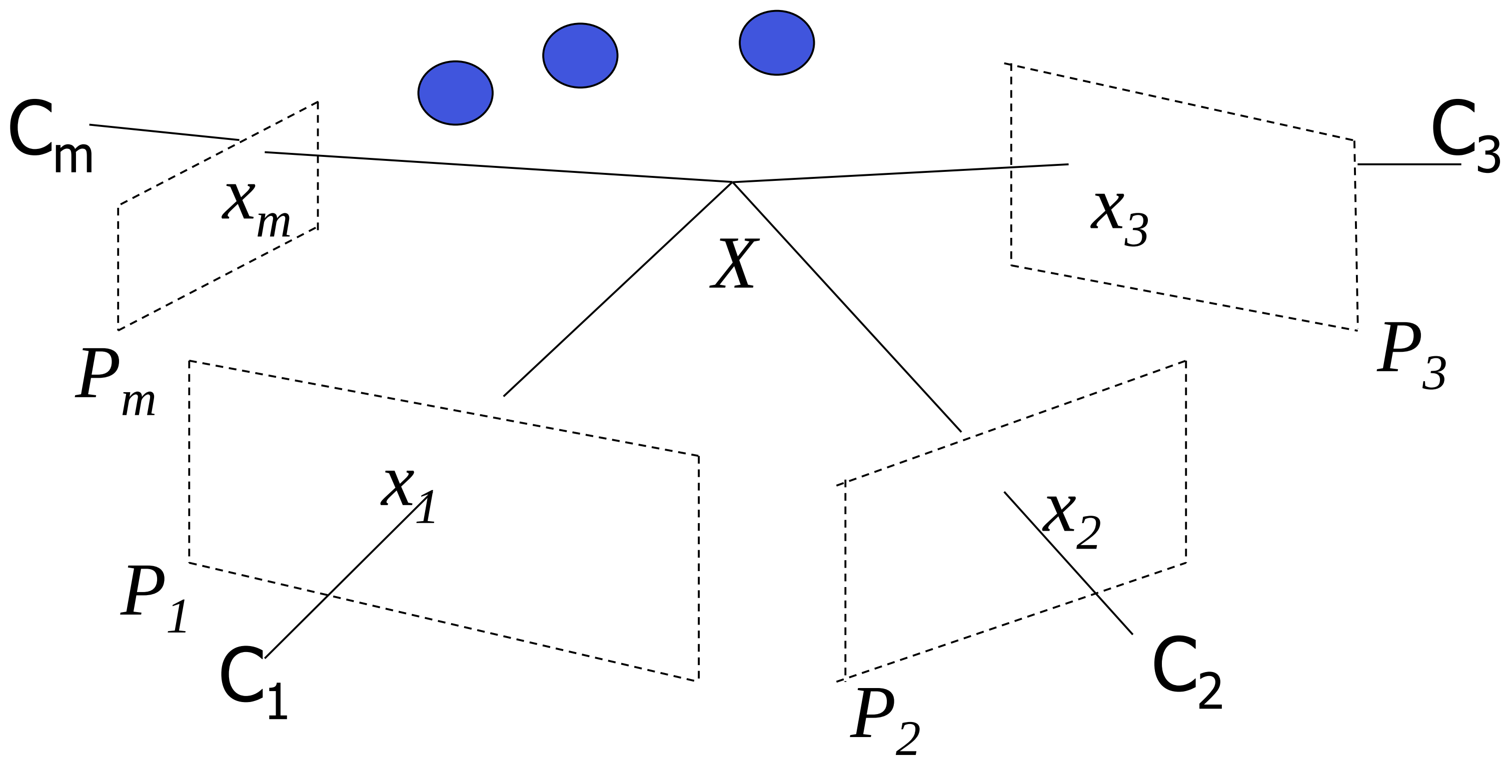
Multiview Geometry

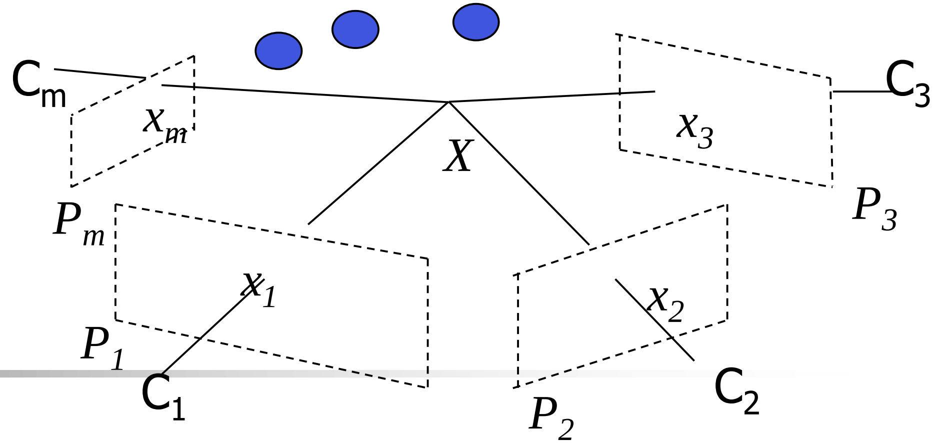
Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Multiview Geometry



Given corresponding m images ($\{x_{ij}\}$) for n scene points $\{X_j\}$'s, estimate P_i's and X_j's.

Bundle Adjustment

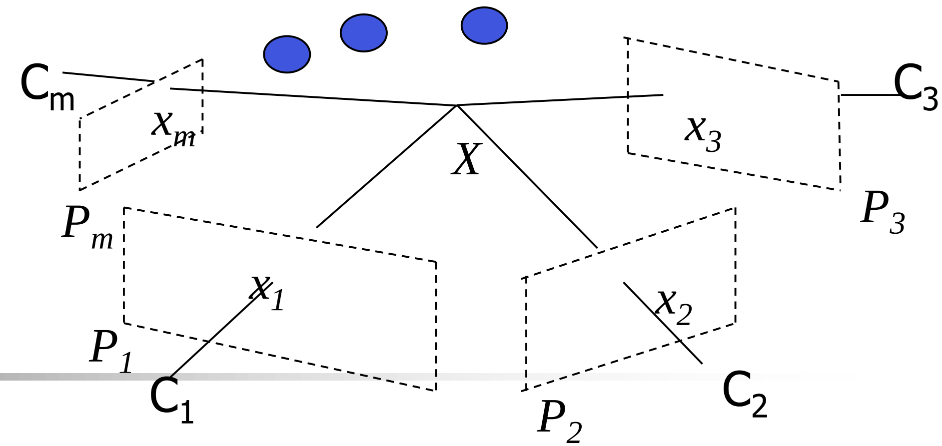


Adjustment of bundle of rays going through each camera center to the set of 3D points.

$$\min_{\{\widehat{P_i}\},\{\widehat{X_j}\}} \sum_{i,j} d\left(\widehat{P}_i \widehat{X}_j, x_{ij}\right)^2$$

- Tolerant to missing data.
- Requires a good initialization.
- For n points and m views → $3n+11m$ unknowns
- With over-parameterization: → $3n+12m$
- Reduce n and / or m by solving on a subset and merging solutions.
- Interleave of estimates: Alternate minimizing reprojection error by varying P_i's and X_j's.

Alternate Minimization



Adjustment of bundle of rays going through each camera center to the set of 3D points.

$$\min_{\{\widehat{P_i}\},\{\widehat{X_j}\}} \sum_{i,j} d\left(\widehat{P}_i \widehat{X}_j, x_{ij}\right)^2$$

- Form an initial set of scene points $\{X_j\}$, $j=1,2,..n$.
- Given $\{(X_j, x_{kj}), j=1,2,..n\}$ for k th camera estimate P_k using DLT or any NL optimization technique.
- Given $\{P_k, k=1,2,..m\}$ estimate $\{X_j\text{'s}\}$ by forming equations:

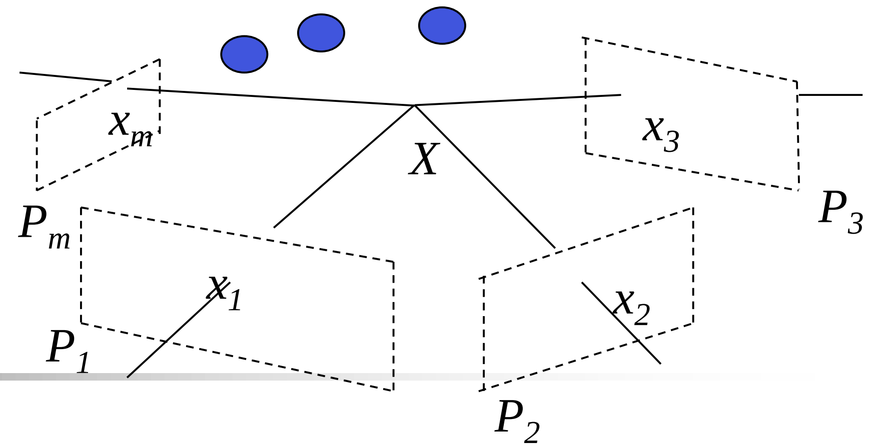
$$x_{kj} \times P_k X_j = 0$$

- Solve the above using DLT or other methods.

Methods for initial solution:

- For affine cameras: Factorization
- For projective cameras: Iterative factorization

Affine Reconstruction



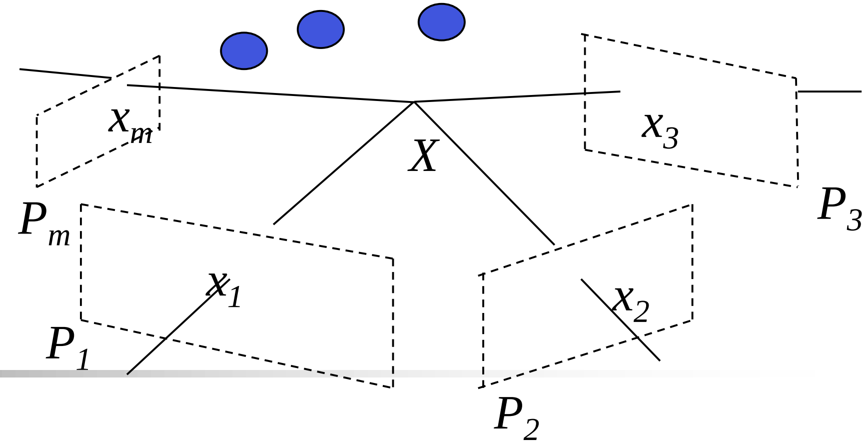
$$\min_{\{M_i,t_i\},\{X_j\}} \sum_{i,j} \left\| \boldsymbol{x}_{ij} - \widehat{\boldsymbol{x}_{ij}} \right\|^2 \qquad \hat{\boldsymbol{x}} = \begin{bmatrix} x \\ y \end{bmatrix} = M_{2\times 3} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \boldsymbol{t}$$

$$\rightarrow \min_{\{M_i,t_i\},\{X_j\}} \sum_{i,j} \left\| \boldsymbol{x}_{ij} - \left(M_i X_j + \boldsymbol{t}_i \right) \right\|^2$$

In affine projection,
centroid of points in 3D → Centroid of the projections.

Translate every point in every view such that centroid is $(0,0)^T$ in every view. Centroid in 3D is $(0,0,0)^T$ and $\mathbf{t_i}$'s are $(0,0)^T$. → $x'_{ij} = x_{ij} - \langle x_{ij} \rangle$

Affine Reconstruction



$$\min_{\{M_i,t_i\},\{X_j\}} \sum_{i,j} \left\| \boldsymbol{x}_{ij} - (M_i X_j + \boldsymbol{t}_i) \right\|^2$$

$$\frac{\partial \sum_{i,j} \left\| \boldsymbol{x}_{ij} - (M_i X_j + \boldsymbol{t}_i) \right\|^2}{\partial \boldsymbol{t}_i} = 0$$

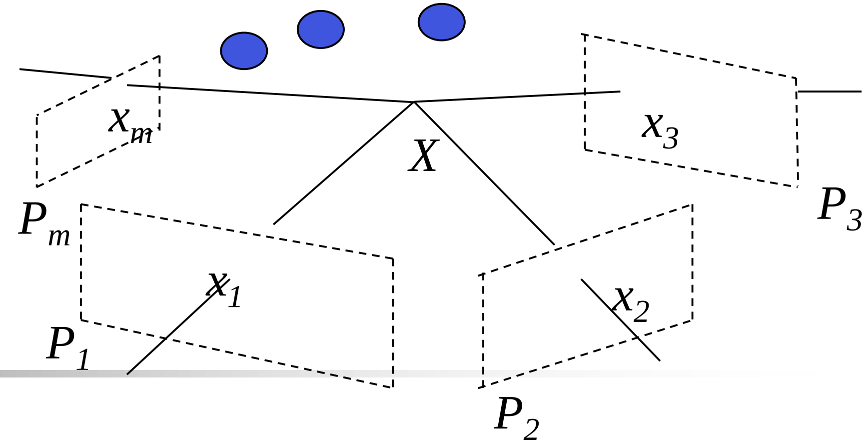
$(0,0,0)^T$

$$\boldsymbol{t}_i = \langle \boldsymbol{x}_{ij} \rangle - M_i \langle X_j \rangle$$
$$= \langle \boldsymbol{x}_{ij} \rangle$$

$\boldsymbol{x}'_{ij} = \boldsymbol{x}_{ij} - \langle \boldsymbol{x}_{ij} \rangle$ ➔ $$\min_{\{M_i\},\{X_j\}} \sum_{i,j} \left\| \boldsymbol{x}'_{ij} - (M_i X_j) \right\|^2$$

Factorize data

Factorization of data



$$\min_{\{M_i\},\{X_j\}} \sum_{i,j} \left\| \boldsymbol{x}'_{ij} - (M_i X_j) \right\|^2$$

$$\boldsymbol{W}_{2m\times n} = \begin{bmatrix} \boldsymbol{x}_{11} & \boldsymbol{x}_{12} & \cdots & \boldsymbol{x}_{1n} \\ \boldsymbol{x}_{21} & \boldsymbol{x}_{22} & \cdots & \boldsymbol{x}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}_{m1} & \boldsymbol{x}_{m2} & \cdots & \boldsymbol{x}_{mn} \end{bmatrix}$$

$$\boldsymbol{M}_{2m\times 3} = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_m \end{bmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$$

$$D = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}$$

$$\boldsymbol{X}_{3\times n} = [X_1 \quad X_2 \quad \cdots \quad X_n]$$

Factorize

$$\boldsymbol{W} = \boldsymbol{M}\boldsymbol{X}$$

$$\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$$

Estimates

$$\boldsymbol{W} = U_{2m\times n} D_{n\times n} V_{n\times n}^T$$

SVD

$$[u_1 \quad u_2 \quad \cdots \quad u_n]$$

$$\widehat{\boldsymbol{M}} = [\sigma_1 u_1 \quad \sigma_2 u_2 \quad \sigma_3 u_3]$$

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$$

Affine ambiguity and Euclidean upgrade

Any 3x3 non-singular matrix.

$$\boldsymbol{W} = \boldsymbol{M}\boldsymbol{X} \rightarrow \boldsymbol{W} = \underbrace{\boldsymbol{M}Q}_{\text{Affine camera matrices}}\underbrace{Q^{-1}\boldsymbol{X}}_{\text{3D points}}$$

3D points

Affine camera matrices

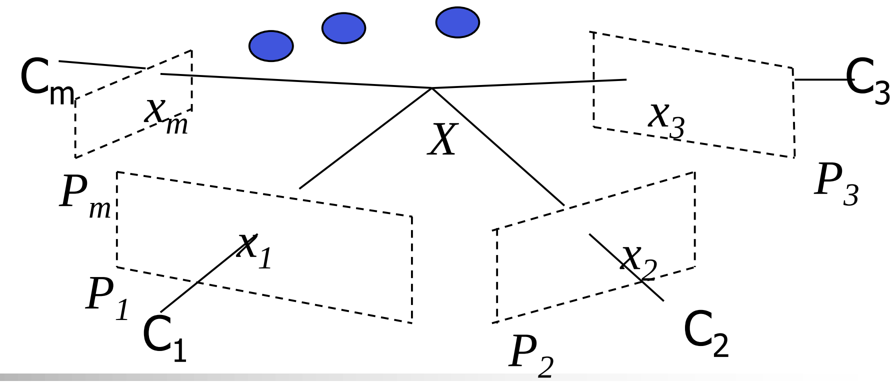
$$Let\ M_i = \begin{bmatrix} [\boldsymbol{a}_{i1}]_{1\times 3}^T \\ [\boldsymbol{a}_{i2}]_{1\times 3}^T \end{bmatrix} \rightarrow M_i Q = \begin{bmatrix} \boldsymbol{a}_{i1}^T Q \\ \boldsymbol{a}_{i2}^T Q \end{bmatrix}$$

For every i th ($i=1,2,..m$) camera, orthographic constraints:

$$\boldsymbol{a}_{i1}^T Q Q^T \boldsymbol{a}_{i2} = 0$$
$$\boldsymbol{a}_{i1}^T Q Q^T \boldsymbol{a}_{i1} = 1$$
$$\boldsymbol{a}_{i2}^T Q Q^T \boldsymbol{a}_{i2} = 1$$

- Solve for $S=QQ^T$ using LSE.
- Get Q using Cholesky's decompisition.
- Q is obtained upto arbitrary rotation.

Projective factorization



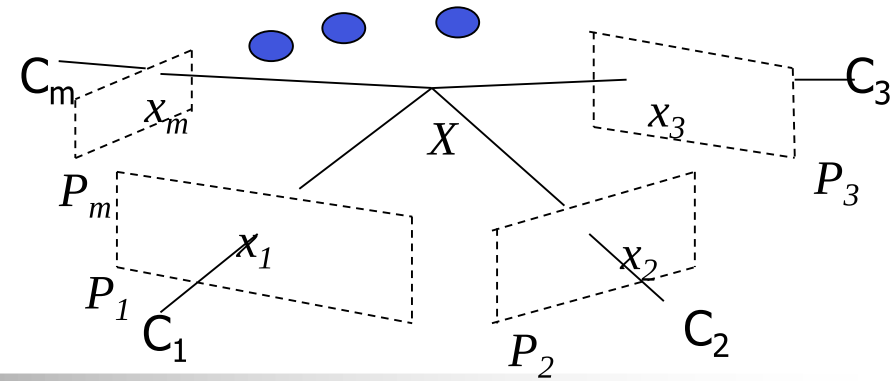
Projective depth factor

$$\boldsymbol{x}_{ij} \equiv P_i \boldsymbol{X}_j \quad \rightarrow \quad \lambda_{ij}\boldsymbol{x}_{ij} = P_i \boldsymbol{X}_j$$

$$\boldsymbol{W}_{2m\times n} = \begin{bmatrix} \lambda_{11}\boldsymbol{x}_{11} & \lambda_{12}\boldsymbol{x}_{12} & \cdots & \lambda_{1n}\boldsymbol{x}_{1n} \\ \lambda_{21}\boldsymbol{x}_{21} & \lambda_{22}\boldsymbol{x}_{22} & \dots & \lambda_{2n}\boldsymbol{x}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{m1}\boldsymbol{x}_{m1} & \lambda_{m2}\boldsymbol{x}_{m2} & \dots & \lambda_{mn}\boldsymbol{x}_{mn} \end{bmatrix}$$

$$\boldsymbol{P}_{3m\times 4} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \qquad \boldsymbol{X}_{4\times n} = [X_1 \quad X_2 \quad \dots \quad X_n] \qquad \boldsymbol{W} = \boldsymbol{P}\boldsymbol{X}$$

Iterative factorization starting
With a set of initial depth factors.



The Algorithm

$$\boldsymbol{W}_{2m\times n} = \begin{bmatrix} \lambda_{11}\boldsymbol{x}_{11} & \lambda_{12}\boldsymbol{x}_{12} & \cdots & \lambda_{1n}\boldsymbol{x}_{1n} \\ \lambda_{21}\boldsymbol{x}_{21} & \lambda_{22}\boldsymbol{x}_{22} & \cdots & \lambda_{2n}\boldsymbol{x}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{m1}\boldsymbol{x}_{m1} & \lambda_{m2}\boldsymbol{x}_{m2} & \cdots & \lambda_{mn}\boldsymbol{x}_{mn} \end{bmatrix} \quad \boldsymbol{P}_{3m\times 4} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}$$

$$\boldsymbol{X}_{4\times n} = [X_1 \quad X_2 \quad \cdots \quad X_n] \qquad \boldsymbol{W} = \boldsymbol{P}\boldsymbol{X}$$

1. Choose initial values of λs.
2. Solve for $\boldsymbol{P}$ and $\boldsymbol{X}$ using SVD method as before.
3. Recompute λs.
4. Iterate till it converges.

- Normalize data before processing, and apply inverse transformation after getting the result.
- Normalize columns and rows of λ to make them unit norm.

Euclidean rectification

Any 4x4 non-singular matrix

$$\boldsymbol{W} = \boldsymbol{PX} \;\rightarrow\; \boldsymbol{W} = \underbrace{\boldsymbol{P}Q}_{\text{Camera matrices}}\underbrace{Q^{-1}\boldsymbol{X}}_{\text{3D points in homogeneous coordinates}}$$

$$Let\ Q = [[Q_3]_{4\times3} \quad [q_4]_{4\times1}]$$

$$P_i = \begin{bmatrix} p_{i1}^T \\ p_{i2}^T \\ p_{i3}^T \end{bmatrix} \quad P_iQ = \begin{bmatrix} p_{i1}^T Q_3 & p_{i1}^T q_4 \\ p_{i2}^T Q_3 & p_{i2}^T q_4 \\ \underbrace{p_{i3}^T Q_3}_{\text{Needs to be orthogonal.}} & p_{i3}^T q_4 \end{bmatrix}$$

$$p_{i1}^T Q_3 Q_3^T p_{i2} = 0$$
$$p_{i1}^T Q_3 Q_3^T p_{i3} = 0$$
$$p_{i3}^T Q_3 Q_3^T p_{i2} = 0$$
$$p_{i1}^T Q_3 Q_3^T p_{i1} = p_{i2}^T Q_3 Q_3^T p_{i2}$$

Solve for Q.

Solve for Q_3 and q_4

$$p_{i1}^T Q_3 Q_3^T p_{i2} = 0$$
$$p_{i1}^T Q_3 Q_3^T p_{i3} = 0$$
$$p_{i3}^T Q_3 Q_3^T p_{i2} = 0$$

$$p_{i1}^T Q_3 Q_3^T p_{i1} = p_{i2}^T Q_3 Q_3^T p_{i2} = p_{i3}^T Q_3 Q_3^T p_{i3}$$

$$A = Q_3 Q_3^T \qquad Q_3 = U\sqrt{D}$$

SVD

Solve for A. $\quad A = UDV^T$

q_4 can be determined by (arbitrarily) picking the origin of the frame attached to the 1st camera as the origin of the world coordinates.

Thank you!