Computer Vision II: Multiple View Geometry (IN2228)

Chapter 06 2D-2D Geometry
(Part 1 Overview and Fundamentals)

Dr. Haoang Li

01 June 2023  11:00-11:45
Outline

- Overview of 2D-2D Geometry
- Two-view Geometric Constraints
- Eight-point Method
Overview of Two-view Geometry

- Intuitive Illustration

✓ Camera pose estimation
We can easily imagine that the image A is obtained by the left camera (eye) and image B is obtained by the right camera (eye).

Camera motion can be inferred from two consecutive image frames.
Overview of Two-view Geometry

✓ Intuitive Illustration

✓ Camera pose estimation

We infer the camera motion from some object correspondences. Objects can be further abstracted by points and lines.

Object correspondences

Point correspondences
Overview of Two-view Geometry

- Intuitive Illustration

✓ 3D reconstruction
Overview of Two-view Geometry

- Intuitive Illustration

✓ 3D reconstruction
Given a pair of 2D points in two images, the 3D point's position in space is found as the intersection of the two projection rays.
Overview of Two-view Geometry

Problem Formulation

2D-2D point correspondence

Estimated poses and 3D structure
Overview of Two-view Geometry

Problem Formulation

- Can we solve the estimation of relative motion \((R,T)\) without any prior information about 3D points? Yes! The next couple of slides prove that this is possible.

- Once \((R,T)\) are known, the 3D points can be triangulated using the triangulation algorithm (i.e., least square approximation plus reprojection error minimization)

\[
P_i = ? = \begin{cases} \begin{align*} K_1, R_1, T_1 & = \? \hfill K_2, R_2, T_2 = \? \\ K_1, R_1, T_1 & = \? \hfill K_2, R_2, T_2 = ? \end{align*} \end{cases}
\]
Overview of Two-view Geometry

- Problem Formulation

- Recover simultaneously 3D scene structure and camera poses (up to scale) from two images. (More specifically, camera pose first, followed by 3D structure.)
- Intrinsic parameters of camera is known from calibration. We can also handle uncalibrated case.
Overview of Two-view Geometry

Problem Formulation

Given a set of $n$ point correspondences $\{p_i^1 = (u_i^1, v_i^1), p_i^2 = (u_i^2, v_i^2)\}$ between two images, where $i = 1 \ldots n$, the goal is to simultaneously
• estimate the 3D points $P^i$ and
• the camera relative-motion parameters ($R$, $T$)

Perspective projection

\[
\lambda_1^i \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = K_1[I|0] \cdot \begin{bmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{bmatrix}
\]

\[
\lambda_2^i \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix} = K_2[R|T] \cdot \begin{bmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{bmatrix}
\]
Overview of Two-view Geometry

- Scale Ambiguity

If we rescale the entire scene and camera views, the projections (in pixels) of the scene points in both images remain exactly the same:

- Reduce the size of 3D object: smaller projection
- Reduce the distance from camera to 3D object: bigger projection
- **Simultaneously** reduce the size of 3D object and reduce the distance from camera to 3D object?
Overview of Two-view Geometry

- Scale Ambiguity

✓ For monocular case, it is not possible to recover the absolute scale of the scene.

✓ Thus, only 5 degrees of freedom are measurable:
  • Three parameters to describe the rotation
  • Two parameters for the parameters for the translation up to a scale (we can only compute the direction of translation but not its length)
Overview of Two-view Geometry

Number of Point Correspondences

- **4n knowns:**
  - n correspondences; each one \((u^i_1, v^i_1)\) and \((u^i_2, v^i_2)\), \(i = 1 ... n\)

- **5 + 3n unknowns**
  - 5 for the motion up to a scale (3 for rotation, 2 for translation)
  - 3n is number of coordinates of n 3D points \((x, y, z)\)

If and only if the number of independent equations \(\geq\) number of unknowns

\[
4n \geq 5 + 3n \quad \Rightarrow \quad n \geq 5
\]
Overview of Two-view Geometry

- Number of Point Correspondences

- In 1913, Kruppa showed that 5 image correspondences is the minimal case and that there can be at up to 11 solutions [1].

- In 1981, the first popular solution uses 8 points and is called the 8 point algorithm or Longuet Higgins algorithm [2].

- In 2004, Nister proposed the first efficient and non-iterative solution. It uses Groebner basis decomposition [3].

Geometric Constraints

- **Epipolar planes and lines**

  - The camera centers $C_l$ and $C_r$ and the image point $p$ and $p'$ determine the so-called epipolar plane.
  - The intersections of the epipolar plane with the two image planes are called epipolar lines.
Geometric Constraints

- Epipolar planes and lines

- The epipolar line is the projection of a back projected ray $\pi^{-1}(p)$ onto the other camera image.
- The epipole is the projection of the optical center on the other camera image.
- A pair of images has two epipoles.
Geometric Constraints

- Essential Matrix

Coplanarity constraint

\[ \overline{p}_1 = \begin{bmatrix} \overline{u}_1 \\ \overline{v}_1 \\ 1 \end{bmatrix}, \quad \overline{p}_2 = \begin{bmatrix} \overline{u}_2 \\ \overline{v}_2 \\ 1 \end{bmatrix} \]

Normalized image coordinates

\[ \overline{p}_1, \overline{p}_2, T \text{ are coplanar} \]

\[ \overline{p}_2^T \cdot n = 0 \Rightarrow \overline{p}_2^T \cdot (T \times \overline{p}_1') = 0 \Rightarrow \overline{p}_2^T (T \times \overline{R \overline{p}_1}) = 0 \]

Right camera frame

Orthogonality

Normal of epipolar plane

From dot product to matrix multiplication

R and T denote the rotation and translation from C1 to C2. (arrow of T doesn’t represent the direction of T, but “the coordinates of the center of C1” in C2.)
Geometric Constraints

- Essential Matrix

Coplanarity constraint

\[ \tilde{p}_2^T (T \times (R\tilde{p}_1)) = 0 \]

\[ \Rightarrow \tilde{p}_2^T [T_x] R \tilde{p}_1 = 0 \]

Skew-symmetric matrix

\[ a \times b = [a]_\times b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \]

Definition of essential matrix

\[ E = [T_x] R \quad \text{essential matrix} \]

\[ R \quad \text{and} \quad T \quad \text{can be computed from} \ E \]
Geometric Constraints

- From Essential Matrix to Fundamental Matrix

So far, we have assumed that the camera intrinsic parameters are known and we have used normalized image coordinates to get the epipolar constraint for calibrated cameras:

\[
\begin{bmatrix}
\bar{u}_2^i \\
\bar{v}_2^i \\
1
\end{bmatrix}
E
\begin{bmatrix}
\bar{u}_1^i \\
\bar{v}_1^i \\
1
\end{bmatrix} = 0
\]

\[
\begin{bmatrix}
\bar{u}_1^i \\
\bar{v}_1^i \\
1
\end{bmatrix} = K_1^{-1}
\begin{bmatrix}
u_1^i \\
v_1^i \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
\bar{u}_2^i \\
\bar{v}_2^i \\
1
\end{bmatrix} = K_2^{-1}
\begin{bmatrix}
u_2^i \\
v_2^i \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
\bar{u}_2^i \\
\bar{v}_2^i \\
1
\end{bmatrix} = K_2^{-T} E K_1^{\frac{1}{2}}
\begin{bmatrix}
u_2^i \\
v_2^i \\
1
\end{bmatrix} = 0
\]

Normalized image coordinates
Geometric Constraints

- Fundamental Matrix

✓ Definition of fundamental matrix

\[
\begin{bmatrix}
    u_2^i \\
    v_2^i \\
    1
\end{bmatrix}^T \begin{bmatrix}
    K_2^{-T} & E & K_1^{-1}
\end{bmatrix} \begin{bmatrix}
    u_1^i \\
    v_1^i \\
    1
\end{bmatrix} = 0
\]

Advantage: Based on fundamental matrix, we work directly in ordinary image plane, instead of normalized image plane.
Geometric Constraints

✓ Computation of Fundamental/Essential Matrix

✓ Eight-point method (Direct linear transform--DLT)
  • Essential matrix
  • Fundamental matrix

✓ Five-point method (introduce in the next class)
  • Essential matrix
Eight-point Method

- Classical Version

- We first take essential matrix estimation for example [1].

- Each pair of point correspondences \( \mathbf{p}_1 = (\bar{u}_1, \bar{v}_1, 1)^T, \quad \mathbf{p}_2 = (\bar{u}_2, \bar{v}_2, 1)^T \)

  provides a linear equation:

  \[
  \mathbf{p}_2^T \mathbf{E} \mathbf{p}_1 = 0
  \]

  Normalized image coordinates

  \[
  \mathbf{E} = \begin{bmatrix}
  e_{11} & e_{12} & e_{13} \\
  e_{21} & e_{22} & e_{23} \\
  e_{31} & e_{32} & e_{33}
  \end{bmatrix}
  \]

Eight-point Method

- Classical Version

\[
\begin{align*}
\overline{u}_2\overline{u}_1 e_{11} + \overline{u}_2\overline{v}_1 e_{12} + \overline{u}_2 e_{13} + \overline{v}_2\overline{u}_1 e_{21} + \overline{v}_2\overline{v}_1 e_{22} + \overline{v}_2 e_{23} + \overline{u}_1 e_{31} + \overline{v}_1 e_{32} + e_{33} &= 0
\end{align*}
\]

Note:
- The 8-point algorithm assumes that the entries of $E$ are all independent. This is not true since, for the calibrated case, they depend on 5 parameters ($R$ and $T$).
- The 5-point algorithm (introduced later) uses the epipolar constraint considering the dependencies among all entries.
Eight-point Method

- Classical Version

For $n$ points, we can write

Normalized image coordinates

\[
\begin{bmatrix}
\bar{u}_2 
\bar{u}_1 \n
\bar{v}_2 
\bar{v}_1 

\vdots 
\vdots 

\bar{u}_2^n 
\bar{v}_2^n
\end{bmatrix}
\begin{bmatrix}
\bar{u}_1 
\bar{v}_1 
\vdots 
\vdots 
\bar{u}_1^n 
\bar{v}_1^n
\end{bmatrix}
= 0
\Rightarrow Q \cdot \bar{E} = 0
\]
Eight-point Method

➢ Classical Version

Minimal solution
• \( Q_{(n \times 9)} \) should have rank 8 to have a unique (up to a scale) non-trivial solution \( \vec{E} \)
• Different E matrices up to scale lead to the same result of rotation and translation (norm=1)
• Each point correspondence provides 1 independent equation
• Thus, 8 point correspondences are needed

Over-determined solution
• \( n > 8 \) points
• A solution is to minimize \( ||Q\vec{E}||^2 \) subject to the constraint \( ||\vec{E}||^2 = 1 \)
• The solution is the eigenvector corresponding to the smallest eigenvalue of the matrix \( Q^TQ \)
Eight-point Method

- Extension to Fundamental matrix

- Similarly, eight-point method for fundamental matrix can be formulated as

\[
\begin{bmatrix}
  u_2^i \\
  v_2^i \\
  1
\end{bmatrix}^T \begin{bmatrix}
  u_1^i \\
  v_1^i \\
  1
\end{bmatrix} = 0 \quad \Rightarrow \quad \begin{bmatrix}
  u_2^1 u_1^1 & u_2^1 v_1^1 & u_2^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\
  u_2^2 u_1^2 & u_2^2 v_1^2 & u_2^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  u_2^n u_1^n & u_2^n v_1^n & u_2^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n & u_1^n & v_1^n & 1
\end{bmatrix} \begin{bmatrix}
  f_{11} \\
  f_{12} \\
  f_{13} \\
  f_{21} \\
  f_{22} \\
  f_{23} \\
  f_{31} \\
  f_{32} \\
  f_{33}
\end{bmatrix} = 0
\]

- We use the original image coordinates instead of normalized image coordinates.
Eight-point Method

- Normalized Version

Motivation

- Orders of magnitude difference between column of data matrix.
- Least-square method yields poor results.

![Matrix equation]

\[
\begin{bmatrix}
| f_{11} | \\
| f_{12} | \\
| f_{13} | \\
| f_{21} | \\
| f_{22} | \\
| f_{23} | \\
| f_{31} | \\
| f_{32} | \\
| f_{33} | \\
\end{bmatrix} = 0
\]

![Data matrix]

\[
\begin{bmatrix}
250906.38 & 183269.57 & 921.81 & 208931.10 & 146766.13 & 738.21 & 272.19 & 198.81 & 1.00 \\
2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 & 1.00 \\
416374.23 & 871684.30 & 935.47 & 408110.89 & 854564.92 & 916.90 & 445.10 & 931.61 & 1.00 \\
191103.60 & 171759.40 & 410.27 & 416493.62 & 374125.90 & 893.65 & 465.99 & 418.65 & 1.00 \\
48908.86 & 30401.76 & 57.89 & 29604.57 & 105309.58 & 352.87 & 646.22 & 525.15 & 1.00 \\
164706.04 & 546559.67 & 813.17 & 1998.37 & 6668.15 & 9.86 & 202.65 & 672.14 & 1.00 \\
116407.01 & 2727.75 & 138.89 & 169941.27 & 3992.21 & 202.77 & 839.12 & 19.64 & 1.00 \\
135364.58 & 75411.13 & 198.72 & 411350.03 & 229127.76 & 603.79 & 681.28 & 379.48 & 1.00 \\
\end{bmatrix}
\]

~10000  ~10000  ~100   ~10000  ~10000  ~100   ~100   ~100   ~100   1

Orders of magnitude difference between column of data matrix → least-squares yields poor results
Eight-point Method

- Normalized Version

Motivation

✓ Poor numerical conditioning, which makes results very sensitive to noise
✓ This problem can be fixed by rescaling the data: Normalized 8-point algorithm [1]

Eight-point Method

Normalised Version

In the original 1997 paper, Hartley proposed to rescale the two 2D point sets such that the centroid of each set is 0 and the mean standard deviation $\sqrt{2}$ (equivalent to having the points distributed around a circled passing through the four corners of the $[-1,1] \times [-1,1]$ square).
Eight-point Method

- Normalized Version

This can be done for every point as follows:

$$\hat{p}^i = \frac{\sqrt{2}}{\sigma} (p^i - \mu)$$

where $\mu = (\mu_x, \mu_y) = \frac{1}{N} \sum_{i=1}^{n} p^i$ is the centroid and $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{n} \|p^i - \mu\|^2}$ is the standard deviation of the point set.

This transformation can be expressed in matrix form using homogeneous coordinates:

$$\hat{p}^i = \begin{bmatrix} \frac{\sqrt{2}}{\sigma} & 0 & -\frac{\sqrt{2}}{\sigma} \mu_x \\ 0 & \frac{\sqrt{2}}{\sigma} & -\frac{\sqrt{2}}{\sigma} \mu_y \\ 0 & 0 & 1 \end{bmatrix} p^i$$
Eight-point Method

$view\text{-}\text{Normalized Version}

The Normalized 8-point algorithm can be summarized in three steps:

1. Normalize the point correspondences: \( \hat{p}_1 = B_1 p_1 \), \( \hat{p}_2 = B_2 p_2 \)
2. Estimate normalized \( \hat{F} \) with 8 point algorithm using normalized coordinates \( \hat{p}_1, \hat{p}_2 \)
3. Compute unnormalized \( F \) from \( \hat{F} \)

\[
\begin{bmatrix}
  u_2^i \\
  v_2^i \\
  1
\end{bmatrix}^T
F
\begin{bmatrix}
  u_1^i \\
  v_1^i \\
  1
\end{bmatrix} = 0
\]
Eight-point Method

- Normalized Version

Comparison between Normalized and non-normalized versions

<table>
<thead>
<tr>
<th></th>
<th>8-point</th>
<th>Normalized 8-point</th>
<th>Nonlinear refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Ep. Line Distance</td>
<td>2.33 pixels</td>
<td>0.92 pixel</td>
<td>0.86 pixel</td>
</tr>
</tbody>
</table>
Summary

- Overview of 2D-2D Geometry
- Two-view Geometric Constraints
- Eight-point Method
Thank you for your listening!
If you have any questions, please come to me :-)