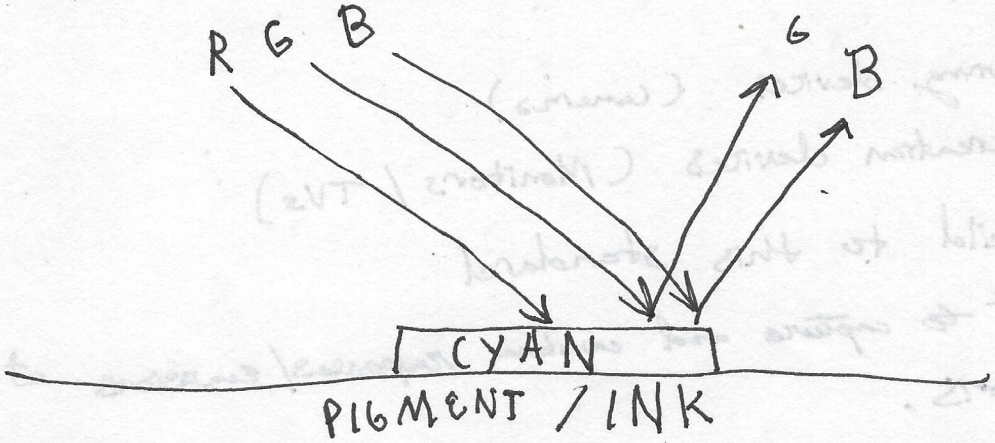
(10) Let's talk about Ink

SUBTRACTIVE Color space

When you look at the color red on a page, you do not see Red light shining off that object, Ambient light, (white light), is shining in and what being reflected is red.

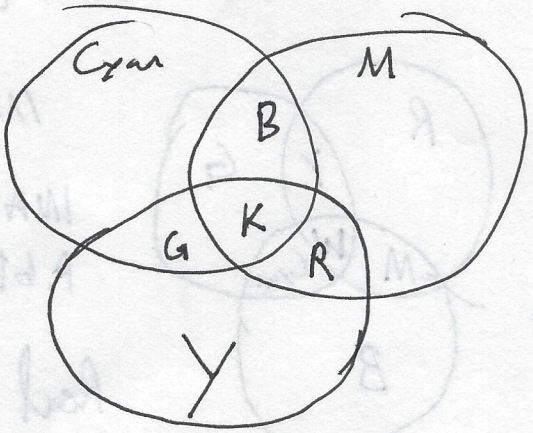
Say we have Cyan on a piece of paper.



Green and Blue is reflected, that combo is cyan.

This called subtractive because we are taking meaning Light and removing different colors from it.

Black has its own cartrage as we don't want a crummy black



Color TERMS

Brightness / Intensity / value = "How much" Light There is

(How bright the light is, how intense the light is)

HUE = Dominant Color — is it greenish, is it reddish, is it purplish

SATURATION = Purity / STRENGTH of color

(The tint of a color) Pastel

The Hue and the SAturation define the chromaticity of the color. The value is only the intensity of the color.

The color information is in the Hue and saturation.

1 numbers for intensity,
2 numbers for color space location.

Interms of the CIE, say you want to make a new color, thus you need a combination of RGB, As in a Linear Combination

AMOUNTS of Reference CIE primaries needed to form a color are called The Tristimulus values

X, Y, Z - sometimes denoted

(2)

We characterize a color By

$$\frac{X}{X+Y+Z}, \frac{Y}{X+Y+Z}, \frac{Z}{X+Y+Z}$$

So how do we get X?

$$X = \int_0^{\infty} \underbrace{I(\lambda)}_{\text{Spectral Power Distribution}} \cdot \underbrace{X(\lambda)}_{\text{color-matching function "RED"}} \, d\lambda$$

"choosing Red"

Define this as the actual red

A monitor can't contain all the colors we see.

51:44

choose color mode CYMK

to get a good approx of what you see on screen to what you see printed

COLOR SPACES

RGB - matlab uses

CMYK - photoshop

HSV, HSI - intensity

(Hue saturation value, cylinder cap colors)

RGB - WEB SAFE (00, 33, 66, 99, CC, FF)

17 colors

6 levels of Red
6 levels of Green
6 levels of Blue

→ 216 colors

→ # 00 00 00 BLACK
R G B

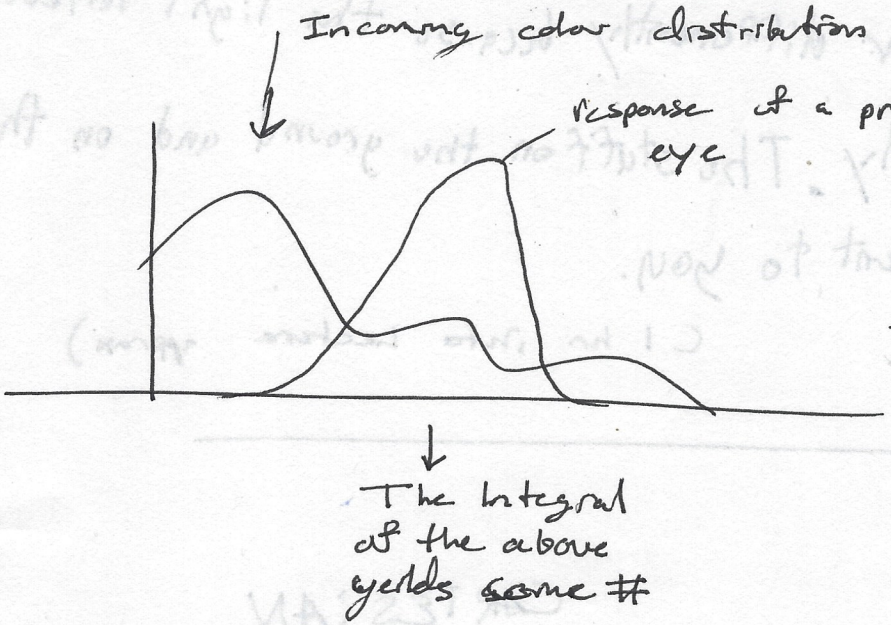
(255) # FF FF FF White
in each R G B

Some freaky stuff,

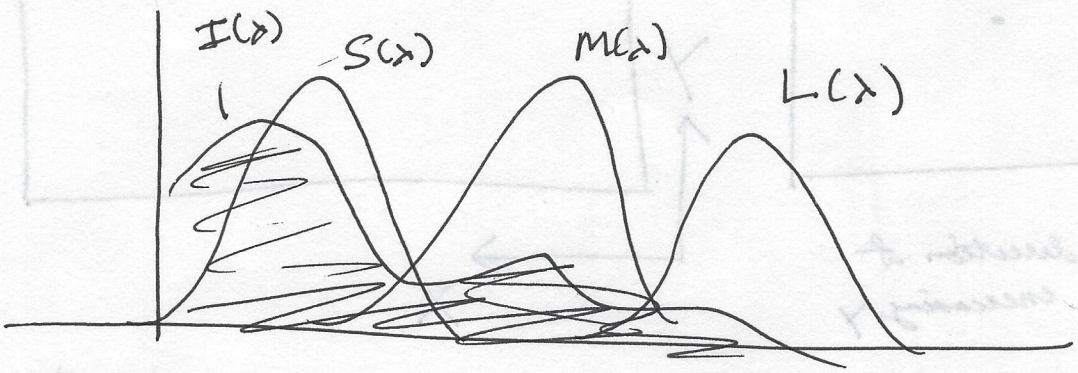
So you only have 3 types of cones in your eye.

~~In theory there~~

In theory, you can have different ambient color that is perceived by you to be the same color value.



So say for the three cones S, M, L



The true color is this complicated spectral combination, I(λ)

What you are getting in your visual system are three numbers, #, #, #

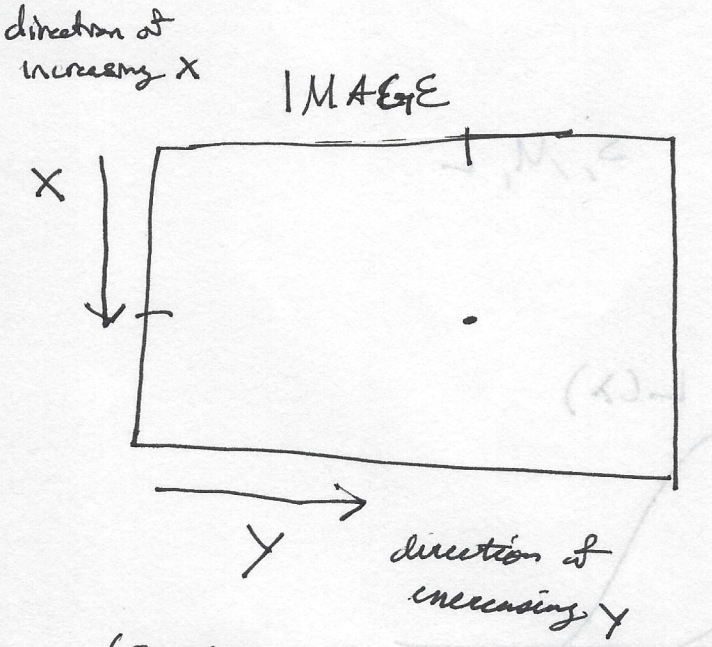
(14) These three numbers are like a projection of that complicated thing above ($I(\lambda)$) onto the spectral responses of your cones.

So there can be a ton of colors (perhaps infinitely many) that you perceive the same way

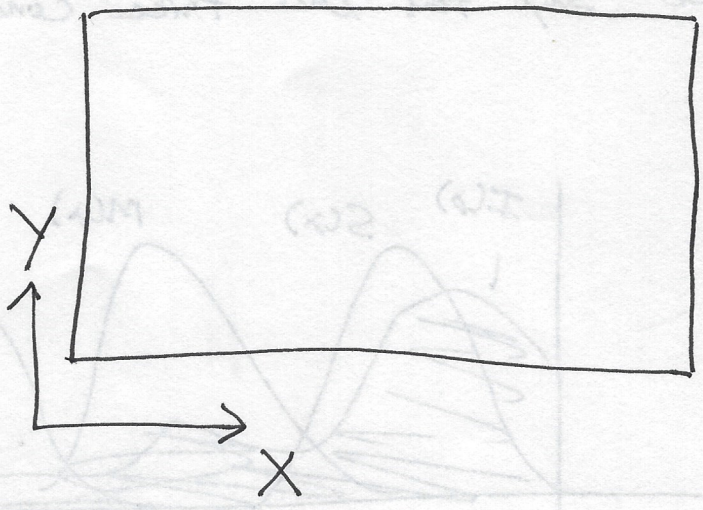
Ex, The lights on the ceiling can appear the same, but the objects can appear differently because the light reflects off objects very differently. The stuff on the ground and on the Table can look very different to you.

I want to see this! (1 hr into lecture approx)

MATLAB IMAGE



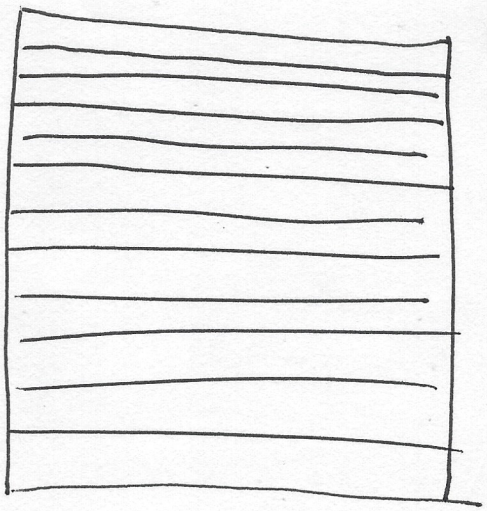
CARTESIAN



(200, 300)

So do some mental gymnastics to ask matlab the correct thing.

Say we want to create an Image in matlab.



0 ——— 0

255 ——— 255

1 x 256

$$\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 255 \end{bmatrix}_{256} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

$= 256 \times 256$

→ It has constant rows

~~Lecture 2~~ Lecture 2 matlab

Reading Images in matlab

```
im = imread('1280px-EN_Spectrum.svg.png');
whos
```

Size	Class
685 x 1280 x 3	uint8
↑ 3 color chenels	↑ unsigned 8 bit integer

```
>> imshow(im)
>> [x,y] = ginput(1)
```

disconnect how matlab thinks of an Image vs you (how you think of an Image)

ginput gives us something in cartesian coordinates

```
>> im(x, y, :)
```

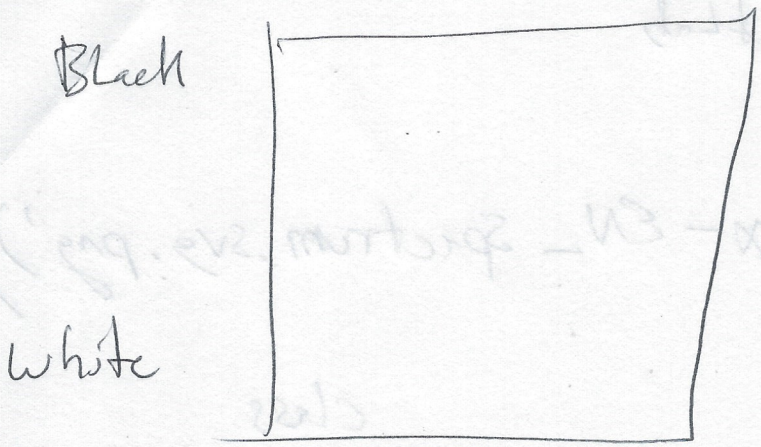
↑ ↑ → [R G B]

These must be ints

(Does the RGB match where I clicked?)

(2)

make an Intensity Image



1 x 256

[1 1 ---- 1]

$$\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 255 \end{bmatrix}$$

= 256 x 256

Transposed

```
im = [0:255]' * ones(1,256);
```

```
whos
```

```
imshow(im)
```

→ Looks wrong

This Imshow command doesn't know what kind of Image you are giving it

```
imshow(im, [])
```

color in this.

↑ I want you to scale the darkest Image to ~~white~~ Black and the Lightest colors to white

This may not look like a constant ramp,

too much white or too much black

color intensity is mapped ~~into your~~ logarithmically

into your visual system

---

to generate color image

>> im(:,:,3) = ones(256,1)*[0:255];

we are switching the order and assigning that ~~blue~~ new thing to the blue channel

whos

im 256x256x3

Green channel is all zero

Red ~~blue~~ channel is top to bottom ramp

blue channel is left to right ramp

(we kept im but changed that color slot)

>> imshow(im, [])

This gets wierd results

try

>> imshow(uint8(im))

turns im into 8-bit int for display

(4) we have no red to all red
and no blue to all blue
and we get a color wash in the middle

if the image is floating point or Double
imshow doesn't know what to do with it

uint8(100) + uint8(200)
= 255 (saturates, uint8 saturates,
maybe do operations as a
double first)