Two-view geometry

Tomáš Svoboda, svoboda@cmp.felk.cvut.cz
Czech Technical University in Prague, Center for Machine Perception
http://cmp.felk.cvut.cz
Last update: October 29, 2007

Talk Outline

- Epipolar geometry
- Estimation of the Fundamental matrix
- Camera motion
- Reconstruction of scene structure
Two projections of a rigid 3D scene

- The projections are clearly different.
- Can the difference tell something about the camera positions?
- and about the scene structure?
The projections are clearly different.

Can the difference tell something about the camera positions?

and about the scene structure?

It can! (to both)
Can we find a relation between corresponding projections regardless of the scene structure?
Back project the ray

\[ X_9 = \lambda P^1 + u_9^1 + C^1 \]
Project the camera center to the second image

\[ X_9 = \lambda P^1 + u_9 + C^1 \]

\[ e^2 = P^2 C^1 \]
The corresponding projection must lie on a specific line

\[ X_9 = \lambda P^1 + u_9 + C^1 \]
Derivation of the **Fundamental matrix**

We already know: \( \mathbf{e}^2 = \mathbf{P}^2\mathbf{C}^1 \)

Projection to the camera 2: \( \mathbf{u}_9^2 = \mathbf{P}^2(\lambda \mathbf{P}^1 + \mathbf{u}_9^1 + \mathbf{C}^1) \)

Line is a cross product of the points lying on it: \( \mathbf{e}^2 \times \mathbf{u}_9^2 = \mathbf{l}_9^2 \)

Putting together: \( \mathbf{e}^2 \times (\mathbf{P}^2\lambda \mathbf{P}^1 + \mathbf{P}^2\mathbf{C}^1) = \mathbf{l}_9^2 \)

Clearly \( \mathbf{e}^2 \times \mathbf{P}^2\mathbf{C}^1 = 0 \), then: \( \mathbf{e}^2 \times \lambda \mathbf{P}^2 \mathbf{P}^1 + \mathbf{u}_9^1 = \mathbf{l}_9^2 \)

But we also know \( \mathbf{l}_9^2 \top \mathbf{u}_9^2 = 0 \) since the point \( \mathbf{u}_9^2 \) must lie on the line \( \mathbf{l}_9^2 \).
Derivation of the **Fundamental matrix**, cont.

\[ e^2 \times \lambda p^2 p^{1+} u_9^1 = l_9^2 \]

But we also know \( l_9^2 \top u_9^2 = 0 \) since the point \( u_9^2 \) must lie on the line.

Introducing a small matrix trick \([e]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}\)

we may rewrite the cross product as a matrix multiplication

\[ l_9^2 = \left([e^2]_\times \lambda p^2 p^{1+}\right) u_9^1 \]

Inserting into \( l_9^2 \top u_9^2 = 0 \) yields:

\[ u_9^1 \top \underbrace{\left([e^2]_\times \lambda p^2 p^{1+}\right)}_{F} \top u_9^2 = 0 \]

\[ u_9^2 \top Fu_9^1 = 0 \]
$\mathbf{u}_i^2 \mathbf{F} \mathbf{u}_i^1 = 0$ holds for any corresponding pair $\mathbf{u}_i^1, \mathbf{u}_i^2$.

$\mathbf{F}$ does not depend on the scene structure, only on cameras.

All epipolar lines intersect in epipoles.
Epipolar geometry—overview
Epipolar geometry—what is it good for
Epipolar geometry—what is it good for
Epipolar geometry—what is it good for
Epipolar geometry—what is it good for
Fundamental matrix, so what . . .

Motion and 3D structure is where?
Essential matrix

For the Fundamental matrix we derived

\[ \mathbf{u}_i^1 \top \left( [\mathbf{e}^2] \times \mathbf{P}^2 \mathbf{P}^1^+ \right) \top \mathbf{u}_i^2 = 0 \]

\( \mathbf{u} \) denote point coordinates in pixels. Let coincide the world system with the coordinate system of the first camera.

\[ \mathbf{u}^1 = \mathbf{K}^1 \left[ \mathbf{I} \ 0 \right] \mathbf{X} \quad \mathbf{u}^2 = \mathbf{K}^2 \left[ \mathbf{R} \ \mathbf{t} \right] \mathbf{X} \]

Remind the normalized image coordinates \( \mathbf{x} = \mathbf{K}^{-1} \mathbf{u} \). We can define normalized cameras \( \mathbf{x} = \hat{\mathbf{P}} \mathbf{X} \) and insert the equation above.

\[ \mathbf{x}_i^1 \top \left( [\mathbf{x}_e^2] \times \hat{\mathbf{P}}^2 (\hat{\mathbf{P}}^1)^+ \right) \top \mathbf{x}_i^2 = 0 \]

where \( \mathbf{E} \) is the Essential matrix
Essential matrix — cont’d

\[
E = [x_e^2] \times \hat{P}^2(\hat{P}^1)^+ \\
= [x_e^2] \times [ R \hspace{0.5cm} t ] [ I \hspace{0.5cm} 0 ]^+ \\
= [x_e^2] \times R
\]

\[
x_e^2 = \hat{P}^2C^1 \\
= [ R \hspace{0.5cm} t ] \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
= t
\]

\[
E = [t] \times R
\]

E comprises the motion between cameras!

after simple manipulation, we see \( E = K^2^T FK^1 \)
3D scene reconstruction—Linear method

A scene point $\mathbf{X}$ is observed by two cameras $\mathbf{P}^1$ and $\mathbf{P}^2$. Assume we know its projections $[u^j, v^j]^\top$

$$\mathbf{u} = \mathbf{PX}, \quad u = \frac{p^\top_1 \mathbf{X}}{p^\top_3 \mathbf{X}}, \quad u(p^\top_3 \mathbf{X}) - p^\top_1 \mathbf{X} = 0,$$

the same derivation for $\mathbf{v}$ and for both cameras:

$$\begin{bmatrix}
    u^1 p^\top_3 - p^\top_1 \\
    v^1 p^\top_3 - p^\top_1 \\
    u^2 p^\top_3 - p^\top_1 \\
    v^2 p^\top_3 - p^\top_1 
\end{bmatrix} \begin{bmatrix}
    \mathbf{X}
\end{bmatrix} = \begin{bmatrix}
    0
\end{bmatrix}$$

Set of linear homogeneous equations. A standard LSQ solution\(^1\) may be used.

Not an optimal solution. It minimizes algebraic not geometric error. More methods can be found in [3, Chapter 12]

\(^1\)http://cmp.felk.cvut.cz/cmp/courses/XE33PVR/WS20072008/Lectures/Supporting/constrained_lsq.pdf
Errors in reconstruction

- the bigger angle between rays the better reconstruction, however . . .
- also the more difficult image matching

\(^2\text{Sketch borrowed from [2]}\)
Problems with image matching

Good for matching, bad for reconstruction
Problems with image matching

Good for reconstruction, bad for matching
Estimation of $F$ or $E$ from corresponding point pairs

$$u_i^2 \top Fu_i^1 = 0$$

for any pair of matching points. Each matching pair gives one linear equation

$$u^2u^1f_{11} + u^2v^1f_{12} + u^2f_{13} \ldots = 0$$

which may be rewritten as a vector inner product

$$[u^2u^1, u^2v^1, u^2, v^2u^1, v^2v^1, v^2, u^1, v^1, 1]f = 0$$

A set of $n$ pairs forms a set of linear equations

$$Af = \begin{bmatrix}
  u_1^2u_1^1 & u_1^2v_1^1 & u_1^2 & v_1^2u_1^1 & v_1^2v_1^1 & v_1^2 & u_1^1 & v_1^1 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  u_n^2u_n^1 & u_n^2v_n^1 & u_n^2 & v_n^2u_n^1 & v_n^2v_n^1 & v_n^2 & u_n^1 & v_n^1 & 1
\end{bmatrix}f = 0$$
Estimation of $F$—normalized 8-point algorithm

Solution of

$$Af = \begin{bmatrix} u_1^2 u_1^1 & u_1^2 v_1^1 & u_1^2 & v_1^2 u_1^1 & v_1^2 v_1^1 & v_1^2 & u_1^1 & v_1^1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n^2 u_n^1 & u_n^2 v_n^1 & u_n^2 & v_n^2 u_n^1 & v_n^2 v_n^1 & v_n^2 & u_n^1 & v_n^1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} = 0$$

is a standard **LSQ** solution\(^3\)

**Point normalization**

Consider a point pair $\mathbf{u}^1 = [150, 250, 1]^\top$, $\mathbf{u}^2 = [250, 350, 1]^\top$. It is clear that row elements in $A$ are unbalanced.

$$\mathbf{a}^\top = [10^6, 10^6, 10^3, 10^6, 10^6, 10^3, 10^3, 10^3, 10^0]$$

This influences the numerical stability. Solution: normalization of the point coordinates before computation.

\(^3\)http://cmp.felk.cvut.cz/cmp/courses/XE33PVR/WS20072008/Lectures/Supporting/constrained_lsq.pdf
Estimation of $F$—normalized 8-point algorithm

Transform the coordinates of points so that the centroid is at the origin of coordinates nad RMS distance is equal to $\sqrt{2}$.

$\hat{u}^1 = T^1 u^1$ and $\hat{u}^2 = T^2 u^2$, where $T^i$ are $3 \times 3$ normalizing matrices including translation nad scaling.

Compute $\hat{F}$ by using the standard LSQ method, $\hat{u}^2^T \hat{F} \hat{u}^1 = 0$. Denormalize the solution $F = T^2^T \hat{F} T^1$

Historical remarks

The linear algorithm for estimation epipolar geometry (calibrated case—essential matrix) was suggest in [5]. The normalization for the uncalibrated case (fundamental matrix) was introduced in [4].
Point normalization

original points

normalized points
Zero motion

we derived

\[ E = [t] \times R \]

what happens if \( t = 0 \)?
Common $t = 0$ case—I mage Panoramas
What are the differences in images
general motion
What are the differences in images?

general motion

- objects in different depths make occlusions
- the mapping is certainly not 1:1
What are the differences in images rotation
What are the differences in images rotation

- no occlusions
- the mapping may be 1:1
Mapping between images
The book [3] is the ultimate reference. It is a must read for anyone wanting use cameras for 3D computing.

Details about matrix decompositions used throughout the lecture can be found at [1]


End
\[ X_9 = \lambda P^1 + u_9^1 + C^1 \]
\[ X_9 = \lambda P^1 + u^1_9 + C^1 \]

\[ e^2 = P^2 C^1 \]
$X_9 = \lambda P^1 + u_9^1 + C^1$

e^2 = P^2 C^1
\( \mathbf{X}_9 = \lambda P^1 + u_9^1 + C^1 \)