Computer Graphics

Adam Finkelstein

Princeton University

COS 426, Spring 2015

Overview

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

Administrative Stuff

- Instructors
  - Adam Finkelstein
  - Huiwen Chang and Maciej Halber
- Book
  - *Computer Graphics with OpenGL, 4^{th} Ed,*
    Hearn, Baker, and Carithers,
    Prentice Hall, 2010. ISBN: 978-0136053583
- Web page
  - www.cs.princeton.edu/courses/archive/spr15/cos426/

www.cs.princeton.edu/courses/archive/spr15/cos426/index.php

COS 426:
Computer Graphics
Spring 2015

General | Syllabus | Coursework

General Information

Description: This course will study topics in computer graphics, covering methods in image processing, modeling, rendering, and animation.

Prerequisites: The course is appropriate for students who have taken COS217 and COS226 (or the equivalent). Javascript will be used as the main programming language.

Coursework: The grade will be based on four programming assignments (50%), two exams (25%), a final project (25%).

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

Instructors: Professor Adam Finkelstein with TAs: Huiwen Chang and Maciej Halber.

Students: requires PU login

Time/place: Lecture: Tue & Thu 3-4:20pm, Friend 006
Precepts: Wed 7:30-8:30pm, Friend 008.
Office Hours: TBA

Questions: We will use Piazza to handle Q&A this semester. Please post your questions there instead of mailing the staff, if possible.

Coursework

- Exams (25%)
  - In class (3/12 and 4/30)
- Programming Assignments (50%)
  - Assignment #1: Image Processing (due 2/22)
  - Assignment #2: Modeling (due 3/8)
  - Assignment #3: Rendering (due 4/12)
  - Assignment #4: Animation (due 4/26)
- Final Project (25%)
  - Game! (due at end of semester)

COS 426:
Computer Graphics
Spring 2015

General | Syllabus | Coursework

Assignments

There will be 4 programming assignments:

- Assignment #1: Image Processing (due Sun 2/22)
- Assignment #2: Modeling (due Sun 3/8)
- Assignment #3: Rendering (due Sun 4/12)
- Assignment #4: Animation (due Sun 4/26)

These will be linked as the assignments go online.

Exams

There will be 2 exams. Both are closed-book. However, you may bring a 8.5x11" cheat-sheet with writing on both sides, if you wish. Examples of previous exams can be found here.

- Exam #1: Everything before spring break (in class on Thurs March 12)
- Exam #2: Everything after spring break (in class on Thurs April 30)

Programming Resources

Assignments will be implemented in javascript. Any computer and browser may be used for development though we expect to

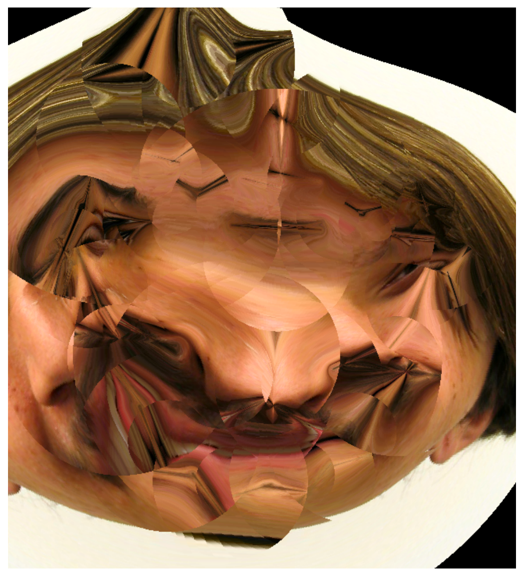
Programming Assignments

- When?
  - Roughly every 2-3 weeks
- Where?
  - Anywhere you want, e.g. home or clusters
- How?
  - Javascript
  - Some OpenGL
- What?
  - Basic feature lists
  - Extra credit lists
  - Art contest

Art Contest

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

Cool Images/Videos
(Jimmy Zuber, CS 426, Spring 2014)



Bloopers
(Alex Combs, CS 426, Spr05)

Characters for web banner

Collaboration Policy

- Overview:
  - You must write your own code
  - You must not leverage code written by others
  - You must reference your resources
- 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 …
  - Show your code to another student (e.g. post on web)
  - Look at code written by another student
  - Leverage code acquired from other sources

Questions / Discussion

- Piazza (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

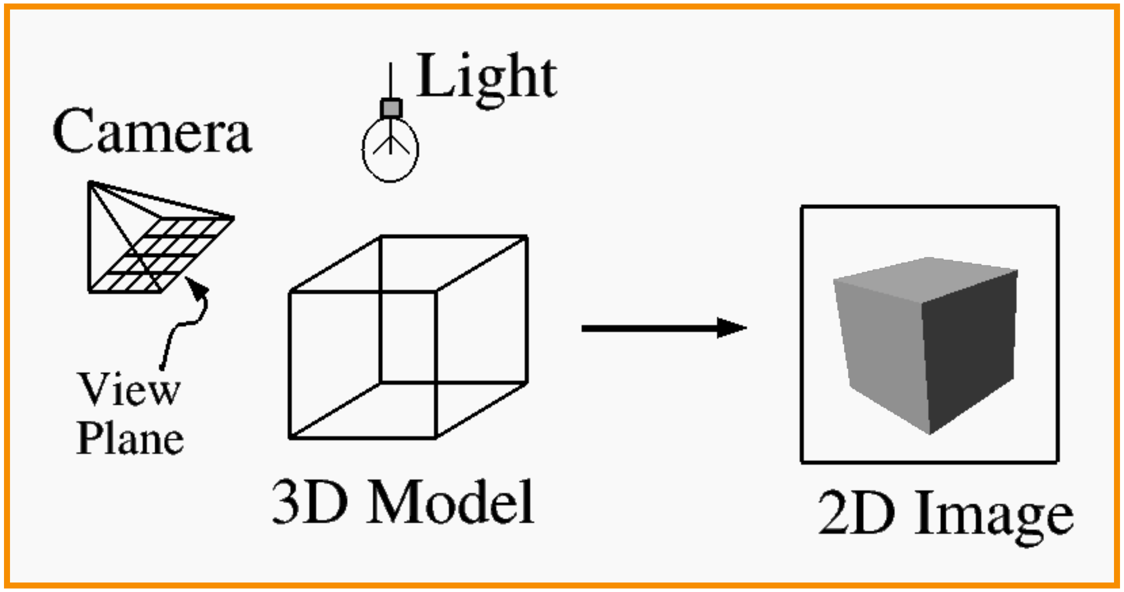
- Schedule
  - Wed 7:30-8:30 (Friend 008 – just down the hall)

Overview

- Administative 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

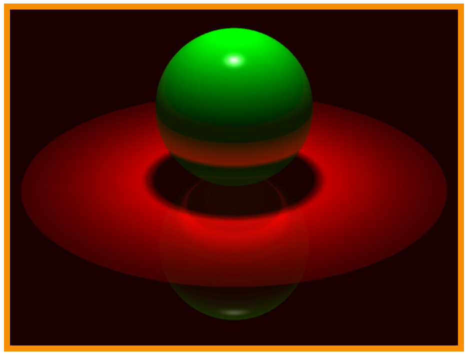
I. Imaging

II. Modeling

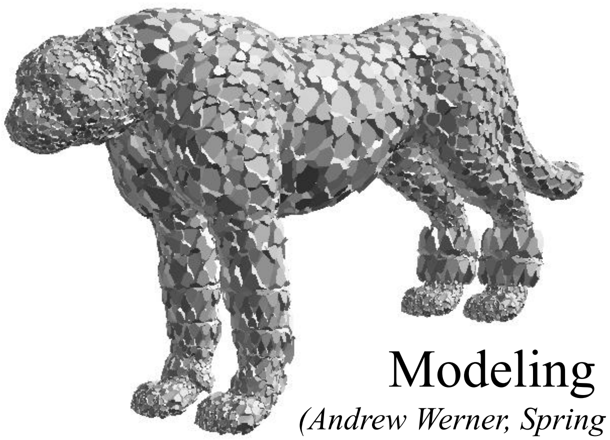
III. Rendering

IV. Animation

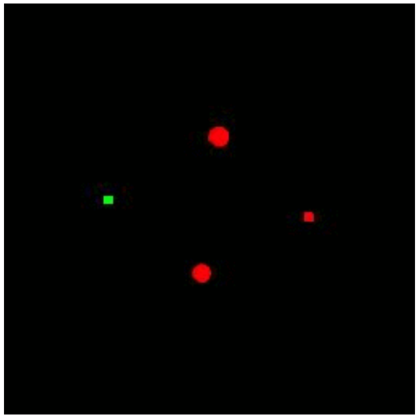
Image Processing
(Rusty Coleman, CS426, Fall99)



Rendering
(Michael Bostock, CS426, Fall99)



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

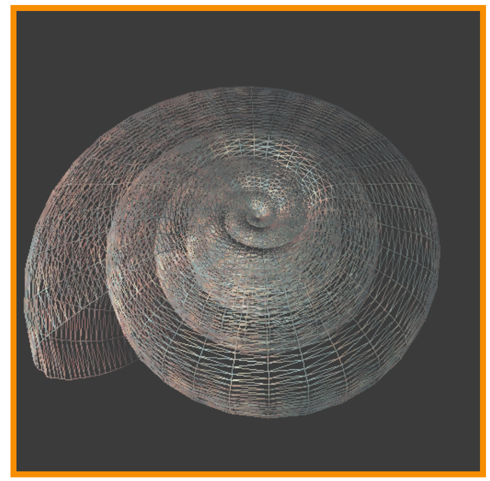
Image Composition
(Michael Bostock, CS426, Fall99)

Image Morphing
(Ianf, Wikipedia)

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)



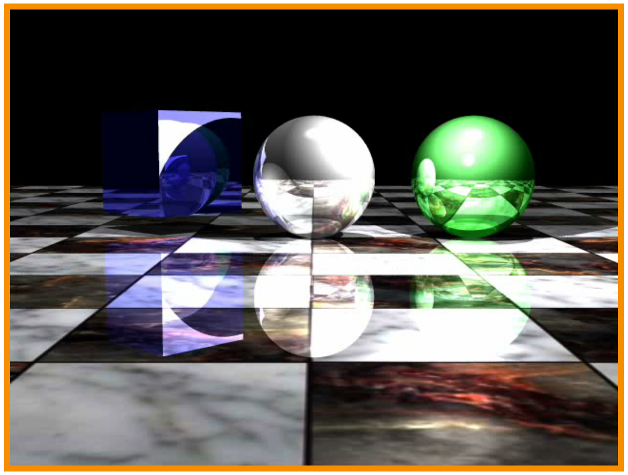
Shell

(Douglas Turnbull, CS 426, Fall99)

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

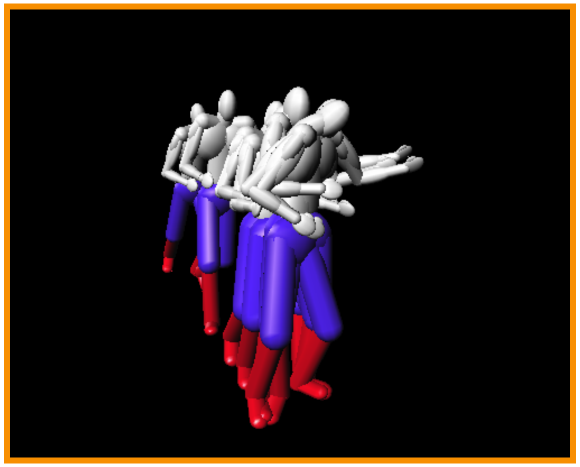
Pixel Shading
(Final Fantasy, Square Pictures)



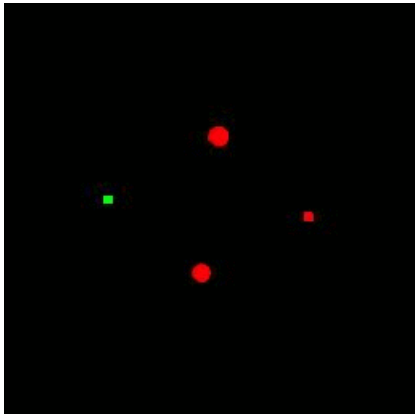
Ray Tracing
(Sid Kapur, CS 426, Spr04)

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
(Riley Thomasson, Spring 2014)

Applications

➔ **Entertainment**

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

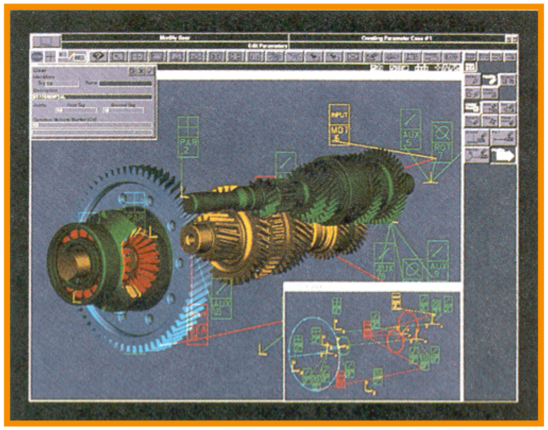
Geri's Game

(Pixar Animation Studios)

Applications

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

Los Angeles Airport
(Bill Jepson, UCLA)

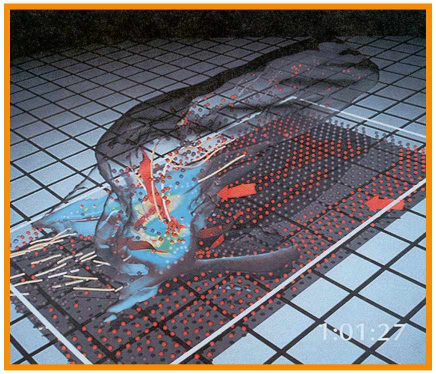


Gear Shaft Design
(Intergraph Corporation)

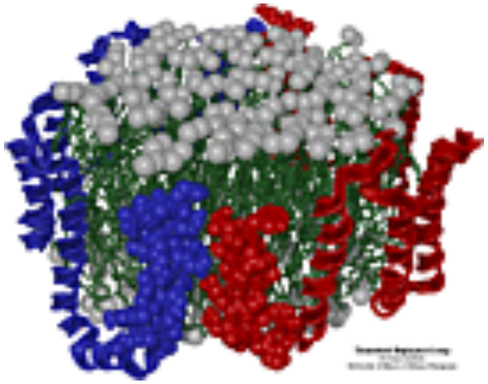
Boeing 777 Airplane
(Boeing Corporation)

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)



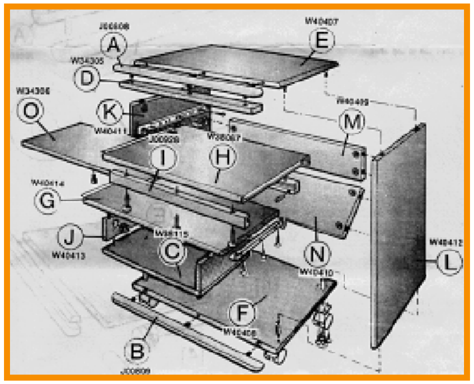
Apo A-1
(Theoretical Biophysics Group, University of Illinois at Urbana-Champaign)

Visible Human
(National Library of Medicine)

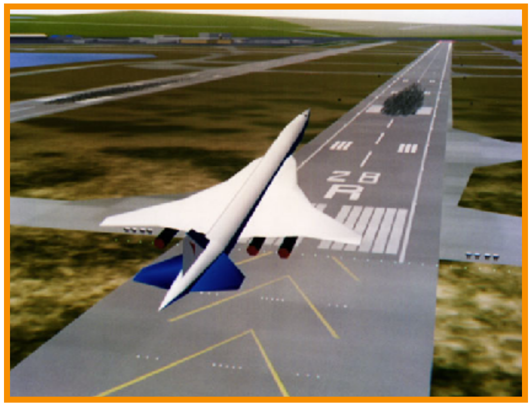
Applications

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

Driving Simulation
(Evans & Sutherland)



Desk Assembly
(Silicon Graphics, Inc.)



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

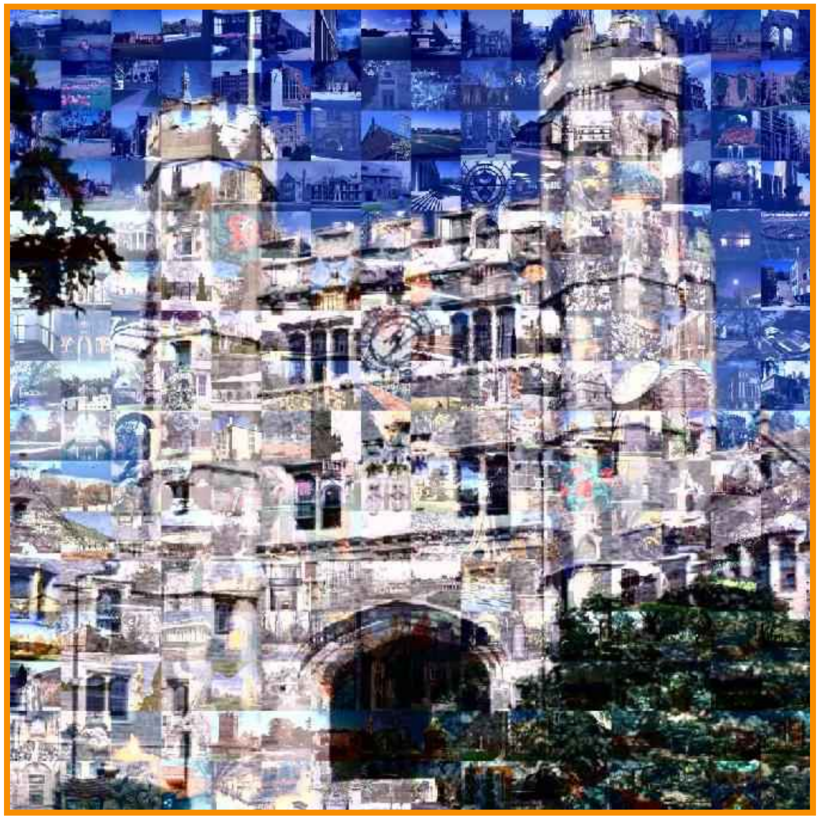
- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- → **E-commerce**
- Computer art

Interactive Kitchen Planner
(Matsushita)

Virtual Phone Store
(Lucent Technologies)

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 …

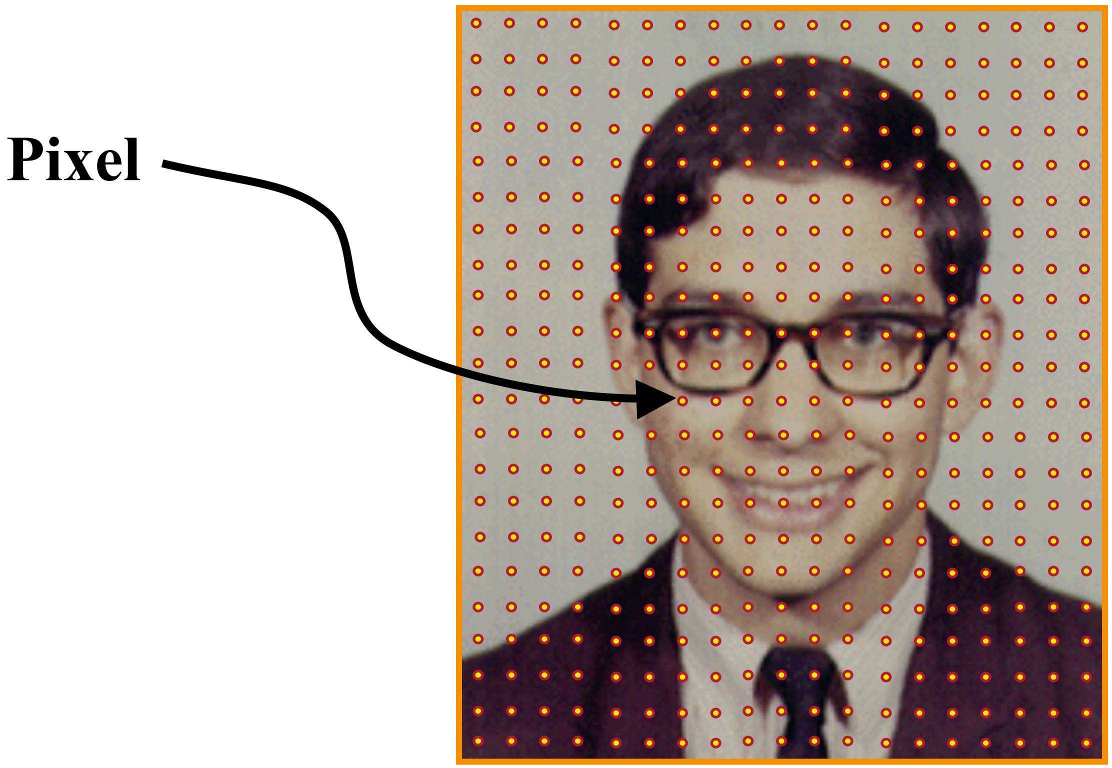
What is an Image?

What is an Image?

An image is a 2D rectilinear array of pixels

Digital image

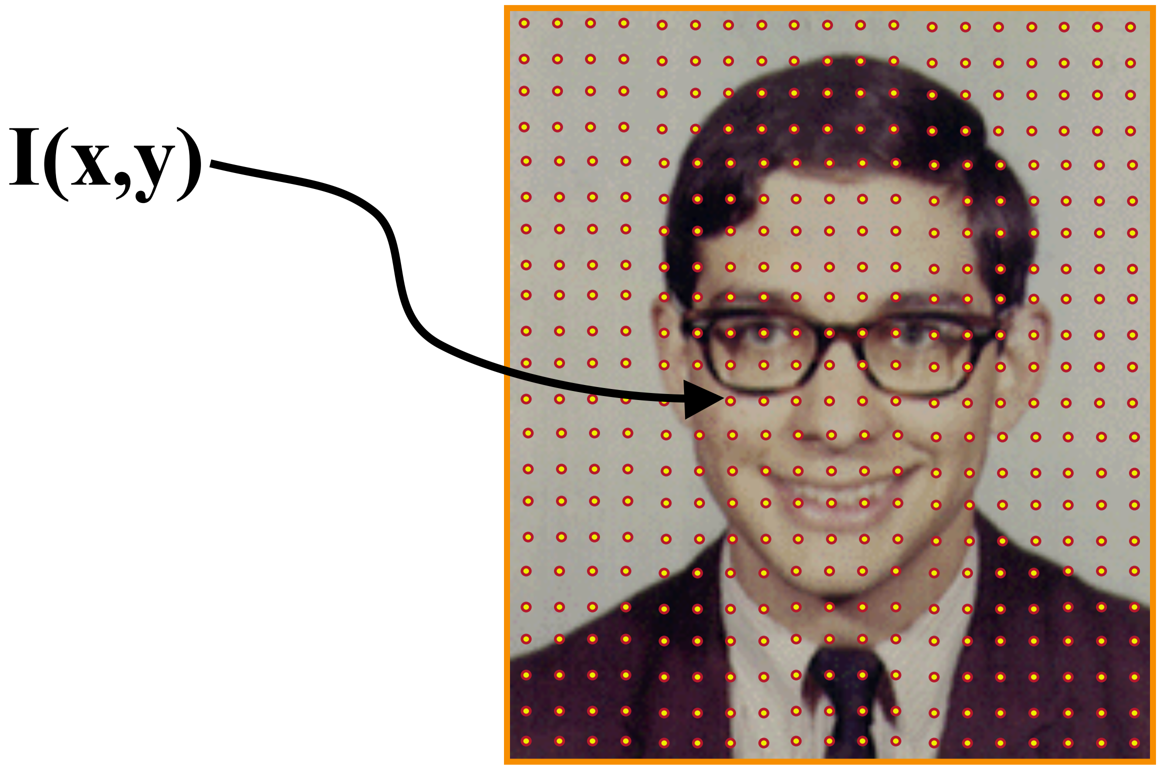
What is a Pixel?



Digital image

What is a Pixel?

Sample of a function at a position

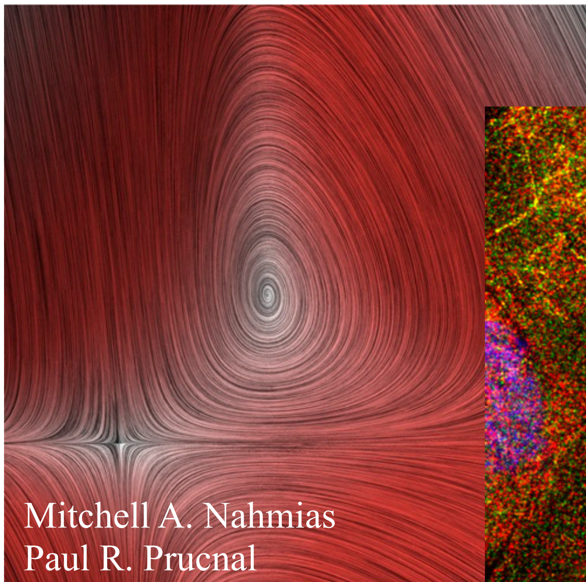


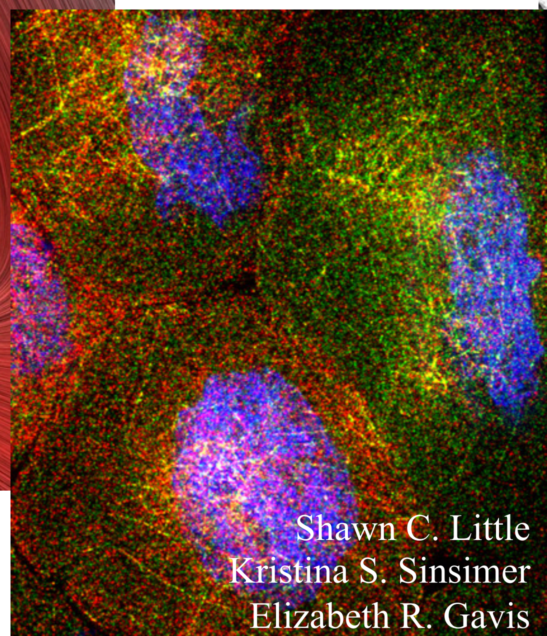
Digital image

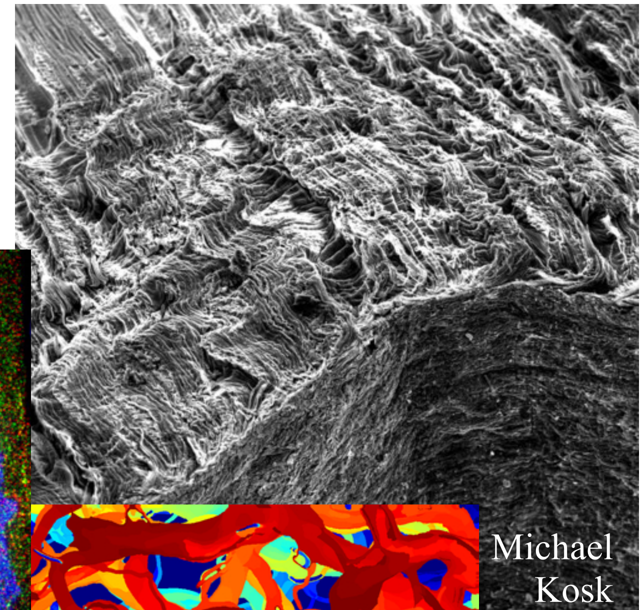
What Function?

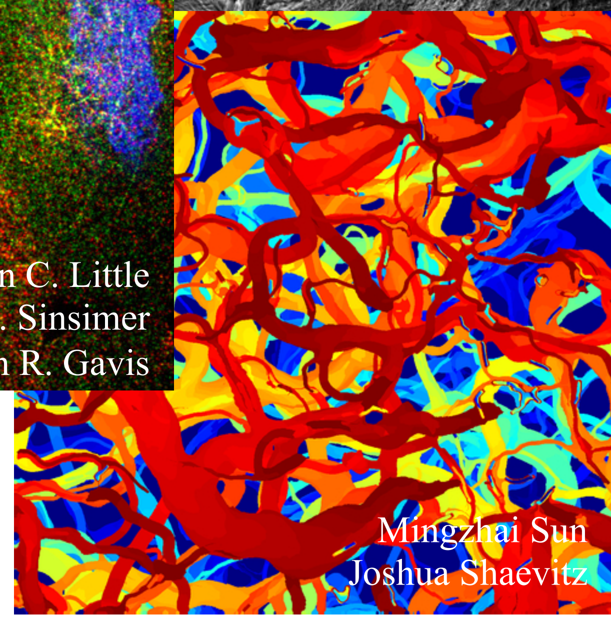
What Function?

Could be any function …







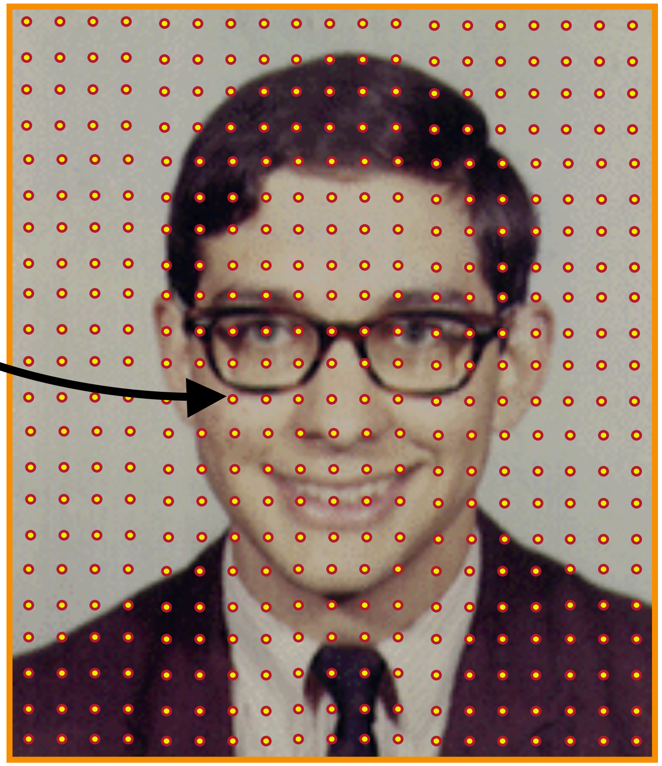


Art of Science
(Friend Center hallway)

What Function?

What about photographic images?

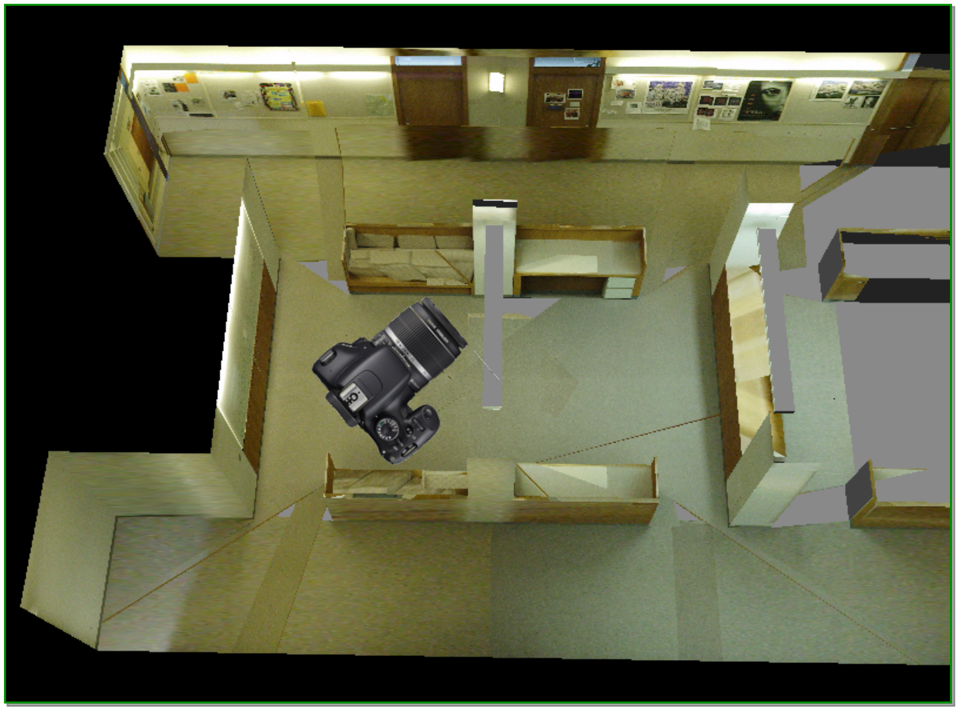
I(x,y)?



Digital photograph

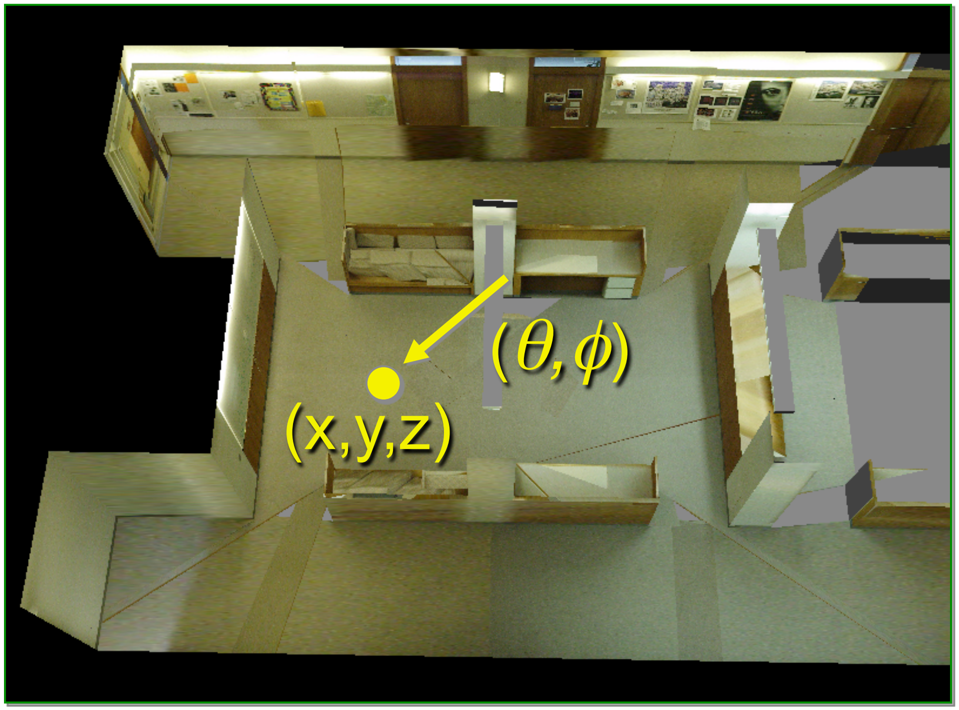
Plenoptic Function

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



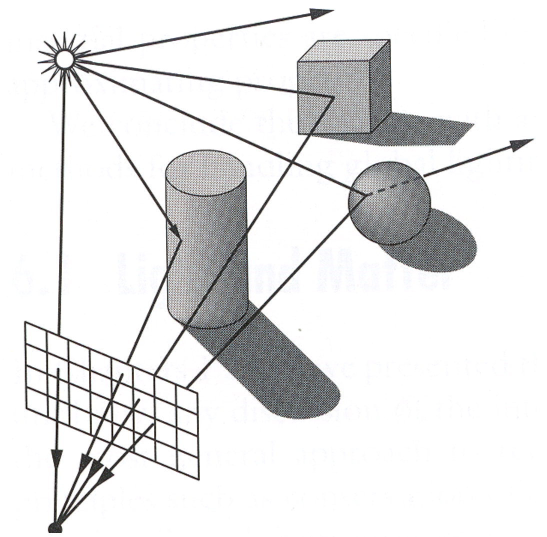
Plenoptic Function

- 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 (θ,ϕ),
  - at any time (t),
  - at any frequency (λ)



Photographic Images

- An idealized photographic image contains a 2D array of samples of the 7D plenoptic function
  - at a particular camera viewpoint,
  - for 2D array of directions,
  - at a certain time,
  - at certain frequencies

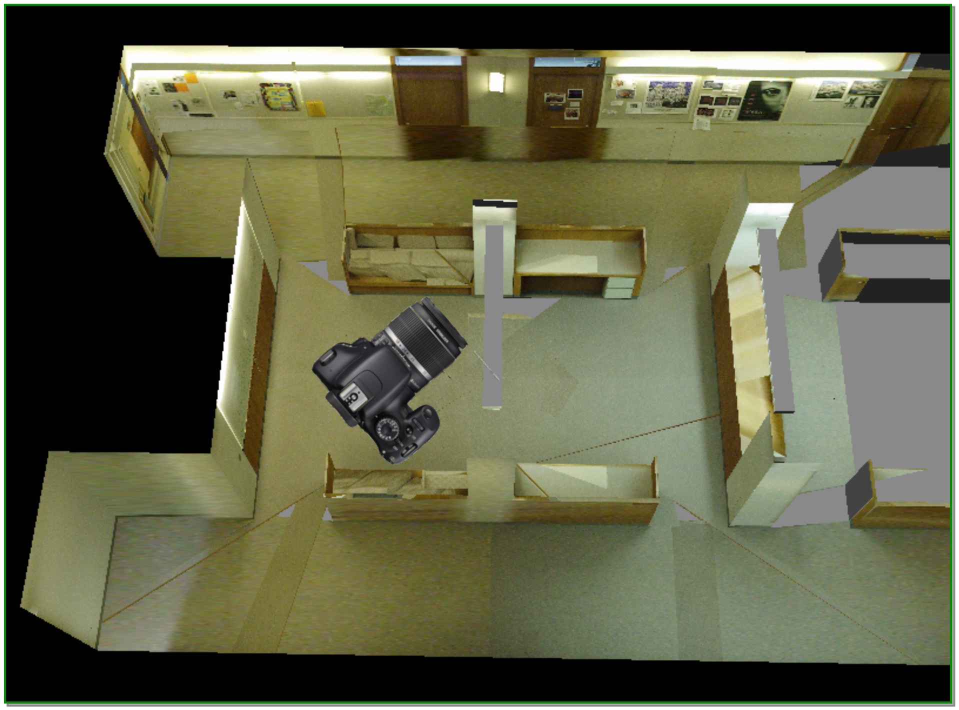


Camera viewpoint

Photographic Images

In practice, can't measure plenoptic function directly

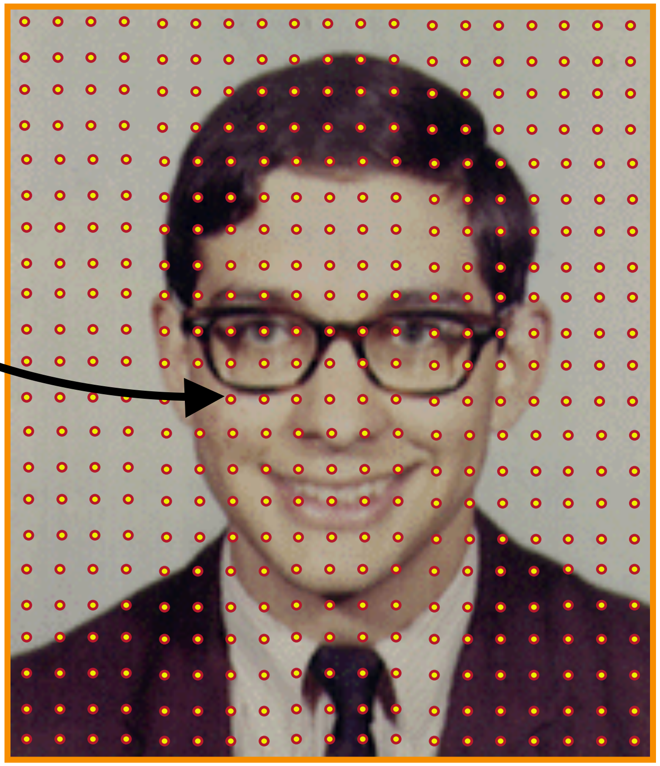
- Photoreceptors in eye
- Film in a traditional camera
- CCD cells in digital camera



Photographic Images

Photographic pixels as finite samples of the plenoptic function

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



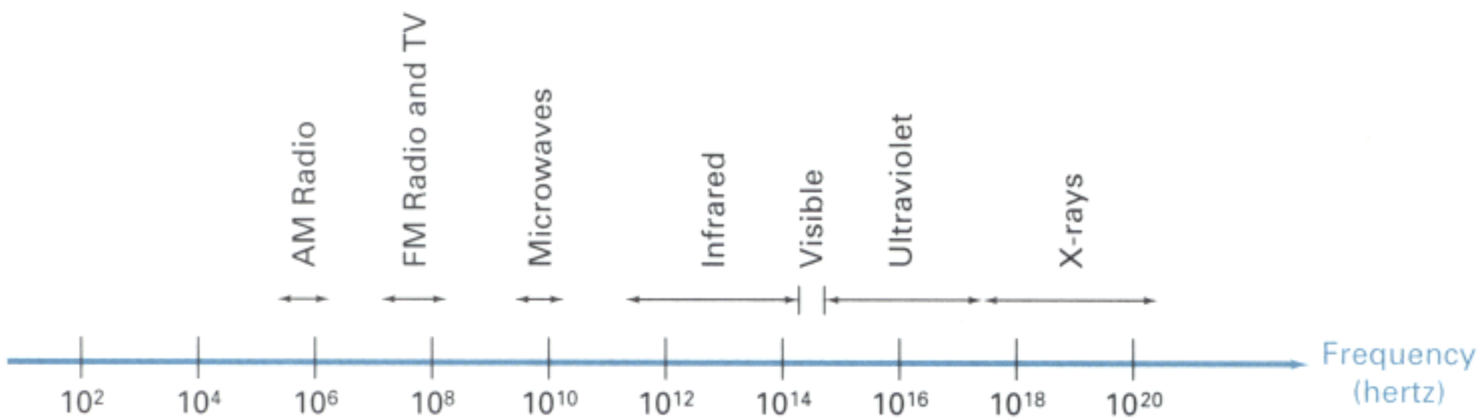
Digital photograph

What Frequencies?

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

Electromagnetic Spectrum

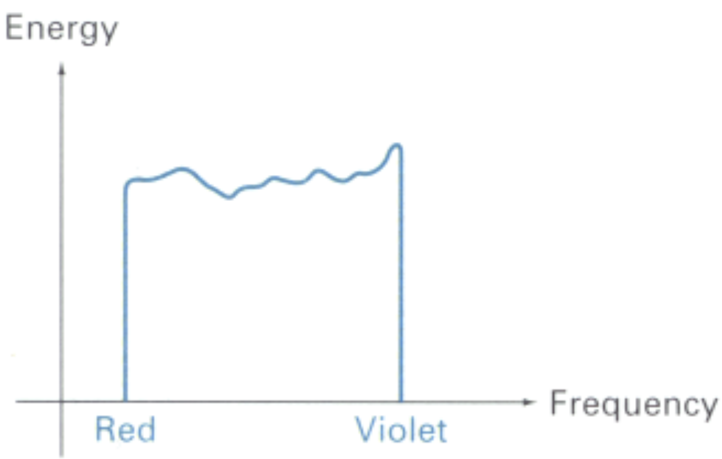
- Visible light frequencies range between ...
  - Red = 4.3×10^{14} hertz (700nm)
  - Violet = 7.5×10^{14} hertz (400nm)



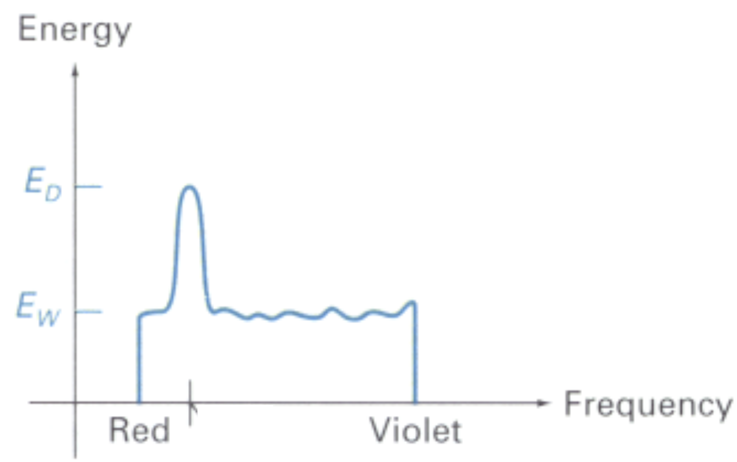
Figures 15.1 from H&B

Color

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



White Light



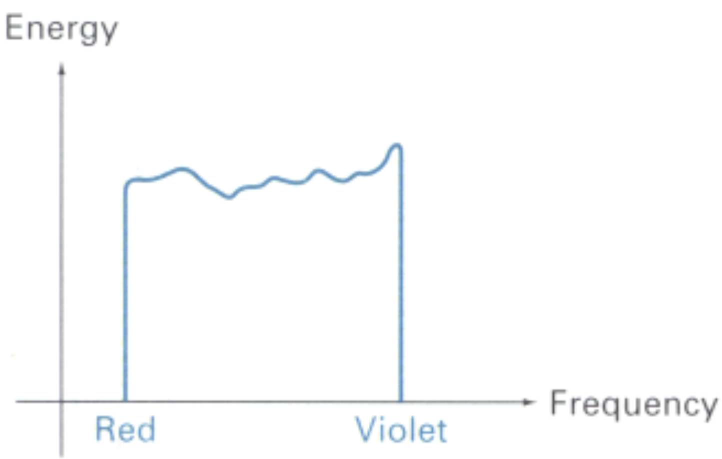
Orange Light

Figures 15.3-4 from H&B

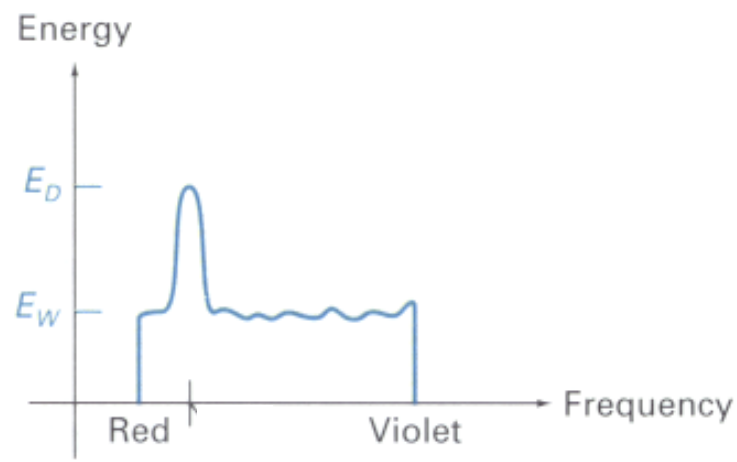
Color

How do we represent a color in a computer?

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



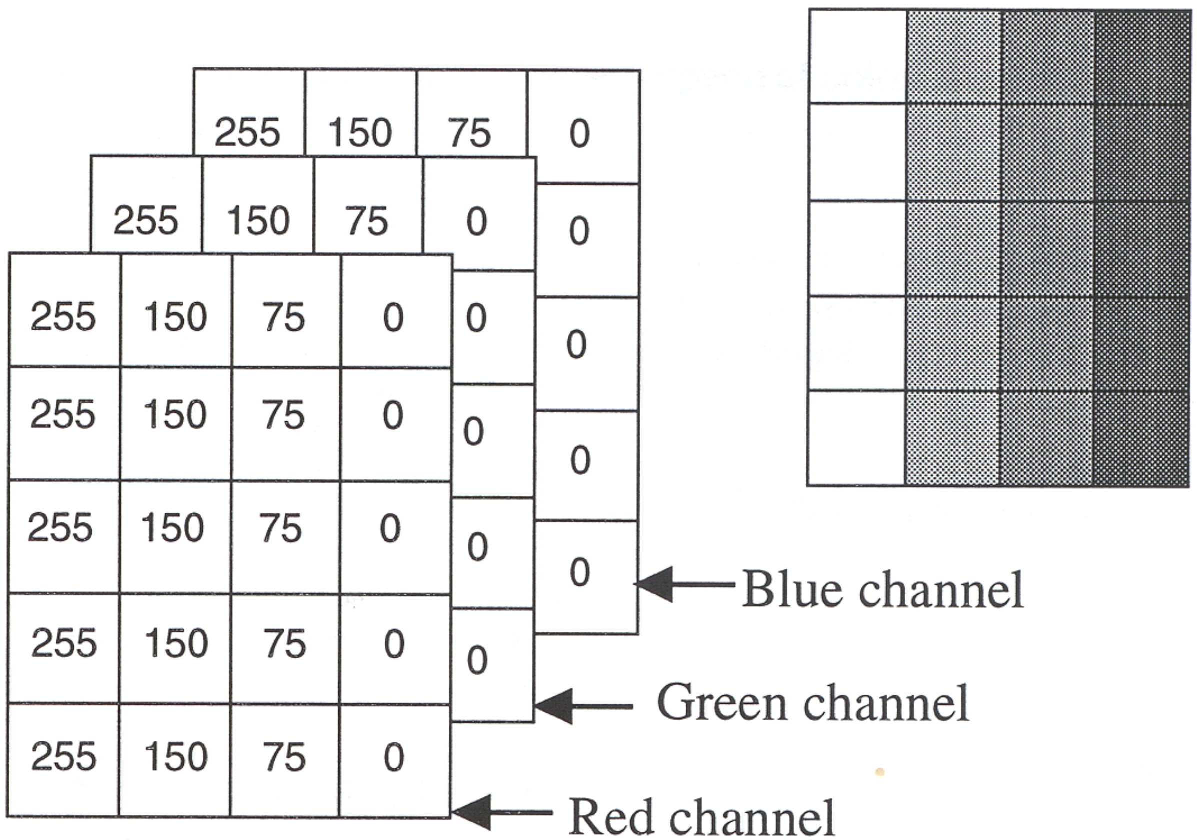
White Light



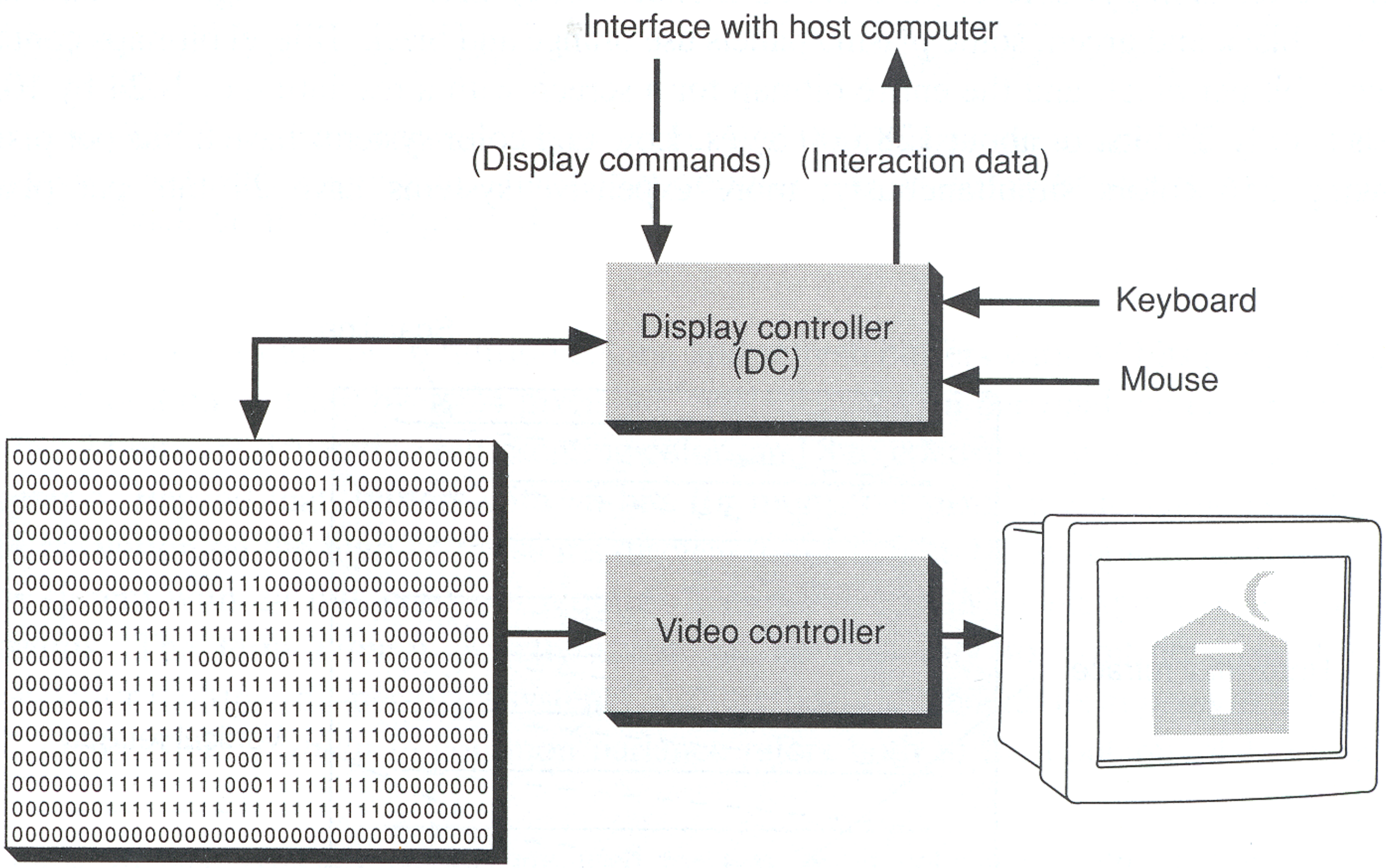
Orange Light

Figures 15.3-4 from H&B

Color Frame Buffer



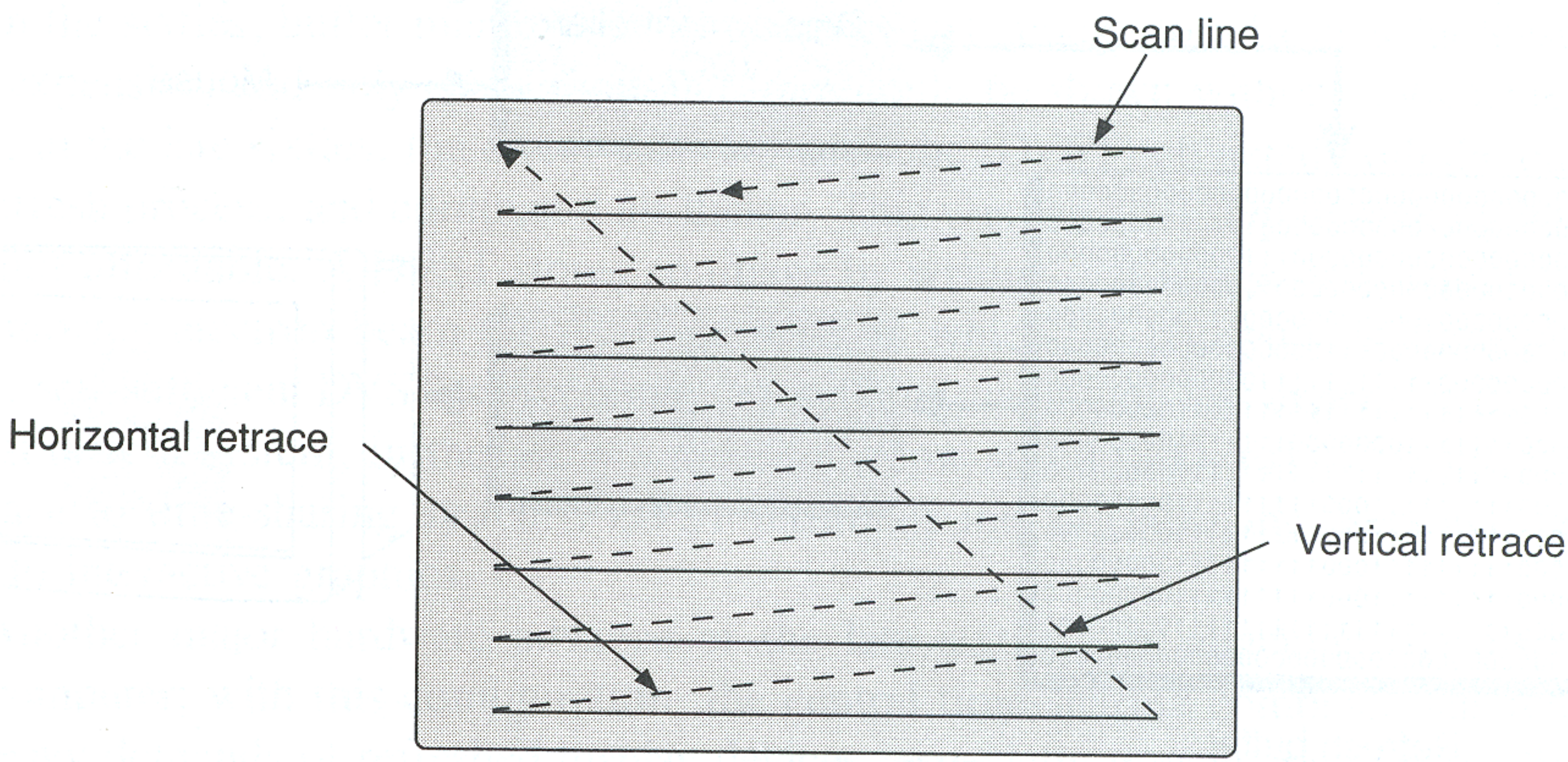
Frame Buffer Display



Frame Buffer

Figure 1.2 from FvDFH

Frame Buffer Display



Refresh rate is usually 60-75Hz

Figure 1.3 from FvDFH

Frame Buffer Display

- Video display devices
  - ➢ Liquid Crystal Display (LCD)
  - ➢ Cathode Ray Tube (CRT)
  - Plasma panels
  - 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)

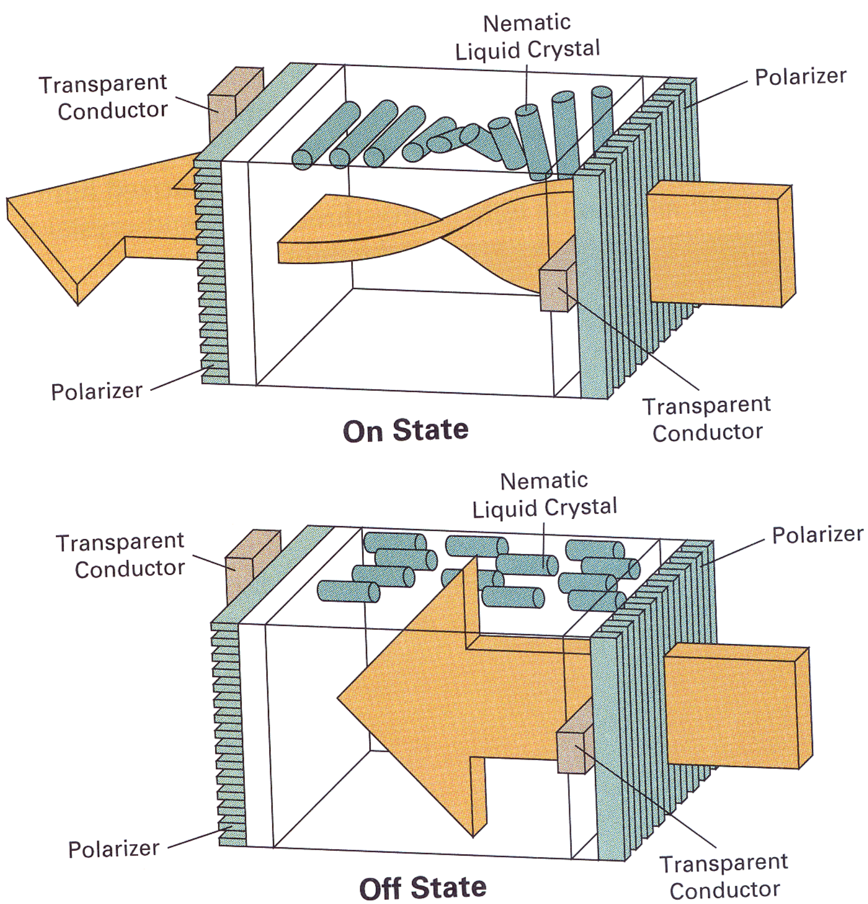
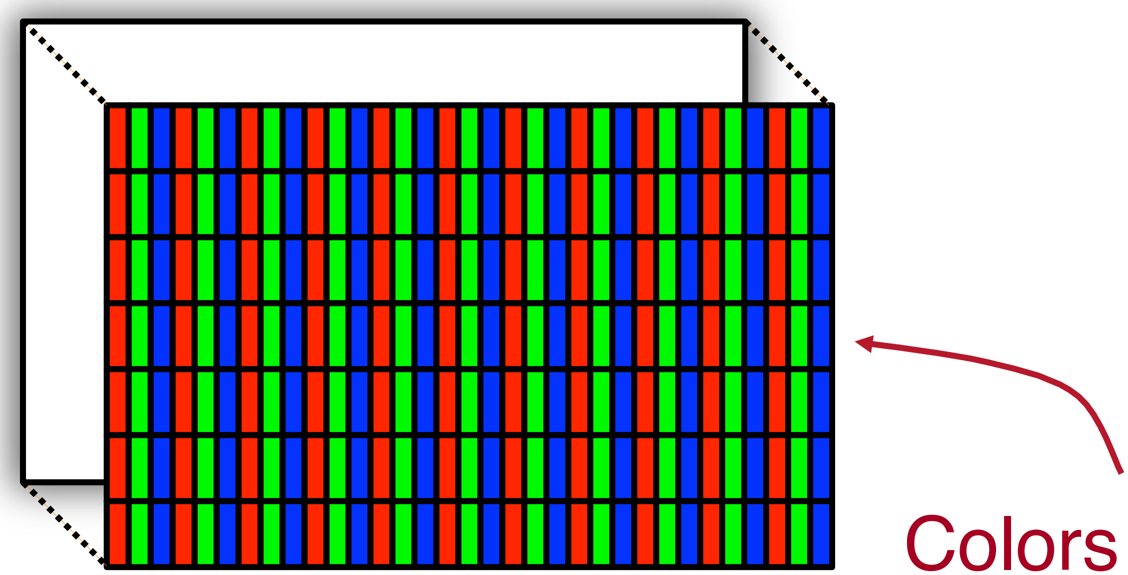


Figure 2.16 from H&B

Frame Buffer Display

Example: liquid crystal display (LCD)



Colors are interleaved

Frame Buffer Display

- Example: cathode ray tube (CRT)

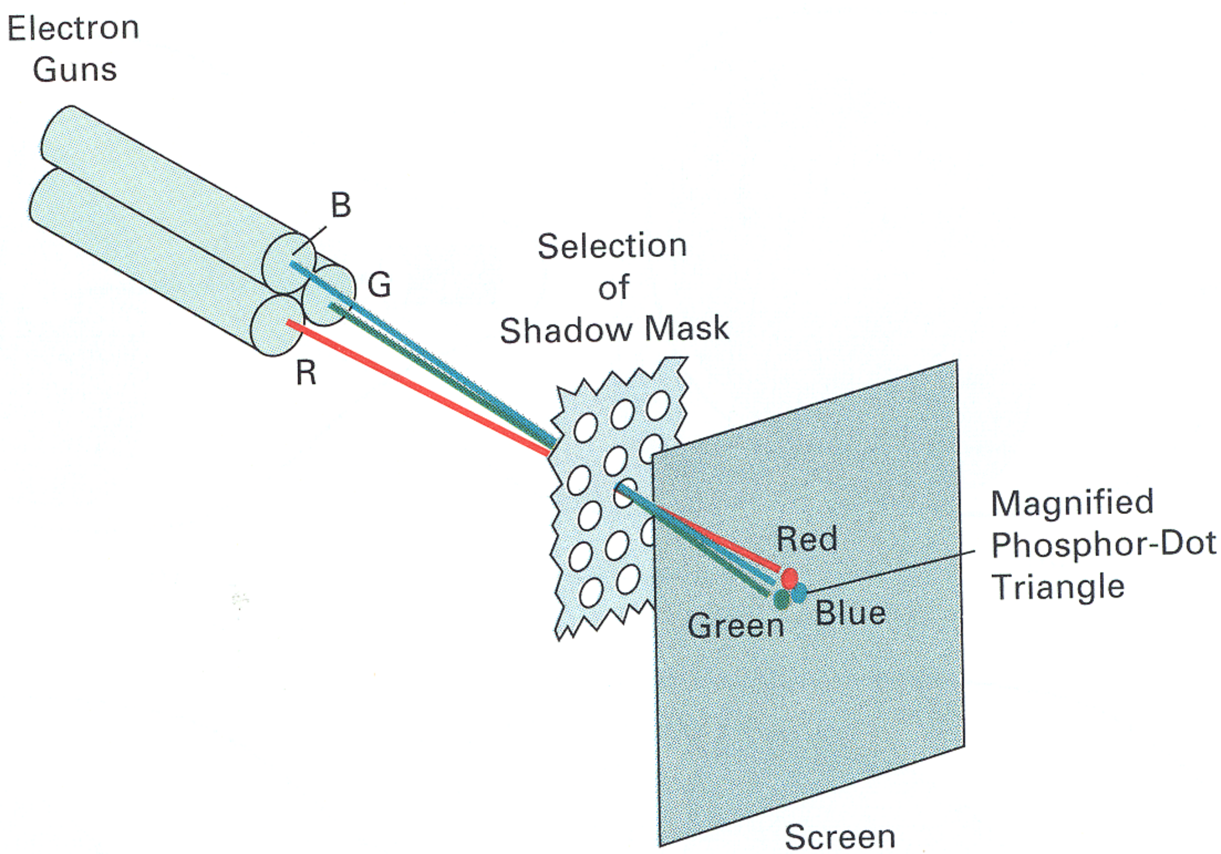
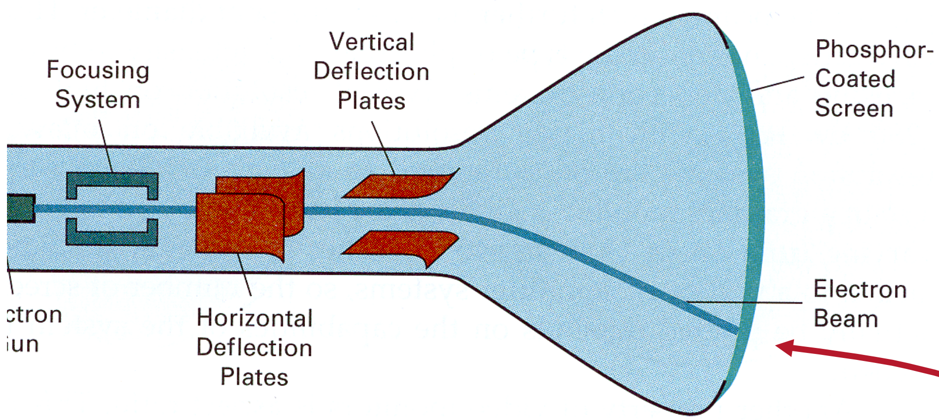


Figure 2.8 from H&B

Frame Buffer Display

Note: image is an array of samples – continuous function is “reconstructed” during display



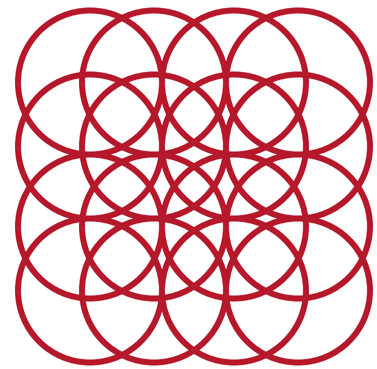
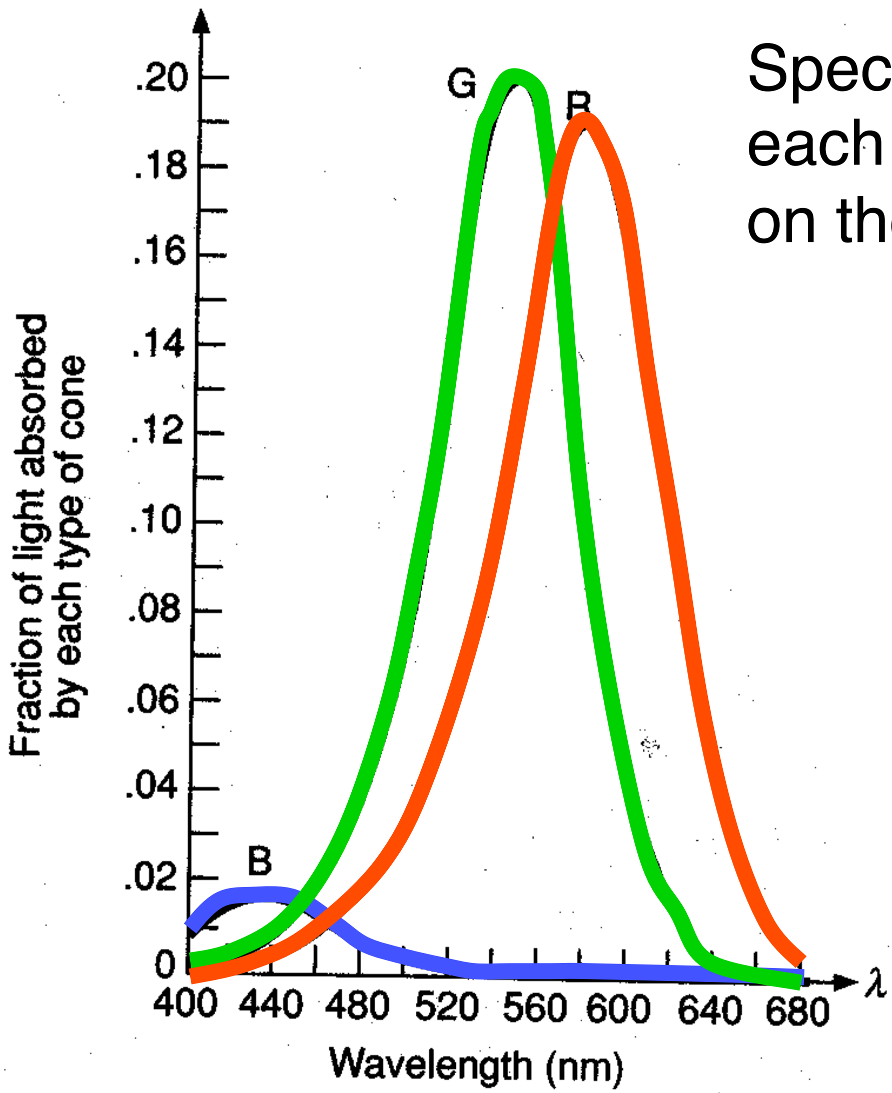


Image is reconstructed by displaying pixels with finite area (Gaussian)

Color

Why red, green, and blue (RGB)?

Human Color Perception

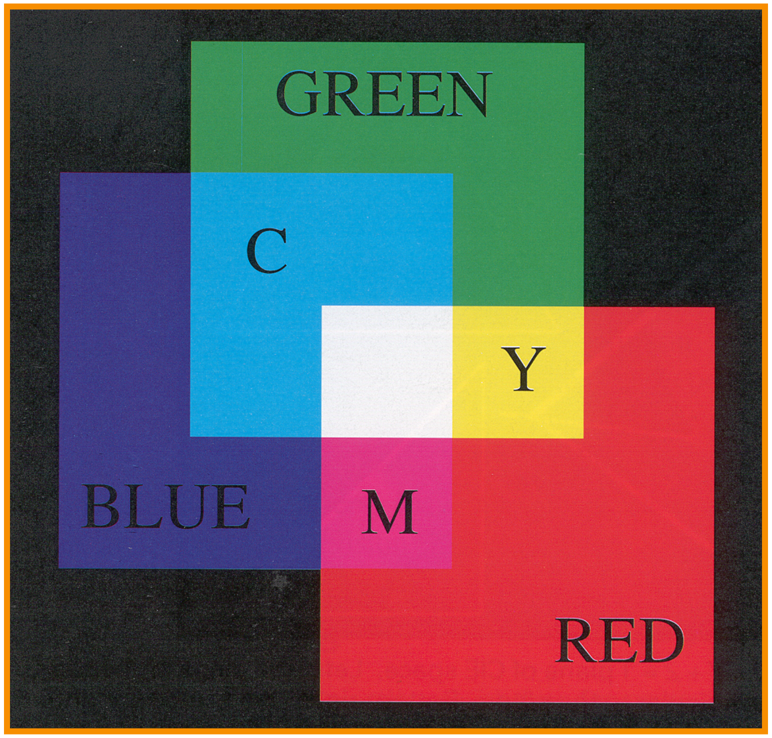


Spectral-response functions of each of the three types of cones on the human retina.

Tristimulus theory of color

Figure 13.18 from FvDFH

RGB Color Model

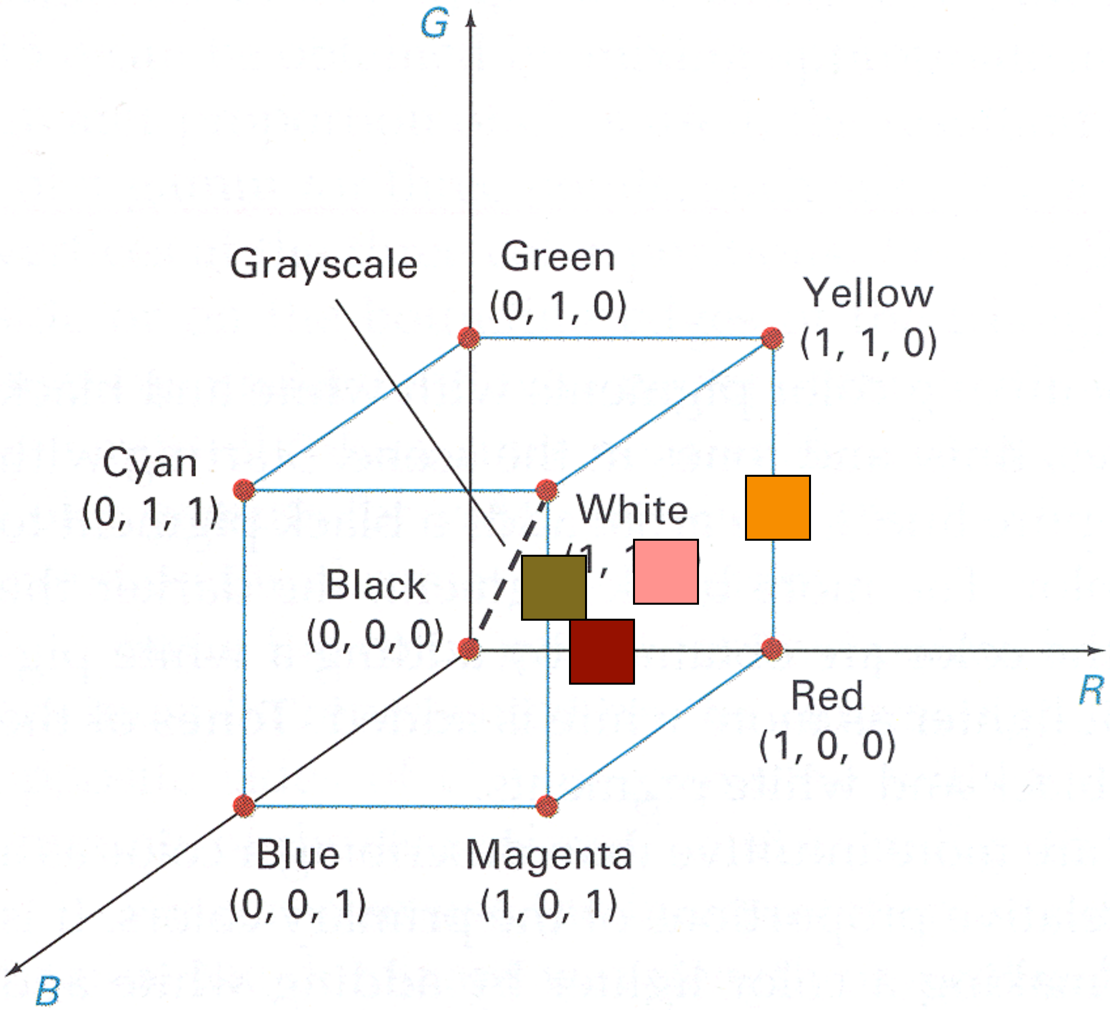


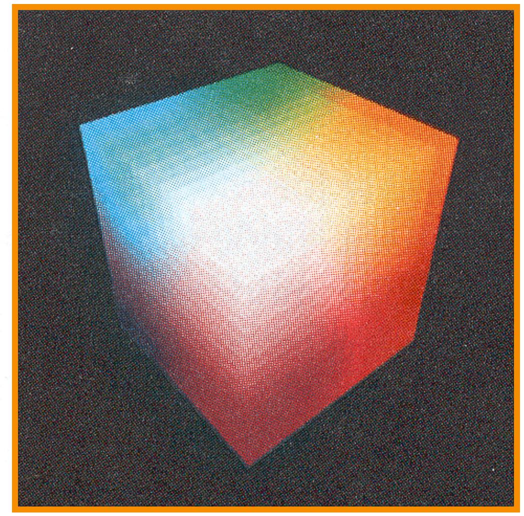
Colors are additive

R	G	B	Color
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	?

Plate II.3 from FvDFH

RGB Color Cube





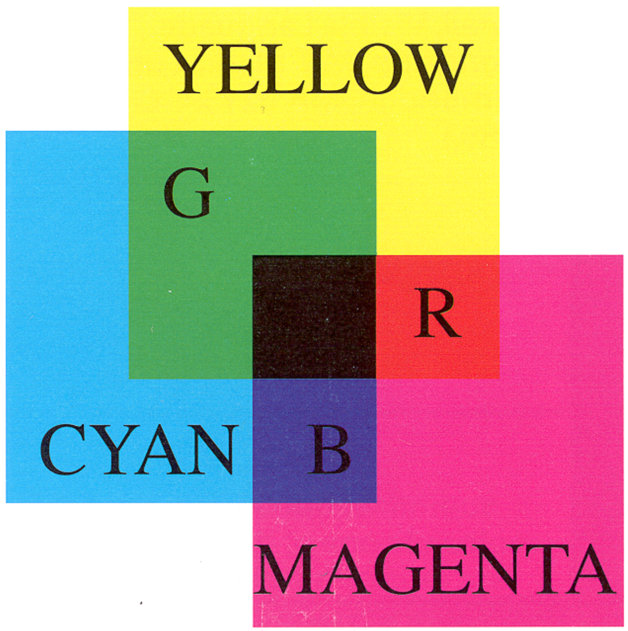
Figures 15.11&15.12 from H&B

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	

Plate II.7 from FvDFH

CMY Color Model

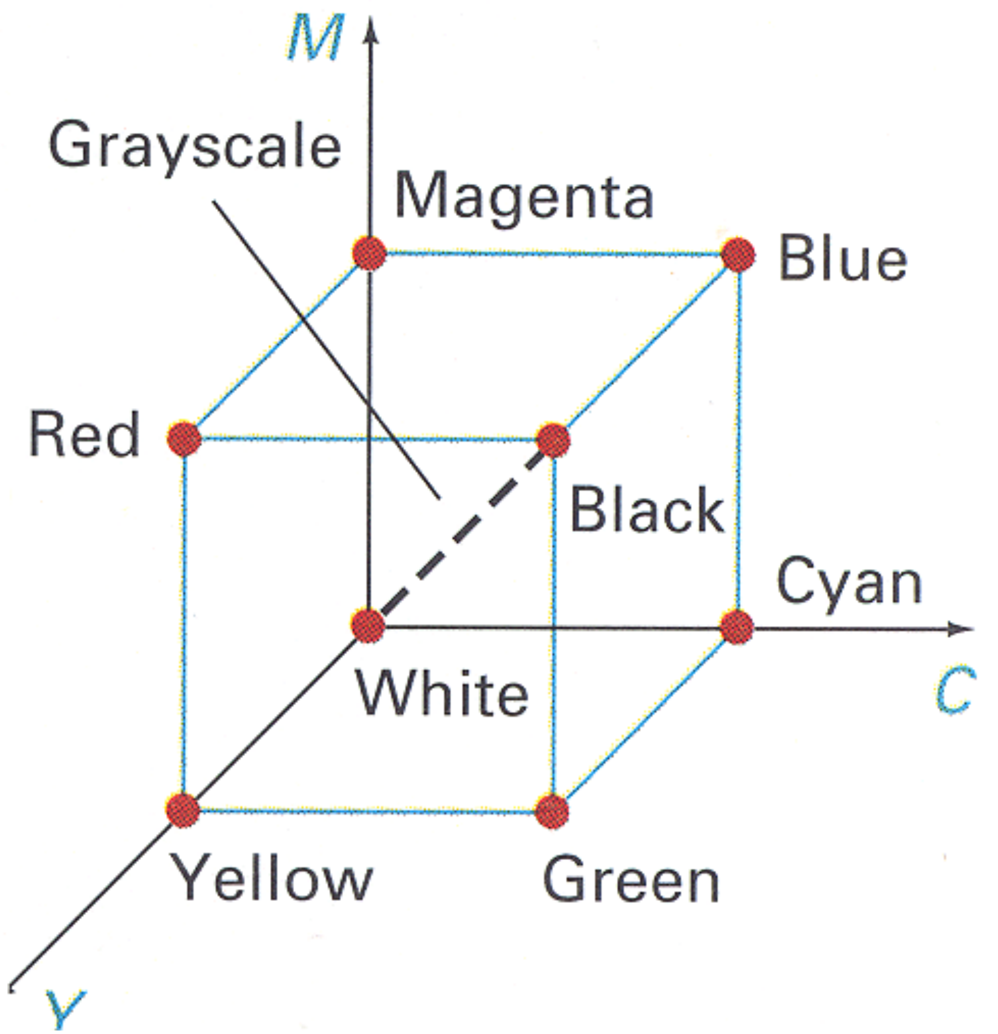
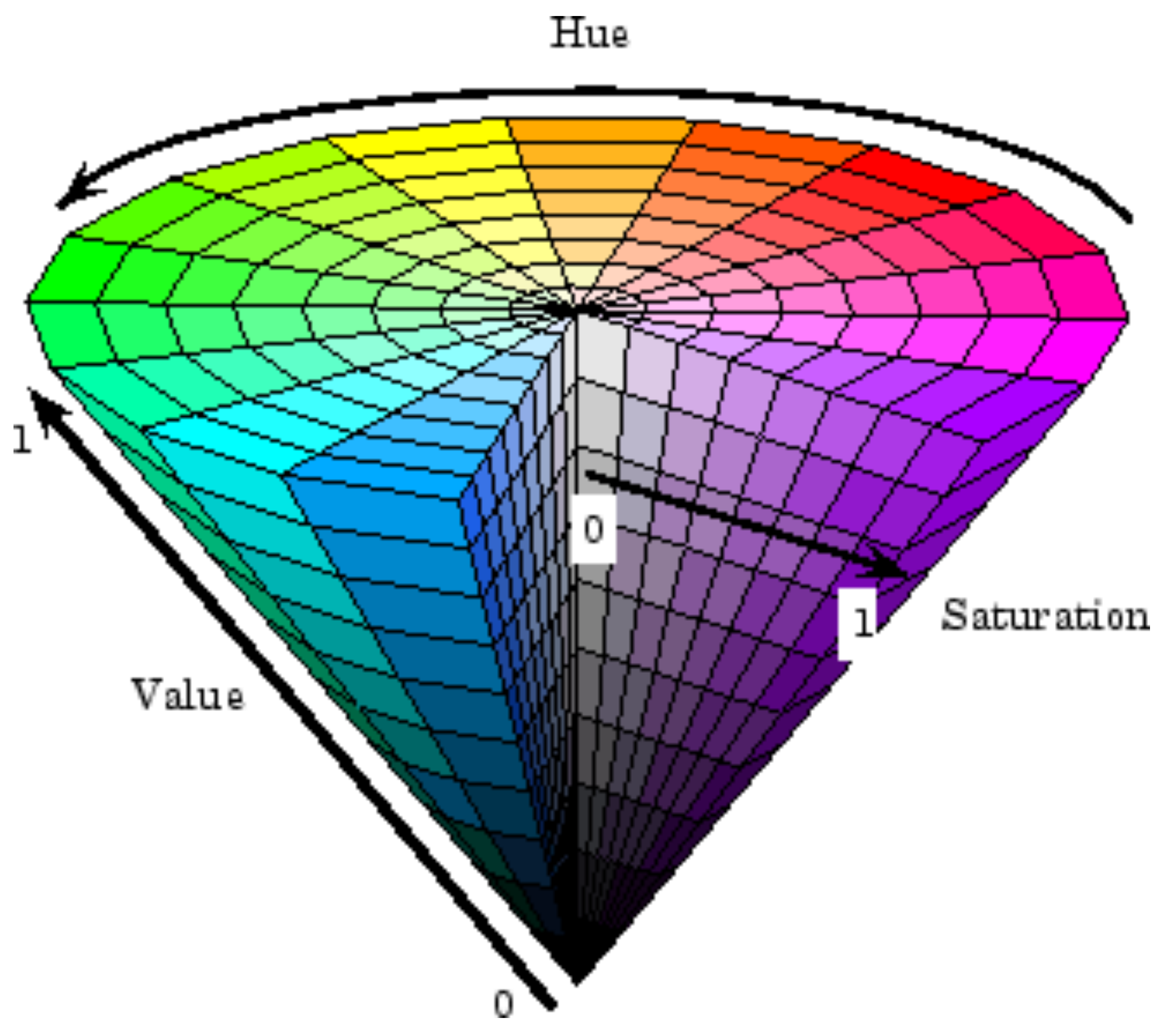
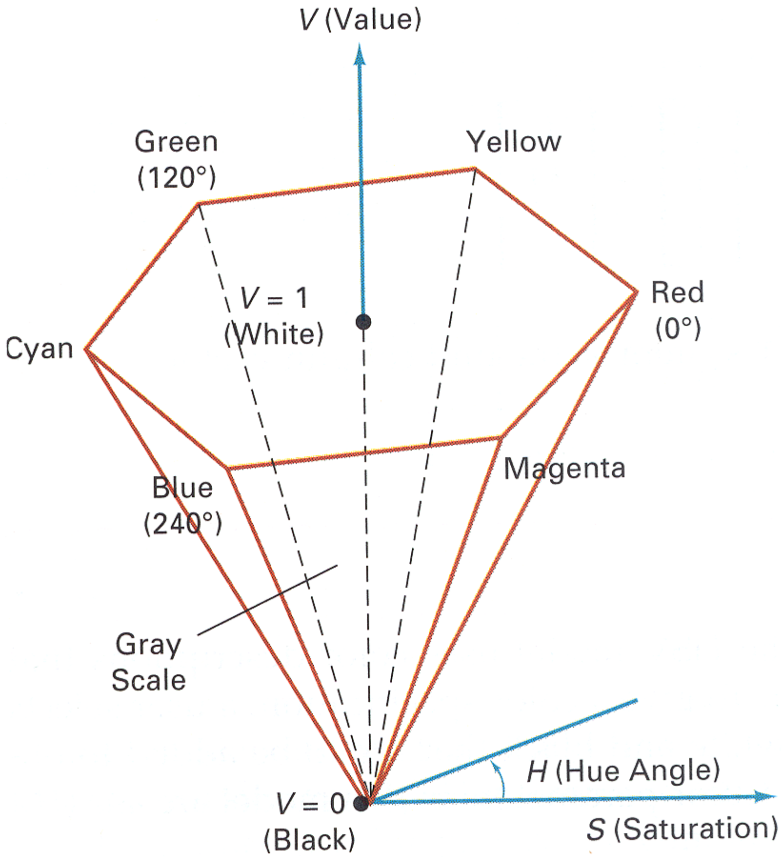


Figure 15.14 from H&B

HSV Color Model



HSV Color Model



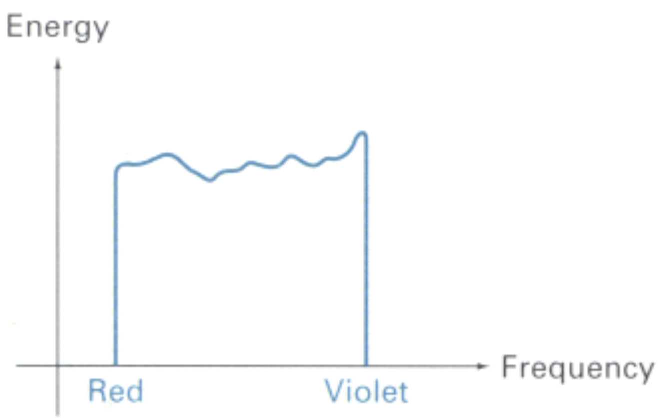
H	S	V	Color
0	1.0	1.0	Red
120	1.0	1.0	Green
240	1.0	1.0	Blue
*	0.0	1.0	White
*	0.0	0.5	Gray
*	*	0.0	Black
60	1.0	1.0	
270	0.5	1.0	
270	0.0	0.7	

Useful for user interfaces
because dimensions are intuitive

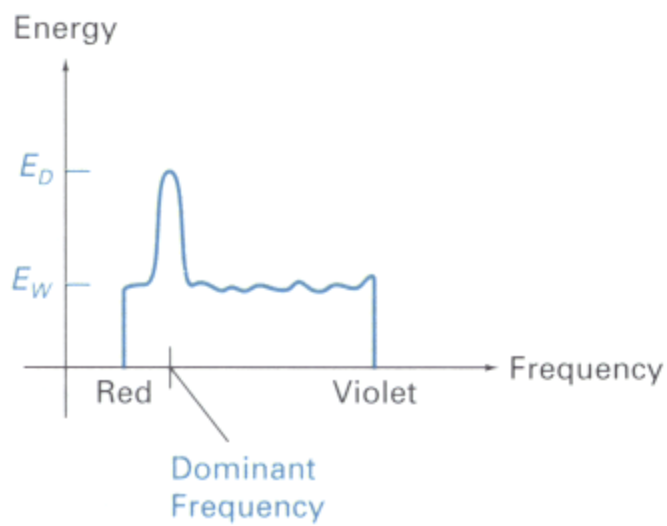
Figure 15.16&15.17 from H&B

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

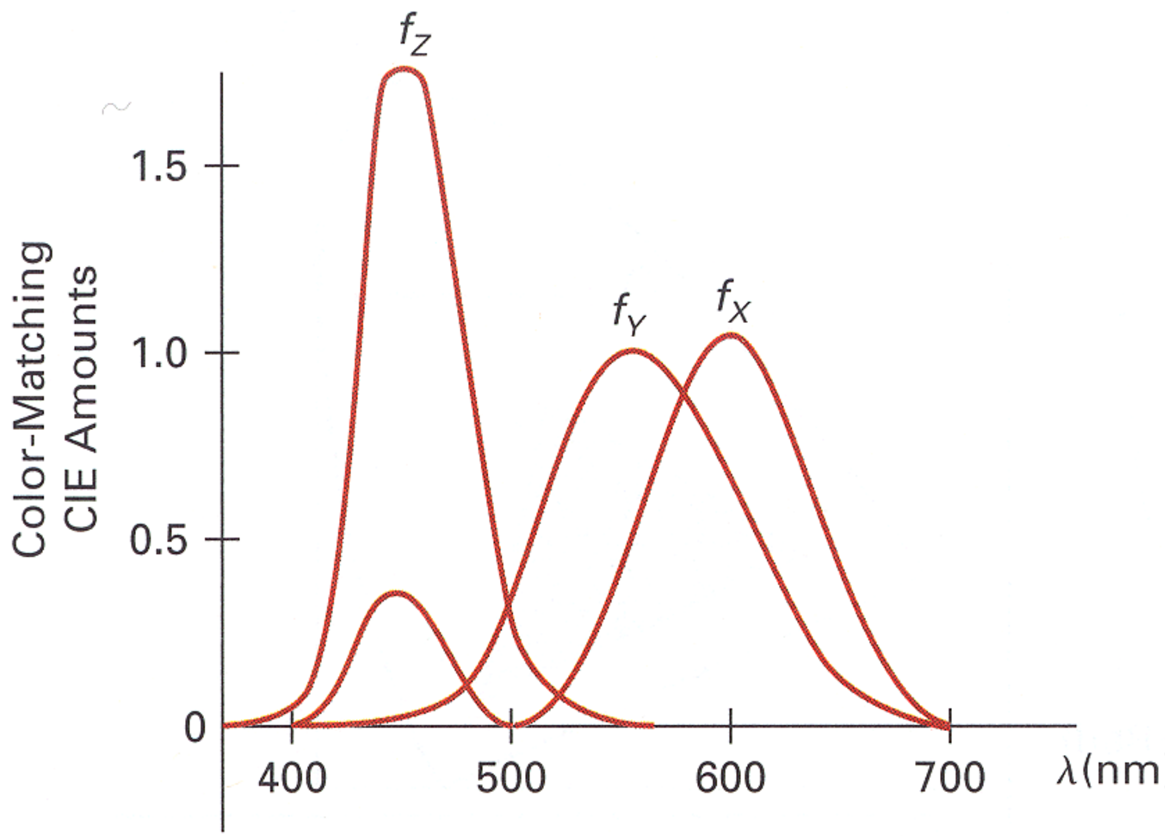


Orange Light

Figures 15.3-4 from H&B

XYZ Color Model (CIE)

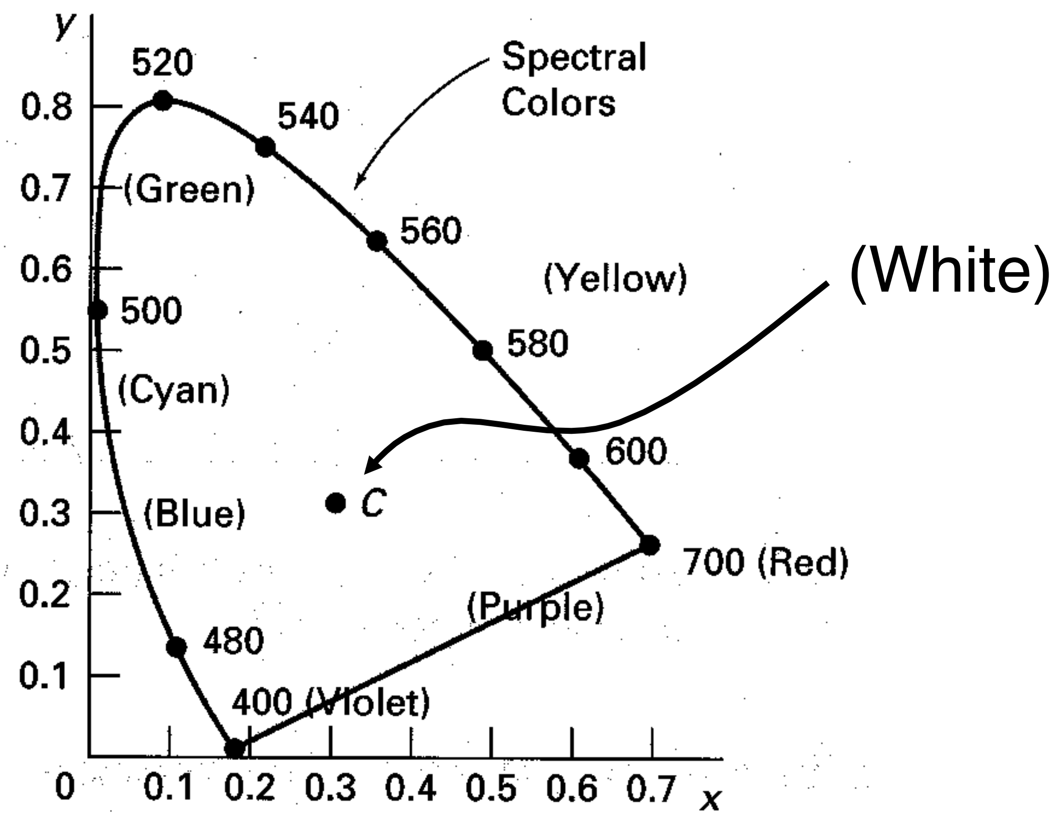
Figure 15.6 from H&B



Derived from perceptual experiments
All spectra that map to same XYZ give same visual sensation

XYZ Color Model (CIE)

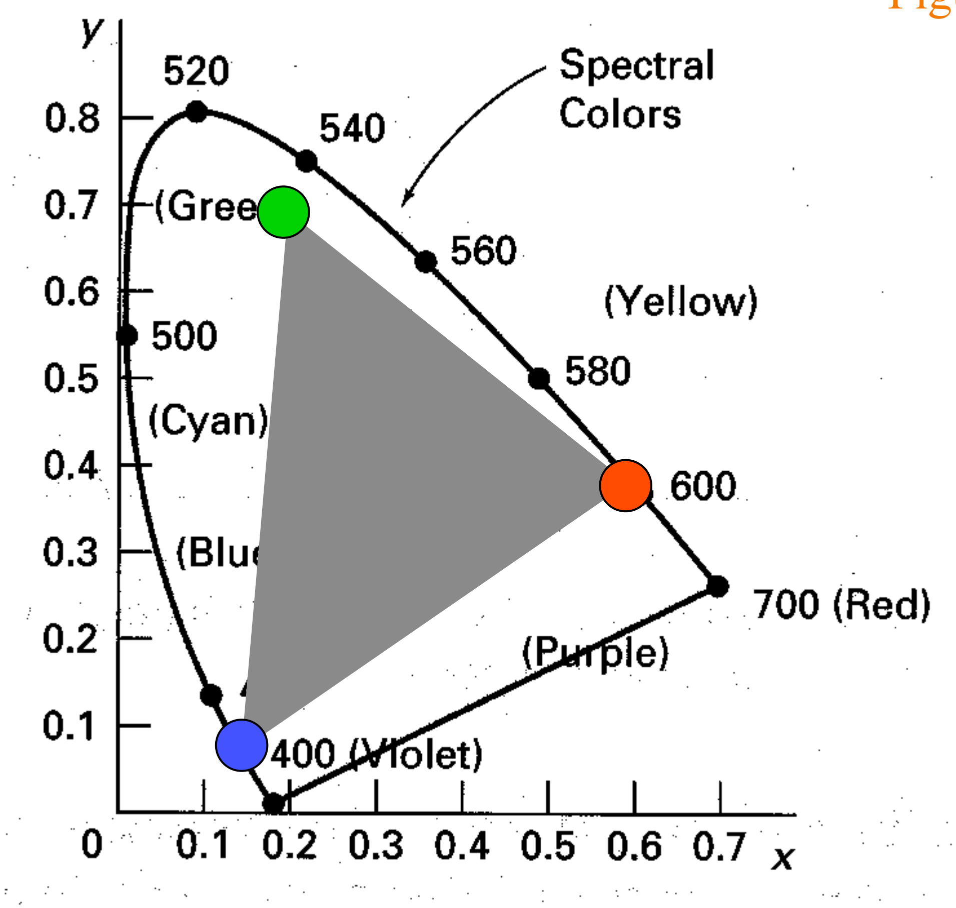
Figure 15.7 from H&B



Normalized amounts of X and Y for colors in visible spectrum

XYZ Color Model (CIE)

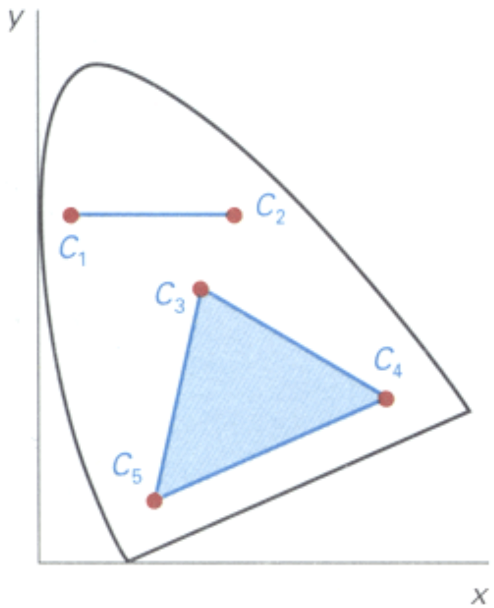
Figure 15.13 from H&B



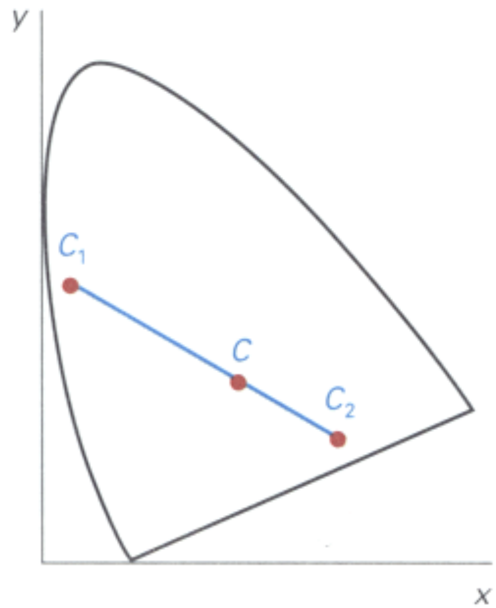
Useful for reasoning about coverage of color gamuts

XYZ Color Model (CIE)

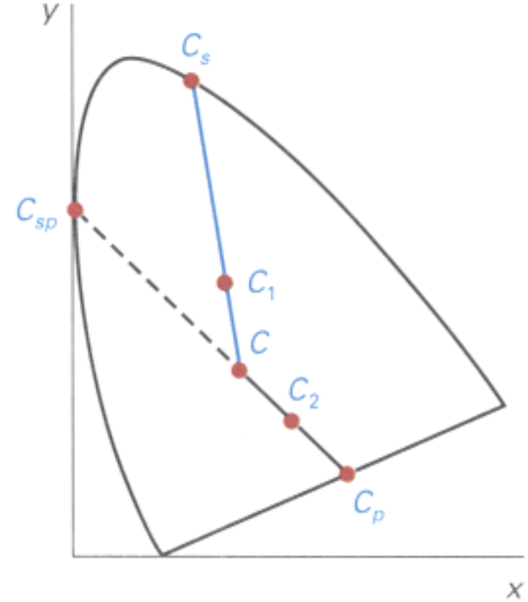
Figures 15.8-10 from H&B



Compare Color Gamuts



Identify Complementary Colors

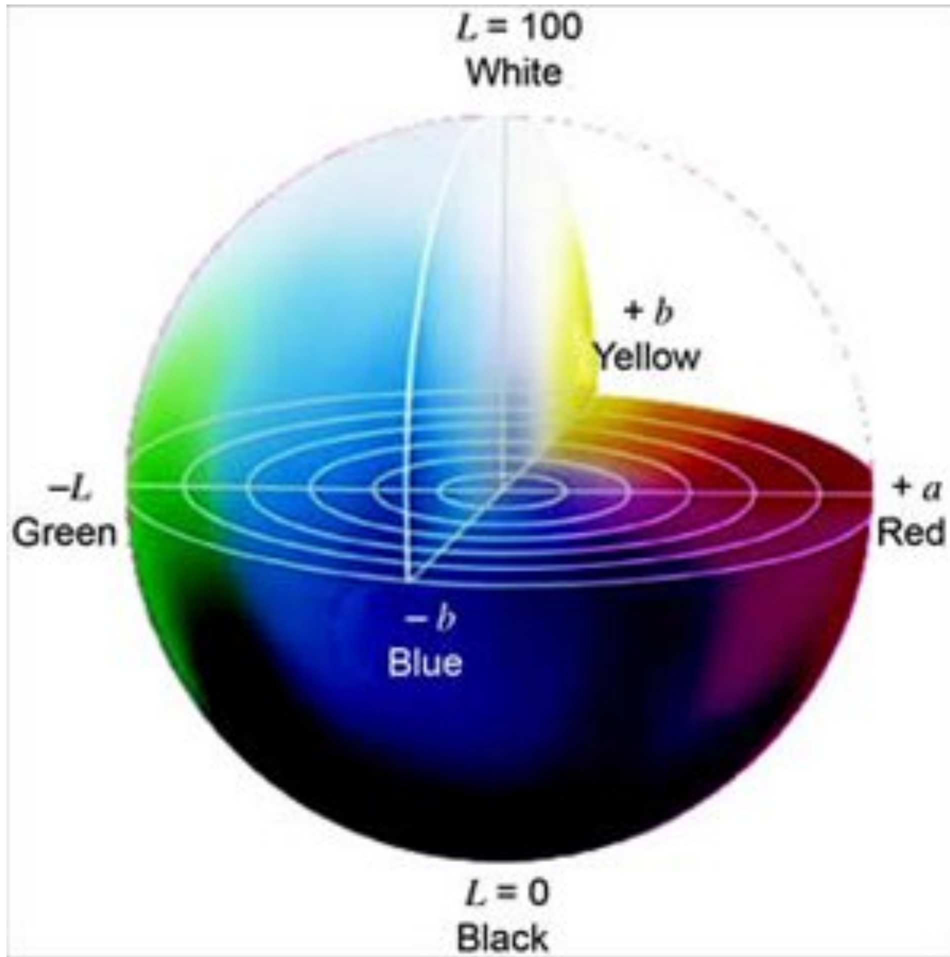


Determine Dominant Wavelength and Purity

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!