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

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4 Mon.</td>
<td>0. Introduction</td>
</tr>
<tr>
<td>17.4 Mon.</td>
<td>— EASTER MONDAY —</td>
</tr>
<tr>
<td>24.4 Mon.</td>
<td>2. Projective Geometry in 2D: b. Projective Transformations</td>
</tr>
<tr>
<td>1.5 Mon.</td>
<td>— LABOR DAY —</td>
</tr>
<tr>
<td>8.5 Mon.</td>
<td>3. Projective Geometry in 3D: a. Projective Space</td>
</tr>
<tr>
<td>15.5 Mon.</td>
<td>3. Projective Geometry in 3D: b. Quadrics, Transformations</td>
</tr>
<tr>
<td>22.5 Mon.</td>
<td>4. Estimating 2D Transformations: a. Direct Linear Transformation</td>
</tr>
<tr>
<td>29.5 Mon.</td>
<td>4. Estimating 2D Transformations: b. Iterative Minimization</td>
</tr>
<tr>
<td>5.6 Mon.</td>
<td>— PENTECOSTE DAY —</td>
</tr>
<tr>
<td>12.6 Mon.</td>
<td>5. Interest Points: a. Edges and Corners</td>
</tr>
<tr>
<td>19.6 Mon.</td>
<td>5. Interest Points: b. Image Patches</td>
</tr>
<tr>
<td>3.7 Mon.</td>
<td>6. Simultaneous Localization and Mapping: b. Triangulation</td>
</tr>
</tbody>
</table>
Outline

1. Overview of SLAM
2. Camera Models
3. Two Cameras and the Fundamental Matrix
4. Triangulation
5. Putting it all Together
Outline

1. Overview of SLAM

2. Camera Models

3. Two Cameras and the Fundamental Matrix

4. Triangulation

5. Putting it all Together
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

Fig. 6.1. Pinhole camera geometry. C is the camera centre and p the principal point. The camera centre is here placed at the coordinate origin. Note the image plane is placed in front of the camera centre.

Computations that the point \((X, Y, Z)\) is mapped to the point \((fX/Z, fY/Z, f)\) on the image plane. Ignoring the final image coordinate, we see that \((X, Y, Z) \mapsto (fX/Z, fY/Z)\) describes the central projection mapping from world to image coordinates. This is a mapping from Euclidean 3-space \(\mathbb{R}^3\) to Euclidean 2-space \(\mathbb{R}^2\).

The centre of projection is called the camera centre. It is also known as the optical centre. The line from the camera centre perpendicular to the image plane is called the principal axis or principal ray of the camera, and the point where the principal axis meets the image plane is called the principal point. The plane through the camera centre parallel to the image plane is called the principal plane of the camera.

Central projection using homogeneous coordinates. If the world and image points are represented by homogeneous vectors, then central projection is very simply expressed as a linear mapping between their homogeneous coordinates. In particular, \((6.1)\) may be written in terms of matrix multiplication as

\[
\begin{pmatrix}
X \\
y \\
z \\
1
\end{pmatrix} \mapsto \begin{pmatrix}
fX \\
fY \\
Z
\end{pmatrix} = \begin{pmatrix}
0 & f & 0 & 0 \\
0 & 0 & f & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \begin{pmatrix}
X \\
y \\
z \\
1
\end{pmatrix}.
\]

\([HZ04, \text{p. 154}]\)

The matrix in this expression may be written as \(\text{diag}(f, f, 1)\) \([I | 0]\) where \(\text{diag}(f, f, 1)\) is a diagonal matrix and \([I | 0]\) represents a matrix divided up into a \(3 \times 3\) block (the identity matrix) plus a column vector, here the zero vector.

We now introduce the notation \(X\) for the world point represented by the homogeneous 4-vector \((X, Y, Z, 1)\), \(x\) for the image point represented by a homogeneous 3-vector, and \(P\) for the \(3 \times 4\) homogeneous camera projection matrix. Then \((6.2)\) is written compactly as

\[x = PX\]

which defines the camera matrix for the pinhole model of central projection as \(P = \text{diag}(f, f, 1)\) \([I | 0]\).
Pinhole Camera / Homogeneous Coordinates

inhomogeneous coordinates:

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

homogeneous coordinates:

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

\[P = \text{diag}(f, f, 1)[I \mid 0]\]
Pinhole Camera / Principal Point Offset

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

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

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')

\[ p = R(p' - c') \]

\[
\begin{pmatrix}
    x' \\
y' \\
z' \\
1
\end{pmatrix}
\mapsto
\begin{pmatrix}
    R & 0 & 0 \\
    0 & 1 & 0 \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x' \\
y' \\
z' \\
1
\end{pmatrix}
- \begin{pmatrix}
    x_c' \\
y_c' \\
z_c' \\
1
\end{pmatrix}
\]

\[ P = KR[I | -c'] \]

without explicit camera center:

\[ P = K[R | t], \quad t := -Rc' \]
CCD Cameras

CCD camera:

- pixels may be not square – different width $\alpha_x$ and height $\alpha_y$

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

- finite projective camera:

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

- $s$ 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 $3 \times 3$ 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 $3 \times 4$ matrix $P$ with det $P_{1:3,1:3} \neq 0$ is such a finite projective camera
Outline

1. Overview of SLAM

2. Camera Models

3. Two Cameras and the Fundamental Matrix

4. Triangulation

5. Putting it all Together
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
Epipolar Geometry / Example

Fig. 9.3. Converging cameras. (a) Epipolar geometry for converging cameras. (b) and (c) A pair of images with superimposed corresponding points and their epipolar lines (in white). The motion between the views is a translation and rotation. In each image, the direction of the other camera may be inferred from the intersection of the pencil of epipolar lines. In this case, both epipoles lie outside of the visible image.

- An epipolar plane is a plane containing the baseline. There is a one-parameter family (a pencil) of epipolar planes.
- An epipolar line is the intersection of an epipolar plane with the image plane. All epipolar lines intersect at the epipole. An epipolar plane intersects the left and right image planes in epipolar lines, and defines the correspondence between the lines.

Examples of epipolar geometry are given in figure 9.3 and figure 9.4. The epipolar geometry of these image pairs, and indeed all the examples of this chapter, is computed directly from the images as described in section 11.6 (p 290).

9.2 The fundamental matrix

The fundamental matrix is the algebraic representation of epipolar geometry. In the following we derive the fundamental matrix from the mapping between a point and its epipolar line, and then specify the properties of the matrix.

Given a pair of images, it was seen in figure 9.1 that to each point \( x \) in one image, there exists a corresponding epipolar line \( l' \) in the other image. Any point \( x' \) in the second image matching the point \( x \) must lie on the epipolar line \( l' \). The epipolar line...
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 \( \ell' \) of their possible correspondences in the second view.
Fundamental Matrix (2/2)

- **construct \( \ell \):**
  1. possible 3D source points of \( x = PX \):

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

  2. their 2D images in second view:

\[
x' = P'(P^+ x + \lambda C) = P' P^+ x + \lambda P' C
\]

esp. \( x' := P' P^+ x, \quad for \ \lambda := 0 \)

\[
e' = P' C, \quad for \ \lambda := \infty \ epipole \ of \ second \ view
\]

  3. \( \ell' \) 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

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

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.

\[ \tilde{P} = PH, \quad \tilde{P}' = P'H, \quad \tilde{C} = H^{-1}C \]

\[ \tilde{P}^+ = H^{-1}P^+ \]

\[ \tilde{F} = [\tilde{P}' \tilde{C}] \times \tilde{P}' \tilde{P}^+ \]

\[ = [P'HH^{-1}C] \times P'HH^{-1}P^+ = [P'C] \times P'P^+ = F \]

▷ Cameras can be chosen as

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

\[ \sim F(P, P') = [e'] \times K'RK^{-1} = [e'] \times [e'] \times F \propto F \]
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 $3 \times 3$ 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

\[ (x', y')F(x, y)^T = 0 \]

\[ \Rightarrow \begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix} \text{vect}(F) = 0 \]

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

\[
\begin{pmatrix}
x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\
x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\
\vdots \\
x'_Nx_N & x'_Ny_N & x'_N & y'_Nx_N & y'_Ny_N & y'_N & x_N & y_N & 1
\end{pmatrix} \text{vect}(F) = 0
\]

- 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 \( \det F = 0 \):

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

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

[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 $\det F = 0$:

   $\det(\lambda F_1 + (1 - \lambda) F_2) = 0$

Find root $\lambda^* \text{ of this polynomial of degree 3, then}$

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

- 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

1. Overview of SLAM
2. Camera Models
3. Two Cameras and the Fundamental Matrix
4. Triangulation
5. Putting it all Together
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:

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

yielding

$$\begin{pmatrix}
x_3P_{1,.}^T - x^TP_{3,1} \\
x_3P_{2,.}^T - x^TP_{3,2} \\
x_3P_{3,.}^T - x^TP_{3,3}
\end{pmatrix} X = 0$$

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

Solve $AX = 0$ for

$$A(x, P, x', P') := \begin{pmatrix}
x_3P_{1,.}^T - x^TP_{3,1} \\
x_3P_{2,.}^T - x^TP_{3,2} \\
x'_3P_{1,.}^T - x'^TP_{3,1}' \\
x'_3P_{2,.}^T - x'^TP_{3,2}'
\end{pmatrix}$$
Linear Triangulation (2/2)

- Exact solutions to

\[ AX = 0, \quad X \neq 0 \]

for a $4 \times 4$ matrix $A$ may not exist if noise is involved.

- Solve approximately via SVD:

\[ A = USV^T, \quad S = \text{diag}(s_1, s_2, s_3, s_4), \quad s_1 \geq s_2 \geq s_3 \geq s_4, \quad \text{SVD} \]

\[ X \approx V_{:,4} \]
Iterative Minimization of the Reconstruction Error

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

$$\text{minimize } d(x_n, \hat{x}_n)^2 + d(x'_n, \hat{x}'_n)^2$$

with $\hat{x}_n := PX_n = X_n, \quad n = 1, \ldots, N,$ for $P := [I \mid 0]$

$\hat{x}'_n := P'X_n, \quad n = 1, \ldots, N,$

over $X_n$

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

1. Overview of SLAM
2. Camera Models
3. Two Cameras and the Fundamental Matrix
4. Triangulation
5. Putting it all Together
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

Figure 10.3. Projective reconstruction. (a) Original image pair. (b) 2 views of a 3D projective reconstruction of the scene. The reconstruction requires no information about the camera matrices, or information about the scene geometry. The fundamental matrix $F$ is computed from point correspondences between the images, camera matrices are retrieved from $F$, and then 3D points are computed by triangulation from the correspondences. The lines of the wireframe link the computed 3D points.

This is an enormously significant result, since it implies that one may compute a projective reconstruction of a scene from two views based on image correspondences alone, without knowing anything about the calibration or pose of the two cameras involved. In particular the true reconstruction is within a projective transformation of the projective reconstruction. Figure 10.3 shows an example of 3D structure computed as part of a projective reconstruction from two images.

In more detail suppose the true Euclidean reconstruction is $(P_E, P'_E, \{X_E^i\})$ and the projective reconstruction is $(P, P'_H, \{X^i\})$, then the reconstructions are related by a non-singular matrix $H$ such that

$$P_E = PH^{-1},$$
$$P'_E = P'_H H^{-1},$$
$$X_E^i = H X^i$$

(10.1)

where $H$ is a 4×4 homography matrix which is unknown but the same for all points.

For some applications projective reconstruction is all that is required. For example, questions such as "at what point does a line intersect a plane?", "what is the mapping between two views induced by particular surfaces, such as a plane or quadric?" can be dealt with directly from the projective reconstruction. Furthermore it will be seen in the sequel that obtaining a projective reconstruction of a scene is the first step towards affine or metric reconstruction.

10.4 Stratified reconstruction

The "stratified" approach to reconstruction is to begin with a projective reconstruction and then to refine it progressively to an affine and finally a metric reconstruction, if...
Example / Affine Reconstruction

Note: Additional knowledge: three sets of parallel lines.

[HZ04, p. 270]
Example / Metric Reconstruction

Figure 10.5. Metric reconstruction. The affine reconstruction of Figure 10.4 is upgraded to metric by computing the image of the absolute conic. The information used is the orthogonality of the directions of the parallel line sets shown in Figure 10.4, together with the constraint that both images have square pixels. The square pixel constraint is transferred from one image to the other using \( H_∞ \).

(a) Two views of the metric reconstruction. Lines which are perpendicular in the scene are perpendicular in the reconstruction and also the aspect ratio of the sides of the house is veridical.

(b) Two views of a texture mapped piecewise planar model built from the wireframes.

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)
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 infinite (esp. affine) cameras; pinhole camera
Summary (2/4)

- The geometric relation between two 2D views on a 3D scene can be represented by the $3 \times 3$ 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:
  \[
  F = [e']_\times P'P^+,
  \]
  with epipole in 2nd view $e'$
- For a fundamental matrix $F$, several pairs of cameras are possible. Two canonical cameras $P, P'$ can be computed as:
  \[
  P = [I | 0], \quad P' = [[e']_\times F | e']
  \]
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 $4 \times 4$ 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].
Richard Hartley and Andrew Zisserman.  
*Multiple view geometry in computer vision.*  