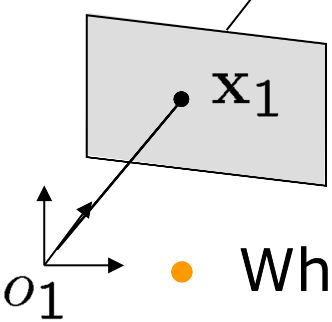
Geometry and Algebra of Multiple Views

Yi Ma
EECS, University of California, Berkeley

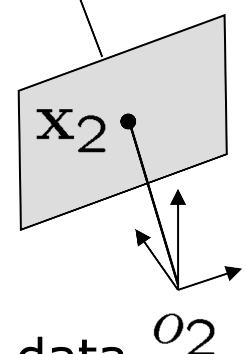
"Algebra is but written geometry; geometry is but drawn algebra."

– Sophie Germain

Two View Geometry



- From two views
  - Can recover motion: 8-point algorithm
  - Can recover structure: triangulation



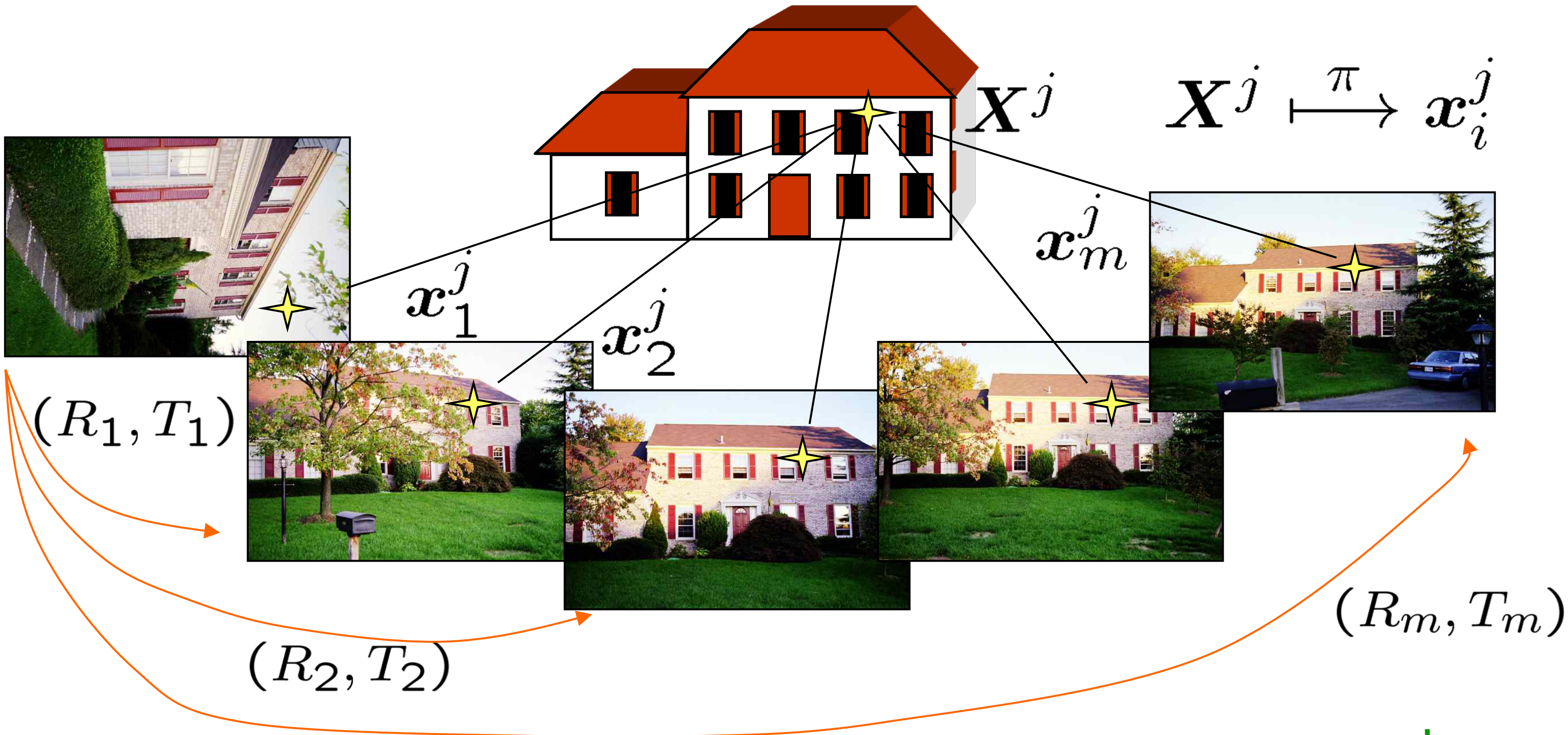
- Why multiple views?
  - Get more robust estimate of structure: more data
  - No direct improvement for motion: more data & unknowns

Why multiple views?

- Cases that cannot be solved with two views
  - Uncalibrated case: need at least three views
  - Line features: need at least three views
- Some practical problems with using two views
  - Small baseline: good tracking, poor motion estimates
  - Wide baseline: good motion estimates, poor correspondences
- With multiple views one can
  - Track at high frame rate: tracking is easier
  - Estimate motion at low frame rate: throw away data

Problem Formulation

Input: Corresponding images (of "features") in multiple images.
Output: Camera motion, camera calibration, object structure.



measurements

$$F(R_i, T_i, \boldsymbol{X}^j) = \sum_{j=1}^{n} \sum_{i=1}^{m} \| \boldsymbol{x}_i^j - \pi(\underbrace{R_i \boldsymbol{X}^j + T_i}_{\text{unknowns}}) \|^2$$

Motivating Examples

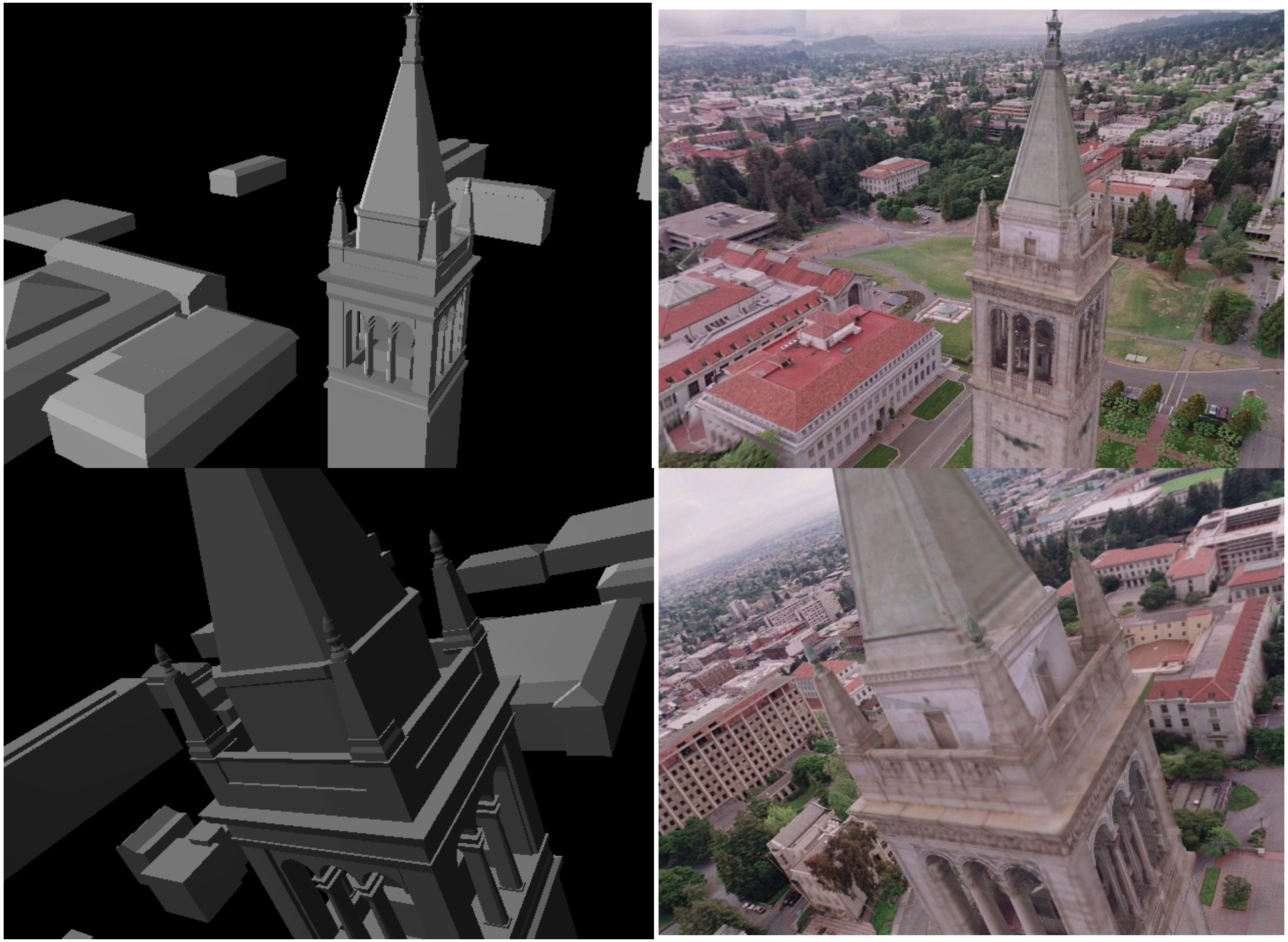


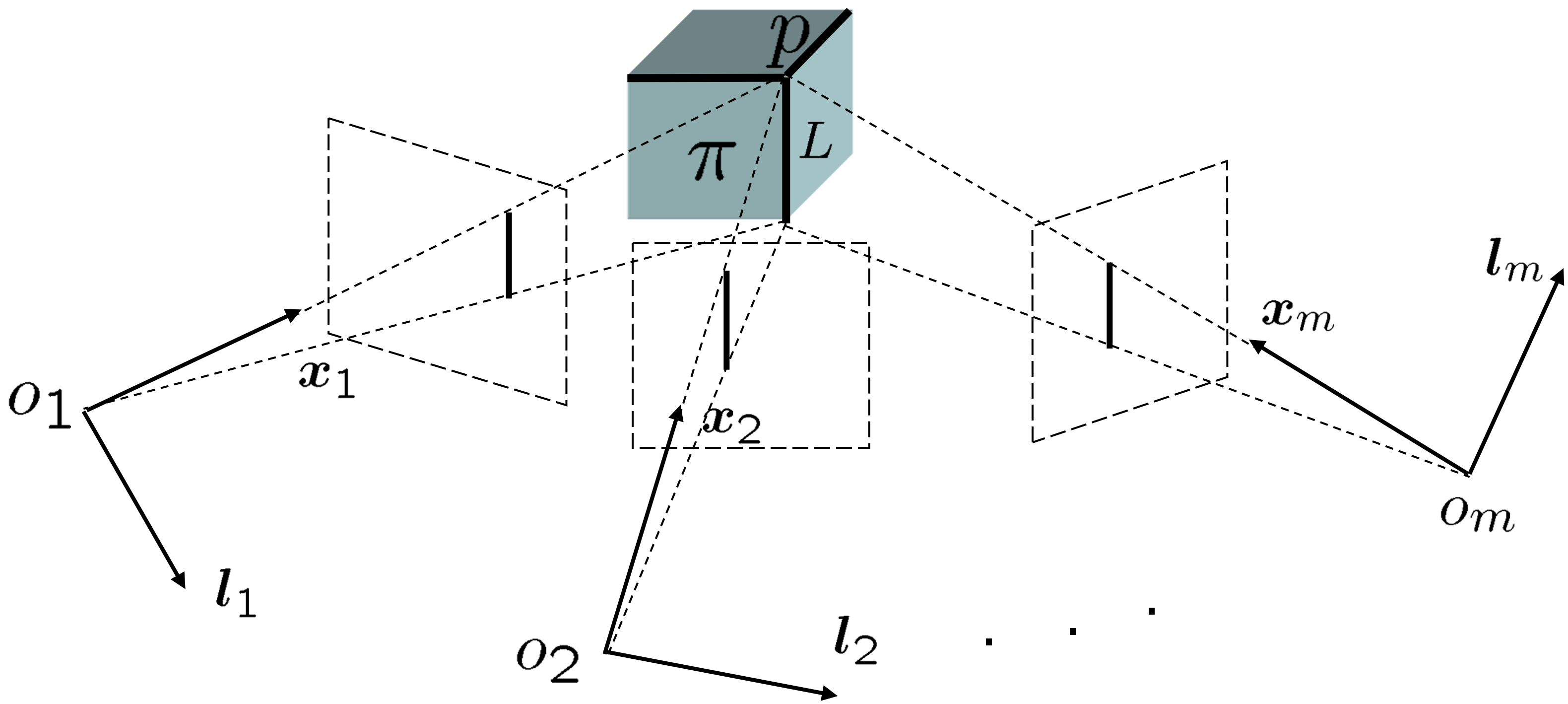
Image courtesy of Paul Debevec

Using geometry to tackle the optimization problem

- What are the basic relations among multiple images of a point/line?
  - Geometry and algebra
- How can I use all the images to reconstruct camera pose and scene structure?
  - Algorithm
- Examples
  - Synthetic data
  - Vision based landing of unmanned aerial vehicles

Incidence Relations among Features

"Pre-images" are all incident at the corresponding features



Projection: Point Features

Homogeneous coordinates of a 3-D point p

$$\mathbf{X} = [X, Y, Z, W]^T \in \mathbb{R}^4, \quad (W = 1)$$

Homogeneous coordinates of its 2-D image

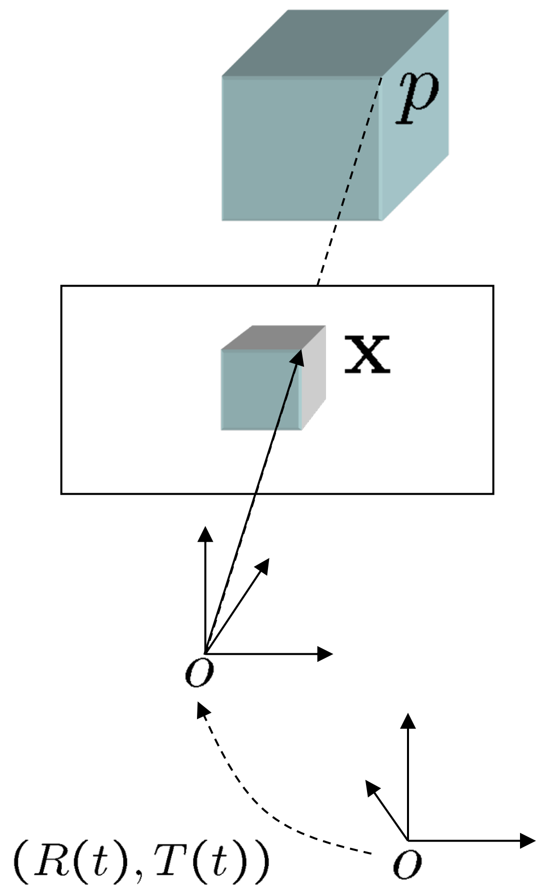
$$\boldsymbol{x} = [x, y, z]^T \in \mathbb{R}^3, \quad (z = 1)$$

Projection of a 3-D point to an image plane

$$\boxed{\lambda(t)\boldsymbol{x}(t) = \Pi(t)\boldsymbol{X}}$$

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

$$R(t) \rightarrow A(t)R(t),\ T(t) \rightarrow A(t)T(t)$$



Multiple View Matrix for Point Features

- WLOG choose frame 1 as reference

$$\boldsymbol{x}_i \times (\lambda_i \boldsymbol{x}_i = \lambda_1 R_i \boldsymbol{x}_1 + T_i) \Longleftrightarrow \begin{bmatrix} \widehat{\boldsymbol{x}_i} R_i \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_i} T_i \end{bmatrix} \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} = 0$$

$$\boxed{M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}}$$

- Rank deficiency of Multiple View Matrix

$$\boxed{\text{rank}(M_p) \leq 1}$$

$\text{rank}(M_p) = 1$ (generic)

$\text{rank}(M_p) = 0$ (degenerate)

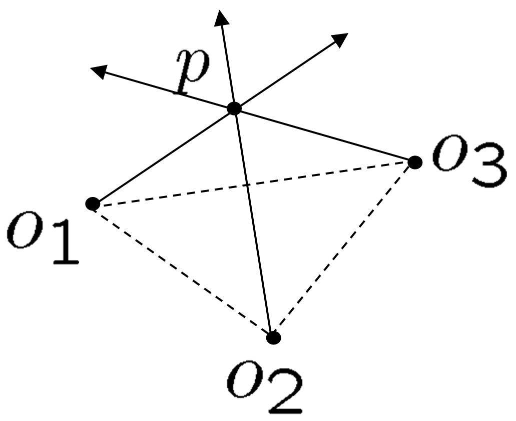
Multiple View Matrix for Point Features

$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}$$

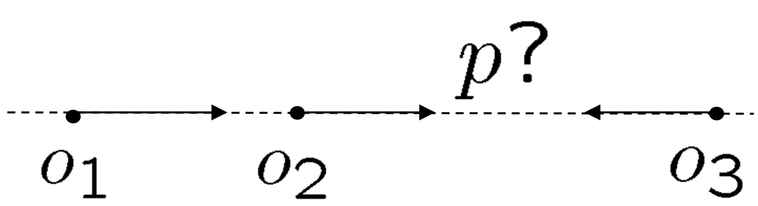
$$0 \leq \mathrm{rank}(M_p) \leq 1$$

M_p encodes exactly the 3-D information missing in one image.

$\mathrm{rank}(M_p) = 1$



$\mathrm{rank}(M_p) = 0$



Rank Conditions vs. Multifocal Tensors

$$\text{rank}\begin{bmatrix} \widehat{\boldsymbol{x}_2}R_2\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2}T_2 \\ \boxed{\widehat{\boldsymbol{x}_3}R_3\boldsymbol{x}_1 \quad \widehat{\boldsymbol{x}_3}T_3} & \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m}R_m\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m}T_m \end{bmatrix} = 1 \;\Rightarrow\; \boldsymbol{x}_i^T\widehat{T_i}R_i\boldsymbol{x}_1 = 0$$

Two views: epipolar constraint

$$\text{rank}\begin{bmatrix} \boxed{\begin{matrix}\widehat{\boldsymbol{x}_2}R_2\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2}T_2 \\ \widehat{\boldsymbol{x}_3}R_3\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3}T_3\end{matrix}} \\ \begin{matrix}\vdots & \vdots \\ \widehat{\boldsymbol{x}_m}R_m\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m}T_m\end{matrix} \end{bmatrix} = 1 \;\Rightarrow\; \widehat{\boldsymbol{x}_i}(R_i\boldsymbol{x}_1T_j^T - T_i\boldsymbol{x}_1^TR_j^T)\widehat{\boldsymbol{x}_j} = 0$$

Three views: trilinear constraints

- Other relationships among four or more views, e.g. quadrilinear constraints, are algebraically dependent!

Traditional Multifocal or Multilinear Constraints

- Given m corresponding images of n points $\mathbf{x}_i^j$

$$\lambda_i^j \boldsymbol{x}_i^j = \Pi_i \boldsymbol{X}^j \qquad \Pi_i = [R_i, T_i]$$

- This set of equations is equivalent to

Heyden et.al.

$$\underbrace{\begin{bmatrix} \Pi_1 & \boldsymbol{x}_1 & 0 & \cdots & 0 \\ \Pi_2 & 0 & \boldsymbol{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Pi_m & 0 & \cdots & 0 & \boldsymbol{x}_m \end{bmatrix}}_{H_p \in \mathbb{R}^{3m \times (m+4)}} \begin{bmatrix} \boldsymbol{X} \\ -\lambda_1 \\ \vdots \\ -\lambda_m \end{bmatrix} = 0$$

$$\det\left(H_{(m+4)\times(m+4)}\right) = 0$$

Faugeras et.al.

$$\underbrace{\begin{bmatrix} \widehat{\boldsymbol{x}_1}\Pi_1 \\ \widehat{\boldsymbol{x}_2}\Pi_2 \\ \vdots \\ \widehat{\boldsymbol{x}_m}\Pi_m \end{bmatrix}}_{F_p \in \mathbb{R}^{3m \times 4}} \boldsymbol{X} = 0$$

$$\det\left(F_{4\times 4}\right) = 0$$

- Multilinear constraints among 2, 3, 4 views

Reconstruction Algorithm for Point Features

- Given m images of n points $\{(\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_i^j, \ldots, \boldsymbol{x}_m^j)\}_{j=1,\ldots,n}^{i=1,\ldots,m}$

$$\lambda^j \begin{bmatrix} \widehat{\boldsymbol{x}}_2^j R_2 \boldsymbol{x}_1^j \\ \widehat{\boldsymbol{x}}_3^j R_3 \boldsymbol{x}_1^j \\ \vdots \\ \widehat{\boldsymbol{x}}_m^j R_m \boldsymbol{x}_1^j \end{bmatrix} + \begin{bmatrix} \widehat{\boldsymbol{x}}_2^j T_2 \\ \widehat{\boldsymbol{x}}_3^j T_3 \\ \vdots \\ \widehat{\boldsymbol{x}}_m^j T_m \end{bmatrix} = 0 \qquad \in \mathbb{R}^{3(m-1)\times 1}$$

$$P_i \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = \begin{bmatrix} \lambda^1 \boldsymbol{x}_1^1 \otimes \widehat{\boldsymbol{x}_i^1} & \widehat{\boldsymbol{x}_i^1} \\ \lambda^2 \boldsymbol{x}_1^2 \otimes \widehat{\boldsymbol{x}_i^2} & \widehat{\boldsymbol{x}_i^2} \\ \vdots & \vdots \\ \lambda^n \boldsymbol{x}_1^n \otimes \widehat{\boldsymbol{x}_i^n} & \widehat{\boldsymbol{x}_i^n} \end{bmatrix} \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = 0 \qquad \in \mathbb{R}^{3n\times 1}$$

$$\boxed{\text{If } n \geq 6, \text{ in general } \operatorname{rank}(P_i) = 11}$$

Reconstruction Algorithm for Point Features

Given *m* images of *n* (>6) points

For the *j*th point

$$\begin{bmatrix} \widehat{\boldsymbol{x}_2^j} R_2 \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_2^j} T_2 \\ \widehat{\boldsymbol{x}_3^j} R_3 \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_3^j} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m^j} R_m \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_m^j} T_m \end{bmatrix} \begin{bmatrix} \lambda^j \\ 1 \end{bmatrix} = 0 \overset{\text{SVD}}{\Rightarrow} \lambda^{j^s}$$

Iteration

For the *i*th image

$$\begin{bmatrix} \lambda^1 \boldsymbol{x}_1^1 \otimes \widehat{\boldsymbol{x}_i^1} & \widehat{\boldsymbol{x}_i^1} \\ \lambda^2 \boldsymbol{x}_1^2 \otimes \widehat{\boldsymbol{x}_i^2} & \widehat{\boldsymbol{x}_i^2} \\ \vdots & \vdots \\ \lambda^n \boldsymbol{x}_1^n \otimes \widehat{\boldsymbol{x}_i^n} & \widehat{\boldsymbol{x}_i^n} \end{bmatrix} \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = 0 \overset{\text{SVD}}{\Rightarrow} (R_i^s, T_i^s)$$

Reconstruction Algorithm for Point Features

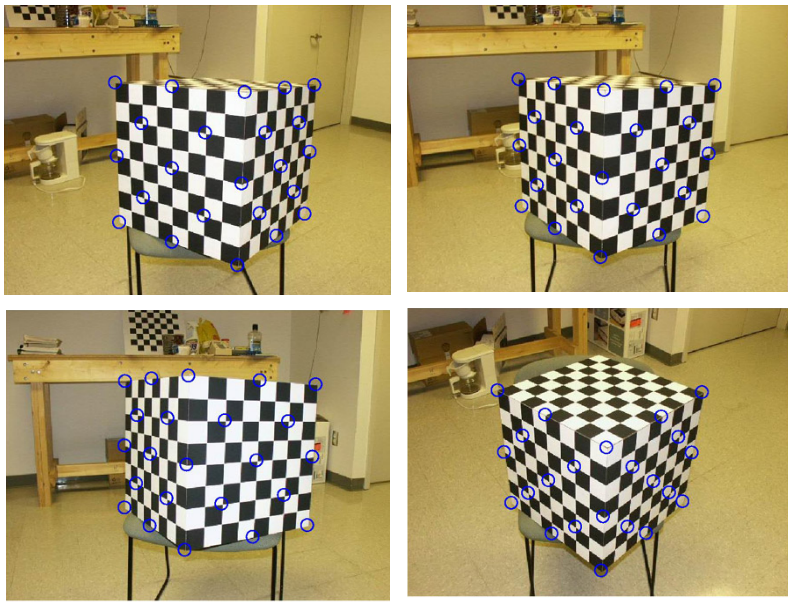
1. Initialization
   - Set k=0
   - Compute (R_2, T_2) using the 8-point algorithm
   - Compute $\lambda^j = \lambda^j_k$ and normalize so that $\alpha^1_k = 1$

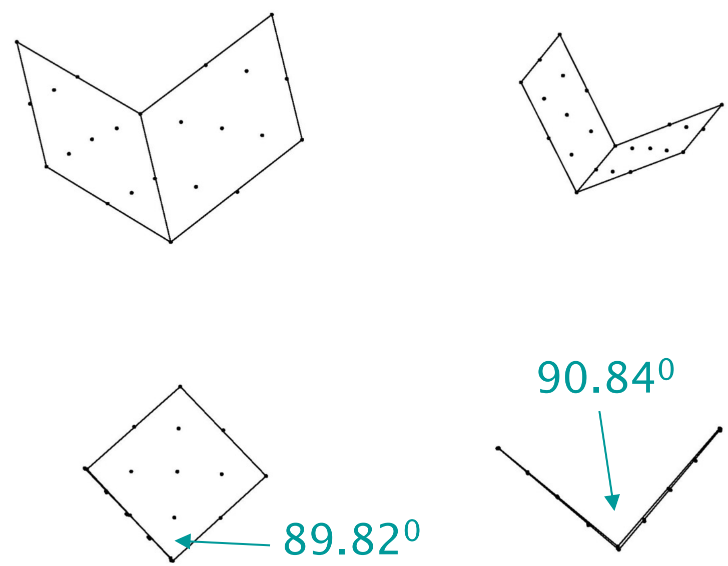
2. Compute $(\tilde{R}_i, \tilde{T}_i)$ as the null space of $P_{i=2,\ldots,m}$.

3. Compute new $\lambda^j = \lambda^j_{k+1}$ as the null space of $M^j, j = 1, \ldots, n.$ Normalize so that $\lambda^1_{k+1} = 1$

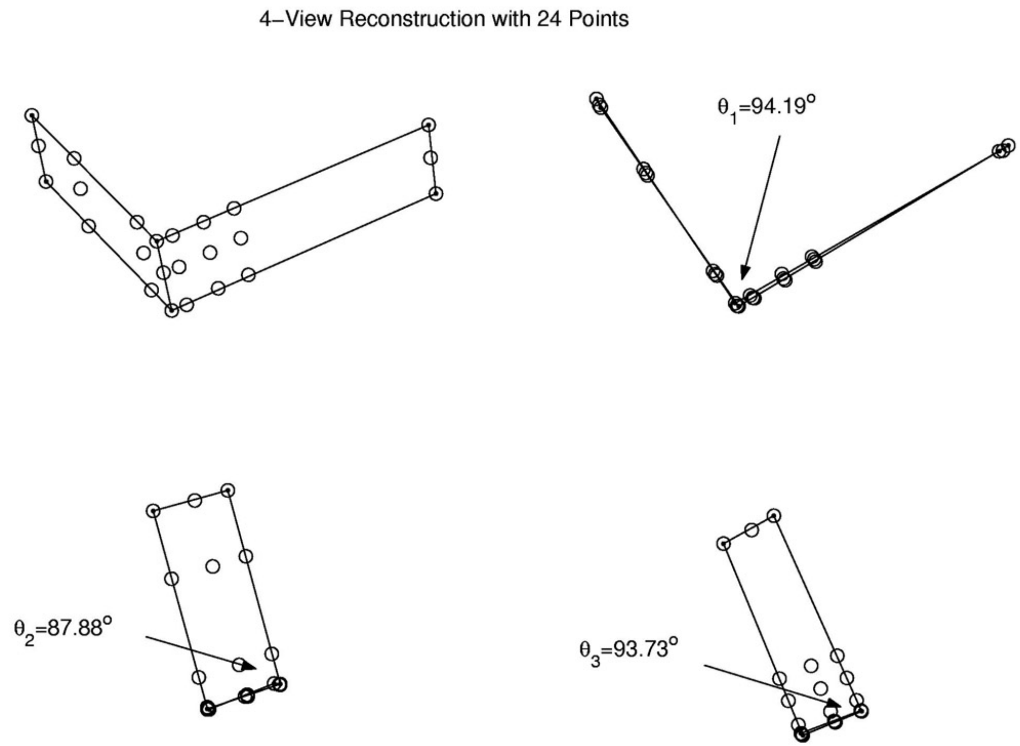
4. If $||\lambda_k - \lambda_{k+1}|| \leq \epsilon$ stop, else k=k+1 and goto 2.

Reconstruction Algorithm for Point Features





Reconstruction Algorithm for Point Features



Multiple View Matrix for Line Features

Homogeneous representation of a 3-D line L

$$\mathbf{X} = \mathbf{X}_o + \mu \mathbf{V}, \quad \mathbf{X}_o, \mathbf{V} \in \mathbb{R}^4, \mu \in \mathbb{R}$$

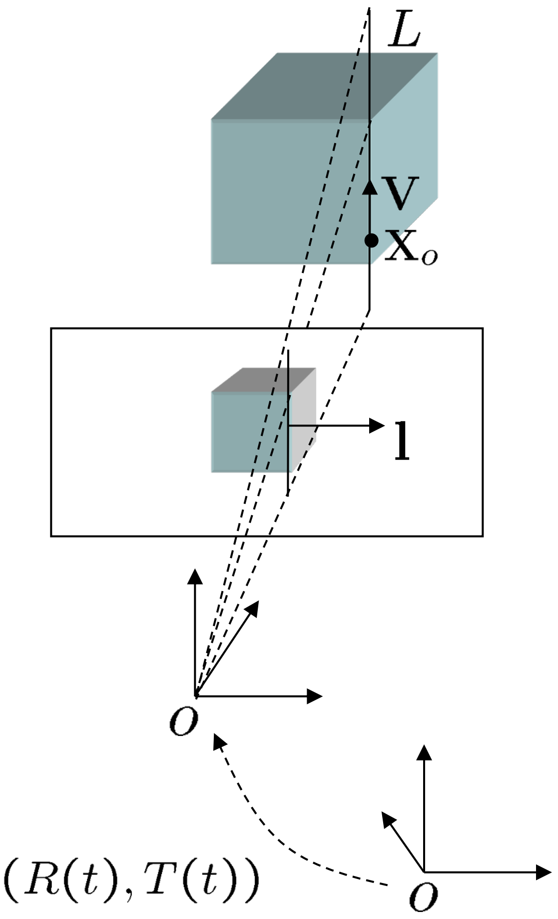
Homogeneous representation of its 2-D image

$$\boldsymbol{l} = [a, b, c]^T \in \mathbb{R}^3 \quad ax + by + c = 0$$

Projection of a 3-D line to an image plane

$$\boxed{\boldsymbol{l}(t)^T \mathbf{x}(t) = \boldsymbol{l}(t)^T \Pi(t) \boldsymbol{X} = 0}$$

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



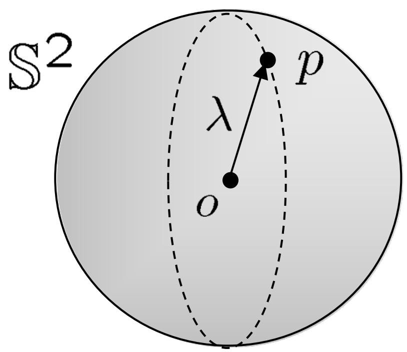
Multiple View Matrix for Line Features

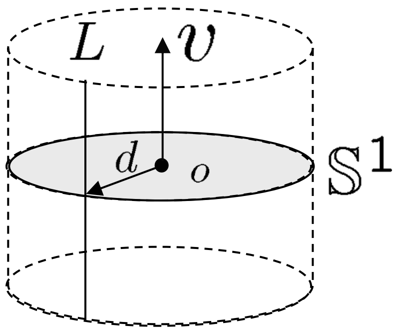
- Point Features
- Line Features

$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}, \quad M_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4}$$

$$\boxed{\text{rank}(M) = 1}$$

$$M_p \begin{pmatrix} \lambda \\ 1 \end{pmatrix} = 0 \qquad [\ v^T \quad d\ ]$$



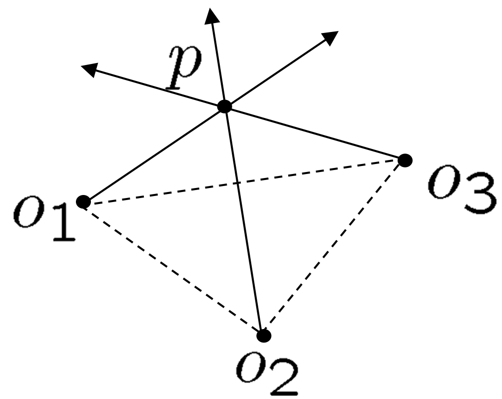


M encodes exactly the 3-D information missing in one image.

Multiple View Matrix for Line Features

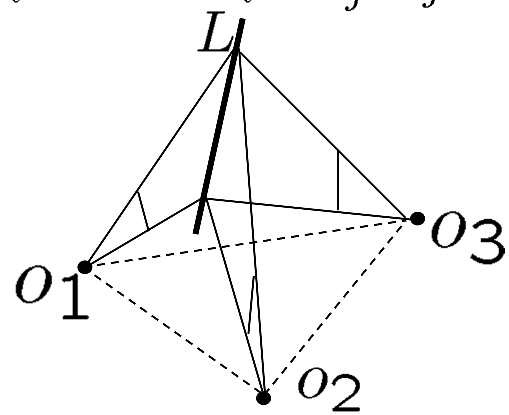
$$\text{rank}(M_p) = 1$$

$$\widehat{\boldsymbol{x}_i}(T_i\boldsymbol{x}_1^T R_j^T - R_i\boldsymbol{x}_1 T_j^T)\widehat{\boldsymbol{x}_j} = 0$$

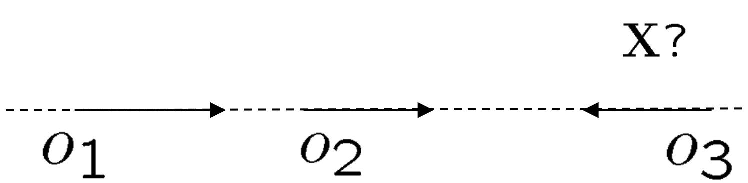


$$\text{rank}(M_l) = 1$$

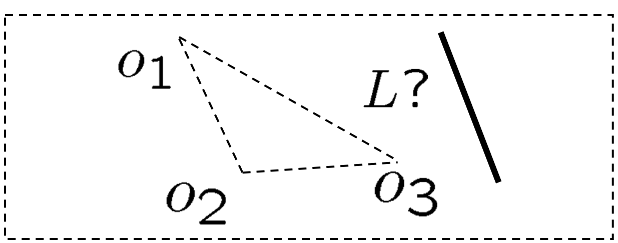
$$\boldsymbol{l}_j^T T_j \boldsymbol{l}_i^T R_i \widehat{\boldsymbol{l}_1} - \boldsymbol{l}_i^T T_i \boldsymbol{l}_j^T R_j \widehat{\boldsymbol{l}_1} = 0$$



$$\text{rank}(M_p) = 0$$



$$\text{rank}(M_l) = 0$$



Multiple View Matrix for Line Features

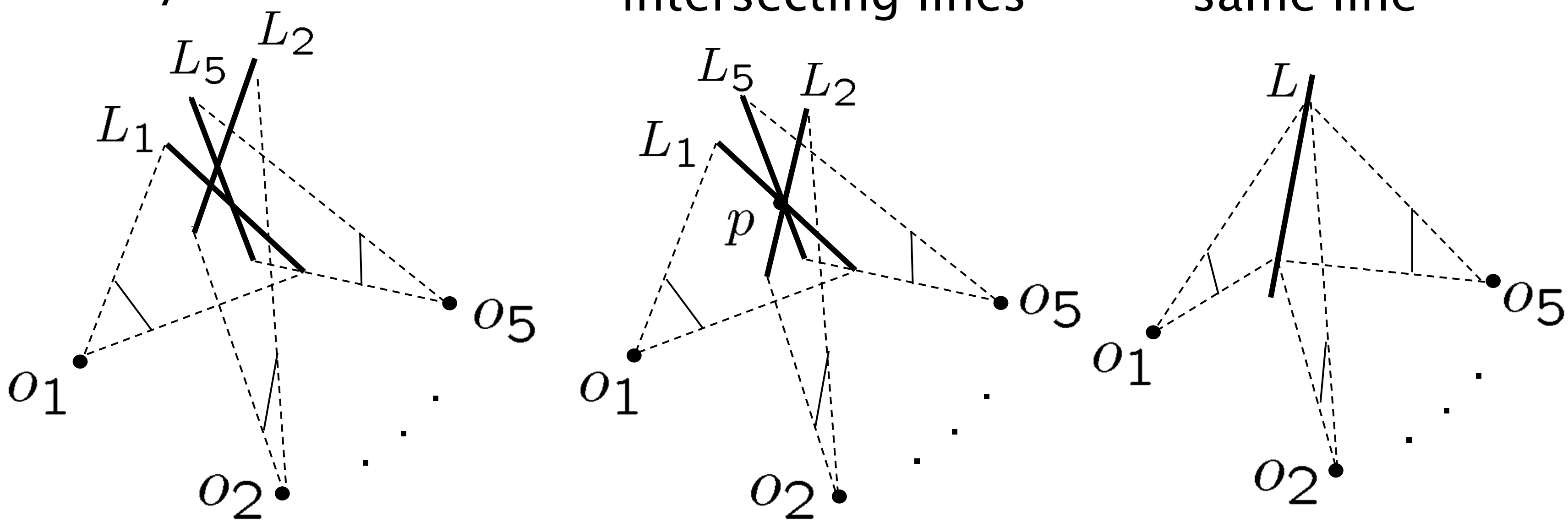
$$M_l = \begin{bmatrix} l_2^T R_2 \widehat{l_1} & l_2^T T_2 \\ l_3^T R_3 \widehat{l_1} & l_3^T T_3 \\ l_4^T R_4 \widehat{l_1} & l_4^T T_4 \\ l_5^T R_5 \widehat{l_1} & l_5^T T_5 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad \mathrm{rank}(M_l) = 3, 2, 1.$$

$\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3, \boldsymbol{l}_4, \boldsymbol{l}_5$ each is an image of a (different) line in 3-D:

Rank =3
any lines

Rank =2
intersecting lines

Rank =1
same line



Reconstruction Algorithm for Line Features

1. Initialization
   - Set k=0
   - Compute (R_2, T_2) and (R_3, T_3) using linear algorithm
   - Compute $\lambda^j = \lambda_k^j$ and normalize so that $\alpha_k^1 = 1$

2. Compute $(\tilde{R}_i, \tilde{T}_i)$ as the null space of $P_{i=2,\ldots,m}$.

3. Compute new $\lambda^j = \lambda_{k+1}^j$ as the null space of $M^j, j = 1, \ldots, n.$ Normalize so that $\lambda_{k+1}^1 = 1$

4. If $||\lambda_k - \lambda_{k+1}|| \leq \epsilon$ stop, else k=k+1 and goto 2.

Universal Rank Constraint

- What if I have both point and line features?
  - Traditionally points and lines are treated separately
  - Therefore, joint incidence relations not exploited
- Can we express joint incidence relations for
  - Points passing through lines?
  - Families of intersecting lines?

Universal Rank Constraint

- **The Universal Rank Condition** for images of a point on a line

$$M \doteq \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \end{bmatrix}, \quad \text{where} \quad \begin{cases} D_i \doteq & \boldsymbol{x}_i \ \text{ or } \ \widehat{\boldsymbol{l}}_i, \\ D_i^{\perp} \doteq & \widehat{\boldsymbol{x}_i} \ \text{ or } \ \boldsymbol{l}_i^T. \end{cases}$$

1. If $D_1 = \widehat{\boldsymbol{l}_1}$ and $D_i^{\perp} = \widehat{\boldsymbol{x}_i}$ for some $i \geq 2$, then:

$$\boxed{1 \leq \operatorname{rank}(M) \leq 2.}$$

-Multi-nonlinear constraints among 3, 4-wise images.

2. Otherwise:

$$\boxed{0 \leq \operatorname{rank}(M) \leq 1.}$$

-Multi-linear constraints among 2, 3-wise images.

Multiview Formulation: The Multiple View Matrix

$$M \doteq \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \end{bmatrix}, \quad \text{where} \quad \begin{cases} D_i \doteq & \boldsymbol{x}_i \quad \text{or} \quad \widehat{\boldsymbol{l}}_i, \\ D_i^{\perp} \doteq & \widehat{\boldsymbol{x}_i} \quad \text{or} \quad \boldsymbol{l}_i^T. \end{cases}$$

Point Features

$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2},$$

$$0 \leq \text{rank}(M_p) \leq 1$$

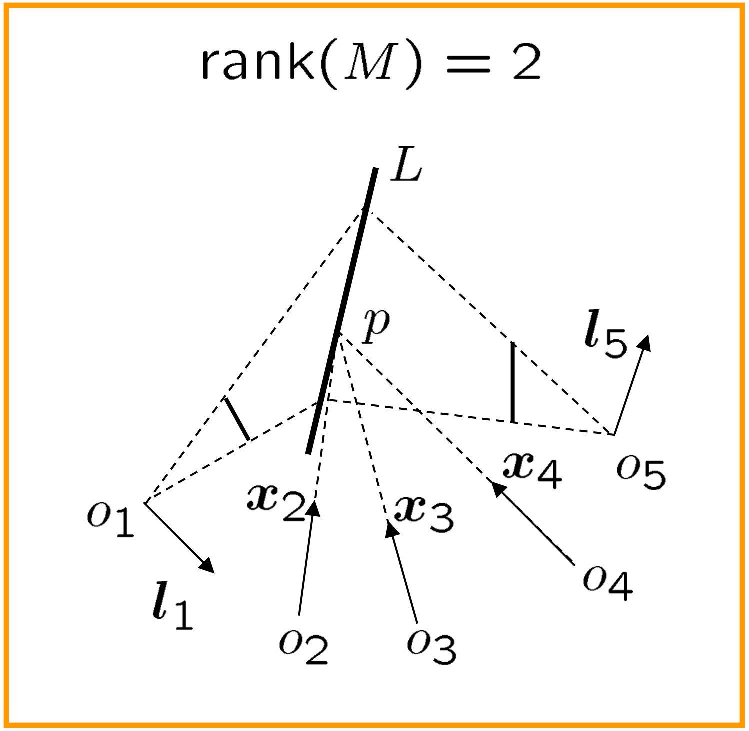
Line Features

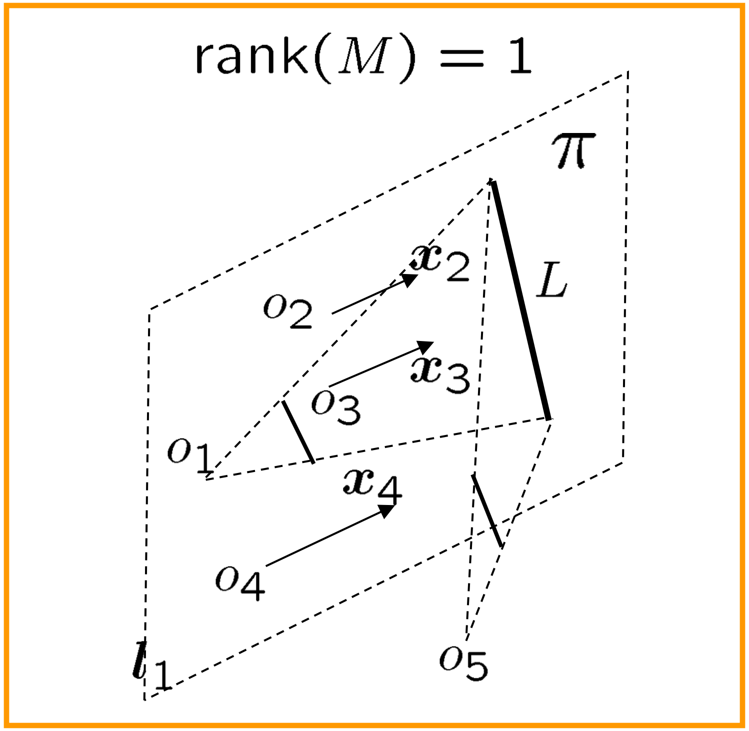
$$M_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4}$$

$$0 \leq \text{rank}(M_l) \leq 1$$

Universal Rank Constraint: points and lines

$$M = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \widehat{\boldsymbol{l}_1} & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \widehat{\boldsymbol{l}_1} & \widehat{\boldsymbol{x}_3} T_3 \\ \widehat{\boldsymbol{x}_4} R_4 \widehat{\boldsymbol{l}_1} & \widehat{\boldsymbol{x}_4} T_4 \\ \boldsymbol{l}_5^T R_5 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_5^T T_5 \end{bmatrix} \in \mathbb{R}^{10\times 4}, \quad 1 \leq \text{rank}(M) \leq 2.$$

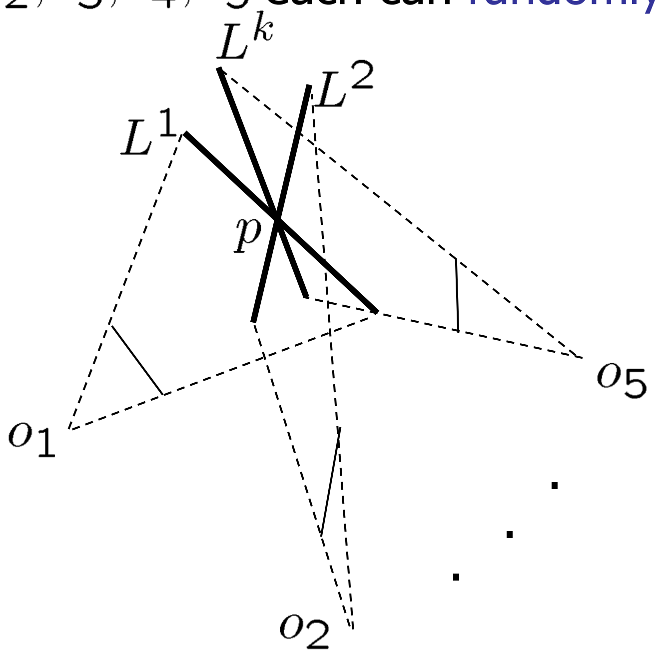




Universal Rank Constraint: family of intersecting lines

$$\tilde{M}_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \boldsymbol{l}_4^T R_4 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_4^T T_4 \\ \boldsymbol{l}_5^T R_5 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_5^T T_5 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad 1 \leq \mathrm{rank}(\tilde{M}_l) \leq 2.$$

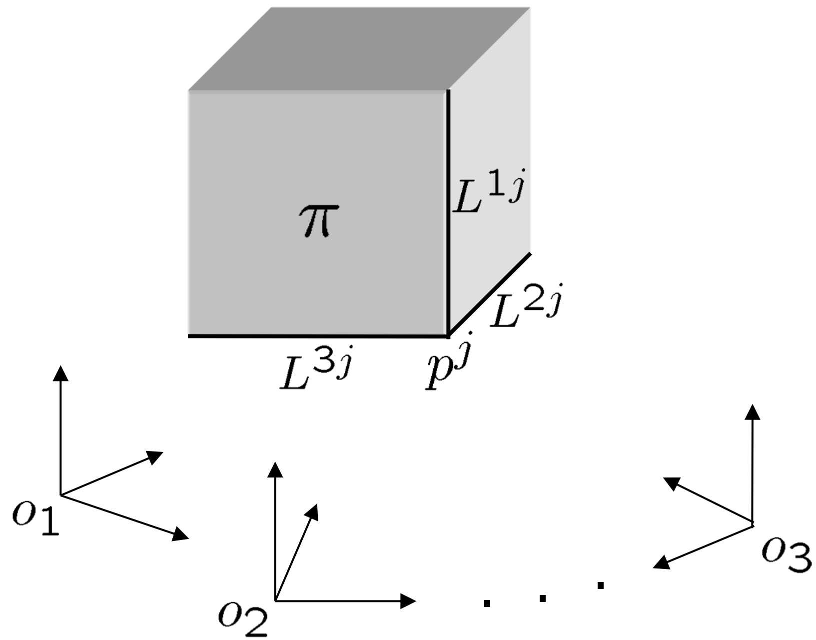
$\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3, \boldsymbol{l}_4, \boldsymbol{l}_5$ each can randomly take the image of any of the lines:



Nonlinear constraints among up to four views

Universal Rank Constraint: multiple images of a cube

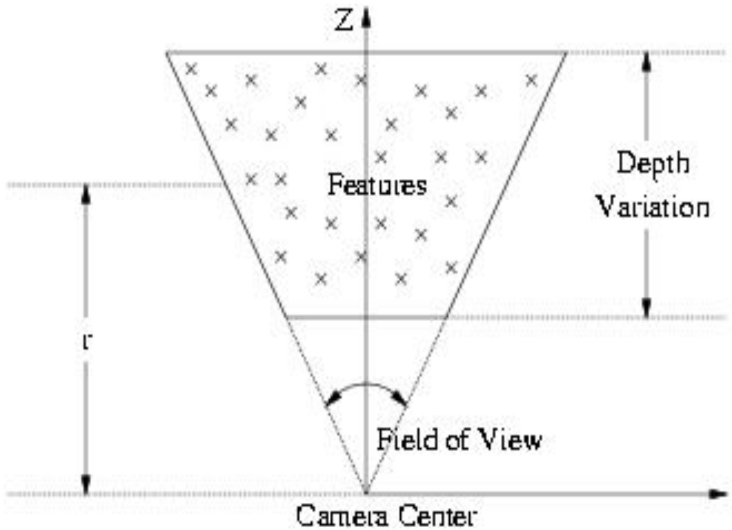
Three edges intersect at each vertex.

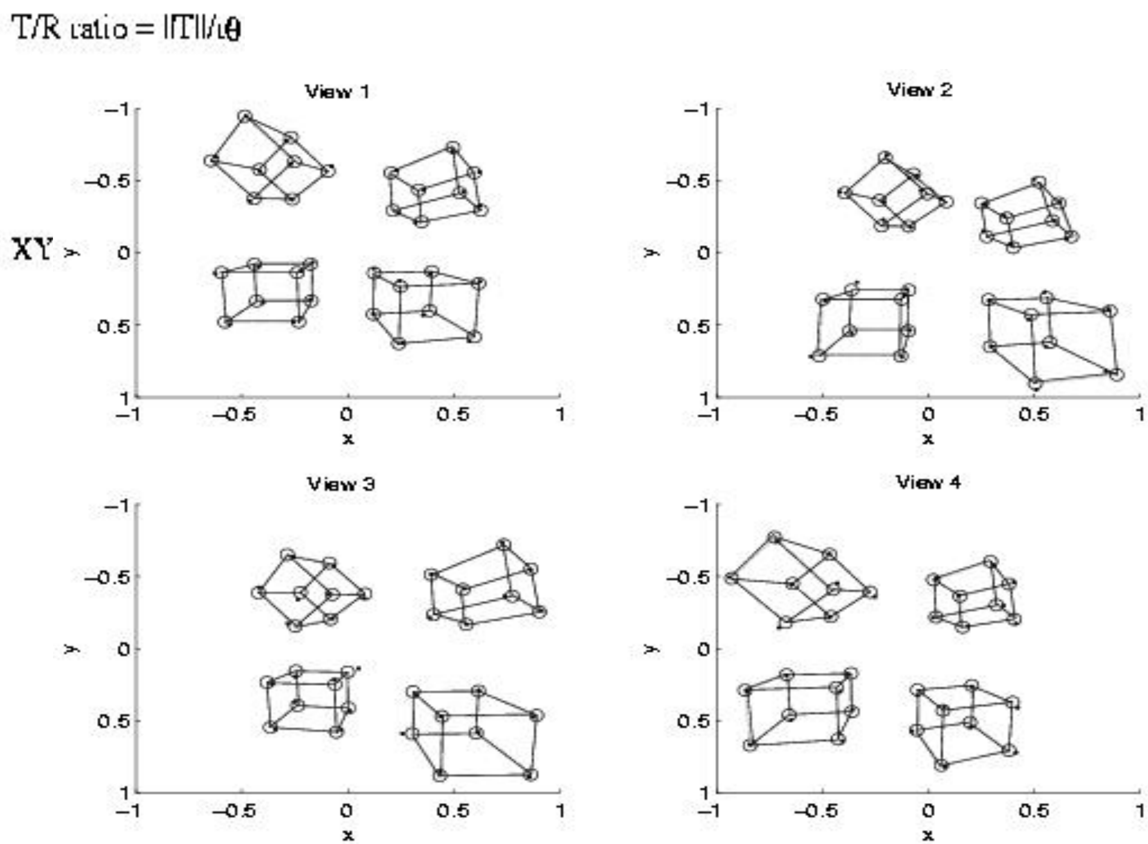


$$M^j = \begin{bmatrix} \widehat{\boldsymbol{x}_2^j} R_2 \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_2^j} T_2 \\ \boldsymbol{l}_2^{1jT} R_2 \boldsymbol{x}_1^j & \boldsymbol{l}_2^{1jT} T_2 \\ \boldsymbol{l}_2^{2jT} R_2 \boldsymbol{x}_1^j & \boldsymbol{l}_2^{2jT} T_2 \\ \boldsymbol{l}_2^{3jT} R_2 \boldsymbol{x}_1^j & \boldsymbol{l}_2^{3jT} T_2 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m^j} R_m \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_m^j} T_m \\ \boldsymbol{l}_m^{1jT} R_m \boldsymbol{x}_1^j & \boldsymbol{l}_m^{1jT} T_m \\ \boldsymbol{l}_m^{2jT} R_m \boldsymbol{x}_1^j & \boldsymbol{l}_m^{2jT} T_m \\ \boldsymbol{l}_m^{3jT} R_m \boldsymbol{x}_1^j & \boldsymbol{l}_m^{3jT} T_m \end{bmatrix}$$

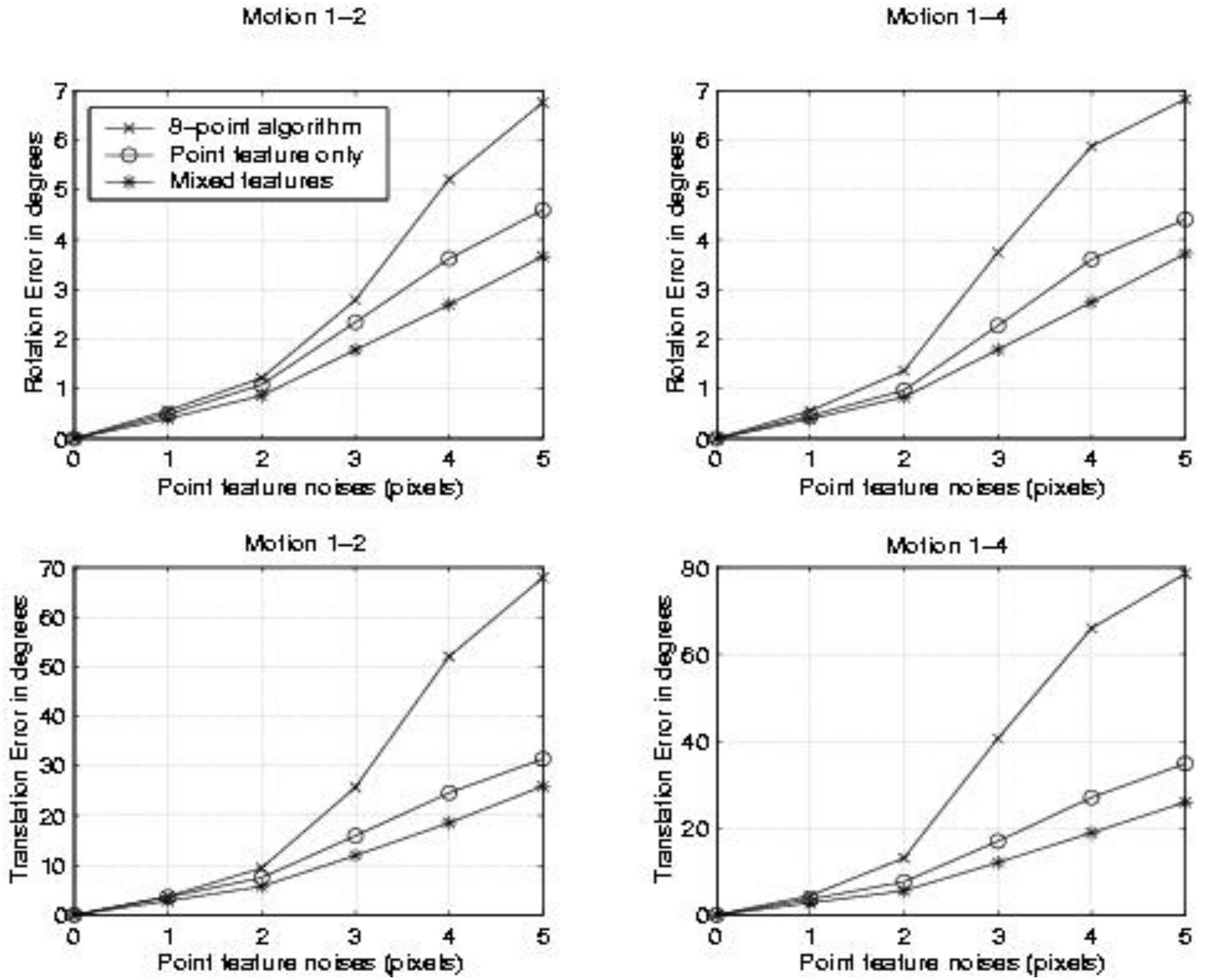
$$0 \leq \text{rank}(M^j) \leq 1$$

Universal Rank Constraint: multiple images of a cube

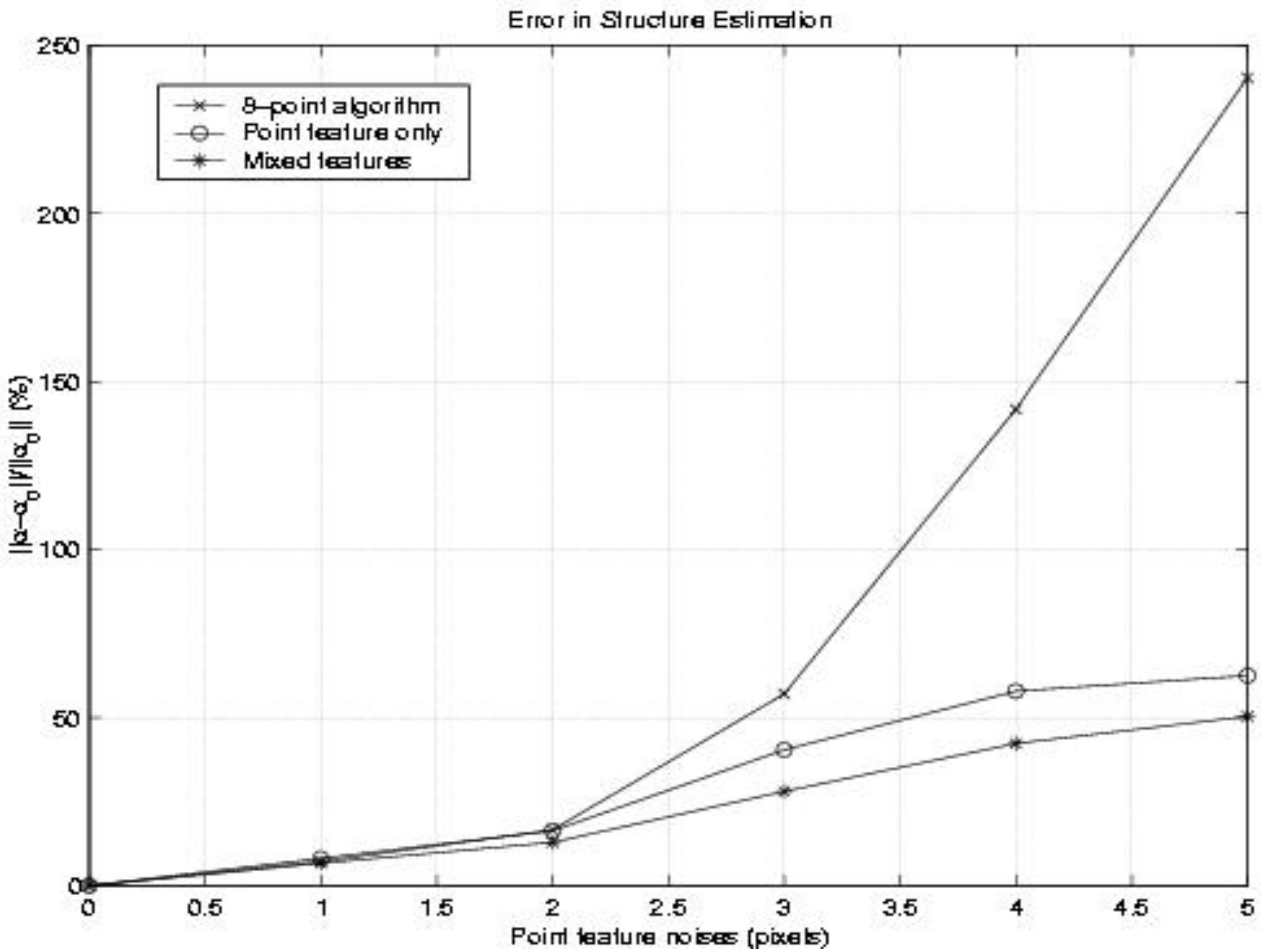




Universal Rank Constraint: multiple images of a cube



Universal Rank Constraint: multiple images of a cube



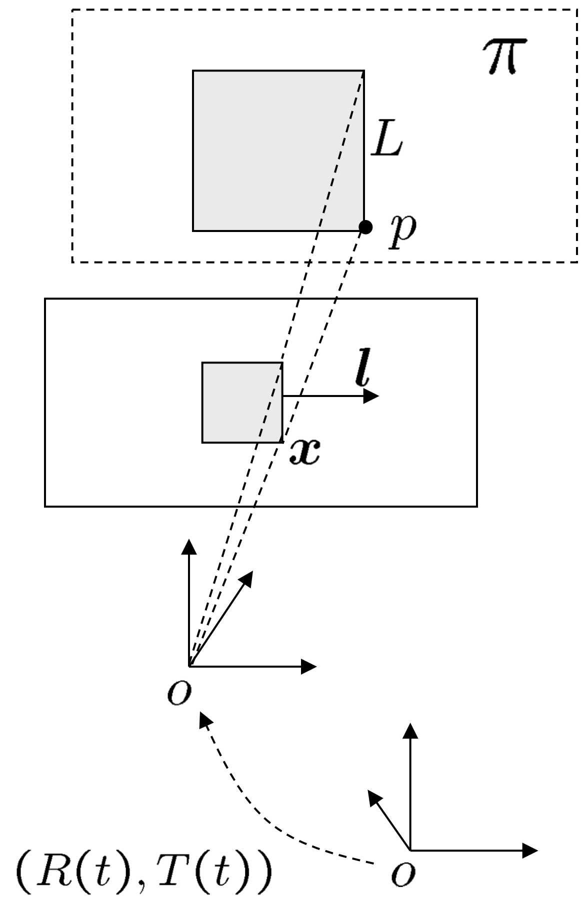
Multiple View Matrix for Coplanar Point Features

Homogeneous representation of a 3-D plane π

$$aX + bY + cZ + d = 0.$$

$$\pi \mathbf{X} = 0, \quad \pi = [\pi^1, \pi^2] : \pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}$$

$$M \doteq \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \\ \pi^1 D_1 & \pi^2 \end{bmatrix}$$



Corollary [Coplanar Features]

Rank conditions on the new extended M remain exactly the same!

Multiple View Matrix for Coplanar Point Features

Given that a point and line features lie on a plane π in 3-D space:

In addition to previous constraints, it simultaneously gives homography:

$$\boxed{\widehat{\boldsymbol{x}_i}(R_i\pi^2 - T_i\pi^1)\boldsymbol{x}_1 = 0}$$

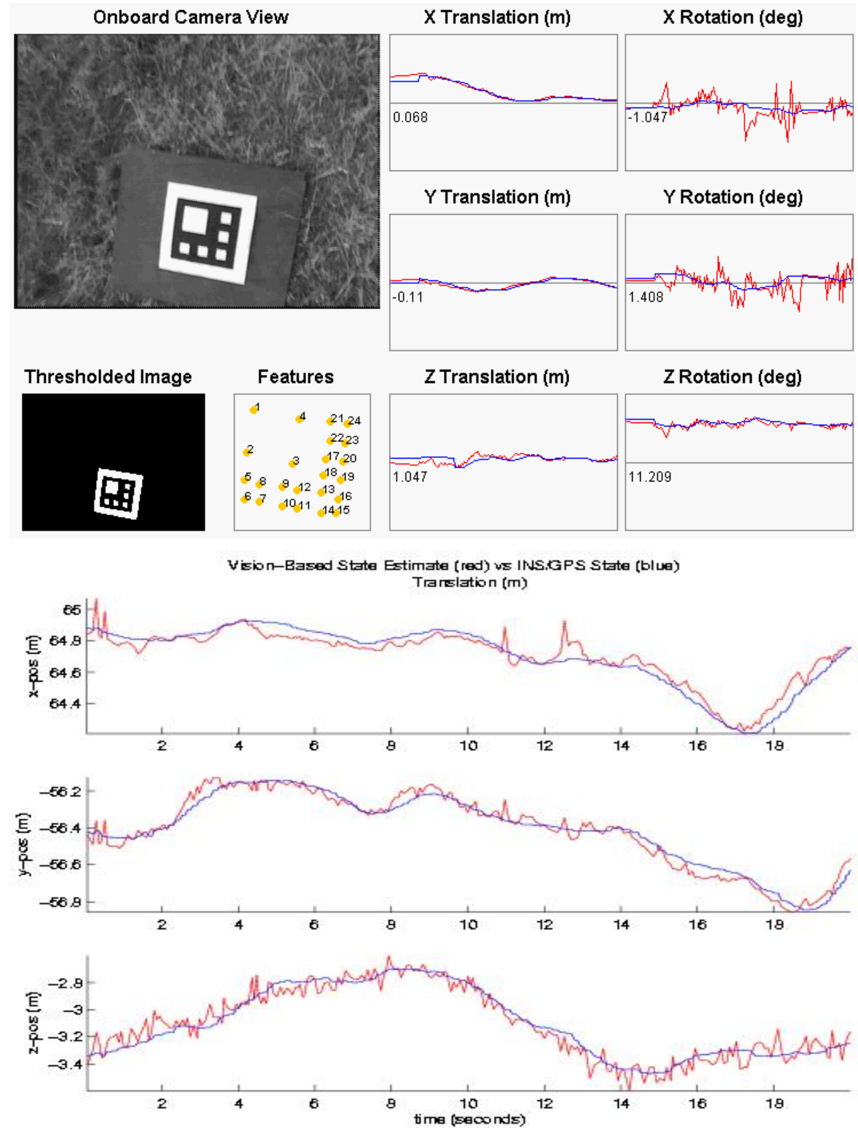
$$\boxed{\boldsymbol{l}_i^T(R_i\pi^2 - T_i\pi^1)\widehat{\boldsymbol{l}_1} = 0}$$

$$0 \leq \mathrm{rank}(M_p) \leq 1$$

$$0 \leq \mathrm{rank}(M_l) \leq 1$$

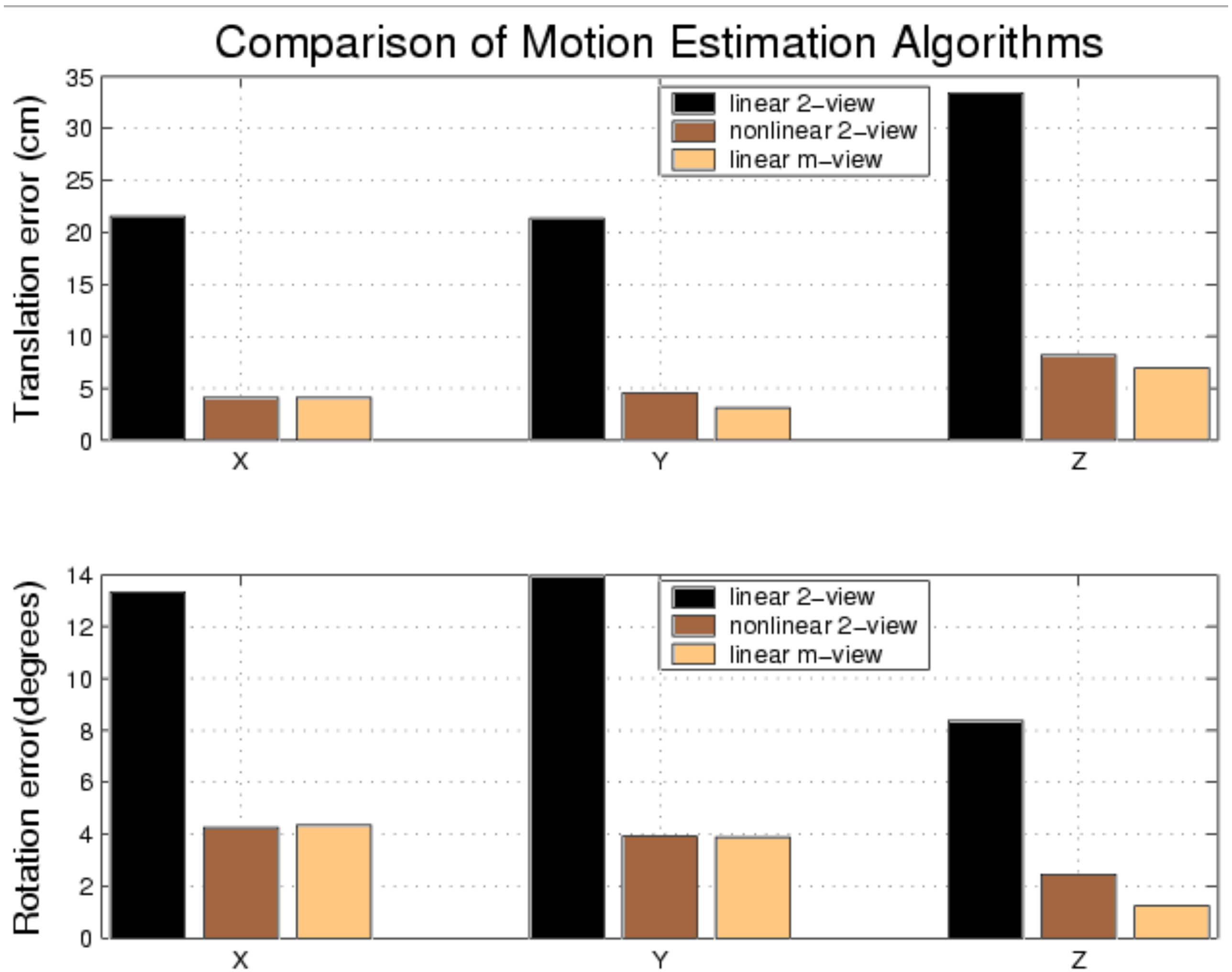
$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2}R_2\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2}T_2 \\ \widehat{\boldsymbol{x}_3}R_3\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3}T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m}R_m\boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m}T_m \\ \pi^1\boldsymbol{x}_1 & \pi^2 \end{bmatrix} \in \mathbb{R}^{(3m-2)\times 2}, \quad M_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2\widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3\widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m\widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \\ \pi^1\widehat{\boldsymbol{l}_1} & \pi^2 \end{bmatrix} \in \mathbb{R}^{m\times 4}$$

Example: Vision-based Landing of a Helicopter



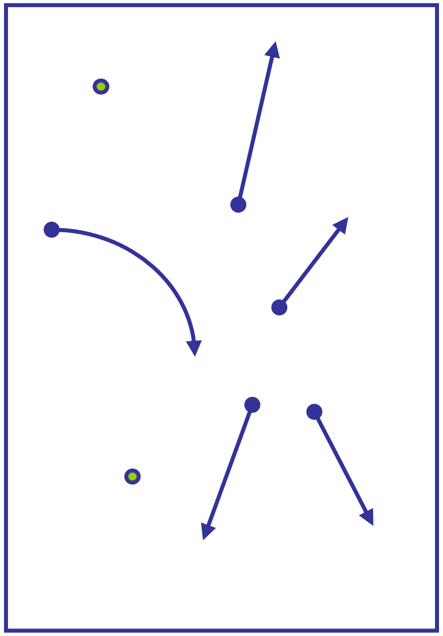
Rate: 10Hz
Accuracy: 3-5cm, 4°

Example: Vision-based Landing of a Helicopter



General Rank Constraint for Dynamic Scenes

For a fixed camera, assume the point moves with constant acceleration:



$$X(t) = X_0 + t v_0 + \frac{t^2}{2} a$$

$$\bar{\mathbf{X}} = \begin{bmatrix} X_0 \\ v_0 \\ a \\ 1 \end{bmatrix} \in \mathbb{R}^{10}$$

$$\bar{\Pi}(t) = \begin{bmatrix} I & tI & \frac{t^2}{2} I & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 10}$$

$$\lambda(t)\boldsymbol{x}(t) = \bar{\Pi}(t)\bar{\mathbf{X}}, \quad \mathbf{X} \in \mathbb{R}^{10}, \boldsymbol{x} \in \mathbb{R}^3$$

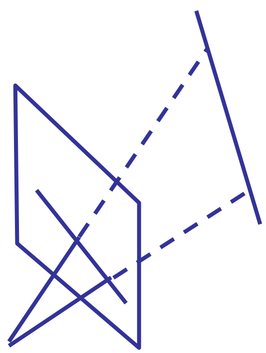
Before: $\lambda(t)\boldsymbol{x}(t) = \Pi(t)\mathbf{X}, \quad \mathbf{X} \in \mathbb{R}^4, \boldsymbol{x} \in \mathbb{R}^3.$

Now: $\lambda(t)\boldsymbol{x}(t) = \Pi(t)\mathbf{X}(t), \quad \mathbf{X}(t) = \underbrace{[b_1(t), b_2(t), \ldots, b_{n+1}(t)]}_{\text{Time base}}\bar{\mathbf{X}}$

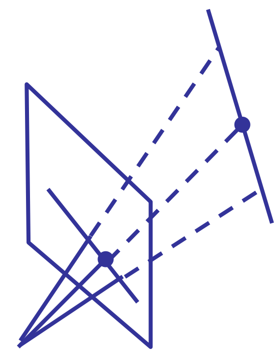
$$\lambda(t)\boldsymbol{x}(t) = \bar{\Pi}(t)\bar{\mathbf{X}}, \quad \bar{\mathbf{X}} \in \mathbb{R}^{n+1}, \boldsymbol{x} \in \mathbb{R}^3.$$

General Rank Constraint for Dynamic Scenes

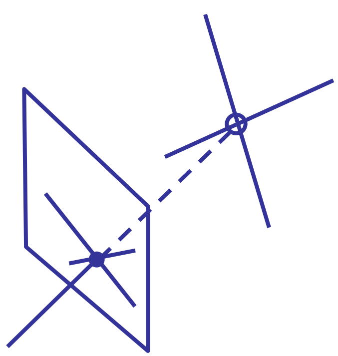
Single hyperplane



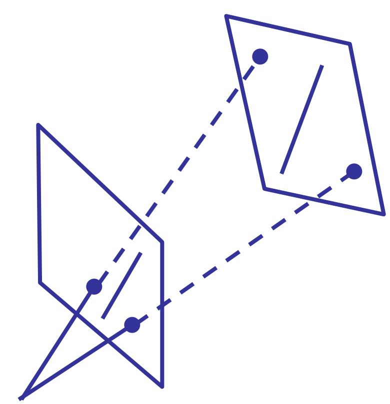
Inclusion



Intersection



Restriction to a hyperplane



General Rank Constraint for Dynamic Scenes

Projection from *n*–dimensional space to *k*–dimensional space:

$$\lambda_i \boldsymbol{x}_i = \bar{\Pi}_i \mathbf{X}_i, \quad i = 1, \ldots, m,$$

$$\bar{\Pi}_i = \left[\begin{array}{cc} \bar{R}_i & \bar{T}_i \end{array} \right],$$

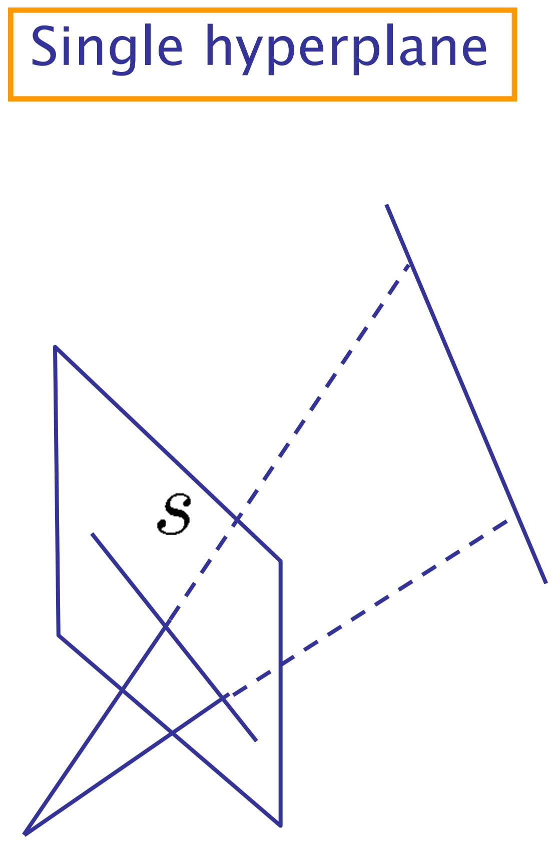
$$\bar{R}_i \in \mathbb{R}^{(k+1)\times(k+1)}, \quad \bar{T}_i \in \mathbb{R}^{(k+1)\times(n-k)}$$

We define the multiple view matrix as:

$$M \doteq \left[\begin{array}{cc} (D_2^{\perp})^T \bar{R}_2 D_1 & (D_2^{\perp})^T \bar{T}_2 \\ (D_3^{\perp})^T \bar{R}_3 D_1 & (D_3^{\perp})^T \bar{T}_3 \\ \vdots & \vdots \\ (D_m^{\perp})^T \bar{R}_m D_1 & (D_m^{\perp})^T \bar{T}_m \end{array} \right],$$

where D_i ‘s and $D_i^{\perp}$ ‘s are images and coimages of hyperplanes.

General Rank Constraint for Dynamic Scenes



Projection from $\mathbb{R}^n$ to $\mathbb{R}^k$

$$D_1 = s_1 \quad D_i^{\perp} = s_i^{\perp}$$

$$0 \leq \text{rank}(M) \leq (n-k)$$

Projection from $\mathbb{R}^3$ to $\mathbb{R}^2$

$$D_1 = s_1 \quad D_i^{\perp} = s_i^{\perp}$$

$$0 \leq \text{rank}(M) \leq 1$$

Summary

- Incidence relations $\Leftrightarrow$ rank conditions
- Rank conditions $\Rightarrow$ multiple-view factorization
- Rank conditions imply all multi-focal constraints
- Rank conditions for points, lines, planes, and (symmetric) structures.