Lecture 4:
Light + Color

Administrative

HW0 is out, due 1/27

Office Hours and **Discussion Sections** start this week
See Google Calendar for schedule and info

Debugging

Learning to debug is an important goal for this class!

Don't: Debug via Piazza

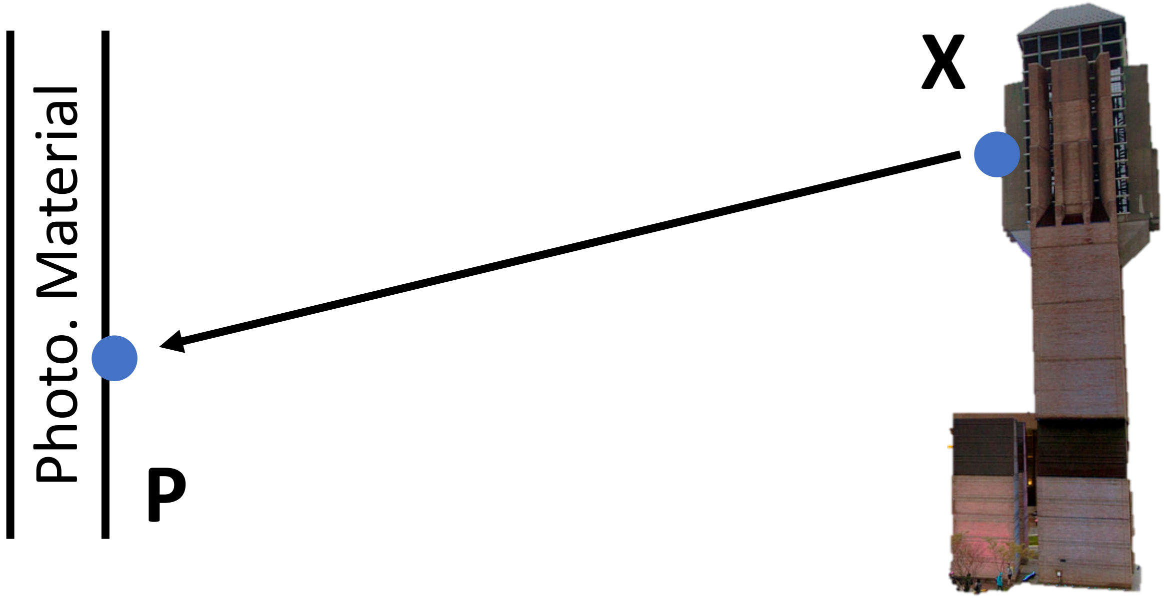
Do:

- Google to find documentation for numpy / other libraries
- Figure out how numpy / other APIs work by poking around in an interactive shell, small standalone script, or notebook
- Try to isolate and reproduce errors in small standalone cases (unit testing!)

Recap: Projection

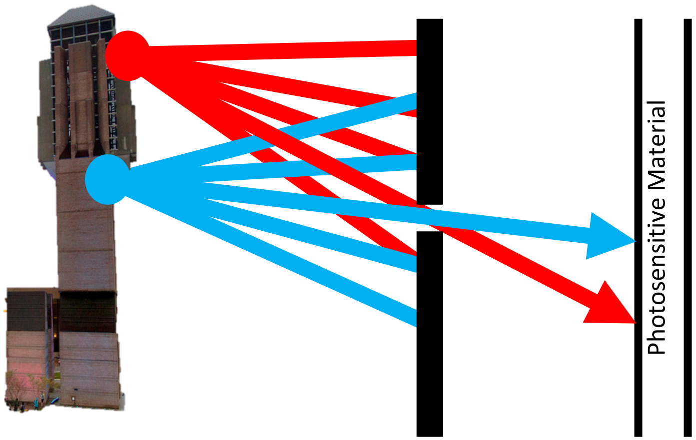
Image → $\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{X}$ ← *World*

Intrinsic *Extrinsic*



Recap: Lenses

Pinhole Model



Mathematically correct
Not quite correct in practice
Reasonable approximation

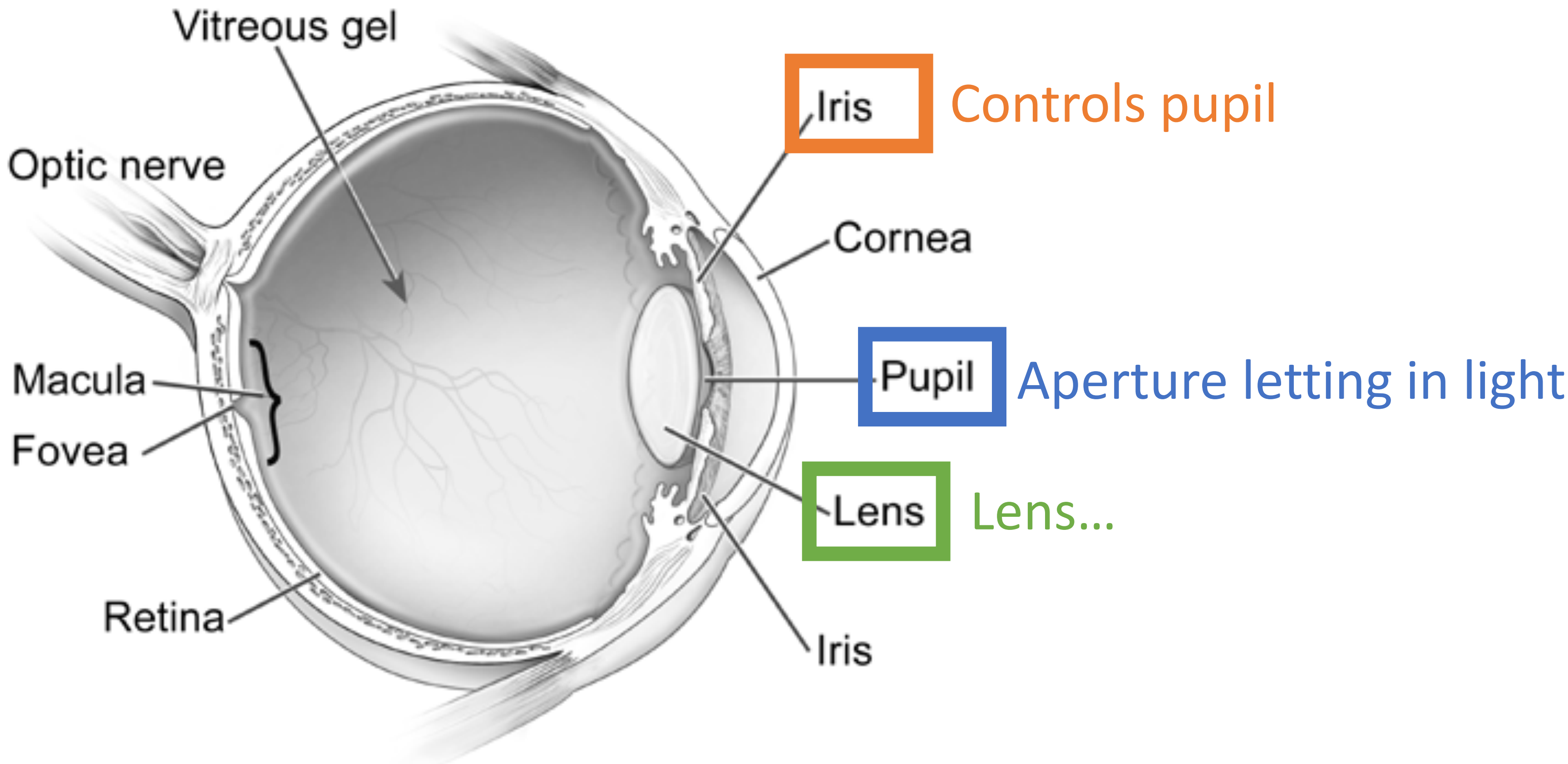
Reality: Lenses

Necessary in practice
Introduce complications
Complications fixable

Today

- A little bit about light and how you represent it
- A little bit about lighting and how it works

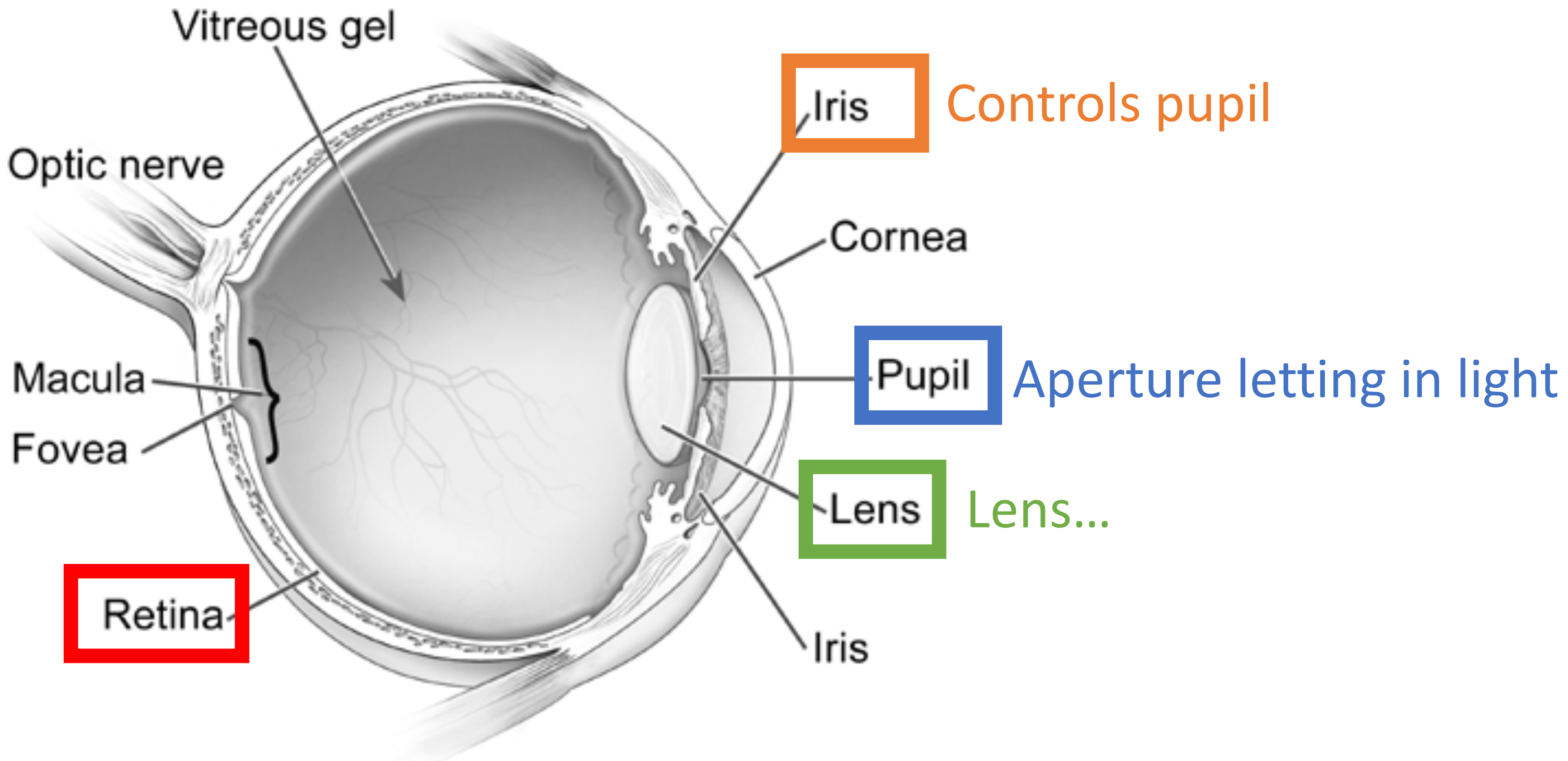
Your Very Own Camera



Where's the film/CCD?

Slide Credit: NIH

Your Very Own Camera



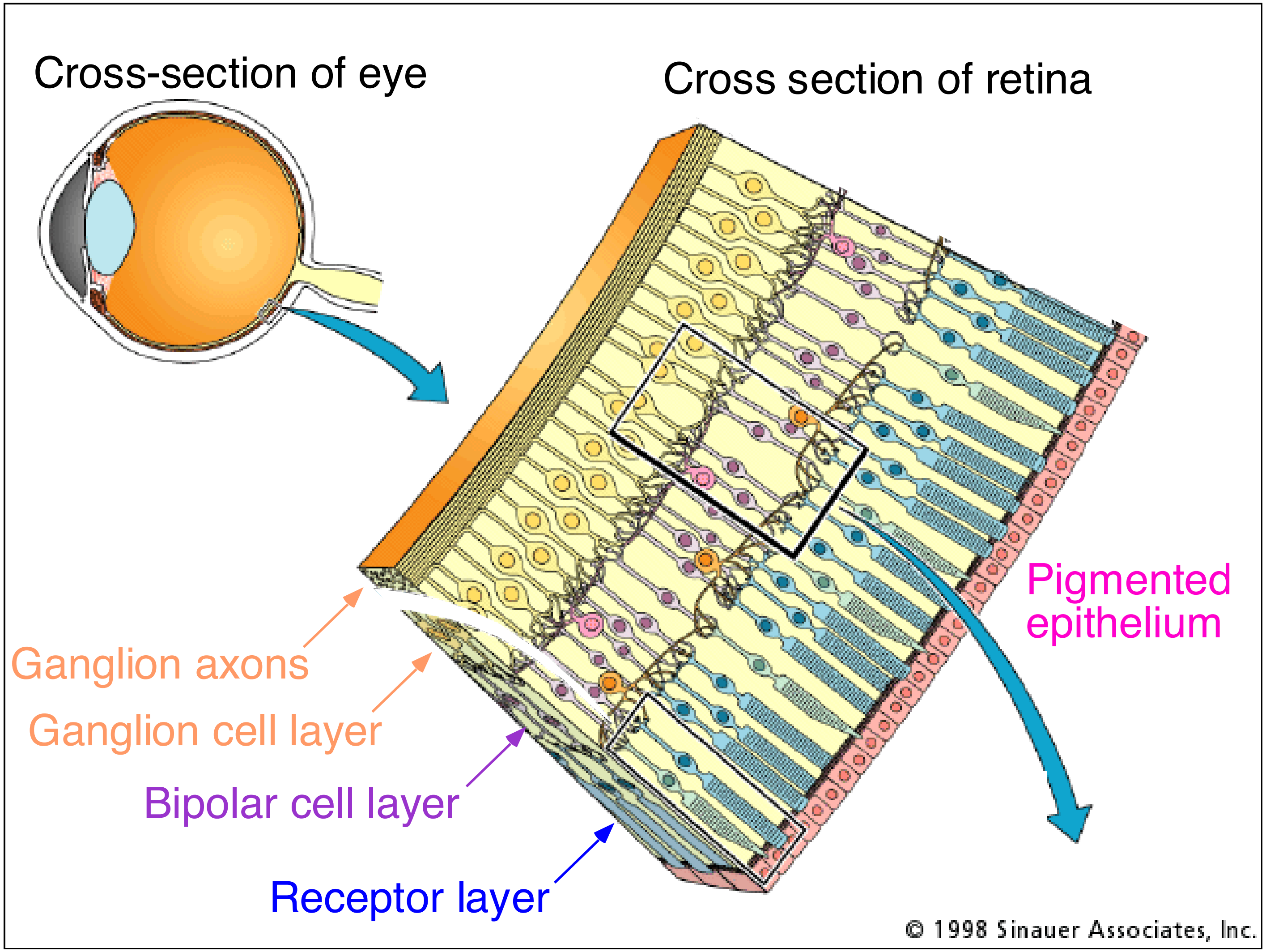
Where's the film/CCD?

Slide Credit: NIH

Demo Time

https://bit.ly/2lNJ3xc

What is the Retina made of?

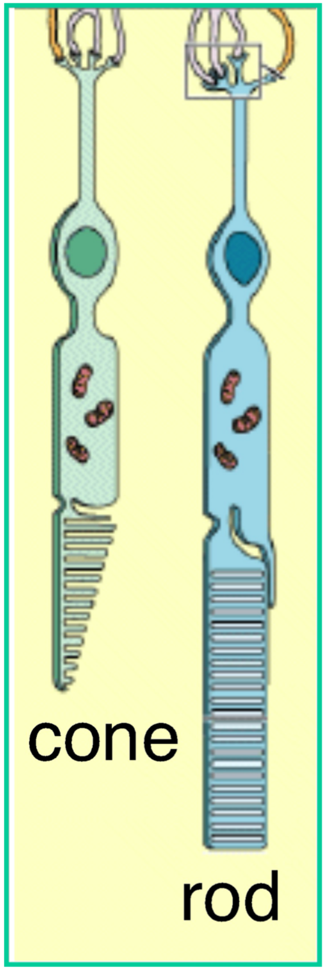


Slide Credit: J. Hays

Two Types of Photo Receptors

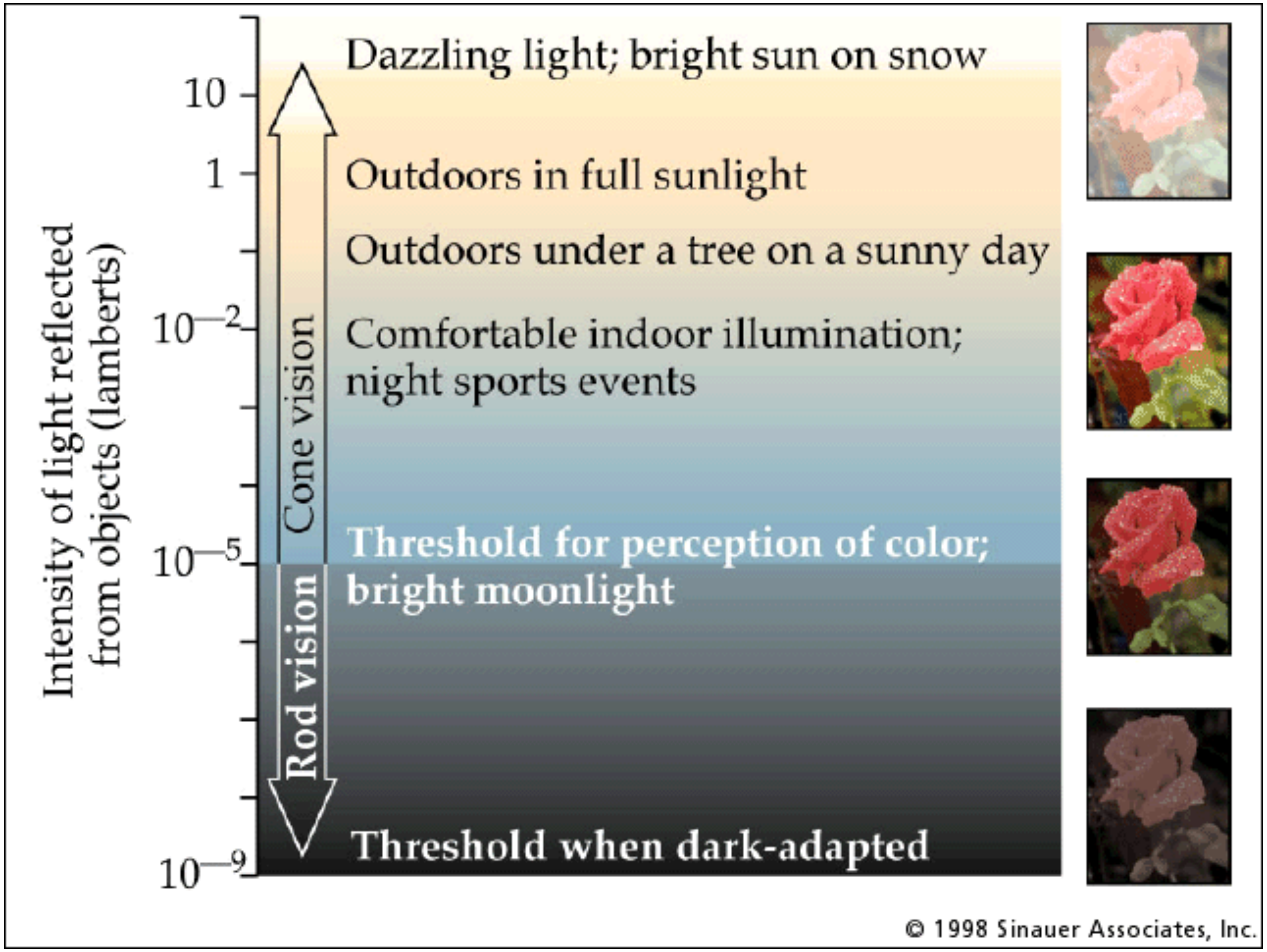
Cones
- cone-shaped
- less sensitive
- operate in high light
- color vision

Rods
- rod-shaped
- highly sensitive
- operate at night
- gray-scale vision



Slide Credit: J. Hays

Rod / Cone Sensitivity



Slide Credit: J. Hays

Rod / Cone Distribution

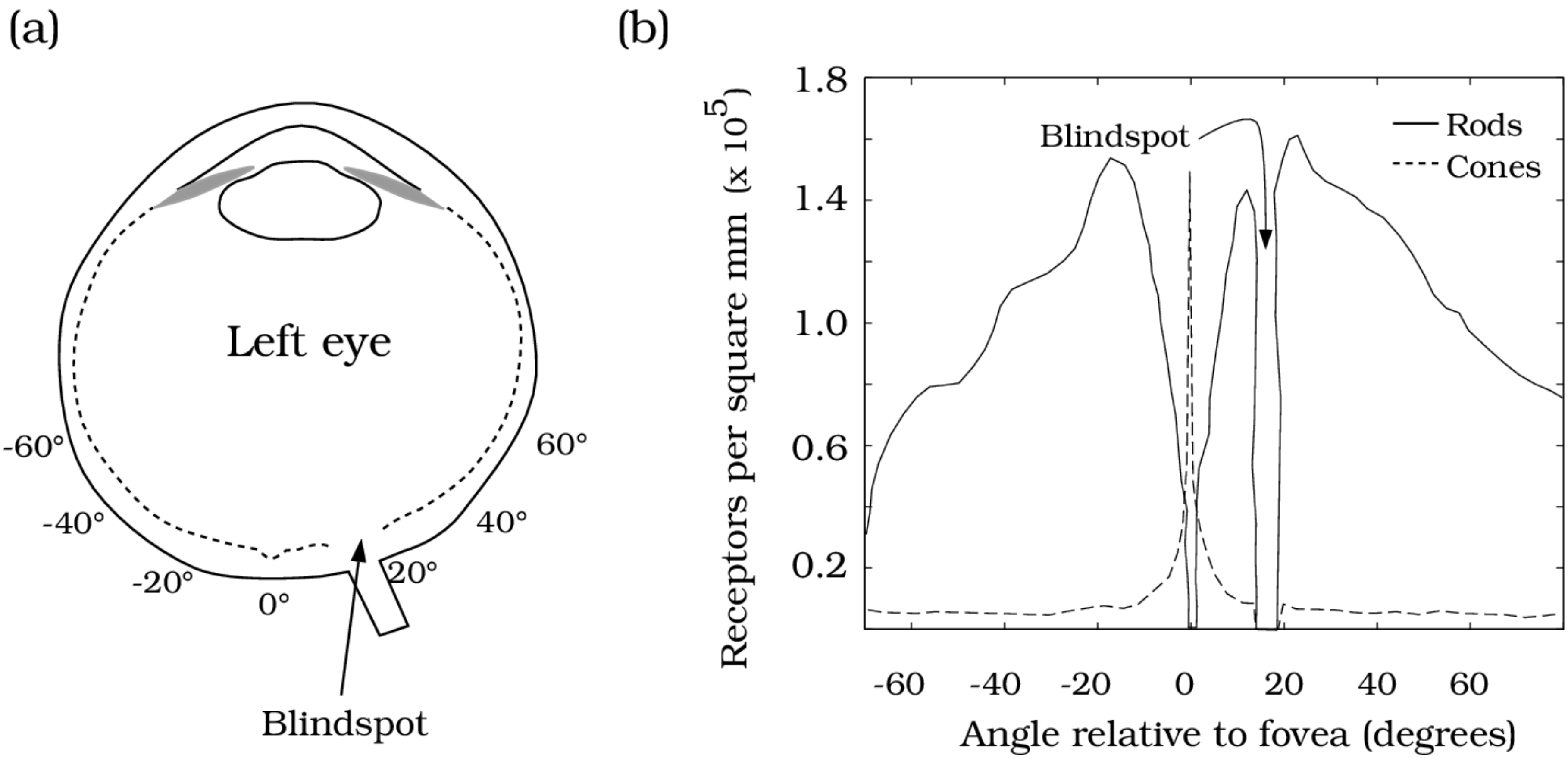
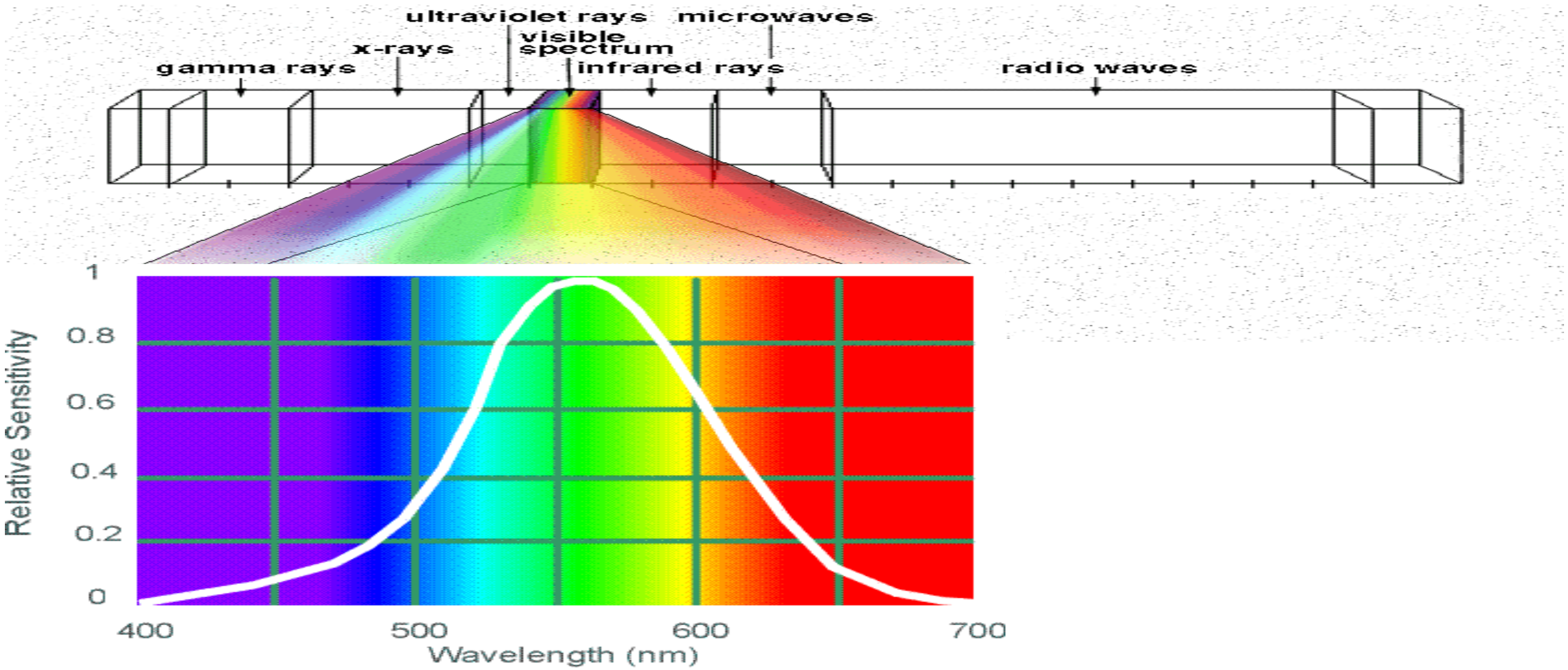


Diagram Credit: B. A. Wandell, *Foundations of Vision*

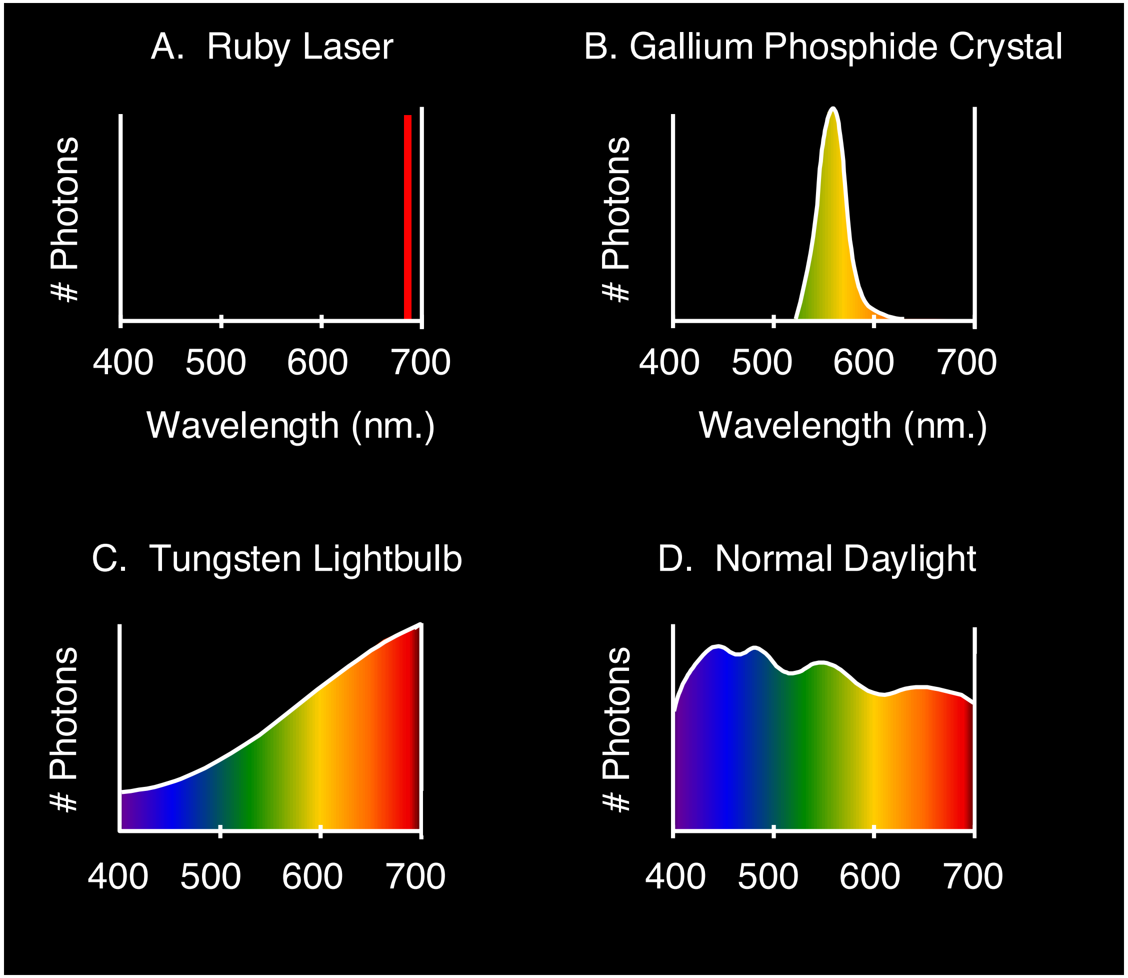
Electromagnetic Spectrum



Why do we see light in these wavelengths?

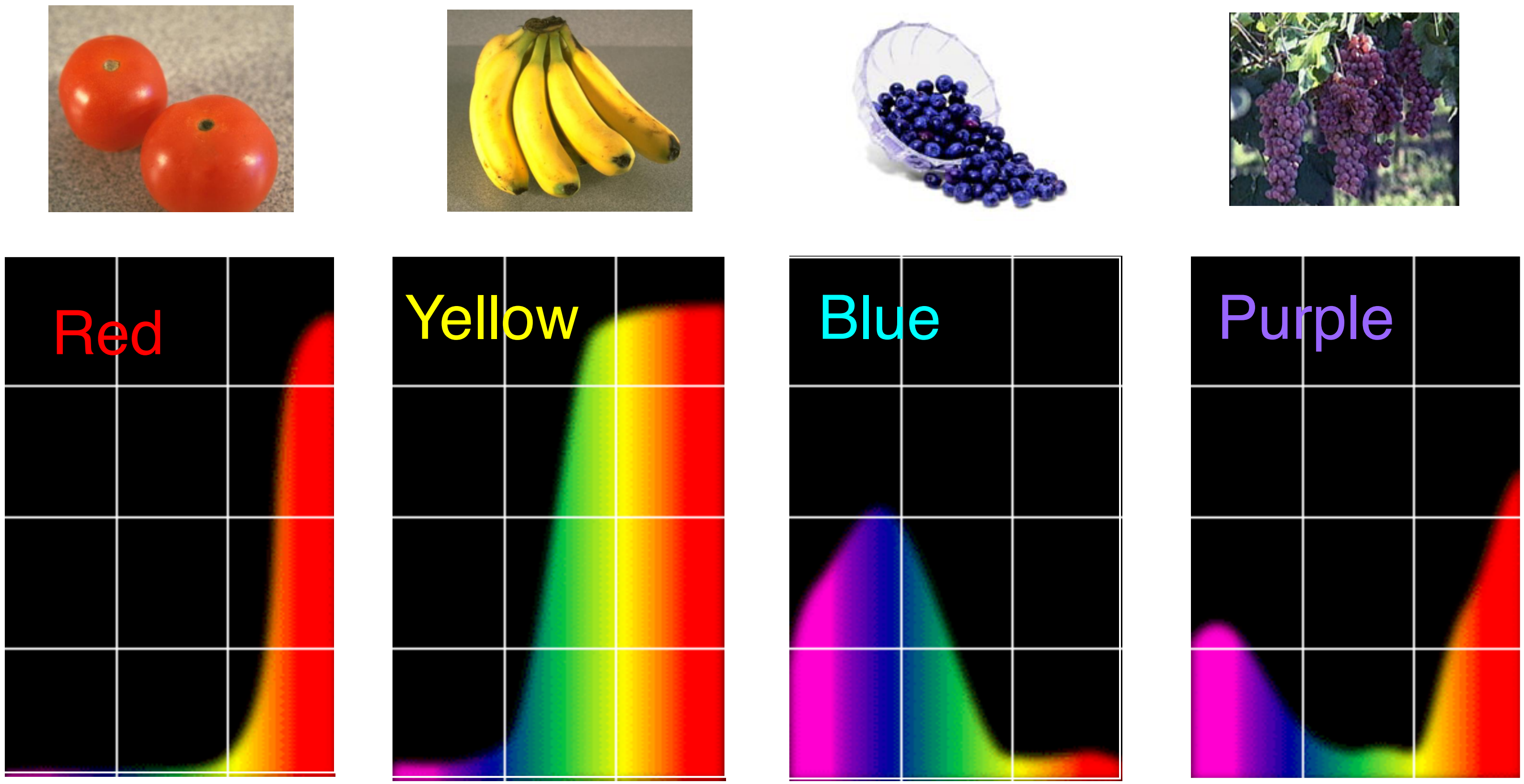
Slide Credit: J. Hays

The Physics of Light



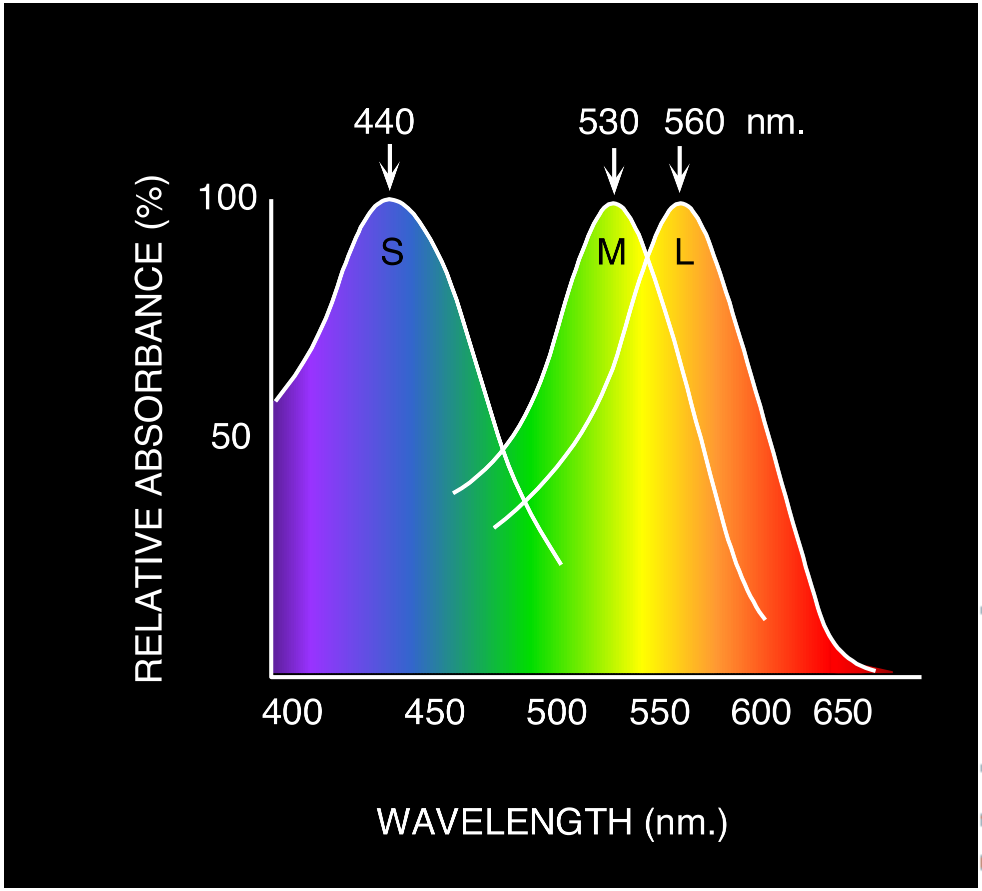
Slide Credit and Copyright: S. Palmer

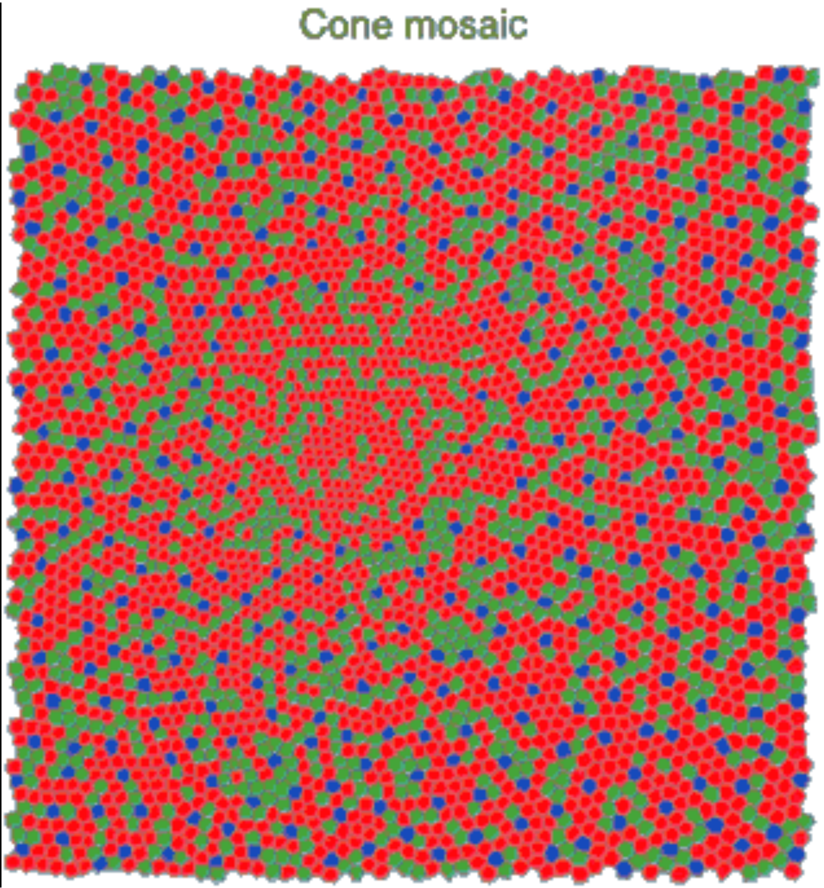
The Physics of Light



Slide Credit and Copyright: S. Palmer

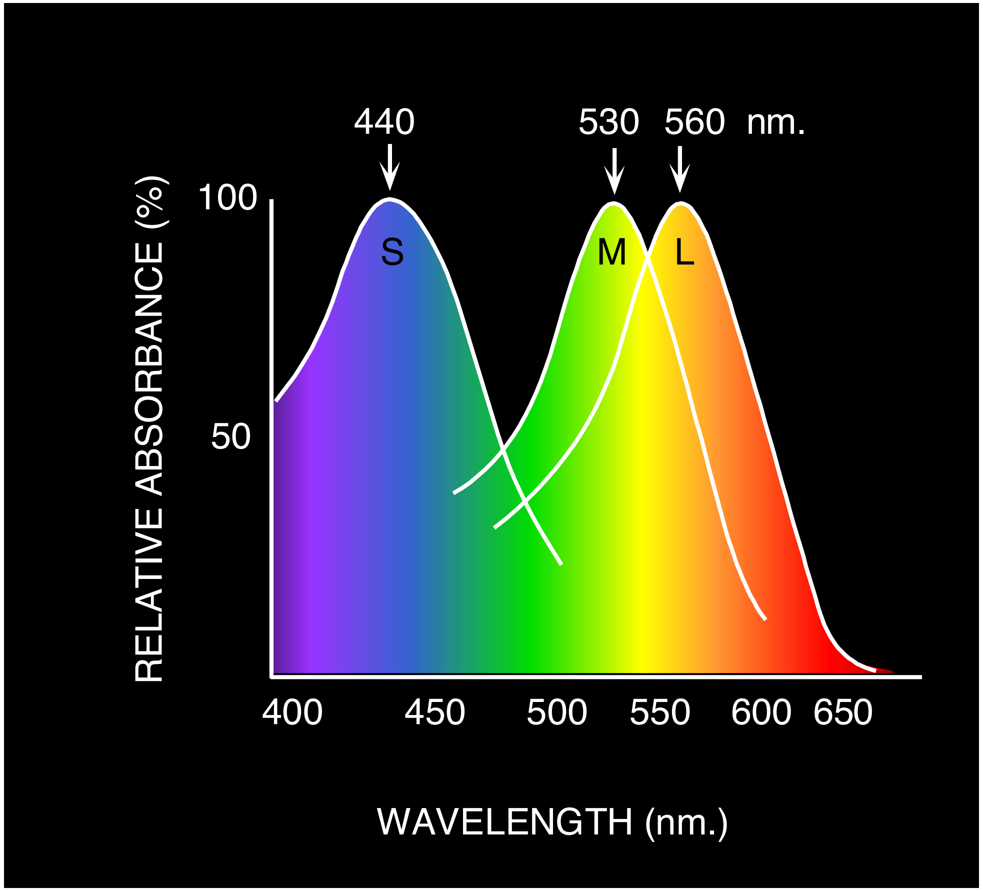
The Physics of Light

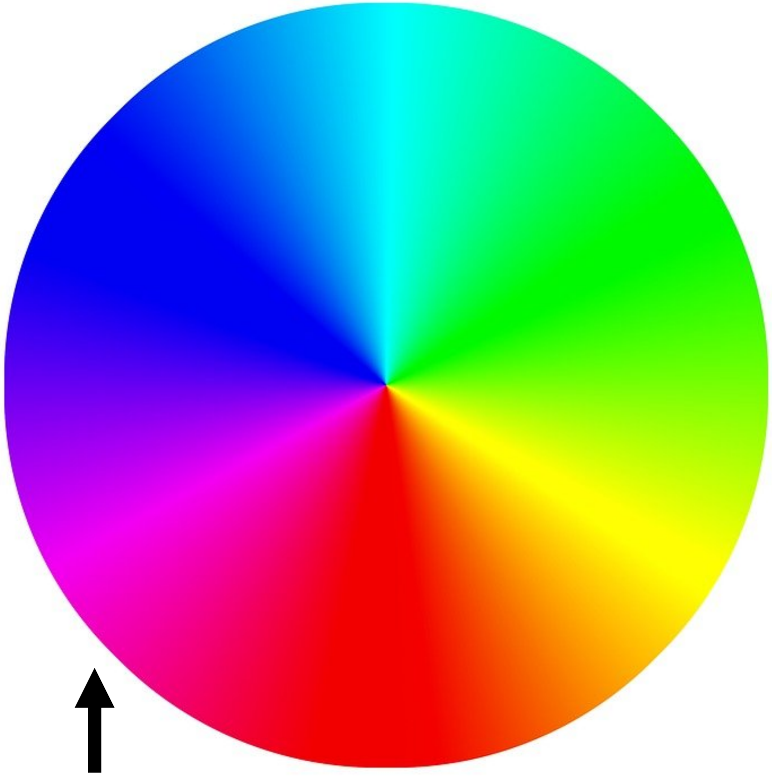




Slide Credit and Copyright: S. Palmer

The Physics of Light

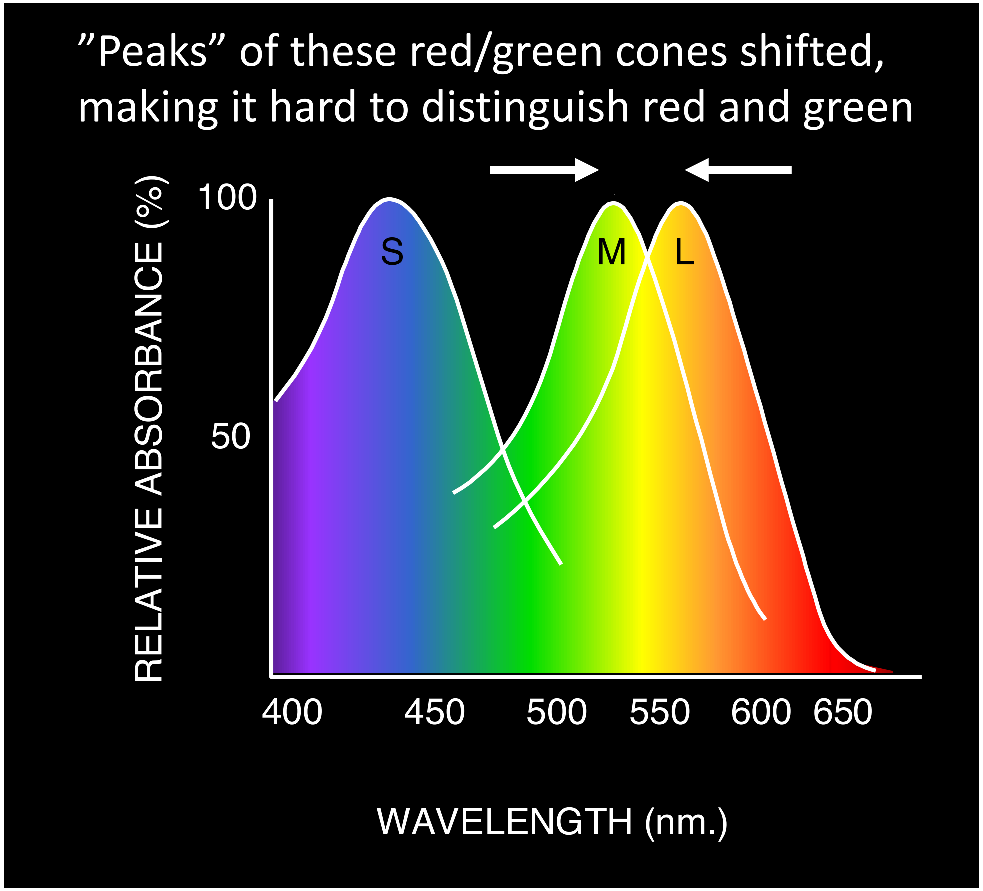




Smooth transition from violet to red is an artifact of our visual system!

Slide Credit and Copyright: S. Palmer

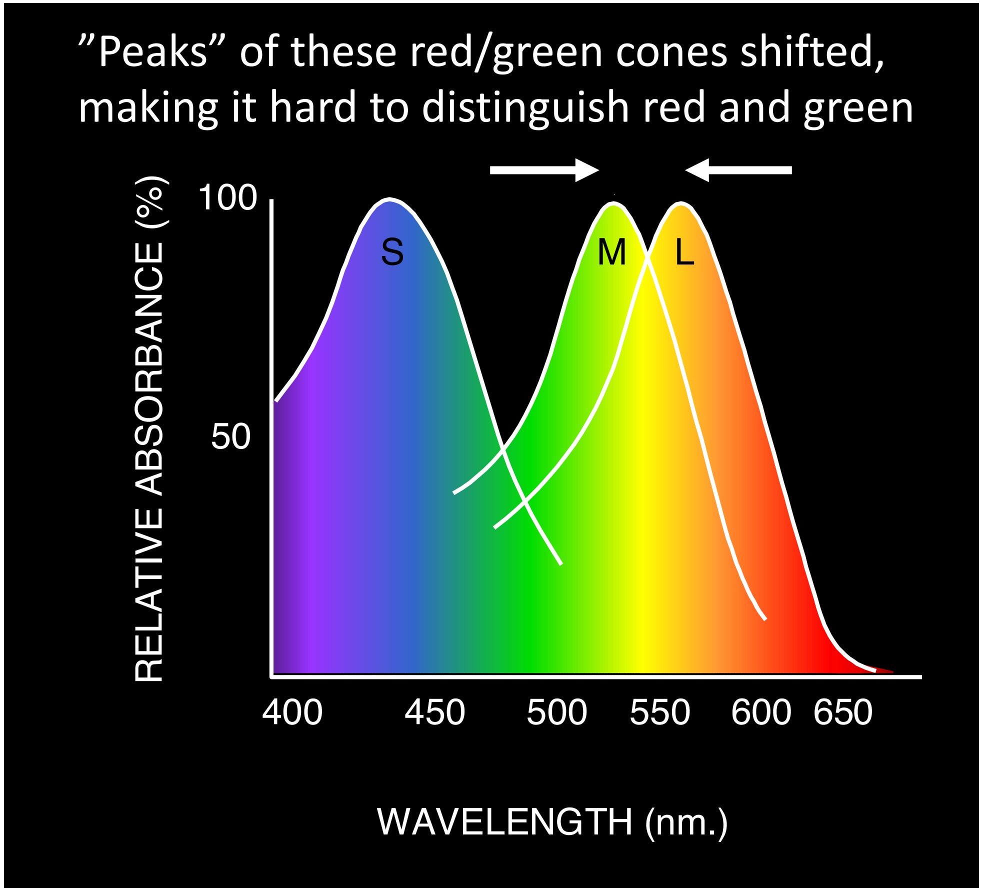
Red-Green Color Blindness



Four possibilities:

- **Deuteranomaly**: Green cone shifted toward red
- **Protanomaly**: Red cone shifted toward green
- **Deuteranopia**: Green cone missing
- **Protanopia**: Red cone missing

Red-Green Color Blindness

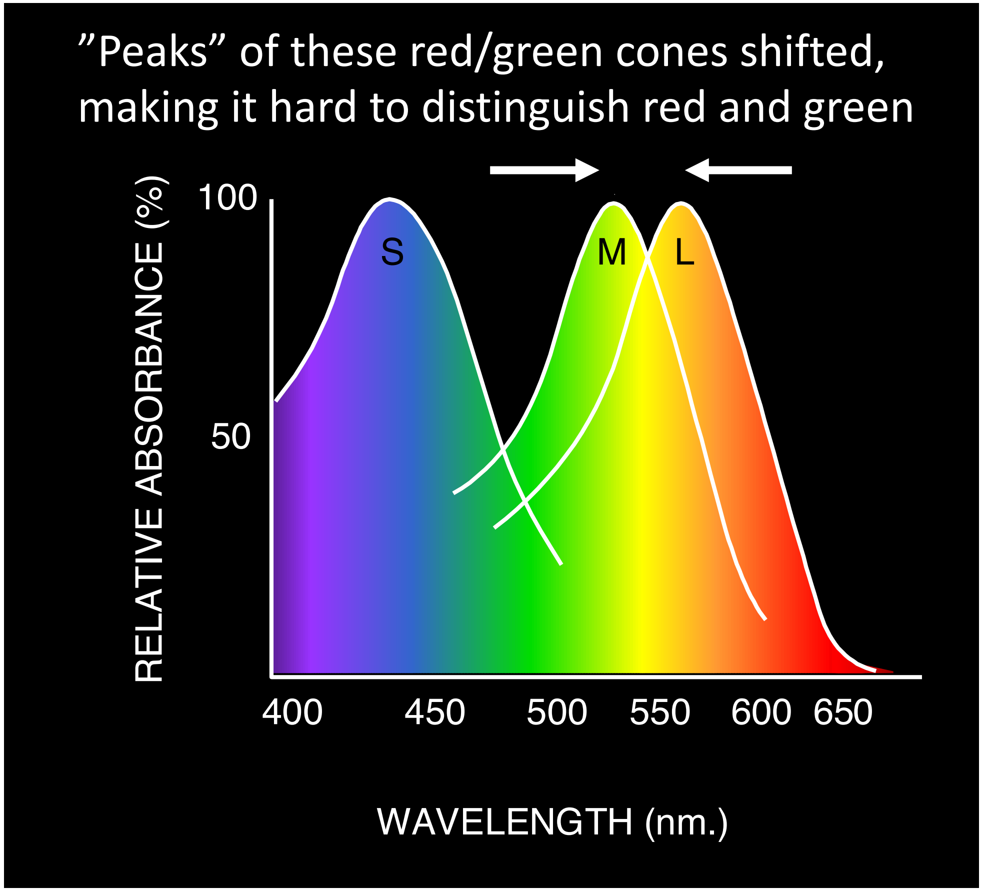


Four possibilities:

- **Deuteranomaly**: Green cone shifted toward red
- **Protanomaly**: Red cone shifted toward green
- **Deuteranopia**: Green cone missing
- **Protanopia**: Red cone missing

- Genes for red and green cones on X chromosome
- Men (XY): ~8% red-green color blind, (Northern European descent)
- Women (XX): ~0.5% red-green color blindness (Northern European descent)

Red-Green Color Blindndess



Four possibilities:

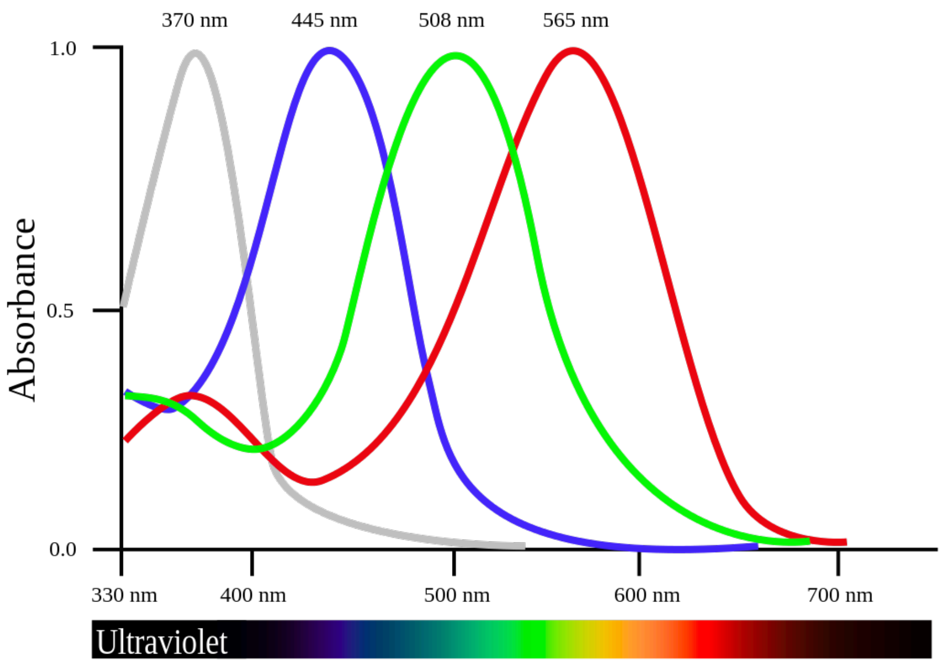
- **Deuteranomaly**: Green cone shifted toward red
- **Protanomaly**: Red cone shifted toward green
- **Deuteranopia**: Green cone missing
- **Protanopia**: Red cone missing

- Genes for red and green cones on X chromosome
- Men (XY): ~8% red-green color blind, (Northern European descent)
- Women (XX): ~0.5% red-green color blindness (Northern European descent)
- Up to 15% of women may have four types of cones! (Tetrachromacy)

Jordan and Mollon, "A study of women heterozygous for colour deficiencies", Vision Research 1993

Color Vision in Animals

Birds have four types of cones: can see ultraviolet light



Some flowers have “Nectar Guides” visible under UV light

Visible Light UV light

Human cones also sensitive to UV light! But blocked by lens

Image source: Wikipedia

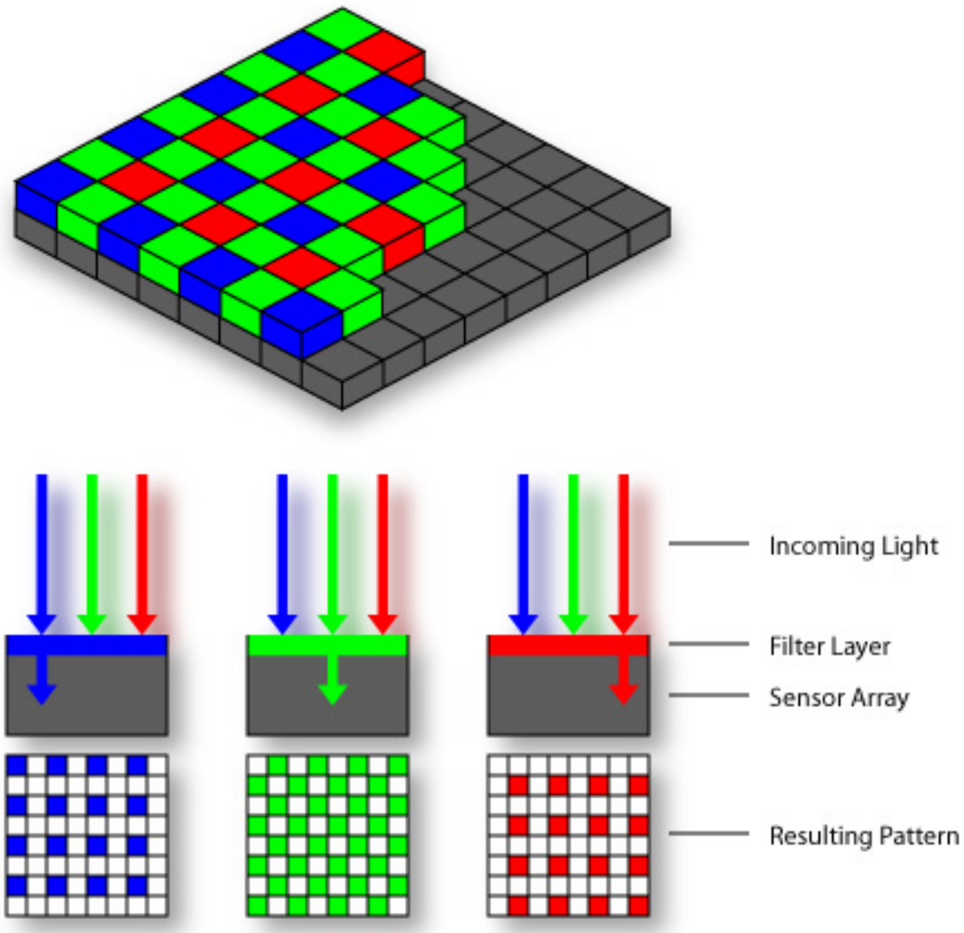
Image Source: Wikipedia

Color Vision in Animals

Mantis Shrimp: Up to 16 types of photoreceptors!
Can also detect polarization of light!

Image source: Wikipedia

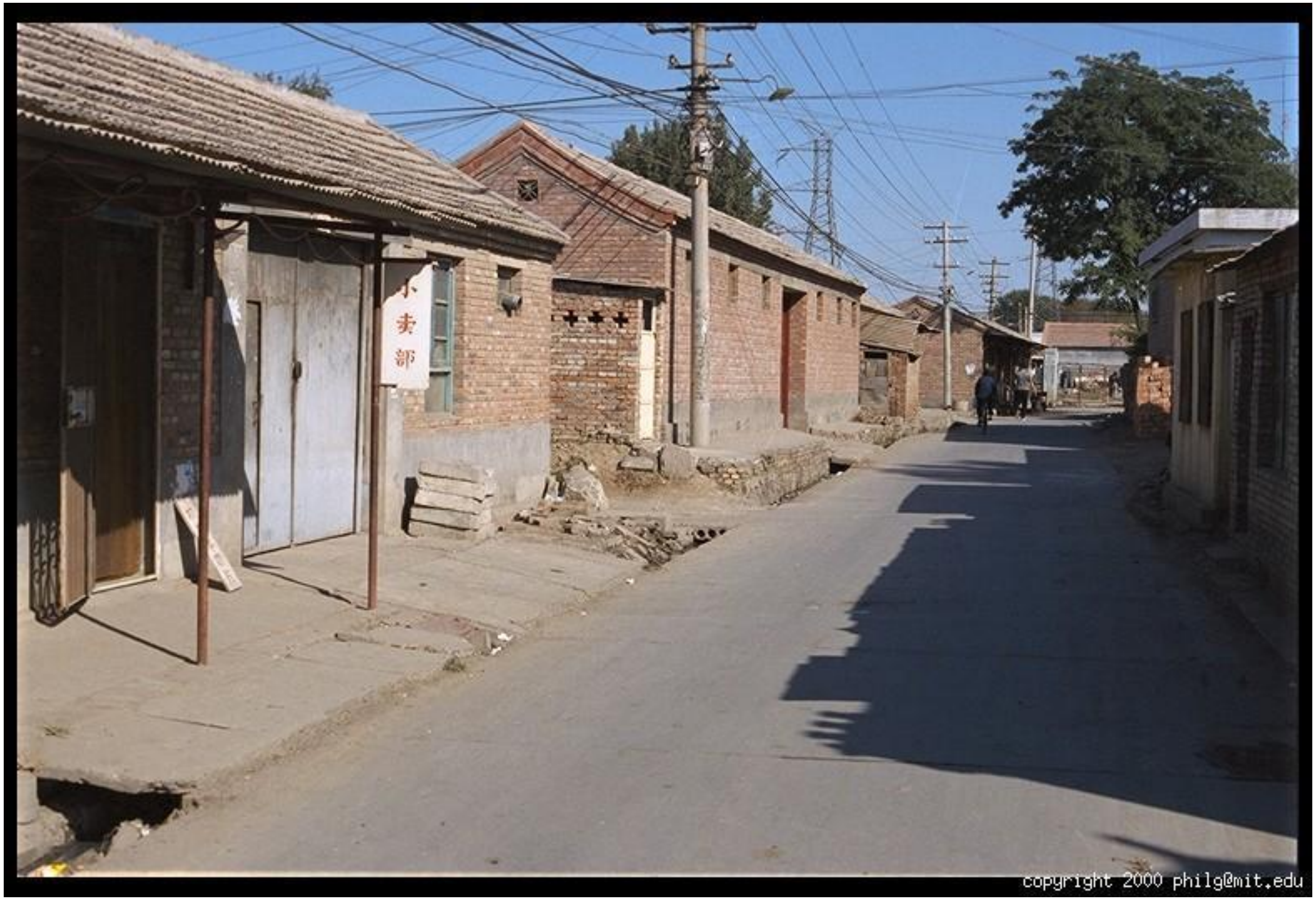
Artificial Cones



Estimate RGB
at ‘G’ cells from
neighboring values

Slide Credit: S. Seitz

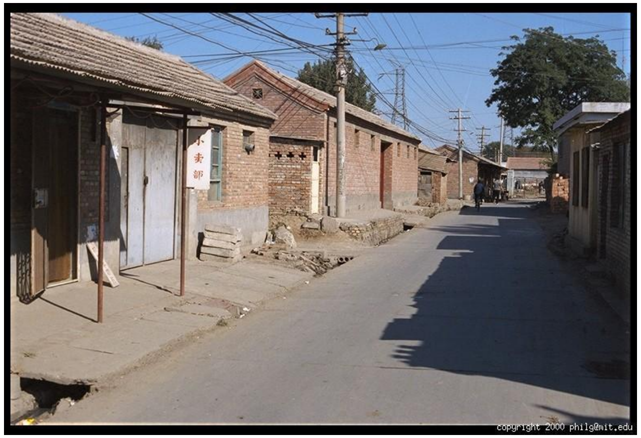
Color Images



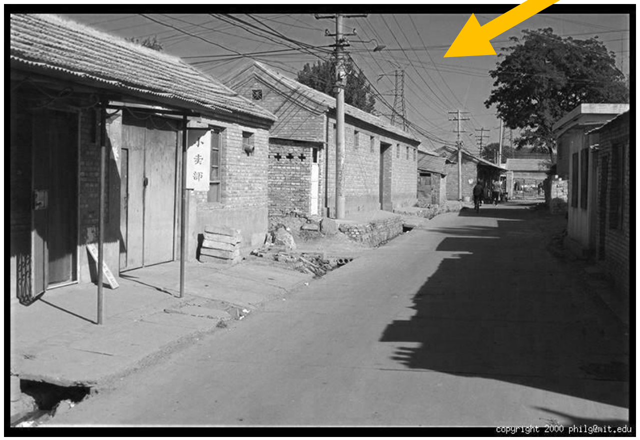
Slide Credit: J. Hays

Color Images

Combined



Red

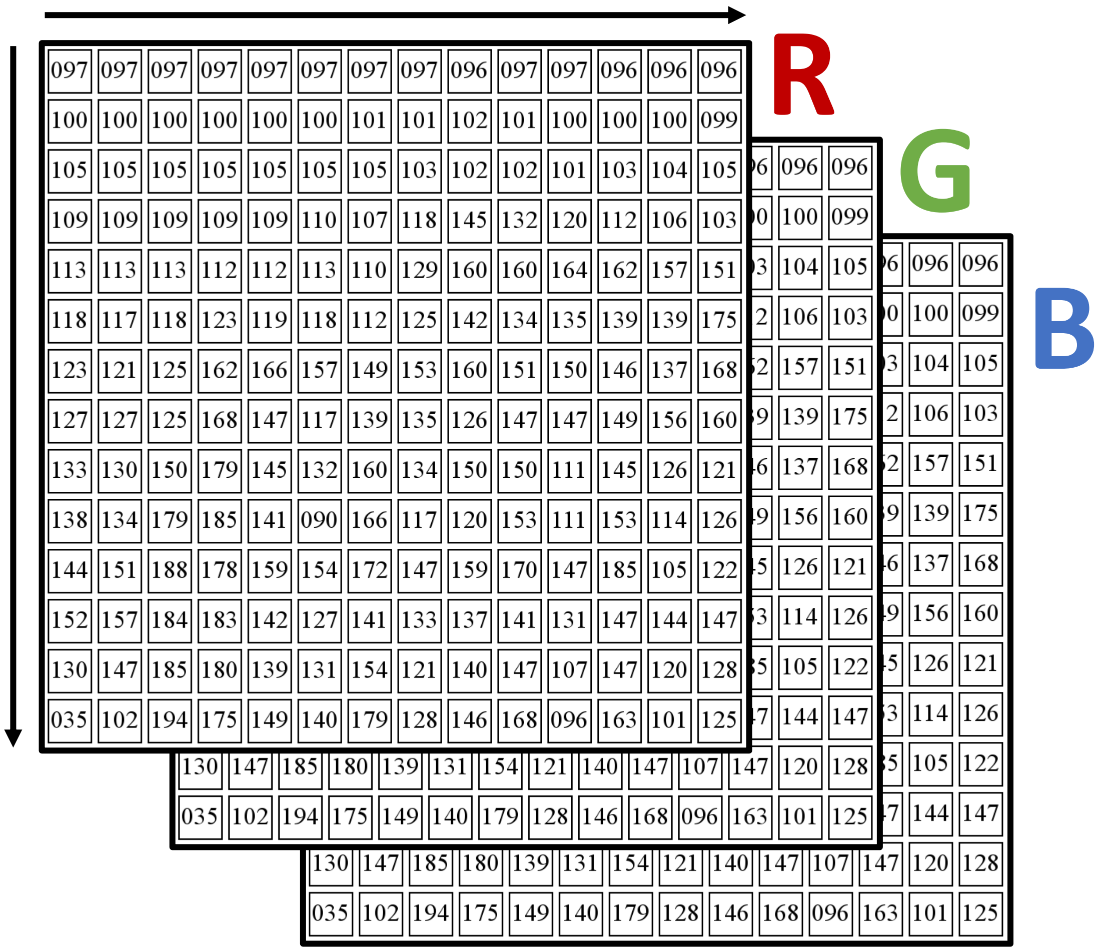


Green

Blue

Slide Credit: J. Hays

Images in Python



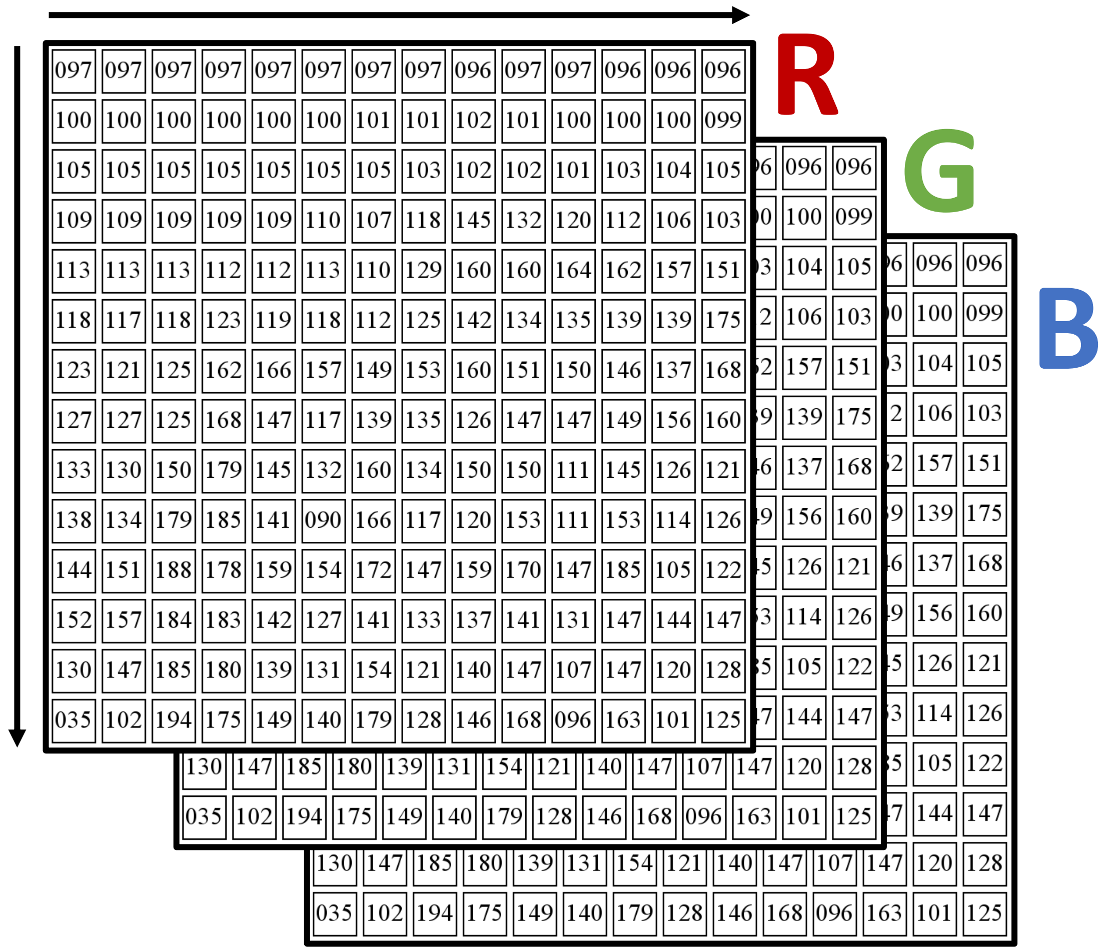
Images in Python

Images are matrix / tensor `im`

`im[0,0,0]`
top, left, red

`im[y,x,c]`
row y, column x, channel c

`im[H-1,W-1,2]`
bottom right blue

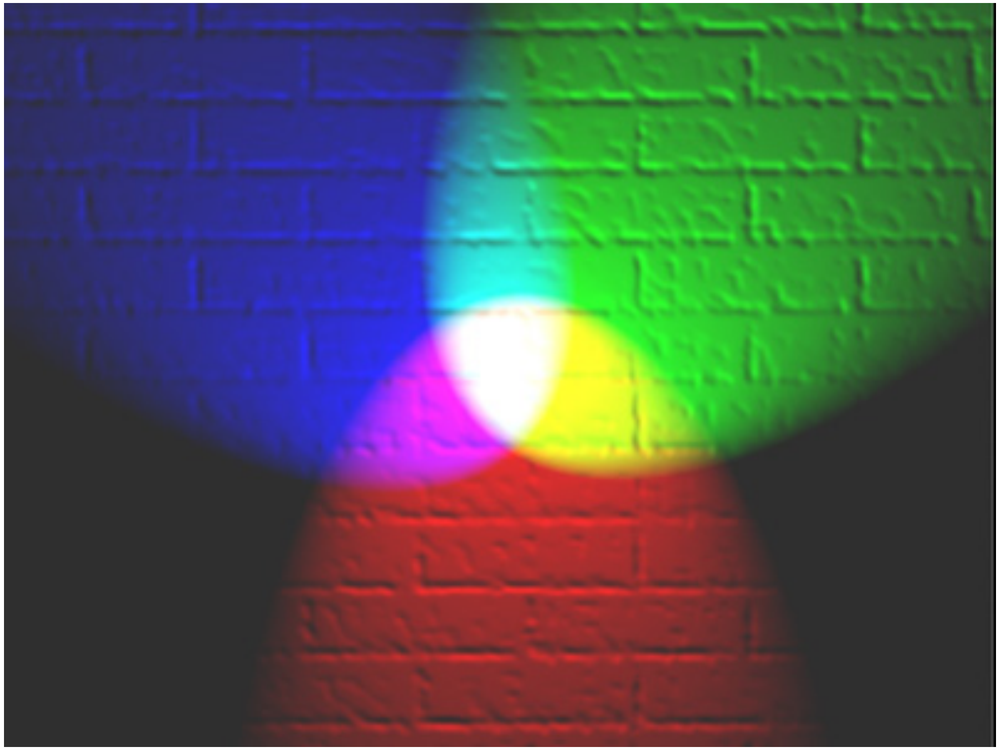


Slide inspired by James Hays

5 Things to Remember

1. Origin is top left
2. Rows are first index (**what's the fastest direction for accessing?)**
3. Usually referred to as HWC (Height x Width x Channel) But you'll sometimes see CHW (especially with neural networks)
4. Typically stored as uint8 [0,255]
5. for y in range(H): for x in range(W): will run <u>1 million times</u> for a 1000x1000 image. *A 4GHz processor can do only 4K clock cycles per pixel per second.*

Representing Colors



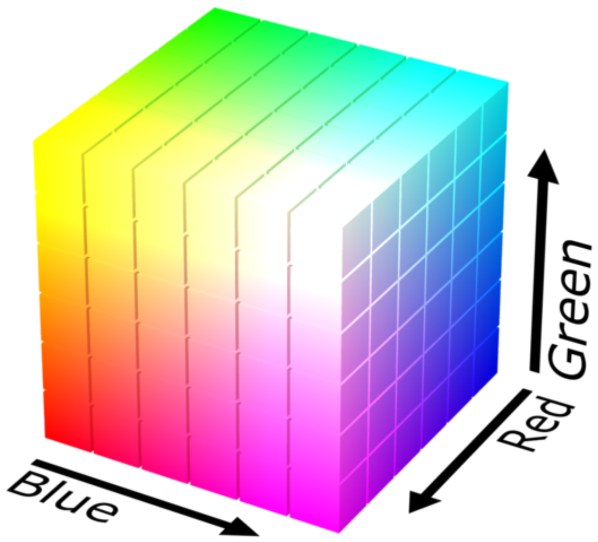
Discussion time: how many numbers do you actually need for colored light? Assume all tuples (R,G,B) are legitimate colors (they are).

Image Credit: http://en.wikipedia.org/wiki/File:RGB_illumination.jpg

One Option: RGB

Pros
1. Simple
2. Common

Cons
1. Distances don't make sense
2. Correlated



Slide Credit: J. Hays, RGB cube: https://en.wikipedia.org/wiki/RGB_color_model

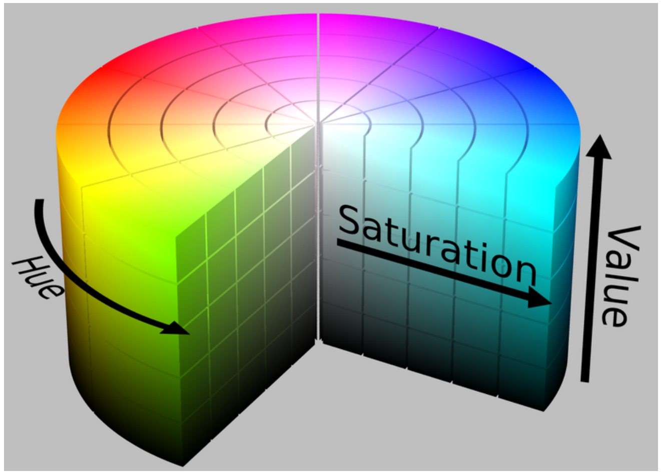
Another Option: HSV

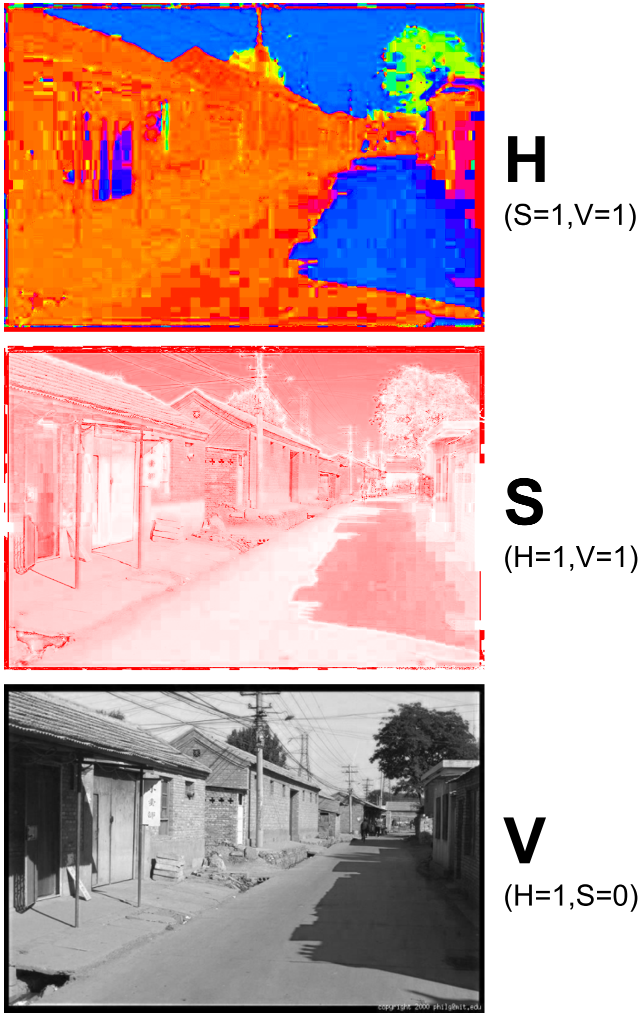
Pros

1. Intuitive for picking colors
2. Sort of common
3. Fast to convert

Cons

1. Not as good as other better spaces





Slide Credit: J. Hays, HSV cylinder: https://en.wikipedia.org/wiki/HSL_and_HSV

Another Option: YUV / YCbCr

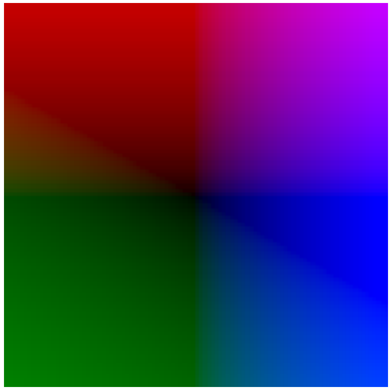
<u>Pros</u>

1. Great for transmission / compression

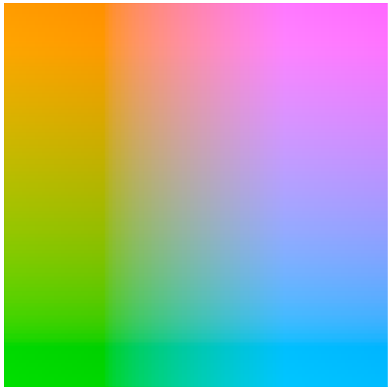
<u>Cons</u>

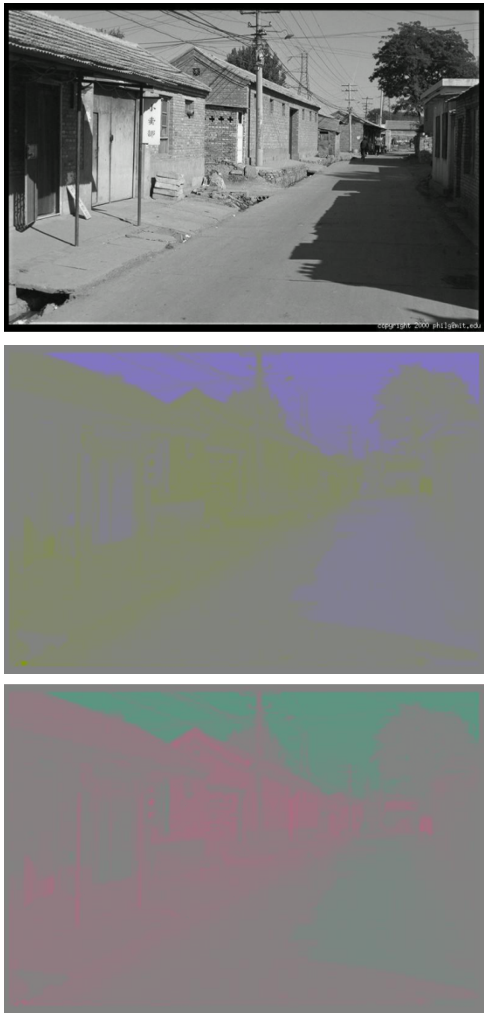
1. Not as good as other better smart color spaces

Y = 0



Y = 0.5





Y
(Cb=0.5, Cr=0.5)

Cb
(Y=0.5, Cr=0.5)

Cr
(Y=0.5, Cb=05)

Slide Credit: J. Hays, YUV cube: https://en.wikipedia.org/wiki/YUV

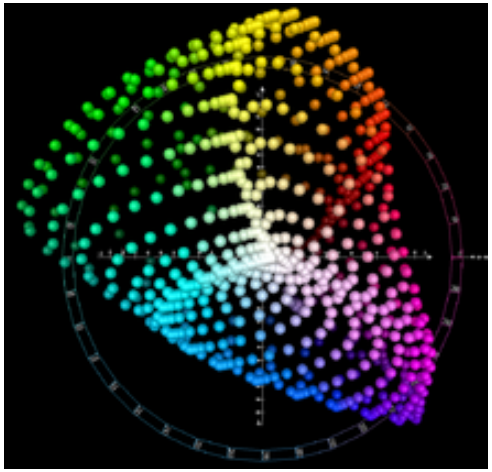
Another Option: LAB

Pros

1. Distances correspond with human judgment
2. Safe

Cons

1. Complex to calculate (don't write it yourself, lots of fp calculations)



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Slide Credit: J. Hays, Lab diagram cube: https://en.wikipedia.org/wiki/CIELAB_color_space

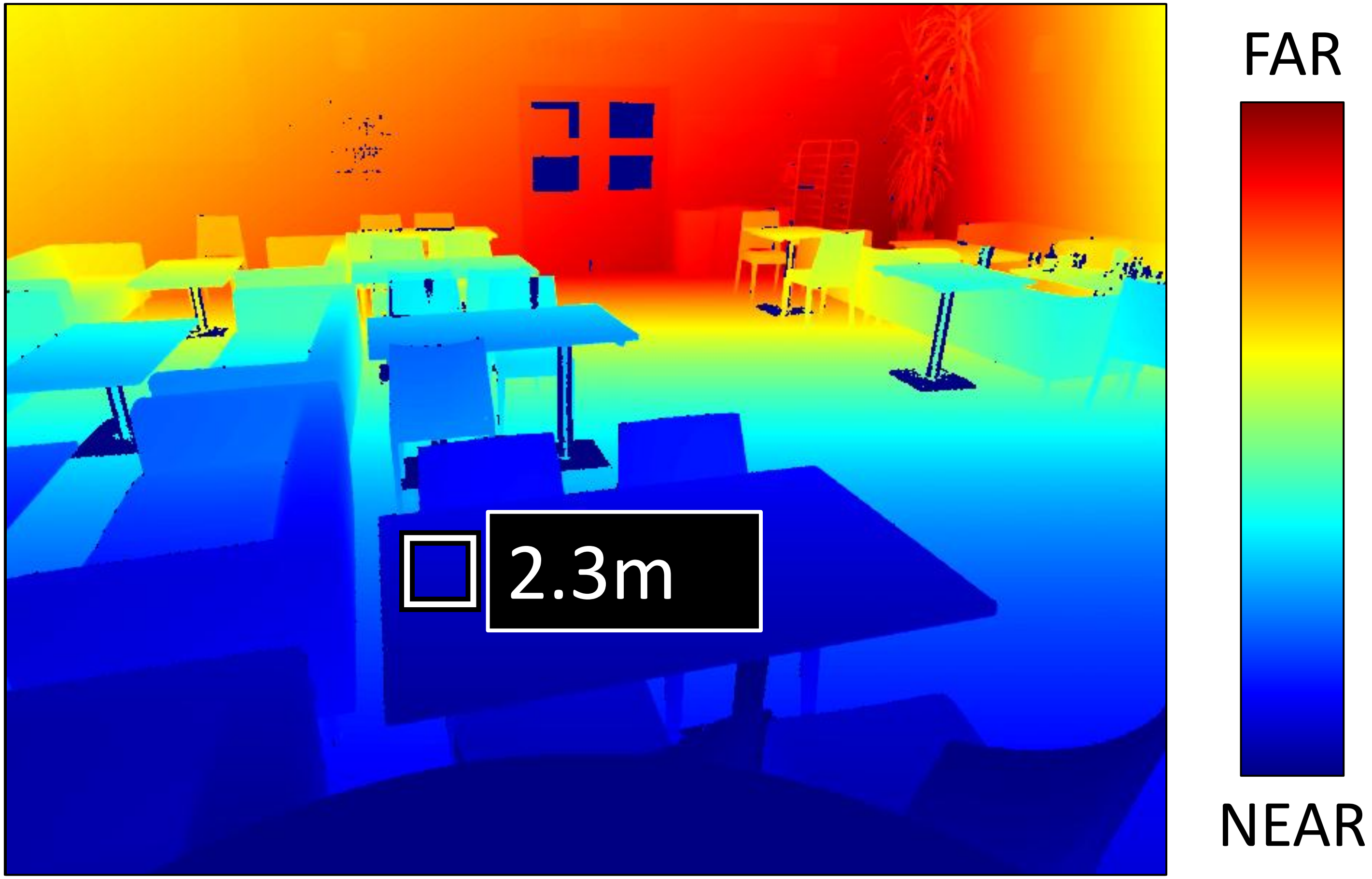
Why are there so many?

- Each serves different functions
  - RGB: sort of intuitive, standard, everywhere
  - HSV: good for picking, fast to compute
  - YCbCr/YUV: fast to compute, compresses well
  - Lab: the right(?) thing to do, but “slow” to compute
- Pick based on what you need and don’t sweat it: color really isn’t crucial

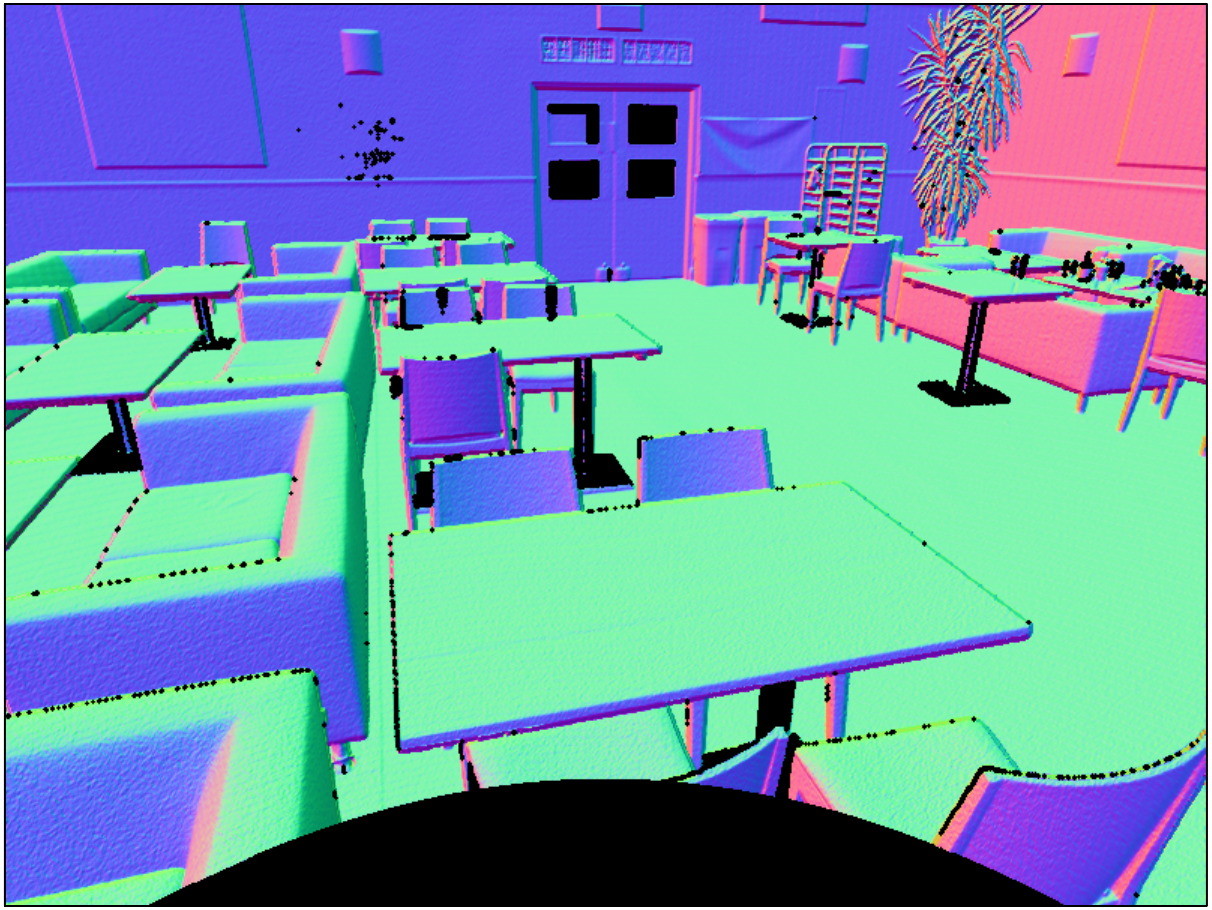
Other Types of “Images”

- Almost all of this class is about ordinary RGB images because this has driven a lot of applications
- However, there are lots of other images

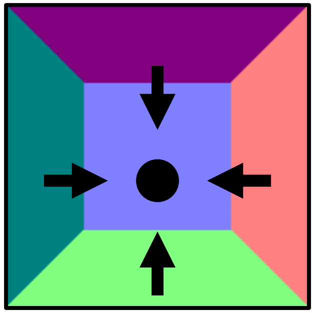
Depth Map



Surface Normals



Room



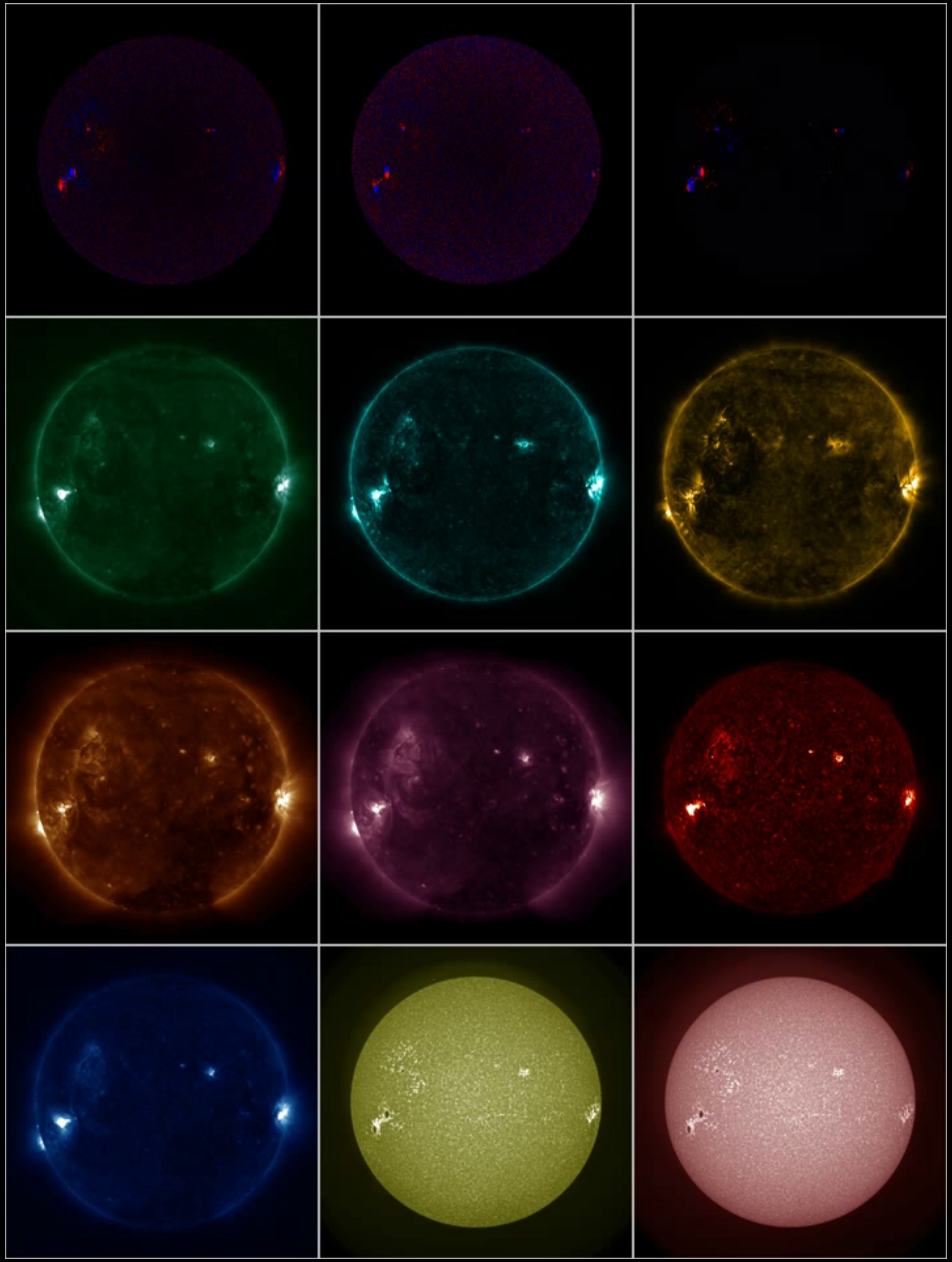
Legend

Science Data

Magnetic Field in:
x, y, z

Light at 9 ~wavelenths:
9.4nm, 13.1nm, 17.1nm
19.3nm, 21.1nm, 30.4nm
33.5nm, 160nm, 170nm

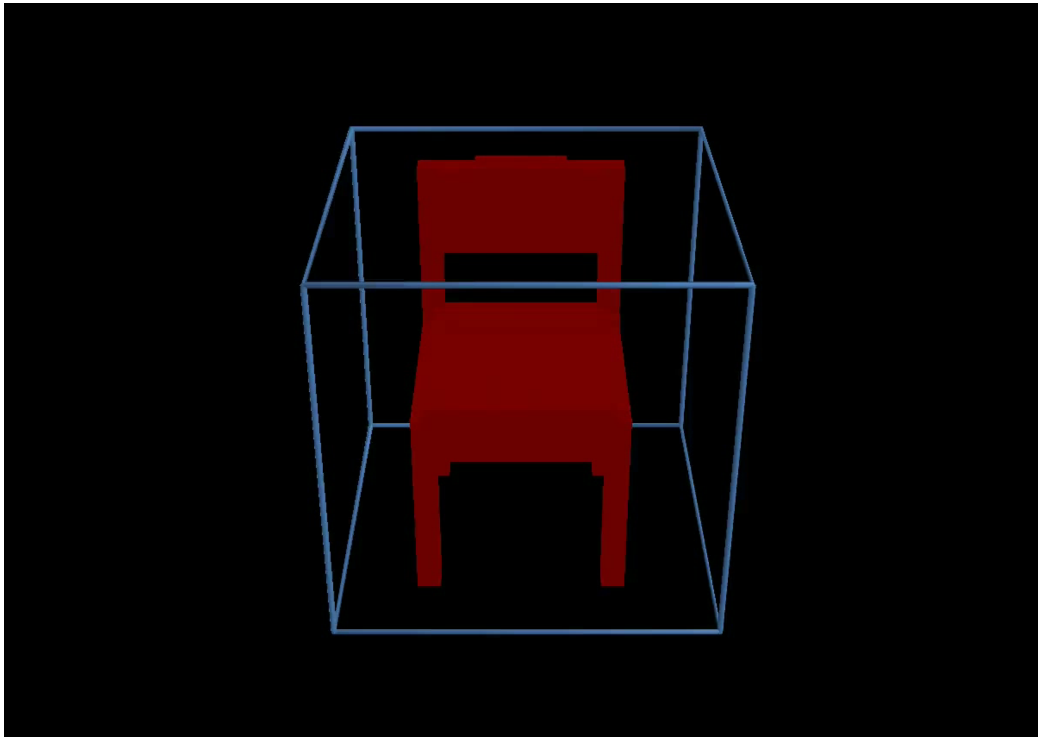
NASA Solar Dynamics Observatory
observing solar flare



Volumes

Volumes: images with more dimensions.

Emerge in 3D reconstruction, medical imaging, temporal data



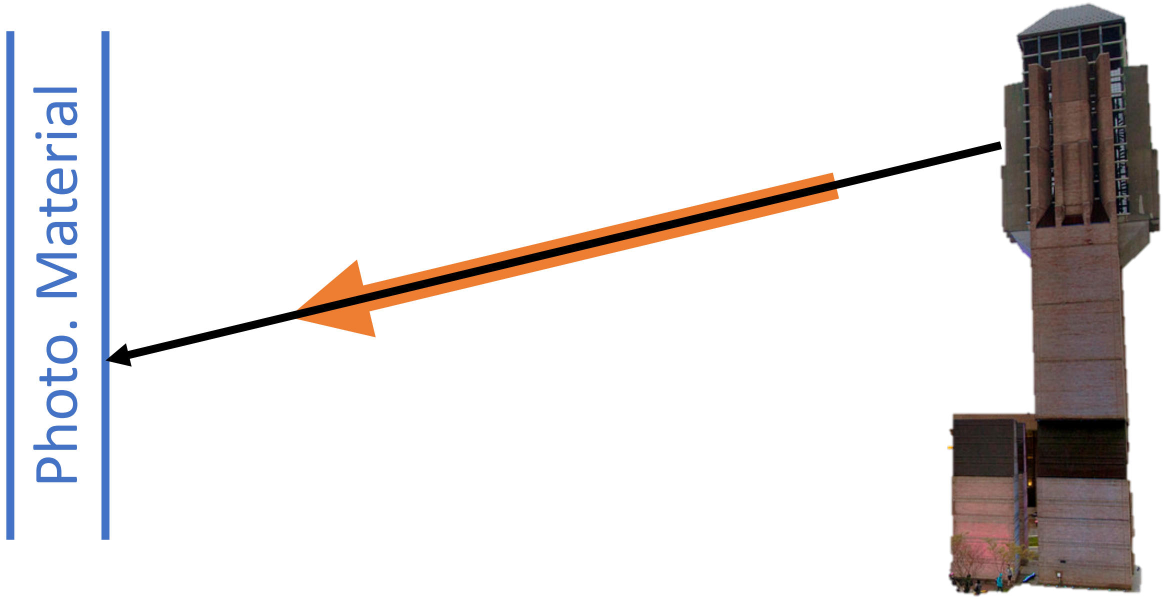
From: Girdhar et al., *Learning a predictable and generative vector representation for objects*. ECCV 2016

Other Types of “Images”

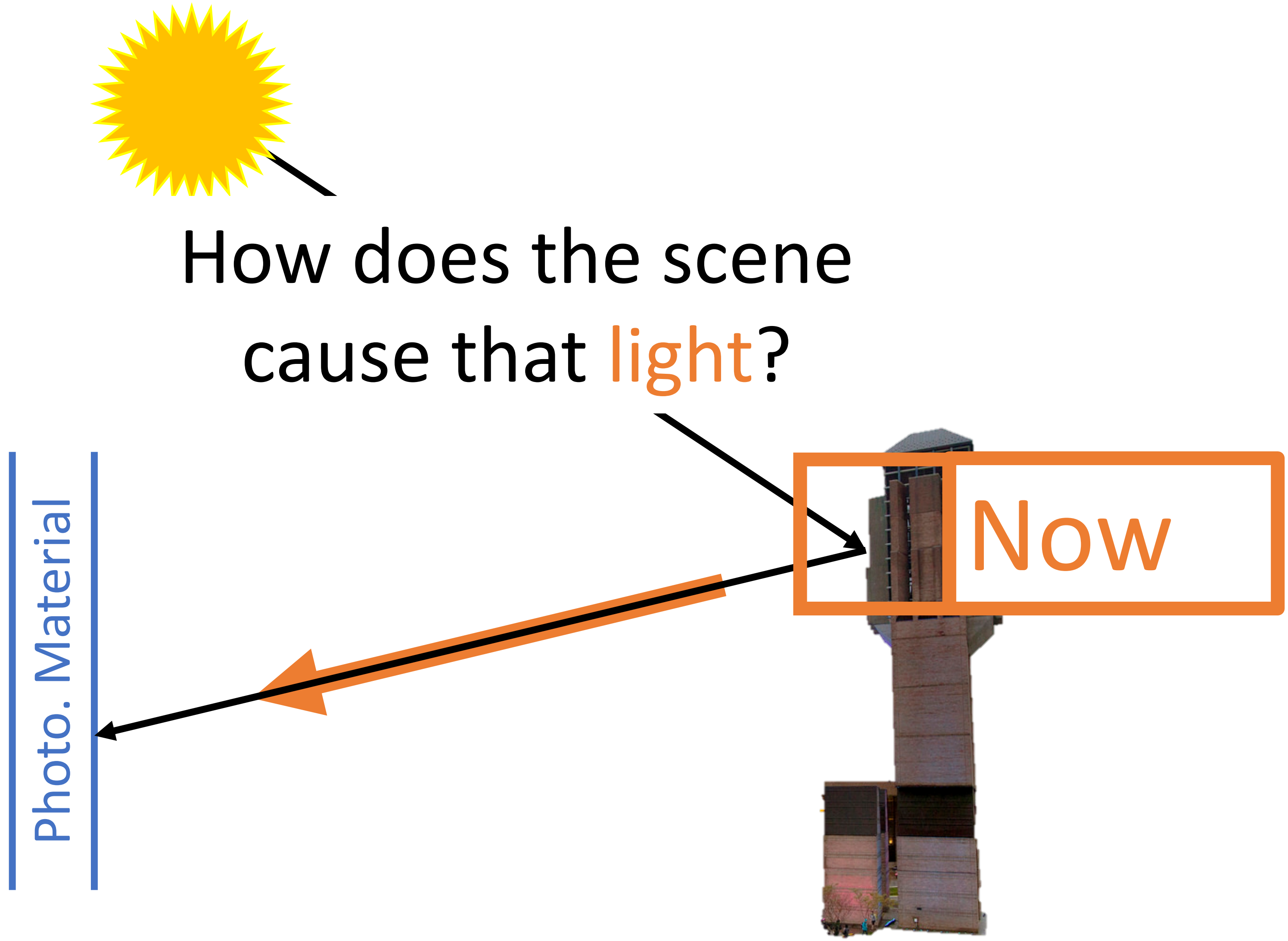
- Almost all of this class is about ordinary RGB images because this has driven a lot of applications
- However, there are lots of other images

So Far:

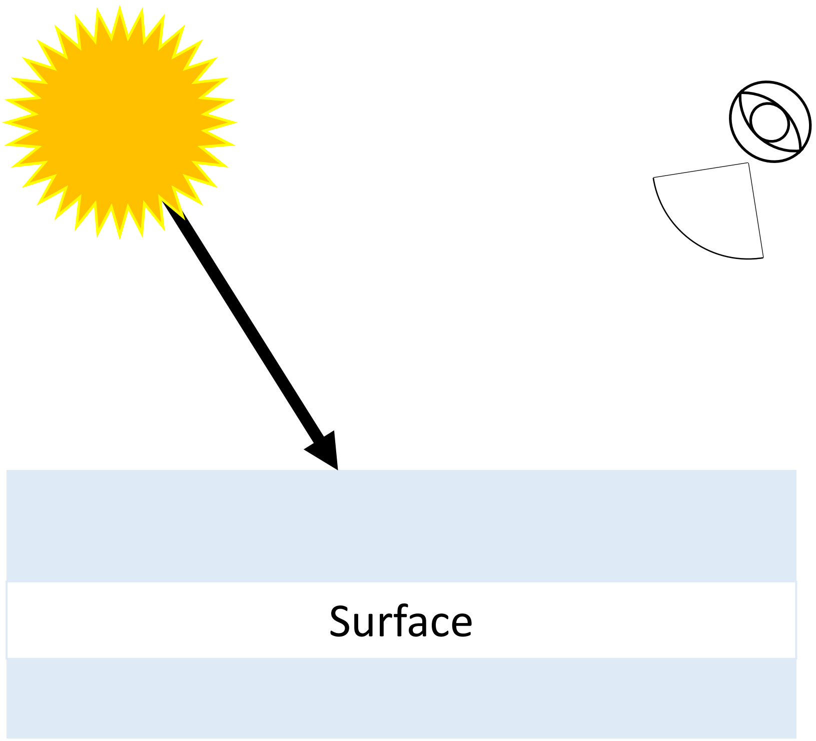
How do we represent light
and its storage on film?



So Far:

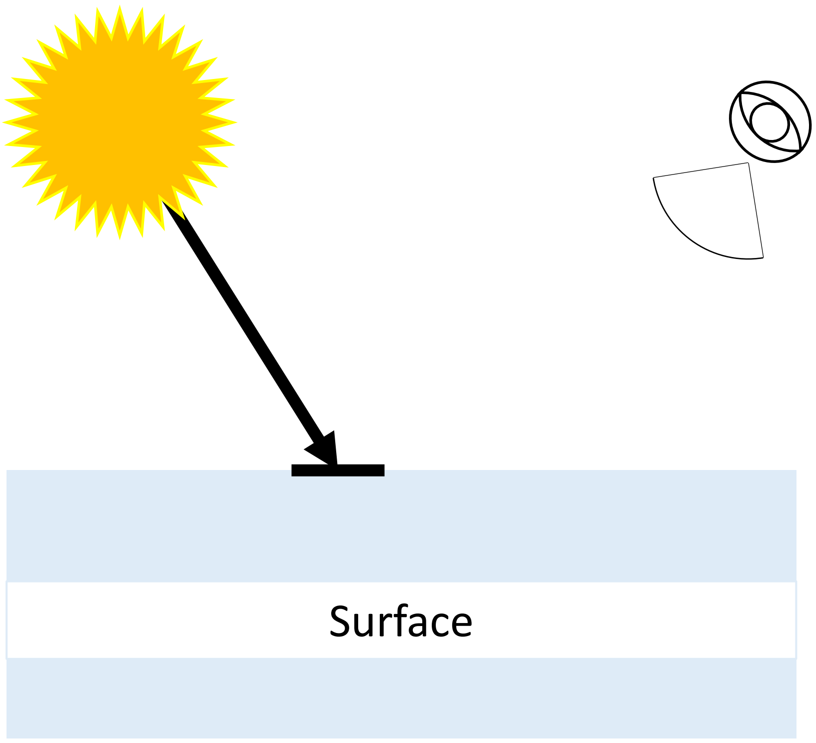


Light and Surfaces



What happens when light hits a surface?

Light and Surfaces

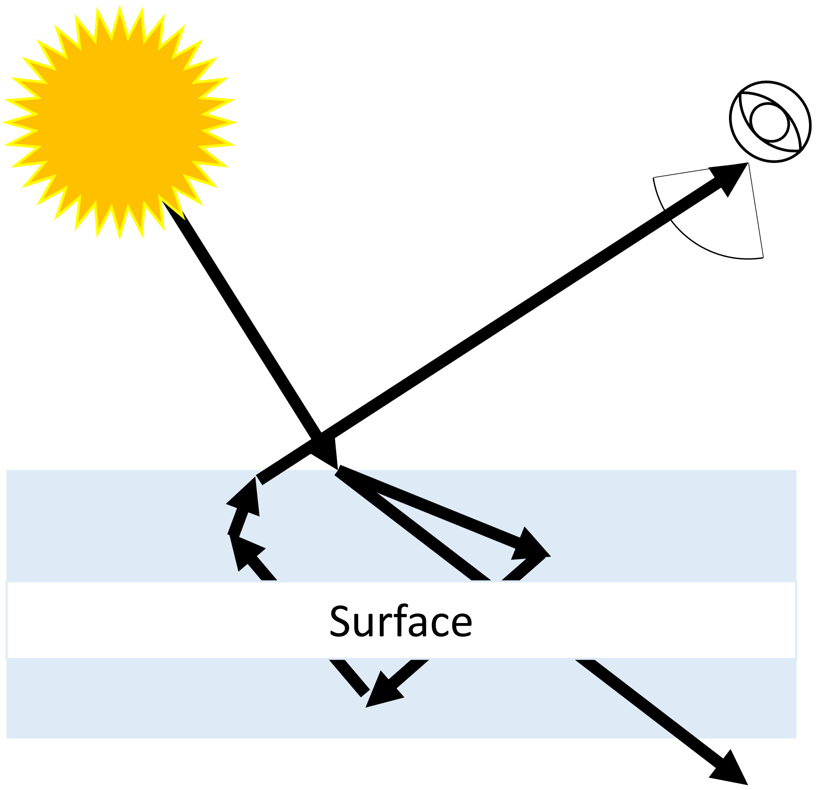


What happens when light hits a surface?

1. Absorbed

It's absorbed and converted into some other form of energy (e.g., a black shirt getting hot in the sun)

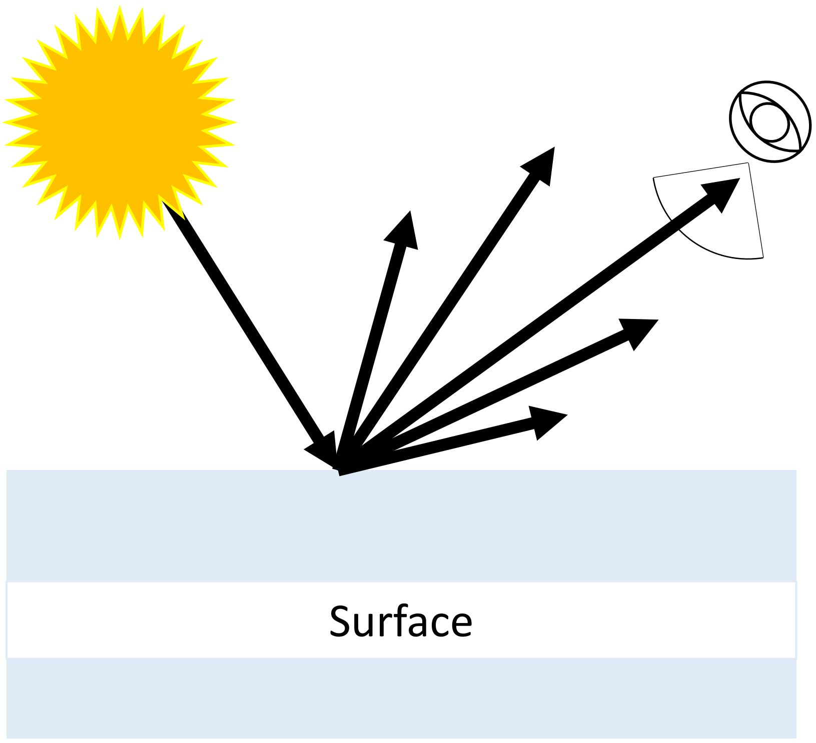
Light and Surfaces



What happens when light hits a surface?

2. Transmitted
Possibly bouncing around before going through or out (e.g. lenses bend and go through, milk bounces around)

Light and Surfaces

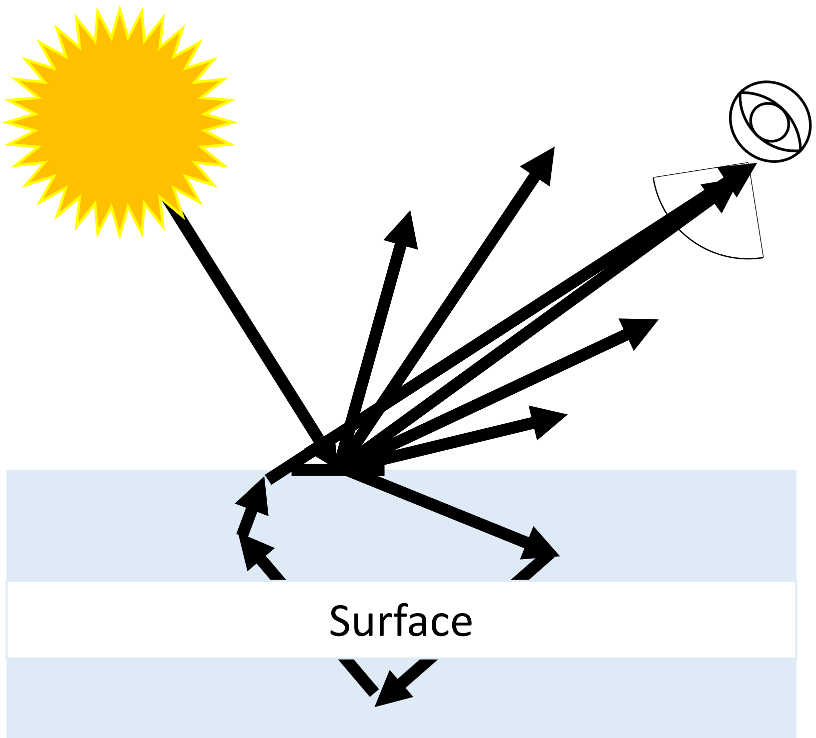


What happens when light hits a surface?

3. Reflected

It's reflected back, in one or more directions with varying amounts (e.g., mirror, or a white surface)

Light and Surfaces

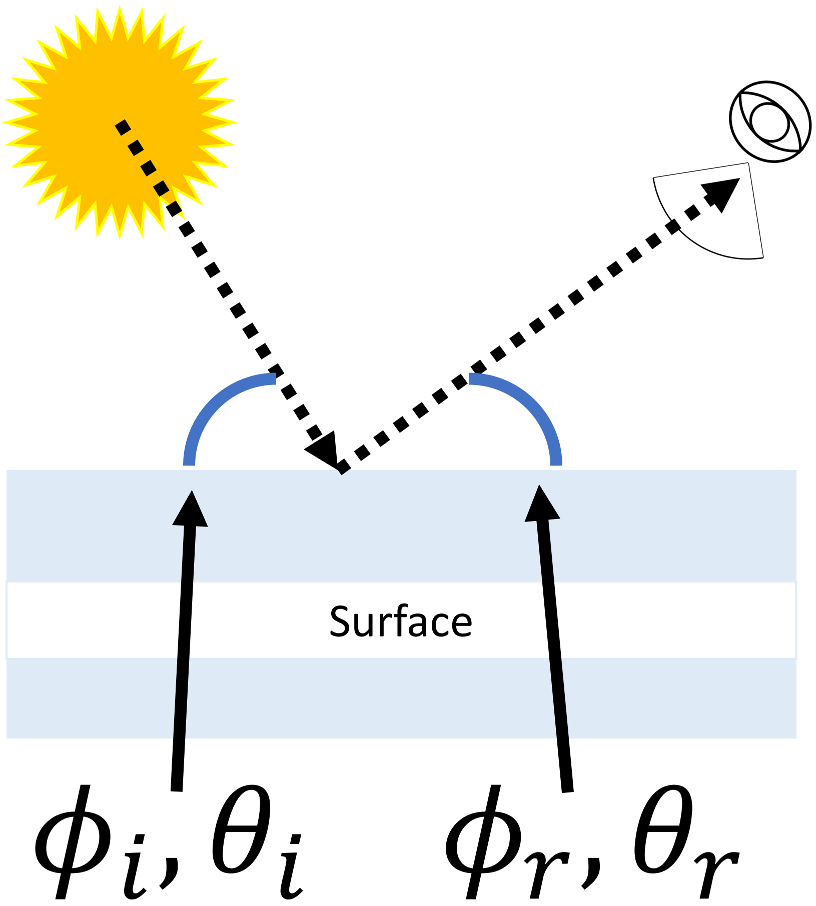


What happens when light hits a surface?

4. Everything

All of the above! Real surfaces often have combinations of all of these options.

Modeling Light and Surfaces



<u>Opaque Reflections</u>

Bi-directional reflectance distribution function (BRDF):

% reflected given

incident angle to light

reflected angle to the viewer.

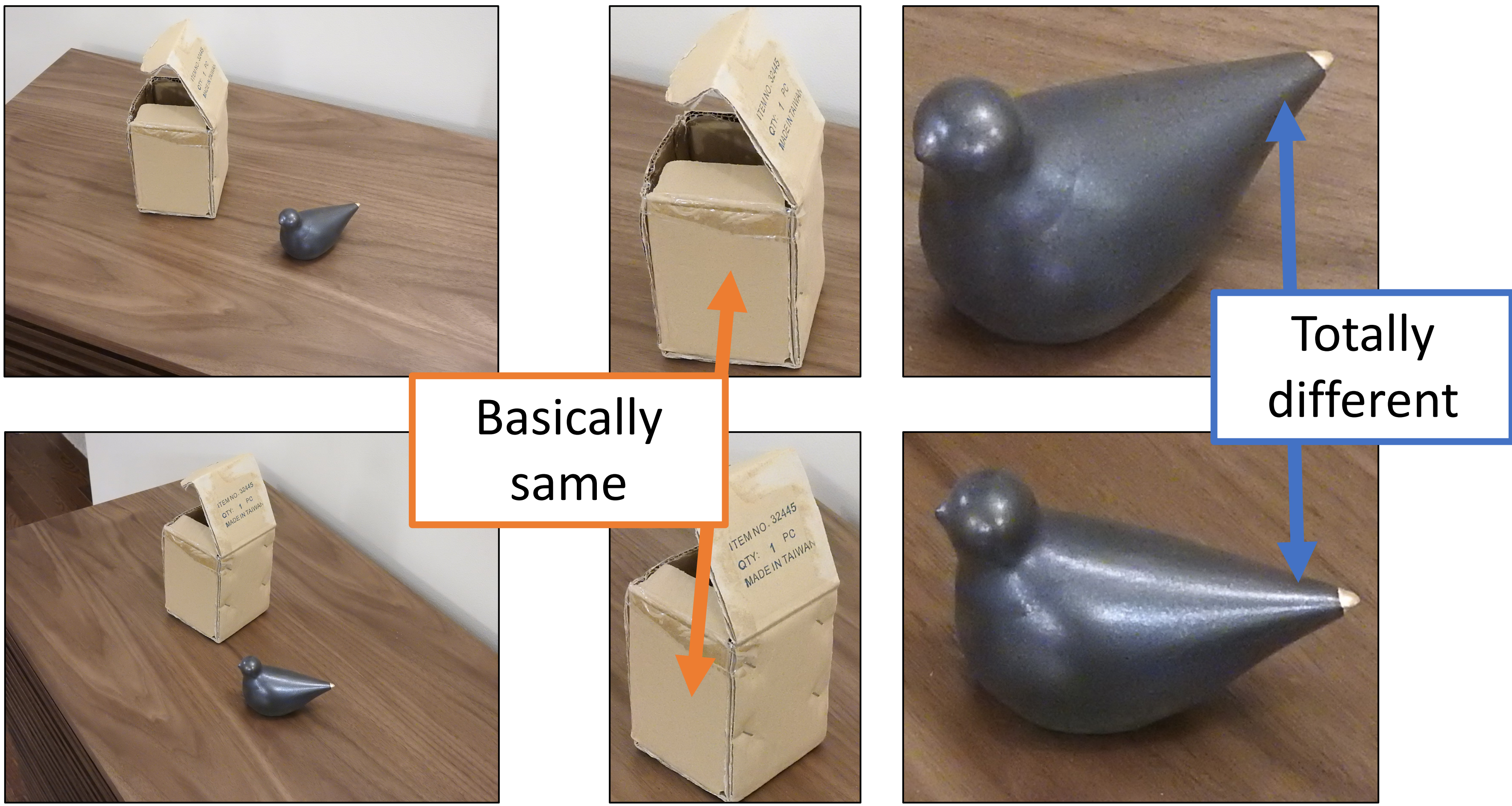
Note: have not specified form of function.

Specular and Diffuse Reflectionv

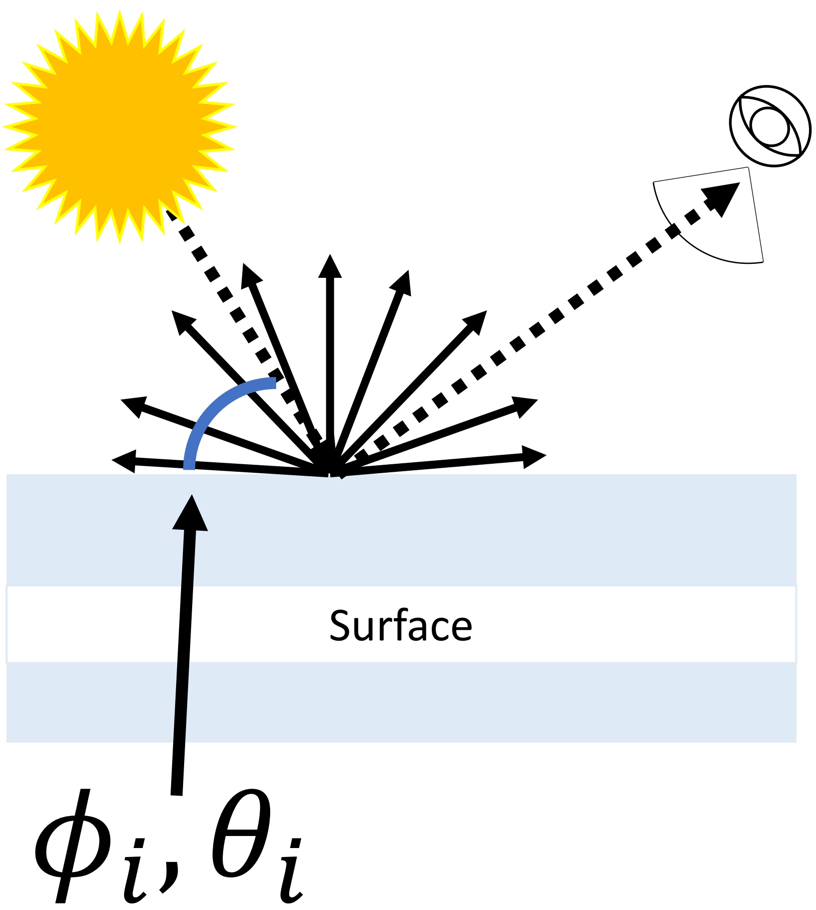
Same lighting, as close as possible camera settings, but different **location**

Specular and Diffuse Reflection

Diffuse   Specular



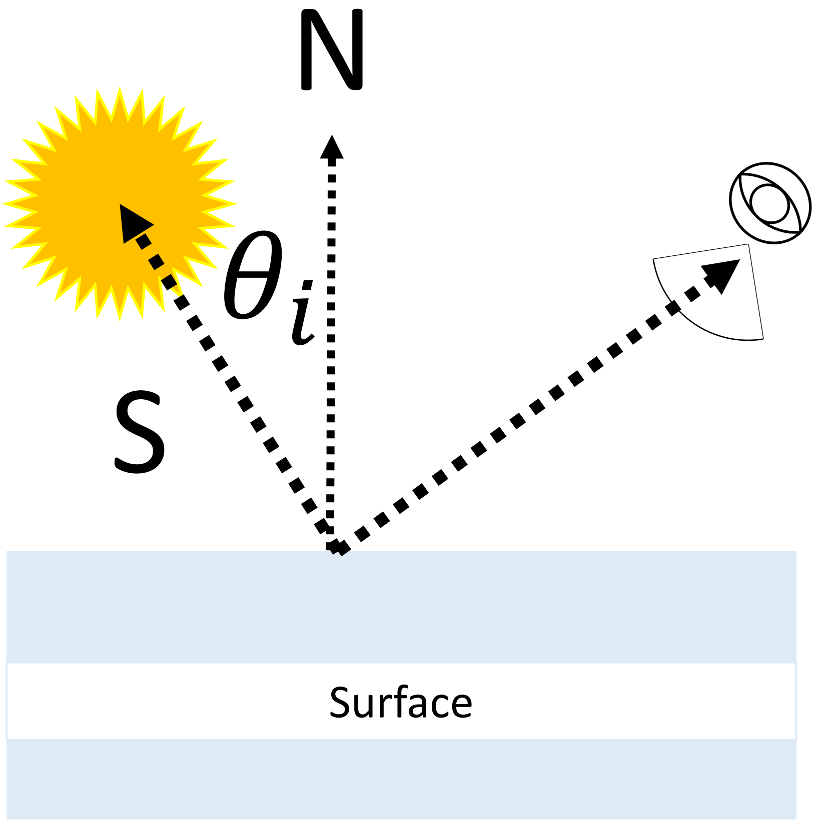
Diffuse Reflection



<u>Lambertian Surface</u>

Light depends **only** on orientation of surface ϕ_i, θ_i to light. Result of random small facets. Looks identical at all views.

Diffuse Reflection



Lambert's Law

N: surface normal

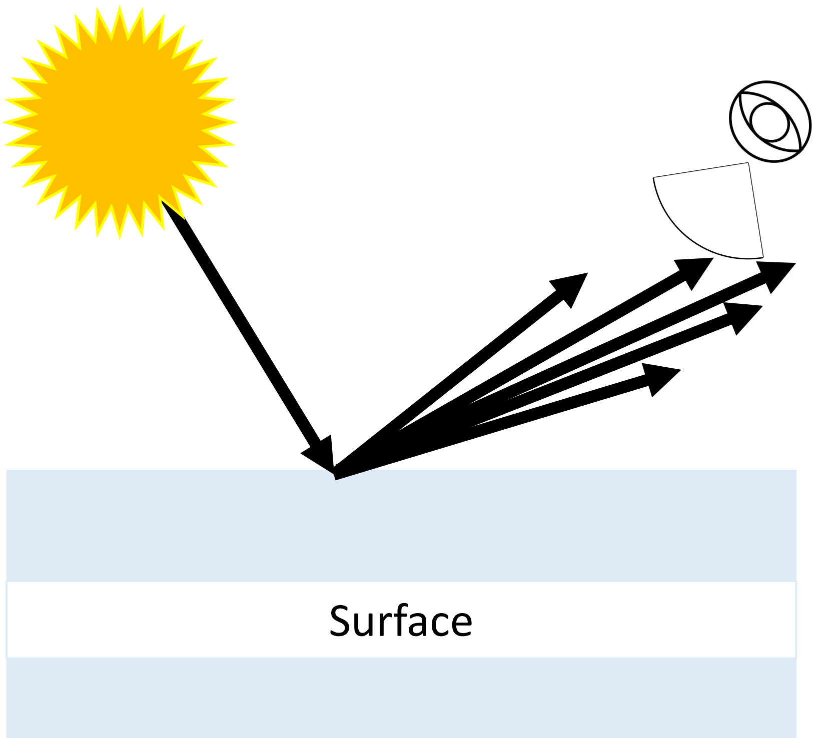
S: source direction **and** strength

ρ: how much is reflected

$$B = \rho \boldsymbol{N} \cdot \boldsymbol{S}$$

$$B = \rho ||\boldsymbol{S}|| \cos(\theta)$$

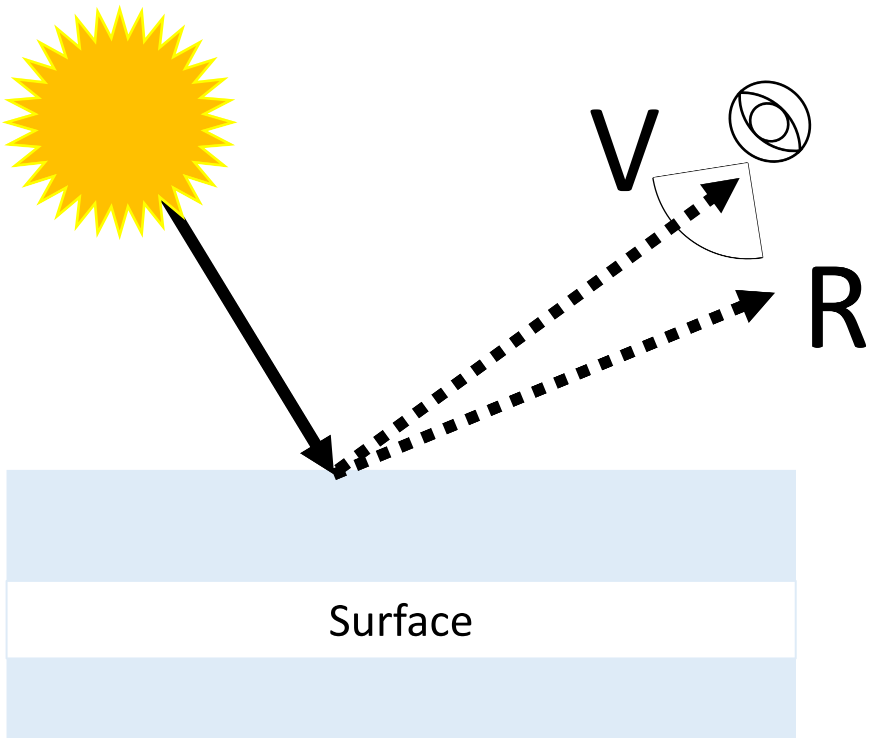
Specular Reflection



Specular Surface

Light reflected like a mirror, but spreads out in a “lobe” around the reflection ray

Specular Reflection



Phong Model

V: vector to viewer

R: reflection ray

α: shininess constant

$$B = (V^T R)^\alpha$$

BRDFs can be very complex

Slide Credit: L. Lazebnik

How is this useful?

Shape from Shading

Lambert's Law: for every pixel i

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

Reflected Light (1 dim)	Surface Orientation (3? dim)	Illumination Global, (3 dim)

Given: illumination and light, recover normals

Potential problems?

Shape from Shading

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

1D, **fixed**

actually 2D **unknown**

3D, **fixed**

- System of equations that's underdetermined (N equations, 2N unknowns, N+3 known)
  - **Solution**: Add more equations that enforce smoothness or finding a single surface.

Realistic Shape from Shading

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

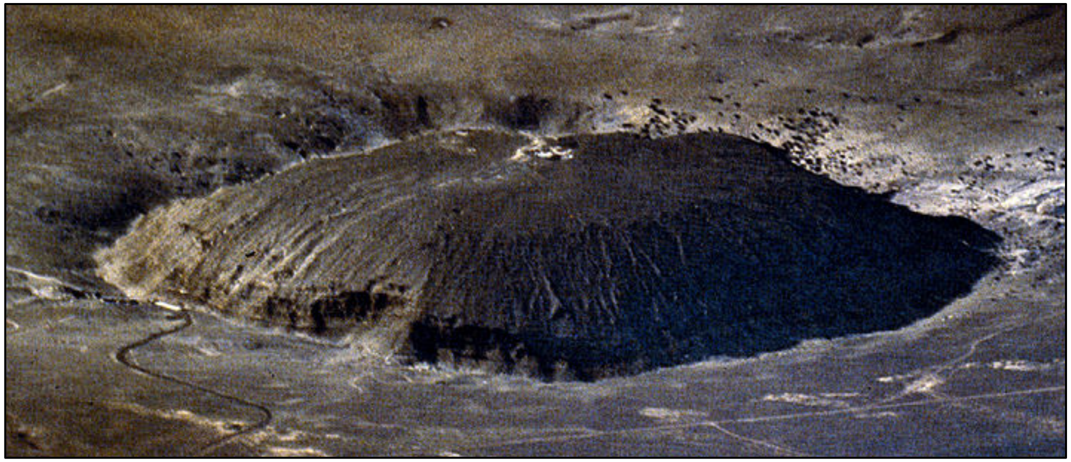
1D, **fixed**

2D **unknown**

3D, **unknown**

- System of equations that's underdetermined (N equations, 2N+3 unknowns)
- **Solution**: need prior beliefs to disambiguate.

Ambiguity



Ambiguity

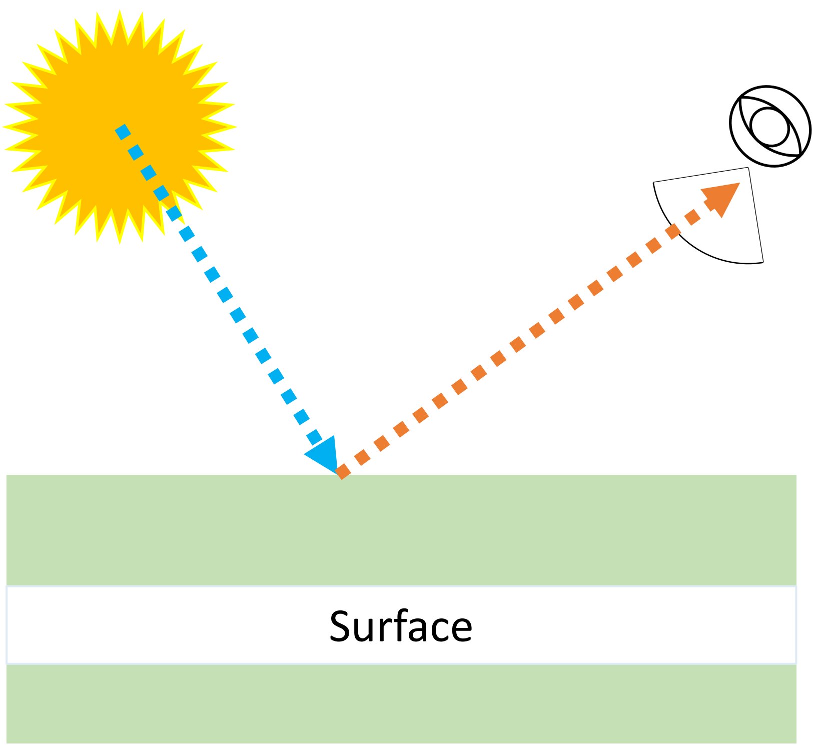
Humans assume light from above (and the blueness also tells you distance)

Photo Credit: https://en.wikipedia.org/wiki/Meteor_Crater

Shape from Shading in Practice

https://www.youtube.com/watch?v=4GiLAOtjHNo

Modeling Light and Surfaces



Color that reaches eye depends on **color of light** and **color of material**

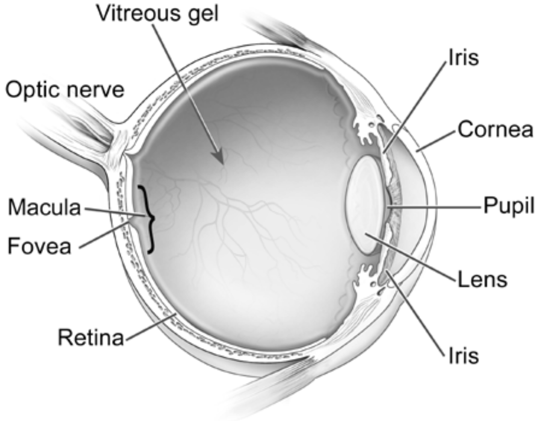
Color Ambiguity: “The Dress”

Blue / Black dress under yellow light?

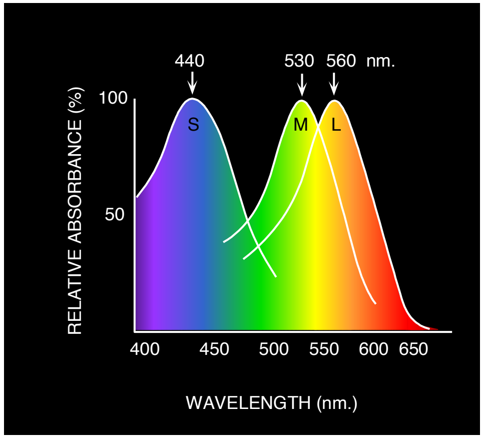
White / Gold dress under blue light?

https://en.wikipedia.org/wiki/The_dress

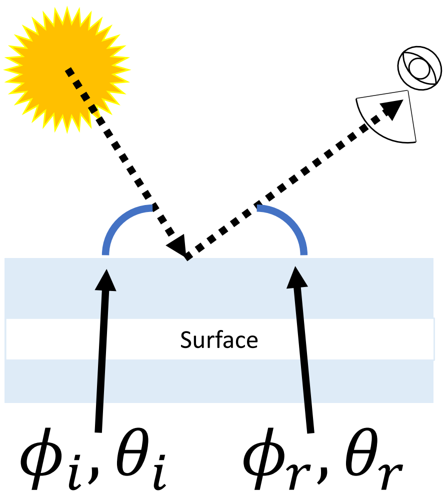
Recap



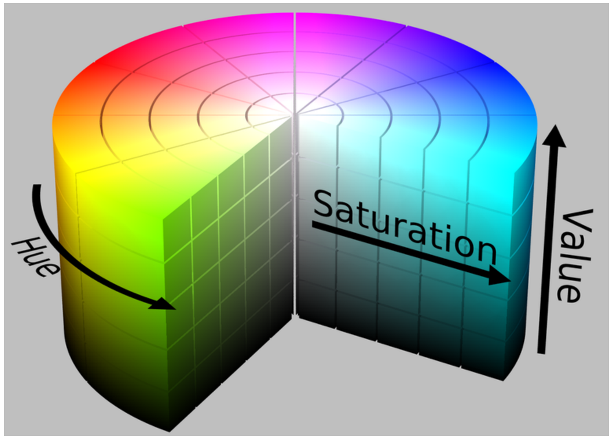
Human Eye



Color Vision



Light and Surfaces



Color Spaces

Next Lecture:
Math Review