PS2 Review

CS231A

Computer Vision: From 3D Reconstruction to Recognition

Spring 2016

Outline

Q2: Estimating Fundamental Matrix

Q3: Affine Structure From Motion

Q4: Projective Triangulation in Structure From Motion

Least Squares Eight Point Algorithm

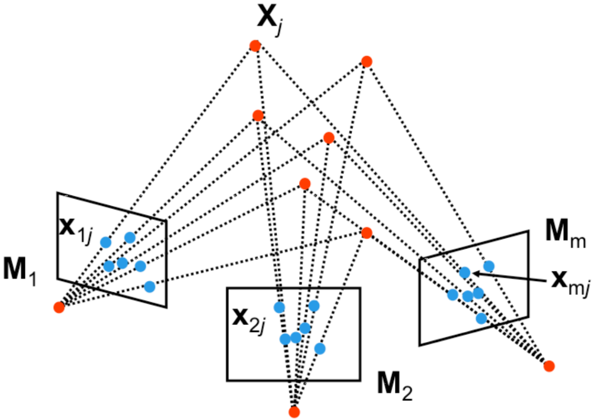
- Recall $p^T F p' = 0$
- Construct W matrix such that $W\hat{F} = 0$, where $\hat{F}$ is F arranged as a vector
- Run SVD on W, such that $W = U\Sigma V^\top$
- $\hat{F}$ is equal to the last column of V.
- Rearrange $\hat{F}$ into it's matrix form (now F_m will be this matrix)
- Find the closest rank 2 matrix to F_m. Do this by running SVD on F_m, such that $F_m = U\Sigma V^\top$
- Set the smallest singular value to zero ($\Sigma_{3,3} = 0$) and we find $F = U\Sigma V^\top$

Problems with Eight Point Algorithm

- W matrix is highly unbalanced and can ruin SVD computation
- Each entry in W should be around the same order of magnitude
- **Solution:** Need to normalize each of the correspondences and then denormalize returned F

Outline

Overall Structure From Motion Problem



- Given n 3D points and m images:
  - Motion: the projective matrices M_j
  - Structure: the 3D locations X_i

Affine Structure From Motion

- Simpler problem if we assume cameras are affine:

$$M = \begin{bmatrix} A_{2\times 3} & b_{2\times 1} \\ 0 & 1 \end{bmatrix}$$

$$x_{ij} = A_i X_j + b_i$$

- **Problem:** Find all A_i, X_j and b_i given correspondences $x_i j$.
- Unknowns: $3n$ from n 3D points and $8m$ from m camera parameters. In total, $3n + 8m$ variables
- Equations: $m \times n$ correspondences. Each correspondence gives 2 constraints. In total, $2mn$ constraints

Factorization Method Part 1

- Center the data:

$$\begin{aligned}\hat{x}_{ij} &= x_{ij} - \frac{1}{n}\sum_{k=1}^{n} x_{ik} \\ &= A_i X_j + b_i - \frac{1}{n}\sum_{k=1}^{n} A_i X_k - \frac{1}{n}\sum_{k=1}^{n} b_i \\ &= A_i (X_j - \frac{1}{n}\sum_{k=1}^{n} X_k)\end{aligned}$$

- Place world axes at centroid of 3D locations:

$$\sum_{k=1}^{n} X_k = 0$$

$$\hat{x}_{ij} = A_i X_j$$

Factorization Method Part 2

- Setup equation as follows:

$$\begin{bmatrix} \hat{x_{11}} & \hat{x_{12}} & \dots & \hat{x_{1n}} \\ \hat{x_{21}} & \hat{x_{22}} & \dots & \hat{x_{2n}} \\ & & \ddots & \\ \hat{x_{m1}} & \hat{x_{m2}} & \dots & \hat{x_{mn}} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}$$

$$D = MS$$

$$D = U\Sigma V^T$$

$$D = U_3\Sigma_3 V_3^T$$

- Thus, $M = U_3$ and $S = \Sigma_3 V_3^T$

Outline

Computing R, T From Essential Matrix E

- Recall $E = [T]_\times R$
- $[T]_\times$ is a skew-symmetric matrix and R is orthogonal
- Let $Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Note that $Z = \mathrm{diag}(1,1,0)W$ or $\mathrm{diag}(1,1,0)W^T$ up to sign
- Skew-symmetric matrices can be written as $kUZU^T$ for orthogonal matrix U and coefficient k
- Using SVD, $E = U\Sigma V^T$, with $\Sigma = \mathrm{diag}(1,1,0)$

Computing R, T From Essential Matrix E

- If $E = U\text{diag}(1,1,0)V^T$, then

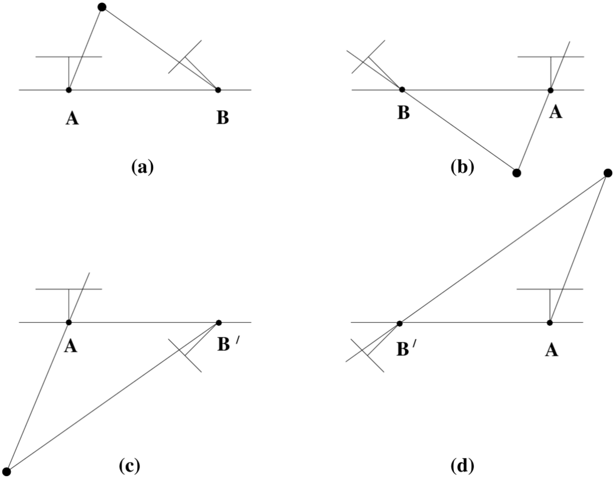
$$E = UZU^TUWV^T$$

or

$$E = UZU^TUW^TV^T$$

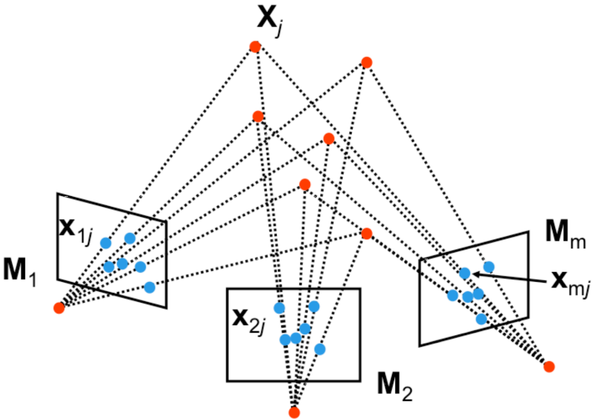
- Thus $[T]_\times = UZU^T$ and $R = UWV^T$ or UW^TV^T
- Since Z is known up to sign, $T = u_3$ or $-u_3$ where u_3 is the 3rd column vector of U

Disambiguating R, T



- Logically, what you're seeing has to be in front of both cameras

Relating Triangulation to Structure From Motion Problem



- Triangulation: Given motion M and image correspondences x, compute structure X

Linear Triangulation

- Want to find the least squares solution to triangulation
- Given $x_{ij} = M_i X_j$, then $x_{ij} \times M_i X_j = 0$
- Setup with form $AX_j = 0$

$$
\begin{aligned}
x_{ij} \times M_i X_j &= 0 \\
[x_{ij}]_\times M_i X_j &= 0
\end{aligned}
$$

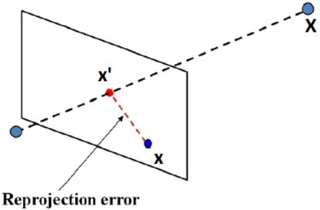
- Thus $A = [x_{ij}]_\times M_i$
- Because of noise, we minimize $\|AX_j\|$ subject to $\|X_j\| = 1$.
- Solved using SVD: If $A = U\Sigma V^T$, then X_j is the last column of V (proof in HZ 592)
- Use homogeneous coordinates to circumvent $\|X_j\| = 1$ constraint

Nonlinear Optimization

- Can also optimize finding structure using some objective function
- Common one is the reprojection error for image location x:

$$\text{error} = x' - x$$

where x' when you project 3D location X into some image.



- Newton step - covered in class prerequisites