

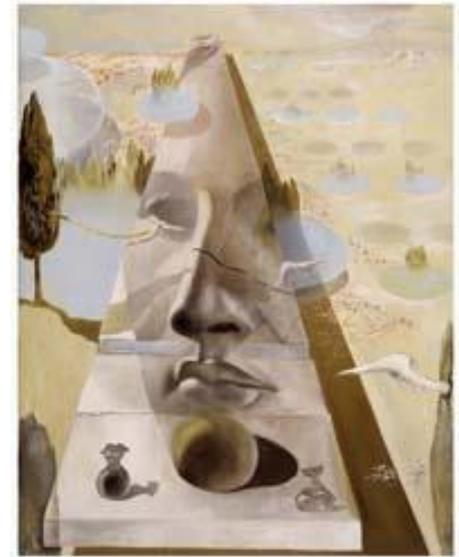
# Lecture 7

## Multi-view geometry

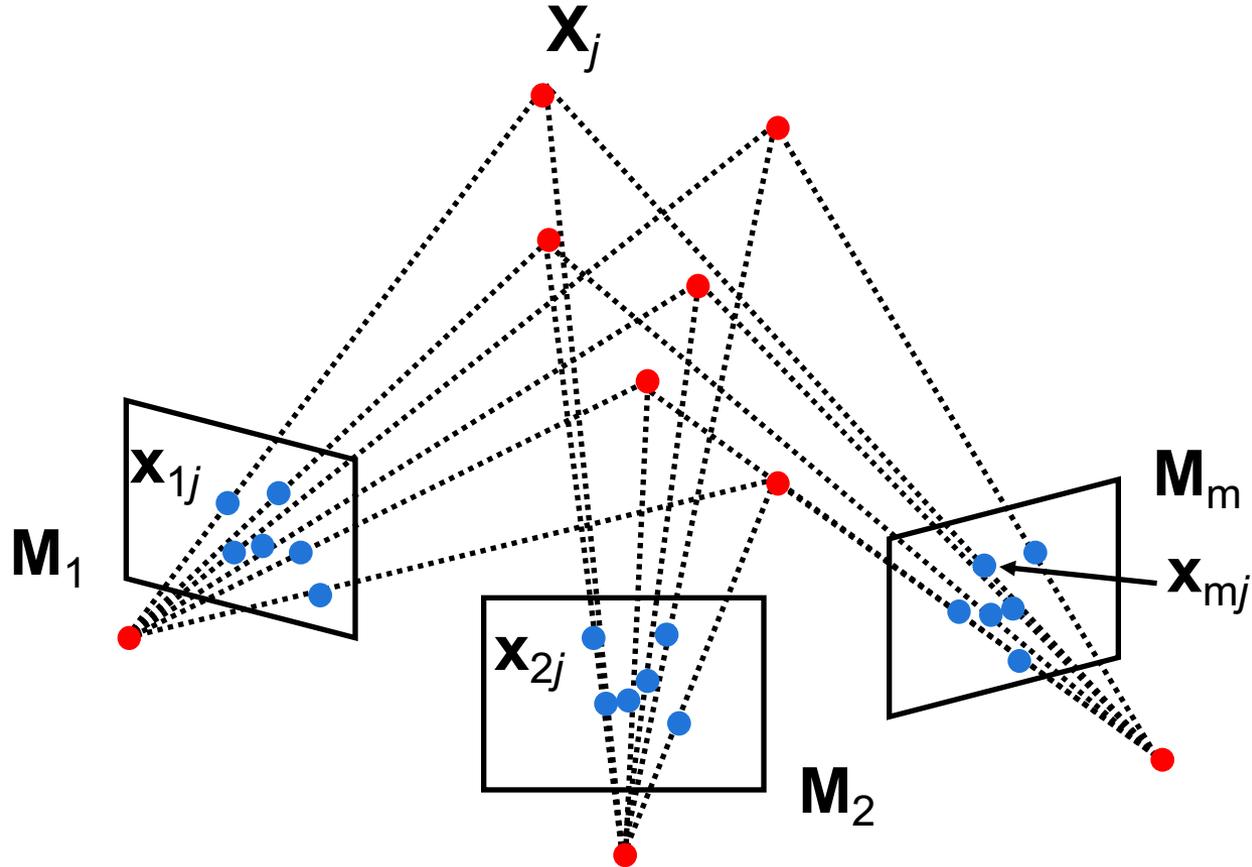
- The SFM problem
- Affine SFM
- Perspective SFM
- Self-calibration
- Applications

### 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”



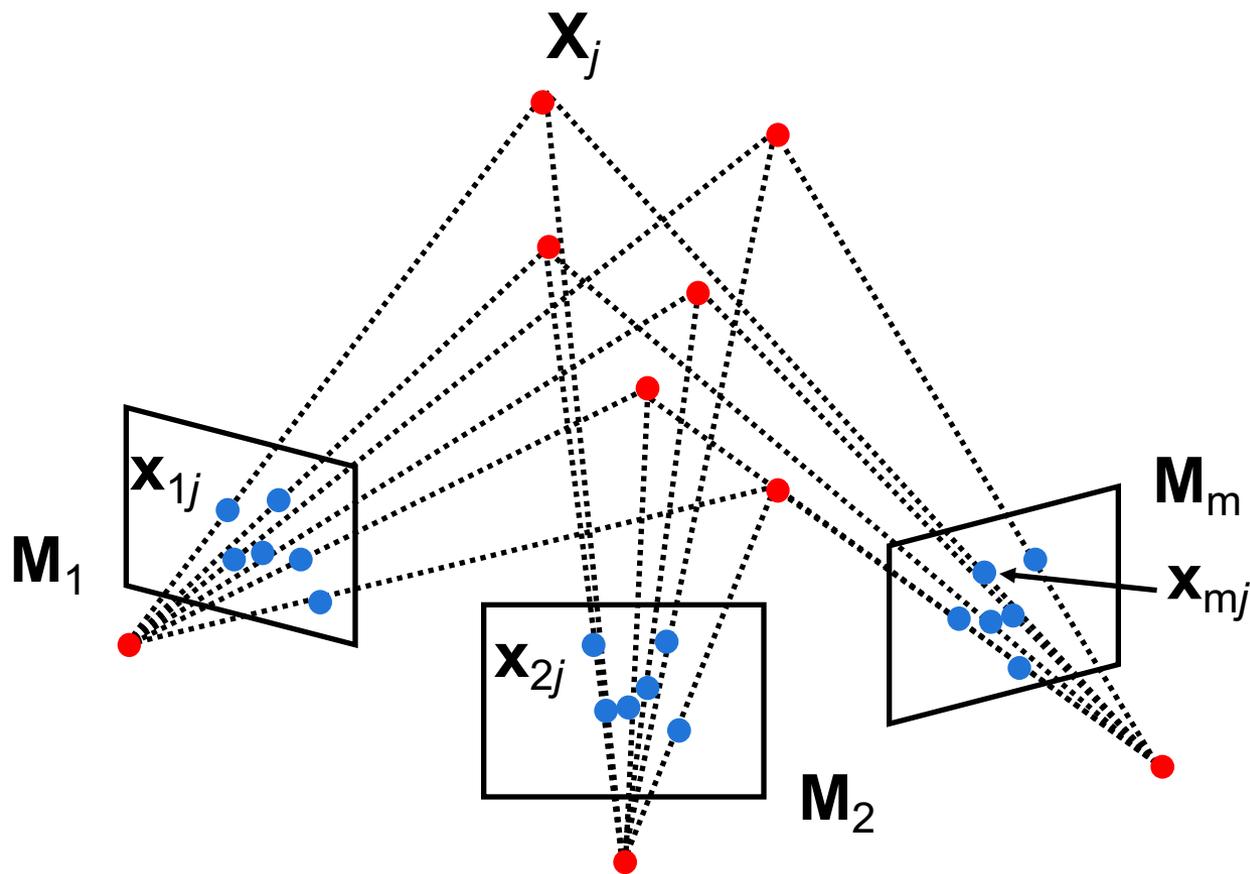
# Structure from motion problem



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

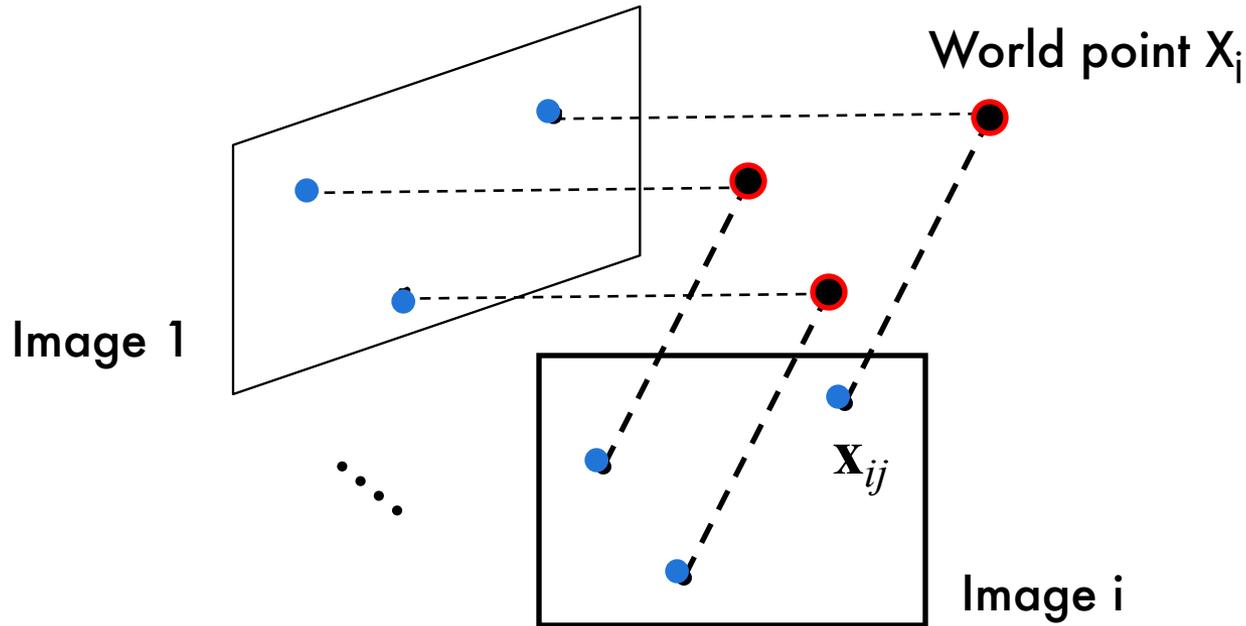


From the  $m \times n$  observations  $x_{ij}$ , estimate:

- $m$  projection matrices  $M_i$
- $n$  3D points  $X_j$

motion  
structure

# Affine structure from motion (simpler problem)



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

- $m$  projection matrices  $M_i$  (affine cameras)
- $n$  3D points  $X_j$

## Perspective

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

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

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

---

## Affine

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

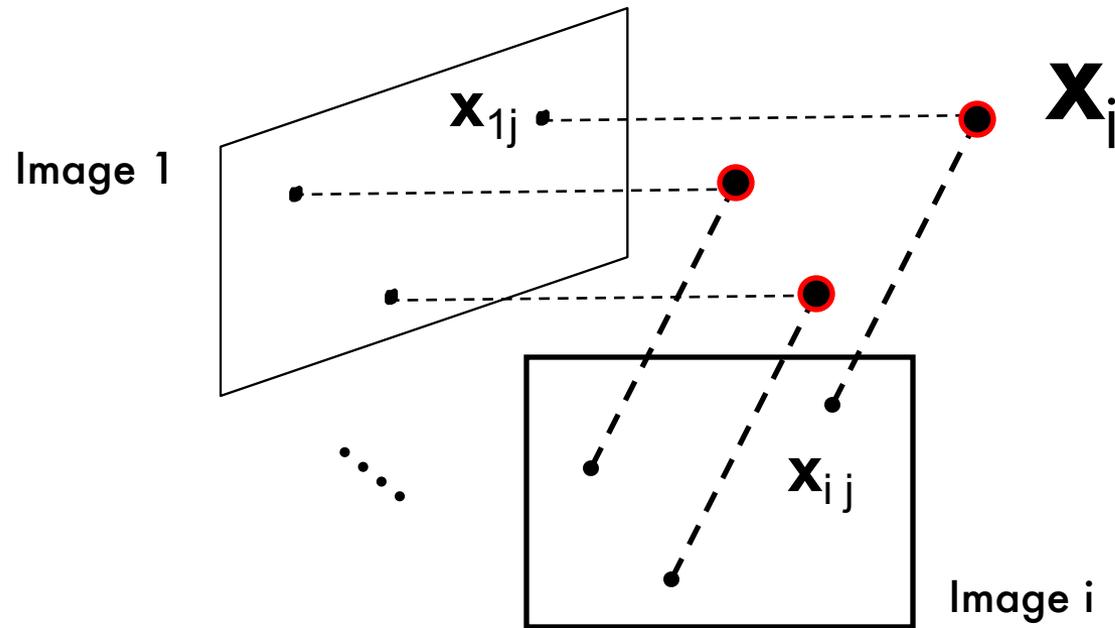
$$M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{2 \times 3} & \mathbf{b}_{2 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{m}_1 & & \\ & \mathbf{m}_2 & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}^E = \left( \underset{\substack{\uparrow \\ \text{magnification}}}{\mathbf{m}_1} \mathbf{X}, \underset{\substack{\uparrow \\ \text{magnification}}}{\mathbf{m}_2} \mathbf{X} \right)^T = \begin{bmatrix} \mathbf{A}_{2 \times 3} & \mathbf{b}_{2 \times 1} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{A} \mathbf{X}^E + \mathbf{b}$$

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

[Eq. 3]

# Affine cameras



For the affine case (in Euclidean space)

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

$2 \times 1$        $2 \times 3$        $3 \times 1$        $2 \times 1$

# 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  $8m + 3n - 8$  unknowns

# The Affine Structure-from-Motion Problem

Two approaches:

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

- Factorization method

# 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

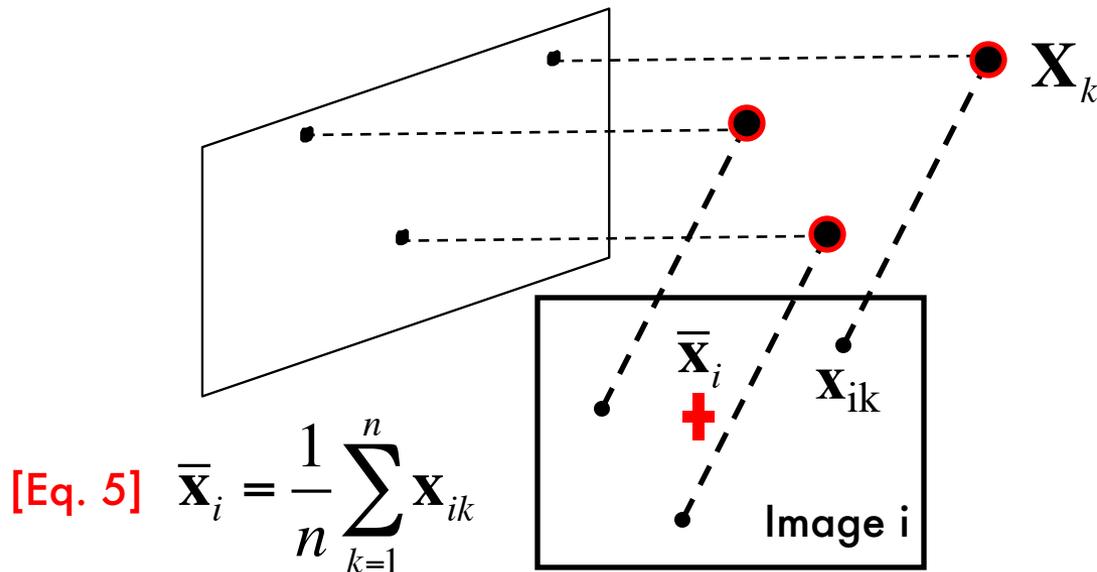
C. Tomasi and T. Kanade [Shape and motion from image streams under orthography: A factorization method](#). *IJCV*, 9(2):137-154, November 1992.

- Data centering
- Factorization

# A factorization method - Centering the data

Centering: subtract the centroid of the image points

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



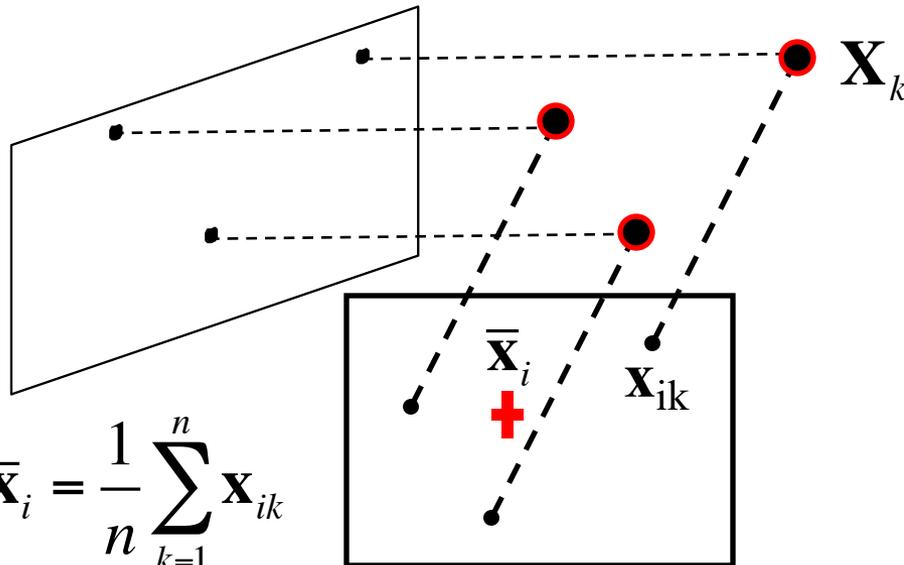
# A factorization method - Centering the data

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]**



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

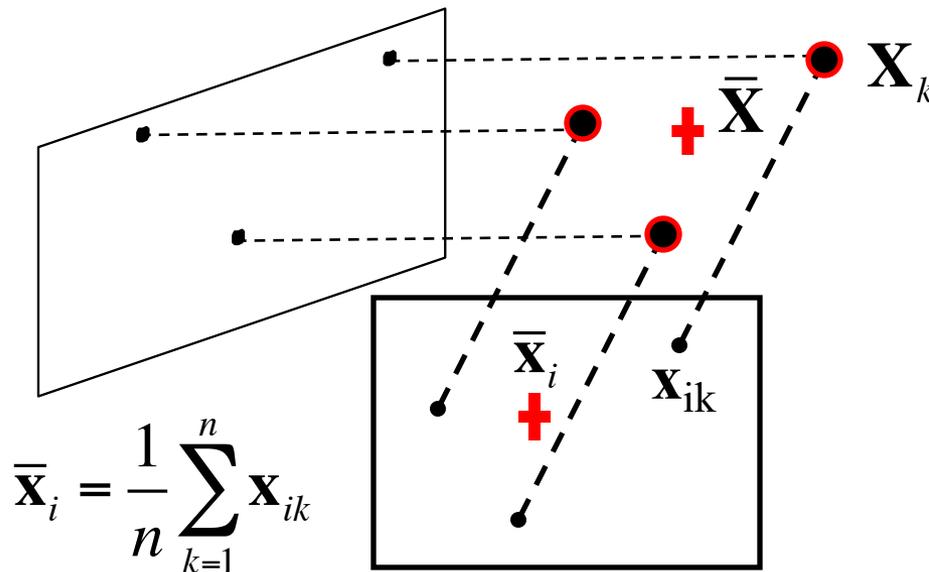
# A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$\begin{aligned}
 \text{[Eq. 6]} \quad \hat{\mathbf{x}}_{ij} &= \mathbf{x}_{ij} - \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i - \frac{1}{n} \sum_{k=1}^n \mathbf{A}_i \mathbf{X}_k - \frac{1}{n} \sum_{k=1}^n \mathbf{b}_i \\
 &= \mathbf{A}_i \left( \mathbf{X}_j - \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \right) = \mathbf{A}_i (\mathbf{X}_j - \bar{\mathbf{X}}) \\
 &= \mathbf{A}_i \hat{\mathbf{X}}_j \quad \text{[Eq. 8]}
 \end{aligned}$$

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

[Eq. 4]



$$\bar{\mathbf{X}}_i = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik}$$

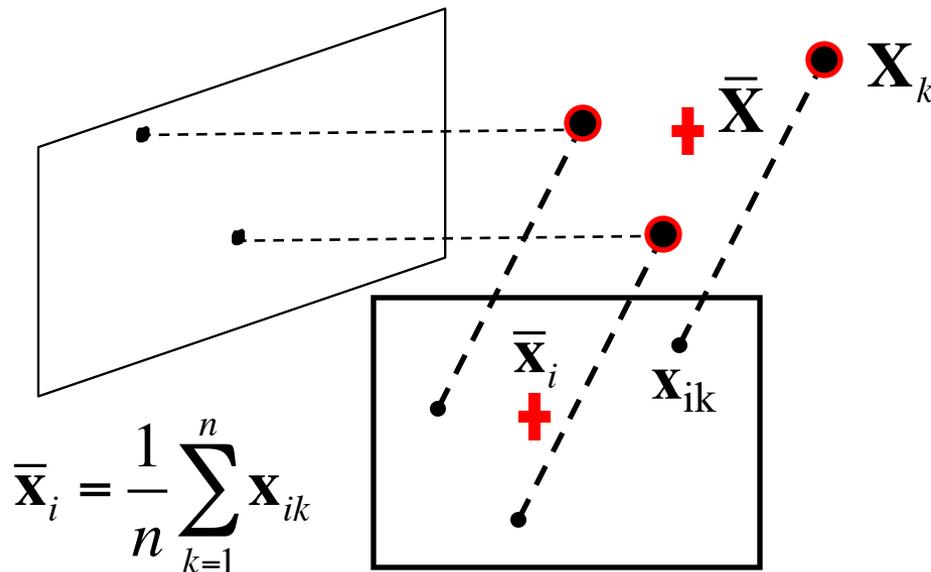
$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \quad \text{[Eq. 7]}$$

Centroid of 3D points

# A factorization method - Centering the data

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

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



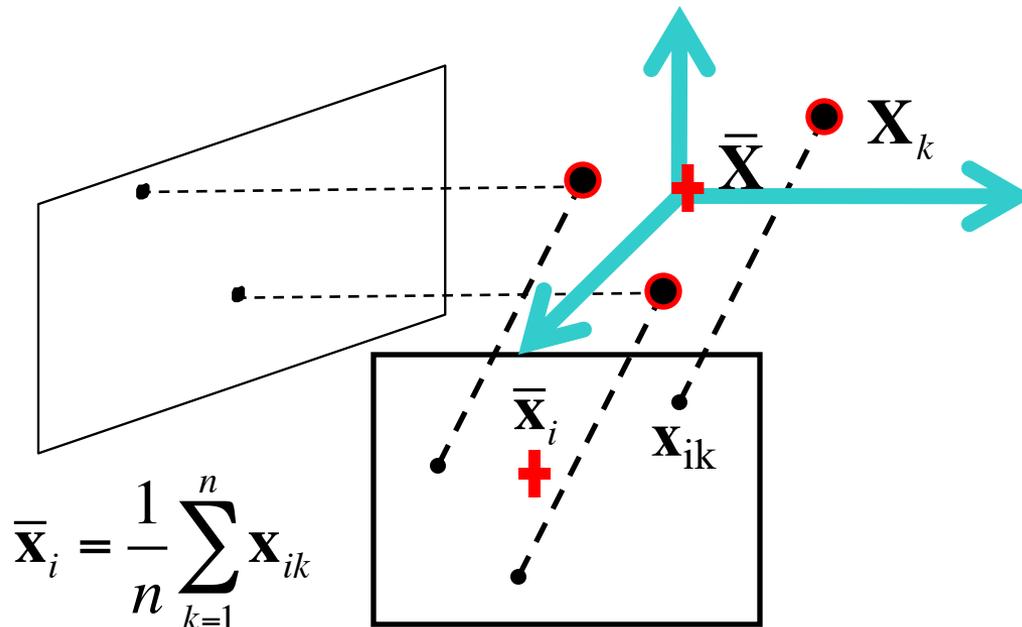
$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \quad [\text{Eq. 7}]$$

Centroid of 3D points

# A factorization method - Centering the data

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

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



$$\bar{\mathbf{x}}_i = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_{ik}$$

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

Centroid of 3D points

# A factorization method - factorization

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

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

  
**points ( n )**

  
**cameras ( 2 m )**

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

# A factorization method - factorization

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

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} \quad [\text{Eq. 10}]$$

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

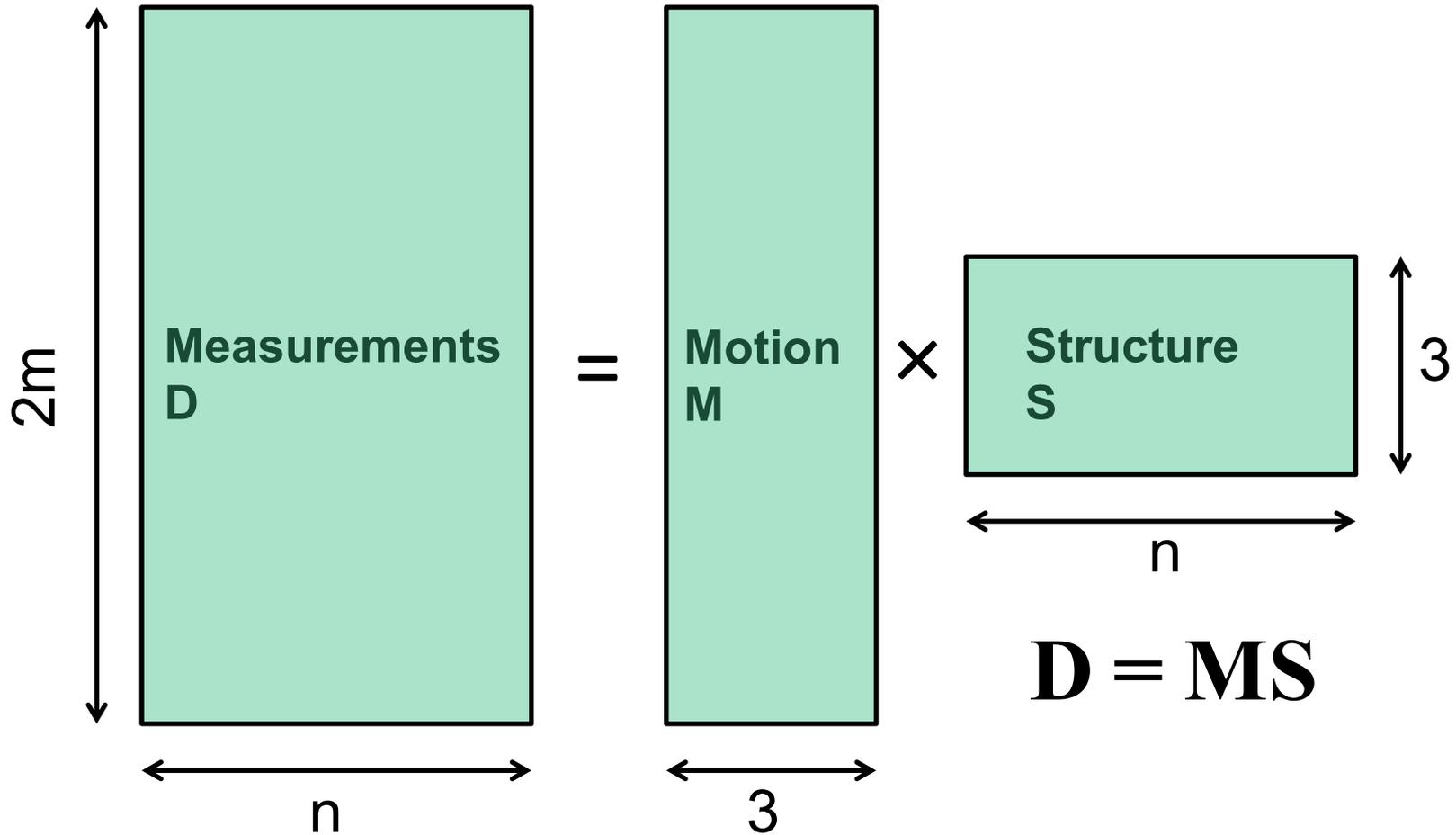
Each  $\hat{\mathbf{x}}_{ij}$  entry is a  $2 \times 1$  vector!

$\mathbf{A}_i$  is  $2 \times 3$  and  $\mathbf{X}_i$  is  $3 \times 1$

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)

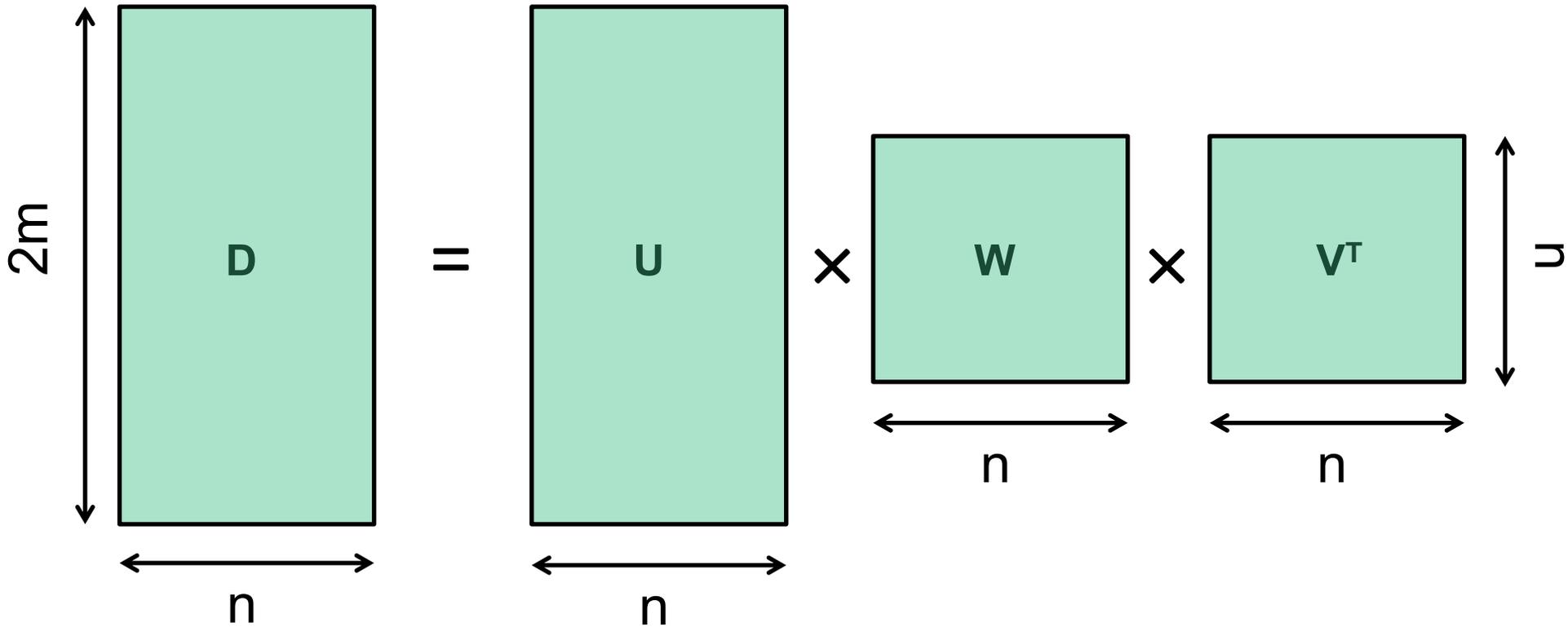
# Factorizing the Measurement Matrix

How to factorize D?



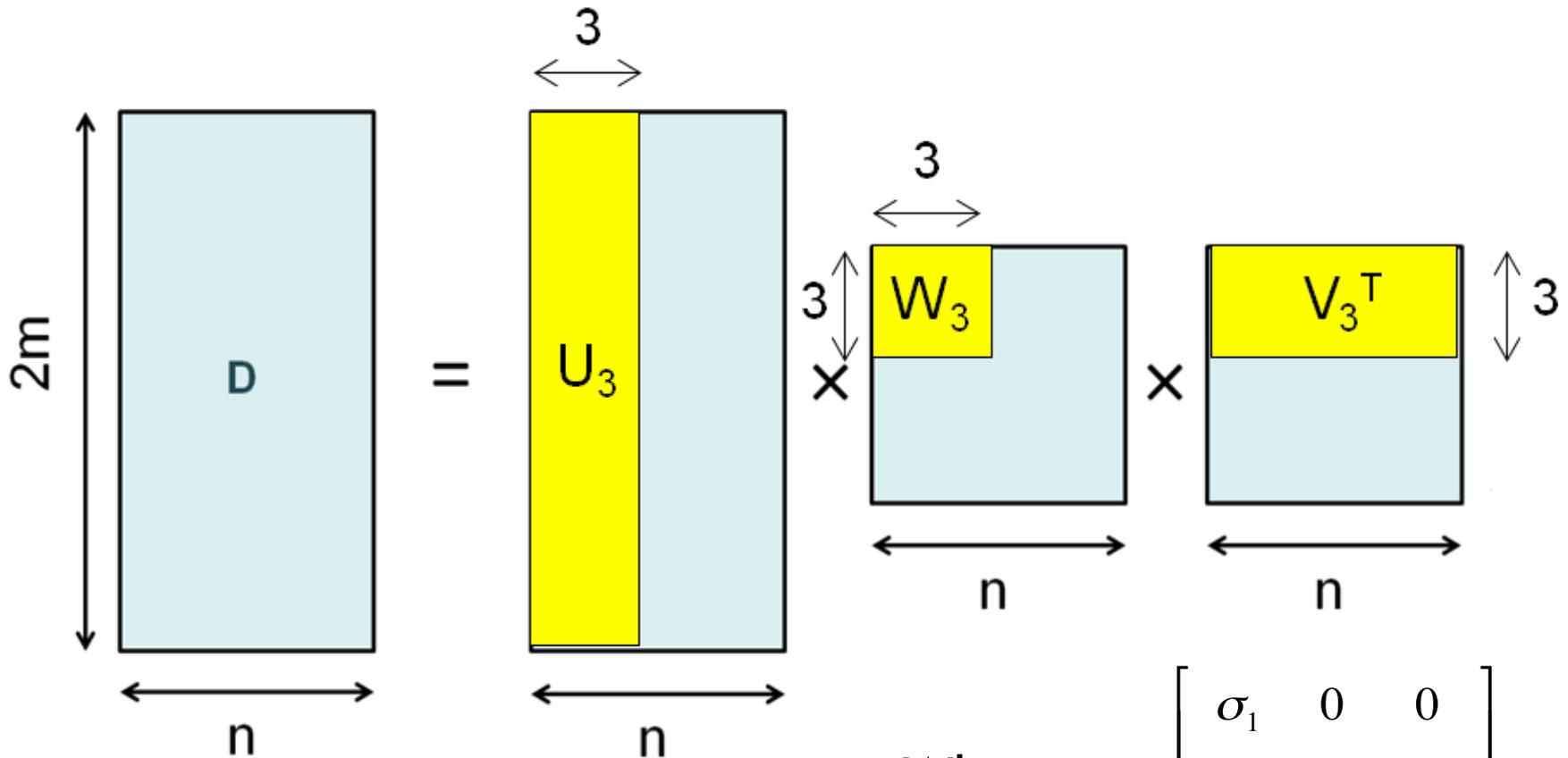
# Factorizing the Measurement Matrix

- By computing the Singular value decomposition of  $D$ !



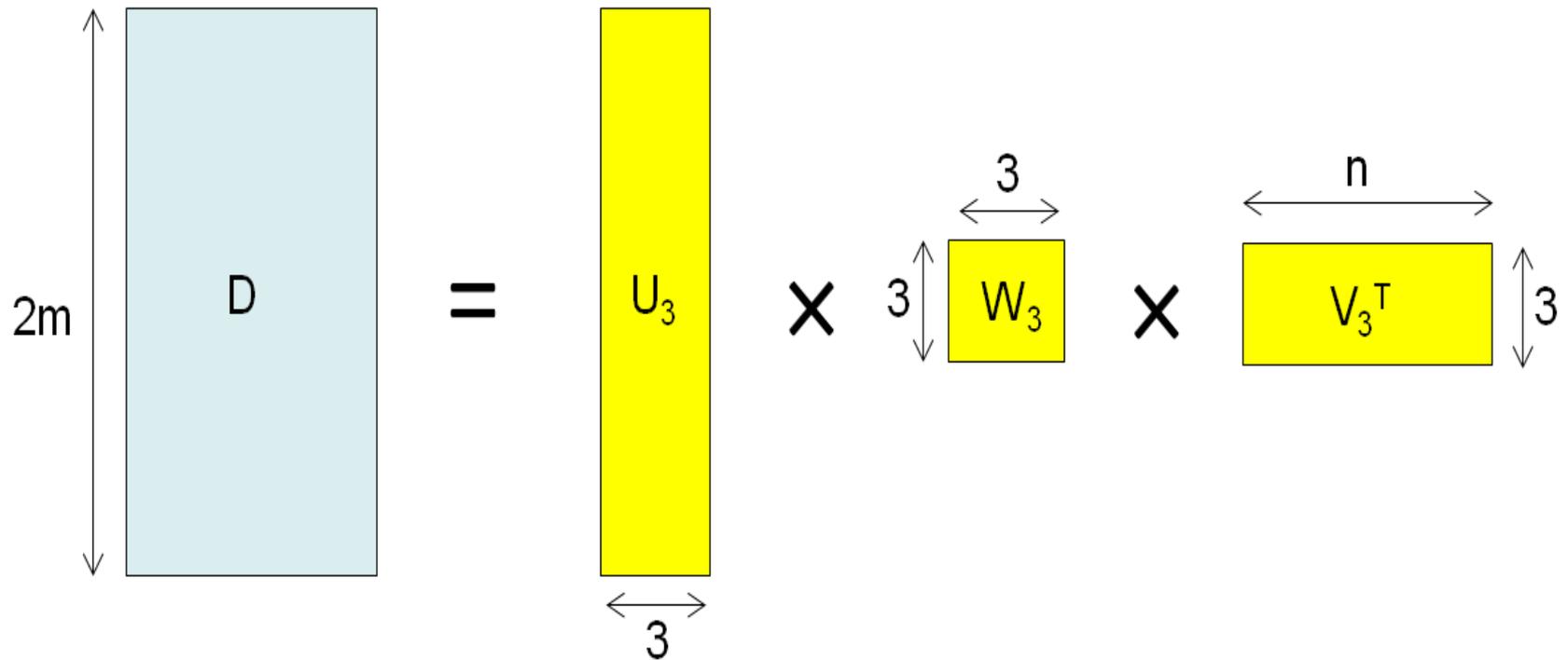
# Factorizing the Measurement Matrix

Since  $\text{rank}(D)=3$ , there are only 3 non-zero singular values  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$

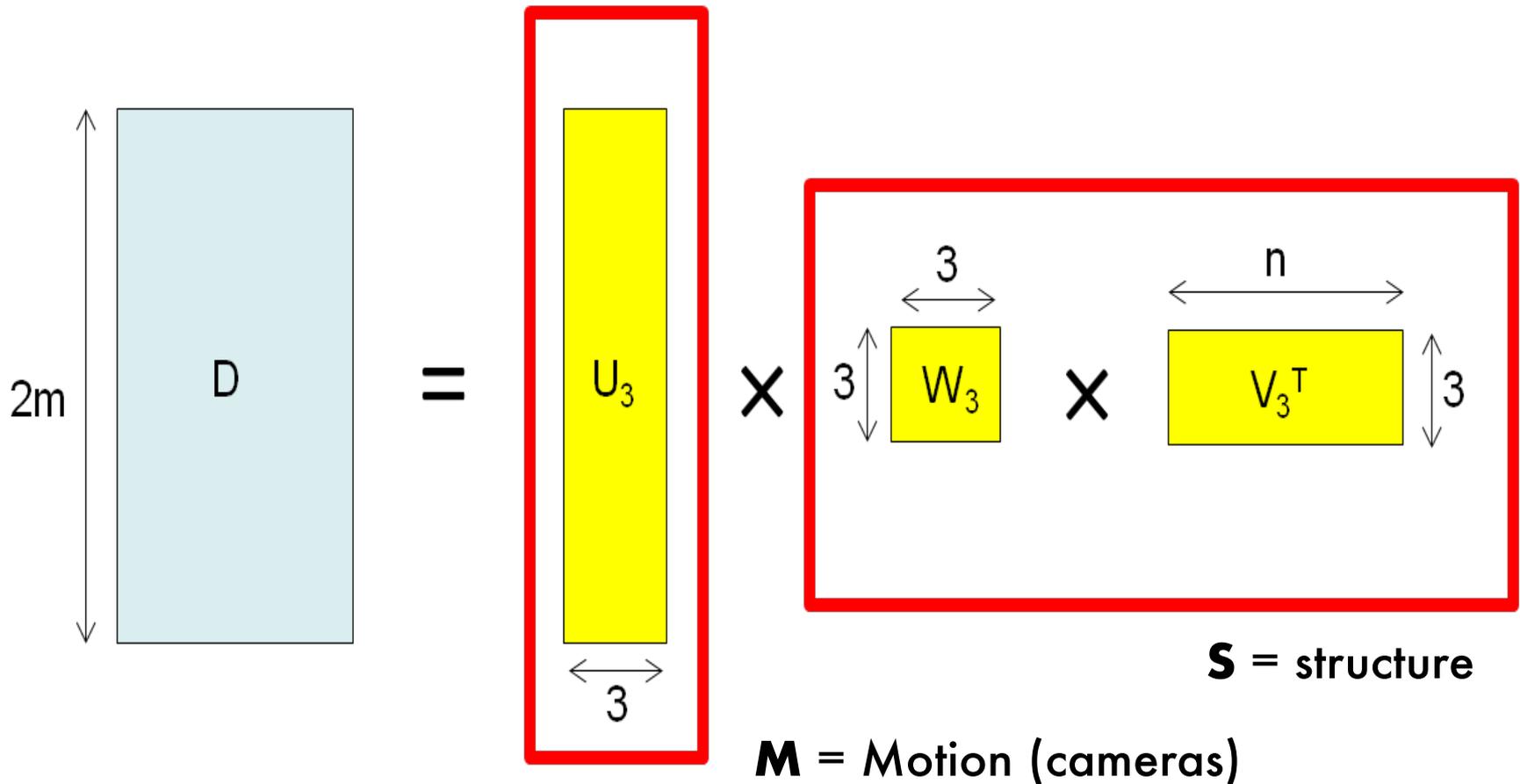


Where  $W_3 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$  [Eq. 11]

# 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

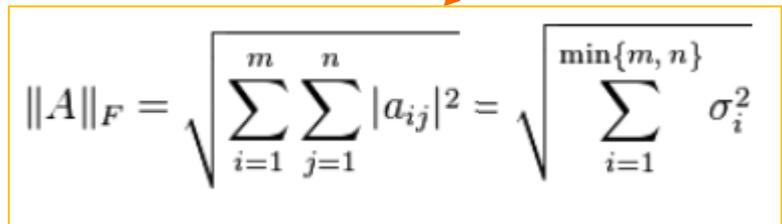
$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T = \mathbf{U}_3 (\mathbf{W}_3 \mathbf{V}_3^T) = \mathbf{M} \mathbf{S} \quad [\text{Eq. 12}]$$

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.

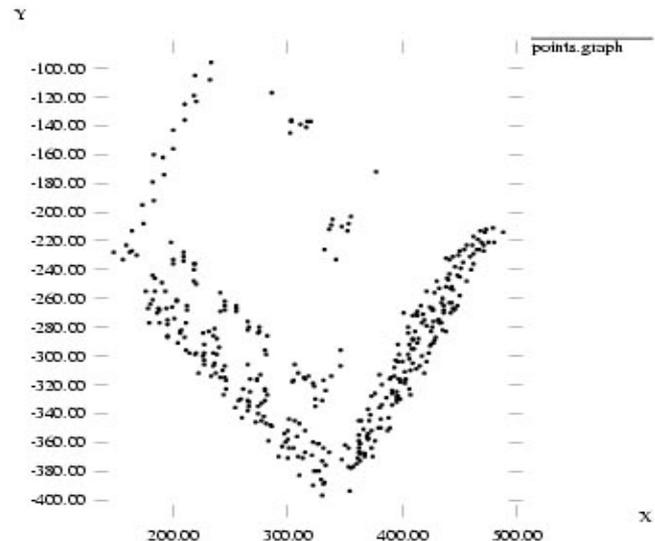
$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T \quad \left\{ \begin{array}{l} \mathbf{M} \approx \mathbf{U}_3 \\ \mathbf{S} \approx \mathbf{W}_3 \mathbf{V}_3^T \end{array} \right.$$


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

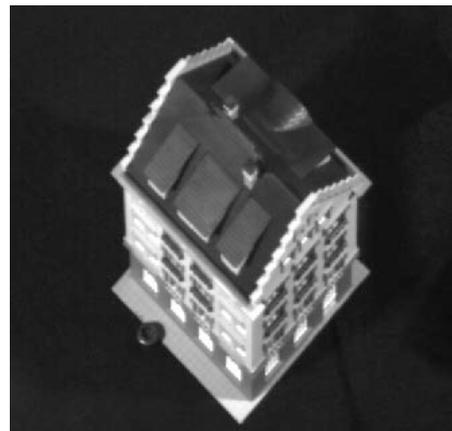
# Reconstruction results



1

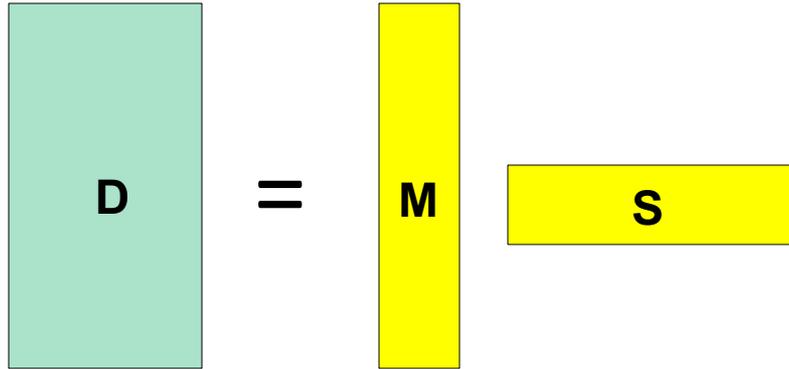


120



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



# Affine Ambiguity

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

$\underbrace{\hspace{10em}}_{\mathbf{M}^*} \quad \underbrace{\hspace{10em}}_{\mathbf{S}^*}$

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

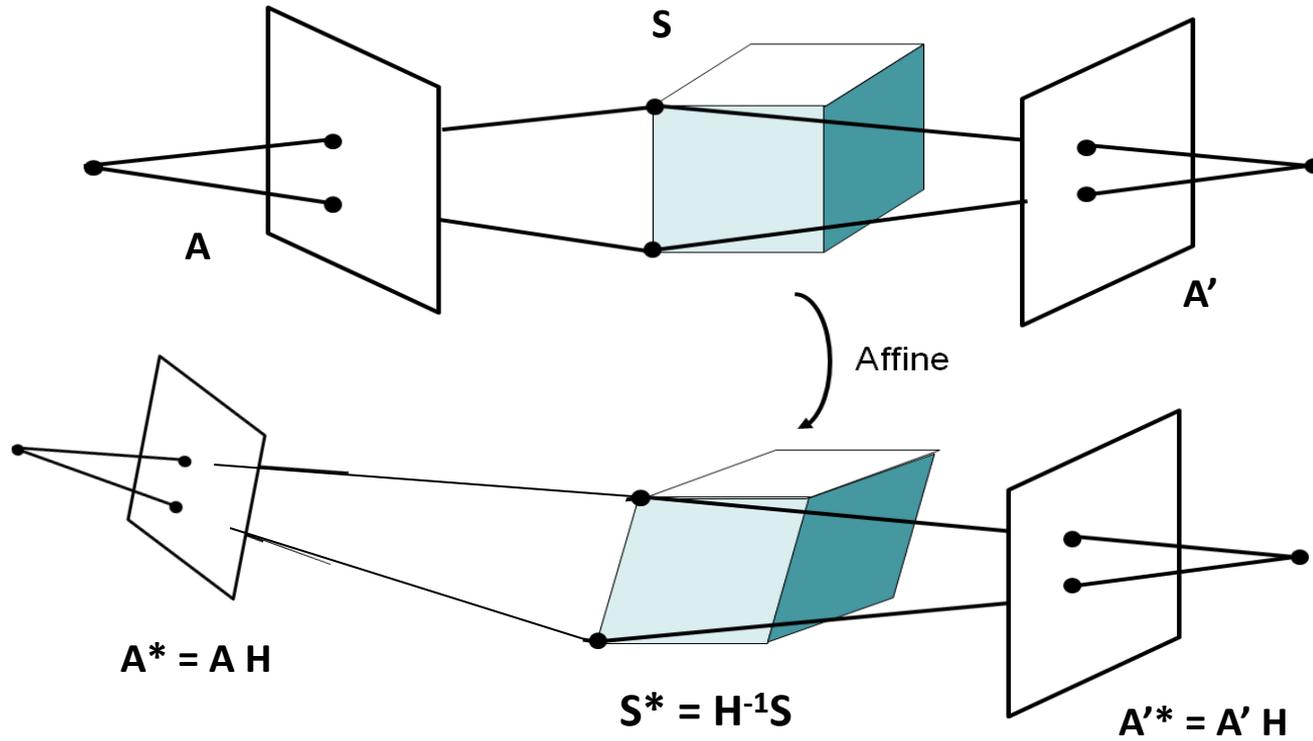
$$\mathbf{M}^* = \mathbf{M} \mathbf{H}$$

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

where  $\mathbf{H}$  is an arbitrary 3x3 matrix describing an affine transformation

- Additional constraints must be enforced to resolve this ambiguity

# Affine Ambiguity



# 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

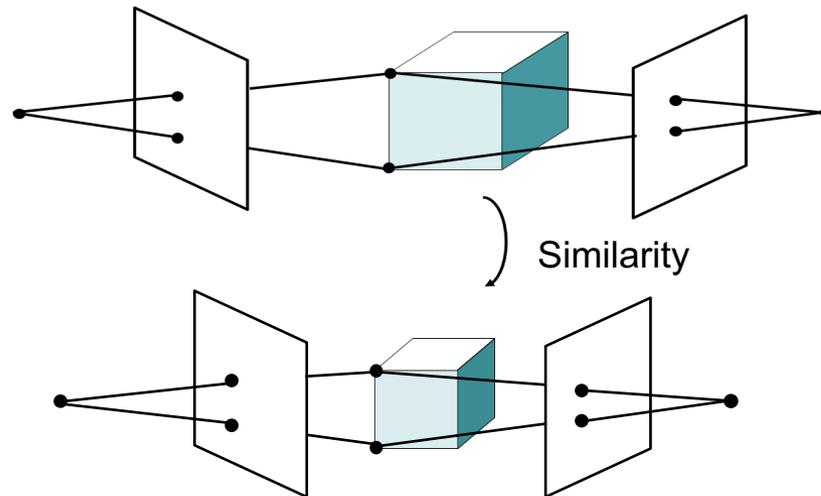
Problem: estimate  $m$  matrices  $\mathbf{A}_i$ ,  $m$  matrices  $\mathbf{b}_i$   
and the  $n$  positions  $\mathbf{X}_j$  from the  $m \times n$  observations  $\mathbf{x}_{ij}$  .

How many equations and how many unknowns?

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

# 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

# Similarity Ambiguity

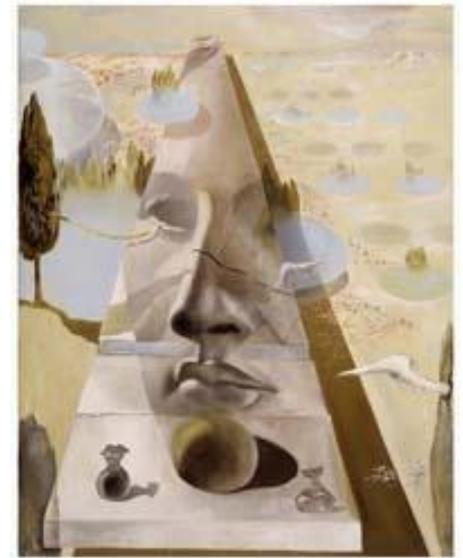
- It is impossible, based on the images alone, to estimate the absolute scale of the scene



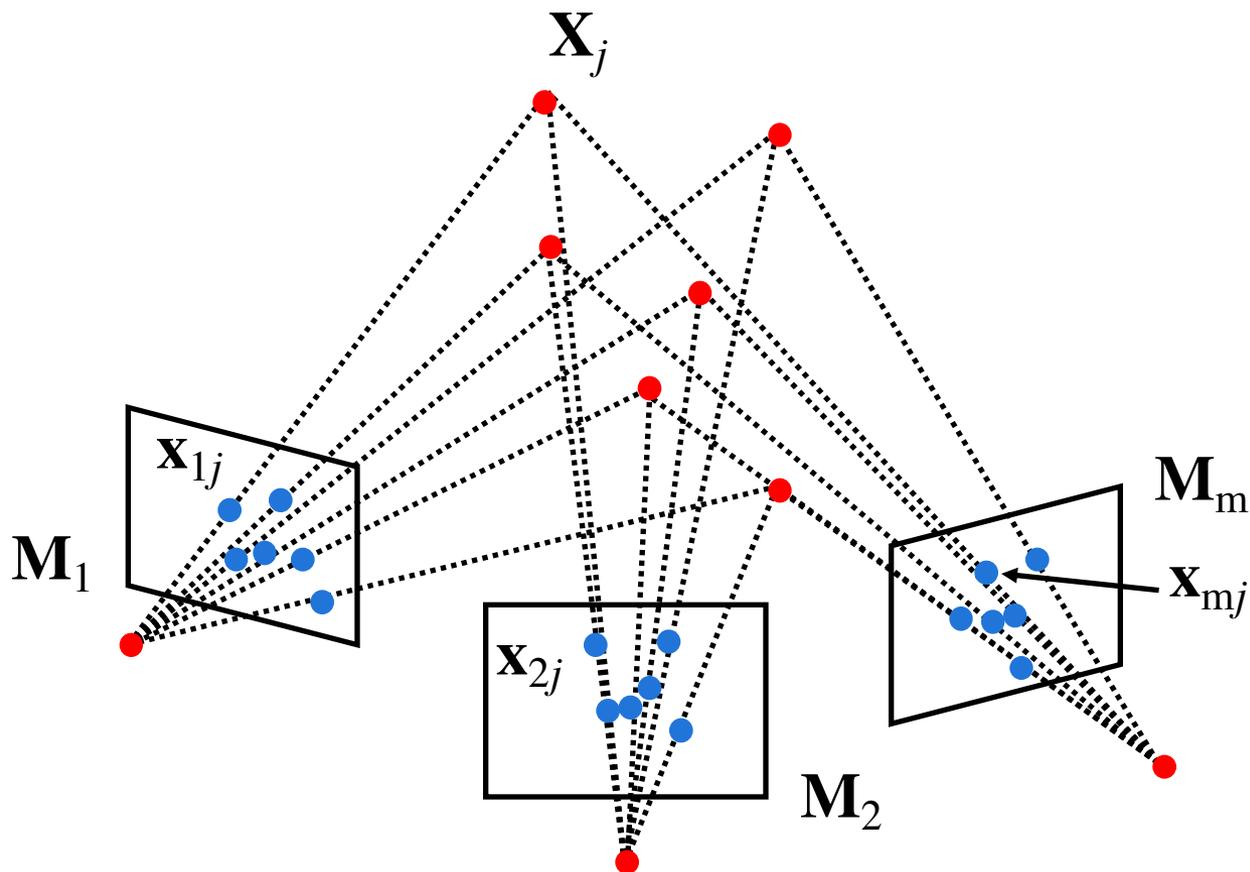
# Lecture 7

## Multi-view geometry

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



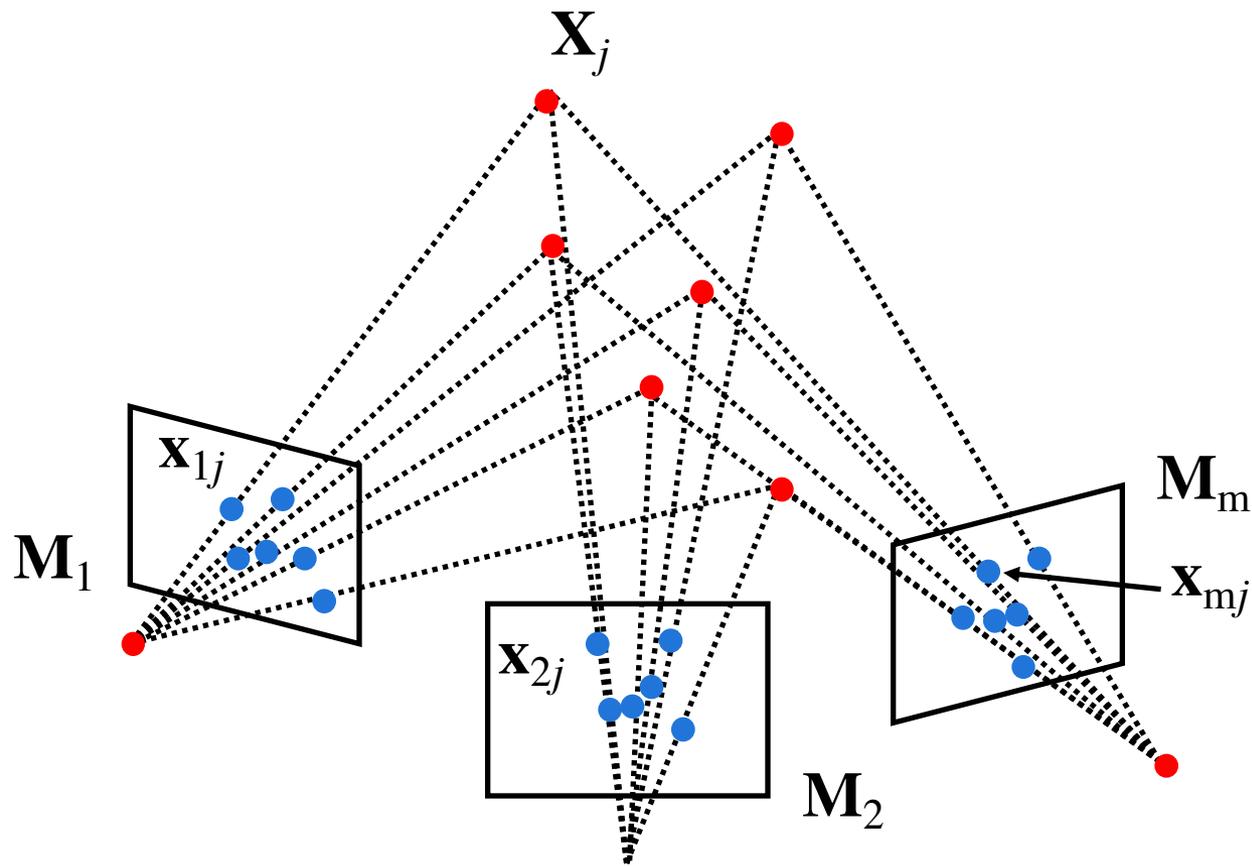
# Structure from motion problem



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

- $m$  projection matrices  $M_i = \text{motion}$
- $n$  3D points  $X_j = \text{structure}$

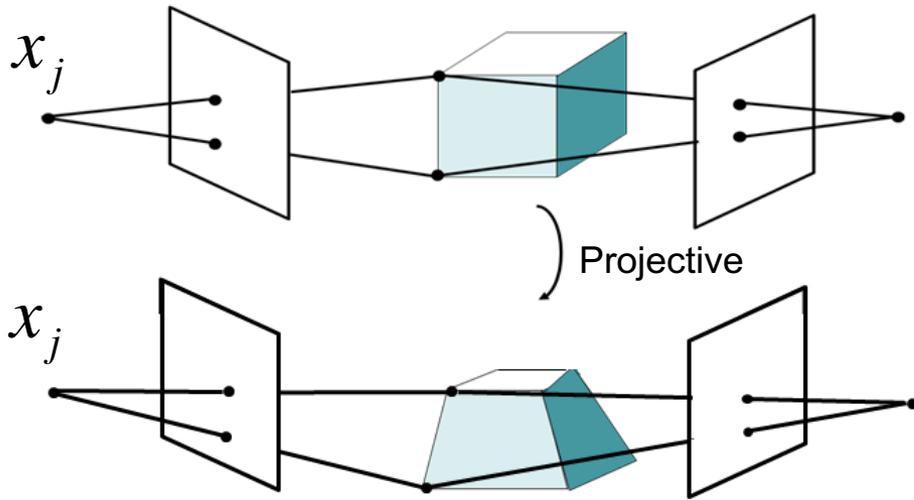
# Structure from motion problem



$m$  cameras  $M_1 \dots M_m$

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

# Structure from Motion Ambiguities



- In the general case (nothing is known) the ambiguity is expressed by an arbitrary 4X4 projective transformation

$$x_j = M_i X_j$$

$$M_i = K_i [R_i \quad T_i]$$

$$H X_j$$

$$M_j H^{-1}$$

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

# The Structure-from-Motion Problem

Given  $m$  images of  $n$  fixed points  $X_j$  we can write

$$x_{ij} = M_i X_j \quad \text{for } i = 1, \dots, m \quad \text{and } j = 1, \dots, n$$

N. of cameras                  N. of points

**Problem:** estimate  $m$   $3 \times 4$  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  $4 \times 4$  projective (where the  $4 \times 4$  projective is defined up to scale)

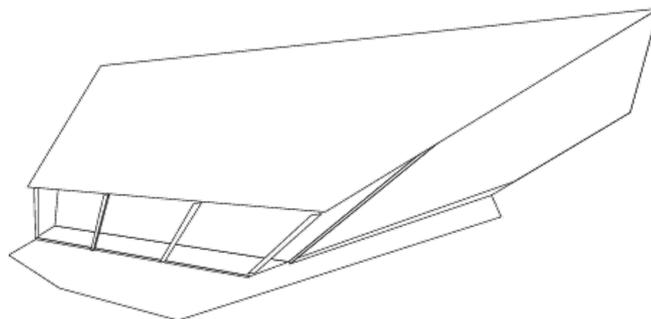
- How many equations and how many unknowns?

$2m \times n$  equations in  $11m + 3n - 15$  unknowns

# Projective Ambiguity

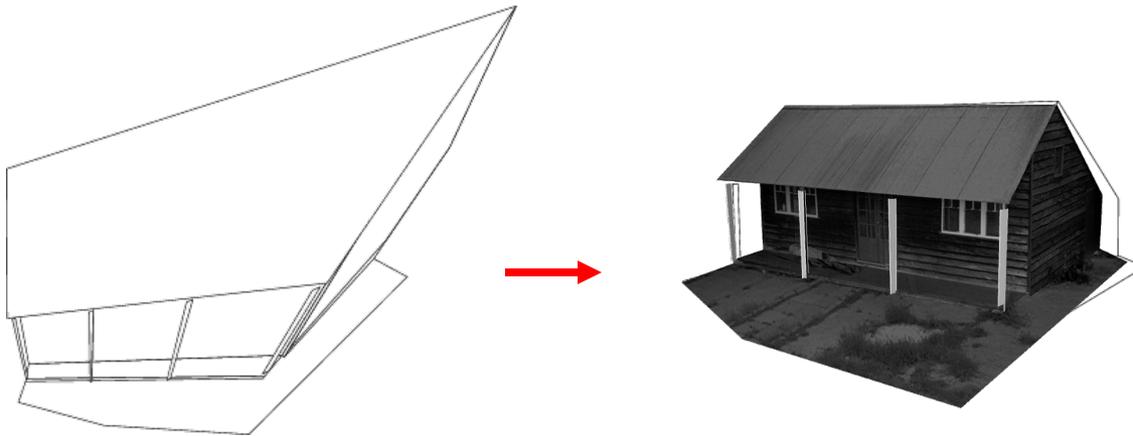


S =



# 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

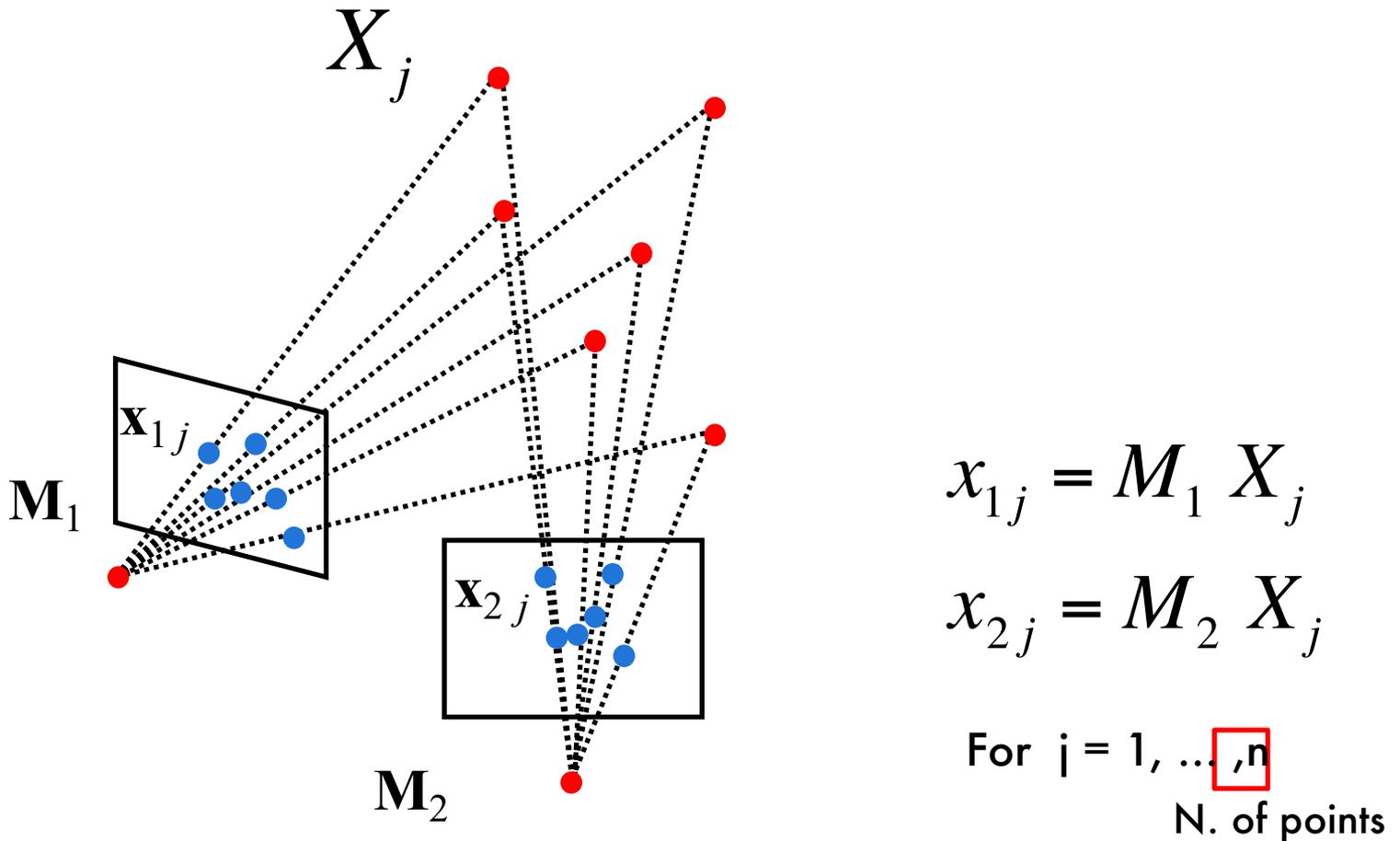
- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

## 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)

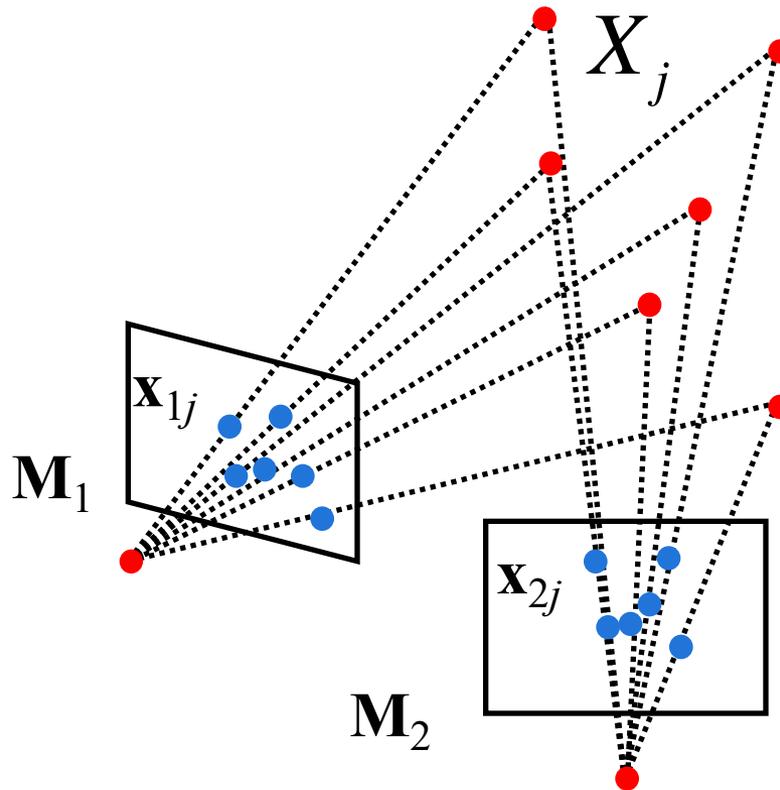


From at least 8 point correspondences, compute  $F$  associated to camera 1 and 2

# 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:

$$M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix}$$

[Eq. 3] Canonical perspective camera

$$M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix}$$

[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

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

$$\mathbf{x}' = [\mathbf{A} | \mathbf{b}] \tilde{\mathbf{X}} = [\mathbf{A} | \mathbf{b}] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} = \mathbf{A} [\mathbf{I} | \mathbf{0}] \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ 1 \end{bmatrix} + \mathbf{b} = \mathbf{A} [\mathbf{I} | \mathbf{0}] \tilde{\mathbf{X}} + \mathbf{b} = \mathbf{A} \mathbf{x} + \mathbf{b} \quad [\text{Eq. 7}]$$

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

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

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

# 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] 
$$\begin{cases} \tilde{M}_1 = M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \\ \tilde{M}_2 = M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix} \\ \tilde{\mathbf{X}} = H \mathbf{X} \end{cases} \quad \begin{aligned} \mathbf{x} &= M_1 H^{-1} H \mathbf{X} = [\mathbf{I} \mid \mathbf{0}] \tilde{\mathbf{X}} \\ \mathbf{x}' &= M_2 H^{-1} H \mathbf{X} = [\mathbf{A} \mid \mathbf{b}] \tilde{\mathbf{X}} \end{aligned} \quad \text{[Eq. 6]}$$

⋮

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

$\mathbf{x}'^T [\mathbf{b}_\times] \mathbf{A} \mathbf{x} = 0$  is this familiar?

$\mathbf{F} = [\mathbf{b}_\times] \mathbf{A}$

fundamental matrix!

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

# Compute cameras

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

## Compute $\mathbf{b}$ :

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

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

- 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  $\mathbf{b}^T \mathbf{F} = 0$  [Eq. 12-bis]

# Compute cameras

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

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

# Interpretation of $\mathbf{b}$

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

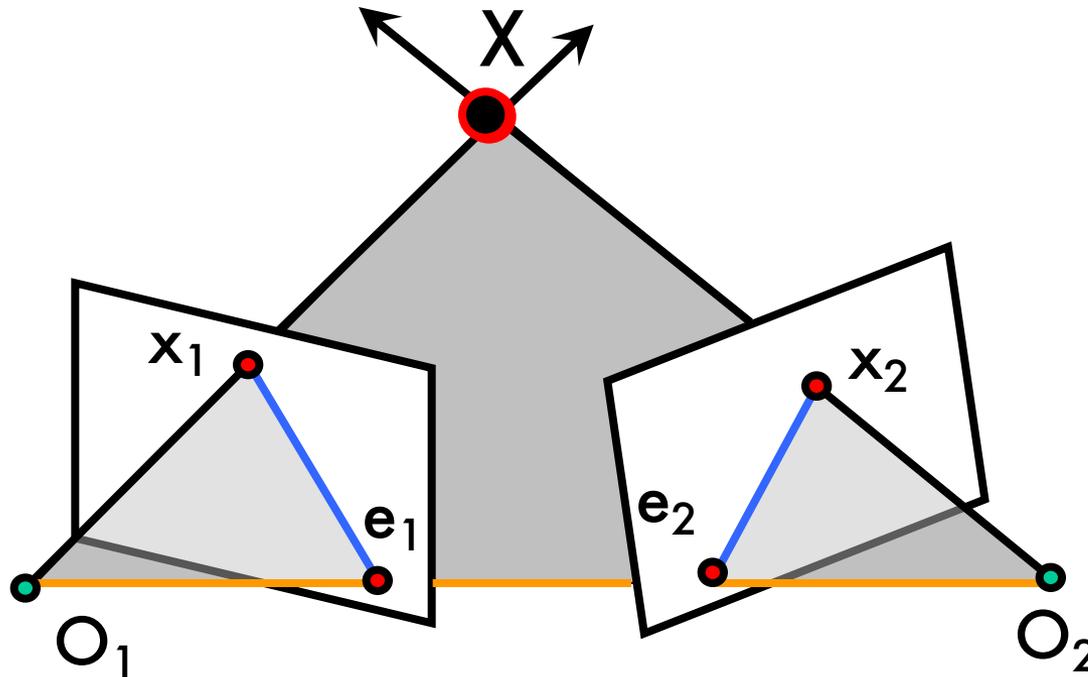
$$\mathbf{F} = [\mathbf{b}_\times] \mathbf{A}$$

[Eq. 11]

$$\left\{ \begin{array}{l} \mathbf{F} \mathbf{b} = 0 \quad [\text{Eq. 12}] \\ \mathbf{b}^T \mathbf{F} = 0 \quad [\text{Eq. 12-bis}] \end{array} \right.$$

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

# Epipolar Constraint [lecture 5]



$F x_2$  is the epipolar line associated with  $x_2$  ( $l_1 = F x_2$ )

$F^T x_1$  is the epipolar line associated with  $x_1$  ( $l_2 = F^T x_1$ )

$F$  is singular (rank two)

$$F e_2 = 0 \quad \text{and} \quad F^T e_1 = 0$$

$F$  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!

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



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

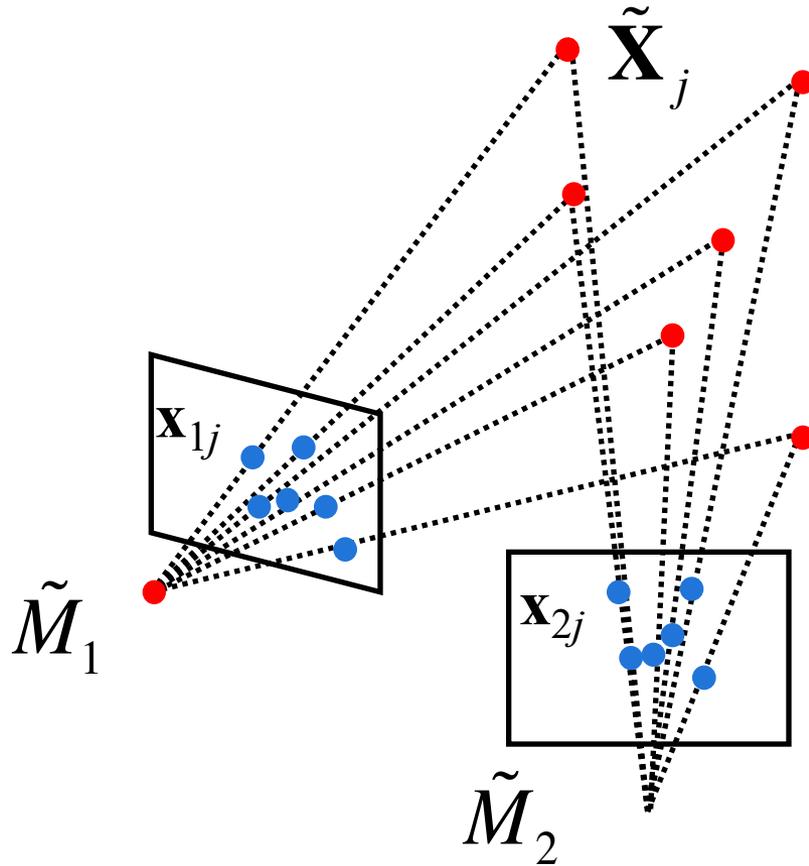
[Eq. 15]

[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



$$\mathbf{x}_{1j} = \tilde{M}_2 \tilde{\mathbf{X}}_j$$

$$\mathbf{x}_{2j} = \tilde{M}_1 \tilde{\mathbf{X}}_j$$

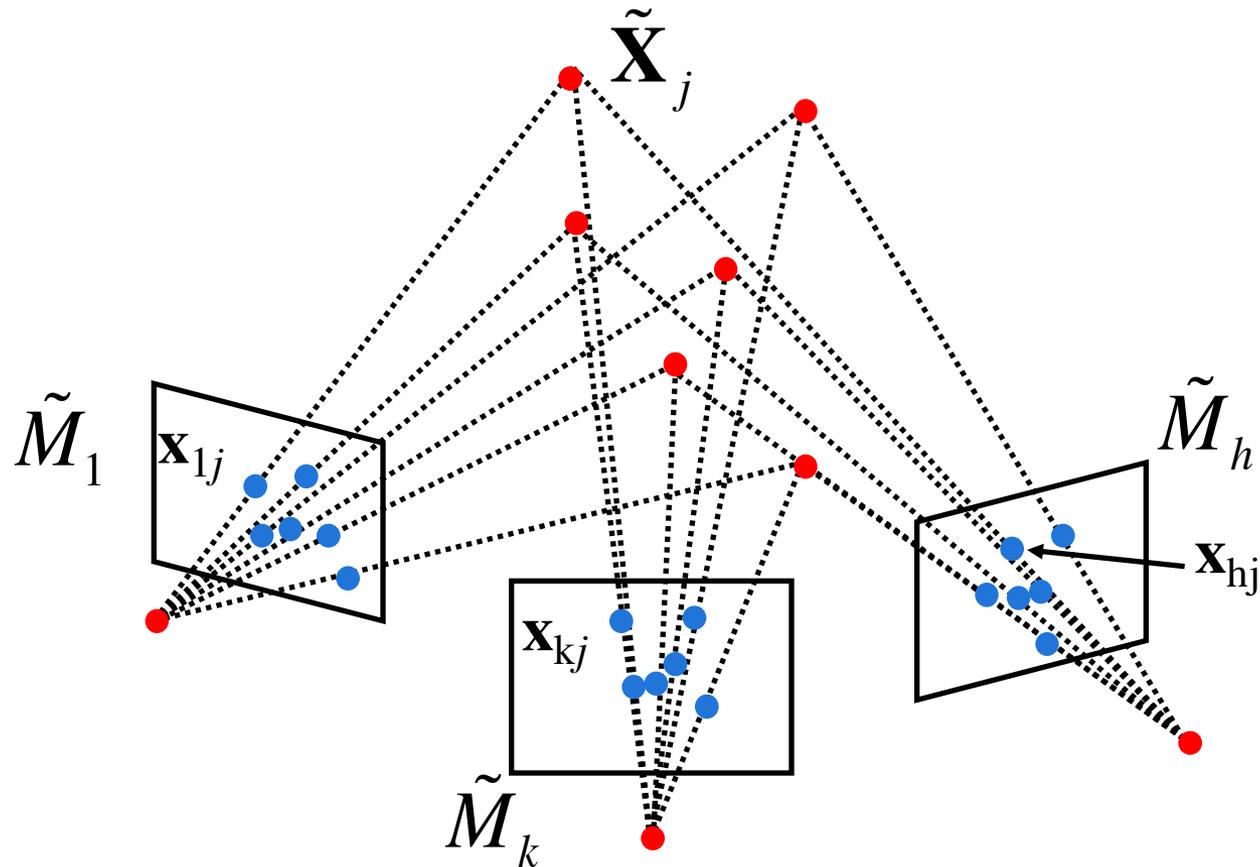
$$\tilde{M}_1 = \begin{bmatrix} I & 0 \end{bmatrix}$$

$$\rightarrow \tilde{\mathbf{X}}_j \text{ For } j = 1, \dots, n$$

$$\tilde{M}_2 = \begin{bmatrix} - & [\mathbf{e}_x] \mathbf{F} & \mathbf{e} \end{bmatrix}$$

3D points can be computed from camera matrices via SVD (see page 312 of HZ for details)

# Algebraic approach: the N-views case



- 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  $I_h$

- Pairwise solutions may be combined together using *bundle adjustment*

# 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

The bundle adjustment approach addresses some of these limitations

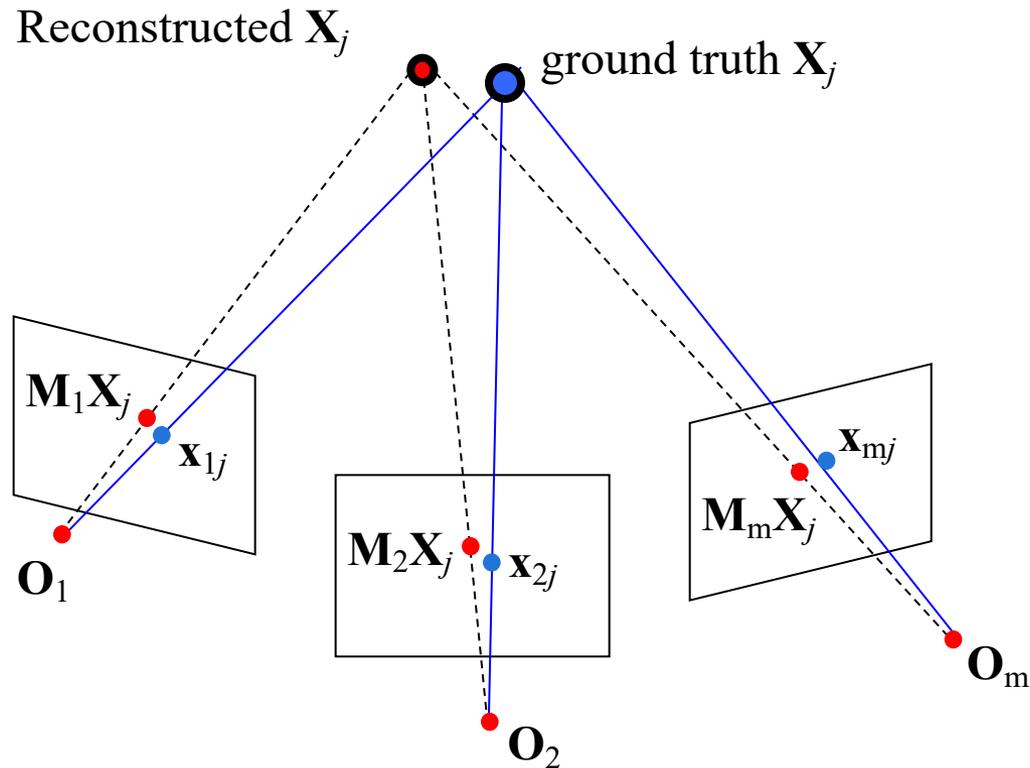
# Structure-from-Motion Algorithms

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

# Bundle adjustment

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

$$E(\mathbf{M}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{M}_i \mathbf{X}_j)^2$$





# Bundle adjustment

- **Advantages**

- Handle large number of views
- Handle missing data

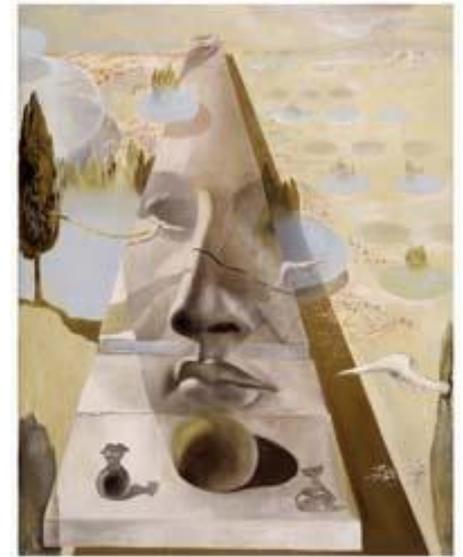
- **Limitations**

- 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

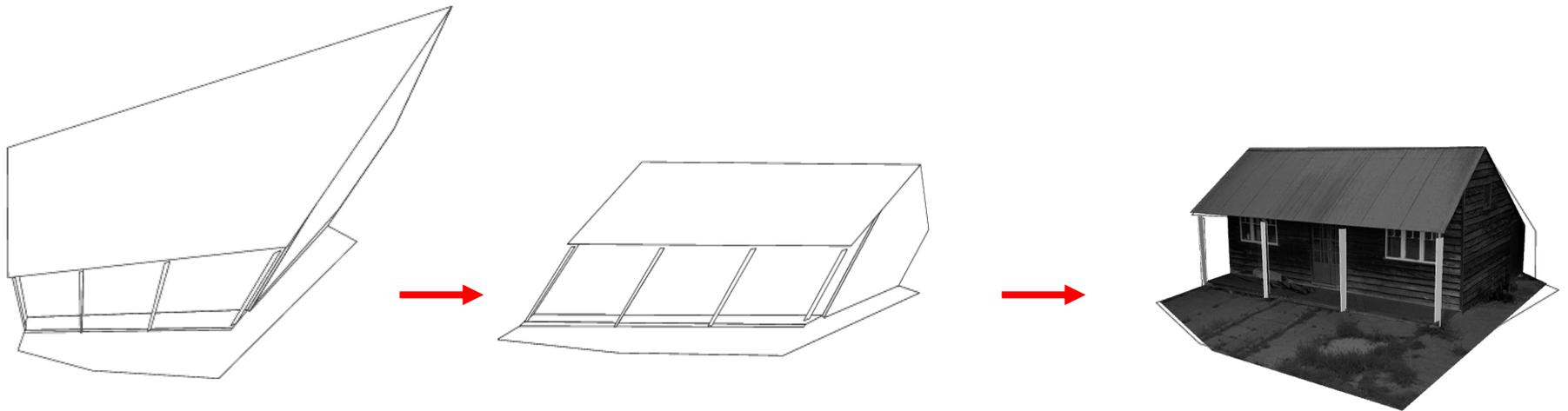
## Multi-view geometry

- 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



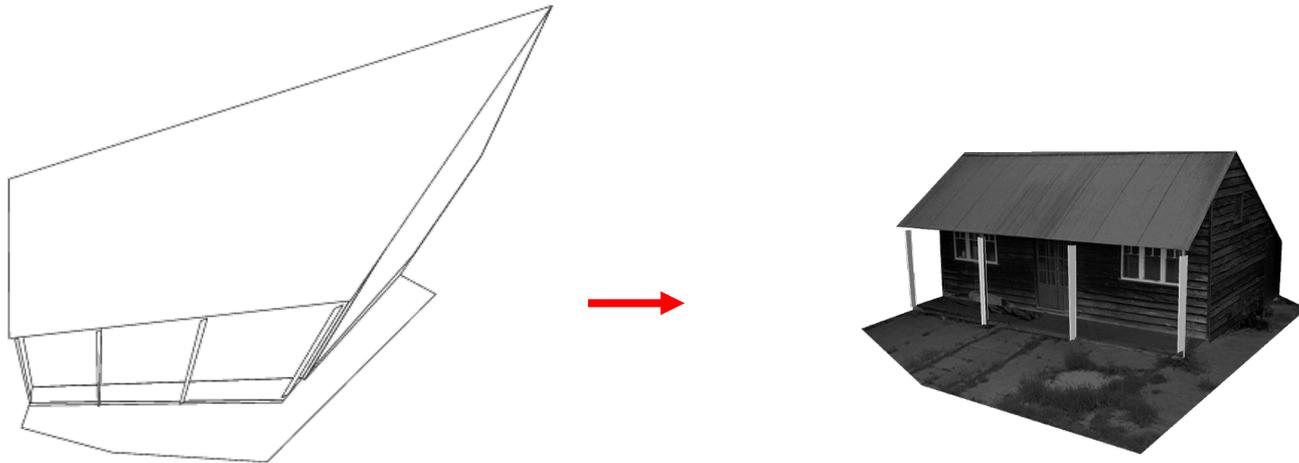
# Self-calibration

[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]

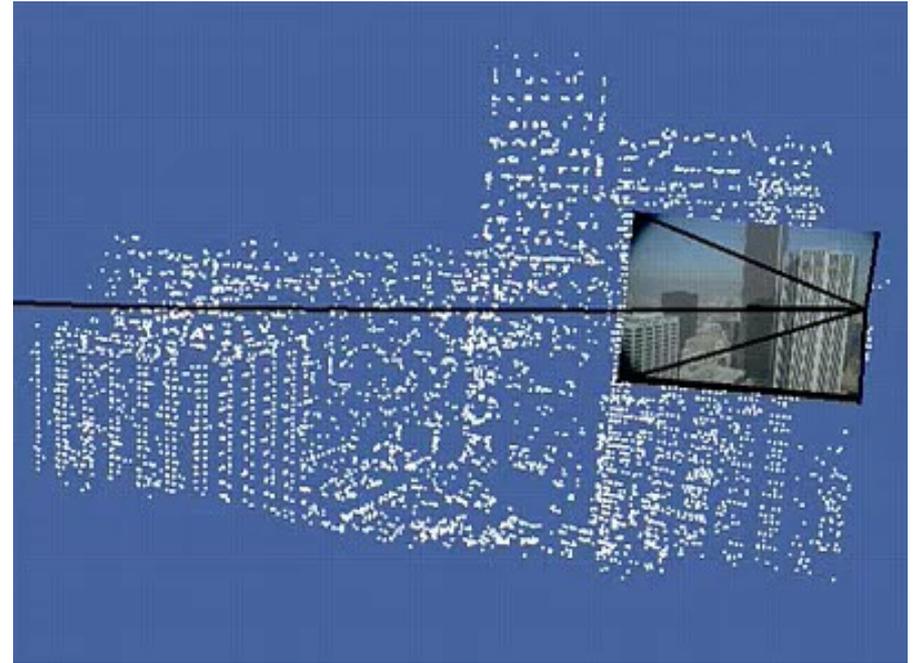
# Lecture 7

## Multi-view geometry



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

# Structure from motion problem



Courtesy of Oxford **Visual Geometry Group**

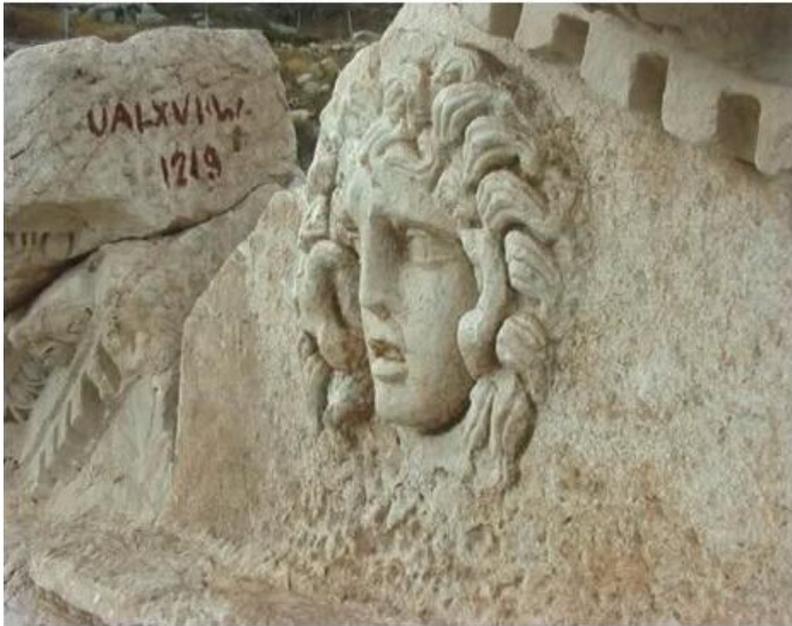
Lucas & Kanade, 81  
Chen & Medioni, 92  
Debevec et al., 96  
Levoy & Hanrahan, 96  
Fitzgibbon & Zisserman, 98  
Triggs et al., 99  
Pollefeys et al., 99  
Kutulakos & Seitz, 99

Levoy et al., 00  
Hartley & Zisserman, 00  
Dellaert et al., 00  
Rusinkiewicz 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

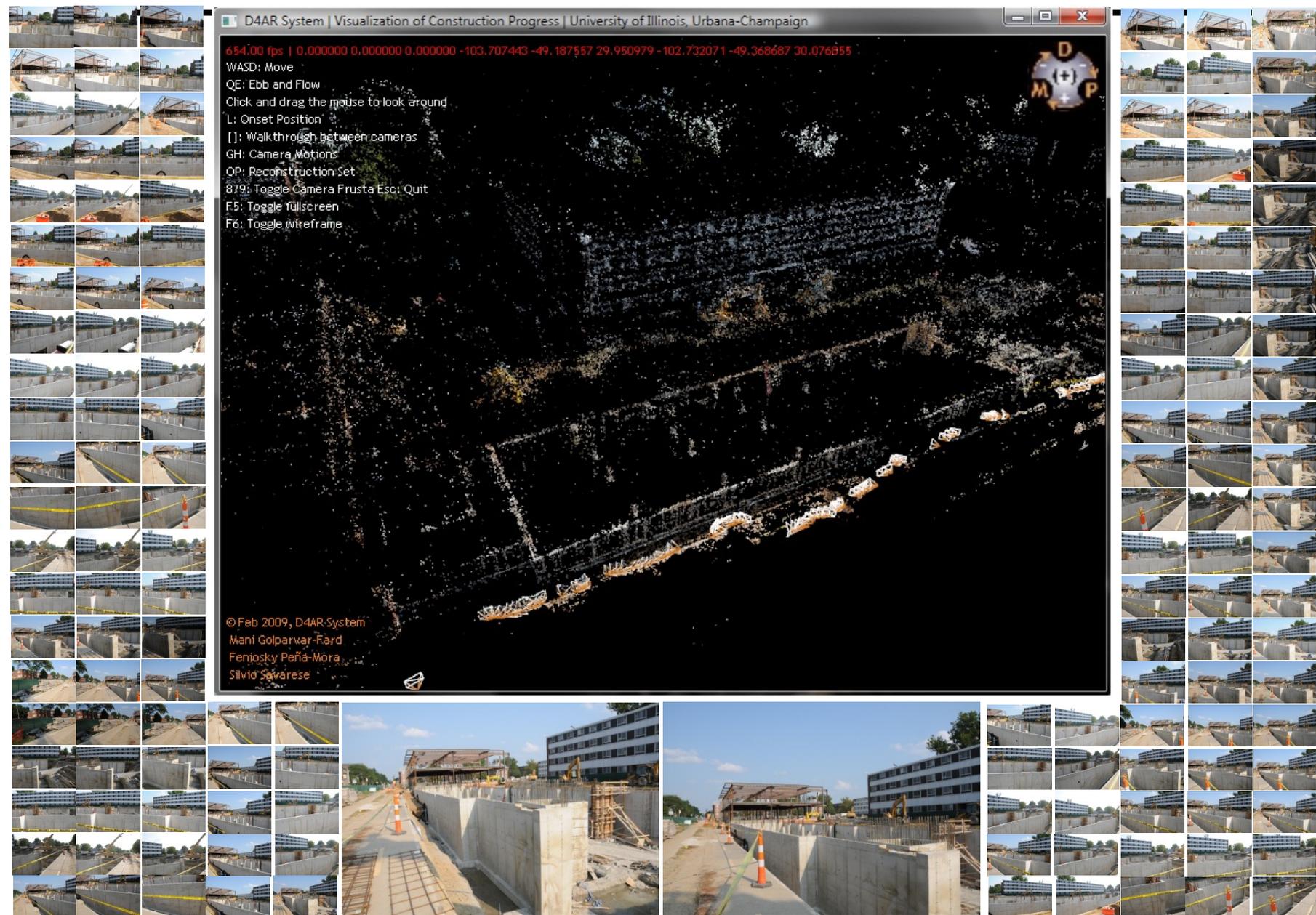
M. Pollefeys et al 98–



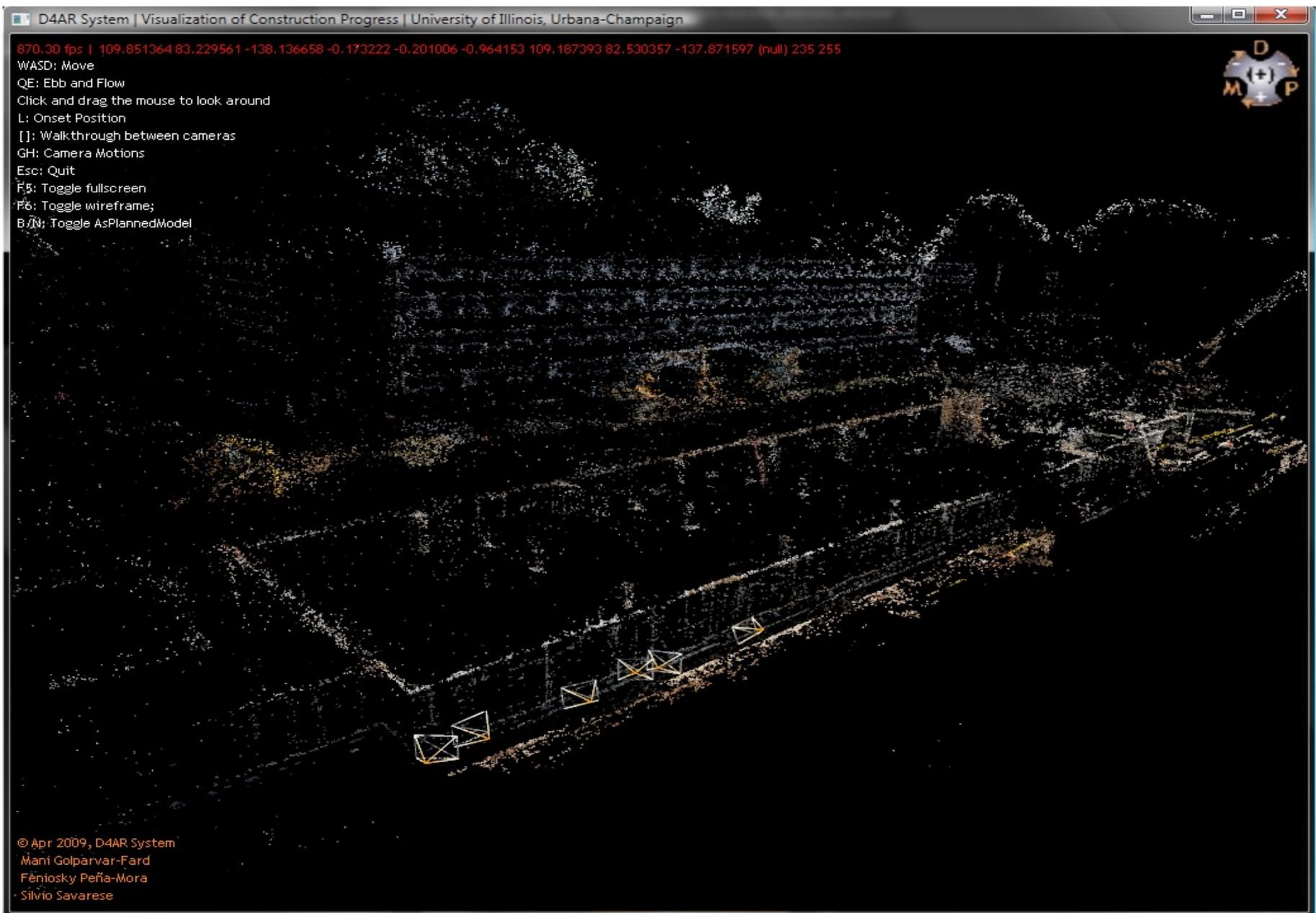
# Incremental reconstruction of construction sites

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

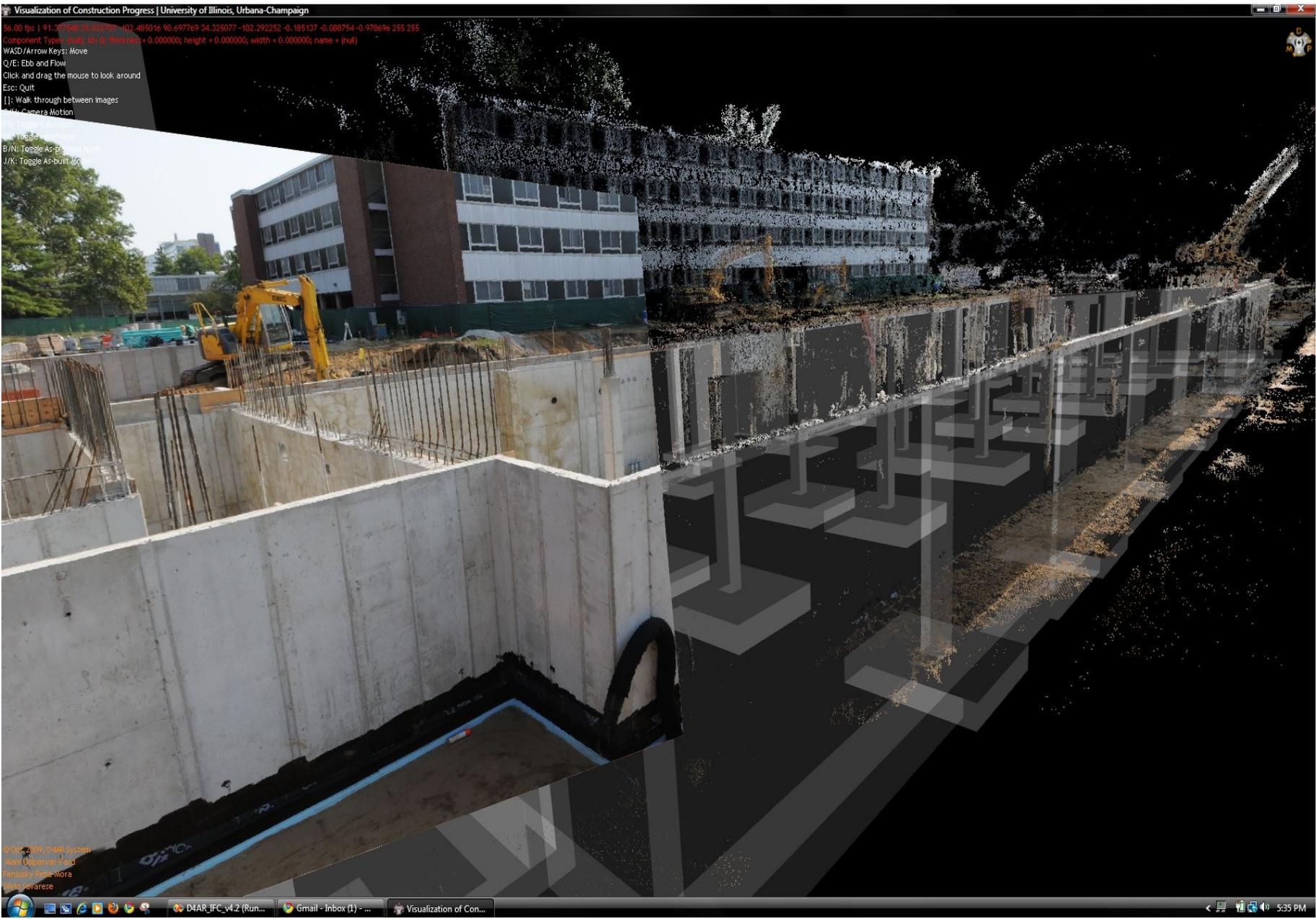
Golparvar-Fard, Pena-Mora, Savarese 2008



# 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

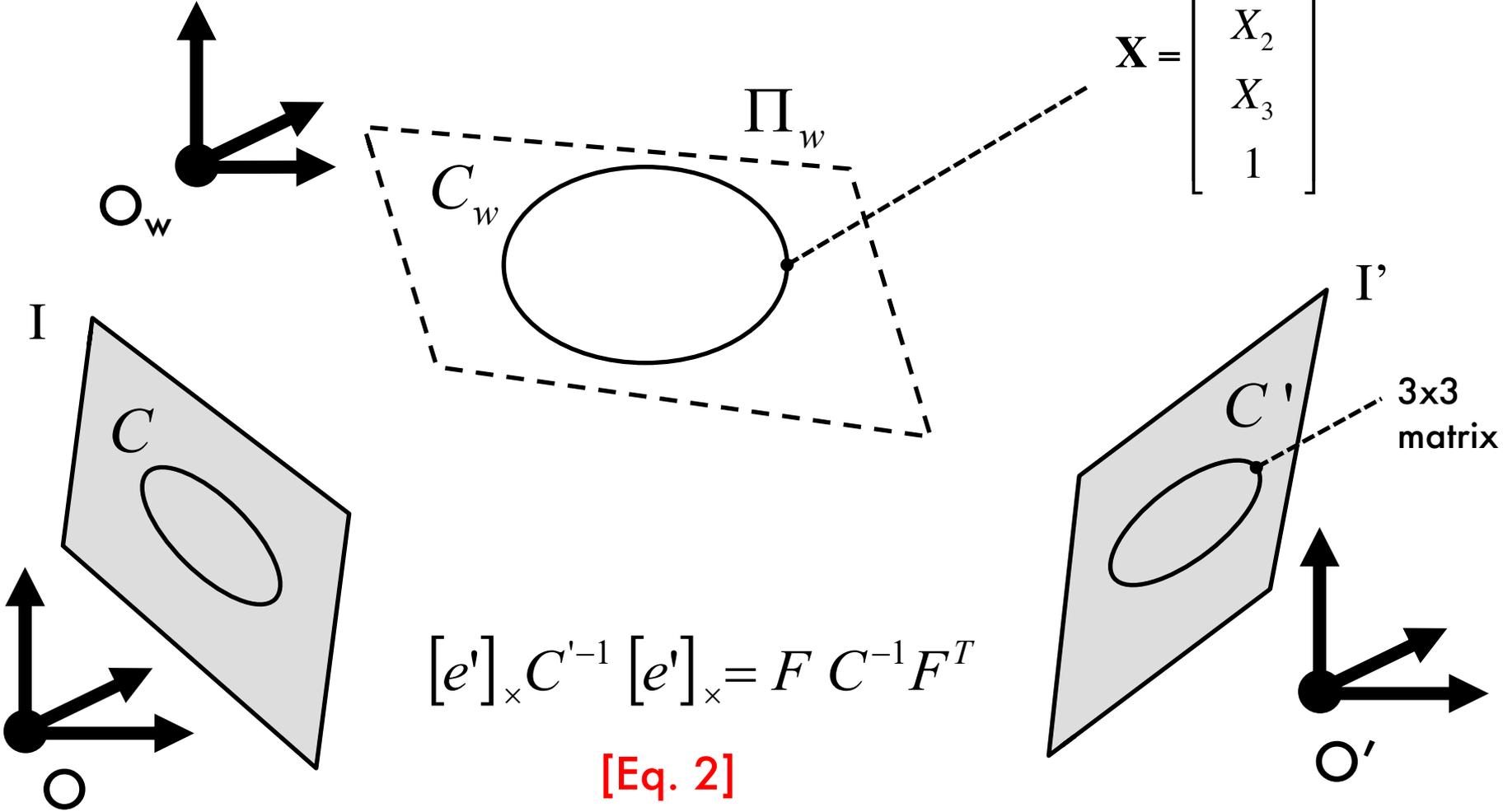
We use the following results:

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

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

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

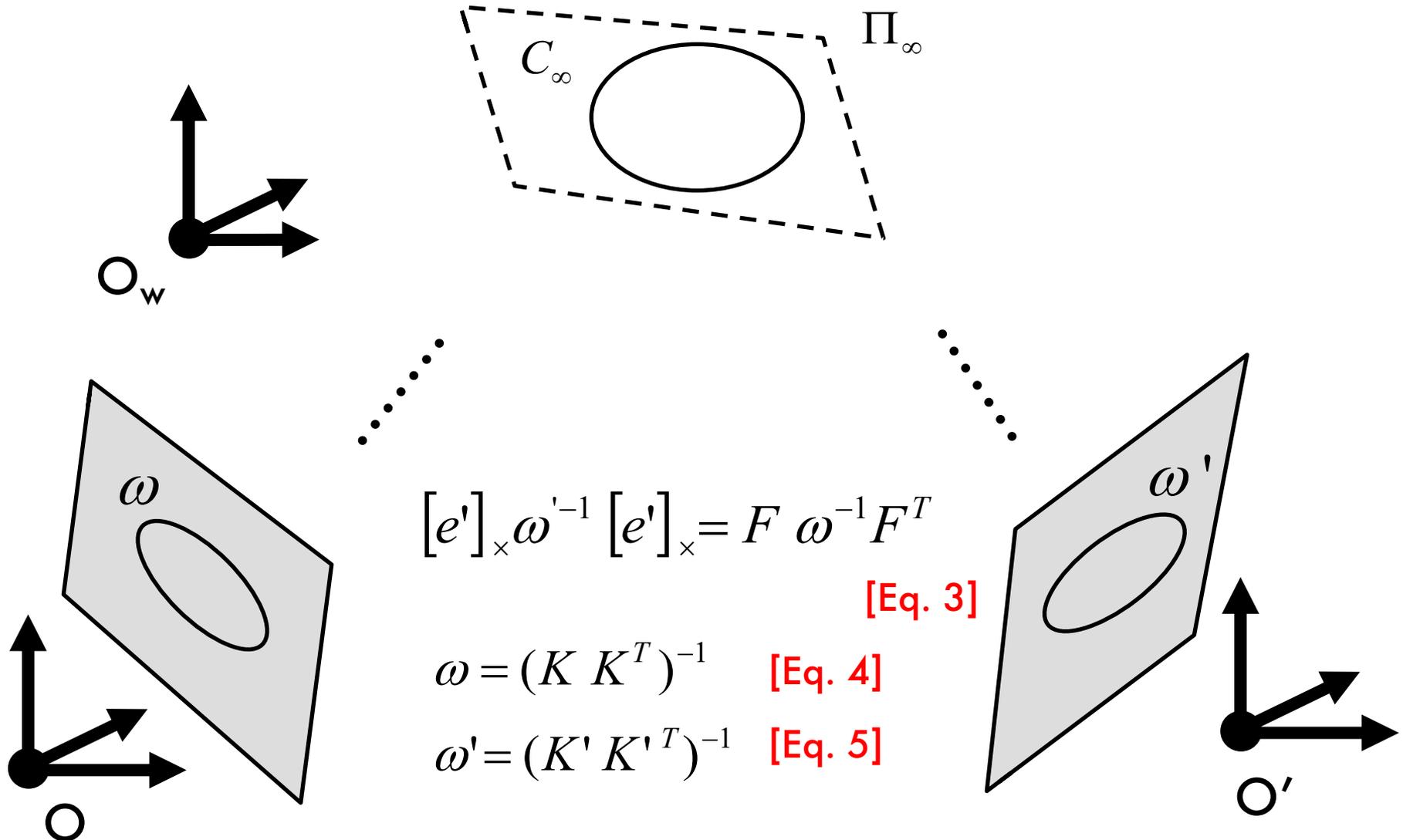


$$[e']_x C'^{-1} [e']_x = F C^{-1} F^T$$

$$\text{[Eq. 2]}$$

# 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  $\sigma_i$  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]**  $[e']_{\times} \omega^{-1} [e']_{\times} = F \omega^{-1} F^T$  becomes trivial

Since:  $F = [e']_{\times}$

# 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:

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

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

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

# Algebraic approach Multi-view approach

Suppose we have a projective reconstruction

Let  $H$  be a homography such that:

$$\left\{ \begin{array}{l} \text{First perspective camera is canonical: } \tilde{M}_1 = [ I \quad 0 ] \text{ [Eq. 11]} \\ \text{i}^{\text{th}} \text{ perspective reconstruction of the camera (known): } \tilde{M}_i = [ A_i \quad b_i ] \\ \hspace{20em} \text{[Eq. 12]} \end{array} \right.$$

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

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:

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

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

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

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  $K_s$  to generate enough equations on the unknowns

<i>Condition</i>	<i>N. Views</i>
<ul style="list-style-type: none"><li>• Constant internal parameters</li></ul>	3
<ul style="list-style-type: none"><li>• Aspect ratio and skew known</li><li>• Focal length and offset vary</li></ul>	4
<ul style="list-style-type: none"><li>• Skew =0, all other parameters vary</li></ul>	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