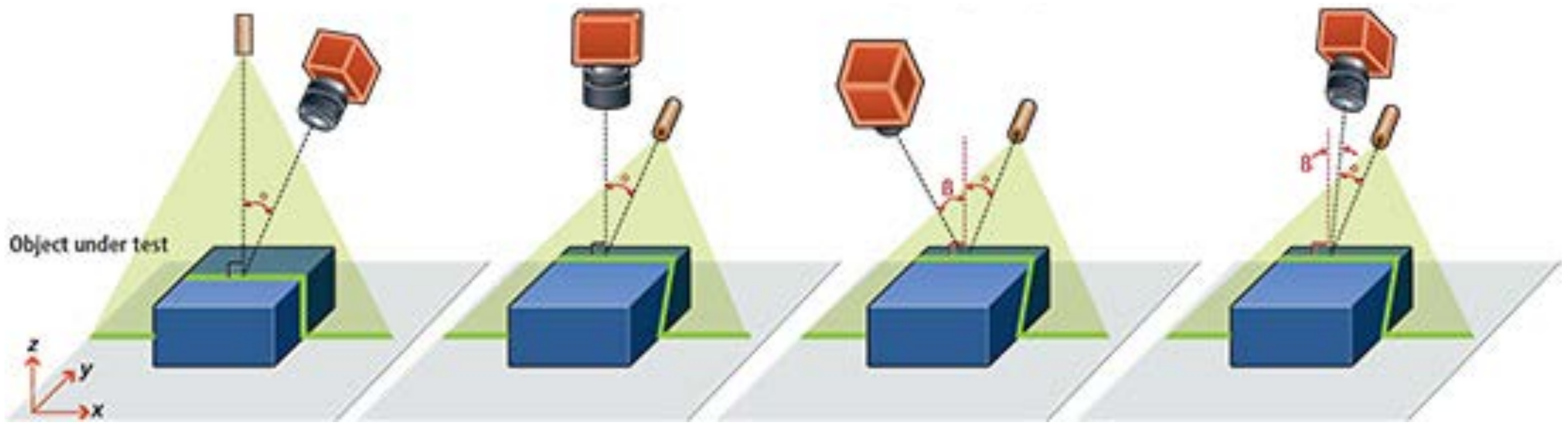
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



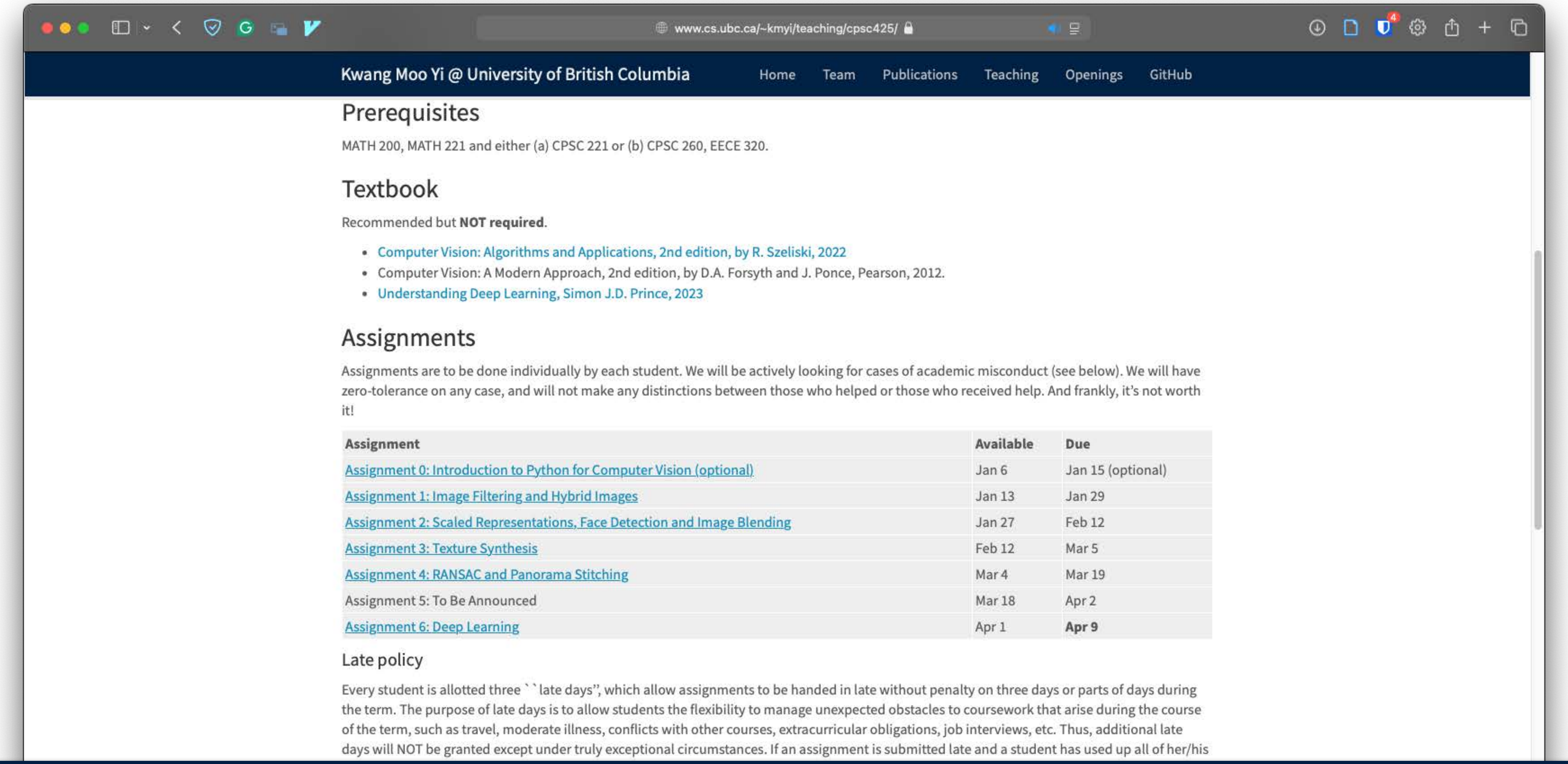
Lecture 2: Image Formation

(slide credits / thanks to **Bob Woodham, Jim Little, Fred Tung, Leonid Sigal, and Matthew Brown**)

Waitlisted Students

All **course materials** will be on the **public webpage**, and there will be

\+ **no graded assessment** before **add drop deadline** (Jan 17th)



www.cs.ubc.ca/~kmyi/teaching/cpsc425/

Kwang Moo Yi @ University of British Columbia | Home | Team | Publications | Teaching | Openings | GitHub

Prerequisites

MATH 200, MATH 221 and either (a) CPSC 221 or (b) CPSC 260, EECE 320.

Textbook

Recommended but **NOT required**.

- Computer Vision: Algorithms and Applications, 2nd edition, by R. Szeliski, 2022
- Computer Vision: A Modern Approach, 2nd edition, by D.A. Forsyth and J. Ponce, Pearson, 2012.
- Understanding Deep Learning, Simon J.D. Prince, 2023

Assignments

Assignments are to be done individually by each student. We will be actively looking for cases of academic misconduct (see below). We will have zero-tolerance on any case, and will not make any distinctions between those who helped or those who received help. And frankly, it's not worth it!

Assignment	Available	Due
Assignment 0: Introduction to Python for Computer Vision (optional)	Jan 6	Jan 15 (optional)
Assignment 1: Image Filtering and Hybrid Images	Jan 13	Jan 29
Assignment 2: Scaled Representations, Face Detection and Image Blending	Jan 27	Feb 12
Assignment 3: Texture Synthesis	Feb 12	Mar 5
Assignment 4: RANSAC and Panorama Stitching	Mar 4	Mar 19
Assignment 5: To Be Announced	Mar 18	Apr 2
Assignment 6: Deep Learning	Apr 1	**Apr 9**

Late policy

Every student is allotted three ``late days'', which allow assignments to be handed in late without penalty on three days or parts of days during the term. The purpose of late days is to allow students the flexibility to manage unexpected obstacles to coursework that arise during the course of the term, such as travel, moderate illness, conflicts with other courses, extracurricular obligations, job interviews, etc. Thus, additional late days will NOT be granted except under truly exceptional circumstances. If an assignment is submitted late and a student has used up all of her/his

- Waitlist resolution will follow the standard department policy https://www.cs.ubc.ca/students/undergrad/courses/waitlists

https://www.cs.ubc.ca/~kmyi/teaching/cpsc425/

This Lecture

Topics: Image Formation

— Image Formation

— Cameras and Lenses

— Projection

Readings:

— **Today's** Lecture: Szeliski Chapter 2, Forsyth & Ponce (2nd ed.) 1.1.1 — 1.1.3

— **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.1, 4.5

Lecture 2: Goal

To understand how images are formed

(and develop relevant mathematical concepts and abstractions)

What is **Computer Vision**?

Compute vision, broadly speaking, is a research field aimed to enable computers to **process and interpret visual data**, as sighted humans can.

Sensing Device

Interpreting Device

Image Credit: https://www.flickr.com/photos/flamephoenix1991/8376271918

Image (or video)

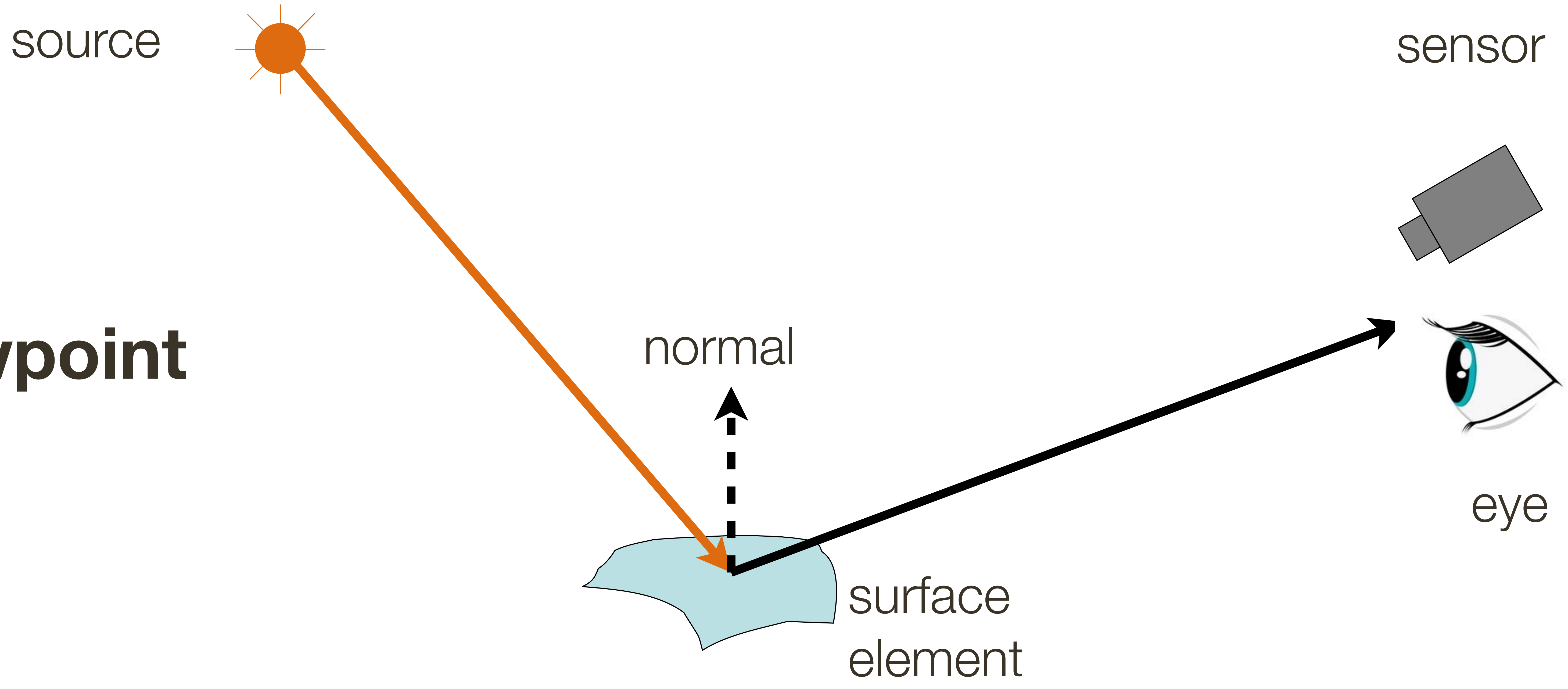
Interpretation

blue sky,
trees,
fountains,
UBC, …

Overview: Image Formation, Cameras and Lenses

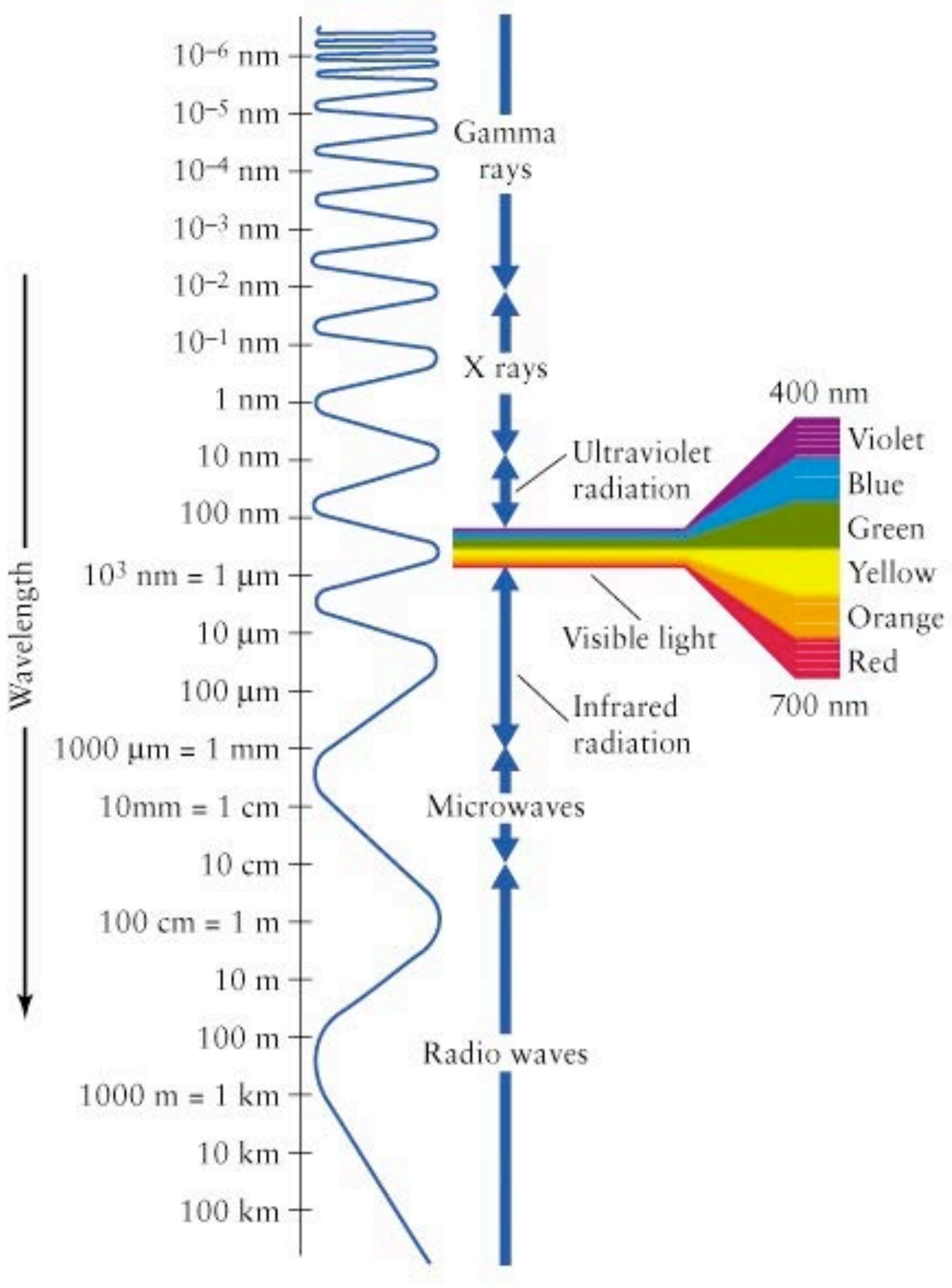
The **image formation process** that produces a particular image depends on

- **Lighting** condition
- Scene **geometry**
- **Surface** properties
- Camera **optics** and **viewpoint**



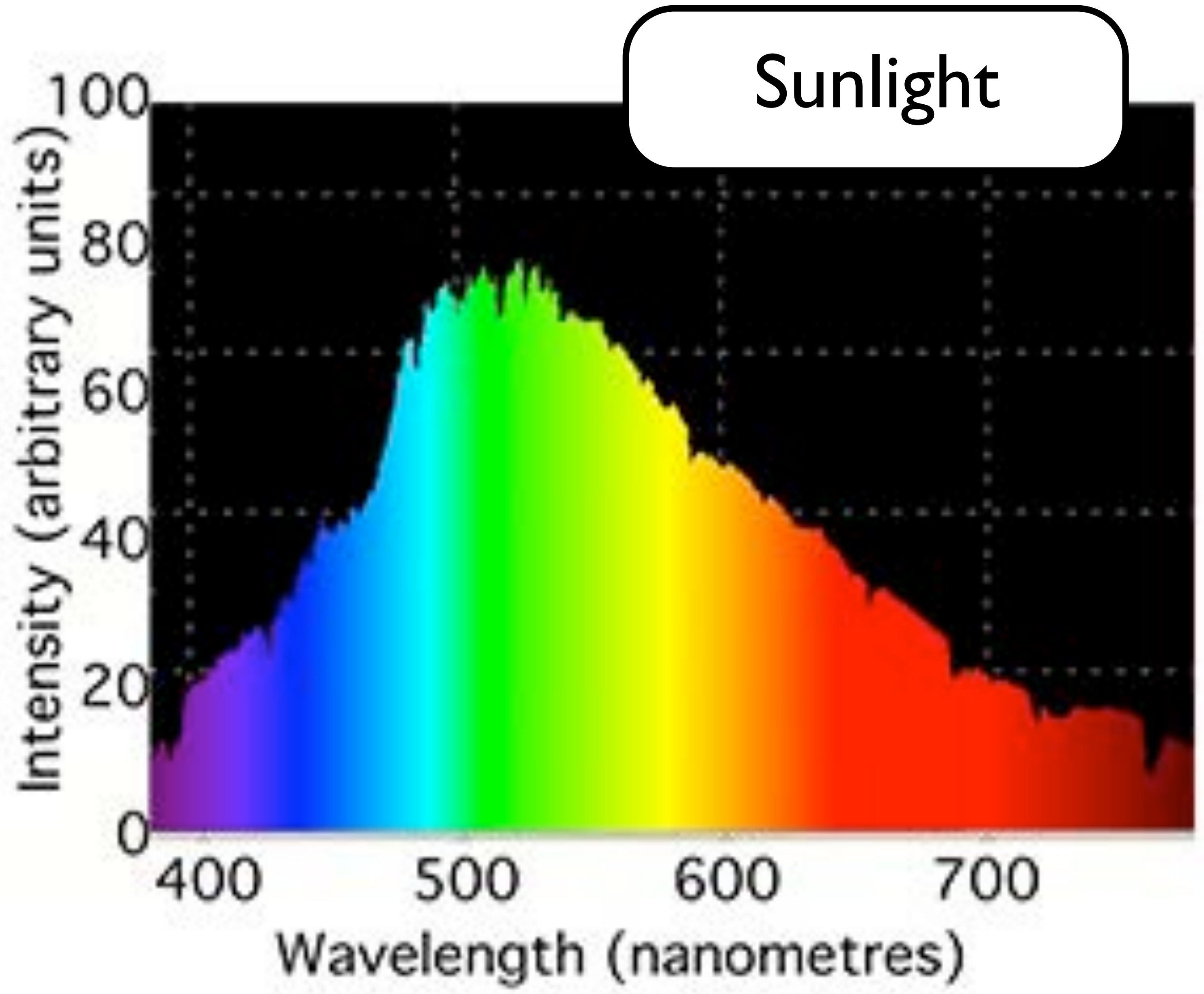
Sensor (or eye) **captures amount of light** reflected from the object

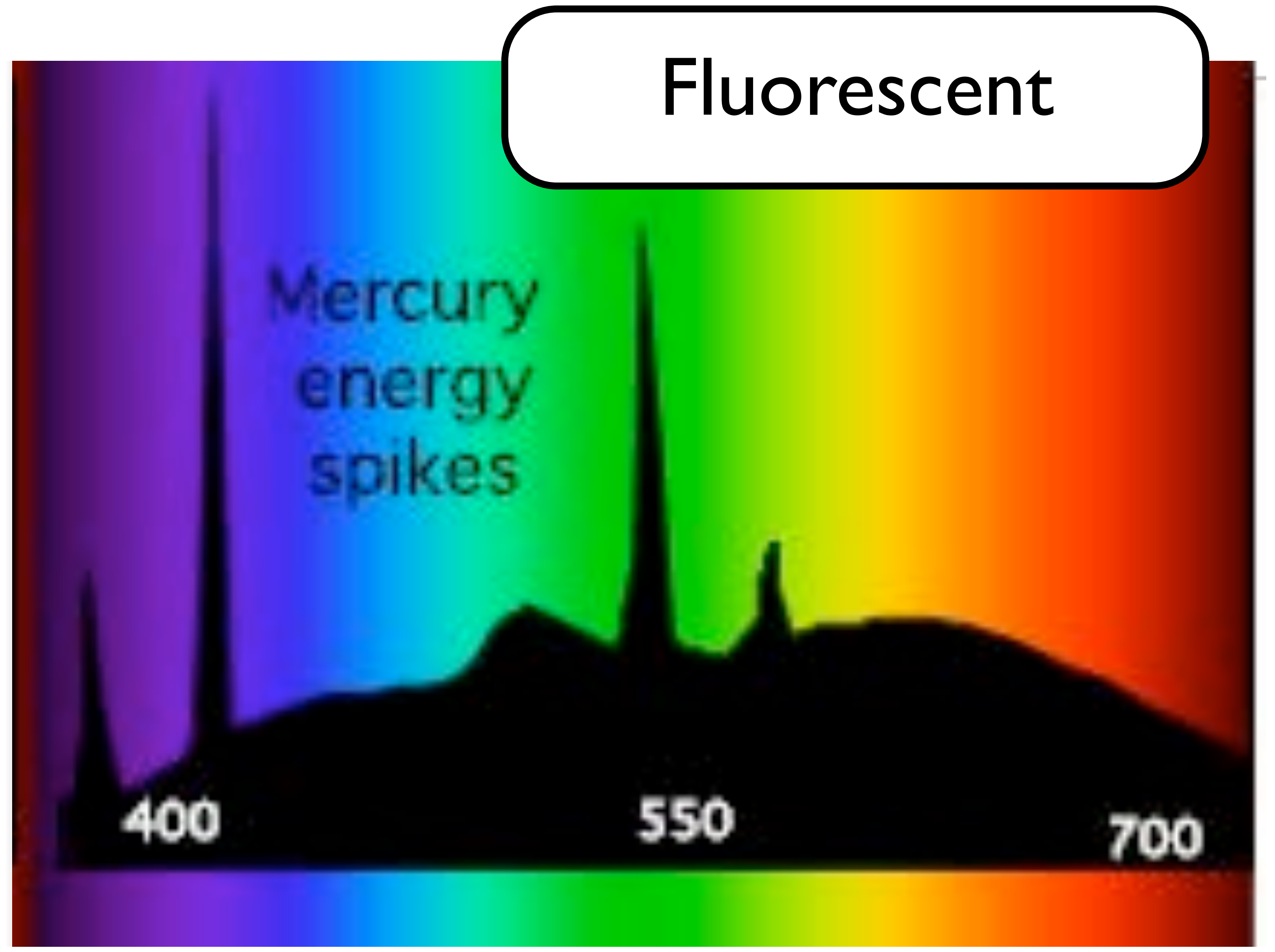
Light and Color



- Light is electromagnetic radiation in the 400-700nm band
- This is the peak in the spectrum of sunlight passing through the atmosphere
- Newton's Prism experiment showed that white light is composed of all frequencies
- Black is the absence of light!

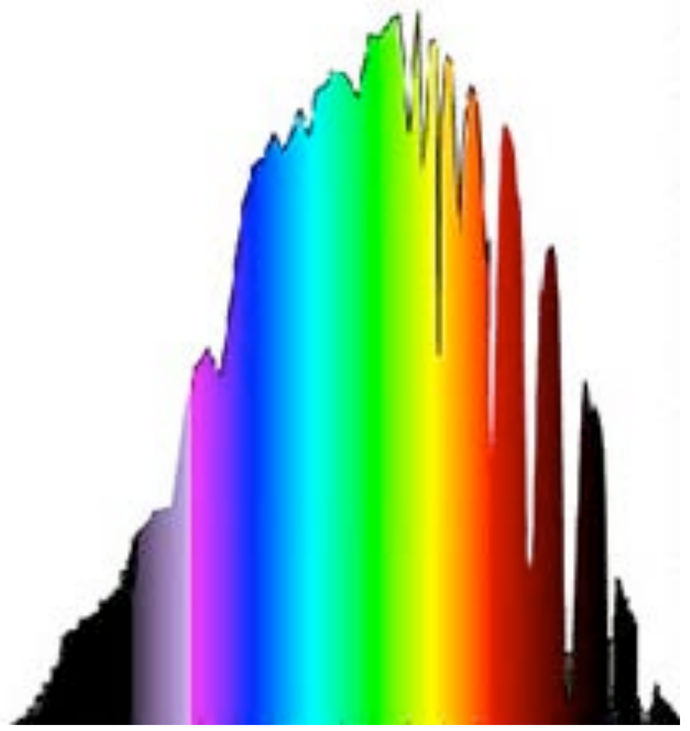
Spectral Power Distribution

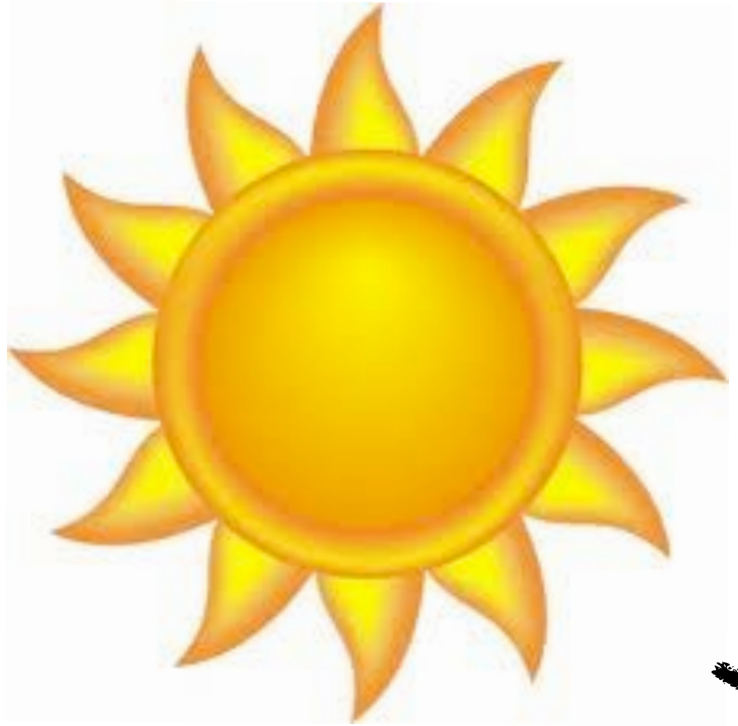




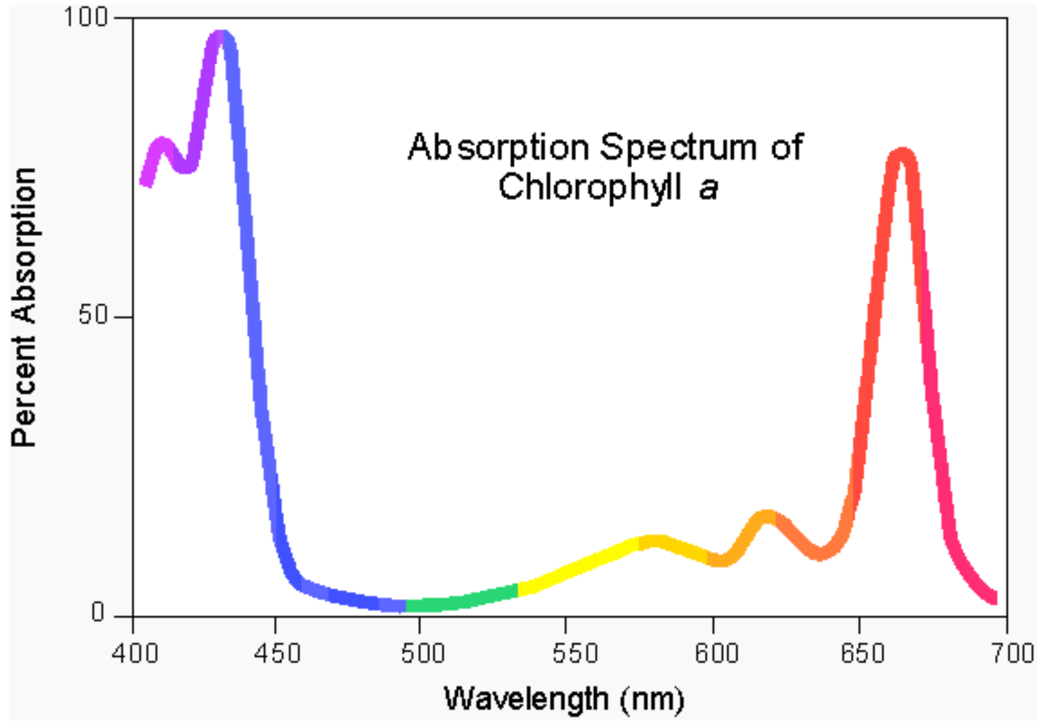
- The spectral distribution of energy in a light ray determines its colour
- Surfaces reflect light energy according to a spectral distribution as well
- The combination of incident spectra and reflectance spectra determines the light colour

Spectral Reflectance Example





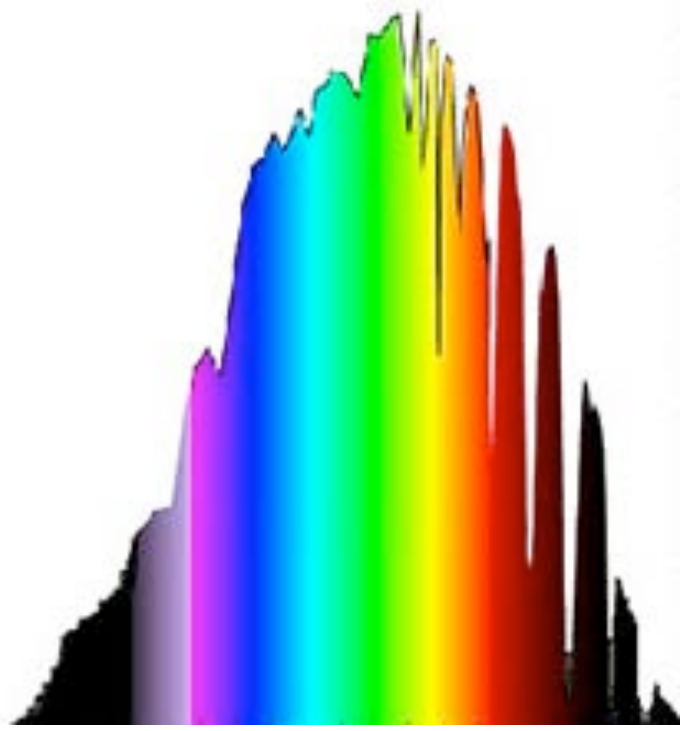
$E(\lambda)$

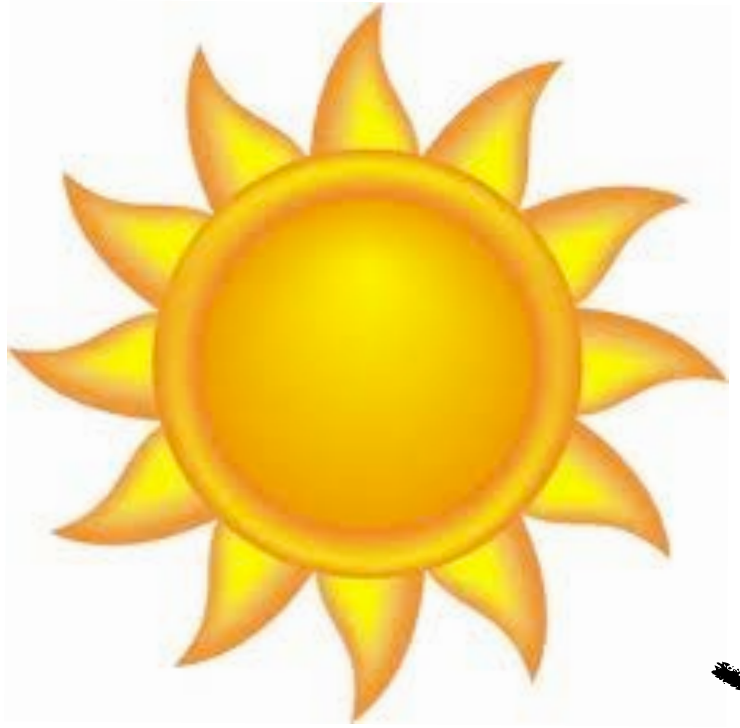


$S(\lambda)$

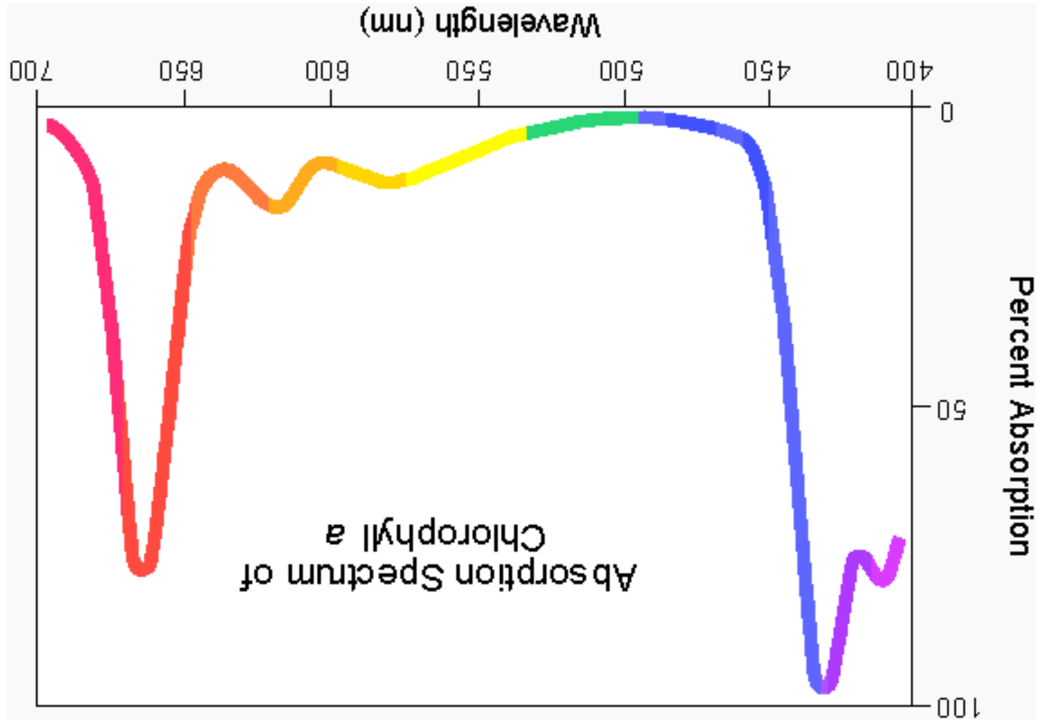
$E(\lambda)S(\lambda)$

Spectral Reflectance Example





$E(\lambda)$



$S(\lambda)$

$E(\lambda)S(\lambda)$

Our brains already knows this

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

Our brains already knows this

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

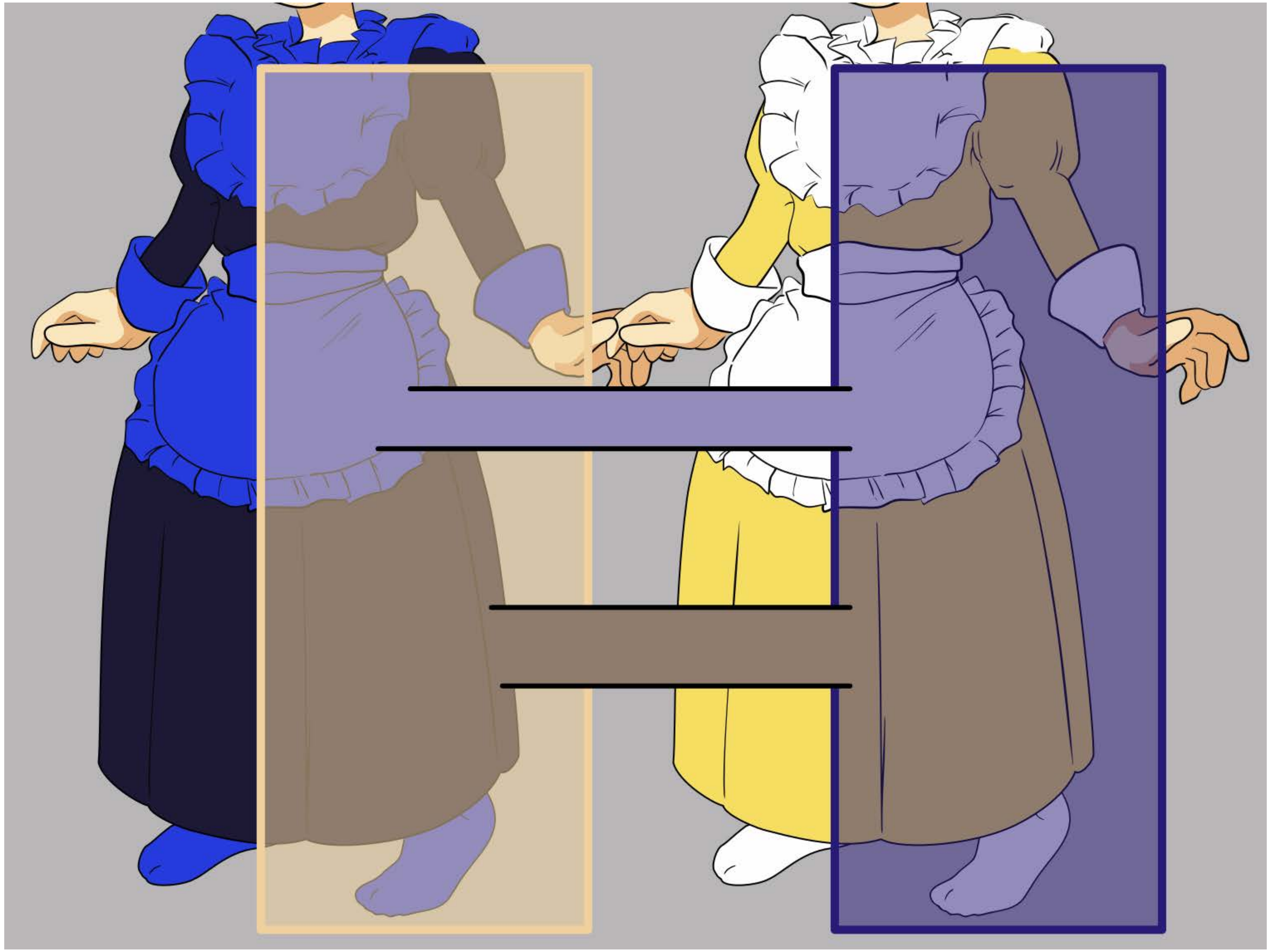


Figure design by Kasuga~jawiki; vectorization by Editor at Large; "The dress" modification by Jahobr, CC-BY-SA 2.5 Generic

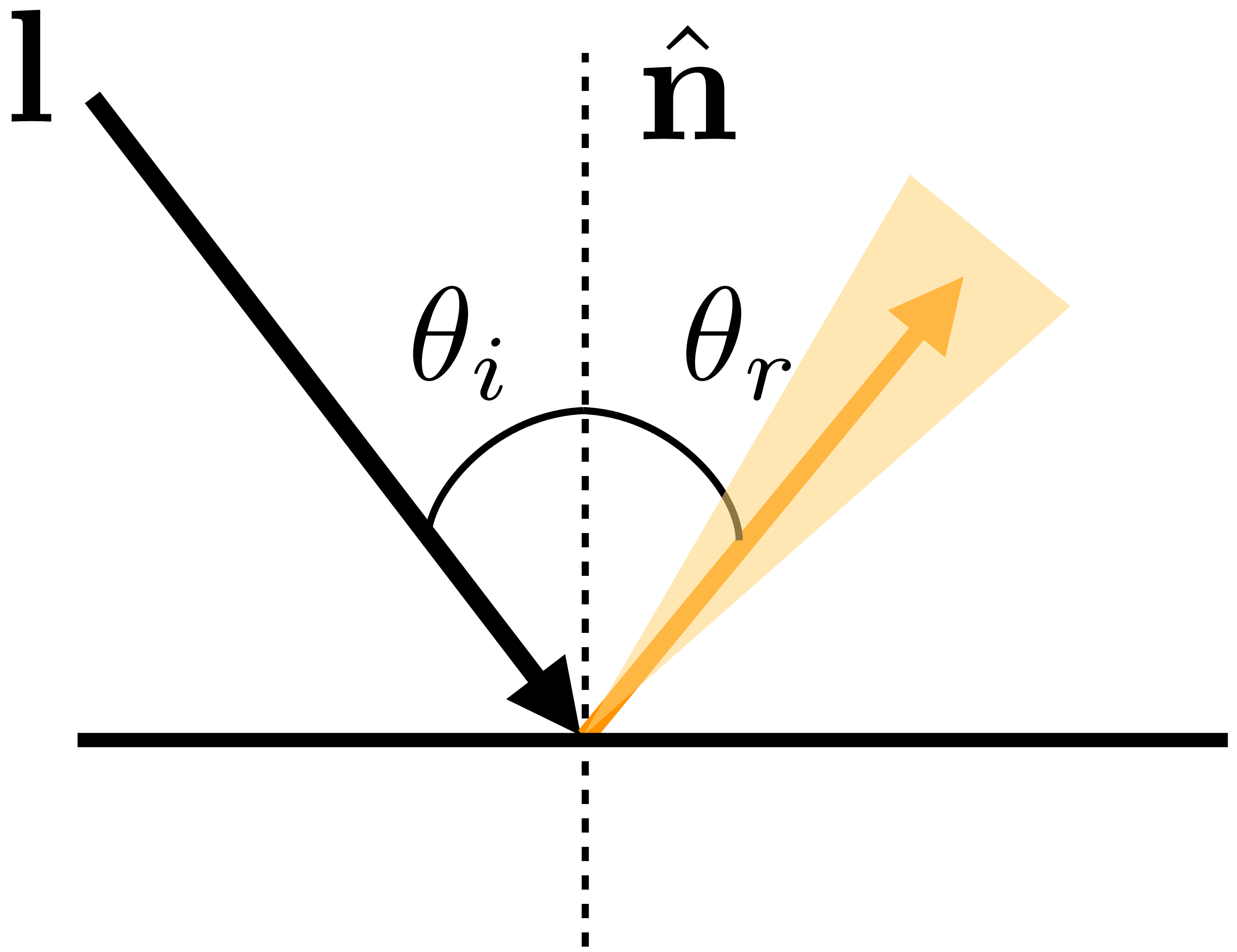
Surface Reflectance

- Reflected intensity also depends on geometry: surface orientation, viewer position, shadows, etc.

It also depends on surface properties, e.g., diffuse or specular

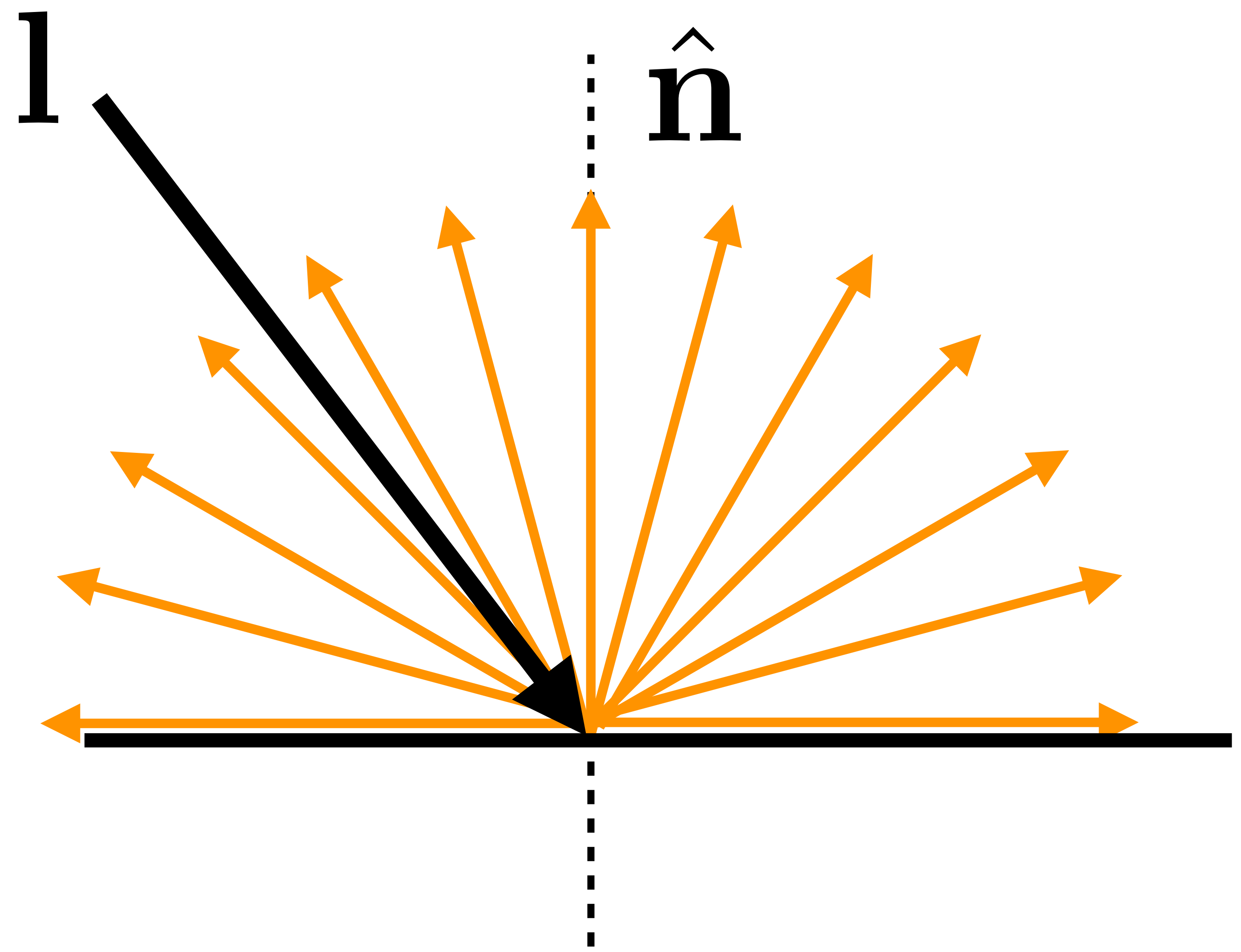
Diffuse and Specular Reflection

- A pure mirror reflects light along a line symmetrical about the surface normal
- A pure diffuse surface scatters light equally in all directions



Pure Mirror Reflection

$$\theta_i = \theta_r$$

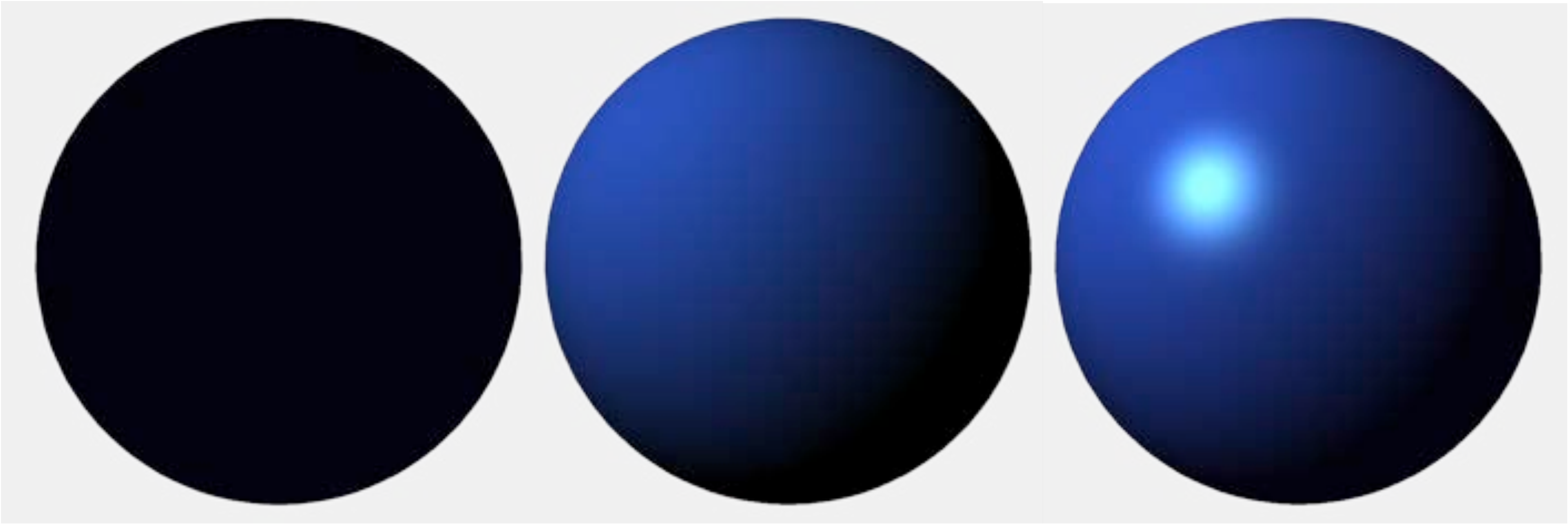


Lambertian Reflection
(Diffuse)

Specular surfaces directly reflect over a small angle

Diffuse and Specular Reflection

- A sphere lit with ambient, +diffuse, +specular reflectance



Ambient +Diffuse +Specular

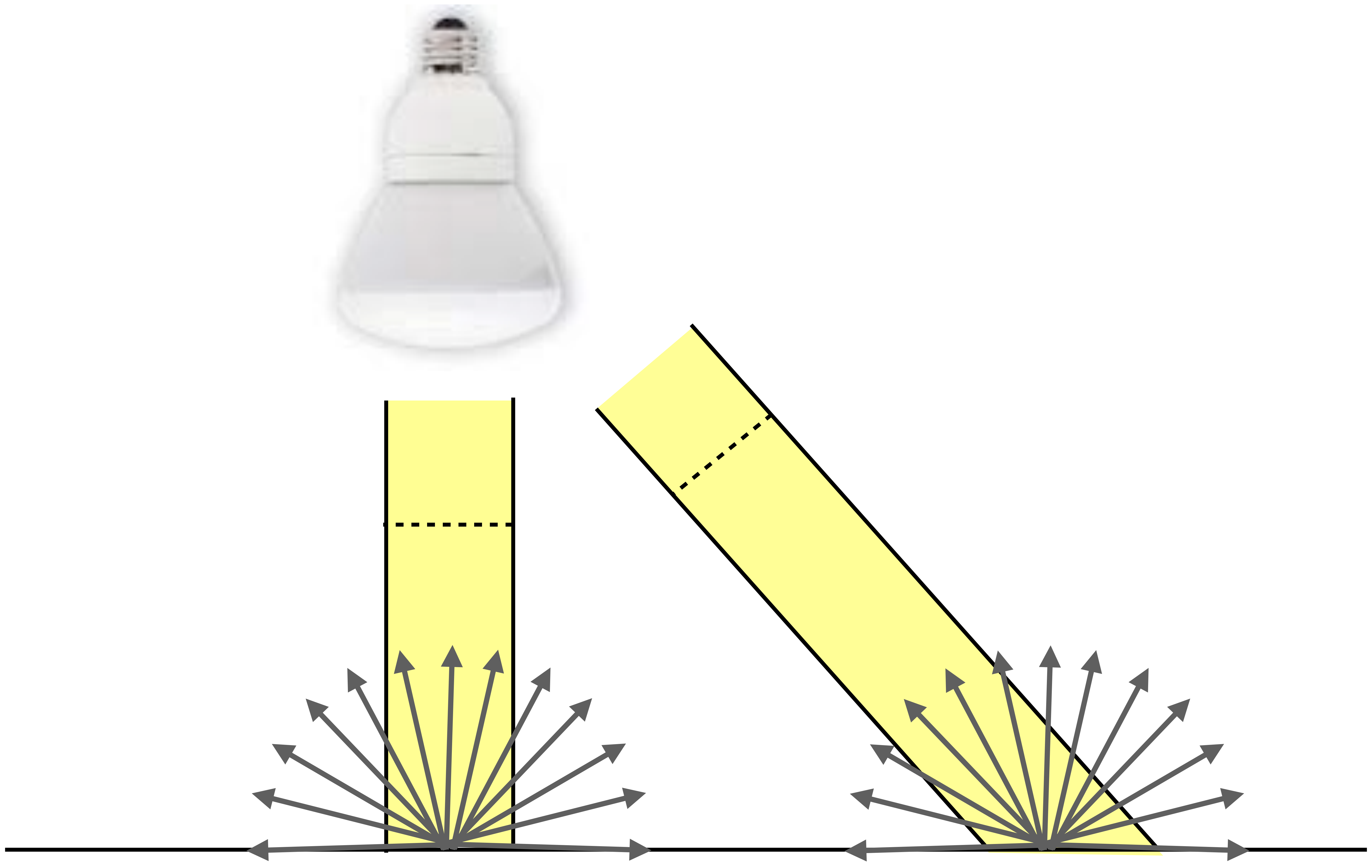
Diffuse and Specular Reflection

- A motivating example that uses this model

[Video from https://machinelearning.apple.com/research/neural-3d-relightable reproduced for educational purposes]

Diffuse Reflection

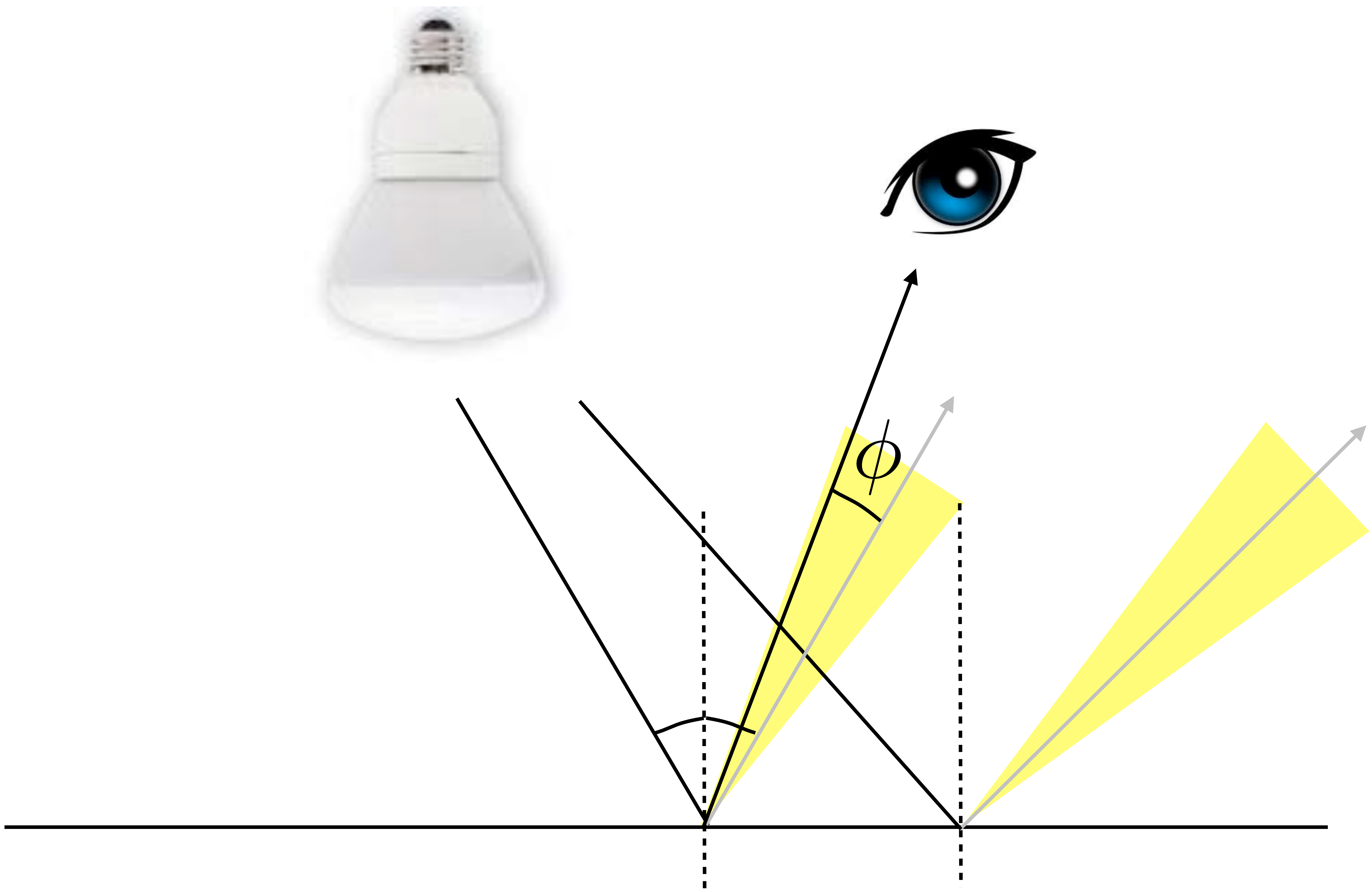
- Light is reflected equally in all directions (Lambertian surface)
- But the amount of light reaching unit surface area depends on the angle between the light and the surface...

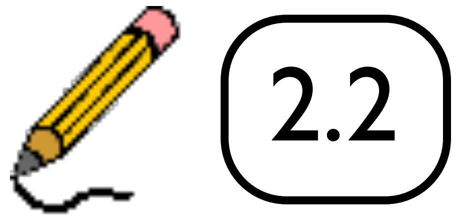


2.1

Specular Reflection

- Light reflected strongly around the mirror reflection direction
- Intensity depends on viewer position

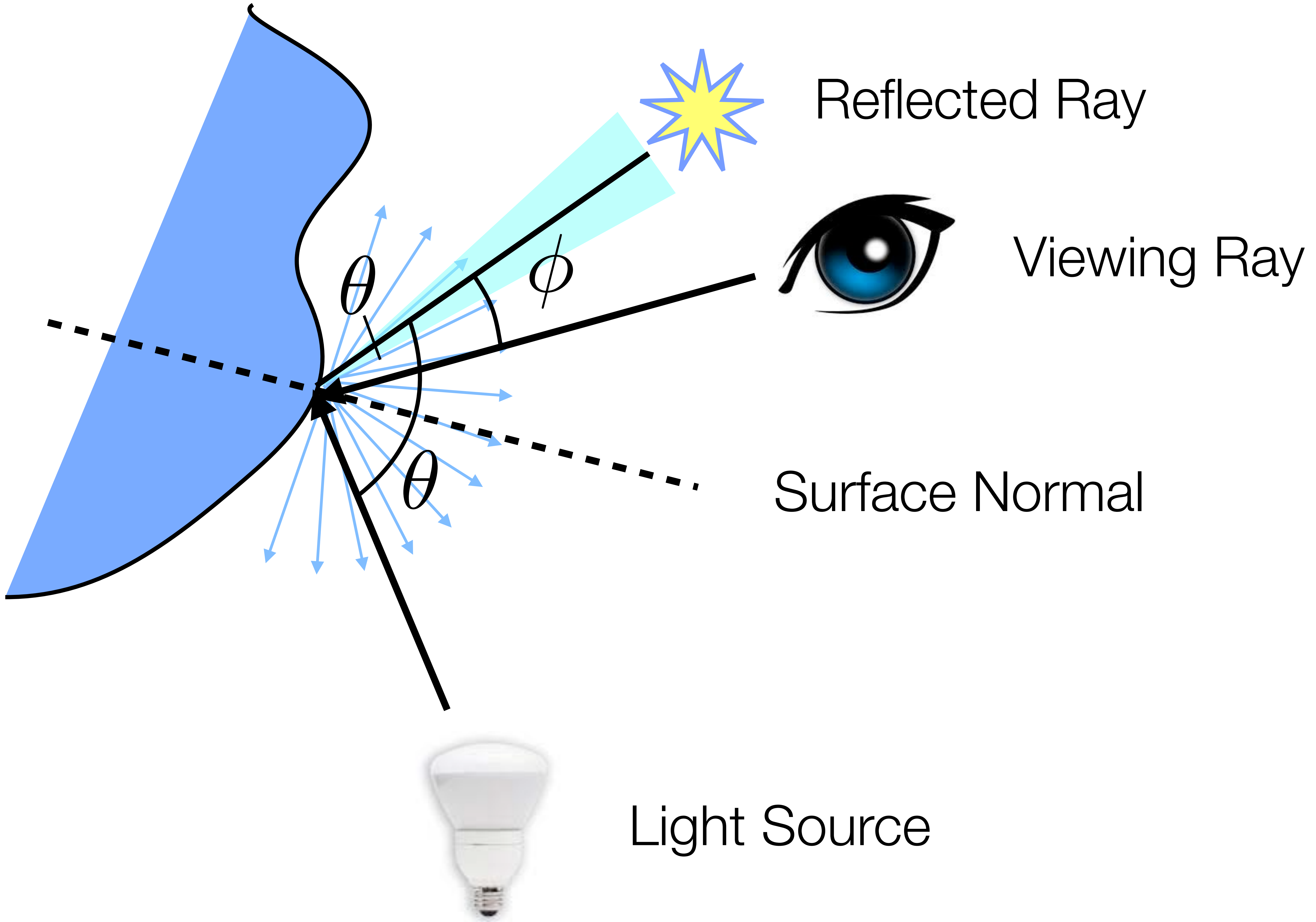




Phong Illumination Model

- Includes ambient, diffuse and specular reflection

$$\boxed{I = k_a i_a + k_d i_d \cos\theta + k_s i_s \cos^{\alpha}\phi}$$

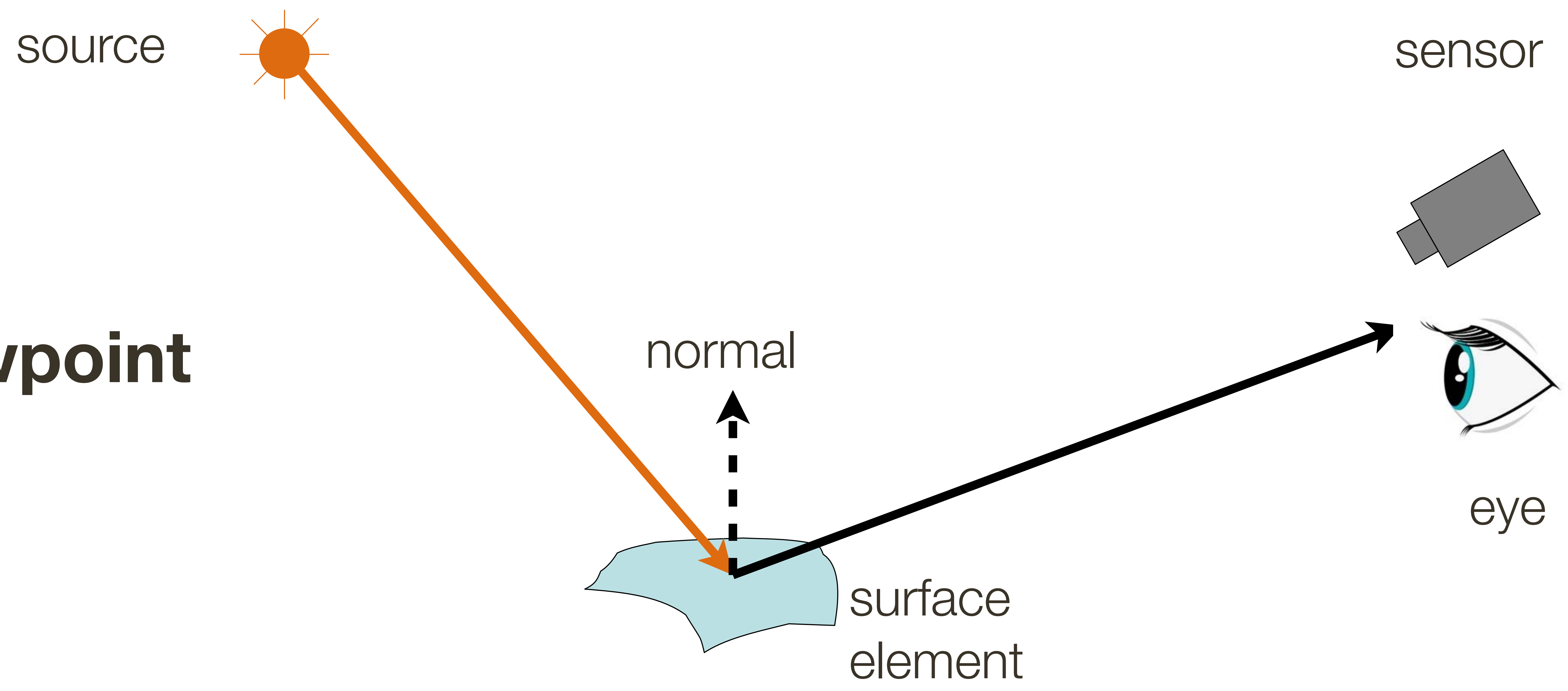


Overview: Image Formation, Cameras and Lenses

Coming back to here…

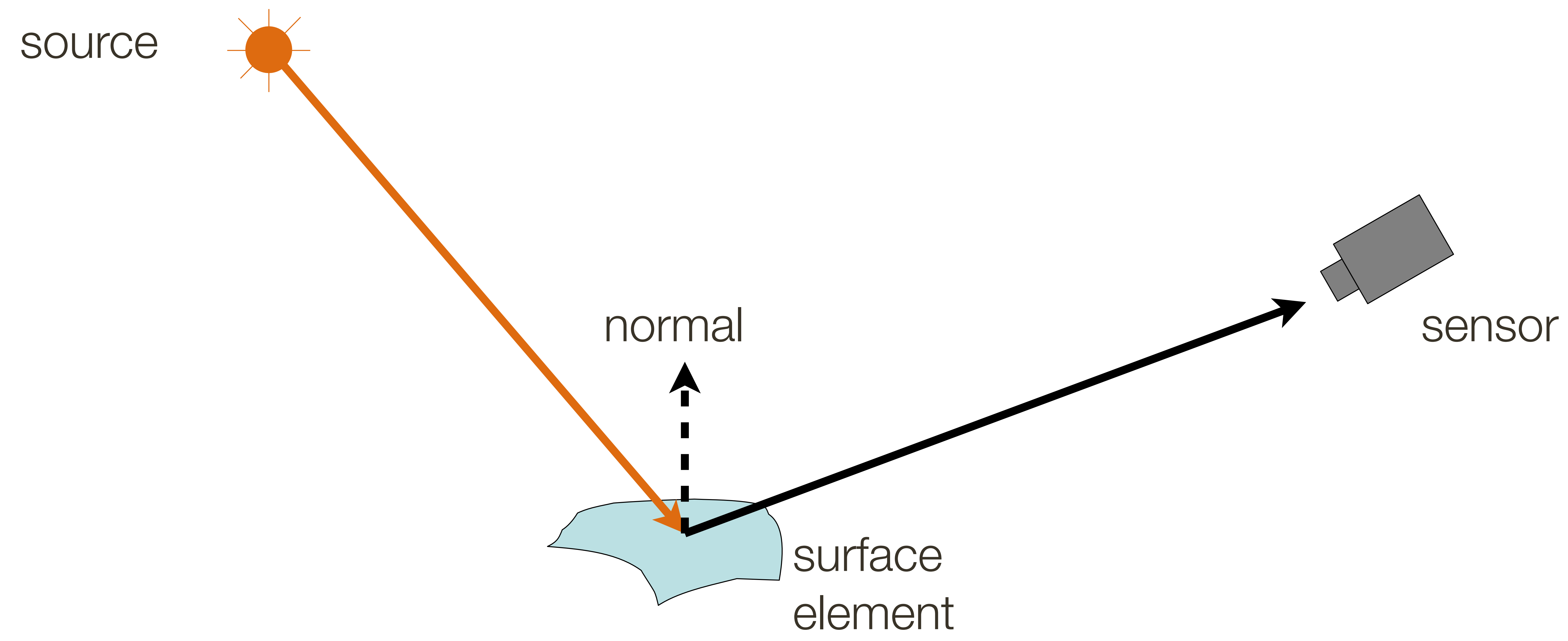
The **image formation process** that produces a particular image depends on

- **Lighting** condition
- Scene **geometry**
- **Surface** properties
- Camera **optics** and **viewpoint**

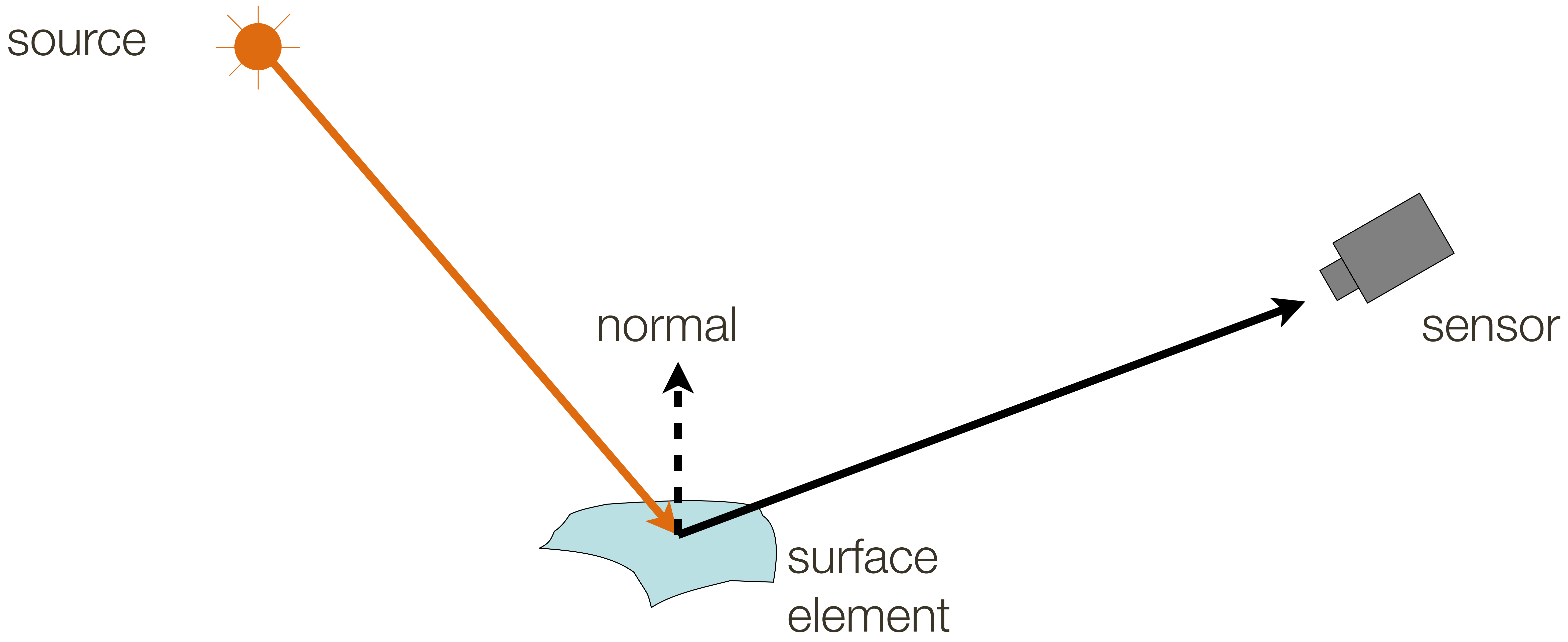


Sensor (or eye) **captures amount of light** reflected from the object

(small) **Graphics** Review

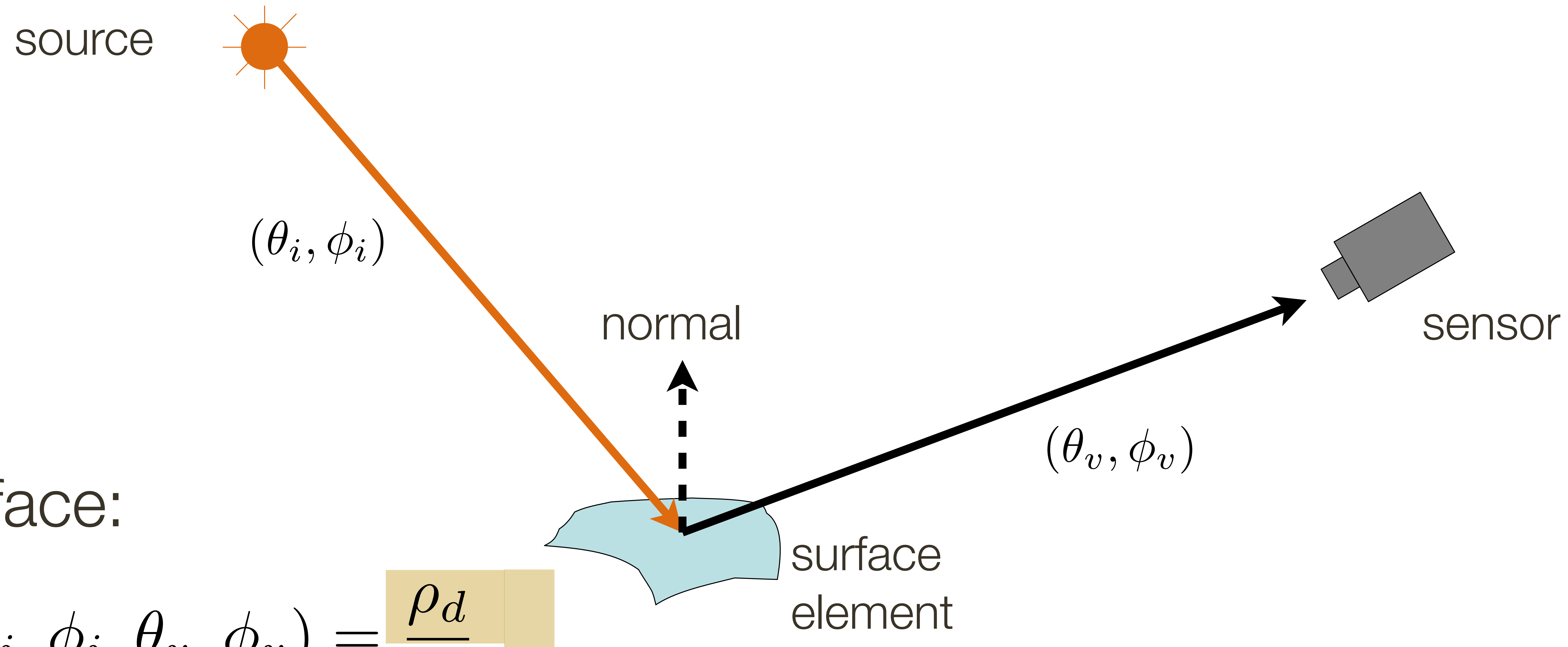


(small) **Graphics** Review



(small) **Graphics** Review

Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



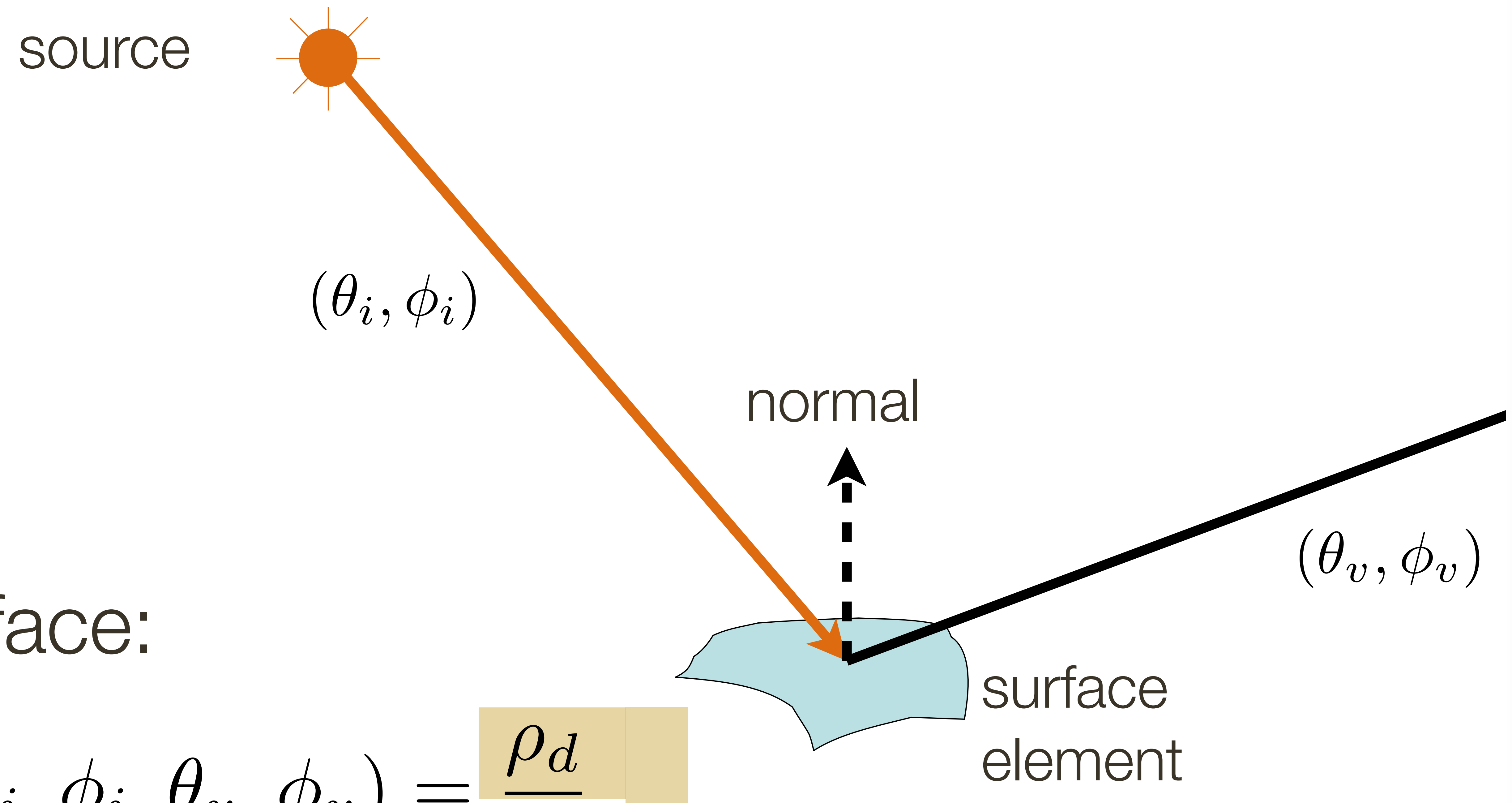
Lambertian surface:

$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

constant, called **albedo**

(small) **Graphics** Review

Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

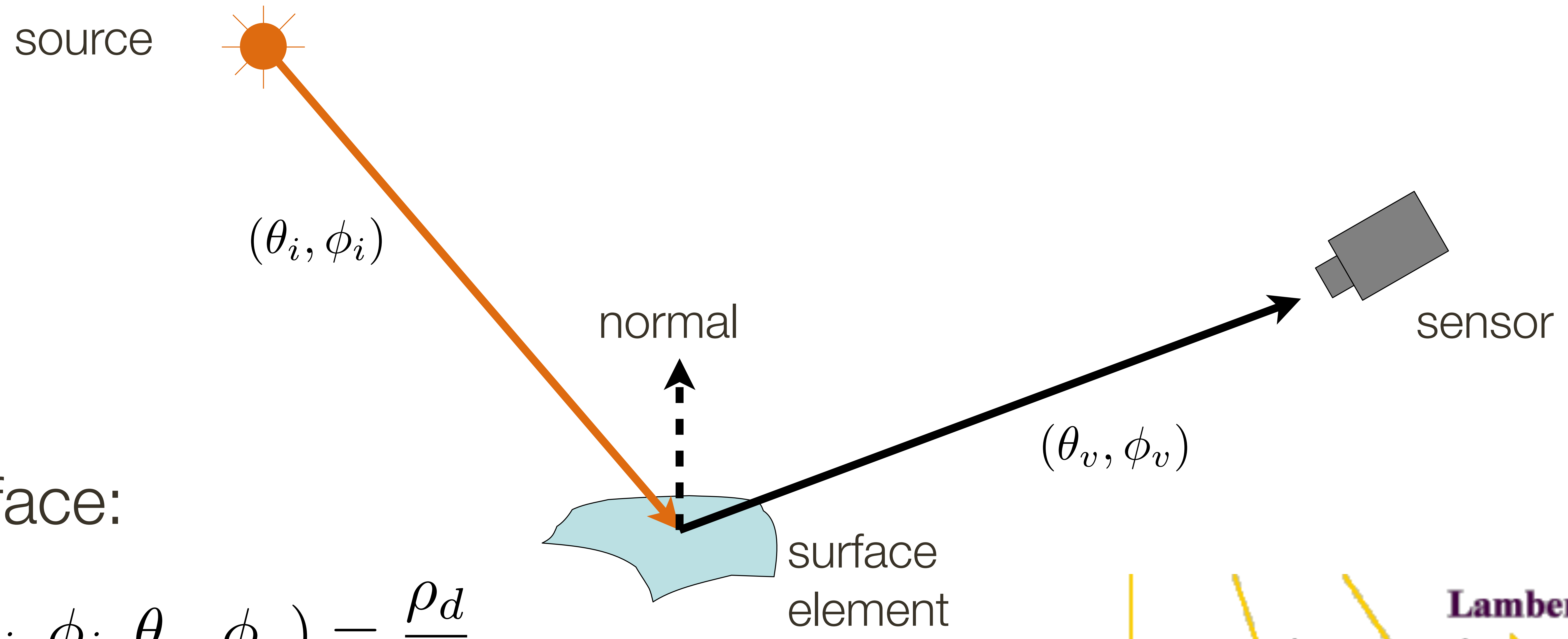
constant, called **albedo**

Surface type	Typical value
Fresh asphalt	0.03 – 0.04
Open ocean	0.06
Conifer forest (summer)	0.08 – 0.15
Worn asphalt	0.12
Deciduous trees	0.15 – 0.18
Sand	0.15 – 0.45
Tundra	0.18 – 0.25
Agricultural crops	0.18 – 0.25
Bare soil	0.17
Green grass	0.20 – 0.25
Dessert sand	0.30 – 0.40
Snow	0.40 – 0.90
Ocean ice	0.50 – 0.70
Fresh snow	0.80 – 0.90

Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

(small) **Graphics** Review

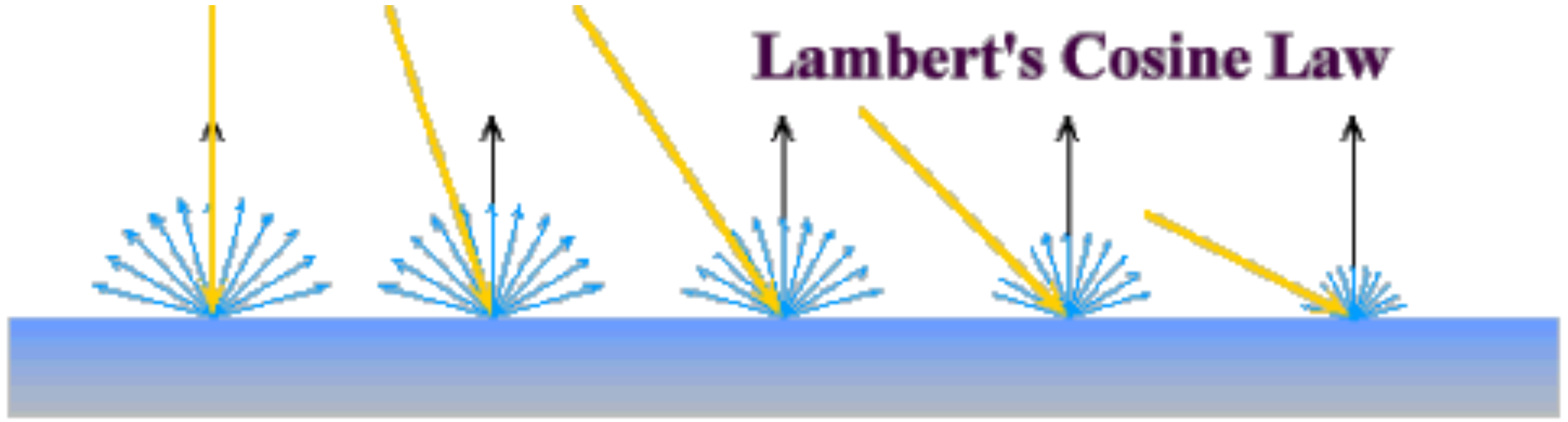
Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

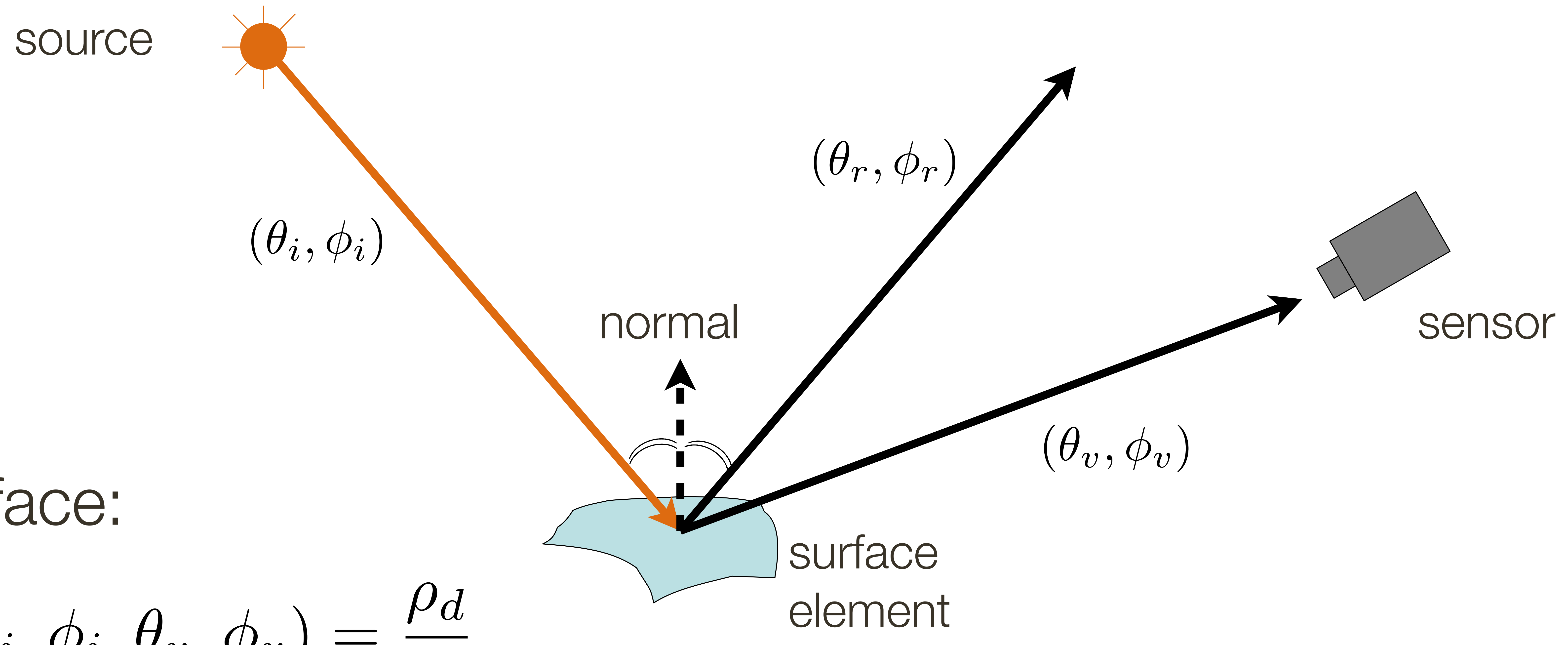
$$L = \frac{\rho_d}{\pi} I(\vec{i} \cdot \vec{n})$$



Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

(small) **Graphics** Review

Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

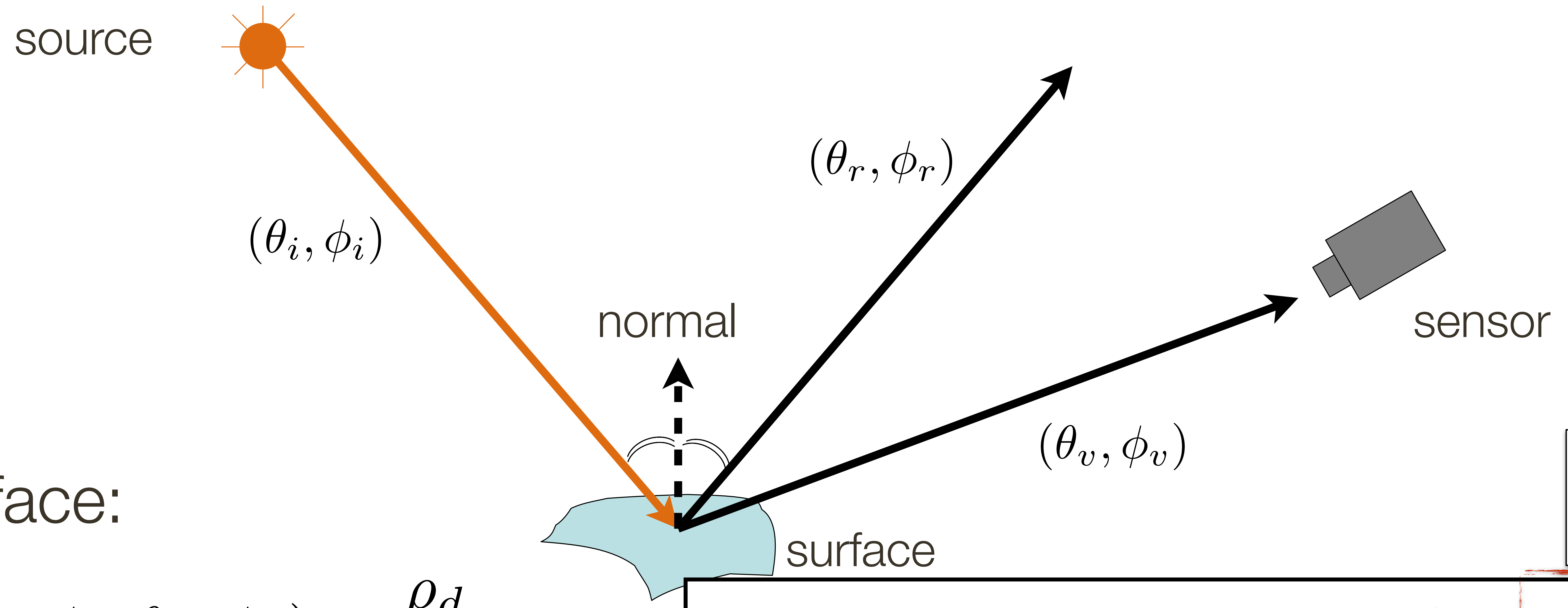
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

Mirror surface: all incident light reflected in one directions $(\theta_v, \phi_v) = (\theta_r, \phi_r)$

Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

(small) **Graphics** Review

Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

Recall...

$$I = k_a i_a + k_d i_d \cos\theta + \boxed{k_s i_s \cos^{\alpha}\phi}$$

Mirror surface: all incident light reflected in one directions $(\theta_v, \phi_v) = (\theta_r, \phi_r)$

Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

Reflectance in Vision

Reflectance in Graphics

[Video is from https://www.youtube.com/watch?v=AdTxrggo8e8 reproduced for educational purposes]

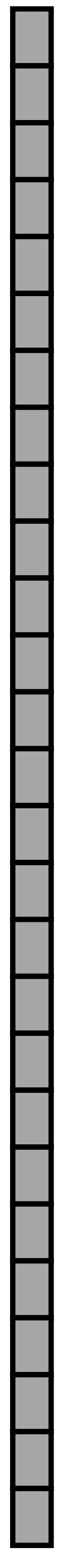
Cameras

Old school **film** camera

Digital CCD/CMOS camera

Let’s say we have a **sensor** …

Digital CCD/CMOS camera



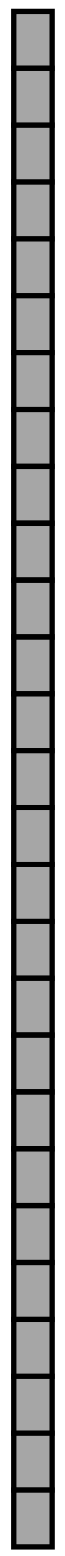
digital sensor
(CCD or
CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

… and the **object** we would like to photograph

What would an image taken like this look like?

real-world
object



digital sensor
(CCD or
CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

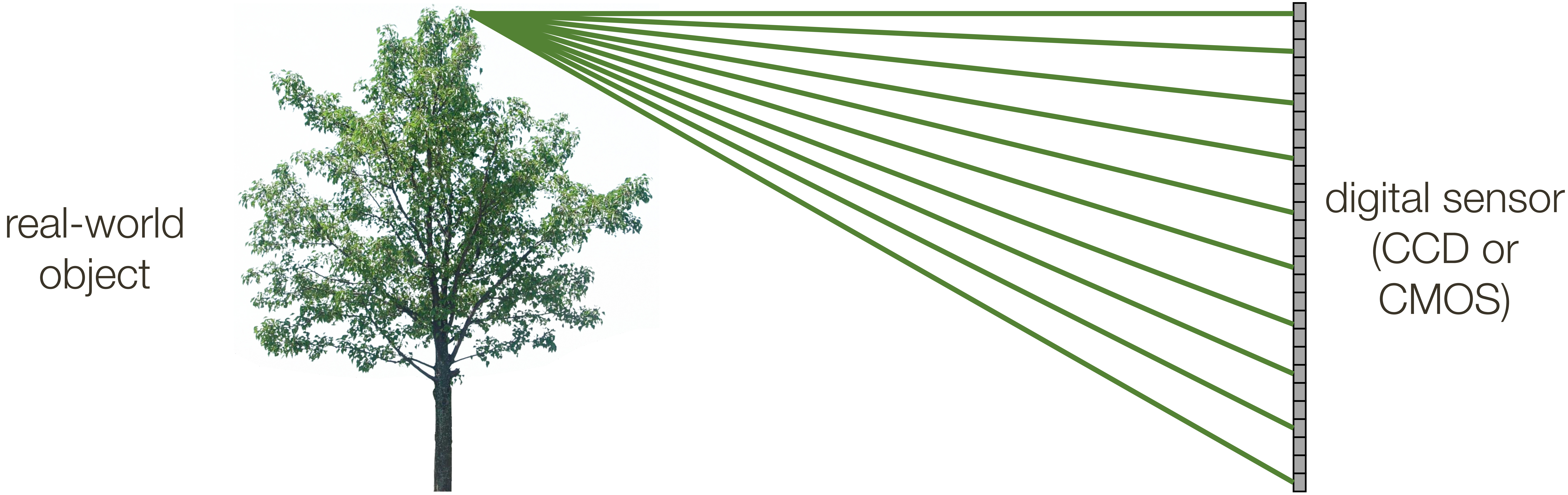
Bare-sensor imaging

real-world object

digital sensor (CCD or CMOS)

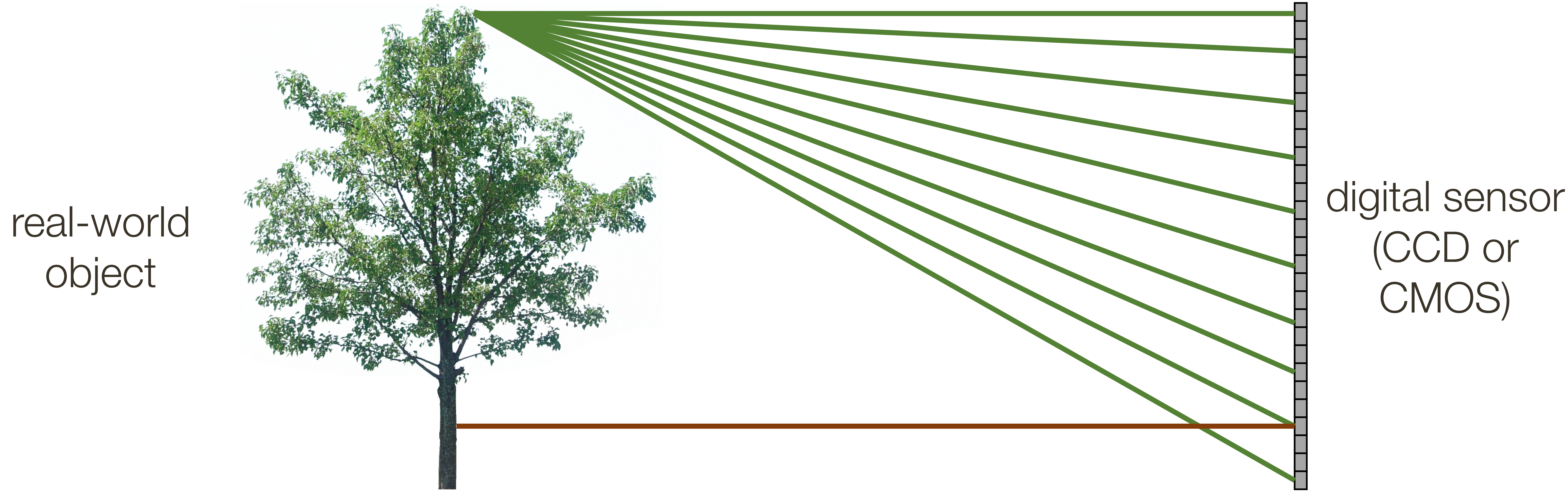
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging



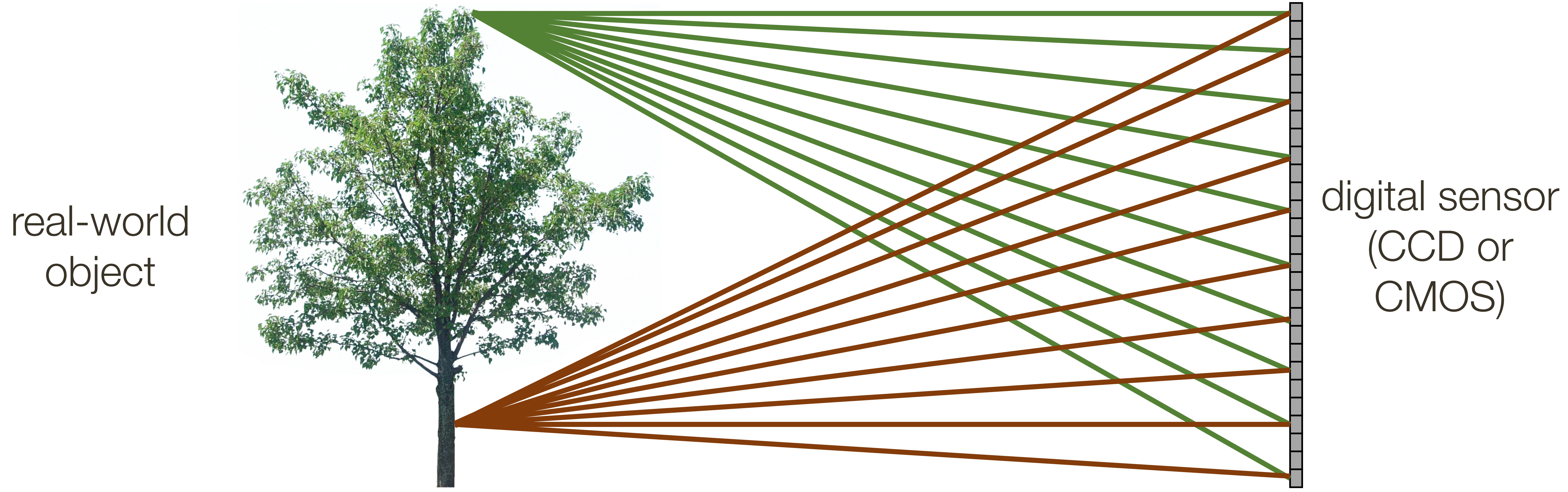
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging



All scene points contribute to all sensor pixels

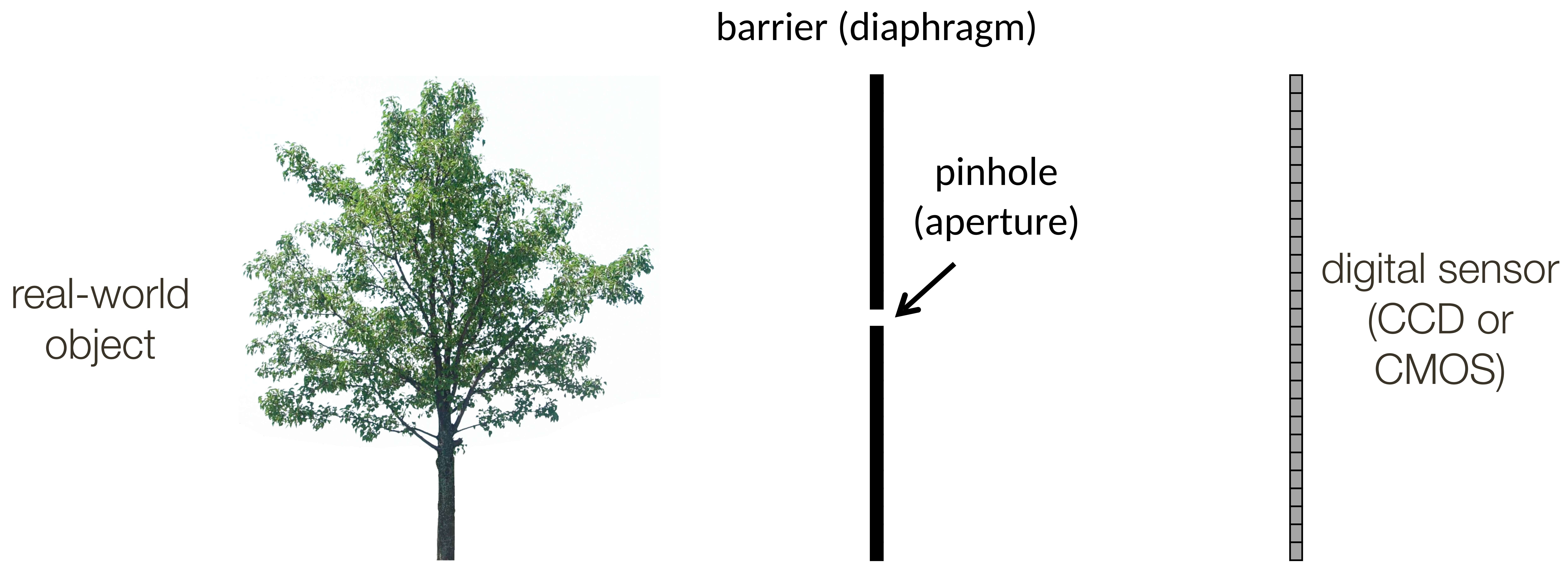
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging

All scene points contribute to all sensor pixels

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

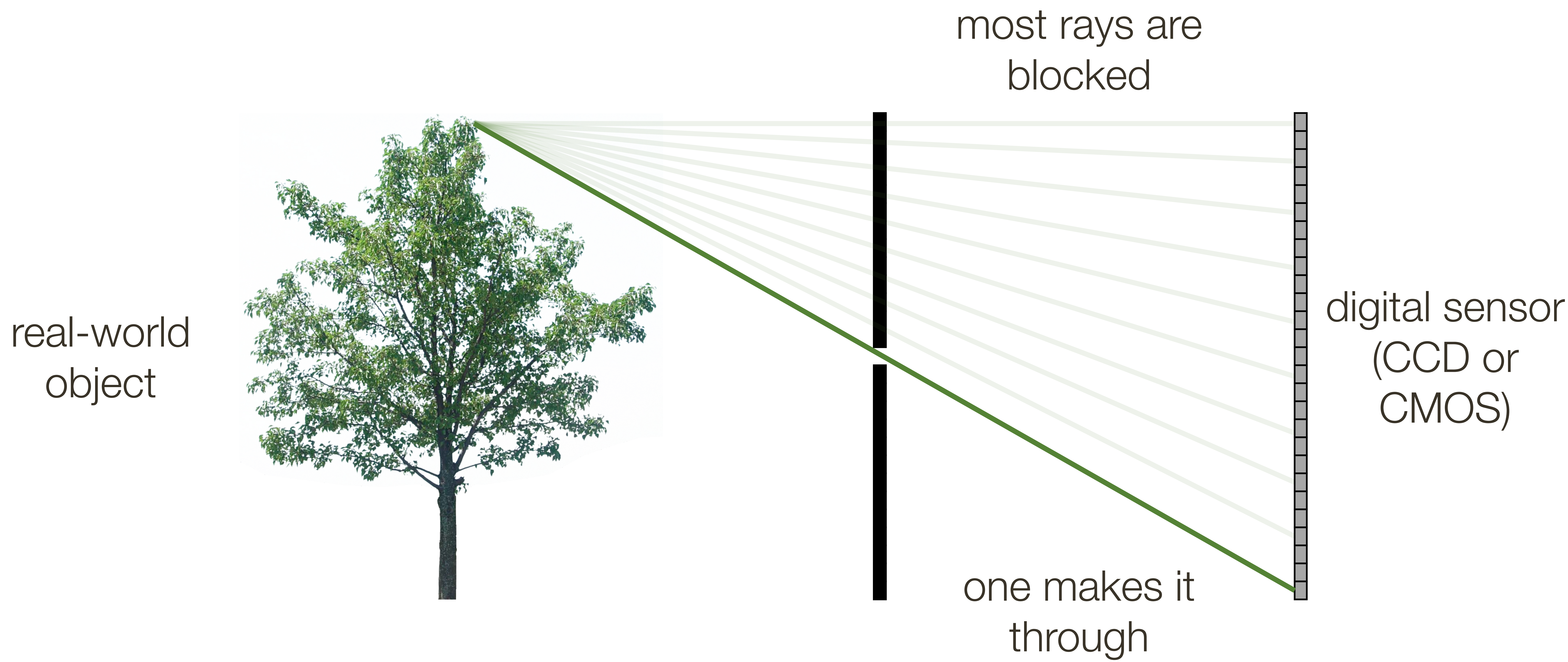
Pinhole Camera



What would an image taken like this look like?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

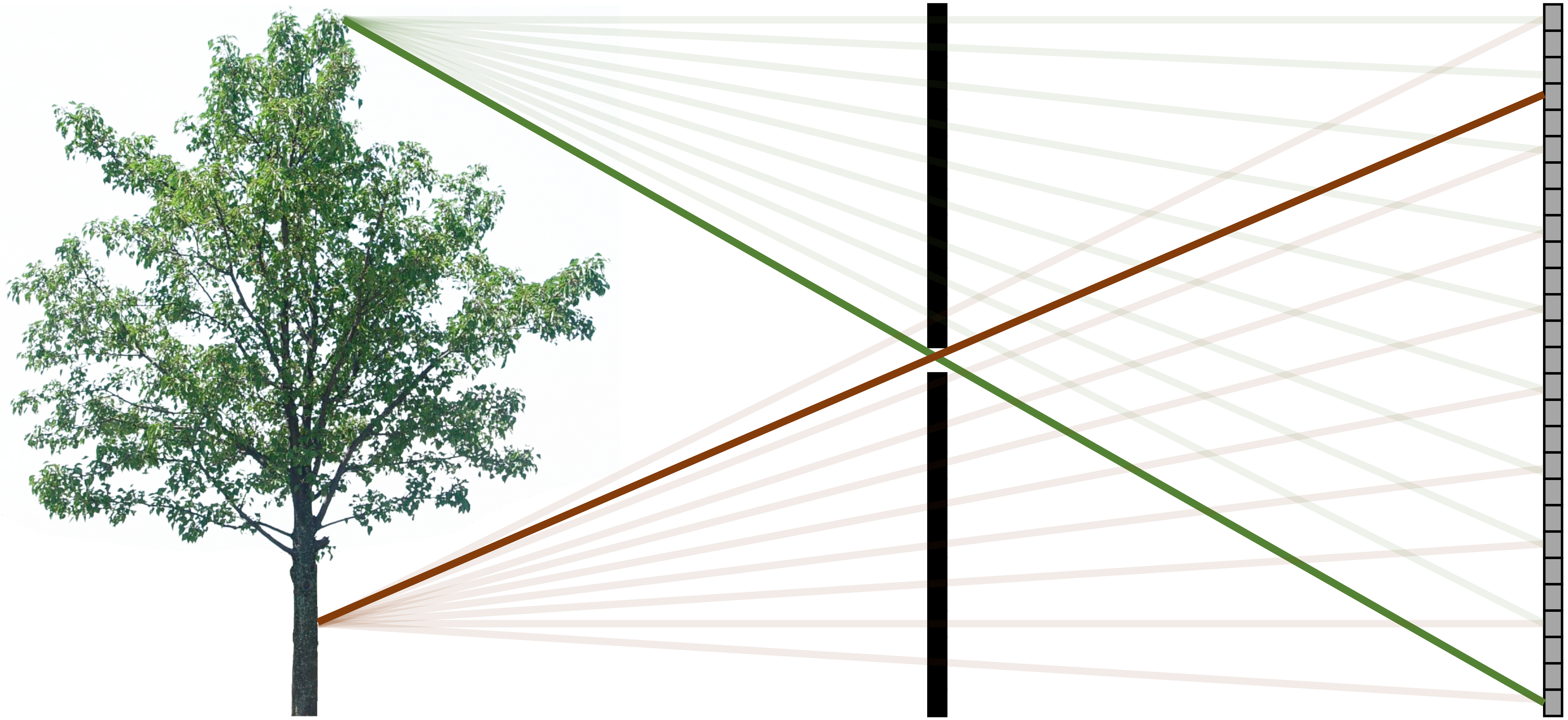
Pinhole Camera



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Pinhole Camera

real-world
object

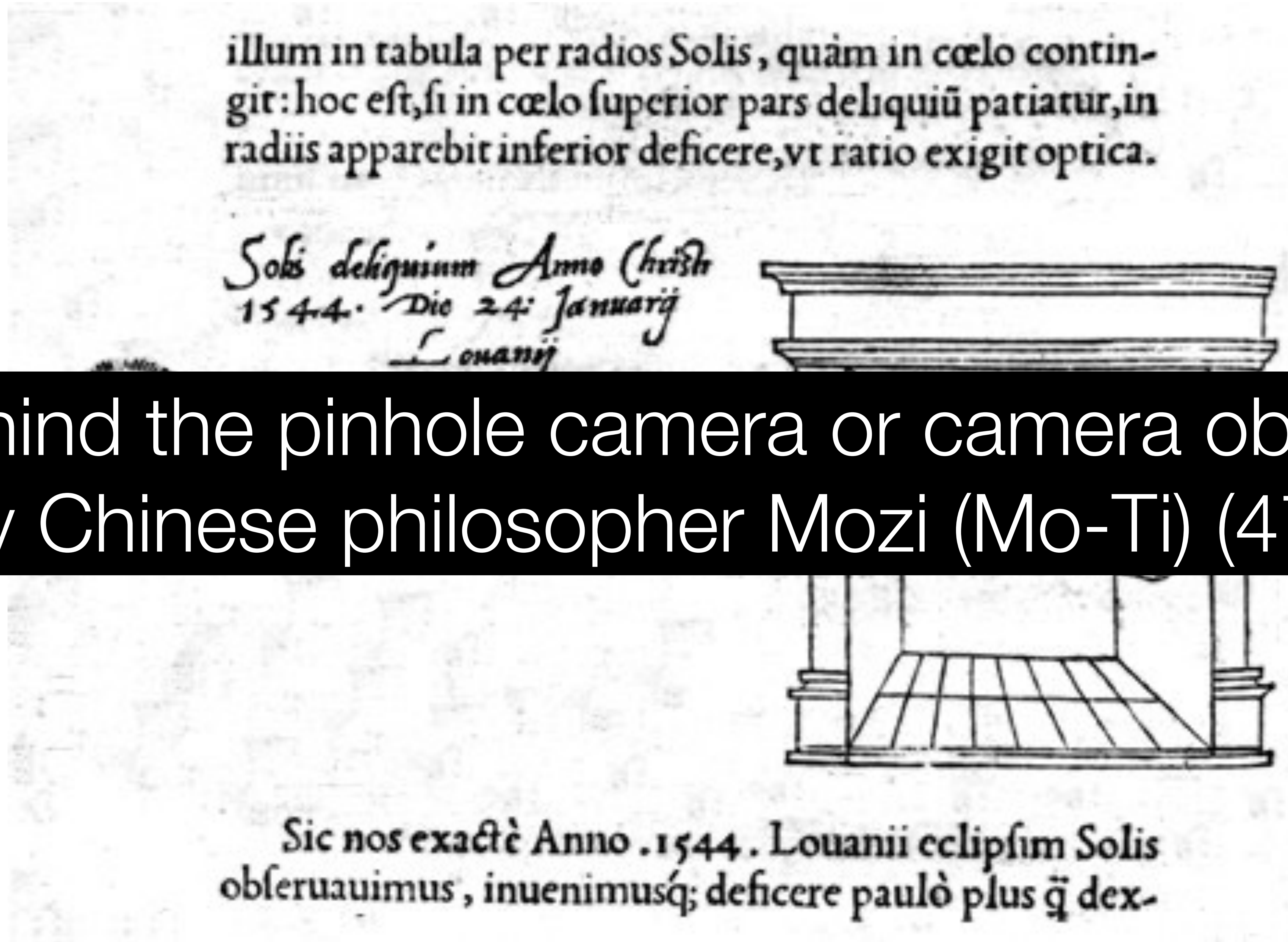


digital sensor
(CCD or
CMOS)

Each scene point contributes to only one sensor pixel

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Camera Obscura (latin for “dark chamber”)



illum in tabula per radios Solis, quàm in cœlo contin-
git: hoc eſt, ſi in cœlo ſuperior pars deliquiũ patiatur, in
radiis apparebit inferior deficere, vt ratio exigit optica.

Sic nos exactè Anno .1544. Louanii eclipſim Solis
obſeruauimus, inuenimusq; deficere paulò plus q̃ dex-

principles behind the pinhole camera or camera obscura were first mentioned by Chinese philosopher Mozi (Mo-Ti) (470 to 390 BCE)

Reinerus Gemma-Frisius observed an eclipse of the sun at Louvain on January 24, 1544. He used this illustration in his book, “De Radio Astronomica et Geometrica,” 1545. It is thought to be the first published illustration of a camera obscura.

Credit: John H., Hammond, “Th Camera Obscure, A Chronicle”

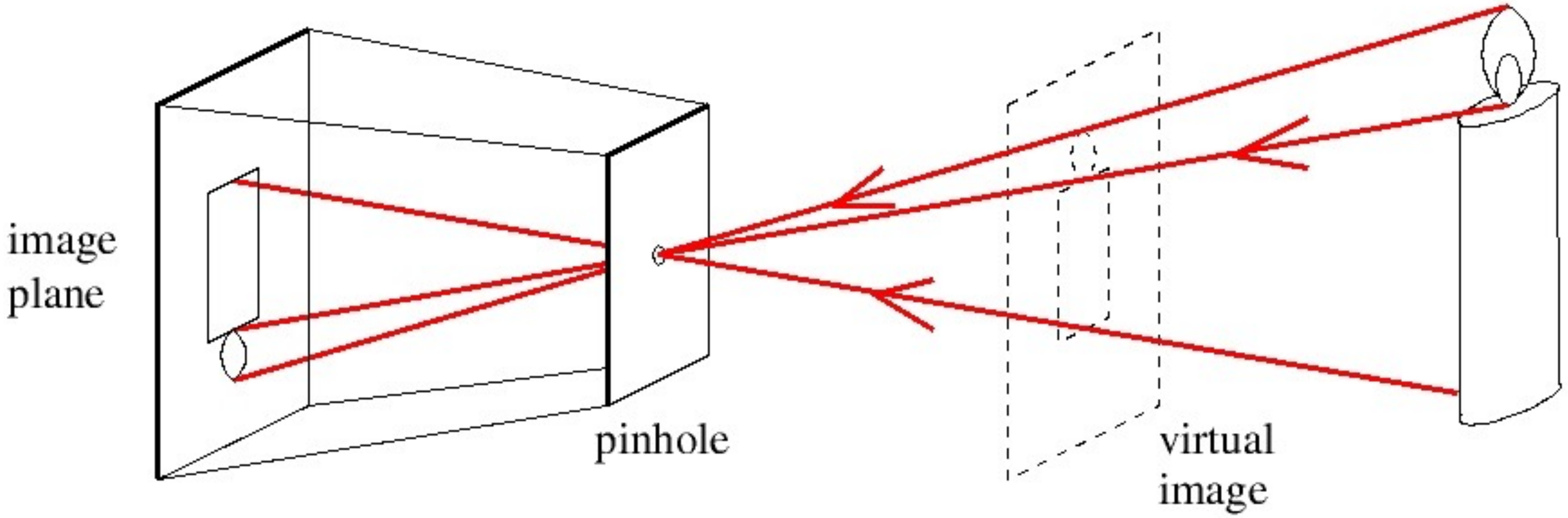
First **Photograph** on Record

La table servie

Credit: Nicéphore Niepce, 1822

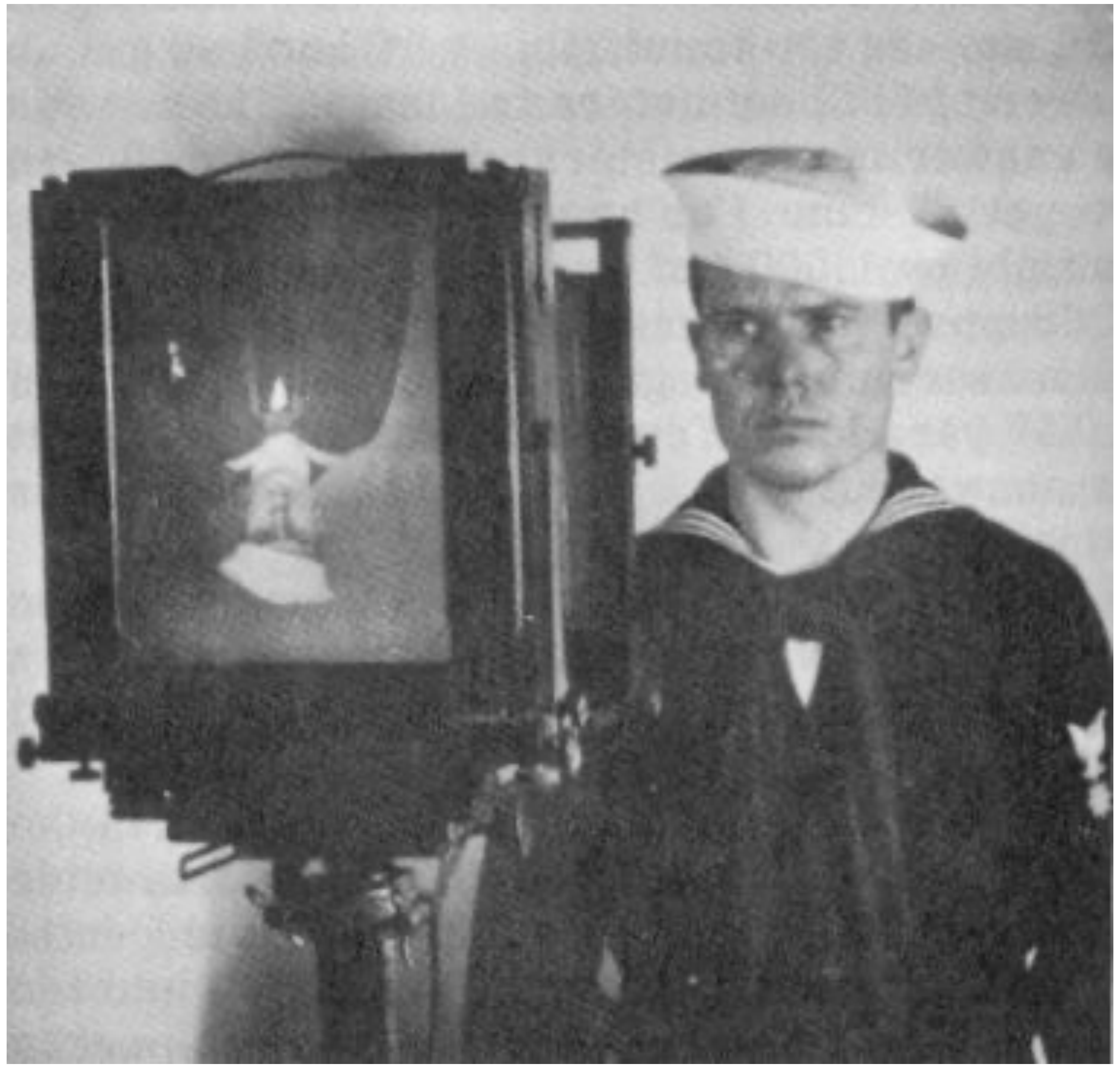
Pinhole Camera

A pinhole camera is a box with a small hole (**aperture**) in it



Forsyth & Ponce (2nd ed.) Figure 1.2

Image **Formation**



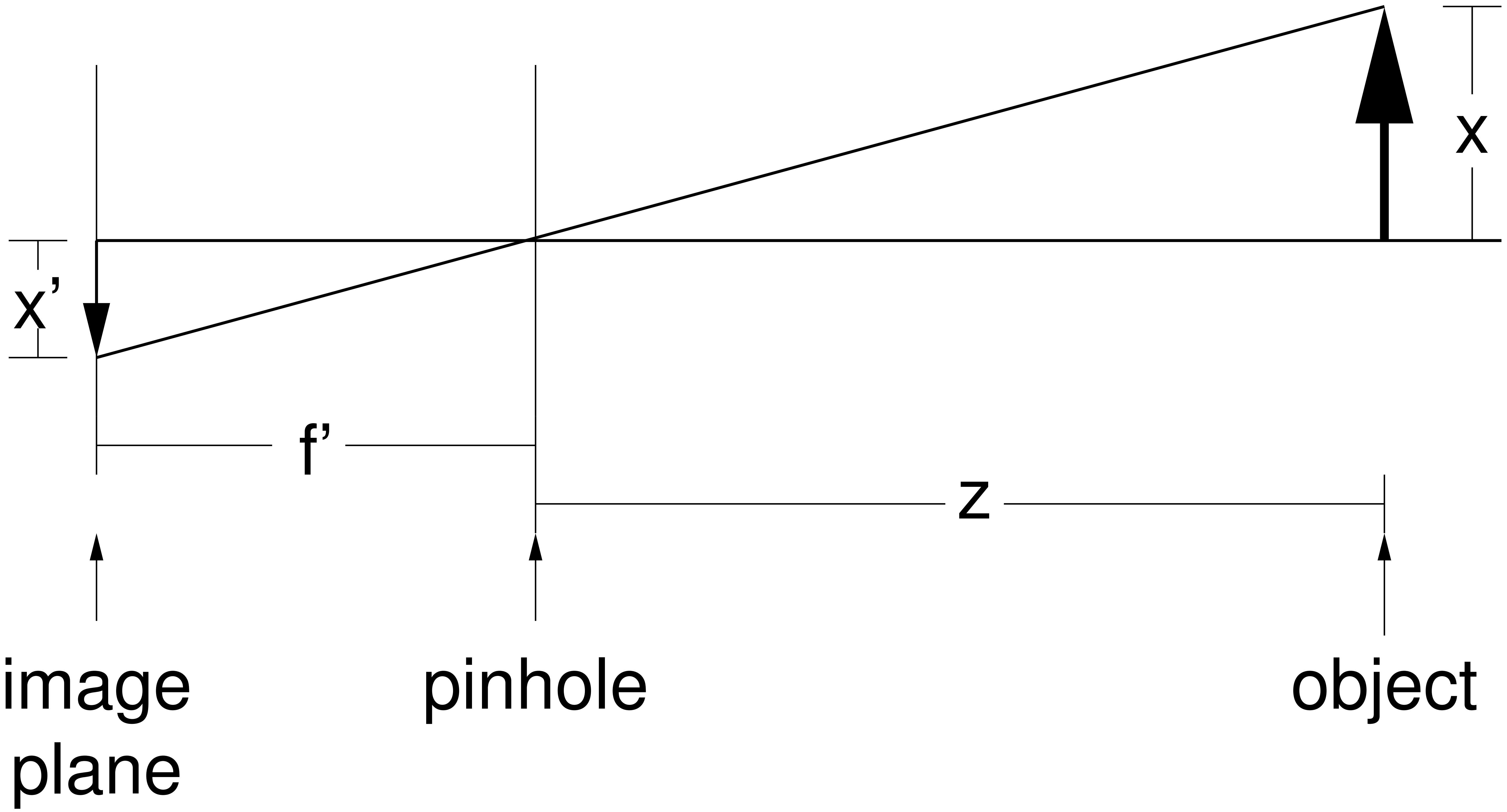
Forsyth & Ponce (2nd ed.) Figure 1.1

Credit: US Navy, Basic Optics and Optical Instruments. Dover, 1969

Accidental Pinhole Camera

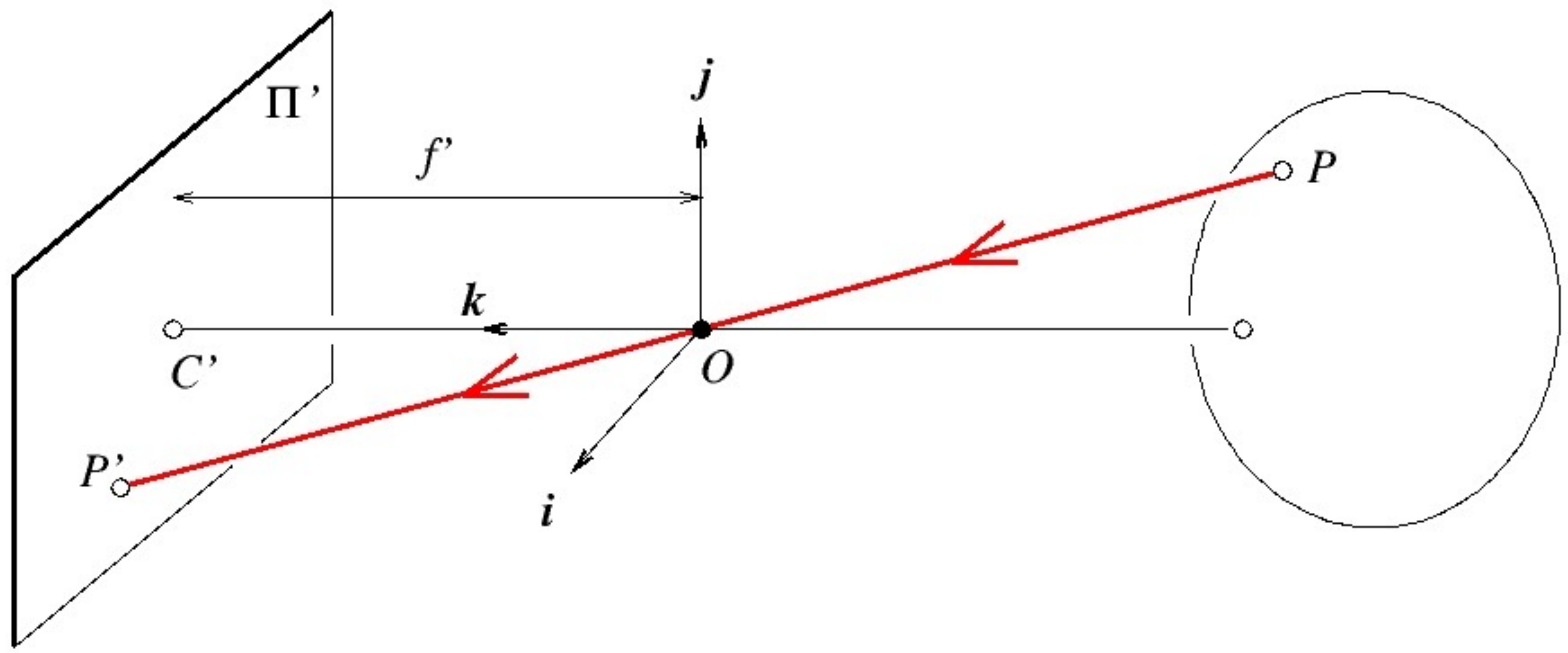
Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Pinhole Camera



2.3

Perspective Projection



3D object point

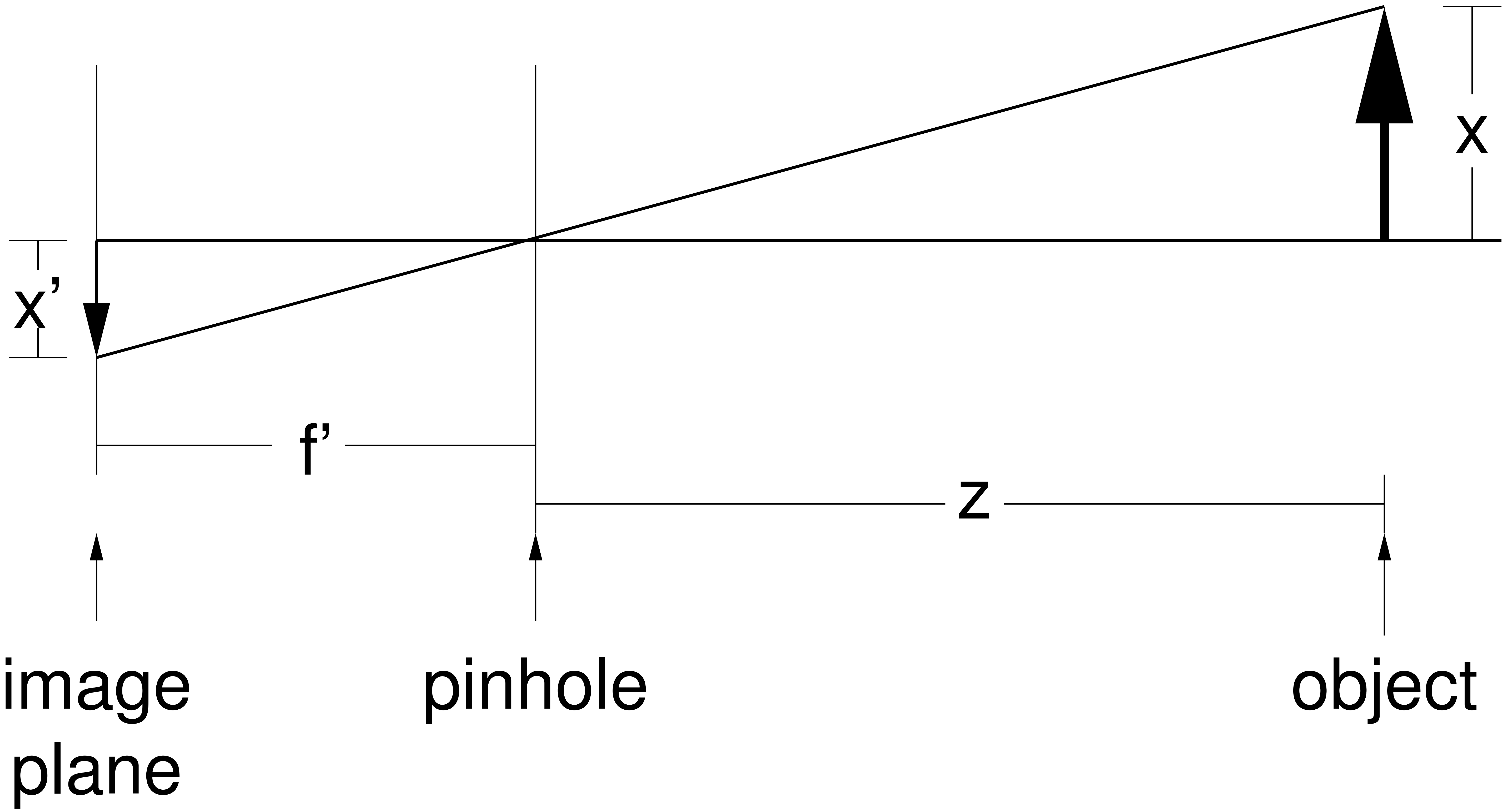
Forsyth & Ponce (1st ed.) Figure 1.4

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$ projects to 2D image point $$P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$ where

$$\boxed{\begin{aligned} x' &= f' \frac{x}{z} \\ y' &= f' \frac{y}{z} \end{aligned}}$$

Pinhole Camera

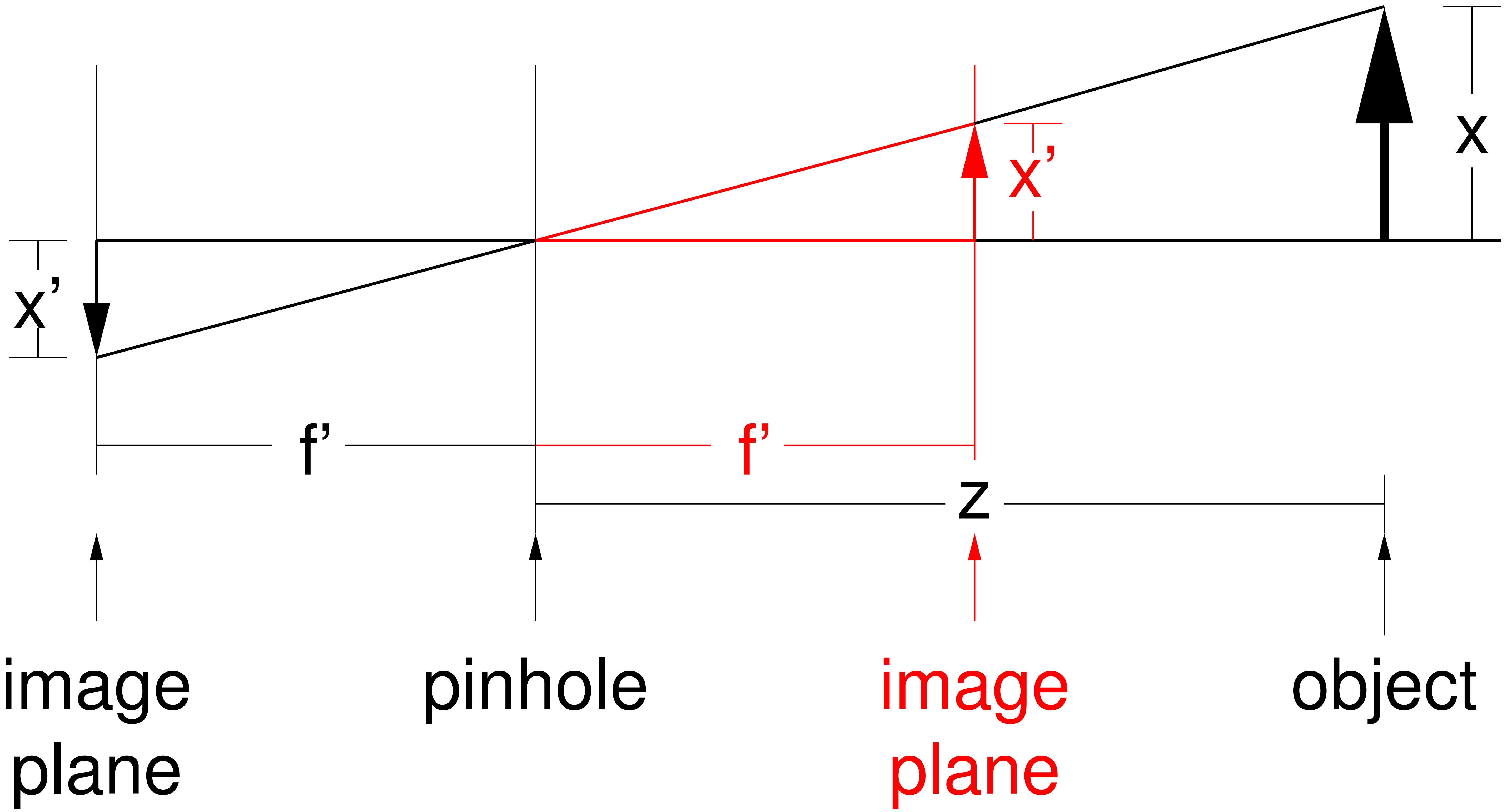
f’ is the **focal length** of the camera



Note: In a pinhole camera we can adjust the focal length, all this will do is change the **size** of the resulting image

Pinhole Camera

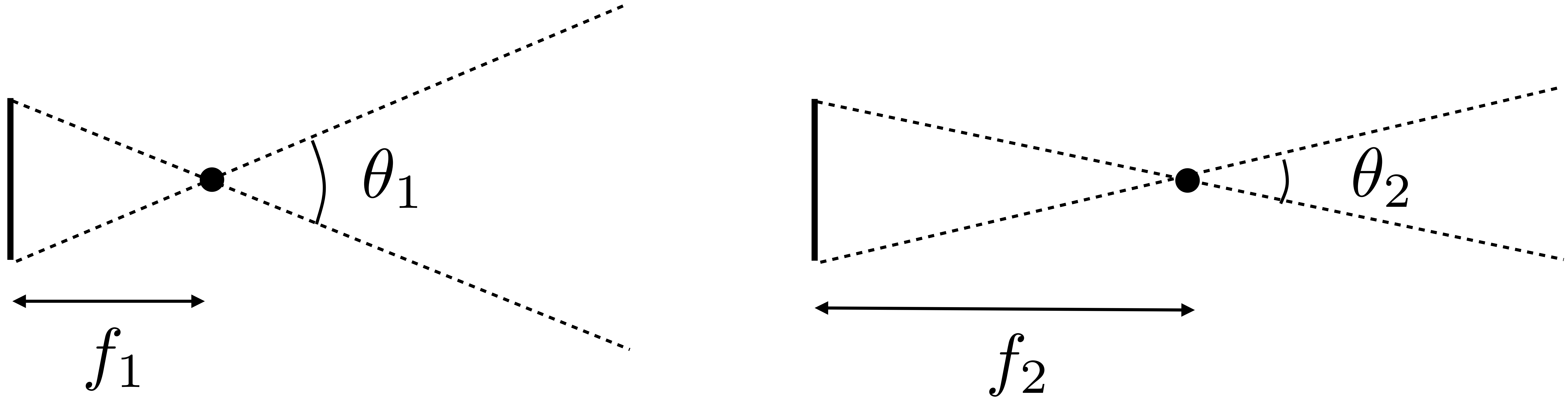
It is convenient to think of the **image plane** being in front of the pinhole



What happens if object moves towards the camera? Away from the camera?

Focal Length

- For a fixed sensor size, focal length determines the field of view (fov)



2.5 **Q:** What is the field of view of a full-frame (35mm) camera with a 50mm lens? 100mm lens?

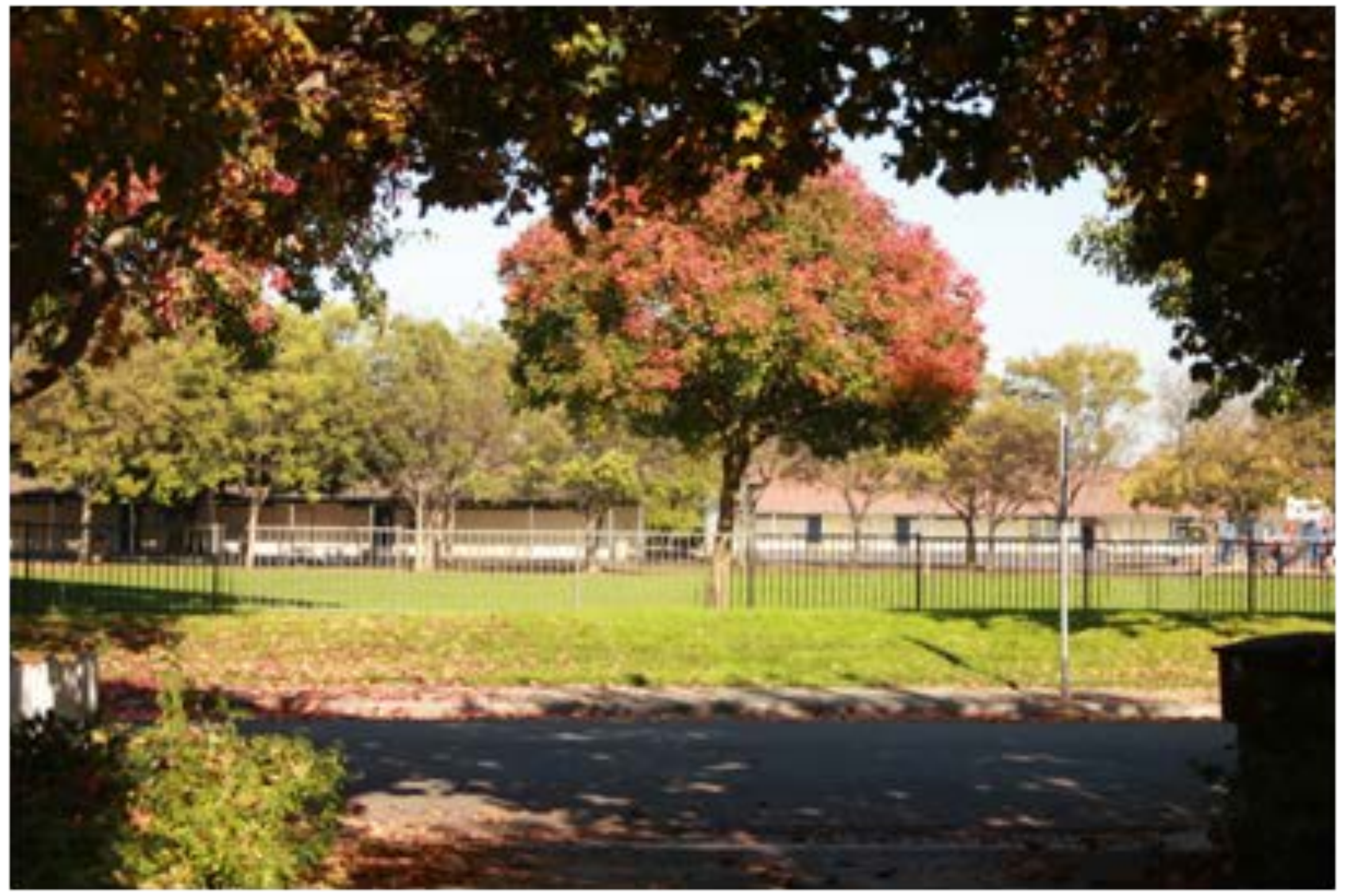
Sensor size

Focal length

Focal Length

28 mm

35 mm

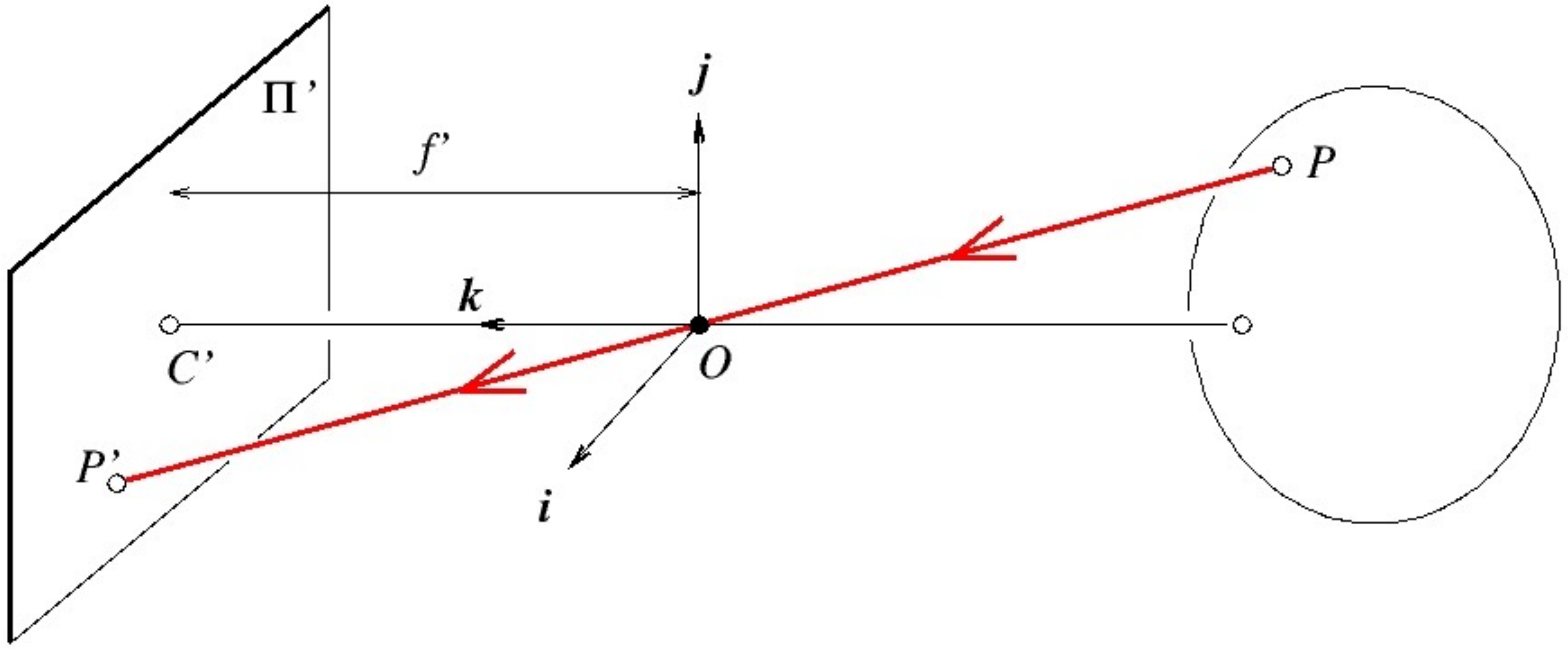


50 mm

70 mm

Perspective Projection: Matrix Form

Camera Matrix



$$\mathbf{C} = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3D object point

Forsyth & Ponce (1st ed.) Figure 1.4

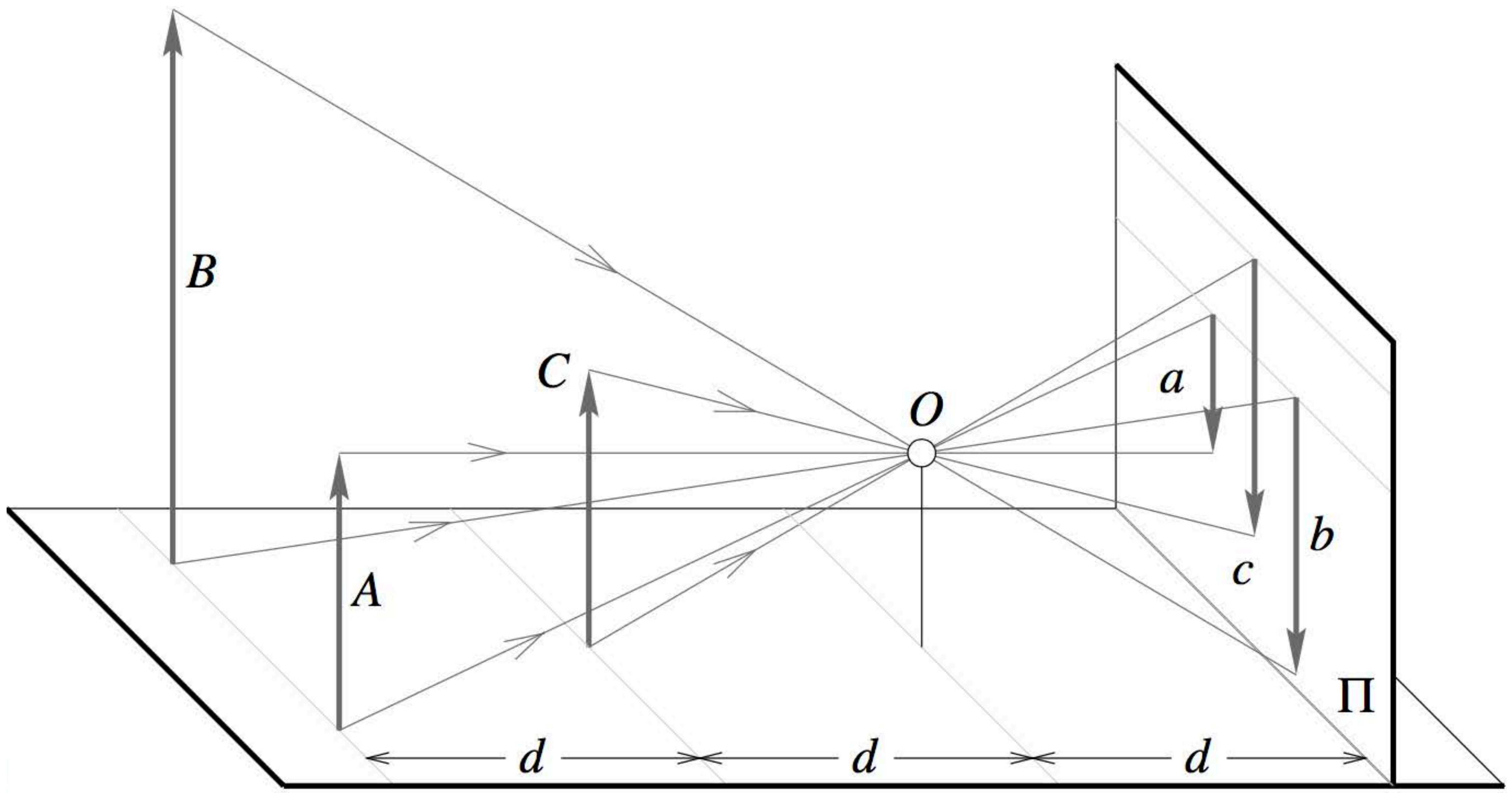
$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{sP' = \mathbf{C}P}$

(s is a scale factor)

2.4

Perspective Effects

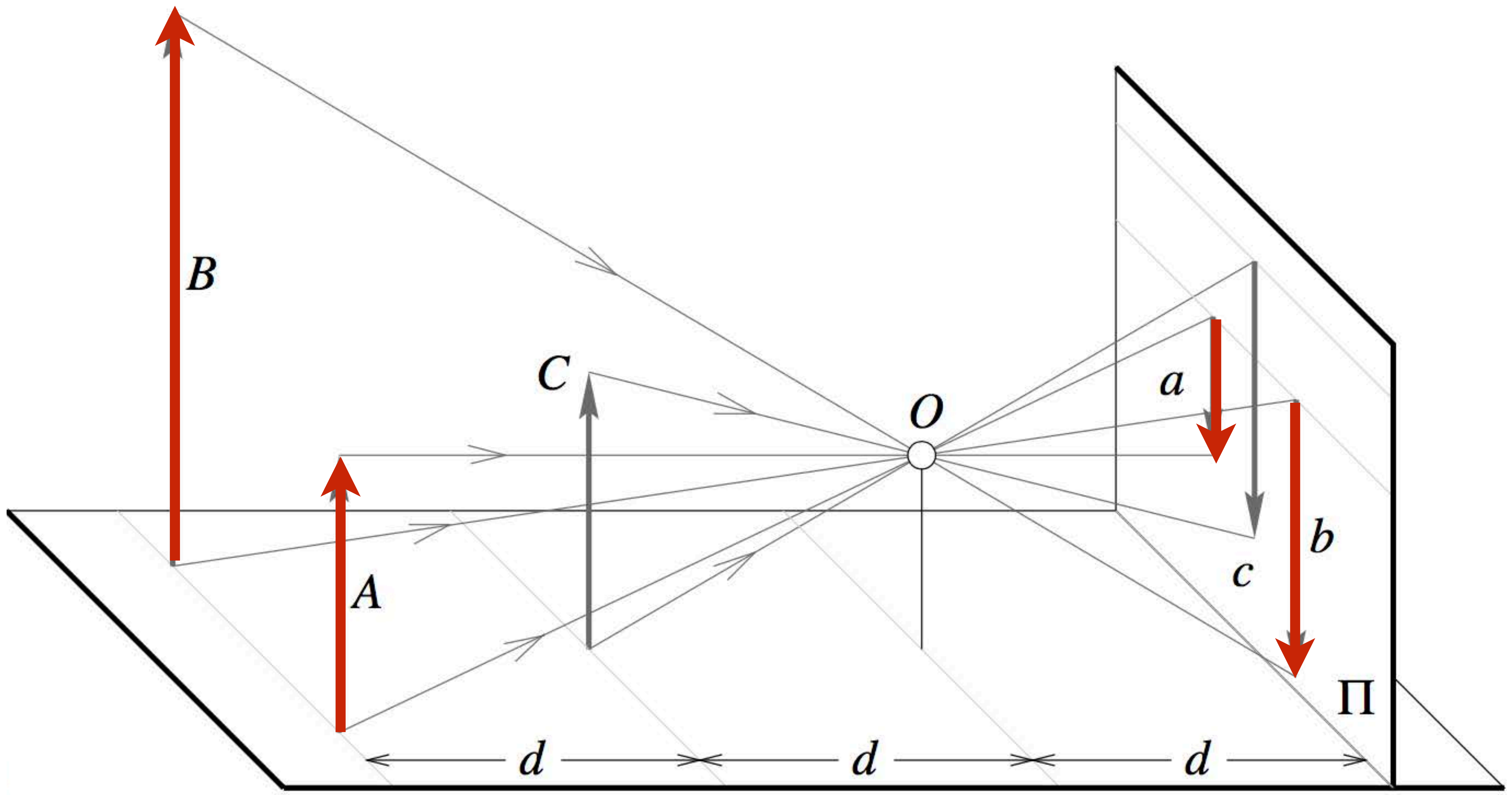
Far objects appear **smaller** than close ones



Forsyth & Ponce (2nd ed.) Figure 1.3a

Perspective Effects

Far objects appear **smaller** than close ones

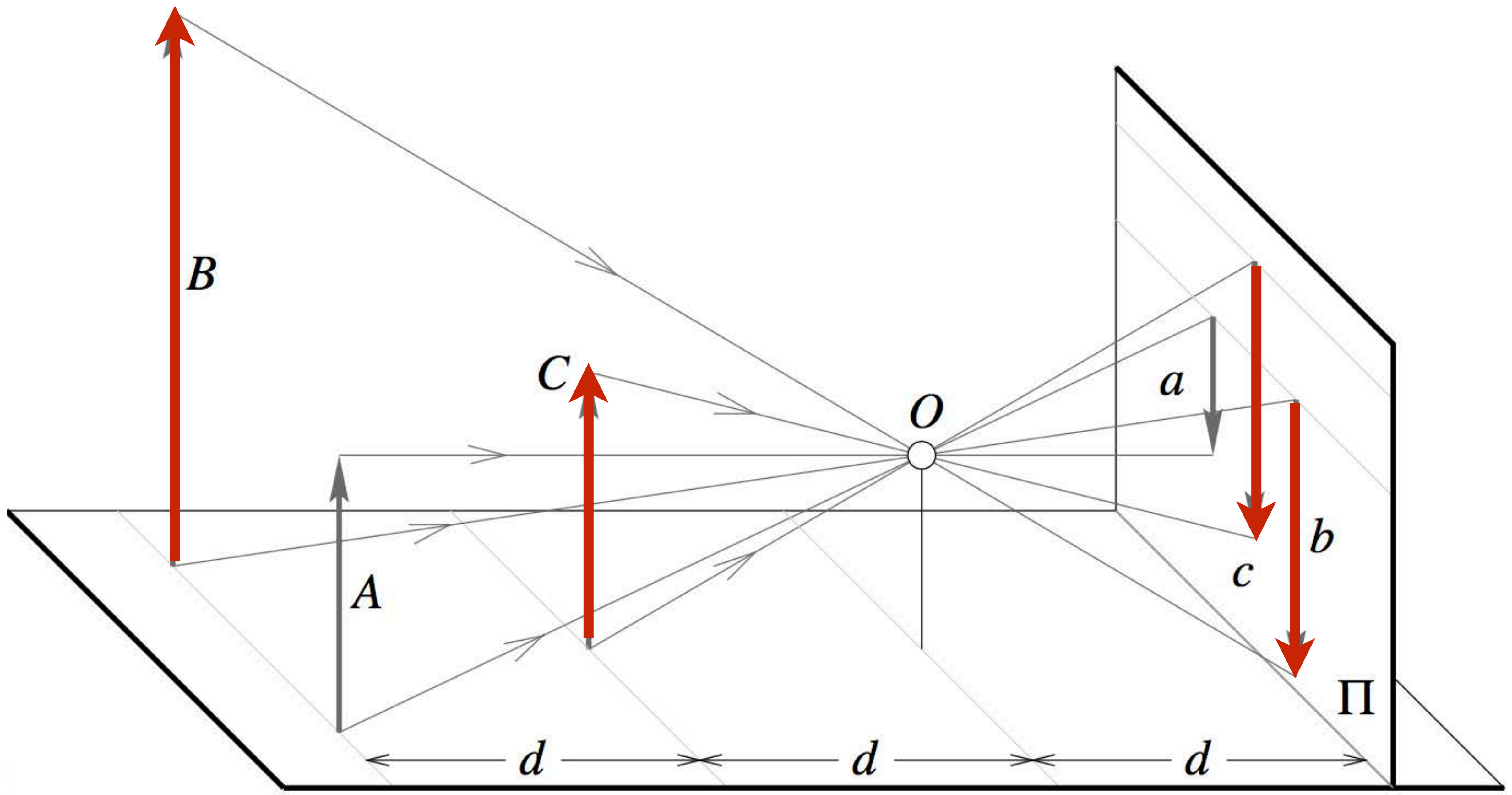


Forsyth & Ponce (2nd ed.) Figure 1.3a

Size is **inversely** proportions to distance

Perspective Effects

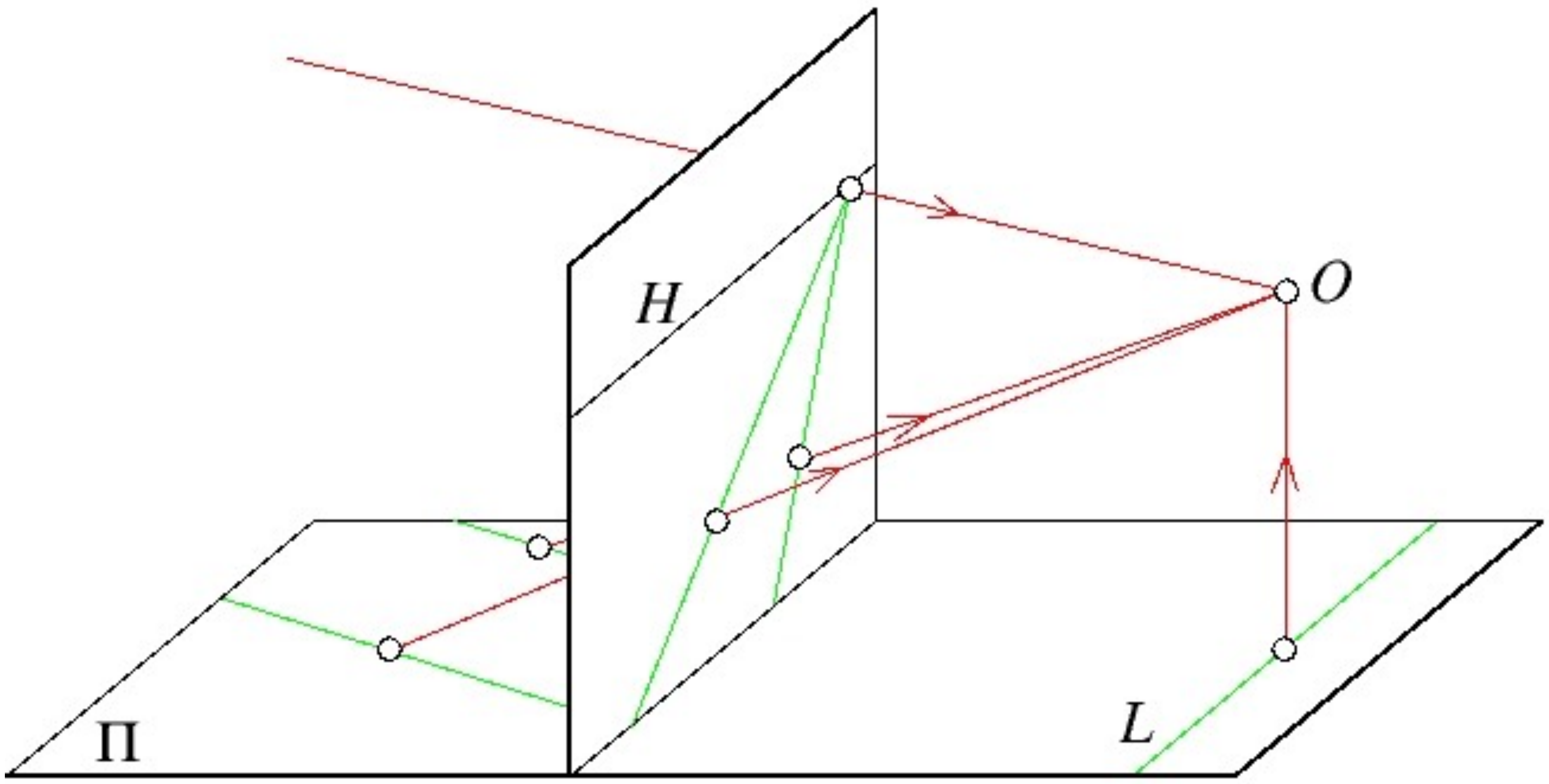
Far objects appear **smaller** than close ones



Forsyth & Ponce (2nd ed.) Figure 1.3a

Perspective Effects

Parallel lines meet at a point (**vanishing point**)



Forsyth & Ponce (1st ed.) Figure 1.3b

Vanishing Points

Each set of parallel lines meets at a different point

— the point is called the **vanishing point**

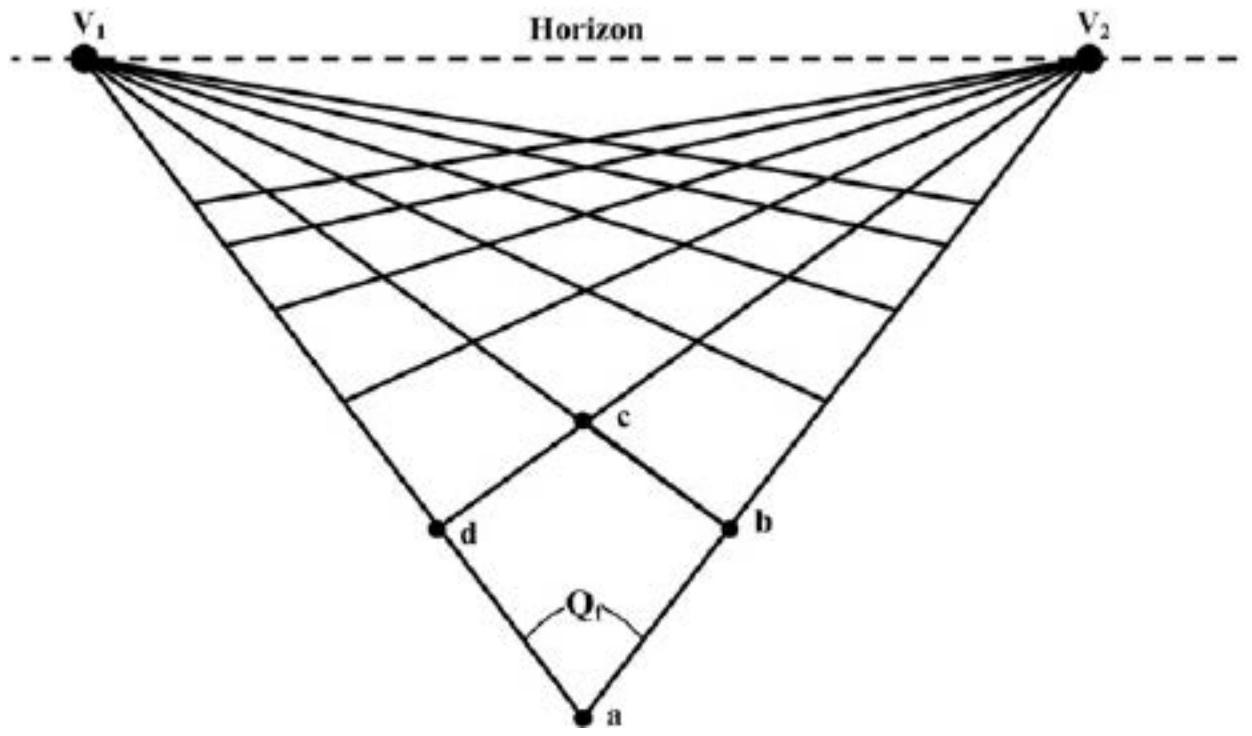
Vanishing Points

Each set of parallel lines meets at a different point

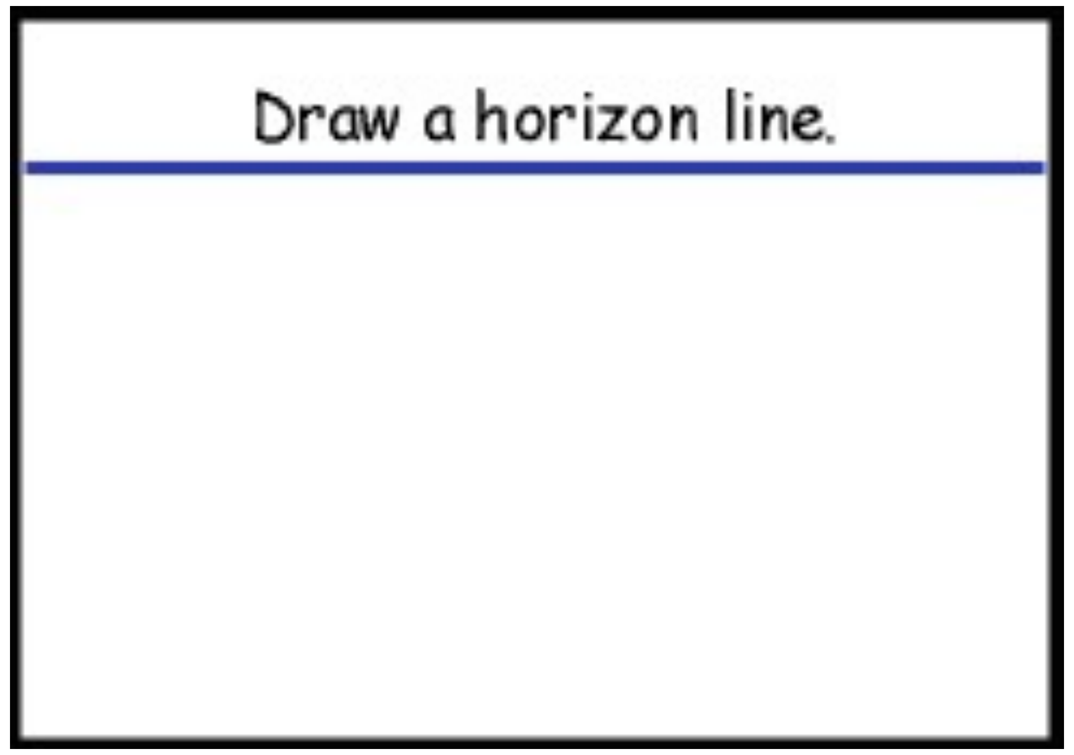
— the point is called the **vanishing point**

Sets of parallel lines on the same plane lead to **collinear** vanishing points

— the line is called a **horizon** for that plane

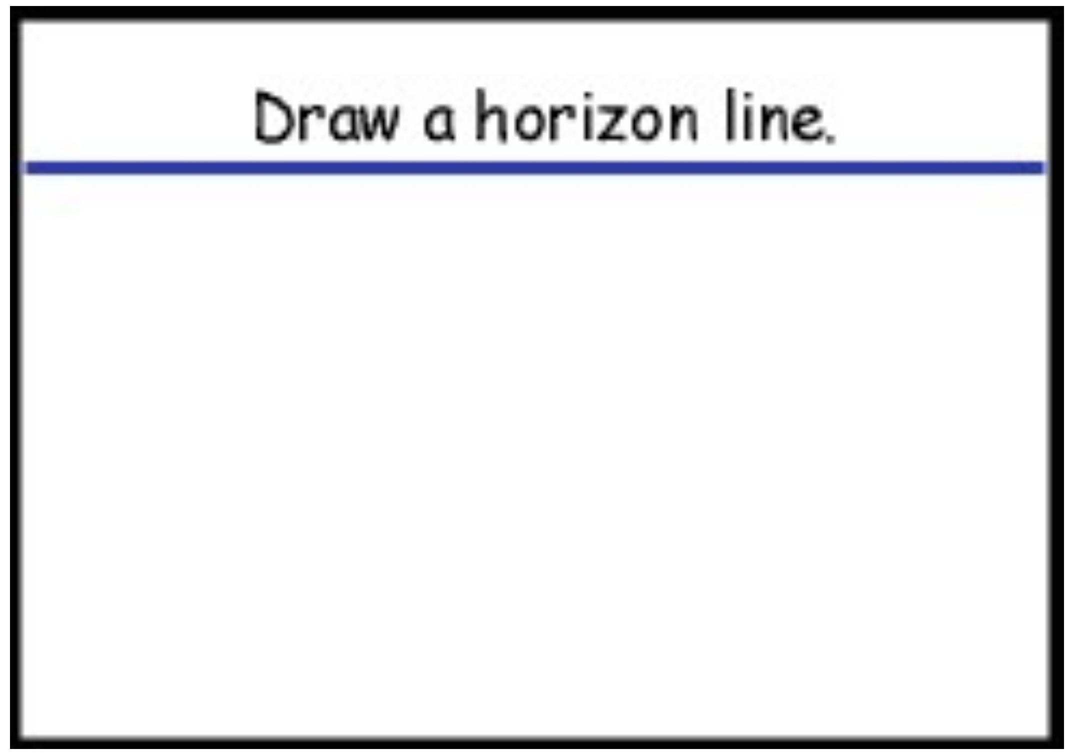


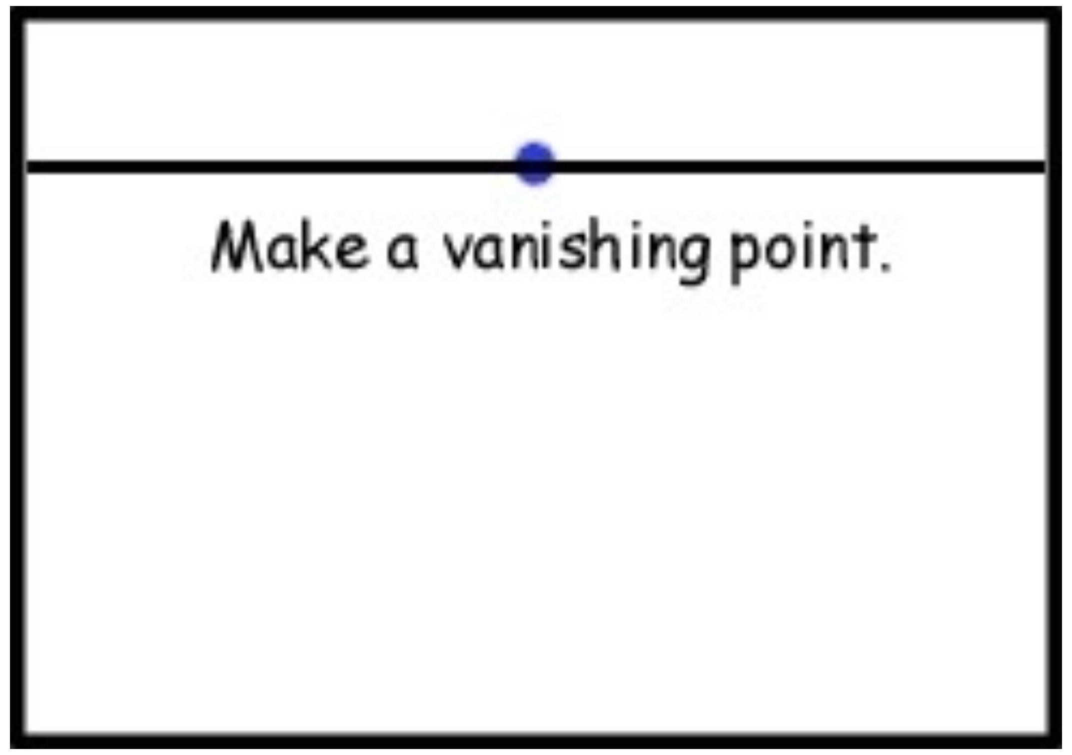
Vanishing Points



Slide Credit: David Jacobs

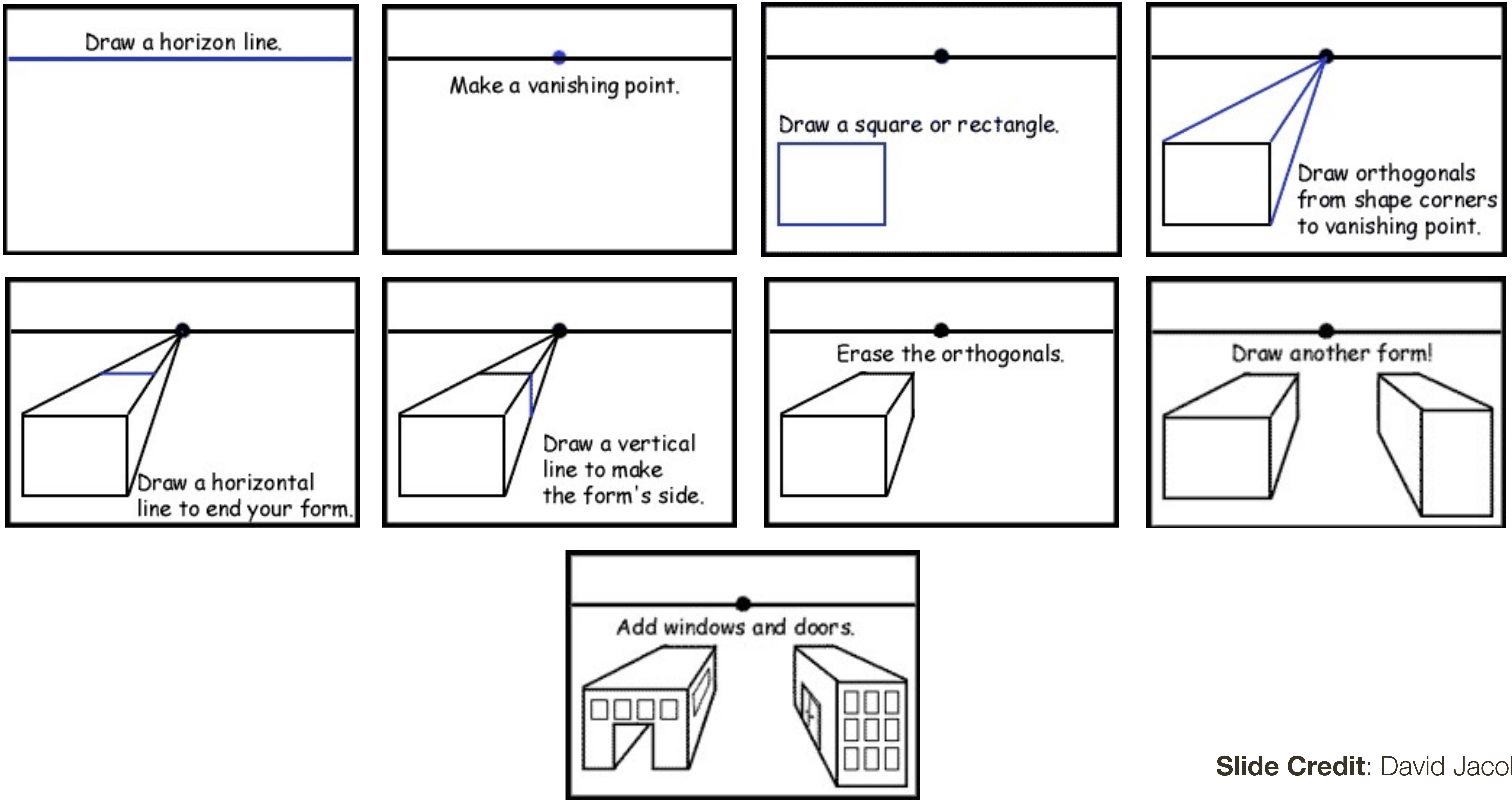
Vanishing Points





Slide Credit: David Jacobs

Vanishing Points



Slide Credit: David Jacobs

Vanishing Points

Each set of parallel lines meets at a different point

— the point is called the **vanishing point**

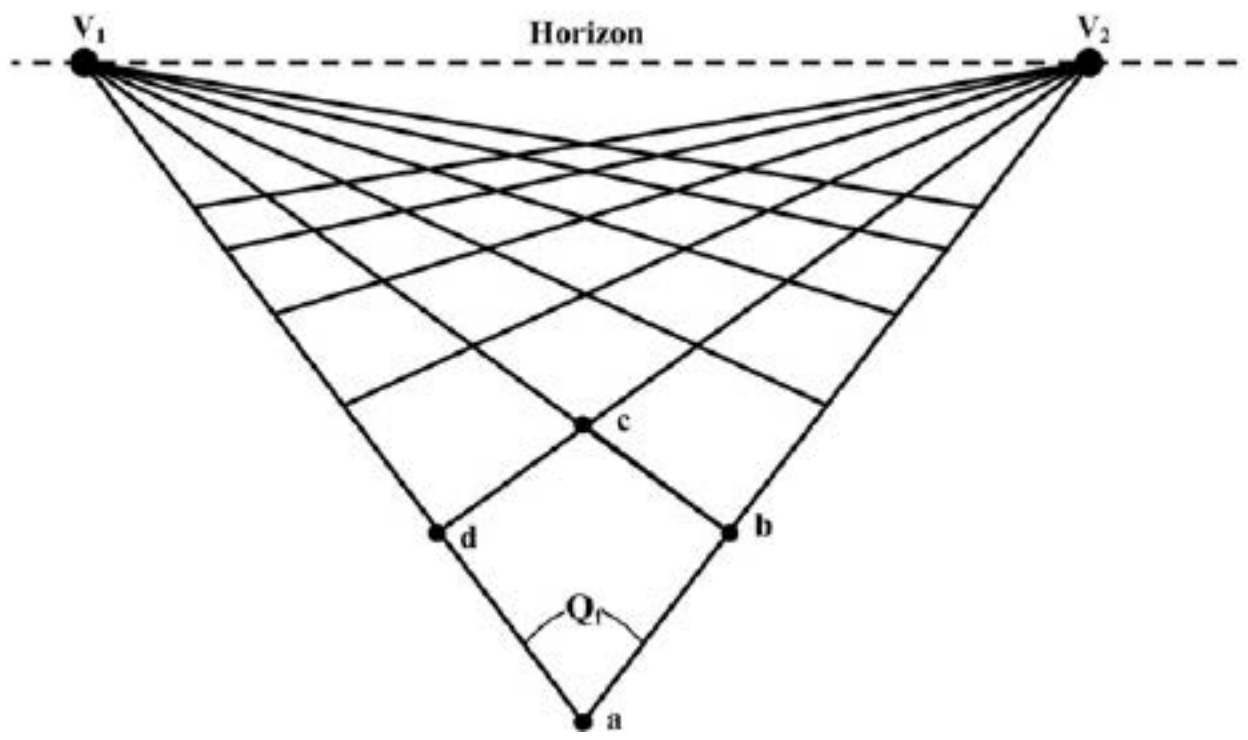
Sets of parallel lines on the same plane lead to **collinear** vanishing points

— the line is called a **horizon** for that plane

A good way to **spot fake images**

— scale and perspective do not work

— vanishing points behave badly

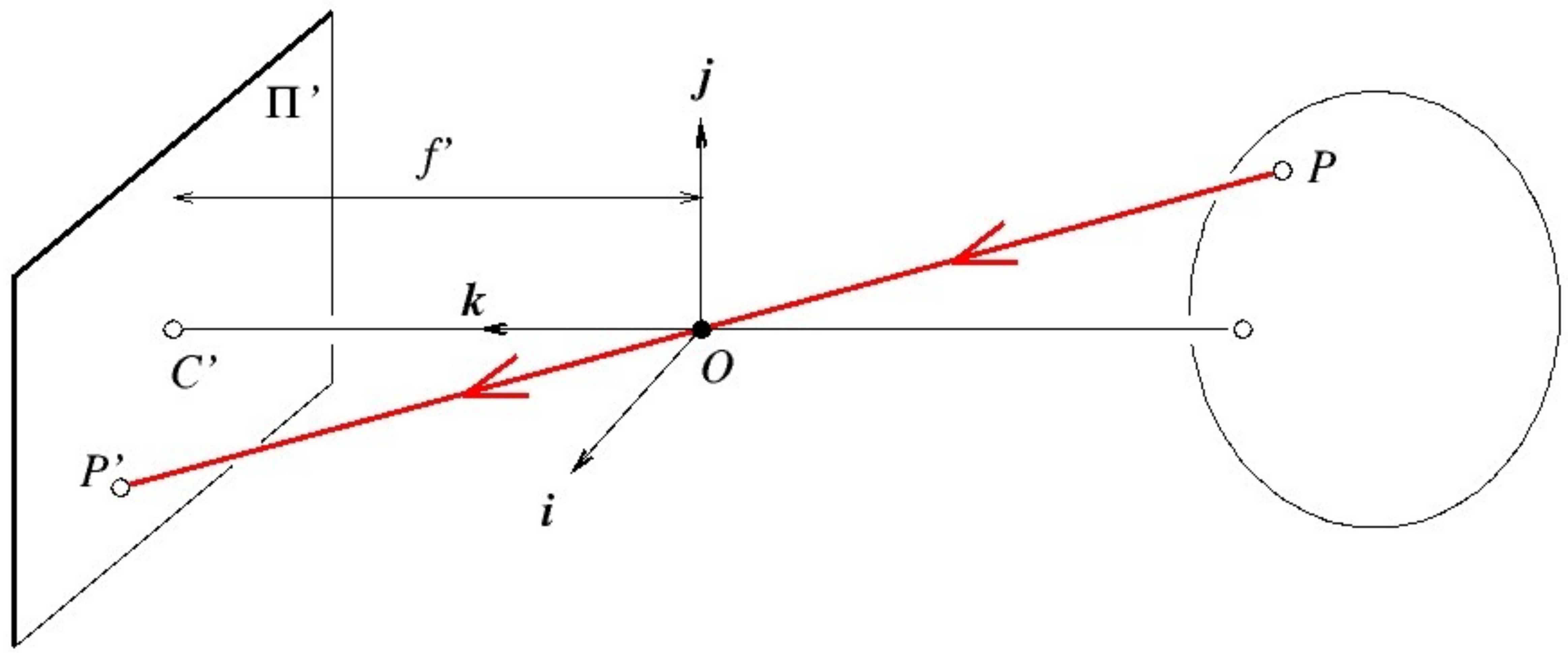


Spotting fake images with **Vanishing** Points

[Sarkar et al., 2023, Image from https://projective-geometry.github.io/ reproduced for educational purposes.]

Perspective Projection: Matrix Form

Camera Matrix



$$\mathbf{C} = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Forsyth & Ponce (1st ed.) Figure 1.4

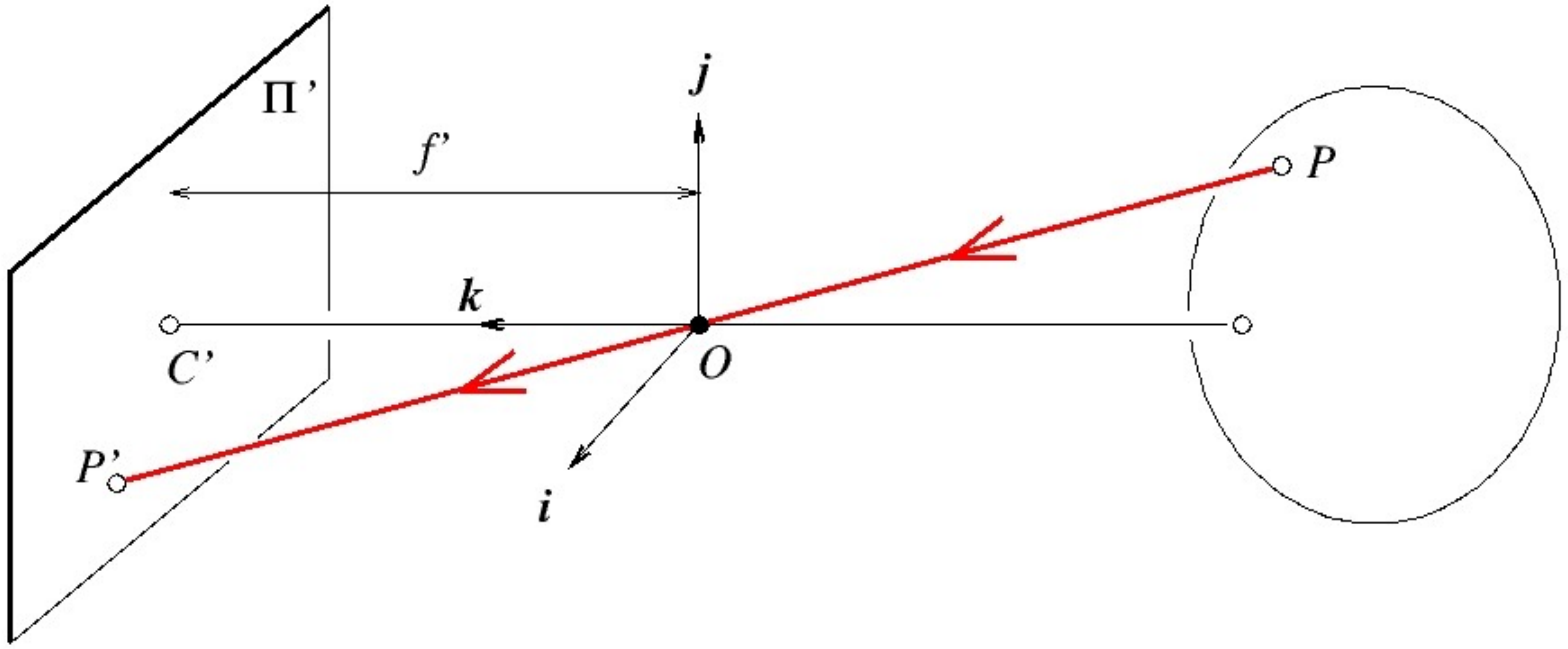
3D object point

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{sP' = \mathbf{C}P}$

(s is a scale factor)

2.4

Aside: Camera Matrix



Forsyth & Ponce (1st ed.) Figure 1.4

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3D object point

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Pixels are squared / lens is perfectly symmetric

Sensor and pinhole perfectly aligned

Coordinate system centered at the pinhole

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f'_x & 0 & 0 & 0 \\ 0 & f'_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

~~Pixels are squared / lens is perfectly symmetric~~

Sensor and pinhole perfectly aligned

Coordinate system centered at the pinhole

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f'_x & 0 & 0 & c_x \\ 0 & f'_y & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

~~Pixels are squared / lens is perfectly symmetric~~

~~Sensor and pinhole perfectly aligned~~

Coordinate system centered at the pinhole

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f'_x & 0 & 0 & c_x \\ 0 & f'_y & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbb{R}_{4\times 4}$$

~~Pixels are squared / lens is perfectly symmetric~~

~~Sensor and pinhole perfectly aligned~~

~~Coordinate system centered at the pinhole~~

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Aside: Camera Matrix

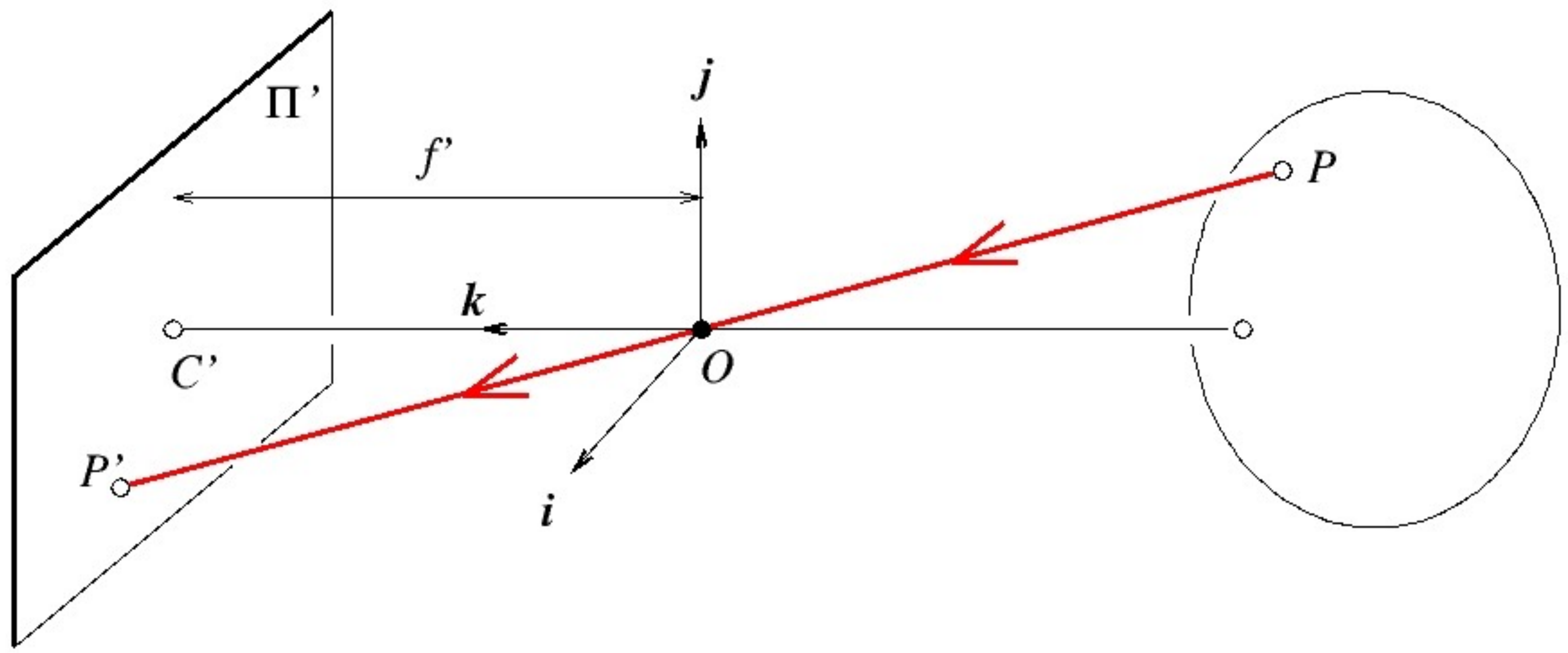
Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f'_x & 0 & 0 & c_x \\ 0 & f'_y & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbb{R}_{4\times 4}$$

Camera calibration is the process of estimating the parameters of the camera matrix based on a set of 3D-2D correspondences
(usually requires a pattern whose structure and size are known)

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ where $\boxed{P' = \mathbf{C}P}$

Perspective Projection



3D object point

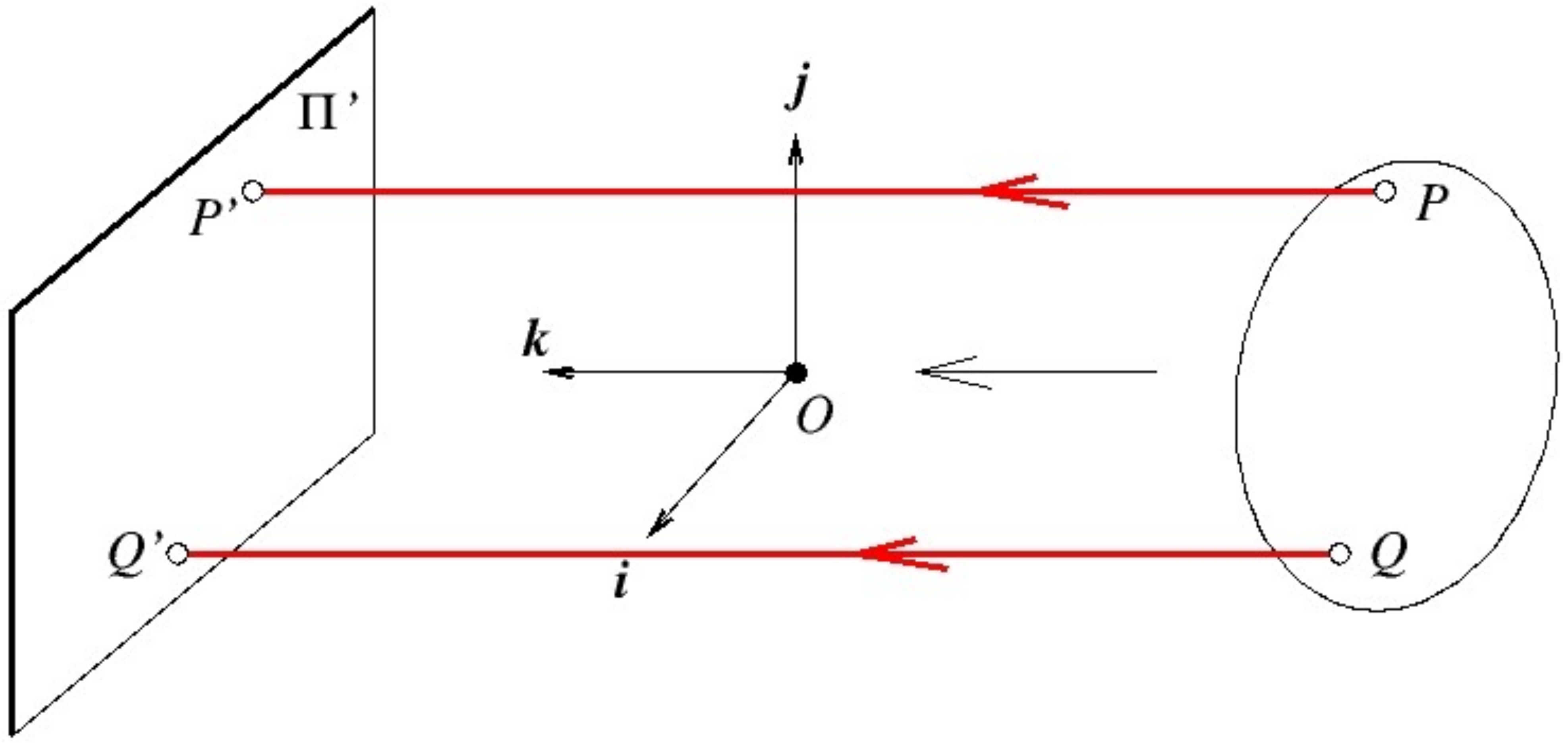
Forsyth & Ponce (1st ed.) Figure 1.4

$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ where

$$\boxed{\begin{aligned} x' &= f' \frac{x}{z} \\ y' &= f' \frac{y}{z} \end{aligned}}$$

Note: this assumes world coordinate frame at the optical center (pinhole) and aligned with the image plane, image coordinate frame aligned with the camera coordinate frame

Orthographic Projection



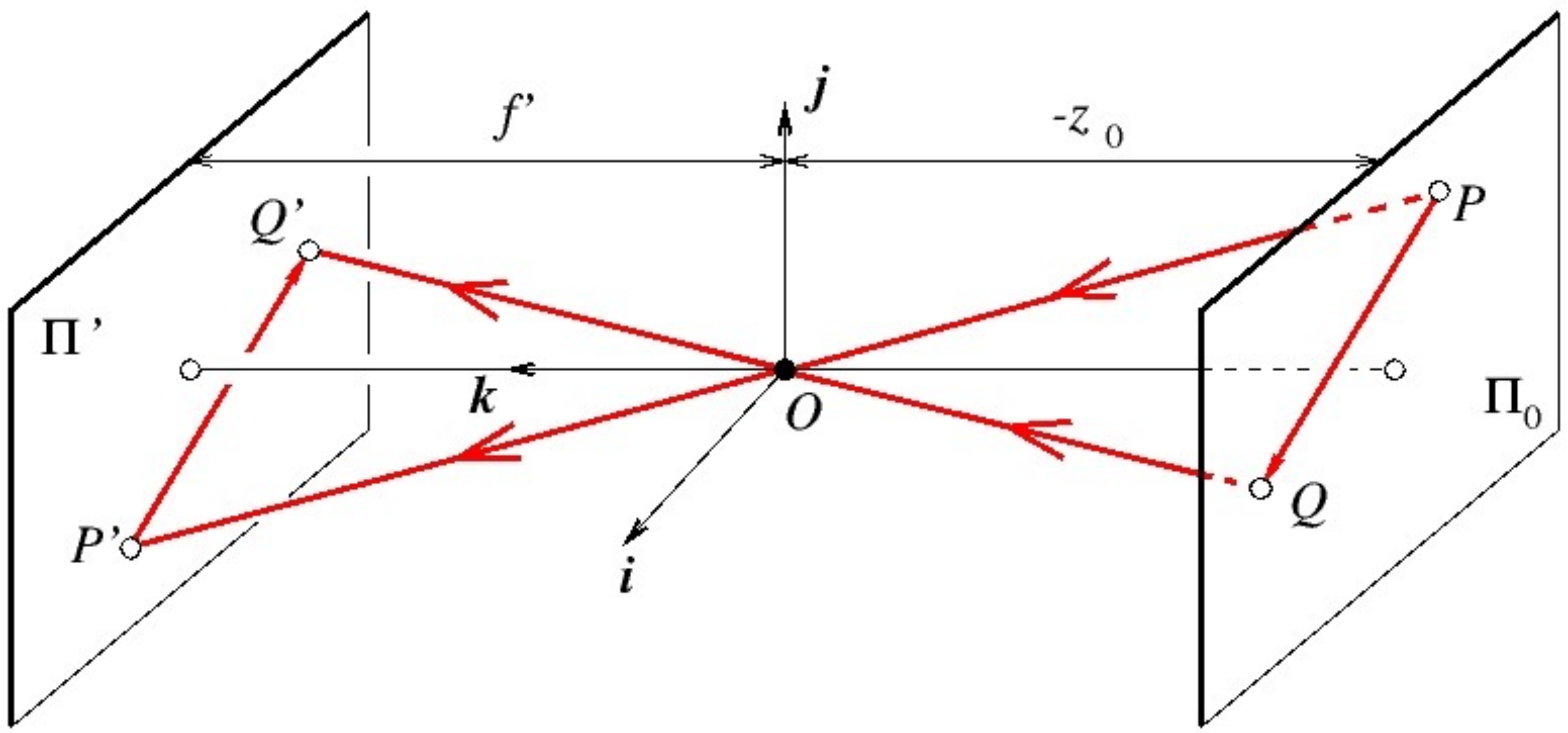
Forsyth & Ponce (1st ed.) Figure 1.6

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$

where $\boxed{\begin{array}{rcl} x' & = & x \\ y' & = & y \end{array}}$

Weak Perspective

3.1



Forsyth & Ponce (1st ed.) Figure 1.5

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ in Π_0 projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$

where $\boxed{\begin{array}{rcl} x' & = & mx \\ y' & = & my \end{array}}$ and $m = \frac{f'}{z_0}$

Summary of **Projection Equations**

3D object point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ projects to 2D image point $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ where

Perspective	$x' = f' \frac{x}{z}$ $y' = f' \frac{y}{z}$	
Weak Perspective	$x' = m\,x$ $y' = m\,y$	$m = \frac{f'}{z_0}$
Orthographic	$x' = x$ $y' = y$	

Projection Models: Pros and Cons

Weak perspective (including orthographic) has simpler mathematics

— accurate when object is small and/or distant

— useful for recognition

Perspective is more accurate for real scenes

When **maximum accuracy** is required, it is necessary to model additional details of a particular camera

— use perspective projection with additional parameters (e.g., lens distortion)

Projection Illusion

Our brains also know this perspective model very well!

Why **Not** a Pinhole Camera?

– If pinhole is **too big** then many directions are averaged, blurring the image

– If pinhole is **too small** then diffraction becomes a factor, also blurring the image

– Generally, pinhole cameras are **dark**, because only a very small set of rays from a particular scene point hits the image plane

– Pinhole cameras are **slow**, because only a very small amount of light from a particular scene point hits the image plane per unit time

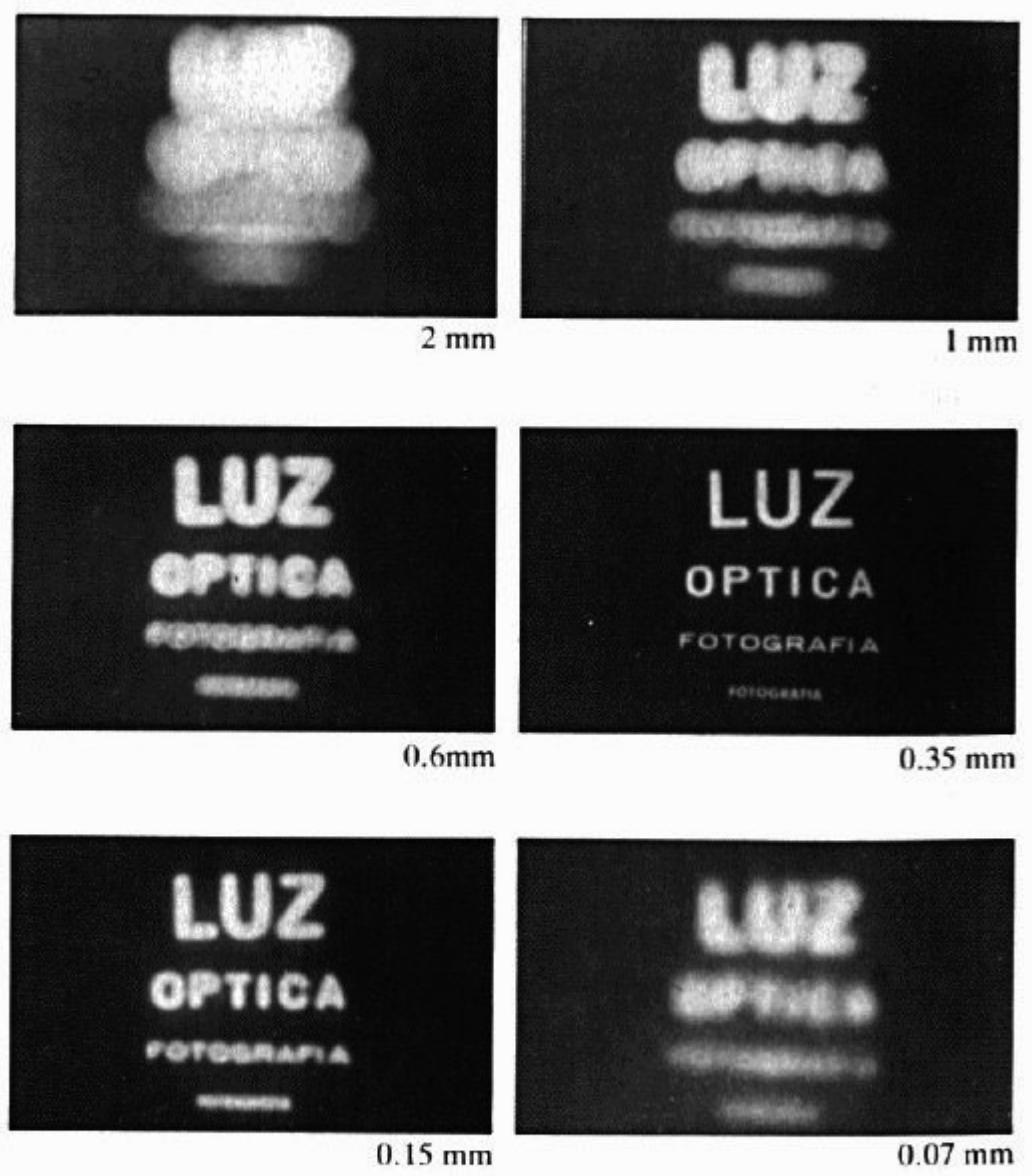
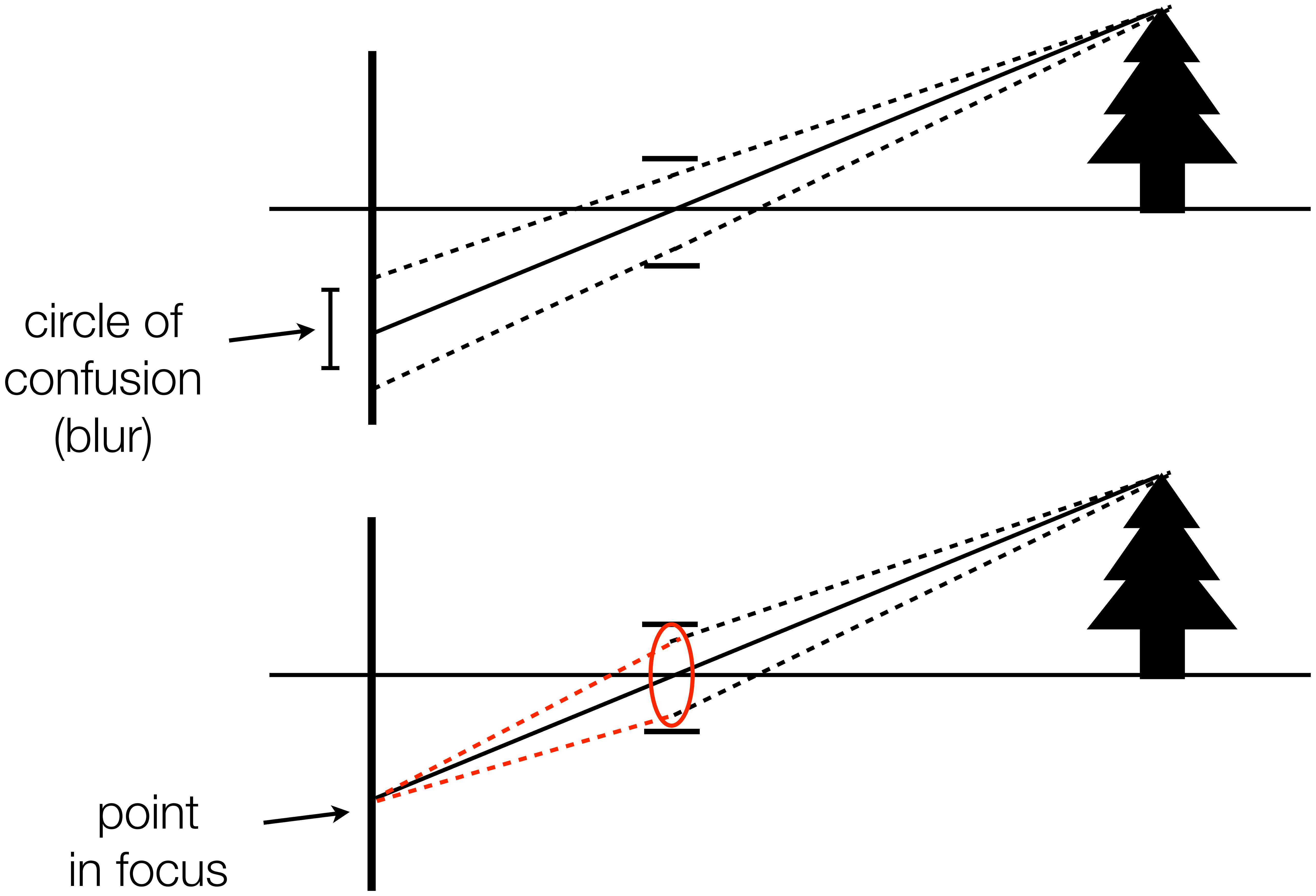


Image Credit: Credit: E. Hecht. "Optics," Addison-Wesley, 1987

Reason for **Lenses**

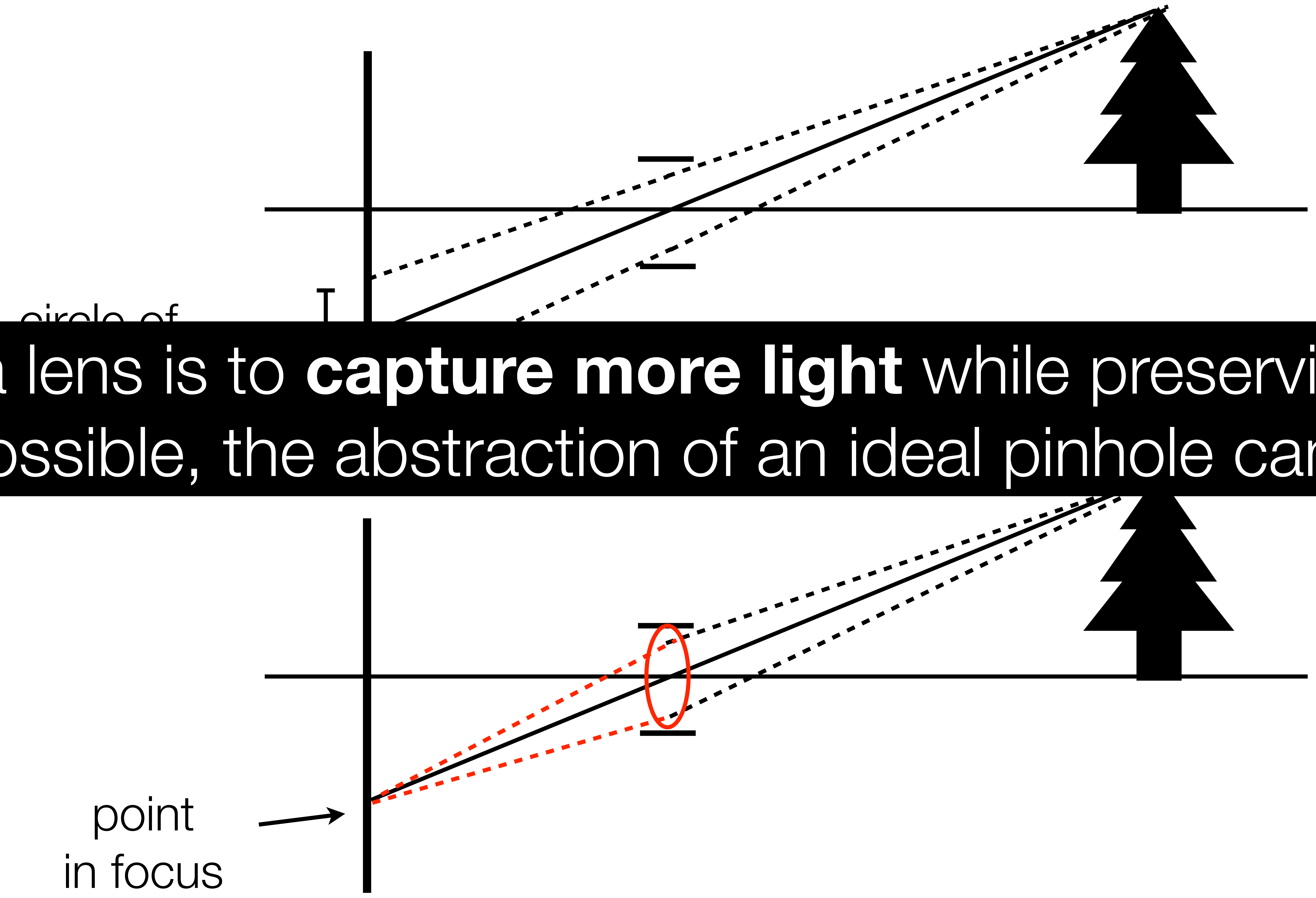
A real camera must have a finite aperture to get enough light, but this causes blur in the image



Solution: use a **lens** to focus light onto the image plane

Reason for **Lenses**

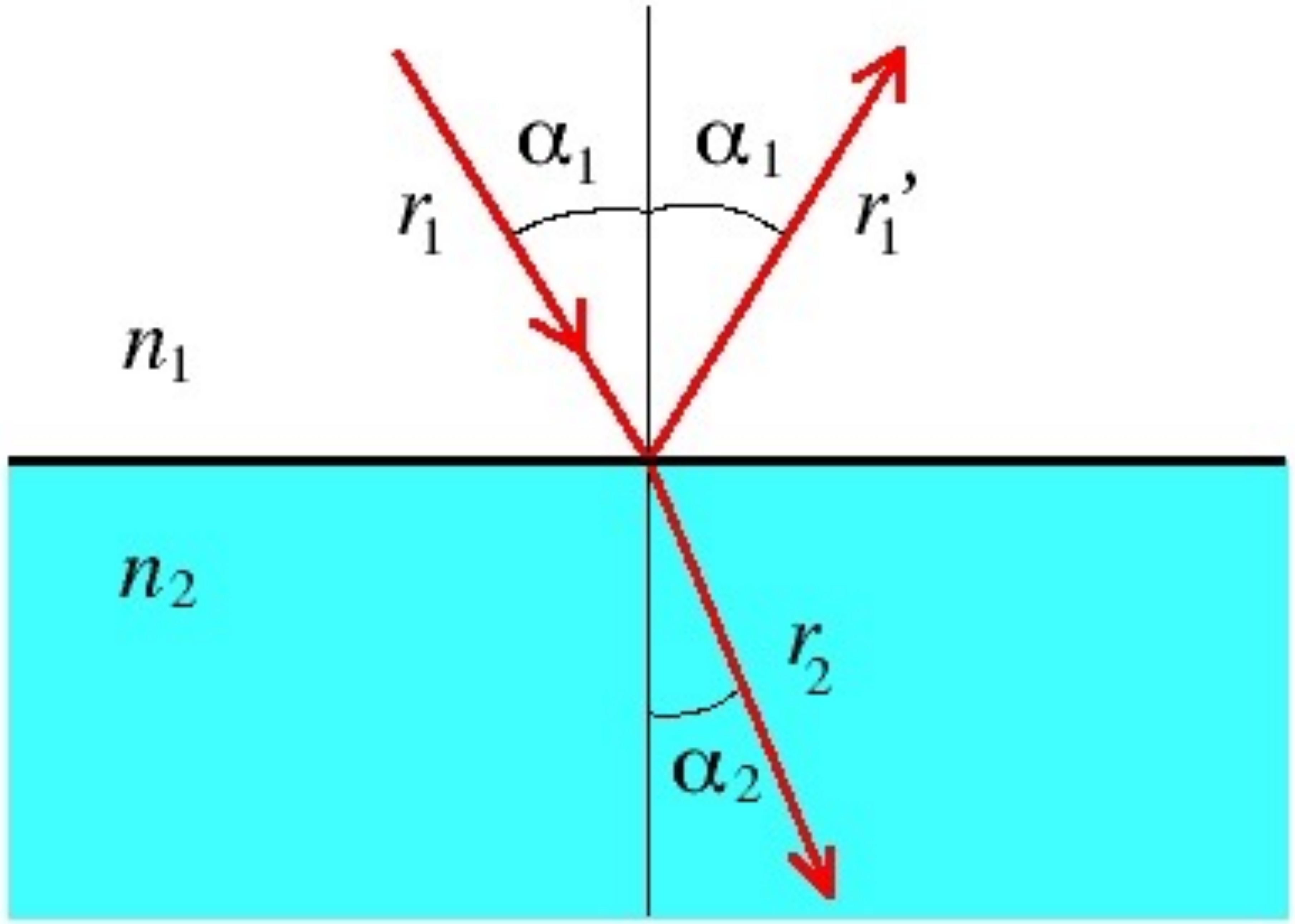
A real camera must have a finite aperture to get enough light, but this causes blur in the image



The role of a lens is to **capture more light** while preserving, as much as possible, the abstraction of an ideal pinhole camera.

Solution: use a **lens** to focus light onto the image plane

Snell's Law



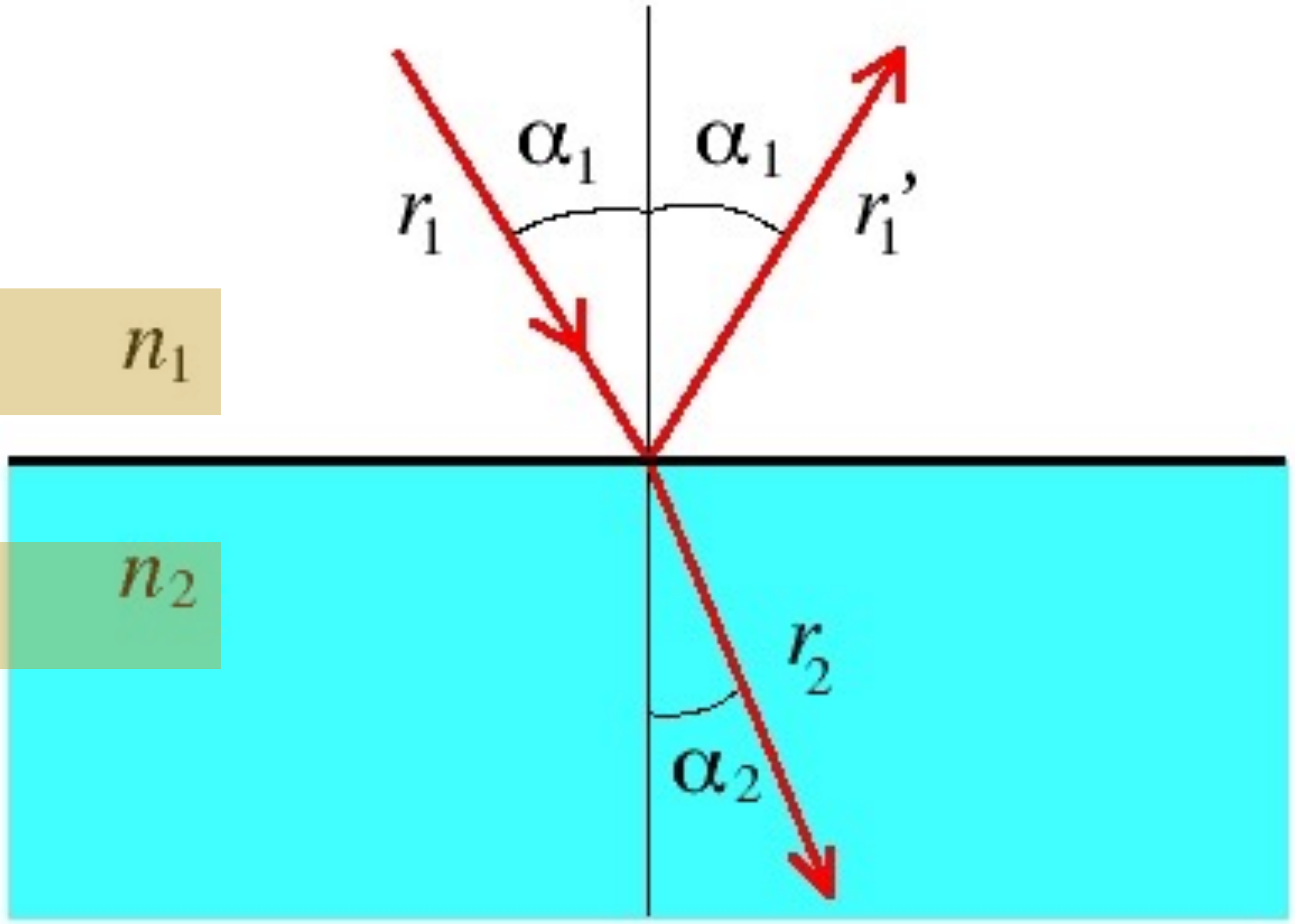
$$\boxed{n_1 \sin \alpha_1 = n_2 \sin \alpha_2}$$

Snell's Law

n_1

Index of **refraction**

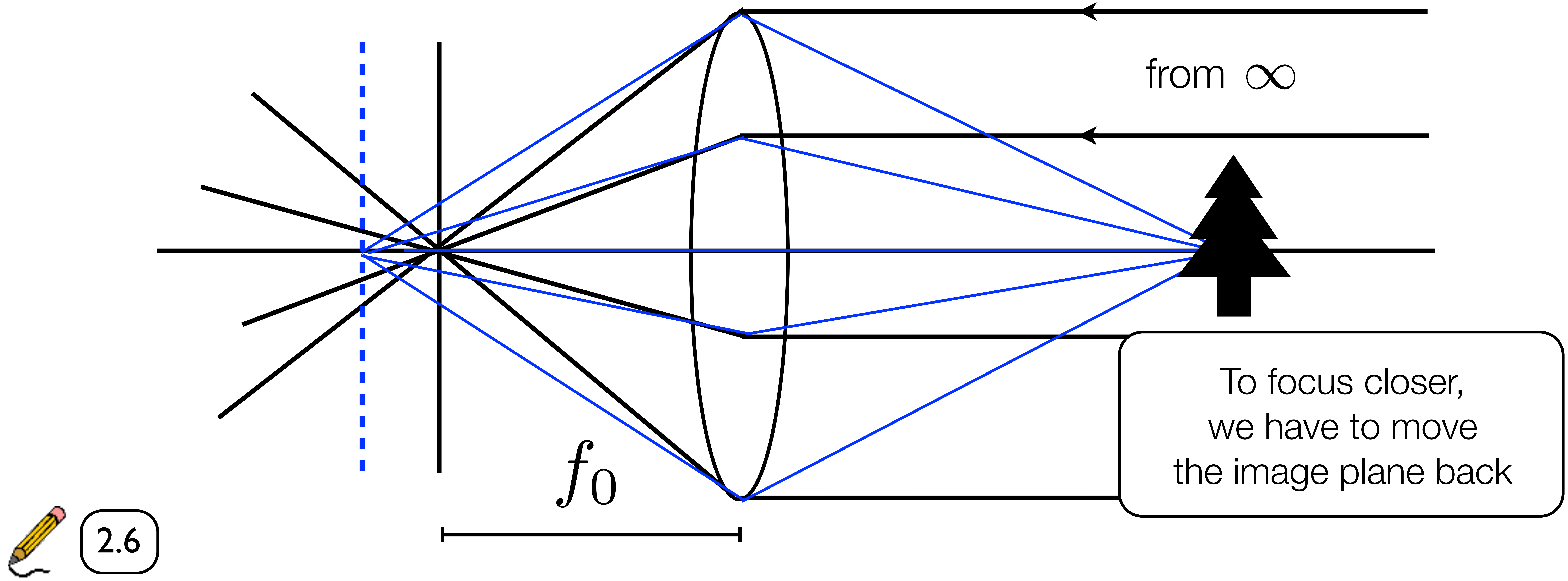
n_2



$$\boxed{n_1 \sin \alpha_1 = n_2 \sin \alpha_2}$$

Lens Basics

- A lens focuses rays from infinity at the focal length of the lens
- Points passing through the centre of the lens are not bent



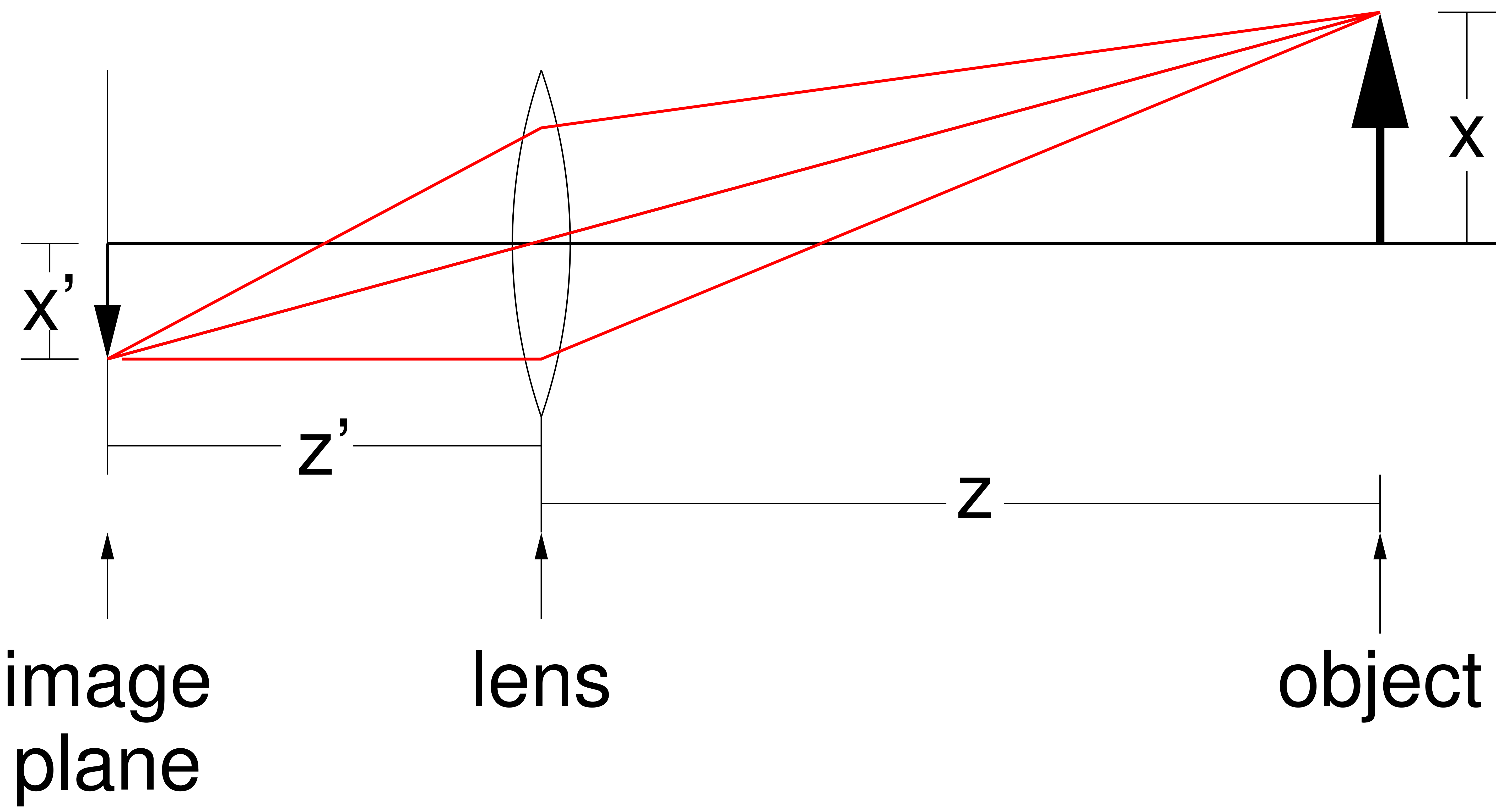
- We can use these 2 properties to find the **thin** lens equation

Lens Basics

- A 50mm lens is focussed at infinity. It now moves to focus on something 5m away. How far does the lens move?

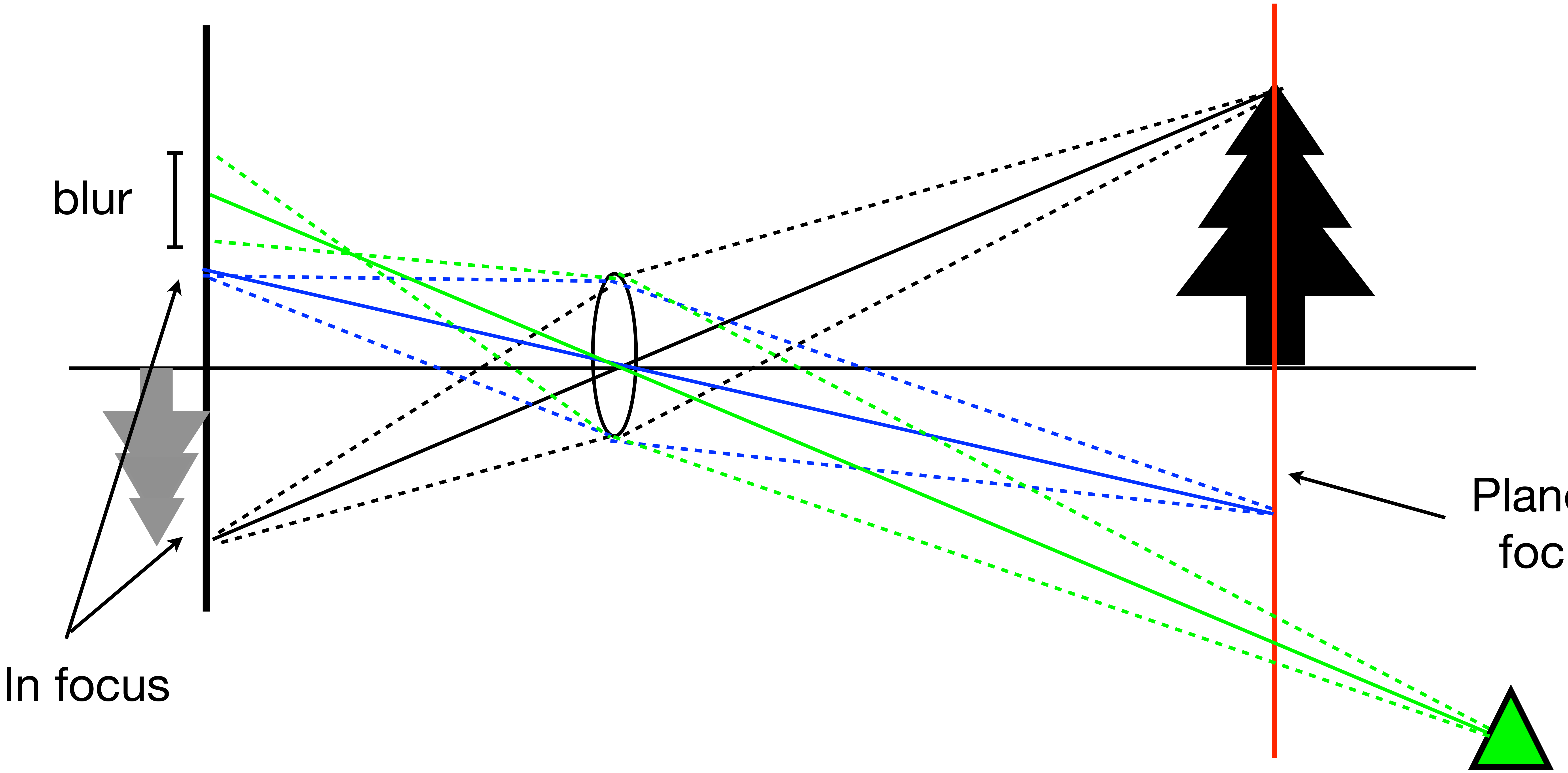
2.6

Pinhole Model **with Lens**



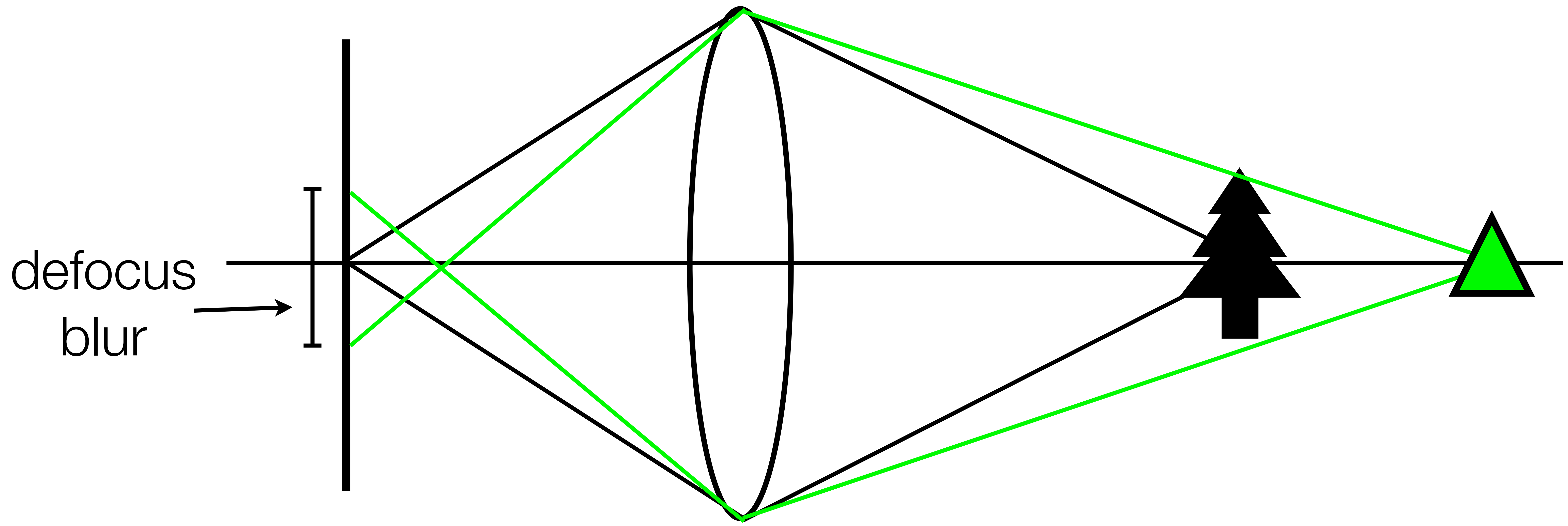
Lens Basics

- Lenses focus all rays from a plane in the world

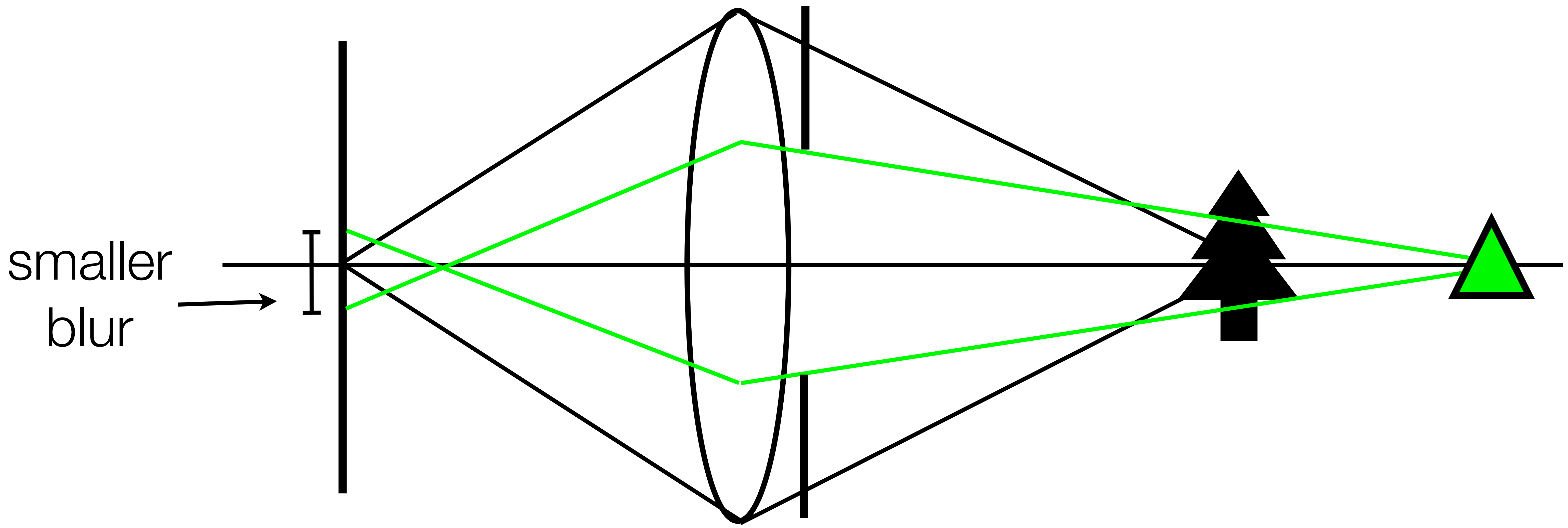


- Objects off the plane are blurred depending on distance

Effect of Aperture Size



Smaller aperture ⇒ smaller blur, larger **depth of field**



Depth of Field

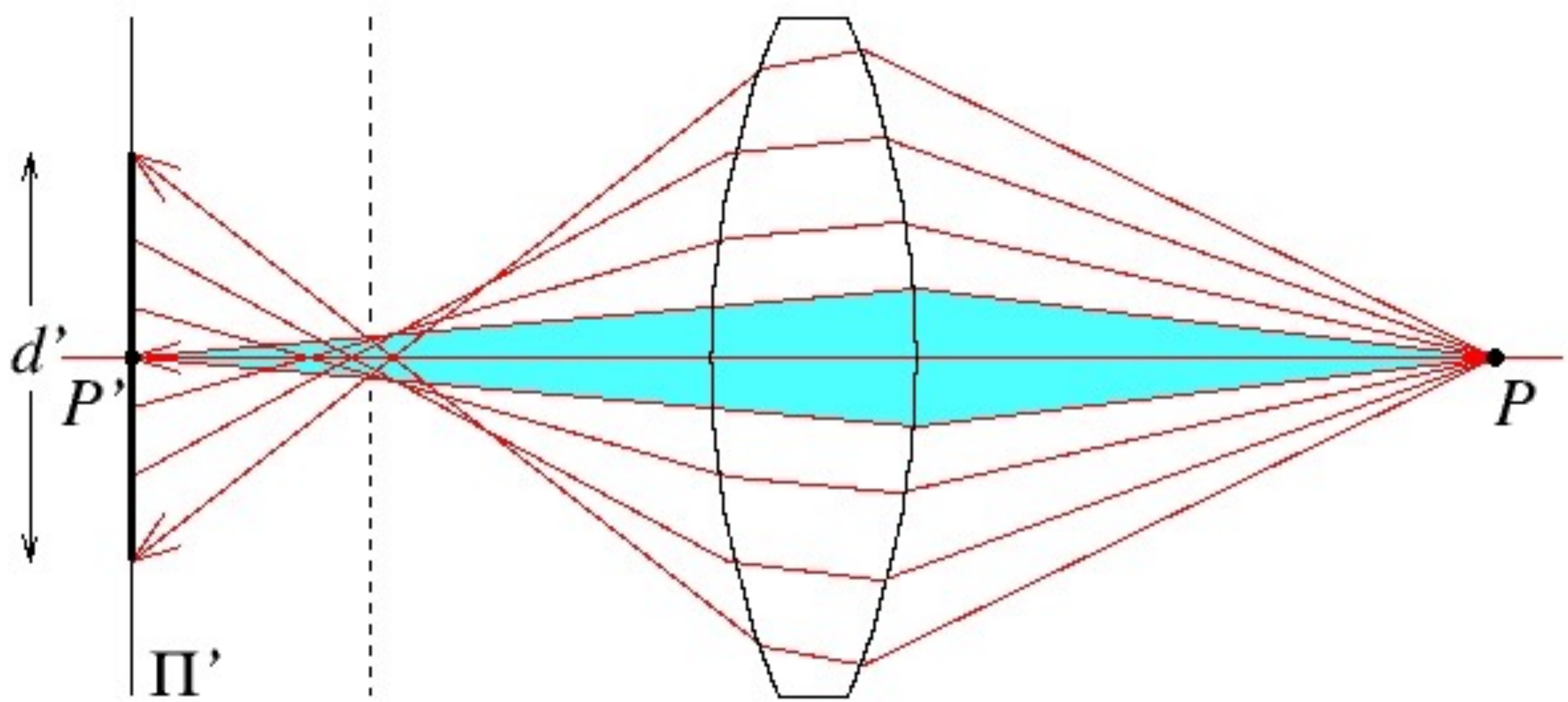
- Photographers use large apertures to give small depth of field

Aperture size = f/N, ⇒ large N = small aperture

Real Lenses

- Real Lenses have multiple stages of positive and negative elements with differing refractive indices
- This can help deal with issues such as chromatic aberration (different colours bent by different amounts), vignetting (light fall off at image edge) and sharp imaging across the zoom range

Spherical **Aberration**



Forsyth & Ponce (1st ed.) Figure 1.12a

Spherical **Aberration**

Un-aberrated image

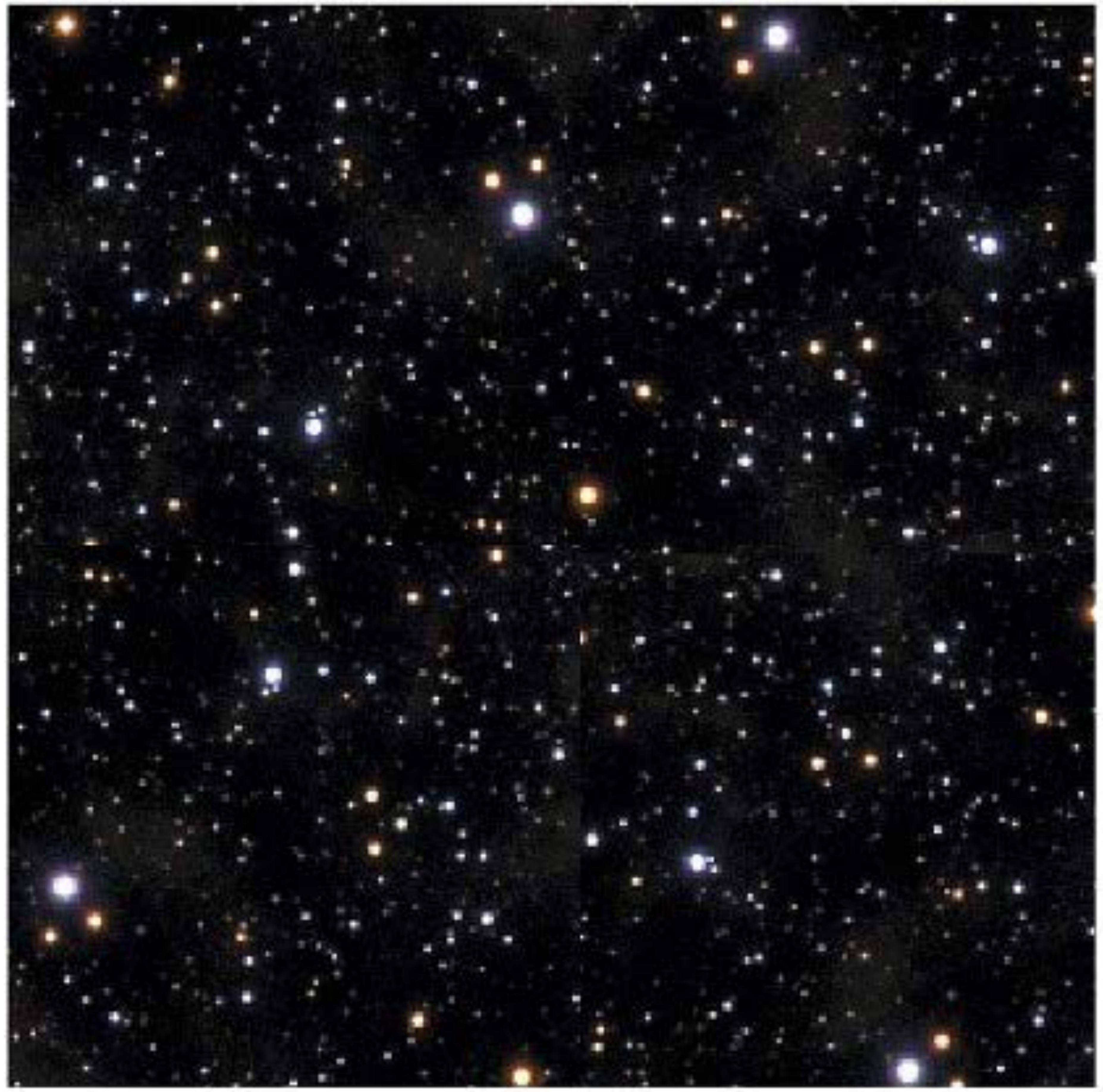
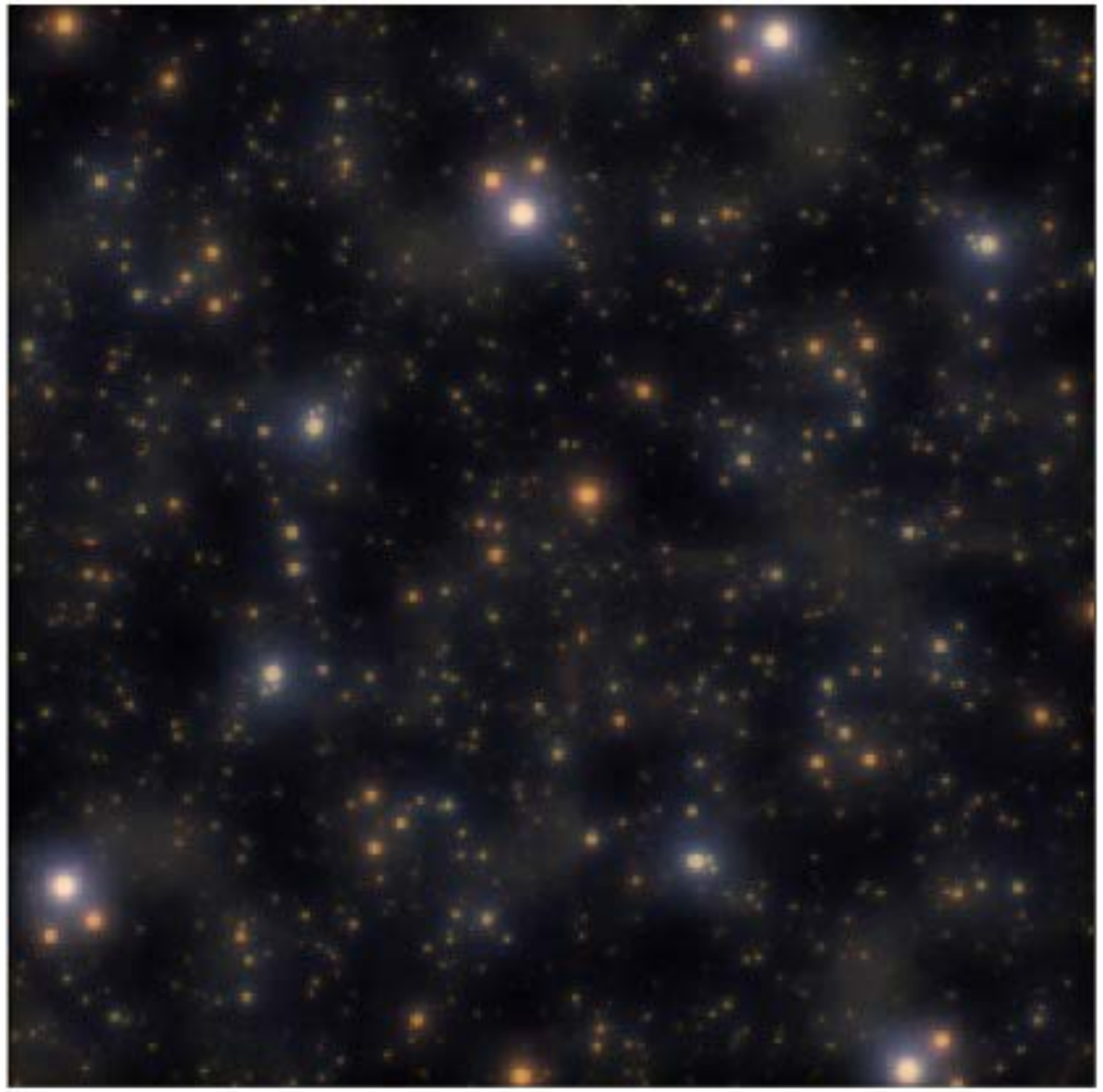
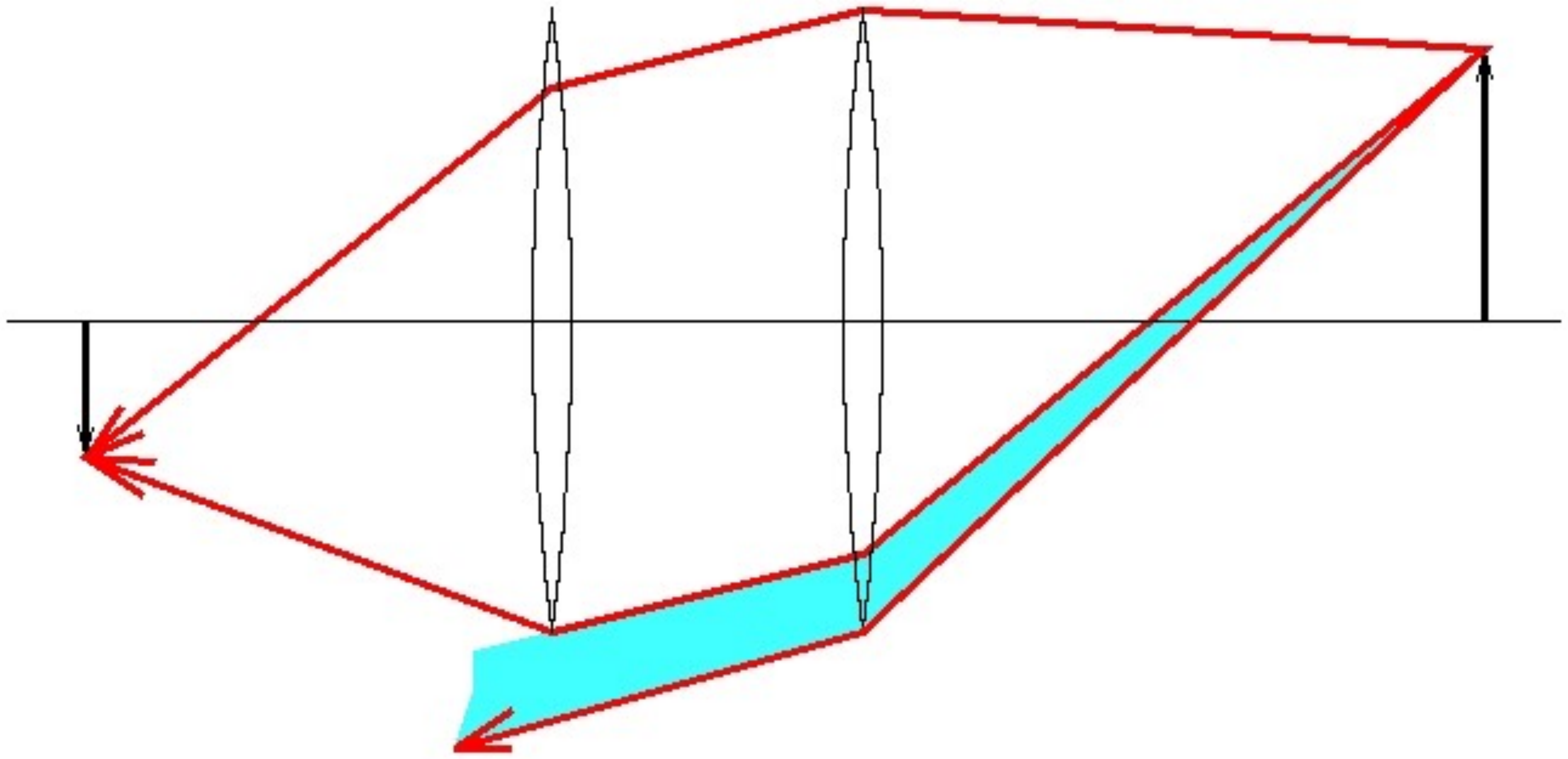


Image from lens with Spherical Aberration



Vignetting

Vignetting in a two-lens system



Forsyth & Ponce (2nd ed.) Figure 1.12

The shaded part of the beam **never reaches** the second lens

Vignetting

Image Credit: Cambridge in Colour

Chromatic **Aberration**

— Index of **refraction depends on wavelength**, λ, of light

— Light of different colours follows different paths

— Therefore, not all colours can be in equal focus

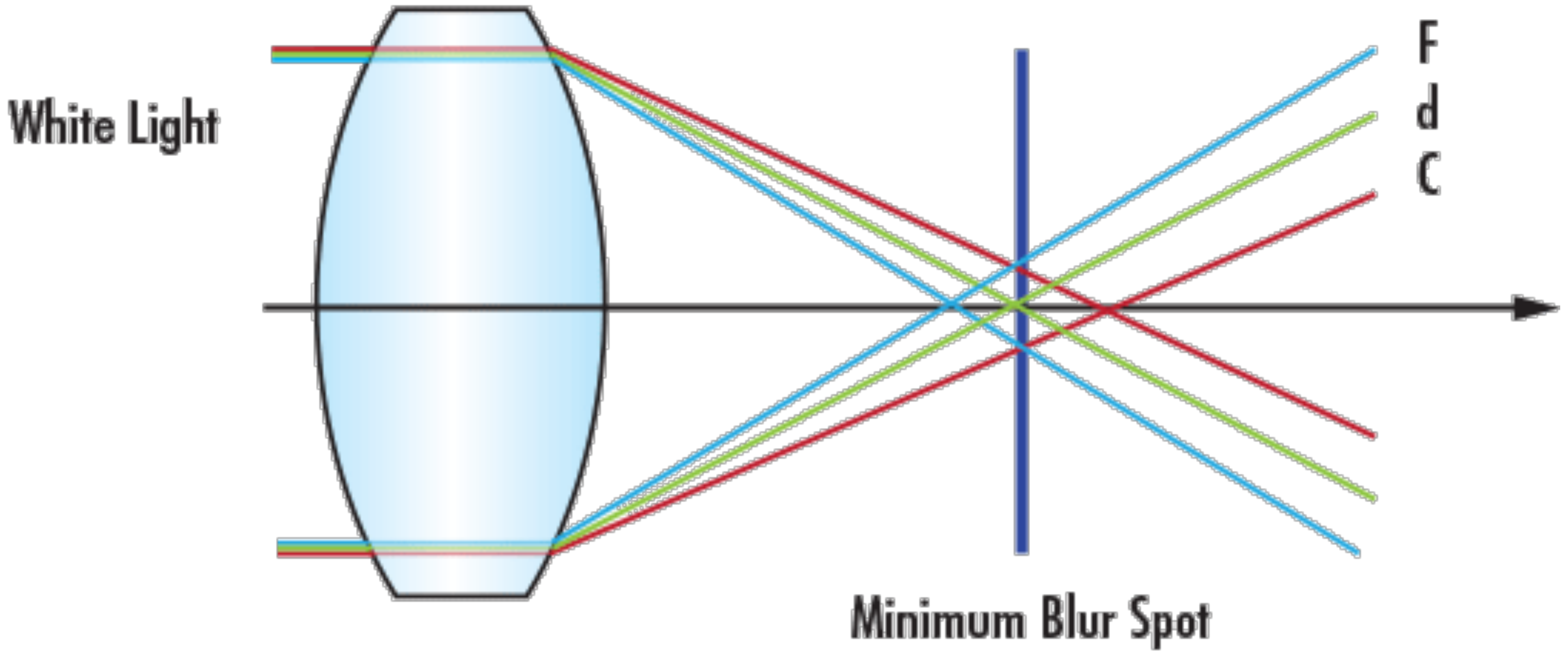
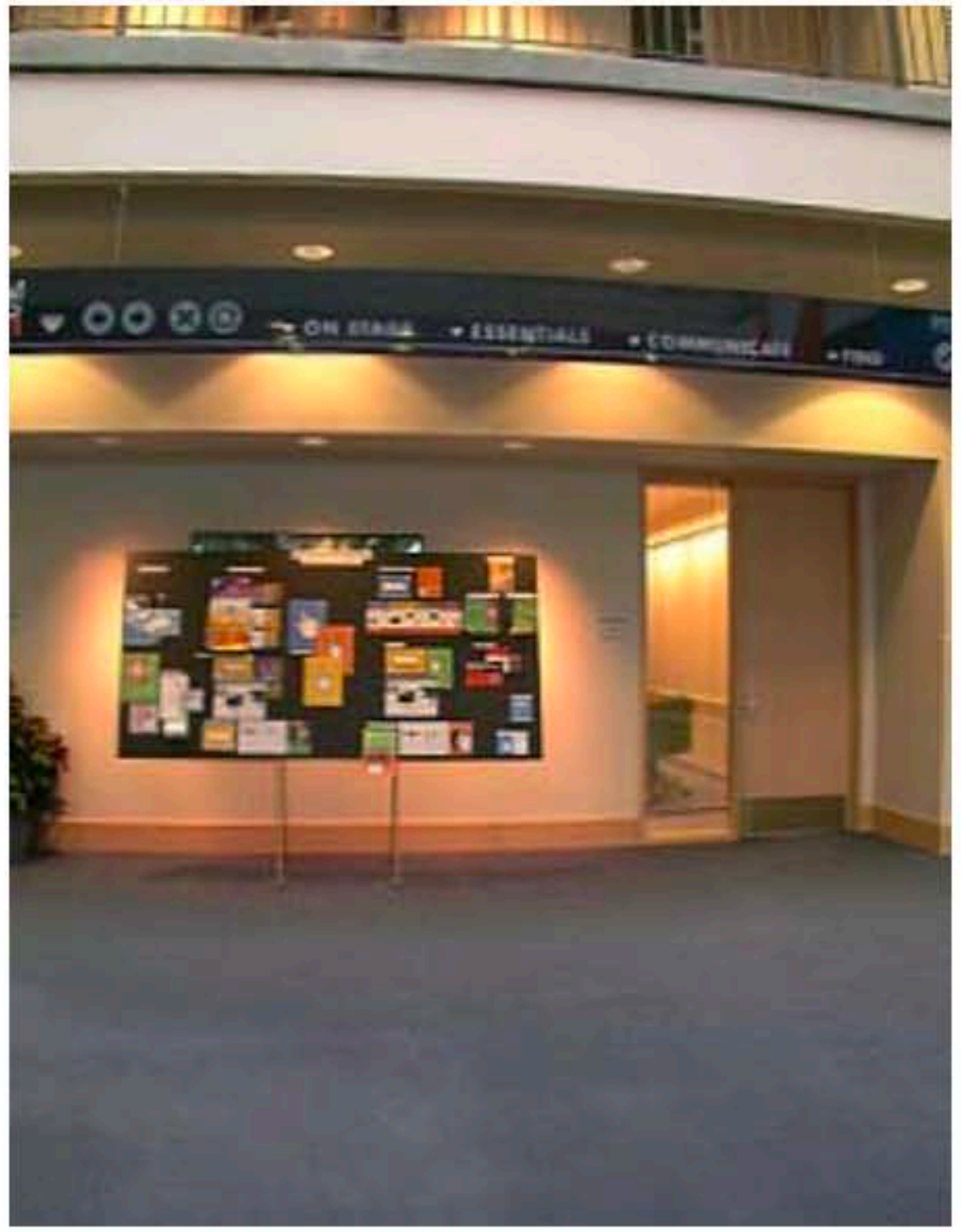
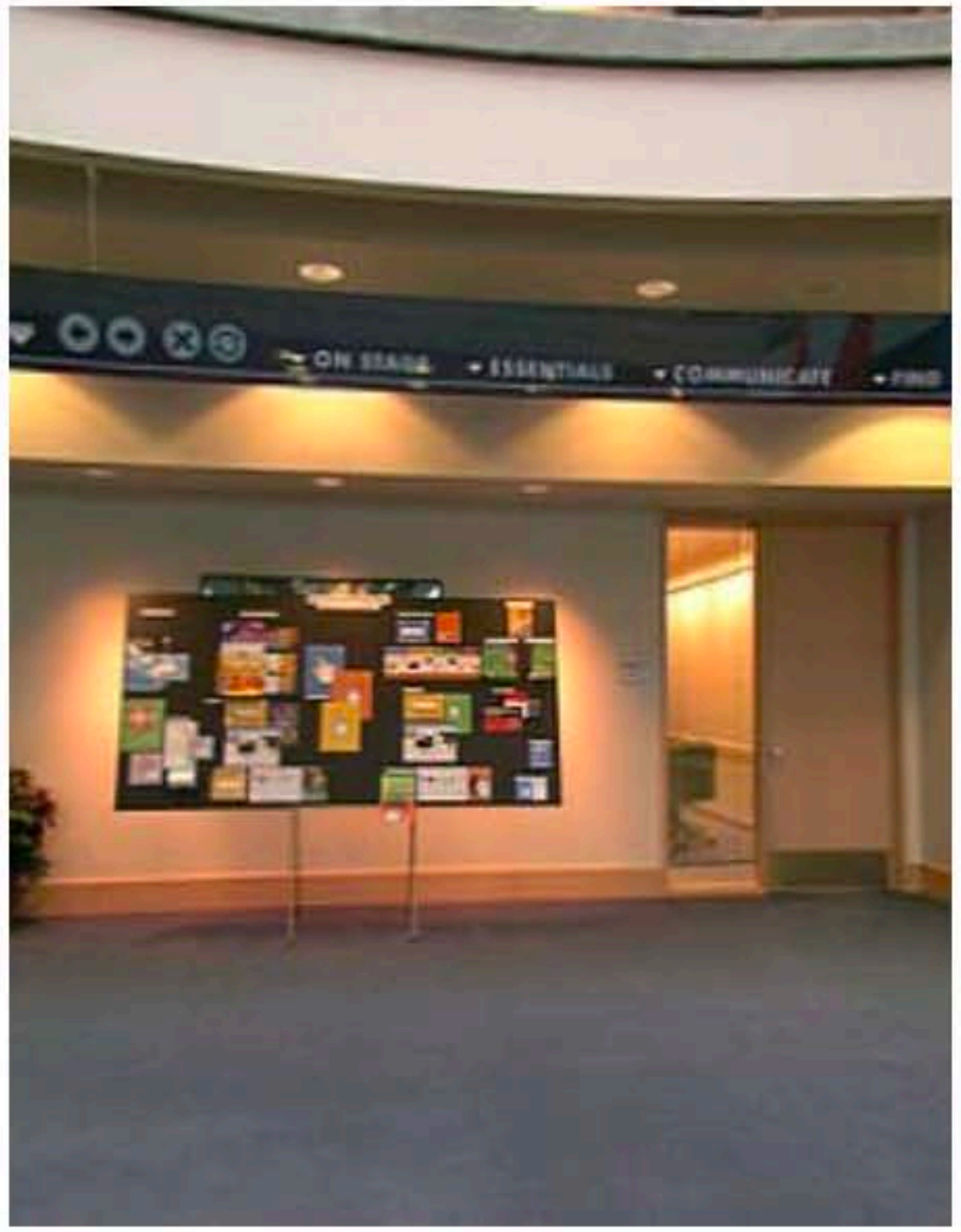


Image Credit: Trevor Darrell

Lens **Distortion**





Fish-eye Lens

Szeliski (1st ed.) Figure 2.13

Lines in the world are no longer lines on the image, they are curves!

Other (Possibly Significant) **Lens Effects**

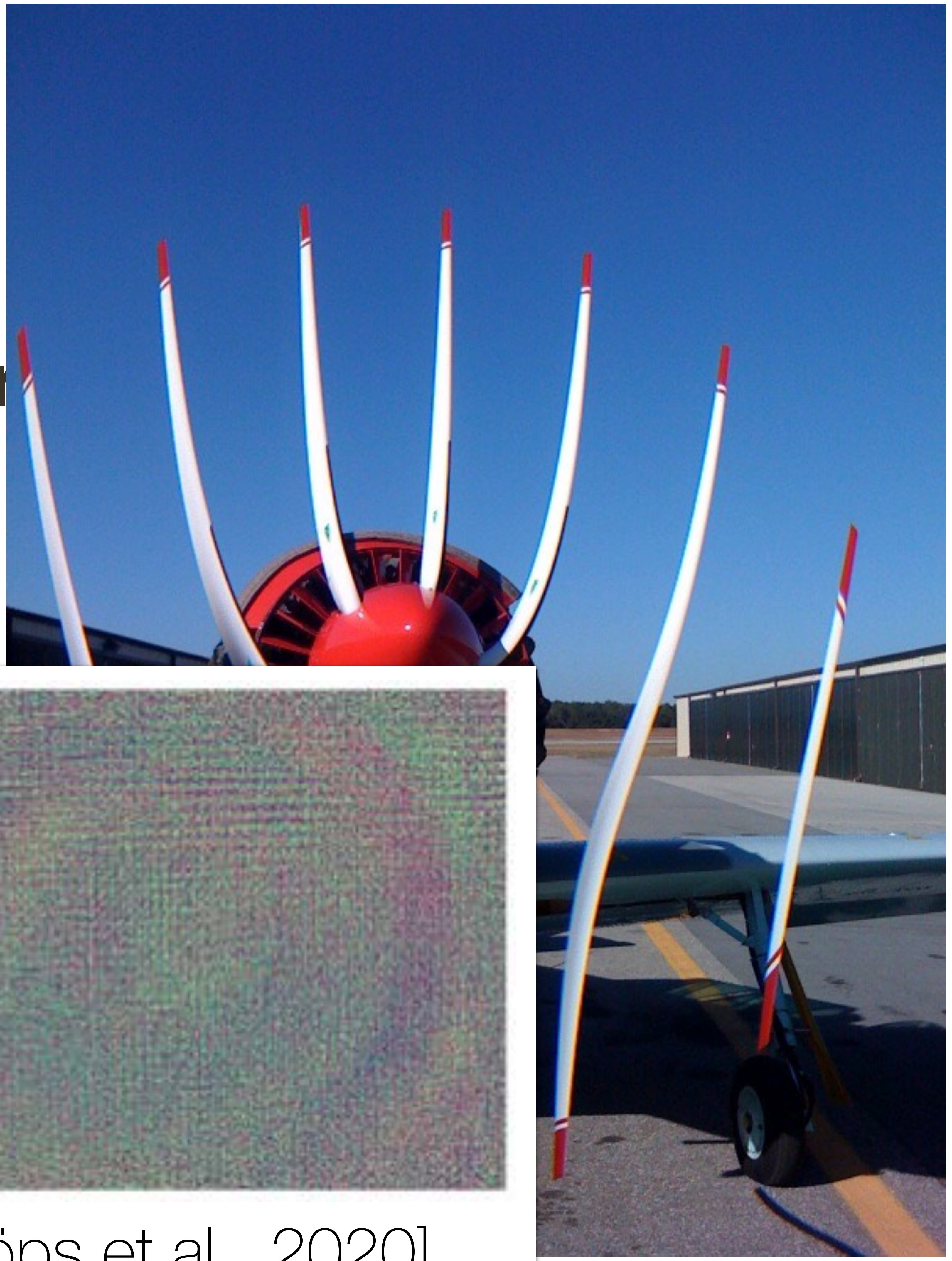
Scattering at the lens surface

— Some light is reflected at each lens surface

There are other **geometric phenomena/distor**

— pincushion distortion

— barrel distortion



nragsdale/3192314056/

Parametric calibration errors

[Schöps et al., 2020]

Image from [Schöps et al., 2019]. Reproduced for educational purposes.

Lecture **Summary**

— We discussed a “physics-based” approach to image formation. Basic abstraction is the **pinhole camera**.

— **Lenses overcome limitations** of the pinhole model while trying to preserve it as a useful abstraction

— Projection equations: **perspective**, weak perspective, orthographic

— Thin lens equation

— Some “aberrations and **distortions**” persist (e.g. spherical aberration, vignetting)