Multiple View Geometry in Computer Vision

Prasanna Sahoo
Department of Mathematics
University of Louisville
Computation of the Trifocal Tensor

Lecture 23

April 12, 2005
In our last lecture we examined

- how to compute epipoles $e'$ and $e''$ from the $T_i$ matrices,
- how to extract fundamental matrices $F_{12}$ and $F_{13}$ from the trifocal tensor $[T_1, T_2, T_3]$, and
- how to retrieve camera matrices $P'$ and $P''$. 
We also learned tensor notations and reexamined various incidence relations using tensor notations:

(1) Line-line-line correspondence

\[(l_r \varepsilon^{ris}) \ l'_j \ l''_k \ T^j_i \ = \ 0^s\]

(i2) Point-line-line correspondence

\[x^i \ l'_j \ l''_k \ T^j_i \ = \ 0\]
(3) Point-line-point correspondence

\[ x^i l'_j (x''^k \varepsilon_{kqs}) T^{jq}_i = 0_s \]

(4) Point-point-line correspondence

\[ x^i (x'^j \varepsilon_{jpr}) l''_k T^{pk}_i = 0_r \]

(5) Point-point-point correspondence

\[ x^i (x'^j \varepsilon_{jpr}) (x''^k \varepsilon_{kqs}) T^{pq}_i = 0_{rs} \]
In this lecture we will discuss various algorithms for computing the **trifocal tensor** given point-point-point correspondences in three images.
Linear Equations for the Trifocal Tensor

Given a 3-point correspondence

\[ x \leftrightarrow x' \leftrightarrow x'' \]

the trifocal tensor relationship is

\[ x^i x'^j x''^k \epsilon_{jpr} \epsilon_{kqs} T^{pq}_{i} = 0_{rs} \]  \hspace{1cm} (1)

for \( i, j, k, p, q, r, s = 1, 2, 3 \).
• If we substitute $x = (x, y, 1)^T$, $x' = (x', y', 1)^T$ and $x'' = (x'', y'', 1)^T$, then (1) yields 9 different equations.

• However only **four** of these equations are linearly independent.

• For every given point correspondence $x \leftrightarrow x' \leftrightarrow x''$ we get 4 linearly independent equations.

• These four equations are:
\begin{align*}
x & \left( x' x'' T_{1}^{33} - x'' T_{1}^{13} - x' T_{1}^{31} + T_{1}^{11} \right) \\
  & \quad + y \left( x' x'' T_{2}^{33} - x'' T_{2}^{13} - x' T_{2}^{31} + T_{2}^{11} \right) \\
  & \quad + \left( x' x'' T_{3}^{33} - x'' T_{3}^{13} - x' T_{3}^{31} + T_{3}^{11} \right) = 0.
\end{align*}

\begin{align*}
x & \left( x' y'' T_{1}^{33} - y'' T_{1}^{13} - x' T_{1}^{32} + T_{1}^{12} \right) \\
  & \quad + y \left( x' y'' T_{2}^{33} - y'' T_{2}^{13} - x' T_{2}^{32} + T_{2}^{12} \right) \\
  & \quad + \left( x' y'' T_{3}^{33} - y'' T_{3}^{13} - x' T_{3}^{32} + T_{3}^{12} \right) = 0.
\end{align*}
\[ x \left( y' x'' T_{1}^{33} - x'' T_{1}^{23} - y' T_{1}^{31} + T_{1}^{21} \right) \]
\[ + y \left( y' x'' T_{2}^{33} - x'' T_{2}^{23} - y' T_{2}^{31} + T_{2}^{21} \right) \]
\[ + (y' x'' T_{3}^{33} - x'' T_{3}^{23} - y' T_{3}^{31} + T_{3}^{21}) = 0. \]

\[ x \left( y' y'' T_{1}^{33} - y'' T_{1}^{23} - y' T_{1}^{32} + T_{1}^{22} \right) \]
\[ + y \left( y' y'' T_{2}^{33} - y'' T_{2}^{23} - y' T_{2}^{32} + T_{2}^{22} \right) \]
\[ + (y' y'' T_{3}^{33} - y'' T_{3}^{23} - y' T_{3}^{32} + T_{3}^{22}) = 0. \]
Hence the relationship

\[ x^i x'^j x''^k \varepsilon_{jpr} \varepsilon_{kqs} T^p_i T^q_j = 0_{rs} \]

- is linear in the entries of \( \mathcal{T} \).
- Each point correspondence gives 4 equations.
- \( \mathcal{T} \) has 27 entries - defined up to scale.
- 7 point correspondence gives 28 equations.
If we have \( n \geq 7 \) correspondences, then the total set of equations can be written in the form

\[ \textbf{A} \textbf{t} = 0 \]  

(2)

where \( \textbf{t} \) is a 27-vector made out of the entries from the trifocal tensor \( \mathcal{T} \), and \( \textbf{A} \) is a \( 4n \times 27 \) matrix.
Other point or line correspondences also yield constraint on $\mathcal{T}$, and they can be used to form $A t = 0$.

<table>
<thead>
<tr>
<th>Correspondence</th>
<th>Relation</th>
<th># of equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>three points</td>
<td>$x^i x'^j x''^k \varepsilon_{jpr} \varepsilon_{kqs} T_{i}^{pq} = 0_{rs}$</td>
<td>4</td>
</tr>
<tr>
<td>two points, one line</td>
<td>$x^i (x'^j \varepsilon_{jpr}) l''<em>k T</em>{i}^{pk} = 0_r$</td>
<td>2</td>
</tr>
<tr>
<td>one point, two lines</td>
<td>$x^i l'_j l''<em>k T</em>{i}^{jk} = 0$</td>
<td>1</td>
</tr>
<tr>
<td>three lines</td>
<td>$(l_r \varepsilon^{ris}) l'_j l''<em>k T</em>{i}^{jk} = 0^s$</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1. Summary of relations
Solving the equations

Given 26 equations we can solve for the 27 entries of the trifocal tensor $T$.

- Need 7 point correspondences
- or 13 line correspondences
- or some mixture.
Consider the matrix equation

\[ A \mathbf{t} = 0 \]

where \( A \) is a \( n \times 27 \) matrix.

- With 26 equations we can find an exact solution.
- With more than 26 equations, we look for least-squares solutions.
Solution:

- Take the SVD: \( A = U D V^T \).

- Solution is in the last column of \( V \).

- Minimizes \( ||A t|| \) subject to \( ||t|| = 1 \).
  (use solution from SVD as initial approximation)

Normalization of data is essential.
Linear least-square algorithm

- Trifocal tensor $\mathcal{T}$ can be determined from 7 or more point correspondences.

- **Procedure:**
  1. Normalize image points.
  2. Determine $\hat{\mathcal{T}}$ for normalized points.
  3. Denormalize: determine $\mathcal{T}$ from $\hat{\mathcal{T}}$. 
1. Data Normalization

- We need to translate our points so their centroid is at the origin.

- Then we want to scale them so their RMS distance from the origin is $\sqrt{2}$.

- Construct $H$ from a translation and a scaling component: $H = T_{scale} T_{trans}$. 
• The translation component is

\[
T_{\text{trans}} = \begin{pmatrix}
1 & 0 & -\bar{x} \\
0 & 1 & -\bar{y} \\
0 & 0 & 1
\end{pmatrix}
\]

where \( \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \) and \( \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \).
• The scaling component is

\[
T_{\text{scale}} = \begin{pmatrix}
\frac{\sqrt{2}}{\sigma} & 0 & 0 \\
0 & \sqrt{2} & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

where \( \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} [(x_i - \bar{x})^2 + (y_i - \bar{y})^2]} \).
Normalized image coordinates are obtained using

\[ \hat{x} = H \ x, \]
\[ \hat{x}' = H' \ x' \]
\[ \hat{x}'' = H'' \ x'' \]

where \( x \) is the image point in the first view, \( x' \) is the image point in the second view, and \( x'' \) is the image point in the third view.
2. Determining $\hat{T}$ from normalized coordinates

★ For a 3-point correspondence $x \leftrightarrow x' \leftrightarrow x''$ of the normalized points the trifocal tensor relationship is

$$x^i x'^j x''^k \varepsilon_{jpr} \varepsilon_{kqs} T_{pq}^{i} = 0_{rs}$$

for $i, j, k, p, q, r, s = 1, 2, 3$.

★ From $n$ such 3-point correspondences, one can have

$$A t = 0,$$

where $A$ is a $4n \times 27$ matrix, and $t$ is a 27-vector.
Solution:

- If there are only 7 point correspondences
  
  - Take the SVD: \( A = U D V^T \).
  
  - Solution is in the last column of \( V \).
If there are more than 7 point correspondences, we try to find a least squares solution.

For this we iteratively minimize

\[ \|A t\| \quad \text{subject to} \quad \|t\| = 1 \]

using ML algorithm.

The solution from SVD is used as an initial approximation to start the iteration.
3. Denormalization

Must perform denormalization on the computed $\widehat{T}$ to work with the original image coordinates:

$$T_i^{jk} = H_i^r \left( H'^{-1} \right)_s^j \left( H''^{-1} \right)_t^k \widehat{T}_{rst}.$$
Drawbacks of least-square algorithm

This linear least-square algorithm is a fast and easy way to compute the trifocal tensor, it has two major drawbacks.

• The first drawback is that it does not take into account the 8 intrinsic constraints of the trifocal tensor.
• The second drawback is that it does not minimize the errors in the original measurements.
The iterative algorithms such as

- Algebraic minimization algorithm
- Geometric distance minimization
  - Gold standard (MLE)
  - Sampson distance
  - Mahalanobis distance
- RANSAC method

take into account the 8 intrinsic constraints and also minimize the measurement errors.
Constraints

- $\mathcal{T}$ has 27 entries, defined up to scale.
- Geometry only has 18 degrees of freedom.
  - 3 camera matrices accounts for $3 \times 11 = 33$ dof.
  - Invariant under projective transformation (15 dof).
  - Total $33 - 15 = 18$ dof.
  - Number of intrinsic constraints: $26 - 18 = 8$.
- $\mathcal{T}$ must satisfy some of these intrinsic constraints to be a valid trifocal tensor.
What are the constraints

Some of the constraints are easy to find.
(i) Each $T_{jk}$ must have rank 2.
(ii) Their null spaces must lie in a plane.
(iii) This gives 4 constraints in all.
(iv) 4 other constraints are not so easily formulated.
Constraints through parametrization

- Define $\mathcal{T}$ in terms of a set of parameters.
- Only valid $\mathcal{T}$’s may be generated from parameters.
  Recall formula for $\mathcal{T}$:
  \[ T^{jk}_i = a^j_i b^k_4 - a^4_i b^k_i. \]
- Valid trifocal tensors are generated by this formula.
- Parameters are in the entries $a^j_i$ and $b^k_i$.
- Over-parametrized: 24 parameters in all.
Algebraic minimization algorithm

Objective

Given a set of point and line correspondences in three views, compute the trifocal tensor.

Algorithm

(1) From the set of point and line correspondences compute the set of equations of the form $A_t \neq 0$. 
(2) Solve $At = 0$ using least-square method to find an initial estimate of the trifocal tensor $T_{i}^{jk}$.

(3) Find the two epipoles $e'$ and $e''$ from $T_{i}^{jk}$. For this first find the left null-vector $u_{i}$ of the $3 \times 3$ matrix $T_{i}$. The epipole $e'$ is the left null-vector of the $3 \times 3$ matrix $[u_{1}, u_{2}, u_{3}]$. Similarly, if $v_{i}$ is the right null-space of $3 \times 3$ matrix $T_{i}$, then the epipole $e''$ right null-space of $3 \times 3$ matrix $[u_{1}, u_{2}, u_{3}]$. 

32
(4) Construct the $27 \times 18$ matrix $E$ such that $t = Eg$

where $t$ is the vector entries of $T_{i}^{jk}$, $g$ is the vector representing entries of $a_{i}^{j}$ and $b_{i}^{k}$. Matrix $E$ expresses the linear relationship $T_{i}^{jk} = a_{i}^{j}e^{\mu k} - e^{ij}b_{i}^{k}$.

(5) Solve the minimization problem: minimize $\|AEg\|$ subject to $\|Eg\| = 1$ using LM algorithm.

Compute the error vector $\epsilon = AEg$. 
(6) **Iteration:** The mapping \((e', e'') \mapsto \epsilon\) maps from \(\mathbb{R}^6\) to \(\mathbb{R}^{27}\). Iterate on the last two steps with varying \(e'\) and \(e''\) using LM algorithm to find the optimal \(e'\) and \(e''\). Hence find the optimal \(t = Eg\) containing the entries of \(T_{ij}^{jk}\).
Please read

• Geometric distance minimization method, and

• RANSAC method

from the text book.
Class Presentations

• Class presentation starts from April 19 and ends on April 28.

• You must attend all presentations or you will loose 20% of your earned points on the project.

• You should participate in the question and answer session after the presentation.
END