[COLOR MODEL OF LIGHT (RGB)]

LECTURE 16

We know,

I = intensity of light

There, is three things to light :
i) brightness
ii) color
iii) saturation (colorness)

In Real Life :

$$I \begin{cases} \to I_R \\ \to I_G \\ \to I_B \end{cases}$$

Whoever holds the highest value will dominate.

(There's two important things to consider)

1) Monitor, color (emits light as a source) → RGB

2) Paper print, pigment (absorps lights) → CMY

(white)

$W = R + G + B$

$C = W - R \longrightarrow R+G+B-R \longrightarrow (G+B)$

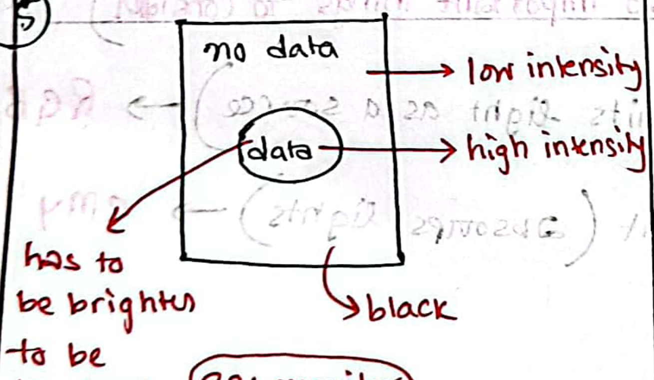
$M = W - G \longrightarrow R+G+B-G \longrightarrow (R+B)$

$Y = W - B \longrightarrow R+G+B-B \longrightarrow (R+G)$

⊛ objects that emit light	⊛ objects that can't be seen without light
Additive Color Model (RGB)	**Subtractive Color Model (CMY)**
1. Refers to RGB	1. Refers to CMY
2. Works on the principle of emission	2. Reflection (seen through reflection)
3. Additive : increase in color causes effective increase in intensity	3. Subtractive : increase in color causes effective decrease in intensity.
4. For making monitor color	4. for making paper color.

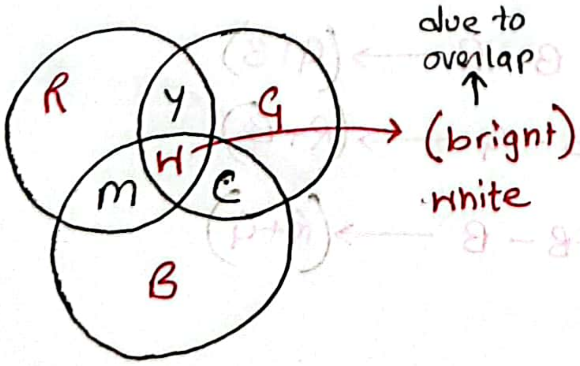
Additive Color Model (RGB)

5.



eg: monitor

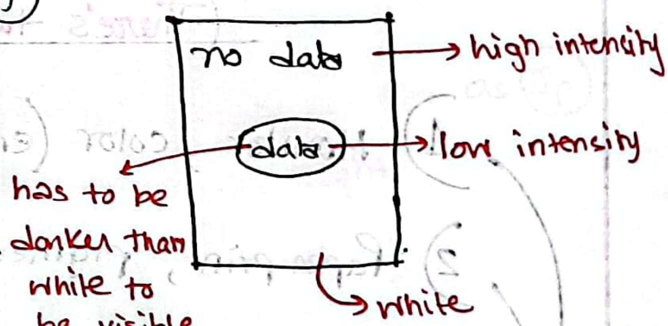
6.



∅ additive so highest intensity or brightness at white

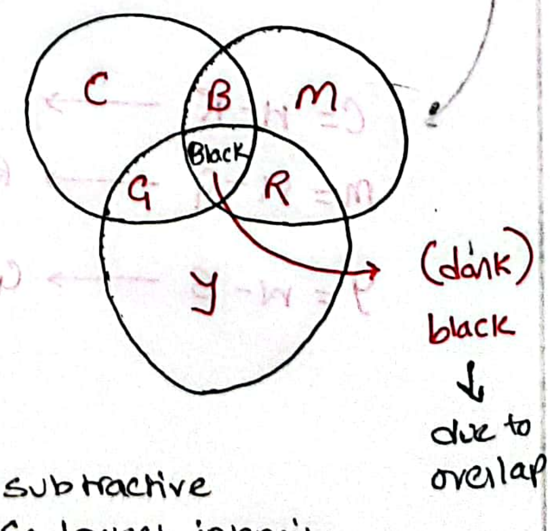
Subtractive Color Model (CMY)

5.



eg: paper

6.



∅ subtractive so lowest intensity at black.

$$R, G, B \underset{(1-R,\ 1-G,\ 1-B)}{\overset{(1-c,\ 1-m,\ 1-y)}{\longleftrightarrow}} C, m, y$$

$(0.7, 0.6, 0.8)$ $\quad$ $(0.3, 0.4, 0.2)$

$R = 1 - 0.3$
$G = 1 - 0.4$
$B = 1 - 0.8$

← by subtracting RGB or cmy from 1 we can move to either side →

$c = 1 - 0.7$
$m = 1 - 0.6$
$y = 1 - 0.8$

Example for pure (RED)

RGB		cmy
$(1, 0, 0)$	$\longrightarrow$	$(0, 1, 1)$

only red with high intensity

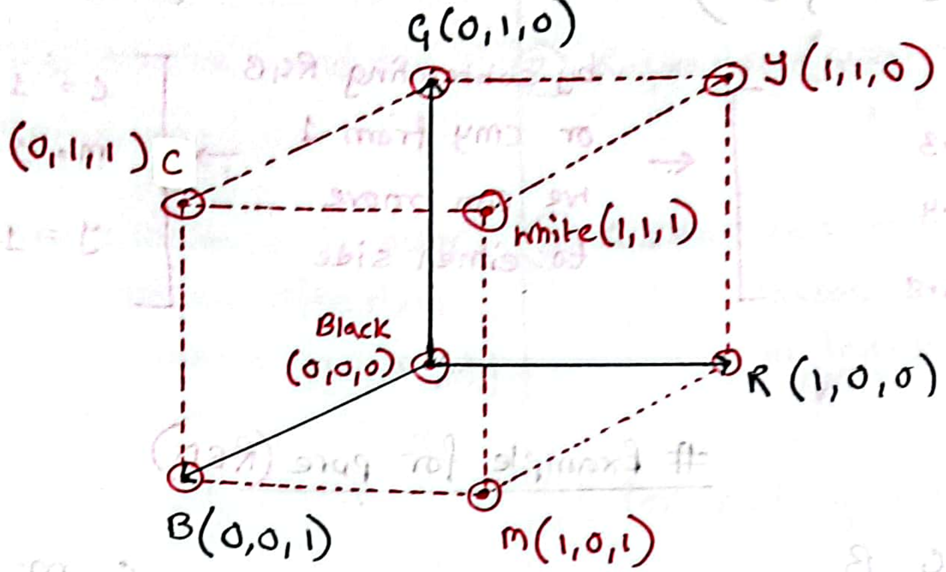
most common reflector of red, So high red color.

Ø

R = Red
G = Green
B = Blue

C = Cyan
M = Magenta
Y = Yellow

Ø RGB color Cube Model



Ø each axis is 8 bit : (0 ~ 255)

So this model can represent : $2^8 \times 2^8 \times 2^8$ = (16 millon colors)

Ø (32 bit color)

RGBA

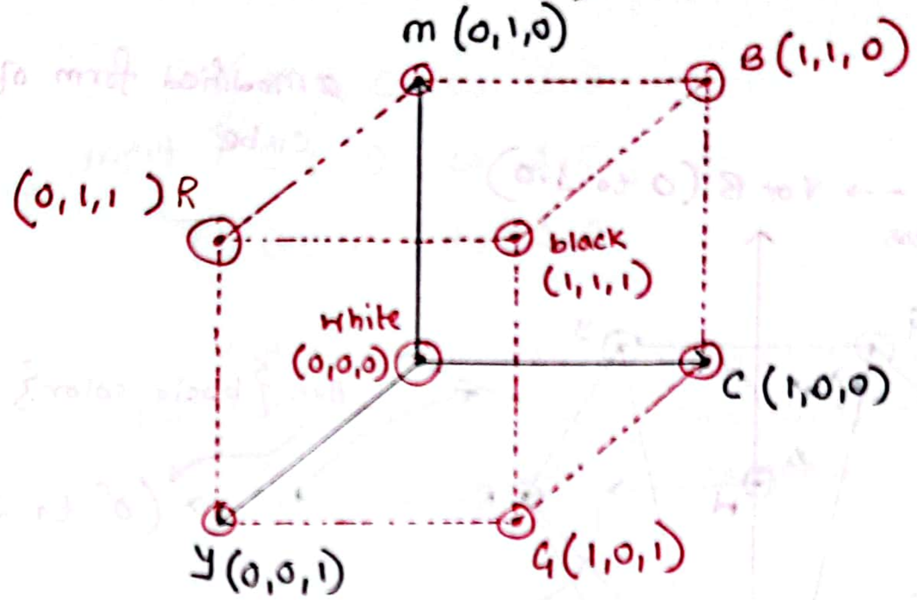
↳ Alpha channel

(helps to create more convincing gradients, shadows & transparenencies & can support 4 billion color combinations)

↳ can be used to correctly produce real life object.

Ø Now, for CMY cube, the diagonal opposites will be switched.

∅ (CMY Color Cube)



(R, G, B) ⟶ is theoretically a floating point value

(*) Good for hardware

(*) But no one uses RGB model in software because it doesn't present us with the following infos

- ∅ Brightness ?
- ∅ Hue ?
- ∅ Saturation ?

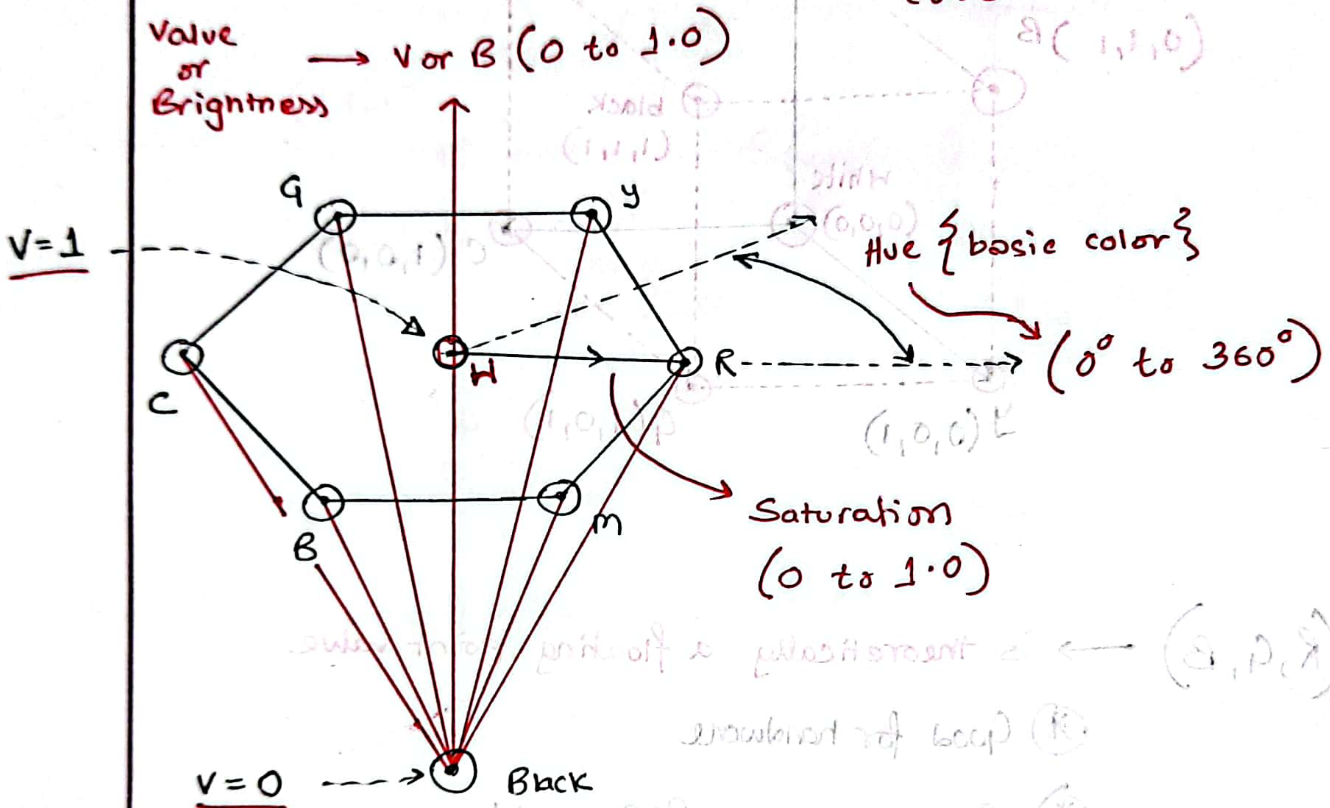
∅ Hence for representation we use :

(i) HSV / HSB
(ii) HSL / HLS

models.

(i) Ø HSV / HSB Color Model : (Representation model)

Ø Modified form of RGB cube



eg:

RGB

(0.8, 0.6, 0.2)

↓

(0.4, 0.3, 0.1)

→ This two RGB values represent the same color, which is based on the mutual proportion of the 3 values

But, the upper one has a higher brightness.

(RGB to HSV: conversion)

input
- R : 0 to 1.0
- G : 0 to 1.0
- B : 0 to 1.0

output
- H : 0° to 360° → Hue (basic color)
- S : 0 to 1.0 → Saturation
- V : 0 to 1.0 → Brightness.

where,

$V = \max(R, G, B)$ → the value of V will depend on the maximum in between R, G & B

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)}$$

H = will depend on dominance,

for code (P.T.O) →

Next Lecture (17)