S J P N Trust's

Hirasugar Institute of Technology, Nidasoshi.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka, Affiliated to VTU Belagavi.

Accredited at "A" Grade by NAAC and Recognized Under Section 2(f) of UGC Act, 1956.

Module-3

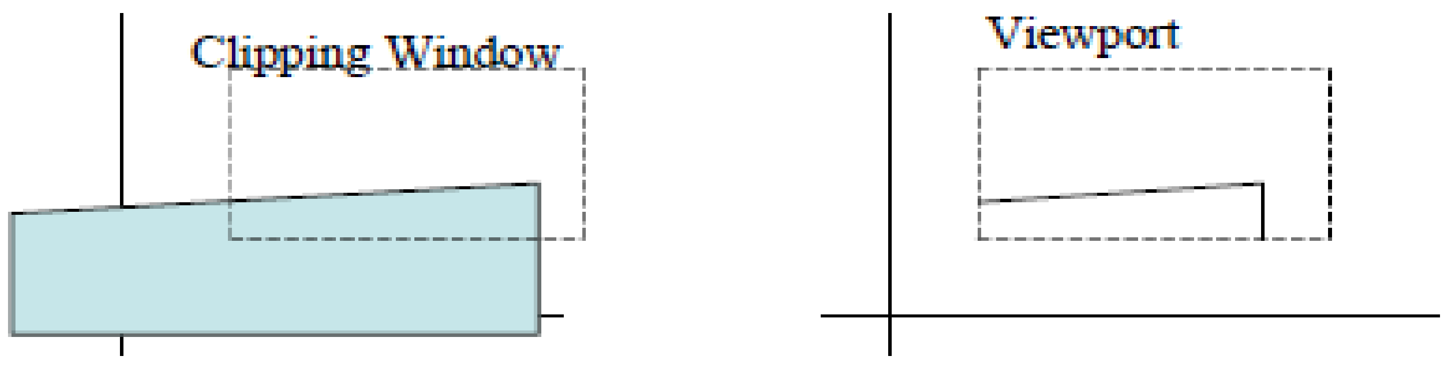
18CS62

Clipping,3D Geometric Transformations, Color and Illumination Models:

Prof. Rahul Palakar

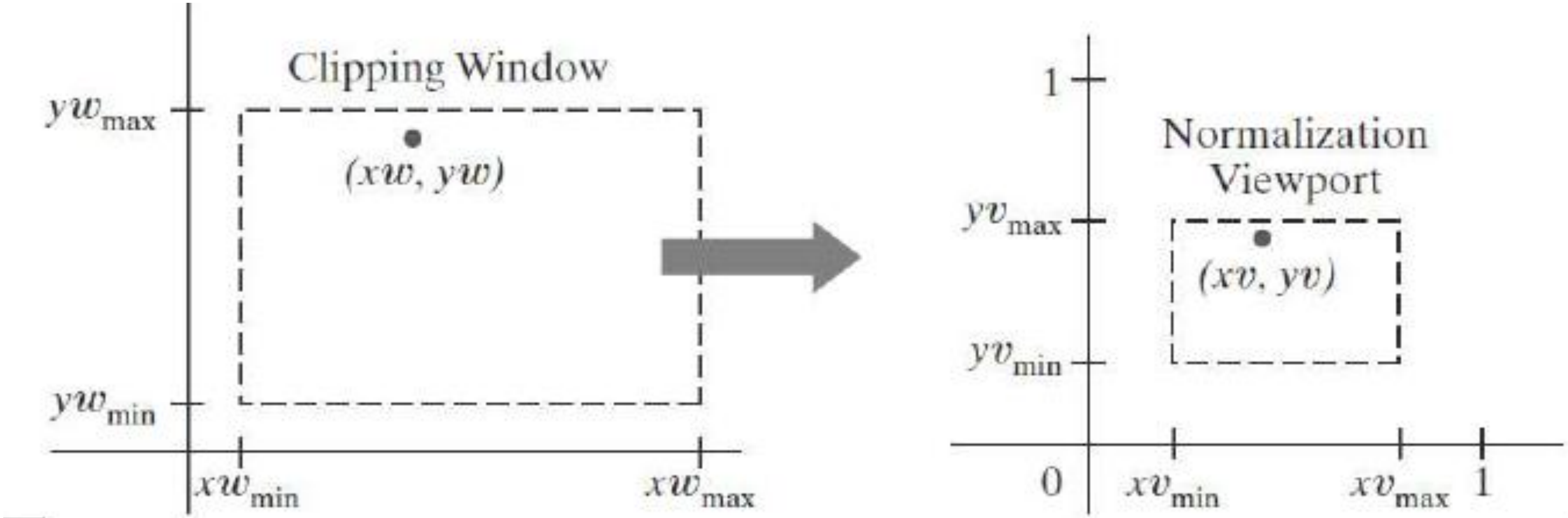
The Two Dimensional Viewing Pipeline

- Clipping window -- the part of two dimensional scene that it to be displayed
- Viewport -- window where data from clipping window will be displayed
- Mapping between these two called 2D viewing transformation



Window to Viewport

- Object descriptions are transferred to this normalized space using a transformation that maintains the same relative placement of a point in the viewport as it had in the clipping window
- Position (xw, yw) in the clipping window is mapped to position (xv, yv) in the associated viewport



- To transform the world-coordinate point into the same relative position within the viewport, we require,

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

- Solving these equations for (xv, yv)

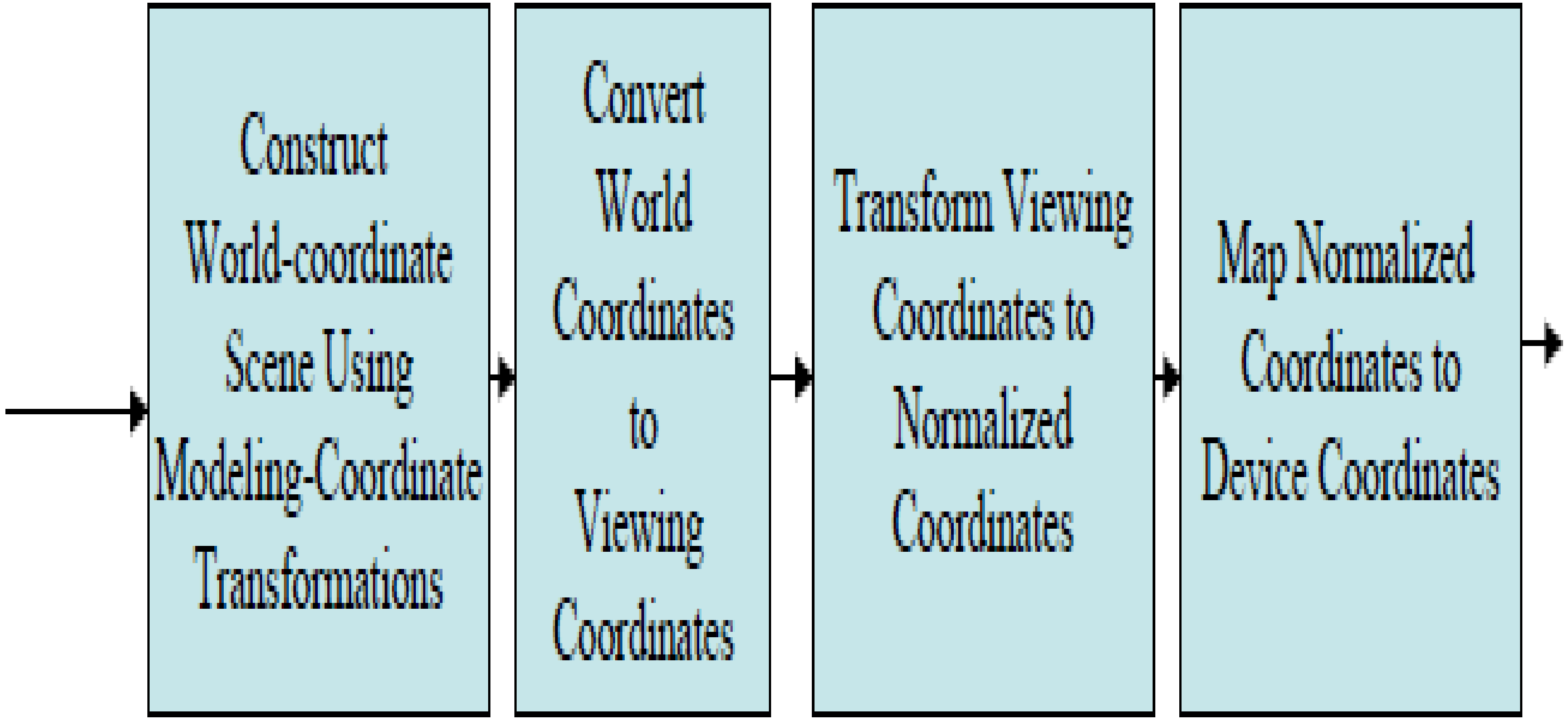
$$xv = s_x xw + t_x$$

$$yv = s_y yw + t_y$$

such that $s_x = \dfrac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$ $\quad t_x = \dfrac{xw_{max}xv_{min} - xw_{min}xv_{max}}{xw_{max} - xw_{min}}$

$$s_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}} \quad t_y = \frac{yw_{max}yv_{min} - yw_{min}yv_{max}}{yw_{max} - yw_{min}}$$

Two Dimensional Viewing Pipeline



The Clipping Window

- Most graphics packages support rectangular clipping regions
- Some systems support rotated 2D viewing frames, but usually clipping window must be specified in world coordinates.
- Can set up a **viewing coordinate system** within the world-coordinate frame.
- This viewing frame provides a reference for specifying a rectangular clipping window with any specified orientation and position

- Choose **P**0=(x0,y0) base position, and a vector **V** that defines the y view direction.
- V is called the **view up** vector.
- Alternative we could have used a rotational angle.

Getting into the Viewing Frame

- Translate the viewing origin to the world origin.
- Rotate the viewing system to align with the world frame.

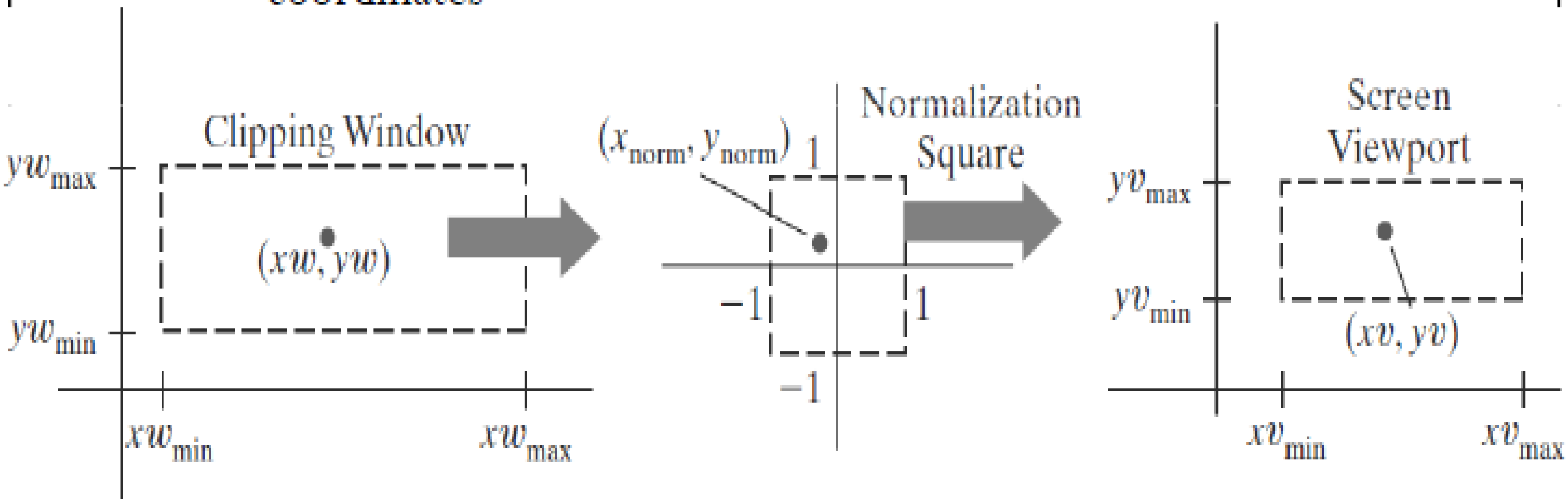
$$\mathbf{M}_{WC,VC} = \mathbf{RT}$$

- Suppose **P0**=(1,2) and **V**=(4,3). Calculate $\mathbf{M}_{WC,VC}$
- To specify clipping window now give its lower and upper corner positions

Clipping Window into Normalized Viewport

- Before mapping to the actual viewport we map into a normalized viewport of sides between 0 and 1 in each axis.
- We clip as we map into this normalized region.
- Then we do a straightforward transformation from the normalized viewport to the actual viewport.

- Clipping window into a normalized square
  - clipping routines applied on normalized coordinates
  - transfer the scene description to a viewport specified in screen coordinates



  - transfer the contents of the clipping window into the normalization square - similar to window-to-viewport transformation

- Window to normalized square transformation matrix
  - Obtained by *xvmin* , *yvmin* = –1 and *xvmax*, *yvmax*= +1

$$\mathbf{M}_{\text{window, normsquare}} = \begin{bmatrix} \frac{2}{xw_{\max} - xw_{\min}} & 0 & -\frac{xw_{\max} + xw_{\min}}{xw_{\max} - xw_{\min}} \\ 0 & \frac{2}{yw_{\max} - yw_{\min}} & -\frac{yw_{\max} + yw_{\min}}{yw_{\max} - yw_{\min}} \\ 0 & 0 & 1 \end{bmatrix}$$

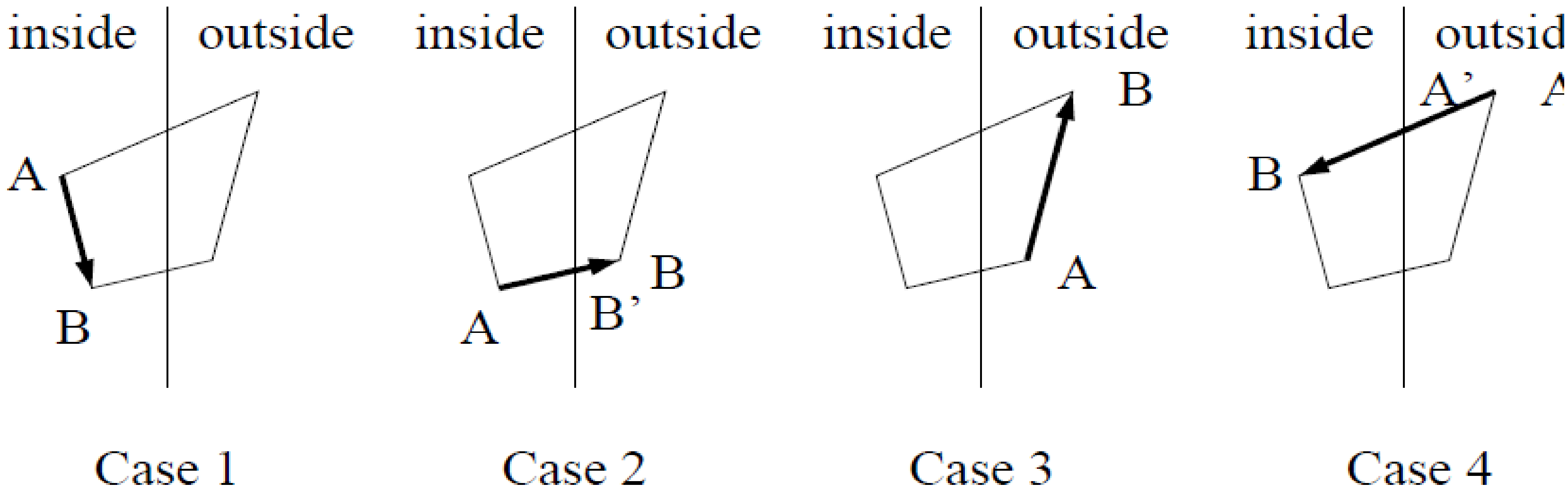
- Normalized square to viewport transformation matrix
  - Obtained by *xwmin* , *ywmin* = –1 and *xwmax*, *ywmax*= +1

$$\mathbf{M}_{\text{normsquare, viewport}} = \begin{bmatrix} \frac{xv_{\max} - xv_{\min}}{2} & 0 & \frac{xv_{\max} + xv_{\min}}{2} \\ 0 & \frac{yv_{\max} - yv_{\min}}{2} & \frac{yv_{\max} + yv_{\min}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Sutherland-Hodgman algorithm for Clipping polygons

- Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons.
- The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.
- The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Working



Assuming vertex A has already been processed,

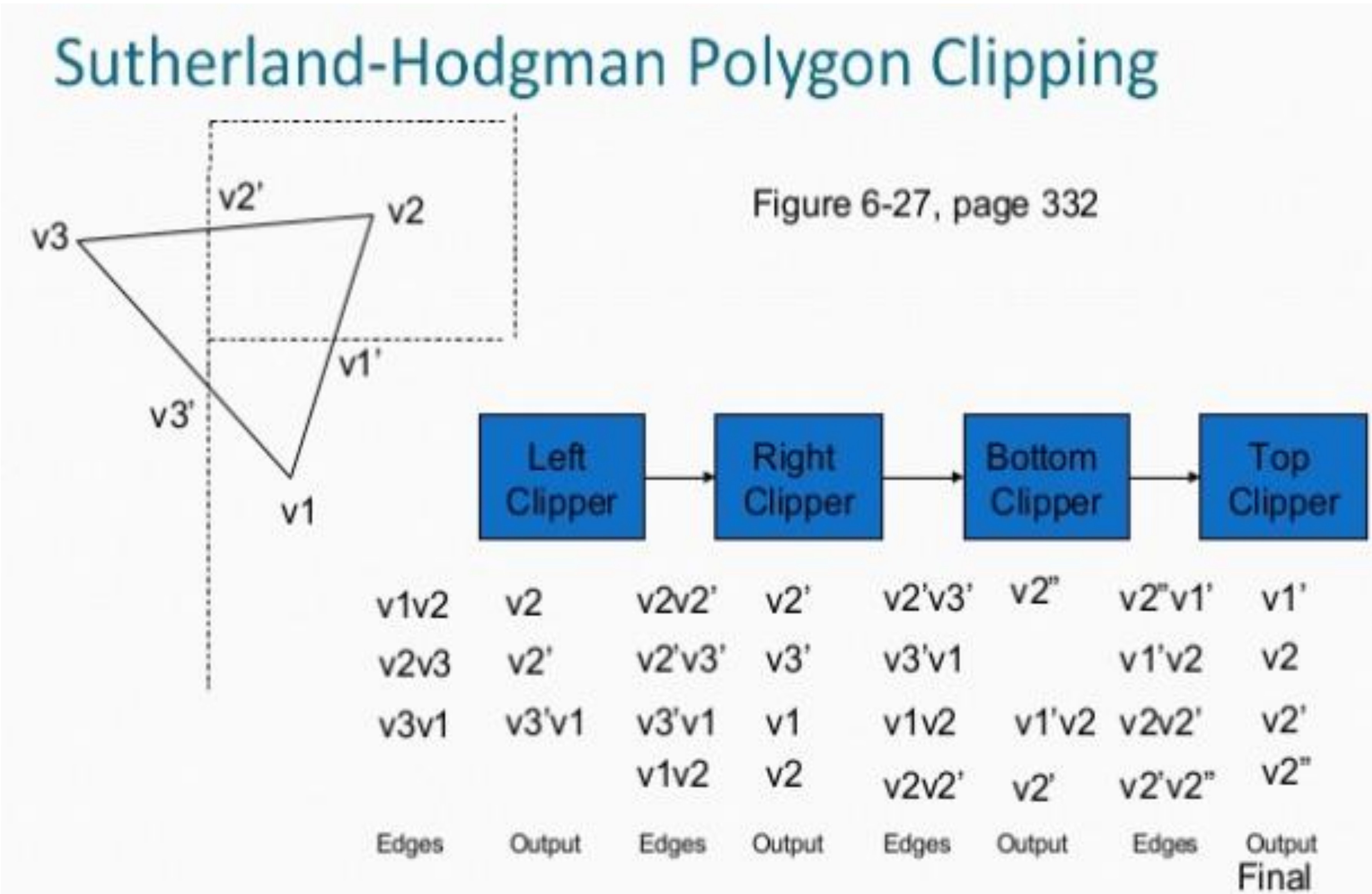
Case 1 — vertex B is added to the output list

Case 2 — vertex B' is added to the output (edge AB is clipped to AB')

Case 3 — no vertex added (segment AB clipped out)

Case 4 — vertices A' and B are added to the output (edge AB is clipped to A'B)

Example


Sutherland-Hodgman Polygon Clipping
v2'
v2
v3
v1'
v3'
v1
Figure 6-27, page 332
Left Clipper
Right Clipper
Bottom Clipper
Top Clipper
v1v2 v2 v2v2' v2' v2'v3' v2" v2"v1' v1'
v2v3 v2' v2'v3' v3' v3'v1 v1'v2 v2
v3v1 v3'v1 v3'v1 v1 v1v2 v1'v2 v2v2' v2'
v1v2 v2 v2v2' v2' v2'v2" v2"
Edges Output Edges Output Edges Output Edges Output
Final

3-D Geometric Transformations

Geometric Transformation : The object itself is moved relative to a stationary coordinate system or background.

With respect to some 3-D coordinate system, an object Obj is considered as a set of points.

$$Obj = \{ P(x,y,z)\}$$

If the Obj moves to a new position, the new object Obj' is considered:

$$Obj' = \{ P'(x',y',z')\}$$

Translation

Moving an object is called a translation. We translate an object by translating each vertex in the object.

$$x' = x + tx$$

$$y' = y + ty$$

$$z' = z + tz$$

The translating distance pair(tx, ty, tz) is called a **translation vector or shift vector.**

We can also write this equation in a single Matrix using column vectors:

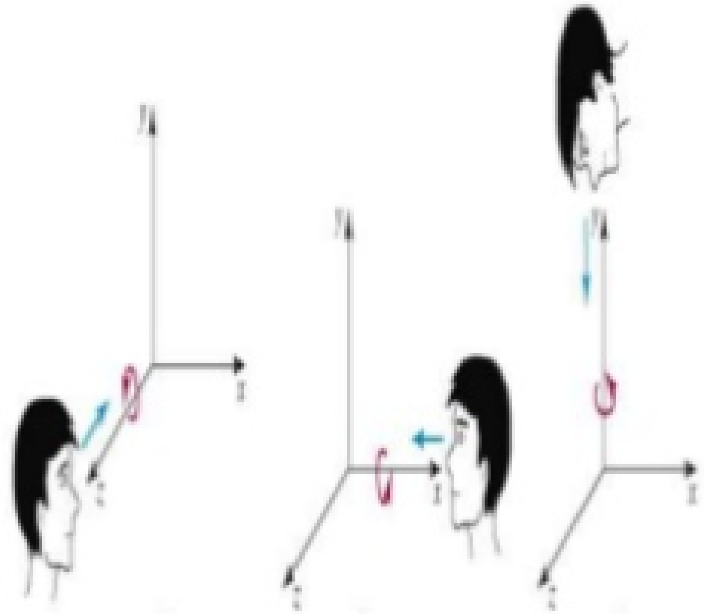
$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

3D ROTATION

Where an object is to be rotated about an axis that is parallel to one of the coordinate axis, we can obtain the desired rotation with the following transformation sequence.

Coordinate axis rotation

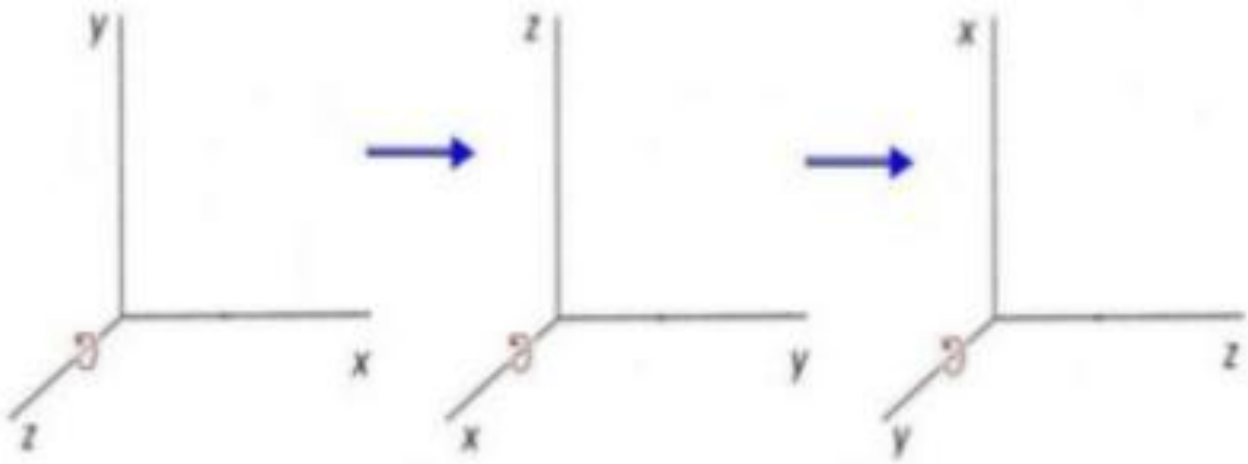
Z- axis Rotation(Roll)
Y-axis Rotation(Yaw)
X-axis Rotation(Pitch)



COORDINATE AXIS ROTATION

- Obtain rotations around other axes through cyclic permutation of coordinate parameters:

$$x \rightarrow y \rightarrow z \rightarrow x$$



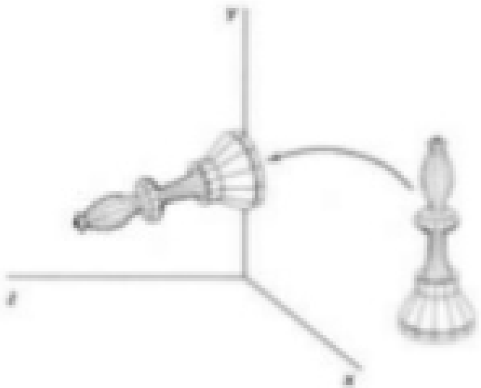
X-AXIS ROTATION

The equation for X-axis rotation

$x' = x$

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

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



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Y-AXIS ROTATION

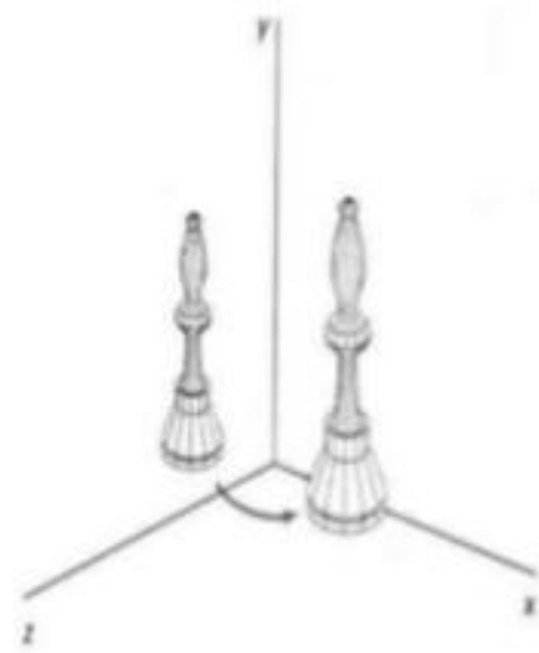
The equation for Y-axis rotaion

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

$y' = y$

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



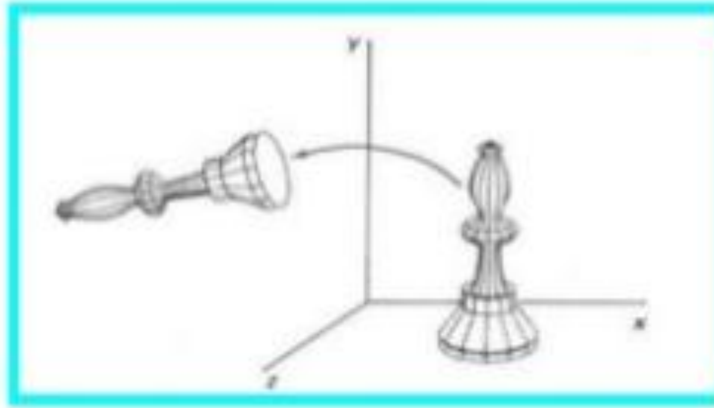
Z-AXIS ROTATION

The equation for Y-axis rotaion

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

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

$z' = z$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

Changing the size of an object is called Scaling . The scale factor s determines whether the scaling is a magnification, $s > 1$, Or a reduction, $s < 1$. Scaling with respect to the origin, where the origin remains fixed,

$$S_{sx,sy,sz} \Rightarrow \begin{cases} x' = x \cdot sx \\ y' = y \cdot sy \\ z' = z \cdot sz \end{cases}$$

The transformation equations can be written in the matrix form:

$$\begin{vmatrix} x' \\ y' \\ z' \end{vmatrix} = \begin{vmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & sz \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

3D REFLECTION

- Reflection in computer graphics is used to emulate reflective objects like mirrors and shiny surfaces
- Reflection may be an x-axis

y-axis , z-axis. and also in
the planes xy-plane,yz-plane , and
zx-plane.

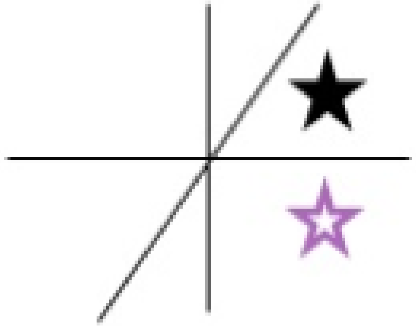
Reflection relative to a given
Axis are equivalent to 180
Degree rotations

3D REFLECTION

- Reflection about x-axis:-

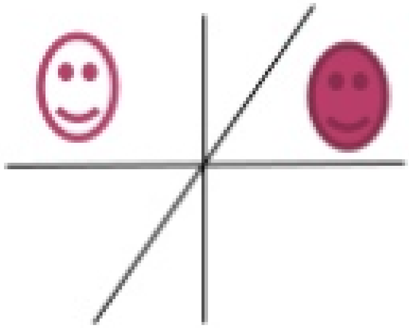
x’=x y’=-y z’=-z

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$



Reflection about y-axis:-

y’=y x’=-x z’=-z



3D REFLECTION

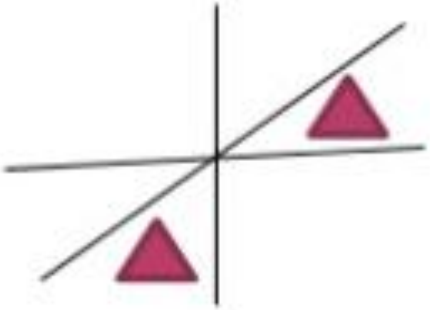
- The matrix for reflection about y-axis:-

$$\begin{matrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

- Reflection about z-axis:-

x'=-x　　y'=-y　　z'=z

$$\begin{matrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$



S J P N Trust's

Hirasugar Institute of Technology, Nidasoshi.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka and Affiliated to VTU Belagavi.

Accredited at 'A' Grade by NAAC

Programmes Accredited by NBA: CSE, ECE, EEE & ME

Module-3
18CS62

Clipping,3D Geometric Transformations, Color and Illumination Models:

Prof. Rahul Palakar

3D Rotation around arbitrary axis 1

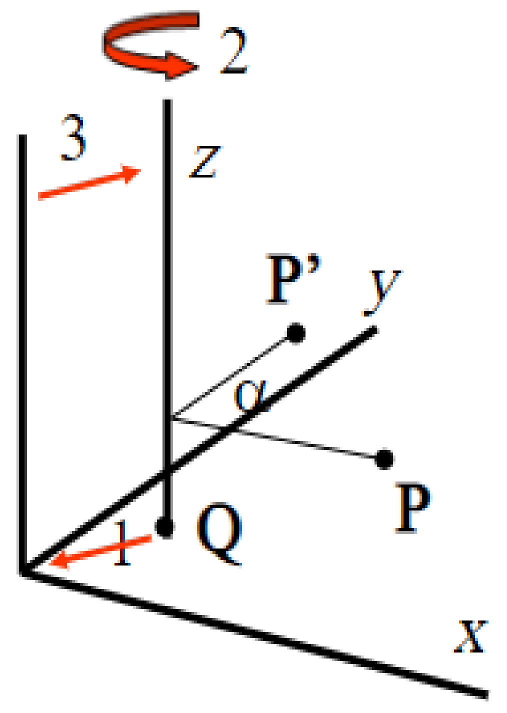
Rotation around axis, parallel to coordinate axis, through point **Q**.

For example. the z - as. Similar as 2D rotation :

1. Translate over -**Q**;
2. Rotate around z - axis;
3. Translate back over **Q**.

Or :

$$\mathbf{P}' = \mathbf{T}(\mathbf{Q})\mathbf{R}_z(\alpha)\mathbf{T}(-\mathbf{Q})\mathbf{P}$$



Rotation About an Arbitrary Axis in Space

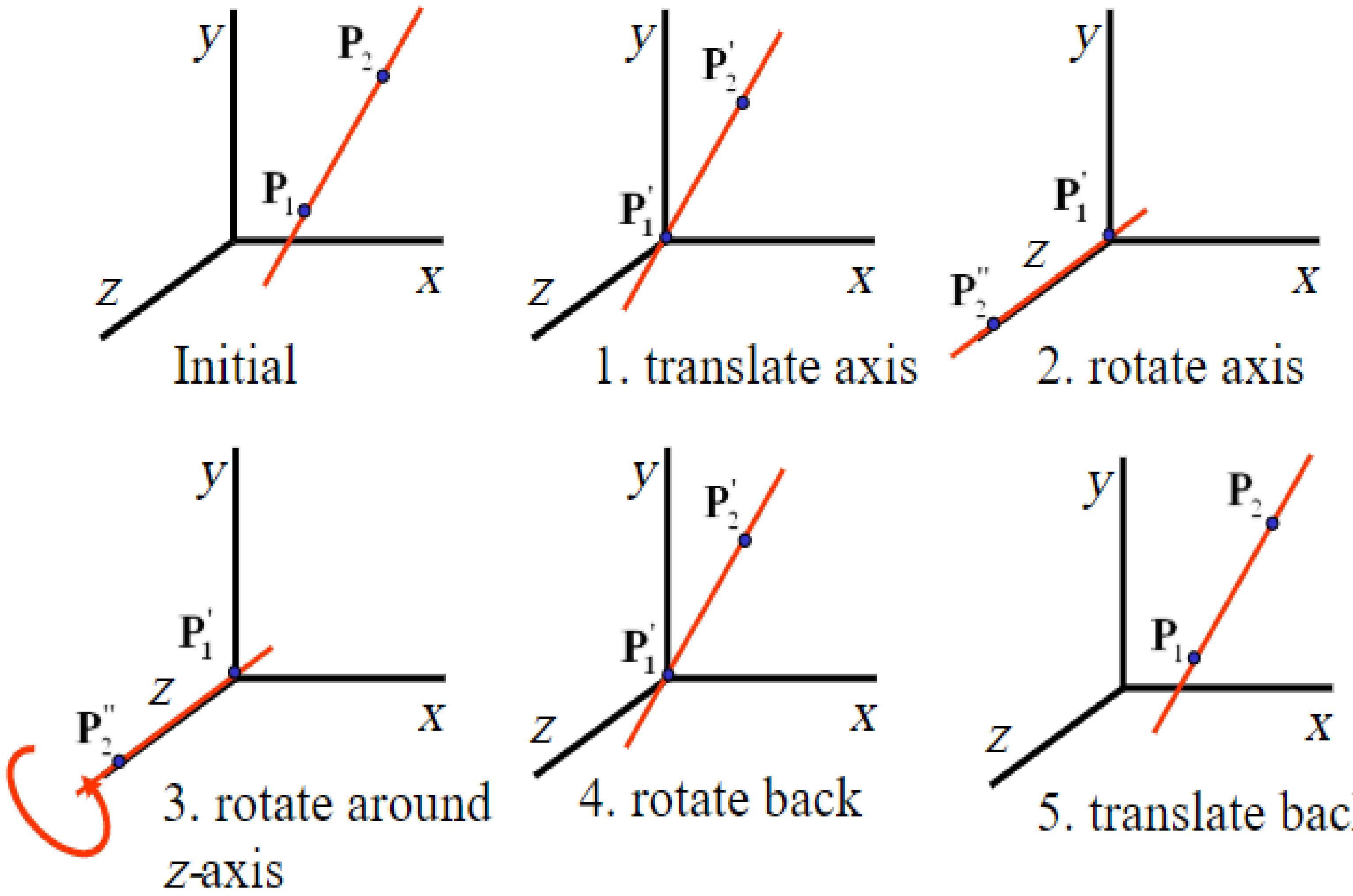
Assume, we want to perform a rotation by θ degrees, about an axis in space passing through the point (x_0, y_0, z_0) with direction cosines (c_x, c_y, c_z).

1. First of all, translate by:

$$|T| = -(x_0, y_0, z_0)^T$$

2. Next, we rotate the axis into one of the principle axes, let's pick, Z ($|R_x|$, $|R_y|$).
3. We rotate next by θ degrees in Z ($|R_z(\theta)|$).
4. Then we undo the rotations to align the axis.
5. We undo the translation: translate by $(-x_0, -y_0, -z_0)^T$

3D Rotation around arbitrary axis 3



The tricky part of the algorithm is in step(2), as given before.

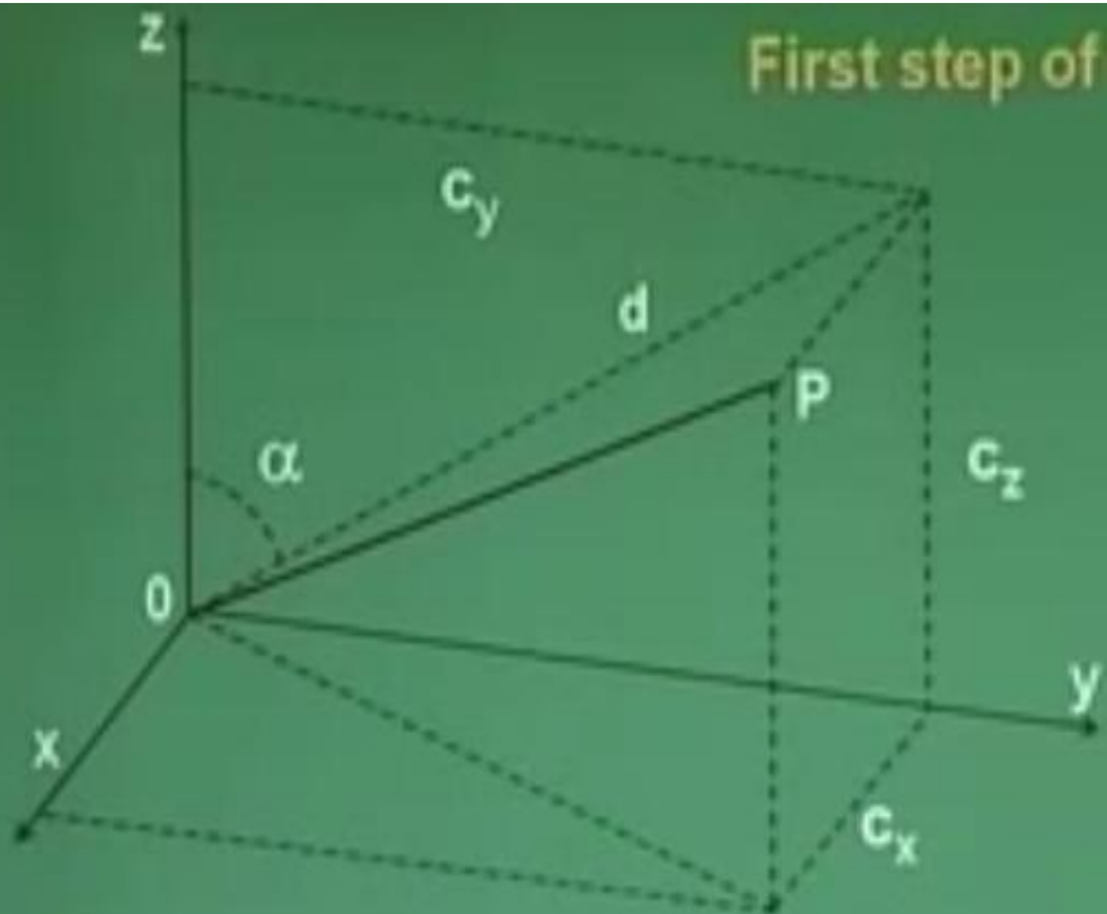
This is going to take 2 rotations:

i) About x-axis
(to place the axis in the xz plane)

and

ii) About y-axis
(to place the result coincident with the z-axis).

First step of Rotation:



Rotation about x by α:

How do we determine α?

Project the unit vector, along OP, into the yz plane.

The y and z components, c_y and c_z, are the direction cosines of the unit vector along the arbitrary axis.

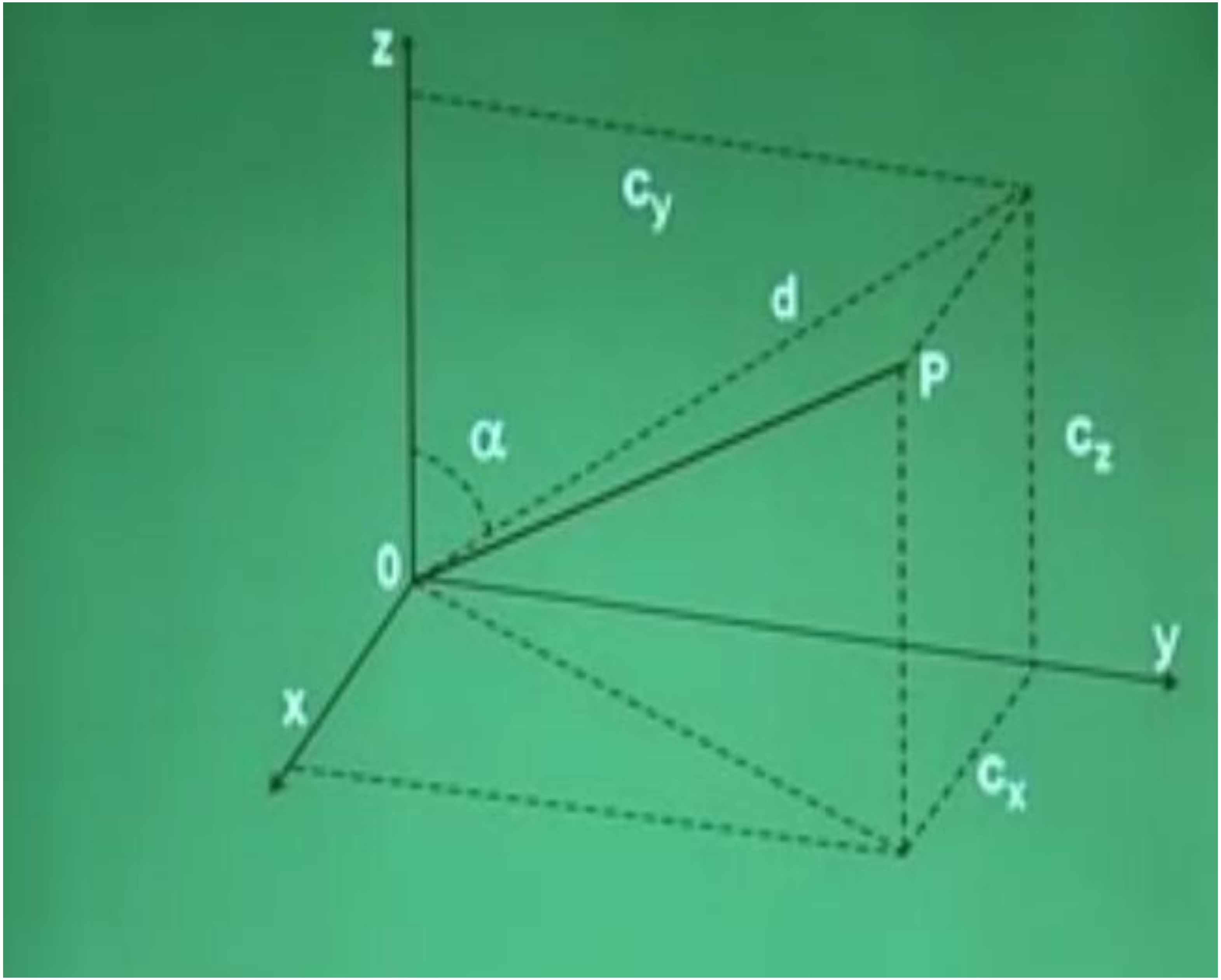
It can be seen from the diagram, that :

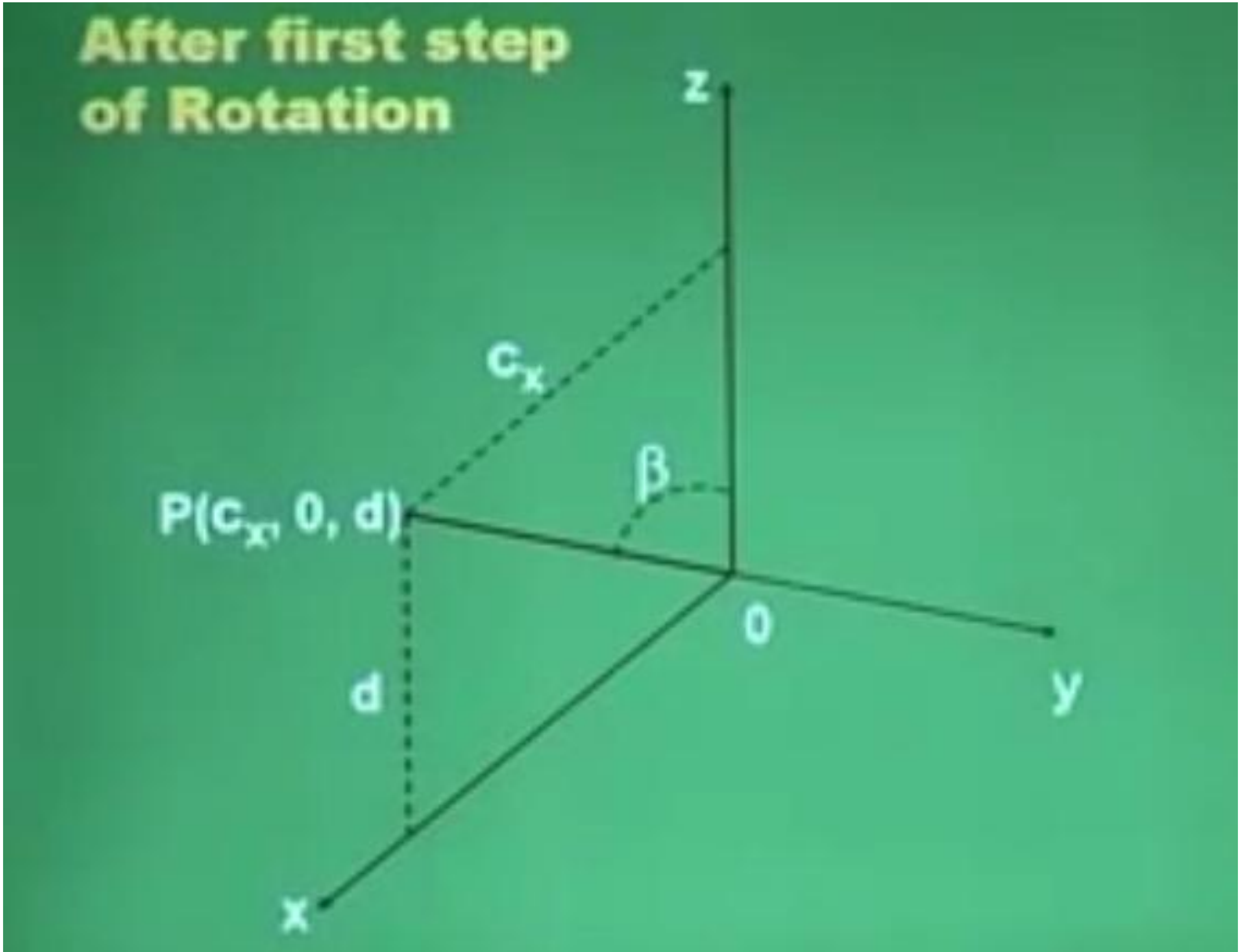
$$d = \mathrm{sqrt}\left(C_y'^2 + C_z'^2\right)$$

$$\cos(\alpha) = {C_z}/{d}$$

$$\sin(\alpha) = {C_y}/{d}$$

$$\alpha = \sin^{-1}\left[\frac{c_y}{\sqrt{c_y^2 + c_z^2}}\right]$$


z
c_y
d
P
α
c_z
0
y
x
c_x


After first step
of Rotation
z
c_x
β
P(c_x, 0, d)
0
d
y
x

Rotation by β about y:

How do we determine β?

Steps are similar to that done for α:

- Determine the angle β to rotate the result into the Z axis:
- The x component is c_x and the z component is d.

$$\cos(\beta) = d = d/(\text{length of the unit vector})$$

$$\sin(\beta) = c_x = c_x/(\text{length of the unit vector}).$$

Final Transformation for 3D rotation, about an arbitrary axis:

$$M = |T|\ |R_x|\ |R_y|\ |R_z|\ |R_y|^{-1}\ |R_x|^{-1}\ |T|^{-1}$$

Final Transformation matrix for 3D rotation, about an arbitrary axis:

$$M = |T|\,|R_x|\,|R_y|\,|R_z|\,|R_y|^{-1}\,|R_x|^{-1}\,|T|^{-1}$$

where:

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\qquad R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_z/d & -C_y/d & 0 \\ 0 & C_y/d & C_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$R_y = \begin{bmatrix} d & 0 & -C_x & 0 \\ 0 & 1 & 0 & 0 \\ C_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\qquad R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

S J P N Trust's

Hirasugar Institute of Technology, Nidasoshi.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka and Affiliated to VTU Belagavi.

Accredited at 'A' Grade by NAAC

Programmes Accredited by NBA: CSE, ECE, EEE & ME

Module-3

18CS62

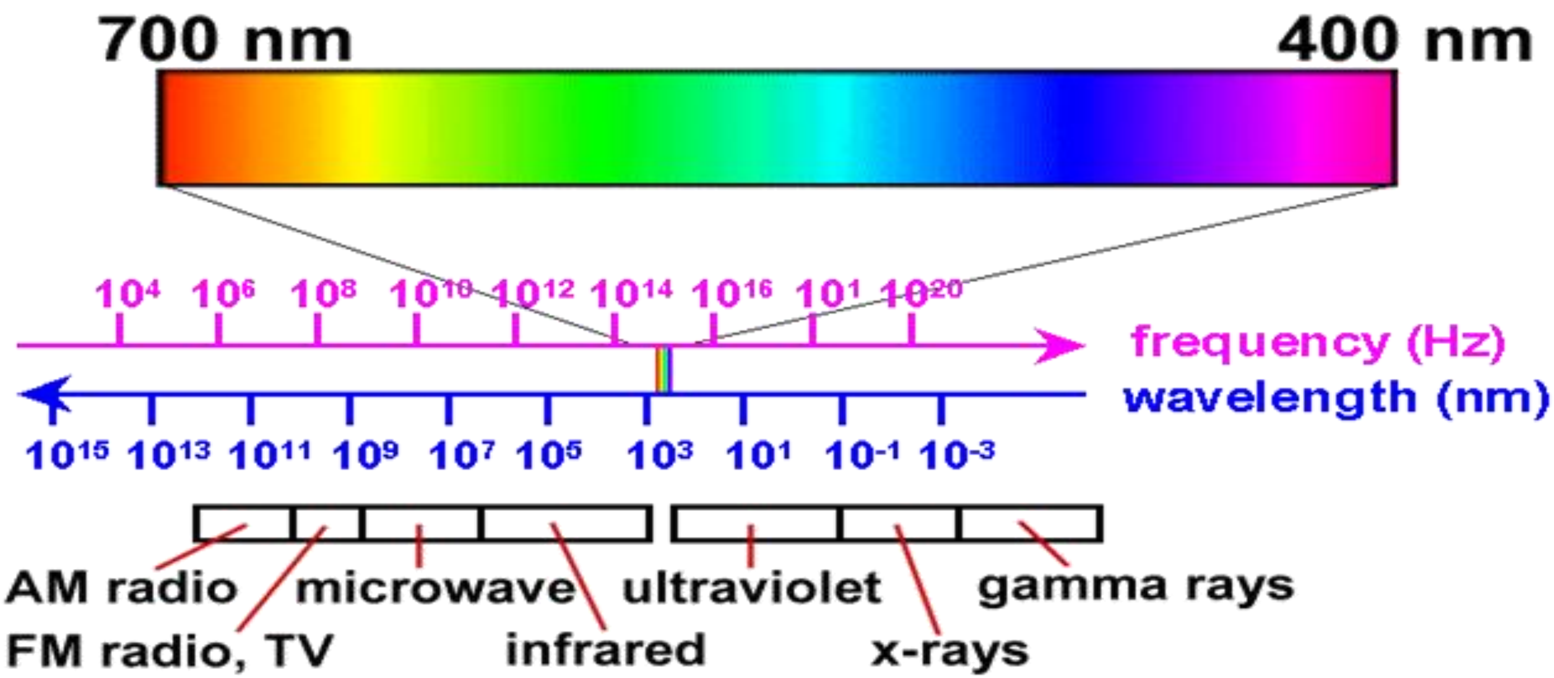
Clipping,3D Geometric Transformations, Color and Illumination Models:

Color Models

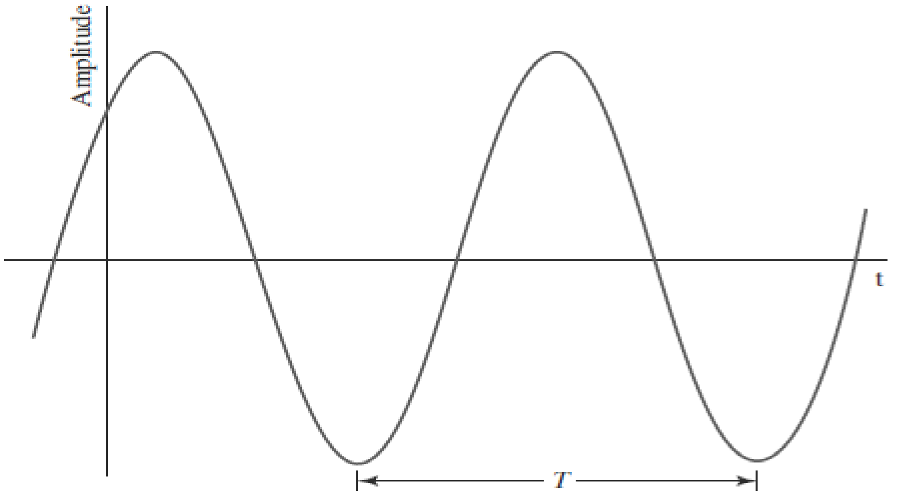
Prof. Rahul Palakar

Properties of Light

- What is light?
  - "light" = narrow frequency band of electromagnetic spectrum
- Each frequency value within the visible region of the electromagnetic spectrum corresponds to a distinct **spectral color.**
- The Electromagnetic Spectrum
  - Red: 3.8×10^{14} hertz
  - Violet: 7.9×10^{14} hertz



- In the wave model of electromagnetic radiation, light can be described as oscillating transverse electric and magnetic fields propagating through space.
- For each spectral color, the rate of oscillation of the field magnitude is given by the frequency *f*.



- The time between any two consecutive positions on the wave that have the same amplitude is called the *period* (T) of the wave, which is the inverse of the frequency (i.e., $T = 1/f$).
- And the distance that the wave has traveled from the beginning of one oscillation to the beginning of the next oscillation is called the *wavelength* (λ).
- Speed of light (c): $c = \lambda f$.
- A light source such as the sun or a standard household light bulb emits all frequencies within the visible range to produce white light.

- When white light is incident upon an opaque object, some frequencies are reflected and some are absorbed.
- The combination of frequencies present in the reflected light 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.
- In this case, we say that the perceived light has a **dominant frequency** (or **dominant wavelength**) at the red end of the spectrum.
- The dominant frequency is also called the **hue,** or simply the **color,** of the light.

Psychological Characteristics of Color

- When we view a source of light, our eyes respond to the color (or dominant frequency) and two other basic sensations.

1. **brightness:** which corresponds to the total light energy and can be quantified as the luminance of light.

2. purity, or the **saturation:** Purity describes how close a light appears to be to a pure spectral color, such as red.

Another term, **chromaticity**, is used to refer collectively to the two properties describing color characteristics: purity and dominant frequency (hue).

White Light

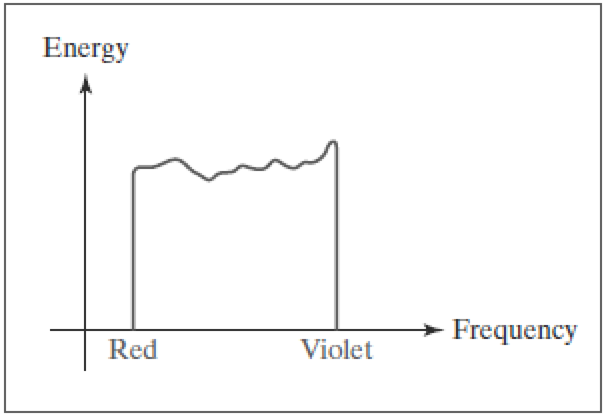


FIGURE 3
Energy distribution for a white light source.

Dominant frequency light

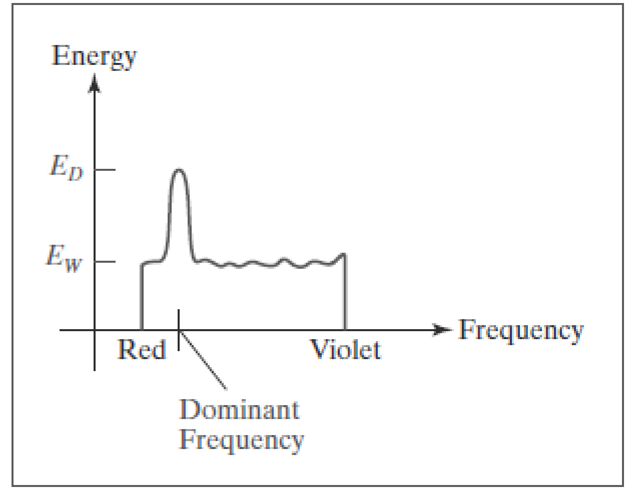


FIGURE 4
Energy distribution for a light source with a dominant frequency near the red end of the frequency range.

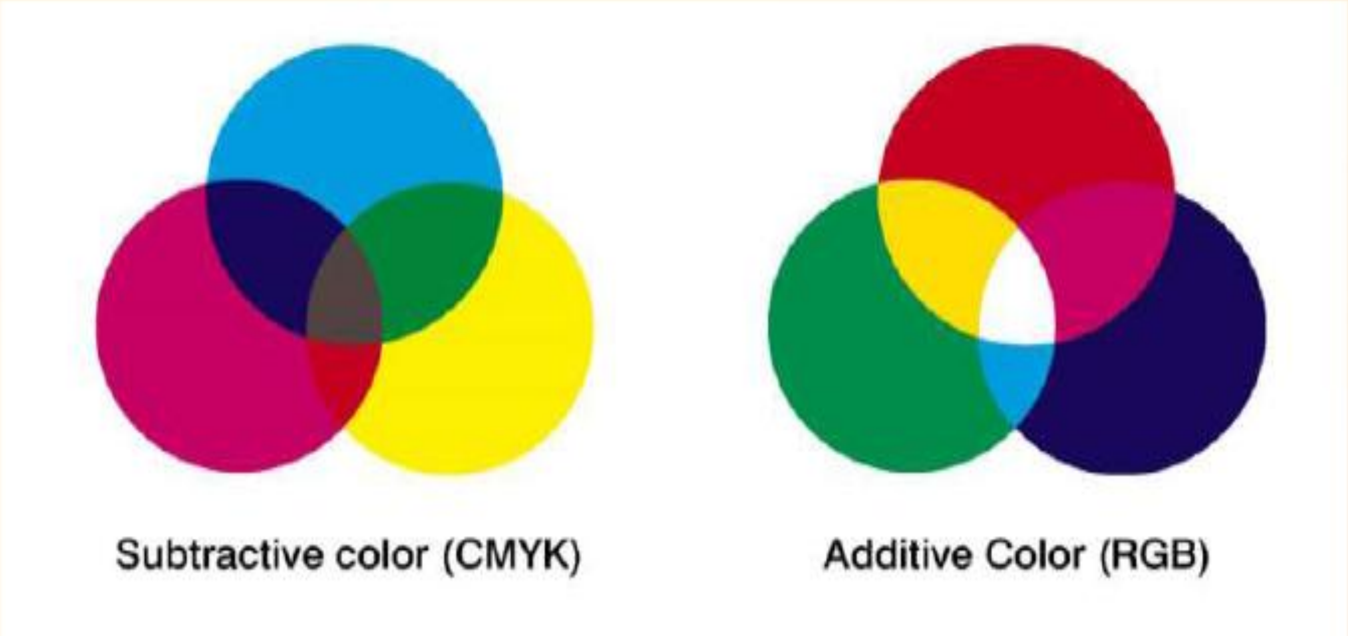
- We can calculate the brightness of the source as the area under the curve, which gives the total energy density emitted.
- Purity (saturation) depends on the difference between E_D and E_w.
- The larger the energy E_D of the dominant frequency compared to the white-light component E_W, the higher the purity of the light.
- We have a purity of 100 percent when E_W = 0 and a purity of 0 percent when $E_W = E_D$.

Color Model

- A color model is an orderly system for **creating a whole range of colors** from a small set of **primary colors**. There are two types of color models, those that are **subtractive** and those that are **additive**.
- ***Additive*** color models **use light** to display color while ***subtractive*** models **use printing inks**. Colors perceived in additive models are the result of **transmitted light**. Colors perceived in subtractive models are the result of **reflected light**.

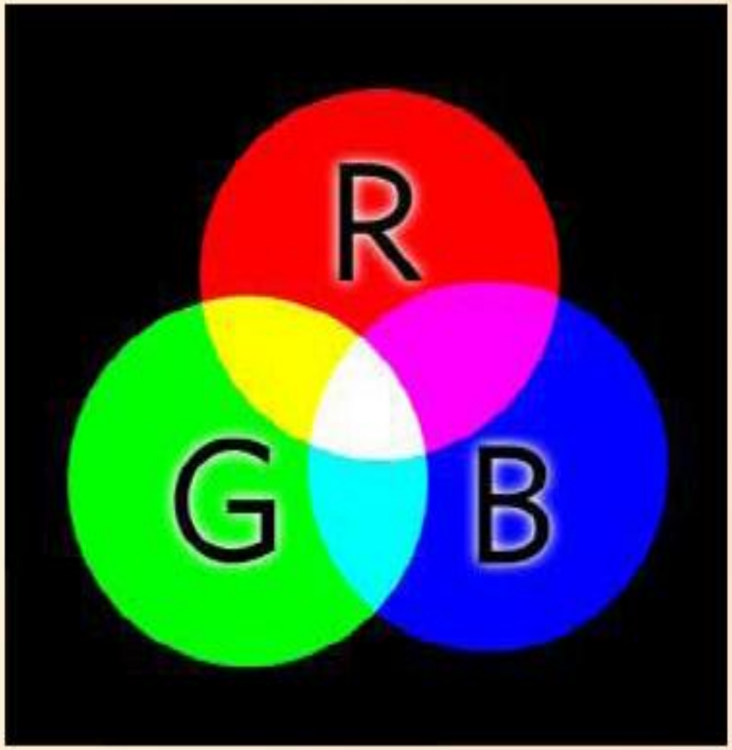
Color Model

- There are several established color models used in computer graphics, but the two most common are the **RGB model** (Red-Green-Blue) for computer display and the **CMYK model** (Cyan-Magenta-Yellow-Black) for printing.



RGB Color Model

- RGB is an additive color model For computer displays **uses light** to display color , Colors result from **transmitted light**
- Red + Green + Blue = White

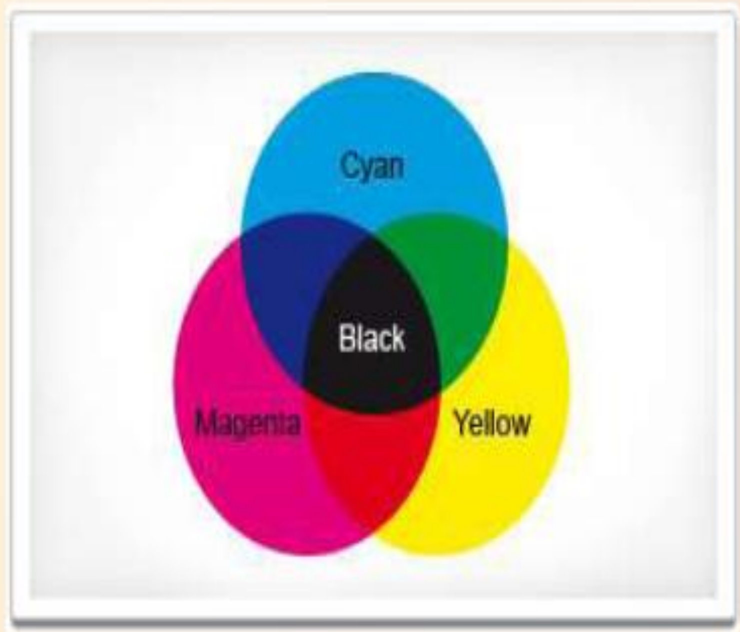


CMYK Color Model

CMYK (subtractive color model) is the standard color model used in offset printing for full-color documents. Because such printing uses inks of these four basic colors, it is often called **four-color printing**.

- Where two colors of RGB overlaps, we see a new color formed by mixing of the two additive primaries. These new colors are:

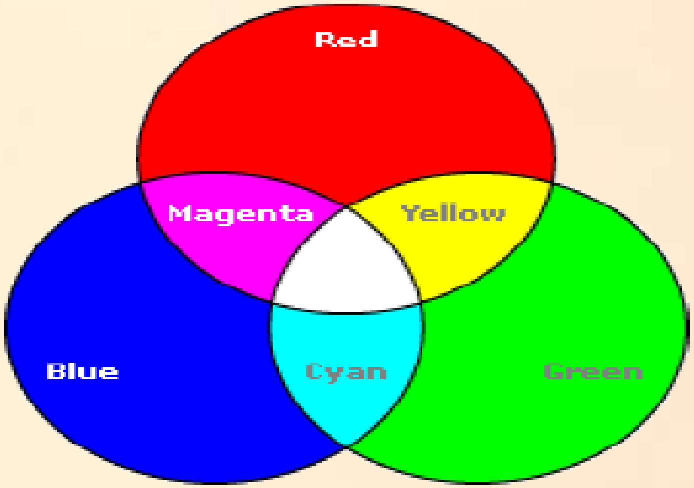
- A greenish blue called cyan.
- A blushed red called magenta.
- A bright yellow.
- The key color , **Black**.



CMYK Color Model

We can express this effect pseudo-algebraically. Writing R, G and B for **red, green** and **blue**, C, M and Y for cyan, magenta and yellow, and **W** for **white**, and using (+) to mean additive mixing of light, and (–) to mean subtraction of light, we have:

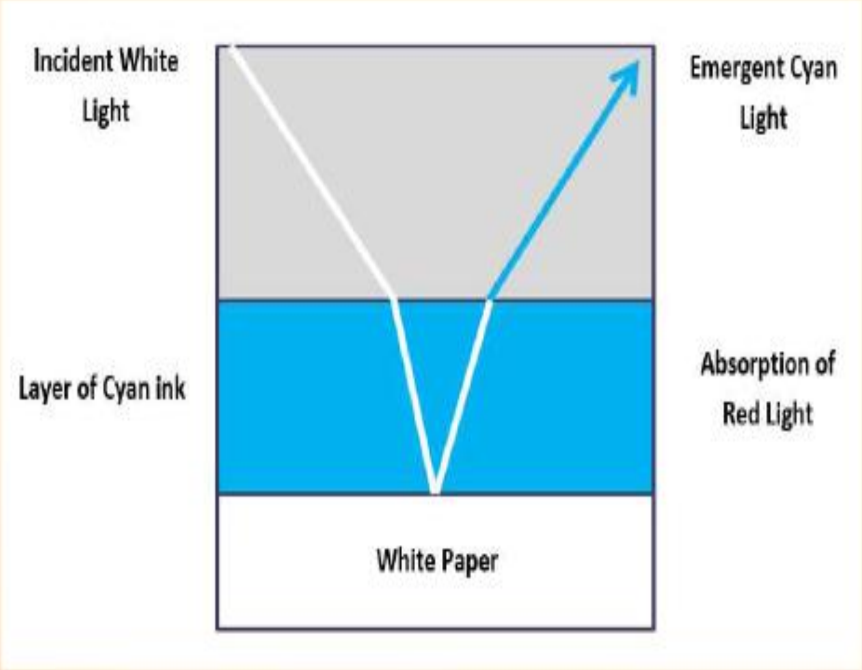
- C (cyan) = G + B = W - R
- M (magenta) = R + B = W - G
- Y (yellow) = R + G = W – B

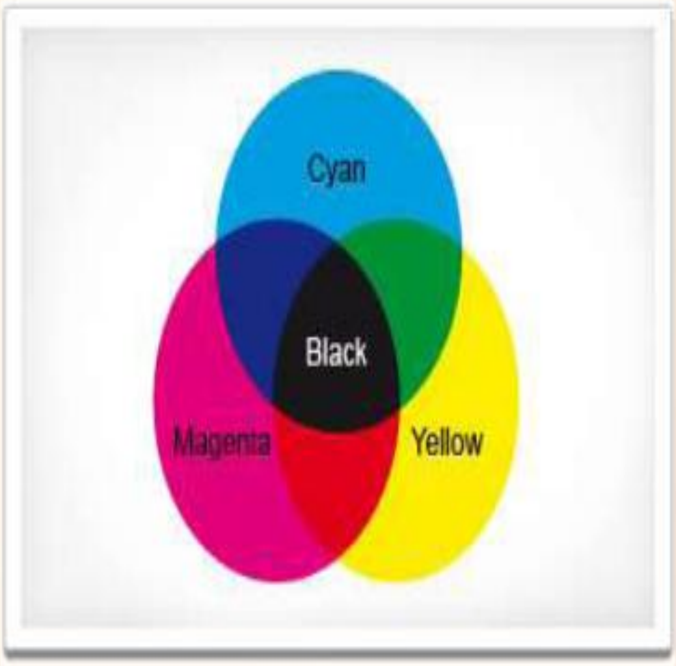


In each equation, the colour on the left is called the **complementary** colour of the one at the extreme right; for example, magenta is the complementary colour of green.

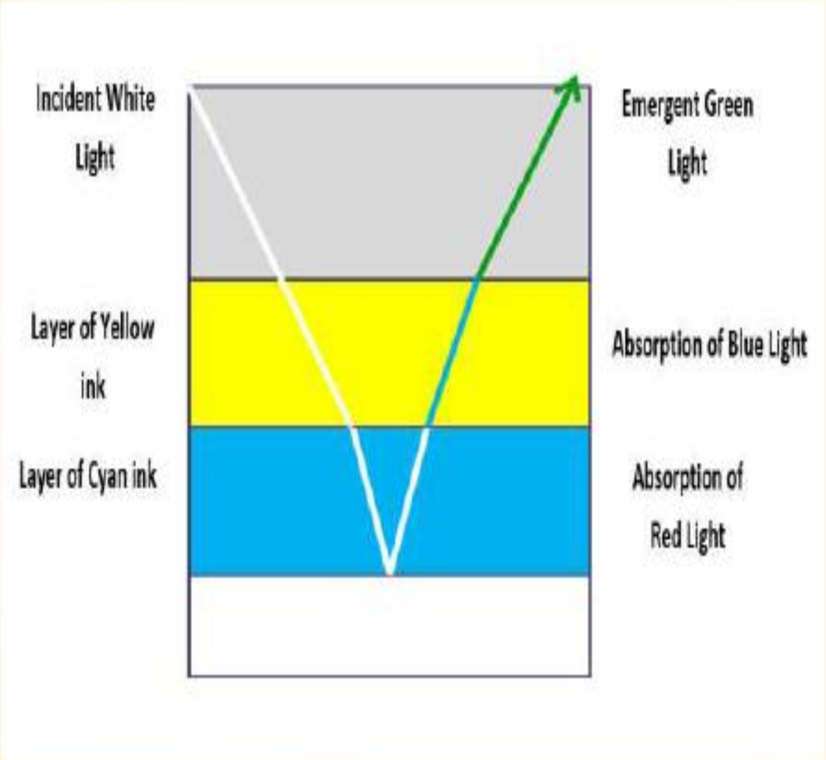
The process of reflection

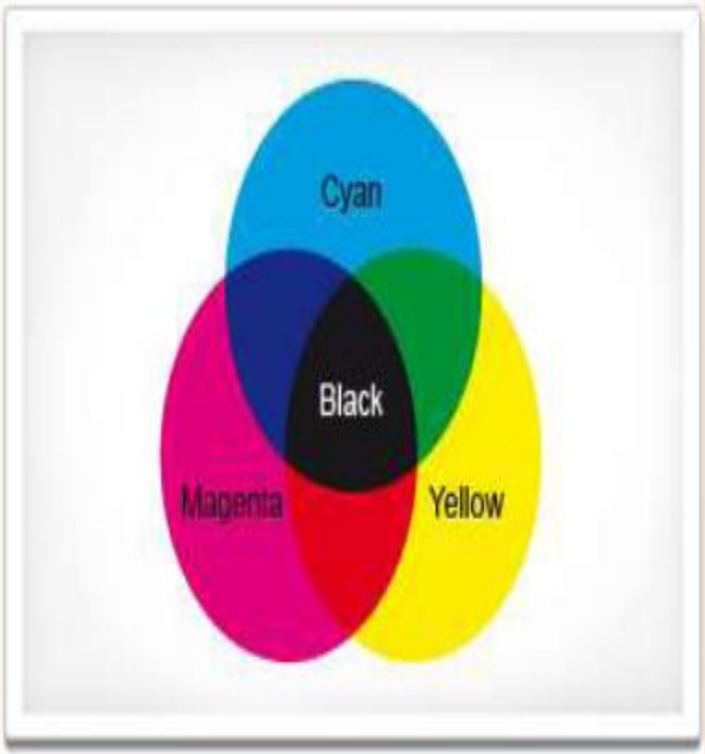
when we talk of 'cyan ink', we mean ink that, when it is applied to **white paper** and illuminated by white light will **absorb** the **red** component, allowing the **green** and **blue**, which combine to produce the **cyan** colour, to be reflected back.





• If we apply a layer of such an ink to white paper, and then add a layer of yellow, the yellow ink will absorb incident **blue** light, so the combination of the cyan and yellow inks produces a **green** colour.



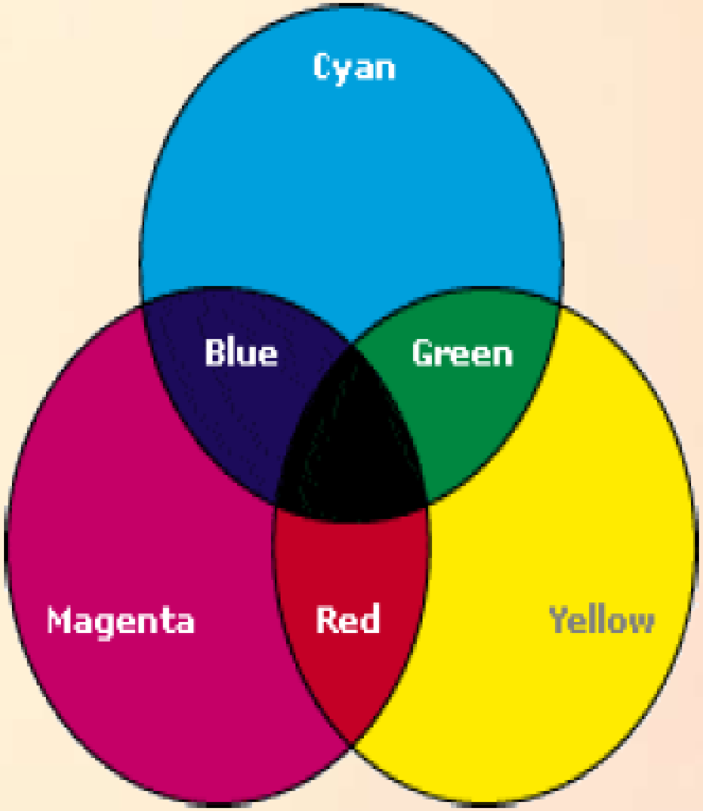


Similarly, combining cyan and magenta inks produces **blue.** A combination of all three colours will **absorb** all incident light, producing **black**.

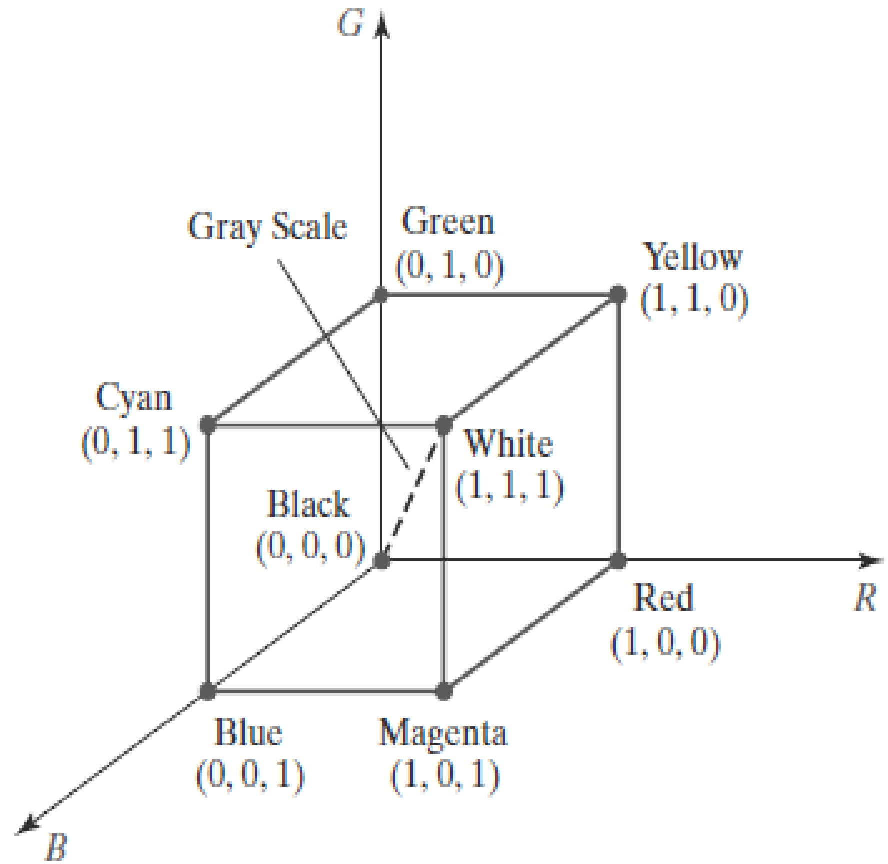
CMY(K): printing

- Cyan, Magenta, Yellow (Black) – CMY(K)
- A subtractive color model

dye color	*absorbs*	*reflects*
cyan	red	blue and green
magenta	green	blue and red
yellow	blue	red and green
black	all	none



These subtractive primaries are the primary colours which **the artist working in conventional media must use**. Besides, the range of colours that can be produced **by mixing of primary coloured paint**.



Shades of gray are represented along the main diagonal of the cube from the origin (black) to the white vertex.

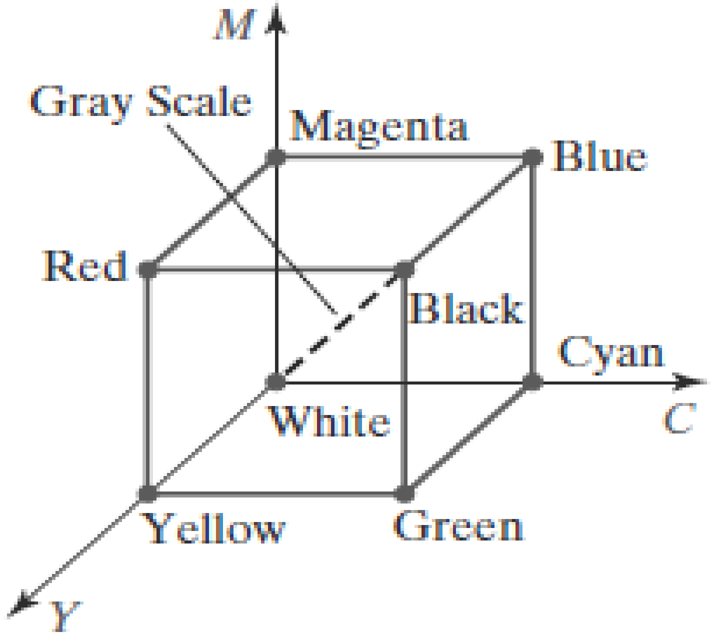


FIGURE 13
The CMY color model. Positions within the unit cube are described by subtracting the specified amounts of the primary colors from white.

Thank you

S J P N Trust's

Hirasugar Institute of Technology, Nidasoshi.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka and Affiliated to VTU Belagavi.

Accredited at 'A' Grade by NAAC

Programmes Accredited by NBA: CSE, ECE, EEE & ME

Module-3
18CS62

Clipping,3D Geometric Transformations, Color and Illumination Models:

Illumination Models

Prof. Rahul Palakar

Illumination Model

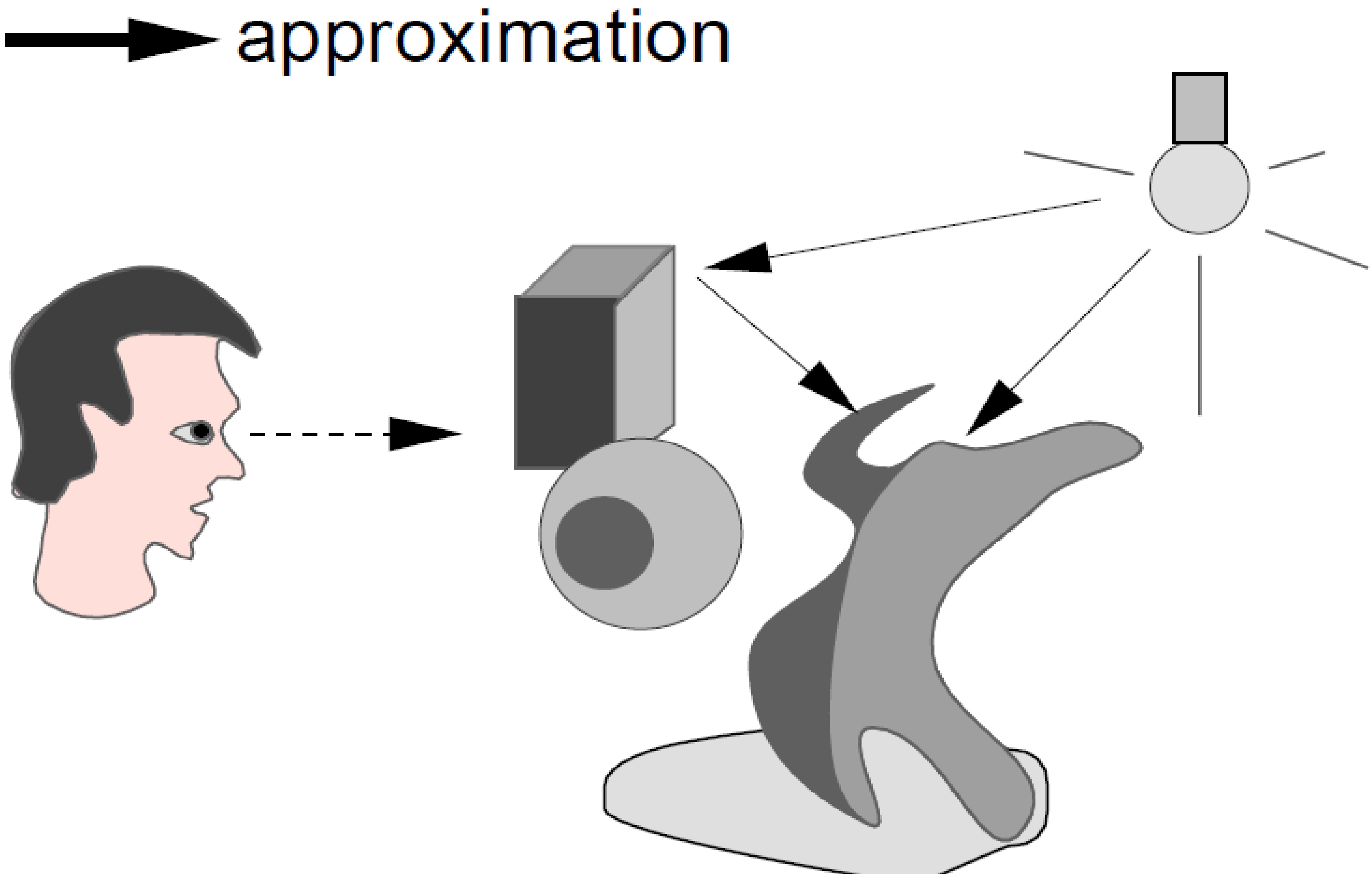
- **Motivation:** In order to produce realistic images, we must simulate the appearance of surfaces under various lighting conditions
- An illumination model, also called a lighting model and sometimes referred to as a shading model, is used to calculate the intensity of light that we should see at a given point on the surface of an object.

- Surface rendering means a procedure for applying a lighting model to obtain pixel intensities for all the projected surface positions in a scene.
- A surface-rendering algorithm uses the intensity calculations from an illumination model to determine the light intensity for all projected pixel positions for the various surfaces in a scene.
- Surface rendering can be performed by applying the illumination model to every visible surface point

Illumination Models

- Illumination models are used to generate the colour of an object's surface at a given point on that surface.
- The factors that govern the illumination model determine the visual representation of that surface.
- Due to the relationship defined in the model between the surface of the objects and the lights affecting it, illumination models are also called shading models or lighting models.

- Modeling the colors and lighting effects that we see on an object is a complex process.



Light sources

- Sun, light bulbs, and any other light–emitting sources
- How about light–reflected sources?

Point light sources

- emits light equally in all directions

Spotlights

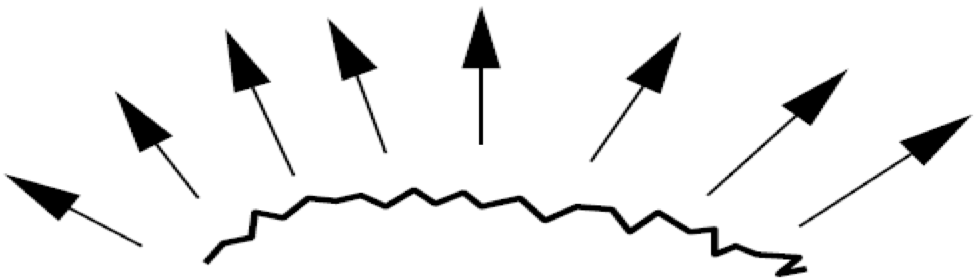
- a narrow range of angles through which light is emitted

Distant light sources

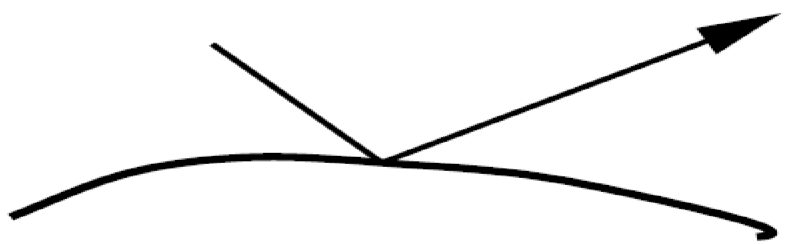
- parallel light , sun

Surface types

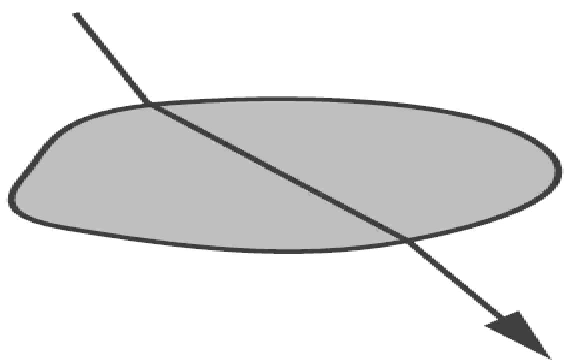
- Rough, grainy surfaces tend to scatter light.



- Glossy, shiny surfaces result in highlighting effect



- Transparent surfaces can transmit light



Basic illumination models

1. Ambient light
2. Diffuse reflection
3. Specular reflection

Ambient light:

- model the combination of light reflections from surrounding objects in the scene
- no spatial or directional characteristics

$\mathbf{I = k_a I_a}$ where I_a is the intensity of the ambient light, and k_a (the ambient reflection coefficient) is the percentage of ambient light reflected from the object's surface.