Introduction to Computer Vision for Robotics

Jan-Michael Frahm

Overview

- Camera model
- Multi-view geometry
- Camera pose estimation
- Feature tracking & matching
- Robust pose estimation
Introduction to Computer Vision for Robotics

Homogeneous coordinates

Unified notation:
include origin in affine basis

Affine basis matrix

Properties of affine transformation

Transformation $T_{affine}$ combines linear mapping and coordinate shift in homogeneous coordinates

- Linear mapping with $A_{3x3}$ matrix
- Coordinate shift with $t_3$ translation vector

$M' = T_{affine}M = \begin{bmatrix} A_{3x3} & t_3 \end{bmatrix}M$

$T_{affine} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- Parallelism is preserved
- Ratios of length, area, and volume are preserved
- Transformations can be concatenated:
  if $M_1 = T_1M$ and $M_2 = T_2M_1$ $\Rightarrow$ $M_2 = T_2T_1M = T_2T_1M$
Projective geometry

- Projective space $P^2$ is space of rays emerging from $O$
  - view point $O$ forms projection center for all rays
  - rays $v$ emerge from viewpoint into scene
  - ray $g$ is called projective point, defined as scaled $v$: $g = \lambda v$

$g(\lambda) = \frac{\lambda}{\lambda} = \lambda v, \lambda \in \mathbb{R} \neq 0$

Projective and homogeneous points

- Given: Plane $\Pi$ in $R^2$ embedded in $P^2$ at coordinates $w=1$
  - viewing ray $g$ intersects plane at $v$ (homogeneous coordinates)
  - all points on ray $g$ project onto the same homogeneous point $v$
  - projection of $g$ onto $\Pi$ is defined by scaling $v = g / w$

Finite and infinite points

- All rays \( g \) that are not parallel to \( \Pi \) intersect at an affine point \( v \) on \( \Pi \).
- The ray \( g(w=0) \) does not intersect \( \Pi \). Hence \( v_\infty \) is not an affine point but a direction. Directions have the coordinates \((x,y,z,0)^T\).
- Projective space combines affine space with infinite points (directions).

Affine and projective transformations

- Affine transformation leaves infinite points at infinity

\[
M'_\infty = T_{\text{affine}}M_\infty \Rightarrow \begin{bmatrix} X' \\ Y' \\ Z' \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}
\]

- Projective transformations move infinite points into finite affine space

\[
M = T_{\text{projective}}M_\infty \Rightarrow \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} X' \\ Y' \\ Z' \\ w' \end{bmatrix} = \lambda \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}
\]

Example: Parallel lines intersect at the horizon (line of infinite points).

We can see this intersection due to perspective projection!
Pinhole Camera (Camera obscura)

Camera obscura
(France, 1830)

Interior of camera obscura
(Sunday Magazine, 1838)

Pinhole camera model

Image sensor
aperture
object
View direction
Focal length f
image
center

Introduction to Computer Vision for Robotics
Perspective projection

- Perspective projection in $\mathbb{P}^3$ models pinhole camera:
  - scene geometry is affine $\mathbb{P}^3$ space with coordinates $M=(X,Y,Z,1)^T$
  - camera focal point in $O=(0,0,0,1)^T$, camera viewing direction along $Z$
  - image plane $(x,y)$ in $\Pi(\mathbb{P}^2)$ aligned with $(X,Y)$ at $Z=Z_0$
  - Scene point $M$ projects onto point $M_p$ on plane surface

Projective Transformation

- Projective Transformation maps $M$ onto $M_p$ in $\mathbb{P}^3$ space

$$\rho_{M_p} = T_p M \Rightarrow \rho = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{z_0} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\rho = \frac{Z}{Z_0} = \text{projective scale factor}$$

- Projective Transformation linearizes projection
Perspective Projection

Dimension reduction from $\mathbb{P}^3$ into $\mathbb{P}^2$ by projection onto $\Pi(\mathbb{P}^2)$

Perspective projection $P_p$ from $\mathbb{P}^3$ onto $\mathbb{P}^2$:

$$\rho m_p = D_p T_p M = P_0 M \Rightarrow \rho \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \rho \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad \rho = \frac{Z}{Z_0}$$

Image plane and image sensor

- A sensor with picture elements (Pixel) is added onto the image plane

Pixel coordinates $m = (y, x)^T$

Image center $c = (c_x, c_y)^T$

Pixel scale $f = (f_x, f_y)^T$

Focal length $Z_0$

Projection center

Image-sensor mapping: $m = K m_p$

- Pixel coordinates are related to image coordinates by affine transformation $K$ with five parameters:
  - Image center $c$ at optical axis
  - Distance $Z_0$ (focal length) and Pixel size determines pixel resolution $f_x, f_y$
  - Image skew $s$ to model angle between pixel rows and columns
Projection in general pose

\[
T_{\text{cam}} = \begin{bmatrix} R & C \\ 0^T & 1 \end{bmatrix}
\]

Projection: \( \rho m_p = PM \)

\[
T_{\text{scene}} = T_{\text{cam}}^{-1} = \begin{bmatrix} R^T & -R^T C \\ 0^T & 1 \end{bmatrix}
\]

World coordinates

Projection matrix \( P \)

- Camera projection matrix \( P \) combines:
  - inverse affine transformation \( T_{\text{cam}}^{-1} \) from general pose to origin
  - Perspective projection \( P_0 \) to image plane at \( Z_0 = 1 \)
  - affine mapping \( K \) from image to sensor coordinates

scene pose transformation: \( T_{\text{scene}} = \begin{bmatrix} R^T & -R^T C \\ 0^T & 1 \end{bmatrix} \)

projection: \( P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [I \ 0] \)

sensor calibration: \( K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \)

\[
\Rightarrow \rho m = PM, \quad P = KP_0 T_{\text{scene}} = K \begin{bmatrix} R^T & -R^T C \end{bmatrix}
\]
2-view geometry: F-Matrix

Projection onto two views:

\[ P_0 = K_0 R_0^{-1} [I \ 0] \]
\[ \rho_0 m_b = P_0 M = K_0 R_0^{-1} [I \ 0] M \]
\[ \Rightarrow \rho_0 m_b = K_0 R_0^{-1} [I \ 0] M_{\infty} \]
\[ P_1 = K_1 R_1^{-1} [I \ -C_1] \]
\[ \rho_1 m_1 = P_1 M = K_1 R_1^{-1} [I \ -C_1] M \]
\[ \Rightarrow \rho_1 m_1 = K_1 R_1^{-1} [I \ 0] M_{\infty} + K_1 R_1^{-1} [I \ -C_1] O \]
\[ \rho_0 m_b = K_0 R_0^{-1} K_1 R_1^{-1} \rho_1 m_1 - K_1 R_1^{-1} C_1 \]
\[ \Rightarrow \rho_0 m_1 = \rho_0 H_{\infty} m_b + e_1 \]

Epipolar line

\[ M = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = M_{\infty} + O \]

\[ \rho_{\infty} \Rightarrow \rho_{\infty} = 1 \]

The Fundamental Matrix \( F \)

- The projective points \( e_1 \) and \( (H_{\infty} m_b) \) define a plane in camera 1 (epipolar plane \( \Pi_e \))
- the epipolar plane intersect the image plane 1 in a line (epipolar line \( u_e \))
- the corresponding point \( m_1 \) lies on that line: \( m_1^T u_e = 0 \)
- If the points \( (e_1), (m_1), (H_{\infty} m_b) \) are all collinear, then the collinearity theorem applies: \( (m_1^T e_1 \times H_{\infty} m_b) = 0 \).

\[ \text{collinearity of } m_1, e_1, H_{\infty} m_b \Rightarrow m_1^T (\{ e_1, H_{\infty} m_b \}) = 0 \]

\[ [e]_x = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \]

\[ F_{3 \times 3} \]

Fundamental Matrix \( F \)

Epipolar constraint

\[ F = [e_1]_x H_{\infty} \]

\[ m_1^T F m_b = 0 \]
Estimation of $F$ from correspondences

- Given a set of corresponding points, solve linearly for the 9 elements of $F$ in projective coordinates.
- Since the epipolar constraint is homogeneous up to scale, only eight elements are independent.
- Since the operator $[e]_x$ and hence $F$ have rank 2, $F$ has only 7 independent parameters (all epipolar lines intersect at $e$).
- Each correspondence gives 1 collinearity constraint.

$=>$ solve $F$ with minimum of 7 correspondences.

For $N>7$ correspondences minimize distance point-line:

$$\sum_{n=0}^{N} (m_{i,n}^T F m_{0,n})^2 \Rightarrow \min!$$

$$m_{i,n}^T F m_{0,n} = 0 \quad \text{det}(F) = 0 \quad \text{(Rank 2 constraint)}$$

The Essential Matrix $E$

- $F$ is the most general constraint on an image pair. If the camera calibration matrix $K$ is known, then more constraints are available.
- Essential Matrix $E$

$$m_{i,n}^T F m_{0,n} = (K\tilde{m}_i)^T F (K\tilde{m}_0) = \tilde{m}_i^T \underbrace{(K^T FK)}_{E} \tilde{m}_0$$

$$E = [e]_x R \text{ with } [e]_x = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix}$$

- $E$ holds the relative orientation of a calibrated camera pair. It has 5 degrees of freedom: 3 from rotation matrix $R_{ik}$, 2 from direction of translation $e$, the epipole.
Estimation of $P$ from $E$

- From $E$ we can obtain a camera projection matrix pair: $E = U \text{diag}(0,0,1)V^T$
- $P_0 = [I_{3x3} | 0_{3x1}]$ and there are four choices for $P_1$:

$$P_1 = [UWV^T | +u_3] \text{ or } P_1 = [UWV^T | -u_3] \text{ or } P_1 = [UW^TV^T | +u_3] \text{ or } P_1 = [UW^TV^T | -u_3]$$

with $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

four possible configurations:

only one with 3D point in front of both cameras

---

3D Feature Reconstruction

- corresponding point pair $(m_0, m_1)$ is projected from 3D feature point $M$
- $M$ is reconstructed from by $(m_0, m_1)$ triangulation
- $M$ has minimum distance of intersection

$$\|d\|^2 \Rightarrow \text{min!}$$

constraints:

$$l_0^T d = 0$$

$$l_1^T d = 0$$

minimize reprojection error:

$$(m_0 - P_0 M)^2 + (m_1 - P_1 M)^2 \Rightarrow \text{min.}$$
Multi View Tracking

- 2D match: Image correspondence \((m_1, m_i)\)
- 3D match: Correspondence transfer \((m_i, M)\) via \(P_i\)
- 3D Pose estimation of \(P_i\) with \(m_i - P_i M \Rightarrow \text{min.}\)

\[
\text{Minimize global reprojection error: } \sum_{i=0}^{N} \sum_{k=0}^{K} \| m_{k,i} - P_i M \|_2^2 \Rightarrow \text{min!}
\]

Correspondences matching vs. tracking

- Image-to-image correspondences are essential to 3D reconstruction

**SIFT-matcher**
- Extract features independently and then match by comparing descriptors [Lowe 2004]

**KLT-tracker**
- Extract features in first images and find same feature back in next view [Lucas & Kanade 1981], [Shi & Tomasi 1994]
  - Small difference between frames
  - Potential large difference overall
SIFT-detector

- Scale and image-plane-rotation invariant feature descriptor
  [Lowe 2004]

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters
Difference of Gaussian for Scale invariance

- Difference-of-Gaussian with constant ratio of scales is a close approximation to Lindeberg’s scale-normalized Laplacian [Lindeberg 1998]
Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:
  \[ D(x) = D + \frac{\partial D}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \]
- Offset of extremum (use finite differences for derivatives):
  \[ \hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \]

Orientation normalization

- Histogram of local gradient directions computed at selected scale
- Assign principal orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)
Example of keypoint detection

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)

(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

SIFT vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions
  example 2x2 histogram array

Image gradients  →  Keypoint descriptor
Sift feature detector

Introduction to Computer Vision for Robotics

Robust data selection: RANSAC

• Estimation of plane from point data

Select $m$ samples

Compute $n$-parameter solution

Evaluate on (potential plane points)

{inlier samples}

Best solution so far?

yes

Keep it

no

1 - (1 - (\#inlier/\(#\text{potential plane points}\))) \times \text{steps} > 0.99

Best solution, \{inlier\}, \{outlier\}
RANSAC: Evaluate Hypotheses

- Evaluate cost function

\[
0 \leq \epsilon^2 \leq \frac{c}{1+c} \quad \text{if} \quad 2\sqrt{\epsilon^2 + c - c(1 + \epsilon^2)} \leq 1
\]

\[
\frac{c}{1+c} \leq \epsilon^2 < \frac{c}{1+c} \quad \text{else}
\]

- Tuple plane hypotheses
- Potential points

Robust Pose Estimation Calibrated Camera

Bootstrap

{2D-2D correspondences}

E-RANSAC (5-point algorithm)

Estimate \( P_0, P_1 \)

\( P_0, P_1 \)

nonlinear refinement with all inliers

Triangulate points

known 3D points

{2D-3D correspondences}

RANSAC (3-point algorithm)

nonlinear refinement with all inliers

\( P \)
References


---

References