



# Computer Graphics



Felix Heide

Princeton University

COS 426, Spring 2022

# Overview



- Administrative stuff
  - People, times, places, etc.
- Syllabus
  - What will I learn in this course?
- Imaging
  - Getting started ...

# Administrative Stuff



- **Instructors**

- Prof: Felix Heide
- TAs: Ethan Tseng, Chloe Qiu, Vivien Nguyen, Yuting Yang
- Lab TAs: Edward Yang, and TBD

- **Book**

- *Computer Graphics with OpenGL, 4<sup>th</sup> Ed*,  
Hearn, Baker, and Carithers,  
Prentice Hall, 2010. ISBN: 978-0136053583

- **Enrollment**

- If you are not enrolled, see me after class.

- **Web page**

- [www.cs.princeton.edu/courses/archive/spring22/cos426/](http://www.cs.princeton.edu/courses/archive/spring22/cos426/)



# COS 426: Computer Graphics

## Spring 2022

## Syllabus

### Description

Computer graphics is the intersection of computer science, geometry, physics, and art. This course will study topics in this broad and remarkable field, with an emphasis on practical methods and applications. In particular, the course will provide an extensive introduction to image processing, modeling, rendering, and computer animation. The goal of this course is to equip students with the various tools and techniques they need to build large projects with significant graphical components; this includes applications for realizing artistic visions (art and architecture), user interaction (UI/UX development), entertainment products (video games, CGI, animations, and augmented reality),

### Contents

[Syllabus](#)[Description](#)[Prerequisites](#)[Lectures and Precepts](#)[Required Reading](#)[Staff](#)[Precepts](#)

# Coursework



- Exam (15%)
  - In person, for now (March 3)
- Programming Assignments (60%)
  - Assignment #1: Image Processing
  - Assignment #2: Modeling
  - Assignment #3: Ray Tracer
  - Assignment #4: Rasterizer
  - Assignment #5: Animation
- Final Project (20%)
  - Your choice! (due Dean's Date)
- Participation (5%)

# Programming Assignments



- When?
  - Roughly every 2-3 weeks
- How?
  - Javascript
  - Some OpenGL (WebGL, GLSL)
- What?
  - Basic feature lists
  - Extra credit lists
  - Art contest

# Art and Simulation Contest

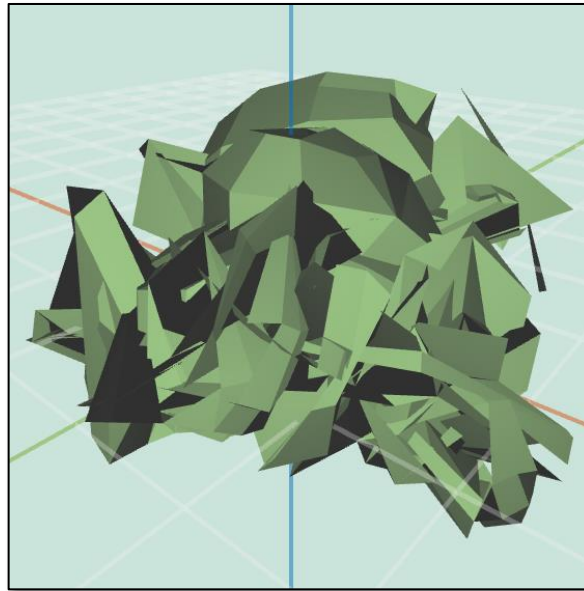


- Everybody should submit entries!
  - 1 point for submitting
  - 2 points for winning



Cool Images/Videos

*(Jad Bechara, CS 426, Spring 2018)*



Bloopers

*(Reed Tantiviramanond, CS 426, Spr15)*



Characters for web banner

# Collaboration Policy – See Website



- Overview:
  - You must type your own code, but may work in pairs.
  - You must reference your resources.
  - See policy on course web, and ask when in doubt.
- It's OK to ...
  - Talk with other students about ideas, approaches, etc.
  - Get ideas from information in books, wikipedia, etc.
  - Use “support” code provided with our assignments
- It's NOT OK to ...
  - Share your code digitally (e.g. post on web, email)
  - Copy code directly from another student
  - Leverage code acquired from other sources



# Questions / Discussion



- Piazza ([www.piazza.com](http://www.piazza.com))
  - View announcements
  - Post questions to the class
  - Answer other students questions
  - Set up for everyone enrolled as of today
  - Use this instead of email to instructors/TAs (can send private messages)



# Precepts

- When and Where

- Thu 7:30 - 8:20 pm
- Fri 10:00 - 10:50 am
- Fri 11:00- 11:50 am

We will poll precepts to see if other times are more convenient.

- <https://registrar.princeton.edu/course-offerings/course-details?term=1224&courseid=002073>
- Attend either as you prefer – they will be equivalent
- In person / over zoom potentially in some cases.

- Attendance

- Topics vary, so attend the ones that help you
- This week: getting up to speed in Javascript

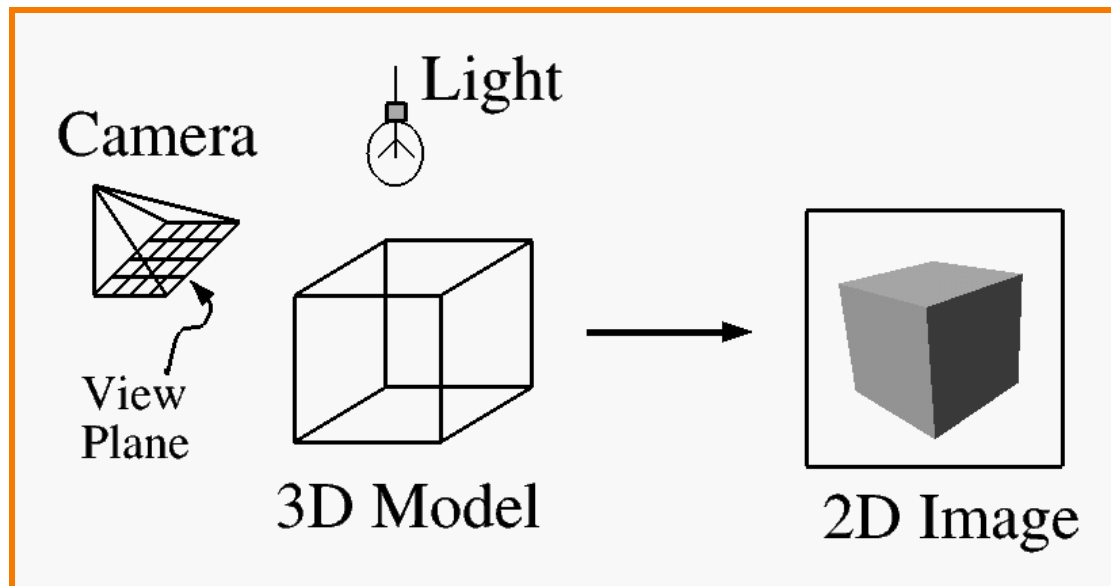
# Overview



- Administrative stuff
  - People, times, places, etc.
- Syllabus
  - What will I learn in this course?
- Imaging
  - Getting started ...

# Introduction

- What is computer graphics?
  - Imaging = *representing 2D images*
  - Modeling = *representing 3D objects*
  - Rendering = *constructing 2D images from 3D models*
  - Animation = *simulating changes over time*



# Syllabus



I. Imaging

II. Modeling

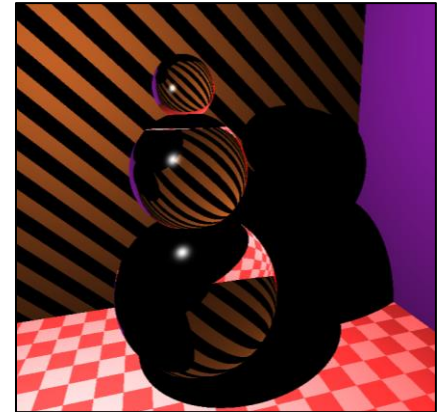
III. Rendering

IV. Animation



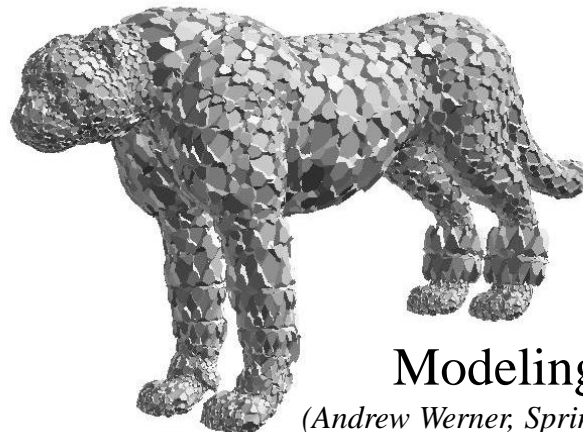
Image Processing

*(Rusty Coleman, CS426, Fall99)*



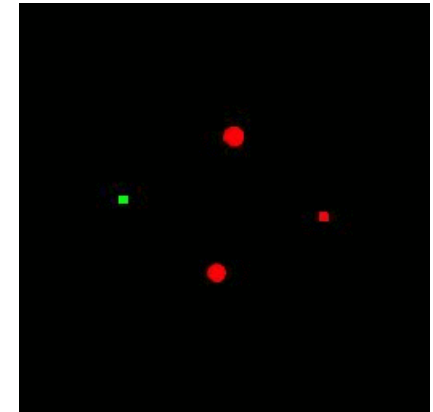
Rendering

*(David Paulk, CS426, Spr2015)*



Modeling

*(Andrew Werner, Spring 2014)*



Animation

*(Riley Thomasson, Spring 2014)*

# Part I: Imaging



- Image Basics
  - Definition
  - Color models
- Image Representation
  - Sampling
  - Reconstruction
  - Quantization & Aliasing
- Image Processing
  - Filtering
  - Warping
  - Composition
  - Morphing



Image Composition  
(Michael Bostock, CS426, Fall99)



Image Morphing  
(Reilly Bova, CS426 Fall 2018)

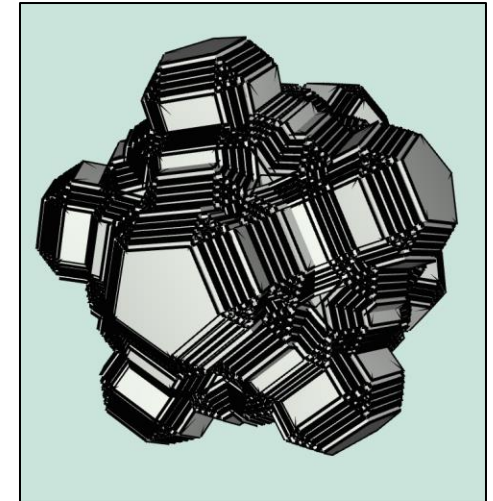
# Part II: Modeling



- Representations of geometry
  - Curves: splines
  - Surfaces: meshes, splines, subdivision
  - Solids: voxels, CSG, BSP
- Procedural modeling
  - Sweeps
  - Fractals
  - Grammars



*(Brendan Chou, Spring 2014)*



*(John Whelchel,  
CS 426, Spr2015)*

# Part III: Rendering



- 3D Rendering Pipeline
  - Modeling transformations
  - Viewing transformations
  - Hidden surface removal
  - Illumination, shading, and textures
  - Scan conversion, clipping
  - Hierarchical scene graphics
  - OpenGL
- Global illumination
  - Ray tracing
  - Radiosity



Pixel Shading  
*(Final Fantasy, Square Pictures)*

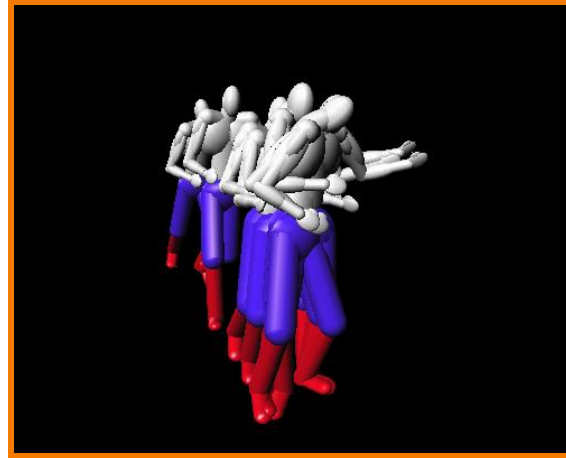


Global Illumination  
*(Diana Liao, CS 426, Spr15)*



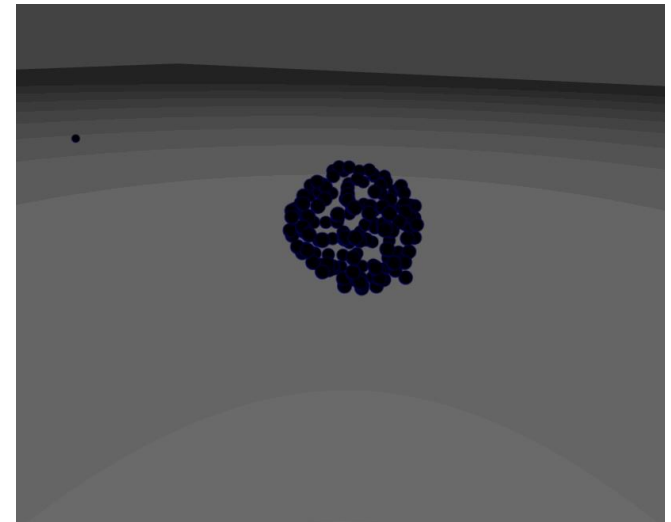
# Part IV: Animation

- Keyframing
  - Kinematics
  - Articulated figures
- Motion capture
  - Capture
  - Warping
- Dynamics
  - Physically-based simulations
  - Particle systems
- Behaviors
  - Planning, learning, etc.



Dancing Guy  
(Jon Beyer, CS426, Spr05)

Particle system  
(Drew Wallace, Spring 2015)



# Applications



## → Entertainment

- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art



Geri's Game  
(Pixar Animation Studios)



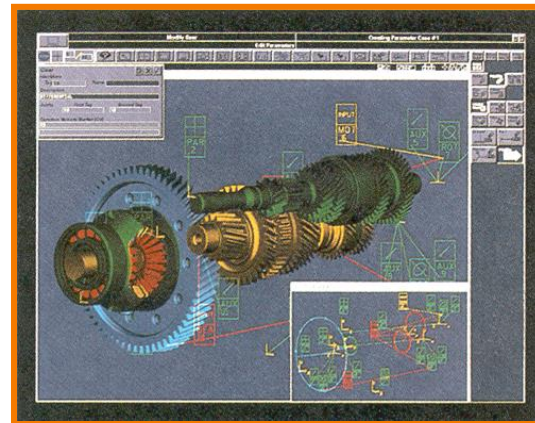
# Applications



- Entertainment
- ➔ **Computer-aided design**
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art



Los Angeles Airport  
(Bill Jepson, UCLA)



Gear Shaft Design  
(Intergraph Corporation)

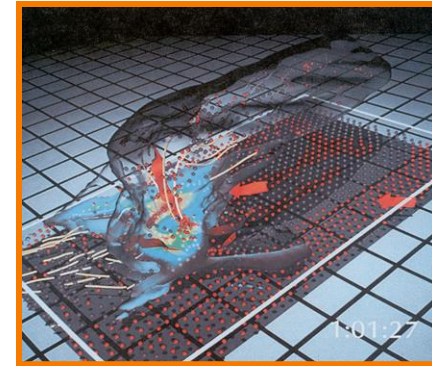


Boeing 777 Airplane  
(Boeing Corporation)

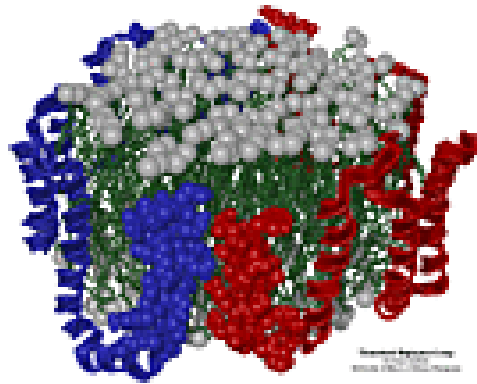
# Applications



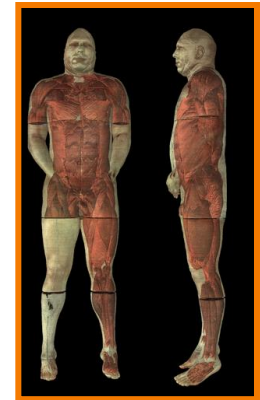
- Entertainment
- Computer-aided design
- ➔ **Scientific visualization**
- Training
- Education
- E-commerce
- Computer art



Airflow Inside a Thunderstorm  
(Bob Wilhelmson,  
University of Illinois at Urbana-Champaign)



Apo A-1  
(Theoretical Biophysics Group,  
University of Illinois at Urbana-Champaign)

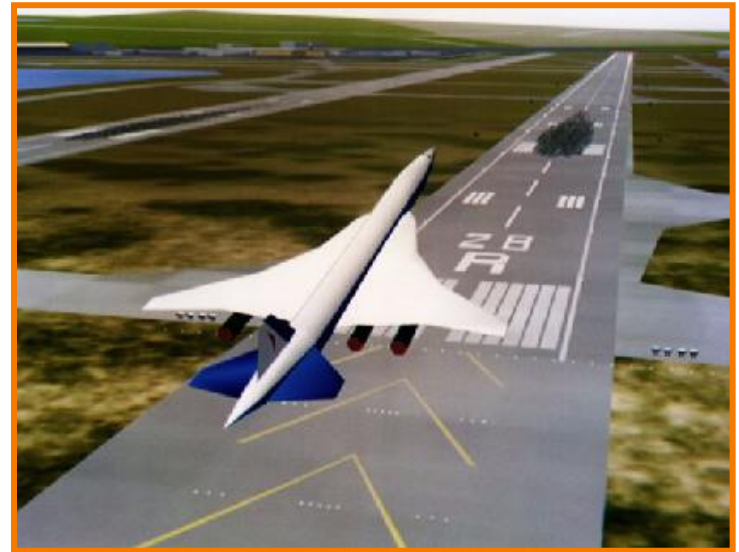


Visible Human  
(National Library of Medicine)

# Applications



- Entertainment
- Computer-aided design
- Scientific visualization
- ➔ **Training**
- Education
- E-commerce
- Computer art



Early Flight Simulation  
(NASA)

# Applications

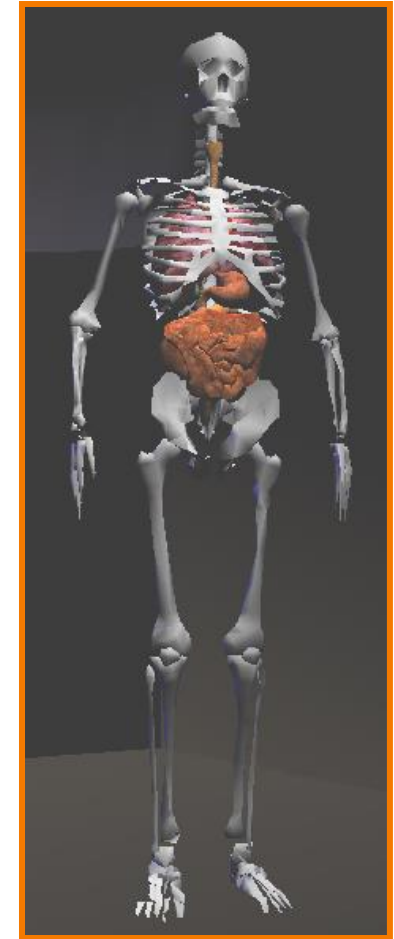
- Entertainment
- Computer-aided design
- Scientific visualization
- Training

## ➔ Education

- E-commerce
- Computer art



Forum of Trajan  
(Bill Jepson, UCLA)



Human Skeleton  
(SGI)

# Applications



- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- ➔ **E-commerce**
- Computer art



Virtual Stores  
(Matterport)

# Applications



- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- ➔ **Computer art**



Blair Arch  
(Marissa Range '98)



# Overview



- Administrative stuff
  - People, times, places, etc.
- Syllabus
  - What will I learn in this course?
- **Imaging**
  - **Let's get started ...**

# What is an Image?



# What is an Image?

An image is a 2D rectilinear array of pixels

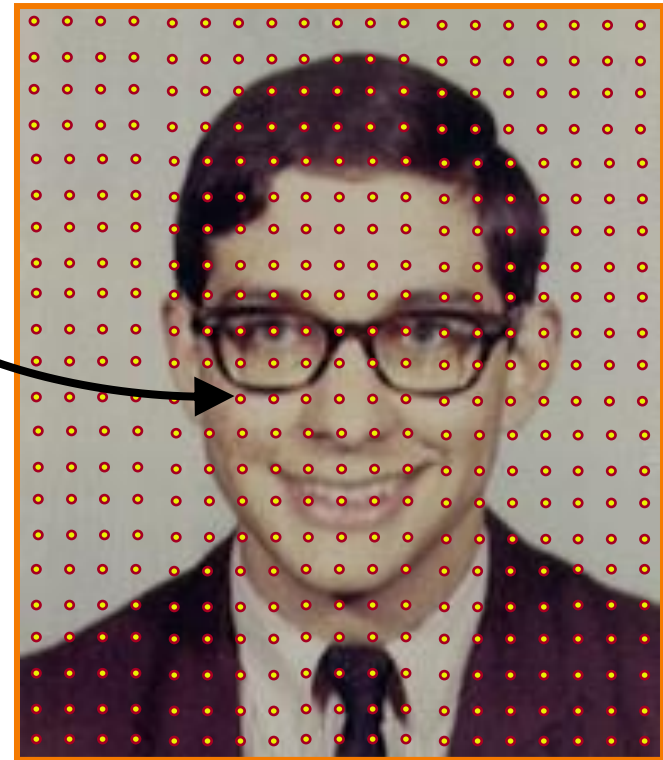
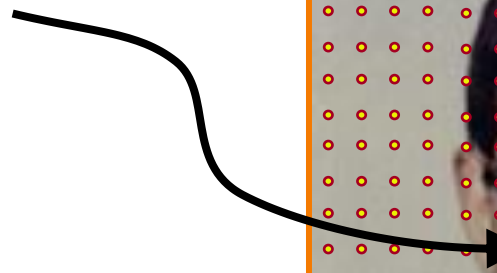


Digital image

# What is a Pixel?



**Pixel**



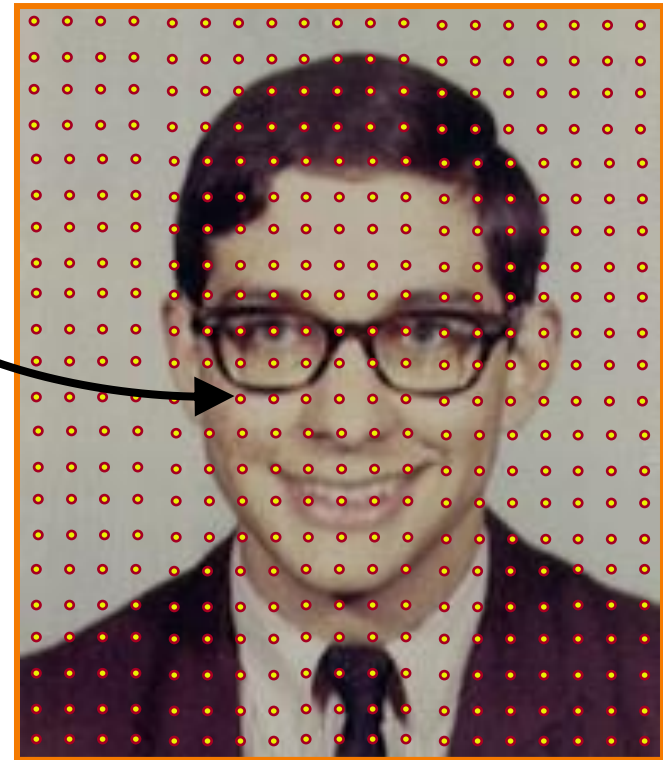
Digital image

# What is a Pixel?



Sample of a function at a position

$I(x,y)$



Digital image

# What Function?

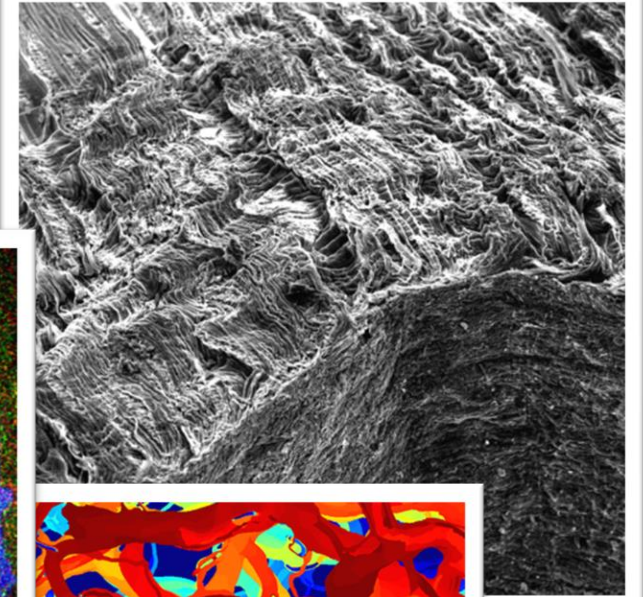


# What Function?

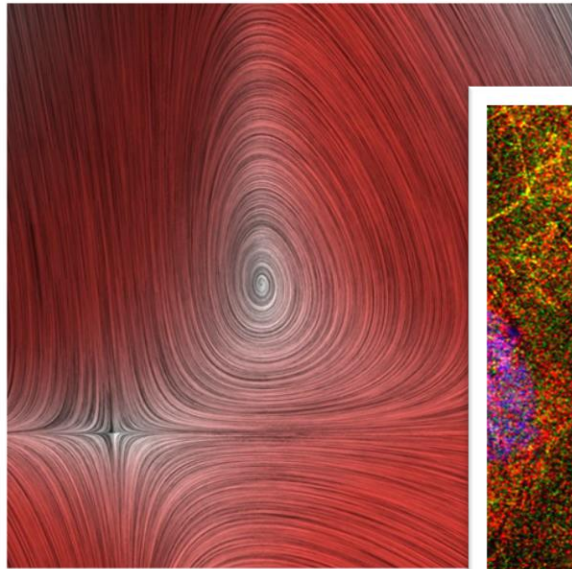


Could be any function ...

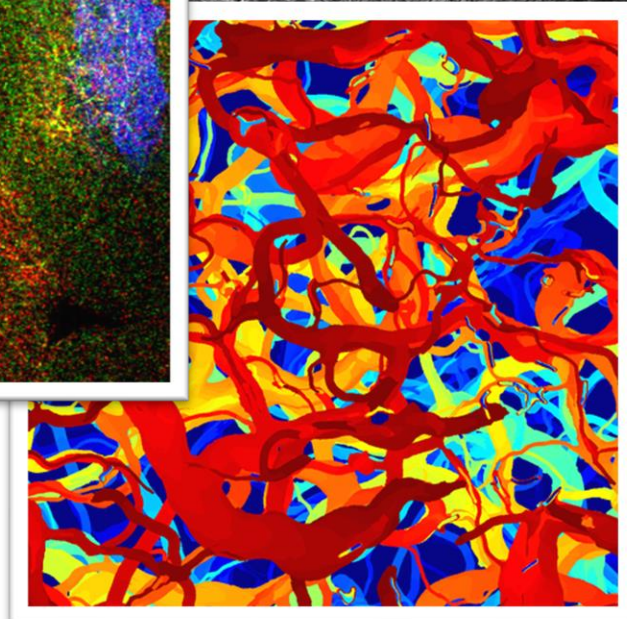
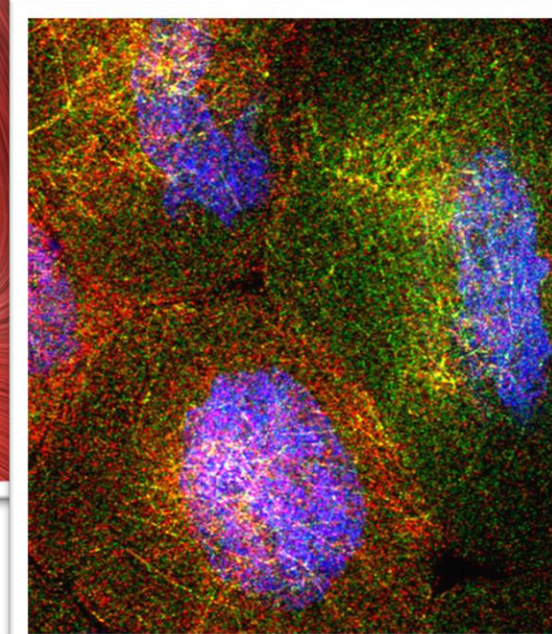
Shawn C. Little  
Kristina S. Sinsimer  
Elizabeth R. Gavis



Michael  
Kosk



Mitchell A. Nahmias  
Paul R. Prucnal



Mingzhai Sun  
Joshua Shaevitz

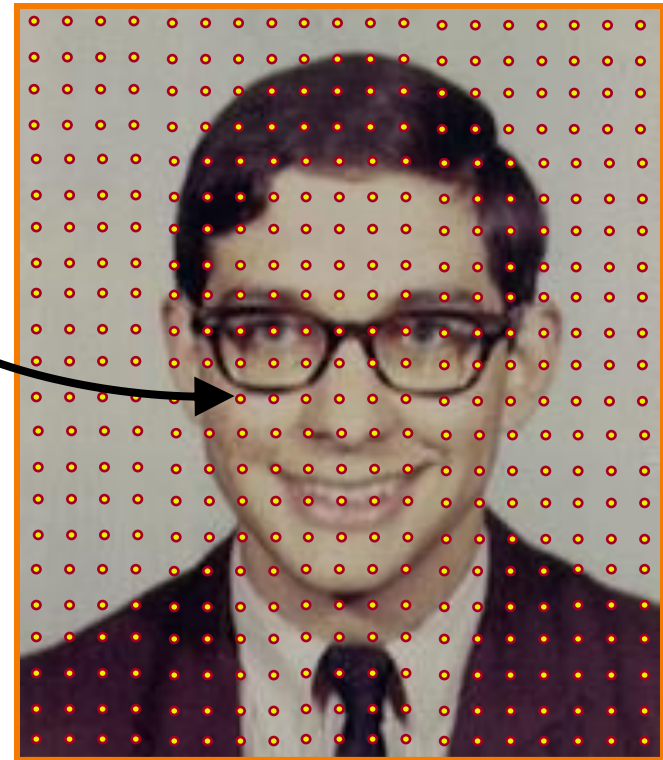
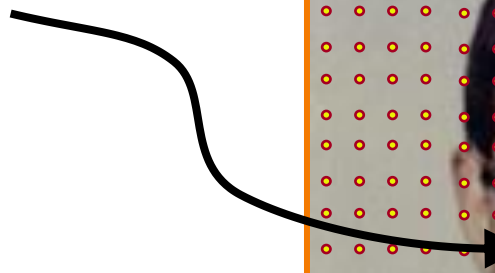
Art of Science  
(Friend Center hallway)

# What Function?



What about photographic images?

$I(x,y)$



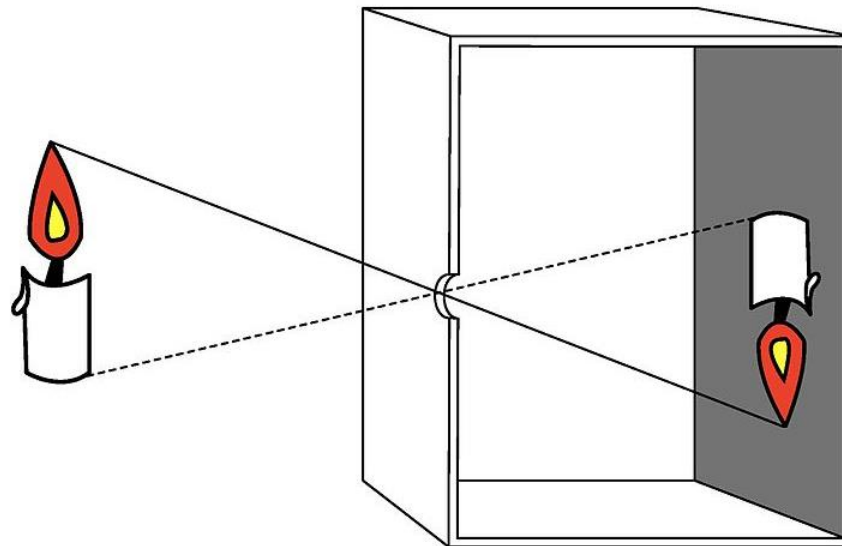
Digital photograph



# Plenoptic Function



Each pixel of a photographic image is a function of radiance arriving at a sensor.

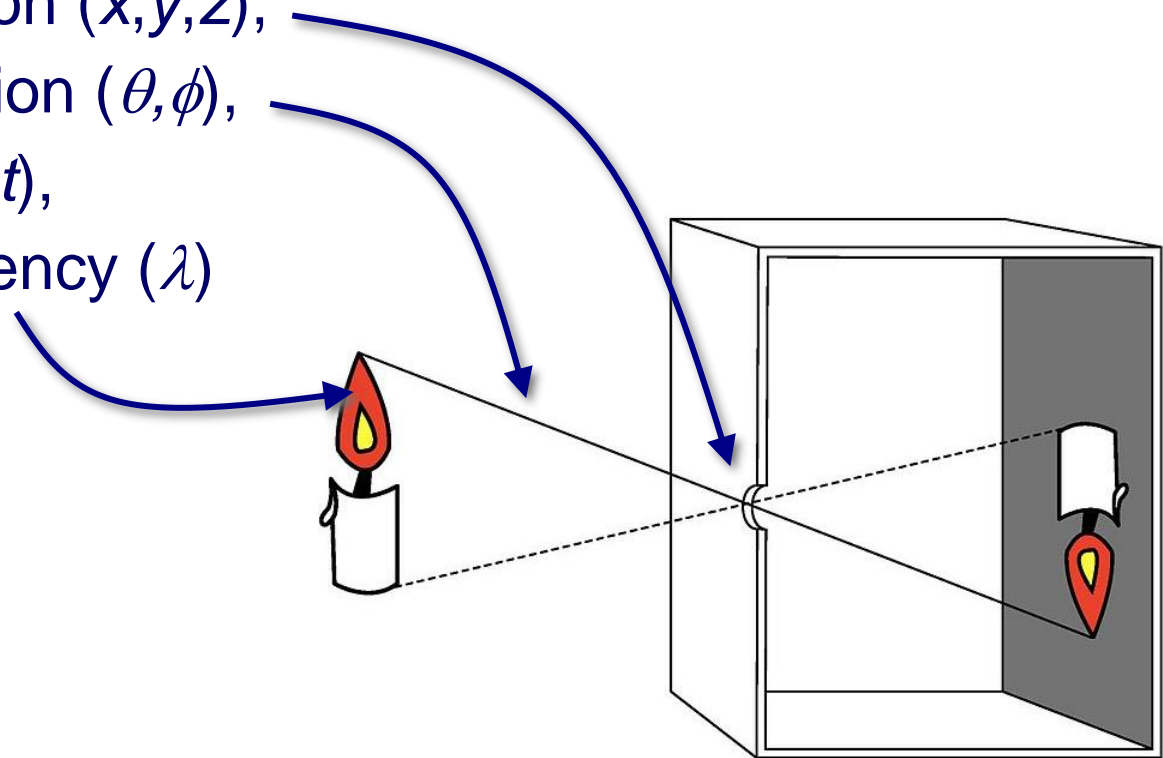


*Carlo Benini*

# Plenoptic Function

The 7D plenoptic function  $L(x, y, z, \theta, \phi, t, \lambda)$  describes the radiance arriving ...

- at any position  $(x, y, z)$ ,
- in any direction  $(\theta, \phi)$ ,
- at any time  $(t)$ ,
- at any frequency  $(\lambda)$

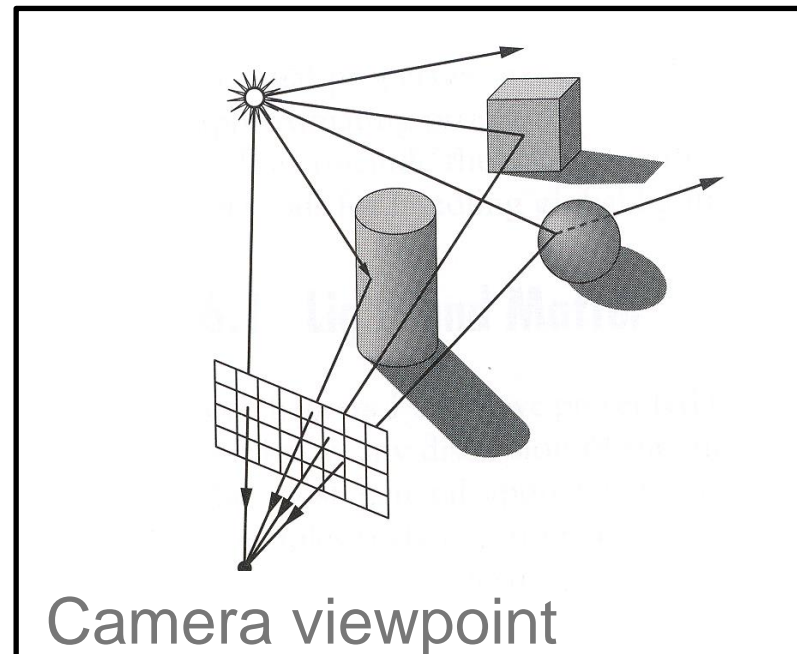


*Carlo Benini*

# Photographic Images

An idealized photographic image contains a 2D array of samples of the 7D plenoptic function

- at a particular camera viewpoint,
- for a 2D array of directions,
- at a certain time,
- at certain frequencies

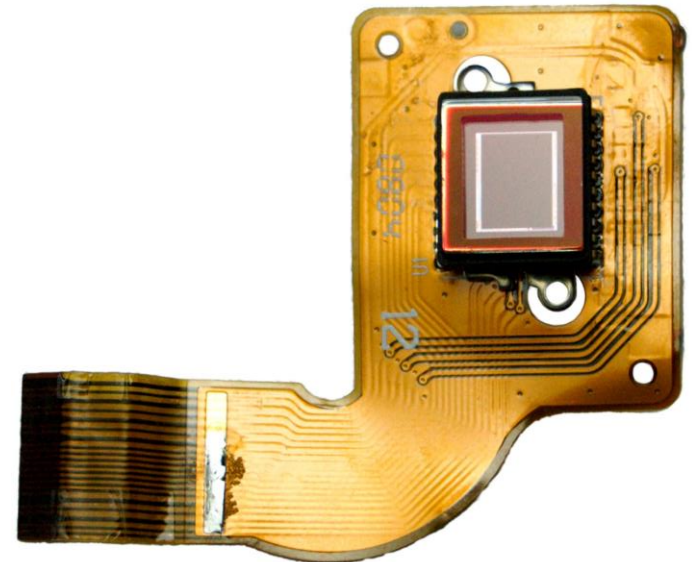
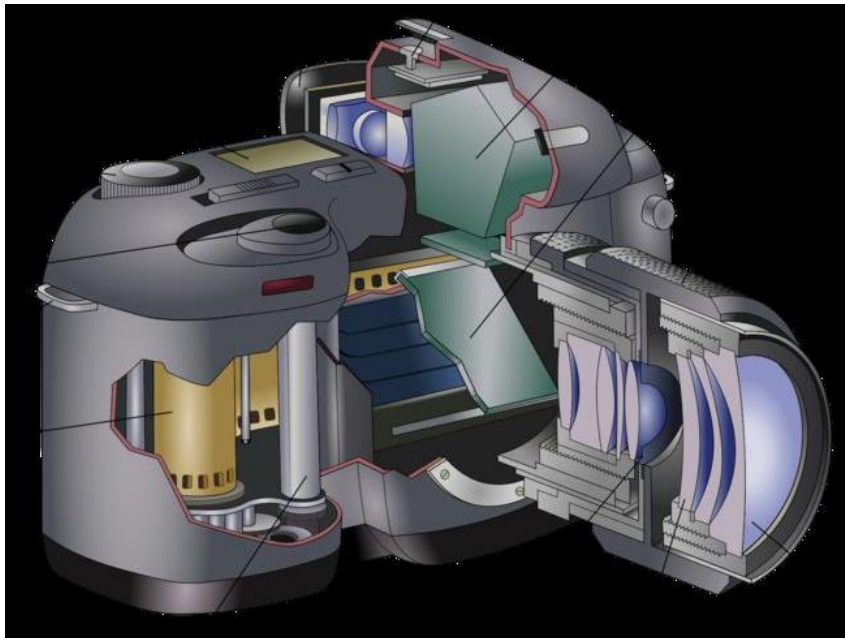
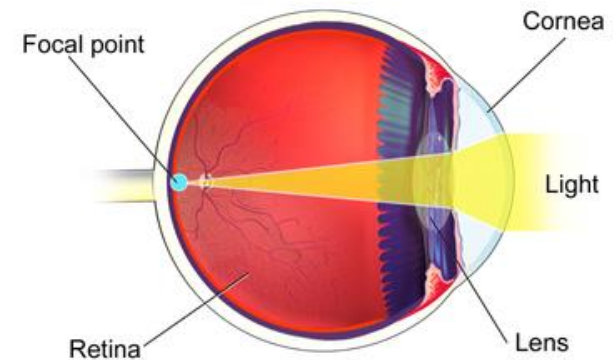


# Photographic Images



In practice, can't measure plenoptic function directly

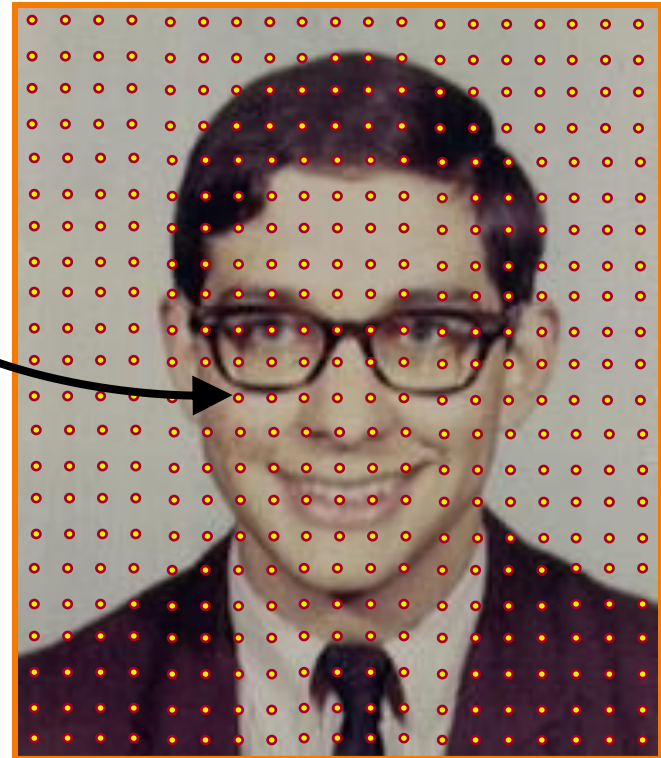
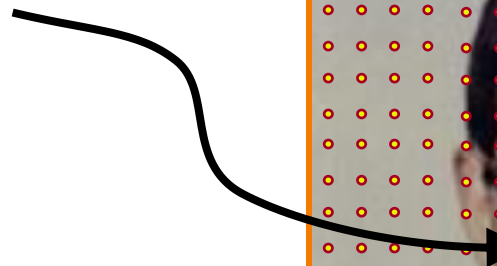
- Photoreceptors in eye
- Film in a traditional camera
- CCD cells in digital camera



# Photographic Images

Photographic pixels as finite samples of the plenoptic function

$$f(x, y, z, \theta, \phi, t, \lambda)$$



Digital photograph

# What Frequencies / Wavelengths?



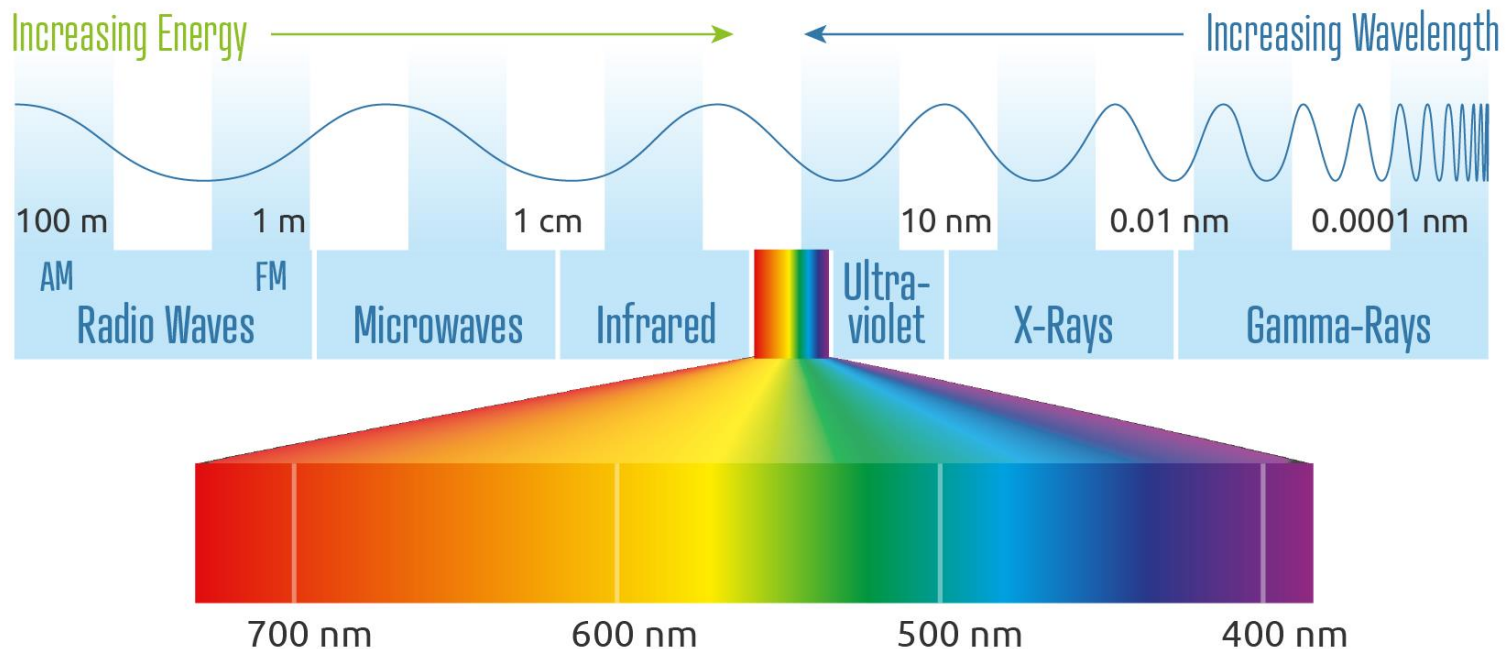
$$f(x, y, z, \theta, \phi, t, \lambda)$$

Corresponding frequency is:  $\frac{c}{\lambda}$



# Electromagnetic Spectrum

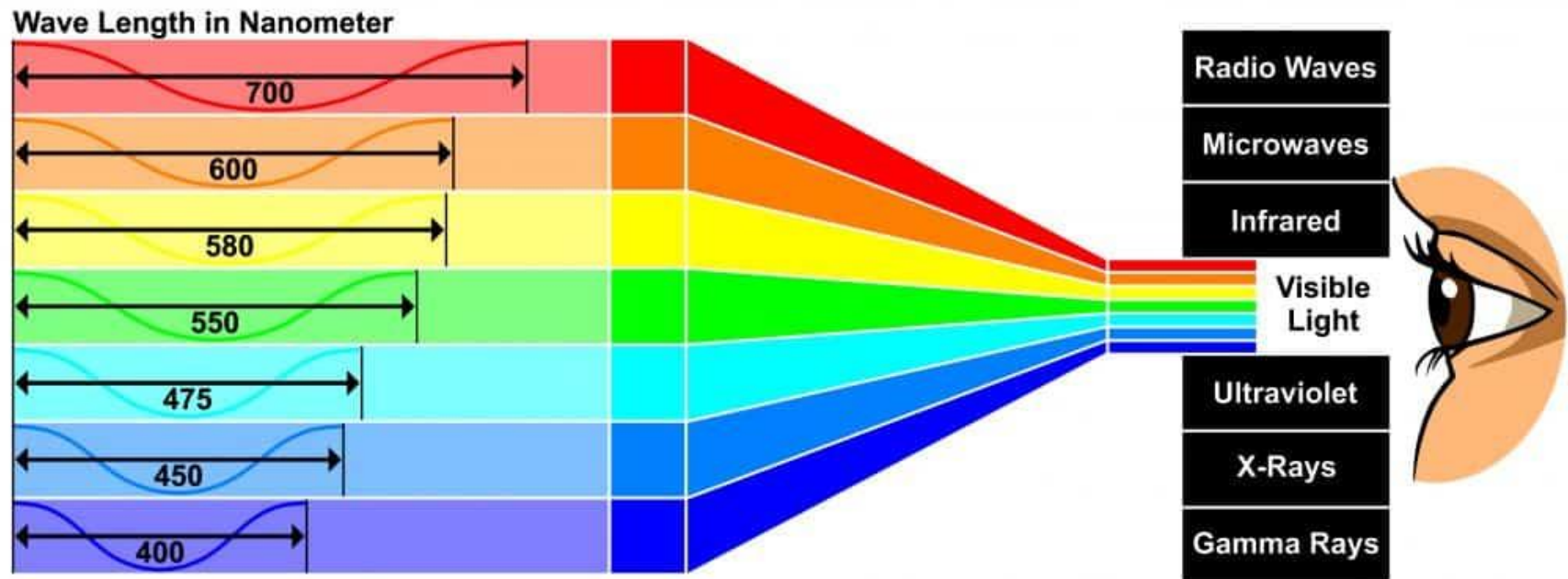
- Visible light frequencies range between ...
  - Red =  $4.3 \times 10^{14}$  hertz (700nm)
  - Violet =  $7.5 \times 10^{14}$  hertz (400nm)





# Electromagnetic Spectrum

- Visible light frequencies range between ...
  - Red =  $4.3 \times 10^{14}$  hertz (700nm)



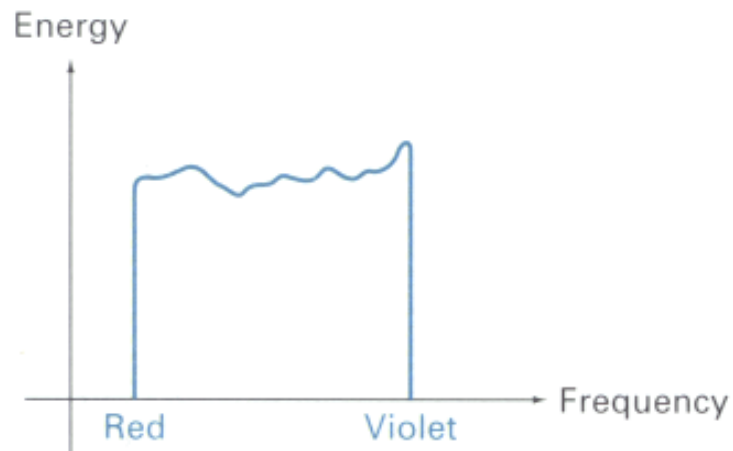
Figures 15.1 from H&B



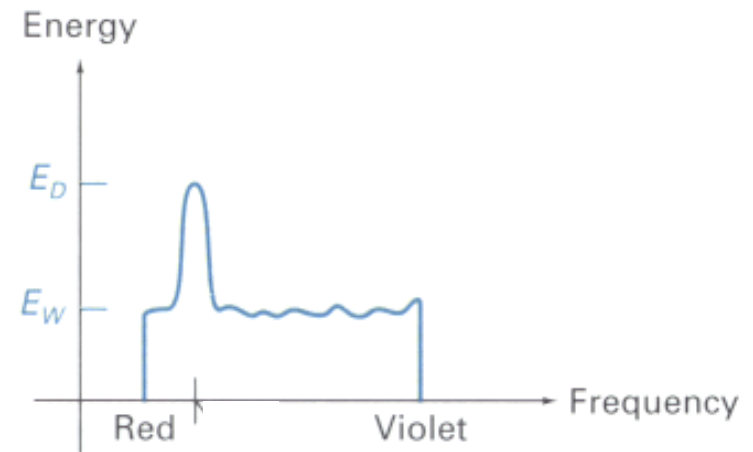
# Color



- The color of light is characterized by its **spectrum**
  - Magnitude of energy at every visible frequency



White Light



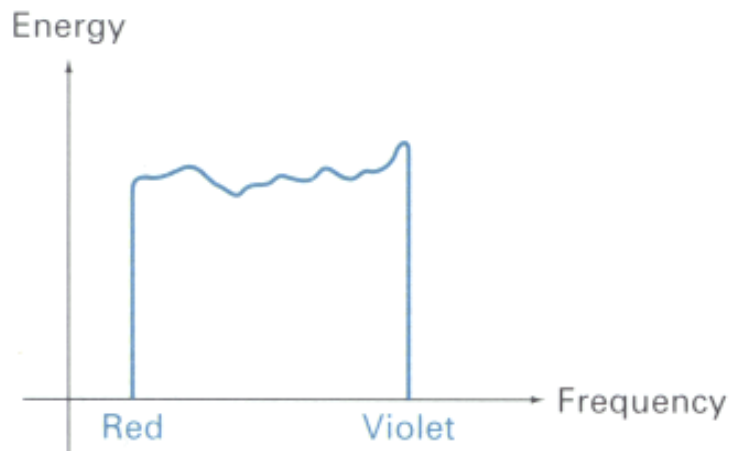
Orange Light

# Color

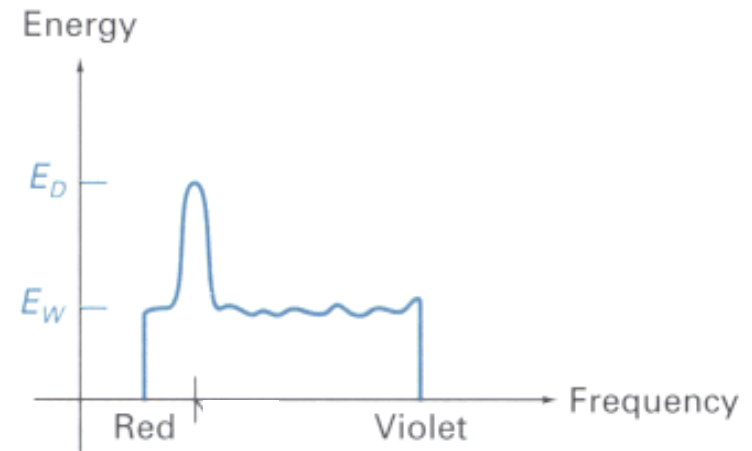


How do we represent a color in a computer?

Must store a finite amount of data to represent magnitudes for infinite number of frequencies

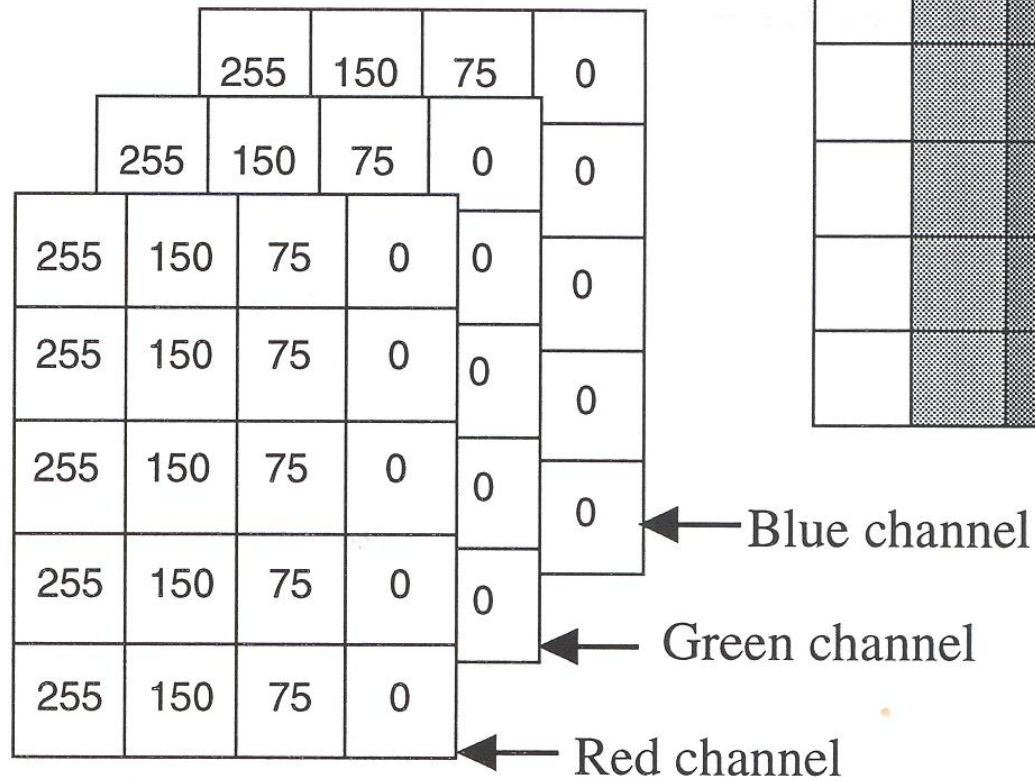


White Light



Orange Light

# Color Frame Buffer



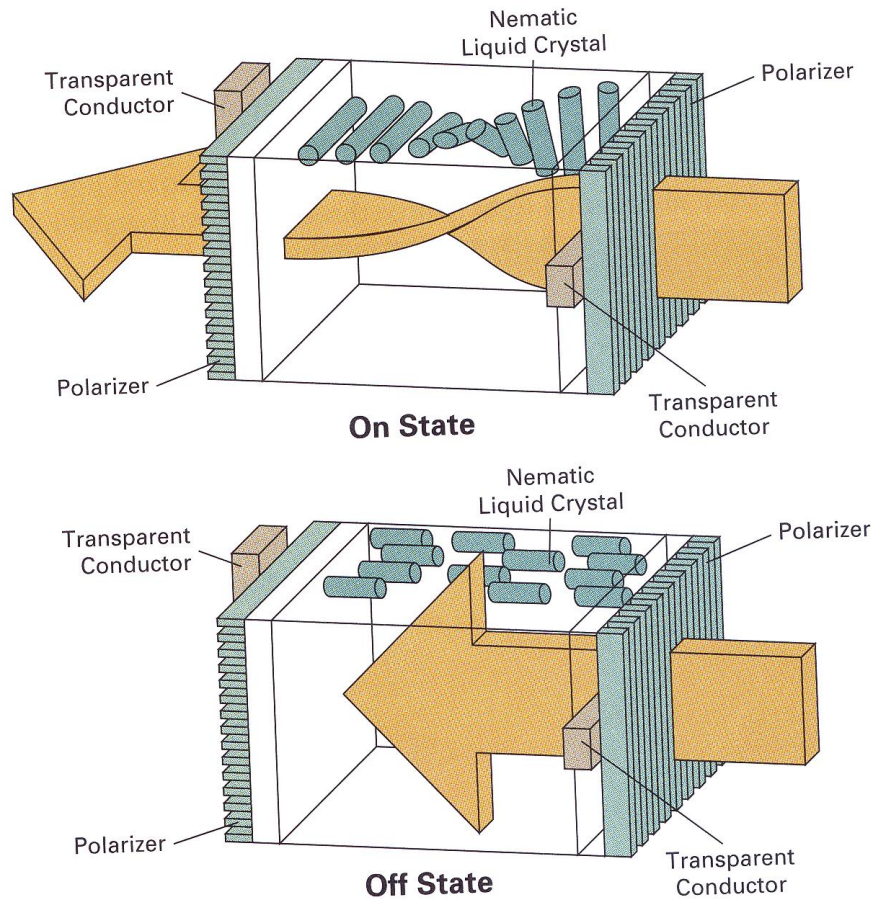


# Frame Buffer Display

- Video display devices
  - Liquid Crystal Display (LCD)
  - Thin-film electroluminescent displays
  - Light-emitting diodes (LED)
- Hard-copy devices
  - Ink-jet printer
  - Laser printer
  - Film recorder
  - Electrostatic printer
  - Pen plotter

# Frame Buffer Display

Example: liquid crystal display (LCD)



Polarization:

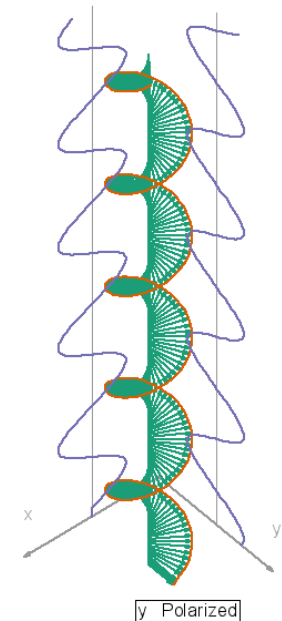
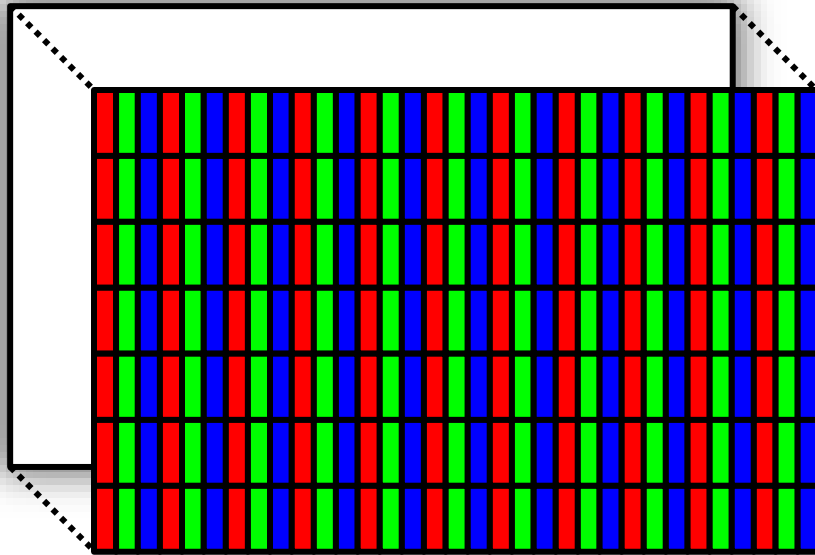


Figure 2.16 from H&B

# Frame Buffer Display

LCD up close



Colors are interleaved

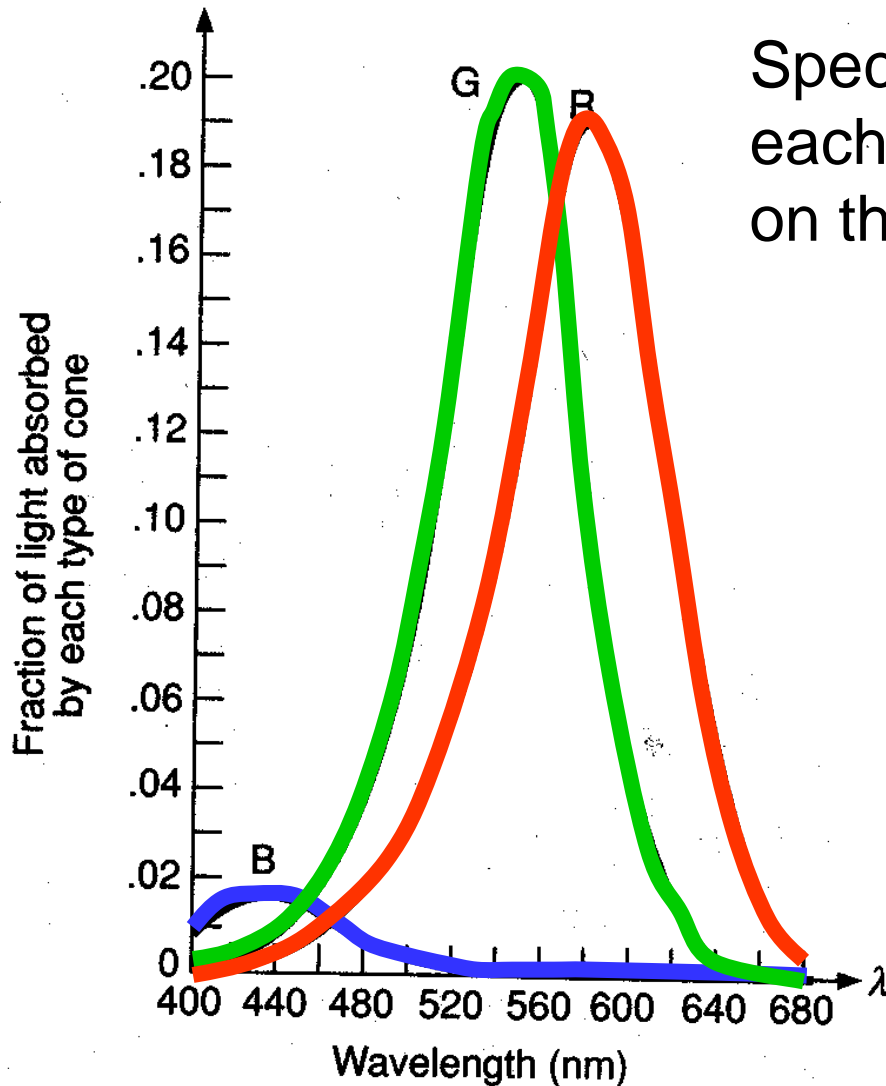


# Color



Why red, green, and blue (RGB)?

# Human Color Perception



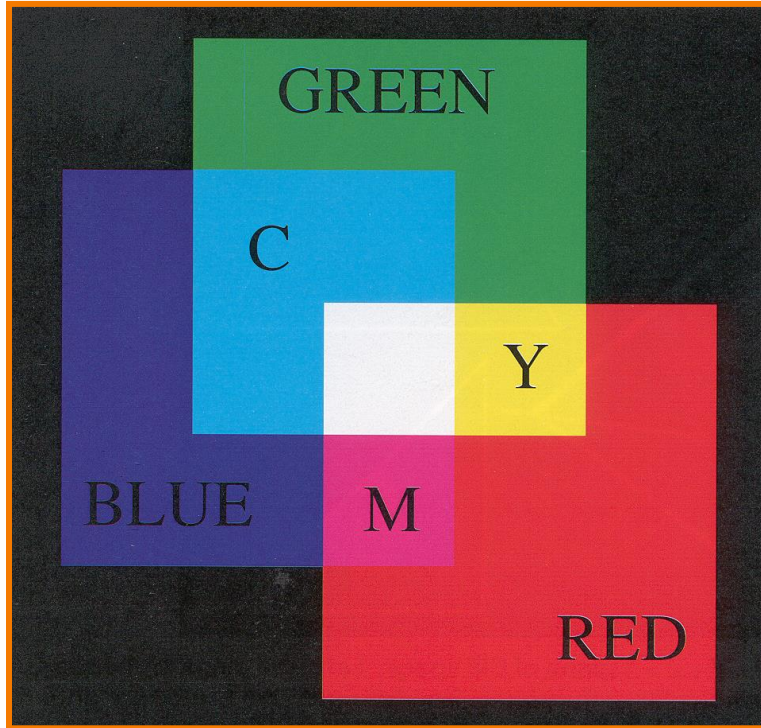
Spectral-response functions of each of the three types of cones on the human retina.

Tristimulus  
theory of color


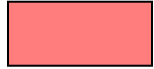


Figure 13.18 from FvDFH



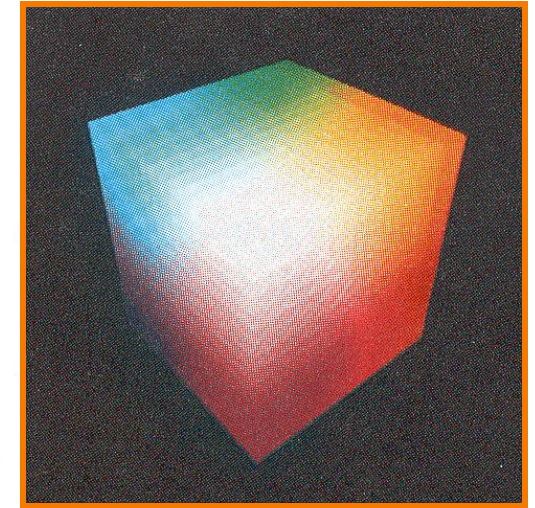
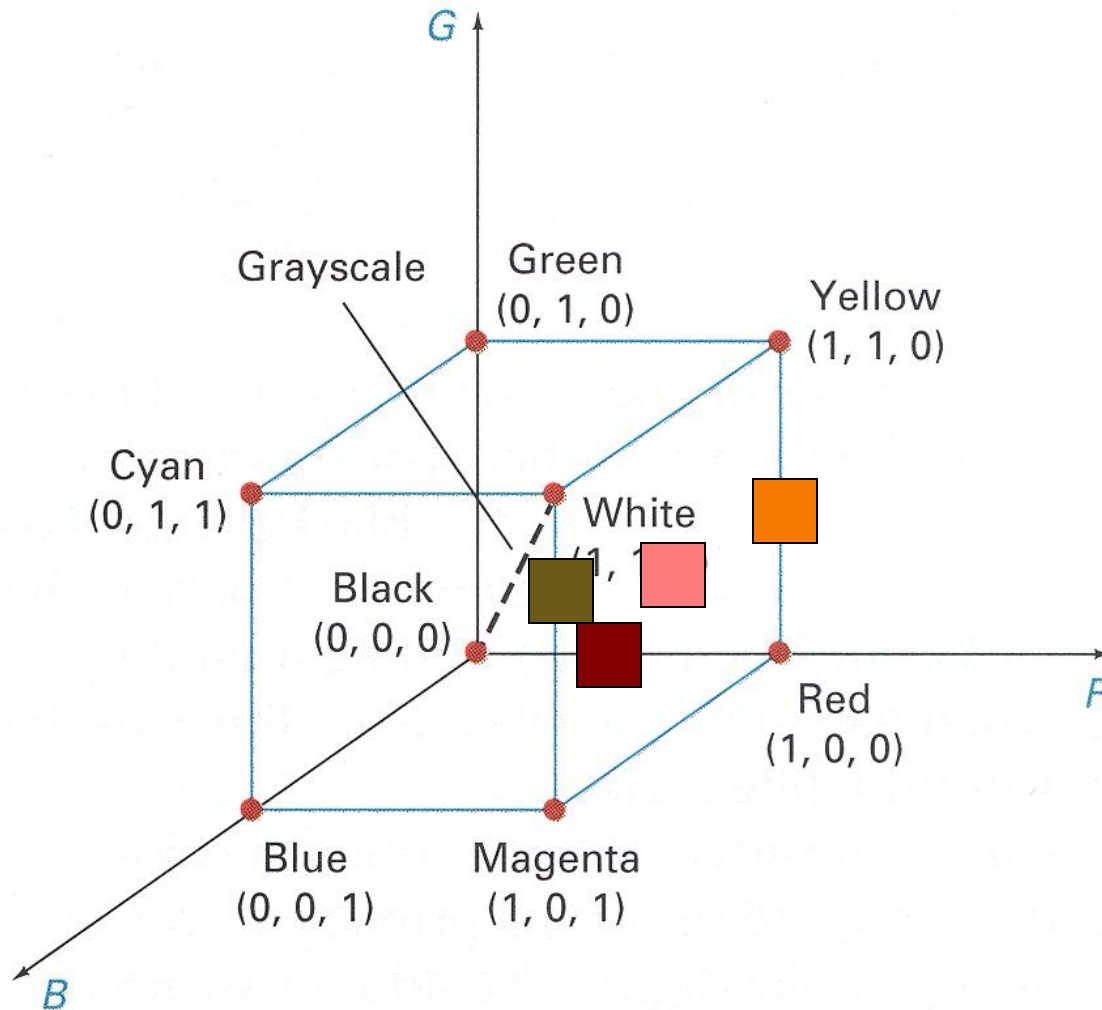
# RGB Color Model



Colors are additive

<b>R</b>	<b>G</b>	<b>B</b>	<b>Color</b>
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
0.5	0.0	0.0	? 
1.0	0.5	0.5	? 
1.0	0.5	0.0	? 
0.5	0.3	0.1	? 

# RGB Color Cube



Figures 15.11&15.12 from H&B

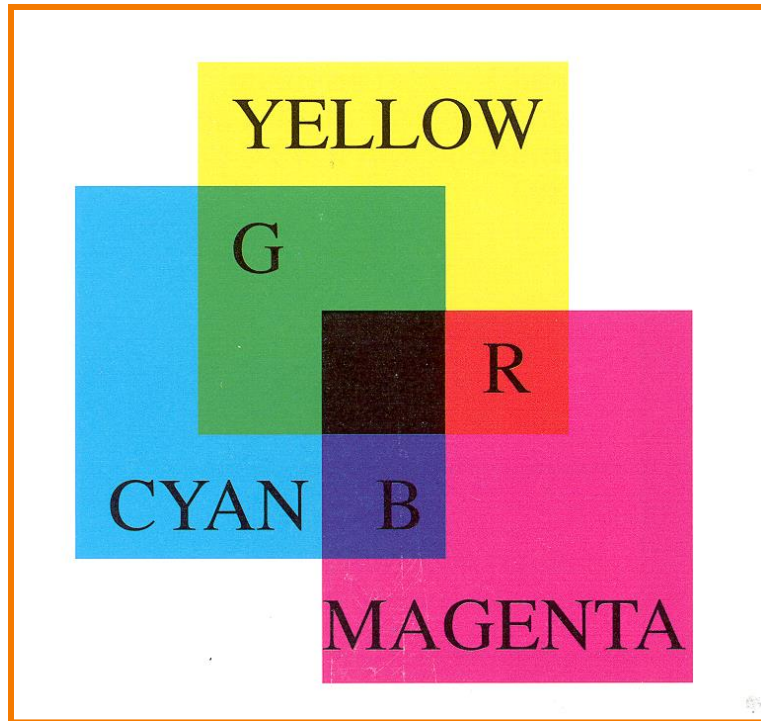


# Other Color Models




- CMY
- HSV
- XYZ
- La\*b\*
- Others

Different color models are useful for different purposes

# CMY Color Model



Useful for printers  
because 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	

# CMY Color Model

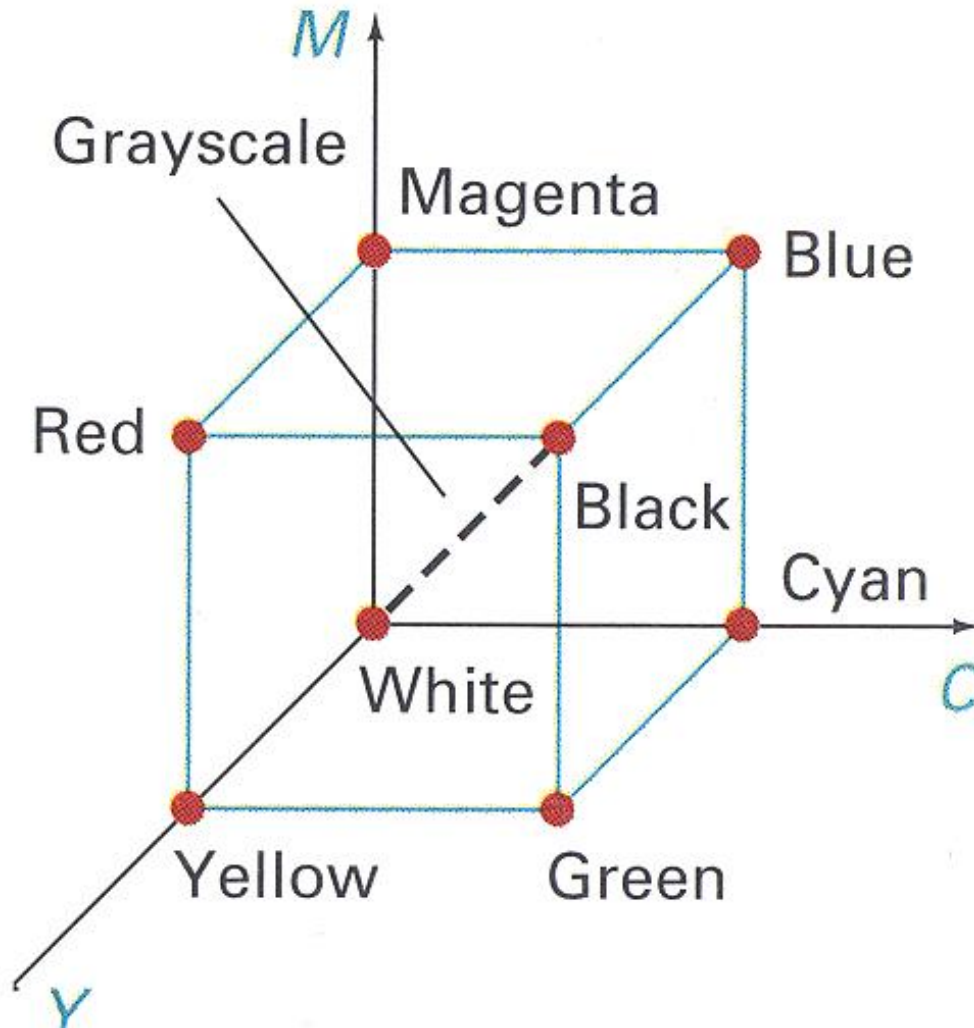
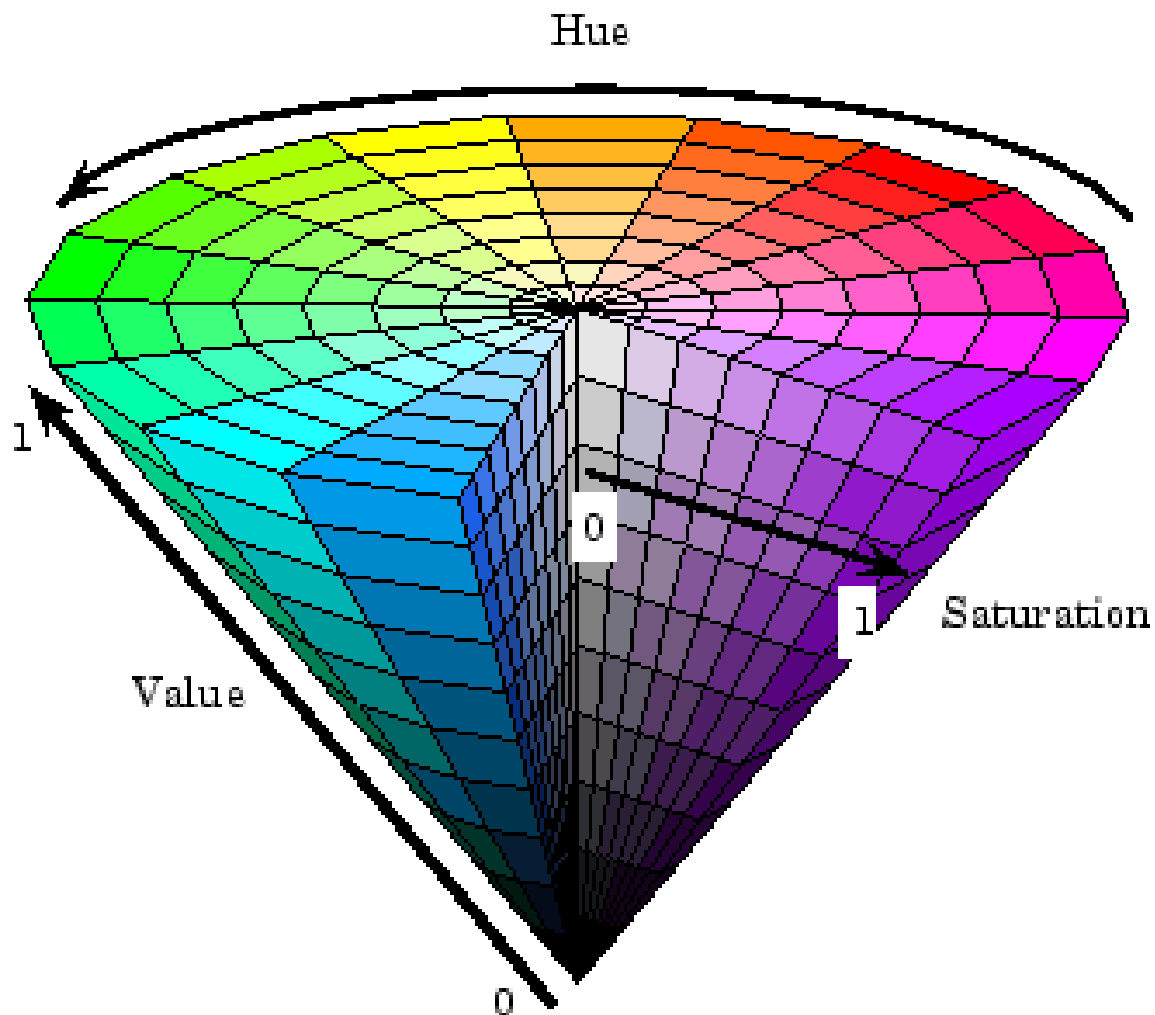


Figure 15.14 from H&B

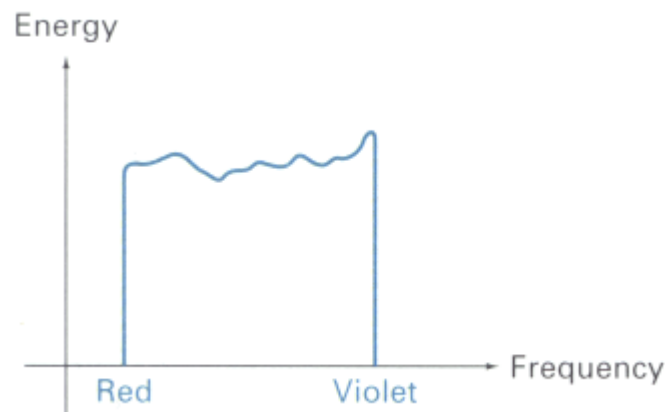
# HSV Color Model



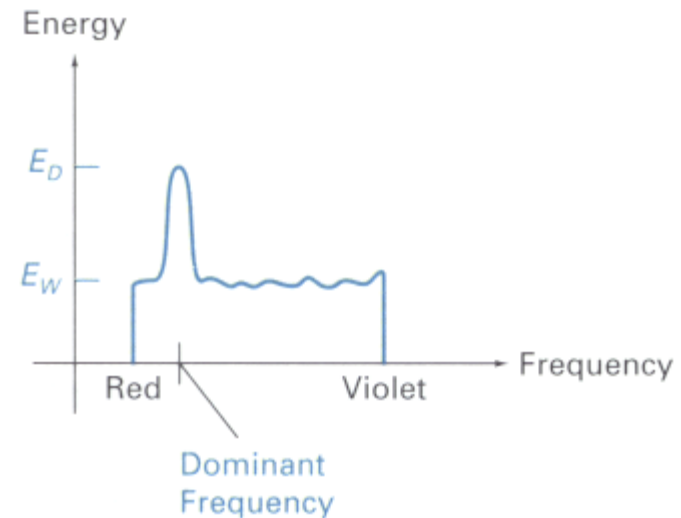


# HSV Color Model

- HSV interpretation in terms of color spectrum
  - Hue = dominant frequency (highest peak)
  - Saturation = excitation purity (ratio of highest to rest)
  - Value = luminance (area under curve)

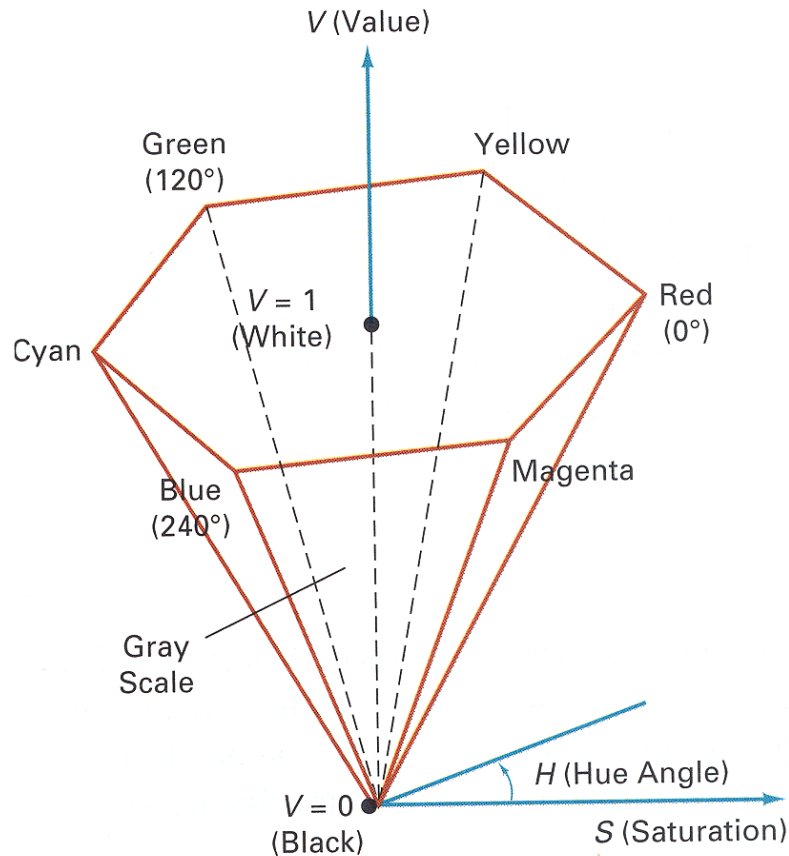





White Light



Orange Light

# HSV Color Model



H	S	V	Color
0	1.0	1.0	Red
120	1.0	1.0	Green
240	1.0	1.0	Blue
*	0.0	1.0	White
*	0.0	0.5	Gray
*	*	0.0	Black
60	1.0	1.0	
270	0.5	1.0	
270	0.0	0.7	

Useful for user interfaces  
because dimensions are intuitive

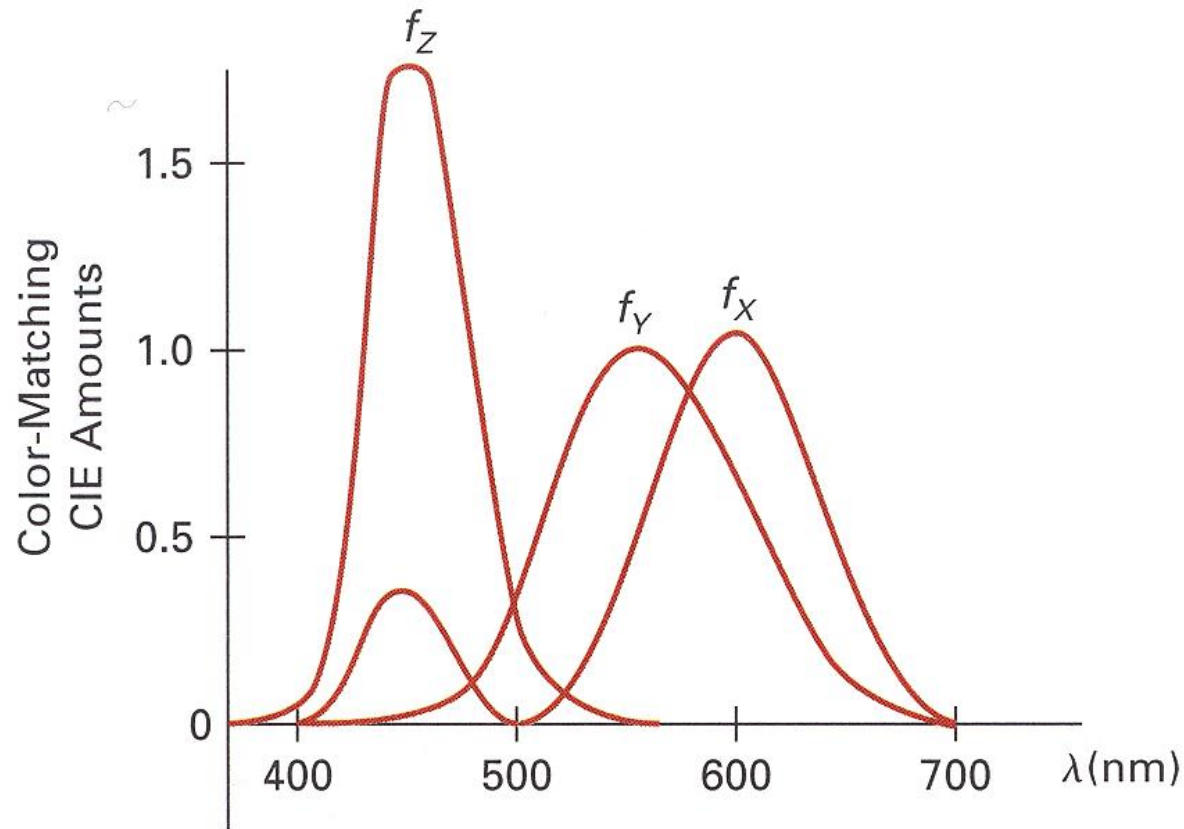
Figure 15.16&15.17 from H&B



# XYZ Color Model (CIE)



Figure 15.6 from H&B



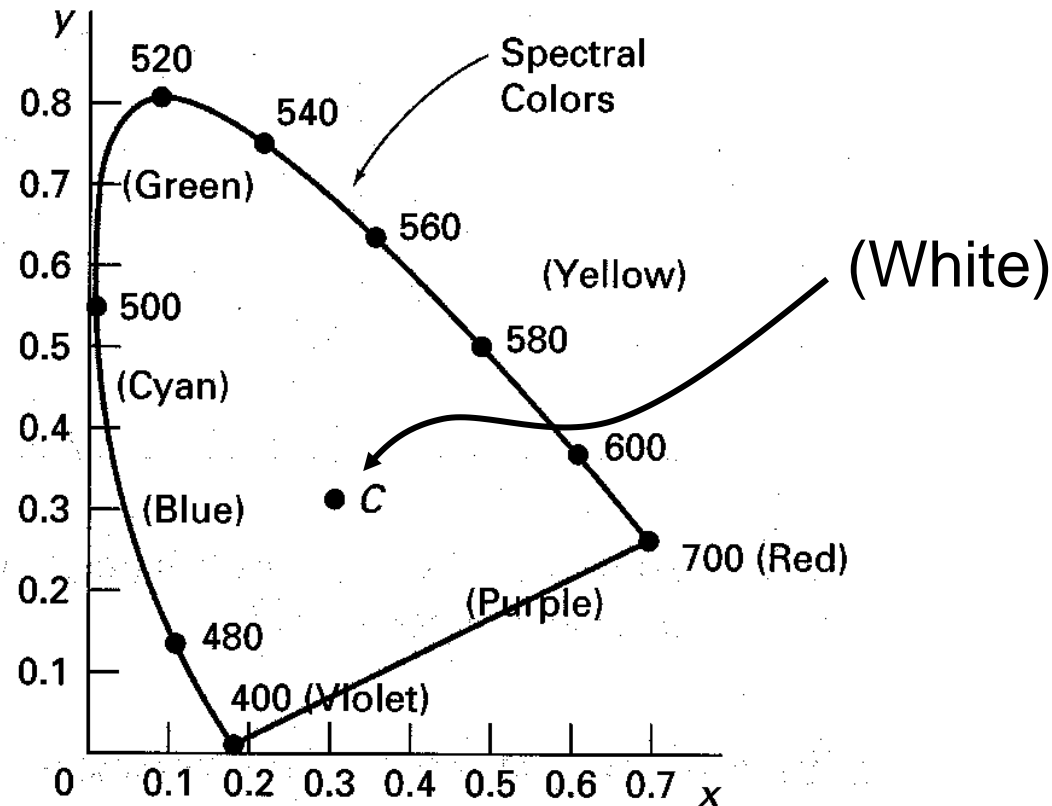
Derived from perceptual experiments

All spectra that map to same XYZ give same visual sensation

# XYZ Color Model (CIE)



Figure 15.7 from H&B

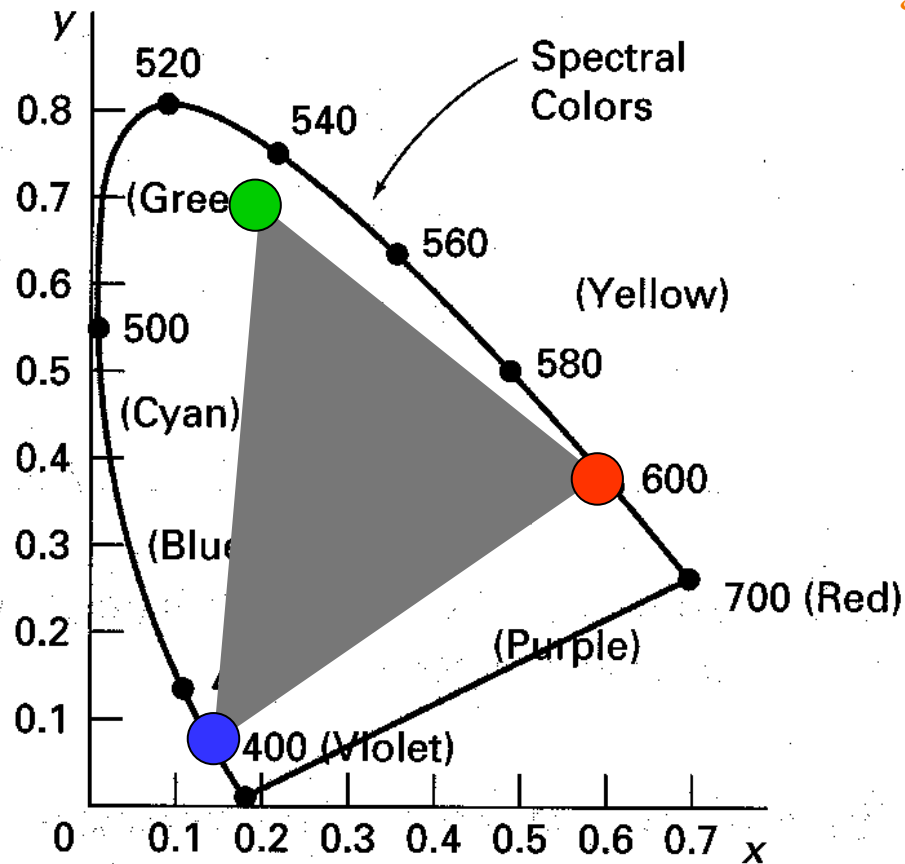


Normalized amounts of X and Y for colors in visible spectrum

# XYZ Color Model (CIE)



Figure 15.13 from H&B

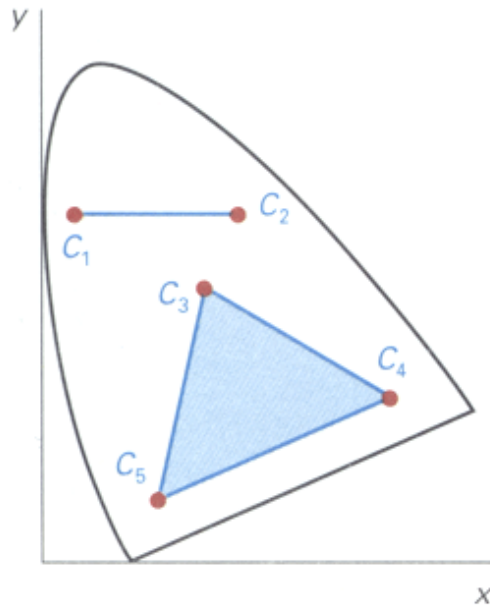


Useful for reasoning about coverage of color gamuts

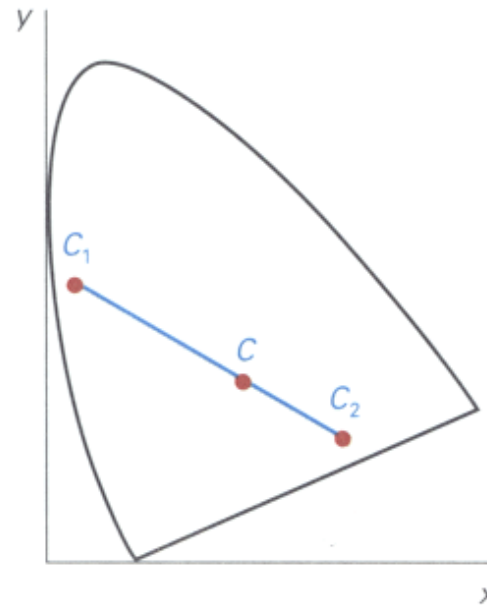
# XYZ Color Model (CIE)



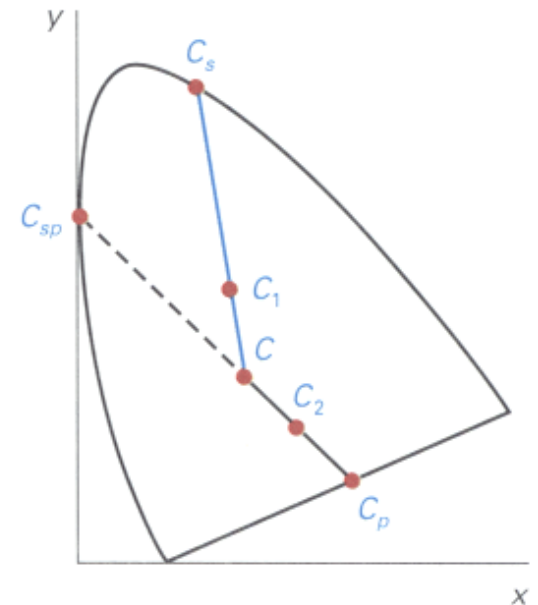
Figures 15.8-10 from H&B



Compare  
Color  
Gamuts



Identify  
Complementary  
Colors

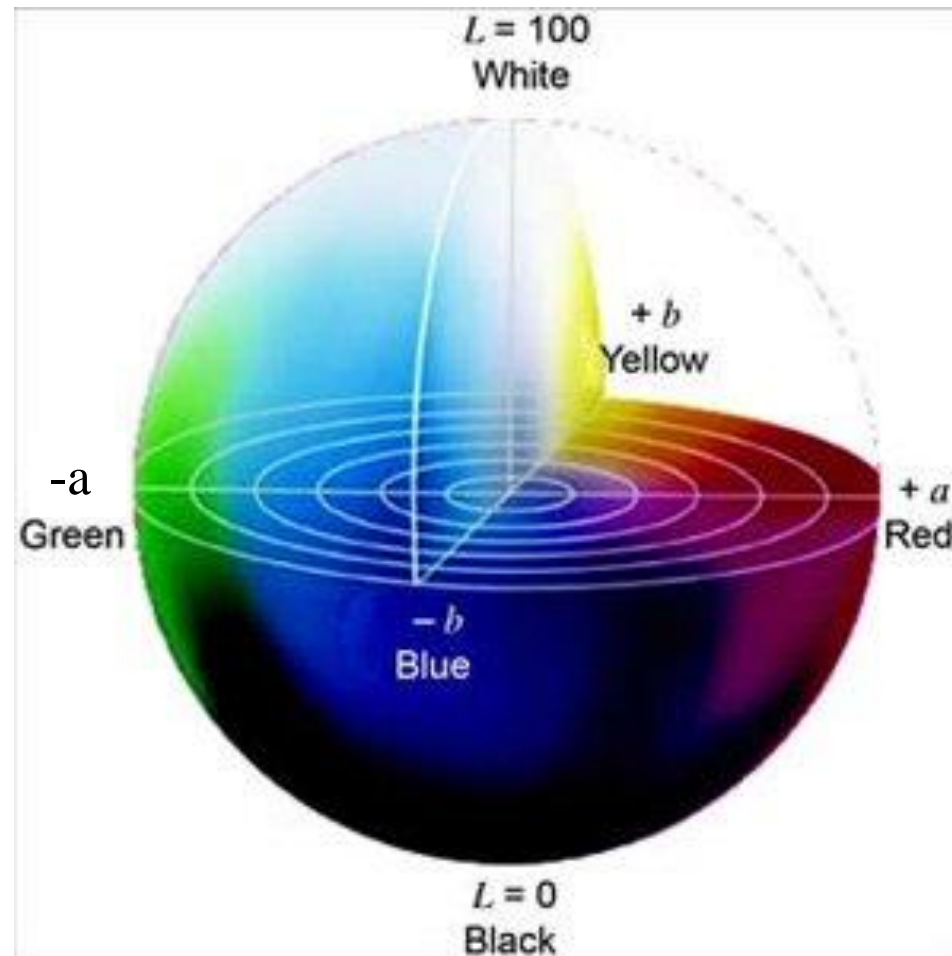


Determine  
Dominant Wavelength  
and Purity

Useful for characterizing perceptual qualities of colors

# La\*b\* Color Model

Non-linear  
compression  
of XYZ  
color space  
based on  
perception



Useful for measuring perceptual differences between colors

# Summary



- Images
  - Pixels are samples
  - Photographs sample plenoptic function
- Colors
  - Tristimulus theory of color
  - Different color models for different devices, uses, etc.
  - RGB model is common due to human perception
  - CIE Chromaticity Diagram
- Next time
  - Image processing!