

UNIT - 3

3-D Transformation -

① Translation -

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow P' = T \cdot P$$

$x' = x + t_x$
 $y' = y + t_y$
 $z' = z + t_z$

② Rotation -About z -axis

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

About x -axis

$$x' = x$$

$$y' = y \cos \theta - z \sin \theta$$

$$z' = y \sin \theta + z \cos \theta$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

About y -axis

$$x' = z \cos \theta - y \sin \theta$$

$$y' = y$$

$$z' = z \sin \theta + y \cos \theta$$

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

③ Scaling -

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow P' = S \cdot P$$

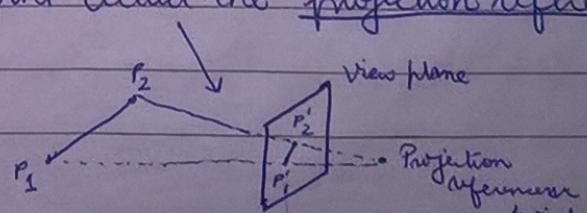
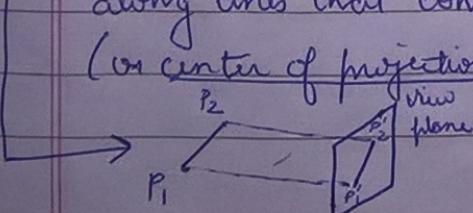
$x' = x \cdot S_x$
 $y' = y \cdot S_y$
 $z' = z \cdot S_z$

Projection -

① To Project the 3D objects onto the 2D view plane. There are two basic projection methods - Parallel and Perspective.

Parallel Projection - Coordinate positions are transformed to the view plane along parallel lines.

Perspective Projection - Object positions are transformed to the view plane along lines that converge to a point called the projection reference point (or center of projection)



The projection view of an object is determined by calculating the intersection of the projection lines with the new plane.

② Types of Parallel Projections-

Orthographic Parallel Projection- When the projection is perpendicular to the new plane. They are most often used to produce the front, side and rare (called elevations) and top views (called plan views).

Projection that display more than one face of an object such views are called anomalous orthographic projection. Most commonly used anomalous projection is the isometric projection.

$$x_p = x \quad \text{and} \quad y_p = y \quad \text{and} \quad z\text{-coordinate remains same (XY-plane view)}$$

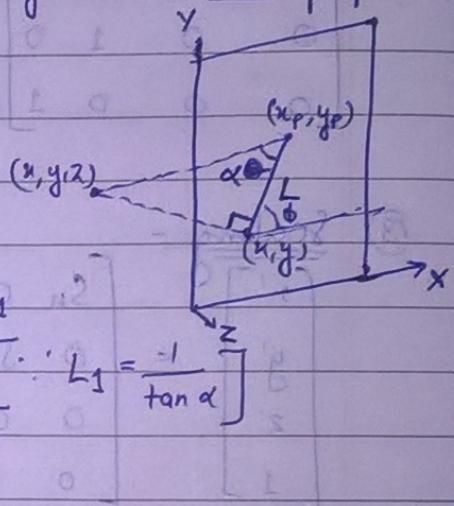
Oblique parallel projection - When the projection is not perpendicular to the new plane.

$$x_t = x + h \cos \phi \quad y_t = y + h \sin \phi$$

length h depends on α and z -coordinate

$$\tan \alpha = \frac{z}{h} \Rightarrow h = \frac{z}{\tan \alpha} = z L_1 \quad \left[\because L_1 = \frac{1}{\tan \alpha} \right]$$

$$\Rightarrow x_t = x + z(L_1 \cos \phi) \quad \& \quad y_t = y + z(L_1 \sin \phi)$$



Transformation matrix for any parallel projection,

$$M_{\text{parallel}} = \begin{bmatrix} 1 & 0 & L_1 \cos \phi & 0 \\ 0 & 1 & L_1 \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $L_1 = \frac{1}{\tan \alpha}$

When $\alpha = 45^\circ$, the views obtained are the cavalier projection

When $\tan \alpha = 2$ or $\alpha = 63.4^\circ$, the views obtained are cabinet projection

③ Types of Perspective Projection -

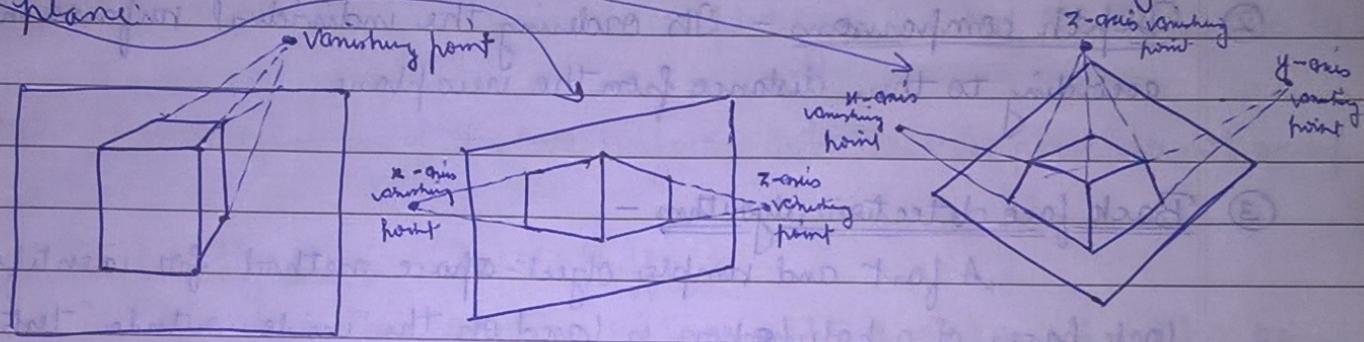
The point at which a set of projected parallel lines appears to converge is called a vanishing point.

The vanishing point for any set of lines that are parallel to one of the principal axes of an object is referred to as a principal vanishing point.

One-principle vanishing point - One principal axis intersecting the view plane.

Two-principle vanishing point - Two principal axes intersecting the view plane.

Three-principle vanishing point - Three principal axes intersecting the view plane.



Transformation matrix for any perspective projection,

$$\begin{bmatrix} u_n \\ y_n \\ z_n \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/dp & z_{vp}(z_{prp}/dp) \\ 0 & 0 & -1/dp & z_{prp}/dp \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{where } clp = z_{prp} - z_{vp}$$

$$h = \frac{z_{prp} - z}{dp}$$

Therefore, $u_p = \frac{u_n}{h}$ and $y_p = \frac{y_n}{h}$

Hidden Surface Elimination -

① In a given set of 3D objects and viewing specifications, we wish to determine which lines or surfaces of the objects are visible, so that we can display only the visible lines or surfaces. This process is known as hidden surface or hidden line elimination or visible-surface detection.

Visible surface detection are broadly classified into two categories.

Object-space methods - It is implemented in the physical coordinate system in which objects are described.

Image-space methods - It is implemented in the screen coordinate system in which objects are viewed (projected images).

② Depth comparisons - Its ordering the individual surfaces in a scene according to their distance from the view plane.

③ Back face detection algorithm -

A fast and simple object-space method for identifying the back faces of a polygon is based on the "inside-outside" test.

A point (x, y, z) is inside a polygon surface if,

$$Ax + By + Cz + D \leq 0 \quad \text{and } A, B, C, D \rightarrow \text{plane parameters}$$

If V is a vector in viewing direction and N is the normal vector of a surface, then this polygon is a back face if,

$$V \cdot N > 0$$

If V is parallel to z axis, then $V = (0, 0, V_z)$ and

$$V \cdot N = V_z \cdot C \quad (C \rightarrow z \text{ component of the normal vector})$$

Polygon is a back face if

$$C \leq 0$$

④ Painter's Algorithm (Depth Sort Algorithm) -

The basic idea is to paint the polygons into the frame buffer in order of decreasing distance from the viewpoint. The process involves following basic functions -

(1) Sorting of polygons in order of decreasing depth.

(2) ~~Rendering~~ Scan conversion of polygons in order, starting with the polygon of greatest depth.

If a depth overlap is detected at any point in the list, we need to make some additional comparisons to determine whether any of the surfaces should be reordered.

We make the following tests for each surface that overlaps with S. If any one of these test is true, no reordering is necessary for that surface. These tests are listed in order of increasing difficulty -

- (1) The bounding rectangles in the XY plane for the two surfaces do not overlap.
 - (2) Surface S is completely behind the overlapping surface relative to the viewing position.
 - (3) The overlapping surface is completely in front of the S relative to the viewing position.
 - (4) The projection of the two surfaces onto the view plane do not overlap.
- ⑤ Z-Buffer Algorithm - (Depth-Buffer)

This algorithm compares surface depths at each pixel position on the projection plane. The surface depth is measured from the view plane along the z axis of a viewing system.

Algorithm -

- (1) Initialize the depth buffer and refresh buffer so that for all buffer positions (n, y) ,

$$\text{depth}(n, y) = 0, \quad \text{refresh}(n, y) = I_{\text{background}}$$

- (2) For each position on each polygon surface, compare depth values to previously stored values in the depth buffer to determine visibility.

- Calculate the depth z for each (n, y) position on the polygon
- If $z > \text{depth}(n, y)$, then set

$$\boxed{\text{depth}(n, y) = z, \quad \text{refresh}(n, y) = I_{\text{surface}}(n, y)}$$

Depth buffer contains real depth values for the visible surfaces and the refresh buffer contains the corresponding intensity values for those surfaces.

$$\text{Initial value, } z = -\frac{Ax - By - D}{C}, \quad \text{Next position } z' = -\frac{A(n+1) - By - D}{C} \Rightarrow \boxed{z' = z - \frac{A}{C}}$$

$$\text{Left edge of the polygon, } n' = \frac{n-1}{m}; \quad z' = z + \frac{A/m + B}{C} \Rightarrow \boxed{z' = z + \frac{B}{C}} \quad (\text{for vertical edge})$$

Curve Generation -

1. Curve Generation -

(1) Control Points - Points that control the shape of the curve.

Multiparameter points - Curve is a multiple valued function.

Axes independent - Shape of the object do not change when changing the coordinate system.

Global and local control points - Local control points are those that a curve changes its shape in a particular region whereas global control points change the entire shape of the curve.

Variation diminishing property → Curve tends to be smooth.

Versatility - Variety of shape of the curve.

Order of continuity -

zero order continuity - When two curves meet.

first order continuity - Curves to be tangent at a point of intersection.

second order continuity - Curvature should be same.

2. Bézier Curves - It is determined by a defining polygon.

Highly useful and convenient for curve and surface design.

Easy to implement. Specially discuss Cubic Bézier curve.

→ Properties of Bézier curves -

(1) Basic functions are real.

(2) Same end points as the guiding polygon.

(3) Degree of polynomial = Total control points - 1.

- (4) Generally follow the shape of the defining polygon.

→ (5) Direction of the tangent vector at the end points is the same as that of the vector determined by first and last segments.

- (6) lies entirely within the convex hull formed by control points.

(7) Convex hull property ensures that polynomial smoothly follow the control points.

(8) Variation diminishing property.

(9) Invariant under an affine transformation.

* Cubic Bézier curve has four control points.

A
3

Bézier matrix for periodic cubic polynomial is,

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \therefore P(u) = U \cdot M_B \cdot G_B \text{ where } G_B = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$\Rightarrow P(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u) P_3 + u^3 P_4$$

→ Bernstein

③ B-Spline curves -

It is nonlocal because each vertex B_i is associated with a unique basis function.

$$P(u) = \sum_{i=1}^{n+1} B_i N_{i,k}(u) ; \quad u_{\min} \leq u \leq u_{\max}, \quad 2 \leq k \leq n+1$$

$B_i \rightarrow$ position vectors of $n+1$ defining polygon vertices.

$N_{i,k}(u) \rightarrow$ i^{th} Normalized B-spline basis functions of order K

$N_{i,k}(u)$ is defined as -

$$N_{i,k}(u) = \begin{cases} 1, & \text{if } u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - u_i) \cdot N_{i,k-1}(u)}{u_{i+k-1} - u_i} + \frac{(u_{i+k} - u) \cdot N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}}$$

Three types of 'knot vector' uniform, open uniform and non uniform.

Uniform - $N_{i,k}(u) = N_{i-1,k}(u-1) = N_{i+1,k}(u+1)$ e.g.: [0 1 2 3 4]

Open uniform - nearly like Bézier curve.

$$u_i = 0 \quad \text{for } 1 \leq i \leq k \quad \text{e.g. } K=2 \quad [0 \ 0 \ 1 \ 2 \ 3 \ 3]$$

$$u_i = i-k \quad \text{for } k+1 \leq i \leq n+1$$

$$u_i = n-k+2 \quad \text{for } n+2 \leq i \leq n+k+1$$

Properties of B-spline curve -

(1) The sum of the B-spline basis functions for any parametric value u is 1

$$\text{i.e. } \sum_{i=1}^{n+1} N_{i,k}(u) = 1$$

- (2) Each basis function is +ve or 0 for all parameter values, i.e. $N_{i,k} \geq 0$
- (3) Except for $K=1$ each basis function has precisely one maximum value.
- (4) Maximum order of the curve = No. of vertices of defining polygon
- (5) Degree of B-Spline polynomial is independent on the no. of vertices
- (6) Allow local control
- (7) Variation diminishing property
- (8) Generally follow the shape of defining polygon
- (9) lies entirely within the convex hull
- (10) Any affine transformation can be applied to the curve

Basic Illumination Model - Calculate the intensity of light that we should

① see at a given point on the surface of an object

② Diffuse Reflection -

When we assume that going up, down, right and left is of same amount then we can say that the reflections are constant over each surface of the object and they are independent of the viewing direction. Such a reflection called as diffuse reflection.

③

④ Specular Reflection -

The phenomenon of reflection of incident light in a concentrated region around the specular reflection angle is called specular reflection. Due to specular reflection at the highlight, the surface appears to be not in its original colour, but white, the colour of the incident light.

The angle ϕ between vector R (ideal specular reflection) and vector V (viewer) is called viewing angle.

⑤ Gouraud Shading -

The polygon surface is displayed by linearly interpolating intensity values across the surface. Here, intensity values for each polygon are matched with the values of adjacent polygons along the common edges.

By performing following calculations we can display polygon surface with Gouraud shading:-

(1) Determine the average unit normal vector at each polygon vertex.

$$N_V = \frac{\sum_{i=1}^n N_i}{\left| \sum_{i=1}^n N_i \right|}$$

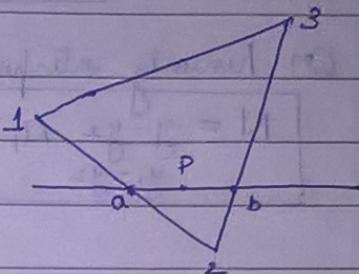
where n is the no. of surface normals of polygons sharing that vertex.

(2) Apply an illumination model to each polygon vertex to determine the vertex intensity.

$$I_a = \frac{y_a - y_2}{y_3 - y_2} I_1 + \frac{y_1 - y_a}{y_1 - y_2} I_2$$

$$I_b = \frac{y_b - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_b}{y_3 - y_2} I_2$$

$$I_p = \frac{x_b - x_p}{x_b - x_a} I_a + \frac{x_p - x_a}{x_b - x_a} I_b$$



(3) linearly interpolate the vertex intensities over the surface of the polygon

If intensity at edge position (x, y) , $I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2$

then intensity along this edge for the next line scan, $y+1$ as

$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Similarly,

$$I' = I + \frac{I_b - I_a}{x_b - x_a}$$

Advantages -

- (1) Removes intensity discontinuities.
- (2) Combined with hidden surface algorithm to fill the visible polygons

Disadvantages -

- (1) Highlights on the surface are sometimes displayed with anomalous shape.
- (2) Linear intensity interpolation causes Mach bands (bright or dark intensity)
- (2) Sharp drop of intensity values on the polygon surface cannot be displayed

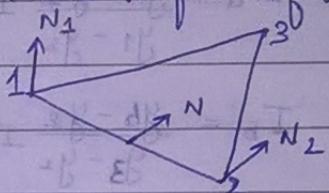
④ Phong Shading - (normal vector interpolation shading) -
 Interpolates the surface normal vector \mathbf{N} , instead of the intensity.
 By performing following steps we can display polygon surface using
 Phong Shading -

(1) Determine the average unit normal vector at each polygon vertex.

$$N_v = \frac{\sum_{i=1}^n N_i}{\left| \sum_{i=1}^n N_i \right|}$$

(2) Linearly interpolate the vertex normals over the surface of the polygon.

$$\mathbf{N} = \frac{y_2 - y_1}{y_1 - y_2} \mathbf{N}_1 + \frac{y_1 - y_2}{y_1 - y_2} \mathbf{N}_2$$



(3) Apply an illumination model along each scan line to determine projected pixel intensities for the surface points.

Advantages -

(1) Display more realistic highlights on the surface.

(2) Reduces the Mach-band effect.

(3) give more accurate results

Disadvantage - Requires more calculation and greatly increases the cost of shading deeply.

⑤ Ray Tracing -

We can bounce the ray around the picture from one surface to another surface, it contributes the intensity for that surface.

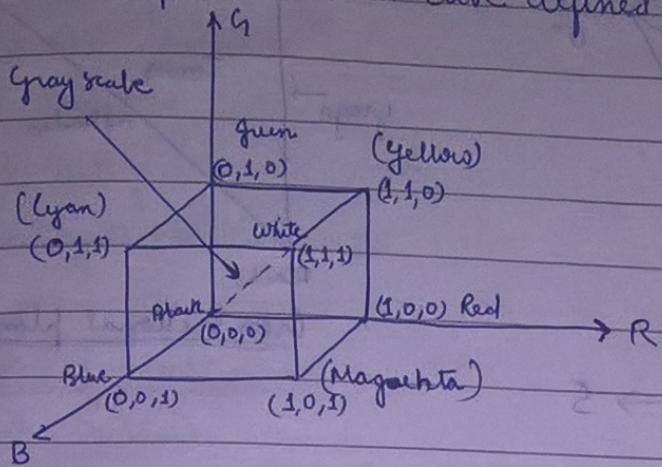
Ray tracing tree is the combination of reflection and refraction.

⑥ Colour Models -

It is a specification of a 3D colour coordinate system and a visible subset in the coordinate system within which all colours in a particular colour range lie.

RGB colour Model (Red, Green, Blue) - Used in colour CRT

Represent as unit cube defined on R, G and B axes.



Resultant Colour, $C_g = RR + Gg + BB$

YIQ Colour model -

Used for the broadcast TV colour system.

$Y \rightarrow$ luminance (Brightness) information

$I \& Q \rightarrow$ chromaticity (hue and purity) information

$I \rightarrow$ orange - cyan hue information

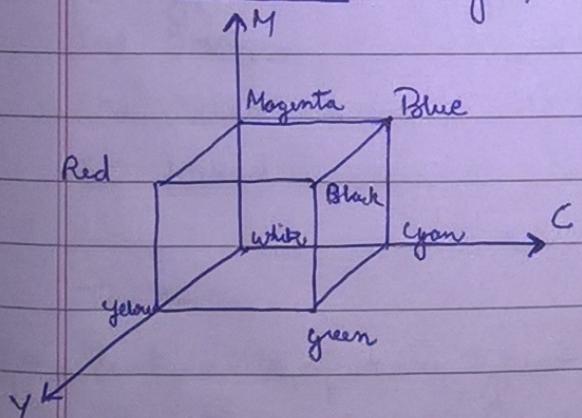
$Q \rightarrow$ green - magenta hue information

Made by NTSC (National Television System Committee)

NTSC encoder \rightarrow RGB signals converted into YIQ values

NTSC decoder \rightarrow YIQ values converted into RGB signals

CMY Colour Model - (Cyan, Magenta, Yellow) Used in colour output (printers).



CMY to RGB -

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

RGB to CMY -

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

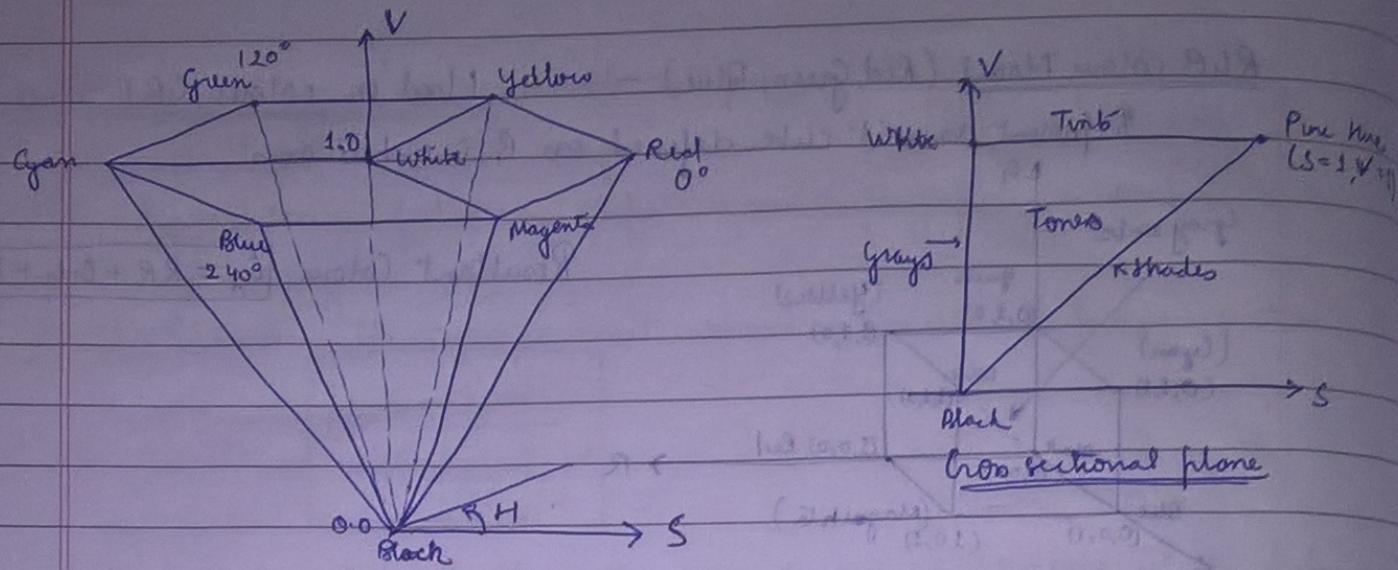
HSV Colour Model - user oriented model.

Three colour parameter -

Hue (H) \rightarrow distinguishes among colour such as red, green, purple and yellow

Saturation (S) \rightarrow refers to how far colour is from a gray of equal intensity.

Value (V) \rightarrow level of brightness.



- white material \Rightarrow V
 - yellow free (yellow light) \rightarrow around $\sim V$
 - yellow free (yellow light) \rightarrow $\sim R + T$
 - yellow free (yellow light) \rightarrow $\sim I$
 - yellow free (yellow light) \rightarrow $\sim B$
 (yellowish) yellow material (yellow light) \rightarrow $\sim R + I + B$
 yellow \Rightarrow V at 100% \rightarrow $\sim V$ \rightarrow white \Rightarrow $\sim V$
 bright \Rightarrow V at 100% \rightarrow $\sim V$ \rightarrow white \Rightarrow $\sim V$

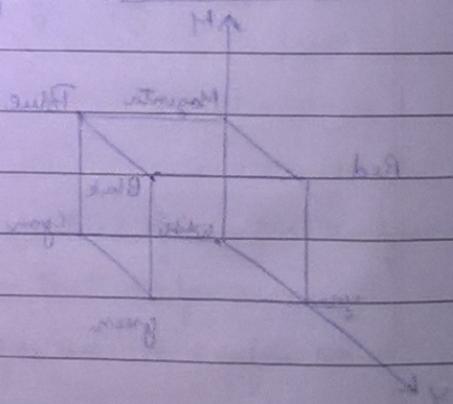
(yellow light) yellow material (yellow light) \rightarrow yellow material \Rightarrow V

- V at 100%

- B at 100%

H

$$\begin{array}{c}
 M = R + B = Y \\
 M = L + a + b \\
 M = P - Z = M
 \end{array}$$



Yellow material \rightarrow - yellow material \Rightarrow V

- yellow material \rightarrow V

Yellow has yellowing, but no blue material gives no yellowing \rightarrow (V) with

blue light for yellow \rightarrow yellow material \rightarrow (V) without blue

material for blue \rightarrow (V) without