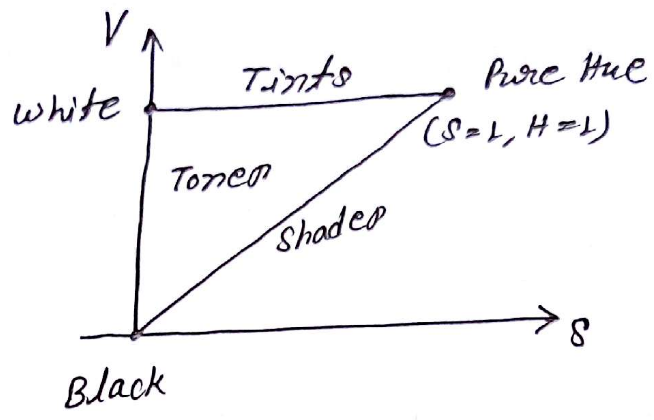


[HSV Hexcone]

The boundary of the hexagon represents the various hues. As hue varies from 0° to 360°, the corresponding colors vary from red through yellow, green, cyan, blue, magenta and back to red; so that there are red values at both 0° and 360°. vertices of the hexagon are separated by 60° intervals.

As saturation varies from 0 to 1, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component)

As value or brightness varies from 0 to 1, the corresponding colors become increasingly brighter. In the hexagon, saturation is measured along a horizontal axis and value is along a vertical axis.



The pure hue are given for V=1 and S=1. Tints are formed by adding white that is decreasing S. Shades are formed by decreasing V, that is adding black and tones by decreasing both V and S.

Hue → varies colors

Saturation → shades

Value → Brightness (↑↓ Black & White)

——— * ———