Stereo and Epipolar geometry

Jana Kosecka

Previously

- Image Primitives (feature points, lines, contours)
- Today:
  - How to match primitives between two (multiple) views
  - Goals: 3D reconstruction, recognition
  - Stereo matching and reconstruction (canonical configuration)
  - Epipolar Geometry (general two view setting)

Why Stereo Vision?

- 2D images project 3D points into 2D:
  - 3D points on the same viewing line have the same 2D image:
  - 2D imaging results in depth information loss

Canonical Stereo Configuration

- Assumes (two) cameras
- Known positions and focal lengths
- Recover depth

\[ Z = f \frac{f - p}{T - X_1 - X_2} \]

\[ Z = \frac{RT}{\text{disparity}} \]
Depth can be recovered by triangulation

Prerequisite: Finding Correspondences

Random Dot Stereo-grams

B. Julesz: showed that the depth can be perceived in the absence of any identifiable objects in correspondence

Autostereograms

- Depth perception from one image
- Viewing trick the brain by focusing at the plane behind - match can be established perception of 3D

Correspondence Problem

- Two classes of algorithms:
  - Correlation-based algorithms
    - Produce a DENSE set of correspondences
  - Feature-based algorithms
    - Produce a SPARSE set of correspondences

Stereo – Photometric Constraint

- Same world point has same intensity in both images.
  - Lambertian fronto-parallel
  - Issues (noise, specularities, foreshortening)

- Difficulties – ambiguities, large changes of appearance, due to change of viewpoint, non-uniqueness
Stereo Matching

For each scanline, for each pixel in the left image:
- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost
- This will never work, so: improvement match windows

Comparing Windows:

Minimize \( \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2 \)  
Sum of Squared Differences

Maximize \( C_{fg} = \sum_{[i,j] \in R} f(i,j)g(i,j) \)
Cross correlation

It is closely related to the SSD:

\[
SSD = \sum_{[i,j] \in R} (f - g)^2 = \sum_{[i,j] \in R} f^2 + \sum_{[i,j] \in R} g^2 - 2\sum_{[i,j] \in R} fg
\]

For each window, match to closest window on epipolar line in other image.

Window size

- Effect of window size

Better results with adaptive window


(slides O. Camps) 9

(slides O. Camps) 10

(slides O. Camps) 11

(slides O. Camps) 12
Stereo results

- Data from University of Tsukuba

Results with window correlation

- Window-based matching (best window size)
- Ground truth

Results with better method

- State of the art
- Ground truth

More of advanced stereo (later)

- Ordering constraint
- Dynamic programming
- Global optimization
Two View – General Configuration

- Motion between the two views is not known

Given two views of the scene recover the unknown camera displacement and 3D scene structure

Pinhole Camera Imaging Model

- 3D points $X = [X, Y, Z, W]^T \in \mathbb{R}^4$, ($W = 1$)
- Image points $x = [x, y]^T \in \mathbb{R}^3$, ($z = 1$)
- Perspective Projection $\lambda x = X$
  $\lambda = Z \ x = \frac{X}{Z} \ y = \frac{Y}{Z}$
- Rigid Body Motion $\Pi = [R, T] \in \mathbb{R}^{3 \times 4}$

$\lambda x' = K \Pi x = \begin{bmatrix} f_s x' & f_s p_x & a_x \\ f_s y' & f_s p_y & a_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ T \\ X \end{bmatrix}$

Rigid Body Motion – Two Views

$X = [X, Y, Z, 1]^T$

$x = [x, y, 1]^T$

$\lambda x = \Pi x = [R, T] x$

$\Pi = [R, T] \in \mathbb{R}^{3 \times 4}$

3D Structure and Motion Recovery

Euclidean transformation

$\lambda x_2 = R \lambda x_1 + T$

Measurements unknowns

Corresponding points

$\sum_{i=1}^{n} ||x_i - \pi(R_i, T_i, X)||^2 + ||x_i - \pi(R_i, T_i, X)||^2$

Find such Rotation and Translation and Depth that the reprojection error is minimized

Two views $\rightarrow$ 200 points
6 unknowns – Motion 3 Rotation, 3 Translation
- Structure 200x3 coordinates
- (-) universal scale

Difficult optimization problem
Notation

- Cross product between two vectors in

\[ c = a \times b \]

where

\[
\hat{a} = \begin{bmatrix}
0 & -a_3 & a_2 \\
a_3 & 0 & -a_1 \\
-a_2 & a_1 & 0
\end{bmatrix}
\]

Epipolar Geometry

- Epipolar lines \( l_1, l_2 \)
- Epipoles \( e_1, e_2 \)
- Epipolar transfer
- Additional constraints

\[ l_1 \sim E^T x_2 \quad l_1^T x_1 = 0 \quad l_2 \sim E x_1 \]

\[ E e_1 = 0 \quad l_1^T e_1 = 0 \quad e_2 E^T = 0 \]

Characterization of Essential Matrix

\[ x_2^T \hat{T} R x_1 = 0 \]

Essential matrix \( E = \hat{T} R \) special 3x3 matrix

\[
x_2^T \begin{bmatrix}
e_1 & e_2 & e_3 \\
e_4 & e_5 & e_6 \\
e_7 & e_8 & e_9
\end{bmatrix} x_1 = 0
\]

(Essential Matrix Characterization)
A non-zero matrix \( E \) is an essential matrix iff its SVD: \( E = U \Sigma V^T \) satisfies: \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \) with \( \sigma_1 = \sigma_2 \neq 0 \) and \( \sigma_3 = 0 \) and \( U, V \in SO(3) \)
Estimating Essential Matrix

- Find such Rotation and Translation that the epipolar error is minimized
  \[ \min_{E} \sum_{i=1}^{n} (x_i^T E x_i)^2 \]
- Space of all Essential Matrices is 5 dimensional
- 3 DOF Rotation, 2 DOF – Translation (up to scale)
- Denote \( a = [x_1 \otimes x_2, x_1 \otimes x_2, x_1 \otimes x_2, x_1 \otimes x_2, x_1 \otimes x_2, x_1 \otimes x_2]^T \)
  \[ E^a = [c_1, c_4, c_7, c_2, c_5, c_8, c_3, c_6, c_9]^T \]
- Collect constraints from all points
  \[ \chi E^a = 0 \]

Estimating Essential Matrix

\[ \min_{E} \sum_{i=1}^{n} (x_i^T E x_i)^2 \Rightarrow \min_{E^a} \| \chi E^a \|^2 \]

Estimating Essential Matrix

\[ \begin{align*}
\text{Solution is} & \quad \text{Eigenvector associated with the smallest eigenvalue of } \chi^T \chi \\
& \quad \text{If } \text{rank}(\chi^T \chi) < 8 \text{ degenerate configuration} \\
& \quad E, \text{estimated using linear least squares} \\
& \quad \text{unstack } E_i \rightarrow F
\end{align*} \]

Projection on to Essential Space

\[ \begin{align*}
\text{(Project onto a space of Essential Matrices) } & \quad F \rightarrow E \\
& \quad \text{If the SVD of a matrix } F \in \mathbb{R}^{3 \times 3} \text{ is given by } F = U \text{diag}(\sigma_1, \sigma_2, \sigma_3) V^T \\
& \quad \text{then the essential matrix } E \text{ which minimizes the} \\
& \quad \text{Frobenius distance } |E - F|^2 \text{ is given by } E = U \text{diag}(\sigma, 0, 0) V^T \\
& \quad \text{with } \sigma = \frac{\sigma_1 + \sigma_2}{2}
\end{align*} \]

Pose Recovery from Essential Matrix

\[ E = \hat{T} R \]

(Pose Recovery)

There are two relative poses \((R, T)\) with \( R \in \mathbb{R}^3 \) and \( T \in \mathbb{S}^2 \), corresponding to a non-zero matrix essential matrix.

\[ E = U \Sigma V^T \]

\[ \begin{align*}
(\hat{T}_1, R_1) &= (UR_2(\frac{1}{\sqrt{2}}) \Sigma U^T, UR_2(\frac{1}{\sqrt{2}}) V^T) \\
(\hat{T}_2, R_2) &= (UR_2(-\frac{1}{\sqrt{2}}) \Sigma U^T, UR_2(-\frac{1}{\sqrt{2}}) V^T)
\end{align*} \]

\[ \Sigma = \text{diag}(1, 1, 0) \quad R_x(\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

- Twisted pair ambiguity \((R_2, T_2) = (e^{R_2} R_1, -T_1)\)

Two view linear algorithm - summary

\[ E = \{ \hat{T} R | R \in SO(2), T \in \mathbb{S}^2 \} \]

- Solve the LLSE problem:
  \[ \min_{E} \sum_{j=1}^{n} (x_j^T E x_j)^2 \Rightarrow \chi^T E = 0 \]
  \[ \text{Solution eigenvector associated with smallest eigenvalue of } \chi^T \chi \]

- Compute SVD of \( F \) recovered from data
  \[ E^x \rightarrow F \rightarrow U \Sigma V^T \]

- Project onto the essential manifold:
  \[ \Sigma' = \text{diag}(1, 1, 0) \quad E = U \Sigma' V^T \]

- 8-point linear algorithm

- Recover the unknown pose:
  \[ (\hat{T}, R) = (UR_2(\pm \frac{1}{\sqrt{2}}) \Sigma U^T, UR_2(\pm \frac{1}{\sqrt{2}}) V^T) \]
Pose Recovery

- There are two pairs \((R, T)\) corresponding to essential matrix.
- Positive depth constraint disambiguates the impossible solutions.
- Translation has to be non-zero.
- Points have to be in general position:
  - degenerate configurations: planar points
  - quadratic surface
- Linear 8-point algorithm.
- Nonlinear 5-point algorithms yield up to 10 solutions.

\[ \lambda_2 \mathbf{x}_2 = R \lambda_1 \mathbf{x}_1 + \gamma T \]

\[ \lambda_1^j \mathbf{x}_1^j R \mathbf{x}_1^j + \gamma \mathbf{x}_2^j T = 0, \quad j = 1, 2, \ldots, n \]

- Eliminate one of the scale's unknowns.

\[ M^j \lambda^j \mathbf{x}_1^j = [x_1^j R x_1^j, x_2^j T] \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0 \]

3D Structure Recovery

- Alternatively recover each point depth separately.

Example

Two views

Point Feature Matching

Example

Epipolar Geometry
Planar homography
Linear mapping relating two corresponding planar points in two views

Decomposition of H (into motion and plane normal)
- Algebraic elimination of depth $x_2^2 H x_1 = 0$
- can be estimated linearly $H_{12} = \lambda H$
- Normalization of $H = H_{12}/\lambda$
- Decomposition of H into 4 solutions

Uncalibrated Camera
Linear transformation $x' = [x' \ y' \ 1] = K x = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Uncalibrated Epipolar Geometry
Epipolar constraint $x_2^T K^{-T} T R K^{-1} x_1 = 0$
Fundamental matrix $F = K^{-T} T R K^{-1}$
Properties of the Fundamental Matrix

- Epipolar lines \( l_1, l_2 \)
- Epipoles \( e_1, e_2 \)

\[
\begin{align*}
    l_1 & \sim F^x x_2' \\
    F e_1 & = 0 \\
    l_2 & \sim F x_1' \\
    F e_2 & = 0
\end{align*}
\]

There is little structure in the matrix \( F \) except that \( \det(F) = 0 \)

Epipolar Geometry for Parallel Cameras

- Epipoles are at infinity
- Epipolar lines are parallel to the baseline

Estimating Fundamental Matrix

- Find such \( F \) that the epipolar error is minimized
  \[
  \min_F \sum_{j=1}^n (x_2^j)^T F x_1^j \]

- Fundamental matrix can be estimated up to scale

- Denote \( a = x_1^j \otimes x_2^j \)

- Collect constraints from all points

\[
\min_F \sum_{j=1}^n (x_2^j)^T F x_1^j \rightarrow min_{F} \| \chi F \|^2
\]
**Two view linear algorithm – 8-point algorithm**

- Solve the LLSE problem:
  \[ \min_F \sum_{j=1}^3 (x_j^T F x_j)^2 \Rightarrow \min_F \| \chi F \|_2^2 = 0 \]
- Solution eigenvector associated with smallest eigenvalue of \( \chi^T \chi \)
- Compute SVD of \( F \) recovered from data
  \[ F = U \Sigma V^T \quad \Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \]
- Project onto the essential manifold:
  \[ \Sigma' = \text{diag}(\sigma_1, \sigma_2, 0) \quad F = U \Sigma' V^T \]
- cannot be unambiguously decomposed into pose and calibration
  \[ F = K^{-T} P K^{-1} \]

**Dealing with correspondences**

- Previous methods assumed that we have exact correspondences
- Followed by linear least squares estimation
- Correspondences established either by tracking (using affine or translational flow models)
- Or wide-baseline matching (using scale/rotation invariant features and their descriptors)
- In many cases we get incorrect matches/tracks

**Robust estimators for dealing with outliers**

- Use robust objective function
  - The M-estimator and Least Median of Squares (LMedS) Estimator (neither of them can tolerate more than 50% outliers)
- The RANSAC (RAndom SAmple Consensus) algorithm
  - Proposed by Fischler and Bolles
  - Popular technique used in Computer Vision community (and else where for robust estimation problems)
- It can tolerate more than 50% outliers

**The RANSAC algorithm**

- Generate \( M \) (a predetermined number) model hypotheses, each of them is computed using a minimal subset of points
- Evaluate each hypothesis
- Compute its residuals with respect to all data points.
- Points with residuals less than some threshold are classified as its inliers
- The hypothesis with the maximal number of inliers is chosen. Then re-estimate the model parameter using its identified inliers.
### RANSAC – Practice

- The theoretical number of samples needed to ensure 95% confidence that at least one outlier free sample could be obtained.
  \[ \rho = 1 - (1 - \epsilon)^k \cdot s \]
- Probability that a point is an outlier: \( 1 - \epsilon \)
- Number of points per sample: \( k \)
- Probability of at least one outlier free sample: \( \rho \)
- Then number of samples needed to get an outlier free sample with probability \( \rho \)
  \[ s = \frac{\log(1 - \rho)}{\log(1 - (1 - \epsilon)^k)} \]

### The difficulty in applying RANSAC

- **Drawbacks of the standard RANSAC algorithm**
  - Requires a large number of samples for data with many outliers (exactly the data that we are dealing with)
  - Needs to know the outlier ratio to estimate the number of samples
  - Requires a threshold for determining whether points are inliers
- **Various improvements to standard approaches**
  [Torr’99, Murray’02, Nister’04, Matas’05, Sutter’05 and many others]

### Adaptive RANSAC

- **s = infinity, sample_count = 0;**
- While \( s > \text{sample\_count} \) repeat
  - Choose a sample and count the number of inliers
  - Set \( \epsilon = 1 - \frac{\text{number\_of\_inliers}}{\text{total\_number\_of\_points}} \)
  - Set \( s \) from \( \epsilon \) and \( \rho = 0.95 \)
  - Increment \( \text{sample\_count} \) by 1
- **terminate**

### RANSAC – Practice

- The theoretical number of samples needed to ensure 95% confidence that at least one outlier free sample could be obtained.
- Example for estimation of essential/fundamental matrix
- Need at least 7 or 8 points in one sample i.e. \( k = 7 \), probability is \( 0.95 \) then the number of samples for different outlier ratio \( \epsilon \)

<table>
<thead>
<tr>
<th>Outlier ratio</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
</tr>
</thead>
<tbody>
<tr>
<td>seven-point algorithm</td>
<td>13</td>
<td>35</td>
<td>106</td>
<td>382</td>
<td>1327</td>
<td>13606</td>
</tr>
<tr>
<td>eight-point algorithm</td>
<td>17</td>
<td>51</td>
<td>177</td>
<td>706</td>
<td>2578</td>
<td>15658</td>
</tr>
</tbody>
</table>

- In practice we do not now the outlier ratio
- Solution adaptively adjust number of samples as you go along
- While estimating the outlier ratio
Robust technique

- Select set of putative correspondences \( x_1^j, x_2^j \)
- \( x_2^j F x_1^j = 0 \)
- Repeat
  1. Select at random a set of 8 successful matches
  2. Compute fundamental matrix
  3. Determine the subset of inliers, compute distance to epipolar line
     \[ d_j = \frac{(s_2^j F s_1^j)^2}{\| s_2^j x_2^j \|^2 + \| s_2^j F s_2^j \|^2} \]
     \( d_j \leq \tau_0 \)
  4. Count the number of points in the consensus set

More correspondences and Robust matching

- RANSAC in action

<table>
<thead>
<tr>
<th>( d_j \leq \tau_0 )</th>
<th>( d_j &gt; \tau_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inliers</td>
<td>Outliers</td>
</tr>
</tbody>
</table>

Epipolar Geometry

- Epipolar geometry in two views
- Refined epipolar geometry using nonlinear estimation of \( F \)
- The techniques mentioned so far simple linear least-squares estimation methods. The obtained estimates are used as initialization for non-linear optimization methods
Special Motions – Pure Rotation

- Calibrated Two views related by rotation only: 
  \[ \frac{X_2}{\lambda} = R_1 X_1 \]

- Uncalibrated Case: 
  \[ \frac{X_2}{\lambda} H X_1 = \frac{X_2}{\lambda} K^{-T} R K^{-1} X_1 = \mathbf{C} \]

- Mapping to a reference view – H can be estimated

- Mapping to a cylindrical surface – applications – image mosaics

Projective Reconstruction

- Euclidean Motion Cannot be obtained in uncalibrated setting (F cannot be uniquely decomposed into R, T and K matrix)

- Can we still say something about 3D?

- Notion of the projective 3D structure (study of projective geometry)

Euclidean vs Projective reconstruction

- **Euclidean reconstruction** – true metric properties of objects, lengths (distances), angles, parallelism are preserved
  - Unchanged under rigid body transformations
  - \( \implies \) Euclidean Geometry – properties of rigid bodies under rigid body transformations, similarity transformation

- **Projective reconstruction** – lengths, angles, parallelism are NOT preserved – we get distorted images of objects – their distorted 3D counterparts \( \implies \) 3D projective reconstruction
  - \( \implies \) Projective Geometry