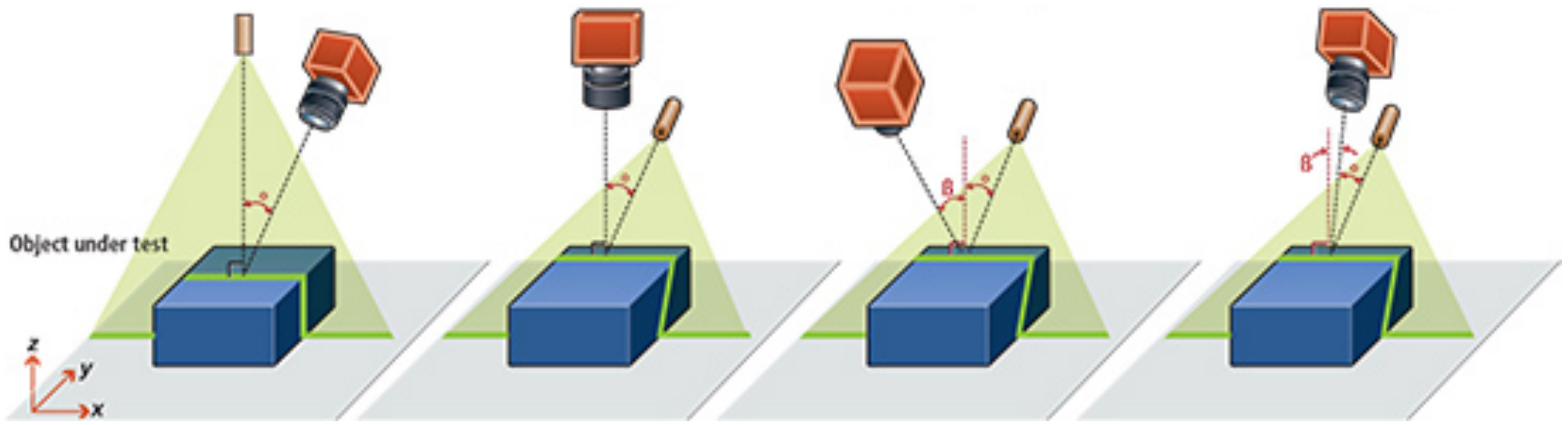
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 3: Image Formation (continued)

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

Menu for Today (**September 11, 2025**)

Topics:

- **Lenses**
- Human **eye** (as a camera)
- Image as a **function**
- **Linear filtering**

Readings:

- **Today's** Lecture: Szeliski Chapter 2, Forsyth & Ponce (2nd ed.) 1.1.1 — 1.1.3
  Szeliski 3.1-3.2, Forsyth & Ponce (2nd ed.) 4.1, 4.5

Reminders:

- Complete **Assignment 0** (ungraded) by Thursday, **September 11**
- **Assignment 1** (graded) is out Thursday, **September 11**
- **Lecture Notes** for Image Formation will be posted by next class

Today's "**fun**" Example #1: Nudging

Today's "**fun**" Example #1: Nudging

Aerial view of the white stripes at the lake shore drive in Chicago.

Today's "**fun**" Example #1: **Anchoring** and Ordering

Champagne, Sparkling, Rose, Sweet Wines

Champagne

CH18	NV	GREMILLET "Brut Selection" - Champagne	$65
CH31	NV	ERNEST RAPENEAU "Selection Brut" - Champagne	$65
CH12	NV	CHAMPAGNE ERNEST RAPENEAU - BRUT - Chardonnay/Pinot Noir/Pinot Meunier -	$75
CH05	NV	DRAPPIER "Carte d'Or" - Champagne	$78
CH30	2007	ERNEST RAPENEAU VINTAGE - Chardonnay/ Pinot Noir - Champagne	$80
CH32	NV	ERNEST RAPENEAU "Premier Cru Brut" - Champagne	$80
CH28	NV	DRAPPIER Brut Rose - Champagne	$85
CH29	2012	DRAPPIER "Millesime Exception" - Champagne	$98
CH11	2008	DRAPPIER " Cuvee Grande Sendree" - Champagne	$130
CH39	NV	ERNEST RAPENEAU "Grande Reserve"- Magnum - Champagne	$130

Sparkling Wines

CH06	NV	IL CORTIGIANO - Prosecco Extra Dry - Veneto	$30
CH17	NV	VALLFORMOSA "Clasic" Semi Seco - Cava	$30
CH24	NV	VEUVE MOISANS "Blanc de Blancs" - Loire Valley	$30
CH25	NV	VALDO - Prosecco Extra Dry - Treviso, Veneto	$30
CH33	NV	VALDO "Origine" Rose - Veneto	$30
CH03	2012	CHATEAU MONTGUERET Saumur Sec Rose - Cabernet Franc - Loire Valley	$32
CH04	NV	CAVA MASET RESERVA BRUT - Macabeo/Xarello/Parellada - Cava	$32
CH14	NV	TRIVENTO "Brut Nature" - Mendoza	$32
CH21	2015	CAMASELLA - Glera - Vaneto	$32
CH02	2013	BRUT D'ARGENT ICE - Chardonnay - France	$35
CH01	NV	VALDO "ORO PURO" Prosecco Superiore - Veneto	$36
CH40	NV	MAISON DARRAGON - AOC Vouvray Brut - Loire Valley	$38
CH09	NV	LOU MIRANDA ESTATE 'LEONE' - Sparkling Shiraz - Barossa Valley	$42

Rose Wines

PO03	2014	CASAL MENDES Rose - Baga - Portugal	$30
RH09	2014	LA VIE EN ROSE - Cinsault - Languedoc	$30
RH69	2015	LES EMBRUNS "La Croix des Saintes" - Sable de Camargue	$30
RH04	2015	LES MAITRES VIGNERONS DE ST TROPEZ - Cotes de Provence	$32
RH15	2015	MANON - COTES DE PROVENCE - Grenache/Cinsault/Syrah. - Provence	$34
RH04M	2015	LES MAITRES VIGNERONS DE LA PRESQU'ILE DE SAINT TROPEZ - Grenache/Mourve	$68

Sweet Wines

AR33	2015	TRIVENTO "Birds & Bees" White - Mendoza	$30
AR34	2016	TRIVENTO "Birds & Bees" Red - Mendoza	$30
AU05	2015	DEAKIN ESTATE - Moscato - Murray Darling	$30
AU12	2016	Chalk Hill - Moscato - McLaren Vale	$30
AU68	NV	WESTEND ESTATE "Richland" - Moscato - New South Wales	$30
AU107	NV	WESTEND ESTATE "Richland" - Pink Moscato - New South Wales	$30

Short Review of **Lecture 2**

Lecture 2: Re-cap Image Formation

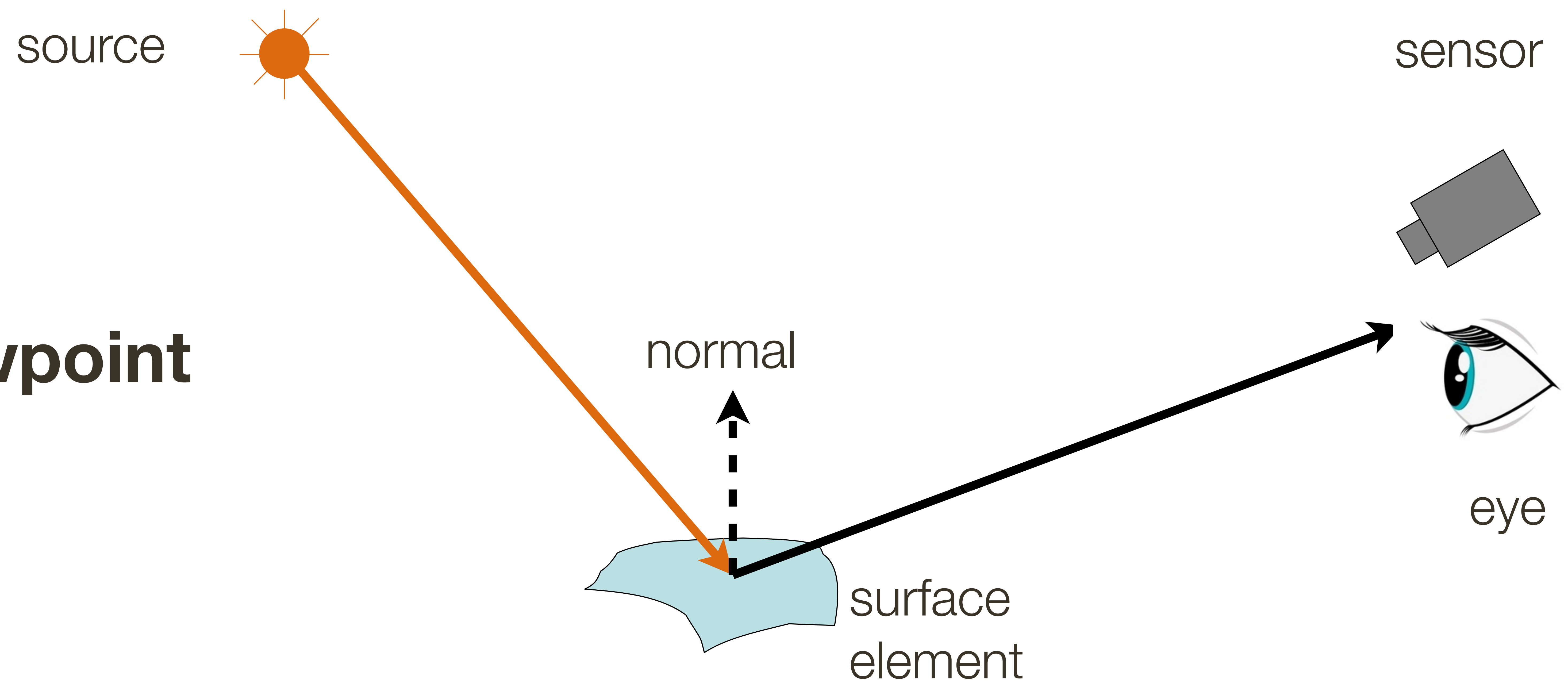
The **image formation process** that produces a particular image depends on

— **Lightening** condition

— Scene **geometry**

— **Surface** properties

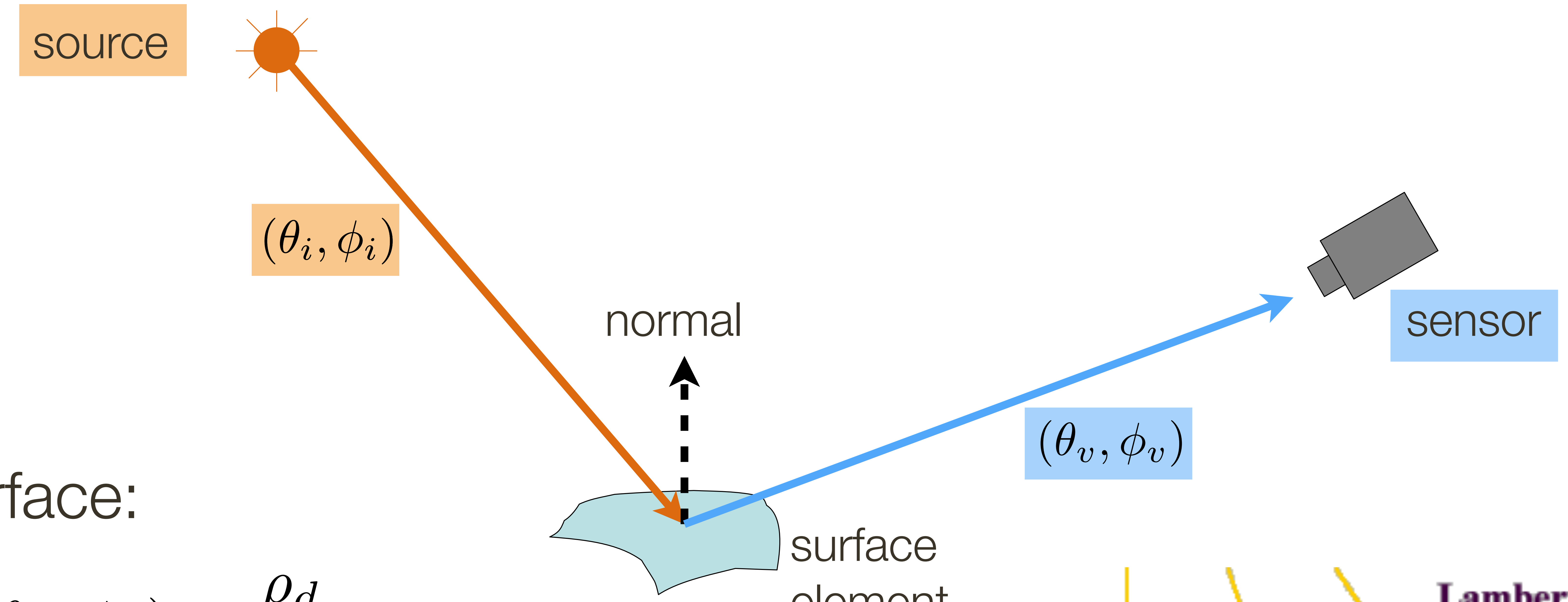
— Camera **optics** and **viewpoint**



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

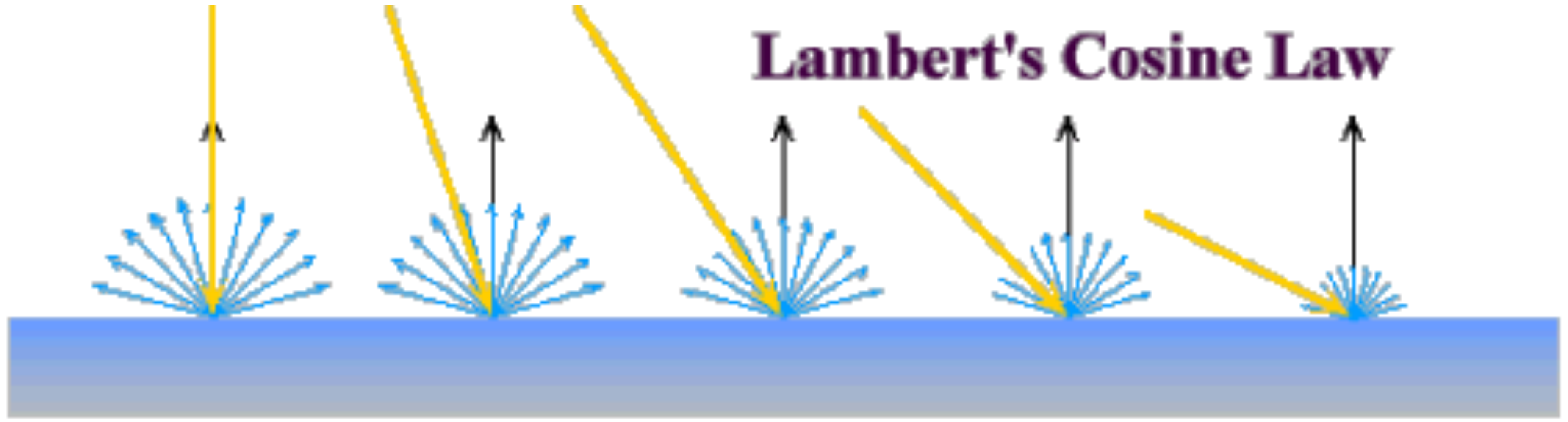
Lecture 2: Re-cap Light and Reflection

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:

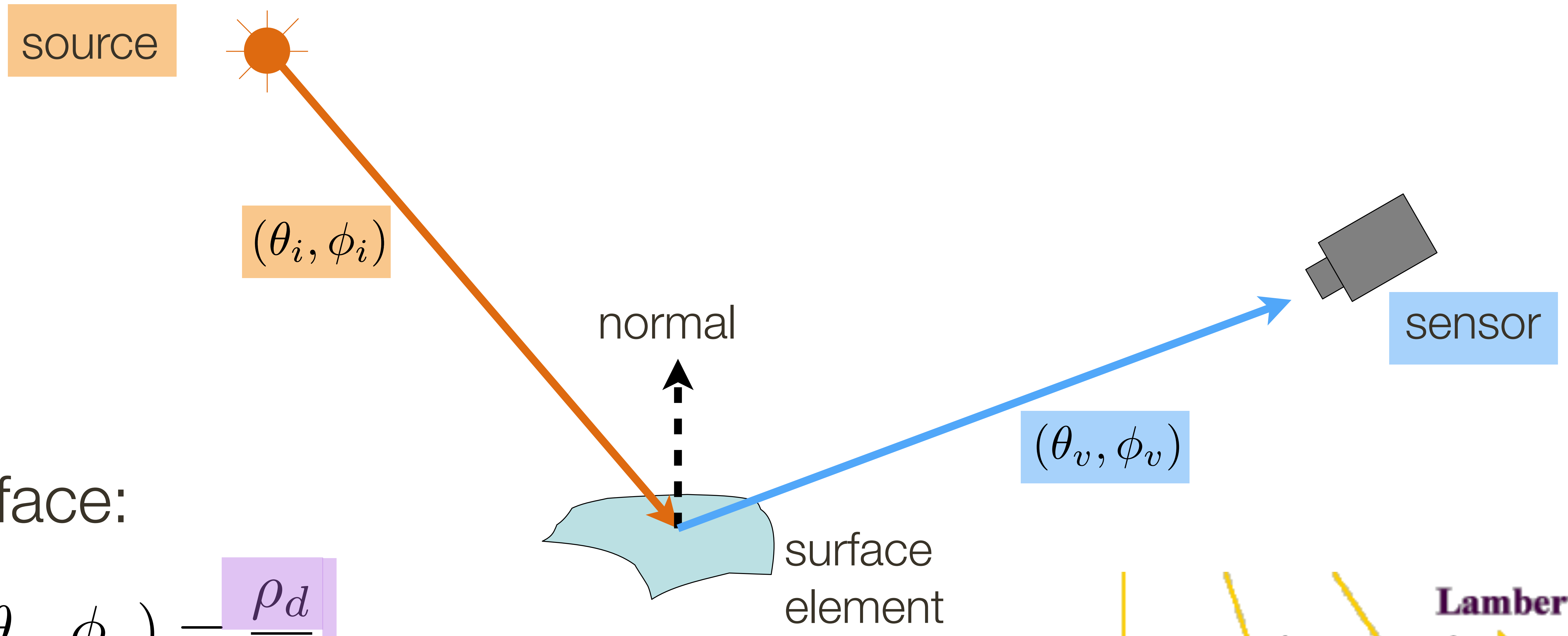
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$



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

Lecture 2: Re-cap Light and Reflection

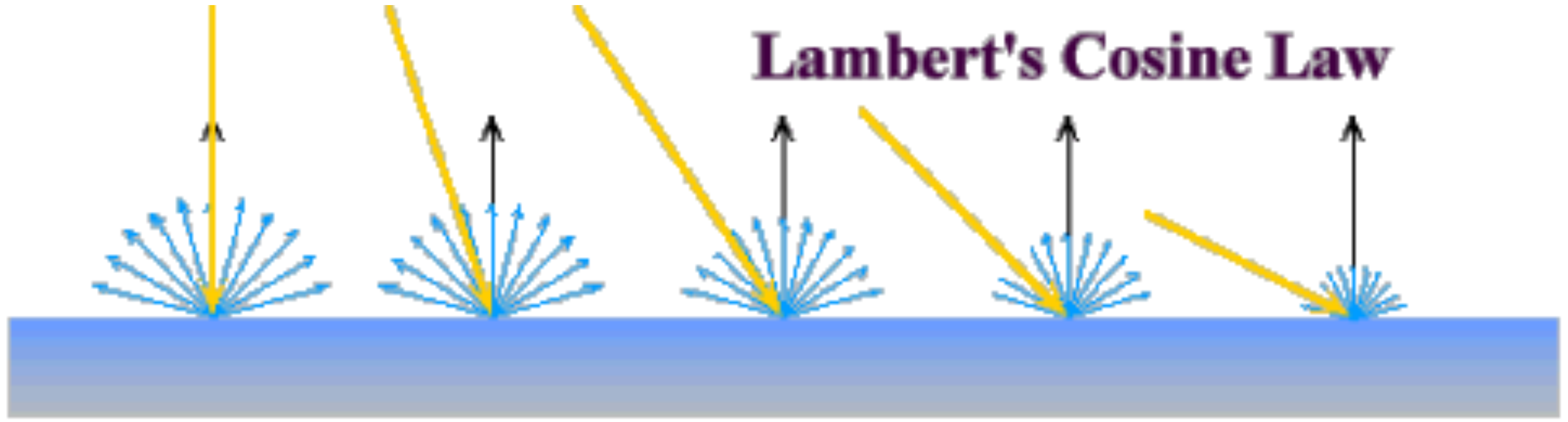
Surface reflection depends on both the **viewing** (θ_v, ϕ_v) and **illumination** (θ_i, ϕ_i) direction, with Bidirectional Reflection Distribution Function: $\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v)$



Lambertian surface:

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

constant, called **albedo**

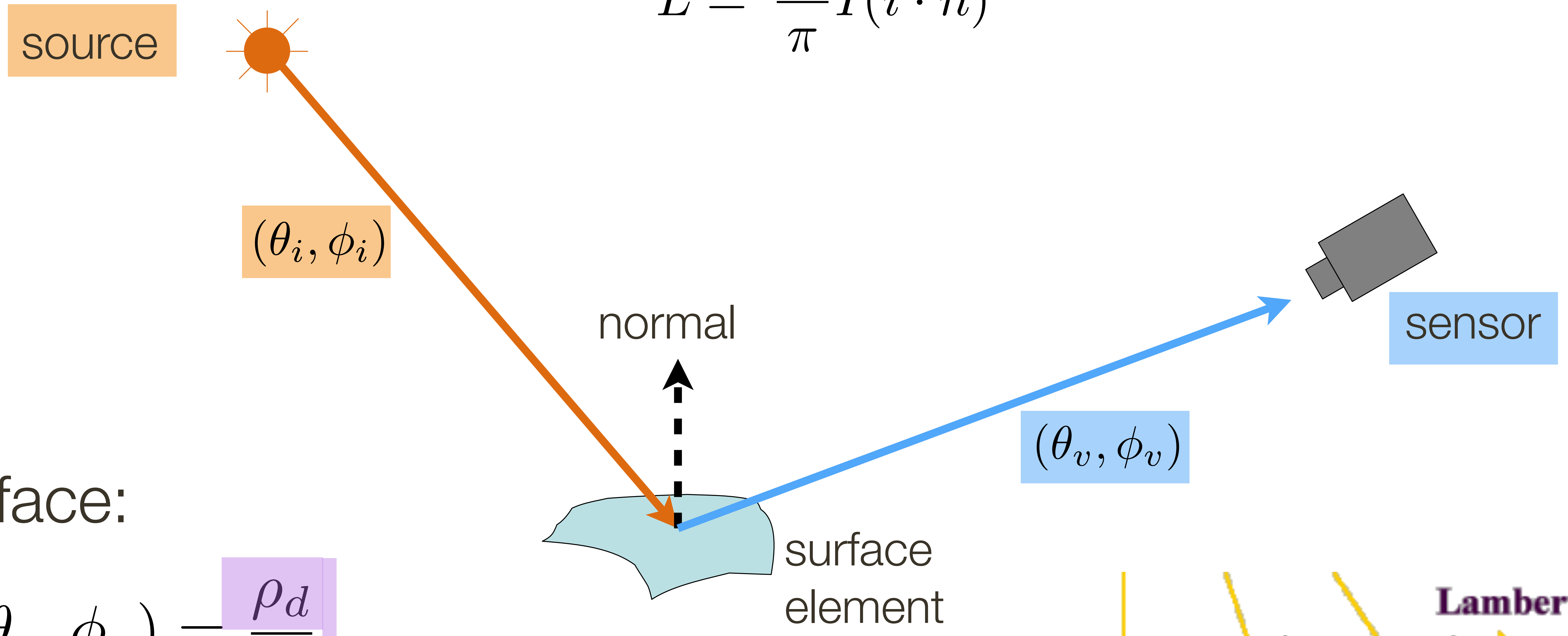


Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 2: Re-cap Light and Reflection

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

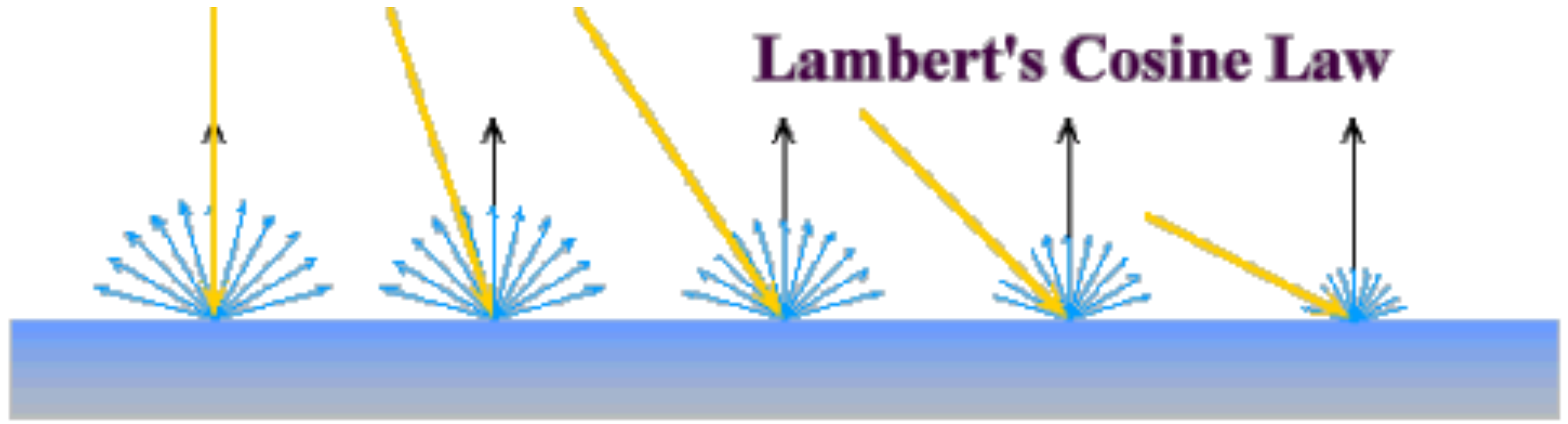
$$L = \frac{\rho_d}{\pi} I(\vec{i} \cdot \vec{n})$$



Lambertian surface:

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

constant, called **albedo**

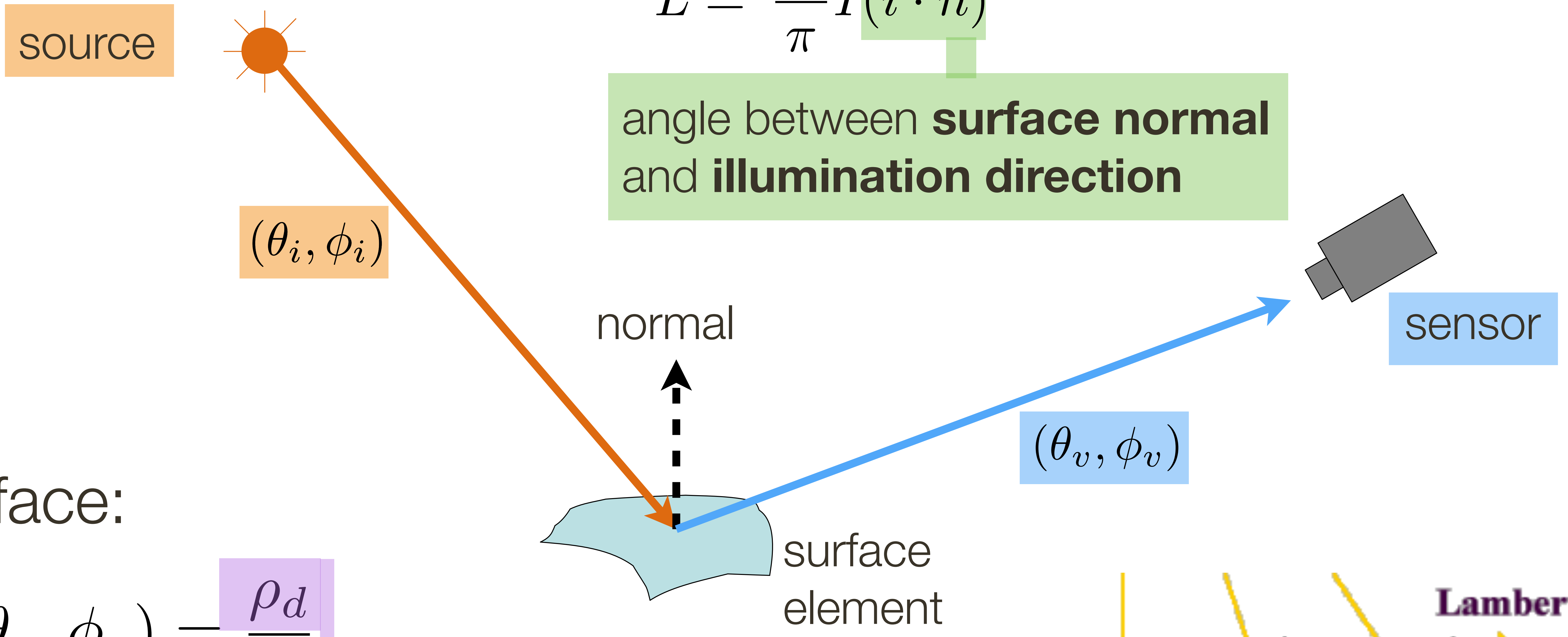


Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 2: Re-cap Light and Reflection

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

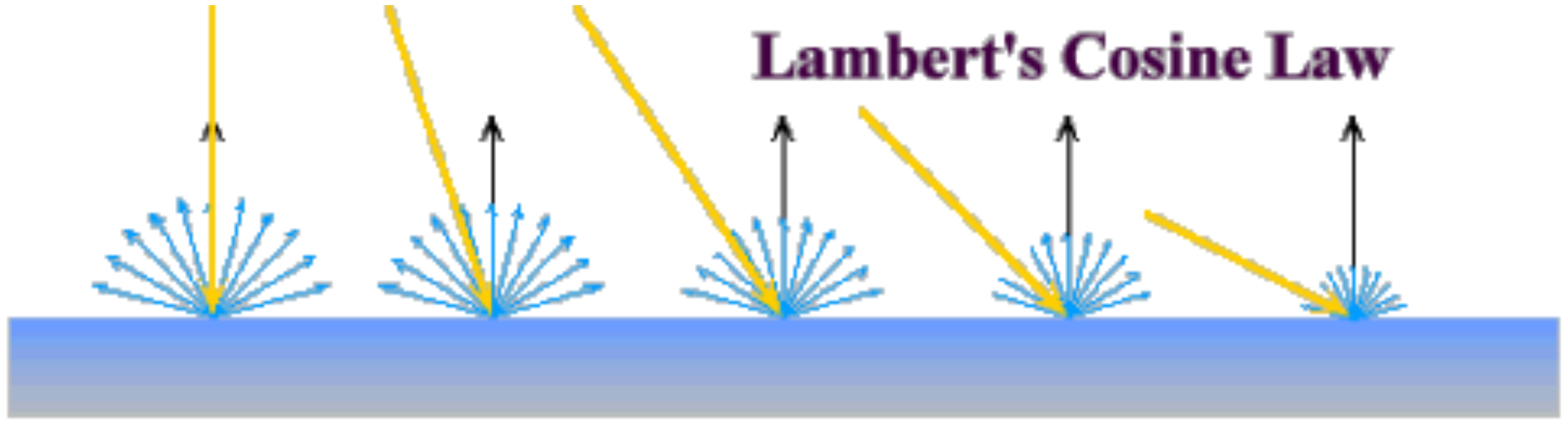
$$L = \frac{\rho_d}{\pi} I(\vec{i} \cdot \vec{n})$$



Lambertian surface:

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

constant, called **albedo**

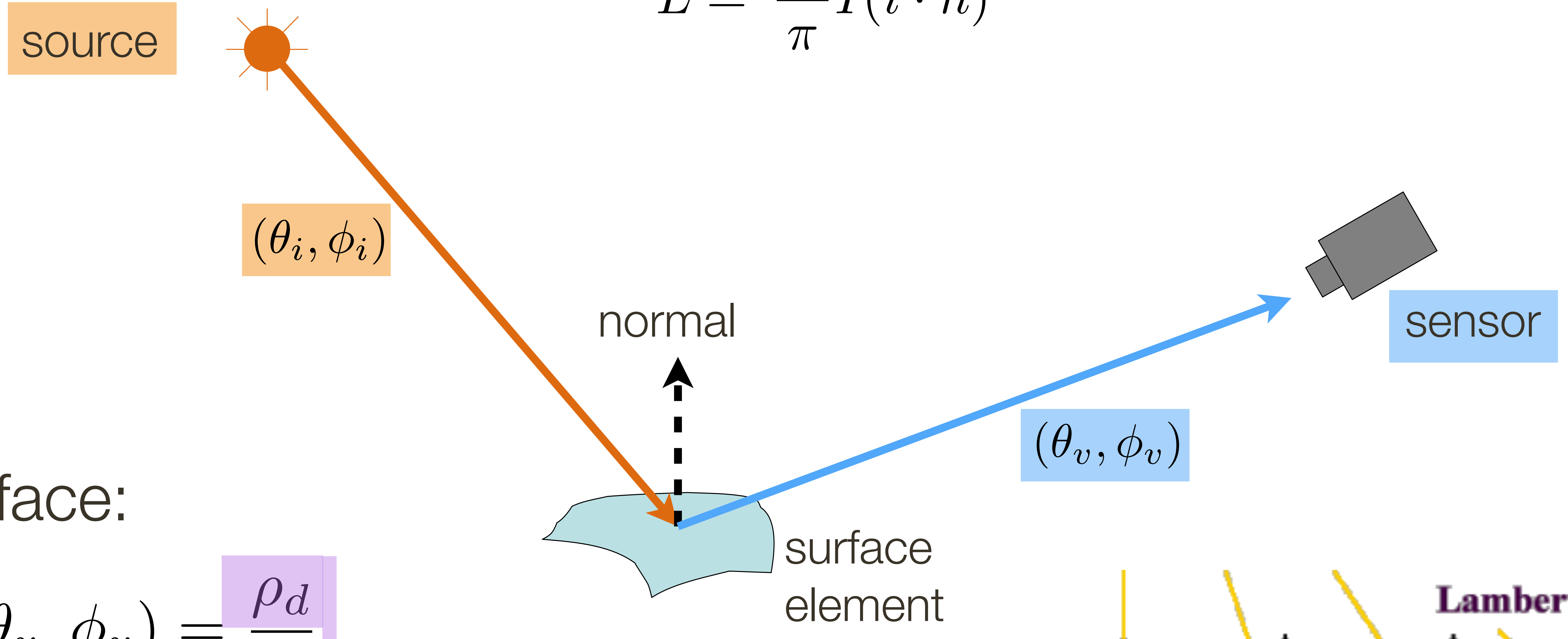


Slide adopted from: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 2: Re-cap Light and Reflection

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

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

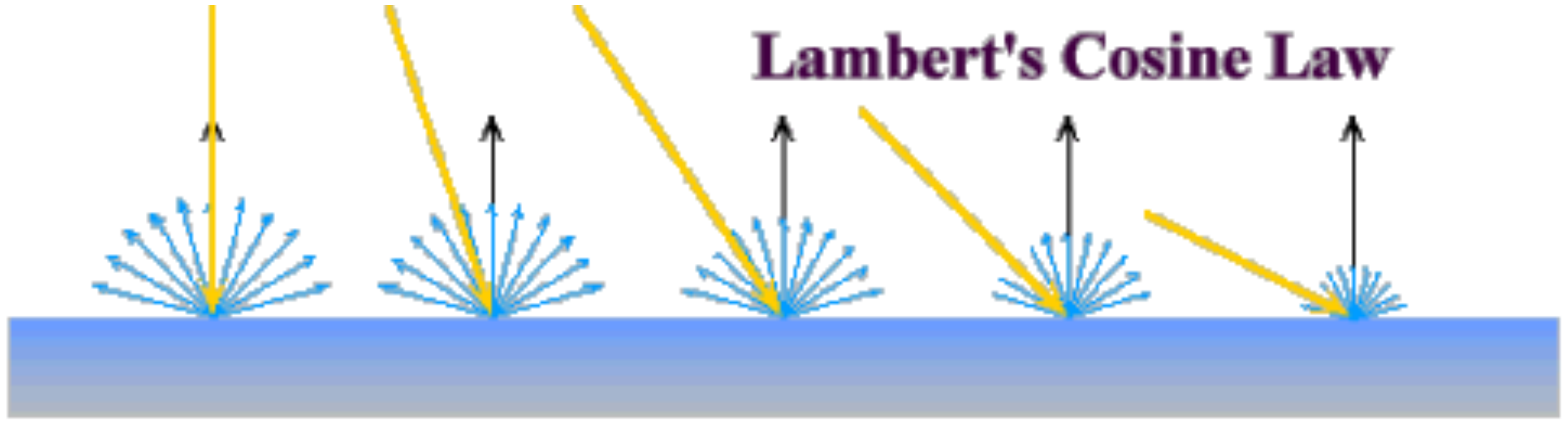


Lambertian surface:

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

constant, called **albedo**

Note: not a function of (θ_v, ϕ_v)



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

Lecture 2: Re-cap Light and Reflection

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

To sum up: For a perfect **lambertian** surface reflected light is

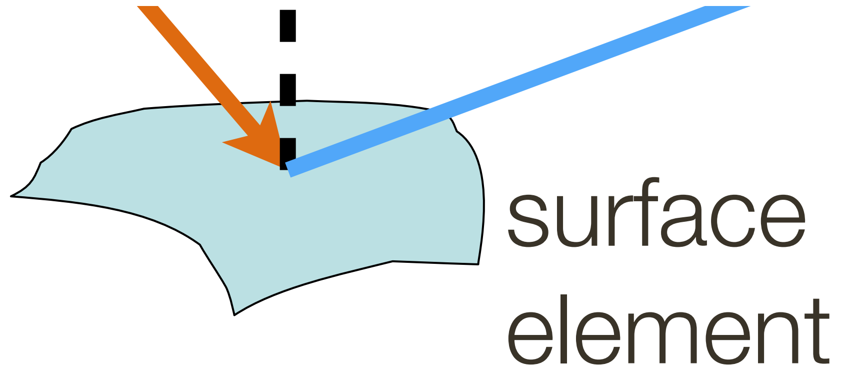
(1) amount and color of incident light — I

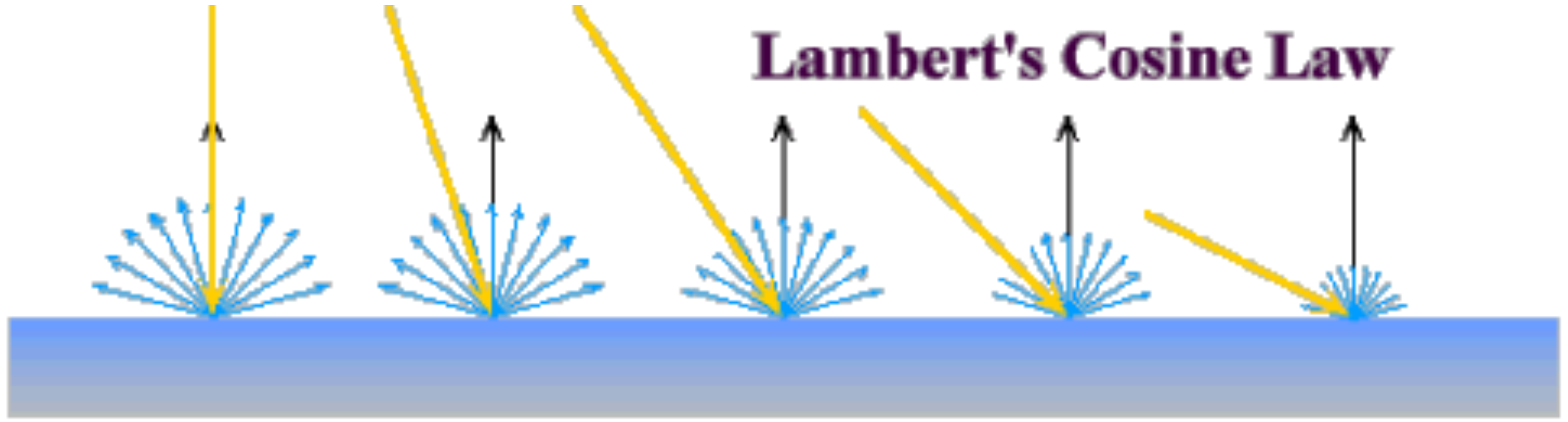
(2) fraction of light being reflected (material) — ρ_d

(3) angle between the light and the surface (geometry) — $(\vec{i} \cdot \vec{n})$

Lambertian surface:

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





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

Lecture 2: Re-cap Light and Reflection

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

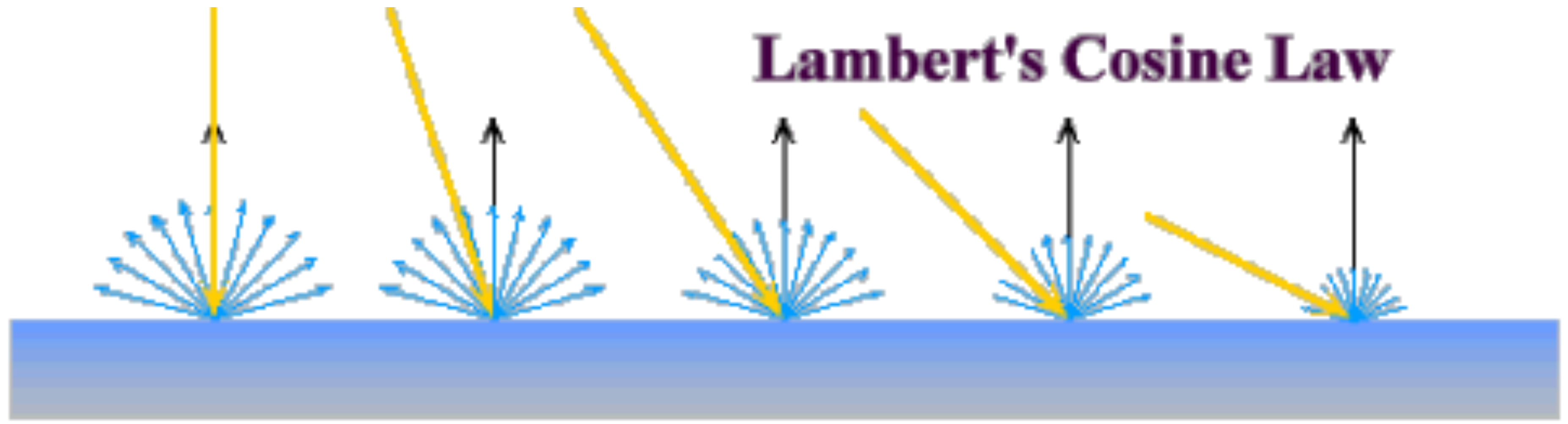
$$L = \frac{\rho_d}{\pi} I(\vec{i} \cdot \vec{n})$$

To sum up: For a perfect **lambertian** surface reflected light is

(1) amount and color of incident light — I
(2) fraction of light being reflected (material) — ρ_d
(3) angle between the light and the surface (geometry) — $(\vec{i} \cdot \vec{n})$

Lambertian surface:

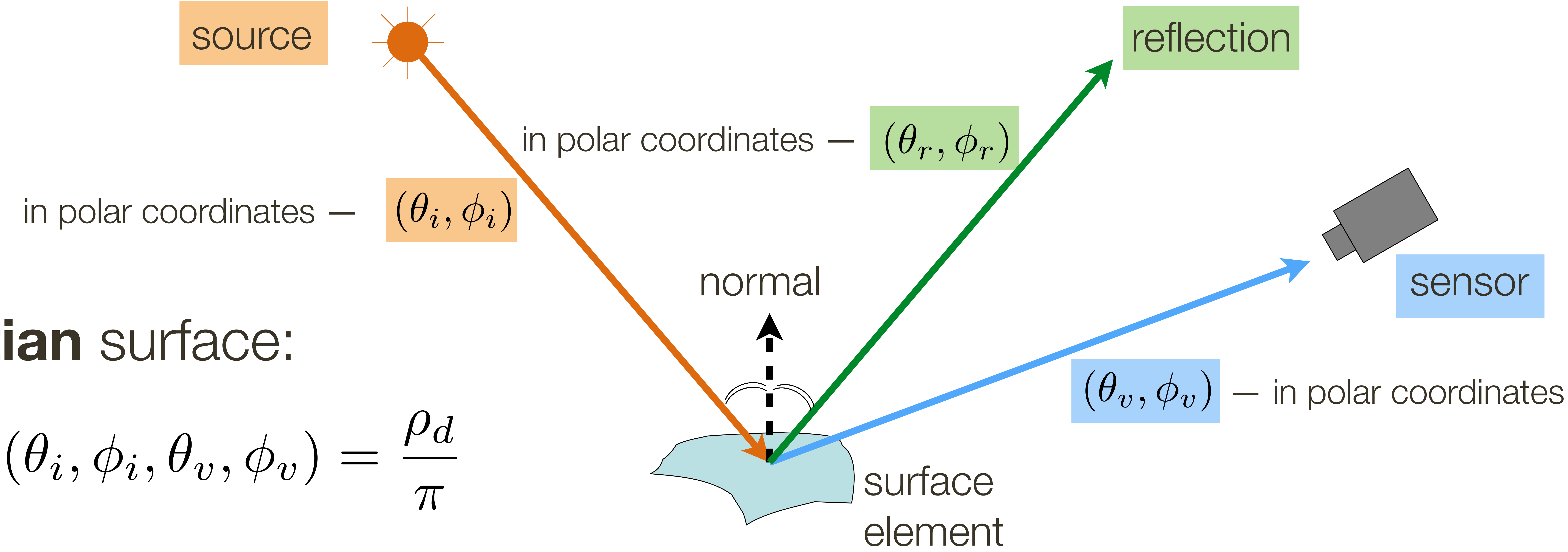
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \frac{\rho_d}{\pi}$$



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

Lecture 2: Re-cap Light and Reflection

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:

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

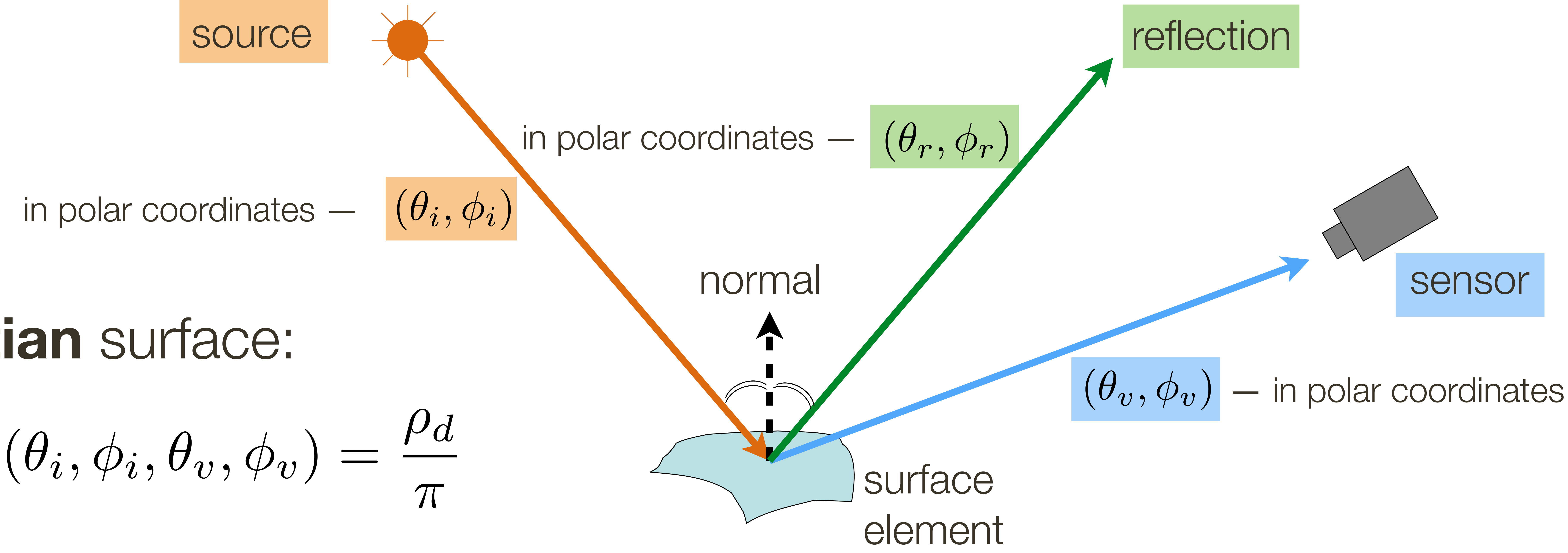
Mirror surface: all incident light reflected in one directions $(\theta_v, \phi_v) = (\theta_r, \phi_r)$

$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \begin{cases} 1 & (\theta_i, \phi_i) == (\theta_v, \phi_v) \\ 0 & otherwise \end{cases}$$

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

Lecture 2: Re-cap Light and Reflection

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 $(\theta_v, \phi_v) = (\theta_r, \phi_r)$

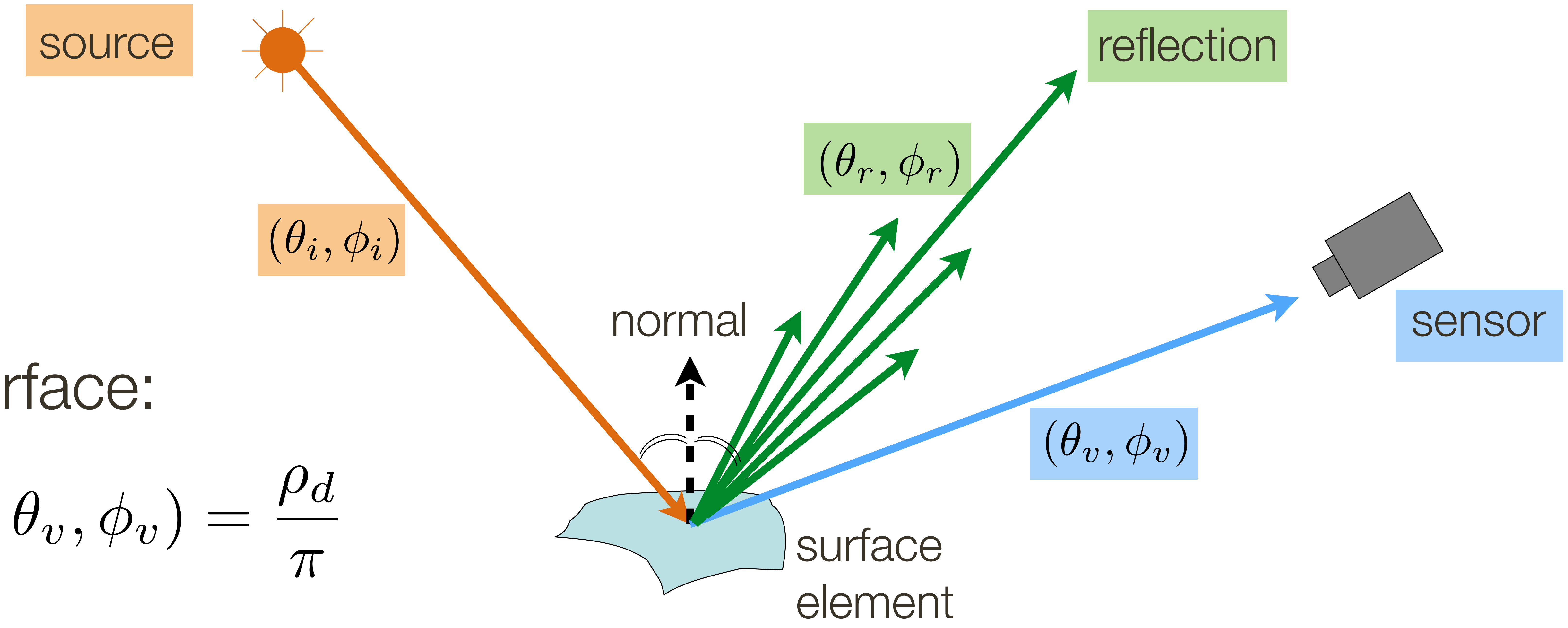
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = \begin{cases} 1 & (\theta_i, \phi_i) == (\theta_v, \phi_v) \\ 0 & otherwise \end{cases}$$

Note: is a function of (θ_v, ϕ_v)

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

Lecture 2: Re-cap Light and Reflection

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:

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

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

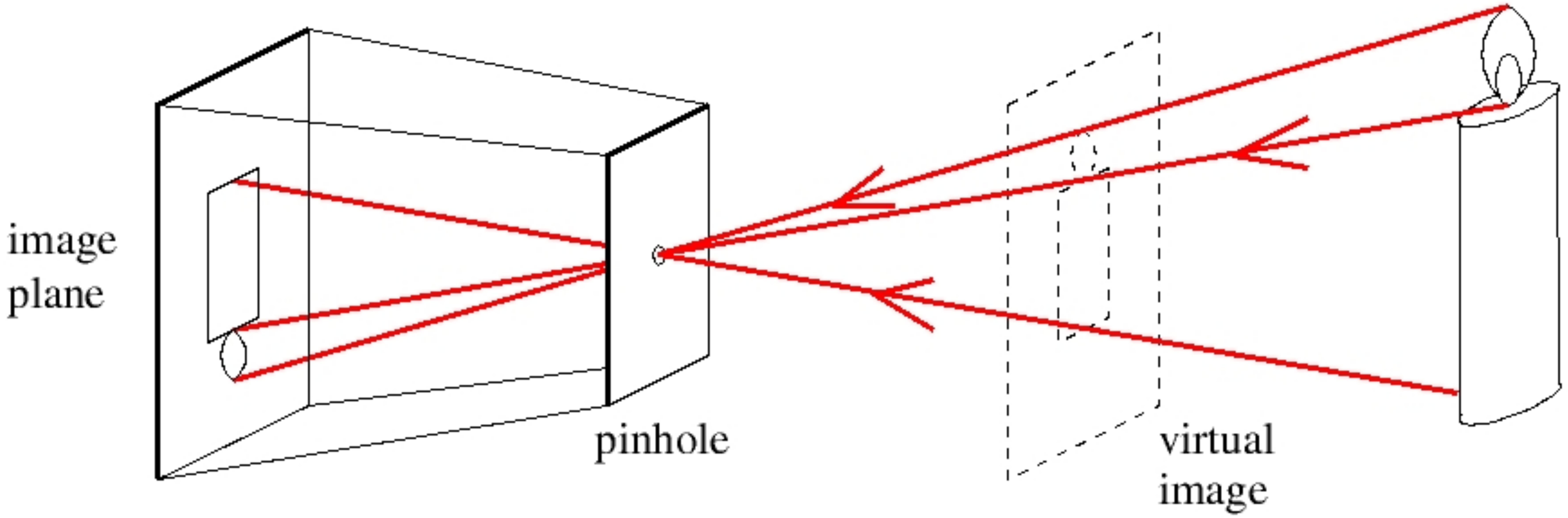
$$\mathbf{BRDF}(\theta_i, \phi_i, \theta_v, \phi_v) = (\vec{i} \cdot \vec{v})^{\alpha}$$

Note: is a function of (θ_v, ϕ_v)

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

Lecture 2: Re-cap Pinhole Camera Abstraction

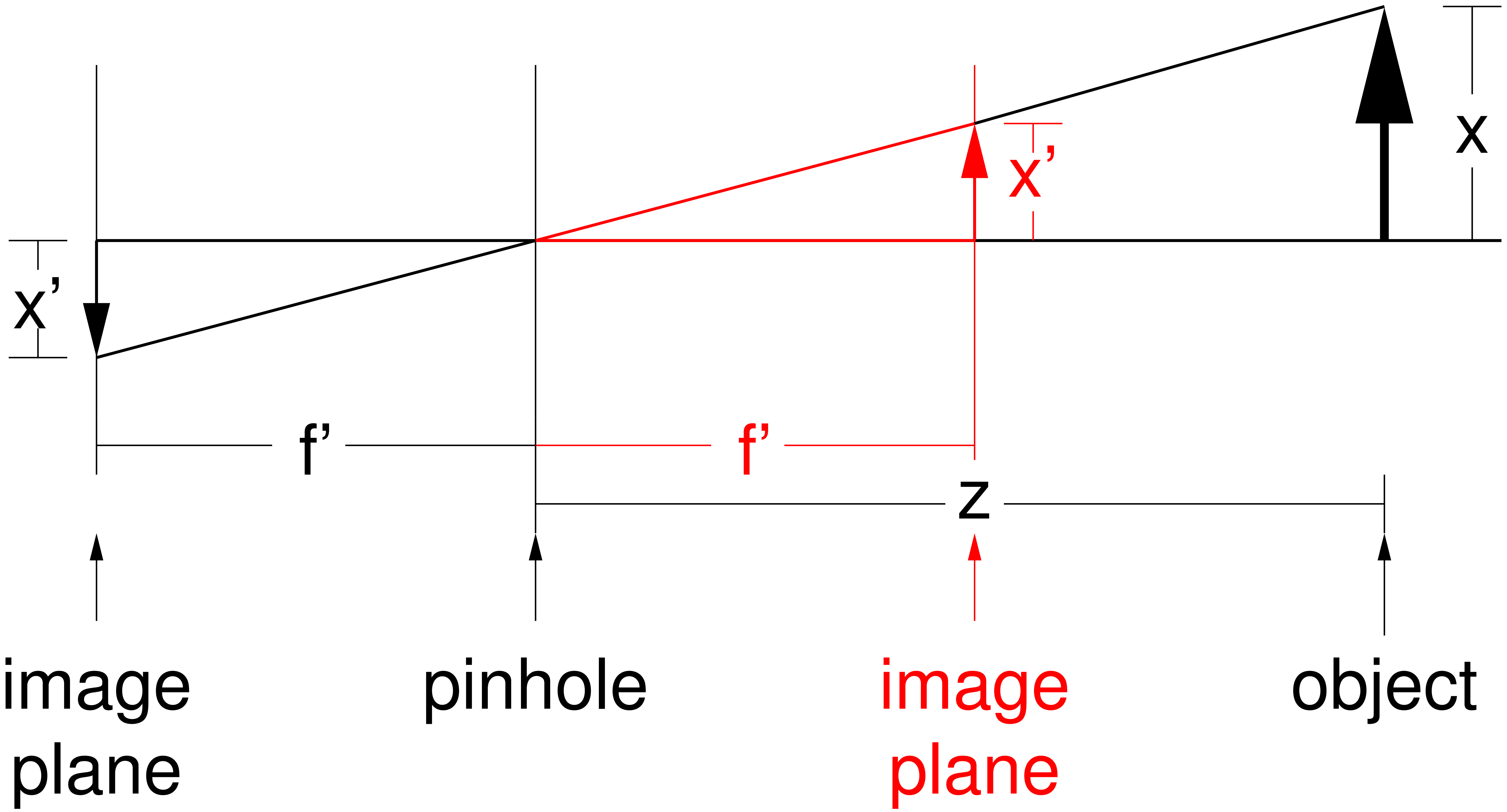
A pinhole camera is a box with a small hall (**aperture**) in it



Forsyth & Ponce (2nd ed.) Figure 1.2

Lecture 2: Re-cap Pinhole Camera Abstraction

Pinhole Camera Abstraction



Lecture 2: Re-cap Projection

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

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

Lecture 2: Re-cap Projection

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	$x' = x$ $y' = y$	

Lecture 2: Re-cap Reason for Lenses

- 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

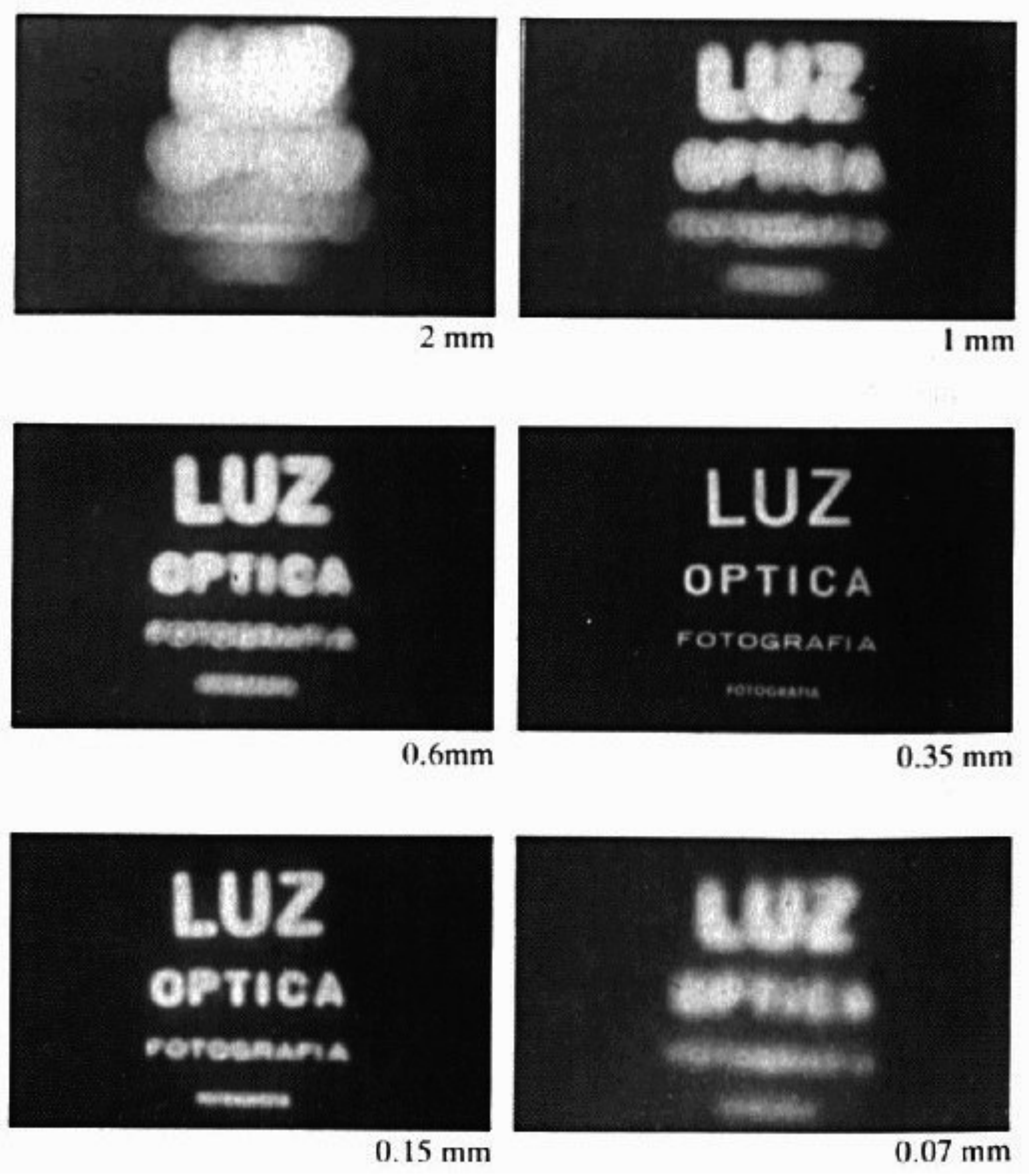
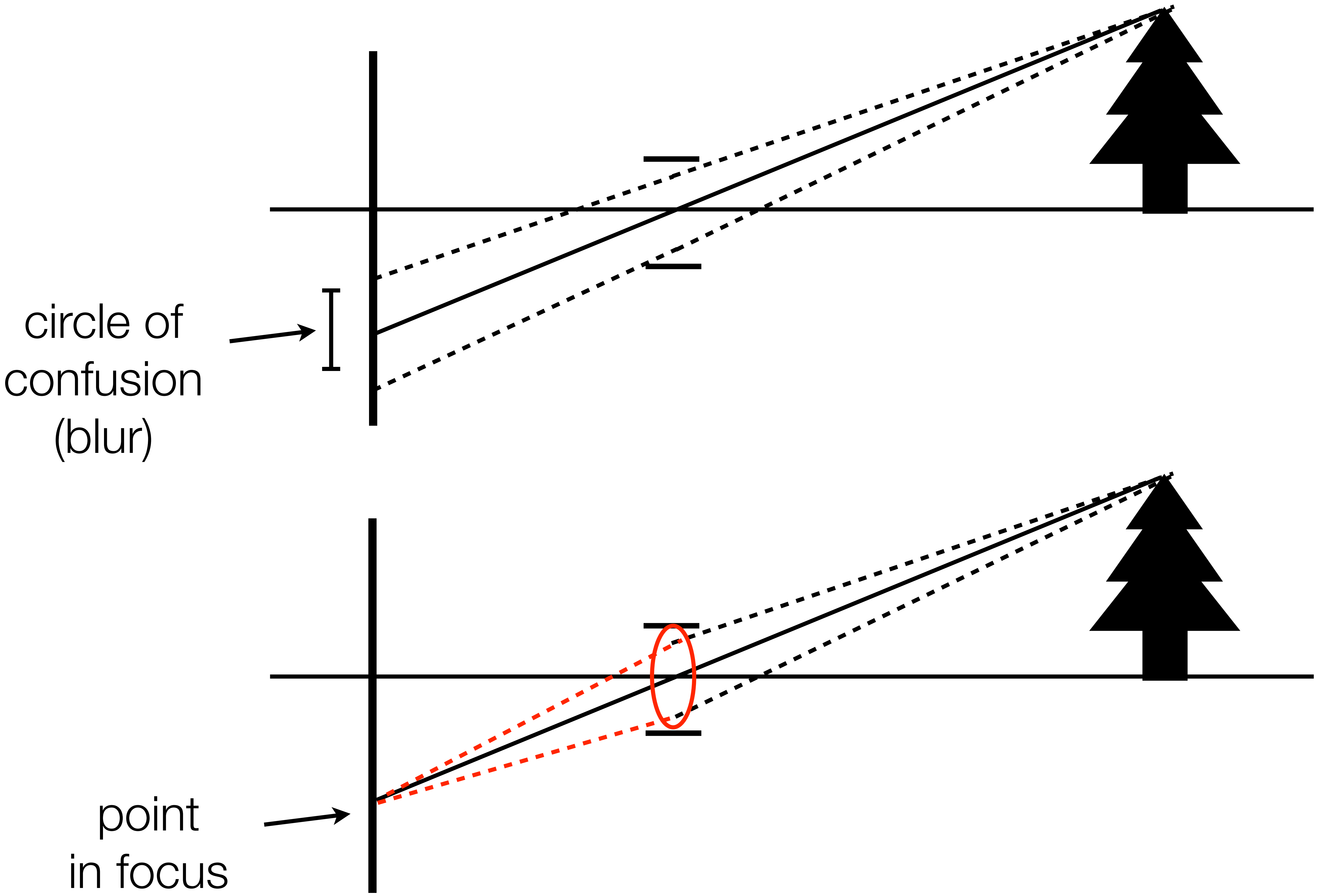


Image Credit: Credit: E. Hecht. "Optics," Addison-Wesley, 1987

Lecture 2: Re-cap Reason for Lenses

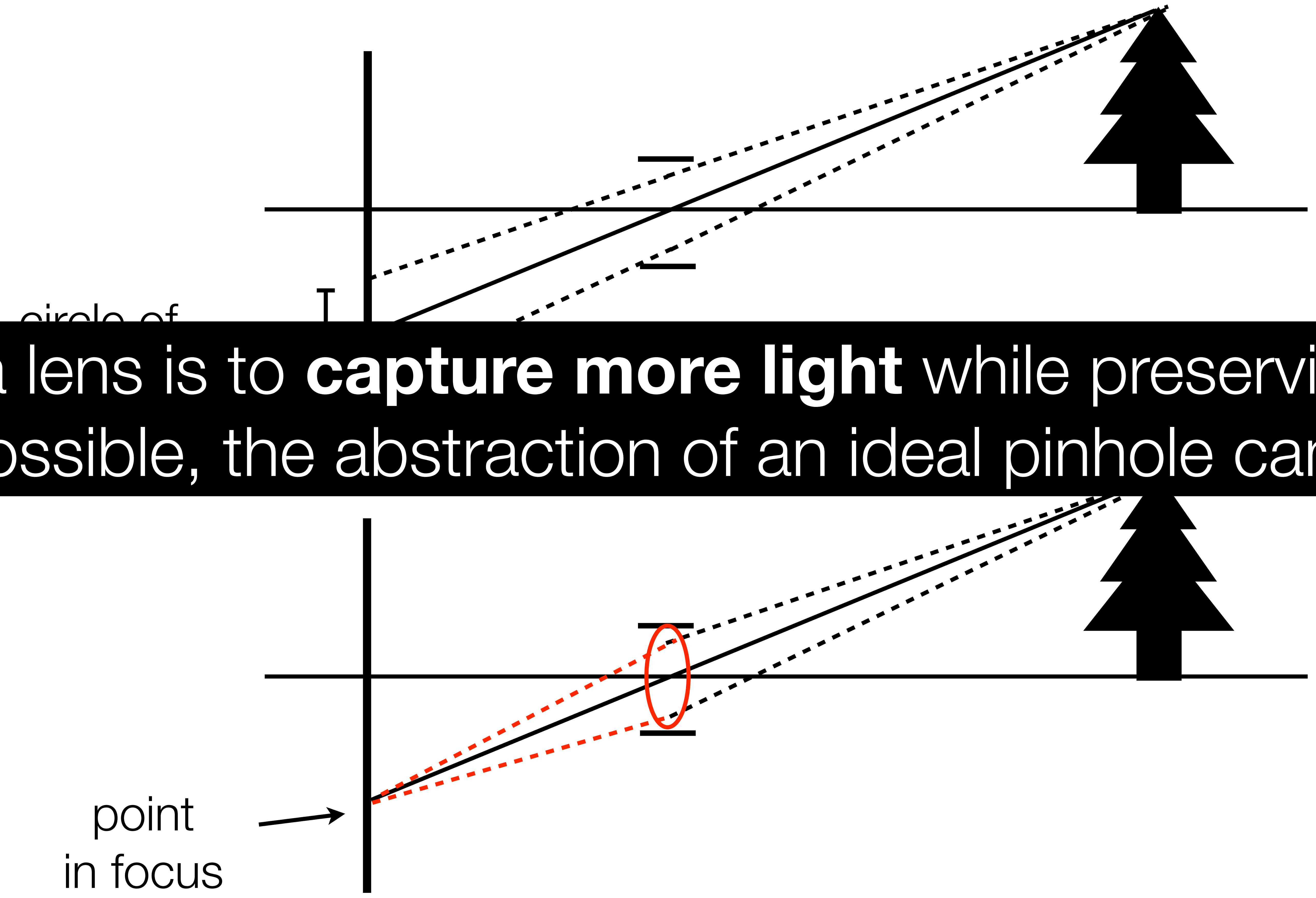
A real camera must have a finite aperture to get enough light, but this causes blur in the image



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

Reason for **Lenses**

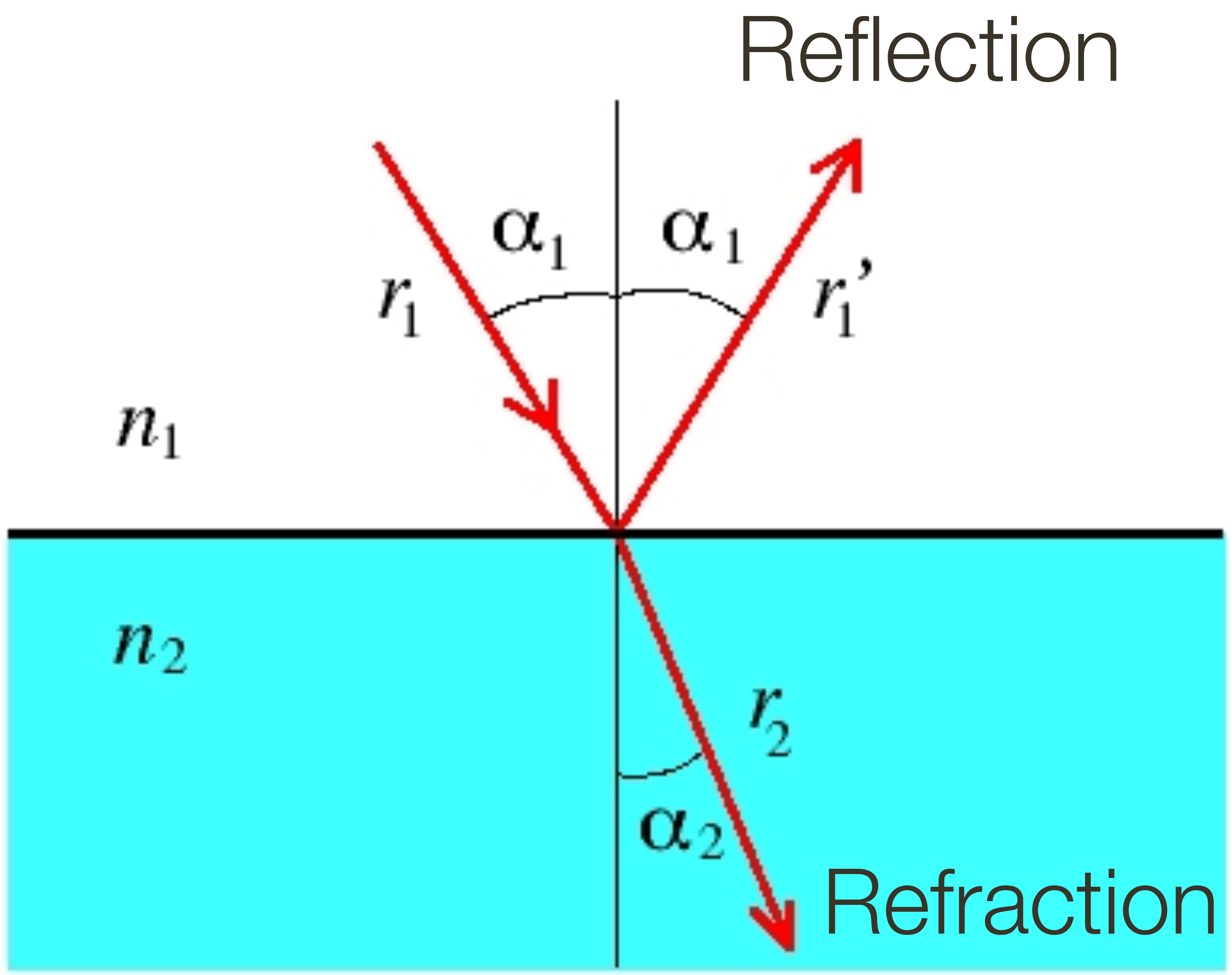
A real camera must have a finite aperture to get enough light, but this causes blur in the image



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



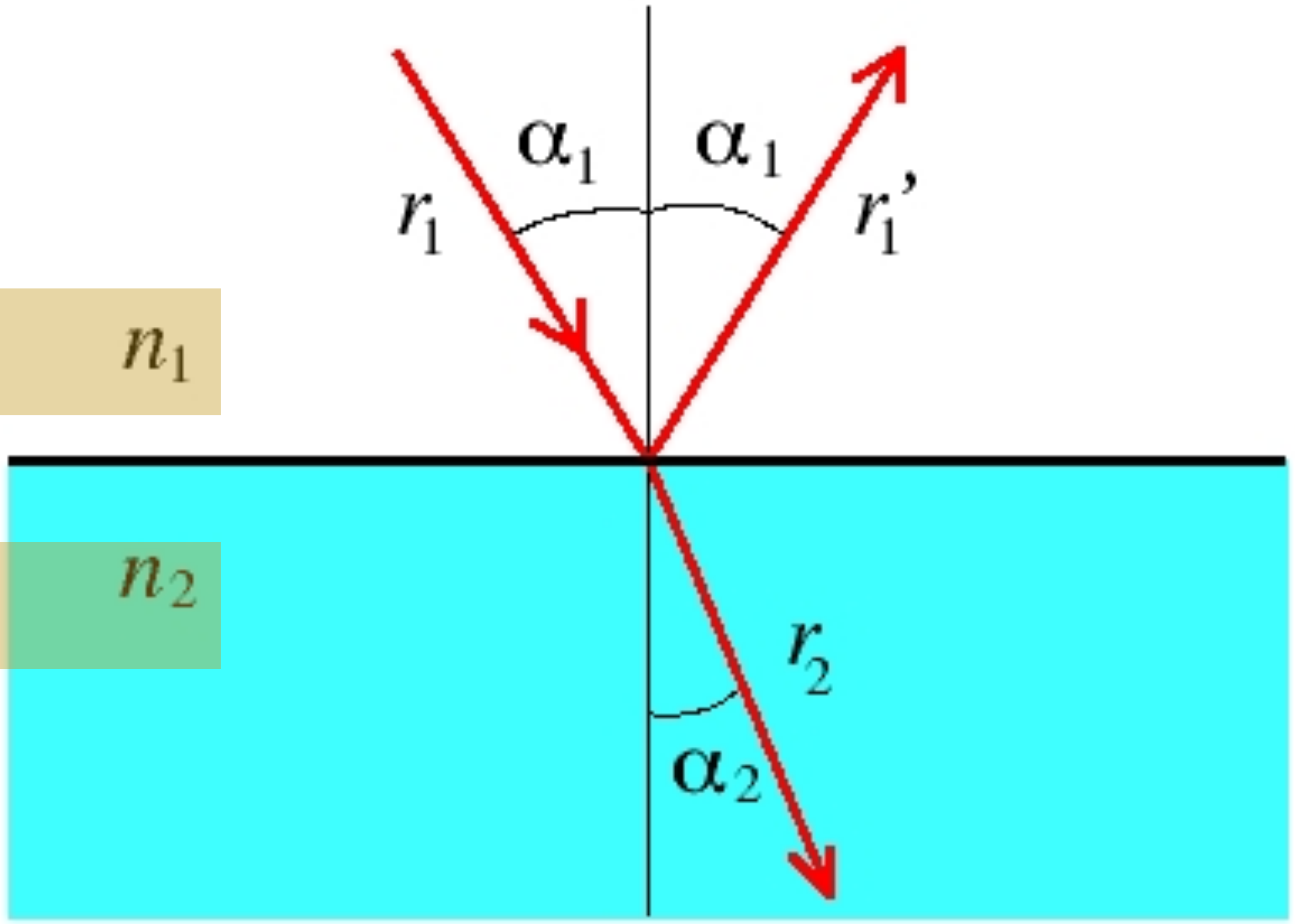
$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$$

Snell's Law

n_1

Index of **refraction**

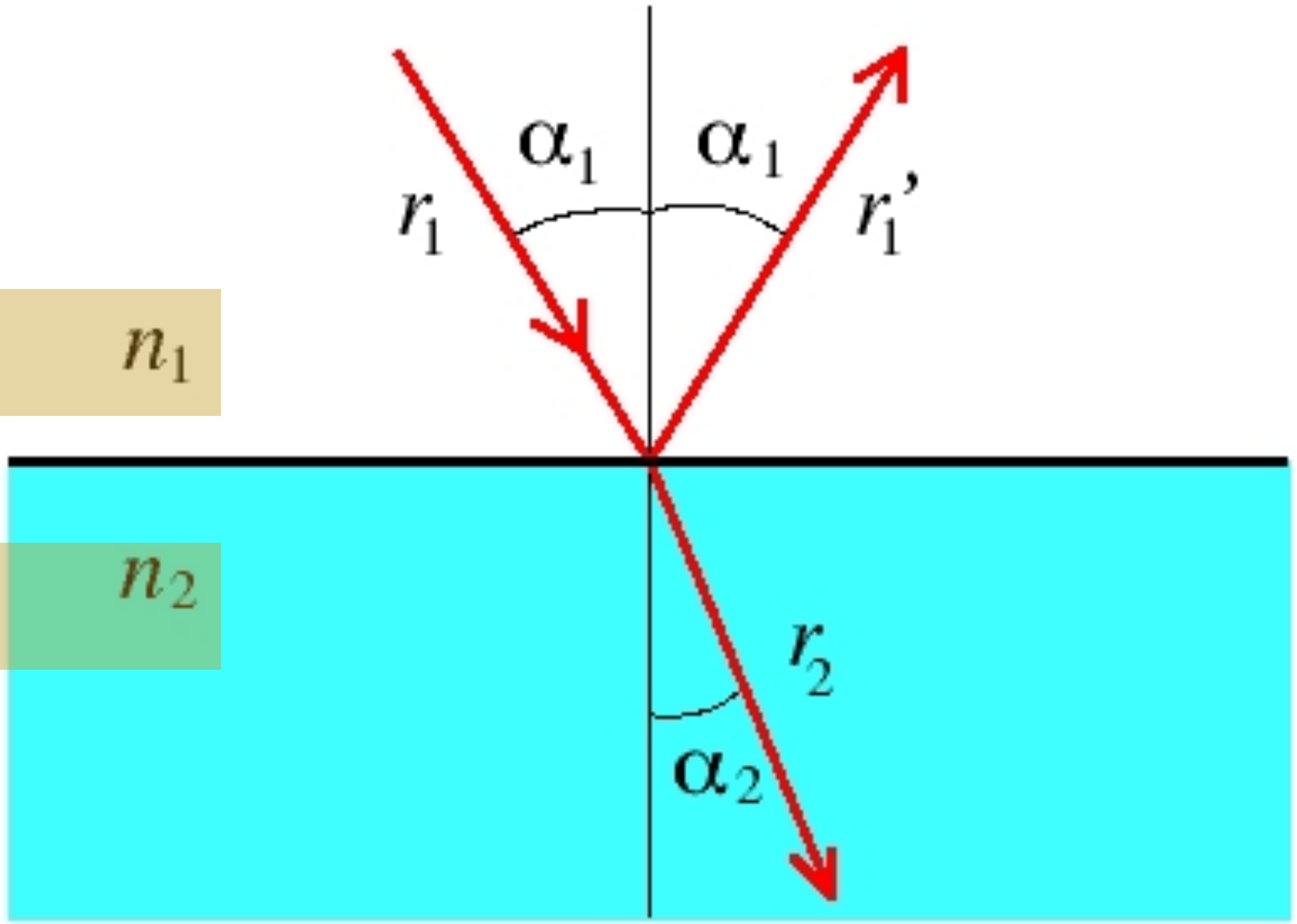
n_2



$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$$

Snell's Law

Exercise: Would it make sense to make the lens from material who's index of refraction equals to air? Why?



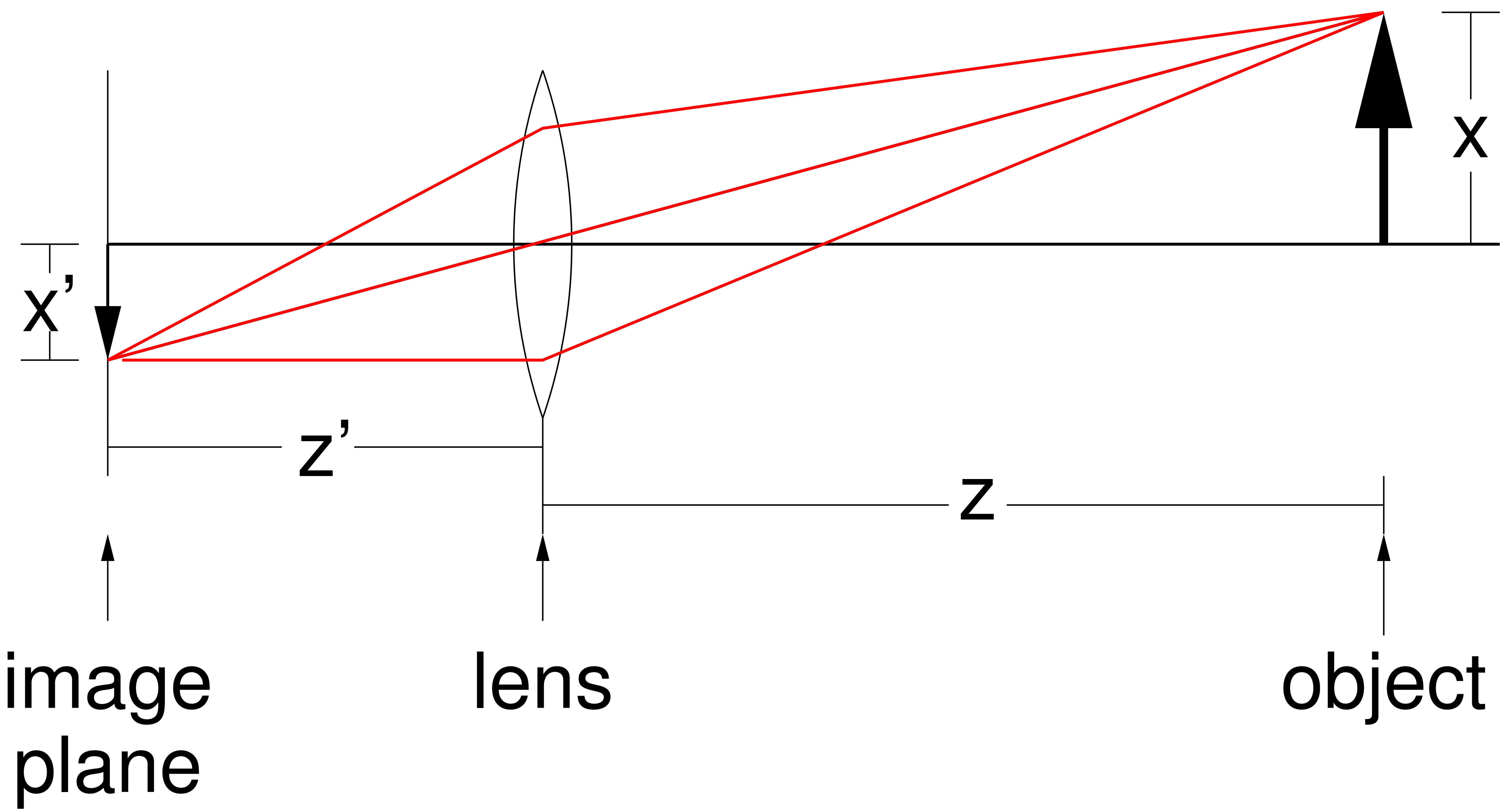
n_1

Index of **refraction**

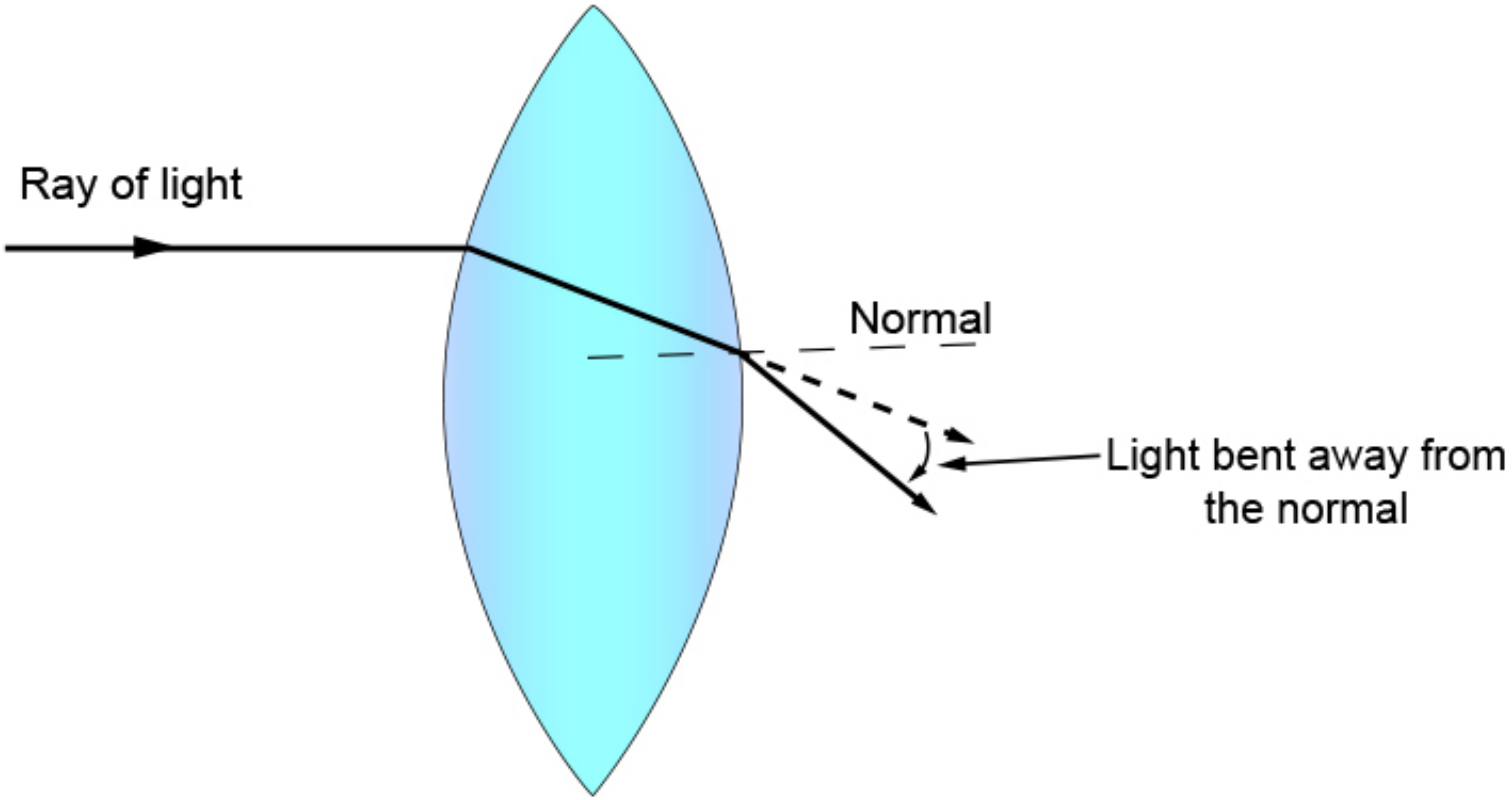
n_2

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$$

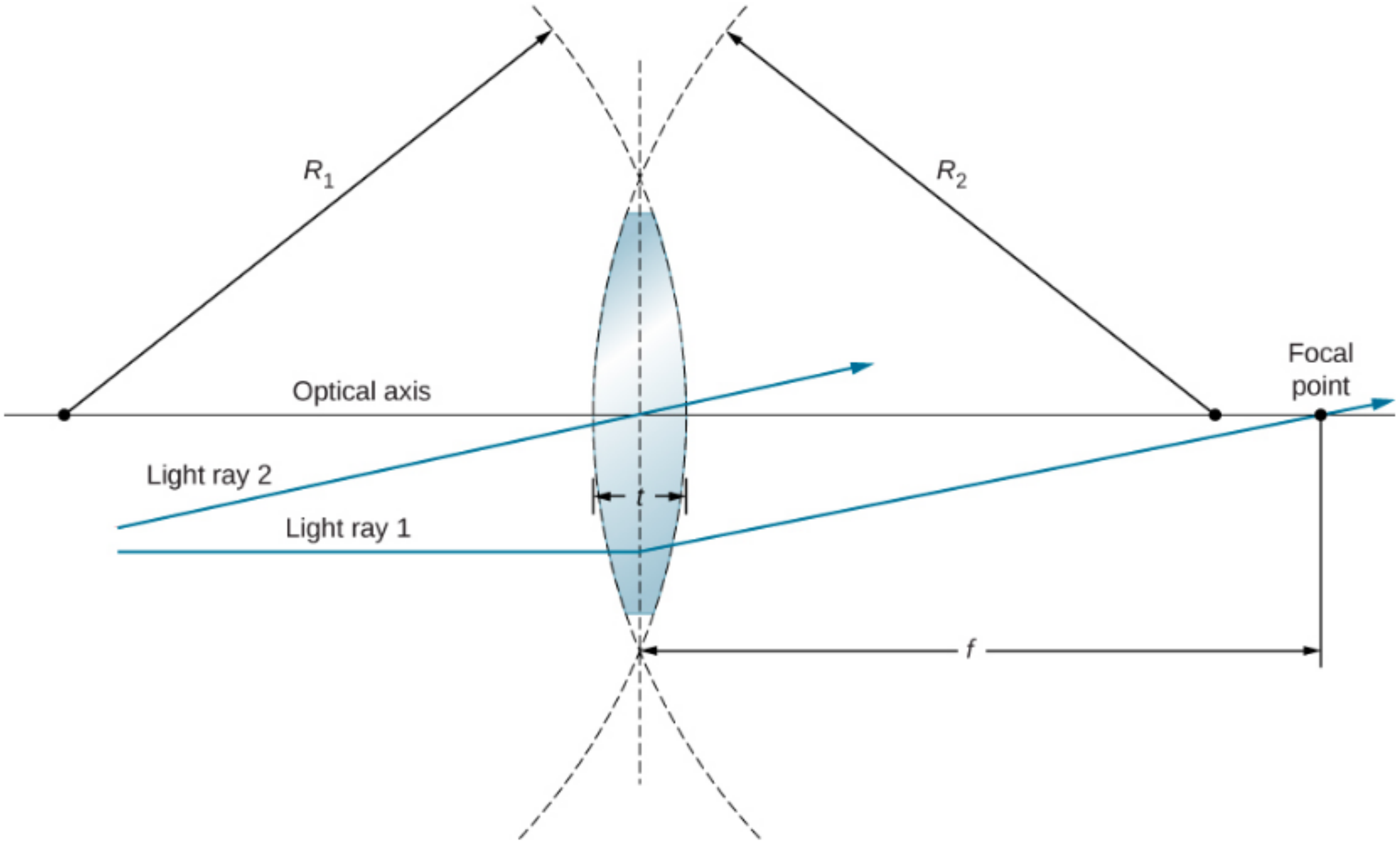
Pinhole Model **with Lens**



General Lens

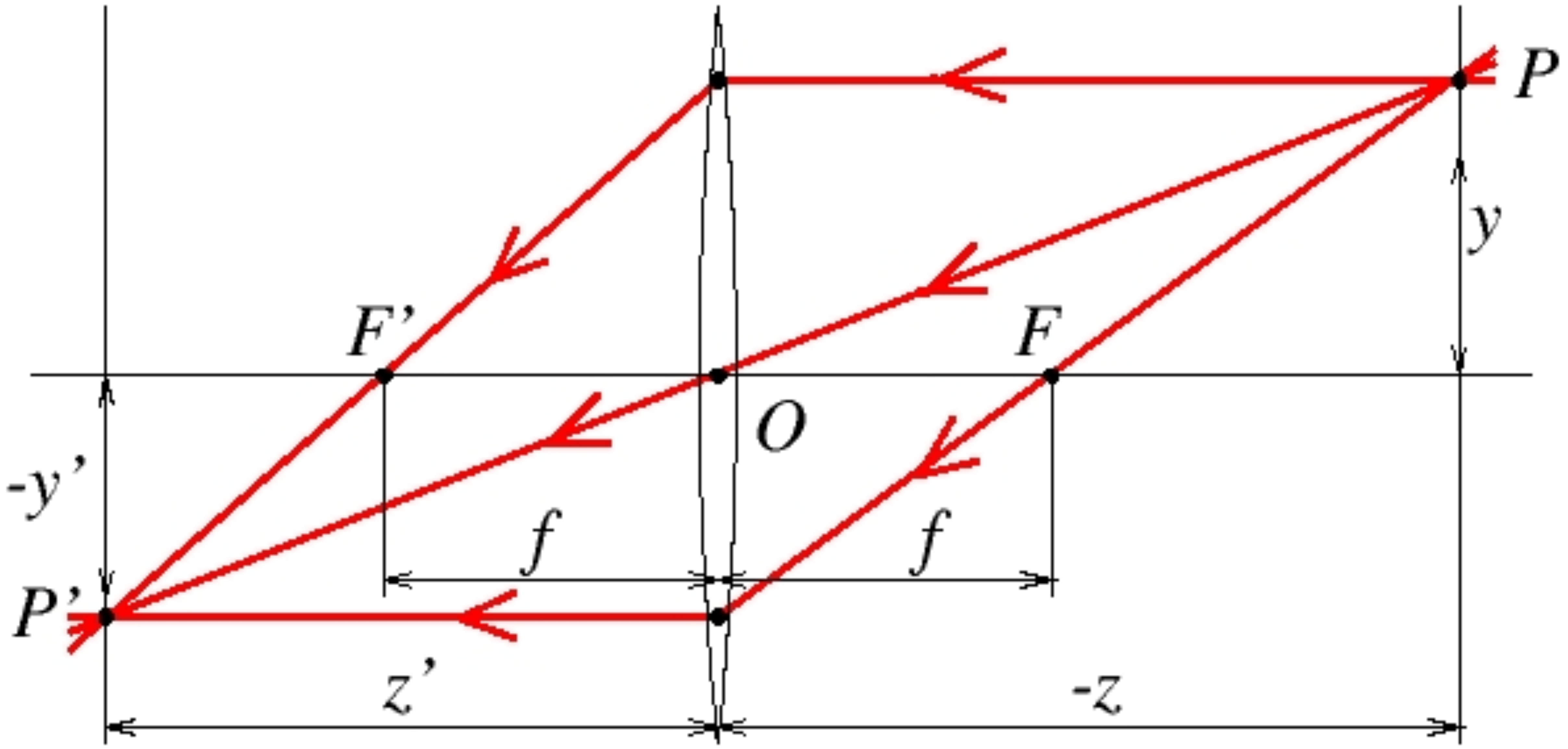


Thin Lens



https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_(OpenStax)/Map%3A_University_Physics_III_-_Optics_and_Modern_Physics_(OpenStax)/02%3A_Geometric_Optics_and_Image_Formation/2.05%3A_Thin_Lenses

Thin Lens Equation

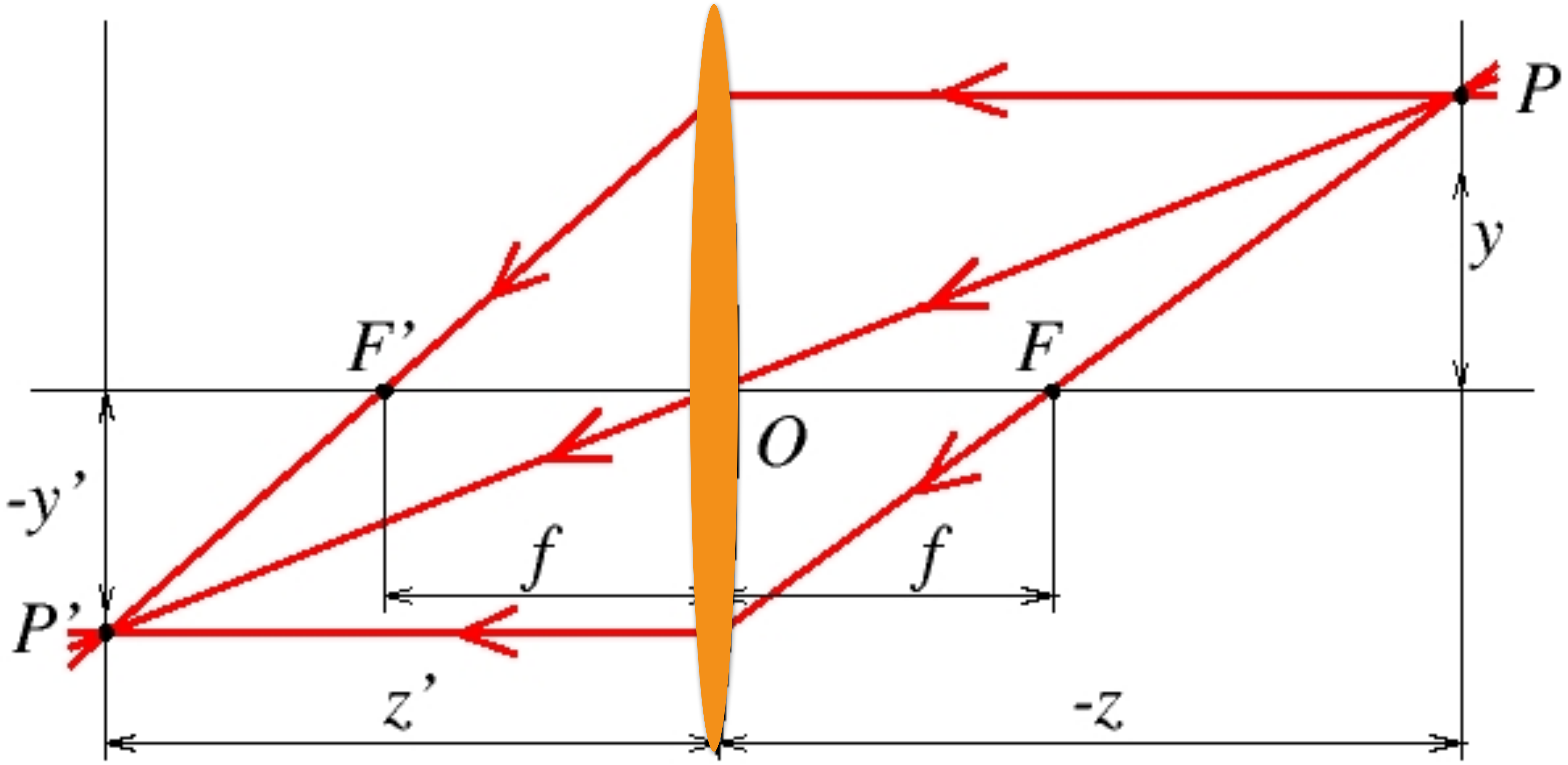


Forsyth & Ponce (1st ed.) Figure 1.9

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Thin Lens Equation

Focal Length: Property of the lens (geometry and refraction index)

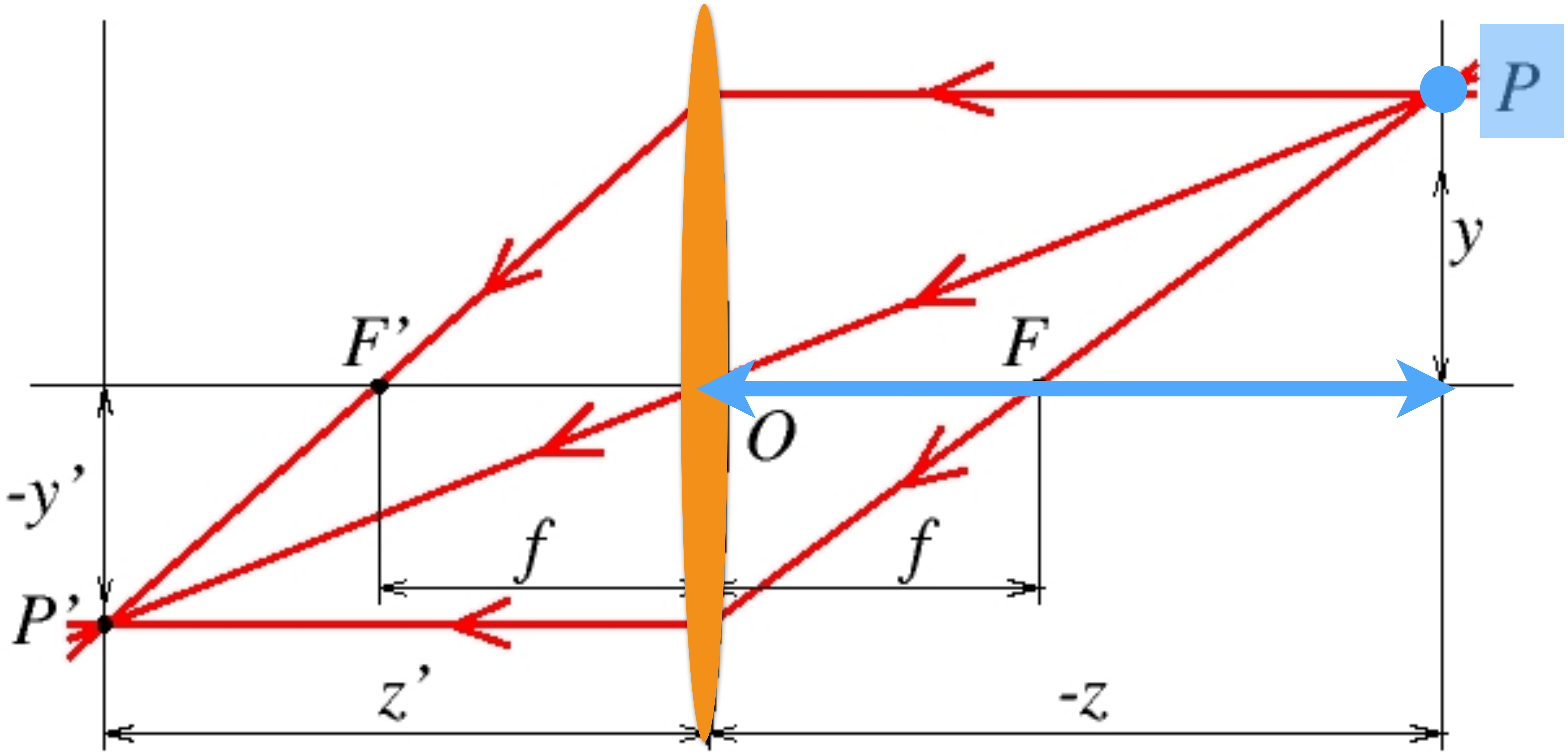


Forsyth & Ponce (1st ed.) Figure 1.9

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Thin Lens Equation

Focal Length: Property of the lens (geometry and refraction index)



Depth of the point (P) in the world

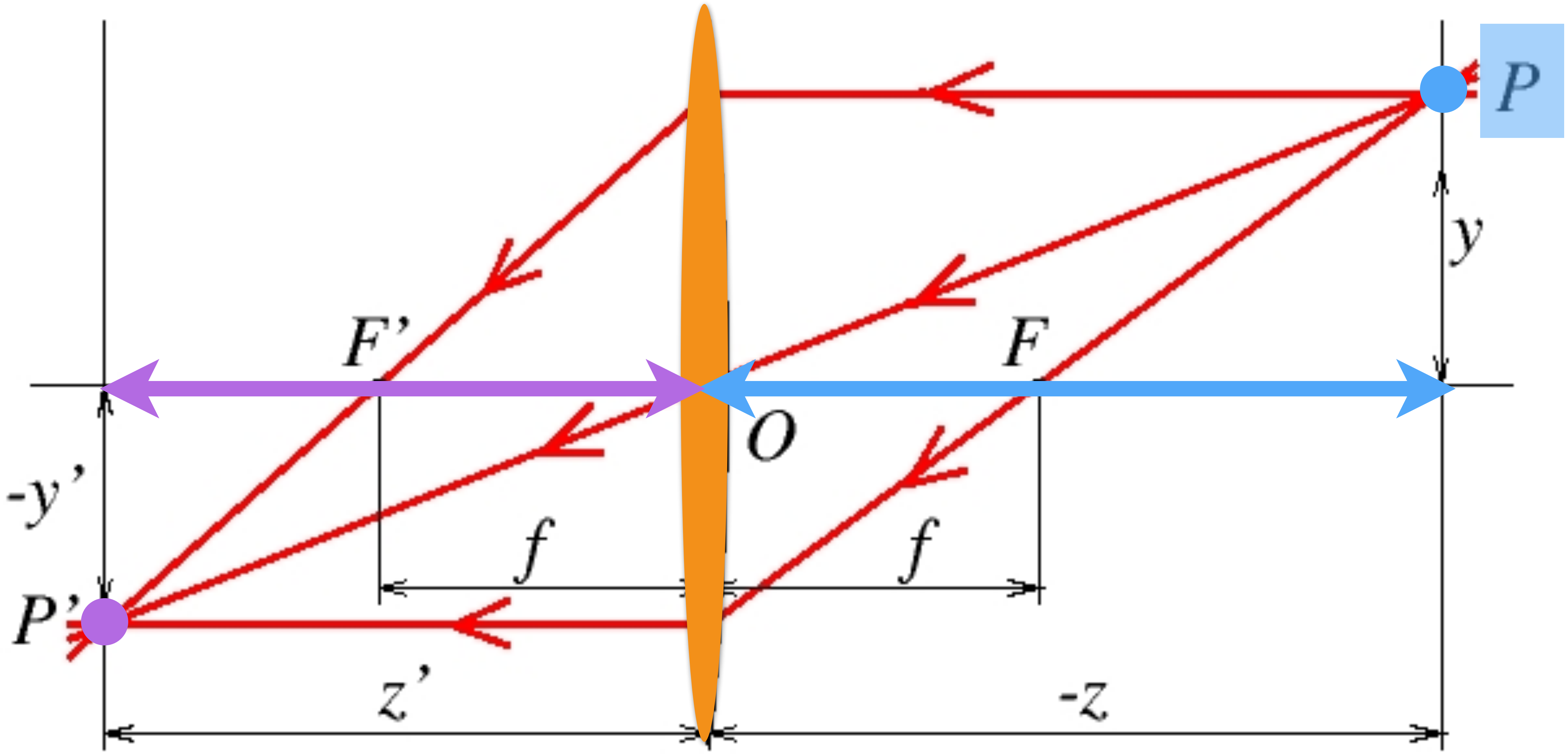
Forsyth & Ponce (1st ed.) Figure 1.9

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Thin Lens Equation

Focal Length: Property of the lens (geometry and refraction index)

Location of the imaging plane where the projection of this point (P) will be in focus



Depth of the point (P) in the world

Forsyth & Ponce (1st ed.) Figure 1.9

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Pinhole Camera with a **Lens**

Perspective Projection: location in the image where a 3D world point projects

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

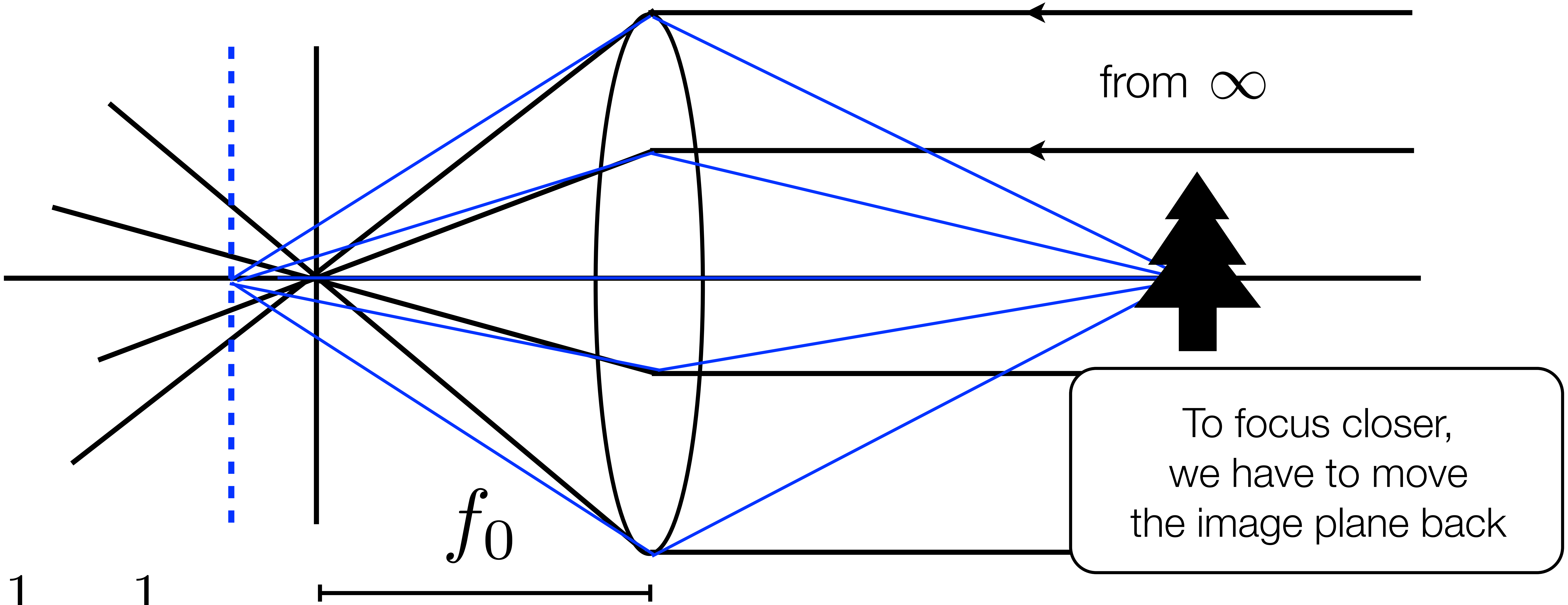
Thin Lens Equation: depth of the imaging plane itself where this point will be in focus

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Lens Basics

A lens focuses parallel rays (from points at infinity) at focal length of the lens

Rays passing through the center of the lens are not bent

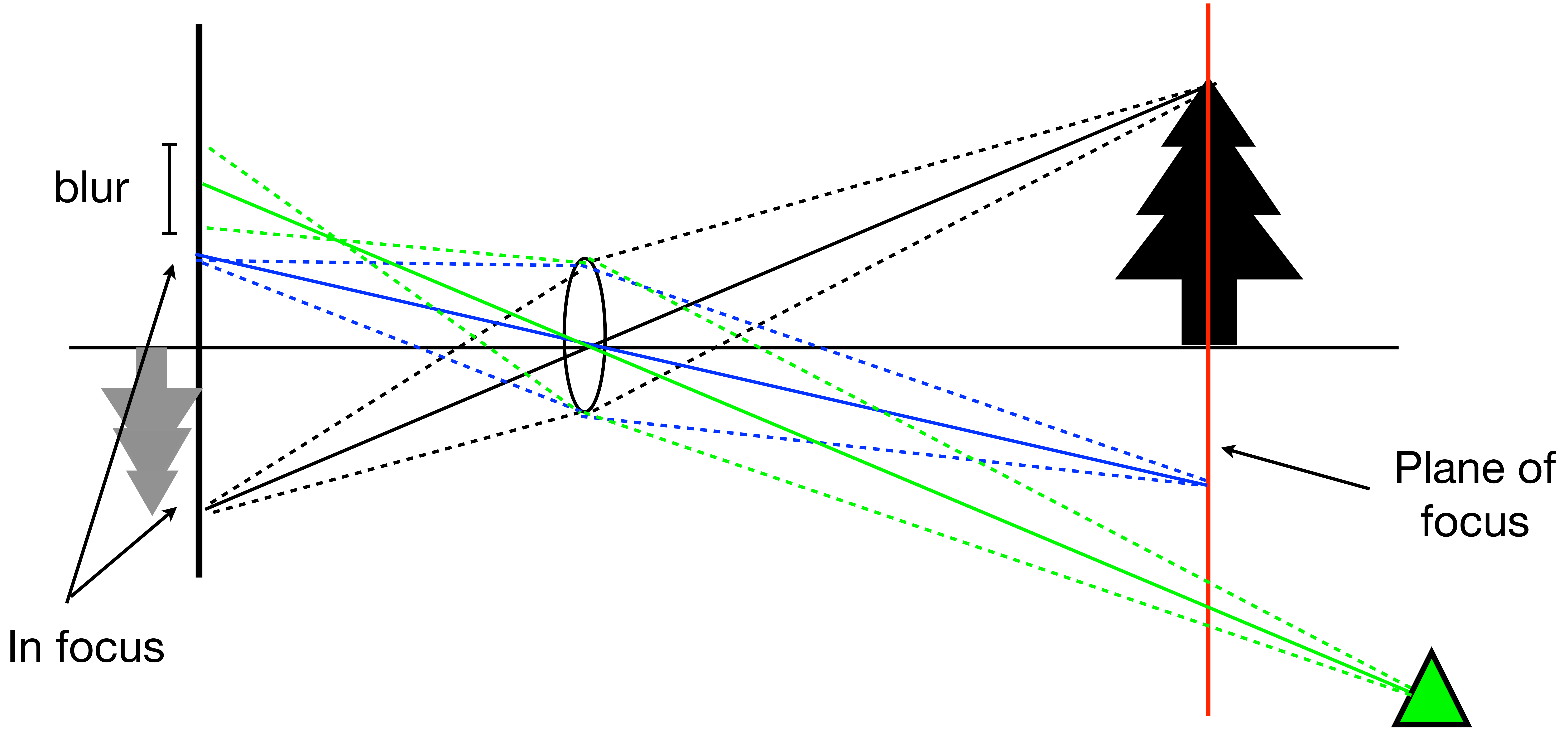


$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Lens Basics

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Lenses focus all rays from a (parallel to lens) plane in the world



Objects off the plane are blurred depending on the distance

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$

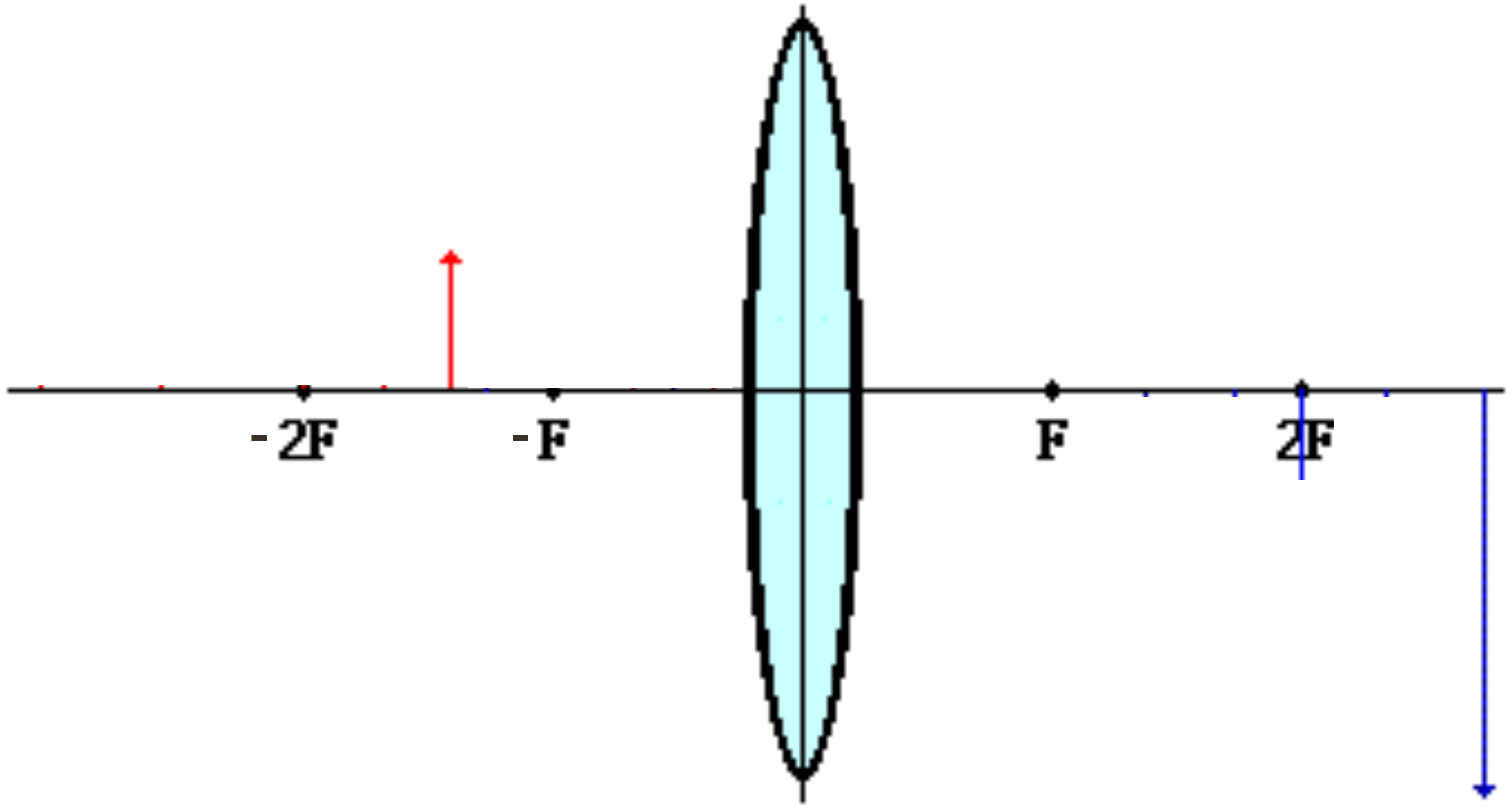
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



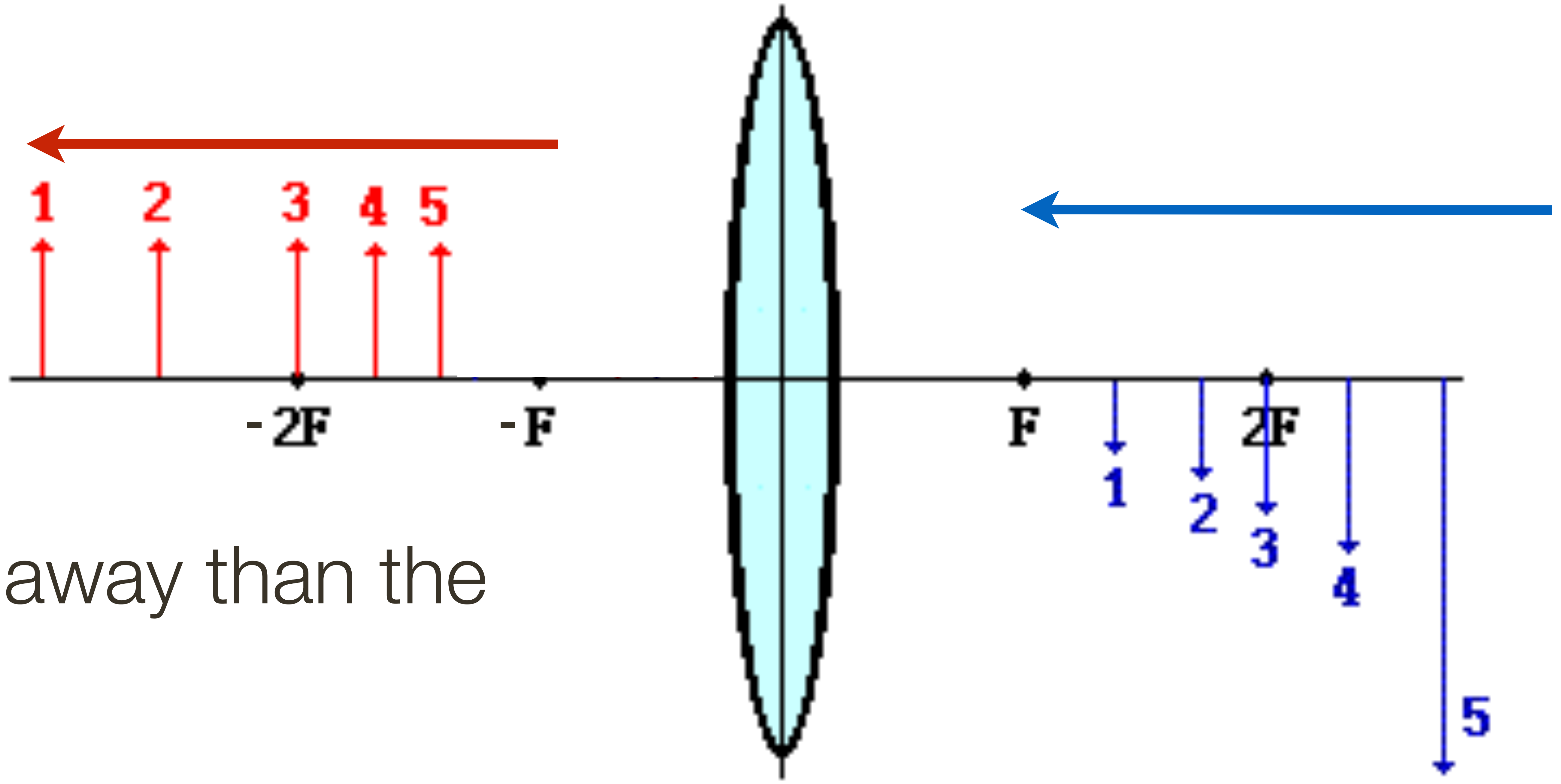
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects **further** away than the **focal length**

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

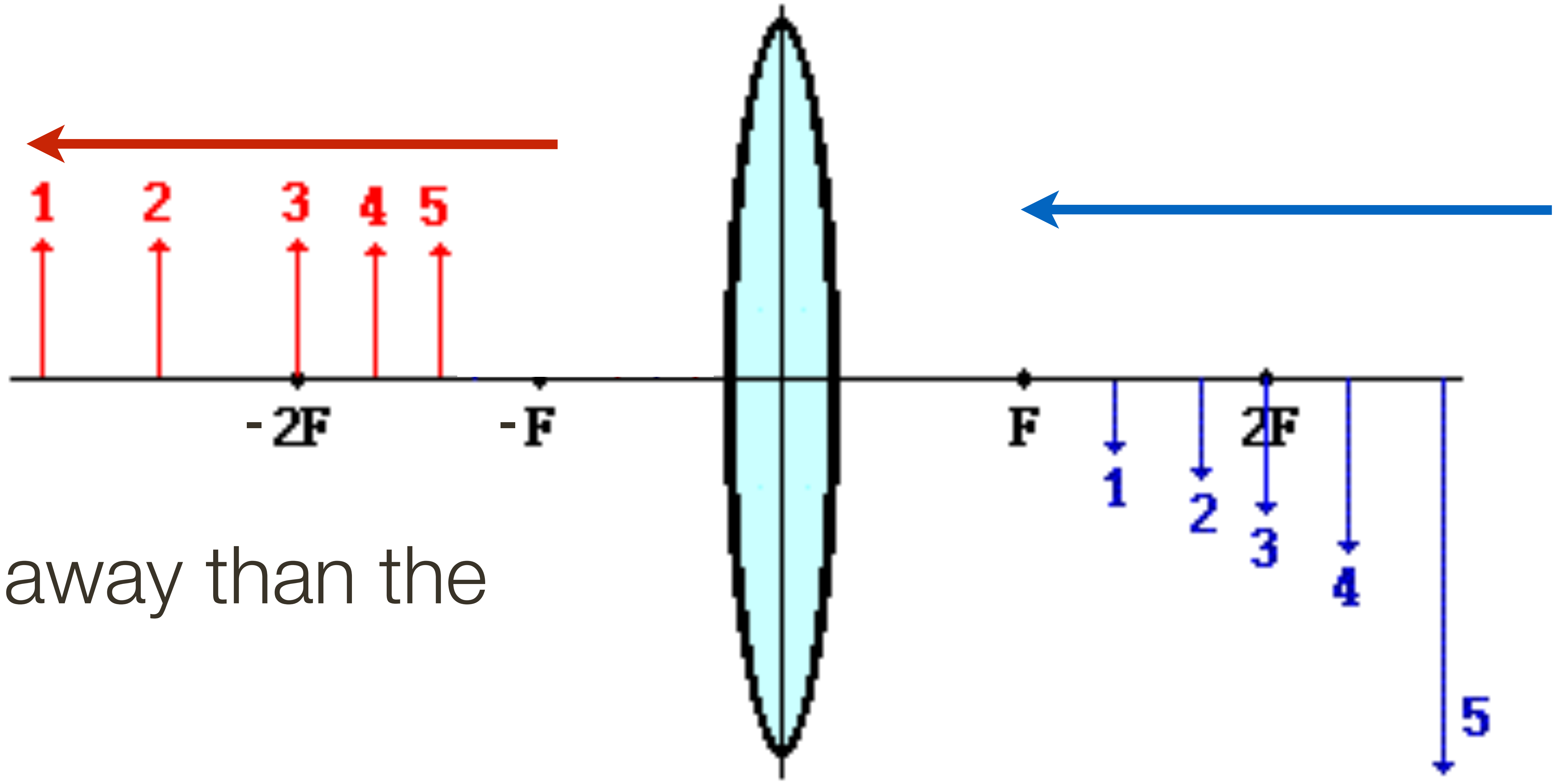
Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$

$$\lim_{z \to -\infty} \frac{zf}{z+f} = f$$

L'Hopital's Rule



Objects **further** away than the **focal length**

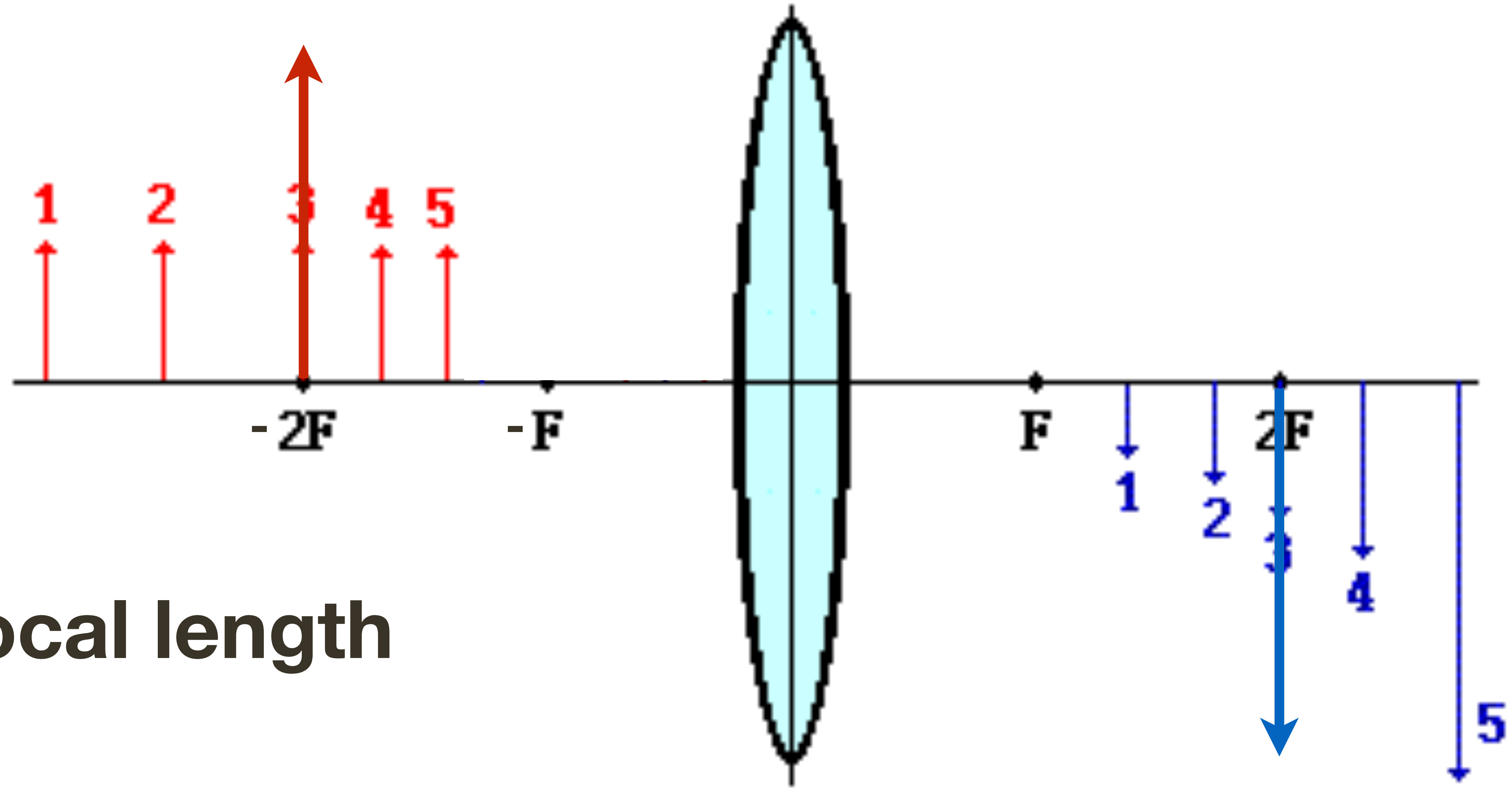
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects at 2 x **focal length**

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

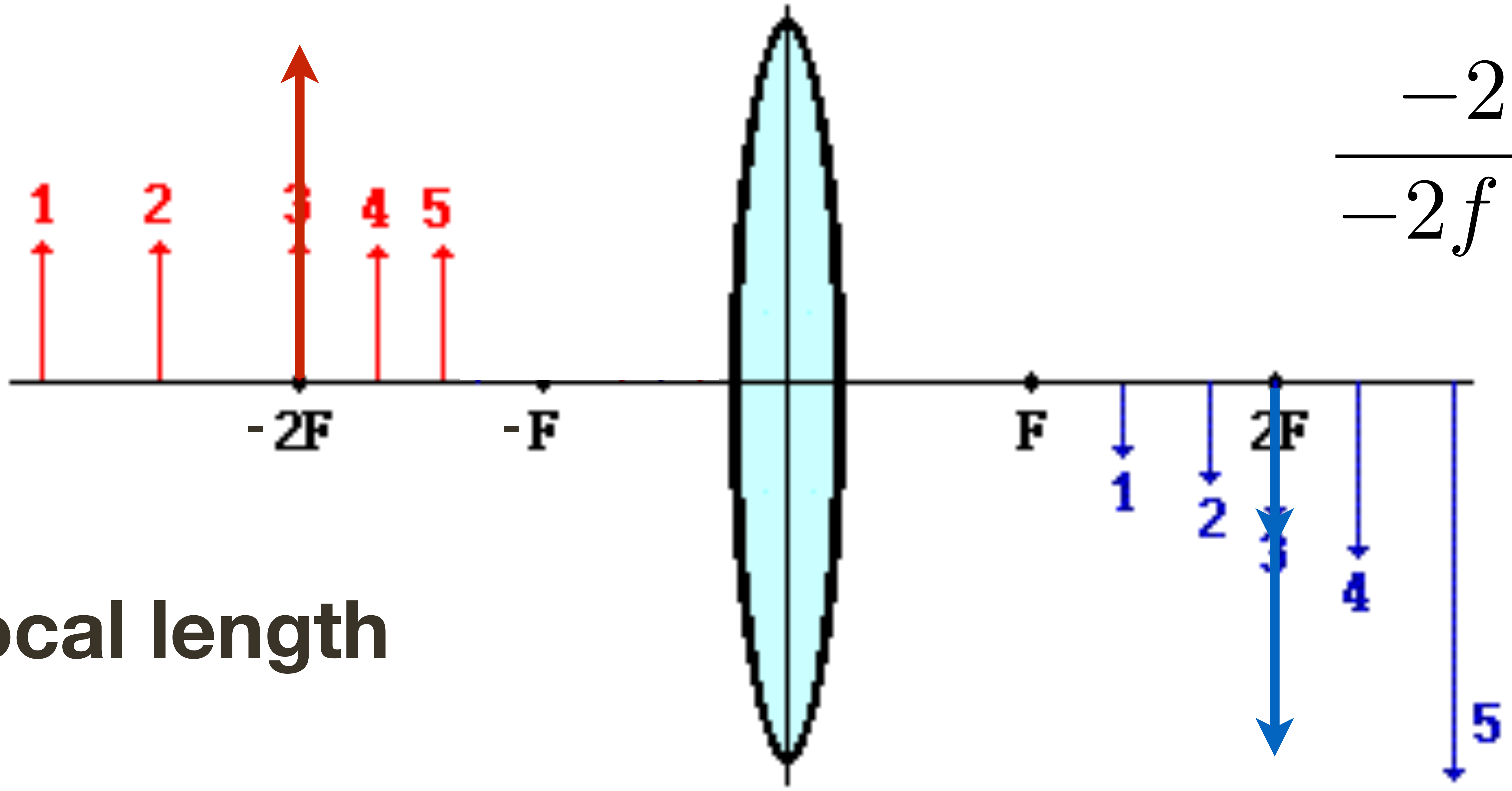
Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$

$$\frac{-2f^2}{-2f+f} = \frac{-2f^2}{-f} = 2f$$



Objects at 2 x **focal length**

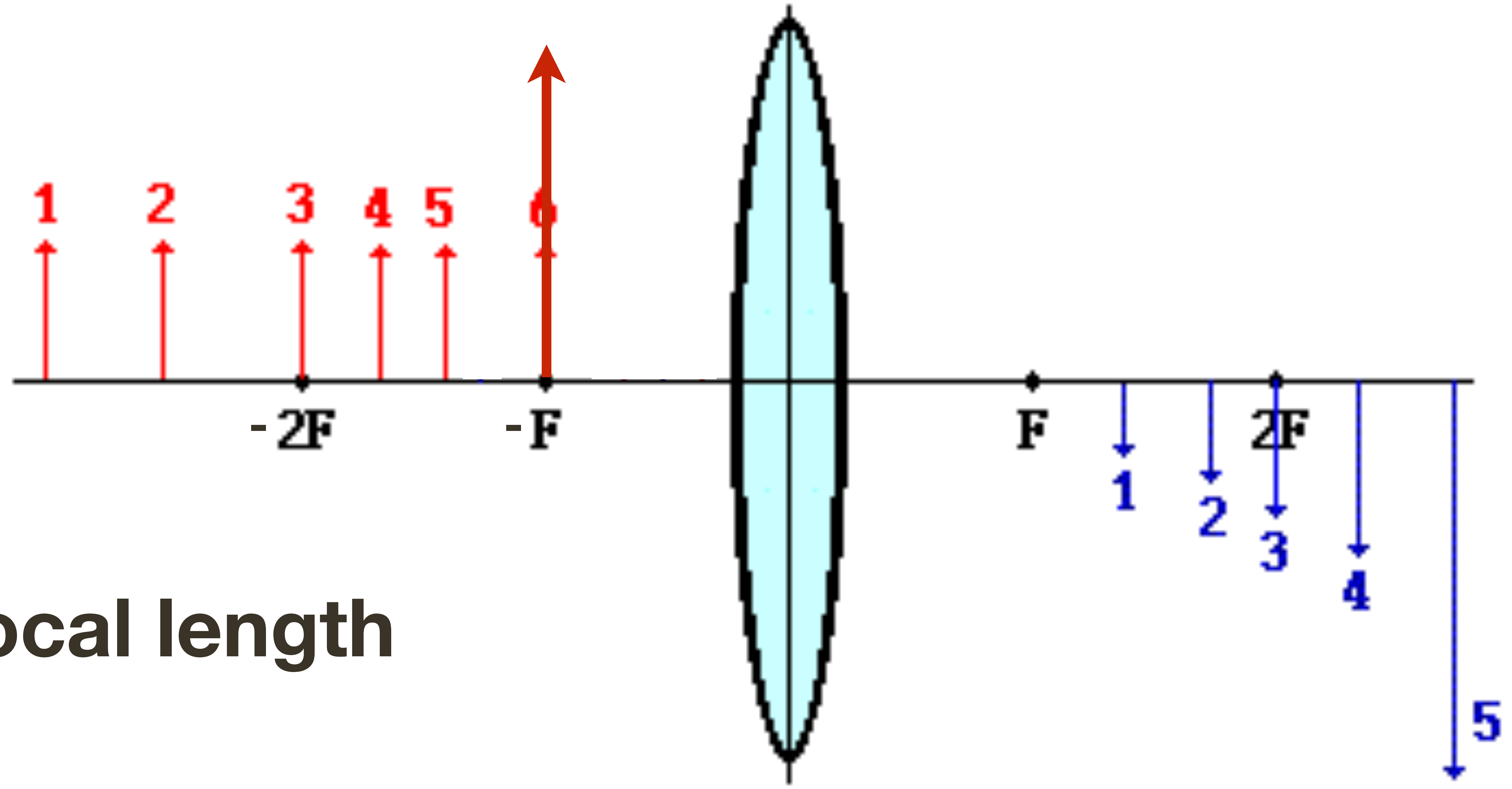
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects at the **focal length**

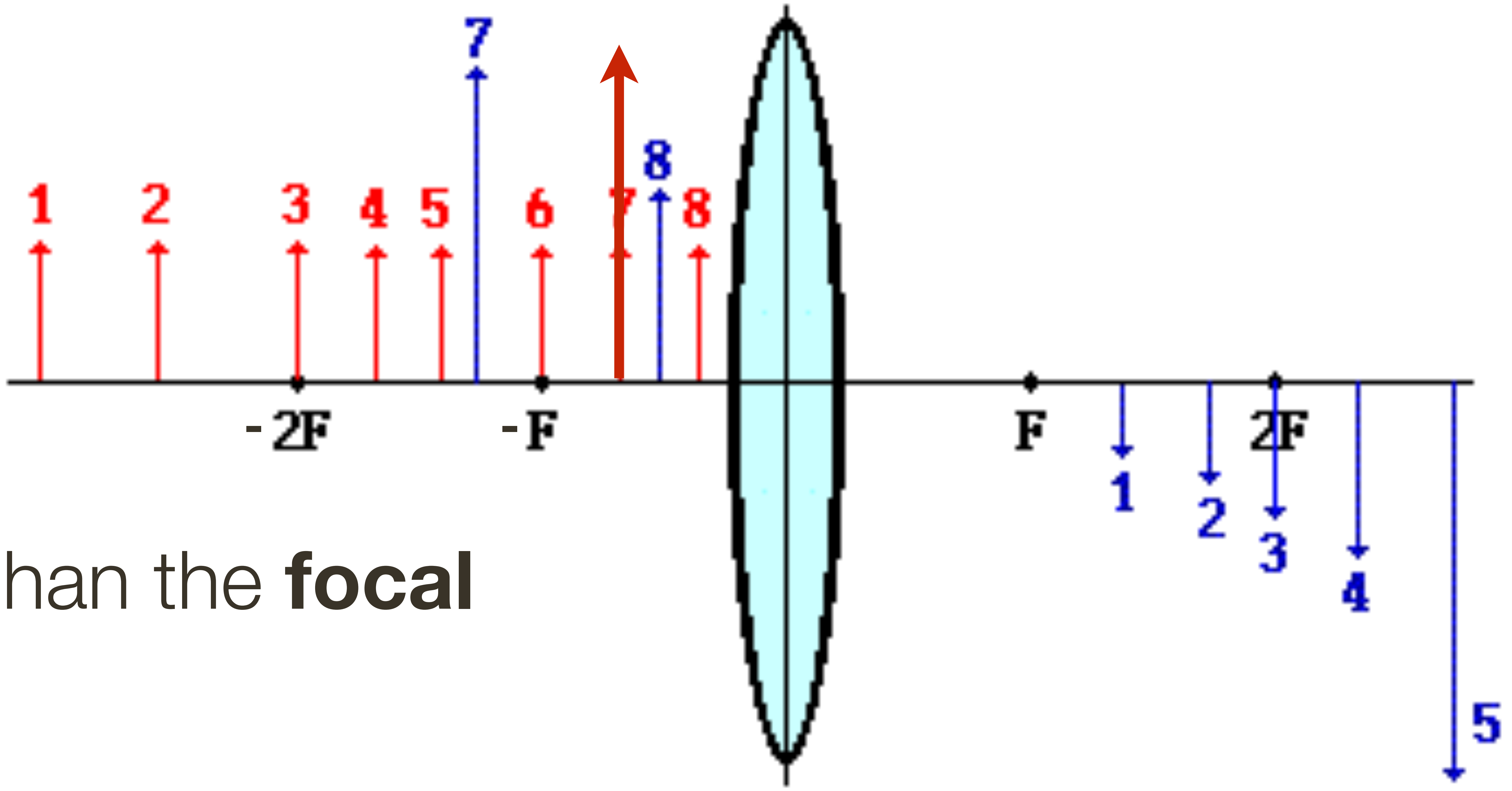
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects **closer** than the **focal length**

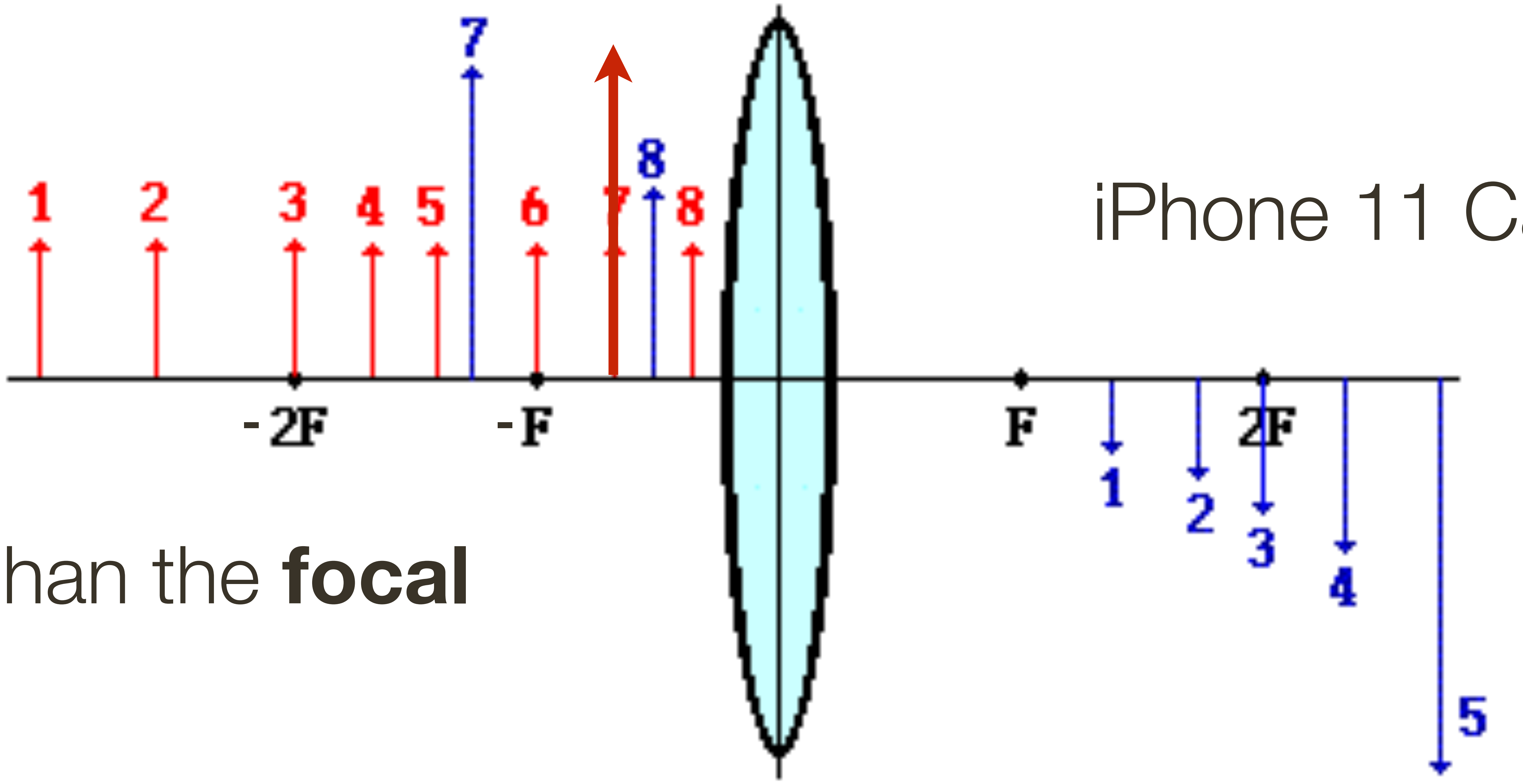
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Perspective Projection + **Thin Lens** Examples

Where would the focusing plane be for various positions of the object?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



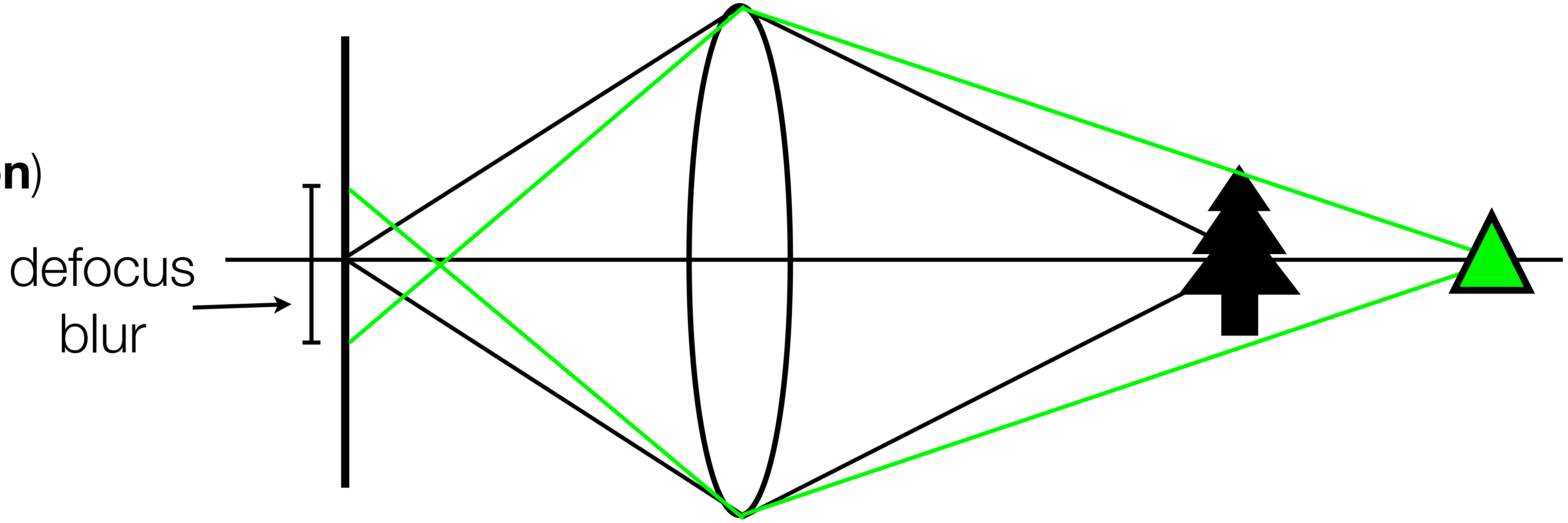
iPhone 11 Camera Lens: **26mm**

Objects **closer** than the **focal length**

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

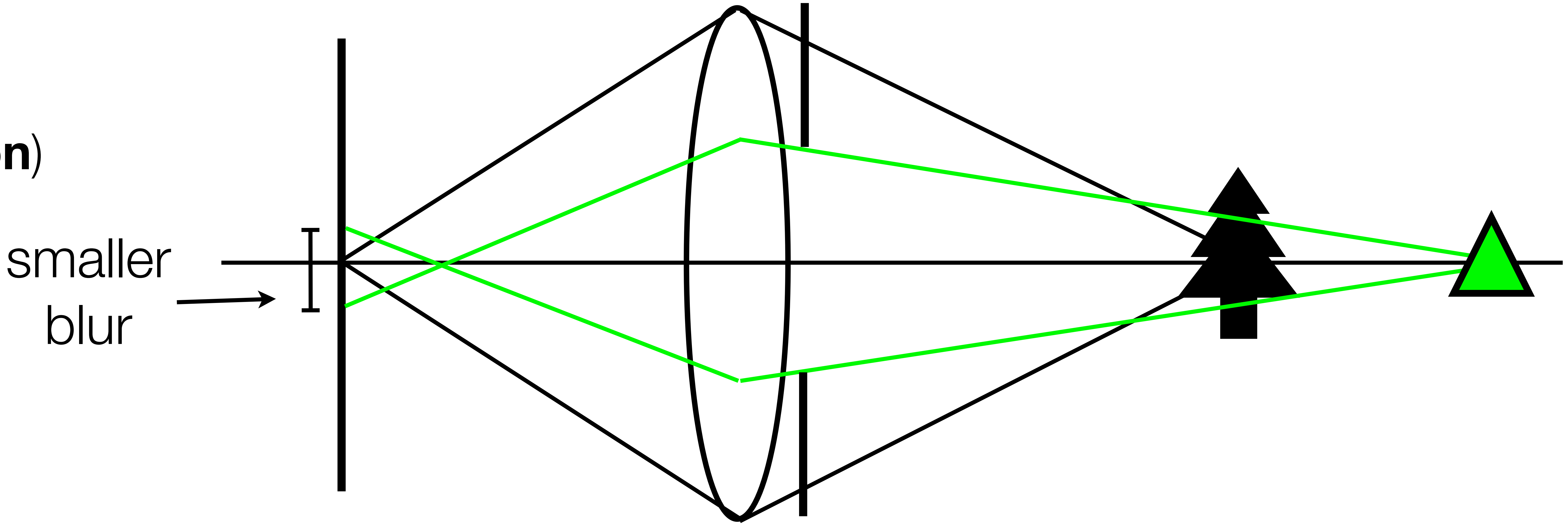
Effect of **Aperture** Size

(also known as
circle of confusion)



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

(also known as
circle of confusion)



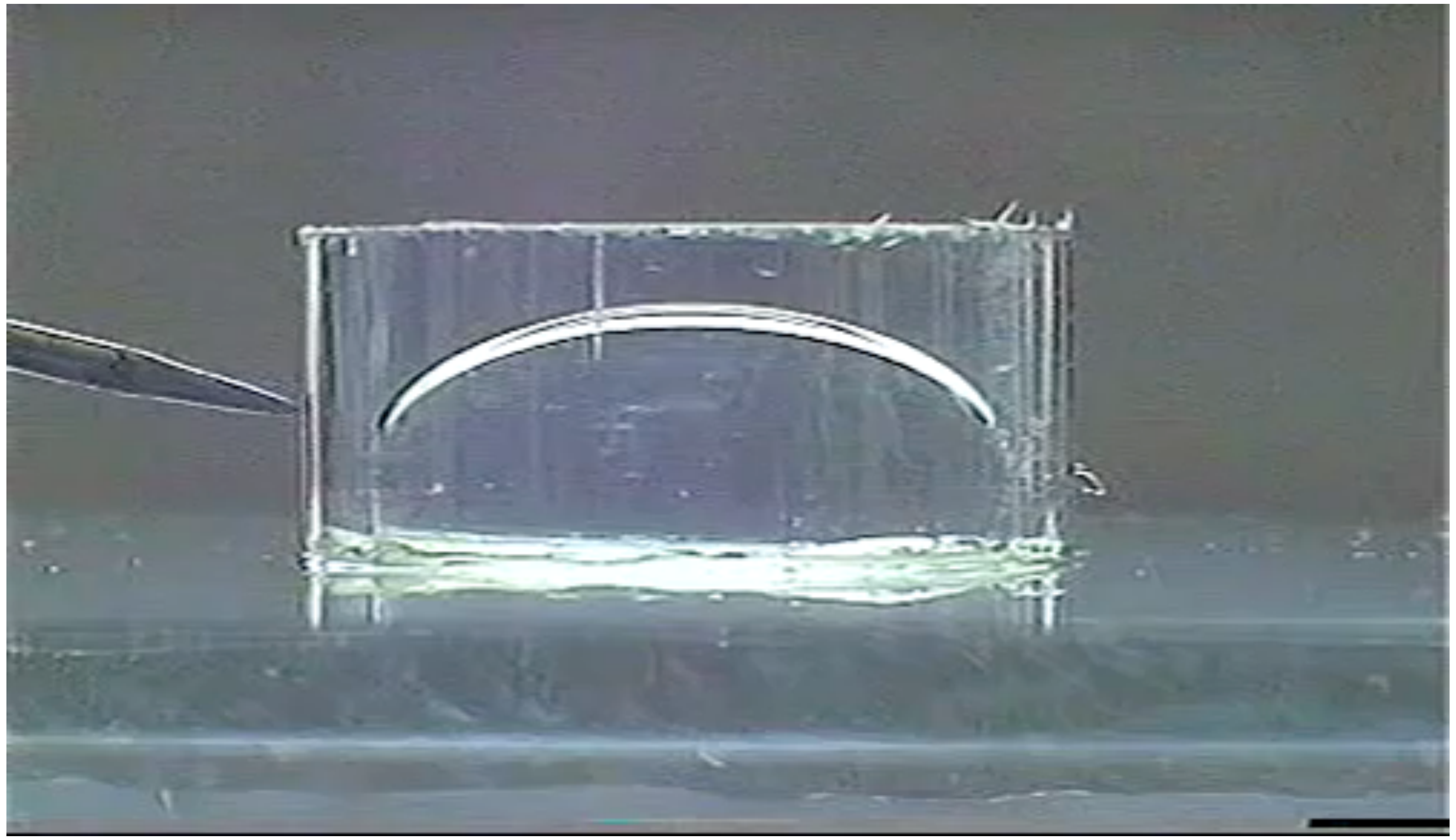
Depth of Field

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

Today's "**fun**" Example #2:

Developed by the French company **Varioptic**, the lenses consist of an oil-based and a water-based fluid sandwiched between glass discs. Electric charge causes the boundary between oil and water to change shape, altering the lens geometry and therefore the lens focal length

The intended applications are: **auto-focus** and **image stabilization**. No moving parts. Fast response. Minimal power consumption.

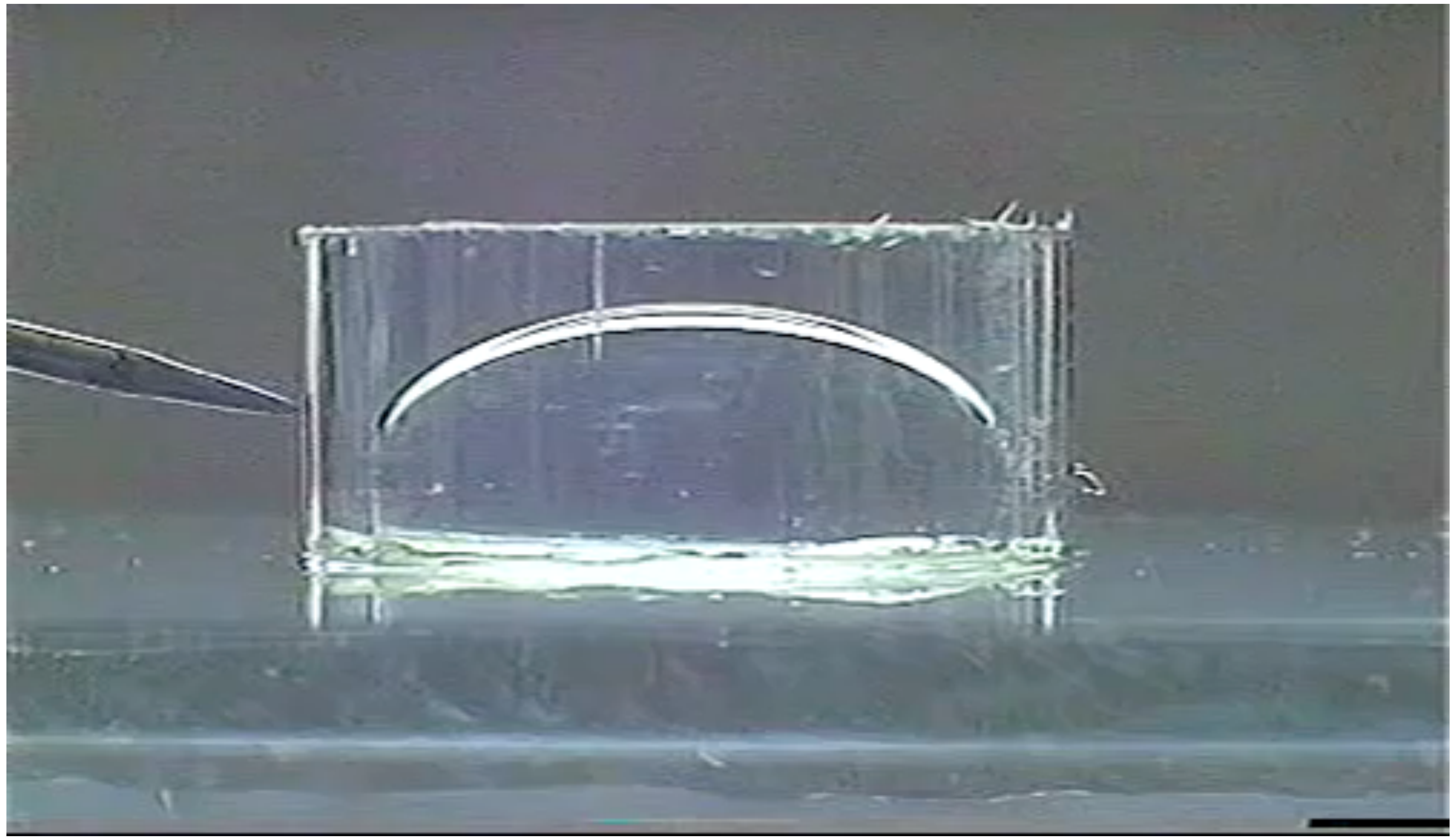


Video Source: https://www.youtube.com/watch?v=2c6lCdDFOY8

Today's "**fun**" Example #2:

Developed by the French company **Varioptic**, the lenses consist of an oil-based and a water-based fluid sandwiched between glass discs. Electric charge causes the boundary between oil and water to change shape, altering the lens geometry and therefore the lens focal length

The intended applications are: **auto-focus** and **image stabilization**. No moving parts. Fast response. Minimal power consumption.



Video Source: https://www.youtube.com/watch?v=2c6lCdDFOY8

Today's "**fun**" Example #2:

Electrostatic field between the column of water and the electron (other side of power supply attached to the pipe) — see full video for complete explanation

Video Source: https://www.youtube.com/watch?v=NjLJ77IuBdM

Today's "**fun**" Example #2:

Electrostatic field between the column of water and the electron (other side of power supply attached to the pipe) — see full video for complete explanation

Video Source: https://www.youtube.com/watch?v=NjLJ77IuBdM

Today's "**fun**" Example #2:

As one example, in 2010, **Cognex** signed a license agreement with Varioptic to add auto-focus capability to it DataMan line of industrial ID readers (press release May 29, 2012)

Video Source: https://www.youtube.com/watch?v=EU8LXxip1NM

Today's "**fun**" Example #2:

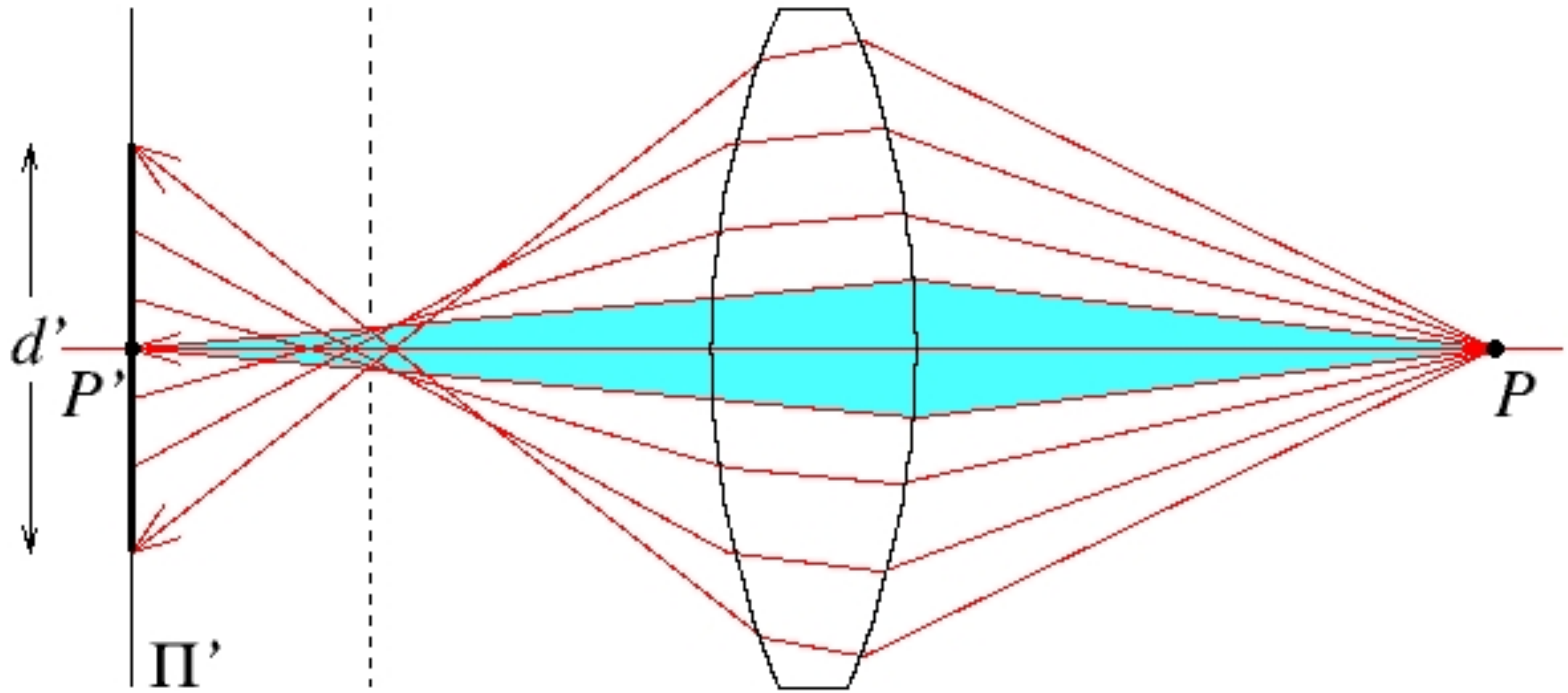
As one example, in 2010, **Cognex** signed a license agreement with Varioptic to add auto-focus capability to it DataMan line of industrial ID readers (press release May 29, 2012)

Video Source: https://www.youtube.com/watch?v=EU8LXxip1NM

Real Lenses

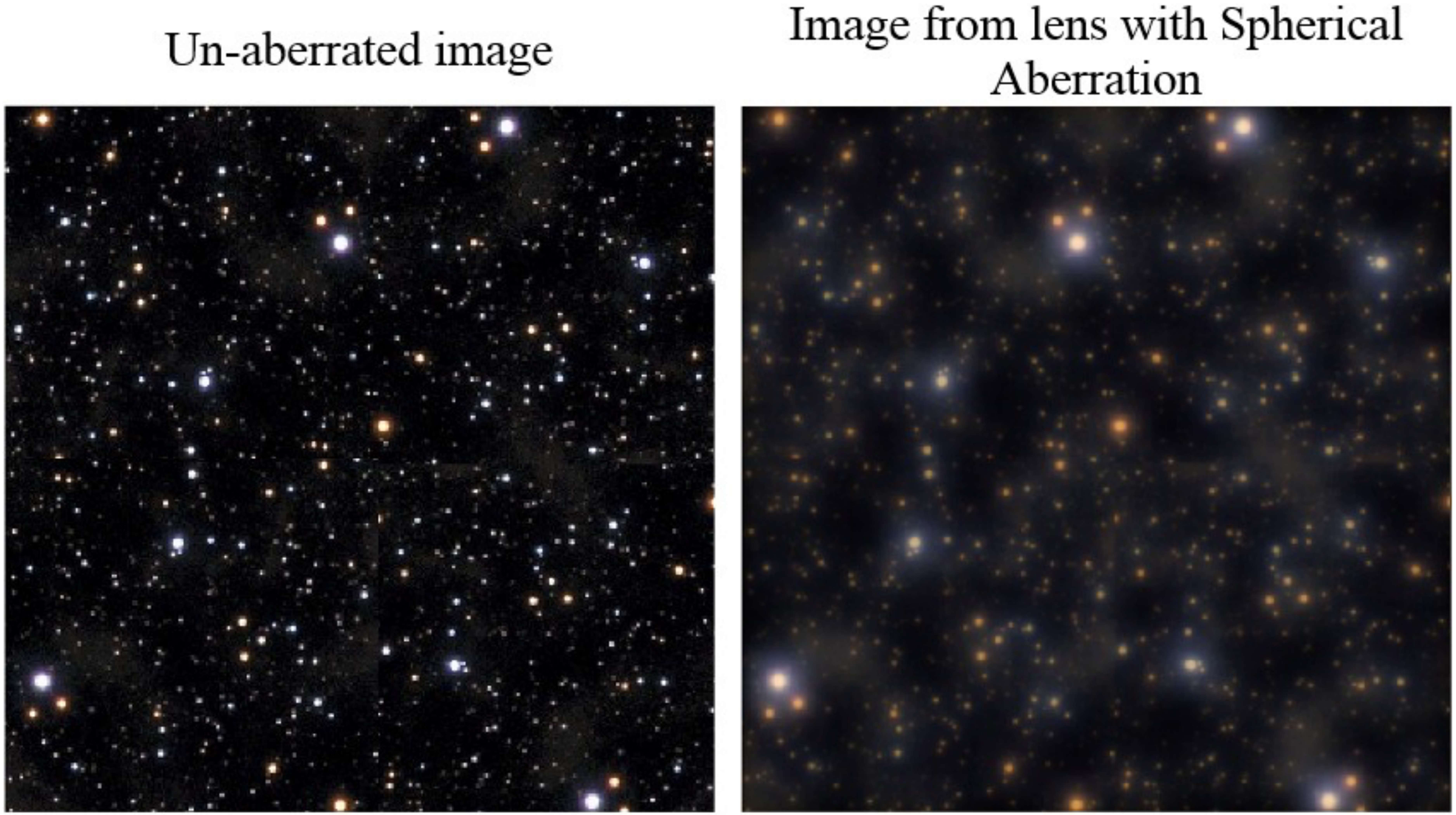
- Real Lenses have multiple stages of positive and negative elements with differing refractive indices
- This can help deal with issues such as chromatic aberration (different colours bent by different amounts), vignetting (light fall off at image edge) and sharp imaging across the zoom range

Spherical **Aberration**

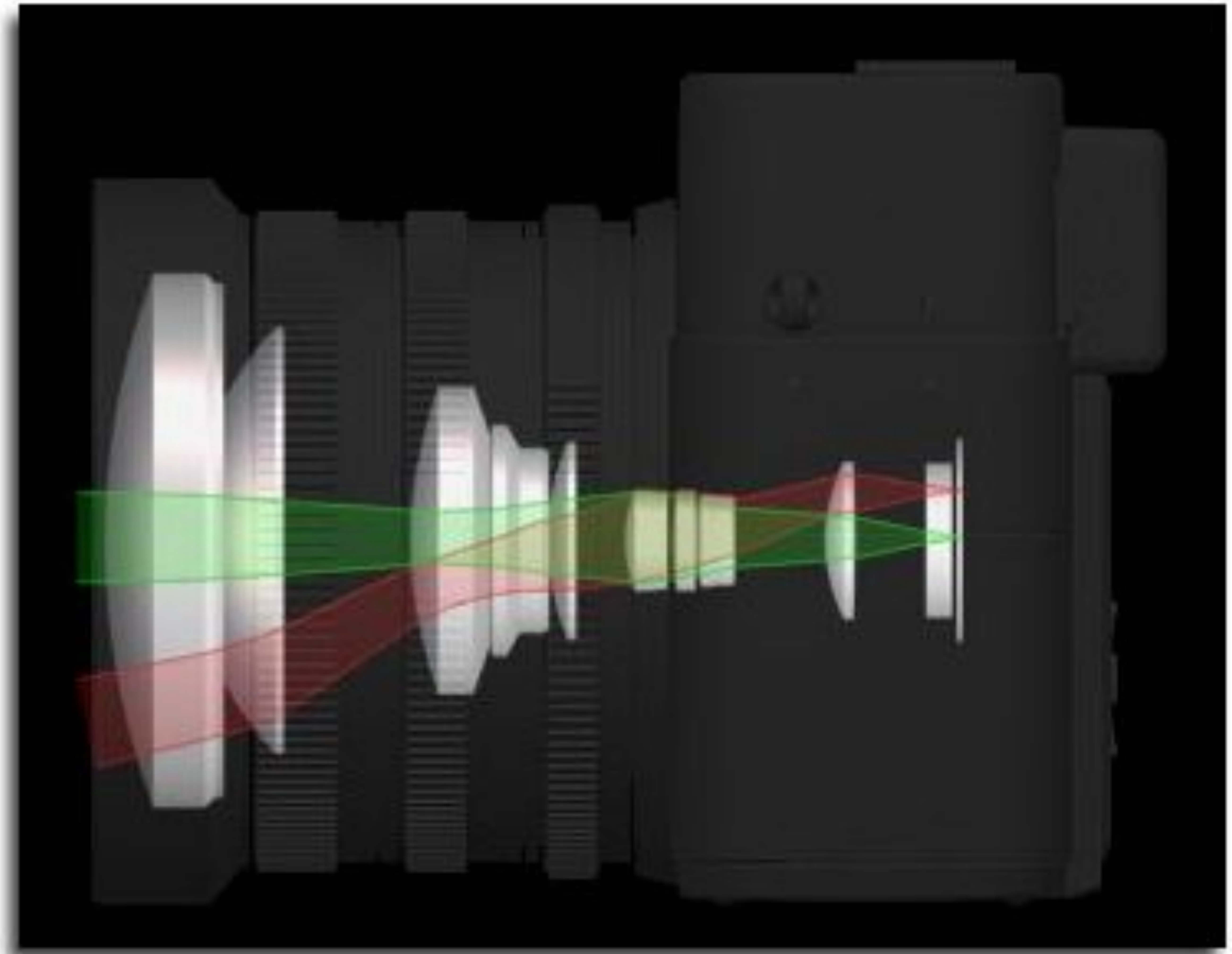


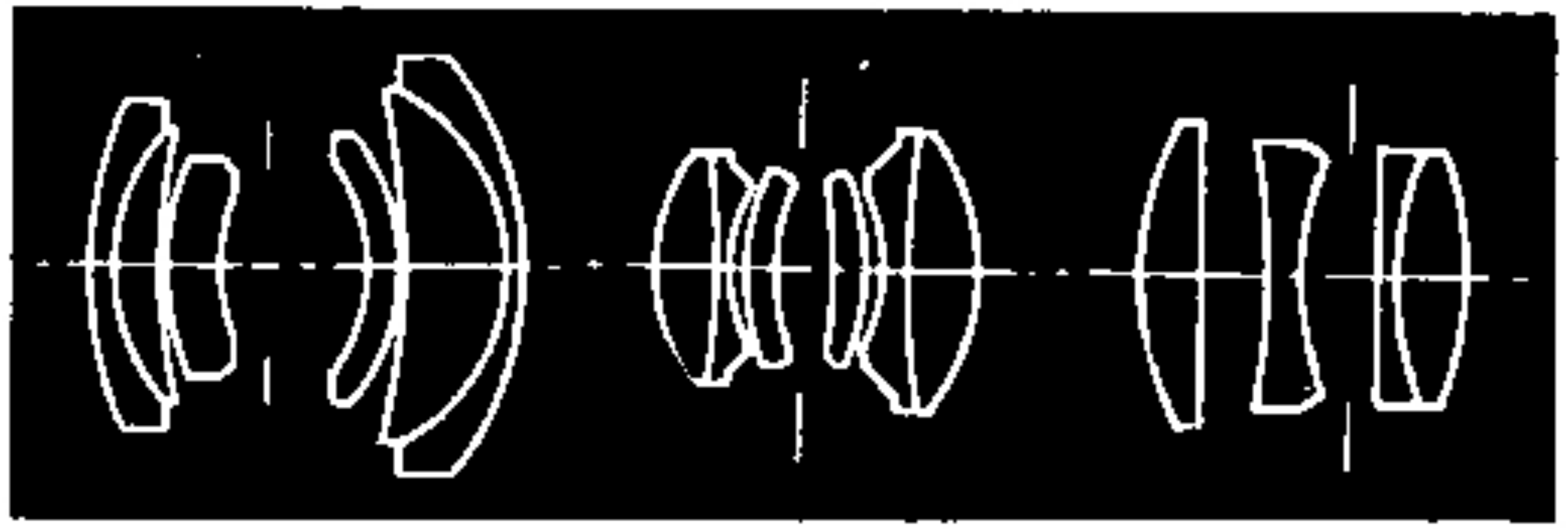
Forsyth & Ponce (1st ed.) Figure 1.12a

Spherical **Aberration**



Compound **Lens Systems**

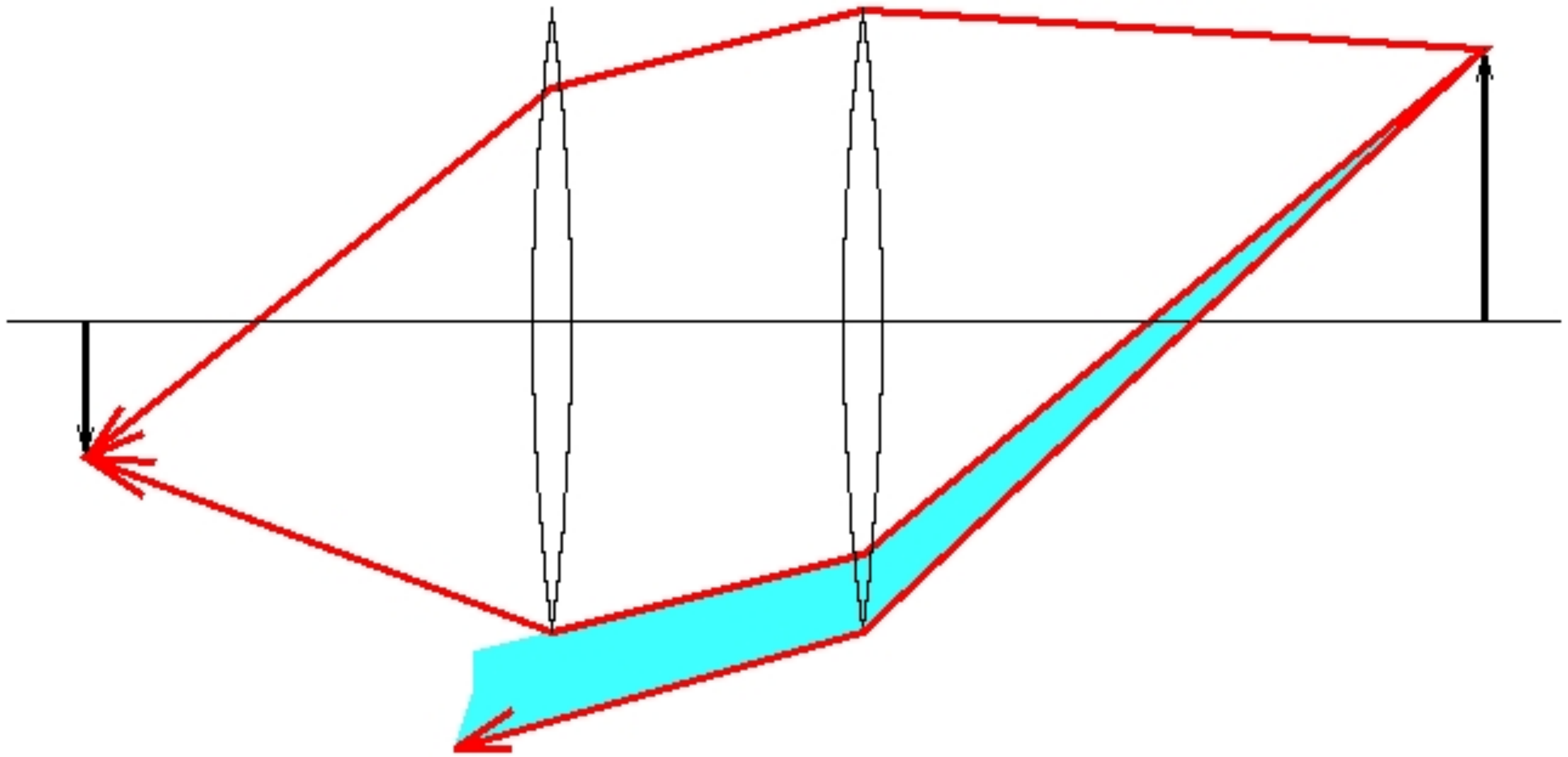




A modern camera lens may contain multiple components, including aspherical elements

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

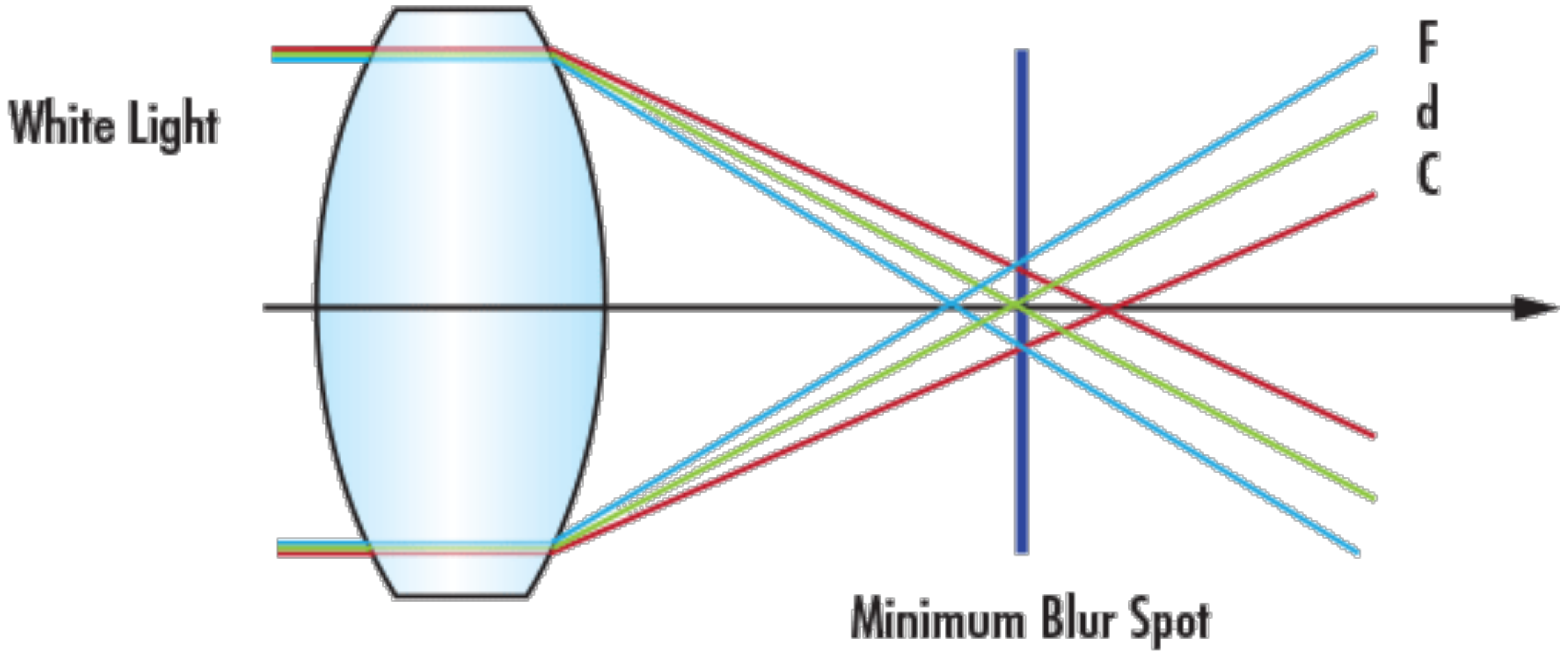


Image Credit: Trevor Darrell

Other (Possibly Significant) **Lens Effects**

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

Scattering at the lens surface

— Some light is reflected at each lens surface

There are other **geometric phenomena/distortions**

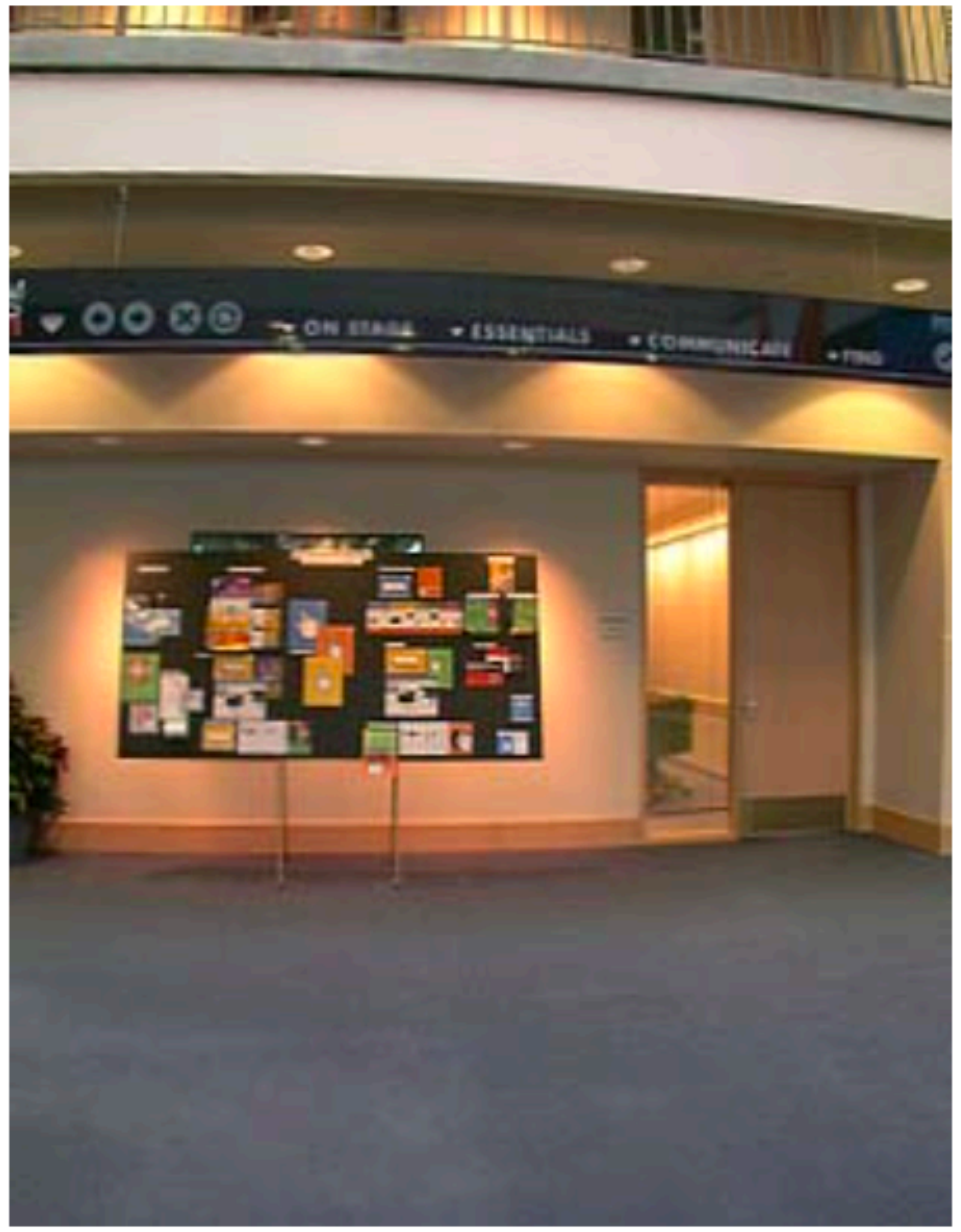
— pincushion distortion

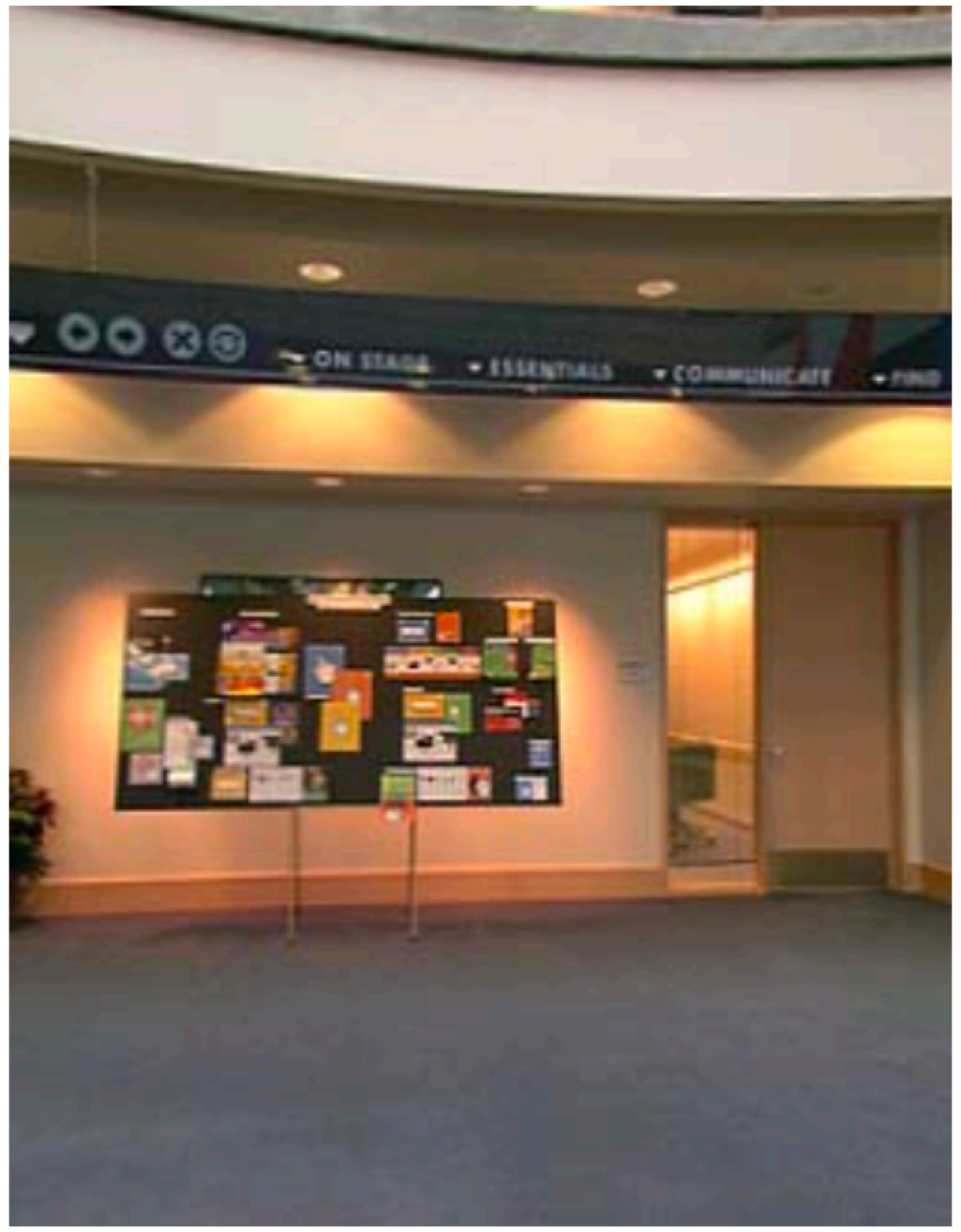
— barrel distortion

— etc

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!

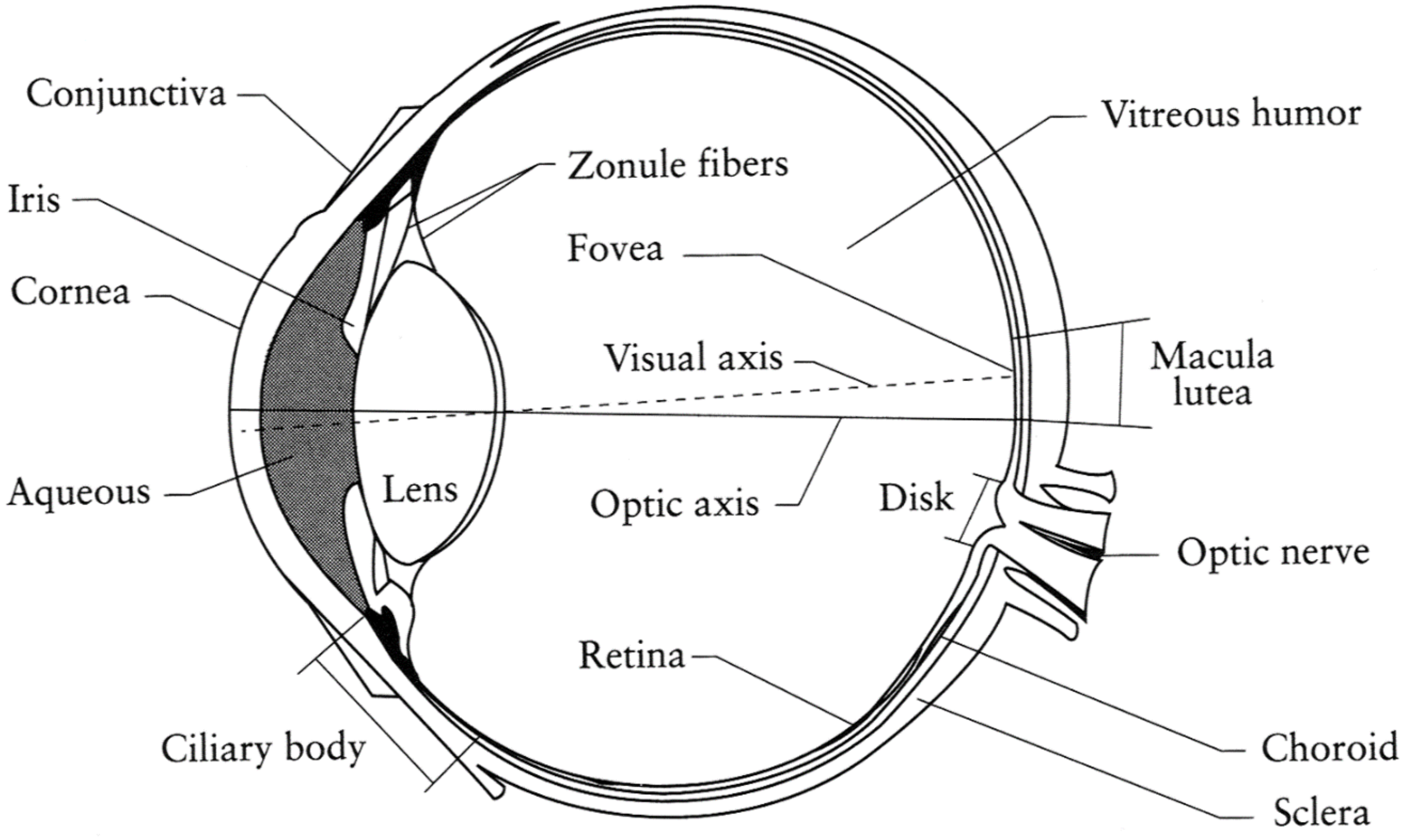
Human Eye

— The eye has an **iris** (like a camera)

— **Focusing** is done by changing shape of lens

— When the eye is properly focused, light from an object outside the eye is imaged on the **retina**

— The retina contains light receptors called **rods** and **cones**

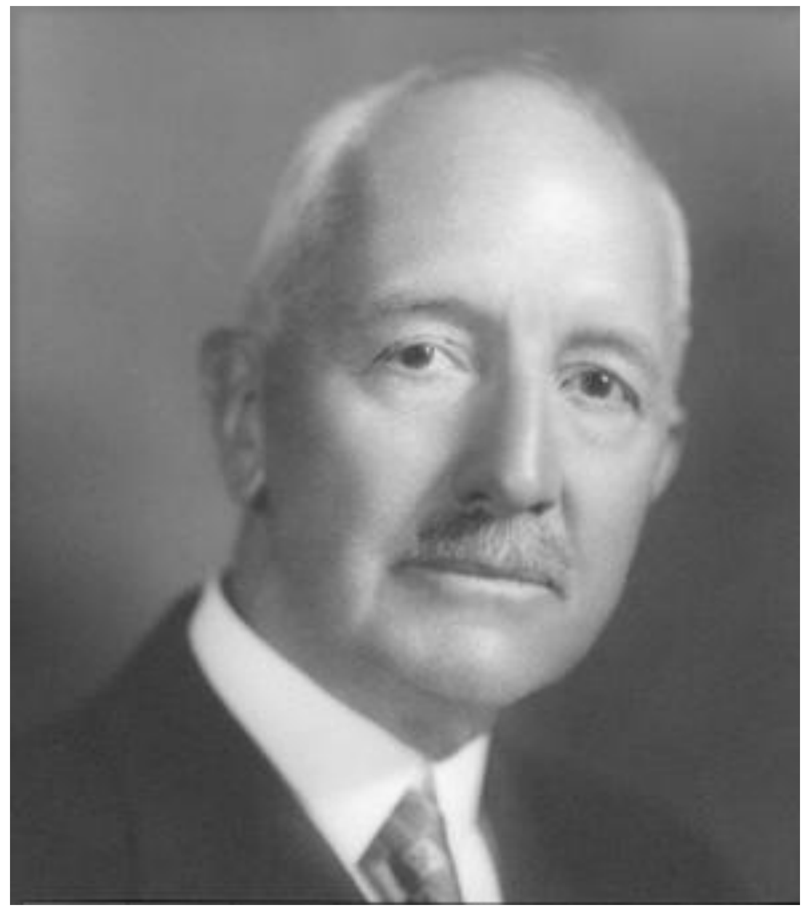


pupil = pinhole / aperture

retina = film / digital sensor

Slide adopted from: Steve Seitz

Fun **Aside**



George M. Stratton

https://io9.gizmodo.com/does-your-brain-really-have-the-power-to-see-the-world-5905180

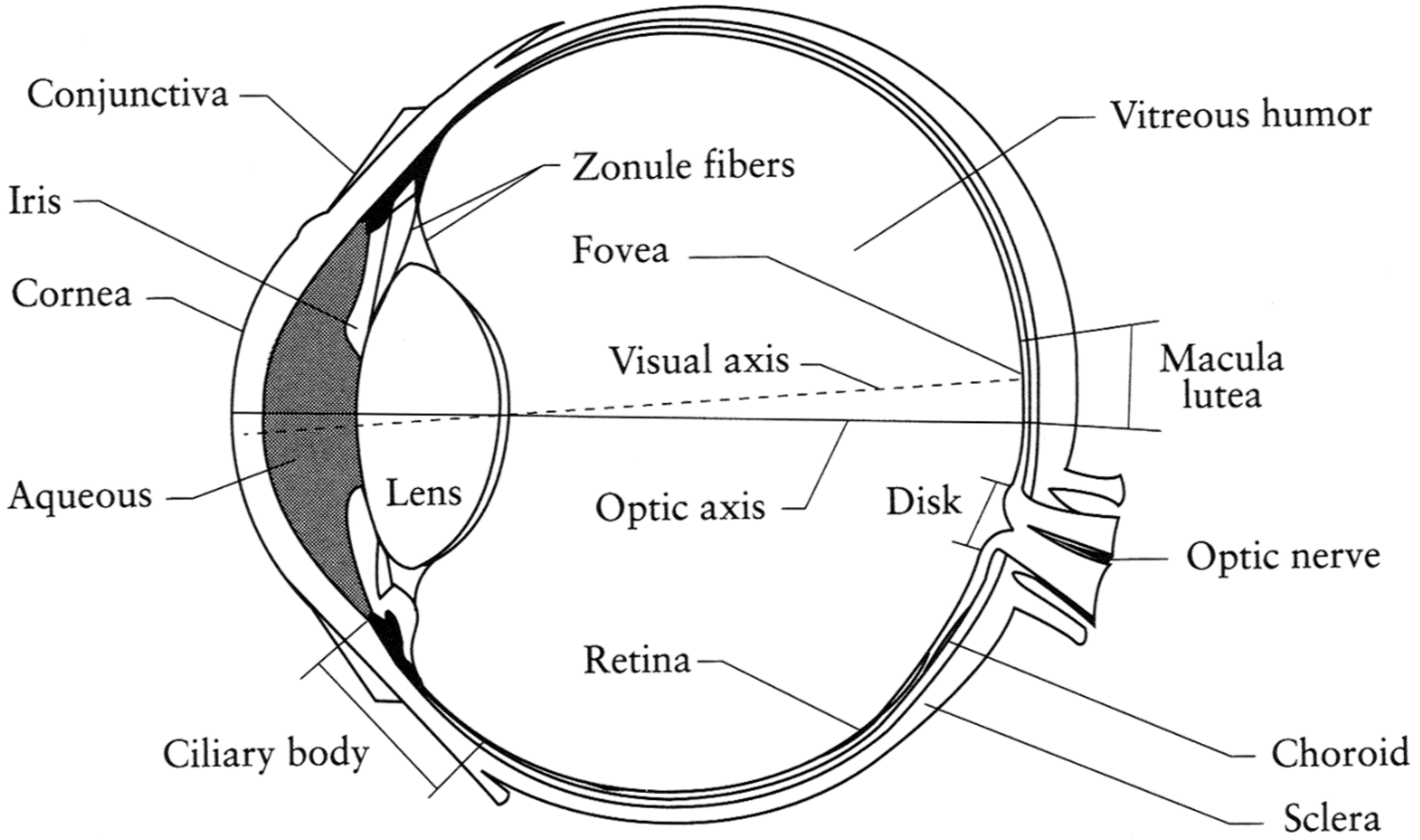
Human Eye

— The eye has an **iris** (like a camera)

— **Focusing** is done by changing shape of lens

— When the eye is properly focused, light from an object outside the eye is imaged on the **retina**

— The retina contains light receptors called **rods** and **cones**



pupil = pinhole / aperture

retina = film / digital sensor

Slide adopted from: Steve Seitz

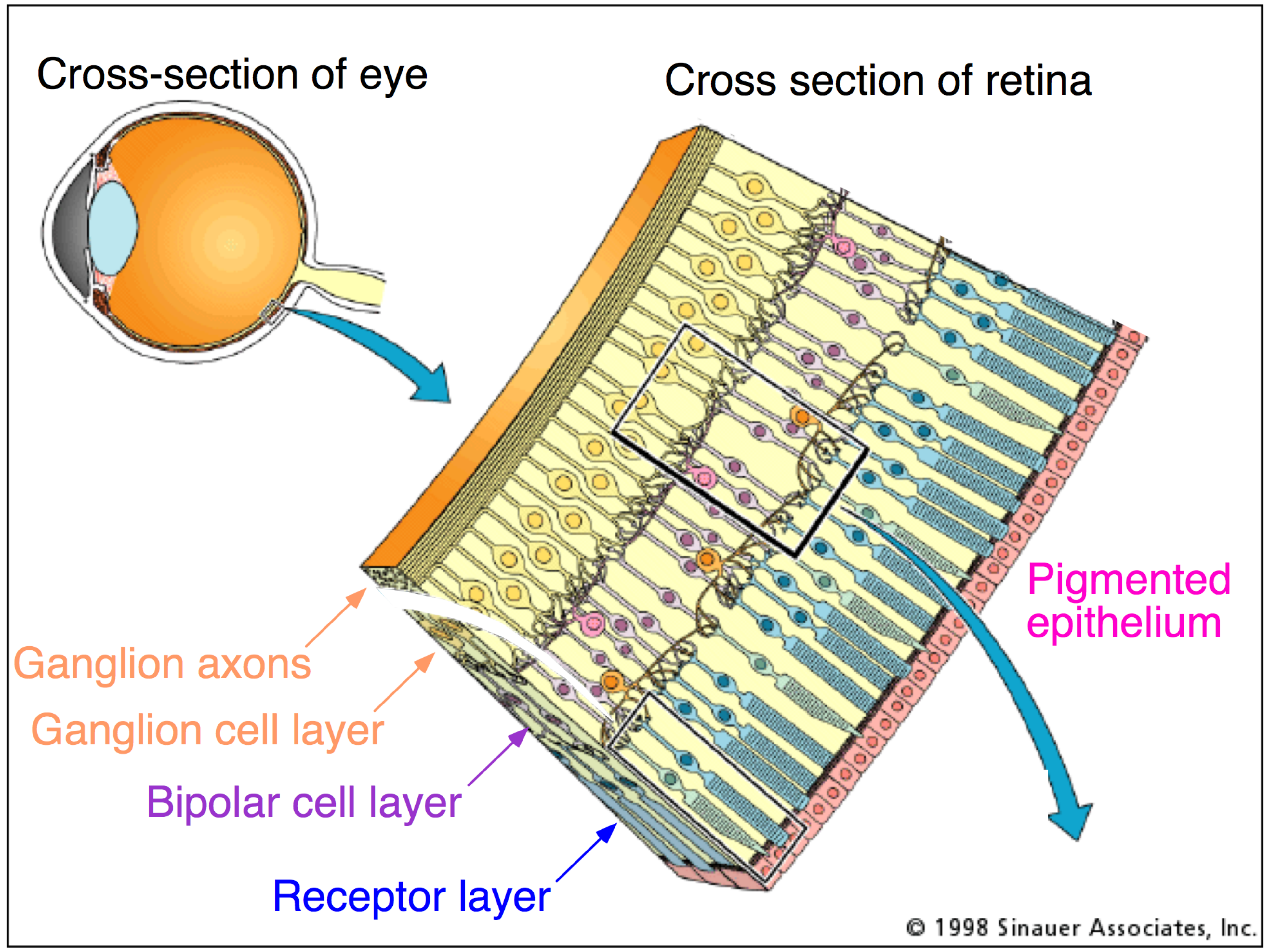
Human Eye

— The eye has an **iris** (like a camera)

— **Focusing** is done by changing shape of lens

— When the eye is properly focused, light from an object outside the eye is imaged on the **retina**

— The retina contains light receptors called **rods** and **cones**



pupil = pinhole / aperture

retina = film / digital sensor

Slide adopted from: Steve Seitz

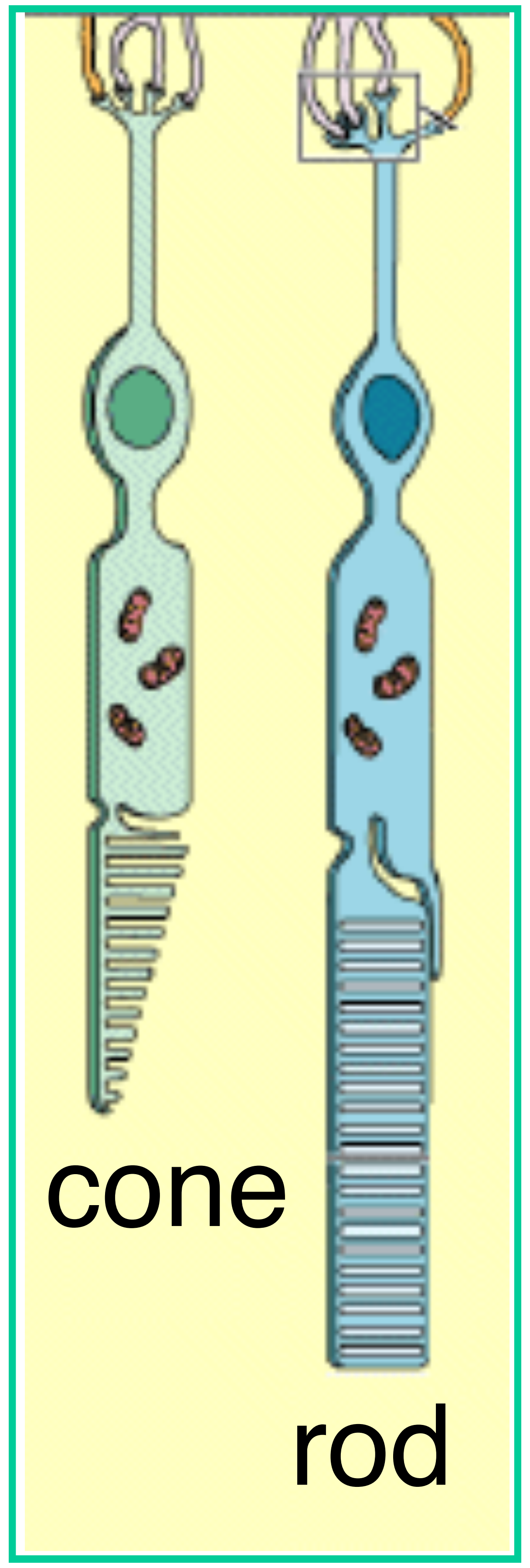
Two-types of **Light Sensitive Receptors**

Rods

75-150 million rod-shaped receptors
not involved in color vision, gray-scale vision only
operate at night
highly sensitive, can responding to a single photon
yield relatively poor spatial detail

Cones

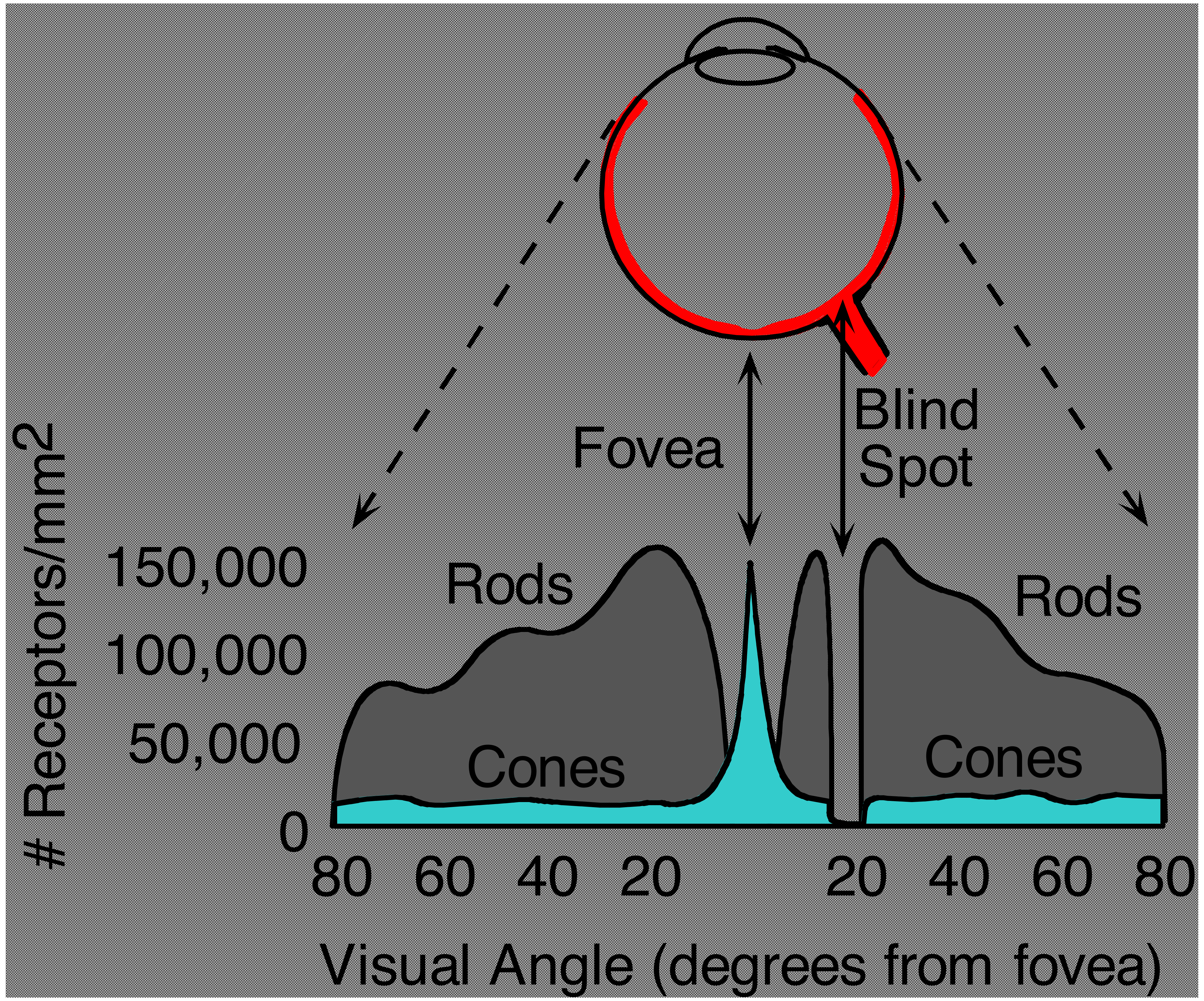
6-7 million cone-shaped receptors
color vision
operate in high light
less sensitive
yield higher resolution



Slide adopted from: James Hays

Human Eye

Density of rods and cones



Slide adopted from: James Hays

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

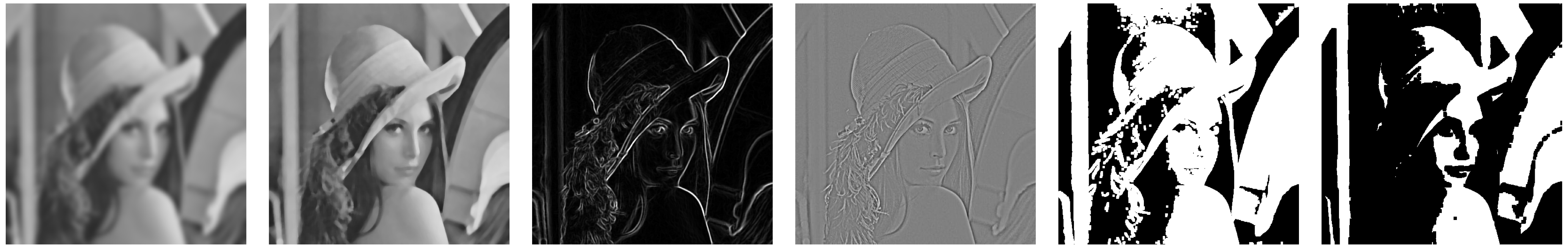
— Thin lens equation

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

— The **human eye** functions much like a camera

THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 3: Image Filtering

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

Goal

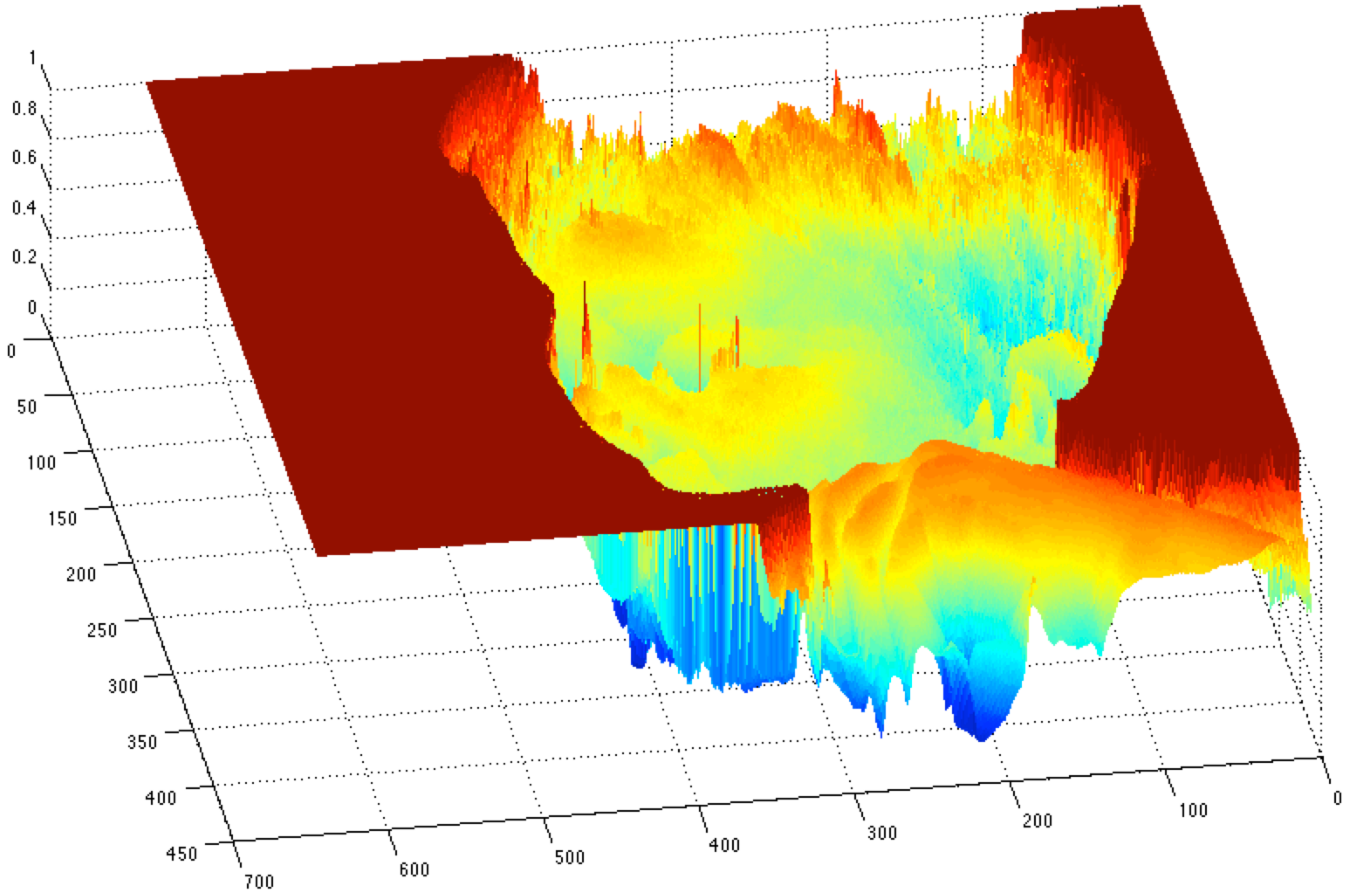
1. Learn how to mathematically describe image processing
2. Basic building blocks

Image as a **2D Function**

A (grayscale) image is a 2D function

grayscale image

$I(X, Y)$



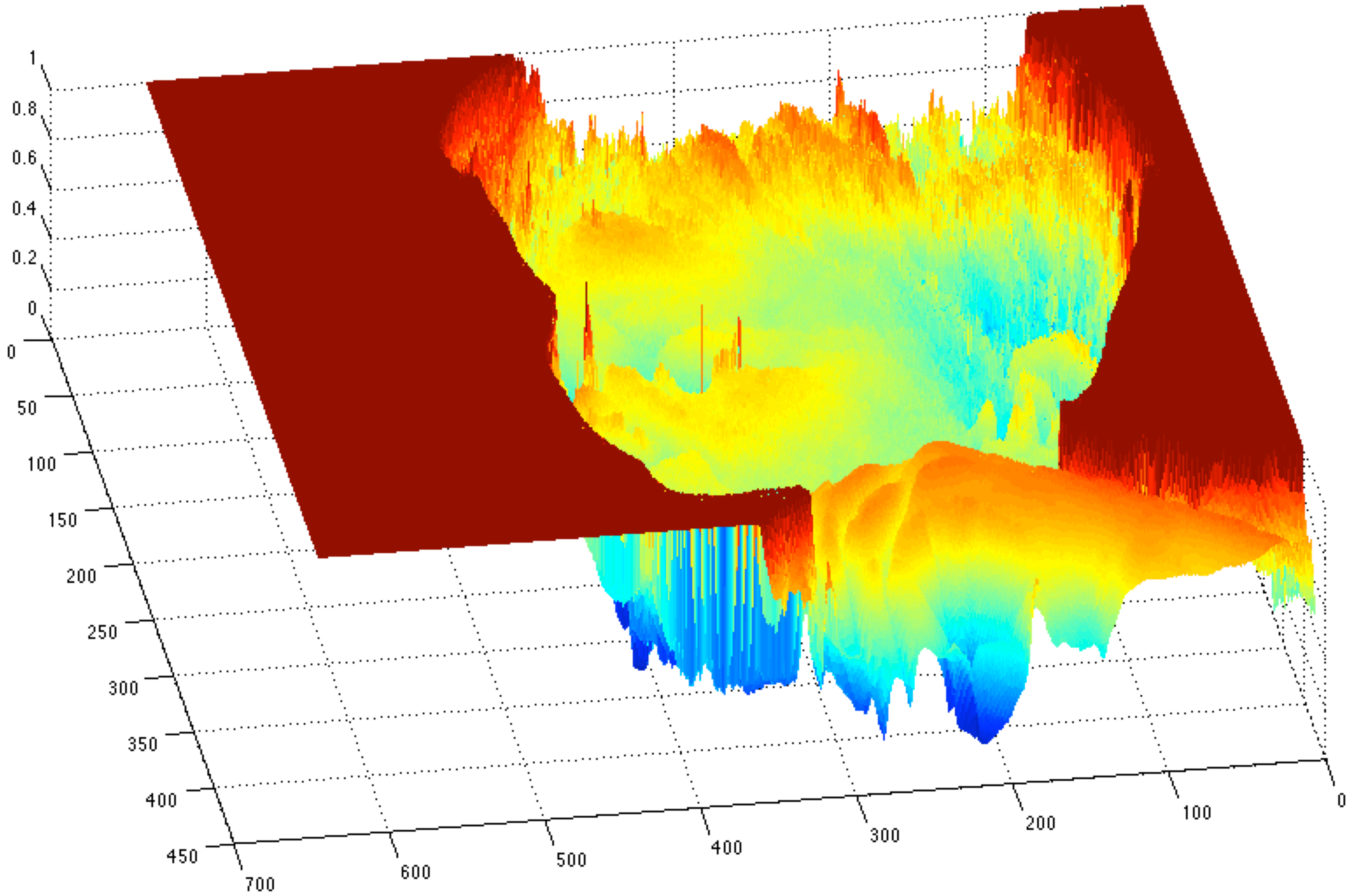
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Image as a **2D Function**

A (grayscale) image is a 2D function

grayscale image

$I(X, Y)$



domain: $(X, Y) \in ([1, width], [1, hight])$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

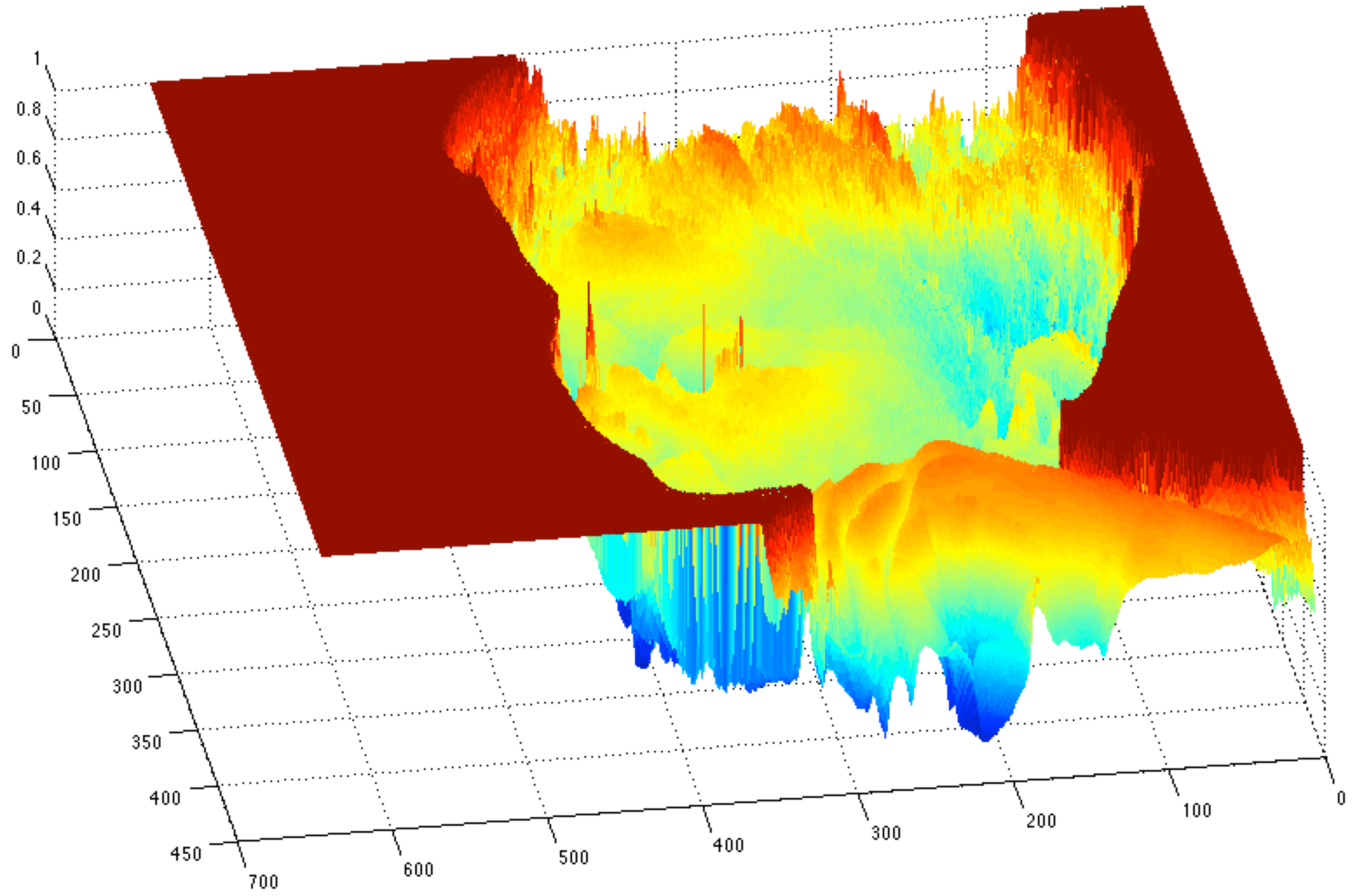
Image as a **2D Function**

A (grayscale) image is a 2D function

grayscale image

What is the **range** of the image function?

$I(X, Y)$



domain: $(X, Y) \in ([1, width], [1, hight])$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Image as a **2D Function**

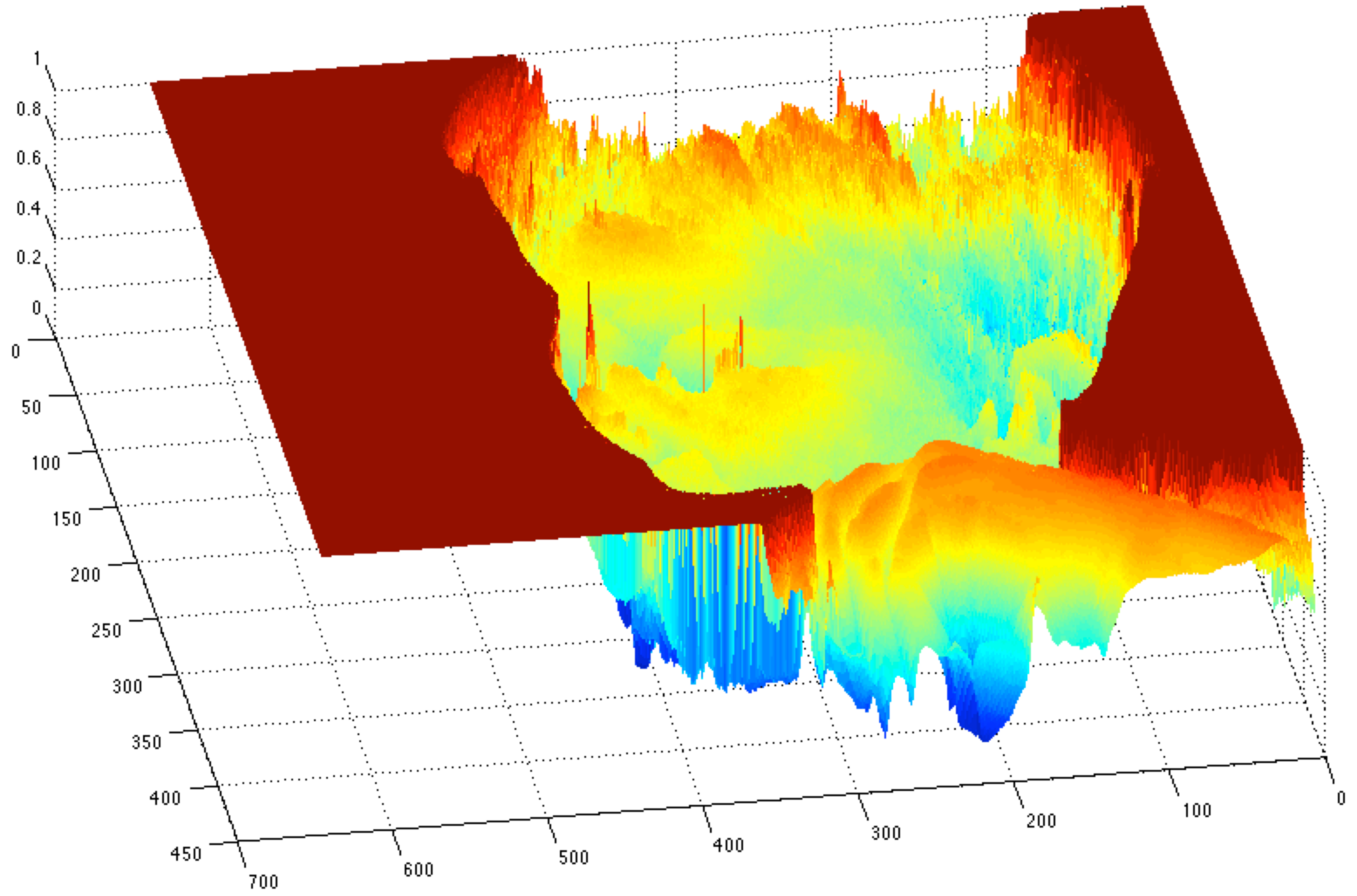
A (grayscale) image is a 2D function

grayscale image

What is the **range** of the image function?

$$I(X,Y) \in [0,255] \in \mathbb{Z}$$

$I(X,Y)$



domain: $(X,Y) \in ([1, width], [1, hight])$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Adding two Images

Since images are functions, we can perform operations on them, e.g., **average**

$I(X,Y)$

$G(X,Y)$

$$\frac{I(X,Y)}{2} + \frac{G(X,Y)}{2}$$

Adding two Images

$$a = \frac{I(X,Y)}{2} + \frac{G(X,Y)}{2}$$

$$b = \frac{I(X,Y) + G(X,Y)}{2}$$

Adding two Images

$$a = \frac{I(X,Y)}{2} + \frac{G(X,Y)}{2}$$

$$b = \frac{I(X,Y) + G(X,Y)}{2}$$

Question:

$a = b$

$a > b$

$a < b$

Adding two Images

Red pixel in camera man image = 98

Red pixel in moon image = 200

$$\frac{98}{2} + \frac{200}{2} = 49 + 100 = 149$$

$$\frac{98 + 200}{2} = \frac{\lfloor 298 \rfloor}{2} = \frac{255}{2} = 127$$

Question:

$a = b$

$a > b$

$a < b$

Adding two Images

It is often convenient to convert images to **doubles** when doing processing

In Python

```
from PIL import Image
img = Image.open('cameraman.png')
import numpy as np
imgArr = np.asfarray(img)

# Or do this
import matplotlib.pyplot as plt
camera = plt.imread('cameraman.png');
```

Adding two Images

This will save you a **LOT** of headache in homeworks:

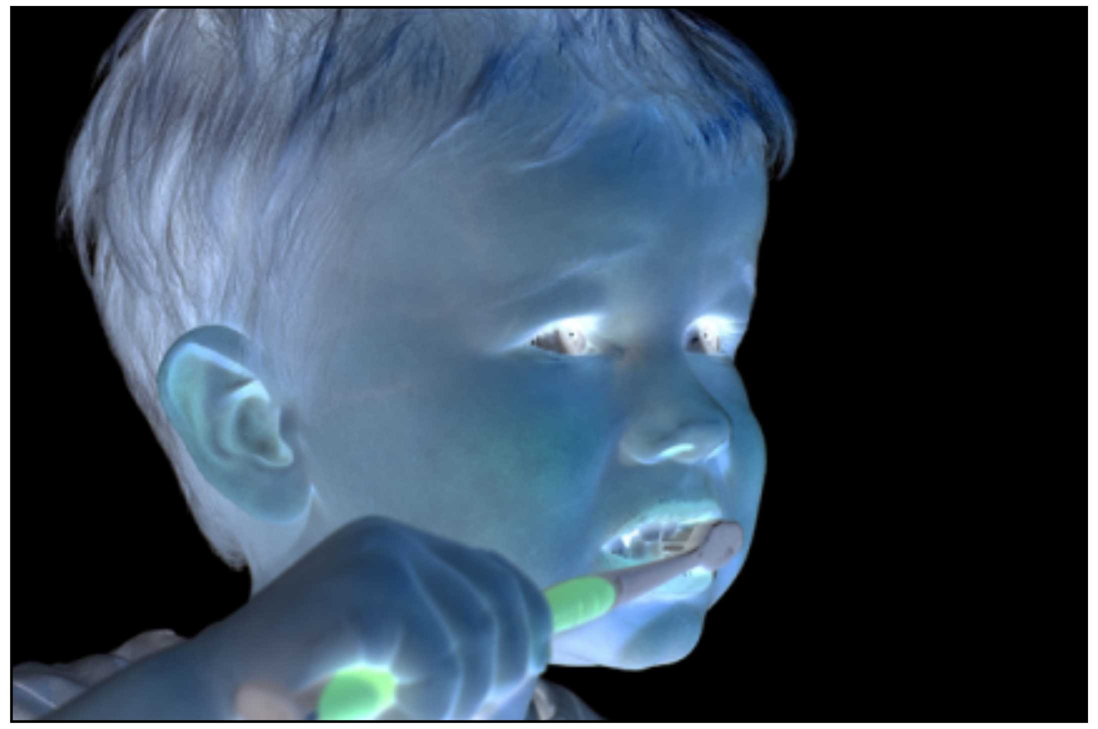
1. Convert to **doubles**
2. (optionally) Normalize image to [0,1] range (by dividing by 255)
3. Perform any **computations** needed
4. (optionally) Undo normalization (by multiplying by 255)
5. **Clamp** values between [0, 255]
6. Convert to **uint8**

What types of **transformations** can we do?

$I(X,Y)$

Filtering

$I'(X,Y)$

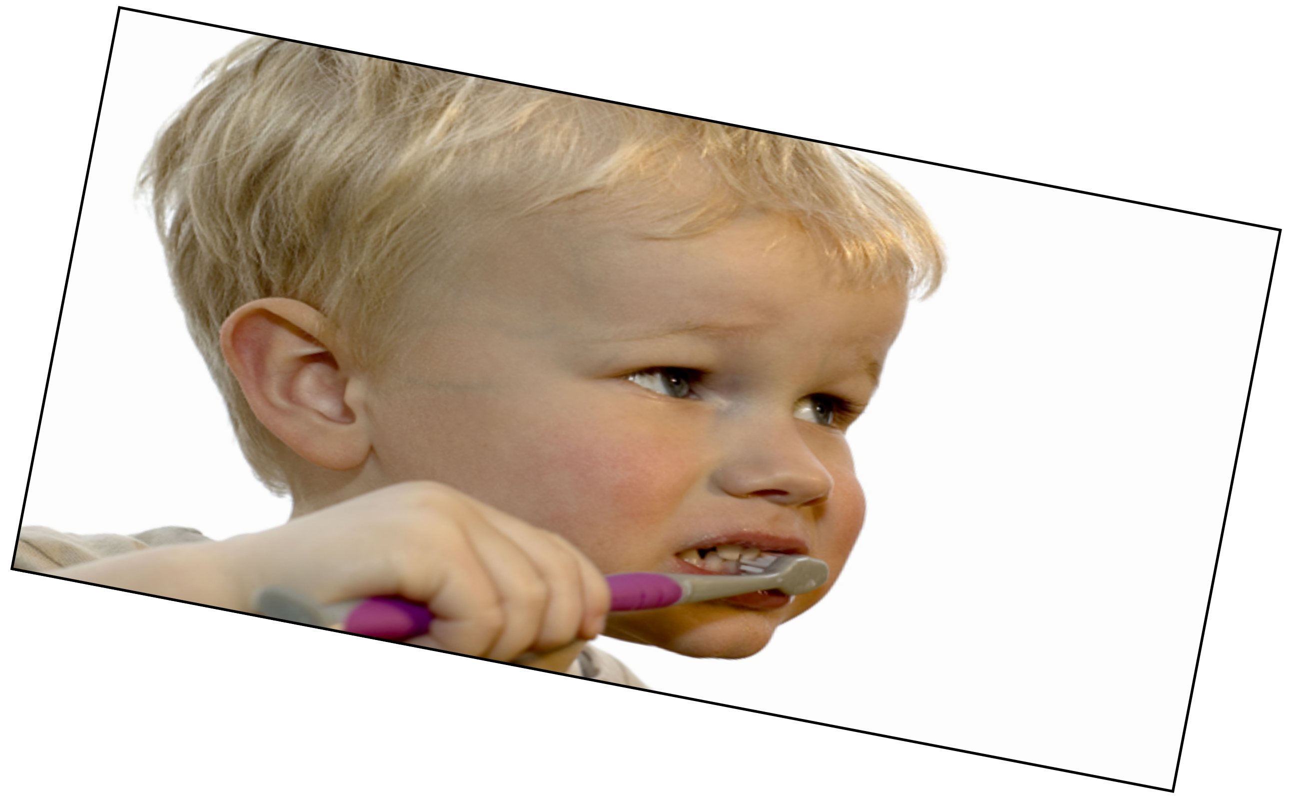


changes range of image function

$I(X,Y)$

Warping

$I'(X,Y)$

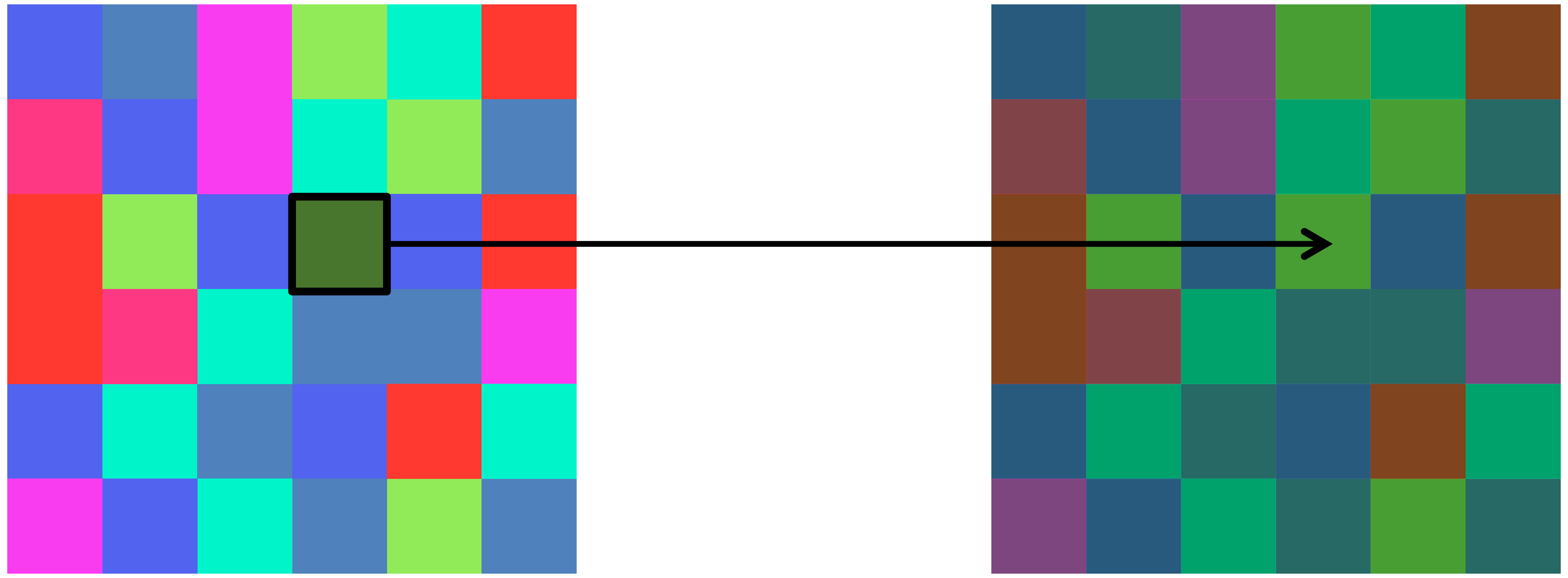


changes domain of image function

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

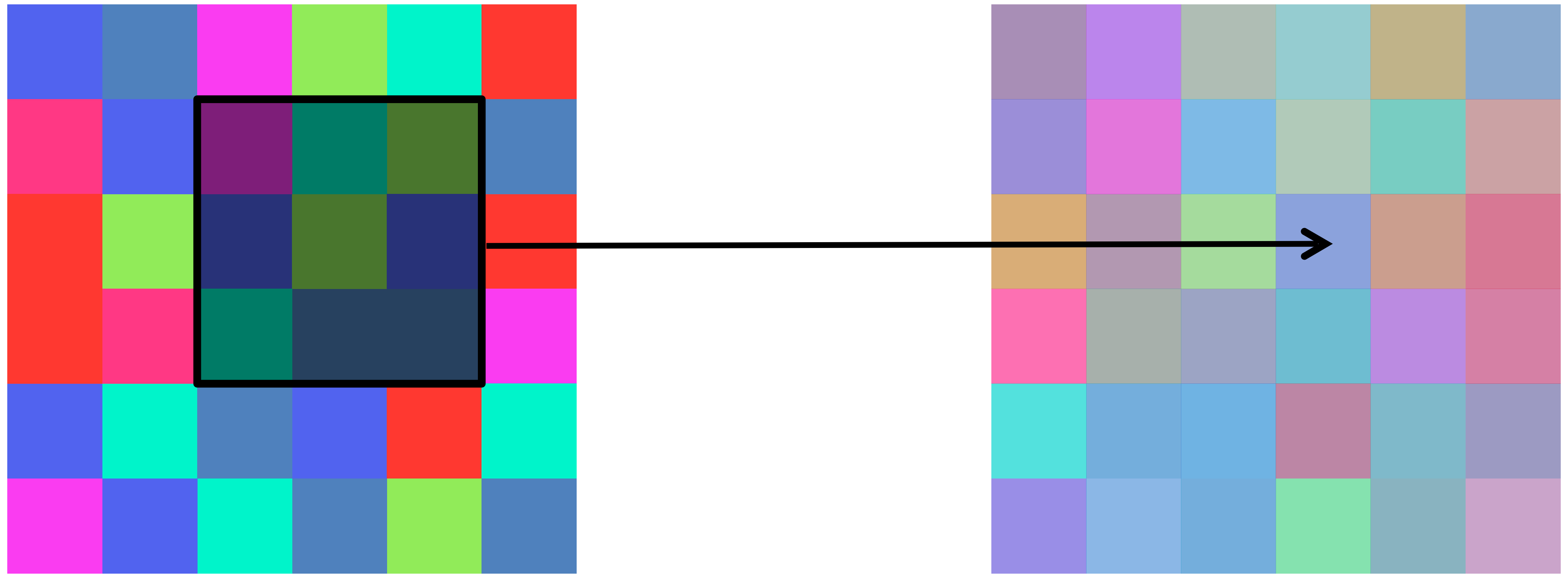
What types of **filtering** can we do?

Point Operation



point processing

Neighborhood Operation



“filtering”

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$I(X, Y)$

darken

lower contrast

non-linear lower contrast

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

darken

lower contrast

non-linear lower contrast

$I(X, Y)$

$I(X, Y) - 128$

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$I(X,Y)$

darken

$I(X,Y) - 128$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

invert

lighten

raise contrast

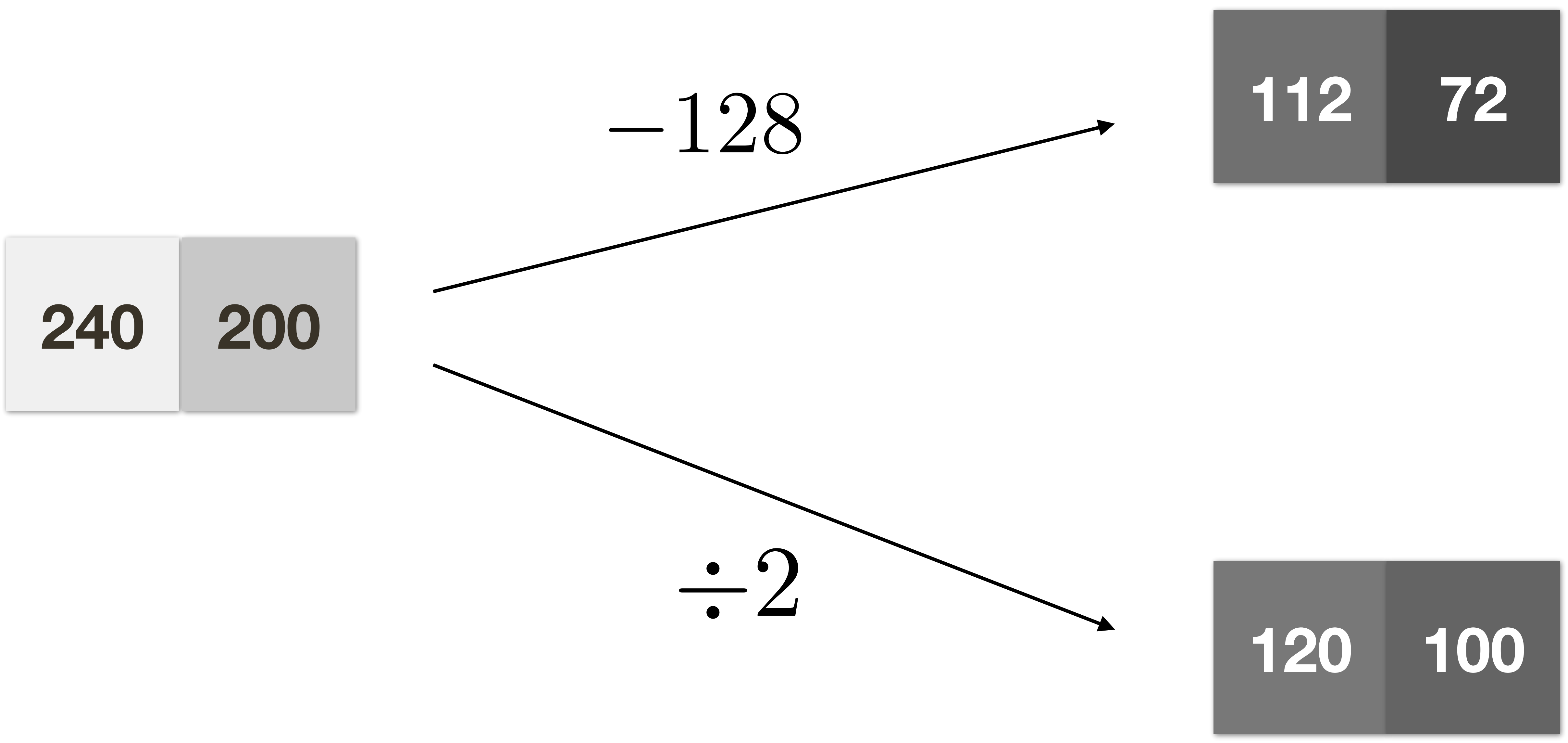
non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Brightness v.s. **Contrast**

Brightness: all pixels get lighter/darker, relative difference between pixel values stays the same

Contrast: relative difference between pixel values becomes higher / lower



Examples of **Point Processing**

original

$I(X,Y)$

darken

$I(X,Y) - 128$

lower contrast

$\frac{I(X,Y)}{2}$

non-linear lower contrast

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

$$I(X,Y) \times 2$$

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

$$I(X,Y) \times 2$$

non-linear raise contrast

$$\left(\frac{I(X,Y)}{255}\right)^{2} \times 255$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$I(X,Y)$

darken

$I(X,Y) - 128$

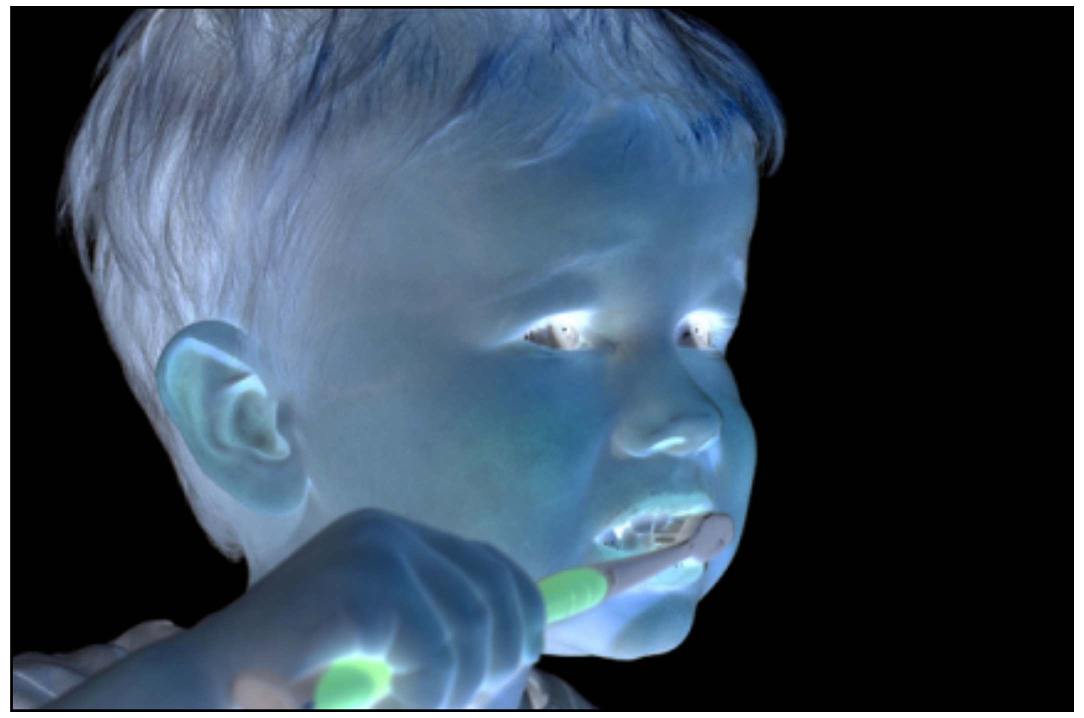
lower contrast

$\frac{I(X,Y)}{2}$

non-linear lower contrast

$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$

invert



$255 - I(X,Y)$

lighten

$I(X,Y) + 128$

raise contrast

$I(X,Y) \times 2$

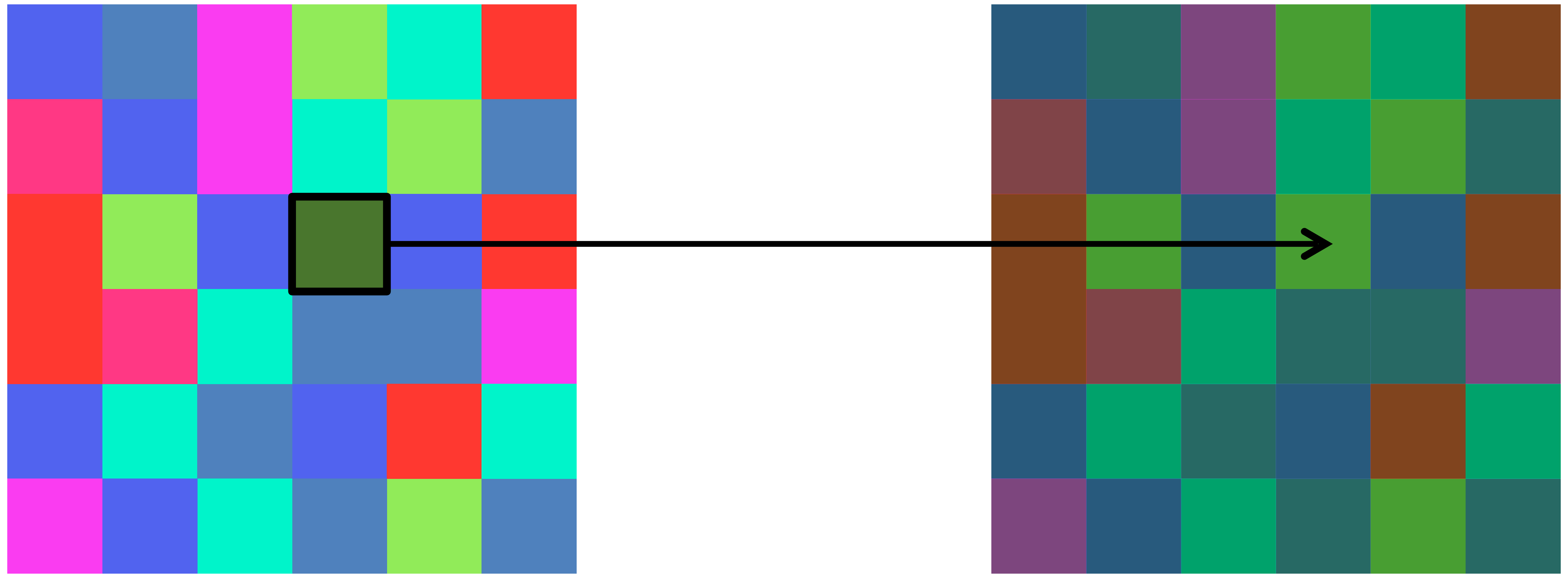
non-linear raise contrast

$\left(\frac{I(X,Y)}{255}\right)^{2} \times 255$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

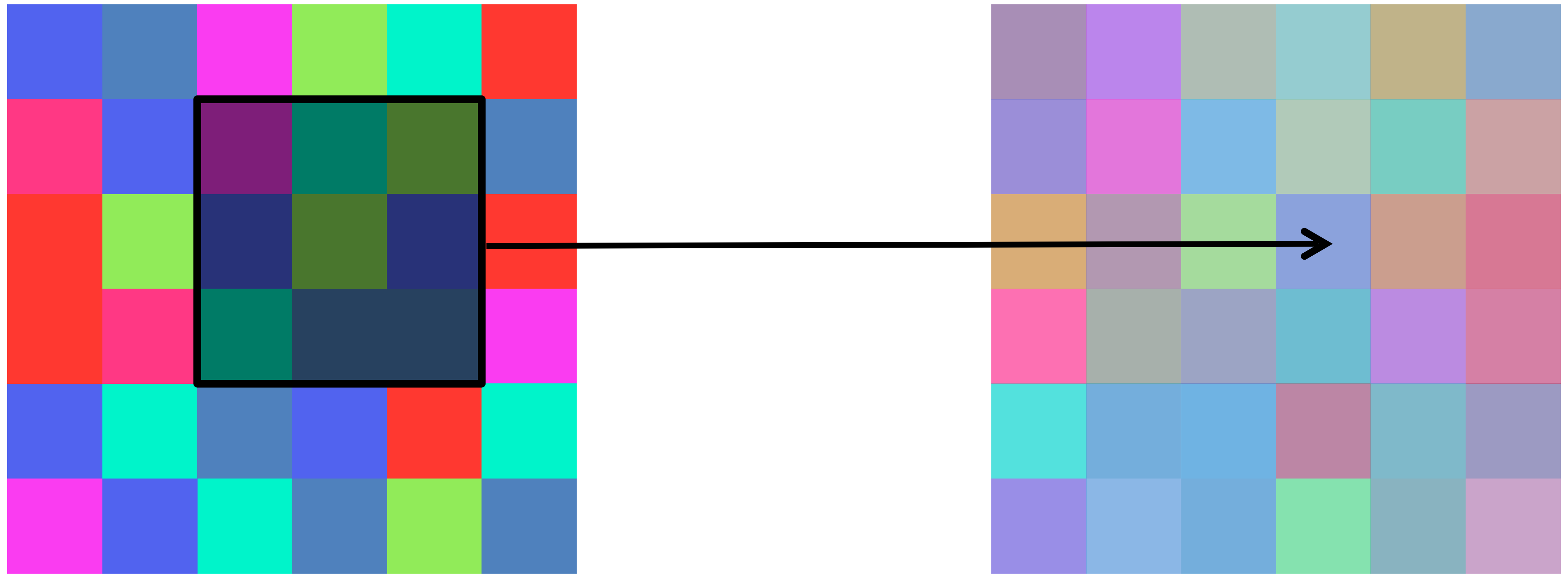
What types of **filtering** can we do?

Point Operation



point processing

Neighborhood Operation



"filtering"

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Neighborhood Operators (Filtering)

Original Image

blur

sharpen

edge filter

Non-Linear Neighborhood Operators (Filtering)

Original Image

edge preserving smoothing

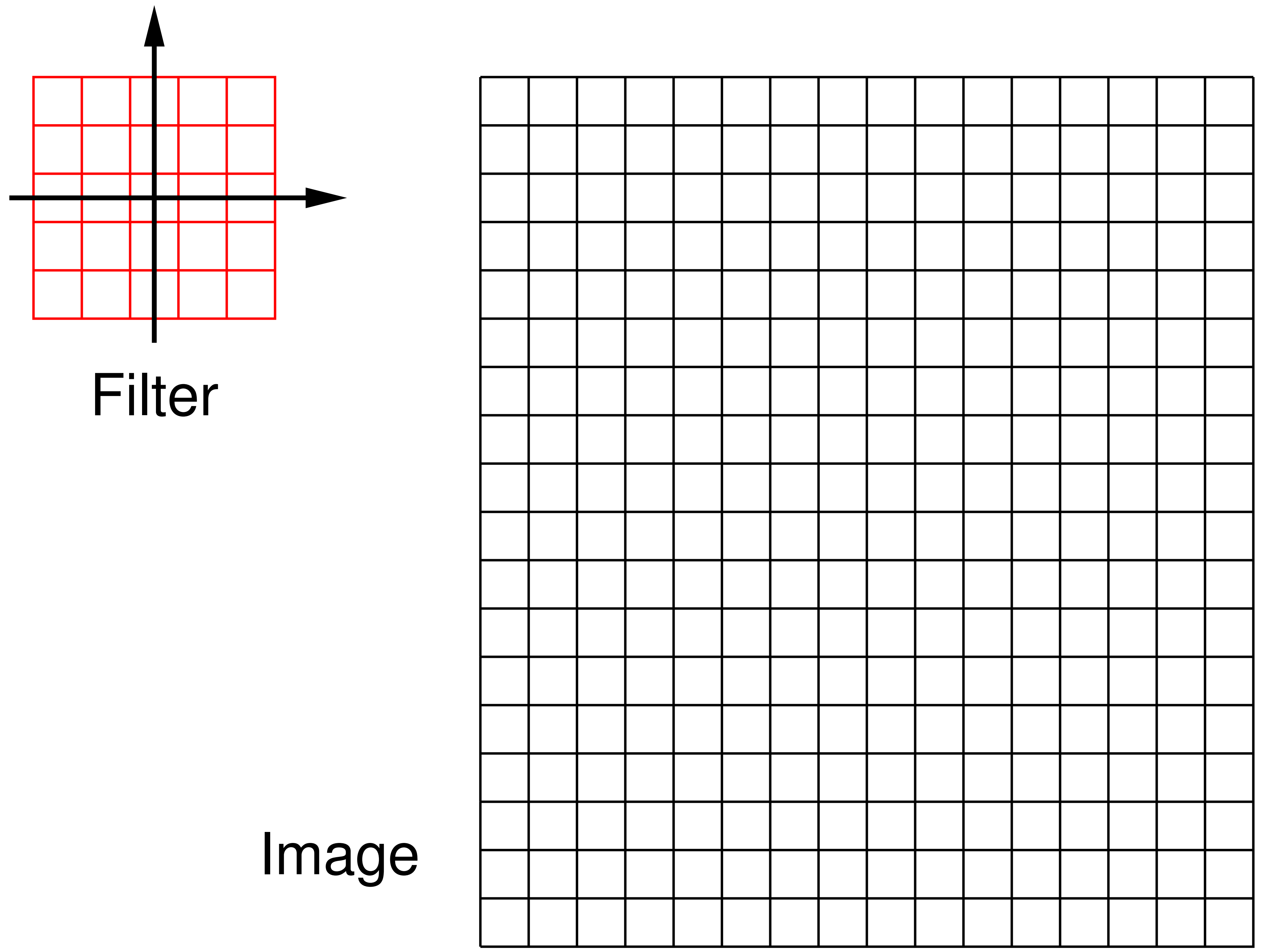
median

cenny edges

Linear **Filters**

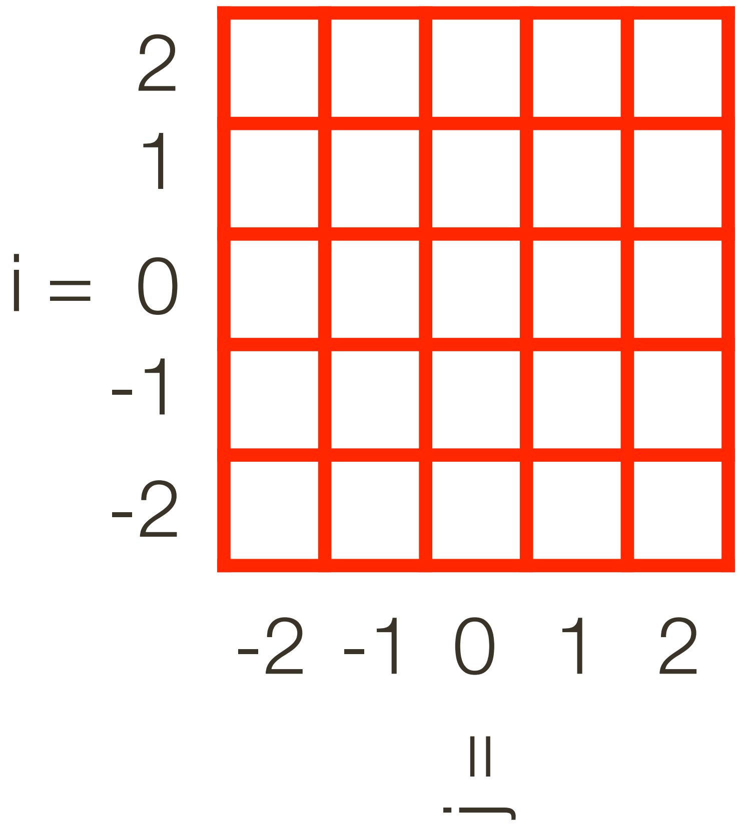
Let $I(X, Y)$ be an $n \times n$ digital image (for convenience we let width = height)

Let $F(X, Y)$ be another $m \times m$ digital image (our "**filter**" or "**kernel**")



For convenience we will assume m is odd. (Here, $m = 5$)

Linear **Filters**



Let $k = \left\lfloor \frac{m}{2} \right\rfloor$

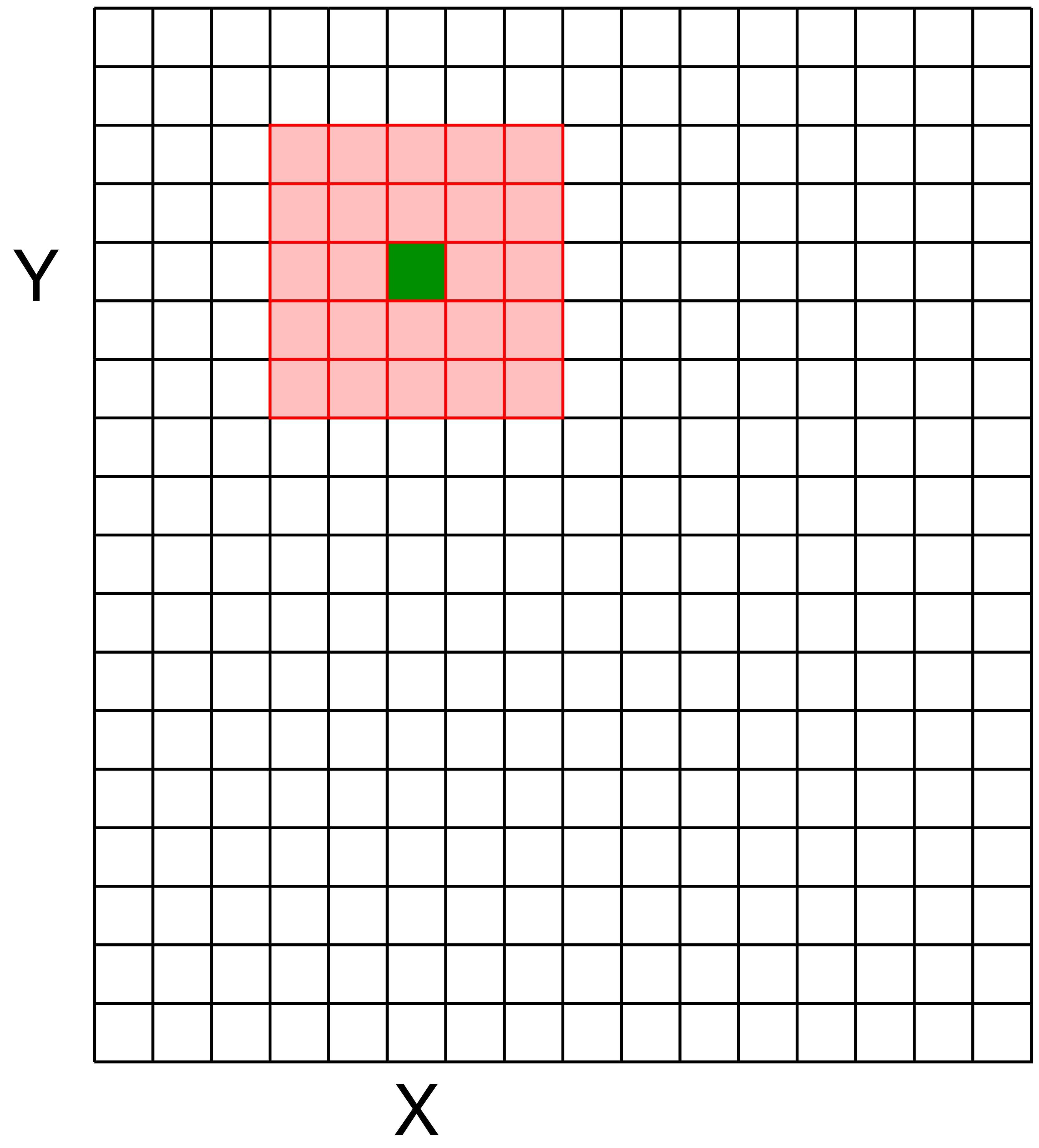
Compute a new image, $I'(X,Y)$, as follows

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Intuition: each pixel in the output image is a linear combination of the same index pixel and its neighboring pixels in the original image

Linear **Filters**

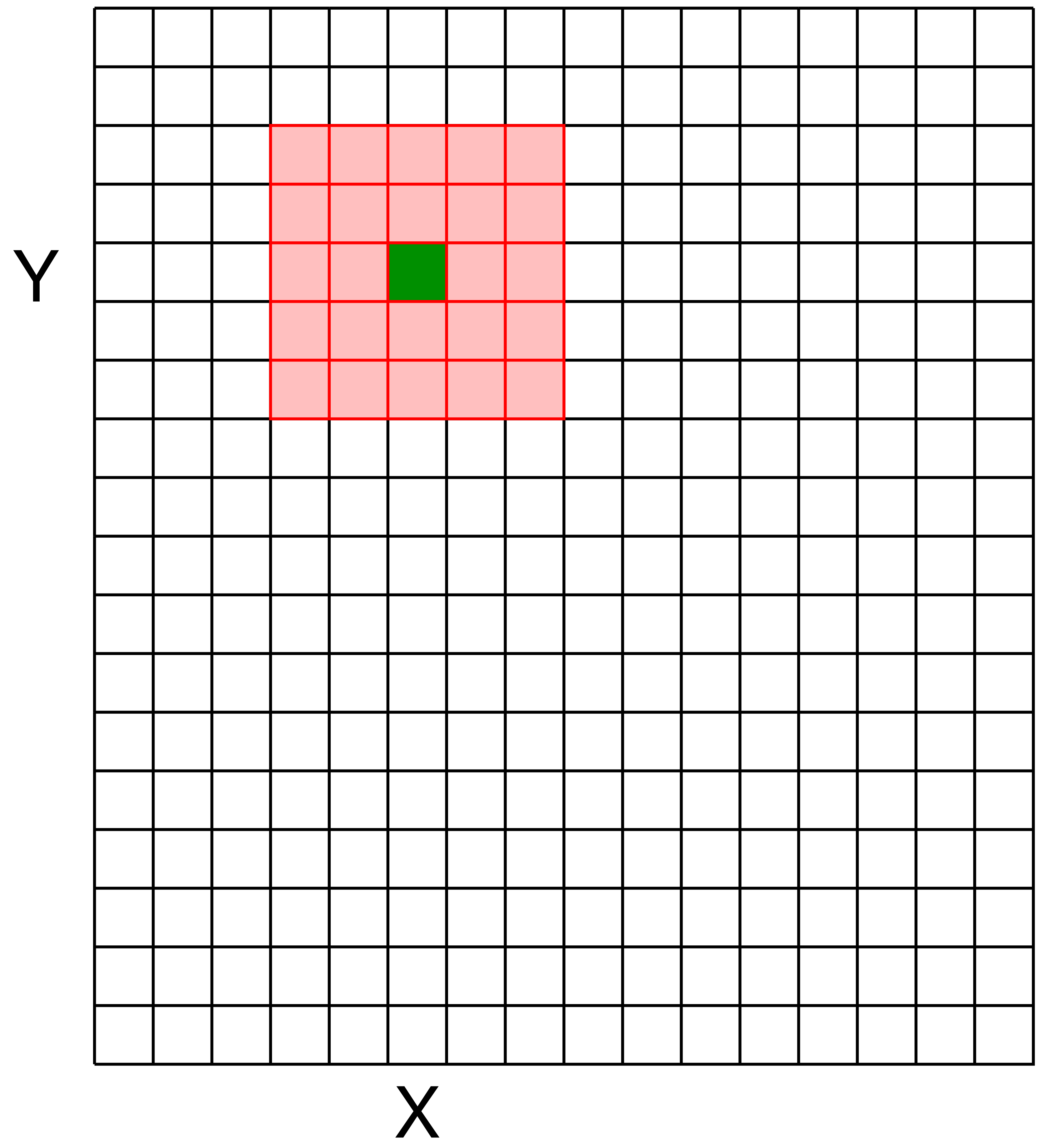
For a given X and Y, superimpose the filter on the image centered at (X, Y)



Linear **Filters**

For a given X and Y, superimpose the filter on the image centered at (X, Y)

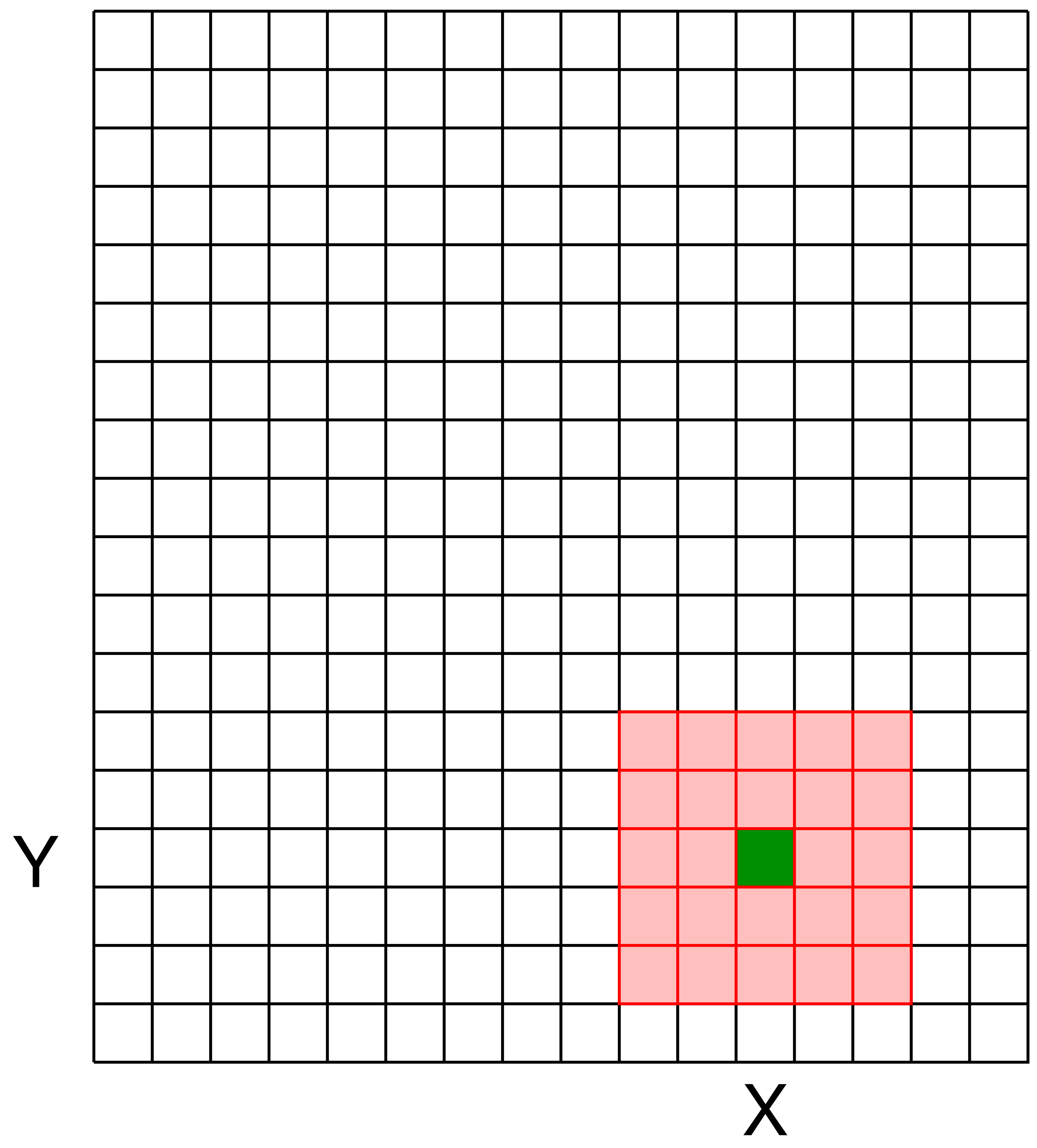
Compute the new pixel value, $I'(X, Y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(X, Y)$ and the corresponding values in the filter



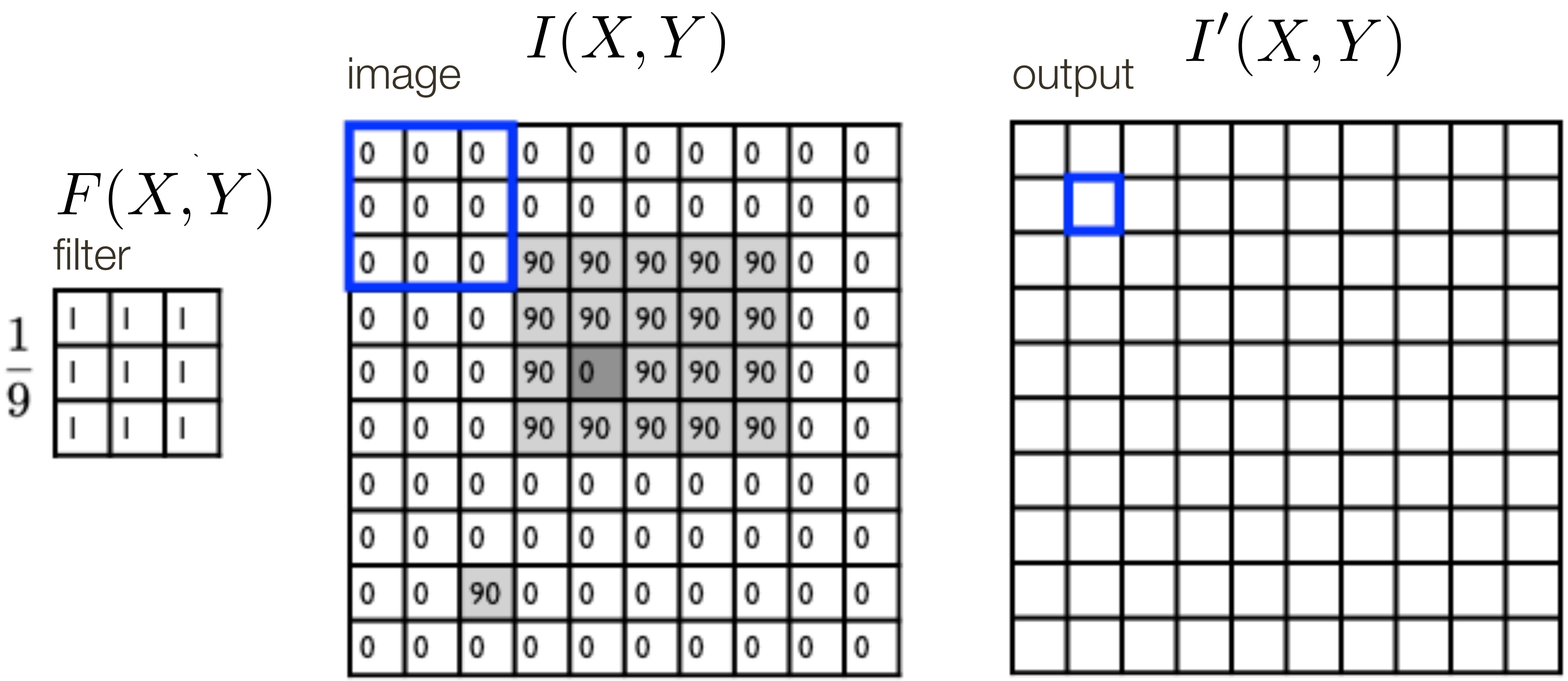
Linear **Filters**

The computation is repeated for each

$$(X, Y)$$



Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0								

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0								

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10							

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

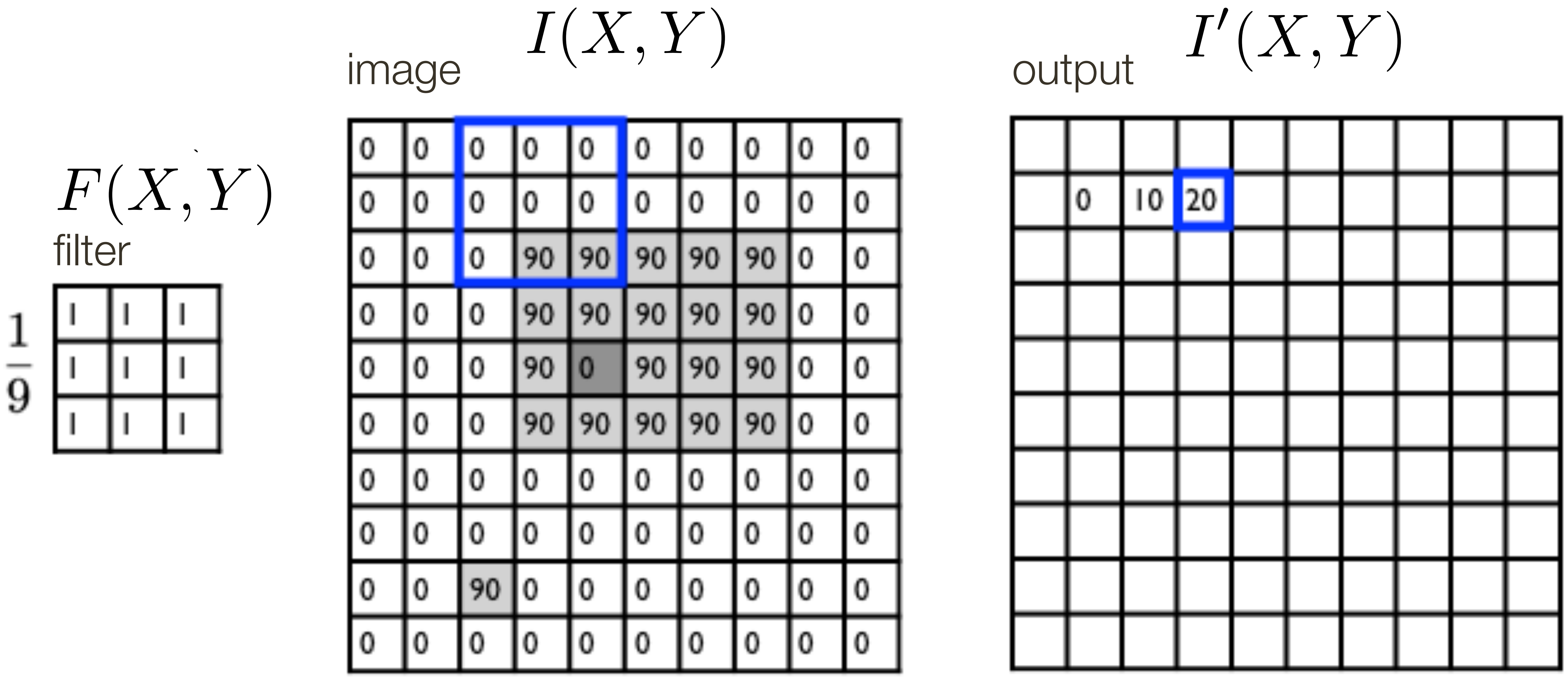
	0	10							

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20						

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

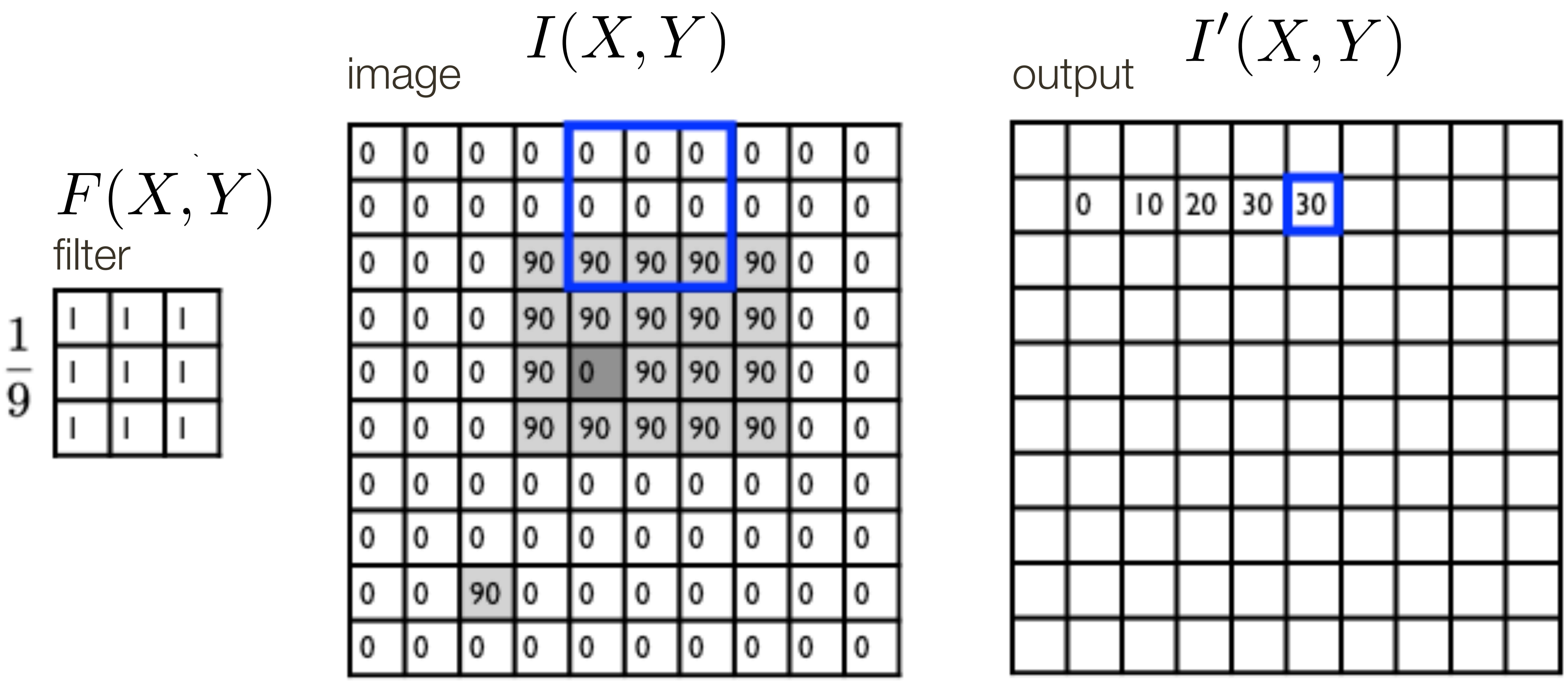
output $I'(X,Y)$

	0	10	20	30					

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30			

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20		

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) \, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

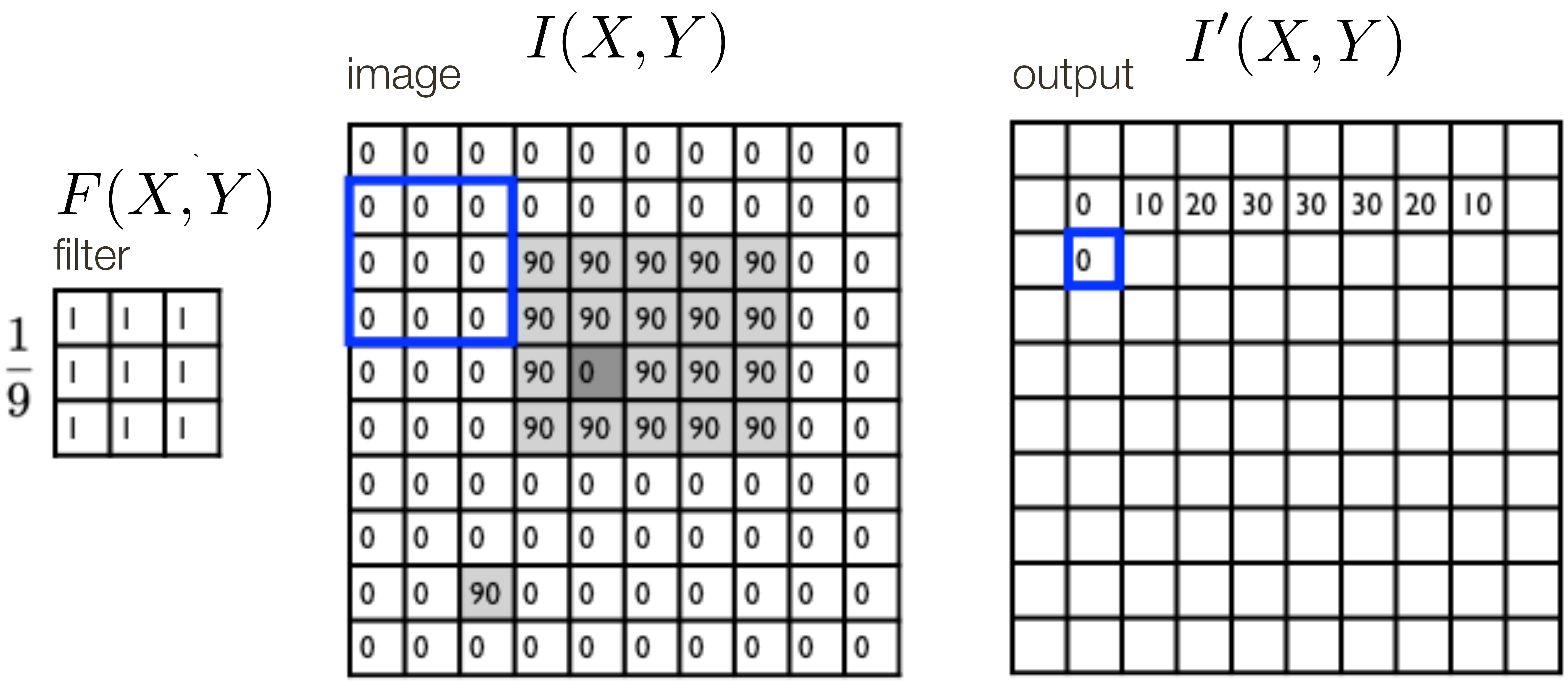
	0	10	20	30	30	30	20	10	

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20							

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\,\underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40						

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0								

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30							

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\, \underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

output

filter

image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10								

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10	10	10	10	0	0	0	0	

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10	10	10	10	0	0	0	0	

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear **Filters**

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

For a give X and Y, superimpose the filter on the image centered at (X, Y)

Compute the new pixel value, $I'(X, Y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(X, Y)$ and the corresponding values in the filter

Linear **Filters**

Let's do some accounting …

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — $I'(X,Y)$

filter — $F(i,j)$

image (signal) — $I(X+i, Y+j)$

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X,Y), there are $m \times m$ multiplications

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X, Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X, Y)

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X,Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X,Y)

Total: $m^2 \times n^2$ multiplications

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

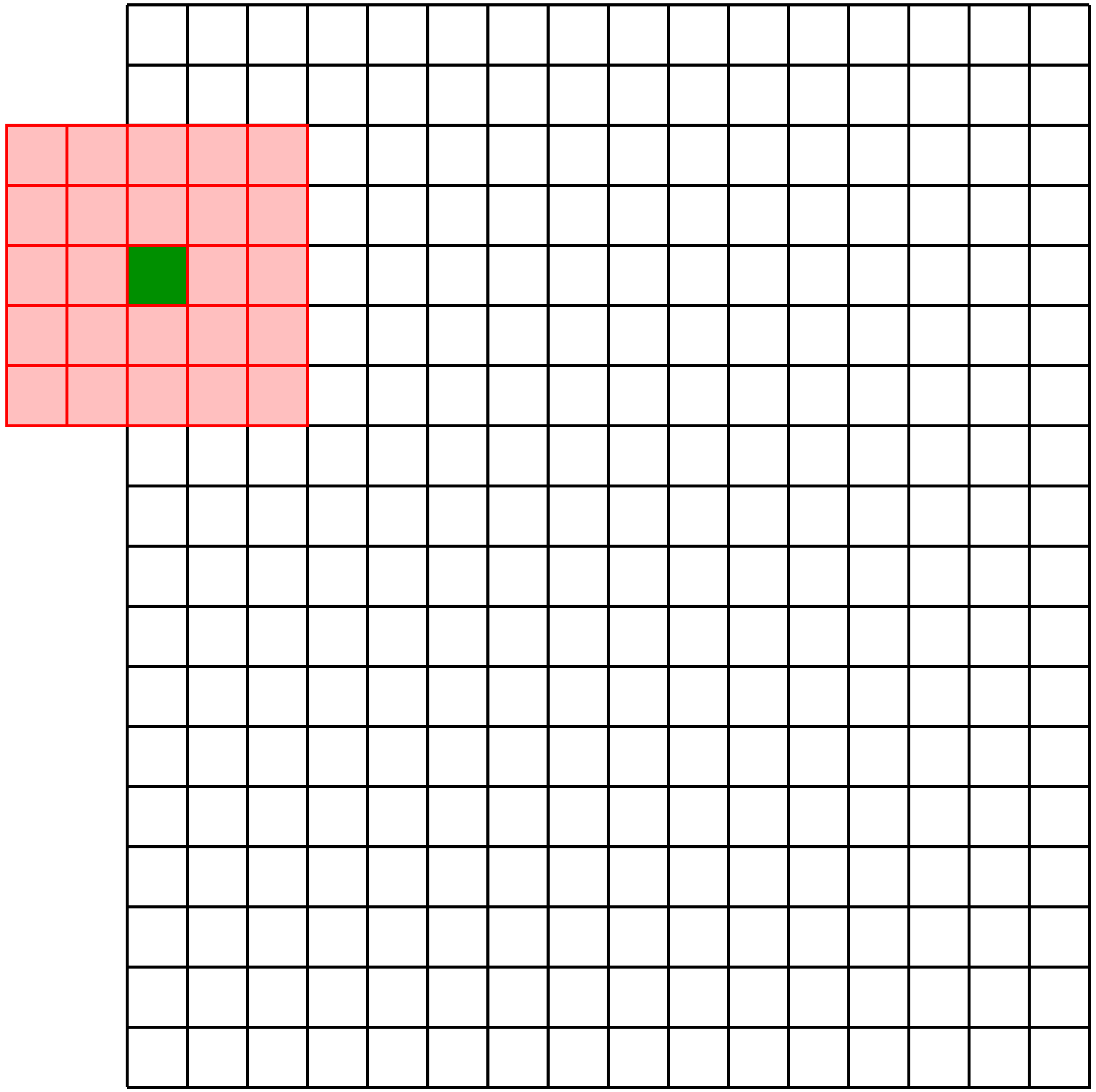
At each pixel, (X,Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X,Y)

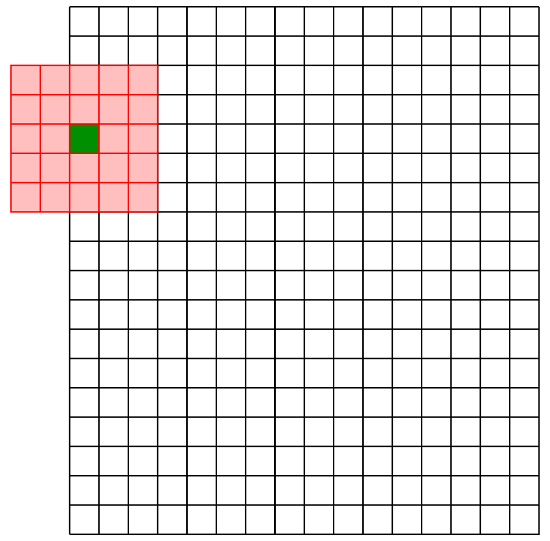
Total: $m^2 \times n^2$ multiplications

When m is fixed, small constant, this is $\mathcal{O}(n^2)$. But when $m \approx n$ this is $\mathcal{O}(m^4)$.

Linear Filters: **Boundary** Effects



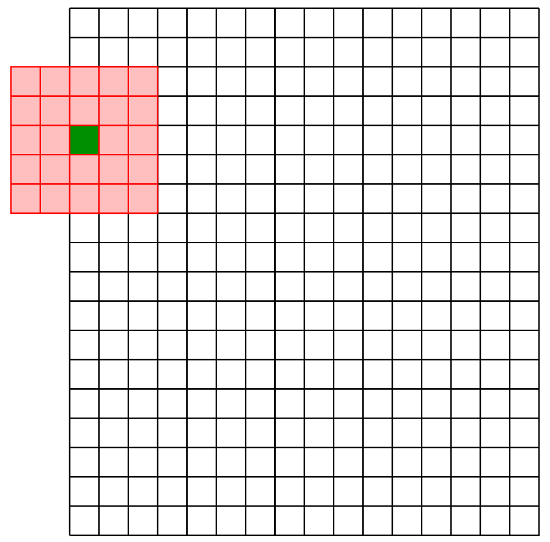
Linear Filters: **Boundary** Effects



Four standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

Linear Filters: **Boundary** Effects

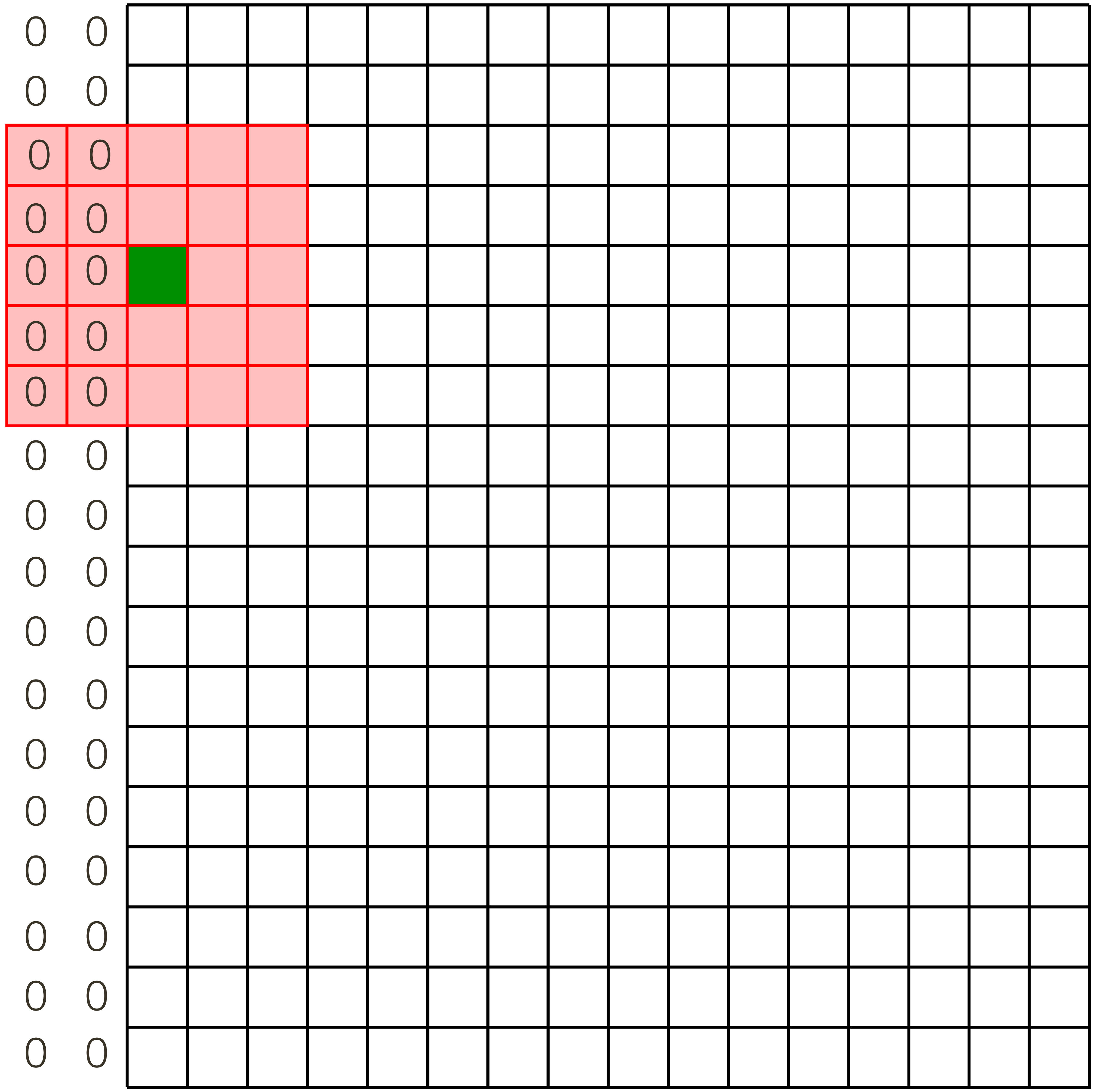


Four standard ways to deal with boundaries:

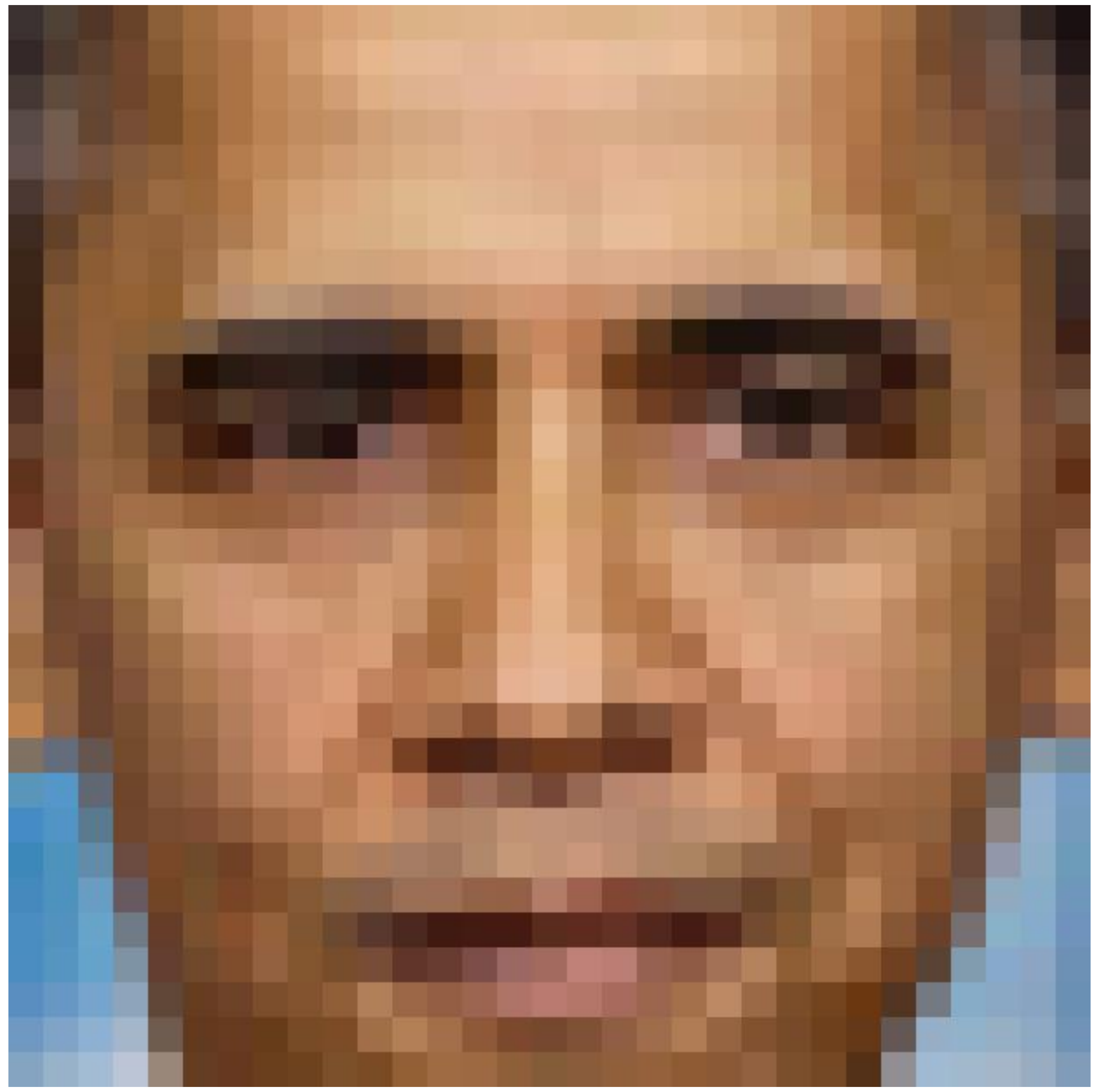
1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

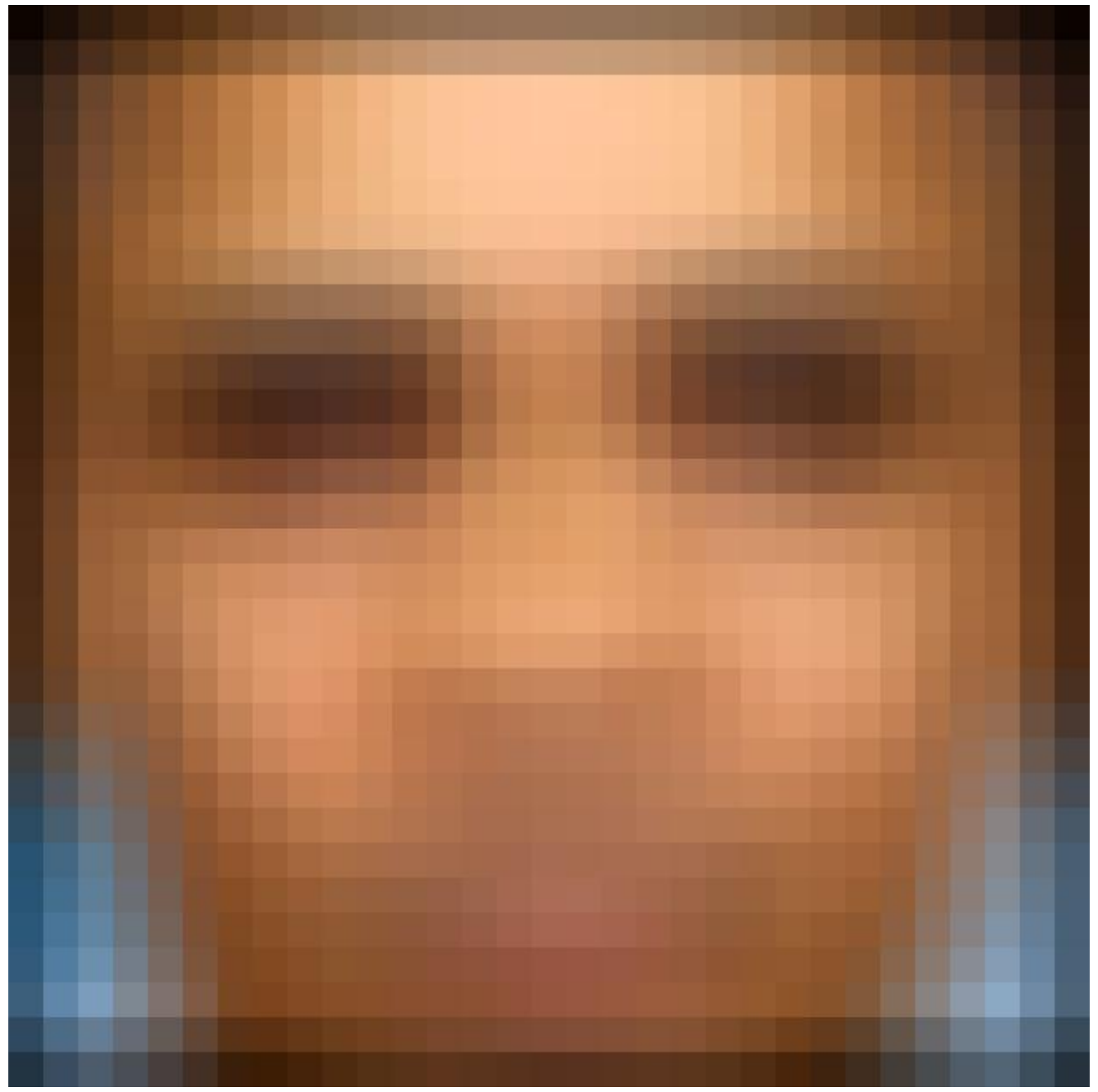
Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects

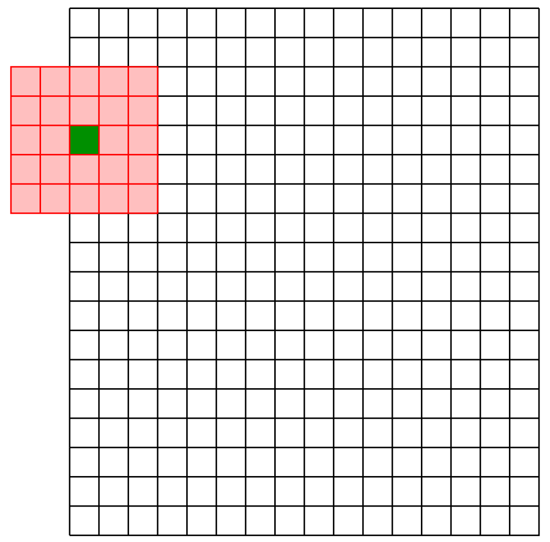


$$* \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} =$$



Notice **decrease** in brightness at edges

Linear Filters: **Boundary** Effects



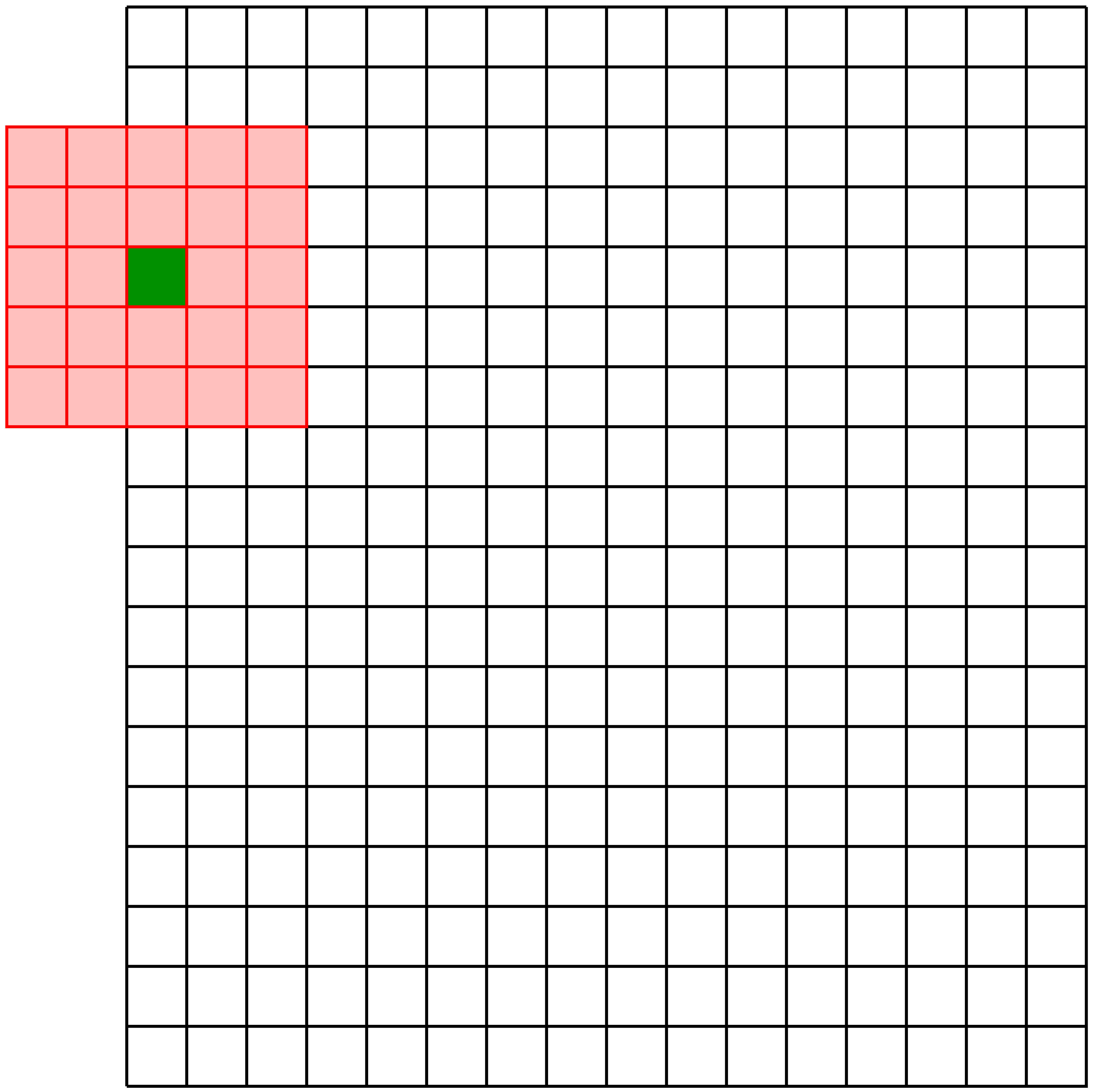
Four standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

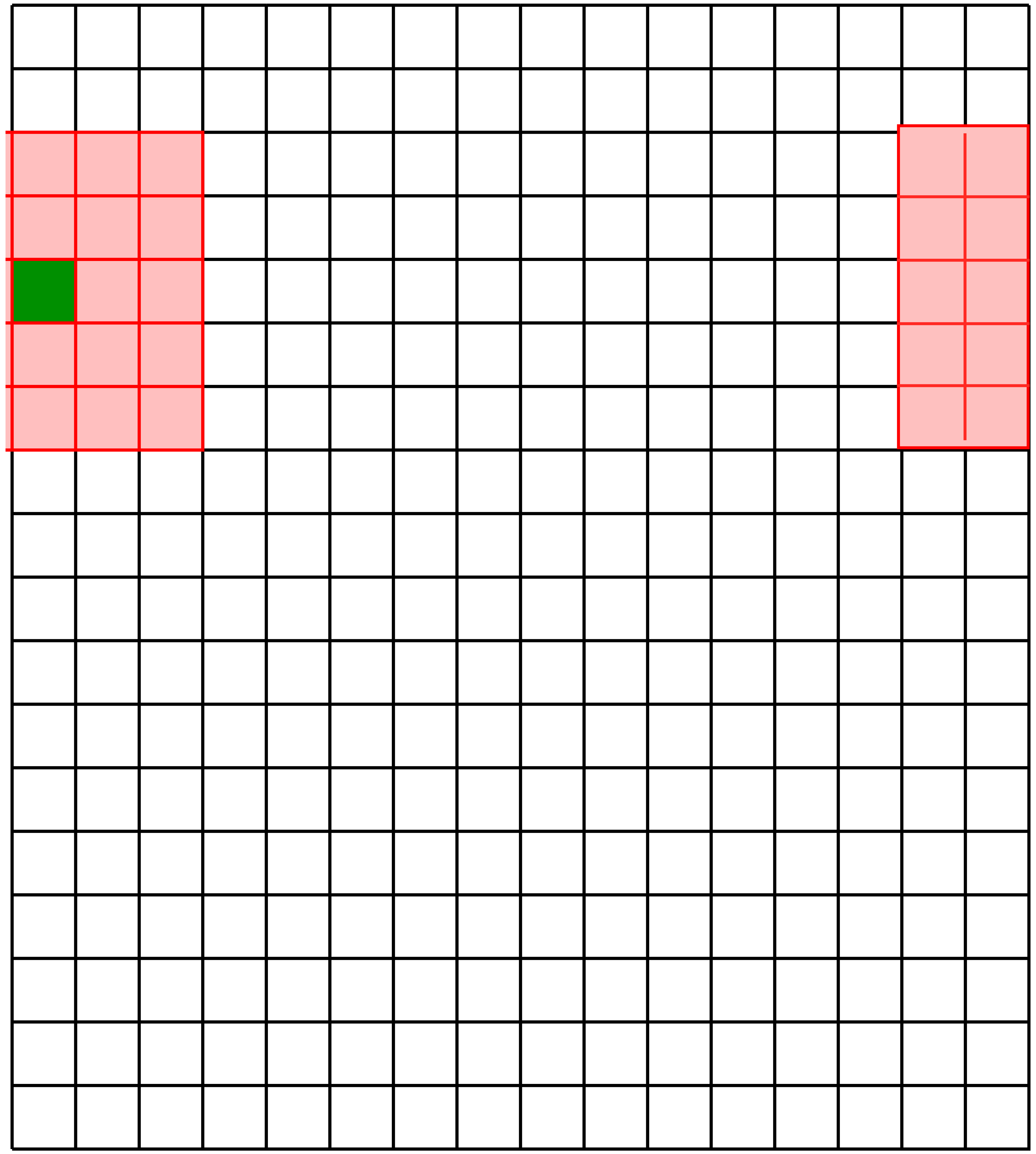
2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

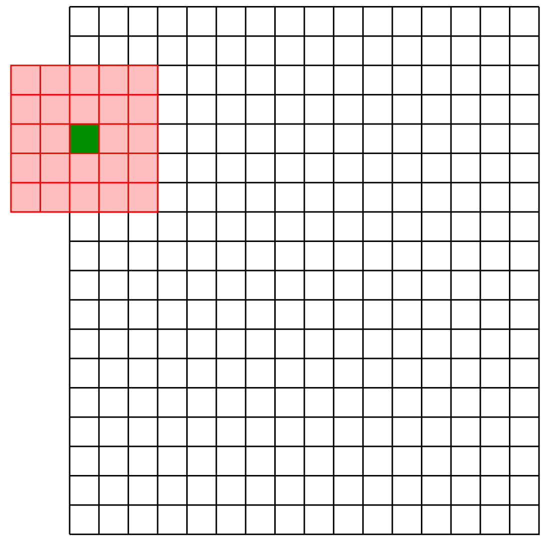
Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Four standard ways to deal with boundaries:

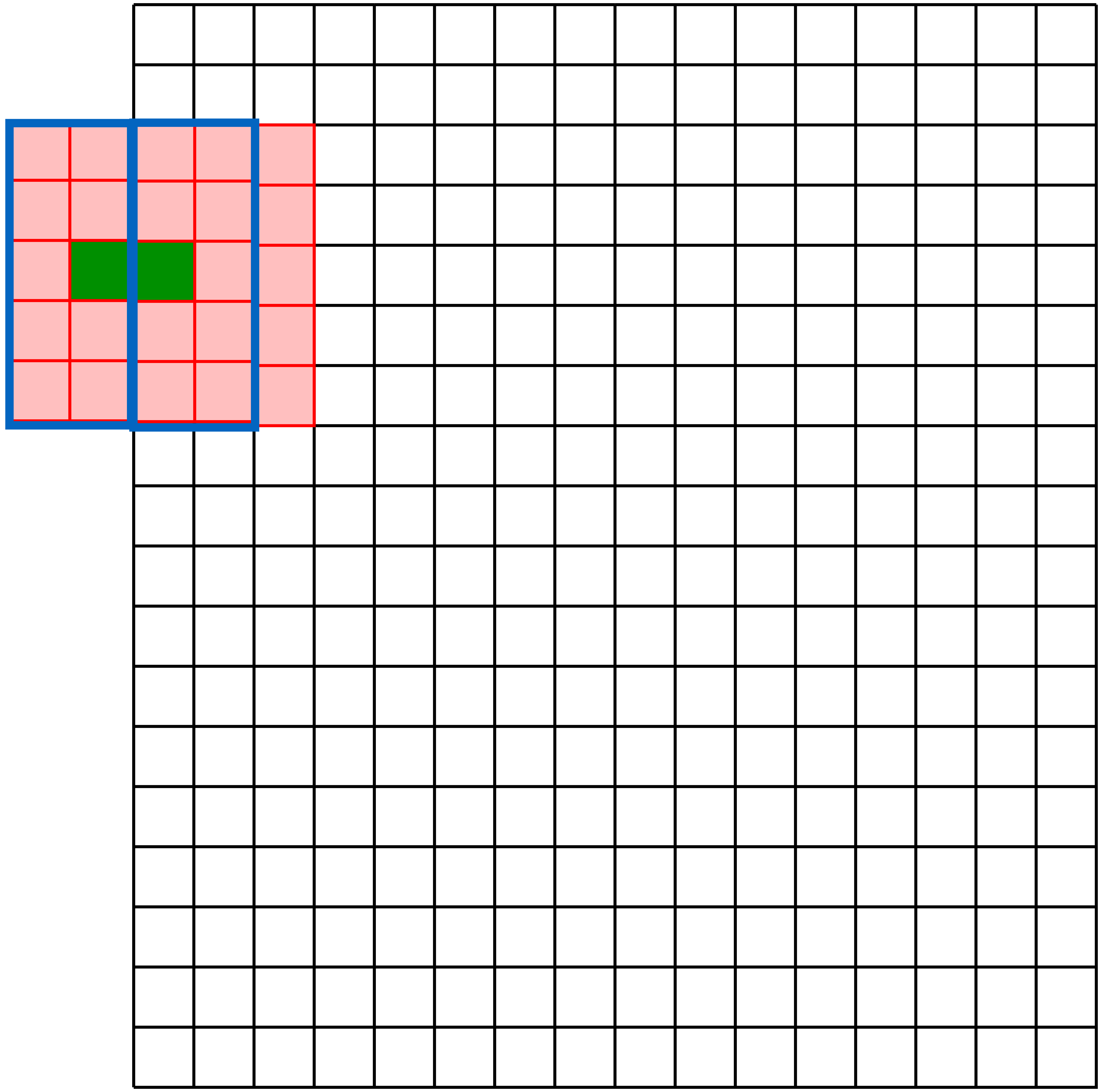
1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

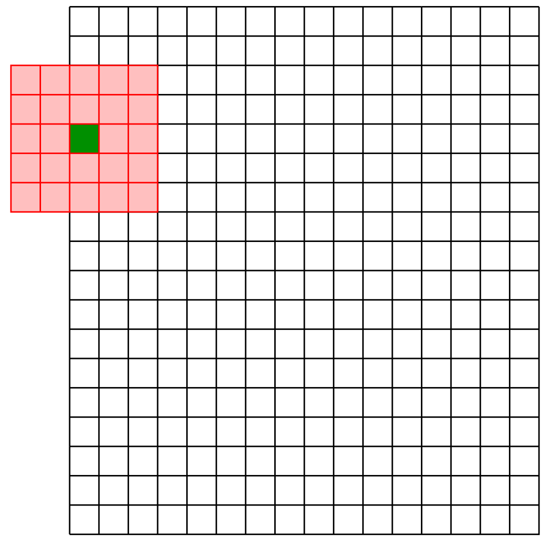
3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

4. **Reflect boarder**: Copy rows/columns locally by reflecting over the edge

Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Four standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

4. **Reflect boarder**: Copy rows/columns locally by reflecting over the edge