



UNIVERSITÀ DEGLI STUDI
DI MILANO

Corso di
Visione Artificiale

Laurea Magistrale in Informatica (F94)

Docente:
Raffaella Lanzarotti

*Dipartimento di Informatica
Università degli Studi di Milano*

Where are we?

First part:

Image formation and Early vision

- Image formation
 - **Geometric Camera Models**
 - Color spaces
- Image Processing
 - Punctual and spatial processing
 - Feature Extraction
- Reconstruction
 - Camera calibration
 - Stereo Vision
 - Structure from Motion and RGB-d Cameras
 - Optical flow and Tracking

Second part:

Machine learning for CV

- Linear Neural Network
- Multi Layer Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks
- Transformers
- Variational Auto-Encoders
- Generative Adversarial Networks
- Graph Neural Networks
- Self-supervised learning
- Vision Language Models

PINHOLE CAMERA

Ideal camera model

Chapter 1 – Forsyth Ponce

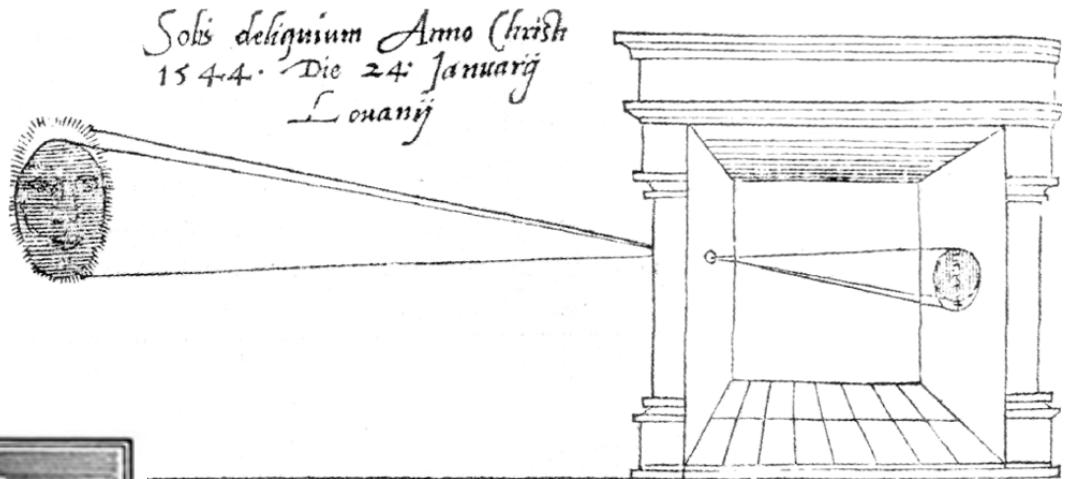
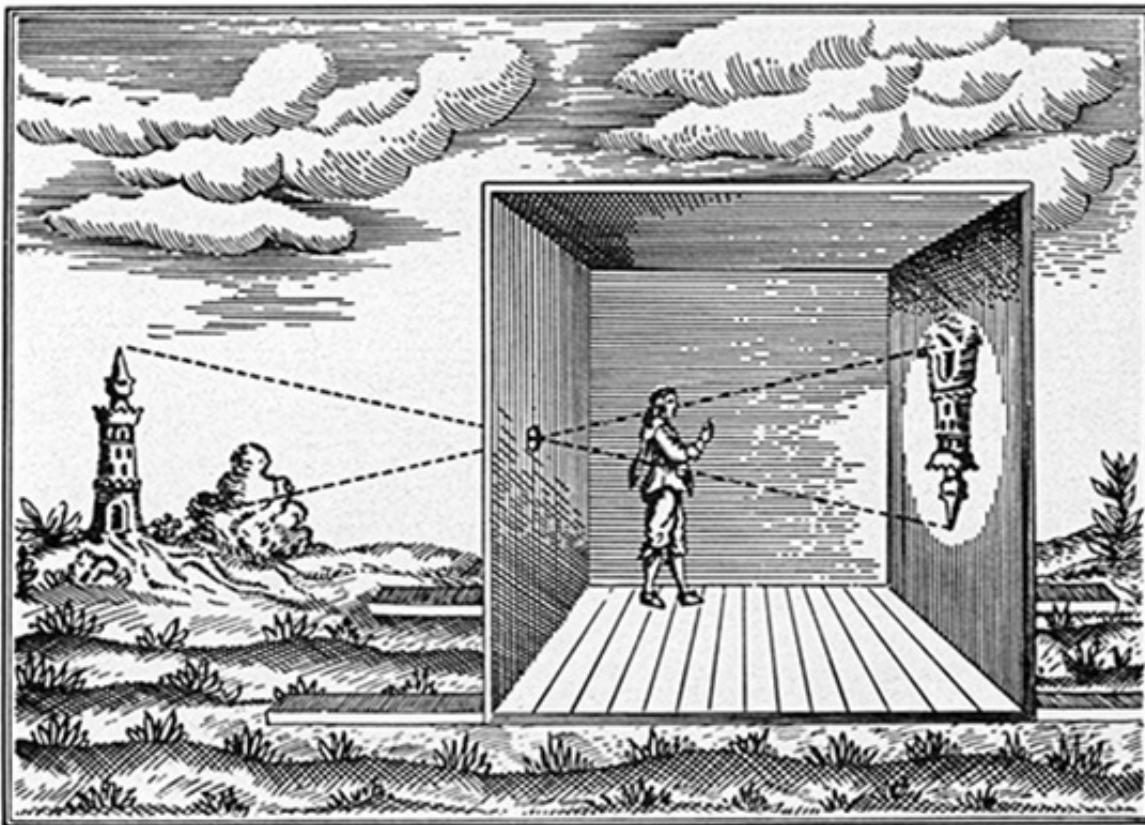
credits, F. Pedersini, S. Nayar

3

Pinhole camera model

“Camera obscura”

- Brunelleschi / Leonardo da Vinci, 1544
- generation system of a **perspective projection** of the front scene
(like the human eye)



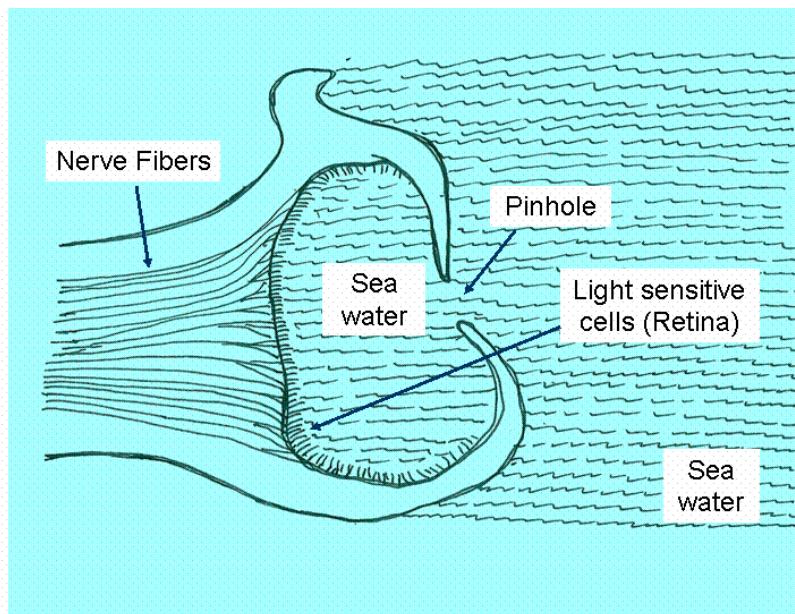
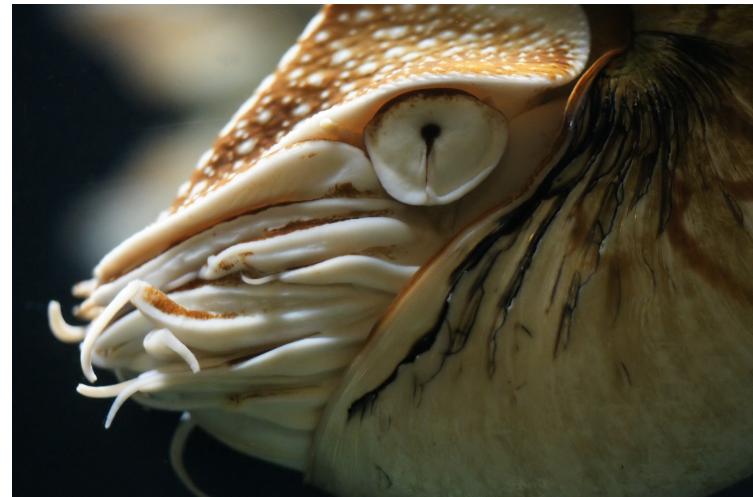
Perspective Projection:

Function that maps
a 3D space
onto a 2D plane
(image plane)

$$f: \mathbb{R}^3 \longrightarrow \mathbb{R}^2$$

Pinhole eye of *Nautilus pompilius*

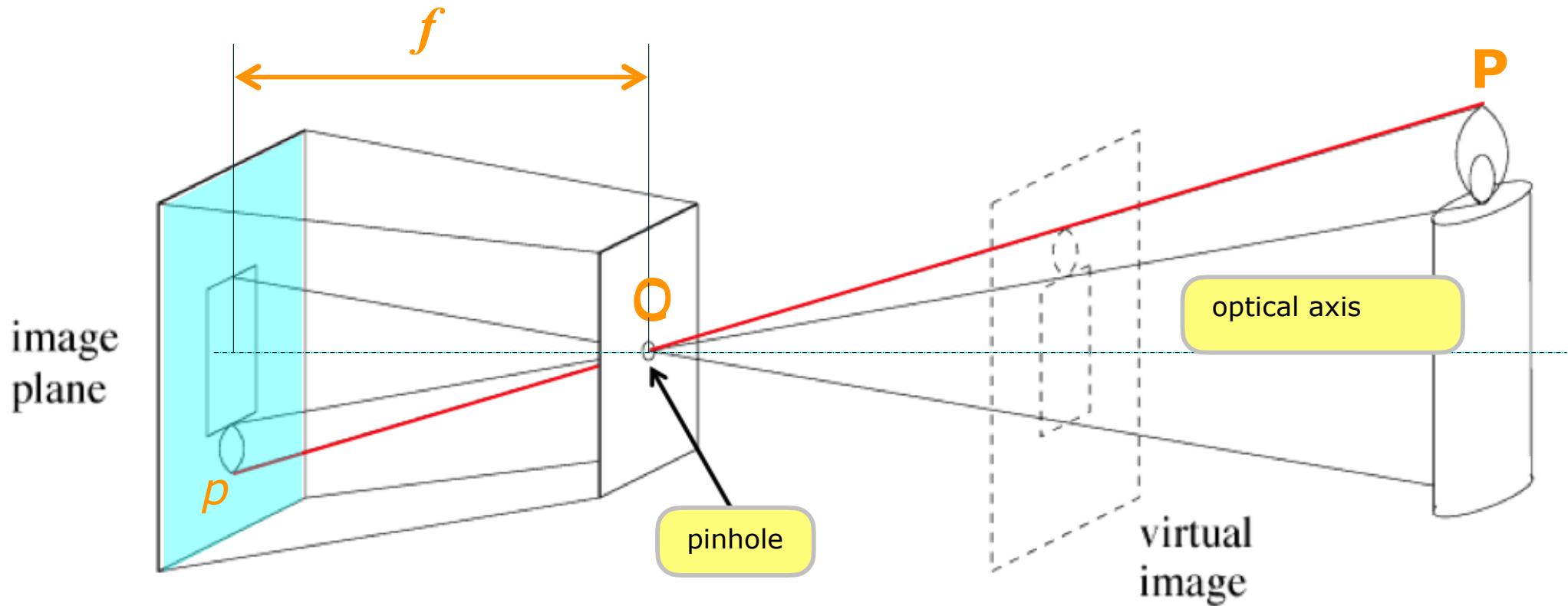
- *Nautilus pompilius'* eye does not have a lens:
- It uses a large pinhole to create an image



Perspective – the pinhole model

Pinhole camera model:

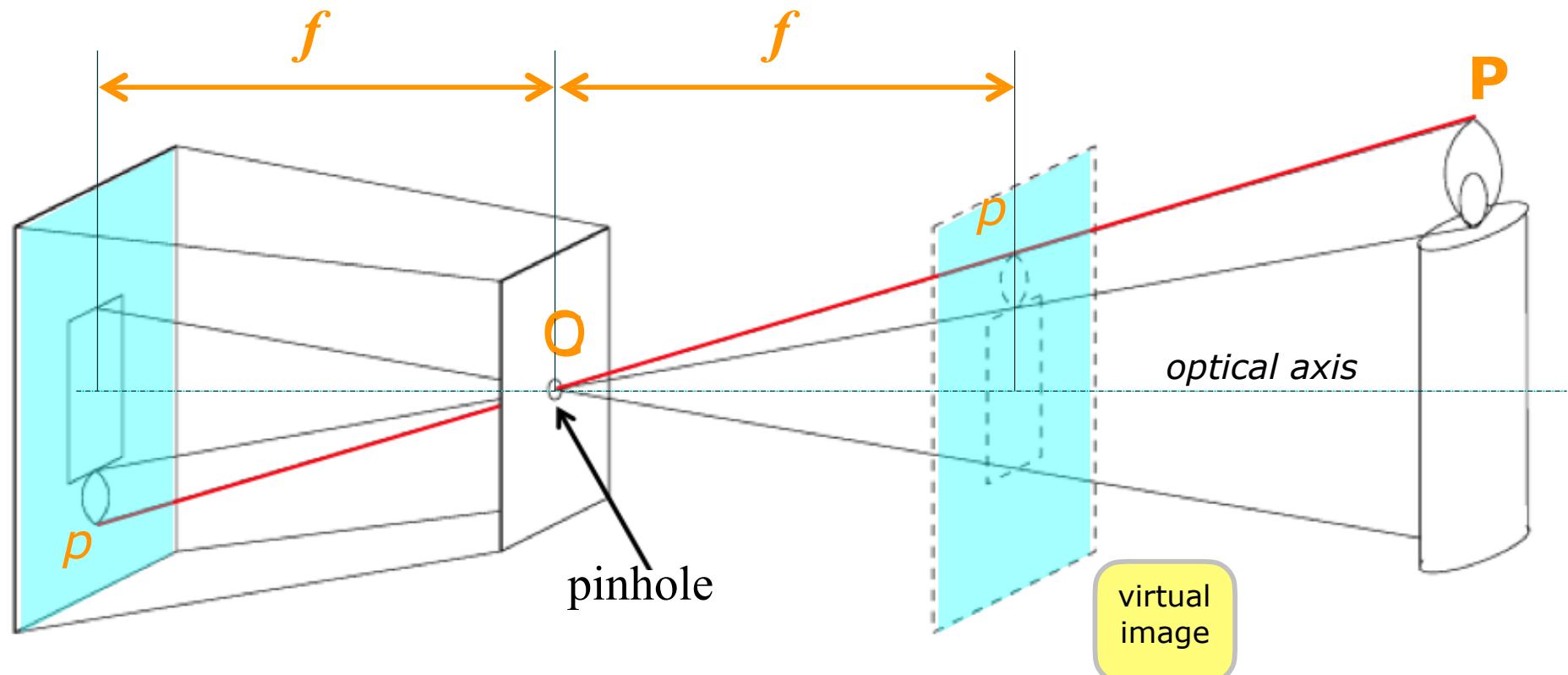
- from each point of the scene P , only the beam passing through P and hole O (ideally a point) reaches the image plane in p



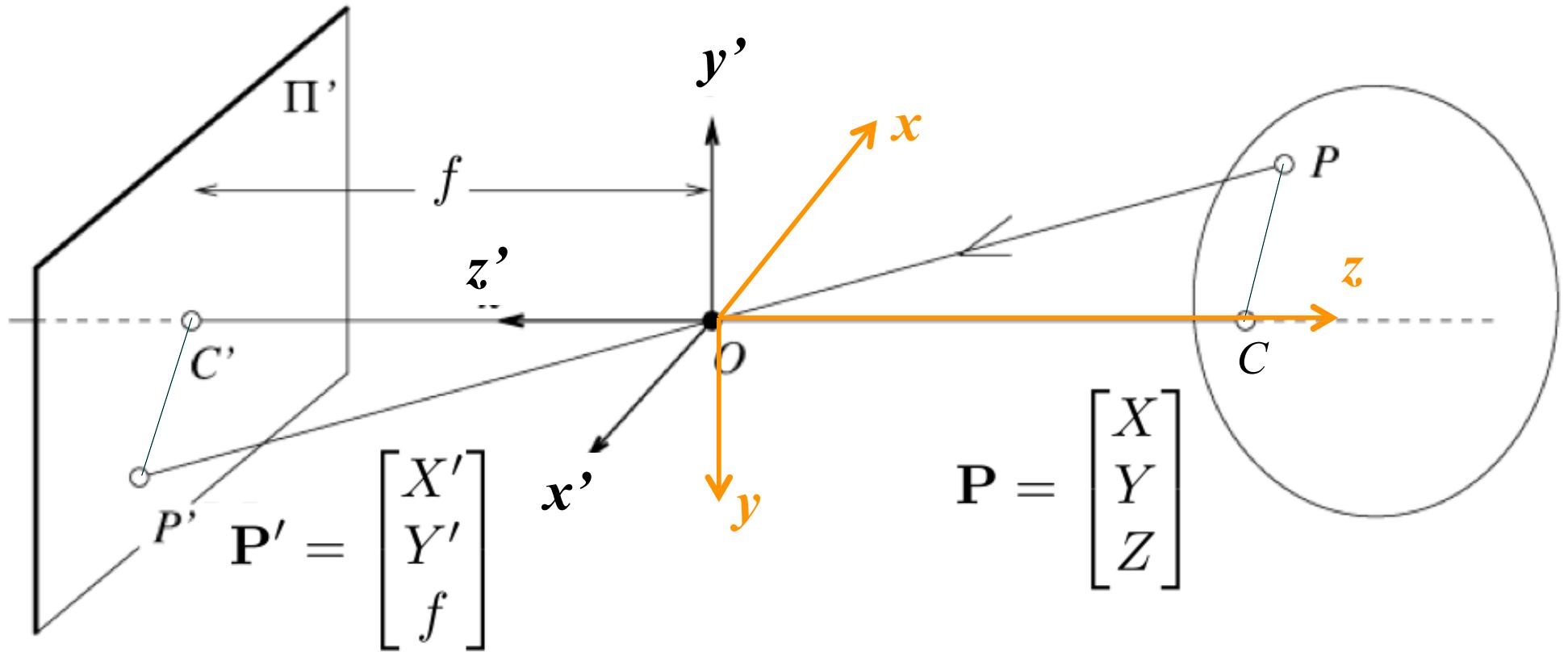
Perspective – the pinhole model

Pinhole: real and virtual image

- **real image** inverted, **virtual image** not inverted and same distance f wrt the pinhole O (geometrically equivalent)
- **f – focal length:** distance pinhole – image plane

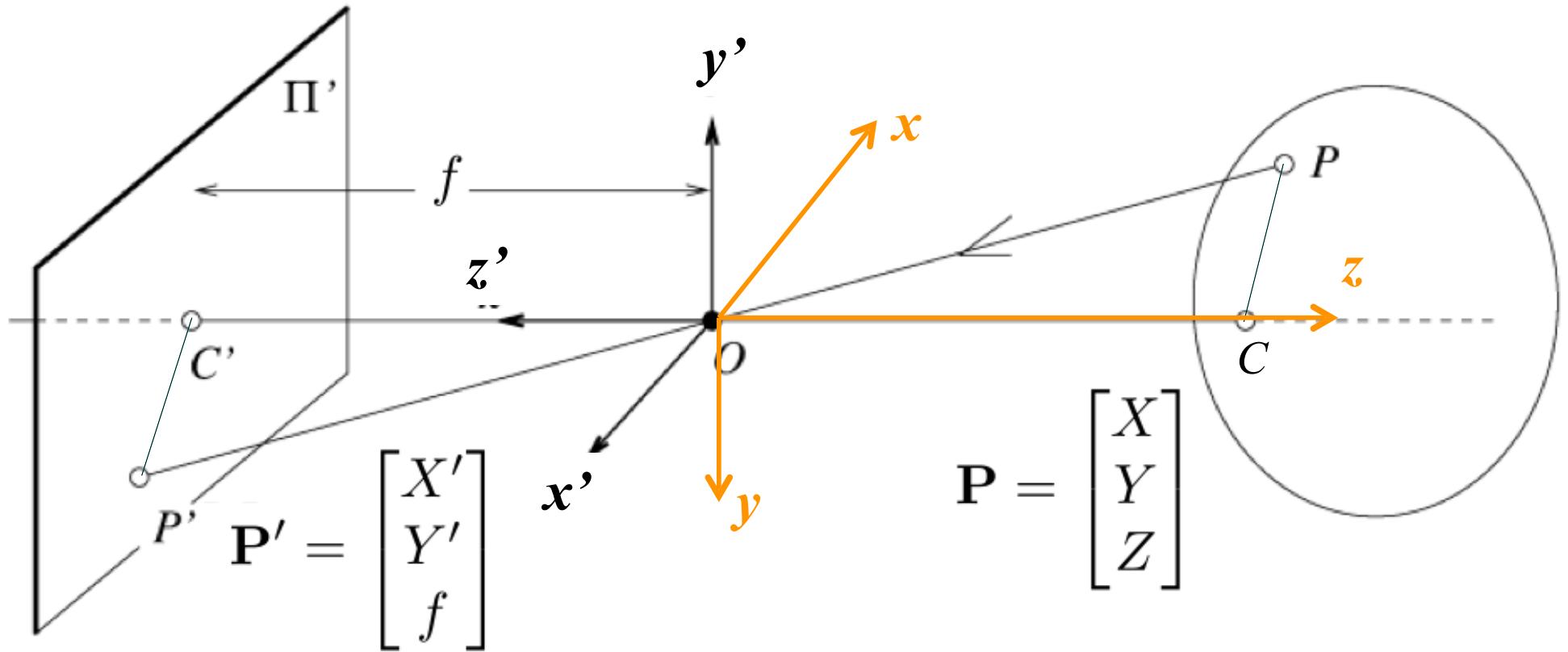


The pinhole mathematical model



- O: Origin (pinhole)
- $[\mathbf{x} \mathbf{y} \mathbf{z}]$: **World Coordinate System (WCS)**
- $[\mathbf{x}'\mathbf{y}'\mathbf{z}']$: **Camera Coordinate System (CCS)**
- Π' : Image plane
- P, P' : Scene (wrt WCS) and Image point (wrt CCS)⁸

The pinhole mathematical model, cont.

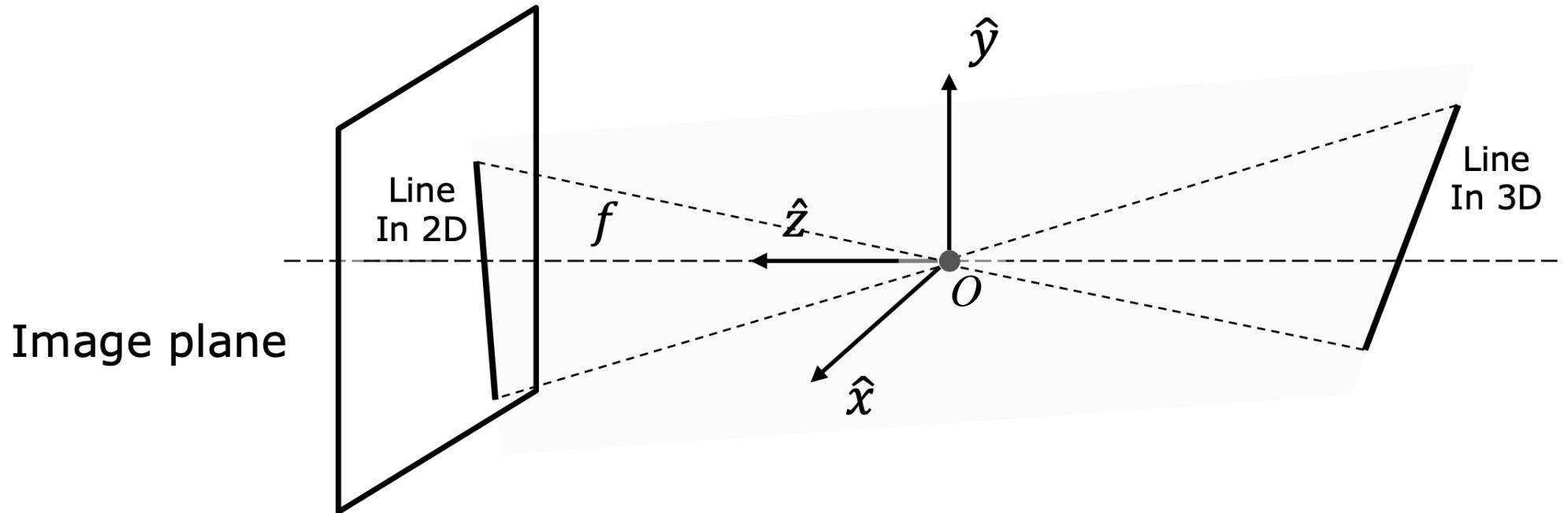


Mathematical model (wrt Image Plane):

- **The triangles OPC and $OP'C'$ are similar**

Perspective equations: $X' = f \frac{X}{Z}; \quad Y' = f \frac{Y}{Z}$

Perspective Properties: Projection of a line



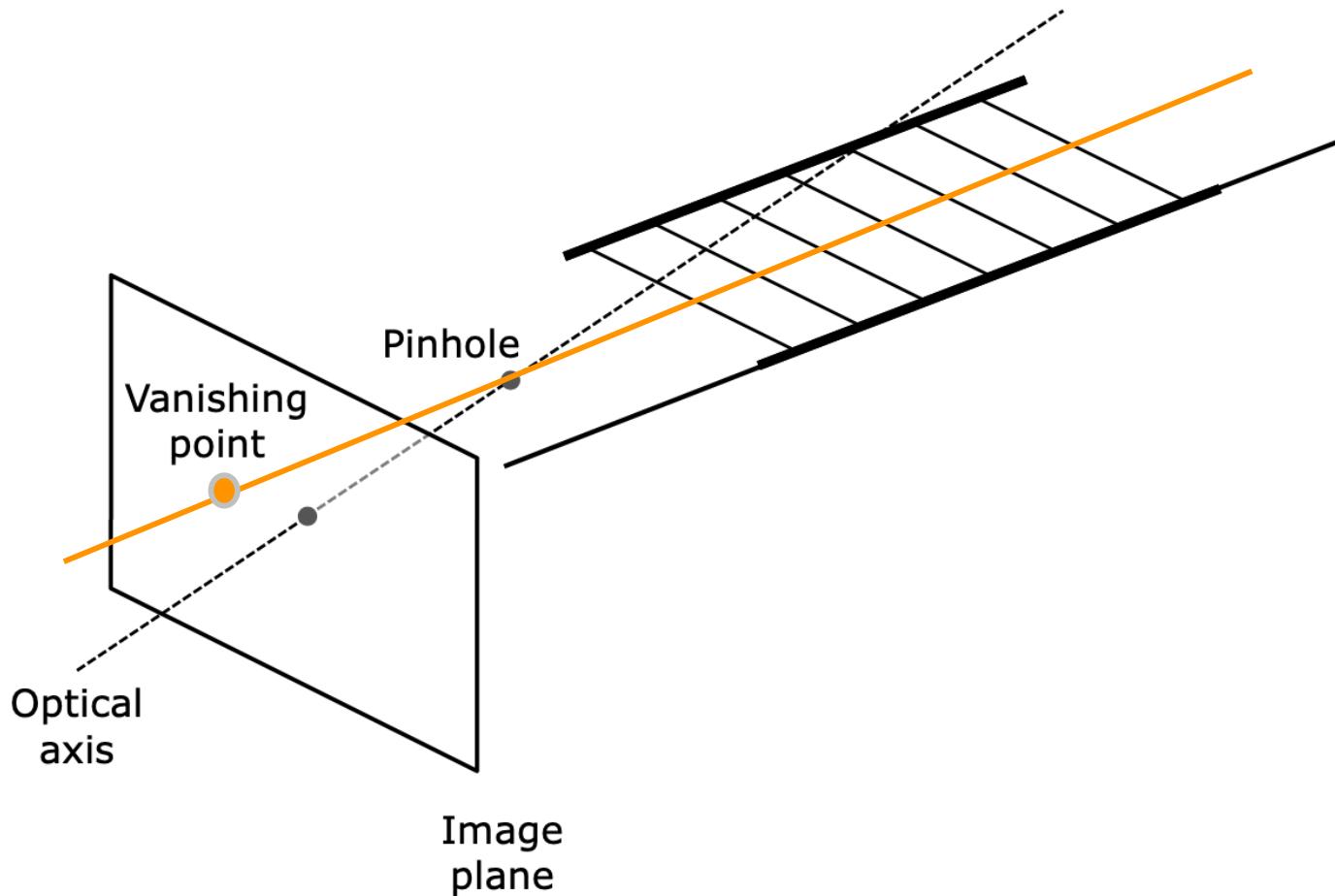
- Straight line in scene remains straight in image

Perspective Properties: Vanishing point



- Parallel scene lines converge at a single image point: the **vanishing point**
- Its location depends on orientation of parallel scene lines

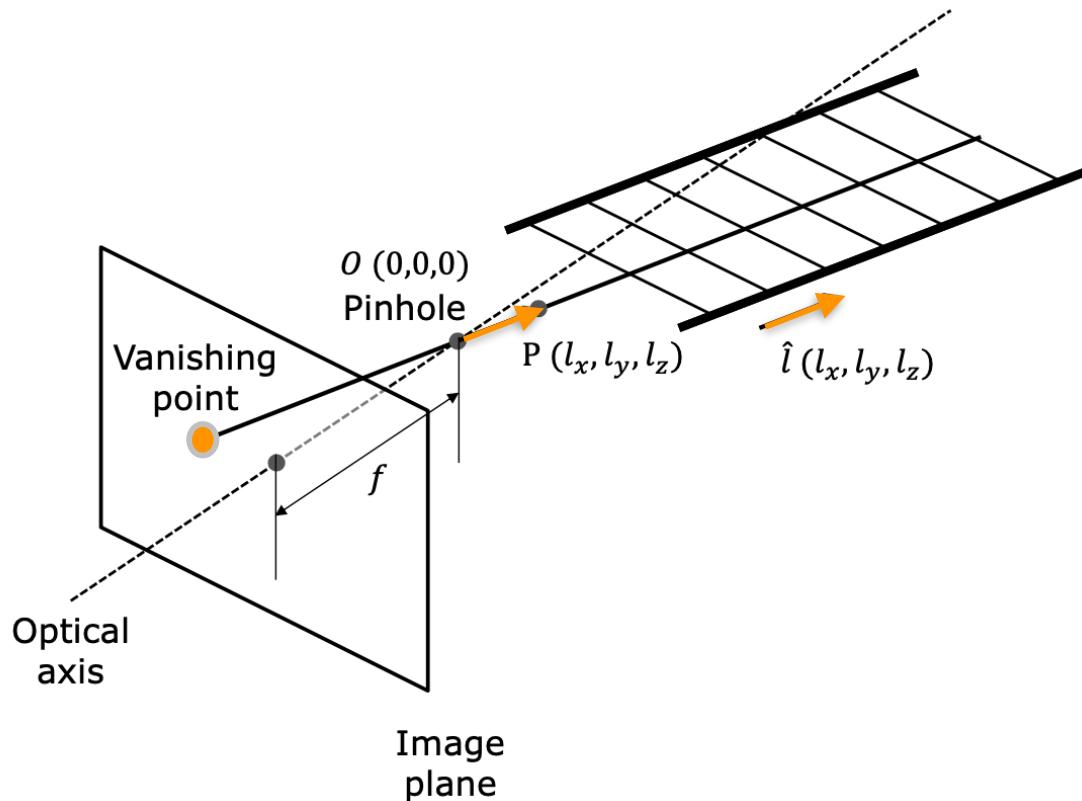
Perspective Properties: Finding the vanishing point



- **Intuitively:**

- trace a line parallel to the track AND passing through the Pinhole.
- the point where this line pierces the image is the location of the vanishing point

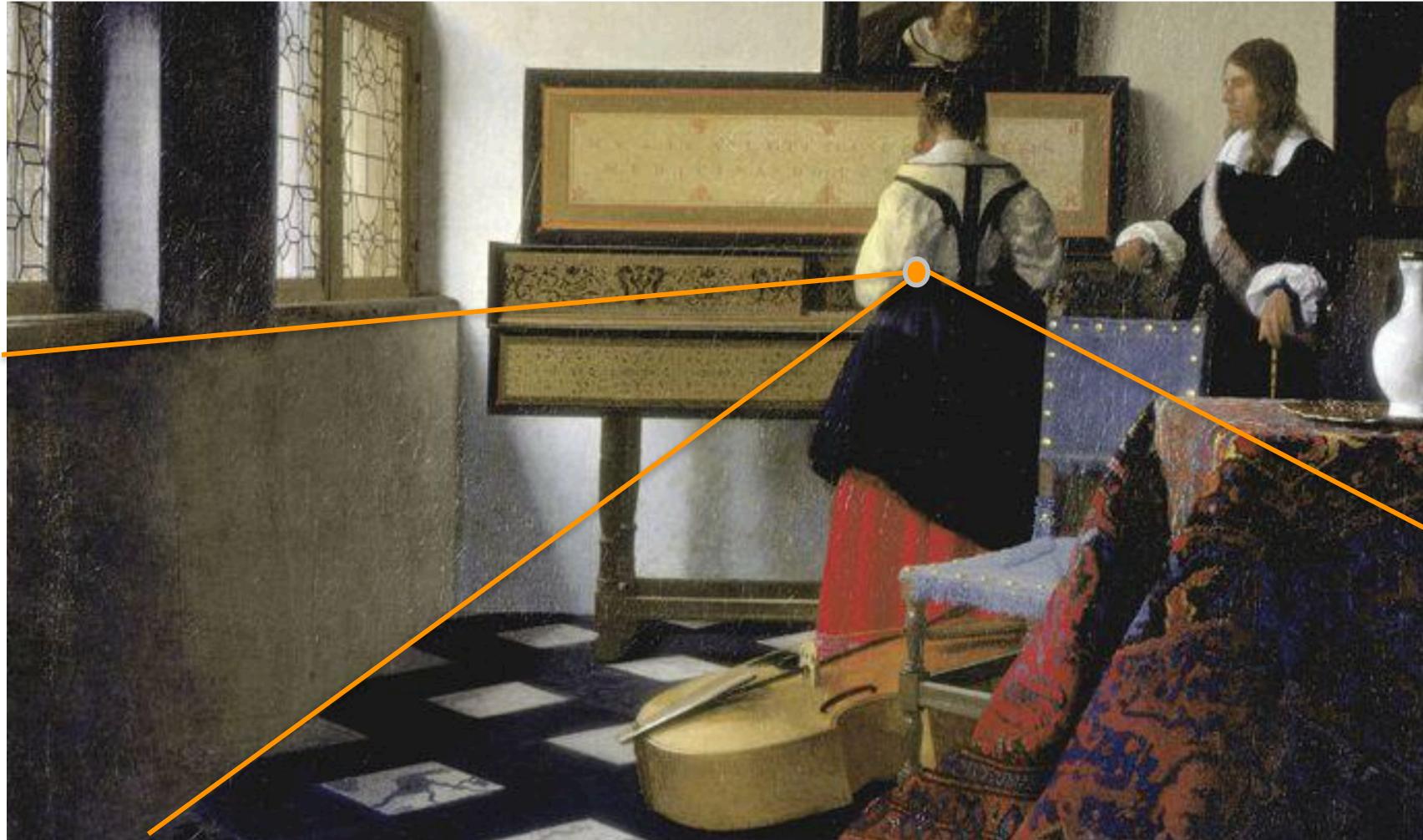
Perspective Properties: Finding the vanishing point



- **Mathematically:**

- define the direction of the line $\hat{l} = (l_x, l_y, l_z)$
- determine P as the point at distance \hat{l} from the origin
- **vanishing point:** projection of point P :
$$\left(f \frac{l_x}{l_z}, f \frac{l_y}{l_z} \right)$$

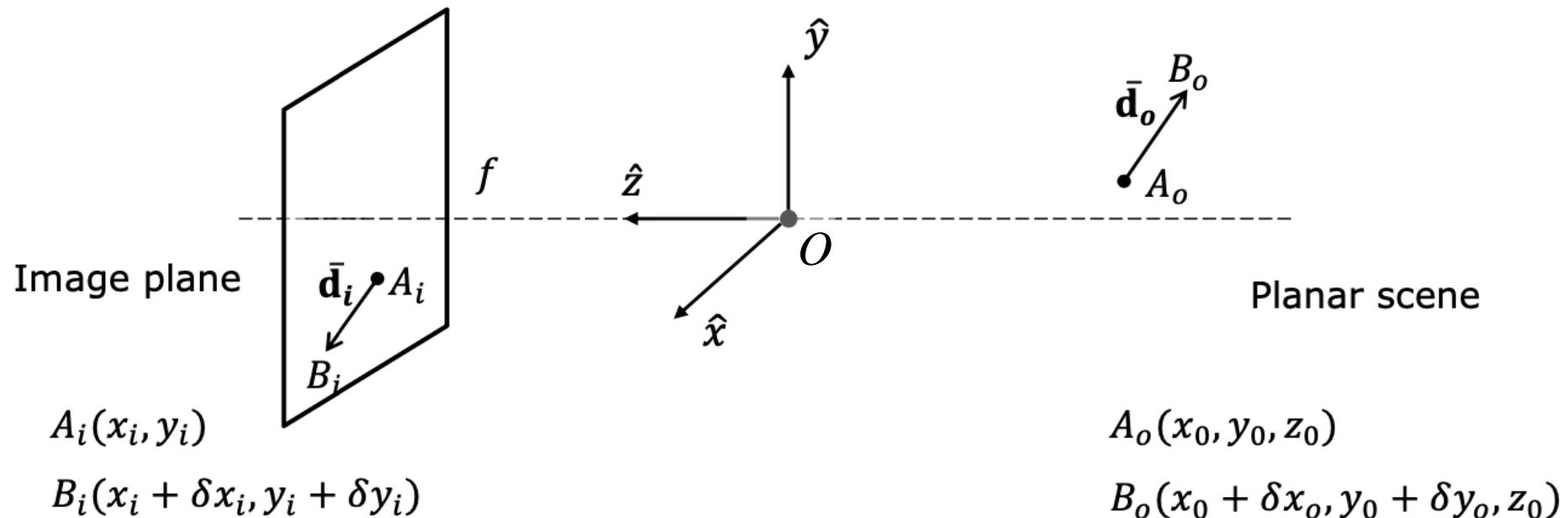
Vanishing point used in art



The Music Lesson, Johannes Vermeer, c. 1662-1664

- To make the attention converge to the most important subject in the scene

Perspective Properties: Image magnification



Magnification: ratio between the apparent size of an object in an image ($\|\bar{\mathbf{d}}_i\|$) relative to its actual size in real life ($\|\bar{\mathbf{d}}_o\|$)

$$|m| = \frac{\|\bar{\mathbf{d}}_i\|}{\|\bar{\mathbf{d}}_o\|} = \frac{\sqrt{\delta x_i^2 + \delta y_i^2}}{\sqrt{\delta x_o^2 + \delta y_o^2}}$$

Perspective Properties: Image magnification

- recap of the quantity at stake:
 - world coords:

$$A_o = (x_o, y_o, z_o)$$

$$B_o = (x_o + \delta x_o, y_o + \delta y_o, z_o)$$

$$|m| = \frac{\|\mathbf{d}_i\|}{\|\mathbf{d}_0\|} = \frac{\sqrt{\delta x_i^2 + \delta y_i^2}}{\sqrt{\delta x_0^2 + \delta y_0^2}}$$

- image coords:

$$A_i = (x_i, y_i)$$

$$B_i = (x_i + \delta x_i, y_i + \delta y_i)$$

→ to compute m , let's start by rewriting the image coords using perspective projection:

$$x_i = f \frac{x_0}{z_0}, \quad y_i = f \frac{y_0}{z_0},$$

$$x_i + \delta x_i = f \frac{x_0 + \delta x_0}{z_0}, \quad y_i + \delta y_i = f \frac{y_0 + \delta y_0}{z_0}$$

Perspective Properties: Image magnification

Substituting x_i

$$1. \quad x_i = f \frac{x_0}{z_0}$$

$$x_i + \delta x_i = f \frac{x_0 + \delta x_0}{z_0}$$

$$2. \quad f \frac{x_0}{z_0} + \delta x_i = f \frac{x_0}{z_0} + f \frac{\delta x_0}{z_0}$$

$$3. \quad \delta x_i = f \frac{\delta x_0}{z_0}$$

The same wrt y_i

$$1. \quad y_i = f \frac{y_0}{z_0}$$

$$y_i + \delta y_i = f \frac{y_0 + \delta y_0}{z_0}$$

$$2. \quad f \frac{y_0}{z_0} + \delta y_i = f \frac{y_0}{z_0} + f \frac{\delta y_0}{z_0}$$

$$3. \quad \delta y_i = f \frac{\delta y_0}{z_0}$$

Perspective Properties: Image magnification

Given: $\delta x_i = f \frac{\delta x_0}{z_0}, \quad \delta y_i = f \frac{\delta y_0}{z_0}$

Magnification becomes:

$$|m| = \frac{\|\mathbf{d}_i\|}{\|\mathbf{d}_0\|} = \frac{\sqrt{\delta x_i^2 + \delta y_i^2}}{\sqrt{\delta x_0^2 + \delta y_0^2}} = \left| \frac{f}{z_0} \right|$$

Perspective Properties: Image magnification

- $m = \frac{f}{z_0} \rightarrow$ Image size inversely proportional to depth



Perspective Properties: Image magnification

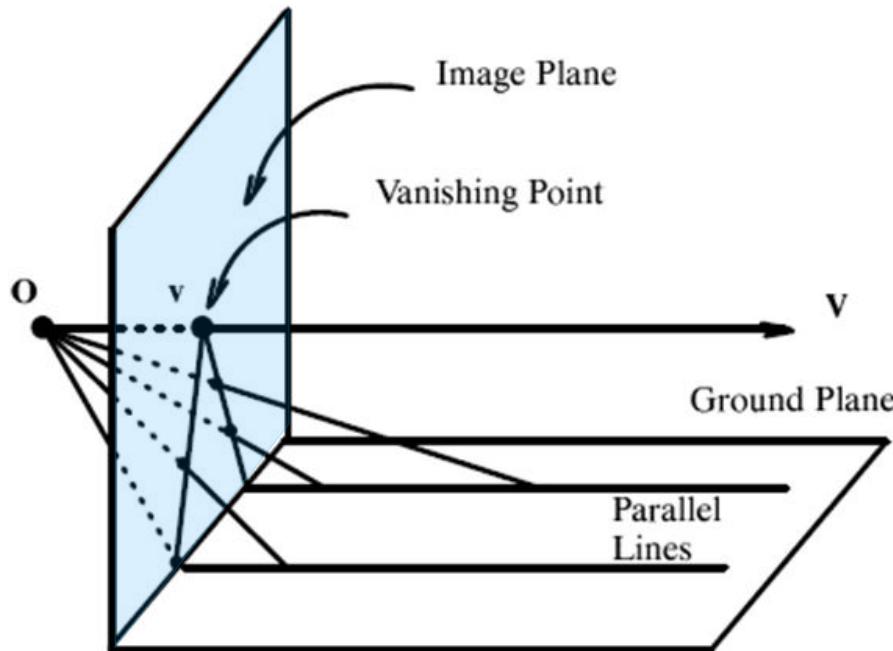
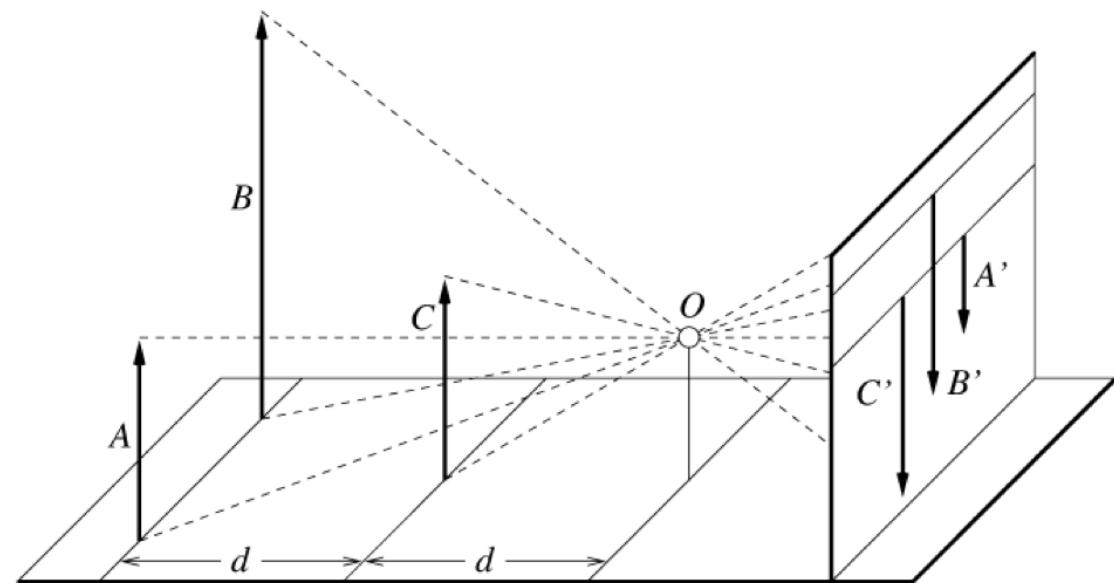
- **Remark:** m can be assumed to be constant if the range of scene depth Δz (size of the object) is much smaller than the average scene depth z
- That's not the case when the size of an object is significant compared to its distance from the camera.



- Eg: **selfies**: the nose is more magnified than other parts of the head as it is closed to the camera

Perspective properties (central perspective) [RECAP]

- **Image Size** inversely proportional to the distance
- **Straight lines** remain straight
- **Corners** are not preserved
- **Parallel lines** → converging lines



What is the ideal pinhole size?

Dependency on the Pinhole diameter:

- **reducing:**

- More and more focused images
(but there is the **diffraction limit**)
- Scarce light energy

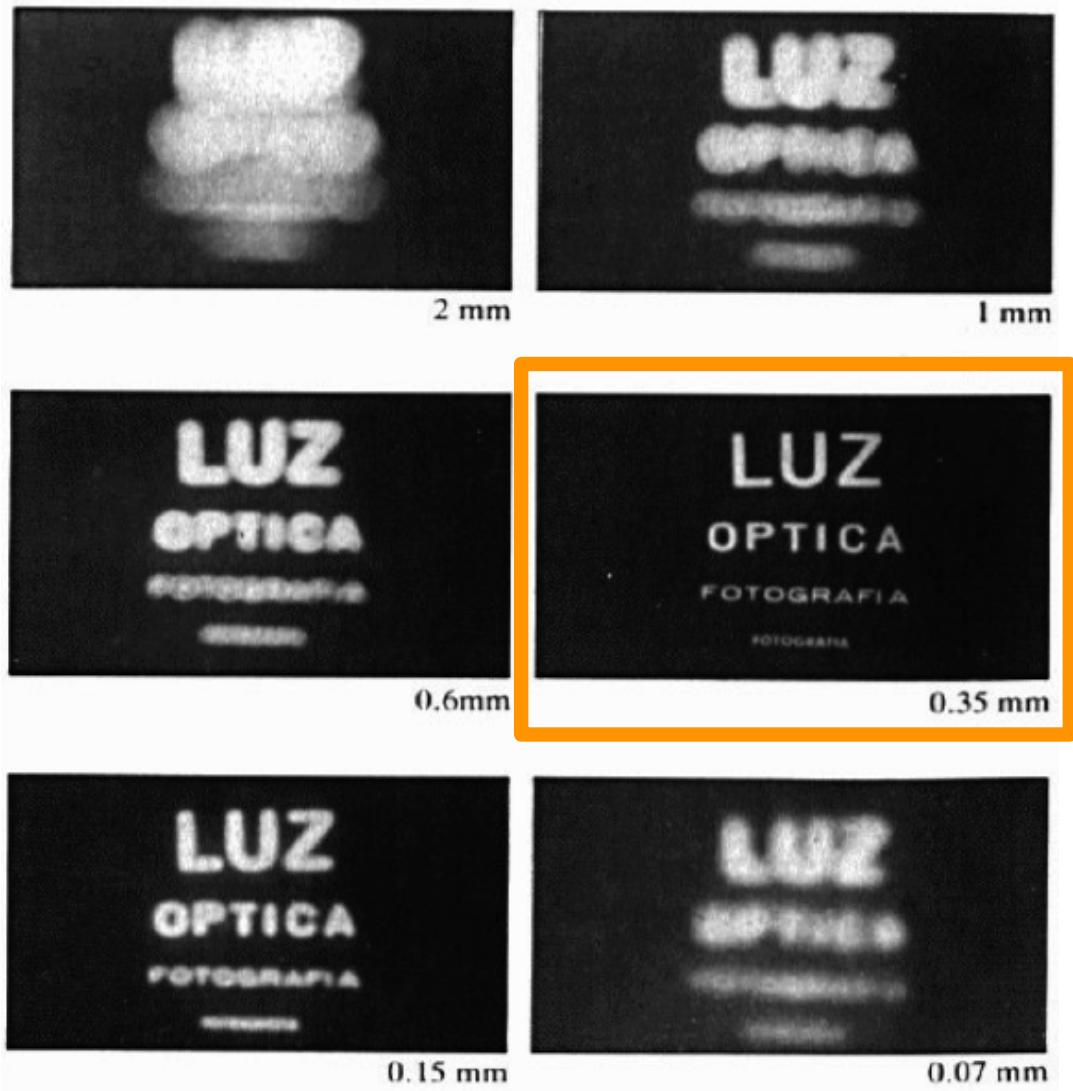
- **augmenting:**

- More light
- More and more blurred images

- **Ideal pinhole diameter**

$$d \approx 2\sqrt{f\lambda}$$

- f : focal length
- λ :wavelength



What about the exposure time?

- focus everywhere with:
- $f = 73mm, d = 0.2mm$
- **Exposure: $T = 12s!!!$**
- Too long: we need lens



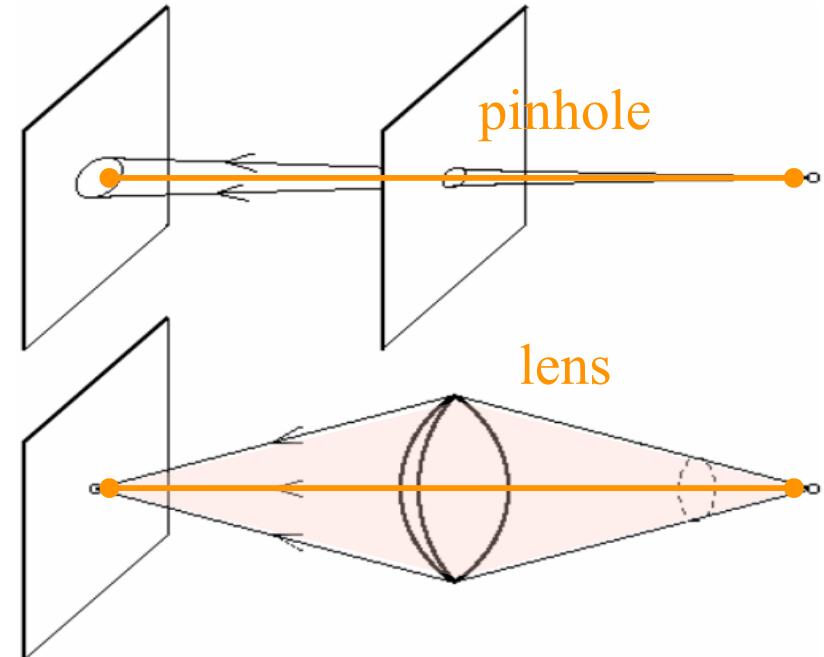
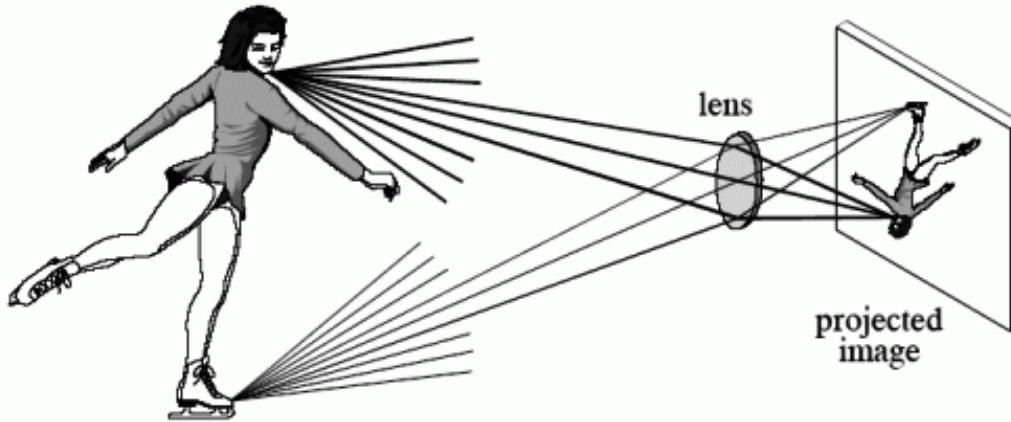
THIN LENS CAMERA

Chapter 1 – Forsyth Ponce

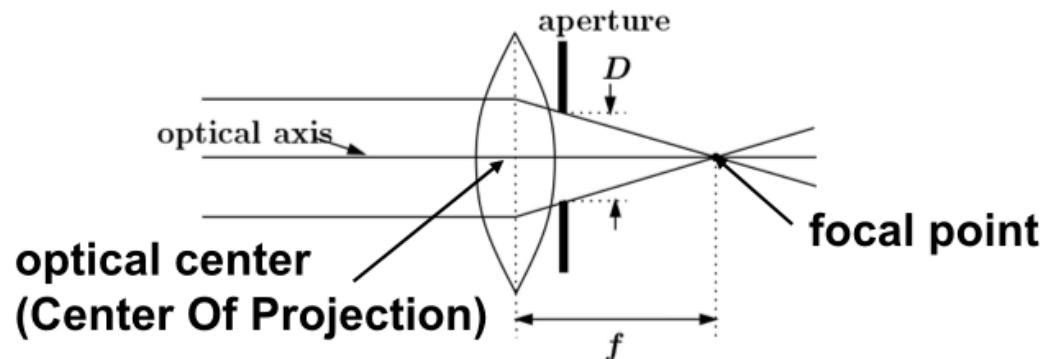
Pinhole vs. Lens

A **pinhole** camera captures a beam of light (object-image straight line)

A **lens** captures all the rays of light that reach its opening



Lens Aperture:
diameter/focal length ratio



$$\text{Aperture: } A = \frac{D}{f}$$

Camera with lens

Geometrical model of a “**thin lens**” (converging lens)

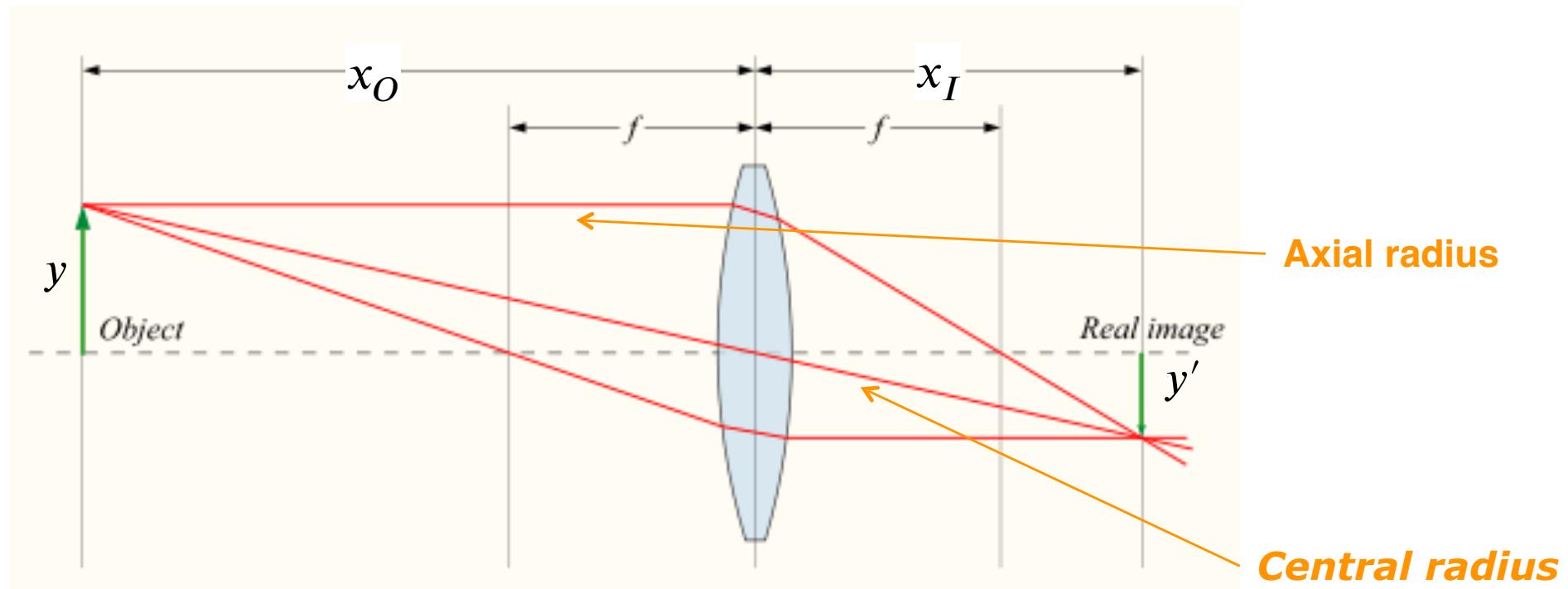
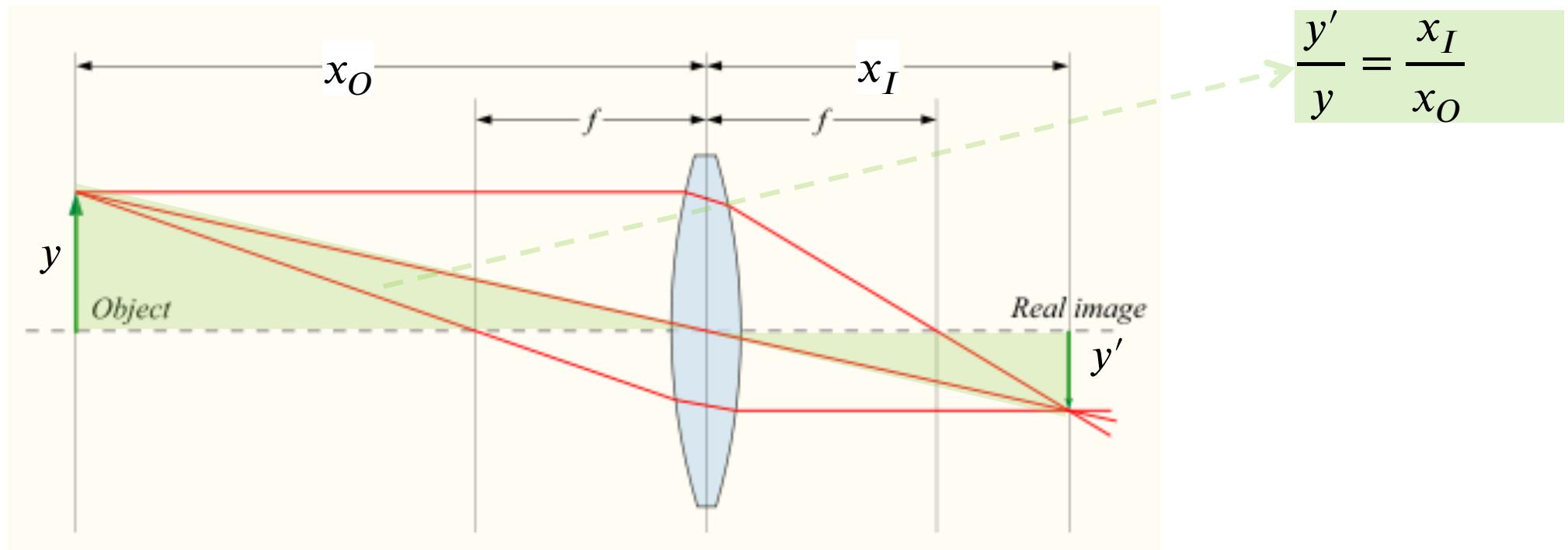


Image construction

- The **central radius** passes the lens centre without refraction
- The **axial radius** is parallel to the optical axis, and is refracted passing through the focal point

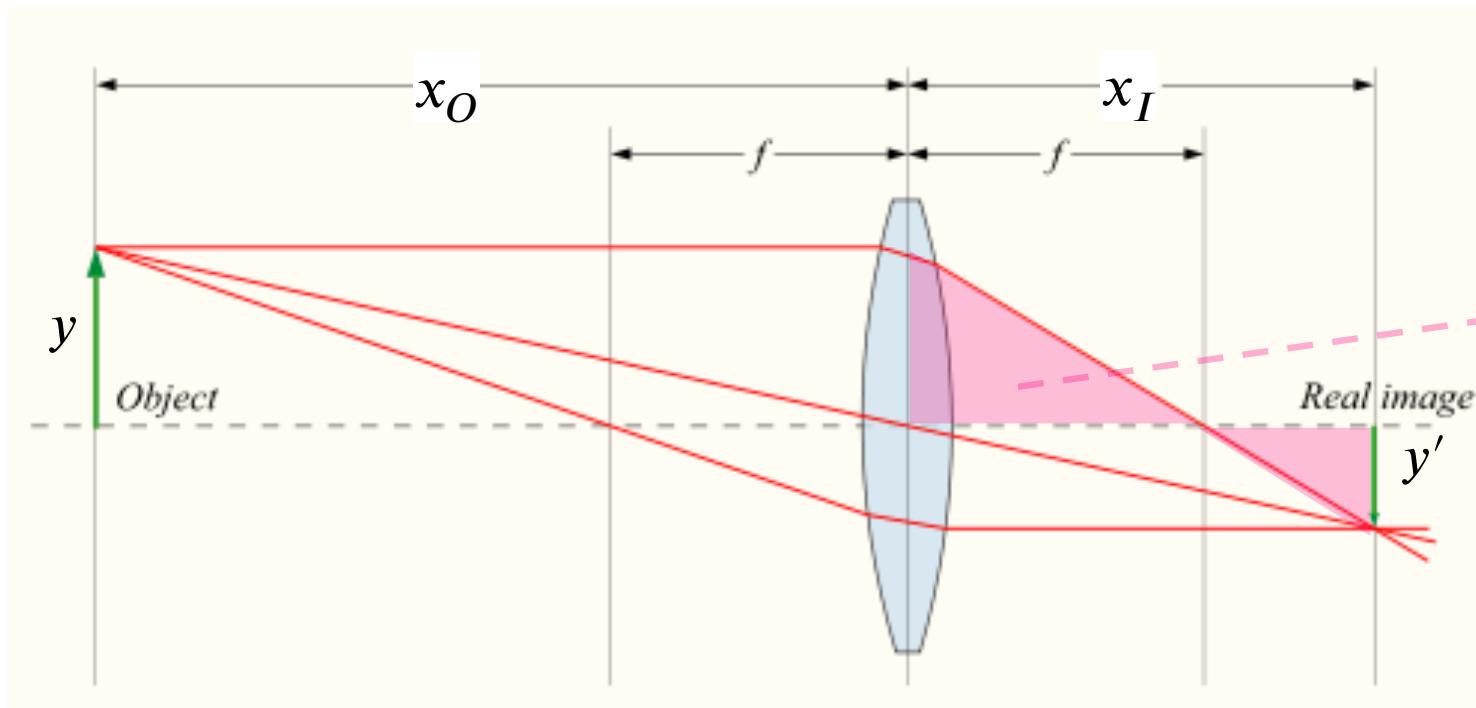
Camera with lens

1. From the similarity between the **green** triangles we have:



Camera with lens

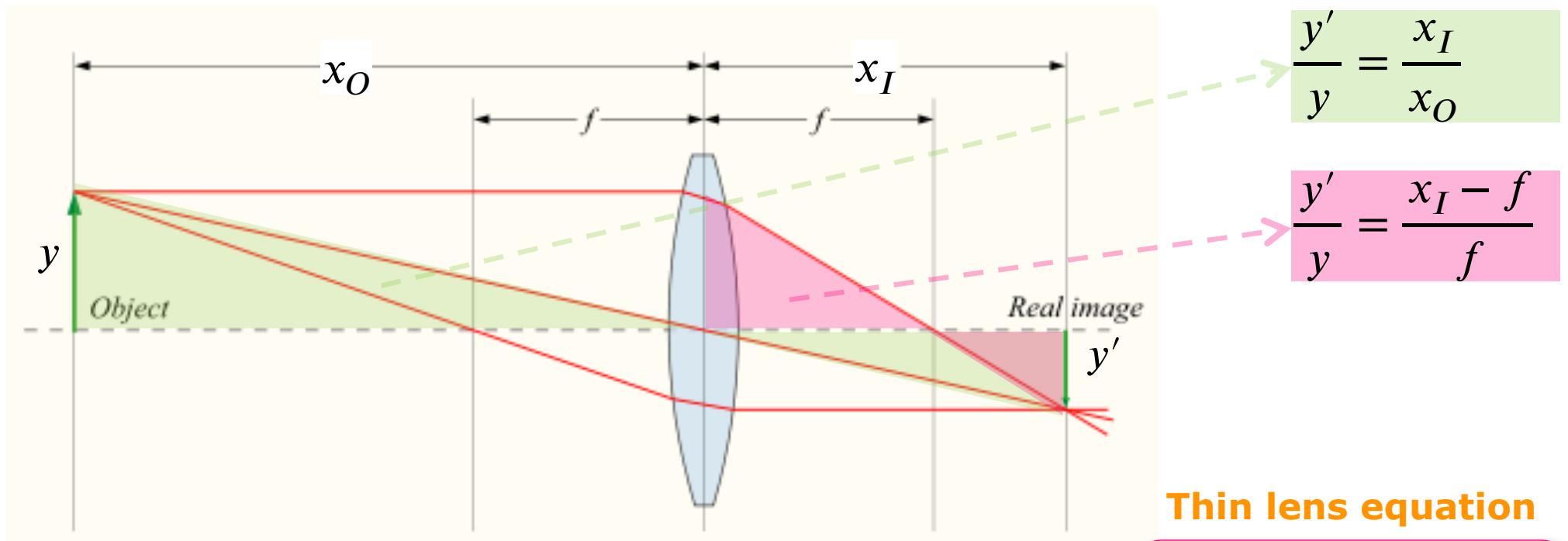
2. From the similarity between the **pink** triangles we have:



$$\frac{y'}{y} = \frac{x_I - f}{f}$$

Camera with lens: Gaussian lens law

3. By equating the two equations and dividing by x_I :



$$\frac{x_I}{x_O} = \frac{x_I - f}{f} = \frac{x_I}{f} - 1 \quad : \frac{1}{x_I} \rightarrow \frac{1}{x_O} = \frac{1}{f} - \frac{1}{x_I}$$

$$\frac{1}{x_I} + \frac{1}{x_O} = \frac{1}{f}$$

Camera with lens: Gaussian lens law

Thin lens equation

$$\frac{1}{x_I} + \frac{1}{x_O} = \frac{1}{f}$$

Sumup: with the thin lens the geometrical model does not change, while we have infocus distance depending on x_o

Examples:

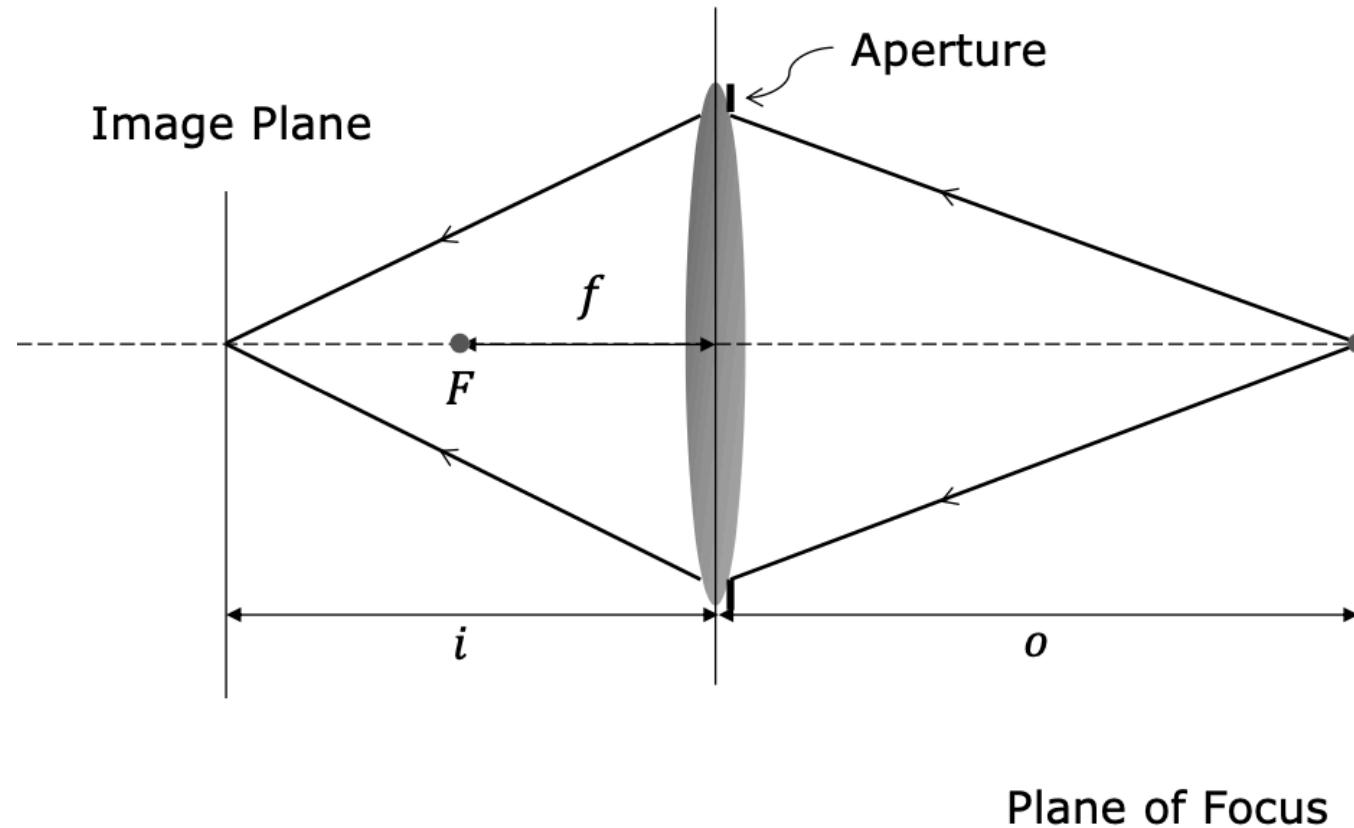
$$x_O \rightarrow \infty \implies \frac{1}{x_I} = \frac{1}{f} \implies x_I = f$$

$$x_O = 2f \implies \frac{1}{x_I} + \frac{1}{2f} = \frac{1}{f} \implies x_I = 2f$$

If $f = 50\text{mm}$, $x_o = 300\text{mm}$, then image distance is $x_i = 60\text{mm}$

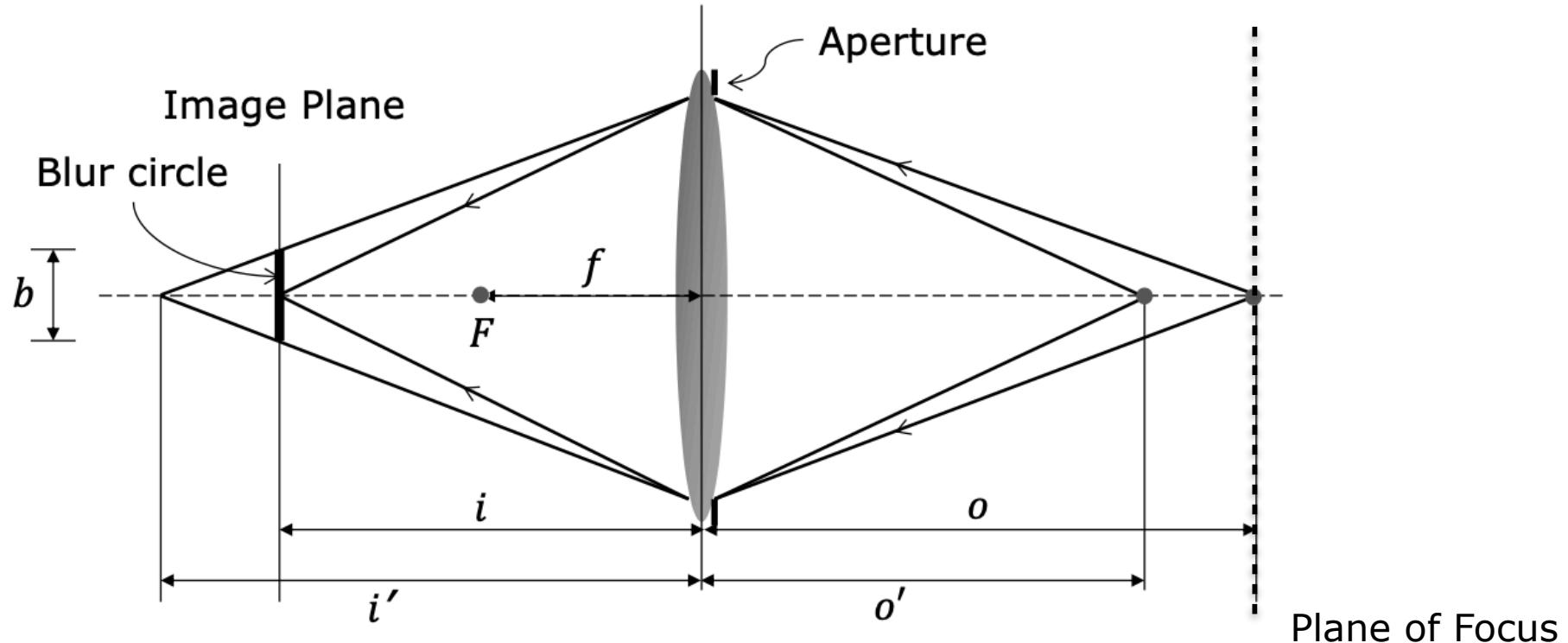
Lens defocus

- With a **lens** instead of a **pinhole** not everything is in focus (with a pinhole, yes)



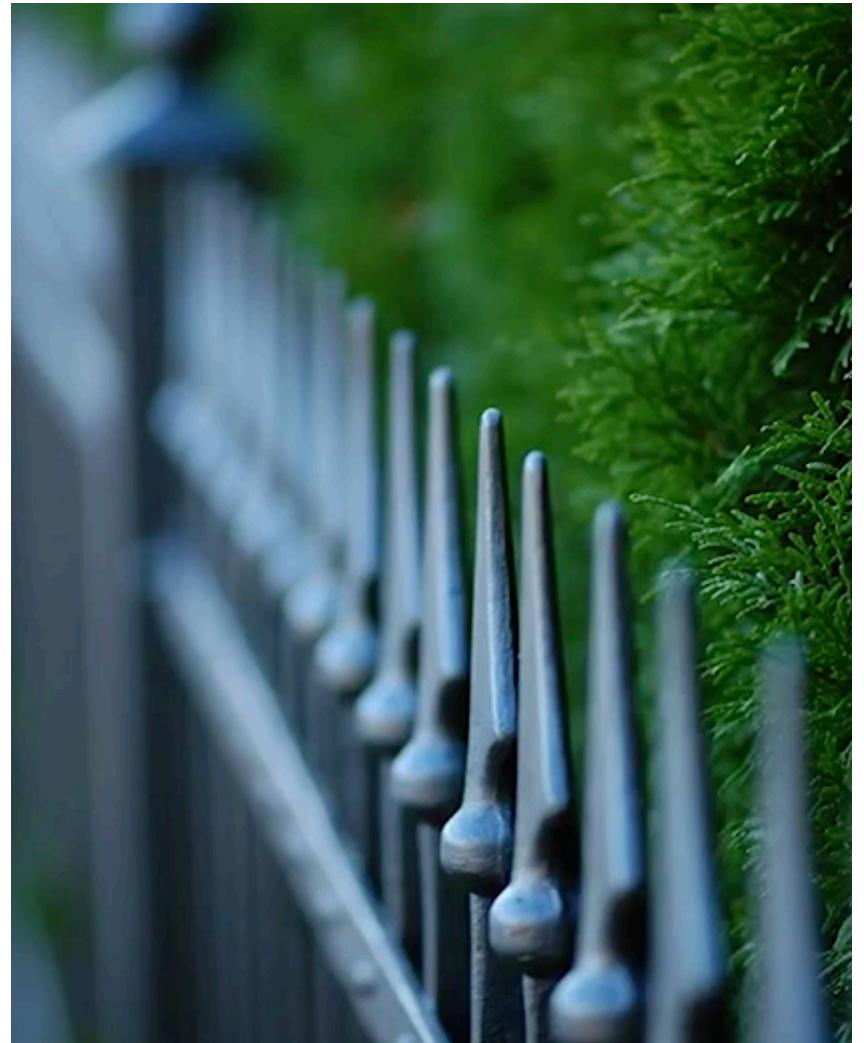
- fixing all the camera parameters, only the points on the plane of focus will be in focus on the image plane

Lens defocus



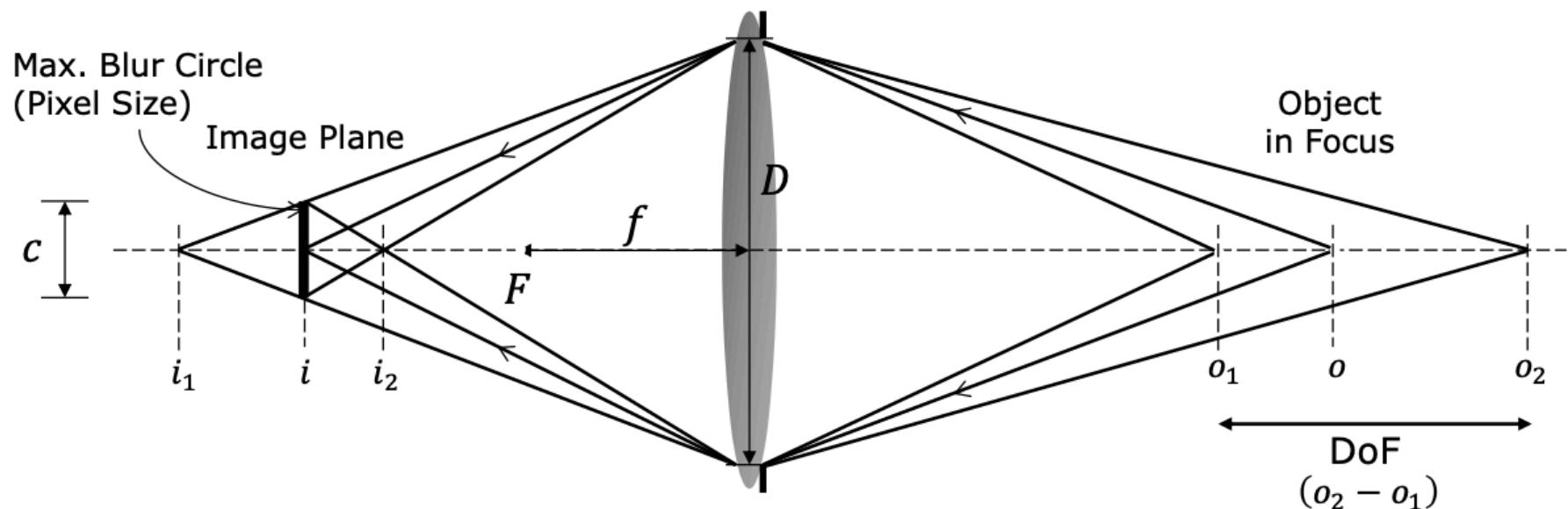
- taking a point o' outside the Plane of focus, closer to the lens
- its image is formed behind the image plane
- on the image plane: a **blur circle**
- $b \propto \text{Aperture}$

Depth of field (DoF)



DoF: range of object distances over which the image is “sufficiently well” focused, i.e., range over which blur b , is less than pixel size

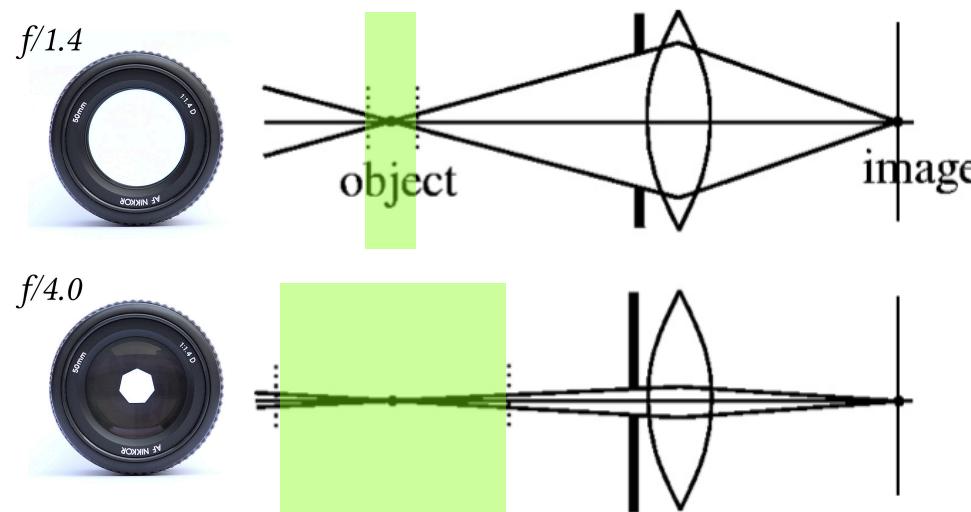
Depth of field (DoF)



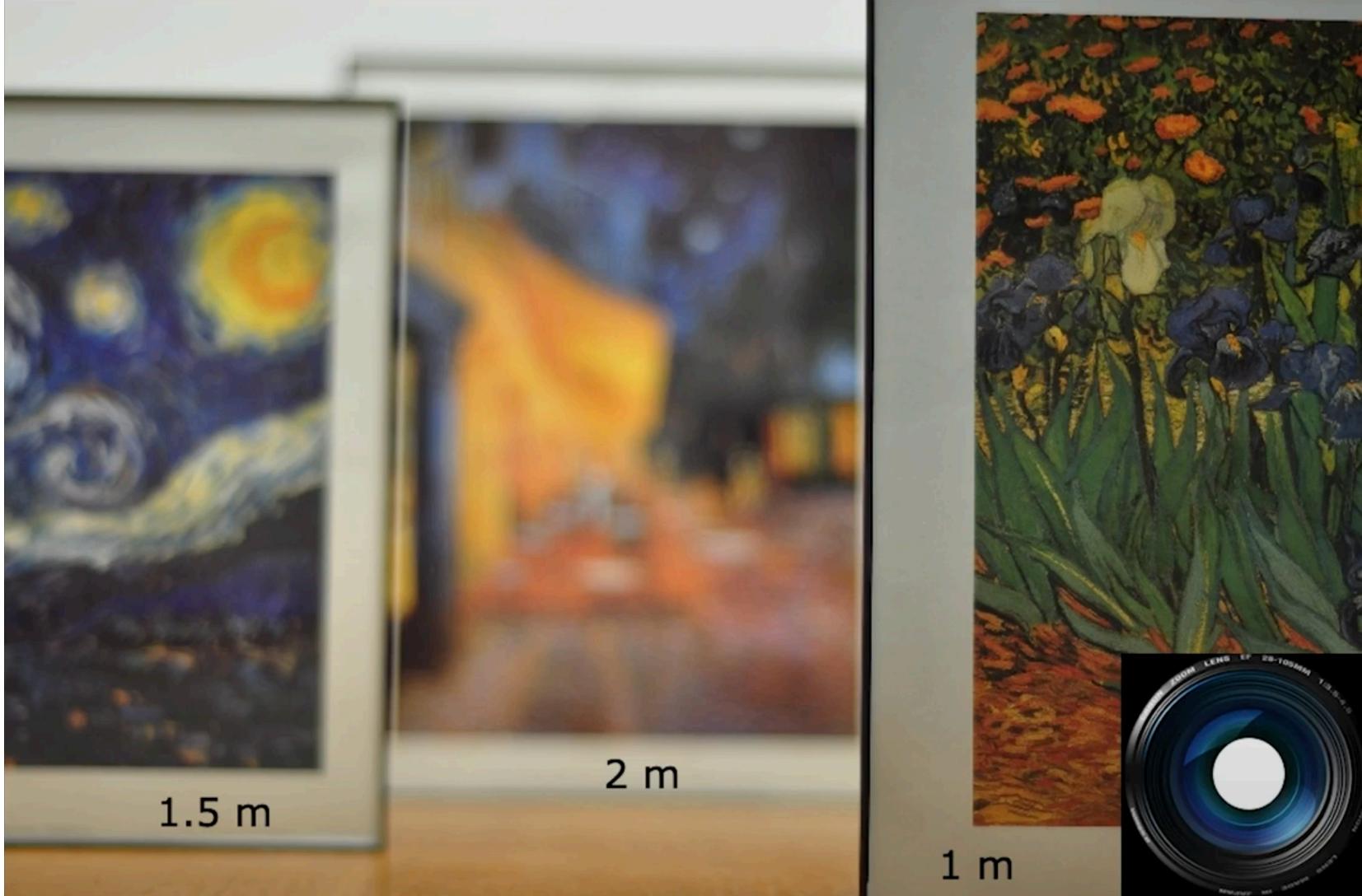
If o_1 and o_2 are the nearest and farthest distances respectively for which blur circle is maximum c , then:

$$DoF = o_2 - o_1 \propto \frac{1}{A}$$

c is taken as the pixel dimension

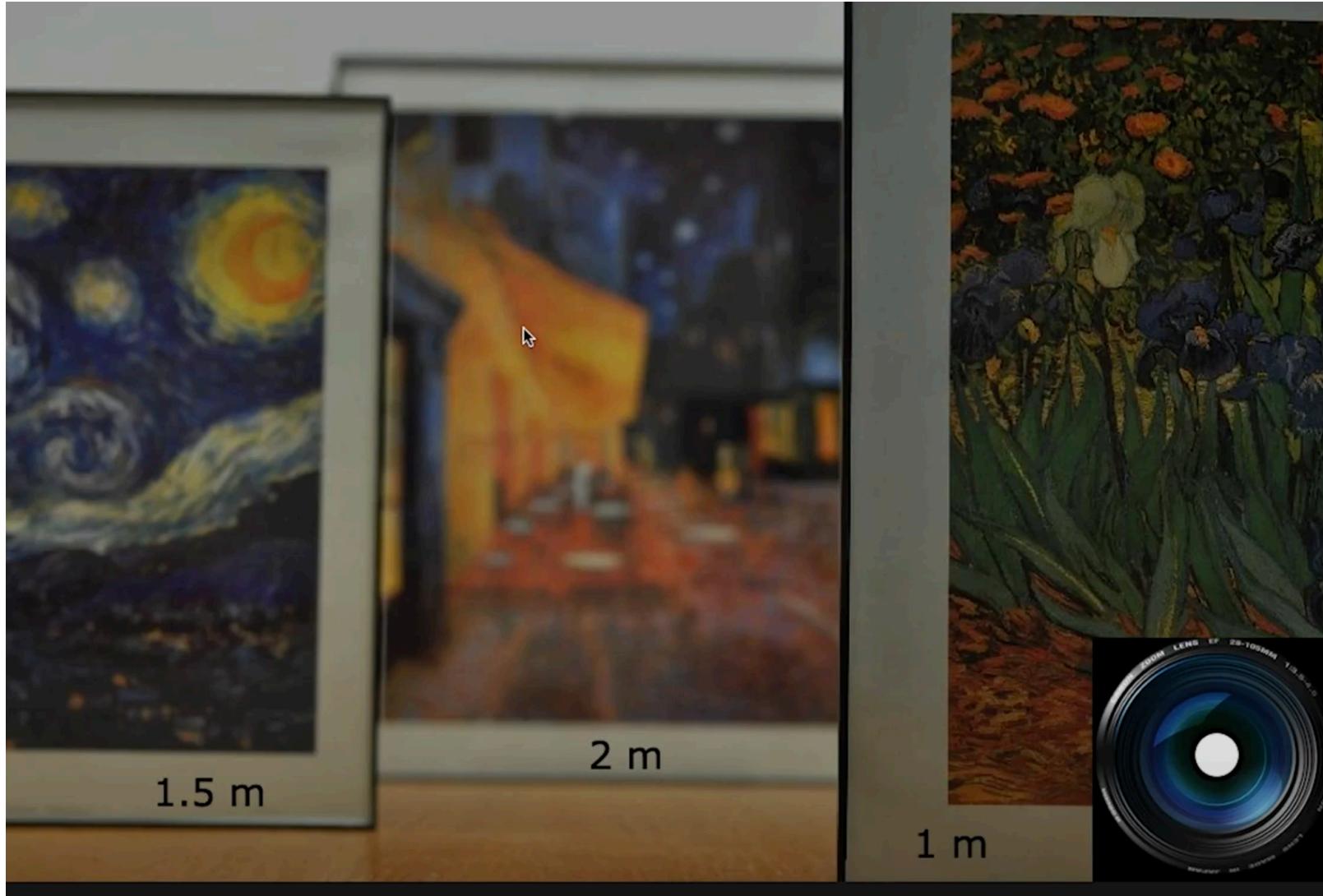


Aperture size: DoF vs Brightness



Focal Length 50 mm, Focus = 1 m, Aperture D = 25 mm

Aperture size: DoF vs Brightness



Focal Length 50 mm, Focus = 1 m, Aperture D = 12.5 mm

Aperture size: DoF vs Brightness



Focal Length 50 mm, Focus = 1 m, Aperture D = 3.125 mm

Aperture size: DoF vs Brightness

- Large Aperture
 - Bright Image or Short Exposure Time
 - Shallow Depth of Field
- Small Aperture
 - Dark Image or Long Exposure Time
 - Large Depth of Field

THICK LENS CAMERA

Real camera

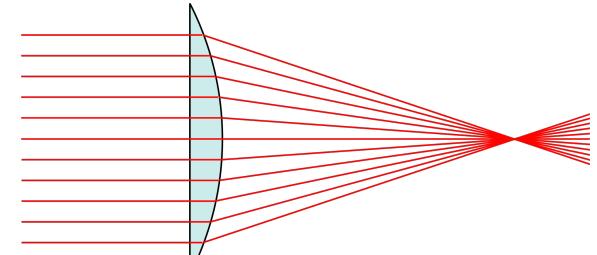
Chapter 1 – Forsyth Ponce

Lens distortions

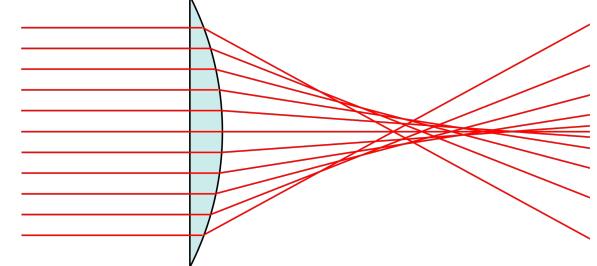
Optical aberrations of a lens

A real (thick) lens does not behave exactly like a thin lens

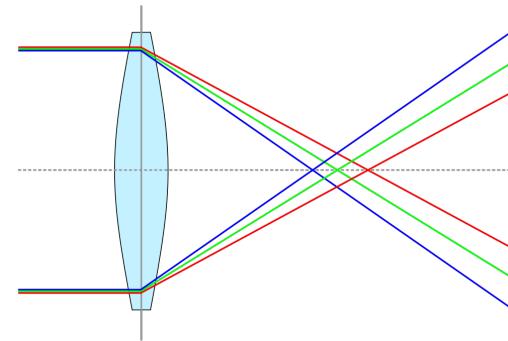
- **Spherical Aberration, Coma, Astigmatism**
 - Rays refracted by the lens form a circle of confusion on the image plane
 - **effect: unfocused image**
- **Chromatic aberration**
 - different lens behavior for different colors (wavelengths)
 - **effect: color separation**
- **Radial Distortion**
 - Image magnification changes as the distance from the central point
 - **effect: image deformation!**



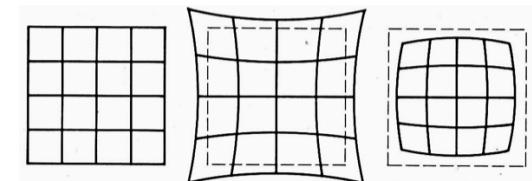
Ideal (thin) lens



Spherical Aberration



Chromatic aberration



Radial distortion

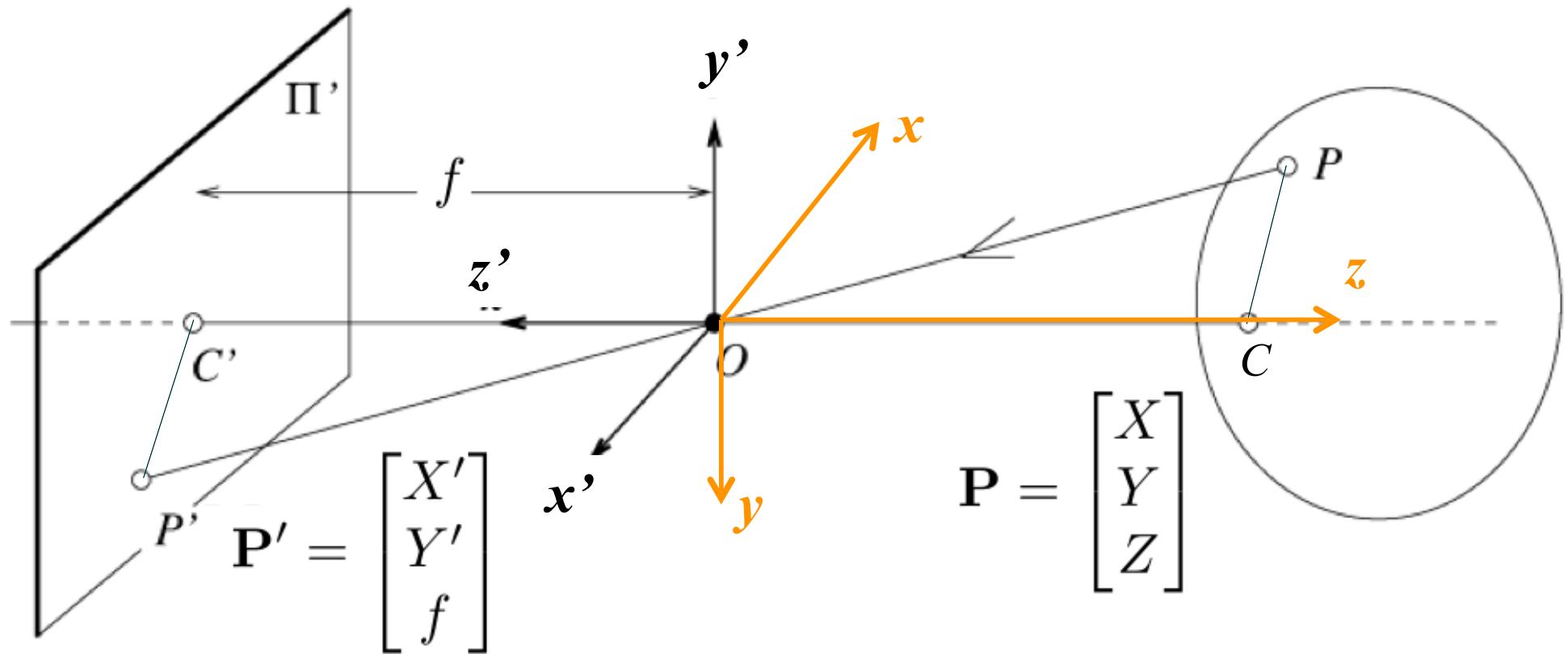
IMAGE FORMATION

Linear camera model ("pinhole")

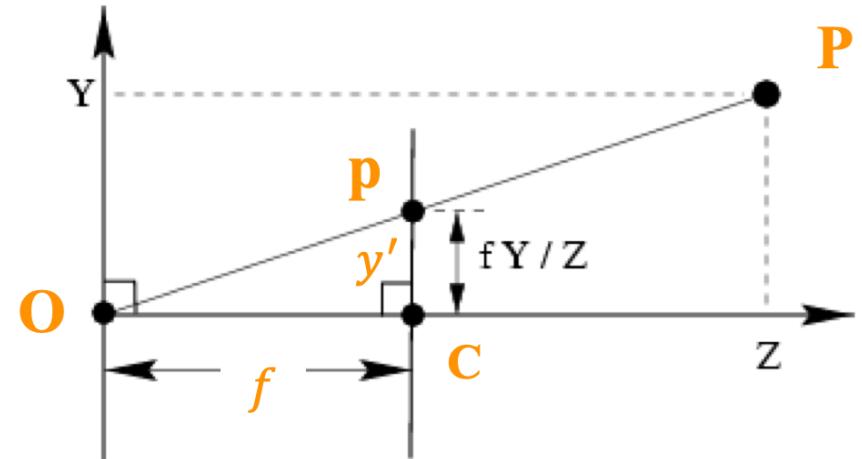
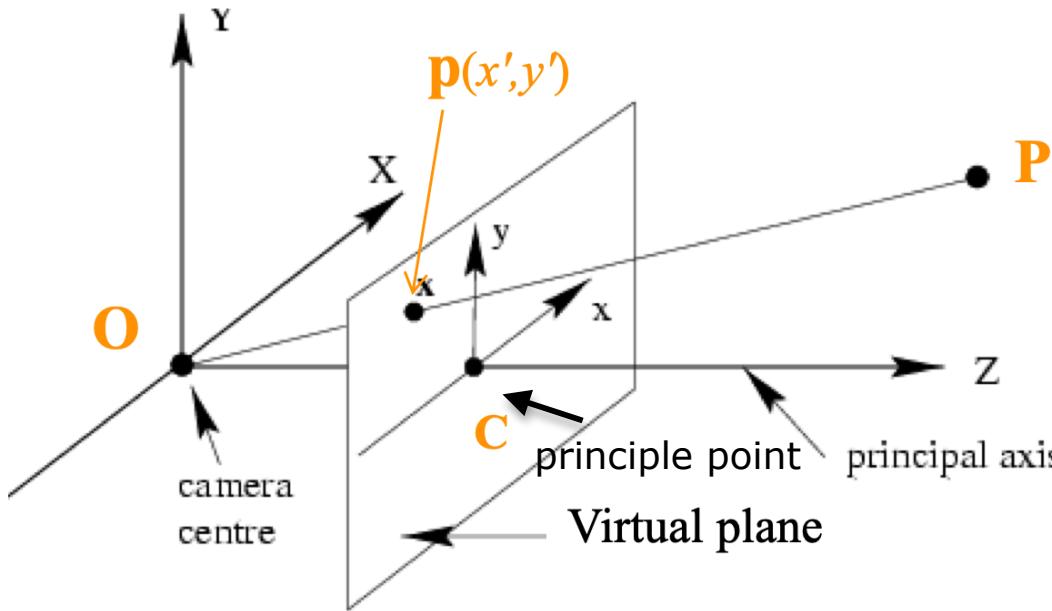
Chapter 1 – Forsyth Ponce

Forward Imaging Model

- Model that relates points in the real world P with their image points P' on the image plane



The pinhole mathematical model



- From the perspective equation it holds $\rightarrow x' = f \frac{X}{Z}; y' = f \frac{Y}{Z}$

- In Matricial form:

$$\mathbf{p} = \begin{bmatrix} x' \\ y' \end{bmatrix} \in \mathbb{R}^2; \quad \mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3 \quad \rightarrow$$

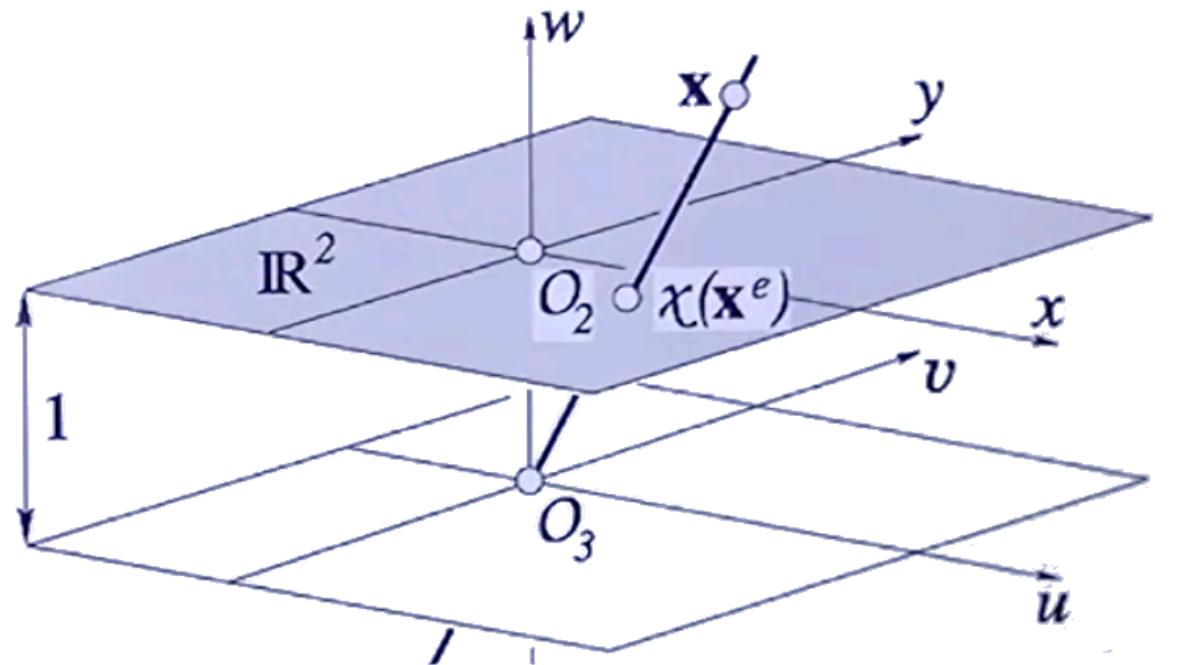
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & 0 \\ 0 & \frac{f}{Z} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} : \quad \mathbf{p} = \mathbf{M} \cdot \mathbf{P}$$

- Remark: **non linear!**

Homogeneous coordinates

- The homogeneous representation of a 2D point $\mathbf{x} = (x, y)$ is a 3D point $\tilde{\mathbf{x}} = (u, v, w)$, where the third coordinate $w \neq 0$ is called fictitious such that:

$$x = \frac{u}{w}, \quad y = \frac{v}{w}$$



- It holds:

$$\text{Euclidean } \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\text{Homogeneous } \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

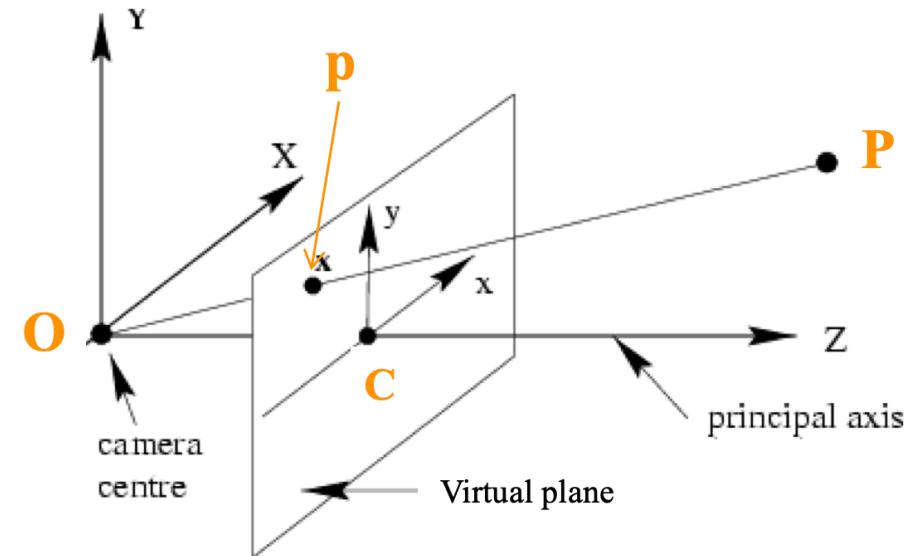
- Similarly, for any n-dimensional point

Perspective projection in homogeneous coordinates

Euclidean space

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \tilde{\mathbf{p}} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{P}} = \lambda \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Projective space



for $\lambda = 1$:

Scale factor: Z

$$\tilde{\mathbf{p}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{Z}X \\ \frac{f}{Z}Y \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \cdot \tilde{\mathbf{P}}$$

*Projection: Matrix product
K [3x4]*

In homogeneous coordinates the perspective projection becomes **linear!**

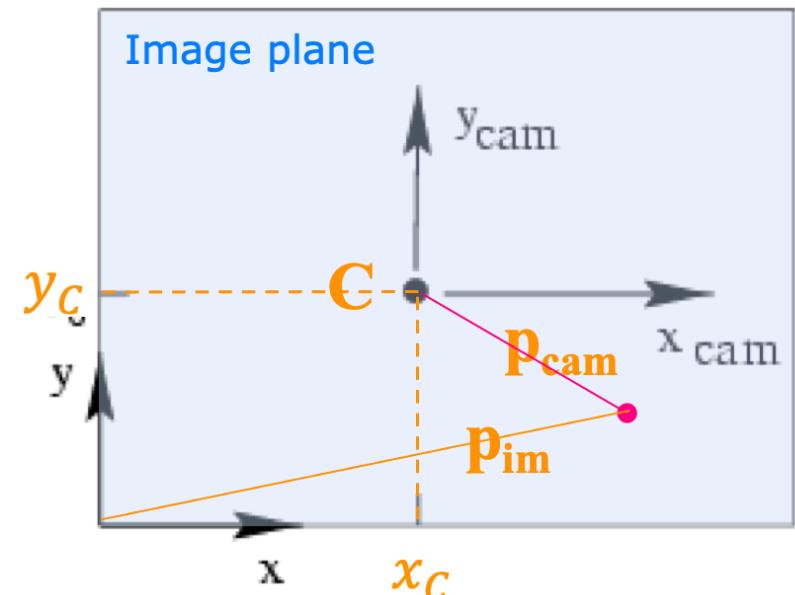
Perspective projection – a more complete camera model

1. Image coordinates (2D):

- Camera coords: \mathbf{p}_{cam} – wrt *pinhole*
- Image coords: \mathbf{p}_{im} – wrt the image plane (camera sensor)

Properties Image coords:

- axes parallel to the sensor axes
- unit of measurement: **pixel**



2. Optical centre/Principle point

- $C = \langle x_C, y_C \rangle$

$$\mathbf{p}_{\text{im}} = \mathbf{p}_{\text{cam}} + \mathbf{C} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_C \\ y_C \end{bmatrix} \Rightarrow \tilde{\mathbf{p}}_{\text{im}} = \begin{bmatrix} x + x_C \\ y + y_C \\ 1 \end{bmatrix} = \begin{bmatrix} fX/Z + x_C \\ fY/Z + y_C \\ 1 \end{bmatrix}$$

K_I : Intrinsic calibration matrix

Scale factor: Z

$$\tilde{\mathbf{P}} \xrightarrow{\text{projection}} \tilde{\mathbf{p}}_{\text{im}} = \begin{bmatrix} \frac{fX}{Z} + x_C \\ \frac{fY}{Z} + y_C \\ 1 \end{bmatrix} \equiv \begin{bmatrix} fX + x_C Z \\ fY + y_C Z \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & x_C & 0 \\ 0 & f & y_C & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}_I \cdot \tilde{\mathbf{P}}$$

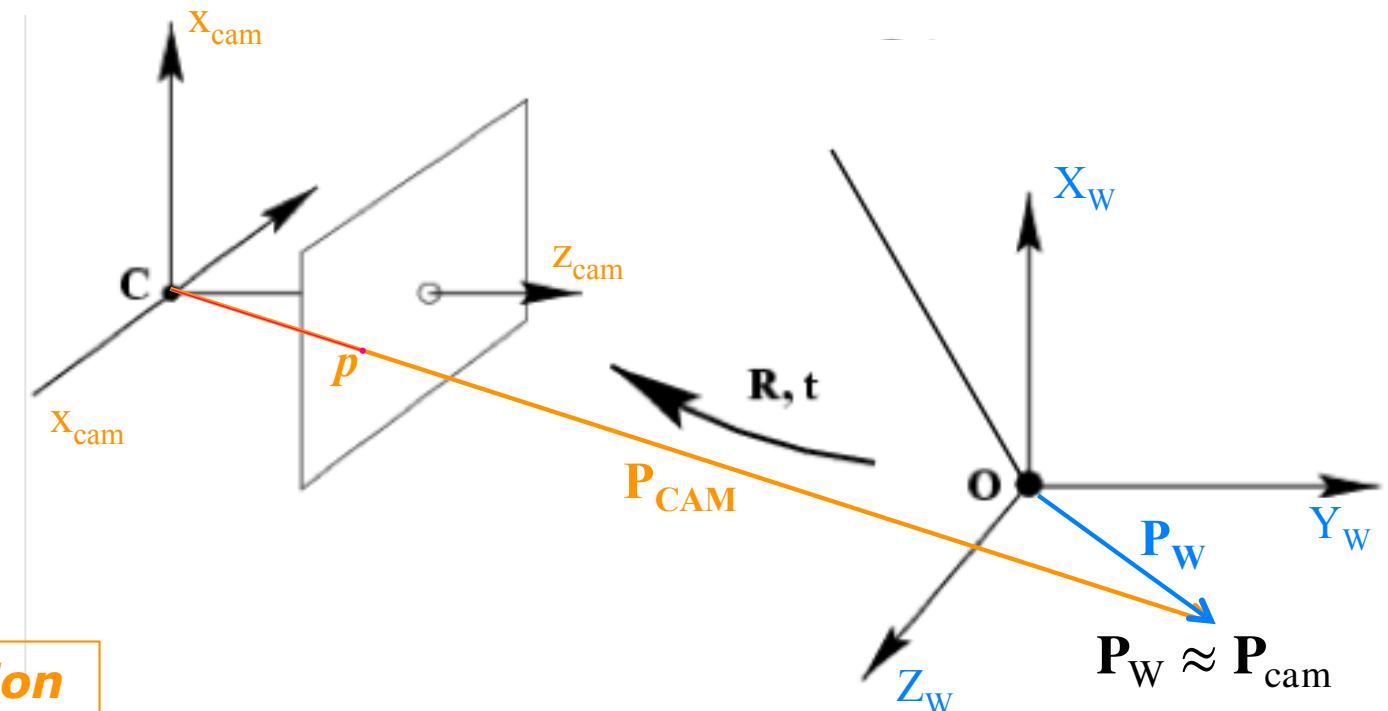
Perspective projection – a more complete camera model

3. Reference System change (World coordinates → Camera coordinates)

- **world coords** – 3D ref. System expressed wrt the **scene**
- **camera coords** – 3D ref. System expressed wrt the **camera**

- **System change:
3D roto-translation**

$$\mathbf{P}_{\text{cam}} = \begin{bmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ z_{\text{cam}} \end{bmatrix}; \quad \mathbf{P}_W = \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix}$$



Rotation **Translation**

$$\mathbf{P}_{\text{cam}} = \mathbf{R} \cdot \mathbf{P}_W + \mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{cases} x_{\text{cam}} = \mathbf{r}_1 \cdot \mathbf{P}_W + t_x \\ y_{\text{cam}} = \mathbf{r}_2 \cdot \mathbf{P}_W + t_y \\ z_{\text{cam}} = \mathbf{r}_3 \cdot \mathbf{P}_W + t_z \end{cases}$$

Rotation matrix

R: 3D rotation matrix composed of 3 rotations, on the 3 axes x,y,z

- described by 3 rotation angles (**Eulero angles**)

$$\begin{aligned} \mathbf{R} &= \mathbf{R}(\varphi, \vartheta, \rho) = \mathbf{R}(\varphi) \cdot \mathbf{R}(\vartheta) \cdot \mathbf{R}(\rho) = \\ &= \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\vartheta) & 0 & -\sin(\vartheta) \\ 0 & 1 & 0 \\ \sin(\vartheta) & 0 & \cos(\vartheta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix} \end{aligned}$$

Around Z *Around Y* *Around X*

← *Rotation order* →

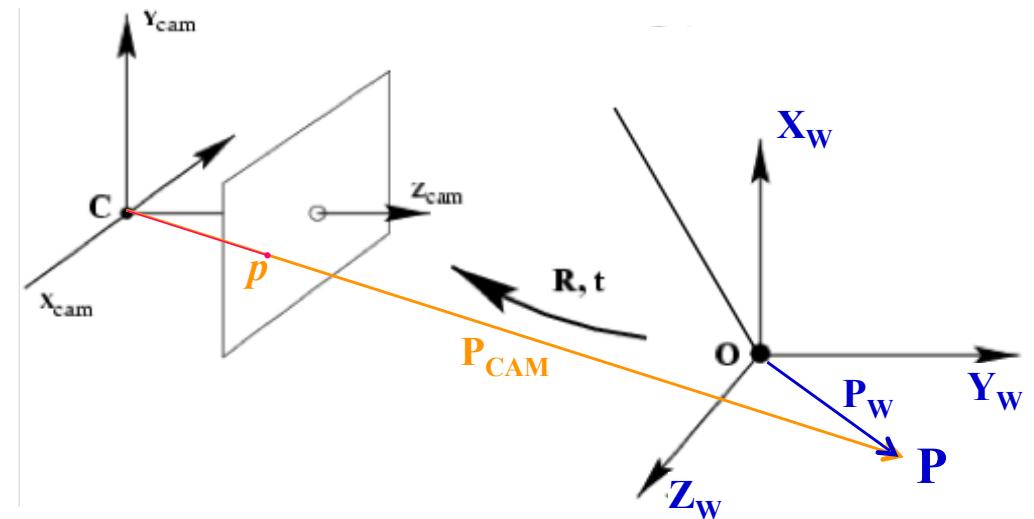
Perspective projection – a more complete camera model

Geometric relationship between Camera coordinates and World coordinates in homogeneous coords:

$$\tilde{\mathbf{P}}_{CAM} = \begin{bmatrix} X_{CAM} \\ Y_{CAM} \\ Z_{CAM} \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \cdot \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{T} \\ \hline \mathbf{0} & 1 \end{array} \right] \cdot \tilde{\mathbf{P}}_W = \mathbf{E} \cdot \tilde{\mathbf{P}}_W$$

$$\mathbf{E} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{T} \\ \hline \mathbf{0} & 1 \end{array} \right]$$

E: Extrinsic Roto-translation matrix [4 × 4]



Complete perspective projection camera model

By combining all relationships, we obtain the complete perspective projection model (in homogeneous coordinates):

- Roto-translation:

$$\underbrace{\tilde{\mathbf{P}}_w \mapsto \tilde{\mathbf{P}}_{CAM}}_{\text{Roto-translation}} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{P}}_w = \mathbf{E} \cdot \tilde{\mathbf{P}}_w$$

- Perspective projection:

$$\underbrace{\tilde{\mathbf{P}}_{CAM} \mapsto \tilde{\mathbf{p}}_{IM}}_{\text{Perspective projection}} = \begin{bmatrix} x_{IMM} \\ y_{IMM} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & x_c & 0 \\ 0 & f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \tilde{\mathbf{P}}_{CAM} = \mathbf{K}_I \cdot \tilde{\mathbf{P}}_{CAM}$$

- Complete projective model:

$$\underbrace{\tilde{\mathbf{P}}_w \mapsto \tilde{\mathbf{p}}_{IM}}_{\text{Complete projective model}} = \begin{bmatrix} f & 0 & x_c & 0 \\ 0 & f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{P}}_w = \mathbf{K}_I \cdot \mathbf{E} \cdot \tilde{\mathbf{P}}_w = \mathbf{M} \cdot \tilde{\mathbf{P}}_w$$

**Intrinsic
calibration
matrix [3x4]**

**Extrinsic
rototranslation
matrix [4x4]**

**Projection Matrix
[3x4]**

Complete perspective projection camera model

$$\tilde{\mathbf{p}}_{IM} = \begin{bmatrix} f & 0 & x_c & 0 \\ 0 & f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \left[\begin{array}{c|c} \mathbf{R} & \mathbf{T} \\ \hline \mathbf{0} & 1 \end{array} \right] \cdot \tilde{\mathbf{P}}_W = \mathbf{M}(\xi) \cdot \tilde{\mathbf{P}}_W$$

- **Linear model in 11 parameters (\mathbf{M}_{3x4} , up to scale)**
 - **only 9 params are independent:**
$$\xi = [\mathbf{R}, \mathbf{T}, f, \mathbf{C}] = [\varphi, \vartheta, \rho, t_x, t_y, t_z, f, x_c, y_c]$$
- **Extrinsic Parameters:** depend on the relative position camera-scene
 - **Rotation:** Euler angles: $\mathbf{R} = [\varphi, \theta, \rho]$
 - **Translation:** translation vector: $\mathbf{T} = [t_x, t_y, t_z]$
- **Intrinsic Parameters:** depend on the camera characteristics
 - **Focal length:** f
 - **Optical Centre position:** $C = \langle x_c, y_c \rangle$

IMAGE FORMATION

Geometric Camera Models

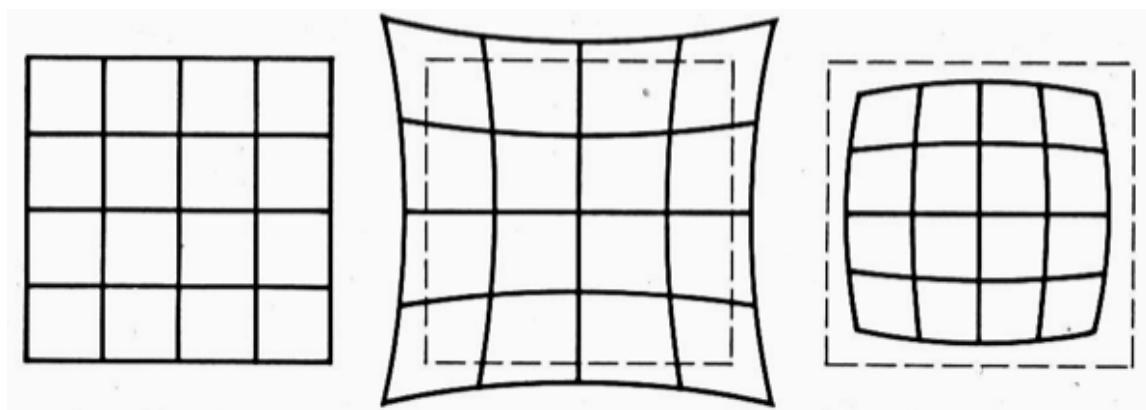
Real camera model

Chapter 1 – Forsyth Ponce

Radial distortion

Image magnification changes as the distance from the central point

- **effect: image deformation:** shift of the image point with respect to the linear geometric model



No distortion
 $k = 0$

"pincushion"
distortion
 $k < 0$

"barrel"
distortion
 $k > 0$

→ It has to be inserted into
the camera geometric model



Geometrical model with radial distortion

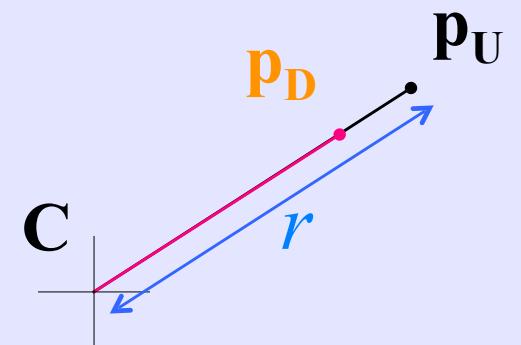
- Radial distortion:
→ varies along the ray length from the optical centre to the image point

Data:

$\mathbf{p}_U = (x_U, y_U)$: Undistorted position
(obtained by the linear model)

$\mathbf{p}_D = (x_D, y_D)$: Real position, Distorted

imagine



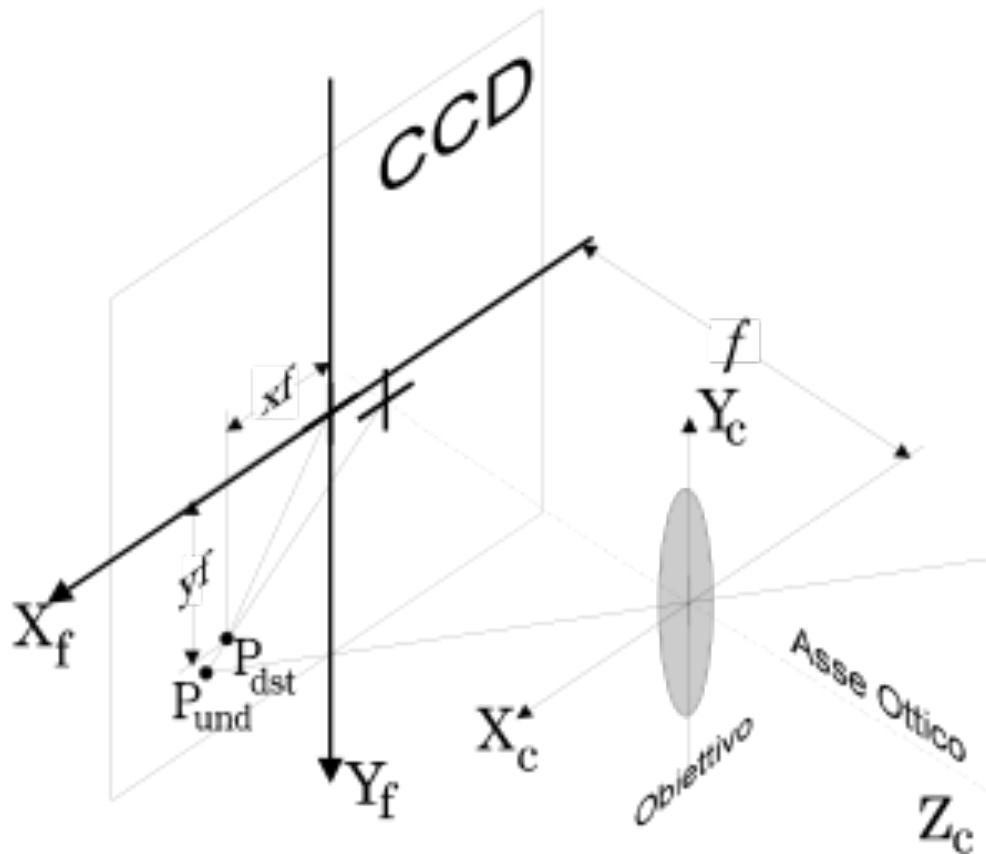
$$r^2 = |\mathbf{p}_U|^2 = x_U^2 + y_U^2$$

$$\mathbf{p}_D = \begin{bmatrix} x_D \\ y_D \end{bmatrix} = \mathbf{p}_U \left(1 + k_1 r^2 + k_2 r^4 + \dots \right) = \begin{bmatrix} x_U (1 + k_1 r^2 + k_2 r^4 + \dots) \\ y_U (1 + k_1 r^2 + k_2 r^4 + \dots) \end{bmatrix}$$

- \mathbf{p}_U is obtained with the linear model, then compute \mathbf{p}_D .

Radial distortion introduces **non-linearity** into the camera model!

Complete geometrical model with radial distortion



- ❖ **Roto-translation:**

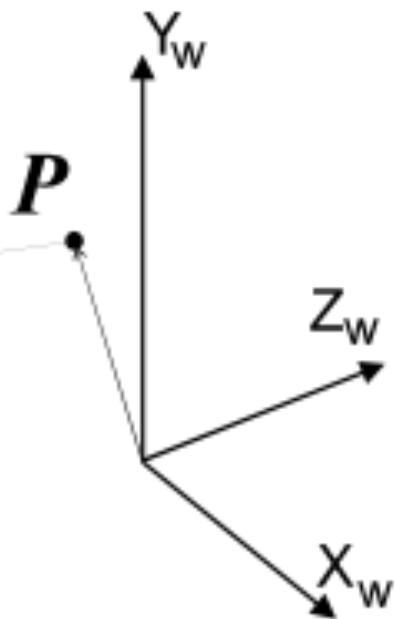
$$\tilde{\mathbf{P}}_w \mapsto \tilde{\mathbf{P}}_{CAM} = \mathbf{E} \cdot \tilde{\mathbf{P}}_w$$

- ❖ **Perspective Projection:**

$$\tilde{\mathbf{P}}_{CAM} \mapsto \tilde{\mathbf{p}}_{IM} = \mathbf{K}_I \cdot \tilde{\mathbf{P}}_{CAM}$$

- ❖ **Radial distortion correction**

$$\tilde{\mathbf{p}}_{IM} \mapsto \mathbf{p}_D = \mathbf{p}_U \left(1 + k_1 r^2 + k_2 r^4 + \dots \right)$$



Where are we?

First part:

Image formation and Early vision

- Image formation
 - Geometric Camera Models
 - **Color spaces**
- Image Processing
 - Punctual and spatial processing
 - Feature Extraction
- Reconstruction
 - Camera calibration
 - Stereo Vision
 - Structure from Motion and RGB-d Cameras
 - Optical flow and Tracking

Second part:

Machine learning for CV

- Linear Neural Network
- Multi Layer Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks
- Transformers
- Variational Auto-Encoders
- Generative Adversarial Networks
- Graph Neural Networks
- Self-supervised learning
- Vision Language Models

IMAGE ACQUISITION AND REPRESENTATION

Image acquisition

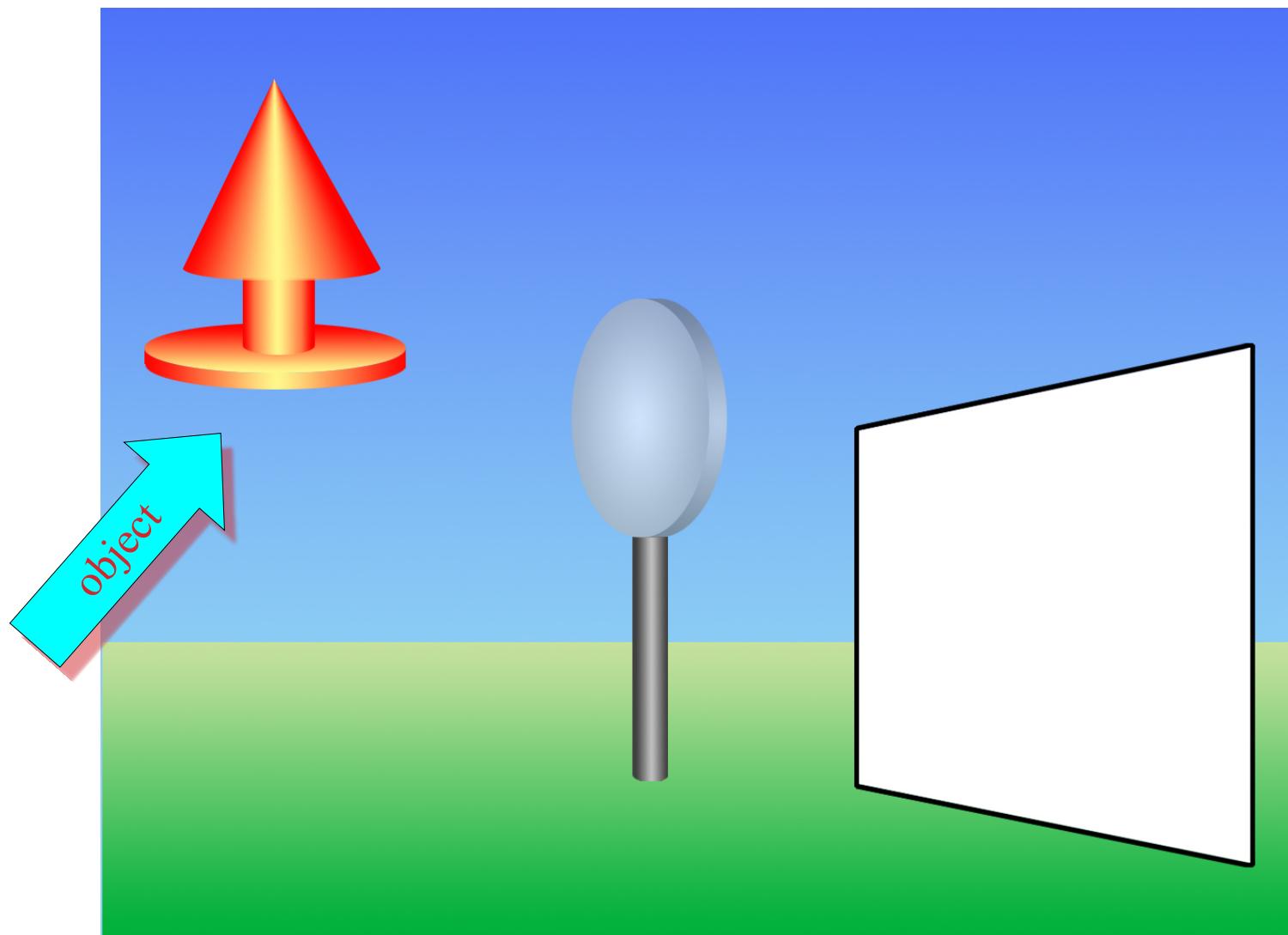


Image acquisition

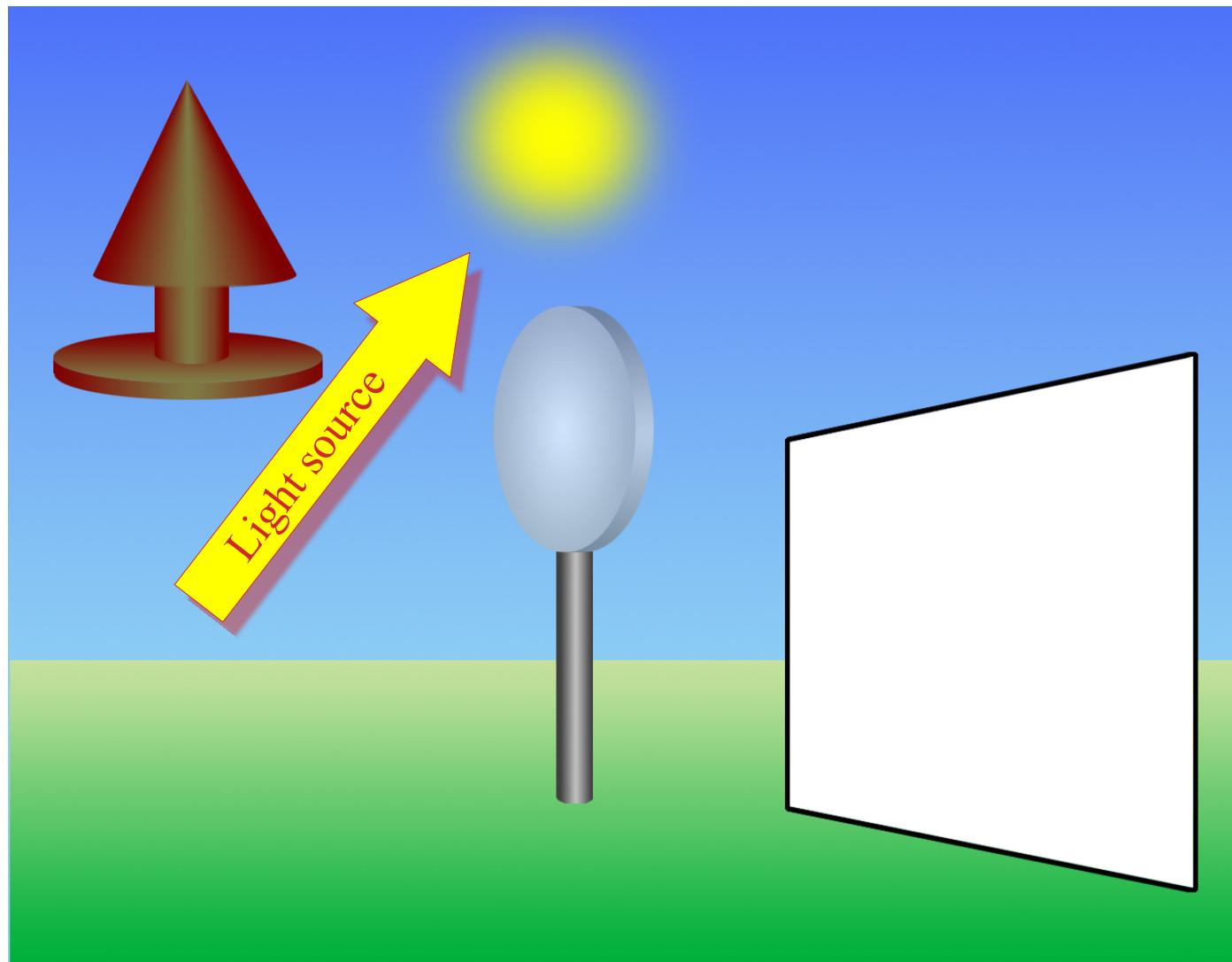


Image acquisition

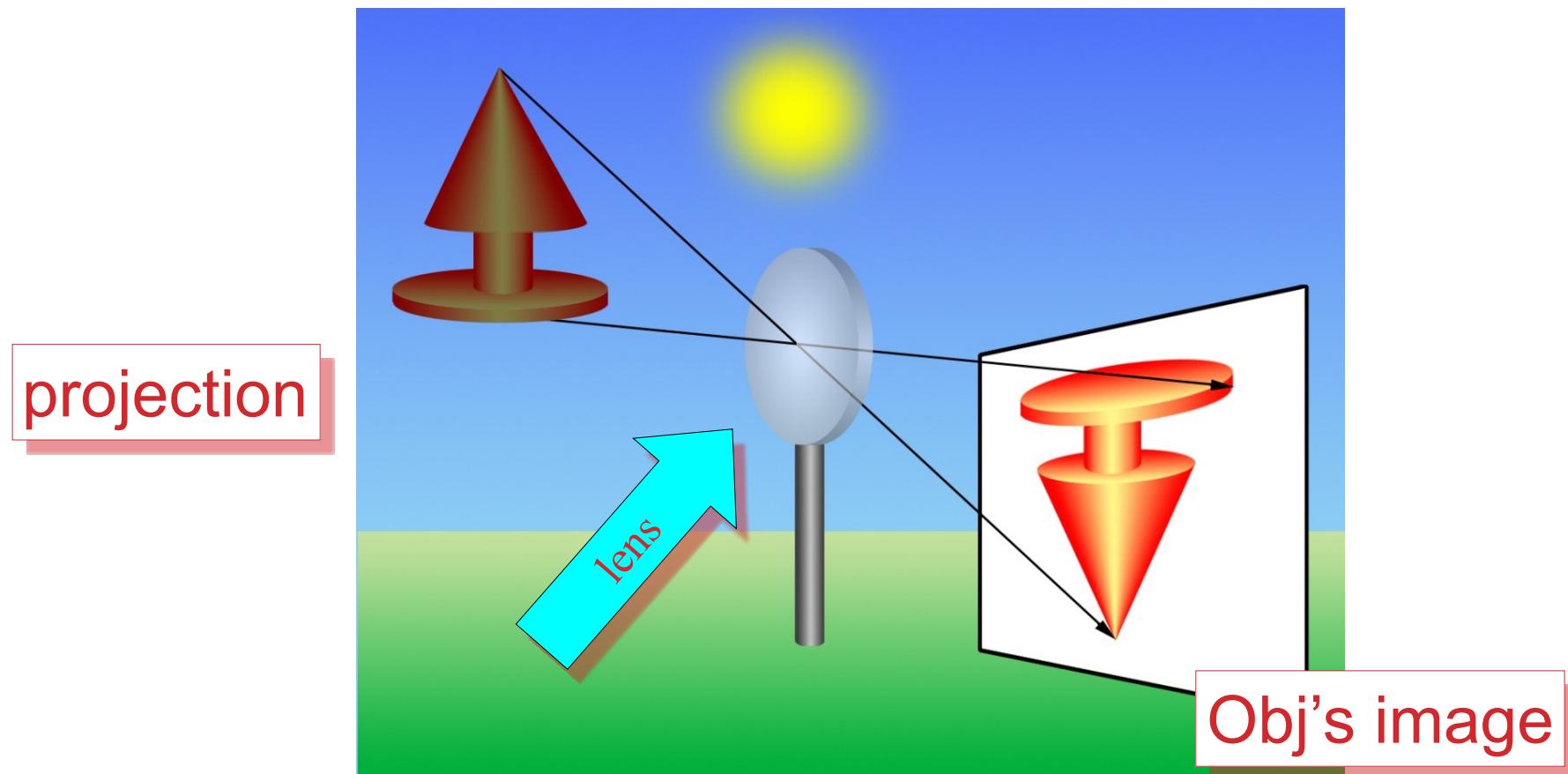


Image acquisition: sampling and quantization

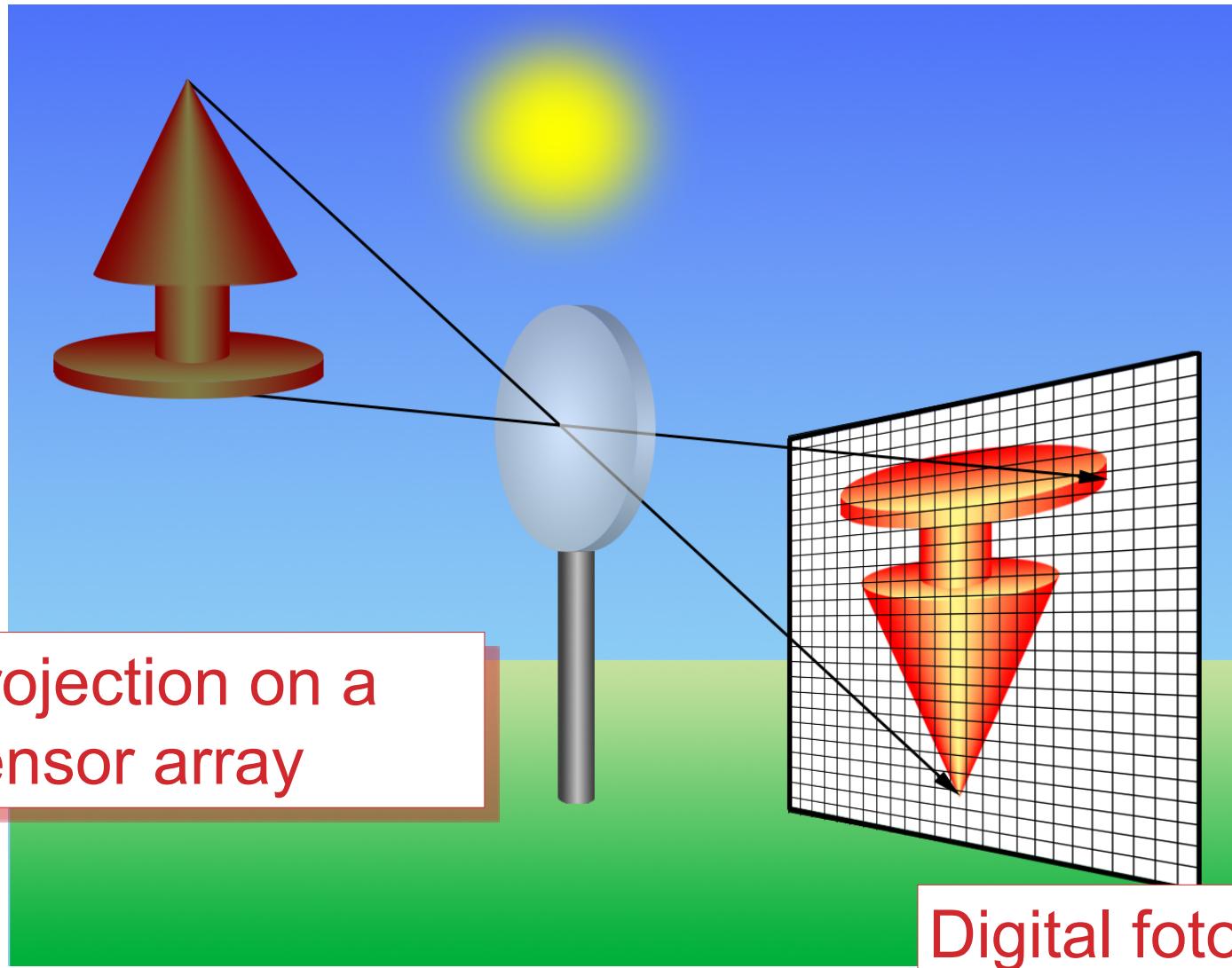


Image acquisition

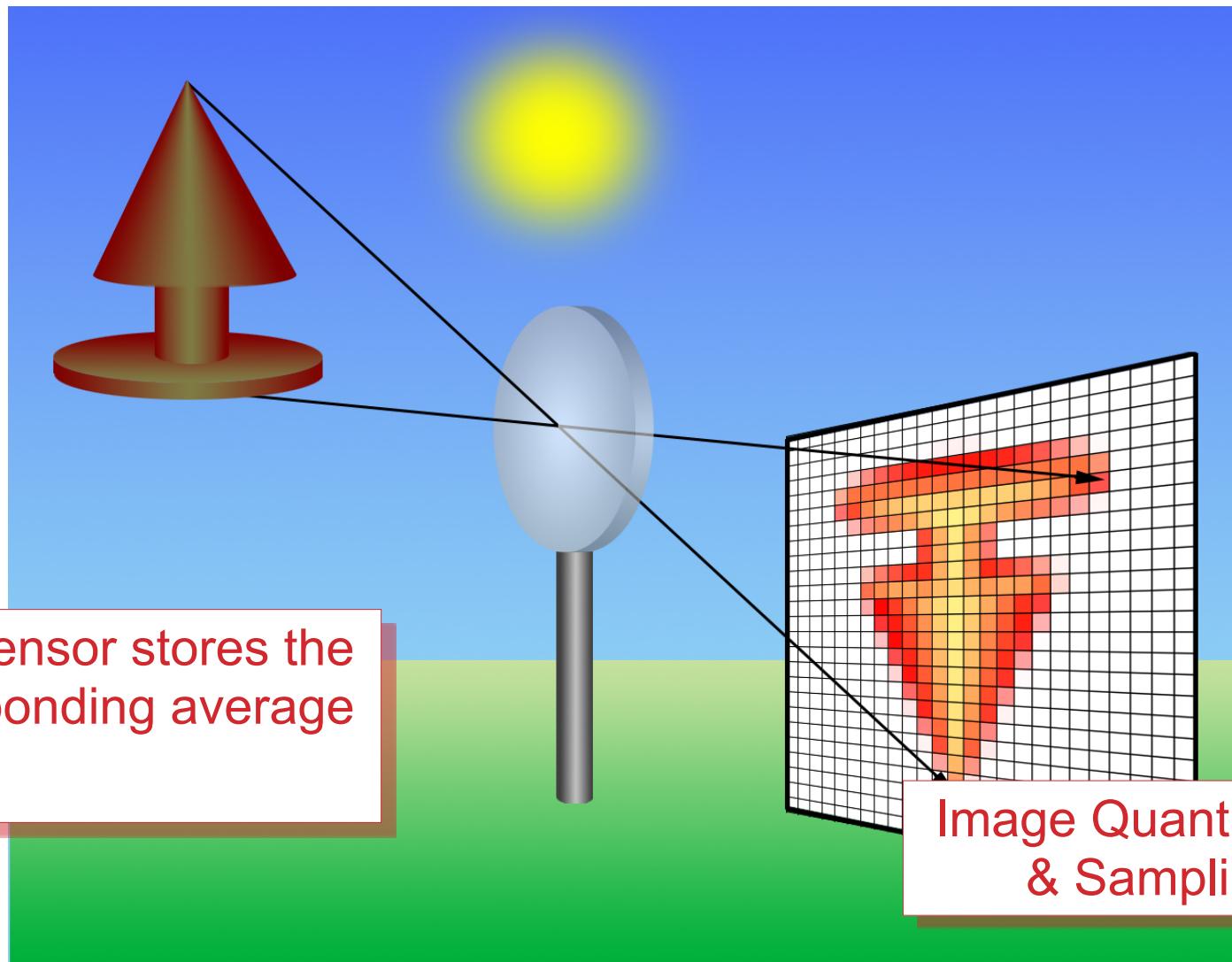


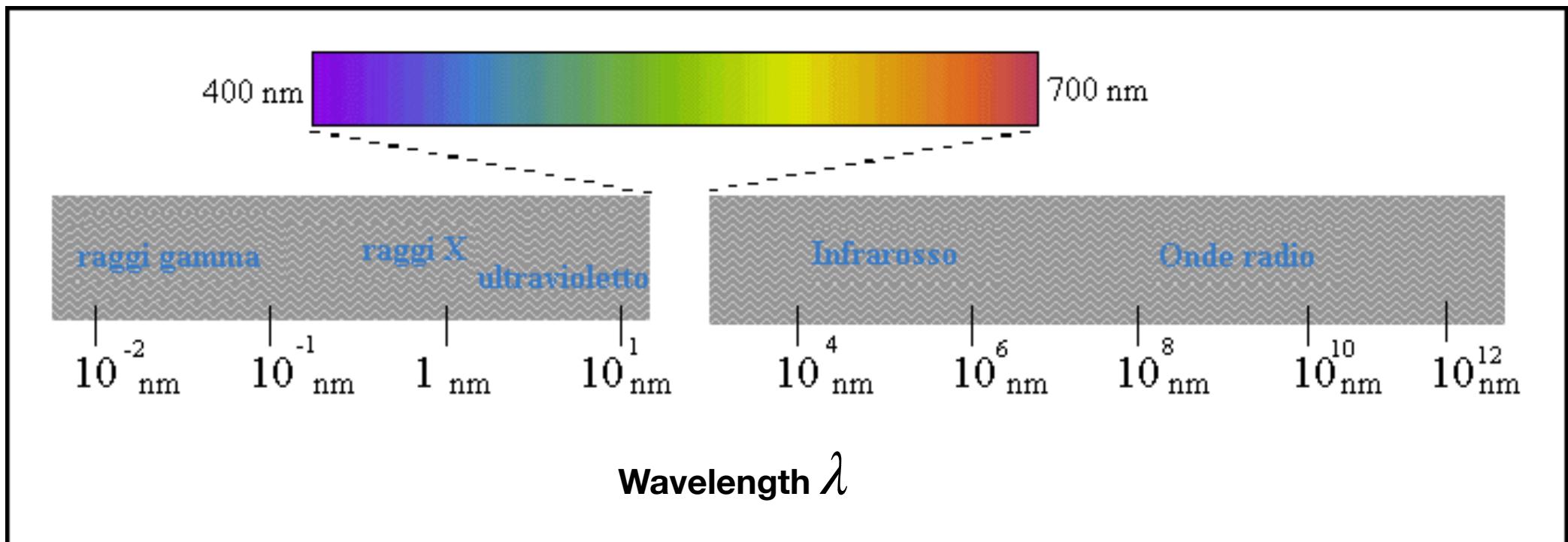
Image formation

- Light source (irradiance) is characterized by its 3d position (x, y, z) and its wavelengths λ

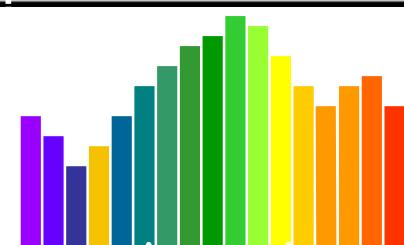
$$E(x, y, z, \lambda)$$



Light source: Electromagnetic Spectrum



- Each light source has its **Spectrum**: range and power of wavelengths of electromagnetic radiation that it emits [W]



- **Irradiance**: amount of electromagnetic radiation that is received by a surface per unit area per unit time [W/m^2]

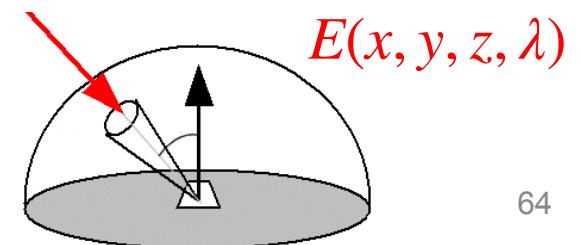


Image formation

- Light source (irradiance) is characterized by its 3d position (x, y, z) and its wavelengths λ
- Each point in the scene has a reflectivity function

$$r(x, y, z, \lambda)$$



Reflectance spectra

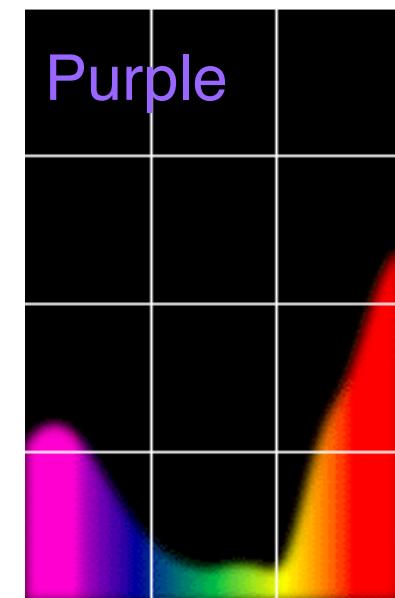
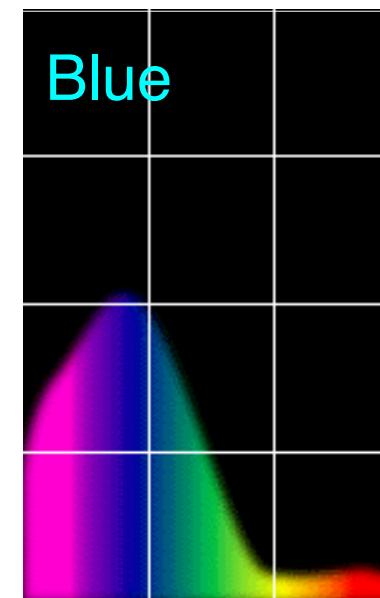
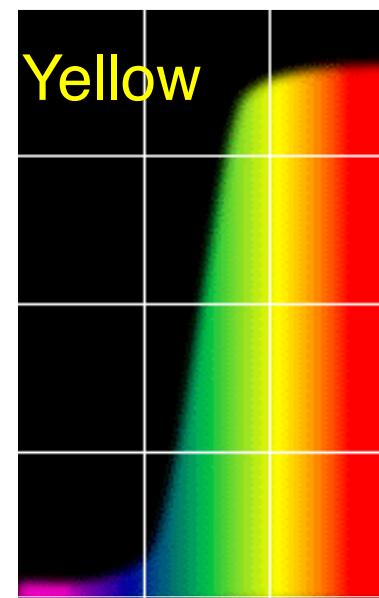
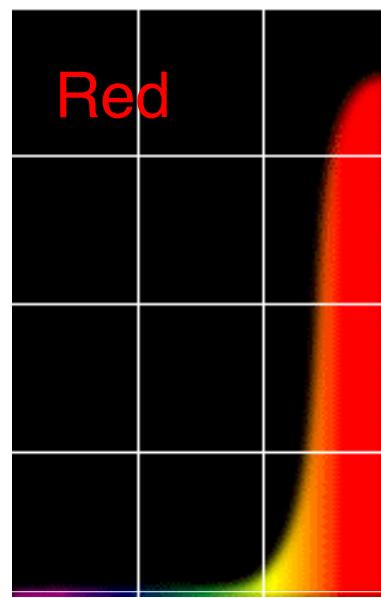


Image formation

- Light source (irradiance) is characterized by its 3d position (x, y, z) and its wavelengths λ : $E(x, y, z, \lambda)$
- Each point in the scene has a reflectivity function $r(x, y, z, \lambda)$
- These quantities are generally combined in a multiplicative way, producing a reflected light $c(x, y, z, \lambda)$
 $c(x, y, z, \lambda)$ is the input to the camera!



Blender 2.80.0 (2019-09-25)

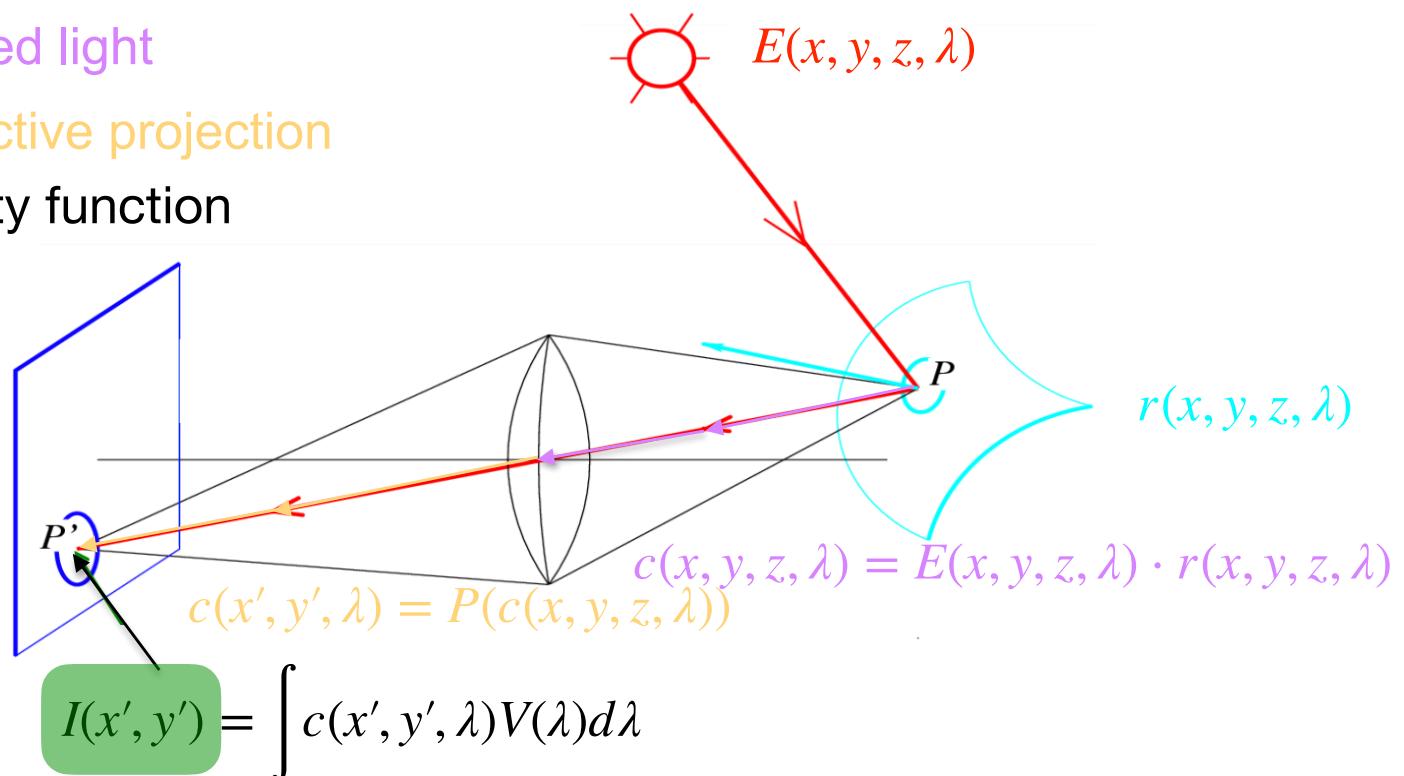
Blender 2.80.0 (2019-09-25)

$$\rightarrow c(x, y, z, \lambda) = E(x, y, z, \lambda) \cdot r(x, y, z, \lambda)$$

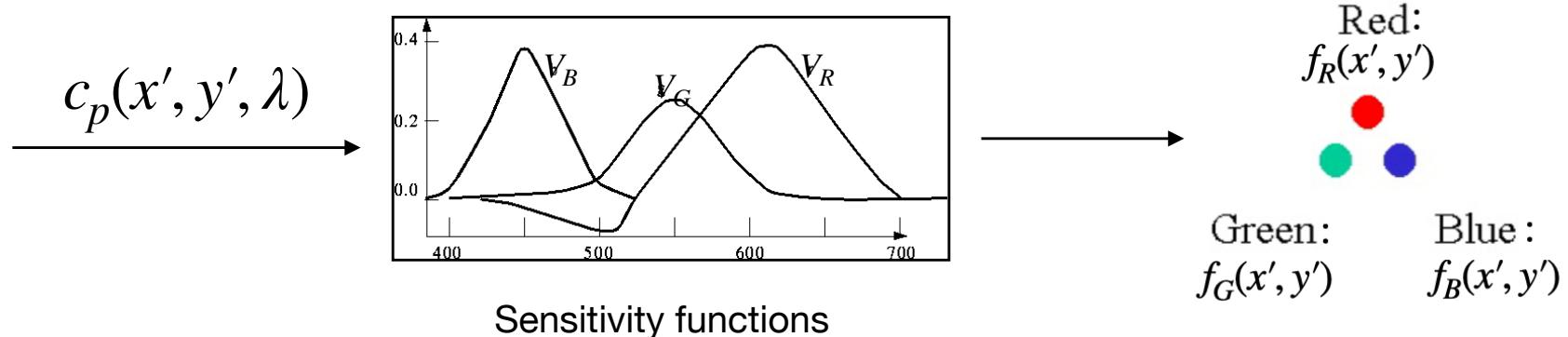
Camera($c(x, y, z, \lambda)$)

Image formation

- What determines the pixel value $I(x', y')$?
 - Light source properties
 - Surface properties
 - Reflected light
 - Perspective projection
 - Sensitivity function



Sensitivity functions



$$f_R(x', y') = \int c_p(x', y', \lambda) \cdot V_R(\lambda) d\lambda$$

$$f_G(x', y') = \int c_p(x', y', \lambda) \cdot V_G(\lambda) d\lambda$$

$$f_B(x', y') = \int c_p(x', y', \lambda) \cdot V_B(\lambda) d\lambda$$

- **Spectral sensitivity** refers to a sensor's response to different wavelengths of light.
- It is described by a **sensitivity function**, which quantifies how much a sensor detects light at each wavelength.

Image formation: summing up

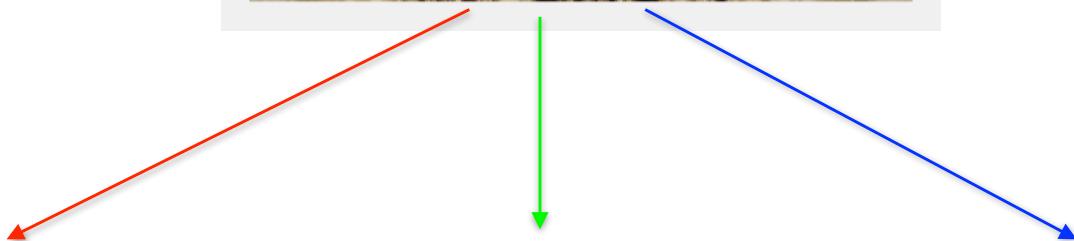
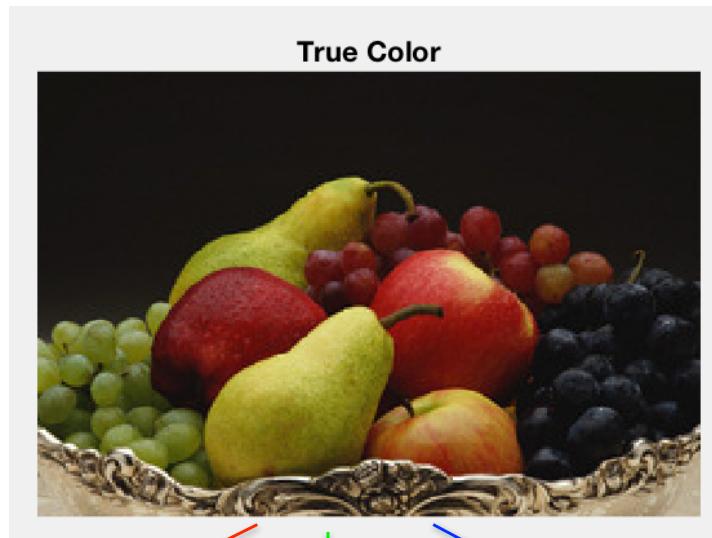
- The image function $f_C(x', y')$ ($C = R, G, B$) is formed as:

$$f_C(x', y') = \int c_p(x', y', \lambda) V_C(\lambda) d\lambda \quad (2)$$

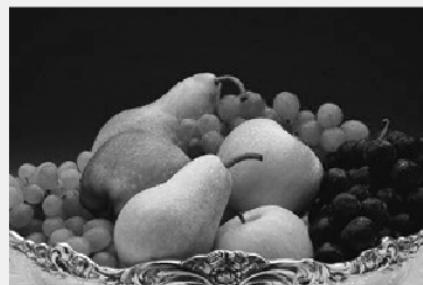
- It is the result of:

1. Incident light $E(x, y, z, \lambda)$ at the point (x, y, z) in the scene,
2. The reflectivity function $r(x, y, z, \lambda)$ of this point,
3. The formation of the reflected light $c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda)$,
4. The projection of the reflected light $c(x, y, z, \lambda)$ from the *three dimensional* world coordinates to *two dimensional* camera coordinates which forms $c_p(x', y', \lambda)$,
5. The sensitivity function(s) of the camera $V_C(\lambda)$.

A true color image



Red



Green



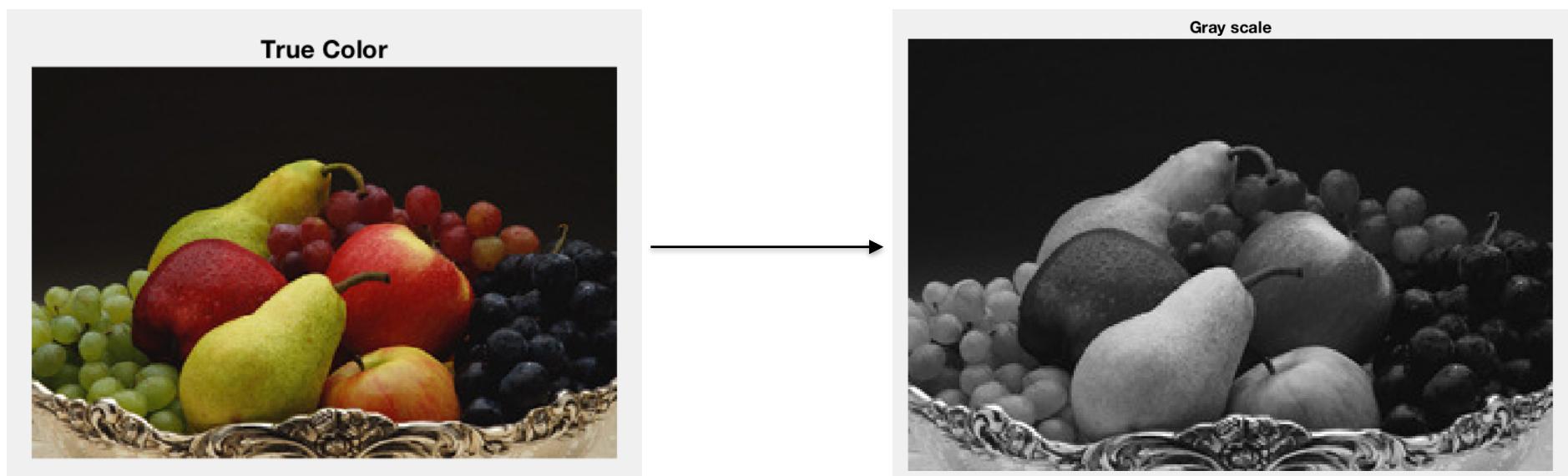
Blue



Luminance image

- For each pixel we have the corresponding amount of light (luminance):

$$Y = 0.299R + 0.587G + 0.114B$$



Other Color spaces: HSV

HSV:

- ❖ **H**: Hue
- ❖ **S**: Saturation
- ❖ **V**: Value or Intensity

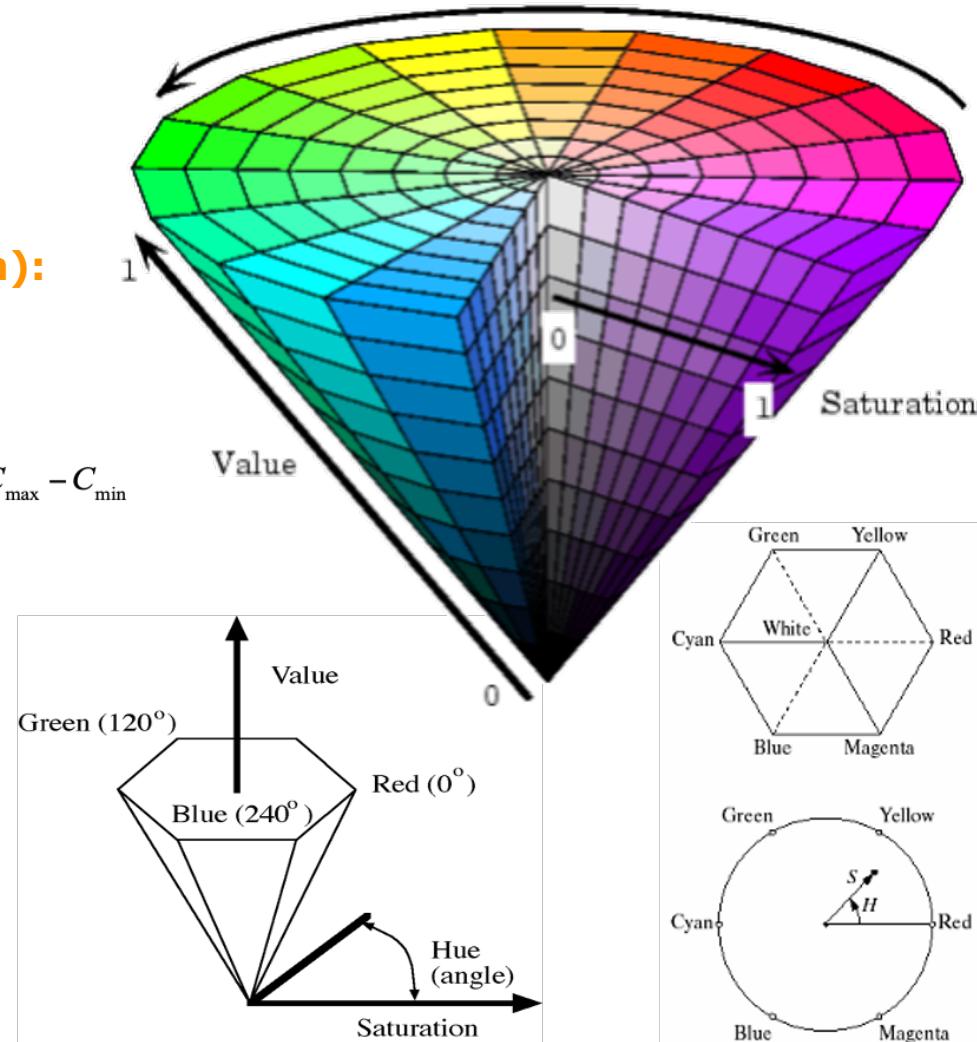
RGB → HSV (nonlinear conversion):

$$R' = \frac{R}{255}, \quad G' = \frac{G}{255}, \quad B' = \frac{B}{255}$$

$$C_{\max} = \max(R', G', B'), \quad C_{\min} = \min(R', G', B'), \quad \Delta = C_{\max} - C_{\min}$$

$$H = \begin{cases} 0^\circ, & \Delta = 0 \\ 60^\circ \cdot \left(\frac{G' - B'}{\Delta} \bmod 6 \right), & C_{\max} = R' \\ 60^\circ \cdot \left(\frac{B' - R'}{\Delta} + 2 \right), & C_{\max} = G' \\ 60^\circ \cdot \left(\frac{R' - G'}{\Delta} + 4 \right), & C_{\max} = B' \end{cases}$$

$$S = \begin{cases} 0, & C_{\max} = 0 \\ \frac{\Delta}{C_{\max}}, & C_{\max} \neq 0 \end{cases}; \quad V = C_{\max}$$



Other Color spaces: CIE-Lab

CIE L*a*b* (1976)

- **perceptually uniform** space: a given numerical change corresponds to a similar perceived change in color
- L^* : perceptual lightness
- a^* , b^* : spread over the 4 unique colors of human vision (red, green, blue, yellow)
- Non linear conversion s.t. Koenderink:
“an awful mix of magical numbers and arbitrary functions that somehow ‘fit’ the eye measure”

