Computer Vision 1

THEO GEVERS, SHAODI YOU, THOMAS MENSINK AND PASCAL METTES

MASTER AI

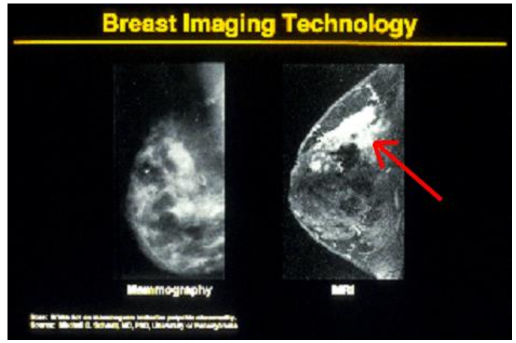
UNIVERSITY OF AMSTERDAM

Disclaimer

- Many of the slides used here are obtained from online resources (including many open lecture materials) without appropriate acknowledgement. They are used here for the sole purpose of classroom teaching. All the credit and all the copyrights belong to the original authors. You should not copy it, redistribute it, put it online, or use it for any other purposes than for this course.

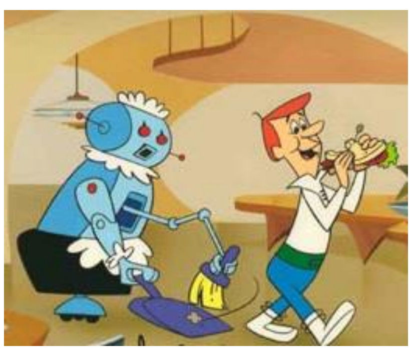
Last Week

Safety



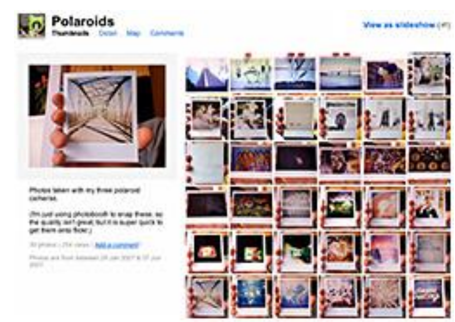
Health

Security



Comfort

Fun



Access

Last Week

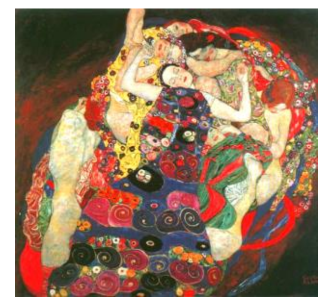
- Viewpoint variation

- Illumination change

- Orientation and scale

- Occlusion

- Clutter

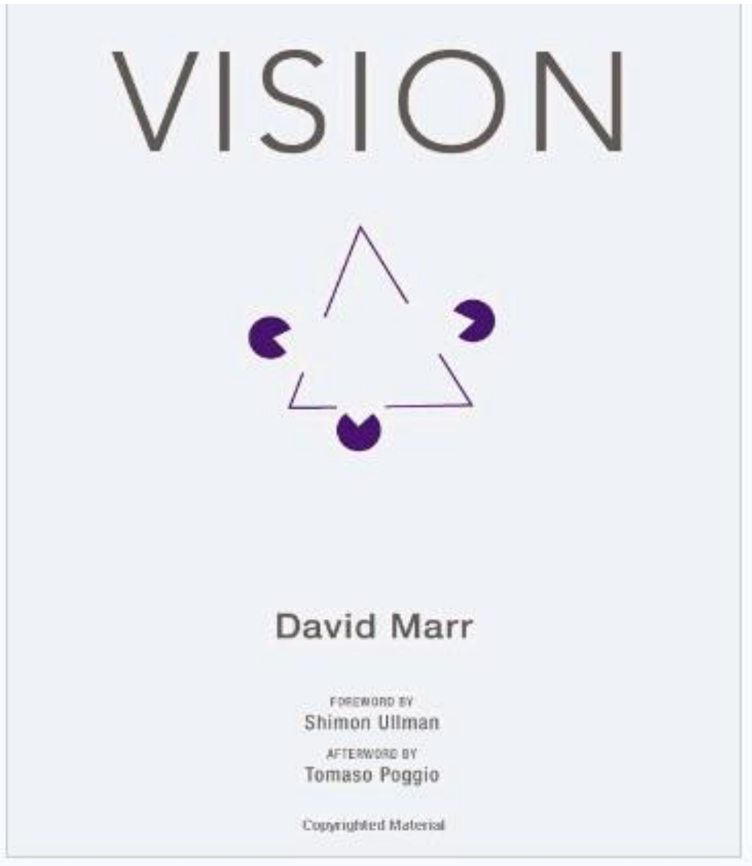


- Appearance change

What is Vision?

- Is [the art] *``to know what is where, by looking.''*

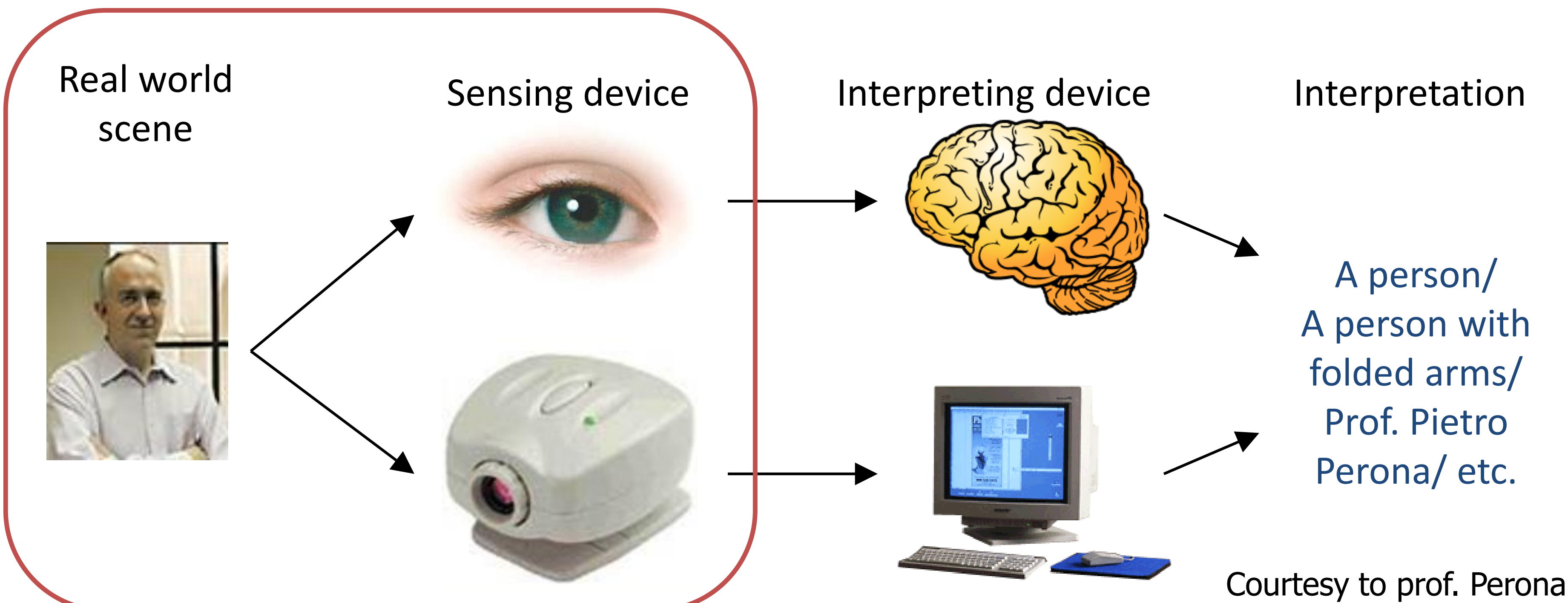
David Marr (1945 --1980).



The computer vision problem

- Make a computer to see and to understand images

- We know it is physically possible – we do it every day and effortlessly!



Lectures/Theory

- 02-09-2019, 17:00-19:00, H0.08, **Introduction** (Szeliski 1) – **Theo Gevers**
- 09-09-2019, 17:00-19:00, H0.08, **Image Formation** (Szeliski: 2.1.1 + 2.1.2 + 2.1.5 +
- 2.2 + 2.3.2 + 2.3.3) – **Theo Gevers/Shaodi You**
- 16-09-2019, 17:00-19:00, H0.08, **Image Processing** (Szeliski: 3.1 + 3.2 + 3.3 + 4 +
- 8.4) – **Shaodi You**
- 23-09-2019, 17:00-19:00, H0.08, **Object Recognition** (Szeliski: 14; Bengio 5.7) – **Shaodi You**
- 30-09-2019, 17:00-19:00, H0.08, **Deep Learning** (Szeliski: 3.2; Bengio: 6 + 7.4 + 7.7 +
- 7.9 + 9) – **Thomas Mensink**
- 07-10-2019, 17:00-19:00, H0.08, **Deep Video** (Bengio: 10.1 + 10.2 (intro, .2, .3) + 10.3 + 10.4 + 10.5 + 10.7 + 10.10) – **Pascal Mettes**
- 14-10-2019, 17:00-19:00, H0.08, **Applications** (Szeliski: 12.6.2 + 12.6.3 + 12.2.4; Bengio: 14.1+ 14.2+14.3+14.6+15.1+ 15.2)
- 25-10-2019, 13:00-16:00, **Written Exam**

Human eye Geometry and Optics

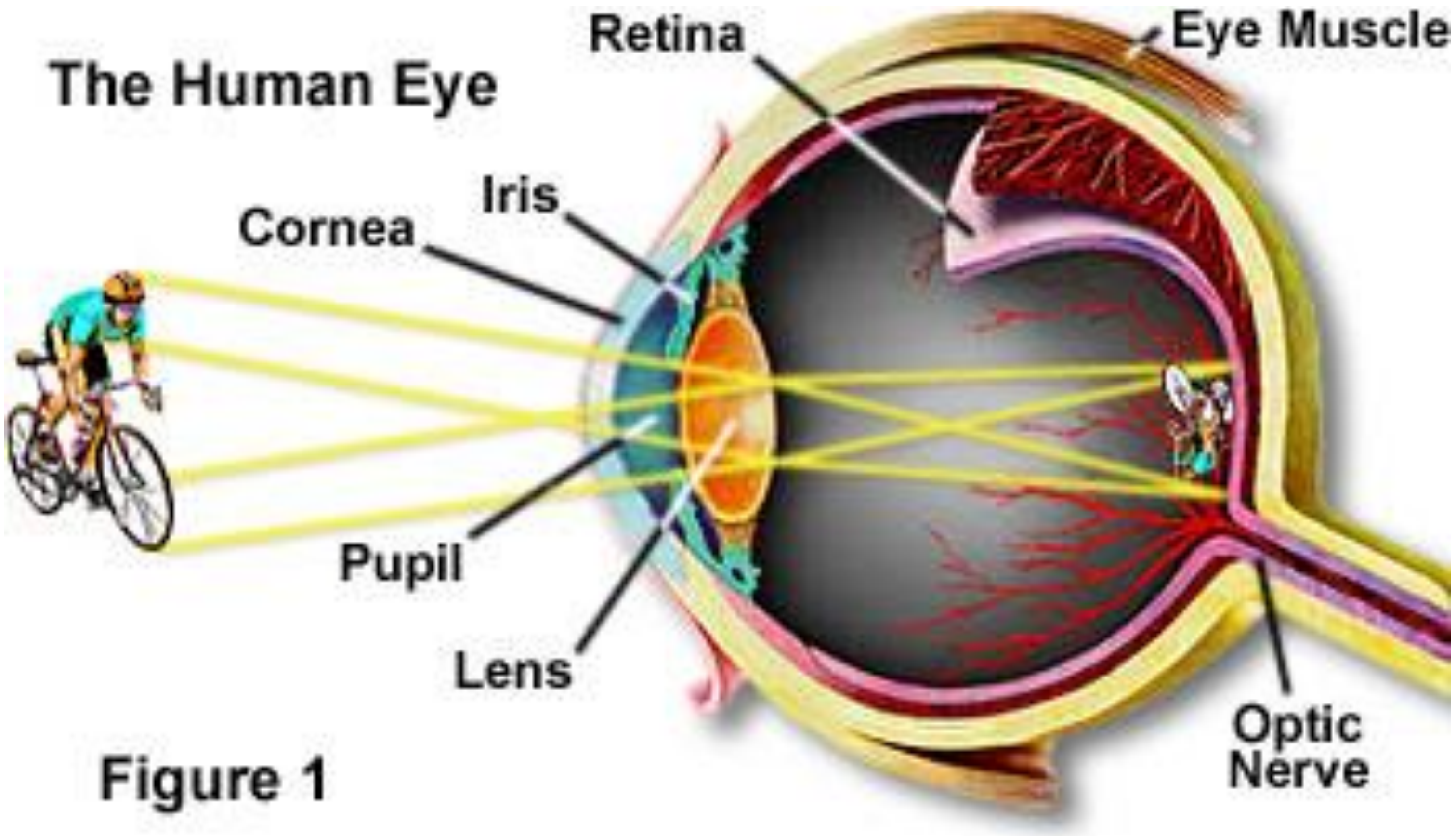
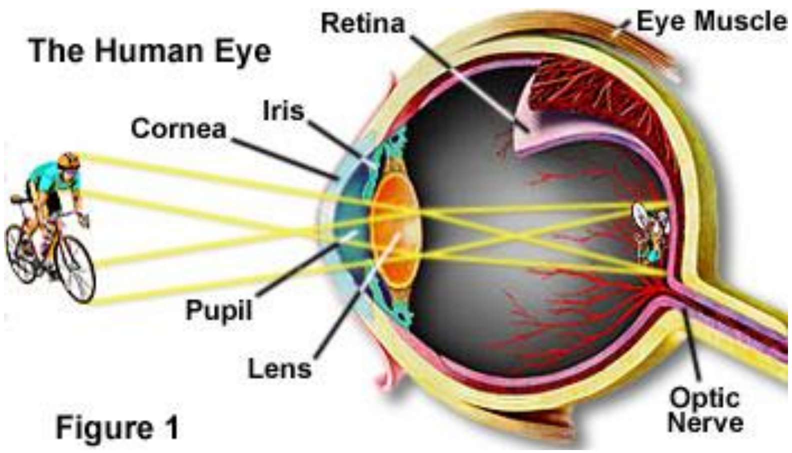
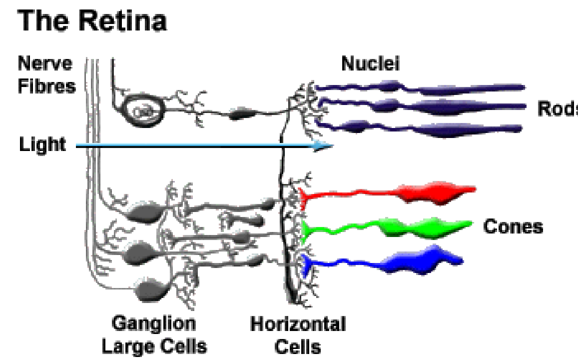
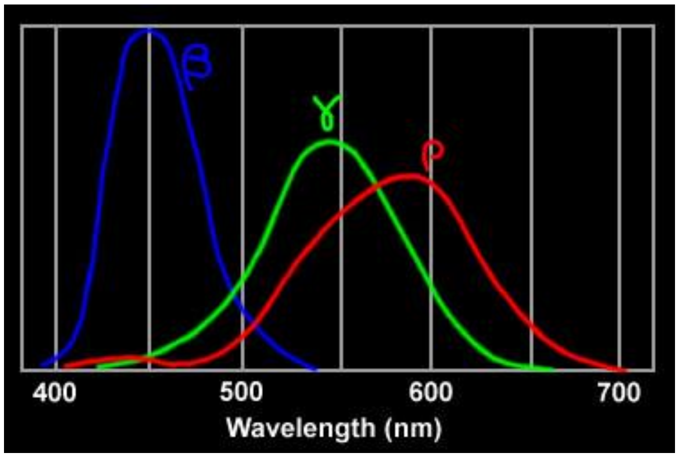


Image Courtesy to Florida State University

Human Color Perception



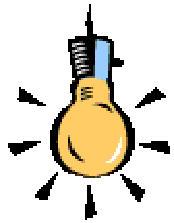


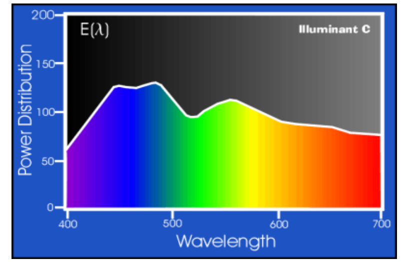


- A site about human color perception:
  - http://www.photo.net/photo/edscott/vis00010.htm

Light, Object and Sensor

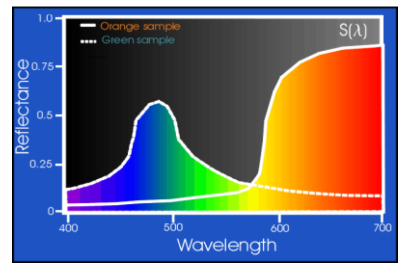
Light source





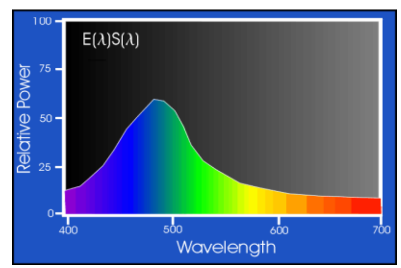
$e(\lambda)$

Object



$\rho(\lambda)$

Sensor



$f(\lambda)$

$$R = \int_\lambda e(\lambda)\rho(\lambda) f_R(\lambda) d\lambda, \quad G = \int_\lambda e(\lambda)\rho(\lambda) f_G(\lambda) d\lambda, \quad B = \int_\lambda e(\lambda)\rho(\lambda) f_B(\lambda) d\lambda$$

Image formation

- There are two parts to the image formation process:
  - The **geometry of image formation**, which determines where in the image plane the projection of a point in the scene will be located.
  - The **physics of light**, which determines the brightness of a point in the image plane as a function of illumination and surface properties.

Today's class: Image Formation

1. <u>Projective Geometry and Camera Models</u>
2. Light and Color Models
    2.1 Digital Image Presentation
    2.2 Light, Object and Sensor
    2.3 Color Systems
    2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

Pinhole Camera

- Abstract camera model - box with a small hole in it
- The simplest device to form an image of a 3D scene on a 2D surface.
- Rays of light pass through a "pinhole" and form an inverted image of the object on the image plane.
- Pinhole cameras work in practice

f = focal length
c = center of the camera

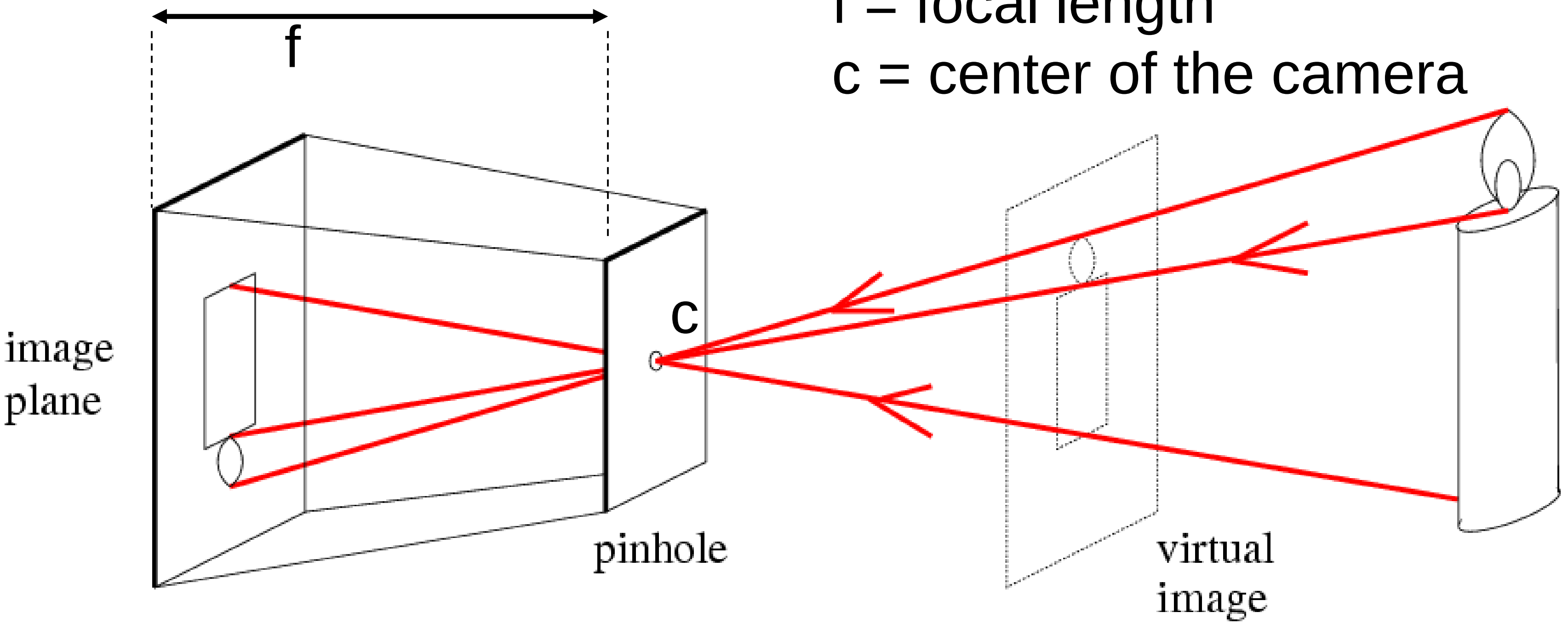
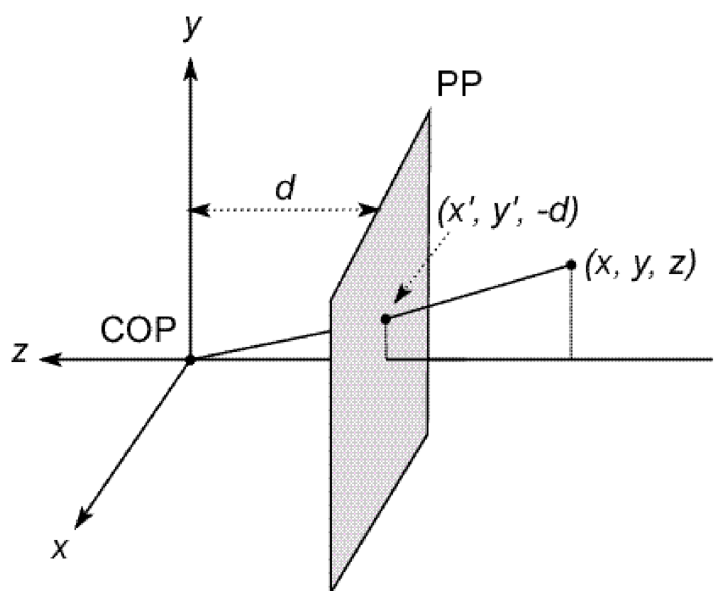


Figure from Forsyth

Modeling Projection: 3D->2D

The coordinate system

- We will use the pin-hole model as an approximation
- Put the optical center (**C**enter **O**f **P**rojection) at the origin
- Put the image plane (**P**rojection **P**lane) *in front* of the COP
- The camera looks down the *negative* z axis



Slide from Jinxiang Chai

Modeling Projection: 3D->2D

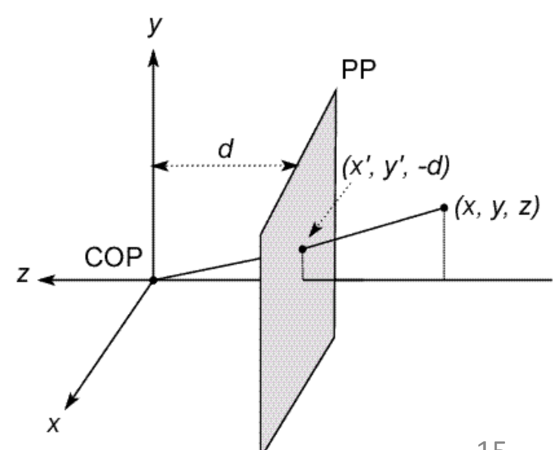
Projection equations

- Compute intersection with PP of ray from (x,y,z) to COP
- Derived using similar triangles

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z},\ -d)$$

- We get the projection by throwing out the last coordinate:

$$(x, y, z) \rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$



Slide from Jinxiang Chai

Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \ -d\frac{y}{z})$$

divide by third coordinate

- This is known as **perspective projection**
  - The matrix is the **projection matrix**
  - Can also formulate as a 4x4

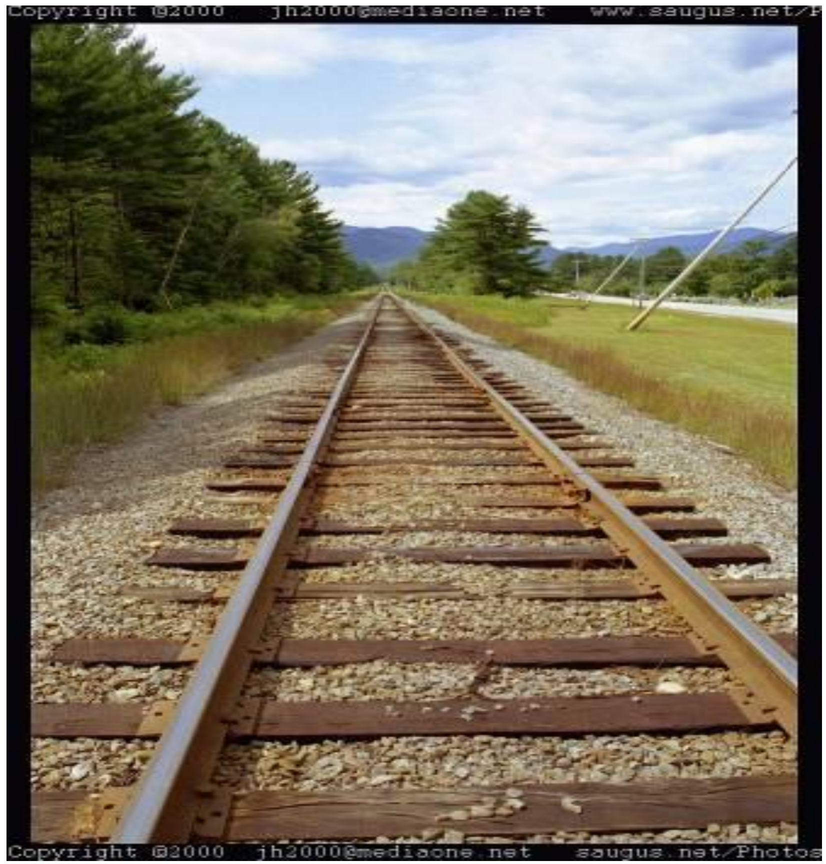
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \ -d\frac{y}{z})$$

divide by fourth coordinate

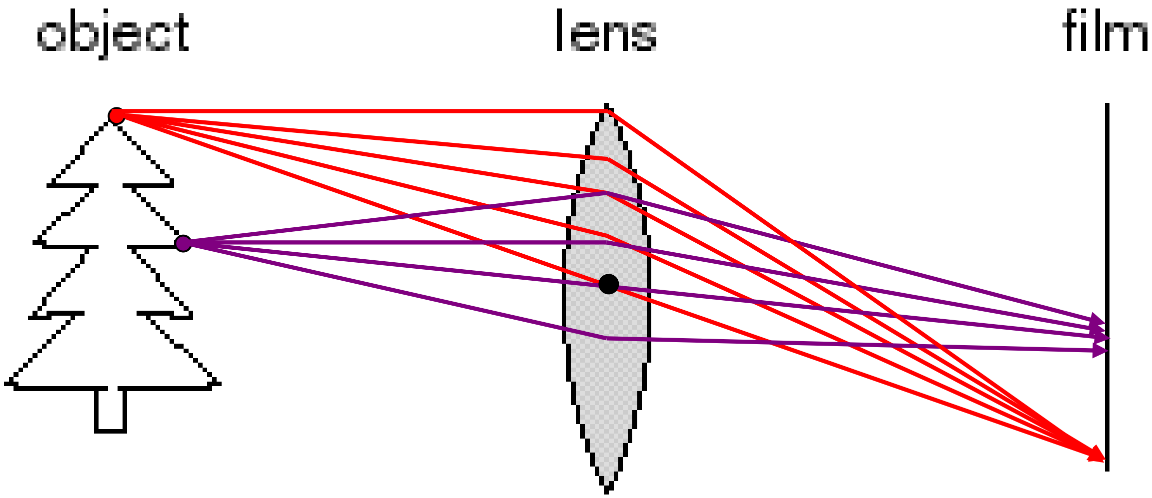
Vanishing Points and Lines

Parallel lines in the world intersect in the image at a "vanishing point"

$$\Rightarrow (-d\frac{x}{z}, \; -d\frac{y}{z})$$



From pin-hole camera to zoom lens



Slide by Steve Seitz

Putting It Together

From world coordinate to image coordinate

Viewport projection	Perspective projection	View transformation

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \Leftrightarrow \begin{bmatrix} s_x & 0 & u_0 \\ 0 & -s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_3^{\mathbf{T}} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

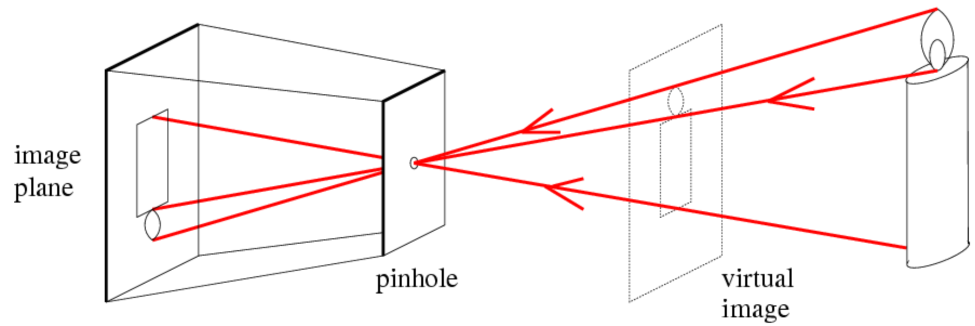
Image resolution, aspect ratio

Focal length

The relative position & orientation between camera and objects

Camera Parameters

Totally 11 parameters,



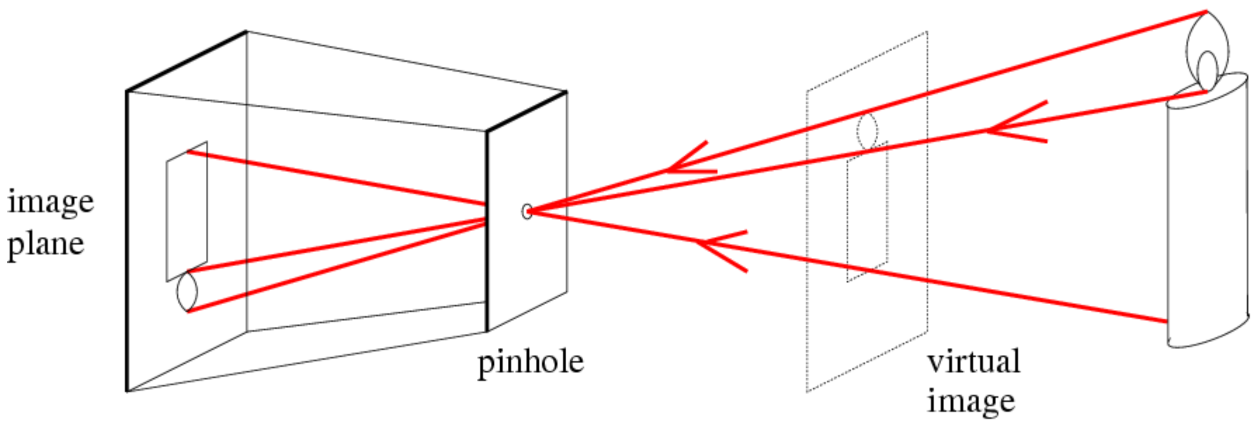
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \Leftrightarrow \begin{vmatrix} s_x & 0 & u_0 \\ 0 & -s_y & v_0 \\ 0 & 0 & 1 \end{vmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_3^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

Intrinsic camera parameters

extrinsic camera parameters

Things to Remember

- Pinhole camera model and camera matrix

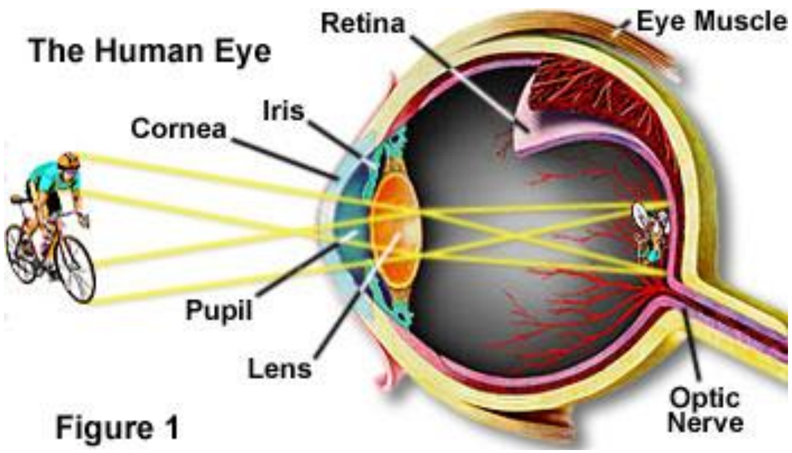


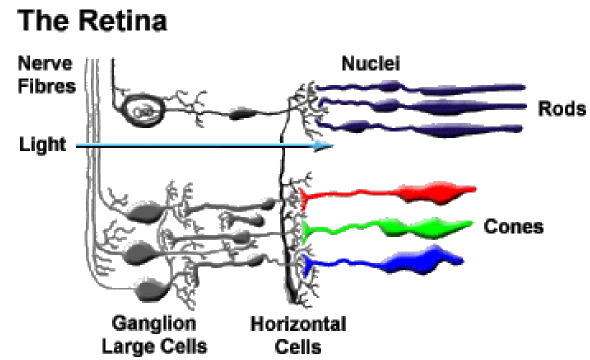
Today's class: Image Formation

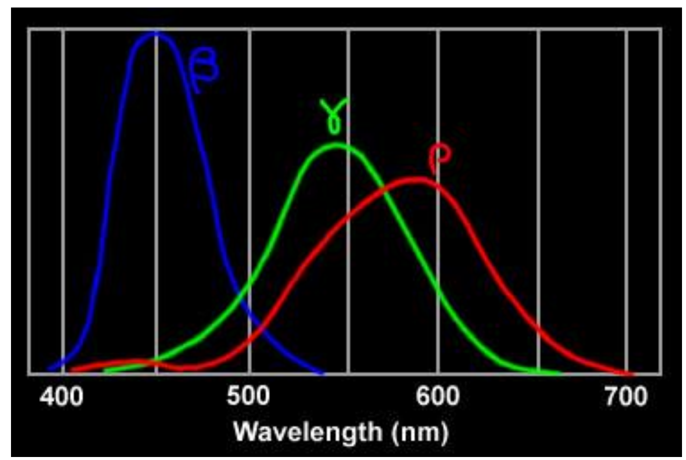
1. Projective Geometry and Camera Models
2. Light and Color Models
    - 2.1 Digital Image Presentation
    - 2.2 Light, Object and Sensor
    - 2.3 Color Systems
    - 2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

Human Color Perception







- A site about human color perception:
  - http://www.photo.net/photo/edscott/vis00010.htm

Sensor Array

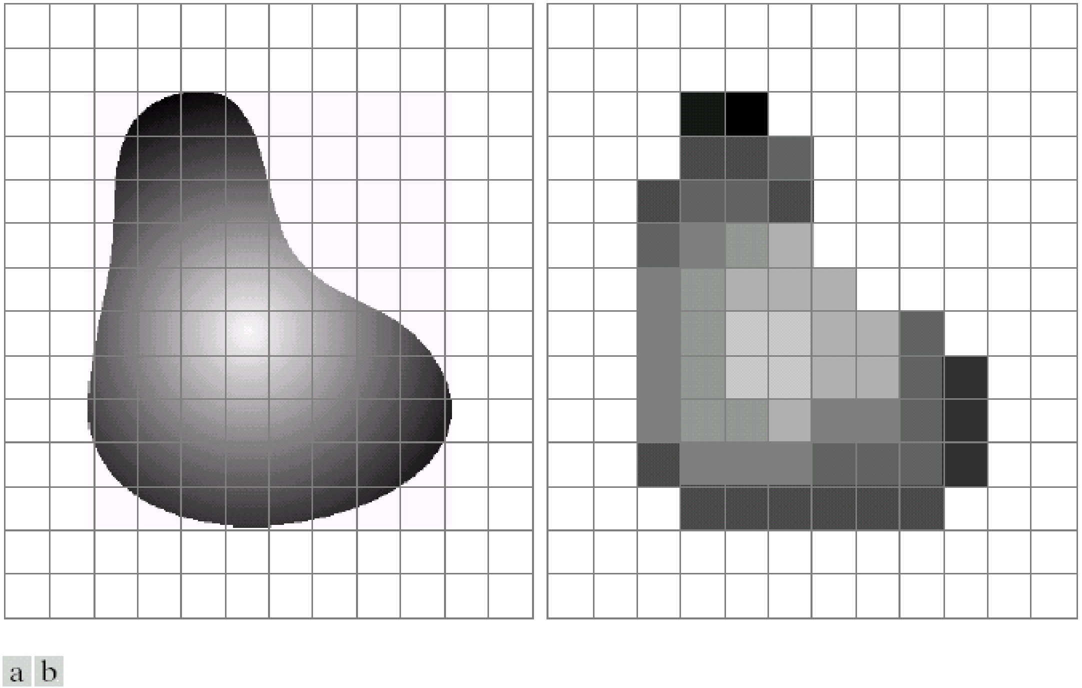
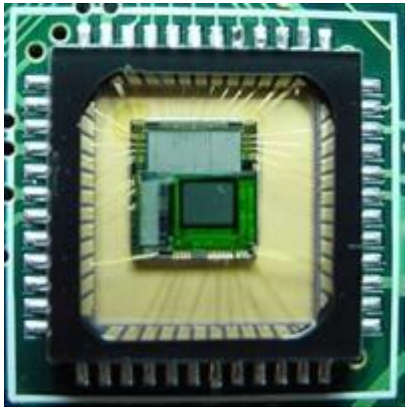


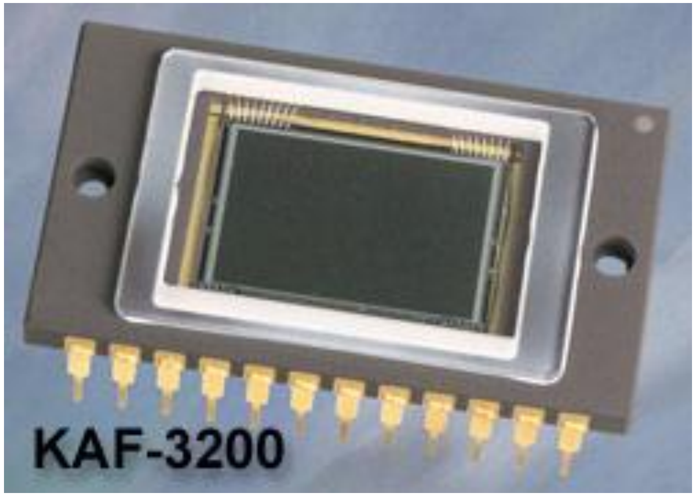
FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.



CMOS sensor

Image Sensors : Array Sensor

Charge-Coupled Device (CCD)

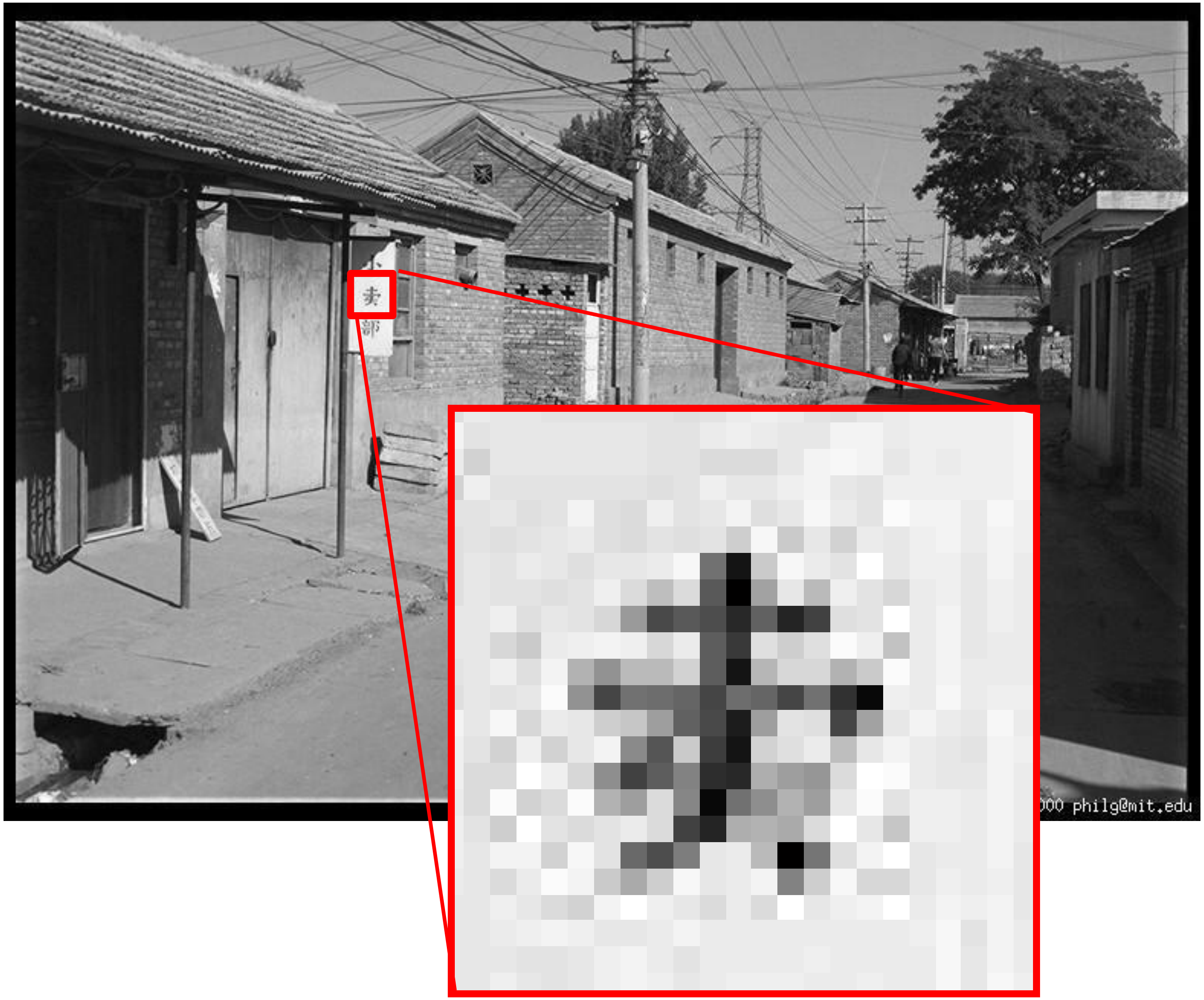


CCD KAF-3200E from Kodak.
(2184 x 1472 pixels,
Pixel size 6.8 microns^2)

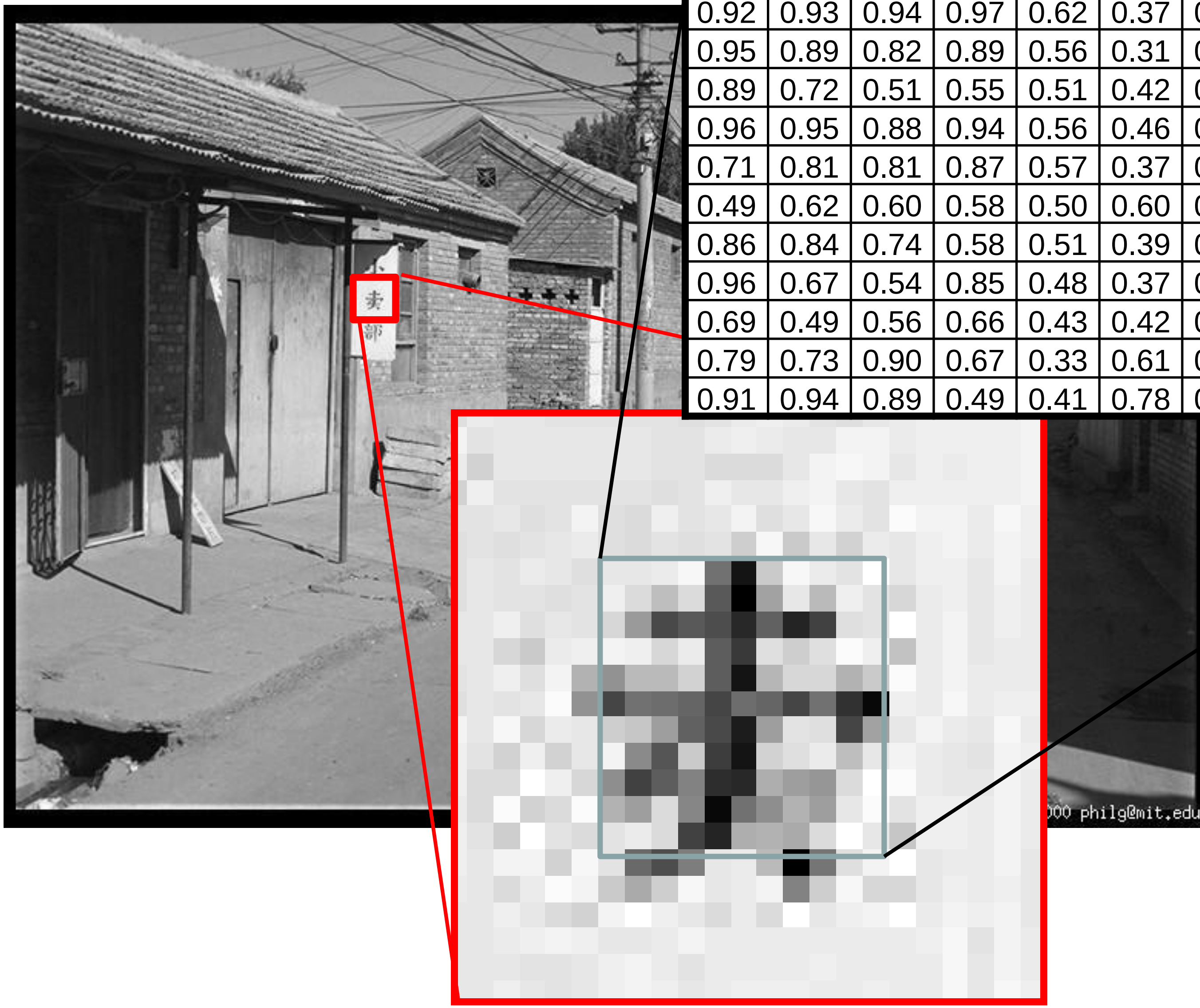
- Used for convert a continuous image into a digital image
- Contains an array of light sensors
- Converts photon into electric charges accumulated in each sensor unit

Digital image representation

The digital image (pixel matrix)



The digital image (pixel matrix)



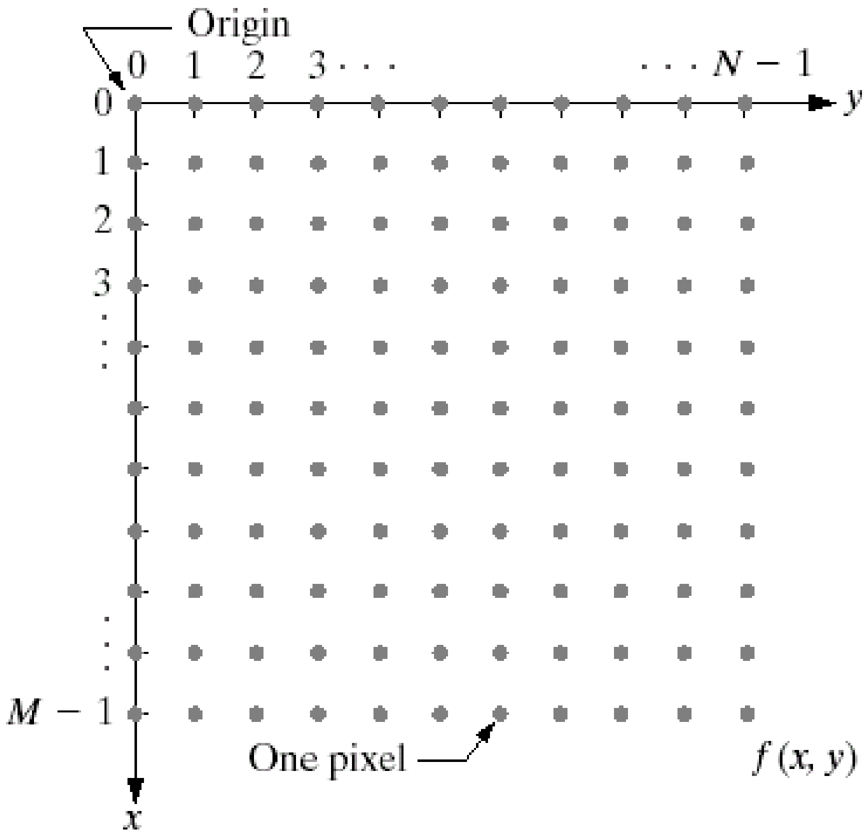
0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

Fundamentals of Digital Images

Image "After snow storm"

- **An image=a function of spatial coordinates**.
- **Spatial coordinate**: (x,y) for 2D case such as photograph,
  (x,y,z) for 3D case such as CT scan images
  (x,y,t) for video.
- The function f may represent intensity (for greyscale images) or color (for color images) or other associated values.

Conventional Coordinate for Image Representation

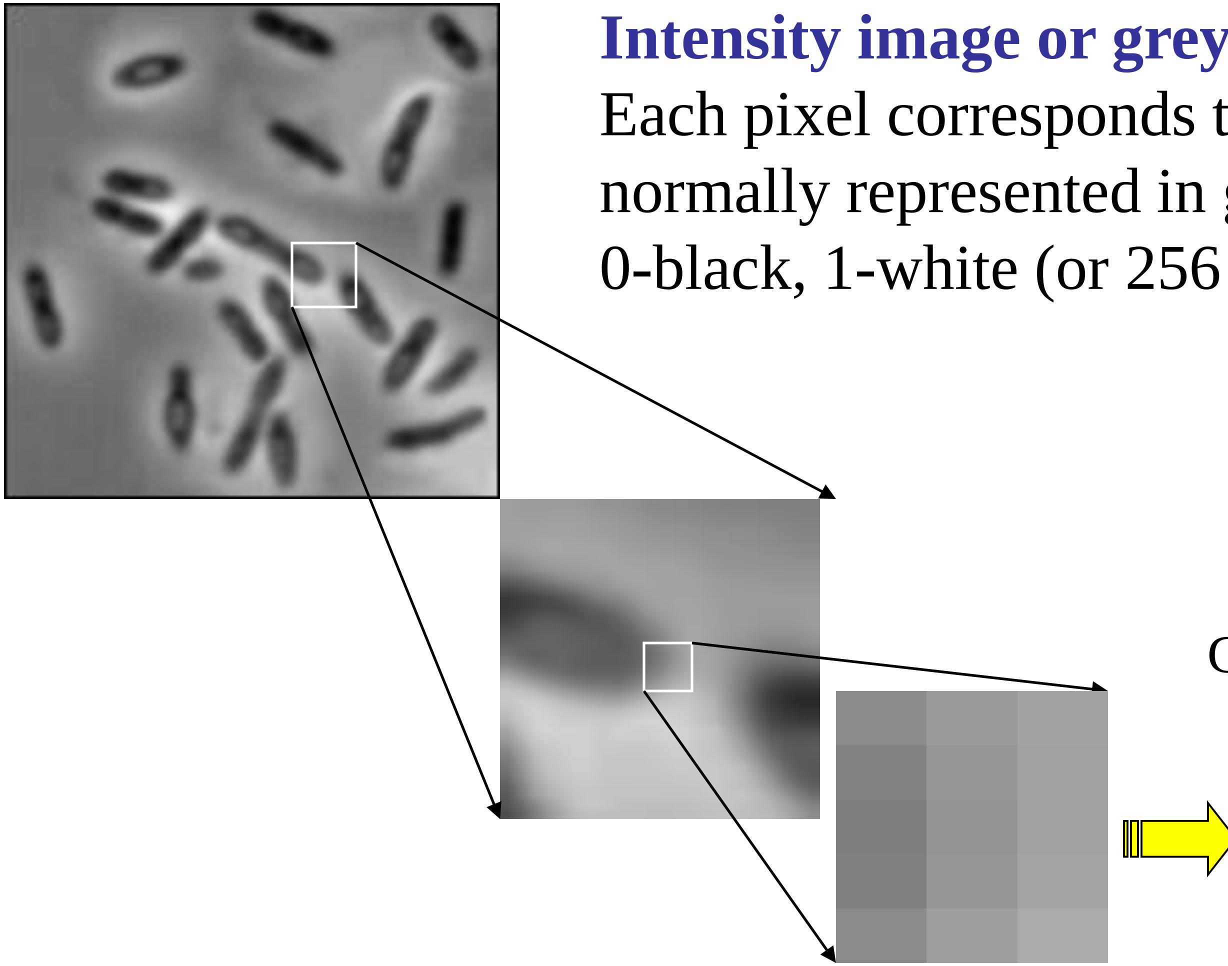


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Grey-scale Image

Intensity image or grey-scale image

Each pixel corresponds to light intensity normally represented in gray scale, e.g. 0-black, 1-white (or 256 in 8-bit scale)



Gray scale values

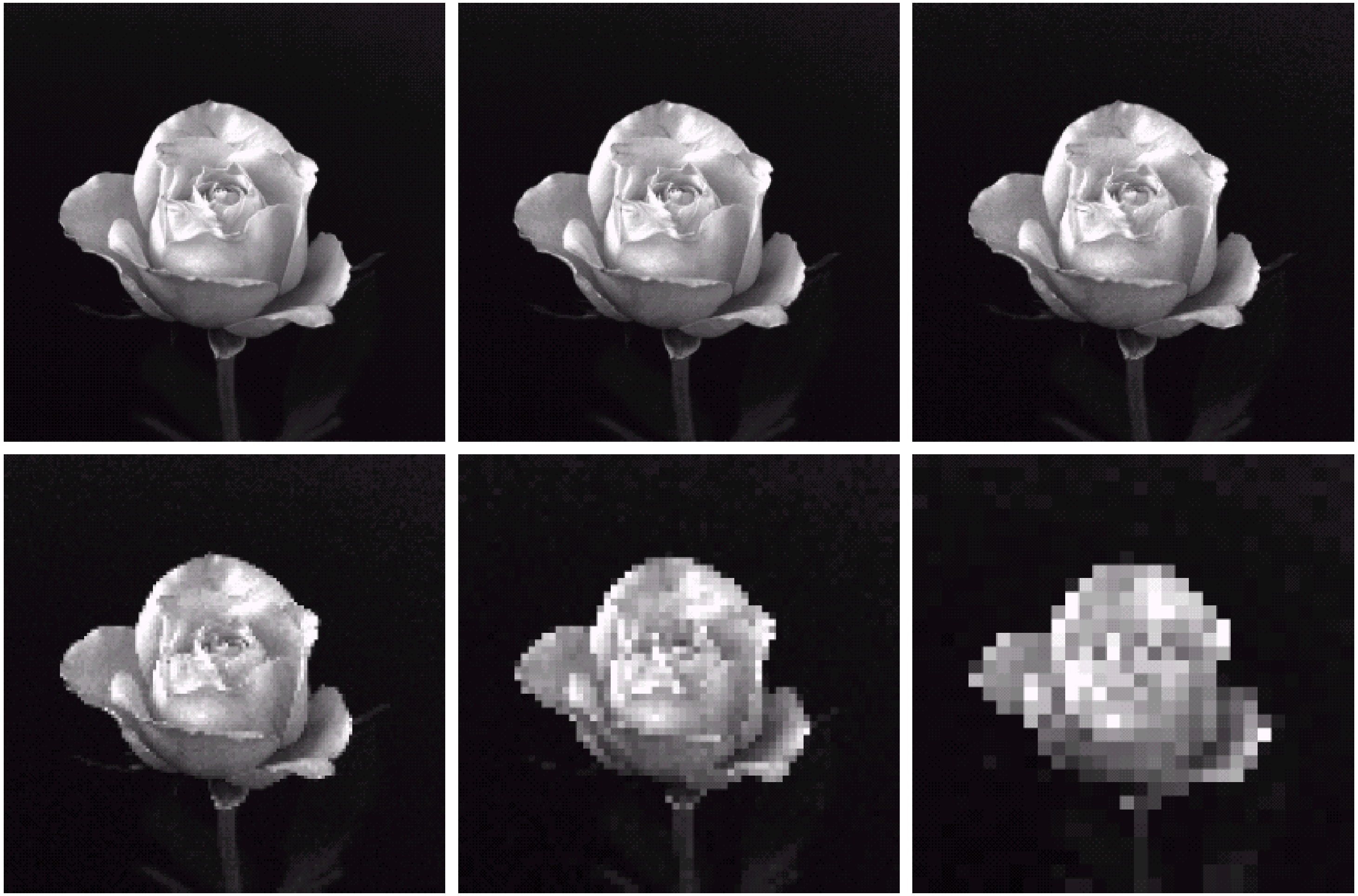
$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

Effect of Spatial Resolution

FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Effect of Spatial Resolution



a b c
d e f

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Effect of Quantization Levels

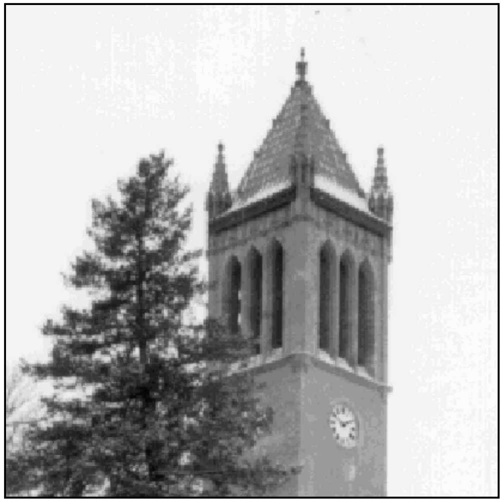
256 levels (8-bit)

128 levels (7-bit)

64 levels (6-bit)

32 levels (5-bit)

Effect of Quantization Levels (cont.)



16 levels (4-bit)

8 levels (3-bit)

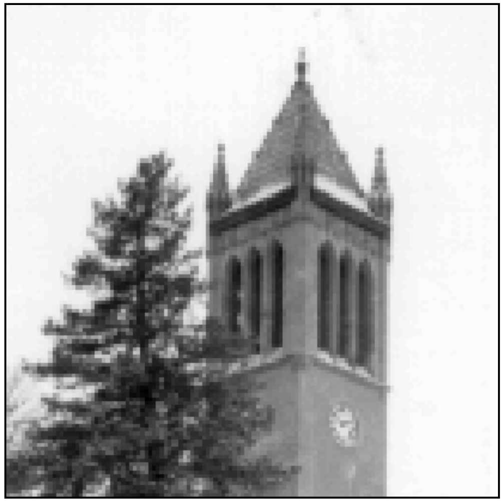
In this image, it is easy to see false contour.

4 levels (2-bit)

2 levels (1-bit)

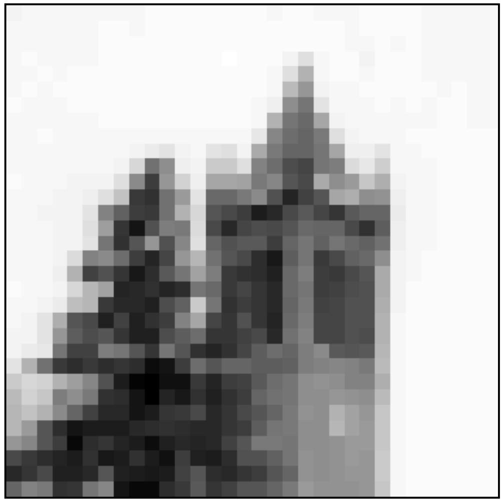
Effect of Spatial Resolution

256x256 pixels



128x128 pixels

64x64 pixels



32x32 pixels

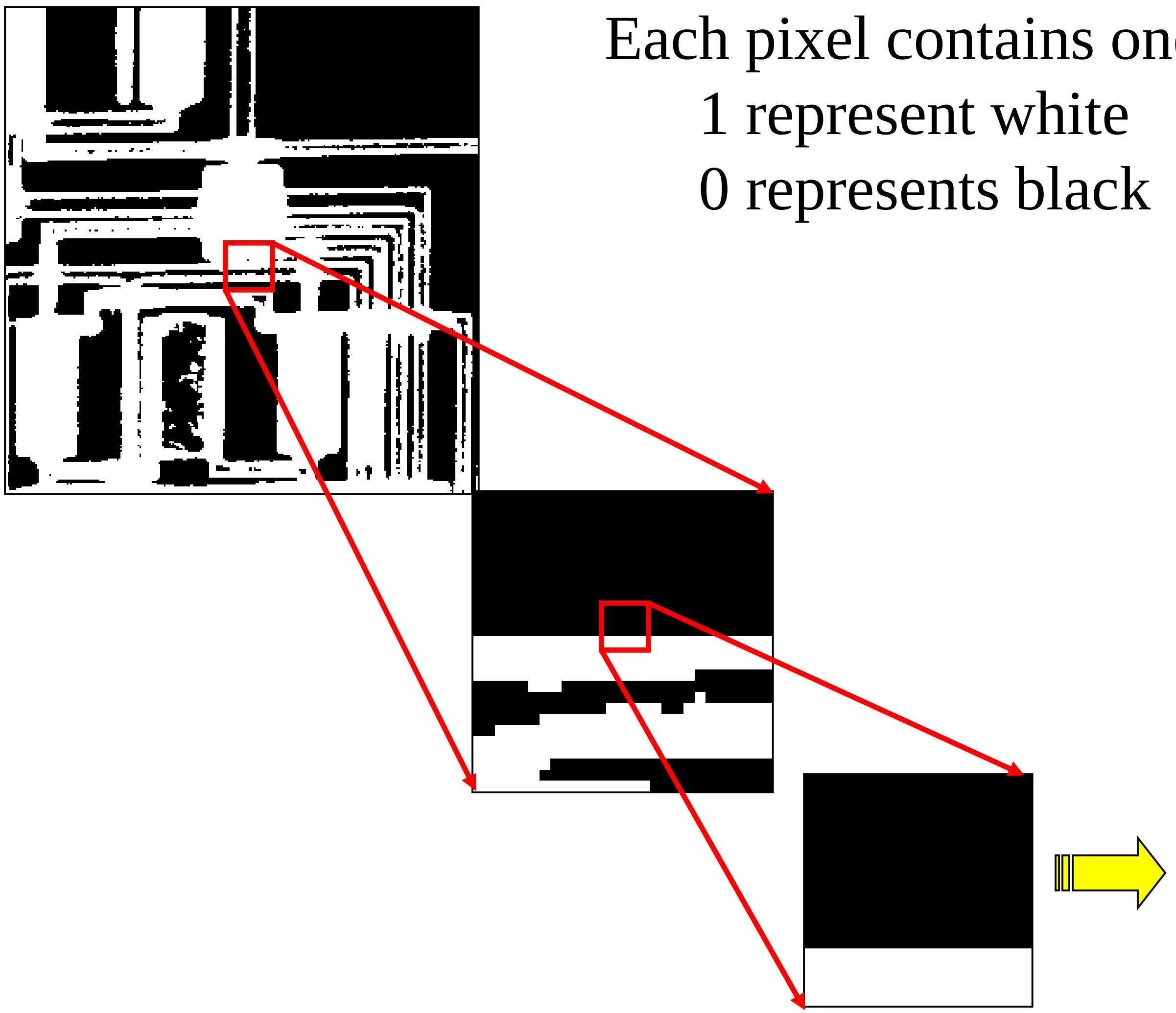
Two-level image: binary image

Binary image or black and white image

Each pixel contains one bit :

1 represent white

0 represents black



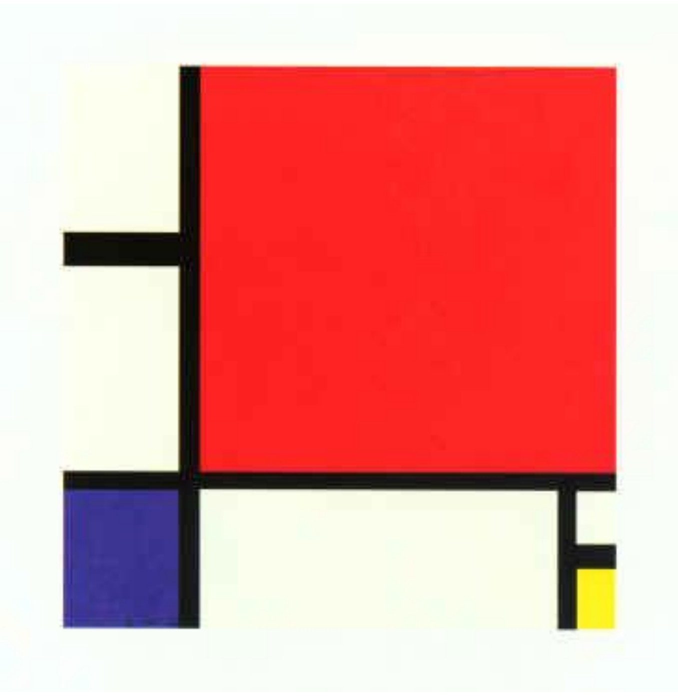
Binary data

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Colour images

A colour image

The science of Light and Colour
Fundamentals of colour science
Hall of fame



Mondrian

Pythagoras: undulation theory

Aristoteles: curpus theory

Newton 1665 “Opticks”

Planck, Einstein and Bohr “Quantum mechanics”

Goethe 1840 “Farbenlehre”

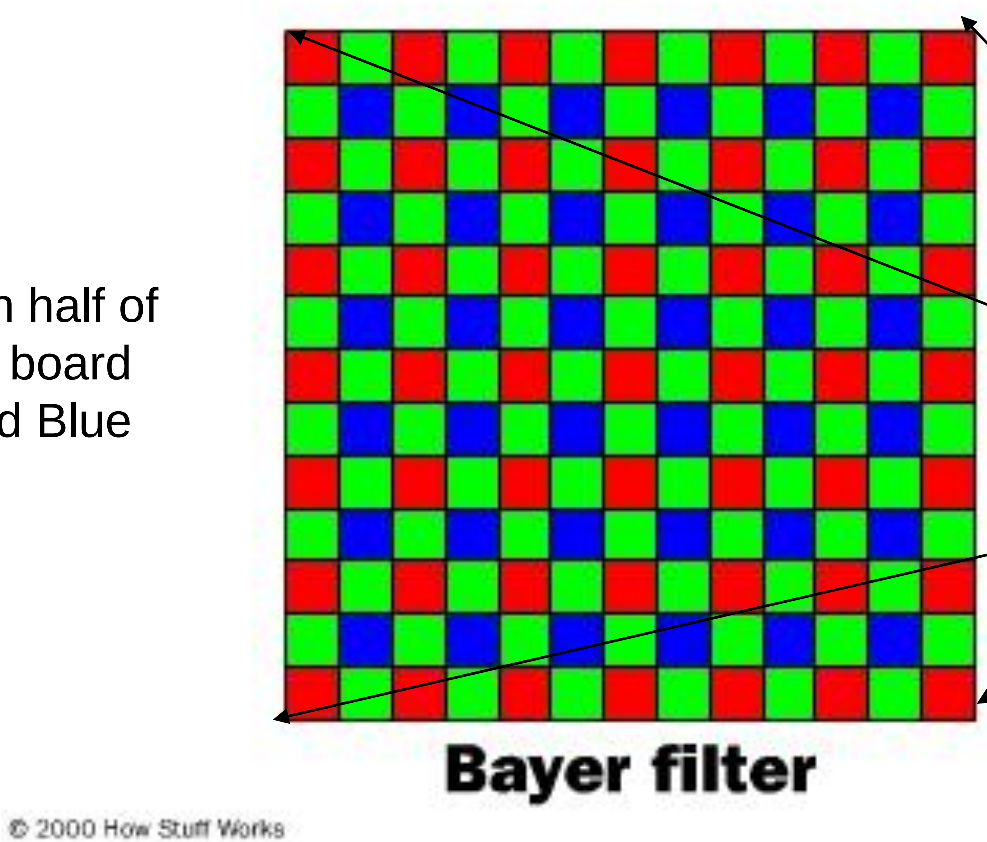
Munsell 1905 “A Colour Notation”

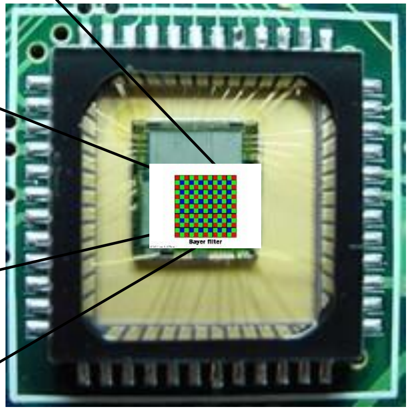
Descartes, Schopenhauer,

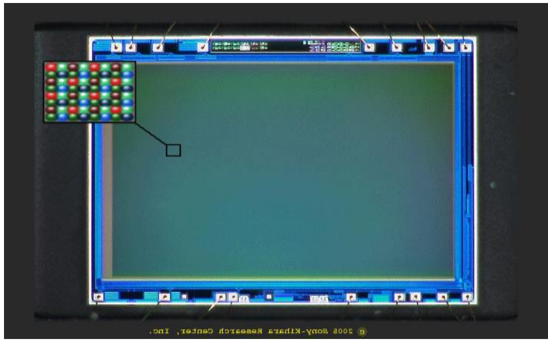
Hegel, Wittgenstein…and many others

How to form colour images?

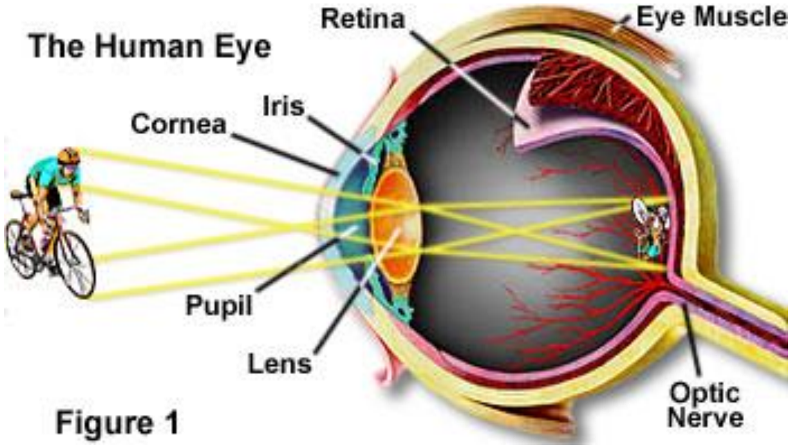
Bayer filter:
Green fills in half of the checker board and Red and Blue fill the rest.

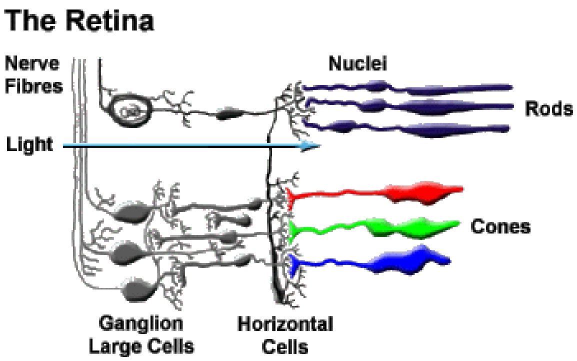


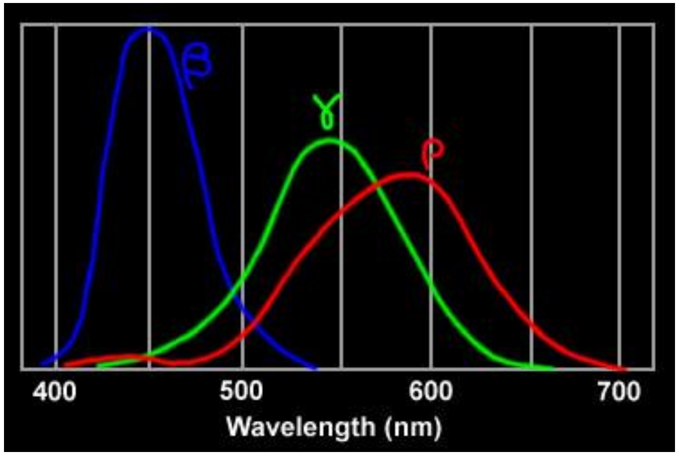




Human Color Perception



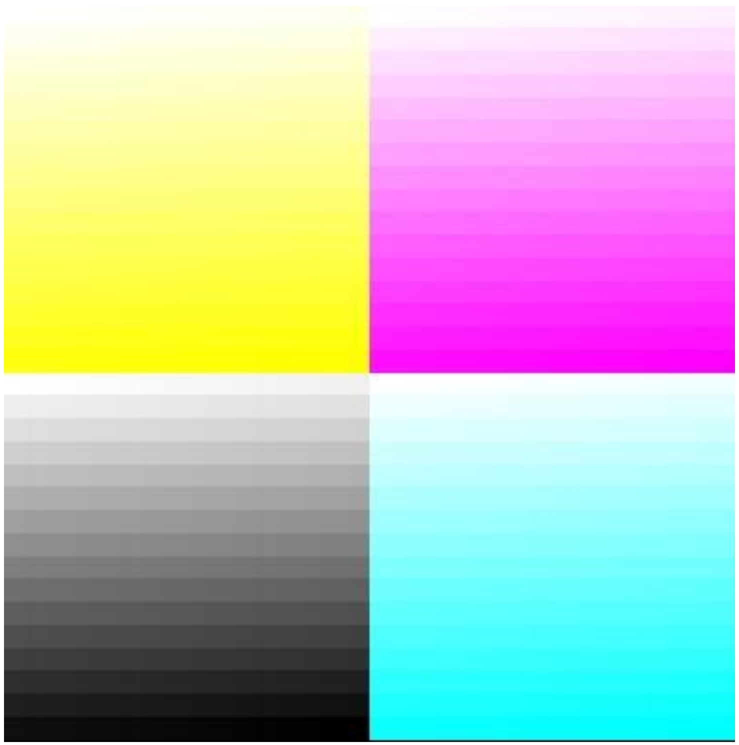




- A site about human color perception:
  - http://www.photo.net/photo/edscott/vis00010.htm

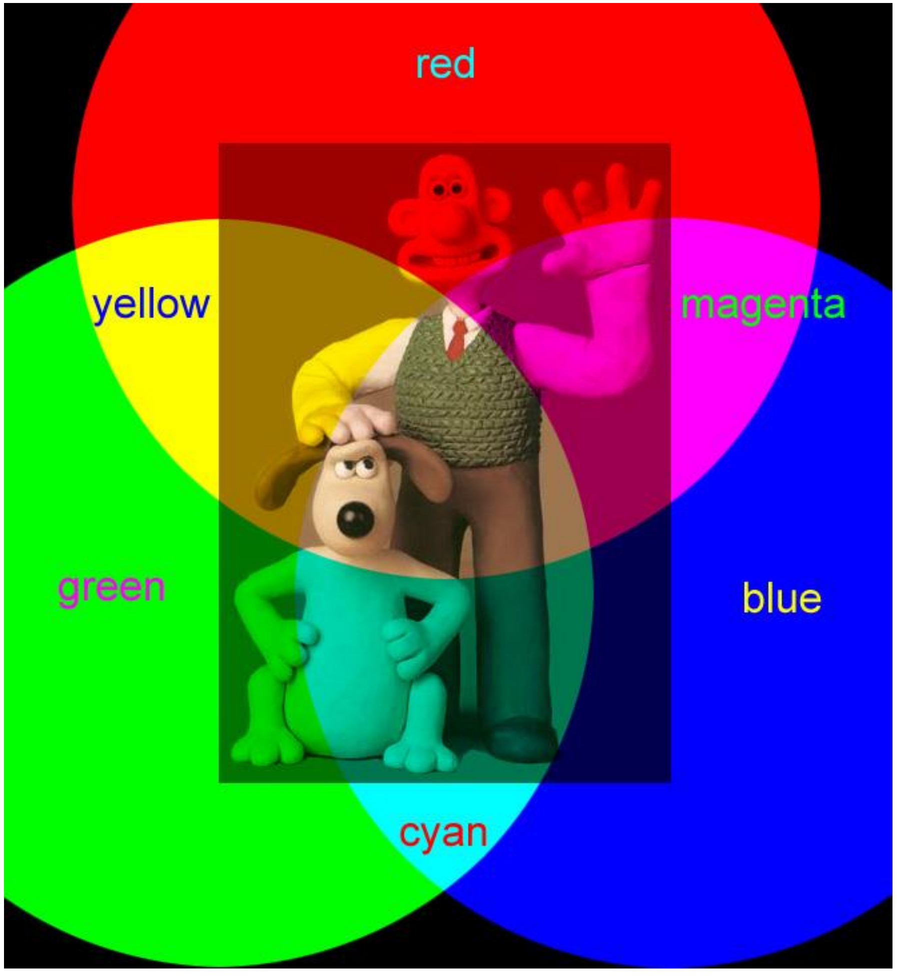
How many colors can be seen?

- Human eyes can distinguish about
  - 128 different hues (colors)
  - 130 different saturation (colorfulness) levels



Colour Images

- Are constructed from three intensity maps.
- Each intensity map is projected through a color filter (*e.g.*, red, green, or blue) to create a monochrome image.
- The intensity maps are overlaid to create a color image.
- Each pixel in a color image is a three element vector.



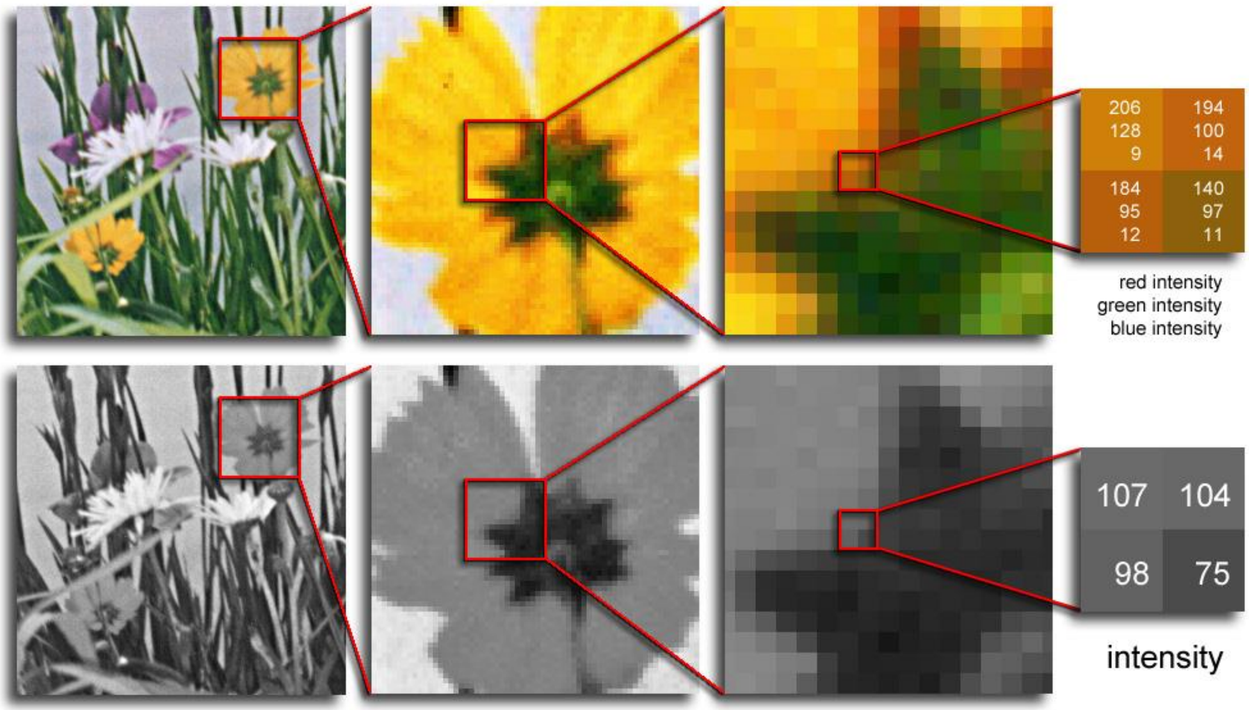
44

Colour Image

Colour images have 3 values per pixel; greyscale images have 1 value per pixel.

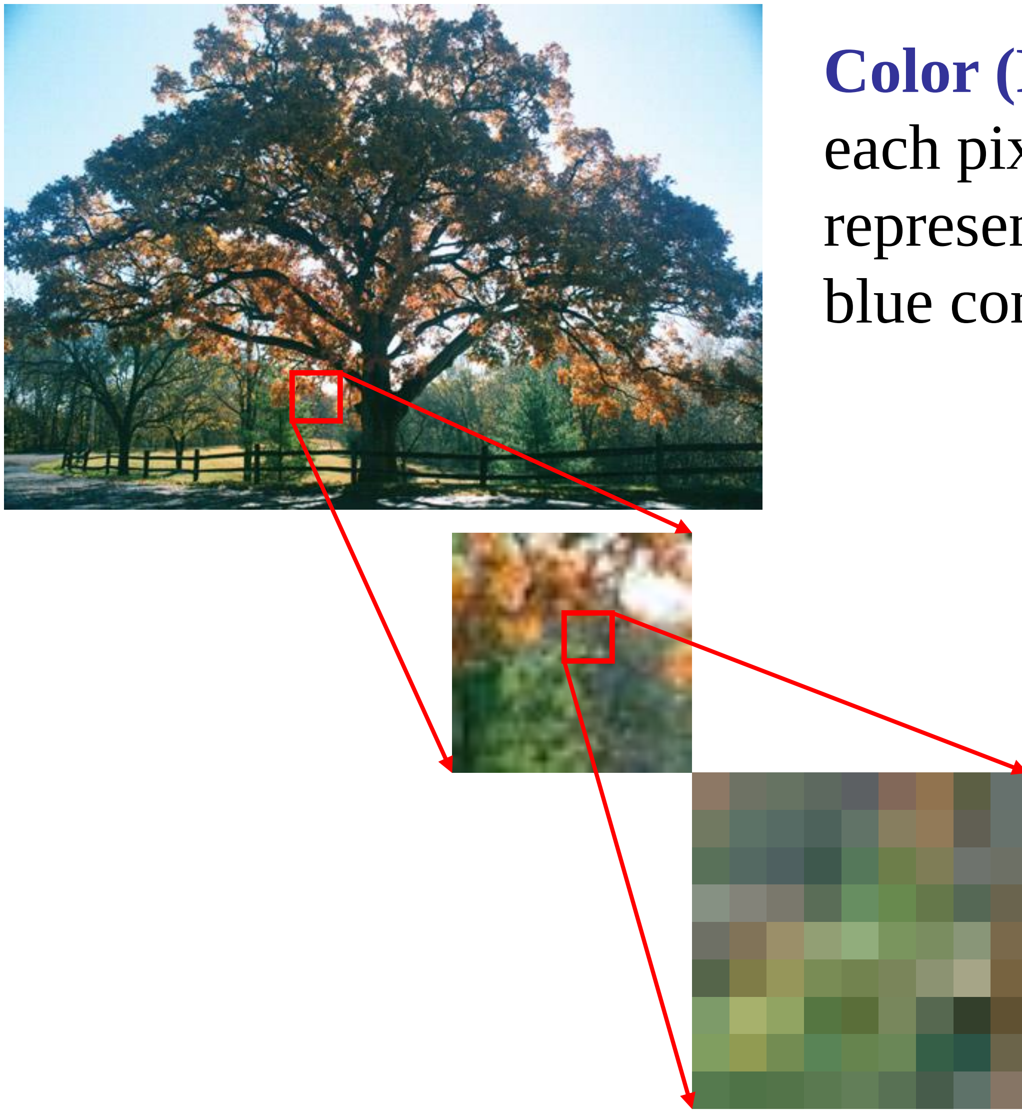
a grid of squares, each of which contains a single colour

each square is called a pixel (for *picture element*)



45

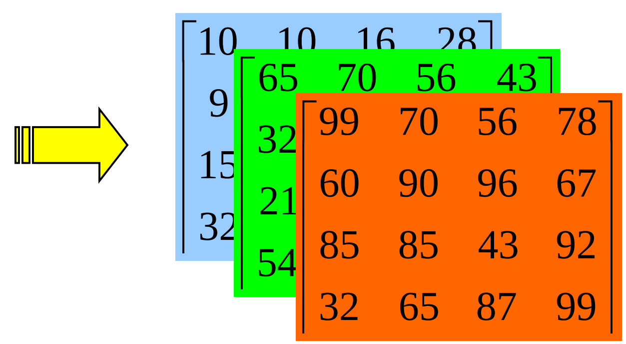
Colour Image



Color (RGB) image:
each pixel contains a vector representing red, green and blue components.

a multidimensional array of numbers (such as intensity image) or vectors (such as color image)

RGB components



Colour Images in Matlab

- Images represented as a matrix
- Suppose we have a NxM RGB image called “im”
  - im(1,1,1) = top-left pixel value in R-channel
  - im(y, x, b) = y pixels down, x pixels to right in the bth channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - **Convert to double format (values 0 to 1) with im2double(). (important !)**

column

row

R

0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

G

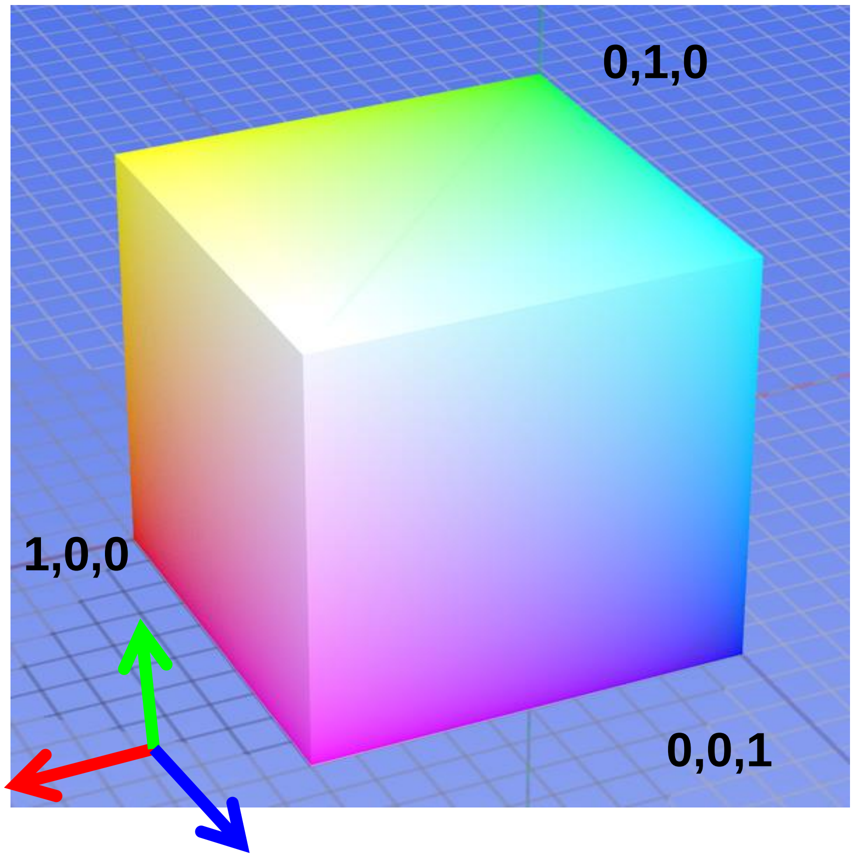
									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

B

									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

RGB space

Default color space



R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)

Some drawbacks

- Strongly correlated channels
- Not perceptually meaningful.

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Today’s class: Image Formation

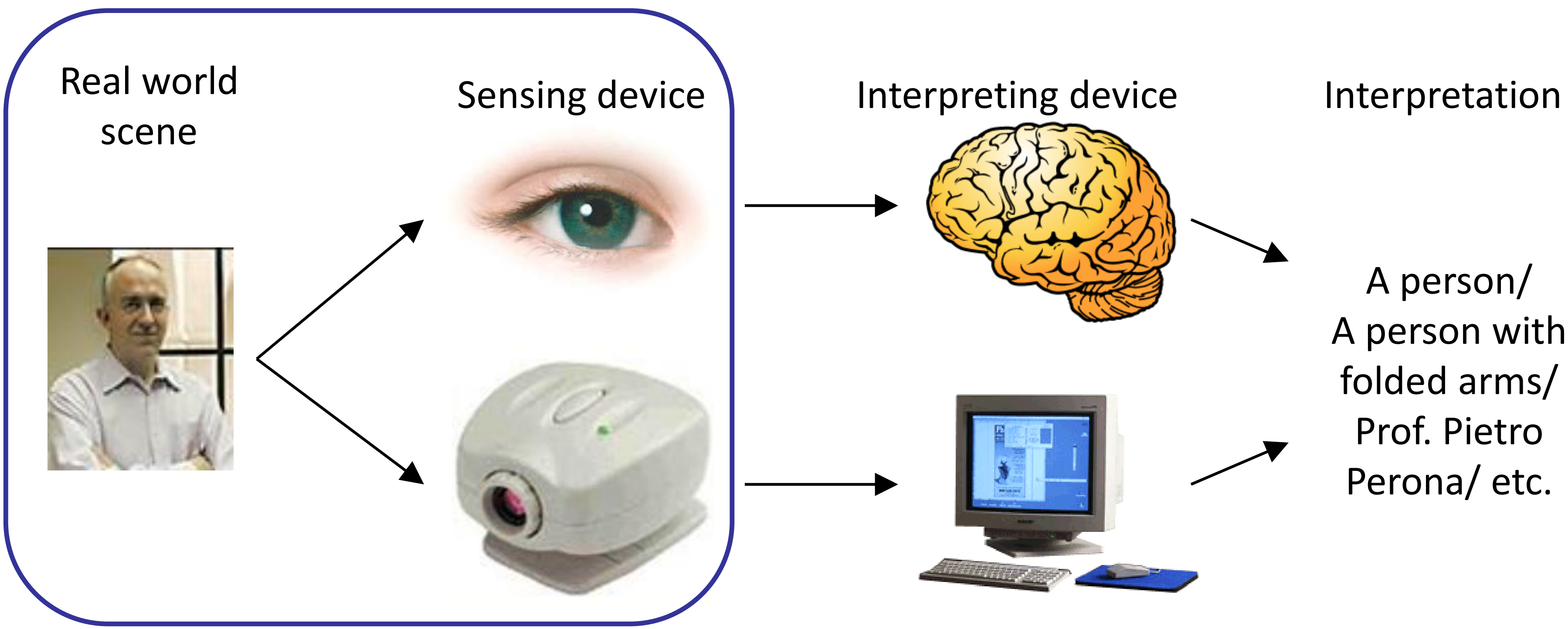
1. Projective Geometry and Camera Models
2. <u>Light and Color Models</u>
   - 2.1 Digital Image Presentation
   - 2.2 <u>Light, Object and Sensor</u>
   - 2.3 Color Systems
   - 2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

The computer vision problem

- Make a computer to see and to understand images

- We know it is physically possible – we do it every day and effortlessly!



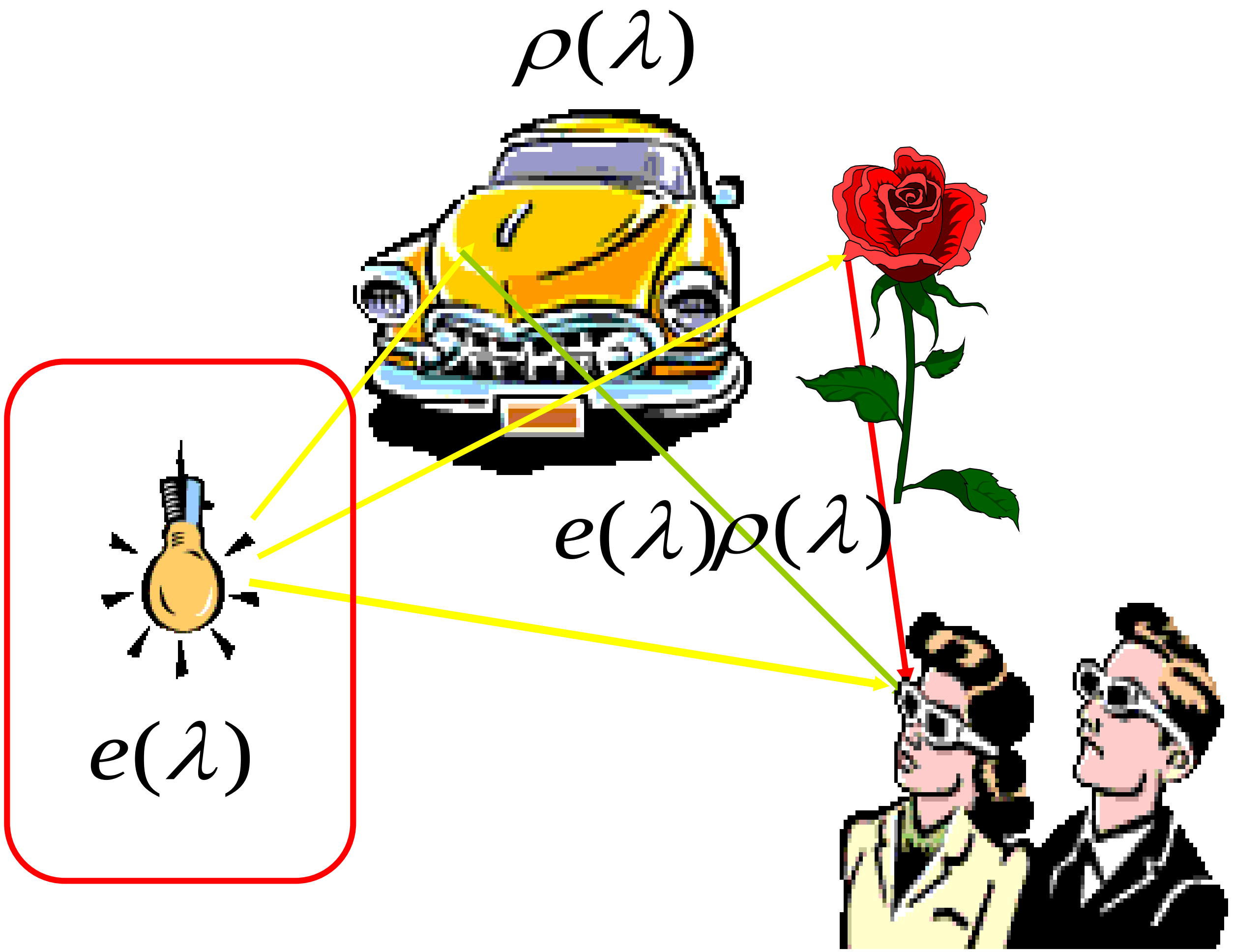
What makes an image?

the triplet light-objects-observer

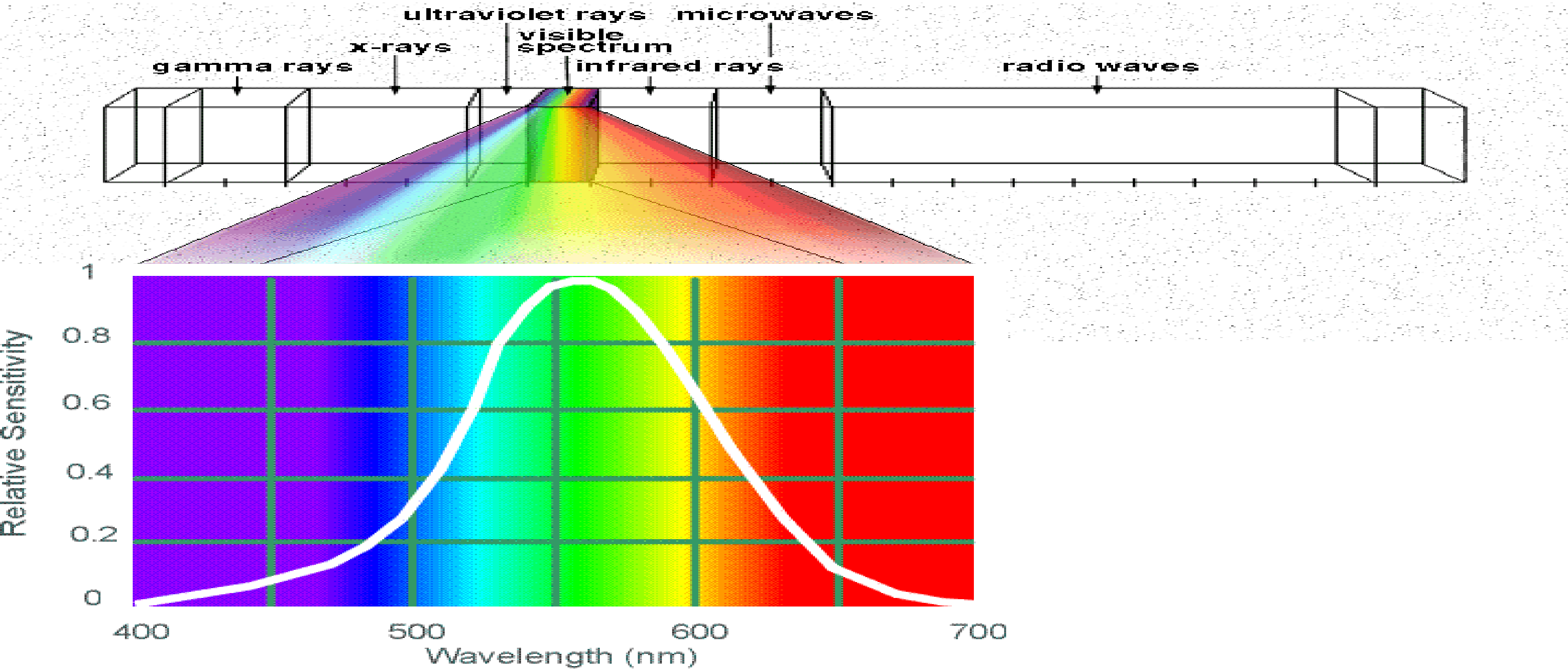
Light source

Object(s)

Sensor



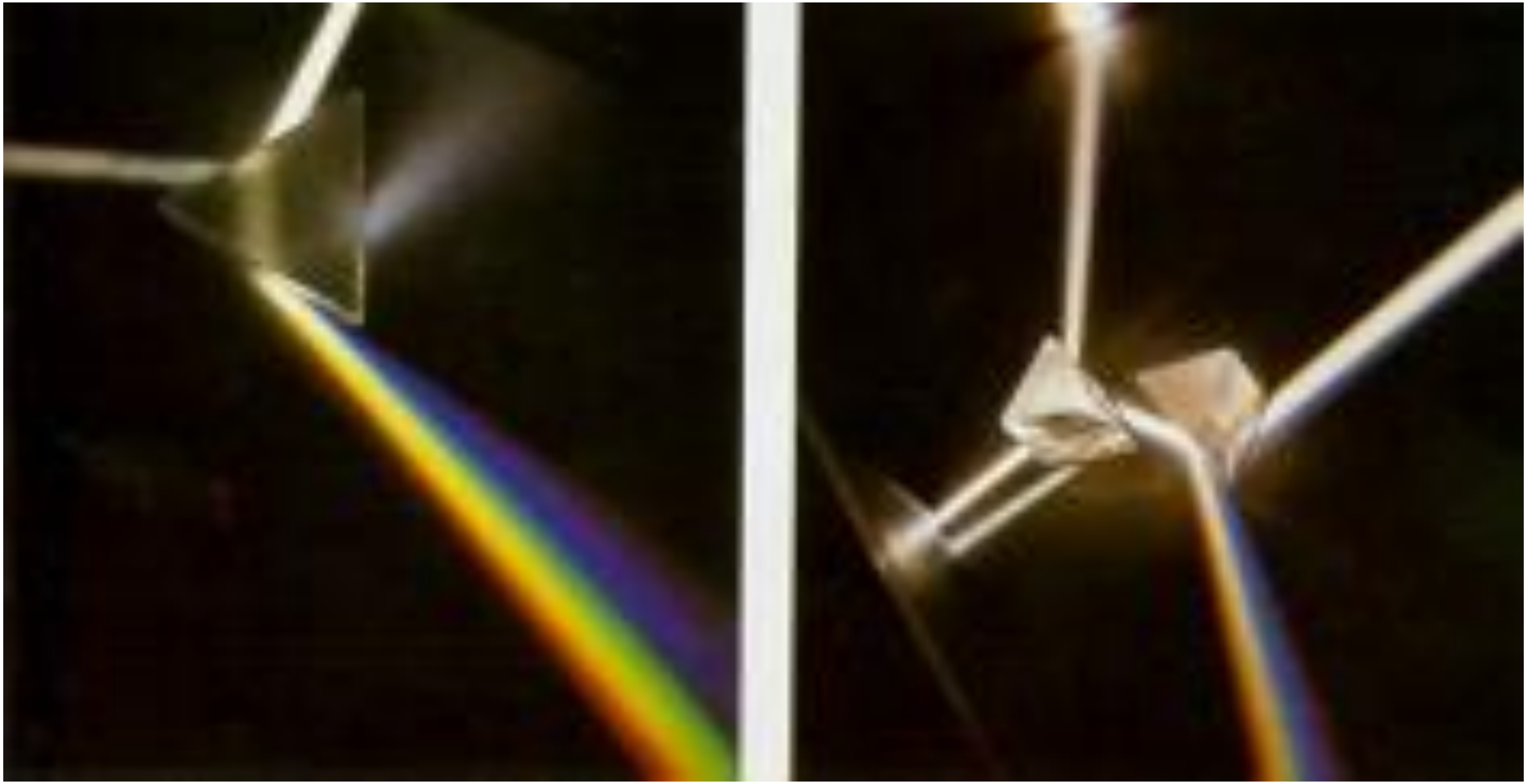
Electromagnetic Spectrum



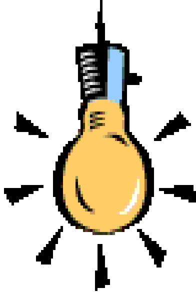
Human Luminance Sensitivity Function

http://www.yorku.ca/eye/photopik.htm

Electromagnetic Spectrum



Light sources and illuminants



Light sources:

sun, candle,

fluorescent lamp,

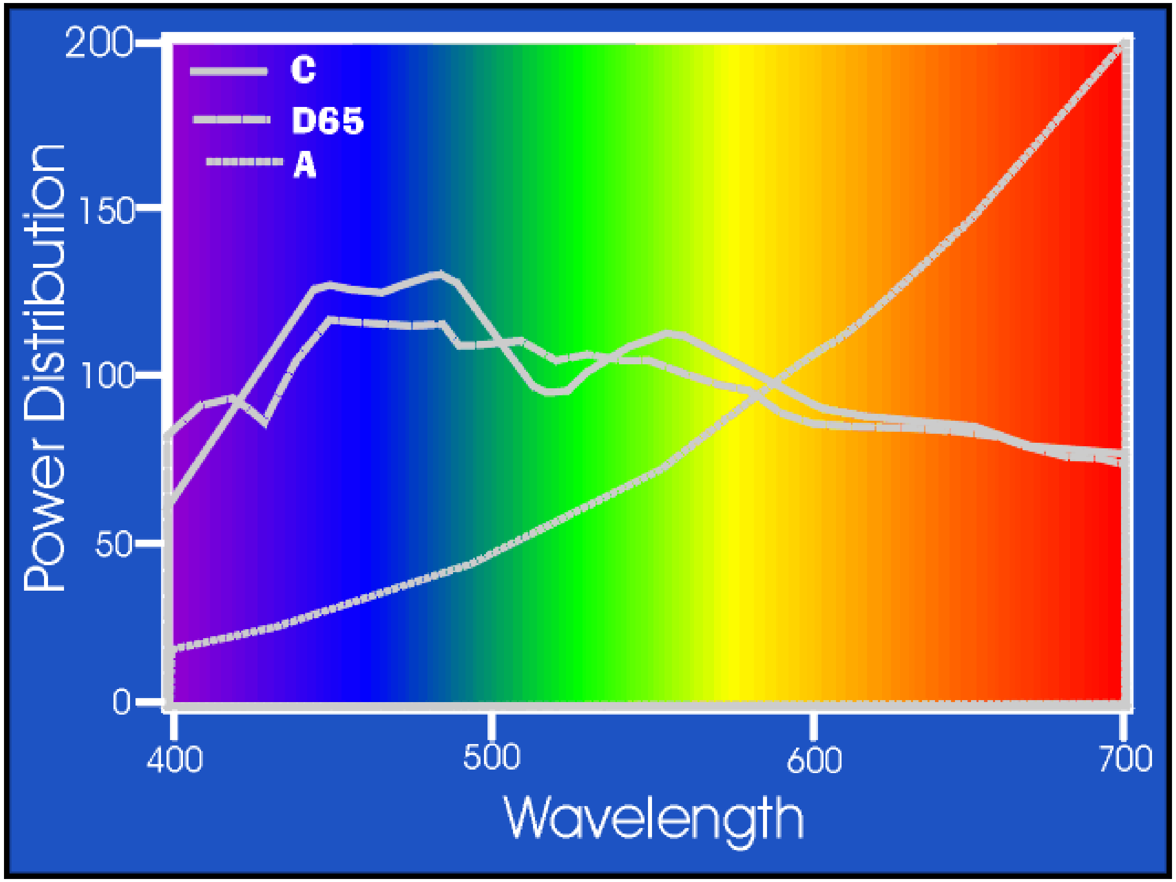
incandescent lamp

Illuminants:

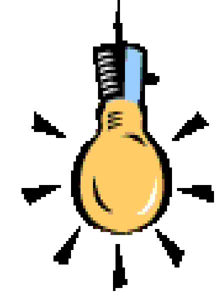
illuminant A

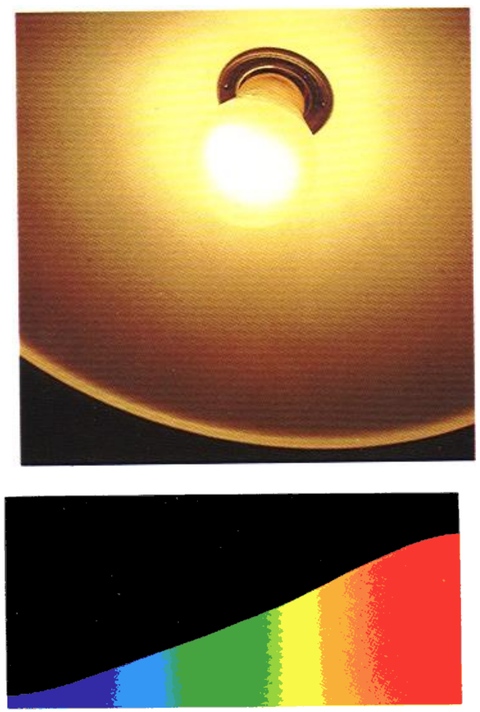
illuminant D65

illuminant C

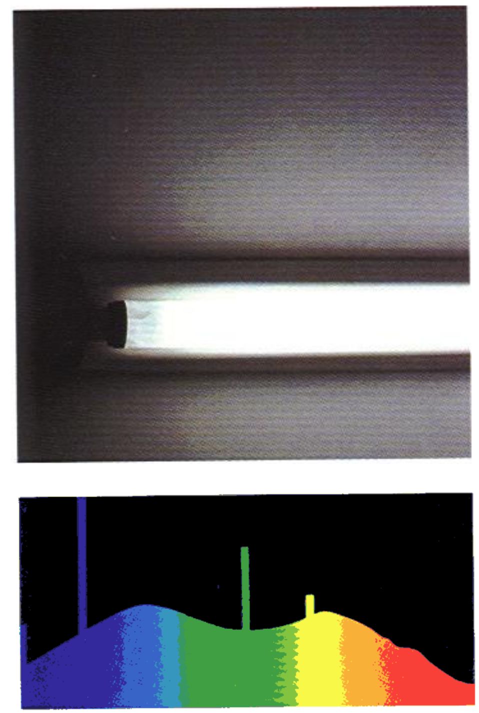


Light sources and illuminants



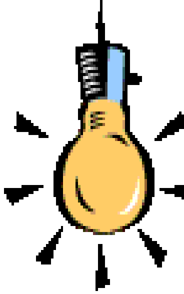


Incandescent lamp

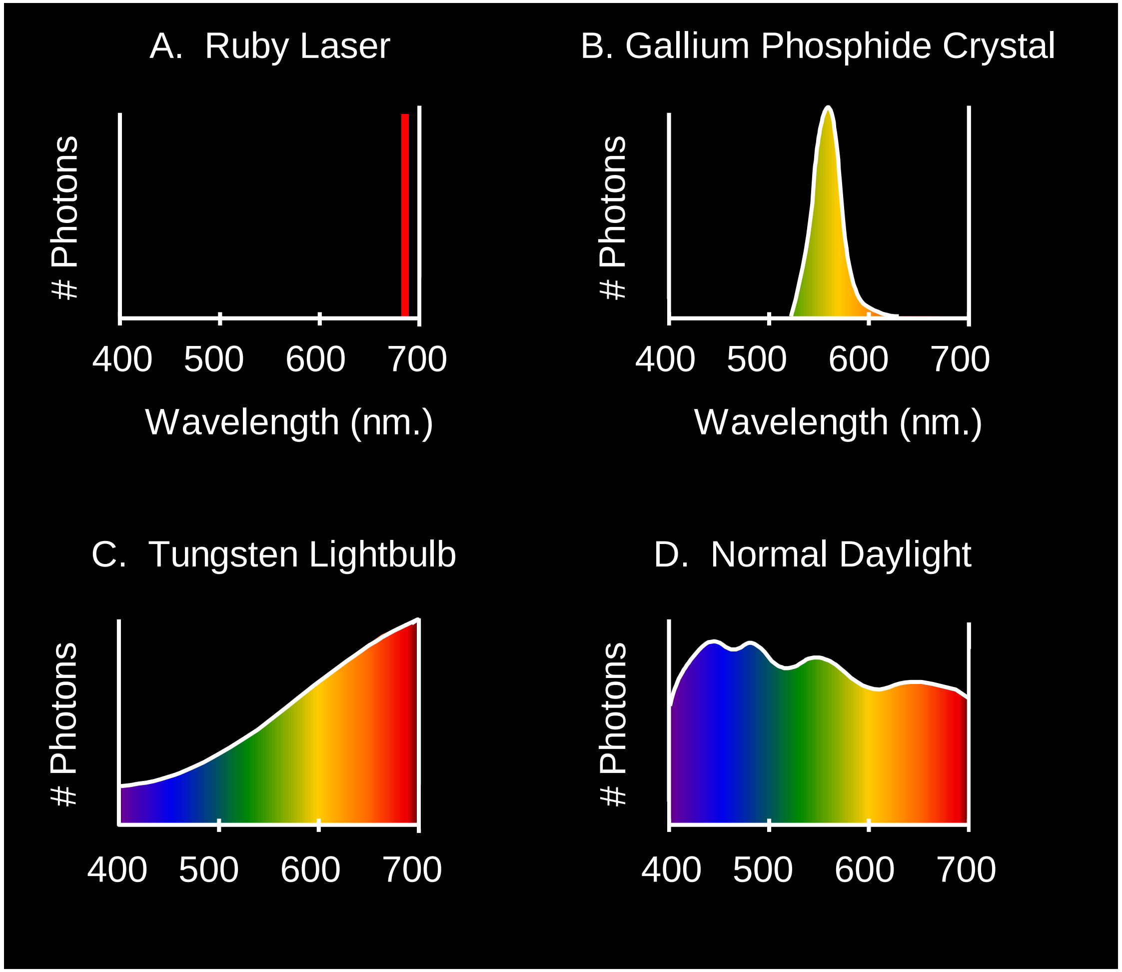


Fluorescent lamp

The Physics of Light



Some examples of the spectra of light sources

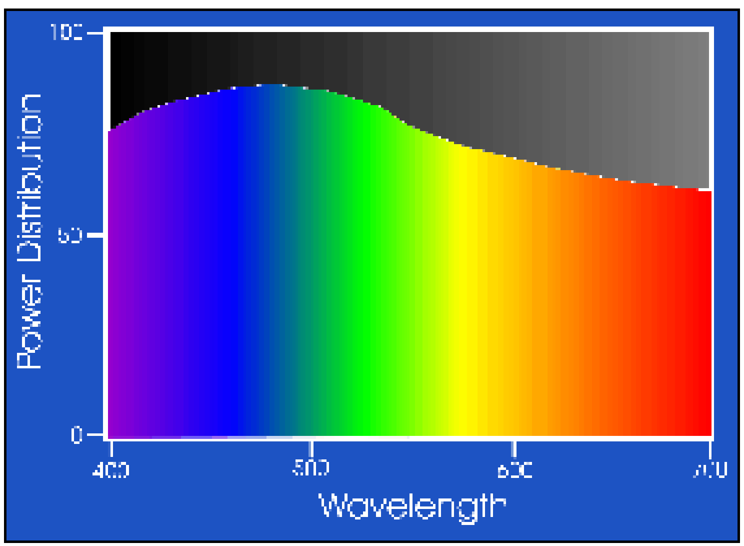


Spectral Power Distribution

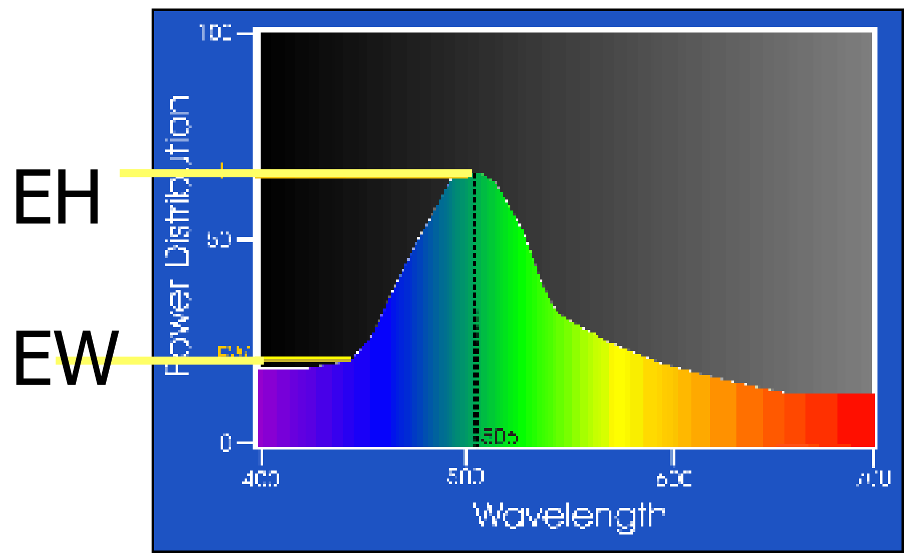
Hue: dominant wavelength of the SPD: EH

Saturation: purity of the colour: EH-EW

Intensity: brightness of the colour: EW



White light



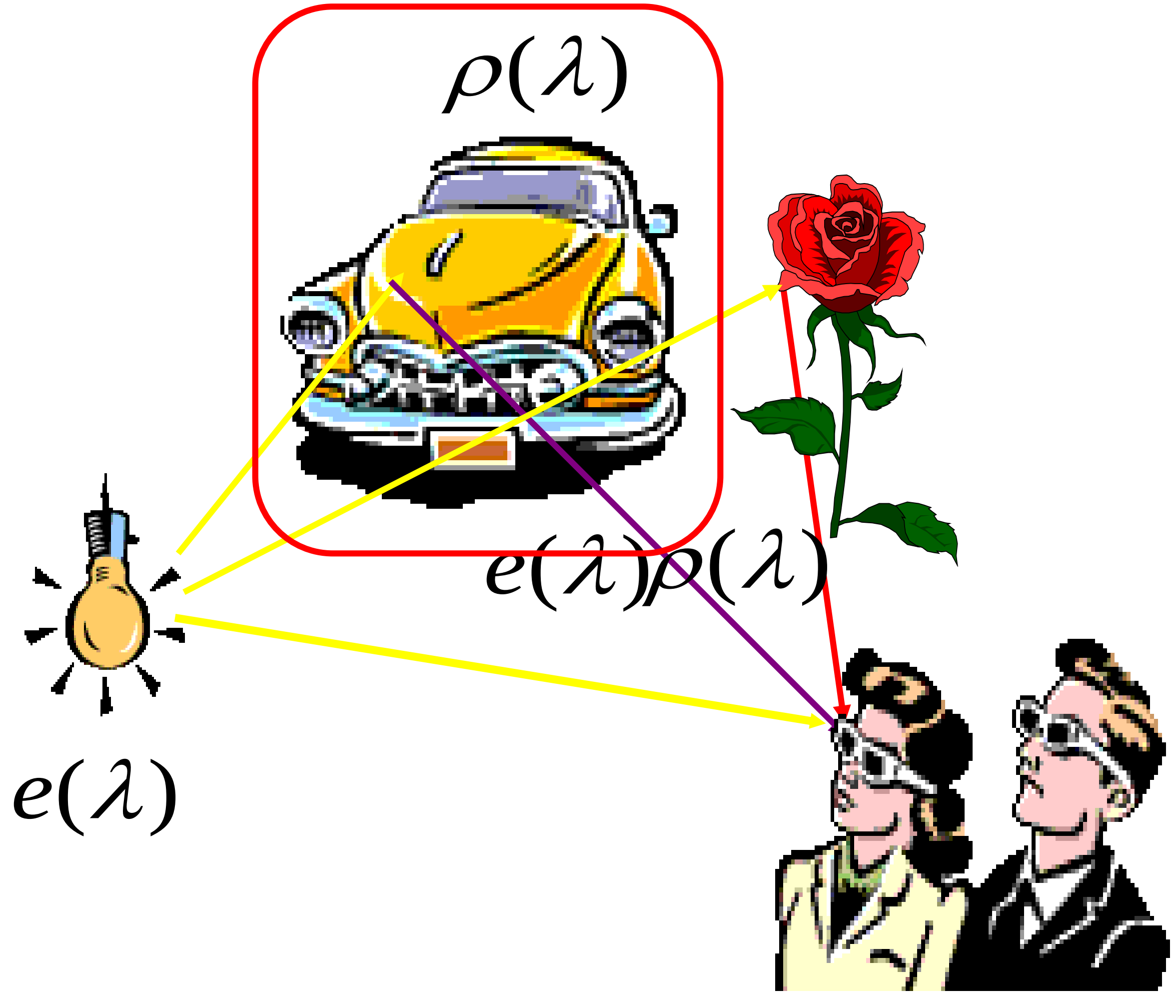
Green light

What makes an image?
the triplet light-objects-observer

Light source

Object(s)

Sensor



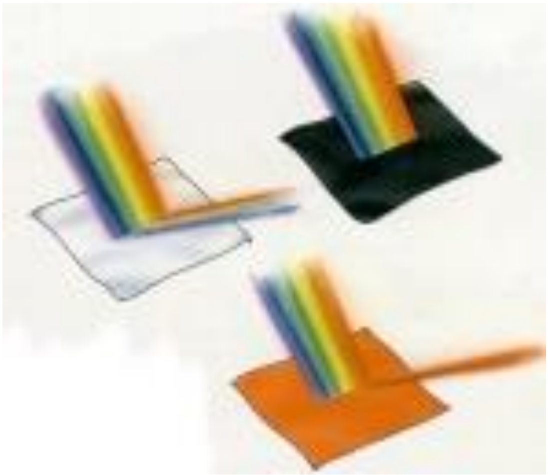
Object Colours

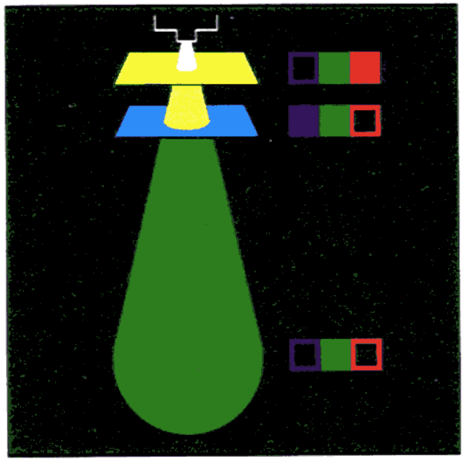
Materials:

- Transparent
- Opaque

Spectral Reflectance

$$\rho(\lambda)$$



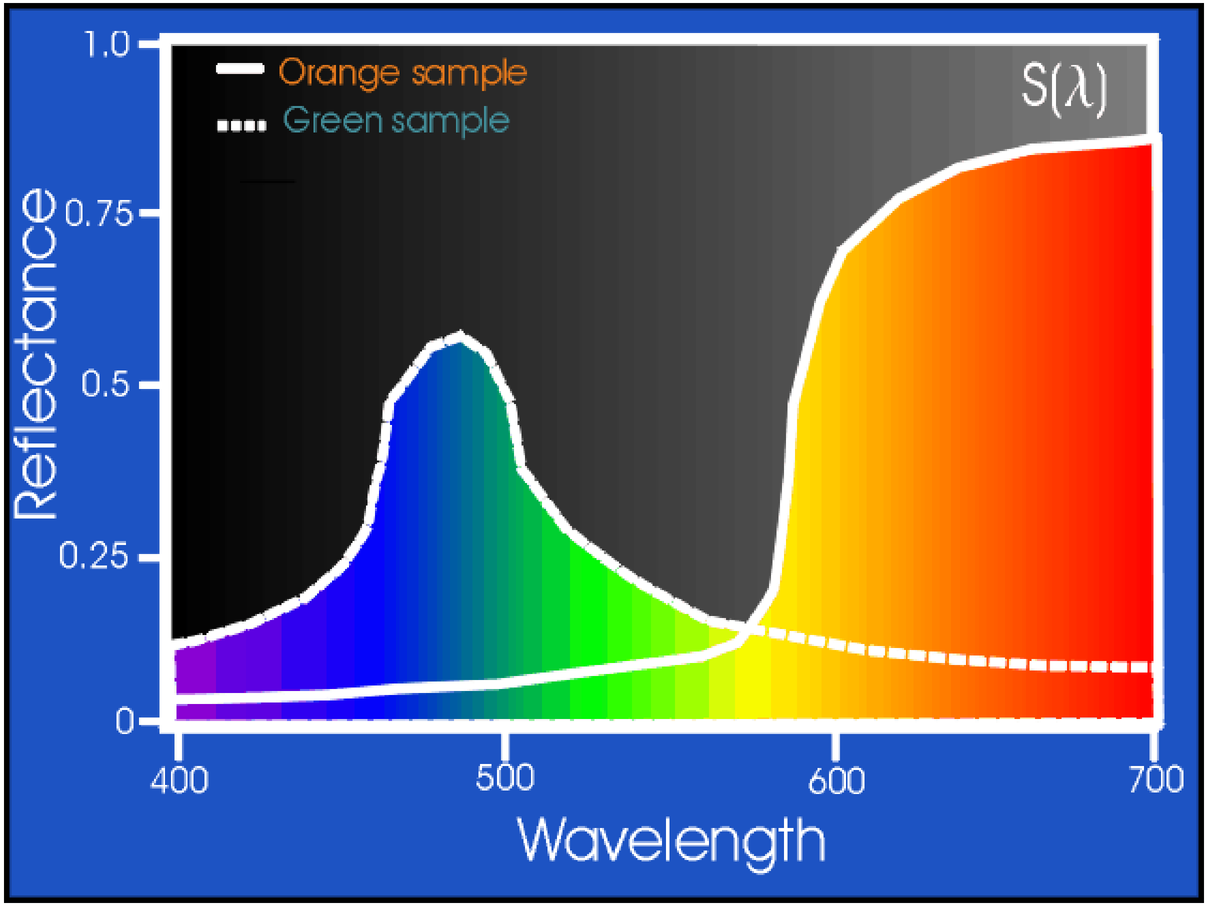


Object Colours

Material

spectrophotometer

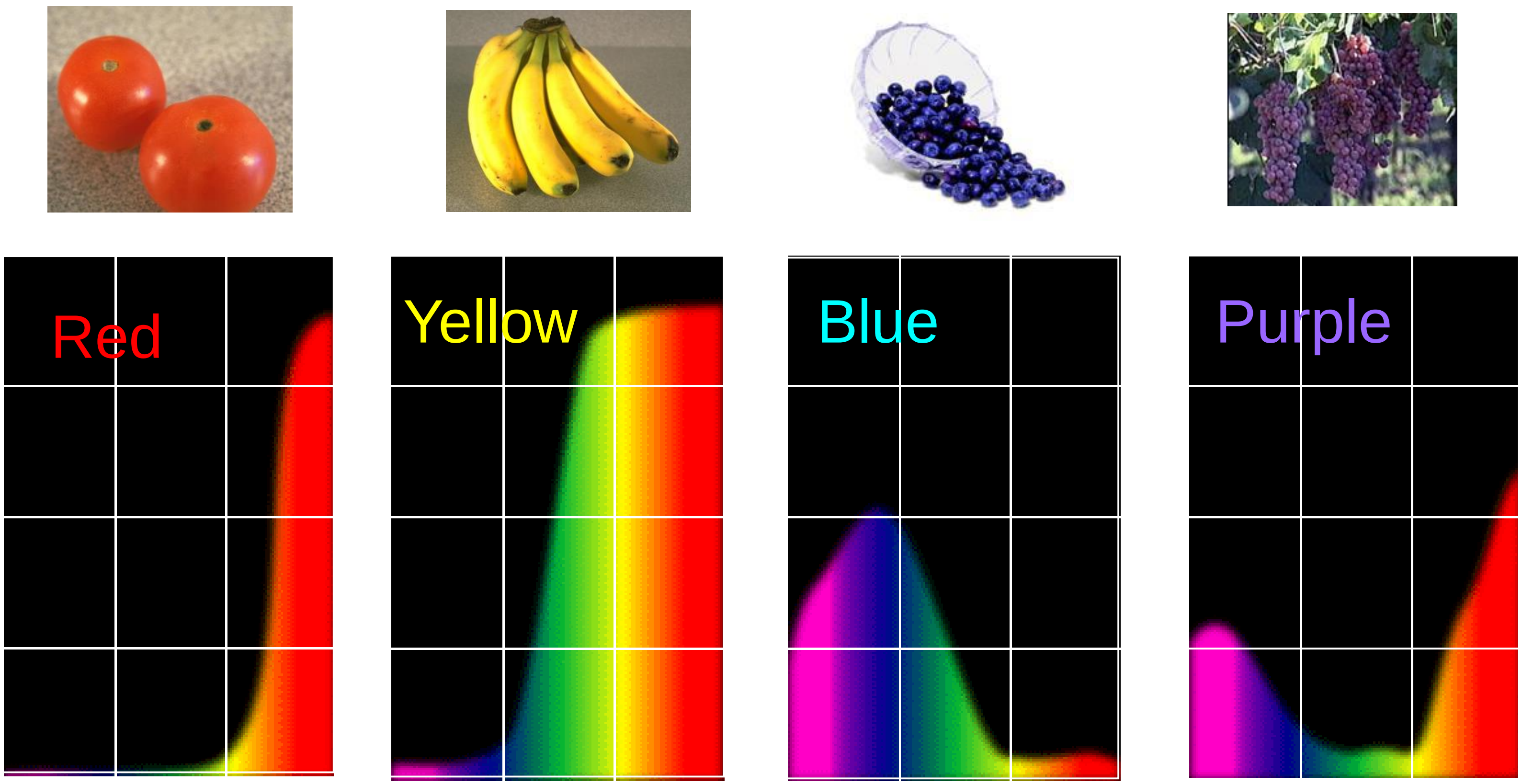
Reflectance curve



Object Colours

Some examples of the reflectance spectra of surfaces

% Photons Reflected



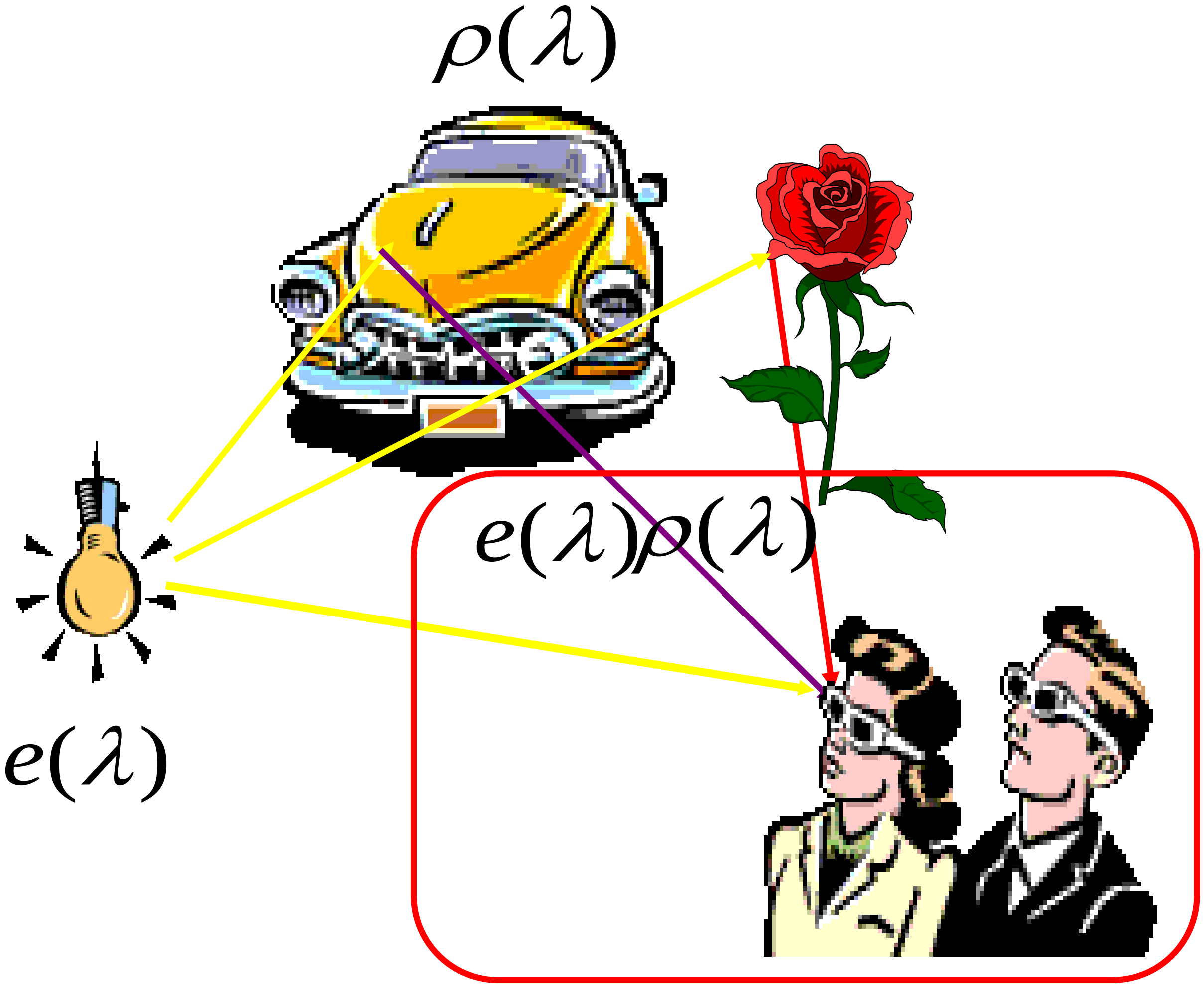
Wavelength (nm)

What makes an image?
the triplet light-objects-observer

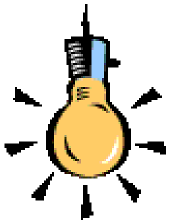
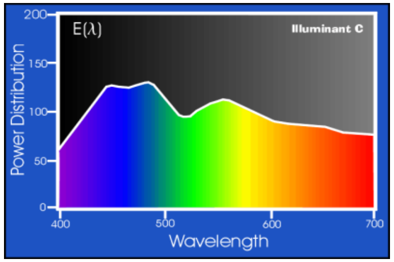
Light source

Object(s)

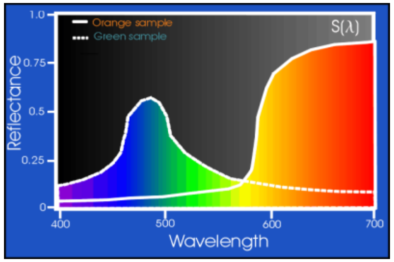
Sensor



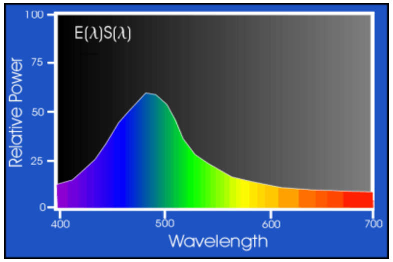
Light, Object and Sensor

Light source  

$$e(\lambda)$$

Object  

$$\rho(\lambda)$$

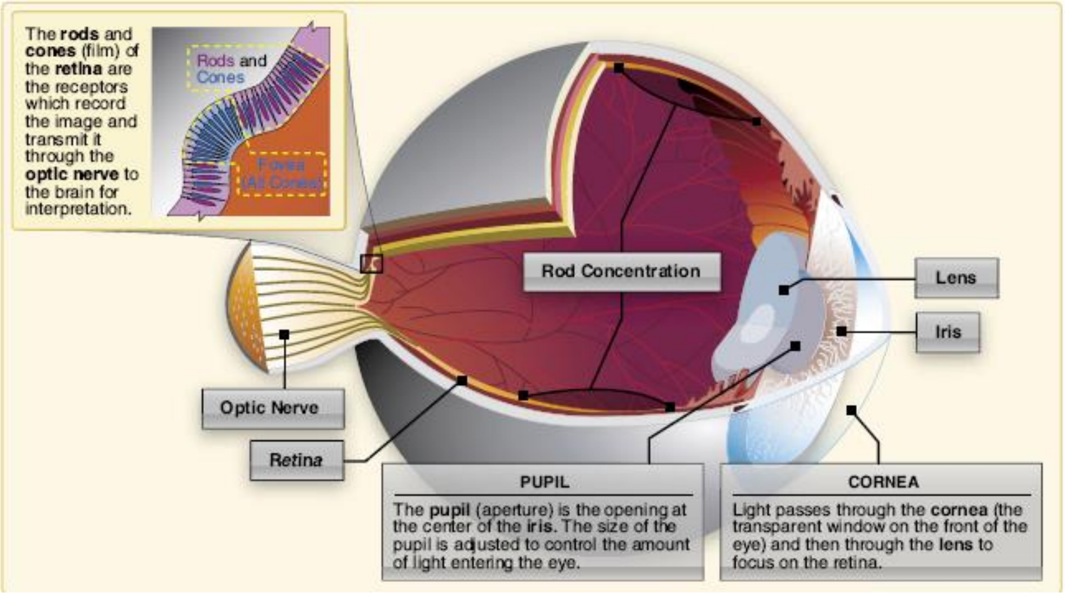
Sensor  

$$f(\lambda)$$

$$R = \int_\lambda e(\lambda)\rho(\lambda) f_R(\lambda) d\lambda, \quad G = \int_\lambda e(\lambda)\rho(\lambda) f_G(\lambda) d\lambda, \quad B = \int_\lambda e(\lambda)\rho(\lambda) f_B(\lambda) d\lambda$$

Human Eye

- Retina contains light sensitive cells that convert light energy into electrical impulses that travel through nerves to the brain.
- Brain interprets the electrical signals to form images.



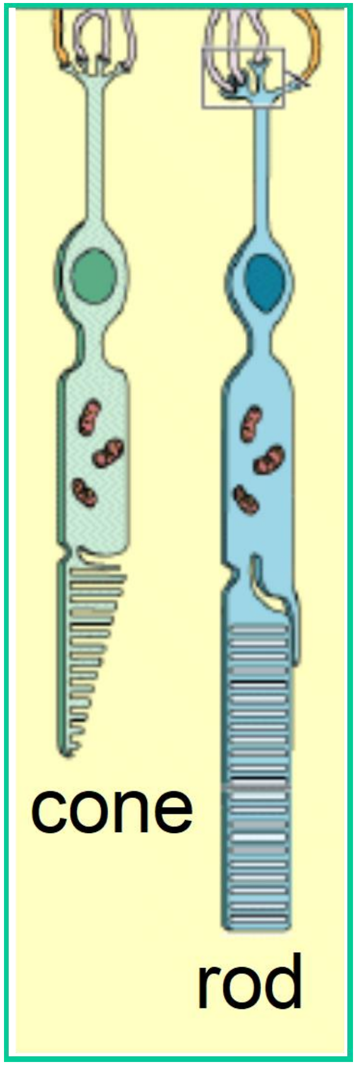
Two Types of Light-Sensitive Receptors

Cones

cone-shaped
less sensitive
operate in high light
color vision

Rods

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



Slide source: James Hays

Light Detection: Rods and Cones

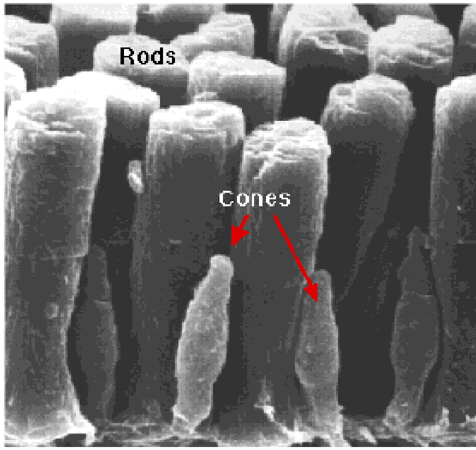
Rods:

-120 million rods in retina

-1000X more light sensitive than Cones

- Discriminate B/W brightness in low illumination

- Short wave-length sensitive



Cons:

- 6-7 million cones in the retina

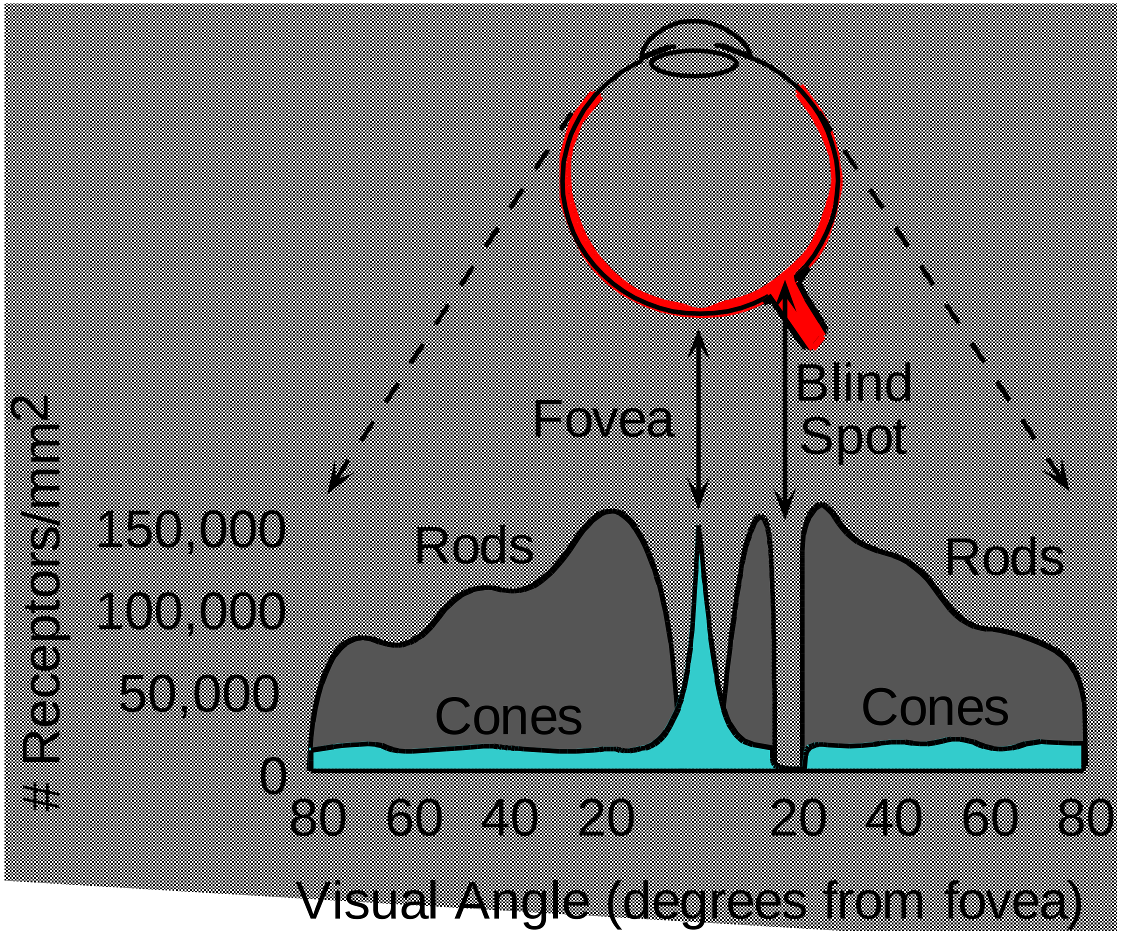
- Responsible for high-resolution vision

- Discriminate Colors

- Three types of color sensors: 64% red, 32% green, 2% blue)

- Sensitive to any combination of three colors

Distribution of Rods and Cones

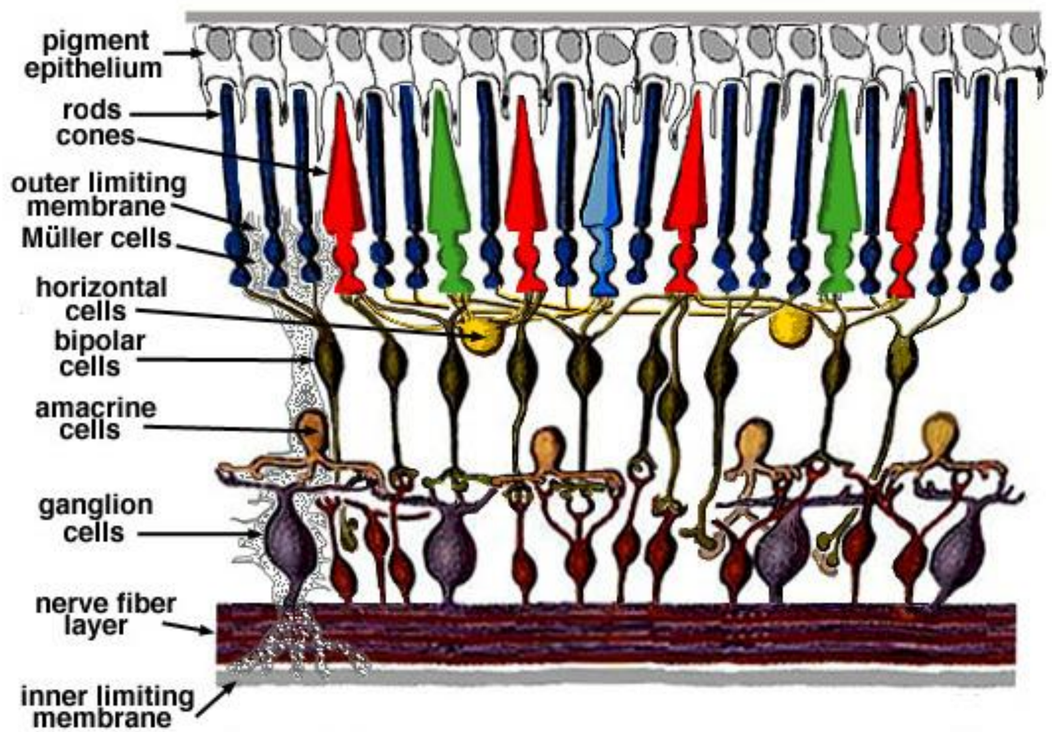


Night Sky: why are there more stars off-center?
Averted vision: http://en.wikipedia.org/wiki/Averted_vision

Slide source: James Hays

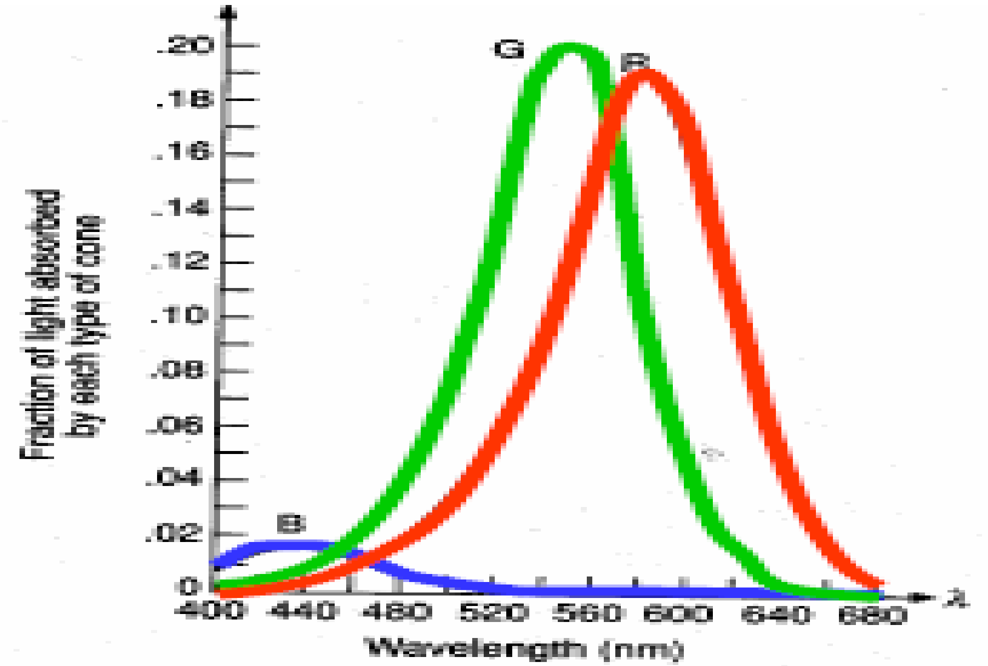
The Retina

- 0.5 mm thick
- The photosensors (the rods and cones) lie outermost in the retina.
- Interneurons
- Ganglion cells (the output neurons of the retina) lie innermost in the retina closest to the lens and front of the eye.



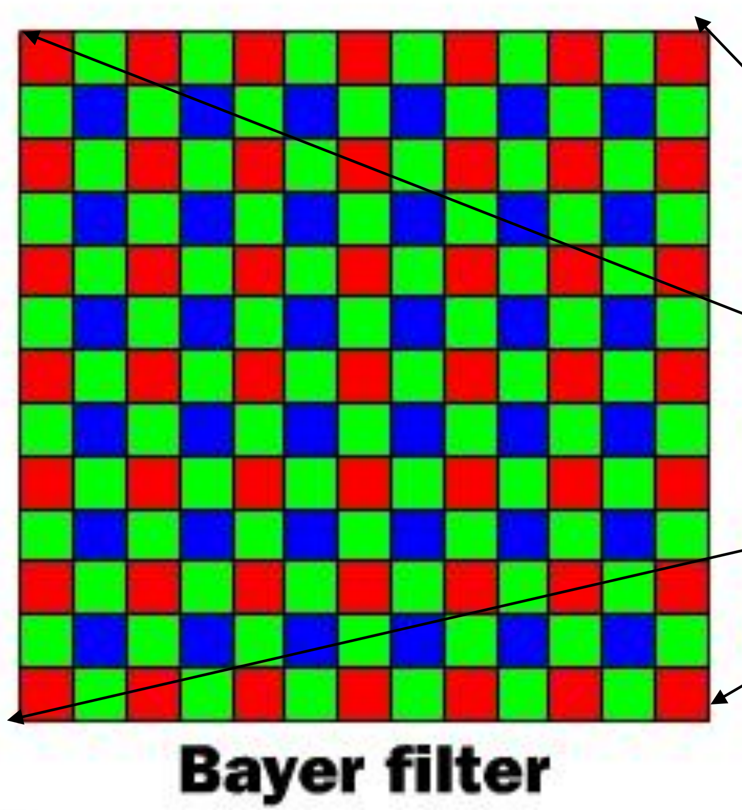
Tristimulus of Color Theory

Spectral-response functions of each of the three types of cones

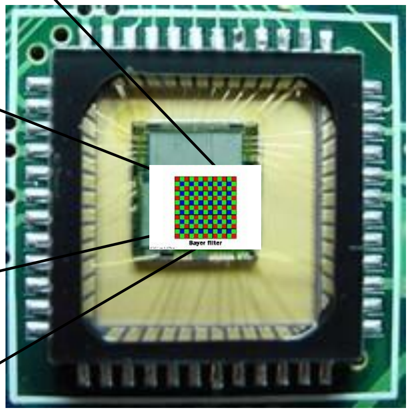


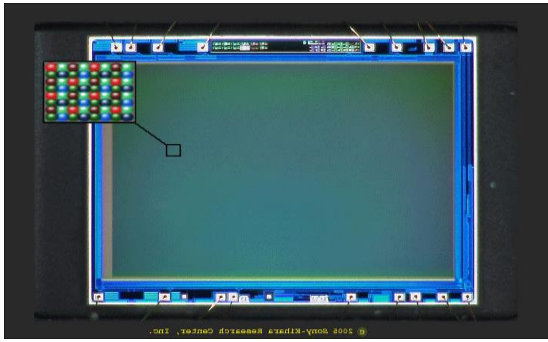
How to form colour images?

Bayer filter:
Green fills in half of the checker board and Red and Blue fill the rest.



© 2000 How Stuff Works





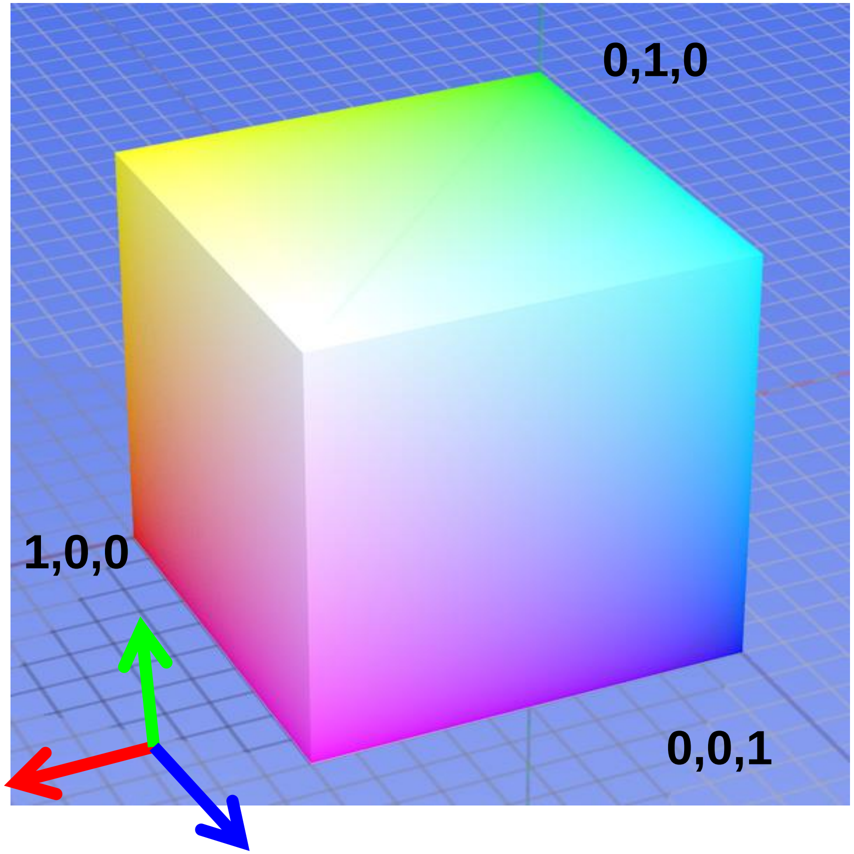
Today's class: Image Formation

1. Projective Geometry and Camera Models
2. <u>Light and Color Models</u>
    - 2.1 Digital Image Presentation
    - 2.2 Light, Object and Sensor
    - 2.3 <u>Color Systems</u>
    - 2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

RGB space

Default color space

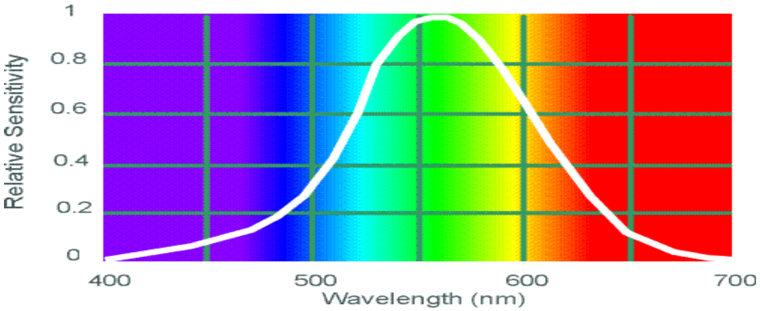


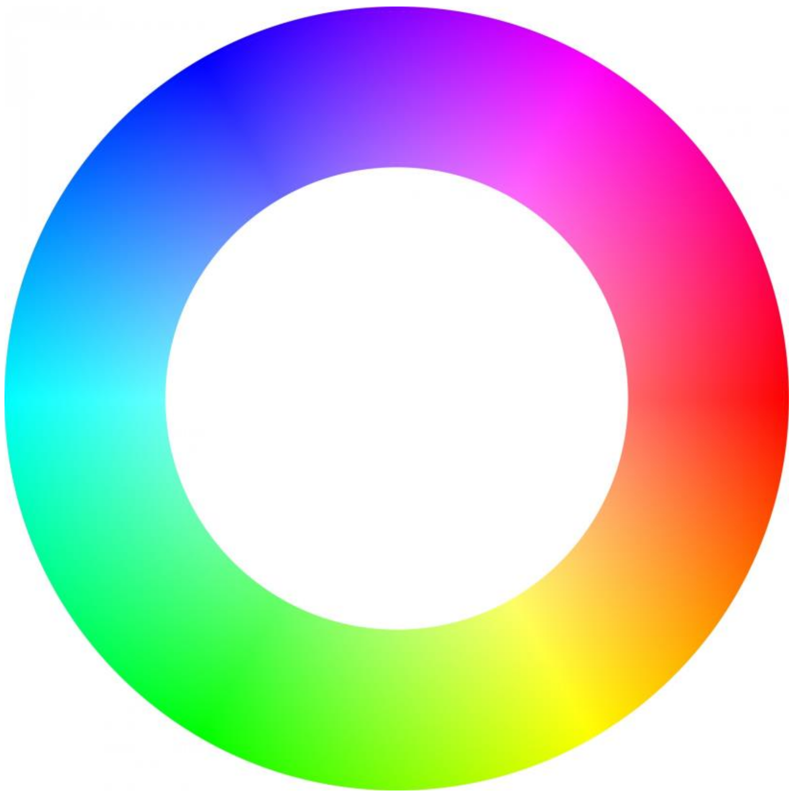
Some drawbacks

- Strongly correlated channels
- Not perceptually meaningful.

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Color and hue





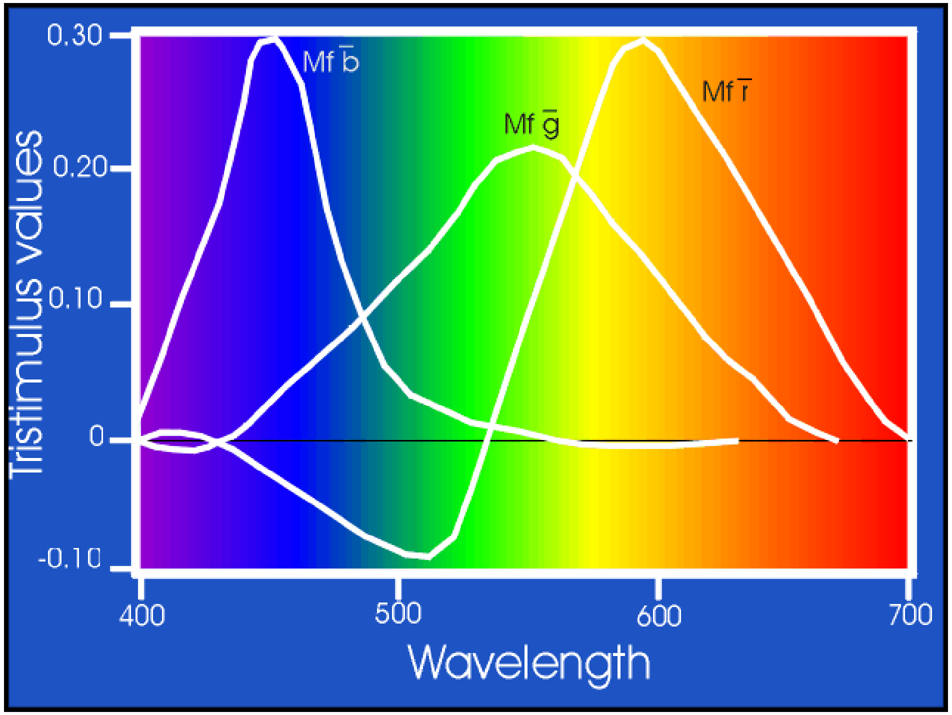
Observer: Trichromacy

Young-Helmholtz approach

Tristimulus values R, G, and B

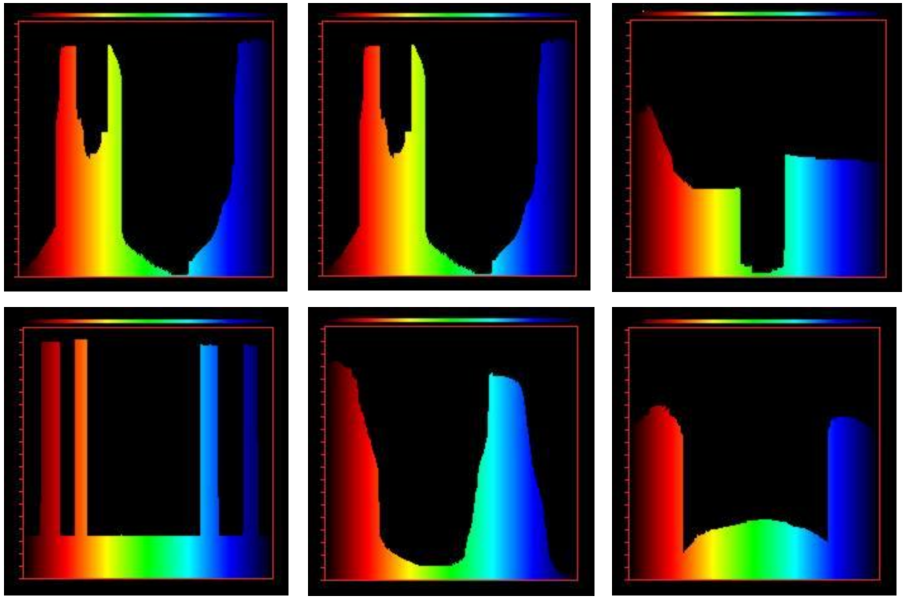
Wright (7) Guild (10)

Stiles and Burch (50)



Spectral Energy Distribution

The six spectra below look the same purple to normal color-vision people

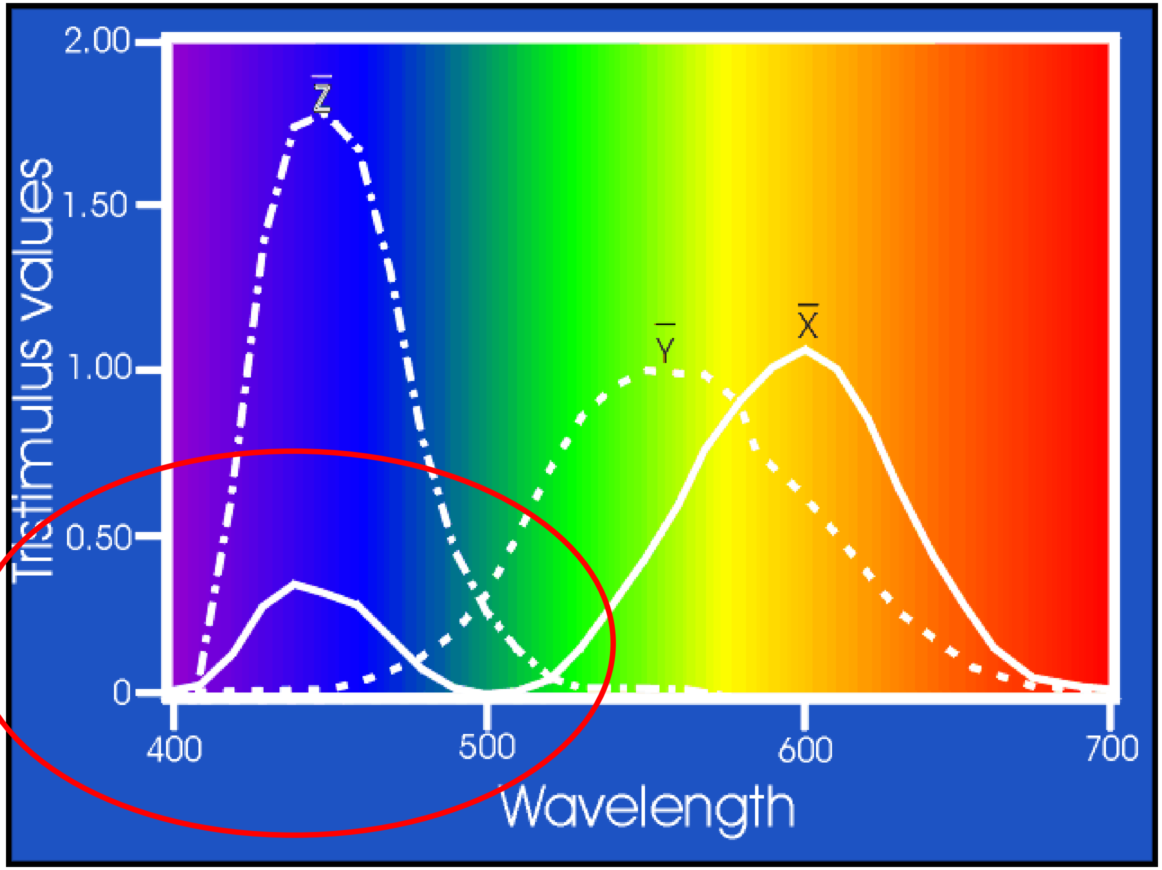


Colorimetry: CIE XYZ-system

$$X = \int_{\lambda} e(\lambda)\rho(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = \int_{\lambda} e(\lambda)\rho(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = \int_{\lambda} e(\lambda)\rho(\lambda)\bar{z}(\lambda)d\lambda$$

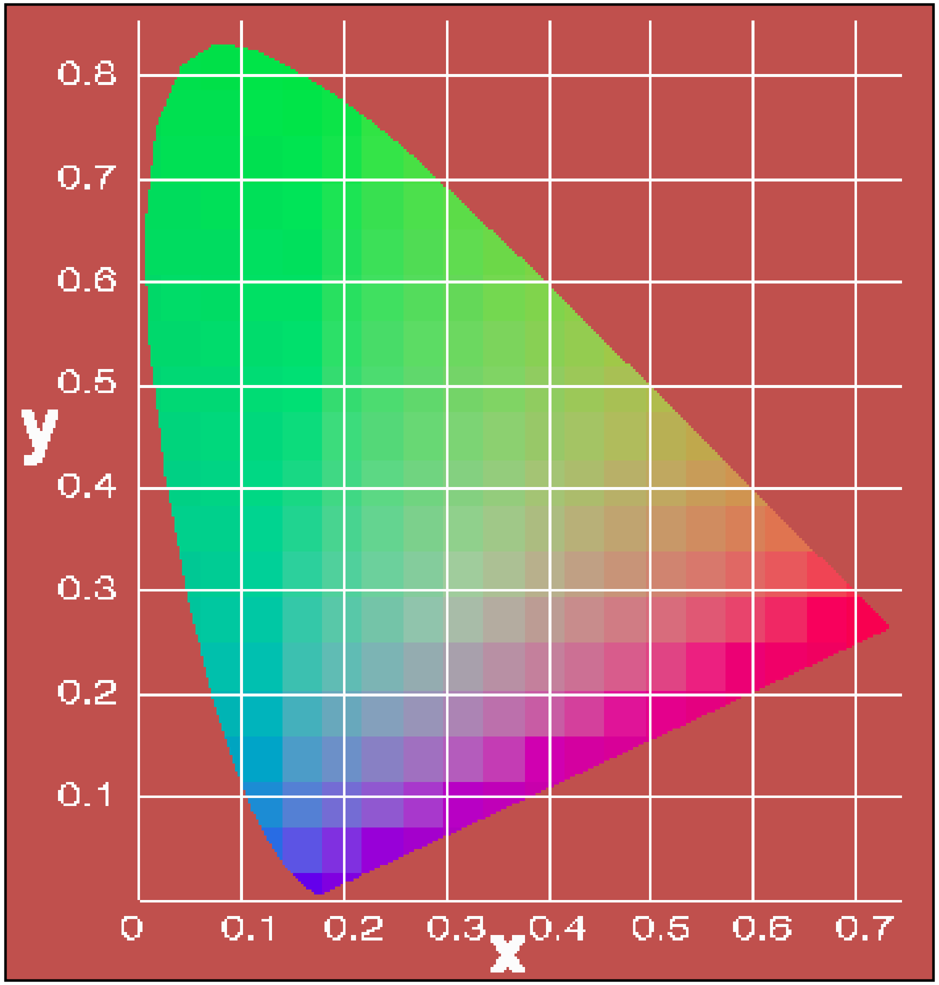


Colorimetry: CIE xy-system

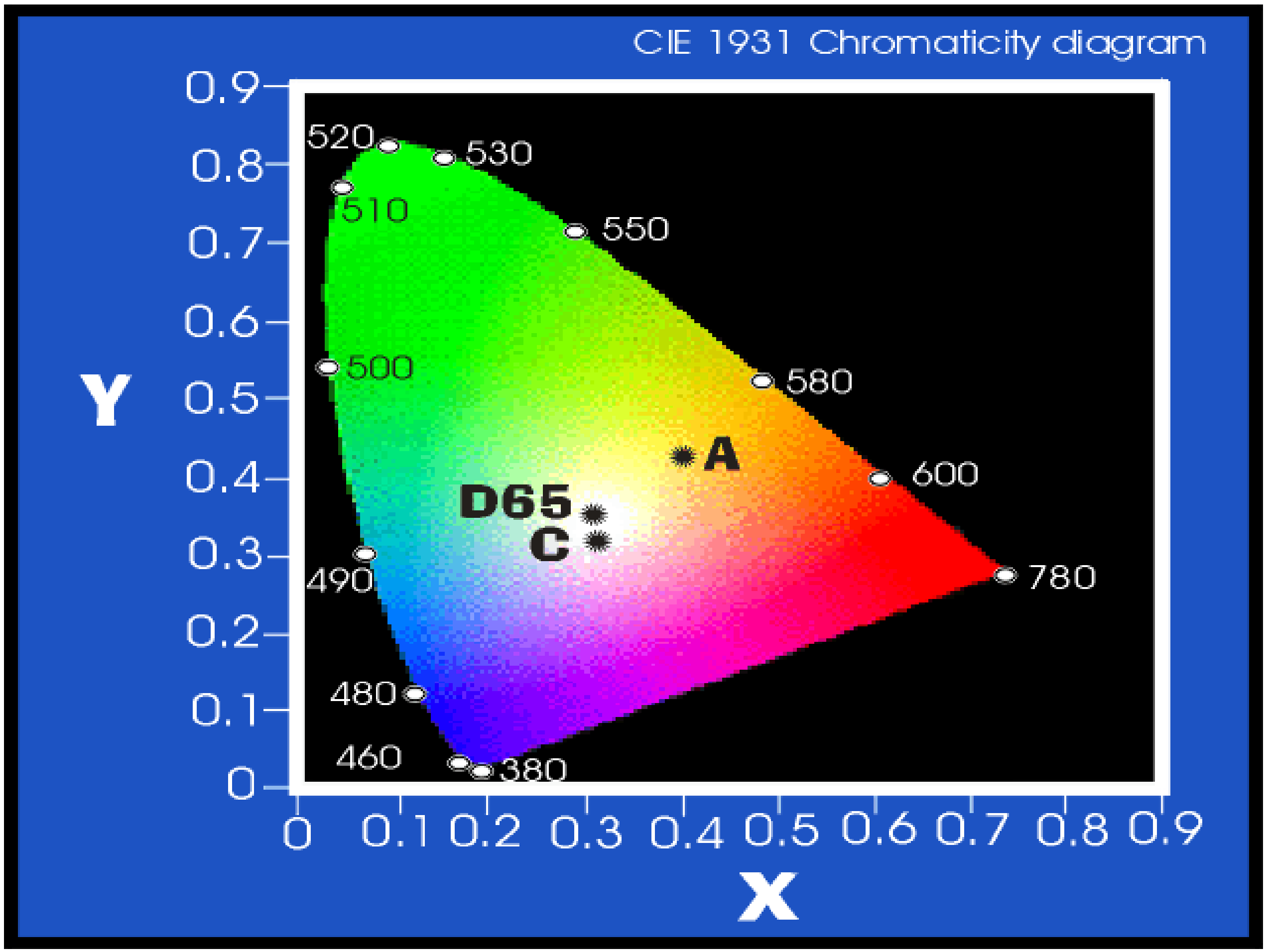
$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$



Colorimetry: Illuminants in the xy-plane



Light Sources and Illuminants

Light sources:

sun, candle,

fluorescent lamp,

incandescent lamp

Illuminants:

illuminant A

illuminant D65

illuminant C

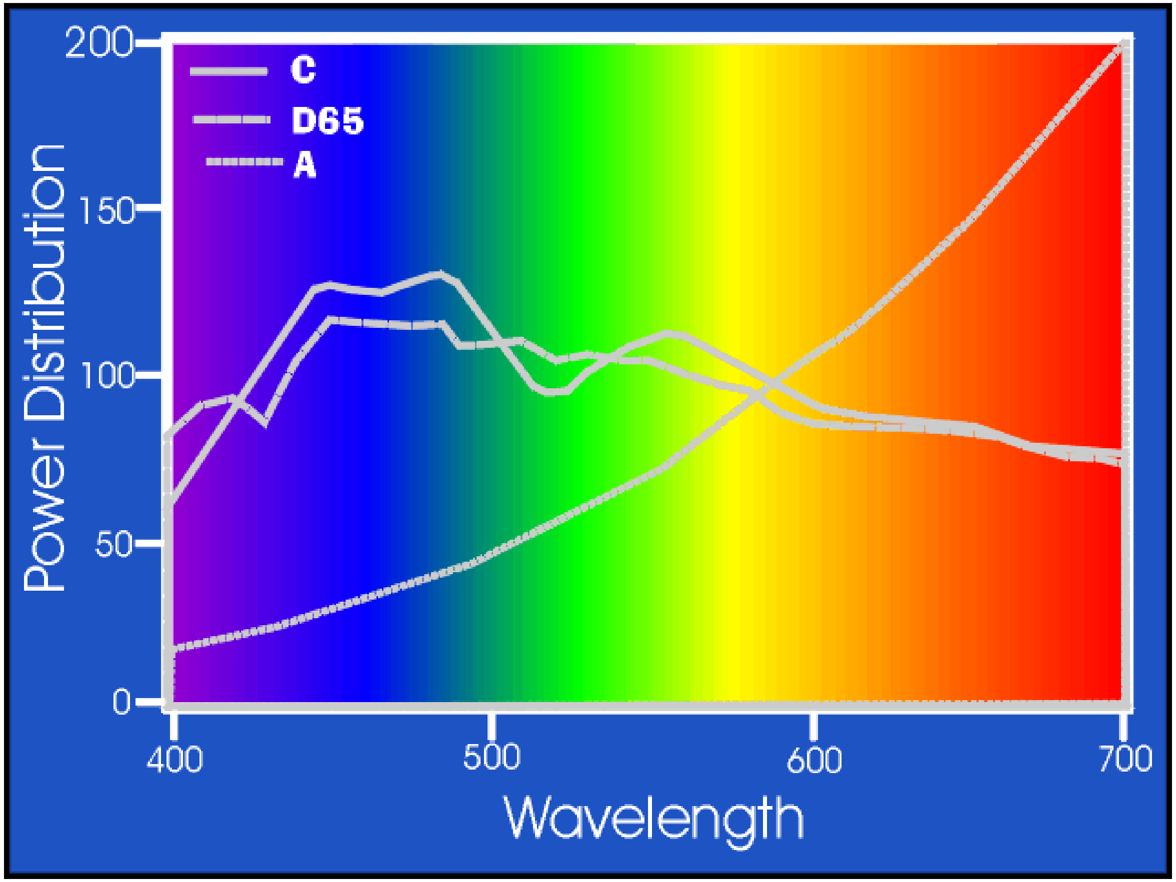
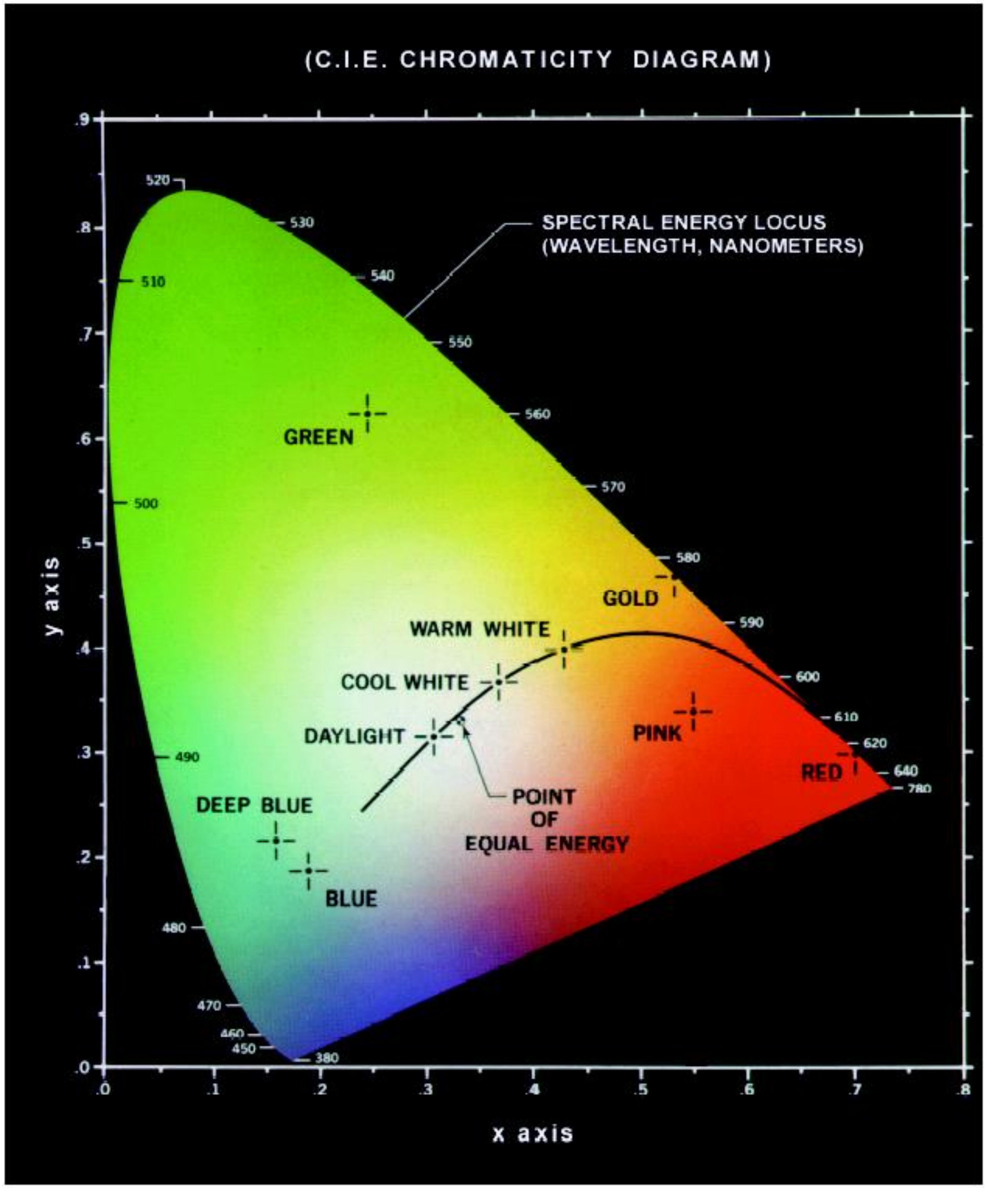
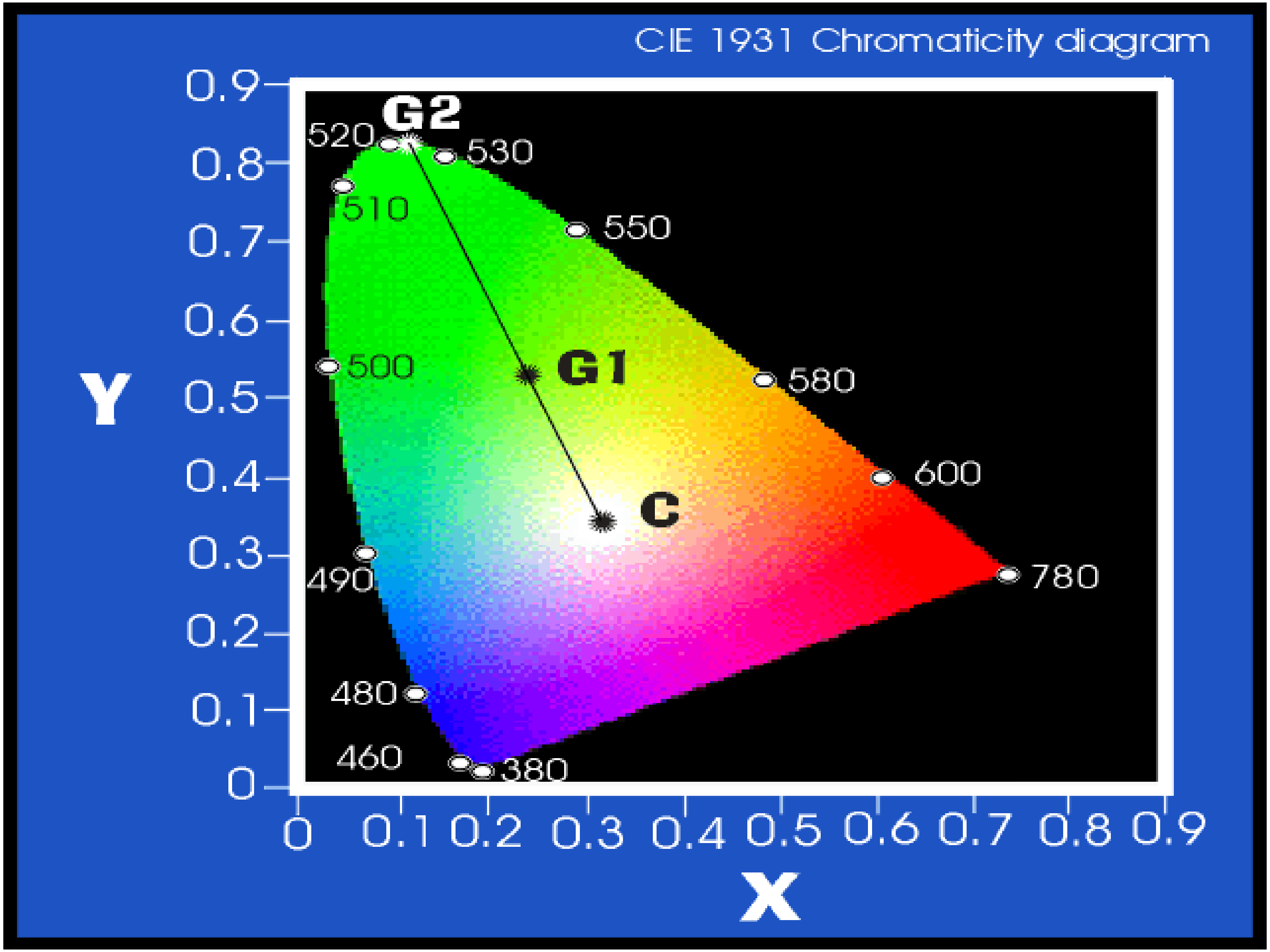


FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)



Colorimetry: HSI in the xy-plane

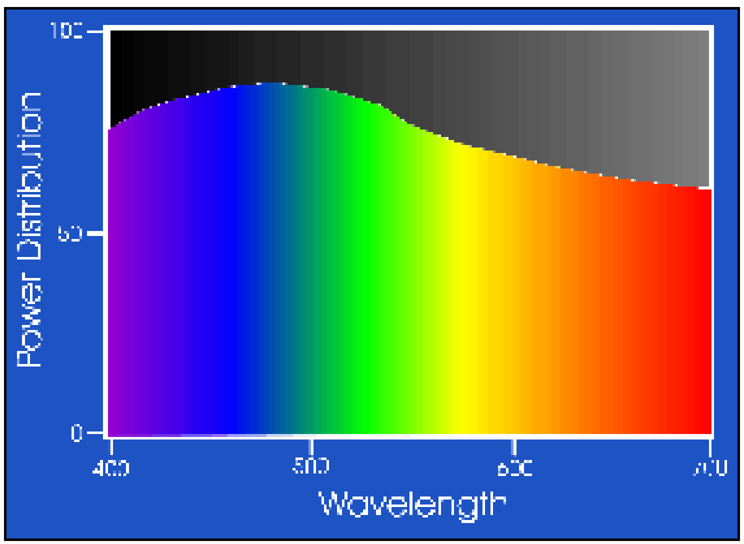


Spectral Power Distribution

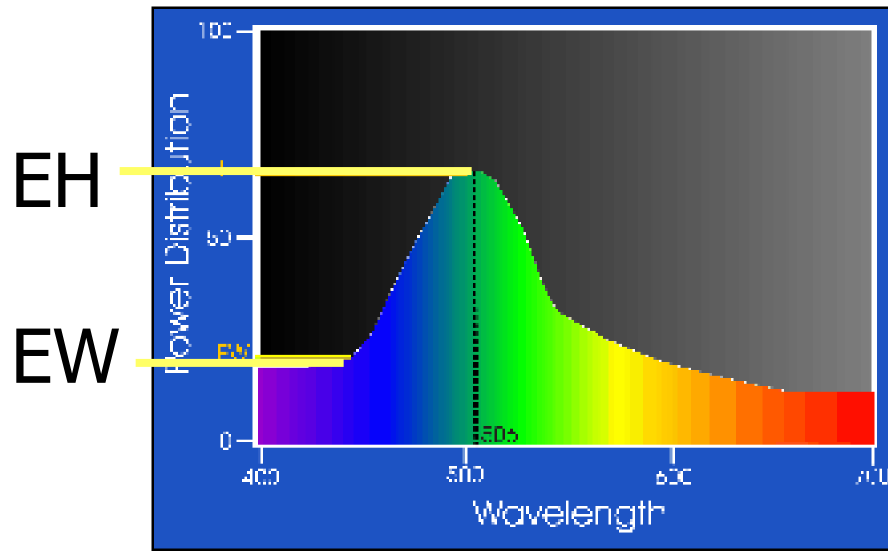
Hue: dominant wavelength of the SPD: EH

Saturation: purity of the colour: EH-EW

Intensity: brightness of the colour: EW



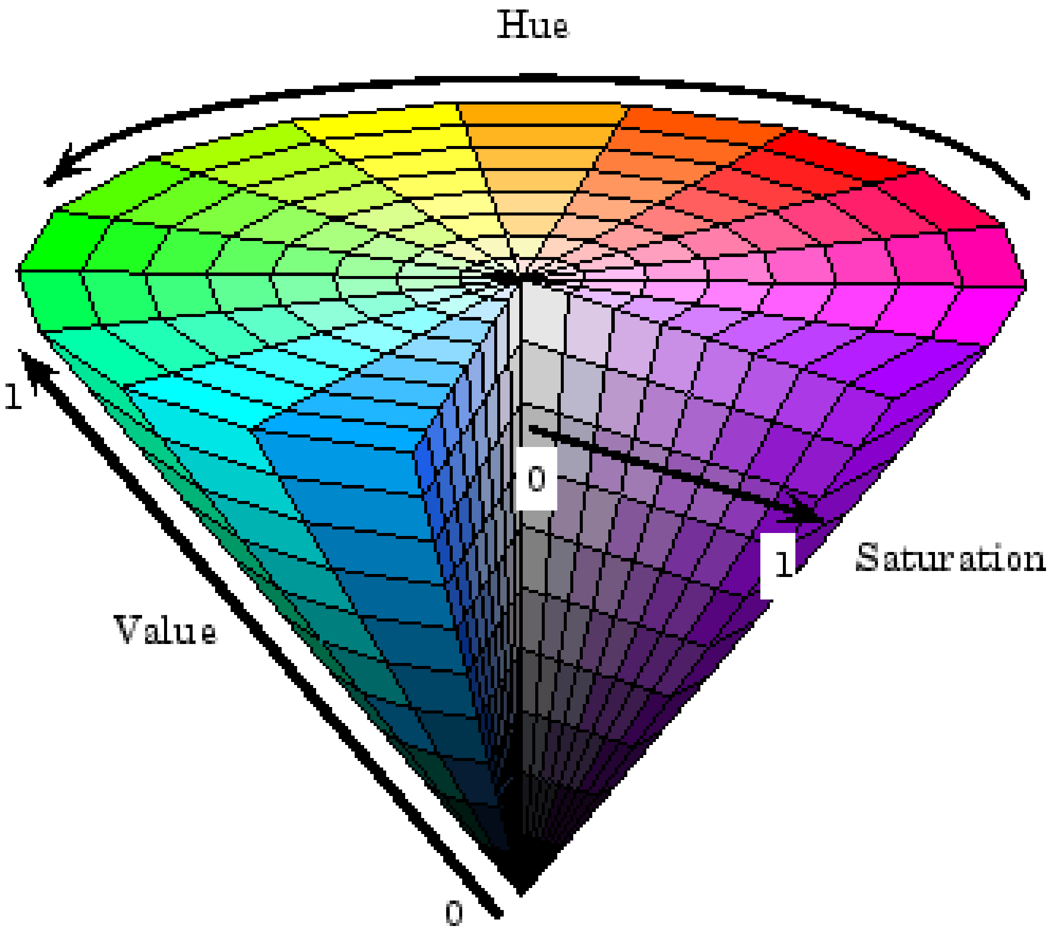
White light

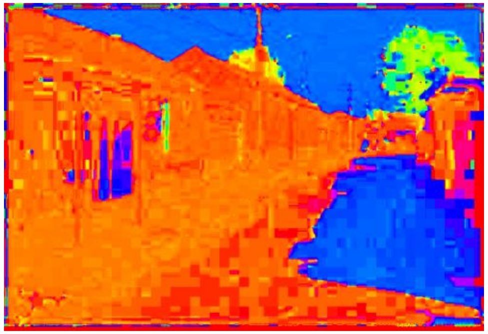


Green light

Color Spaces: HSV

Intuitive color space



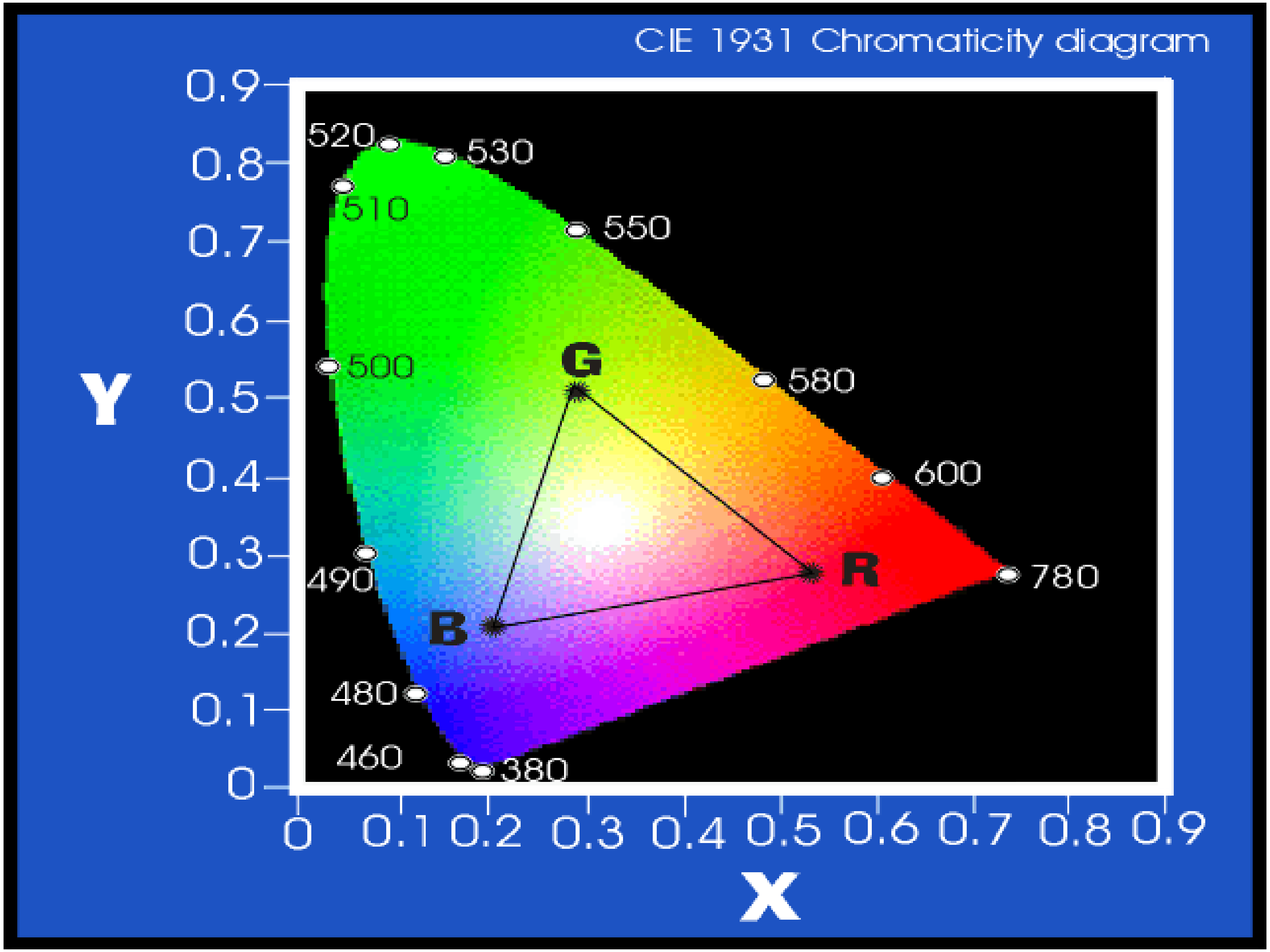


H
(S=1,V=1)

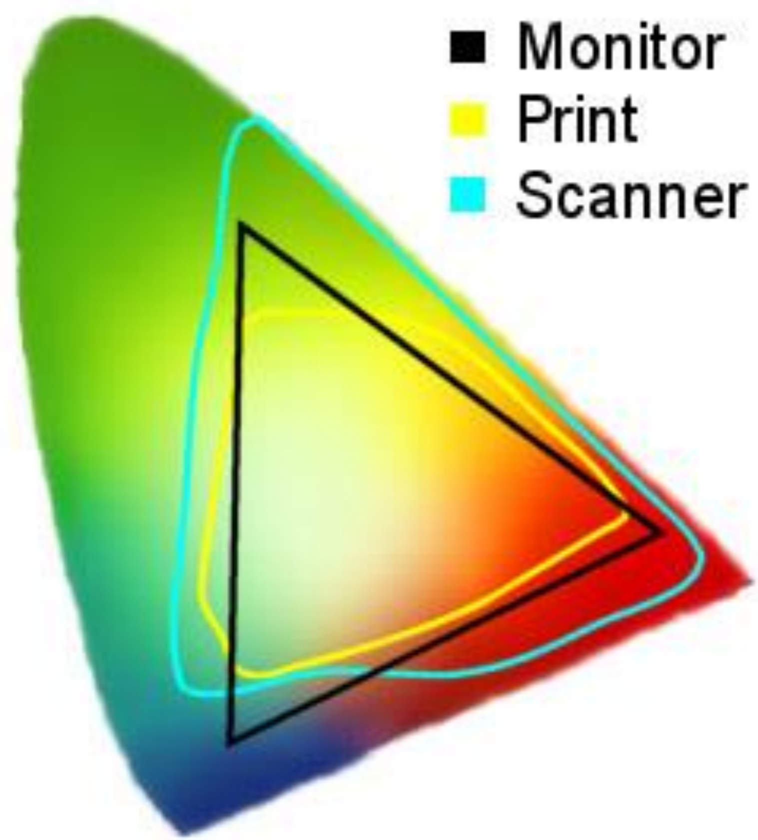
S
(H=1,V=1)

V
(H=1,S=0)

Colour Gamuts in the xy-plane



Monitor/Print/Scanner Gamut



Today's class: Image Formation

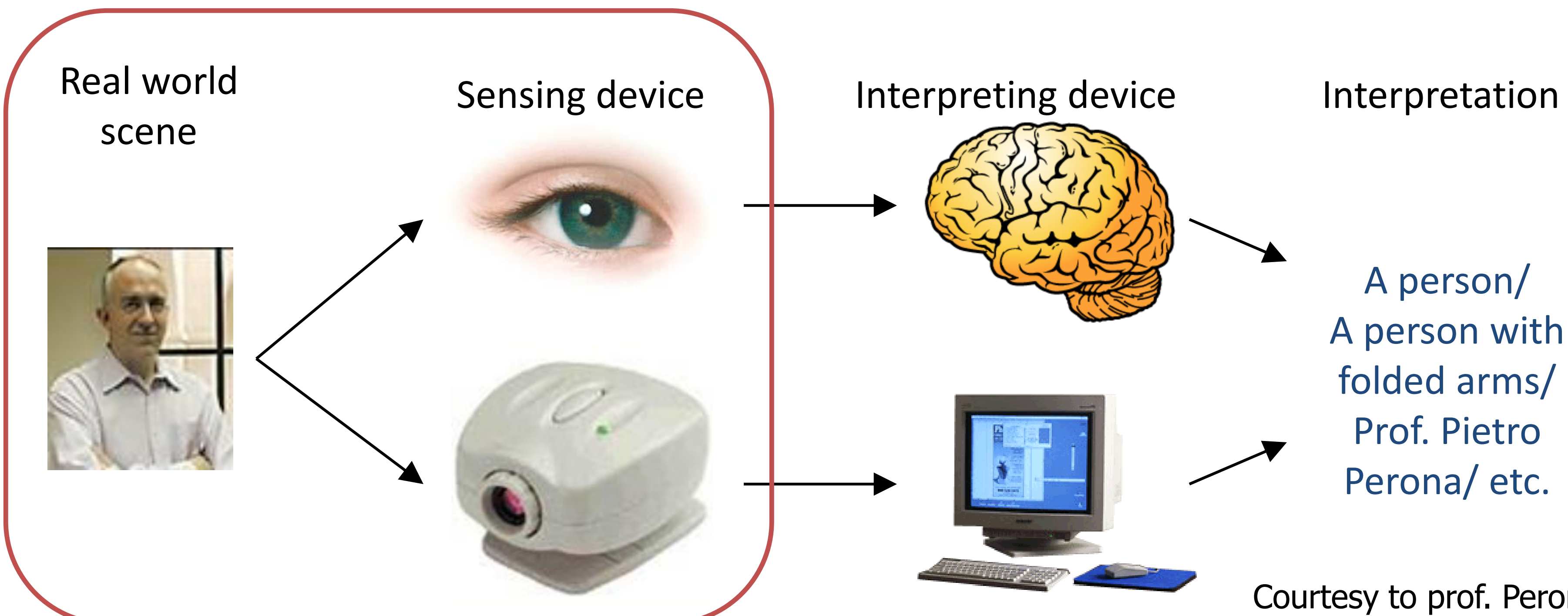
1. Projective Geometry and Camera Models
2. Light and Color Models
    2.1 Digital Image Presentation
    2.2 Light, Object and Sensor
    2.3 Color Systems
    2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

The computer vision problem

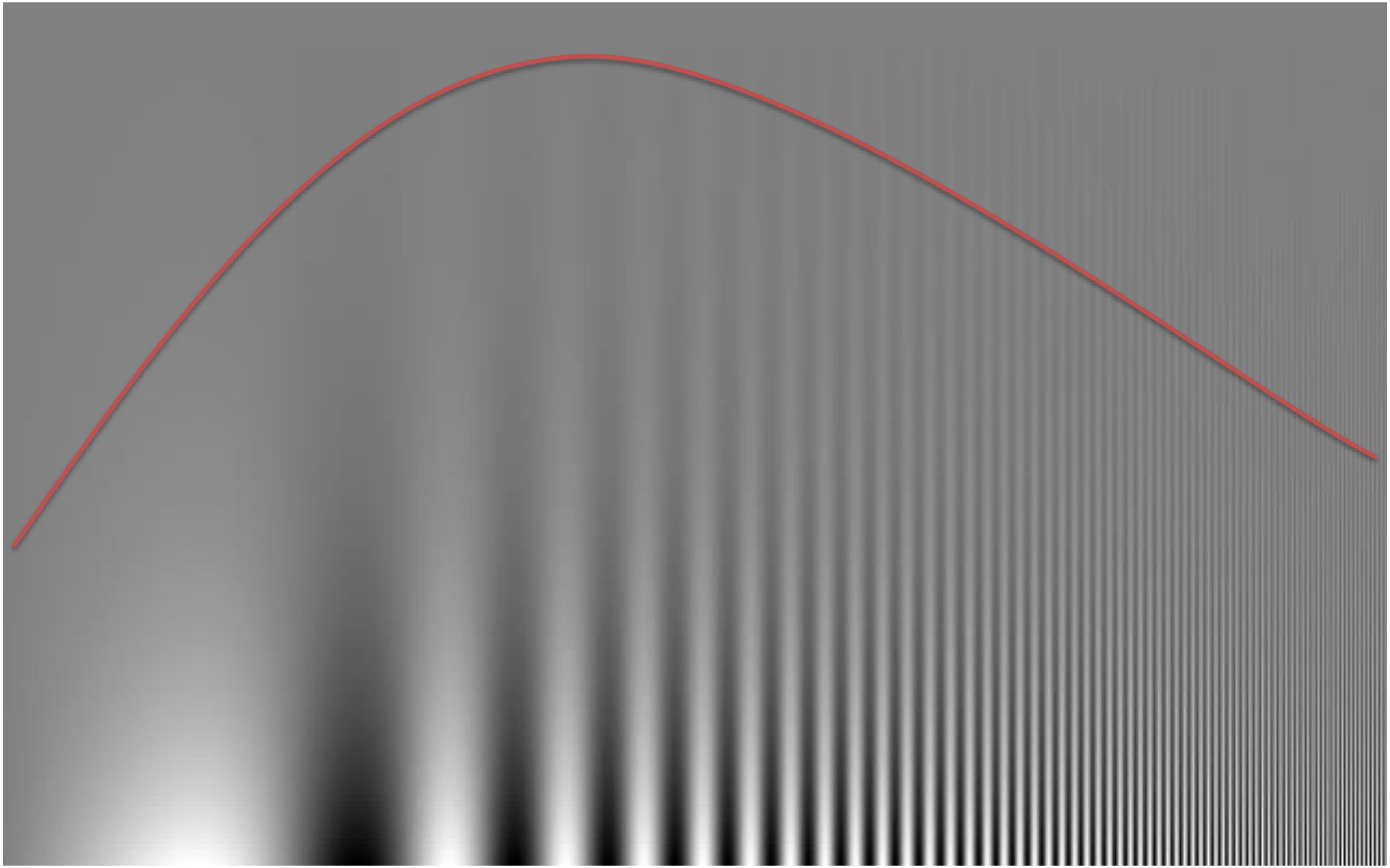
- Make a computer to see and to understand images

- We know it is physically possible – we do it every day and effortlessly!



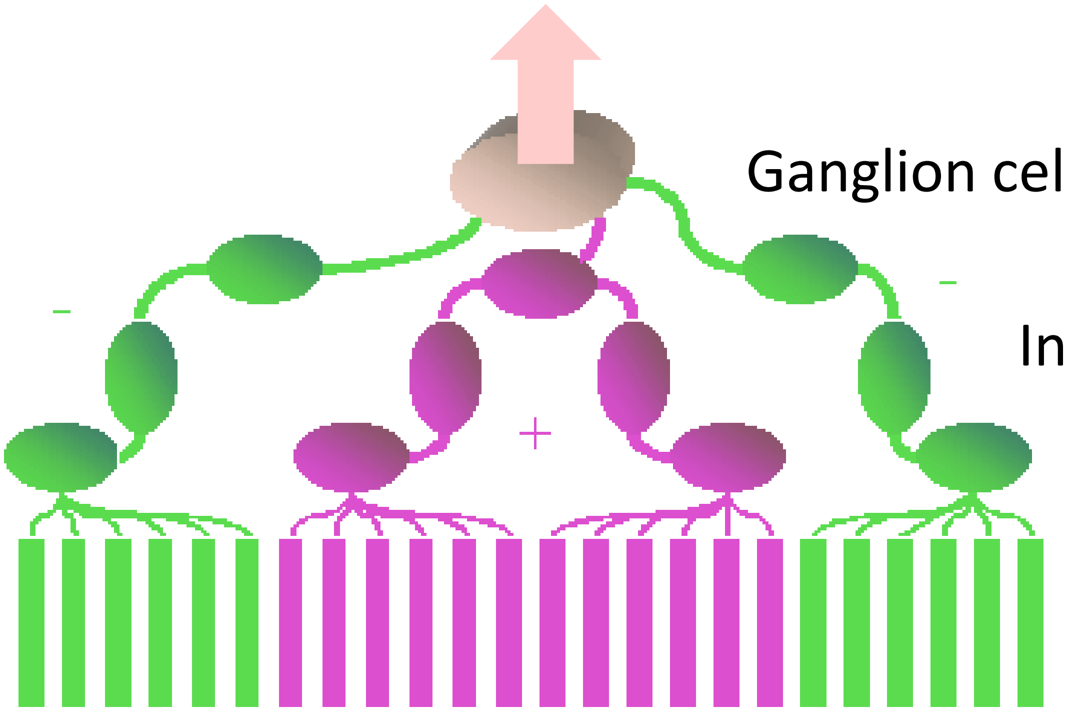
Courtesy to prof. Perona

Contrast Campbell-Gibson Curve



Receptive Fields

The ganglion cell produces some background response even when there is no light on its receptive field

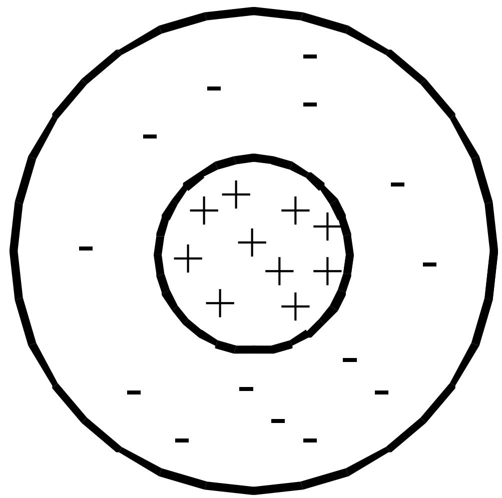


Ganglion cell

Intermediate neural cells

Photoreceptors:
Pink is excitatory
Green is inhibitory

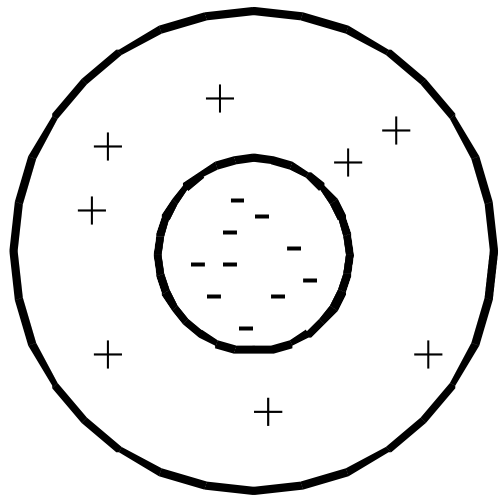
“On-Center” Ganglion Cell



Responds maximally to light increments in the center,

and light decrements in the surround.

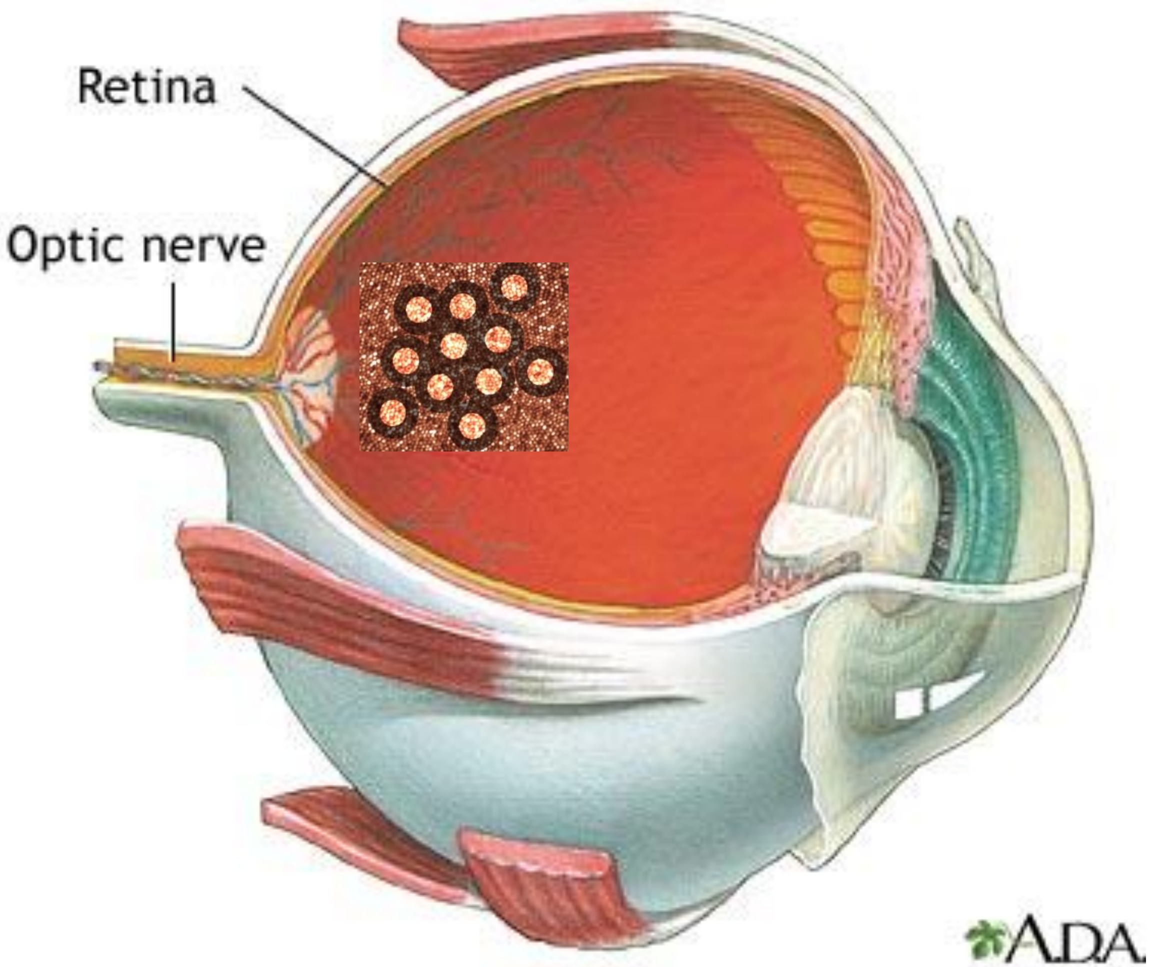
"Off-Center" Ganglion Cell



Responds maximally to light decrements in the center,

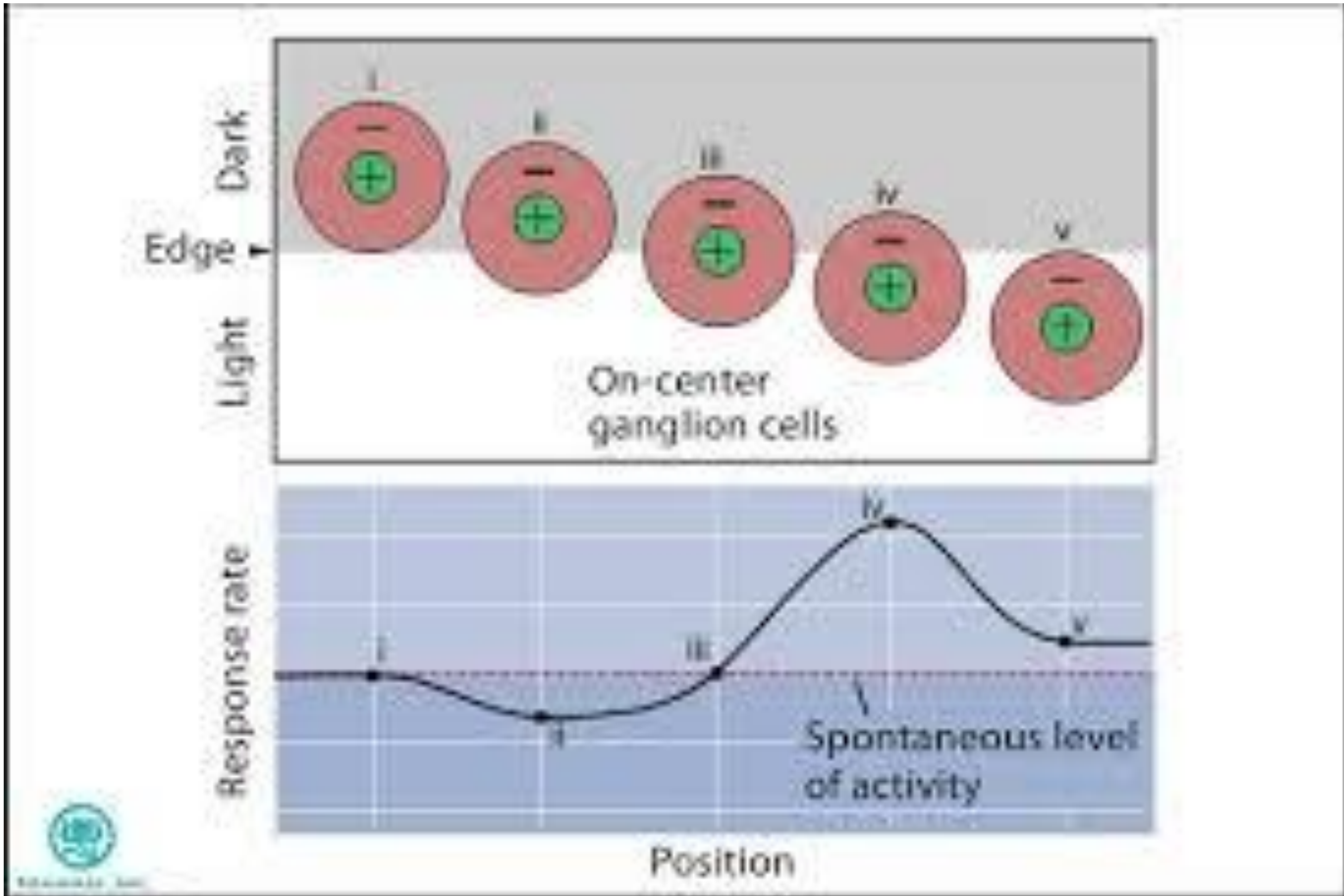
and light increments in the surround.

Receptive Fields on the Retina

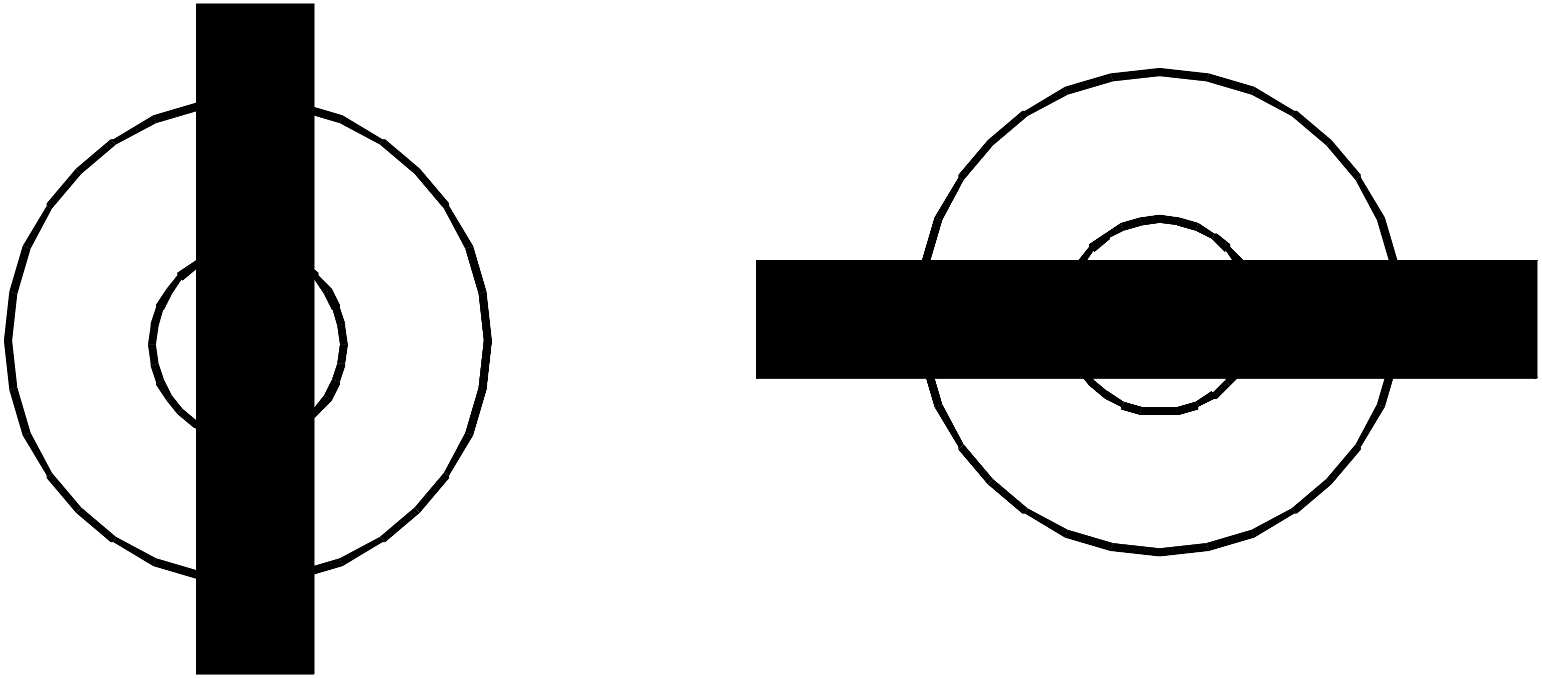


The size of the receptors and the receptive field are shown here much larger than actual size!

Responses: ON-Centre Ganglion Cells



Ganglion cells have no orientation preference.



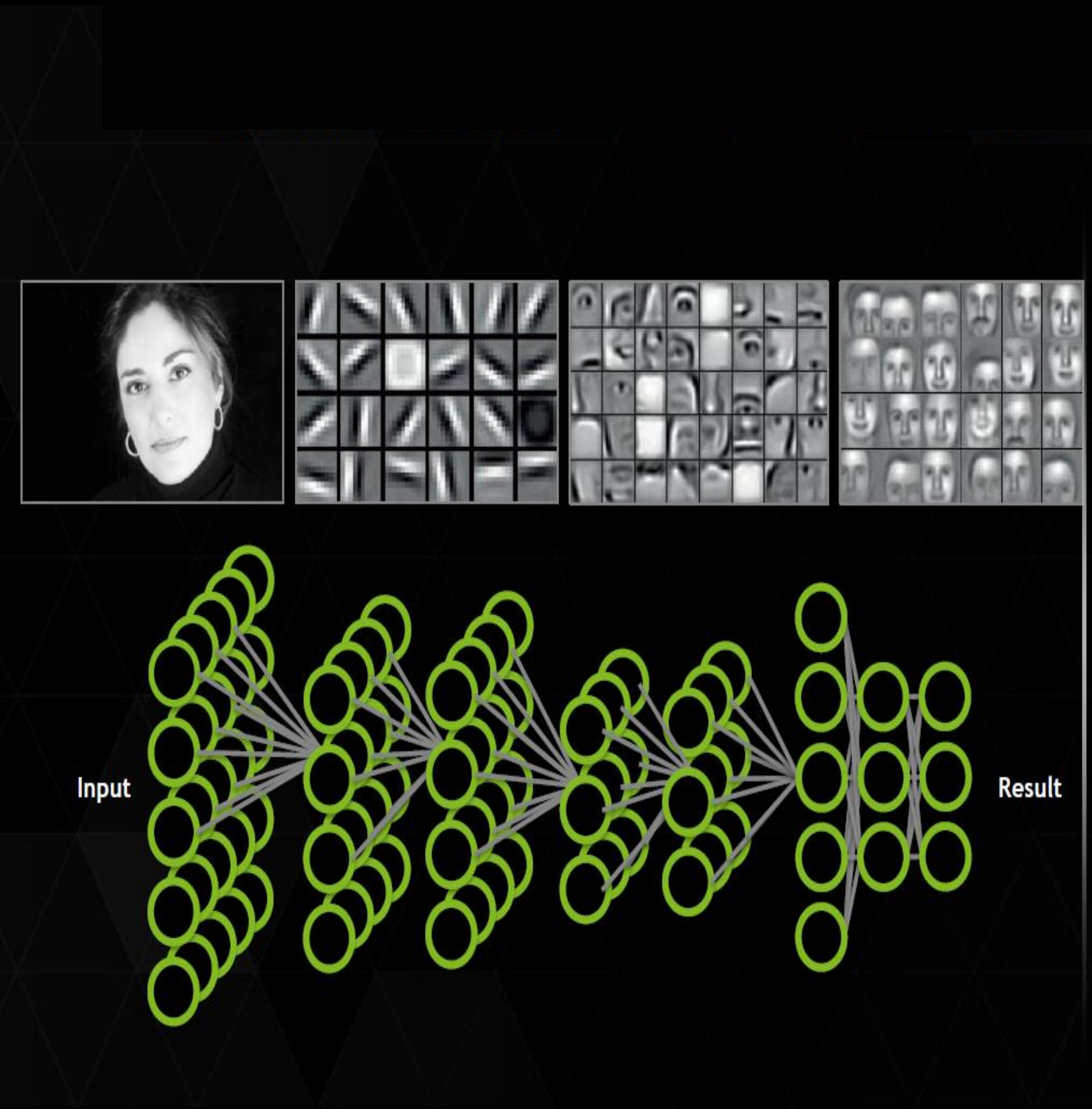
Retinal ganglion cells respond to edges

Input image
(cornea)

"Neural image"
(retinal ganglion cells)

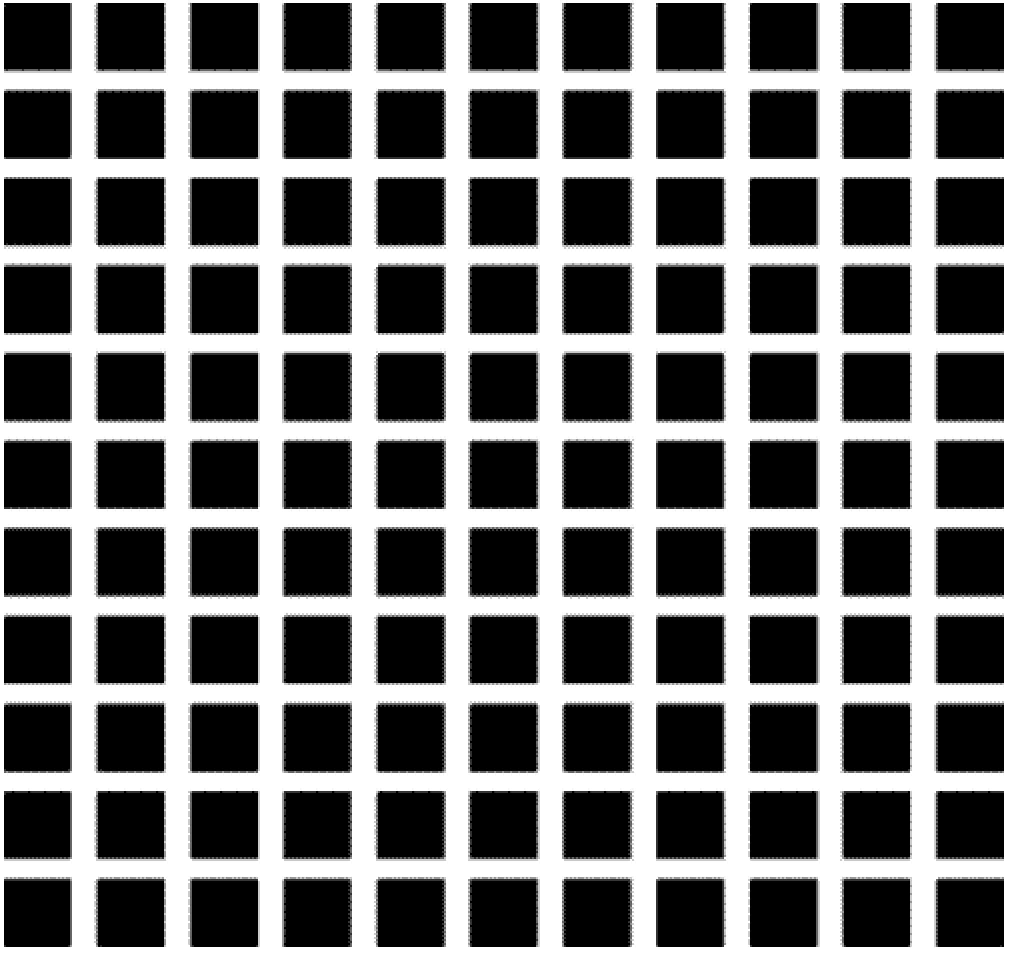
Center-surround receptive fields: emphasize edges.

Deep Learning

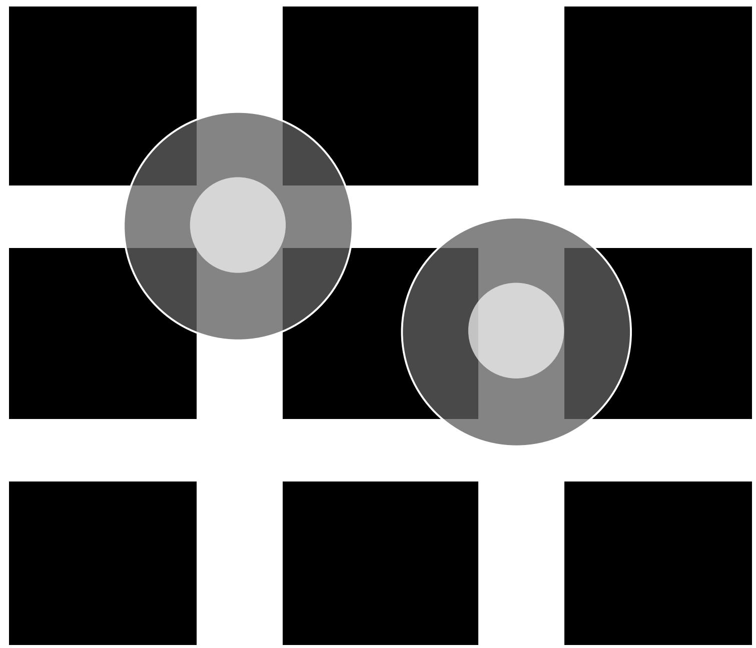


(Source: NVIDIA)

Hermann Grid Illusion



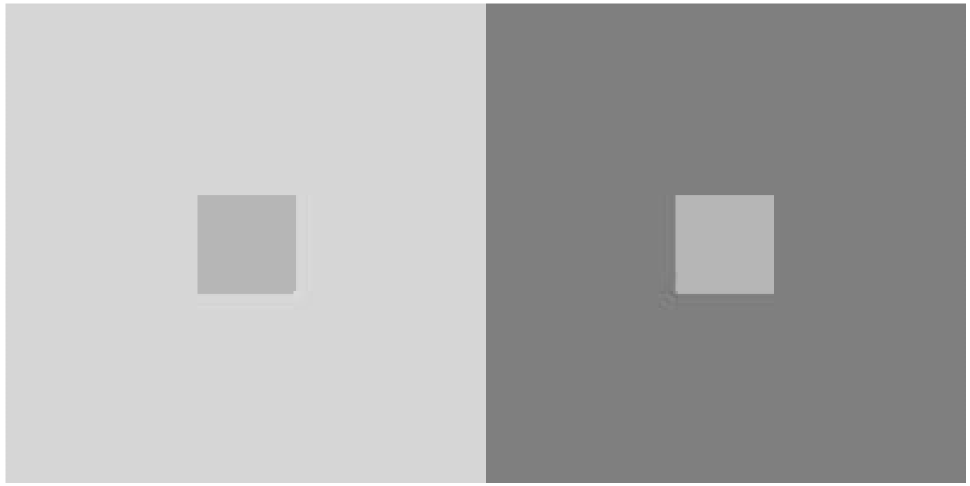
Hermann Grid Illusion: Explanation



When the image is on the first receptive field there is more light falling on the surround (inhibitory) than in the second position

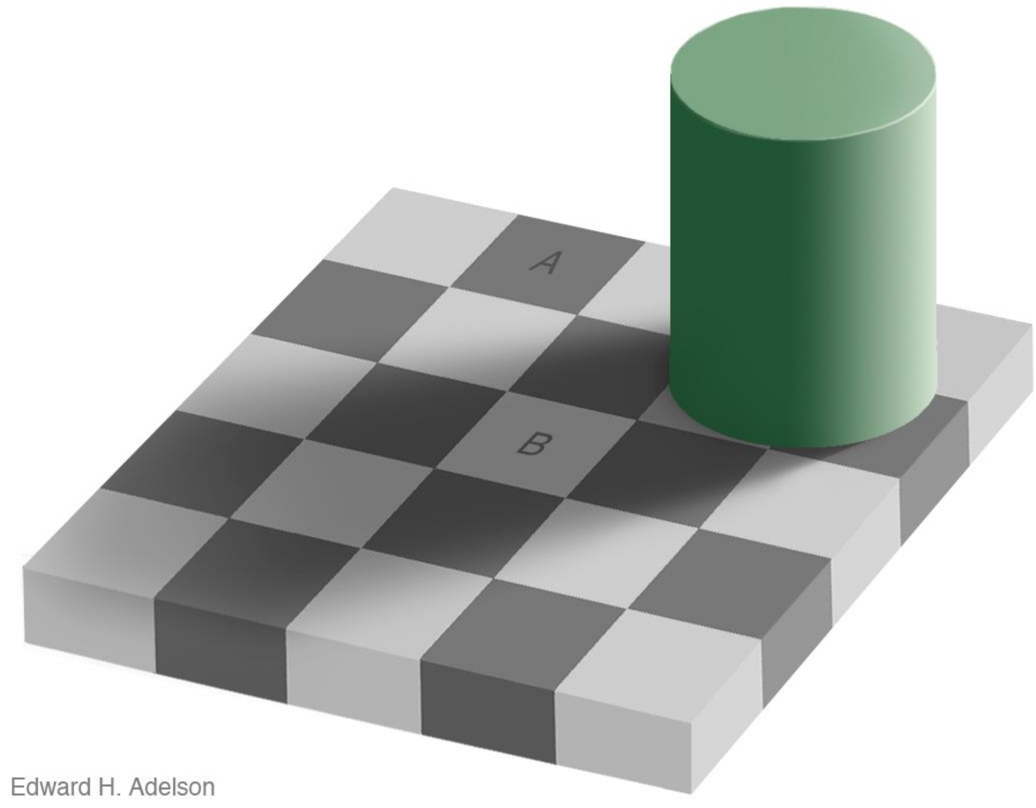
So there is more suppression and the illusion of a dark spot at the first location

Simultaneous Lightness Contrast

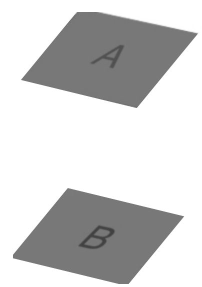


- Occurs when the lightness of an area is influenced by neighboring regions
- Our perception of lightness is not objective, but depends on the surrounding area
- The center square on the right looks lighter because the surrounding area is a darker gray

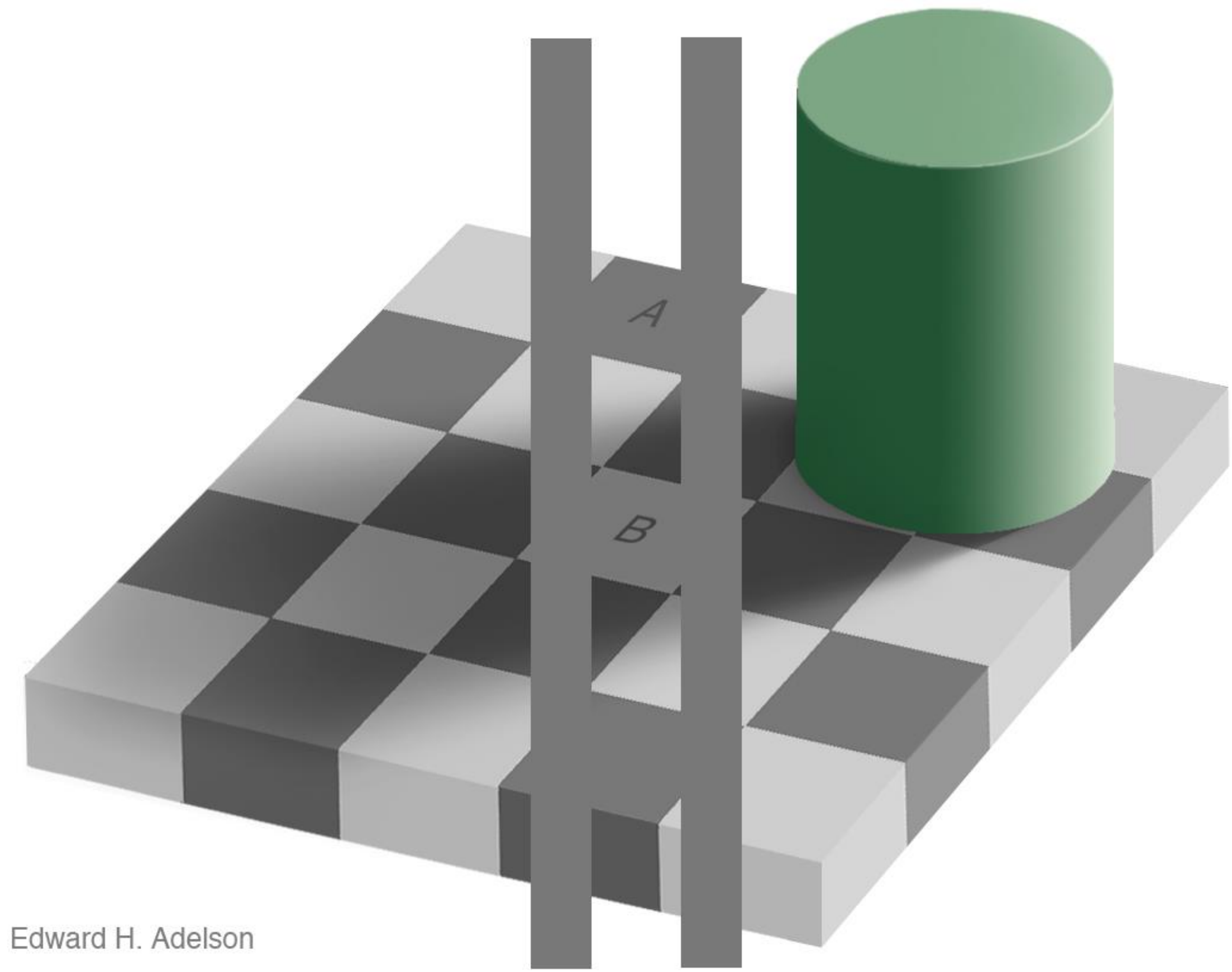
Perception of Intensity



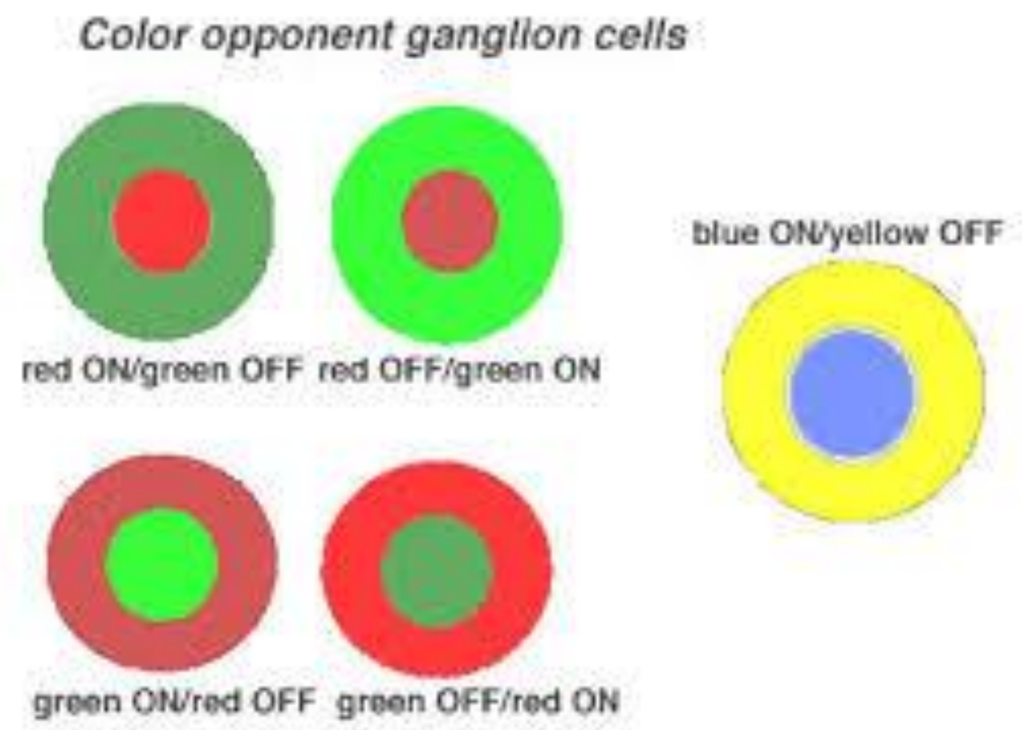
Perception of Intensity



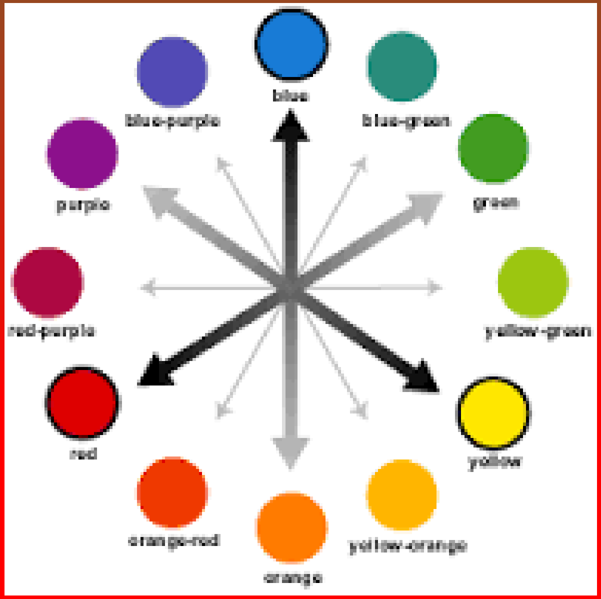
Perception of Intensity

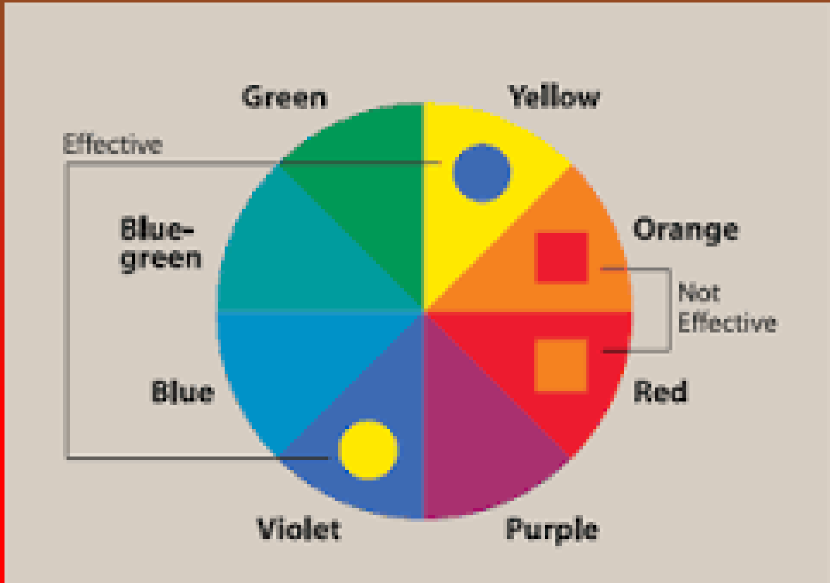


Ganglion Cells: Receptive Fields / Opponent Colors

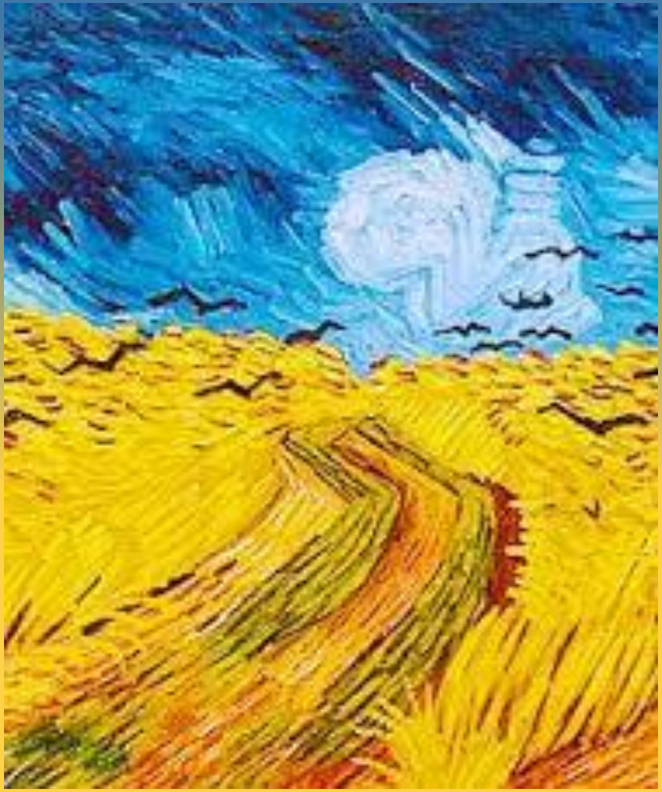


Opponent Colors



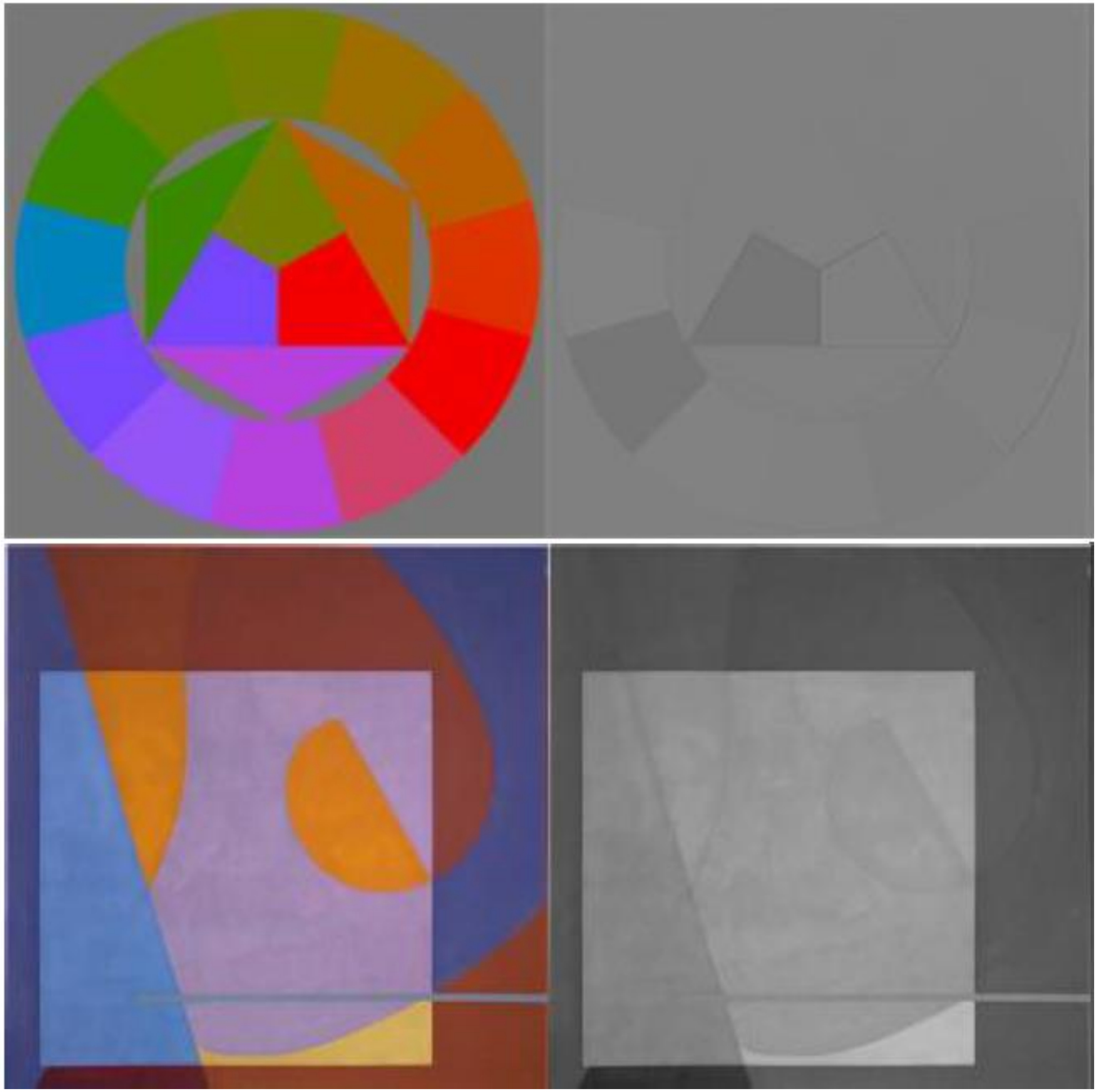


Van Gogh



Color contrast

Color contrast

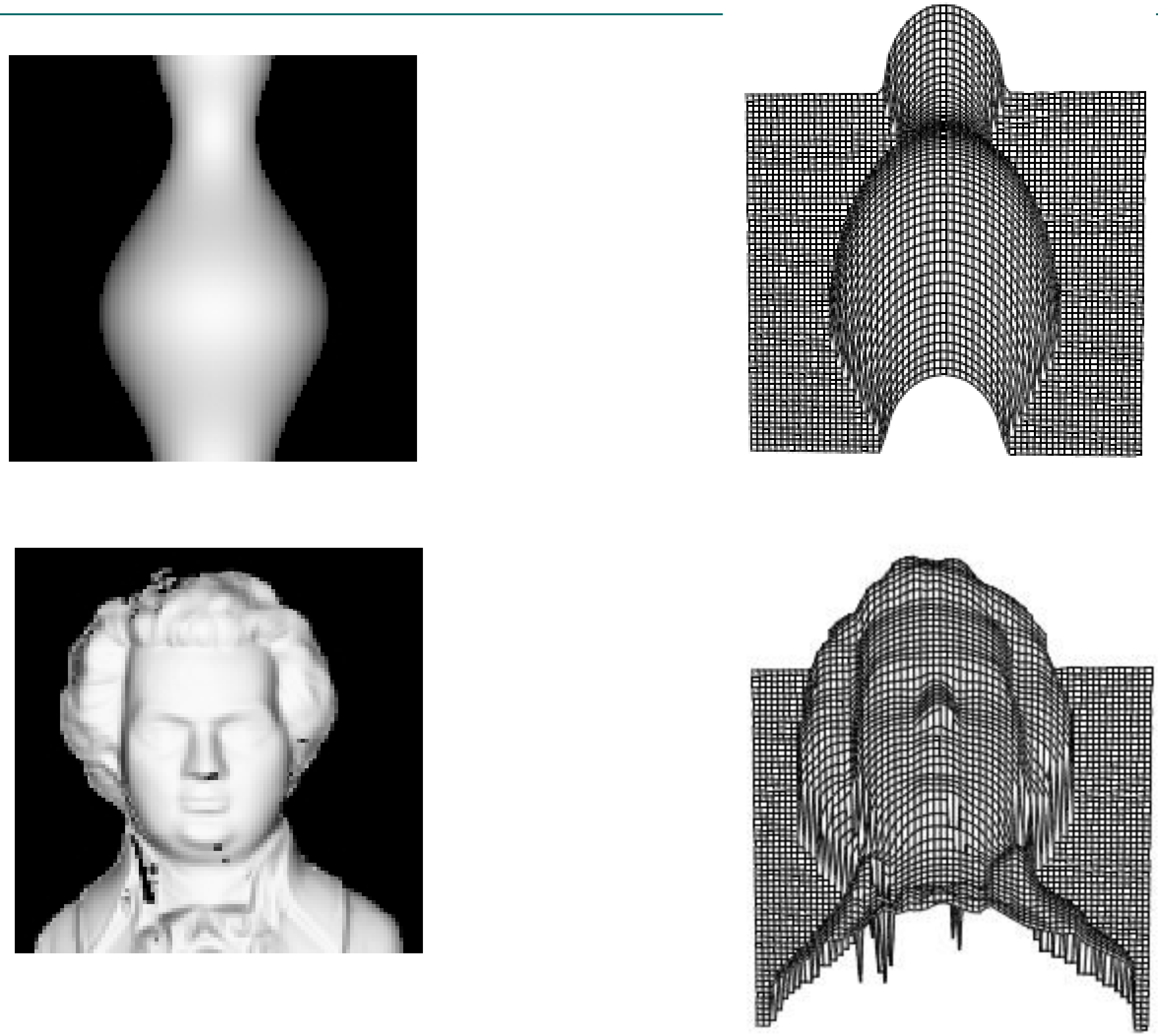


Today’s class: Image Formation

1. Projective Geometry and Camera Models
2. Light and Color Models
    2.1 Digital Image Presentation
    2.2 Light, Object and Sensor
    2.3 Color Systems
    2.4 Contrast
3. <u>Reflection Models, Shape from Shading and Photometric Stereo</u>

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai

Shape from shading

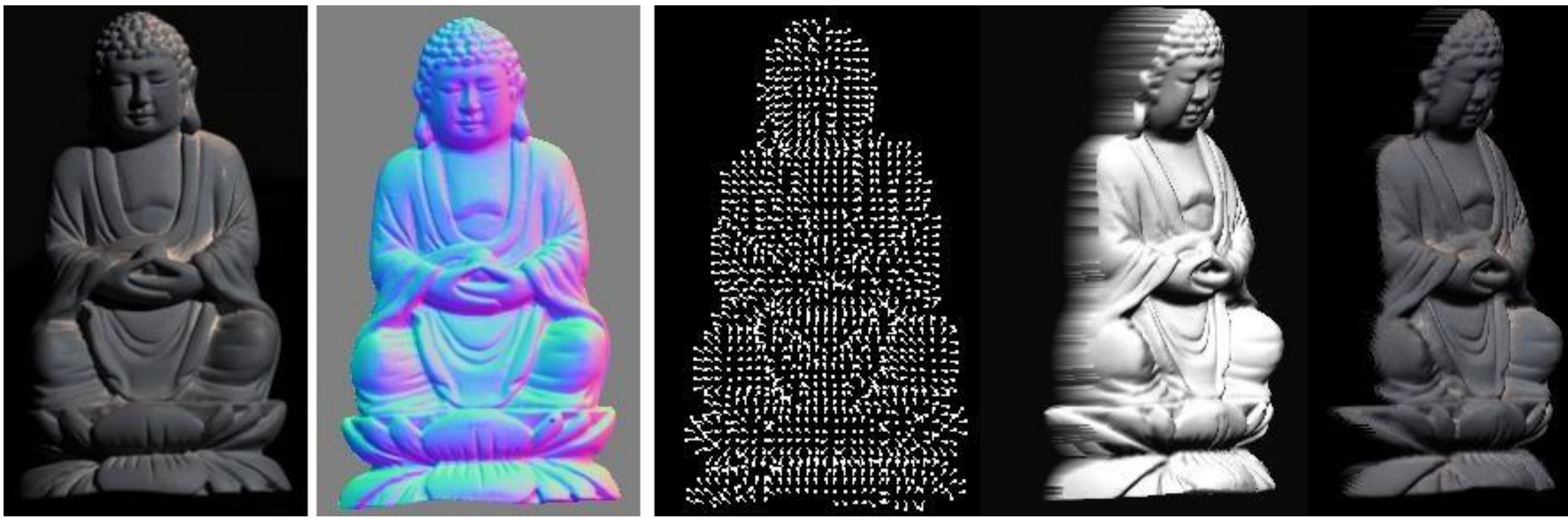


Today: Photometric Stereo

Key Idea: use pixel brightness to understand shape

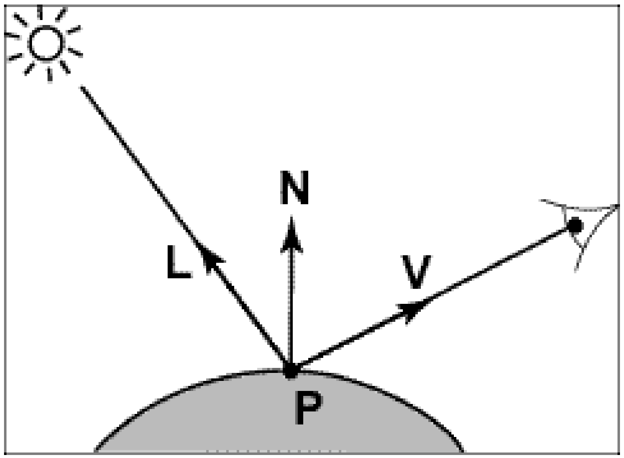
Photometric Stereo

What results can you get?



Input (1 of 12) | Normals (RGB colormap) | Normals (vectors) | Shaded 3D rendering | Textured 3D rendering

Modeling Image Formation

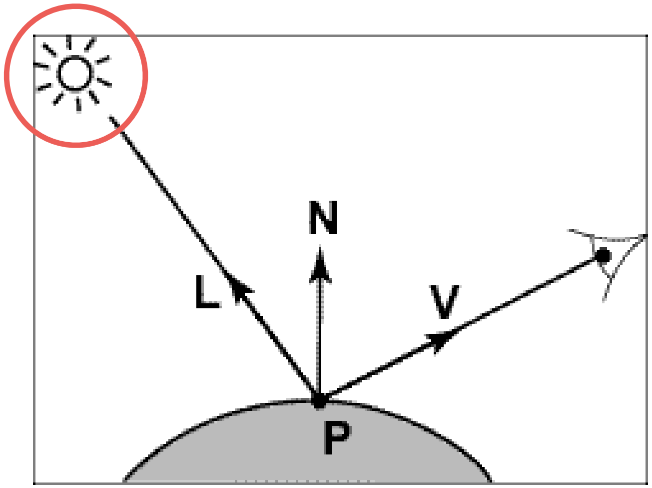


Now we need to reason about:

- How light interacts with the scene
- How a pixel value is related to light energy in the world

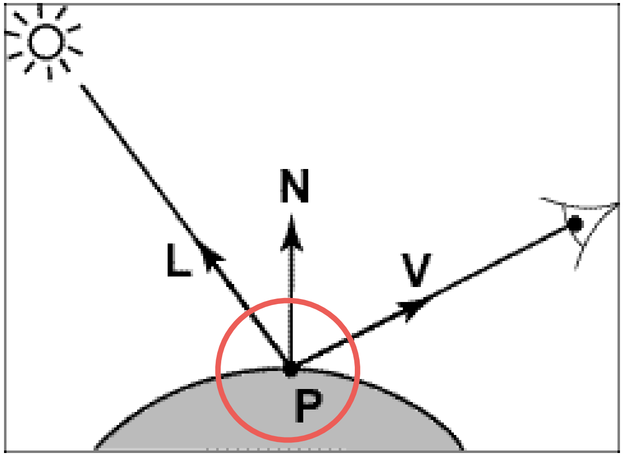
Track a “ray” of light all the way from light source to the sensor

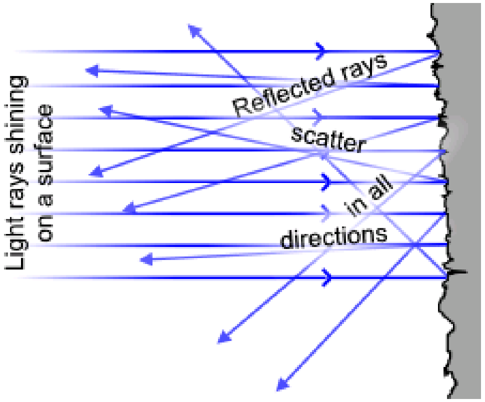
Directional Lighting



- Key property: all rays are parallel
- Equivalent to an infinitely distant point source

Lambertian Reflectance



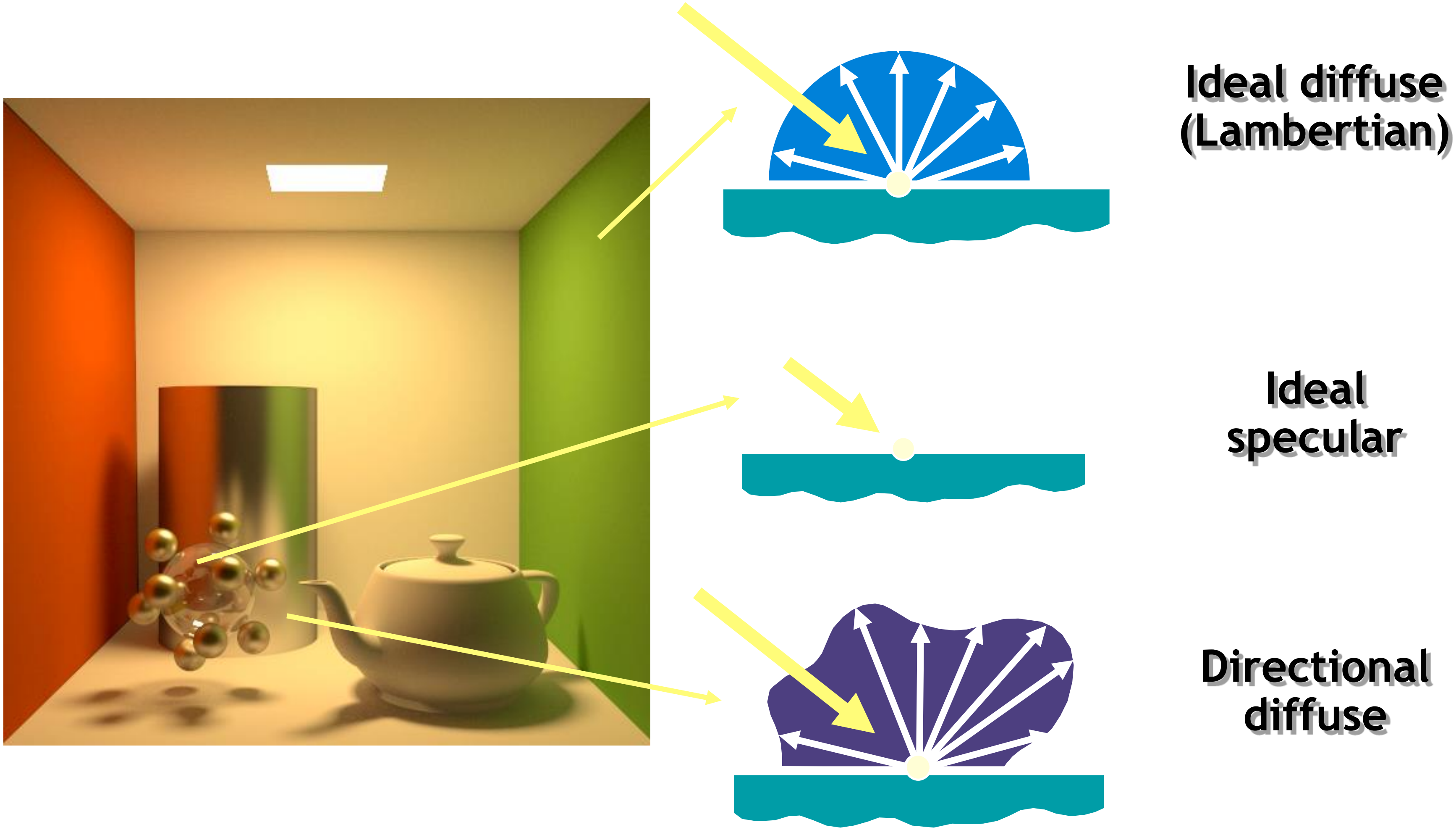


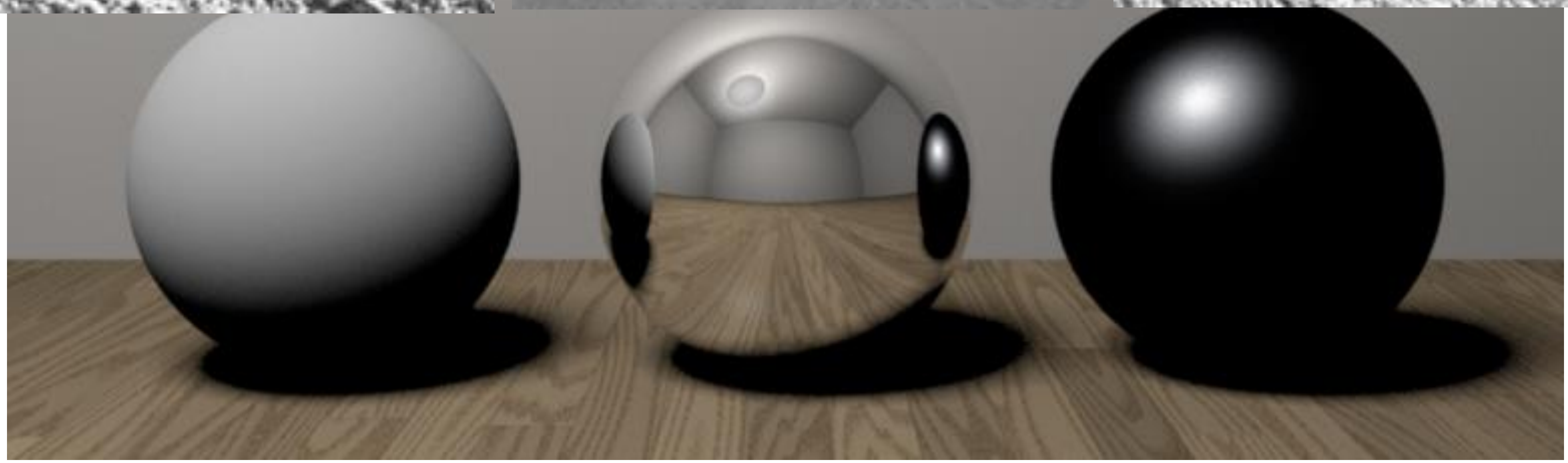
$$I = N \cdot L$$

Image intensity	=	Surface normal	•	Light direction

Image intensity $\propto$ cos(angle between N and L)

Materials - Three Forms

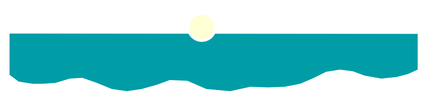




Ideal diffuse (Lambertian)

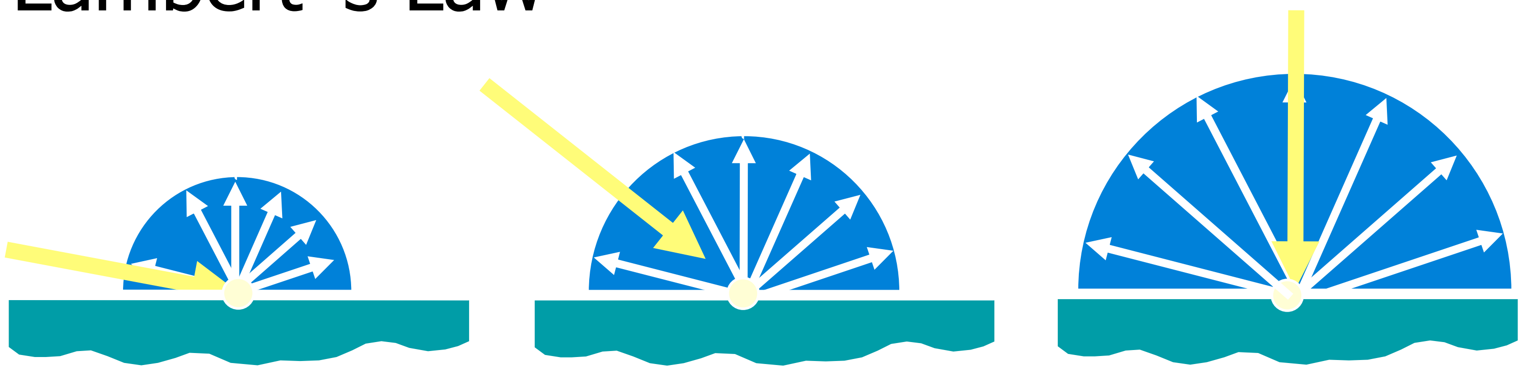
Ideal specular

Directional diffuse



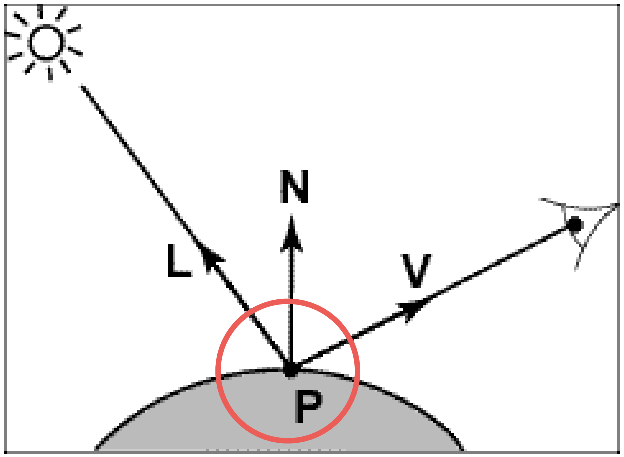
Ideal Diffuse

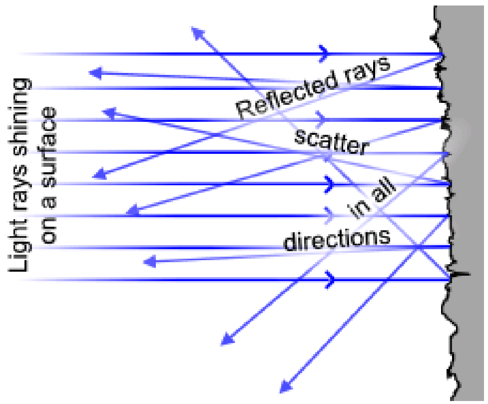
- Lambert's Law



$$I_{diffuse} = I_{light} k_d cos(\theta)$$
$$I_{diffuse} = I_{light} k_d N.L$$

Lambertian Reflectance



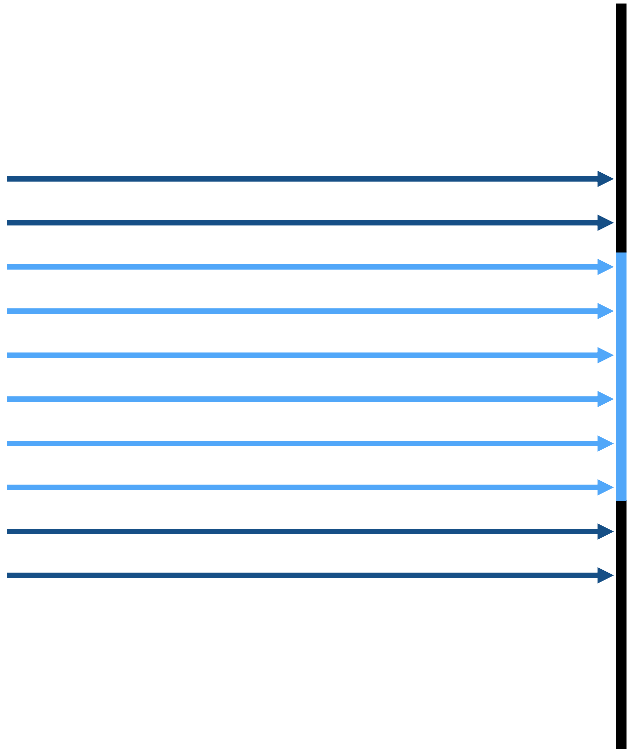


1. Reflected energy is proportional to cosine of angle between L and N **(incoming)**

2. Measured intensity is viewpoint-independent **(outgoing)**

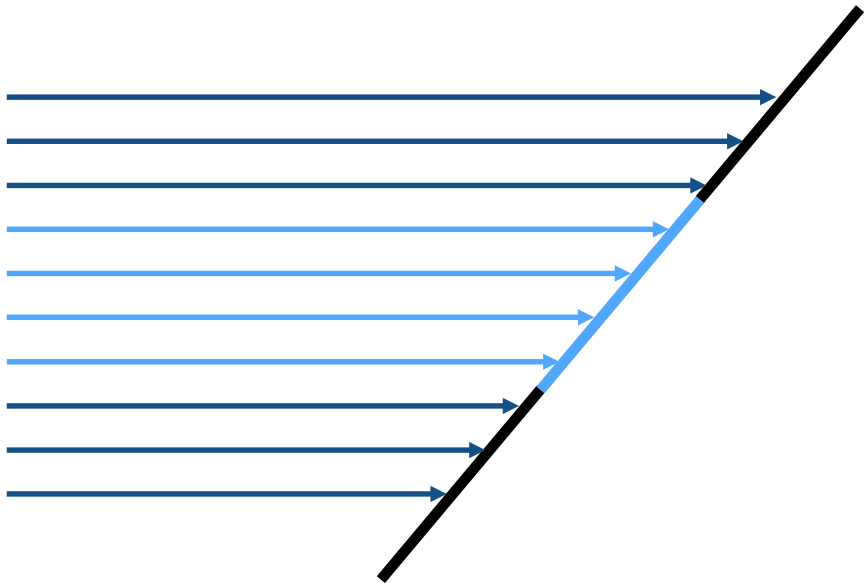
Lambertian Reflectance: Incoming

1. Reflected energy is proportional to cosine of angle between L and N



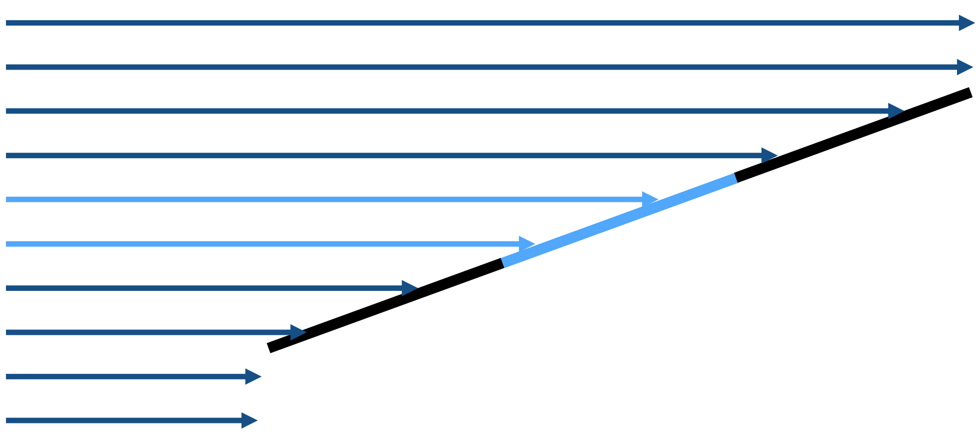
Lambertian Reflectance: Incoming

1. Reflected energy is proportional to cosine of angle between L and N



Lambertian Reflectance: Incoming

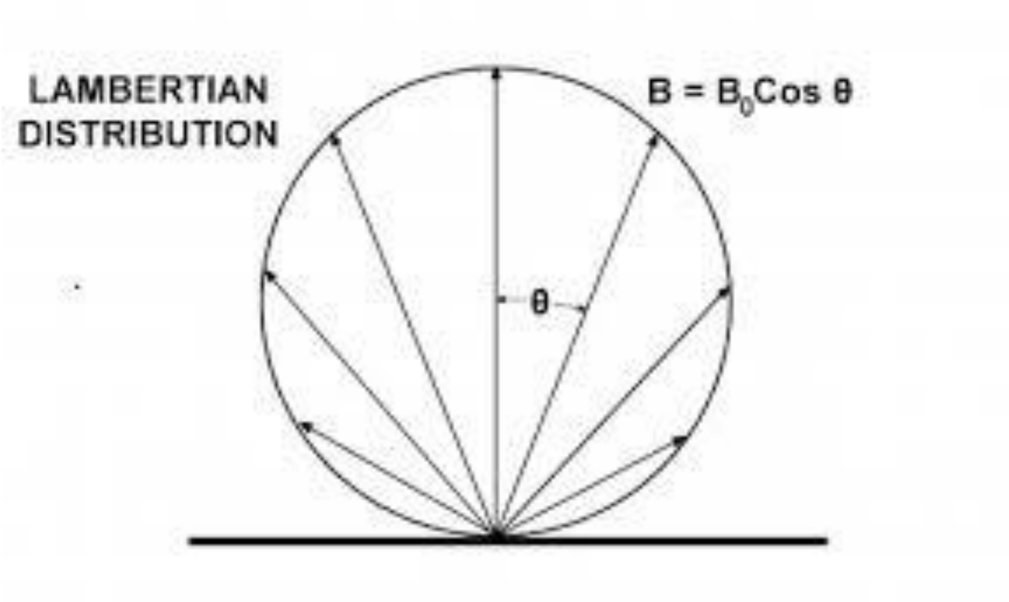
1. Reflected energy is proportional to cosine of angle between L and N



Light hitting surface is proportional to the cosine

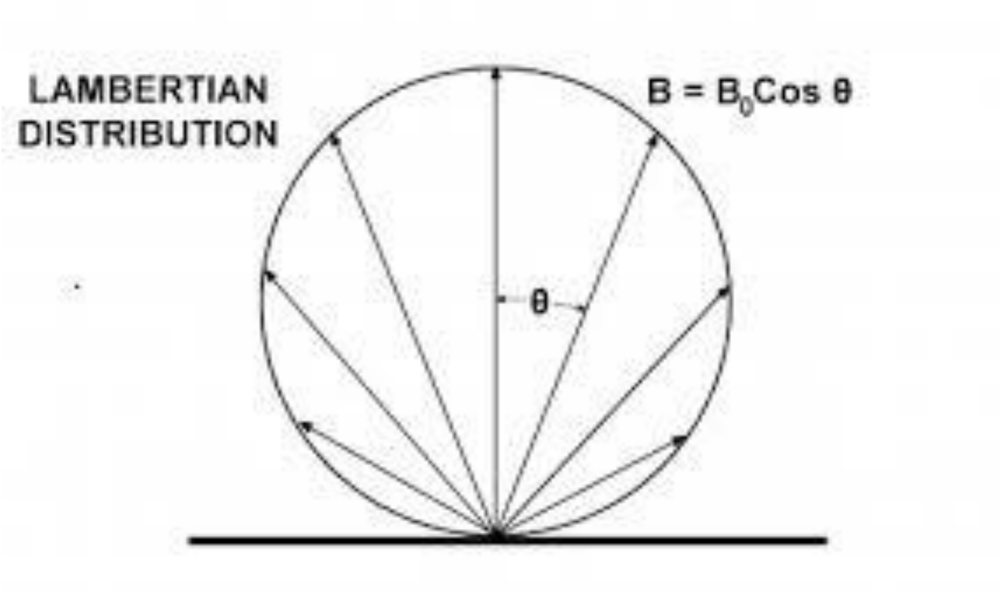
Lambertian Reflectance: Outgoing

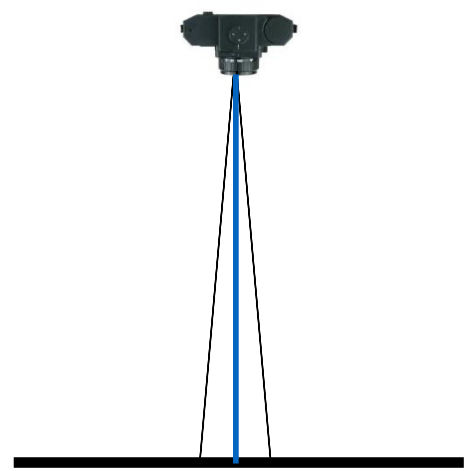
1. Radiance (what we see) is viewpoint-independent



Lambertian Reflectance: Outgoing

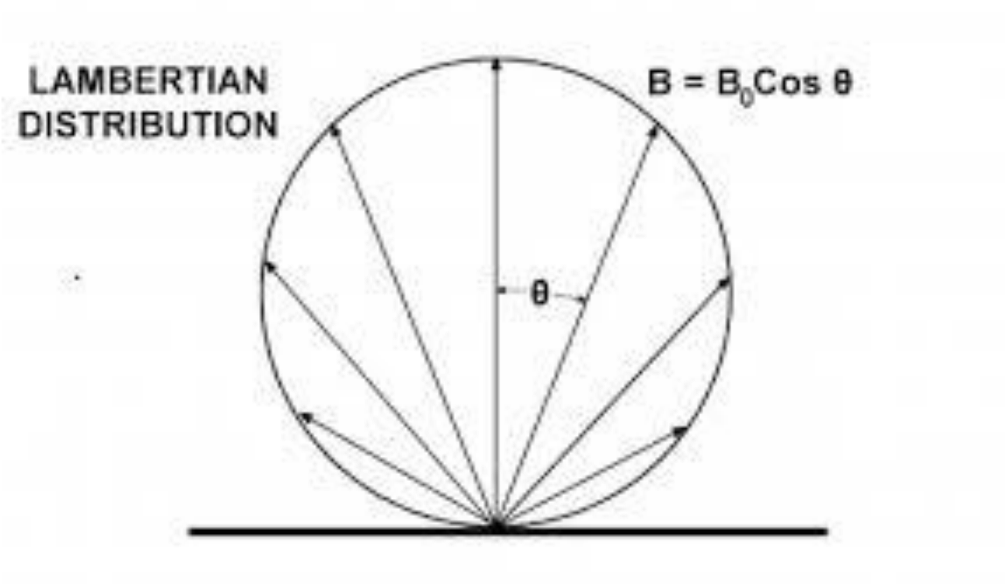
1. Radiance (what the eye sees) is viewpoint-independent

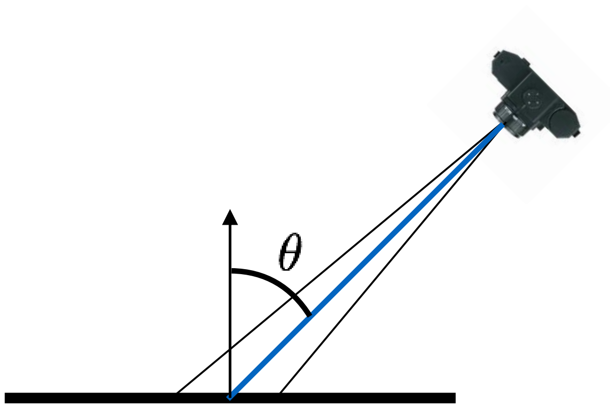




Lambertian Reflectance: Outgoing

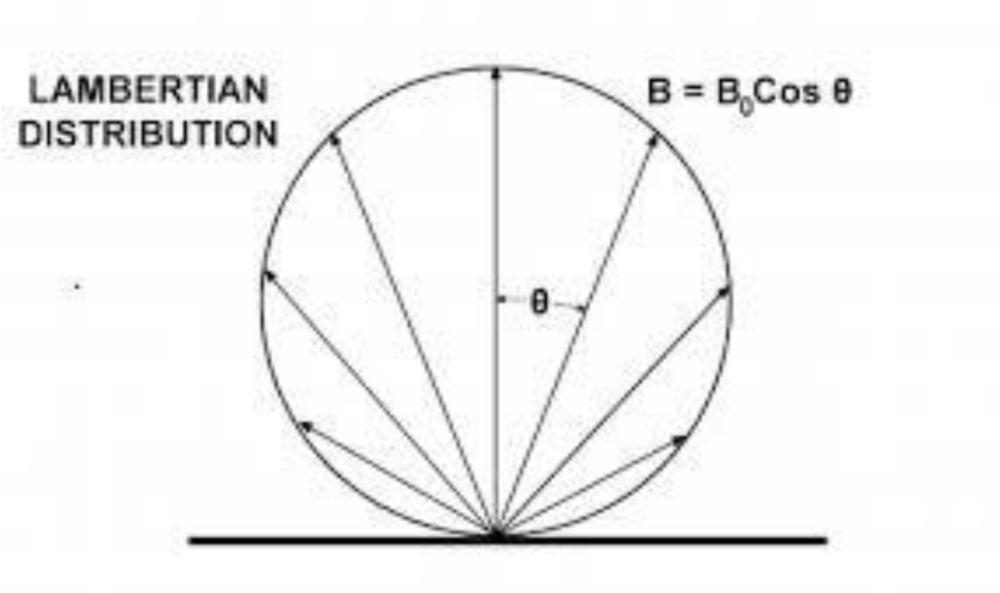
1. Measured intensity is viewpoint-independent

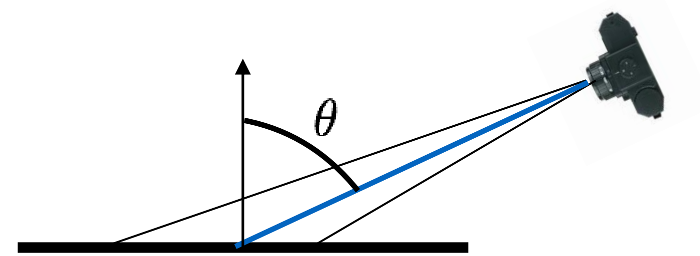




Lambertian Reflectance: Outgoing

1. Measured intensity is viewpoint-independent



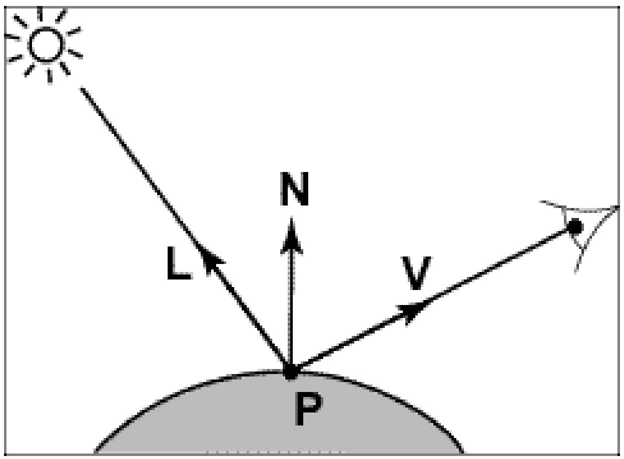


A cos (θ)

Radiance
(what eye sees)

$$\propto B_0 \cancel{\cos(\theta)} \frac{1}{\cancel{\cos(\theta)}}$$

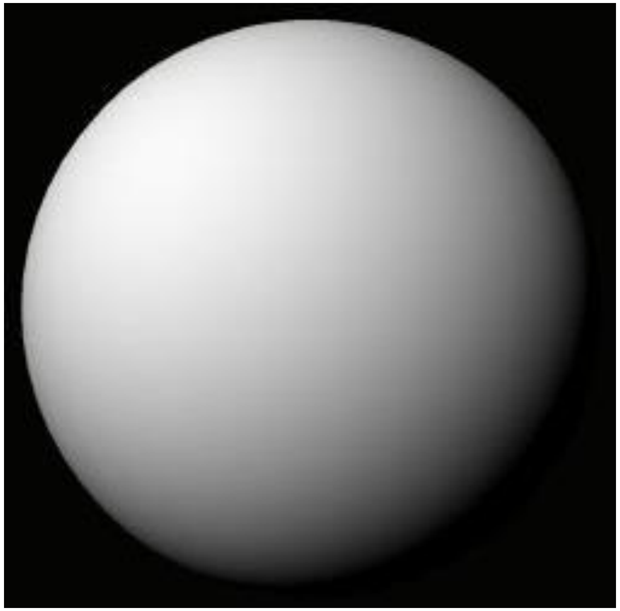
Image Formation Model: Final



$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

1. Diffuse albedo: what fraction of incoming light is reflected?
   - Introduce scale factor k_d
2. Light intensity: how much light is arriving?
   - Compensate with camera exposure (global scale factor)
3. Camera response function
   - Assume pixel value is linearly proportional to incoming energy (perform radiometric calibration if not)

A Single Image: Shape from Shading



$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

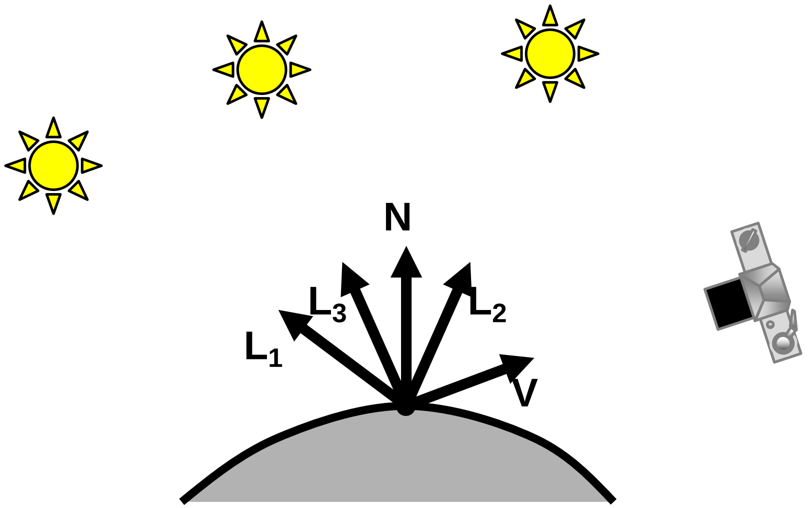
Assume k_d is 1 for now.

What can we measure from one image?

- $\cos^{-1}(I)$ is the angle between N and L
- Add assumptions:
  - A few known normals (e.g. silhouettes)
  - Smoothness of normals

In practice, SFS doesn’t work very well:
assumptions are too restrictive,
too much ambiguity in nontrivial scenes.

Multiple Images: Photometric Stereo



$$I_1 = k_d \mathbf{N} \cdot \mathbf{L_1}$$
$$I_2 = k_d \mathbf{N} \cdot \mathbf{L_2}$$
$$I_3 = k_d \mathbf{N} \cdot \mathbf{L_3}$$

Write this as a matrix equation:

$$\left[\begin{array}{ccc} I_1 & I_2 & I_3 \end{array}\right] = k_d \mathbf{N}^T \left[\begin{array}{ccc} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{array}\right]$$

Solving the Equations

$$\underbrace{\begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 1\times 3}} = \underbrace{k_d \mathbf{N}^T}_{\substack{\mathbf{G} \\ 1\times 3}} \underbrace{\begin{bmatrix} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{bmatrix}}_{\substack{\mathcal{L} \\ 3\times 3}}$$

$$\mathbf{G} = \mathbf{I}\mathbf{L}^{-1}$$

$$k_d = \|\mathbf{G}\|$$

$$\mathbf{N} = \frac{1}{k_d}\mathbf{G}$$

Solving the Equations

$$\underbrace{\begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 1\times 3}} = \underbrace{k_d \mathbf{N}^T}_{\substack{\mathbf{G} \\ 1\times 3}} \underbrace{\begin{bmatrix} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{bmatrix}}_{\substack{\mathcal{L} \\ 3\times 3}}$$

$$\mathbf{G} = \mathbf{I}\mathbf{L}^{-1}$$

- When is L nonsingular (invertible)?
  - >= 3 light directions are linearly independent, or:
  - All light direction vectors cannot lie in a plane.
- What if we have more than one pixel?
  - Stack them all into one big system.

More than Three Lights

$$\begin{bmatrix} I_1 & \dots & I_n \end{bmatrix} = k_d \mathbf{N}^T \begin{bmatrix} \mathbf{L_1} & \dots & \mathbf{L_n} \end{bmatrix}$$

- Solve using least squares (normal equations):

$$\begin{aligned} \mathbf{I} &= \mathbf{GL} \\ \mathbf{IL^T} &= \mathbf{GLL^T} \\ \mathbf{G} &= (\mathbf{IL^T})(\mathbf{LL^T})^{-1} \end{aligned}$$

- Equivalently use SVD
- Given G, solve for N and k_d as before.

More than one pixel

Previously:

$$\underset{1 \times \#\ \text{images}}{\boxed{I}} = \underset{1 \times 3}{\boxed{N}} * \underset{3 \times \#\ \text{images}}{\boxed{L}}$$

More than one pixel

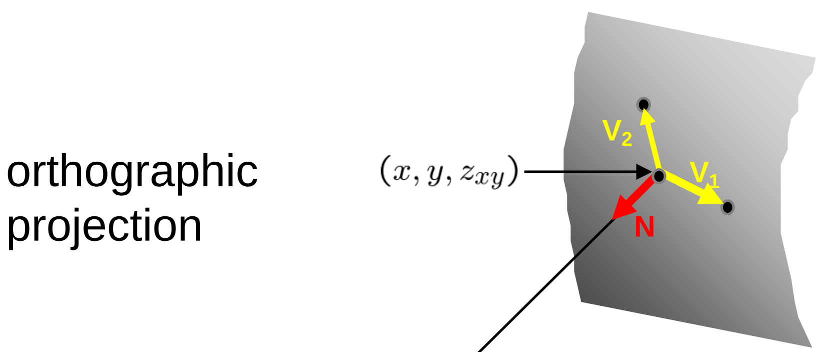
Stack all pixels into one system:

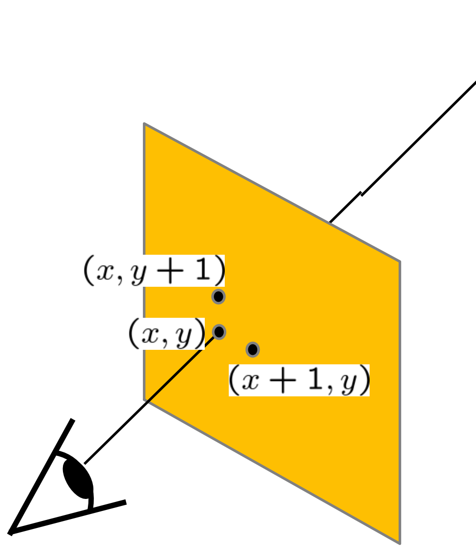
$$\underset{p \times \#\text{ images}}{I} = \underset{p \times 3}{N} * \underset{3 \times \#\text{ images}}{L}$$

Solve as before.

Depth Map from Normal Map

- We now have a surface normal, but how do we get depth?





Assume a smooth surface

$$
\begin{aligned}
V_1 &= (x+1, y, z_{x+1,y}) - (x, y, z_{xy}) \\
&= (1, 0, z_{x+1,y} - z_{xy})
\end{aligned}
$$

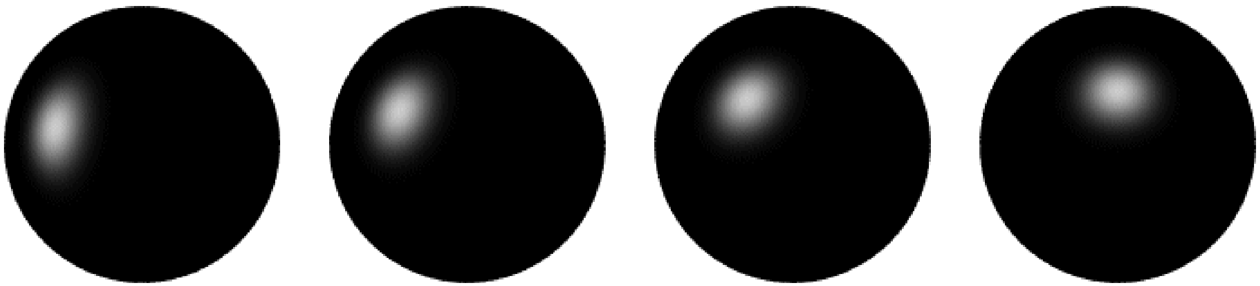
$$
\begin{aligned}
0 &= N \cdot V_1 \\
&= (n_x, n_y, n_z) \cdot (1, 0, z_{x+1,y} - z_{xy}) \\
&= n_x + n_z(z_{x+1,y} - z_{xy})
\end{aligned}
$$

Get a similar equation for $\mathbf{V_2}$

- Each normal gives us two linear constraints on z
- compute z values by solving a matrix equation

Determining Light Directions

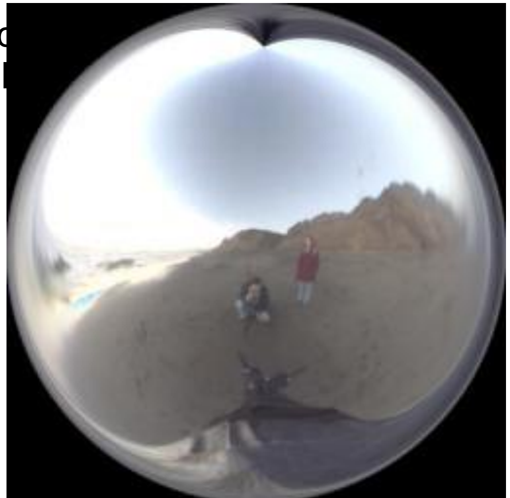
- Trick: Place a mirror ball in the scene.



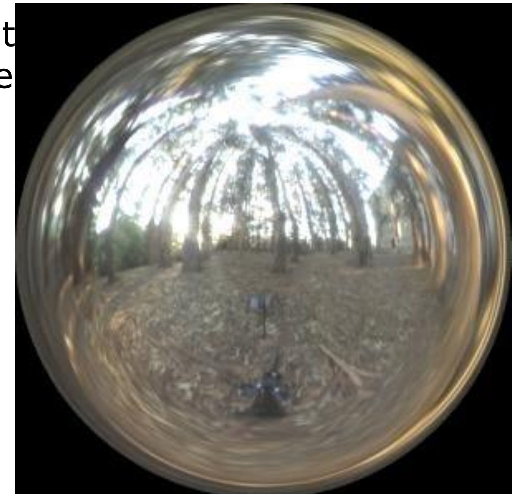
- The location of the highlight is determined by the light source direction.

Real-World HDR Lighting Environments

Funsto
Beac

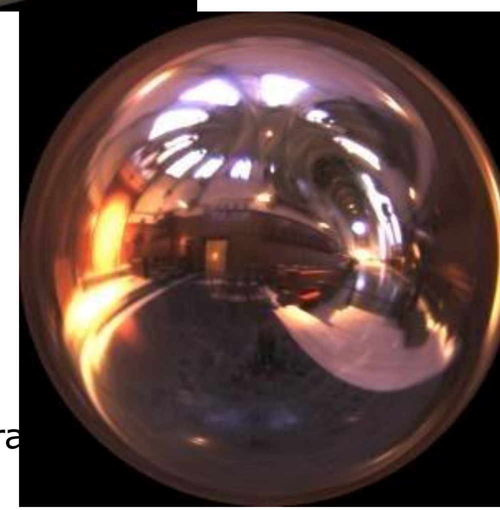


Eucalypt
Grove



Uffizi
Gallery

Grace
Cathedra

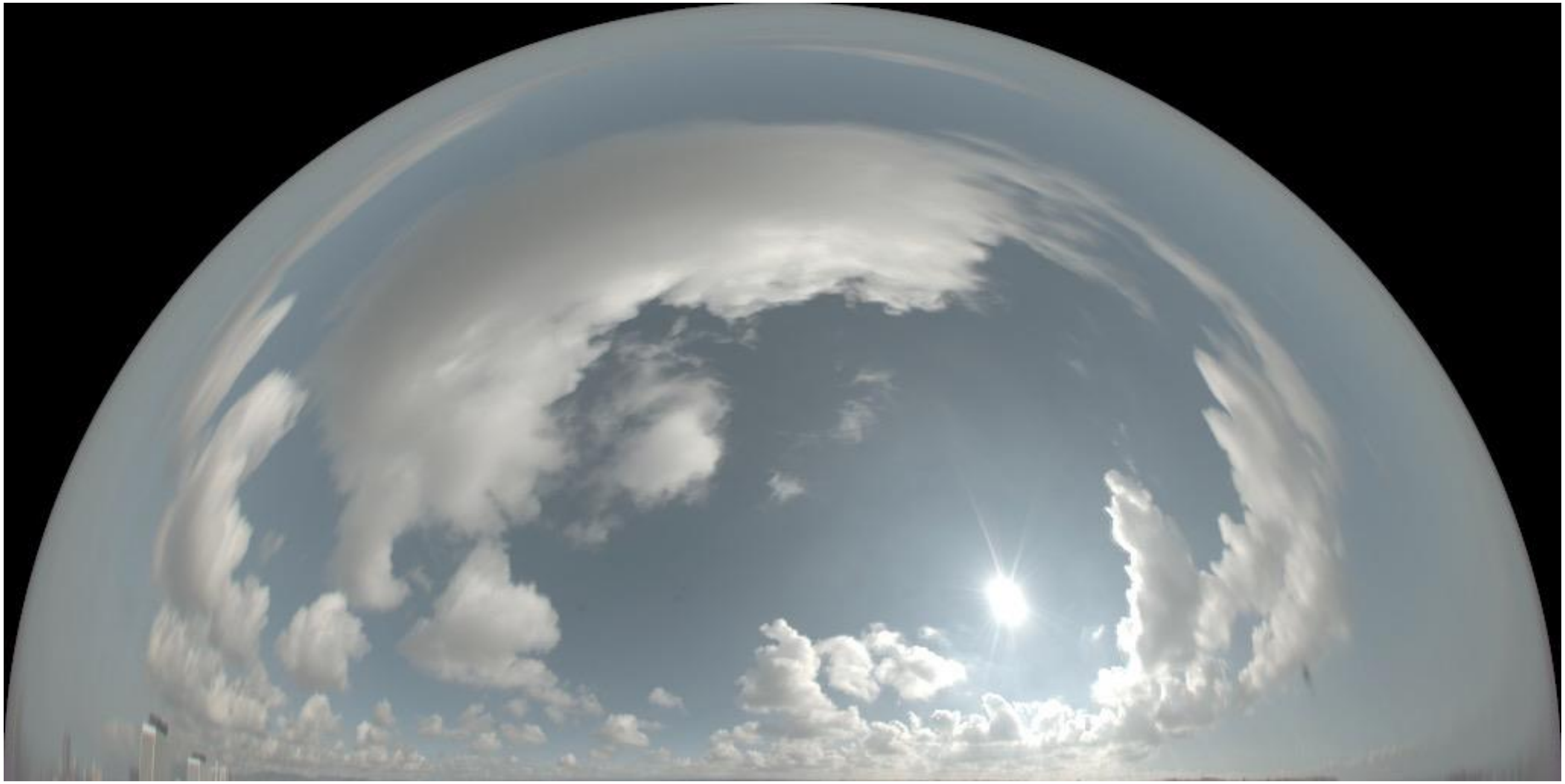


Lighting Environments from the Light Probe Image Gallery:
http://www.debevec.org/Probes/

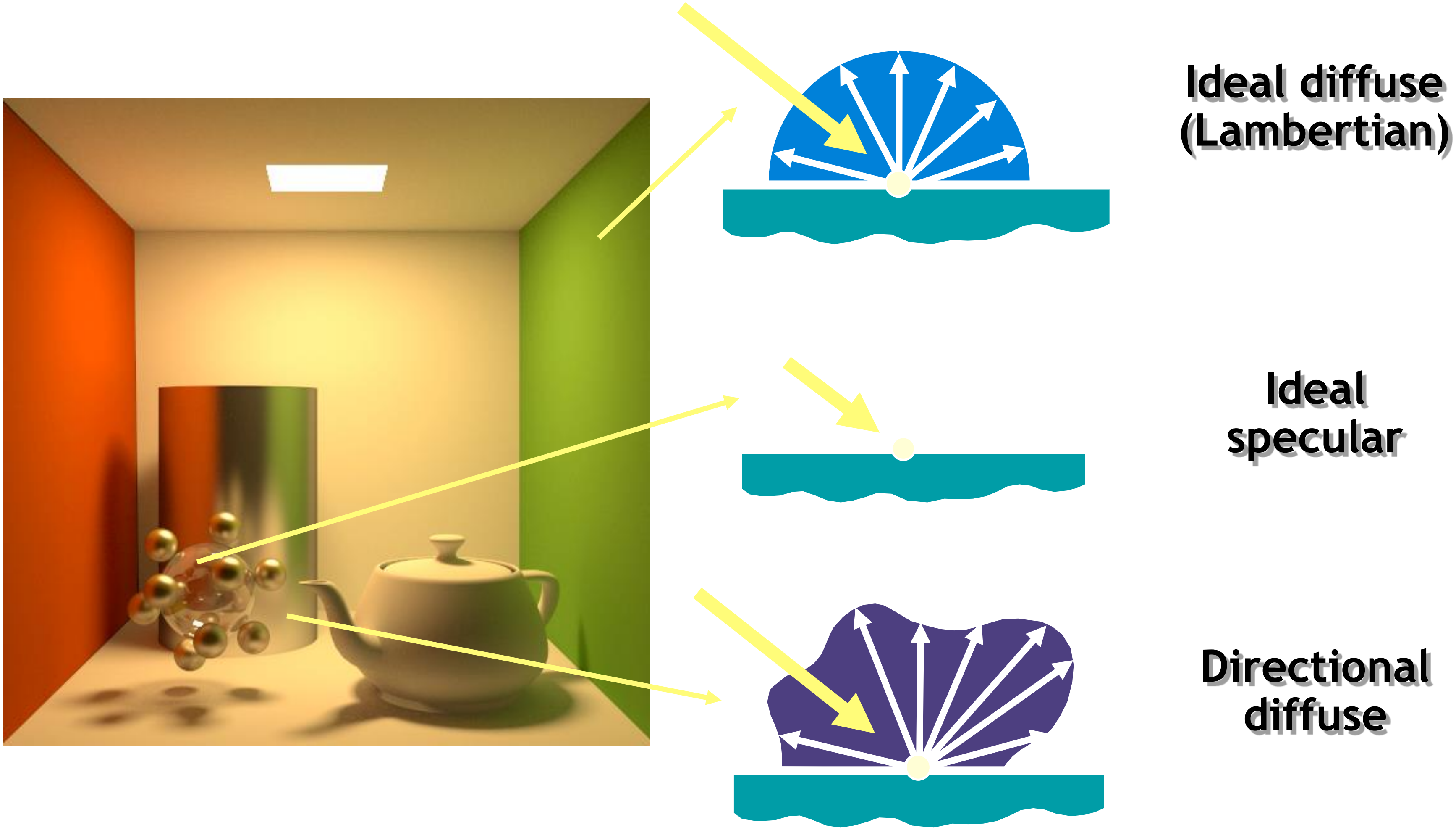
Mirrored Sphere

Canon
Canon
REMOTE SWITCH

HDRI Sky Probe

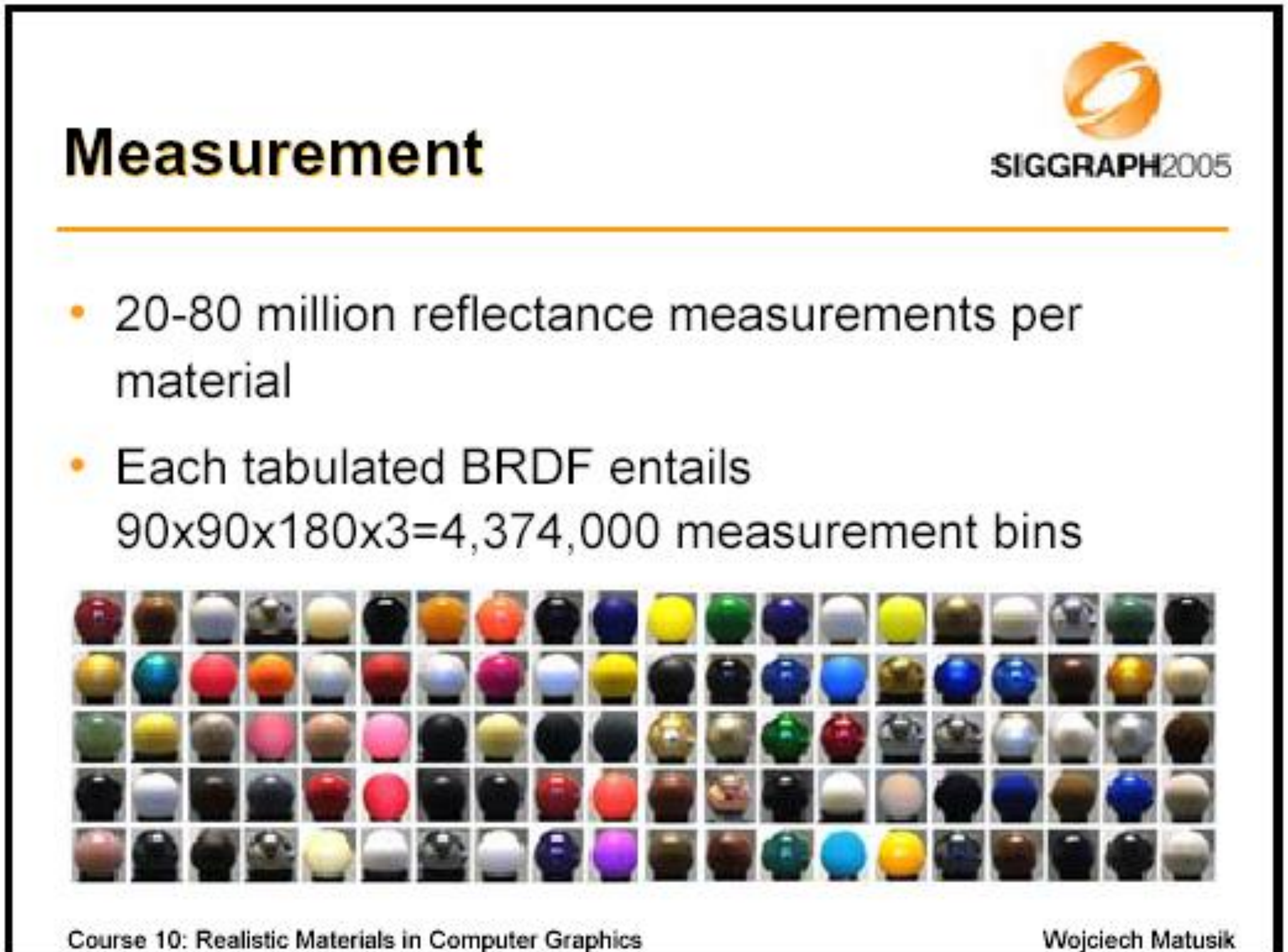


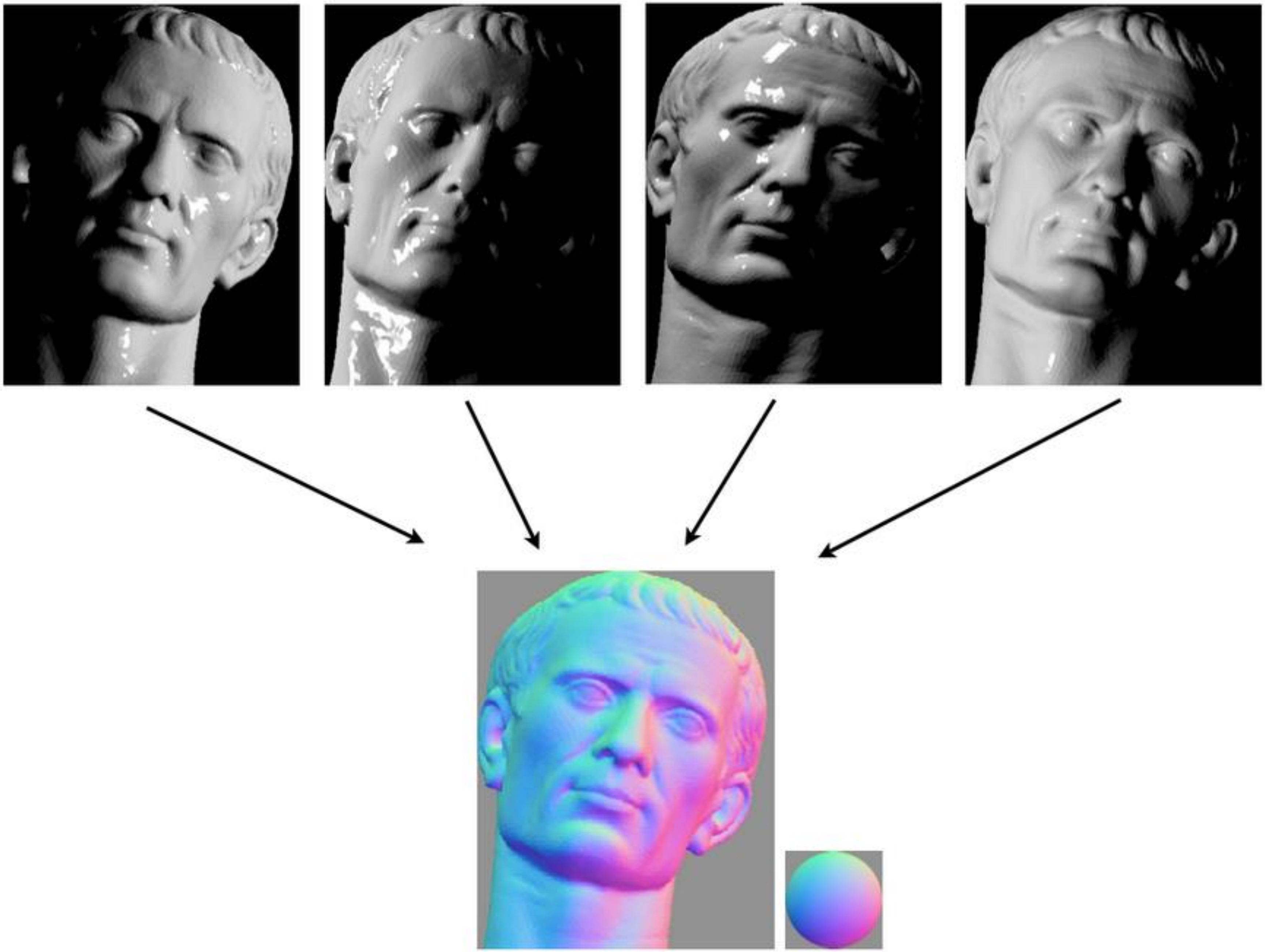
Materials - Three Forms



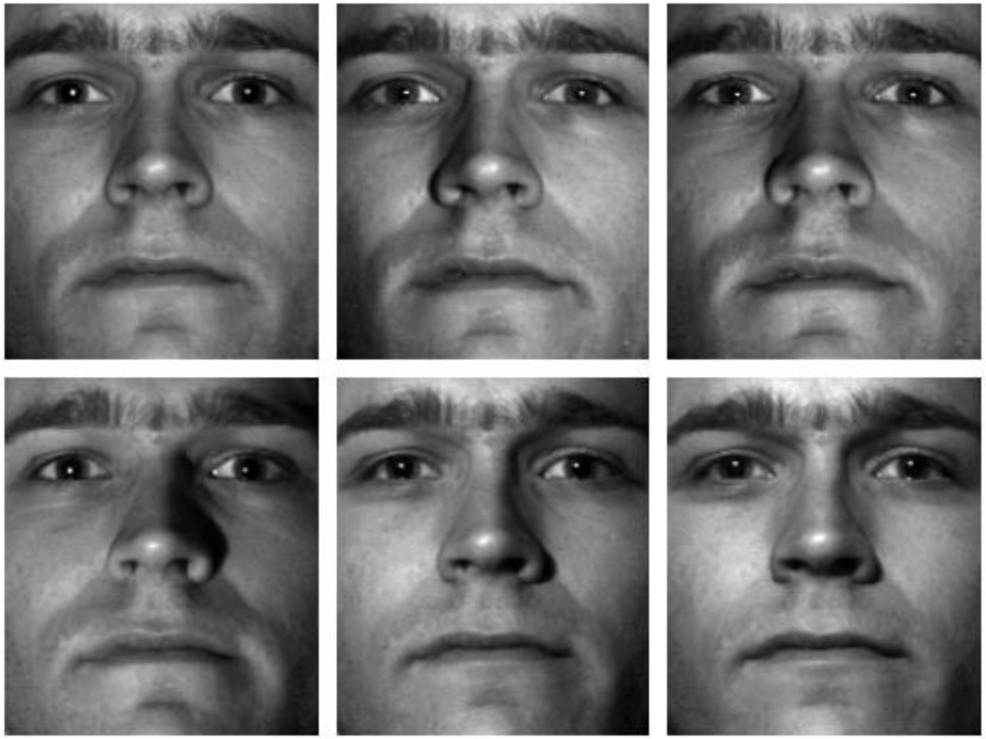
BRDF databases

- MERL (Matusik et al.): 100 isotropic, 4 nonisotropic, dense





Results



from Athos Georghiades

Today's class: Image Formation

1. Projective Geometry and Camera Models
2. Light and Color Models
   - 2.1 Digital Image Presentation
   - 2.2 Light, Object and Sensor
   - 2.3 Color Systems
   - 2.4 Contrast
3. Reflection Models, Shape from Shading and Photometric Stereo

Including slides from Derek Hoiem, Alexei Efros, Steve Seitz, and David Forsyth, James Hays, Jinxiang Chai