

Lecture 7

Multi-view geometry

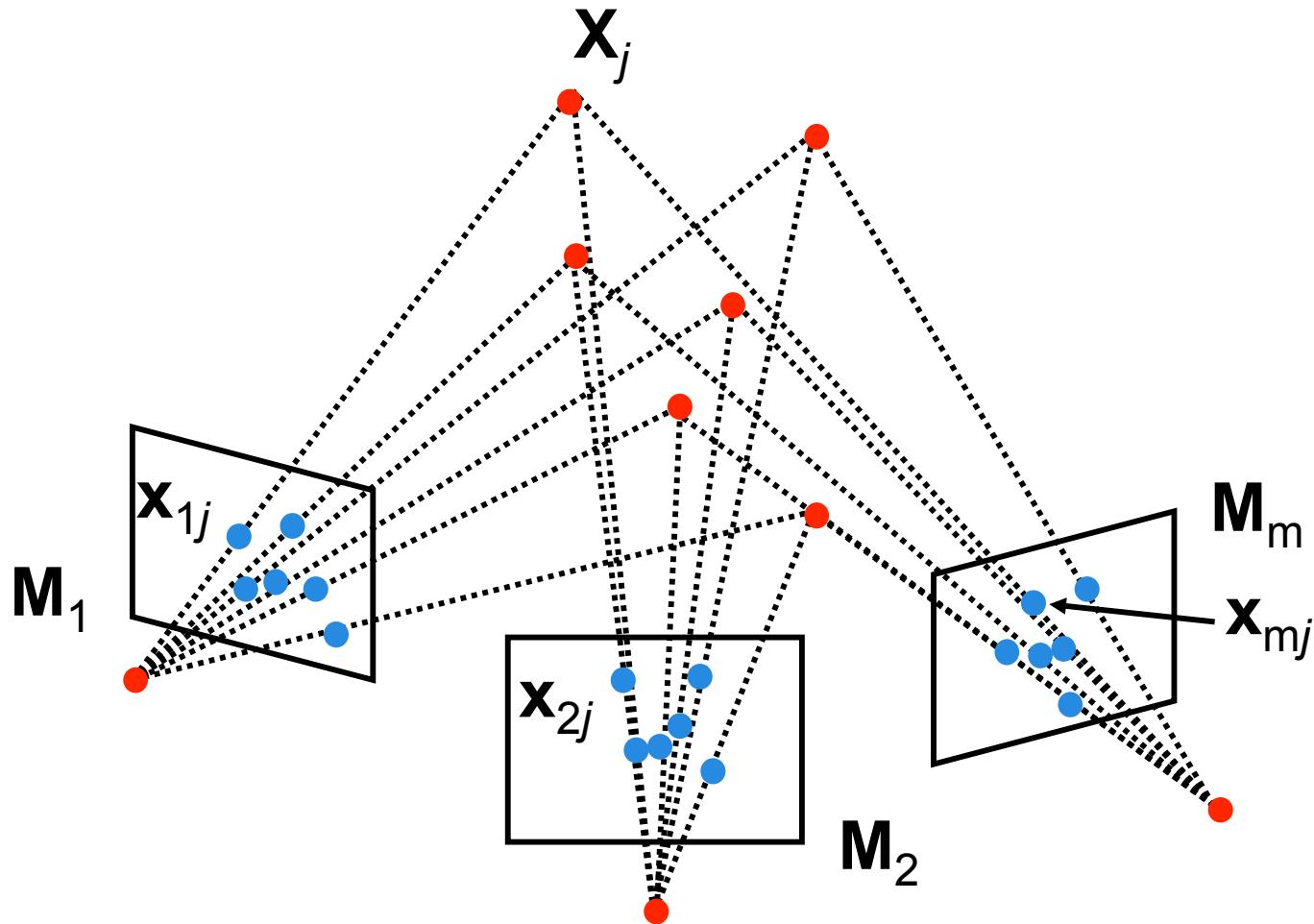


- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications

Reading:

- [HZ] Chapter 10 “3D reconstruction of cameras and structure”
Chapter 18 “N-view computational methods”
Chapter 19 “Auto-calibration”
- [FP] Chapter 13 “projective structure from motion”
- [Szelisky] Chapter 7 “Structure from motion”

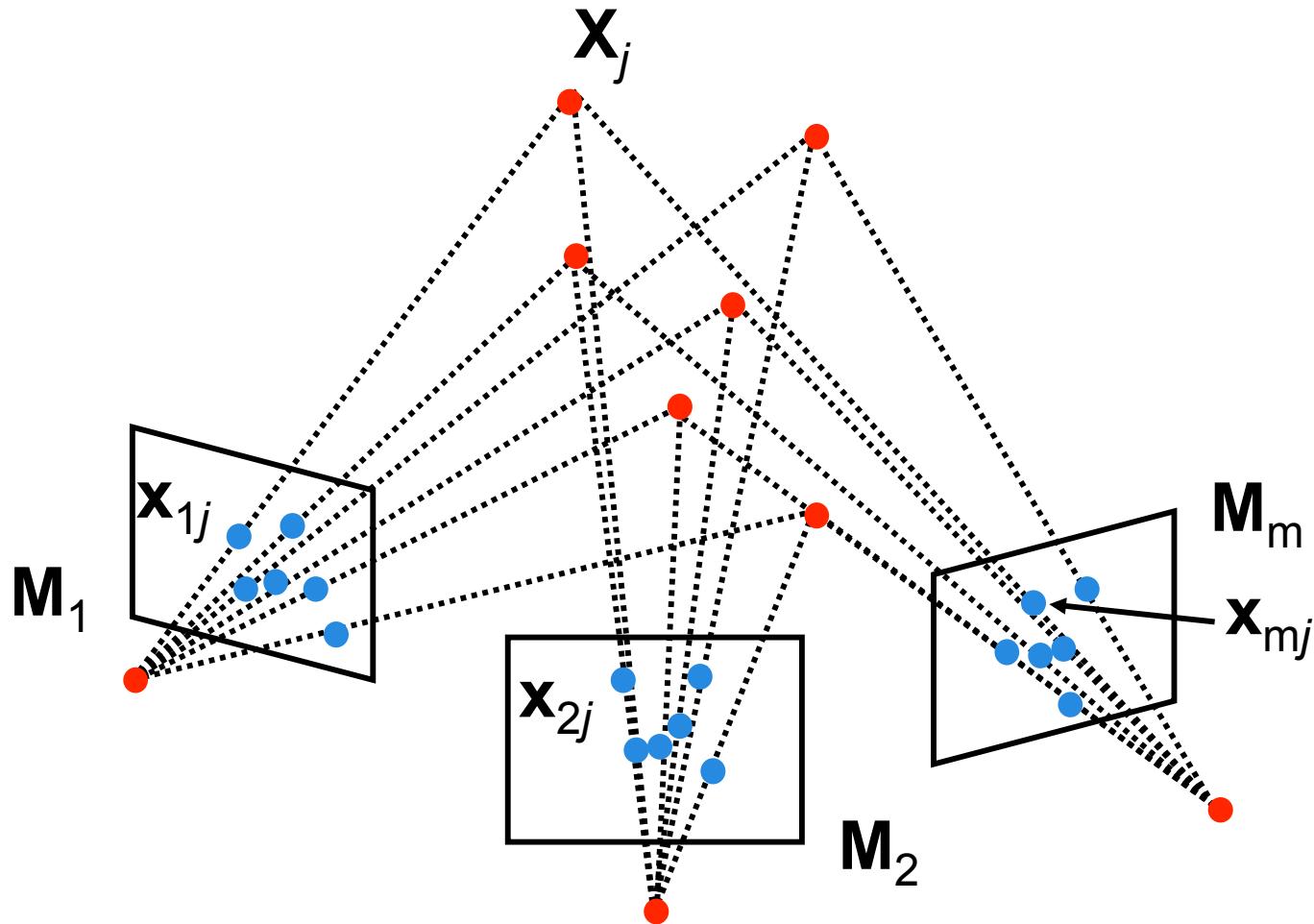
Structure from motion problem



Given m images of n fixed 3D points

$$\bullet \mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

Structure from motion problem

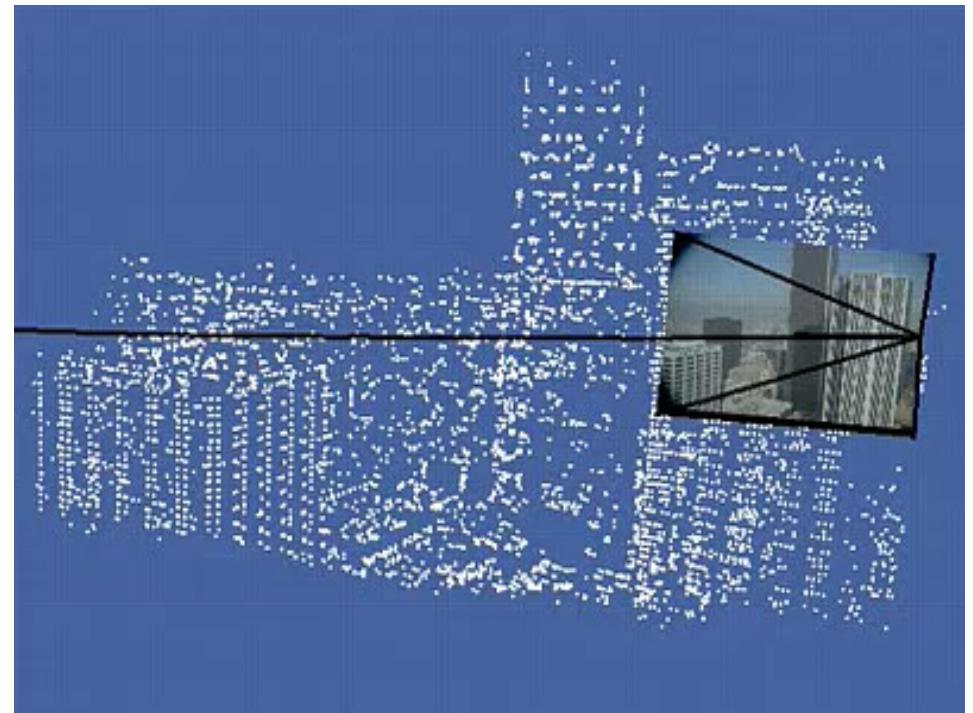


From the $m \times n$ observations \mathbf{x}_{ij} , estimate:

- m projection matrices \mathbf{M}_i
- n 3D points \mathbf{X}_j

motion
structure

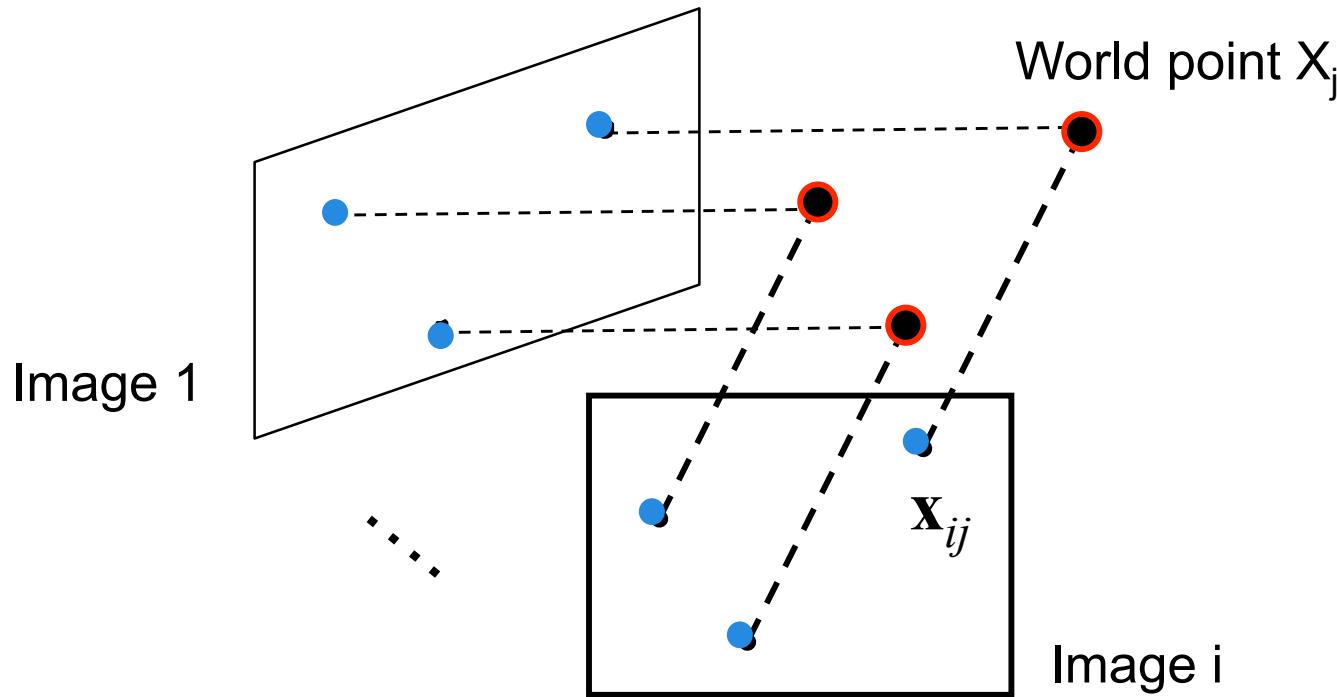
Structure from motion problem



Courtesy of Oxford **Visual Geometry Group**

Affine structure from motion

(simpler problem)



From the $m \times n$ observations x_{ij} , estimate:

- m projection matrices \mathbf{M}_i (affine cameras)
- n 3D points \mathbf{X}_j

Perspective

$$\mathbf{x} = M \mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \mathbf{X} \\ \mathbf{m}_2 \mathbf{X} \\ \mathbf{m}_3 \mathbf{X} \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

$$\mathbf{x}^E = \left(\frac{\mathbf{m}_1 \mathbf{X}}{\mathbf{m}_3 \mathbf{X}}, \frac{\mathbf{m}_2 \mathbf{X}}{\mathbf{m}_3 \mathbf{X}} \right)^T$$

Affine

$$\mathbf{x} = M \mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \mathbf{X} \\ \mathbf{m}_2 \mathbf{X} \\ 1 \end{bmatrix}$$

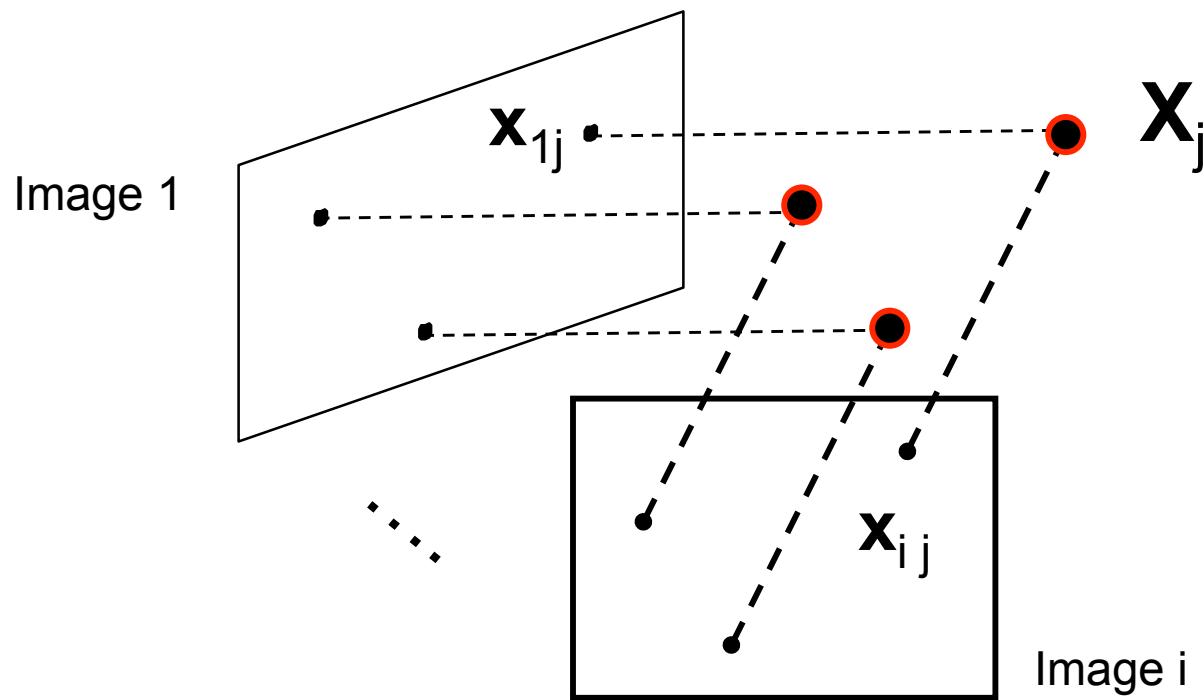
$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{2x3} & \mathbf{b}_{2x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}^E = (\mathbf{m}_1 \mathbf{X}, \mathbf{m}_2 \mathbf{X})^T = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{AX}^E + \mathbf{b}$$

[Eq. 3]

$$\mathbf{X}^E = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Affine cameras



For the affine case (in Euclidean space)

$$\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i \quad [\text{Eq. 4}]$$

2x1 2x3 3x1 2x1

The Affine Structure-from-Motion Problem

Given m images of n fixed points \mathbf{X}_j we can write

$$\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i \quad \text{for } i = 1, \dots, m \quad \text{and } j = 1, \dots, n$$

N. of cameras N. of points

Problem: estimate m matrices \mathbf{A}_i , m matrices \mathbf{b}_i and the n positions \mathbf{X}_j from the $m \times n$ observations \mathbf{x}_{ij} .

How many equations and how many unknowns?

$2m \times n$ equations in $8m+3n$ unknowns

With $m=2$ cameras, I need at least $n=16$ 3D points
With $m=3$ cameras, I need at least $n=8$ 3D points

The Affine Structure-from-Motion Problem

Two approaches:

- Algebraic approach (affine epipolar geometry; estimate F ; cameras; points)
- Factorization method

A factorization method – Tomasi & Kanade algorithm

C. Tomasi and T. Kanade

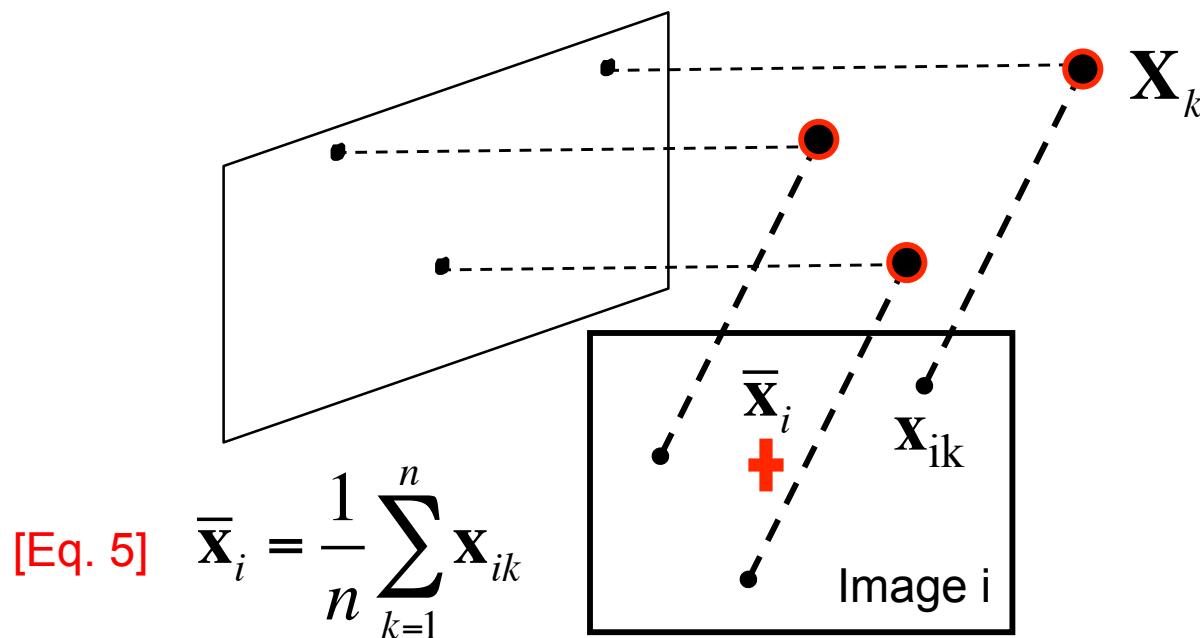
[Shape and motion from image streams under orthography: A factorization method.](#) IJCV, 9(2):137-154,
November 1992.

- Data centering
- Factorization

A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$[\text{Eq. 6}] \quad \hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik} \quad \bar{\mathbf{x}}_i$$



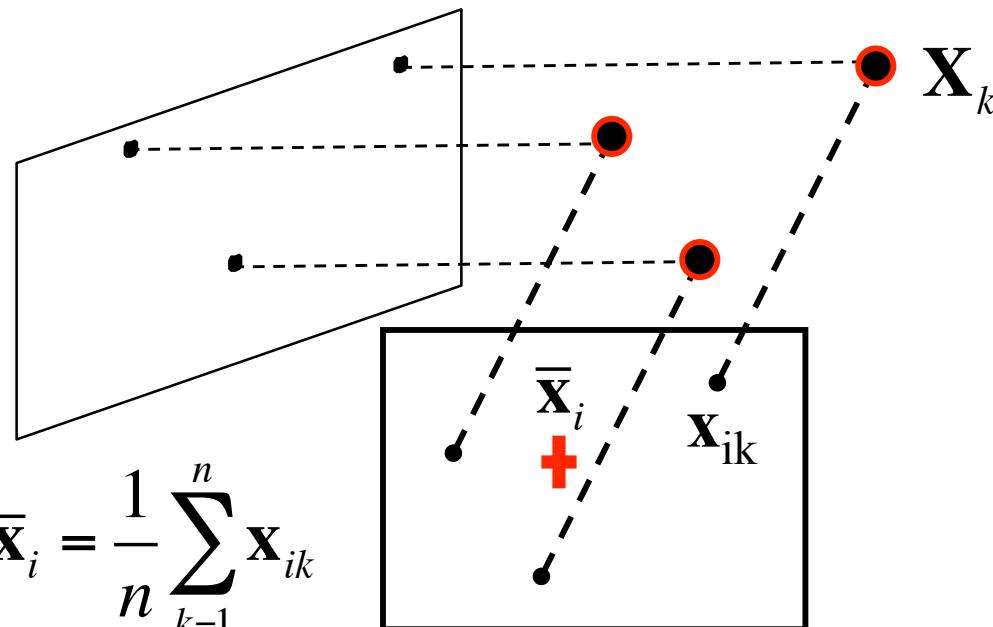
A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$[\text{Eq. 6}] \quad \hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i - \frac{1}{n} \sum_{k=1}^n \mathbf{A}_i \mathbf{X}_k - \frac{1}{n} \sum_{k=1}^n \mathbf{b}_i$$

$$\mathbf{x}_{ik} = \mathbf{A}_i \mathbf{X}_k + \mathbf{b}_i$$

[Eq. 4]

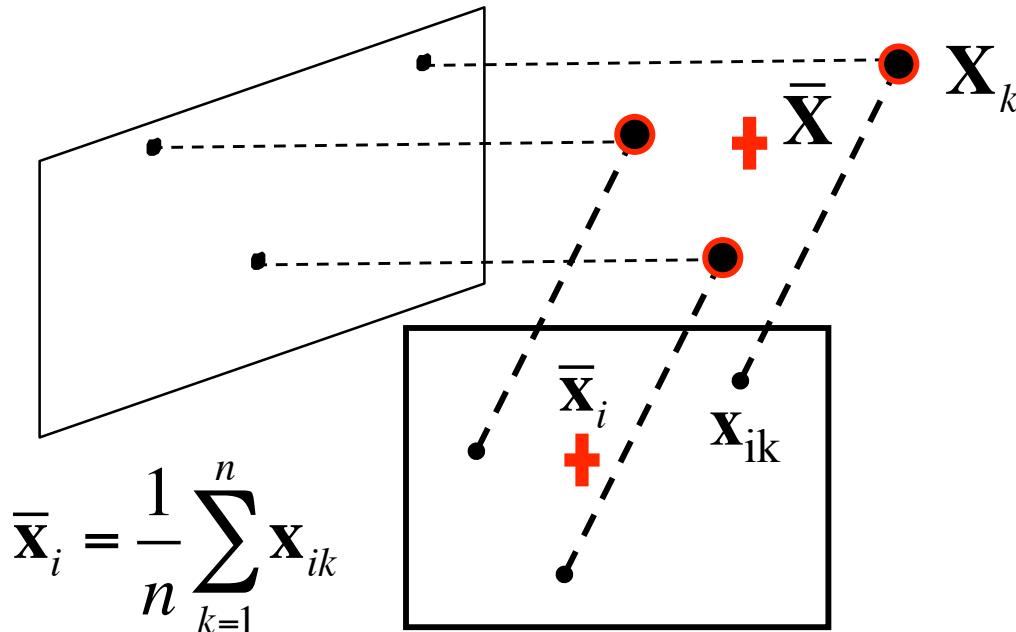


$$[\text{Eq. 5}] \quad \bar{\mathbf{x}}_i = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik}$$

A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$\begin{aligned}\hat{\mathbf{x}}_{ij} &= \mathbf{x}_{ij} - \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i - \frac{1}{n} \sum_{k=1}^n \mathbf{A}_i \mathbf{X}_k - \frac{1}{n} \sum_{k=1}^n \mathbf{b}_i \\ \mathbf{x}_{ik} &= \mathbf{A}_i \mathbf{X}_k + \mathbf{b}_i \\ &= \mathbf{A}_i \left(\mathbf{X}_j - \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \right) = \mathbf{A}_i (\mathbf{X}_j - \bar{\mathbf{X}}) \\ &= \mathbf{A}_i \hat{\mathbf{X}}_j \quad [\text{Eq. 8}]\end{aligned}$$



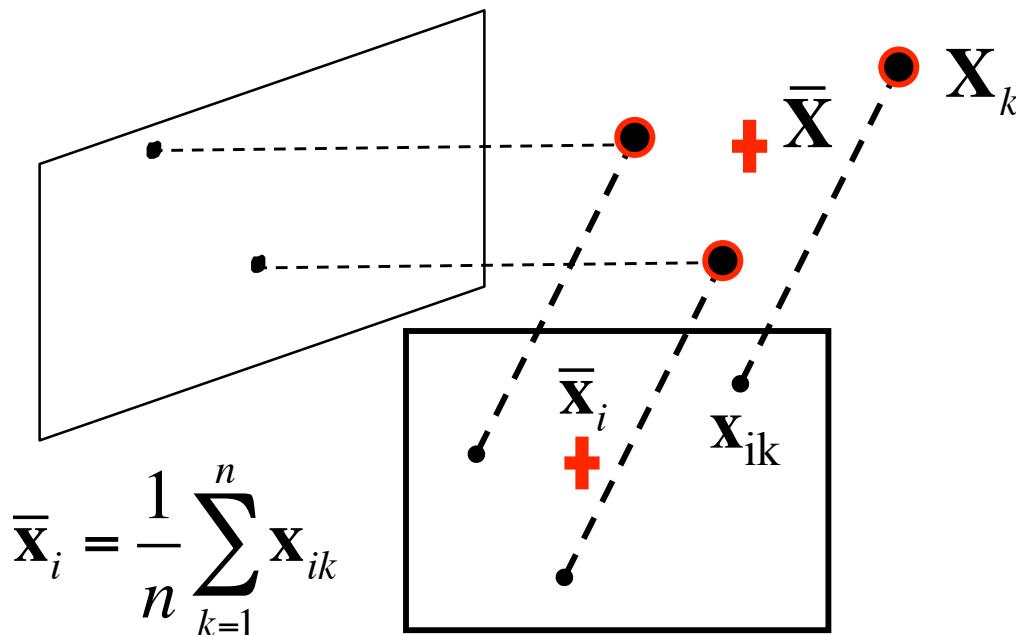
$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \quad [\text{Eq. 7}]$$

Centroid of 3D points

A factorization method - Centering the data

Thus, after centering, each **normalized** observed point is related to the 3D point by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \hat{\mathbf{x}}_j \quad [\text{Eq. 8}]$$



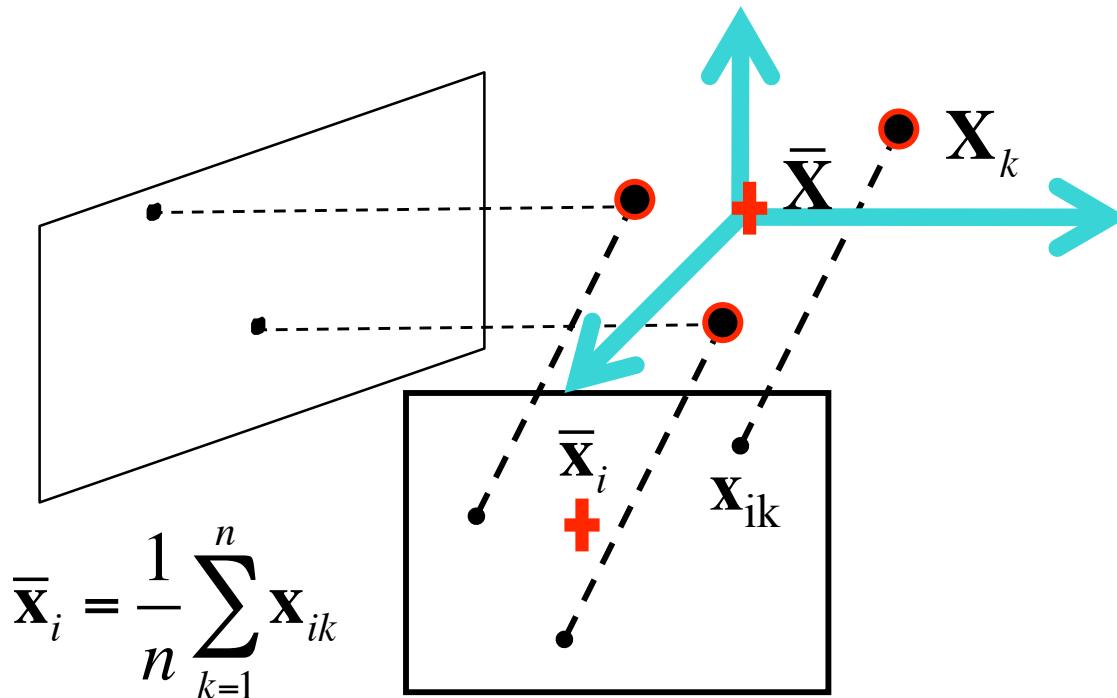
$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \quad [\text{Eq. 7}]$$

Centroid of 3D points

A factorization method - Centering the data

If the centroid of points in 3D = center of the world reference system

$$\hat{\mathbf{X}}_{ij} = \mathbf{A}_i \hat{\mathbf{X}}_j = \mathbf{A}_i \mathbf{X}_j \quad [\text{Eq. 9}]$$



$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \quad [\text{Eq. 7}]$$

Centroid of 3D points

A factorization method - factorization

Let's create a $2m \times n$ data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ \vdots & \ddots & & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}$$

points (n)

cameras
($2m$)

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2×1 vector!

A factorization method - factorization

Let's create a $2m \times n$ data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ \vdots & \ddots & & \vdots \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

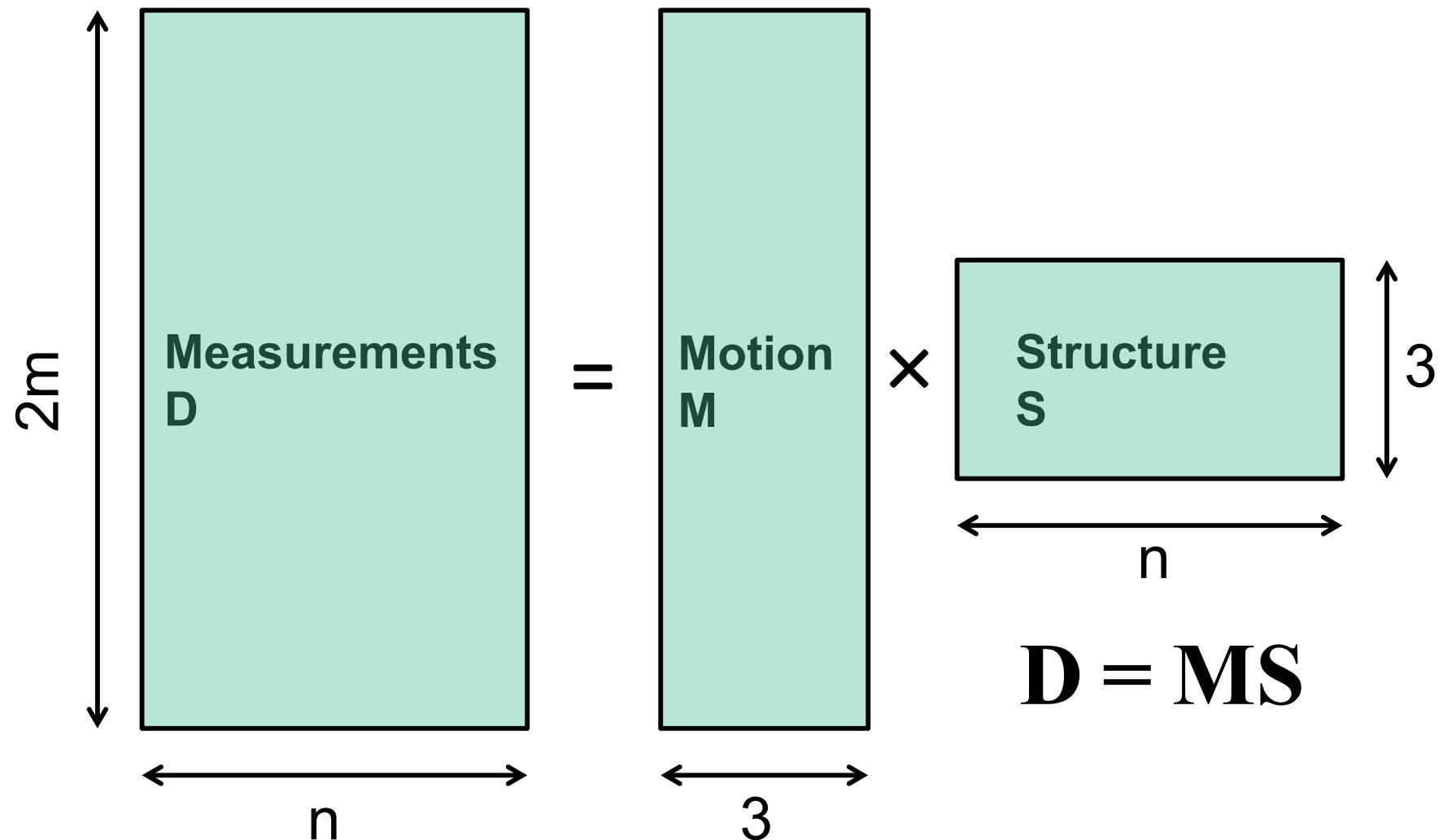
$(2m \times n)$ cameras $(2m \times 3)$ points $(3 \times n)$ \mathbf{S}

[Eq. 10]

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2×1 vector!
 \mathbf{A}_i is 2×3 and \mathbf{X}_j is 3×1

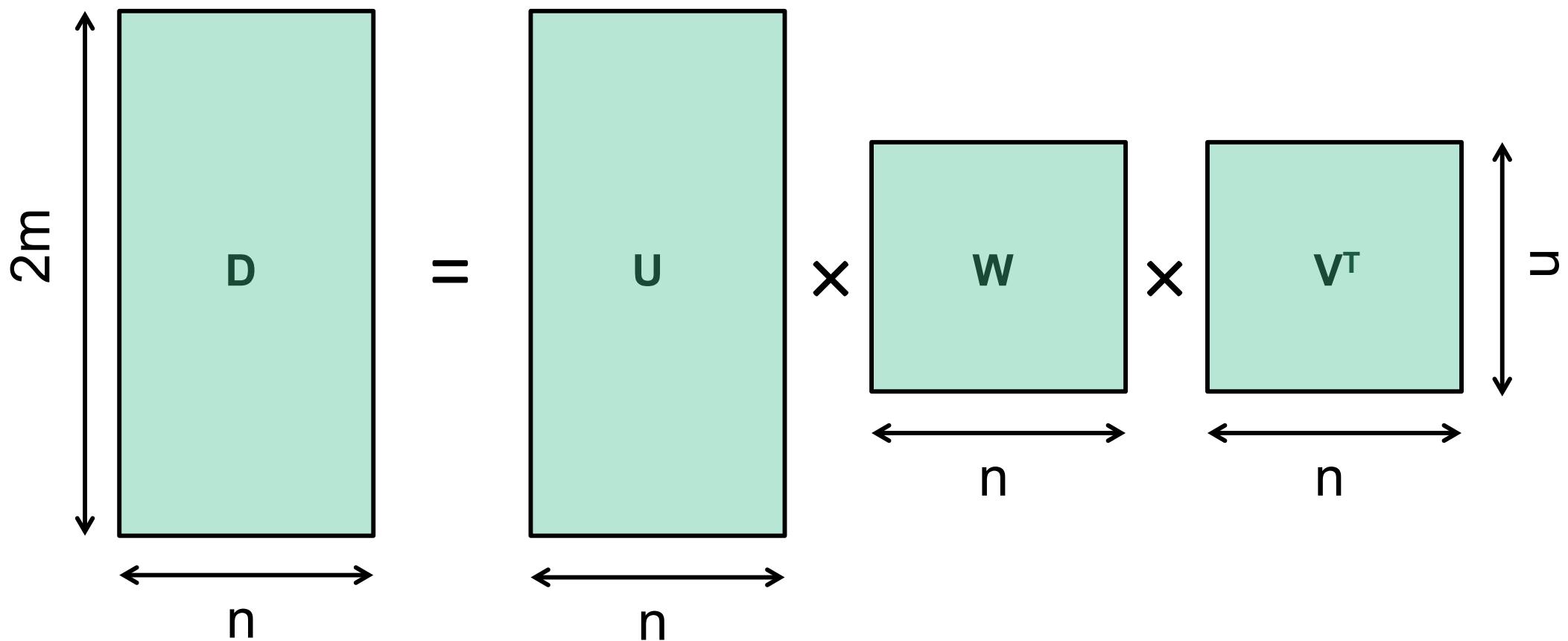
The measurement matrix $\mathbf{D} = \mathbf{M} \mathbf{S}$ has rank 3
(it's a product of a $2m \times 3$ matrix and $3 \times n$ matrix)

Factorizing the Measurement Matrix



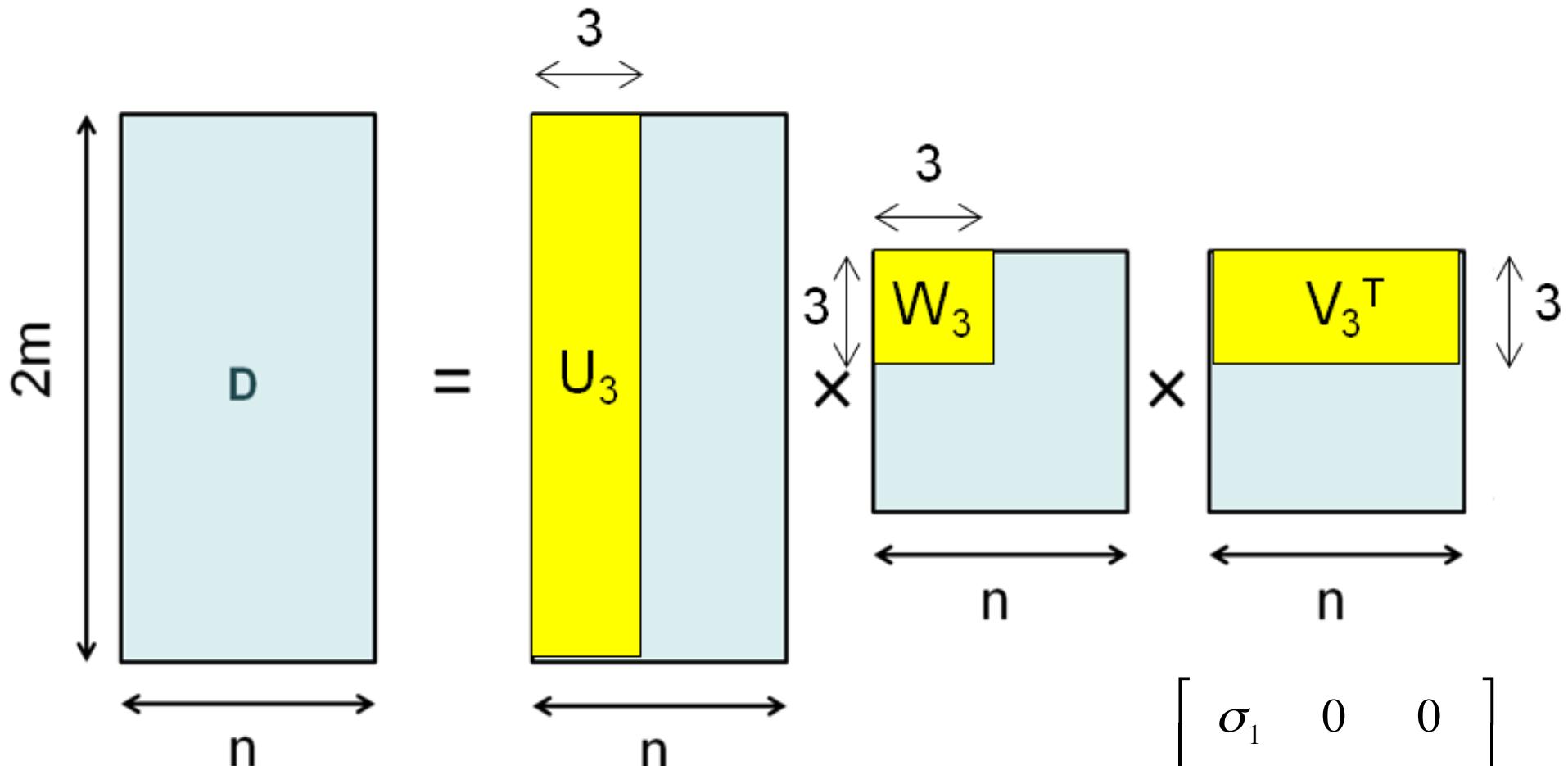
Factorizing the Measurement Matrix

- How to factorize D? By computing the Singular value decomposition of D!



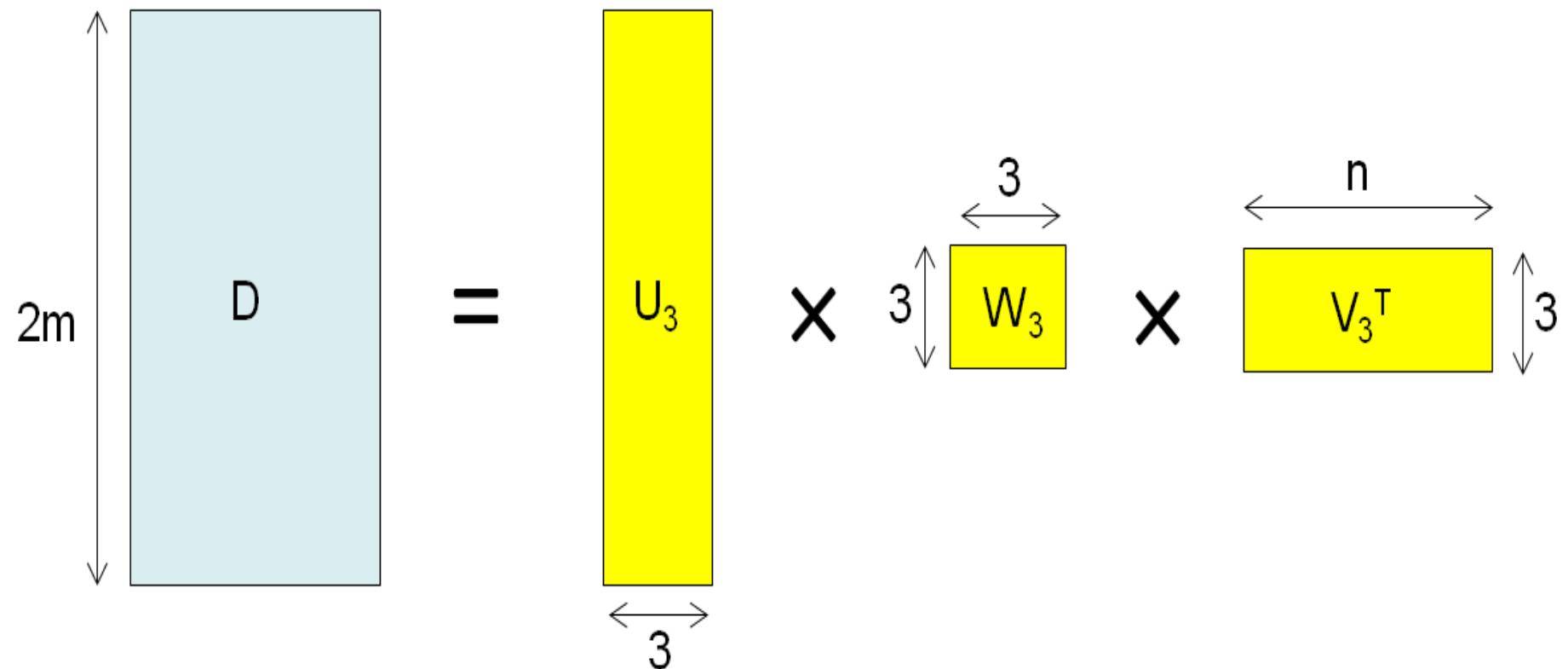
Factorizing the Measurement Matrix

Since rank (D)=3, there are only 3 non-zero singular values σ_1 , σ_2 and σ_3

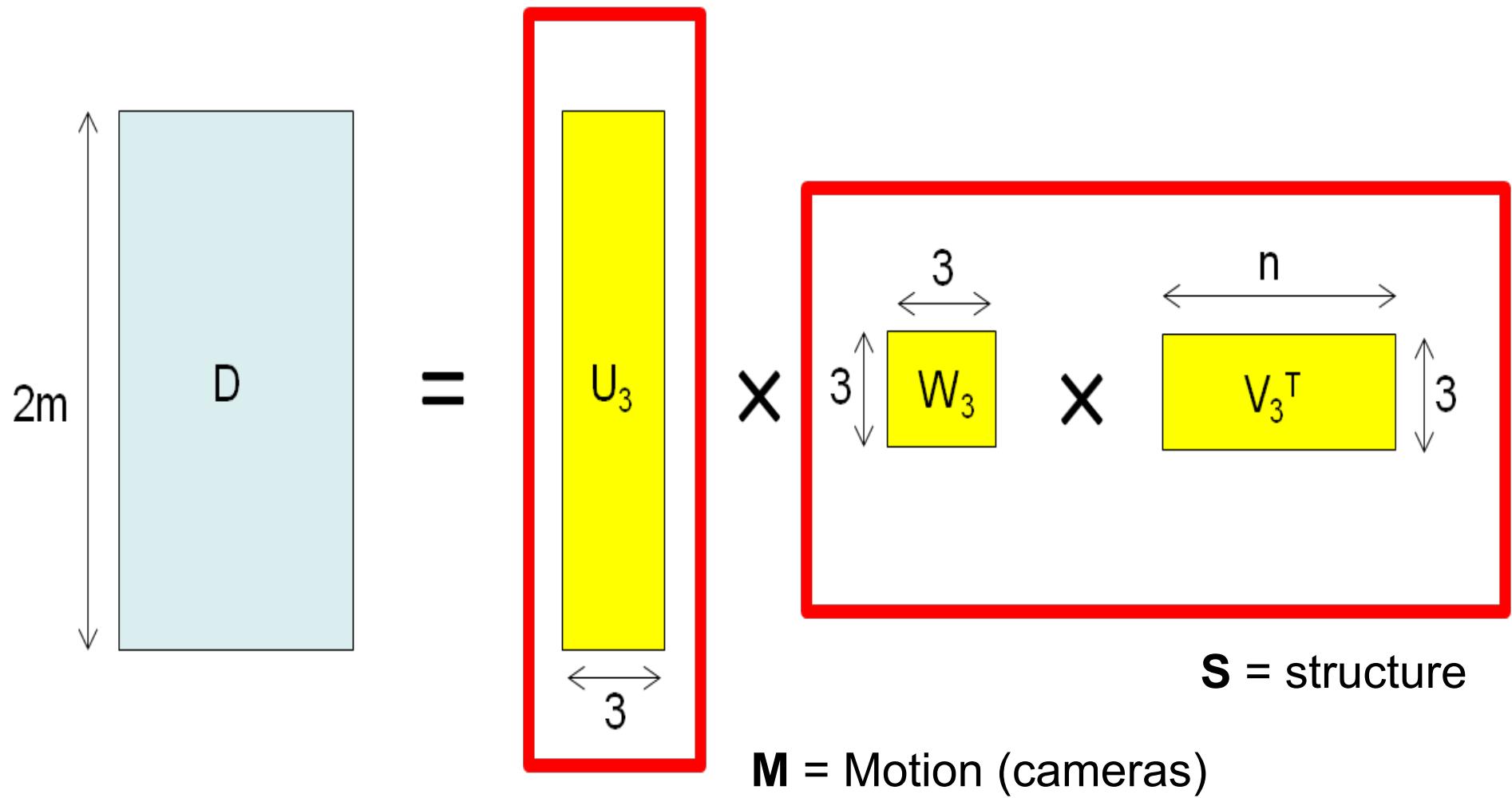


Where $W_3 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$ [Eq. 11]

Factorizing the Measurement Matrix



Factorizing the Measurement Matrix



$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T = \mathbf{U}_3 (\mathbf{W}_3 \mathbf{V}_3^T) = \mathbf{M} \mathbf{S} \quad [\text{Eq. 12}]$$

Factorizing the Measurement Matrix

$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T = \mathbf{U}_3 (\mathbf{W}_3 \mathbf{V}_3^T) = \mathbf{M} \mathbf{S} \quad [\text{Eq. 12}]$$

What is the issue here? \mathbf{D} has rank>3 because of:

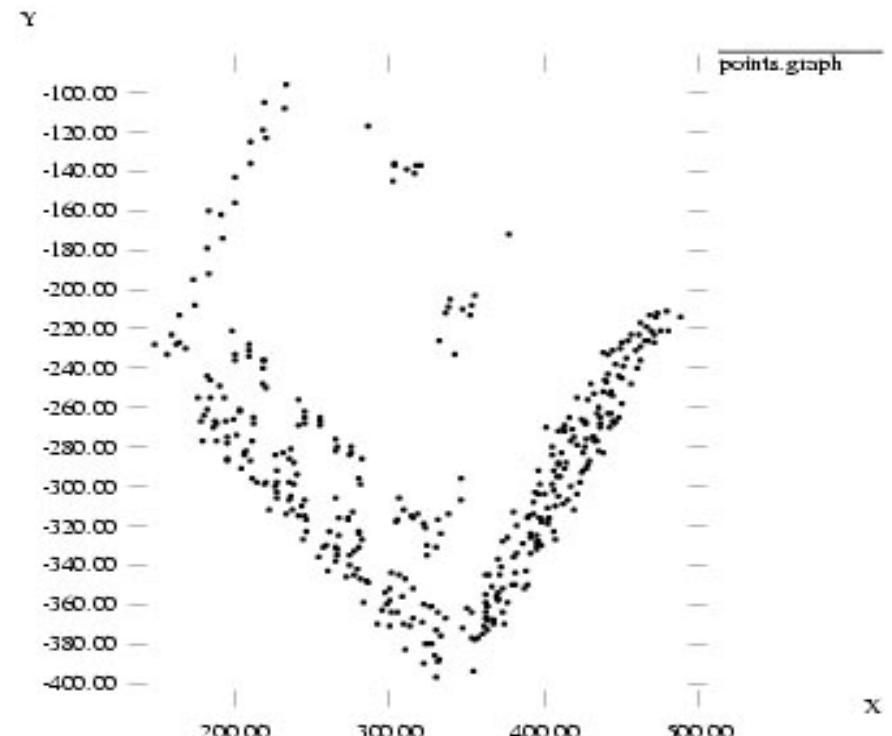
- measurement noise
- affine approximation

Theorem: When \mathbf{D} has a rank greater than 3, $\mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T$ is the best possible rank-3 approximation of \mathbf{D} in the sense of the Frobenius norm.

$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T \quad \left\{ \begin{array}{l} \mathbf{M} \approx \mathbf{U}_3 \\ \mathbf{S} \approx \mathbf{W}_3 \mathbf{V}_3^T \end{array} \right.$$

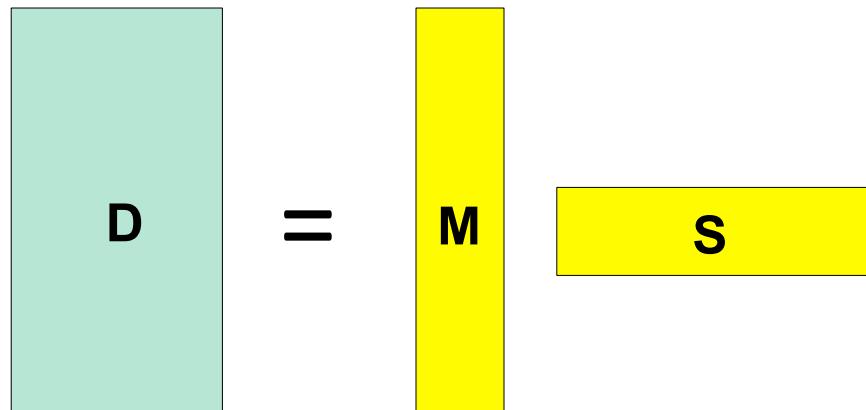
$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m, n\}} \sigma_i^2}$$

Reconstruction results



C. Tomasi and T. Kanade. [Shape and motion from image streams under orthography: A factorization method](#). *IJCV*, 9(2):137-154, November 1992.

Affine Ambiguity

$$D = M S$$


Affine Ambiguity

$$\mathbf{D} = \mathbf{M} \mathbf{H} \times \mathbf{H}^{-1} \mathbf{S}$$

The diagram illustrates the decomposition of a matrix \mathbf{D} into four components: \mathbf{M} , \mathbf{H} , \mathbf{H}^{-1} , and \mathbf{S} . The matrix \mathbf{D} is represented by a light green rectangle on the left. An equals sign follows it. To the right of the equals sign is a vertical yellow rectangle labeled \mathbf{M} . To the right of \mathbf{M} is a horizontal yellow rectangle containing \mathbf{H} and \mathbf{H}^{-1} separated by a times symbol (\times). To the right of \mathbf{H}^{-1} is another horizontal yellow rectangle labeled \mathbf{S} . Below the \mathbf{H} and \mathbf{H}^{-1} terms is a brace labeled \mathbf{M}^* , and below the \mathbf{H}^{-1} and \mathbf{S} terms is a brace labeled \mathbf{S}^* .

- The decomposition is not unique. We get the same \mathbf{D} by applying the transformations:

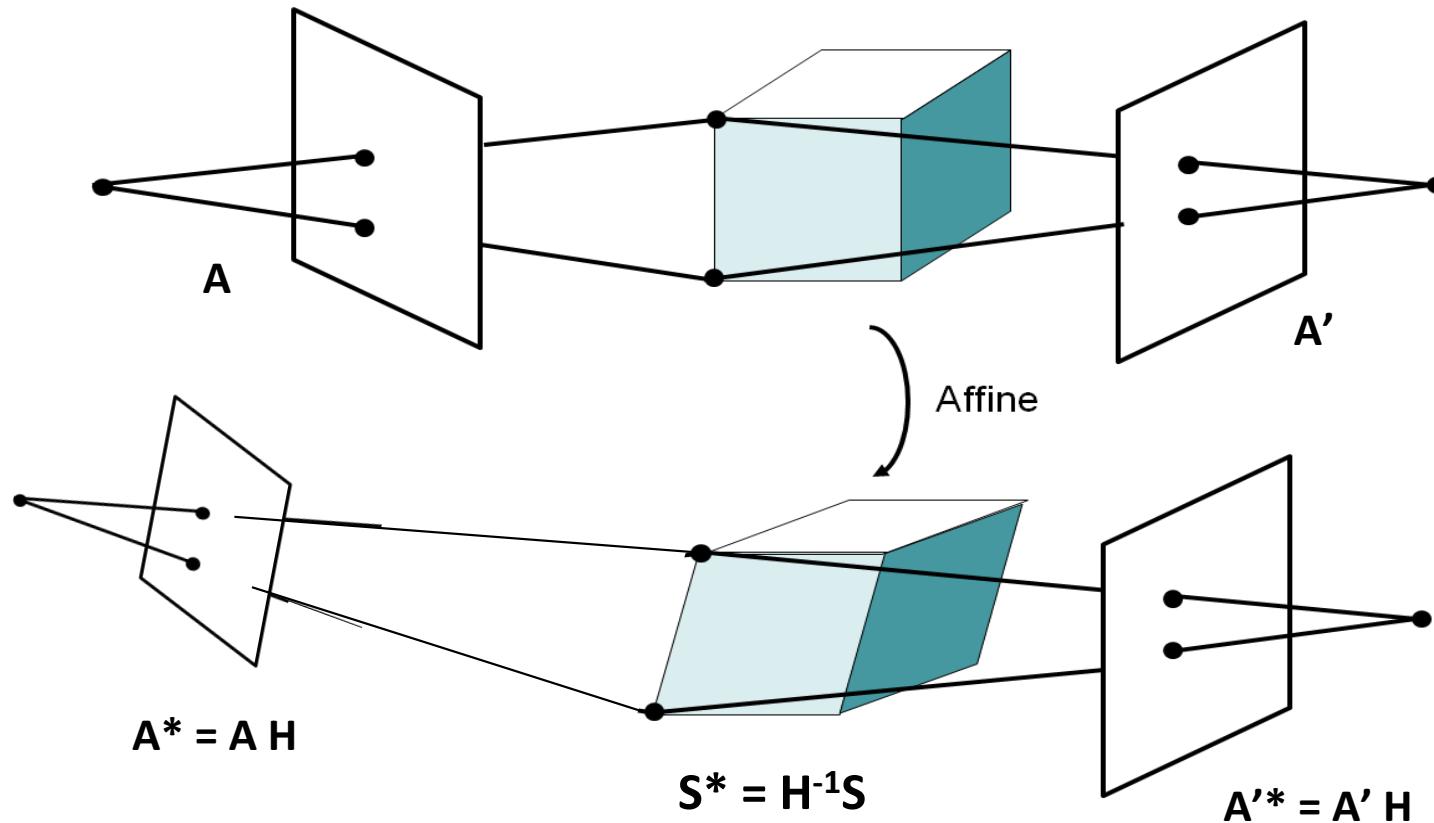
$$\mathbf{M}^* = \mathbf{M} \mathbf{H}$$

$$\mathbf{S}^* = \mathbf{H}^{-1} \mathbf{S}$$

where \mathbf{H} is an arbitrary 3×3 matrix describing an affine transformation

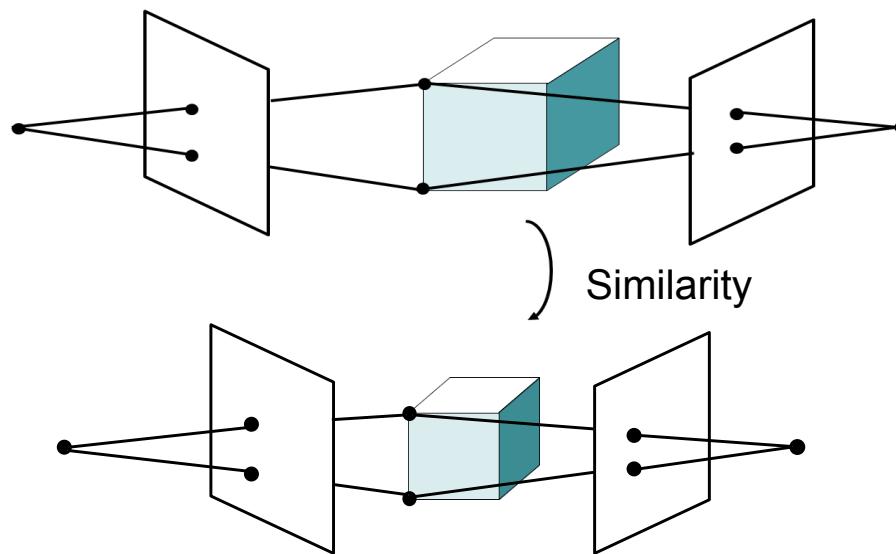
- Additional constraints must be enforced to resolve this ambiguity

Affine Ambiguity



Similarity Ambiguity

- The scene is determined by the images only up a **similarity transformation** (rotation, translation and scaling)
- This is called **metric reconstruction**

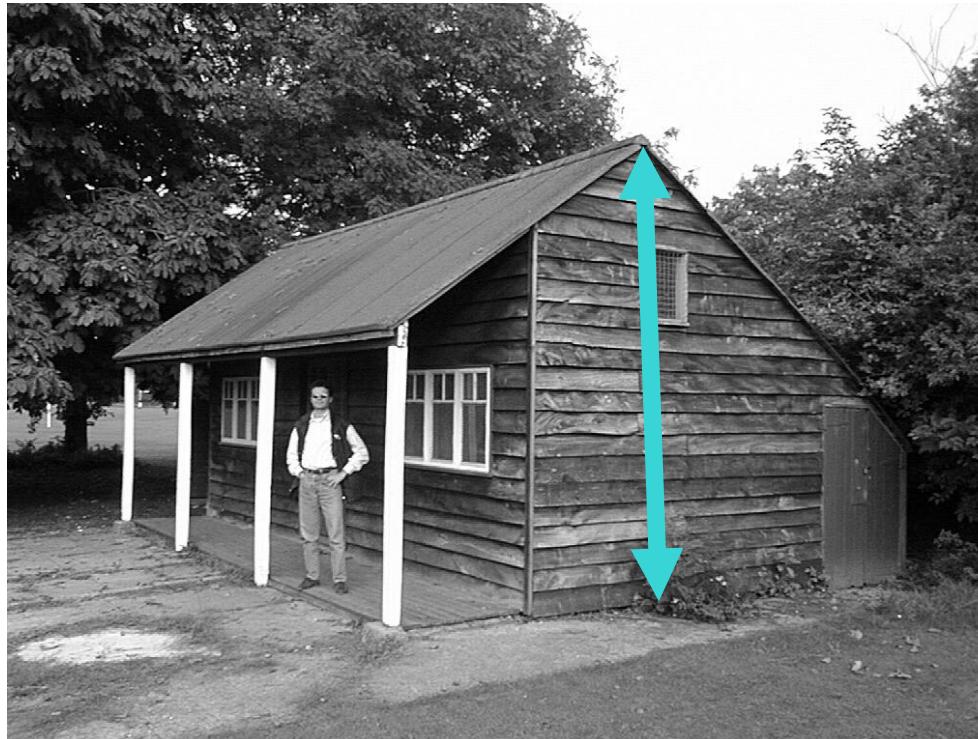


- The ambiguity exists even for (intrinsically) calibrated cameras
- For calibrated cameras, the similarity ambiguity is the **only** ambiguity

[Longuet-Higgins '81]

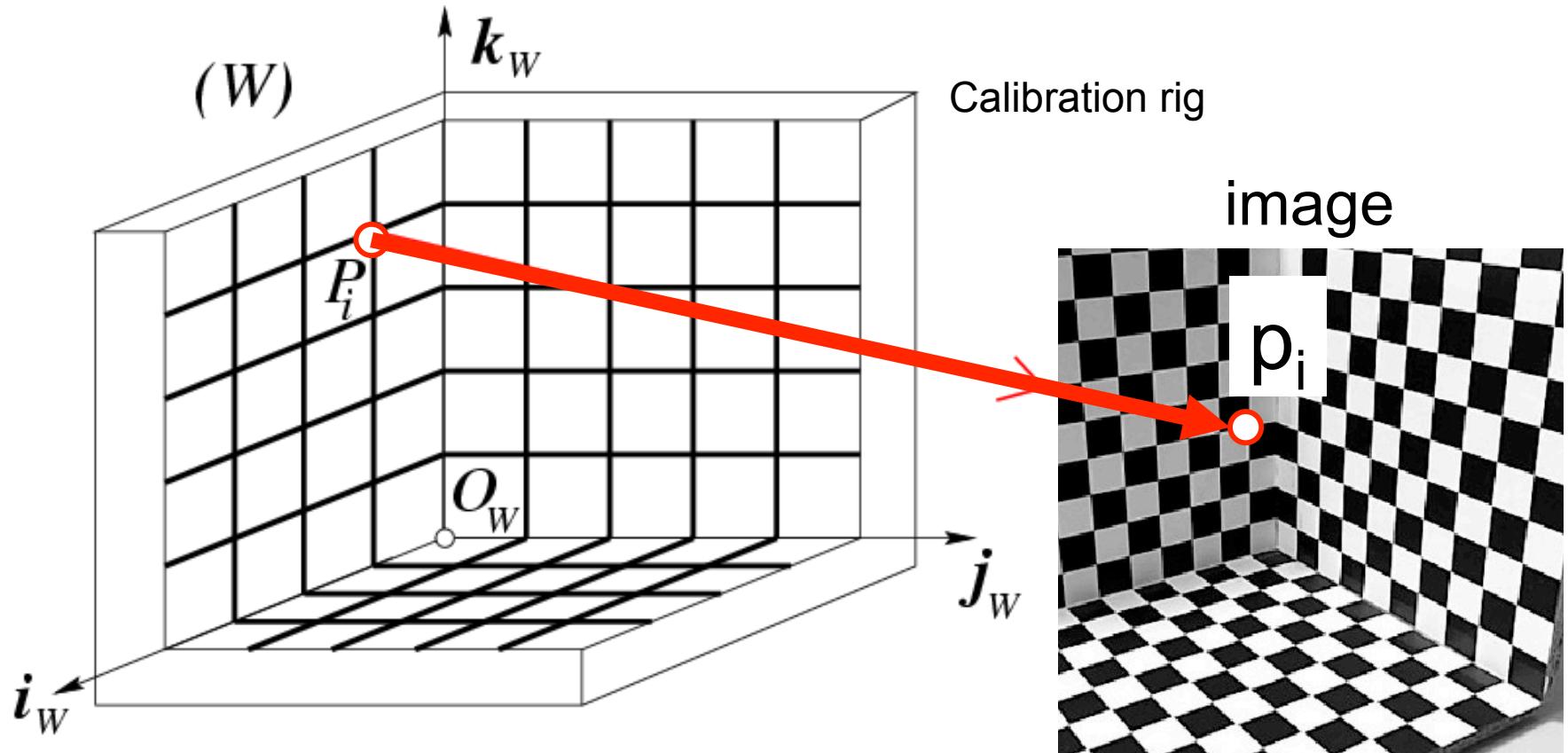
Similarity Ambiguity

- It is impossible, based on the images alone, to estimate the absolute scale of the scene (i.e. house height)



<http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl>

Resolving the similarity ambiguity



While calibrating a camera, we make assumptions about the geometry of the world

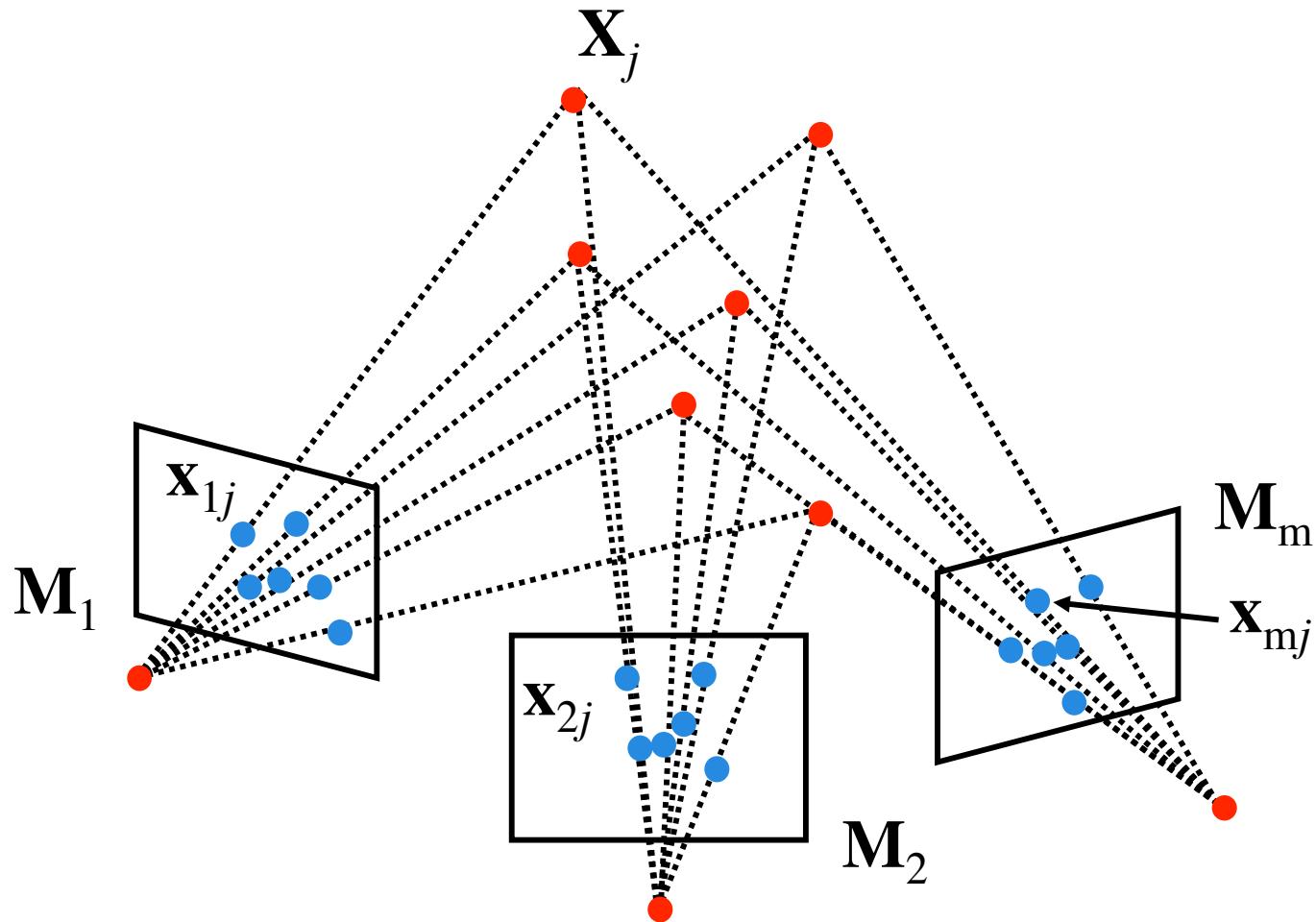
Lecture 7

Multi-view geometry



- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications

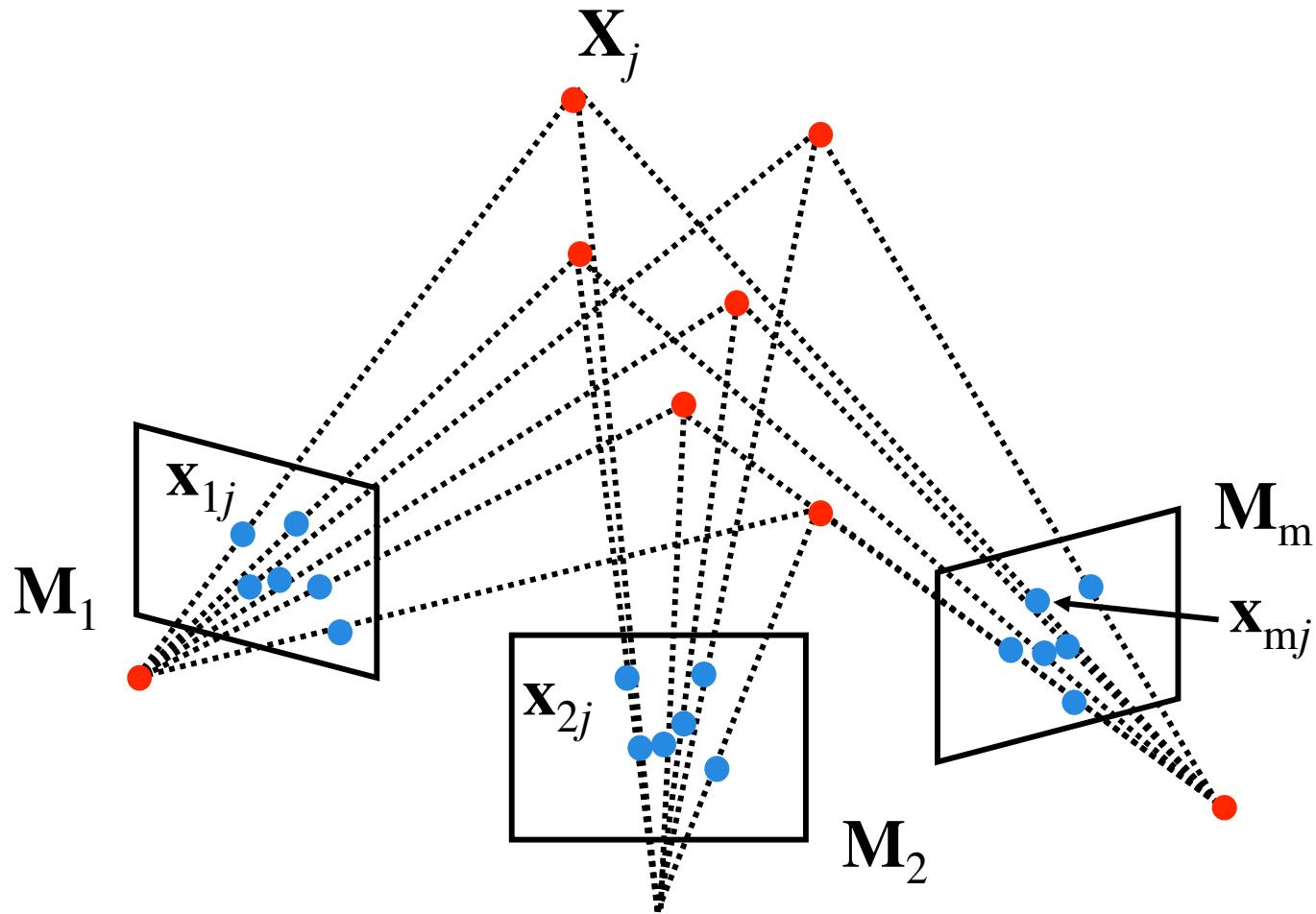
Structure from motion problem



From the $m \times n$ observations \mathbf{x}_{ij} , estimate:

- m projection matrices \mathbf{M}_i = motion
- n 3D points \mathbf{X}_j = structure

Structure from motion problem



m cameras $M_1 \dots M_m$

$$M_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix}$$

The Structure-from-Motion Prob

for $i = 1, \dots, m$

Given m images of n fixed points X_j we can write

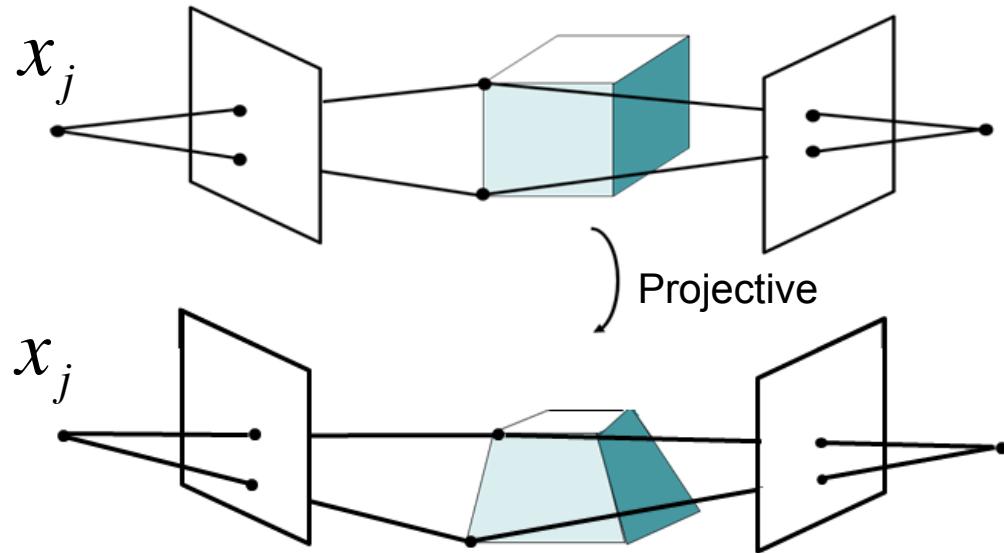
$$x_{ij} = M_i X_j \quad \begin{matrix} \text{for } i = 1, \dots, m \\ \text{N. of cameras} \end{matrix} \quad \begin{matrix} \text{and } j = 1, \dots, n \\ \text{N. of points} \end{matrix}$$

Problem: estimate m 3×4 matrices M_i and n positions X_j from $m \times n$ observations x_{ij} .

- If the cameras are not calibrated, cameras and points can only be recovered up to a 4×4 projective (16 parameters)
- Given two cameras, how many points are needed?
- How many equations and how many unknowns?

$2m \times n$ equations in $11m + 3n - 16$ unknowns

Structure from Motion Ambiguities



- In the general case (nothing is known) the ambiguity is expressed by an arbitrary **affine** or **projective transformation**

$$x_j = M_i \boxed{X_j}$$

↓

$$\boxed{H X_j}$$

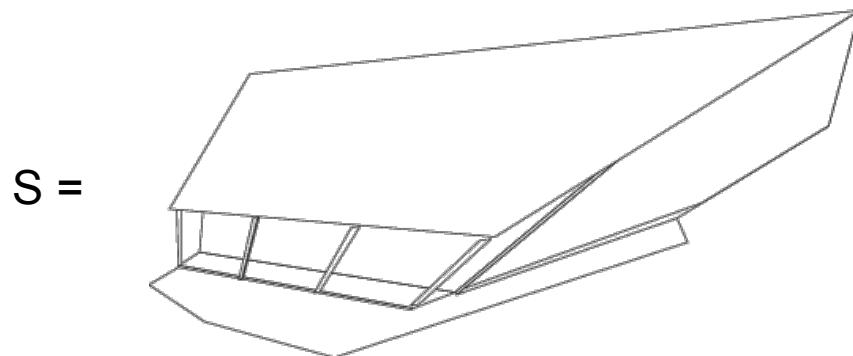
$$\boxed{M_i} = K_i [R_i \ T_i]$$

↓

$$\boxed{M_i H^{-1}}$$

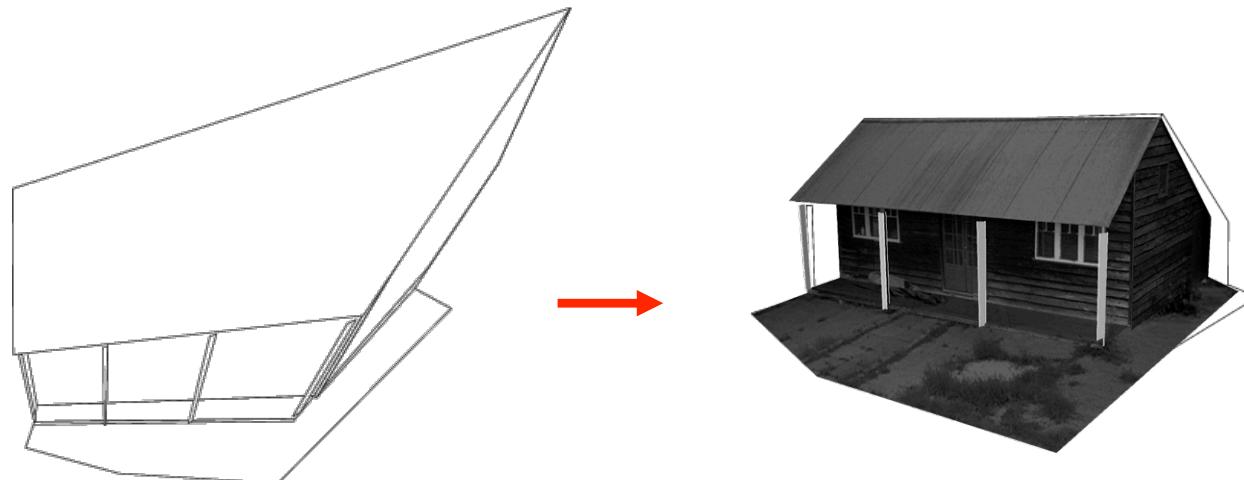
$$x_j = M_i X_j = (M_i H^{-1})(H X_j)$$

Projective Ambiguity



Metric reconstruction (upgrade)

- The problem of recovering the metric reconstruction from the perspective one is called **self-calibration**



Structure-from-Motion methods

1. Recovering structure and motion up to perspective ambiguity

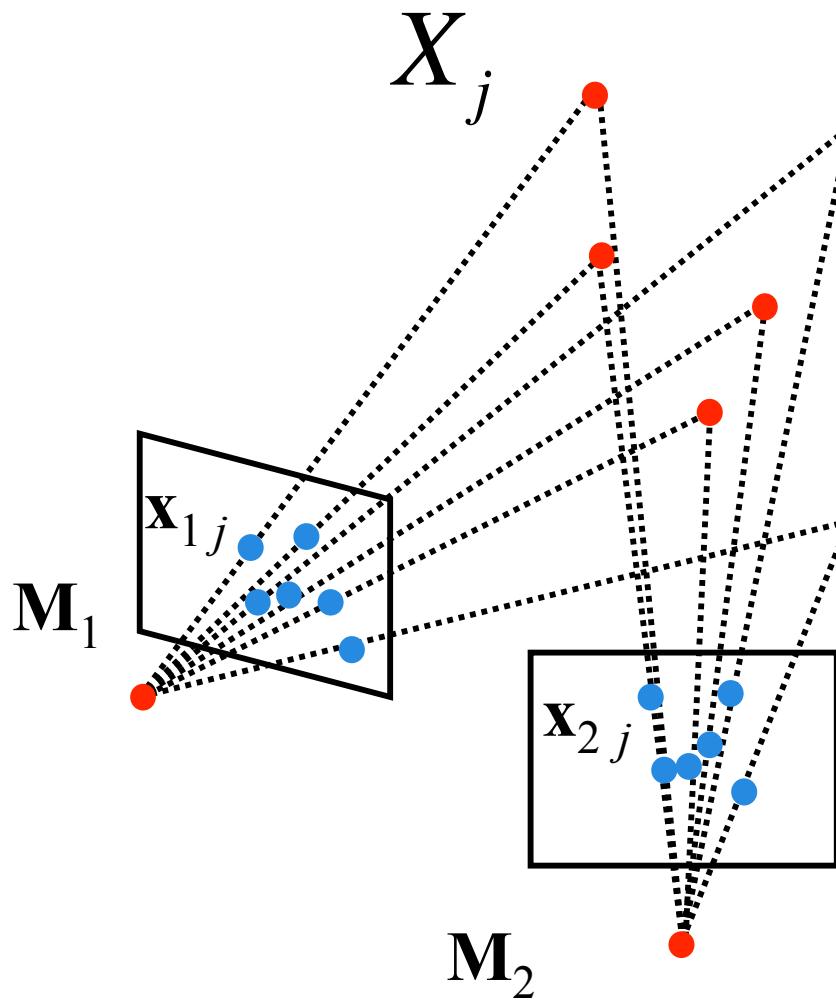
- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

2. Resolving the perspective ambiguity

Algebraic approach (2-view case)

1. Compute the fundamental matrix F from two views
2. Use F to estimate projective cameras
3. Use these cameras to triangulate and estimate points in 3D

Algebraic approach (2-view case)



$$x_{1j} = M_1 X_j$$

$$x_{2j} = M_2 X_j$$

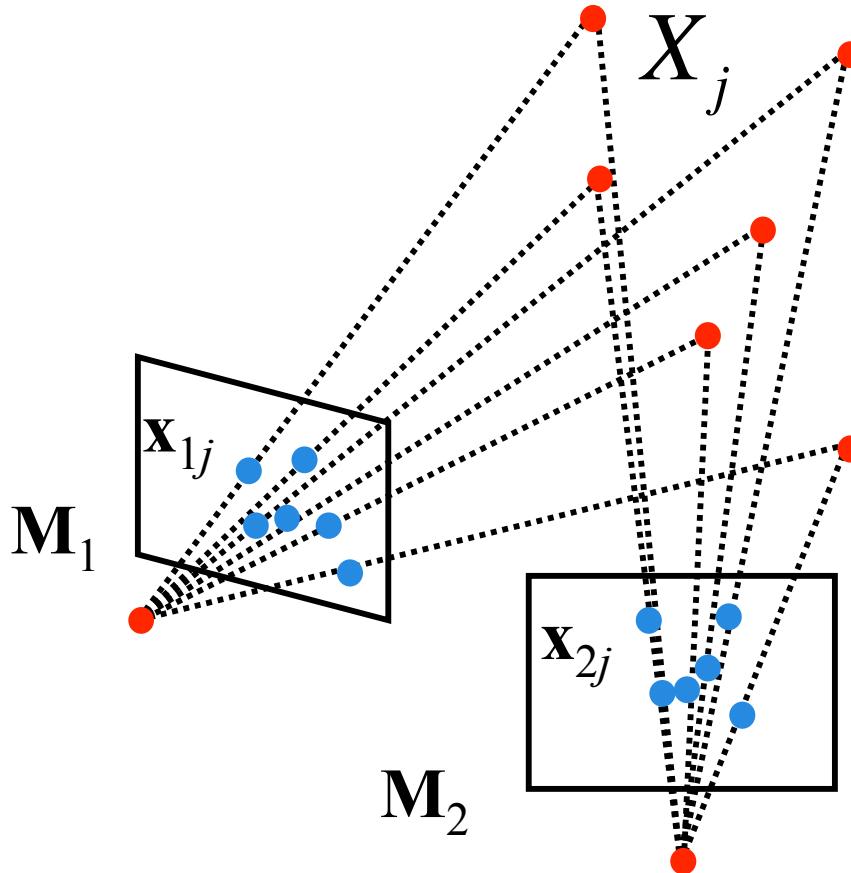
For $j = 1, \dots, n$
N. of points

From at least 8 point correspondences, compute F associated to camera 1 and 2

Algebraic approach (2-view case)

1. Compute the fundamental matrix F from two views
(eg. 8 point algorithm)
2. Use F to estimate projective cameras
3. Use these cameras to triangulate and estimate points in 3D

Algebraic approach (2-view case)



$$x_{1j} = M_1 X_j$$

$$x_{2j} = M_2 X_j$$

For $j = 1, \dots, n$
N. of points

Because of the projective ambiguity, we can always apply a projective transformation H such that:

$$M_1 H^{-1} = [I \quad 0]$$

[Eq. 3] Canonical perspective camera

$$M_2 H^{-1} = [A \quad b]$$

[Eq. 4]

Algebraic approach (2-view case)

- Call \mathbf{X} a generic 3D point \mathbf{X}_{ij}
- Call \mathbf{x} and \mathbf{x}' the corresponding observations to camera 1 and respectively

$$\left\{ \begin{array}{l} \tilde{M}_1 = M_1 H^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \quad \mathbf{x} = M_1 \mathbf{X} = M_1 H^{-1} H \mathbf{X} = [\mathbf{I} | \mathbf{0}] \tilde{\mathbf{X}} \quad [\text{Eq. 6}] \\ \tilde{M}_2 = M_2 H^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \quad \mathbf{x}' = M_2 \mathbf{X} = M_2 H^{-1} H \mathbf{X} = [\mathbf{A} | \mathbf{b}] \tilde{\mathbf{X}} \\ \tilde{\mathbf{X}} = \mathbf{H} \mathbf{X} \end{array} \right.$$

$$\mathbf{x}' = [\mathbf{A} | \mathbf{b}] \tilde{\mathbf{X}} = [\mathbf{A} | \mathbf{b}] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} = \mathbf{A} [\mathbf{I} | \mathbf{0}] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} + \mathbf{b} = \mathbf{A} [\mathbf{I} | \mathbf{0}] \tilde{\mathbf{X}} + \mathbf{b} = \mathbf{Ax} + \mathbf{b} \quad ?$$

[Eq. 7]

$$\mathbf{x}' \times \mathbf{b} = (\mathbf{Ax} + \mathbf{b}) \times \mathbf{b} = \mathbf{Ax} \times \mathbf{b} \quad [\text{Eq. 8}]$$

$$\mathbf{x}'^T \cdot (\mathbf{x}' \times \mathbf{b}) = \mathbf{x}'^T \cdot (\mathbf{Ax} \times \mathbf{b}) = 0 \quad [\text{Eq. 9}]$$

$$\mathbf{x}'^T (\mathbf{b} \times \mathbf{Ax}) = 0 \quad [\text{Eq. 10}]$$

Cross product as matrix multiplication

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times] \mathbf{b}$$

Algebraic approach (2-view case)

[Eqs. 5]

$$\left\{ \begin{array}{l} \tilde{M}_1 = M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \quad \mathbf{x} = M_1 H^{-1} H \mathbf{X} = [\mathbf{I} | \mathbf{0}] \tilde{\mathbf{X}} \\ \tilde{M}_2 = M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix} \quad \mathbf{x}' = M_2 H^{-1} H \mathbf{X} = [\mathbf{A} | \mathbf{b}] \tilde{\mathbf{X}} \\ \tilde{\mathbf{X}} = \mathbf{H} \mathbf{X} \\ \vdots \end{array} \right. \quad [Eq. 6]$$

$$\mathbf{x}'^T (\mathbf{b} \times \mathbf{A} \mathbf{x}) = 0 \quad [Eq. 10]$$

$$\mathbf{x}'^T [\mathbf{b}_x] \mathbf{A} \mathbf{x} = 0 \quad \text{is this familiar?}$$

$$\mathbf{F} = [\mathbf{b}_x] \mathbf{A}$$

fundamental matrix!

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$$

Compute cameras

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b}_x] \mathbf{A} = \mathbf{b} \times \mathbf{A} \quad [\text{Eq. 11}]$$

Compute \mathbf{b} :

- Let's consider the product $\mathbf{F} \mathbf{b}$

$$\mathbf{F} \cdot \mathbf{b} = [\mathbf{b}_x] \mathbf{A} \cdot \mathbf{b} = \mathbf{b} \times \mathbf{A} \cdot \mathbf{b} = 0 \quad [\text{Eq. 12}]$$

- Since \mathbf{F} is singular, we can compute \mathbf{b} as least sq. solution of $\mathbf{F} \mathbf{b} = 0$, with $|\mathbf{b}|=1$ using SVD
- Using a similar derivation, we have that $\mathbf{b}^T \mathbf{F} = 0 \quad [\text{Eq. 12-bis}]$

Compute cameras

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b}_x] \mathbf{A} \quad \left\{ \begin{array}{l} \mathbf{F} \mathbf{b} = 0 \quad [\text{Eq. 12}] \\ \mathbf{b}^T \mathbf{F} = 0 \quad [\text{Eq. 12-bis}] \end{array} \right.$$

[Eq. 11]

Compute \mathbf{A} :

- Define: $\mathbf{A}' = -[\mathbf{b}_x] \mathbf{F}$
- Let's verify that $[\mathbf{b}_x] \mathbf{A}'$ is equal to \mathbf{F} :

Indeed: $[\mathbf{b}_x] \mathbf{A}' = -[\mathbf{b}_x][\mathbf{b}_x] \mathbf{F} = (\mathbf{b} \mathbf{b}^T - |\mathbf{b}|^2 \mathbf{I}) \mathbf{F} = \mathbf{b} \mathbf{b}^T \mathbf{F} + |\mathbf{b}|^2 \mathbf{F} = 0 + 1 \cdot \mathbf{F} = \mathbf{F}$

[Eq. 13]

- Thus, $\mathbf{A} = \mathbf{A}' = -[\mathbf{b}_x] \mathbf{F}$

[Eqs. 14]

$$\tilde{\mathcal{M}}_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad \tilde{\mathcal{M}}_2 = \begin{bmatrix} -[\mathbf{b}_x] \mathbf{F} & \mathbf{b} \end{bmatrix}$$

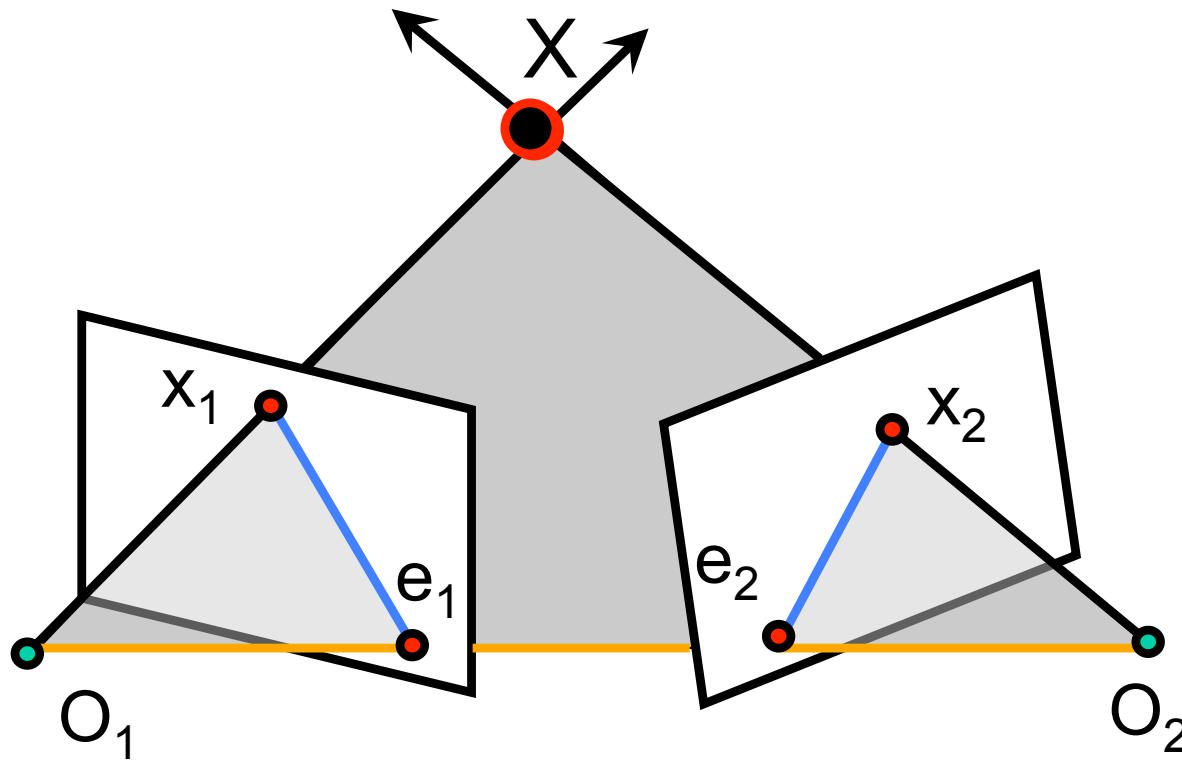
Interpretation of \mathbf{b}

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b}_x] \mathbf{A} \quad \left\{ \begin{array}{l} \mathbf{F} \mathbf{b} = 0 \quad [\text{Eq. 12}] \\ \mathbf{b}^T \mathbf{F} = 0 \quad [\text{Eq. 12-bis}] \end{array} \right.$$

[Eq. 11]

What's $\mathbf{b}??$

Epipolar Constraint [lecture 5]



$F x_2$ is the epipolar line associated with x_2 ($I_1 = F x_2$)

$F^T x_1$ is the epipolar line associated with x_1 ($I_2 = F^T x_1$)

F is singular (rank two)

$F e_2 = 0$ and $F^T e_1 = 0$

F is 3x3 matrix; 7 DOF

Interpretation of \mathbf{b}

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad \mathbf{F} = [\mathbf{b}_x] \mathbf{A} \quad \begin{cases} \mathbf{F} \mathbf{b} = 0 \\ \mathbf{b}^T \mathbf{F} = 0 \end{cases}$$

[Eq. 11]

\mathbf{b} is an epipole!

$$\tilde{\mathcal{M}}_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad \tilde{\mathcal{M}}_2 = \begin{bmatrix} - & [\mathbf{b}_x]^T \mathbf{F} & \mathbf{b} \end{bmatrix}$$



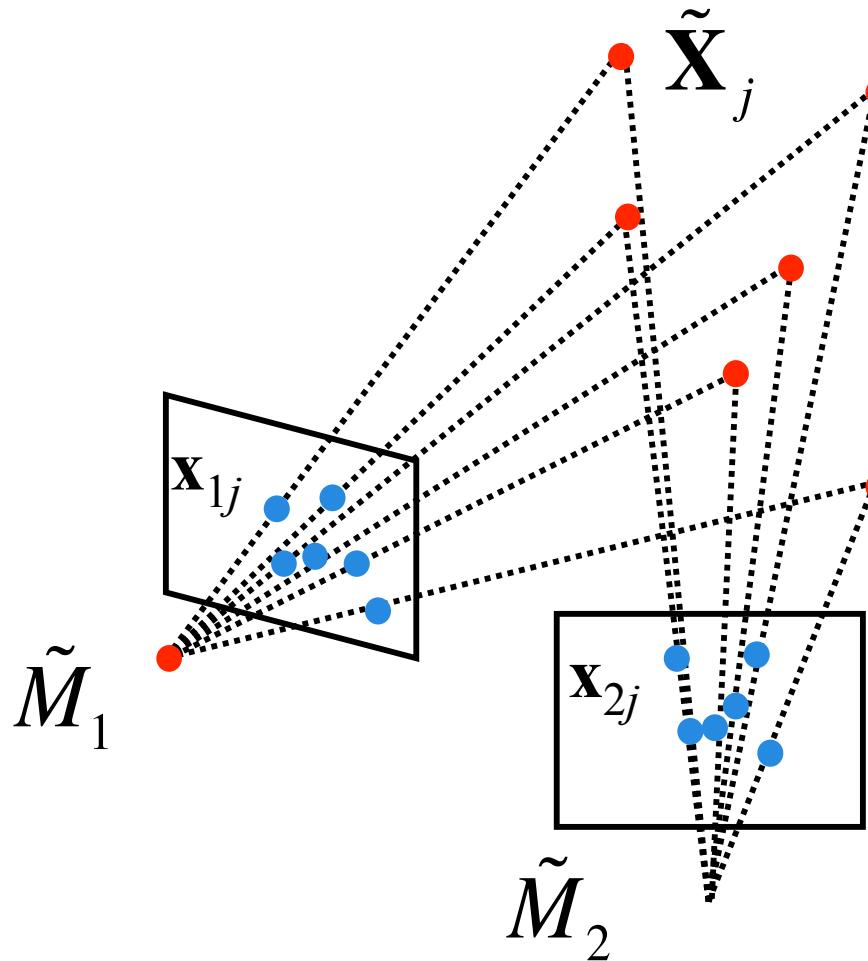
$$\tilde{\mathcal{M}}_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad \tilde{\mathcal{M}}_2 = \begin{bmatrix} - & [\mathbf{e}_x]^T \mathbf{F} & \mathbf{e} \end{bmatrix}$$

[Eq. 15] [Eq. 16]

Algebraic approach (2-view case)

1. Compute the fundamental matrix F from two views
(eg. 8 point algorithm)
2. Use F to estimate projective cameras
3. Use these cameras to triangulate and estimate
points in 3D

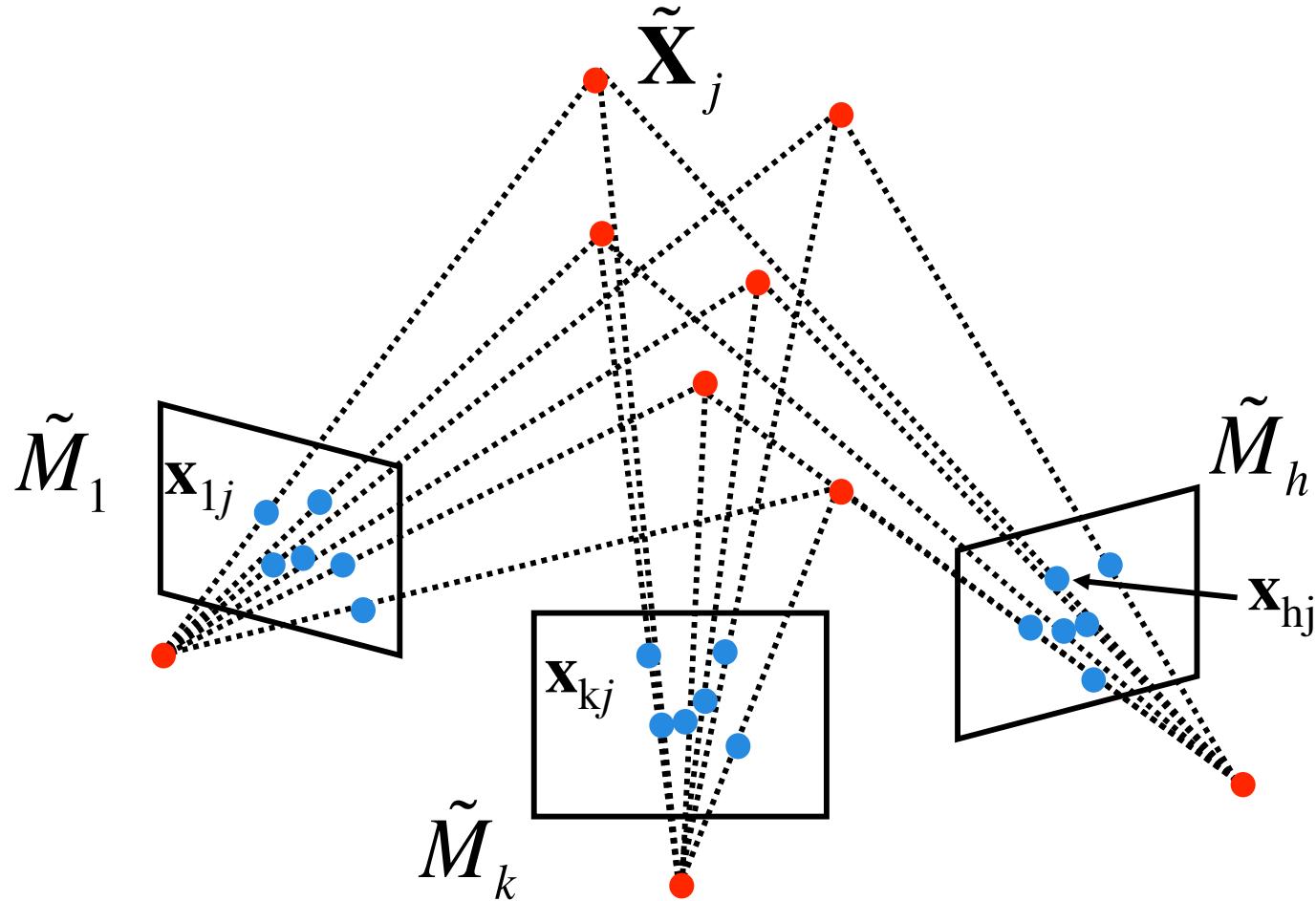
Triangulation



$$\begin{aligned}x_{1j} &= \tilde{M}_2 \tilde{\mathbf{X}}_j \\x_{2j} &= \tilde{M}_1 \tilde{\mathbf{X}}_j\end{aligned}$$

$$\begin{aligned}\tilde{M}_1 &= \begin{bmatrix} I & 0 \end{bmatrix} & \rightarrow & \tilde{\mathbf{X}}_j \quad \text{For } j = 1, \dots, n \\ \tilde{M}_2 &= \begin{bmatrix} - & [\mathbf{e}_x] \mathbf{F} & \mathbf{e} \end{bmatrix}\end{aligned}$$

Algebraic approach: the N-views case



Solutions (motion and structure) for each image pairs I_k and I_h

From I_k and $I_h \rightarrow \tilde{M}_k, \tilde{M}_h, \tilde{X}_{[k,h]}$

3D points associated to point correspondences available between I_k and I_h

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

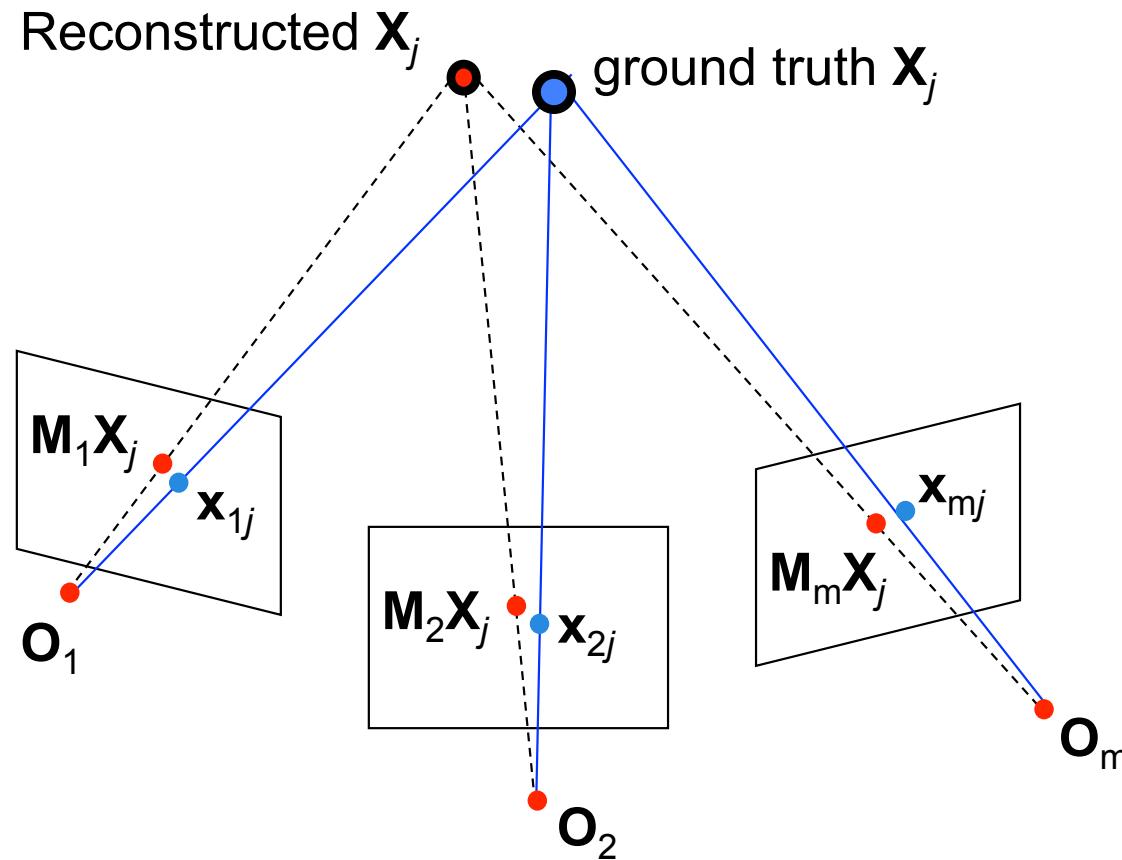
Limitations of the approaches so far

- Factorization methods assume all points are visible.
This not true if:
 - occlusions occur
 - failure in establishing correspondences
- Algebraic methods work with 2 views

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizes re-projection error

$$E(M, X) = \sum_{i=1}^m \sum_{j=1}^n D(x_{ij}, M_i X_j)^2$$



General Calibration Problem

$$E(M, X) = \sum_{i=1}^m \sum_{j=1}^n D(x_{ij}, M_i X_j)^2$$

↑
measurements ↑
parameters

D is the nonlinear mapping

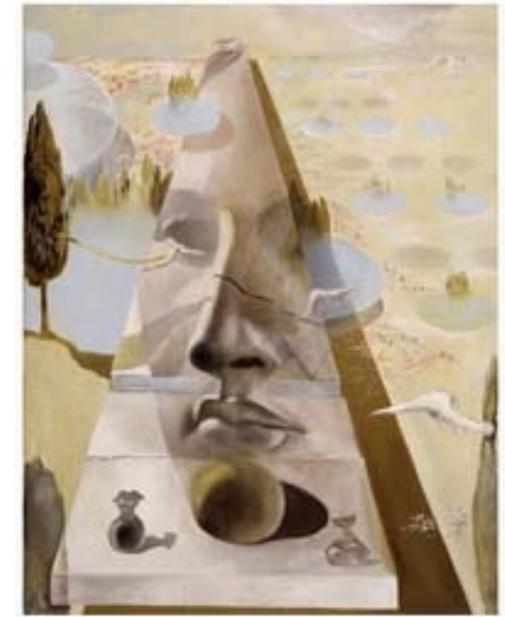
- Newton Method
- Levenberg-Marquardt Algorithm
 - Iterative, starts from initial solution
 - May be slow if initial solution far from real solution
 - Estimated solution may be function of the initial solution
 - Newton requires the computation of J, H
 - Levenberg-Marquardt doesn't require the computation of H

Bundle adjustment

- **Advantages**
 - Handle large number of views
 - Handle missing data
- **Limitations**
 - Large minimization problem (parameters grow with number of views)
 - Requires good initial condition
 - Used as the final step of SFM (i.e., after the factorization or algebraic approach)
 - Factorization or algebraic approaches provide a initial solution for optimization problem

Lecture 7

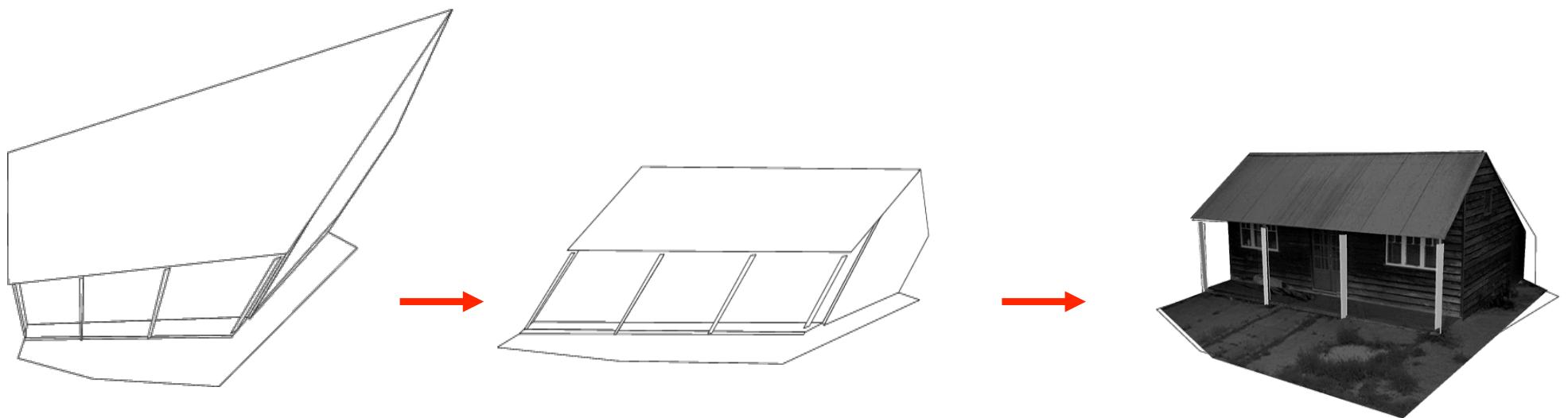
Multi-view geometry



- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications

Self-calibration

- **Self-calibration** is the problem of recovering the metric reconstruction from the perspective (or affine) reconstruction
- We can self-calibrate the camera by making some assumptions about the cameras



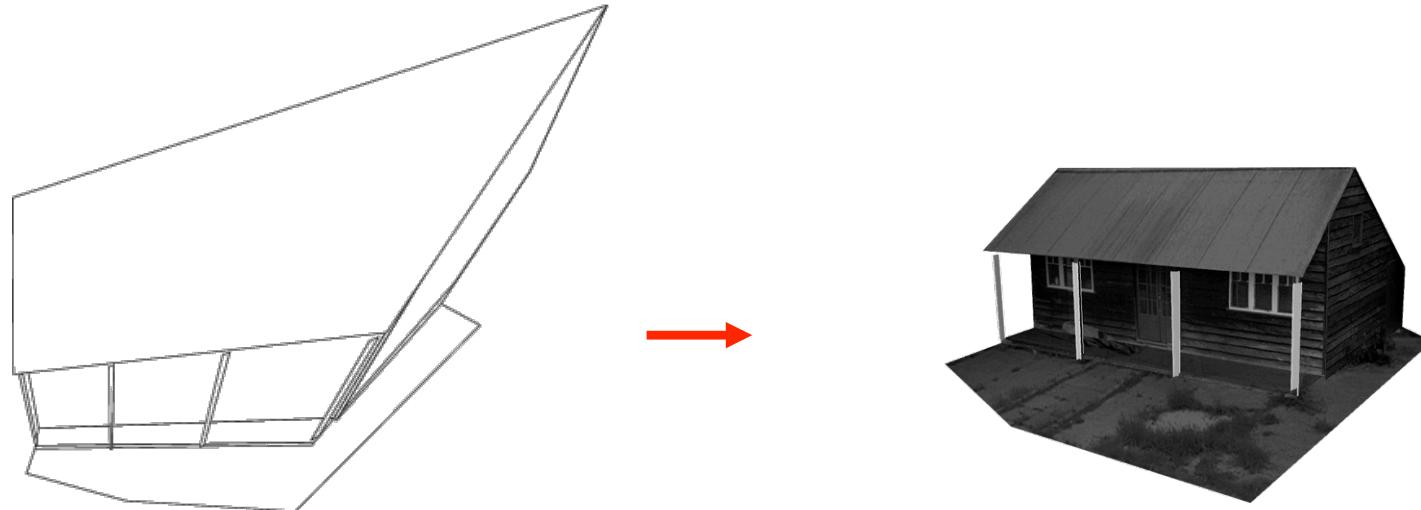
Self-calibration

[HZ] Chapters 19 “Auto-calibration”

Several approaches:

- Use single-view metrology constraints (lecture 4)
- Direct approach (Kruppa Eqs) for 2 views
- Algebraic approach
- Stratified approach

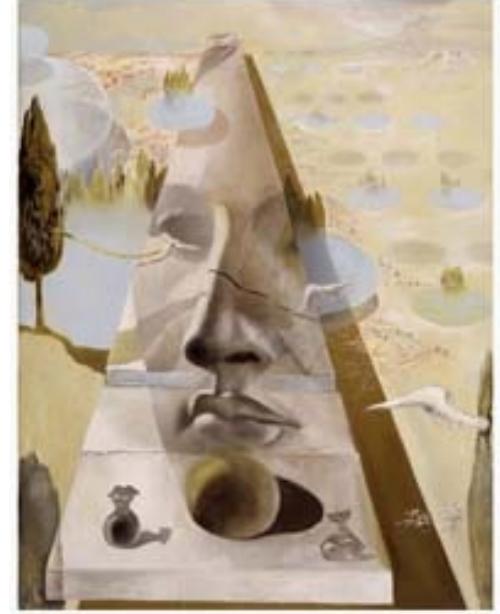
Inject information about the camera
during the bundle adjustment optimization



For calibrated cameras, the similarity ambiguity is the
only ambiguity [Longuet-Higgins '81]

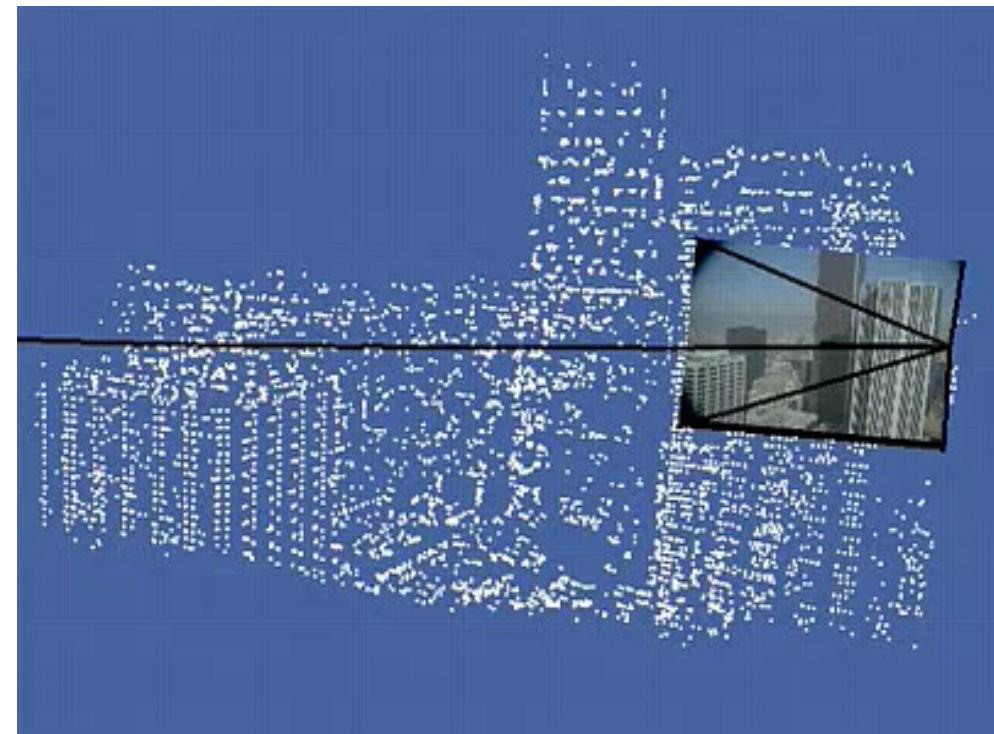
Lecture 7

Multi-view geometry



- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications

Structure from motion problem



Courtesy of Oxford **Visual Geometry Group**

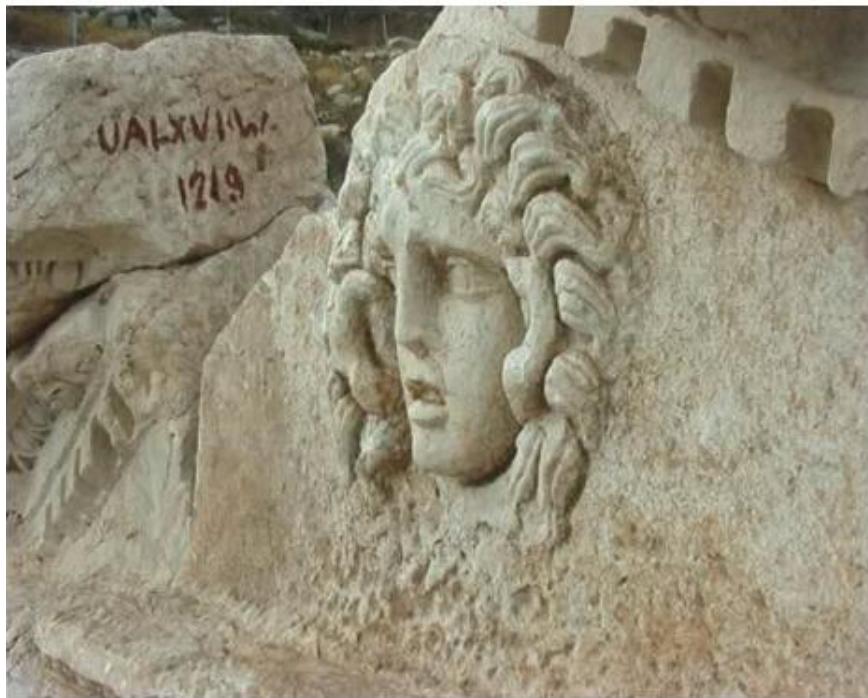
Lucas & Kanade, 81
Chen & Medioni, 92
Debevec et al., 96
Levoy & Hanrahan, 96
Fitzgibbon & Zisserman, 98
Triggs et al., 99
Pollefeys et al., 99
Kutulakos & Seitz, 99

Levoy et al., 00
Hartley & Zisserman, 00
Dellaert et al., 00
Rusinkiewic et al., 02
Nistér, 04
Brown & Lowe, 04
Schindler et al, 04
Lourakis & Argyros, 04
Colombo et al. 05

Golparvar-Fard, et al. JAEI 10
Pandey et al. IFAC , 2010
Pandey et al. ICRA 2011
Microsoft's PhotoSynth
Snavely et al., 06-08
Schindler et al., 08
Agarwal et al., 09
Frahm et al., 10

Reconstruction and texture mapping

M. Pollefeys et al 98---



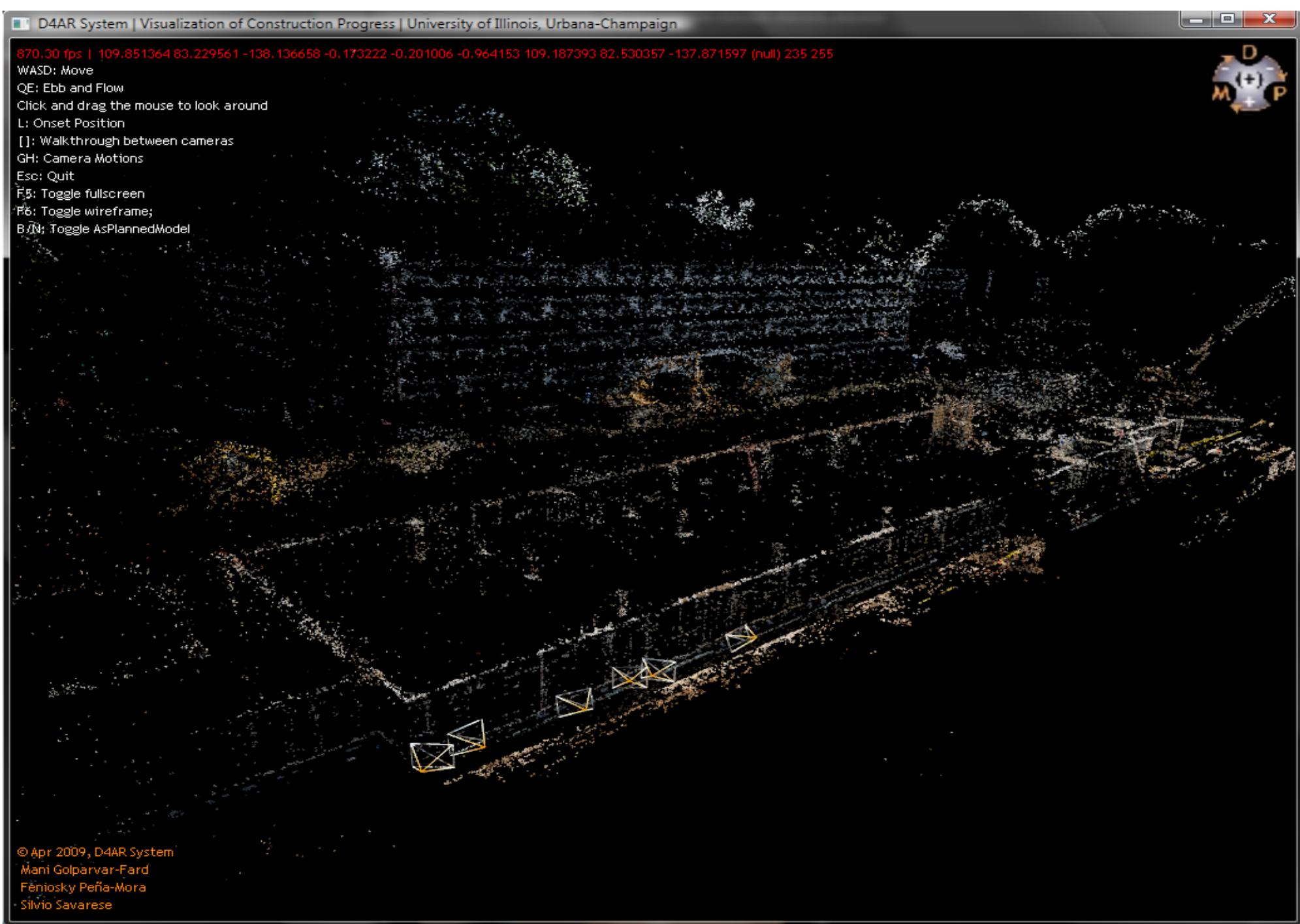
Incremental reconstruction of construction sites

Initial pair – 2168 & Complete Set 62,323 points, 160 images

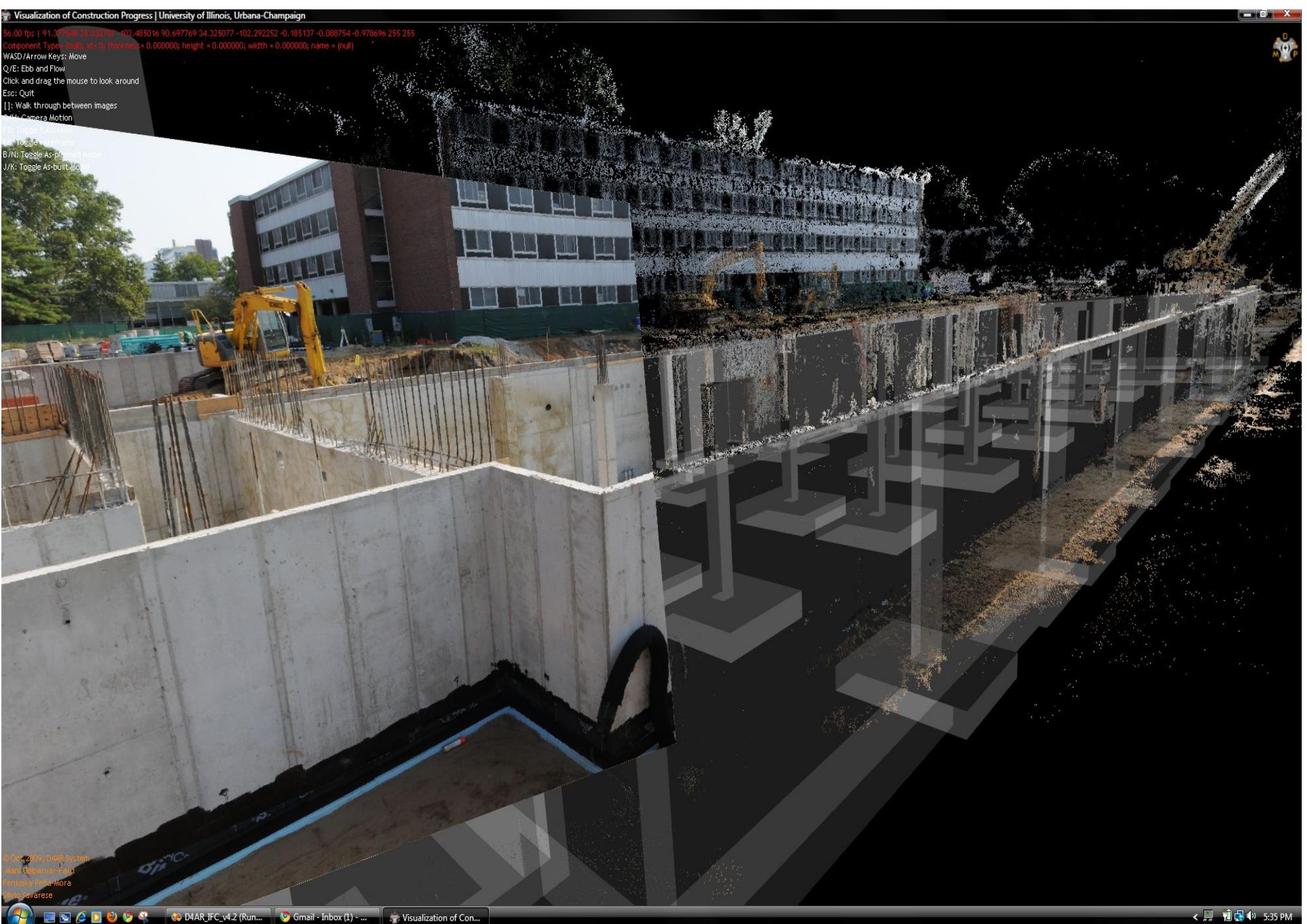
Golparvar-Fard, Pena-Mora, Savarese 2008



Reconstructed scene + Site photos



Reconstructed scene + Site photos



Results and applications

Noah Snavely, Steven M. Seitz, Richard Szeliski, "[Photo tourism: Exploring photo collections in 3D](#)," ACM Transactions on Graphics (SIGGRAPH Proceedings), 2006,



Next lecture

- Active Stereo & Volumetric Stereo

Appendix

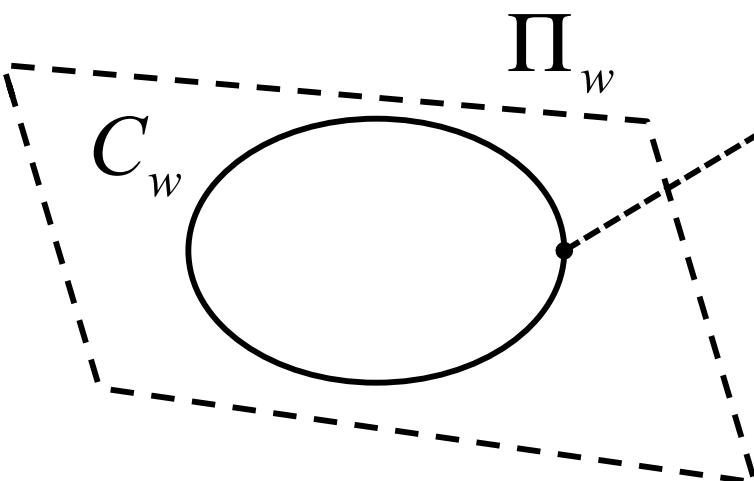
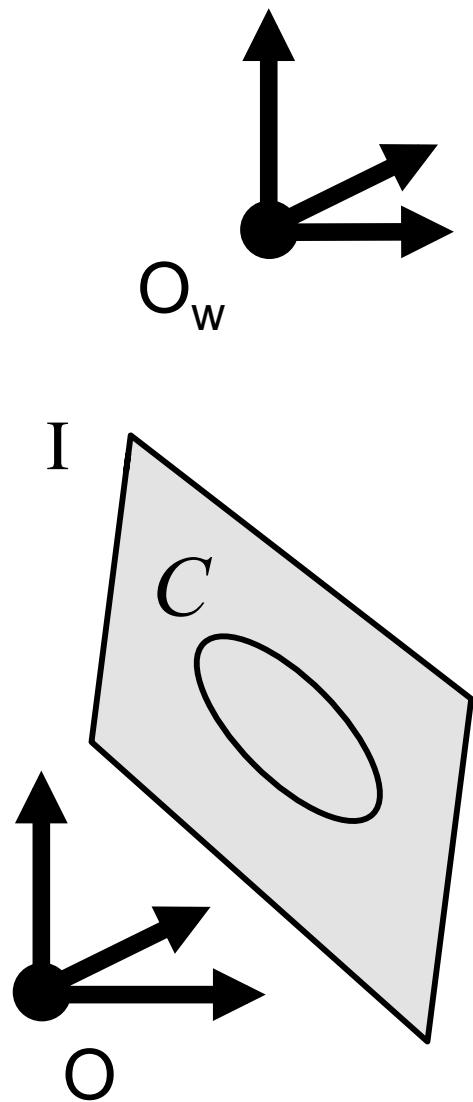
Direct approach

We use the following results:

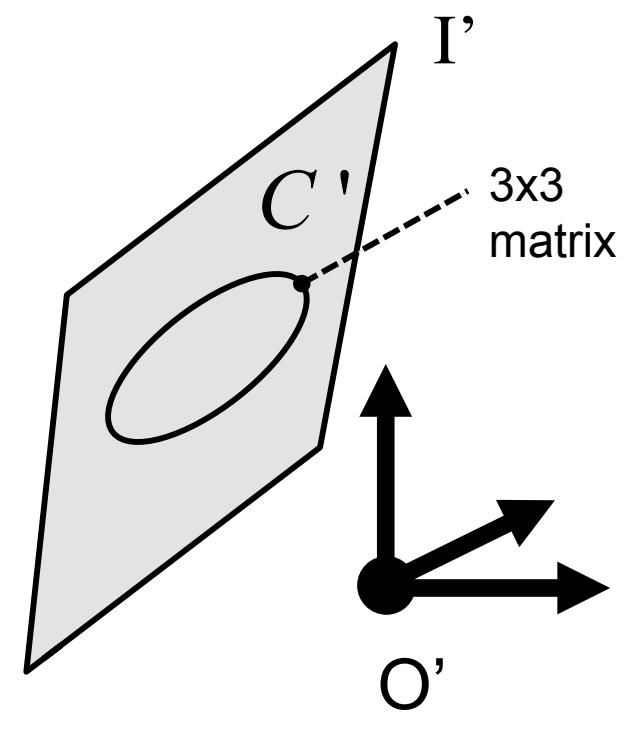
1. A relationship that maps conics across views
2. Concept of absolute conic and its relationship to K
3. The Kruppa equations

Projections of conics across views

$$X^T C_w X = 0 \quad [\text{Eq. 1}]$$



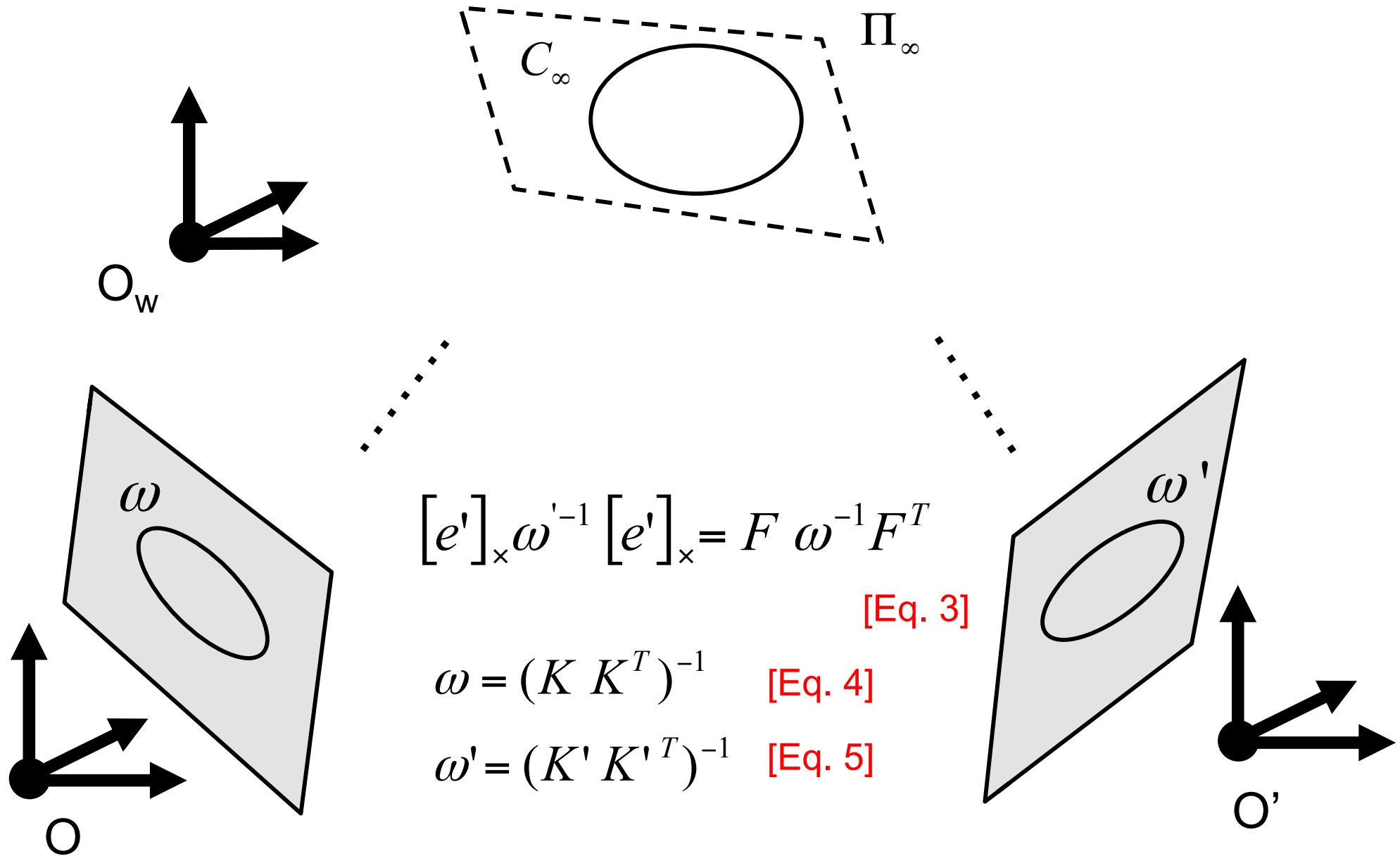
$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$



$$[e']_x C'^{-1} [e']_x = F C^{-1} F^T \quad [\text{Eq. 2}]$$

Projection of absolute conics across views

From lecture 4, [HZ] page 210, sec. 8.5.1



Kruppa equations

[Faugeras et al. 92]

From [HZ] page 471

$$\begin{pmatrix} u_2^T K' K'^T u_2 \\ -u_1^T K' K'^T u_2 \\ u_1^T K' K'^T u_1 \end{pmatrix} \times \begin{pmatrix} \sigma_1^2 v_1^T K K^T v_1 \\ \sigma_1 \sigma_2 v_1^T K K^T v_2 \\ \sigma_2^2 v_2^T K K^T v_2 \end{pmatrix} = 0 \quad [\text{Eq. 6}]$$

where u_i , v_i and σ_i are the columns and singular values of SVD of F

These give us two independent constraints in the elements of K and K'

Kruppa equations

[Faugeras et al. 92]

$$\begin{pmatrix} u_2^T K' K'^T u_2 \\ -u_1^T K' K'^T u_2 \\ u_1^T K' K'^T u_1 \end{pmatrix} \times \begin{pmatrix} \sigma_1^2 v_1^T K K^T v_1 \\ \sigma_1 \sigma_2 v_1^T K K^T v_2 \\ \sigma_2^2 v_2^T K K^T v_2 \end{pmatrix} = 0$$

$$\frac{u_2^T K K^T u_2}{\sigma_1^2 v_1^T K K^T v_1} = \frac{-u_1^T K K^T u_2}{\sigma_1 \sigma_2 v_1^T K K^T v_2} = \frac{u_1^T K K^T u_1}{\sigma_2^2 v_2^T K K^T v_2} \quad [\text{Eq. 7}]$$

- Let's make the following assumption: $K' = K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}$ [Eq. 8]

$$[\text{Eq. 9}] \quad \alpha f^2 + \beta f + \gamma = 0 \longrightarrow f$$

Kruppa equations

[Faugeras et al. 92]

- Powerful if we want to self-calibrate 2 cameras with unknown focal length
- Limitations:
 - Work on a camera pair
 - Don't work if $R=0$

[Eq. 10] $\begin{bmatrix} e' \end{bmatrix}_x \omega^{-1} \begin{bmatrix} e' \end{bmatrix}_x = F \omega^{-1} F^T$ becomes trivial
Since: $F = \begin{bmatrix} e' \end{bmatrix}_x$

Self-calibration

[HZ] Chapters 19 “Auto-calibration”

Several approaches:

- Use single-view metrology constraints (lecture 4)
- Direct approach (Kruppa Eqs) for 2 views
- Algebraic approach
- Stratified approach

Algebraic approach Multi-view approach

Suppose we have a projective reconstruction $\{\tilde{M}_i, \tilde{X}_j\}$

Let H be a homography such that:

$$\begin{cases} \text{First perspective camera is canonical: } \tilde{M}_1 = [\begin{array}{cc} I & 0 \end{array}] \text{ [Eq. 11]} \\ \text{i}^{\text{th}} \text{ perspective reconstruction of the camera (known): } \tilde{M}_i = [\begin{array}{cc} A_i & b_i \end{array}] \text{ [Eq. 12]} \end{cases}$$

$$[\text{Eq. 13}] \quad (A_i - b_i p^T) K_1 K_1^T (A_i - b_i p^T)^T = K_i K_i^T \quad i=2 \dots m$$

$$[\text{Eq. 14}] \quad H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix} \quad \begin{aligned} p &\text{ is an unknown } 3 \times 1 \text{ vector} \\ K_1 \dots K_m &\text{ are unknown} \end{aligned}$$

Algebraic approach

Multi-view approach

Suppose we have a projective reconstruction

Let H be a homography such that:

$$\left\{ \begin{array}{l} \text{First perspective camera is canonical: } \tilde{M}_1 = [\begin{array}{cc} I & 0 \end{array}] \text{ [Eq. 11]} \\ \text{i}^{\text{th}} \text{ perspective reconstruction of the camera (known): } \tilde{M}_i = [\begin{array}{cc} A_i & b_i \end{array}] \text{ [Eq. 12]} \end{array} \right.$$

$$[\text{Eq. 13}] \quad (A_i - b_i p^T) K_1 K_1^T (A_i - b_i p^T)^T = K_i K_i^T \quad i=2 \dots m$$

- How many unknowns?
- 3 from p
 - 5 m from $K_1 \dots K_m$

How many equations? 5 independent equations [per view]

Algebraic approach

Multi-view approach

Suppose we have a projective reconstruction

Let H be a homography such that:

$$\left\{ \begin{array}{l} \text{First perspective camera is canonical: } \tilde{M}_1 = [\begin{array}{cc} I & 0 \end{array}] \text{ [Eq. 11]} \\ \text{\textit{i}th perspective reconstruction of the camera (known): } \tilde{M}_i = [\begin{array}{cc} A_i & b_i \end{array}] \\ \text{[Eq. 12]} \end{array} \right.$$

Assume all camera matrices are identical: $K_1 = K_2 \dots = K_m$

$$[\text{Eq. 15}] \quad \left(A_i - b_i p^T \right) K \ K^T \left(A_i - b_i p^T \right)^T = K \ K^T \quad i=2\dots m$$

- How many unknowns?
- 3 from p
 - 5 from K

How many equations? 5 independent equations [per view]

We need at least 3 views to solve the self-calibration problem

Algebraic approach

Art of self-calibration:

Use assumptions on Ks to generate enough equations on the unknowns

<i>Condition</i>	<i>N. Views</i>
• Constant internal parameters	3
• Aspect ratio and skew known • Focal length and offset vary	4
• Skew =0, all other parameters vary	8

Issue: the larger is the number of view,
the harder is the correspondence problem

Bundle adjustment helps!

SFM problem - summary

1. Estimate structure and motion up perspective transformation
 1. Algebraic
 2. factorization method
 3. bundle adjustment
2. Convert from perspective to metric (self-calibration)
3. Bundle adjustment

** or **

1. Bundle adjustment with self-calibration constraints