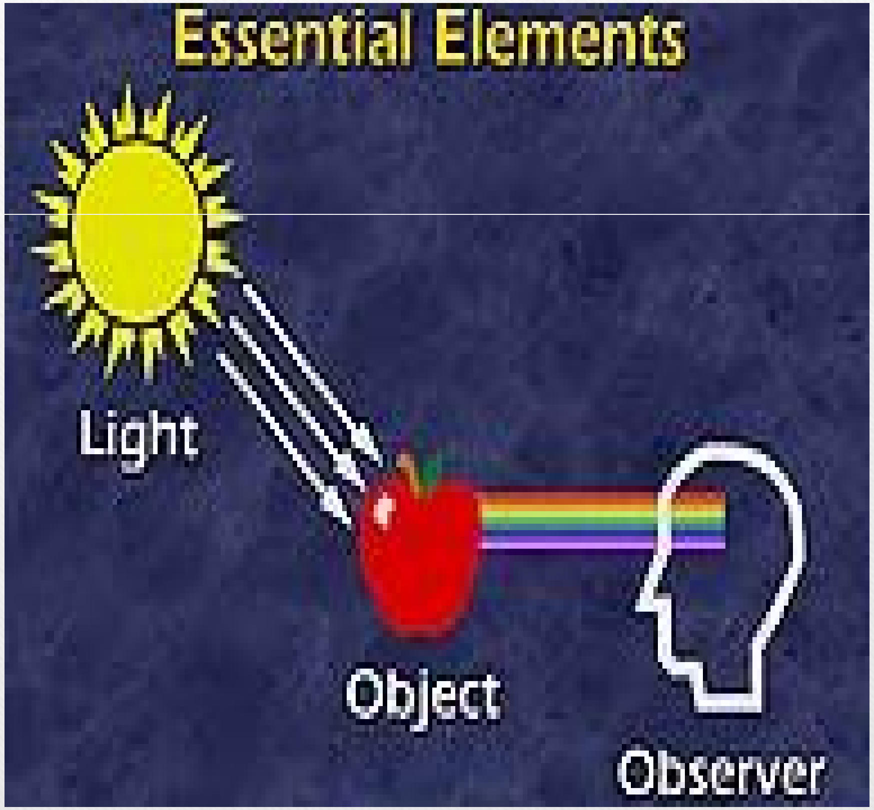
# *COLOR MODEL*

# *&*

# *RENDERING*

# What is the color?

- **Color** is a sensation produced by the human eye and nervous system.




To see color, three essential elements must be present:

•light,

•an illuminated object,

•and an observer.

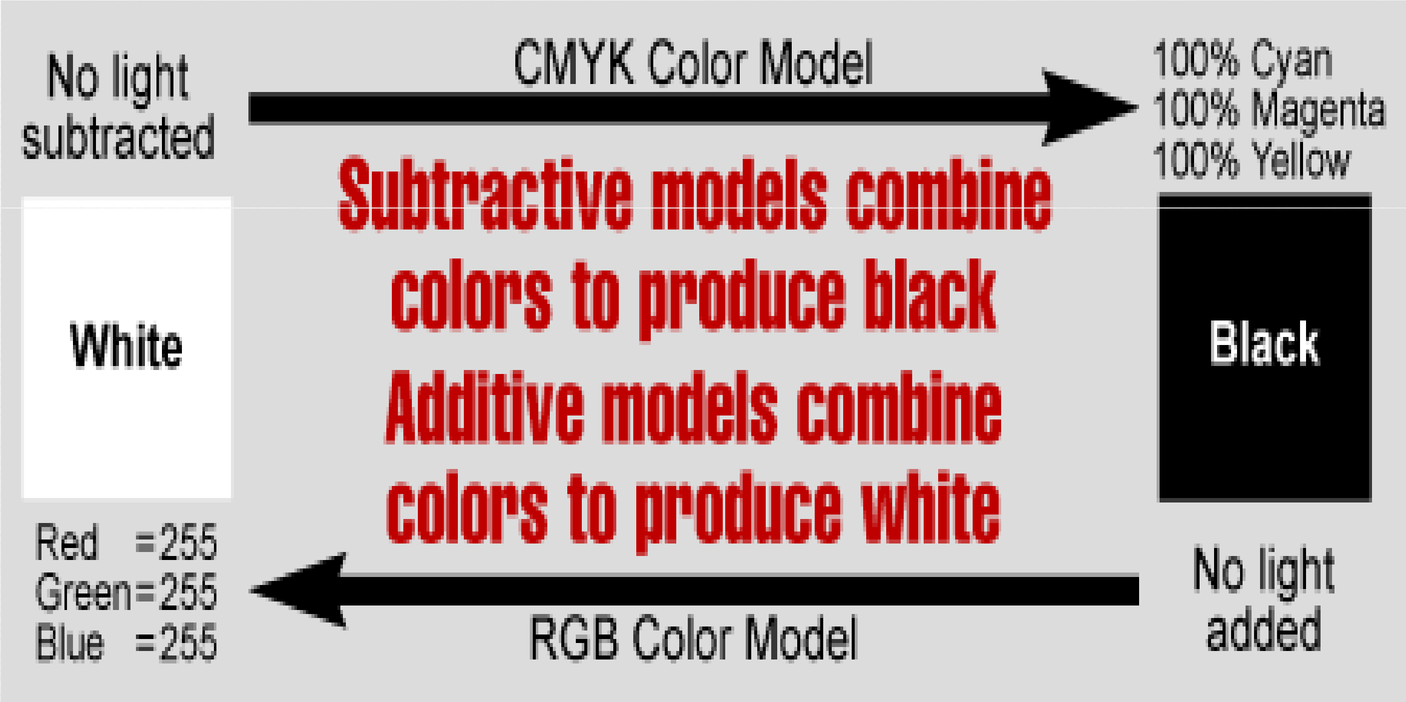
# *Color Model*

- **A *color model*** is an orderly system for creating a whole range of colors from a small set of primary colors.
- The range of colors that can be described by combinations of other colors is called a ***Color Gamut.***
- Color model is of 2 types:-

  * Additive

  * Subtractive

# Additive v/s Subtractive



# Color model for Raster Graphics

- **Hardware Oriented models**

  RGB color model

  CMY color model

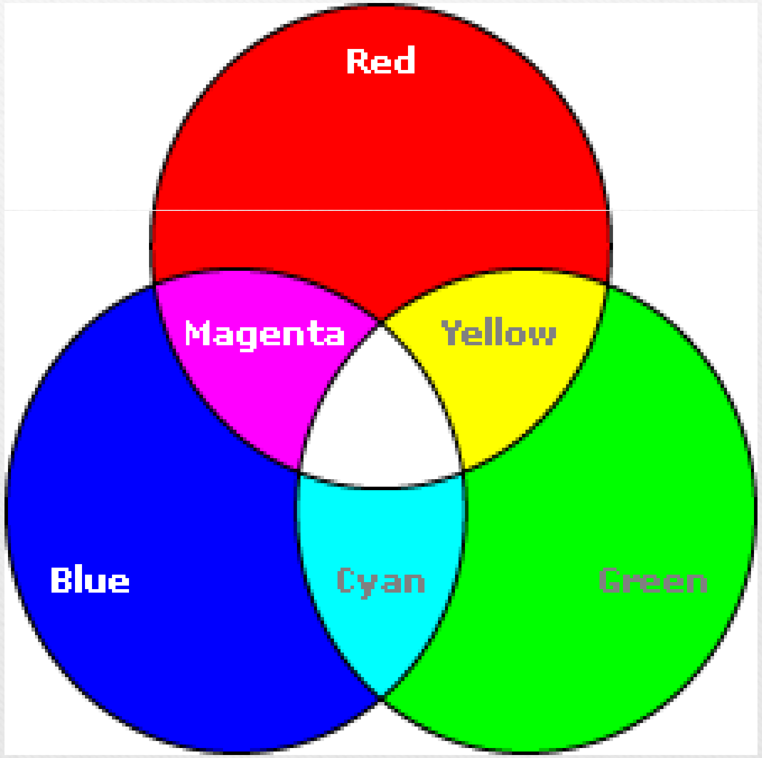
  YIQ color model

- **User Oriented model**

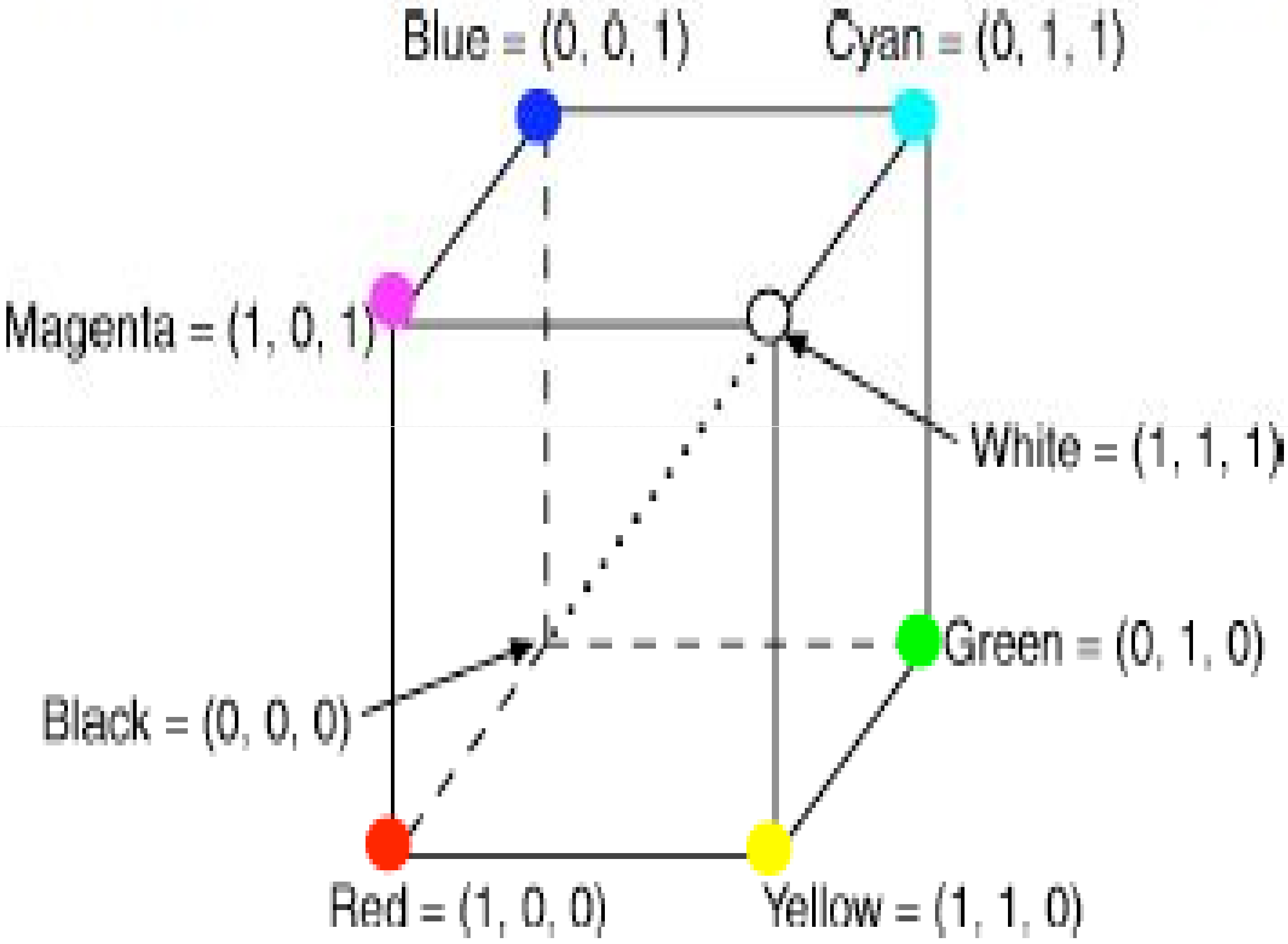
  HSV color model

# RGB Color Model:

* Additive color model.
* For computer displays.
* Uses light to display color.
* Colors result from transmitted light.
* Red + Green + Blue = White.

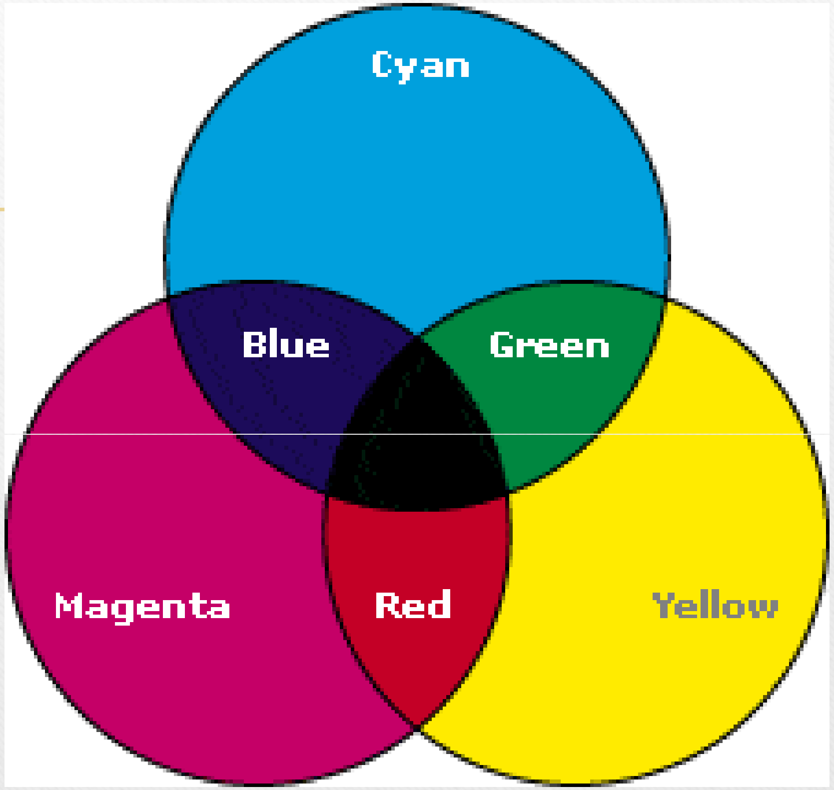


# The RGB cube (Grays on dotted main diagonal)



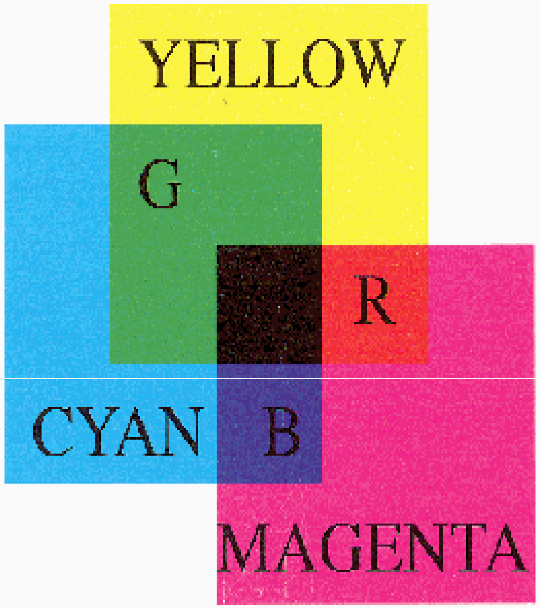
# CMY(K) Color Model:

* Subtractive color model.
* Used in ink-jet plotters that deposit pigment on paper.
* Uses ink to display color.
* Colors result from reflected light.
* K for Black




* Cyan + Magenta + Yellow = Black.

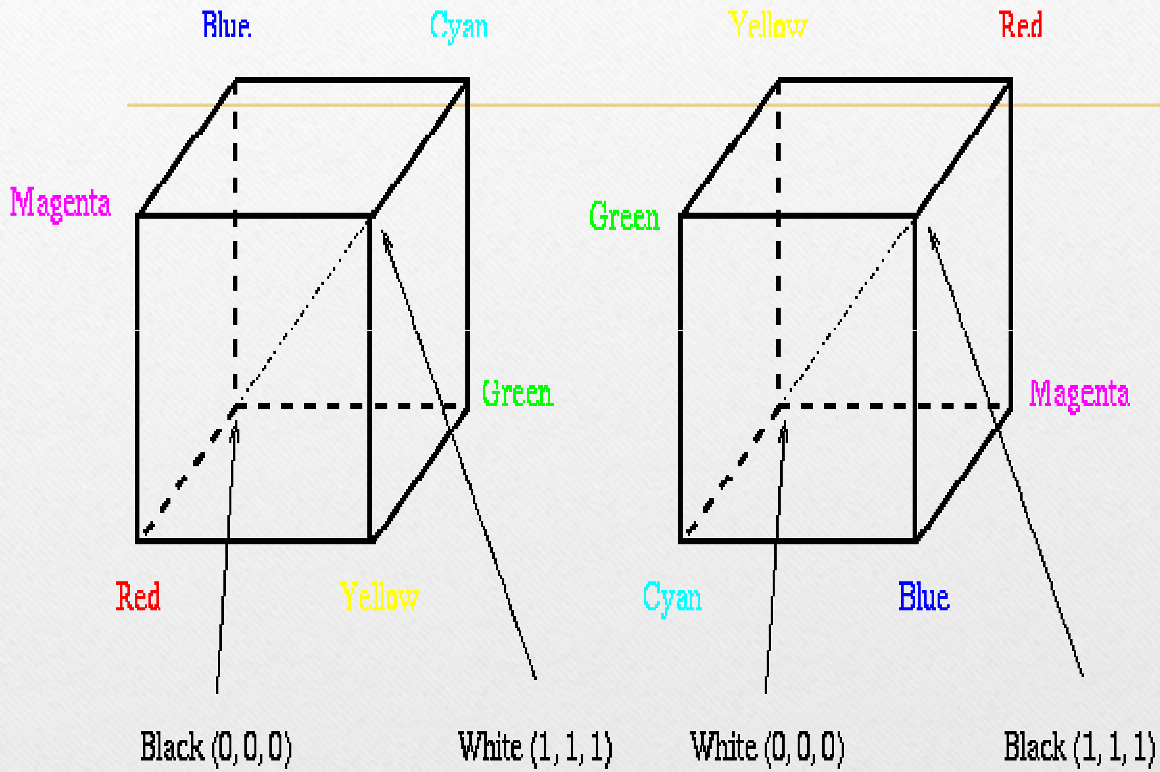
# CMY(K) Color Model:




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 |
| 0.5 | 0.0 | 0.0 | |
| 1.0 | 0.5 | 0.5 | |
| 1.0 | 0.5 | 0.0 | |

- Cyan, magenta, and yellow are complements of red, green, and blue.




The RGB Cube    The CMY Cube

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

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

# YIQ color model

- Y => Luminous or Brightness information.
- I => Hue (The degree to which a stimulus can be described as similar to or different from stimuli ).

  For eg. Rust = root color or hue is orange

  Navy = root color or hue is blue
- Q => Purity (Intensity of hue)

- **Y parameter**

The combination of Red, Green & Blue intensities are chosen for Y parameter to give standard luminously curve. Since Y contains luminous information, black and white television monitors use only Y signal. It occupies about 4 MHz bandwidth.

- **I parameter**

  It contains Orange, Cyan hue information that provides the Flesh tone shading (use color range to pull darker flesh from the shadows of images)

  It occupies the a bandwidth of approximately 1.5 MHz.

- **Q parameter**

  It contains Green, Magenta hue information.

  It occupies about 0.6 MHz bandwidth.

- RGB signal can be converted to television signal using NTSC encoder which converts RGB values to YIQ values, then modulate and superimpose the I and Q information on the Y signal.
- Conversion form RGB values to YIQ using following transformation:-

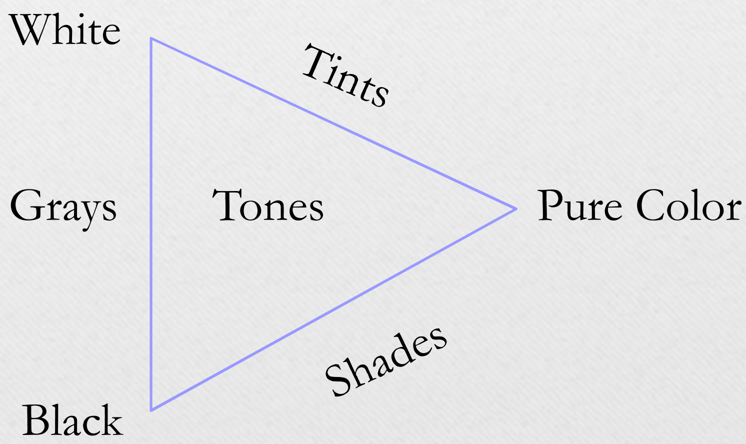
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- NTSC video signal can be converted to RGB signal using NTSC decoder, which separates the video signal into YIQ components, then converts to RGB using inverse matrix transformation :-

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.620 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.108 & 1.705 \end{bmatrix} . \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

# HSV color model

- Artists often specify color as tints, shades, and tones of saturated (pure) pigments
- Tint: Gotten by adding white to a pure pigment, decreasing saturation
- Shade: Gotten by adding black to a pure pigment, decreasing lightness
- Tone: Gotten by adding white and black to a pure pigment



# HSV color model

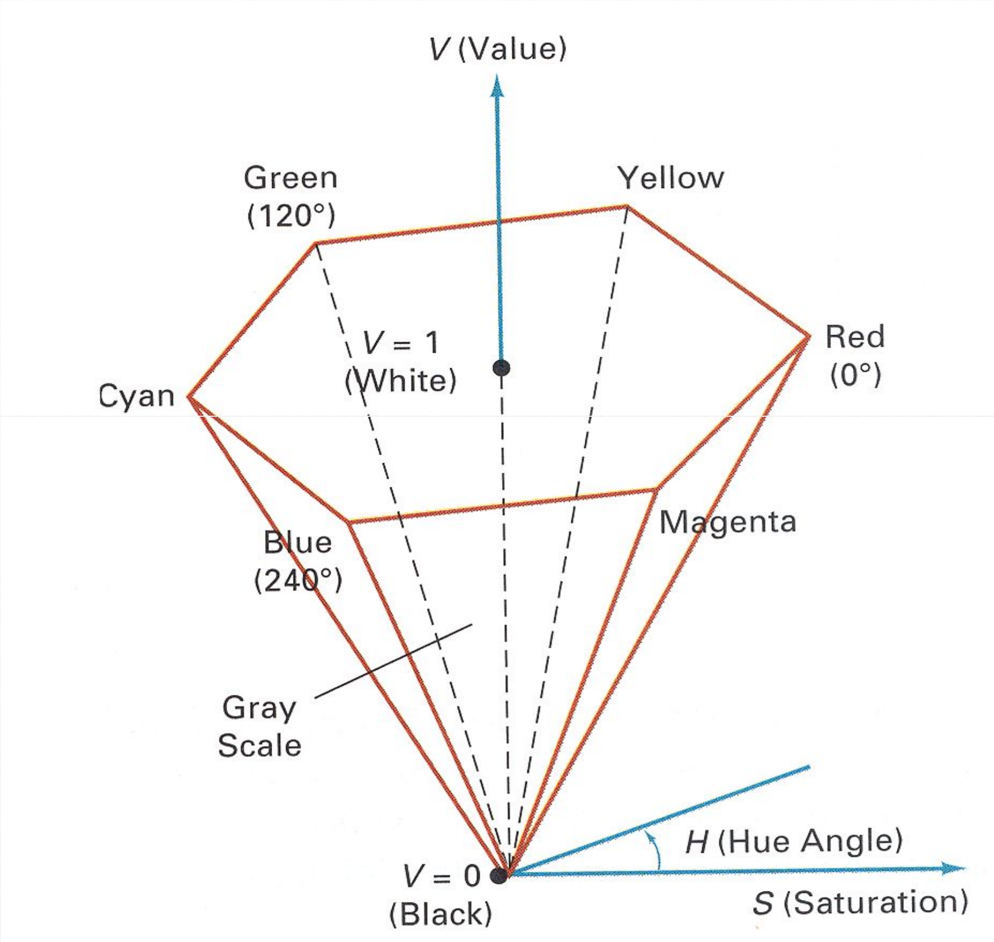
Computer scientists frequently use an intuitive color space that corresponds to tint, shade, and tone:

- Hue - The color we see (red, green, purple)

- Saturation - How far is the color from gray (pink is less saturated than red, sky blue is less saturated than royal blue)

- Brightness (Luminance) - How bright is the color (how bright are the lights illuminating the object?)

# HSV color model



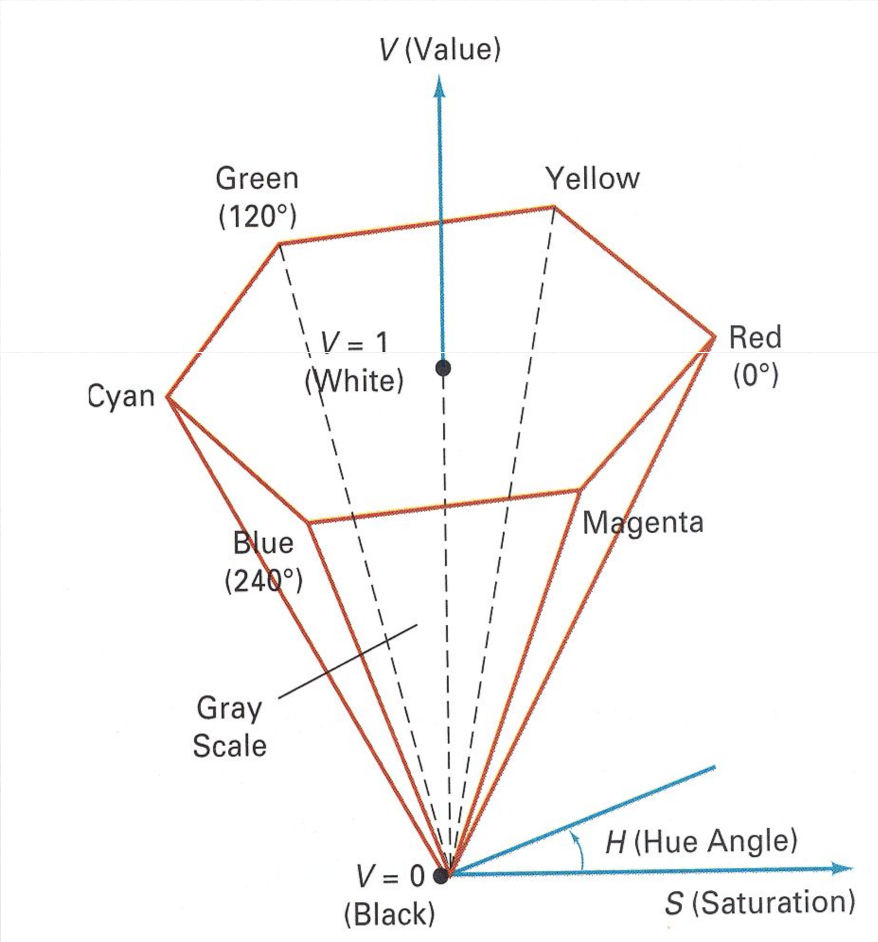

***Hue (H) is the angle around the vertical axis***

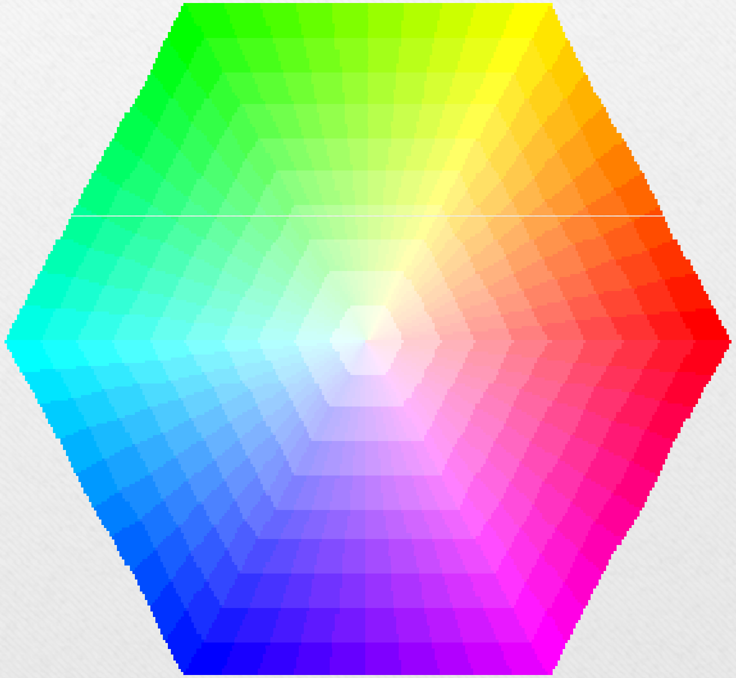
***Saturation (S) is a value from 0 to 1 indicating how far from the vertical axis the color lies***

***Value (V) is the height of the hexcone"***

# HSV color model

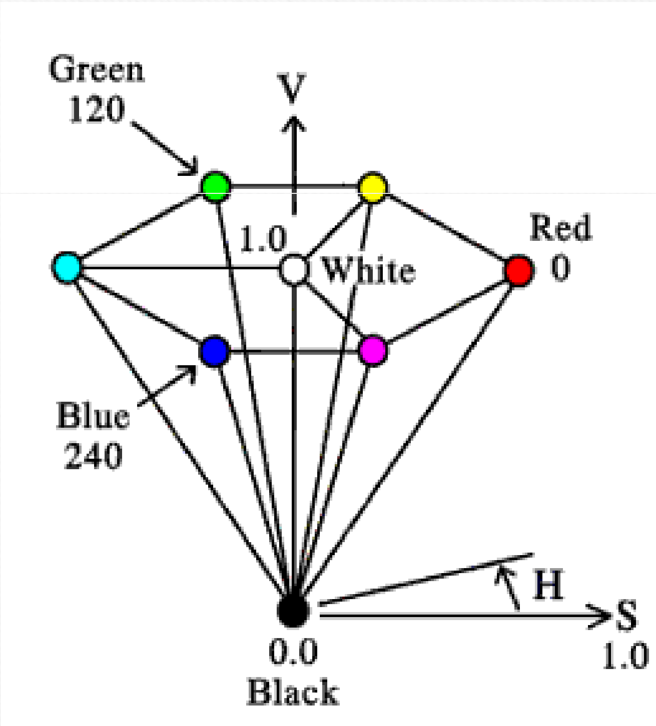
*A top-down view of hex cone*

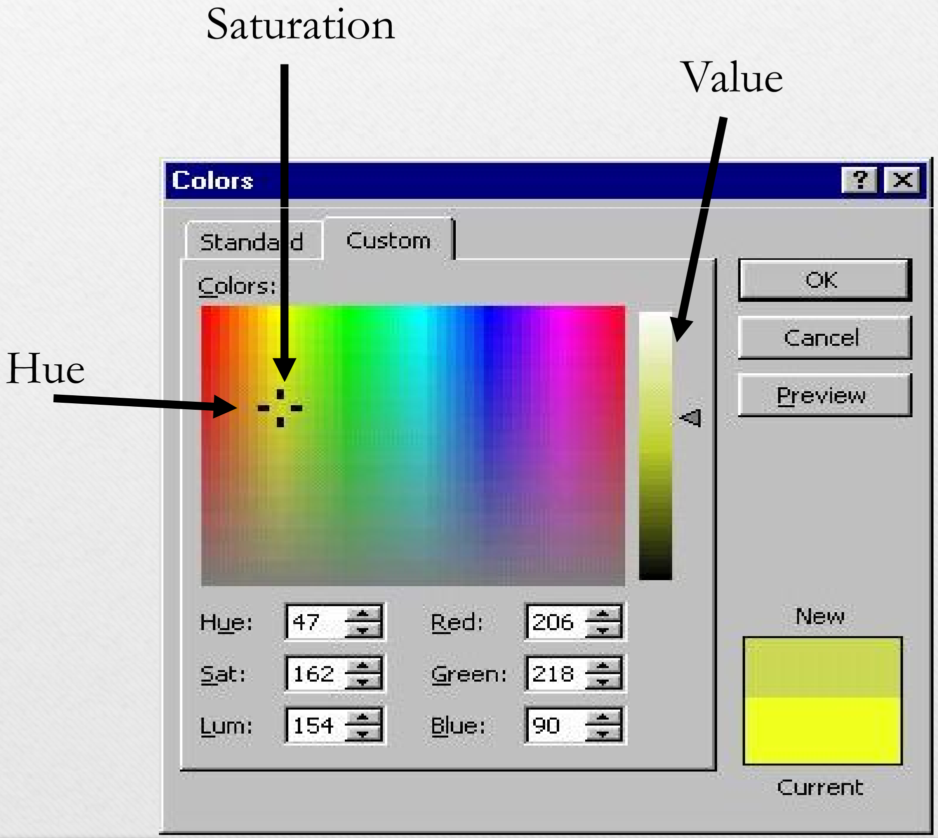


# HSV color model

- A more intuitive color space
  - H = Hue S = Saturation V = Value (or brightness)



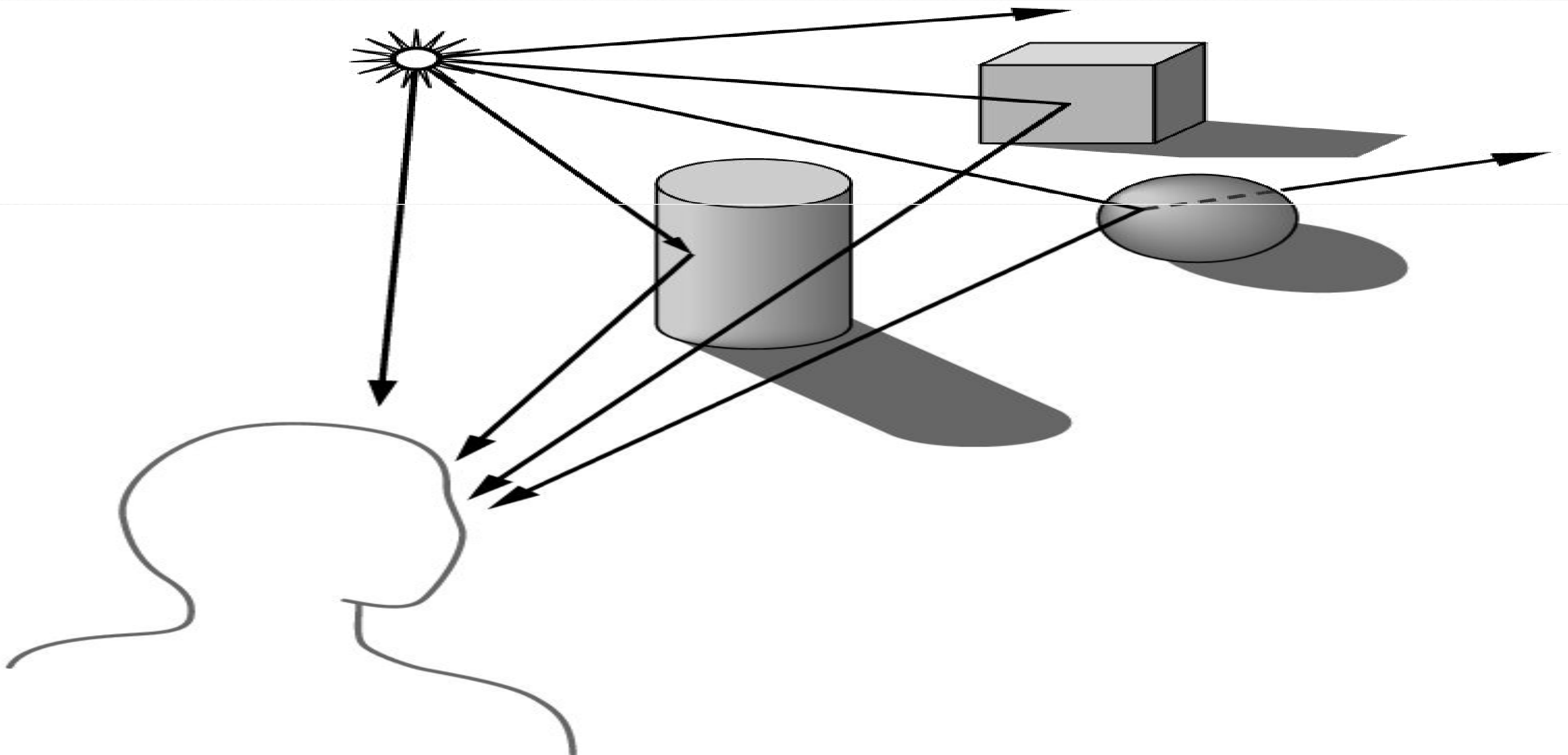



# *Illumination model*

# *and*

# *Shading*

- How do you make something look 3D?
- *Shading* that is appropriate for the lighting and the primary cue for 3D appearance.



# Light

- Light energy falling on a surface can be:

Absorbed    Reflected    Transmitted

They make an object visible

- The amount of energy absorbed, transmitted or deflected depends on the *wavelength or intensity of the light.*

- The characteristics of the light reflected or transmitted depends on:

  * The surface orientation

  * The surface properties of the object

  * Composition of the light source

  * Direction of the light source

  * Geometry of the light source

# Illumination model

- Illumination model is used to calculate the intensity of the light that we should see at a given point on the surface of the object.
- It is also known as Lighting model or Shading model. There are 2 types of illumination model:-

(1) **Local illumination model**:- Direct illumination of surface by light sources.

(2) **Global illumination model**:- All light or surface interactions for entire environment.

- A common simple illumination model is built from 3 components:-

  * Ambient light

  * Diffuse reflections

  * Specular reflection

# Ambient light

- Uniform from all directions (an object is illuminated uniformly).
- Not dependent on light, view or object direction, nor distance to anything else.
- La = Ka * Ia
- Where,

**La** is a reflected ambient light .

**Ka** is an ambient coefficient.

**Ia** is an ambient light.

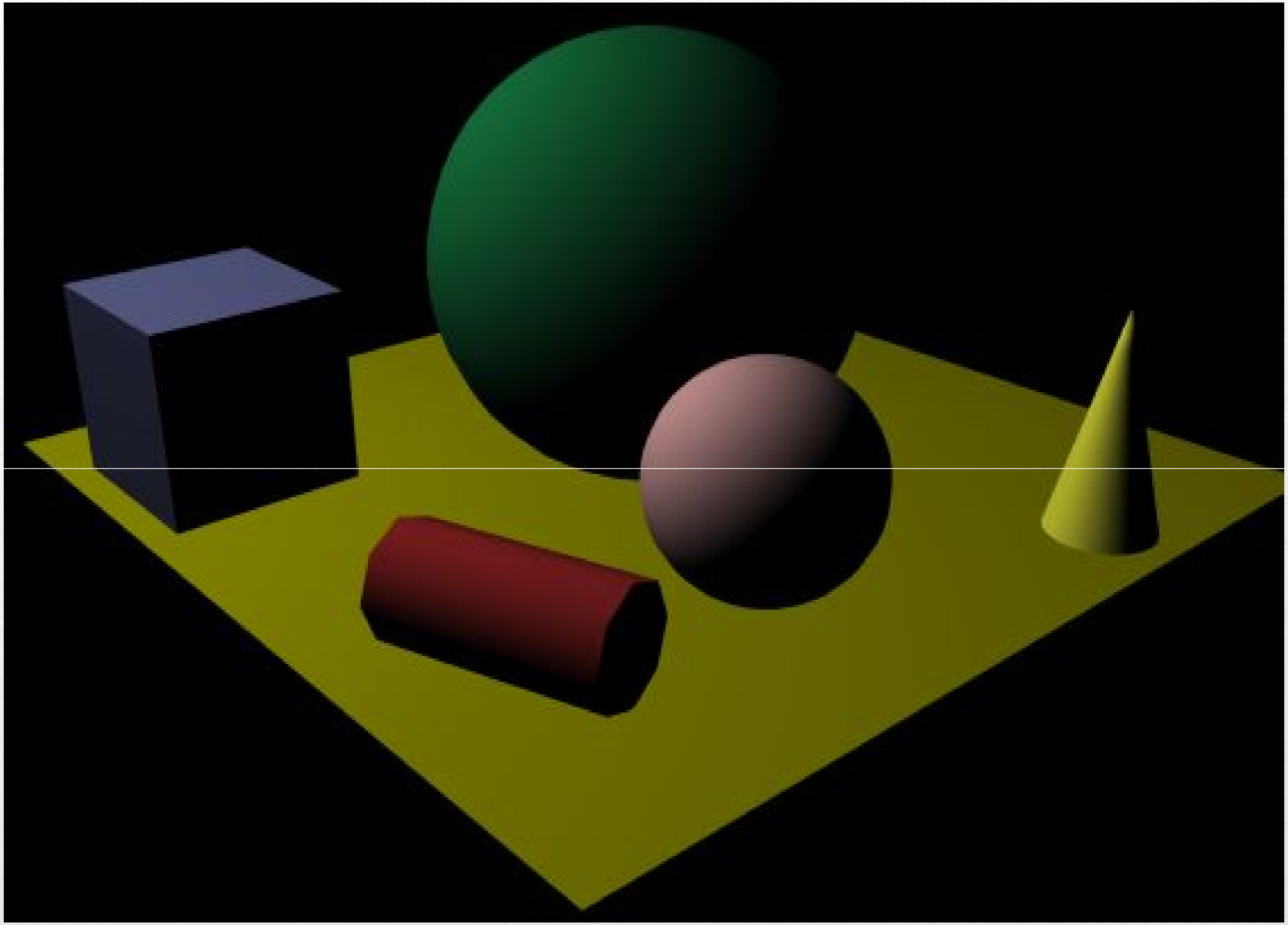
# Diffuse Reflection

- Diffuse reflections consider Point lights (Radiates equal intensity in all direction ) which comes from one direction to generate shading properties a change in color intensity across the surface of an object in relation to light sources.
- Depends on direction of light (L) and surface normal (N)

$$Id = Ip(L.N)$$

where, Ip is intensity of point light

-



# Specular reflection

- Component of reflection due to mirror-like reflection off shiny surface.
- Depends on perfect reflection direction, viewer direction, and surface normal

$$Is = Ip(R.V)^{n}$$

where n is Specular exponent, determining falloff rate.

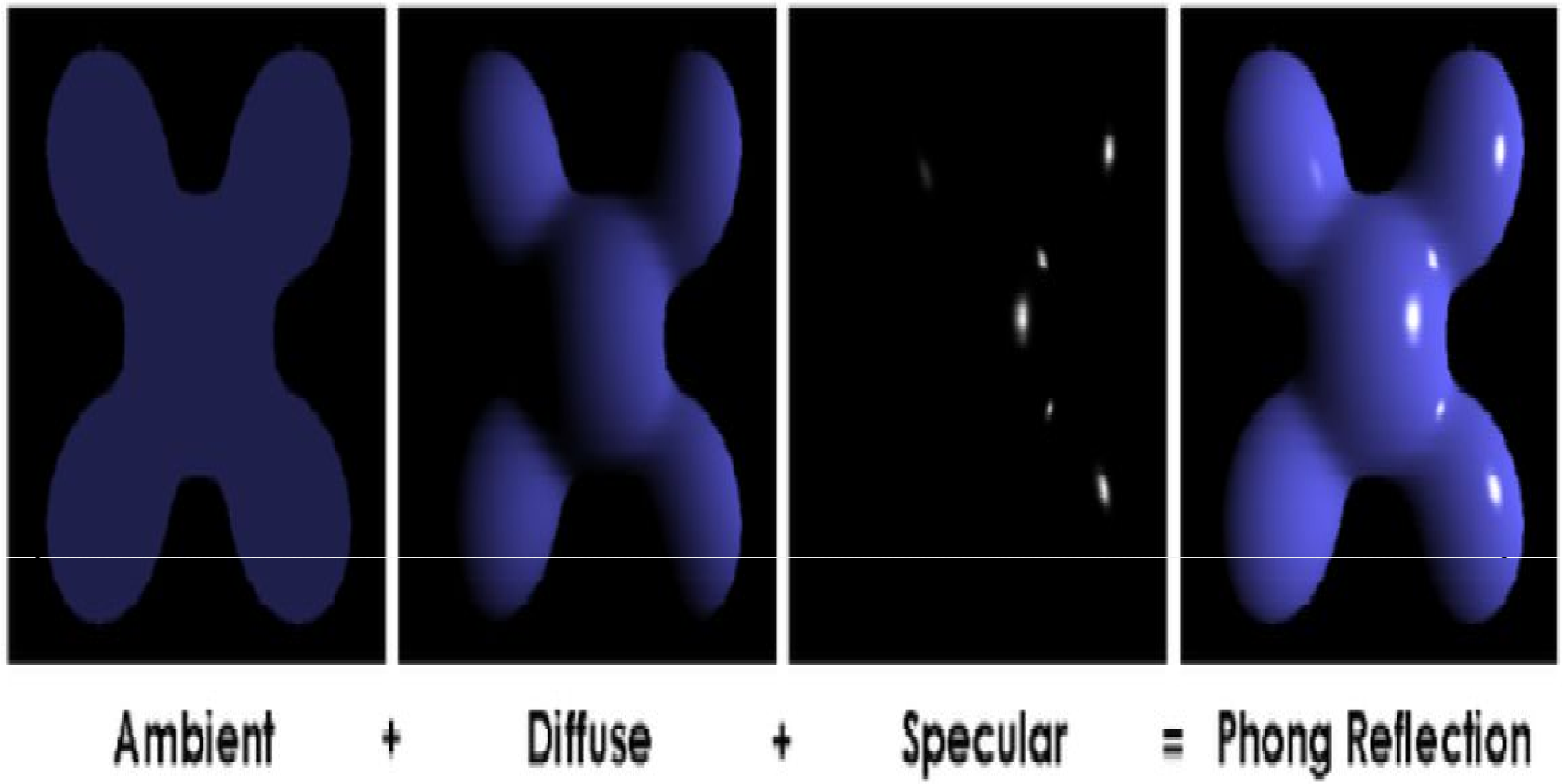
- Shininess

# Gouraud shading

- Gouraud shading, developed by Henri Gouraud in 1971, was one of the first shading techniques developed in computer graphics.
- Gouraud shading is most often used to achieve continuous lighting on triangle surfaces by computing the lighting at the corners of each triangle and linearly interpolating the resulting colors for each pixel covered by the triangle.
- Gouraud shading - Wikipedia, the free encyclopedia.htm

# Phong Shading

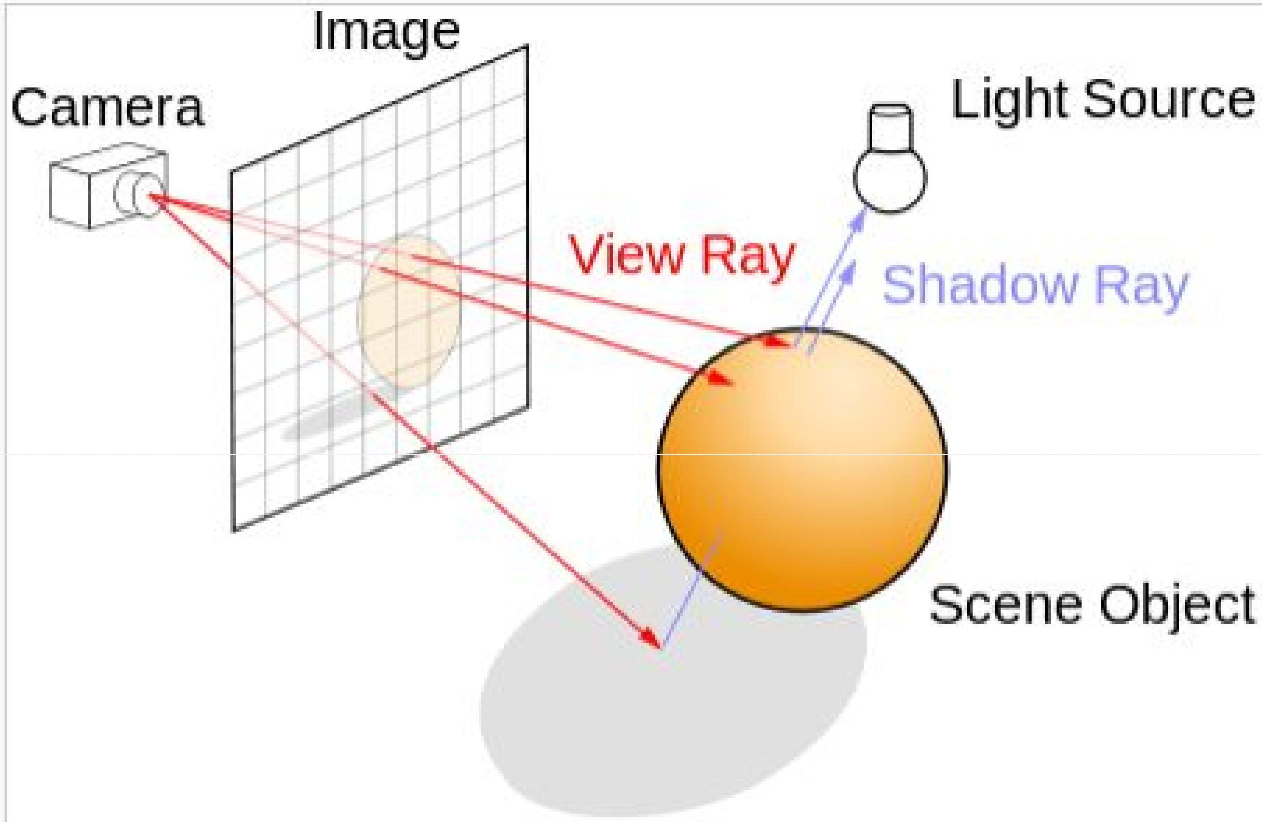
- It describes the way a surface reflects light as a combination of the Diffuse reflection of rough surfaces with the Specular reflection of shiny surfaces .




Visual illustration of the Phong equation: here the light is white, the ambient and diffuse colors are both blue, and the specular color is white, reflecting a small part of the light hitting the surface, but only in very narrow highlights. The intensity of the diffuse component varies with the direction of the surface, and the ambient component is uniform (independent of direction).

# Ray tracing

- It is a technique for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects.




The ray tracing algorithm builds an image by extending rays into a scene

# Thank You