

Principles of Robot Autonomy I

Camera models and camera calibration

Attendance Form

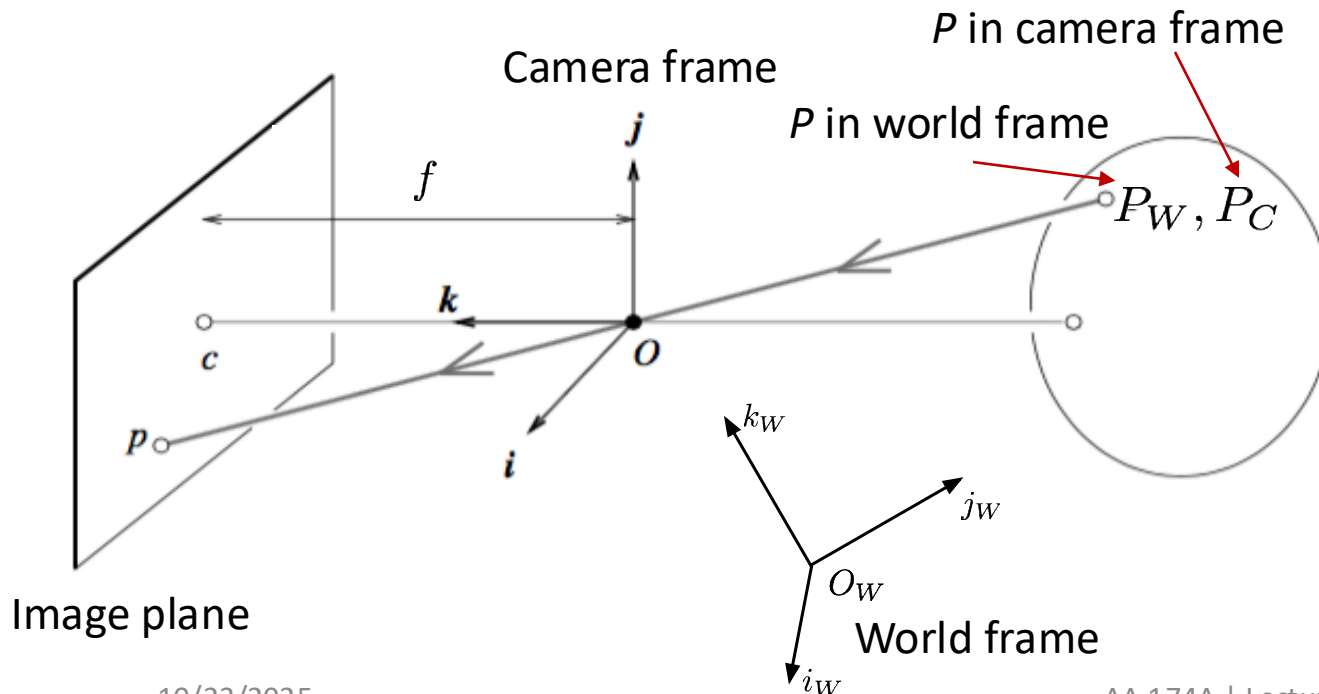


Agenda

- Agenda
 - Perspective projections
 - Camera calibration
 - Basic concepts in 3D reconstruction
- Readings:
 - Chapter 7, sections 7.1 – 7.4 in D. Gammelli, J. Lorenzetti, K. Luo, G. Zardini, M. Pavone. *Principles of Robot Autonomy*. 2026.

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 ∞*)



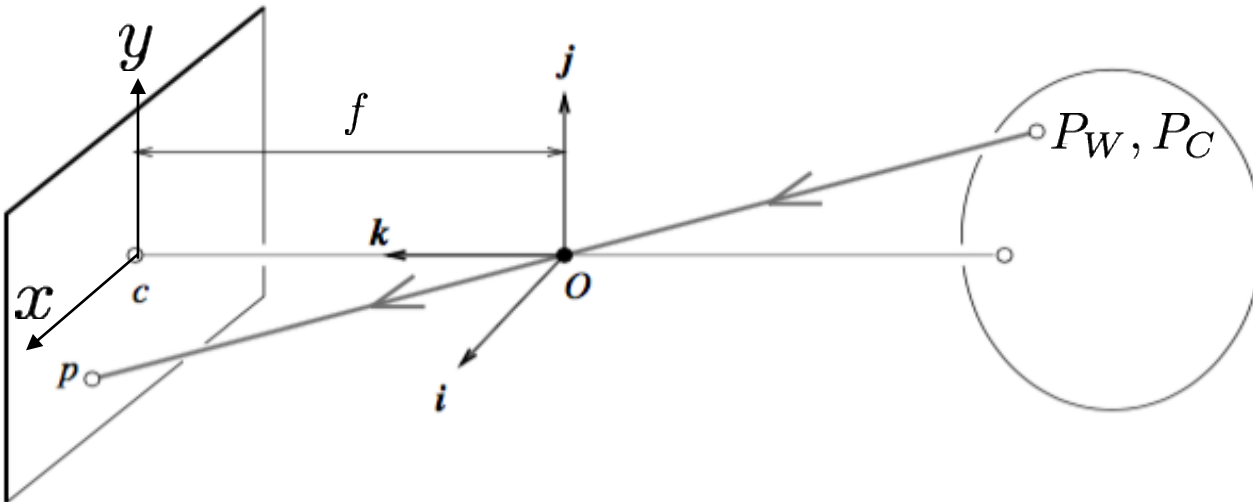
Roadmap:

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

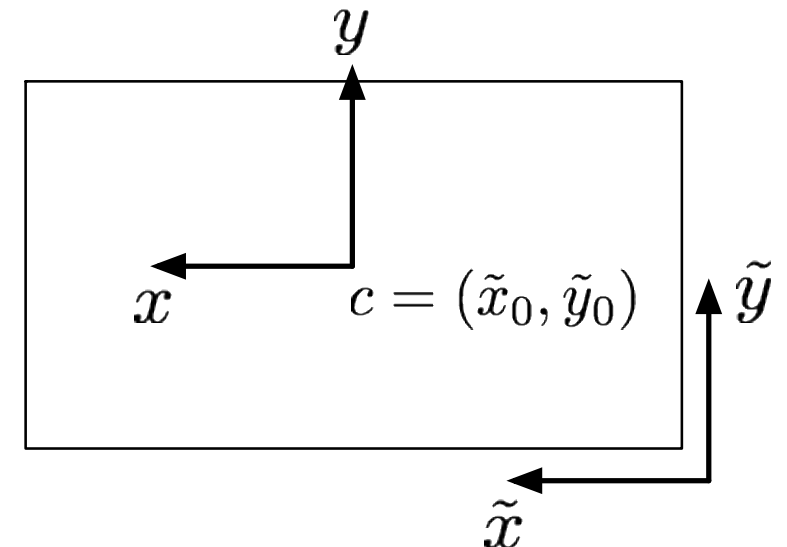
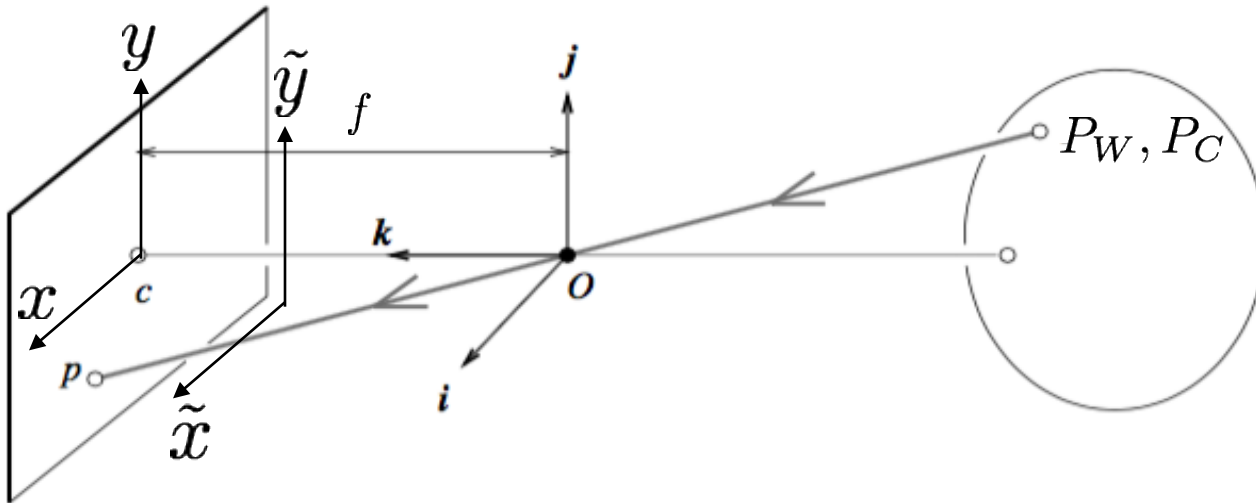
$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases}$$



Step 2.a

- Actual origin of the camera coordinate system is usually at a corner (e.g., top left, bottom left)

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



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

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

Nonlinear transformation

Homogeneous coordinates

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

Inhomogeneous \rightarrow homogeneous

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

Homogeneous \rightarrow inhomogeneous

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

Perspective projection in homogeneous coordinates

- Projection can be equivalently written in homogeneous coordinates

$$\overbrace{\begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}^K \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha X_c + u_0 Z_c \\ \beta Y_c + v_0 Z_c \\ Z_c \end{pmatrix}$$

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
camera coordinates

Skewness

- In some (rare) cases

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

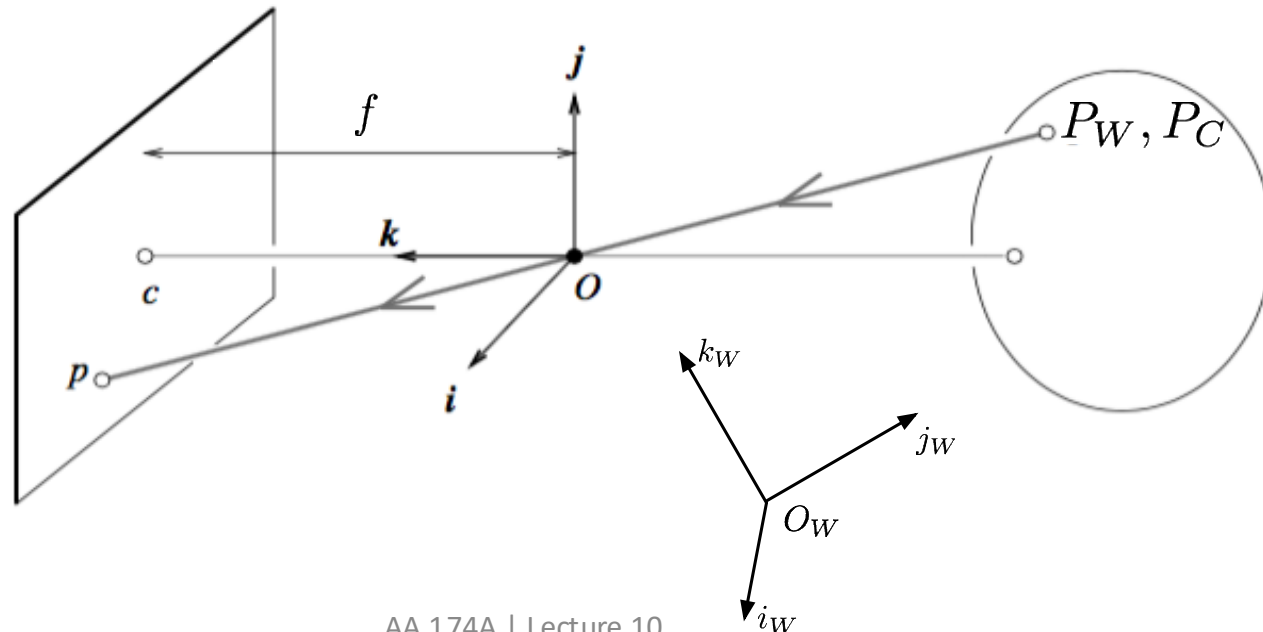
Skew parameter



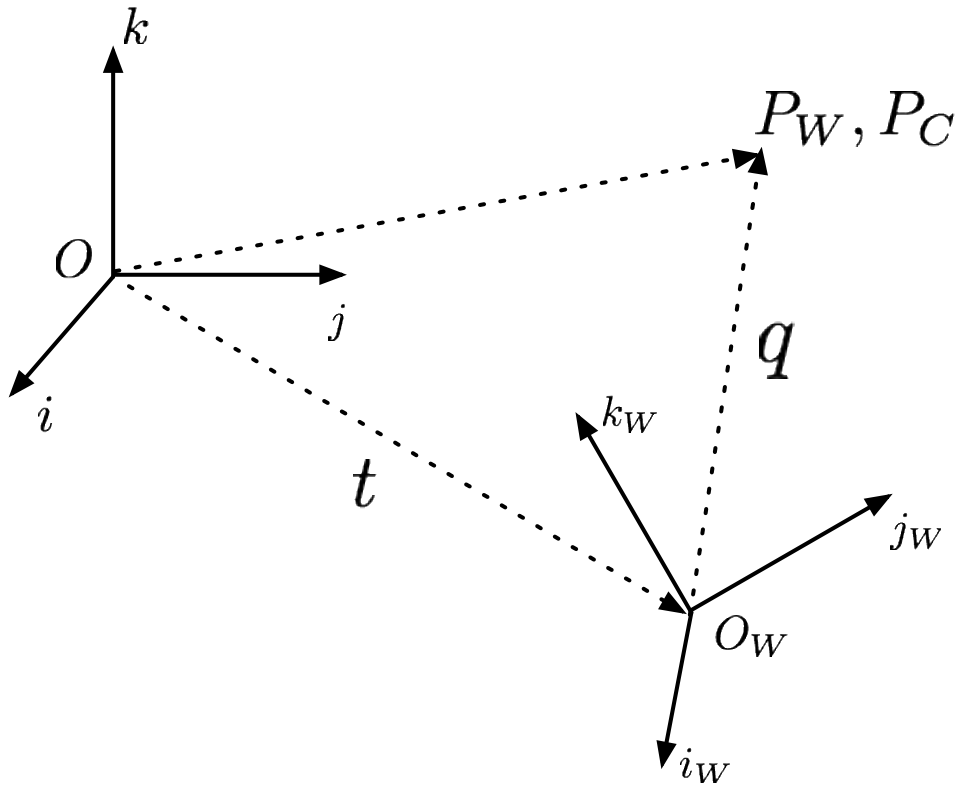
- When is $\gamma \neq 0$?
 - x- and y-axis of the camera are not perpendicular (unlikely)
 - For example, as a result of taking an image of an image
- Five parameters in total!

Step 3

- So far, 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

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

$$\Rightarrow P_C = t + R P_W$$

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

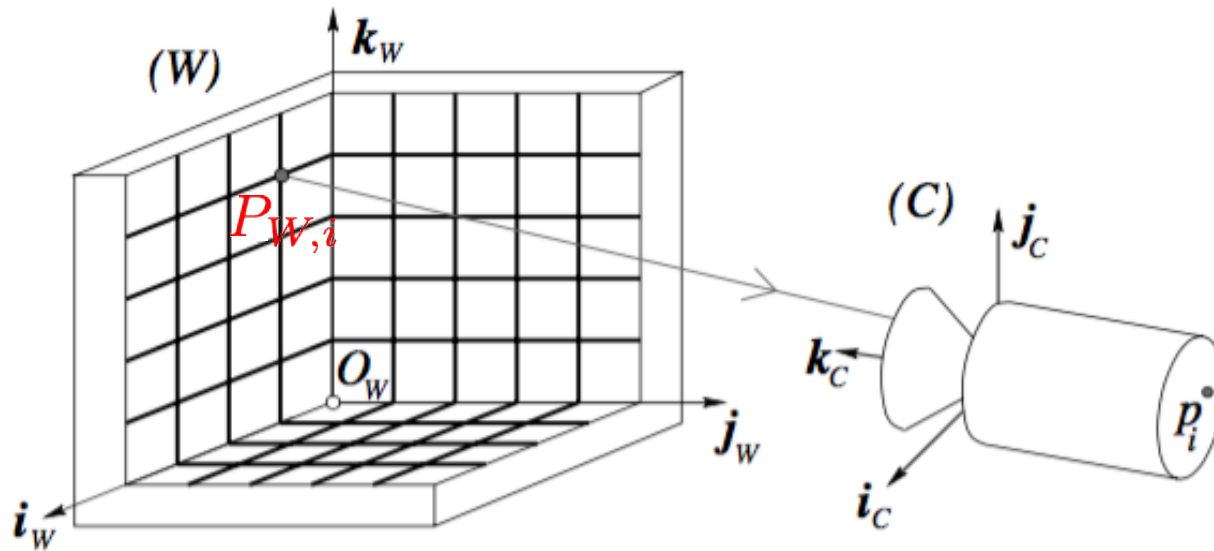
$$p^h = K [R \quad t] P_W^h$$

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



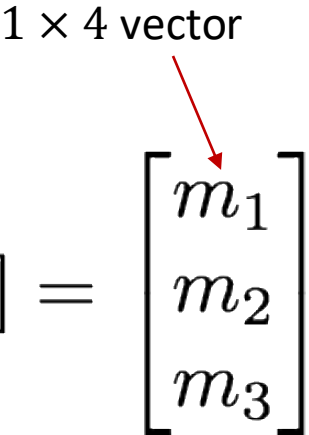
Strategy: given known correspondences $p_i \leftrightarrow P_{W,i}$, compute unknown parameters K, R, t by applying perspective projection

$P_{W,1}, P_{W,2}, \dots, P_{W,n}$ 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

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


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

$$\begin{aligned} u_i &= \frac{m_1 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h} & \text{or} & & u_i (m_3 \cdot P_{W,i}^h) - m_1 \cdot P_{W,i}^h &= 0 \\ v_i &= \frac{m_2 \cdot P_{W,i}^h}{m_3 \cdot P_{W,i}^h} & & & v_i (m_3 \cdot P_{W,i}^h) - m_2 \cdot P_{W,i}^h &= 0 \end{aligned}$$

Calibration problem

- Stacking all equations together

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

$2n \times 12$ matrix of known coefficients 12×1 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

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

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

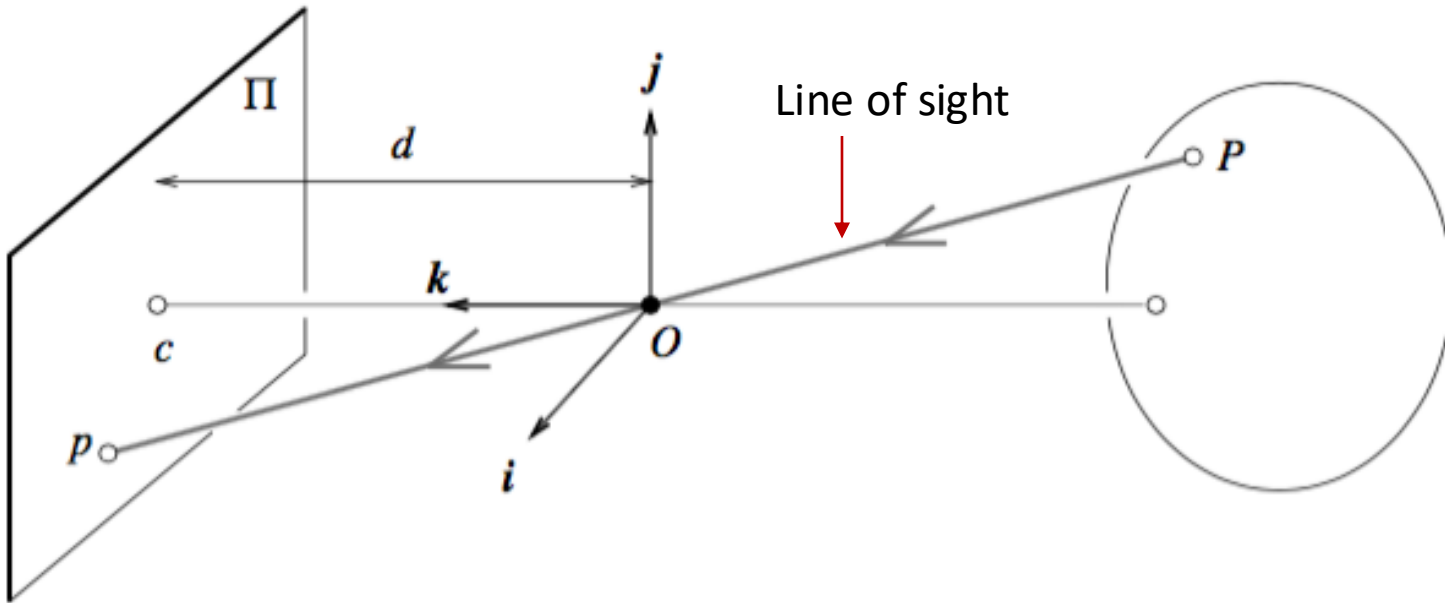
Step 2

- 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 $M_{1:3,1:3}$ is decomposed into the product of an upper triangular matrix K and a rotation matrix R
- These concepts will be investigated further in **Problem 1 in HW3**

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 !

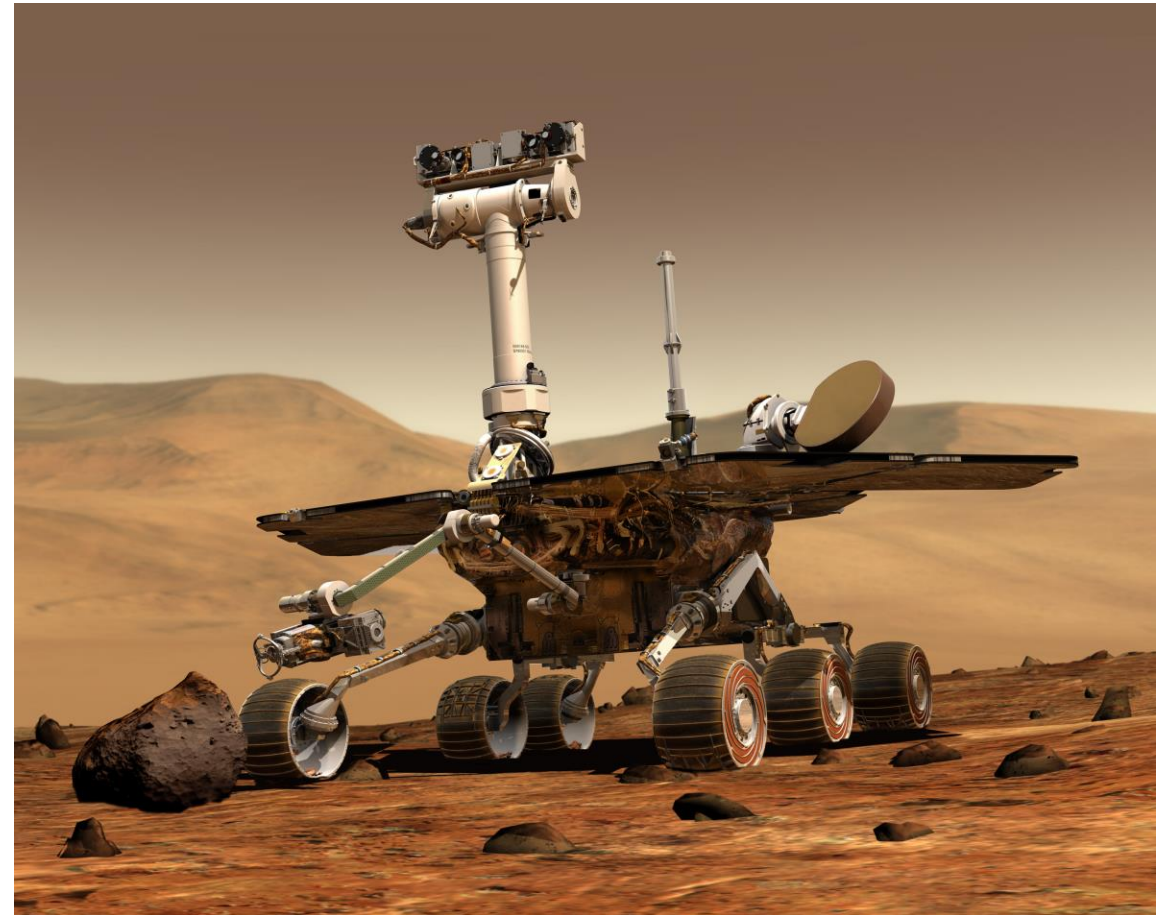
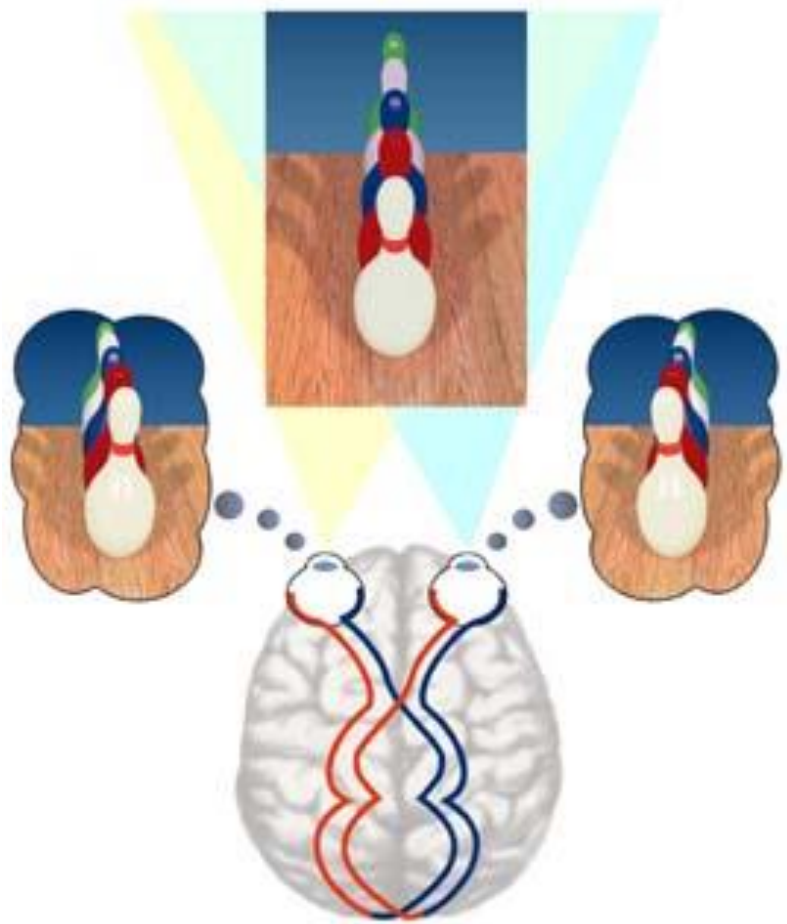
Issues with recovering structure



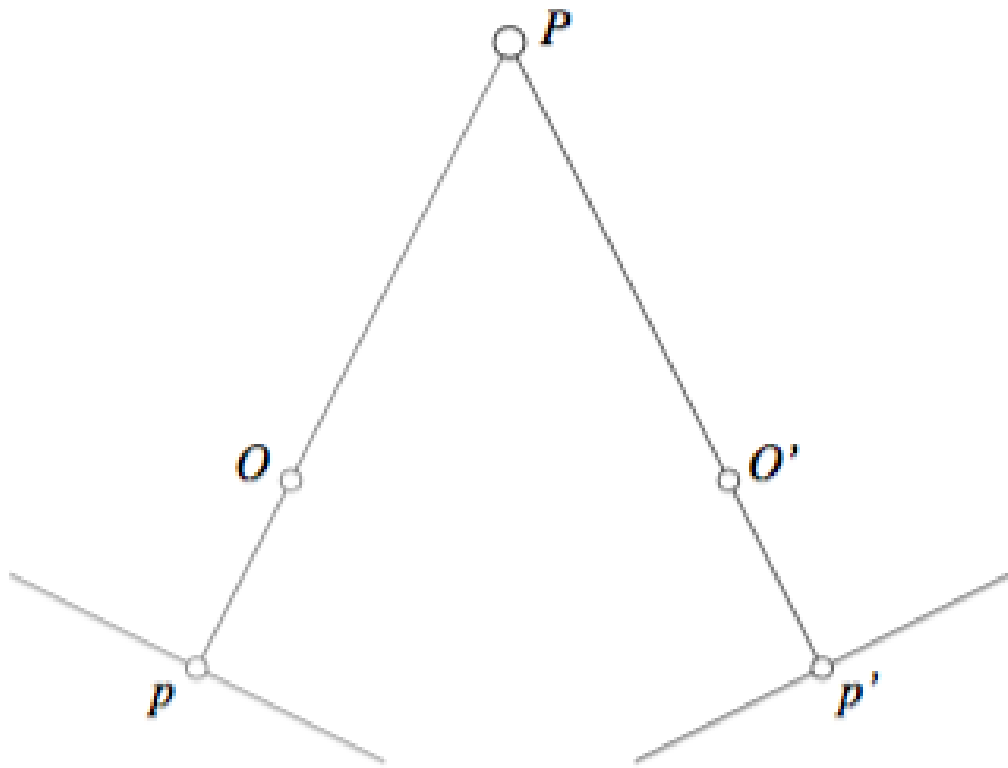
Recovering structure

- **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

Stereopsis, or why we have two eyes

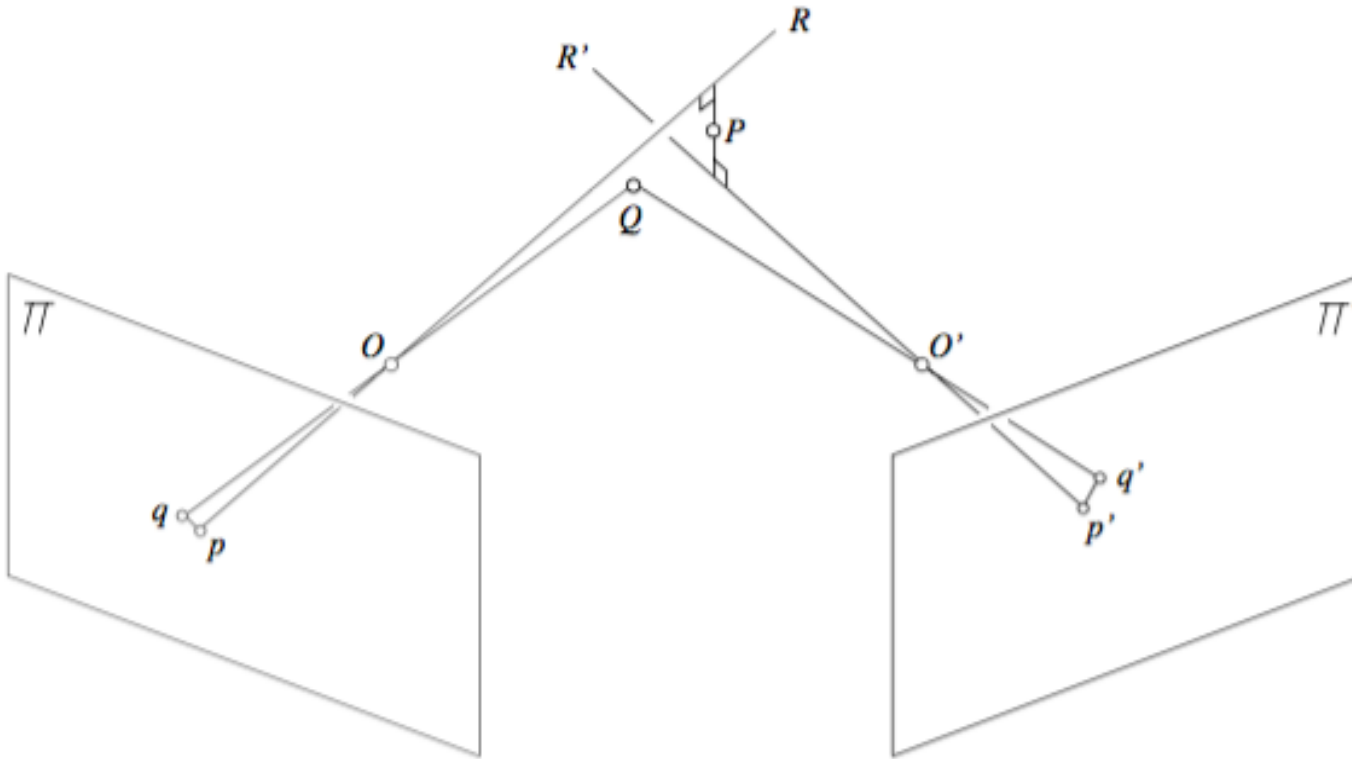


Binocular reconstruction



- **Given:** *calibrated* stereo rig and two image matching points p and p'
- **Find** corresponding scene point by intersecting the two rays \overline{Op} and $\overline{O'p'}$ (process known as **triangulation**)

Approximate triangulation



- Due to noise, triangulation problem is often solved as finding the point Q with images q and q' that minimizes

$$\underbrace{d^2(p, q) + d^2(p', q')}_{\text{Re-projection error}}$$

Next time: image processing,
feature detection & description

