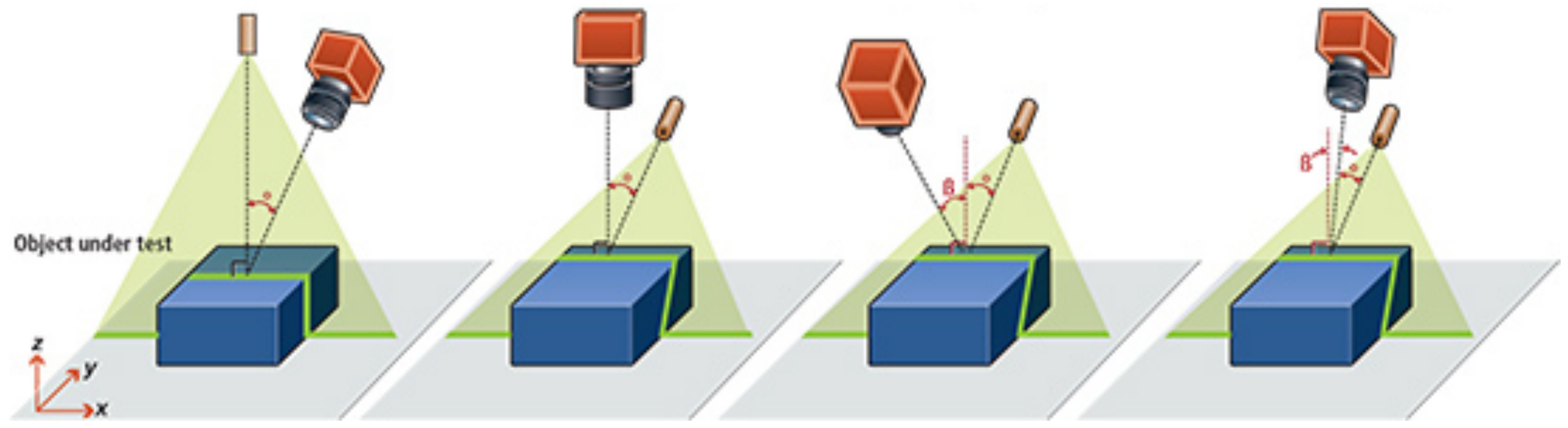
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 2: Image Formation

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today (**September 11, 2020**)

Topics:

— Image Formation
— Cameras and Lenses
— Projection

Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 1.1.1 — 1.1.3
— **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.1, 4.5

Reminders:

— Complete **Assignment 0** (ungraded) by Wednsday, **September 16**
— Google **Colab tutorials** next week
— **TA and Office** hours are posted and will start on Monday, **September 14**

Today’s “fun” Example

Today's "fun" Example

Photo credit: reddit user Liammm

Today's "fun" Example: **Eye Sink Illusion**

Pereidolia

Photo credit: reddit user Liammm

Today's "fun" Example: **Eye Sink Illusion**

"Tried taking a picture of a sink draining, wound up with a picture of an eye instead"

Photo credit: reddit user Liammm

Lecture 1: Re-cap

Types of computer vision **problems**:

— Computing properties of the 3D world from visual data (***measurement***)
— Recognition of objects and scenes (***perception and interpretation***)
— Search and interact with visual data (***search and organization***)
— Manipulation or creation of image or video content (***visual imagination***)

Computer vision **challenges**:

— Fundamentally **ill-posed**
— Enormous **computation** and **scale**
— Lack of fundamental understanding of how **human perception** works

Lecture 1: Re-cap

Computer vision technologies have moved **from research labs into commercial products and services**. Examples cited include:

— broadcast television sports

— electronic games (Microsoft Kinect)

— biometrics

— image search

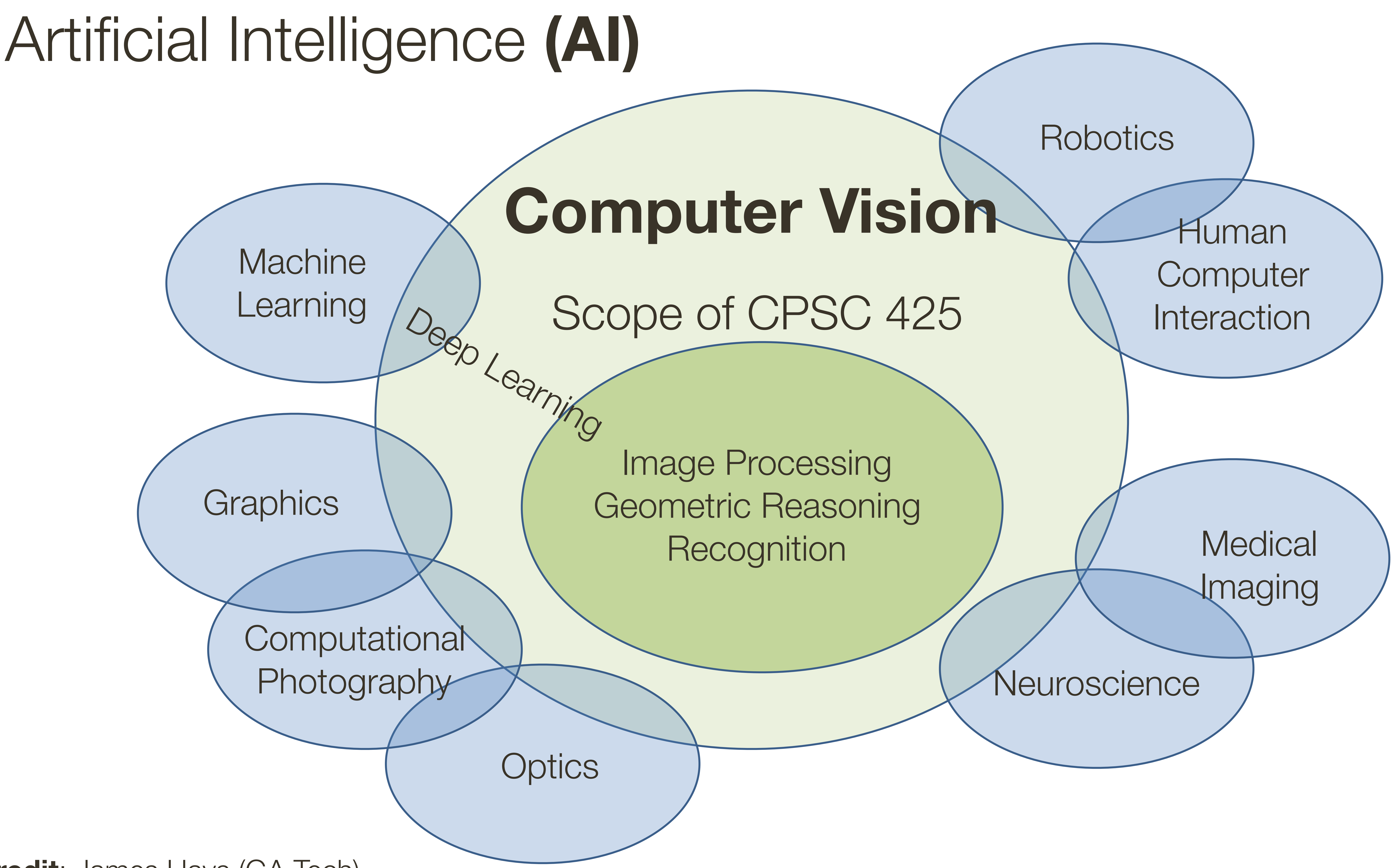
— visual special effects

— medical imaging

— robotics

… many others

Related Disciplines



Slide Credit: James Hays (GA Tech)

Related Disciplines: Vision and Graphics

Slide Credit: Kristen Grauman (UT Austin)

Related Disciplines: Vision and Graphics

Model

Slide Credit: Kristen Grauman (UT Austin)

Related Disciplines: Vision and Graphics

Model

Slide Credit: Kristen Grauman (UT Austin)

Related Disciplines: Vision and Graphics

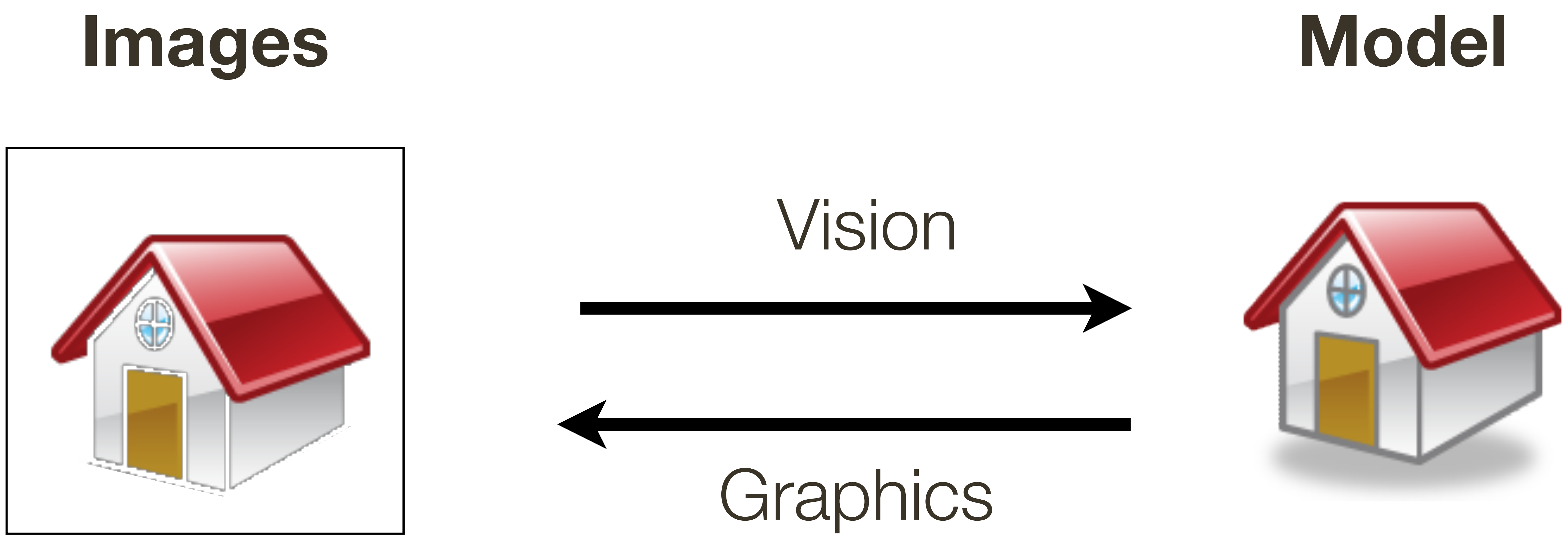
Images

Model

Graphics

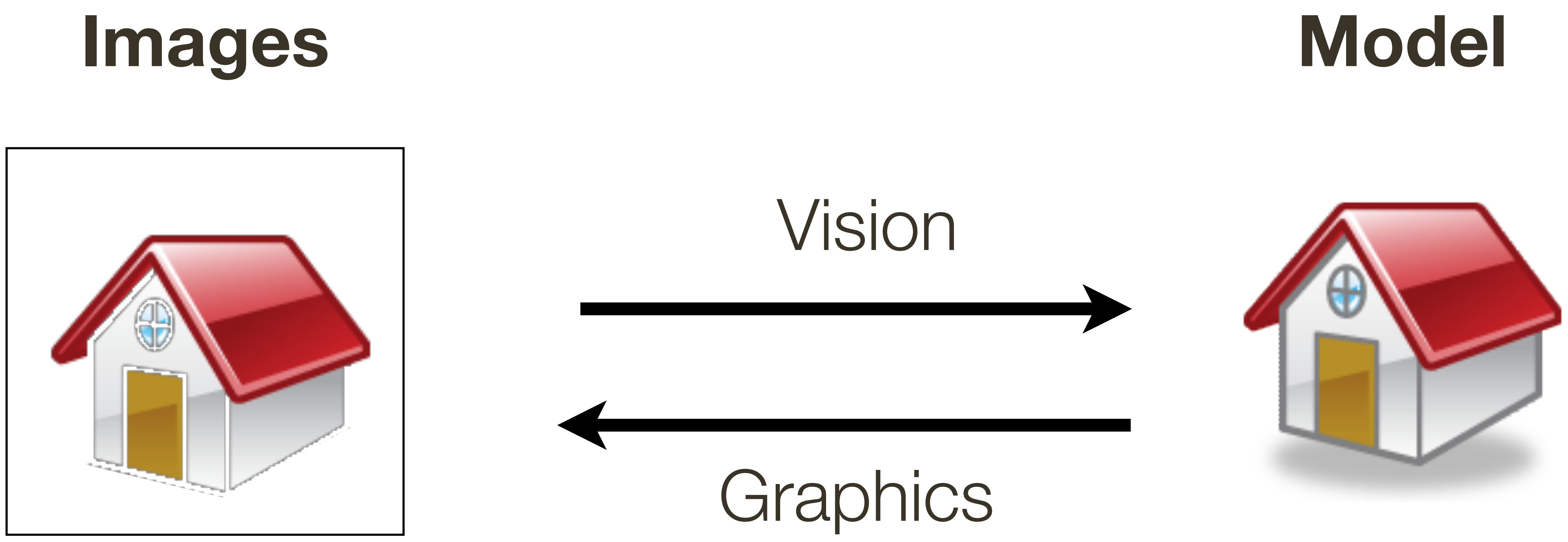
Slide Credit: Kristen Grauman (UT Austin)

Related Disciplines: Vision and Graphics



Slide Credit: Kristen Grauman (UT Austin)

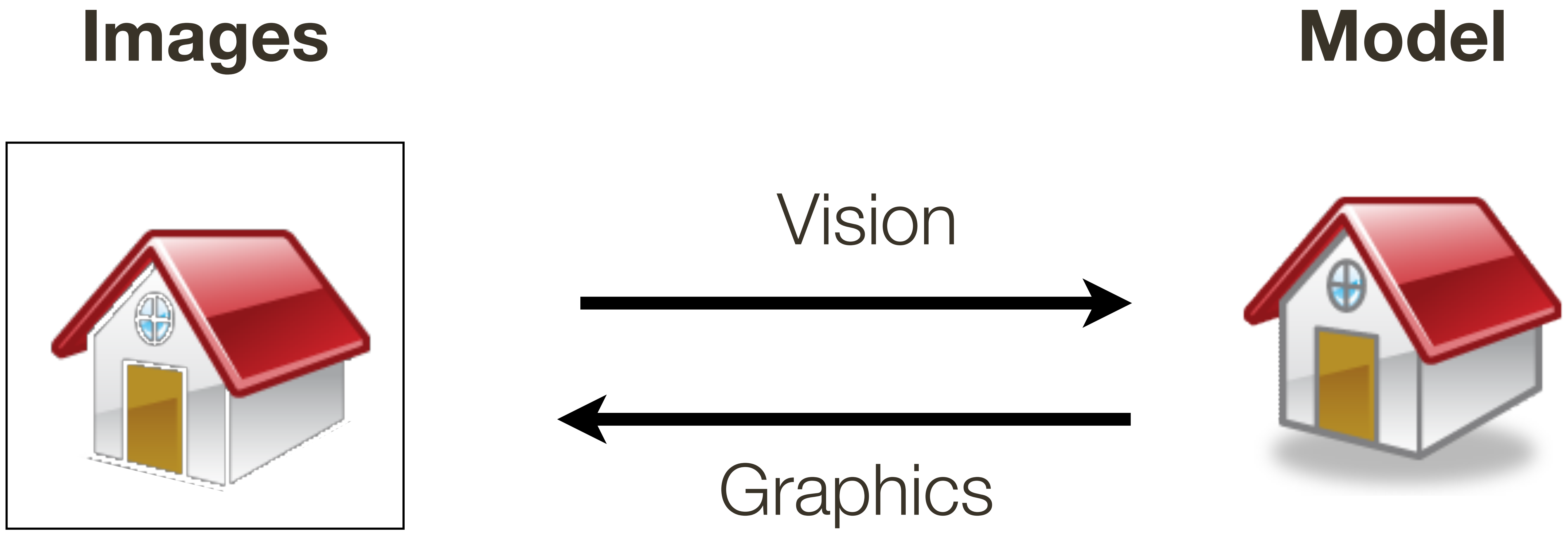
Related Disciplines: Vision and Graphics



Inverse problems: analysis and synthesis

Slide Credit: Kristen Grauman (UT Austin)

Related Disciplines: Vision and Graphics



Inverse problems: analysis and synthesis

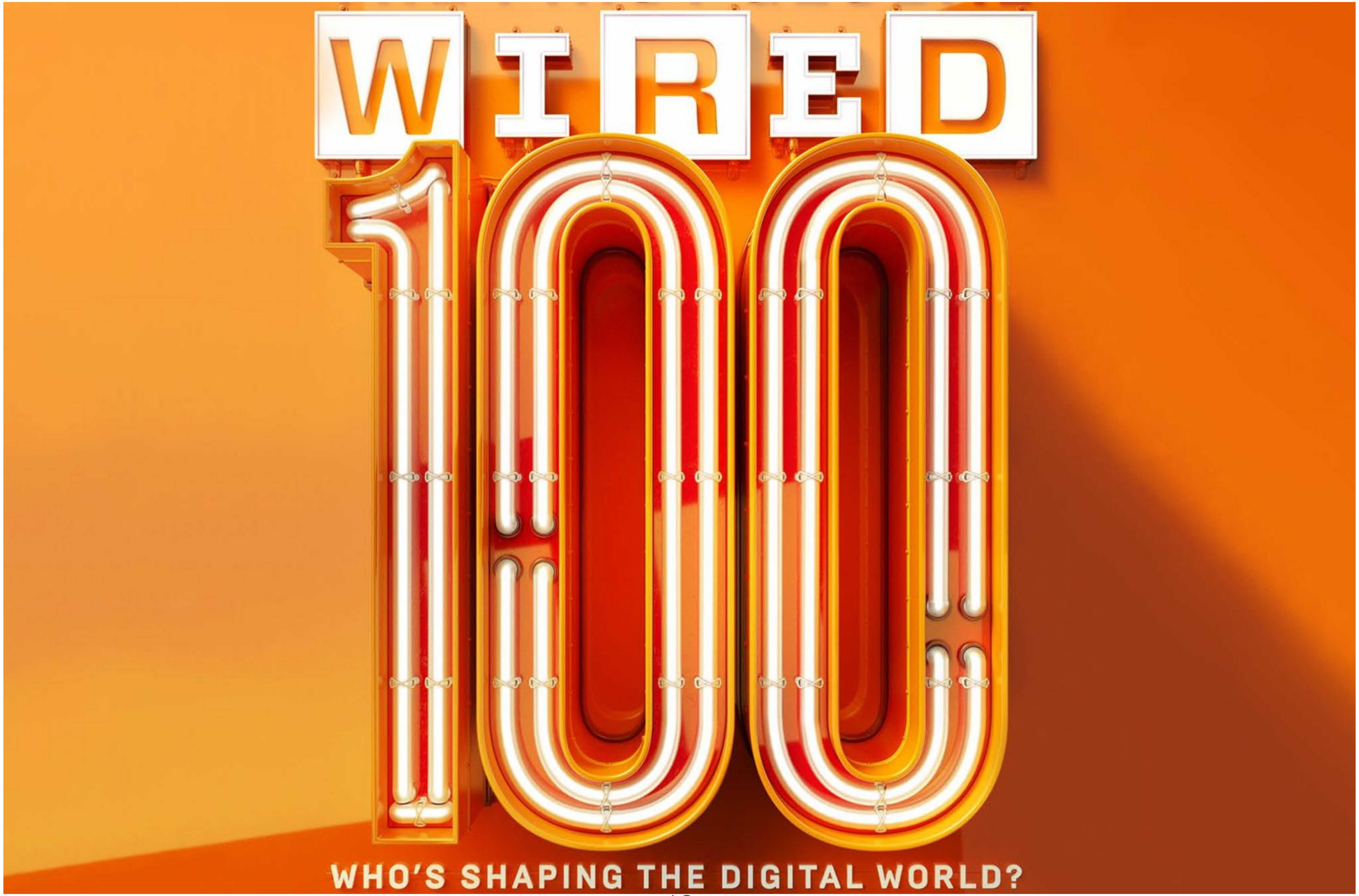
(it is sometimes useful to think about computer **vision as inverse graphics**)

Slide Credit: Kristen Grauman (UT Austin)

Why Study Computer Vision?

It is one of the **most exciting areas of research** in computer science

Among the **fastest growing technologies** in the industry today


WIRED
100
WHO'S SHAPING THE DIGITAL WORLD?

Wired's 100 **Most Influential People** in the World

63. Yann Lecun

Director of AI research, Facebook, Menlo Park

LeCun is a leading expert in deep learning and heads up what, for Facebook, could be a hugely significant source of revenue: understanding its user's intentions.

62. Richard Branson

Founder, Virgin Group, London

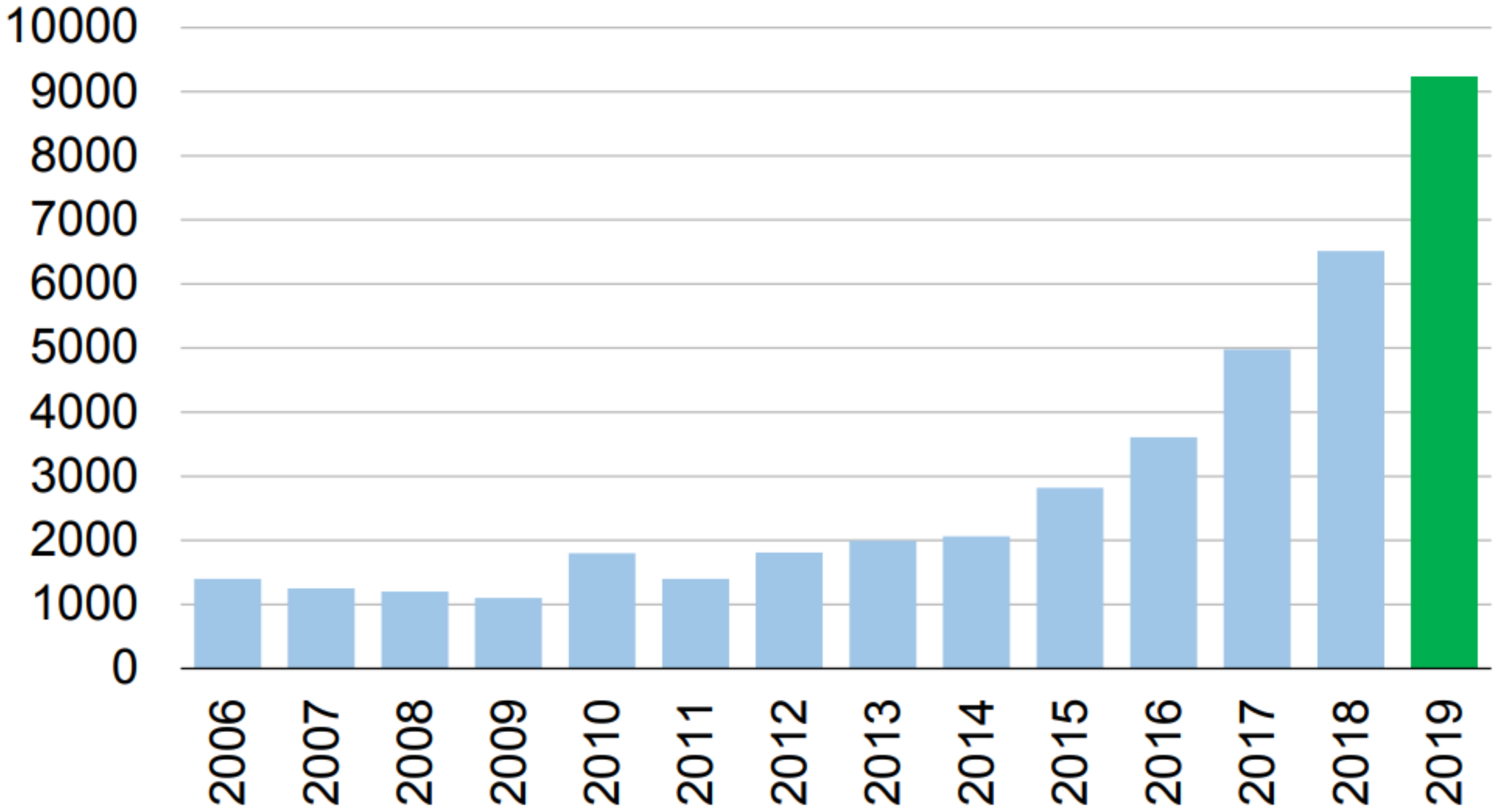
Branson saw his personal fortune grow £550 million when Alaska Air bought Virgin America for $2.6 billion in April. He is pressing on with civilian space travel with Virgin Galactic.

61. Taylor Swift

Entertainer, Los Angeles


CVPR
2019
Long Beach, CA
June 16th - June 20th

CVPR **Attendance**



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.

Image (or video)

Sensing Device

Interpreting Device

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

Interpretation

blue sky,
trees,
fountains,
UBC, …

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.

Image (or video)

Sensing Device

Interpreting Device

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

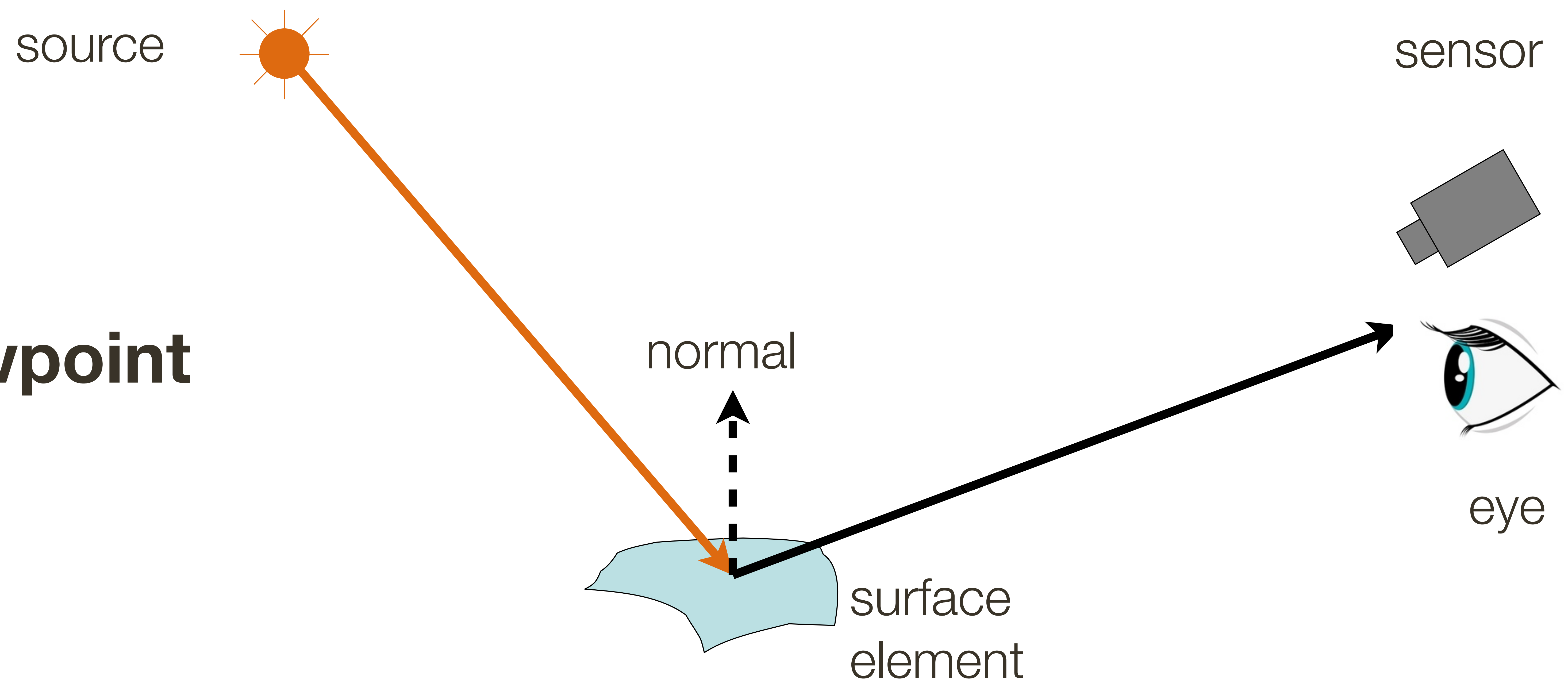
Interpretation

blue sky,
trees,
fountains,
UBC, …

Overview: Image Formation, Cameras and Lenses

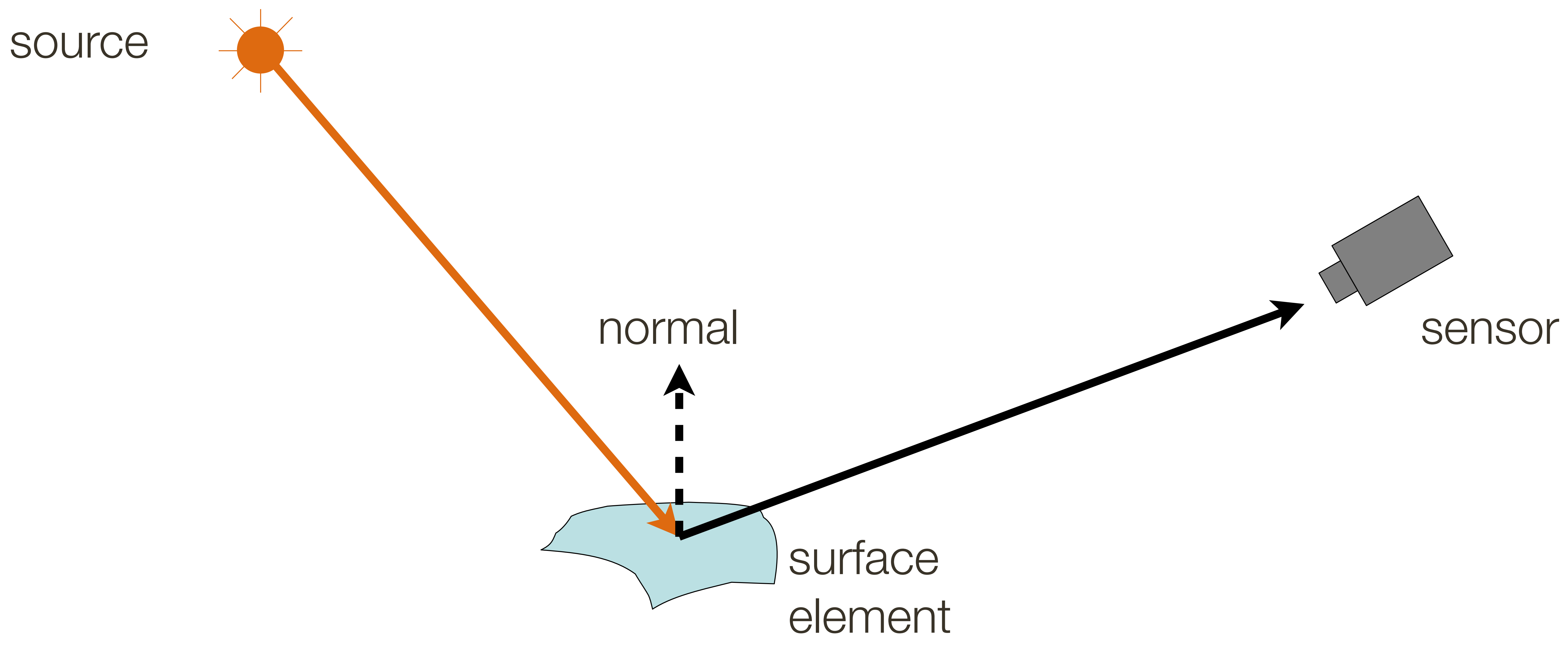
The **image formation process** that produces a particular image depends on

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

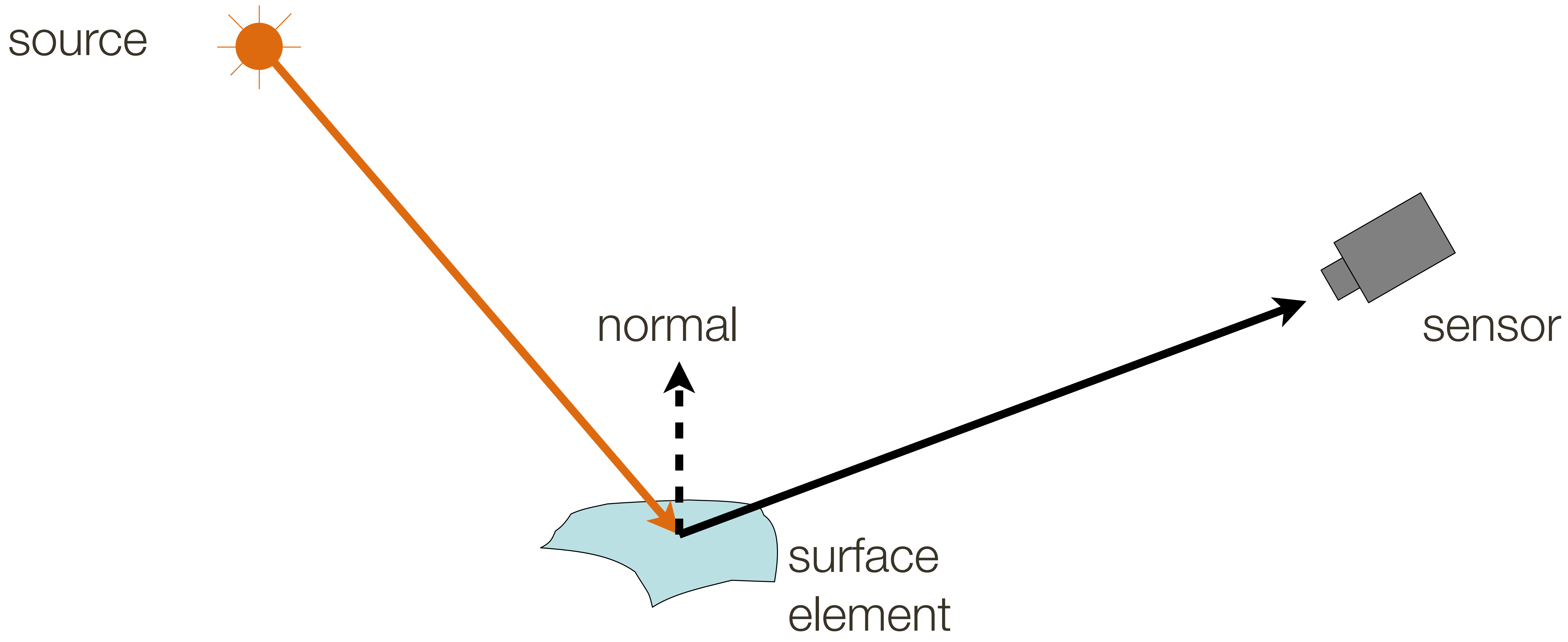


Sensor (or eye) **captures amount of light** reflected from the object

(small) **Graphics** Review

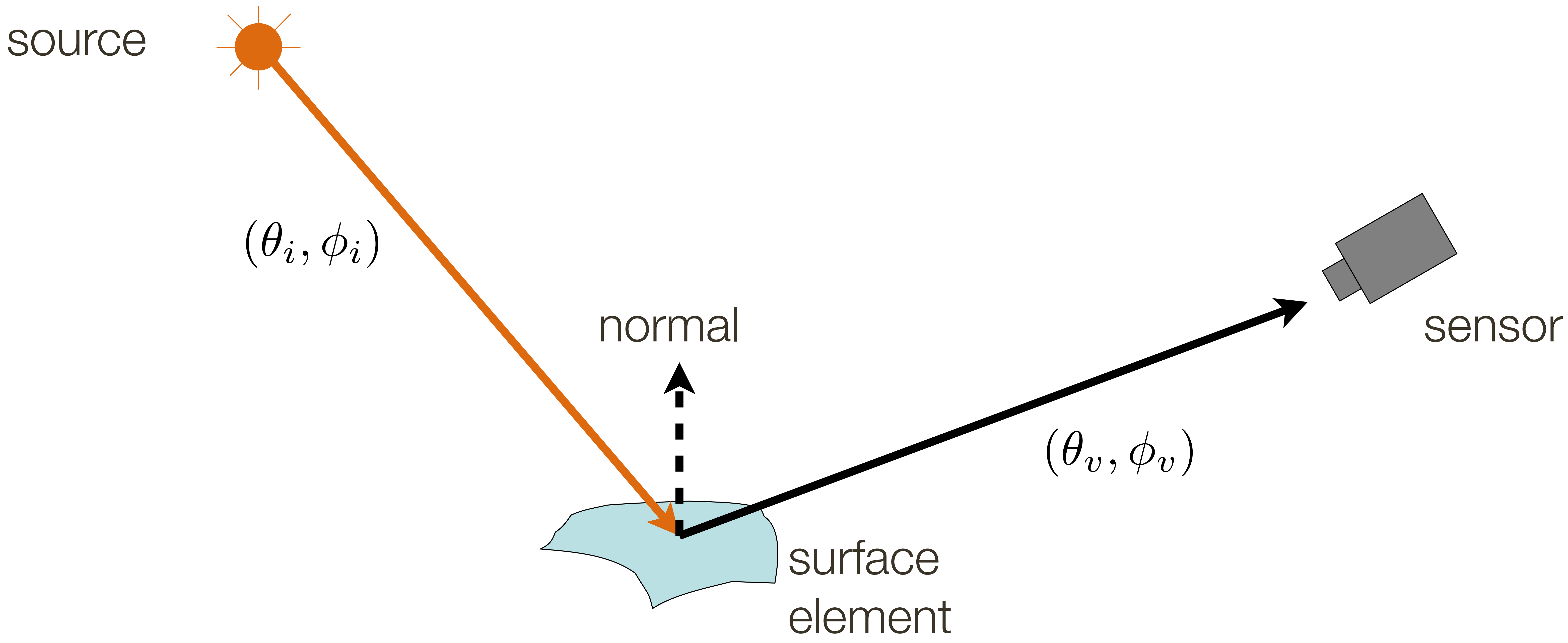


(small) **Graphics** Review



(small) **Graphics** Review

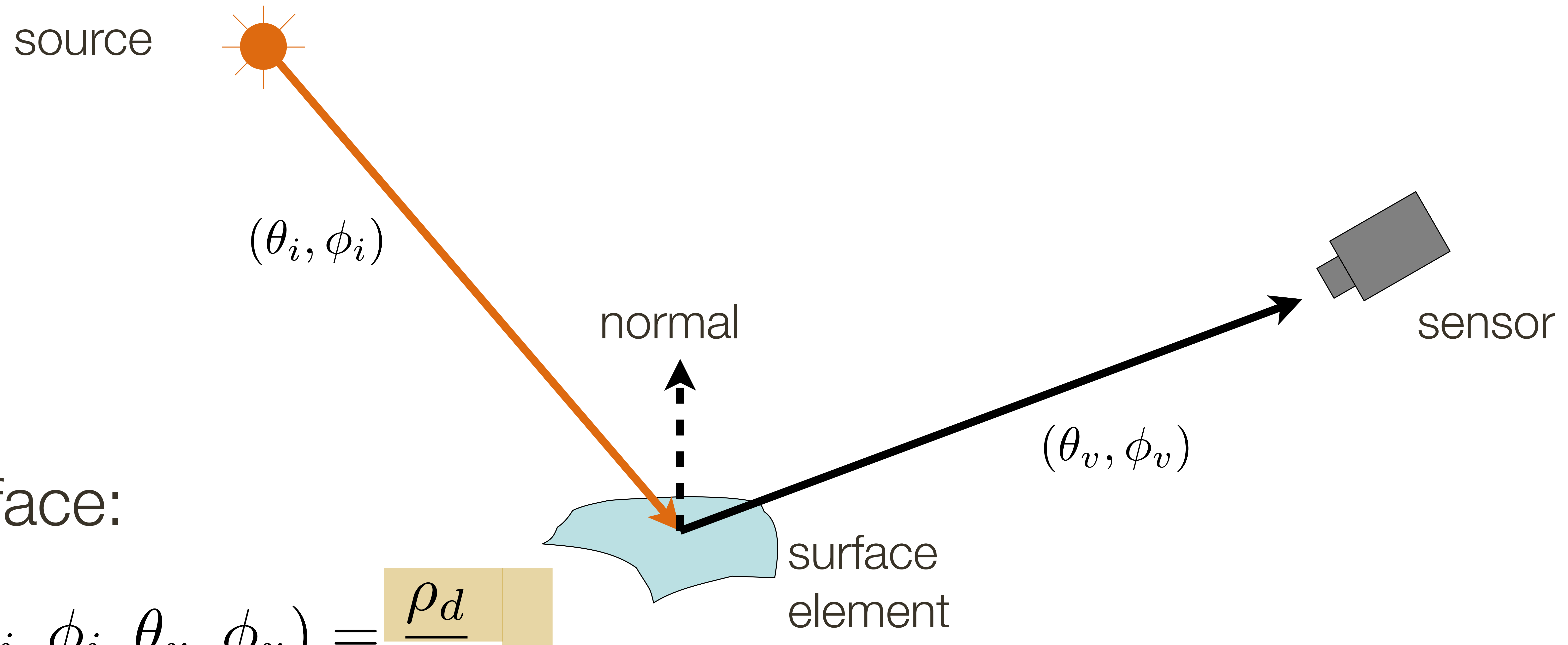
Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



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:

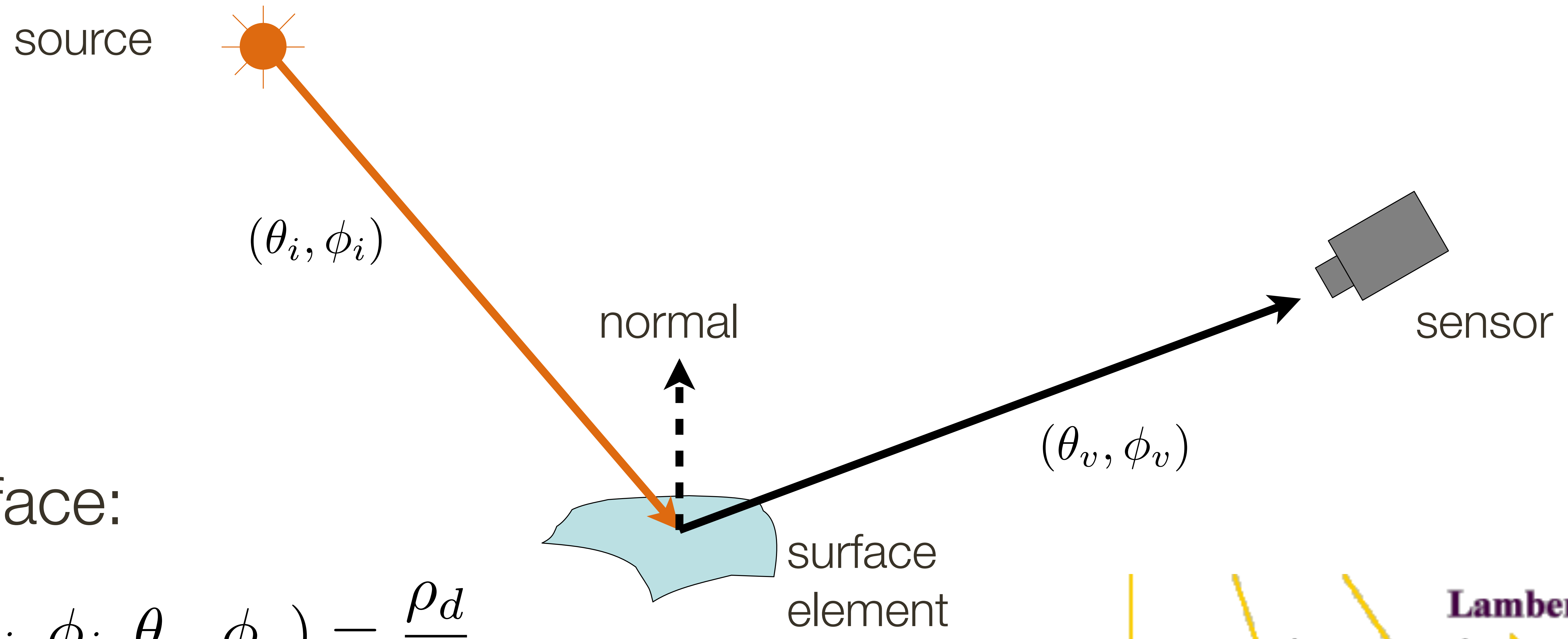
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$

constant, called **albedo**

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

(small) **Graphics** Review

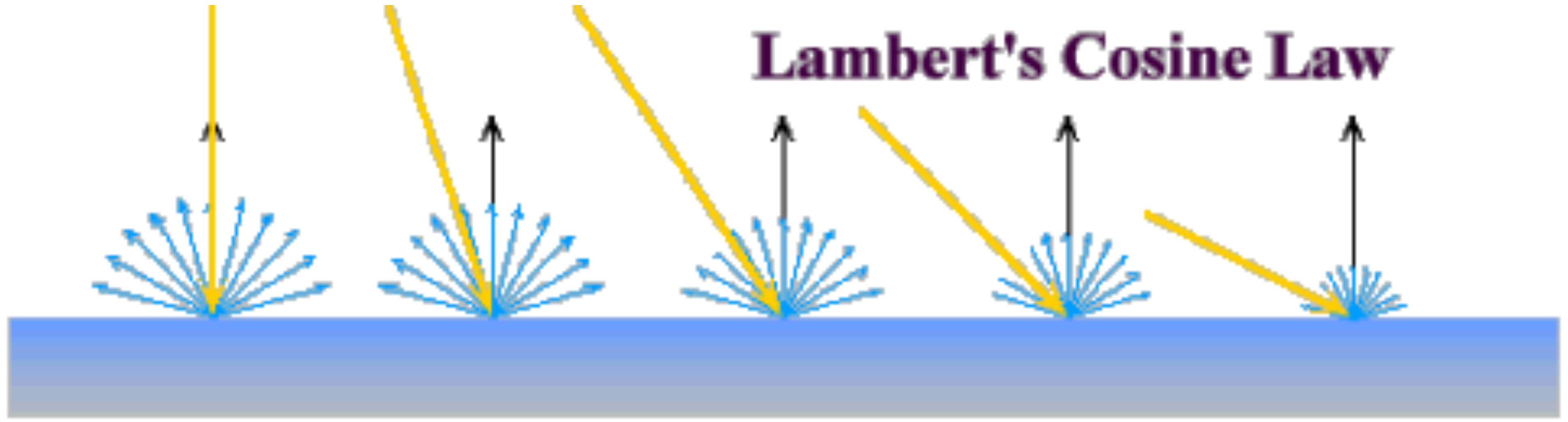
Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

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

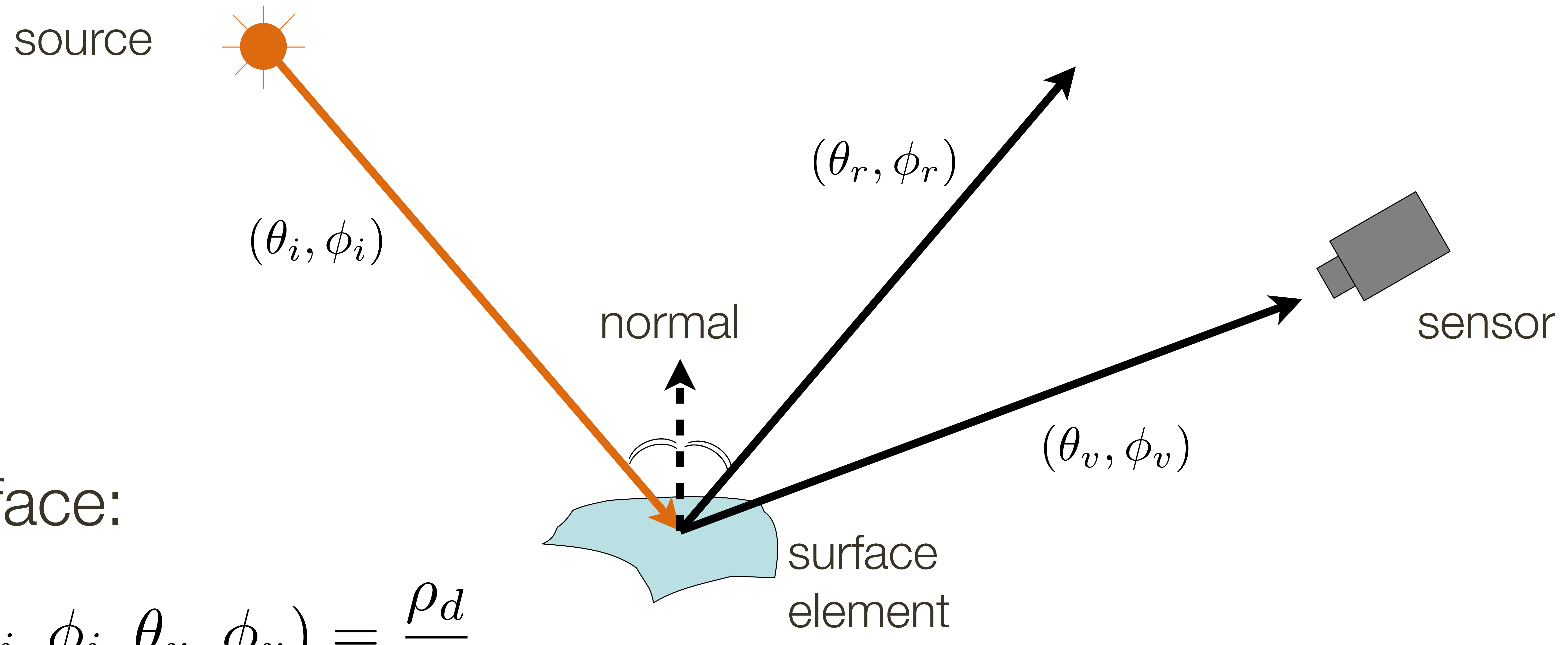
$$L = \frac{\rho_d}{\pi} I(\vec{i} \cdot \vec{n})$$



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)

Cameras

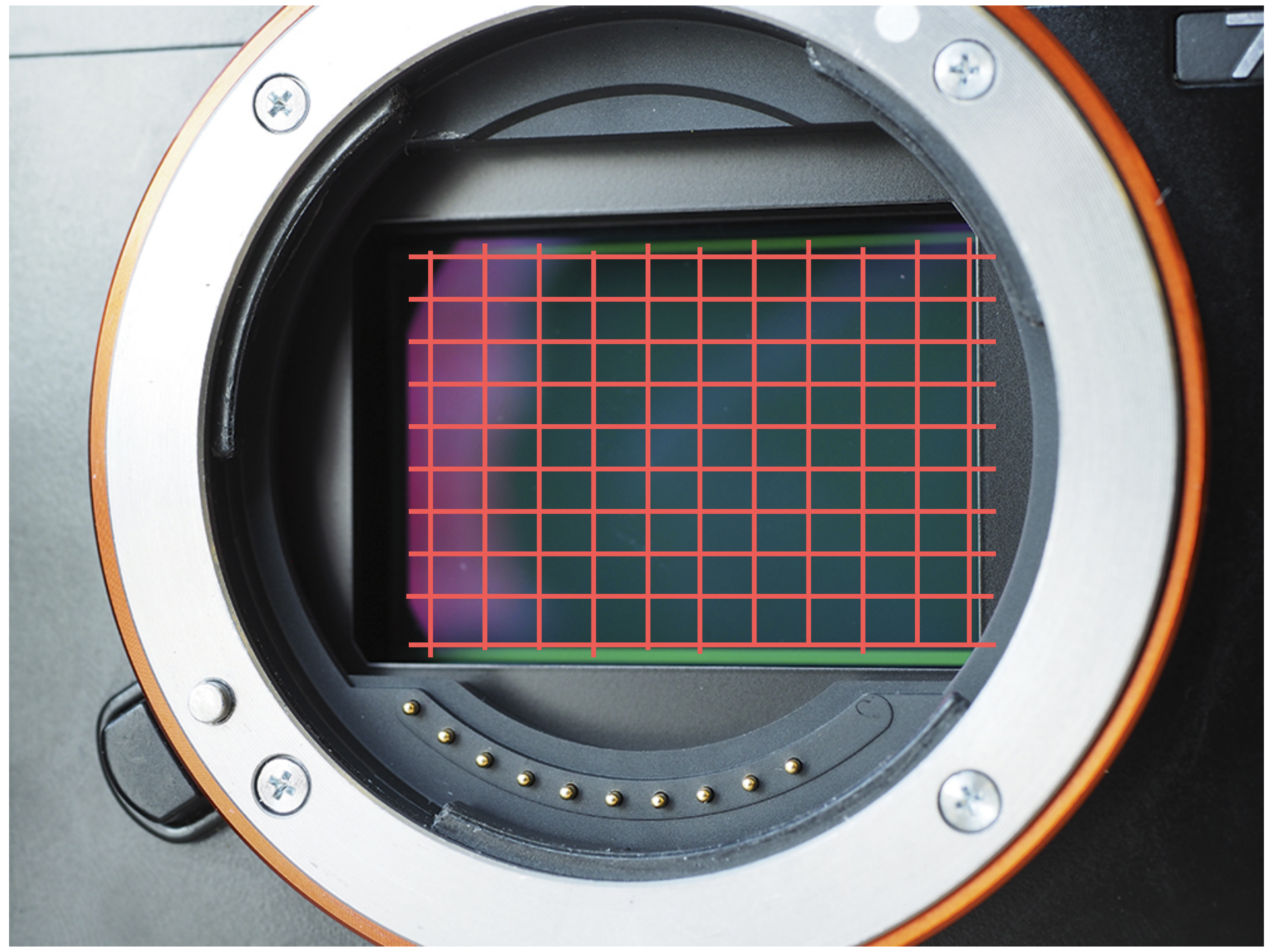
Old school **film** camera

Digital CCD/CMOS camera

Cameras

Old school **film** camera

Digital CCD/CMOS camera



Let's say we have a **sensor** …

Digital CCD/CMOS camera

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

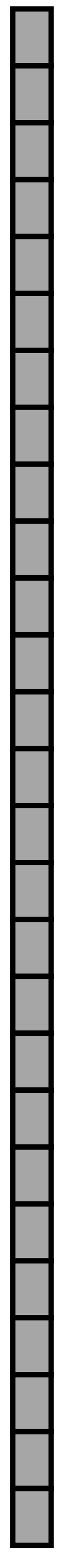
Let's say we have a **sensor** ...

Digital CCD/CMOS camera

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Let's say we have a **sensor** …

Digital CCD/CMOS camera



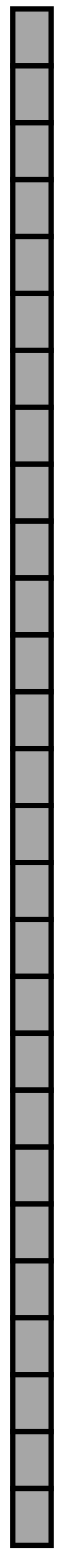
digital sensor
(CCD or
CMOS)

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



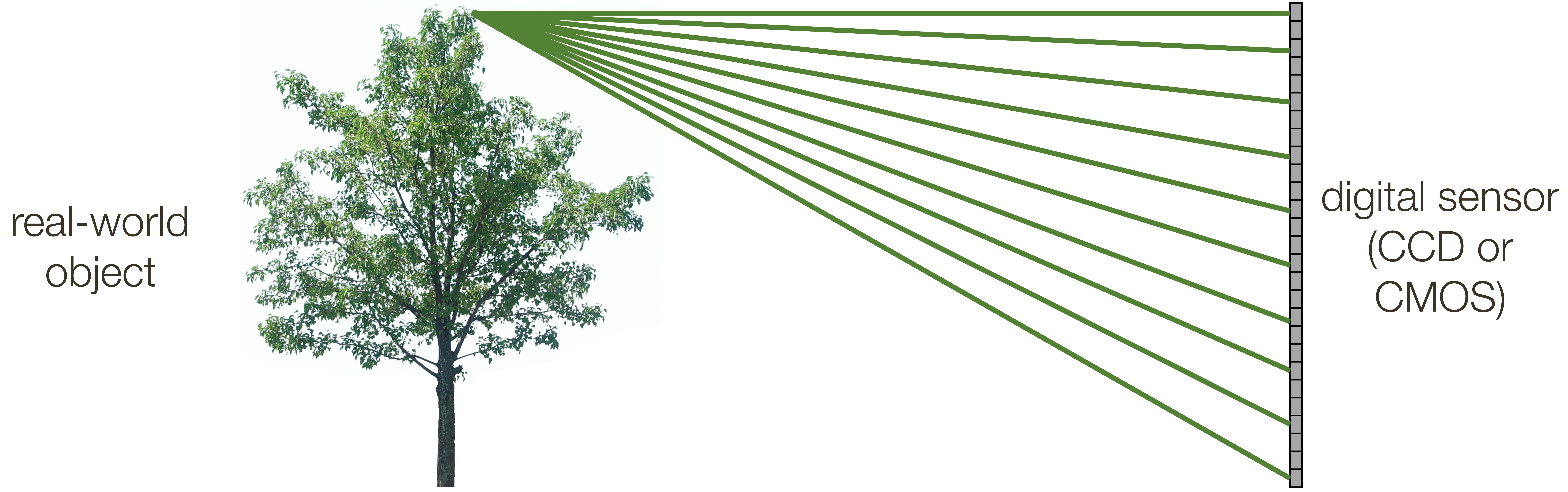
digital sensor (CCD or CMOS)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging

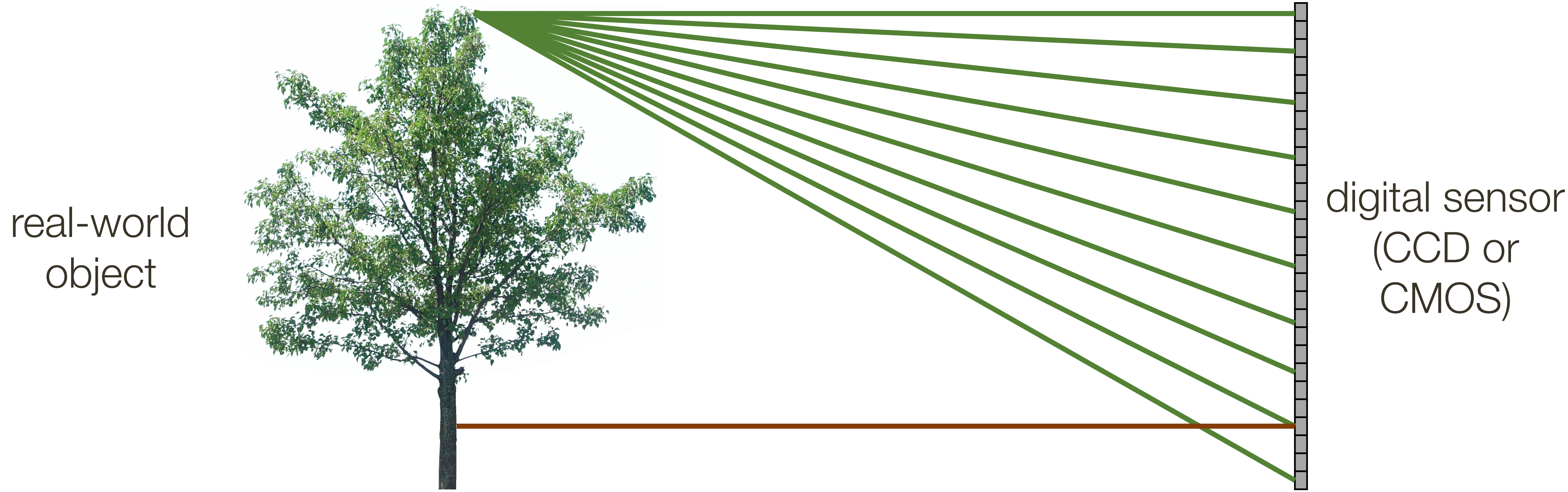
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

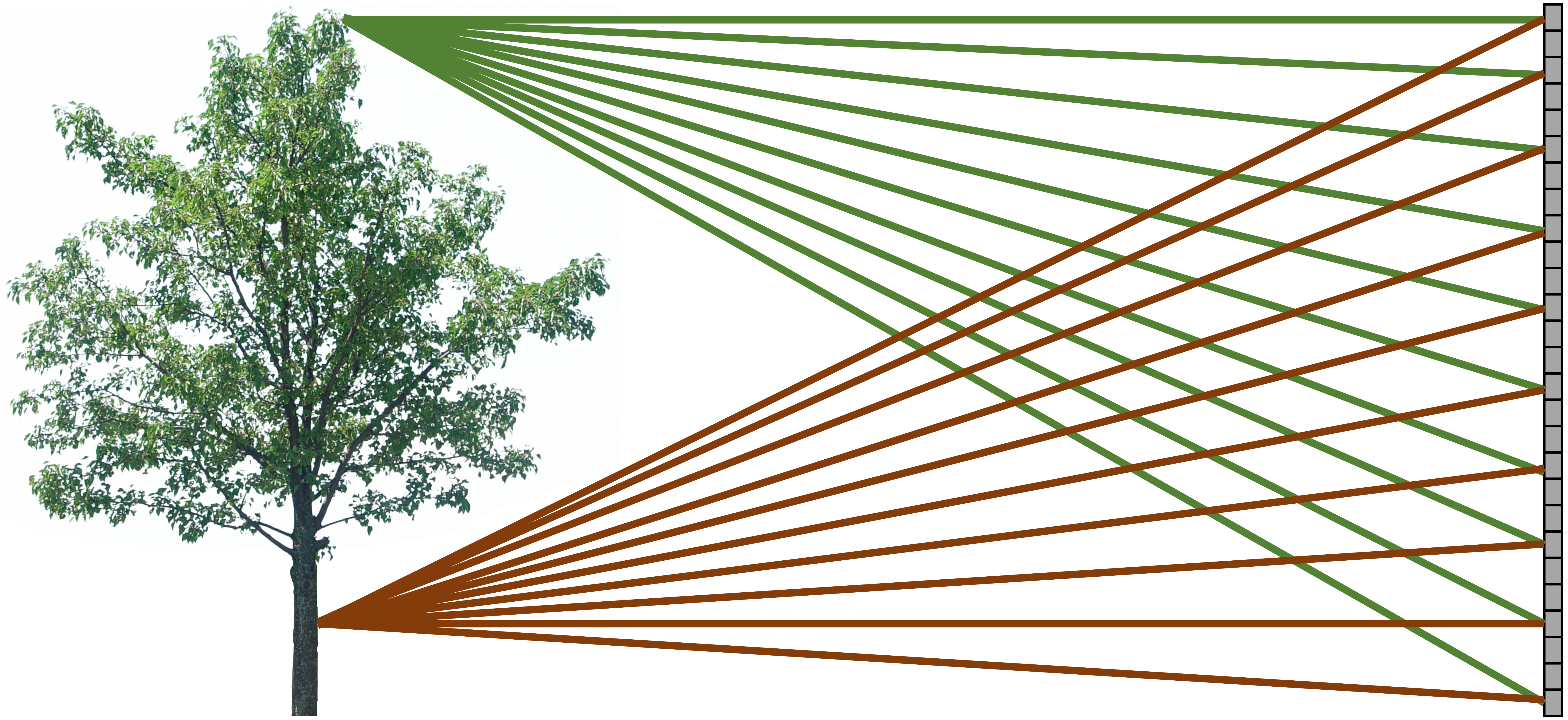
Bare-sensor imaging



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging

real-world object



digital sensor (CCD or CMOS)

All scene points contribute to all sensor pixels

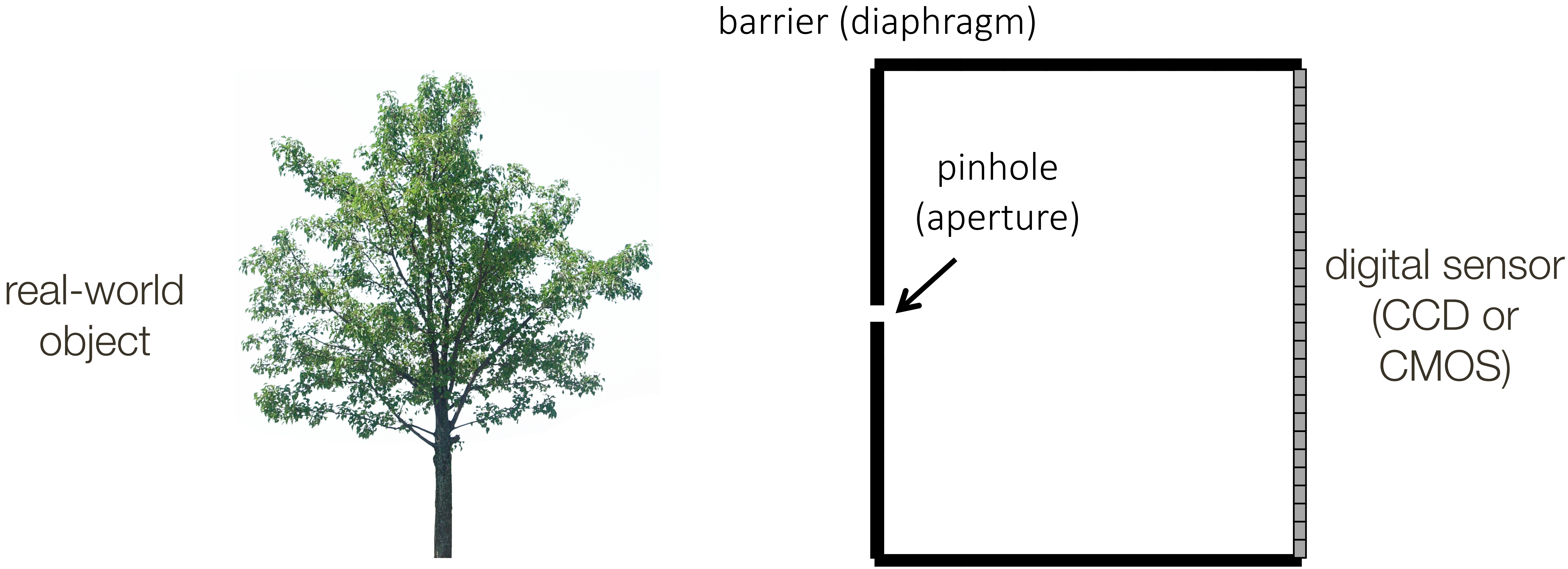
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Bare-sensor imaging

All scene points contribute to all sensor pixels

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

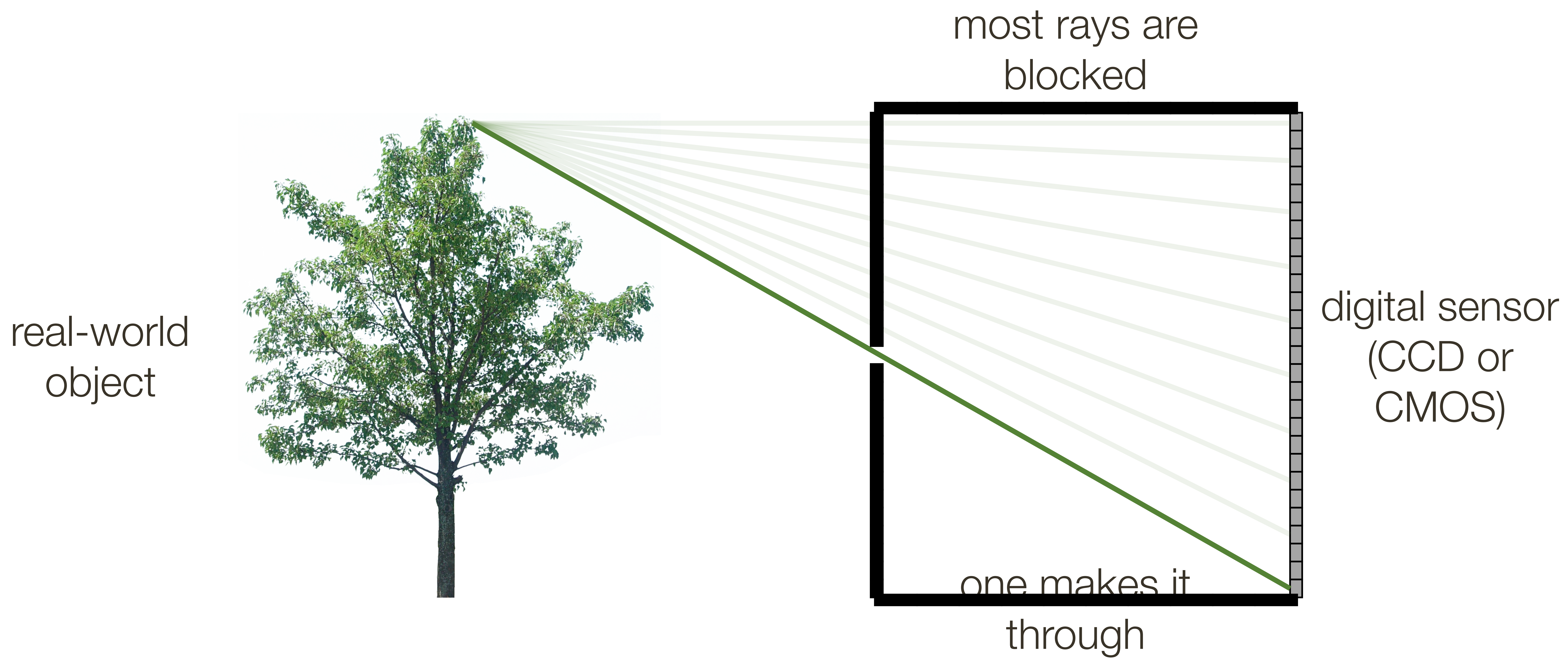
Pinhole Camera



What would an image taken like this look like?

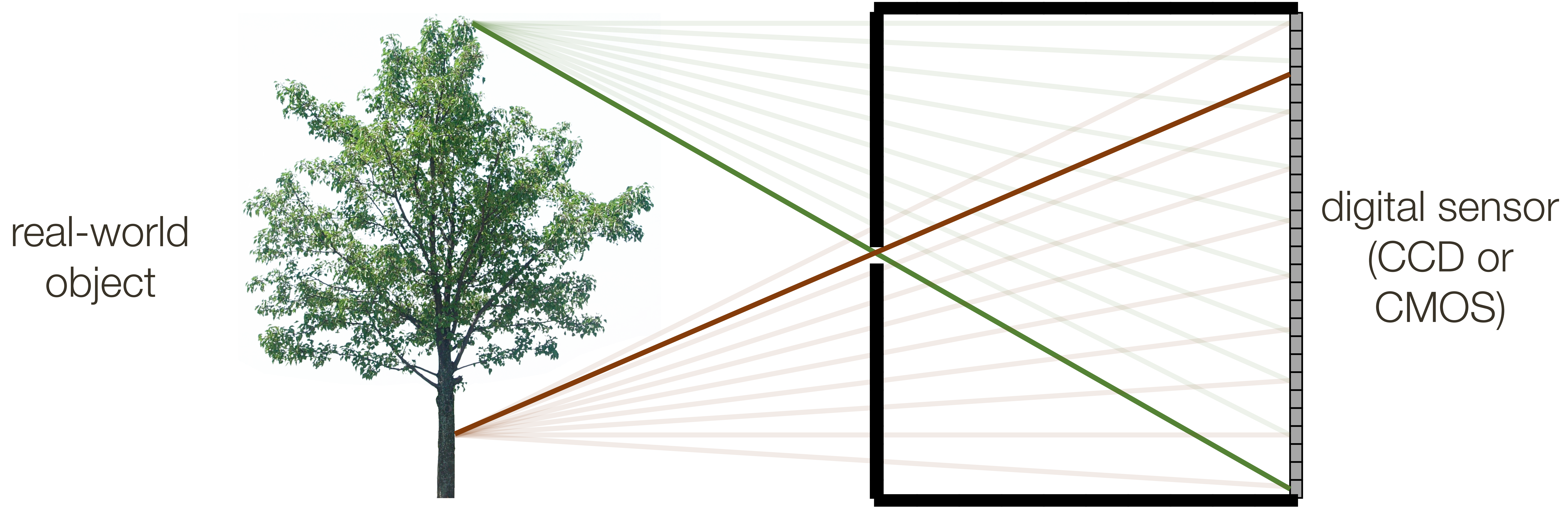
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Pinhole Camera



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

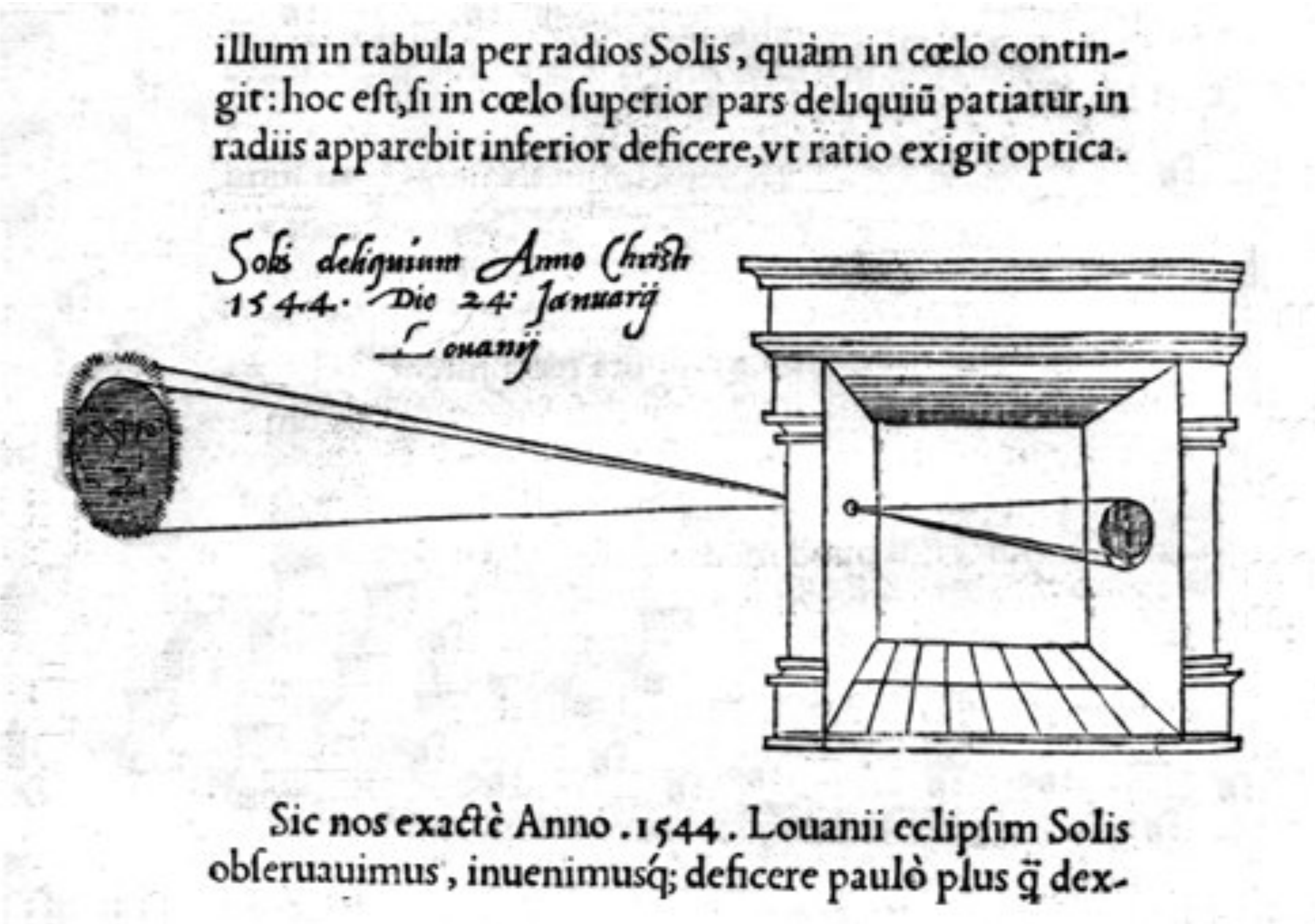
Pinhole Camera



Each scene point contributes to only one sensor pixel

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Camera Obscura (latin for “dark chamber”)



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”

Camera Obscura (latin for “dark chamber”)


illum in tabula per radios Solis, quàm in cœlo contingit: 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”

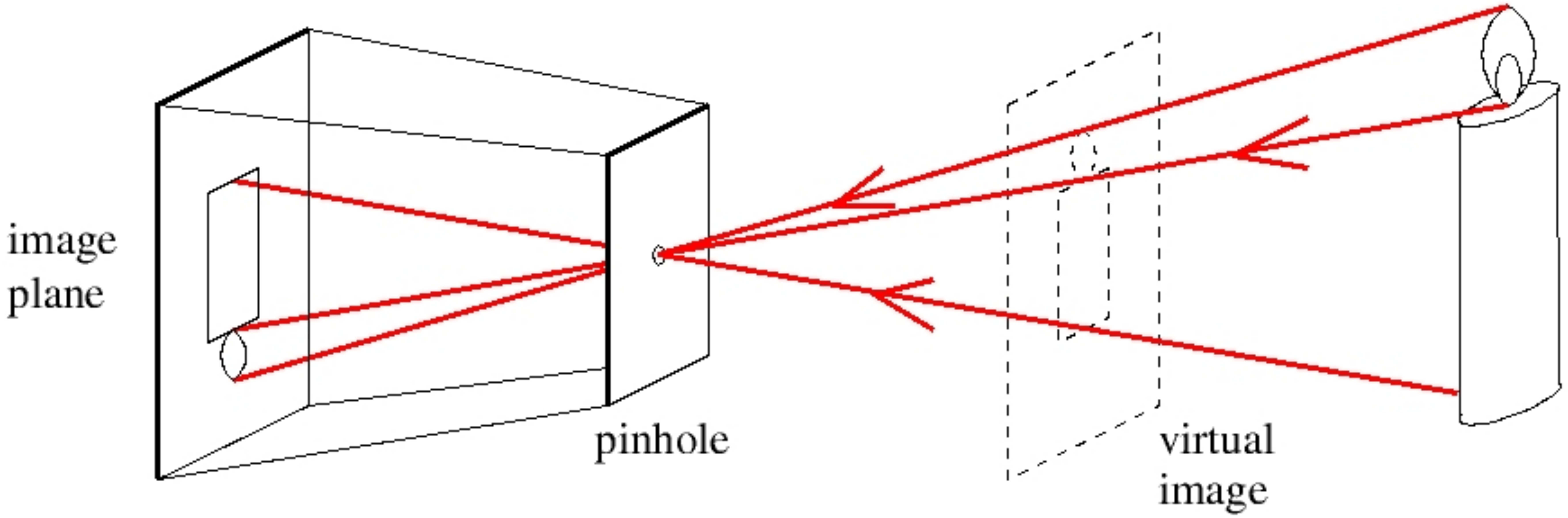
First **Photograph** on Record

La table servie

Credit: Nicéphore Niepce, 1822

Pinhole Camera

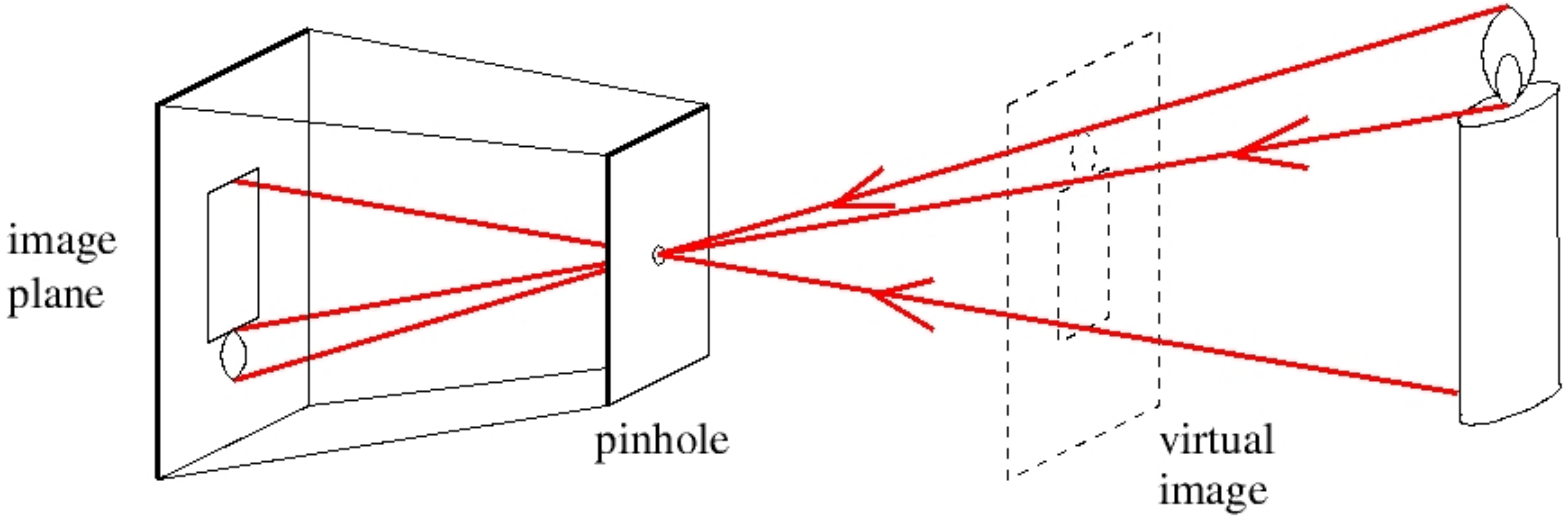
A pinhole camera is a box with a small hall (**aperture**) in it



Forsyth & Ponce (2nd ed.) Figure 1.2

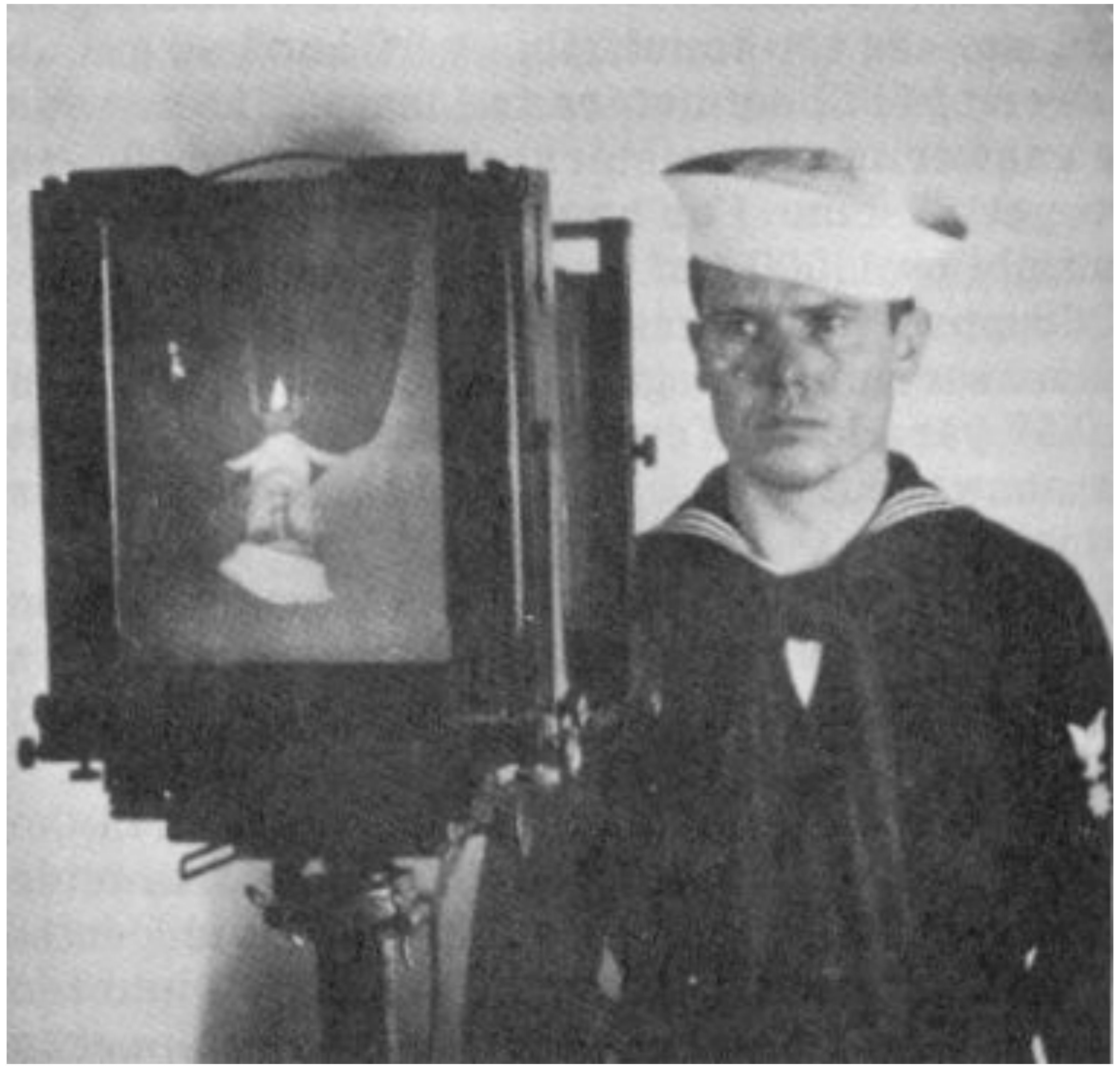
Pinhole Camera

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



Forsyth & Ponce (2nd ed.) Figure 1.2

Image **Formation**



Forsyth & Ponce (2nd ed.) Figure 1.1

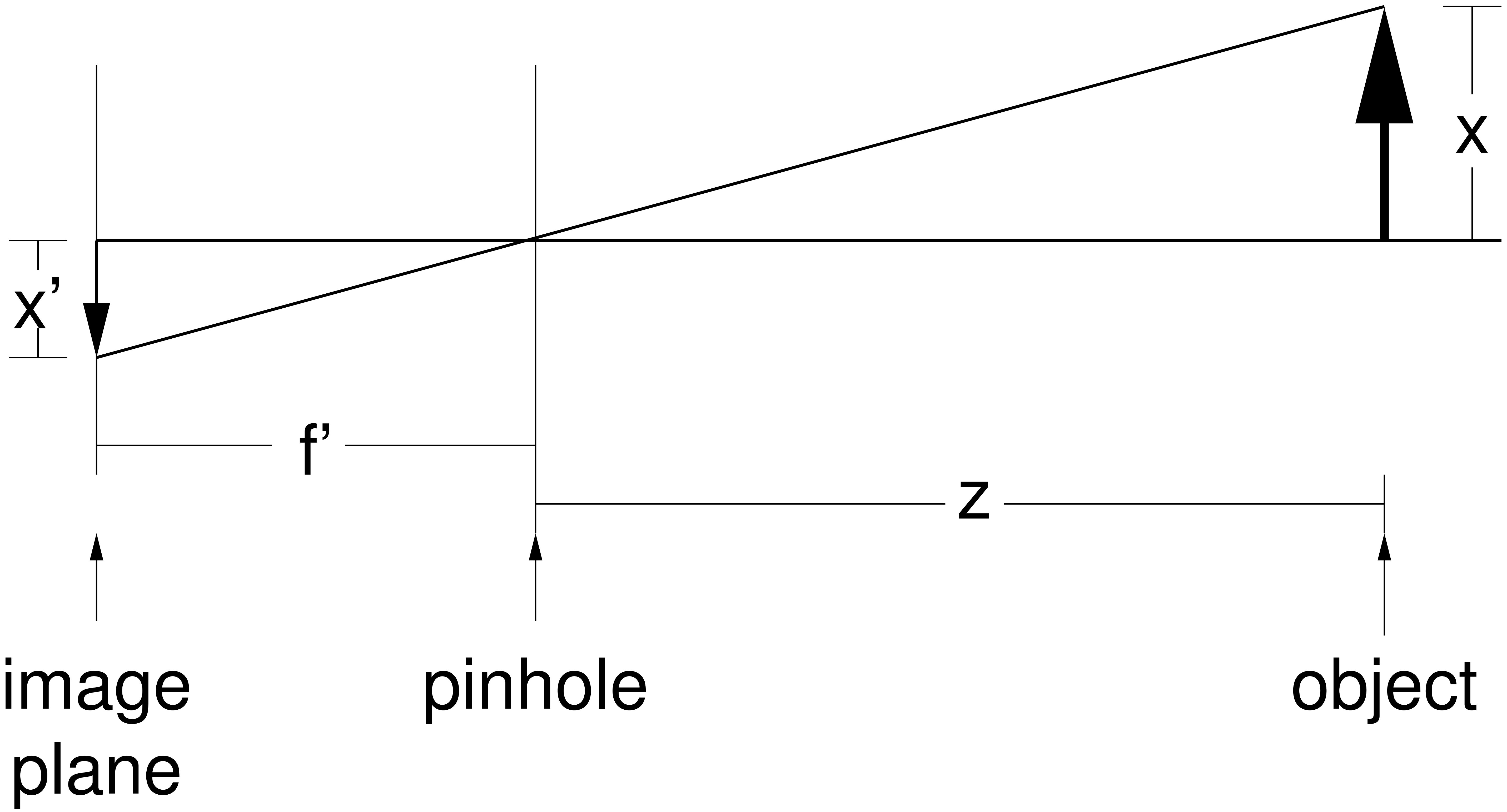
Credit: US Navy, Basic Optics and Optical Instruments. Dover, 1969

Accidental Pinhole Camera

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

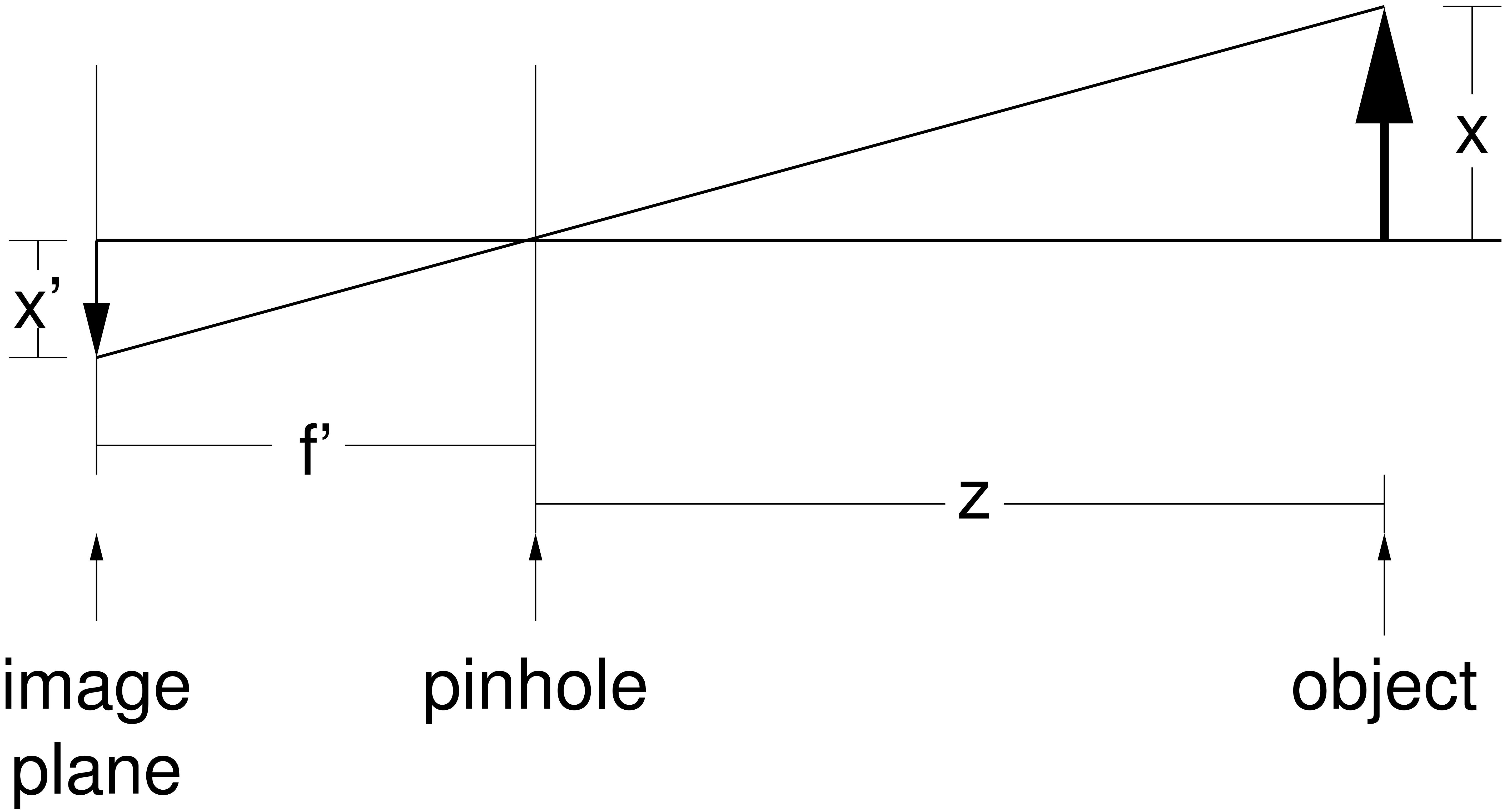
Pinhole Camera (Simplified)

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



Pinhole Camera (Simplified)

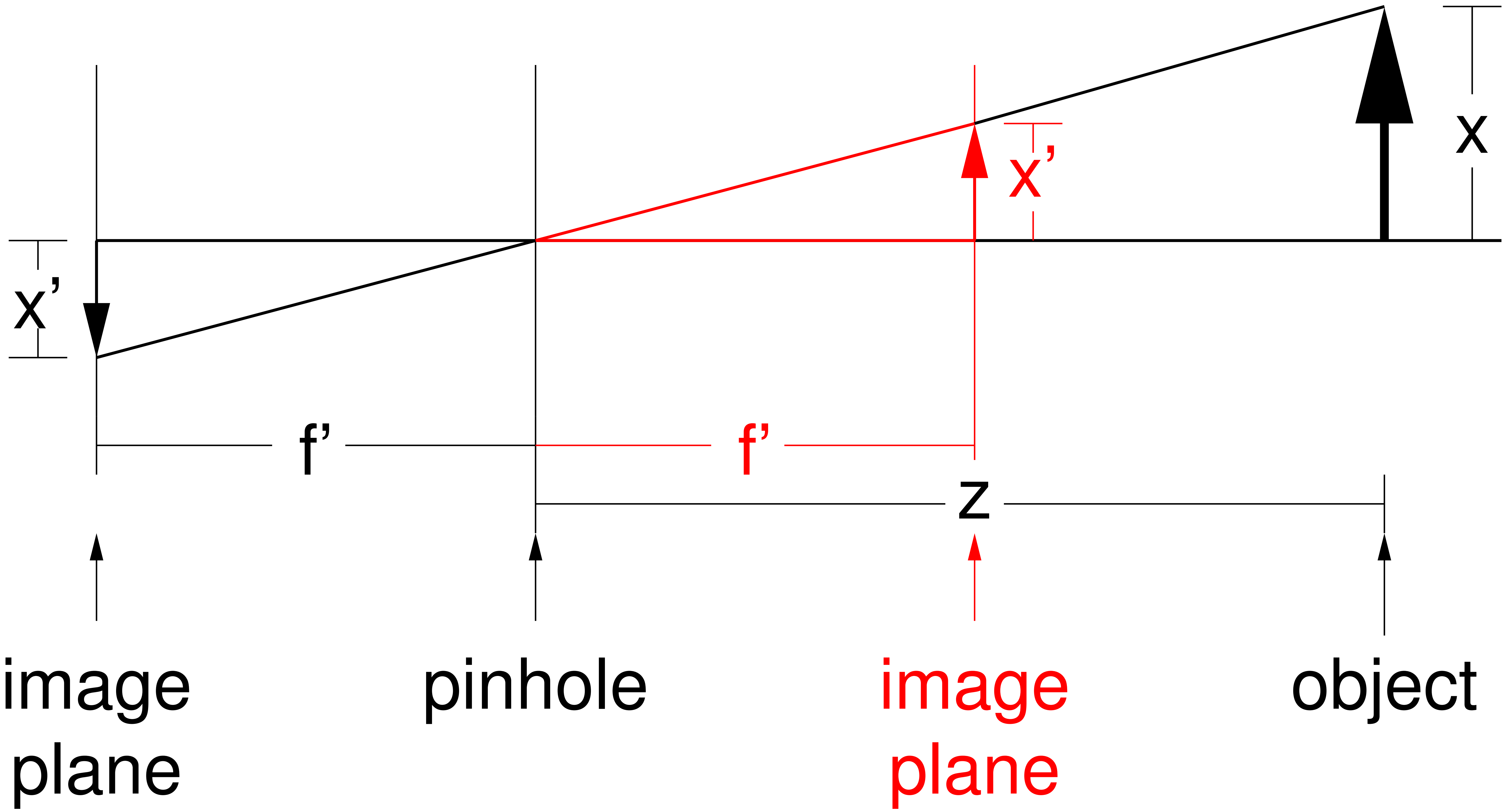
f’ is the **focal length** of the camera



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

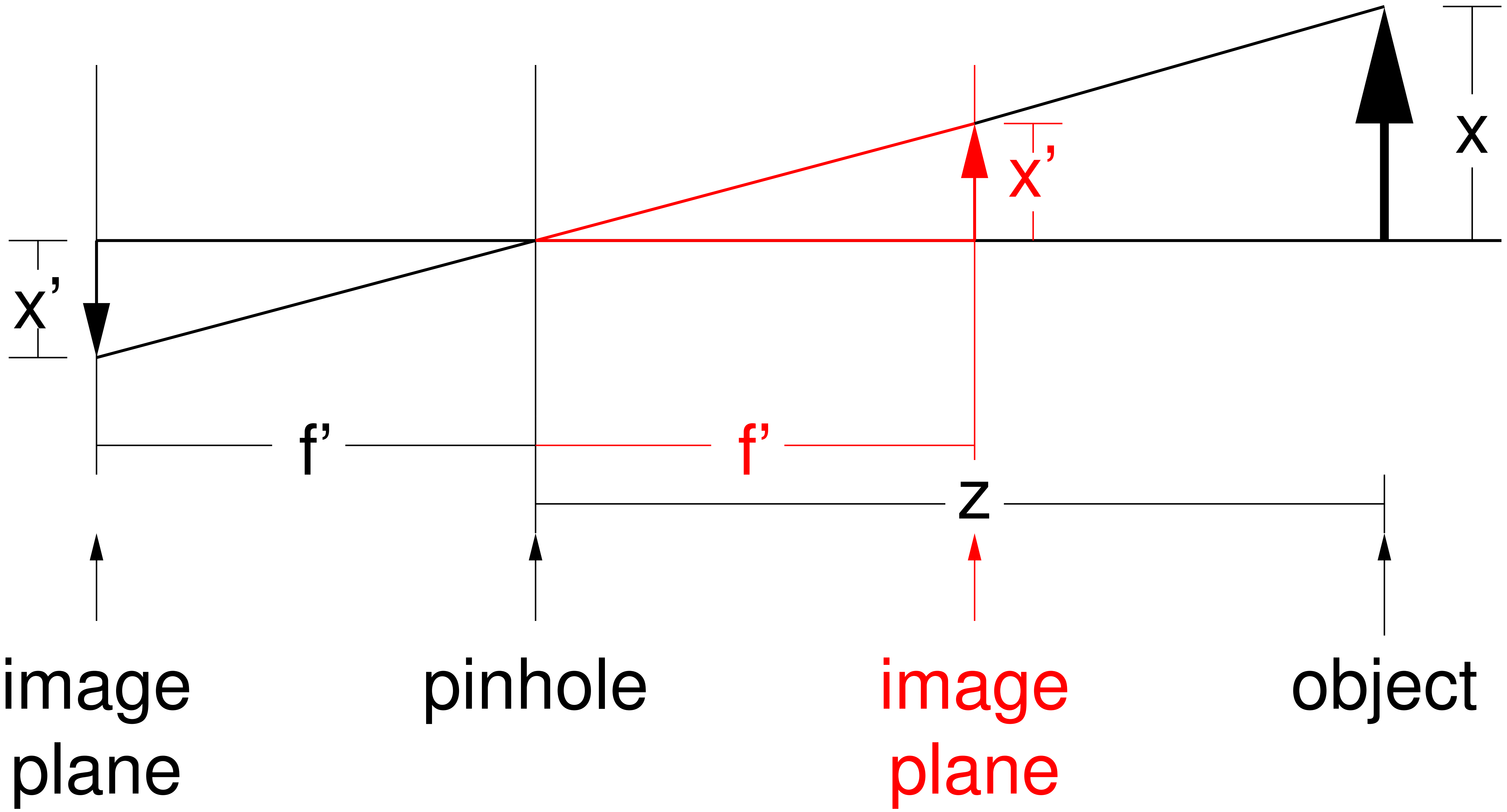
Pinhole Camera (Simplified)

It is convenient to think of the **image plane** which is in from of the pinhole



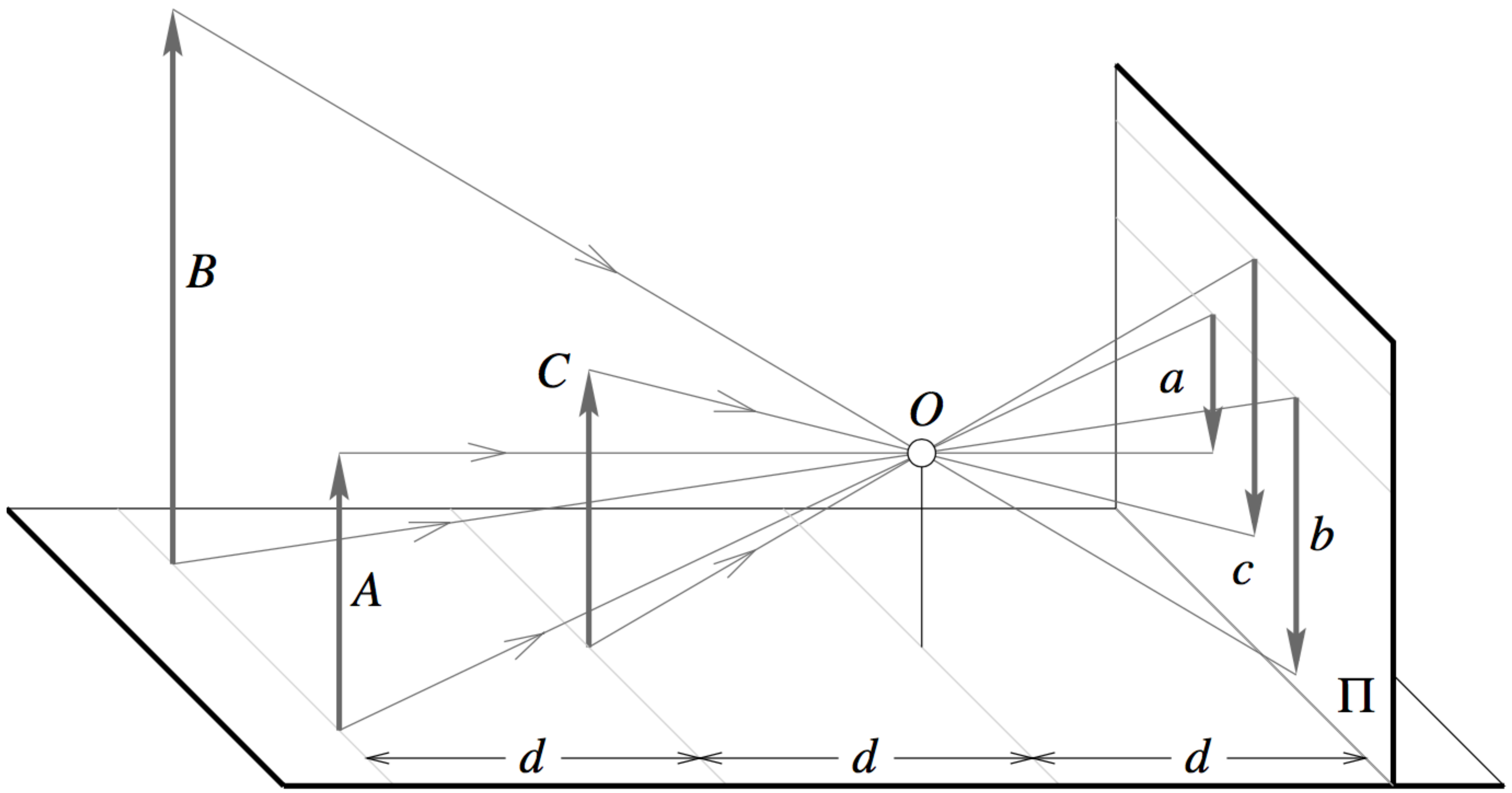
Pinhole Camera (Simplified)

It is convenient to think of the **image plane** which is in from of the pinhole



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

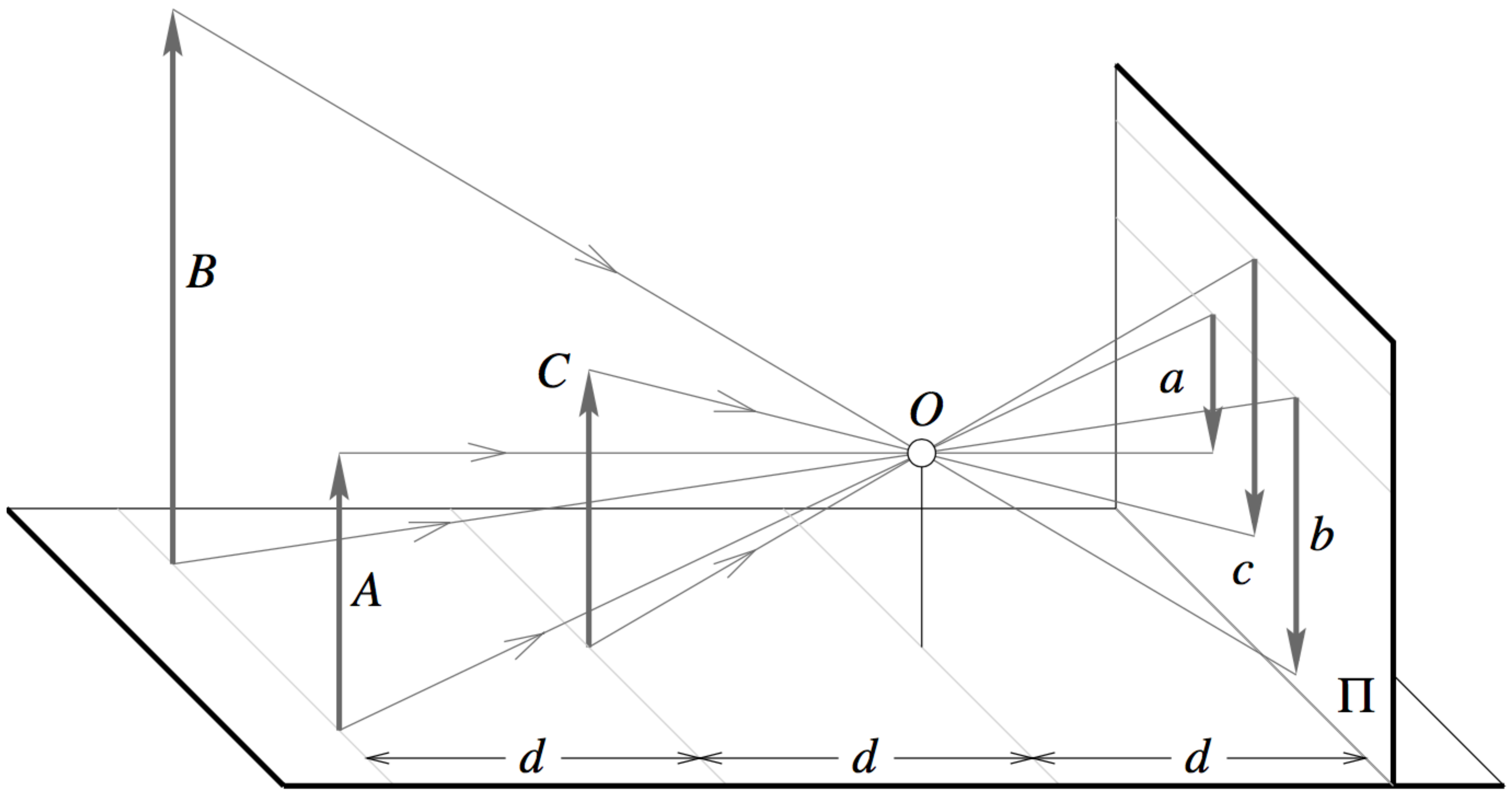
Perspective Effects



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

Perspective Effects

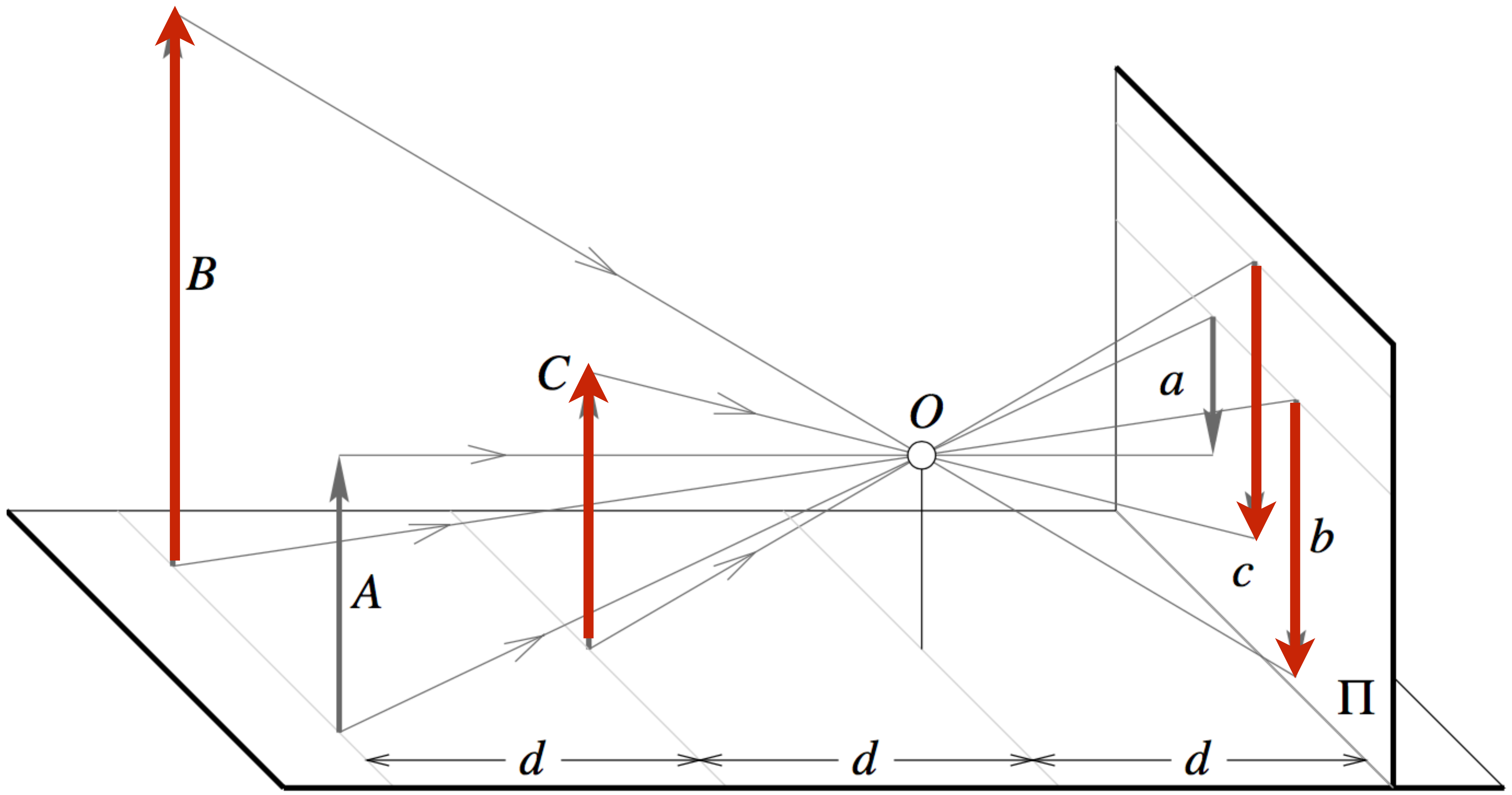
Far objects appear **smaller** than close ones



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

Perspective Effects

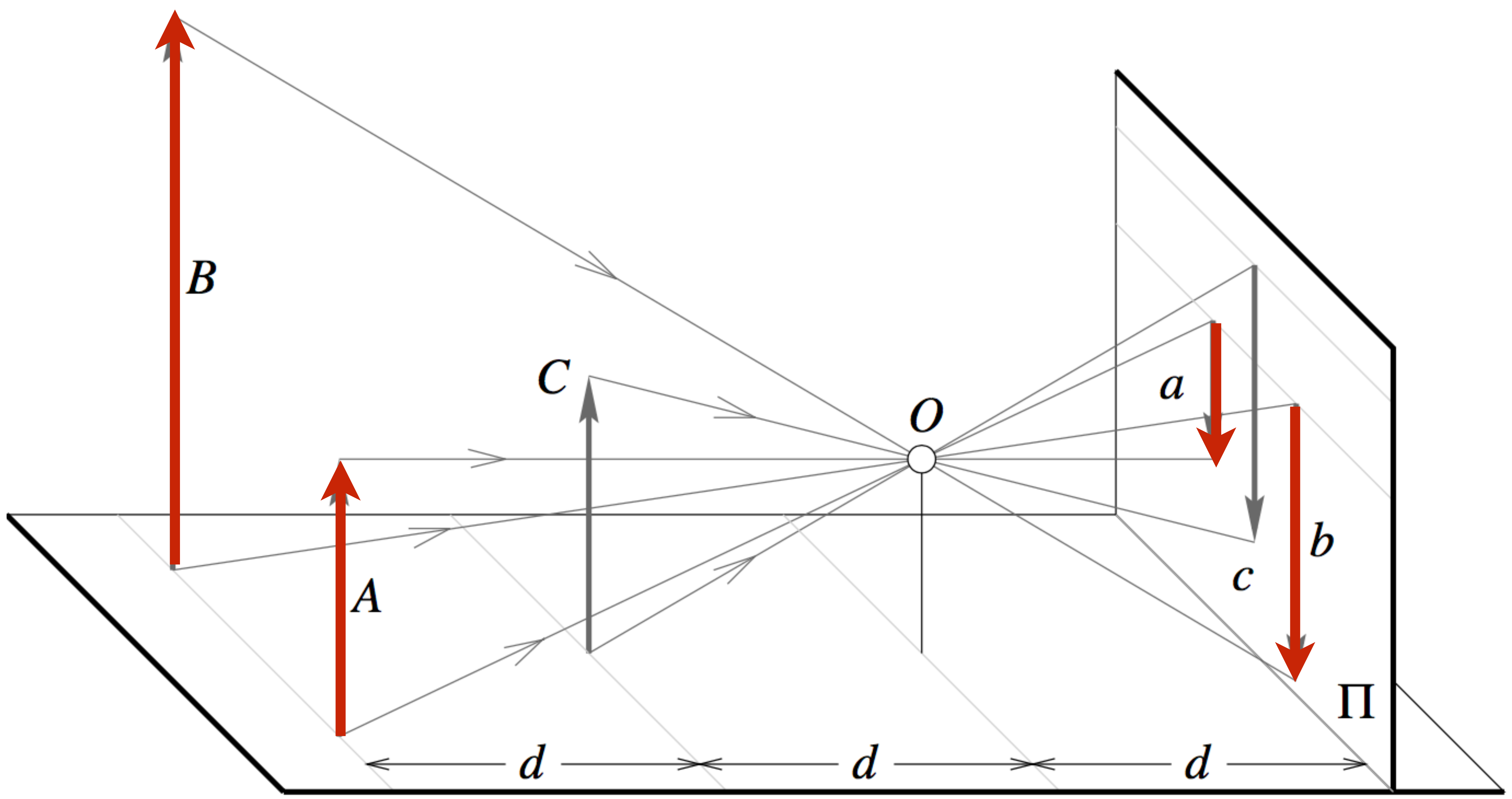
Far objects appear **smaller** than close ones



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

Perspective Effects

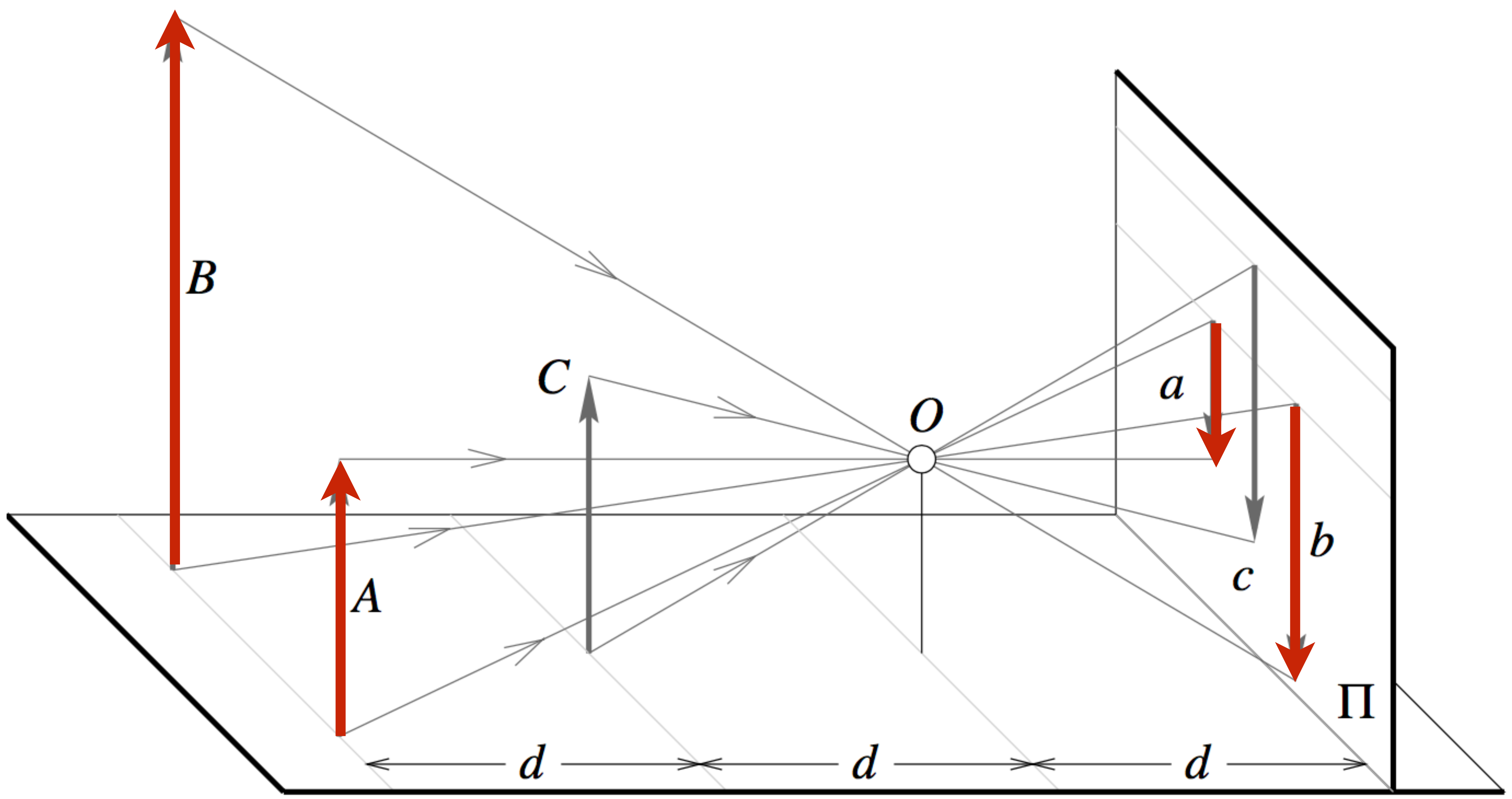
Far objects appear **smaller** than close ones



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

Perspective Effects

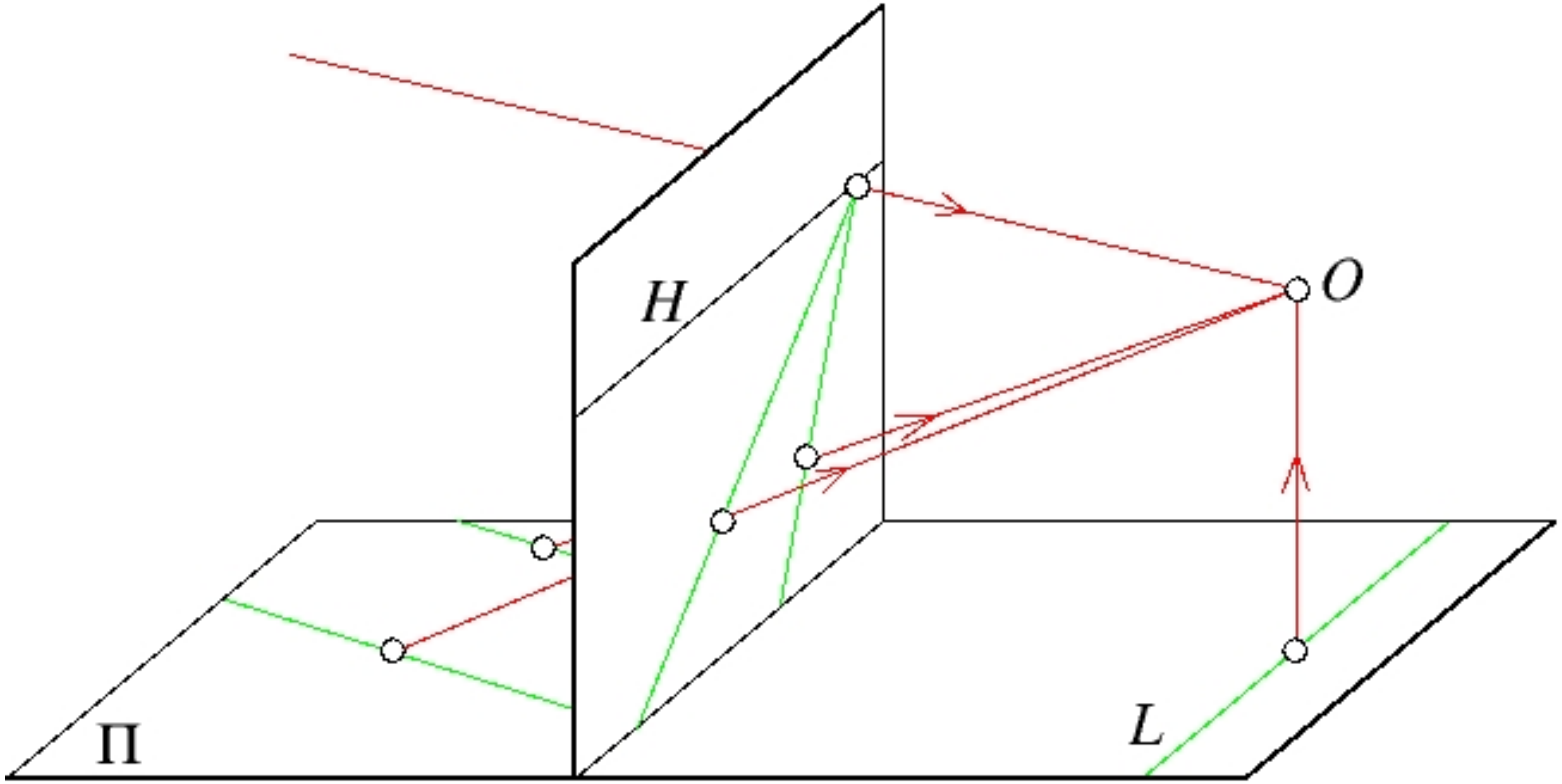
Far objects appear **smaller** than close ones



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

Size is **inversely** proportions to distance

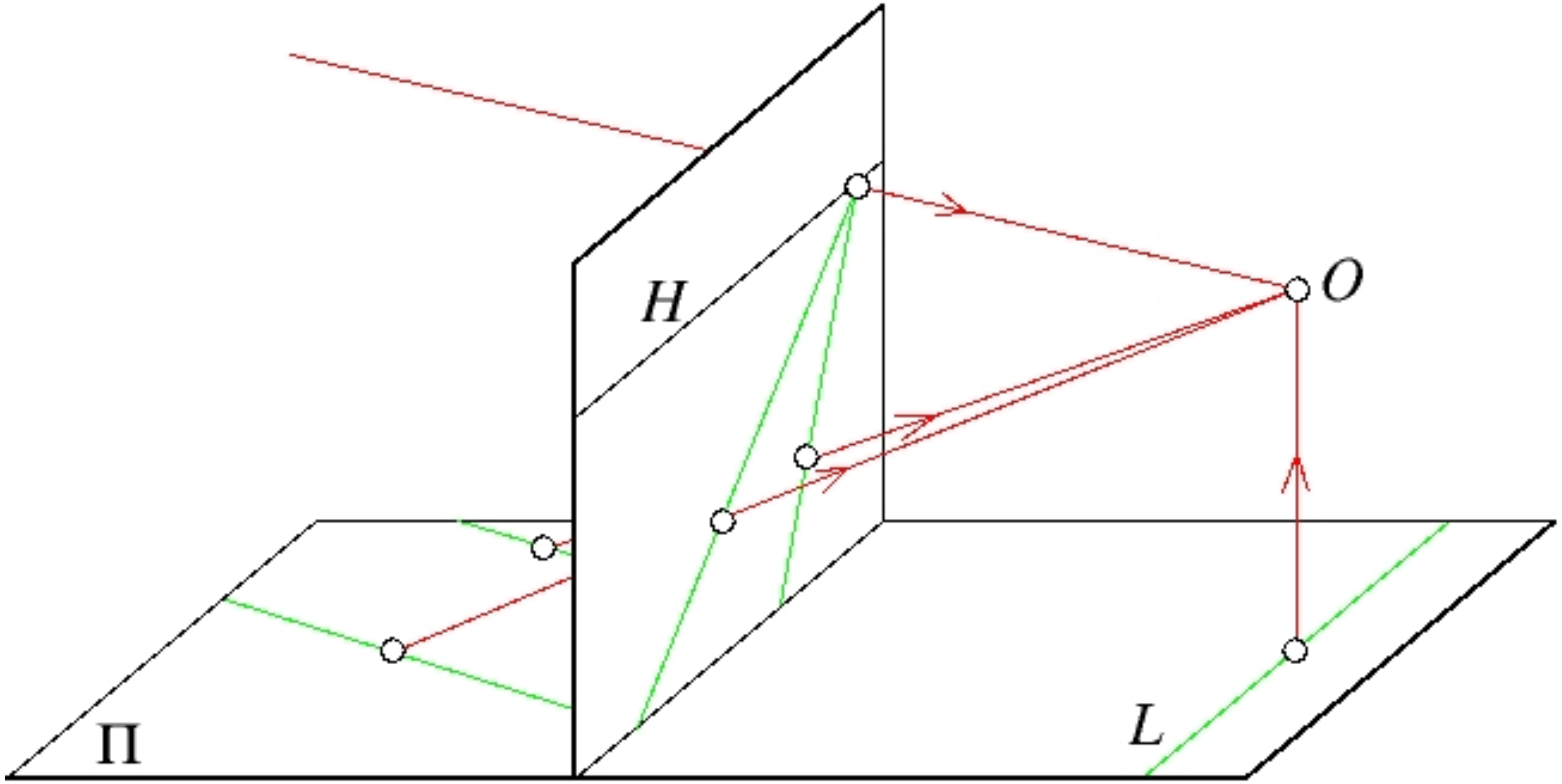
Perspective Effects



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

Perspective Effects

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



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

Vanishing Points

Each set of parallel lines meet at a different point

— the point is called **vanishing point**

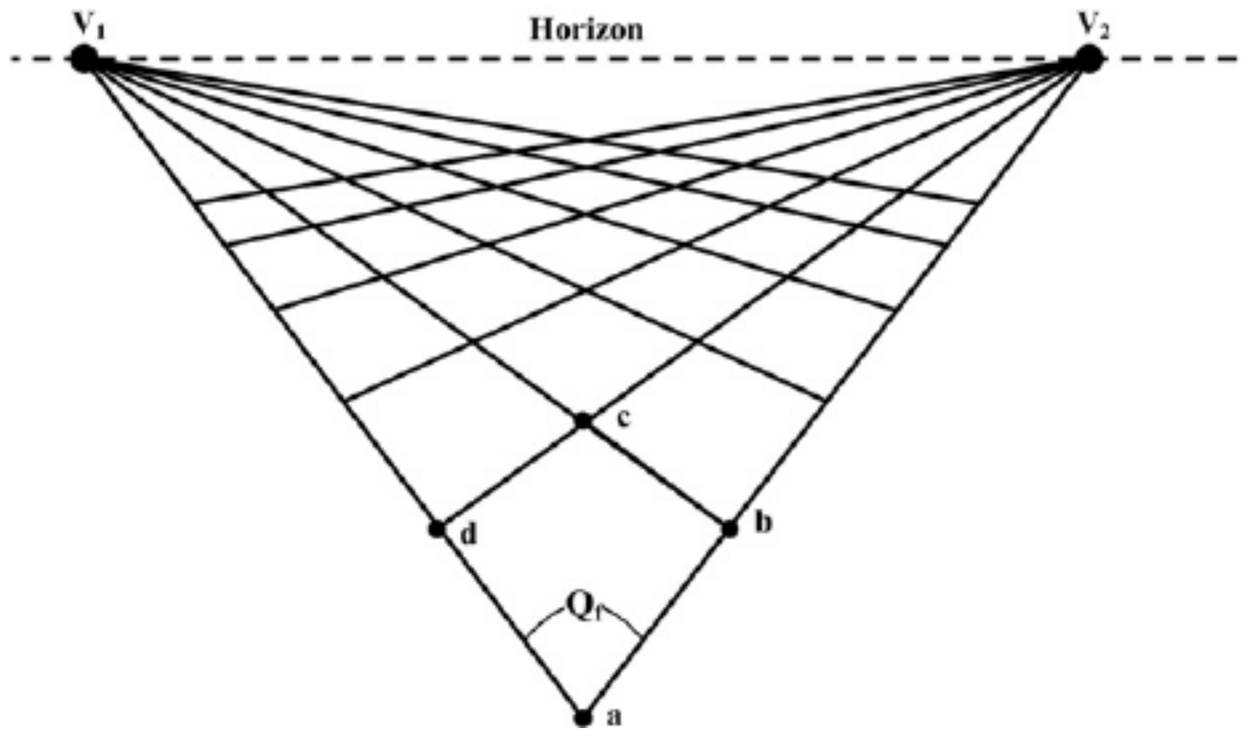
Vanishing Points

Each set of parallel lines meet at a different point

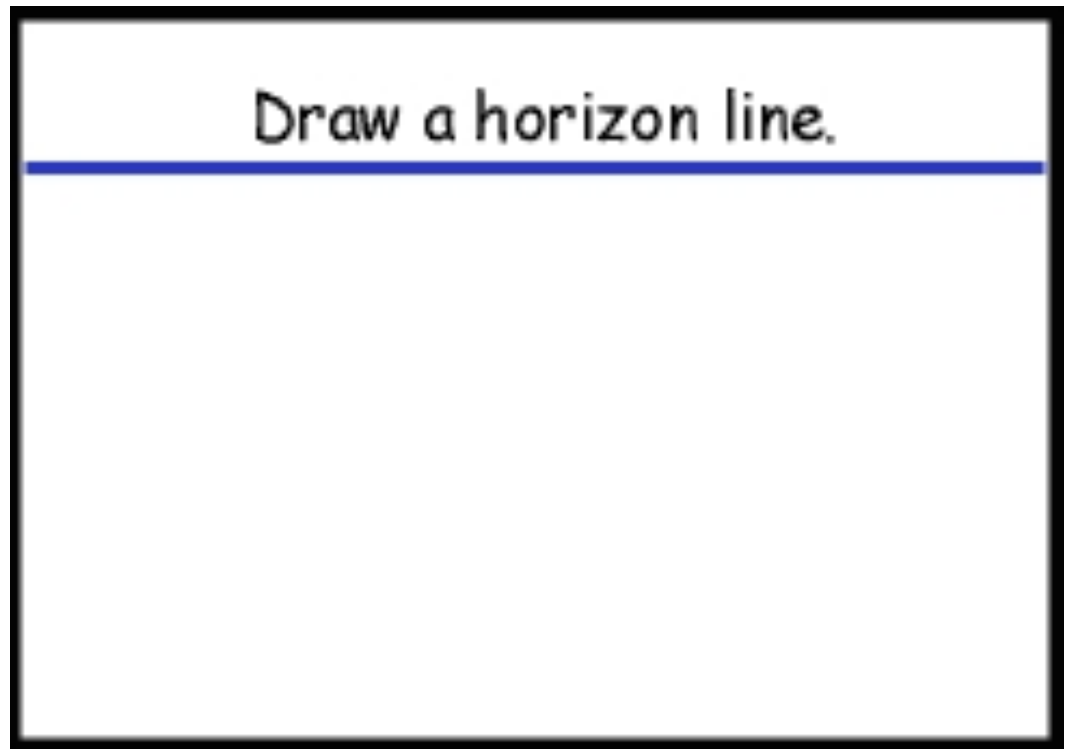
— the point is called **vanishing point**

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

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

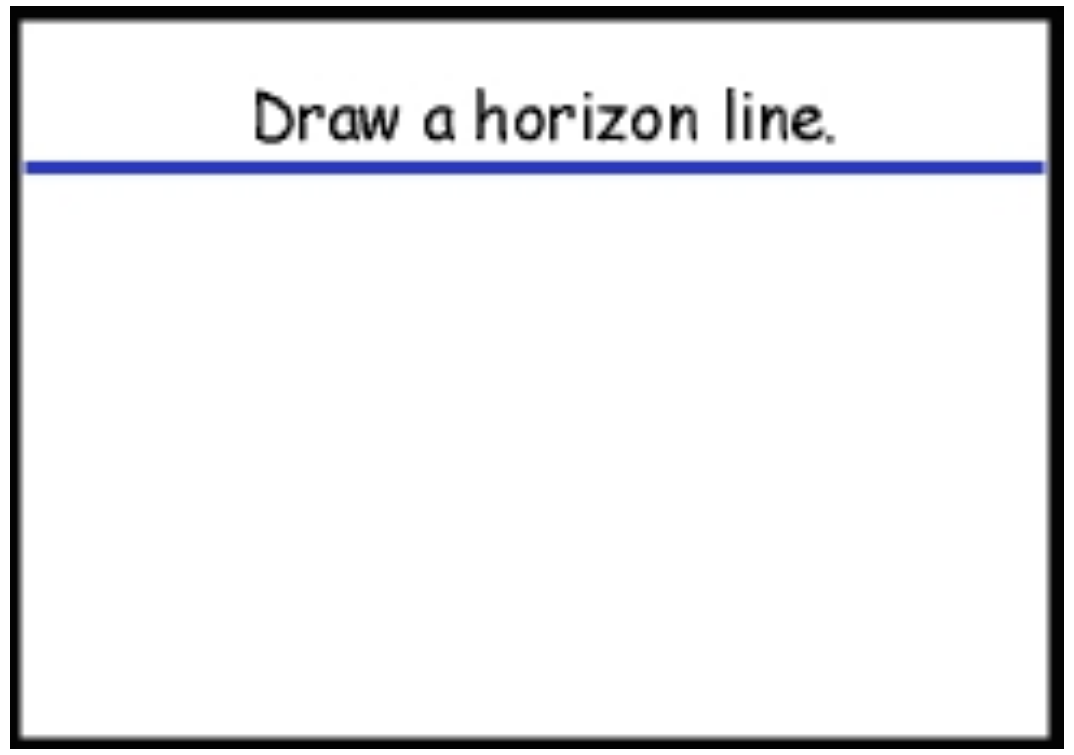


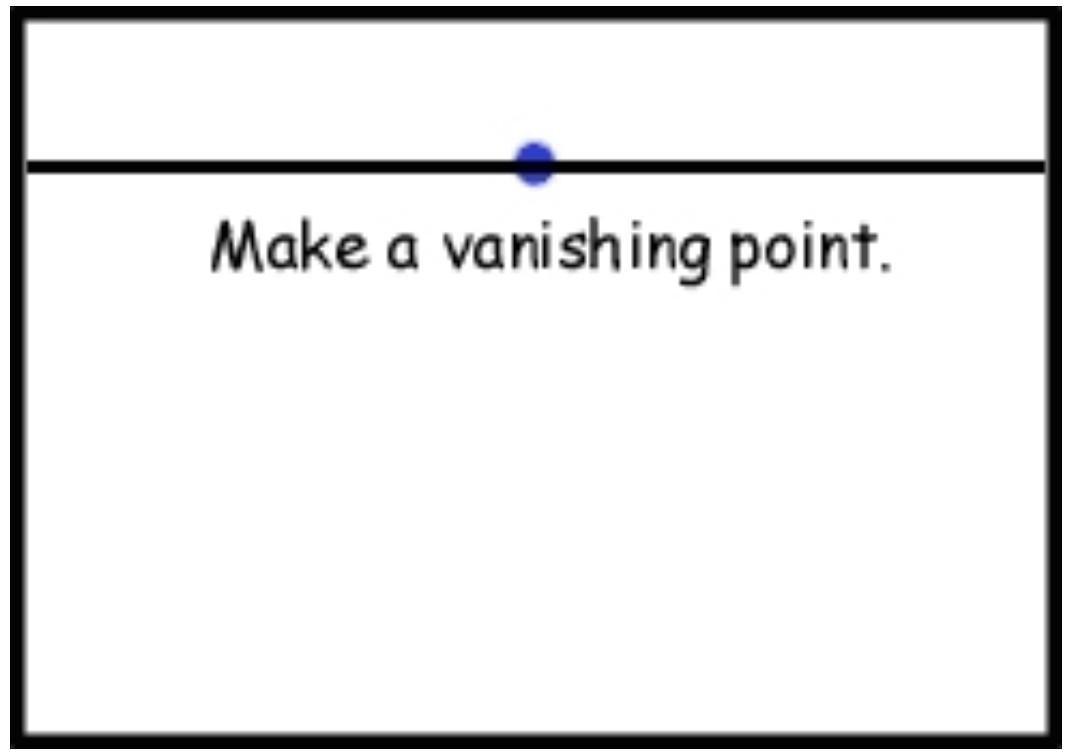
Vanishing Points



Slide Credit: David Jacobs

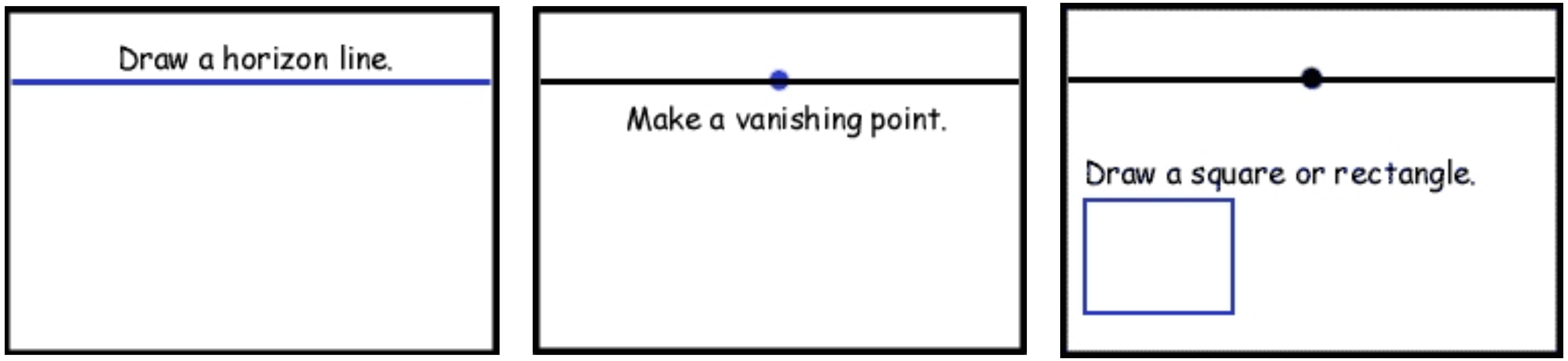
Vanishing Points





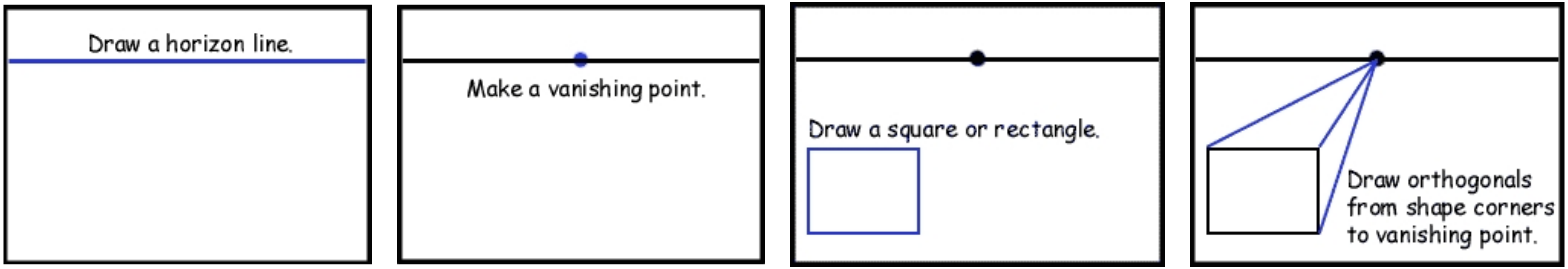
Slide Credit: David Jacobs

Vanishing Points



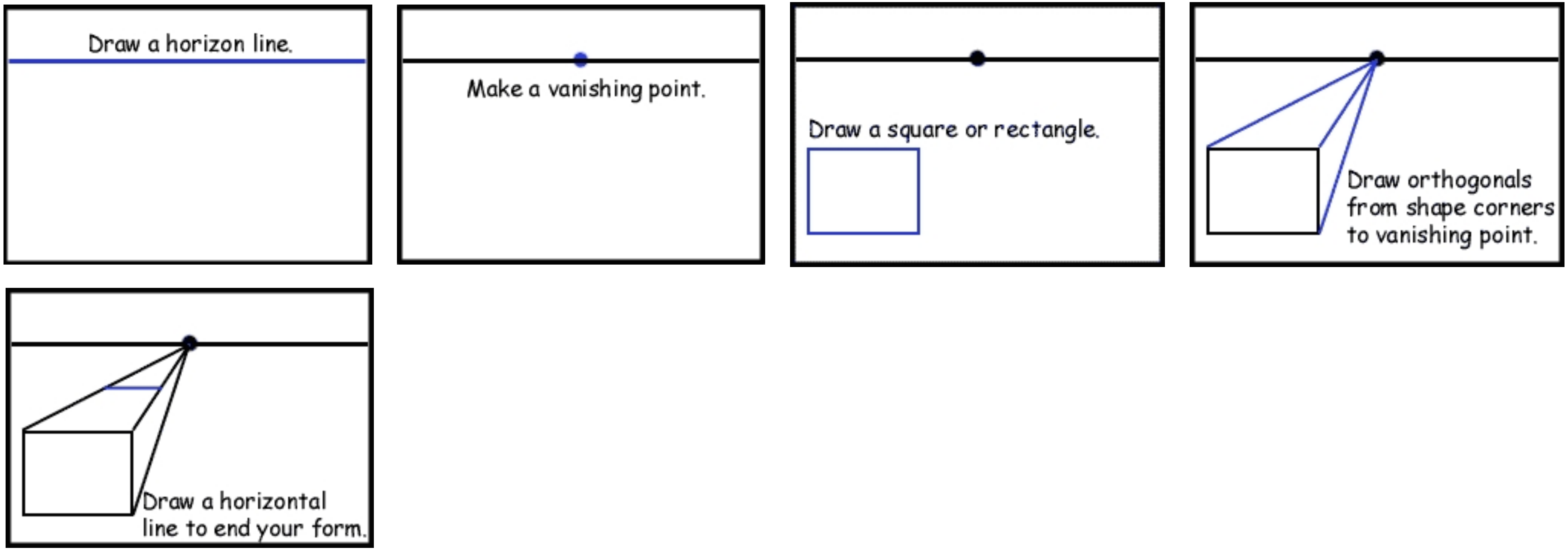
Slide Credit: David Jacobs

Vanishing Points



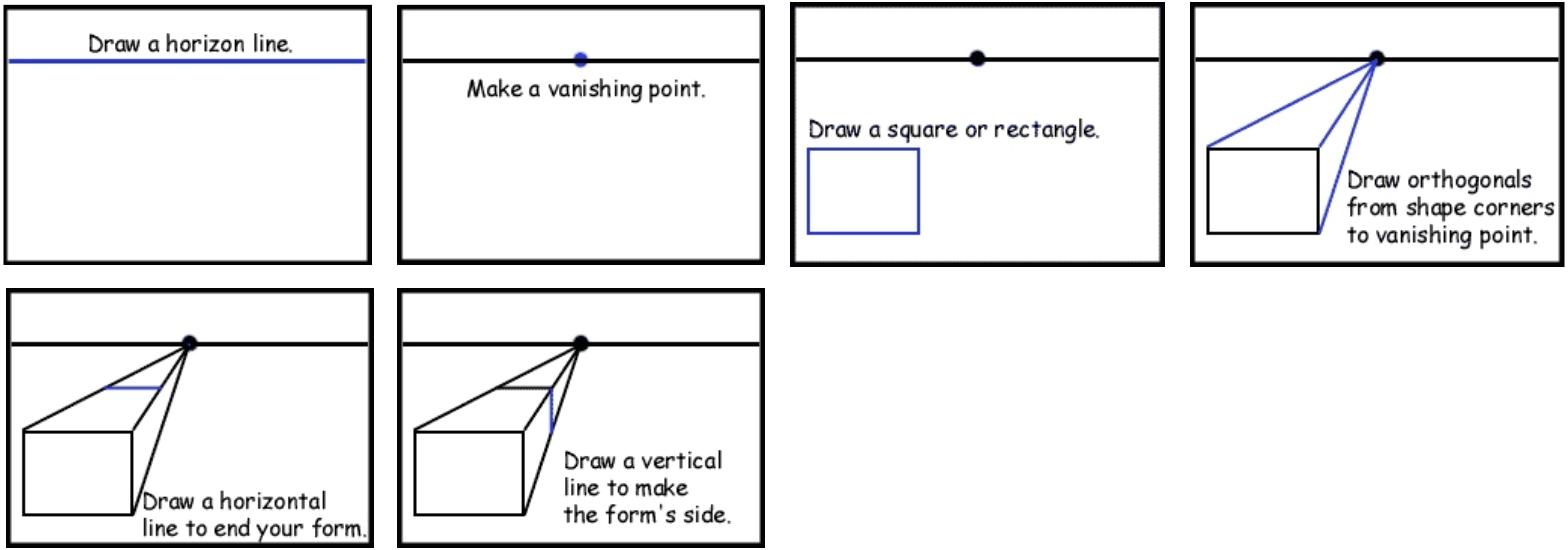
Slide Credit: David Jacobs

Vanishing Points



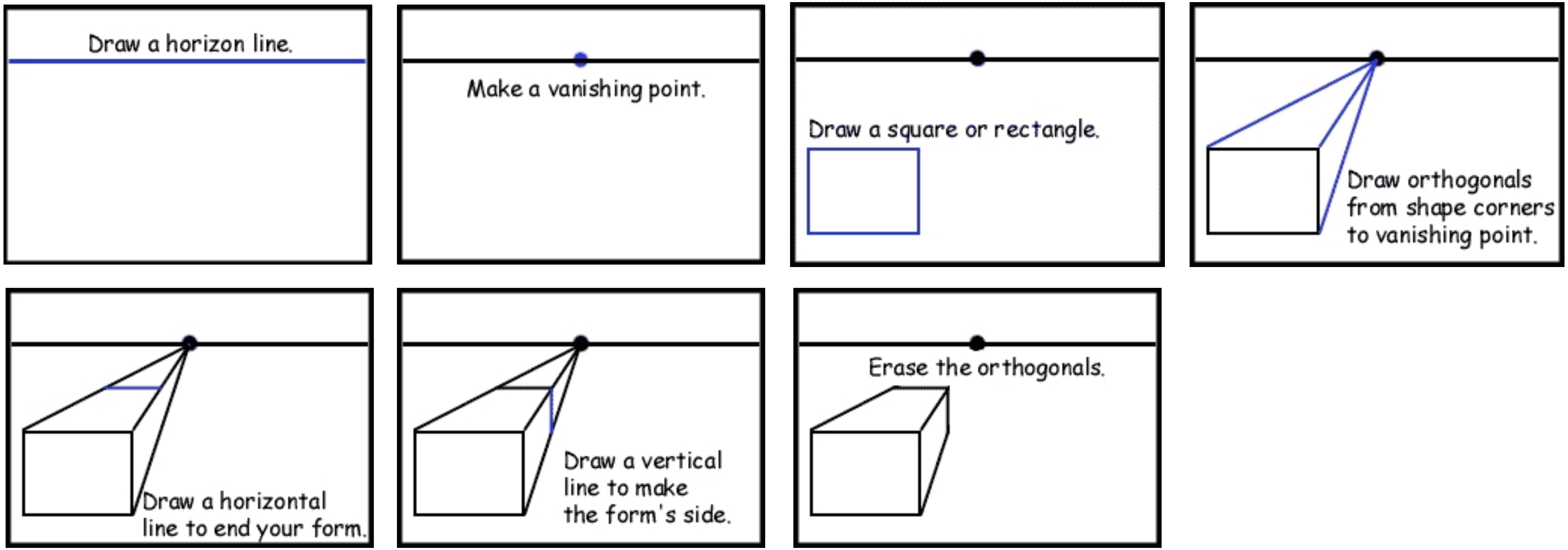
Slide Credit: David Jacobs

Vanishing Points



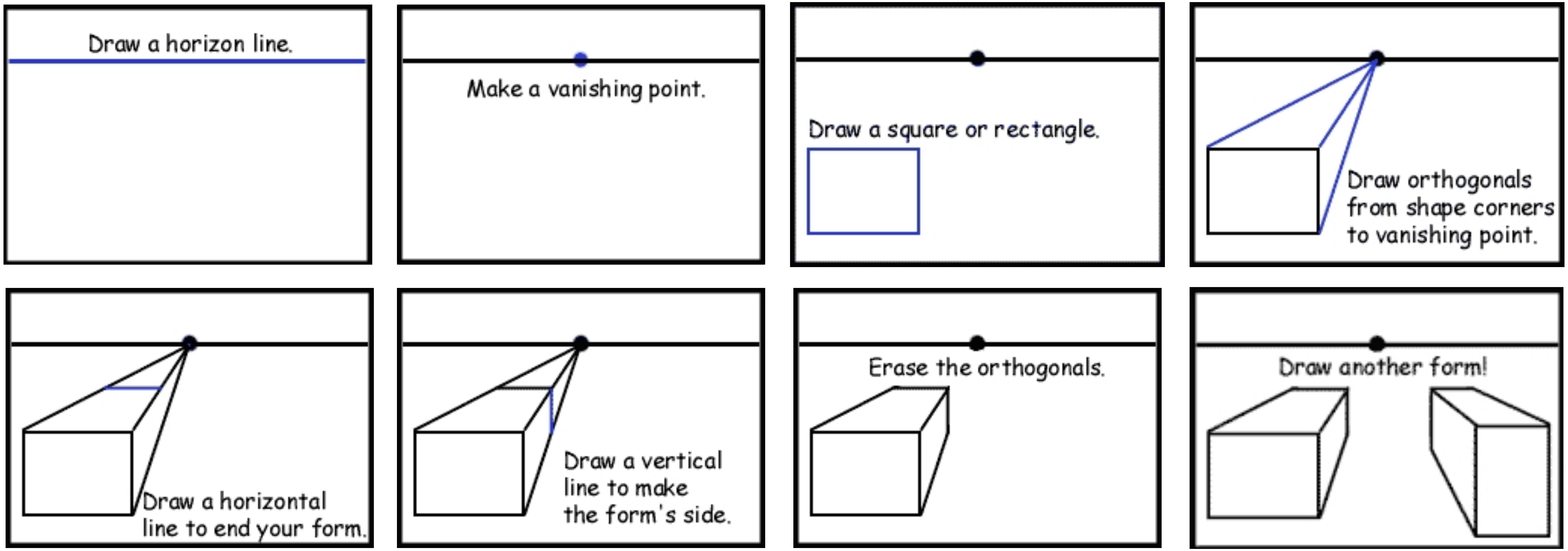
Slide Credit: David Jacobs

Vanishing Points



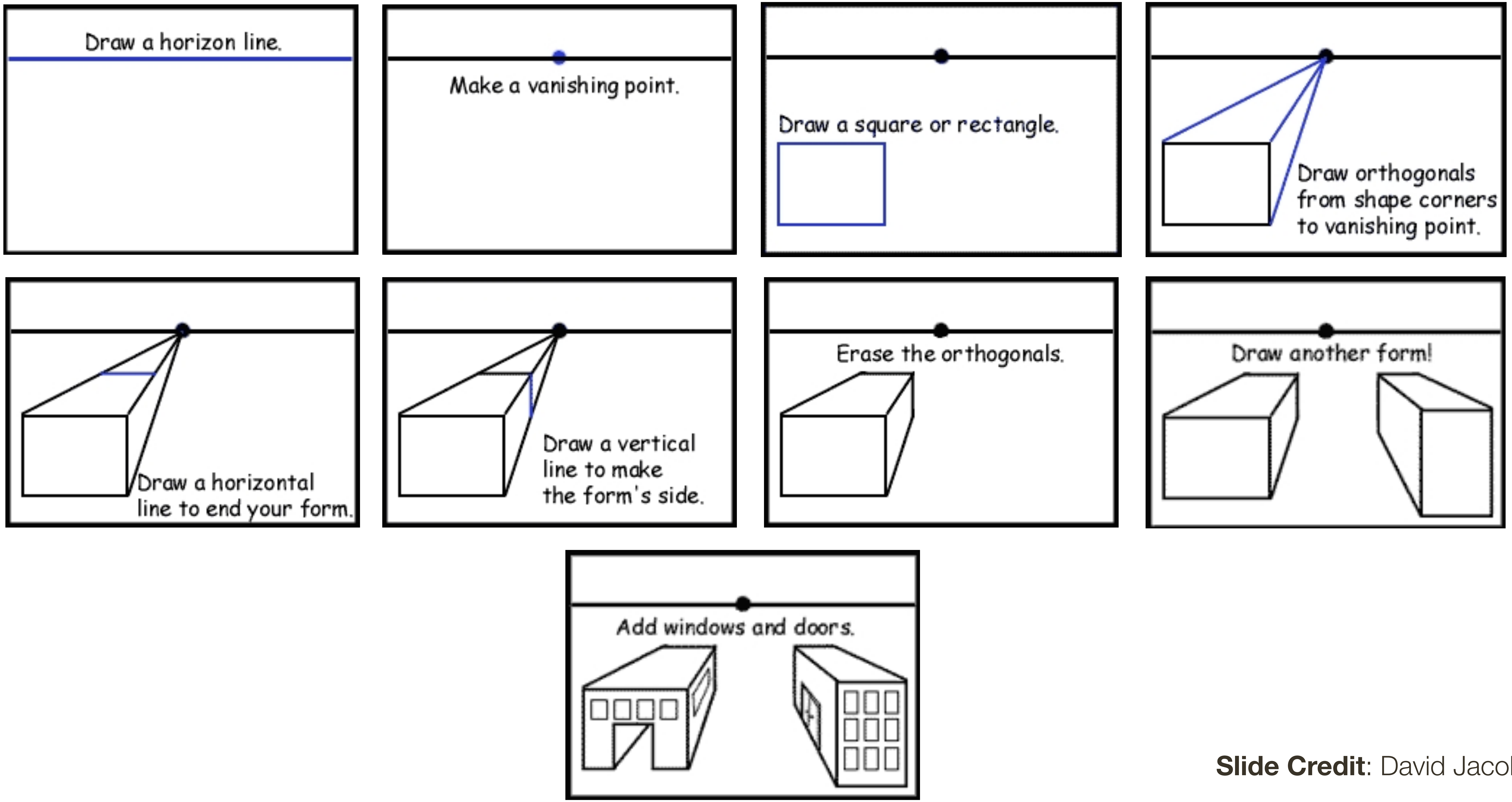
Slide Credit: David Jacobs

Vanishing Points



Slide Credit: David Jacobs

Vanishing Points



Slide Credit: David Jacobs

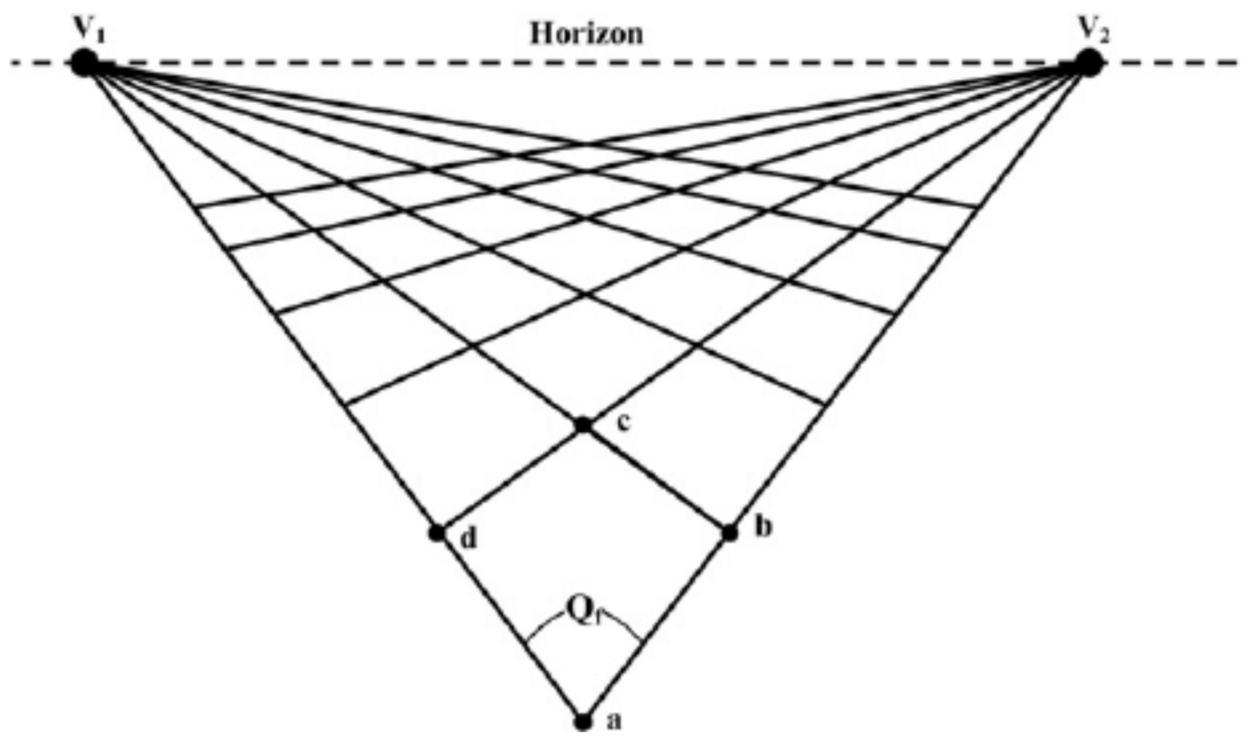
Vanishing Points

Each set of parallel lines meet at a different point

— the point is called **vanishing point**

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

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



Good way to **spot fake images**

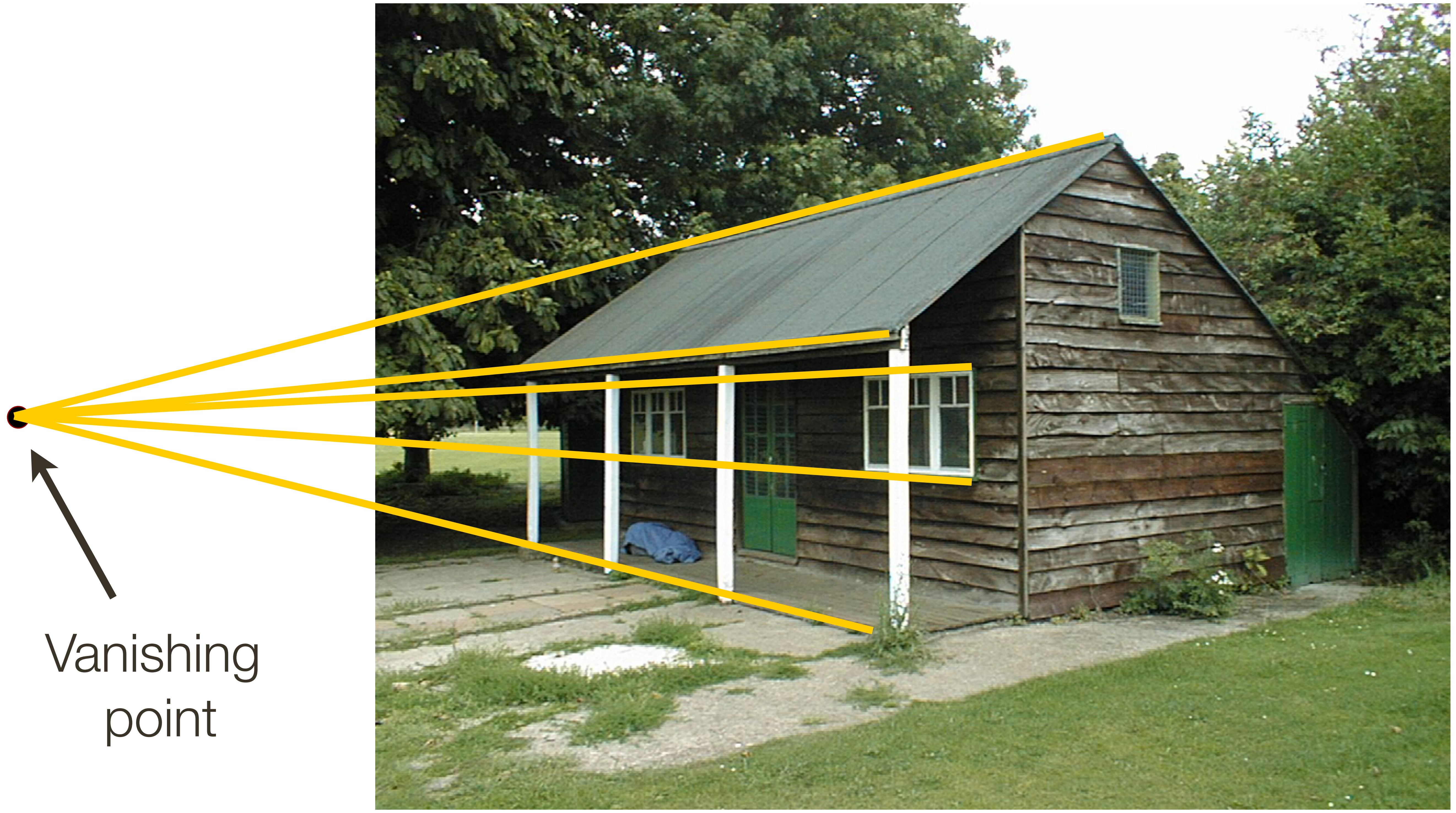
— scale and perspective do not work

— vanishing points behave badly

Vanishing Points

Slide Credit: Efros (Berkeley), photo from Criminisi

Vanishing Points

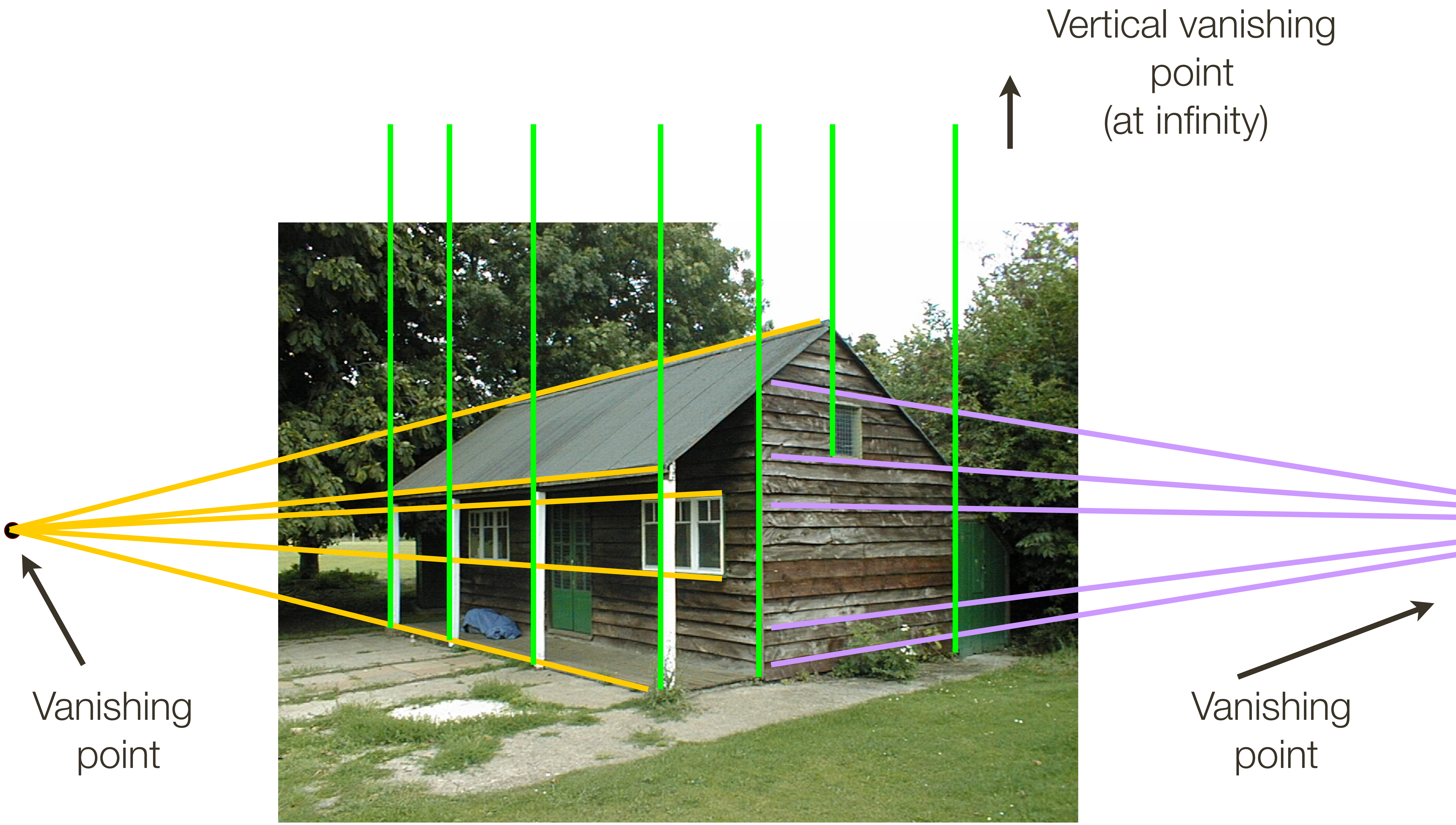


Slide Credit: Efros (Berkeley), photo from Criminisi

Vanishing Points

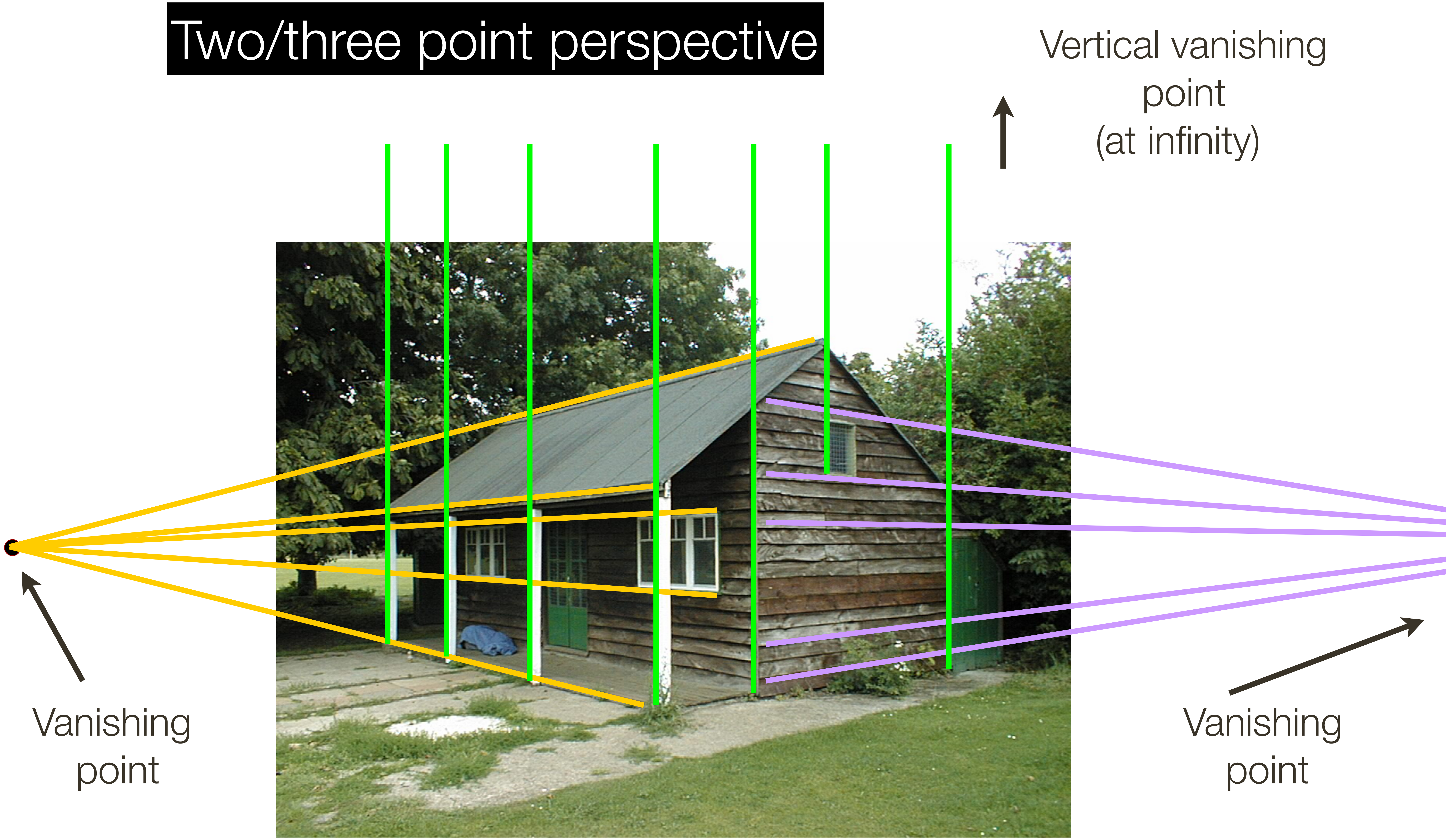
Slide Credit: Efros (Berkeley), photo from Criminisi

Vanishing Points



Slide Credit: Efros (Berkeley), photo from Criminisi

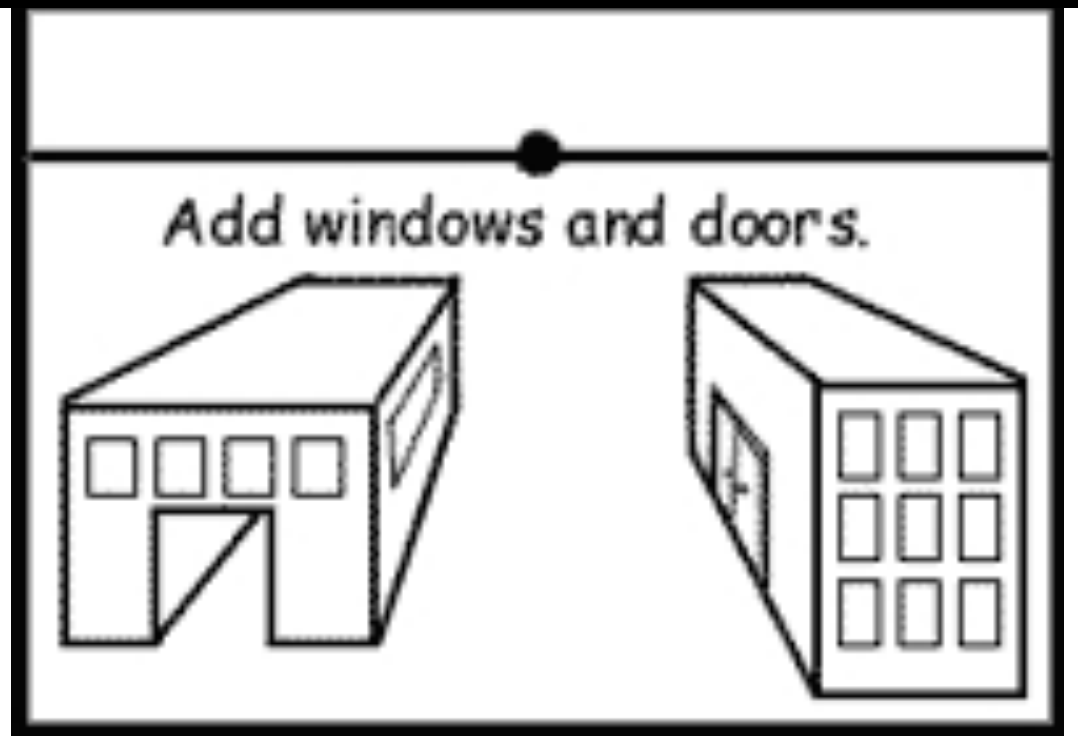
Vanishing Points



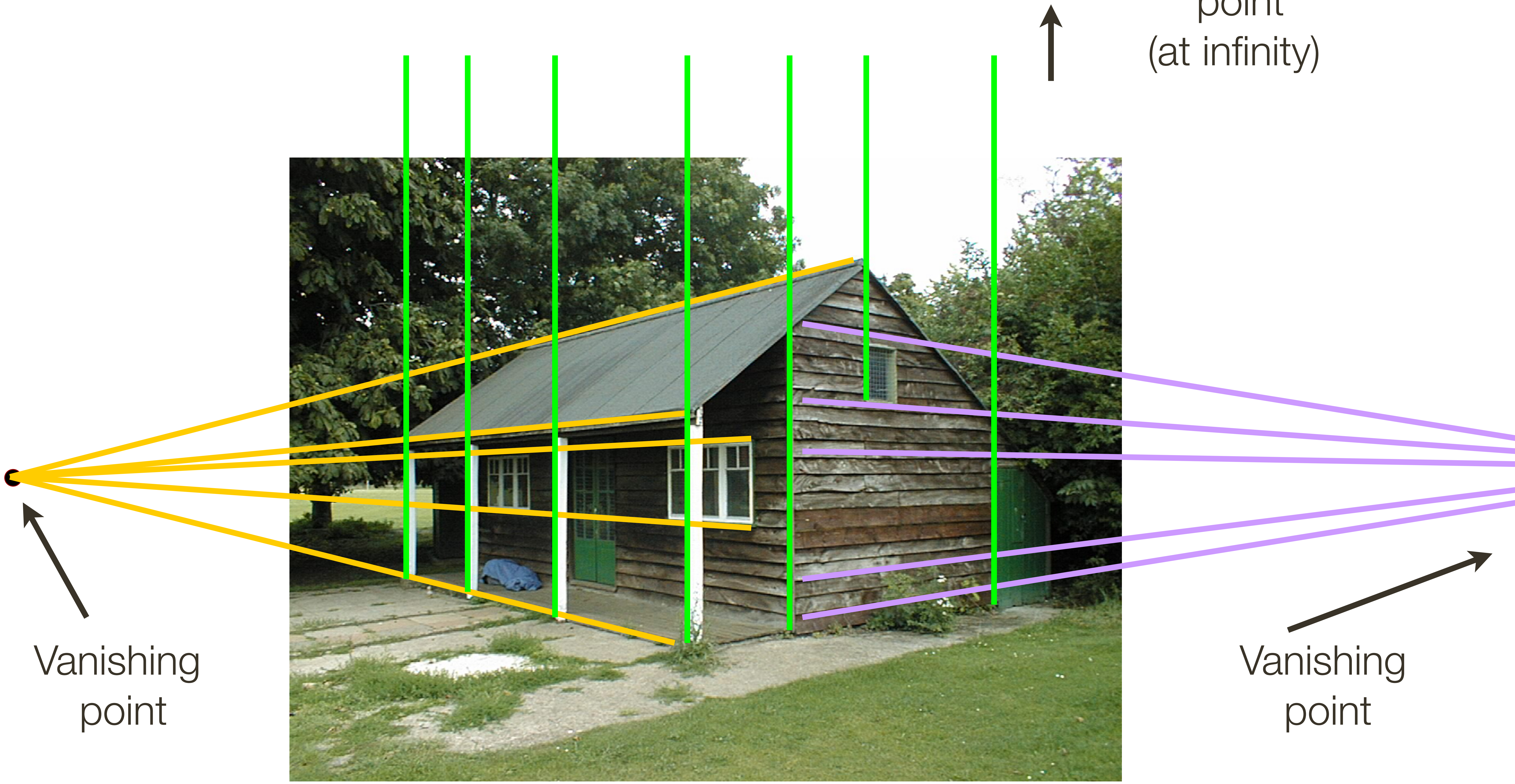
Slide Credit: Efros (Berkeley), photo from Criminisi

Vanishing Points

One point perspective



Two/three point perspective



Slide Credit: Efros (Berkeley), photo from Criminisi

Perspective Aside

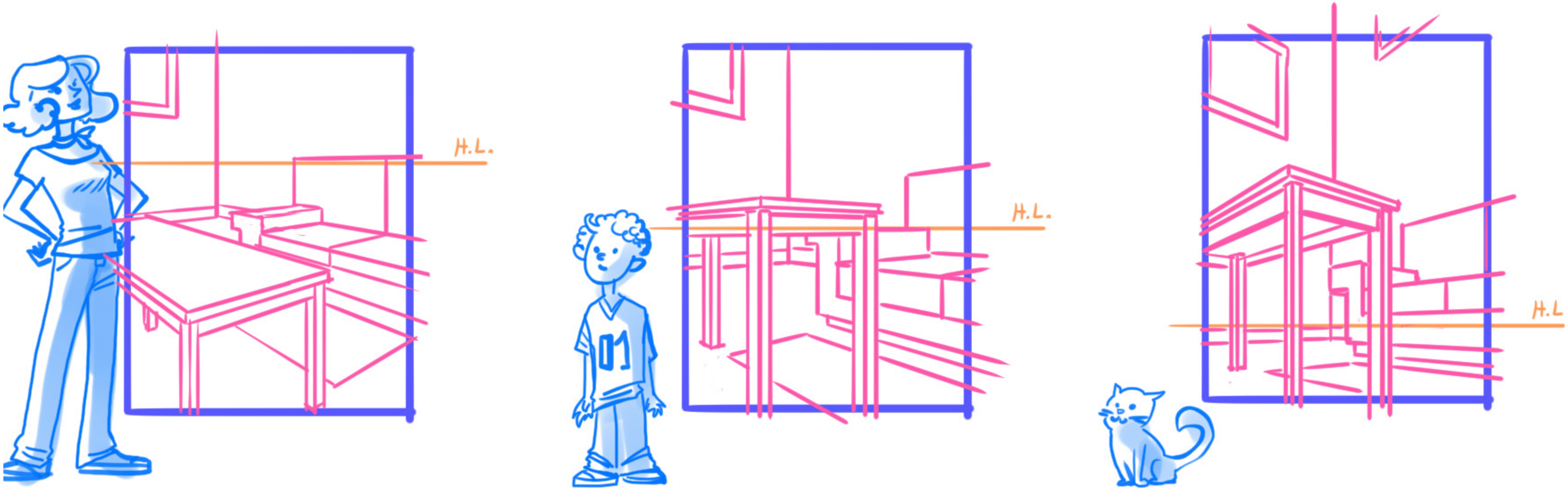
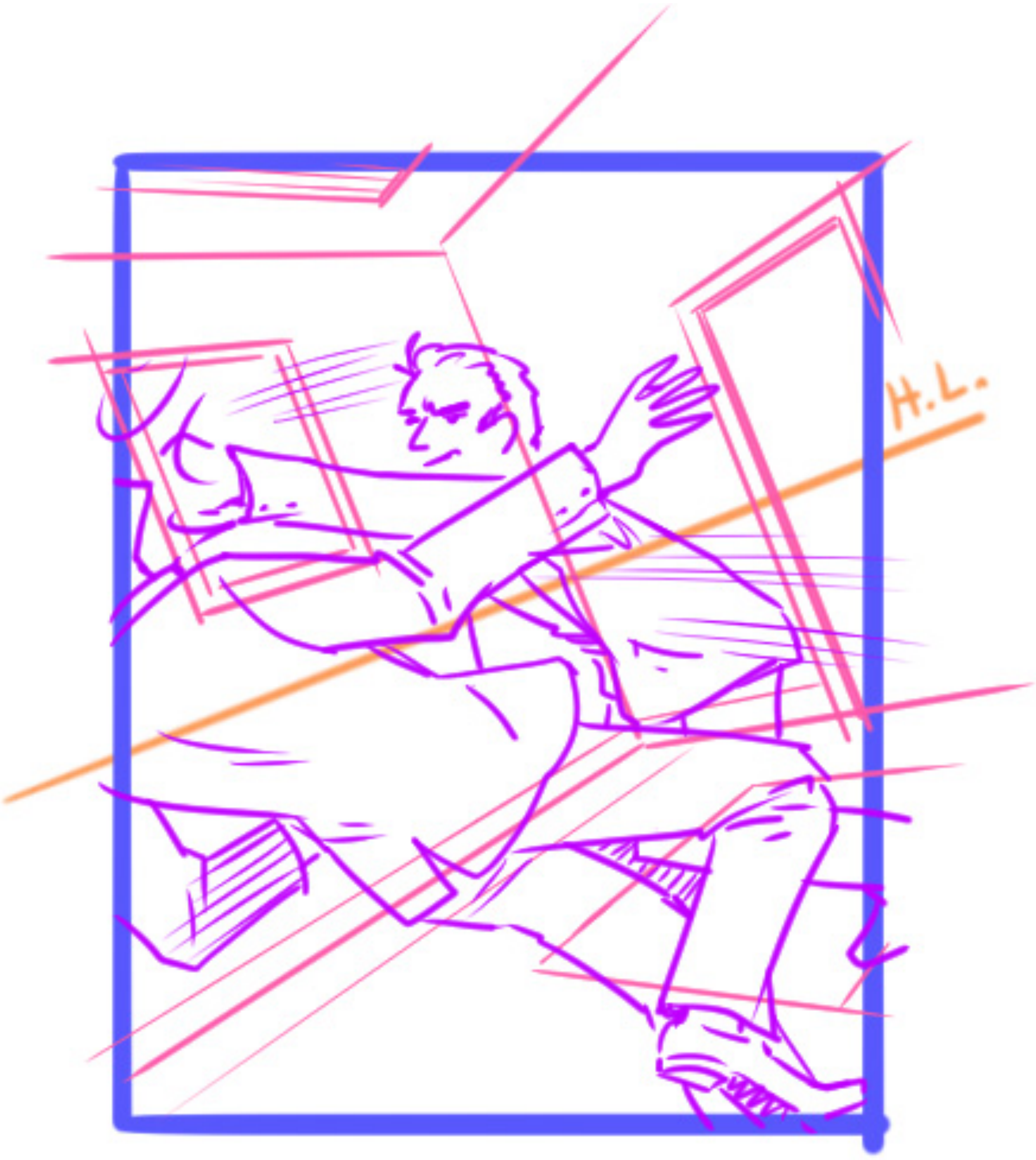


Image credit: http://www.martinacecilia.com/place-vanishing-points/

Perspective Aside



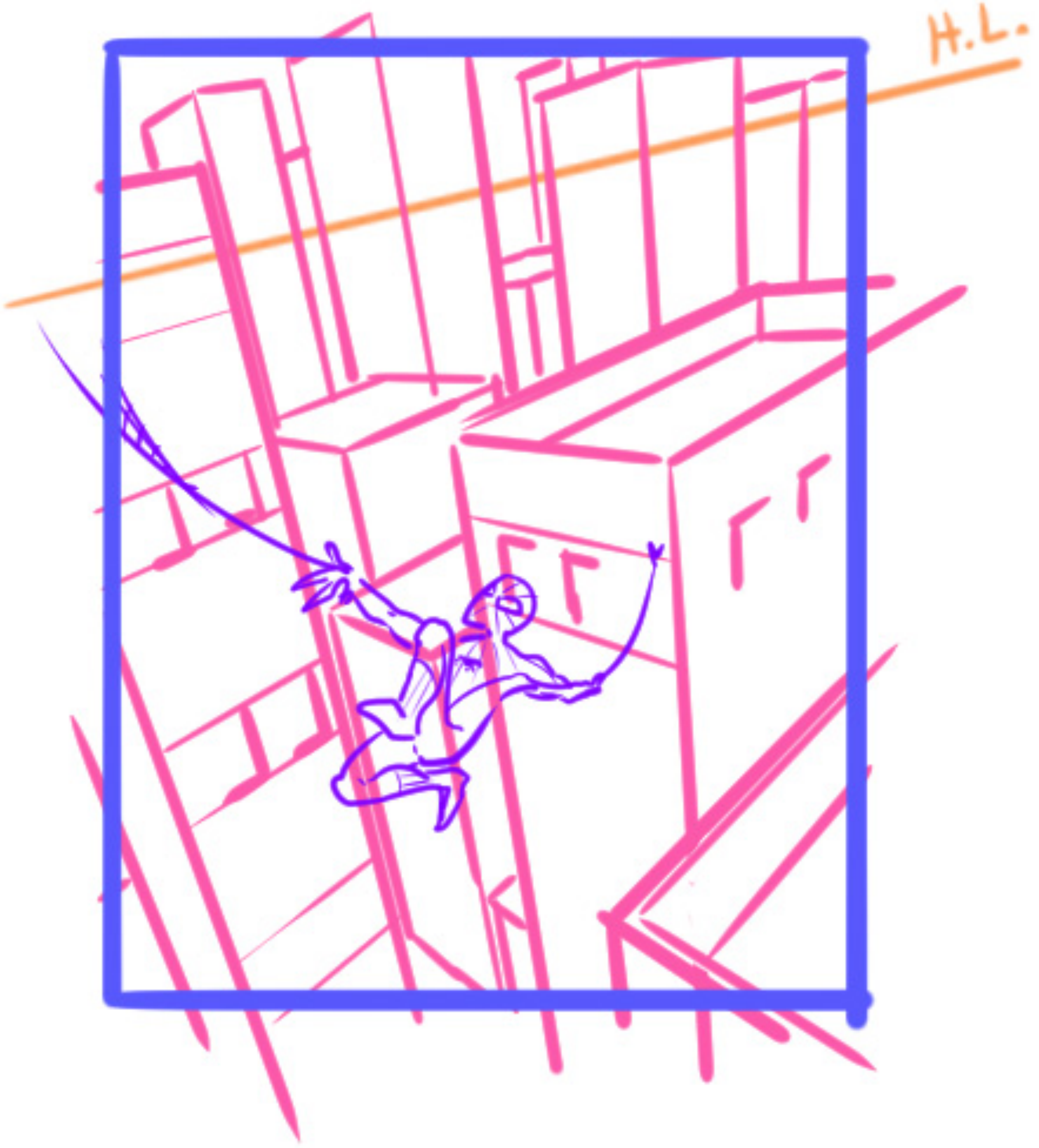


Image credit: http://www.martinacecilia.com/place-vanishing-points/

Properties of Projection

- **Points** project to **points**
- **Lines** project to **lines**
- **Planes** project to the **whole** or **half** image
- Angles are **not** preserved

Properties of Projection

— **Points** project to **points**

— **Lines** project to **lines**

— **Planes** project to the **whole** or **half** image

— Angles are **not** preserved

Degenerate cases

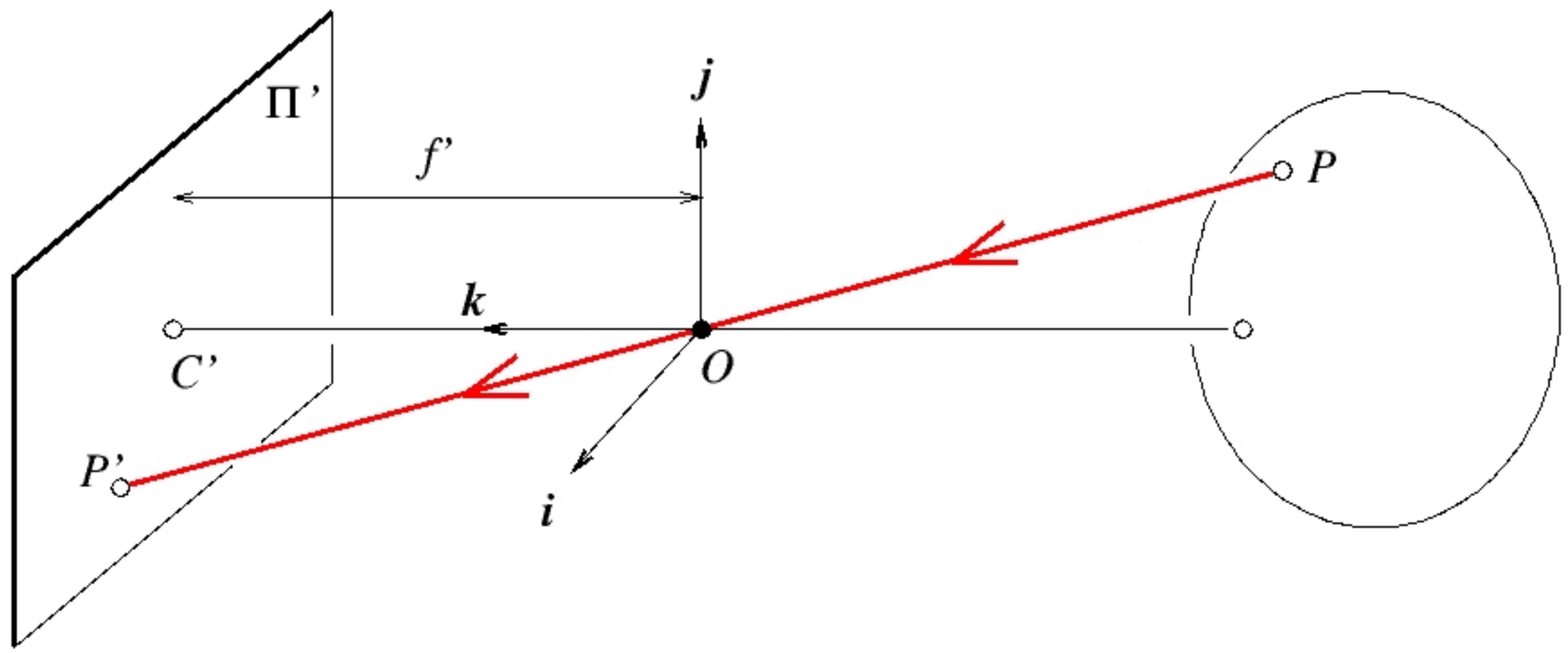
— Line through focal point projects to a point

— Plane through focal point projects to a line

Projection Illusion

Projection Illusion

Perspective Projection



3D object point

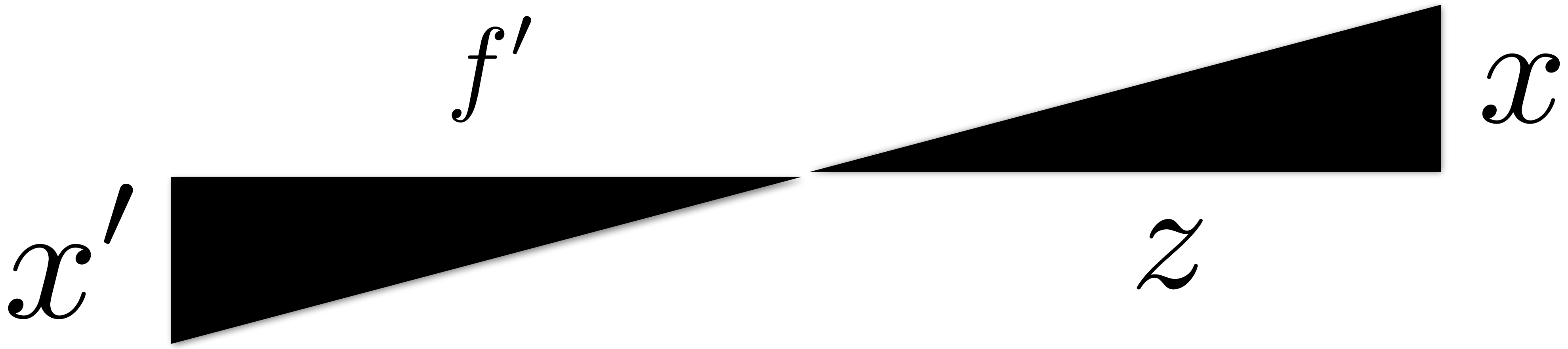
Forsyth & Ponce (1st ed.) Figure 1.4

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$ 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

Perspective Projection: Proof



3D object point

Forsyth & Ponce (1st ed.) Figure 1.4

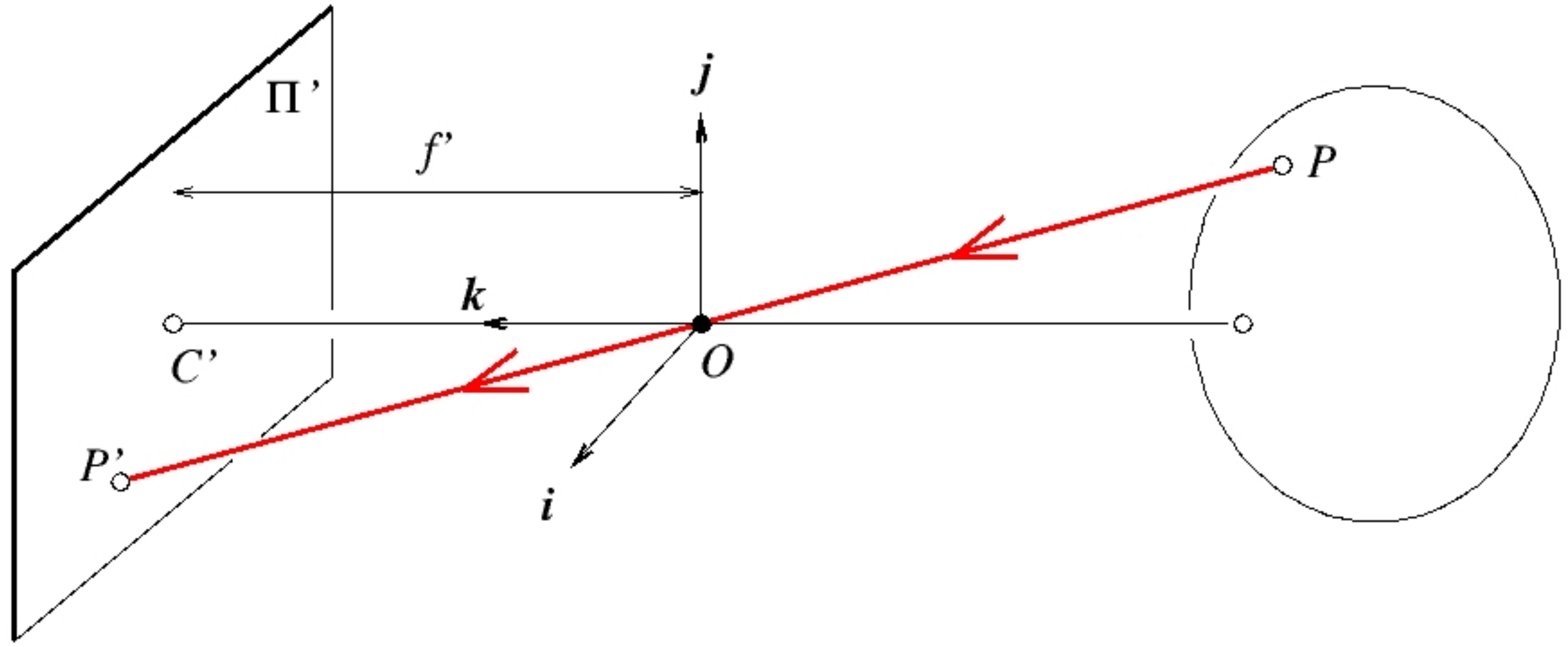
$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$ projects to 2D image point $$P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$ where

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

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

Aside: Camera Matrix

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 $\boxed{P' = \mathbf{C}P}$

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

Aside: Camera Matrix

$$x' = f'\frac{x}{z}$$
$$y' = f'\frac{y}{z}$$

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

$$x' = f'\frac{x}{z}$$
$$y' = f'\frac{y}{z}$$

Camera Matrix

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

$$\begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f'x \\ f'y \\ z \end{bmatrix} = \begin{bmatrix} \frac{f'x}{z} \\ \frac{f'y}{z} \\ 1 \end{bmatrix}$$

$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ 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

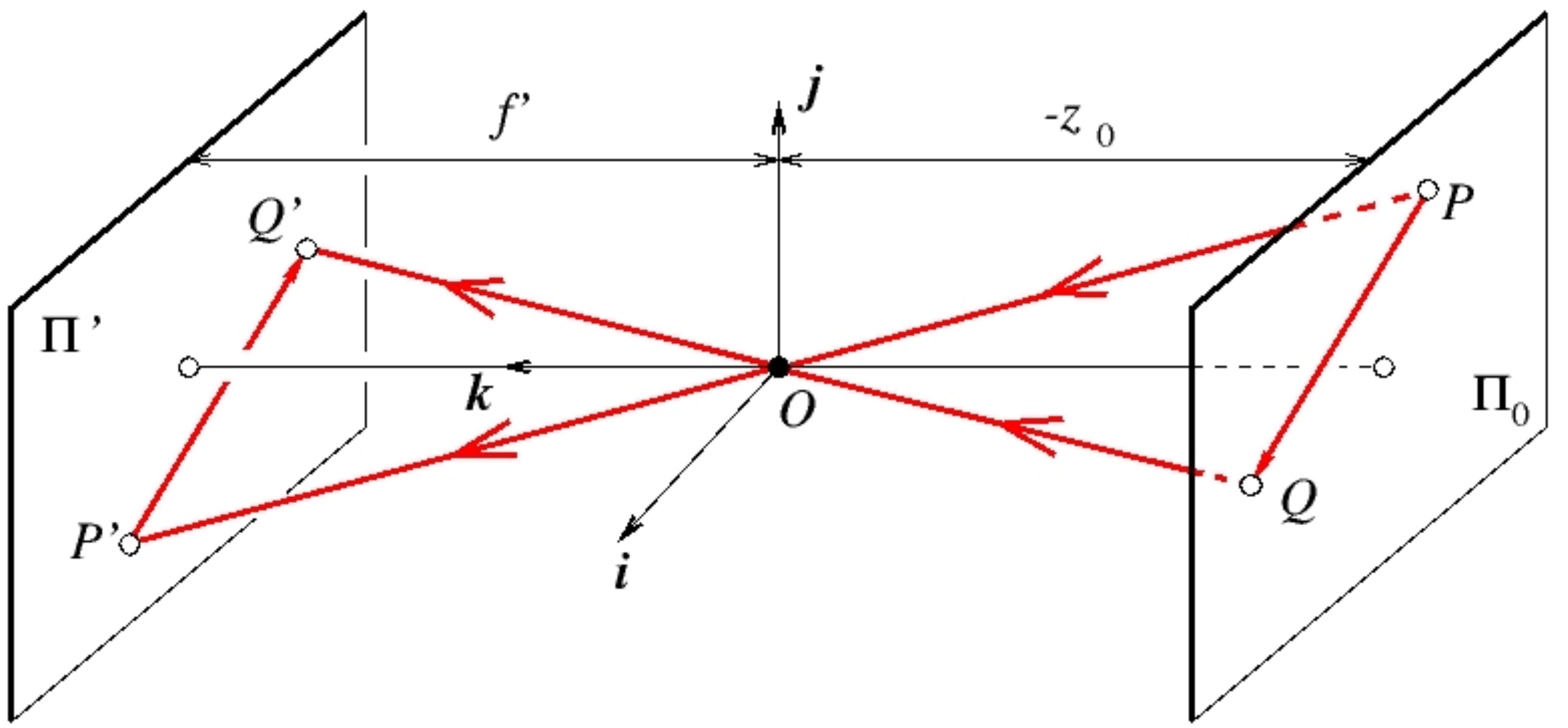
Camera Matrix

$$\mathbf{C} = \begin{bmatrix} f'_x & 0 & 0 & c_x \\ 0 & f'_y & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbb{R}_{4\times4}$$

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

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

Weak Perspective

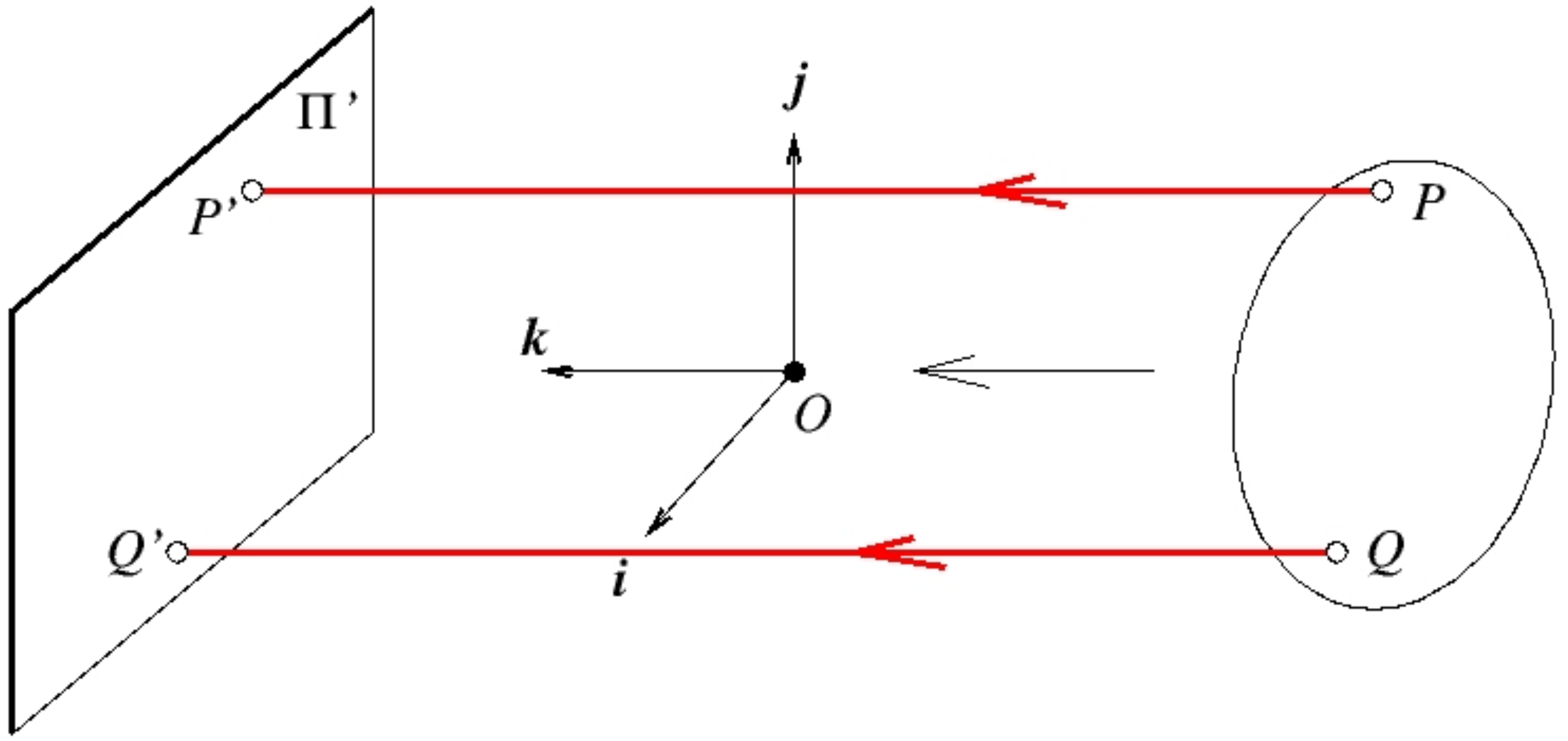


Forsyth & Ponce (1st ed.) Figure 1.5

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 $m = \dfrac{f'}{z_0}$

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

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)

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

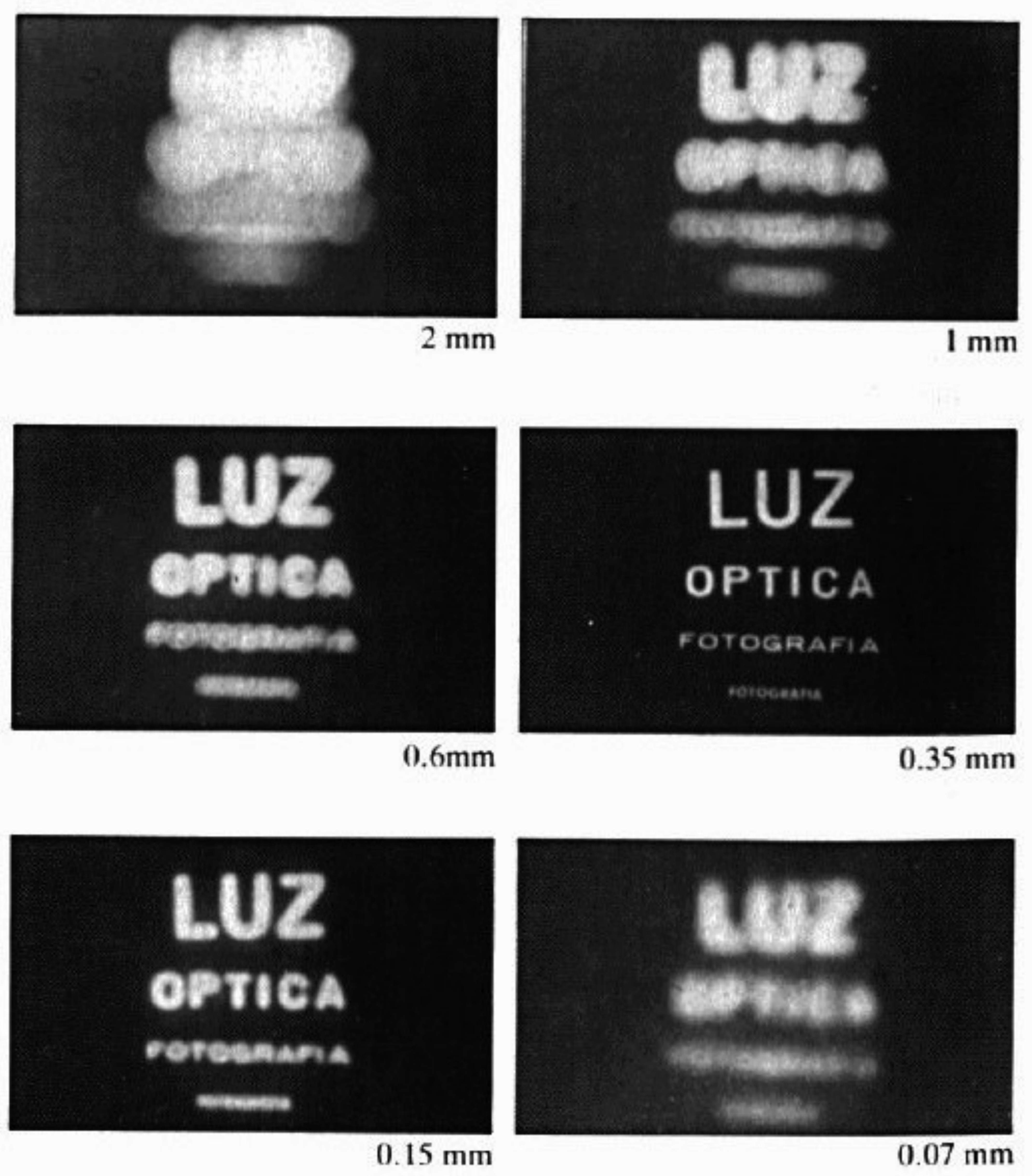
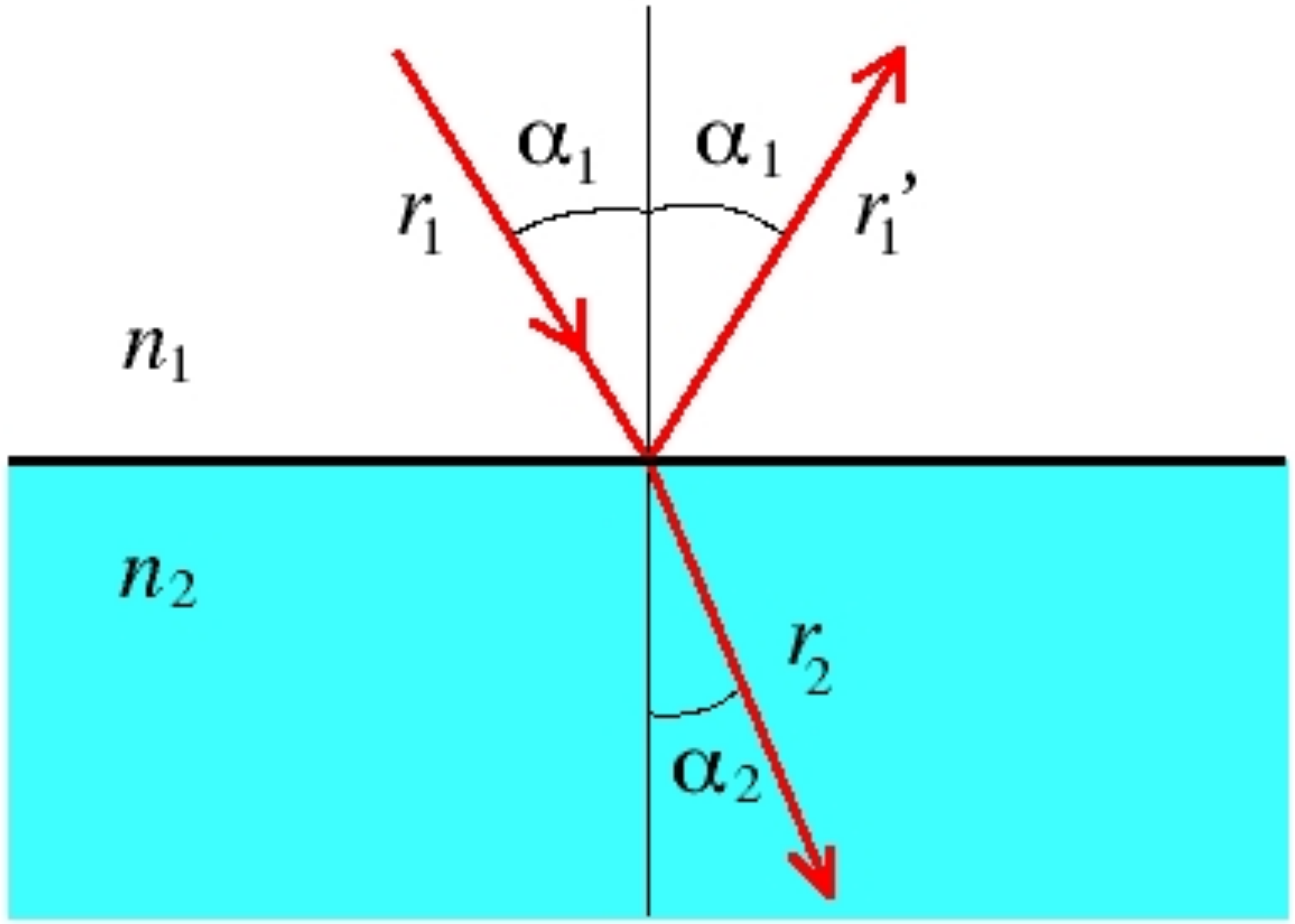


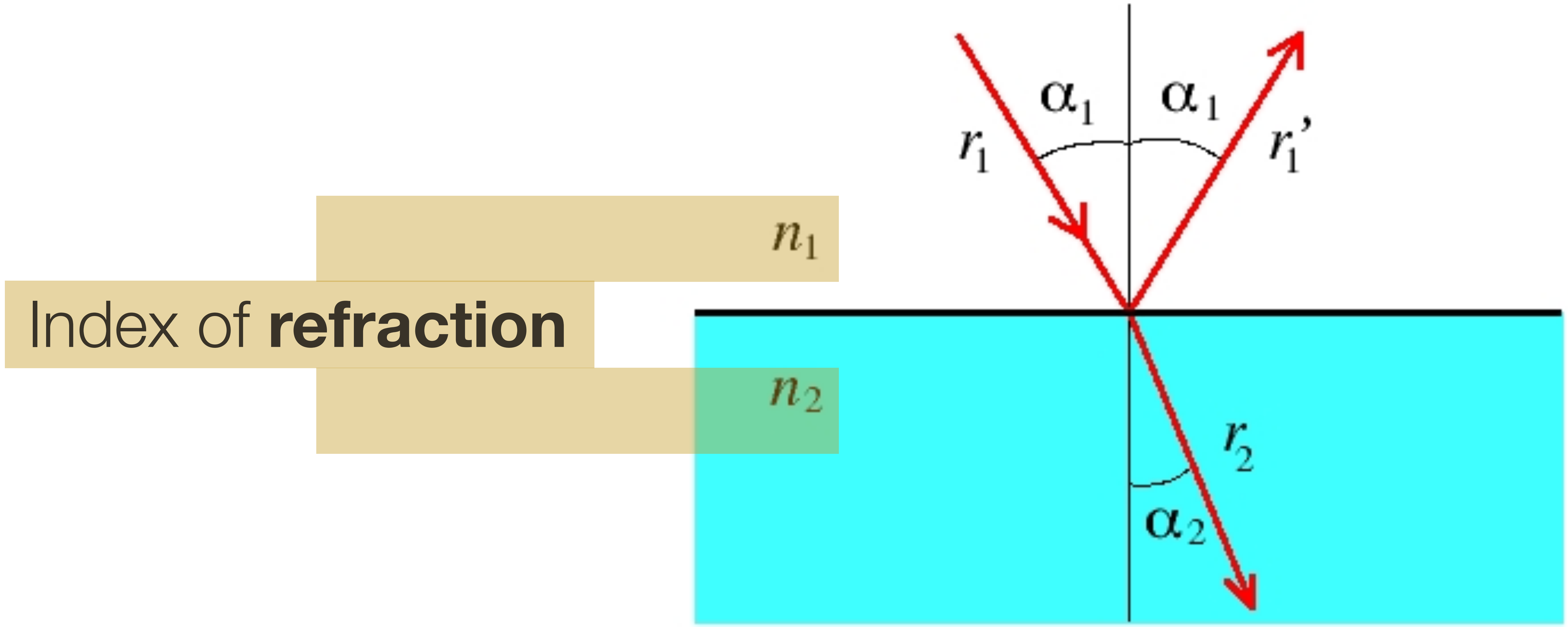
Image Credit: Credit: E. Hecht. “Optics,” Addison-Wesley, 1987

Snell's Law



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

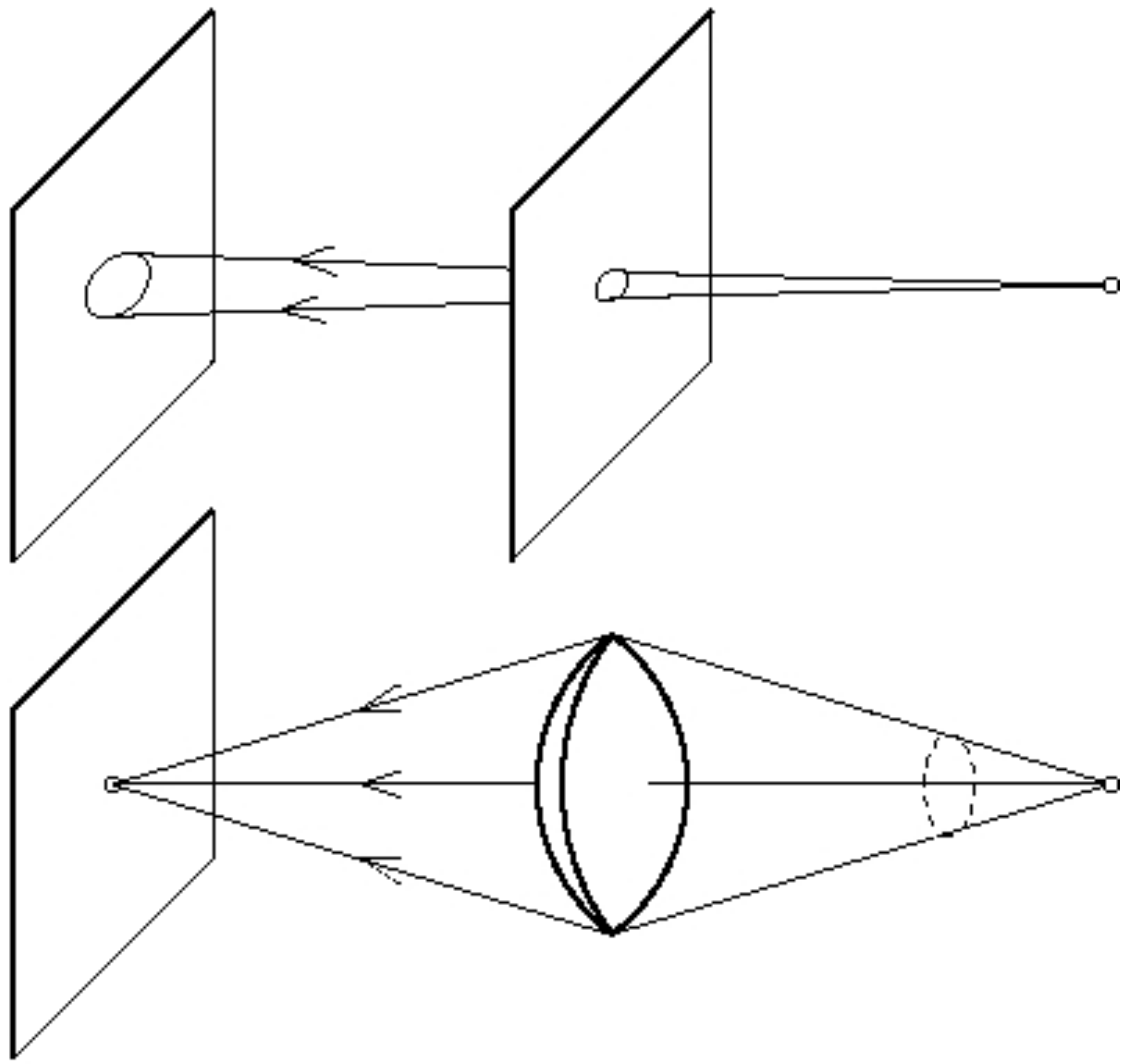
Snell's Law



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

Reason for **Lenses**

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



Reminders

Readings:

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

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

Reminders:

— Complete **Assignment 0** (ungraded) by Wednsday, **September 16**

— **WWW**: http://www.cs.ubc.ca/~lsigal/teaching.html

— **Piazza**: piazza.com/ubc.ca/winterterm22020/cpsc425201/home