RGPV Notes Hub

IT601 – Computer Graphics & Multimedia

Unit – III

Topic – 18: Sutherland – Hodgman Polygon Clipping Algorithm

Sutherland – Hodgman Polygon Clipping Algorithm

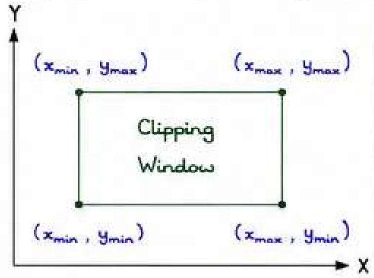
1. Introduction

Polygon clipping is the process of removing the parts of a polygon that lie outside a specified clipping window. The Sutherland – Hodgman algorithm is a simple and efficient polygon clipping algorithm which works for any convex clipping window.

2. Clipping Window

A rectangular clipping window is defined by its boundaries.

- x_{min} – Left boundary
- x_{max} – Right boundary
- y_{min} – Bottom boundary
- y_{max} – Top boundary



3. Basic Idea

The clipping window is considered as a sequence of four (or more) boundaries in the order:

Left → Right → Bottom → Top

The subject polygon is clipped successively against each boundary. The output of one boundary becomes the input polygon for the next boundary.

4. Inside Test for Boundaries

A point (x, y) is inside a boundary if:

Boundary	Inside Condition
Left	$x \ge x_{min}$
Right	$x \le x_{max}$
Bottom	$y \ge y_{min}$
Top	$y \le y_{max}$

Inside → I Outside → O

5. Algorithm

Input : Subject polygon P (with vertices in order), Clipping window (convex).

Output : Clipped polygon P_{out}.

Step 1 : Set P_{out} = P (input polygon).

Step 2 : For each boundary (Left, Right, Bottom, Top) do the following steps on P_{out} :

(a) Take the first vertex as S (starting vertex).

(b) For each vertex E (ending vertex) in P_{out} (in order) do :

- If E is inside and S is inside → Output E.
- If E is inside and S is outside → Output the intersection point of SE with the boundary, then output E.
- If E is outside and S is inside → Output the intersection point of SE with the boundary.
- If E is outside and S is outside → Output nothing.

Then S = E (move to next edge).

Step 3 : The output after Top boundary is the final clipped polygon.

6. Intersection with Boundaries

For an edge from $S(x_1, y_1)$ to $E(x_2, y_2) = P_1P_2$, intersection with each boundary is :

- Left $(x = x_{min})$:

$$y = y_1 + (y_2 - y_1) \cdot \frac{x_{min} - x_1}{x_2 - x_1}$$

- Right $(x = x_{max})$:

$$y = y_1 + (y_2 - y_1) \cdot \frac{x_{max} - x_1}{x_2 - x_1}$$

- Bottom $(y = y_{min})$:

$$x = x_1 + (x_2 - x_1) \cdot \frac{y_{min} - y_1}{y_2 - y_1}$$

- Top $(y = y_{max})$:

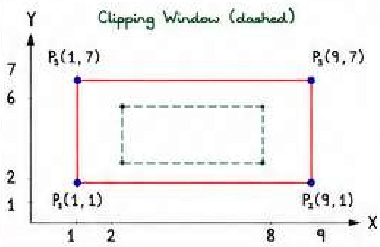
$$x = x_1 + (x_2 - x_1) \cdot \frac{y_{max} - y_1}{y_2 - y_1}$$

7. Example

Clipping Window : $x_{min} = 2$, $x_{max} = 8$, $y_{min} = 2$, $y_{max} = 6$

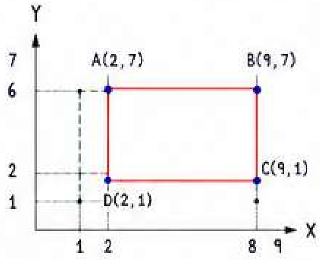
Input Polygon (in order) : $P_1(1, 1)$, $P_2(9, 1)$, $P_3(9, 7)$, $P_4(1, 7)$

We clip against Left → Right → Bottom → Top



(ii) Clip against Left boundary (x = 2)

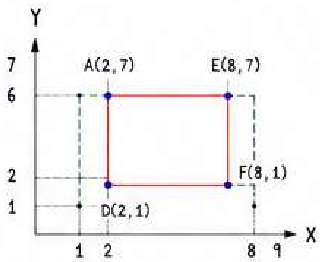
Output after Left clip



A(2,7), B(9,7), C(9,1), D(2,1)

(iii) Clip against Right boundary (x = 8)

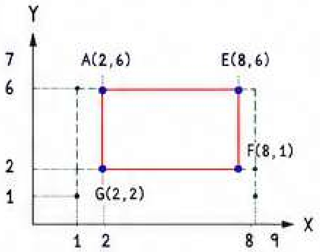
Output after Right clip



A(2,7), E(8,7), F(8,1), D(2,1)

(iii) Clip against Bottom boundary (y = 2)

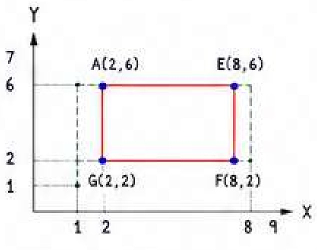
Output after Bottom clip



A(2,6), E(8,6), F(8,2), G(2,2)

(iv) Clip against Top boundary (y = 6)

Output after Top clip (Final)



Final Clipped Polygon :
(2,6) → (8,6) → (8,2) → (2,2)

8. Advantages

- ✓ Simple and easy to implement.
- ✓ Guarantees a convex output polygon when clipping window is convex.
- ✓ Efficient for small number of boundaries (like rectangle).
- ✓ Works for any number of edges in input polygon.

9. Limitations

- ✗ Works only for convex clipping windows.
- ✗ Not suitable for concave windows.
- ✗ More computations for polygons with large number of vertices.

10. Applications

- Computer graphics and visualization.
- CAD / CAM systems.
- GIS and mapping systems.
- Image processing.
- Simulation and gaming.
- Robot path planning.

11. Key Points

- Window is treated as sequence of boundaries : Left → Right → Bottom → Top.
- At each boundary, output of previous step becomes input.
- Uses parametric line intersection.
- Final output is the clipped polygon completely inside the window.

12. Summary

The Sutherland – Hodgman polygon clipping algorithm clips a polygon successively against each boundary of a convex clipping window using inside – outside tests and intersection calculations. It is simple, systematic and widely used for convex clipping windows.

Typical Exam Questions

1. Explain Sutherland – Hodgman polygon clipping algorithm with neat diagram.
2. Describe the steps of Sutherland – Hodgman algorithm.
3. Clip the given polygon against a rectangular window using Sutherland – Hodgman algorithm.
4. Write short notes on polygon clipping.
5. Compare Sutherland – Hodgman with Weiler – Atherton polygon clipping algorithm.

Key Formulas

- Left $(x = x_{min})$: $y = y_1 + (y_2 - y_1)(x_{min} - x_1) / (x_2 - x_1)$
- Right $(x = x_{max})$: $y = y_1 + (y_2 - y_1)(x_{max} - x_1) / (x_2 - x_1)$
- Bottom $(y = y_{min})$: $x = x_1 + (x_2 - x_1)(y_{min} - y_1) / (y_2 - y_1)$
- Top $(y = y_{max})$: $x = x_1 + (x_2 - x_1)(y_{max} - y_1) / (y_2 - y_1)$

I = Inside O = Outside

RGPV Notes Hub

Keep Learning, Keep Growing !

RGPV Notes Hub

IT601 - Computer Graphics & Multimedia

Unit - III

Introduction to Hidden Surface Elimination

Topic - 18
Introduction to Hidden Surface Elimination

1. Introduction

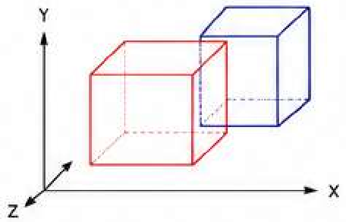
In a 3D scene, some surfaces are not visible to the observer because they are occluded by other surfaces. Such surfaces are called hidden surfaces. Hidden surface elimination (HSE) is the process of identifying visible surfaces or parts of surfaces and removing hidden ones from the display. HSE is essential for realistic and correct image generation.

2. Need for HSE

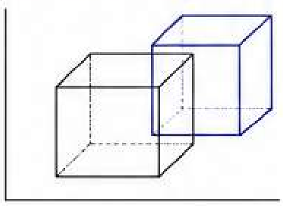
* To obtain a realistic view of a scene.
* To reduce the amount of data to be displayed.
* To improve the efficiency of rendering.
* To remove visual confusion caused by hidden parts.

3. Example

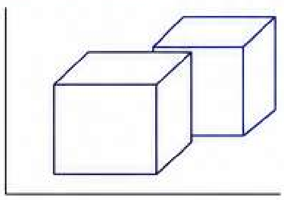
(a) Actual 3D Scene



(b) Without HSE (All edges drawn)



(c) With HSE (Hidden parts removed)

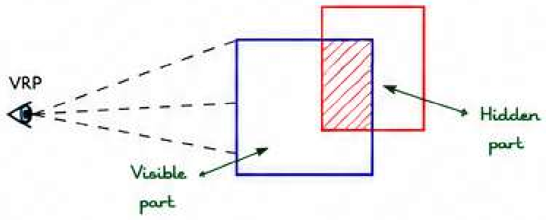


Key Points

* Hidden surface problem is fundamental in CG.
* HSE ensures only visible parts are drawn.
* Efficiency and accuracy depend on the method used.
* Choice of algorithm depends on the type of objects and applications.

4. Basic Concept

A surface is visible if every point on it is visible from the view reference point (VRP). If any part of a surface is occluded by another surface, it is (partially) hidden.



Front surface (closer to VRP)

Back surface (farther from VRP)

5. Categories of Hidden Surfaces

* Completely hidden - Entire surface is not visible.
* Partially hidden - Only a portion of the surface is visible.
* Self-hidden - A surface hides some of its own parts (e.g., concave objects).

6. General Approaches for HSE

There are two broad approaches for hidden surface elimination:

(i) Object-space methods

Hidden surfaces are identified in 3D (world) coordinates before projection.

Examples: Depth buffer (Z-buffer), Octree method, BSP tree, Scan-line method.

(ii) Image-space methods

Hidden surfaces are identified after projection in 2D (screen) coordinates.

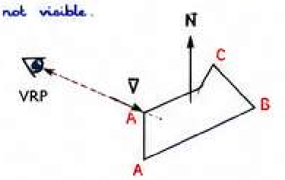
Examples: Back-face elimination, Area subdivision, Painter's algorithm.

7. Common HSE Algorithms (Overview)

Algorithm	Type	Key Idea
1. Back-face elimination	Image-space	Remove polygons facing opposite to viewing direction.
2. Depth buffer (Z-buffer)	Object-space	Store depth of nearest surface at each pixel.
3. Scan-line (AEL)	Image-space	Process scan lines using active edge list and depth.
4. BSP Tree	Object-space	Divide space using planes to order visibility.
5. Octree method	Object-space	Subdivide space into octants and test visibility.

8. Back-Face Elimination (Basic Idea)

For a polygon, if the angle between its normal and the viewing direction is more than 90°, it is facing away from the observer and hence is not visible.



Condition for visibility:

$\vec{N} \cdot \vec{V} < 0 \rightarrow$ Visible

$\vec{N} \cdot \vec{V} > 0 \rightarrow$ Not visible

Where,
N = Normal vector of polygon
V = Viewing vector (from polygon to VRP)

9. Factors Affecting HSE

* Complexity of the scene.
* Number of objects and surfaces.
* Type of objects (convex / concave).
* Real-time or non-real-time application.
* Available memory and computing power.

10. Applications

* Realistic 3D rendering and visualization.
* CAD / CAM systems.
* Flight simulators.
* Virtual reality.
* Scientific visualization.
* Video games and animations.

11. Advantages

✓ Produces realistic images.
✓ Improves display efficiency.
✓ Reduces amount of data to be processed.
✓ Essential for complex 3D scenes.
✓ Enhances performance in rendering.

12. Limitations

✗ Some algorithms are computationally expensive.
✗ Difficulty with complex, highly concave objects.
✗ Requires more memory (e.g., Z-buffer).
✗ Algorithms may be sensitive to numerical errors.

13. Summary

Hidden surface elimination is the process of determining and displaying only the visible parts of objects in a scene. It is essential for realistic and correct image generation. Various algorithms exist for HSE, each with its own advantages and limitations. The choice of algorithm depends on the application, scene complexity and available resources.

Typical Exam Questions

1. What is hidden surface elimination? Why is it necessary?
2. Write short notes on hidden surface elimination.
3. Explain the basic approaches for hidden surface elimination.
4. Explain back-face elimination with diagram.
5. Compare object-space and image-space methods for HSE.

Key Formulas / Conditions

* Back-face test: $\vec{N} \cdot \vec{V} < 0 \rightarrow$ Visible; $\vec{N} \cdot \vec{V} > 0 \rightarrow$ Not visible
* Depth comparison (Z-buffer): If $Z_{new} < Z_{buffer} \rightarrow$ Update pixel; Else $\rightarrow$ No update

RGPV Notes Hub

Keep Learning, Keep Growing!

RGPV Notes Hub

IT601 – Computer Graphics & Multimedia

Unit – III

Basic Illumination Model

Topic – 18 Basic Illumination Model

1. Introduction

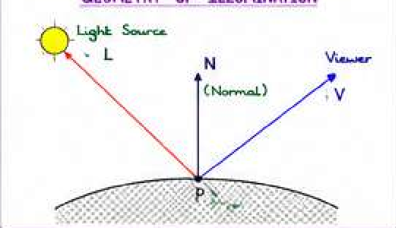
Illumination (or shading) is the process of determining the brightness (intensity or color) of a surface point as seen by an observer under light.

The basic illumination model explains how light interacts with a surface and how the final intensity is computed.

2. Components of Illumination

- Light Source (L) : Emits light.
- Object (Surface point P) : Receives light.
- Viewer (V) : Observes the illuminated object.
- Normal (N) : Unit vector perpendicular to the surface at P.
- Material Properties : Describe how the surface reflects light.

Geometry of Illumination



3. Basic Illumination Equation

The intensity I at a point P is computed as:

$$\boxed{I = I_a + I_d + I_s}$$

Where,

I_a = Ambient component (constant low level light)

I_d = Diffuse component (matte reflection)

I_s = Specular component (shiny reflection)

4. Components in Detail

(ii) Ambient Component (I_a)

Represents indirect light present in the environment. It is independent of direction.

$$\boxed{I_a = k_a I_a}$$

Where, k_a = ambient reflection coefficient

I_a = ambient light intensity

(ii) Diffuse Component (I_d)

Due to light reflected equally in all directions (Lambert's cosine law).

$$\boxed{I_d = k_d I_L \max(0, \vec{N} \cdot L)}$$

Where, k_d = diffuse reflection coefficient

I_L = intensity of light source

$\vec{N} \cdot L$ = cos θ (θ = angle between N and L)

(iii) Specular Component (I_s)

Due to mirror-like reflection in a particular direction (towards the viewer).

$$\boxed{I_s = k_s I_L \max(0, \vec{R} \cdot V)^n}$$

Where, k_s = specular reflection coefficient

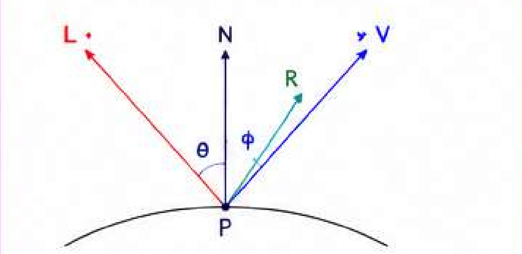
R = reflection vector

V = viewing (eye) vector

n = shininess (specular exponent)

5. Vectors Used

Vector	Description
L	Unit vector from point P to light source
N	Unit normal vector at point P
V	Unit vector from point P to viewer
R	Reflection of −L about N



Reflection vector R is given by:

$$\boxed{R = 2(\vec{N} \cdot L)N - L}$$

6. Complete Basic Illumination Model

$$\boxed{I = k_a I_a + k_d I_L \max(0, \vec{N} \cdot L) + k_s I_L \max(0, \vec{R} \cdot V)^n}$$

Where,

k_a, k_d, k_s = reflection coefficients

I_a = ambient light intensity

I_L = intensity of the light source

$\vec{N} \cdot L$ = cosine of angle between N and L

$\vec{R} \cdot V$ = cosine of angle between R and V

n = shininess exponent

7. Example

Given:

$k_a = 0.2$, $k_d = 0.6$, $k_s = 0.5$, $n = 20$

$I_a = 0.2$, $I_L = 1.0$

$\vec{N} \cdot L = \cos\theta = 0.707$

$\vec{R} \cdot V = \cos\phi = 0.866$

Calculate:

$I_a = 0.2 \times 0.2 = 0.04$

$I_d = 0.6 \times 1.0 \times 0.707 = 0.4242$

$I_s = 0.5 \times 1.0 \times (0.866)^{20} \approx 0.5 \times 0.0117 = 0.00585$

Total Intensity, $I = 0.04 + 0.4242 + 0.00585$

$= 0.47005$

8. Effect of Parameters

Parameter	Effect
k_a	Controls overall ambient brightness.
k_d	Higher value → brighter matte surface.
k_s	Higher value → brighter specular highlight.
n	Higher value → smaller, sharper highlight.
I_L	More light intensity → brighter surface.

9. Assumptions

- Light source is a point source.
- Surface is opaque and smooth locally.
- Reflection is computed at a single point.
- No inter-reflection between objects.

10. Advantages

- ✓ Simple and easy to implement.
- ✓ Produces realistic looking images.
- ✓ Works in real-time applications.
- ✓ Widely used in computer graphics pipeline.

11. Limitations

- ✗ Does not account for shadows.
- ✗ Assumes single light source.
- ✗ Not accurate for complex global illumination.
- ✗ Ignores reflection between objects.

12. Applications

- 3D modeling and rendering.
- CAD / CAM systems.
- Video games and animation.
- Virtual reality.
- Scientific visualization.

13. Summary

The basic illumination model computes the intensity at a surface point as the sum of ambient, diffuse and specular components. It depends on the position of light, viewer and surface normal and material properties. This model forms the foundation for more advanced illumination and rendering techniques.

Typical Exam Questions

1. Explain basic illumination model.
2. Derive the illumination equation.
3. What are ambient, diffuse and specular components?
4. Show the geometry of illumination with diagram.
5. What is the effect of shininess exponent (n)?
6. Write short notes on reflection vector.

Key Formulas

- $I = I_a + I_d + I_s$
- $I_a = k_a I_a$
- $I_d = k_d I_L \max(0, N \cdot L)$
- $I_s = k_s I_L \max(0, \vec{R} \cdot V)^n$
- $R = 2(N \cdot L)N - L$

Symbols

N – Normal vector

L – Light vector

V – Viewing vector

R – Reflection vector

I – Intensity

RGPV Notes Hub

Keep Learning, Keep Growing!

IT601 – Computer Graphics & Multimedia

Unit – III

Basic Illumination Model

Topic – 18
Basic Illumination Model

1. Introduction

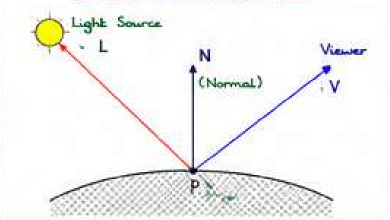
Illumination (or shading) is the process of determining the brightness (intensity or color) of a surface point as seen by an observer under light.

The basic illumination model explains how light interacts with a surface and how the final intensity is computed.

2. Components of Illumination

- Light Source (L) : Emits light.
- Object (Surface point P) : Receives light.
- Viewer (V) : Observes the illuminated object.
- Normal (N) : Unit vector perpendicular to the surface at P.
- Material Properties : Describe how the surface reflects light.

Geometry of Illumination



3. Basic Illumination Equation

The intensity I at a point P is computed as:

$$I = I_a + I_d + I_s$$

Where,
I_a = Ambient component (constant low level light)
I_d = Diffuse component (matte reflection)
I_s = Specular component (shiny reflection)

4. Components in Detail

(ii) Ambient Component (I_a)

Represents indirect light present in the environment. It is independent of direction.

$$I_a = k_a I_a$$

Where, k_a = ambient reflection coefficient
I_a = ambient light intensity

(ii) Diffuse Component (I_d)

Due to light reflected equally in alll directions (Lambert's cosine law).

$$I_d = k_d I_L \max(0, \vec{N}\cdot L)$$

Where, k_d = diffuse reflection coefficient
I_L = intensity of light source
$N\cdot L$ = cos θ (θ = angle between N and L)

(iii) Specular Component (I_s)

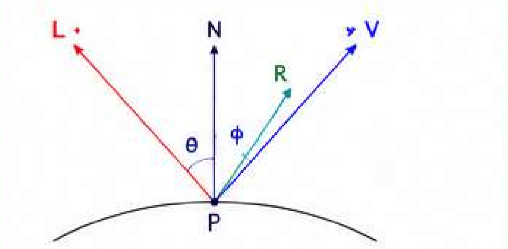
Due to mirror-like reflection in a particular direction (towards the viewer).

$$I_s = k_s I_L \max(0, \vec{R}\cdot V)^n$$

Where, k_s = specular reflection coefficient
R = reflection vector
V = viewing (eye) vector
n = shininess (specular exponent)

5. Vectors Used

Vector	Description
L	Unit vector from point P to light source
N	Unit normal vector at point P
V	Unit vector from point P to viewer
R	Reflection of −L about N



Reflection vector R is given by:

$$R = 2(\vec{N}\cdot L)N - L$$

6. Complete Basic Illumination Model

$$I = k_a I_a + k_d I_L \max(0, \vec{N}\cdot L) + k_s I_L \max(0, \vec{R}\cdot V)^n$$

Where,
k_a, k_d, k_s = reflection coefficients
I_a = ambient light intensity
I_L = intensity of the light source
$\vec{N}\cdot L$ = cosine of angle between N and L
$\vec{R}\cdot V$ = cosine of angle between R and V
n = shininess exponent

7. Example

Given:
$k_a = 0.2$, $k_d = 0.6$, $k_s = 0.5$, $n = 20$
$I_a = 0.2$, $I_L = 1.0$
$\vec{N}\cdot L = \cos\theta = 0.707$
$\vec{R}\cdot V = \cos\phi = 0.866$

Calculate:
$I_a = 0.2 \times 0.2 = 0.04$
$I_d = 0.6 \times 1.0 \times 0.707 = 0.4242$
$I_s = 0.5 \times 1.0 \times (0.866)^{20} \approx 0.5 \times 0.0117 = 0.00585$

Total Intensity, $I = 0.04 + 0.4242 + 0.00585 = 0.47005$

8. Effect of Parameters

Parameter	Effect
k_a	Controls overall ambient brightness.
k_d	Higher value → brighter matte surface.
k_s	Higher value → brighter specular highlight.
n	Higher value → smaller, sharper highlight.
I_L	More light intensity → brighter surface.

9. Assumptions

- Light source is a point source.
- Surface is opaque and smooth locally.
- Reflection is computed at a single point.
- No inter-reflection between objects.

10. Advantages

- ✓ Simple and easy to implement.
- ✓ Produces realistic looking images.
- ✓ Works in real-time applications.
- ✓ Widely used in computer graphics pipeline.

11. Limitations

- ✗ Does not account for shadows.
- ✗ Assumes single light source.
- ✗ Not accurate for complex global illumination.
- ✗ Ignores reflection between objects.

12. Applications

- 3D modeling and rendering.
- CAD/CAM systems.
- Video games and animation.
- Virtual reality.
- Scientific visualization.

13. Summary

The basic illumination model computes the intensity at a surface point as the sum of ambient, diffuse and specular components. It depends on the position of light, viewer and surface normal and material properties. This model forms the foundation for more advanced illumination and rendering techniques.

Typical Exam Questions

1. Explain basic illumination model.
2. Derive the illumination equation.
3. What are ambient, diffuse and specular components?
4. Show the geometry of illumination with diagram.
5. What is the effect of shininess exponent (n)?
6. Write short notes on reflection vector.

Key Formulas

- $I = I_a + I_d + I_s$
- $I_a = k_a I_a$
- $I_d = k_d I_L \max(0, N\cdot L)$
- $I_s = k_s I_L \max(0, \vec{R}\cdot V)^n$
- $R = 2(N\cdot L)N - L$

Symbols

N – Normal vector
L – Light vector
V – Viewing vector
R – Reflection vector
I – Intensity

Diffuse Reflection

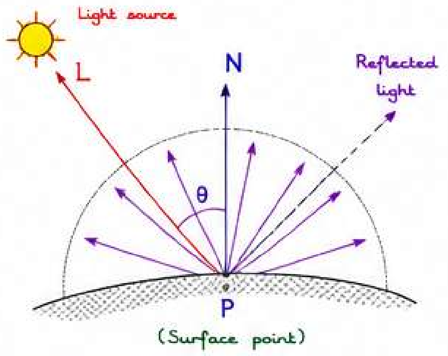
1. Introduction

Diffuse reflection (Lambertian reflection) occurs when light is reflected equally in all directions over the hemisphere above the surface. The brightness of a surface point depends on the angle between the light direction and the surface normal.

2. Ideal Diffuse Surface

- The surface appears equally bright from all viewing directions.
- Brightness depends only on the angle θ between light vector L and normal N).
- Also called Lambertian surface.

3. Geometry of Diffuse Reflection



For a perfect diffuse surface, reflected light is distributed uniformly over the hemisphere.

4. Lambert's Cosine Law

The intensity of diffuse reflection is proportional to the cosine of the angle θ between the light direction (L) and the surface normal (N).

$$I_d \propto \cos\theta$$
$$I_d = k_d I_L \cos\theta$$

Where,

I_d = diffuse reflected intensity
I_L = intensity of incident light
k_d = diffuse reflection coefficient ($0 \le k_d \le 1$)
θ = angle between L and N

5. Vector Form

Using dot product :

$$I_d = k_d I_L (N \cdot L)$$

Where,

$N \cdot L = \cos\theta = N_x L_x + N_y L_y + N_z L_z$

6. Diffuse Reflection Model

$$I_d = k_d I_L \max(0, N \cdot L)$$

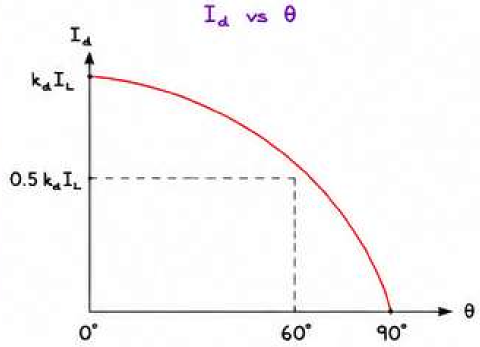
Explanation :

- If $(N \cdot L) > 0$ → light is on the front side of the surface → diffuse reflection occurs.
- If $(N \cdot L) \le 0$ → light is on the back side → no diffuse reflection.

7. Diagram – Variation with θ

θ (degrees)	cos θ	Diffuse Intensity I_d
0°	1.000	Maximum ($k_d I_L$)
30°	0.866	$0.866\ k_d I_L$
60°	0.500	$0.500\ k_d I_L$
90°	0.000	Zero
> 90°	< 0	Zero (no reflection)

I_d vs θ



8. Example

Given :

$I_L = 120$ cd, $k_d = 0.7$, $\theta = 60°$

Find diffuse reflected intensity I_d.

Solution :

$\cos 60° = 0.5$

$I_d = k_d I_L \cos\theta$
$= 0.7 \times 120 \times 0.5$
$= 42$ cd

Diffuse reflected intensity = 42 cd

9. Properties

- Independent of viewing direction.
- Depends only on the angle between light and surface normal.
- Energy is conserved (reflected over hemisphere).
- Produces matte (non-shiny) appearance.

10. Advantages

- ✓ Simple and easy to compute.
- ✓ Realistic for matte surfaces.
- ✓ Efficiency in real-time rendering.
- ✓ Forms the base for more complex illumination models.

11. Limitations

- ✗ Does not produce shiny highlights.
- ✗ Not suitable for specular or glossy surfaces.
- ✗ Assumes uniform reflection over hemisphere (ideal case).

12. Applications

- 3D computer graphics and visualization.
- CAD / CAM systems.
- Matte surface rendering.
- Scientific and medical visualization.
- Simulation and training systems.
- Animation and video games.

13. Summary

Diffuse reflection follows Lambert's cosine law, where the reflected intensity is proportional to the cosine of the angle between light direction and surface normal. It assumes that light is reflected equally in all directions over the hemisphere. This forms the basic component of the illumination model used in computer graphics.

Typical Exam Questions

1. Define diffuse reflection.
2. State Lambert's cosine law.
3. Derive the diffuse reflection equation.
4. Explain the role of surface normal in diffuse reflection.
5. What is Lambertian surface?
6. Compute diffuse intensity for given values of I_L, k_d and θ.
7. Draw the geometry of diffuse reflection.

Key Formulas

- Cosine of angle :
  $\cos\theta = N \cdot L$
- Lambert's cosine law :
  $I_d = k_d I_L \cos\theta$
- Diffuse reflection model :
  $I_d = k_d I_L \max(0, N \cdot L)$

Symbols Used

L → Light vector (from point to light source)
N → Surface normal vector
θ → Angle between L and N
k_d → Diffuse reflection coefficient
I_L → Incident light intensity
I_d → Diffuse reflected intensity

RGPV Notes Hub

IT601 - Computer Graphics & Multimedia

Unit - III

Specular Reflection

Topic - 18 Specular Reflection

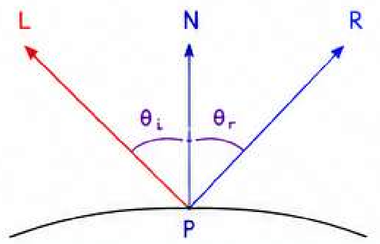
1. Introduction

Specular reflection (mirror like reflection) occurs at smooth surfaces where light is reflected in a particular direction. The intensity of specular reflection depends on the angle between the viewer direction and the perfect mirror reflection direction.

2. Ideal Specular Surface

* Surface is perfectly smooth.
* All reflected light travels in a single direction (mirror direction).

3. Geometry of Specular Reflection



Law of Reflection :
Angle of incidence (θ_i) = Angle of reflection (θ_r)
Both angles are measured from the normal (N).

4. Perfect Reflection Direction

The perfect (ideal) reflection direction R is obtained by reflecting the light vector L about the normal N.

$$\boxed{R = 2(N \cdot L)N - L}$$

Where,

N = unit surface normal at point P
L = unit vector from point P to light source
R = unit perfect reflection vector
(N · L) = dot product of N and L

5. Specular Reflection Model

The specular reflected intensity I_s is given by the Phong specular reflection model.

$$\boxed{I_s = k_s I_L \max(0, R \cdot V)^n}$$

Where,

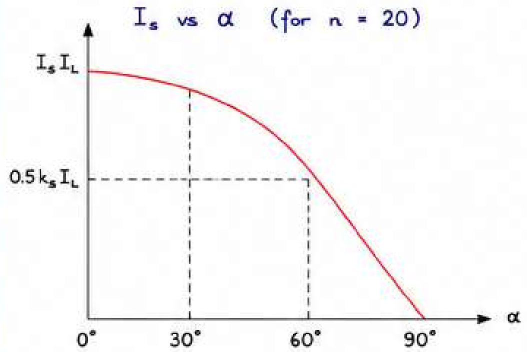
k_s = specular reflection coefficient $(0 \le k_s \le 1)$
I_L = intensity of the light source
R = perfect reflection vector (as above)
V = unit vector from point P to the viewer
n = shininess exponent $(n \ge 0)$
(R · V) = dot product of R and V

6. Explanation of Model

* R · V gives the cosine of the angle (α) between the perfect reflection direction R and the viewer direction V.
* If $(R \cdot V) \le 0$ → viewer is on the other side → no specular reflection.
* As (R · V) approaches 1 (i.e., $\alpha \to 0°$), intensity I_s is maximum.
* Shininess exponent n controls the size of the highlight. Higher n → smaller, sharper highlight.

7. Diagram - Variation with α

α (degrees)	cos α (R · V)	$I_s = k_s I_L (R \cdot V)^n$ (n=20)
0°	1.000	$k_s I_L$ (Maximum)
30°	0.866	$0.015\ k_s I_L$
60°	0.500	$9.54 \times 10^{-7}\ k_s I_L$
75°	0.259	$\sim 1.28 \times 10^{-13}\ k_s I_L$
90°	0.000	0 (No reflection)



8. Example

Given :

$k_s = 0.8$, $I_L = 100$ cd, $n = 20$
Angle between R and V, $\alpha = 30°$
Find specular reflected intensity I_s.

Solution :

$\cos 30° = 0.866$

$$\begin{aligned} I_s &= k_s I_L (\cos\alpha)^n \\ &= 0.8 \times 100 \times (0.866)^{20} \\ &= 80 \times 0.015 \\ &\approx 1.2 \text{ cd} \end{aligned}$$

Specular reflected intensity = 1.2 cd

9. Properties

* Reflection is mirror like.
* Depends on viewer position.
* Very sensitive to surface smoothness.
* Produces shiny highlights.
* Energy is conserved in the ideal case.

10. Advantages

✓ Produces realistic shiny surfaces.
✓ Used to simulate metals, polished objects, water, etc.
✓ Essential for photorealistic rendering.
✓ Controlled easily by parameters (k_s and n).

11. Limitations

✗ Phong model is an approximation.
✗ Does not account for multiple reflections.
✗ Not accurate for very complex optical phenomena.

12. Applications

* 3D computer graphics and visualization.
* CAD / CAM systems.
* Simulating metals, plastics, water.
* Animation and video games.
* Virtual reality and simulations.

13. Summary

Specular reflection occurs from smooth surfaces where light is reflected in a specific direction. The intensity of the specular component depends on the angle between the perfect reflection direction and the viewer direction. The Phong specular reflection model is widely used to compute specular highlights in computer graphics.

Typical Exam Questions

1. Define specular reflection.
2. Derive the perfect reflection vector formula.
3. Write the Phong specular reflection model.
4. Explain the effect of shininess exponent.
5. What is the difference between diffuse and specular reflection?
6. Draw the geometry of specular reflection with diagram.
7. Calculate specular reflected intensity for given values.

Key Formulas

* $R = 2(N \cdot L)N - L$
* $I_s = k_s I_L \max(0, R \cdot V)^n$
* $\cos\alpha = R \cdot V$
* $0 \le k_s \le 1$, $n \ge 0$

Symbols Used

L - Light vector (point to light source)
N - Surface normal (unit vector)
R - Perfect reflection vector
V - Viewing vector (to viewer)
θ_i - Angle of incidence
θ_r - Angle of reflection $(\theta_i = \theta_r)$
α - Angle between R and V
I_s - Specular reflected intensity
k_s - Specular reflection coefficient
n - Shininess exponent

RGPV Notes Hub

Keep Learning, Keep Growing!

RGPV Notes Hub

IT601 - Computer Graphics & Multimedia

Unit - III

Phong Shading

Topic - 18 Phong Shading

1. Introduction

Phong shading is a local illumination model that computes the color and intensity at each point (pixel) on a surface using the Phong reflection model. It provides smooth shading and realistic appearance of surfaces.

2. Phong Reflection Model

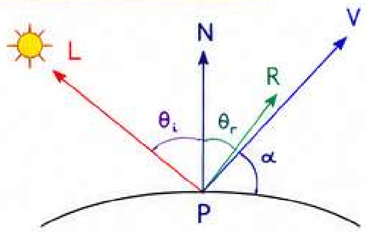
The model combines three components of illumination :

1. Ambient Reflection (I_a)
2. Diffuse Reflection (I_d)
3. Specular Reflection (I_s)

The total intensity is the sum of these components.

$$\boxed{I = I_a + I_d + I_s}$$

3. Geometry and Vectors



- L : Light vector (from P to light source)
- N : Surface normal at P (unit vector)
- V : View vector (from P to viewer/eye)
- R : Reflection vector (mirror direction)
- θ_i : Angle between L and N
- θ_r : Angle between R and N
- α : Angle between R and V

4. Components of Illumination

(i) Ambient Component (I_a)

- Represents indirect light present in the environment.

$$\boxed{I_a = k_a I_a}$$

where, k_a = ambient reflection coefficient
I_a = ambient light intensity

(ii) Diffuse Component (I_d)

- Due to Lambertian reflection.

$$\boxed{I_d = k_d I_L \max(0, N \cdot L)}$$

where, k_d = diffuse reflection coefficient
I_L = intensity of light source
$N \cdot L = \cos\theta_i$

(iii) Specular Component (I_s)

- Due to mirror like reflection.

$$\boxed{I_s = k_s I_L \max(0, \vec{R} \cdot V)^n}$$

where, k_s = specular reflection coefficient
$R \cdot V = \cos\alpha$
n = shininess exponent ($n \geq 0$)

5. Phong Shading Equation

Combining all the three components, the Phong shading equation is :

$$\boxed{\begin{aligned} I = {} & k_a I_a + k_d I_L \max(0, N \cdot L) \\ & + k_s I_L \max(0, \vec{R} \cdot V)^n \end{aligned}}$$

The value of I is clamped in the range

$0 \leq I \leq 1$
or
$0 \leq I \leq I_{max}$ (usually 1 or 255).

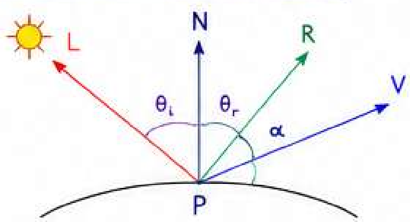
6. Steps to Compute Phong Intensity

1. Find surface normal N at point P.
2. Compute vector L from P to light source.
3. Compute viewing vector V from P to viewer.
4. Compute reflection vector R using :

$$\boxed{R = 2(N \cdot L)\, N - L}$$

5. Compute $\theta_i = \cos^{-1}(N \cdot L)$
6. Compute $\alpha = \cos^{-1}(R \cdot V)$
7. Compute I_a, I_d, I_s.
8. Compute total intensity I.

7. Diagram - Phong Reflection



8. Effect of Shininess Exponent (n)

n value	Highlight Size	Surface Appearance
Small (e.g., 1-10)	Large	Dull surface
Medium (e.g., 20-50)	Medium	Semi shiny
Large (e.g., > 100)	Very small	Very shiny surface

9. Example

Given : $k_a = 0.2$, $k_d = 0.6$, $k_s = 0.5$
$I_a = 0.2$, $I_L = 1.0$, $n = 20$
$N \cdot L = 0.707$, $R \cdot V = 0.866$

Find Phong intensity I.

Solution :

$I_a = k_a I_a = 0.2 \times 0.2 = 0.04$
$I_d = k_d I_L (N \cdot L) = 0.6 \times 1 \times 0.707 = 0.4242$
$I_s = k_s I_L (R \cdot V)^n = 0.5 \times 1 \times (0.866)^{20}$
$(0.866)^{20} = 0.0117$ (approx.)
$I_s = 0.5 \times 0.0117 = 0.00585$

$$\boxed{\begin{aligned} I &= I_a + I_d + I_s = 0.04 + 0.4242 + 0.00585 \\ I &= 0.47005 \text{ (approx.)} \end{aligned}}$$

10. Properties

- Produces smooth shading.
- Models both matte and shiny surfaces.
- More realistic than flat or Gouraud shading.
- Depends on surface normal, light and viewer positions.
- Very sensitive to shininess exponent.

11. Advantages

- ✓ Gives realistic lighting effects.
- ✓ Produces smooth highlight.
- ✓ Handles a wide range of materials.
- ✓ Easily controllable using parameters (k_a, k_d, k_s, n).

12. Limitations

- ✗ Computationally expensive.
- ✗ Requires normal at each point.
- ✗ Assumes single point light source.
- ✗ Does not handle shadows.

13. Applications

- Realistic 3D rendering and animation.
- CAD/CAM systems.
- Video games and virtual reality.
- Scientific visualization.
- Simulations requiring realistic lighting.

14. Summary

Phong shading is a local illumination model based on the Phong reflection model which combines ambient, diffuse and specular components of light. The intensity at each point is computed using surface normal, light direction and viewer direction. It produces smooth and realistic shading and is widely used in computer graphics to simulate real world lighting.

Typical Exam Questions

1. Define Phong shading.
2. Derive the Phong shading equation.
3. Explain the components of Phong reflection model.
4. Explain the role of shininess exponent.
5. Draw the geometry of Phong shading.
6. Calculate Phong intensity for given values.
7. Compare Phong and Gouraud shading.

Key Formulas

- $R = 2(N \cdot L)N - L$
- $I = k_a I_a + k_d I_L \max(0, N \cdot L) + k_s I_L \max(0, \vec{R} \cdot V)^n$
- $0 \leq I \leq 1$
- $N \cdot L = \cos\theta_i$, $R \cdot V = \cos\alpha$

Symbols Used

- L - Light vector
- N - Surface normal
- V - View vector
- R - Reflection vector
- I_a - Ambient intensity
- I_L - Light intensity
- k_a, k_d, k_s - Coefficients
- n - Shininess exponent

RGPV Notes Hub

Keep Learning , Keep Growing !

RGPV Notes Hub

IT601 - Computer Graphics & Multimedia

Unit - III

Gouraud Shading

Topic - 18
Gouraud Shading

1. Introduction

Gouraud shading is an intensity interpolation method for shaded rendering of polygons. Intensities are calculated at the vertices (using a lighting model) and then interpolated across the surface. It produces smooth shading and is computationally efficient.

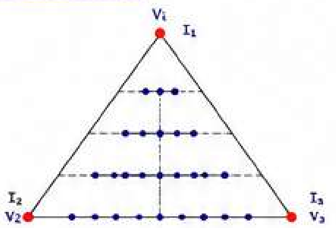
2. Basic Idea

* Compute illumination (intensity) at each vertex of the polygon.
* Interpolate these intensities along the edges.
* Interpolate again across each scan line (or span) to obtain intensity of interior pixels.
* Display the pixel using the interpolated intensity.

3. Algorithm (Overview)

1. Compute vertex normals (if not given).
2. Compute intensity I_L at each vertex using illumination model (e.g. Phong).
3. For each edge, interpolate intensity between the two end vertices.
4. For each scan line, interpolate intensity between the left and right edge intensities.
5. Use interpolated intensity to color the pixels.

4. Illustration



I_1, I_2, I_3 - Intensities at vertices
• - Pixels on scan lines

5. Intensity Interpolation

(i) Along an edge (linear interpolation)

For edge from vertex A (I_A) to B (I_B):

$$I(t) = I_A + (I_B - I_A)\,t \qquad 0 \le t \le 1$$

where t increases from 0 at A to 1 at B.

(ii) Along a scan line

If I_L and I_R are intensities at the left and right intersections of the scan line:

$$I(x) = I_L + (I_R - I_L)\,\frac{x - x_L}{x_R - x_L}$$

where x_L, x_R are left and right x-coordinates.

6. Steps in Detail

1. For each vertex, compute normal N.
2. Evaluate intensity I using illumination model.
3. Sort polygon vertices by y-coordinate.
3. Sort polygon edges by y-coordinate.
4. Process the polygon between successive scan lines.
5. For each scan line:
   * Find intersection points with polygon edges.
   * For each intersection, compute intensity by interpolating along the edge.
   * Pair left and right intersections.
   * Interpolate intensity between these two points and shade the span.
6. Move to next scan line.

7. Example

Triangle with vertex intensities:

$I_1 = 0.9$ (top vertex)
$I_2 = 0.2$ (left vertex)
$I_3 = 0.6$ (right vertex)

Along edge V1-V2, at $t = 0.5$:

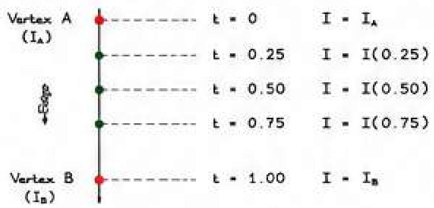
$I = 0.9 + (0.2 - 0.9) \times 0.5 = 0.55$

On a scan line where left intensity $I_L = 0.3$ and right intensity $I_R = 0.7$, for a pixel with x exactly in the middle:

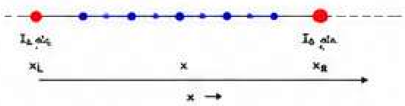
$I = 0.3 + (0.7 - 0.3) \times 0.5 = 0.5$

8. Diagram - Intensity Interpolation

(i) Edge interpolation



(ii) Scan line interpolation



Intensity at any x between x_L and x_R is interpolated linearly.

9. Comparison with Phong Shading

Feature	Gouraud Shading	Phong Shading
Where intensity is computed	At vertices	At every pixel
Interpolation	Yes (linear)	No
Smoothness	Good for most surfaces	Better (especially for small specular highlights)
Cost	Less	More
Best suited for	Large polygons, real-time	Small polygons, high quality

10. Properties

* Produces smooth intensity variation.
* Computationally faster than Phong shading.
* Intensity changes linearly across a polygon.
* Works well when polygons are small.
* Specular highlights may be less accurate (because computed only at vertices).

11. Advantages

✓ Efficient and simple.
✓ Good visual quality with fewer calculations.
✓ Suitable for real-time applications.
✓ Requires intensity calculation only at vertices.

12. Limitations

✗ Less accurate specular highlights.
✗ May show Mach band effect (for large polygons).
✗ Not suitable for very high gloss surfaces.

13. Applications

* Real-time 3D graphics.
* CAD / CAM systems.
* Video games.
* Simulation and training systems.
* Any application requiring smooth shading with good performance.

14. Summary

Gouraud shading determines the intensity at the vertices of a polygon using a lighting model (such as Phong reflection model). These vertex intensities are then linearly interpolated along the edges and across scan lines to compute the intensity of interior pixels. It produces smooth shaded images with low computational cost and is widely used in real-time rendering systems.

Typical Exam Questions

1. Define Gouraud shading.
2. Explain the steps of Gouraud shading algorithm.
3. Derive the formula for intensity interpolation along an edge.
4. Derive the formula for scan line interpolation.
5. Compare Gouraud shading with Phong shading.
6. List advantages and limitations of Gouraud shading.
7. Draw diagram to explain Gouraud shading.

Key Formulas

* Edge interpolation:
  $I(t) = I_A + (I_B - I_A)t, \quad 0 \le t \le 1$
* Scan line interpolation:
  $I(x) = I_L + (I_R - I_L)\,\frac{x - x_L}{x_R - x_L}$
* Reflection vector: $R = 2(N \cdot L)N - L$
* Phong model:
  $I = k_a I_A + k_d I_L \max(0, N \cdot L) + k_s I_L \max(0, R \cdot V)^n$

Symbols

L - Light vector
N - Surface normal
V - View vector
R - Reflection vector
I - Intensity
I_A - Ambient component
I_L - Light intensity
k_a, k_d, k_s - Coefficients
n - Shininess exponent

RGPV Notes Hub

Keep Learning, Keep Growing!

Ray Tracing

1. Introduction

Ray tracing is a realistic rendering technique that simulates the behavior of light rays in a scene. It can produce very realistic images with shadows, reflections, and refractions. It is computationally intensive but very accurate.

2. Basic Idea

* A ray is sent from the eye (through each pixel) into the scene.
* The ray is traced as it intersects objects.
* Shading is computed at the first visible surface hit by the ray.
* Secondary rays are generated for reflection, refraction and shadows.

3. Ray Equation

A ray is defined by an origin P_0 and a direction d.

$$\boxed{R(t) = P_0 + td, \quad t \geq 0}$$

where,
- P_0 - origin of the ray
- d - direction vector (unit vector)
- t - parameter (distance along the ray)

4. Typical Ray Tracing Process

For each pixel generate a primary ray
↓
Find intersection of ray with objects in the scene
↓
Shade the intersection point
↓
Generate secondary rays (reflection / refraction / shadow)
↓
Combine all contributions and compute final color

5. Intersection Tests

∵ Ray must be tested for intersection with all objects.

For a sphere:

Given sphere center C and radius r, solve $\|R(t) - C\|^2 = r^2$

This gives a quadratic equation in t.

Choose the smallest positive t.

For a plane:

Given point P_p on plane and normal n,

$$t = \frac{n \cdot (P_p - P_0)}{n \cdot d}$$

(if $n \cdot d = 0 \rightarrow$ no intersection)

6. Shading Model

At the intersection point P with normal N, the color is computed using an illumination model (e.g., Phong model).

$$I = k_a I_a + k_d I_d + k_s I_s$$

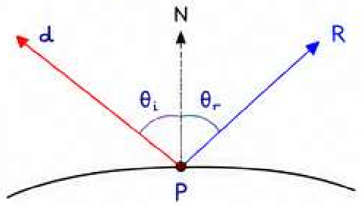
7. Secondary Rays

(i) Reflection Ray

A reflection ray is generated in the direction R:

$$\boxed{R = d - 2(d \cdot N)N}$$

Where d - incident ray direction
N - surface normal (unit vector)



(ii) Refraction Ray

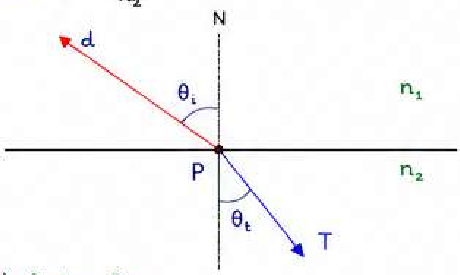
A refraction (transmitted) ray is generated according to Snell's law.

$$\boxed{n_1 \sin\theta_i = n_2 \sin\theta_t}$$

Direction T is given by:

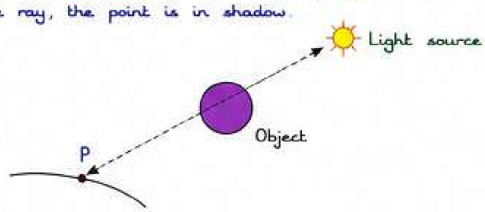
$$\boxed{T = \eta d + (\eta \cos\theta_i - \cos\theta_t) N}$$

where $\eta = \dfrac{n_1}{n_2}$



(iii) Shadow Ray

A shadow ray is sent from the intersection point P toward the light source. If any object blocks the ray, the point is in shadow.



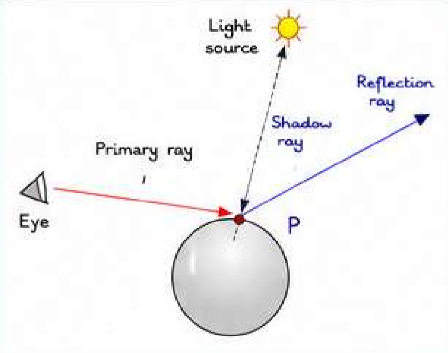
8. Ray Tracing Algorithm (Recursive)

1. Cast primary ray from eye through pixel.
2. Find nearest intersection.
3. Compute local illumination.
4. If the material is reflective, cast reflection ray.
5. If the material is transparent, cast refraction ray.
6. Trace shadow ray toward light sources.
7. Combine all contributions.
8. Return color.

9. Simple Example

Consider a ray from eye hitting a shiny sphere.

- Primary ray hits the sphere.
- Local illumination is computed.
- Reflection ray is generated and traced (may hit another object or background).
- Shadow ray checks if the point is visible to the light source.



10. Advantages

✓ Produces very realistic images.
✓ Handles global illumination effects (shadows, reflections, refractions).
✓ Easy to implement physically (based on real-world optics).

11. Limitations

✗ Computationally very expensive.
✗ High memory requirement.
✗ Slow for complex scenes.
✗ Not suitable for real-time rendering (without optimization).

12. Applications

- Photorealistic image synthesis.
- Movie and animation rendering.
- Architectural visualization.
- Engineering design and simulation.
- Scientific visualization.

13. Summary

Ray tracing simulates the path of light rays from the eye into the scene. It finds the color of a pixel by tracing rays, computing illumination, and considering reflection, refraction and shadows. Although computationally expensive, it produces highly realistic images and is widely used in offline rendering applications.

Typical Exam Questions

1. Define ray tracing.
2. Write and explain ray equation.
3. Explain the ray tracing algorithm.
4. Derive ray-sphere intersection.
5. Derive ray-plane intersection.
6. Write the reflection and refraction equations.
7. Explain shadow ray.
8. List advantages and limitations.

Key Formulas

- Ray equation: $R(t) = P_0 + td$
- Reflection: $R = d - 2(d \cdot N)N$
- Snell's law: $n_1 \sin\theta_i = n_2 \sin\theta_t$
- Refraction: $T = \eta d + (\eta\cos\theta_i - \cos\theta_t)N$
- Illumination: $I = k_a I_a + k_d I_d + k_s I_s$

Symbols Used

- P_0 - Ray origin (eye)
- d - Ray direction
- N - Surface normal
- n_1, n_2 - Refractive indices
- R - Reflection ray
- T - Refraction ray

RGPV

Notes Hub

IT601 - Computer Graphics & Multimedia

Unit - III

RGB Colour Model

Topic - 18
RGB Colour Model

1. Introduction

The RGB colour model is an additive colour model in which colours are created by adding Red (R), Green (G) and Blue (B) light. It is the most commonly used colour model in computer displays, TVs, cameras and projectors.

2. Basic Idea

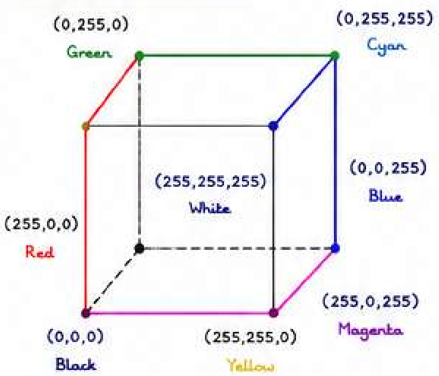
- Each primary colour (R, G, B) can have intensity values from 0 to 255.
- When all three are 0, we get Black.
- When all three are 255, we get White.
- By combining different intensities, we get a wide range of colours.

3. Additive Colour Model

In additive models, colours are created by adding light. More light gives brighter colours.

Resultant Colour = R + G + B

4. RGB Colour Cube



Every point inside the cube represents a colour with a unique combination of R, G and B intensities.

5. Primary Colours

These are the three basic colours in RGB model.

- Red - (255, 0, 0)
- Green - (0, 255, 0)
- Blue - (0, 0, 255)

Typical Exam Questions

1. Define RGB colour model.
2. Explain additive colour model.
3. Draw and explain RGB colour cube.
4. Write the RGB values for (i) White (ii) Black (iii) Cyan (iv) Yellow.
5. How many colours can be represented in 24-bit RGB model?
6. Compare RGB and CMY colour models.
7. What is the significance of 0 - 255 range in RGB model?

6. Secondary Colours

By adding any two primary colours in equal intensity, we get secondary colours.

- Yellow = Red + Green = (255,255,0)
- Cyan = Green + Blue = (0,255,255)
- Magenta = Red + Blue = (255,0,255)

7. Intensity Values

In digital systems, each primary colour has 8 bits (1 byte) → 2^8 = 256 levels (0-255).

A colour is represented as:

(R, G, B) where 0 ≤ R, G, B ≤ 255

8. Common RGB Values

Colour	R	G	B
Black	0	0	0
White	255	255	255
Red	255	0	0
Green	0	255	0
Blue	0	0	255
Yellow	255	255	0
Cyan	0	255	255
Magenta	255	0	255
Gray (x)	x	x	x

9. Colour Combination Examples

(255,0,0) + (0,255,0) = (255,255,0)

(0,255,0) + (0,0,255) = (0,255,255)

(255,0,0) + (0,0,255) = (255,0,255)

(255,0,0) + (0,255,0) + (0,0,255) = (255,255,255)

(0,0,0) + (0,0,0) + (0,0,0) = (0,0,0)

Key Points

- RGB is an additive colour model.
- Used in monitors, TVs, mobile screens, projectors, cameras.
- Each colour is represented by three values (R, G, B).
- Each value ranges from 0 to 255 (8 bits).
- Total number of colours = 256 × 256 × 256 = 16,777,216 (≈ 16.7 million colours).

10. Number of Colours

If each of R, G, B has 256 possible values, then total number of colours that can be represented is:

256 × 256 × 256 = 16,777,216
(≈ 16.7 million colours)

11. RGB vs CMY

Feature	RGB Model	CMY Model
Type	Additive	Subtractive
Primaries	Red, Green, Blue	Cyan, Magenta, Yellow
Used in	Monitors, TVs, Projectors	Printers, Inks, Paints
Black	(0,0,0)	(0,0,0) (Ideally) but uses K in CMYK
White	(255,255,255)	(255,255,255) (Ideally)
Colour Generation	By adding light	By subtracting light

12. Advantages

- ✓ Simple and easy to implement.
- ✓ Wide range of colours.
- ✓ Matches human visual perception.
- ✓ Widely supported by hardware.

13. Limitations

- ✗ Device dependent (different monitors may display colours differently).
- ✗ Not suitable for printing (uses CMY/CMYK).
- ✗ Higher bit depth increases memory requirement.

14. Applications

- Display devices (monitors, TVs, mobiles).
- Digital cameras and scanners.
- Multimedia and animation.
- Image processing and editing software.
- Computer graphics and visualization.

RGPV Notes Hub

Keep Learning, Keep Growing!

RGPV

Notes Hub

IT601 – Computer Graphics & Multimedia

Unit – III

YIQ Colour Model

Topic – 18
YIQ Colour Model

1. Introduction

The YIQ colour model is a colour space used in analogue colour TV systems, especially in NTSC (USA) television. It separates the luminance from the chrominance (colour) components.

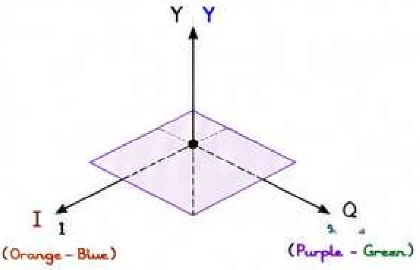
2. Basic Idea

* Y represents the luminance (brightness) component.
* I and Q represent the chrominance (colour) components.
* This separation allows backward compatibility with black-and-white TVs (which use only Y).
* It reduces the bandwidth required for transmitting colour signals.

3. Components

Component	Meaning
Y	Luminance (brightness) Information
I	In-phase chrominance (Orange – Blue)
Q	Quadrature chrominance (Purple – Green)

4. YIQ Colour Space



W – Y axis represents brightness.
I and Q axes represent colour information.

5. YIQ from RGB

The transformation from RGB to YIQ is :

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Where R, G, B are in the range [0, 1] or [0, 255].

Inverse transformation (YIQ to RGB) :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{pmatrix} \begin{pmatrix} Y \\ I \\ Q \end{pmatrix}$$

6. Range of Components

* Y (Luminance) : 0 to 1 (or 0 to 255)
* I (In-phase) : -0.5957 to +0.5957
* Q (Quadrature) : -0.5226 to +0.5226

Note: I and Q can be negative.

7. Properties

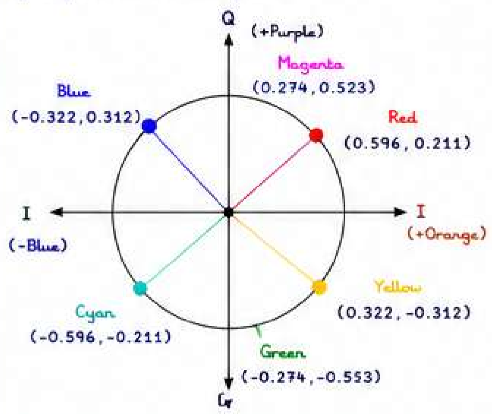
✓ Y carries most of the visual information.
✓ Human eye is more sensitive to brightness than colour.
✓ By transmitting Y with full bandwidth and I, Q with reduced bandwidth, we save transmission bandwidth.

8. Colour Representation (Examples)

Colour	R	G	B	Y	I	Q
White	1	1	1	1.000	0.000	0.000
Black	0	0	0	0.000	0.000	0.000
Red	1	0	0	0.299	0.596	0.211
Green	0	1	0	0.587	-0.274	-0.523
Blue	0	0	1	0.114	-0.322	0.312
Yellow	1	1	0	0.886	0.322	-0.312
Cyan	0	1	1	0.701	-0.596	-0.211
Magenta	1	0	1	0.413	0.274	0.523

9. Chrominance Plane (I-Q Plane)

At a constant luminance (Y), colour is represented by a point on the I-Q plane.



Saturation increases as the point moves away from the center (gray axis). Center represents achromatic colours.

Key Points

* YIQ = Luminance (Y) + Chrominance (I, Q).
* Y is compatible with black and white TV.
* Used in NTSC analogue colour TV system.
* Separates brightness from colour.
* Helps reduce bandwidth requirements.
* Conversion between RGB and YIQ uses simple linear transformations.

10. Why YIQ is used?

✓ Human eye is more sensitive to brightness (Y).
✓ Colour information (I and Q) can be transmitted with less bandwidth.
✓ Black & white TVs can display the Y signal alone.
✓ Improves efficiency in analogue TV transmission (NTSC).

11. YIQ vs RGB

Feature	YIQ Model	RGB Model
Type	Luminance-Chrominance Model	Additive Primary Colour Model
Components	Y (brightness), I (in-phase), Q (quadrature)	R (Red), G (Green), B (Blue)
Purpose	TV transmission, Bandwidth saving	Display devices, Graphics
Compatibility	Compatible with Black & White TV	Not compatible with B/W TV
Used in	NTSC TV system	Computer graphics, Monitors, Cameras

12. Advantages

✓ Efficient bandwidth utilization.
✓ Backward compatible with B/W systems.
✓ Separates brightness from colour.
✓ Good for analogue transmission.

13. Limitations

✗ Not used in modern digital systems (uses RGB or YCbCr).
✗ Negative values for I and Q are not intuitive.
✗ Conversion slightly reduces precision.

14. Applications

* Analogue colour television (NTSC).
* Video signal transmission.
* Broadcast systems.
* Early colour image processing.

15. Summary

The YIQ colour model separates the brightness (Y) from the colour information (I and Q). Y carries most of the visual information, while I and Q represent colour differences. This model is widely used in NTSC television systems to reduce bandwidth and maintain compatibility with black and white receivers.

Typical Exam Questions

1. Define YIQ colour model.
2. Why is YIQ model used in television systems?
3. Derive the transformation matrix from RGB to YIQ.
4. What does Y component represent?
5. What are I and Q components?
6. Explain the significance of YIQ model.
7. Compare YIQ and RGB colour models.
8. Write the inverse transformation from YIQ to RGB.

Key Formulas

* RGB to YIQ : $YIQ = M_{YIQ} \times RGB$

Where,

$$M_{YIQ} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix}$$

* YIQ to RGB : $RGB = M_{YIQ}^{-1} \times YIQ$

Symbols Used

R, G, B : Red, Green, Blue (primary colours)

Y : Luminance component

I : In-phase chrominance

Q : Quadrature chrominance

RGPV

Notes Hub

IT601 – Computer Graphics & Multimedia

Unit – III

CMY Colour Model

Topic – 18

CMY Colour Model

1. Introduction

The CMY colour model is a subtractive colour model used in printing and reflective media. It is the counterpart of the RGB model used in digital displays. CMY uses three primary colours: Cyan, Magenta and Yellow. By subtracting these from white, we produce a wide range of colours.

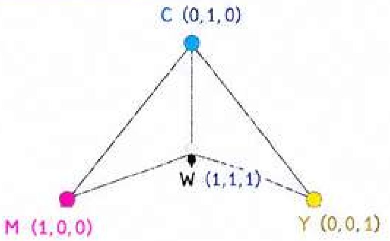
2. Basic Idea

- Cyan absorbs Red light.
- Magenta absorbs Green light.
- Yellow absorbs Blue light.
- Combining the three in different amounts absorbs more light and produces darker colours.
- White is obtained when no ink is present (no light is absorbed).

3. Components

Component	Meaning
C (Cyan)	Absorbs Red light
M (Magenta)	Absorbs Green light
Y (Yellow)	Absorbs Blue light

4. CMY Colour Space



The CMY space is a subtractive system.
The more ink (C, M, Y), the darker the colour.
C + M + Y = Black (K) (in practice)

5. CMY from RGB

CMY values are the complements of RGB values (normalized in range 0 to 1):

$$C = 1 - R$$
$$M = 1 - G$$
$$Y = 1 - B$$

Where $R, G, B \in [0,1]$

6. CMY to RGB (Reverse Conversion)

$$R = 1 - C$$
$$G = 1 - M$$
$$B = 1 - Y$$

7. Range

- $C, M, Y \in [0,1]$
- 0 = No ink   1 = Full ink (maximum absorption)

8. Colour Representation (Examples)

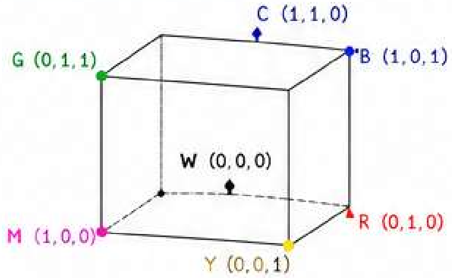
C	M	Y	Colour
0	0	0	White
1	1	1	Black
1	0	0	Cyan
0	1	0	Magenta
0	0	1	Yellow
1	1	0	Blue (C + M)
0	1	1	Green (M + Y)
1	0	1	Red (C + Y)

9. CMY Combination

Any colour can be produced by combining appropriate amounts of C, M and Y.

Colour = (C × Cyan) + (M × Magenta) + (Y × Yellow)
(Absorbs corresponding light)

10. CMY Colour Cube



Opposite corners represent complementary colours.
C + M + Y = Black (K)

11. Advantages

- ✓ Simple and intuitive for printing.
- ✓ Widely used in printers and copiers.
- ✓ Good for understanding subtractive mixing.

12. Limitations

- ✗ Not suitable for displays (they use RGB).
- ✗ Less efficient for representing bright colours.
- ✗ Needs more ink for dark colours.

13. Applications

- Printing and publishing
- Colour printing in magazines, posters
- Commercial printing and packaging
- Design and graphics in print media

14. CMY vs RGB

Feature	CMY Model	RGB Model
Type	Subtractive	Additive
Medium	Reflective (Print)	Emissive (Display)
Primary Colours	Cyan, Magenta, Yellow	Red, Green, Blue
White	No ink (C=0, M=0, Y=0)	All On (R=1, G=1, B=1)
Black	C=1, M=1, Y=1 (Absorbs all)	R=0, G=0, B=0 (All Off)
Applications	Printing, Paper Media	Screens, Monitors, TVs

15. Summary

- CMY model is used in printing (subtractive).
- Uses Cyan, Magenta and Yellow inks.
- CMY values are complements of RGB.
- White = No ink, Black = Full ink of all.
- Widely used in printing and publishing.

16. Limitations

- ✗ Disadvantageous for displays (not additive).
- ✗ Colour gamut is limited compared to RGB.
- ✗ Darker colours need more ink.

17. Key Formulas

- $C = 1 - R$
- $M = 1 - G$
- $Y = 1 - B$
- $R = 1 - C$
- $G = 1 - M$
- $B = 1 - Y$

18. Applications (Summary)

- Printing and publishing
- Colour reproduction in documents
- Packaging and product labels
- Brochures, posters and advertisements

Key Point

CMY subtracts colours from white to produce other colours. It is the foundation of subtractive colour printing.

Typical Exam Questions

1. Define CMY colour model.
2. Derive CMY from RGB.
3. Explain CMY to RGB conversion.
4. Draw CMY colour cube and explain.
5. List applications of CMY model.

Key Formulas

- $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$
- $R = 1 - C$, $G = 1 - M$, $B = 1 - Y$
- $C, M, Y \in [0,1]$

Symbols Used

C – Cyan, M – Magenta
Y – Yellow, R – Red
G – Green, B – Blue
W – White, K – Black

RGPV Notes Hub

Keep Learning, Keep Growing!

HSV Colour Model

1. Introduction

The HSV colour model (Hue, Saturation, Value) represents colours in a cylindrical coordinate system which corresponds closely to the way humans perceive colour. It is widely used in image editing, computer graphics and vision applications.

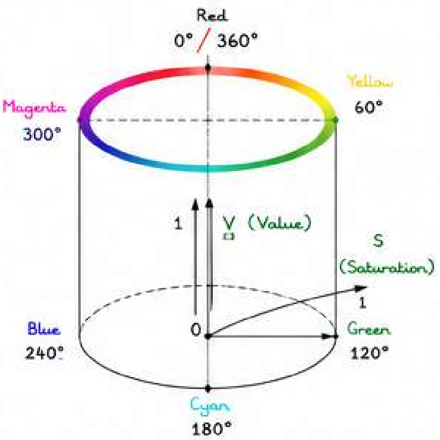
2. Basic Idea

- Hue (H) represents the type of colour (Red, Green, Blue, etc.)
- Saturation (S) represents the purity of the colour.
- Value (V) represents the brightness of the colour.
- HSV separates chromatic content (H,S) from intensity (V).

3. Components

Component	Symbol	Meaning	Range
Hue	H	Type of colour	0° - 360°
Saturation	S	Purity of colour	0 - 1 (0% - 100%)
Value	V	Brightness (intensity)	0 - 1 (0% - 100%)

4. HSV Colour Space



- H : Angle around the cylinder (0° - 360°)
- S : Radius from center (0 at center → gray, 1 at edge → full colour)
- V : Height from bottom to top (0 → black, 1 → full brightness)

5. Achromatic Colours

- When S = 0, colour is a shade of gray.
- When V = 0, colour is black (regardless of H and S).
- When V = 1 and S = 1, we get the pure (bright) colours.

6. Hue Values (Common)

Hue (°)	Colour
0°	Red
60°	Yellow
120°	Green
180°	Cyan
240°	Blue
300°	Magenta
360°	Red (Again)

7. HSV to RGB Conversion

Given H (0°-360°), S (0 - 1), V (0 - 1).

Step 1 : Convert H to sector

$$C = V \times S$$
$$X = C \times (1 - |(H/60) \bmod 2) - 1|)$$
$$m = V - C$$

Step 2 : Determine (R′, G′, B′) based on H sector

H Range (°)	R′	G′	B′
0 - 60	C	X	0
60 - 120	X	C	0
120 - 180	0	C	X
180 - 240	0	X	C
240 - 300	X	0	C
300 - 360	C	0	X

Step 3 : Add m to each component

$$R = R' + m$$
$$G = G' + m$$
$$B = B' + m$$

(R, G, B in range 0 - 1)

8. RGB to HSV Conversion

Given R, G, B in range [0, 1]

Step 1 : Find

$$C_{max} = \max(R, G, B)$$
$$C_{min} = \min(R, G, B)$$
$$\Delta = C_{max} - C_{min}$$

Step 2 : Calculate Value

$$V = C_{max}$$

Step 3 : Calculate Saturation

$$S = \begin{cases} 0 & \text{if } C_{max} = 0 \\ \Delta / C_{max} & \text{otherwise} \end{cases}$$

Step 4 : Calculate Hue

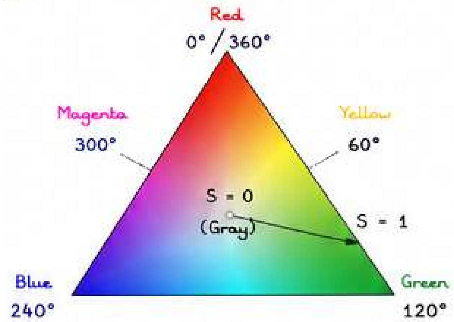
$$H = \begin{cases} 0 & \text{if } \Delta = 0 \\ 60 \times ((G - B)/\Delta \bmod 6 & \text{if } C_{max} = R \\ 60 \times ((B - R)/\Delta + 2) & \text{if } C_{max} = G \\ 60 \times ((R - G)/\Delta + 4) & \text{if } C_{max} = B \end{cases}$$

(Ensure H is in 0° - 360°)

9. HSV Triangle (at constant V)

At a fixed value (V), HSV forms a triangle.

Center (S = 0) → Gray of that value
Edges (S = 1) → Pure colours



10. Examples

H (°)	S	V	Perceived Colour
0	1	1	Bright Red
60	1	1	Bright Yellow
120	1	1	Bright Green
180	1	1	Bright Cyan
240	1	1	Bright Blue
300	1	1	Bright Magenta
Any	0	V	Gray (depends on V)
Any	S	0	Black

11. Advantages

- ✓ More intuitive and closer to human perception.
- ✓ Easy to select and modify colours.
- ✓ Separates intensity (V) from colour (H, S).
- ✓ Widely used in image processing, computer graphics, and UI design.

12. Limitations

- ✗ Not linear with respect to human perception.
- ✗ Hue becomes unstable when Saturation is very low (near gray).
- ✗ Conversion between HSV and RGB involves complex calculations.

13. Applications

- Image editing (brightness, saturation controls)
- Colour based segmentation
- Computer graphics and visualization
- User interface colour pickers
- Lighting and shading models

14. Summary

HSV model represents colour using Hue (H), Saturation (S) and Value (V).
It is a cylindrical model where:
- H controls the type of colour,
- S controls the purity,
- V controls the brightness.

HSV is simple, intuitive and widely used in graphics and vision applications.

Typical Exam Questions

1. Define HSV colour model.
2. Draw and explain HSV colour space.
3. Explain the components of HSV model.
4. Derive HSV to RGB conversion.
5. Derive RGB to HSV conversion.
6. Compare HSV and RGB colour models.
7. List applications of HSV model.

Key Formulas

- $C = V \times S$
- $X = C \times (1 - |(H/60) \bmod 2) - 1|)$
- $m = V - C$
- $R = R' + m$, $G = G' + m$, $B = B' + m$
- $V = C_{max}$
- $S = 0$ if $C_{max} = 0$ else Δ / C_{max}

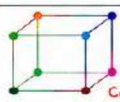
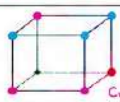
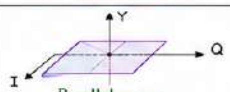
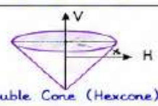
Symbols Used

- H : Hue (0° - 360°)
- S : Saturation (0 - 1)
- V : Value (0 - 1)
- R, G, B : Red, Green, Blue
- C_{max} : Maximum of R, G, B
- C_{min} : Minimum of R, G, B
- Δ : $C_{max} - C_{min}$

RGPV Notes Hub

Keep Learning , Keep Growing !

COMPARISON OF COLOUR MODELS

Feature	RGB Model	CMY Model	YIQ Model	HSV Model	Remarks
Full Form	Red, Green, Blue	Cyan, Magenta, Yellow	Luminance (Y), In-phase (I), Quadrature (Q)	Hue, Saturation, Value	Different approaches to represent colour.
Type	Additive	Subtractive	Luminance - Chrominance	Intuitive / Perceptual (derived from RGB)	Additive models are used in emissive devices; subtractive in reflective media.
Purpose / Use	Display devices (Monitors, TVs, Projectors, Mobile)	Printing, Ink, Paints, Reflexive media	Analog TV transmission (NTSC System)	Image editing, Computer graphics, UI design	YIQ separates brightness from colour for efficient transmission.
Components	R (Red), G (Green), B (Blue)	C (Cyan), M (Magenta), Y (Yellow)	Y (Luminance), I (In-phase), Q (Quadrature)	H (Hue), S (Saturation), V (Value)	Each model uses 3 components.
Component Range	$R, G, B \in [0, 1]$ or $[0, 255]$	$C, M, Y \in [0, 1]$	$Y \in [0, 1]$ $I \in [-0.5957, 0.5957]$ $Q \in [-0.5226, 0.5226]$	$H \in [0°, 360°]$ $S \in [0, 1]$ $V \in [0, 1]$	Ranges may be normalized (0-1) or in 8-bit (0-255).
Basic Idea	Add Red, Green, Blue light in different amounts to get a colour.	Subtract Cyan, Magenta, Yellow inks from white light to get a colour.	Y carries brightness; I and Q carry colour information.	Hue gives the colour type, Saturation gives purity, Value gives brightness.	Different ways of thinking about colour.
White	(1, 1, 1) or (255,255,255)	(0, 0, 0) (No ink)	(Y=1, I=0, Q=0)	(Any H, 0, 1)	How white is represented in each model.
Black	(0, 0, 0)	(1, 1, 1) (Full ink)	(Y=0, I=0, Q=0)	(Any H, 0, 0)	How black is represented in each model.
Colour Space Representation	 Cube	 Cube	 Parallelogram	 Double Cone (Hexcone)	Geometric representation of colour space.
Advantages	✓ Simple and intuitive. ✓ Direct representation of emitted light. ✓ Widely used in displays.	✓ Good for printing. ✓ Inks are inexpensive and easy to use.	✓ Separates brightness from colour. ✓ Saves bandwidth in TV transmission. ✗ Backward compatible with B/W TV.	✓ Close to human perception. ✓ Easy to select and modify colours. ✓ Useful in image processing and UI.	Each model has its own strengths.
Limitations	✗ Not suitable for printing. ✗ Not intuitive for editing hue/saturation.	✗ Not suitable for displays (emissive devices). ✗ Ink imperfections.	✗ Not suitable for general purpose graphics. ✗ Less intuitive.	✗ Conversion from/to RGB involves complex calculations.	No model is perfect for all applications.
Applications	Monitors, TVs, Smartphones, Projectors, LEDs, Cameras.	Printers, Photocopiers, Paints, Newspapers, Magazines.	Analog TV (NTSC), Video signal transmission.	Image editing software (Photoshop), Computer graphics, Visualization.	Depends on the requirement.

Key Takeaways

- RGB is additive and used in devices that emit light.
- CMY is subtractive and used in printing and painting.
- YIQ separates brightness (Y) from colour (I, Q) and is used in TV systems.
- HSV matches human perception and is best for colour manipulation.

Summary

Colour models are mathematical systems used to represent colours in different ways depending on the application. Choosing the right colour model is important for efficient storage, processing, display and reproduction of images.