CMSC 491/691

Lecture 2

Image Formation

Some slides from Jayasuriya, Szeliski, Derpanis

Course Staff

Instructor: Tejas Gokhale
Assistant Professor, CSEE

Wednesday 2:30 – 3:30 PM

ITE 342-B

gokhale@umbc.edu

Office Hours

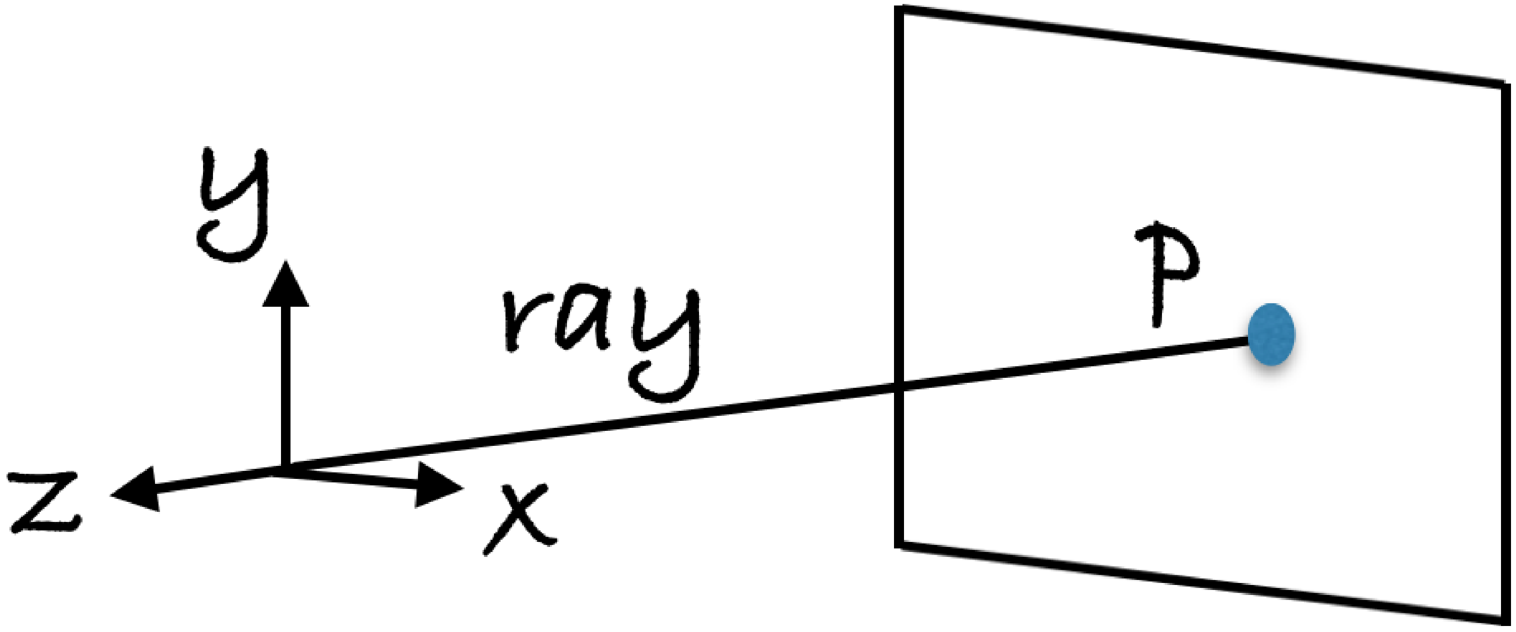
TA: Yu Liu
Ph.D. Student (Research: Computer Graphics)

Monday 2:30 – 3:30 PM
& Tuesday 1 – 2 PM

ITE 340

yul2@umbc.edu

last class …



Let’s say we have a sensor…

digital sensor
(CCD or CMOS)

… and an object we like to photograph

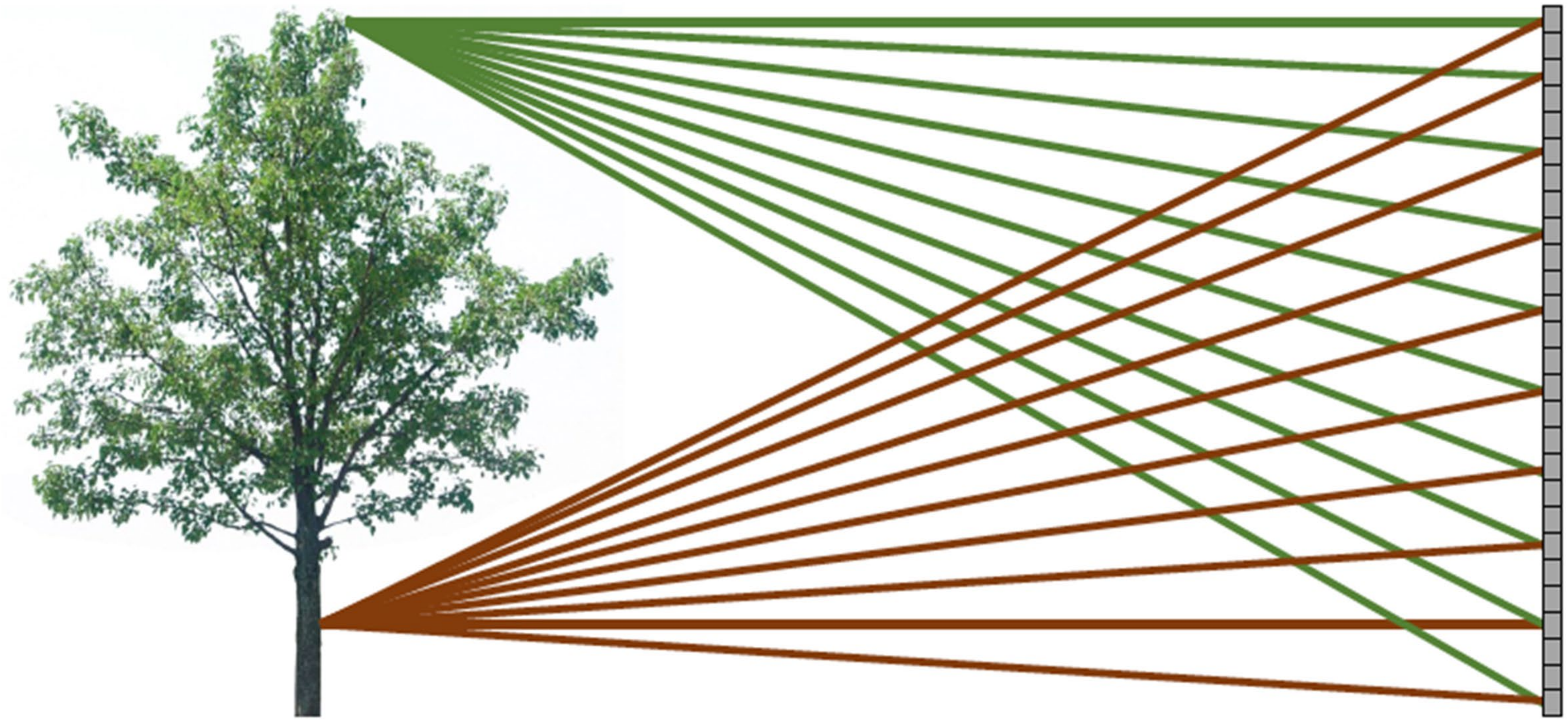
real-world object

digital sensor (CCD or CMOS)

What would an image taken like this look like?

Bare-sensor imaging

real-world object



digital sensor (CCD or CMOS)

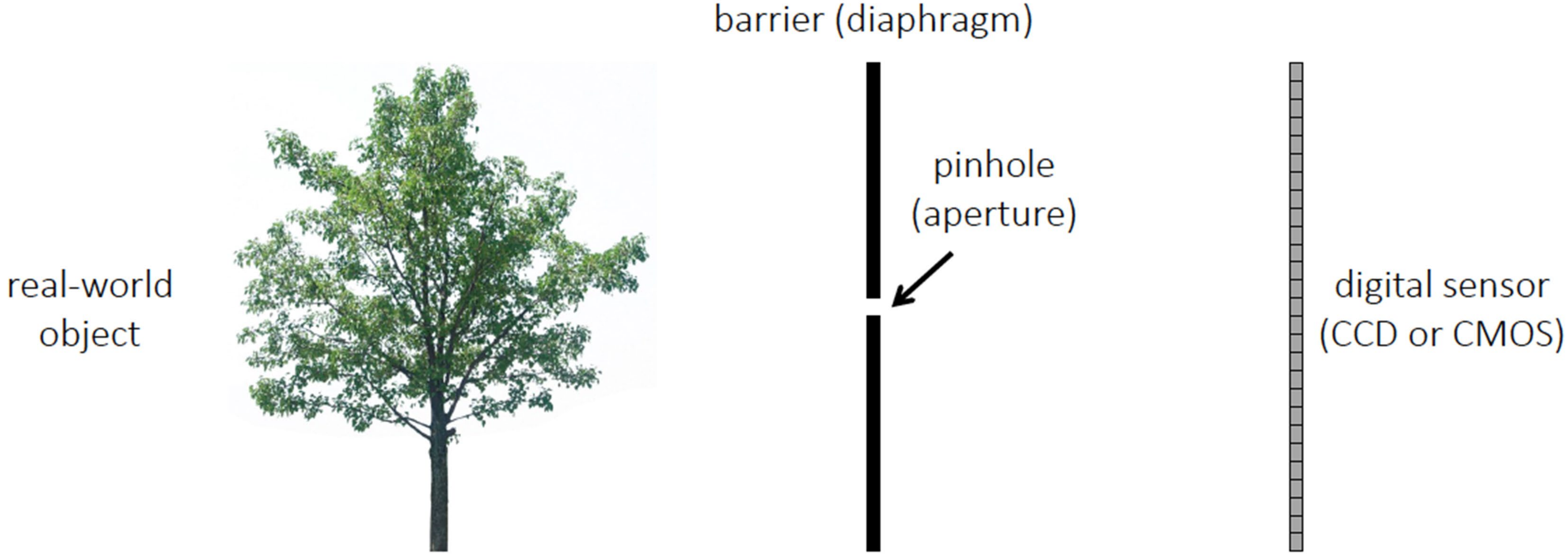
All scene points contribute to all sensor pixels

What does the image on the sensor look like?

Bare-sensor imaging

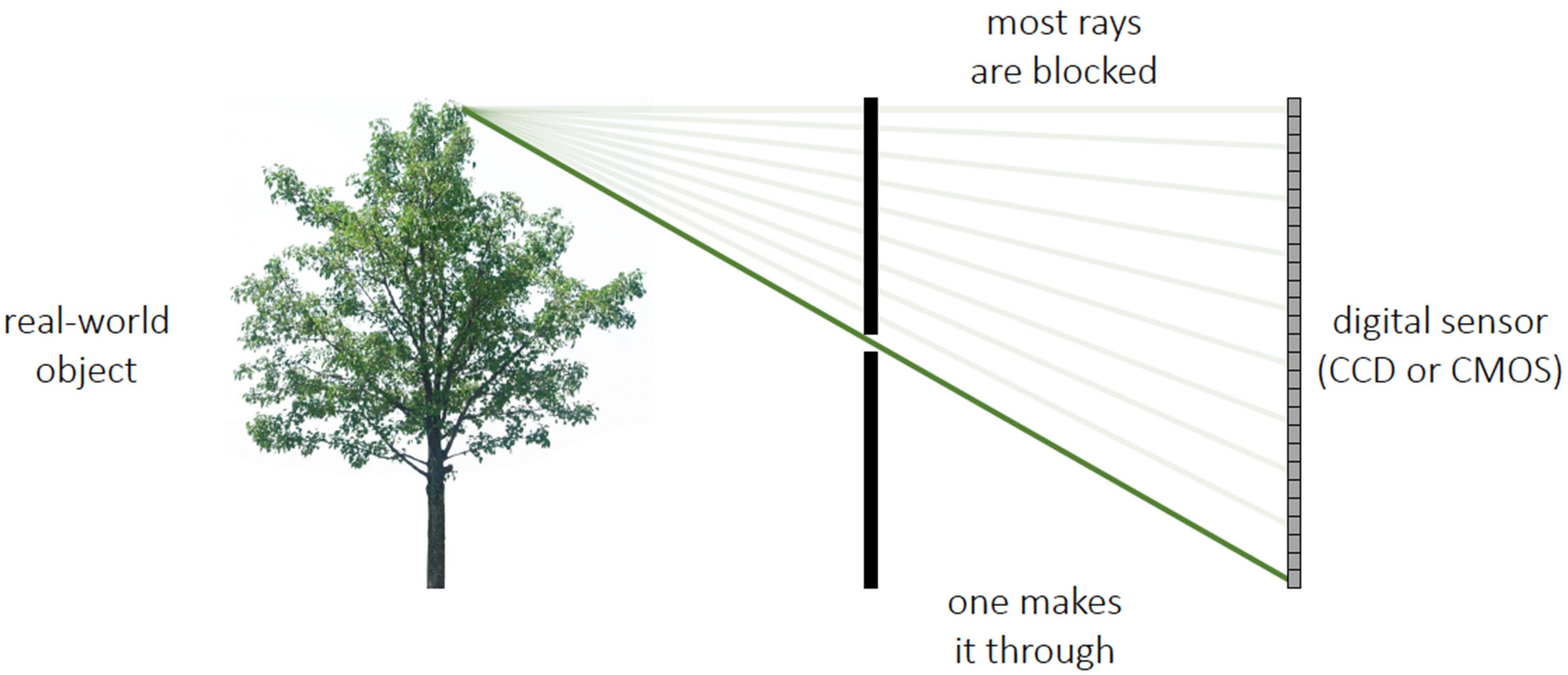
All scene points contribute to all sensor pixels

Let’s add something to this scene

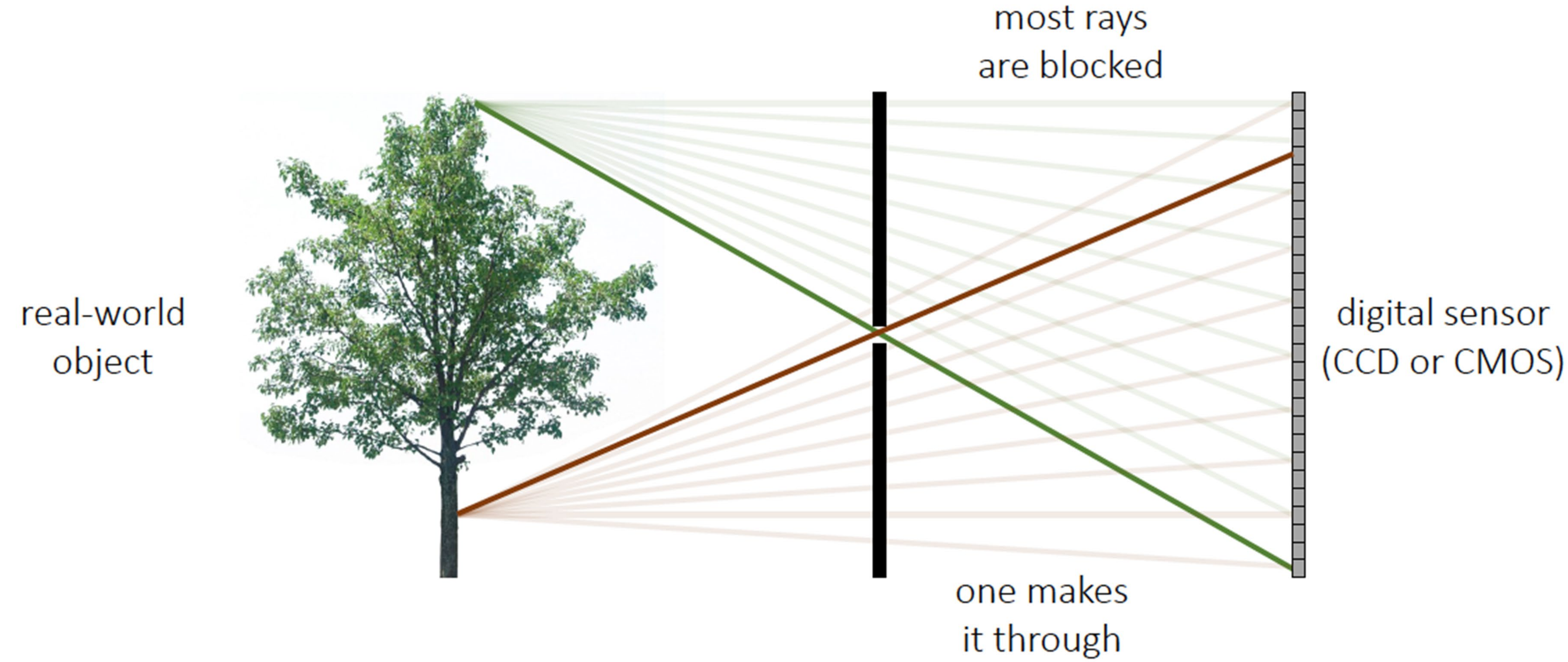


What would an image taken like this look like?

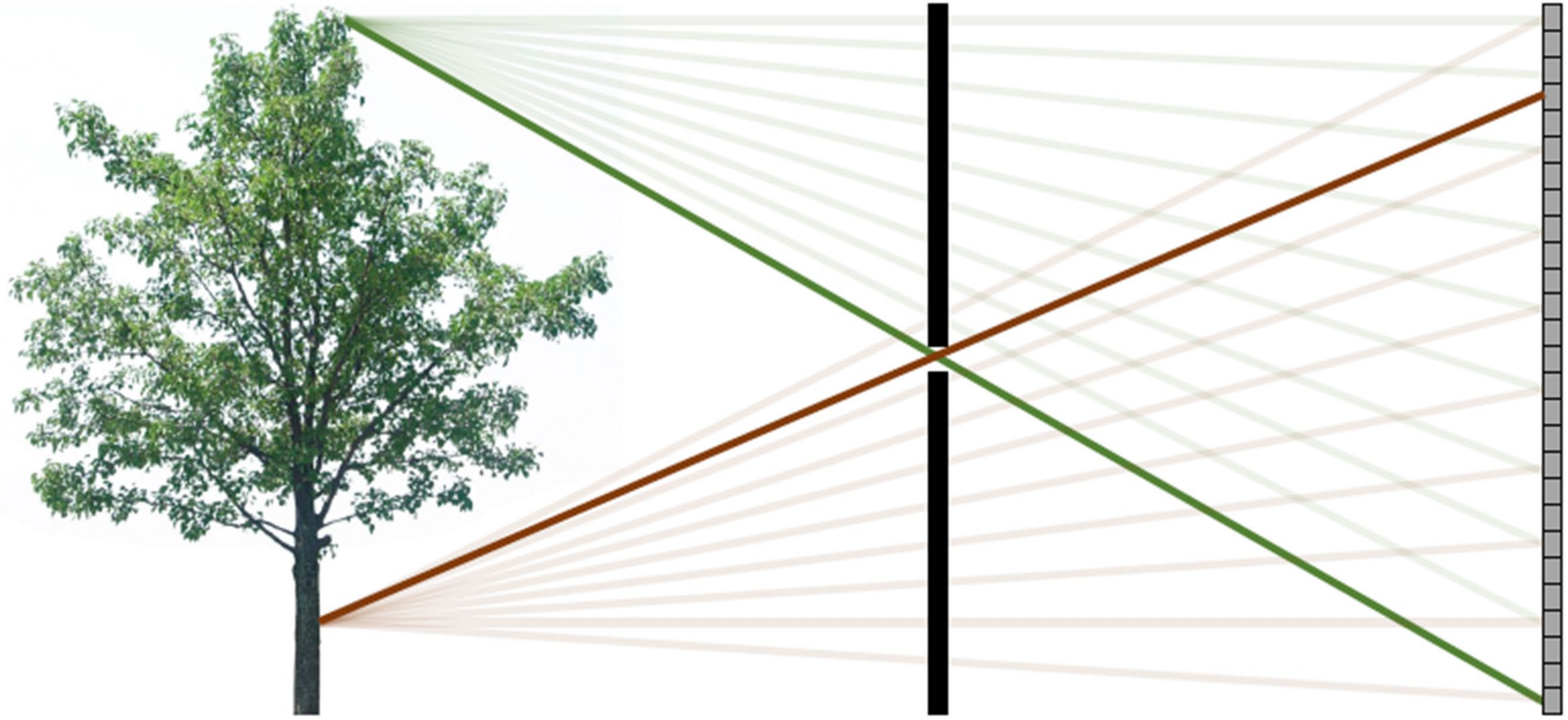
Pinhole imaging



Pinhole imaging



Pinhole imaging



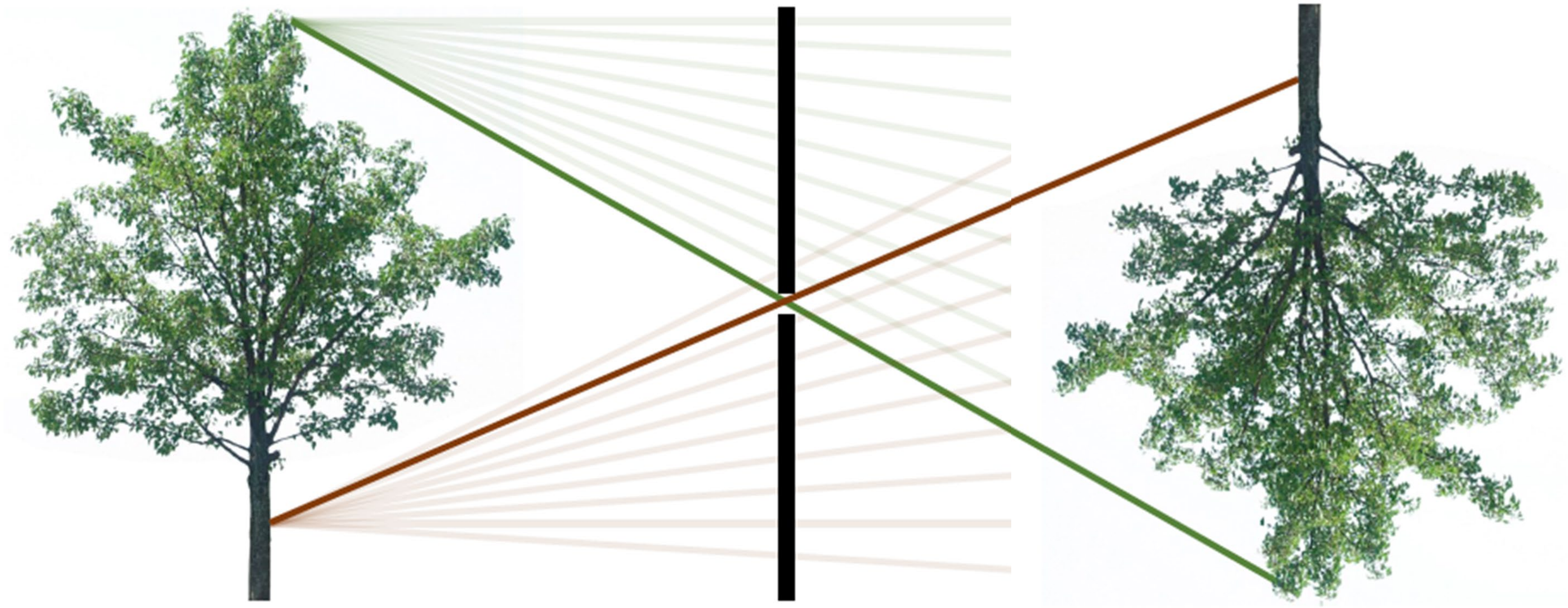
Each scene point contributes to only one sensor pixel

What does the image on the sensor look like?

Pinhole imaging

Pinhole imaging

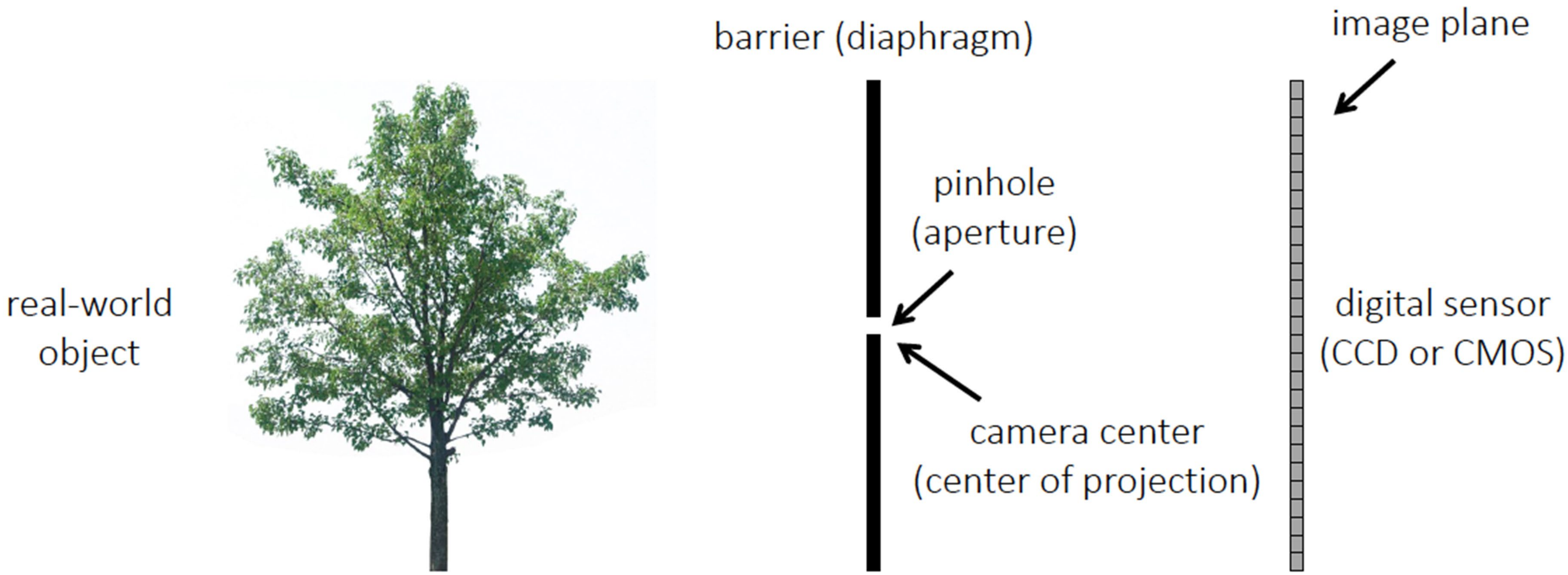
real-world
object



copy of real-world object
(inverted and scaled)

Each scene point contributes to only one sen

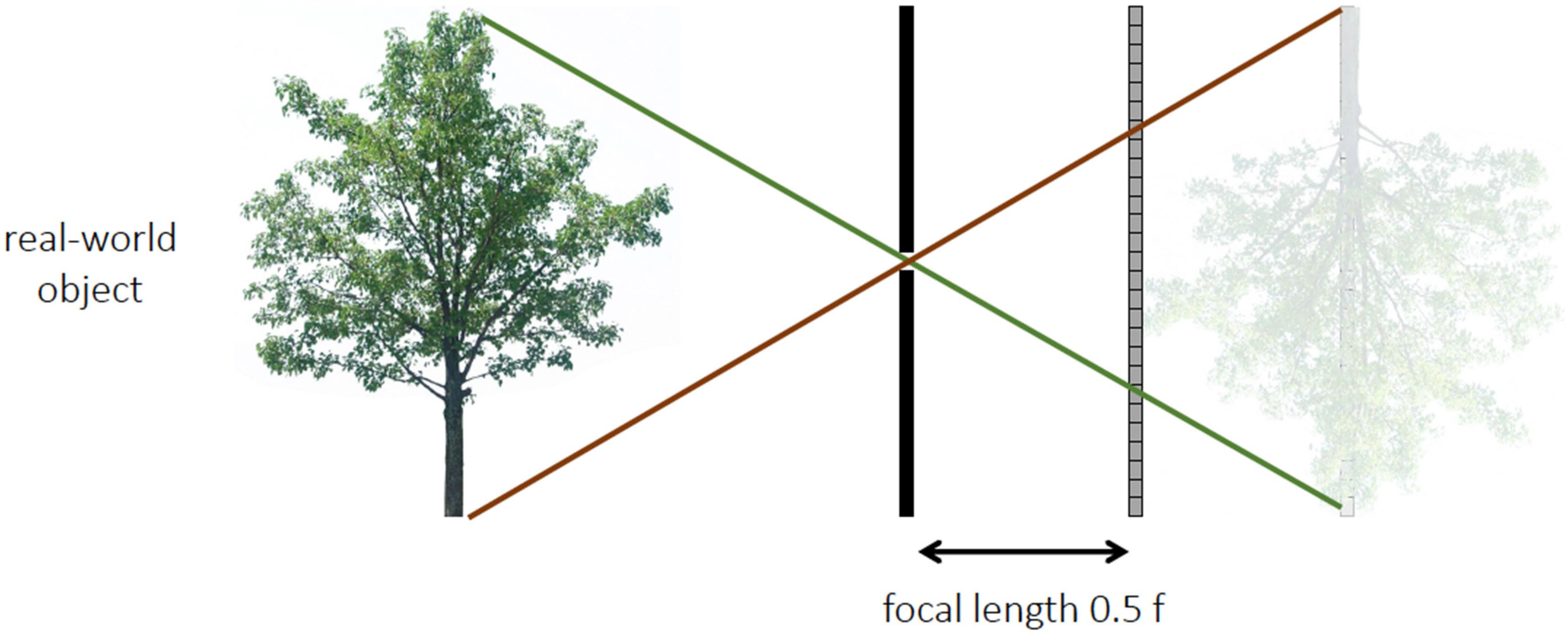
Pinhole camera terms



Focal length

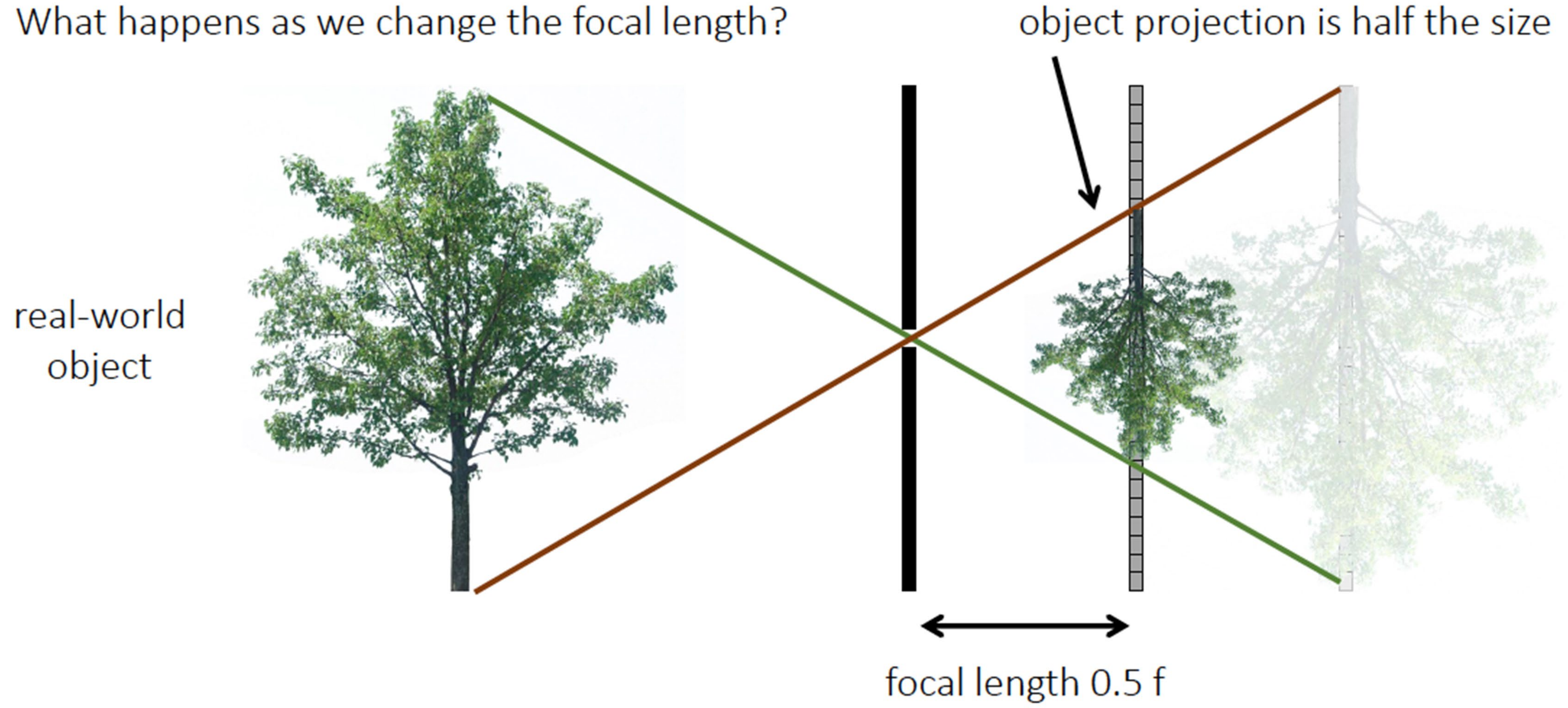
Focal length

What happens as we change the focal length?



Focal length

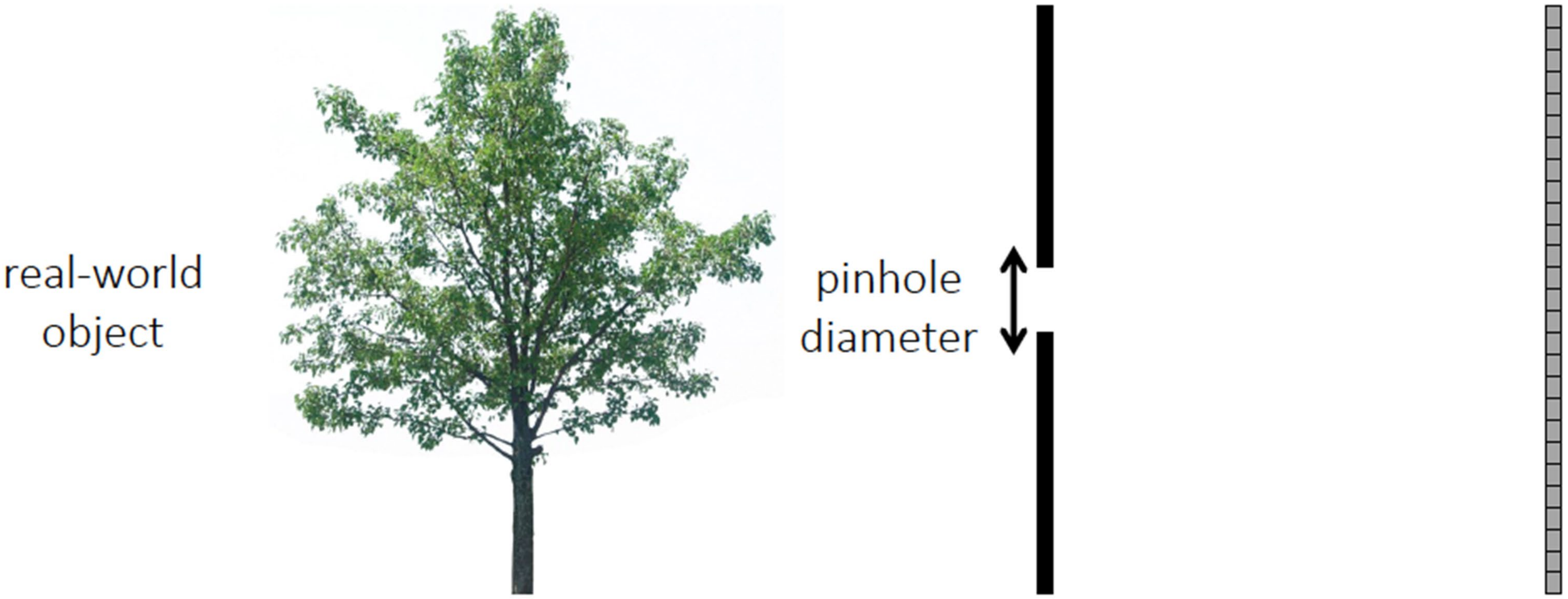
What happens as we change the focal length?



Magnification depends on focal length

Pinhole size

What happens as we change the pinhole diameter?

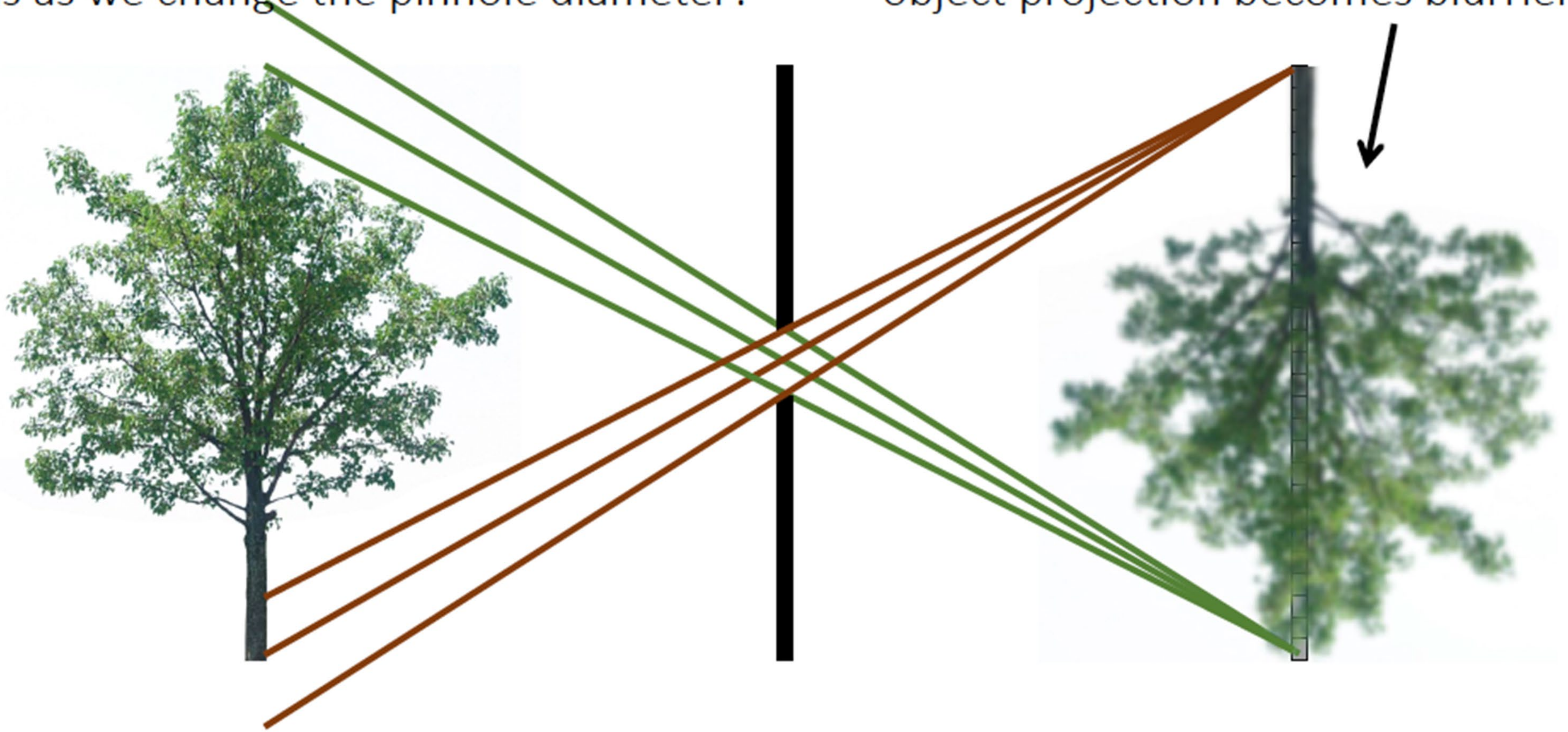


Pinhole size

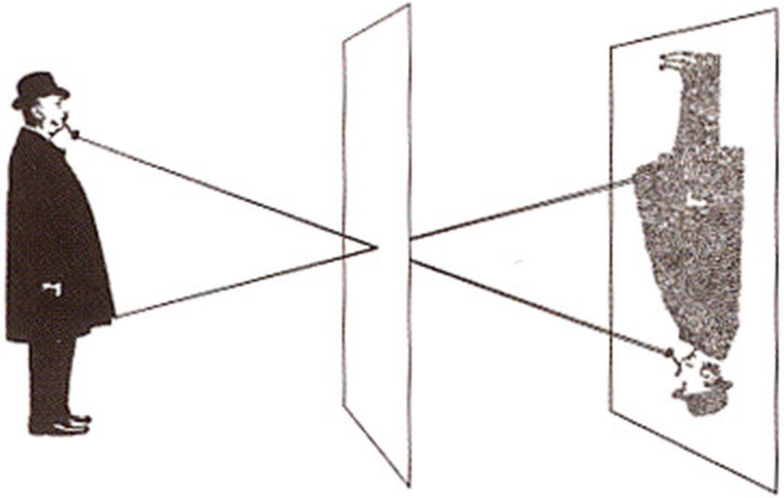
What happens as we change the pinhole diameter?

object projection becomes blurrier

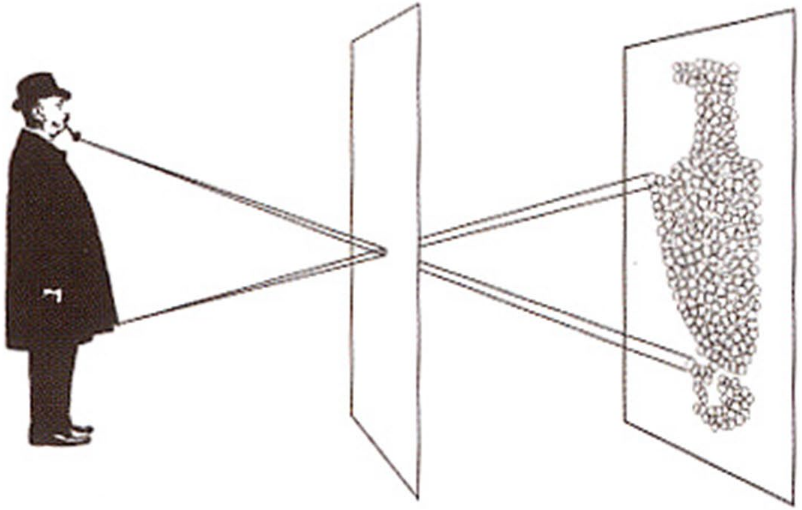
real-world object



Photograph made with small pinhole

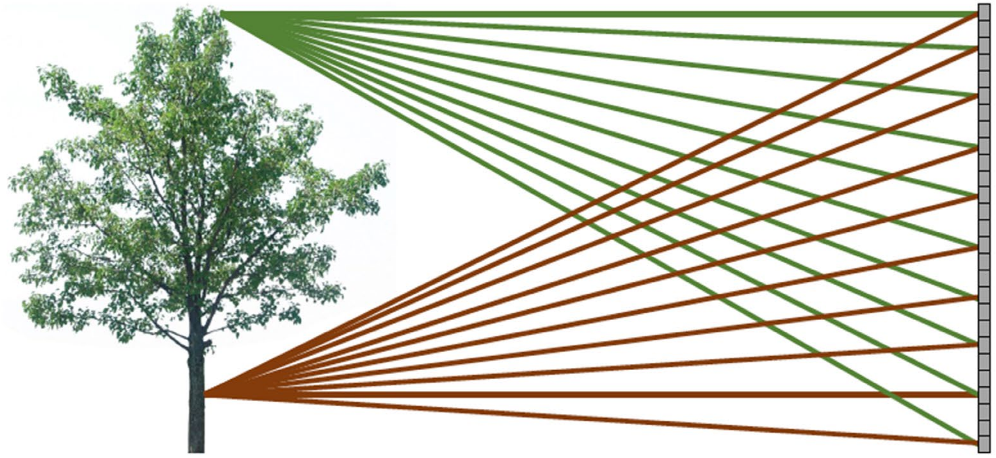


Photograph made with larger pinhole



(Recall) Extreme Case: Infinite Pinhole

Bare-sensor imaging



All scene points contribute to all sensor pixels

Resulting Image

All scene points contribute to all sensor pixels

Problems with Pinholes

- Pinhole size (aperture) must be "very small" to obtain clear images
  - If aperture size is large, images will be blurry
- But if pinhole is made smaller, *less light* is received by the image plane
- If pinhole is as small as the wavelength of light … DIFFRACTION blurs the image!
- Thumb rule for sharp images:

  Pinhole diameter $d = 2\sqrt{f'\lambda}$

  Example: If $f' = 50mm; \lambda_{red} = 600nm$

  then $d = 0.36mm$

Fig. 5.96 The pinhole camera. Note the variation in image clarity as the hole diameter decreases. [Photos courtesy Dr. N. Joel, UNESCO.]

Ok. Pinholes are Cute and Simple.

But they have problems …

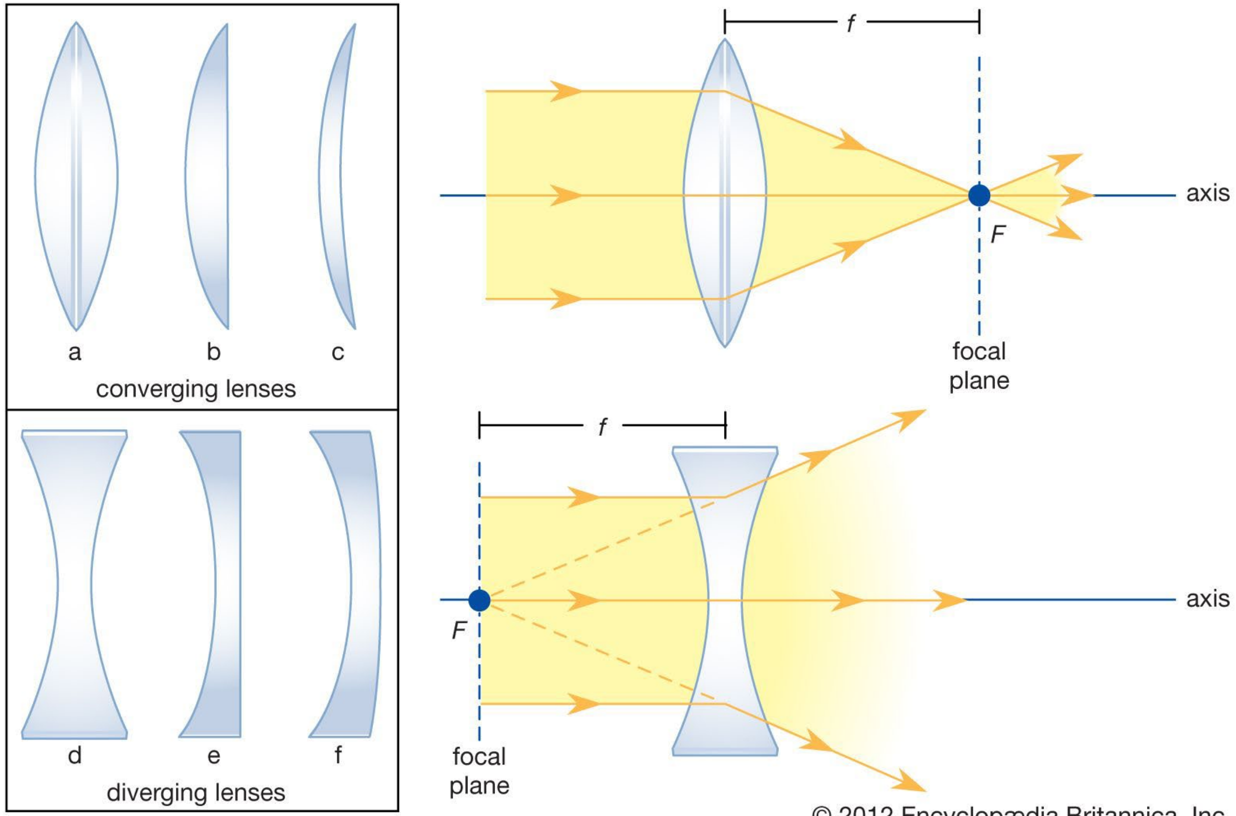
Solution?

Ok. Pinholes are Cute and Simple.

But they have problems …

Solution?

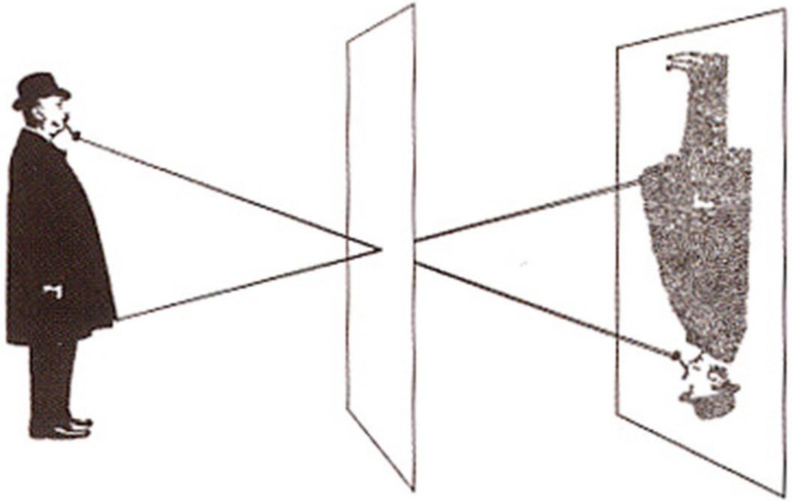
Lenses!



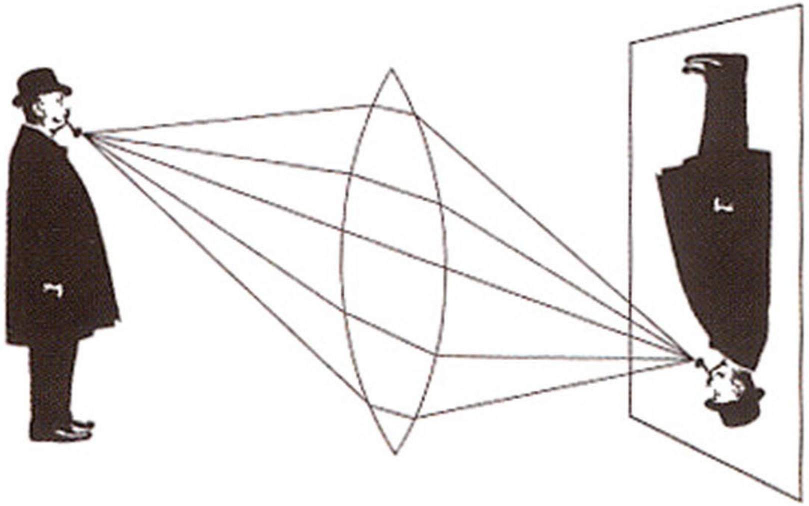
Lenses are Cool

Lenses are Cool

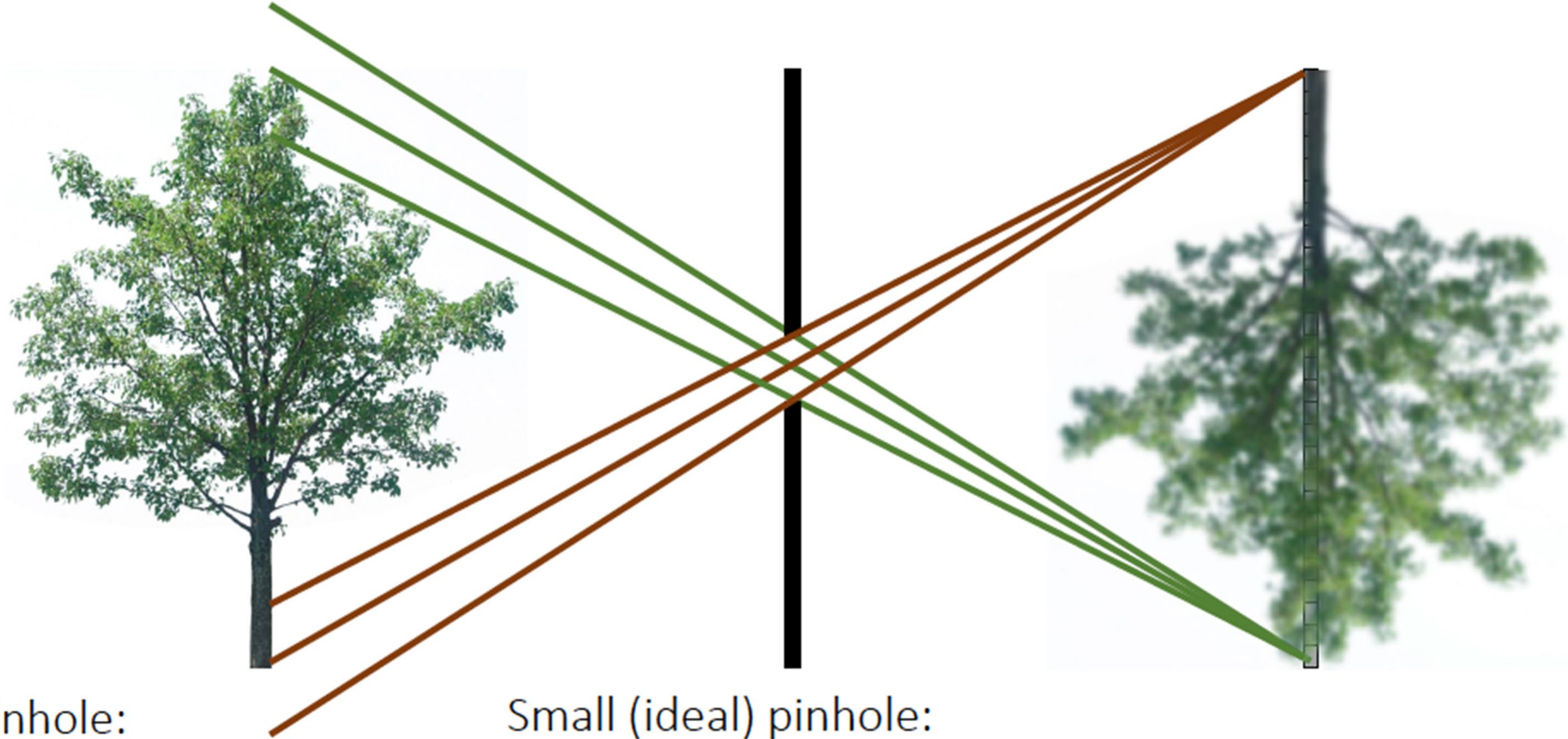
Photograph made with small pinhole



Photograph made with lens



Pinhole camera

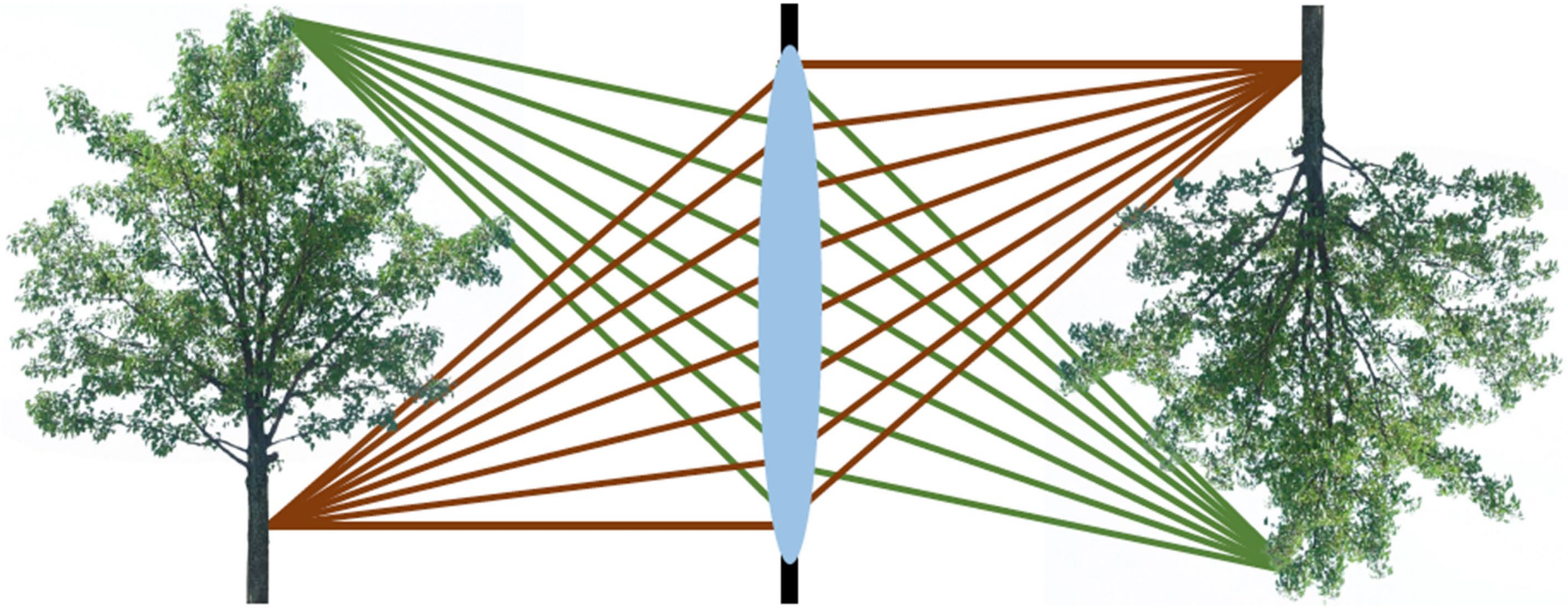


Large pinhole:
1. Image is blurry.
2. Signal-to-noise ratio is high.

Small (ideal) pinhole:
1. Image is sharp.
2. Signal-to-noise ratio is low.

Best of Both Worlds?

Almost, by using lenses

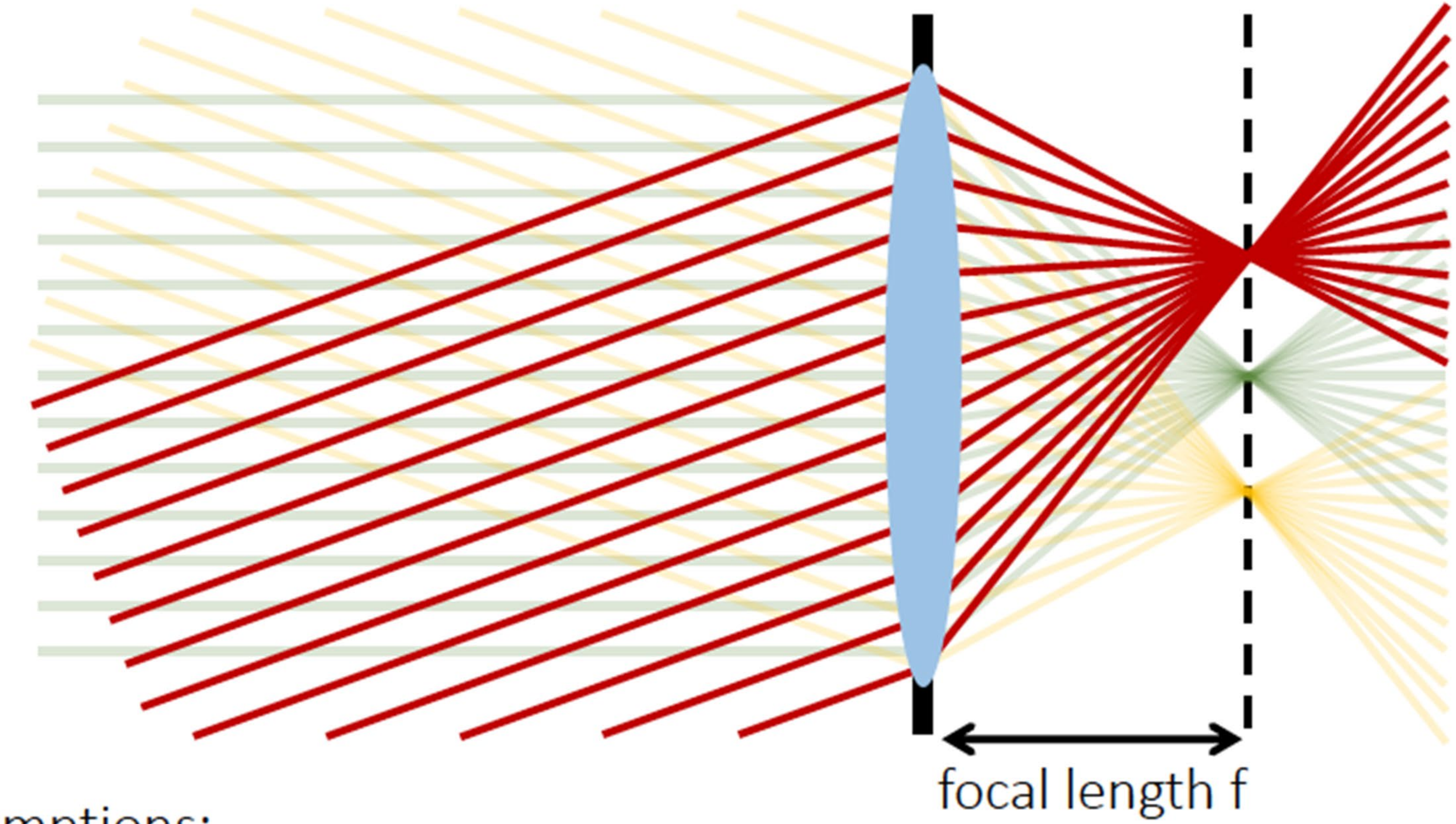


Lenses map “bundles” of rays from points on the scene to the sensor.

How does this mapping work exactly?

Thin lens model

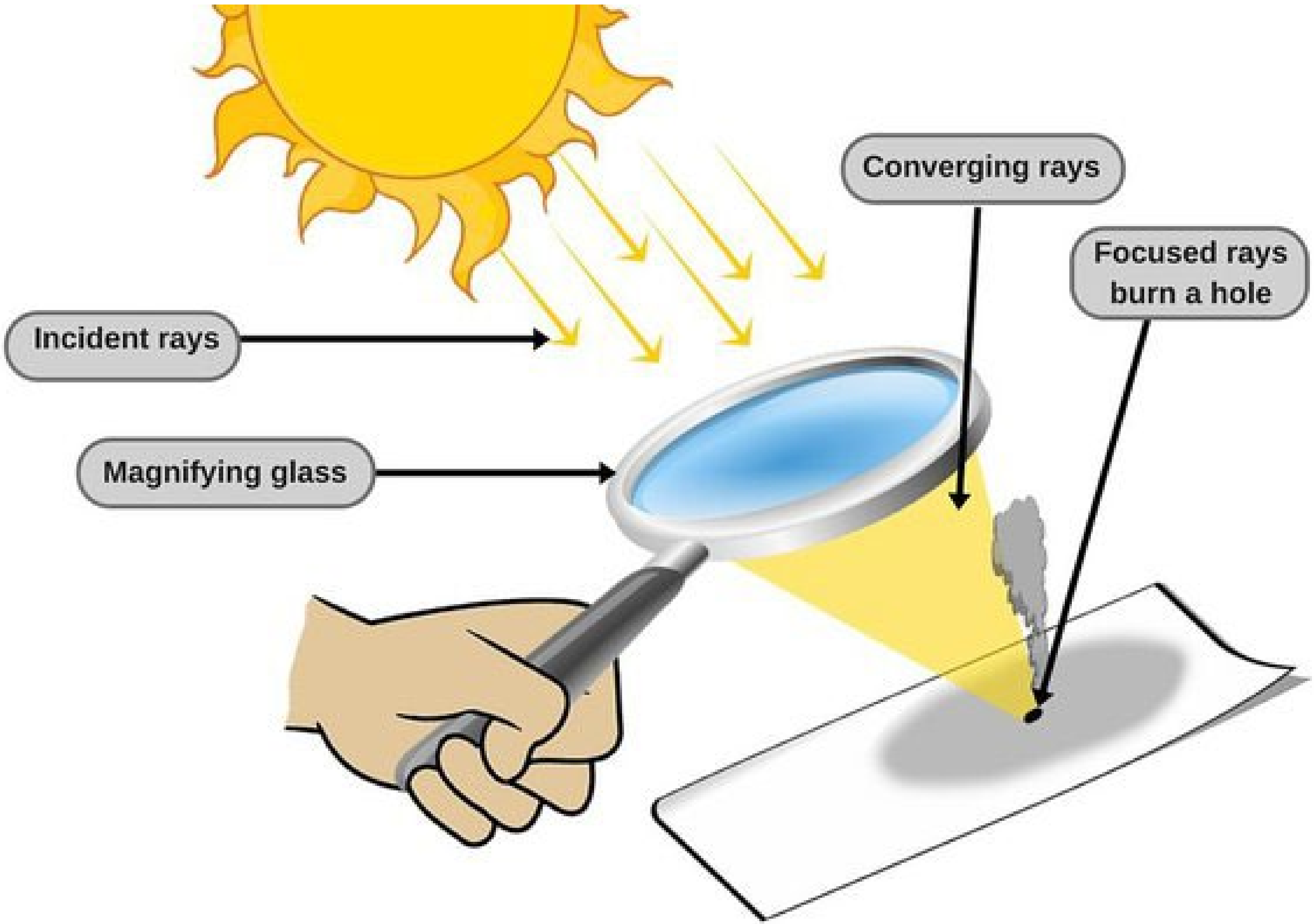
Simplification of geometric optics for well-designed lenses.

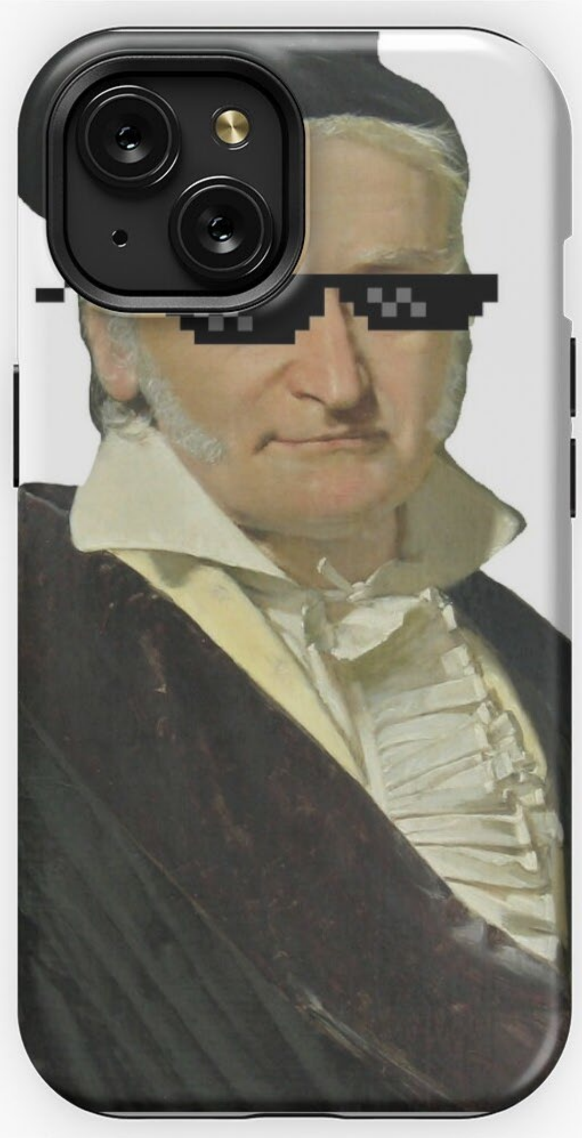


Two assumptions:

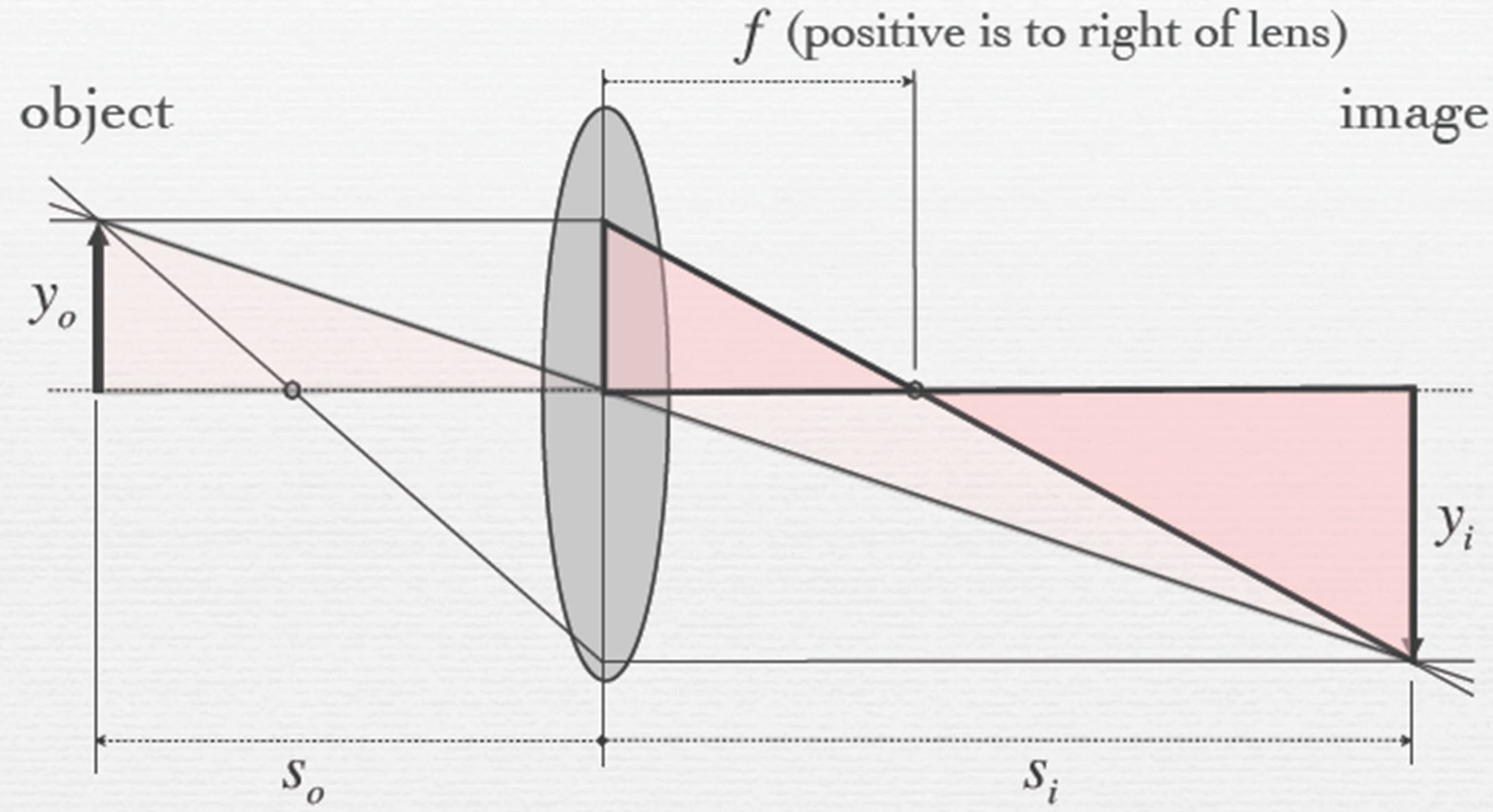
1. Rays passing through lens center are unaffected.
2. Parallel rays converge to a single point located on focal plane.

Can we verify the thin lens model?



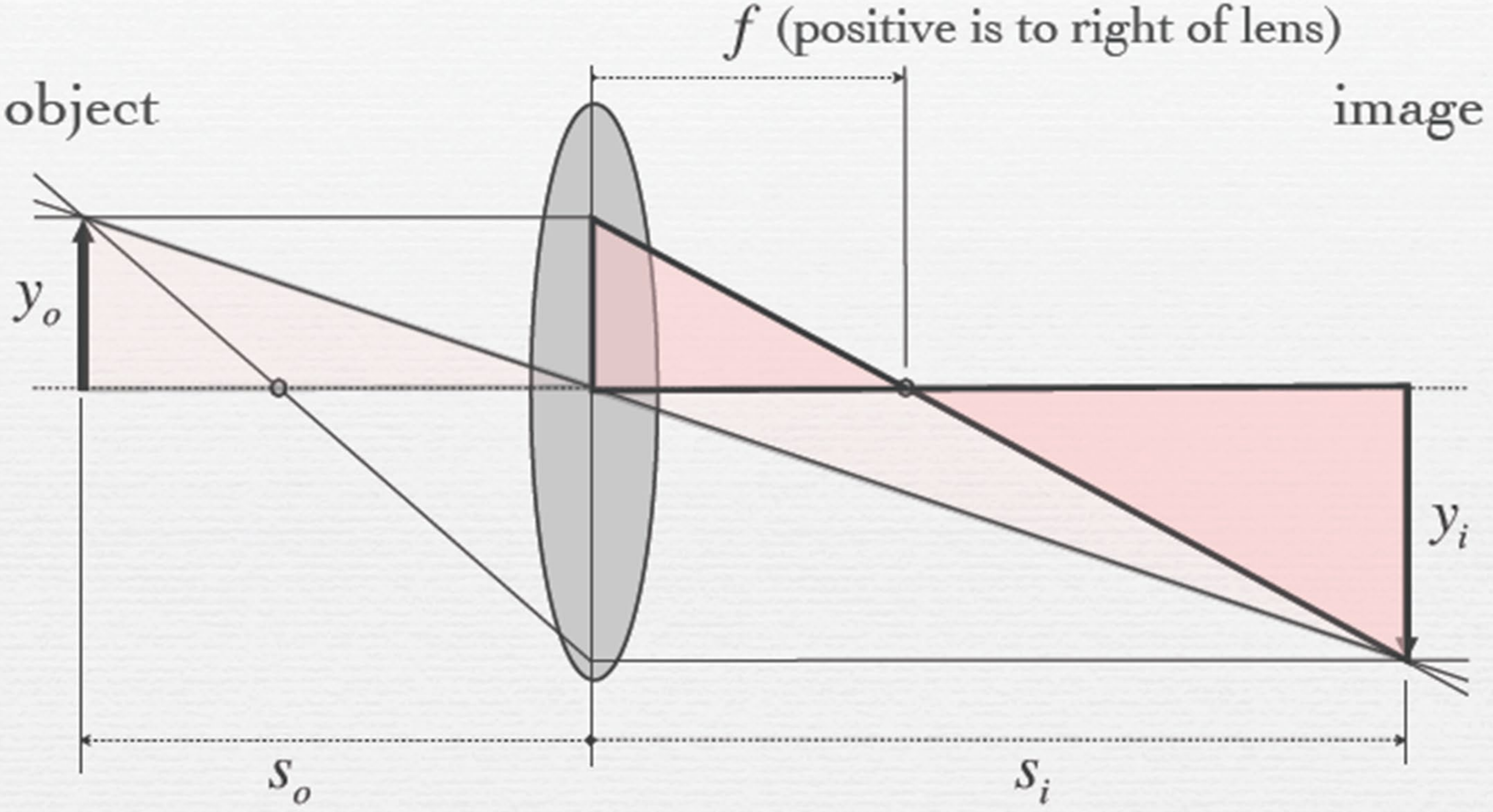


From Gauss's ray construction to the Gaussian lens formula



Exercise: Derive Relationship between s_o, s_i, f

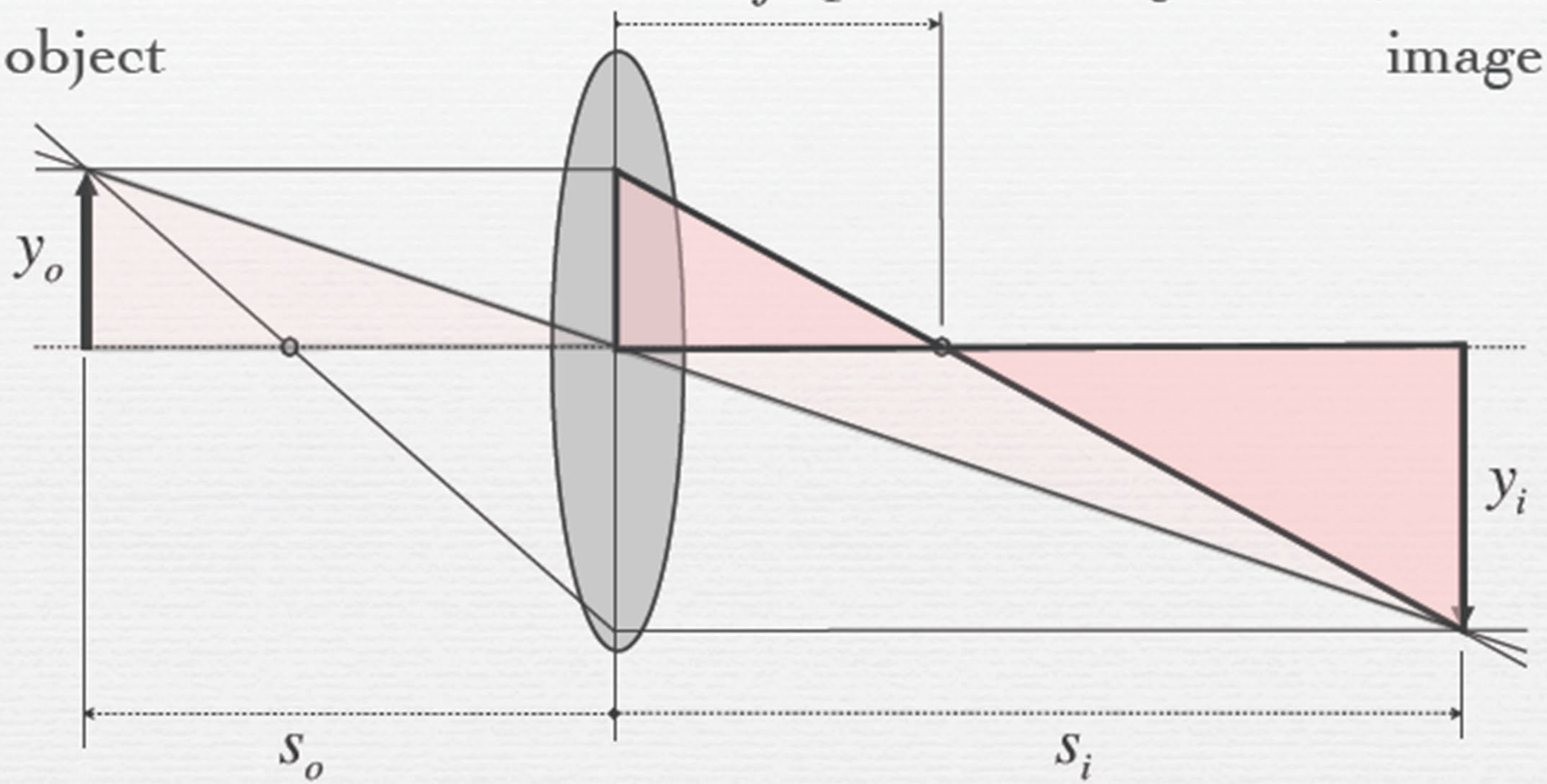
From Gauss's ray construction to the Gaussian lens formula



Exercise: Derive Relationship between s_o , s_i , f

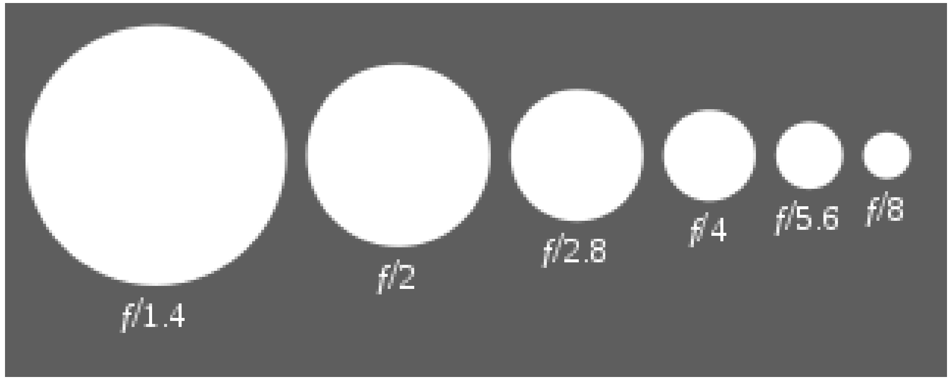
Hint: Similar Triangles

From Gauss's ray construction to the Gaussian lens formula



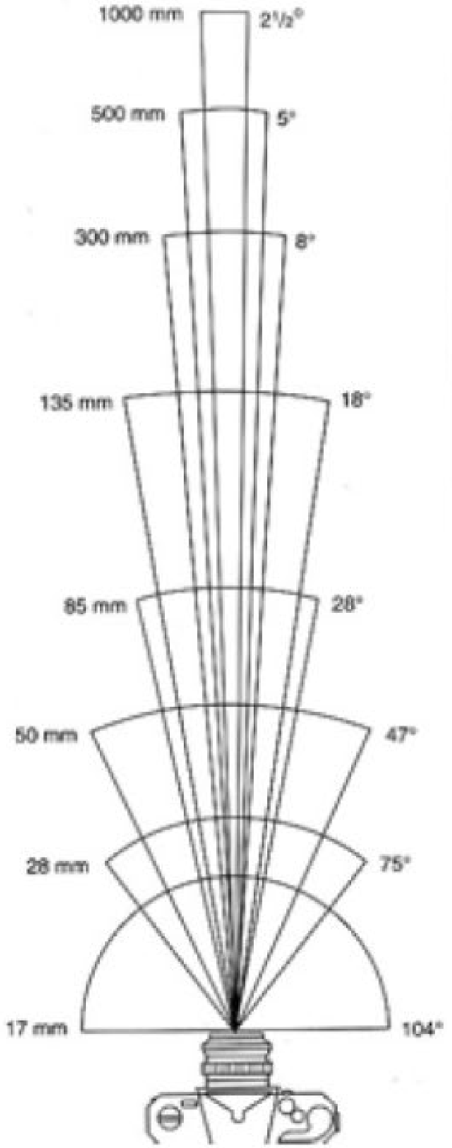
$$\frac{|y_i|}{y_o} = \frac{s_i}{s_o} \quad \text{and} \quad \frac{|y_i|}{y_o} = \frac{s_i - f}{f} \quad \ldots\ldots \quad \boxed{\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}}$$

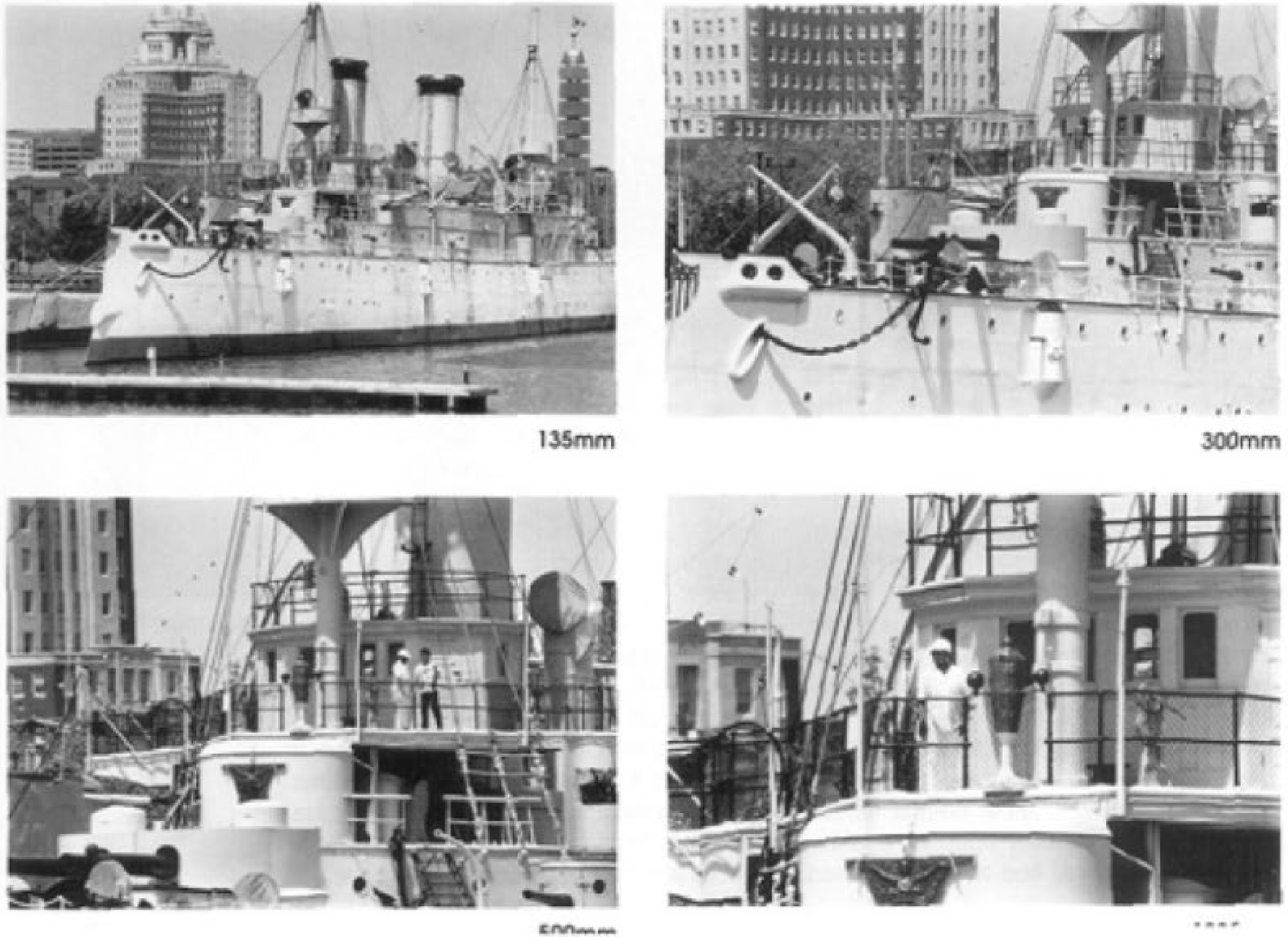
Depth of Field **(effect of varying aperture diameter)**



Smaller aperture → larger DoF

Field of View

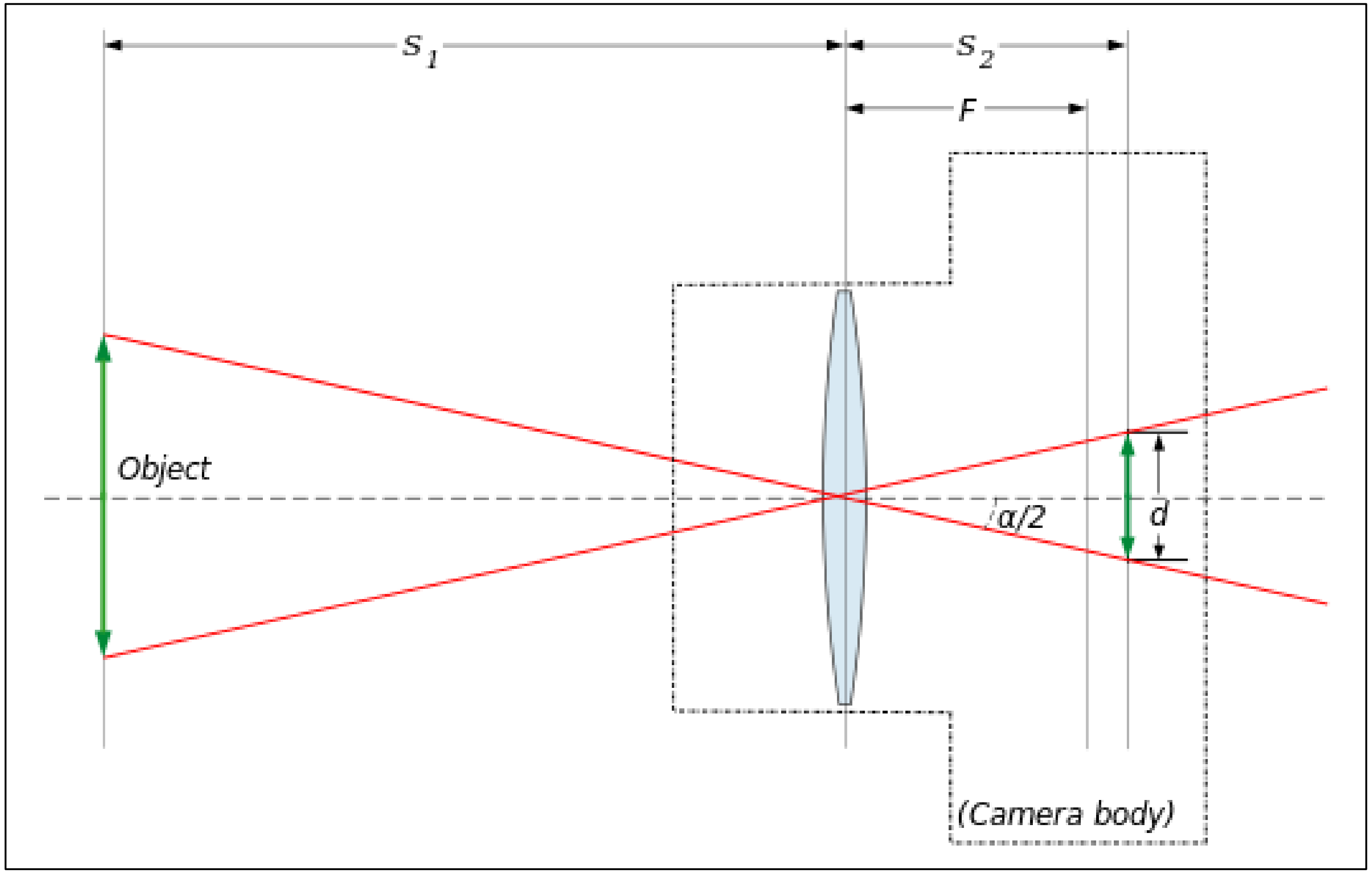




What does FOV depend on?

Slide by A. Efros

Field of View (effect of varying focal length)

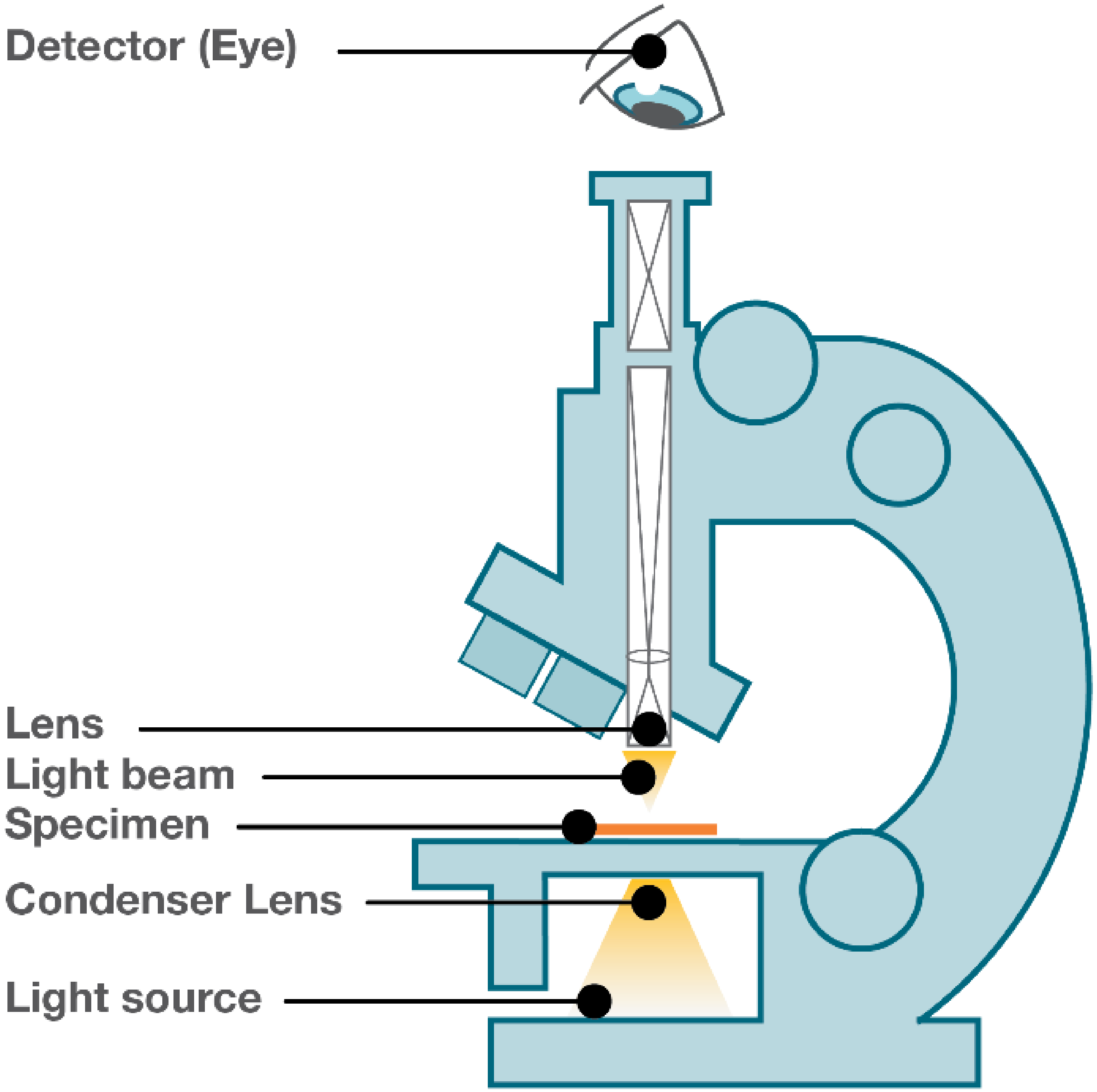


Smaller f → larger DoF

$$\alpha = 2 \arctan \frac{d}{2f}$$

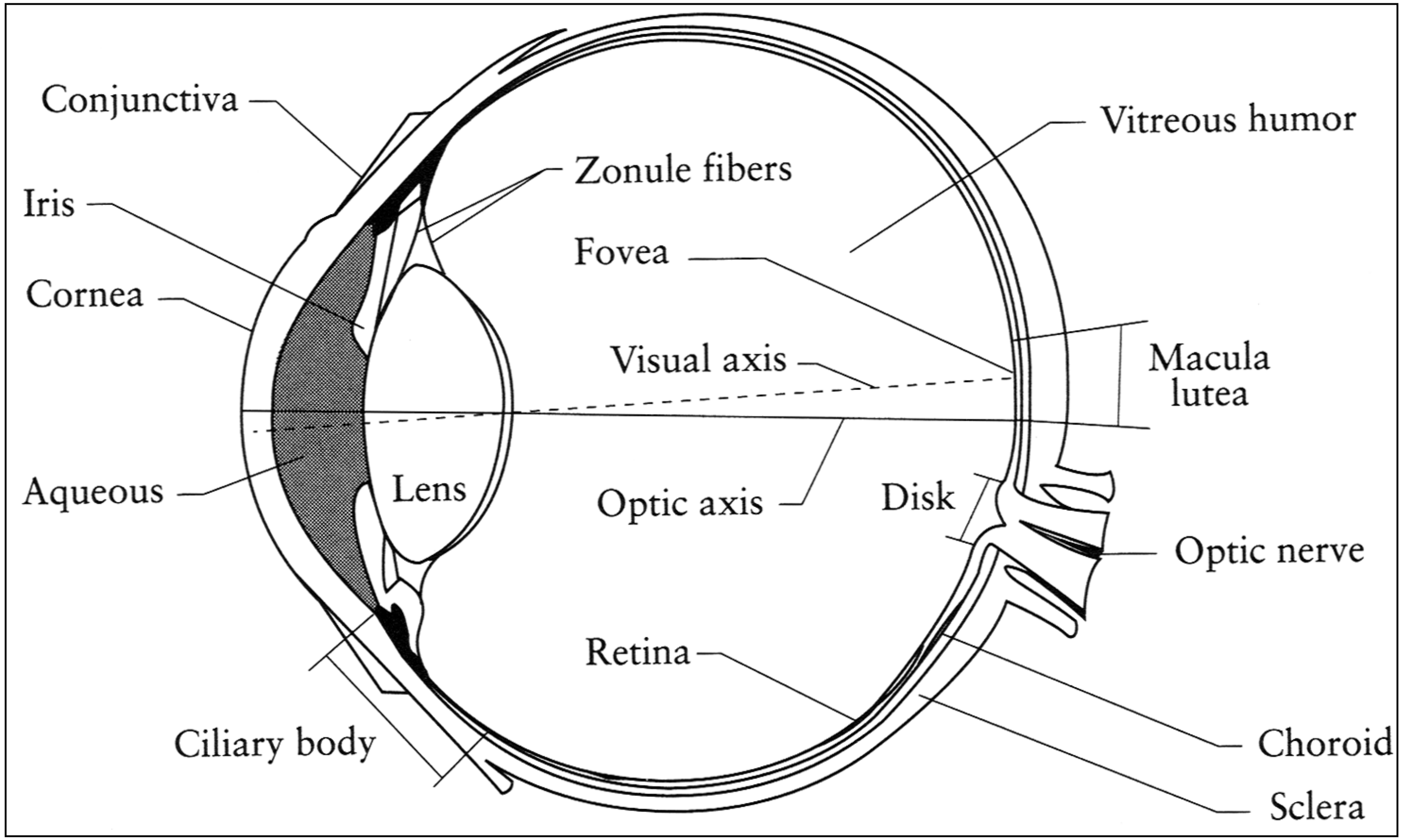
Microscopes?

Same Mechanism



Your Eyes?

The Eye is a “Camera”



- Iris
  - colored annulus with radial muscles
- Pupil
  - the hole (aperture)
  - size is controlled by the iris


Disposable cameras are fun, although it does seem wasteful
YEAH, IT'S EITHER TAPING OR CALLING.
And you don't ever get to see your pictures.
ALSO, HOW DO CAMERAS WORK?

Digital Images

What is Color?

Subjective terms to describe color

Hue
Name of the color
(yellow, red, blue, green, . . .)

Value/Lightness/Brightness
How light or dark a color is.

Saturation/Chroma/Color Purity
How “strong” or “pure” a color is.

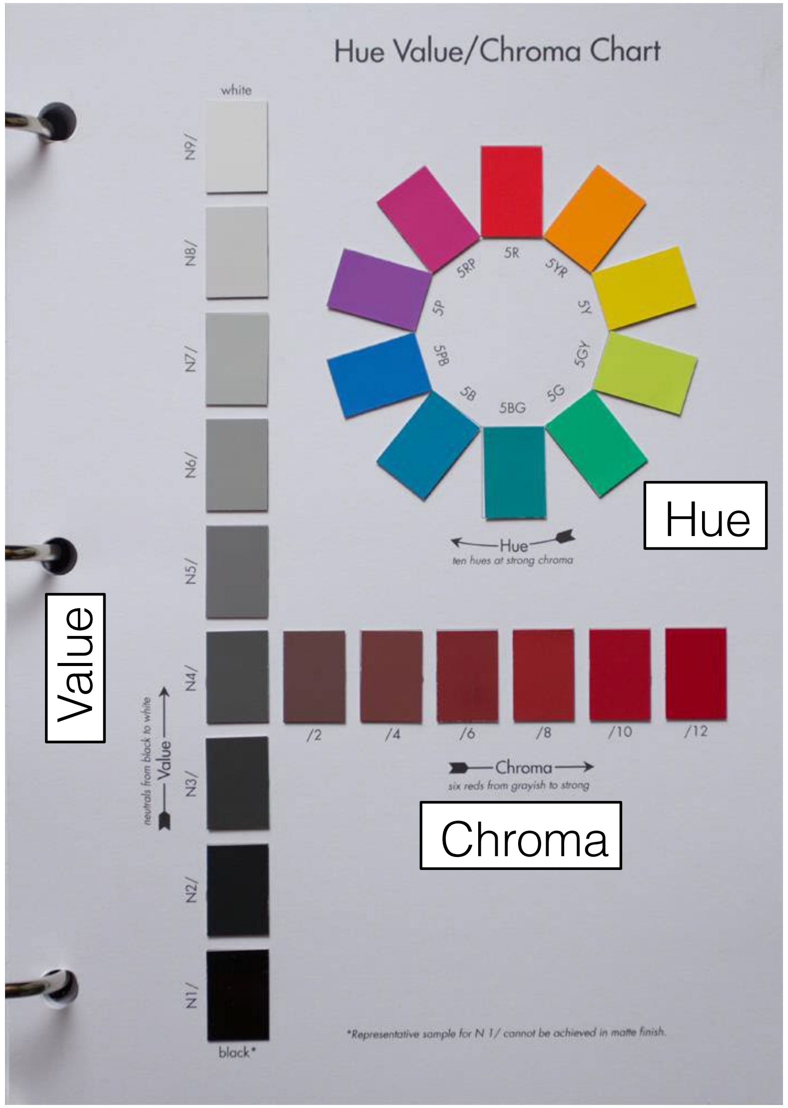
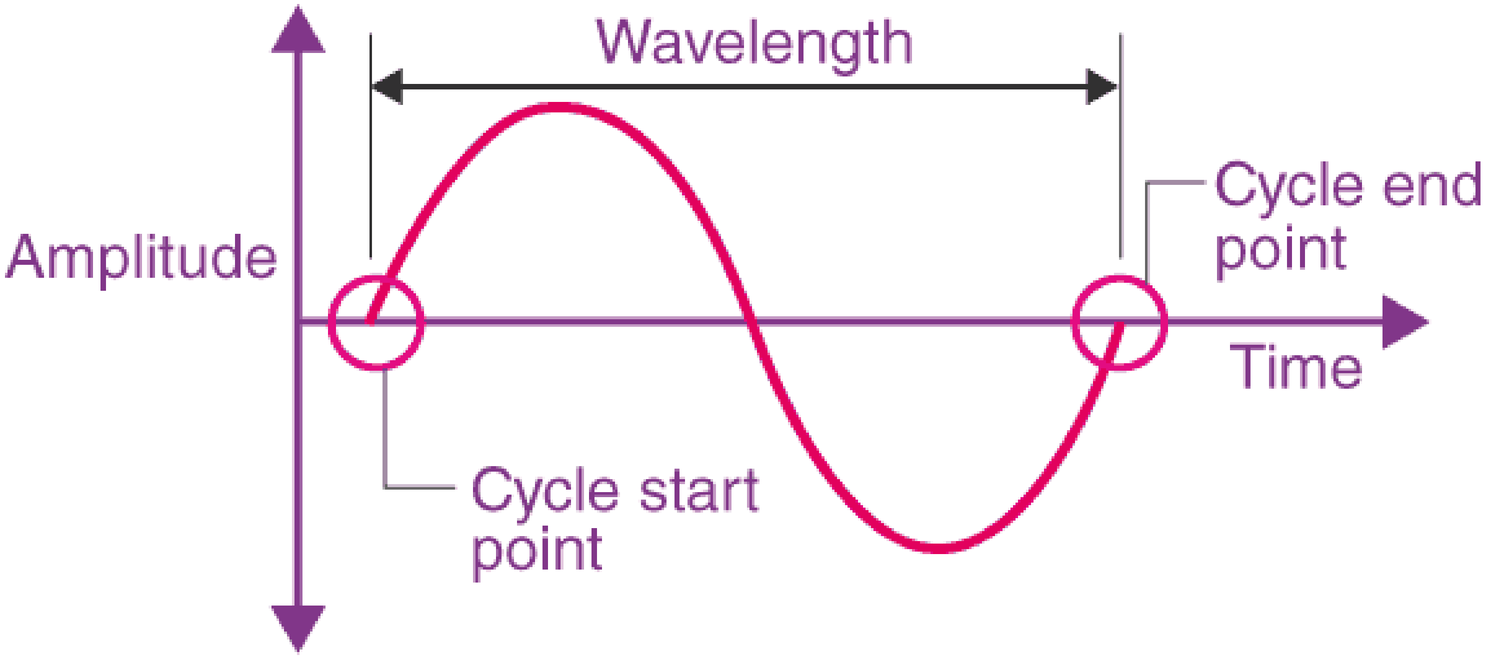


Image from Benjamin Salley. Munsell Student Color Set

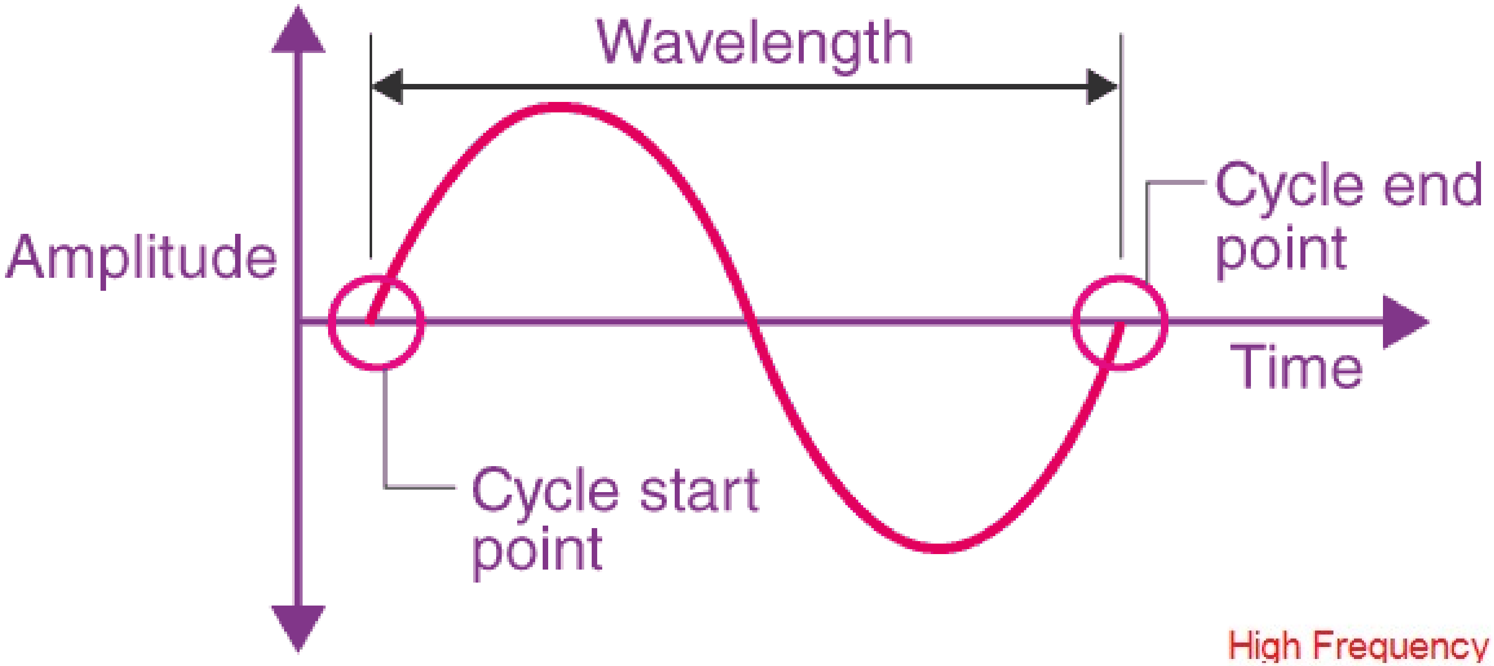
What is Color?

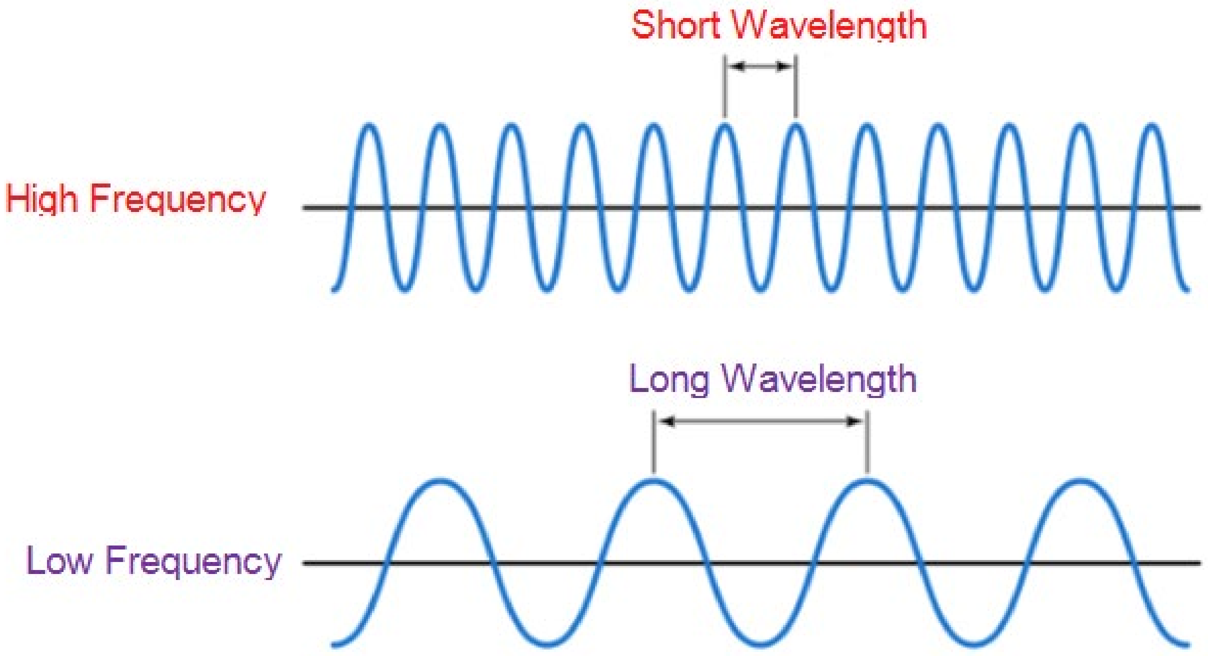
Light waves with different wavelengths
have different color

Recall: Basics of Waves



Recall: Basics of Waves



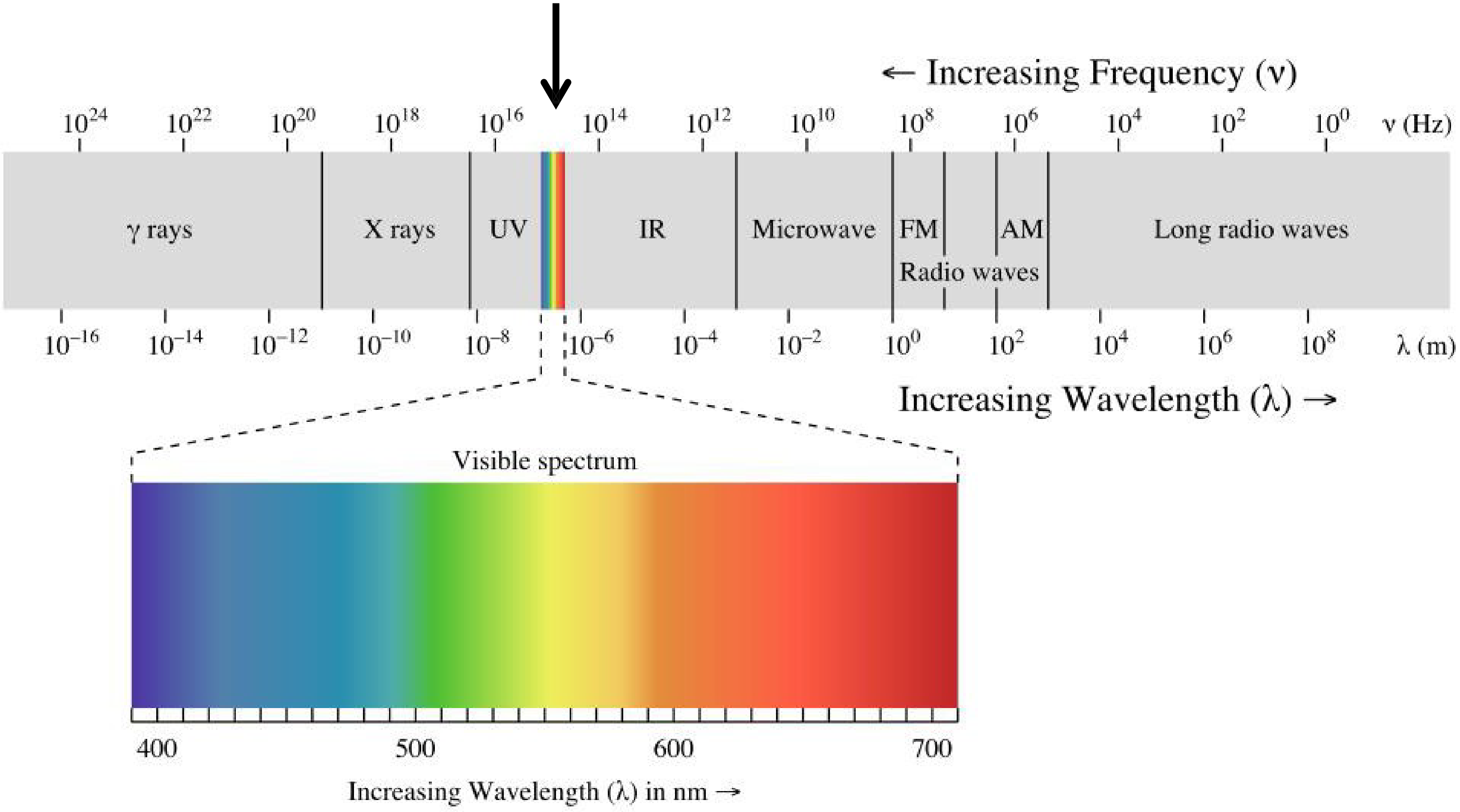


What is Color?

Light waves with different wavelengths
have different color

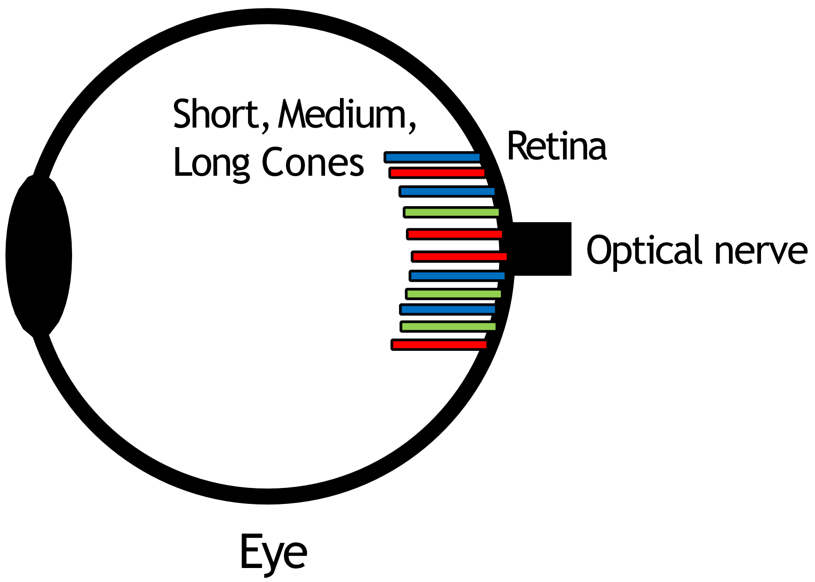
Generally, wavelengths from 380 to 720nm are visible to most humans

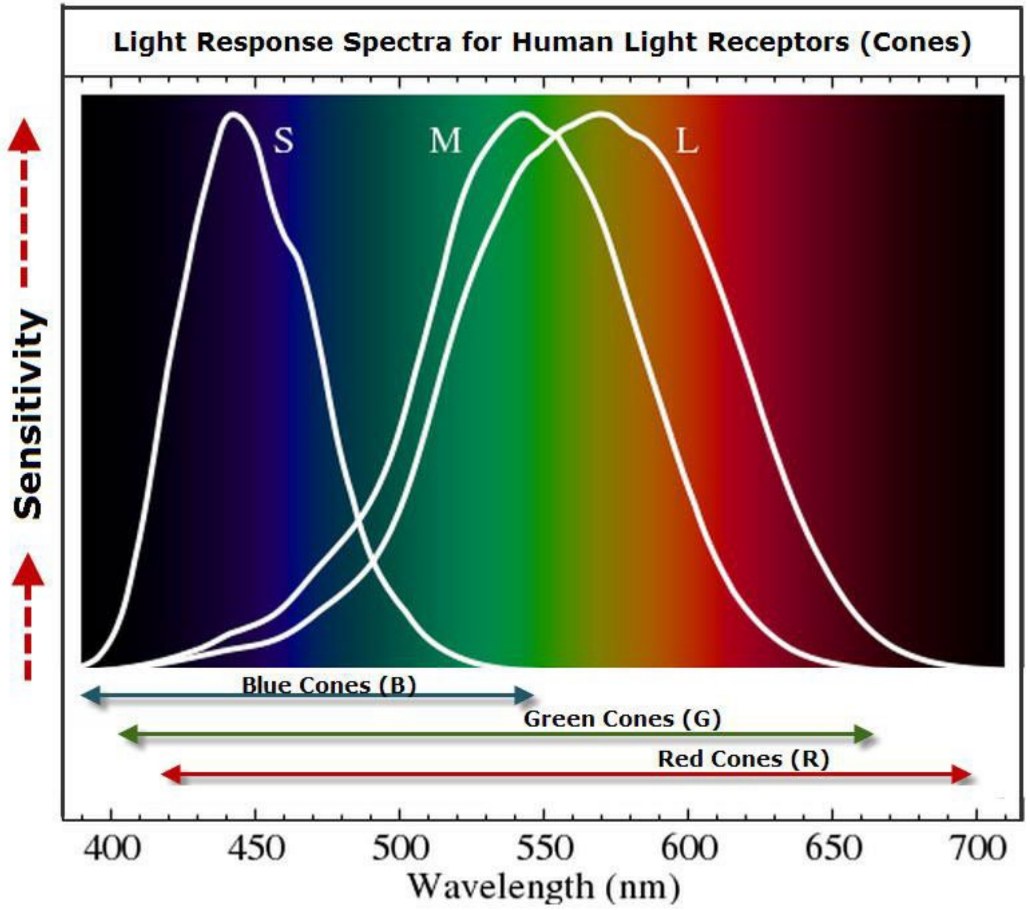
A *very* small range of electromagnetic radiation



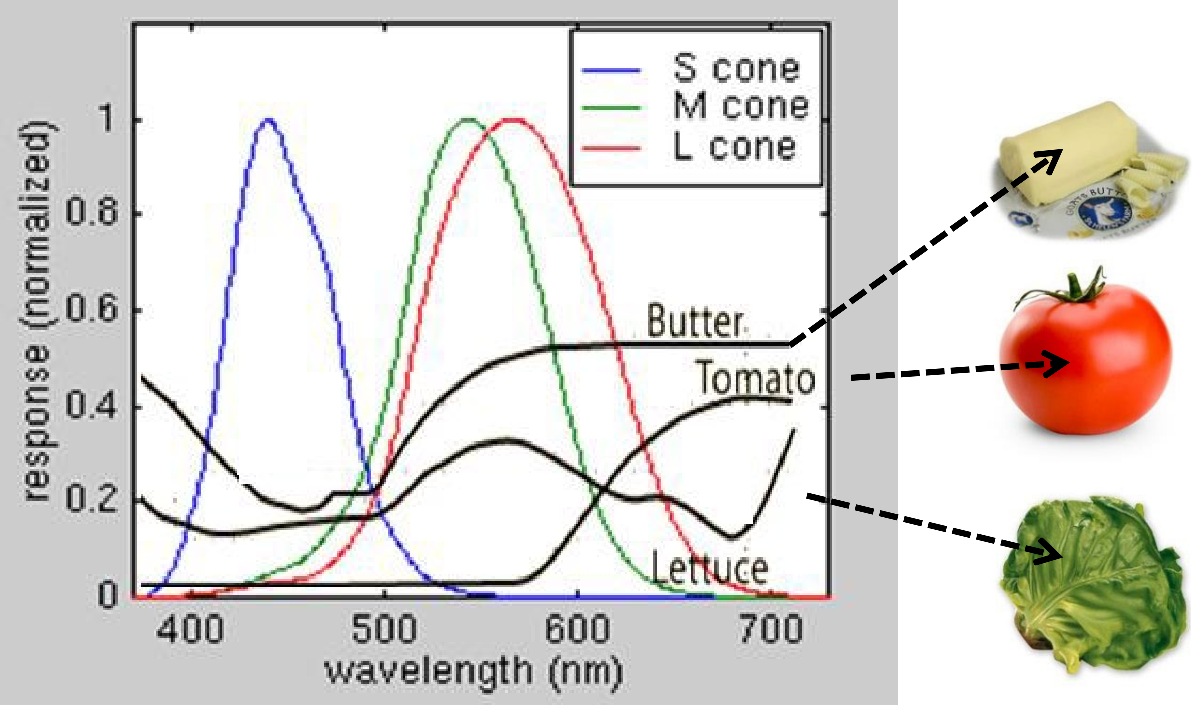
Biology of color sensations

- Our eye has three receptors (cone cells) that respond to visible light and give the sensation of color





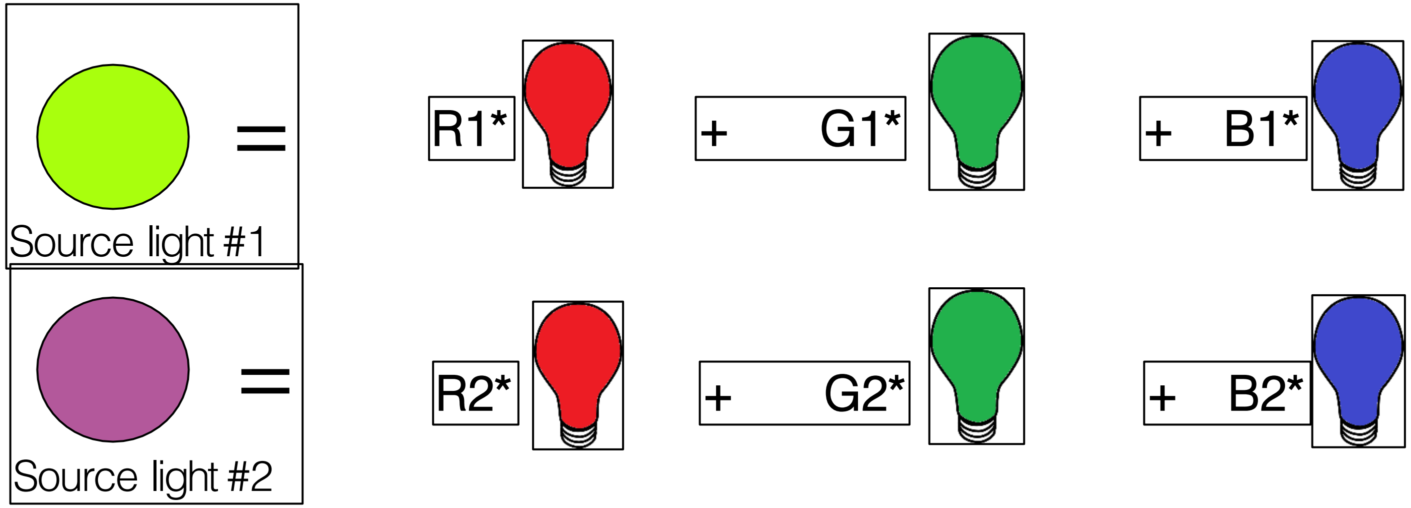
Spectral power distribution (SPD)



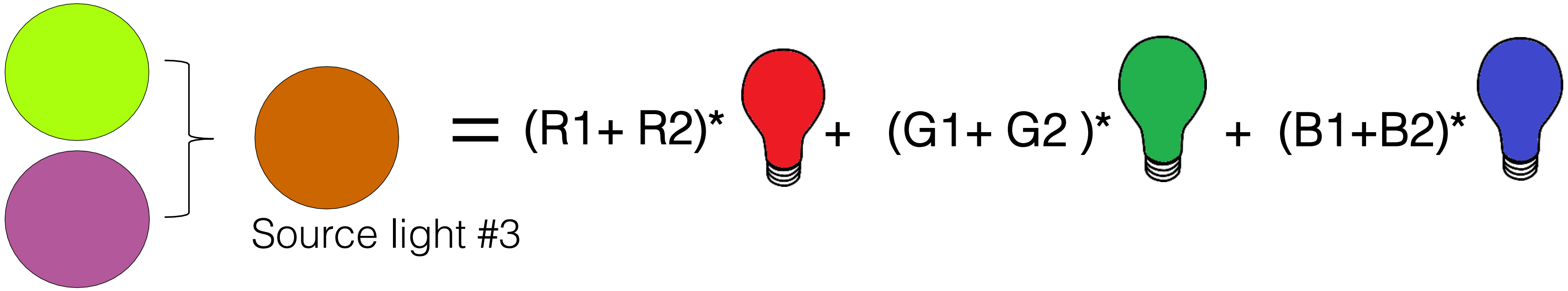
- We rarely see monochromatic light in real world scenes
- Instead, objects reflect a wide range of wavelengths.
- This can be described by a *spectral power distribution* (SPD)
- The SPD plot shows the relative amount of each wavelength reflected over the visible spectrum.

Tristimulus color theory (Grassman's Law)

Source color can be matched by **a linear combination** of three independent "primaries".

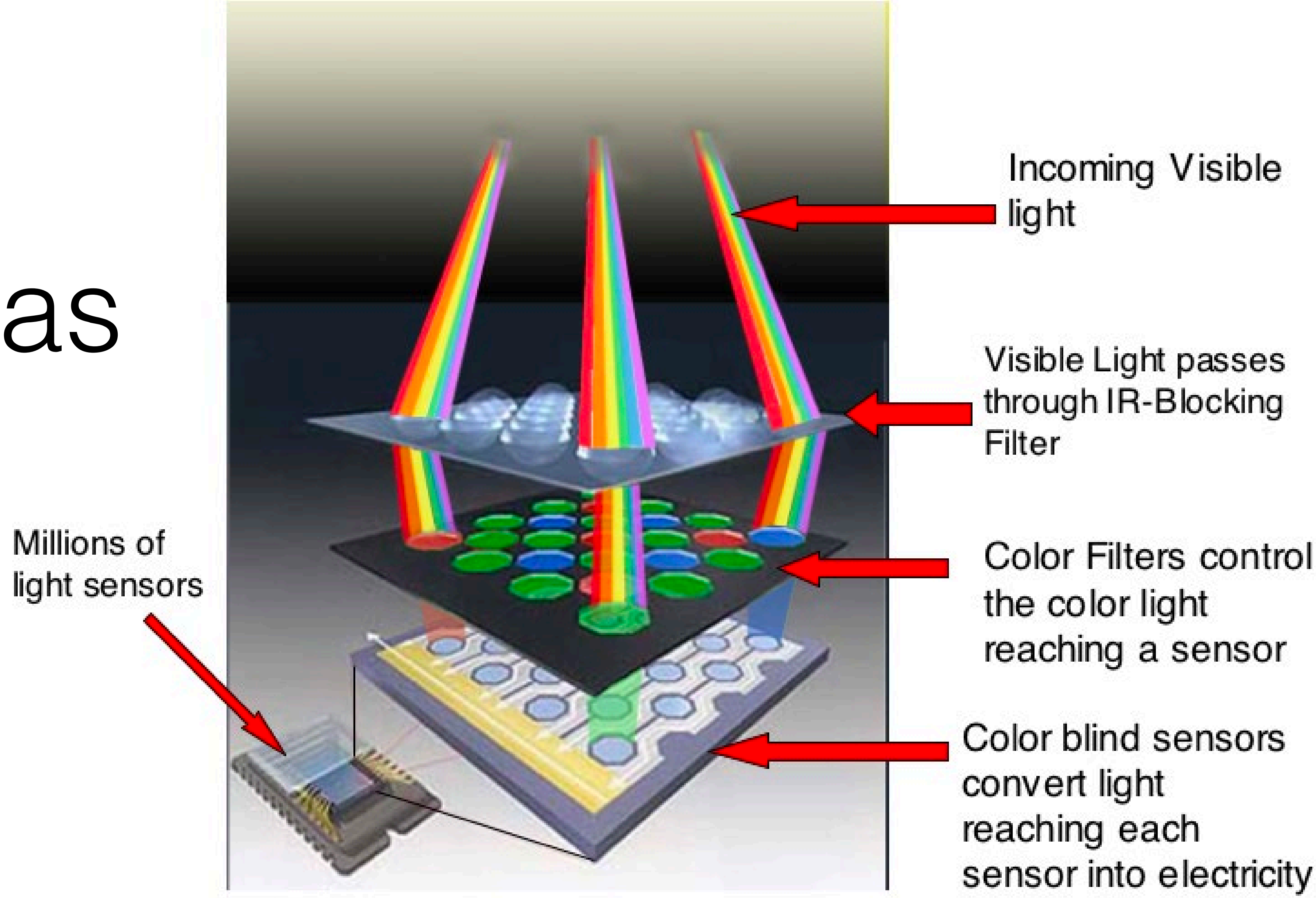


If we combined source lights 1 & 2 to get a new source light 3 → The amount of each primary needed to match the new light #3 is the sum of the weights that matched lights sources #1 & #2.

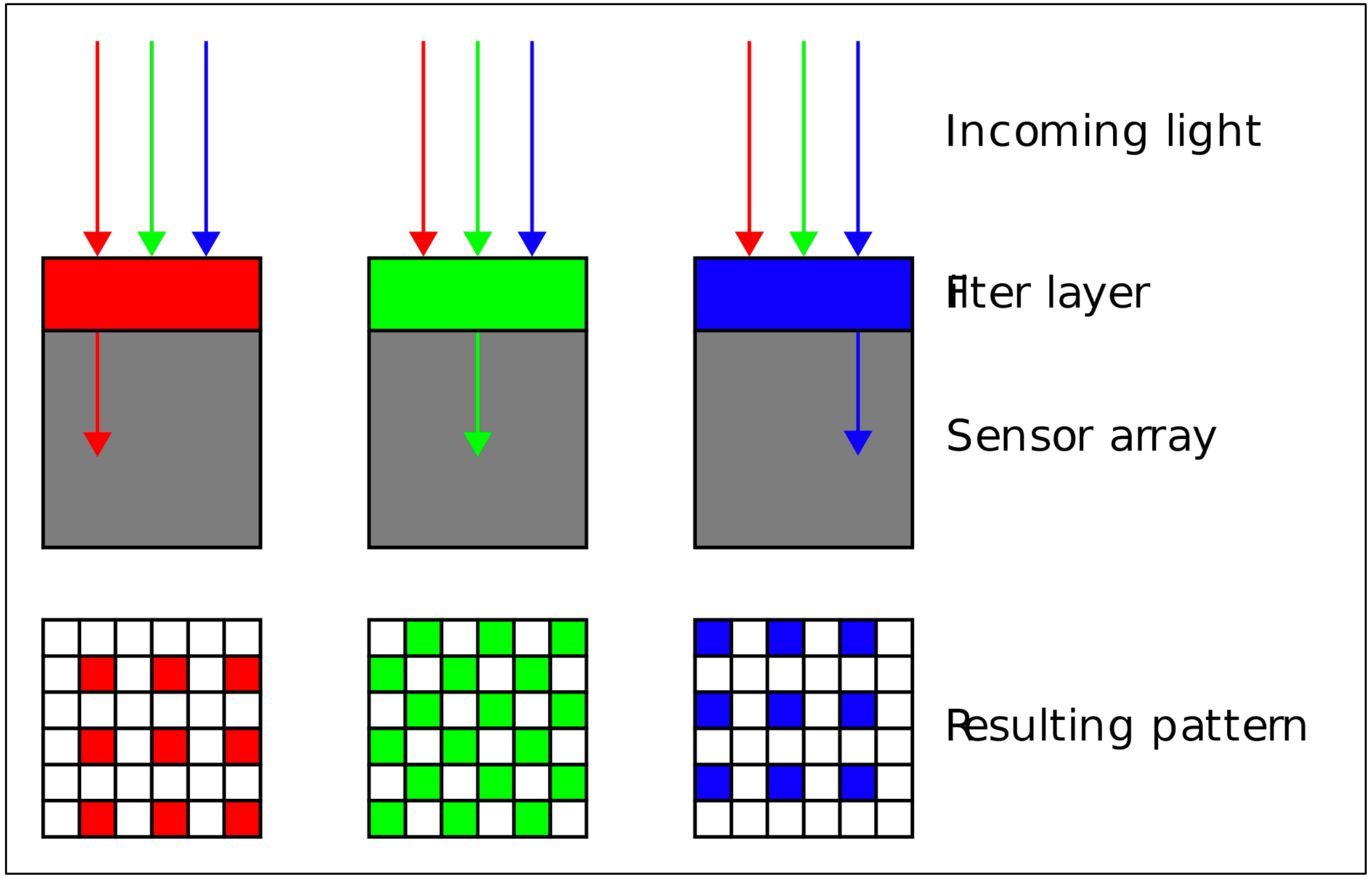


This may seem obvious now, but discovering that "light obeys the laws of linear algebra" was a huge achievement.

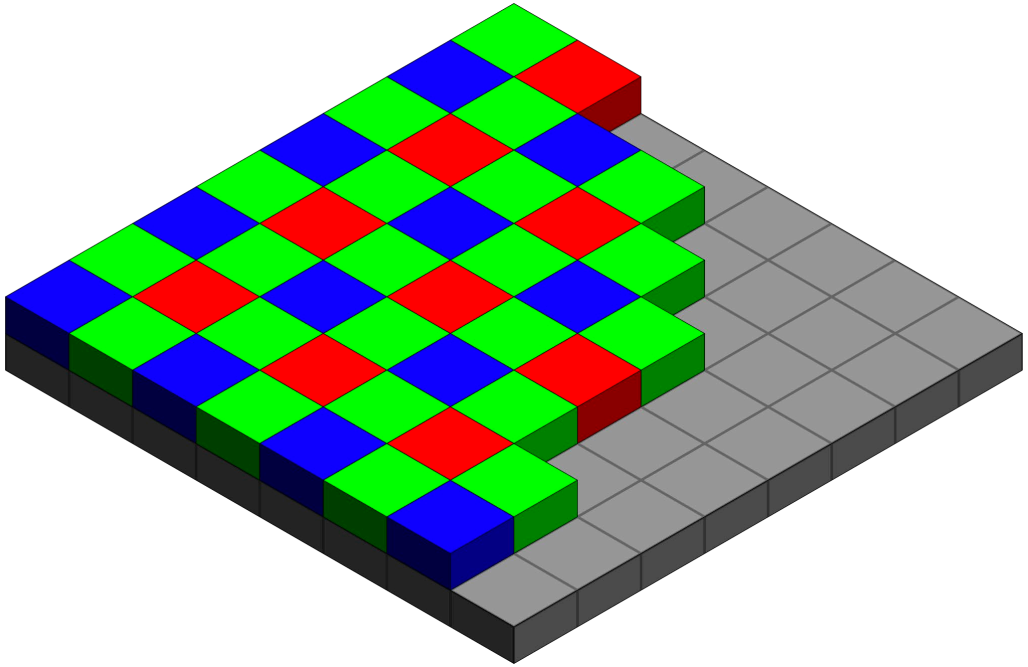
RGB in Cameras



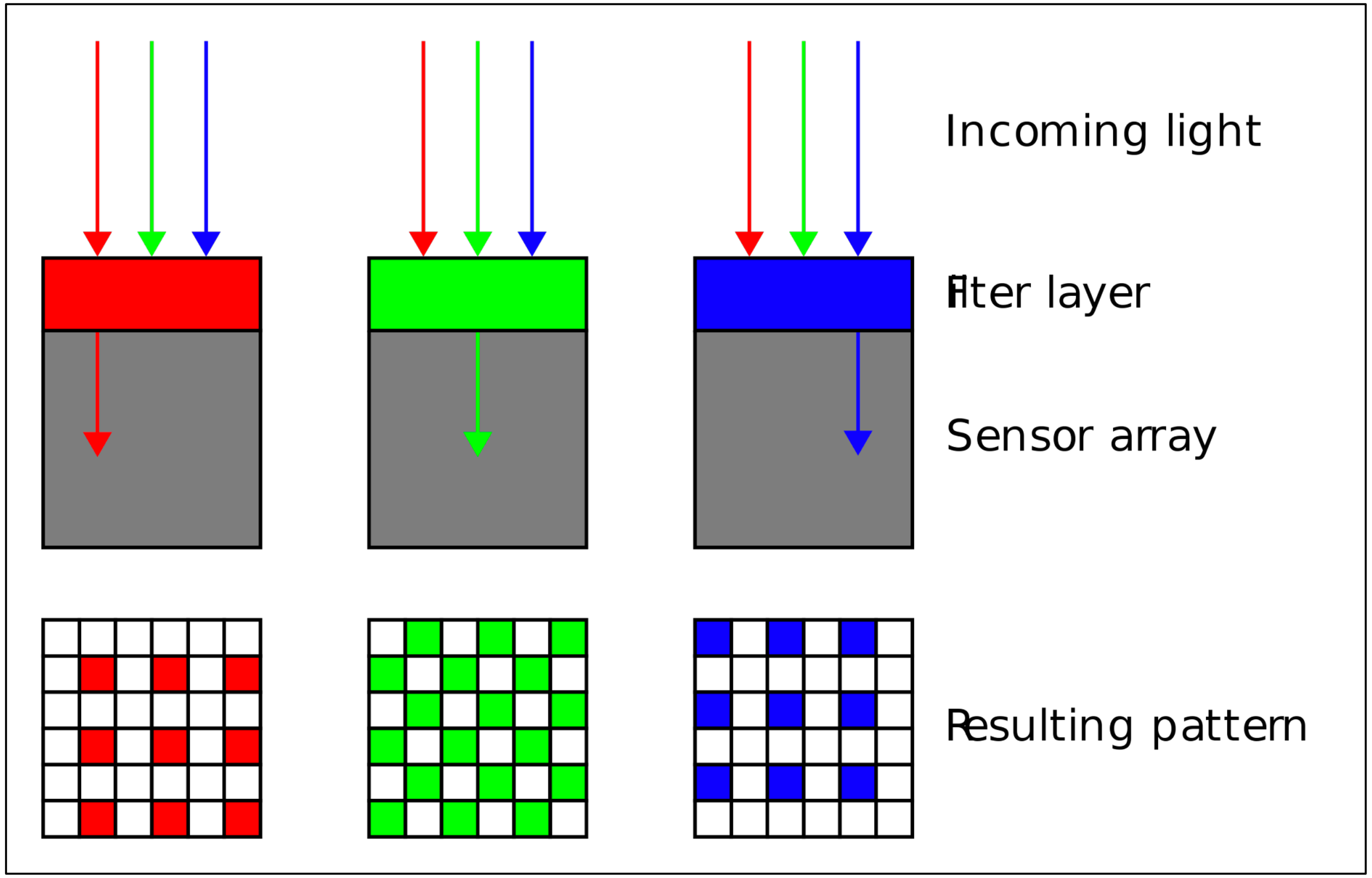
RGB in Cameras – Bayer Pattern



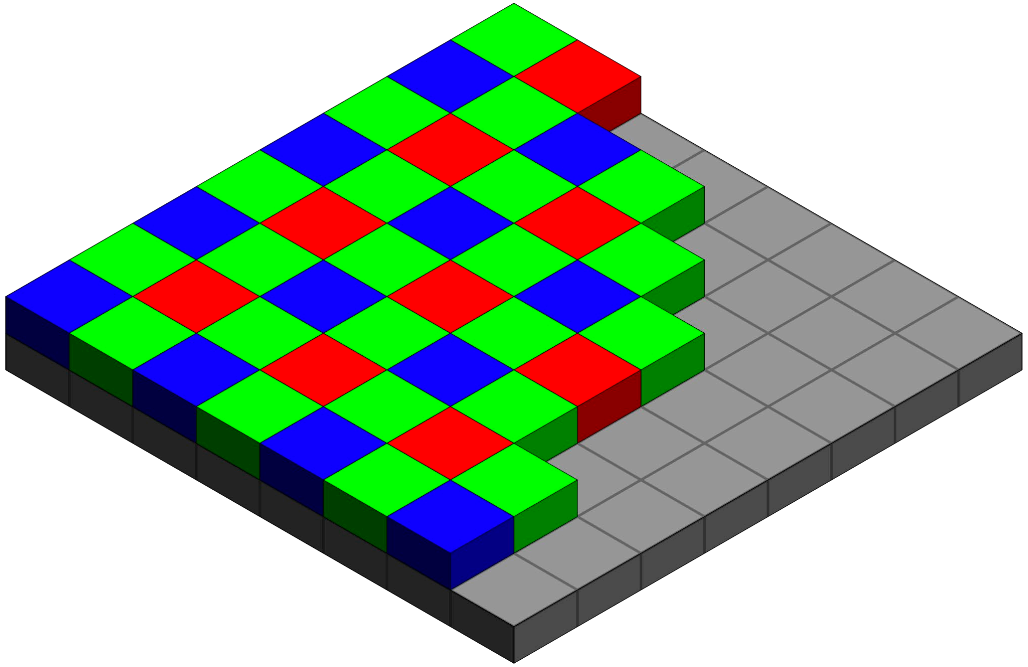
25% pixels see Red
25% pixels see Blue
50% pixels see Green



RGB in Cameras – Bayer Pattern



Then how do we get *all colors* at *all pixels?*



Original (High Resolution)

Bayer (120x80) Intensities

Bayer Color-Coded

After Interpolation

RGB in Cameras – Debayering / Demosaicing

How? → Interpolation !

Method 1: nearest-neighbor interpolation

- For each pixel, for the missing channel, assign the value of the closest pixel with that channel available

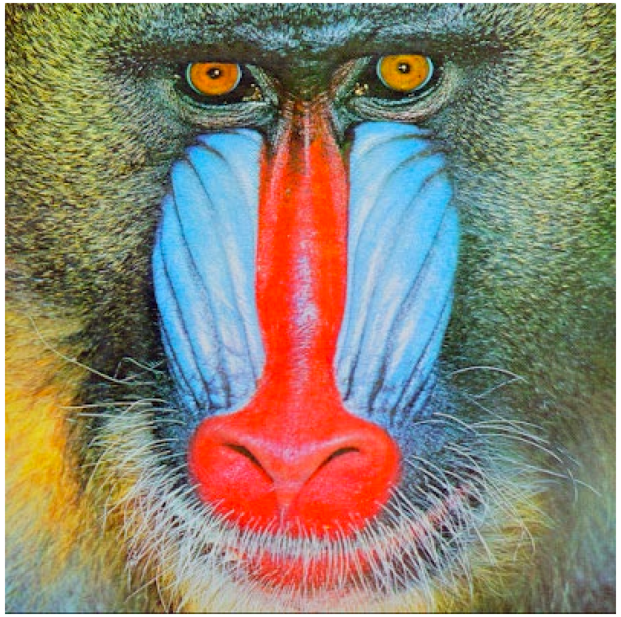
Method 2: Bi-Linear Interpolation

- Red-value of a non-red pixel
  = avg of 2 or 4 adjacent reds
- Similar for green and blue

More Advanced Methods …

Finally !
Digital RGB images!

What we see



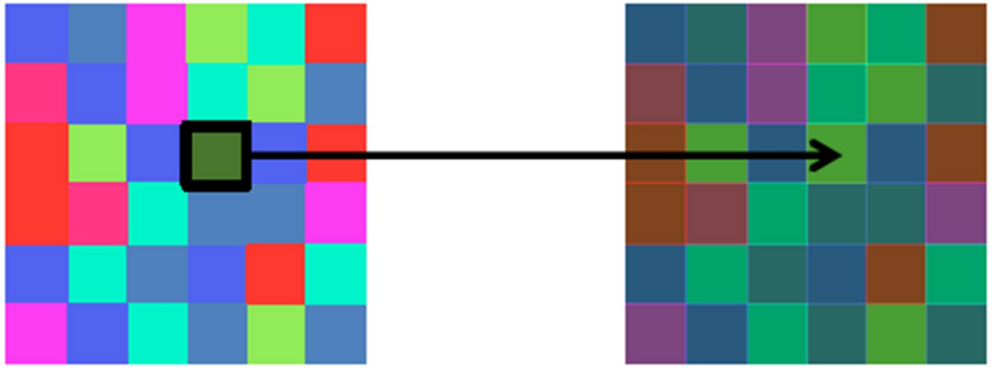
What the camera stores

Computer Vision

"understanding" the visual world by processing (RGB) images

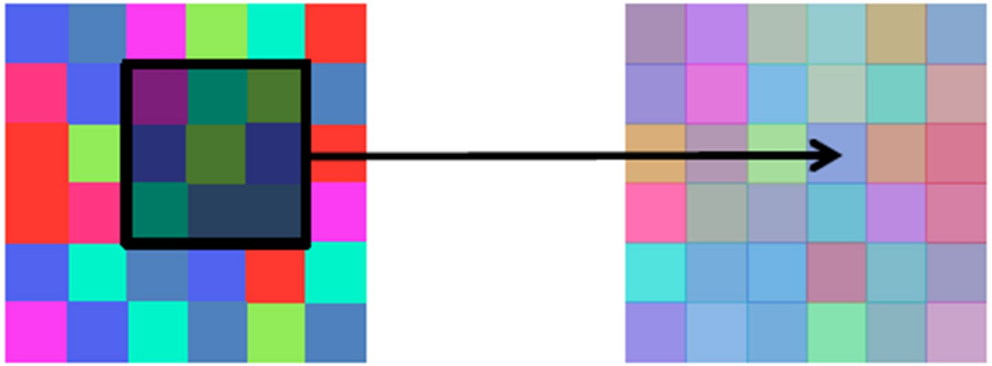
Point Processing vs Image Filtering

Point Operation



point processing

Neighborhood Operation



“filtering”

How would you implement these?

Examples of point processing

original

darken

lower contrast

non-linear lower contrast

How would you implement these?

Examples of point processing

original

$$x$$

darken

$$x - 128$$

lower contrast

$$\frac{x}{2}$$

non-linear lower contrast

$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

How would you implement these?

Examples of point processing

original

$$x$$

darken

$$x - 128$$

lower contrast

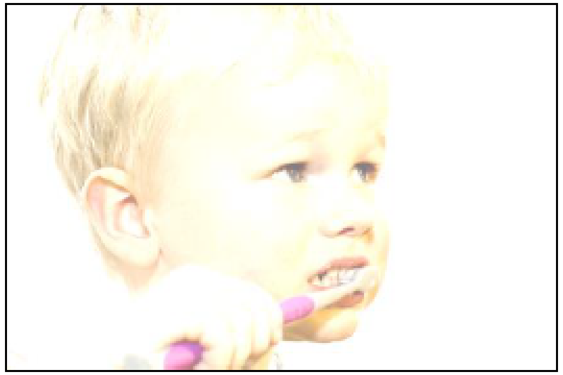
$$\frac{x}{2}$$

non-linear lower contrast

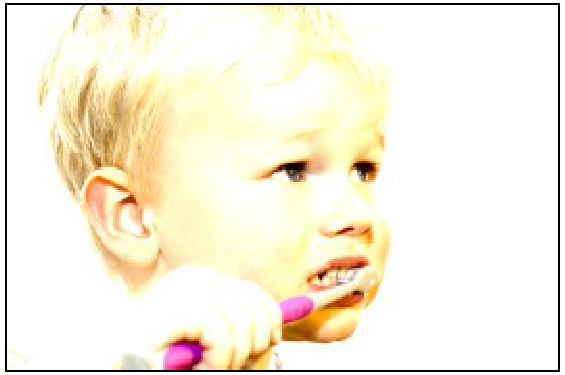
$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert

lighten



raise contrast



non-linear raise contrast

How would you implement these?

Examples of point processing

original

$$x$$

darken

$$x - 128$$

lower contrast

$$\frac{x}{2}$$

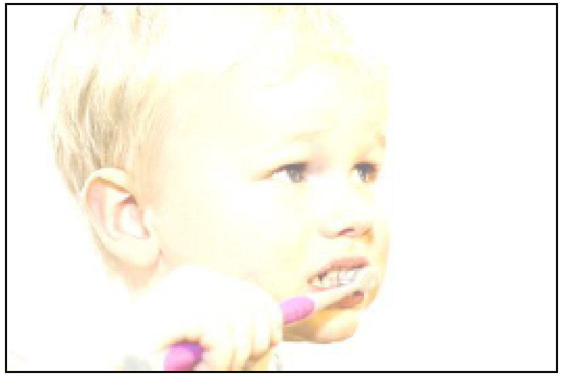
non-linear lower contrast

$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert

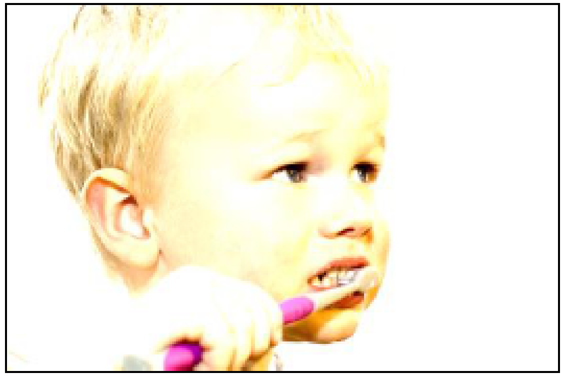
$$255 - x$$

lighten



$$x + 128$$

raise contrast



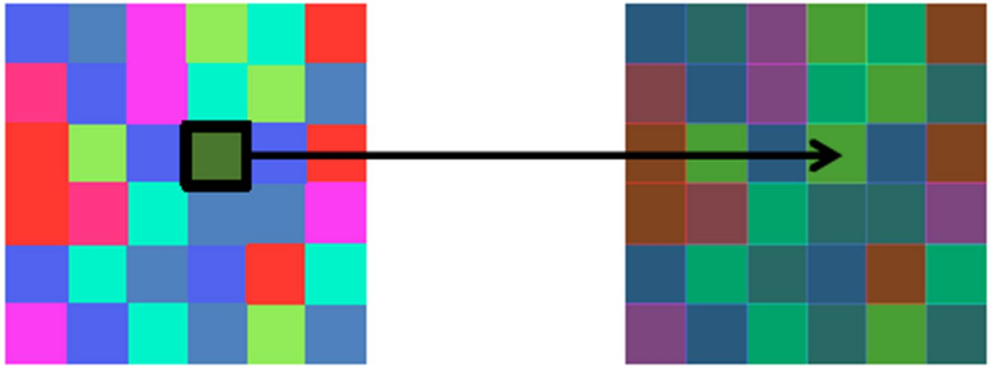
$$x \times 2$$

non-linear raise contrast

$$\left(\frac{x}{255}\right)^{2} \times 255$$

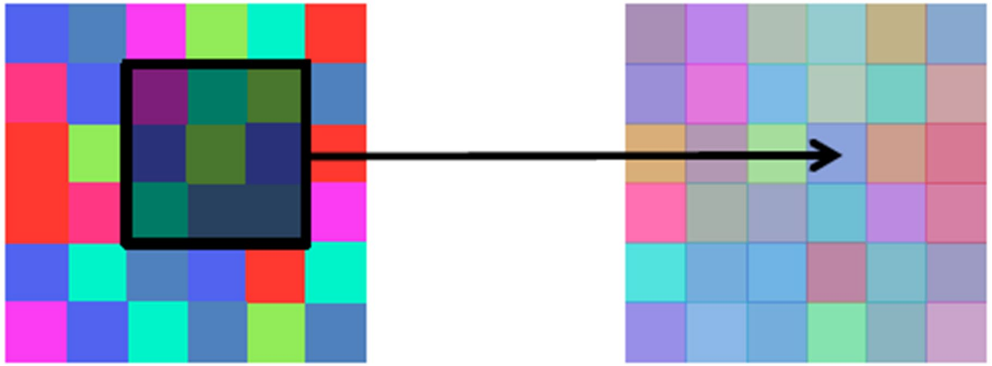
Point Processing vs Image Filtering

Point Operation



point processing

Neighborhood Operation



“filtering”

Next class: Filtering and Convolution