Lecture 7

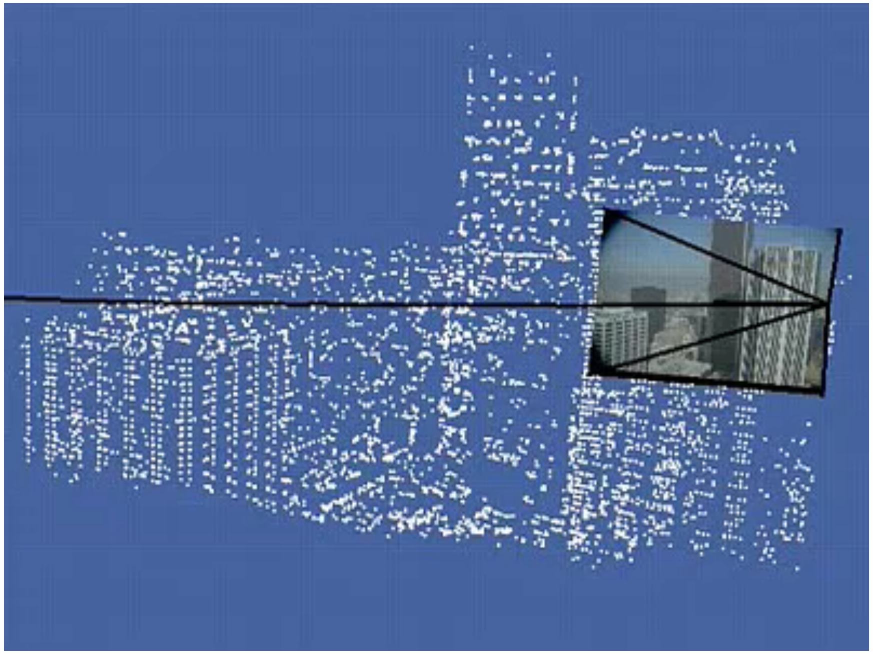
Multi-view geometry

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

Reading:

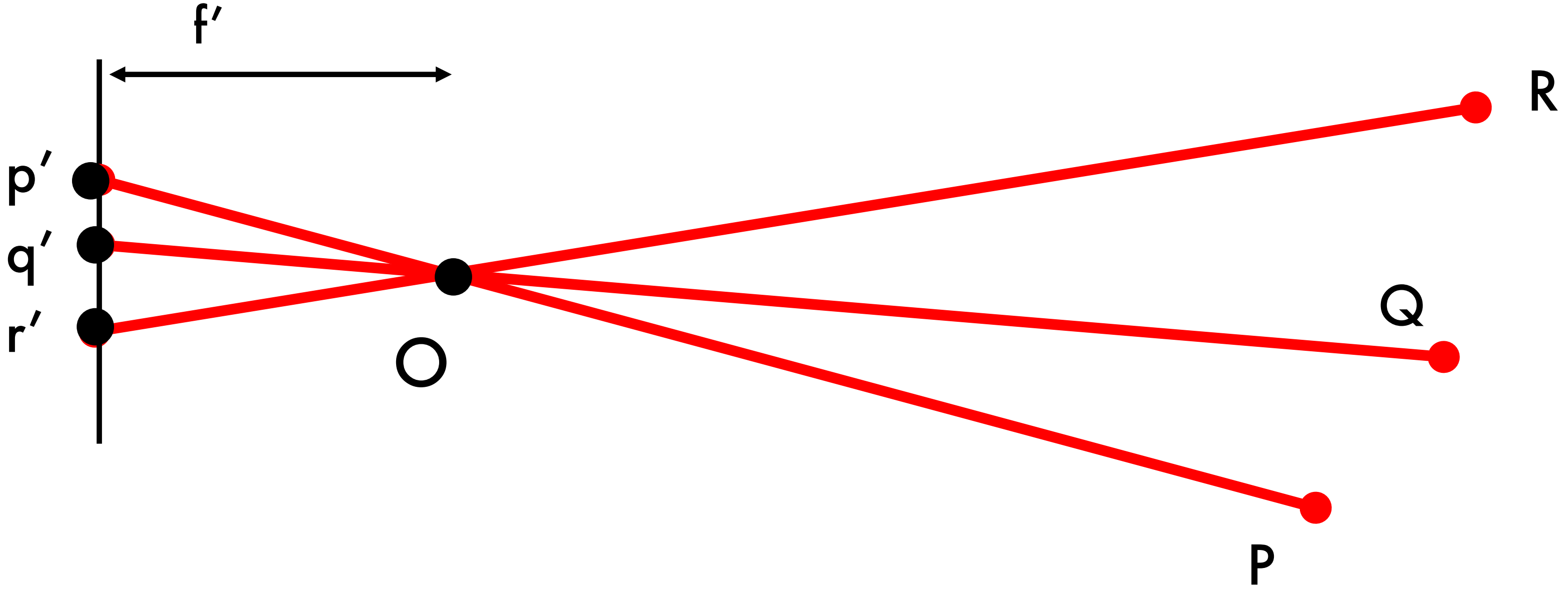
[HZ] Chapter 10 “3D reconstruction of cameras and structure”
Chapter 18 “N-view computational methods”
Chapter 19 “Auto-calibration”
[FP] Chapter 13 “projective structure from motion”
[Szelisky] Chapter 7 “Structure from motion”

Structure from motion problem



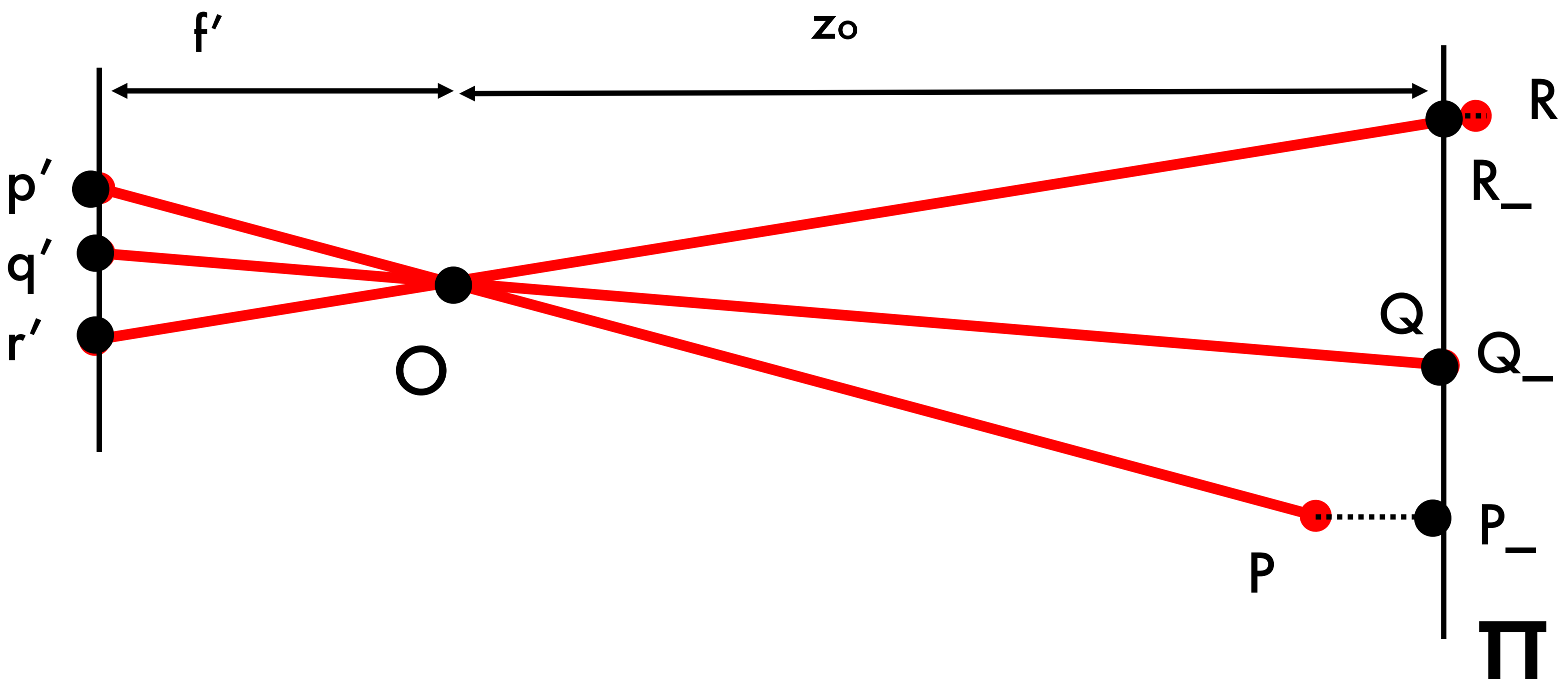
Courtesy of Oxford **Visual Geometry Group**

Projective camera

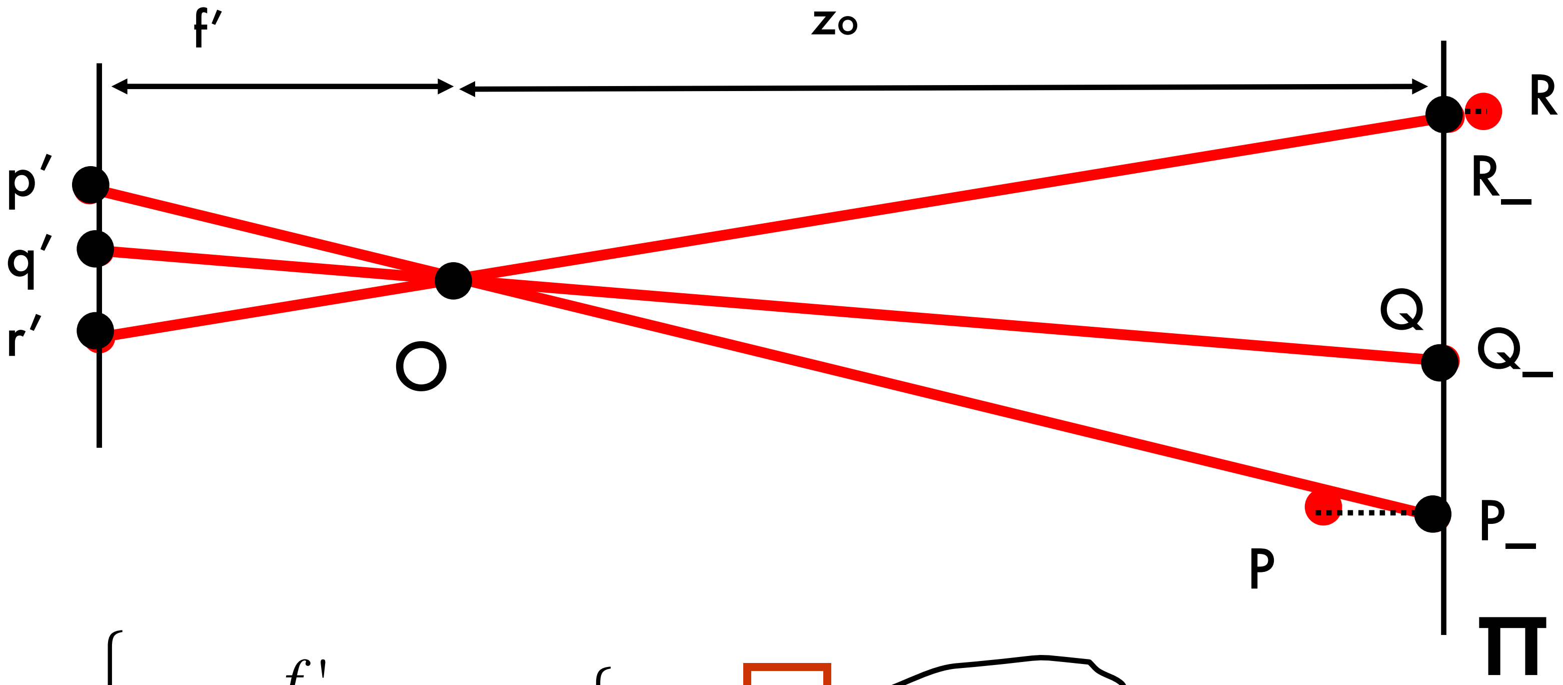


Weak perspective projection

When the relative scene depth is small compared to its distance from the camera



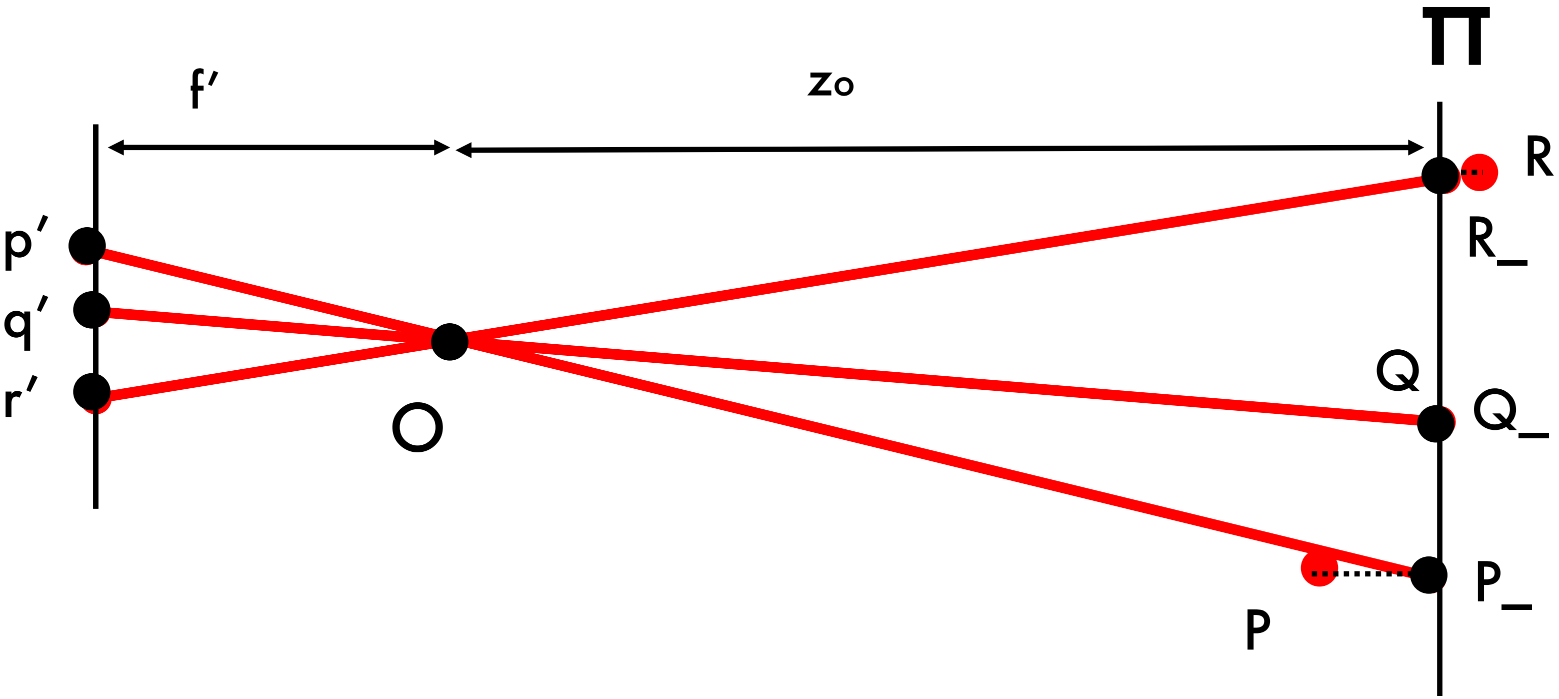
Weak perspective projection



$$\begin{cases} x' = \dfrac{f'}{z} x \\ y' = \dfrac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = \boxed{\dfrac{f'}{z_0}} x \\ y' = \dfrac{f'}{z_0} y \end{cases}$$

Magnification m

Weak perspective projection

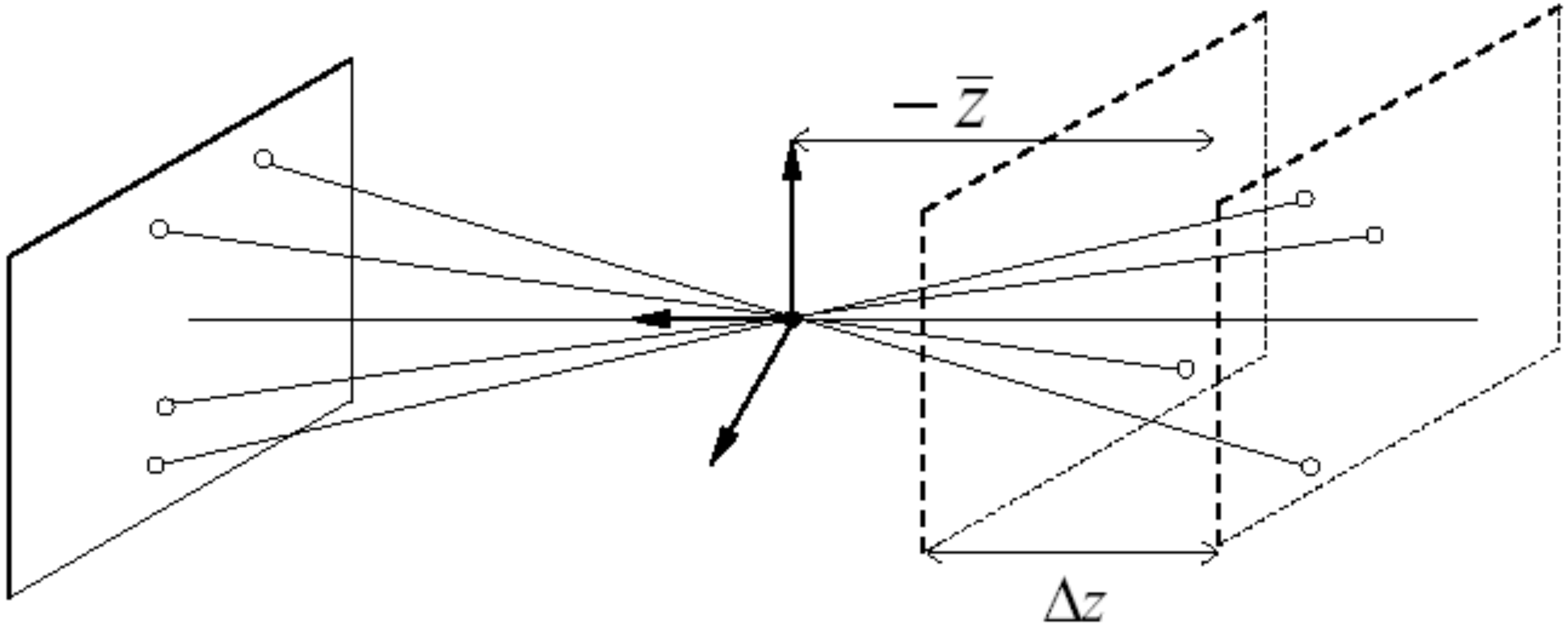


Projective (perspective)

Weak perspective

$$M = K\begin{bmatrix} R & T \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & \mathbf{1} \end{bmatrix} \rightarrow \quad M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Special Case: Weak Perspective (Affine Projection)



If $\Delta z << -\bar{z}$: $\begin{aligned} x' &\approx -mx \\ y' &\approx -my \end{aligned}$ $\quad m = -\dfrac{f'}{\bar{z}}$

Justified if scene depth is small relative to average distance from camera

$$P' = \mathrm{M}\, P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ \mathbf{m}_3 P_W \end{bmatrix}$$

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

$$\xrightarrow{\mathbf{E}} \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right)$$

Perspective: projective transformation

$$P' = \mathrm{M}\, P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ 1 \end{bmatrix}$$

$$M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} & \mathbf{m}_1 & & \\ & \mathbf{m}_2 & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

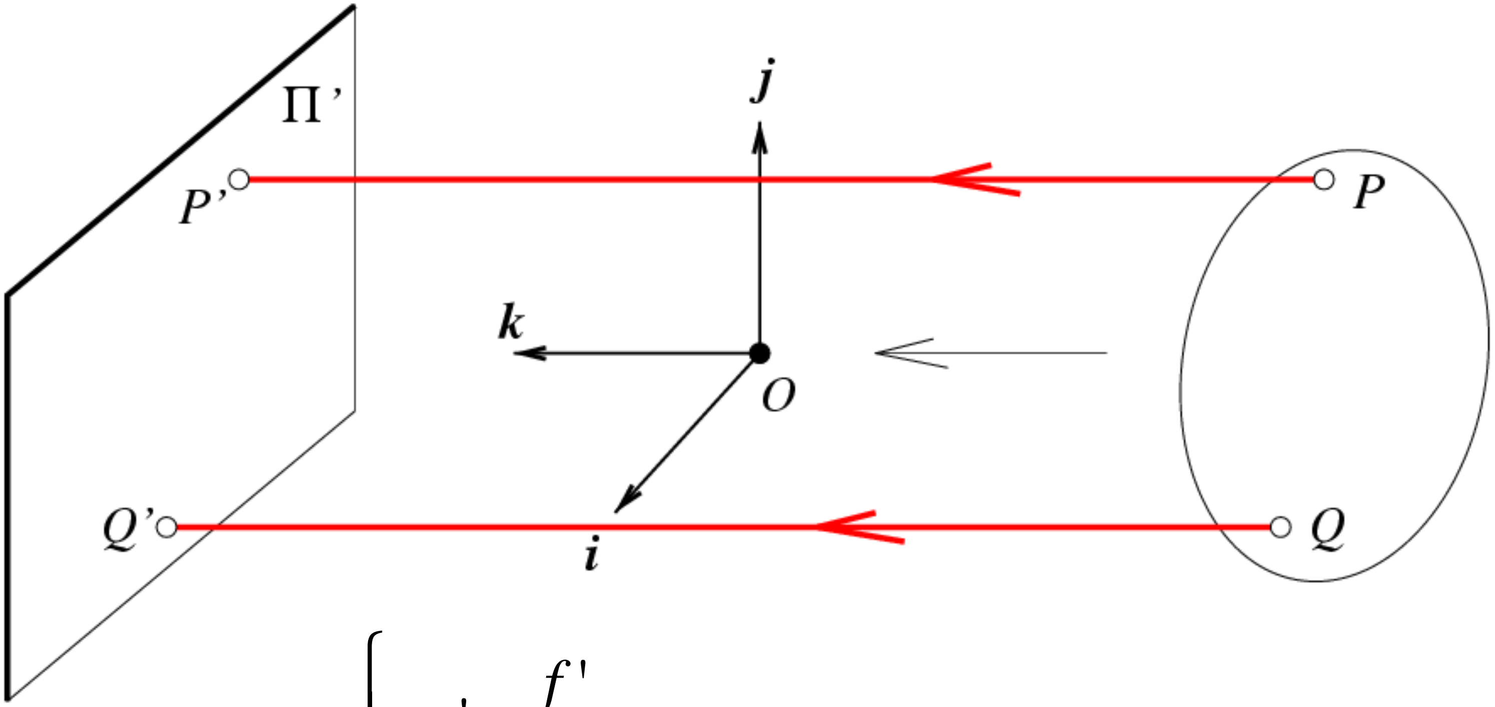
$$\xrightarrow{\mathbf{E}} (\underset{\uparrow}{\mathbf{m}_1} P_w, \underset{\uparrow}{\mathbf{m}_2} P_w)$$

magnification

Weak Prospective: Affine Transformatoin

Orthographic (affine) projection

Distance from center of projection to image plane is infinite



$$\begin{cases} x' = \dfrac{f'}{z} x \\ \\ y' = \dfrac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = x \\ y' = y \end{cases}$$

Affine
Transformation

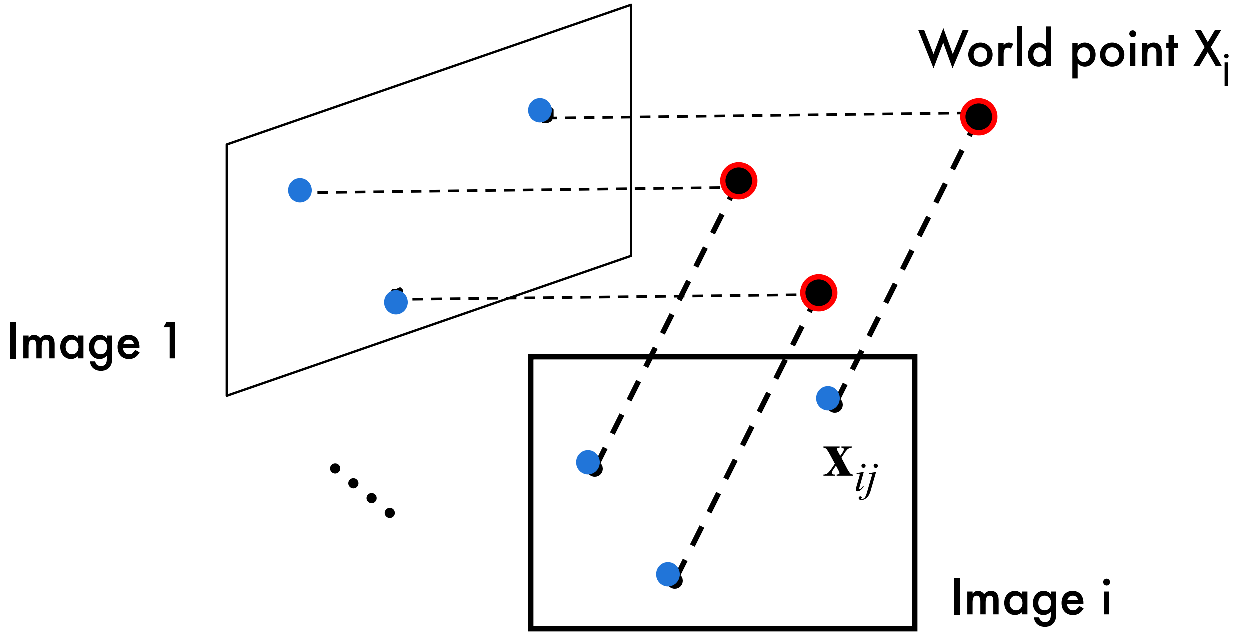
Pros and Cons of These Models

- Weak perspective results in much simpler math.
  - Accurate when object is small and distant.
  - Most useful for recognition.

- Pinhole perspective is much more accurate for modeling the 3D-to-2D mapping.
  - Used in structure from motion or SLAM.

Affine structure from motion

(simpler problem)

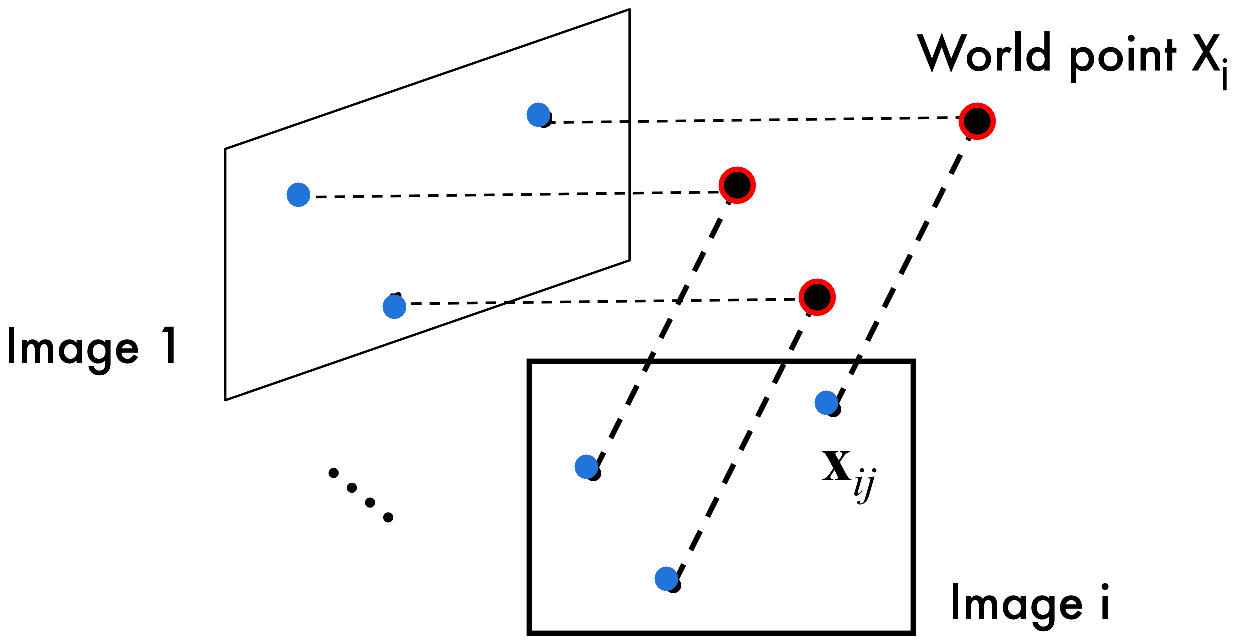


From the mxn observations $\mathbf{x}_{ij}$, estimate:

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

Affine structure from motion

(simpler problem)



For the affine case (in Euclidean space)

$$\underset{2\text{x}1}{\mathbf{x}_{ij}} = \underset{2\text{x}3}{\mathbf{A}_i}\,\underset{3\text{x}1}{\mathbf{X}_j} + \underset{2\text{x}1}{\mathbf{b}_i}$$

[Eq. 4]

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

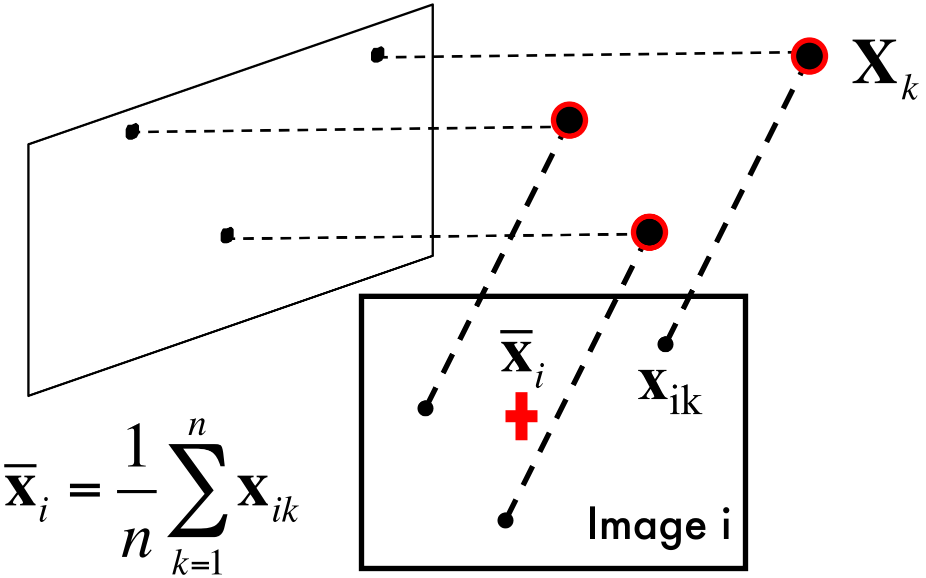
C. Tomasi and T. KanadeShape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

- Data centering
- Factorization

A factorization method - Centering the data

Centering: subtract the centroid of the image points

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



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

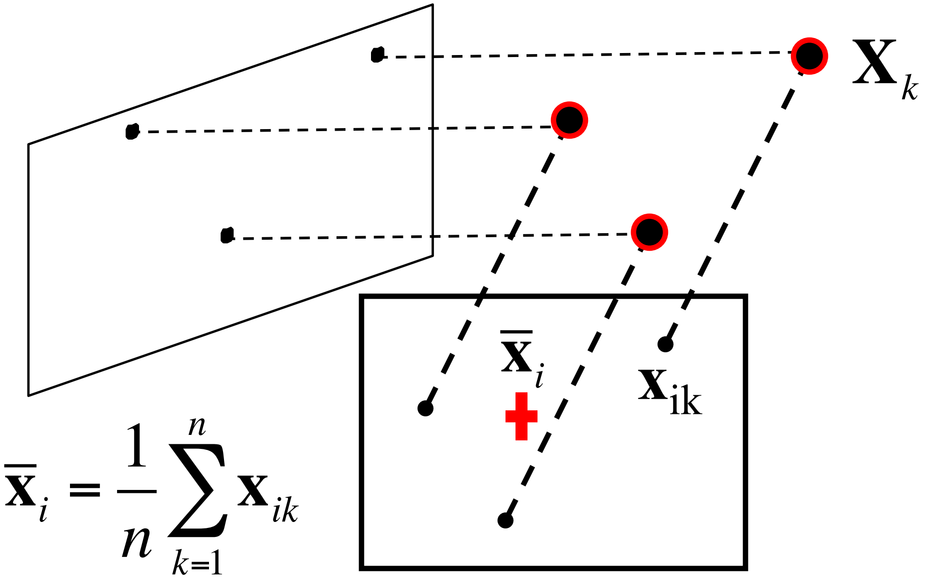
A factorization method - Centering the data

Centering: subtract the centroid of the image points

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

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

[Eq. 4]



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

A factorization method - Centering the data

Centering: subtract the centroid of the image points

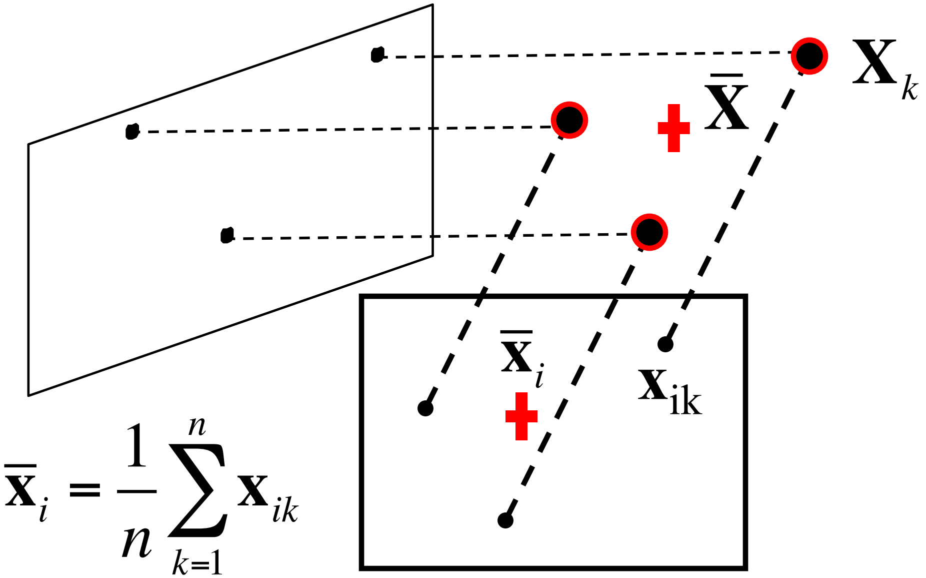
[Eq. 6] $$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\mathbf{A}_i\mathbf{X}_k - \frac{1}{n}\sum_{k=1}^{n}\mathbf{b}_i$$

$$= \mathbf{A}_i\left(\mathbf{X}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k\right) = \mathbf{A}_i\left(\mathbf{X}_j - \bar{\mathbf{X}}\right)$$

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

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

[Eq. 4]



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

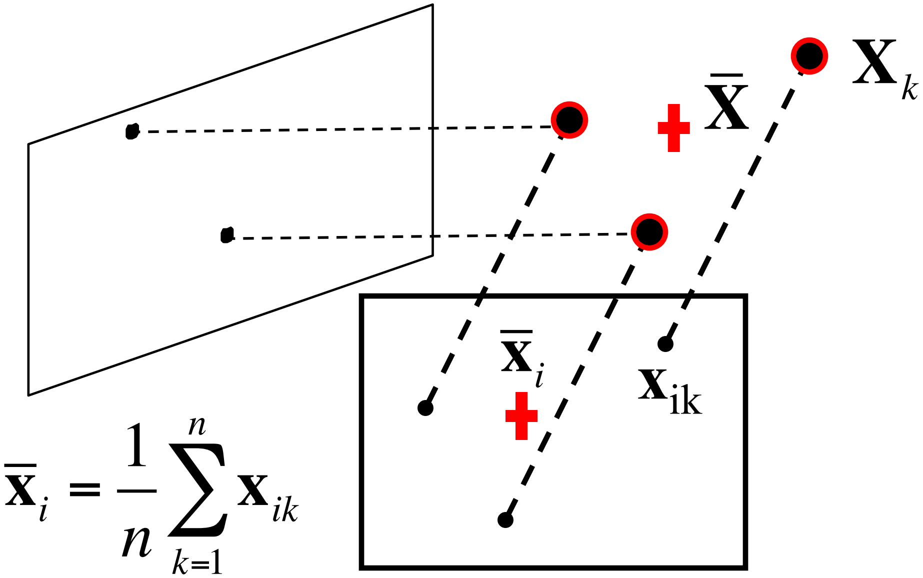
$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k$$ [Eq. 7]

Centroid of 3D points

A factorization method - Centering the data

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

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



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

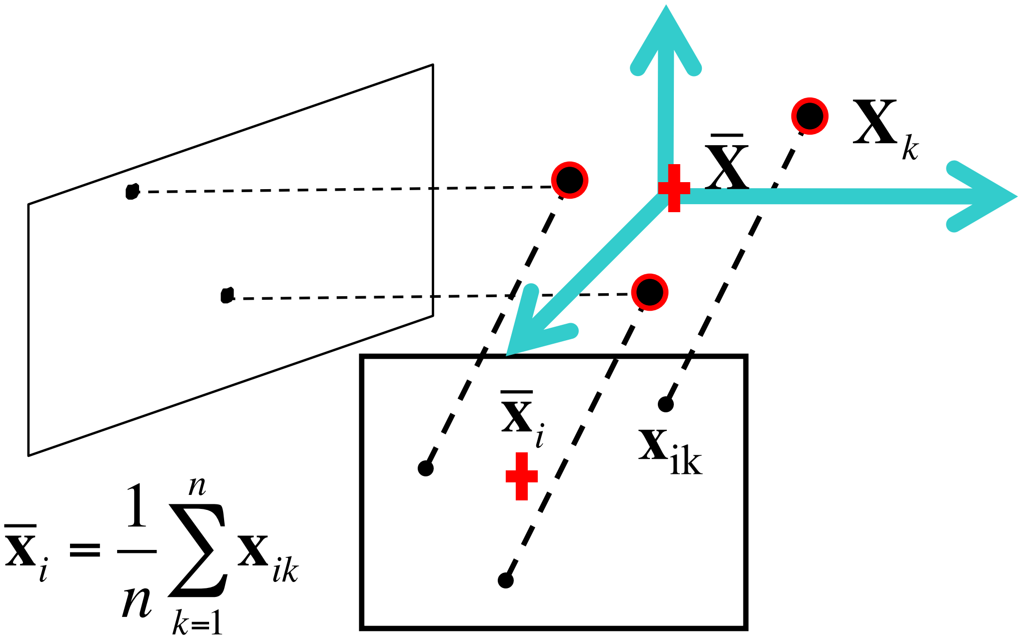
$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k$$ [Eq. 7]

Centroid of 3D points

A factorization method - Centering the data

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

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



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

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

Centroid of 3D points

A factorization method - factorization

Let’s create a 2m × n data (measurement) matrix:

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

cameras ($2m$)

points (n)

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2x1 vector!

A factorization method - factorization

Let's create a 2m × n data (measurement) matrix:

$$\mathbf{D} = \underbrace{\begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}}_{(2m \times n)} = \underbrace{\boxed{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix}}}_{\substack{\text{cameras} \\ (2m \times 3)}} \mathbf{M} \; \underbrace{\boxed{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}}_{\text{points } (3 \times n)} \mathbf{S}$$

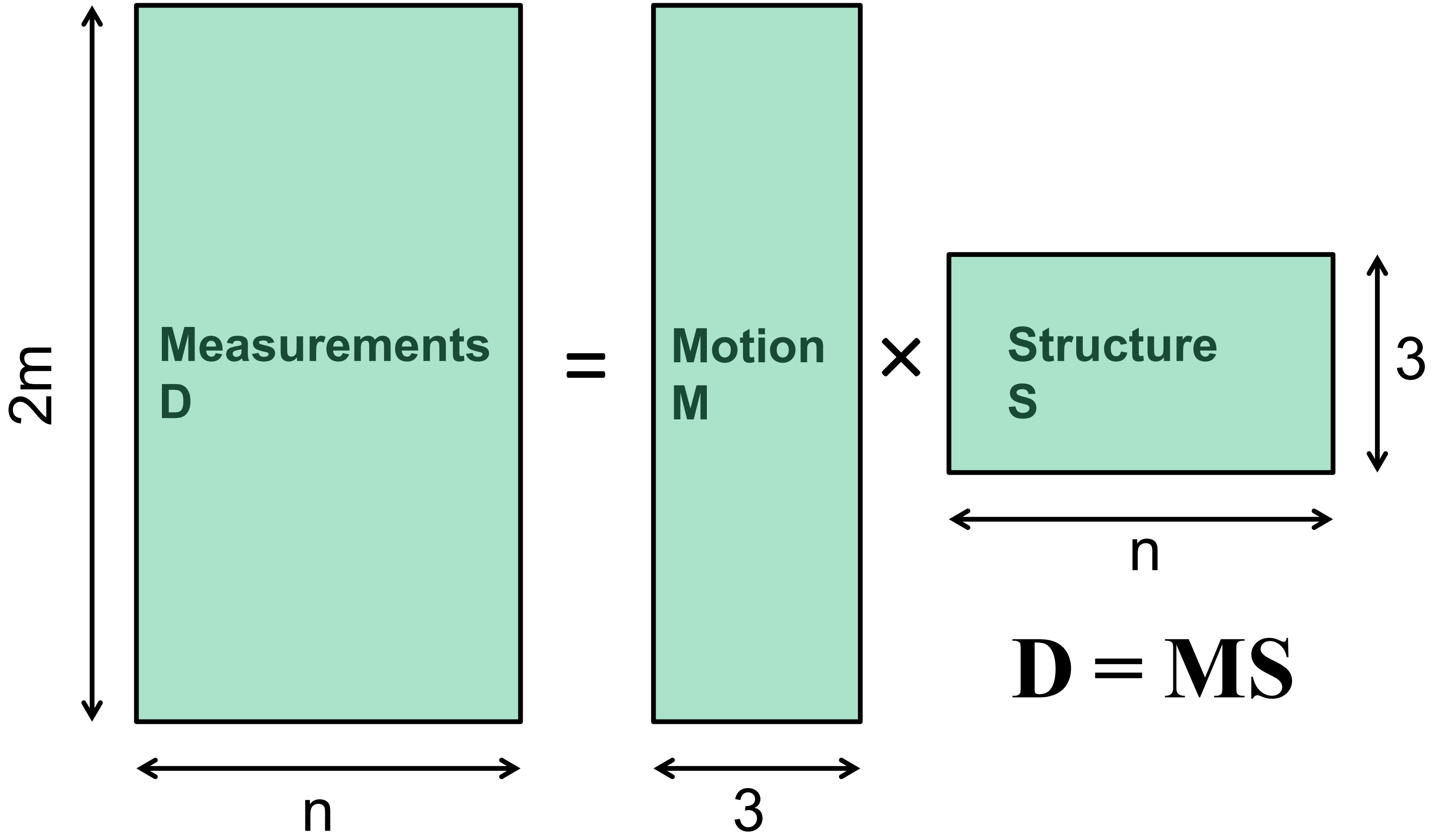
[Eq. 10]

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2x1 vector!
A$_i$ is 2x3 and **X**$_i$ is 3x1

The measurement matrix **D = M S** has rank 3
(it's a product of a 2mx3 matrix and 3xn matrix)

Factorizing the Measurement Matrix

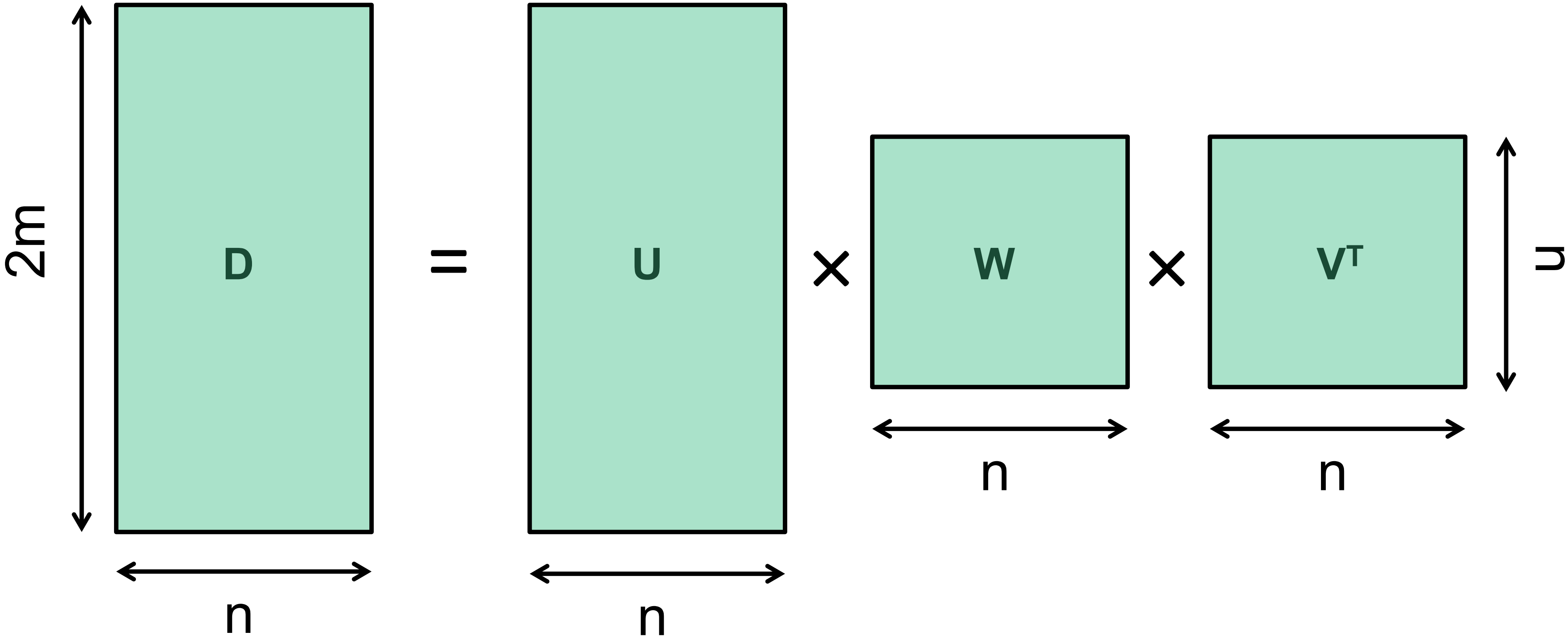
How to factorize D?



$$\mathbf{D} = \mathbf{MS}$$

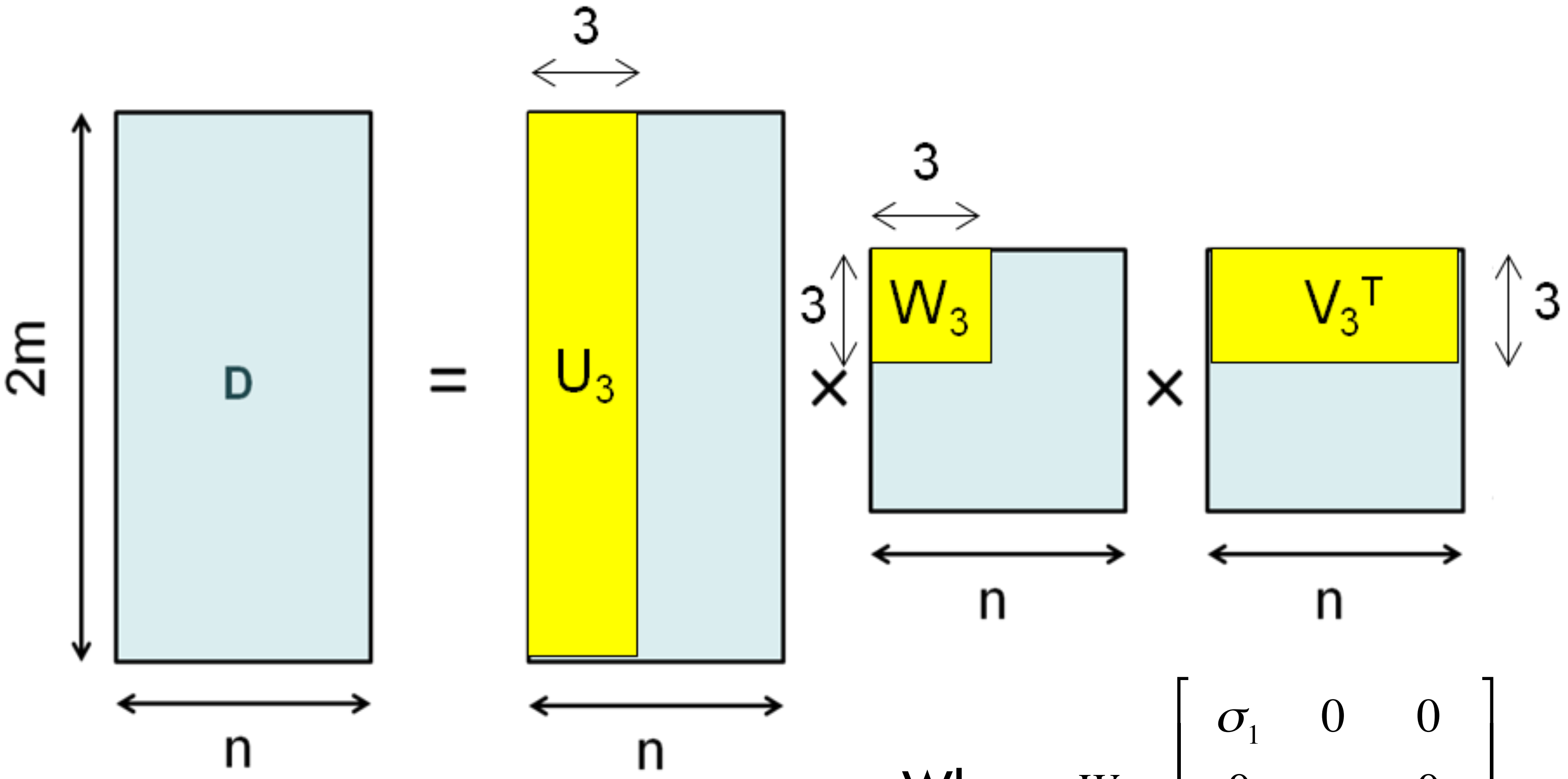
Factorizing the Measurement Matrix

- By computing the Singular value decomposition of D!



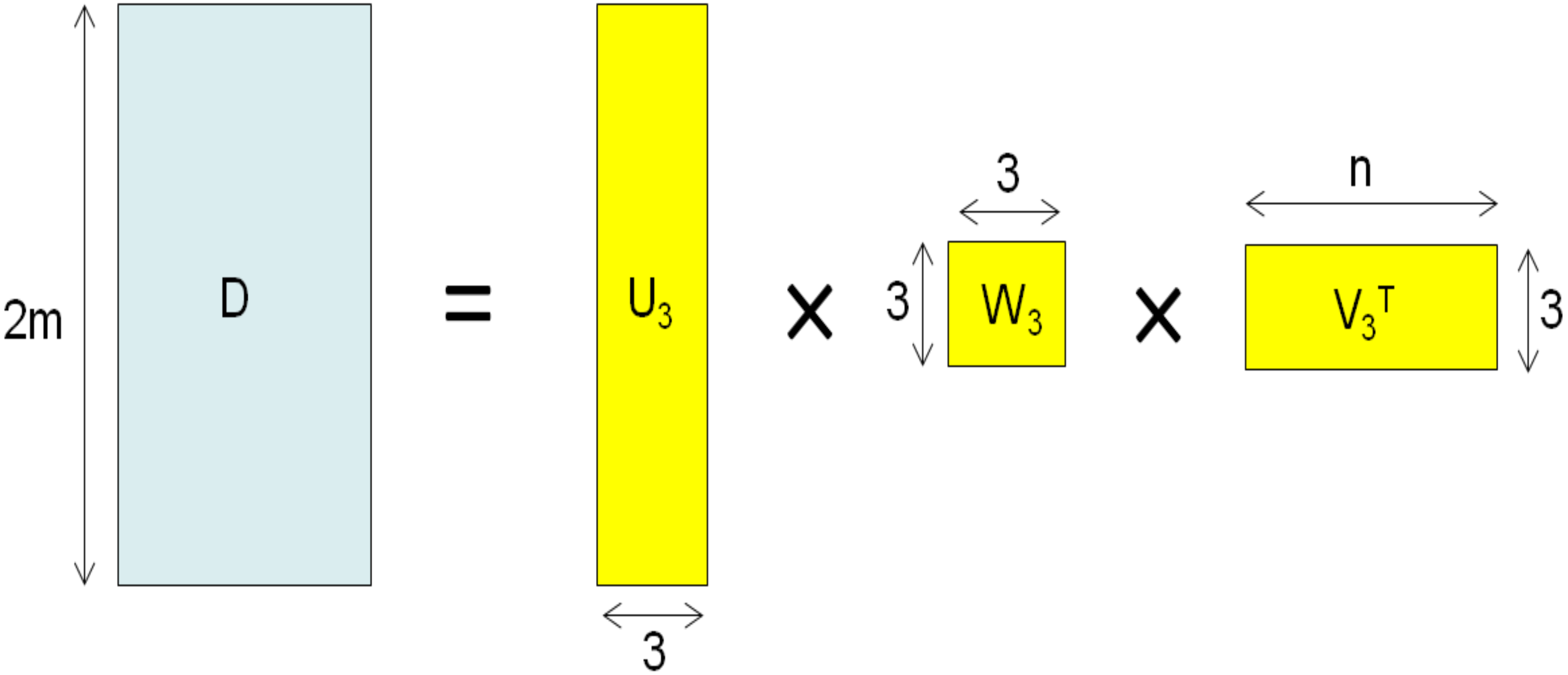
Factorizing the Measurement Matrix

Since rank (D)=3, there are only 3 non-zero singular values σ$_1$, σ$_2$ and σ$_3$

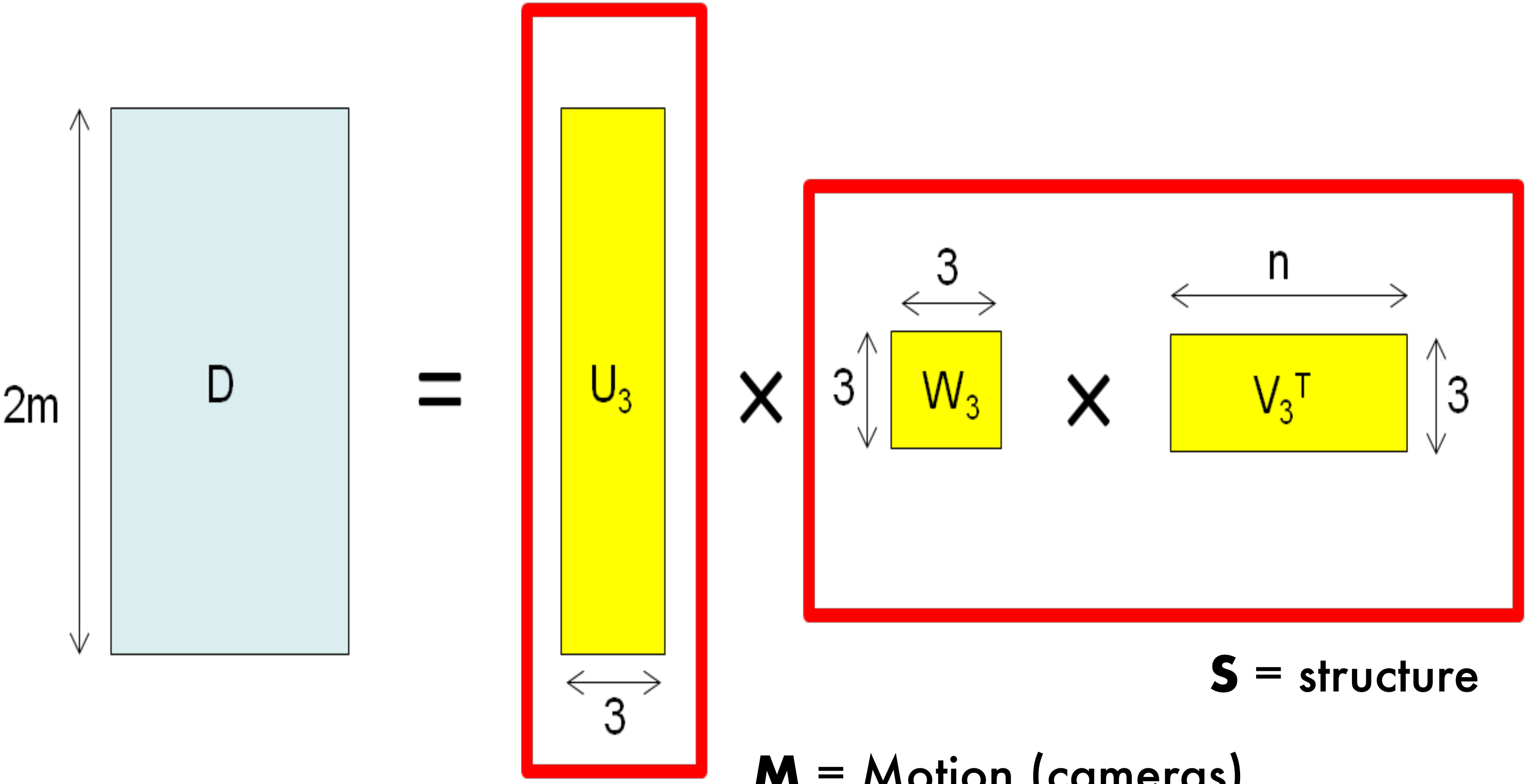


Where $W_3 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$ [Eq. 11]

Factorizing the Measurement Matrix



Factorizing the Measurement Matrix



S = structure

M = Motion (cameras)

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

Factorizing the Measurement Matrix

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

What is the issue here?

D has rank>3 because of:

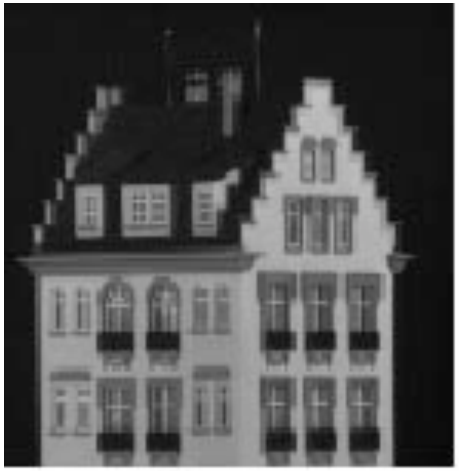
- measurement noise
- affine approximation

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 **D** in the sense of the Frobenius norm.

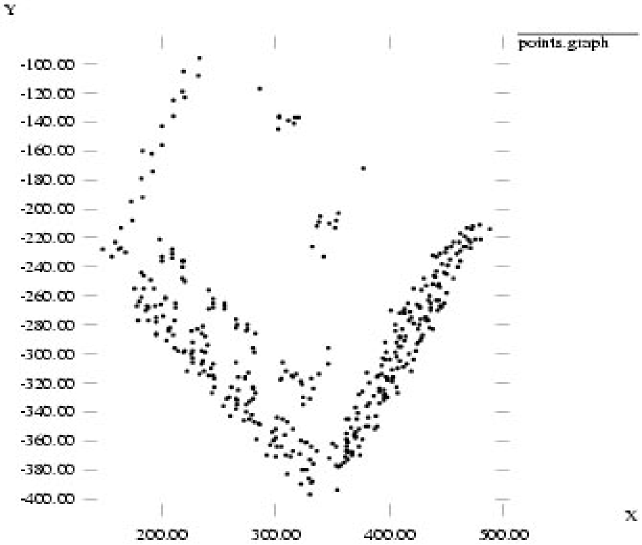
$$\mathbf{D} = \mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T \qquad \begin{cases} \mathbf{M} \approx \mathbf{U}_3 \\ \mathbf{S} \approx \mathbf{W}_3\mathbf{V}_3^T \end{cases}$$

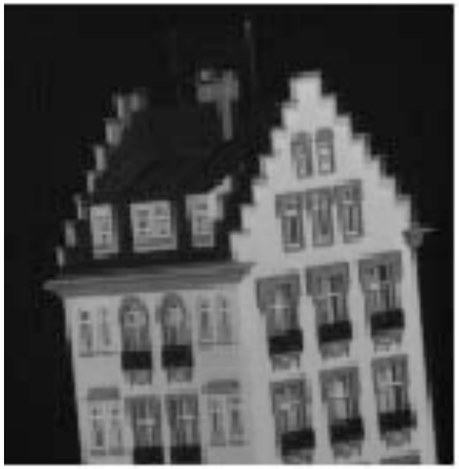
$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$$

Reconstruction results

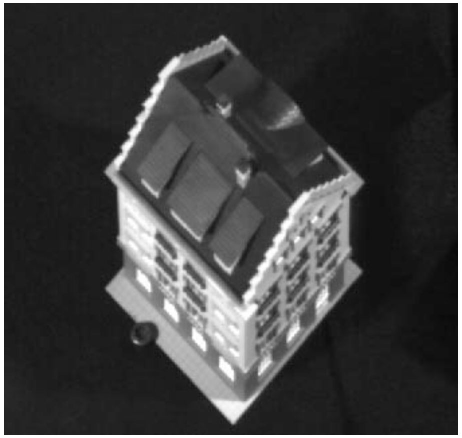


1





120



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

Results

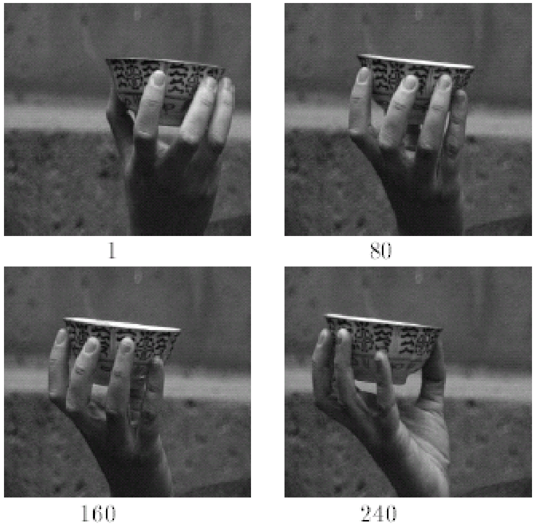


Figure 6.20: Four out of the 240 frames of the cup image stream.

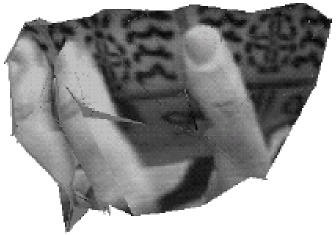


Figure 6.23: A front view of the cup and fingers, with the original image intensities mapped onto the resulting surface.

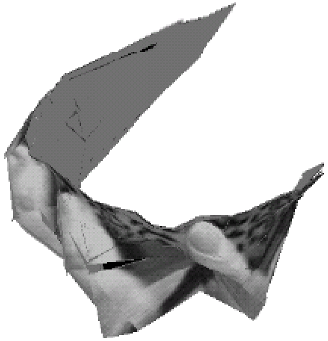
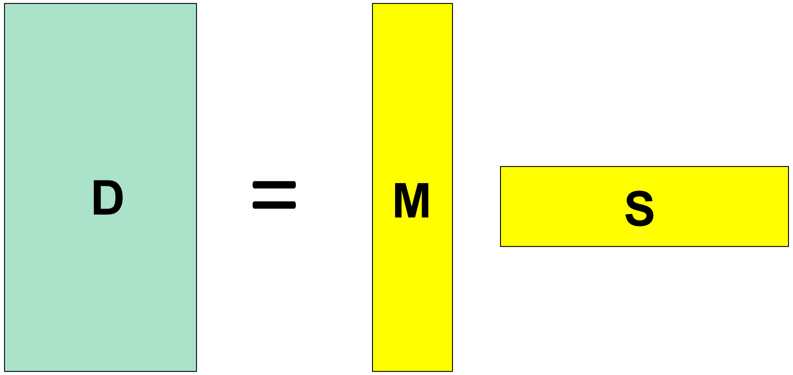
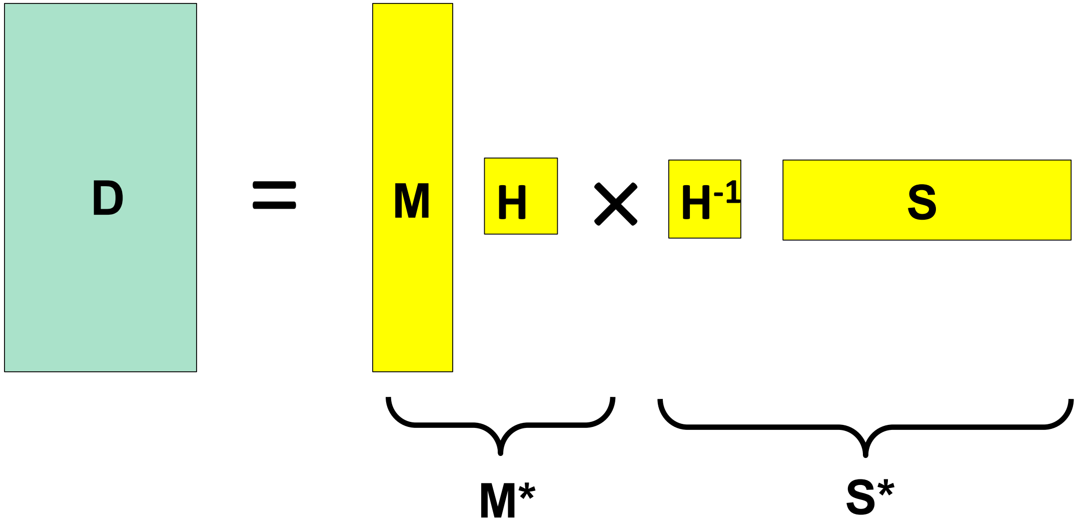


Figure 6.24: A view from above of the cup and fingers with image intensities mapped onto the surface.

Affine Ambiguity



Affine Ambiguity



- The decomposition is not unique. We get the same **D** by applying the transformations:

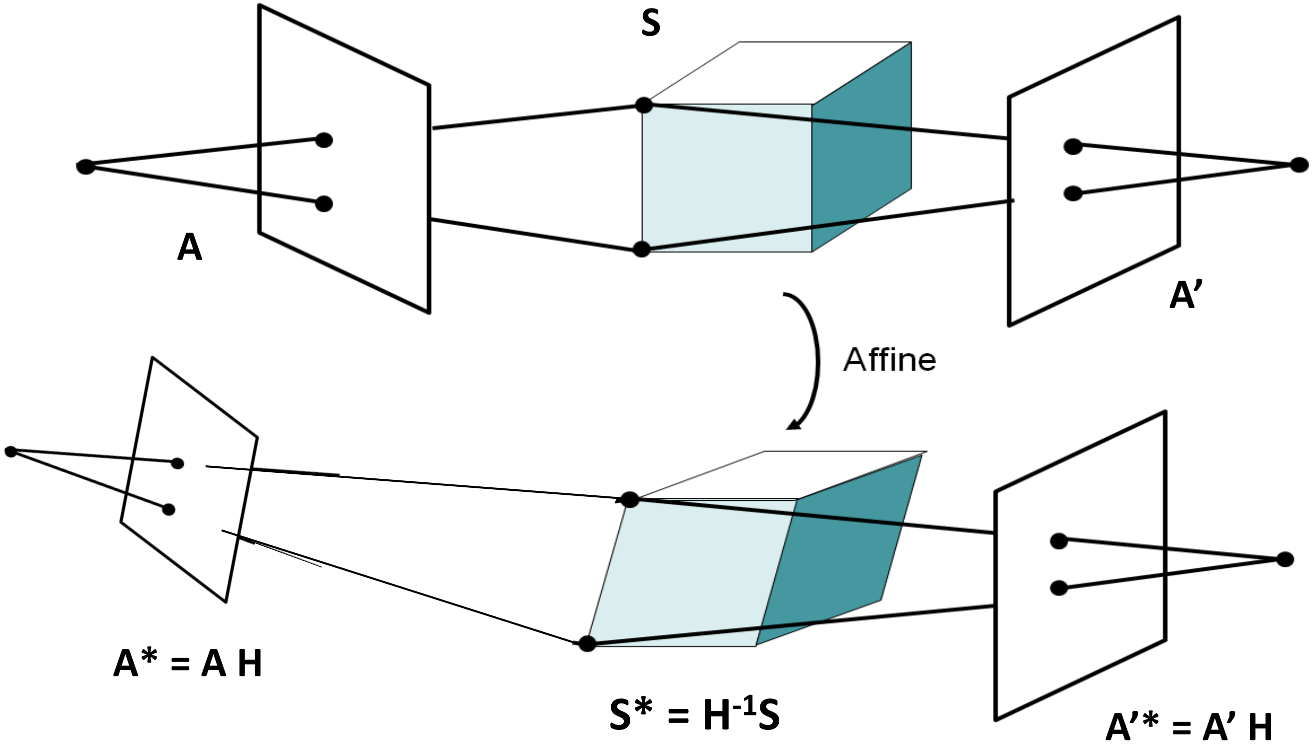
$$\mathbf{M^* = M\,H}$$

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

where **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

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

N. of cameras N. of points

Problem: estimate m matrices A_i, m matrices b_i and the n positions $\mathbf{X}_j$ from the m×n observations $\mathbf{x}_{ij}$.

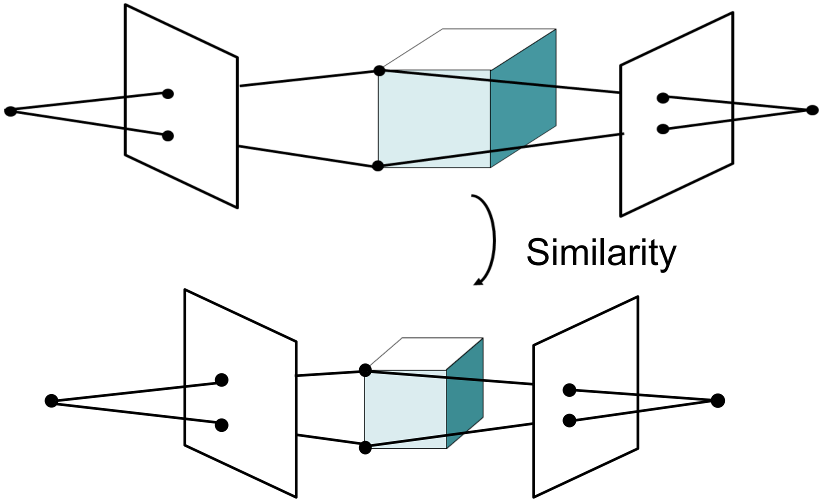
How many equations and how many unknown?

$2m \times n$ 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**



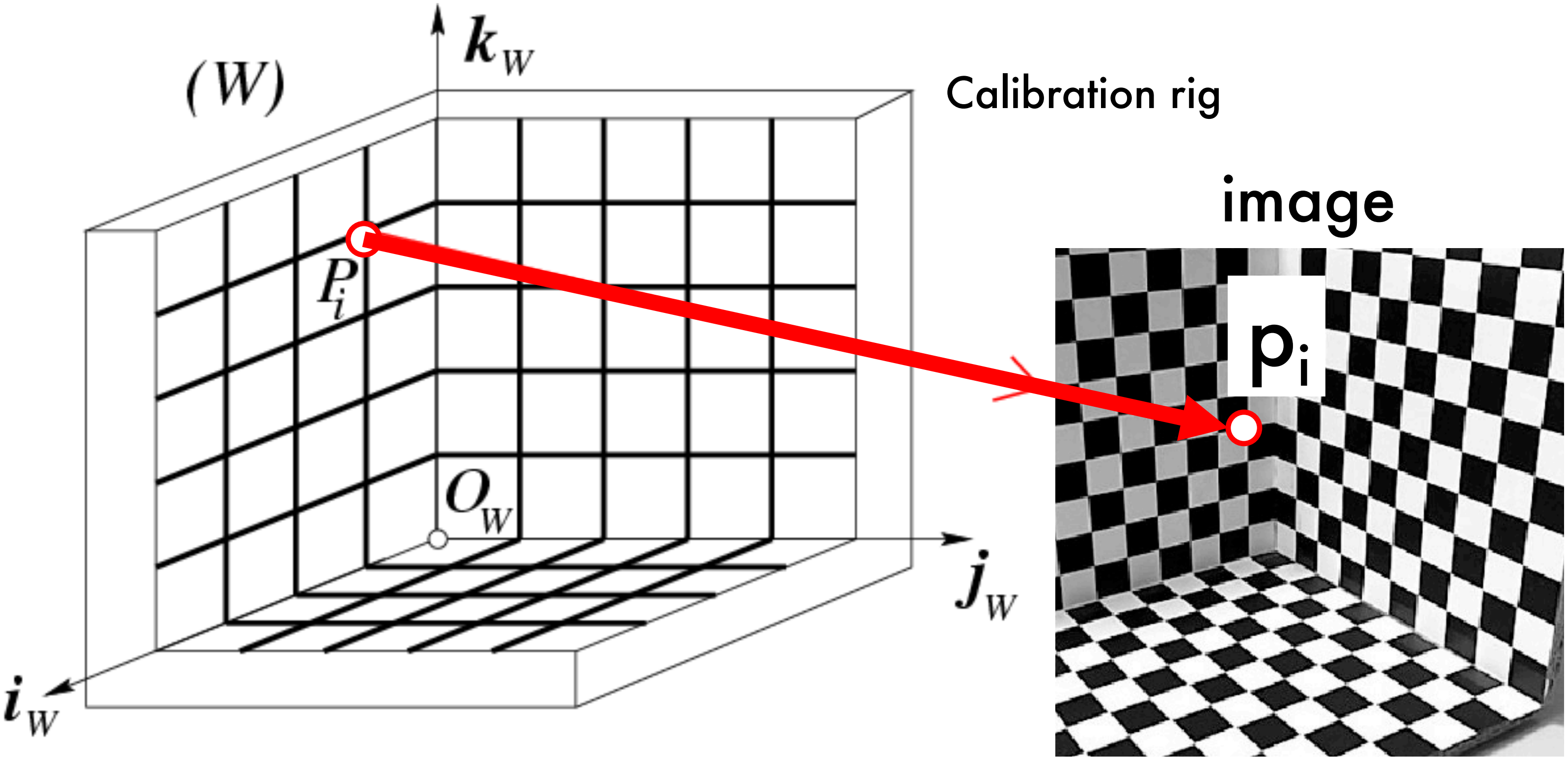
- The ambiguity exists even for (intrinsically) calibrated cameras
- For calibrated cameras, the similarity ambiguity is the only ambiguity

[Longuet-Higgins '81]

Similarity Ambiguity

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

Resolving the similarity ambiguity



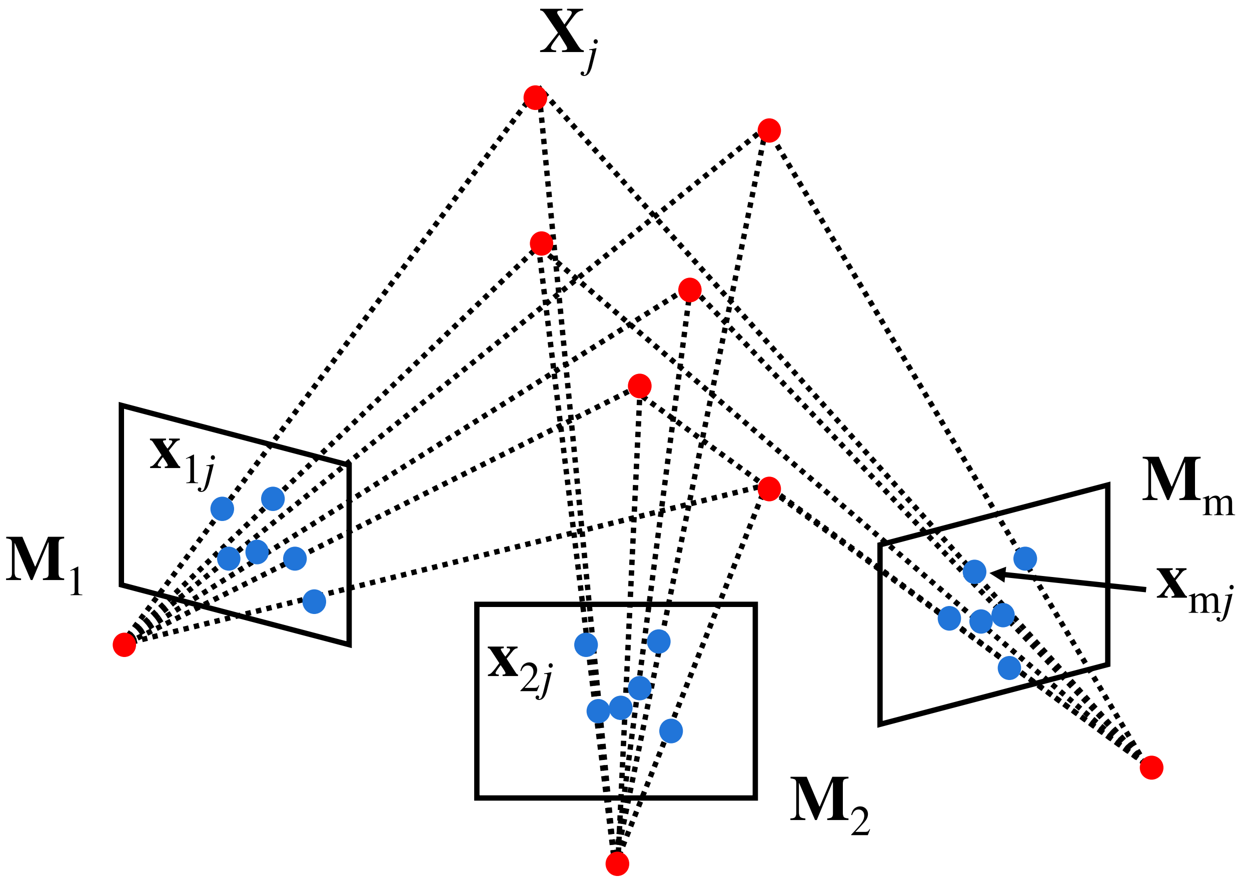
While calibrating a camera, we make assumptions about the geometry of the world

Lecture 7

Multi-view geometry

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

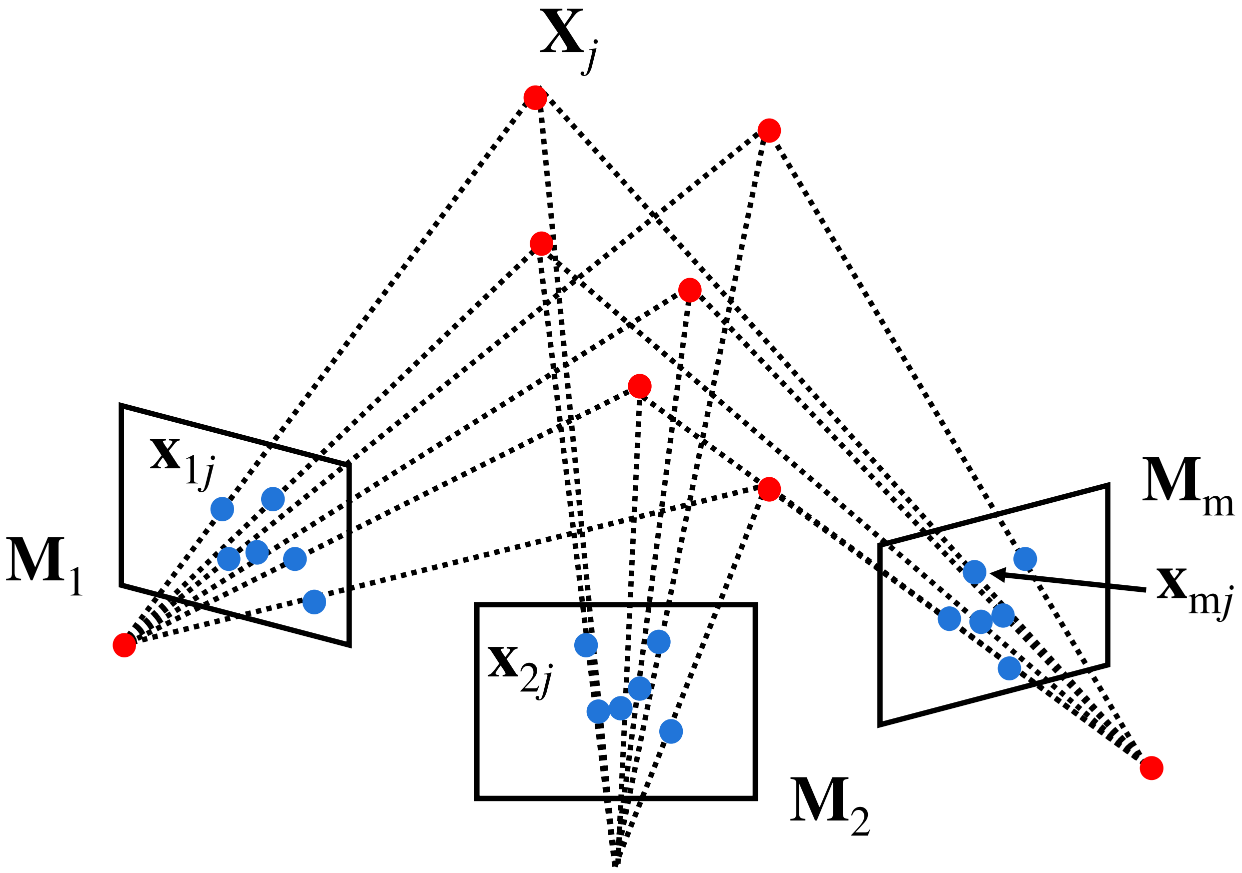
Structure from motion problem



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

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

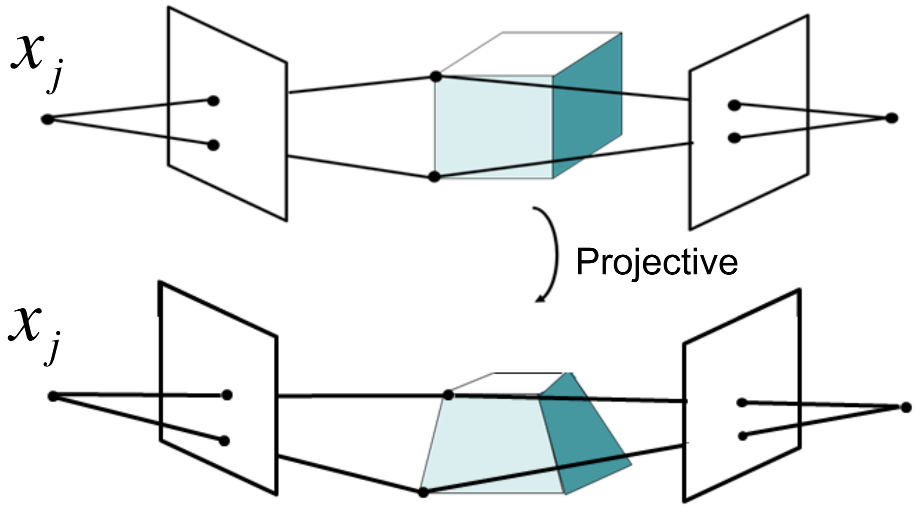
Structure from motion problem



m cameras $M_1 \ldots M_m$

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

Structure from Motion Ambiguities



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

$$\begin{array}{cc} x_j = M_i \boxed{X_j} & \boxed{M_i} = K_i\left[R_i \quad T_i\right] \\ \big\downarrow & \big\downarrow \\ \boxed{H\,X_j} & \boxed{M_j\,H^{-1}} \end{array}$$

$$x_j = M_i\,X_j = \left(M_i\,H^{-1}\right)\left(H\,X_j\right)$$

The Structure-from-Motion Problem

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

$$x_{ij} = M_i \; X_j$$

for i = 1, ...,$\boxed{m}$ and j = 1, ... ,$\boxed{n}$

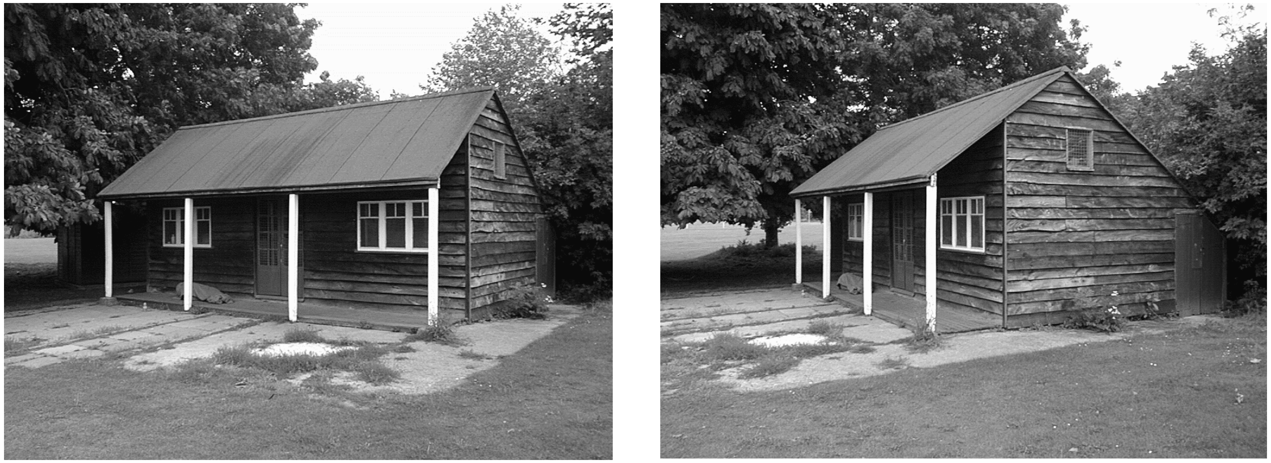
N. of cameras N. of points

Problem: estimate m 3×4 matrices M_i and n positions X_j from m×n obvesrvations x_{ij} .

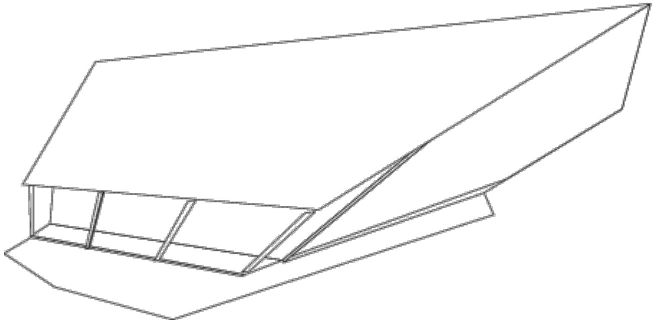
- If the cameras are not calibrated, cameras and points can only be recovered up to a 4x4 projective (where the 4x4 projective is defined up to scale)
- Given two cameras, how many points are needed?
- How many equations and how many unknown?

2m × n equations in 11m+3n – 15 unknowns

Projective Ambiguity



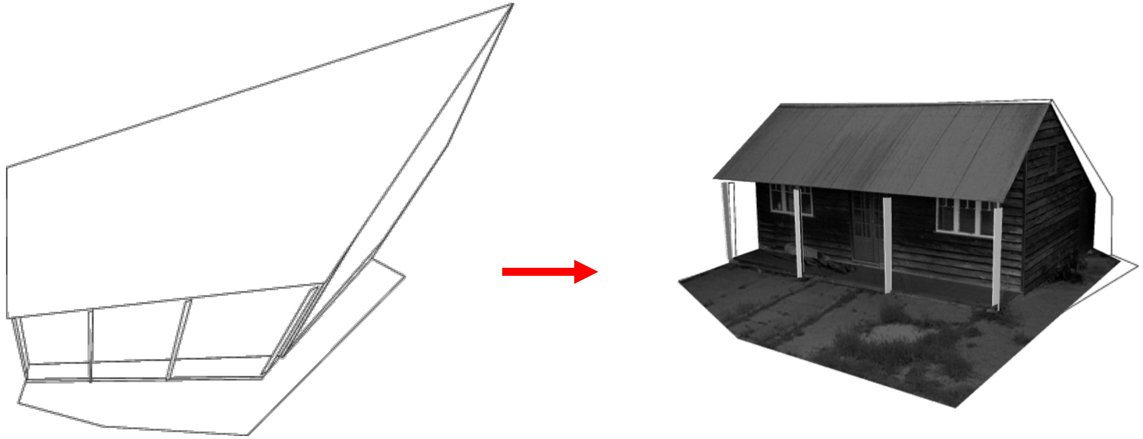
S =



R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, 2003

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

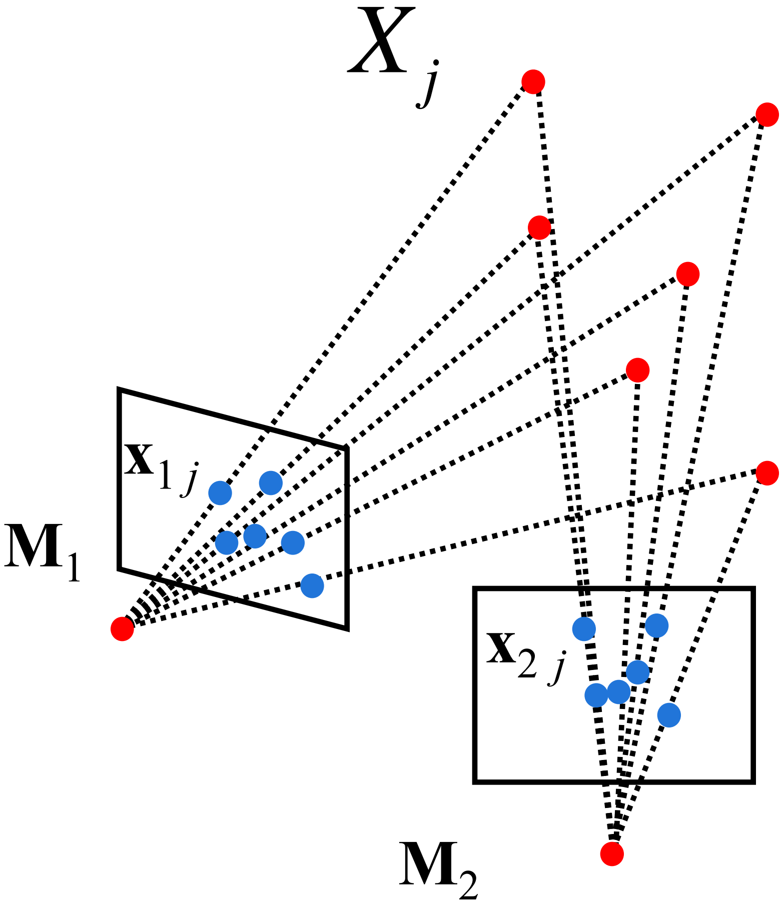
    - Algebraic approach (by fundamental matrix)
    - Factorization method (by SVD)
    - Bundle adjustment

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, … $\boxed{,n}$

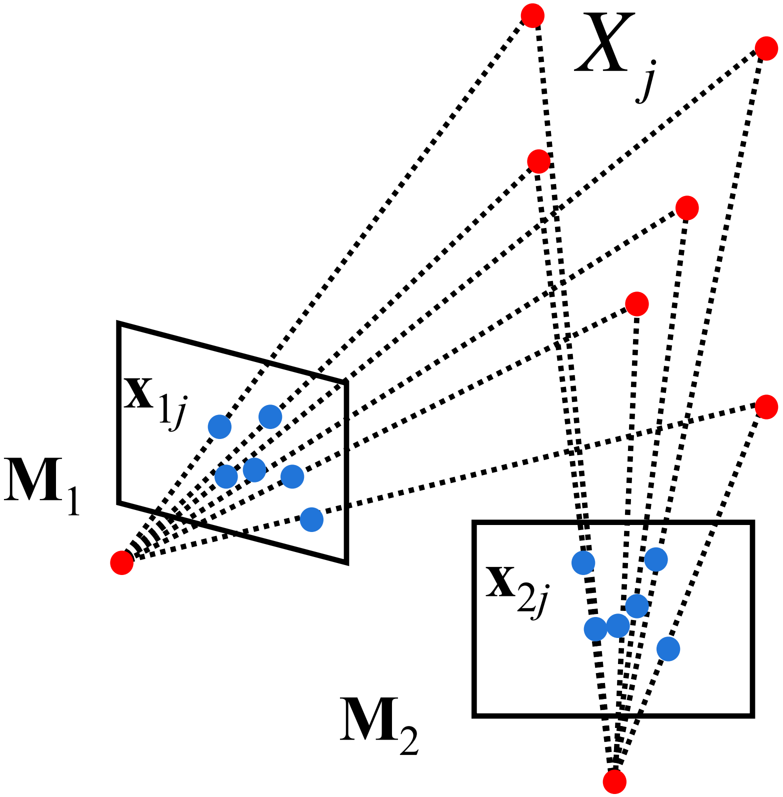
N. of points

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 $j = 1, \ldots, \boxed{,n}$

N. of points

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

$$\mathrm{M}_1 \mathrm{H}^{-1} = \begin{bmatrix} \mathrm{I} & 0 \end{bmatrix}$$

[Eq. 3]

Canonical perspective camera

$$\mathrm{M}_2 \mathrm{H}^{-1} = \begin{bmatrix} \mathrm{A} & \mathrm{b} \end{bmatrix}$$

[Eq. 4]

Algebraic approach (2-view case)

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

[Eqs. 5]

$$\begin{cases} \tilde{M}_1 = M_1\, H^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \\ \tilde{M}_2 = M_2\, H^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \\ \tilde{\mathrm{X}} = \mathrm{H}\, \mathrm{X} \end{cases}$$

$$\mathbf{x} = M_1\, \mathbf{X} = M_1\, H^{-1}\, H\, \mathbf{X} = [\mathbf{I} \,|\, \mathbf{0}]\tilde{\mathbf{X}}$$ [Eq. 6]

$$\mathbf{x}' = M_2\, \mathbf{X} = M_2\, H^{-1}\, H\, \mathbf{X} = [\mathbf{A} | \mathbf{b}]\tilde{\mathbf{X}}$$

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

[Eq. 7]

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

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

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

$$\begin{cases} \tilde{M}_1 = M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \\ \tilde{M}_2 = M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix} \\ \widetilde{\mathrm{X}} = \mathrm{H}\,\mathbf{X} \end{cases}$$

$$\mathbf{x} = M_1 H^{-1} H\,\mathbf{X} = [\mathbf{I}\,|\,\mathbf{0}]\widetilde{\mathbf{X}}$$

$$\mathbf{x}' = M_2 H^{-1} H\,\mathbf{X} = [\mathbf{A}\,|\,\mathbf{b}]\widetilde{\mathbf{X}}$$

[Eq. 6]

$$\vdots$$

$$\mathbf{x}'^{T}(\mathbf{b} \times \mathbf{A}\mathbf{x}) = 0$$

[Eq. 10]

$$\mathbf{x}'^{\mathrm{T}}[\mathbf{b}_{\times}]\mathbf{A}\,\mathbf{x} = 0$$

is this familiar?

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

fundamental matrix!

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

Compute cameras

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

Compute **b**:

- Let's consider the product **F b**

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

- Since **F** is singular, we can compute **b** as least sq. solution of **F b** = 0, with |**b**|=1 using SVD

- Using a similar derivation, we have that $\mathbf{b}^{T}$ **F** = 0 [Eq. 12-bis]

Compute cameras

$$\mathbf{x}'^{\mathrm{T}} \mathrm{F}\, \mathbf{x} = 0 \qquad \mathbf{F} = [\mathbf{b}_\times]\mathbf{A} \qquad \begin{cases} \mathbf{F}\,\mathbf{b} = 0 \\ \mathbf{b}^{\mathrm{T}}\,\mathbf{F} = 0 \end{cases}$$

[Eq. 11]

[Eq. 12]

[Eq. 12-bis]

Compute $\mathbf{A}$:

- Define: $\mathbf{A'} = -[\mathbf{b}_\times]\ \mathbf{F}$

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

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

[Eq. 13]

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

[Eqs. 14]

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

Interpretation of **b**

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

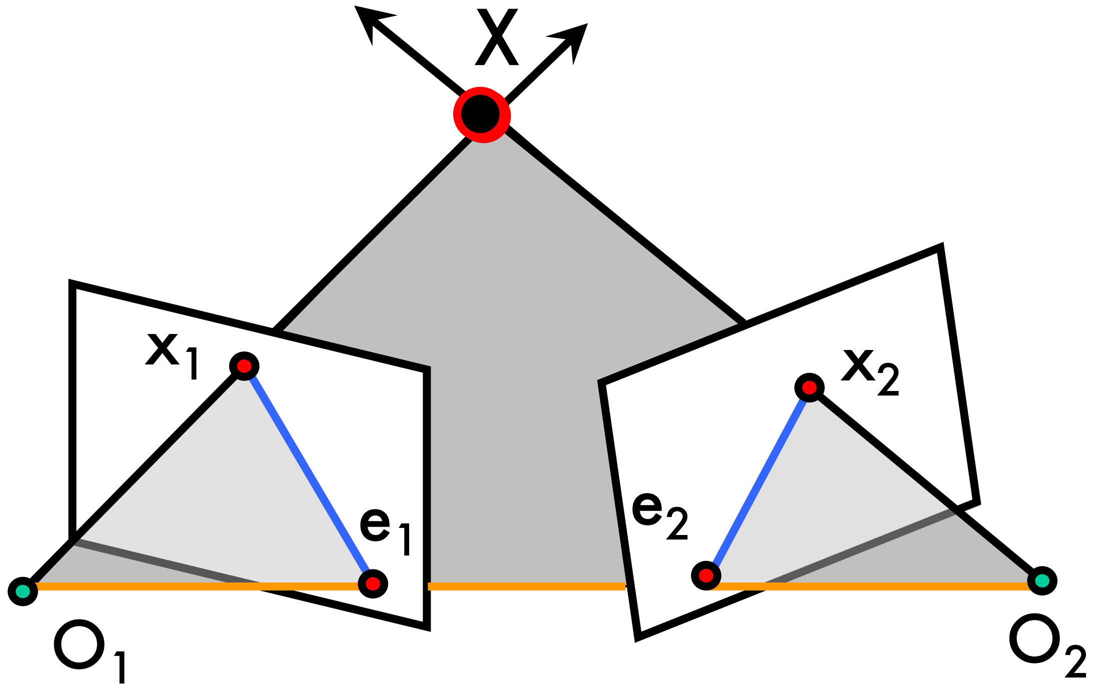
$$\mathbf{F} = [\mathbf{b}_\times]\mathbf{A}$$

[Eq. 11]

$$\begin{cases} \mathbf{F}\,\mathbf{b} = 0 & \text{[Eq. 12]} \\ \mathbf{b}^{\mathrm{T}}\,\mathbf{F} = 0 & \text{[Eq. 12-bis]} \end{cases}$$

What’s **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)

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

F is 3x3 matrix; 7 DOF

Interpretation of **b**

$$\mathbf{x}'^{\mathrm{T}} \mathrm{F}\, \mathbf{x} = 0 \qquad \mathbf{F} = [\mathbf{b}_{\times}]\mathbf{A} \qquad \left\{ \begin{array}{l} \mathbf{F}\,\mathbf{b} = 0 \\ \mathbf{b}^{\mathrm{T}}\,\mathbf{F} = 0 \end{array} \right.$$

[Eq. 11]

b is an epipole!

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

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

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

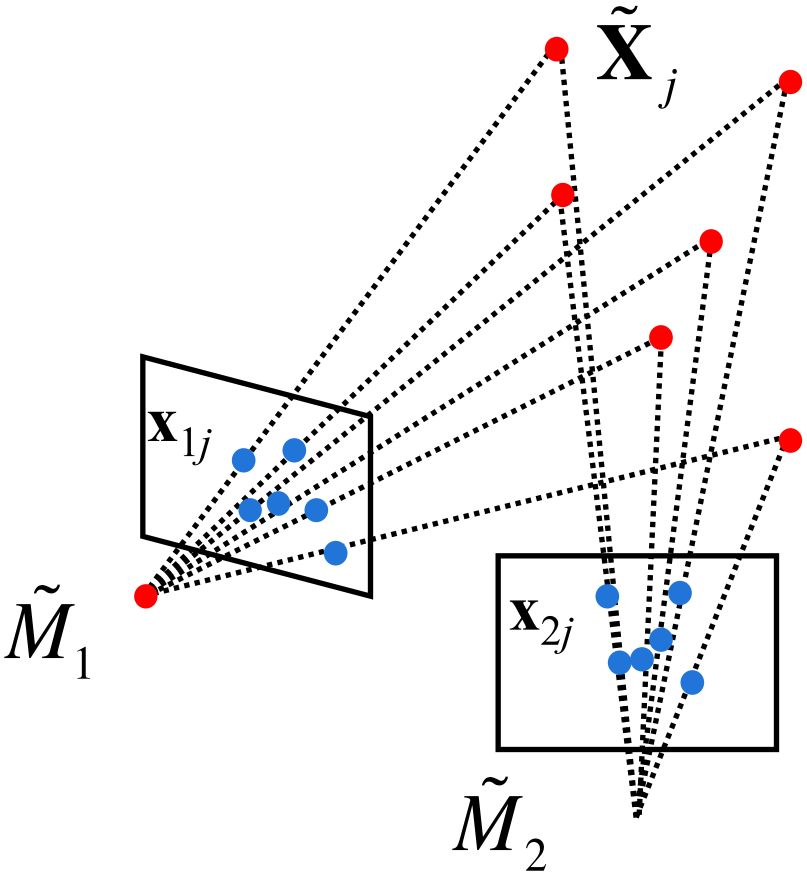
[Eq. 15] [Eq. 16]

HZ, page 254
PF, page 288

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



$$x_{1j} = \tilde{M}_2 \; \tilde{\mathbf{X}}_j$$

$$x_{2j} = \tilde{M}_1 \; \tilde{\mathbf{X}}_j$$

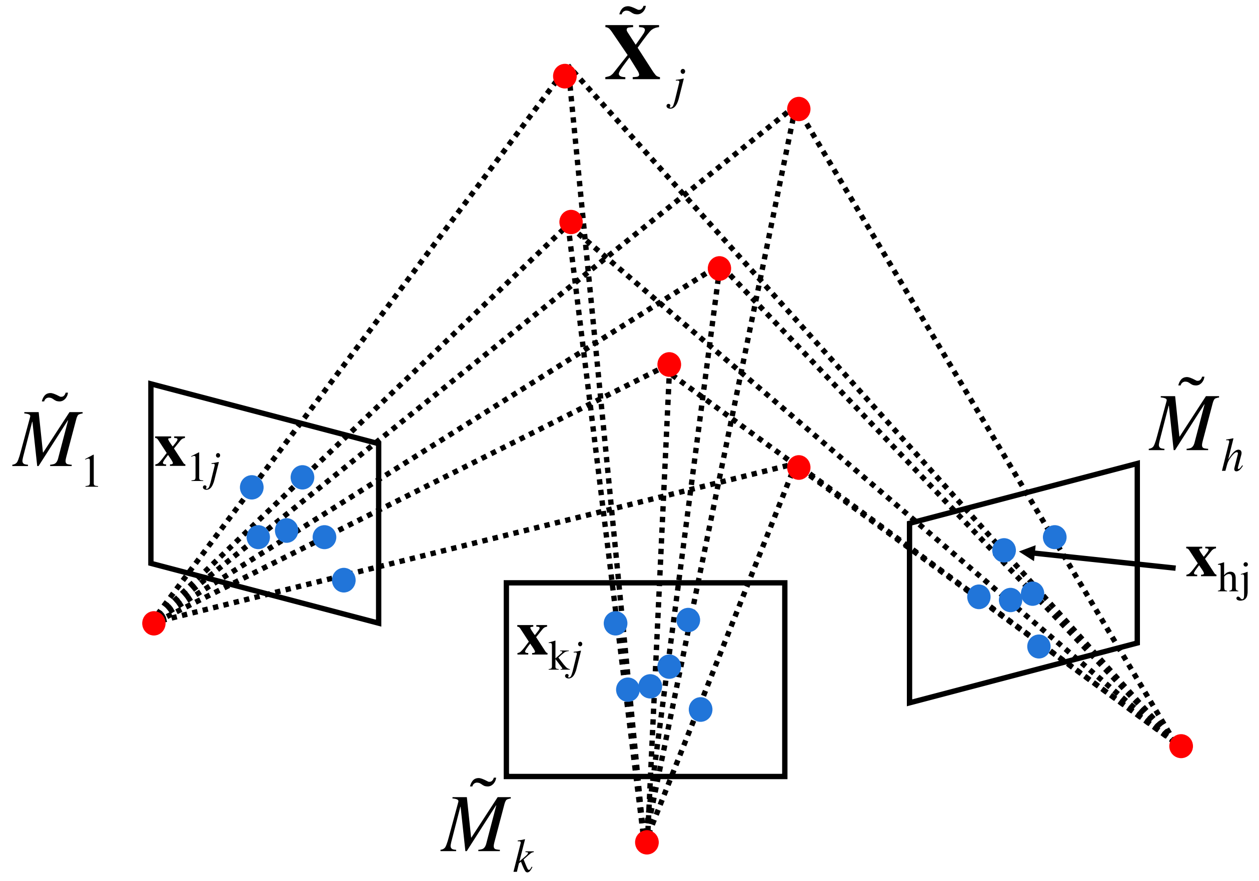
$$\tilde{M}_1 = \begin{bmatrix} I & 0 \end{bmatrix}$$

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

→ $\tilde{\mathbf{X}}_j$ For j = 1, … ,n

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:

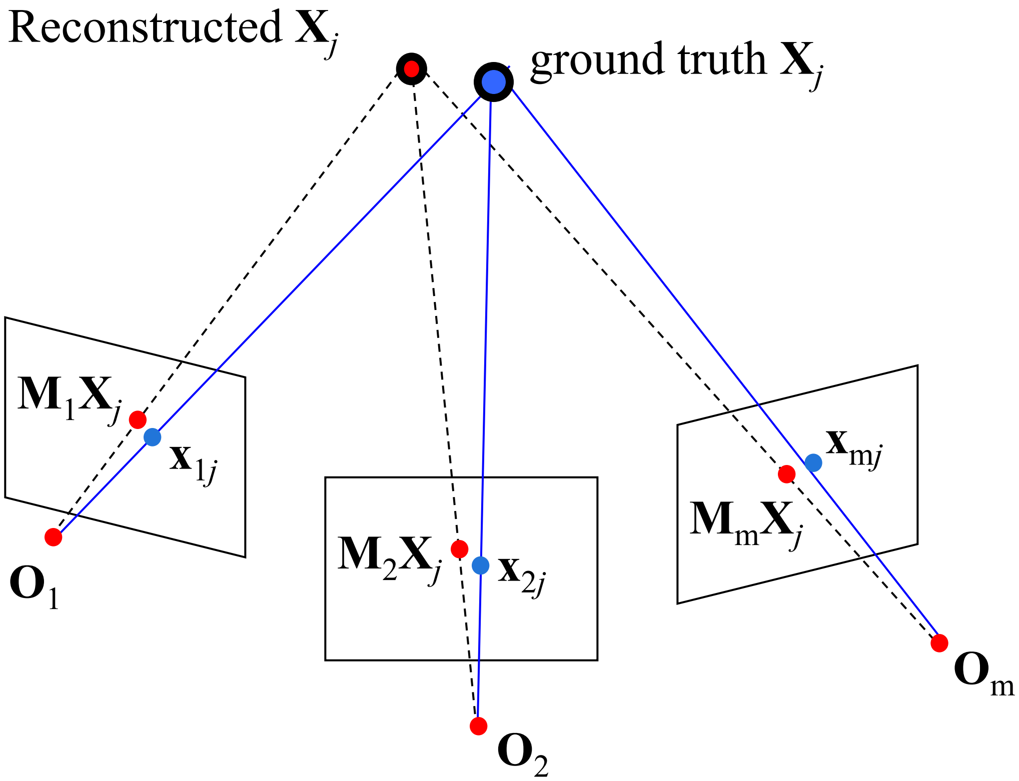
  - occlusions occur
  - failure in establishing correspondences

- Algebraic methods work with 2 views

Bundle adjustment

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

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



Reconstructed $\mathbf{X}_j$
ground truth $\mathbf{X}_j$
$\mathbf{M}_1\mathbf{X}_j$
$\mathbf{x}_{1j}$
$\mathbf{M}_2\mathbf{X}_j$
$\mathbf{x}_{2j}$
$\mathbf{x}_{mj}$
$\mathbf{M}_m\mathbf{X}_j$
$\mathbf{O}_1$
$\mathbf{O}_2$
$\mathbf{O}_m$

General Calibration Problem

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

parameters

measurements

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

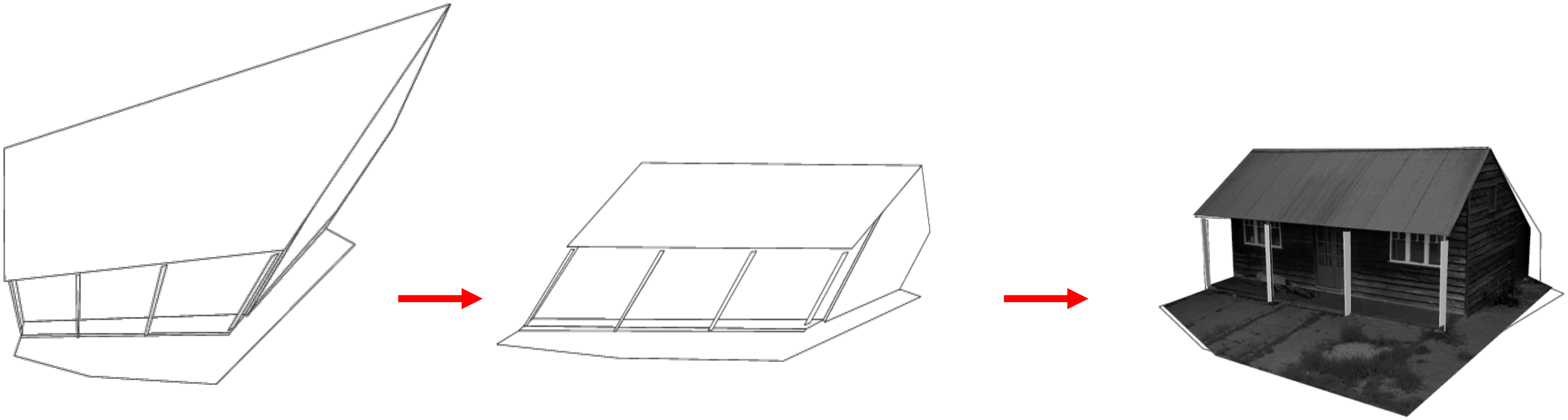
Lecture 7

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



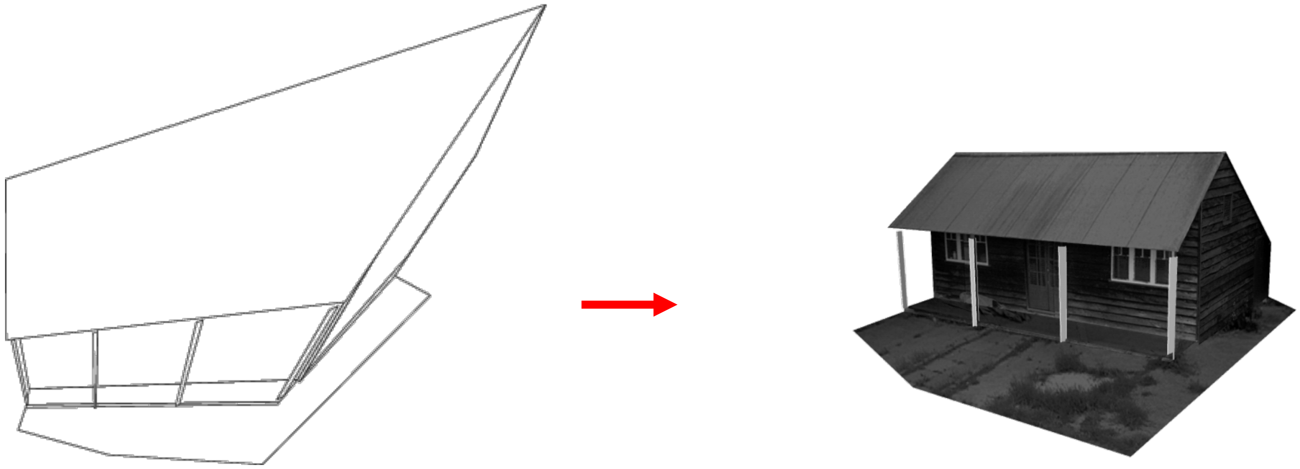
Self-calibration

[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



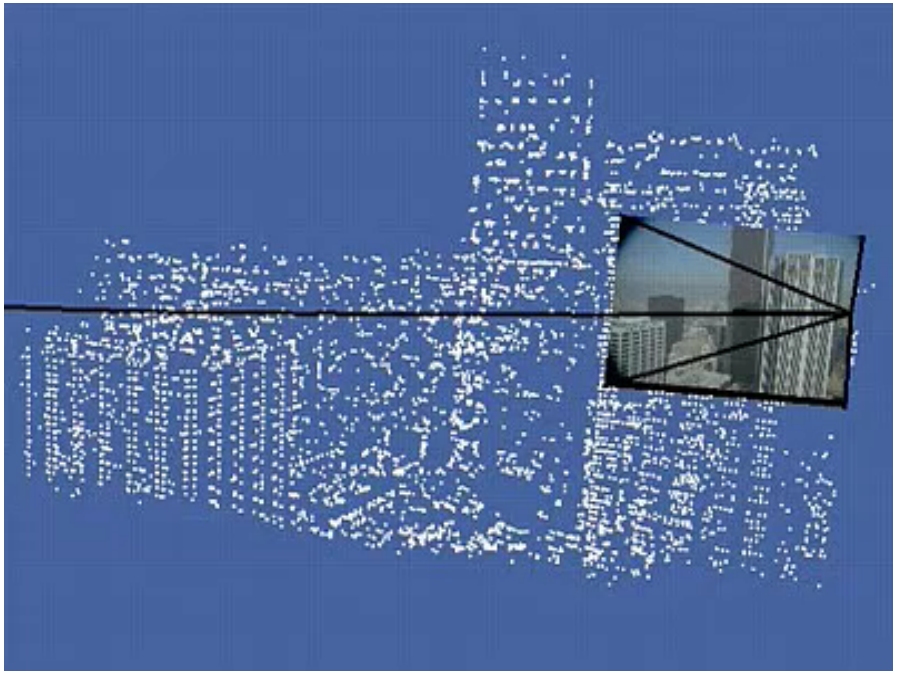
For calibrated cameras, the similarity ambiguity is the only ambiguity [Longuet-Higgins '81]

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**

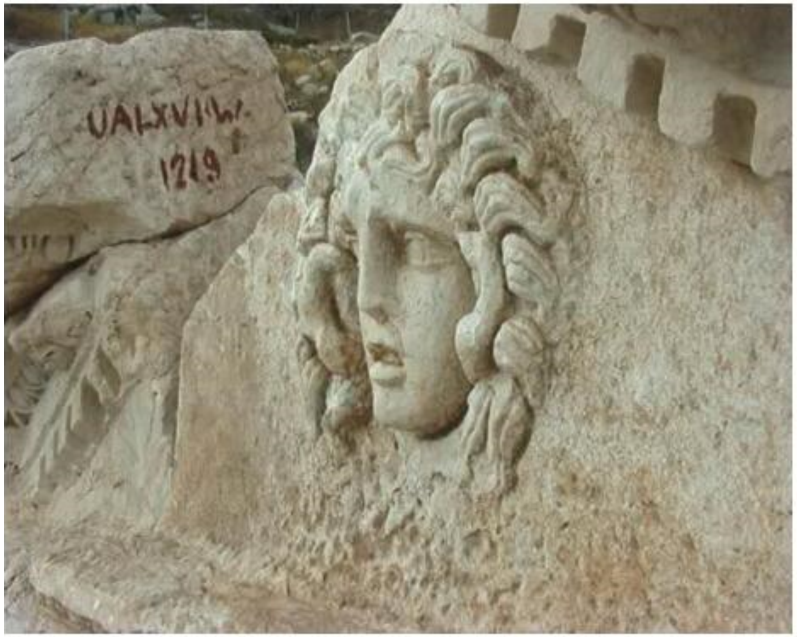
Lucas & Kanade, 81
Chen & Medioni, 92
Debevec et al., 96
Levoy & Hanrahan, 96
Fitzgibbon & Zisserman, 98
Triggs et al., 99
Pollefeys et al., 99
Kutulakos & Seitz, 99

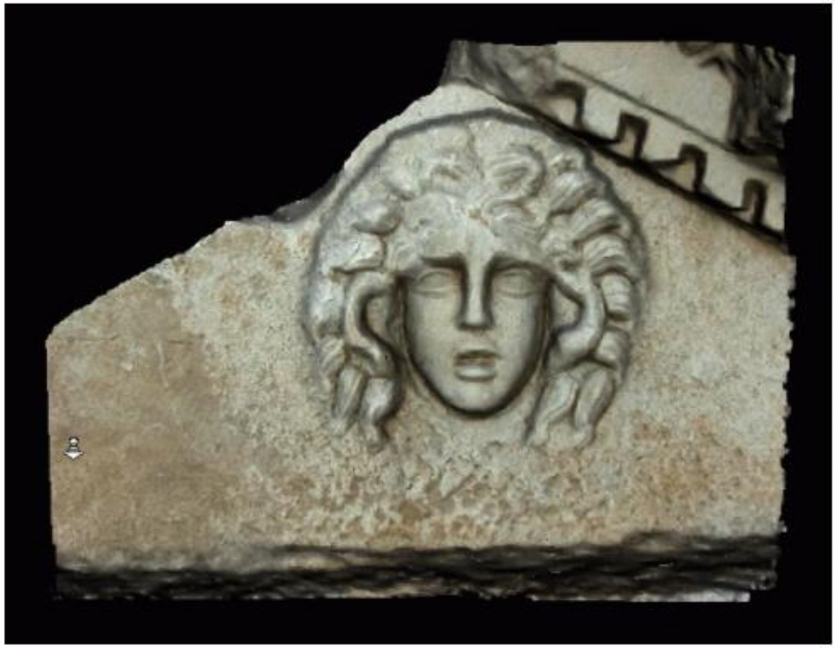
Levoy et al., 00
Hartley & Zisserman, 00
Dellaert et al., 00
Rusinkiewic et al., 02
Nistér, 04
Brown & Lowe, 04
Schindler et al, 04
Lourakis & Argyros, 04
Colombo et al. 05

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

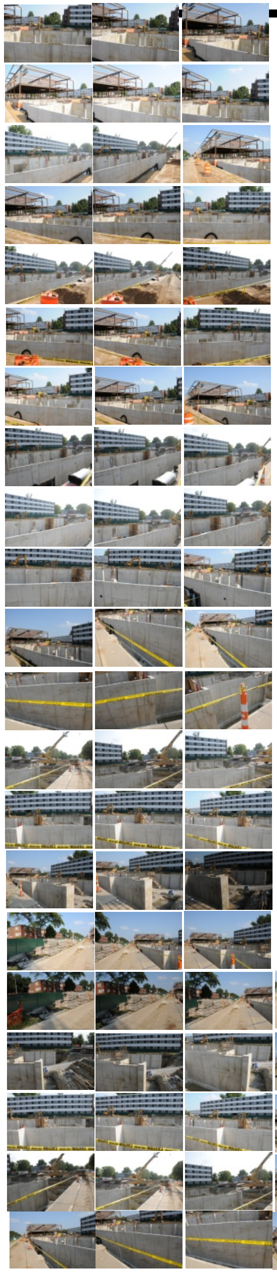
M. Pollefeys et al 98–

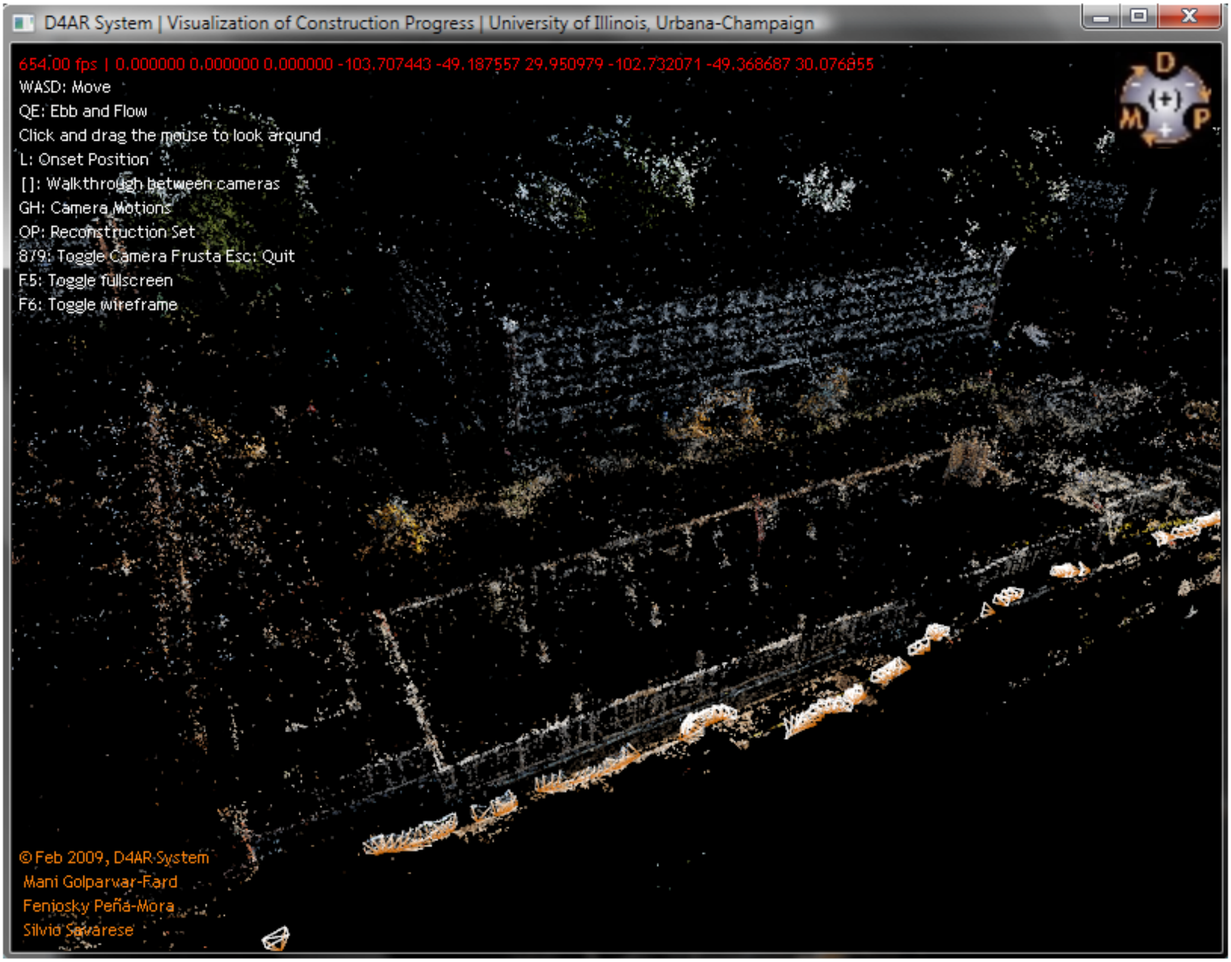




Incremental reconstruction of construction sites

Initial pair – 2168 & Complete Set 62,323 points, 160 images Golparvar-Fard. Pena-Mora, Savarese 2008

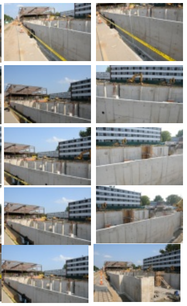


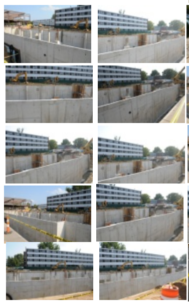


D4AR System | Visualization of Construction Progress | University of Illinois, Urbana-Champaign

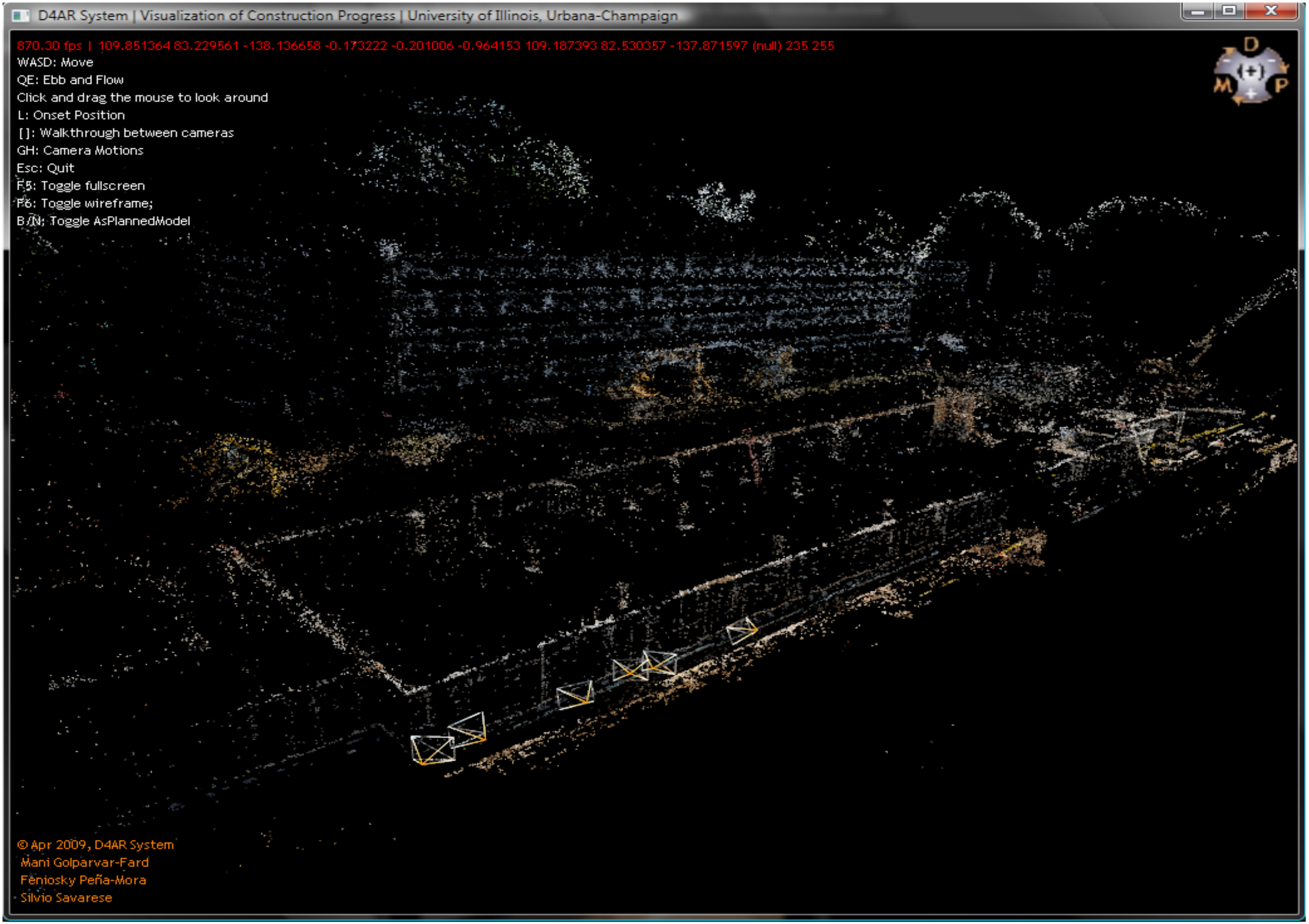
```
654.00 fps | 0.000000 0.000000 0.000000 -103.707443 -49.187557 29.950979 -102.732071 -49.368687 30.076855
WASD: Move
QE: Ebb and Flow
Click and drag the mouse to look around
L: Onset Position
[]: Walkthrough between cameras
GH: Camera Motions
OP: Reconstruction Set
879: Toggle Camera Frusta Esc: Quit
F5: Toggle fullscreen
F6: Toggle wireframe
```

©Feb 2009, D4AR System
Mani Golparvar-Fard
Feniosky Peña-Mora
Silvio Savarese





Reconstructed scene + Site photos

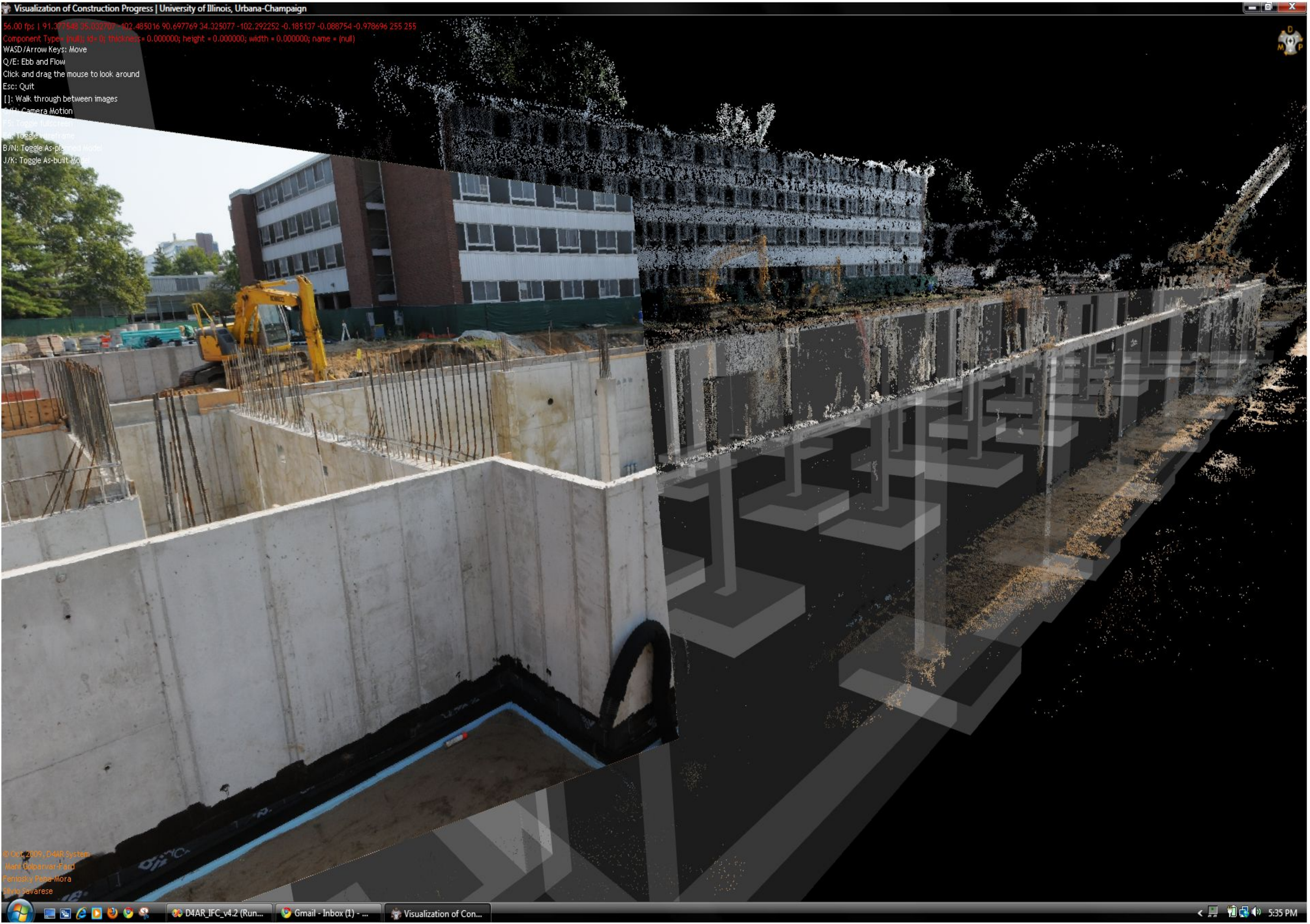


D4AR System | Visualization of Construction Progress | University of Illinois, Urbana-Champaign

```
870.30 fps | 109.851364 83.229561 -138.136658 -0.173222 -0.201006 -0.964153 109.187393 82.530357 -137.871597 (null) 235 255
WASD: Move
QE: Ebb and Flow
Click and drag the mouse to look around
L: Onset Position
[]: Walkthrough between cameras
GH: Camera Motions
Esc: Quit
F5: Toggle fullscreen
F6: Toggle wireframe;
B/N: Toggle AsPlannedModel
```

© Apr 2009, D4AR System
Mani Golparvar-Fard
Feniosky Peña-Mora
Silvio Savarese

Reconstructed scene + Site photos



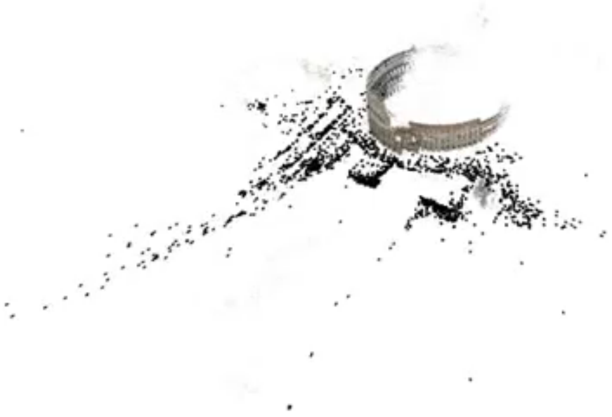
Visualization of Construction Progress | University of Illinois, Urbana-Champaign

```
56.00 fps | 91.377548 35.032707 -102.485016 90.697769 34.325077 -102.292252 -0.185137 -0.088754 -0.978696 255 255
Component Type= (null); id= 0; thickness= 0.000000; height = 0.000000; width = 0.000000; name = (null)
WASD/Arrow Keys: Move
Q/E: Ebb and Flow
Click and drag the mouse to look around
Esc: Quit
[]: Walk through between images
G/H: Camera Motion
F5: Toggle fullscreen
Eo: Toggle wireframe
B/N: Toggle As-planned Model
J/K: Toggle As-built Model
```

© Oct 2009, D4AR System
Mani Golparvar-Fard
Feniosky Pena-Mora
Silvio Savarese

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

- Fitting and Matching

Appendix

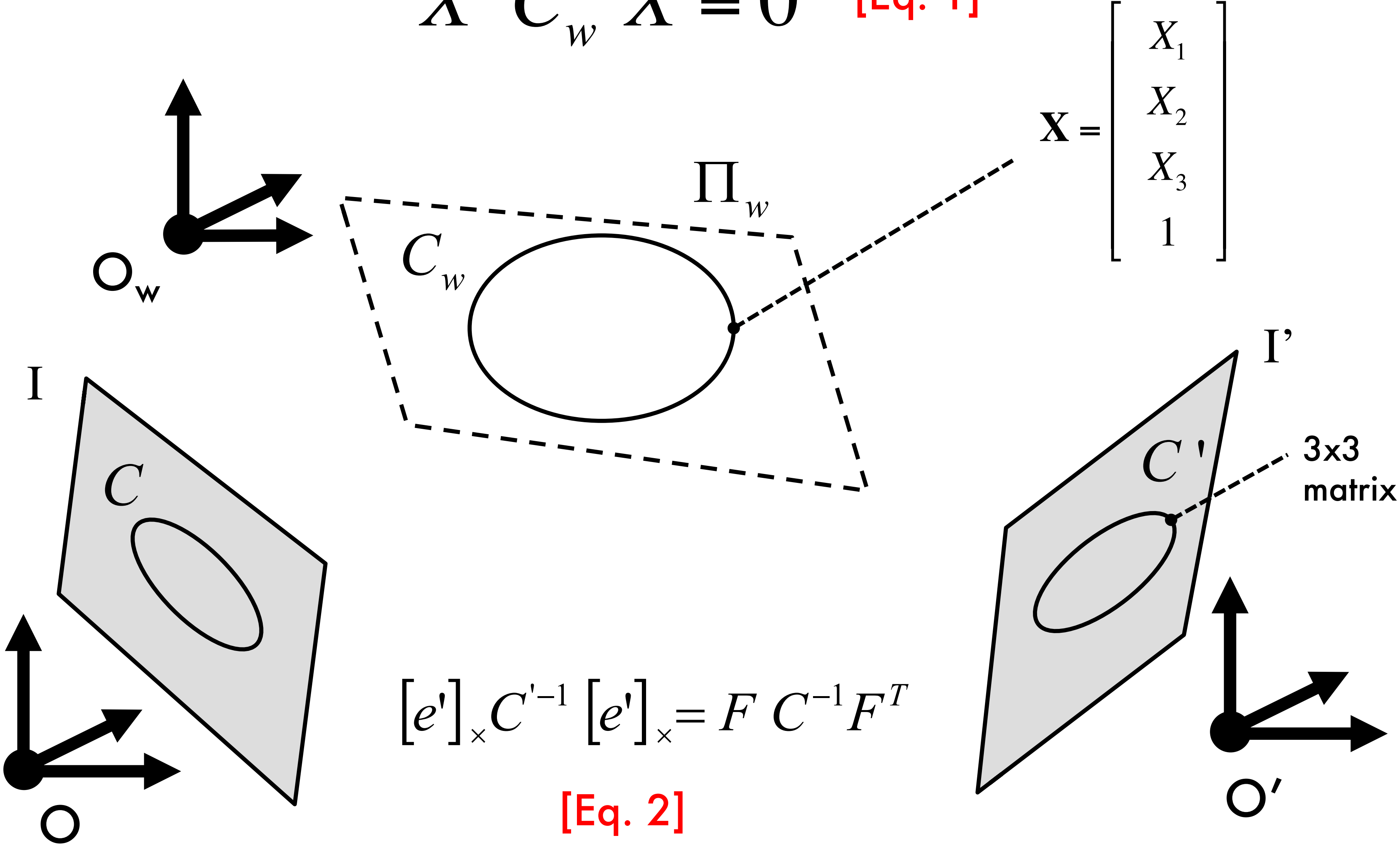
Direct approach

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

$$X^T C_w \; X = 0$$ [Eq. 1]



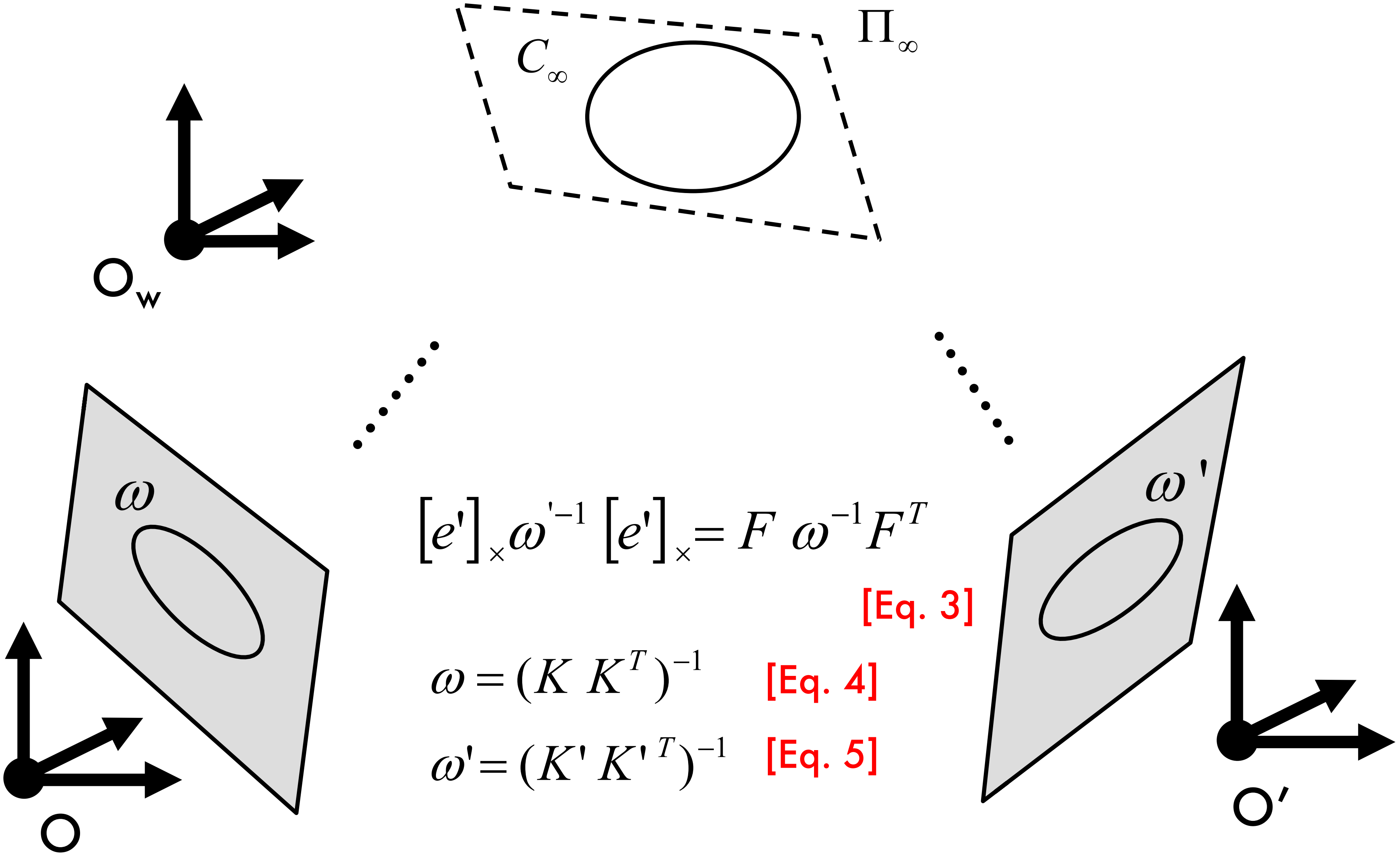
$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$

$$[e']_\times C'^{-1} \; [e']_\times = F \; C^{-1} F^T$$

[Eq. 2]

Projection of absolute conics across views

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



$$[e']_{\times}\,\omega'^{-1}\,[e']_{\times} = F\,\omega^{-1}F^{T}$$

[Eq. 3]

$$\omega = (K\,K^{T})^{-1}$$ [Eq. 4]

$$\omega' = (K'\,K'^{T})^{-1}$$ [Eq. 5]

Kruppa equations

[Faugeras et al. 92]

From [HZ] page 471

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

[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]

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

$$\frac{u_2^T K K^T u_2}{\sigma_1^2 v_1^T K K^T v_1} = \frac{-u_1^T K K^T u_2}{\sigma_1 \sigma_2 v_1^T K K^T v_2} = \frac{u_1^T K K^T u_1}{\sigma_2^2 v_2^T K K^T v_2}$$ [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]

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

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

Auto Calibration

- Auto-calibration is the process of determining internal camera parameters directly from multiple uncalibrated images.
- Once this is done, it is possible to compute a metric reconstruction from the images.
- Auto-calibration avoids the onerous task of calibrating cameras using special calibration objects.
- This gives great flexibility since, for example, a camera can be calibrated directly from an image sequence despite unknown motion and changes in some of the internal parameters.

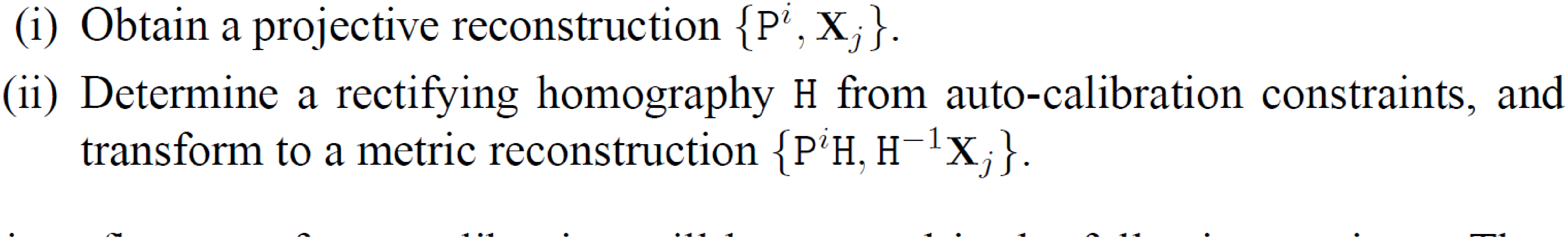
Algebraic Frame work for Auto-calibration

- Suppose we have a set of images acquired by a camera with fixed internal parameters, and that a projective reconstruction is computed from point correspondences across the image set.
- The reconstruction computes a projective camera matrix P*i* for each view. Our constraint is that for the actual cameras the internal parameter matrix K is the same (but unknown) for each view.
- Now, each camera P*i* of the projective reconstruction may be decomposed as P*i* = K*i*[R*i* / **t***i*] but in general the calibration matrix K*i* will differ for each view.
- Thus the constraint will *not* be satisfied by the projective reconstruction.

Algebraic Framework

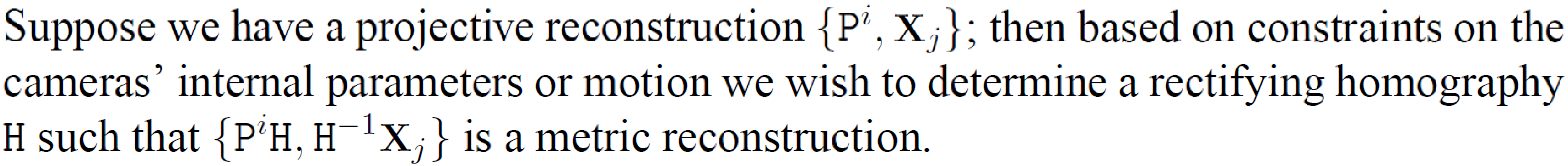
- However, we have the freedom to vary our projective reconstruction by transforming the camera matrices by a homography H.
- Since the actual cameras have fixed internal parameters, there will exist a homography (or a family of homographies) such that the transformed cameras P*i*H do decompose as P*i*H = KR*i*[I / **t***i*], with the same calibration matrix for each camera, so the reconstruction is consistent with the constraint.
- Provided there are sufficiently many views and the motion between the views is general, then this consistency constrains H to the extent that the reconstruction transformed by H is within a similarity transformation of the actual cameras and scene, i.e. we achieve a metric reconstruction.

General approach



(i) Obtain a projective reconstruction $\{\mathtt{P}^i, \mathbf{X}_j\}$.

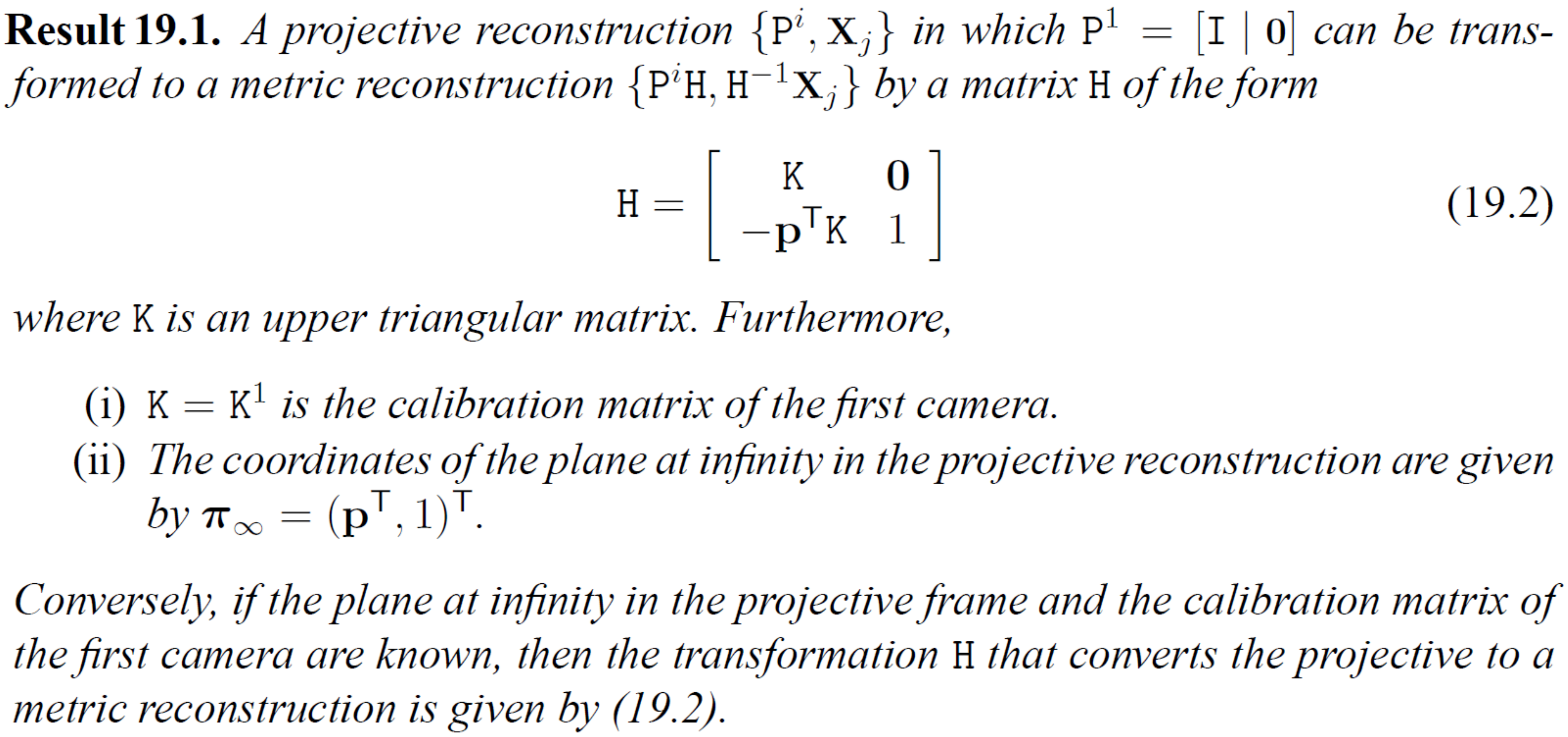
(ii) Determine a rectifying homography $\mathtt{H}$ from auto-calibration constraints, and transform to a metric reconstruction $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$.



Suppose we have a projective reconstruction $\{\mathtt{P}^i, \mathbf{X}_j\}$; then based on constraints on the cameras' internal parameters or motion we wish to determine a rectifying homography $\mathtt{H}$ such that $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$ is a metric reconstruction.

Our goal is to find H

Result



Result 19.1. *A projective reconstruction* $\{\mathtt{P}^i, \mathbf{X}_j\}$ *in which* $\mathtt{P}^1 = [\mathtt{I} \mid \mathbf{0}]$ *can be transformed to a metric reconstruction* $\{\mathtt{P}^i\mathtt{H}, \mathtt{H}^{-1}\mathbf{X}_j\}$ *by a matrix* $\mathtt{H}$ *of the form*

$$\mathtt{H} = \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ -\mathbf{p}^\mathsf{T}\mathtt{K} & 1 \end{bmatrix} \tag{19.2}$$

where $\mathtt{K}$ *is an upper triangular matrix. Furthermore,*

(i) $\mathtt{K} = \mathtt{K}^1$ *is the calibration matrix of the first camera.*
(ii) *The coordinates of the plane at infinity in the projective reconstruction are given by* $\boldsymbol{\pi}_\infty = (\mathbf{p}^\mathsf{T}, 1)^\mathsf{T}$.

Conversely, if the plane at infinity in the projective frame and the calibration matrix of the first camera are known, then the transformation $\mathtt{H}$ *that converts the projective to a metric reconstruction is given by (19.2).*

Suppose that all the cameras have the same internal parameters, so $\mathtt{K}^i = \mathtt{K}$, then (19.4) becomes

$$\mathtt{K}\mathtt{K}^{\mathsf{T}} = \left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^{\mathsf{T}}\right)\mathtt{K}\mathtt{K}^{\mathsf{T}}\left(\mathtt{A}^i - \mathbf{a}^i\mathbf{p}^{\mathsf{T}}\right)^{\mathsf{T}} \quad i = 2, \ldots, m. \tag{19.5}$$

Each view $i = 2, \ldots, m$ provides an equation, and we can develop a counting argument for the number of views required (in principle) in order to be able to determine the 8 unknowns. Each view other than the first imposes 5 constraints since each side is a 3×3 symmetric matrix (i.e. 6 independent elements) and the equation is homogeneous. Assuming these constraints are independent for each view, a solution is determined provided $5(m-1) \geq 8$. Consequently, provided $m \geq 3$ a solution is obtained, at least in principle. Clearly, if m is much larger than 3 the unknowns $\mathtt{K}$ and $\mathbf{p}$ are very over-determined. △

Algebraic approach Multi-view approach

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

Let H be a homography such that:

$$\begin{cases} \text{First perspective camera is canonical: } \tilde{M}_1 = [\;I \quad 0\;] \text{ [Eq. 11]} \\ i^{th} \text{ perspective reconstruction of the camera (known): } \tilde{\mathrm{M}}_{\mathrm{i}} = [\;\mathrm{A}_{\mathrm{i}} \quad \mathrm{b}_{\mathrm{i}}\;] \end{cases}$$

[Eq. 12]

[Eq. 13] $$\left(A_i - b_i p^T\right) K_1\, K_1^T \left(A_i - b_i p^T\right)^T = K_i\;\; K_i^T \qquad \text{i=2...m}$$

[Eq. 14] $$H = \begin{bmatrix} K_1 & 0 \\ -p^T\, K_1 & 1 \end{bmatrix}$$

p is an unknown 3x1 vector

$K_1 \ldots K_m$ are unknown

Algebraic approach Multi-view approach

Suppose we have a projective reconstruction

Let H be a homography such that:

$\begin{cases} \text{First perspective camera is canonical: } \tilde{M}_1 = [\; I \quad 0 \;] \\ i^{th} \text{ perspective reconstruction of the camera (known): } \tilde{\mathrm{M}}_{\mathrm{i}} = [\; \mathrm{A}_{\mathrm{i}} \quad \mathrm{b}_{\mathrm{i}} \;] \end{cases}$ [Eq. 11] [Eq. 12]

[Eq. 13] $$\left(A_i - b_i p^T\right) K_1 \, K_1^T \left(A_i - b_i p^T\right)^T = K_i \;\; K_i^T$$ i=2…m

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

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:

$$\begin{cases} \text{First perspective camera is canonical: } \tilde{M}_1 = [\ I \quad 0\ ] \text{ [Eq. 11]} \\ \text{i}^{\text{th}} \text{ perspective reconstruction of the camera (known): } \tilde{\mathrm{M}}_\mathrm{i} = [\ \mathrm{A}_\mathrm{i} \quad \mathrm{b}_\mathrm{i}\ ] \end{cases}$$

[Eq. 12]

Assume all camera matrices are identical: $\mathrm{K}_1 = \mathrm{K}_2 \dots = \mathrm{K}_\mathrm{m}$

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

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

Condition	*N. Views*
• 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