# Camera Models and Image Formation

Giacomo Boracchi

giacomo.boracchi@polimi.it

February 21st, 2020

USI, Lugano

# What we will see today

- The Pinhole Camera Model
- Camera and Lenses
- Homogeneous Coordinate System
- The Camera Matrix
- Camera Calibration
- An overview on *Photometric* Image Formation (noise and demoisaicing)

Reference HZ, Chapter 6

Watch out:
These simple gemetrical formulas underpin the whole single and multi-view geometry we will see

… and a bit of photography notions that does not hurt

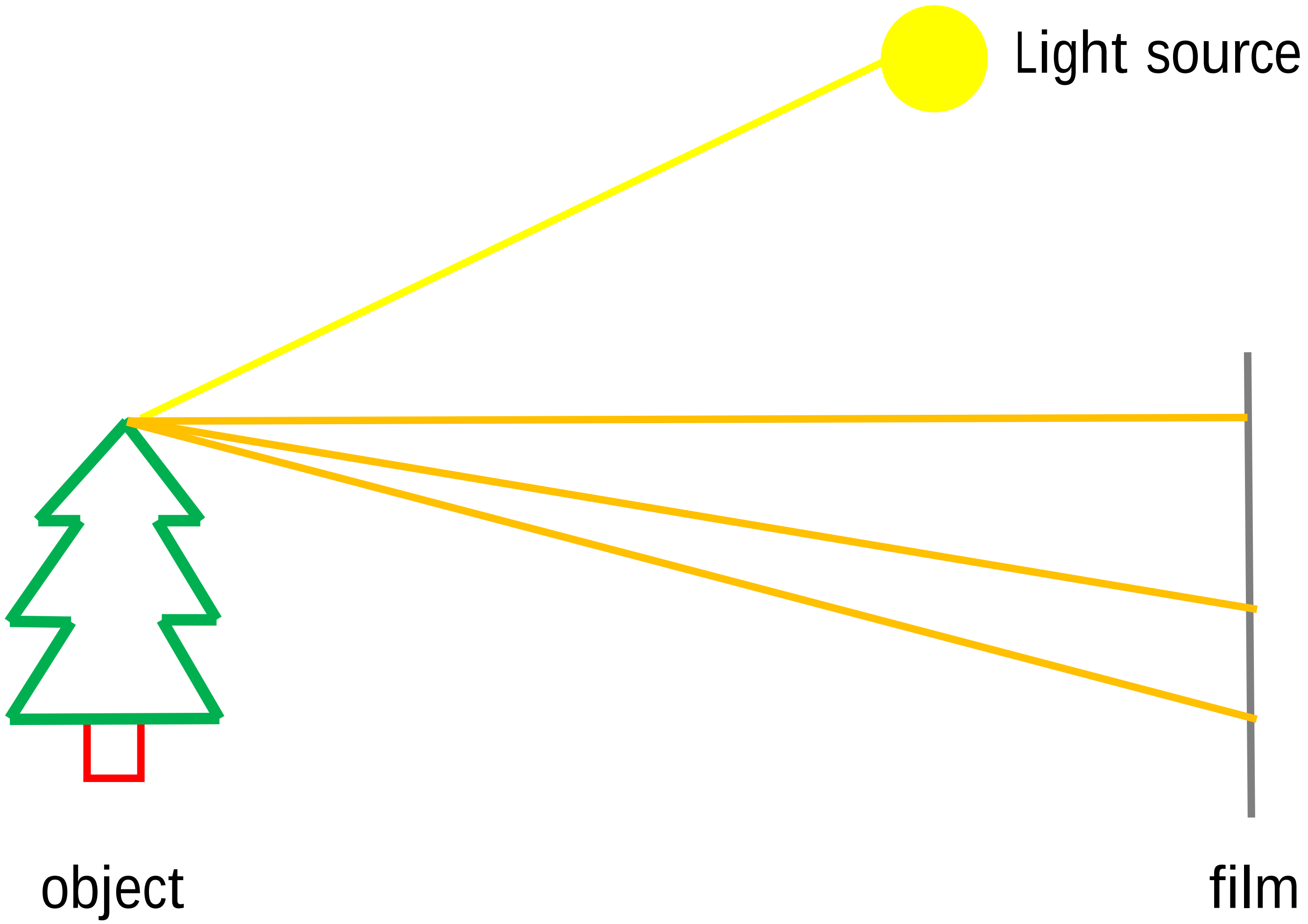
# Pinhole Camera

# Cameras

Let's use a film to acquire a picture

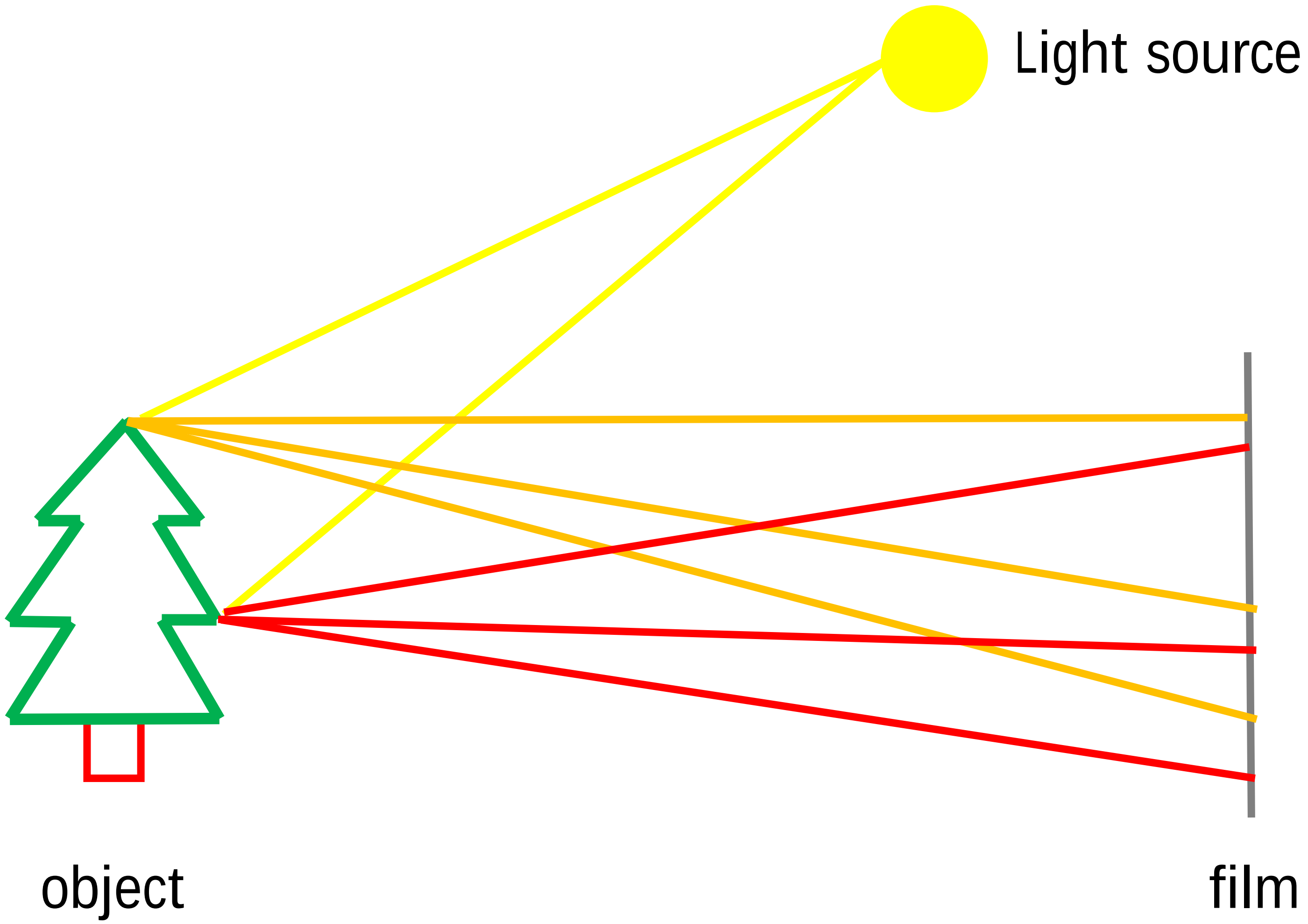
# Cameras

The film in the camera gathers the light reflected by objects



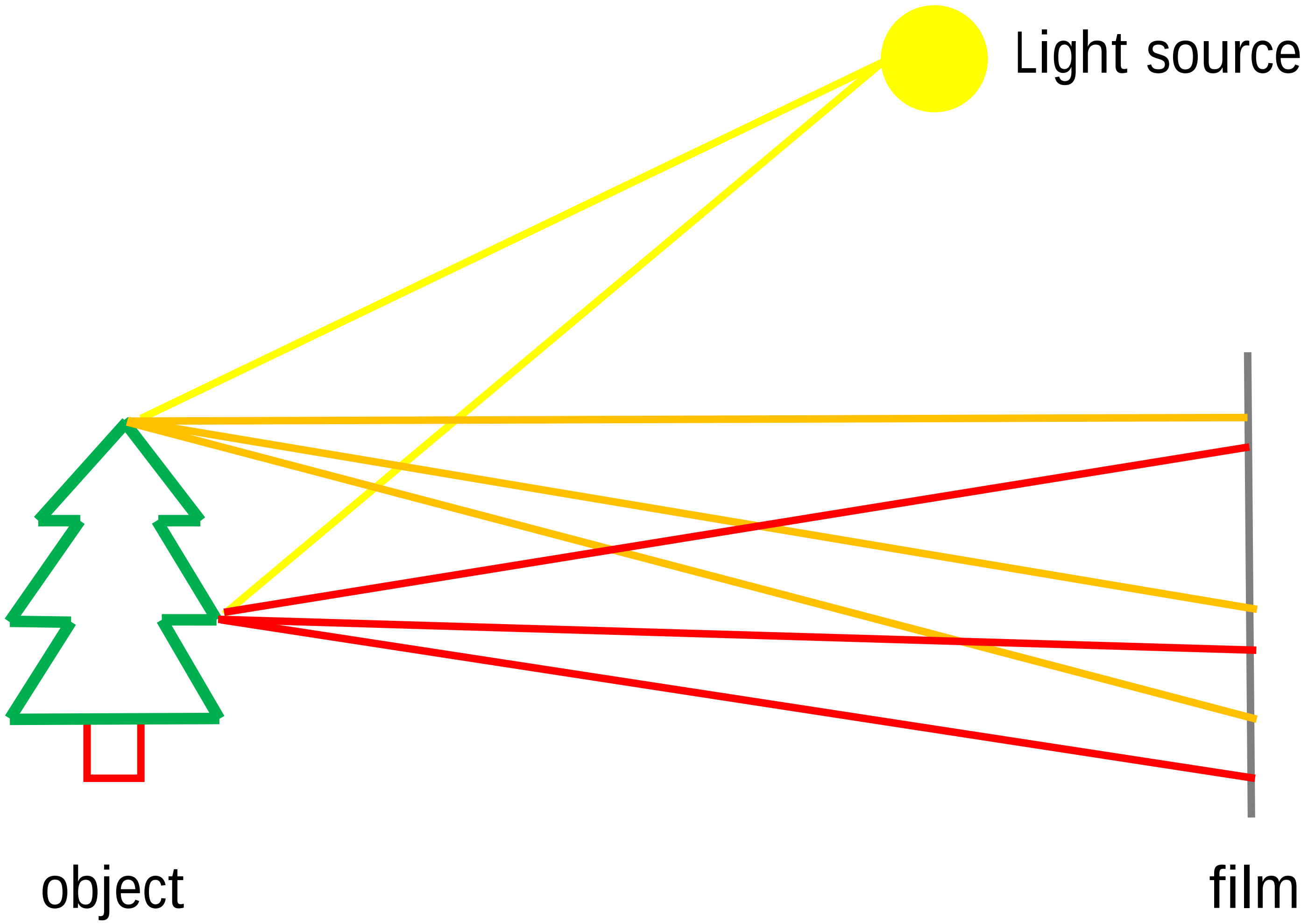
# Cameras

The film in the camera gathers the light reflected by objects



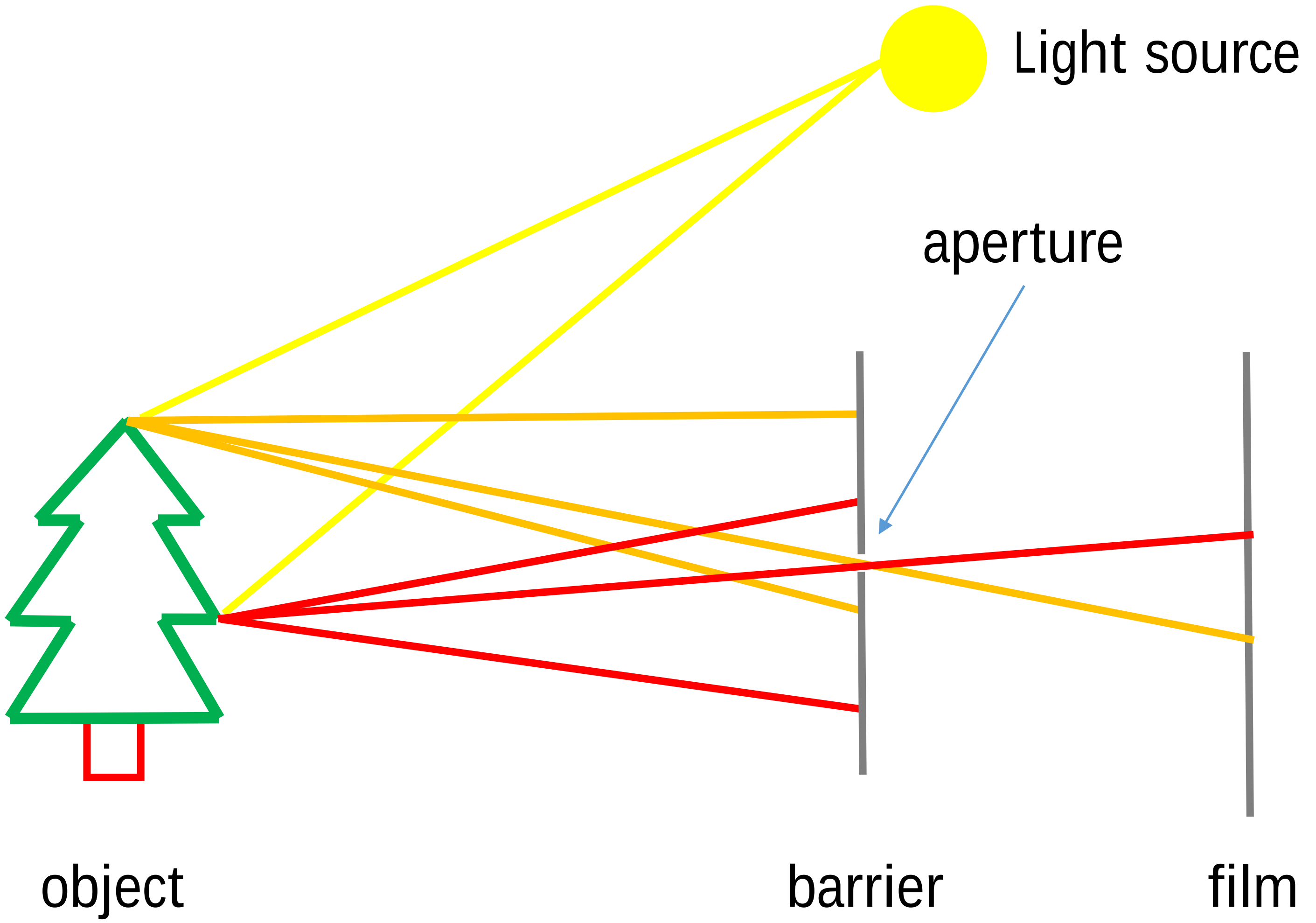
# Cameras

This wouldn’t produce an image



# Pinhole Camera

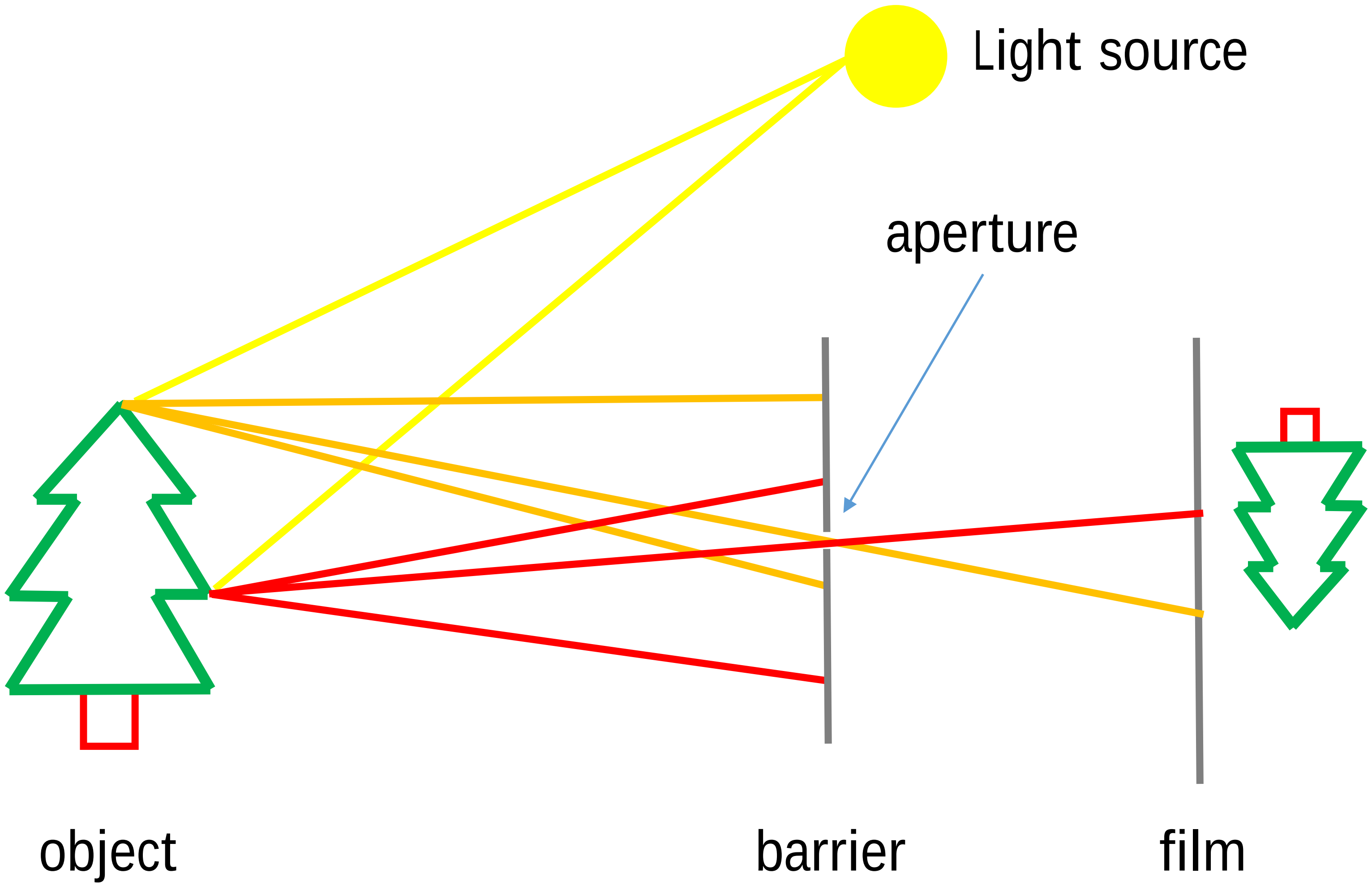
Let's add a barrier in front of the film




Giacomo Boracchi

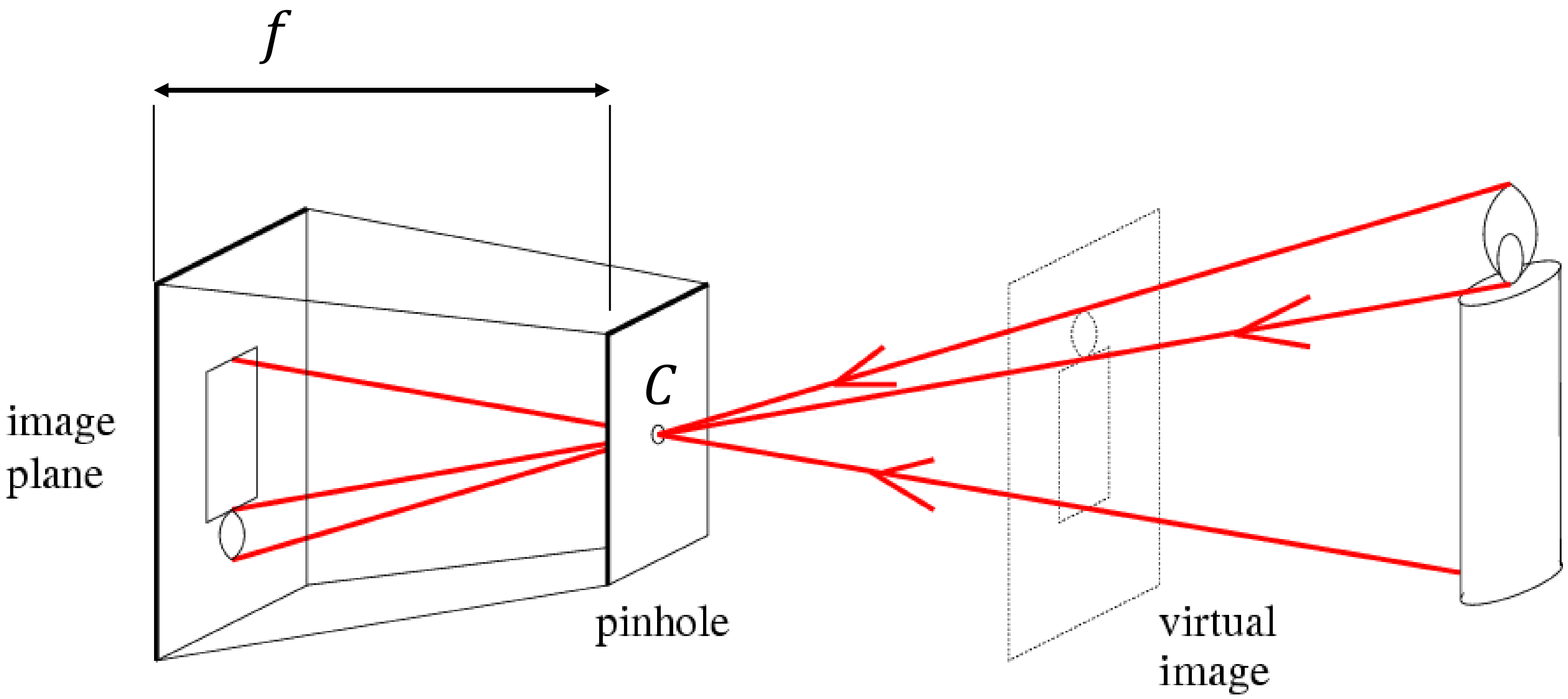
# Pinhole Camera

Let's add a barrier in front of the film



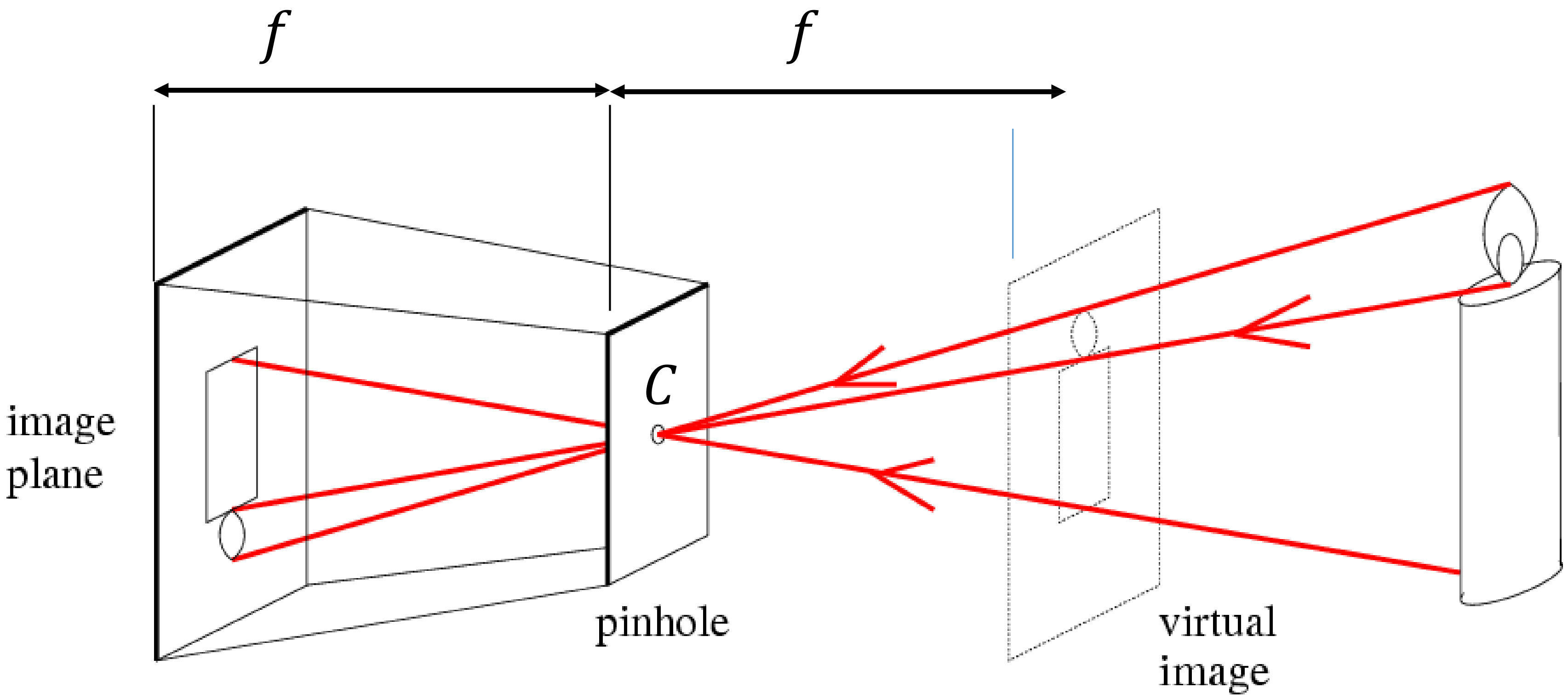

Giacomo Boracchi

# The Pinhole Camera



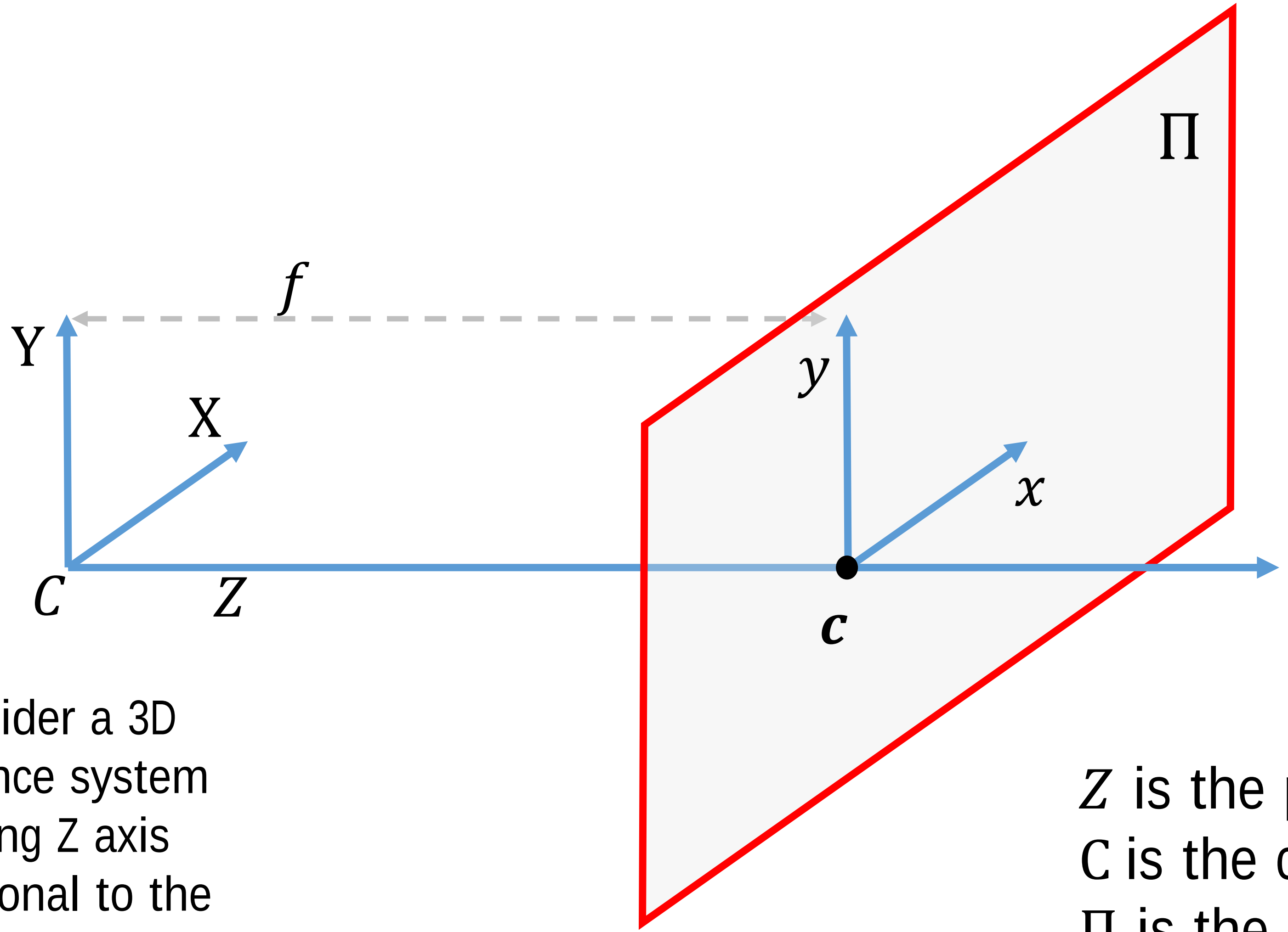

$f$ = focal length
$O$ = aperture = pinhole = center of the camera

# The Pinhole Camera




Projection over the virtual plane is identical to the projection over the image plane up to a scale and vertical flip transformation

# The Setup of a Pinhole Camera



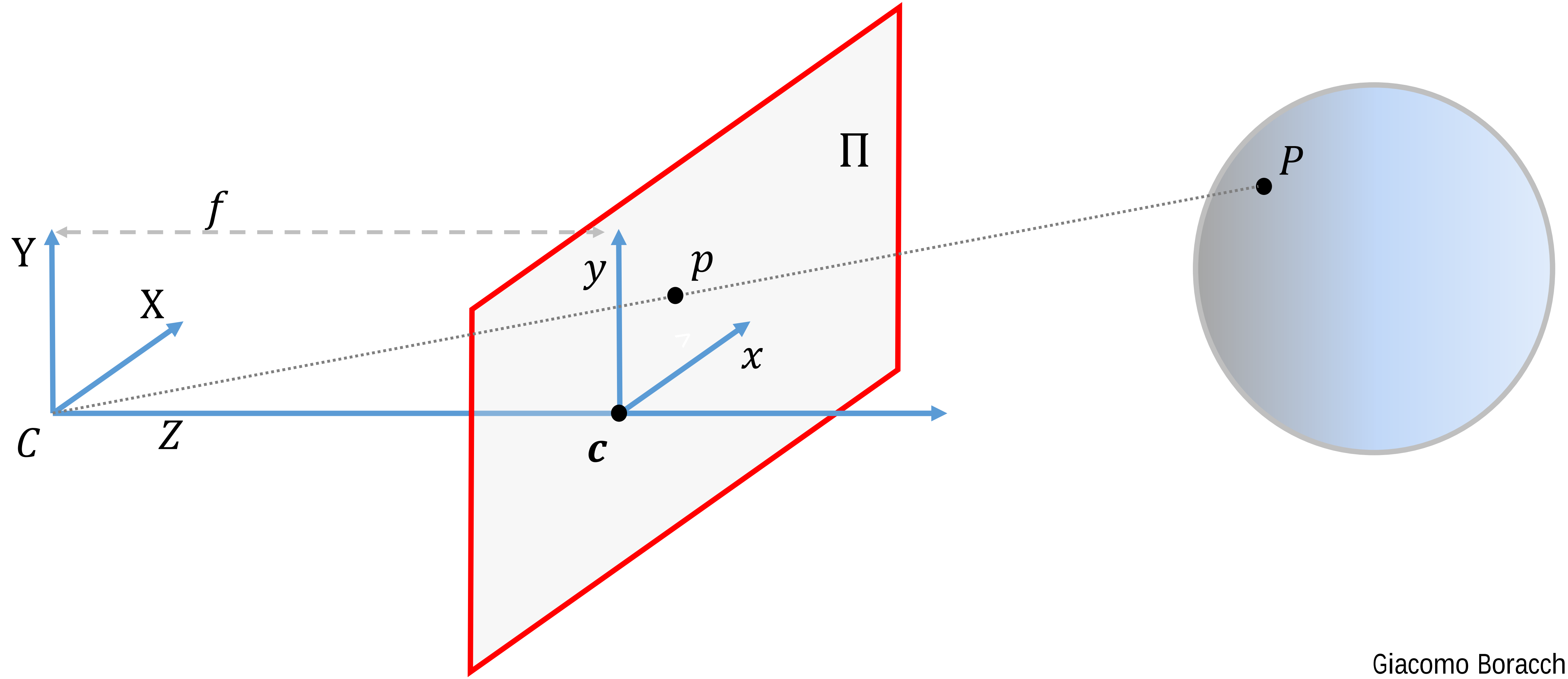

Consider a 3D reference system having Z axis orthogonal to the image plane

$Z$ is the principal (or optical) axis
C is the camera center
Π is the image (or focal) plane (the film)
$\boldsymbol{c}$ is the principal point ($\Pi \cap Z$ axis)
$x, y$ coordinate system in the image plane Π

# Camera Projection

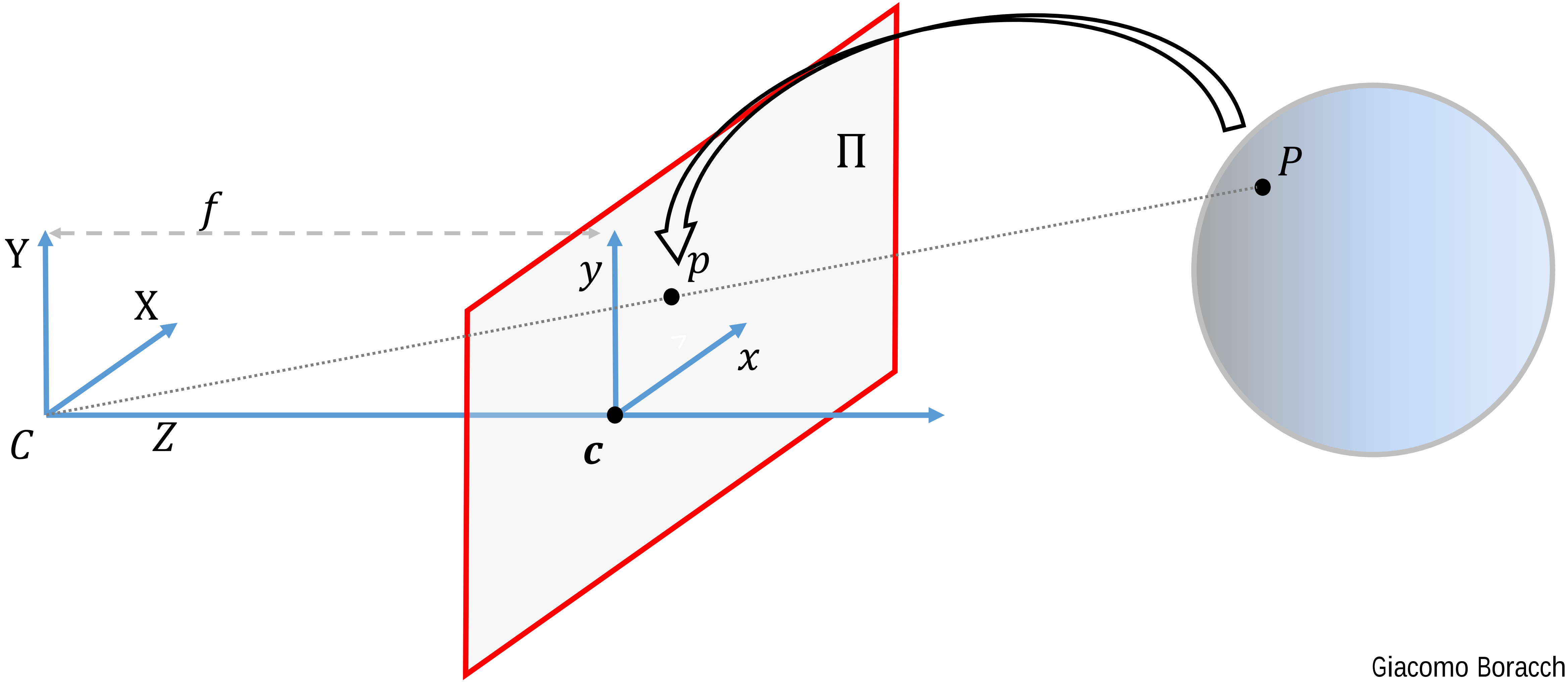
# Camera Projection

Camera Projection: the mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ that associates a 3D point to an point on the image plane

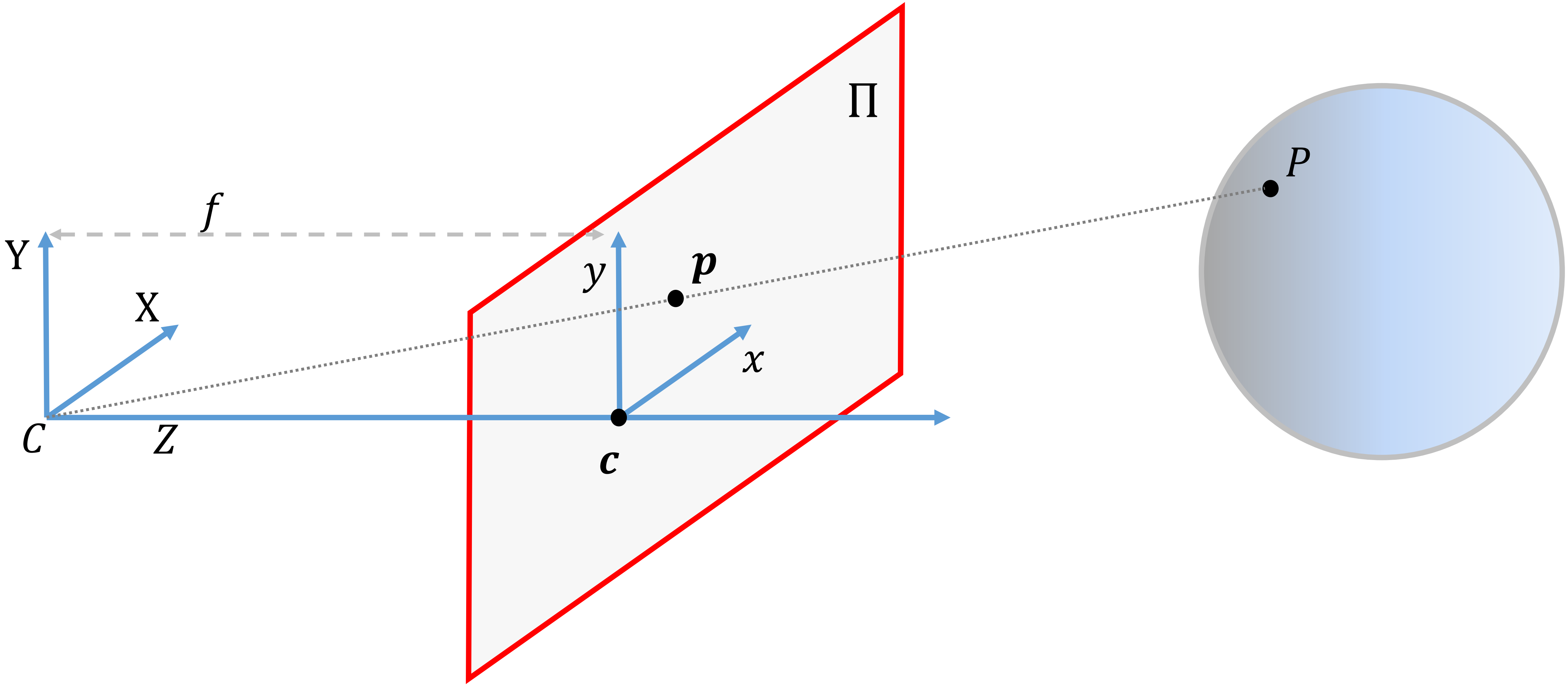


# Camera Projection

Camera Projection: the mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ that associates a 3D point to an point on the image plane

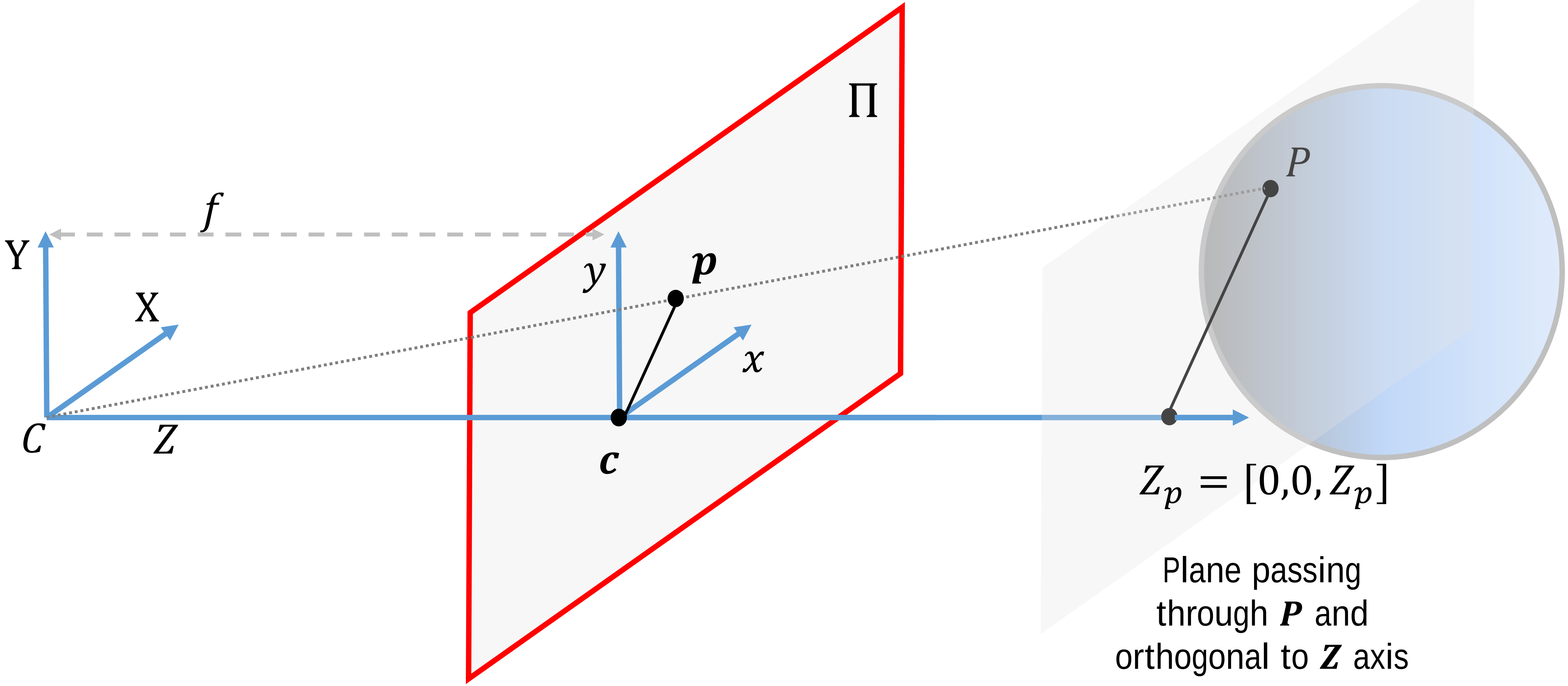


# How to Project Points form 3D world to $\Pi$?




Consider the virtual plane at distance $f$ from the camera center
What is the relation between $P = [X_p, Y_p, Z_p]$ and $\boldsymbol{p} = [x_p, y_p]$?

# How to Project Points form 3D world to $\Pi$?

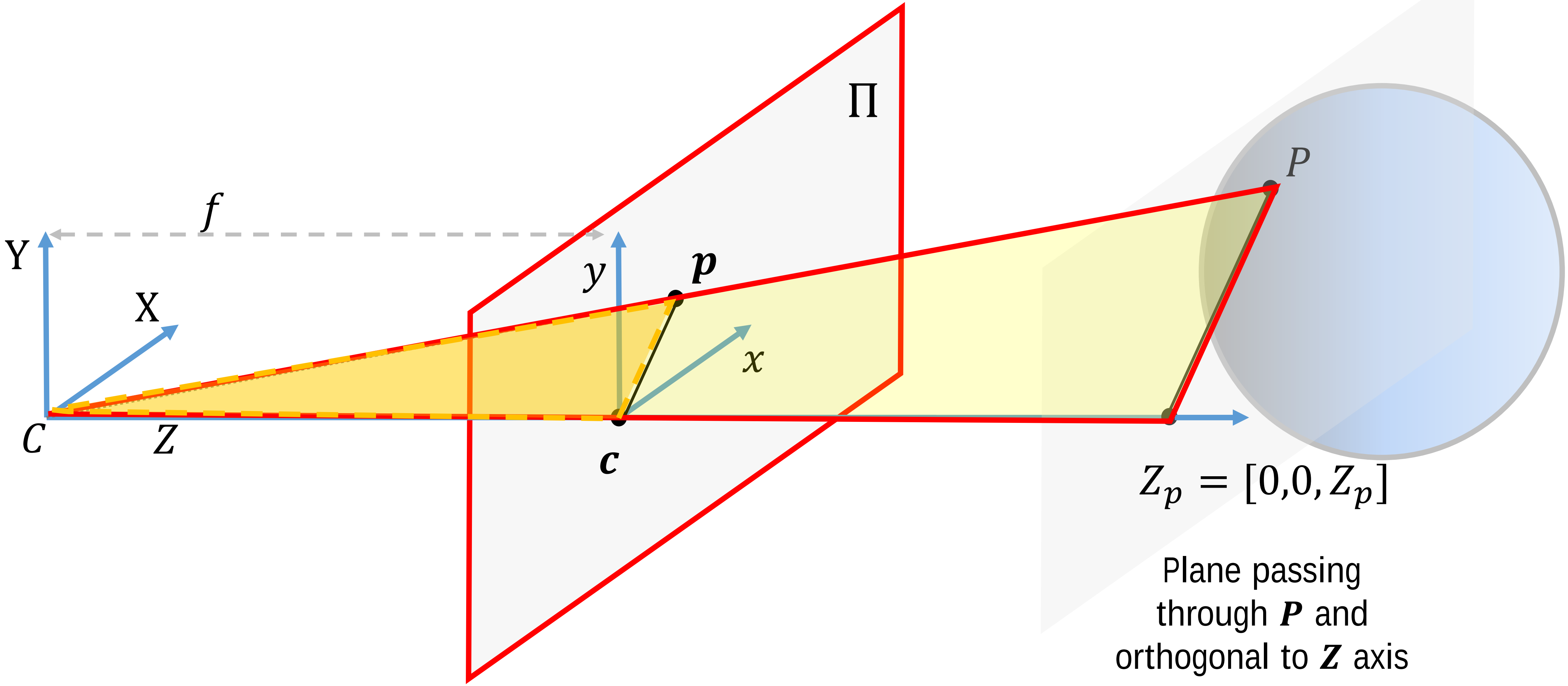



What is the relation between $P = [X_p, Y_p, Z_p]$ and $\boldsymbol{p} = [x_p, y_p]$?

The triangles $\boldsymbol{Ccp}$ and $\boldsymbol{CZ_pP}$ are similar!

And the same holds for $C\boldsymbol{c}y_p$ and $CZ_pY_p$ and for $C\boldsymbol{c}x_p$ and $CZ_pX_p$

# How to Project Points form 3D world to $\Pi$?

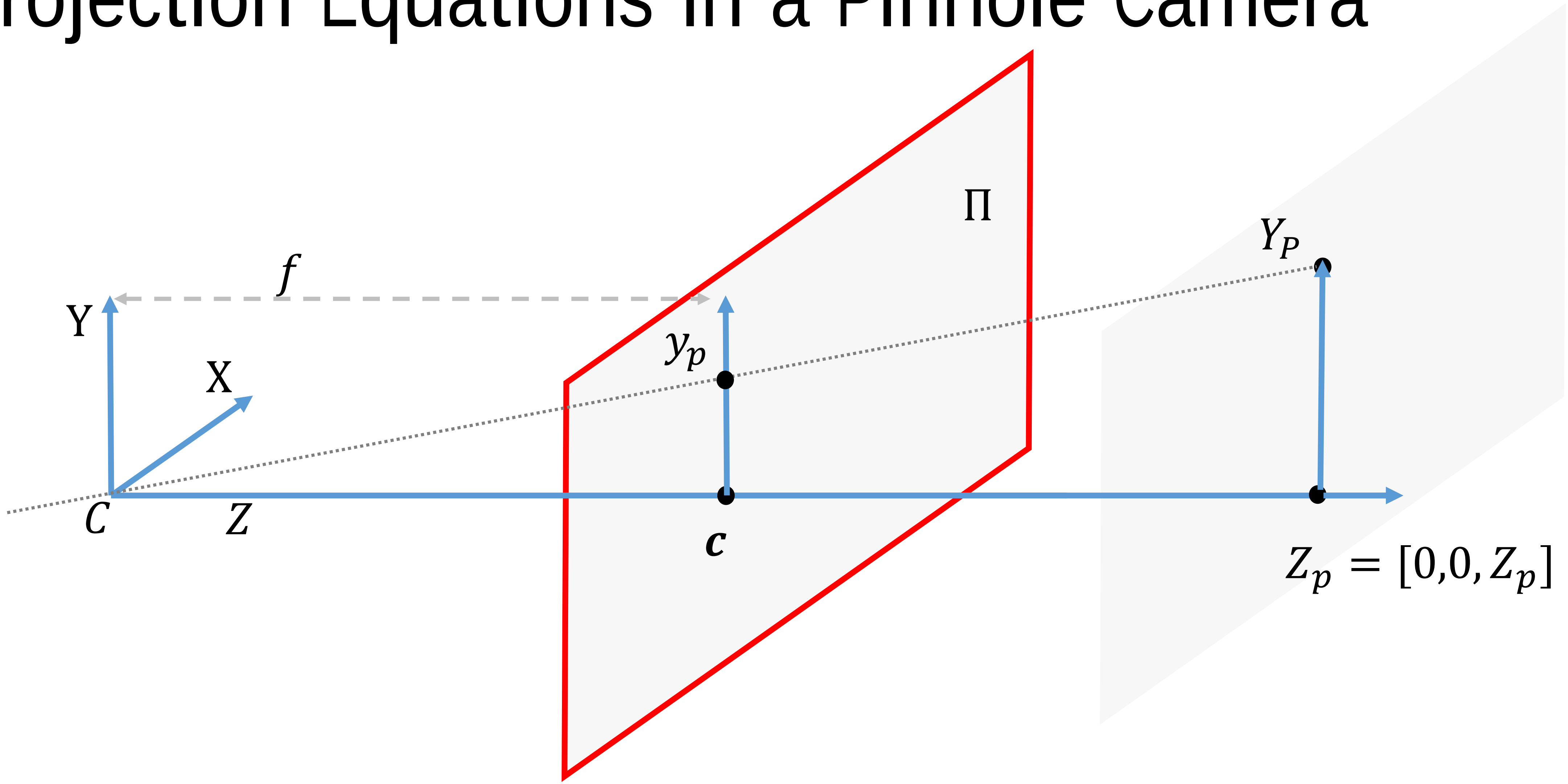



What is the relation between $P = [X_p, Y_p, Z_p]$ and $\boldsymbol{p} = [x_p, y_p]$?

The triangles $\boldsymbol{Ccp}$ and $\boldsymbol{CZ_pP}$ are similar!

And the same holds for $C\boldsymbol{c}y_p$ and $CZ_pY_p$ and for $C\boldsymbol{c}x_p$ and $CZ_pX_p$

# Projection Equations in a Pinhole Camera




from triangle similarity follows $Cc\colon Cy_p = OZ_P\colon Z_PY_P$,

Thus $f\colon y_p = Z_P\colon Y_P$ (similar relation for $x_p$)

# Projection Equations in a Pihole Camera

The equations of Pihole Camera Projection

$$P = [X_P, Y_P, Z_P] \rightarrow \boldsymbol{p} = [x_p, y_p]$$

where

$$\begin{cases} x_p = f * X_P / Z_P \\ y_p = f * Y_P / Z_P \end{cases}$$

# Perspective

$$\begin{cases} x_p = f * X_P / Z_P \\ y_p = f * Y_P / Z_P \end{cases}$$

Division by $Z_P$ is responsible of the perspective vision

- Coordinates of far points (large $Z_P$) are shrinked more than nearby ones (small $Z_P$)

This is perspective: the sizes of objects in the image actually depend on their depths in the scene (distance from the camera center).

# Perspective

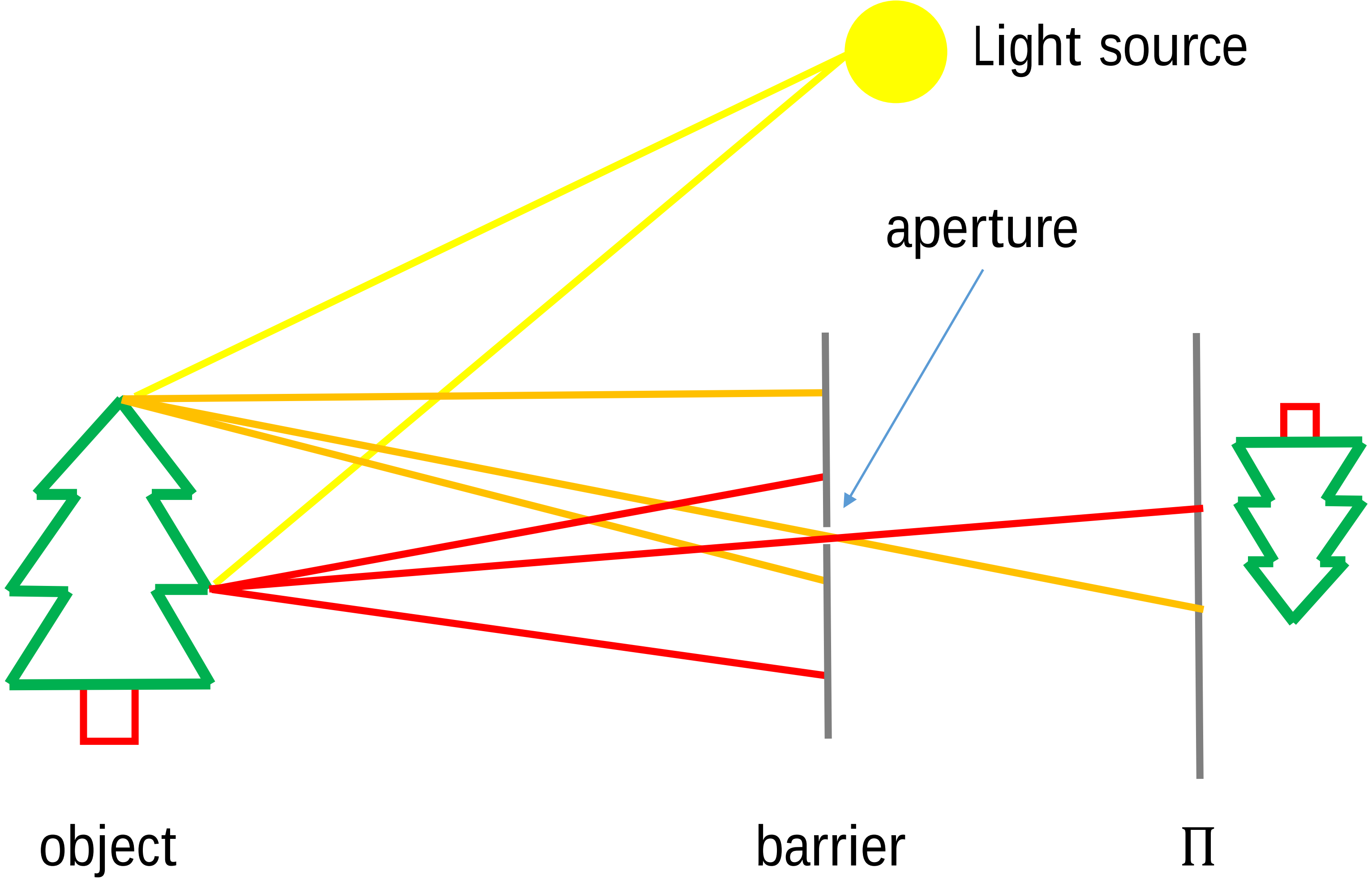
For example, all the train windows are expected to have the same value of the $Y_P$ coordinate.

However, their $y_p$ coordinate in the image plane depend on the actual depth of the 3D points $Z_P$.

This results in the highlighted perspective effects

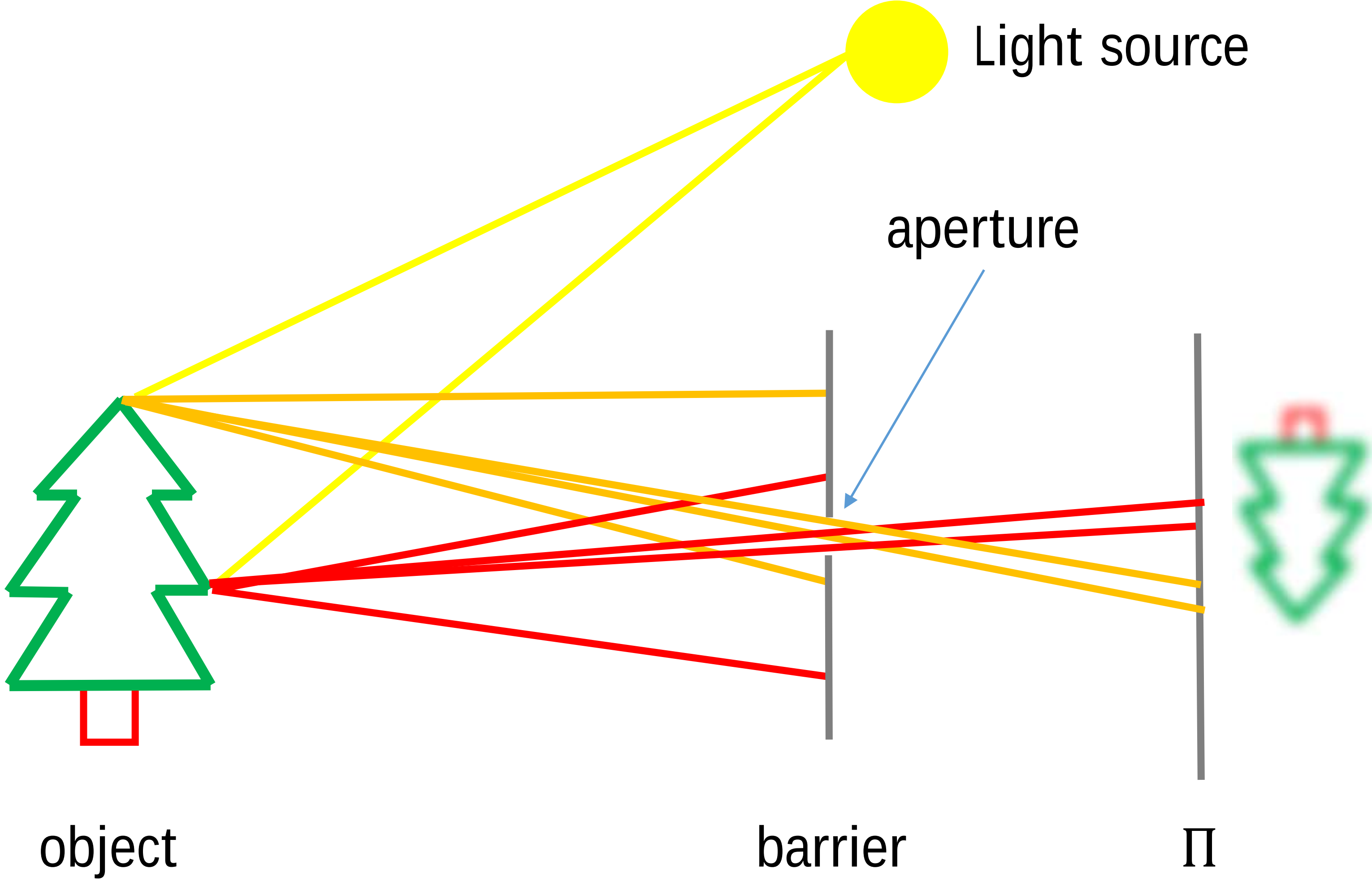
# What About the Aperture Size?

Small aperture guarantees crisp images but very little light reaches the sensor



# What About the Aperture Size?

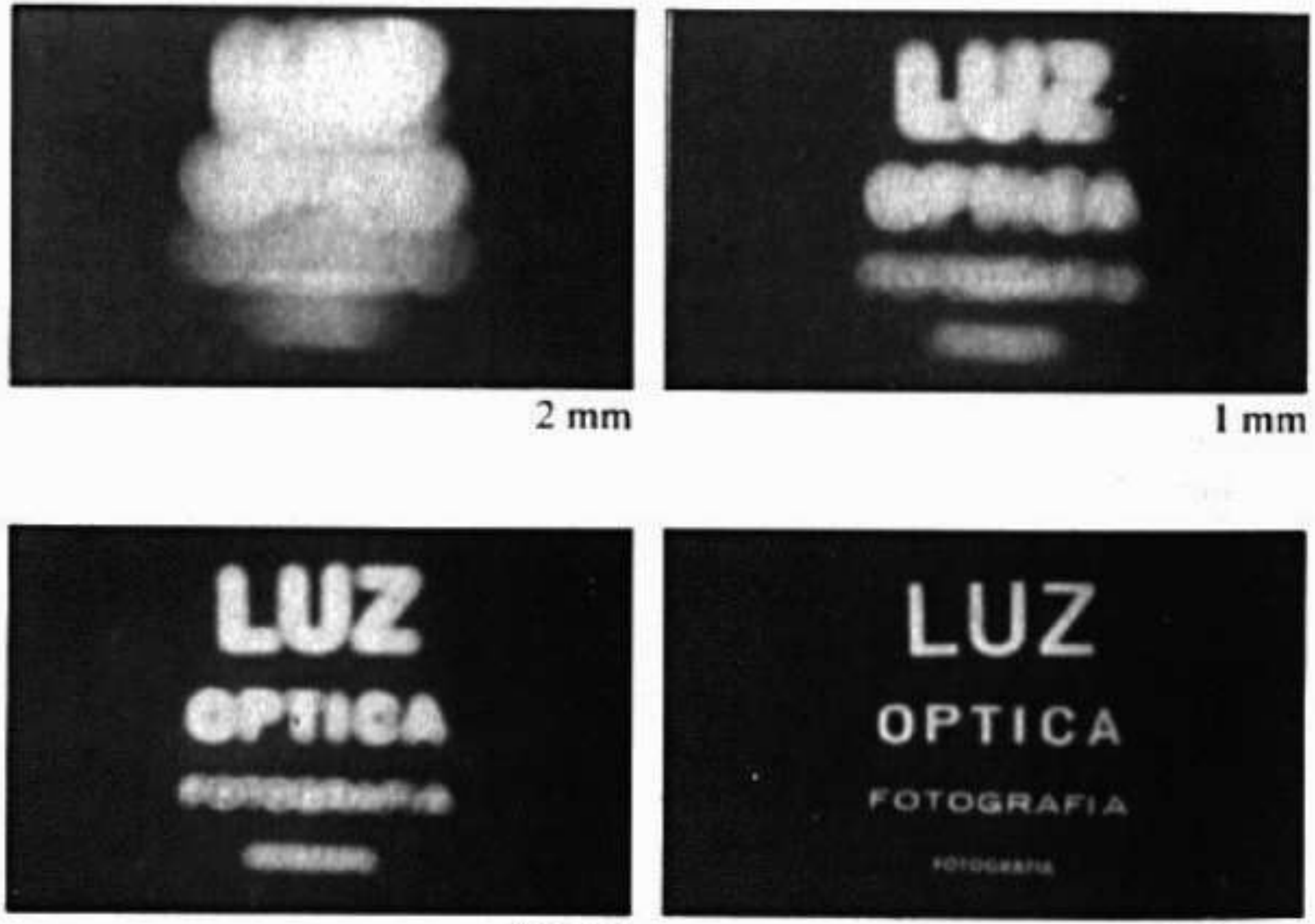
What if we increase the aperture? More raylights will enter, but…



# The aperture in a pinhole camera

With the pinhole camera the aperture size regulates the amount of light reaching the sensor.

Aperture size rules the trade-off between crispness and brightness.




2 mm 1 mm

0.6mm 0.35 mm

# Weak Perspective Camera

# Orthographic camera

When the entire scene is far compared to the depth variations in the scene, then we can write:

$$Z_P \approx Z_0 \pm \Delta_Z$$

where $\Delta_Z \ll Z_0$, such that we can approximate

$$\begin{cases} x_p = f * X_P / Z_P \\ y_p = f * Y_P / Z_P \end{cases}$$

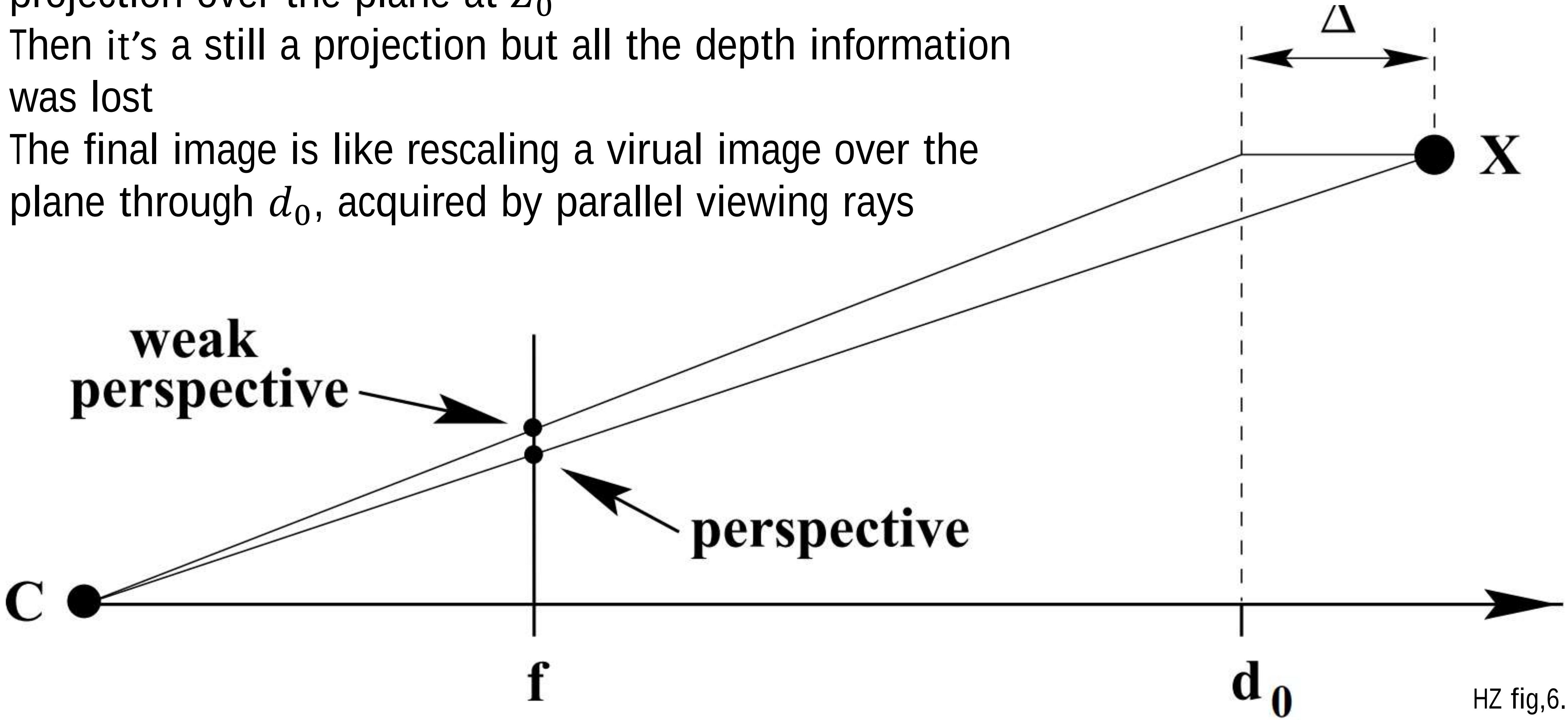
With

$$\begin{cases} x_p = f * \dfrac{X_P}{Z_P} \approx f * \boxed{\dfrac{X_P}{Z}} \\ y_p = f * \dfrac{Y_P}{Z_P} \approx f * \boxed{\dfrac{Y_P}{Z}} \end{cases}$$

These are constant terms, then persepective effects are lost

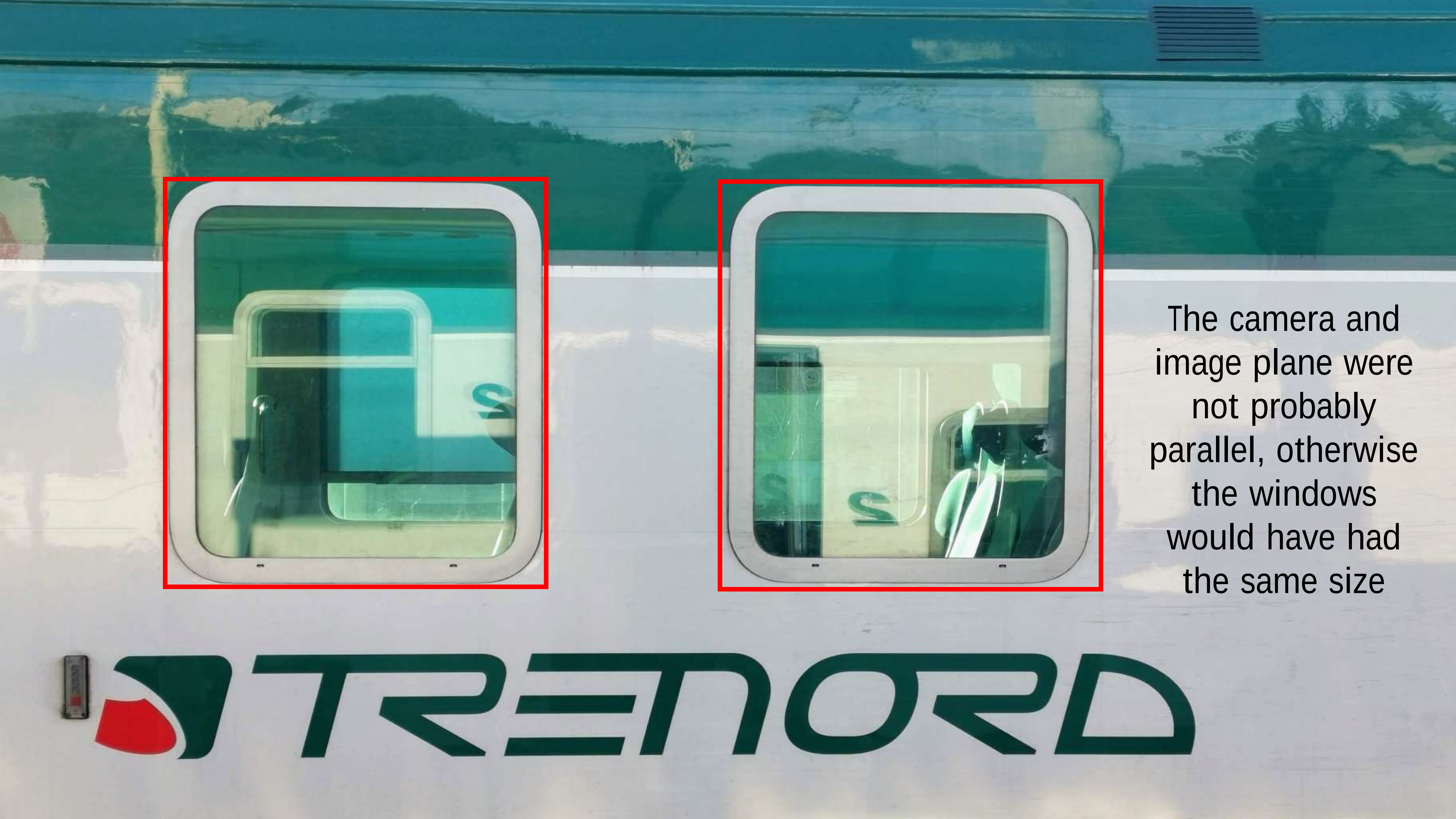
# Weak Perspective

This corresponds to projecting first by an orthogonal projection over the plane at $Z_0$
Then it’s a still a projection but all the depth information was lost
The final image is like rescaling a virual image over the plane through $d_0$, acquired by parallel viewing rays




HZ fig,6.8


TRENORD




The camera and image plane were not probably parallel, otherwise the windows would have had the same size

Do this at home!

For next lecture, shoot two pictures:

1) as projective as possible

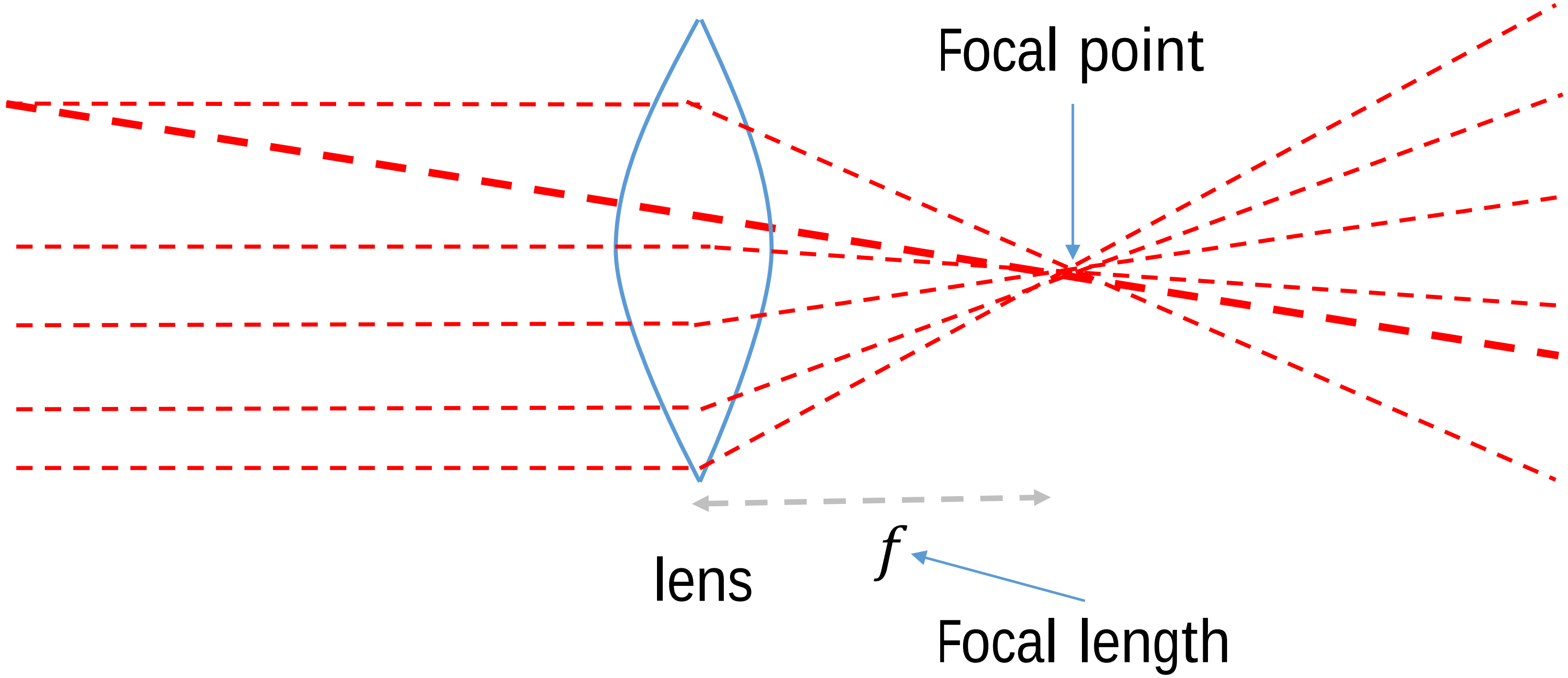
2) as orthographic as possible

# Thin Lenses and Camera

# Lenses Properties

Thin lens properties:

- Lens focus all raylights travelling parallel to the optical axis in one point, known as the focal point
- Raylights passing through the lens center are not deviated

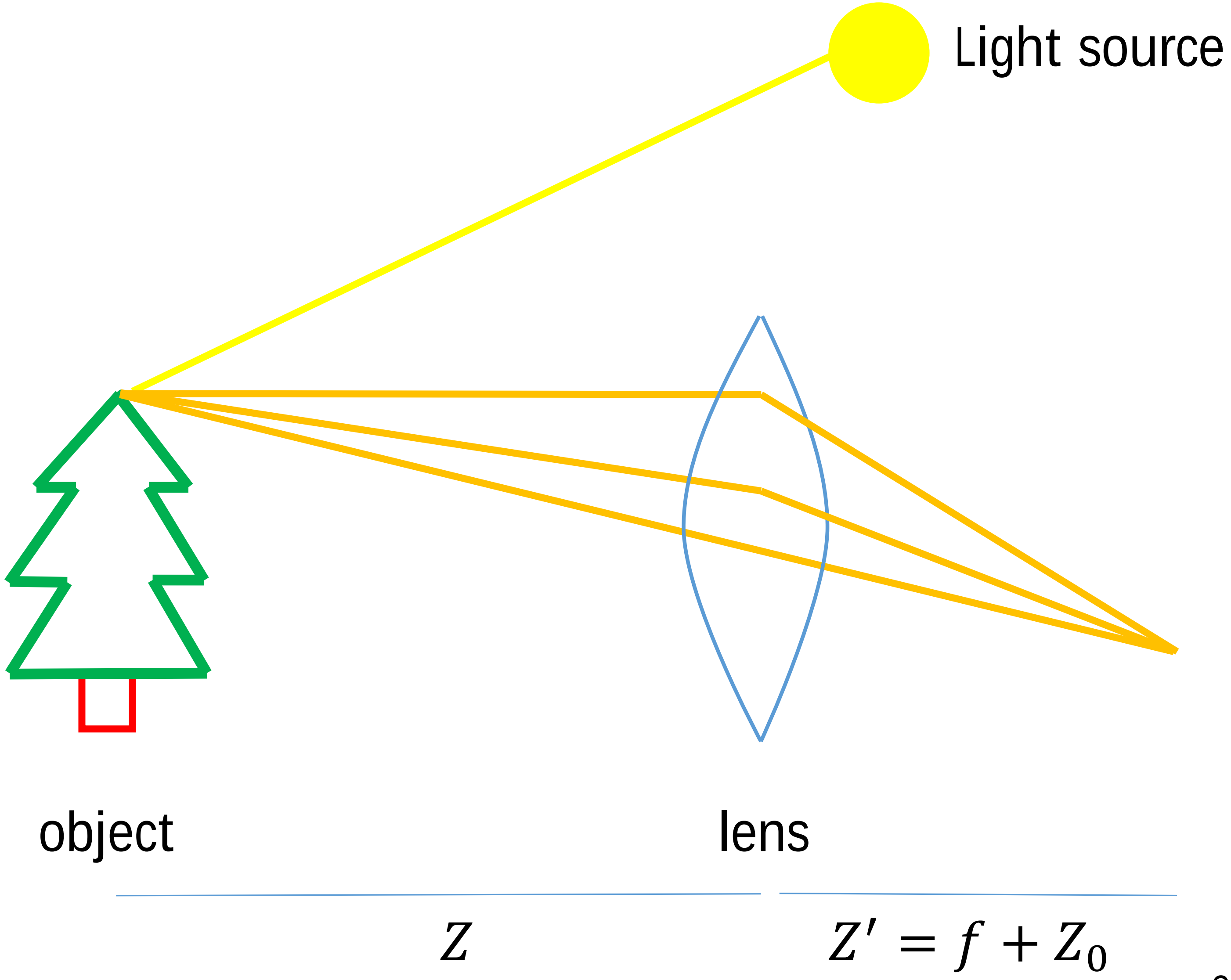


# Lenses Properties

The thin lens equation

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{f}$$

All points at depth $Z$ converge to a single point at distance $Z'$ from the lens axis

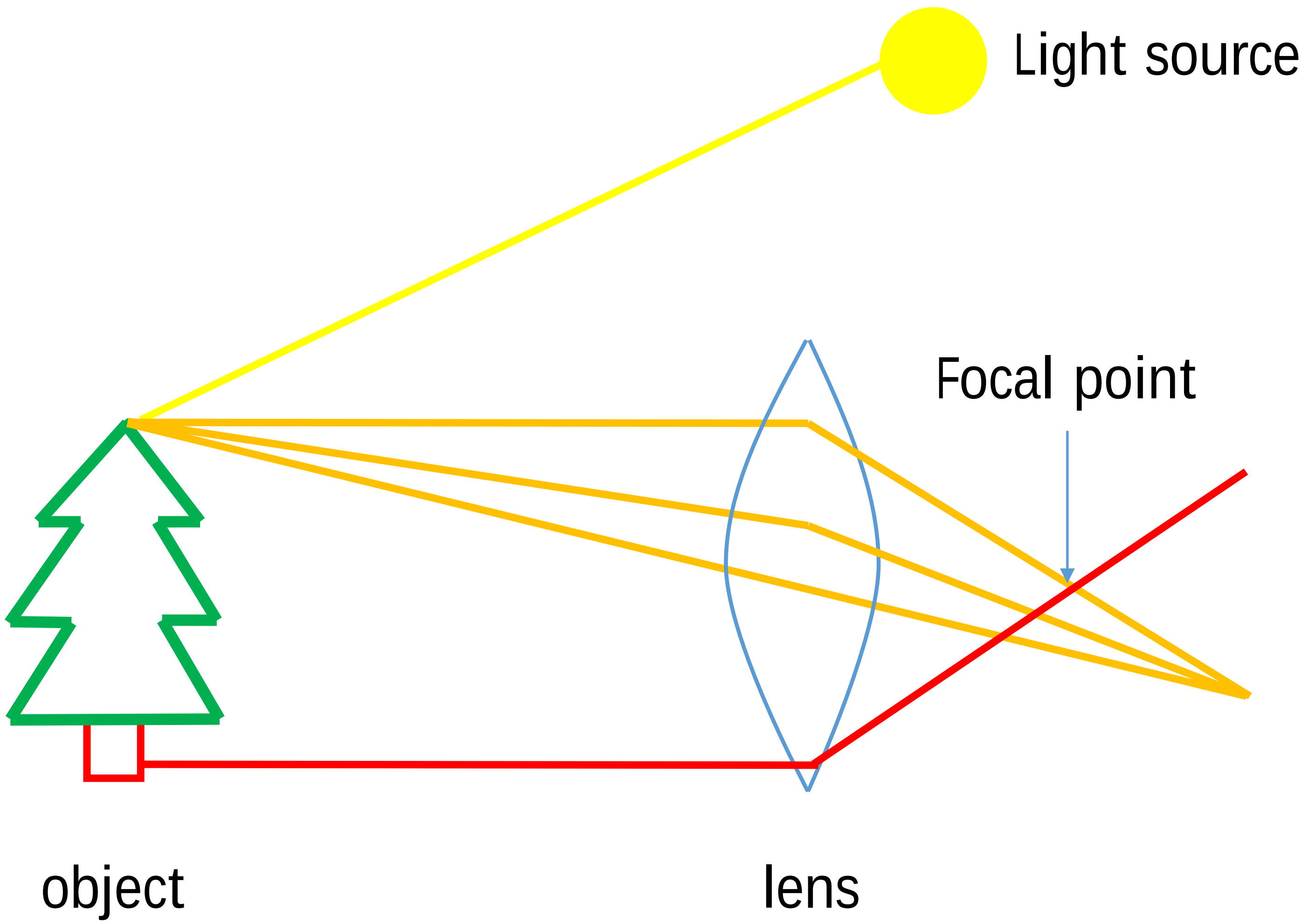



Giacomo Boracchi

# Lenses Properties

Lenses refract incoming raylights.

A properly placed lens guarantees that all rays of light emitted by some point $P$ are refracted by the lens to converge to a single image point $\boldsymbol{p}$.

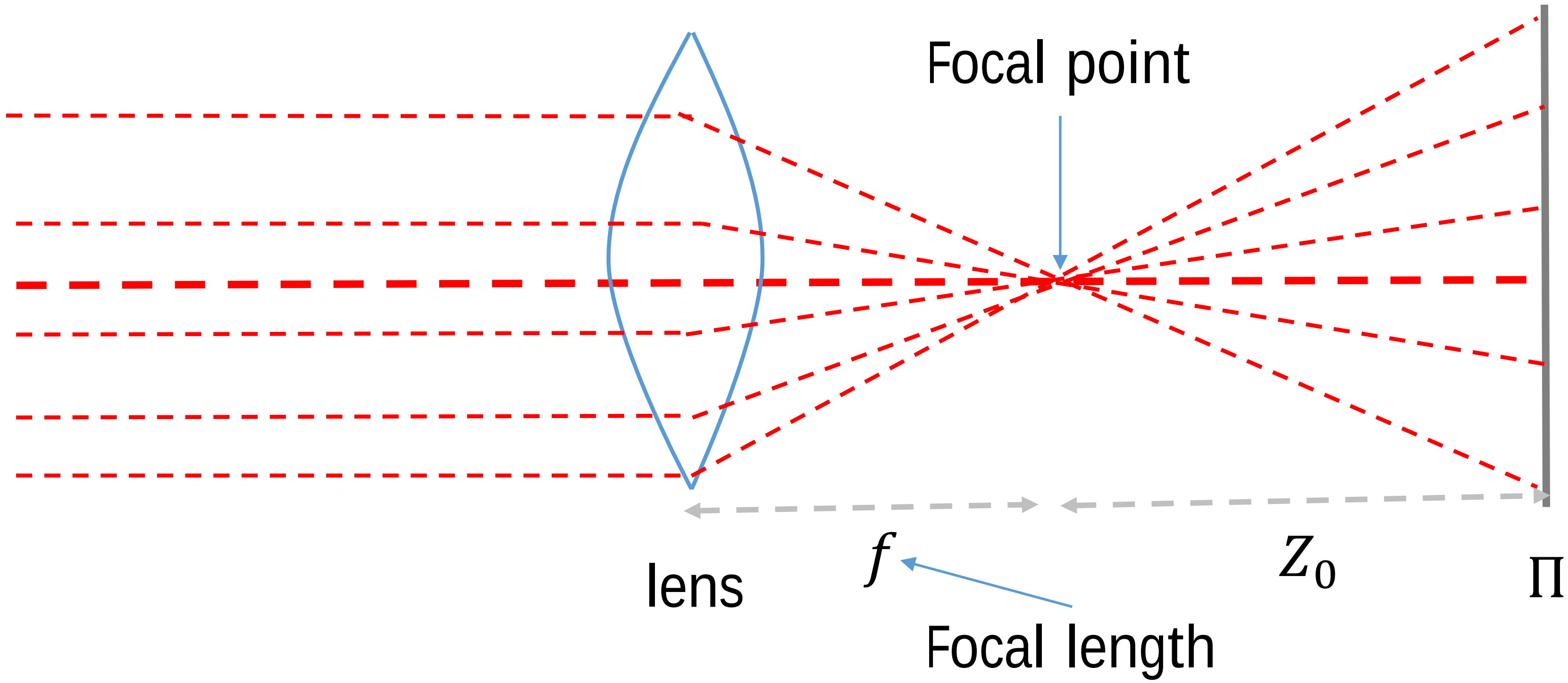


# How to make an orthographic camera?

# Camera with Lenses

Lens focus all the parallel viewing rays through its focal point. Therefore acts as a pinhole camera

Watch out: there are also othe viewing rays!

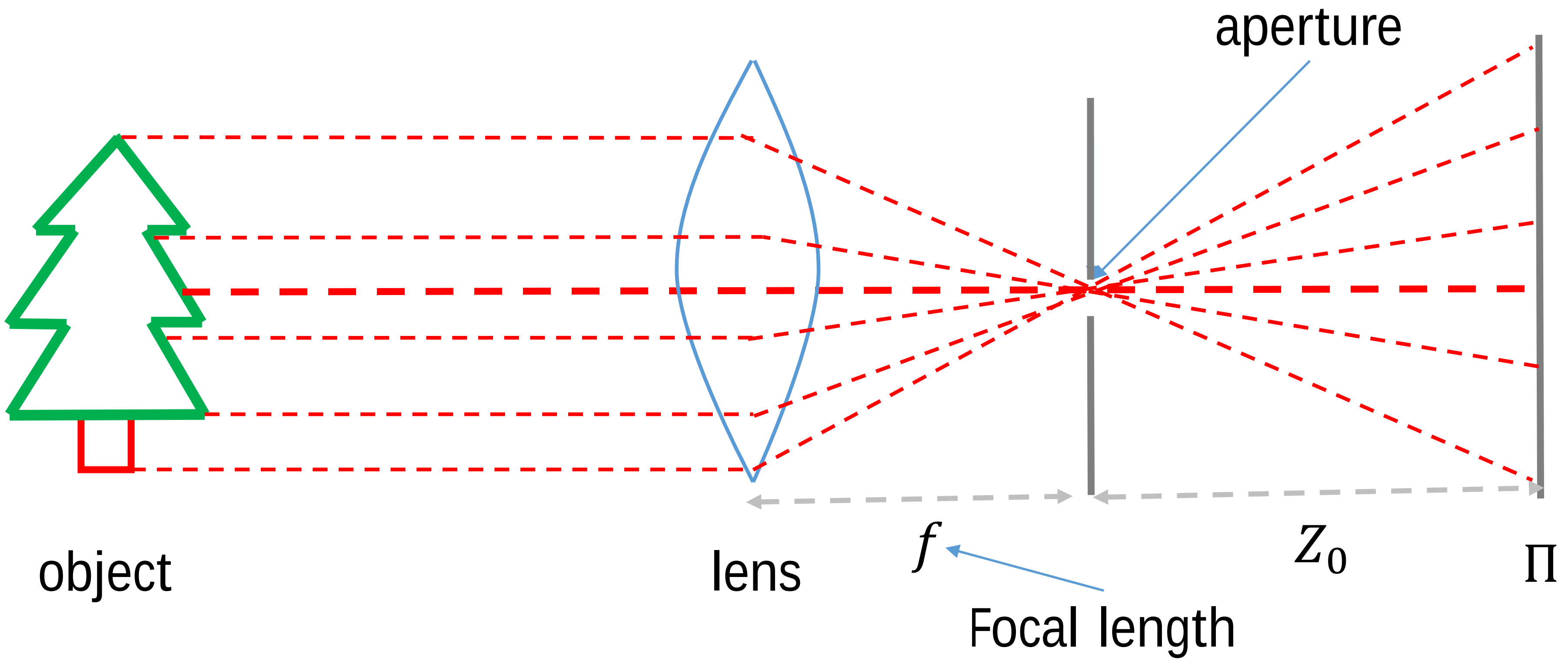


# How to make an orthographic camera?

Add a pinhole exactly at the focus.

This prevents all rays but those orthogonal to the image plane to reach the sensor.

The resulting image has no perspective variations, its just a rescaled copy.



Increase the aperture of this camera, to let more light in

More raylights from different directions will reach the sensor

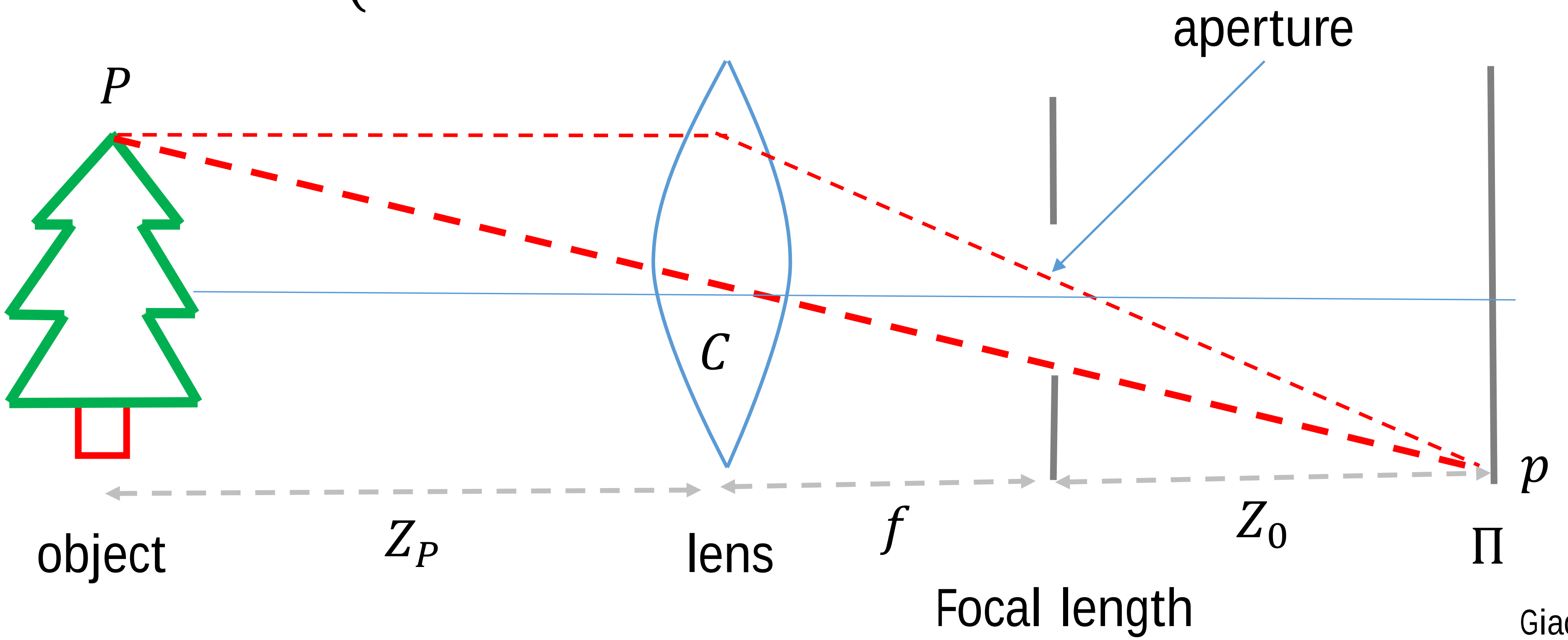
Similar projection expression for both pinhole camera and camera with lenses

# Camera with Lenses: projection equation

$$P = [X_P, Y_P, Z_P] \rightarrow \boldsymbol{p} = [x_p, y_p]$$

where

$$\begin{cases} x_p = (f + Z_0) * X_P / Z_P \\ y_p = (f + Z_0) * Y_P / Z_P \end{cases}$$

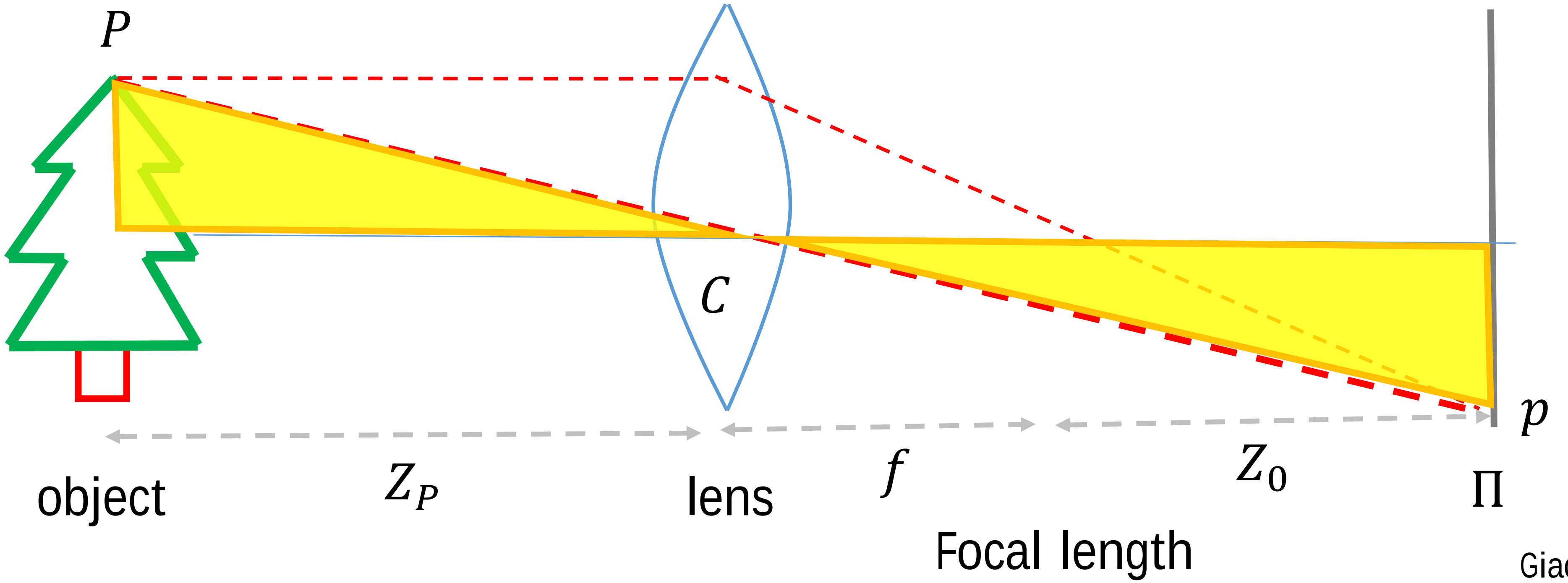


# Camera with Lenses: projection equation

$$P = [X_P, Y_P, Z_P] \rightarrow \boldsymbol{p} = [x_p, y_p]$$

where

$$\begin{cases} x_p = (f + Z_0) * X_P / Z_P \\ y_p = (f + Z_0) * Y_P / Z_P \end{cases}$$



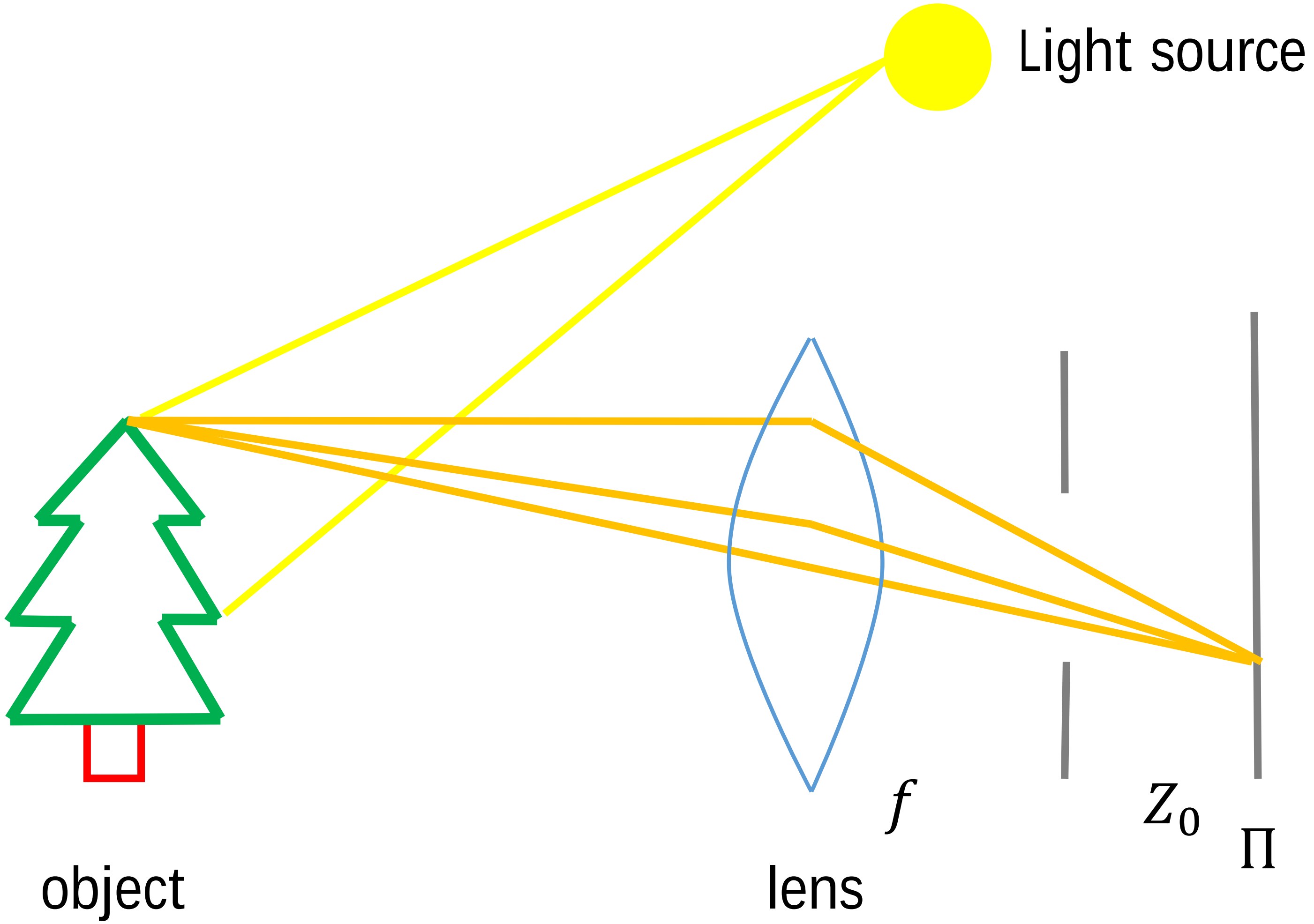
# Depth of Field

# Depth of Field

Only points having a depth $Z$ satisfying:

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{f}$$

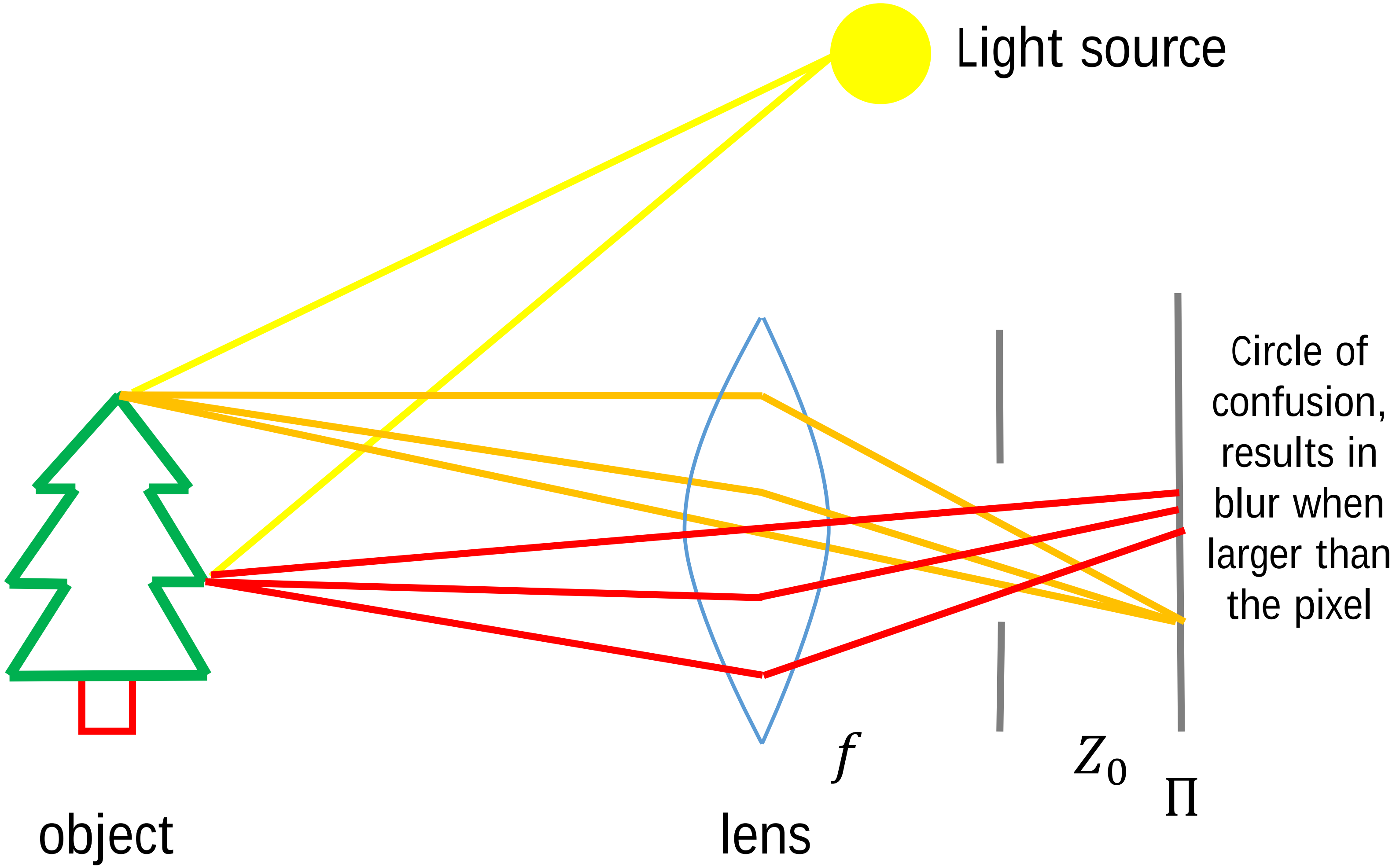
have viewing rays converging over the the image $\Pi$, which has distance $Z' = Z_0 + f$.



# Depth of Field

These points are in focus and this should hold for points at a specific depth.

Points at a different depth (see the red rays) do not converge on $\Pi$, resulting in circle of confusion (blur)

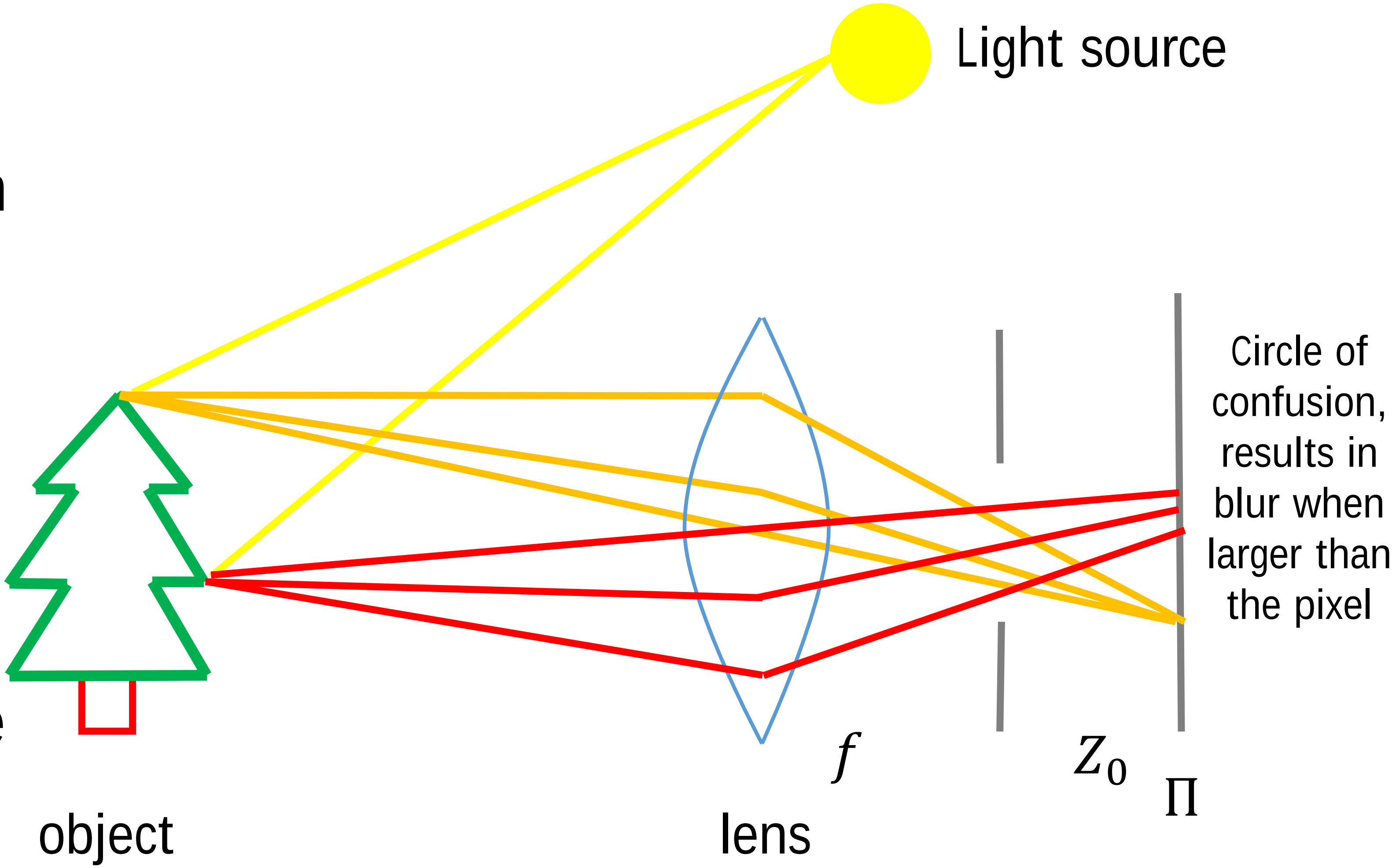


# Depth of Field

Since sensors over the image plane are discrete, there is a depth range such that viewing rays converge to the same pixel over $\Pi$.

The range of depths where point remains in focus is called depth of field

The larger the aperture, the smaller the depth of field

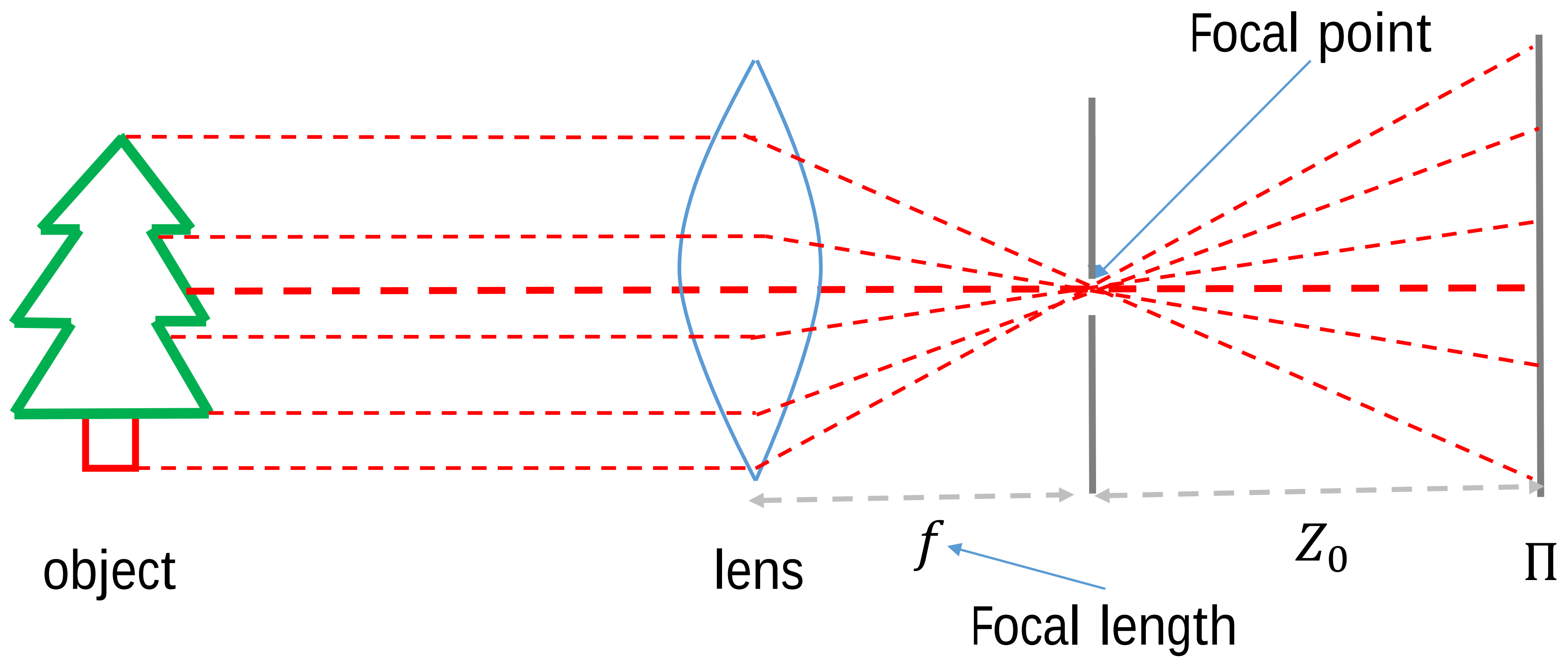


# The Camera Aperture

# Pinhole Aperture + Lenses

This is an orthographic camera, which has infinite depth of field.

All the points are in focus.

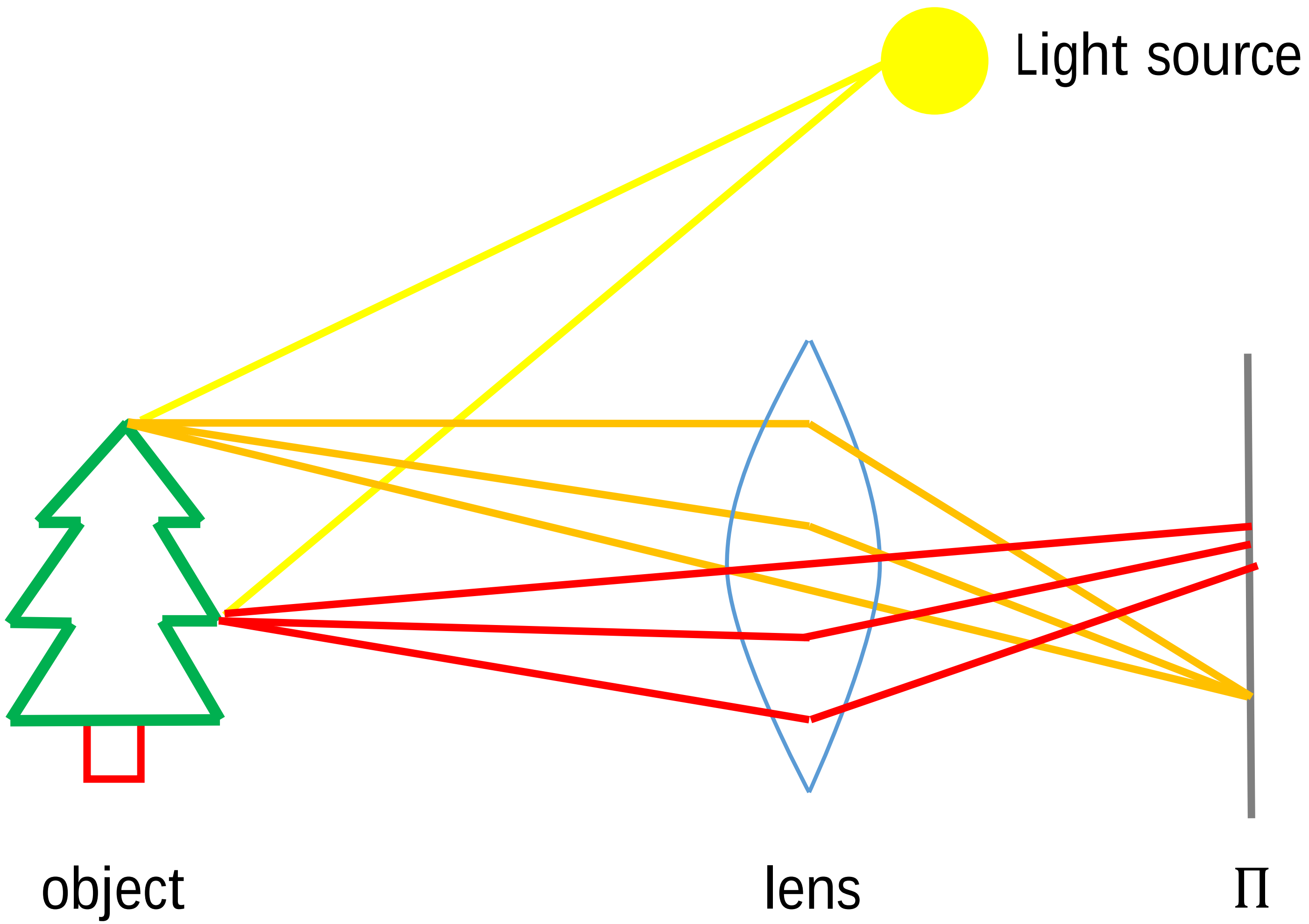


# Lenses

In manual camera mode it is possible to regulate the exposure time and the aperture.

Increasing the aperture reduces the depth of field.

The depth of field can be regulated by modifying the camera focus (lens deformation or translation)



# Lens diameter

Lens Diameter:

- It is inversely prportional to the depth of field
- Directly proportional to the amount of light reaching the sensor
- Inversely proportional to the depth of field

Pinhole camera corresponds to a lens having diameter $\rightarrow \infty$ since all points are in-focus

Lenses are always used on top of a shutter regulating the aperture size and time


The camera lenses are typically visible The shutter behind is closed


Here you can see the shutter regulating the camera aperture

Nayukim https://www.flickr.com/photos/nayukim/3969530649

# A few pictures regulating aperture and exposure

# Aperture F/3.5, exposure = 1/40 s

# The Camera Aperture

# Aperture F/8, exposure = 1/8 s

# Aperture F/22, exposure = 1 s


Very small aperture, depth of field is very large.
A long exposure time is needed to acquire a balanced picture

# Aperture F/3.5, exposure = 1/40 s

# Aperture F/7.1, exposure = 1/40 s


A smaller aperture with the same exposure time would yield a much darker image

# Exif File

Filename - IMG_7743.JPG
Make - Canon
Model - Canon EOS 400D DIGITAL
Orientation - Top left
XResolution - 72
YResolution - 72
ResolutionUnit - Inch
DateTime - 2020:01:26 20:09:28
YCbCrPositioning - Co-Sited
ExifOffset - 196
ExposureTime - 1/40 seconds
FNumber - 3.50
ExposureProgram - Aperture priority
ISOSpeedRatings - 1600
ExifVersion - 0221
DateTimeOriginal - 2020:01:26 20:09:28
DateTimeDigitized - 2020:01:26 20:09:28
ComponentsConfiguration - YCbCr
ShutterSpeedValue - 1/40 seconds
ApertureValue - F 3.50
ExposureBiasValue - 0.00
MeteringMode - Average
Flash - Flash not fired, compulsory flash mode
FocalLength - 18 mm
UserComment -

ExifImageWidth - 3888
ExifImageHeight - 2592
InteroperabilityOffset - 5766
FocalPlaneXResolution - 4433.60
FocalPlaneYResolution - 4453.61
FocalPlaneResolutionUnit - Inch
CustomRendered - Normal process
ExposureMode - Auto
White Balance - Auto
SceneCaptureType - Standard

Maker Note (Vendor): -
Macro mode - Normal
Self timer - Off
Quality - Fine
Flash mode - Not fired
Sequence mode - Single or Timer
Focus mode - One-Shot
Image size - Large
Easy shooting mode - Manual
Digital zoom - None
Contrast - Normal
Saturation - Normal
Sharpness - Normal
ISO Value - 32767

Exposure mode - Av-priority
Focal length - 3.1 - 2.5 mm
Focal units - 18/mm
Flash activity - Not fired
Flash details -
Focus mode 2 - Single
Auto ISO - 100
Base ISO - 1600
White Balance - Auto
Sequence number - 0
Camera Temperature - 27 C
Flash bias - 0 EV
Subject Distance - 0.00
Image Type - Canon EOS 400D DIGITAL
Firmware Version - Firmware 1.0.4
Owner Name - unknown
Camera Serial Number - 560100744 (2162
Sharpness (EOS 1D) - 0
Directory index (EOS 450D) - 0
File index (EOS 450D) - 1
File number - 663 - 9200
Sharpness (A0) - 0
Thumbnail: -
Compression - 6 (JPG)
XResolution - 72

Filename - oman-1.jpg
Make - Canon
Model - Canon EOS 6D
XResolution - 72
YResolution - 72
ResolutionUnit - Inch
Software - Adobe Photoshop
DateTime - 2015:01:25 22:15:57
Artist - Andrea Giuseppe Sanfilippo
ExifOffset - 248
**ExposureTime - 1/50 seconds**
**FNumber - 5**
ExposureProgram - Manual control
**ISOSpeedRatings - 3200**
Recommended Exposure Index - 3200
ExifVersion - 0230
DateTimeOriginal - 2014:12:30 15:55:04
DateTimeDigitized - 2014:12:30 15:55:04

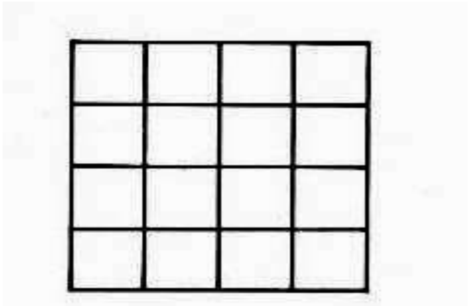
**ShutterSpeedValue - 1/50 seconds**
**ApertureValue - F 5.00**
ExposureBiasValue - 0
MaxApertureValue - F 4.00
MeteringMode - Spot
Flash - Flash not fired, compulsory flash mode
**FocalLength - 24 mm**
SubsecTimeOriginal - 00
SubsecTimeDigitized - 00
FocalPlaneXResolution - 1520.00
FocalPlaneYResolution - 1520.00
FocalPlaneResolutionUnit - Centimeter
CustomRendered - Normal process
ExposureMode - Manual
White Balance - Auto
SceneCaptureType - Standard
Serial Number - 033024001181
Lens Info - 24 105 0.00/0.00 0.00/0.00
Lens Model - EF24-105mm f/4L IS USM

# Radial Distortion

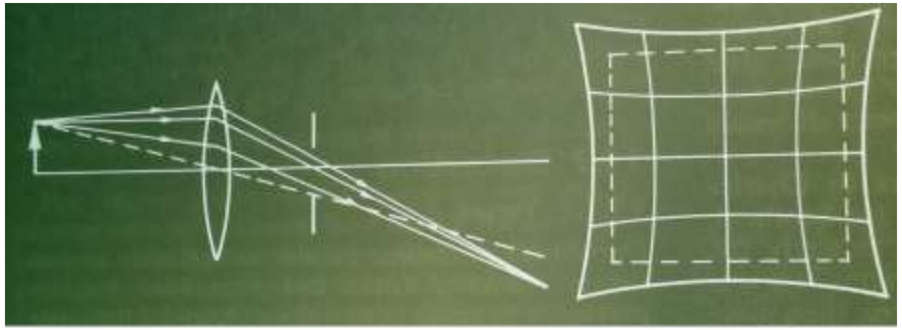
Deviations are most noticeable for rays that pass through the edge of the lens.

This can be also modeled (and compensated) in the camera model

No distortion



Pin cushion



Barrel (fisheye lens)


Image magnification decreases with distance from the optical axis

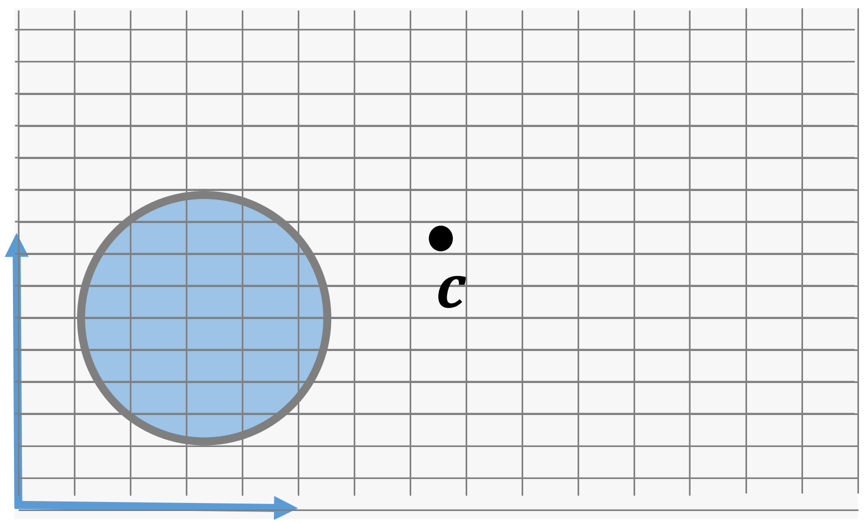
# Intrinsic Camera Parameters

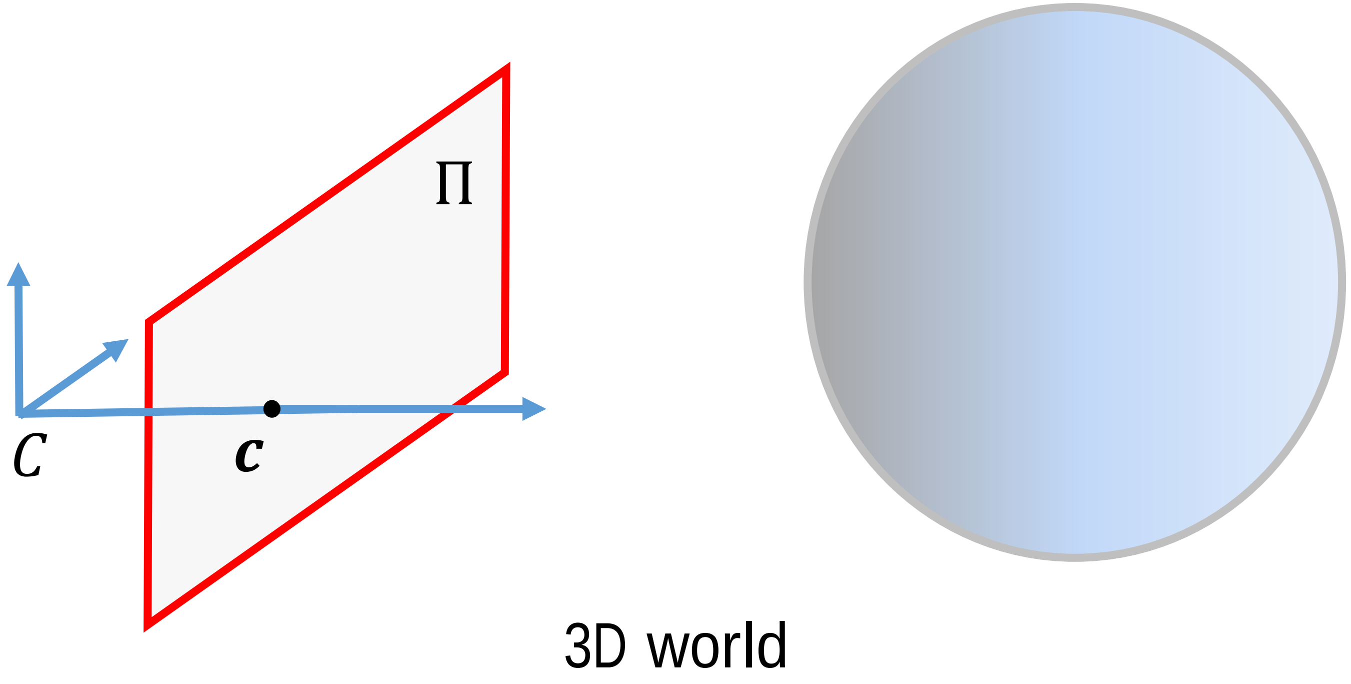
# Camera Projection to Images

Points in images:

- Might refer to a different reference system
- Images are discretized in pixels (which might not be squared)
- The physical sensor might introduce nonlinearities inthe projection

Digital image



# Compensate Offset in Digital images

2D points in the image plane and 2D point in image coordinates differ by an offset $\boldsymbol{c} = \left[c_x, c_y\right]$

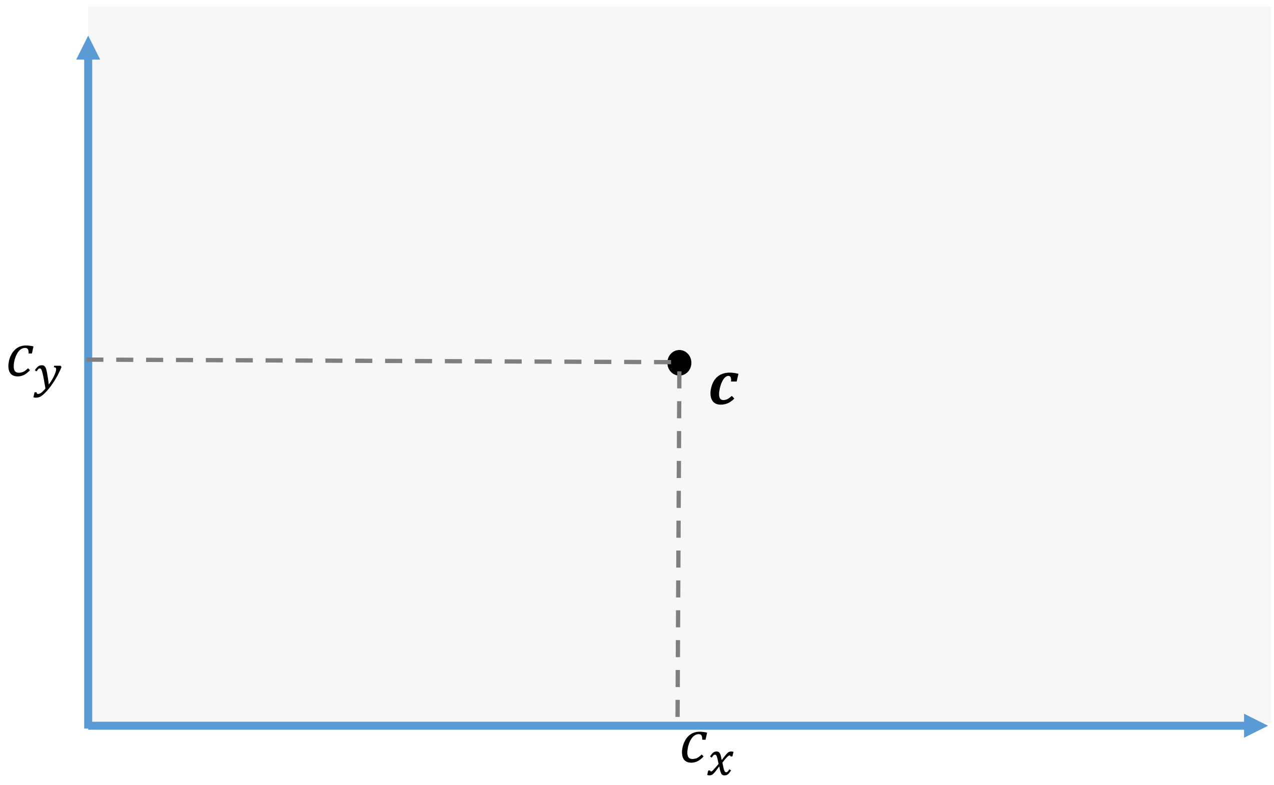
We can easily accomodate this in the camera projection equations

Camera Projection

$$P = [X_P, Y_P, Z_P] \rightarrow \boldsymbol{p} = [x_p, y_p]$$

where

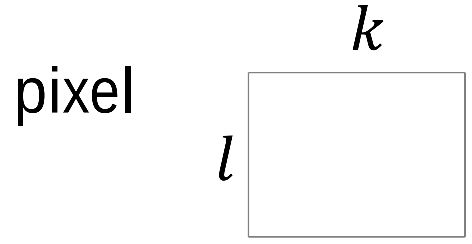
$$\begin{cases} x_p = f * \dfrac{X_P}{Z_P} + c_x \\ y_p = f * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$



# Compensate for different units

Coordinates in images are expressed in terms of pixels, while on the image plane $\Pi$ they are still expressed in cm as in the 3D world

Pixels might have aspect ratio $\neq 1$



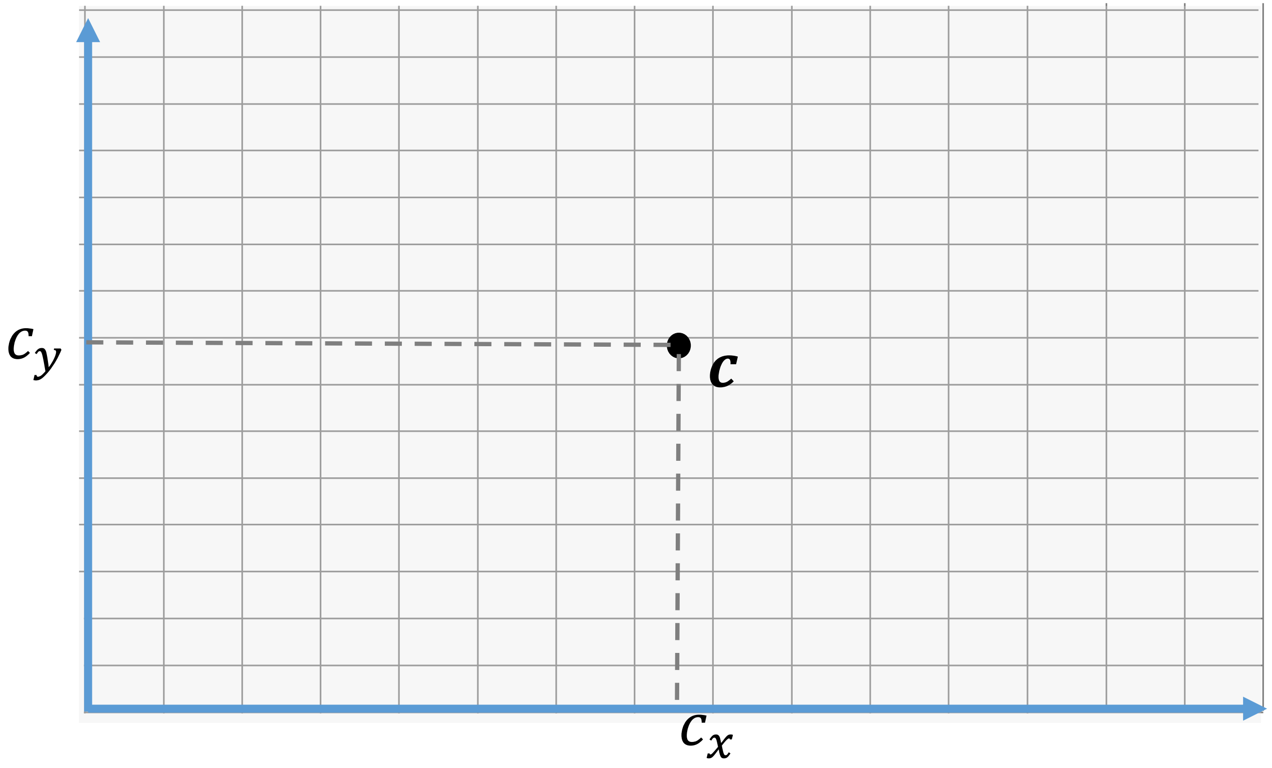

Let $l,k$ be the scaling factor pixel/cm

Camera Projection

$$P = [X_P, Y_P, Z_P] \to p = [x_p, y_p]$$

where

$$\begin{cases} x_p = \overset{\alpha}{\boxed{fk}} * \dfrac{X_P}{Z_P} + c_x \\ y_p = \underset{\beta}{\boxed{fl}} * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$

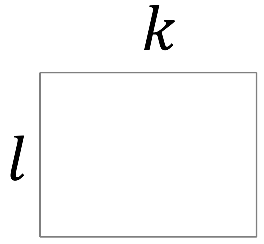


# Compensate for different units

Coordinates in images are expressed in terms of pixels, while on the image plane $\Pi$ they are still expressed in cm as in the 3D world

Pixels might have aspect ratio $\neq 1$

pixel



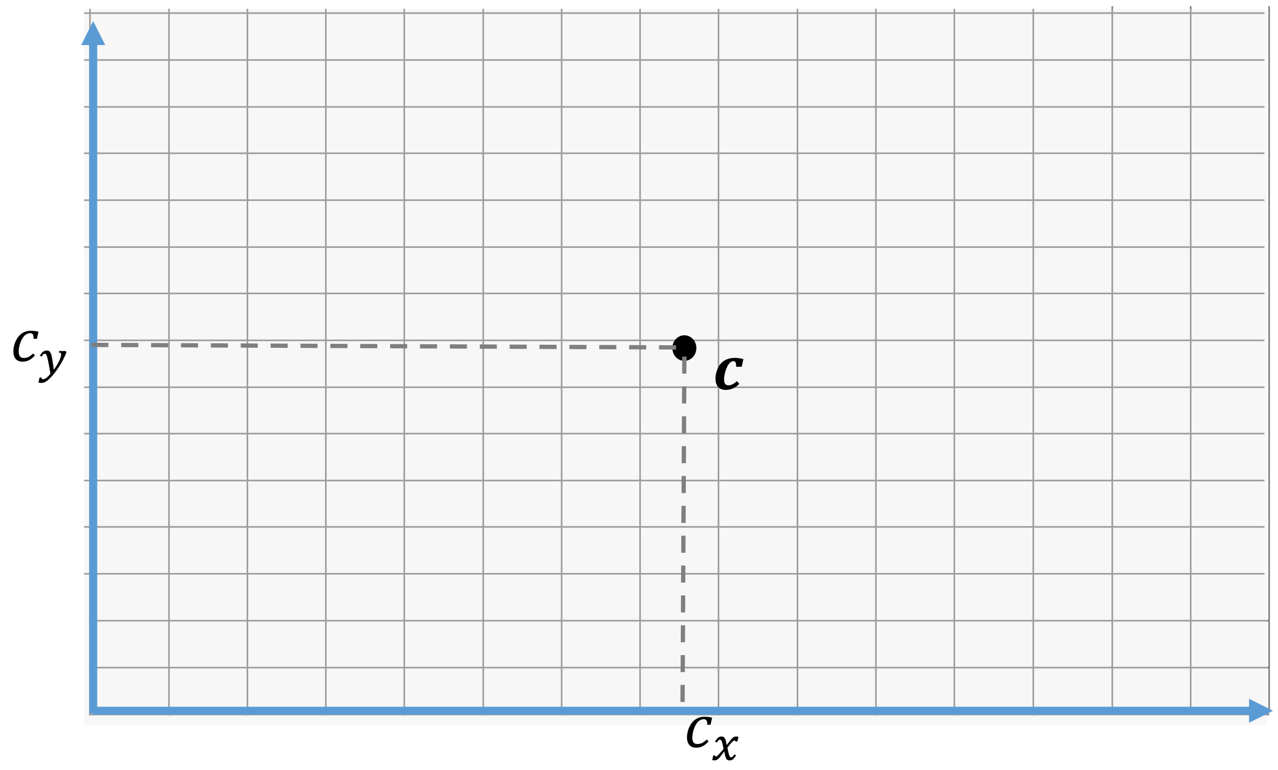

Let $l$,$k$ be the scaling factor pixel/cm

Camera Projection

$$P = [X_P, Y_P, Z_P] \rightarrow p = [x_p, y_p]$$

where

$$\begin{cases} x_p = \alpha * \dfrac{X_P}{Z_P} + c_x \\ y_p = \beta * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$



# Is this a linear mapping?

$$P = [X_P, Y_P, Z_P] \rightarrow p = [x_p, y_p]$$

$$\begin{cases} x_p = \alpha * \dfrac{X_P}{Z_P} + c_x \\ y_p = \beta * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$

# No, it is not! Because of division by $Z_p$ which is an input!

$$P = [X_P, Y_P, Z_P] \rightarrow p = [x_p, y_p]$$

$$\begin{cases} x_p = \alpha * \dfrac{X_P}{Z_P} + c_x \\ y_p = \beta * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$

# Is there a trick to represent this as a linear trasformation?

$$P = [X_P, Y_P, Z_P] \rightarrow p = [x_p, y_p]$$

$$\begin{cases} x_p = \alpha * \dfrac{X_P}{Z_P} + c_x \\ y_p = \beta * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$

Yes, homogeneous coordinates!

# Homogeneous Coordinates

Instead of representing a point as $n - \mathrm{tuple}$ in the Euclidean space, it is represented as a class of equivalence in the space of $(n+1) - \mathrm{tuple}$.

Euclidean plane $\mathbb{R}^2$ -> Homogeneous coordinate system $\mathbb{P}^2$ (Projective plane)

$$\mathbf{p} = \left[x_p, y_p\right]' \in \mathbb{R}^2 \rightarrow \boldsymbol{p} = \left[x_p, y_p, 1\right]' \in \mathbb{P}^2$$

Euclidean plane $\mathbb{R}^3$ -> Homogeneous coordinate system $\mathbb{P}^3$ (Projective space)

$$P = [X_P, Y_P, Z_P]' \in \mathbb{R}^3 \rightarrow \boldsymbol{P} = [X_P, Y_P, Z_P, 1]' \in \mathbb{P}^3$$

# Equivalence Relation in $\mathbb{P}^2$

In homogeneous cooridnates all the points that differs from a scalar multiplication are equivalent

$$\boldsymbol{p} \in \mathbb{P}^2, \boldsymbol{p} = \left[x_p, y_p, 1\right]' = \lambda\left[x_p, y_p, 1\right]' = \left[\lambda x_p, \lambda y_p, \lambda\right]' \forall \lambda \in \mathbb{R}$$

# Mapping Homogeneous ↔ Projective

To move from homogeneous to Euclidean coordinates it is necessary to normalize first by the last component, and then remove the last

Equality between coordinates in $\mathbb{P}^2$ and $\mathbb{R}^2$ occurs only when the last homogeneous coordinate equals $\mathbf{1}$

Homogeneous coordinate system $\mathbb{P}^2$ -> Euclidean plane $\mathbb{R}^2$

$$\boldsymbol{p} = \left[x_p, y_p, w_p\right]' \in \mathbb{P}^2 \rightarrow \left[\frac{x_p}{w_p}, \frac{y_p}{w_p}, 1\right]' \rightarrow \boldsymbol{p} = \left[\frac{x_p}{w_p}, \frac{y_p}{w_p}\right]' \in \mathbb{R}^2$$

Homogeneous coordinate system $\mathbb{P}^3$-> Euclidean plane $\mathbb{R}^3$

$$P = \left[X_P, Y_P, Z_P, W_p\right]' \in \mathbb{P}^3 \rightarrow \left[\frac{X_P}{W_p}, \frac{Y_P}{W_p}, \frac{Z_P}{W_p}, 1\right]' \rightarrow \boldsymbol{p} = \left[\frac{X_P}{W_p}, \frac{Y_P}{W_p}, \frac{Z_P}{W_p}\right]' \in \mathbb{R}^3$$

# A model for $\mathbb{P}^2$

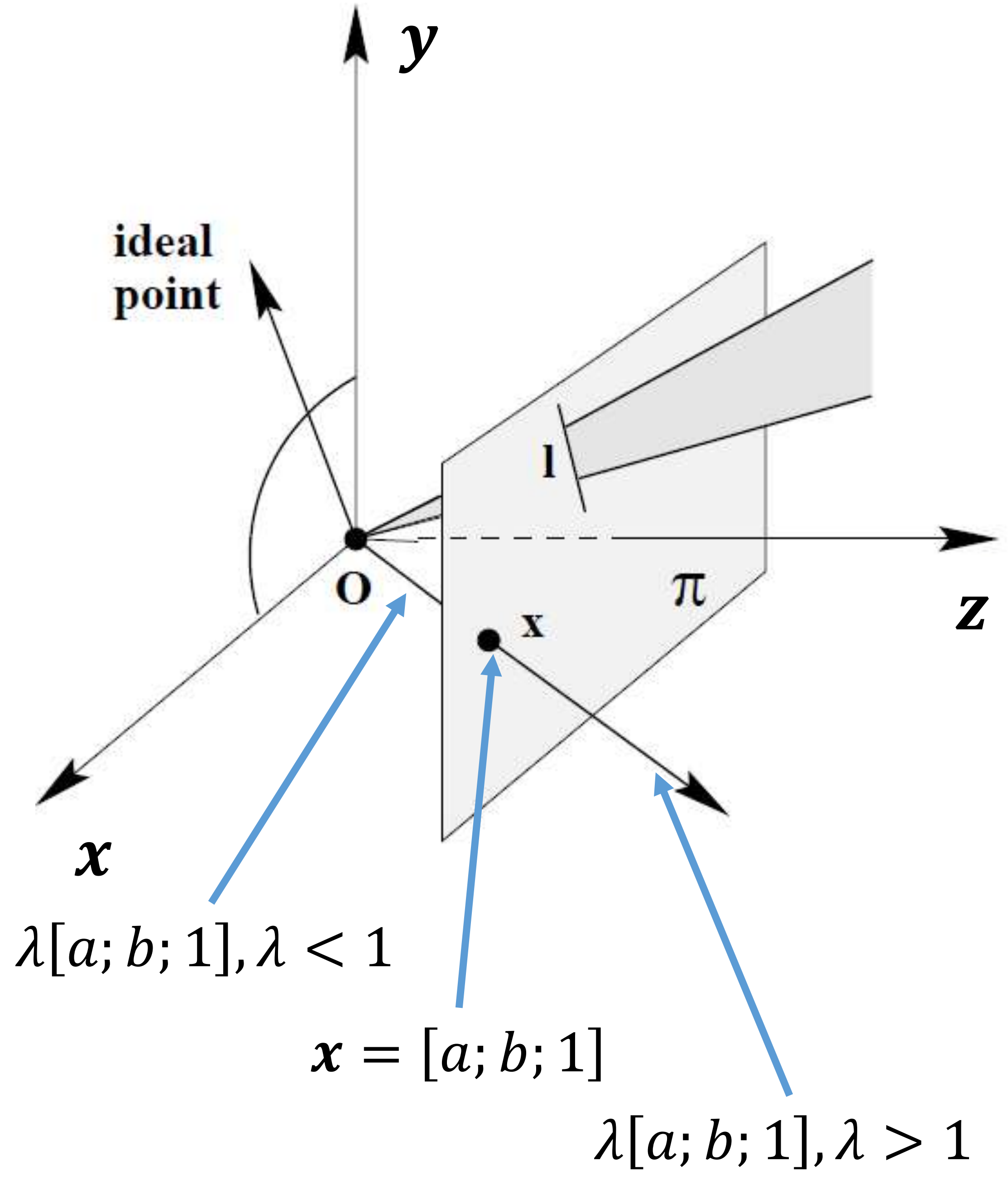
Points of $\mathbb{P}^2$ are represented by all the rays of $\mathbb{R}^3$ through the origin, since all

$$\boldsymbol{x} = \lambda[a; b; 1], \qquad \forall\lambda$$

corresponds to the same point

Intersection with the plane $z = 1$ yields the representative points $[x; y; 1]$ which defines the mapping $\mathbb{P}^2 \leftrightarrow \mathbb{R}^2$

Lines lying in the plane $\boldsymbol{z} = 0$ represent ideal points (directions) as these do not instersect $\Pi$.




Hartley Zisserman Fig.2.1

# ...back to Intrinsic Camera Parameters

# Camera Projection Equations in $\mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$P = [X_P, Y_P, Z_P] \rightarrow p = [x_p, y_p]$$

$$\begin{cases} x_p = \alpha * \dfrac{X_P}{Z_P} + c_x \\ y_p = \beta * \dfrac{Y_P}{Z_P} + c_y \end{cases}$$

# Camera Projection Equations in $\mathbb{P}^3 \to \mathbb{P}^2$

$$\boldsymbol{p} = \begin{bmatrix} \alpha * \frac{X_P}{Z_P} + c_x \\ \beta * \frac{Y_P}{Z_P} + c_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha * X_P + c_x Z_P \\ \beta Y_P + c_y Z_P \\ Z_P \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix}$$

$$\boldsymbol{p} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P$$

$$\boxed{\boldsymbol{p} = MP = K[I_3, \boldsymbol{0}]P}$$

Camera Projection in homogeneous coordinates becomes linear!

Rmk: projection preserves equivalence relation of homogeneous points

Rmk: We will work in homogeneous coordinates, unless stated otherwise

# The Camera Matrix

Let us denote by $M$ the matrix performing the projection,

$$M = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Which can be rewritten as

$$M = K\left[I_{3,} \ \mathbf{0}\right]$$

And the matrix $K$ is the camera matrix

$$K = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

As it contains all the parameters referred to the camera: the aspect ratio and the position of the camera center

# The Camera Matrix

How many parameters in the camera matrix?

$$K = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Four parameters:

- $\alpha, \beta$ the aspect ratio
- $c_x, c_y$ the principal point (image of camera center)

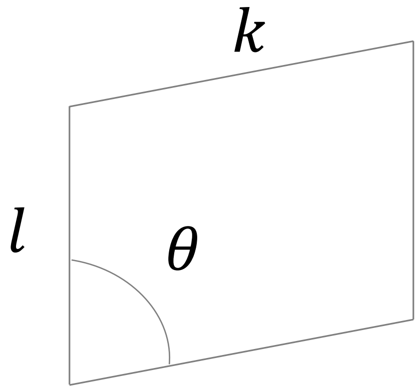
These are the intrinsic parameters of the camera, as they change only then the camera type changes

# The Camera Matrix (including skewness)

How many parameters in the camera matrix?

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \dfrac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

The image reference axis is skewed of an angle $\theta$




Five parameters:

- $\alpha, \beta$ the aspect ratio
- $c_x, c_y$ the principal point (image of camera center)
- $\theta$ is the skewness (when $\theta = \pi/2$, $\cot\theta = 0, \sin\theta = 1$)

These are the intrinsic parameters of the camera, as they change only then the camera type changes

# The Camera Matrix

$K$ is contains the intrinsic camera parameters, which depends only on the camera settings themselves (e.g. the focus)

$$K = \begin{bmatrix} \alpha & -\alpha \cot\theta & c_x \\ 0 & \dfrac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Rmk $K$ is a non-singular matrix, being a triangular matrix with nonzero entries over the diagonal.

# The determinant of a triangular matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}$$

$$\det A = a_{11} \cdot \det \begin{pmatrix} a_{22} & a_{22} & \cdots & a_{2n} \\ & a_{33} & \cdots & a_{3n} \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}$$

# The determinant of a triangular matrix

$$\det A = a_{11}a_{22} \cdot \det \begin{pmatrix} a_{33} & a_{34} & \cdots & a_{3n} \\ & a_{44} & \cdots & a_{4n} \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}$$

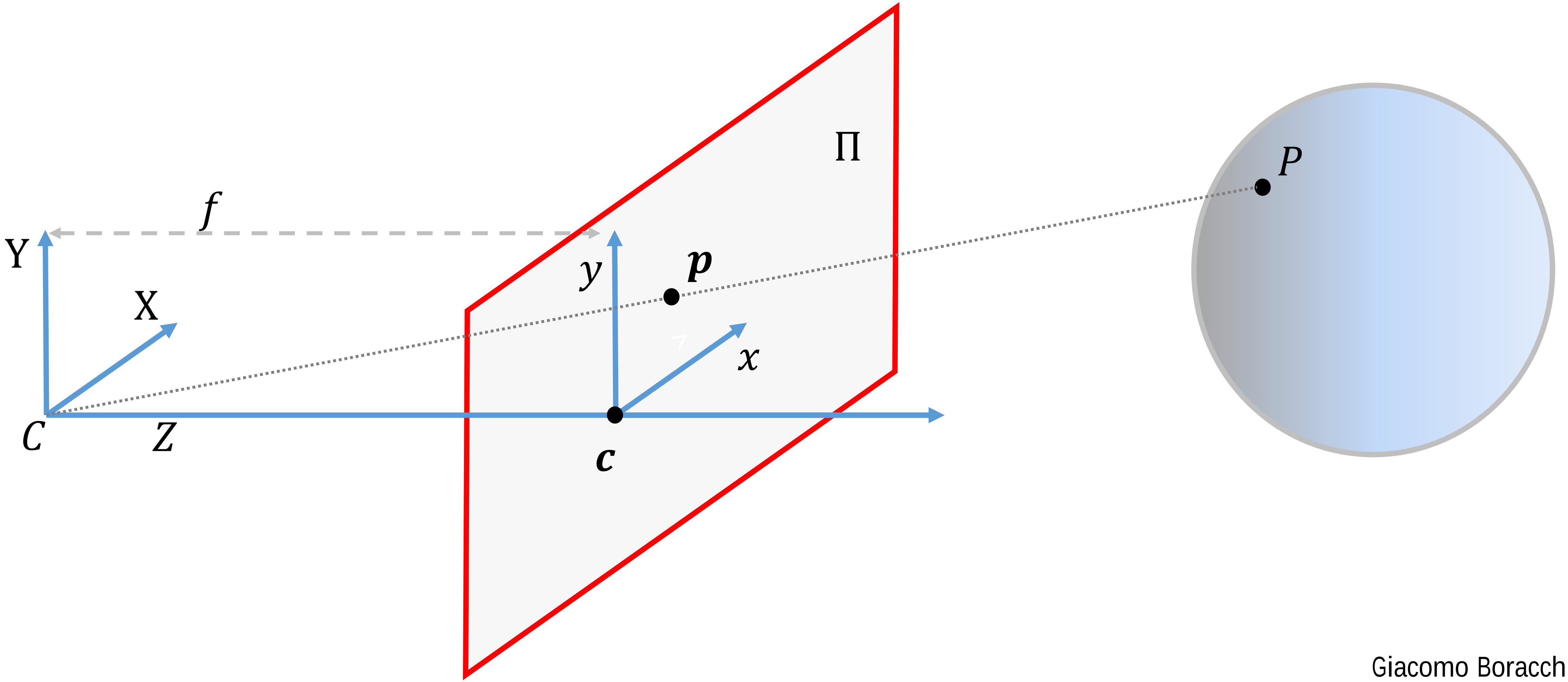
$$\det A = a_{11} \cdots a_{n-2,n-2} \cdot \det \begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ & a_{nn} \end{pmatrix} = a_{11} \cdots a_{nn}$$

# Camera Projection

## Extrinsic Camera Parameters

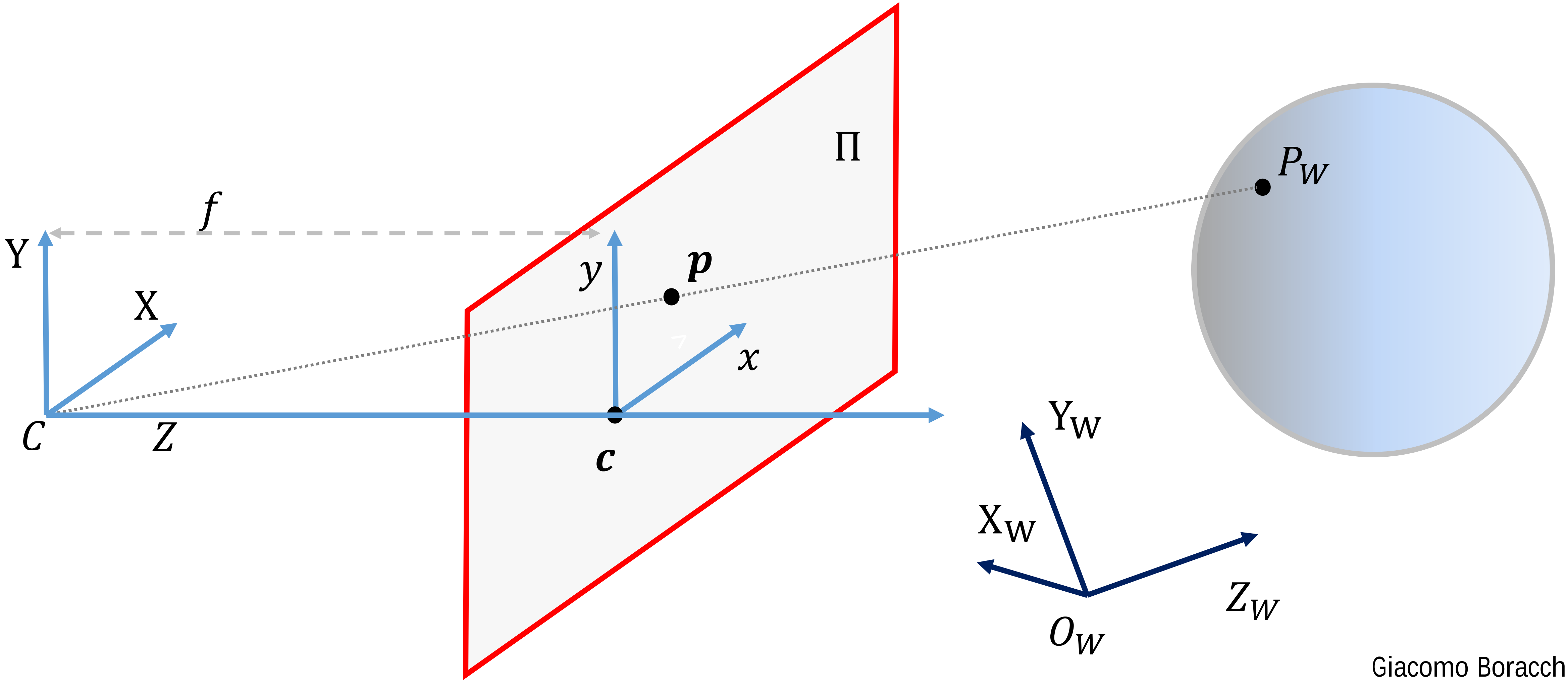
# The 3D reference system

So far we have assumed that the 3D reference system has origin in the camera center $C$.



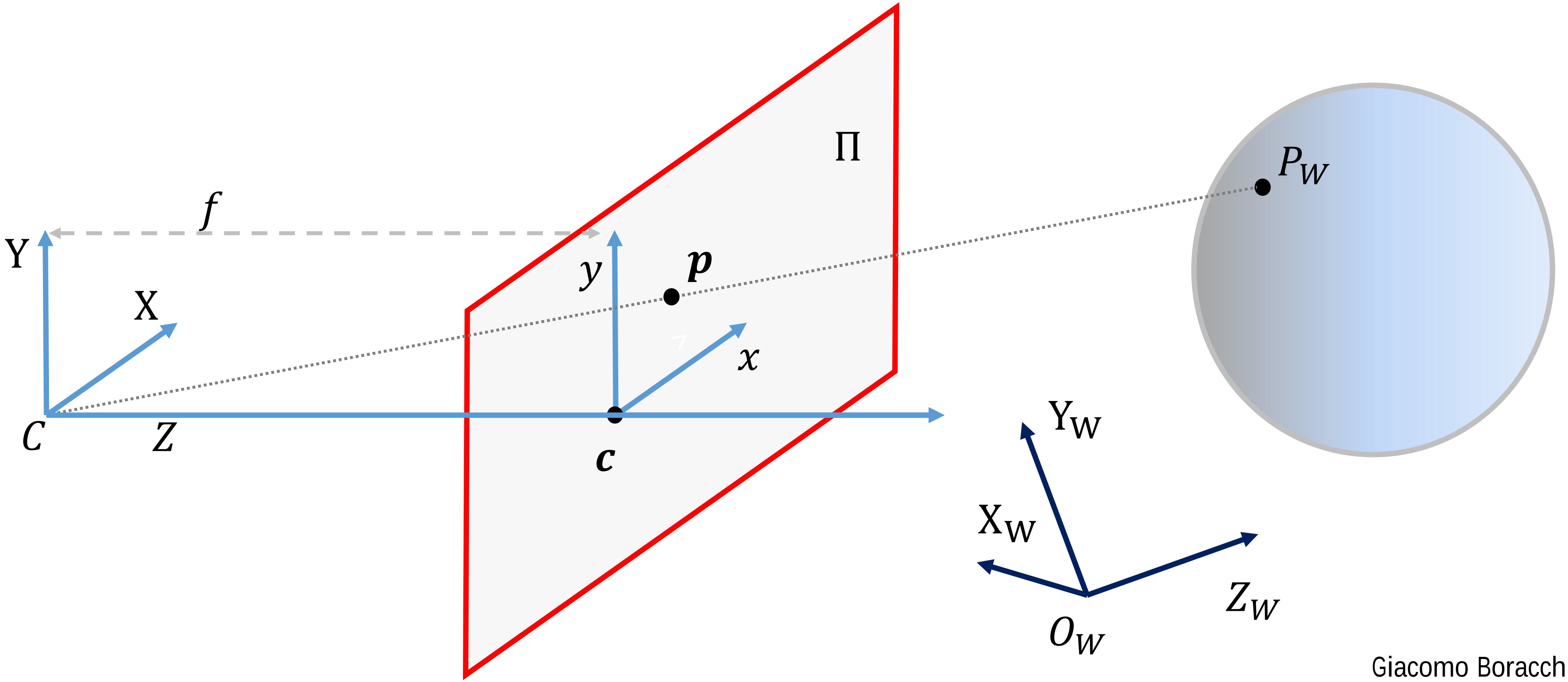
# The 3D reference system

What if we need to consider a different reference system for the 3D world?



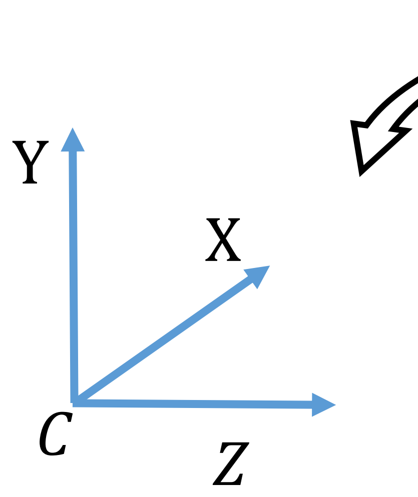
# The 3D reference system

This is very common in multi-camera settings and for camera calibration, where we have to refer to (a common) *world reference system in* $O_W$



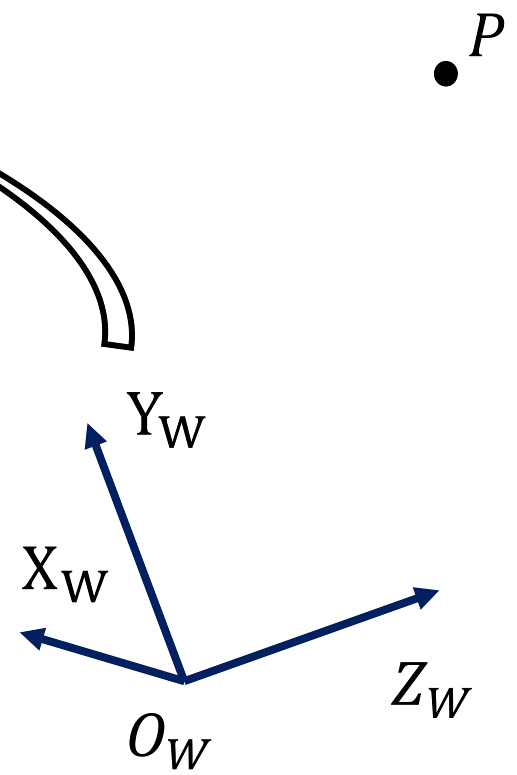
# The 3D reference system

It is enough to map the two reference systems and include this in the camera matrix (it's a matter of composing matrix product!)




Mapping described by
- rotation matrix $R$
- traslation vector $T$

$$P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_W$$



# So, the camera projection becomes

The projection in the reference system at the camera center is

$$\boldsymbol{p} = K[I_3, 0_3]P$$

Coordinate change between world and camera reference system:

$$P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_W$$

The projection becomes

$$\boldsymbol{p} = K[I_3, 0_3]\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_W = K[R, T]P_W$$

Thus, in general, the projection matrix $M \in \mathbb{R}^{3,4}$ can be written as:

$$M = K[R, T]$$

# The Projection Matrix $M = K[R,T]$

The camera projection $\mathbb{P}^3 \to \mathbb{P}^2$ can be written as

$$\boldsymbol{p} = M\ P$$

Where $M \in \mathbb{R}^{3,4}$ is the Projection Matrix

$$M = K[R,T]$$

Intrinsic paramters: $K$ (aspect ratio, camera center, skewness). These only depend on how the camera is build. 5 parameters

Extrinsic parameters: $[R,T]$ (rotation and translation of the reference system in the camera center w.r.t. the world reference system). These only depend on the camera position. 6 parameters (3 rotation angles in $R$, 3 components of $T$)

Overall: 11 prameters in $M$ (a full 3 x 4 homogeneous matrix)

# The Projection Matrix $M = K[R, T]$

The camera projection $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ can be written as

$$\boldsymbol{p} = M\, P$$

Where $M \in \mathbb{R}^{3,4}$ is the Projection Matrix

$$M = K[R, T]$$

Remark:

- $M$ has rank 3, since $M \in \mathbb{R}^{3,4}$
- $KR$ is non singular, since $K$ is upper-triangular (with nonzero diagonal) and $R$ is a rotation matrix

*If* $KR$ would have been singular, the projection output would be a line or a point, but not a 2D object.

# The Camera Center $C$

The 3D coordinates of the center of a camera having matrix $M \in \mathbb{R}^{3,4}$ satisfy

$$C \in RNS(M)$$

Where $RNS(\cdot)$ denotes the Right Null Space.

Note that when the $RNS(M)$ has dimension 1 (i.e. always but in degenerate cases) all the points $C \in RNS(M)$ coincide in the homogenous space.

*Proof*

Let us consider $C \in RNS(M)$, then

$$MP = M(P + \lambda C) \quad \forall \lambda \in \mathbb{R}, \qquad \forall P \in \mathbb{P}^3$$

This means that the line $P + \lambda C$ does not change its projection through $M$, thus $P + \lambda C$ is a viewing ray. Since this has to hold $\forall P \in \mathbb{P}^3$, this means that $C$ is the camera center.

The converse is trivial because given the camera center $C$, then $M(P + \lambda C) = MP \;\; \forall P \in \mathbb{P}^3$ (since $P + \lambda C$ is homogeneous coordinates is the viewing ray), thus $C \in RNS(M)$

# Camera Calibration

# Camera Calibration

Procedure to deduce the projection matrix $M = K\ [R, T]$ from a set of images.

Calibration procedure requires multiple correspondences between scene points (whose 3D location in known) and image points

$$\{p_i \ \leftrightarrow P_i, i = 1, \dots, N\}$$

Known calibration rigs can be employed to define the values of $P_i$ and to identify their location in the image, i.e. $p_i$

Once $M$ has been estimated, it is also necessary to estimate the intrinsic and extrinsic camera parameters (i.e., decompose M in $K\ [R, T]$ and extract at least the intrinsic parameters from $K$)

# Calibration Rig

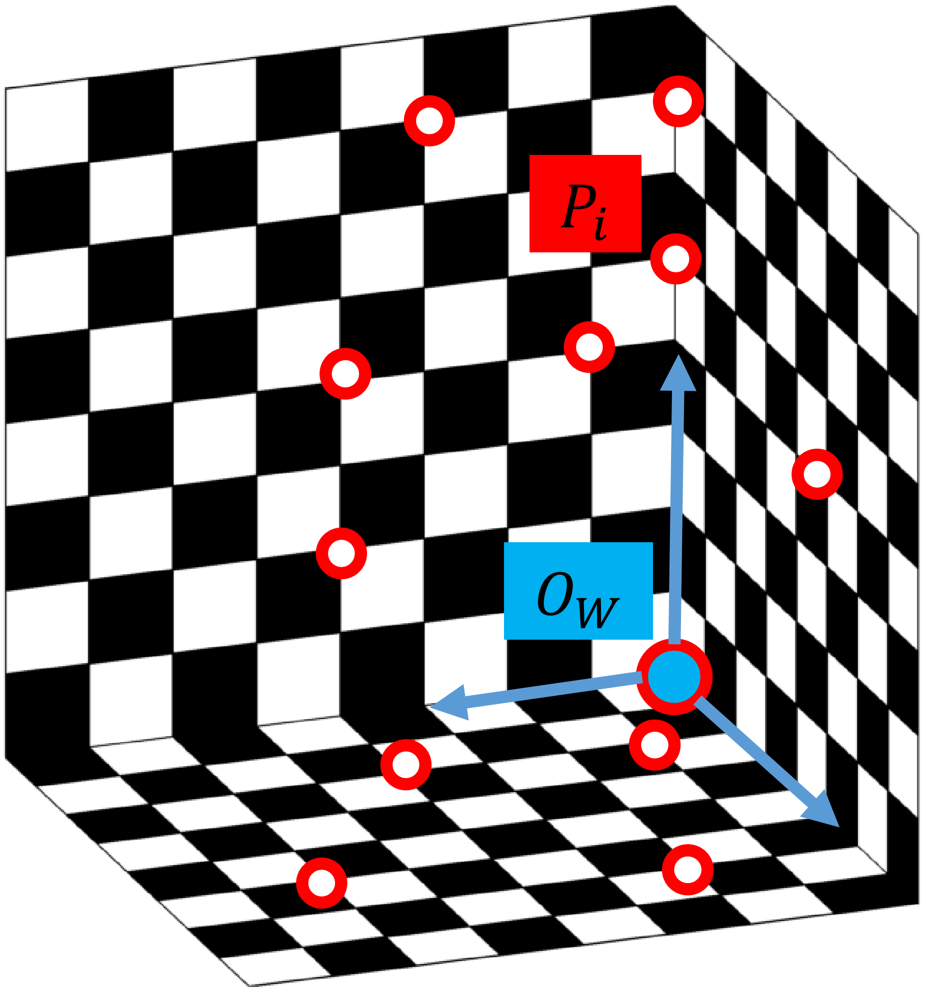


image from Brian moore81

These are 3 chekerboard set perpendicularly

The origin of the world reference system can be aribirarily set in the center $O_W$

The 3D position of all the other points can be exactly defined. These points have to be non-degenerate, i.e. do not have to lie on a plane.

# Calibration Rig: correspondences

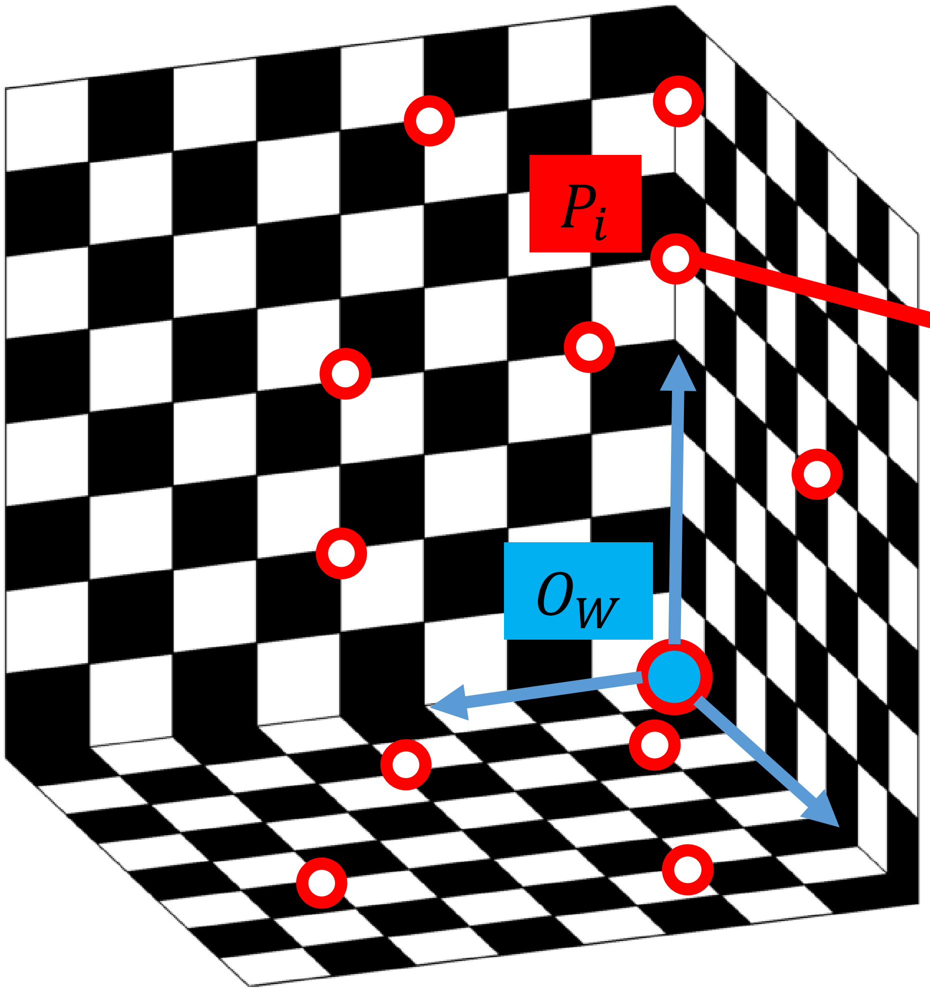
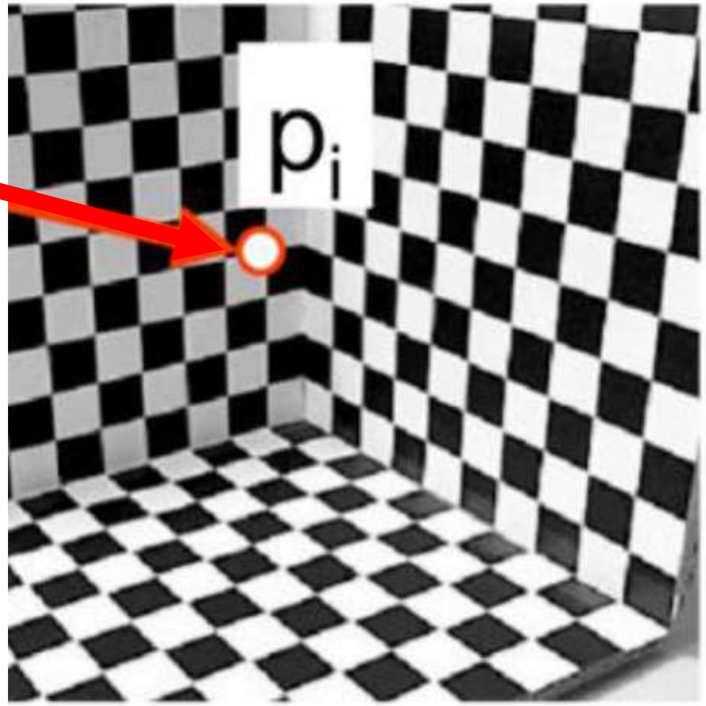



image from Brian moore81

Points can be extracted in the image and easily associated to their exact position in the 3D space, assuming the origin in the center of the calibration rig



# Estimating $M$ from 3D,2D correspondences

Calibration can be performed by solving a linear system on the few estimated correspondences $\{p_i \leftrightarrow P_i, i = 1, \dots, N\}$

$$p_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = MP_i = \begin{bmatrix} \boldsymbol{m_1} P_i \\ \boldsymbol{m_2} P_i \\ \boldsymbol{m_3} P_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{m_1} P_i / \boldsymbol{m_3} P_i \\ \boldsymbol{m_2} P_i / \boldsymbol{m_3} P_i \\ 1 \end{bmatrix}$$

Being $\boldsymbol{m_j}, j = 1, .., 3$ the lines of the matix $M \in \mathbb{R}^{3,4}$

Each correspondence gives 2 equations in the terms of $M$

$$\begin{cases} x_i \boldsymbol{m_3} P_i - \boldsymbol{m_1} P_i = 0 \\ y_i \boldsymbol{m_3} P_i - \boldsymbol{m_2} P_i = 0 \end{cases}$$

Thus, 6 points would be in principle enough to recover $M$

# Estimating $M$ from 3D,2D correspondences

Better to use more points to compensate for errors when estimating $p_i$

$$\begin{cases} x_1 \boldsymbol{m_3}\, P_1 - \boldsymbol{m_1}\, P_1 = 0 \\ y_1 \boldsymbol{m_3}\, P_1 - \boldsymbol{m_2}\, P_1 = 0 \\ \dots \\ y_N \boldsymbol{m_3}\, P_{\boldsymbol{N}} - \boldsymbol{m_2}\, P_N = 0 \end{cases}$$

Remarks

- when $\frac{N}{2} > 11$, the system is overdetermined
- This system admits the trivial solution $M = 0_{3,4}$
- For a nonzero solution $\widetilde{M}$, then $\forall k \in \mathbb{R}$ also $k\widetilde{M}$ is a solution

Regularization and constraints are needed

# Estimating $M$ from 3D,2D correspondences

If we write the system in a matrix form

$$\begin{cases} x_1 \boldsymbol{m_3}\, P_1 - \boldsymbol{m_1}\, P_1 = 0 \\ y_1 \boldsymbol{m_3}\, P_1 - \boldsymbol{m_2}\, P_1 = 0 \\ \dots \\ y_N \boldsymbol{m_3}\, P_{\boldsymbol{N}} - \boldsymbol{m_2}\, P_N = 0 \end{cases} = \begin{bmatrix} P_1' & \mathbf{0}' & -x_1 P_1' \\ \mathbf{0}' & P_1' & -y_1 P_1' \\ \dots & \dots & \dots \\ P_N{}' & \mathbf{0}' & -x_N P_1' \\ \mathbf{0}' & P_N{}' & -y_N P_1' \end{bmatrix} \begin{bmatrix} \boldsymbol{m_1}{}' \\ \boldsymbol{m_2}{}' \\ \boldsymbol{m_3}{}' \end{bmatrix} = \mathbf{0}_{2N}$$

$$\boldsymbol{Pm} = \mathbf{0}_{2N}$$

Where $\mathbf{0}'$ is $[0,0,0,0]$ , $\boldsymbol{P} \in \mathbb{R}^{2N\times 12}, \boldsymbol{m} \in \mathbb{R}^{12}$

$$\widetilde{\boldsymbol{m}} = \underset{\boldsymbol{m}}{\operatorname{argmin}} \|\boldsymbol{Pm}\|_{\boldsymbol{2}}$$

Subject to

$$\|\boldsymbol{m}\|_2 = 1$$

# Solving this linear system

This system can be solved trough the SVD of $\boldsymbol{P}$

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}'$$

Then, $\boldsymbol{m}$ corresponds to the last column of $\boldsymbol{V}$ (we will see later this is solved via the DLT algorithm, check 592 of HZ)

This gives the matrix $M$

Rmk: We still need to compute the intrinsic parameters out of $M$, namely we need to factorize $M$ as $M = K[R, T]$

This is possible, but not shown here.

# Obviously…

This calibration «holds» as long as the camera does not move.

If the camera moves the extrinsic parameters have to be computed from scratches. Intrinsic parameters holds.

If the camera changes its focus, then the intrinsic parameters have to be updated accordingly or computed from scratches.

This is possible as long as the points $P_i$ do not lie on the same plane, otherwise we get into degenerate conditions

# Once Calibrated

Sometimes we are interested in factorizing $M$ as

$$M = K[R, T]$$

since when the camera moves $K$ does not change and does not need to be estimated from scratches.

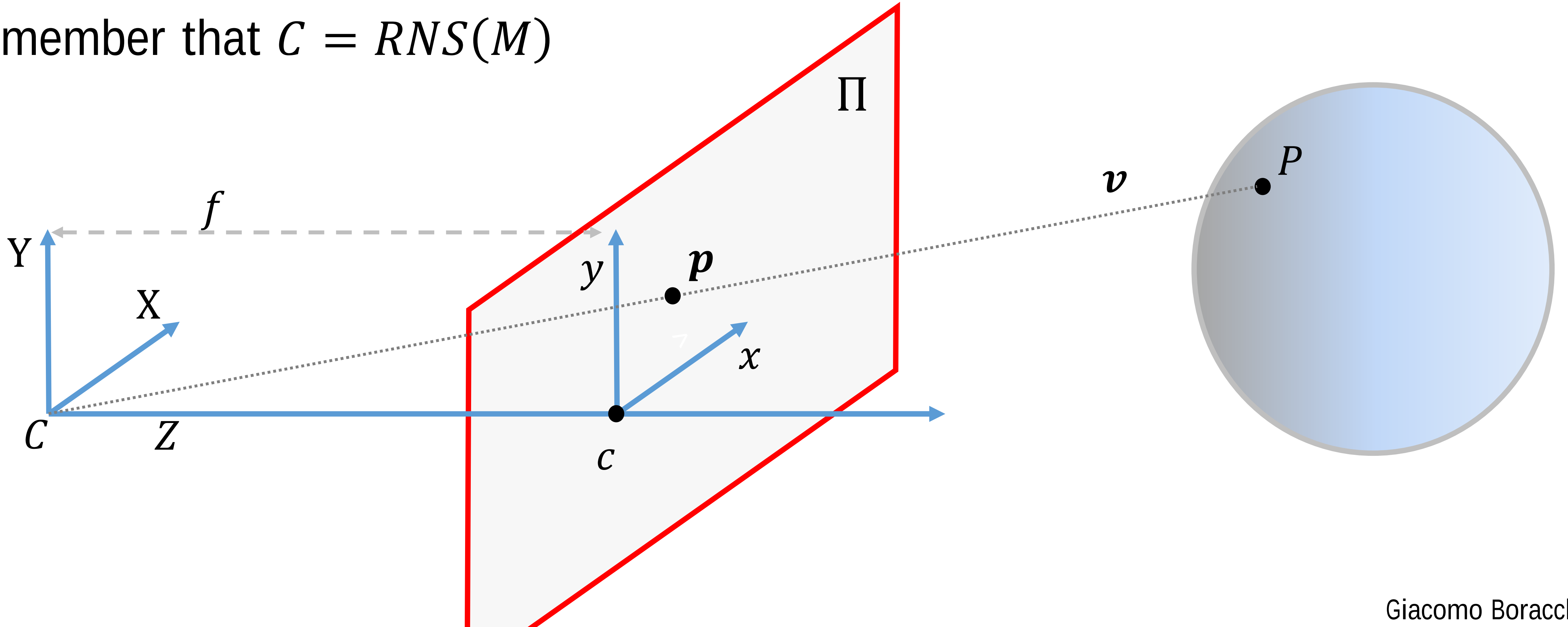
This is possible but not reported here

# Once Calibrated

We can associate to each point on the image, a viewing ray

$$\boldsymbol{v} = C + \lambda \begin{bmatrix} (KR)^{-1}\boldsymbol{p} \\ 0 \end{bmatrix}$$

Remember that $C = RNS(M)$



# Once Calibrated

We can associate to each point on the image, a viewing ray

$$\boldsymbol{v} = C + \lambda \begin{bmatrix} (KR)^{-1}\boldsymbol{p} \\ 0 \end{bmatrix}$$

In fact

$$M\boldsymbol{v} = M\left(C + \lambda \begin{bmatrix} (KR)^{-1}\boldsymbol{p} \\ 0 \end{bmatrix}\right)$$

$$M\boldsymbol{v} = MC + \lambda K[R,T] \begin{bmatrix} (KR)^{-1}\boldsymbol{p} \\ 0 \end{bmatrix}$$

$$M\boldsymbol{v} = \lambda[KR, KT] \begin{bmatrix} (KR)^{-1}\boldsymbol{p} \\ 0 \end{bmatrix}$$

$$M\boldsymbol{v} = \lambda\boldsymbol{p} = \boldsymbol{p} \quad \forall\lambda$$