Three View Geometry

Computer Vision II
CSE 252B
Lecture 16
Announcements

• Assignment 4 is due today, 11:59 PM
• Assignment 5 will be released today
  – Due Mar 22, 11:59 PM
Single view geometry

• Imaging a scene
Two view geometry

• Rotation about the same camera center
Two view geometry

• Imaging a plane
Two view geometry

• Imaging a 3D scene
Imaging geometry models

• Single view
  – Uncalibrated: camera projection matrix
    • 12 elements, 11 degrees of freedom
  – Calibrated: normalized camera projection matrix
    • 12 elements, 6 degrees of freedom

• Two views
  – Rotation about the same camera center
    • Uncalibrated: 2D projective transformation matrix
      – 9 elements, 8 degrees of freedom
    • Calibrated: 3D rotation matrix
      – 9 elements, 3 degrees of freedom
Imaging geometry models

- Two views
  - Imaging a plane
    - Uncalibrated: 2D projective transformation matrix
      - 9 elements, 8 degrees of freedom
    - Calibrated: 2D projective transformation matrix
      - 9 elements, 8 degrees of freedom
  - Imaging a 3D scene
    - Uncalibrated: fundamental matrix
      - 9 elements, 7 degrees of freedom
    - Calibrated: essential matrix
      - 9 elements, 5 degrees of freedom
Imaging geometry models

• Three views
  – Uncalibrated: trifocal tensor
    • 27 elements, 18 degrees of freedom
  – Calibrated: calibrated trifocal tensor
    • 27 elements, 11 degrees of freedom

Not covered in CSE 252B
Tensors

• Notation

A tensor with rank $r + s$ may be of mixed type $(r, s)$, consisting of $r$ contravariant (upper) indices and $s$ covariant (lower) indices.

(Scalars are) zeroth-rank tensors $s$
(Vectors are) first-rank tensors $v_i$ or $v^i$
Second-rank tensors $a_{ij}, a_{ij}^i$, or $a_i^j$
Third-rank tensors $b_{ijk}, b_{ijk}^k$, $b_{ij}^k$, or $b_{ijk}^k$

• Matrix as a tensor

The matrix $A$ where $a_{ij}$ is the element at $i$th row and $j$th column of $A$ corresponds to the second-rank tensor $a_i^j$ with one contravariant (upper) index and one covariant (lower) index.
Tensors

• Contraction

Einstein summation is the convention that repeated indices are implicitly summed over. For tensors, the sum over a repeated upper and lower index in a product is a contraction, e.g.,

\[ a = b_i c^i \quad a = \sum_i b_i c^i \quad a = b^\top c \]

\[ a^i = b_j^i c^j \quad a^i = \sum_j b_j^i c^j \quad a = B c \]

\[ a_i = b_i^j c_j \quad a_i = \sum_j b_i^j c_j \quad a = B^\top c \]

\[ a_i^j = b_k c_i^{jk} \quad a_i^j = \sum_k b_k c_i^{jk} \]

Unlike matrices and vectors, when multiplying tensors, the order of the entries does not matter, e.g.,

\[ a_i^j = b_k c_i^{jk} = c_i^{jk} b_k \]
Tensors

• Points and hyperplanes as tensors

A point as a (first-rank) tensor has a contravariant (upper) index. A hyperplane as a (first-rank) tensor has a covariant (lower) index, e.g., 2D points and 2D lines in three images as tensors are

Point in image 1 \( x^i = (x^1, x^2, x^3) \)
Point in image 2 \( x'^j = (x'^1, x'^2, x'^3) \)
Point in image 3 \( x''^k = (x''^1, x''^2, x''^3) \)

Line in image 1 \( \ell_i = (\ell_1, \ell_2, \ell_3) \)
Line in image 2 \( \ell'_j = (\ell'_1, \ell'_2, \ell'_3) \)
Line in image 3 \( \ell''_k = (\ell''_1, \ell''_2, \ell''_3) \)
Tensors

- 2D points and 2D lines in skew symmetric matrix form as tensors

The skew symmetric matrix form of a 2D point or 2D line corresponds to the (second-rank) tensor resulting from the tensor product of the 2D point or 2D line with a $3 \times 3 \times 3$ permutation tensor $\epsilon_{ijk} = \epsilon^{ijk}$, e.g.,

- Line in image 1: $\ell_r \epsilon^{ris} [\ell]_x$
- Point in image 2: $x^{ij} \epsilon_{jpr} [x']_x$
- Point in image 3: $x'^{mk} \epsilon_{kqs} [x''']_x$

where

$$\epsilon_{ijk} = \epsilon^{ijk} = \begin{cases} 
0 & \text{for } i = j, j = k, \text{ or } i = k \\
1 & \text{for } (i, j, k) \in \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\} \\
-1 & \text{for } (i, j, k) \in \{(1, 3, 2), (3, 2, 1), (2, 1, 3)\}
\end{cases}$$
Tensors

• 2D projective transform as a contraction

A 2D projective transformation as a (second-rank) tensor maps points and lines (first-rank tensors) as a contraction, e.g.,

Map point in image 1 to point in image 2 \( x'^i = x^i h_i^j \quad x' = H_{1,2} x \)
Map point in image 1 to point in image 3 \( x''^k = x^i h_i^k \quad x'' = H_{1,3} x \)
Map line in image 2 to line in image 1 \( \ell_i = \ell'_j h_j^i \quad \ell = H_{1,2}^\top \ell' \)
Map line in image 3 to line in image 1 \( \ell_i = \ell''_k h_k^i \quad \ell = H_{1,3}^\top \ell'' \)
Trifocal tensor

• The trifocal tensor embodies the projective geometry of three views

• Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  – A 2D projective transformation maps a point to a point
  – A 2D projective transformation maps a line to a line
Trifocal tensor

• The trifocal tensor embodies the projective geometry of three views

• Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  – A 2D projective transformation maps a line to a line

The trifocal tensor $\mathcal{T}_{ij}^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

$$\ell_i = \ell'_j \ell''_k \mathcal{T}_{ij}^{jk}$$
$$\ell_i = \ell''_k h^k_i, \text{ where } h^k_i = \ell'_j \mathcal{T}_{jik}^{jk}$$
Trifocal tensor

- The 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view.

\[ \ell_i = \ell''_k h_i^k, \text{ where } h_i^k = \ell'_j T_i^{jk} \]

Induced by a plane backprojected from a line in image 2.
Trifocal tensor

• The trifocal tensor embodies the projective geometry of three views

• Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view
  – A 2D projective transformation maps a point to a point

The trifocal tensor $\mathcal{T}_i^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

$$x''^k = x^i \ell_j \mathcal{T}_i^{jk}$$

$$x''^k = x^i h_i^k, \text{ where } h_i^k = \ell_j \mathcal{T}_i^{jk}$$
Trifocal tensor

- The 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view.

\[ x''^k = x'^i h_i^k, \text{ where } h_i^k = \ell_j^i \mathcal{T}_{ij}^{jk} \]
Trifocal tensor

• The 2D projective transformation induced by a plane backprojected from a line in image 2 maps a point in image 1 on the line in image 2 the same

\[ \ell_i = \ell'_j \ell''_k T_i^{jk} \text{ and } x^i \ell_i = 0 \]

\[ x^i \ell'_j \ell''_k T_i^{jk} = 0 \quad \text{Trilinearity} \]
Trifocal tensor

• The trifocal tensor embodies the projective geometry of three views

• Specifically, the 2D projective transformation between two of the views induced by a plane backprojected from a line in the other view

The trifocal tensor $\mathcal{T}_{i}^{jk}$ is a $3 \times 3 \times 3$ (27 elements, 18 degrees of freedom) tensor with one covariant index and two contravariant indices

\[
x'^{j} = x^{i} \ell_{k}^{j} \mathcal{T}_{i}^{jk} \quad x''^{k} = x^{i} \ell_{j}^{i} \mathcal{T}_{i}^{jk}
\]

\[
x'^{j} = x^{i} h_{i}^{j}, \text{ where } h_{i}^{j} = \ell_{k}^{j} \mathcal{T}_{i}^{jk} \quad x''^{k} = x^{i} h_{i}^{k}, \text{ where } h_{i}^{k} = \ell_{j}^{i} \mathcal{T}_{i}^{jk}
\]

\[
\ell_{i} = \ell_{j}^{i} \ell_{k}^{j} \mathcal{T}_{i}^{jk} \quad \ell_{i} = \ell_{j}^{i} \ell_{k}^{i} \mathcal{T}_{i}^{jk}
\]

\[
\ell_{i} = \ell_{j}^{i} h_{i}^{j}, \text{ where } h_{i}^{j} = \ell_{k}^{j} \mathcal{T}_{i}^{jk} \quad \ell_{i} = \ell_{j}^{i} h_{i}^{k}, \text{ where } h_{i}^{k} = \ell_{j}^{i} \mathcal{T}_{i}^{jk}
\]
Trifocal tensor

• The trilinearities, i.e., incidence relations

Line-line-line correspondence
\[ \ell_r \varepsilon^{ris} \ell_{ij} \ell_{jk} T_{ij}^{jk} = 0^s \]
Point-line-line correspondence
\[ x^i \ell_{ij} \ell_{jk} T_{ij}^{jk} = 0 \]
(Point-line-point correspondence
\[ x^i \ell_{ij} x^{\prime mk} \varepsilon_{kqs} T_{ij}^{jq} = 0_s \]
Point-point-line correspondence
\[ x^i x^{\prime ij} \varepsilon_{jpr} \ell_{jk} T_{ij}^{pk} = 0^r \]
Point-point-point correspondence
\[ x^i x^{\prime ij} x^{\prime mk} \varepsilon_{kqs} T_{ij}^{pq} = 0_{rs} \]

– Remember

Line in image 1 \[ \ell_r \varepsilon^{ris} \]
\[ [\ell]_x \]
Point in image 2 \[ x^i \varepsilon_{jpr} \]
\[ [x^\prime]_x \]
Point in image 3 \[ x^{\prime mk} \varepsilon_{kqs} \]
\[ [x^{\prime\prime}]_x \]

where
\[ \varepsilon_{ijk} = \varepsilon^{ijk} = \begin{cases} 0 & \text{for } i = j, j = k, \text{ or } i = k \\ 1 & \text{for } (i, j, k) \in \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\} \\ -1 & \text{for } (i, j, k) \in \{(1, 3, 2), (3, 2, 1), (2, 1, 3)\} \end{cases} \]
Transfer using the trifocal tensor

• Point-Point-Point
Transfer using the trifocal tensor

- Line-Line-Line
Additional trilinearity constraints

• Point-Line-Point
  – Note: line in image 2 must pass through corresponding point in image 2
Additional trilinearity constraints

- Point-Line-Line
  - Note: lines in images 2 and 3 do not need to correspond, but must pass through corresponding points in images 2 and 3
Trifocal tensor

• Trifocal plane

• Epipoles

The trifocal tensor $\mathcal{T}^{ij}_{i}$ can also be represented as a set of three matrices $\{T_1, T_2, T_3\}$, where

$$T_1 = \begin{bmatrix} \mathcal{T}^{11}_{1} & \mathcal{T}^{12}_{1} & \mathcal{T}^{13}_{1} \\ \mathcal{T}^{21}_{1} & \mathcal{T}^{22}_{1} & \mathcal{T}^{23}_{1} \\ \mathcal{T}^{31}_{1} & \mathcal{T}^{32}_{1} & \mathcal{T}^{33}_{1} \end{bmatrix}, \ T_2 = \begin{bmatrix} \mathcal{T}^{11}_{2} & \mathcal{T}^{12}_{2} & \mathcal{T}^{13}_{2} \\ \mathcal{T}^{21}_{2} & \mathcal{T}^{22}_{2} & \mathcal{T}^{23}_{2} \\ \mathcal{T}^{31}_{2} & \mathcal{T}^{32}_{2} & \mathcal{T}^{33}_{2} \end{bmatrix}, \text{ and } T_3 = \begin{bmatrix} \mathcal{T}^{11}_{3} & \mathcal{T}^{12}_{3} & \mathcal{T}^{13}_{3} \\ \mathcal{T}^{21}_{3} & \mathcal{T}^{22}_{3} & \mathcal{T}^{23}_{3} \\ \mathcal{T}^{31}_{3} & \mathcal{T}^{32}_{3} & \mathcal{T}^{33}_{3} \end{bmatrix}$$

Let $u_i$ and $v_i$ be the left and right null vectors, respectively, of $T_i$ (i.e., $u_i^T T_i = 0^T$ and $T_i v_i = 0$). The epipoles $e_{21}$ and $e_{31}$ are calculated as follows.

$$\begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix} e_{21} = 0$$

$$\begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix} e_{31} = 0$$
Trifocal tensor

The trifocal tensor $\mathcal{T}_{i}^{jk}$ from the canonical cameras $P = [I | 0]$, $P' = B$, and $P'' = C$

$$
\mathcal{T}_{i}^{jk} = b_i^j c_i^k - b_i^j c_i^k
$$

Given general cameras

$$
P = \begin{bmatrix}
a_1^{T} \\
a_2^{T} \\
a_3^{T}
\end{bmatrix}, P' = \begin{bmatrix}
b_1^{T} \\
b_2^{T} \\
b_3^{T}
\end{bmatrix}, \text{ and } P'' = \begin{bmatrix}
c_1^{T} \\
c_2^{T} \\
c_3^{T}
\end{bmatrix}
$$

where $a_i^{T}$, $b_j^{T}$, and $c_k^{T}$ are the $i$th, $j$th, and $k$th rows of $P$, $P'$, and $P''$, respectively, the trifocal tensor $\mathcal{T}_{i}^{jk}$ is given by

$$
\mathcal{T}_{i}^{jk} = (-1)^{i+1} \det \left( \begin{bmatrix}
\sim{a}_i^{T} \\
{b}_j^{T} \\
{c}_k^{T}
\end{bmatrix} \right)
$$

where $\sim{a}_i^{T}$ is the matrix $P$ with the $i$th row omitted
Trifocal tensor

• Error in image point and line measurements
  – Transfer and trilinearity constraints
  – Measured points and lines, and projected points and lines
Estimation of trifocal tensor

\[ x_i = P X_i \forall i, \quad x'_i = P' X_i \forall i, \quad \text{and} \quad x''_i = P'' X_i \forall i, \]
set \( P = [I \mid 0] \) and solve for \( P' \) and \( P'' \)

- Given point correspondences \( x_i \leftrightarrow x'_i \leftrightarrow x''_i \)
- Minimize geometric error \( \sum_i \left( d(x_i, PX_i)^2 + d(x'_i, P'X_i)^2 + d(x''_i, P''X_i)^2 \right) \)
- 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
Outlier rejection

• Even the presence of a single outlier may result in an inaccurate estimate

• Linear estimate
  – Minimizes sum of squared error and, in general, outliers have large error relative to inliers
  – Resulting estimate may not be near global minimum

• Nonlinear estimate
  – Cost function minimizes sum of squared error, so may not converge to accurate solution
    • Alternatively, use a robust cost function
Trifocal tensor, estimation from minimum number of correspondences

- 6 image point correspondences between three images
  - The last 4 image points must be in general position
    - 6 choose 4 = 15 combinations (rearrange, if needed)
- Carlsson-Weinshall duality (interchange camera and points)
  - Dualize first 2 image points
    - Results in 3 dual point correspondences
    - Additional 4 dual point correspondences are 2D projective basis
  - Calculate reduced dual fundamental matrix (5 degrees of freedom) from 7 dual point correspondences
    - 1 or 3 solutions
- Solve for three camera projection matrices
  - 5 3D points are 3D projective basis
  - 1 or 3 solutions for 6th 3D point using dual fundamental matrix
- Calculate trifocal tensor from three camera projection matrices
  - 1 or 3 solutions

See textbook for details
Random sample consensus (RANSAC) and M-estimator sample consensus (MSAC)

Objective is to determine the consensus set with the minimum cost

tol is the tolerance for establishing datum/model compatibility
\( \tau_{\text{cost}} \) is the upper bound on the cost of an acceptable consensus set
maxTrials is the maximum number of attempts to find a consensus set

\[ \text{consensus}_{\text{cost}} = \infty \]

for (trials = 0; trials < maxTrials && consensus_{cost} > \tau_{\text{cost}}; ++\text{trials})

Select a random sample of unique data points
Calculate the model using the random sample
Calculate the error for each data point using model
Calculate the cost
if cost < consensus_{cost}
    consensus_{cost} = cost
    consensus_{\text{model}} = model
Calculate the error for each data point using consensus_{\text{model}}
Calculate the set of inliers (i.e., data points with error \( \leq \) tol)

If multiple solutions, try each one, but it is still a single trial
Outlier rejection

- **RANSAC cost**
  count is the total number of data points
  tol is the tolerance for establishing datum/model compatibility

  ```
  cost = 0
  for (n = 0; n < count; ++n)
      cost += error[n] ≤ tol ? 0 : 1
  ```

- **MSAC cost**
  count is the total number of data points
  tol is the tolerance for establishing datum/model compatibility

  ```
  cost = 0
  for (n = 0; n < count; ++n)
      cost += error[n] ≤ tol ? error[n] : tol
  ```
RANSAC and MSAC

• Maximum number of trials
  – Adaptive maximum number of trials
• The tolerance for establishing datum/model compatibility
RANSAC and MSAC

• Maximum number of trials

The smaller the sample size, the smaller maximum number of trials. Use minimum solution!

<table>
<thead>
<tr>
<th>Sample size</th>
<th>Proportion of outliers $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
</tr>
<tr>
<td>$s$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.3. *The number $N$ of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, $s$, and proportion of outliers, $\epsilon$.***
Adaptive maximum number of trials

$s$ is the sample size

$p$ is the assumed probability that at least one of the random samples does not contain any outliers

$\text{tol}$ is the tolerance for establishing datum/model compatibility

$\tau_{\text{cost}}$ is the upper bound on the cost of an acceptable consensus set

$maxTrials = \infty$

$\text{consensus}_{\text{cost}} = \infty$

for (trials = 0; trials < maxTrials && consensus_{cost} > \tau_{\text{cost}}; ++trials)

Select a random sample of unique data points

Calculate the model using the random sample

Calculate the error for each data point using model

Calculate the cost

if cost < consensus_{cost}

$\text{consensus}_{\text{cost}} = \text{cost}$

$\text{consensus}_{\text{model}} = \text{model}$

Calculate the number of inliers

$w = \frac{\text{number of inliers}}{\text{total number of data points}}$

$maxTrials = \frac{\log(1 - p)}{\log(1 - w^s)}$

Calculate the error for each data point using consensus_{model}

Calculate the set of inliers (i.e., data points with error $\leq \text{tol}$)

If multiple solutions, try each one, but it is still a single trial
RANSAC and MSAC

• The tolerance for establishing datum/model compatibility
  
  Probability $\alpha$ that a data point is an inlier (usually $\alpha$ is chosen to be 0.95)
  Variance $\sigma^2$ of the measurement error (if unknown, assumed to be 1)
  Codimension $m$
  Tolerance is square distance $t^2 = F_m^{-1}(\alpha)\sigma^2$, where $F_m^{-1}(\alpha)$ is the inverse chi-squared cumulative distribution function with $m$ degrees of freedom at the probability $\alpha$

• Codimension
  
  If $w$ is a subspace of a finite-dimensional vector space $v$, then $\text{codim}(w) = \text{dim}(v) - \text{dim}(w)$
  The trifocal tensor is a variety of dimension 3 in $(\tilde{x}, \tilde{y}, \tilde{x}', \tilde{y}', \tilde{x}'', \tilde{y}'') \in \mathbb{R}^6$, so has codimension 3

Use the square Sampson error See textbook for details
Feature detection and matching

Input Images
Feature detection and matching

Detected Corners
Feature detection and matching

Simple Matching

Expect some false matches
Feature detection and matching

Simple Matching
Including Outlier Rejection

No false matches
Linear estimation of trifocal tensor using the direct linear transformation (DLT) algorithm

- Minimize algebraic error

\[ x^i \ell'_j \ell''_k \mathcal{T}_i^{jk} = 0 \quad \text{Trilinearity} \]

\[ (\mathbf{x}^\top \otimes (\ell'^\top \otimes \ell''^\top)) \mathbf{t} = 0 \]

where \( \otimes \) denotes the Kronecker product and

\[ \mathbf{t} = (\mathcal{T}_1^{11}, \mathcal{T}_1^{12}, \mathcal{T}_1^{13}, \mathcal{T}_1^{21}, \mathcal{T}_1^{22}, \mathcal{T}_1^{31}, \mathcal{T}_1^{32}, \mathcal{T}_1^{33}, \mathcal{T}_2^{11}, \mathcal{T}_2^{12}, \mathcal{T}_2^{13}, \mathcal{T}_2^{21}, \mathcal{T}_2^{22}, \mathcal{T}_2^{23}, \mathcal{T}_2^{31}, \mathcal{T}_2^{32}, \mathcal{T}_2^{33}, \mathcal{T}_3^{11}, \mathcal{T}_3^{12}, \mathcal{T}_3^{13}, \mathcal{T}_3^{21}, \mathcal{T}_3^{22}, \mathcal{T}_3^{23}, \mathcal{T}_3^{31}, \mathcal{T}_3^{32}, \mathcal{T}_3^{33})^\top \]
Linear estimation of trifocal tensor using the direct linear transformation (DLT) algorithm

Given $n \geq 7$ point correspondences $x_i \leftrightarrow x'_i \leftrightarrow x''_i$

\[
(x_i^\top \otimes ([x'_i]^\perp \otimes [x''_i]^\perp))t = 0 \forall i, \text{ solve for } t, \text{ where }
\]

\[
t = (T_{11}^{11}, T_{11}^{12}, T_{11}^{13}, T_{11}^{21}, T_{11}^{22}, T_{11}^{23}, T_{11}^{31}, T_{11}^{32}, T_{11}^{33}, T_{12}^{11}, T_{12}^{12}, T_{12}^{13}, T_{12}^{21}, T_{12}^{22}, T_{12}^{23}, T_{23}^{23}, T_{23}^{31}, T_{23}^{32}, T_{23}^{33}, T_{31}^{11}, T_{31}^{12}, T_{31}^{13}, T_{31}^{21}, T_{31}^{22}, T_{31}^{23}, T_{31}^{31}, T_{31}^{32}, T_{31}^{33})^\top
\]

\[
\begin{bmatrix}
  x_1^\top \otimes ([x'_1]^\perp \otimes [x''_1]^\perp) \\
x_2^\top \otimes ([x'_2]^\perp \otimes [x''_2]^\perp) \\
\vdots \\
x_n^\top \otimes ([x'_n]^\perp \otimes [x''_n]^\perp)
\end{bmatrix} t = 0
\]

$4n \times 27$

\[At = 0, \text{ where } A = \begin{bmatrix}
  x_1^\top \otimes ([x'_1]^\perp \otimes [x''_1]^\perp) \\
x_2^\top \otimes ([x'_2]^\perp \otimes [x''_2]^\perp) \\
\vdots \\
x_n^\top \otimes ([x'_n]^\perp \otimes [x''_n]^\perp)
\end{bmatrix}\]

Solve for $t$ using a constrained minimization, subject to a span-space constraint. Minimize $\|At\|$ subject to $\|t\| = 1$ and $t = G\hat{t}$, where $G$ has rank $r \leq 18$.

See textbook for details

But, data normalization must be used
Retrieve camera projection matrices from trifocal tensor

Retrieve epipoles \( \mathbf{e}_{21} \) and \( \mathbf{e}_{31} \), and fundamental matrix \( \mathbf{F}_{1,2} \) from the trifocal tensor. Then, retrieve the camera projection matrices \( \mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \) and \( \mathbf{P}' = [\mathbf{a}_1 \mid \mathbf{a}_2 \mid \mathbf{a}_3 \mid \mathbf{a}_4] \), where \( [\mathbf{a}_1 \mid \mathbf{a}_2 \mid \mathbf{a}_3] \) is full rank, from \( \mathbf{F}_{1,2} \).

Two possible choices of third camera projection matrix \( \mathbf{P}'' \)

\[
\mathbf{P}_1'' = [\mathbf{b}_1 \mid \mathbf{b}_2 \mid \mathbf{b}_3 \mid \mathbf{b}_4], \quad \text{where} \quad \mathbf{b}_{\{1,2,3\}} = (\mathbf{e}_{31} \mathbf{a}_{\{1,2,3\}}^\top - \mathbf{T}_{\{1,2,3\}}^\top) \mathbf{a}_4 \quad \text{and} \quad \mathbf{b}_4 = \mathbf{e}_{31}
\]

or \( \mathbf{P}_2'' = [\mathbf{c}_1 \mid \mathbf{c}_2 \mid \mathbf{c}_3 \mid \mathbf{c}_4] \), where \( \mathbf{c}_{\{1,2,3\}} = (-\mathbf{e}_{31} \mathbf{a}_{\{1,2,3\}}^\top - \mathbf{T}_{\{1,2,3\}}^\top) \mathbf{a}_4 \quad \text{and} \quad \mathbf{c}_4 = -\mathbf{e}_{31} \)

Choose the \( \mathbf{P}'' \) compatible with the trifocal tensor.
Up to 3D projective transformation
Nonlinear estimation using the Levenberg-Marquardt algorithm

Given

Measurement vector $\mathbf{X}$ with associated covariance matrix $\Sigma_X$
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 $\mathbf{e}^T \Sigma_X^{-1} \mathbf{e}$, where $\mathbf{e} = \mathbf{X} - \hat{\mathbf{X}}$
Parameter vector

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

• The trifocal tensor has 18 degrees of freedom, so can be parameterized with 18 parameters
  – Advanced topic (come to office hours)

• Alternatively, fix the first camera projection matrix and adjust the second and third ones
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

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

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

\[
\mathbf{\tilde{v}} = \left( \cos \left( \frac{\|\mathbf{v}\|}{2} \right), \frac{\text{sinc} \left( \frac{\|\mathbf{v}\|}{2} \right)}{2} \mathbf{v}^\top \right)^\top \in \mathbb{R}^n
\]

Unit vector with nonnegative first element

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

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

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

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

\[
\frac{\partial \mathbf{b}}{\partial \mathbf{v}} = \begin{cases} \frac{1}{2} \mathbb{I} & \text{if } \|\mathbf{v}\| = 0 \\
\frac{\text{sinc} \left( \frac{\|\mathbf{v}\|}{2} \right)}{2} \mathbb{I} + \frac{1}{4\|\mathbf{v}\|} \frac{d\text{sinc} \left( \frac{\|\mathbf{v}\|}{2} \right)}{d\|\mathbf{v}\|} \mathbf{v} \mathbf{v}^\top & \text{otherwise} \end{cases}
\]
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

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

Parameter vector \((22 + 3n) \times 1\)

\[
\begin{pmatrix}
\hat{p'}^T, \hat{p''}^T, \hat{X}_1^T, \hat{X}_2^T, \ldots, \hat{X}_n^T
\end{pmatrix}^T
\]

Parameterization of homogeneous vector

Mapping

\[
\hat{x}_i = P\hat{X}_i \forall i
\]

\[
\hat{x}'_i = \hat{p}'\hat{X}_i \forall i
\]

\[
\hat{x}''_i = \hat{p}''\hat{X}_i \forall i
\]

Measurement vector \(6n \times 1\)

\[
\begin{pmatrix}
\hat{x}^T_1, \hat{x}^T_2, \ldots, \hat{x}^T_n, \hat{x}'^T_1, \hat{x}'^T_2, \ldots, \hat{x}'^T_n, \hat{x}''^T_1, \hat{x}''^T_2, \ldots, \hat{x}''^T_n
\end{pmatrix}^T
\]

with associated covariance matrices \(\Sigma_{\hat{x}_1}, \Sigma_{\hat{x}_2}, \ldots, \Sigma_{\hat{x}_n}, \Sigma_{\hat{x}'_1}, \Sigma_{\hat{x}'_2}, \ldots, \Sigma_{\hat{x}'_n}, \Sigma_{\hat{x}''_1}, \Sigma_{\hat{x}''_2}, \ldots, \Sigma_{\hat{x}''_n}\)

Cost

\[
\sum_i \left( \epsilon_i^T \Sigma_{\hat{x}_i}^{-1} \epsilon_i + \epsilon'_i^T \Sigma_{\hat{x}'_i}^{-1} \epsilon'_i + \epsilon''_i^T \Sigma_{\hat{x}''_i}^{-1} \epsilon''_i \right), \text{ where } \epsilon_i = \hat{x}_i - \hat{x}_i, \epsilon'_i = \hat{x}'_i - \hat{x}'_i, \text{ and } \epsilon''_i = \hat{x}''_i - \hat{x}''_i
\]

But, use data normalization
Jacobian
Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

• Jacobian

\[
\begin{align*}
A'_i &= \frac{\partial \hat{x}'_i}{\partial \hat{p}'} = \frac{\partial \hat{x}'_i}{\partial \hat{p}'} \frac{\partial \hat{p}'}{\partial \hat{p}'} \\
A''_i &= \frac{\partial \hat{x}''_i}{\partial \hat{p}''} = \frac{\partial \hat{x}''_i}{\partial \hat{p}''} \frac{\partial \hat{p}''}{\partial \hat{p}'} \\
B_i &= \frac{\partial \hat{x}_i}{\partial \hat{X}_i} = \frac{\partial \hat{x}_i}{\partial \hat{X}_i} \frac{\partial \hat{X}_i}{\partial \hat{x}_i} \\
B'_i &= \frac{\partial \hat{x}'_i}{\partial \hat{X}_i} = \frac{\partial \hat{x}'_i}{\partial \hat{X}_i} \frac{\partial \hat{X}_i}{\partial \hat{x}_i} \\
B''_i &= \frac{\partial \hat{x}''_i}{\partial \hat{X}_i} = \frac{\partial \hat{x}''_i}{\partial \hat{X}_i} \frac{\partial \hat{X}_i}{\partial \hat{x}_i}
\end{align*}
\]
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

$$\mathbf{x} = \mathbf{PX}$$

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

$$\frac{\partial \tilde{x}}{\partial \mathbf{p}} = \frac{1}{w} \begin{bmatrix} \mathbf{X}^\top & 0^\top & -\tilde{x} \mathbf{X}^\top \\ 0^\top & \mathbf{X}^\top & -\tilde{y} \mathbf{X}^\top \end{bmatrix} \quad 2 \times 12$$

and

$$\frac{\partial \tilde{x}}{\partial \mathbf{X}} = \frac{1}{w} \begin{bmatrix} \mathbf{p}^{1\top} - \tilde{x} \mathbf{p}^{3\top} \\ \mathbf{p}^{2\top} - \tilde{y} \mathbf{p}^{3\top} \end{bmatrix} \quad 2 \times 4$$

where $w = \mathbf{p}^{3\top} \mathbf{X}$ and $\mathbf{p}^{i\top}$ is the $i$th row of $\mathbf{P}$. 
Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

- Normal equations matrix

\[ U' = \sum_i A_i'^T \Sigma_{\hat{x}_i}^{-1} A_i' \]  
\[ 11 \times 11 \]

\[ U'' = \sum_i A_i''^T \Sigma_{\hat{x}_i}^{-1} A_i'' \]  
\[ 11 \times 11 \]

\[ V_i = \sum_{j=1}^{3} B_i^{(j)}^T \Sigma_{\hat{x}_i}^{-1} B_i^{(j)} \]  
\[ 3 \times 3 \]

\[ W'_i = A_i'^T \Sigma_{\hat{x}_i}^{-1} B'_i \]  
\[ 11 \times 3 \]

\[ W''_i = A_i''^T \Sigma_{\hat{x}_i}^{-1} B''_i \]  
\[ 11 \times 3 \]
Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

• Normal equations vector

\[
\begin{pmatrix}
\varepsilon_{a'}^T, \varepsilon_{a''}^T, \varepsilon_{b_1}^T, \varepsilon_{b_2}^T, \ldots, \varepsilon_{b_n}^T
\end{pmatrix}^T
\]

where

\[
\varepsilon_{a'} = \sum_i A_i'^T \Sigma_{\hat{x}_i}^{-1} \varepsilon_i'
\quad 11 \times 1
\]
\[
\varepsilon_{a''} = \sum_i A_i''^T \Sigma_{\hat{x}_i}^{-1} \varepsilon_i''
\quad 11 \times 1
\]
\[
\varepsilon_{b_i} = \sum_{j=1}^3 B_i^{(j)}^T \Sigma_{\hat{x}_i^{(j)}}^{-1} \varepsilon_i^{(j)}
\quad 3 \times 1
\]

where \( \varepsilon_i = \hat{x}_i - \hat{x}_i' \), \( \varepsilon_i' = \hat{x}_i' - \hat{x}_i'' \), and \( \varepsilon_i'' = \hat{x}_i'' - \hat{x}_i''' \)
Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

Augmented normal equations

\[ S = \begin{pmatrix}
\zeta^{(2,2)} & \zeta^{(2,3)} \\
\zeta^{(3,2)} & \zeta^{(3,3)}
\end{pmatrix} \]

where

\[
S^{(2,2)} = U'^* - \sum_i W'_i V_i^{*^{-1}} W'_i^T \\
S^{(3,3)} = U'''^* - \sum_i W''_i V_i^{*^{-1}} W''_i^T \\
S^{(2,3)} = -\sum_i W'_i V_i^{*^{-1}} W''_i^T \\
S^{(3,2)} = S^{(2,3)^T} \\
\]

\[ e = (e'^T, e''^T)^T \]

\[
e' = \epsilon_{a'} - \sum_i W'_i V_i^{*^{-1}} \epsilon_{b_i} \]

\[
e'' = \epsilon_{a''} - \sum_i W''_i V_i^{*^{-1}} \epsilon_{b_i} \]

\[ S\delta_a = e, \text{ solve for } \delta_a = (\delta_{a'}^T, \delta_{a''}^T)^T \]

\[ \delta_{b_i} = V_i^{*^{-1}} (\epsilon_{b_i} - W'_i^T \delta_{a'} - W''_i^T \delta_{a''}) \]

where \( U'^* = U' + \lambda I, U'''^* = U'' + \lambda I \), and \( V_i^* = V_i + \lambda I \)
Candidate parameter vector

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

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

\[
\hat{X}_{i_0} = \hat{X}_i + \delta_{b_i}
\]
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.

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

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

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

If unknown, then assume to be identity

Covariance propagation

$$\Sigma_{\tilde{\mathbf{x}}_{DN}} = J\Sigma_{\tilde{\mathbf{x}}}J^\top, \text{ where } J = \frac{\partial\tilde{\mathbf{x}}_{DN}}{\partial\tilde{\mathbf{x}}} = \mathbf{A} = s\mathbf{I}$$
Nonlinear estimation of trifocal tensor using the Levenberg-Marquardt algorithm

1. Initialize the scene points using triangulation  
   Covered next lecture

2. Data normalize the image points

\[
\tilde{x}_{DN,i} = A\tilde{x}_i + t \quad \text{and associated covariance matrices} \quad \Sigma_{\tilde{x}_{DN,i}} = s^2 \Sigma_{\tilde{x}_i}
\]

\[
\tilde{x}'_{DN,i} = A'\tilde{x}'_i + t' \quad \text{and associated covariance matrices} \quad \Sigma_{\tilde{x}'_{DN,i}} = s'^2 \Sigma_{\tilde{x}'_i}
\]

\[
\tilde{x}''_{DN,i} = A''\tilde{x}''_i + t'' \quad \text{and associated covariance matrices} \quad \Sigma_{\tilde{x}''_{DN,i}} = s''^2 \Sigma_{\tilde{x}''_i}
\]

scene points

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

and camera projection matrices

\[
P_{DN} = T[I | 0]U^{-1}
\]

\[
P'_{DN} = T'P'U^{-1}
\]

\[
P''_{DN} = T''P''U^{-1}
\]

3. Estimate the data normalized camera projection matrices \( P'_{DN} \) and \( P''_{DN} \), and data normalized scene points from the data normalized image points

4. Data denormalize the final estimate of the data normalized camera projection matrices

\[
P' = T'^{-1}P'_{DN}U
\]

\[
P'' = T''^{-1}P''_{DN}U
\]
Up to 3D projective transformation
Next lecture

• $n$ view geometry
• Reading
  – Sections 12.1, 12.2, 12.3, and 18.1