AA 274
Principles of Robotic Autonomy

Introduction to computer vision

Stanford University

Autonomous Systems Lab

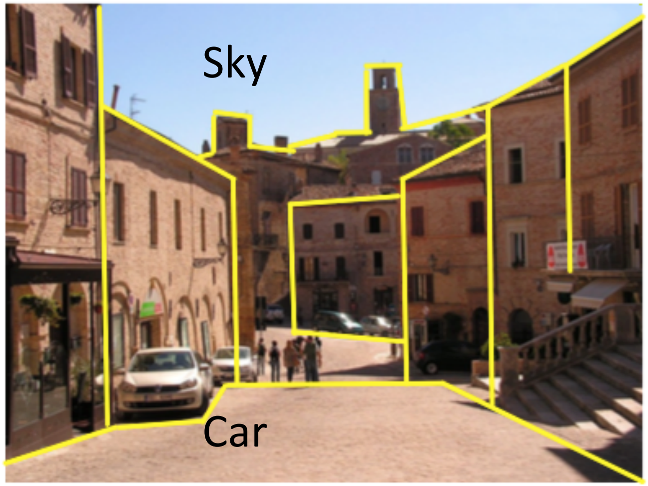
Introduction to computer vision

- Aim
  - Learn about cameras and camera models
  - Learn how to calibrate a camera

- Readings
  - SNS: 4.2.3
  - D. A. Forsyth and J. Ponce [FP]. Computer Vision: A Modern Approach (2nd Edition). Prentice Hall, 2011. Chapter 1.
  - R. Hartley and A. Zisserman [HZ]. Multiple View Geometry in Computer Vision. Academic Press, 2002. Chapter 6.1.
  - Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

Vision

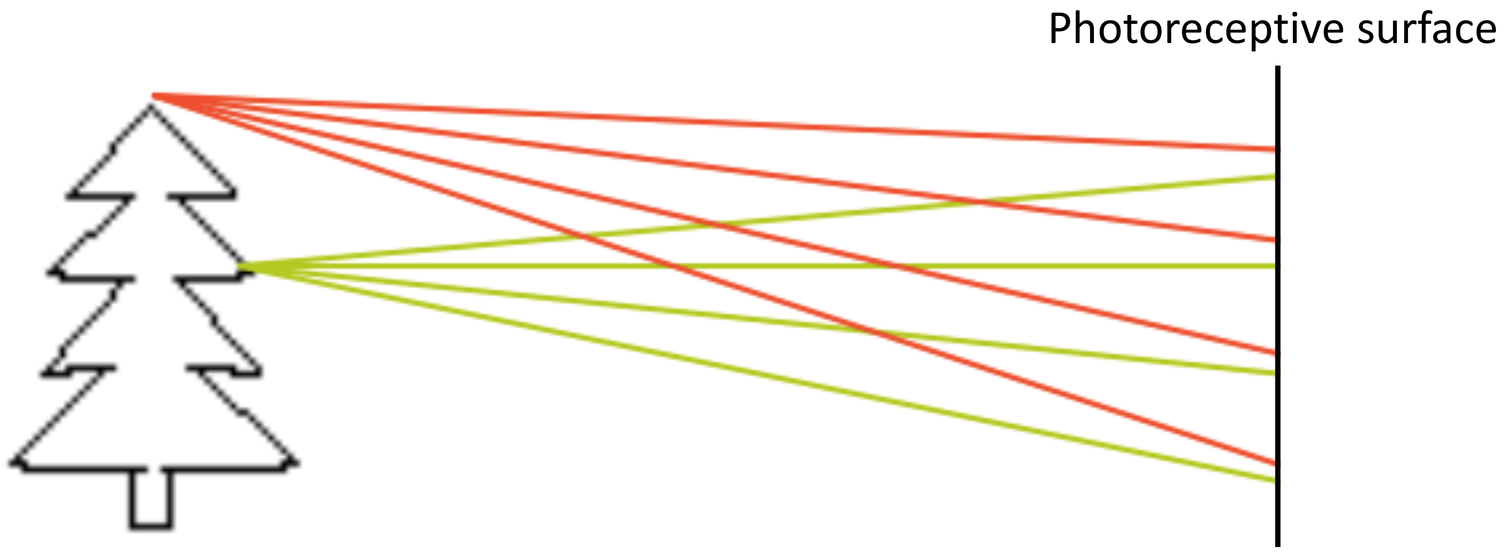
- Vision: ability to interpret the surrounding environment using light in the visible spectrum reflected by objects in the environment
- Human eye: provides enormous amount of information, ~millions of bits per second
- Cameras (e.g., CCD, CMOS): capture light -> convert to digital image -> process to get relevant information (from geometric to semantic)



1. Information extraction
2. Interpretation

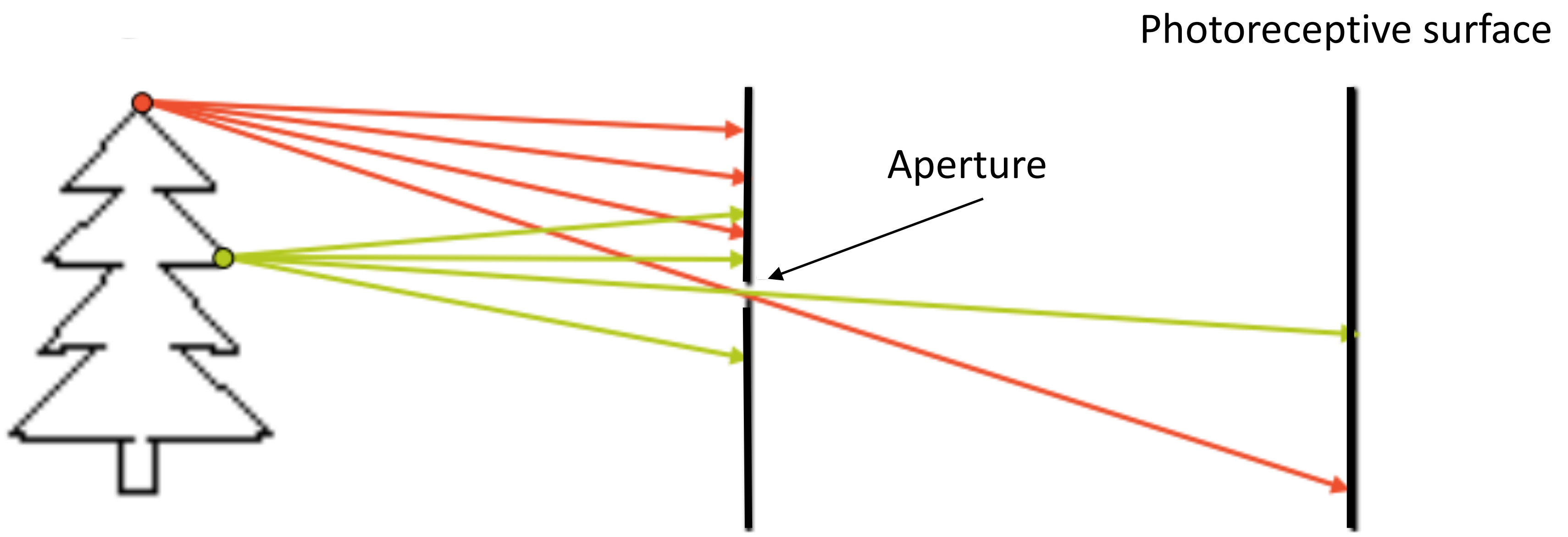
How to capture an image of the world?

- Light is reflected by the object and scattered in all directions
- If we simply add a photoreceptive surface, the captured image will be extremely blurred



Pinhole camera

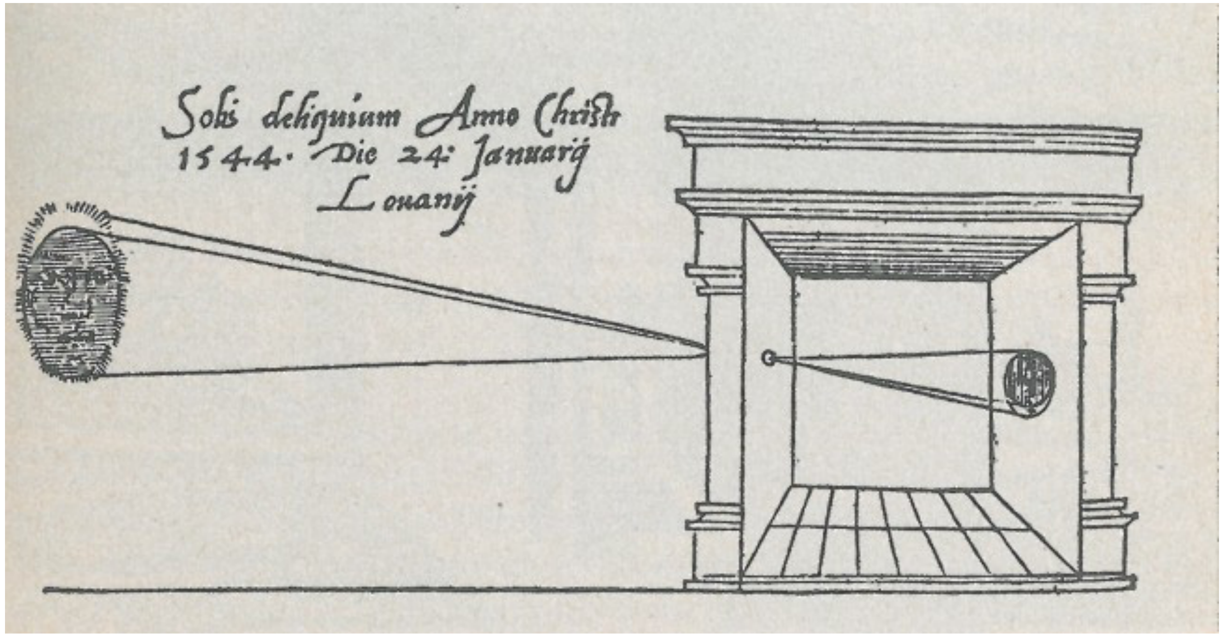
- Idea: add a barrier to block off most of the rays



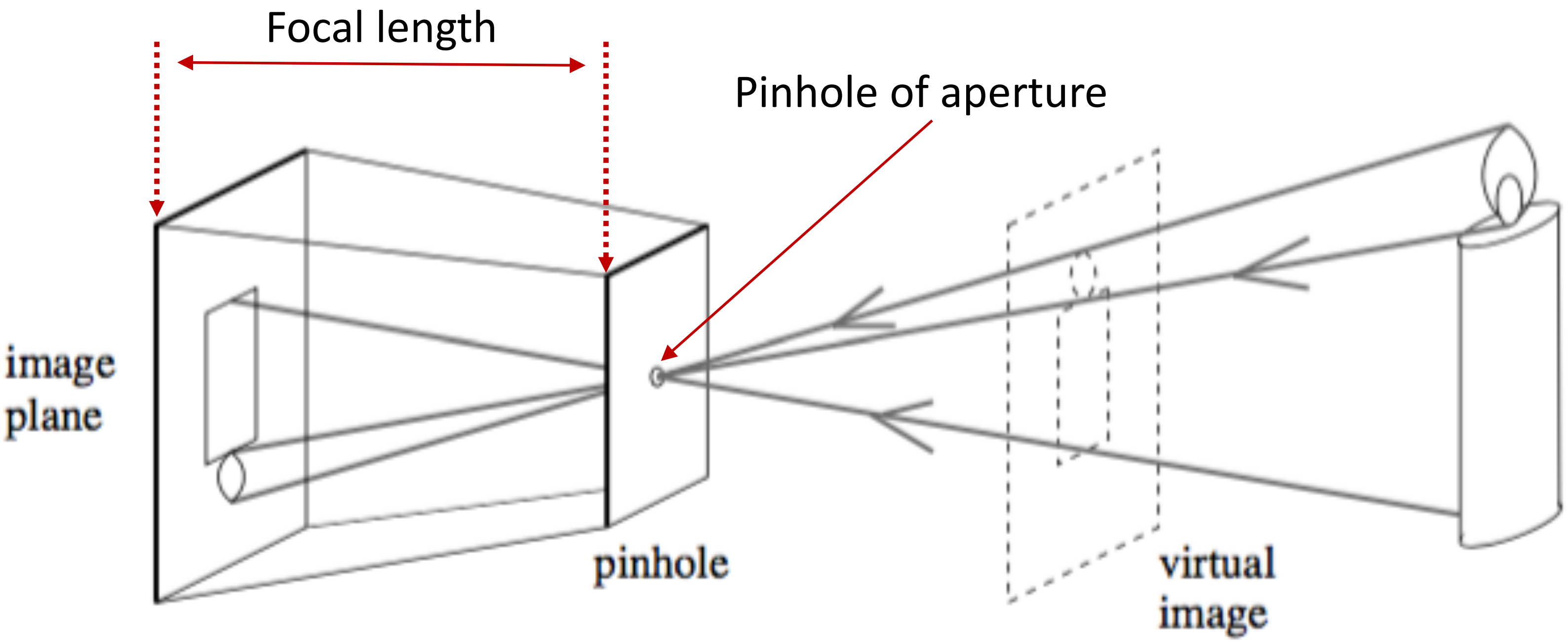
- Pinhole camera: a camera *without a lens* but with a tiny aperture, a *pinhole*

A long history

- Very old idea (several thousands of years BC)
- First clear description from Leonardo Da Vinci (1502)
- Oldest known published drawing of a camera obscura by Gemma Frisius (1544)



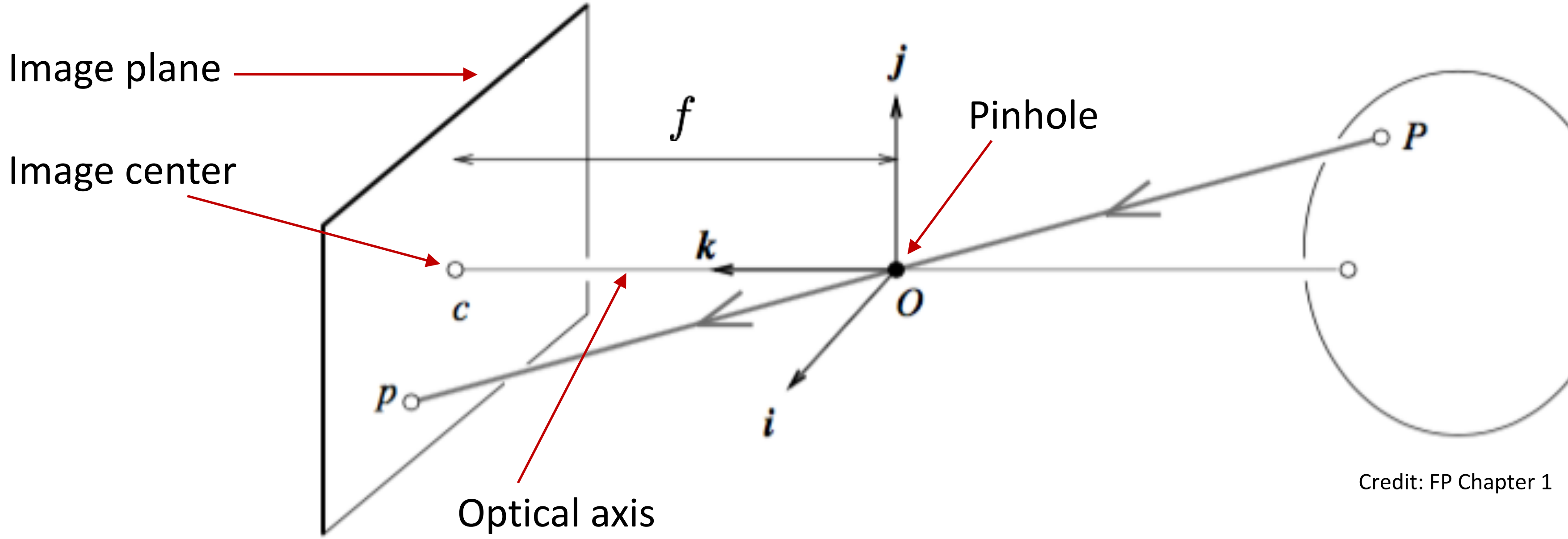
Pinhole camera



Credit: FP Chapter 1

- Perspective projection creates inverted images
- Sometimes it is convenient to consider a *virtual image* associated with a plane lying in front of the pinhole
- Virtual image not inverted but otherwise equivalent to the actual one

Pinhole perspective



$$P = (X, Y, Z)$$

Perspective

$$p = (x, y, z)$$

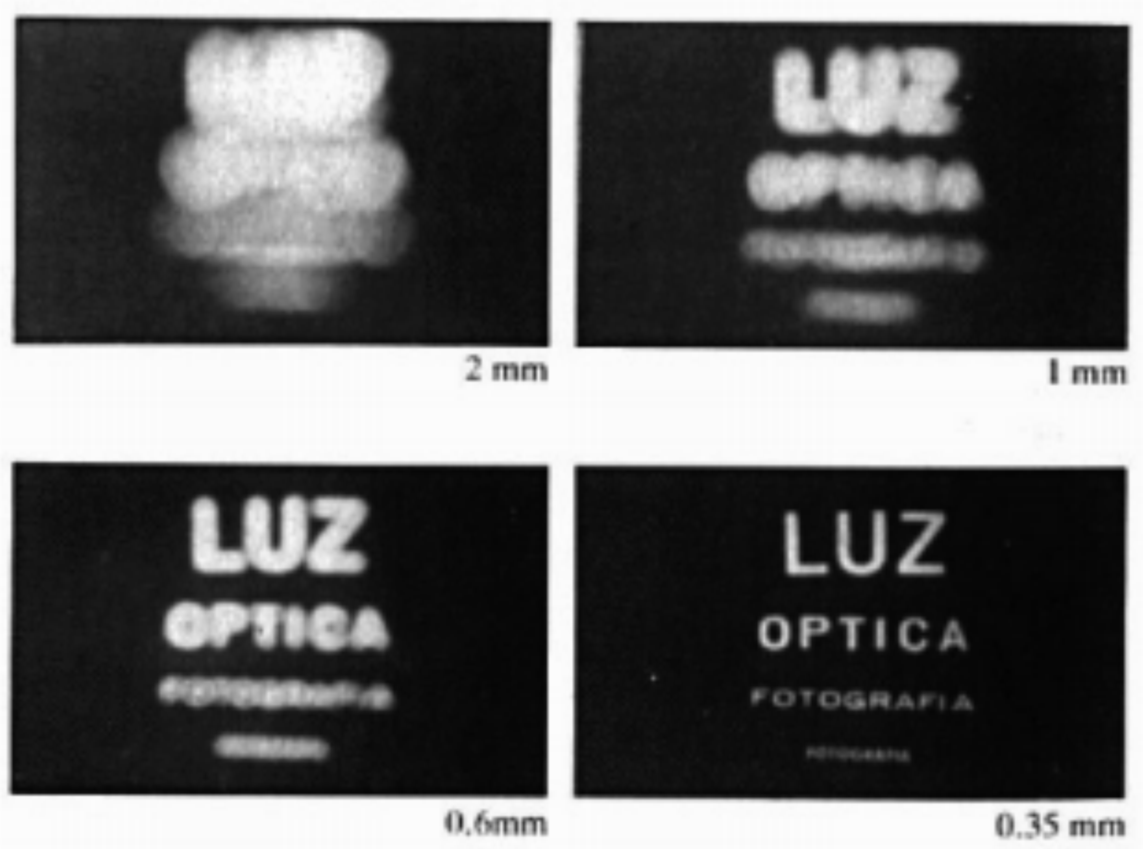
- Since P, O, and p are collinear: $\overline{Op} = \lambda \overline{OP}$ for some $\lambda \in R$
- Also, $z=f$, hence

$$\begin{cases} x = \lambda X \\ y = \lambda Y \\ z = \lambda Z \end{cases} \Leftrightarrow \lambda = \frac{x}{X} = \frac{y}{Y} = \frac{z}{Z} \quad \Rightarrow \quad \begin{cases} x = f\,\frac{X}{Z} \\ y = f\,\frac{Y}{Z} \end{cases}$$

Issues with pinhole camera

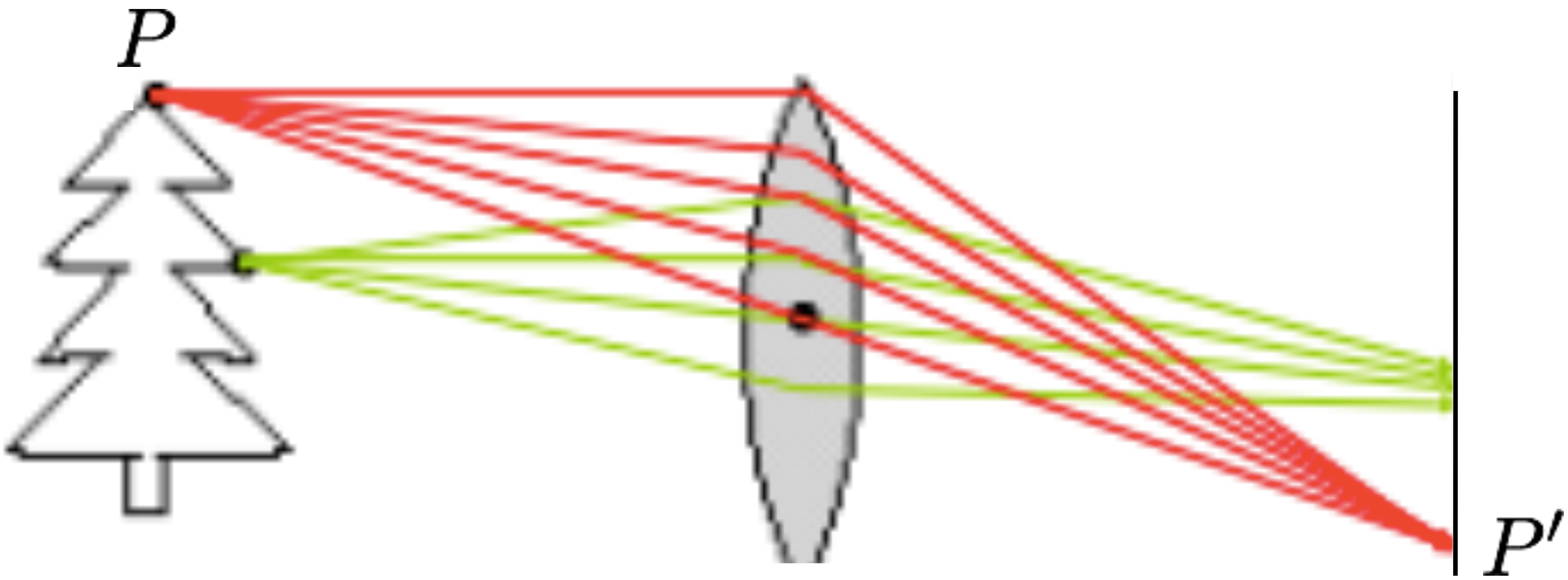
- Larger aperture -> greater number of light rays that pass through the aperture -> blur
- Smaller aperture -> fewer number of light rays that pass through the aperture -> darkness (+ diffraction)

- Solution: add a lens to replace the aperture!

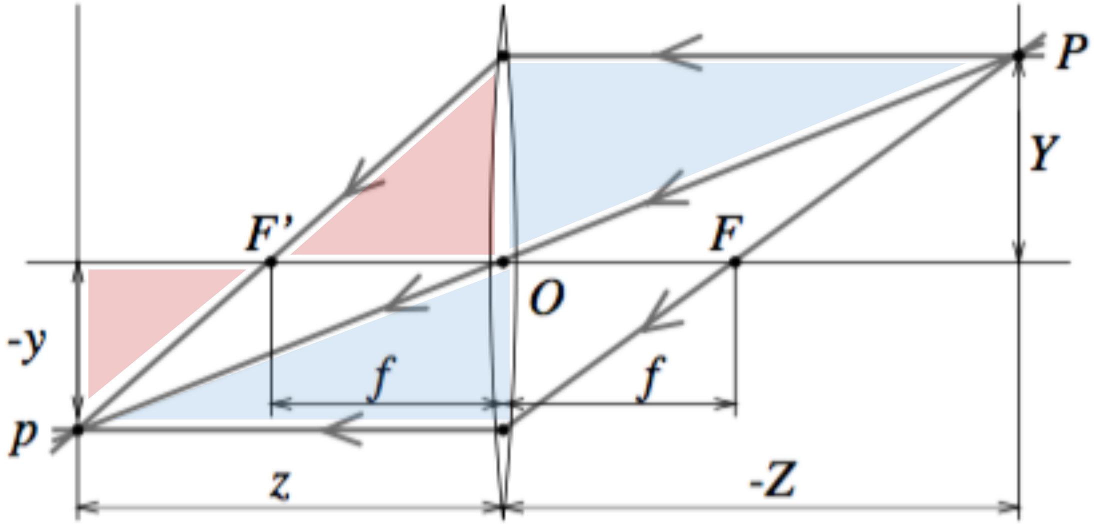


Lenses

- Lens: an optical element that focuses light by means of refraction



Thin lens model



Credit: FP Chapter 1

Key properties (follows from Snell's law) :

1. Rays passing through *O* are not refracted
2. Rays parallel to the optical axis are focused on the *focal point F'*
3. *All* rays passing through *P* are focused by the thin lens on the point *p*

- Similar triangles

$$\frac{y}{Y} = \frac{z}{Z}$$ Blue triangles

$$\frac{y}{Y} = \frac{z-f}{f} = \frac{z}{f} - 1$$ Red triangles

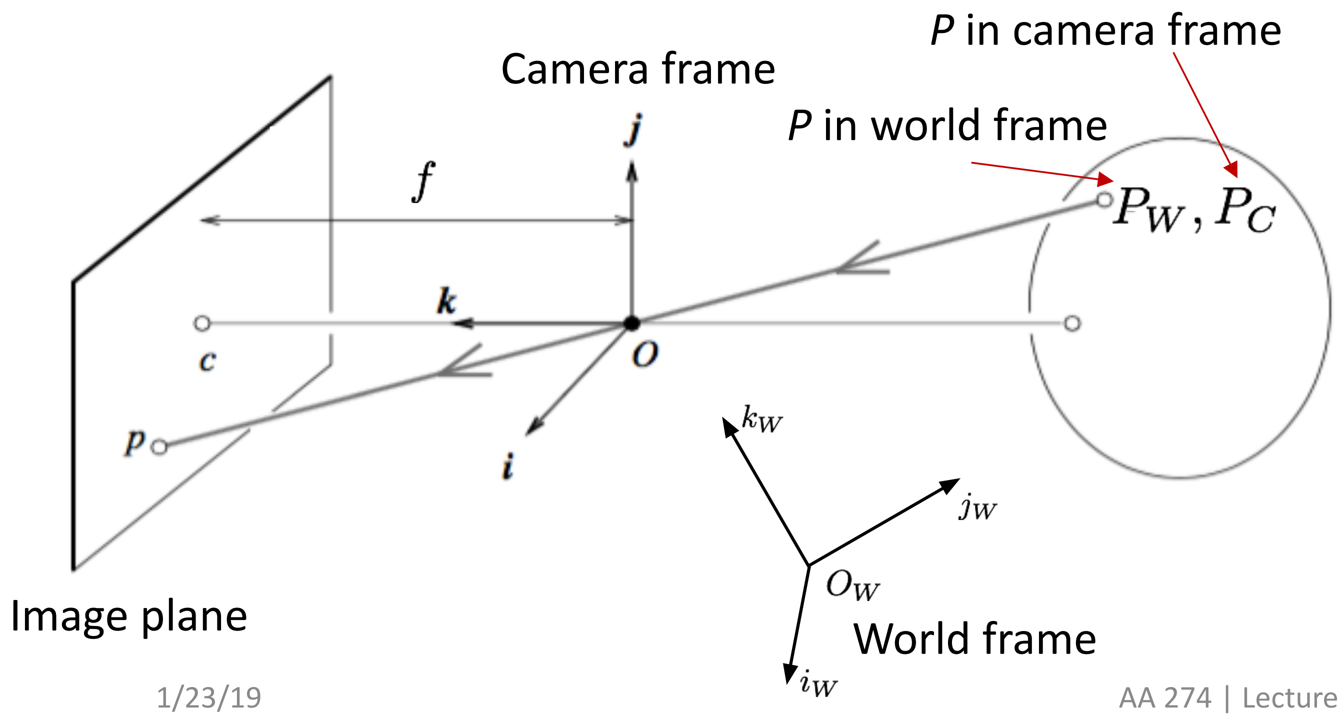
$$\boxed{\Rightarrow \frac{1}{z} + \frac{1}{Z} = \frac{1}{f}}$$ Thin lens equation

Thin lens model

- Key points:
  1. The equations relating the positions of *P* and *p* are exactly the same as under pinhole perspective if one considers *z* as focal length (as opposed to f), since *P* and *p* lie on a ray passing through the center of the lens
  2. Points located at a distance $-Z$ from *O* will be in sharp focus only when the image plane is located at a distance *z* from *O* on the other side of the lens that satisfies the thin lens equation
  3. In practice, objects within some range of distances (called depth of field or depth of focus) will be in acceptable focus
  4. Letting $Z \rightarrow \infty$ shows that *f* is the distance between the center of the lens and the plane where distant objects focus
  5. In reality, lenses suffer from a number of *aberrations*

Perspective projection

- Goal: find how world points map in the camera image
- Assumption: pinhole camera model (*all results also hold under thin lens model, assuming camera is focused at ∞*)



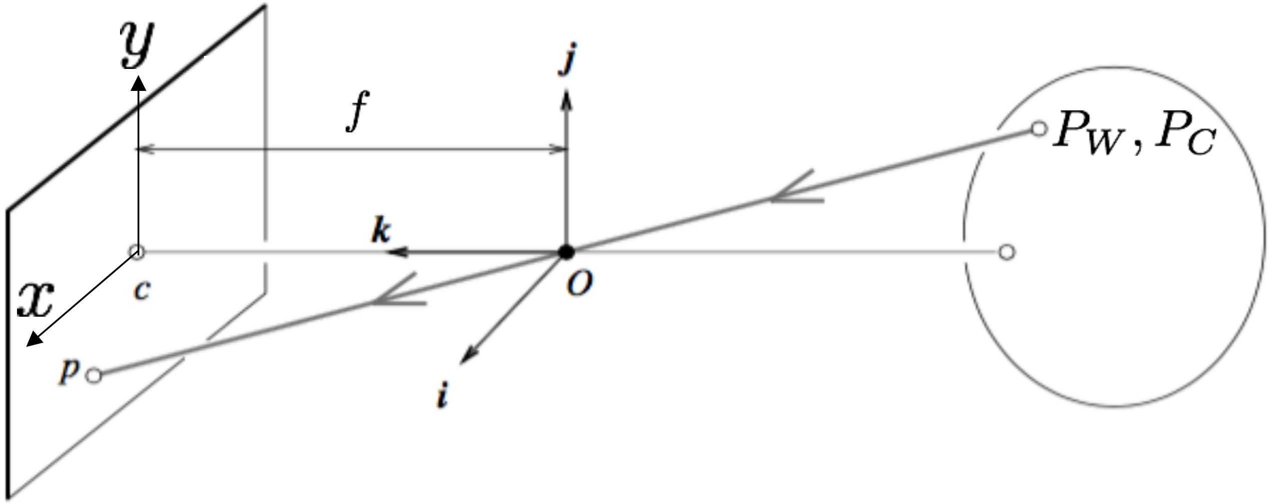
Procedure

1. Map P_c into p (image plane)
2. Map p into (u,v) (pixel coordinates)
3. Transform P_w into P_c

Step 1

- Task: Map $P_c = (X_C, Y_C, Z_C)$ into $p = (x, y)$ (image plane)
- From before

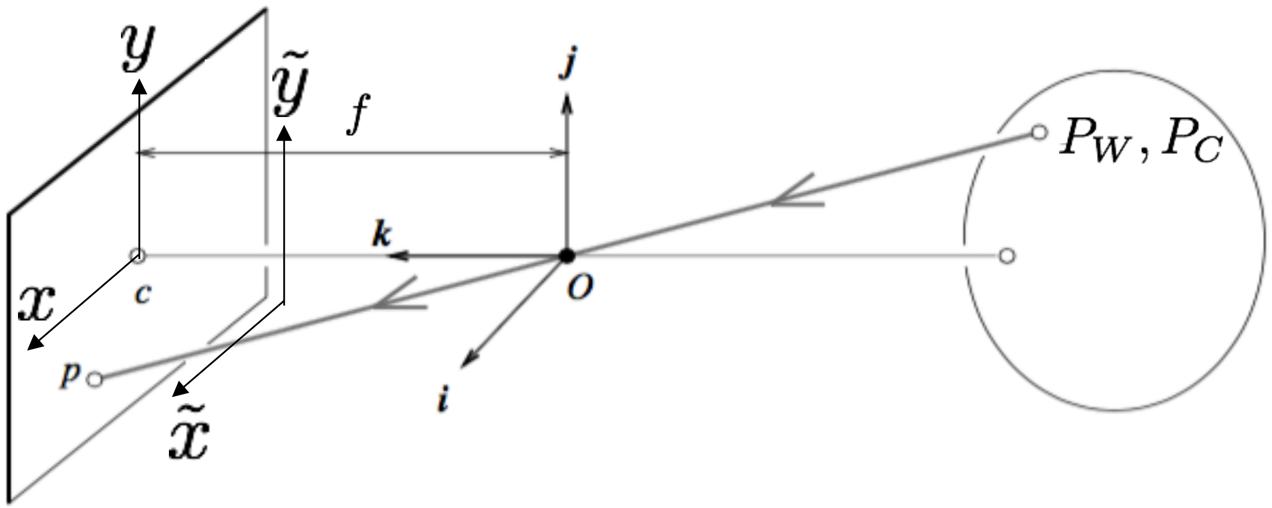
$$\begin{cases} x = f\,\frac{X_C}{Z_C} \\ y = f\,\frac{Y_C}{Z_C} \end{cases}$$

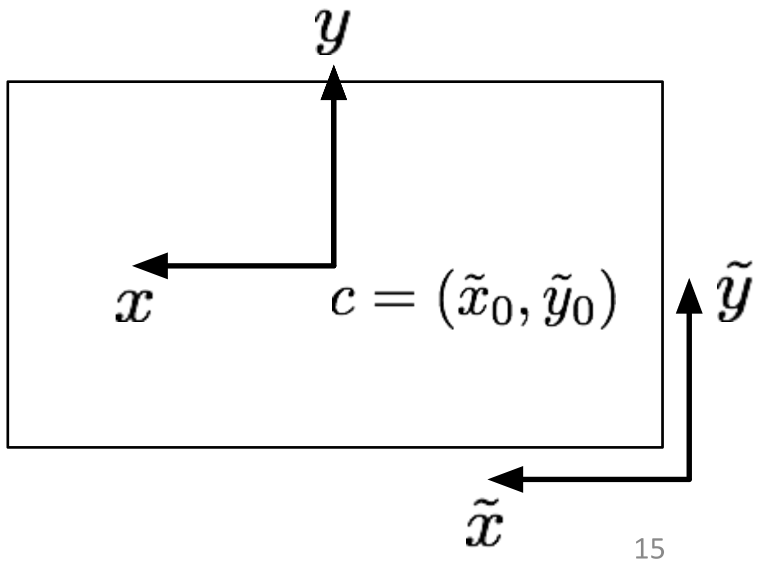


Step 2.a

- Fact: actual origin of the camera coordinate system is usually at a corner (lower left)

$$\tilde{x} = f\,\frac{X_C}{Z_C} + \tilde{x}_0, \qquad \tilde{y} = f\,\frac{Y_C}{Z_C} + \tilde{y}_0,$$





Step 2.b

- Task: convert from image coordinates $(\tilde{x}, \tilde{y})$ to pixel coordinates (u, v)
- Let k_x and k_y be the number of pixels per unit distance in image coordinates in the *x* and *y* directions, respectively

$$u = k_x \tilde{x} = \overbrace{k_x\, f}^{\alpha}\, \frac{X_C}{Z_C} + \overbrace{k_x \tilde{x}_0}^{u_0}$$

$$v = k_y \tilde{y} = \underbrace{k_y\, f}_{\beta}\, \frac{Y_C}{Z_C} + \underbrace{k_y \tilde{y}_0}_{v_0}$$

$$\Rightarrow$$

$$\boxed{\begin{aligned} u &= \alpha \frac{X_C}{Z_C} + u_0 \\ v &= \beta \frac{Y_C}{Z_C} + v_0 \end{aligned}}$$

Nonlinear transformation

Homogeneous coordinates

- Goal: represent the transformation as a linear mapping
- Key idea: introduce homogeneous coordinates

Inhomogenous -> homogeneous

$$\begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Homogenous -> inhomogeneous

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \Rightarrow \begin{pmatrix} x/w \\ y/w \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \Rightarrow \begin{pmatrix} x/w \\ y/w \\ z/w \end{pmatrix}$$

Perspective projection in homogeneous coordinates

- Projection can be equivalently written in homogeneous coordinates

$$\left[\begin{matrix}\overbrace{\boxed{\begin{matrix}\alpha & 0 & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{matrix}}}^{K} & \begin{matrix}0\\ 0\\ 0\end{matrix}\end{matrix}\right]\begin{pmatrix}X_c\\ Y_c\\ Z_c\\ 1\end{pmatrix} = \begin{pmatrix}\alpha X_c + u_0 Z_c\\ \beta Y_c + v_0 Z_c\\ Z_c\end{pmatrix}$$

Camera matrix/
Matrix of intrinsic parameters

P_c in homogeneous coordinates

Homogeneous pixel coordinates

- In homogeneous coordinates, the mapping is linear:

$$p^h = [K \quad 0_{3\times 1}]P_C^h$$

Point p in homogeneous pixel coordinates

Point P_c in homogeneous pixel coordinates

Skewness

- In some (rare) cases

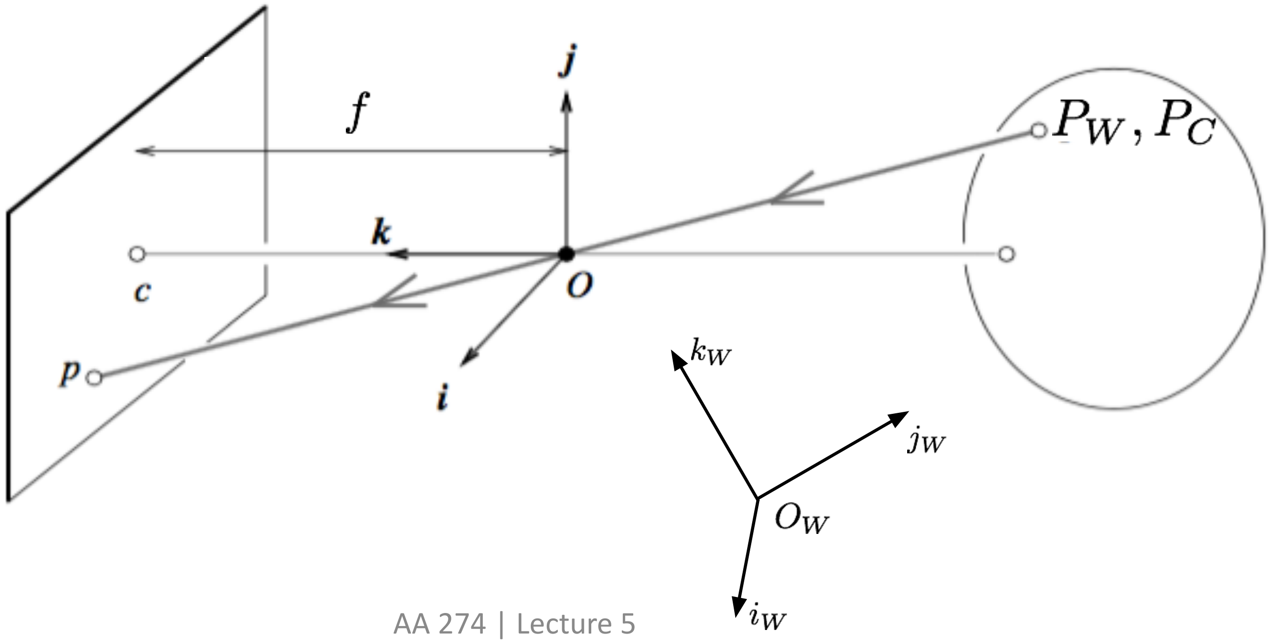
Skew parameter

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

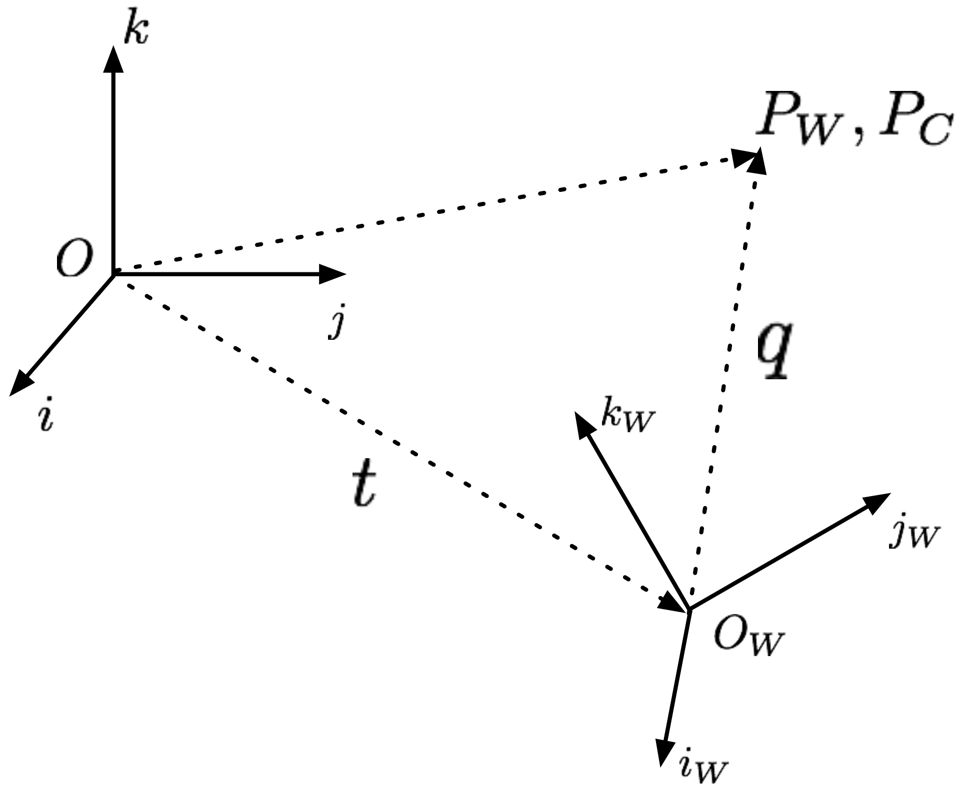
- When is $\gamma \neq 0$?
  - x- and y-axis of the camera are not perpendicular (unlikely)
  - For example, as a result of taking an image of an image
- Five parameters in total!

Step 3

- We have derived a mapping between a point *P* in the 3D camera reference frame to a point *p* in the 2D image plane
- Last step is to include in our mapping an additional transformation to account for the difference between the world frame and the 3D camera reference frame



Rigid transformations



$$P_C = t + q$$

$$q = R\,P_W$$

where *R* is the rotation matrix relating camera and world frames

$$R = \begin{bmatrix} i_W \cdot i & j_W \cdot i & k_W \cdot i \\ i_W \cdot j & j_W \cdot j & k_W \cdot j \\ i_W \cdot k & j_W \cdot k & k_W \cdot k \end{bmatrix}$$

$$\boxed{\Rightarrow P_C = t + R\,P_W}$$

Rigid transformations in homogeneous coordinates

$$\begin{pmatrix} P_C \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{pmatrix} P_W \\ 1 \end{pmatrix}$$

Point P_c in homogeneous coordinates

Point P_w in homogeneous coordinates

Perspective projection equation

- Collecting all results

$$p^h = [K \quad 0_{3\times 1}]P_C^h = K[I_{3\times 3} \quad 0_{3\times 1}]\begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} P_W^h$$

- Hence

Projection matrix *M*

$$\boxed{p^h = K\underbrace{[R \quad t]}_{M}P_W^h}$$

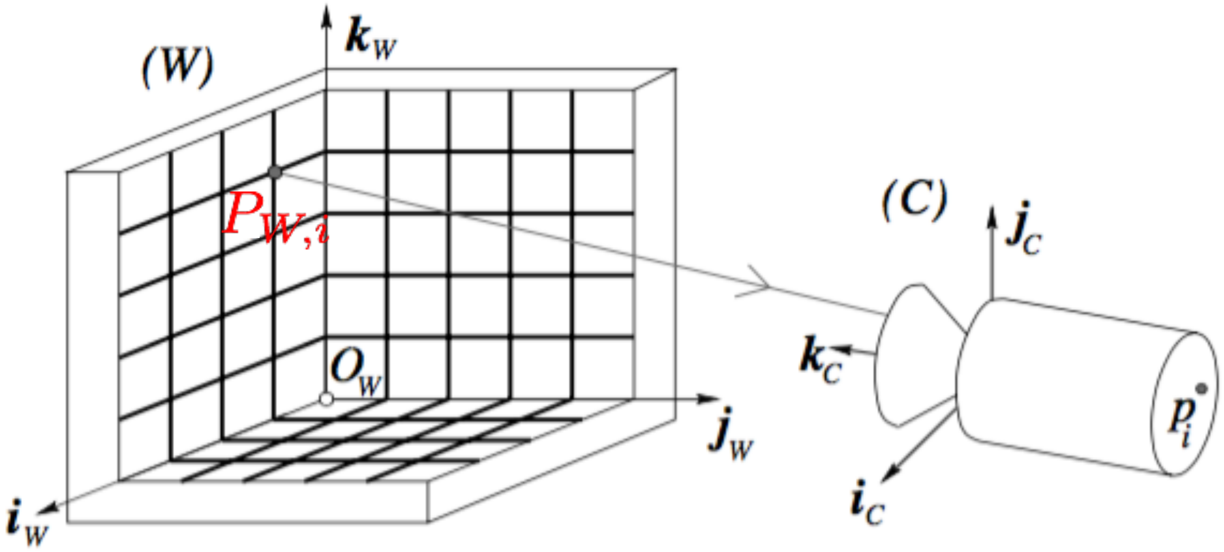
Intrinsic parameters

Extrinsic parameters

- Degrees of freedom: 4 for *K* (or 5 if we also include skewness), 3 for *R*, and 3 for *t*. Total is 10 (or 11 if we include skewness)

Camera calibration: direct linear transformation method

- Goal: find the intrinsic and extrinsic parameters of the camera



Strategy: given known correspondences $p_i \leftrightarrow P_{W,i}$, compute unknown parameters K, R, t by applying perspective projection

$P_{W,1}, P_{W,2}, \dots, P_{W,n}$ with known positions in world frame

$p_1, p_2, \dots, p_n$ with known positions in image frame

Step 1

- First consider combined parameters

1×4 vector

$$p_i^h = M\,P_{W,i}^h,\;\; i = 1,\ldots,n, \qquad \text{where} \quad M = K[R \quad t] = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

- This gives rise to $2n$ component-wise equations, for $i = 1, \ldots, n$

$$u_i = \frac{m_1 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h} \qquad\qquad u_i\,(m_3 \cdot P_{W,i}^h) - m_1 \cdot P_{W,i}^h = 0$$

or

$$v_i = \frac{m_2 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h} \qquad\qquad v_i\,(m_3 \cdot P_{W,i}^h) - m_2 \cdot P_{W,i}^h = 0$$

Calibration problem

- Stacking all equations together

$$\tilde{P}m = 0, \qquad \text{where } m = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix}_{12 \times 1}$$

2*n* x 12 matrix of known coefficients

12 x 1 vector of unknown coefficients

- $\tilde{P}$ contains in block form the known coefficients stemming from the given correspondences
- To estimate 11 coefficients, we need at least 6 correspondences

Solution

- To find non-zero solution

$$\min_{m \in R^{12}} \quad \|\tilde{P}m\|^2$$
$$\text{subject to} \quad \|m\|^2 = 1$$

- Solution: select eigenvector of $\tilde{P}^T\tilde{P}$ with the smallest eigenvalue
- Readily computed via SVD decomposition

Step 2

- Next, we need to extract the camera parameters, i.e., we want to factorize *M* as

$$M = [KR \quad Kt]$$

- This can be done efficiently (indeed, explicitly) by using RQ factorization, whereby the submatrix $\mathrm{M}_{11:33}$ is decomposed into the product of an upper triangular matrix *K* and a rotation matrix *R*

Radial distortion

- So far, we have assumed that a linear model is an accurate model of the imaging process
- For real (non-pinhole) lenses this assumption will not hold

No distortion

Barrel distortion

Pincushion distortion

Credit: SNS

Distortion correction

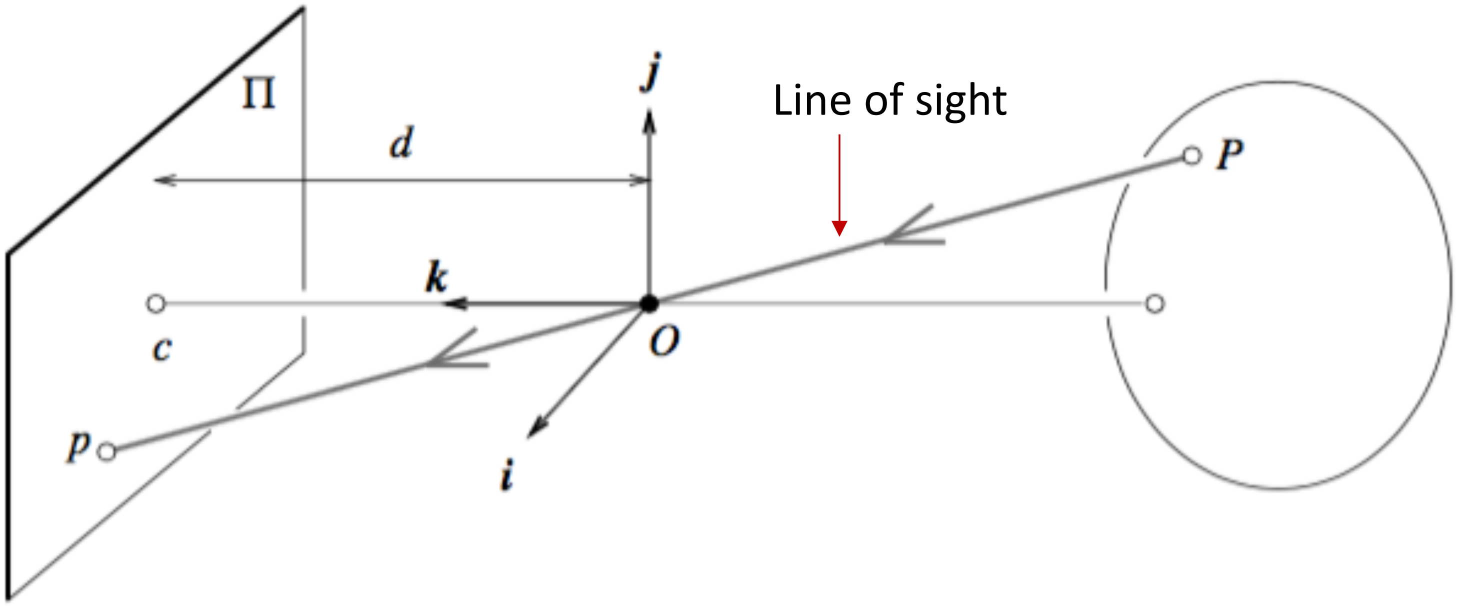
- Transformation from ideal (u, v) to distorted (u_d, v_d) pixel coordinates

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k\,r^2) \begin{bmatrix} u - u_{cd} \\ v - v_{cd} \end{bmatrix} + \begin{bmatrix} u_{cd} \\ v_{cd} \end{bmatrix}$$

where:

  - k: radial distortion parameter
  - $r^2 = (u - u_{cd})^2 + (v - v_{cd})^2$
  - (u_{cd}, v_{cd}) is the center of the distortion
- More sophisticated models are possible
- Calibration will be investigated further in Problem 1 in pset

Measuring depth



$$p^h = K[R \quad t]P_W^h$$

Homogeneous coordinates

Once the camera is calibrated, can we measure the location of a point *P* in 3D given its known observation *p*?

- No: one can only say that *P* is located *somewhere* along the line joining *p* and *O*!

Issues with recovering structure

Recovering structure

- Structure: 3D scene to be reconstructed by having access to 2D images

- Common methods
  1. Through recognition of landmarks (e.g., orthogonal walls)
  2. Depth from focus: determines distance to one point by taking multiple images with better and better focus
  3. Stereo vision: processes two distinct images taken at the *same time* and assumes that the relative pose between the two cameras is known
  4. Structure from motion: processes two images taken with the same or different cameras at *different times* and from different unknown positions

Next time: stereo vision and intro to image processing