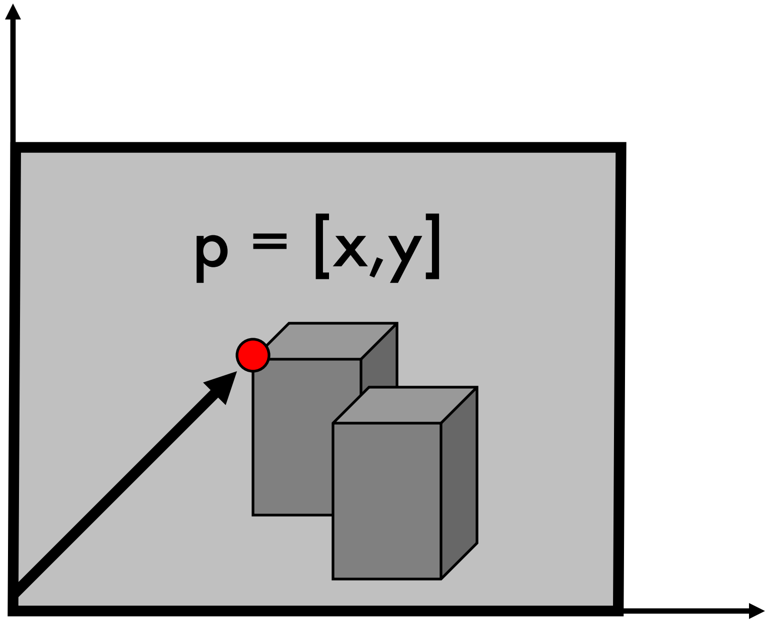
Exteroceptive Sensors

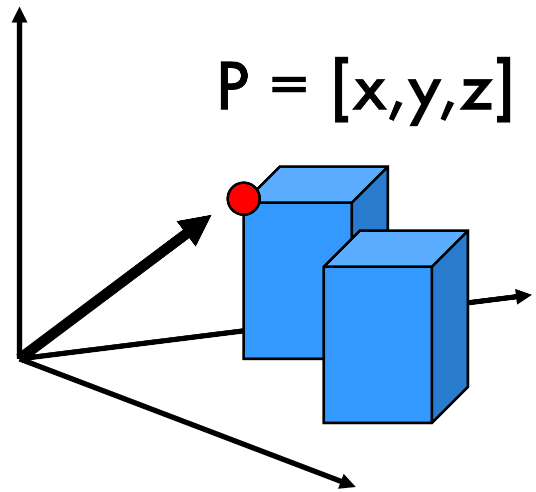
Agenda

1. Basics definitions and properties
2. Geometrical transformations
3. Application: removing perspective distortion – the DLT algorithm

Vectors (i.e., 2D or 3D vectors)

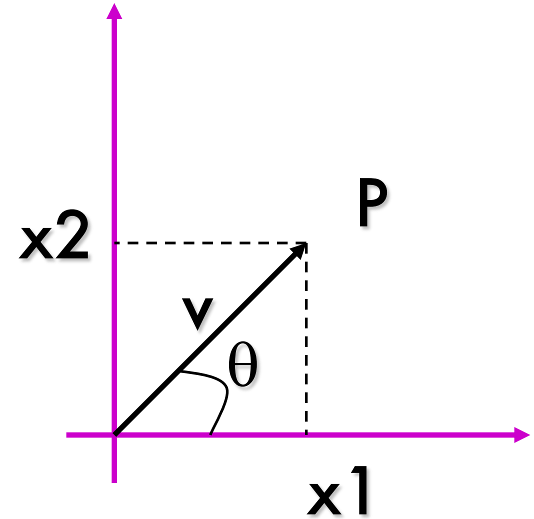


Image



3D world

Vectors (i.e., 2D vectors)



$$\mathbf{v} = (x_1, x_2)$$

Magnitude: $\| \mathbf{v} \| = \sqrt{x_1^{\ 2} + x_2^{\ 2}}$

If $\| \mathrm{v} \| = 1$, $\mathbf{v}$ Is a UNIT vector

$$\frac{\mathbf{v}}{\| \mathbf{v} \|} = \left(\frac{x_1}{\| \mathbf{v} \|}, \frac{x_2}{\| \mathbf{v} \|} \right)$$ Is a unit vector

Orientation: $\theta = \tan^{-1}\left(\frac{x_2}{x_1} \right)$

Matrices

$$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ a_{31} & a_{32} & \cdots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & \boxed{a_{nm}} \end{bmatrix} \quad \longleftrightarrow$$

Pixel's intensity value

Block forms

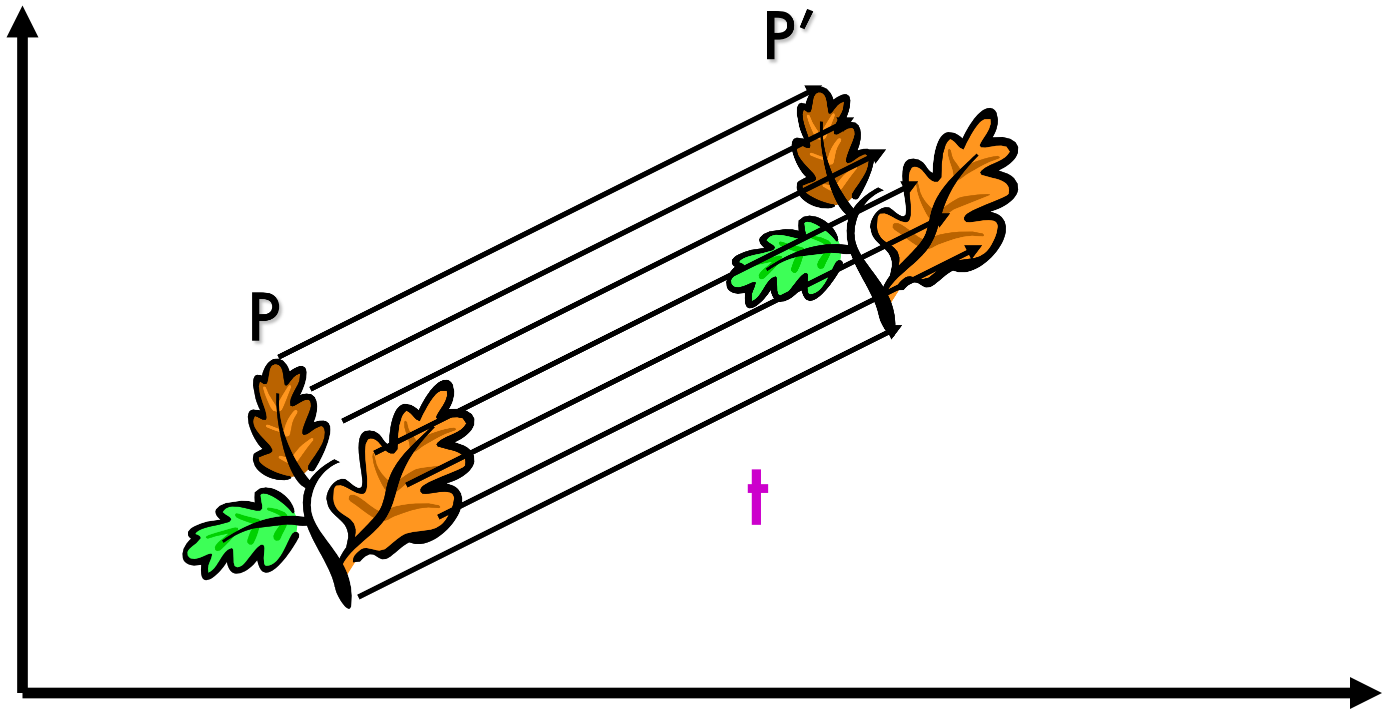
$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

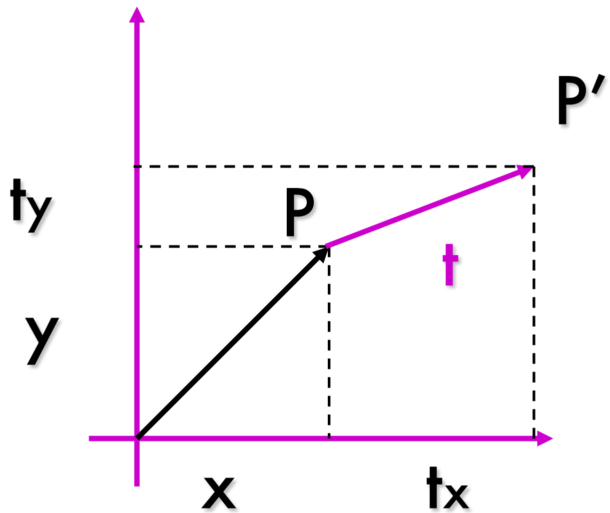
$$t = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

2D Geometrical Transformations

2D Translation



2D Translation Equation

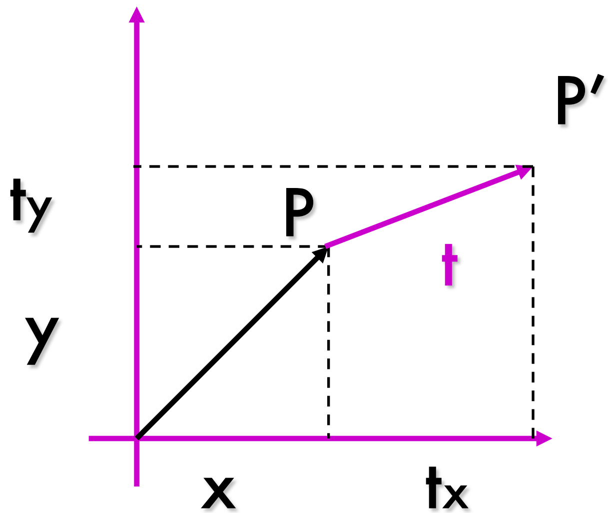


$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{t} = (x + t_x, y + t_y)$$

2D Translation using Matrices



$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Multiply the coordinates by a non-zero scalar and add an extra coordinate equal to that scalar. For example,

$$(x, y) \to (x \cdot z, y \cdot z, z) \quad z \neq 0$$

$$(x, y, z) \to (x \cdot w, y \cdot w, z \cdot w, w) \quad w \neq 0$$

Back to Cartesian Coordinates:

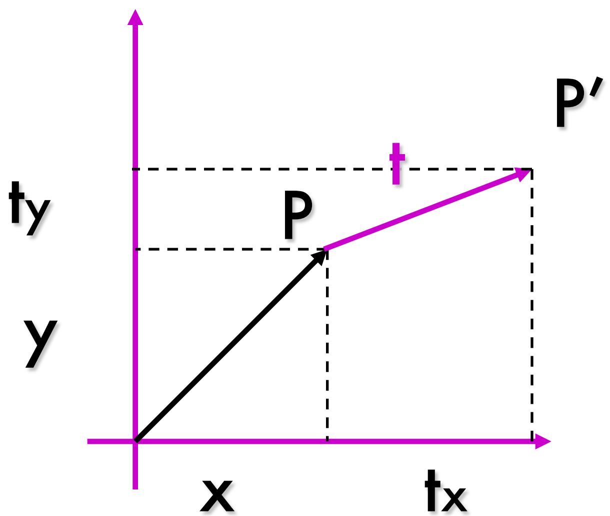
- Divide by the last coordinate and eliminate it. For example,

$$(x, y, z) \quad z \neq 0 \rightarrow (x / z, y / z)$$

$$(x, y, z, w) \quad w \neq 0 \rightarrow (x / w, y / w, z / w)$$

- NOTE: in our example the scalar was 1

2D Translation using Homogeneous Coordinates



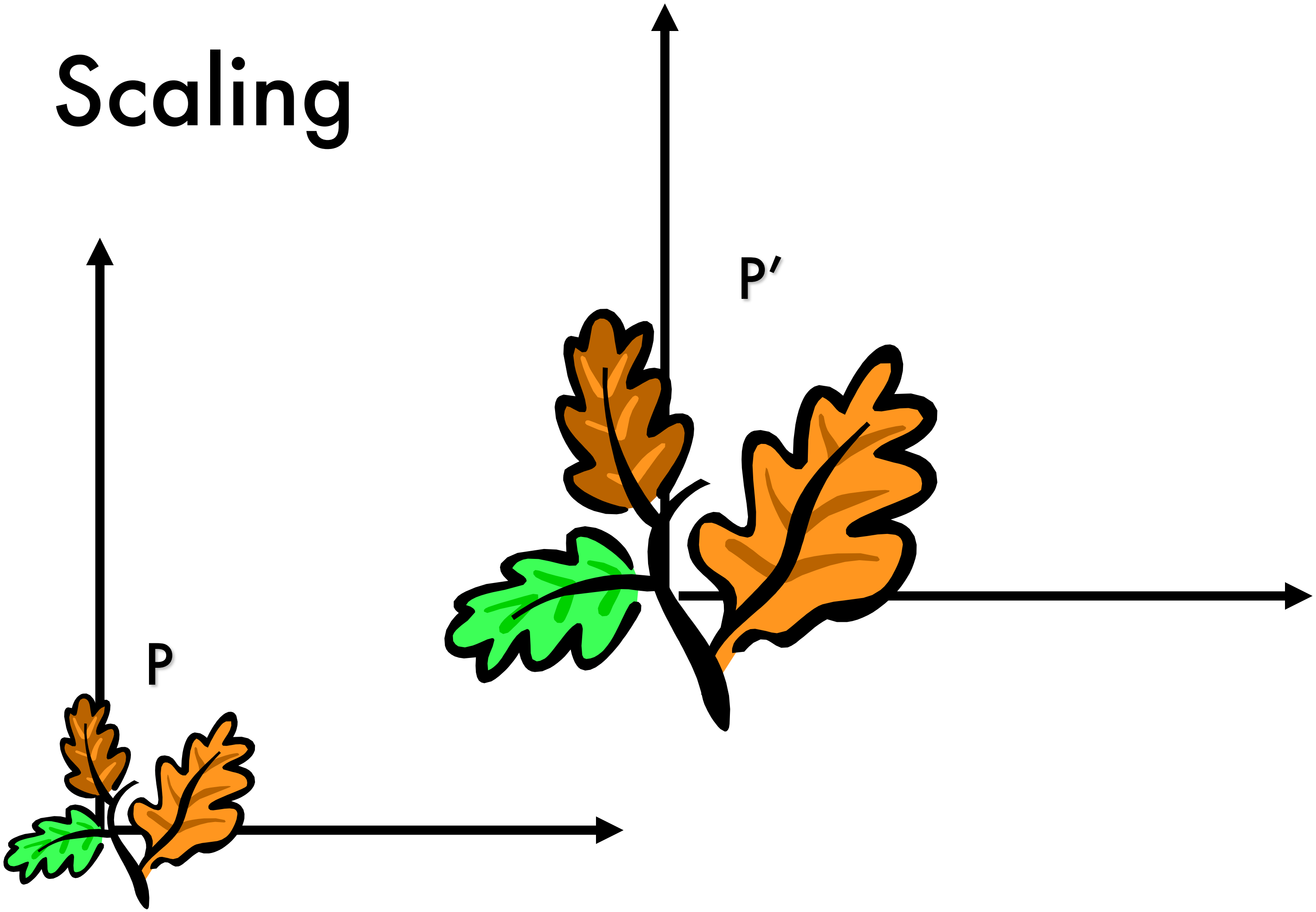
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

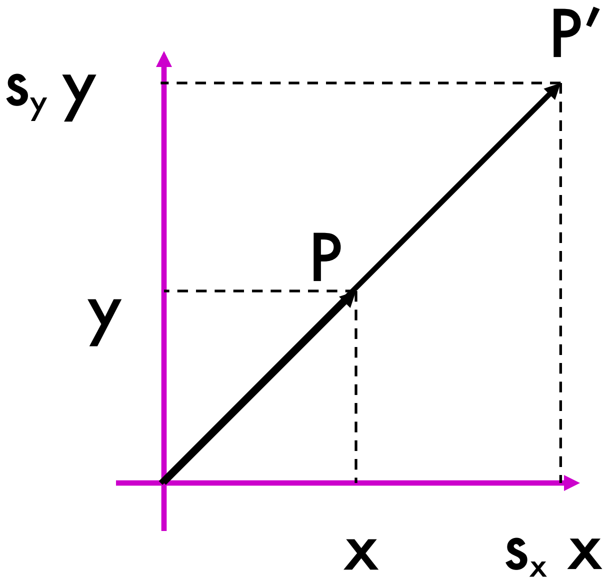
$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P}$$

Scaling



Scaling Equation



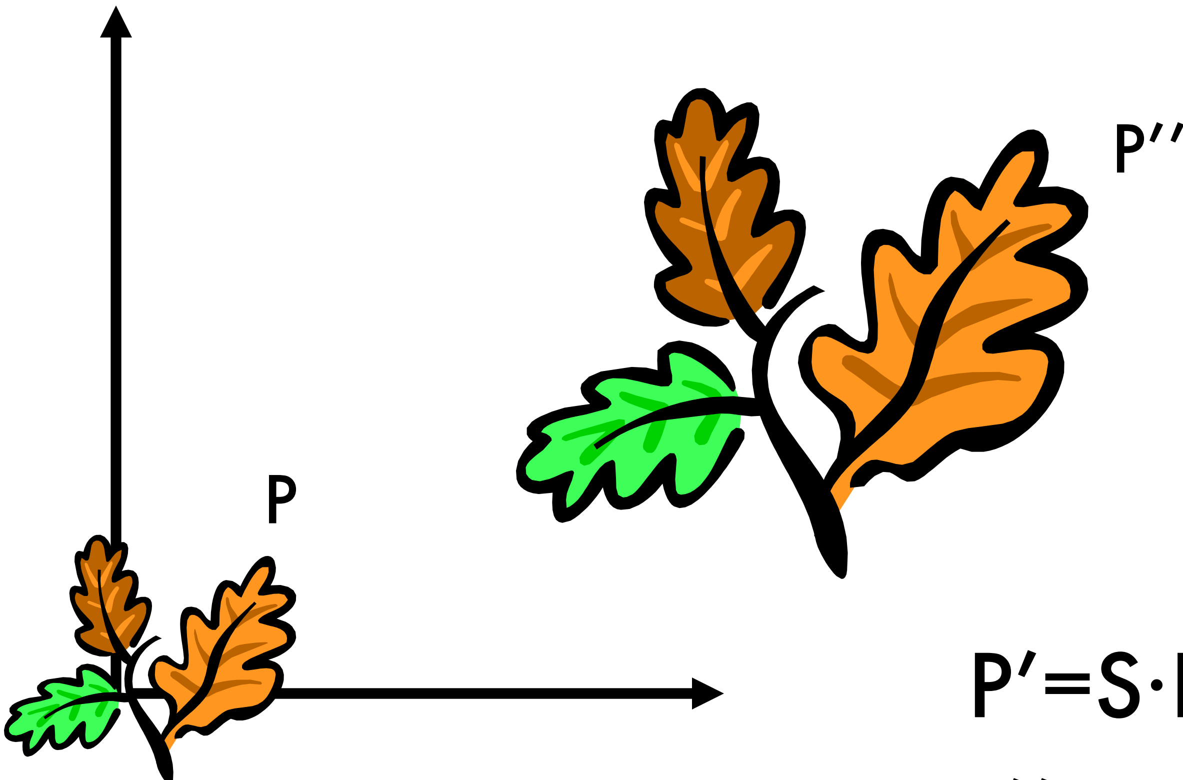
$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

Scaling & Translating



$$P' = S \cdot P$$

$$P'' = T \cdot P'$$

$$P'' = T \cdot P' = T \cdot (S \cdot P) = T \cdot S \cdot P = A \cdot P$$

Scaling & Translating

$$\mathbf{P''} = \mathbf{T}\cdot\mathbf{S}\cdot\mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{A}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

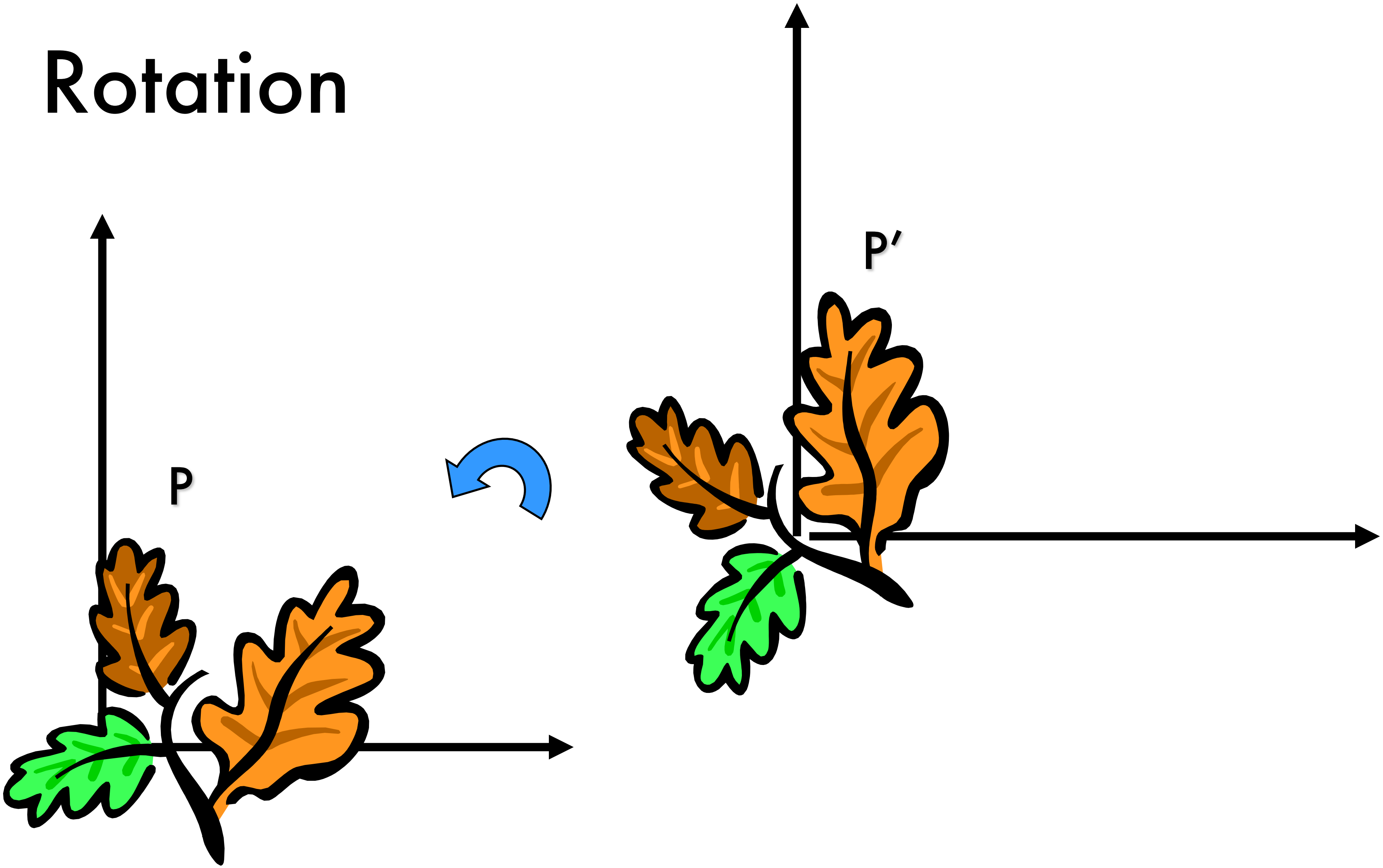
Translating & Scaling = Scaling & Translating ?

$$\mathbf{P}''' = \mathbf{T}\cdot\mathbf{S}\cdot\mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$\mathbf{P}''' = \mathbf{S}\cdot\mathbf{T}\cdot\mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

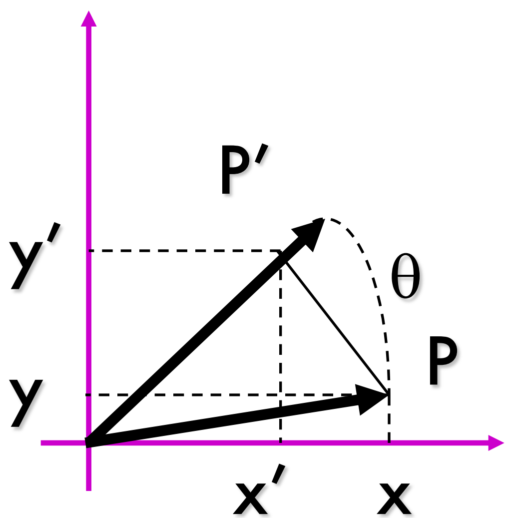
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Rotation



Rotation Equations

Counter-clockwise rotation by an angle θ



$$x' = \cos\theta \; x - \sin\theta \; y$$

$$y' = \cos\theta \; y + \sin\theta \; x$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{R} \;\; \mathbf{P}$$

Degrees of Freedom

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

R is 2x2 $\longrightarrow$ 4 elements

Note: R is an orthogonal matrix and satisfies many interesting properties:

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

Translation + Rotation+ Scaling

P′= (T R S) P

$$\mathbf{P}' = \mathbf{T}\cdot \mathrm{R}\cdot \mathbf{S}\cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} R' & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} R'S & t \\ 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

If $s_x = s_y$, this is a similarity transformation!

Transformation in 2D

-Isometries

-Similarities

-Affinity

-Projective

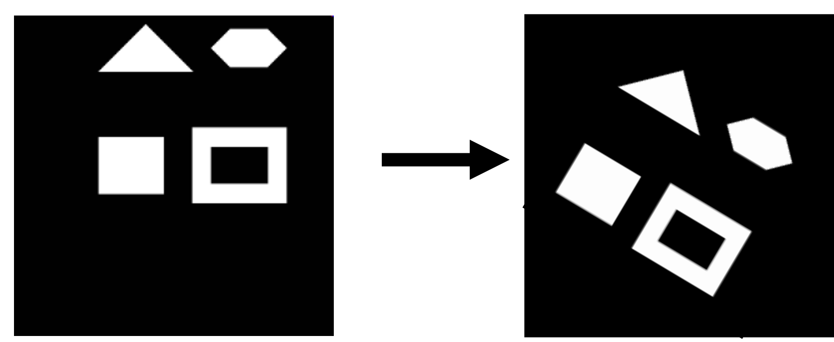
Transformation in 2D

Isometries:

[Euclideans]

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_e \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve distance (areas)
- 3 DOF
- Regulate motion of rigid object

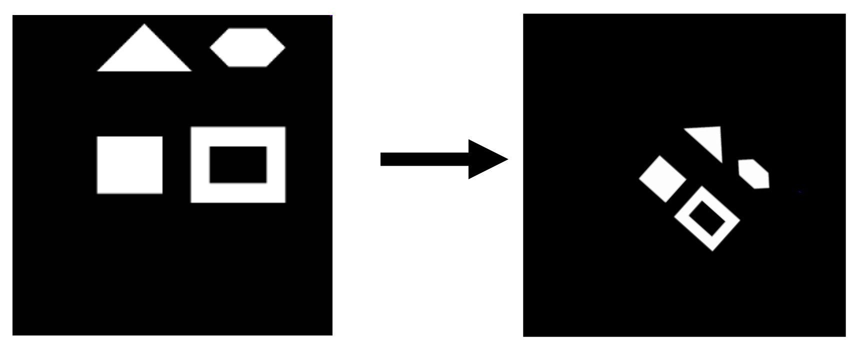


Transformation in 2D

Similarities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\,R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve
    - ratio of lengths
    - angles
- 4 DOF

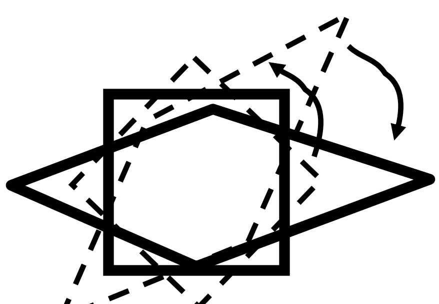


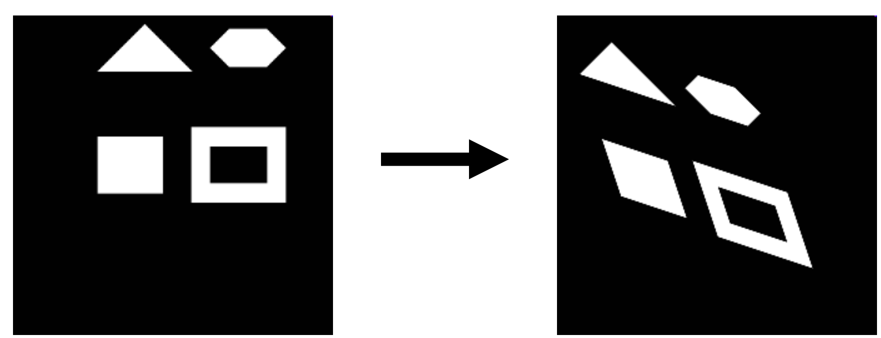
Transformation in 2D

Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \qquad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$





Transformation in 2D

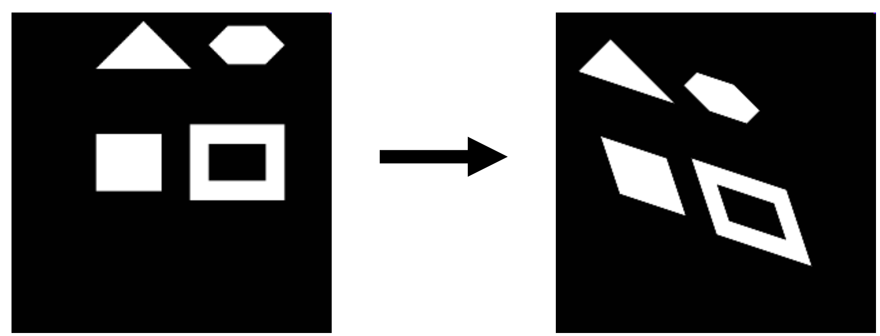
Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \qquad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

-Preserve:
- Parallel lines
- Ratio of areas
- Ratio of lengths on collinear lines
- others...

- 6 DOF

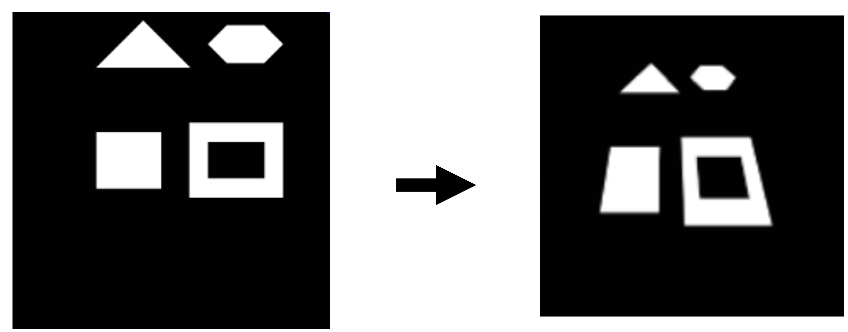


Transformation in 2D

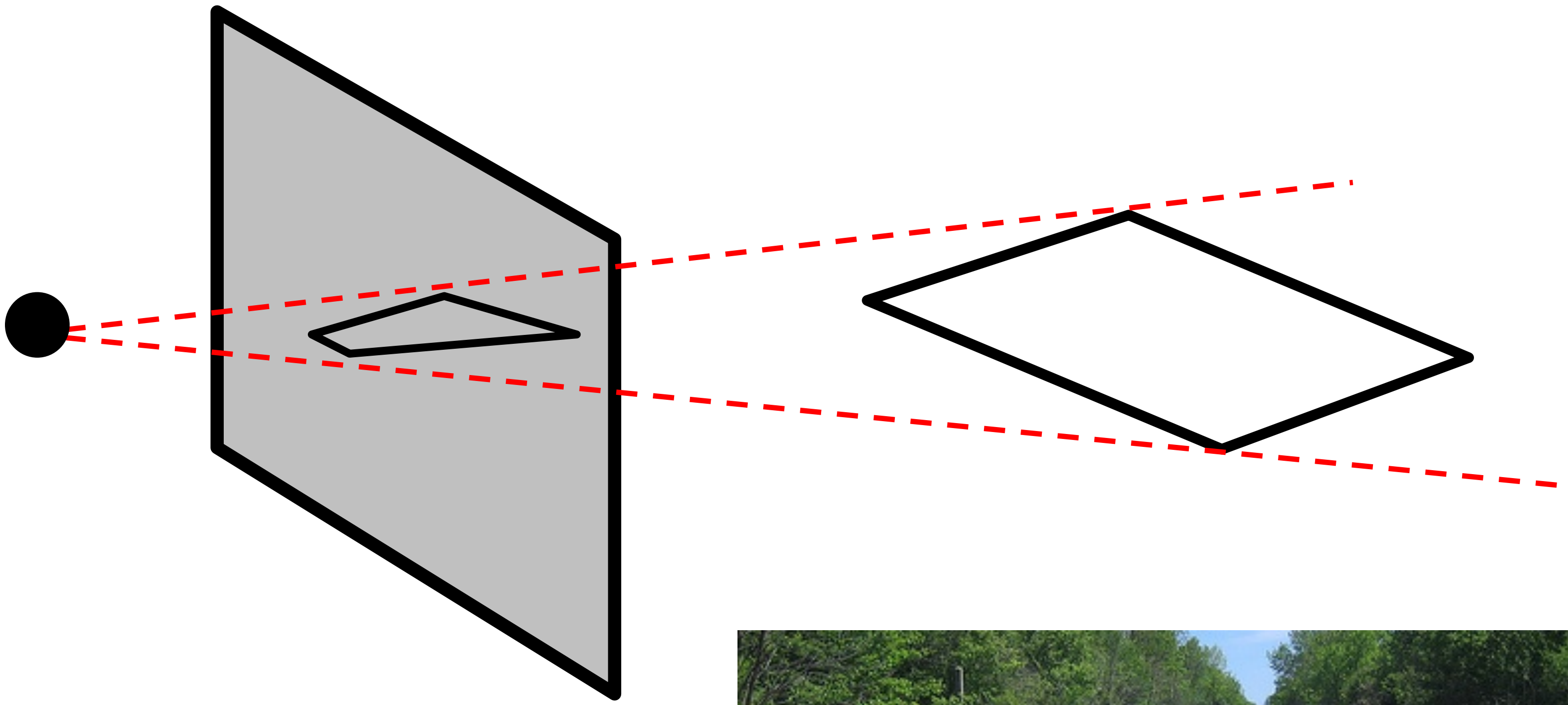
Projective:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ \boxed{v} & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_p \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

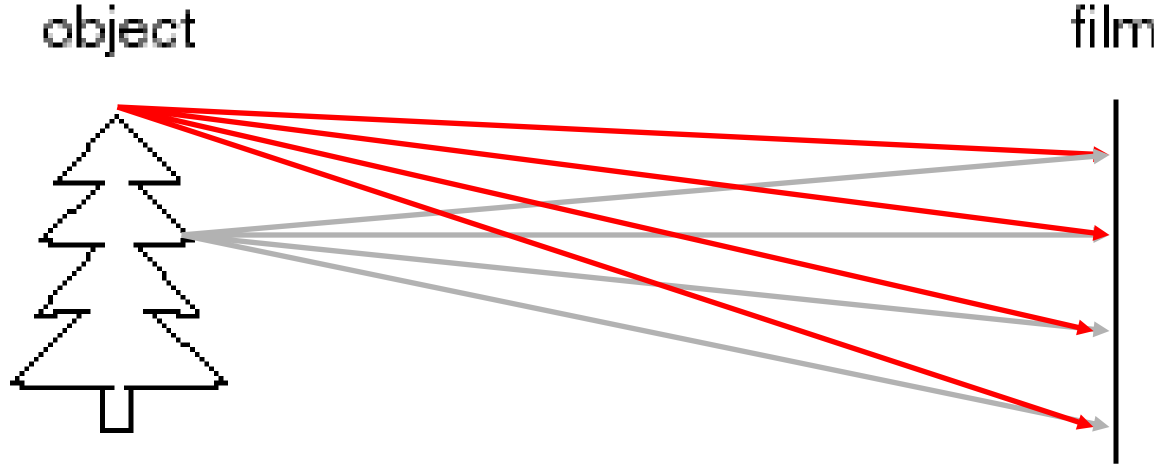
- 8 DOF
- Preserve:
    - cross ratio of 4 collinear points
    - collinearity
    - and a few others...



Transformation in 2D

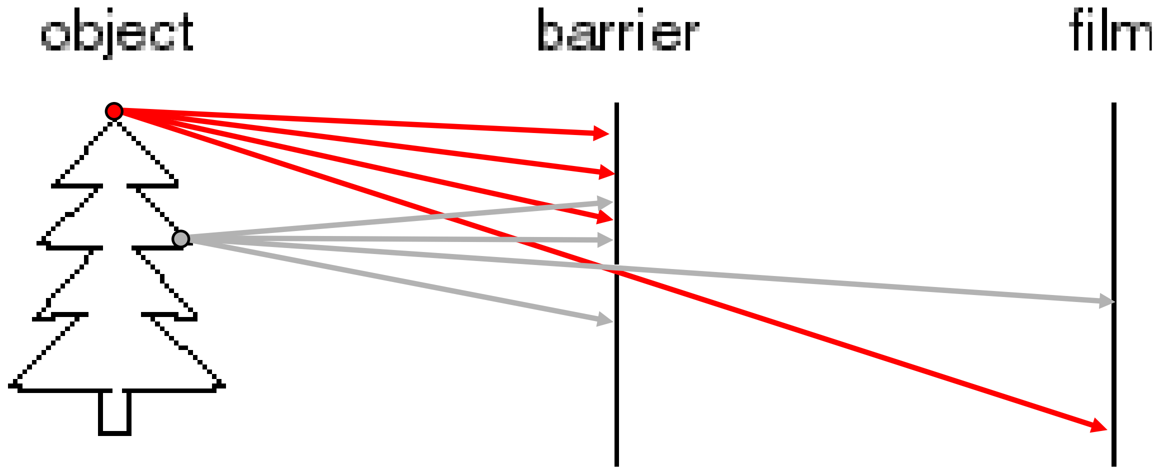


How do we see the world?



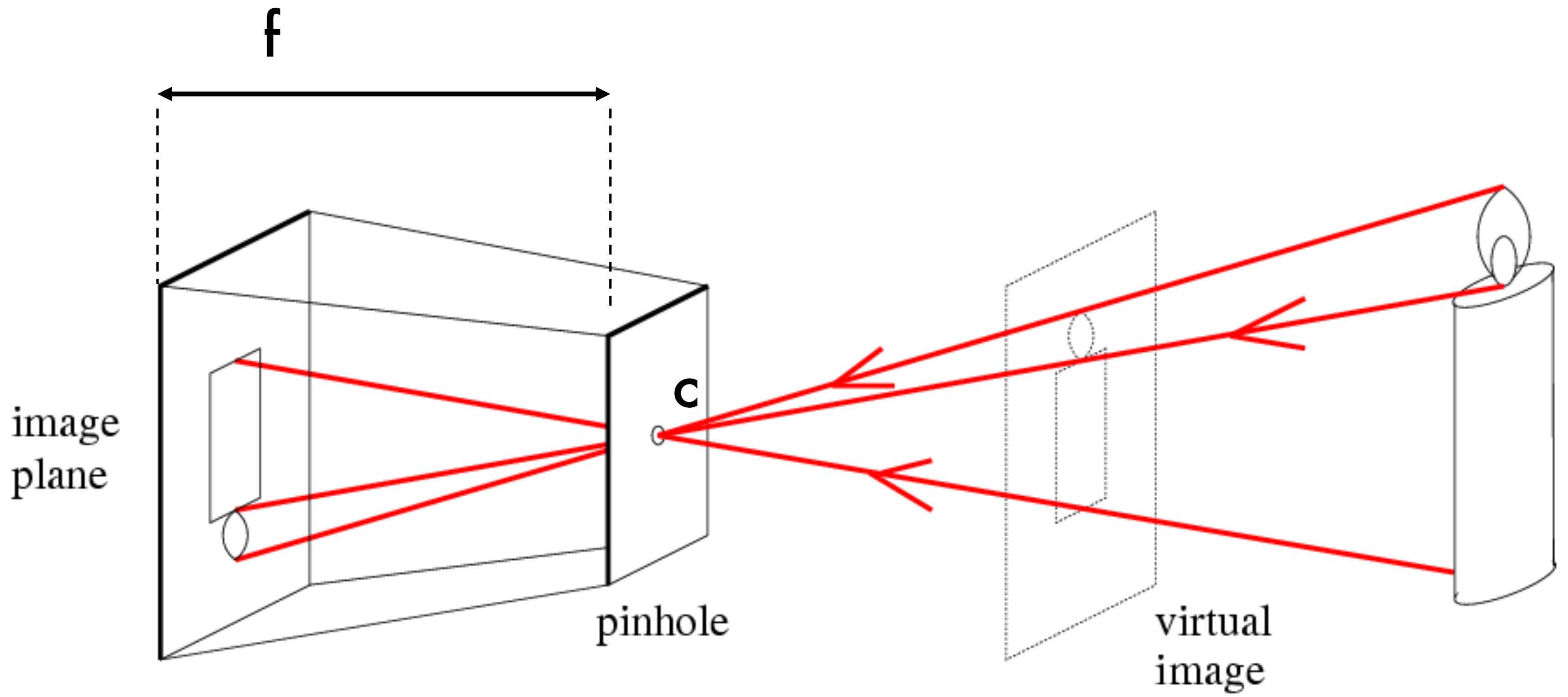
- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

Pinhole camera



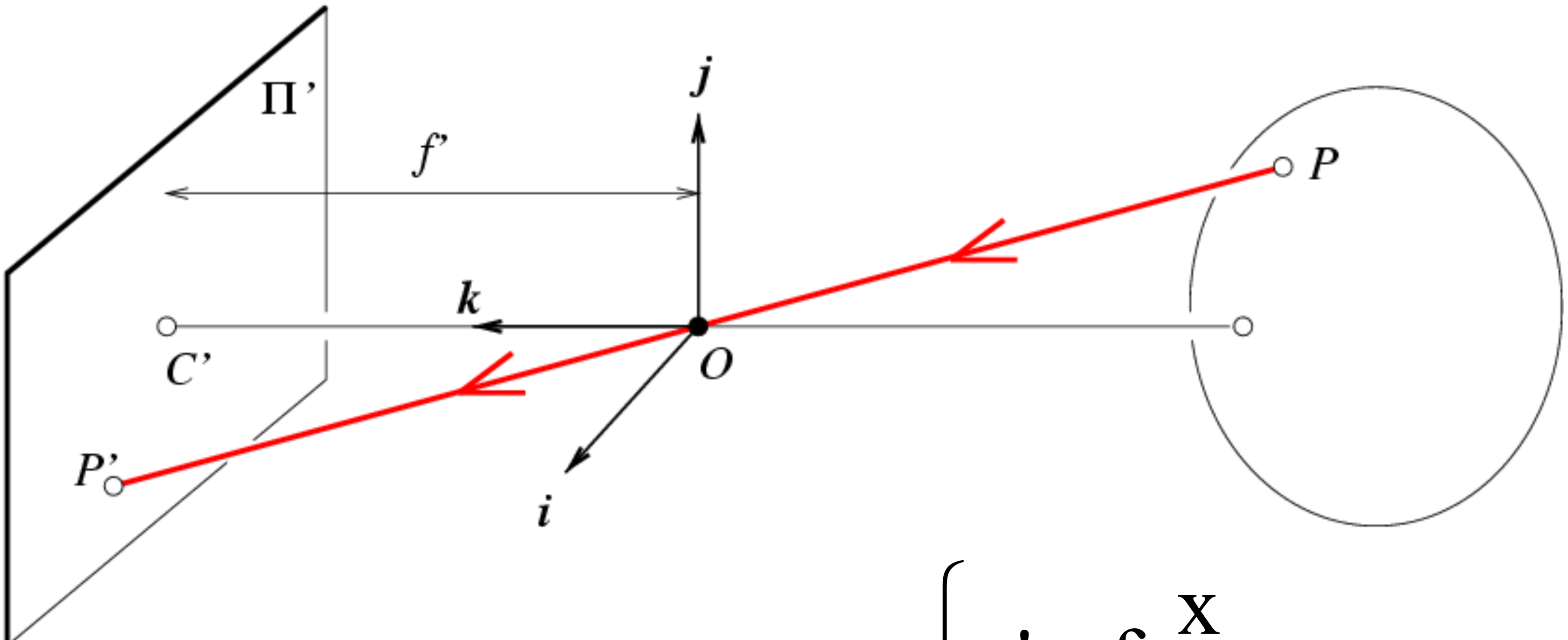
- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**

Pinhole camera



f = focal length
c = aperture = pinhole = center of the camera

Pinhole camera

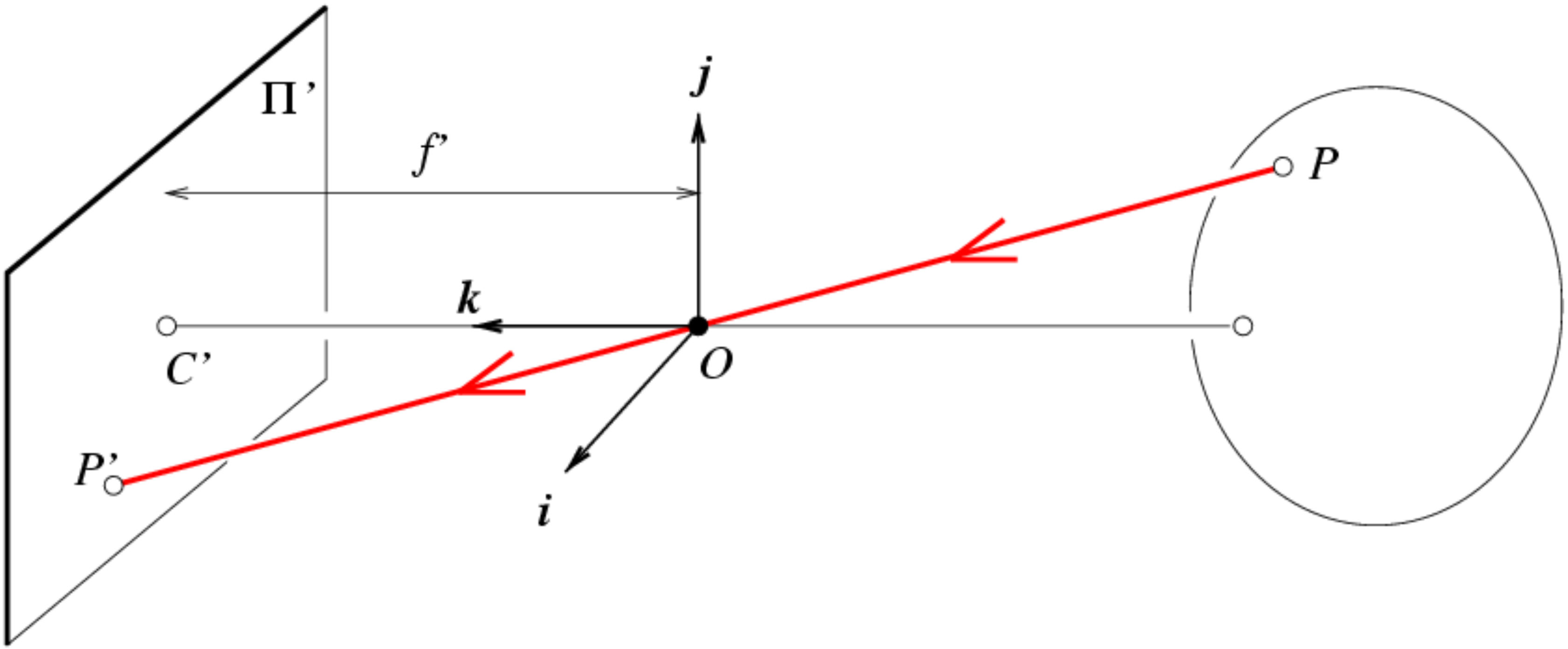


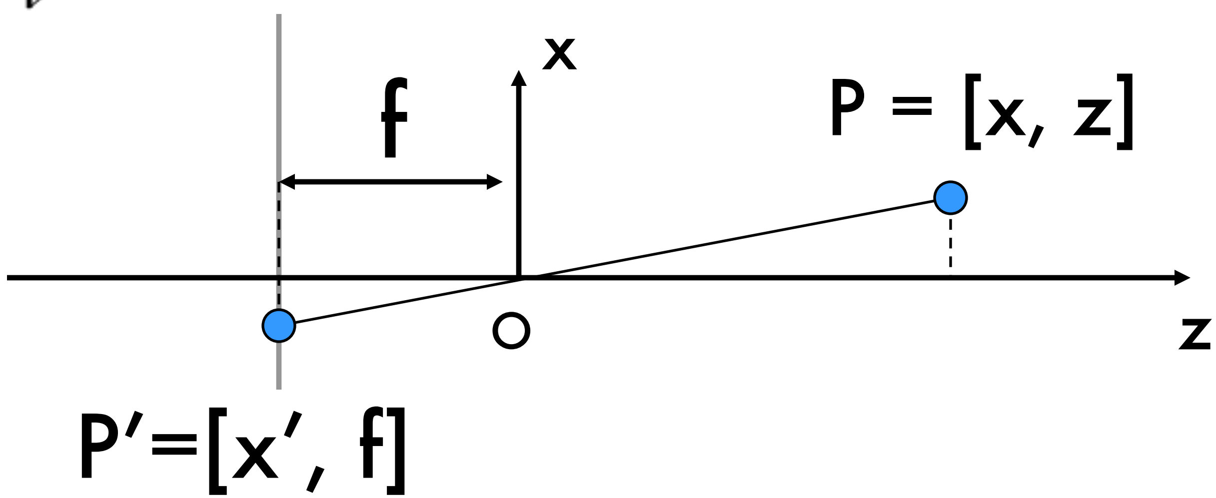
$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{cases} x' = f\,\dfrac{x}{z} \\[2ex] y' = f\,\dfrac{y}{z} \end{cases}$$

Derived using similar triangles

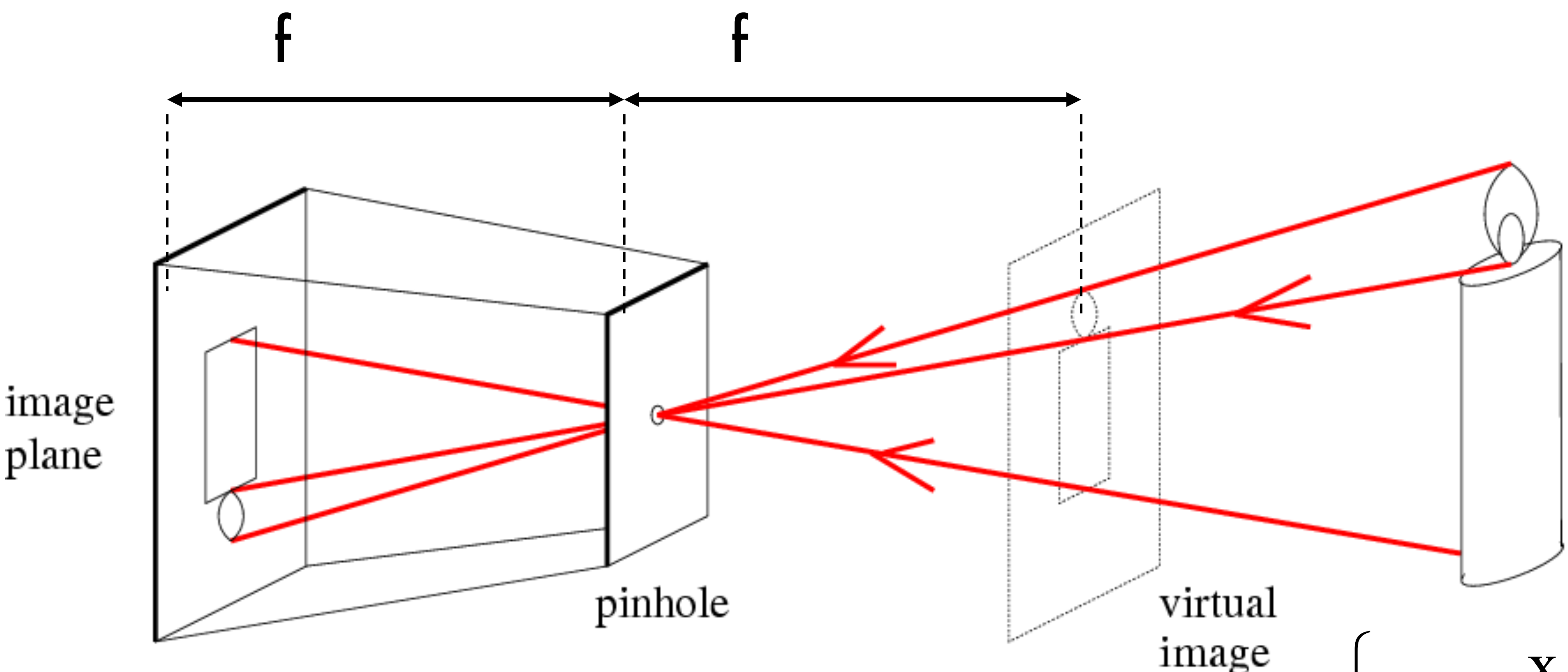
Pinhole camera





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

Pinhole camera



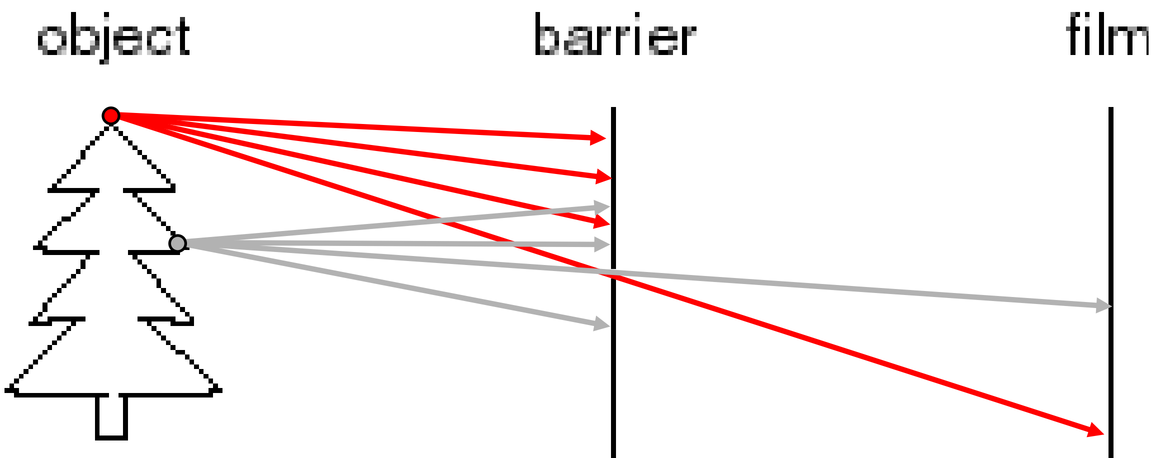
$$\begin{cases} x' = f\,\dfrac{x}{z} \\[2ex] y' = f\,\dfrac{y}{z} \end{cases}$$

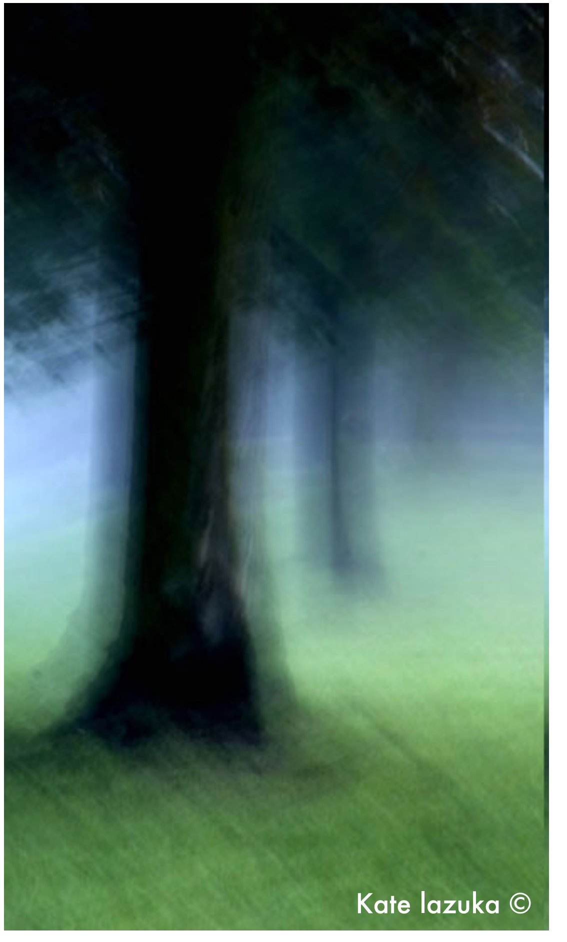
Common to draw image plane *in front* of the focal point.

What's the transformation between these 2 planes?

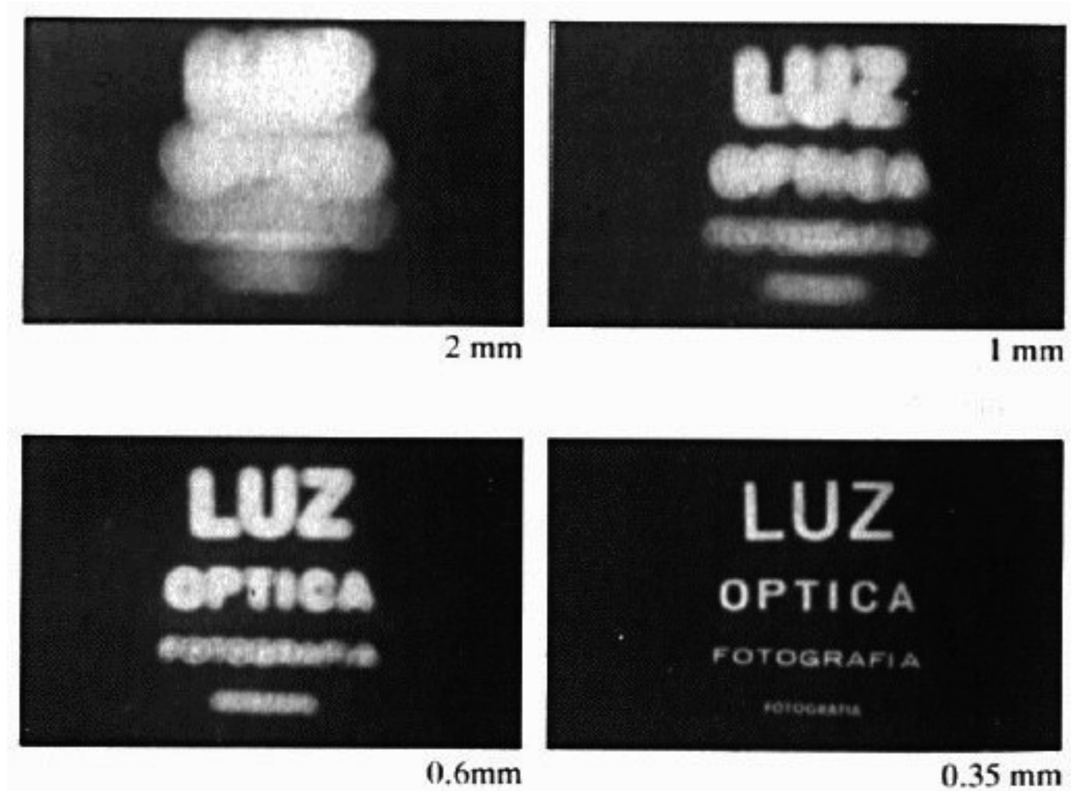
Pinhole camera

Is the size of the aperture important?





Shrinking
aperture
size

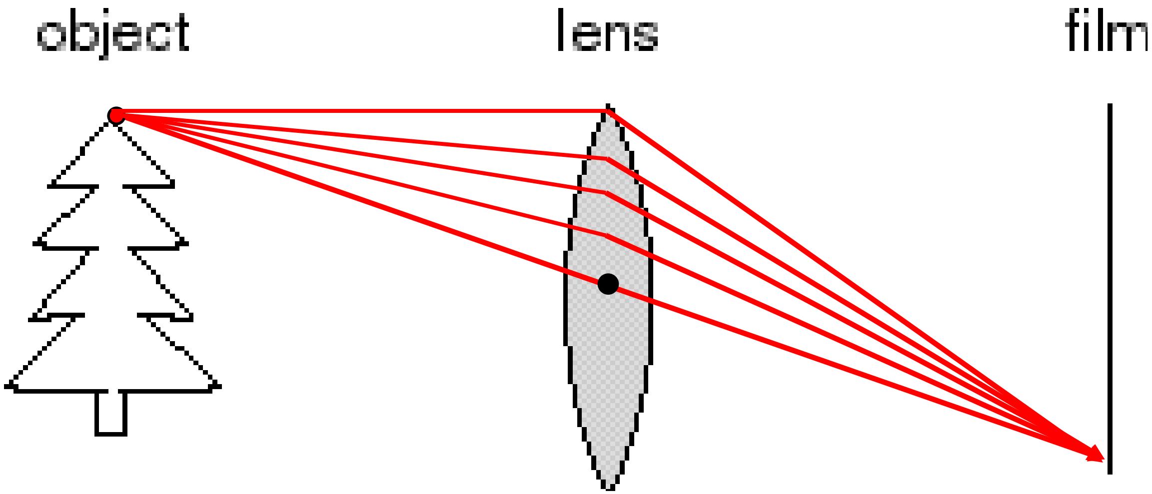


-Why the aperture cannot be too small?

- -Less light passes through
- -Diffraction effect

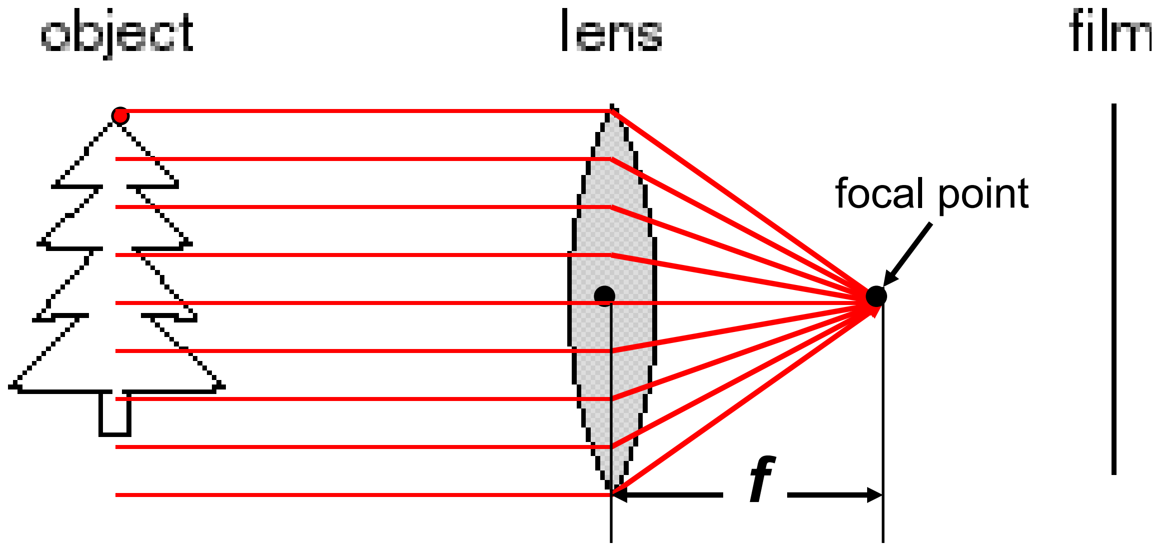
Adding lenses!

Cameras & Lenses



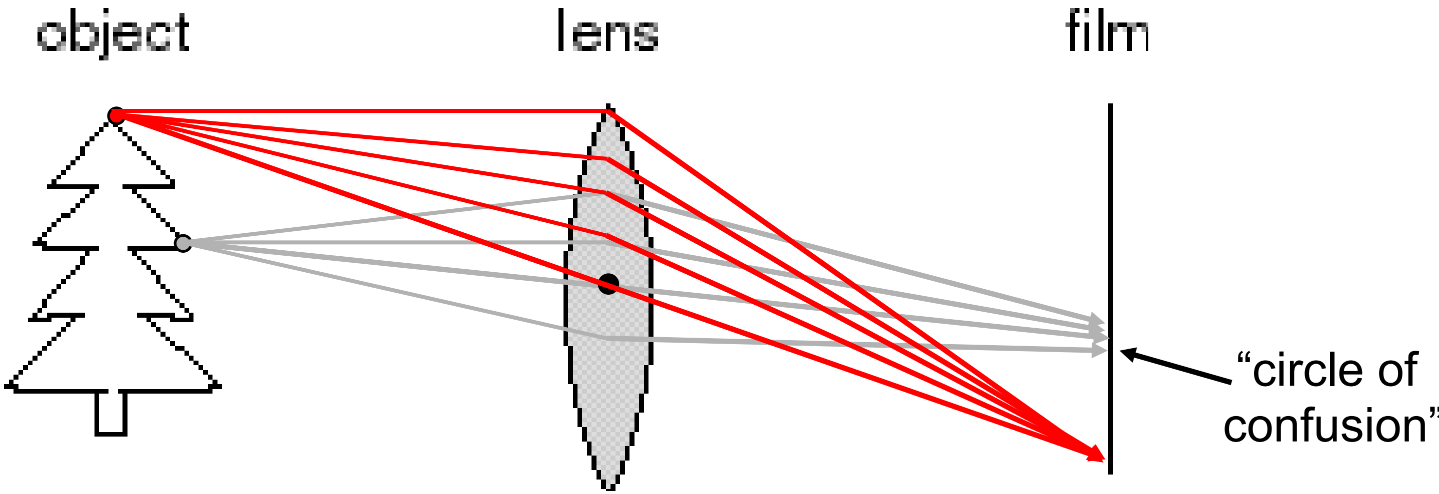
- A lens focuses light onto the film

Cameras & Lenses



- A lens focuses light onto the film
  - Rays passing through the center are not deviated
  - All parallel rays converge to one point on a plane located at the *focal length f*

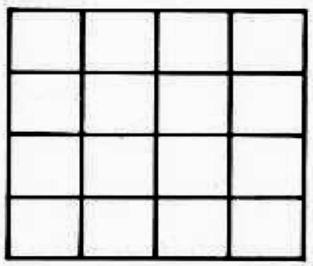
Cameras & Lenses



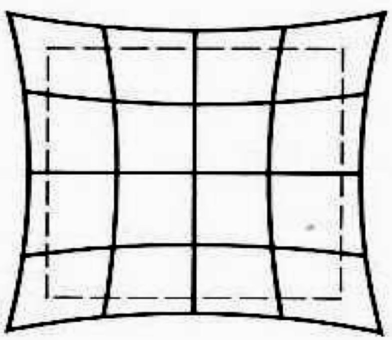
- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
  - Related to the concept of depth of field

Issues with lenses: Radial Distortion

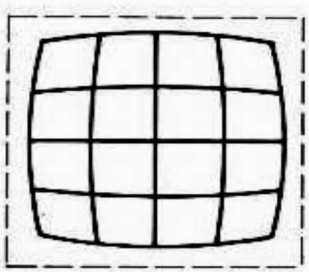
– Deviations are most noticeable for rays that pass through the edge of the lens



No distortion



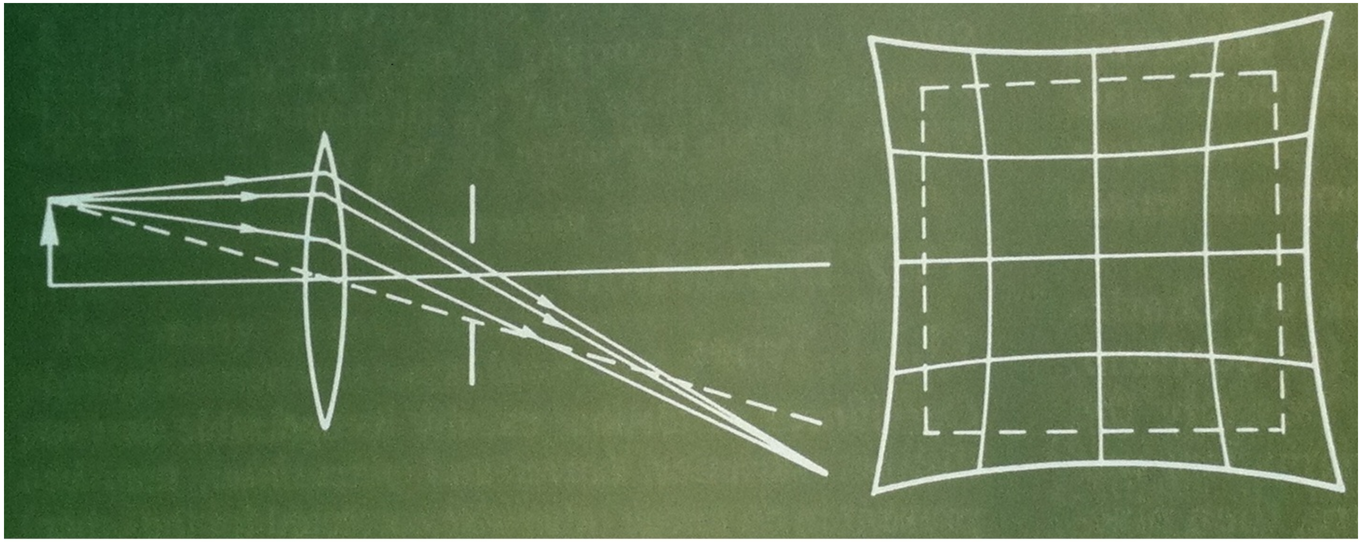
Pin cushion



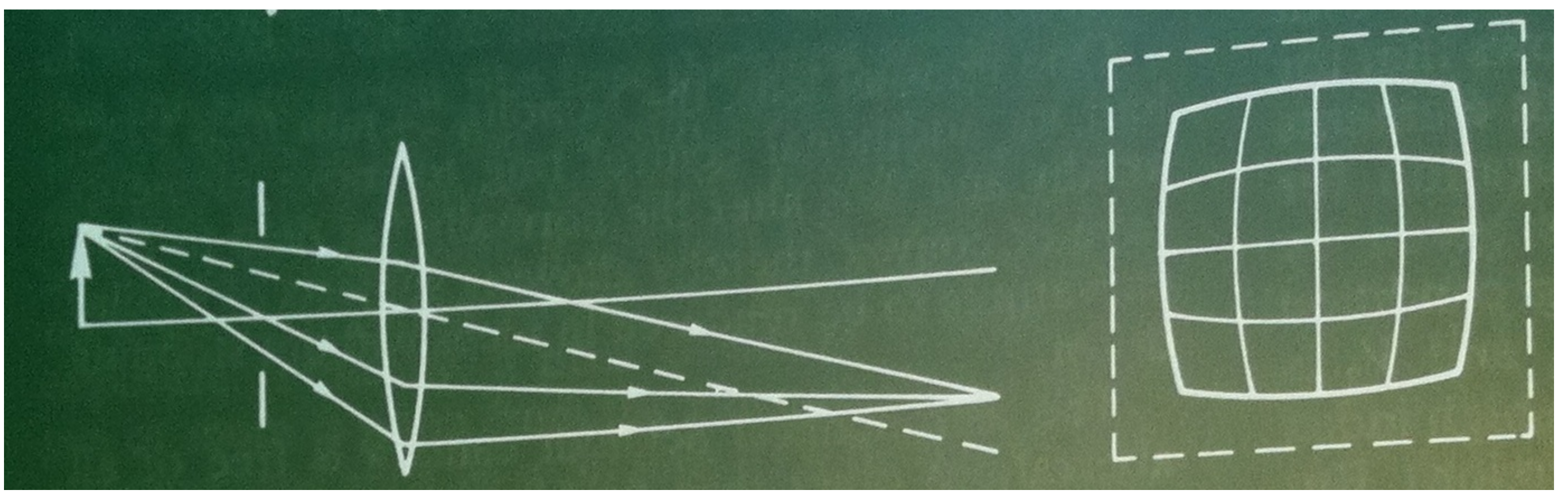
Barrel (fisheye lens)

Image magnification decreases with distance from the optical axis

Issues with lenses: Radial Distortion

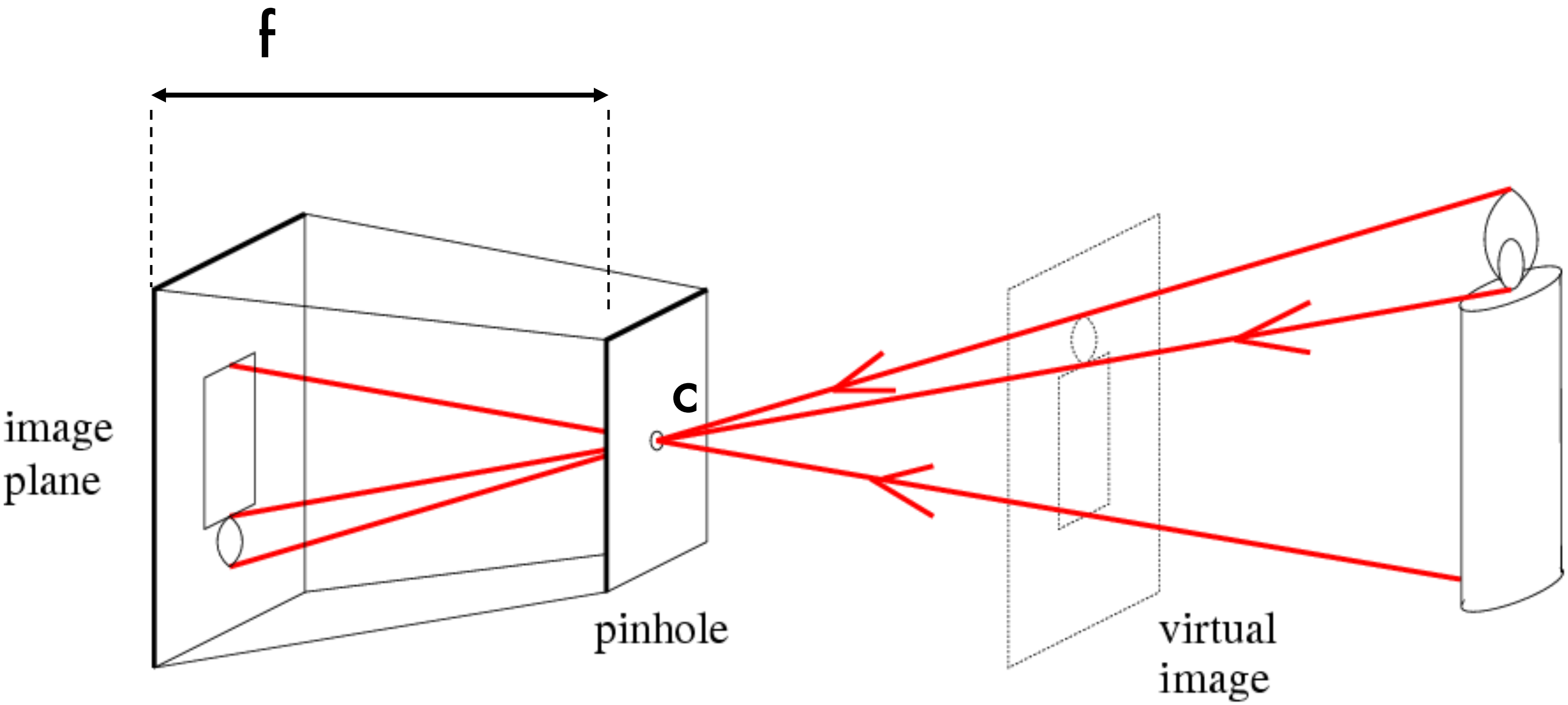


Pin cushion



Barrel (fisheye lens)

Pinhole camera

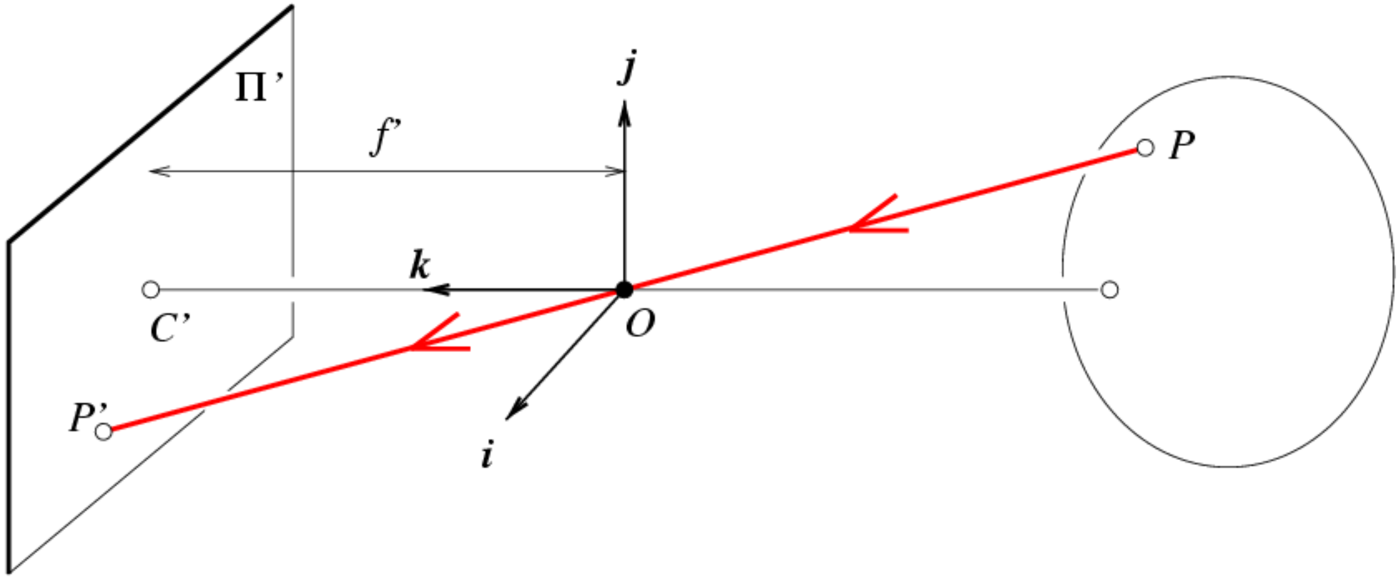


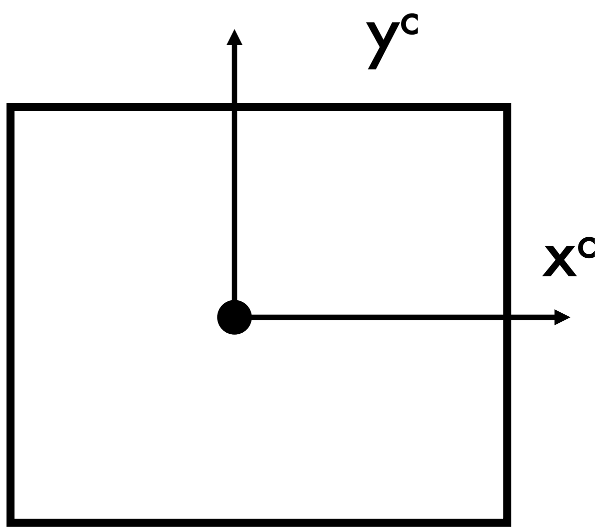
$$(x, y, z) \rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

f = focal length
c = center of the camera

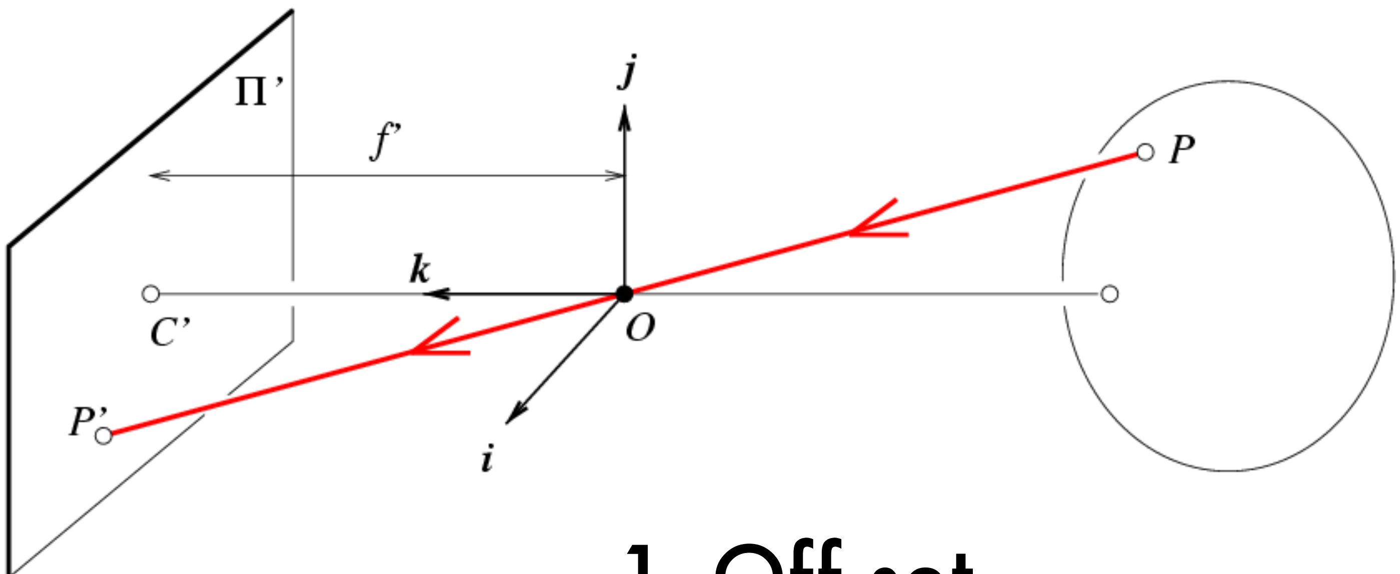
$$\Re^3 \xrightarrow{E} \Re^2$$

Coordinate systems

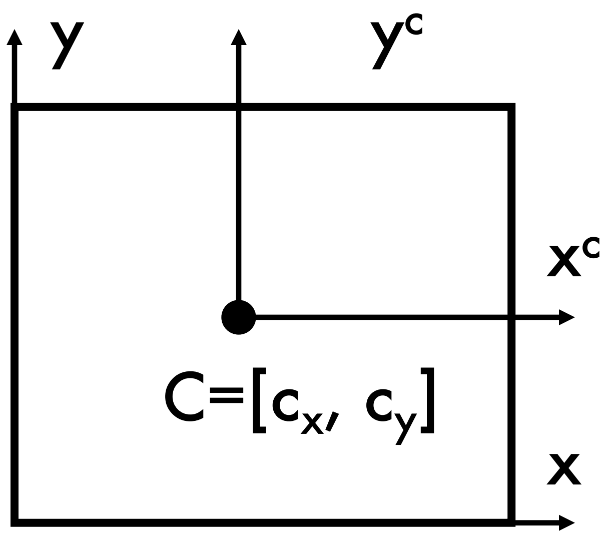




Converting to pixels

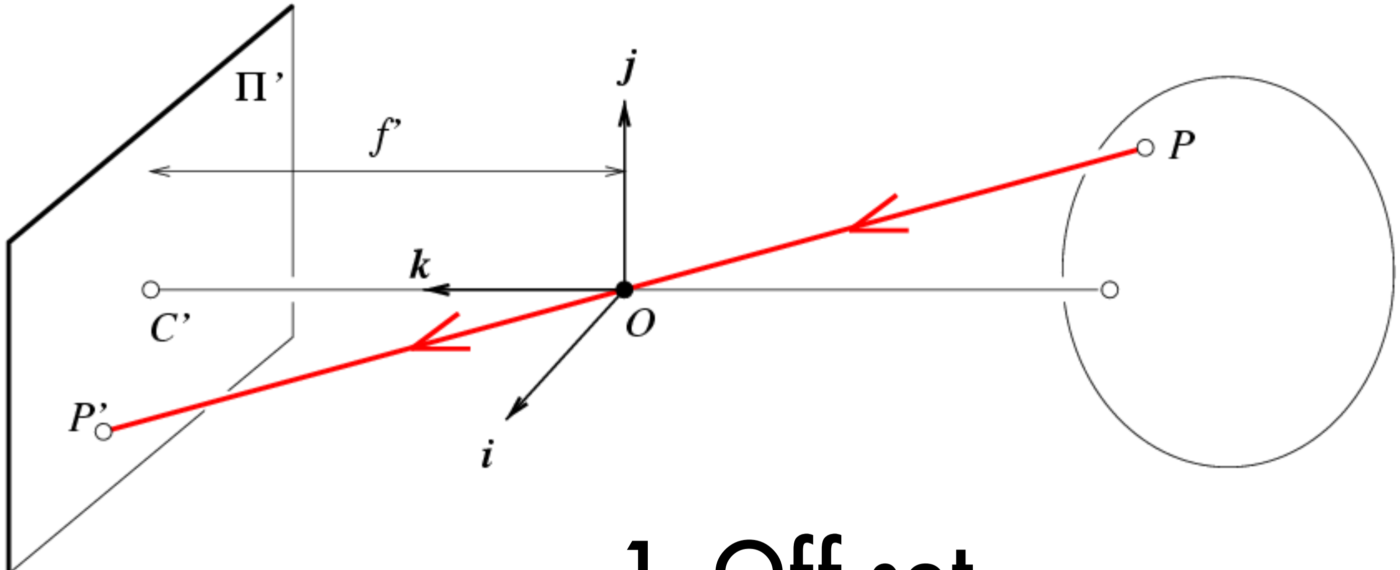


1. Off set

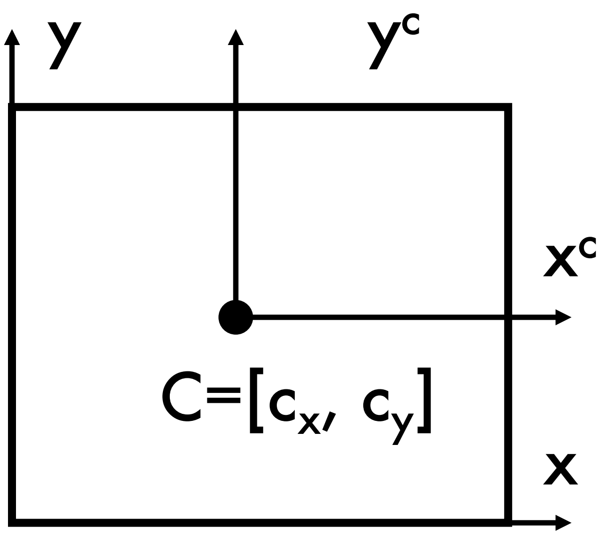


$$(x, y, z) \rightarrow (f\frac{x}{z} + c_x, f\frac{y}{z} + c_y)$$

Converting to pixels



1. Off set
2. From metric to pixels



$$(x,y,z) \rightarrow (\underbrace{\boxed{f\,k}}_{\alpha}\frac{x}{z}+c_x,\ \underbrace{\boxed{f\,l}}_{\beta}\frac{y}{z}+c_y)$$

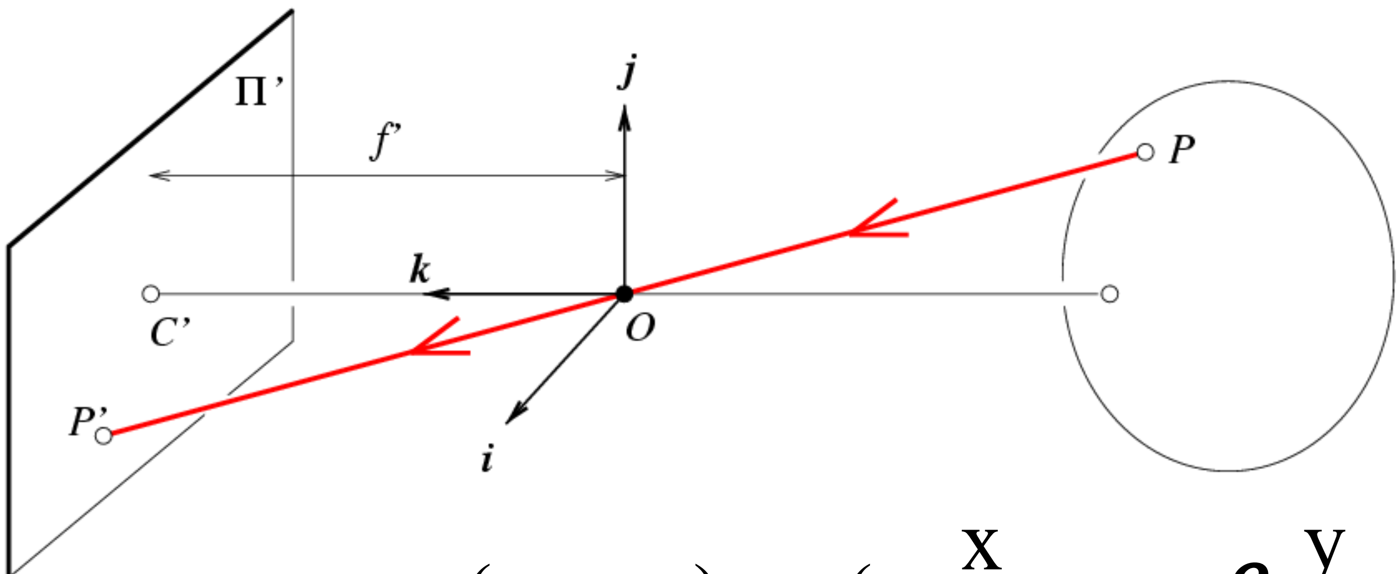
Units: k,l : pixel/m

f : m

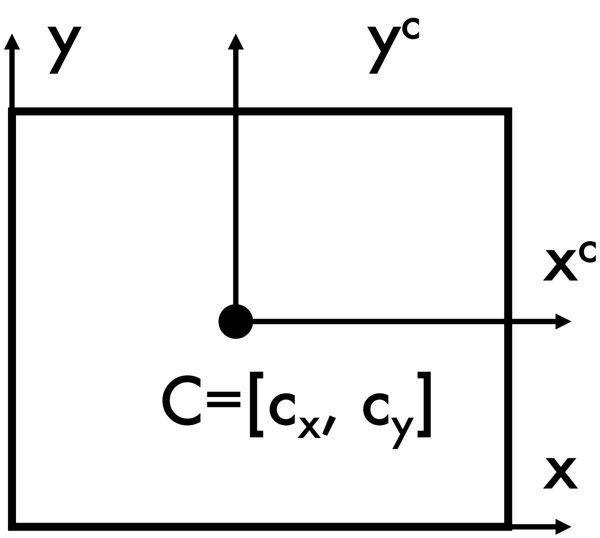
Non-square pixels

α, β : pixel

Converting to pixels



$$(x, y, z) \rightarrow (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)$$



- Matrix form?

A related question:
- Is this a linear transformation?

$$(x, y, z) \rightarrow (f\,\frac{x}{z}, f\,\frac{y}{z})$$

Is this a linear transformation?

No – division by z is nonlinear

How to make it linear?

Homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

- Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

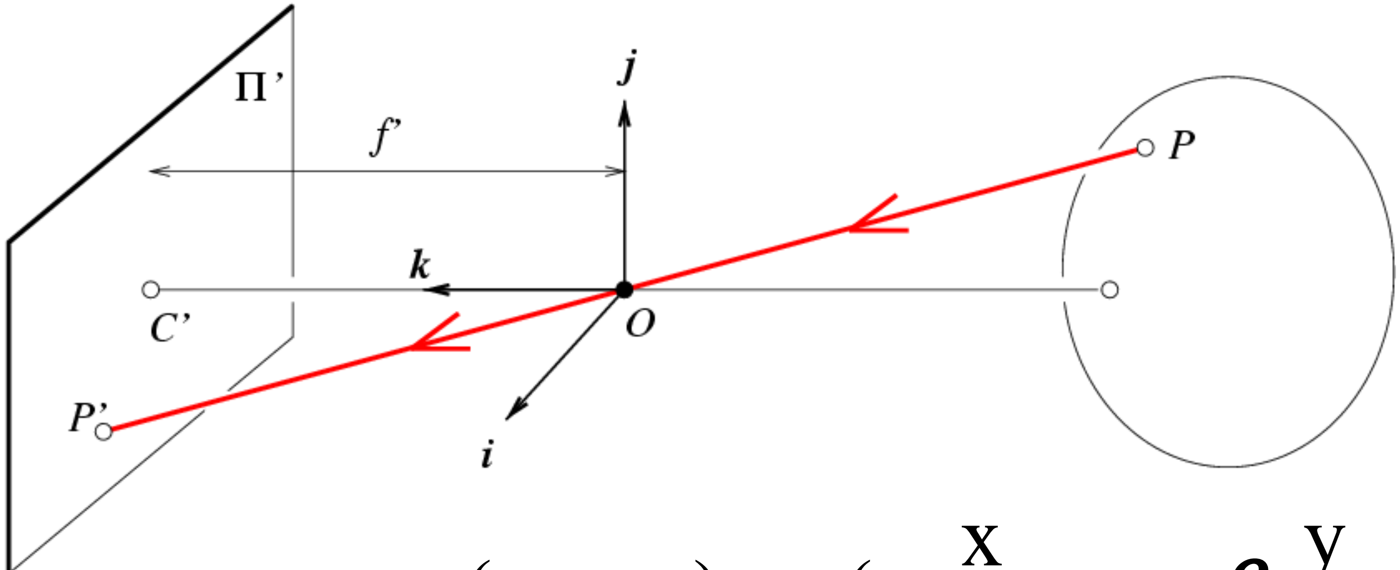
Perspective Projection Transformation

$$X' = \begin{bmatrix} f\,x \\ f\,y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{M} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

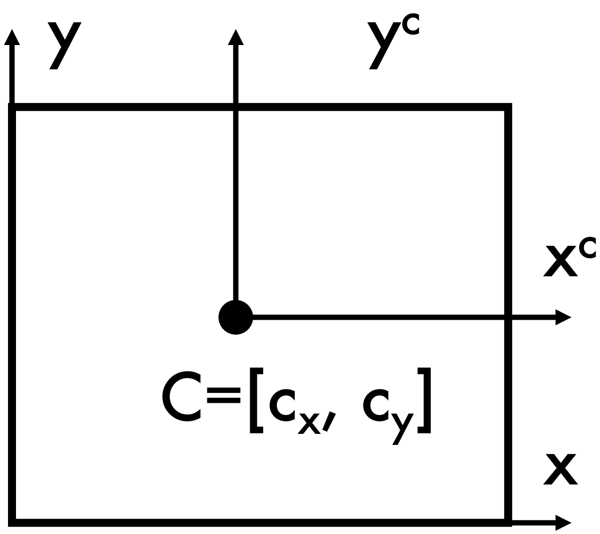
$$X' = M\,X$$

$$\Re^4 \xrightarrow{H} \Re^3$$

Camera Matrix

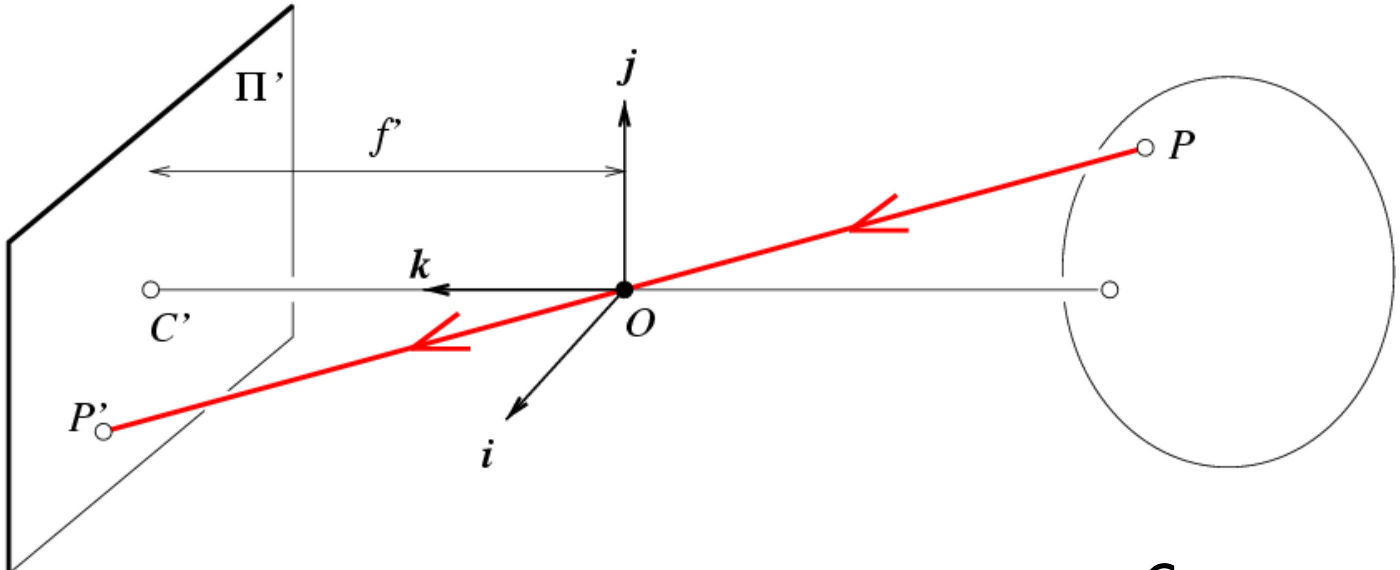


$$(x, y, z) \rightarrow \left(\alpha \frac{x}{z} + c_x, \boldsymbol{\beta} \frac{y}{z} + c_y\right)$$



$$X' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Camera Matrix



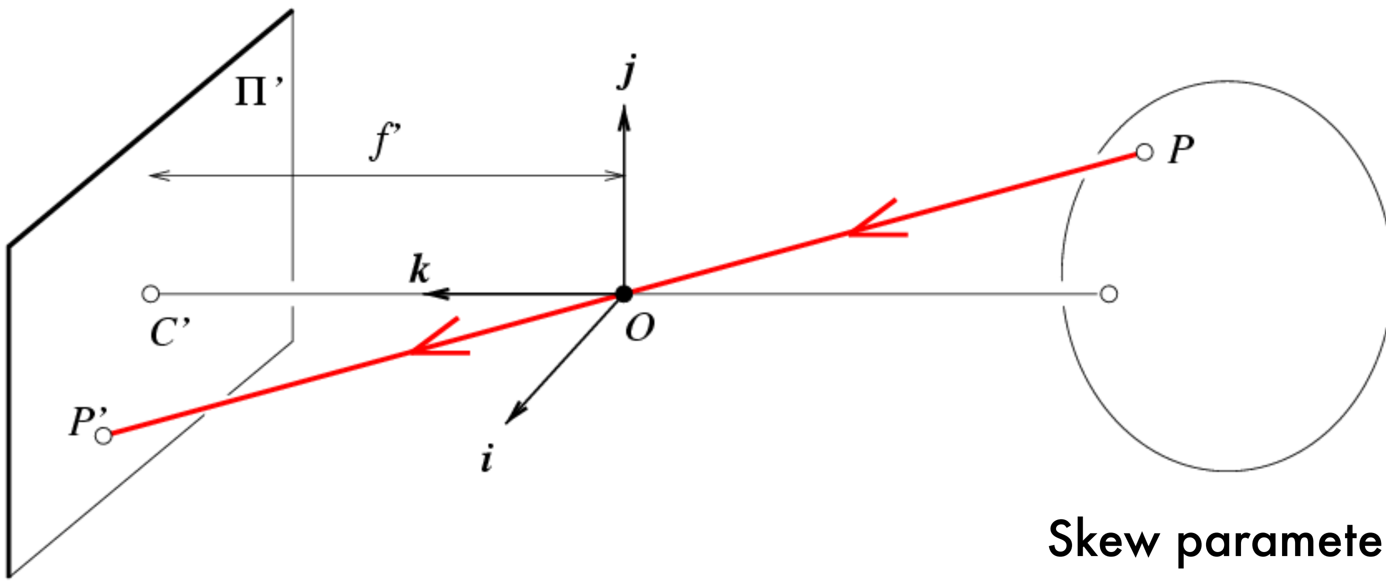
Camera matrix K

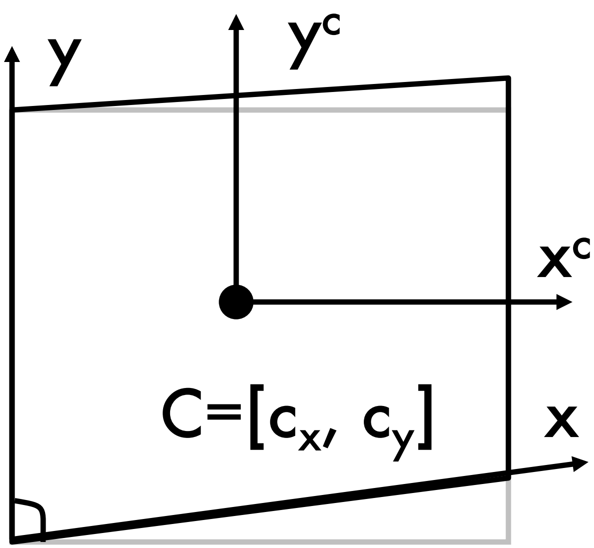
$$X' = M\,X$$

$$= K\begin{bmatrix} I & 0 \end{bmatrix} X$$

$$X' = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Finite projective cameras



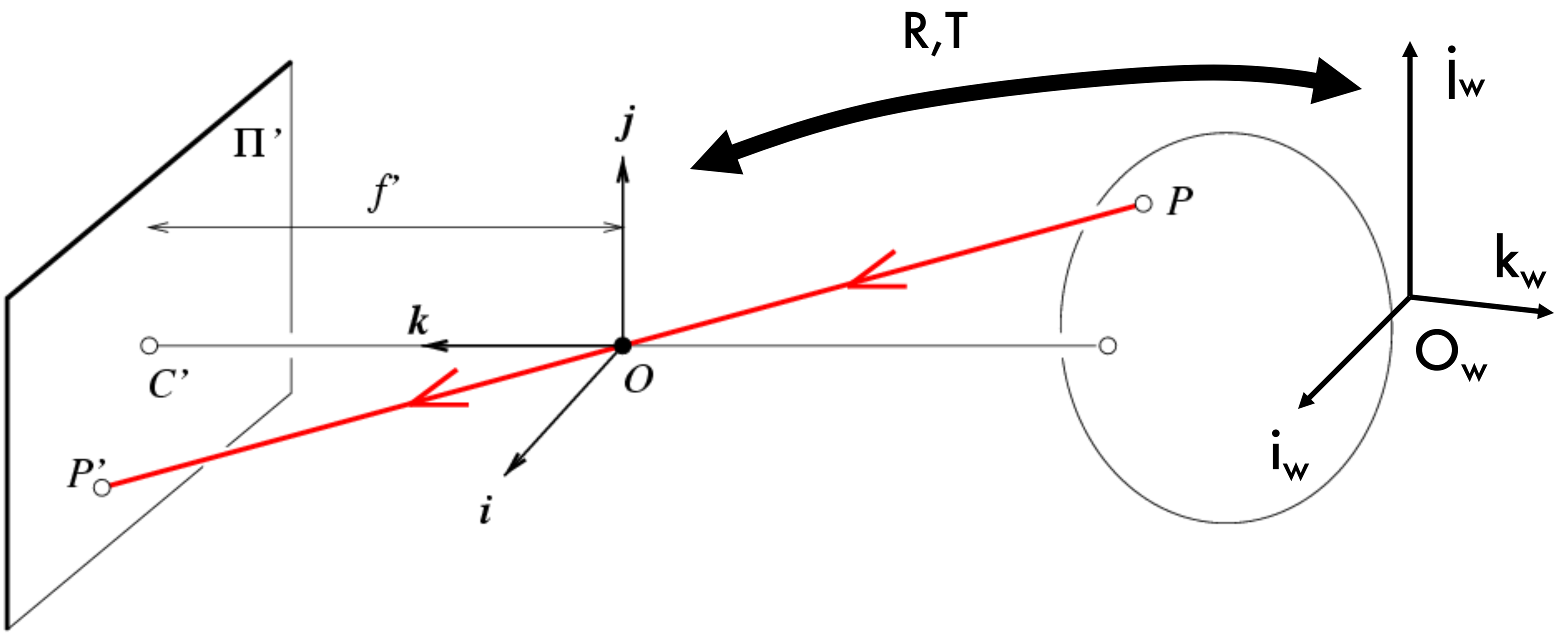


Skew parameter

$$X' = \begin{bmatrix} \alpha & \boxed{s} & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

K has 5 degrees of freedom!

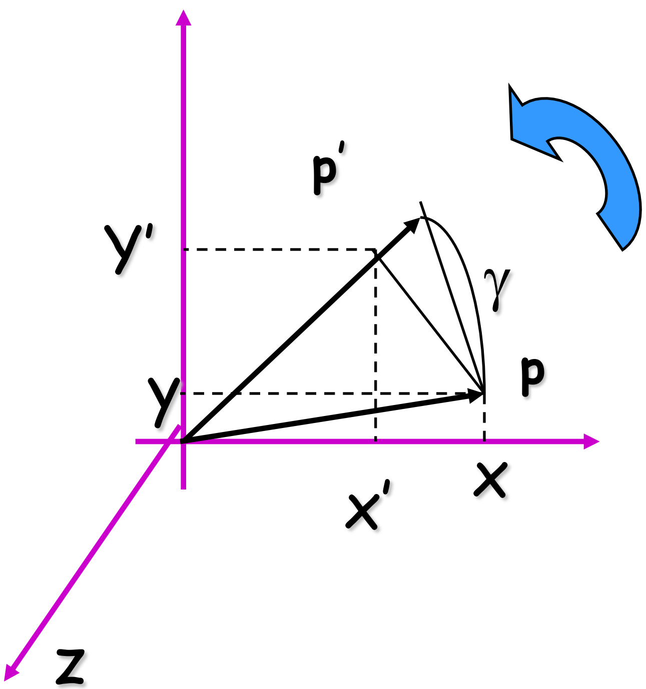
World reference system



- The mapping so far is defined within the camera reference system
- What if an object is represented in the world reference system

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:

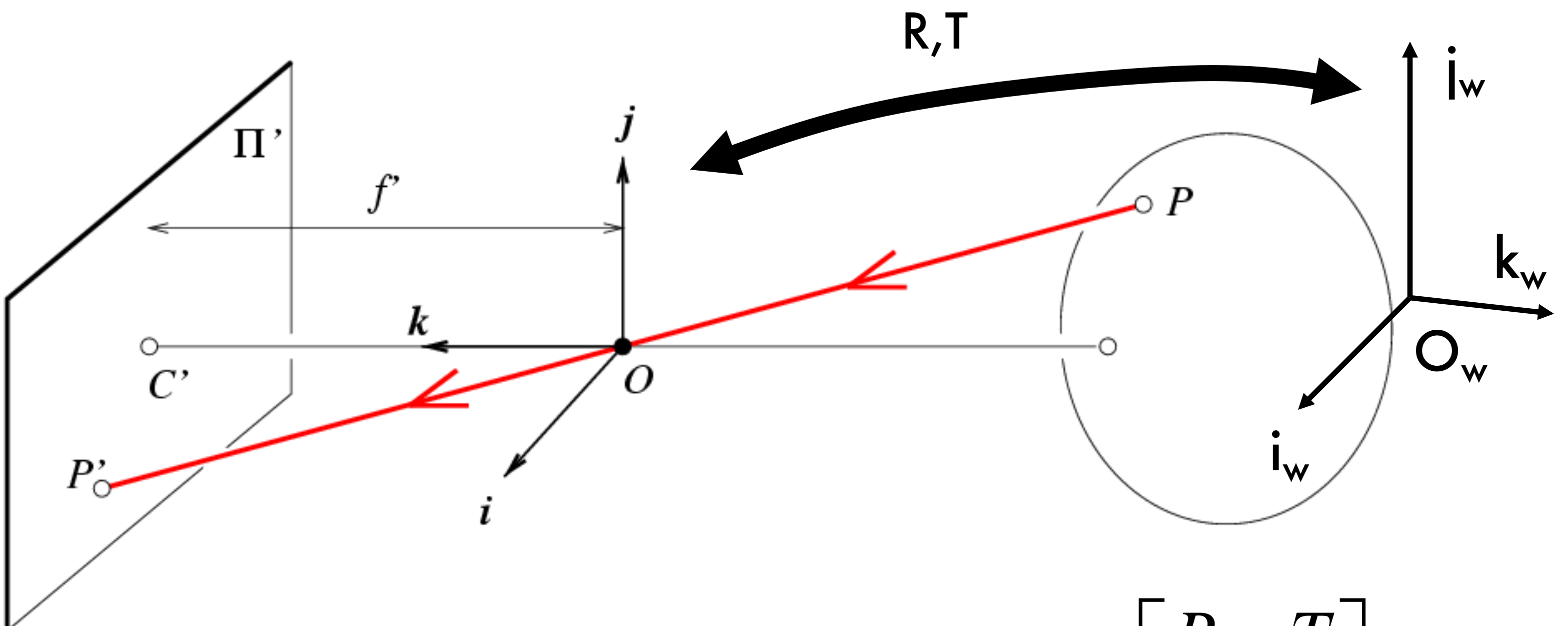


$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

World reference system



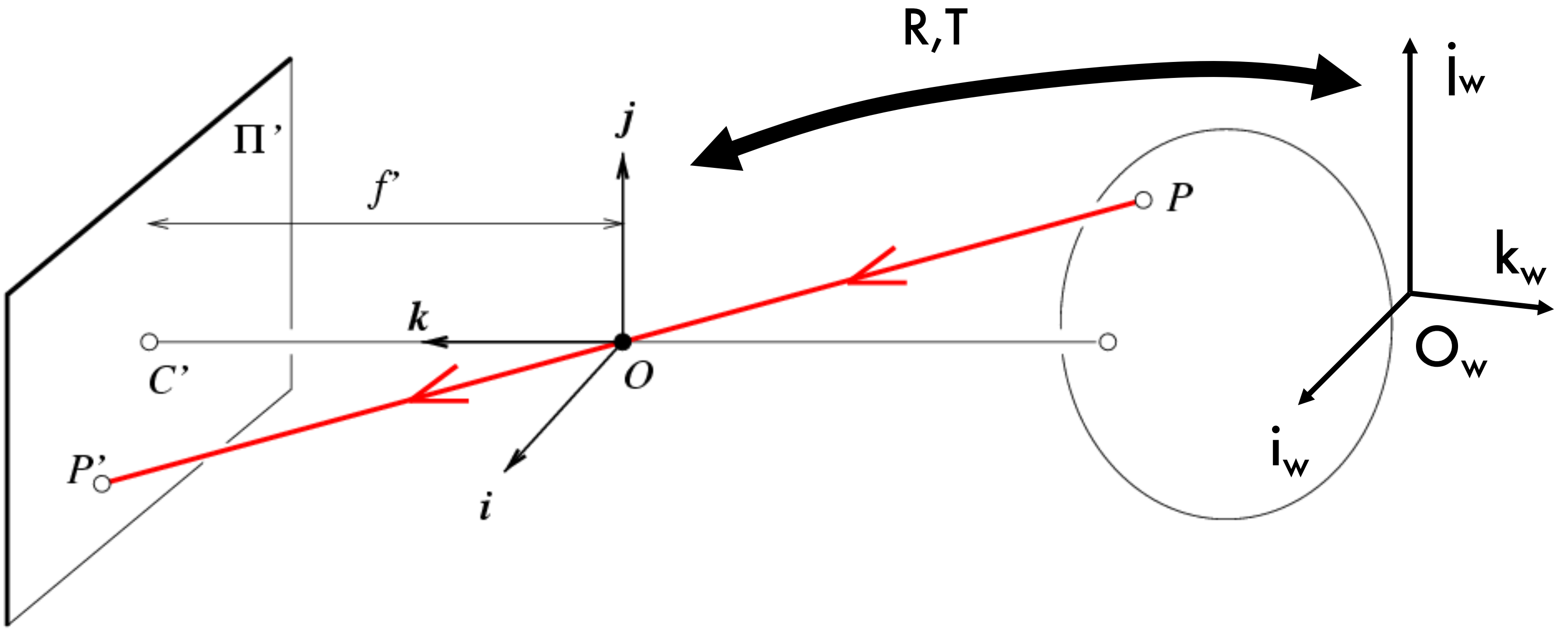
In 4D homogeneous coordinates: $X = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} X_w$

Internal parameters

External parameters

$$X' = K\begin{bmatrix} I & 0 \end{bmatrix} X = K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} X_w = \underbrace{\boxed{K\begin{bmatrix} R & T \end{bmatrix}}}_{M} X_w$$

Projective cameras

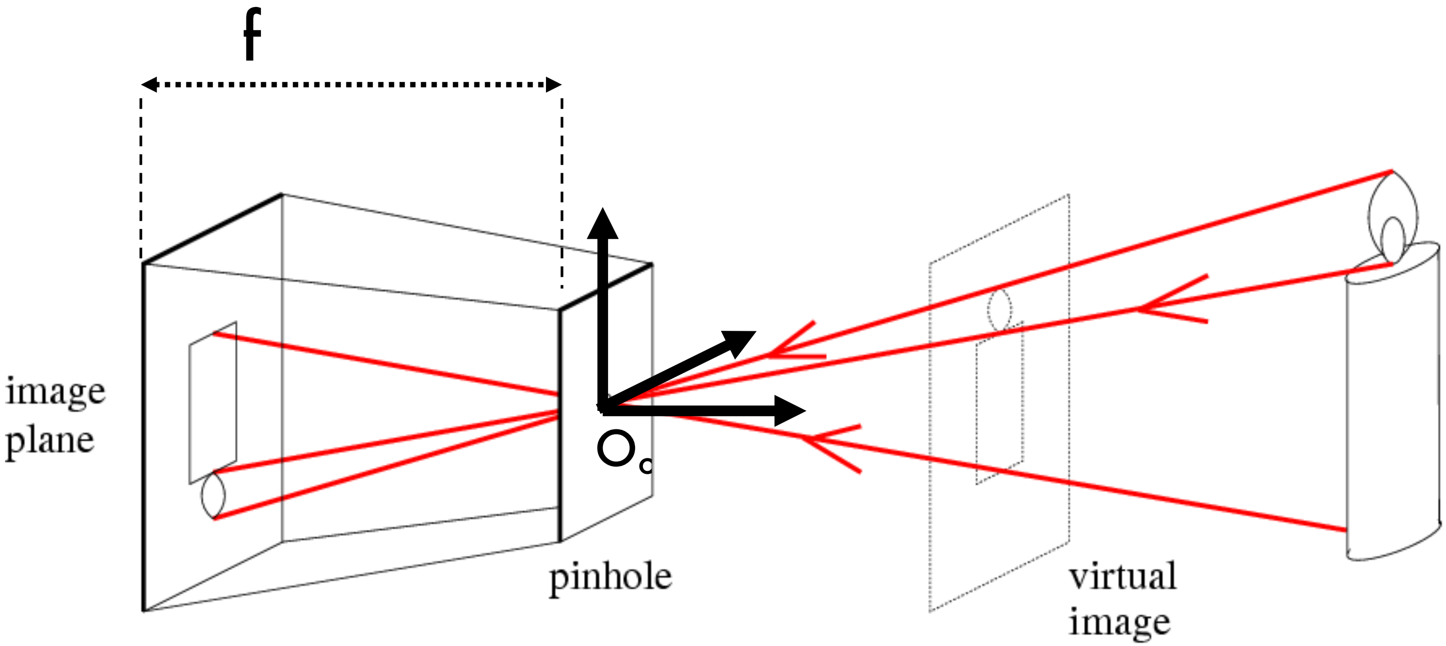


$$X'_{3\times 1} = M_{3x4}\; X_w = \mathrm{K}_{3\times 3}\begin{bmatrix}\mathrm{R} & \mathrm{T}\end{bmatrix}_{3\times 4}\; \mathrm{X}_{w\,4\times 1} \qquad \mathrm{K} = \begin{bmatrix} \alpha & s & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

How many degrees of freedom?

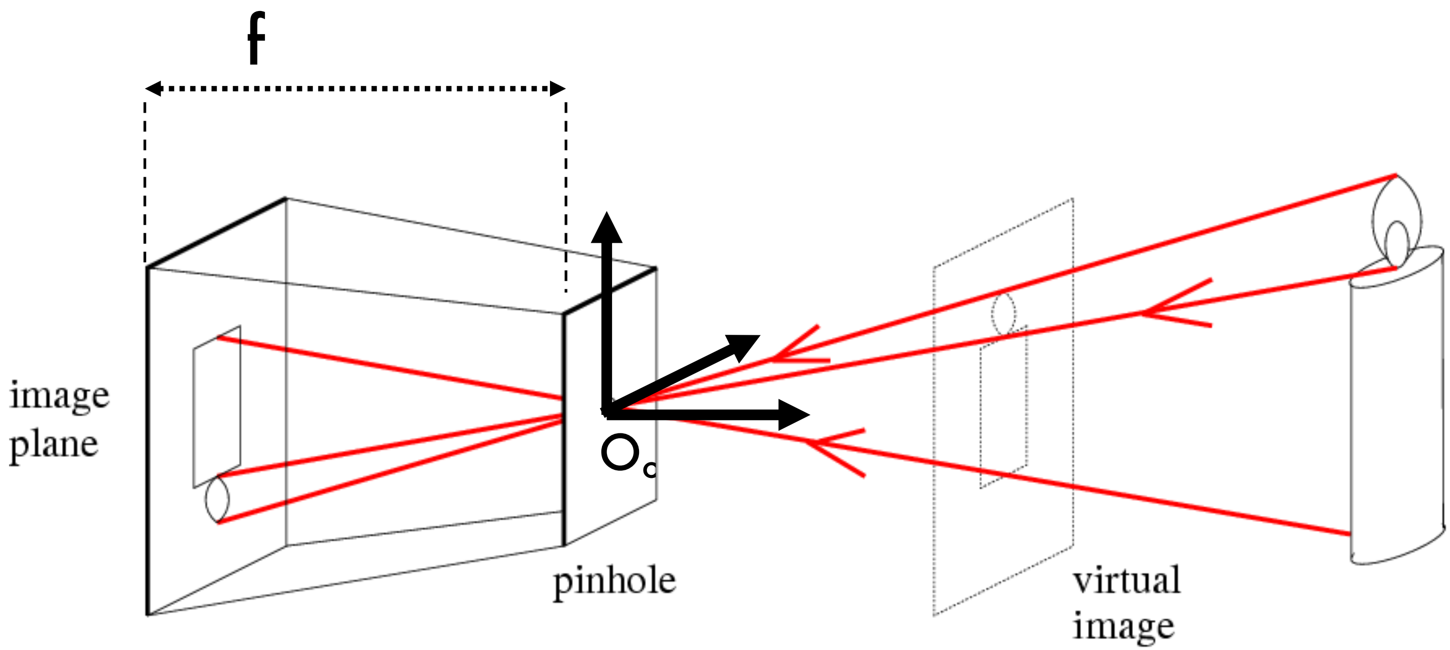
5 + 3 + 3 =11!

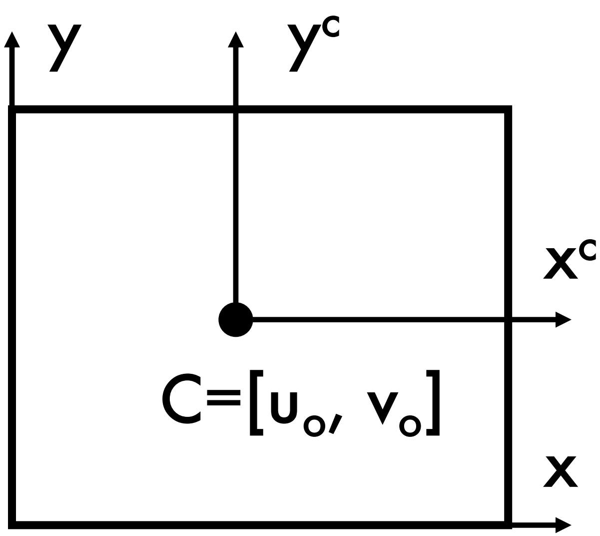
Projective camera



f = focal length

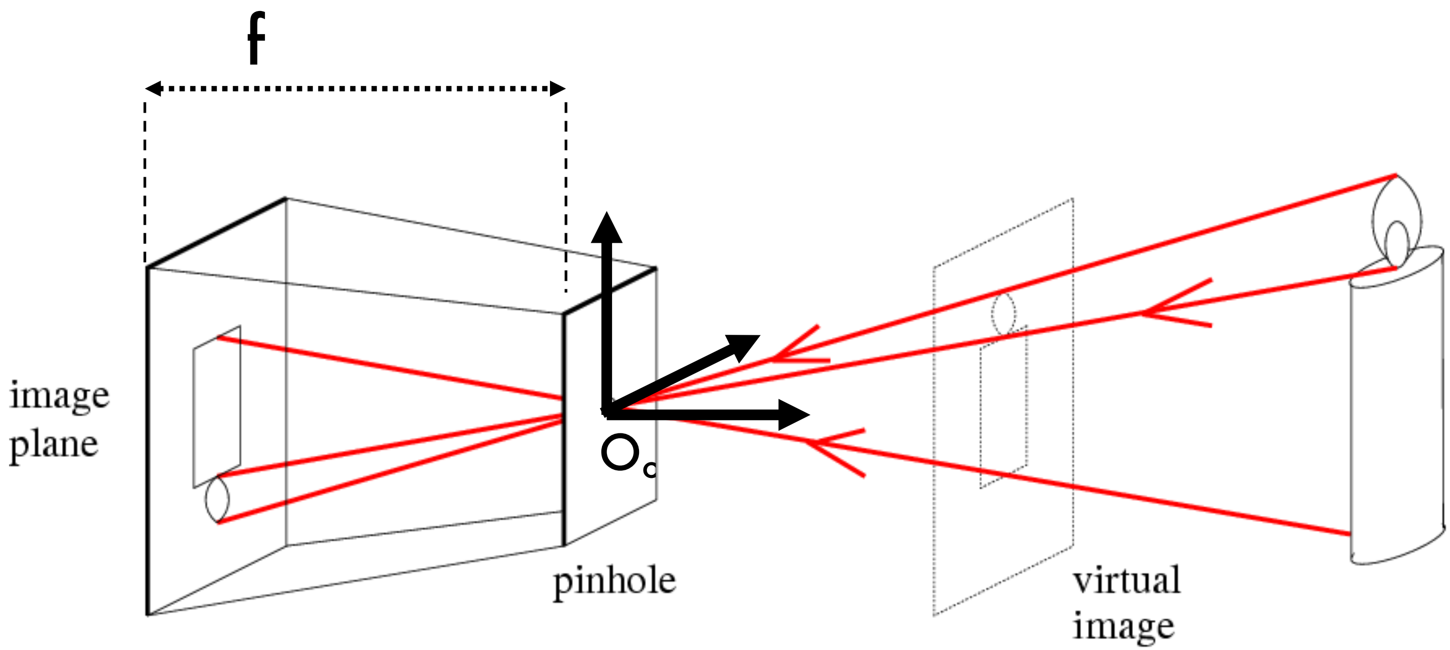
Projective camera





f = focal length
u_o, v_o = offset

Projective camera



Units: k,l [pixel/m]

f [m]

Non-square pixels

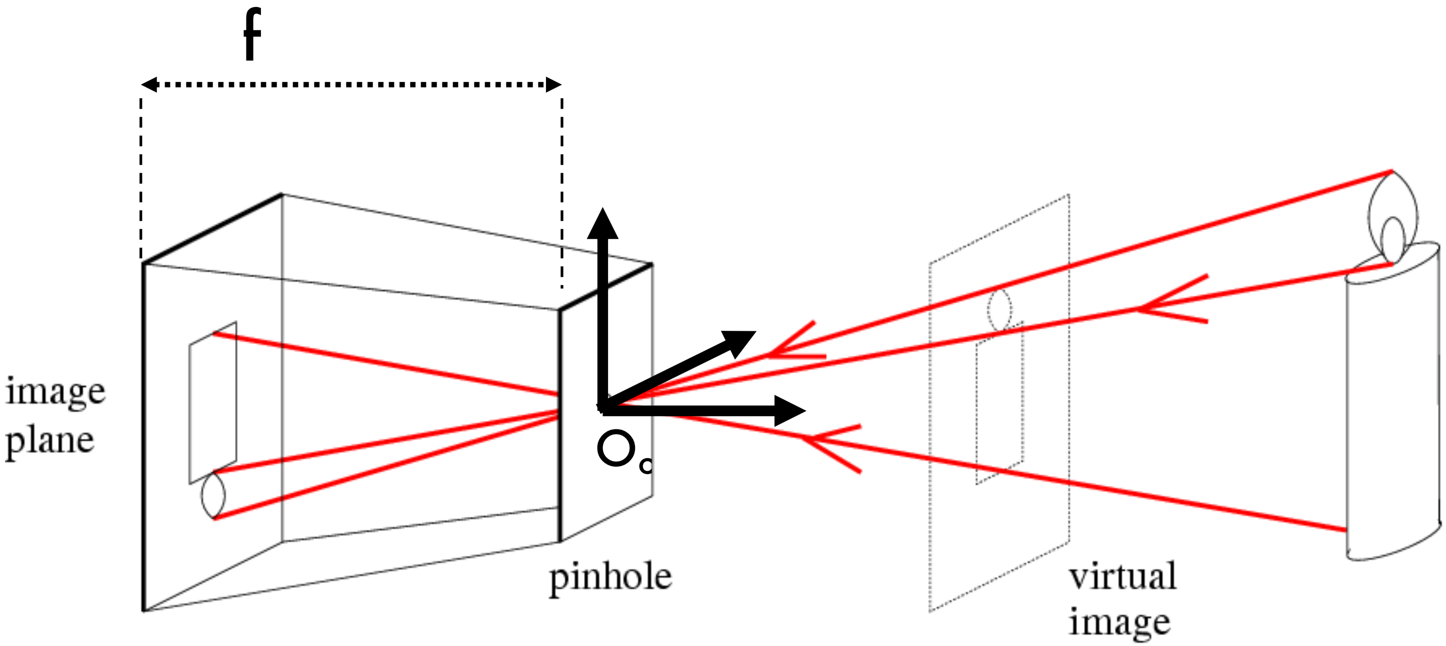
α, β [pixel]

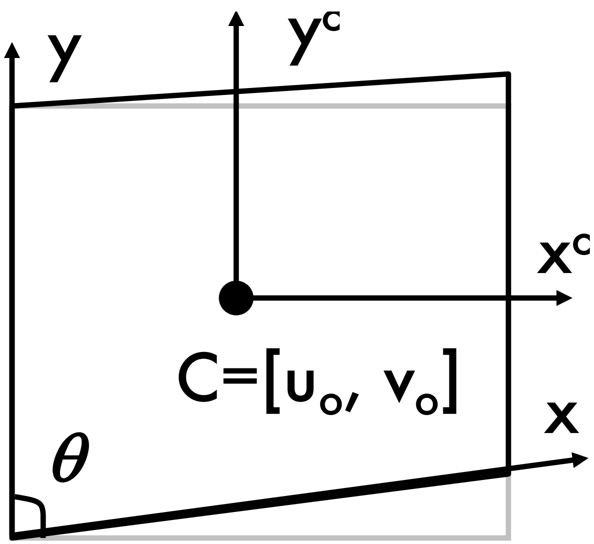
f = focal length

u_o, v_o = offset

α, β $\rightarrow$ non-square pixels

Projective camera





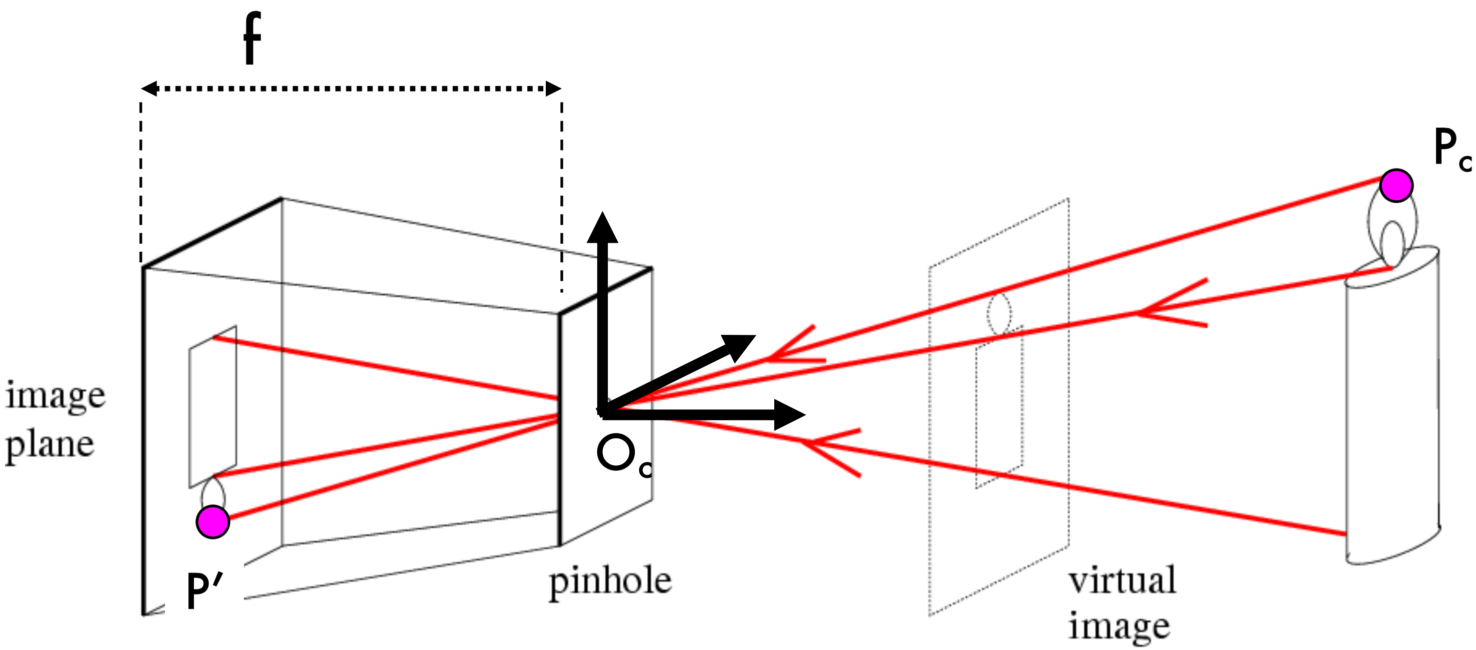
f = focal length

u_o, v_o = offset

$\alpha, \beta \rightarrow$ non-square pixels

θ = skew angle

Projective camera



$$P' = \begin{bmatrix} \alpha & s & u_o & 0 \\ 0 & \beta & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

f = focal length

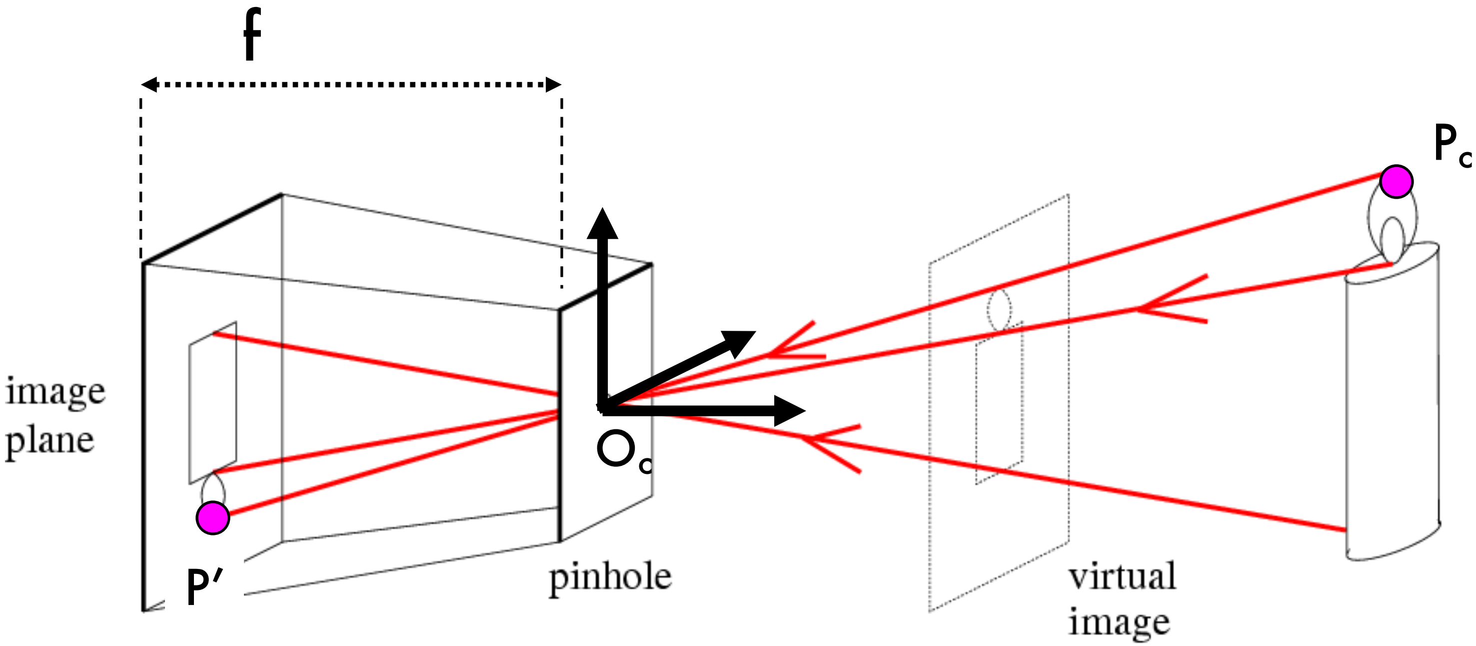
u_o, v_o = offset

$\alpha, \beta \rightarrow$ non-square pixels

θ = skew angle

K has 5 degrees of freedom!

Projective camera



$$P' = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o & 0 \\ 0 & \dfrac{\beta}{\sin\theta} & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

f = focal length

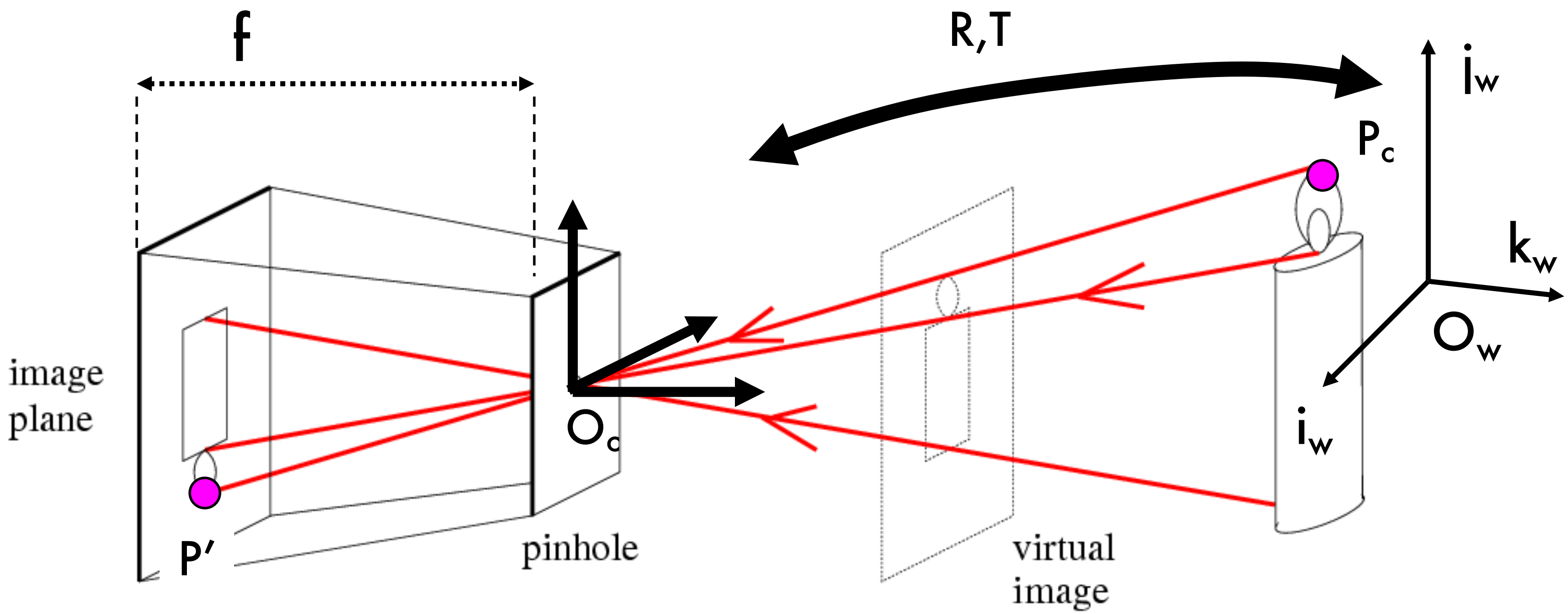
u_o, v_o = offset

$\alpha, \beta \rightarrow$ non-square pixels

θ = skew angle

K has 5 degrees of freedom!

Projective camera



$$P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} P_w$$

$$T = -RO_c$$

f = focal length

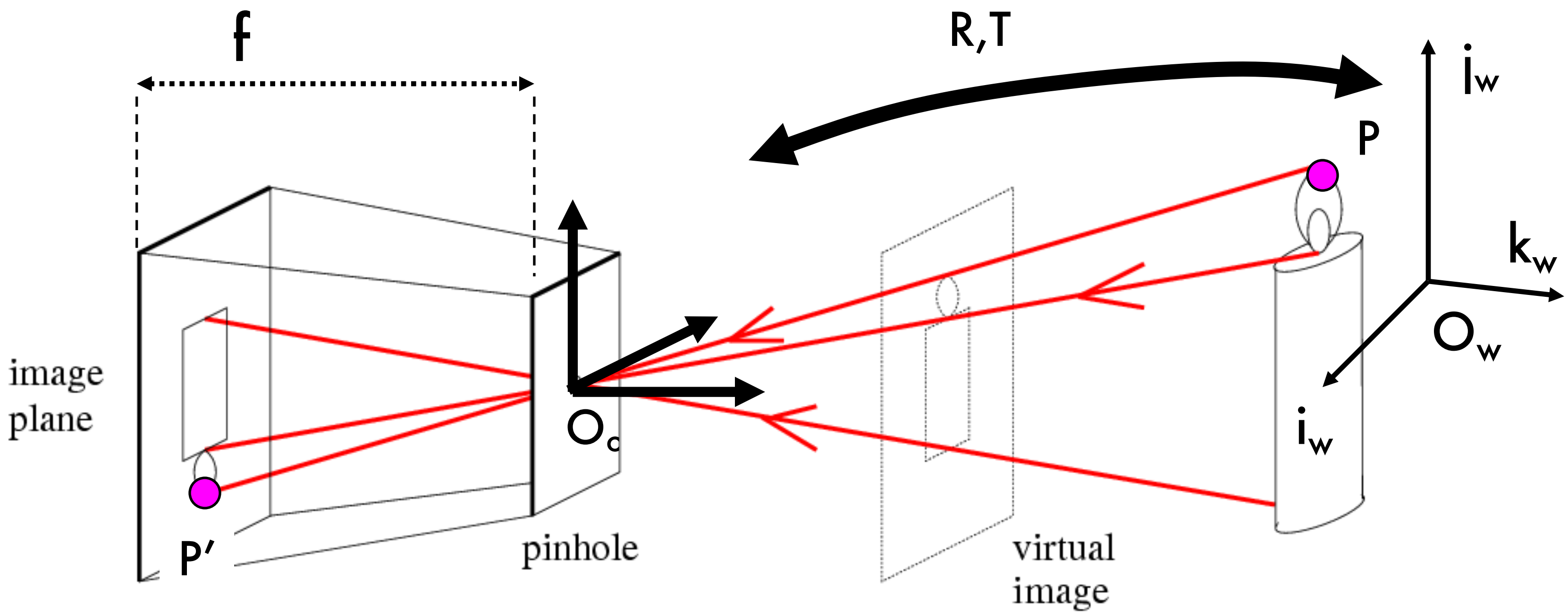
u_o, v_o = offset

$\alpha, \beta \rightarrow$ non-square pixels

θ = skew angle

R,T = rotation, translation

Projective camera



$$P' = M\,P_w$$

$$= \boxed{K}\boxed{\begin{bmatrix} R & T \end{bmatrix}} P_w$$

Internal parameters

External parameters

f = focal length

u_o, v_o = offset

$\alpha, \beta \rightarrow$ non-square pixels

θ = skew angle

R,T = rotation, translation

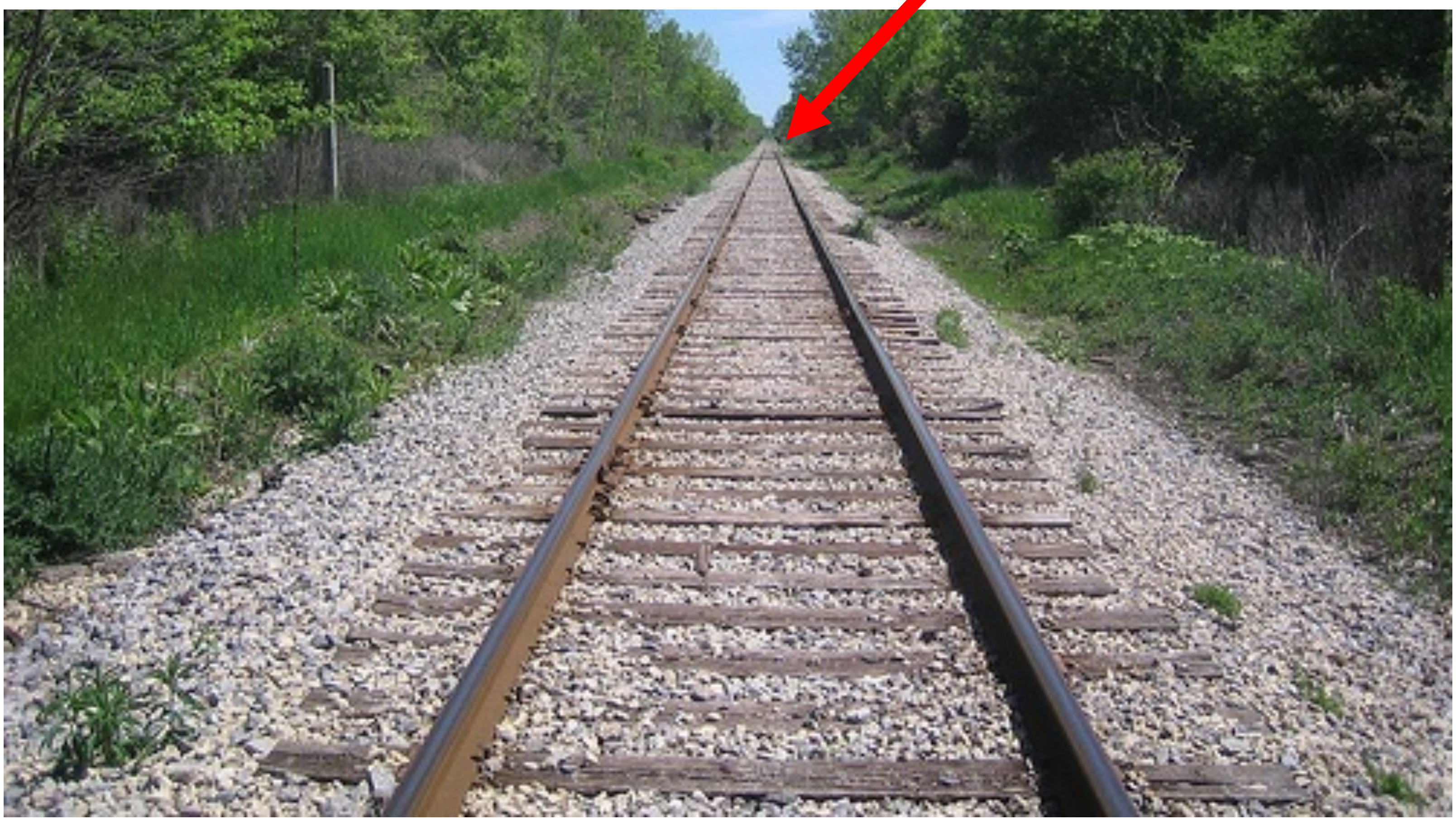
Properties of Projection

- Points project to points
- Lines project to lines
- Distant objects look smaller

Properties of Projection

- Angles are not preserved
- Parallel lines meet!

Vanishing point



Projective camera

$$P' = M\,P_w \quad = \boxed{K}\left[\boxed{R \quad T}\right]P_w$$

Internal parameters

External parameters

Projective camera

$$P' = M\,P_w = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

$$\boxed{\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}}$$

$$K = \begin{bmatrix} \boldsymbol{\alpha} & -\boldsymbol{\alpha}\cot\boldsymbol{\theta} & u_o \\ 0 & \dfrac{\boldsymbol{\beta}}{\sin\boldsymbol{\theta}} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Goal of calibration

Estimate intrinsic and extrinsic parameters from 1 or multiple images

$$P' = M\ P_w \quad = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

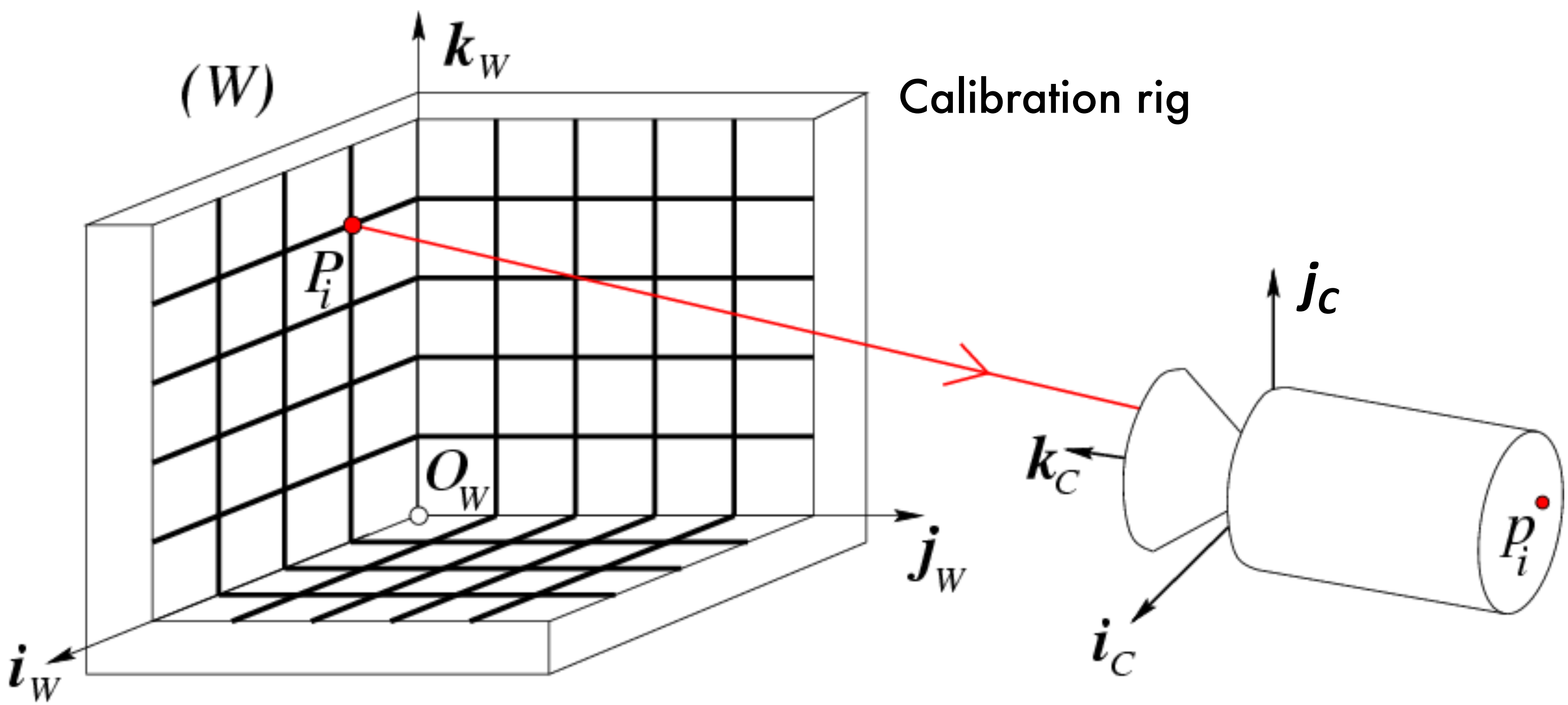
$$\boxed{\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}}$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \frac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Change notation:
$P = P_w$

$p = P'$

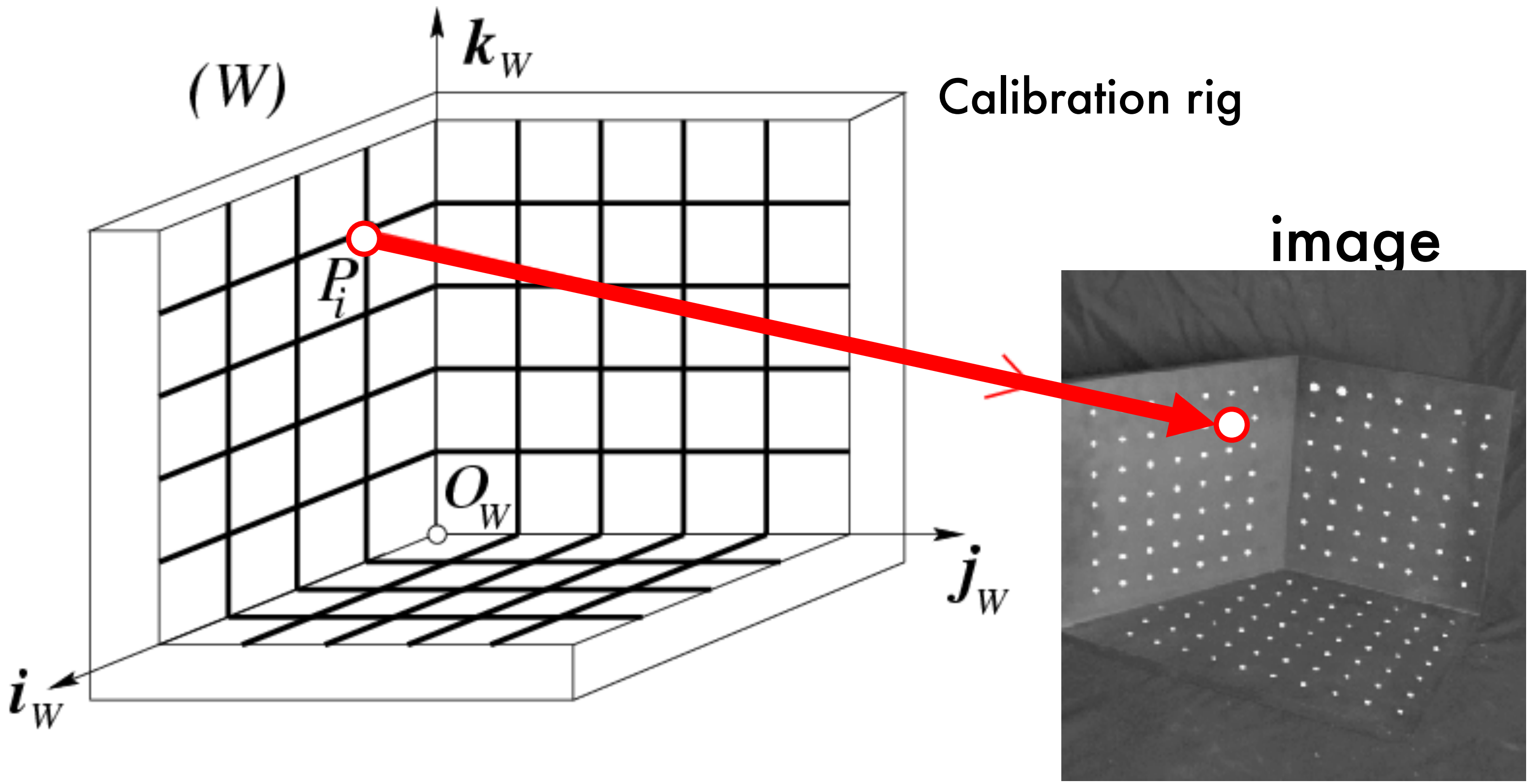
Calibration Problem



- $P_1 \ldots P_n$ with **known** positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ **known** positions in the image

Goal: compute intrinsic and extrinsic parameters

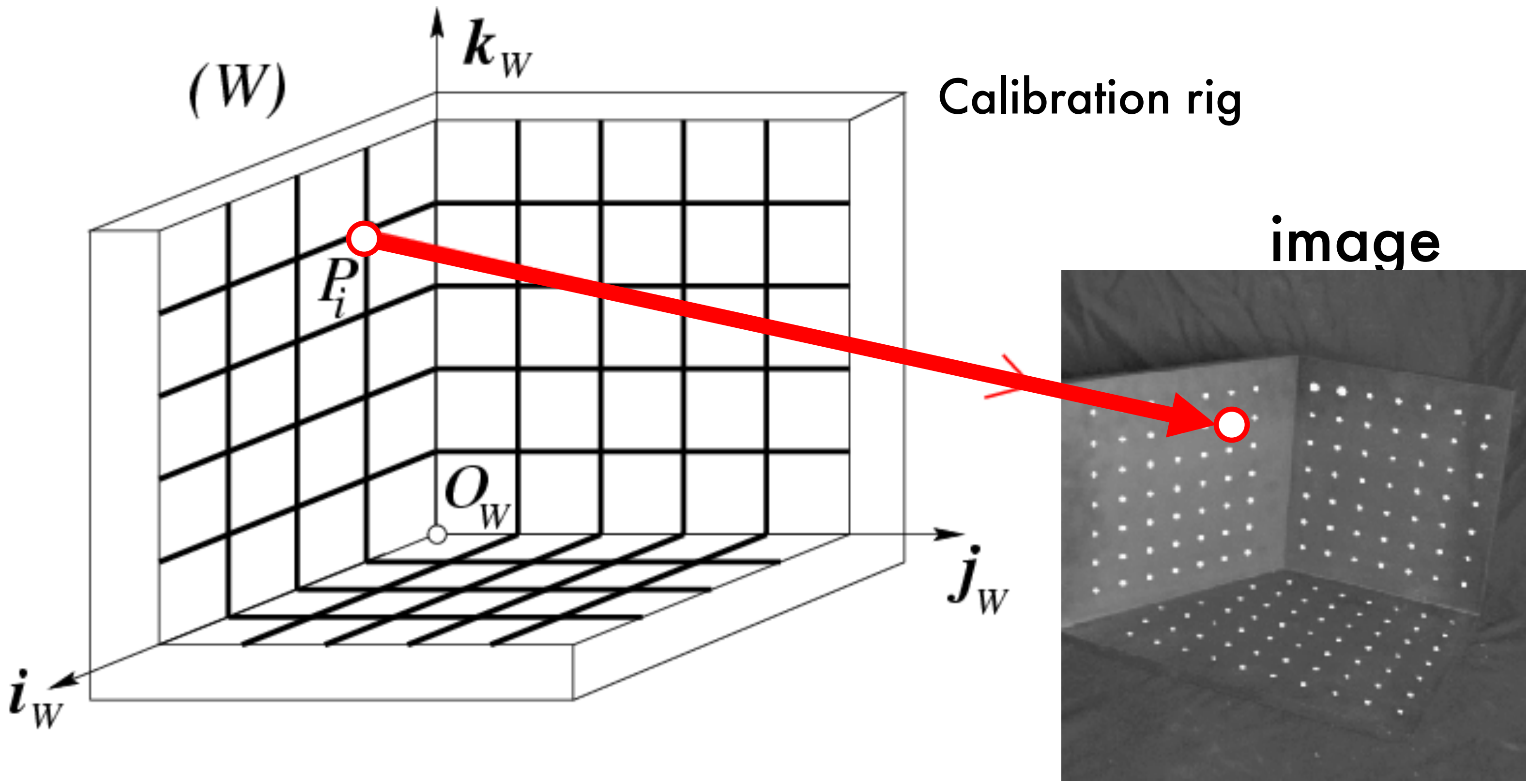
Calibration Problem



- $P_1 \ldots P_n$ with known positions in $[O_W, i_W, j_W, k_W]$
- $p_1, \ldots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

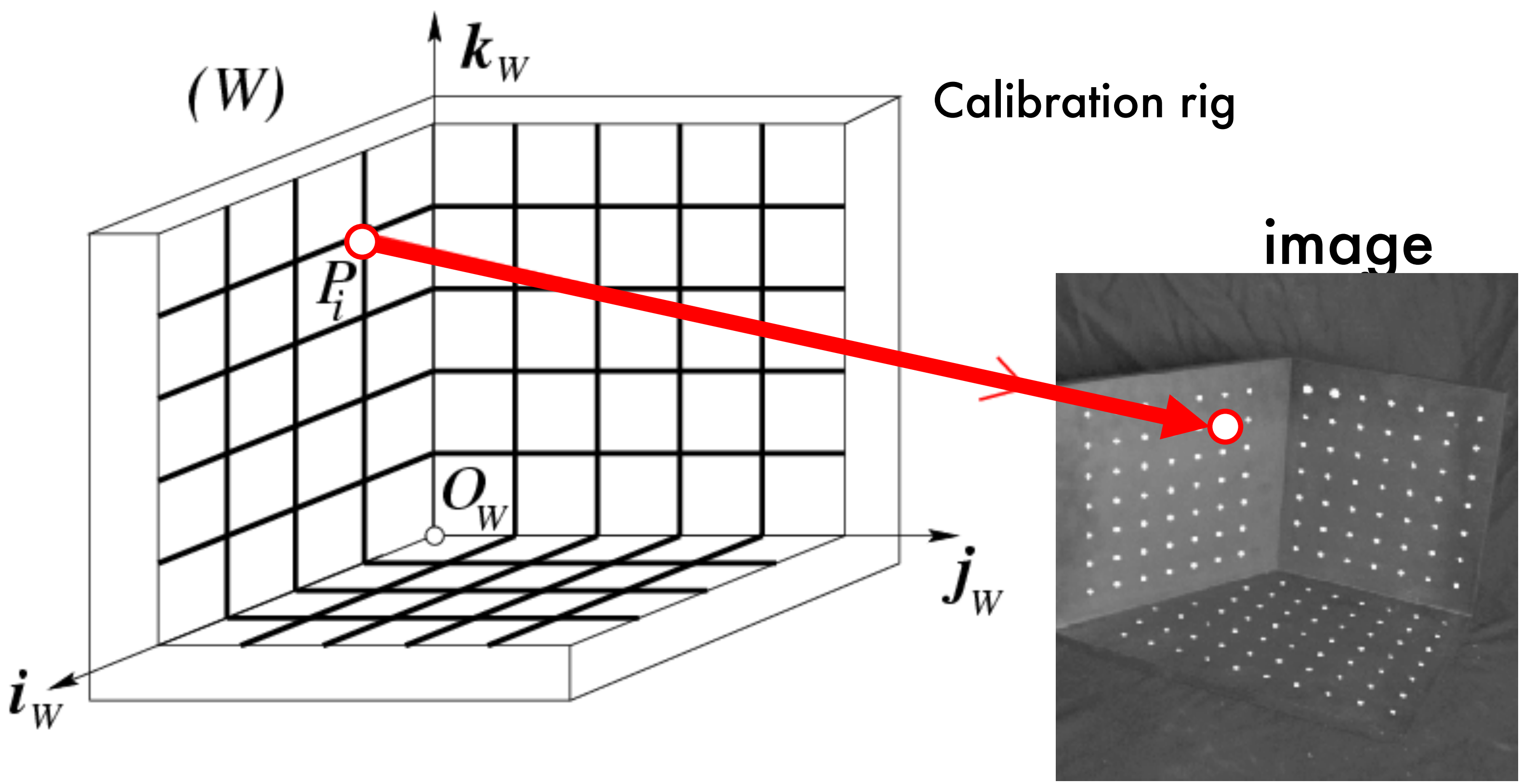
Calibration Problem



How many correspondences do we need?

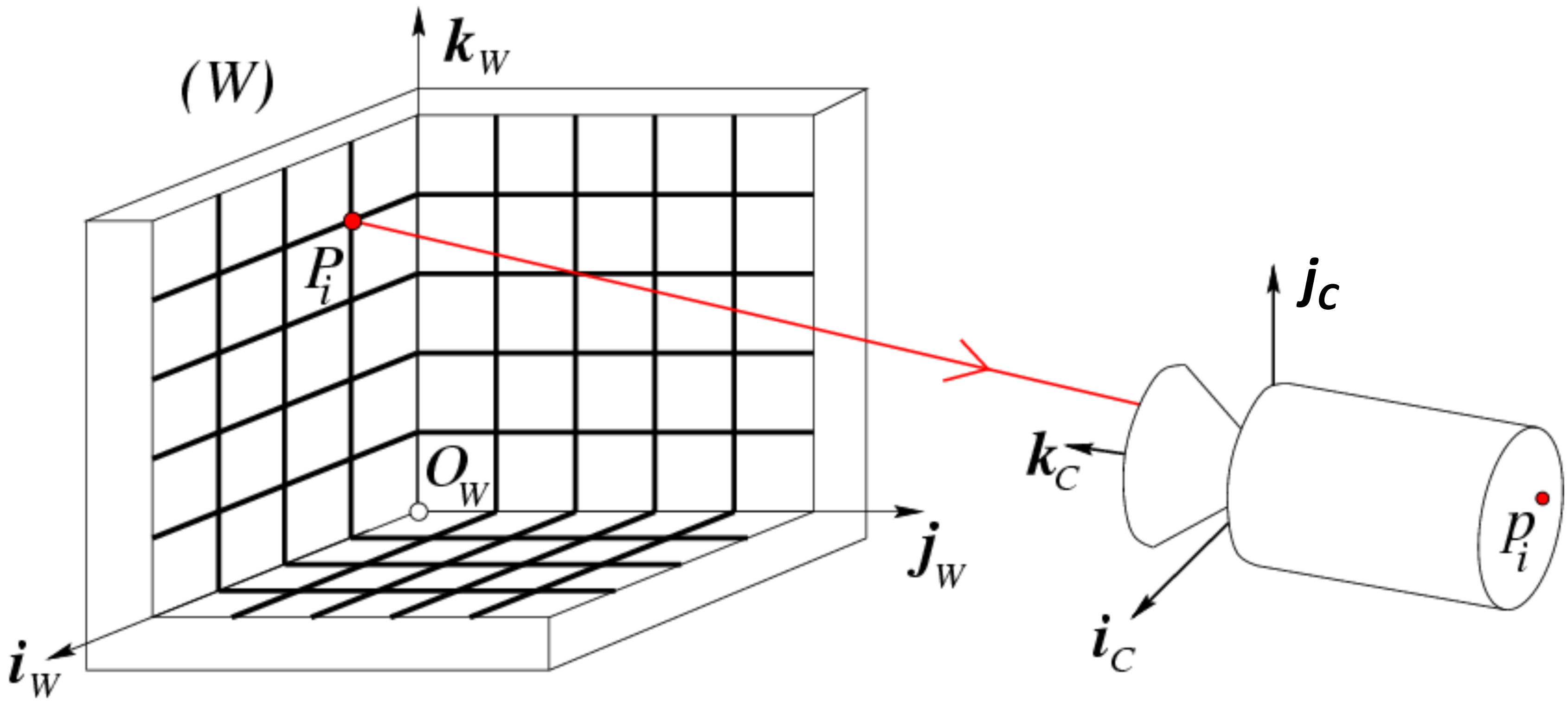
- M has 11 unknown
- We need 11 equations
- 6 correspondences would do it

Calibration Problem



In practice, using more than 6 correspondences enables more robust results

Calibration Problem



$$P_i \rightarrow M\,P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1\,P_i}{\mathbf{m}_3\,P_i} \\ \dfrac{\mathbf{m}_2\,P_i}{\mathbf{m}_3\,P_i} \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 \ P_i}{\mathbf{m}_3 \ P_i} \\ \dfrac{\mathbf{m}_2 \ P_i}{\mathbf{m}_3 \ P_i} \end{bmatrix}$$

$$u_i = \frac{\mathbf{m}_1 \ P_i}{\mathbf{m}_3 \ P_i} \ \rightarrow \ u_i(\mathbf{m}_3 \ P_i) = \mathbf{m}_1 \ P_i \ \rightarrow \ u_i(\mathbf{m}_3 \ P_i) - \mathbf{m}_1 \ P_i = 0$$

$$v_i = \frac{\mathbf{m}_2 \ P_i}{\mathbf{m}_3 \ P_i} \ \rightarrow \ v_i(\mathbf{m}_3 \ P_i) = \mathbf{m}_2 \ P_i \ \rightarrow \ v_i(\mathbf{m}_3 \ P_i) - \mathbf{m}_2 \ P_i = 0$$

Calibration Problem

$$
\begin{cases}
u_1(\mathbf{m}_3\ P_1) - \mathbf{m}_1\ P_1 = 0 \\
v_1(\mathbf{m}_3\ P_1) - \mathbf{m}_2\ P_1 = 0 \\
\vdots \\
u_i(\mathbf{m}_3\ P_i) - \mathbf{m}_1\ P_i = 0 \\
v_i(\mathbf{m}_3\ P_i) - \mathbf{m}_2\ P_i = 0 \\
\vdots \\
u_n(\mathbf{m}_3\ P_n) - \mathbf{m}_1\ P_n = 0 \\
v_n(\mathbf{m}_3\ P_n) - \mathbf{m}_2\ P_n = 0
\end{cases}
$$

Block Matrix Multiplication

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} \\ \mathrm{B}_{21} & \mathrm{B}_{22} \end{bmatrix}$$

What is *AB* ?

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

Calibration Problem

$$\begin{cases} -u_1(\mathbf{m}_3\ P_1) + \mathbf{m}_1\ P_1 = 0 \\ -v_1(\mathbf{m}_3\ P_1) + \mathbf{m}_2\ P_1 = 0 \\ \qquad \vdots \\ -u_n(\mathbf{m}_3\ P_n) + \mathbf{m}_1\ P_n = 0 \\ -v_n(\mathbf{m}_3\ P_n) + \mathbf{m}_2\ P_n = 0 \end{cases} \longrightarrow \boxed{\mathcal{P} m = 0}$$

known

unknown

Homogenous linear system

1x4

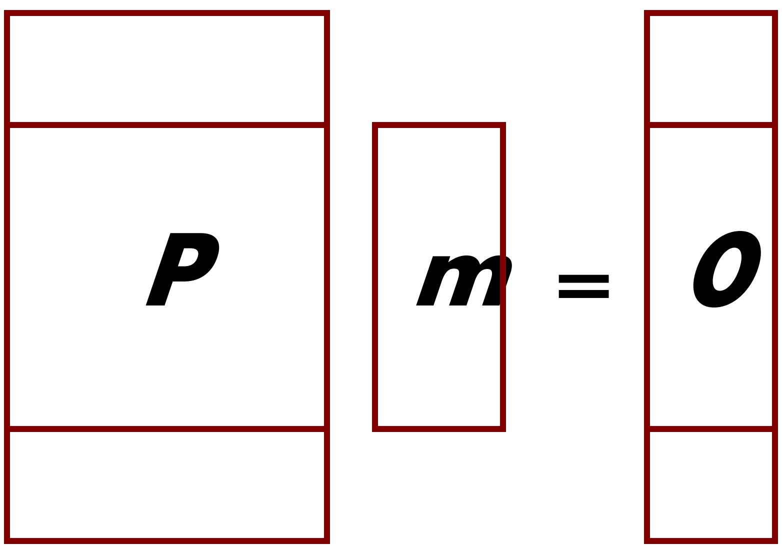
$$\mathcal{P} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{P}_1^T & \mathbf{0}^T & -u_1 \boldsymbol{P}_1^T \\ \mathbf{0}^T & \boldsymbol{P}_1^T & -v_1 \boldsymbol{P}_1^T \\ \ldots & \ldots & \ldots \\ \boldsymbol{P}_n^T & \mathbf{0}^T & -u_n \boldsymbol{P}_n^T \\ \mathbf{0}^T & \boldsymbol{P}_n^T & -v_n \boldsymbol{P}_n^T \end{pmatrix}_{2n \times 12}$$

4x1

$$\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{m}_1^{\mathrm{T}} \\ \mathbf{m}_2^{\mathrm{T}} \\ \mathbf{m}_3^{\mathrm{T}} \end{pmatrix}_{12\times 1}$$

Homogeneous M x N Linear Systems

M=number of equations = 2n
N=number of unknown = 11



Rectangular system (M>N)

- 0 is always a solution
- To find non-zero solution

Minimize $|\mathbf{P}\ \mathbf{m}|^2$

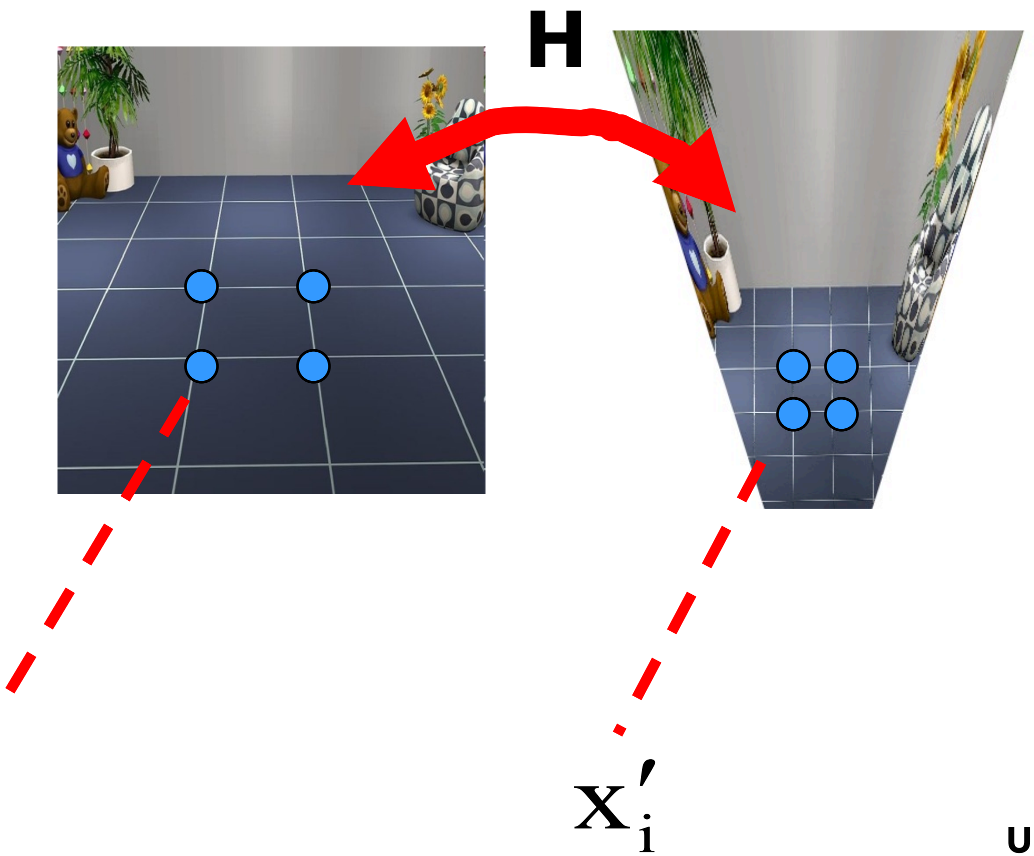
under the constraint $|\mathbf{m}|^2 = 1$

Calibration Problem

$$\mathcal{P}\boldsymbol{m} = 0$$

- How do we solve this homogenous linear system?
- Using DLT (Direct Linear Transformation) algorithm via SVD decomposition

DLT algorithm (Direct Linear Transformation)



$$\mathbf{x}'_i = \mathbf{H}\,\mathbf{x}_i \longrightarrow \underbrace{\mathbf{A}_i}_{\text{Function of measurements}} \overbrace{\mathbf{h}}^{\text{unknown}} = 0$$

Calibration Problem

$$\boxed{\mathcal{P}}\boldsymbol{m} = 0$$

SVD decomposition of P

$$\boxed{U_{2n\times 12}\ D_{12\times 12} V^{T}{}_{12\times 12}}$$

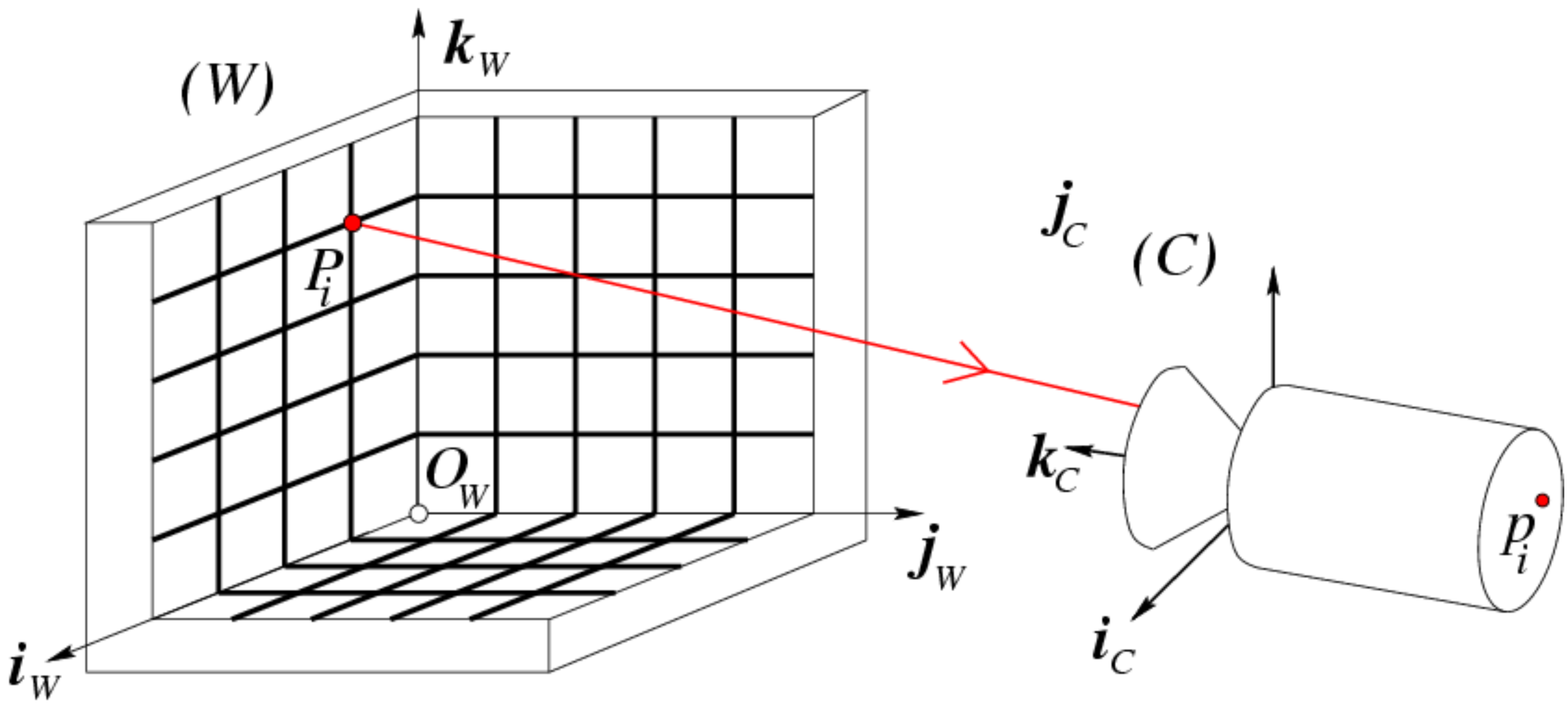
Last column of V gives $\boldsymbol{m}$

$$\downarrow$$

$$M$$

$$M\,P_i \rightarrow p_i$$

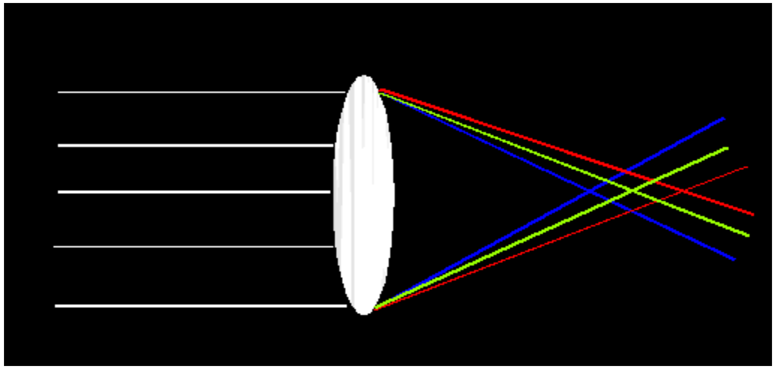
Degenerate cases

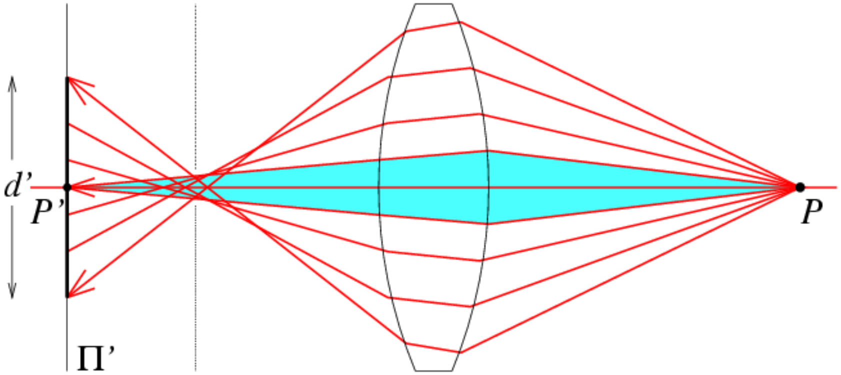


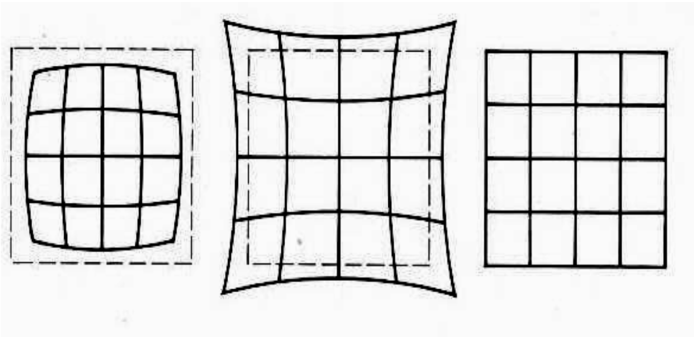
- P_i's cannot lie on the same plane!
- Points cannot lie on the intersection curve of two quadric surfaces

Taking lens distortions into account

- Chromatic Aberration
- Spherical aberration
- Radial Distortion

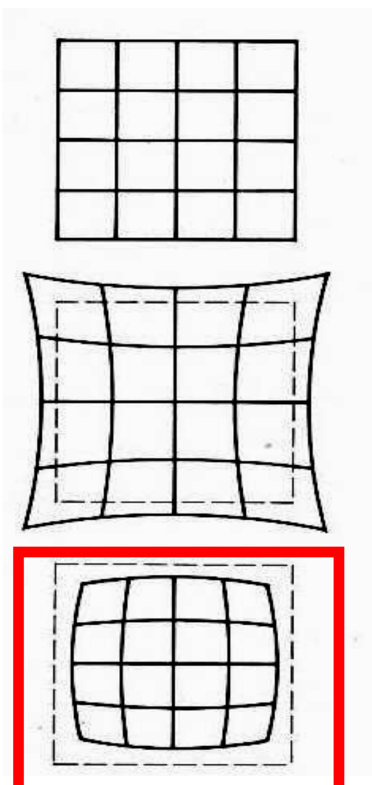






Radial Distortion

- Caused by imperfect lenses
- Deviations are most noticeable for rays that pass through the edge of the lens

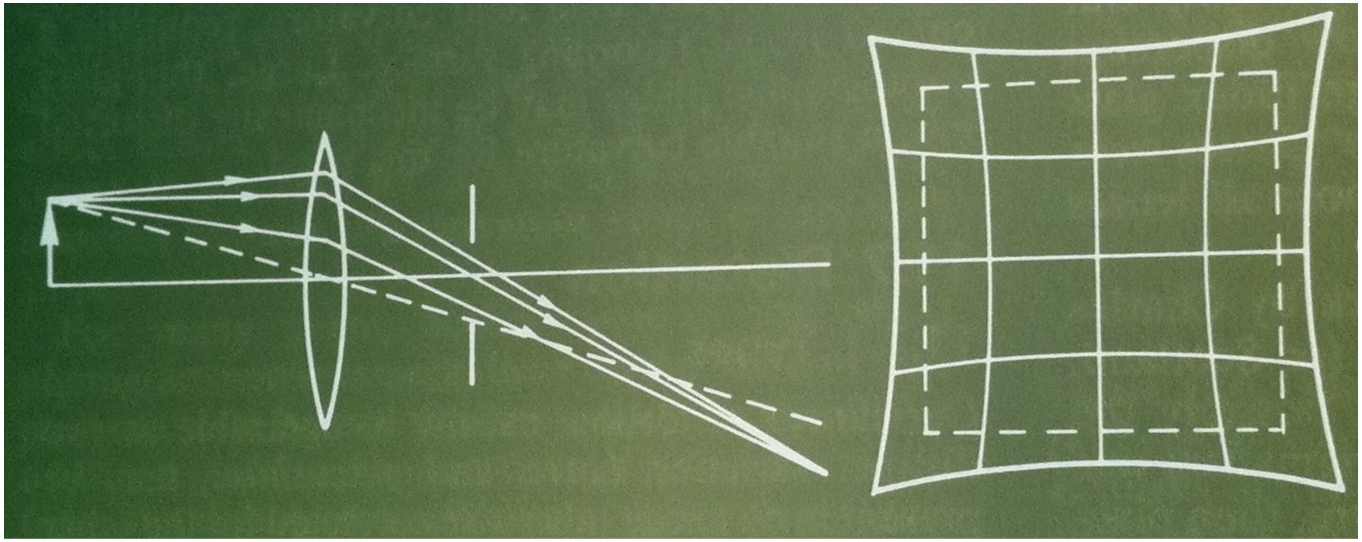


No distortion

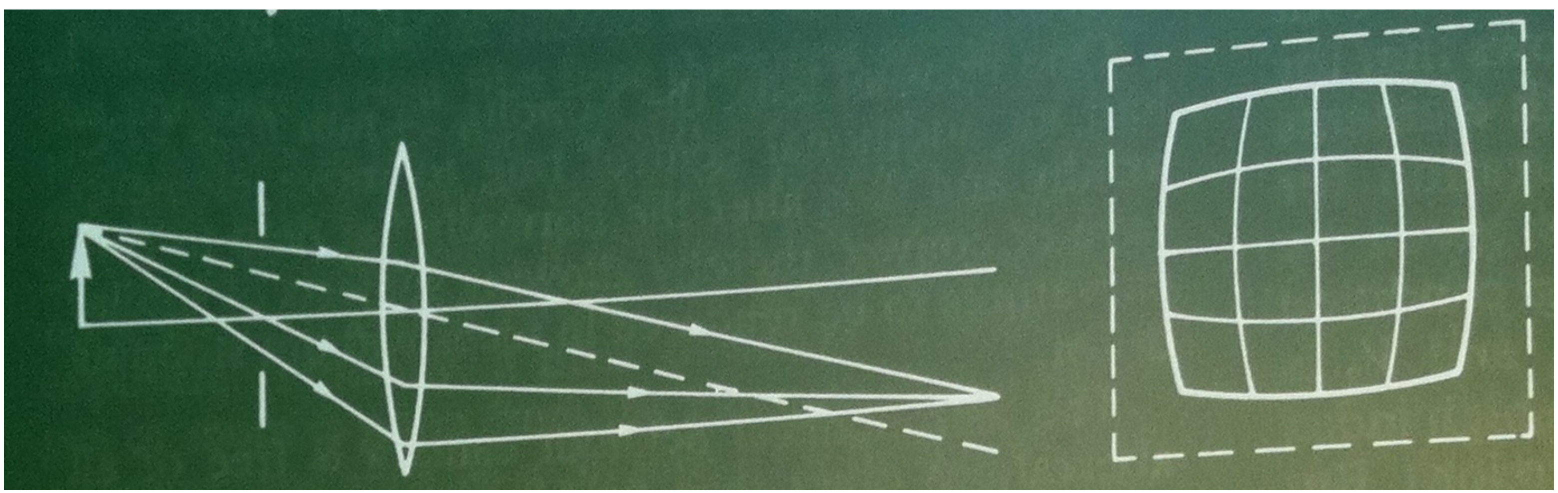
Pin cushion

Barrel

Issues with lenses: Radial Distortion

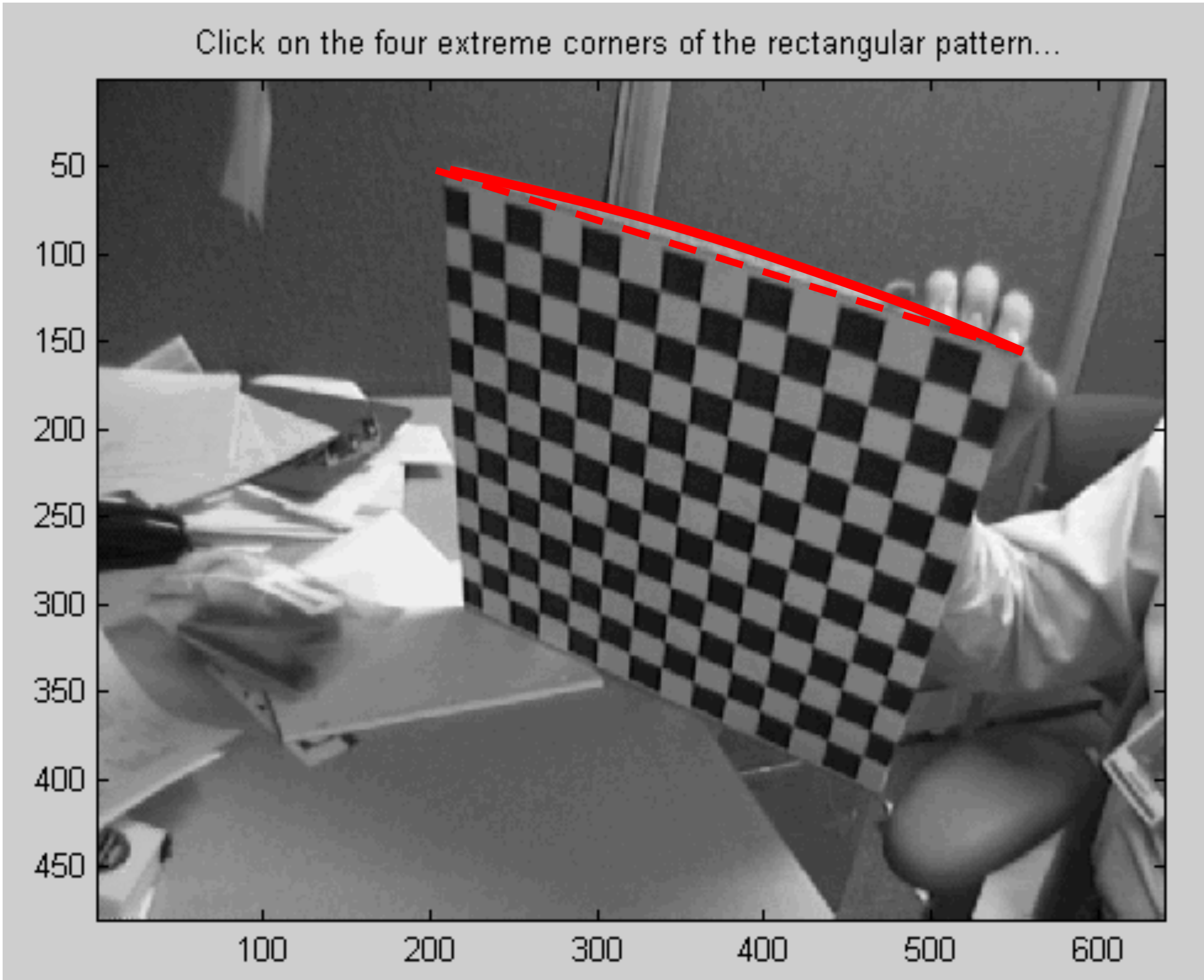


Pin cushion



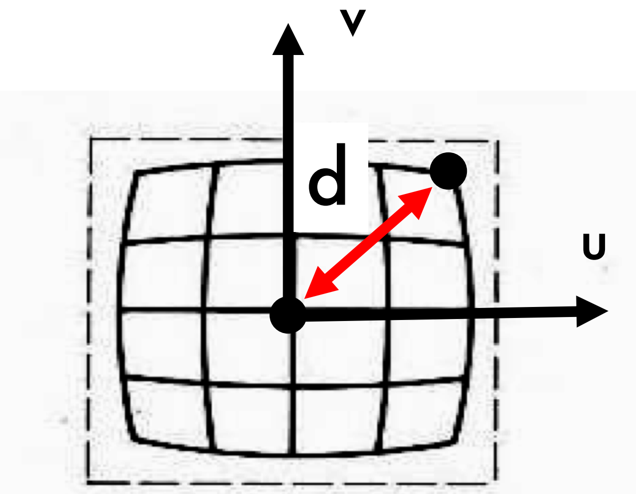
Barrel (fisheye lens)

Radial Distortion



Radial Distortion

Image magnification in(de)creases with distance from the optical center



$$\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M\, P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i$$

$$d^2 = a\,u^2 + b\,v^2 + c\,u\,v$$

To model radial behavior

Distortion coefficient

$$\lambda = \underbrace{1 \pm \sum_{p=1}^{3} \boxed{\kappa_p}\, d^{2p}}_{\text{Polynomial function}}$$

Radial Distortion

$$\underbrace{\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} M}_{Q} P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i \qquad Q = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 P_i}{\mathbf{q}_3 P_i} \\[2ex] \dfrac{\mathbf{q}_2 P_i}{\mathbf{q}_3 P_i} \end{bmatrix} \rightarrow \overbrace{\begin{cases} u_i \mathbf{q}_3 P_i = \mathbf{q}_1 P_i \\ v_i \mathbf{q}_3 P_i = \mathbf{q}_2 P \end{cases}}^{\text{This is not linear system of equations}}$$

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

-Newton Method

-Levenberg-Marquardt Algorithm

- Iterative, starts from initial solution
- May be slow if initial solution far from real solution
- Estimated solution may be function of the initial solution
- Newton requires the computation of J, H
- Levenberg-Marquardt doesn't require the computation of H

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P}_\mathrm{i}}{\mathbf{q}_3 \, \mathrm{P}_\mathrm{i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P}_\mathrm{i}}{\mathbf{q}_3 \, \mathrm{P}_\mathrm{i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

A possible algorithm

1. Solve linear part of the system to find approximated solution
2. Use this solution as initial condition for the full system
3. Solve full system using Newton or L.M.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(P)$$

measurement

parameter

f( ) is nonlinear

Typical assumptions:

- zero-skew, square pixel
- u_o, v_o = known center of the image
- no distortion

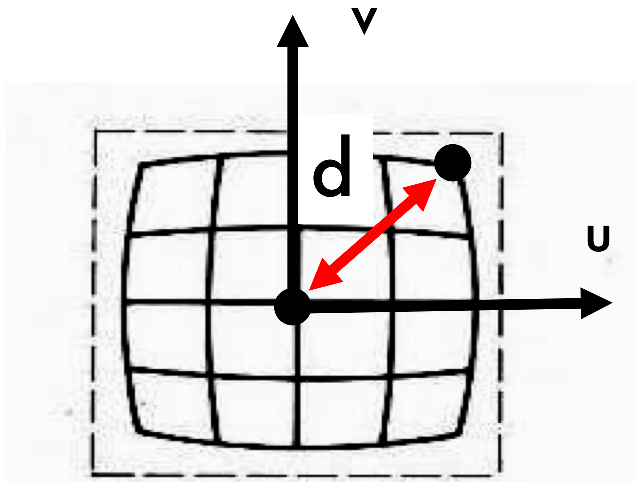
$\longrightarrow$ Just estimate f and R, T

Radial Distortion

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 P_i}{\mathbf{q}_3 P_i} \\ \dfrac{\mathbf{q}_2 P_i}{\mathbf{q}_3 P_i} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

Can estimate m_1 and m_2 and ignore the radial distortion?

Hint:



$$\frac{u_i}{v_i} = \text{slope}$$

Radial Distortion

Estimating m_1 and m_2...

$$\mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 \mathrm{P}_i}{\mathbf{m}_3 \mathrm{P}_i} \\[2ex] \dfrac{\mathbf{m}_2 \mathrm{P}_i}{\mathbf{m}_3 \mathrm{P}_i} \end{bmatrix} \qquad \frac{u_i}{v_i} = \frac{\dfrac{(\mathbf{m}_1 P_i)}{(\mathbf{m}_3 P_i)}}{\dfrac{(\mathbf{m}_2 P_i)}{(\mathbf{m}_3 P_i)}} = \frac{\mathbf{m}_1 P_i}{\mathbf{m}_2 P_i}$$

$$\begin{cases} v_1(\mathbf{m}_1 P_1) - u_1(\mathbf{m}_2 P_1) = 0 \\ v_i(\mathbf{m}_1 P_i) - u_i(\mathbf{m}_2 P_i) = 0 \\ \vdots \\ v_n(\mathbf{m}_1 P_n) - u_n(\mathbf{m}_2 P_n) = 0 \end{cases} \qquad \mathrm{Q}\,\mathbf{n} = 0 \qquad \mathbf{n} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}$$

Tsai technique [87]

Radial Distortion

Once that $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 \, P_i}{\mathbf{m}_3 \, P_i} \\ \dfrac{\mathbf{m}_2 \, P_i}{\mathbf{m}_3 \, P_i} \end{bmatrix}$$

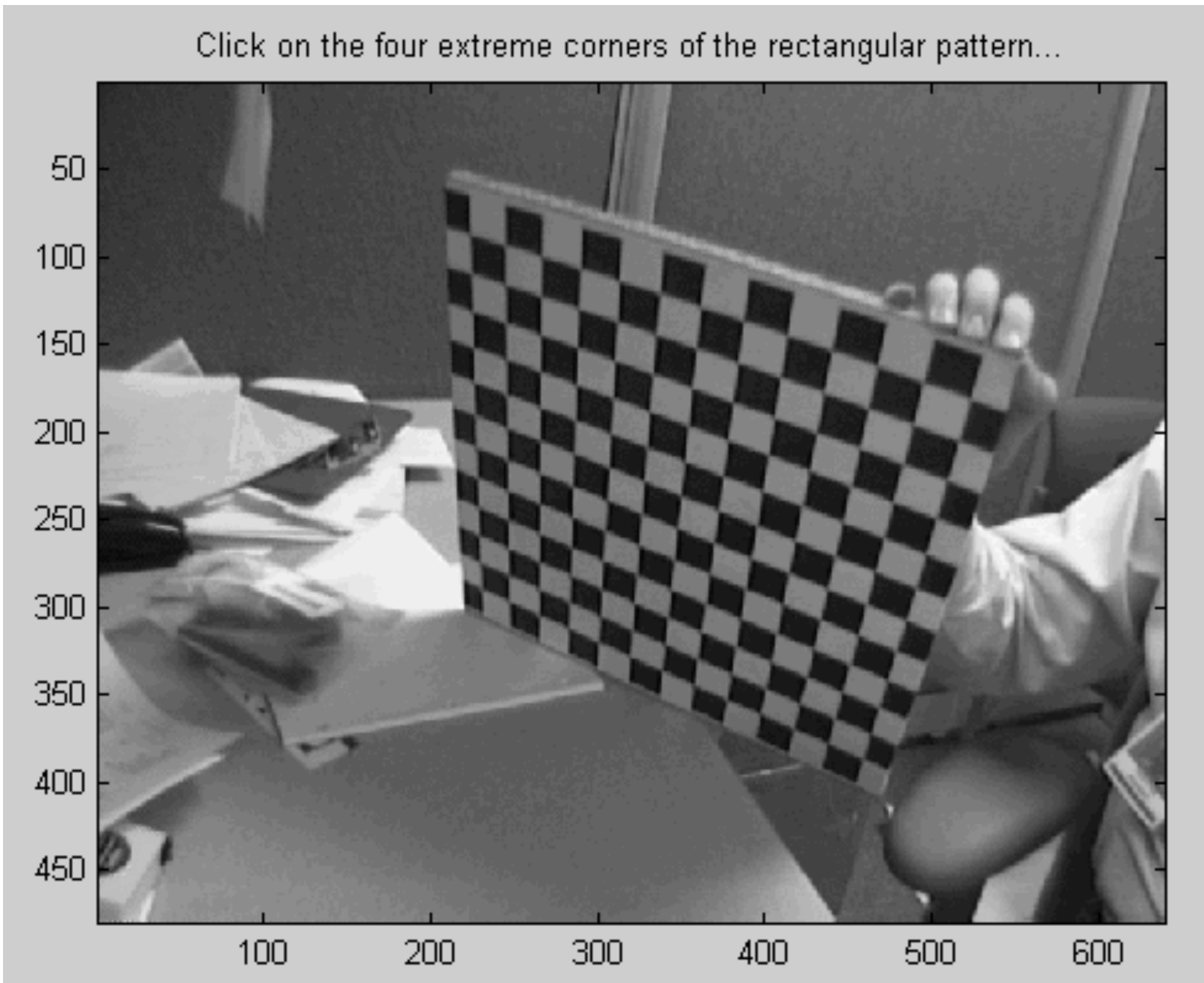
$\mathbf{m}_3$ is non linear function of $\mathbf{m}_1$, $\mathbf{m}_2$, λ

There are some degenerate configurations for which m_1 and m_2 cannot be computed

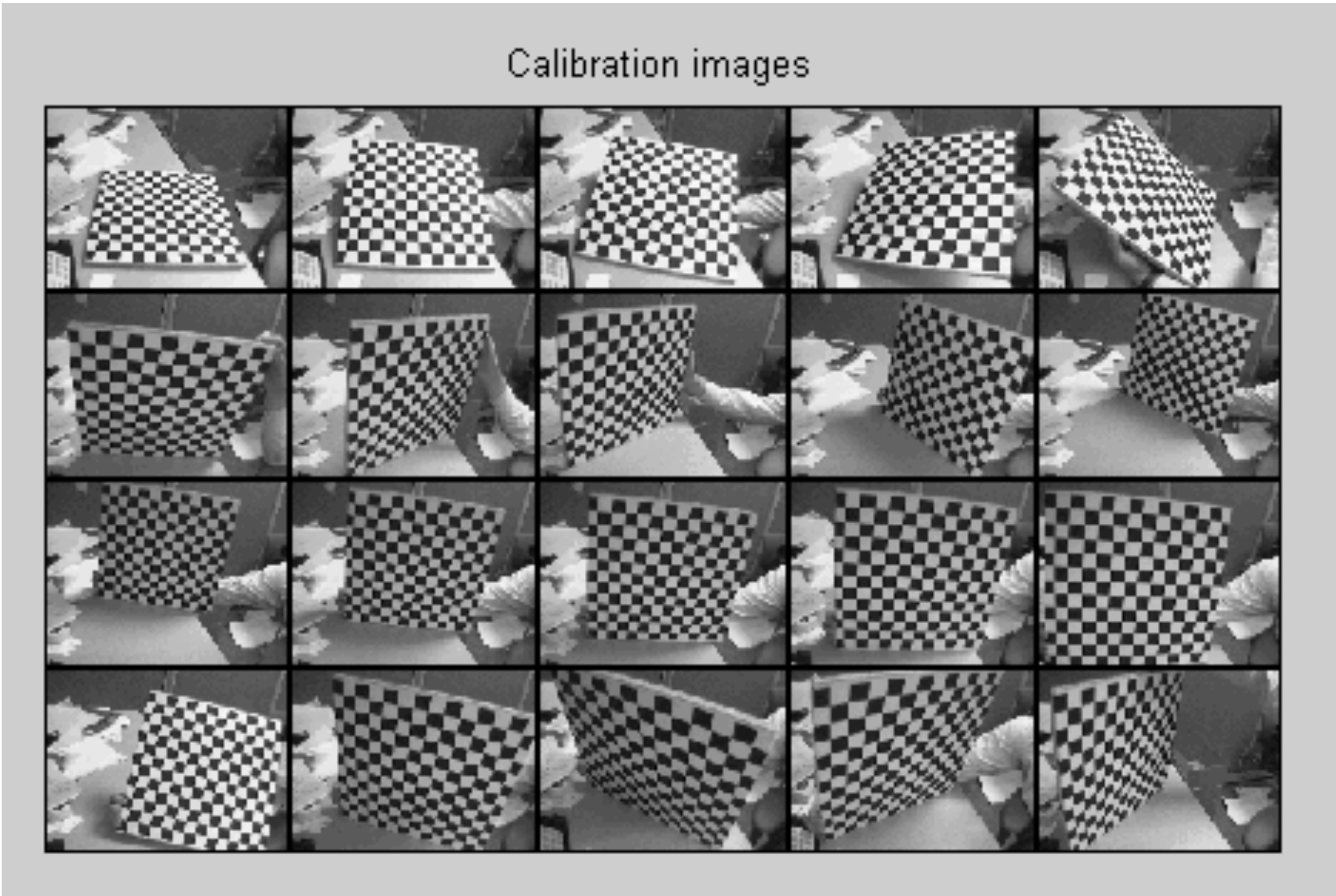
Calibration Procedure

Camera Calibration Toolbox for Matlab
J. Bouguet – [1998-2000]

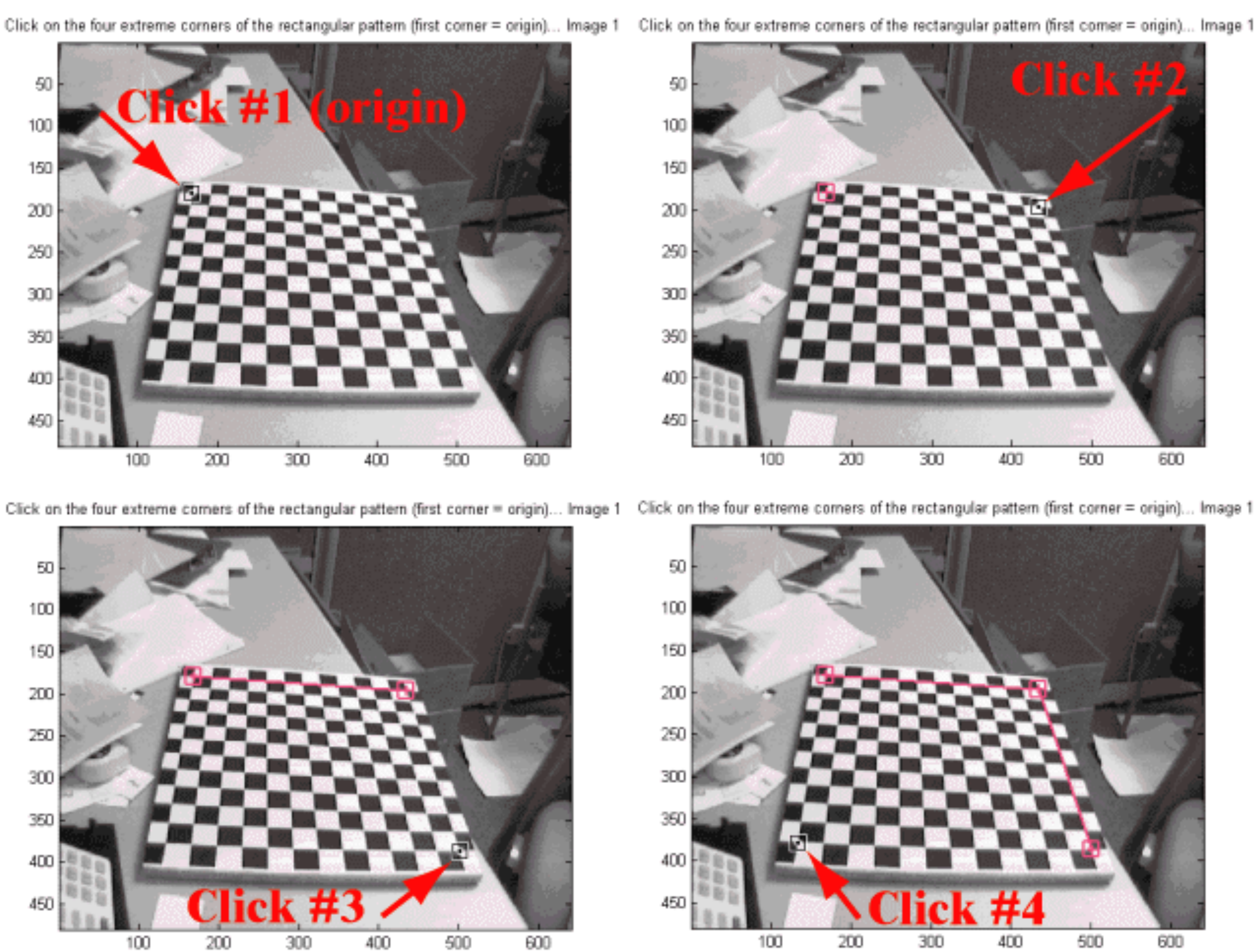
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#examples



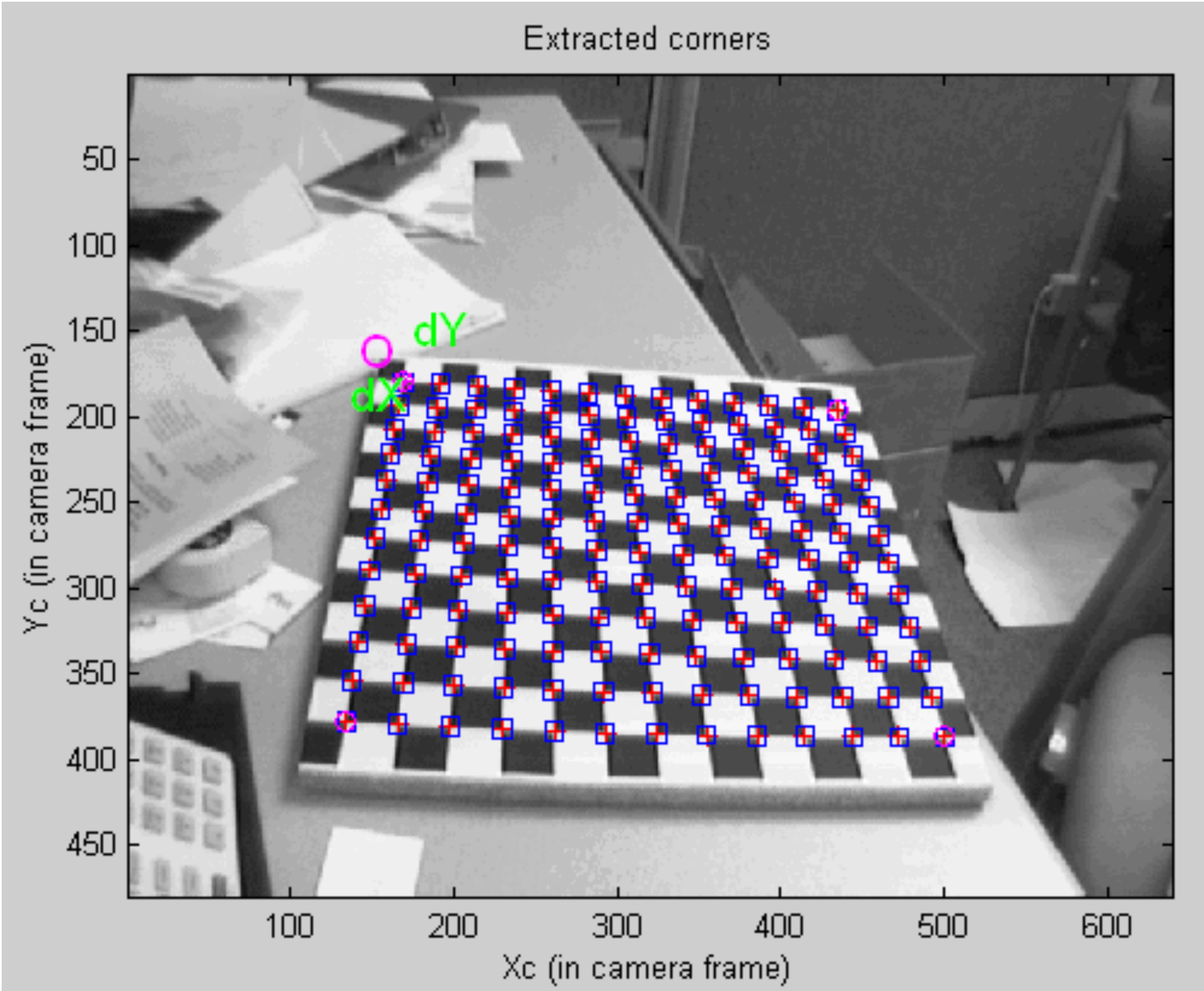
Calibration Procedure



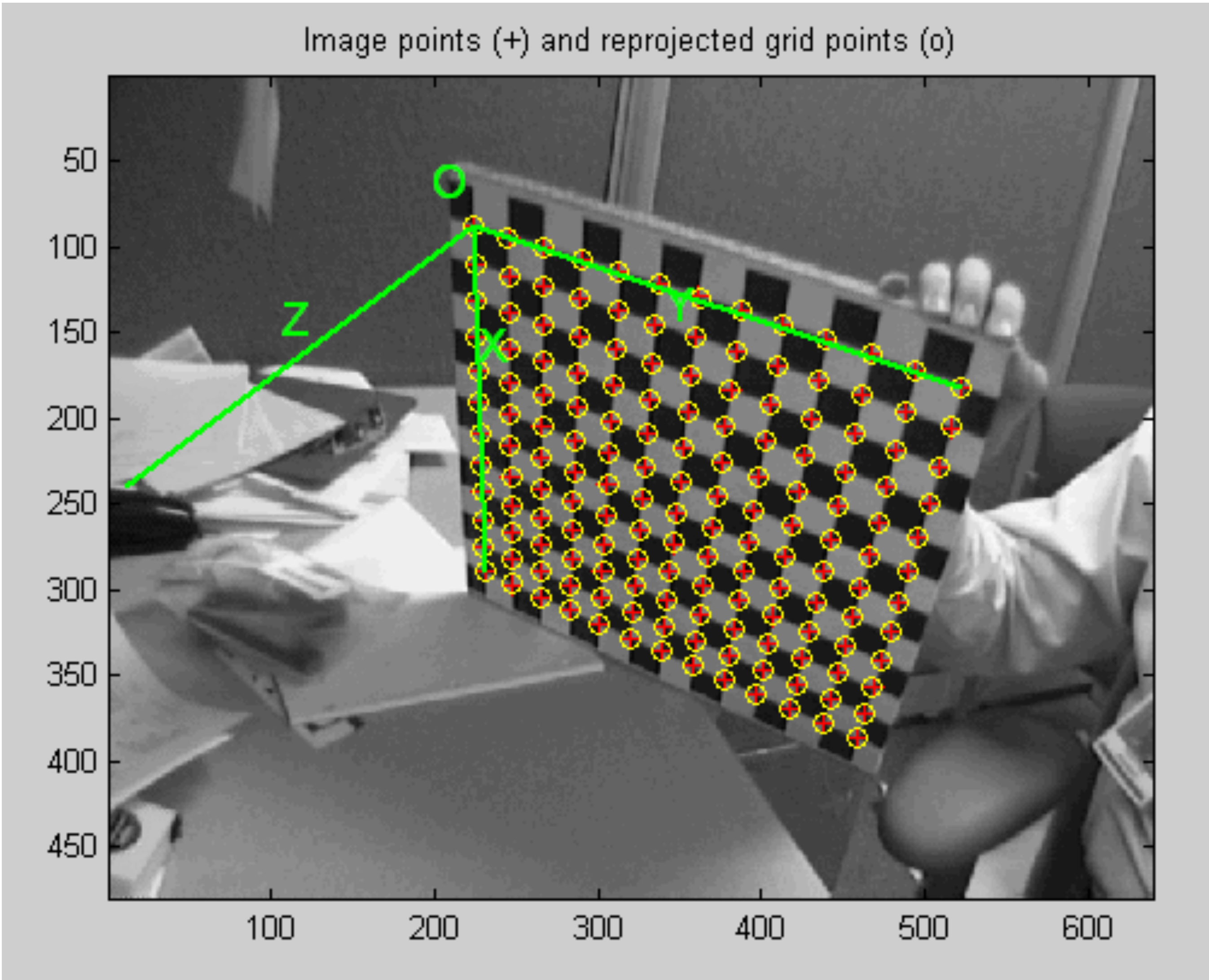
Calibration Procedure



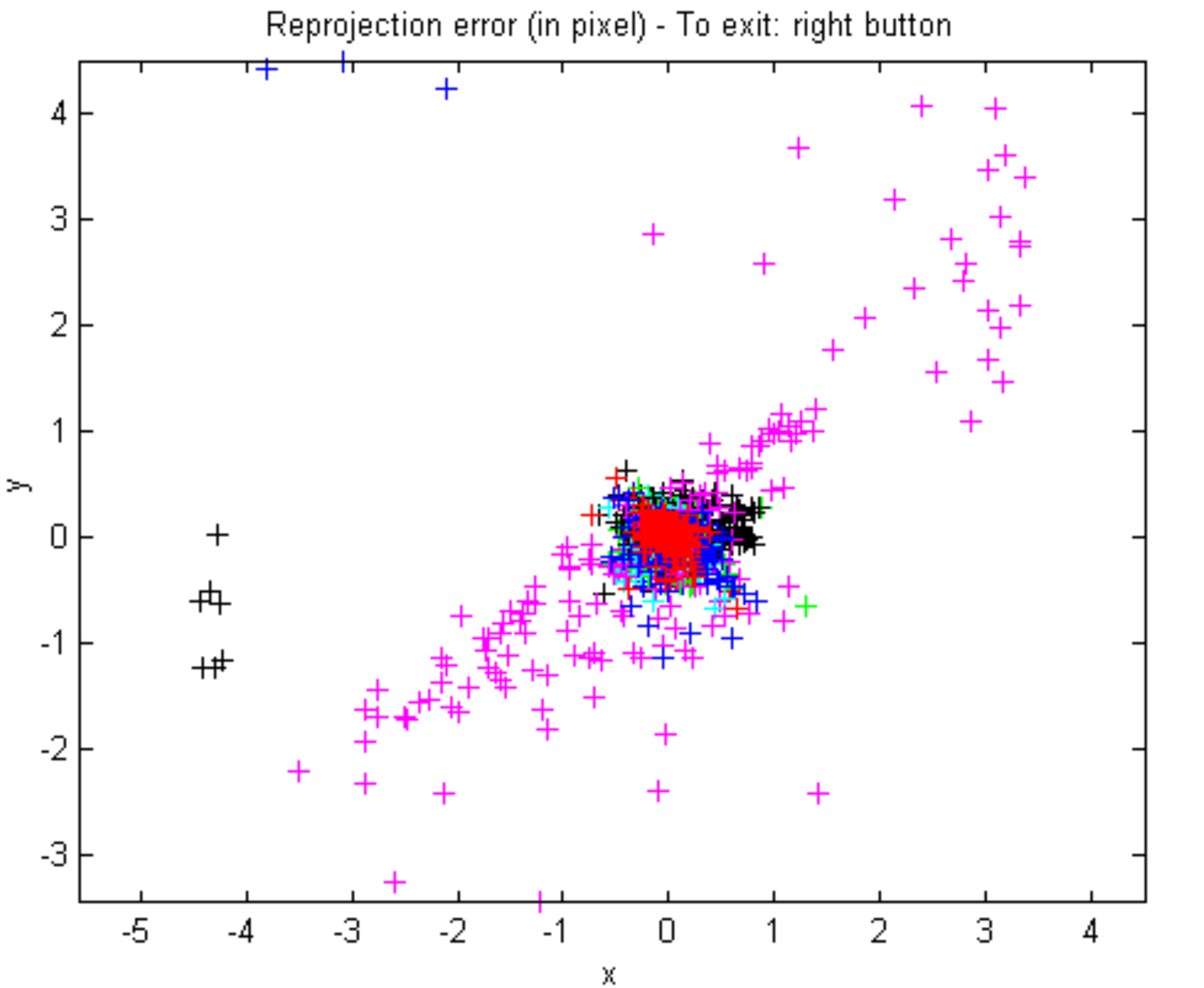
Calibration Procedure



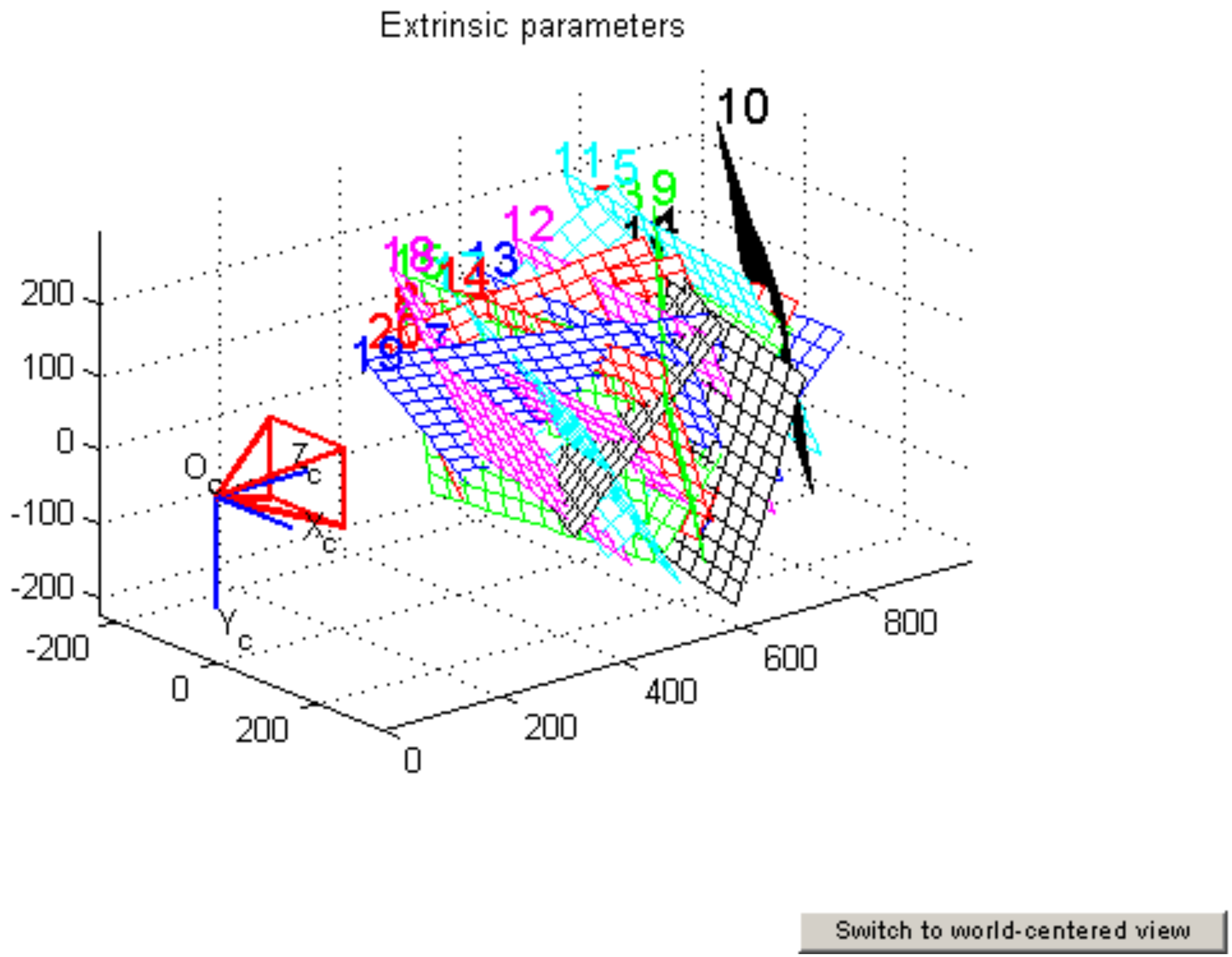
Calibration Procedure



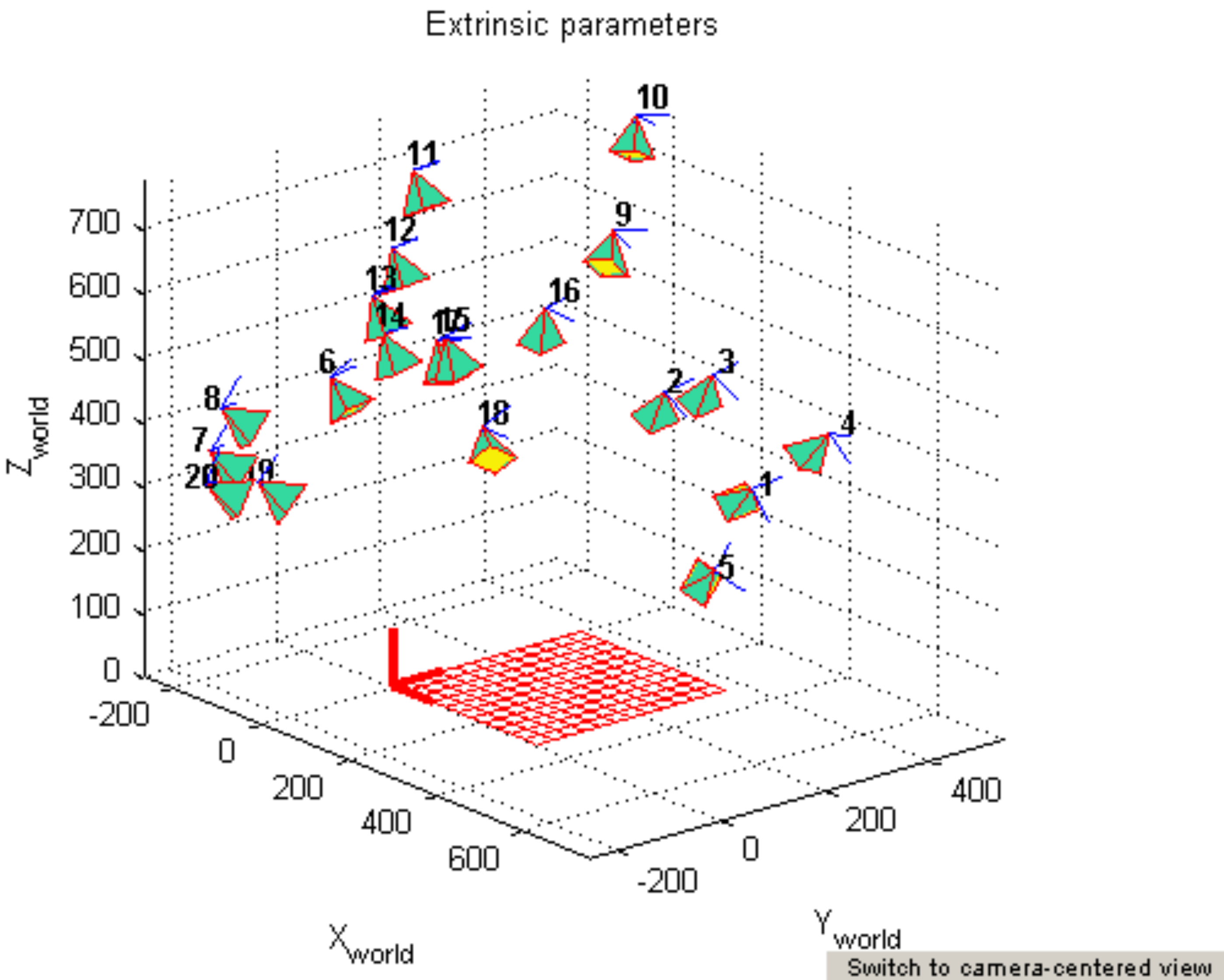
Calibration Procedure



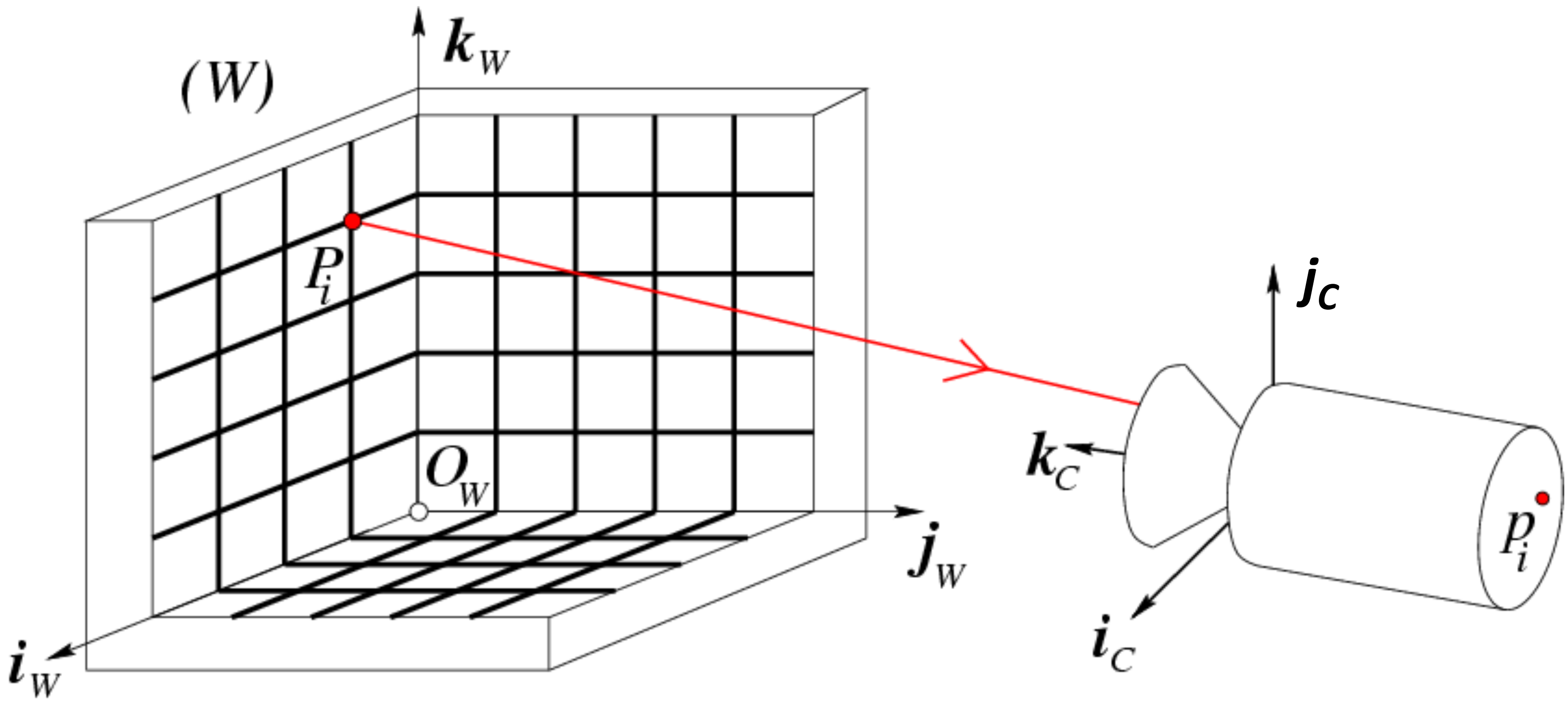
Calibration Procedure



Calibration Procedure



Calibration Problem



$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

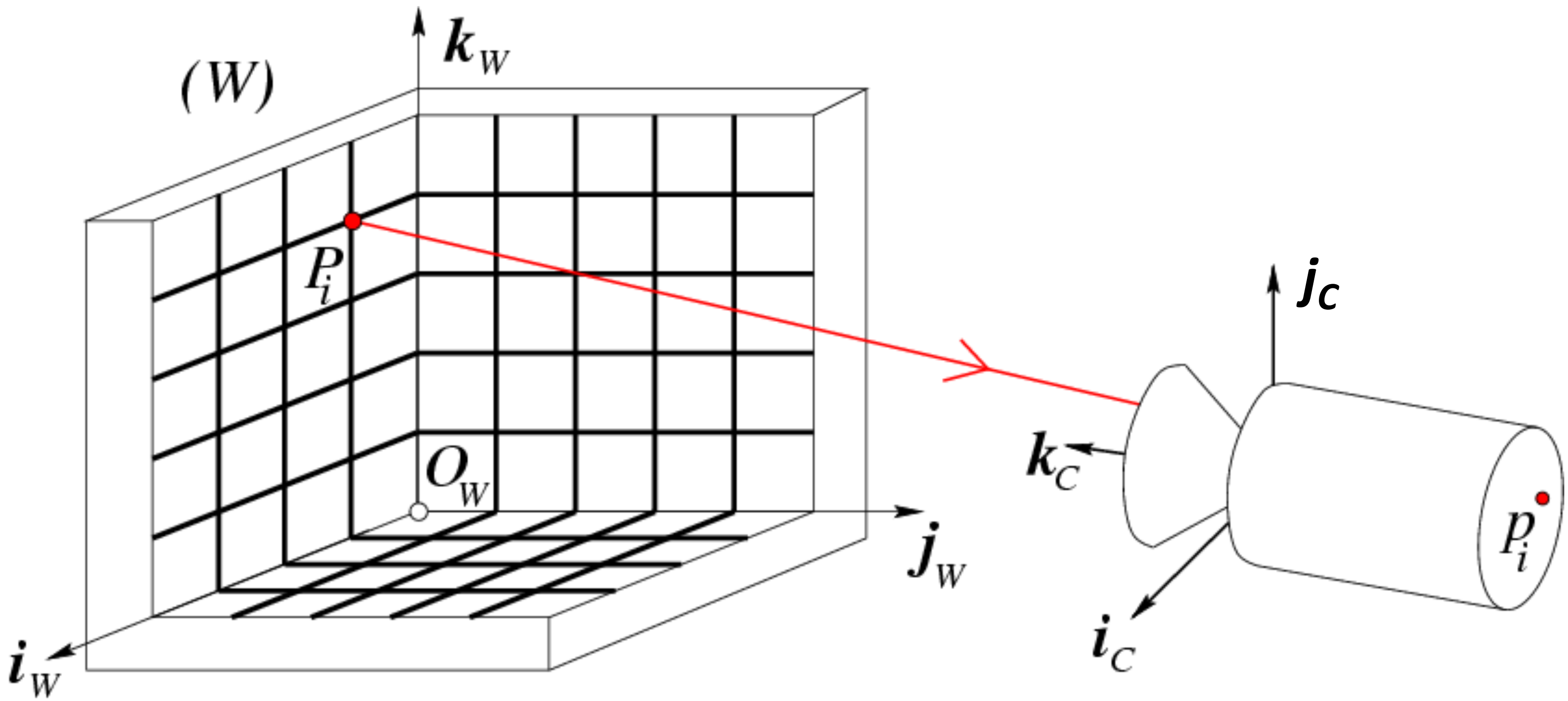
$$P_i \rightarrow M\,P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

World ref. system

In pixels

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

Calibration Problem



$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

11 unknown

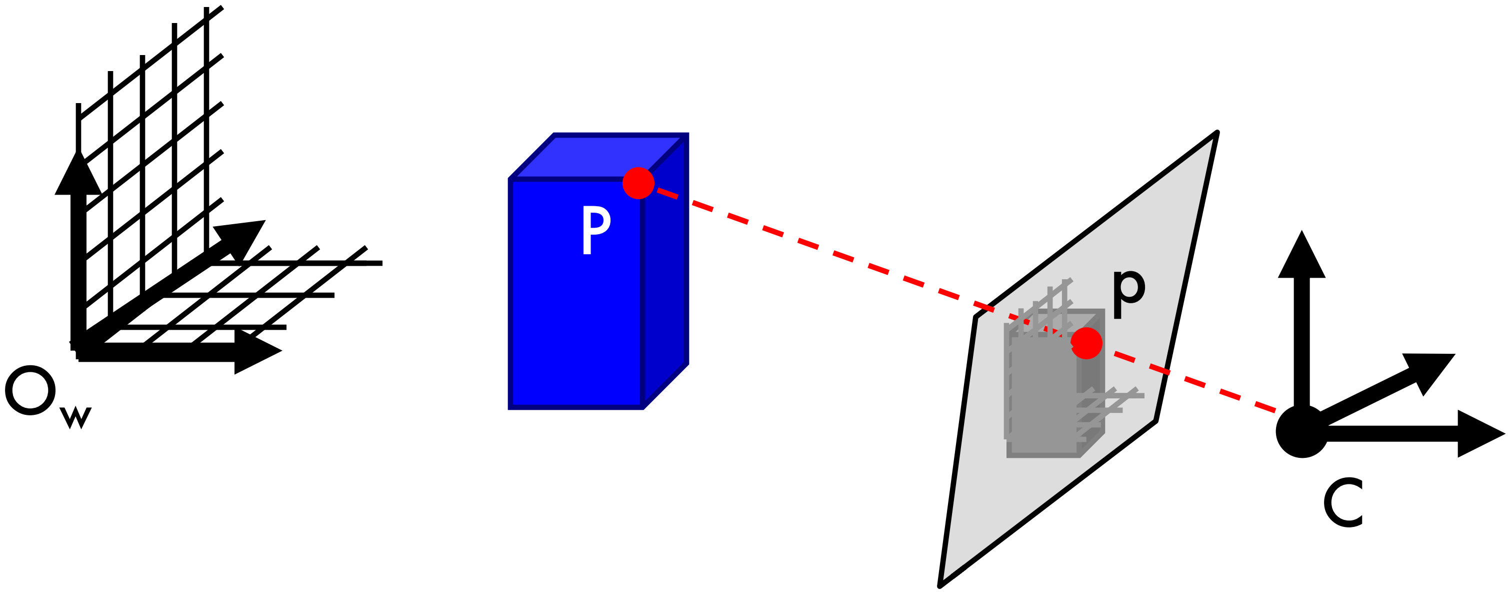
Need at least 6 correspondences

$$P_i \rightarrow M\,P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

World ref. system

In pixels

Once the camera is calibrated...

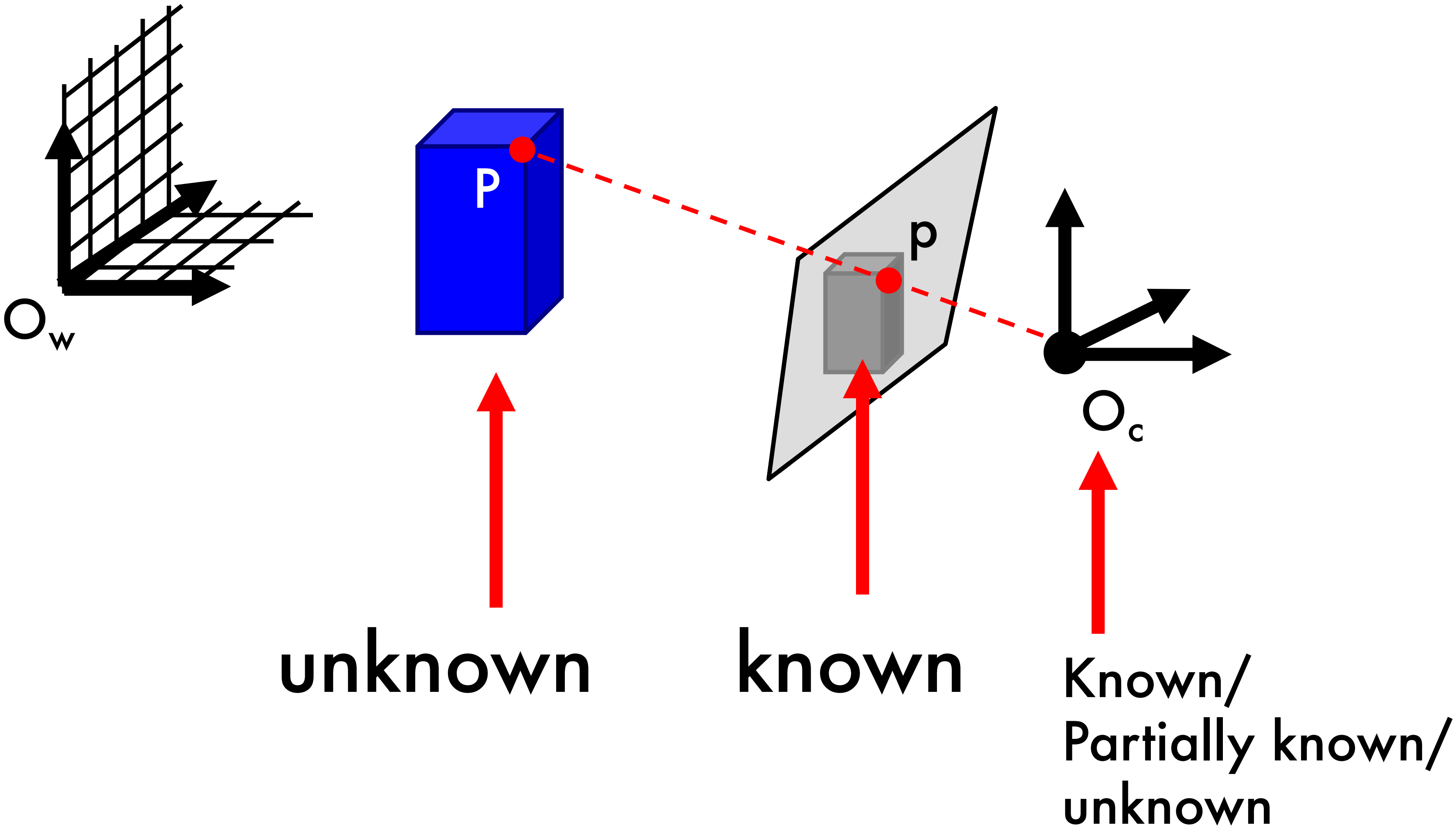


$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

-Internal parameters K are known
-R, T are known – but these can only relate C to the calibration rig

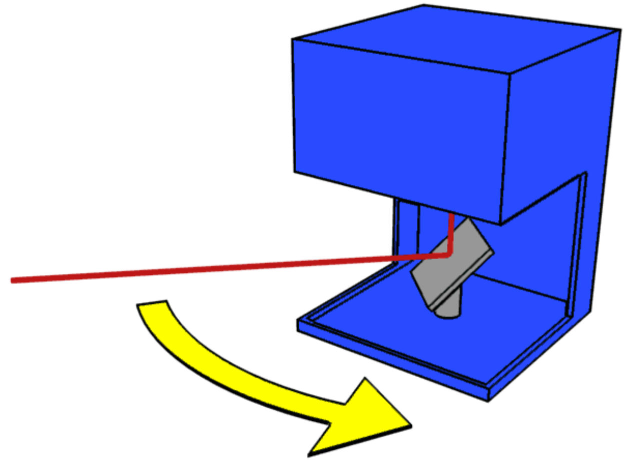
Can I estimate P from the measurement p from a single image?

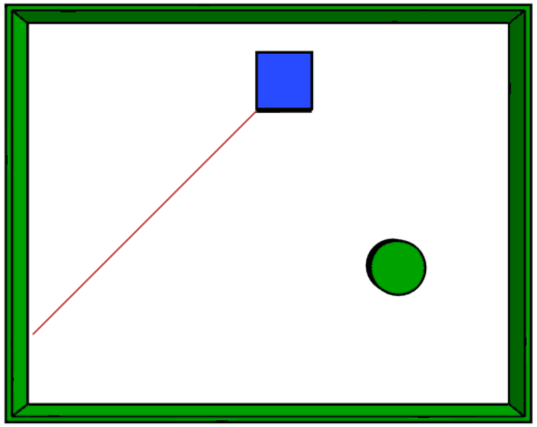
No - in general ☹ [P can be anywhere along the line defined by C and p]

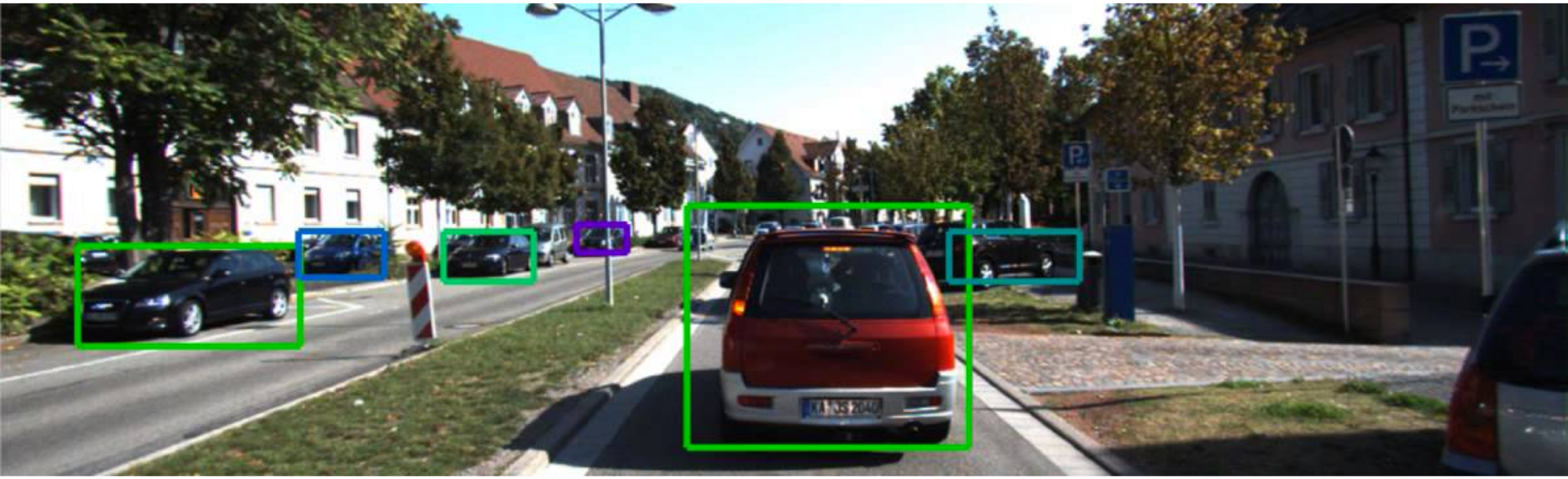


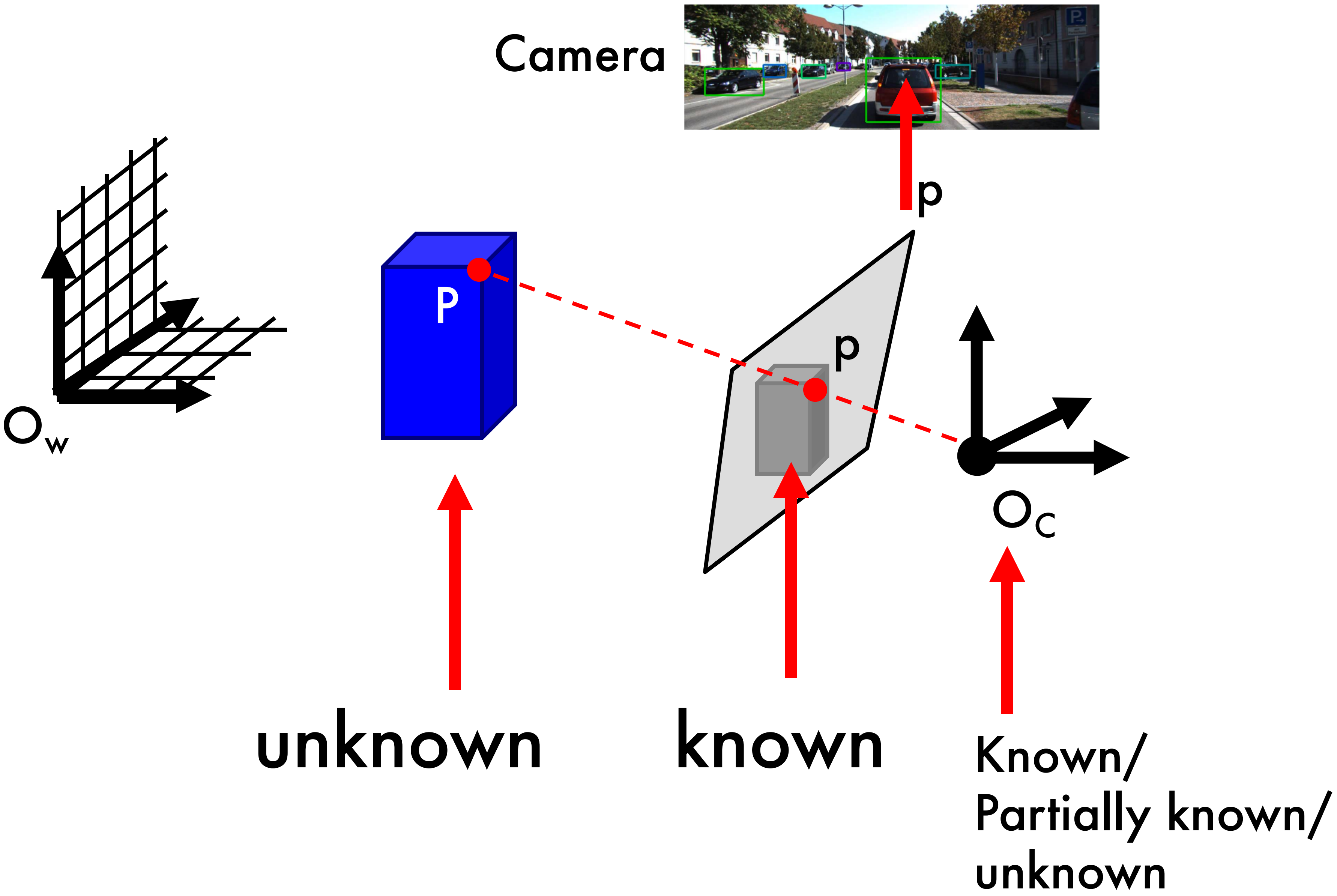
O_w
P
p
O_c
unknown
known
Known/
Partially known/
unknown

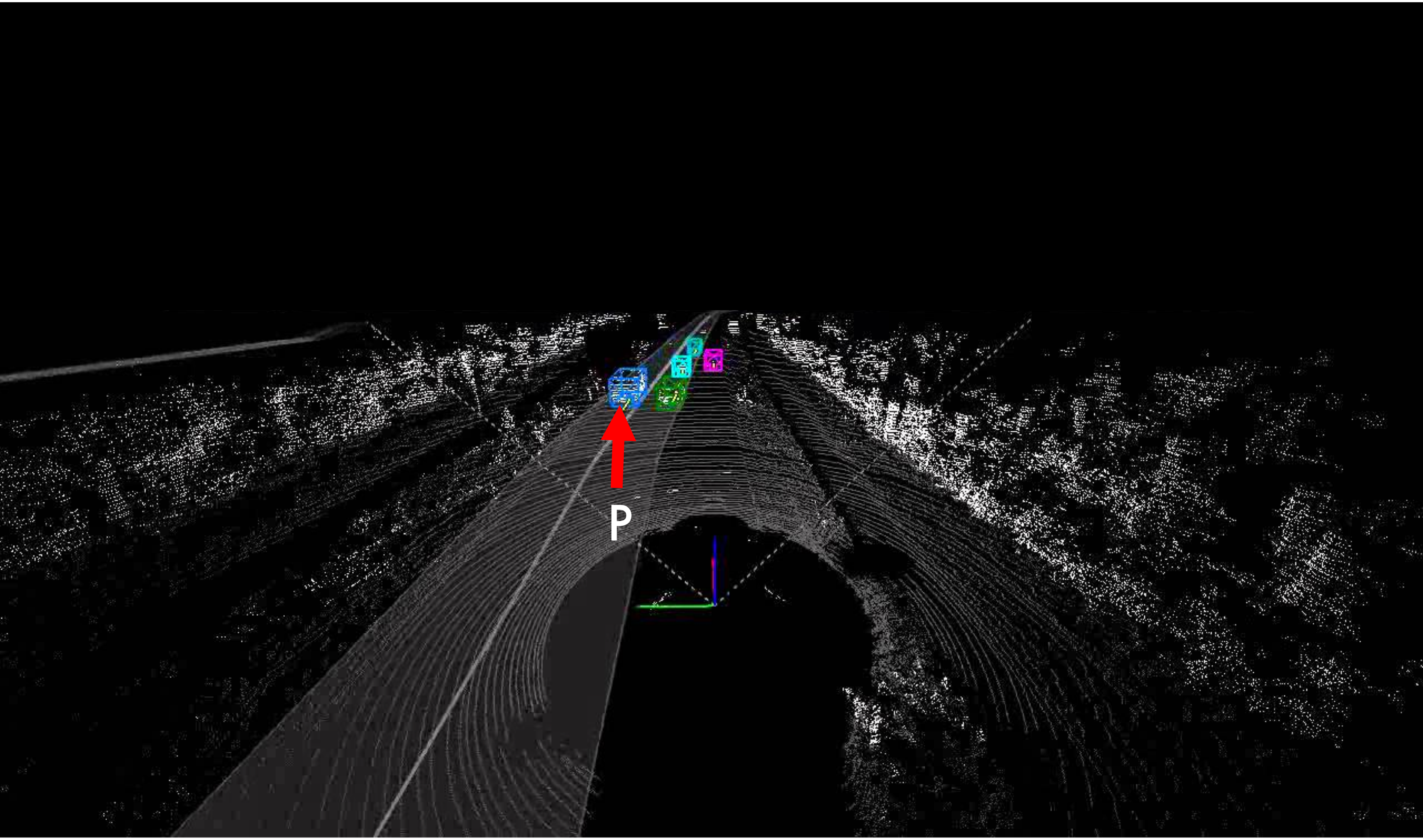
Exteroceptive Sensors



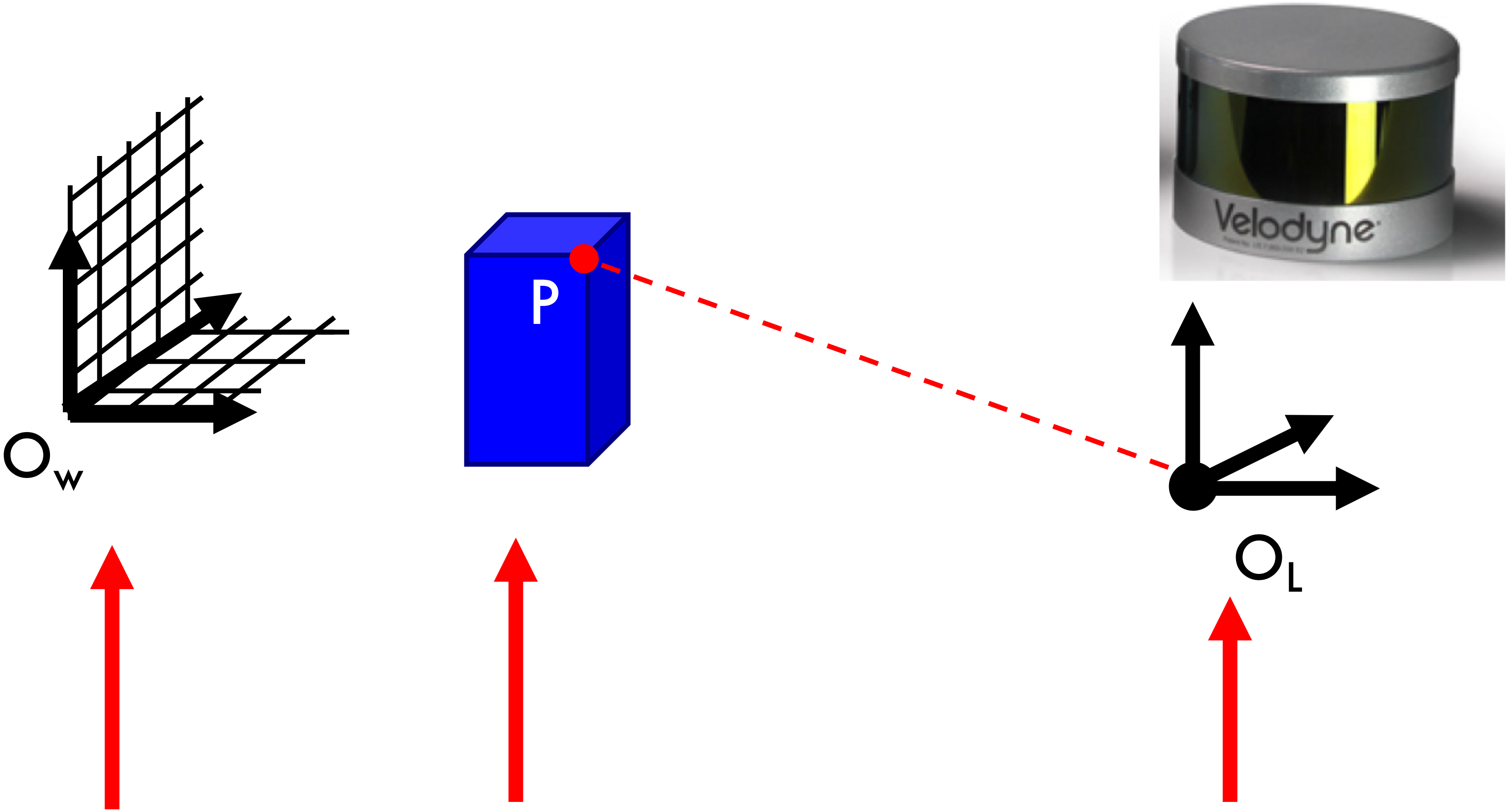





Camera
p
P
p
Ow
Oc
unknown
known
Known/
Partially known/
unknown


P

LIDAR

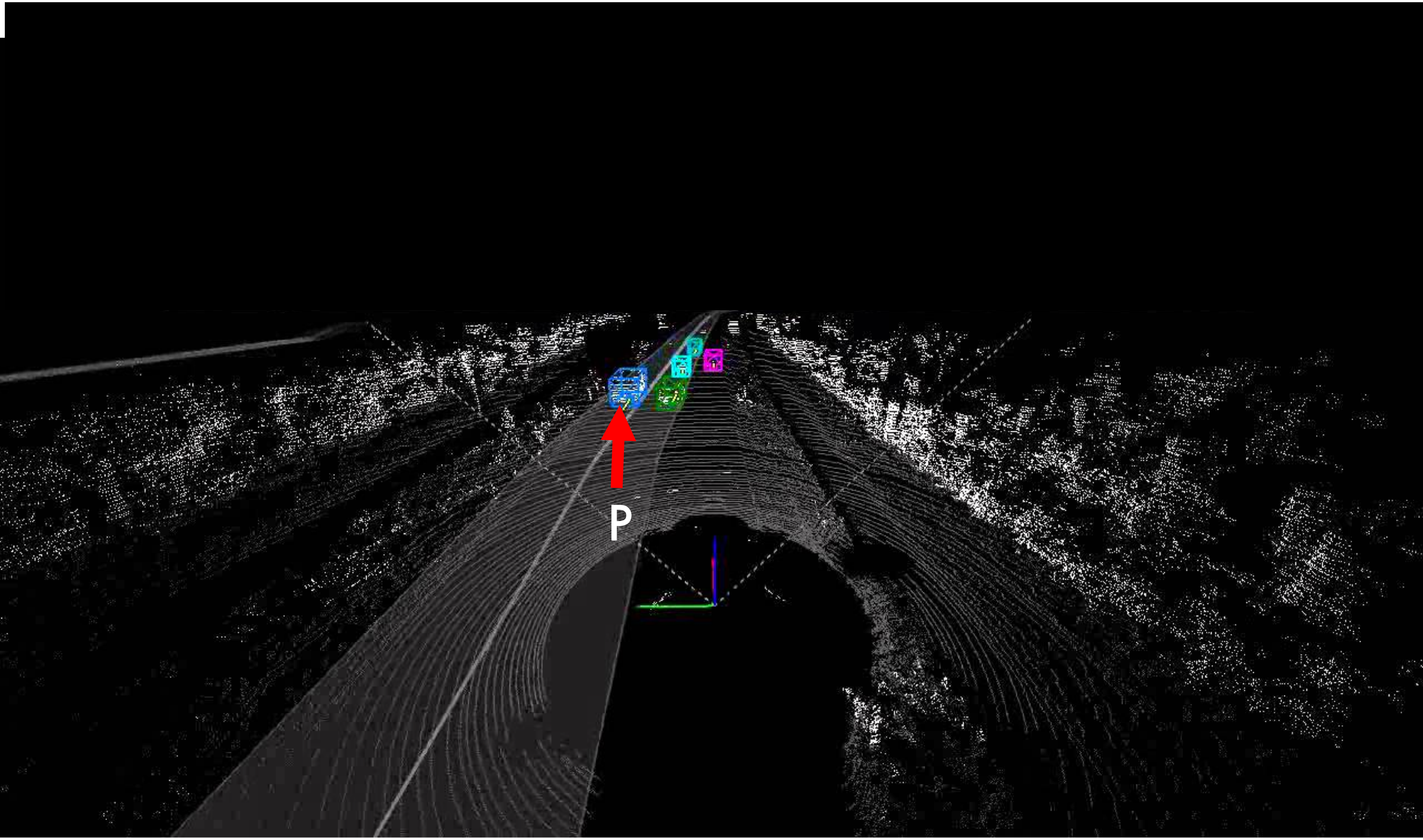


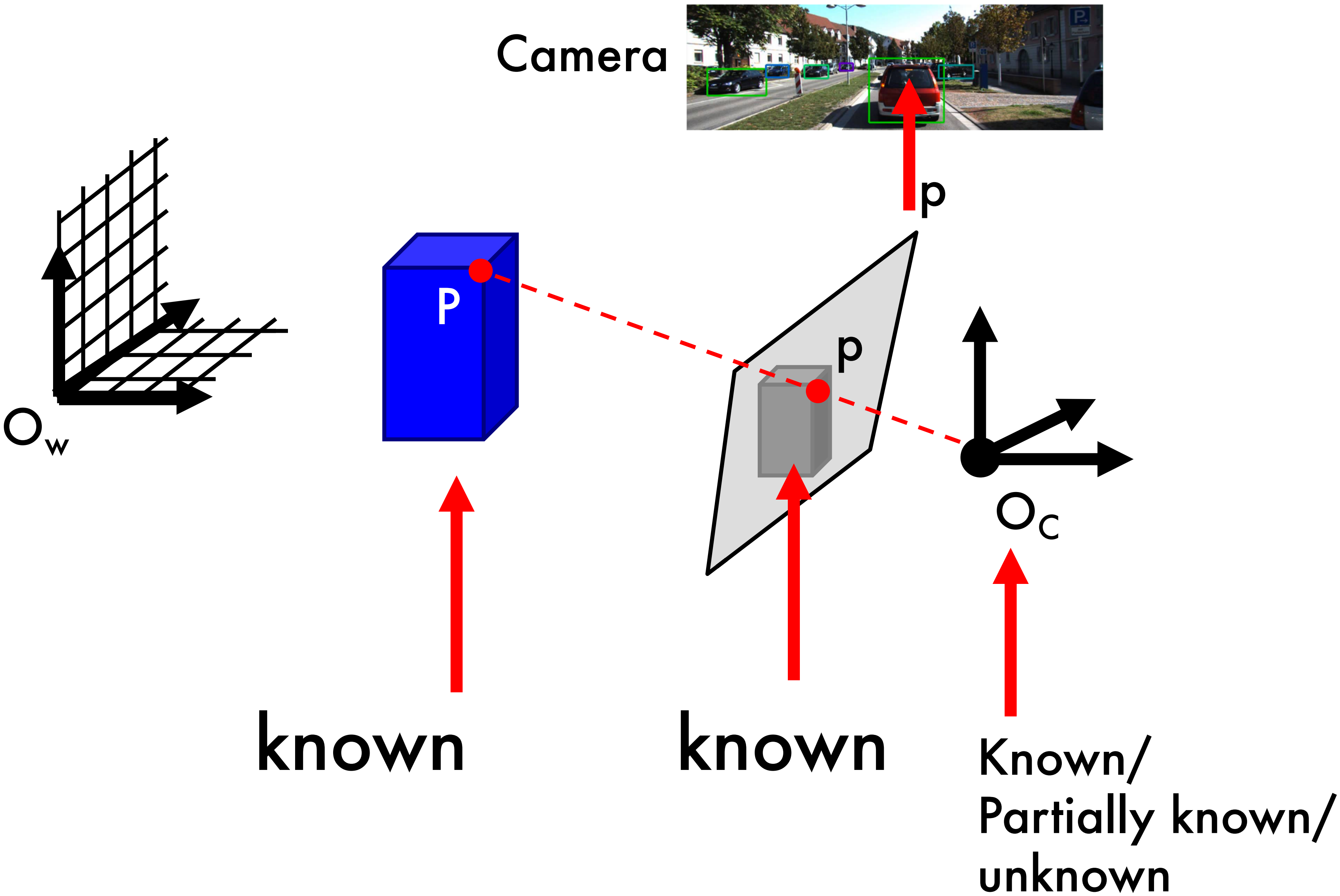
known

Known

$O_w = RT\ O_L$

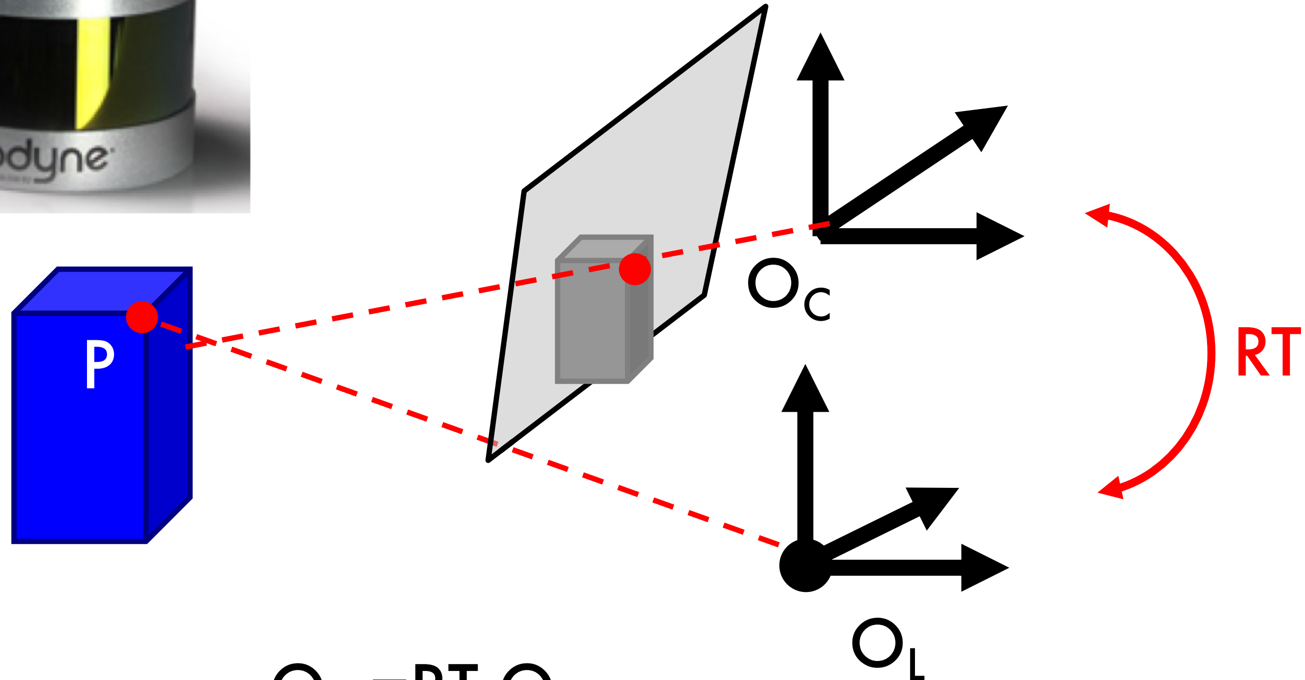
Unknown (Localization Problem...)


P



Camera
p
P
p
Ow
Oc
known
known
Known/
Partially known/
unknown

LIDAR to Camera



$O_C = RT\ O_L$

Unknown
(Calibration Problem...)