COMPUTER GRAPHICS AND MULTIMEDIA - (18MCA41C)

UNIT-III

'Visible-Surface Detection Methods'

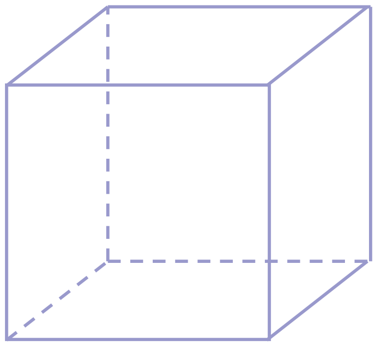
FACULTY:

Dr. R. A. Roseline, M.Sc., M.Phil., Ph.D.,
Associate Professor and Head,
Post Graduate and Research Department of Computer Applications,
Government Arts College (Autonomous), Coimbatore – 641 018.

Contents

Abstract

- Hidden-surface elimination methods
- Identifying visible parts of a scene from a viewpoint
- Numerous algorithms
  - More memory - storage
  - More processing time – execution time
  - Only for special types of objects - constraints
- Deciding a method for a particular application
  - Complexity of the scene
  - Type of objects
  - Available equipment
  - Static or animated scene



Introduction

Classification of Visible-Surface Detection Algorithms

- ***Object-space methods*** vs. ***Image-space methods***
  - Object definition directly vs. their projected images
  - Most visible-surface algorithms use image-space methods
  - Object-space can be used effectively in some cases
    - Ex) Line-display algorithms
- Object-space methods
  - Compares objects and parts of objects to each other
- Image-space methods
  - Point by point at each pixel position on the projection plane

Sorting and Coherence Methods

- To improve performance
- ***Sorting***
  - Facilitate depth comparisons
    - Ordering the surfaces according to their distance from the viewplane
- ***Coherence***
  - Take advantage of regularity
    - Epipolar geometry
    - Topological coherence

Back-Face Detection

Inside-outside test

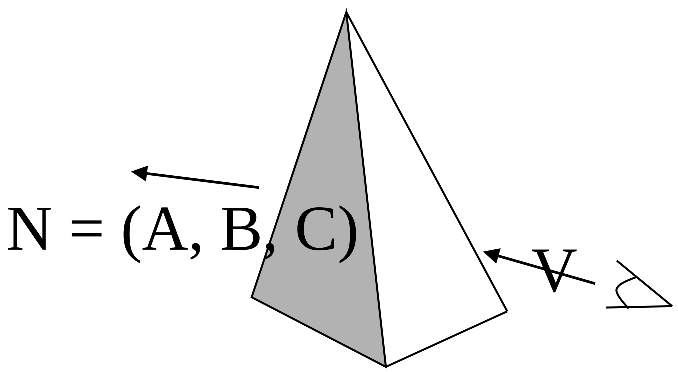
- A point (x, y, z) is "inside" a surface with plane parameters A, B, C, and D if

- The polyg

$$Ax + By + Cz + D < 0$$

- on is a back face if

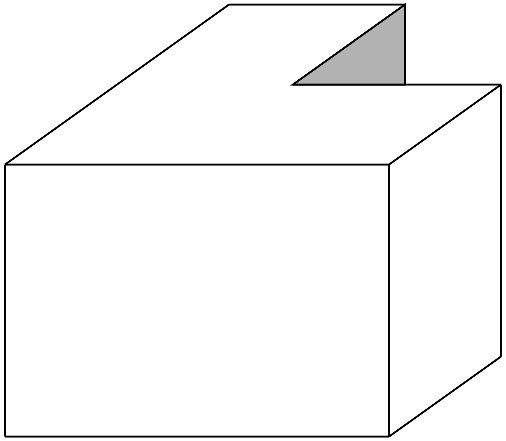
$$V \cdot N > 0$$



  - V is a vector in the viewing direction from the eye(camera)
  - N is the normal vector to a polygon surface

Advanced Configuration

- In the case of ***concave polyhedron***
  - Need more tests
    - Determine faces totally or partly obscured by other faces
  - In general, back-face removal can be expected to eliminate *about half of the surfaces* from further visibility tests

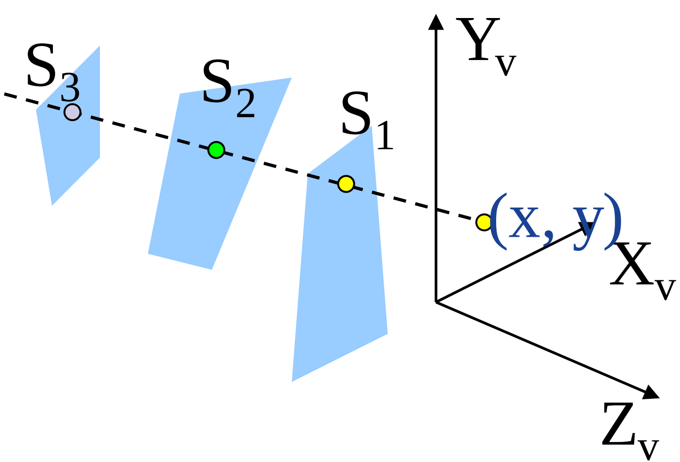


<View of a concave polyhedron with
one face partially hidden by other surfaces>

Depth-Buffer Method

Characteristics

- Commonly used image-space approach
- Compares depths of each pixel on the projection plane
  - Referred to as the ***z-buffer*** method
- Usually applied to scenes of polygonal surfaces
  - Depth values can be computed very quickly
  - Easy to implement



Depth Buffer & Refresh Buffer

- Two buffer areas are required
  - Depth buffer
    - Store depth values for each (x, y) position
    - All positions are initialized to minimum depth
      - Usually 0 – most distant depth from the viewplane
  - Refresh buffer
    - Stores the intensity values for each position
    - All positions are initialized to the background intensity

Algorithm

- Initialize the depth buffer and refresh buffer

  depth(x, y) = 0,   refresh(x, y) = $I_{backgnd}$
- For each position on each polygon surface
  - Calculate the depth for each (x, y) position on the polygon
  - If z > depth(x, y), then set

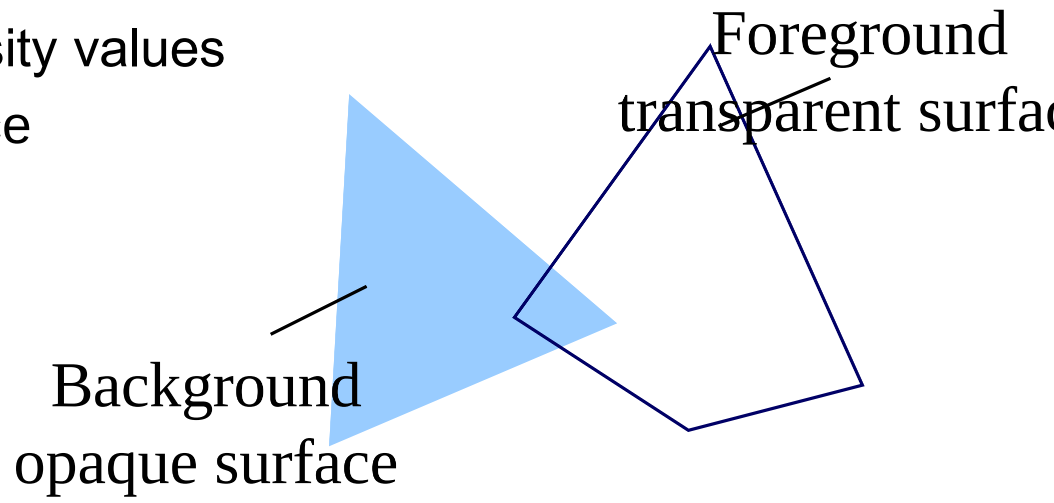
    depth(x, y) = z,   refresh(x, y) = I_{surf}(x, y)
- Advanced
  - With resolution of 1024 by 1024
    - Over a million positions in the depth buffer
  - Process one section of the scene at a time
    - Need a smaller depth buffer
    - The buffer is reused for the next section

A-Buffer Method

Characteristics

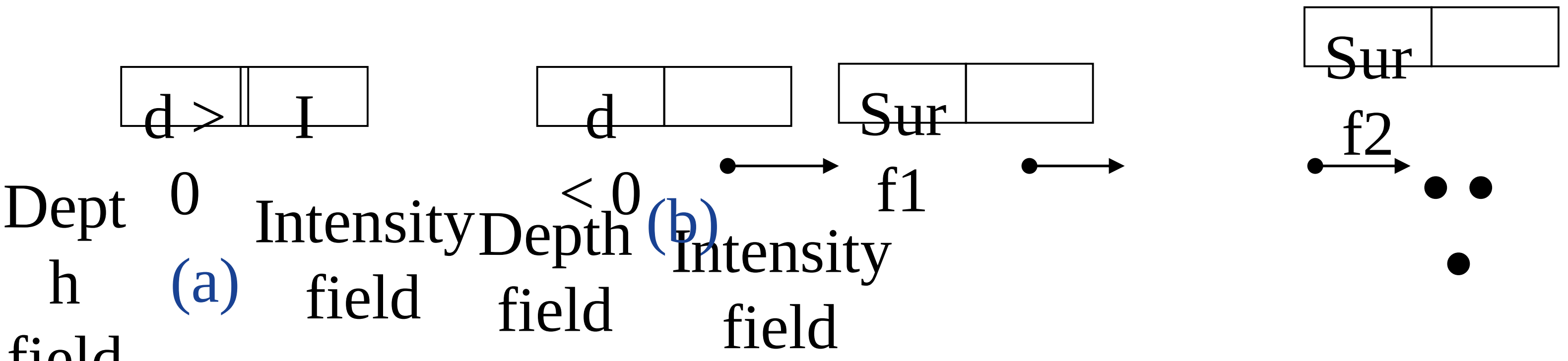
- An extension of the ideas in the depth-buffer method
- The origin of this name
  - At the other end of the alphabet from "z-buffer"
  - Antialiased, area-averaged, accumulation-buffer
  - Surface-rendering system developed by 'Lucasfilm'
    - REYES(Renders Everything You Ever Saw)
- A drawback of the depth-buffer method
  - Deals only with opaque surfaces
  - Can't accumulate intensity values

  for more than one surface



Algorithm(1 / 2)

- Each position in the buffer can reference a linked list of surfaces
  - Several intensities can be considered at each pixel position
  - Object edges can be antialiased
- Each position in the A-buffer has two fields
  - Depth field
    - Stores a positive or negative real number
  - Intensity field
    - Stores surface-intensity information or a pointer value



<Organization of an A-buffer pixel position : (a) single-surface c

Algorithm(2 / 2)

- If the depth field is positive
  - The number at that position is the depth
  - The intensity field stores the RGB
- If the depth field is negative
  - Multiple-surface contributions to the pixel
  - The intensity field stores a pointer to a linked list of surfaces
  - Data for each surface in the linked list

- RGB intensity components
- Opacity parameters(percent of transparency)
- Depth
- Percent of area coverage
- Surface identifier
- Pointers to next surface

Scan-Line Method

Characteristics

- Extension of the scan-line algorithm for filling polygon interiors
  - For all polygons intersecting each scan line
    - Processed from left to right
    - Depth calculations for each overlapping surface
    - The intensity of the nearest position is entered into the refresh buffer

Tables for The Various Surfaces

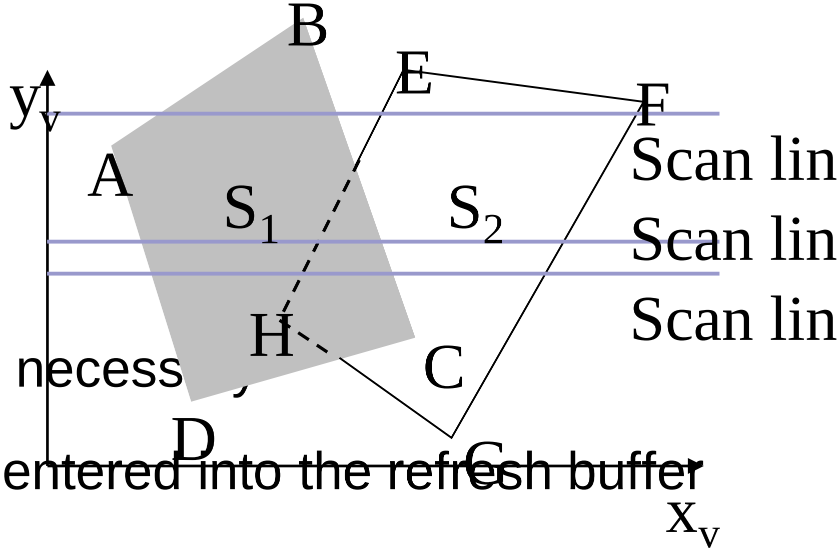
- Edge table
  - Coordinate endpoints for each line
  - Slope of each line
  - Pointers into the polygon table
    - Identify the surfaces bounded by each line
- Polygon table
  - Coefficients of the plane equation for each surface
  - Intensity information for the surfaces
  - Pointers into the edge table

Active List & Flag

- Active list
  - Contain only edges across the current scan line
  - Sorted in order of increasing x
- Flag for each surface
  - Indicate whether inside or outside of the surface
  - At the leftmost boundary of a surface
    - The surface flag is turned on
  - At the rightmost boundary of a surface
    - The surface flag is turned off

Example

- Active list for scan line 1
  - Edge table
    - AB, BC, EH, and FG
    - Between AB and BC, only the flag for surface S_1 is on
      - No depth calculations are necessary
      - Intensity for surface S_1 is entered into the refresh buffer
    - Similarly, between EH and FG, only the flag for S_2 is on

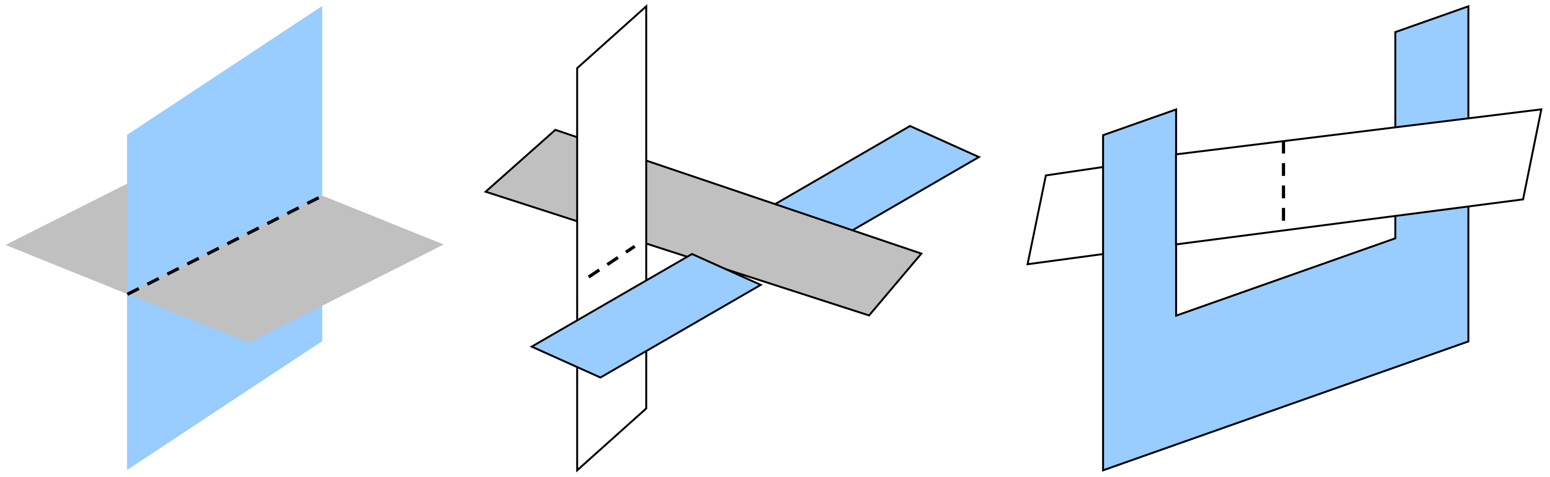


Example(cont.)

- For scan line 2, 3
  - AD, EH, BC, and FG
    - Between AD and EH, only the flag for S_1 is on
    - Between EH and BC, the flags for both surfaces are on
      - Depth calculation is needed
      - Intensities for S_1 are loaded into the refresh buffer until BC
  - Take advantage of coherence
    - Pass from one scan line to next
    - Scan line 3 has the same active list as scan line 2
    - Unnecessary to make depth calculations between EH and BC

Drawback

- Only if surfaces don't cut through or otherwise cyclically overlap each other
  - If any kind of cyclic overlap is present
    - Divide the surfaces



Depth-Sorting Method

Operations

- Image-space and object-space operations
  - Sorting operations in both image and object-space
  - The scan conversion of polygon surfaces in image-space
- Basic functions
  - Surfaces are sorted in order of decreasing depth
  - Surfaces are scan-converted in order, starting with the surface of greatest depth

Algorithm

- Referred to as the ***painter's algorithm***
  - In creating an oil painting
    - First paints the background colors
    - The most distant objects are added
    - Then the nearer objects, and so forth
    - Finally, the foregrounds are painted over all objects
  - Each layer of paint covers up the previous layer
- Process
  - Sort surfaces according to their distance from the viewplane
  - The intensities for the farthest surface are then entered into the refresh buffer
  - Taking each succeeding surface in decreasing depth order

Overlapping Tests

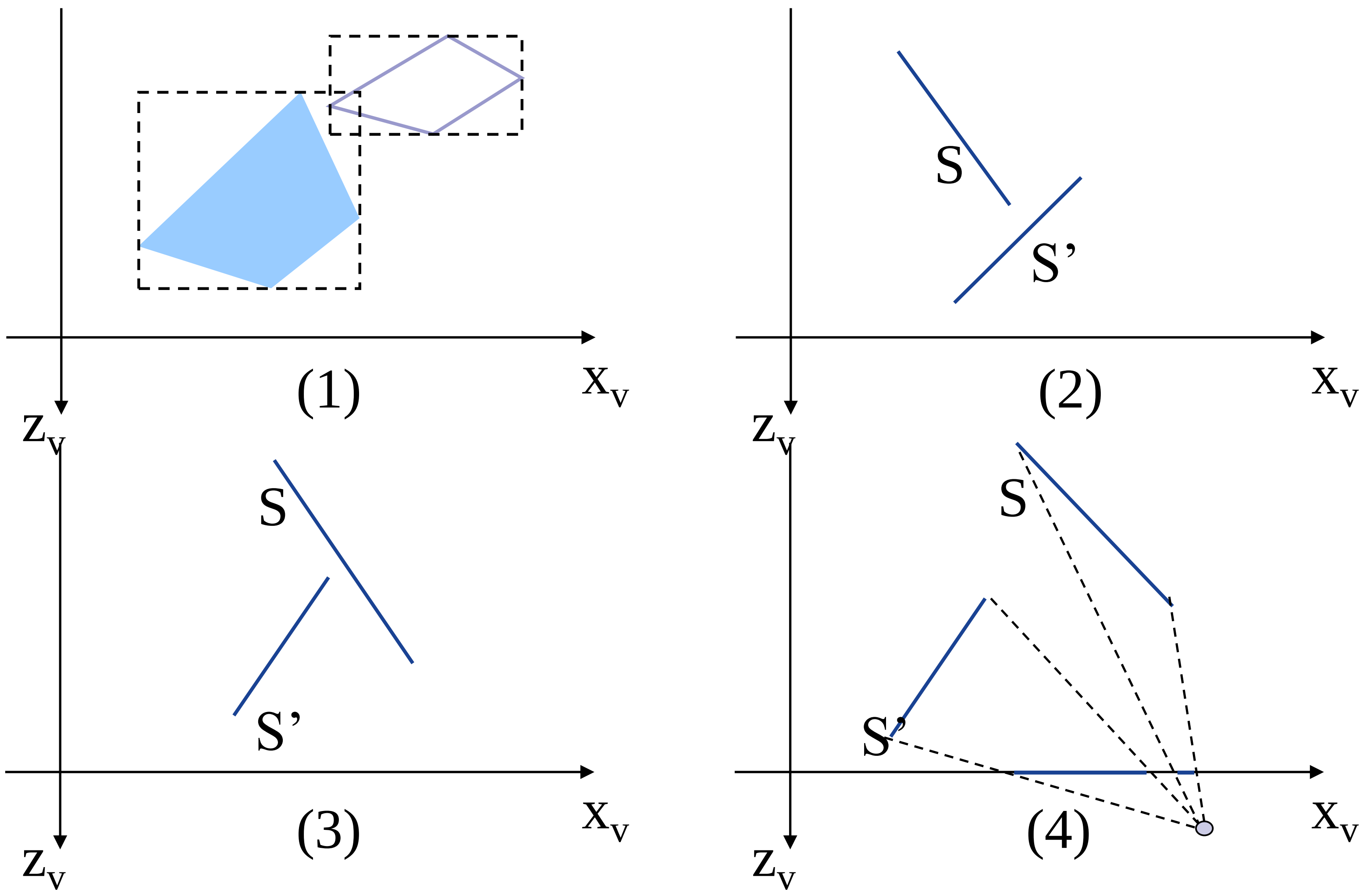
- Tests for each surface that overlaps with S
  - The bounding rectangle in the xy plane for the two surfaces do not overlap (1)
  - Surface S is completely behind the overlapping surface relative to the viewing position (2)
  - The overlapping surface is completely in front of S relative to the viewing position (3)
  - The projections of the two surfaces onto the viewplane do not overlap (4)

Easy ↕ Difficult

if all the surfaces pass at least one of the tests, none of them is behind S

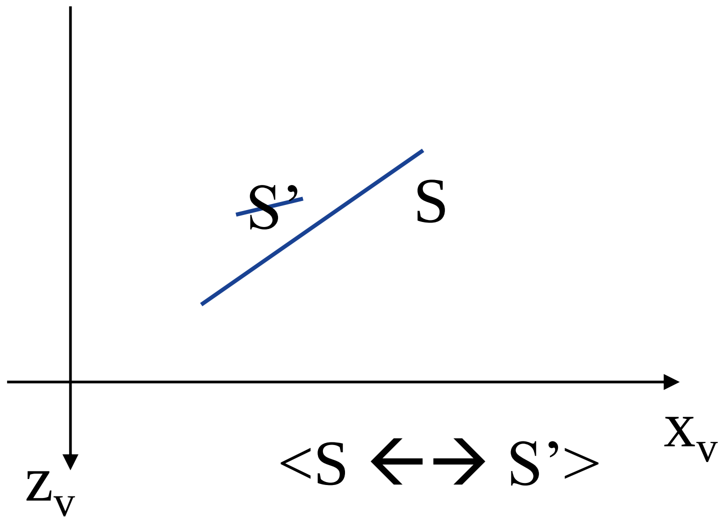
- No reordering is then necessary and S is scan converted

Overlapping Test Examples

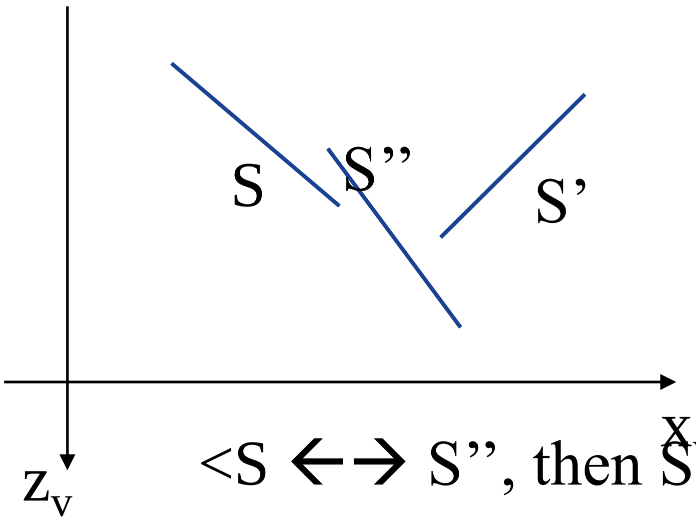


Surface Reordering

- If all four tests fail with S'
  - Interchange surfaces S and S' in the sorted list
  - Repeat the tests for each surface that is reordered in the list



<S ←→ S'>



<S ←→ S'', then S'' ←→ S

Drawback

- If two or more surfaces alternately obscure each other
  - Infinite loop
  - Flag any surface that has been reordered to a farther depth
    - It can't be moved again
  - If an attempt to switch the surface a second time
    - Divide it into two parts to eliminate the cyclic loop
    - The original surface is then replaced by the two new surfaces

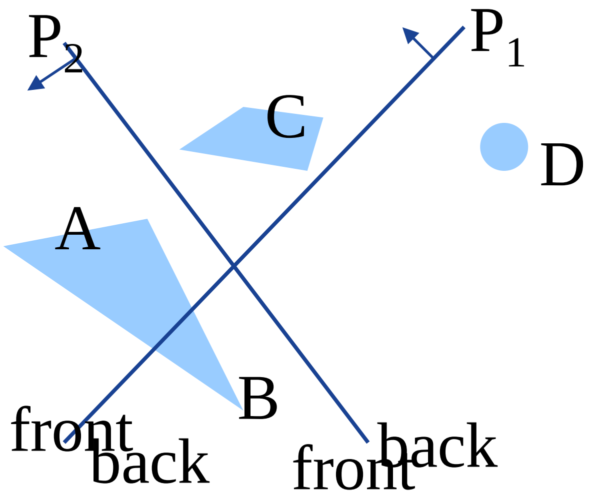
BSP-Tree Method

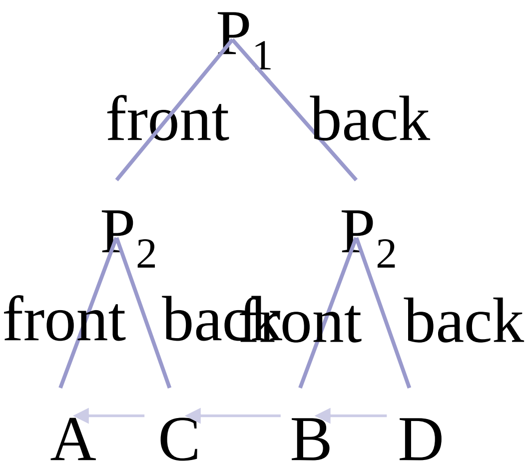
Characteristics

- Binary Space-Partitioning(BSP) Tree
- Determining object visibility by painting surfaces onto the screen from back to front
  - Like the painter's algorithm
- Particularly useful
  - The view reference point changes
  - The objects in a scene are at fixed positions

Process

- Identifying surfaces
  - “inside” and “outside” the partitioning plane
- Intersected object
  - Divide the object into two separate objects(A, B)

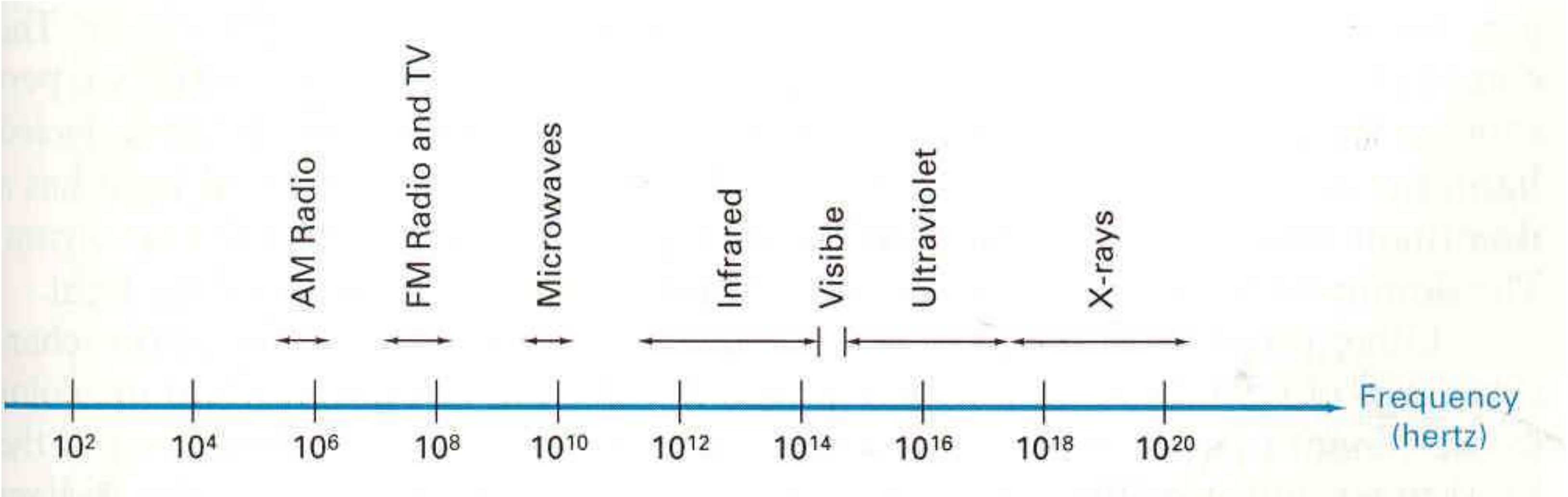




Clolor Models

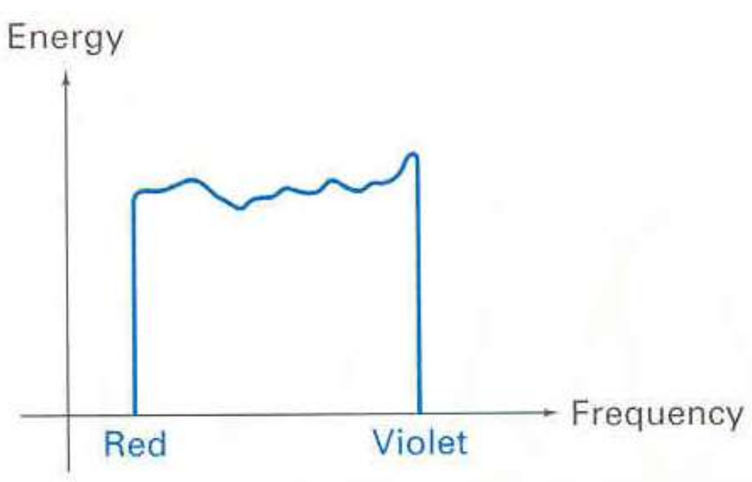
Electromagnetic Spectrum

- Visible Light Frequencies Range between
  - Red: 4.3×10^{14} hertz (700nm)
  - Violet: 7.5×10^{14} hertz (400nm)

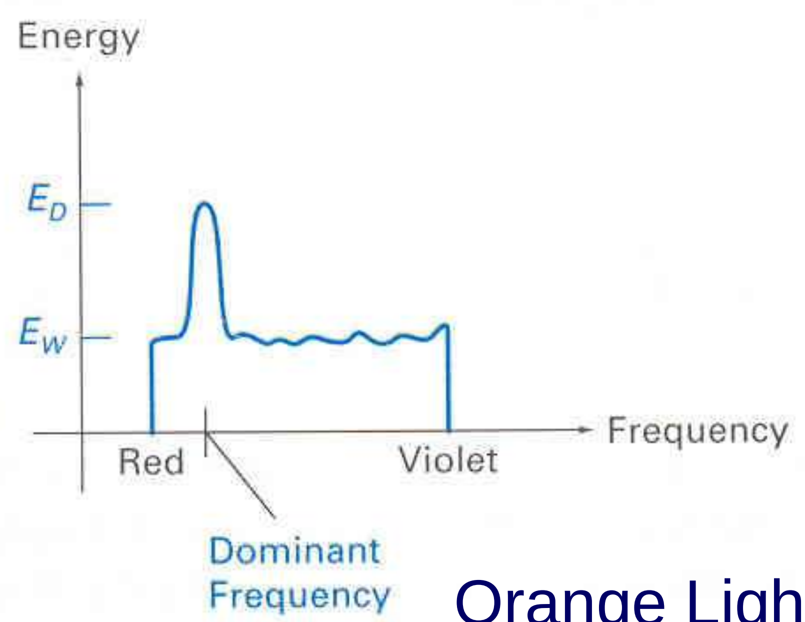


Visible Light

- The Color of Light is Characterized by
  - Hue: dominant frequency (highest peak)
  - Saturation: excitation purity (ratio of highest to rest)
  - Brightness: luminance (area under curve)



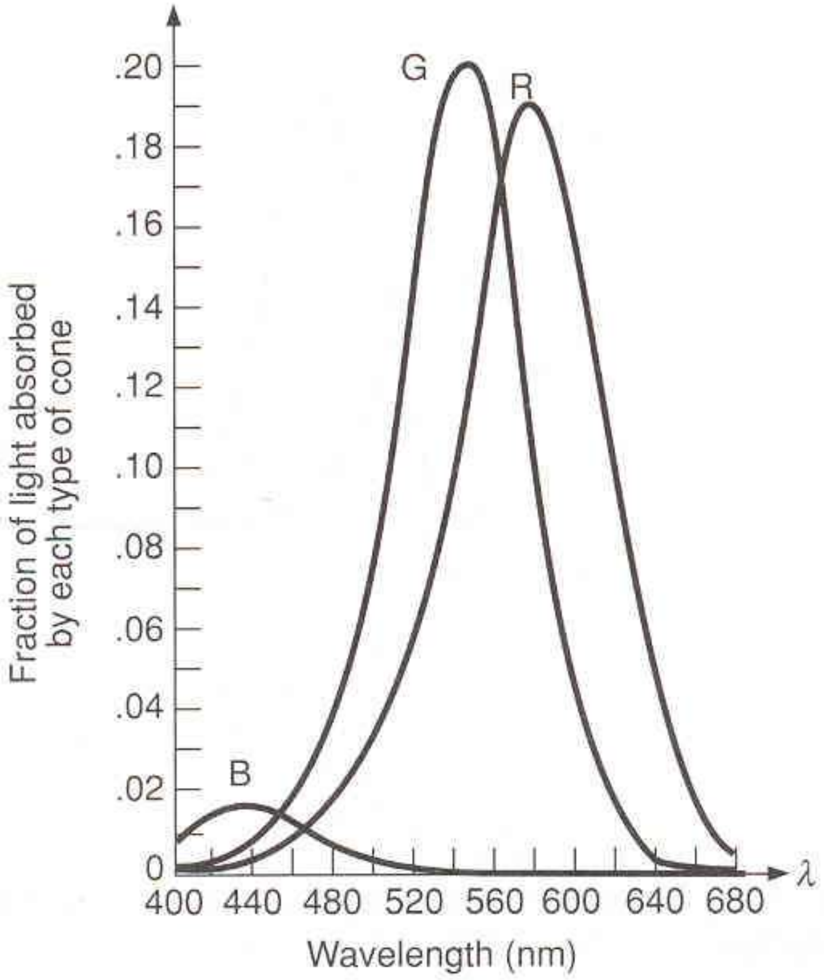
White Light



Orange Light

Color Perception

- Tristimulus Theory of Color
  - Spectral-response functions of each of the three types of cones on the human retina

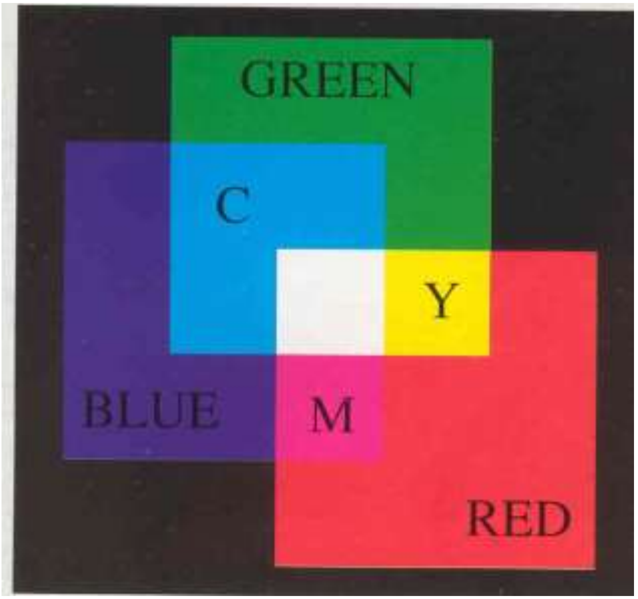


Color Models

- RGB
- XYZ
- CMY
- HSV
- Others

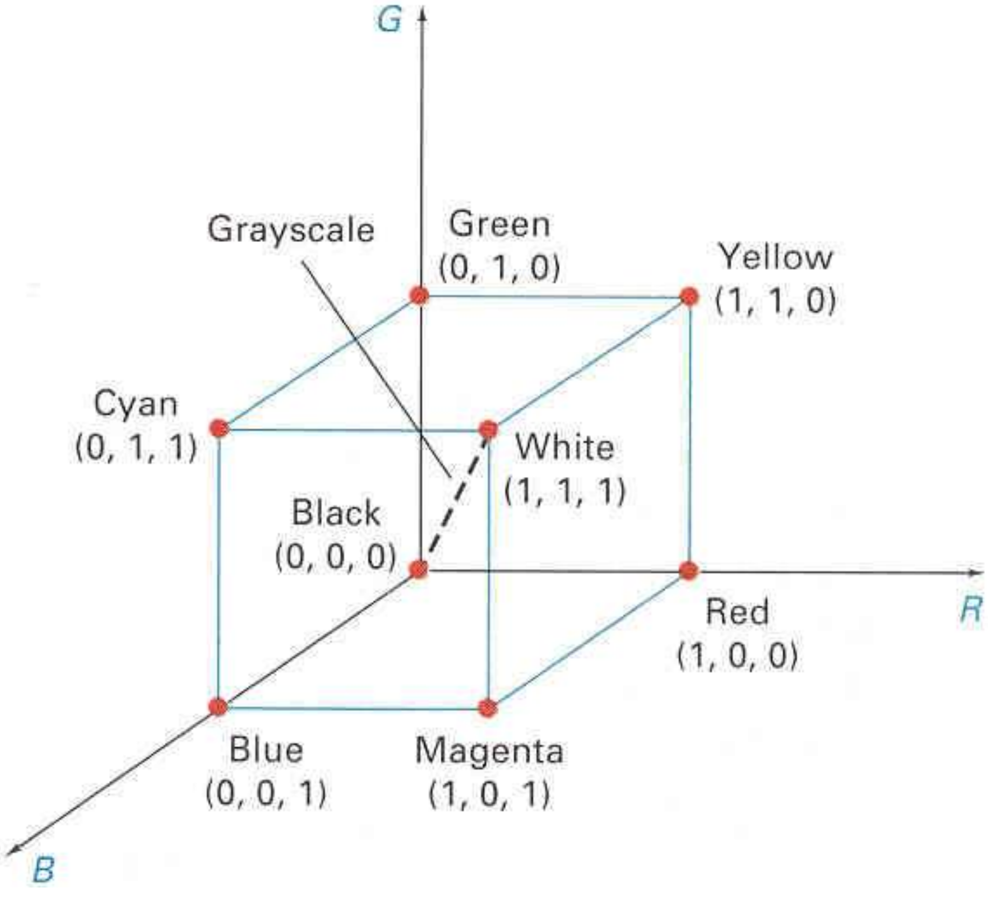
RGB Color Model

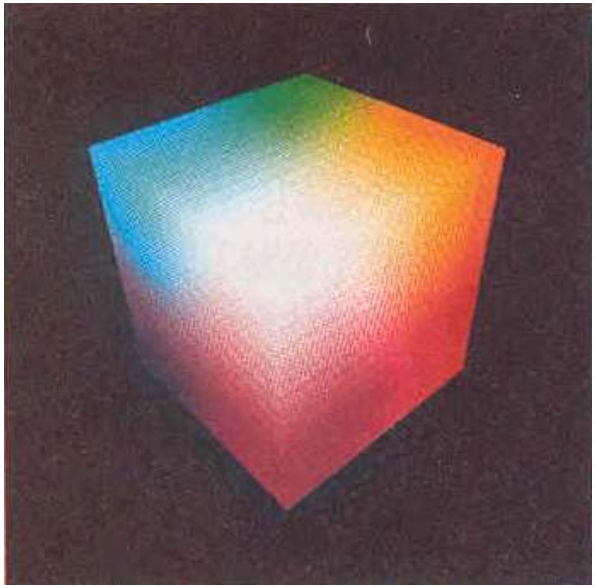
- Colors are Additive



R	G	B	Color
0.0	0.0	0.0	Black
1.0	0.0	0.0	Red
0.0	1.0	0.0	Green
0.0	0.0	1.0	Blue
1.0	1.0	0.0	Yellow
1.0	0.0	1.0	Magenta
0.0	1.0	1.0	Cyan
1.0	1.0	1.0	White

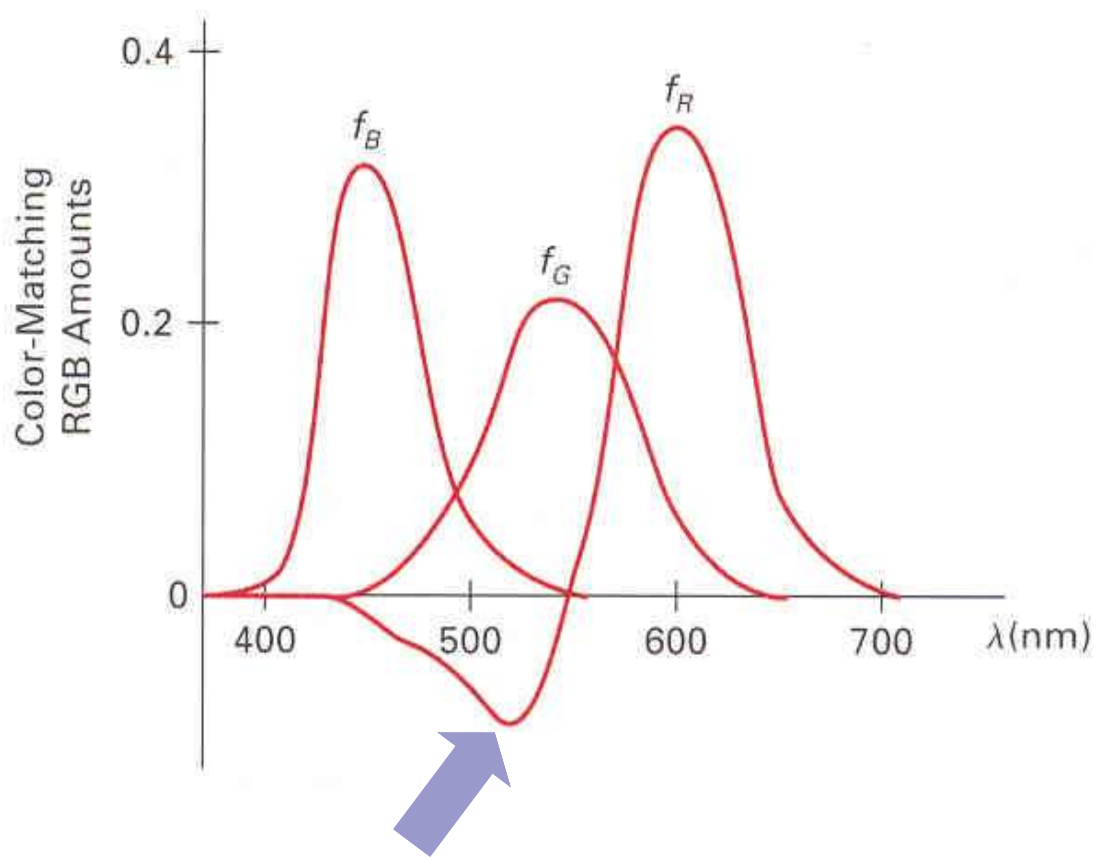
RGB Color Cube





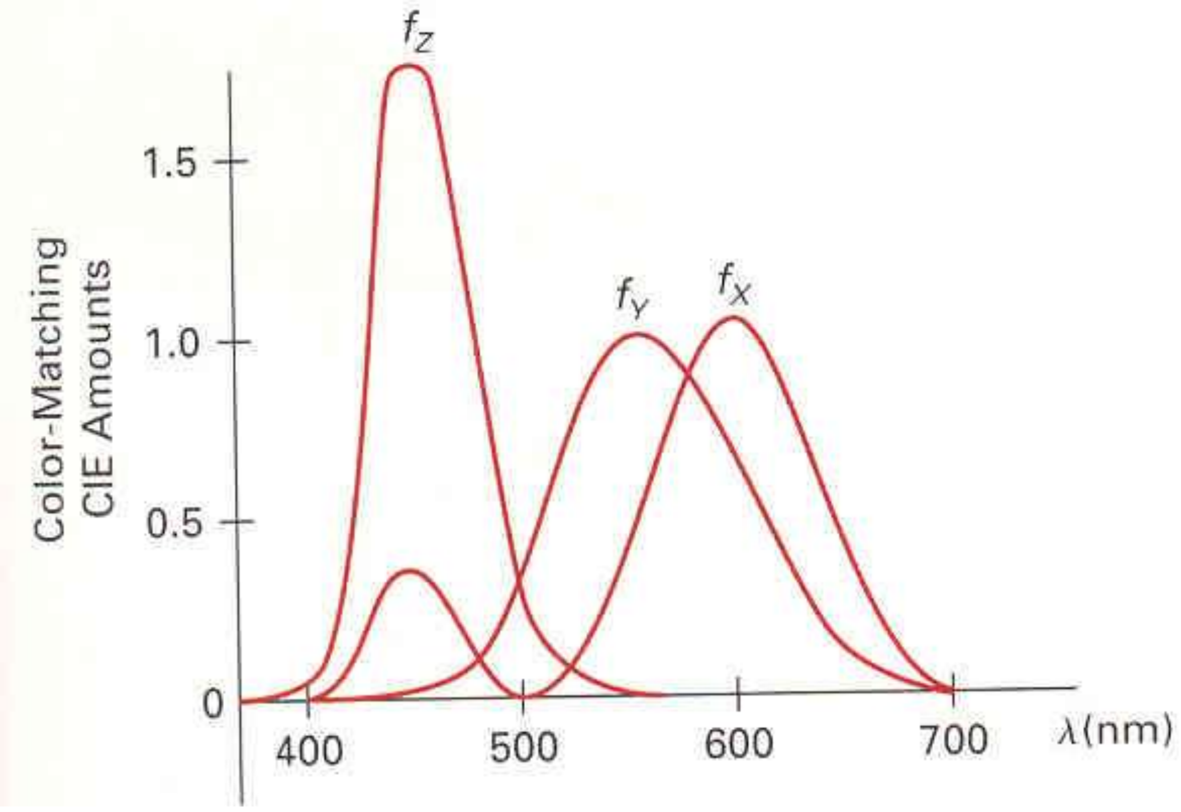
RGB Spectral Colors

- Amounts of RGB Primaries Needed to Display Spectral Colors



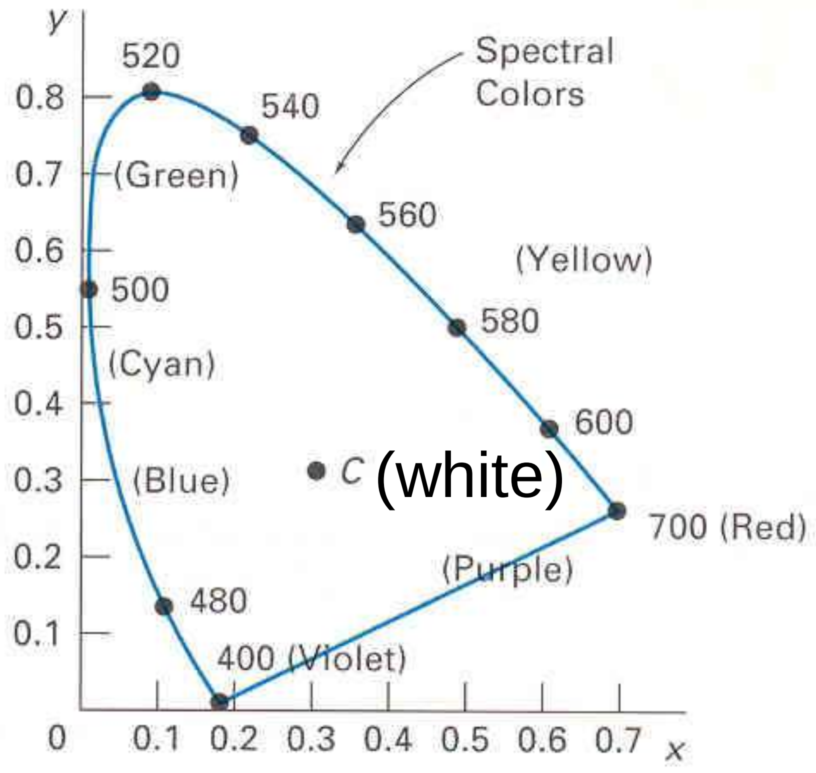
XYZ Color Model (CIE)

- Amounts of CIE Primaries Needed to Display Spectral Colors

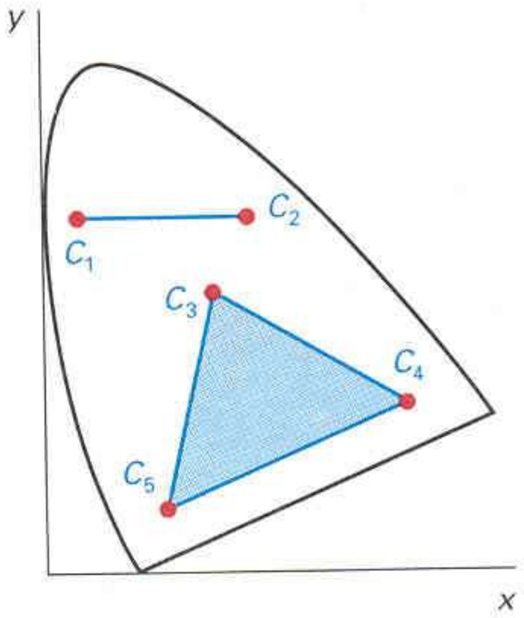


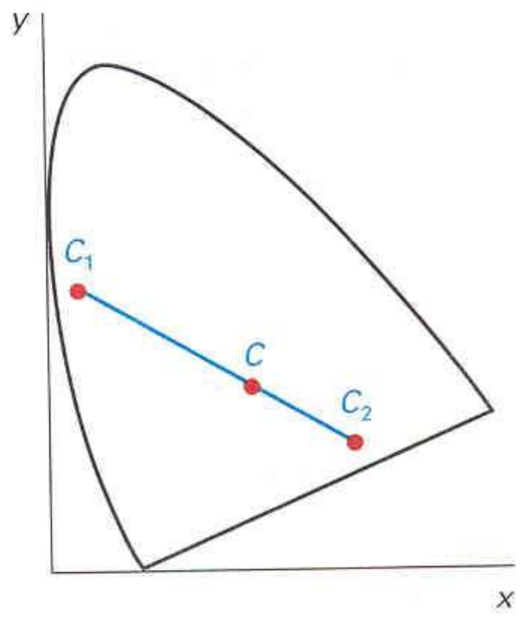
CIE Chromaticity Diagram

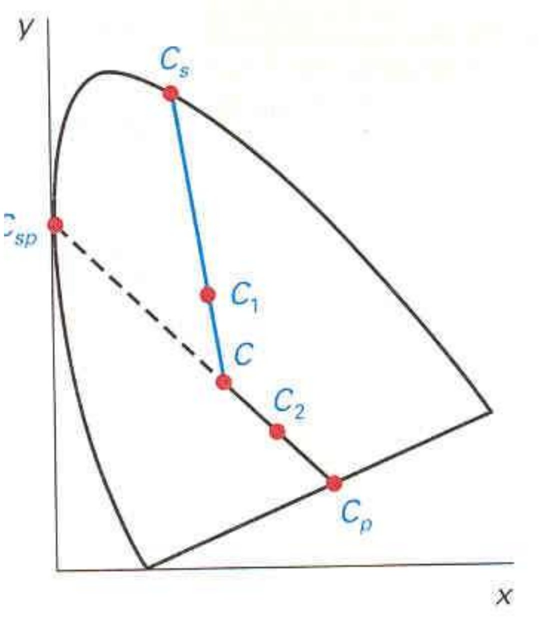
- Normalized Amounts of X and Y for Colors in Visible Spectrum



CIE Chromaticity Diagram







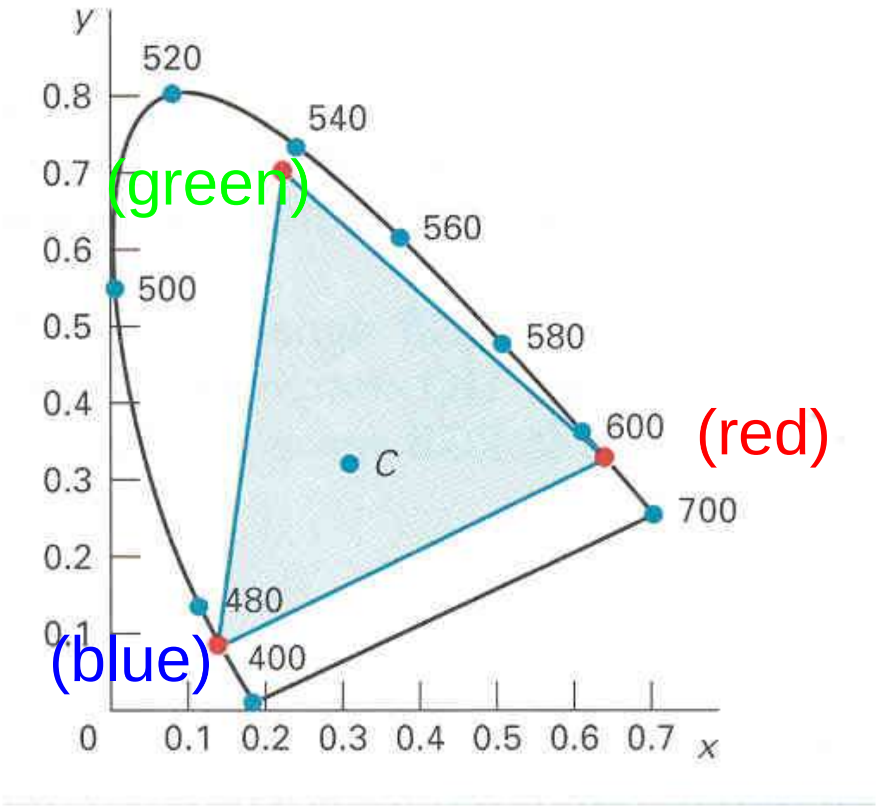
Define
Color
Gamuts

Represent
Complementary
Color

Determine
Dominant Wavelength
and Purity

RGB Color Gamut

- Color Gamut for a Typical RGB Computer Monitor

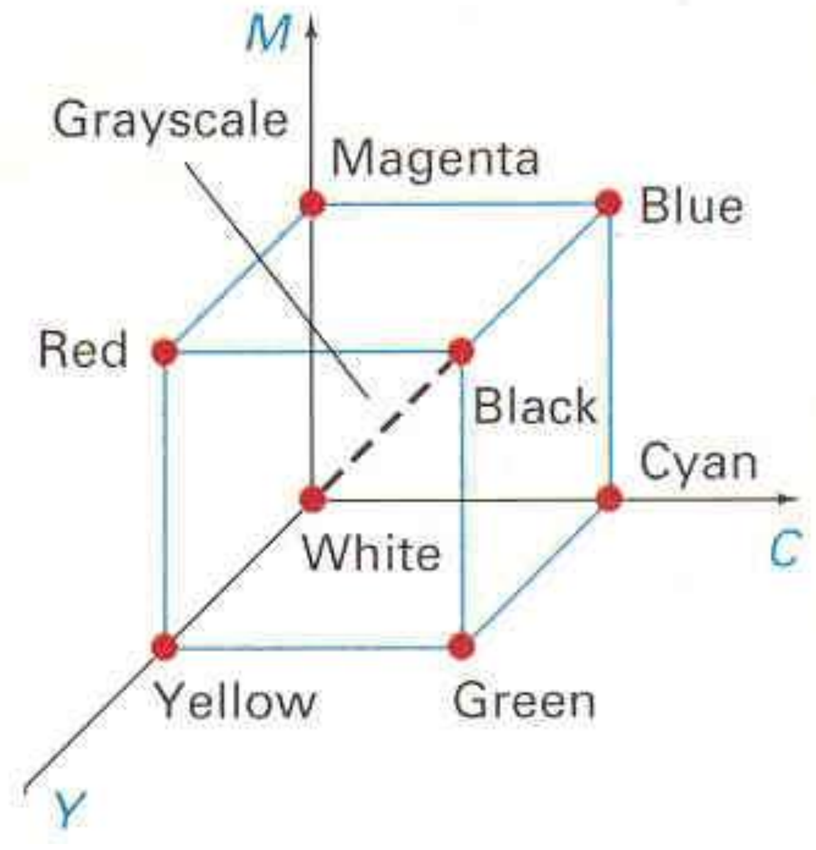


CMY Color Model

- Colors are Subtractive

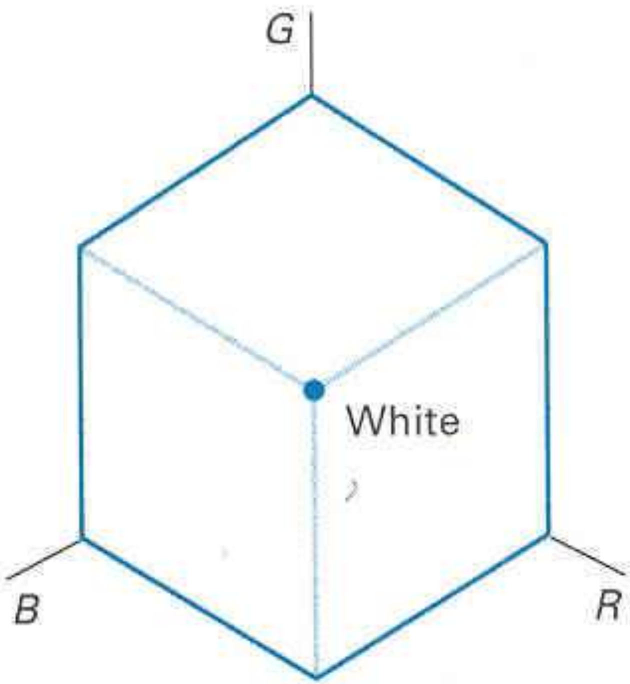
C	M	Y	Color
0.0	0.0	0.0	White
1.0	0.0	0.0	Cyan
0.0	1.0	0.0	Magenta
0.0	0.0	1.0	Yellow
1.0	1.0	0.0	Blue
1.0	0.0	1.0	Green
0.0	1.0	1.0	Red
1.0	1.0	1.0	Black

CMY Color Cube

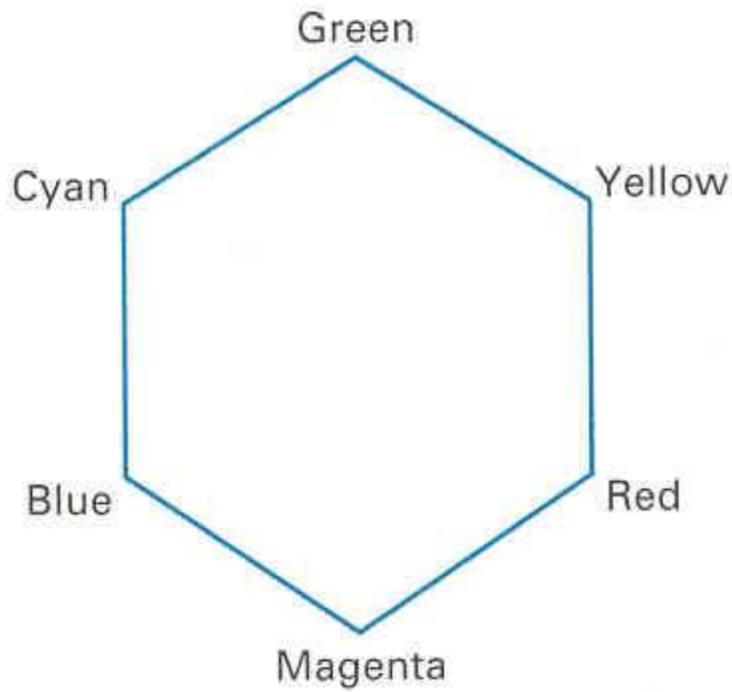


HSV Color Model

- Select a Spectral Color (Hue) and the Amount of White (Saturation) and Black (Value)

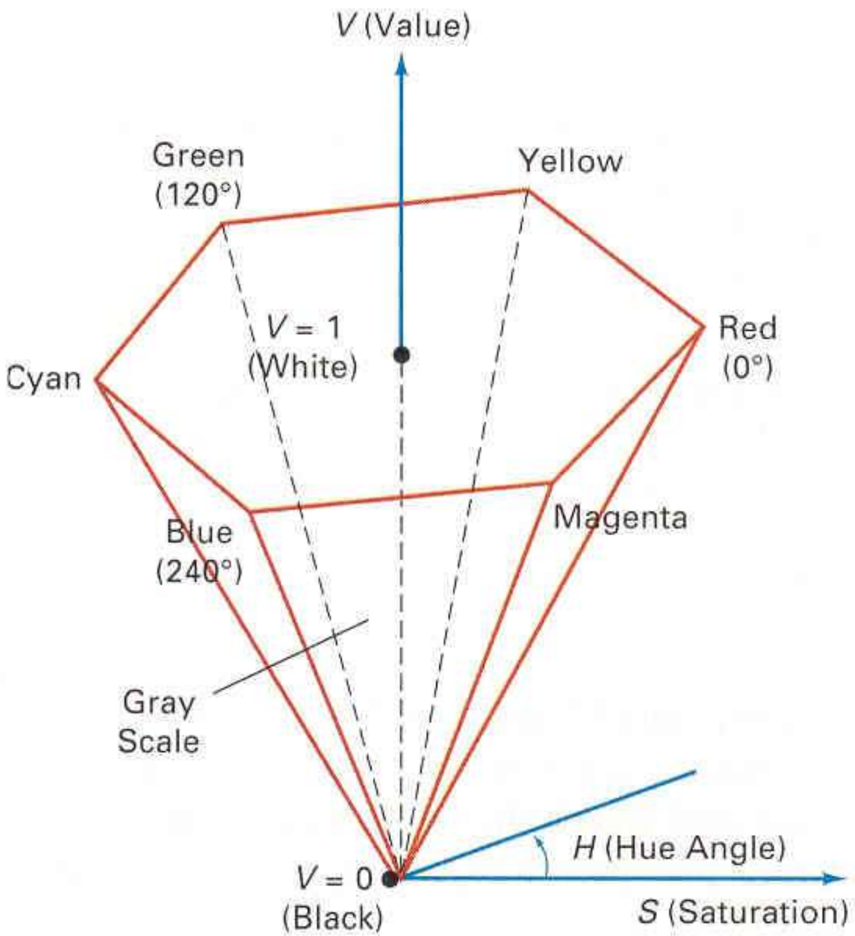


RGB Color Cube



Color Hexagon

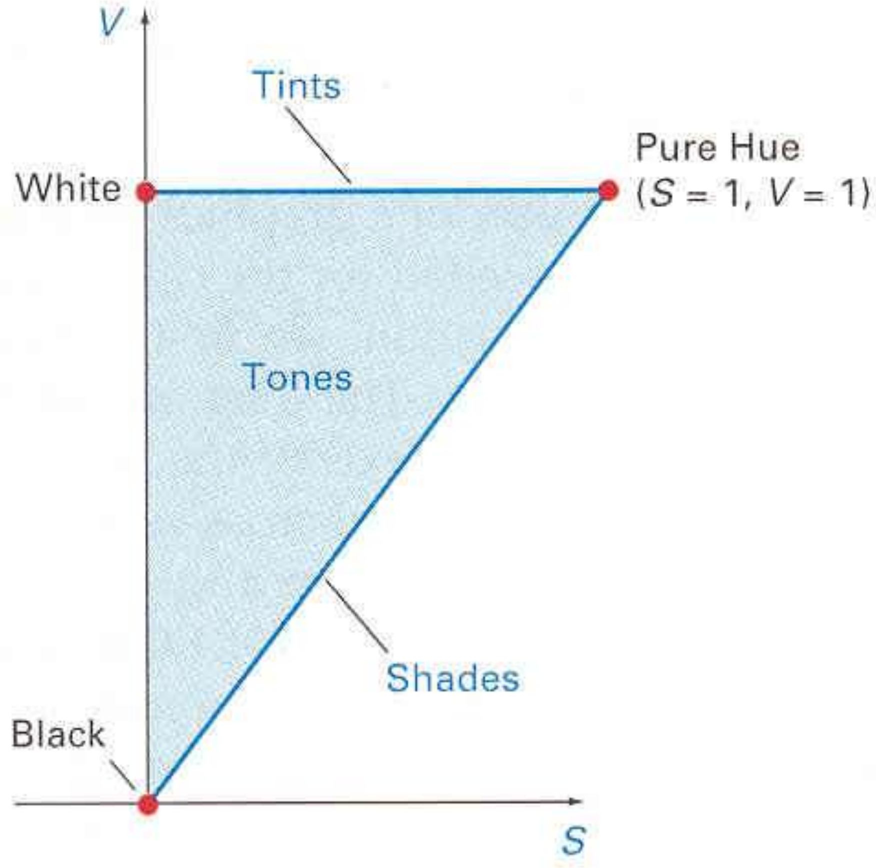
HSV Color Model



H	S	V	Color
0	1.0	1.0	Red
60	1.0	1.0	Yellow
120	1.0	1.0	Green
180	1.0	1.0	Cyan
240	1.0	1.0	Blue
300	1.0	1.0	Magenta
*	0.0	1.0	White
*	0.0	0.5	Gray
*	*	0.0	Black

HSV Color Model

- Cross Section of the HSV Hexcone



Thank you

The Content in this Material are from the Textbooks and Reference books given in the Syllabus