Computer Vision

5. Simultaneous Localization and Mapping (SLAM)

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

Mon. 10.4.	(1)	0. Introduction
		1. Projective Geometry in 2D: a. The Projective Plane
Mon. 17.4.	—	— *Easter Monday* —
Mon. 24.4.	(2)	1. Projective Geometry in 2D: b. Projective Transformations
Mon. 1.5.	—	— *Labor Day* —
Mon. 8.5.	(3)	2. Projective Geometry in 3D: a. Projective Space
Mon. 15.5.	(4)	2. Projective Geometry in 3D: b. Quadrics, Transformations
Mon. 22.5.	(5)	3. Estimating 2D Transformations: a. Direct Linear Transformation
Mon. 29.5.	(6)	3. Estimating 2D Transformations: b. Iterative Minimization
Mon. 5.6.	—	— *Pentecoste Day* —
Mon. 12.6.	(7)	4. Interest Points: a. Edges and Corners
Mon. 19.6.	(8)	4. Interest Points: b. Image Patches
Mon. 26.6.	(9)	5. Simulataneous Localization and Mapping: a. Camera Models
Mon. 3.7.	(10)	5. Simulataneous Localization and Mapping: b. Triangulation

Outline

Outline

Different Approaches to SLAM:

- Kalman filters
- Particle filters / Monte Carlo methods
- Scan matching of range data
- Set-membership techniques
- Bundle adjustment

Outline

1. Overview of SLAM

2. Camera Models

3. Two Cameras and the Fundamental Matrix

4. Triangulation

5. Putting it all Together

Types of Cameras

Camera: Mapping from 3D world to 2D image.

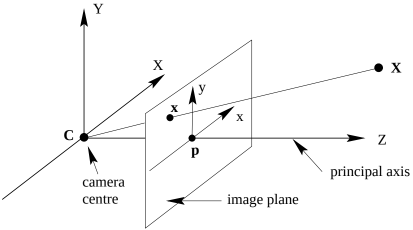
finite camera:

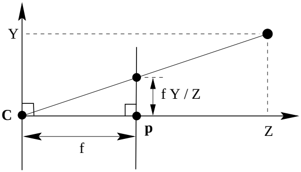
- finite camera center

infinite camera:

- camera center at infinity
- generalization of parallel projection

Pinhole Camera





$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} fx/z \\ fy/z \end{pmatrix}$$

[HZ04, p. 154]

Pinhole Camera / Homogeneous Coordinates

inhomogeneous coordinates:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} fx/z \\ fy/z \end{pmatrix}$$

homogeneous coordinates:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{pmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$P = \mathrm{diag}(f, f, 1)[I \mid 0]$$

Pinhole Camera / Principal Point Offset

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fx/z + p_x \\ fx/z + p_y \\ 1 \end{pmatrix} = \begin{pmatrix} fx + zp_x \\ fy + zp_y \\ z \end{pmatrix} = \begin{pmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$P = \underbrace{\begin{pmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{pmatrix}}_{=:K} [I \mid 0]$$

K is called **camera calibration matrix**.

Pinhole Camera / Camera Rotation and Translation

c': coordinates of camera center in world coordinates

R: rotation of world coordinate frame to camera coordinate frame (around c')

$$p = R(p' - c')$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} - \begin{pmatrix} x_{c'} \\ y_{c'} \\ z_{c'} \\ 1 \end{pmatrix} \right)$$

$$= \begin{pmatrix} R & -Rc' \\ & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

$$P = KR[I \mid -c']$$

without explicit camera center:

$$P = K[R \mid t], \quad t := -Rc'$$

CCD Cameras

CCD camera:

- pixels may be not square – different width α_x and height α_y

$$K = \begin{pmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix}$$

- finite projective camera:

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix}$$

  - s skew
  - usually $s = 0$, but rare cases (e.g., photo from photo)

Finite Projective Camera

- **skew** s:

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix}$$

$$P = K[R \mid t]$$

  - usually $s = 0$, but in rare cases (e.g., photo from photo)
- left 3×3 matrix is non-singular ($\det P_{1:3,1:3} \neq 0$)
- 11 parameters:
  - 5 for K: $\alpha_x, \alpha_y, x_0, y_0, s$
  - 3 for R
  - 3 for t
- any 3×4 matrix P with $\det P_{1:3,1:3} \neq 0$ is such a finite projective camera

Outline

Two Views: Epipolar Geometry

- two 2D views on a 3D scene
  - 3D coordinates X in the 3D scene
  - 2D coordinates x in the first view

$$x = PX$$

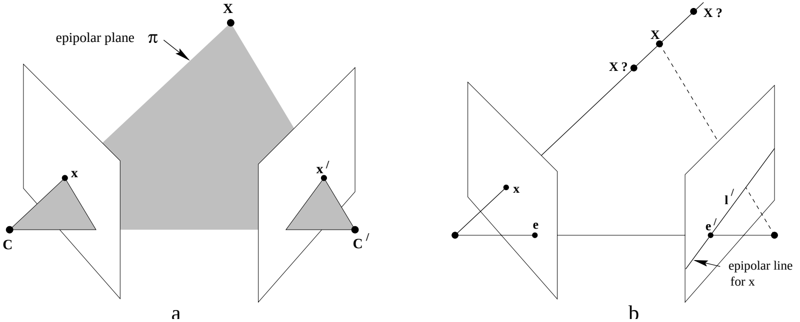
  - 2D coordinates x' in the second view

$$x' = P'X$$

- **epipolar geometry**: describe relation between the two views
- **fundamental matrix** F:

$$x'^T F x = 0 \iff \exists X : x = PX, x' = P'X$$

Epipolar Geometry



baseline: line joining the two camera centers

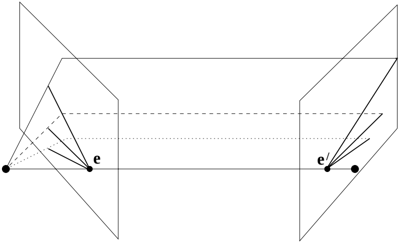
epipole: image of the camera center of the other view (intersection of baseline and image plane)

epipolar planes: planes containing the baseline

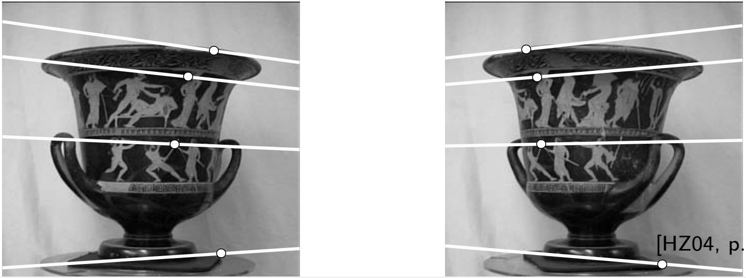
epipolar lines: lines in the image plane through the epipole

[HZ04, p. 240]

Epipolar Geometry / Example



a



[HZ04, p. 241]

Fundamental Matrix

- two views can be described by a map

$$F : x \mapsto \ell'$$

that maps
  - points x in the first view to
  - the epipolar line ℓ' of their possible correspondences in the second view.

Fundamental Matrix (2/2)

- construct ℓ:
  1. possible 3D source points of $x = PX$:

$$X = P^+ x + \lambda C, \quad \lambda \in \mathbb{R} \quad (\text{as } PC = 0)$$

  2. their 2D images in second view:

$$\begin{aligned} x' &= P'(P^+ x + \lambda C) = P' P^+ x + \lambda P' C \\ \text{esp.} \quad x' &:= P' P^+ x, \quad \text{for } \lambda := 0 \\ e' &= P' C, \quad \text{for } \lambda := \infty \text{ epipole of second view} \end{aligned}$$

  3. ℓ' is the line through x' and e':

$$F(x) = e' \times x' = e' \times P' P^+ x$$

- F is linear: **fundamental matrix** $F = [e']_\times P' P^+$

Note: P^+ pseudoinverse, C camera center 1st view, $[a]_\times := \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$.

From Two Cameras to the Fundamental Matrix

$$P = K[I \mid 0]$$
$$P' = K'[R \mid t]$$
$$\rightsquigarrow \quad P^+ = \begin{pmatrix} K^{-1} \\ 0^T \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

1. general case:

$$F = [P'C]_\times P'P^+ = [K't]_\times K'RK^{-1} = [e']_\times K'RK^{-1}$$

2. pure translation ($R = I, K' = K$):

$$F = [K't]_\times K'RK^{-1} = [Kt]_\times = [e']_\times$$

3. pure translation parallel to x-axis ($e' = (1,0,0)^T$):

$$F = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

From the Fundamental Matrix to Two Cameras

- The fundamental matrix does determine two cameras only up to a 3D projectivity.

$$
\begin{aligned}
\tilde{P} = PH, \quad \tilde{P}' &= P'H, \quad \tilde{C} = H^{-1}C \\
\rightsquigarrow \tilde{P}^+ &= H^{-1}P^+ \\
\tilde{F} &= [\tilde{P}'\tilde{C}]_\times \tilde{P}'\tilde{P}^+ \\
&= [P'HH^{-1}C]_\times P'HH^{-1}P^+ = [P'C]_\times P'P^+ = F
\end{aligned}
$$

- Cameras can be chosen as

$$
P = [I \mid 0], \quad P' = [[e']_\times F \mid e']
$$

$$
\rightsquigarrow F(P, P') = [e']_\times K'RK^{-1} = [e']_\times [e']_\times F \propto F
$$

Fundamental Matrix / Properties

- F maps points x of the 1st view to the epipolar line $\ell' := Fx$ of their possibly corresponding points in the 2nd view.
- For corresponding points x, x':

$$x'^T F x = 0$$

- e' is the left nullvector of F: $e'^T F = 0$ (as e' is on all lines Fx)
  e is the right nullvector of F: $Fe = 0$
- F has 7 degrees of freedom.
  - 8 ratios of a 3×3 matrix
  - -1 for $\det F = 0$

Computing the Fundamental Matrix

Different methods:

1. Linear Method I: The 8-Point Algorithm
2. Linear Method II: The 7-Point Algorithm
3. Iterative Minimization of the Reconstruction Error

Linear System of Equations

- every pair $((x, y), (x', y'))$ of corresponding points fullfills

$$(x', y')F(x, y)^T = 0$$

$$\rightsquigarrow \begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix} \text{vect}(F) = 0$$

- for N such pairs $((x_1, y_1), (x'_1, y'_1)), \ldots, ((x_N, y_N), (x'_N, y'_N))$:

$$\begin{pmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2 x_2 & x'_2 y_2 & x'_2 & y'_2 x_2 & y'_2 y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & & & & & & & & \\ x'_N x_N & x'_N y_N & x'_N & y'_N x_N & y'_N y_N & y'_N & x_N & y_N & 1 \end{pmatrix} \text{vect}(F) = 0$$

- linear system of equations: $Af = 0$ for $f = \text{vect}(F)$

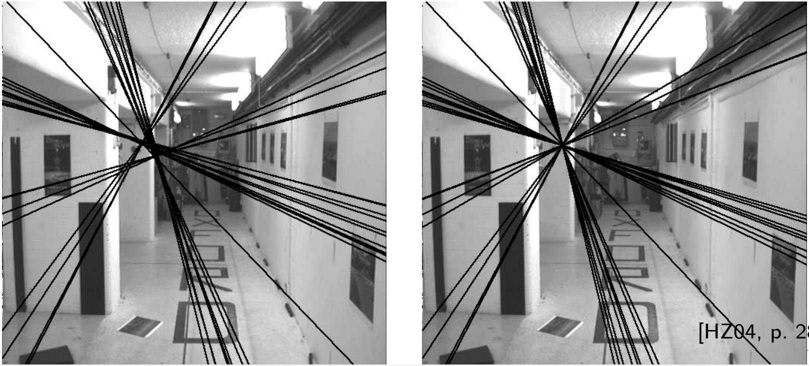
Note: $\text{vect}(A) := (a_{1,1}, a_{1,2}, \ldots, a_{1,M}, a_{2,1}, \ldots, a_{2,M}, \ldots, a_{N,1}, \ldots, a_{N,M})^T$ vectorization.

8-Point Algorithm

1. Solve linear system of equations for 8 corresponding points.
2. Ensure $\det F = 0$:

$$F = USU^T, \quad S = \operatorname{diag}(s_1, \ldots, s_9), s_1 \geq s_2 \geq \cdots \geq s_9 \text{ SVD}$$

$$F' := US'U^T, \quad S' := \operatorname{diag}(s_1, \ldots, s_8, 0)$$



[HZ04, p. 280]

7-Point Algorithm

1. Solve linear system of equations for 7 corresponding points, yielding $\lambda F_1 + (1-\lambda)F_2$
2. Ensure $\det F = 0$:

$$\det(\lambda F_1 + (1-\lambda)F_2) \stackrel{!}{=} 0$$

   Find root λ^* of this polynomial of degree 3, then

$$F := \lambda^* F_1 + (1-\lambda^*)F_2$$

- all linear methods should be used with normalization !
- both, esp. 7-point algorithm often used in RANSAC wrappers.

Iterative Minimization of the Reconstruction Error

$$\text{minimize} \sum_{n=1}^{N} d(x_n, \hat{x}_n)^2 + d(x'_n, \hat{x}'_n)^2$$

- $\hat{x}_n = PX_n = X_n$, for $P = [I \mid 0]$
- $\hat{x}'_n = P'X_n$, for general P'
- $3N + 12$ parameters (for general P')
- as in chapter 3:
  - initialize with linear method: 8-point algorithm
  - initial estimate of X_n by triangulation (see next section)
  - iteratively minimize using Levenberg-Marquardt

Outline

Triangulation

Different methods:

1. Linear triangulation
2. Iterative Minimization of the Reconstruction Error
3. Minimizing Reconstruction Error via Root Finding

Linear Triangulation

- Each 3D point X satisfies:

$$x \stackrel{!}{=} \hat{x} := PX, \quad x' \stackrel{!}{=} \hat{x}' := P'X$$

yielding

$$\begin{pmatrix} x_3 P_{1,.}^T - x^T P_{3,1} \\ x_3 P_{2,.}^T - x^T P_{3,2} \\ x_3 P_{3,.}^T - x^T P_{3,3} \end{pmatrix} X = 0$$

of which 2 rows are independent,
and the same for x' and P'.
Solve $AX = 0$ for

$$A(x, P, x', P') := \begin{pmatrix} x_3 P_{1,.}^T - x^T P_{3,1} \\ x_3 P_{2,.}^T - x^T P_{3,2} \\ x'_3 P'_{1,.}{}^T - x'^T P'_{3,1} \\ x'_3 P'_{2,.}{}^T - x'^T P'_{3,2} \end{pmatrix}$$

Linear Triangulation (2/2)

- Exact solutions to

$$AX = 0, \quad X \neq 0$$

for a 4×4 matrix A may not exist if noise is involved.

- Solve approximately via SVD:

$$A = USV^T, \quad S = \text{diag}(s_1, s_2, s_3, s_4), s_1 \geq s_2 \geq s_3 \geq s_4, \text{SVD}$$
$$X \approx V_{.,4}$$

Iterative Minimization of the Reconstruction Error

- solve N separate problems, one for each point X_n ($n = 1, \ldots, N$):

$$\begin{aligned} &\text{minimize } d(x_n, \hat{x}_n)^2 + d(x'_n, \hat{x}'_n)^2 \\ &\quad \text{with } \hat{x}_n := PX_n = X_n, \quad n = 1, \ldots, N, \quad \text{for } P := [I \mid 0] \\ &\qquad\qquad \hat{x}'_n := P'X_n, \quad n = 1, \ldots, N, \\ &\quad \text{over } X_n \end{aligned}$$

- 3 parameters each (P' is fixed)
- as in chapter 3:
  - iteratively minimize using Levenberg-Marquardt

Outline

Monocular Visual SLAM

Calibrated camera K with known start pose $Q^{(0)}$

Do forever (time t):

1. Get image $I^{(t)}$ from the camera
2. Find interesting points in $I^{(t)}$ and their descriptors
3. Match interesting points of two consecutive images $I^{(t-1)}, I^{(t)}$ based on their descriptors to get corresponding points
4. Minimize reconstruction loss for all corresponding points in the two images to get new camera pose $Q^{(t)}$ and 3D points $X^{(t)}$

- **localization:**
  $Q^{(t)}$ describes the trajectory of the camera
  (and thus the vehicle)
- **mapping:**
  $X^{(t)}$ describes the scene

Many detail problems still to discuss. Many variants exist.

Stereo Visual SLAM

Calibrated cameras K, K' with known start poses $Q^{(0)}, Q'^{(0)}$

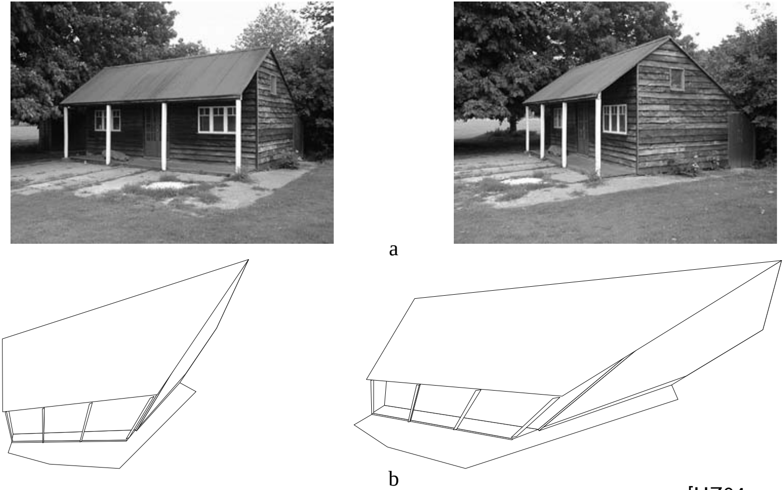
Do forever (time t):

1. Get two images $I^{(t)}, I'^{(t)}$ from the two cameras
2. Find interesting points in both $I^{(t)}, I'^{(t)}$ and their descriptors
3. Match interesting points of all four images $I^{(t-1)}, I'^{(t-1)}, I^{(t)}, I'^{(t)}$ based on their descriptors to get corresponding points
4. Minimize reconstruction loss for all corresponding points in the four images to get new camera poses $Q^{(t)}, Q'^{(t)}$ and 3D points $X^{(t)}$

- **localization:**
  $Q^{(t)}, Q'^{(t)}$ describes the trajectory of the cameras (and thus the vehicle)
- **mapping:**
  $X^{(t)}$ describes the scene

Many detail problems still to discuss. Many variants exist.

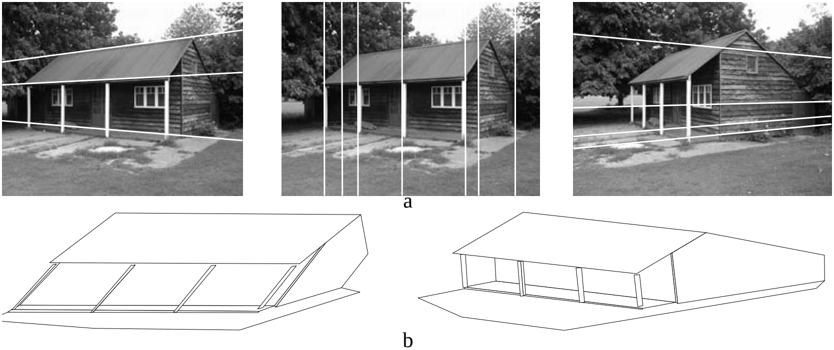
Example / Projective Reconstruction



Note: Additional knowledge: none.

[HZ04, p. 267]

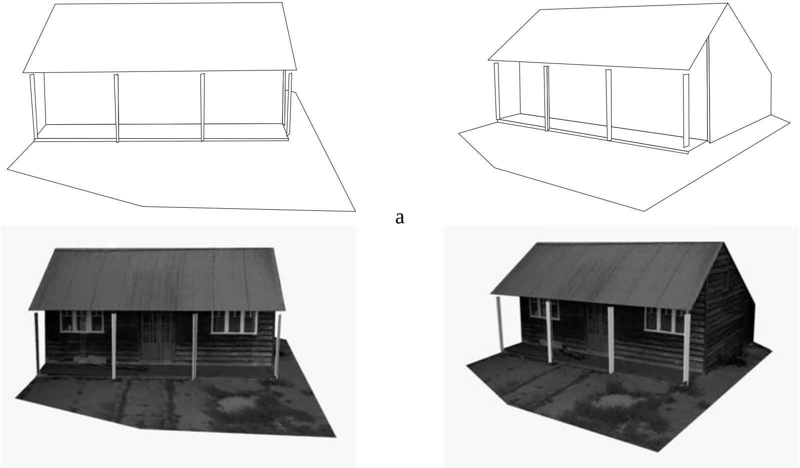
Example / Affine Reconstruction



Note: Additional knowledge: three sets of parallel lines.

[HZ04, p. 270]

Example / Metric Reconstruction



a

b

Note: Additional knowledge: additionally lines in different sets are orthogonal.

[HZ04, p. 274]

Outlook

- methods applicable in two settings:
  - two cameras, single shot: **stereo vision**
  - one camera, sequence of shots: **structure from motion**, **monocular visual SLAM**

Outlook

- methods applicable in two settings:
  - two cameras, single shot: **stereo vision**
  - one camera, sequence of shots: **structure from motion**, **monocular visual SLAM**
- structure from motion:
  - do not compute everything from scratch for every frame
    - **tracking** (computer vision terminology)
    - **online updates** (machine learning terminology)

Outlook

- methods applicable in two settings:
  - two cameras, single shot: **stereo vision**
  - one camera, sequence of shots: **structure from motion**, **monocular visual SLAM**
- structure from motion:
  - do not compute everything from scratch for every frame
    - **tracking** (computer vision terminology)
    - **online updates** (machine learning terminology)
- methods to combine stereo vision and structure from motion
  - two cameras, sequence of shots
  - the very same methods, just for 4 views instead of 2.
  - some new concepts (e.g., trifocal tensor for 3 views)

Summary (1/4)

- There exist several methods for **simultaneous localization and mapping (SLAM)**
  - We discussed: **bundle adjustment**: minimize a loss between
    - in two views observed and
    - from two unknown 2D-projections of unknown 3D points reconstructed

  corresponding points.

- **Cameras** are described by linear projective maps $P : \mathbb{P}^3 \rightarrow \mathbb{P}^2$ ($= 4 \times 3$ matrices)
  usually structured as $P = K[R \mid t]$:
  - **camera calibration matrix** K (5 intrinsic parameters)
  - **camera pose** $[R \mid t]$ (6 external parameters)
  - finite vs infinte (esp. affine) cameras; pinhole camera

Summary (2/4)

- The geometric relation between two 2D views on a 3D scene can be represented by the 3×3 **fundamental matrix** F:
  - maps points in 1st view to **epipolar line** of all possible corresponding points in 2nd view.
  - $x'Fx = 0$ for corresponding points x, x'
  - For two cameras P, P' their fundamental matrix can be computed as:

$$F = [e']_\times P' P^+, \quad \text{with } \textbf{epipole} \text{ in 2nd view } e'$$

  - For a fundamental matrix F, several pairs of cameras are possible. Two **canonical cameras** P, P' can be computed as:

$$P = [I \mid 0], \quad P' = [[e']_\times F \mid e']$$

Summary (3/4)

- To compute the fundamental matrix from point correspondences several methods exist.
  - Problem has 7 degrees of freedom (8 ratios; singular)
  - Linear methods
    - 8-point algorithm: solve a linear system of equations / SVD
    - 7-point algorithm: solve a linear system of equations / SVD
    - enforce singularity
  - Iterative minimization of the reconstruction error
- To estimate 3D point positions from their observed images under known 2D projection(s):
  triangulation. Several methods exist:
  - Linear methods
    - individually for each 3D point
    - solve a 4×4 linear system of equations / SVD
  - Iterative minimization of the reconstruction error
  - Minimizing Reconstruction Error via Root Finding

Summary (4/4)

- **Metric reconstruction:**
  - With just multiple 2D views of a scene, it can only be reconstructed up to a projectivity.
  - requires either background knowledge or
  - **camera calibration**: estimate the intrinsic parameters of the camera calibration matrix from a known scene.

Further Readings

- Reconstruction ambiguity: [HZ04, ch. 10].
- Computing the Fundamental Matrix: [HZ04, ch. 11].
- Triangulation: [HZ04, ch. 12].
- Camera models: [HZ04, ch. 6].
- The Fundamental Matrix: [HZ04, ch. 9].

References

Richard Hartley and Andrew Zisserman.
Multiple view geometry in computer vision.
Cambridge university press, 2004.