Step-by-Step model building

Jana Kosecka
Department of Computer Science
George Mason University
http://www.cs.gmu.edu/~kosecka

Review

1. Feature correspondence
2. Camera Calibration
3. Dense Correspondence
4. Texture mapping
5. Epipolar Rectification
6. Feature selection
7. Sparse Structure and camera motion
8. Euclidean Reconstruction
9. 3-D Model
10. Sparse Structure and motion
11. Euclidean Reconstruction
12. Projective Reconstruction
13. Partial Scene Knowledge
14. Partial Motion Knowledge
15. Partial Calibration Knowledge
16. Camera Self-Calibration
17. Landing
18. Augmented Reality
19. Vision Based Control
Examples

Feature Selection

- Compute Image Gradient \( \nabla I^T = [I_x, I_y] \)
- Compute Feature Quality \( C(x) \) measure for each pixel
  \[ C(x) = det(G) + \mu \text{tr}^2(G) \]
  \[ G = \begin{bmatrix} \Sigma I_{xx} & \Sigma I_{xy} \\ \Sigma I_{yx} & \Sigma I_{yy} \end{bmatrix} \]
- Search for local maxima

Feature Quality Function

Feature Tracking

- Translational motion model
  \[ F(d) = \text{min}_{\lambda \in \mathbb{R}} \int \{I(x, y) - I(x + \lambda d)\} \]

- Closed form solution
  \[ d = -G^{-1}b \]
  \[ G = \begin{bmatrix} \Sigma I_{xx} & \Sigma I_{xy} \\ \Sigma I_{yx} & \Sigma I_{yy} \end{bmatrix} \]
  \[ b = \begin{bmatrix} \Sigma I_{x} \\ \Sigma I_{y} \end{bmatrix} \]

1. Build an image pyramid
2. Start from coarsest level
3. Estimate the displacement at the coarsest level
4. Iterate until finest level

Coarse to fine feature tracking

1. Compute \( \frac{2}{k} \)
2. Warp the window \( W(x) \) in the second image by \( 2d_k \)
3. Update the displacement \( d \leftarrow d + 2d_k \)
4. Go to finer level \( k \leftarrow k - 1 \)
5. At the finest level repeat for several iterations
Optical Flow

- Integrate around over image patch

\[ E_o(u) = \sum_{(x,y)} W(x,y)(\nabla I^T(x,y)u(x,y) + I_t(x,y))^2 \]

- Solve

\[ \nabla E_o(u) = \sum_{W(x,y)} (\nabla I^T u + I_t) \]

\[ = 2 \sum_{W(x,y)} \left( \begin{bmatrix} I_x^2 & I_x I_y & I_y^2 \\ I_x l_x & I_y l_y & l_y^2 \end{bmatrix} u + \begin{bmatrix} I_x l_x \\ I_y l_y \end{bmatrix} \right) \]

\[ G u + b = 0 \]

Affine feature tracking

\[ E(A, d, \lambda, \alpha, \beta) = \sum_{m=1}^{M} \| E(x) - E(x) \| \sum_{m=1}^{M} \sum_{n=1}^{N} \left( \begin{array}{c} x^2 \lambda^2 \\ y^2 \lambda^2 \\ x y \lambda^2 \\ x^2 \alpha^2 \\ y^2 \alpha^2 \\ x y \alpha^2 \\ x^2 \beta^2 \\ y^2 \beta^2 \\ x y \beta^2 \\ 1 \end{array} \right) \]

\[ z = S^{-1} c \]

Tracked Features

Wide baseline matching

Point features detected by Harris Corner detector
Wide baseline Feature Matching

1. Select the features in two views
2. For each feature in the first view
3. Find the feature in the second view that maximizes
4. Normalized cross-correlation measure
   \[ \text{NCC}(f_i, f_j) = \frac{\sum (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum (x_i - \bar{x}_i)^2 \sum (y_i - \bar{y}_i)^2}} \]

Select the candidate with the similarity above selected threshold

More correspondences and Robust matching

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

RANSAC in action

- Inliers
- Outliers

Epipolar Geometry

- Epipolar geometry in two views
- Refined epipolar geometry using nonlinear estimation of \( F \)
Two view initialization

- Recover epipolar geometry
  \[ s^2 Fx_1 = 0 \quad F = X X^T \]
  calibrated $X = I$
- Compute (Euclidean) projection matrices and 3-D struct.
  \[ n_1 = [F, 0, 0] \quad n_2 = [R, 0, 0] \]
  \[ \lambda_1 X_1 = X \quad \lambda_2 X_2 = R X + \gamma T \]

uncalibrated $X$ unknown
- Compute (Projective) projection matrices and 3-D struct.
  \[ n_{1p} = [F, 0, 0] \quad n_{2p} = [F, 0, 0] \]
  \[ \lambda_1 X_{1p} = X_{1p} \quad \lambda_2 X_{2p} = X_{2p} + \gamma T \]

3-D reconstruction from two views

Multi-view reconstruction

- Two view - initialized motion and structure estimates (scales)
- Multi-view factorization - recover the remaining camera positions and refine the 3-D structure by iteratively computing

Repeat until convergence
1. Compute i-th motion given the known structure

\[ R \begin{bmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_i^T \\ \cdots \\ x_n^T \end{bmatrix} = 0 \]

\[ \alpha \delta \approx \frac{1}{2^n} \]

iteration

2. Refine structure estimate given all the motions

\[ \alpha_{i+1} = \frac{\Sigma x_{i+1}}{\Sigma x_i} \cdot \frac{1}{\delta_i} \quad i = 1, \ldots, n \]

Projective Ambiguity

- calibrated case - projection matrices – and Euclidean structure
  \[ \lambda_1 X = \Pi_{1p} X = [P, Y] \]
- Uncalibrated case
  \[ \lambda_1 X = \Pi_{1p} X \]
- for i-th frame
  \[ \Pi_{i} \cdot X_{i} \sim \Pi_{i} \cdot X_{i-1} \]
- for 1st frame
  \[ \lambda_1 X_1 = \Pi_{1p} X_1 = \Pi_{1p} X = [K, X] \]
- Transform projection matrices and 3-D structure by H
  \[ H^{-1} = \begin{bmatrix} \Sigma X_i \end{bmatrix}^{-1} \quad O \]

CS 682
**Upgrade to Euclidean Reconstruction**

- From Projective to Euclidean Reconstruction
  \[ \Pi_{\text{proj}} \sim \Pi_{\text{proj}} \quad \mathbb{K}_0 \sim R^{-1} \mathbb{K}_p \]
- Remove the ambiguity characterized by
  \[ H \sim \begin{bmatrix} K_1 \quad 0 \\ 0 \quad K_2 \end{bmatrix} \quad \Pi_{\text{proj}} \begin{bmatrix} K_3 \\ 0 \quad 0 \quad 1 \end{bmatrix} \sim \kappa \Pi_{\text{proj}} \]
- How to estimate \( H \)? - Absolute quadric constraint
  \[ \Pi_{\text{proj}} \begin{bmatrix} K_1 \quad 0 \\ 0 \quad K_2 \quad 0 \quad 1 \end{bmatrix} \quad \Pi_{\text{proj}} \begin{bmatrix} K_3 \\ 0 \quad 0 \quad 1 \end{bmatrix} \sim \kappa_0 \Pi_{\text{proj}} \]
- \( \mathbb{Q}_0, \mathbb{S}_0 \) be recovered by a nonlinear minimization

---

**Example of multi-view reconstruction**

- Projective Reconstruction
- Euclidean Reconstruction

---

**Nonlinear Refinement**

- Euclidean Bundle adjustment
- Initial estimates of \( \mathbb{K}_e, \mathbb{D}_e, \mathbb{T}_e, \mathbb{X}_e \) are available
- Final refinement, nonlinear minimization with respect to all unknowns
  \[ e_{\text{ref}} = \frac{1}{\text{num} \times \text{num}} \sum_{i=1}^{\text{num}} \sum_{j=1}^{\text{num}} \| \mathbb{X}_i - \mathbb{K}_i \mathbb{H}_i \mathbb{T}_i \mathbb{X}_j \|^2 \]
Example - Euclidean multi-view reconstruction

Recovered model

Euclidean Reconstruction
Epipolar rectification

- Make the epipolar lines parallel
- Dense correspondences along image scanlines
- Computation of warping homographies $H_1, H_2$

1. Map the epipole $E$ to infinity $H_E = (1,0,0)^T$
   - Translate the image center to the origin
   - Rotate around z-axis for the epipole lie on the x-axis
   - Transform the epipole from x-axis to infinity $H_3 = C x_C F x_E$

2. Find a matching transformation $H_1 = H_2 H_3$
   - $H_2$ is compatible with the epipolar geometry $H = x_E F + x_E x_E^T$
   - $H_3$ is chosen to minimize overall disparity

$$\tilde{x}_E^T H_1 x_F = \tilde{x}_E^T H_2 x_F + \tilde{x}_E^T x_E x_E^T x_F = 0$$

---

Dense Matching

- Establish dense correspondences along scanlines
- Standard stereo configuration
- Constraints to guide the search
  1. ordering constraint
  2. disparity constraint – limit on disparity
  3. uniqueness constraint – each point has a unique match in the second view
Dense Matching

Texture mapping, hole filling

Dense Reconstruction

Texture mapping