Computer Vision 1

HC1b

Camera
Image Formation
Color

Dr. Martin Oswald, Dr. Dimitris Tzionas, Dr. Arun Mukundan,

{m.r.oswald, d.tzionas, a.mukundan}@uva.nl

University of Amsterdam

computer vision lab

Outline

- Pinhole Camera
- Geometric Image Formation
- Photometric Image Formation
- Image Representation

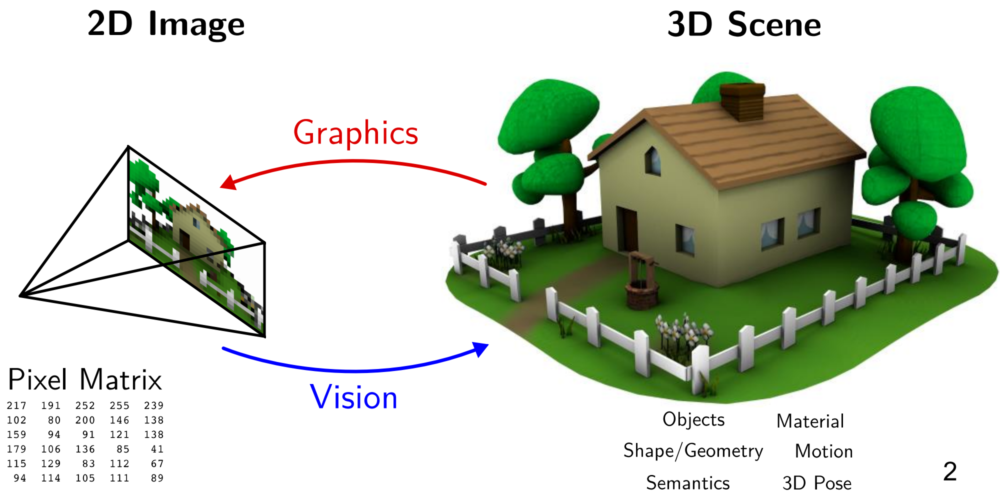
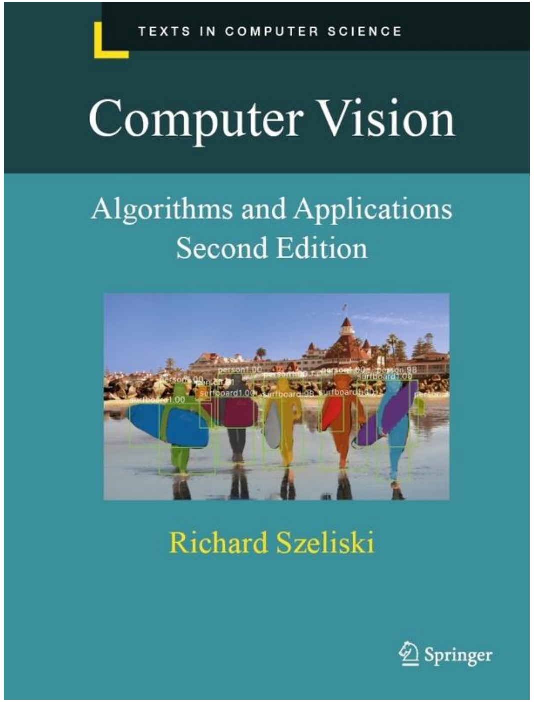


Figure credit:
Andreas Geiger

Textbook – Sections

- Sec. 2.1.[1-2]
- Sec. 2.1.[4-5]
- Sec. 2.2
- Sec. 2.3.[1-2]



Human Vision – Retina

Optics of eye → Create a **focused 2D image** of 3D world **on retina**

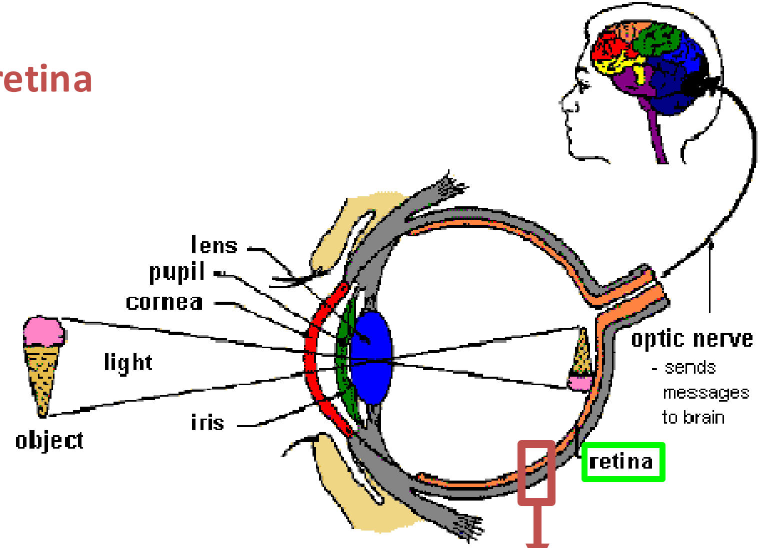
Retina → The eye's innermost & **light-sensitive layer** of tissue

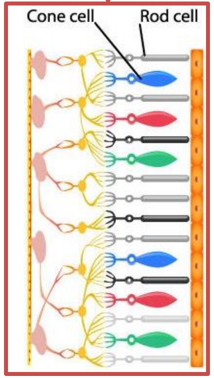
Luminance ~ Sensed in **light-sensitive cells** in **retina**
(visual energy) (**rods** & **cones**)

Collection of measurements → Form 'image'

Collection of pixels on screen → Form digital image

Retina: **Processes** that image &
Send **impulses** along optic nerve to **visual cortex**
to create **visual perception**





Human Vision – Retina

Peyman Milanfar
@docmilanfar

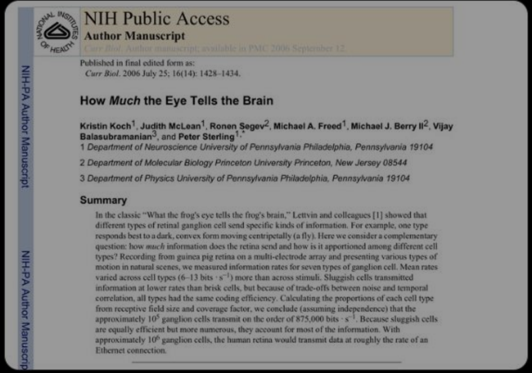
Distinguished Scientist at Google.

The retina is arguably the most impressive part of the brain - it's also the only part of the brain that faces the world directly - it's a sensor and processor in one

Consumes 50% more energy per gram than the rest of the brain.

1000:1 compression from retina to optic nerve
25000:1 from optic nerve to brain

For every 1Gb collected by the retina, 1Mb is sent to the brain thru the optic nerve and < 100bits used, at a rate of about 875Kbps. Similar to broadband internet.


NIH Public Access
Author Manuscript
Curr Biol. Author manuscript; available in PMC 2006 September 12.

Published in final edited form as:
Curr Biol. 2006 July 25; 16(14): 1428–1434.

How *Much* the Eye Tells the Brain

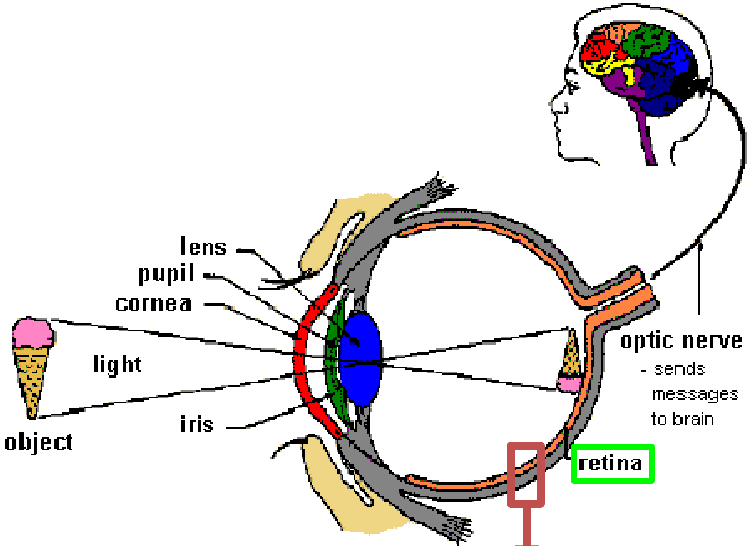
Kristin Koch[1], Judith McLean[1], Ronen Segev[2], Michael A. Freed[1], Michael J. Berry II[2], Vijay Balasubramanian[3], and Peter Sterling[1,*]
1 Department of Neuroscience University of Pennsylvania Philadelphia, Pennsylvania 19104
2 Department of Molecular Biology Princeton University Princeton, New Jersey 08544
3 Department of Physics University of Pennsylvania Philadelphia, Pennsylvania 19104

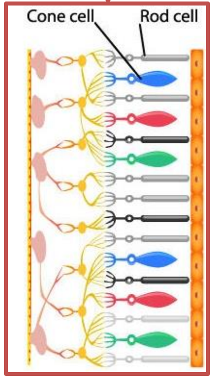
Summary

In the classic "What the frog's eye tells the frog's brain," Lettvin and colleagues [1] showed that different types of retinal ganglion cell send specific kinds of information. For example, one type responds best to a dark, convex form moving centripetally (a fly). Here we consider a complementary question: how *much* information does the retina send and how is it apportioned among different cell types? Recording from guinea pig retina on a multi-electrode array and presenting various types of motion in natural scenes, we measured information rates for seven types of ganglion cell. Mean rates varied across cell types (6–13 bits · s^{-1}) more than across stimuli. Sluggish cells transmitted information at lower rates than brisk cells, but because of trade-offs between noise and temporal correlation, all types had the same coding efficiency. Calculating the proportions of each cell type from receptive field size and coverage factor, we conclude (assuming independence) that the approximately 10^5 ganglion cells transmit on the order of 875,000 bits · s^{-1}. Because sluggish cells are equally efficient but more numerous, they account for most of the information. With approximately 10^6 ganglion cells, the human retina would transmit data at roughly the rate of an Ethernet connection.

NIH-PA Author Manuscript

https://x.com/docmilanfar/status/1914529429445075157





Pinhole Camera – Toy Example

Retina @ eye ~~ Image sensor @ camera

Add light-sensitive 'screen' in front of an object

Lights reflects onto the object & hits the screen

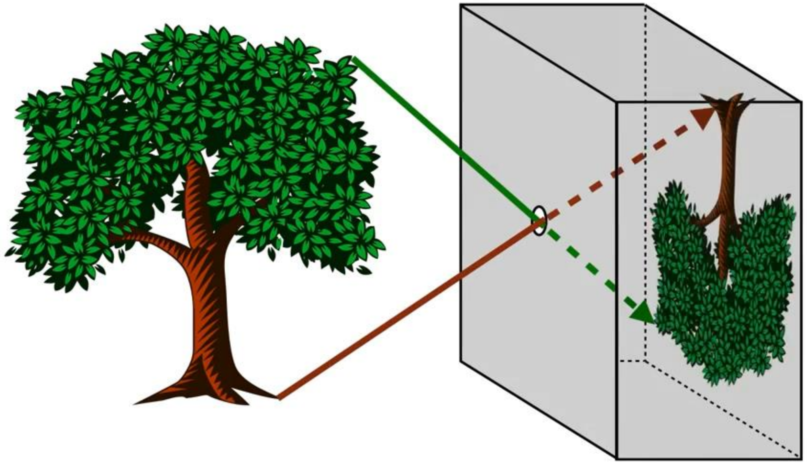
Problem: ∀ object point: Emits light in **all** directions

Image **washed-out** ☹

Sun **washes out++** ☹

Solution: Box with a **hole** → **Filters** *most* light rays

Result: **Crisp** ☺ (but **flipped** ☹) image



Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Real Example

‘Programmatically’

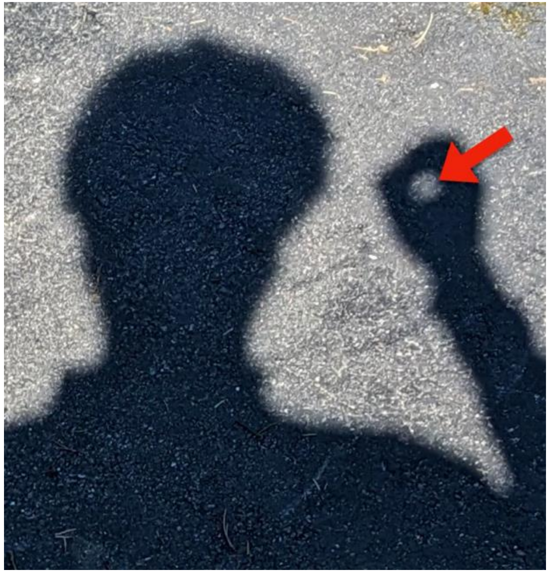
Flip Vertical

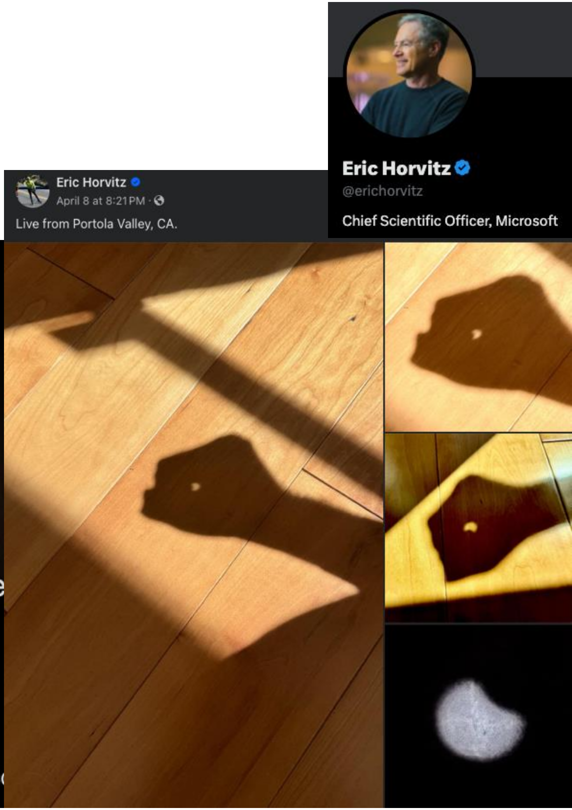
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Real Example

https://youtu.be/gvzpu0Q9RTU

Pinhole Camera – Real Example





Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

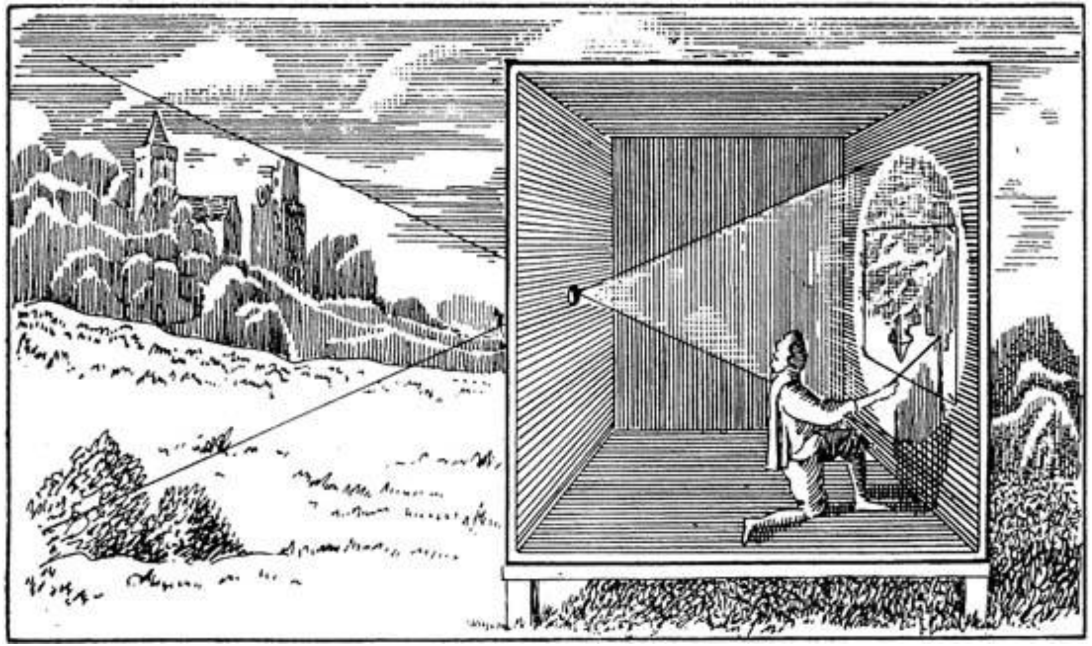
Pinhole Camera – Real Example

Normal
Sun

Solar
Eclipse

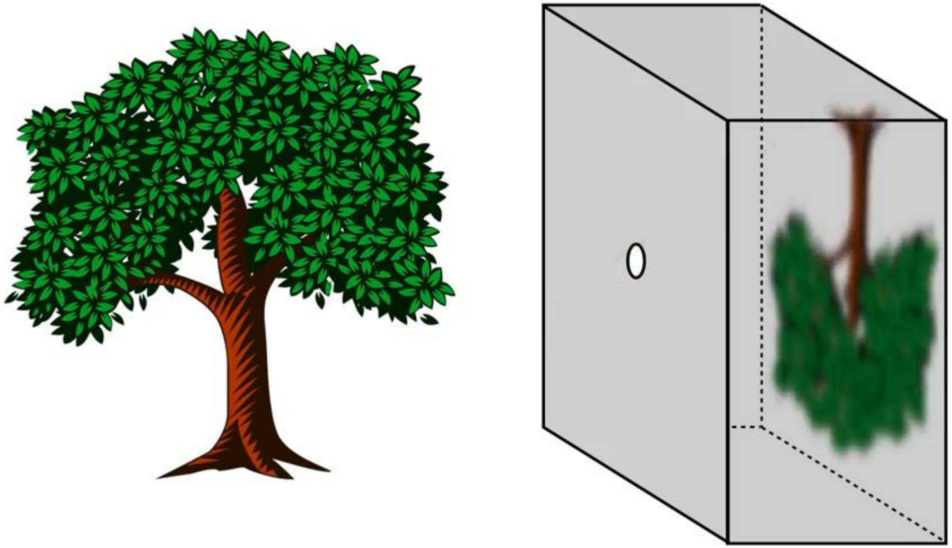
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Camera Obscura



3D → 2D projection
with a human in the loop

Pinhole Camera – Aperture



The hole (*aperture*) lets a narrow beam of light through

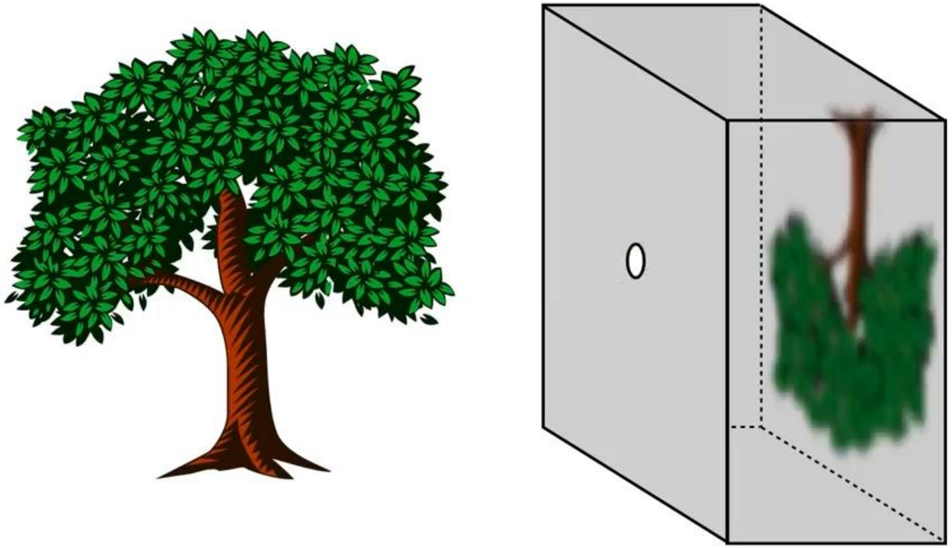
This creates a little *spot of color* on the screen
(that might also *overlap*)

Adds together many overlapping color spots

Result: ***Blurry*** image ☹

Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Aperture Size – Blur

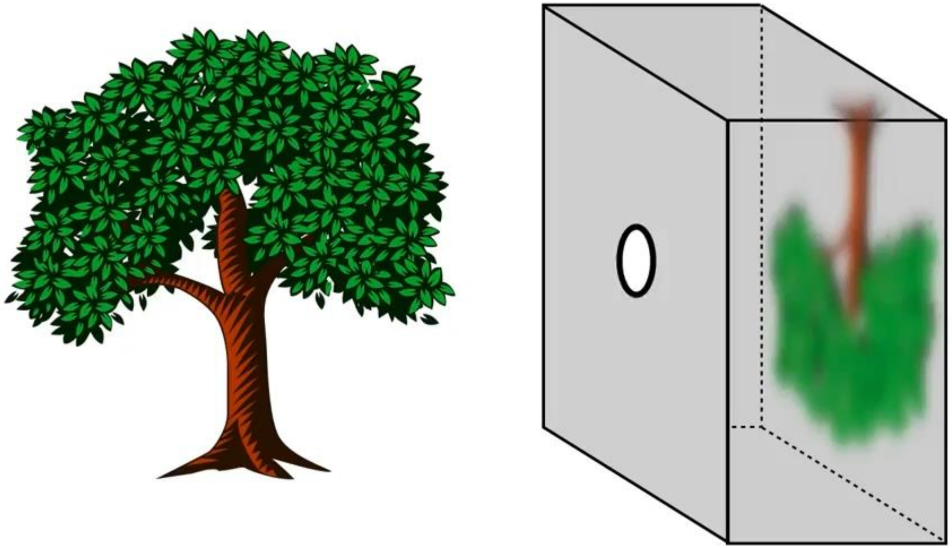


↑ Apperture size
↑ Size of color spots
↑ Blurriness
↓ Sharpness

↓ Apperture size
↓ Size of color spots
↓ Blurriness
↑ Sharpness

Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Aperture Size – Brightness



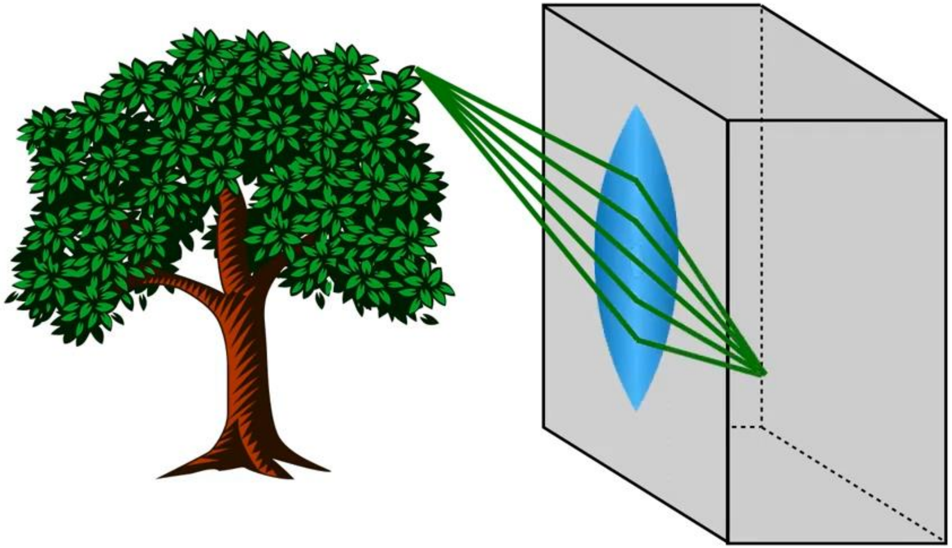
↑ Apperture size
↓ Sharpness
↑ Brightness

Trade off!

↓ Apperture size
↑ Sharpness
↓ Brightness

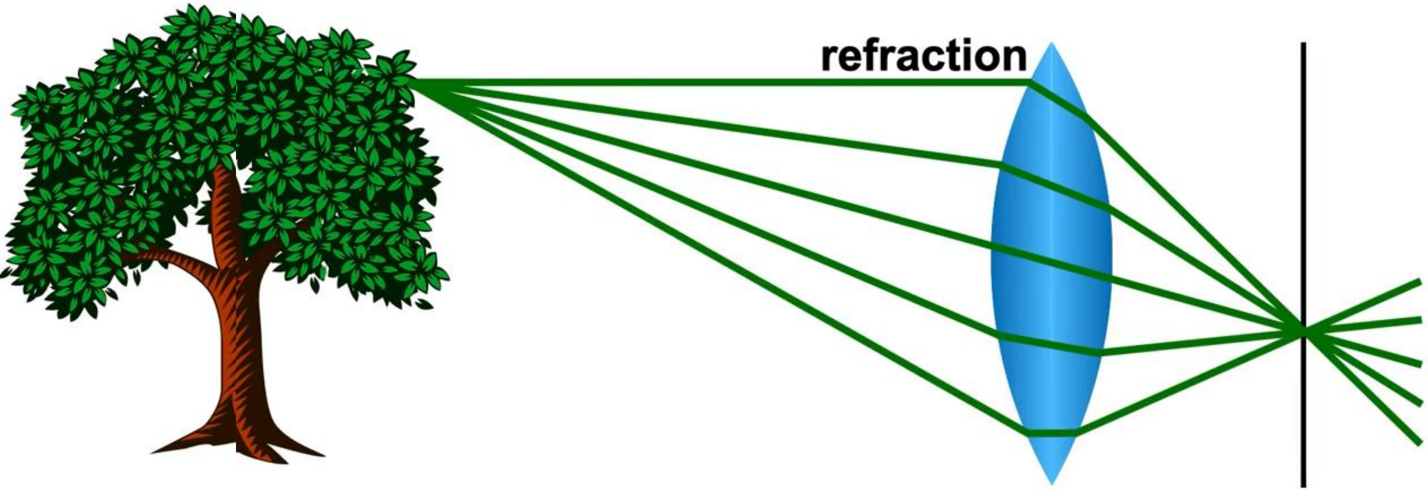
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Capture more Light



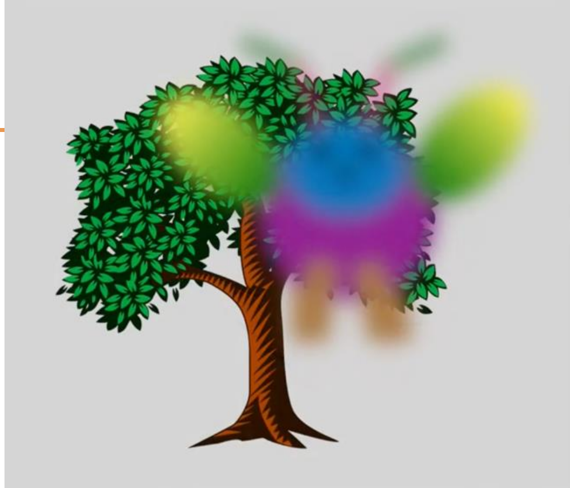
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

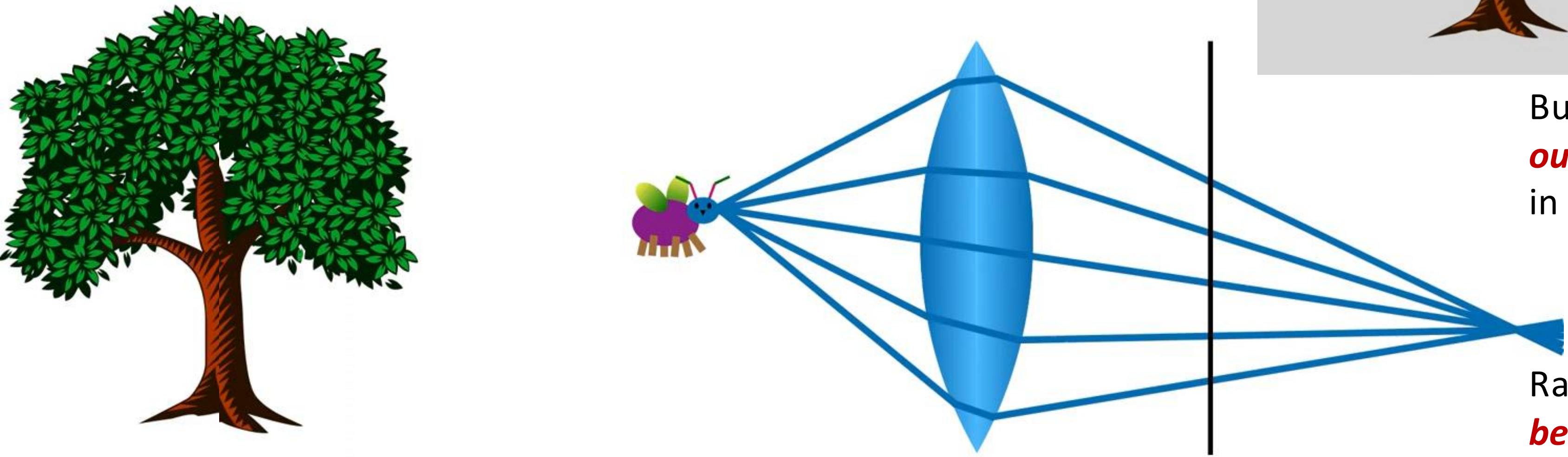
Pinhole Camera – Capture more Light



Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Focus



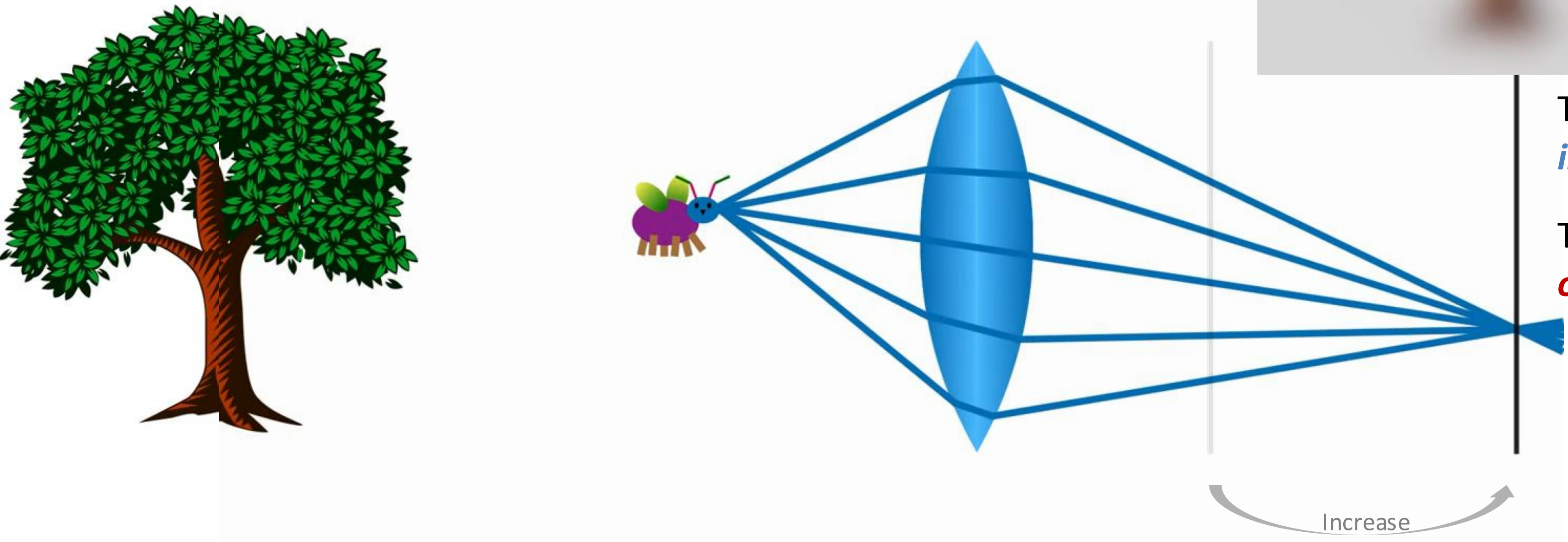


Bug appears ***out of focus*** in the image

Rays converge ***behind*** the image plane

Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Focus

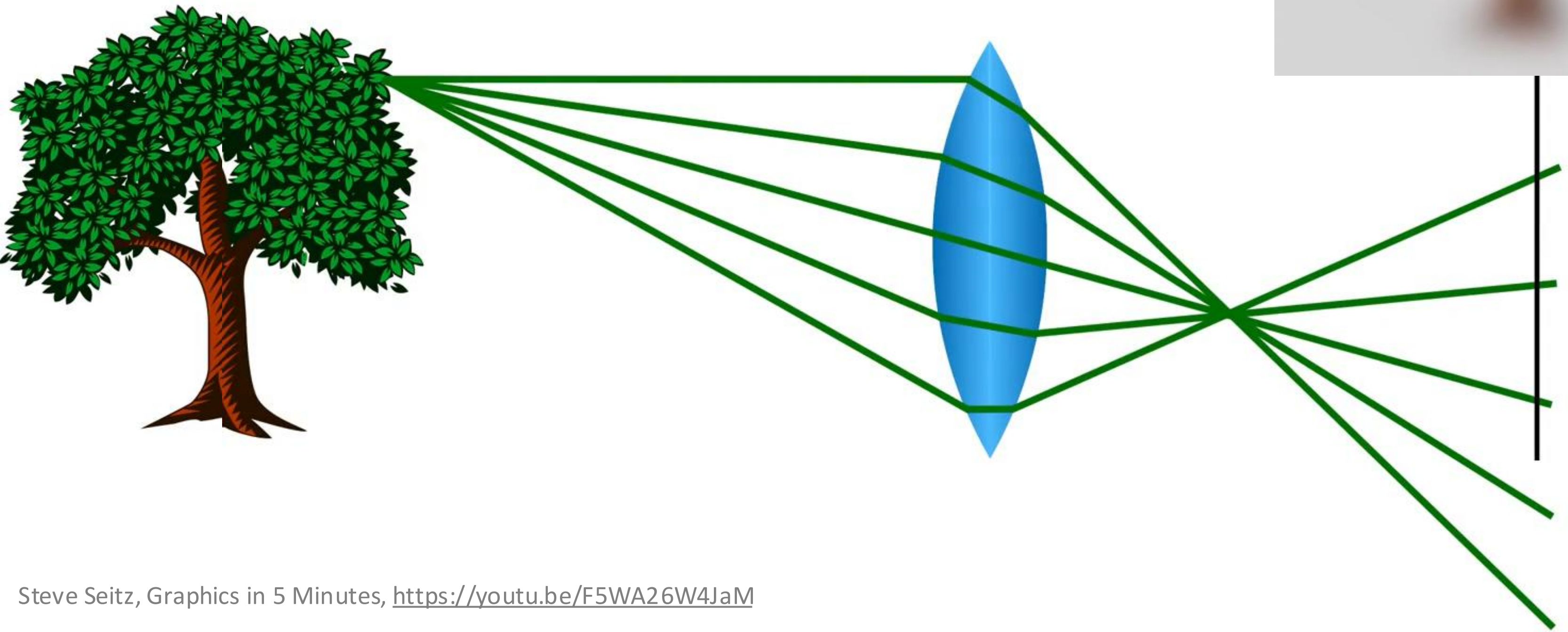


The bug is
in focus

The tree is
out of focus

Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Focus



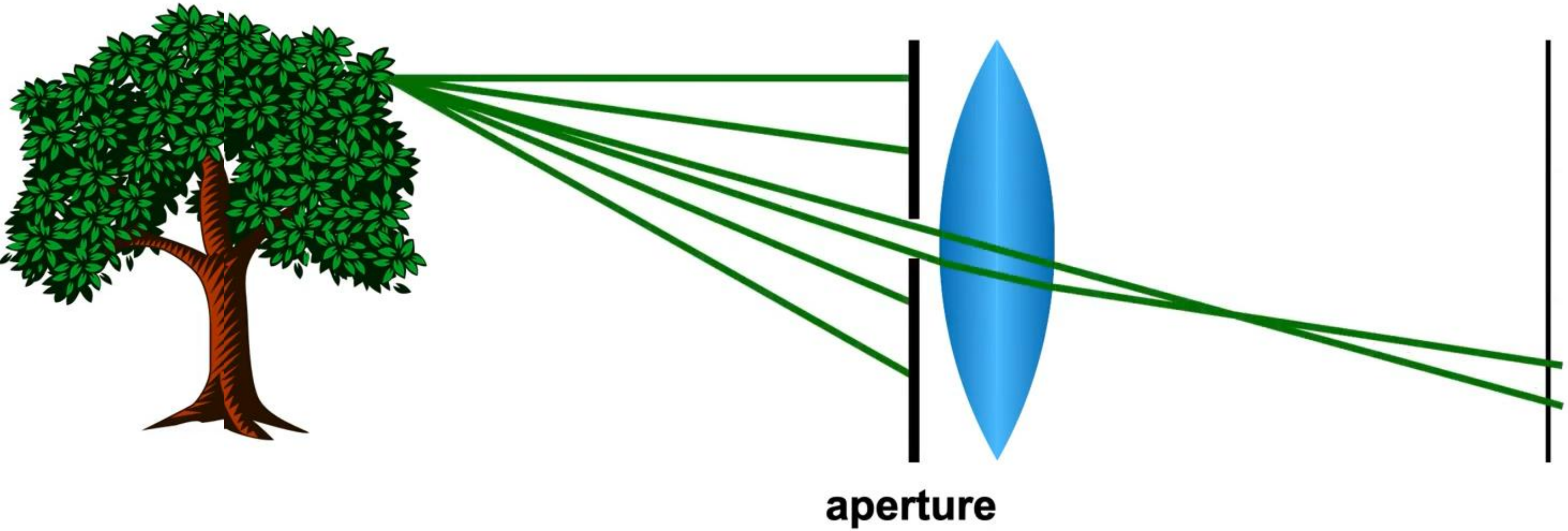
The bug is *in focus*

The tree is ***out of focus***

as the rays converge far in ***front*** of image plane

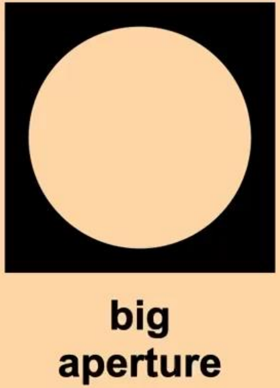
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

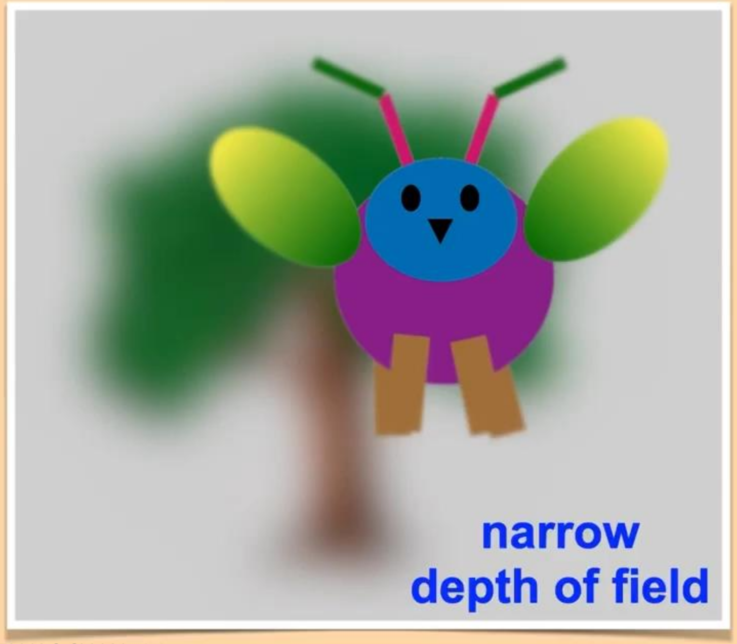
Pinhole Camera – Focus



Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Focus



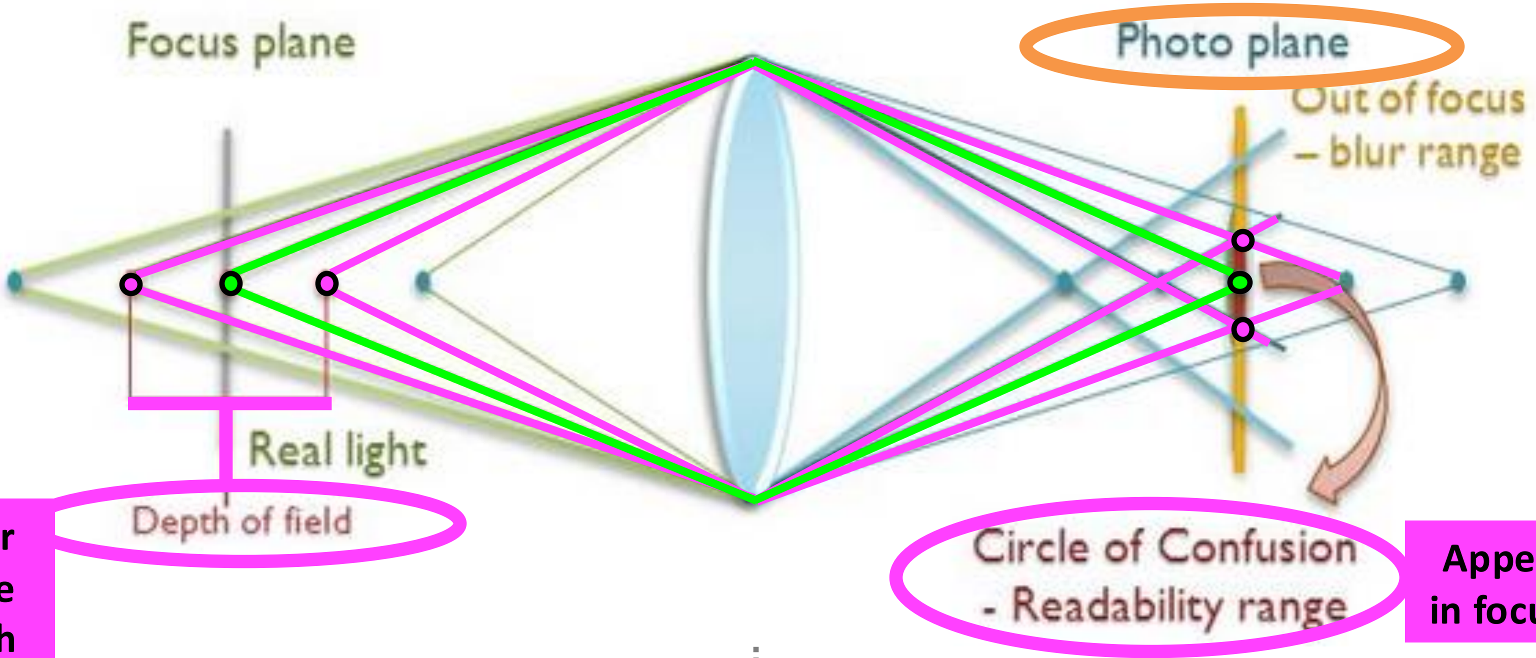


↑ Apperture
↓ Depth of Field

↓ Apperture
↑ Depth of Field

Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Aperture Adjustment – Depth of Field



Whatever lies inside the Depth of Field

Appears to be in focus (sharp)

3D World

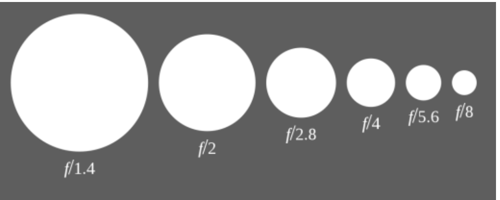
Inside Camera

Aperture Adjustment – Depth of Field

Disclaimer: The f here is just a dummy symbol. It should not be mistaken with the 'focal length f' defined in later slides

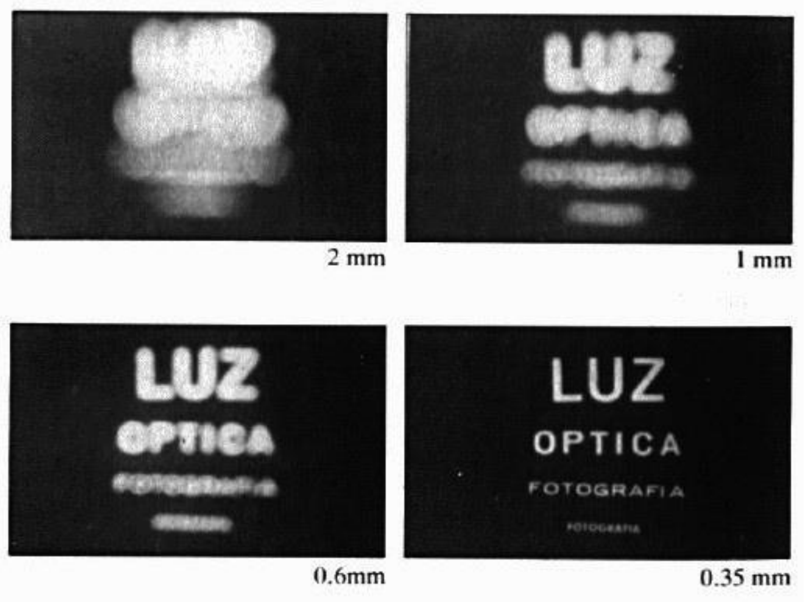
Aperture wide open → f/1.8 f/2.8 f/4.0

f/5.6 f/16 f/22 ← 'Narrow' Aperture



https://en.wikipedia.org/wiki/F-number

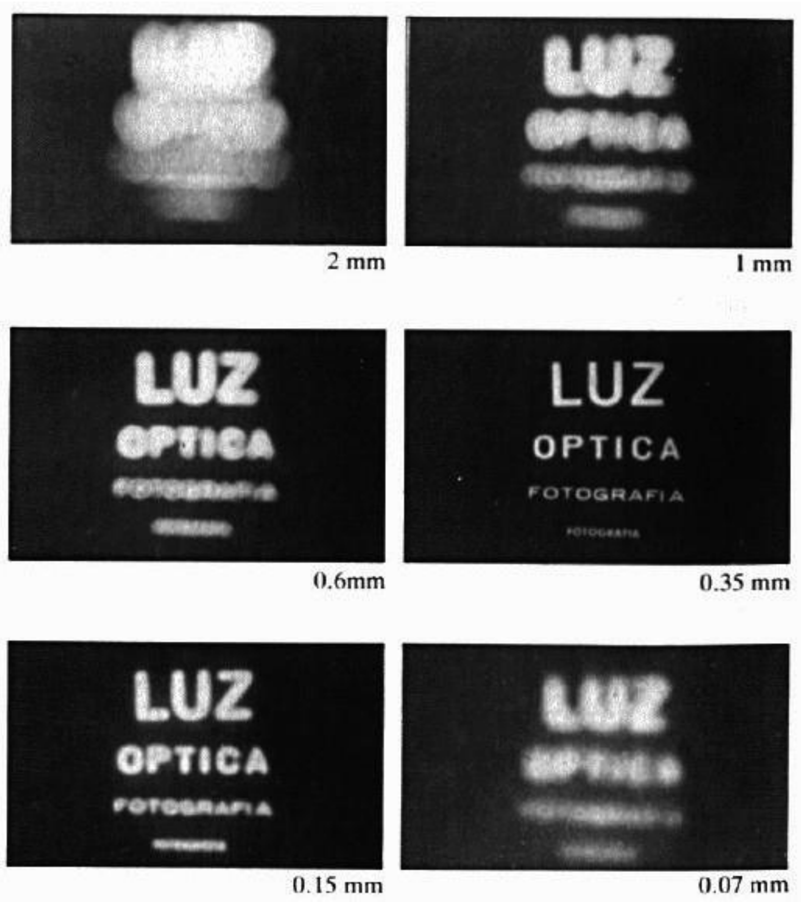
Aperture Shrinking



Why not make the aperture

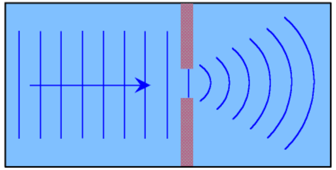
as small as possible?

Aperture Shrinking

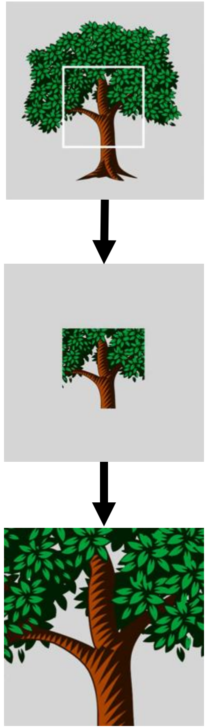


Less light gets through →

Diffraction effects →



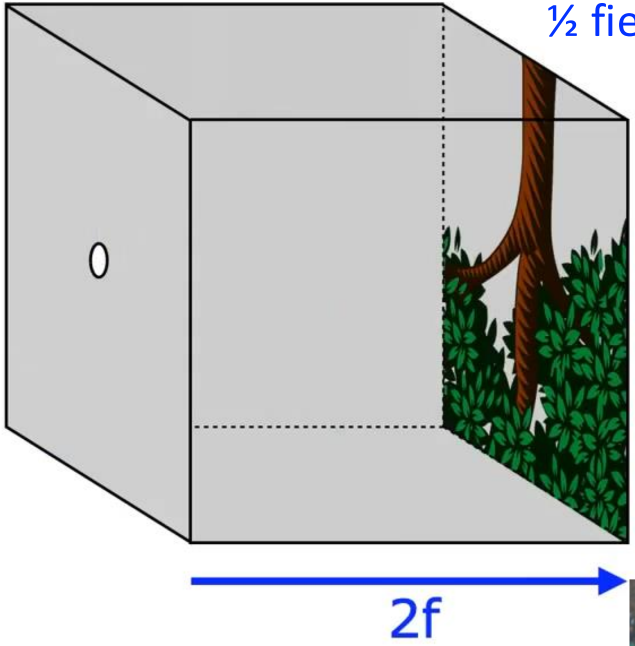
Pinhole Camera – Focal Length



2x magnification

2x magnification

½ field of view

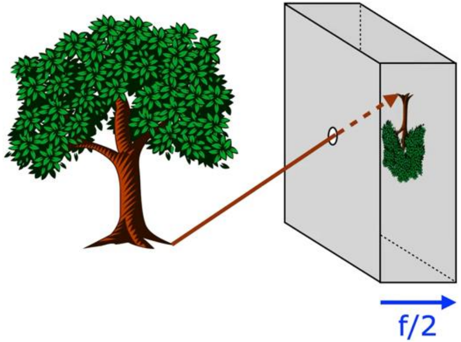


Cannot physically grow camera's body to double f to $2f$

Tele-photo lenses: Image plane far away from optical center

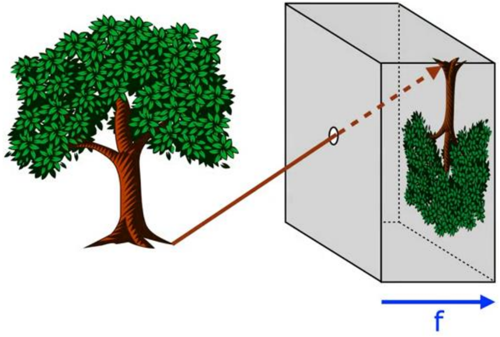
Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Focal Length

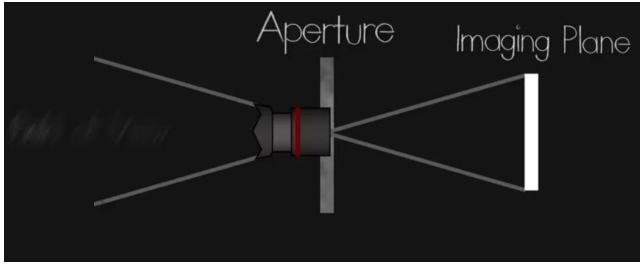


↓ Focal length
↓ Object size (shrinking)
↑ Field of view (more things fit the image)

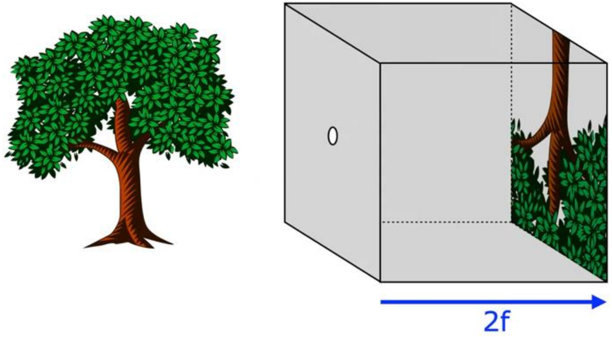
Steve Seitz, Graphics in 5 Minutes
https://youtu.be/F5WA26W4JaM



Impossible to move the image plain!
'Moveable' lens & 'bend' light rays & change focus



Mastering the Art of The Dolly Zoom
https://youtu.be/Amj6RiGiTOE



↑ Focal length
↑ Object size (magnification)
↓ Field of view (less things fit the image)

Outline

- Pinhole Camera
- Geometric Image Formation
- Photometric Image Formation
- Image Representation

Pinhole Camera – Projection Matrix

Camera Coordinate Frame:

- Origin $\boldsymbol{O}$ at camera center (hole)
- Z axis perpendicular to Image Plane I
  (towards outside the camera)
- X and Y axes *parallel* to Image Plane
- Projection plane is the plane $Z = -f$ (*)
  $Z = +f$ (**)

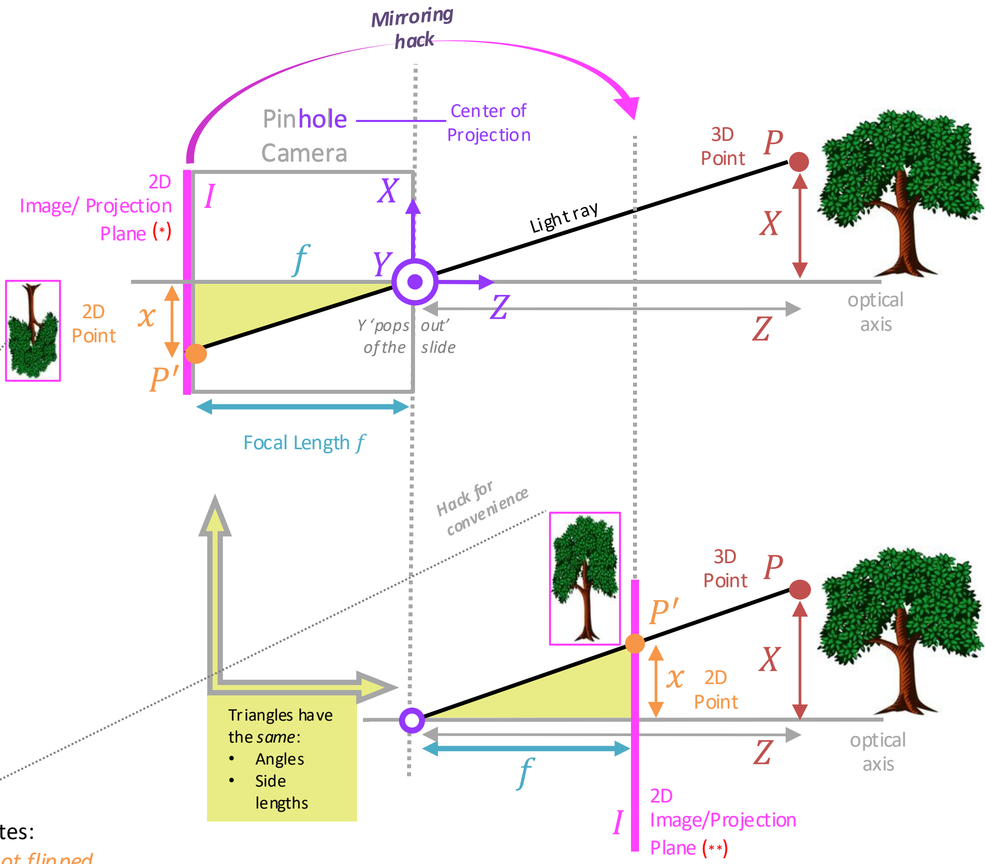
3D point $\xrightarrow{\text{Projection}}$ **2D point**

minus because it is flipped both horiz and vert

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad P' = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix} \xrightarrow{\text{mirroring hack}} \begin{bmatrix} f\dfrac{X}{Z} \\ f\dfrac{Y}{Z} \end{bmatrix}$$

Minus sign denotes:
projected image is *flipped*
(horiz. & vertically)

Plus sign denotes:
projected image is *not flipped*
(see bottom-right plot in this slide)

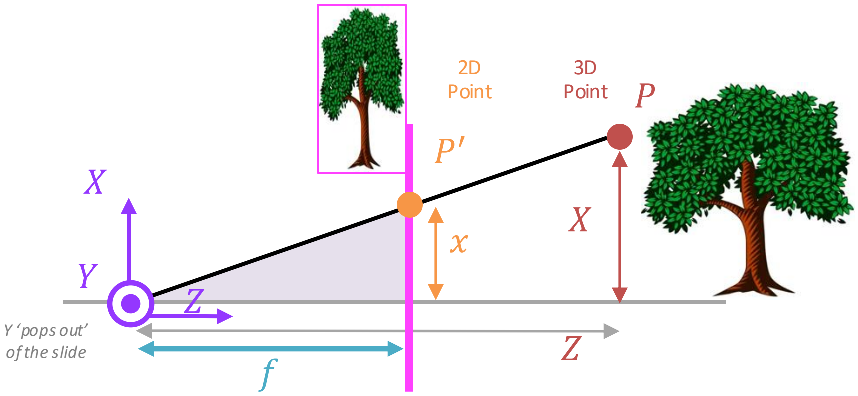


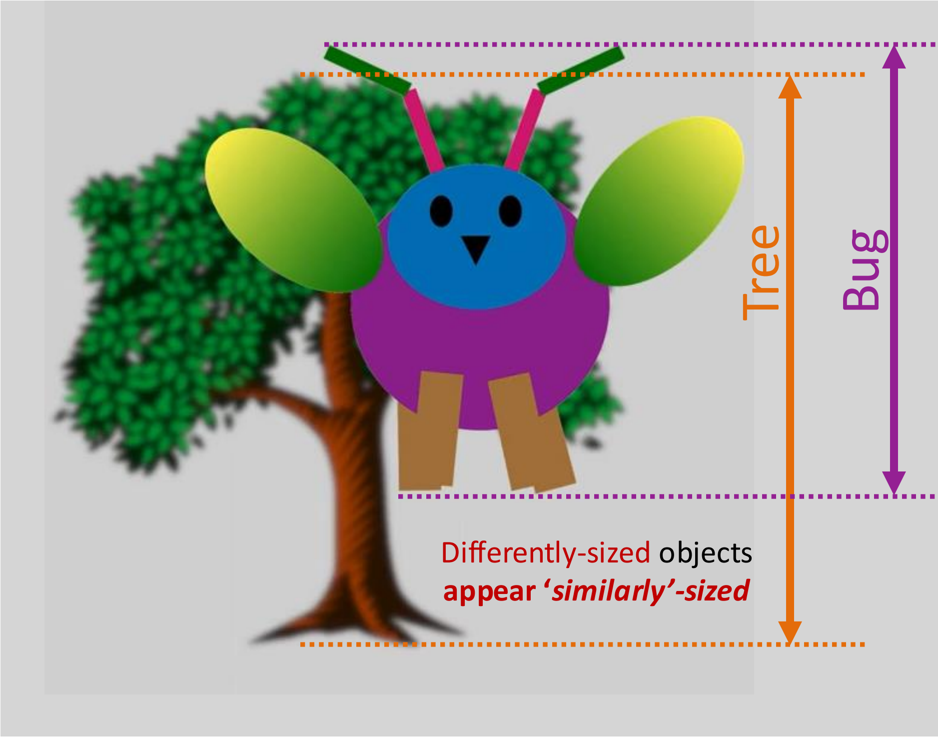
Pinhole Camera – Object Size

Scaled by $\frac{1}{Z}$

3D point $\xrightarrow{\text{Projection}}$ **2D point**

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad P' = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{bmatrix}$$

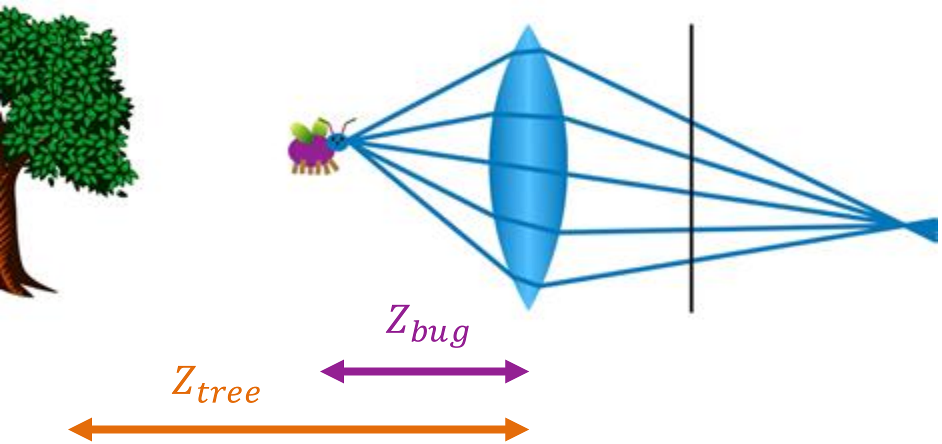


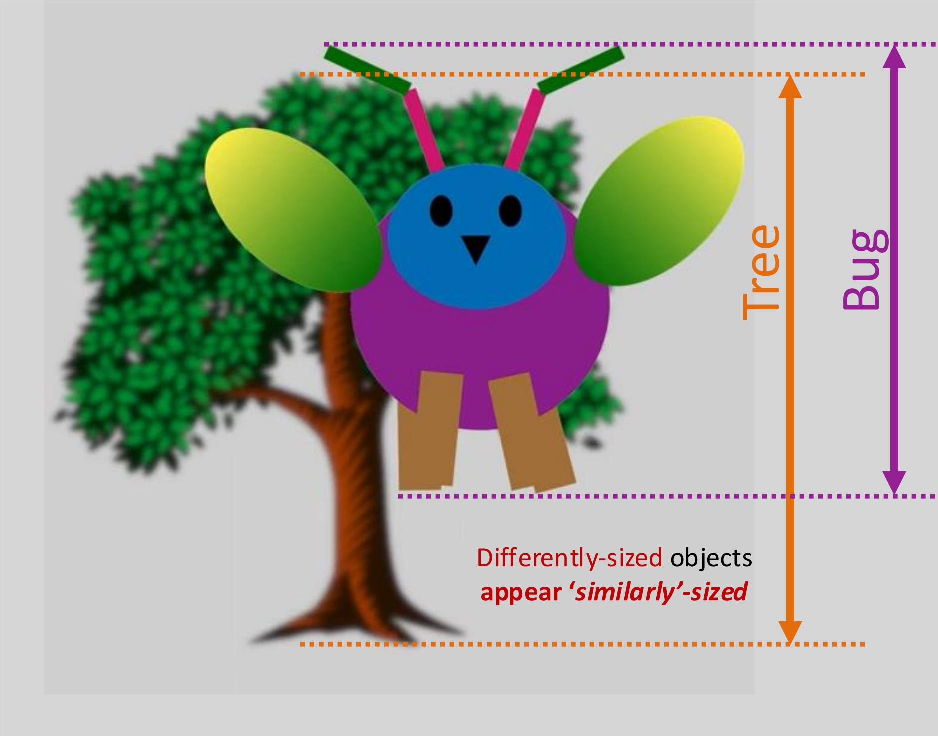


Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Pinhole Camera – Object Size

Scaled **by** $\frac{1}{Z}$



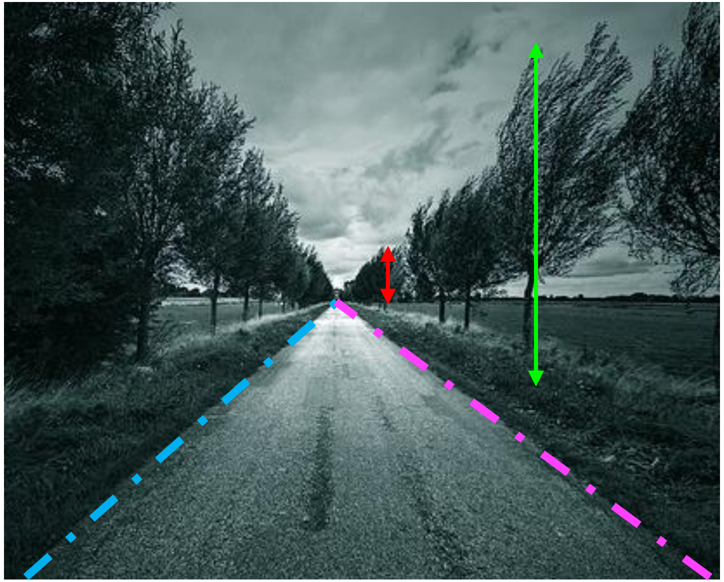


Steve Seitz, Graphics in 5 Minutes, https://youtu.be/F5WA26W4JaM

Projective Geometry

Projective transformation preserves:

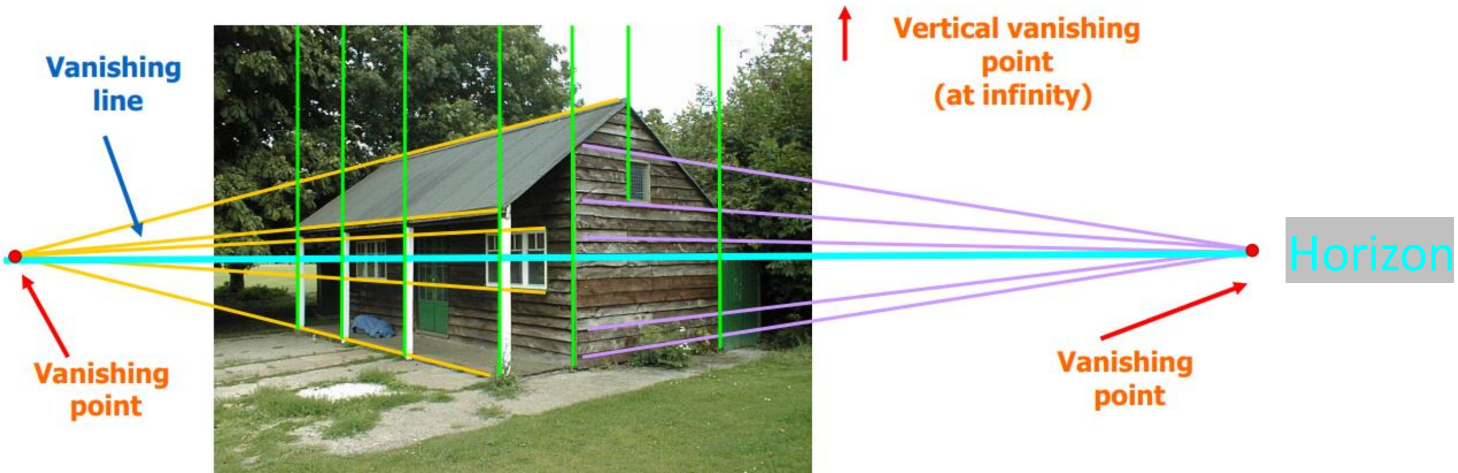
- Collinearity of points (straight lines remain straight)
- **Nothing else!** Changes:
    - **Parallelism:** Parallel lines in 3D, after projection **meet** @ a **vanishing point** in 2D
    - **Length:** Same-height Trees: far-away *VS* nearby
      appear: (small) (big)
    - **Angles:** Depend on viewpoint



Vanishing Points & Lines

Parallel **lines** 'converge' @ **vanishing point**

Can lie both inside & outside the image



Most slides from
Davide Scaramuzza

Vanishing Points & Lines

[📹 Rob Hoeijmakers / robhoeij]
[📍 Amsterdam-Rijn kanaal]

https://x.com/Rainmaker1973/status/1927160448169783414

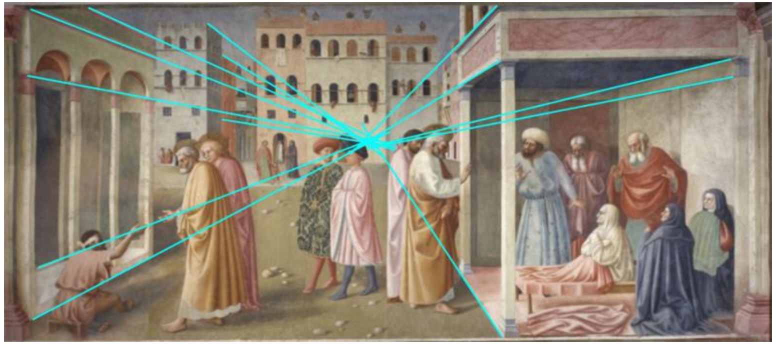
Parallel **lines** intersect @ **vanishing point**

Perspective in Art

Oblique Projection

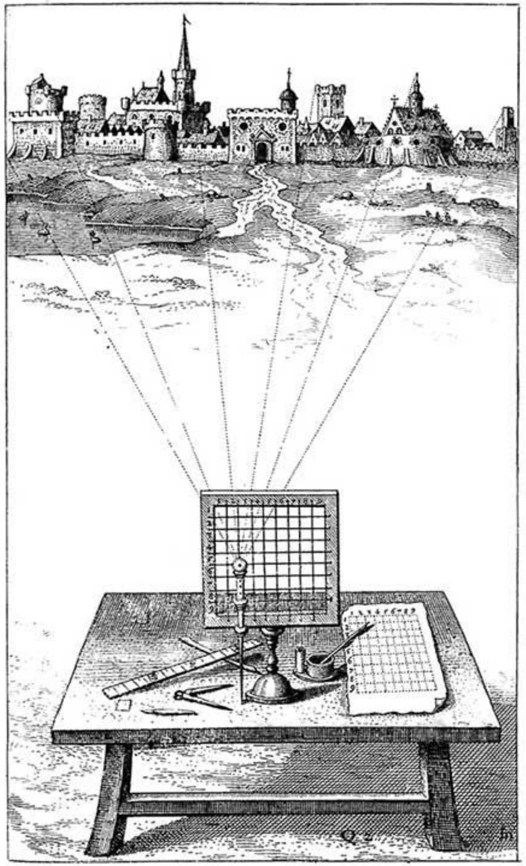
The Birth of Saint John the Baptist: Predella Panel
Giovanni di Paolo, 1454

One-eye Perspective



The Healing of the Cripple and Raising of Tabitha
Masolino, 1427

Perspective in Art



Robert Fludd's sighting grid (1617)

'Dimensionality Reduction' Machine (3D→2D)

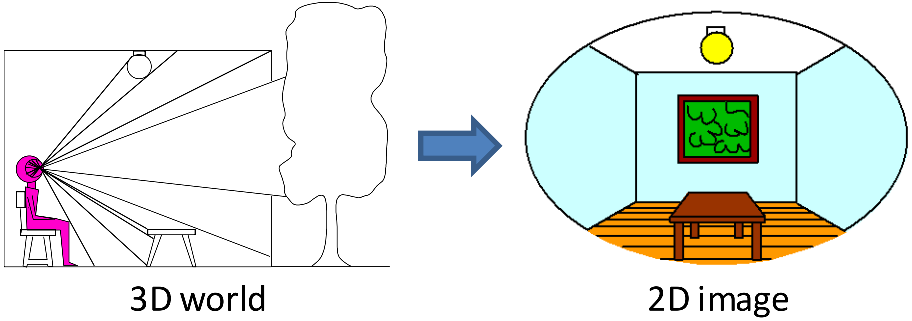
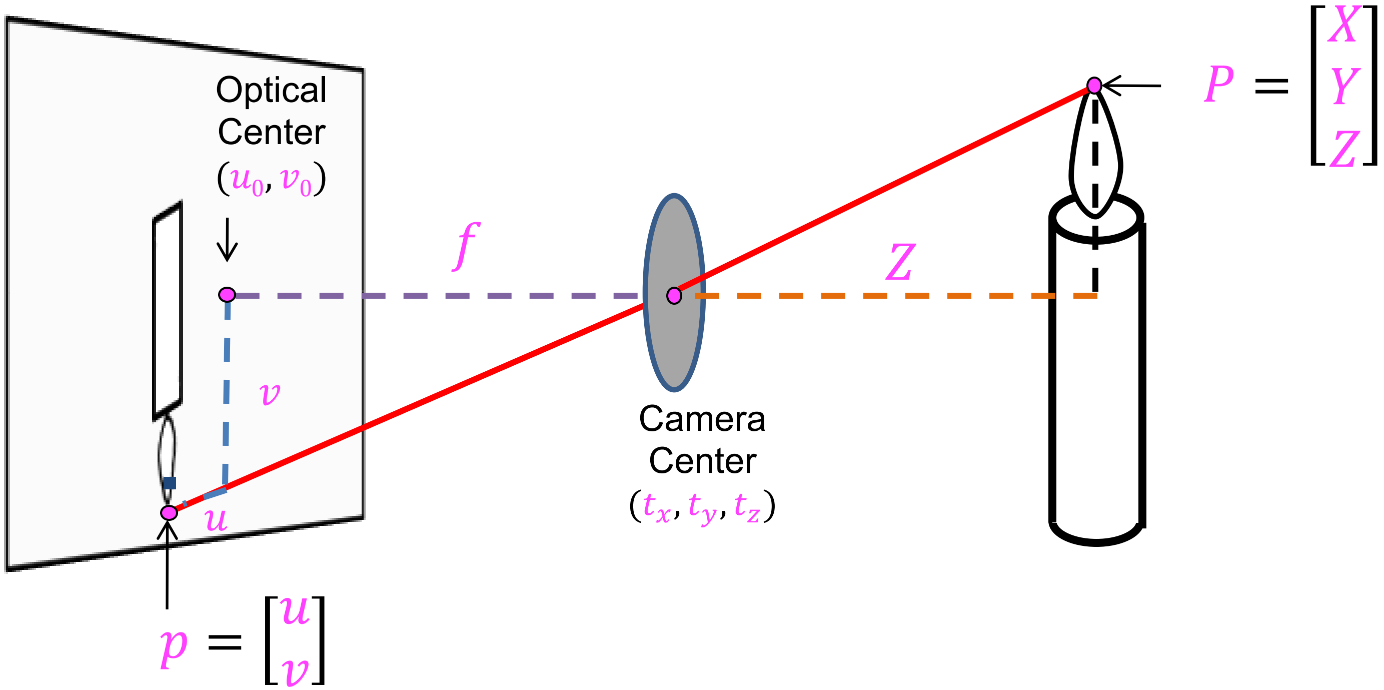


Figure: Stephen E. Palmer, 2002

Perspective in Art

Perspective in Art

Projection: 3D → 2D coordinates



Projection: 3D → 2D coordinates

Length & area size are not preserved!

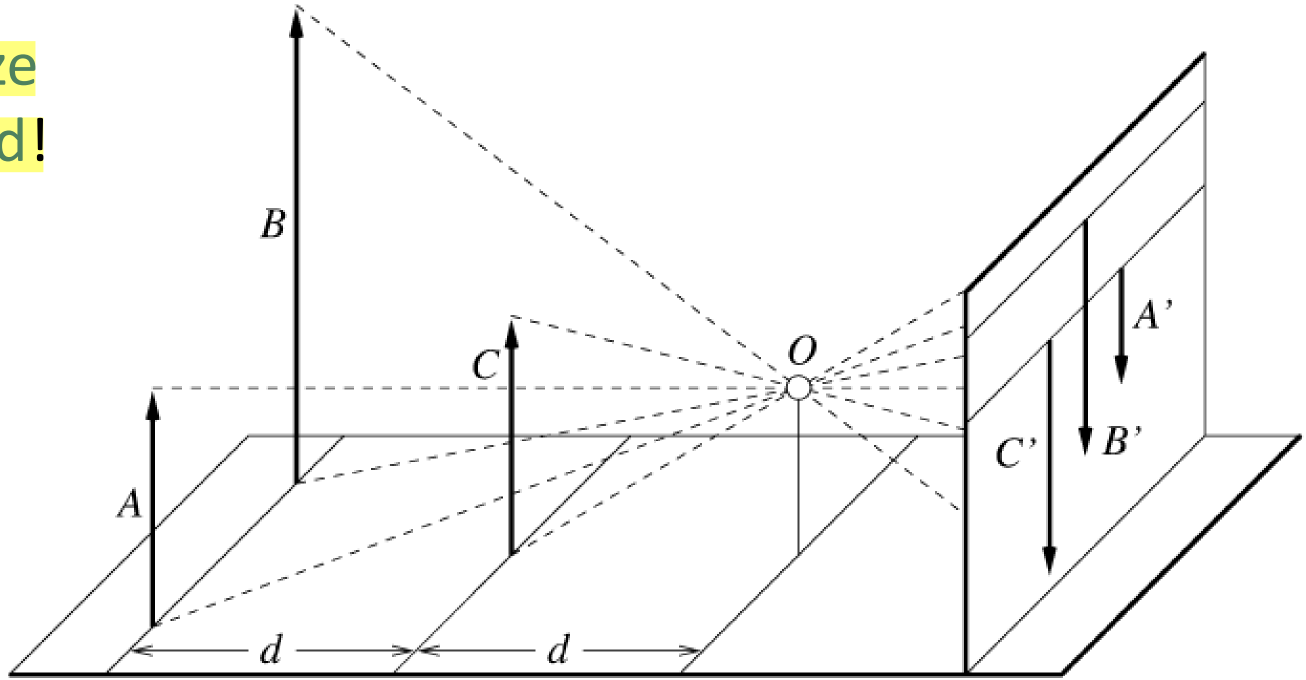


Figure by David Forsyth

Outline

- Pinhole Camera
- Geometric Image Formation
  - Homogeneous Coordinates
- Photometric Image Formation
- Image Representation

Homogeneous Coordinates

Cartesian Coordinates

- **Rotation** → Applied as matrix mult.
- **Translation** → Cannot

We need a new tool
to simplify math

Homogeneous Coordinates

- **Rotation** → Applied as matrix mult.
- **Translation** → Applied as matrix mult.

Cartesian → Homog. coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D (image) coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D (scene) coordinates

Homog. → Cartesian coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

2D (image) coordinates

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

3D (scene) coordinates

Homogeneous Coordinates

Invariant
to scaling

$$k\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

$$\boldsymbol{k \neq 0}$$

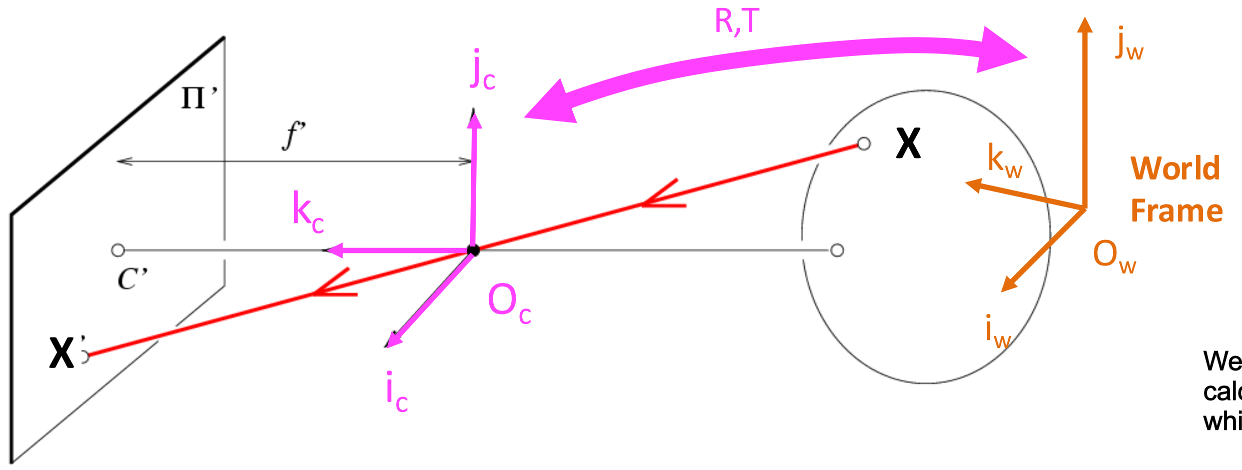
Homogeneous Coordinates

Cartesian Coordinates

A _**Point**_ in Cartesian Coordinates is …

a ___**Ray**___ in Homogeneous Coordinates

3D→2D Projection Matrix



Camera Frame
posed (R, T) w.r.t.
World Frame

We have two coordinate systems so our calcultion might changes a bit depending on which one we are considering

$$\mathbf{x} = \mathbf{K}\underbrace{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}}_{\text{Extrinsic Matrix}}\mathbf{X}$$

Intrinsic Matrix

Extrinsic Matrix

O_w: Center of World Frame
O_c: Center of Camera Frame

X: 3D point in World Coordinates: (X,Y,Z,1)
x: 2D Image Coordinates: (u,v,1), up to scale w

K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)

Note: different books use different notation!

Slide Credit:
S. Saverese

3D→2D Projection Matrix

Extrinsic Assumptions

- **Camera == World Frame**
- No rotation → $R = I$
- Camera at → $T = [0,0,0]$

Intrinsic Assumptions

- Unit aspect ratio → $f_x = f_y = f$
- Optical center at (0,0) → $u_0 = v_0 = 0$
- No skew → $s = 0$

The optical center is the hole

K tells us how the internals of the camera works

$$\mathbf{x} = \underbrace{\mathbf{K}\begin{bmatrix} \overset{\text{Rot}}{\mathbf{I}} & \overset{\text{Transl}}{\mathbf{0}} \end{bmatrix}}_{\text{Projection Matrix}}\mathbf{X}$$

$$\Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \underbrace{\begin{matrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{matrix}}_{K} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Cartesian Coord.
(after dividing with last element)

$$u = x\frac{f}{z}$$

$$v = y\frac{f}{z}$$

Slide Credit:
S. Saverese

3D→2D Projection Matrix

Extrinsic Assumptions

- Camera == World Frame
- No rotation → $R = I$
- Camera at → $T = [0,0,0]$

Intrinsic Assumptions

- Unit aspect ratio → $f_x = f_y = f$
- ~~Optical center at (0,0) → $u_0 = v_0 = 0$~~
- No skew → $s = 0$

$$\mathbf{x} = \underbrace{\mathbf{K}[\mathbf{I} \quad \mathbf{0}]}_{\text{Projection Matrix}}\mathbf{X} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

K

Slide Credit:
S. Saverese

3D→2D Projection Matrix

Extrinsic Assumptions

- **Camera == World Frame**
- No rotation → $R = I$
- Camera at → $T = [0,0,0]$

Intrinsic Assumptions

- ~~Unit aspect ratio~~ → $\cancel{f_x = f_y = f}$
- ~~Optical center at (0,0)~~ → $\cancel{u_0 = v_0 = 0}$
- No skew → $s = 0$

$$\mathbf{x} = \underbrace{\mathbf{K}[\mathbf{I} \quad \mathbf{0}]}_{\text{Projection Matrix}}\mathbf{X} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

K

Slide Credit:
S. Saverese

3D→2D Projection Matrix

Extrinsic Assumptions

- Camera == World Frame
- No rotation → $R = I$
- Camera at → $T = [0,0,0]$

Intrinsic Assumptions

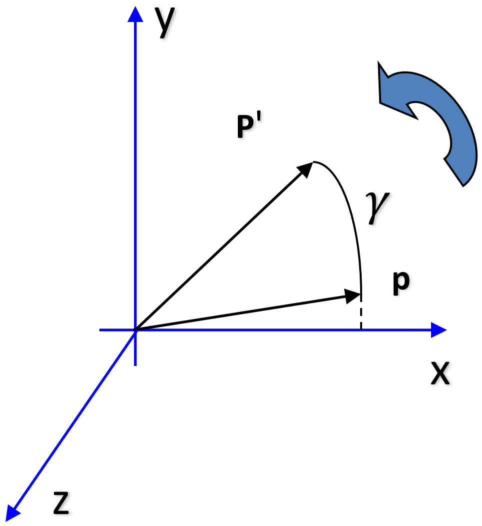
- ~~Unit aspect ratio~~ → $\cancel{f_x = f_y = f}$
- ~~Optical center at (0,0)~~ → $\cancel{u_0 = v_0 = 0}$
- ~~No skew~~ → $\cancel{s = 0}$

$$\mathbf{x} = \underbrace{\mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}}_{\text{Projection Matrix}}\mathbf{X} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

K

The most general form of Intrinsics K

3D Rotation of Points



Rotation around **coordinate axes** (counter-clockwise)

$\alpha \rightarrow$ Angle around axis-x
$\beta \rightarrow$ Angle around axis-y
$\gamma \rightarrow$ Angle around axis-z

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Each of R_x, R_y, R_z has 1 DoF
(number of free parameters)

Complex Rotations → Multiply R_x & R_y & R_z
(the order matters)

Slide Credit:
S. Saverese

Intrinsics & Extrinsics

Camera Frame
Rotated & Translated
w.r.t. World Frame

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{5} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{3 + 3} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Degrees of Freedom (DoF) → 5 + 6 = 11 DoFs in total

Intrinsics & Extrinsics

What if
no rotation?

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Intrinsics & Extrinsics

Identity Rotation →

$R = I$

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \quad \mathbf{t}]\mathbf{X}$$

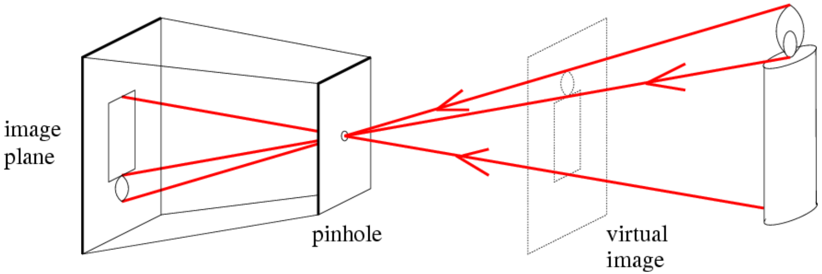
$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

What if
no rotation &
no translation?

→ The World & Camera Frames coincide → Only the Intrinsics suffice for projection

Quick Summary

- Pinhole Camera model
- Camera Matrices for Projection
- Homogeneous Coordinates



Intrinsics

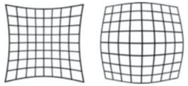
Extrinsics

$$\mathbf{x} = \mathbf{K}\underbrace{[\mathbf{R} \quad \mathbf{t}]}_{\text{Extrinsics}}\mathbf{X}$$

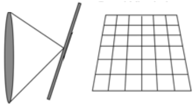
$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Intrinsics – Distorted Image

Front
'fisheye' camera
of FrodoBot
(IRLab, UvA)



Radial distortion



Tangential distortion

Camera is uncalibrated ☹

Need to: Calibrate camera → estimate intrinsics

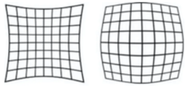
Undistort image via estimated intrinsics

Intrinsics – Distorted Image

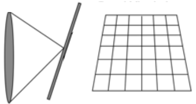
OpenCV tutorial on **Camera Calibration** & image **Undistortion**

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

$$Distortion\ coefs = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$



Radial distortion → $x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$ $y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$



Tangential distortion → $x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$ $y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$

Intrinsics – Undistorted Image – 0 Elements

```
dist_coeff = [-0.000, 0.0000, 0.00000, -0.00000, -0.00000]
```

Intrinsics – Undistorted Image – 1 Element

```
dist_coeff = [-0.217, 0.0000, 0.00000, -0.00000, -0.00000]
```

Intrinsics – Undistorted Image – 2 Elements

```
dist_coeff = [-0.217, 0.0537, 0.00000, -0.00000, -0.00000]
```

Intrinsics – Undistorted Image – 3 Elements

`dist_coeff = [-0.217, 0.0537, 0.00185, -0.00000, -0.00000]`

Intrinsics – Undistorted Image – 4 Elements

```
dist_coeff = [-0.217, 0.0537, 0.00185, -0.00210, -0.00000]
```

Intrinsics – Undistorted Image – 5 Elements

`dist_coeff = [-0.217, 0.0537, 0.00185, -0.00210, -0.00600]`

Intrinsics – Undistorted Image

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

`h,w = [1024, 576]`

Intrinsics – Undistorted Image

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image

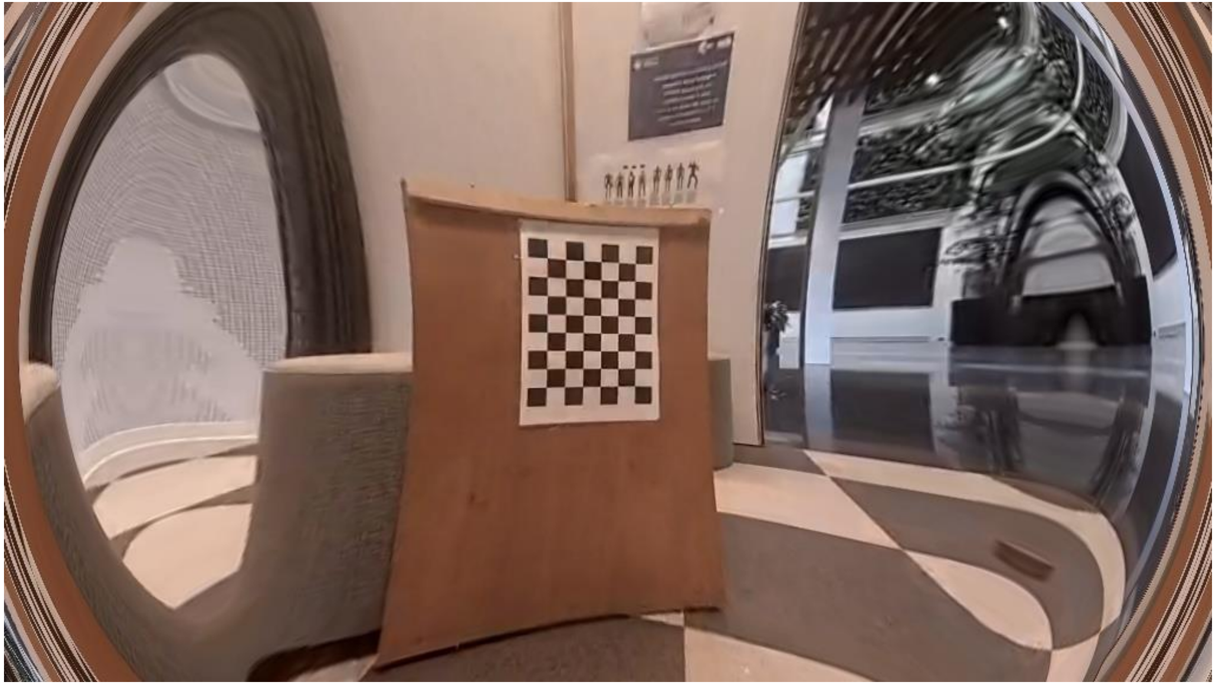
$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

Mess up: f_x, f_y, c_x, c_y = [407.860, 407.866, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image



$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

Mess up: f_x, f_y, c_x, c_y = [200.000, 200.000, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image

$$\text{camera matrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

Mess up: f_x, f_y, c_x, c_y = [300.000, 300.000, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

`h,w = [1024, 576]`

Mess up: f_x, f_y, c_x, c_y = [400.000, 400.000, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

`h,w = [1024, 576]`

Mess up: f_x, f_y, c_x, c_y = [600.000, 600.000, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

Mess up: f_x, f_y, c_x, c_y = [800.000, 800.000, 512.000, 288.000]

Original: f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – Undistorted Image - Skew

Skew

s = [000.0]

$camera\ matrix = \begin{bmatrix} f_x & \boxed{0} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$

h,w = [1024, 576]

f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image - Skew

Mess up:

Skew

s = [100.0]

camera matrix =

$$\begin{bmatrix} f_x & \boxed{S} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image - Skew

Mess up:

Skew

s = [200.0]

$$camera\ matrix = \begin{bmatrix} f_x & \boxed{s} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

h,w = [1024, 576]

f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

Intrinsics – ~~Un~~distorted Image - Skew

Mess up:

Skew

`s = [400.0]`

camera matrix $= \begin{bmatrix} f_x & \boxed{s} & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$

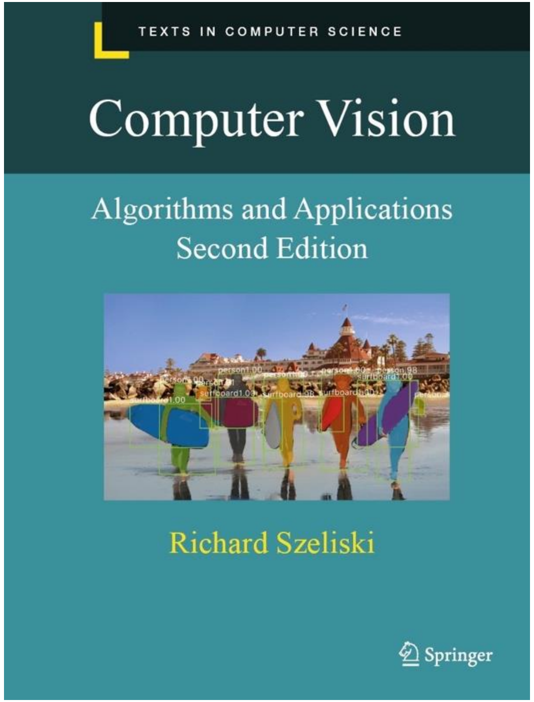
`h,w = [1024, 576]`

f_x, f_y, c_x, c_y = [407.860, 407.866, 533.301, 278.699]

TextBook – Sections

- 2.1.1
- 2.1.2
- 2.1.4

- 2.2
- 2.3.1
- 2.3.2



Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
    - Colour Systems

Image Formation

Scenes are complex

Image Formation

Four main factors influence image intensity values

- Illumination of scene
- Geometry of scene
- Reflectance of visible surfaces
- Camera view & optics

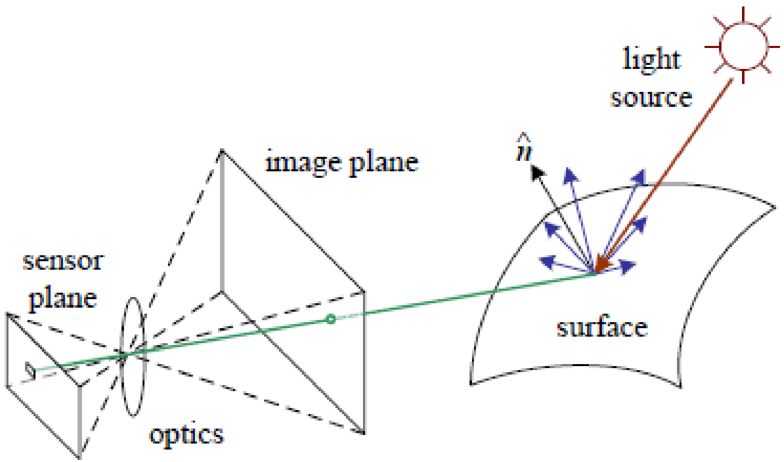


Image Formation – Photometry

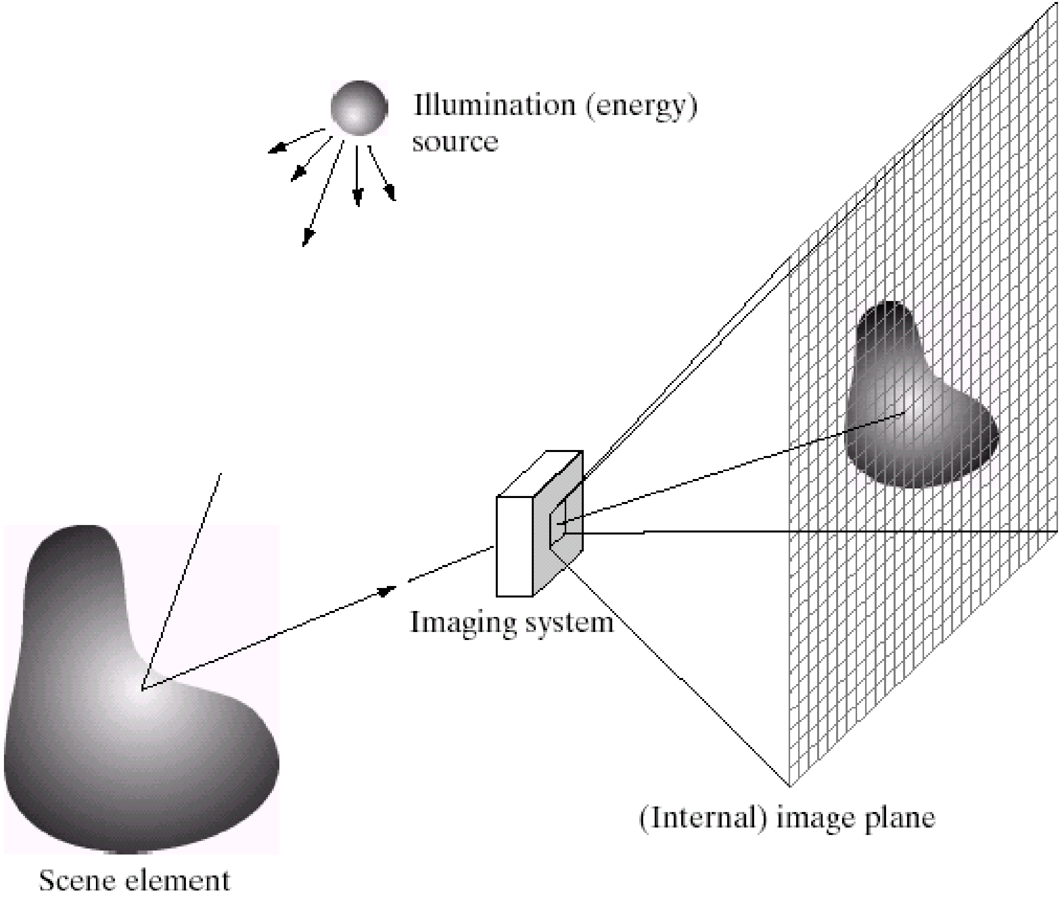
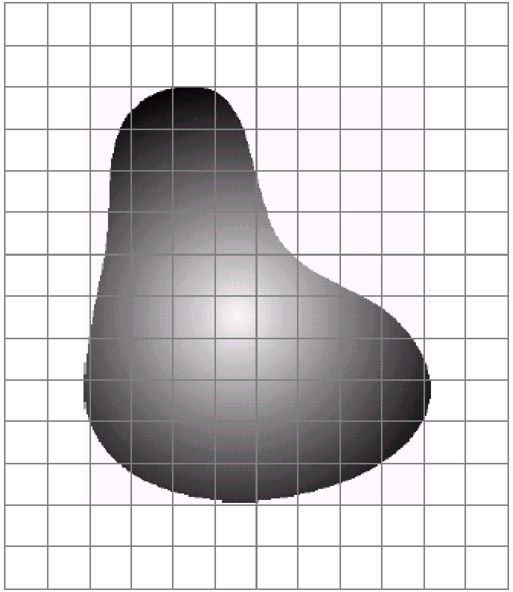
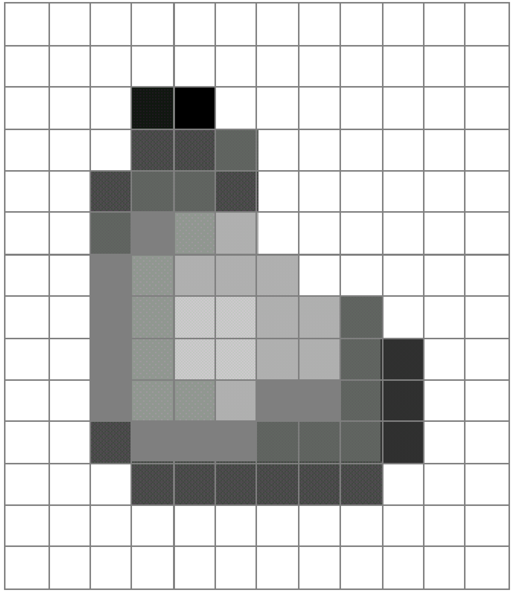


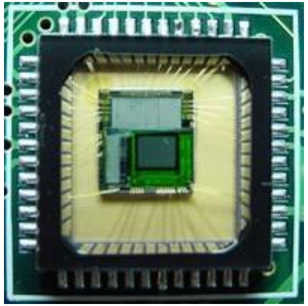
Image Formation – Sensor Array



Continuous image projected onto a sensor array



Result of image sampling & quantization

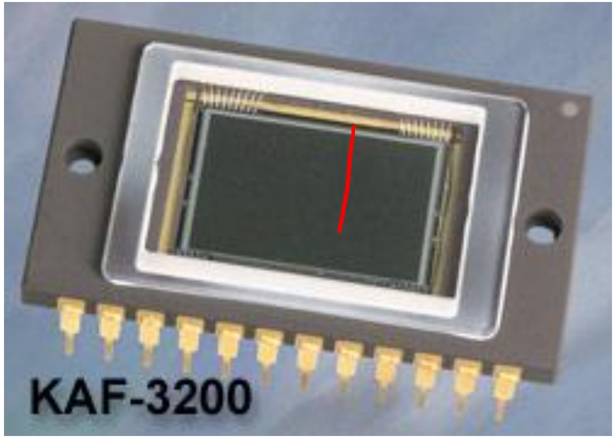


CCD/CMOS sensor

sampling → Spatial position

quantization → Numerical value

Image Formation – Sensor Array



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

Charge-Coupled Device (CCD)

- Converts continuous image → digital image
- Contains an array of light-sensor units
- Converts photons → electric charges accumulated in each sensor unit

Image Formation – Sensor – Imperfections

Low light

Dynamic Range

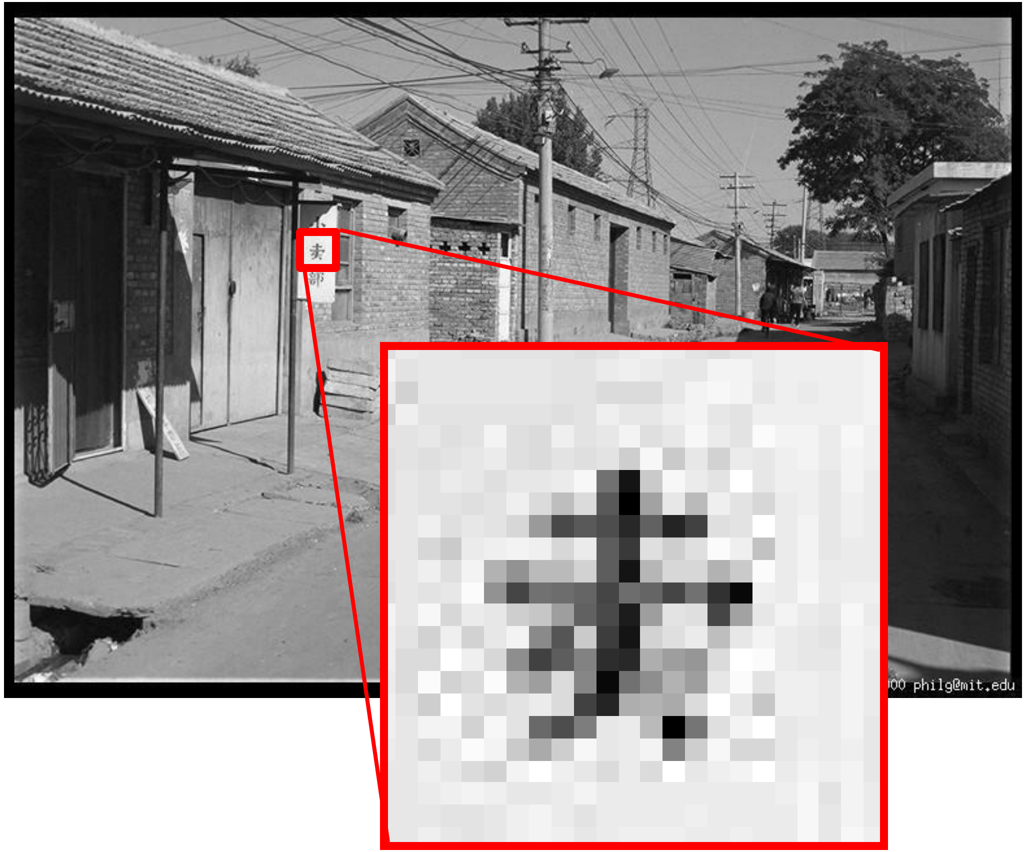
CCD overflow

Rolling Shutter

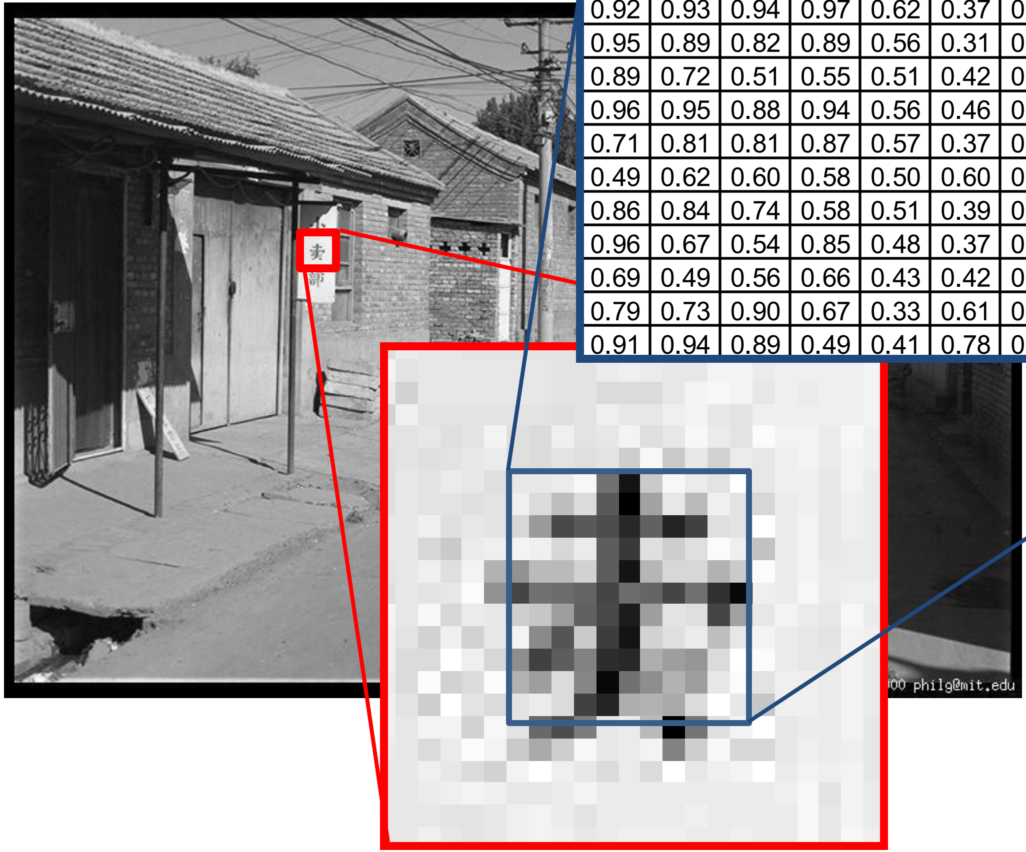
Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
    - Colour Systems

Digital Image – Pixel Matrix

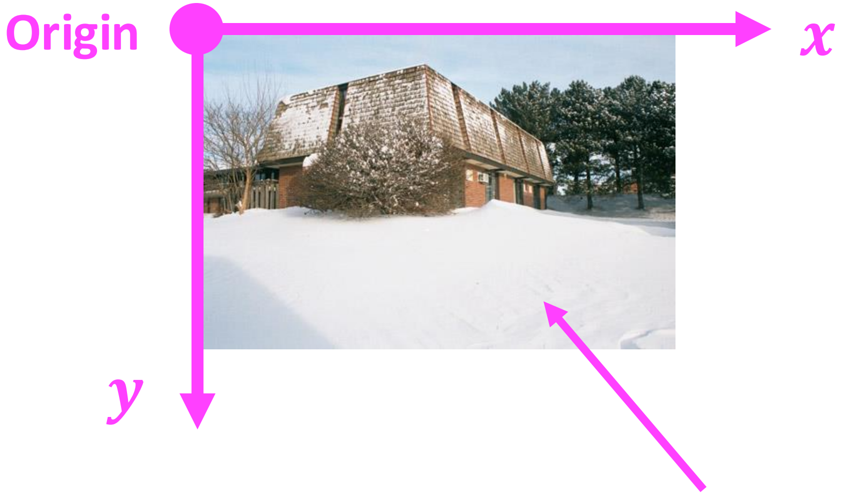


Digital Image – Pixel Matrix



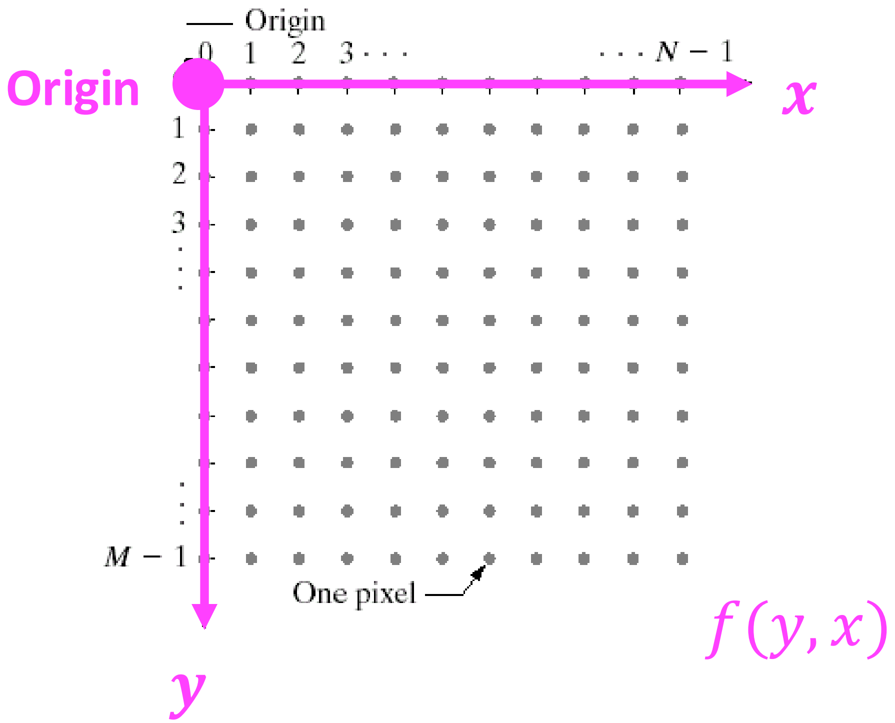
0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	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

Digital image – Fundamentals



- **Image** == a **function** of spatial coordinates $f(y, x)$
- **Spatial coordinates**: (x, y) for 2D case – e.g. photograph
  (x, y, t) for video
  (x, y, z) for 3D case – e.g. CT scans
- The function f may represent intensity (for greyscale images) or color (for color images) or other associated values

Digital image – Coordinate System



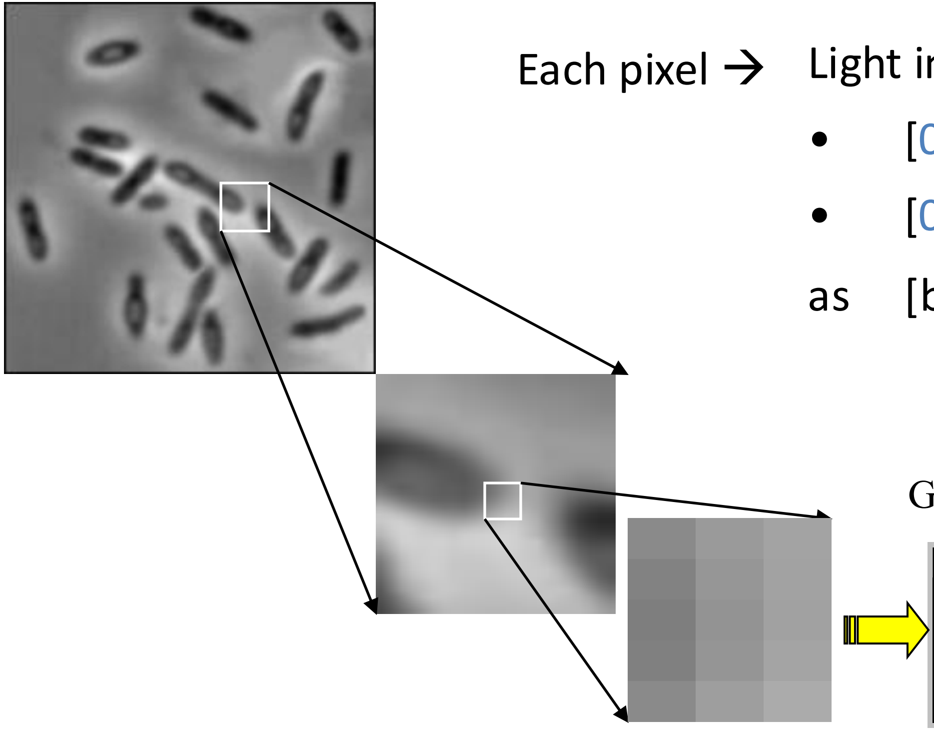
Regular grid of sampling points

Sampling point called a **Pixel**
(PICTure ELement)

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

Digital Image – Grayscale

Intensity- or **Grayscale-** image



Each pixel → Light intensity in the range of

- [0.0, ..., 1.0] (`float`)
- [0, ..., 255] (`uint8`)

as [black, ..., white]

Gray scale values

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

Digital Image – Spatial Resolution

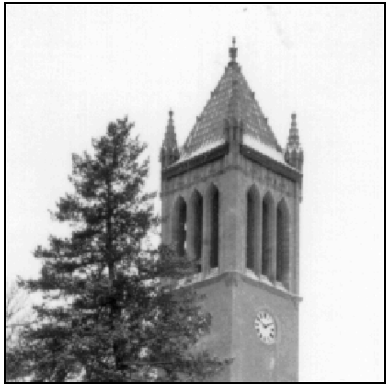
Images from
Rafael C. Gonzalez and Richard E. Wood,
Digital Image Processing, 2nd Edition

Digital Image – Spatial Resolution

… resampled to 1024x1024

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

Digital Image – Quantization Levels



256 levels (8-bit)

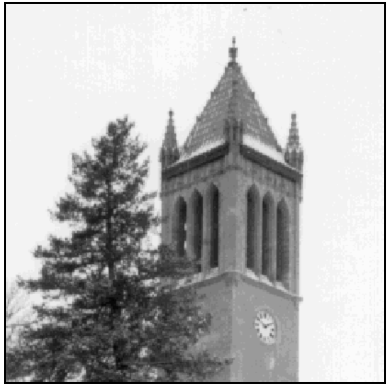
128 levels (7-bit)

64 levels (6-bit)

32 levels (5-bit)

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

Digital Image – Quantization Levels



16 levels (4-bit)

8 levels (3-bit)

In this image
it is easy to see
a ‘false’ contour

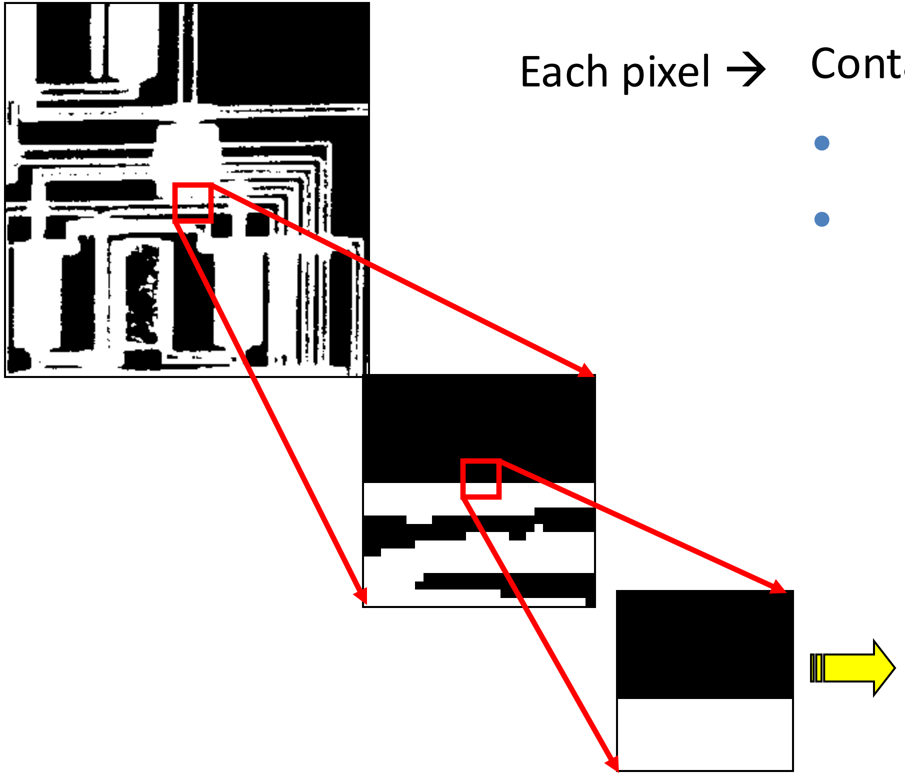
4 levels (2-bit)

2 levels (1-bit)

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

Digital Image – Binary Image

Binary- or **Black-n-White-** image



Each pixel → Contains `1 bit`:

- 0 → black
- 1 → white

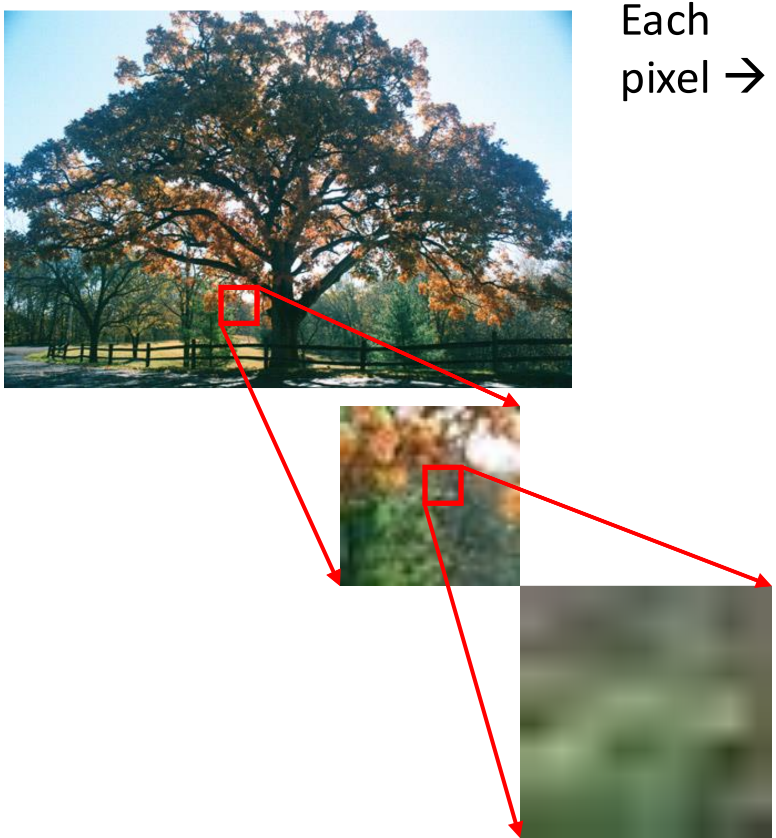
Binary data

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

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

Digital Image – Color

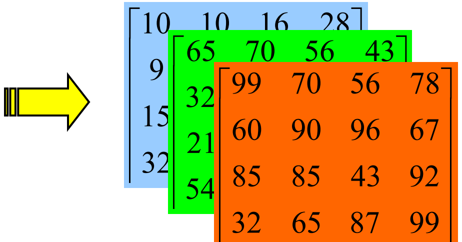
Color (RGB) image



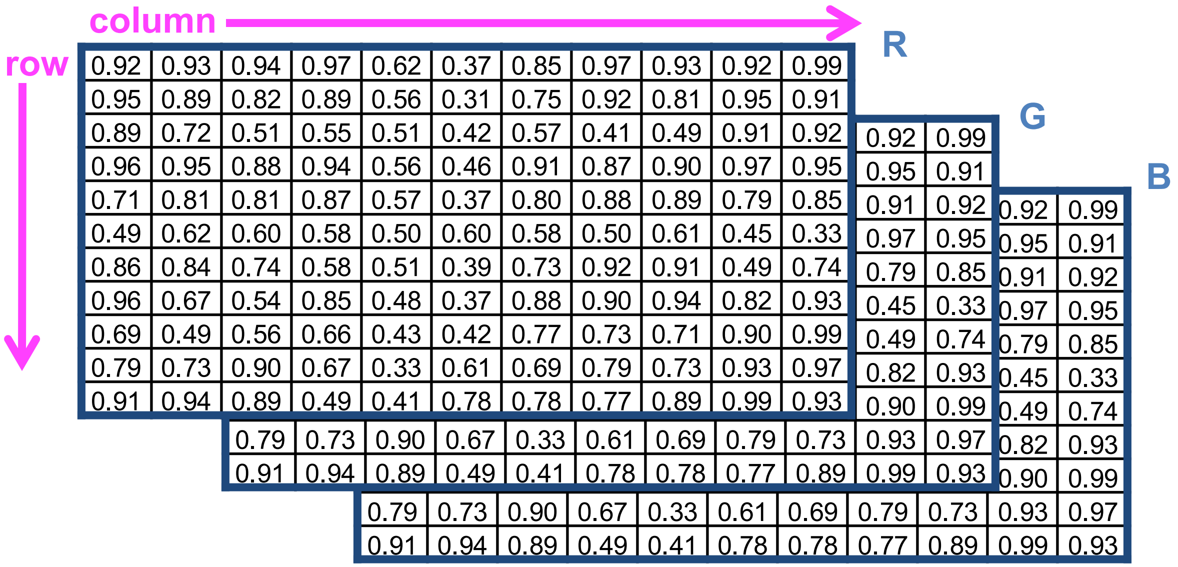
Each pixel → Contains a 3-element vector:

- [0, …, 255] → Red
- [0, …, 255] → Green
- [0, …, 255] → Blue

RGB components



Digital Image – Matlab

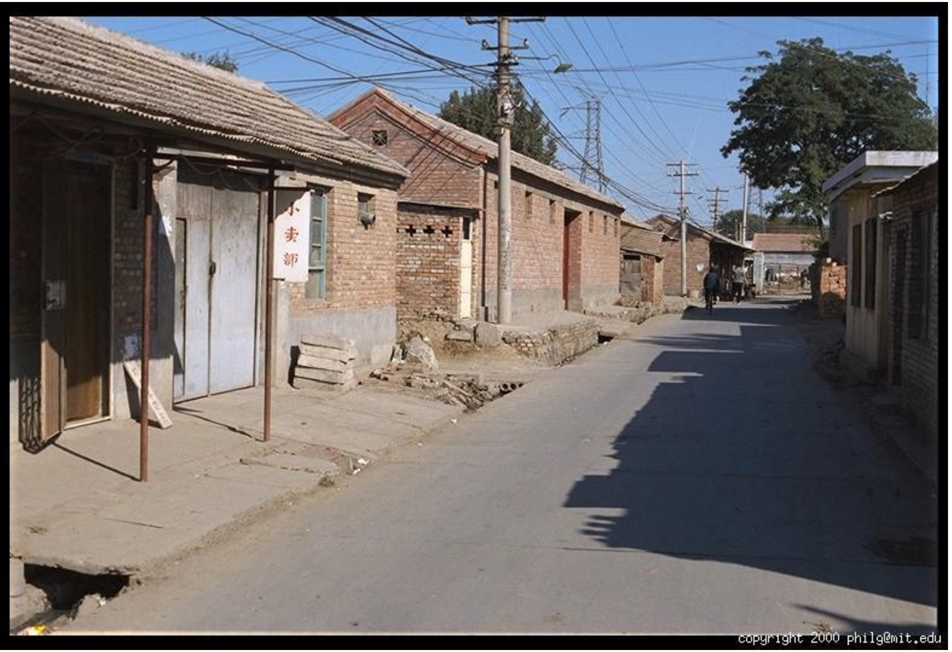


In **OpenCV** the channel order is **BGR**!

- 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 b^{th} channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - **For processing: Convert to double format (values 0 to 1) with im2double() → important!**

Digital Image

Shape information mostly contained
in **intensity** (grayscale) image



Color image

Grayscale image

Outline

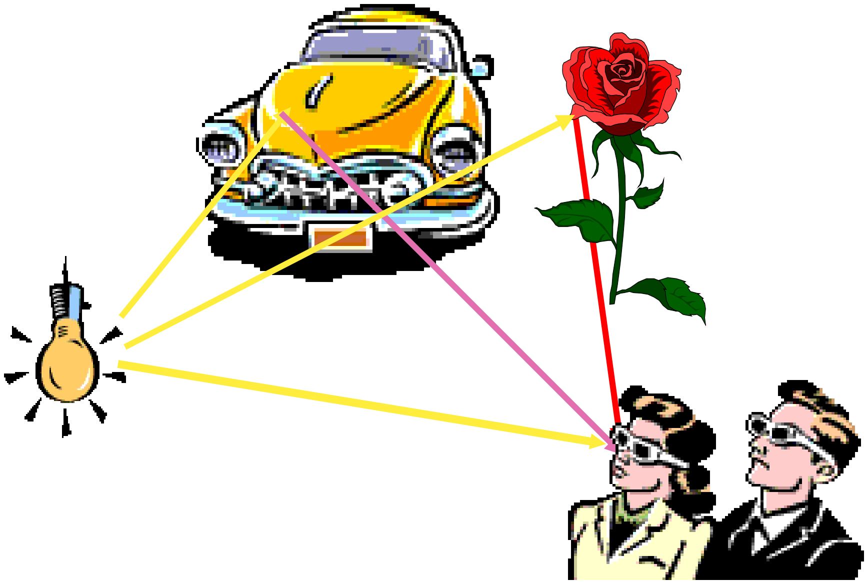
- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
    - Colour Systems

What makes for an image?

Light source

Object (s)

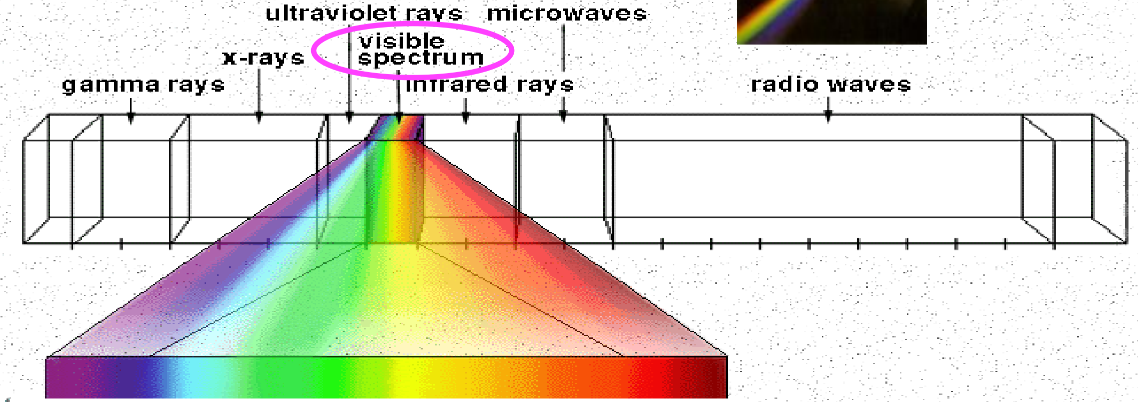
Observer / Sensor

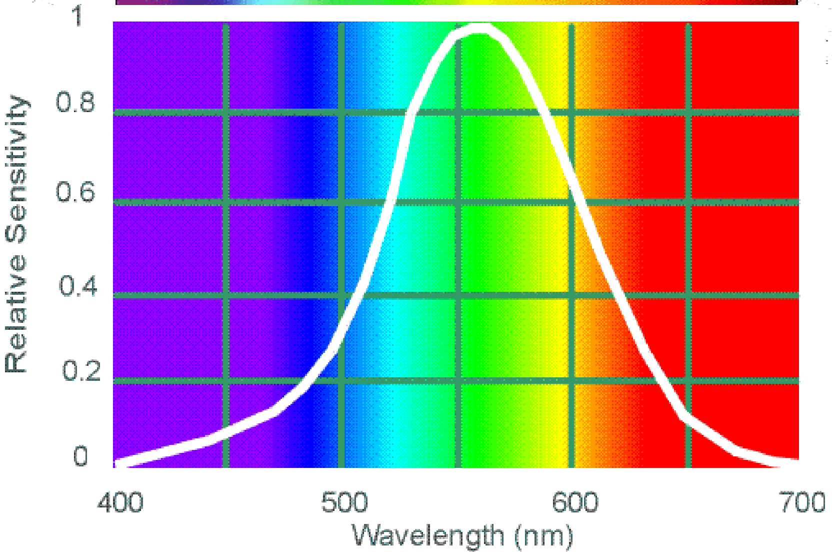


Humans – Sensitivity Function

Most sensitive at middle wavelengths

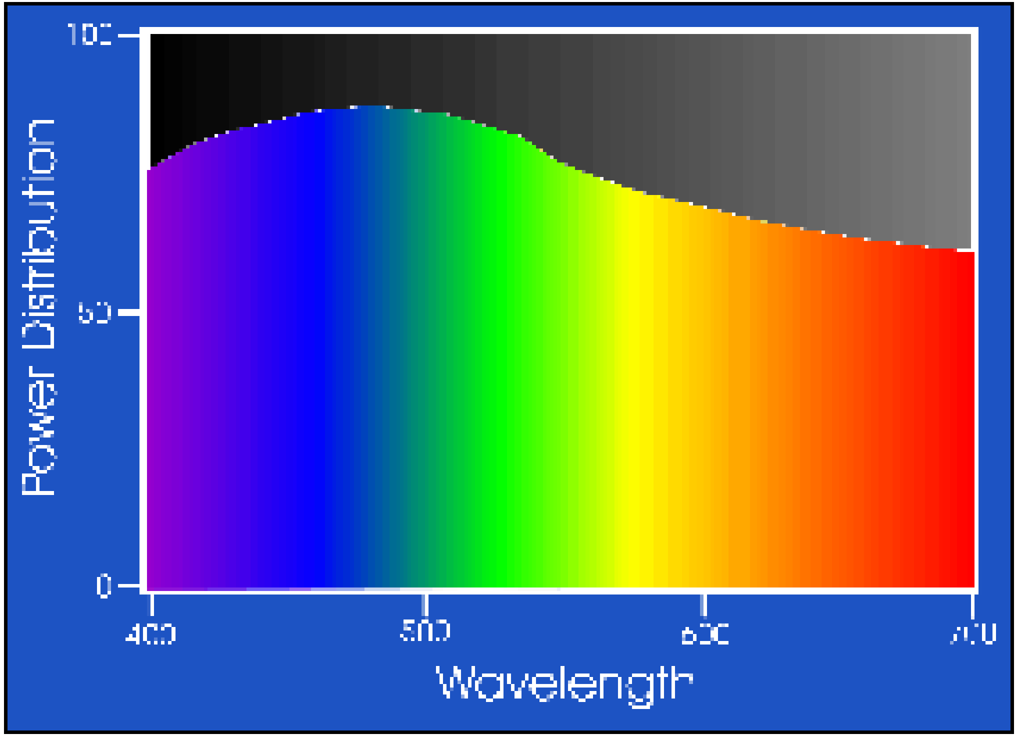
Sensitivity falls off for long & short wavelengths



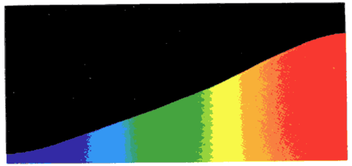


http://www.yorku.ca/eye/photopik.htm

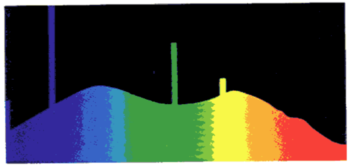
Light Sources – Spectral Power Distribution



Light Sources – Types



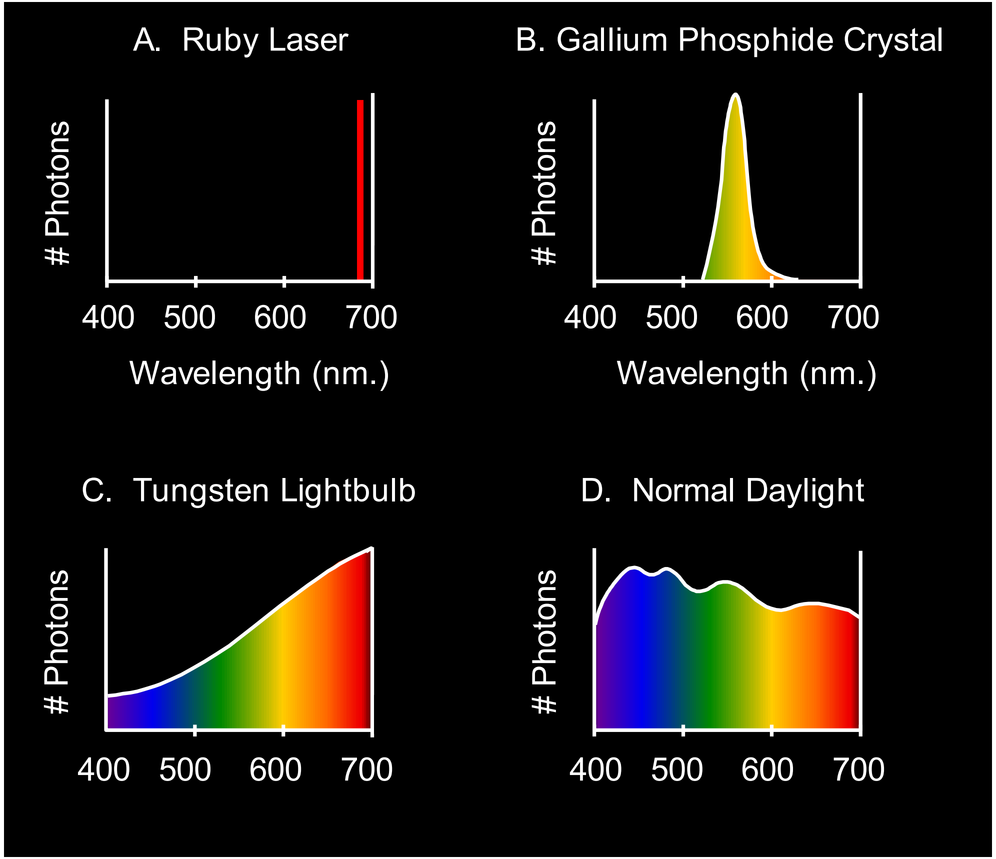
Incandescent
lamp



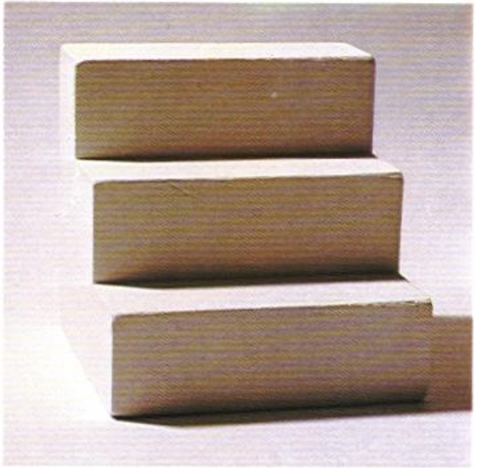
Fluorescent
lamp

Light Sources – Types

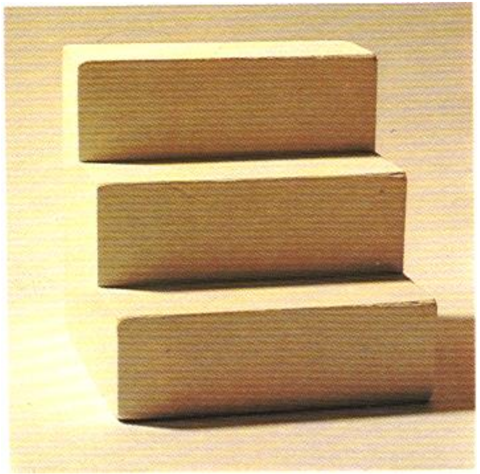
Spectra examples of various light sources



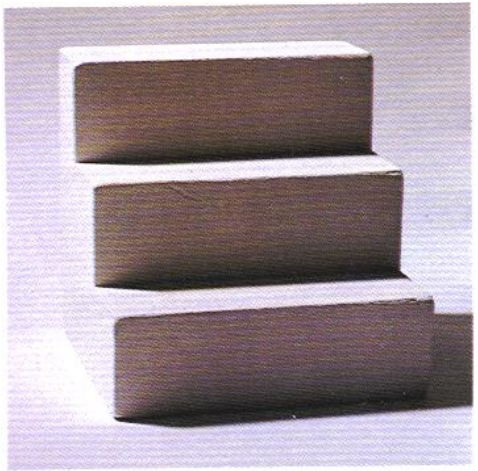
Light Sources – Influence



Average daylight

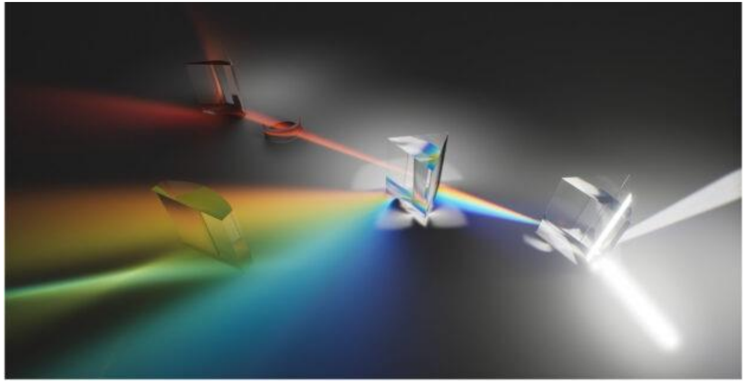


Incandescent lamp



Fluorescent lamp

Light Sources – Computer Graphics



Nvidia, Generating Ray-Traced Caustic Effects in Unreal Engine 4, Dec. 2020

Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
    - Colour Systems

Object Colors

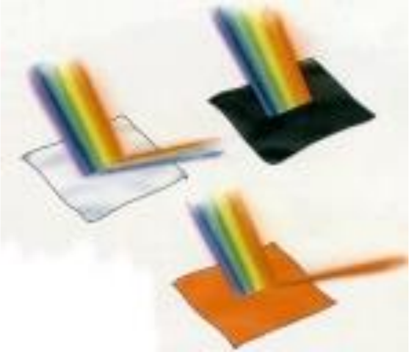
Materials:

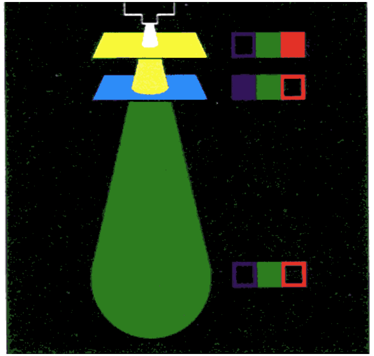
Transparent

Opaque

Spectral Reflectance $\rho(\lambda)$

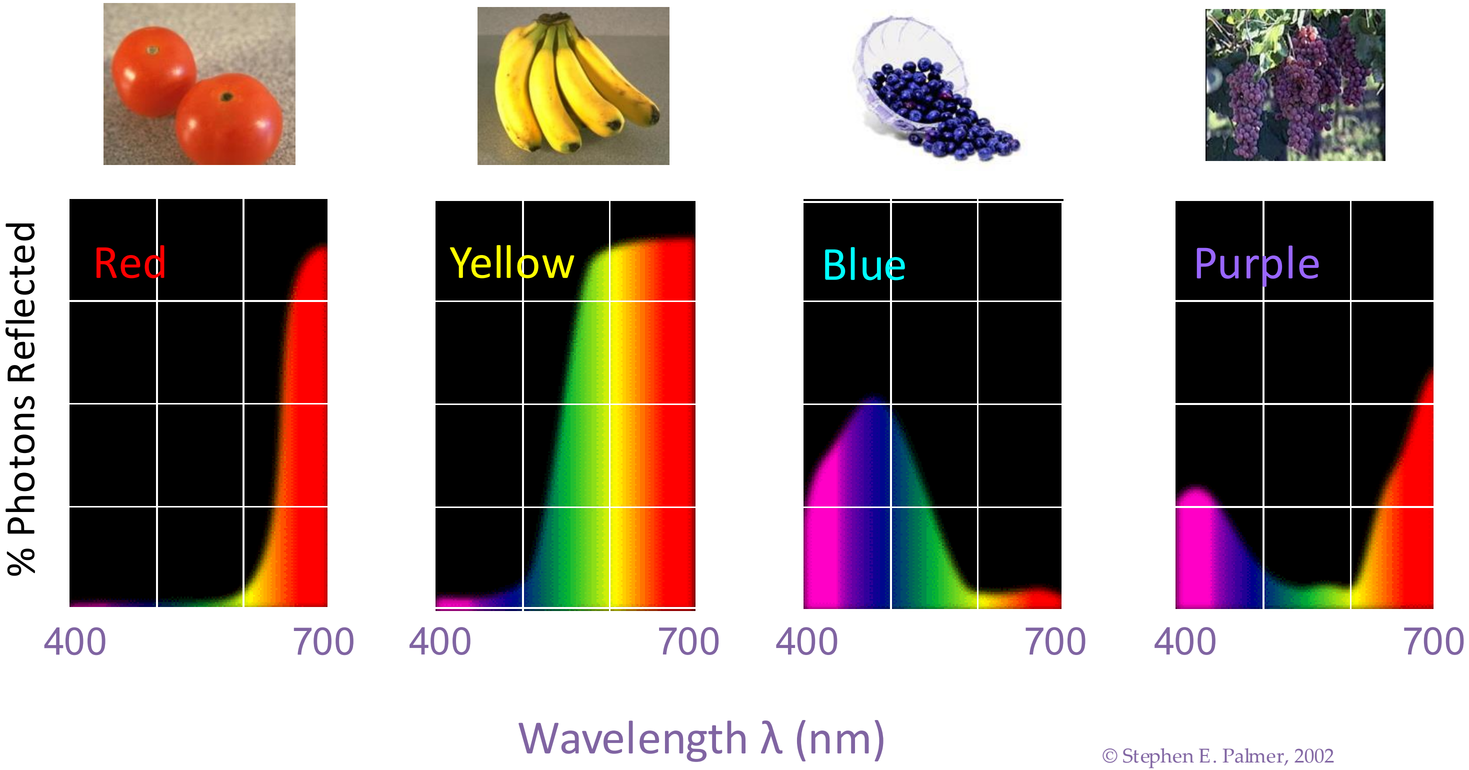
Wavelength of light





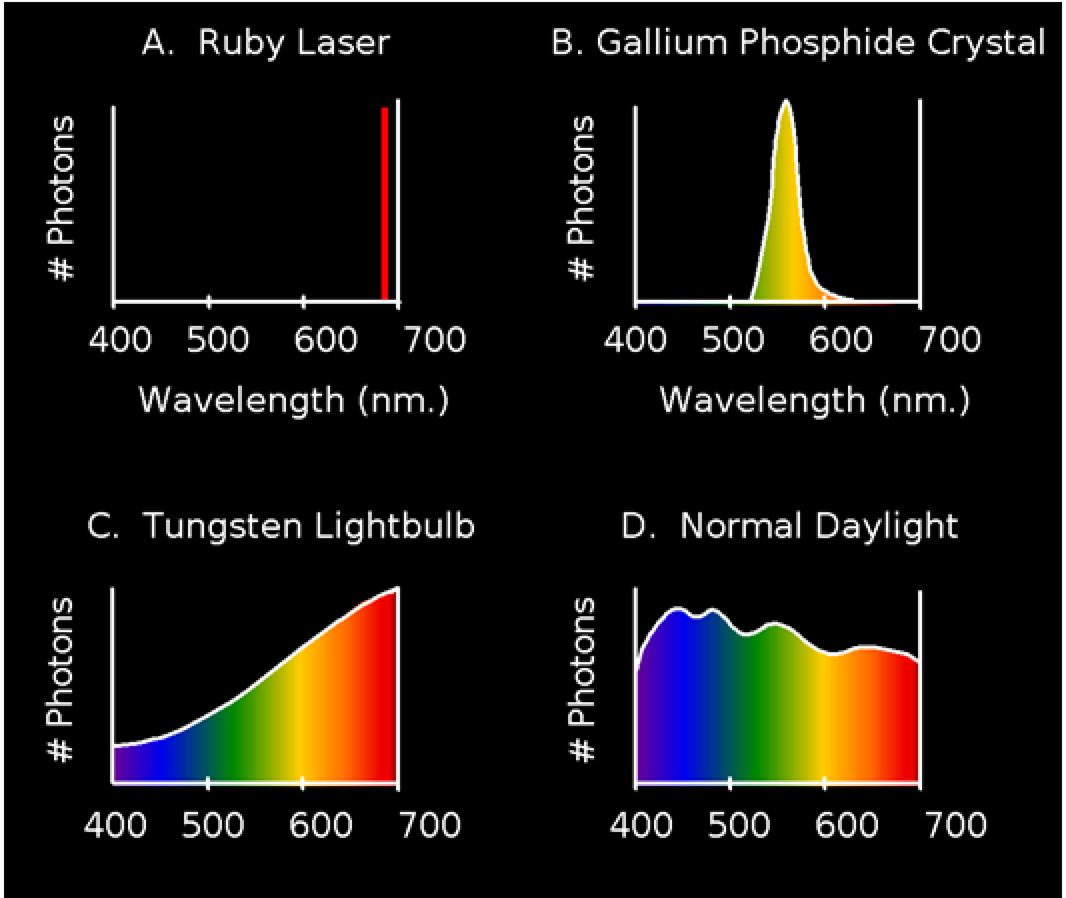
Object Colors

Examples of reflectance spectra of surfaces



Object Colors

Light sources are diverse!

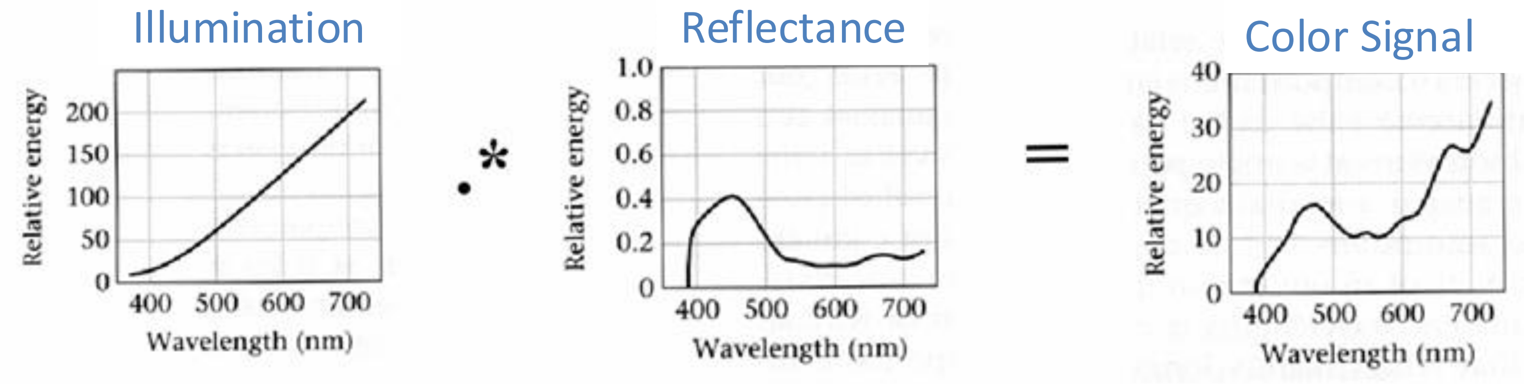


Object Colors

What color is the object?

Depends on **both**:

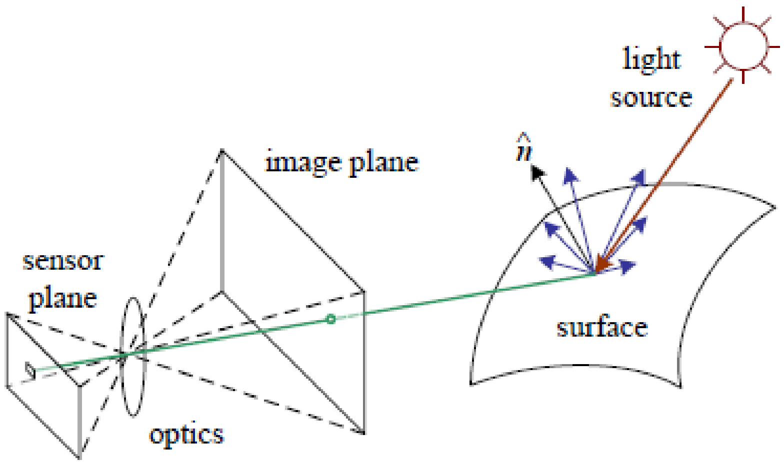
- Incident light – Illumination
- Object's Reflectance



Object Colors

Four main factors influence image intensity values

- Illumination of scene
- Geometry of scene
- Reflectance of visible surfaces
- Camera view & optics



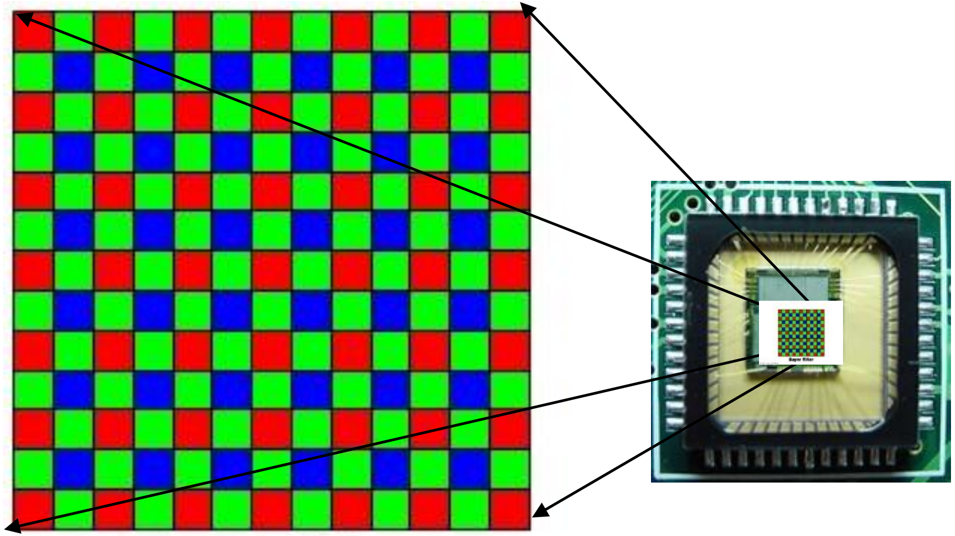
Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
    - Colour Systems

Color images – How to form?

Bayer filter

- Green fills in half of checkerboard
- Red and Blue fill the rest.

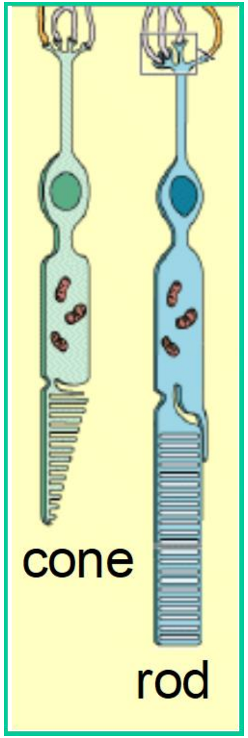


Bayer filter

Rods & Cones – Light-Sensitive Cells

Cones

- cone-shaped
- less sensitive
- operate in strong light
- color vision

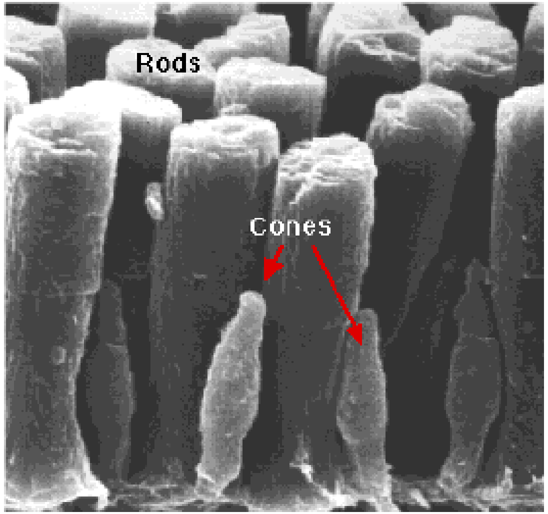


Rods

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

Slide source:
James Hays

Rods & Cones – Attributes



Rods:

- **120 million** rods in retina
- **Low-res**olution **peripheral** vision
- Sense **B/W brightness** in low illumination
- 1000X more light-sensitive than Cones
- Short wave-length sensitive

Cones:

- **6-7 million** cones in the retina
- **High-res**olution vision @ **Fovea**
- Sense Colors
- **3 types**: **64% red, 32% green, 2% blue**)
- Sensitive to **any combination** of 3 colors

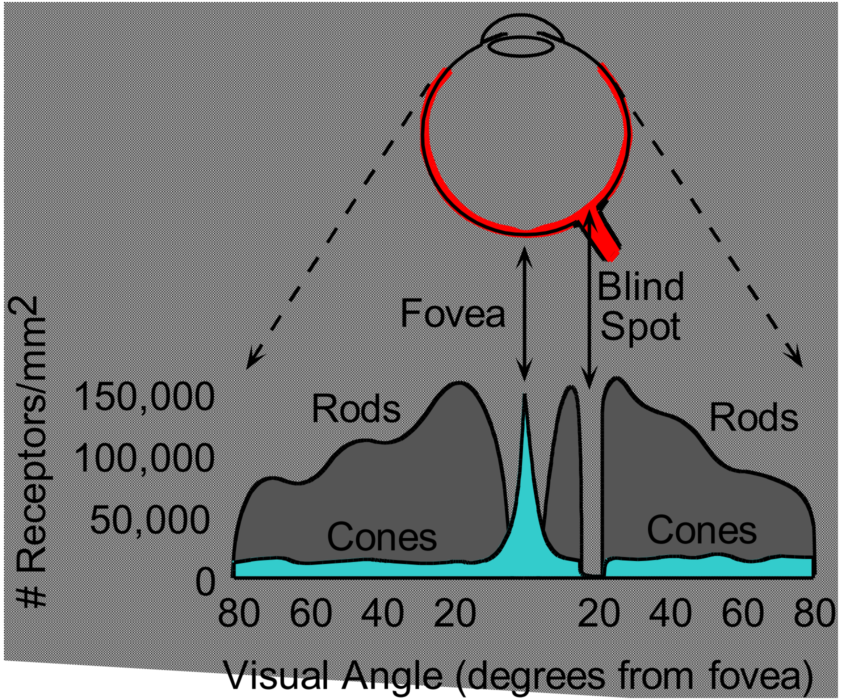
Rods & Cones – Distribution

Night Sky:

Why do we 'see' more stars appearing 'off-center' in our field of view?

Averted vision

Peripheral vision-areas on retina have more rod cells that are sensitive in low-light conditions

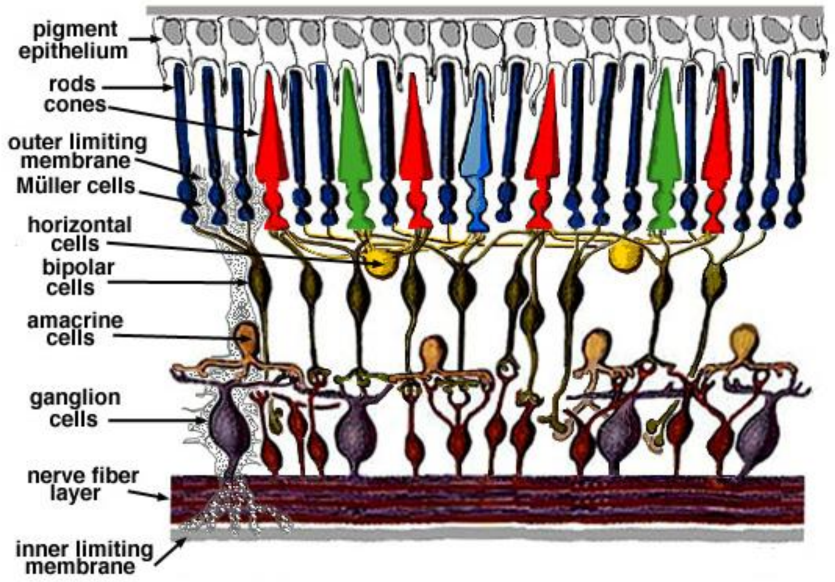


http://en.wikipedia.org/wiki/Averted_vision

Slide source: James Hays

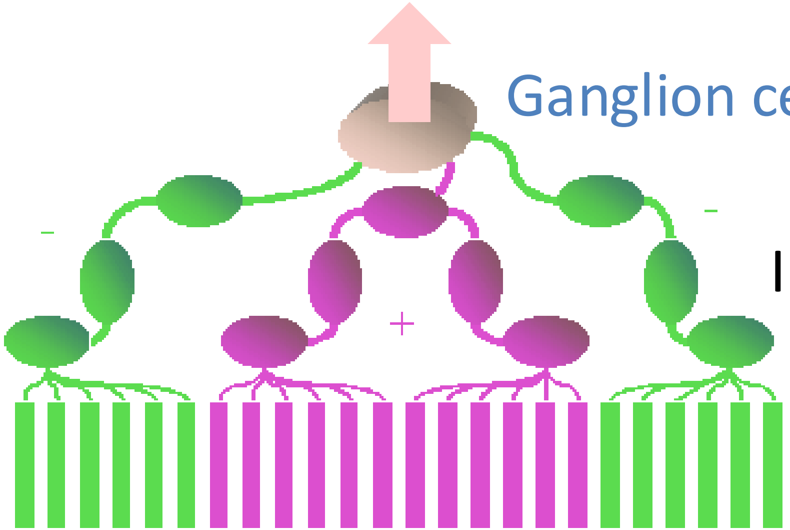
Retina

- 0.5 mm thick
- Photosensors (rods & cones) lie outermost in the retina
- Interneurons
- Ganglion cells (retina's output neurons) lie innermost in the retina closest to the lens & front of the eye



Ganglion Cell

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

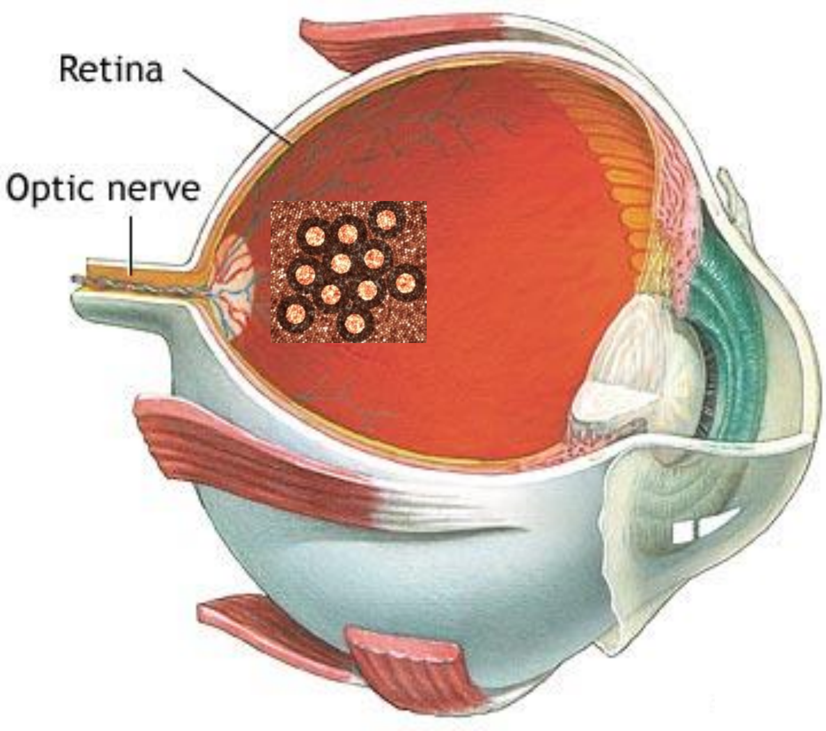


Ganglion cell

Intermediate neural cells

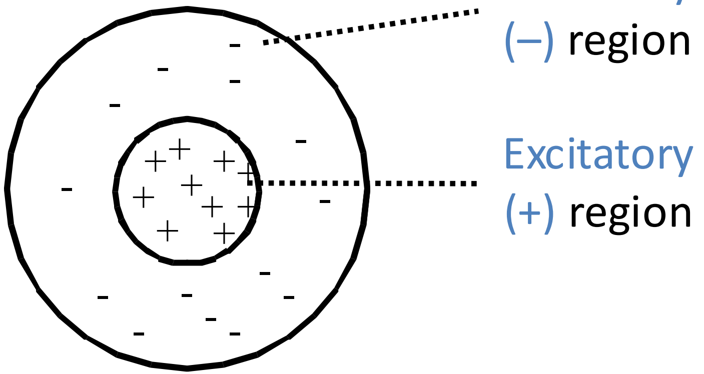
Photoreceptors:
Pink is excitatory
Green is inhibitory

Ganglion Cell – Receptive Fields



The size of receptors & receptive field shown here much larger than actual size!

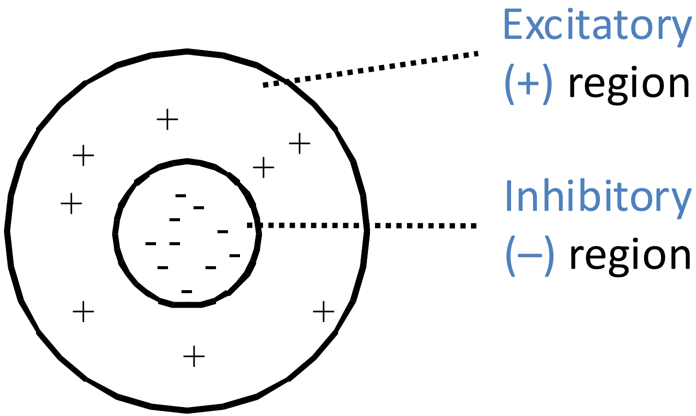
Ganglion Cell – Receptive Fields – On-Center



Responds maximally to light increments @ center

and light decrements in the surround

Ganglion Cell – Receptive Fields – Off-Center



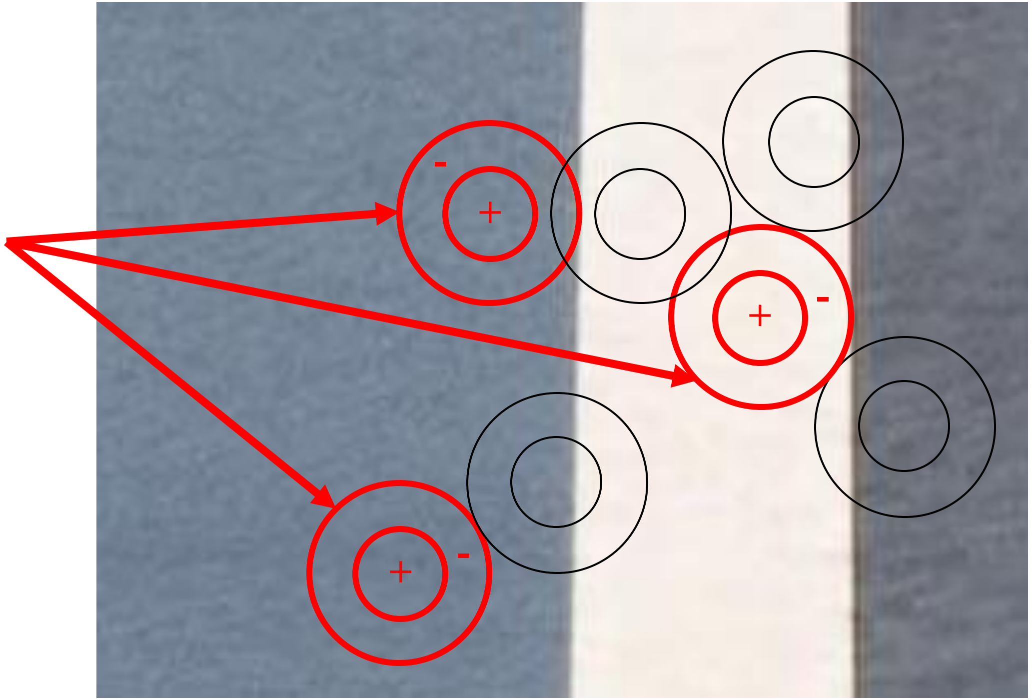
Responds maximally to light decrements in the center

and light increments in the surround

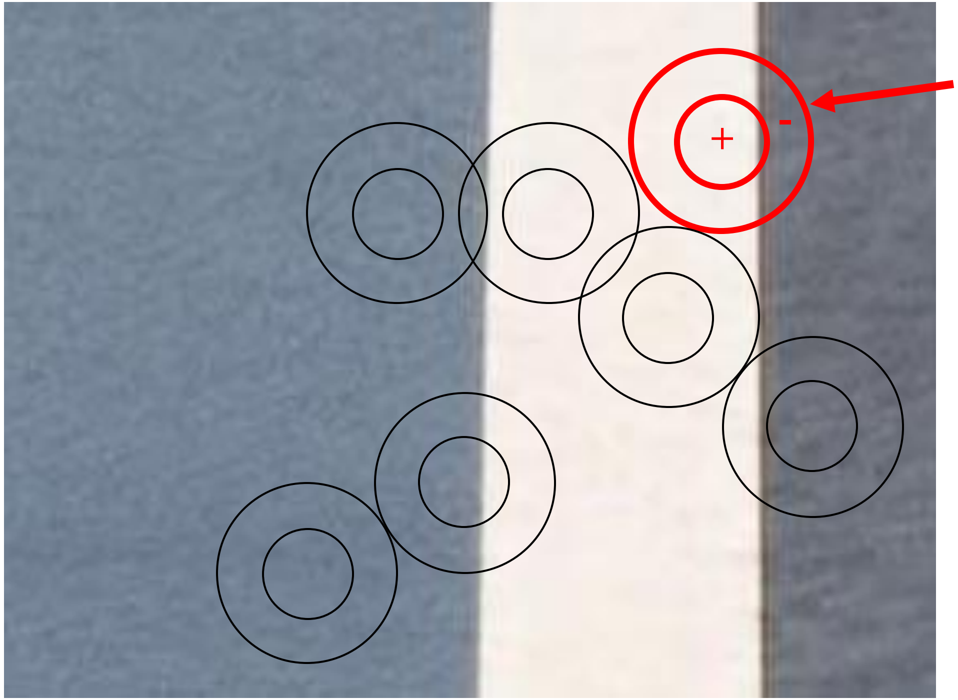
Ganglion Cell – Responses – On-Center

Receptive field uniformly illuminated

Response is unchanged



Ganglion Cell – Responses – On-Center

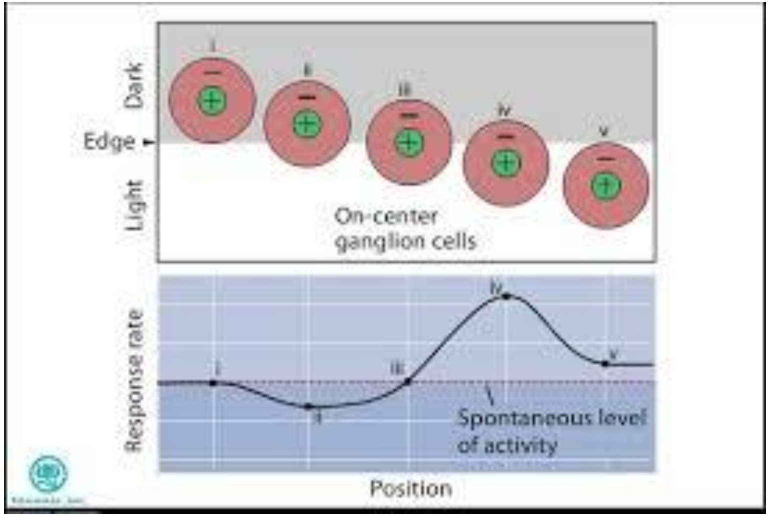


Entire excitatory reg. illuminated

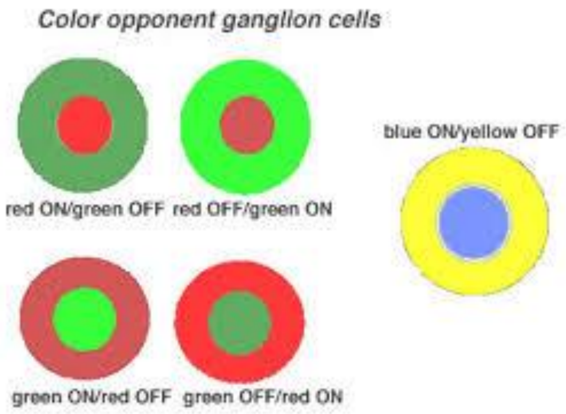
Part of inhibitory reg. not illuminated

Response is increased

Ganglion Cell – Responses – On-Center

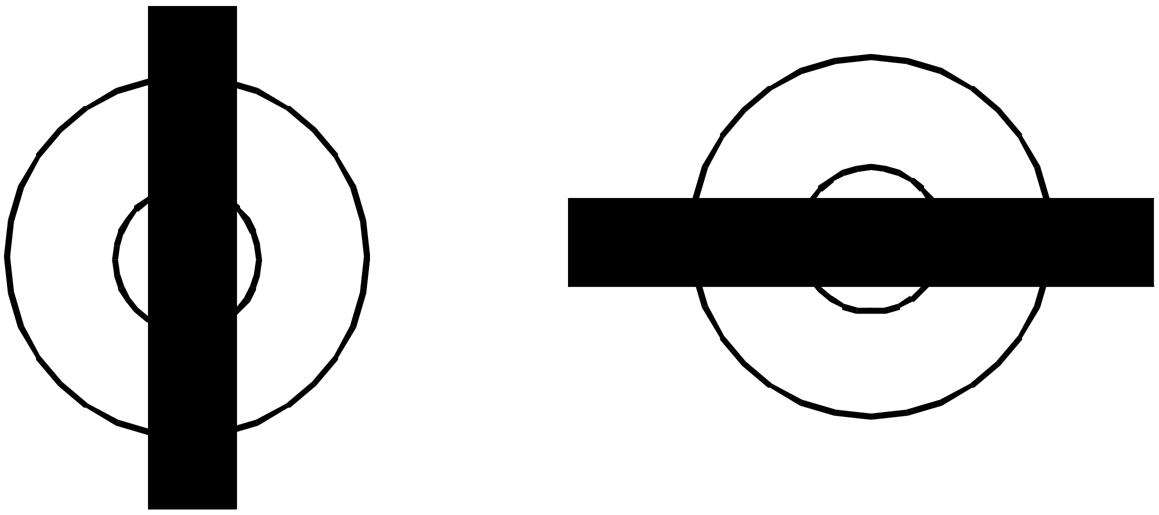


Ganglion Cell – Opponent Colors

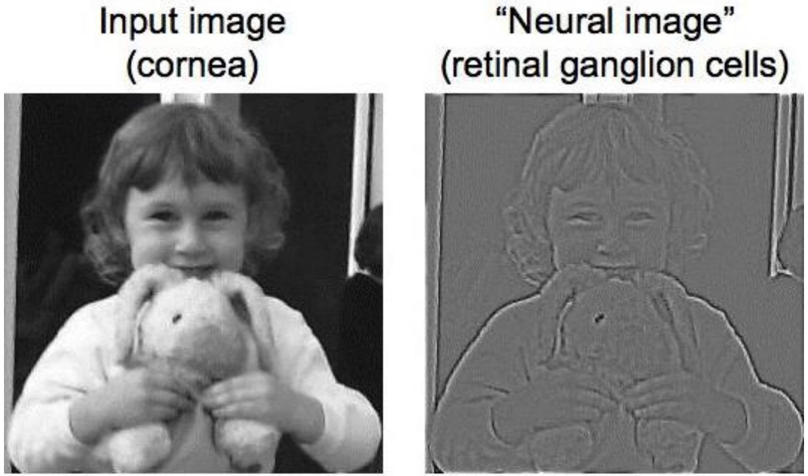


Ganglion Cell – Rotational Equivariance

Ganglion cells have no orientation preference



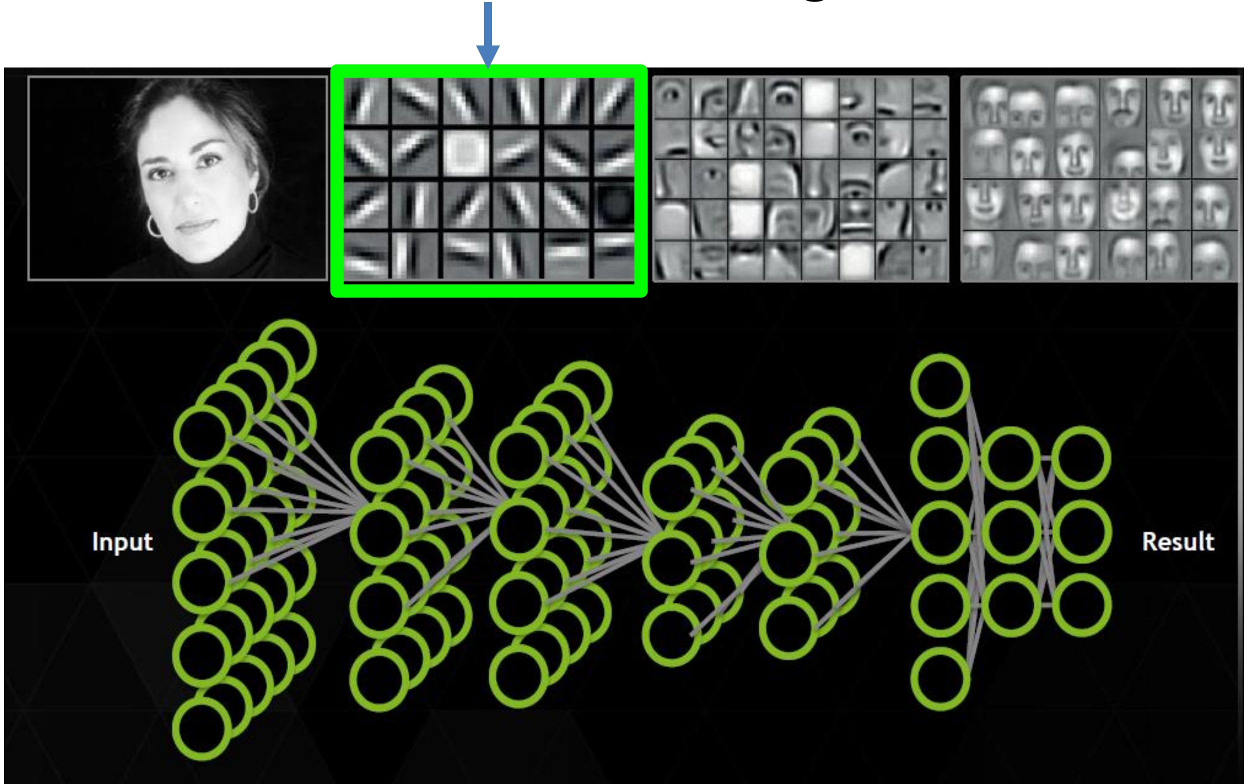
Edge Responses – Ganglion Cell



Ganglion cells respond to edges

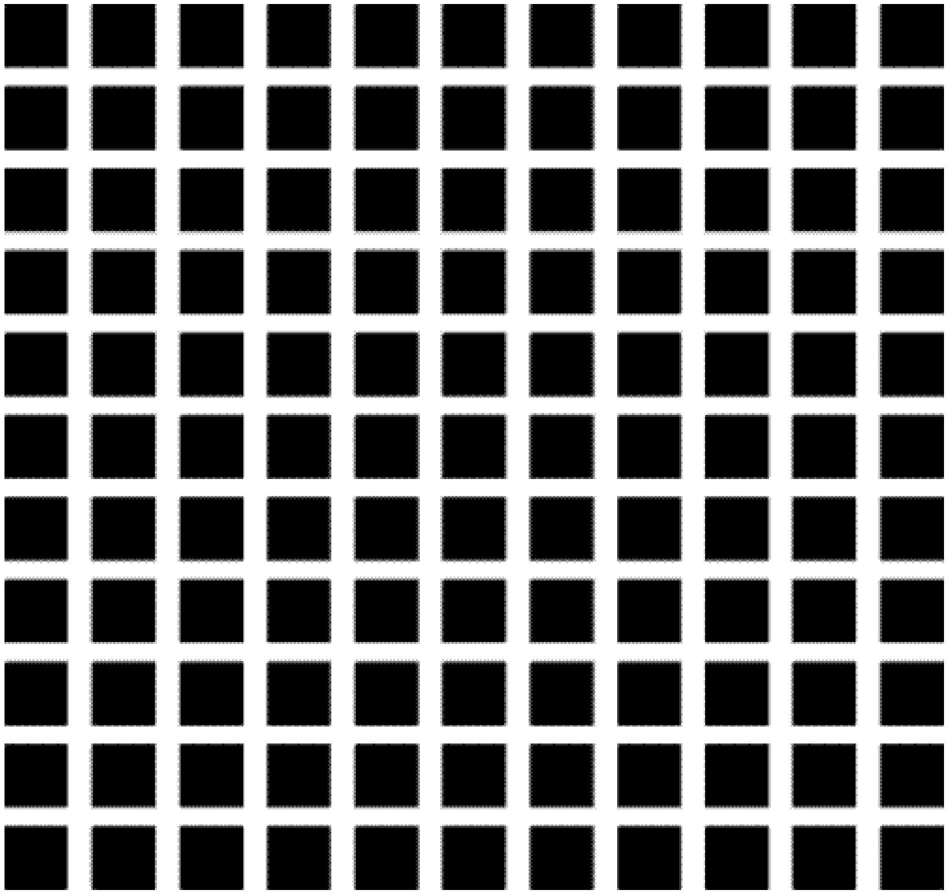
Edge Responses – Neural Nets

Neural Nets: Similar activations to Ganglion Cells at shallow layers



Ganglion Cells – Illusion – Hermann Grid

White grid on
black background

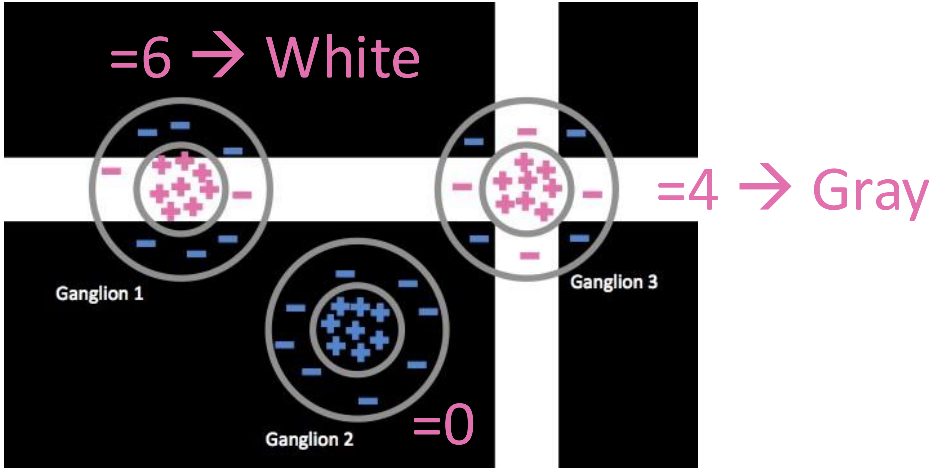


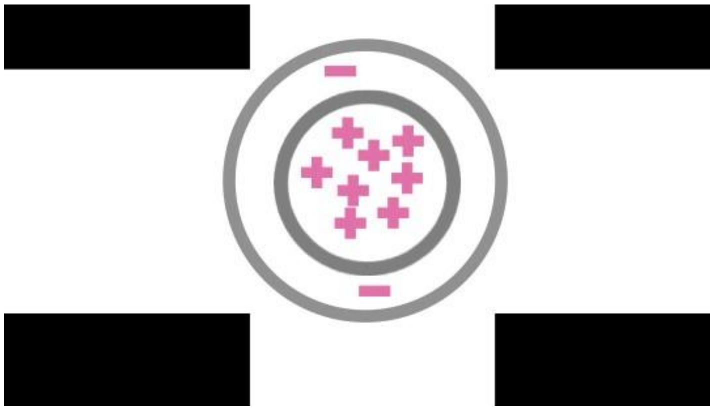
Illusion

Faint grey spots @ intersections

in peripheral areas!

Ganglion Cells – Illusion – Hermann Grid





Illusion – Grey spots @ peripheral intersections

Pink → Inputs **stimulated** by light
Blue → Inputs that **not stimulated**

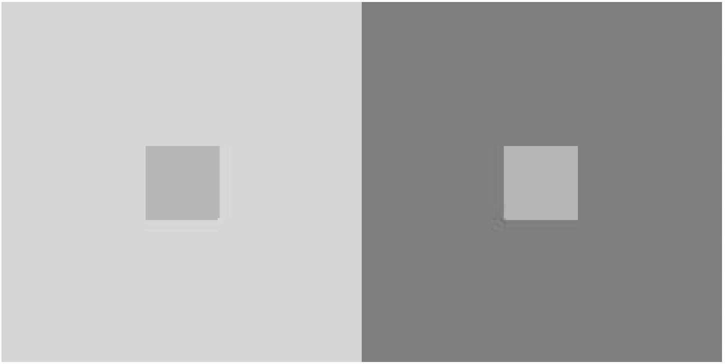
Compute **net** signal between activated + and –

Illusion disappears for intersection we directly look at!

Fovea → Receptive fields smaller
→ Cells @ intersection no longer surrounded by inhibitory components of grid

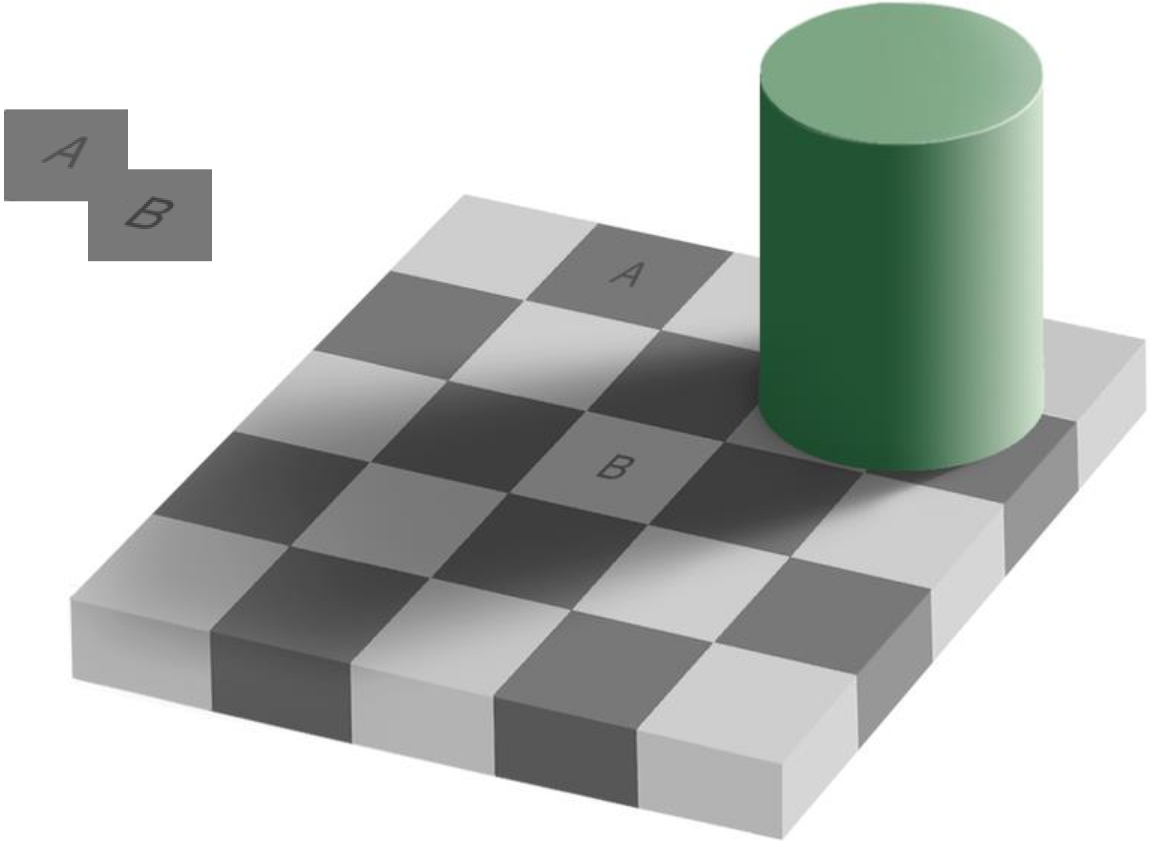
https://www.rosslab.neurobio.pitt.edu/making-sense-of-the-hermann-grid-illusion

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

Simultaneous Lightness Contrast

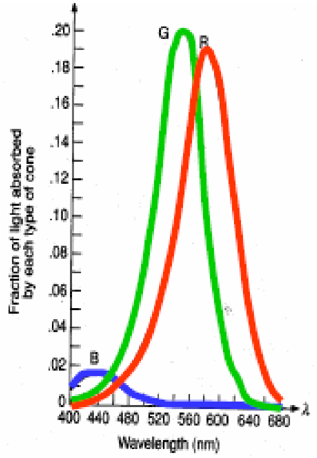


from Ted Adelson

Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
  - Colour Systems

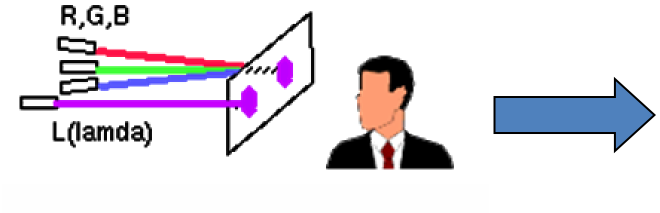
Color – Tristimulus Theory

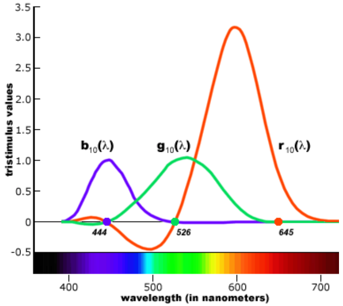


Spectral-response functions of each of 3 cone types

Color matching function based on RGB

- Any spectral color – Linear combination of these primary colors
- Perceptual experiment – Human observers try to match a color of a given wavelength λ by mixing 3 pure wavelengths



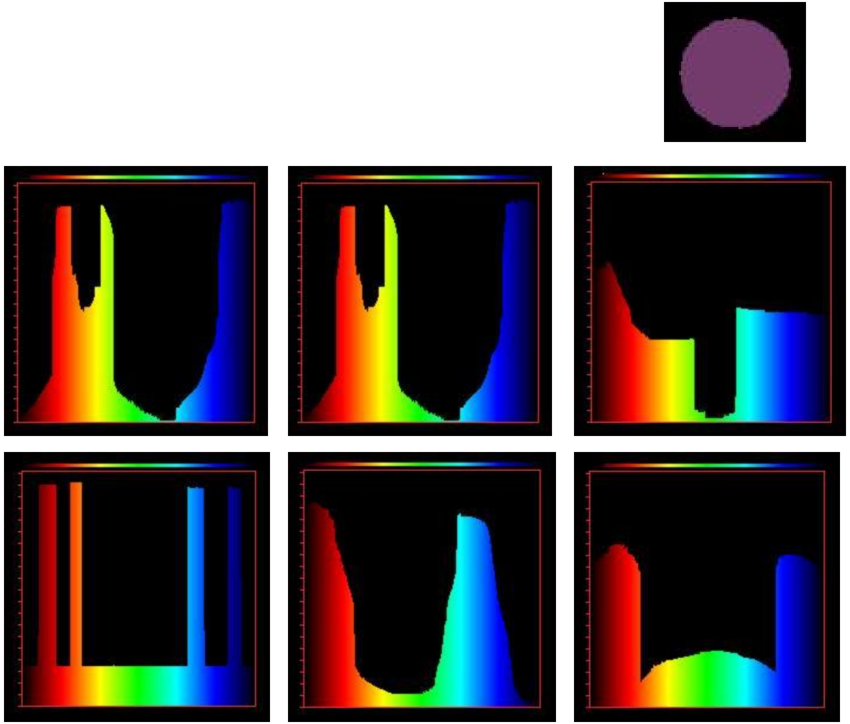


Problem: Sometimes red to be added to the target before a match can be achieved. Shown on the graph with a negative R value

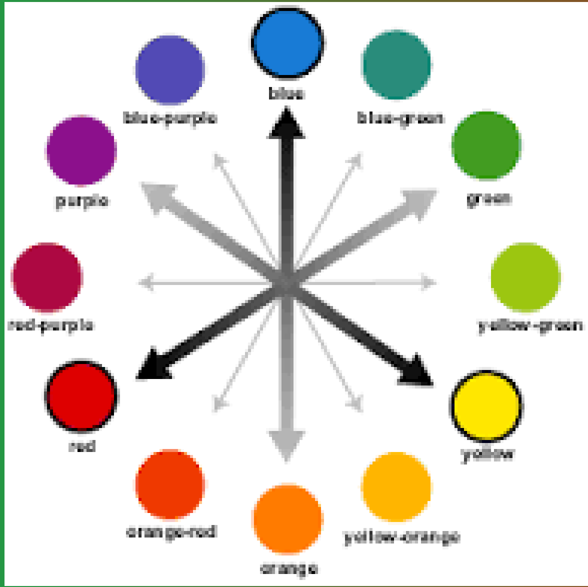
Spectral Energy Distribution

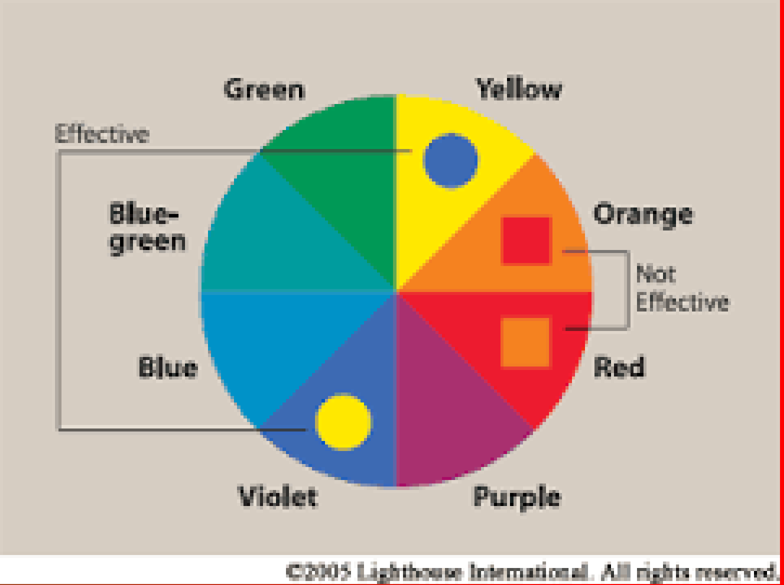
Subjective Perception:

These 6 spectra look like the 'same purple' to *normal* color-vision people

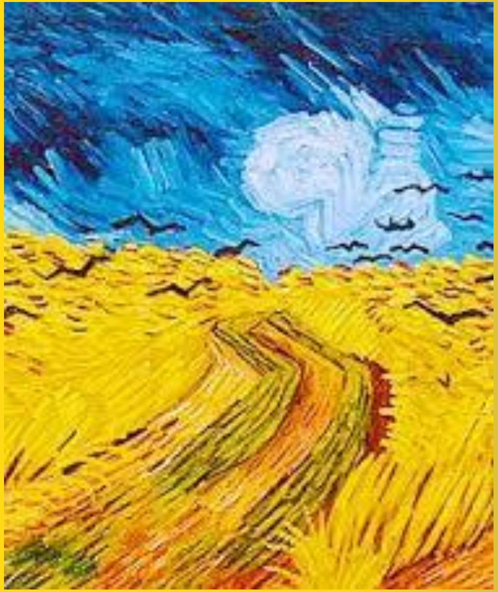


Opponent Colors





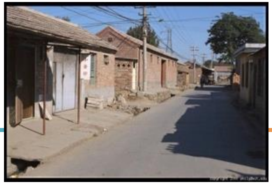
Opponent Colors

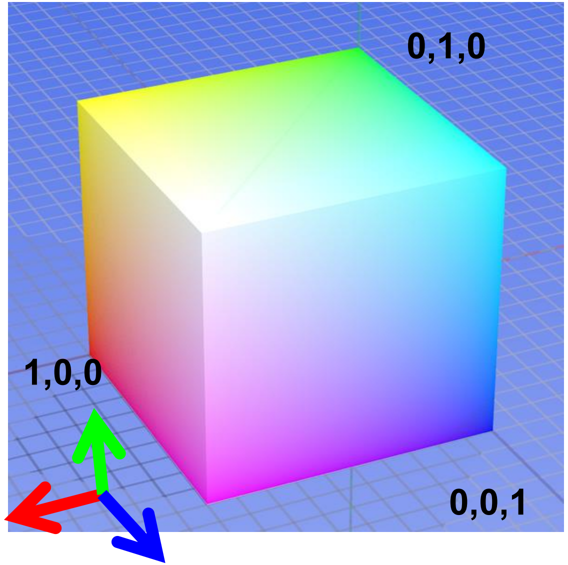


Outline

- Camera Model
  - Pinhole Camera
  - Geometric Image Formation
  - Photometric Image Formation
  - Image Representation
- Color
  - Physical & Biological Model
    - Light Source
    - Object
    - Observer
  - Tristimulus Theory
  - Colour Systems

RGB space





Default color space

Drawbacks:

- Strongly correlated channels
- Perceptually non-meaningful

R
(G=0,B=0)

G
(R=0,B=0)

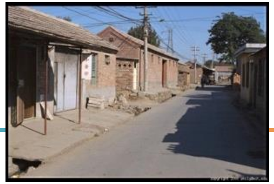
B
(R=0,G=0)

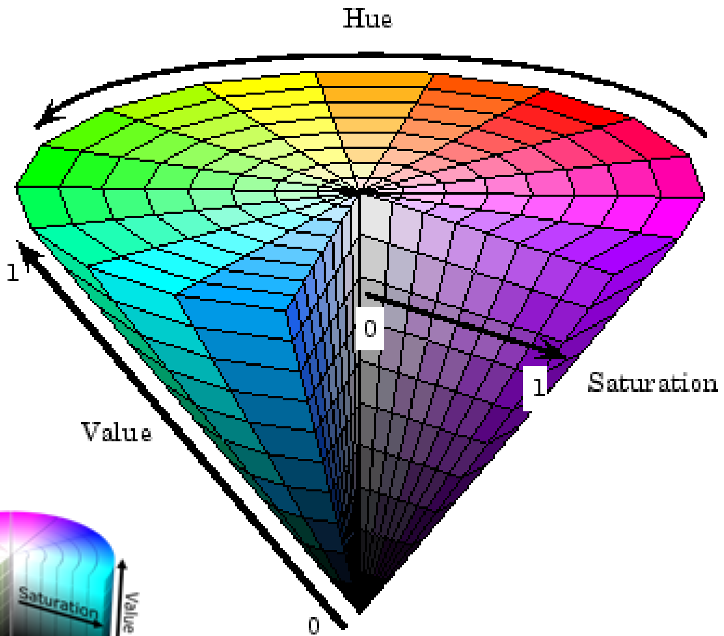
OpenCV order is

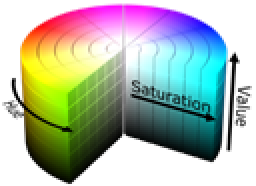
BGR

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

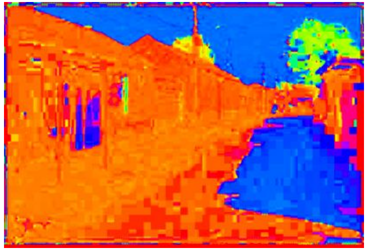
HSV Color Space







Intuitive color space



H
(S=1,V=1)

Hue

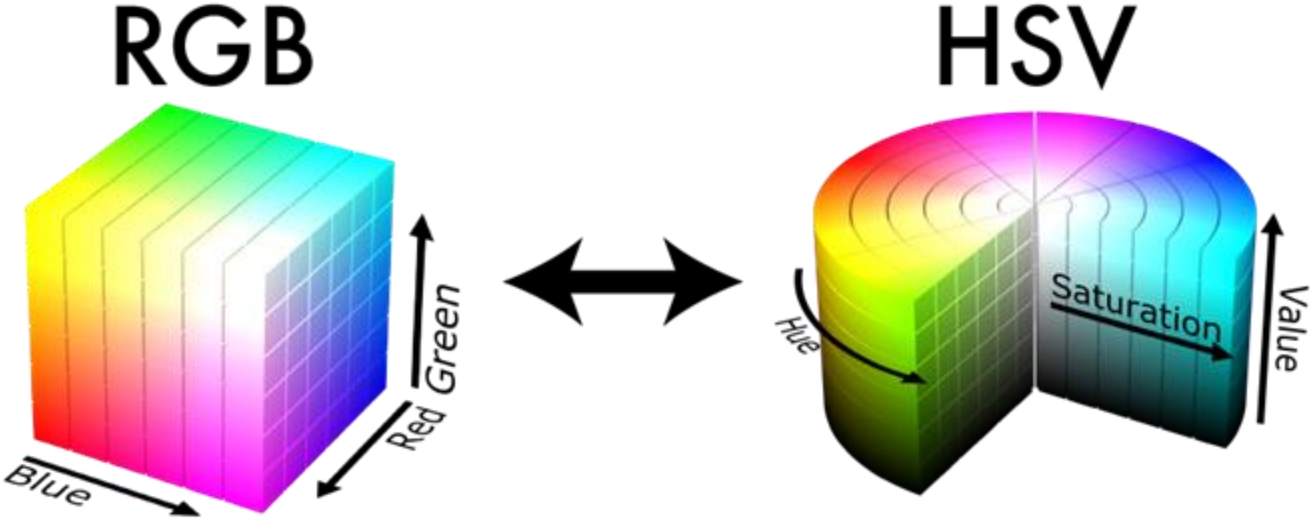
S
(H=1,V=1)

Saturation

V
(H=1,S=0)

Value

Color Space Conversion – RGB → HSV

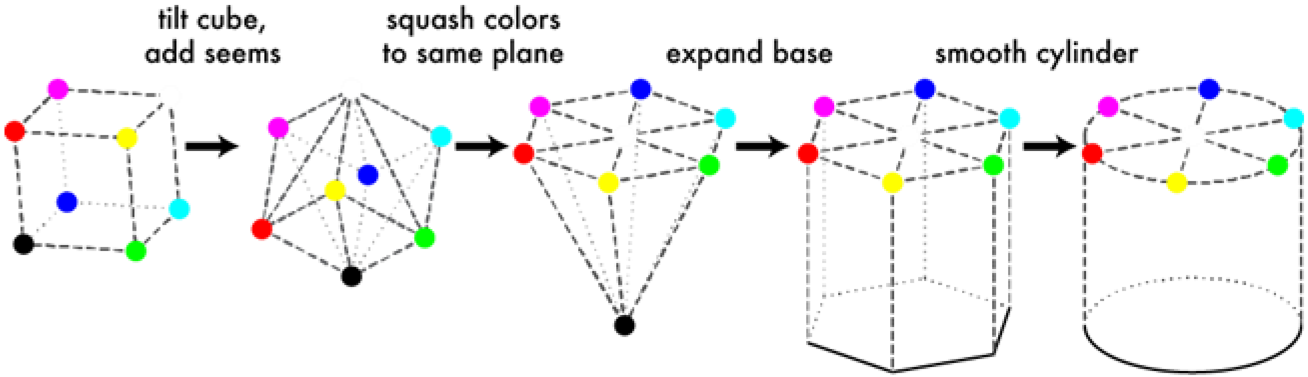


https://en.wikipedia.org/wiki/HSL_and_HSV

Color Space Conversion – RGB → HSV

Geometric Interpretation

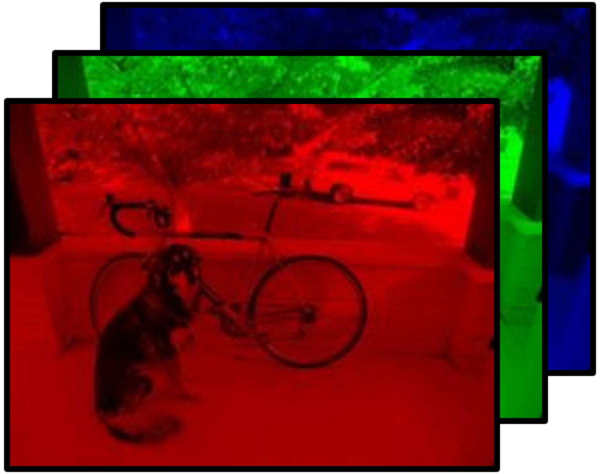
RGB

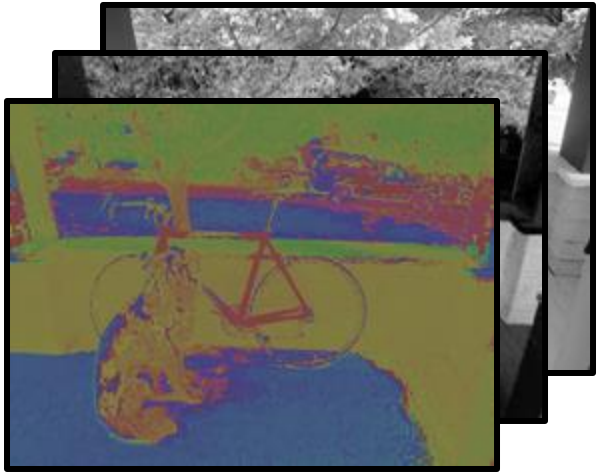


HSV

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

Color Space Conversion – RGB → HSV

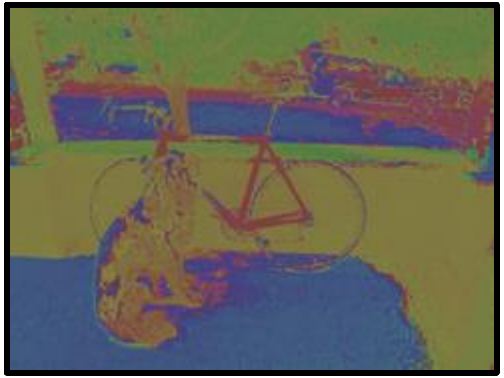




Still 3D tensors

but encode different info

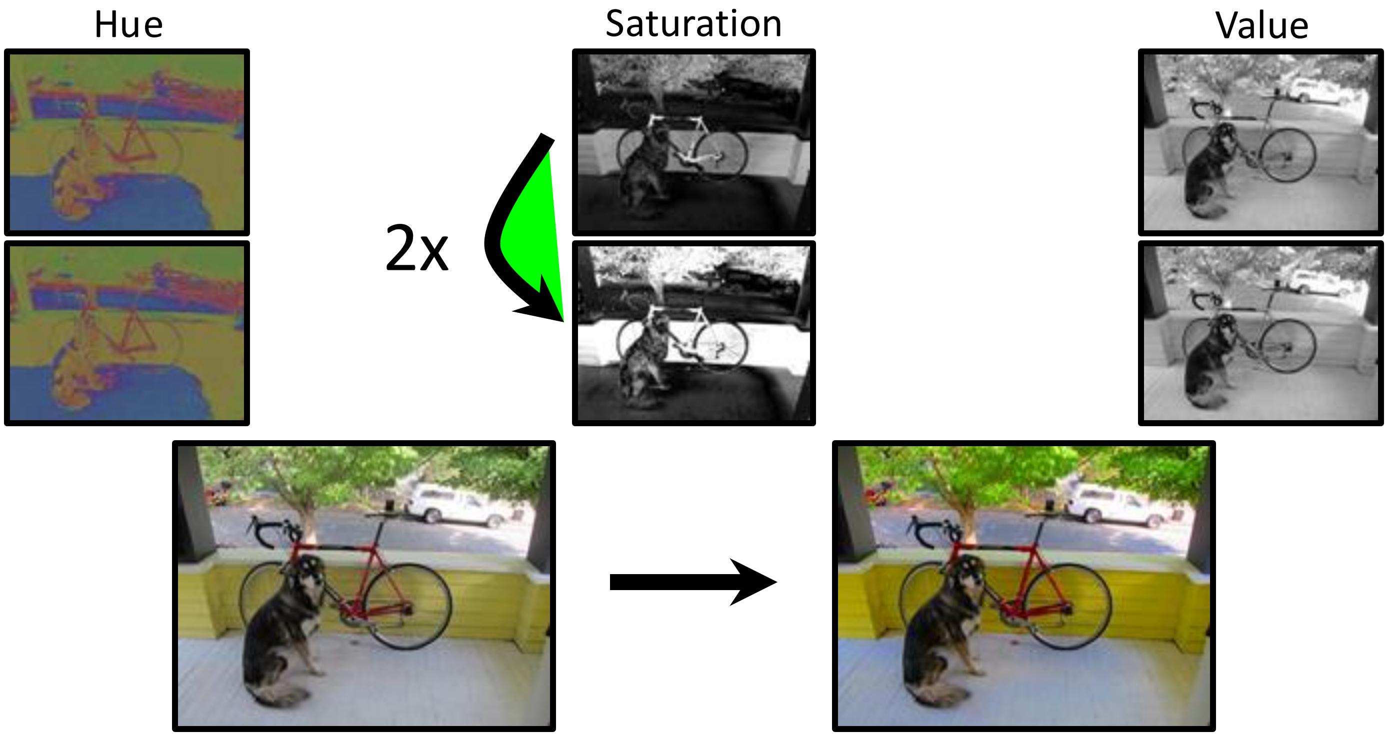
HSV Color Space


Hue


Saturation

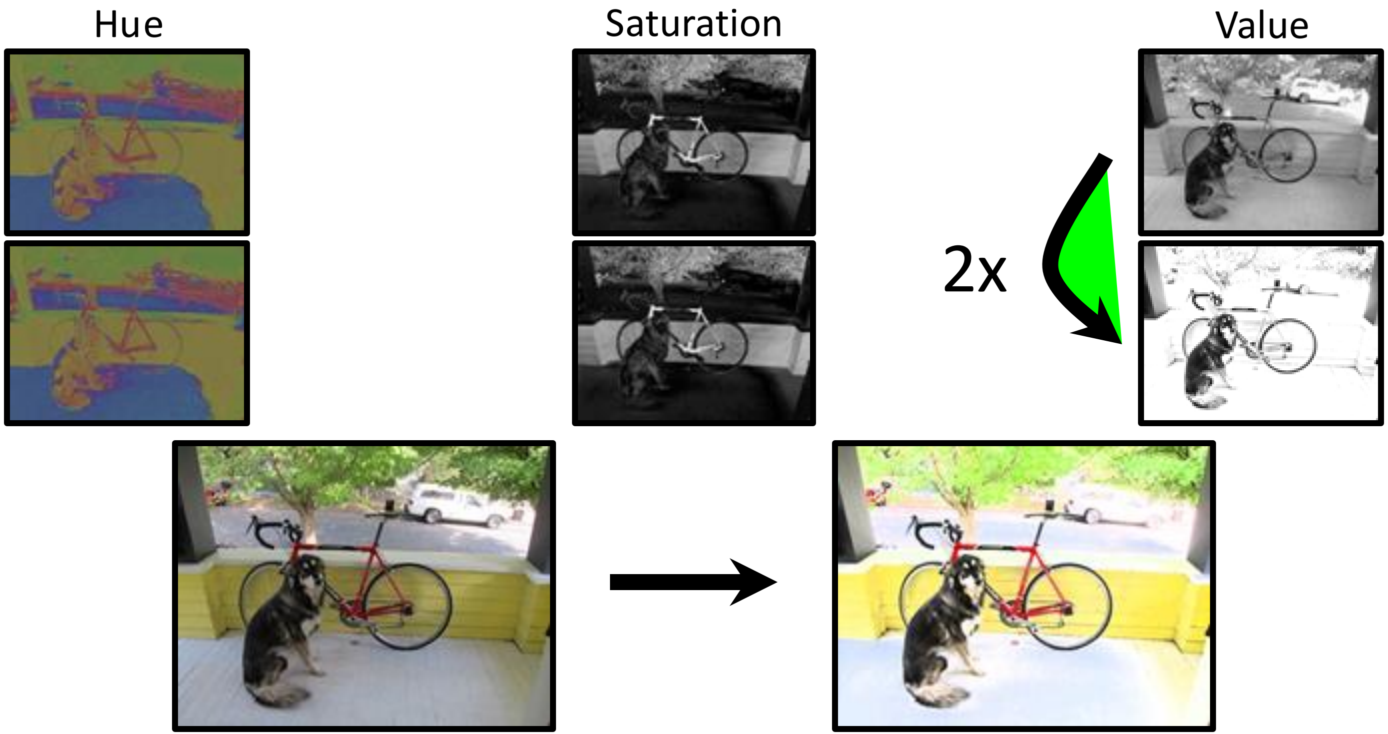

Value

HSV Color Space



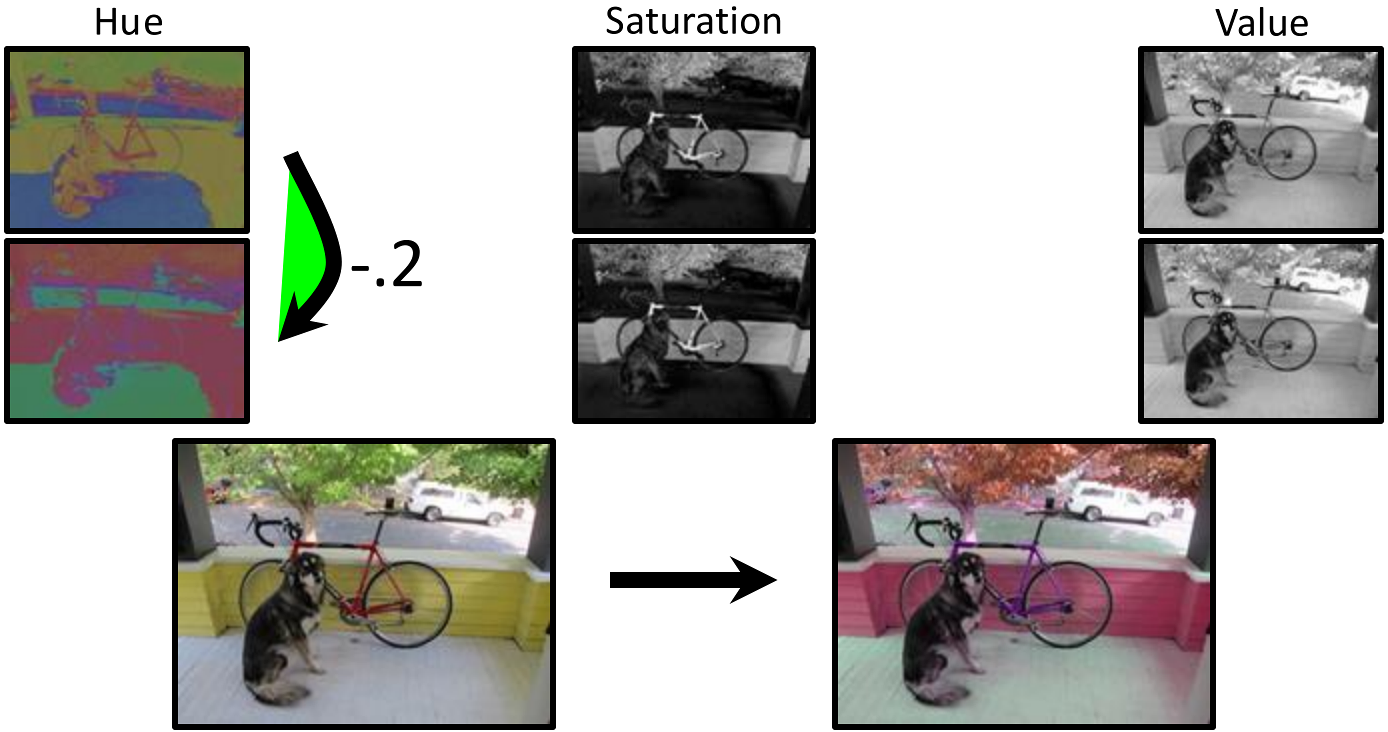
Higher Saturation → More intense colors

HSV Color Space



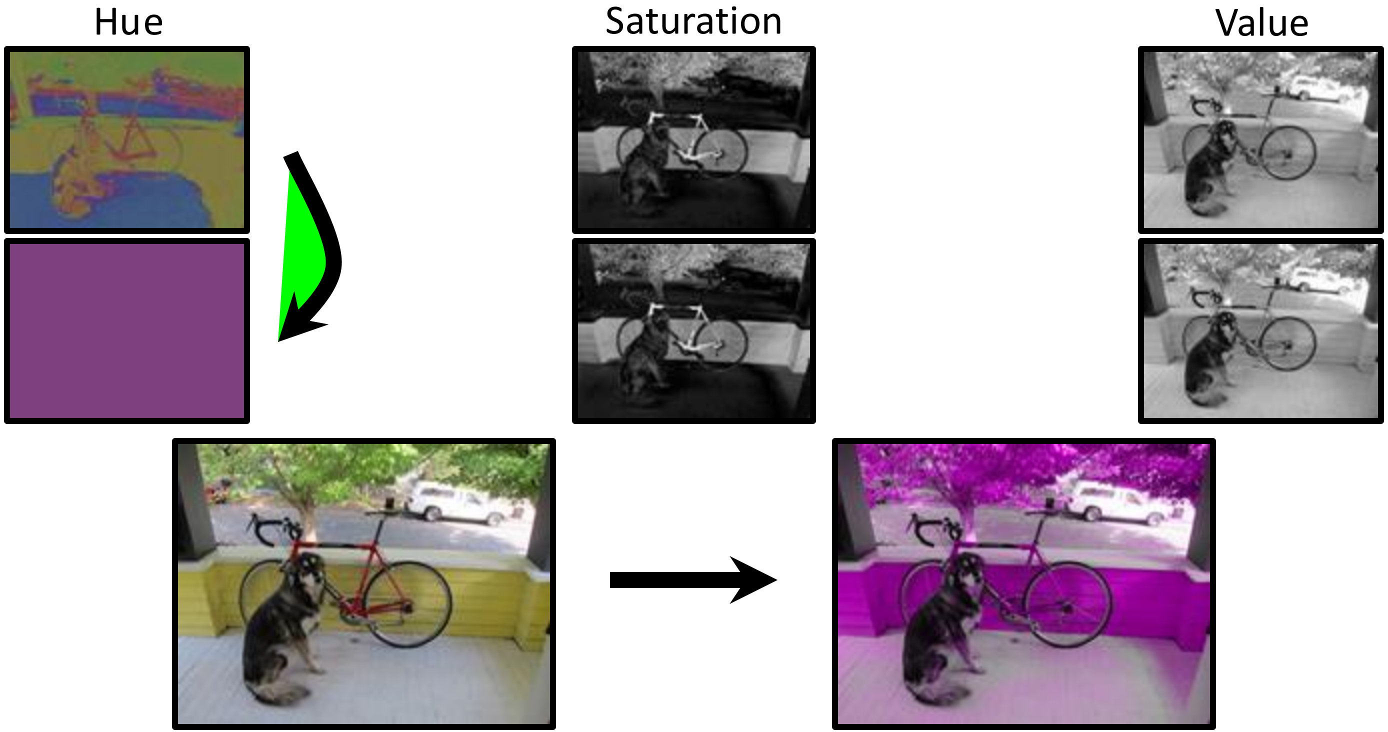
Higher Value → Brighter colors

HSV Color Space



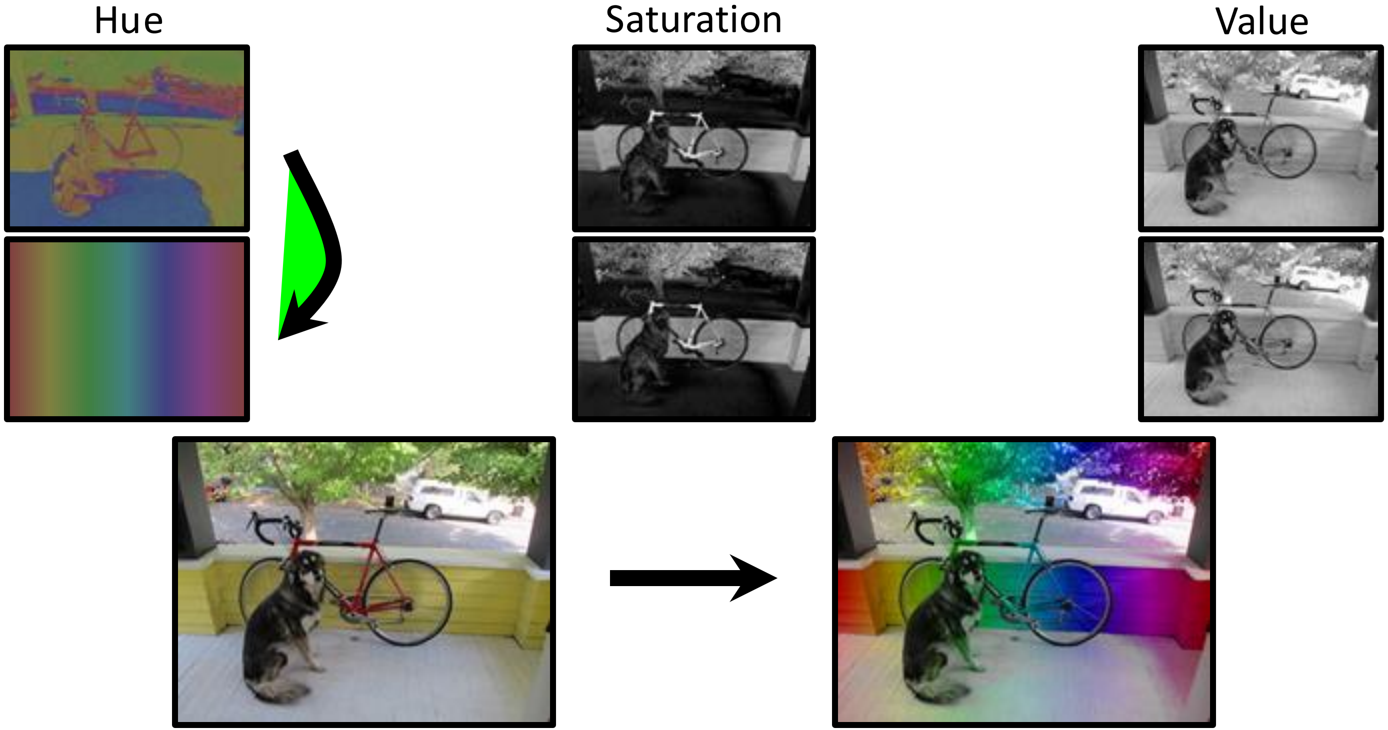
Change Hue → Change Colors

HSV Color Space



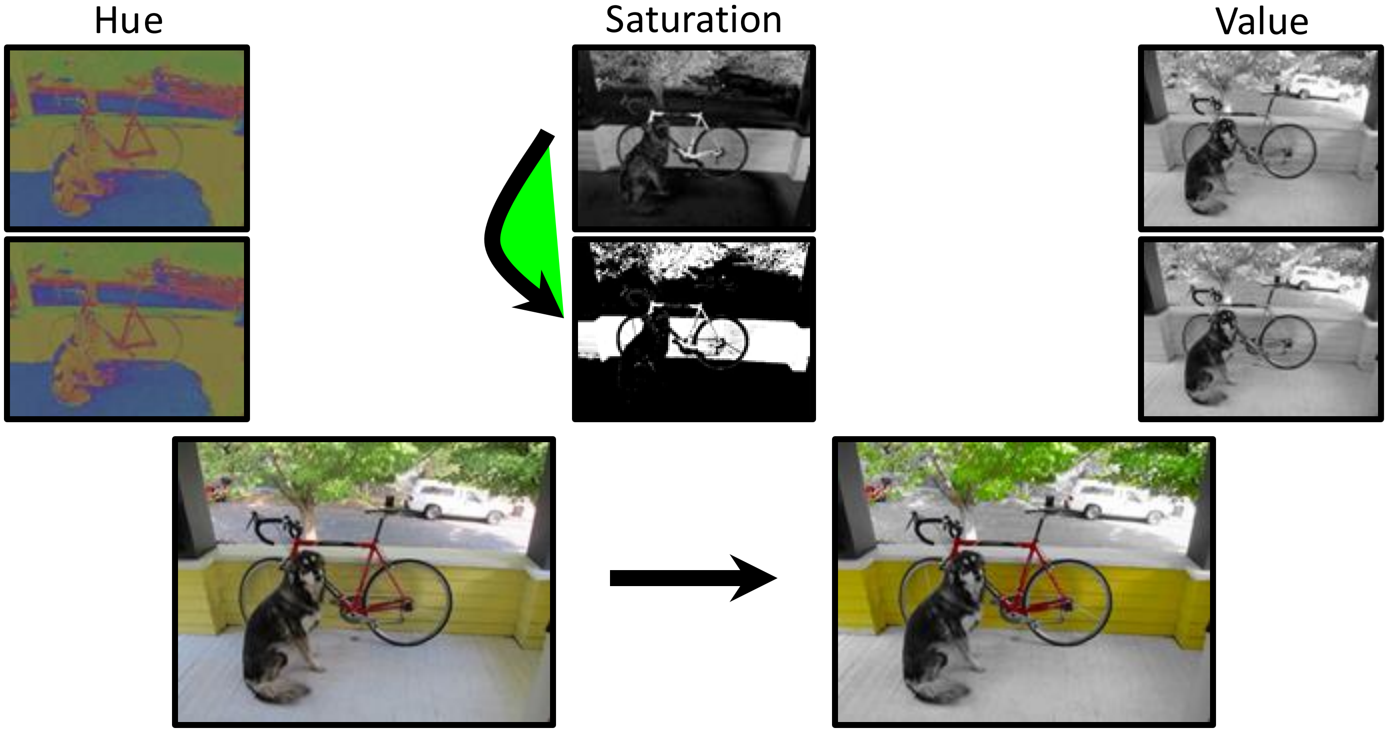
Set Hue to your favorite color!

HSV Color Space



Set Hue to your favorite pattern!

HSV Color Space



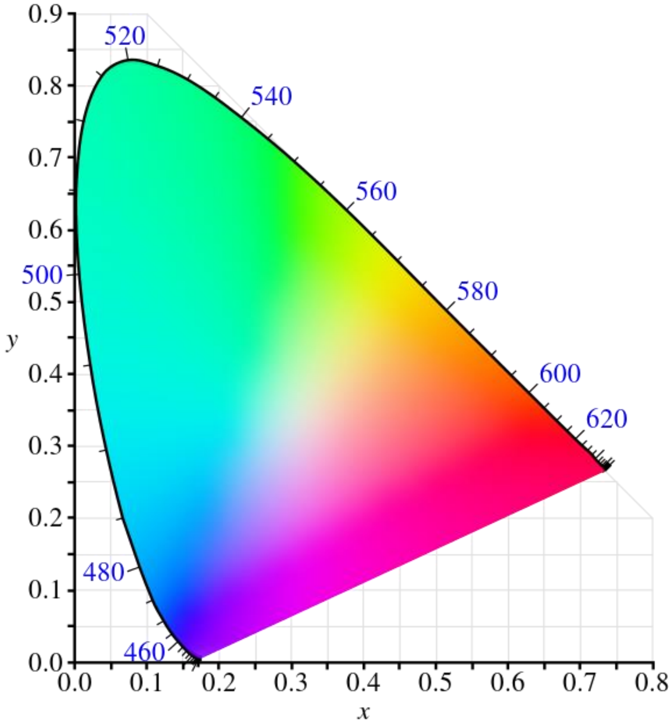
Increase & Threshold Saturation

HSV Color Space

CIE Color Space – Chromaticity Diagram

Wikipedia:

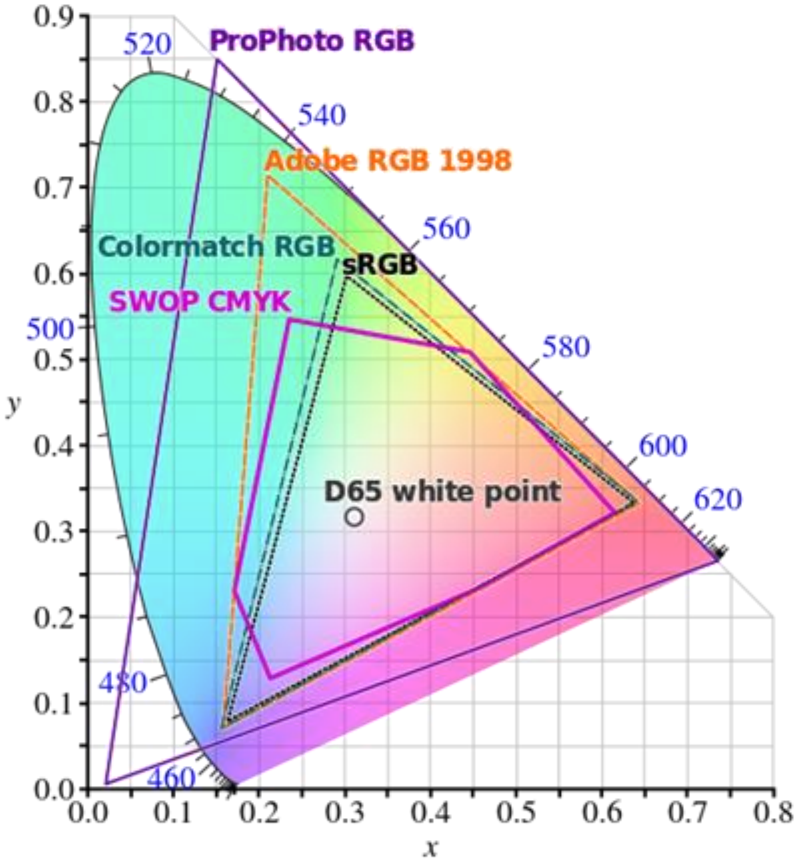
- CIE 1931 color space chromaticity diagram with wavelengths in nanometers.
- The colors depicted depend on the color space of the device on which the image is viewed.



[International Commission on Illumination](#)

Commission Internationale de l'Éclairage (CIE)

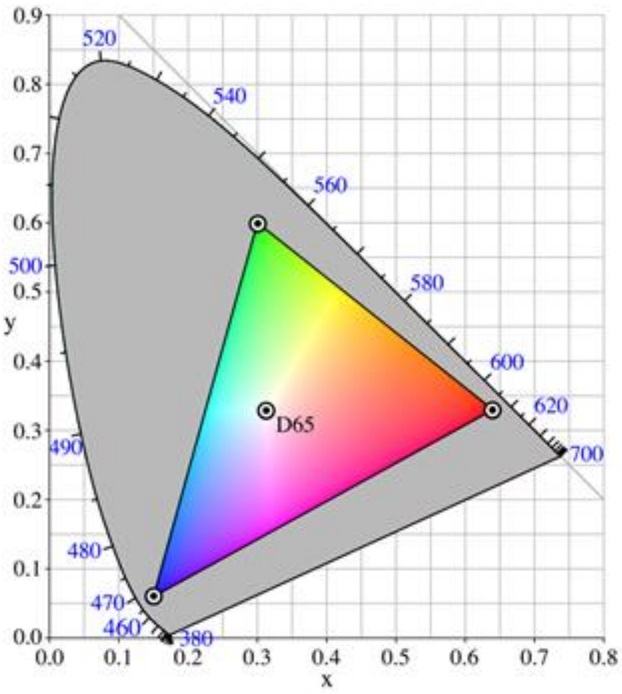
MANY Color Spaces



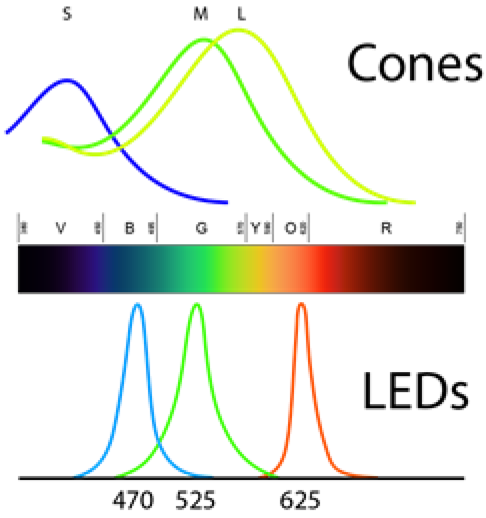
sRGB Primaries

Most widely used color space

Not all colors can be represented with RGB (or any Color Space)



MSFT 1996

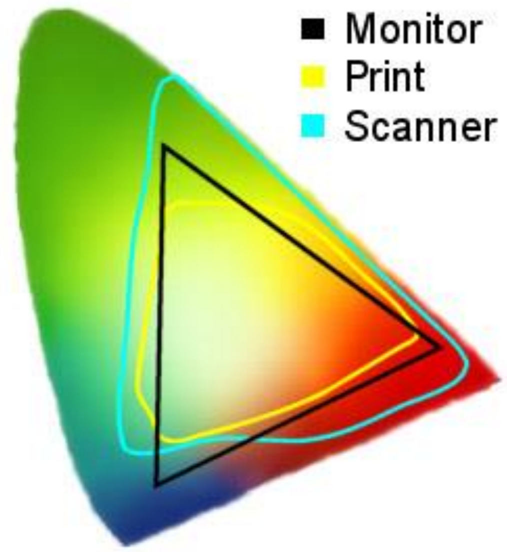


Monitor / Printer / Scanner Gamut

- The gamut is a triangle – All colors a device generates are *positive linear combinations* from *3 light sources* (e.g. a monitor with 3 LED types for R,G,B)

- The 3 light sources are the triangle's vertices

- The 3 light sources are usually not pure color (single wavelength), so that the vertices are usually not on the boundary of the CIE-xy chart
- In a physical color context, they are not pure color.
However, on the device, a single LED color is the purest color it can generate, so they are usually referred as 'pure color' in such context



What does this mean for computers?

- We represent images as grids of pixels
  - Each pixel has a 3-component color: RGB
- Can represent color with 3 numbers
  - #ff00ff; (1.0, 0.0, 1.0); (255, 0, 255); etc…
- Not every color can be represented in RGB
- RGB is made to ‘trick’ humans – not to be accurate

Closing Remarks

Action points

- Check out & start working on Lab0 & Lab1 assignments
- Feel free to discuss on Ed
- Lecture 3 (HC2a) covers the relevant theory

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.

Pinhole Camera

Pinhole camera:

Simple model, approx. perspective projection for image formation

Pinhole is a **point**:

Only 1 ray from any given point enters the camera

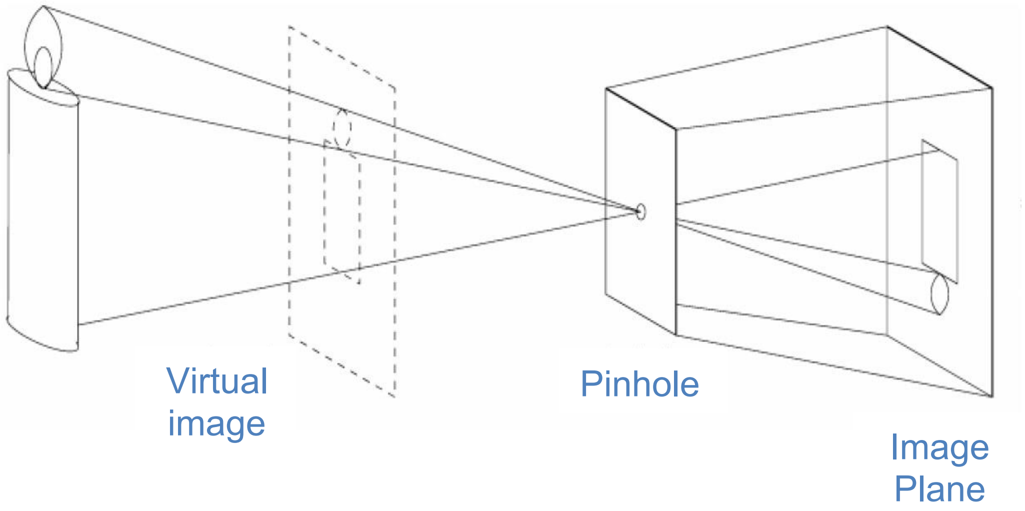


Figure from Forsyth & Ponce