Estimation of Camera Projection Matrix, Nonlinear

Computer Vision II
CSE 252B
Lecture 6
Announcements

• Assignment 2 is due Feb 7, 11:59 PM

• Reading
  – Sections 7.2 and 4.5
Single view geometry
Camera projection
Camera projection matrix

• Projective camera

When camera projection matrix is known up to projective transformation
3 \times 4 homogeneous (i.e., defined up to nonzero scale) matrix (11 degrees of freedom)

\[ x = PX \]

When parameters of camera projection matrix are known or known up to nonzero scale (11 degrees of freedom)

\[ P = K[R \mid t] \]
\[ P = KR[I \mid -\hat{C}] \]

where

- \( K \) Camera calibration matrix
- \( R \) Camera rotation matrix
- \( t \) Camera translation vector
- \( C \) Camera center
Camera projection matrix

\[ x = PX \]
Camera projection matrix

\[ x_i = P X_i \quad \forall i \]
Camera projection matrix

- Error in image point measurements
Camera projection matrix

Linear projection of points in homogeneous coordinates

$$\mathbf{x} = \mathbf{p}\mathbf{X}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} Xp_{11} + Yp_{12} + Zp_{13} +Tp_{14} \\ Xp_{21} + Yp_{22} + Zp_{23} +Tp_{24} \\ Xp_{31} + Yp_{32} + Zp_{33} +Tp_{34} \end{bmatrix}$$

Nonlinear projection of points in inhomogeneous coordinates

$$\tilde{x} = \frac{x}{w} = \frac{Xp_{11} + Yp_{12} + Zp_{13} +Tp_{14}}{Xp_{31} + Yp_{32} + Zp_{33} +Tp_{34}}$$

and

$$\tilde{y} = \frac{y}{w} = \frac{Xp_{21} + Yp_{22} + Zp_{23} +Tp_{24}}{Xp_{31} + Yp_{32} + Zp_{33} +Tp_{34}}$$

Do not forget this is a nonlinear projection
Estimation of camera projection matrix

\[ x_i = P X_i \forall i, \text{ solve for } P \]

- Given point correspondences \( x_i \leftrightarrow X_i \)
- Minimize geometric error \( \sum_i d(x_i, PX_i)^2 \)
- A nonlinear mapping requires a nonlinear optimization problem solver
  - Use linear estimation for initial estimate (to mitigate converging to local optimum)
  - Iterative process to determine global optimum
Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

• Minimize algebraic error between the projected point and two lines intersecting at the measured point

\[
[x]^\perp = \begin{bmatrix} \ell_1^\top \\ \ell_2^\top \end{bmatrix} \ell_1
\]

measured point \( \mathbf{x} \)

projected point \( \mathbf{PX} \)

\[
[\mathbf{x}]^\perp \mathbf{PX} = 0
\]

\[
\mathbf{x} = \mathbf{PX}
\]
Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

Given point correspondences \( x_i \leftrightarrow X_i \)

\[
x_i = PX_i \quad \forall i, \text{ solve for } P
\]

\[
[x_i^\perp \otimes X_i^\top]_i P X_i = 0 \quad \forall i, \text{ solve for } P
\]

\[
([x_i^\perp \otimes X_i^\top]) p = 0 \quad \forall i, \text{ solve for } p = \text{vec}(P^\top)
\]

\[
\begin{bmatrix}
x_1^\perp \otimes X_1^\top \\
x_2^\perp \otimes X_2^\top \\
\vdots \\
x_n^\perp \otimes X_n^\top
\end{bmatrix} p = 0
\]

\[
Ap = 0, \text{ where } A = \begin{bmatrix}
x_1^\perp \otimes X_1^\top \\
x_2^\perp \otimes X_2^\top \\
\vdots \\
x_n^\perp \otimes X_n^\top
\end{bmatrix}
\]

But, data normalization must be used
Linear estimation of camera projection matrix using the direct linear transformation (DLT) algorithm

• **Algorithm**

1. Data normalize the points

\[
x_{DN,i} = T x_i \forall i
\]
\[
X_{DN,i} = U X_i \forall i
\]

2. Estimate the data normalized camera projection matrix \( P_{DN} \) from the data normalized points

3. Data denormalize the data normalized camera projection matrix

\[
x_{DN,i} = P_{DN} X_{DN,i} \forall i
\]
\[
T x_i = P_{DN} U X_i
\]
\[
x_i = T^{-1} P_{DN} U X_i
\]
\[
x_i = P X_i \forall i, \text{ where } P = T^{-1} P_{DN} U
\]
Nonlinear estimation using iterative estimation methods

Measurement vector $X$  
Parameter vector $P$  
Nonlinear function $f$, where $X = f(P)$

Given an estimate of the parameter vector $\hat{P}$ and assuming $f$ is approximately linear in the region about $\hat{P}$

$$X \approx f(\hat{P}) + \frac{\partial \hat{X}}{\partial \hat{P}} (P - \hat{P})$$

$$X \approx \hat{X} + J(P - \hat{P}), \text{ where } \hat{X} = f(\hat{P}) \text{ and } J = \frac{\partial \hat{X}}{\partial \hat{P}}$$

$$X - \hat{X} \approx J(P - \hat{P})$$

$\epsilon \approx J\delta$, solve for $\delta$ where $\epsilon = X - \hat{X}$ and $\delta = P - \hat{P}$

Then

$$\delta = P - \hat{P}$$

$$\hat{P} + \delta = P$$

Perform again (i.e., iterate) using resulting $P$ as $\hat{P}$
Nonlinear estimation using the Levenberg-Marquardt algorithm

Given

Measurement vector $\mathbf{X}$ with associated covariance matrix $\Sigma_{\mathbf{X}}$
Parameter vector $\hat{\mathbf{P}}$ (initial estimate)
Nonlinear function $f$, where $\hat{\mathbf{X}} = f(\hat{\mathbf{P}})$

Objective

Find $\hat{\mathbf{P}}$ that minimizes $\mathbf{e}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{e}$, where $\mathbf{e} = \mathbf{X} - \hat{\mathbf{X}}$
Nonlinear estimation using the Levenberg-Marquardt algorithm

Algorithm

1. \( \lambda = 0.001 \) and \( \epsilon = X - \hat{X} \)

2. Jacobian \( J = \frac{\partial \hat{X}}{\partial \hat{P}} \)

3. Weighted least-squares normal equations \( (J^T \Sigma_X^{-1} J) \delta = J^T \Sigma_X^{-1} \epsilon \)

4. Augmented normal equations \( (J^T \Sigma_X^{-1} J + \lambda I) \delta = J^T \Sigma_X^{-1} \epsilon \), solve for \( \delta \)

5. Candidate parameter vector \( \hat{P}_0 = \hat{P} + \delta \)

6. \( \hat{X}_0 = f(\hat{P}_0) \), then \( \epsilon_0 = X - \hat{X}_0 \)

7. If candidate cost \( \epsilon_0^T \Sigma_X^{-1} \epsilon_0 \geq \epsilon^T \Sigma_X^{-1} \epsilon \), then
   - \( \lambda = 10\lambda \)
   - Go to step 4 (this does not count as an iteration)
   - else
     - \( \hat{P} = \hat{P}_0 \) and \( \epsilon = \epsilon_0 \)
     - \( \lambda = \lambda/10 \)
     - Go to step 2 or terminate (this counts as an iteration)
Nonlinear estimation using the Levenberg-Marquardt algorithm

Termination criteria

- Using ratio tolerance \( \tau_1 \), when

\[
1 - \frac{\text{cost}_{\text{current}}}{\text{cost}_{\text{previous}}} \leq \tau_1
\]

- Using difference tolerance \( \tau_2 \), when

\[
\text{cost}_{\text{previous}} - \text{cost}_{\text{current}} \leq \tau_2
\]

- When the number of function evaluations (i.e., inner loop) has reached

100(number of parameters + 1)
Parameter vector

- May get stuck in local minimum of high dimensional space
- **Use minimal parameterizations towards mitigating this**

![Graph showing local and global minimums](image-url)
Parameterization of a homogeneous vector

Let the homogeneous vector $\tilde{v} = (a, b^\top)^\top \in \mathbb{R}^n$, where $\|\tilde{v}\| = 1$ (i.e., $\tilde{v}$ is a unit vector), be parameterized as

$$v = \frac{2}{\text{sinc}(\cos^{-1}(a))} b \in \mathbb{R}^{n-1}$$

then, if $\|v\| > \pi$, normalized by

$$v - \left(1 - \frac{2\pi}{\|v\|} \left[ \frac{\|v\| - \pi}{2\pi} \right] \right) v$$

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix.
Parameterization of a homogeneous vector

The parameterized homogeneous vector $\mathbf{v}$ is deparameterized as the homogeneous vector

$$
\mathbf{\tilde{v}} = \left( \cos \left( \frac{\|\mathbf{v}\|}{2} \right), \frac{\sin \left( \frac{\|\mathbf{v}\|}{2} \right)}{\frac{\|\mathbf{v}\|}{2}} \mathbf{v}^\top \right)^\top \in \mathbb{R}^n
$$

Unit vector with nonnegative first element

$$
\mathbf{\tilde{v}} = (a, \mathbf{b}^\top)^\top, \text{ where } a = \cos \left( \frac{\|\mathbf{v}\|}{2} \right) \text{ and } \mathbf{b} = \frac{\sin \left( \frac{\|\mathbf{v}\|}{2} \right)}{\frac{\|\mathbf{v}\|}{2}} \mathbf{v}
$$

where $\|\mathbf{\tilde{v}}\| = 1$ and $a$ is nonnegative. For the deparameterization,

$$
\frac{\partial \mathbf{\tilde{v}}}{\partial \mathbf{v}} = \frac{\partial (a, \mathbf{b}^\top)}{\partial \mathbf{v}} = \begin{bmatrix}
\frac{da}{\partial \mathbf{v}} \\
\frac{\partial b}{\partial \mathbf{v}}
\end{bmatrix} \in \mathbb{R}^{n \times (n-1)}
$$

where

$$
\frac{da}{\partial \mathbf{v}} = \begin{cases}
0^\top & \text{if } \|\mathbf{v}\| = 0 \\
-\frac{1}{2} \mathbf{b}^\top & \text{otherwise}
\end{cases}
$$

and

$$
\frac{\partial \mathbf{b}}{\partial \mathbf{v}} = \begin{cases}
\frac{1}{2} \mathbf{I} & \text{if } \|\mathbf{v}\| = 0 \\
\frac{\sin \left( \frac{\|\mathbf{v}\|}{2} \right)}{\frac{\|\mathbf{v}\|}{2}} \mathbf{I} + \frac{1}{4\|\mathbf{v}\|} \frac{d \sin \left( \frac{\|\mathbf{v}\|}{2} \right)}{d \frac{\|\mathbf{v}\|}{2}} \mathbf{v} \mathbf{v}^\top & \text{otherwise}
\end{cases}
$$
Parameterization of a homogeneous vector

- Sinc function

The sinc function

\[
sinc(x) = \begin{cases} 
1 & \text{if } x = 0 \\
\frac{\sin(x)}{x} & \text{otherwise}
\end{cases}
\]

The derivative is given by

\[
\frac{d}{dx} sinc(x) = \begin{cases} 
0 & \text{if } x = 0 \\
\frac{\cos(x)}{x} - \frac{\sin(x)}{x^2} & \text{otherwise}
\end{cases}
\]
Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

Parameter vector

11 x 1
\( \hat{p} \)

Parameterization of homogeneous vector

Mapping

\( \hat{x}_i = \hat{p}X_i \forall i \)

Always use the parameterized homogeneous vector, not the input homogeneous vector or matrix

Measurement vector

2n x 1

\( (\tilde{x}_1^T, \tilde{x}_2^T, \ldots, \tilde{x}_n^T)^T \) with associated covariance matrices \( \Sigma_{\tilde{x}_1}, \Sigma_{\tilde{x}_2}, \ldots, \Sigma_{\tilde{x}_n} \)

Cost

\[ \sum_i \epsilon_i^T \Sigma_{\tilde{x}_i}^{-1} \epsilon_i, \text{ where } \epsilon_i = \tilde{x}_i - \hat{x}_i \]

Jacobian

\[
J = \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_n
\end{bmatrix}, \text{ where } A_i = \frac{\partial \hat{x}_i}{\partial \hat{p}} = \frac{\partial \hat{x}_i}{\partial \hat{p}} \frac{\partial \hat{p}}{\partial \hat{p}}
\]

2n x 11
2 x 12
12 x 11

But, use data normalization
Projection of a point under the camera projection matrix

The homogeneous 3D point $\mathbf{X}$ is projected to the homogeneous 2D point $\mathbf{x}$ under the (homogeneous) camera projection matrix $\mathbf{P}$ by

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

Dehomogenizing the 2D point results in the mapping $\mathbf{X} \mapsto \tilde{\mathbf{x}}$. For this mapping

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}} = \frac{1}{w} \begin{bmatrix} \mathbf{X}^\top & \mathbf{0}^\top & -\tilde{x}\mathbf{X}^\top \\ \mathbf{0}^\top & \mathbf{X}^\top & -\tilde{y}\mathbf{X}^\top \end{bmatrix}$$

and

$$\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{X}} = \frac{1}{w} \begin{bmatrix} \mathbf{p}_1^\top - \tilde{x}\mathbf{p}_3^\top \\ \mathbf{p}_2^\top - \tilde{y}\mathbf{p}_3^\top \end{bmatrix}$$

(Used later in quarter)

where $w = \mathbf{p}_3^\top\mathbf{X}$ and $\mathbf{p}_i^\top$ is the $i$th row of $\mathbf{P}$. 
Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

Normal equations matrix

\[ U = J^T \Sigma^{-1}_x J, \text{ where } \Sigma^{-1}_x = 11 \times 11 \]

\[ \Sigma^{-1}_x = \begin{bmatrix}
\Sigma^{-1}_{\tilde{x}_1} \\
\Sigma^{-1}_{\tilde{x}_2} \\
\vdots \\
\Sigma^{-1}_{\tilde{x}_n}
\end{bmatrix} \]

Normal equations vector

\[ \epsilon_a = J^T \Sigma^{-1}_x \epsilon, \text{ where } \epsilon = X - \hat{X} \]

where \( X = (\tilde{x}_1^T, \tilde{x}_2^T, \ldots, \tilde{x}_n^T)^T \) and \( \hat{X} = (\hat{x}_1^T, \hat{x}_2^T, \ldots, \hat{x}_n^T)^T \)

Augmented normal equations

\[ S = U^* = U + \lambda I \quad 11 \times 11 \]

\[ e = \epsilon_a \quad 11 \times 1 \]

\[ S \delta_a = e, \text{ solve for } \delta_a \quad 11 \times 1 \]

Candidate parameter vector

\[ \hat{p}_0 = \hat{p} + \delta_a \]

Adjustment

Normalize parameterization of a homogeneous vector after adjustment
Data normalization with covariance propagation

Determine the similarity transformation $T$ such that the mean (i.e., centroid) of the transformed points is at the origin and their standard deviation from the origin is $\sqrt{2}$.

$$T = \begin{bmatrix} s & 0 & -s\mu_x \\ 0 & s & -s\mu_y \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } s = \sqrt{\frac{2}{\sigma_x^2 + \sigma_y^2}}$$

where $\mu_x$ and $\sigma_x^2$, and $\mu_y$ and $\sigma_y^2$ are the mean and variance of the $\tilde{x}$ and $\tilde{y}$ coordinates, respectively.

$$\begin{bmatrix} \tilde{x}_{DN} \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix}, \text{ where } A = sI \text{ and } t = \begin{bmatrix} -s\mu_x \\ -s\mu_y \end{bmatrix}$$

$$\begin{bmatrix} \tilde{x}_{DN} \\ 1 \end{bmatrix} = \begin{bmatrix} A\tilde{x} + t \\ 1 \end{bmatrix}$$

$$\tilde{x}_{DN} = A\tilde{x} + t$$

Covariance propagation

$$\Sigma_{\tilde{x}_{DN}} = J\Sigma_{\tilde{x}}J^T, \text{ where } J = \frac{\partial\tilde{x}_{DN}}{\partial\tilde{x}} = A = sI$$

If unknown, then assume to be identity
Nonlinear estimation of camera projection matrix using the Levenberg-Marquardt algorithm

• Algorithm

1. Data normalize the points

\[ \tilde{x}_{\text{DN},i} = A\tilde{x}_i + t \] and associated covariance matrices

\[ \Sigma_{\tilde{x}_{\text{DN},i}} = s^2 \Sigma_{\tilde{x}_i} \]

\[ X_{\text{DN},i} = UX_i \forall i \]

2. Estimate the data normalized camera projection matrix \( P_{\text{DN}} \) from the data normalized points

3. Data denormalize the data normalized camera projection matrix

\[ x_{\text{DN},i} = P_{\text{DN}}X_{\text{DN},i} \forall i \]

\[ Tx_i = P_{\text{DN}}UX_i \]

\[ x_i = T^{-1}P_{\text{DN}}UX_i \]

\[ x_i = PX_i \forall i, \text{ where } P = T^{-1}P_{\text{DN}}U \]
Decomposition of camera projection matrix

If estimated from points in a metric coordinate frame
Solve for camera center \( C \)

\[
P C = 0 \quad \text{(i.e., } C \text{ is null space of } P)\]

Solve for camera calibration matrix \( K \) and camera rotation matrix \( R \)

\[
P = K[R | t] = [KR | Kt] = [M | Kt], \text{ where } M = KR\]

1. \( RQ \) decomposition of \( M \). \( M = KR \), where \( K \) is upper triangular and \( R \) is orthogonal.

2. Scale \( K \) such that \( k_{33} = 1 \). \( K = \frac{1}{k_{33}}K \).

3. Ensure \( k_{11} \) and \( k_{22} \) are positive

\[
M = KR = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}
\]

If \( k_{11} < 0 \), then negate \( k_1 \) and \( r_1^T \)
If \( k_{22} < 0 \), then negate \( k_2 \) and \( r_2^T \)

4. Ensure \( R \) is a special orthogonal. If \( \det(R) = -1 \), then negate \( R \).
Next lecture

• Estimation of camera pose (calibrated camera), linear

• Reading
  – Lepetit, Moreno-Noguer, and Fua, “EPnP: An Accurate O(n) Solution to the PnP Problem”