Principles of Robot Autonomy I

Camera models and camera calibration

Stanford University

IPRL

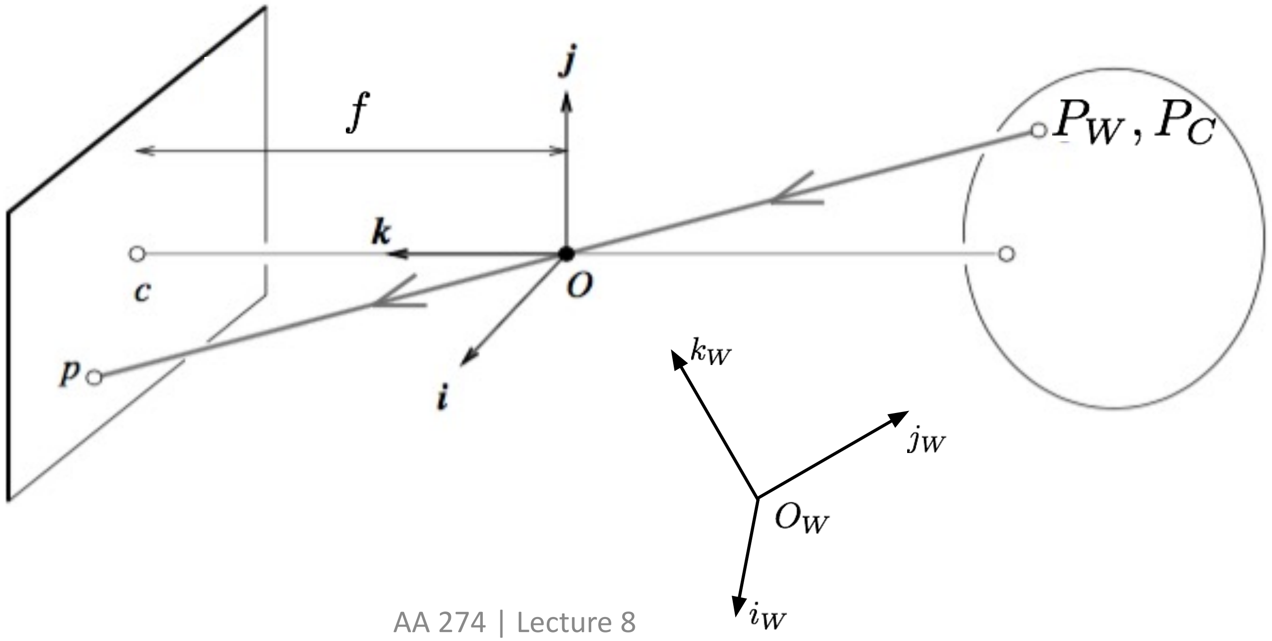
Autonomous Systems Lab

Camera models and camera calibration

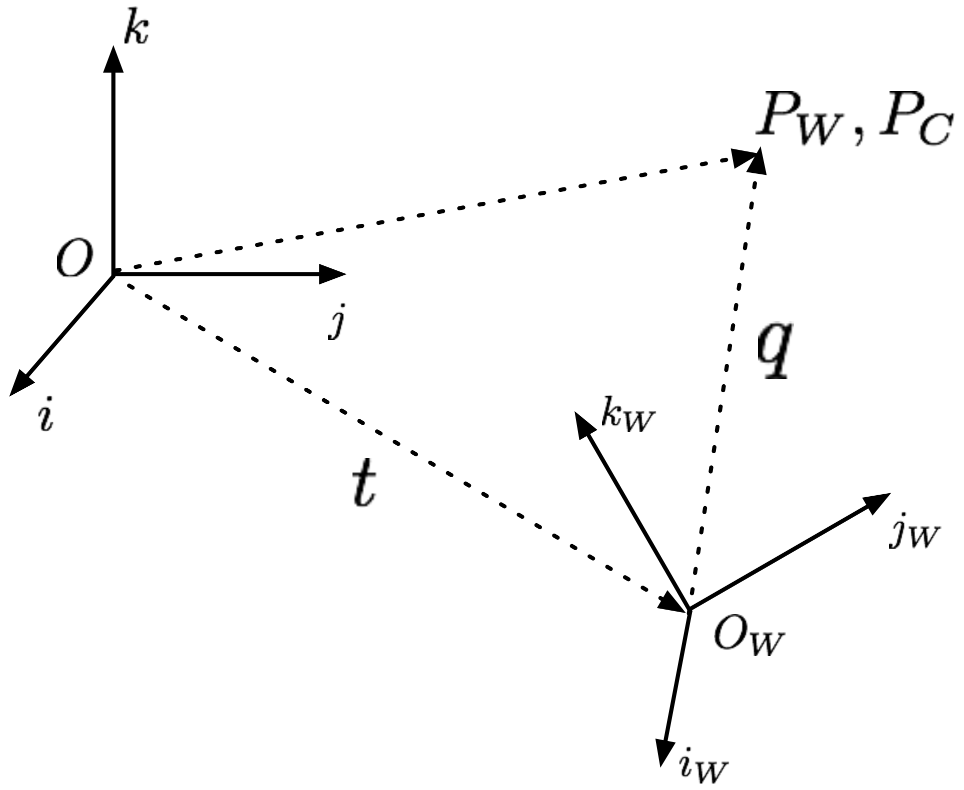
- Aim
  - Learn how to calibrate a camera
  - Learn about 3D reconstruction
- Readings
  - SNS: 4.2.3
  - D. A. Forsyth and J. Ponce [FP]. Computer Vision: A Modern Approach (2nd Edition). Prentice Hall, 2011. Chapter 1.
  - R. Hartley and A. Zisserman [HZ]. Multiple View Geometry in Computer Vision. Academic Press, 2002. Chapter 6.1.
  - Z. Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

Step 3

- In previous lecture, 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

$$p^h = [K \quad 0_{3\times1}]P_C^h = K[I_{3\times3} \quad 0_{3\times1}]\begin{bmatrix} R & t \\ 0_{1\times3} & 1 \end{bmatrix} P_W^h$$

- Hence

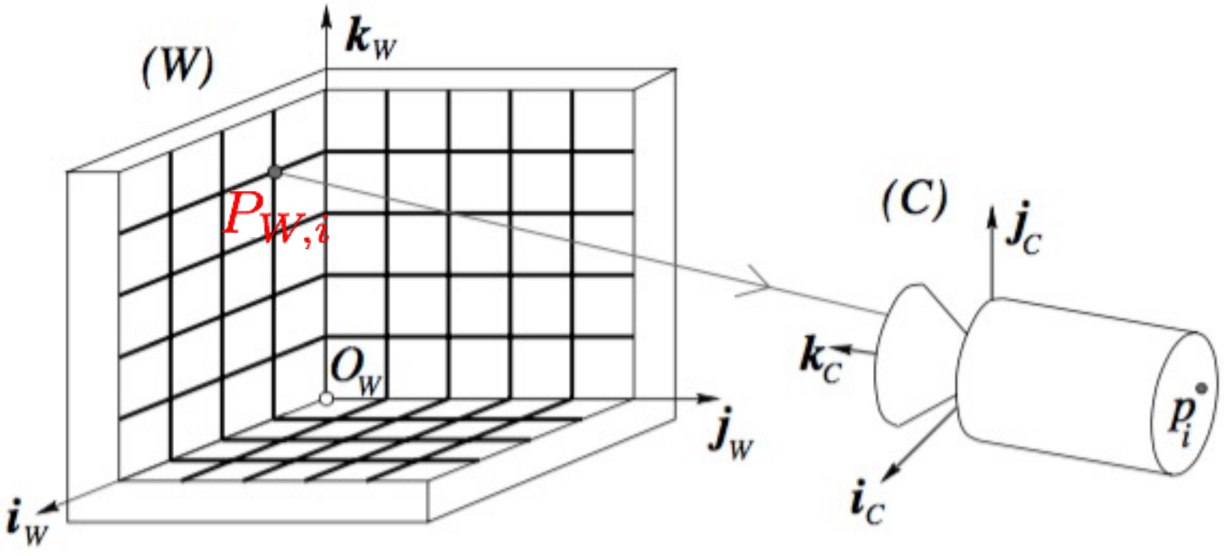
$$\boxed{p^h = \underbrace{K[R \quad t]}_{\text{Projection matrix } M} P_W^h}$$

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

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

1×4 vector

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

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

or

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

Calibration problem

- Stacking all equations together

$$\underbrace{\tilde{P}}_{2n \times 12 \text{ matrix of known coefficients}} \underbrace{m}_{12 \times 1 \text{ vector of unknown coefficients}} = 0, \qquad \text{where } m = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix}_{12 \times 1}$$

- $\tilde{P}$ 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

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

- 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 $\mathrm{M}_{1:3,1:3}$ is decomposed into the product of an upper triangular matrix *K* and a rotation matrix *R*

- Calibration will be investigated in Problem 1 in HW3

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

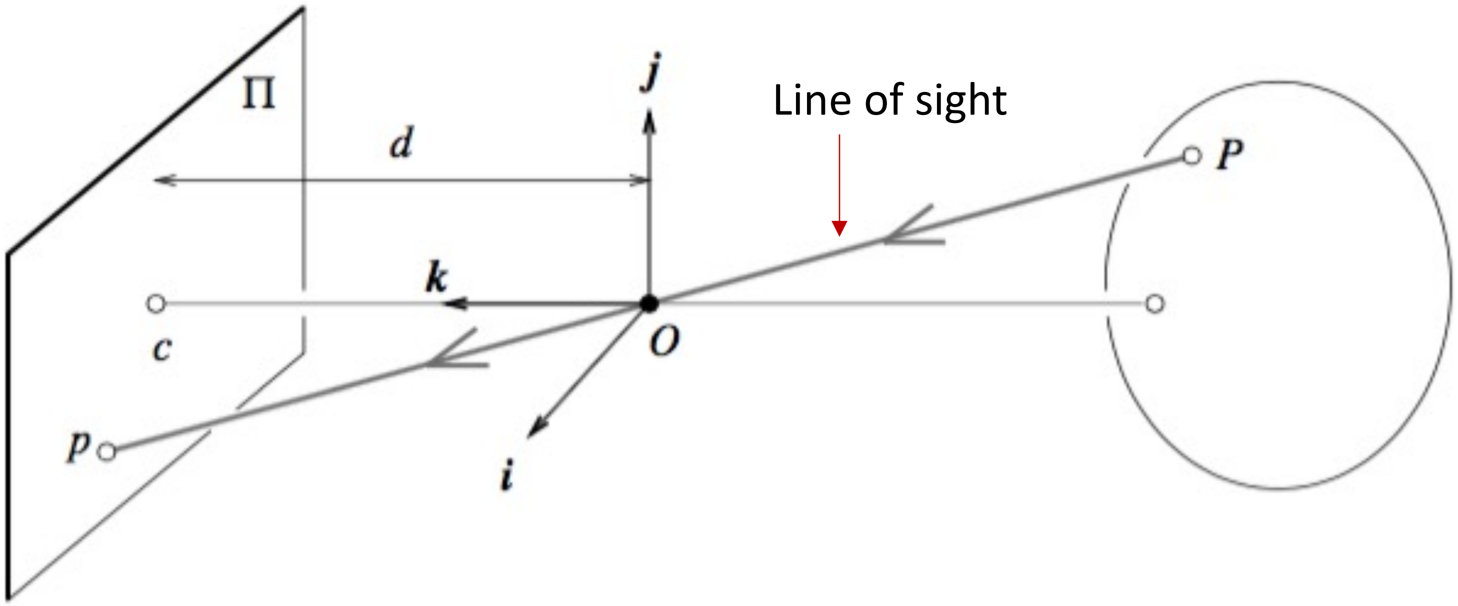
No distortion

Barrel distortion

Pincushion distortion

Credit: SNS

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*!

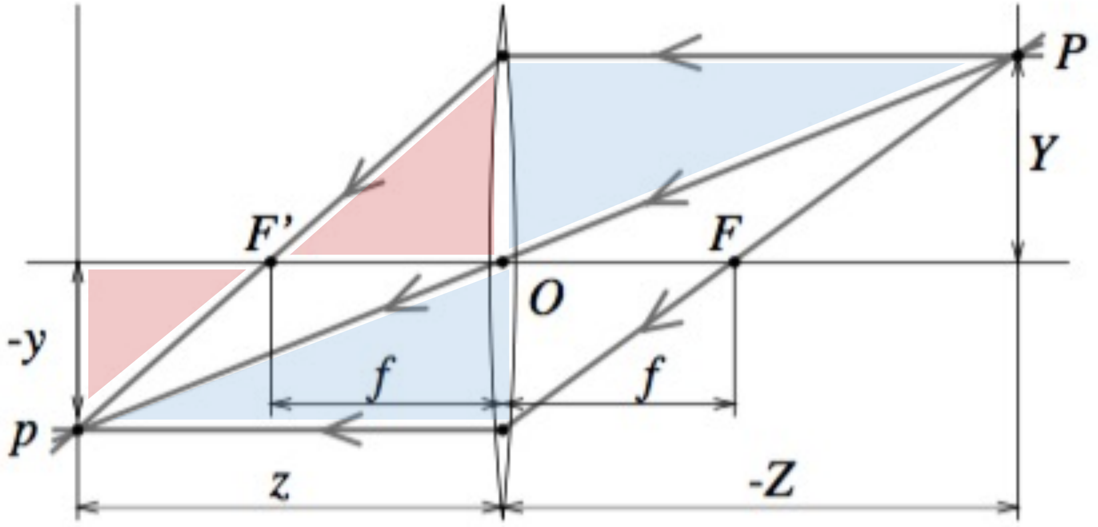
Issues with recovering structure

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

Depth from focus



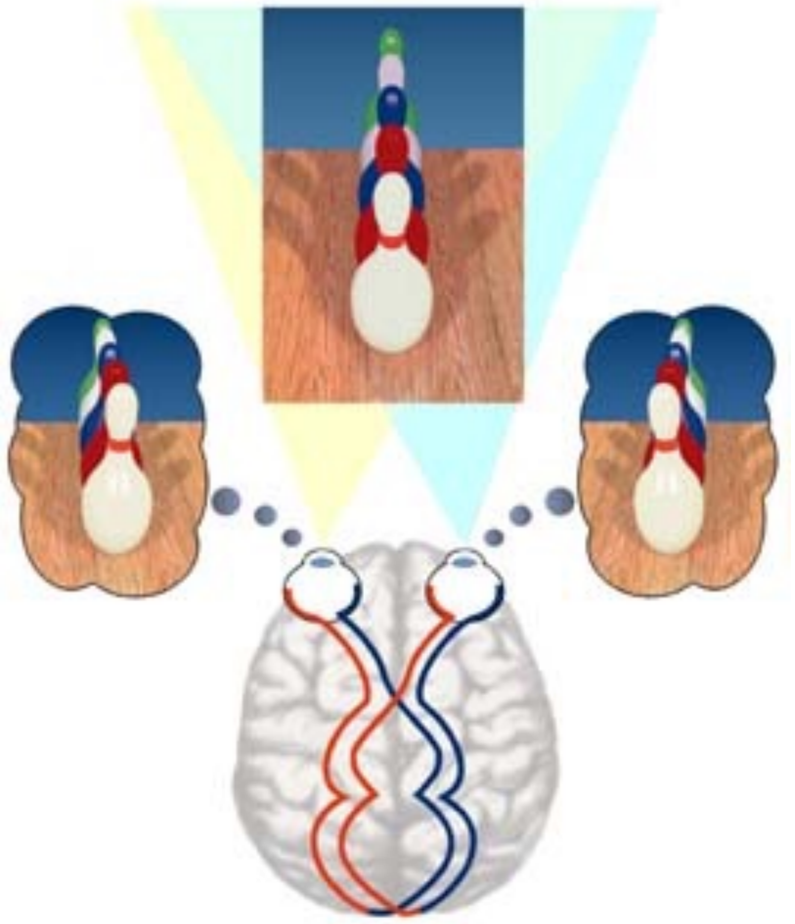
Credit: FP Chapter 1

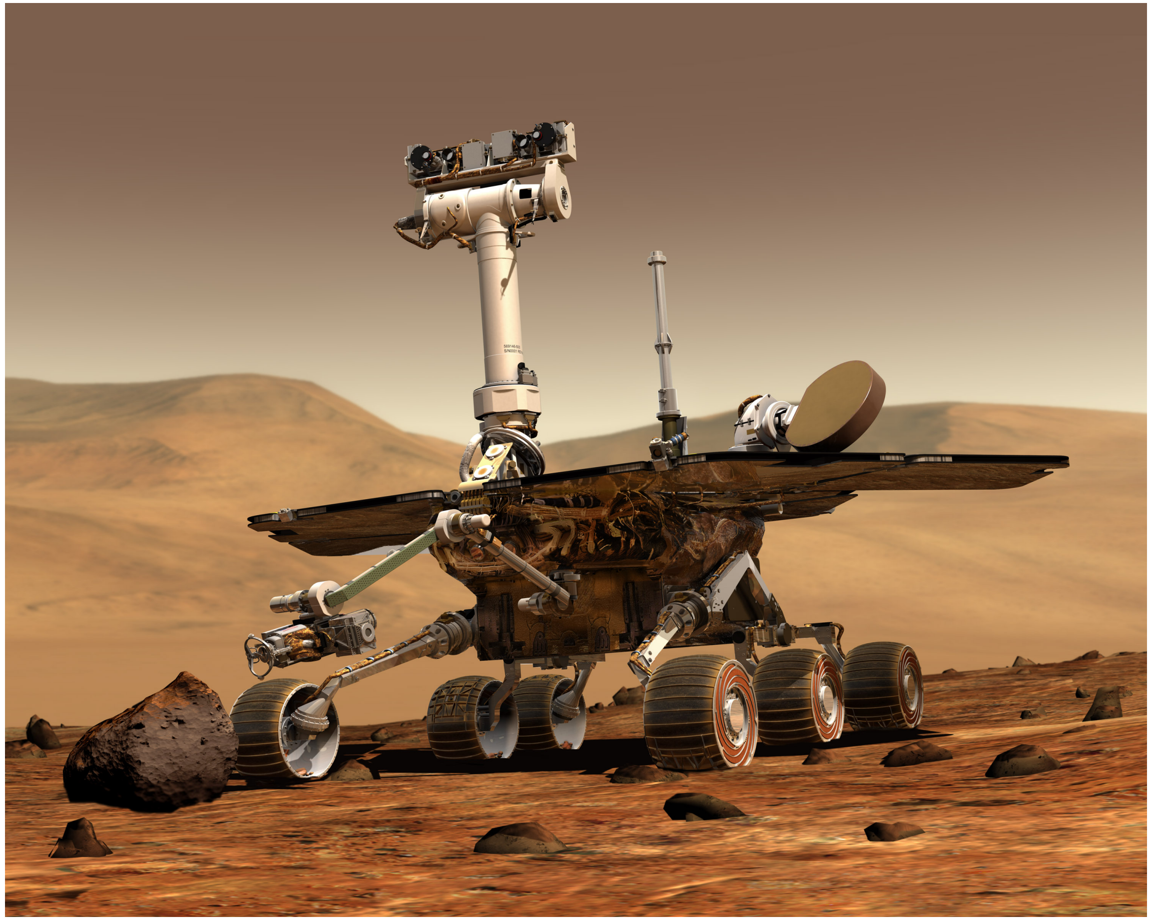
$$\Rightarrow \frac{1}{z} + \frac{1}{Z} = \frac{1}{f}$$

Thin lens equation

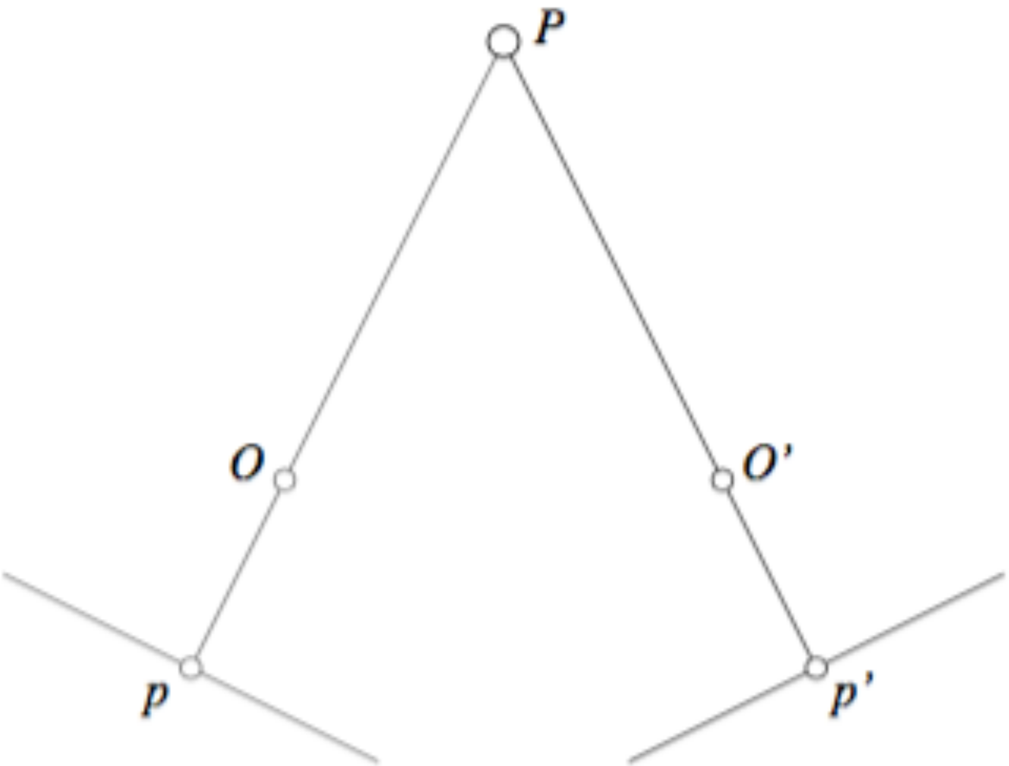
- Take several images until the projection of point *P* is in focus; let *z* denote the distance at which the image is in focus
- Since we know *z* and *f*, through the thin lens equation we obtain *Z*

Stereopsis, or why we have two eyes



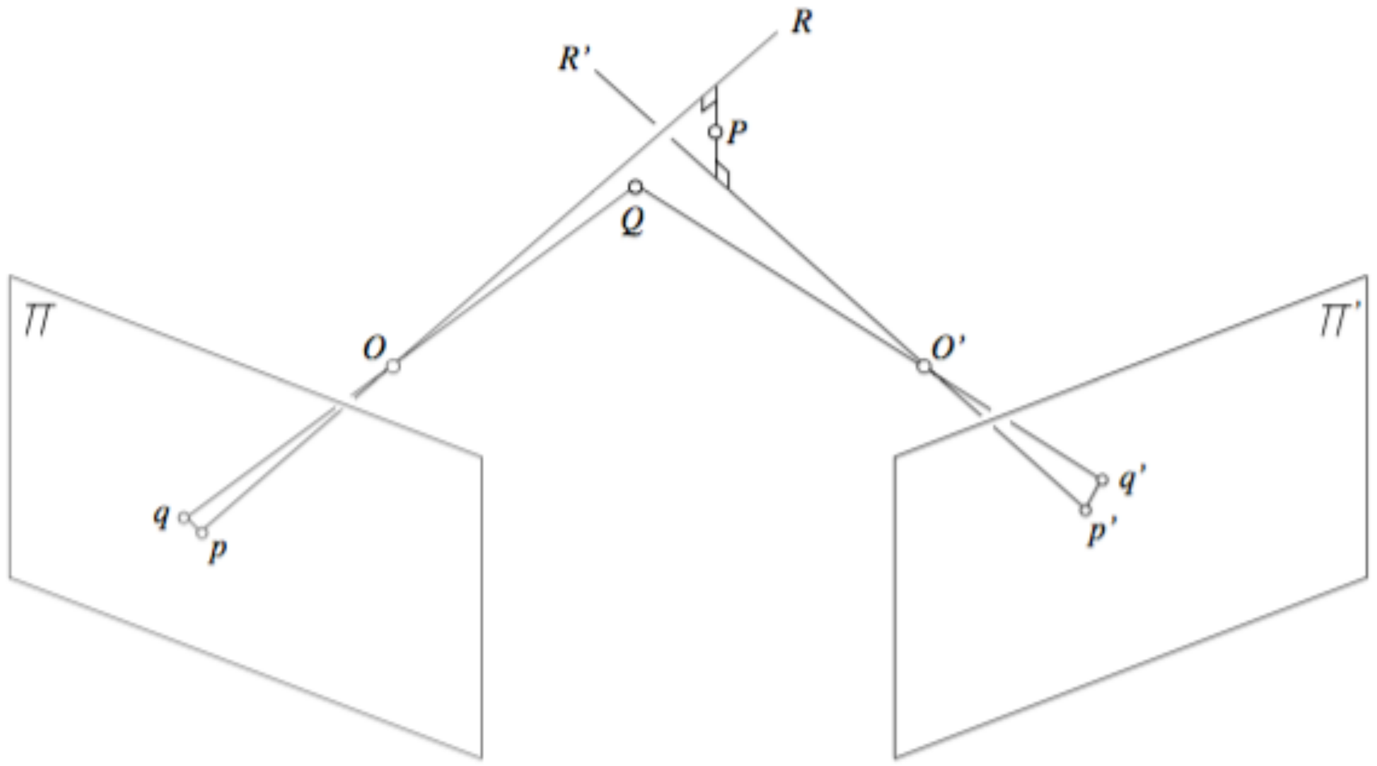


Binocular reconstruction



- 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