EECS 442 – Computer vision

Multiple view geometry
Affine structure from Motion

- Affine structure from motion problem
- Algebraic methods
- Factorization methods

Reading: [HZ] Chapters: 6,14,18
[FP] Chapter: 12

Some slides of this lectures are courtesy of prof. J. Ponce, prof FF Li, prof S. Lazebnik & prof. M. Hebert

Applications

Courtesy of Oxford **Visual Geometry Group**

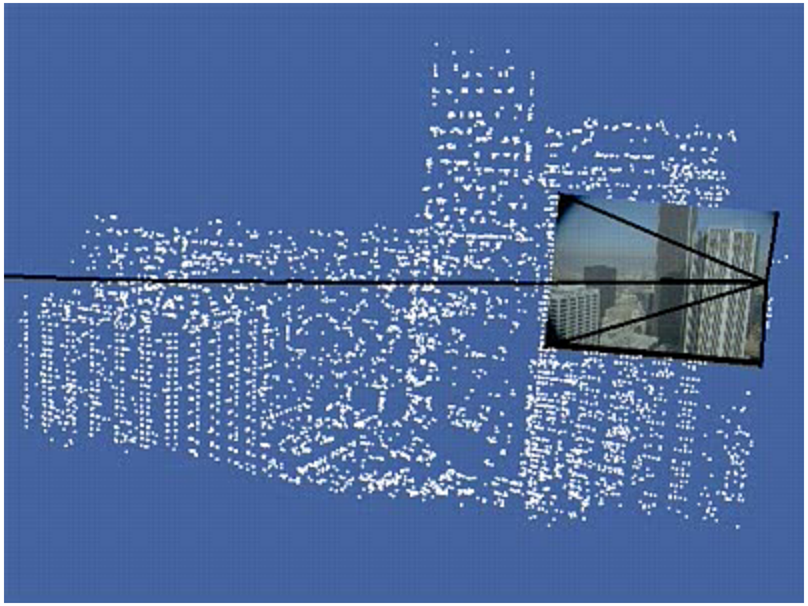




photo by Ceci And Brandon



photo by bryantranbarger



photo by ahobbeslife



photo by john_knowles

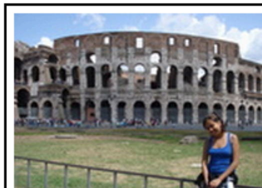


photo by rossemar



photo by Suit Boy

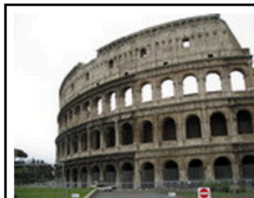


photo by The Rantz's

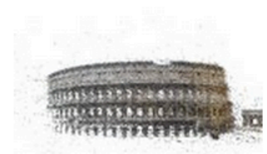



photo by angrypixel



photo by franje123



photo by johnboy101

Photo Tourism

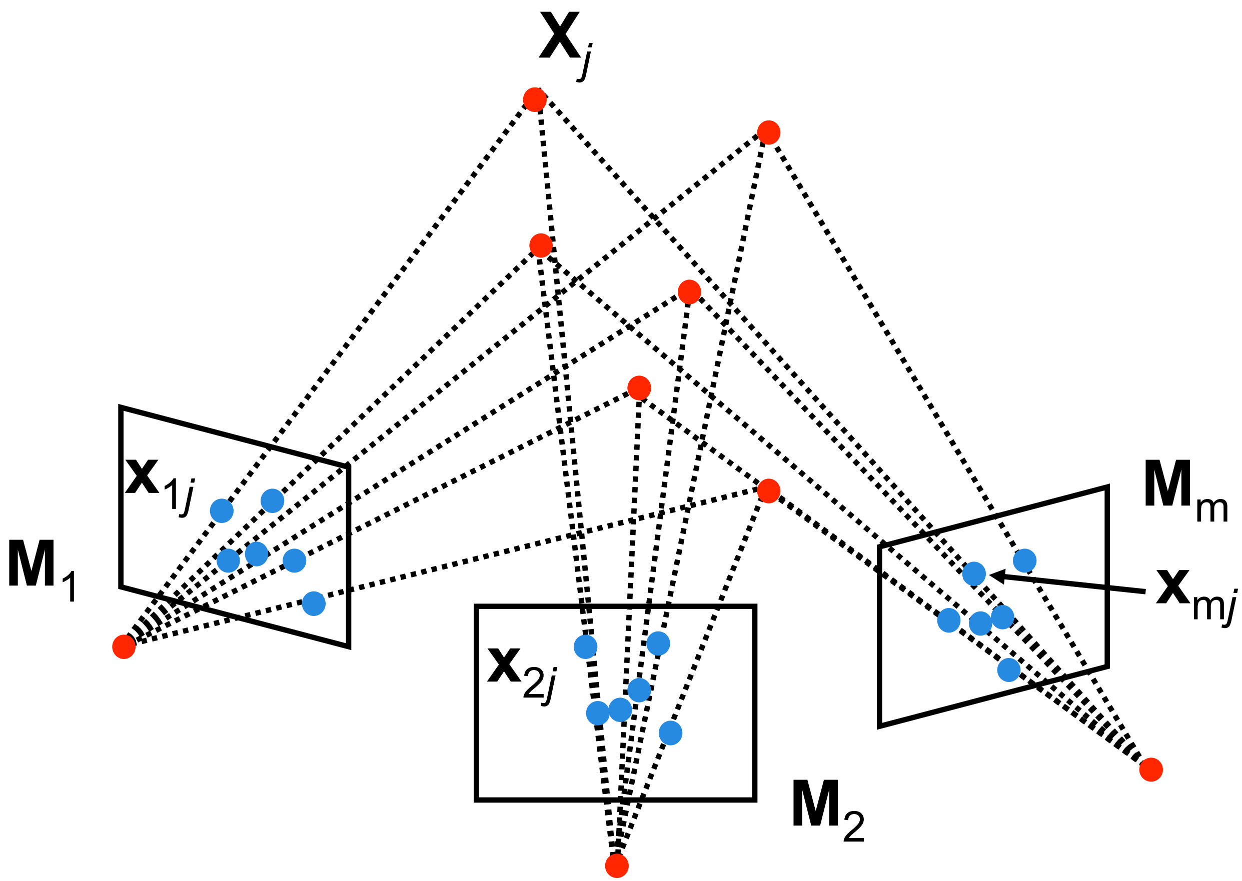
Exploring photo collections in 3D

Noah Snavely   Steven M. Seitz   Richard Szeliski

University of Washington   *Microsoft Research*

SIGGRAPH 2006

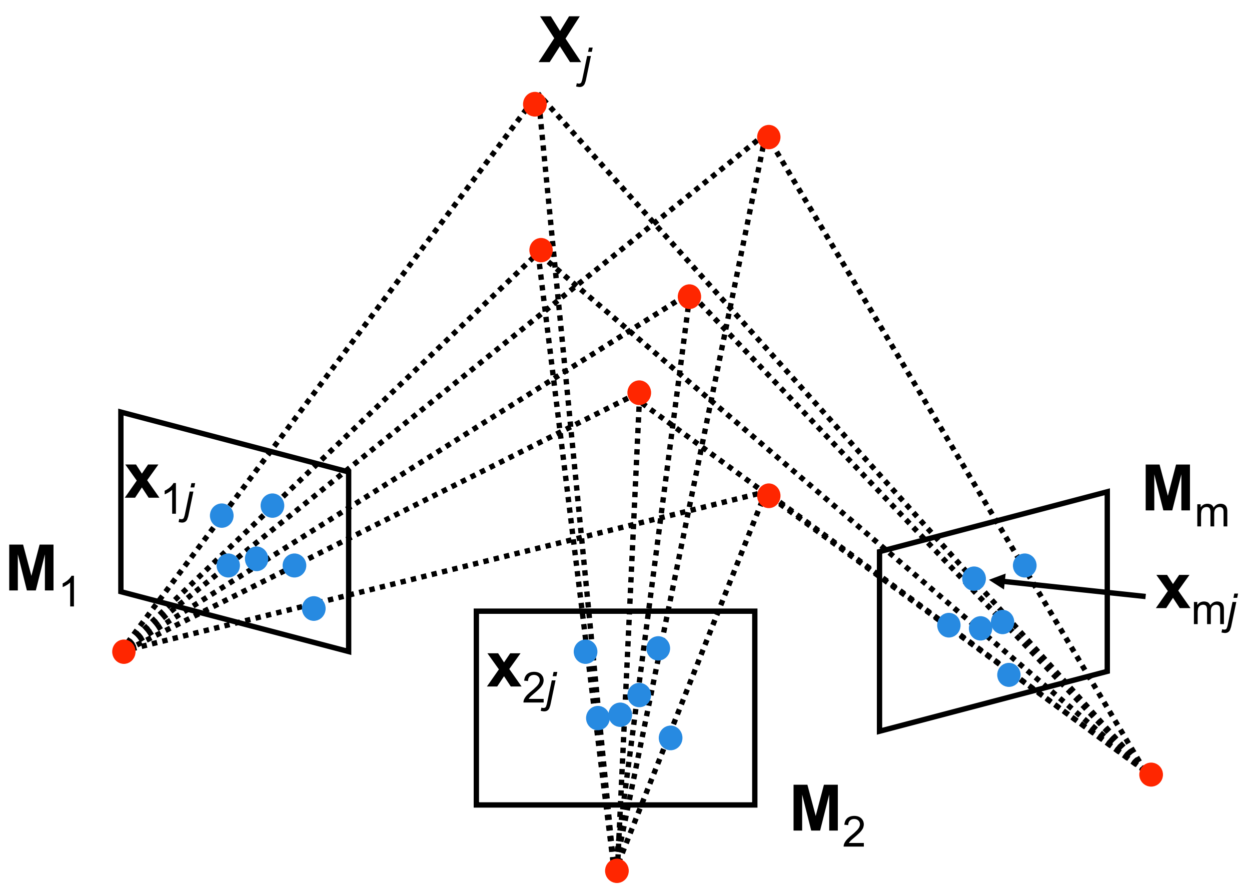
Structure from motion problem



Given *m* images of *n* fixed 3D points

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

Structure from motion problem



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

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

Structure from motion ambiguity

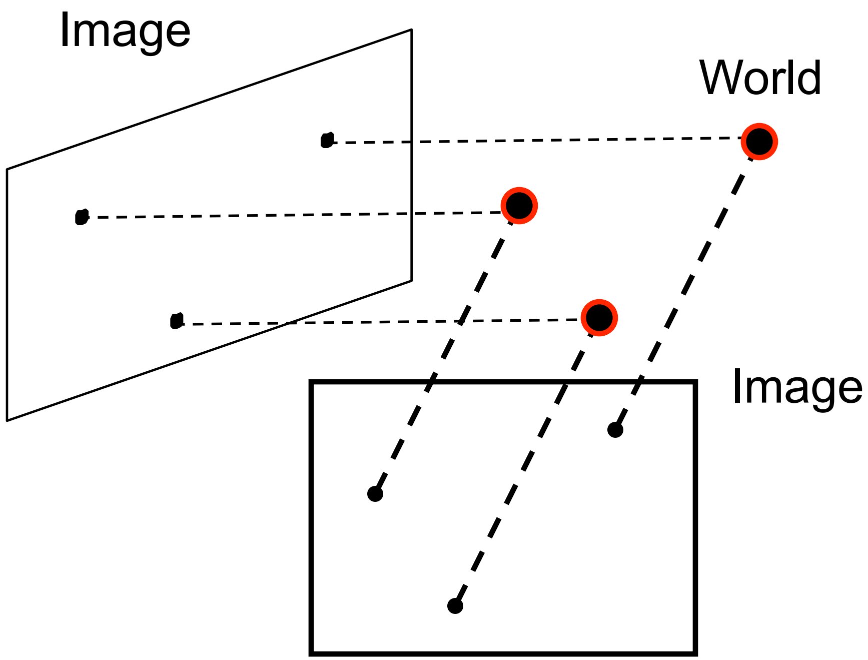
- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\,\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

Affine structure from motion

(simpler problem)

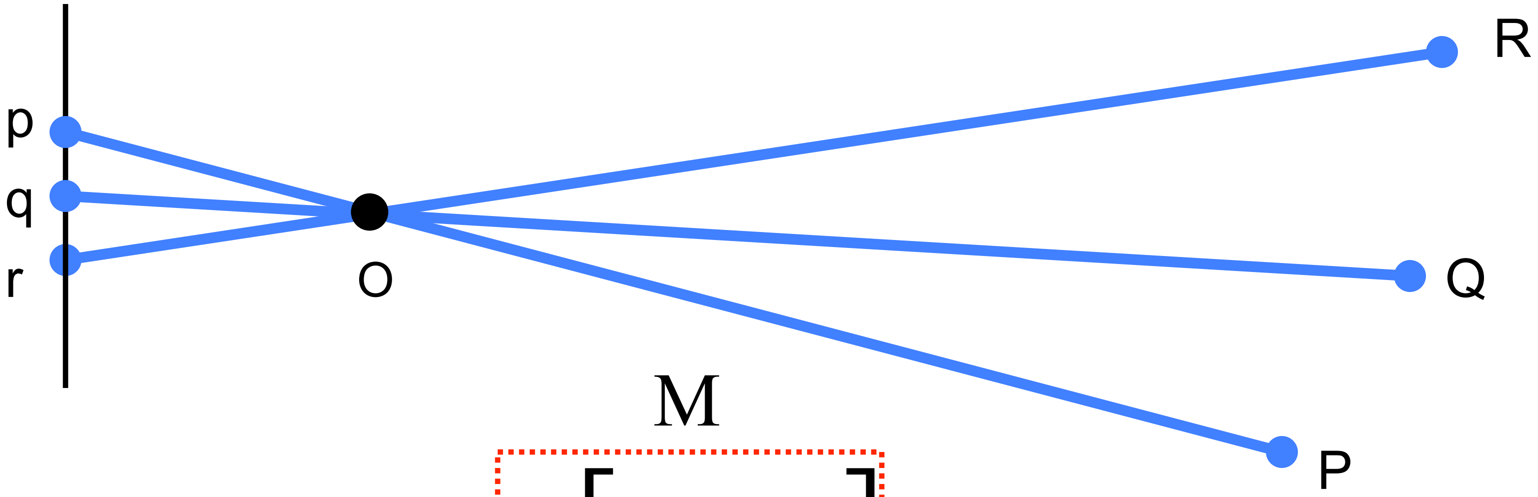


From the m×n correspondences $\mathbf{x}_{ij}$, estimate:

•*m* projection matrices $\mathbf{M}_i$ (affine cameras)

•*n* 3D points $\mathbf{X}_j$

Finite cameras

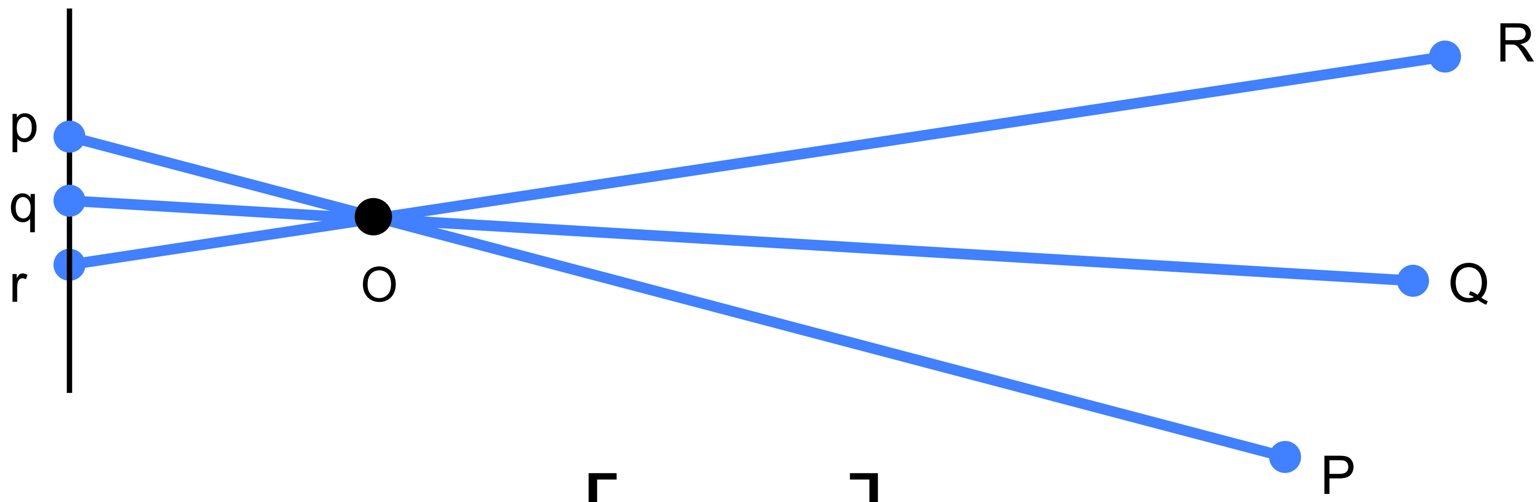


$$x = \underbrace{K\begin{bmatrix} R & T \end{bmatrix}}_{M} X$$

Question:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = ??$$

Finite cameras



$$x = K\begin{bmatrix} R & T \end{bmatrix} X$$

Canonical perspective projection matrix

$$M = \boxed{K_{3x3}} \boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \boxed{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}$$

Affine Homography (in 2D)

Affine homography (in 3D)

$$K = \begin{bmatrix} \alpha_x & s & x_o \\ 0 & \alpha_y & y_o \\ 0 & 0 & 1 \end{bmatrix}$$

Projective & Affine cameras

$$\mathrm{x} = \mathrm{K}\begin{bmatrix} \mathrm{R} & \mathrm{T} \end{bmatrix}\mathrm{X}$$

Projective case

$$\mathrm{K} = \begin{bmatrix} \boldsymbol{\alpha}_{\mathrm{x}} & \mathrm{s} & \mathrm{x}_{\mathrm{o}} \\ 0 & \boldsymbol{\alpha}_{\mathrm{y}} & \mathrm{y}_{\mathrm{o}} \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{M} = \mathrm{K}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \mathrm{R} & \mathrm{T} \\ 0 & 1 \end{bmatrix}$$

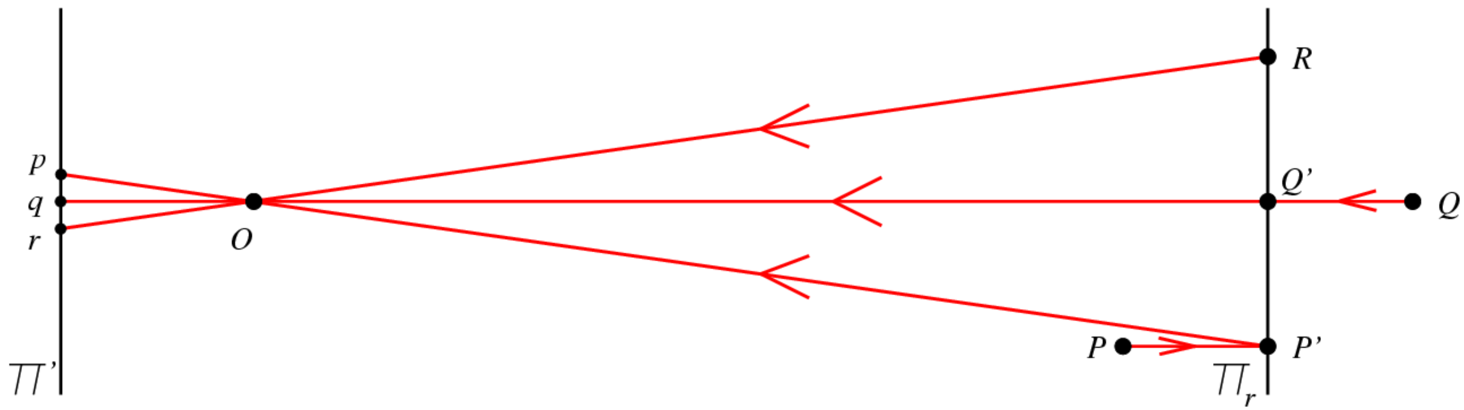
Affine case

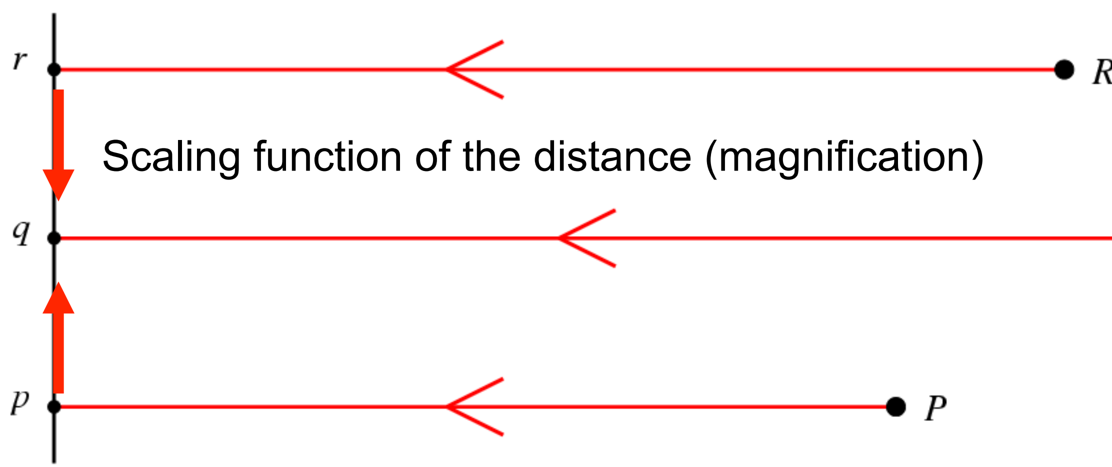
$$K = \begin{bmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad M = K\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

Weak perspective projection

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

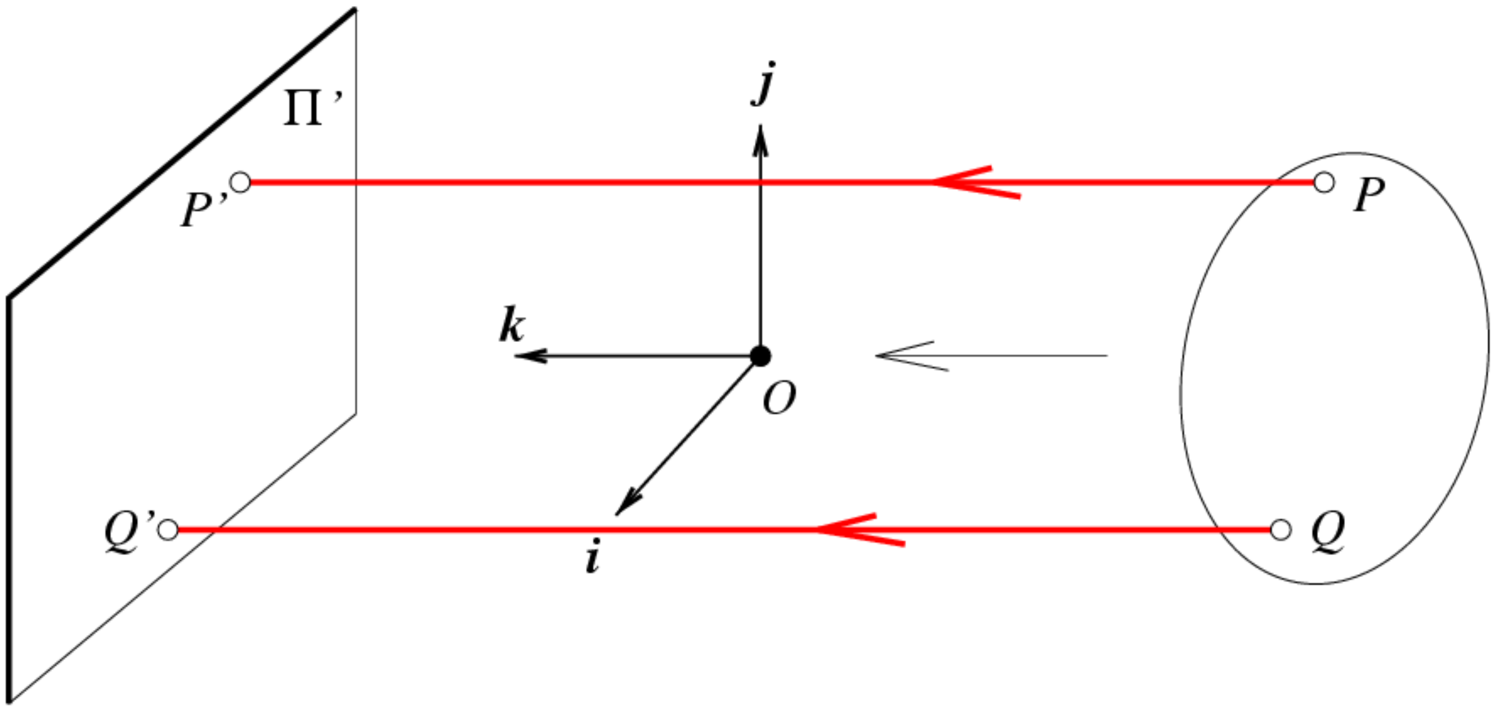
$$\begin{cases} x' = -mx \\ y' = -my \end{cases}$$





Orthographic (affine) projection

When the camera is at a (roughly constant) distance from the scene



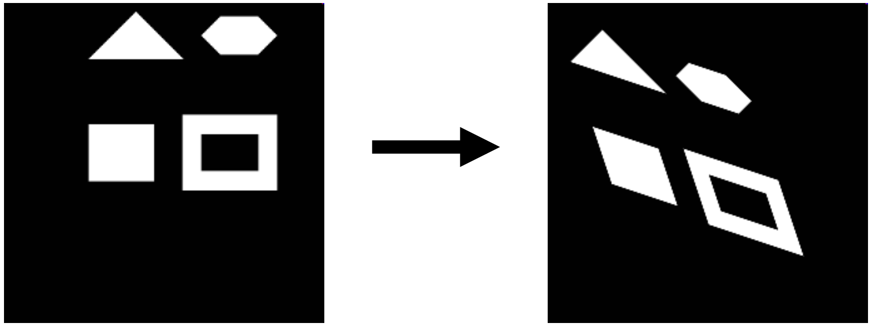
–Distance from center of projection
to image plane is infinite

$$\begin{cases} x' = x \\ y' = y \end{cases}$$

Transformation in 2D

Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Projective & Affine cameras

$$x = K\begin{bmatrix} R & T \end{bmatrix} X$$

Projective case

$$K = \begin{bmatrix} \alpha_x & s & x_o \\ 0 & \alpha_y & y_o \\ 0 & 0 & 1 \end{bmatrix} \qquad M = K\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

Affine case

$$K = \begin{bmatrix} \alpha_x & s & x_o \\ 0 & \alpha_y & y_o \\ 0 & 0 & 1 \end{bmatrix} \qquad M = K\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \boxed{0} & 1 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

Magnification (scaling term)

Parallel projection matrix
(points at infinity are mapped as points at infinity)

Affine cameras

$$\mathrm{x} = \mathrm{K}\begin{bmatrix}\mathrm{R} & \mathrm{T}\end{bmatrix}\mathrm{X}$$ [Homogeneous]

$$K = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{M} = \mathrm{K}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \mathrm{R} & \mathrm{T} \\ 0 & 1 \end{bmatrix}$$

$$M = [3\times 3\,\text{affine}]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}[4\times 4\,\text{affine}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

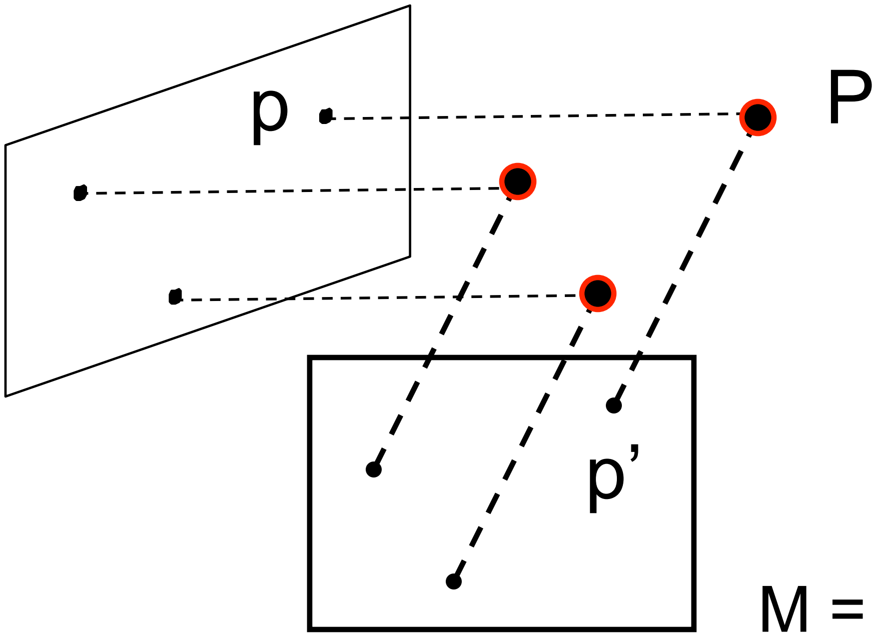
$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{AX} + \mathbf{b} = M_{Euc}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = M_{Euc}\begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix};$$

$$\mathrm{M}_{\mathrm{Euc}} = \mathrm{M} = \begin{bmatrix}\mathrm{A} & \mathrm{b}\end{bmatrix}$$ [non-homogeneous image coordinates]

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

- The point b is the image location of the world origin
- The center of the affine camera is infinity

Affine cameras



M = camera matrix

To recap:

from now on we define M as the camera matrix for the affine case

$$\mathbf{p} = \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{AP} + \mathbf{b} = M \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}; \qquad \mathrm{M} = \begin{bmatrix} \mathrm{A} & \mathrm{b} \end{bmatrix}$$

The Affine Structure-from-Motion Problem

Given m images of n fixed points P_j (=X_i) we can write

$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = \mathcal{A}_i \boldsymbol{P}_j + \boldsymbol{b}_i \quad \text{for} \quad i = 1, \ldots, \boxed{m} \quad \text{and} \quad j = 1, \ldots, \boxed{n}.$$

N of cameras N of points

Problem: estimate the m 2×4 matrices M_i and the n positions P_j from the m×n correspondences $\mathbf{p}_{ij}$.

How many equations and how many unknown?

2m × n equations in 8m+3n unknowns

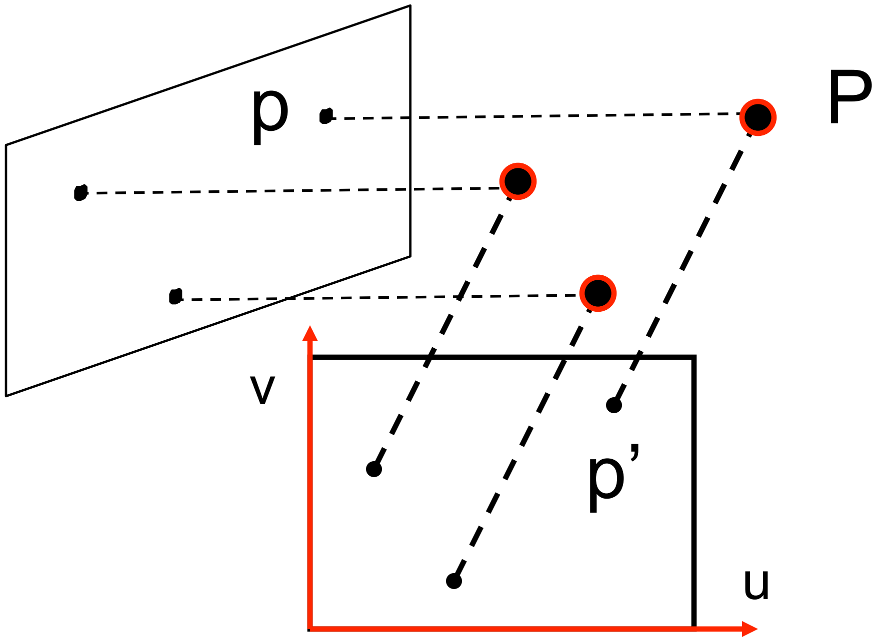
Two approaches:

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

Algebraic analysis (2-view case)

- Derive the fundamental matrix F_A for the affine case
- Compute F_A
- Use F_A to estimate projection matrices
- Use projection matrices to estimate 3D points

1. Deriving the fundamental matrix F_A



Homogeneous system

$$\begin{cases} \boldsymbol{p} = \mathcal{A}\boldsymbol{P} + \boldsymbol{b} \\ \boldsymbol{p}' = \mathcal{A}'\boldsymbol{P} + \boldsymbol{b}' \end{cases} \Rightarrow \begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} \begin{pmatrix} \boldsymbol{P} \\ -1 \end{pmatrix} = \mathbf{0}$$

Dim= ? 4x4

$$\Rightarrow \operatorname{Det} \begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} = 0 \Rightarrow \boxed{\alpha u + \beta v + \alpha' u' + \beta' v' + \delta = 0}$$

Deriving the fundamental matrix F_A

$$\alpha u + \beta v + \alpha' u' + \beta' v' + \delta = 0$$

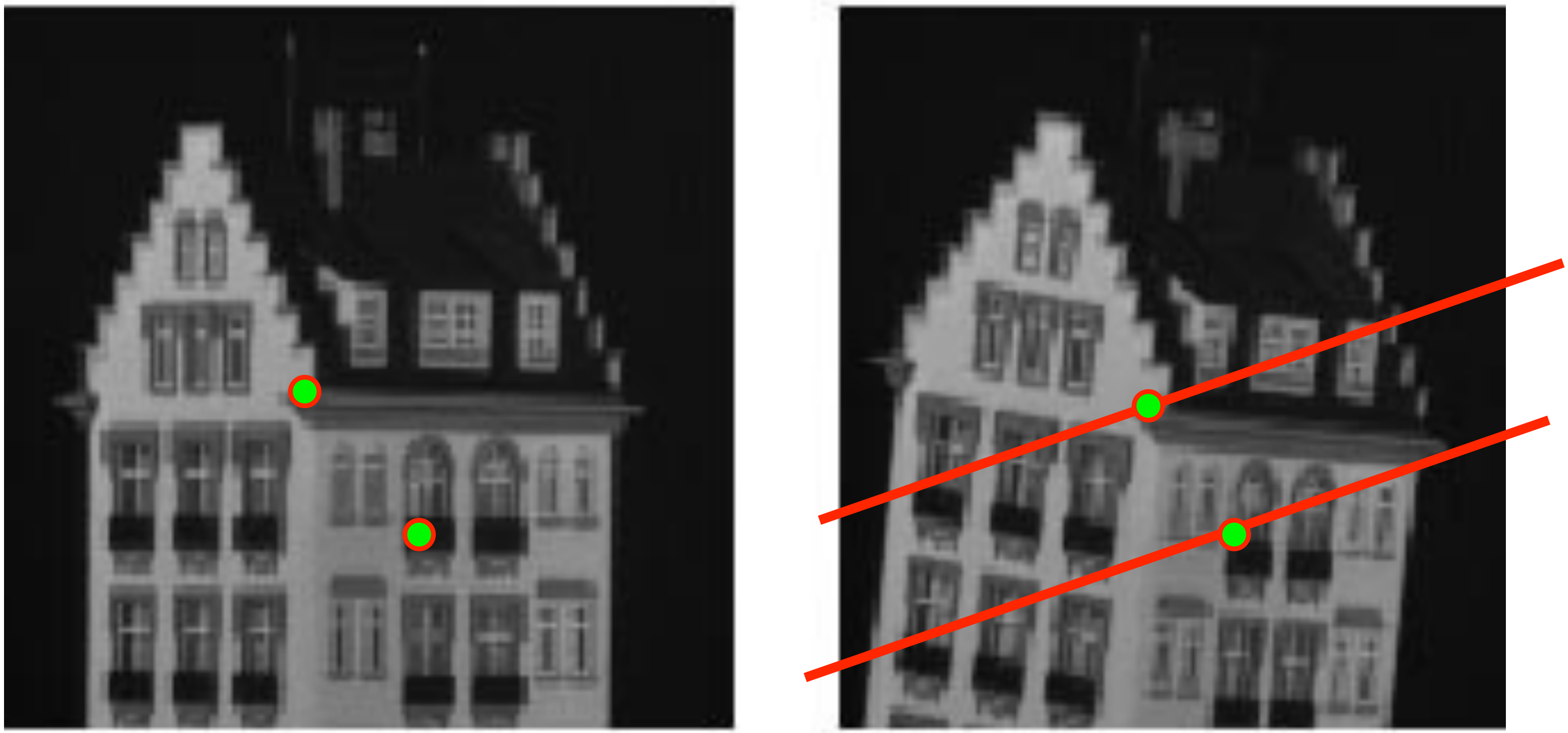
$$(u, v, 1)\mathcal{F}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

where

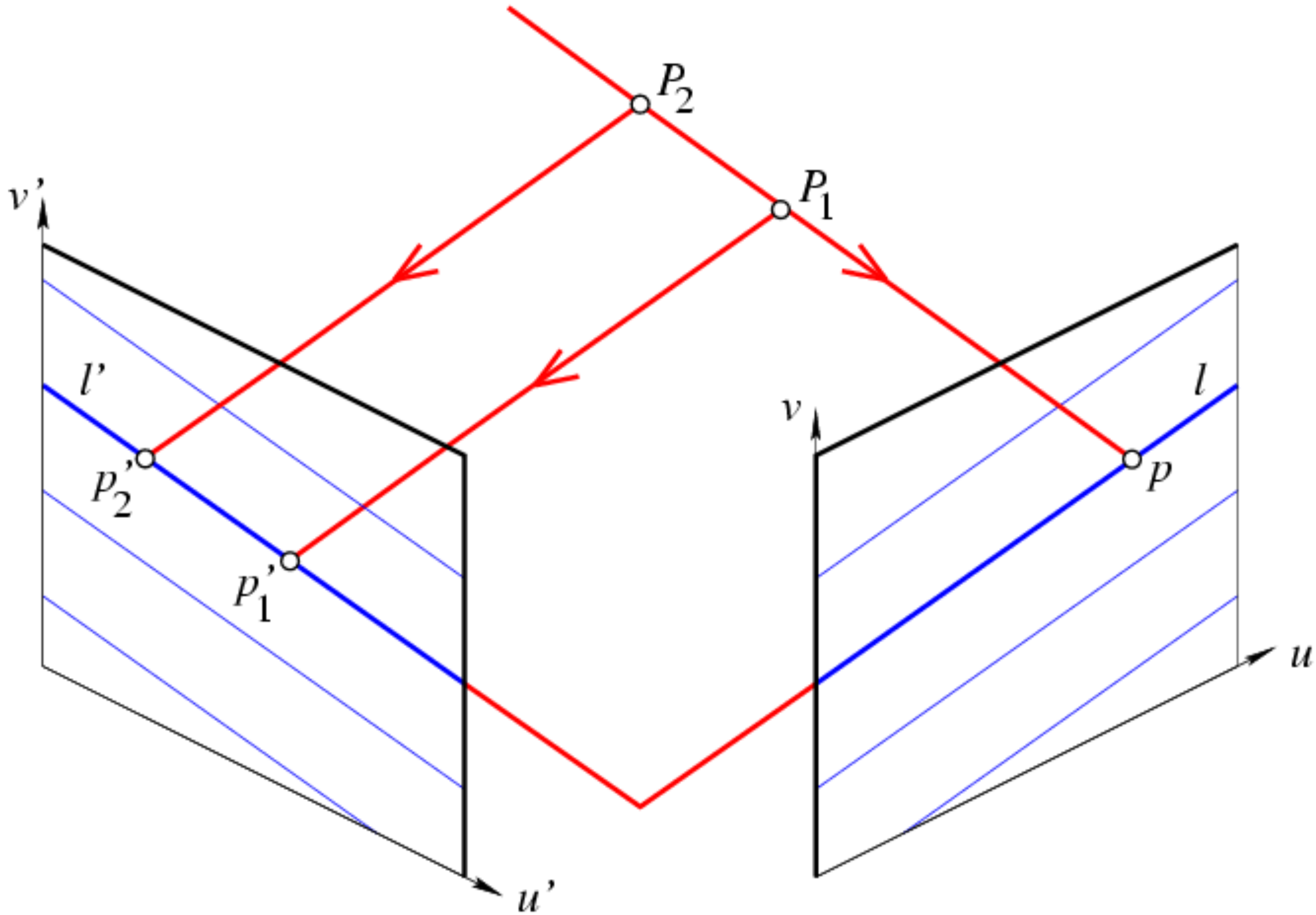
$$\mathcal{F} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 0 & \alpha \\ 0 & 0 & \beta \\ \alpha' & \beta' & \delta \end{pmatrix}$$

The Affine Fundamental Matrix!

Are the epipolar lines parallel or converging?



Affine Epipolar Geometry



Estimating F_A

$$\alpha u + \beta v + \alpha' u' + \beta' v' + \delta = 0$$

- Measurements: u, u’, v, v’
- From n correspondences, we obtain a linear system on the unknown alpha, beta, etc…

$$\begin{bmatrix} u_1' & v_1' & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \mathbf{f} = 0$$

- Computed by least square and by enforcing |f|=1
- SVD

2. Estimating projection matrices from epipolar constraints

If M_i and P_i are solutions,
then M_i' and P_i' are also solutions,

where

$$\mathcal{M}_i' = \mathcal{M}_i \mathcal{Q} \quad \text{and} \quad \begin{pmatrix} \boldsymbol{P}_j' \\ 1 \end{pmatrix} = \mathcal{Q}^{-1} \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix}$$

and

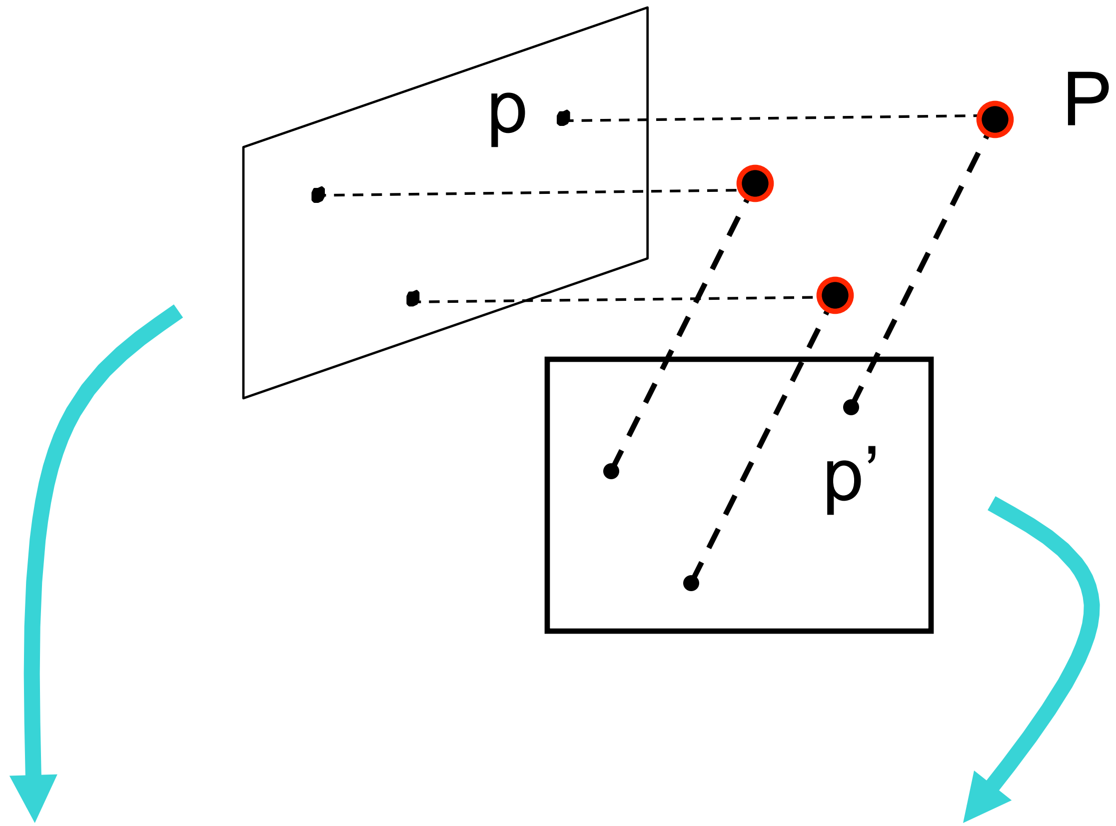
$$\mathcal{Q} = \begin{pmatrix} \mathcal{C} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

Q is an affine transformation.

Proof:

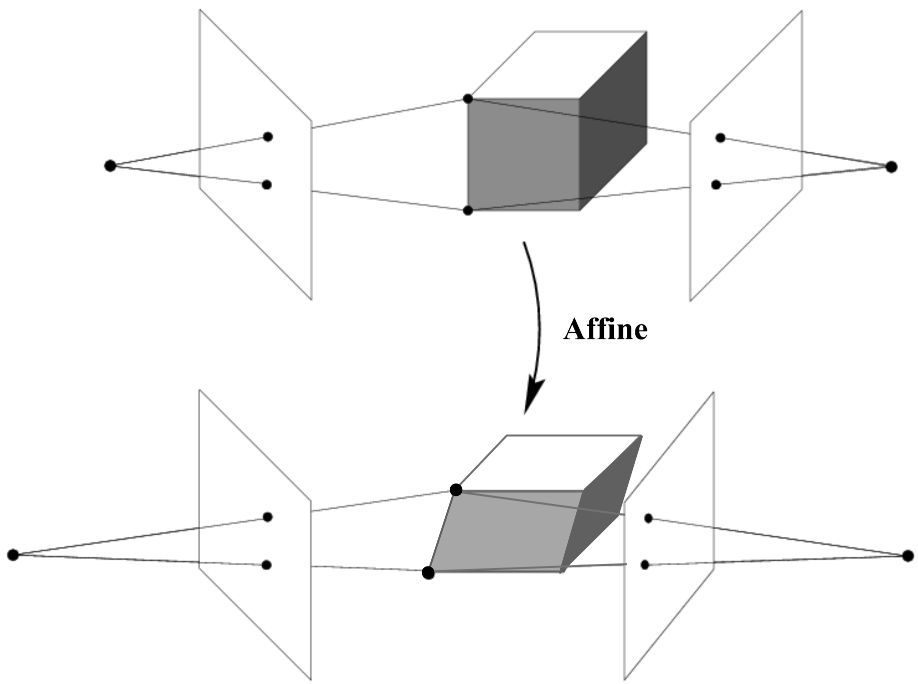
$$\boldsymbol{p}_{ij} = \mathcal{M}_i \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} = (\mathcal{M}_i \mathcal{Q}) \left(\mathcal{Q}^{-1} \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} \right) = \mathcal{M}_i' \begin{pmatrix} \boldsymbol{P}_j' \\ 1 \end{pmatrix} \quad \blacksquare$$

Estimating projection matrices from F_A



$$\mathcal{M} = (\,\mathcal{A} \quad \boldsymbol{b}\,) \qquad \mathcal{M}' = (\,\mathcal{A}' \quad \boldsymbol{b}'\,)$$

Affine ambiguity



$$\mathbf{p} = \mathbf{M}\,\mathbf{P} = \left(\mathbf{M}\,\mathbf{Q}_A^{-1}\right)\left(\mathbf{Q}_A\,\mathbf{P}\right)$$

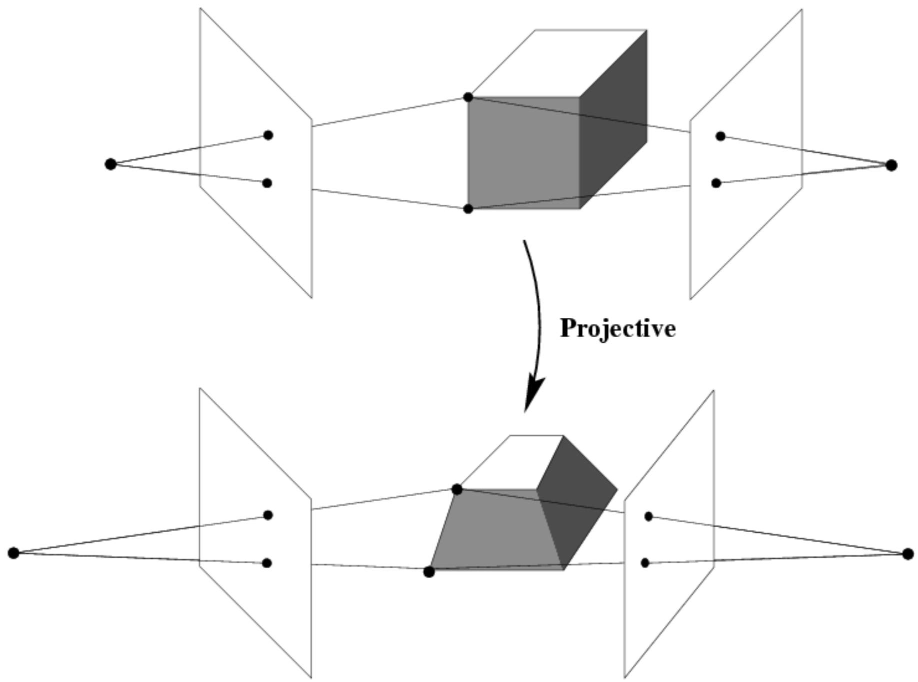
Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

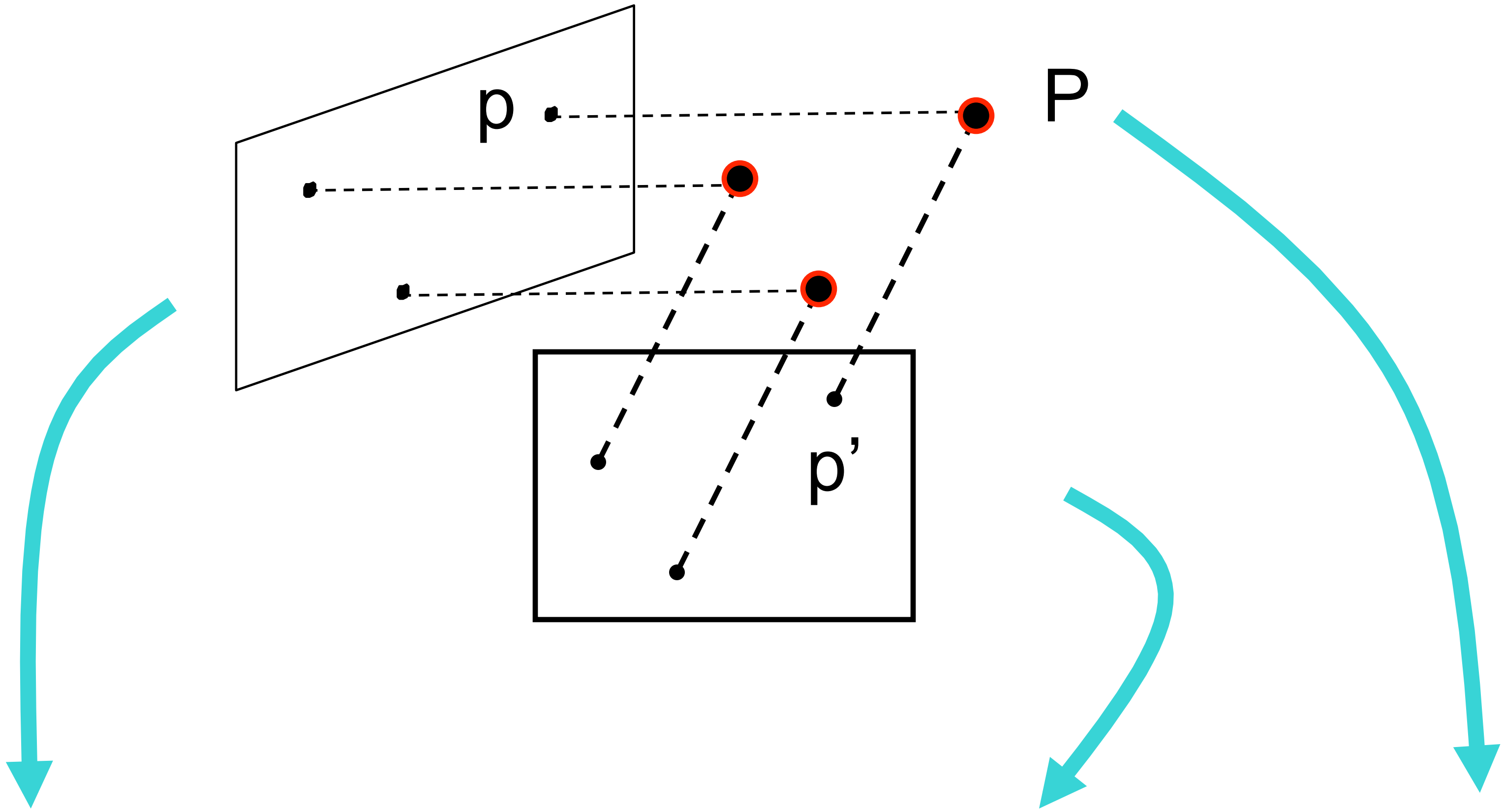
- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, then the images do not change

Projective ambiguity



$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathbf{P}}^{-1}\right)\left(\mathbf{Q}_{\mathbf{P}}\mathbf{X}\right)$$

Estimating projection matrices from F_A



$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}) \qquad \mathcal{M}' = (\mathcal{A}' \quad \boldsymbol{b}') \qquad \boldsymbol{P}$$

Estimating projection matrices from F_A

$$\mathcal{M} = \begin{pmatrix} \mathcal{A} & \boldsymbol{b} \end{pmatrix} \;\downarrow\; \tilde{\mathcal{M}} = \mathcal{M}\mathcal{Q} \;\downarrow\; \tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \;\downarrow\; \tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Choose Q such that…

$$\mathcal{M}' = \begin{pmatrix} \mathcal{A}' & \boldsymbol{b}' \end{pmatrix} \;\downarrow\; \tilde{\mathcal{M}}' = \mathcal{M}'\mathcal{Q} \;\downarrow\; \widetilde{M}' \;\downarrow\; \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix}$$

$$\boldsymbol{P} \;\downarrow\; \tilde{\boldsymbol{P}} = \mathcal{Q}^{-1}\boldsymbol{P} \;\downarrow\; \tilde{\boldsymbol{P}} \;\downarrow\; \tilde{\boldsymbol{P}} = \begin{pmatrix} u \\ v \\ u' \end{pmatrix}$$

Where a,b,c,d can be expressed as function of the parameters of F_A

See HZ page 348

4. Estimating the structure from F_A

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix}$$

A b A' b'

$$\begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} \begin{pmatrix} \boldsymbol{P} \\ -1 \end{pmatrix} = \boldsymbol{0} \quad \Rightarrow$$

$$\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v' - d \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{P}} \\ -1 \end{pmatrix} = 0 \quad \Rightarrow \quad \boxed{\tilde{\boldsymbol{P}} = \begin{pmatrix} u \\ v \\ u' \end{pmatrix}}$$

Can be solved by least square again

Estimating projection matrices from epipolar constraints

$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}) \qquad \mathcal{M}' = (\mathcal{A}' \quad \boldsymbol{b}') \qquad \boldsymbol{P}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\tilde{\mathcal{M}} = \mathcal{M}\mathcal{Q} \qquad \tilde{\mathcal{M}}' = \mathcal{M}'\mathcal{Q} \qquad \tilde{\boldsymbol{P}} = \mathcal{Q}^{-1}\boldsymbol{P}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

Choose Q such that…

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \qquad \tilde{\boldsymbol{P}}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \tilde{A}' = \begin{bmatrix} 0 & 0 & 1 \\ \boxed{a \quad b \quad c} \end{bmatrix}$$

$$\tilde{b} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \qquad \tilde{b}' = \begin{bmatrix} 0 & \boxed{d} \end{bmatrix}^T$$

Canonical affine cameras

Function of the parameters of F

Estimating projection matrices from epipolar constraints

$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}) \qquad \mathcal{M}' = (\mathcal{A}' \quad \boldsymbol{b}') \qquad \boldsymbol{P}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$\tilde{\mathcal{M}} = \mathcal{M}\mathcal{Q} \qquad \tilde{\mathcal{M}}' = \mathcal{M}'\mathcal{Q} \qquad \tilde{\boldsymbol{P}} = \mathcal{Q}^{-1}\boldsymbol{P}$$

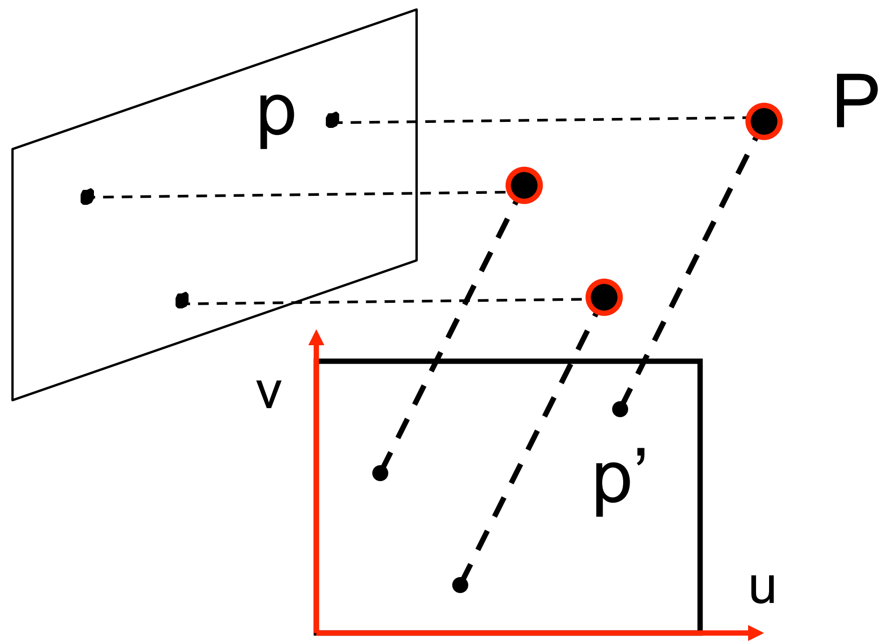
$$\downarrow \qquad \downarrow \qquad \downarrow$$

Choose Q such that…

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \quad \tilde{\boldsymbol{P}}$$

By re-enforcing the epipolar constraint, we can compute a, b, c, d directly from the measurements

Reminder: epipolar constraint



Homogeneous system

$$\begin{cases} \boldsymbol{p} = \mathcal{A}\boldsymbol{P} + \boldsymbol{b} \\ \boldsymbol{p}' = \mathcal{A}'\boldsymbol{P} + \boldsymbol{b}' \end{cases} \Rightarrow \begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} \begin{pmatrix} \boldsymbol{P} \\ -1 \end{pmatrix} = \boldsymbol{0}$$

$$\Rightarrow \boxed{\operatorname{Det} \begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} = 0} \Rightarrow \alpha u + \beta v + \alpha' u' + \beta' v' + \delta = 0$$

Estimating projection matrices from epipolar constraints

$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}) \qquad \mathcal{M}' = (\mathcal{A}' \quad \boldsymbol{b}') \qquad \boldsymbol{P}$$

$$\tilde{\mathcal{M}} = \mathcal{M}\mathcal{Q} \qquad \tilde{\mathcal{M}}' = \mathcal{M}'\mathcal{Q} \qquad \tilde{\boldsymbol{P}} = \mathcal{Q}^{-1}\boldsymbol{P}$$

Choose Q such that…

$$\tilde{\mathcal{M}} = \left(\underbrace{\boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}}}_{\tilde{A}} \; \underbrace{\boxed{\begin{matrix} 0 \\ 0 \end{matrix}}}_{\tilde{b}}\right) \qquad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \qquad \tilde{\boldsymbol{P}}$$

Re-enforce the Epipolar constraint

$$\operatorname{Det}\begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} = 0 \quad \rightarrow \quad \operatorname{Det}\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v' - d \end{pmatrix} = 0$$

Estimating projection matrices from epipolar constraints

$$\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b}) \qquad \mathcal{M}' = (\mathcal{A}' \quad \boldsymbol{b}') \qquad \boldsymbol{P}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\tilde{\mathcal{M}} = \mathcal{M}\mathcal{Q} \qquad \tilde{\mathcal{M}}' = \mathcal{M}'\mathcal{Q} \qquad \tilde{\boldsymbol{P}} = \mathcal{Q}^{-1}\boldsymbol{P}$$

Choose Q such that…

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$\tilde{\mathcal{M}} = \left(\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}}_{A} \; \underbrace{\begin{matrix} 0 \\ 0 \end{matrix}}_{b}\right) \qquad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \qquad \tilde{\boldsymbol{P}}$$

$$\boxed{\operatorname{Det}\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v' - d \end{pmatrix} = au - bv + cu' + v' - d = 0}$$

Estimating projection matrices from epipolar constraints

$$\mathrm{Det}\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v'-d \end{pmatrix} = au - bv + cu' + v' - d = 0$$

- Linear relationship between measurements and unknown

  Unknown: a, b, c, d
  Measurements: u, u', v, v'

- From at least 4 correspondences, we can solve this linear system and compute a, b, c, d (via least square)
- The cameras can be computed
- How about the structure?

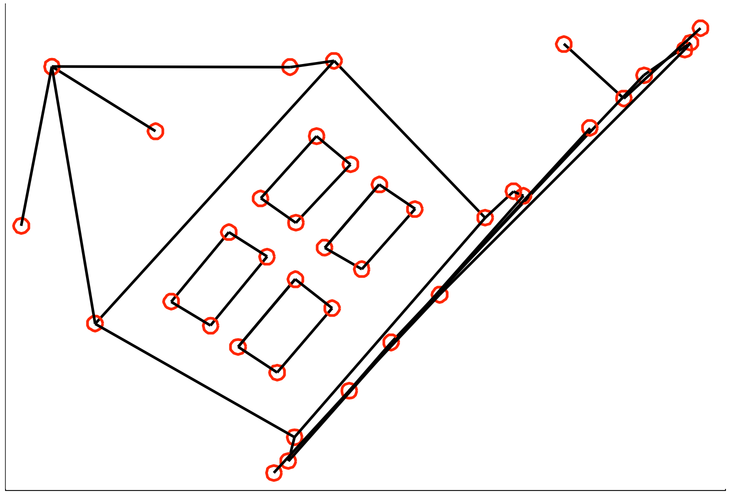
4. Estimating the structure from F_A

$$\tilde{\mathcal{M}} = \begin{pmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}}_{A} & \underbrace{\begin{matrix} 0 \\ 0 \end{matrix}}_{b} \end{pmatrix} \quad \tilde{\mathcal{M}}' = \begin{pmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \quad \tilde{\boldsymbol{P}}$$

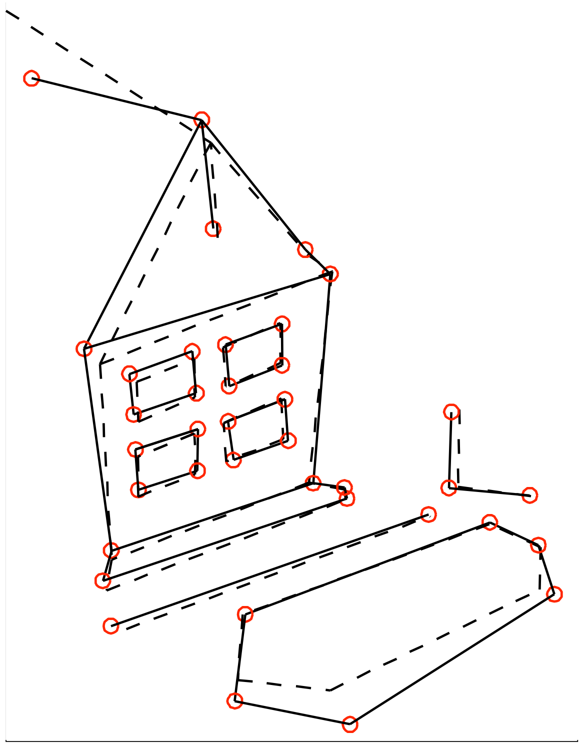
$$\begin{pmatrix} \mathcal{A} & \boldsymbol{p} - \boldsymbol{b} \\ \mathcal{A}' & \boldsymbol{p}' - \boldsymbol{b}' \end{pmatrix} \begin{pmatrix} \boldsymbol{P} \\ -1 \end{pmatrix} = \boldsymbol{0} \quad \Rightarrow$$

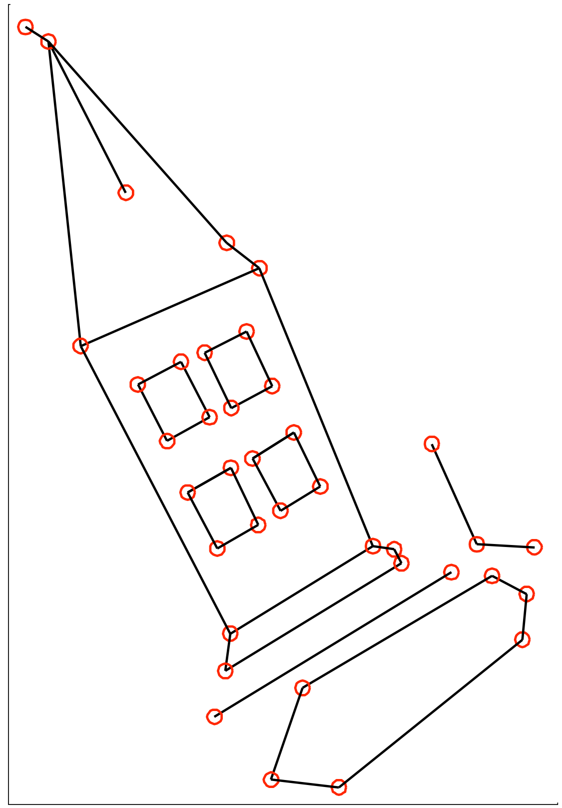
$$\begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & u' \\ a & b & c & v' - d \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{P}} \\ -1 \end{pmatrix} = 0 \quad \Rightarrow \quad \boxed{\tilde{\boldsymbol{P}} = \begin{pmatrix} u \\ v \\ u' \end{pmatrix}}$$

Can be solved by least square again



First reconstruction. Mean re-projection error: 1.6pixel





Second reconstruction. Mean re-projection error: 7.8pixel

A factorization method – Tomasi & Kanade algorithm

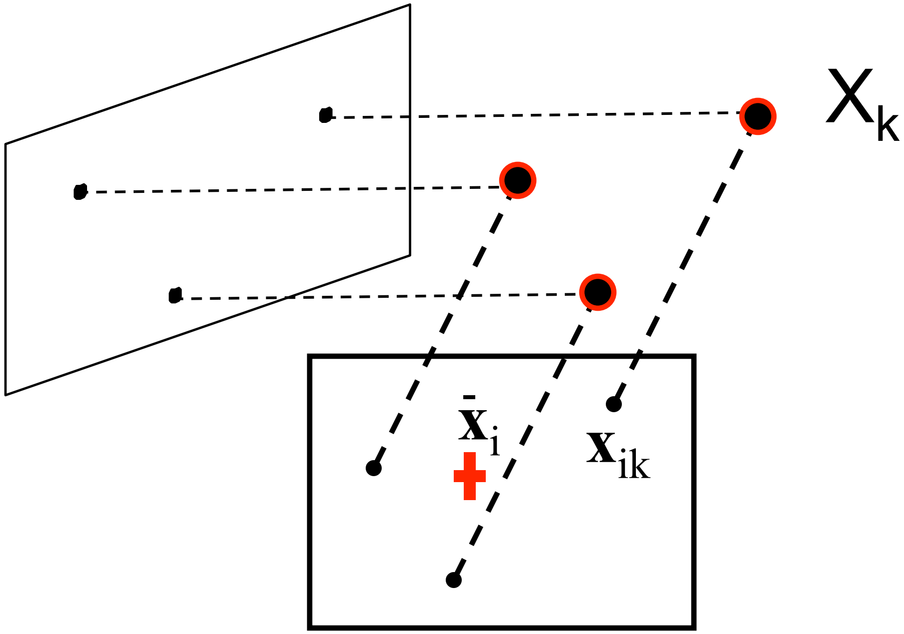
C. Tomasi and T. Kanade.
Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

- Centering the data
- Factorization

A factorization method - Centering the data

Centering: subtract the centroid of the image points

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



A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$\begin{cases} \hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_{ik} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n} \left(\mathbf{A}_i \mathbf{X}_k + \mathbf{b}_i \right) \\ \mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i \end{cases}$$

A factorization method - Centering the data

Centering: subtract the centroid of the image points

$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i\right)$$

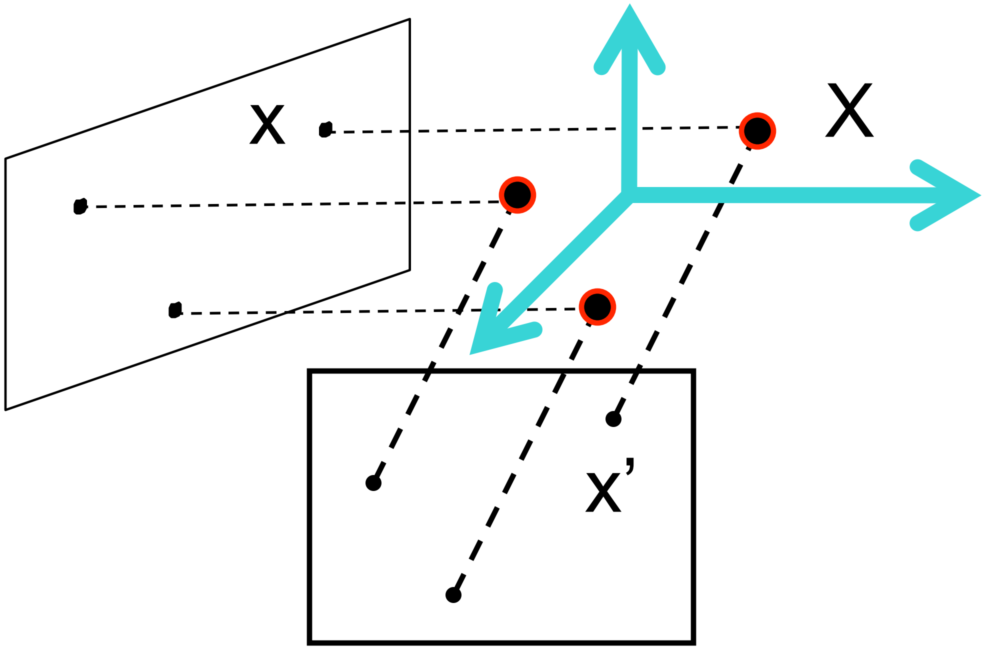
$$= \mathbf{A}_i\left(\mathbf{X}_j - \boxed{\frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k}\right)$$

Assume that the origin of the world coordinate system is at the centroid of the 3D points

After centering, each normalized point $\mathbf{x}_{ij}$ is related to the 3D point $\mathbf{X}_i$ by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i\mathbf{X}_j$$

A factorization method - Centering the data



$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \mathbf{X}_j$$

A factorization method - factorization

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

$$\begin{array}{cc} \mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} & \text{cameras } (2m) \\ \text{points } (n) & \end{array}$$

A factorization method - factorization

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

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

$(2m \times n)$

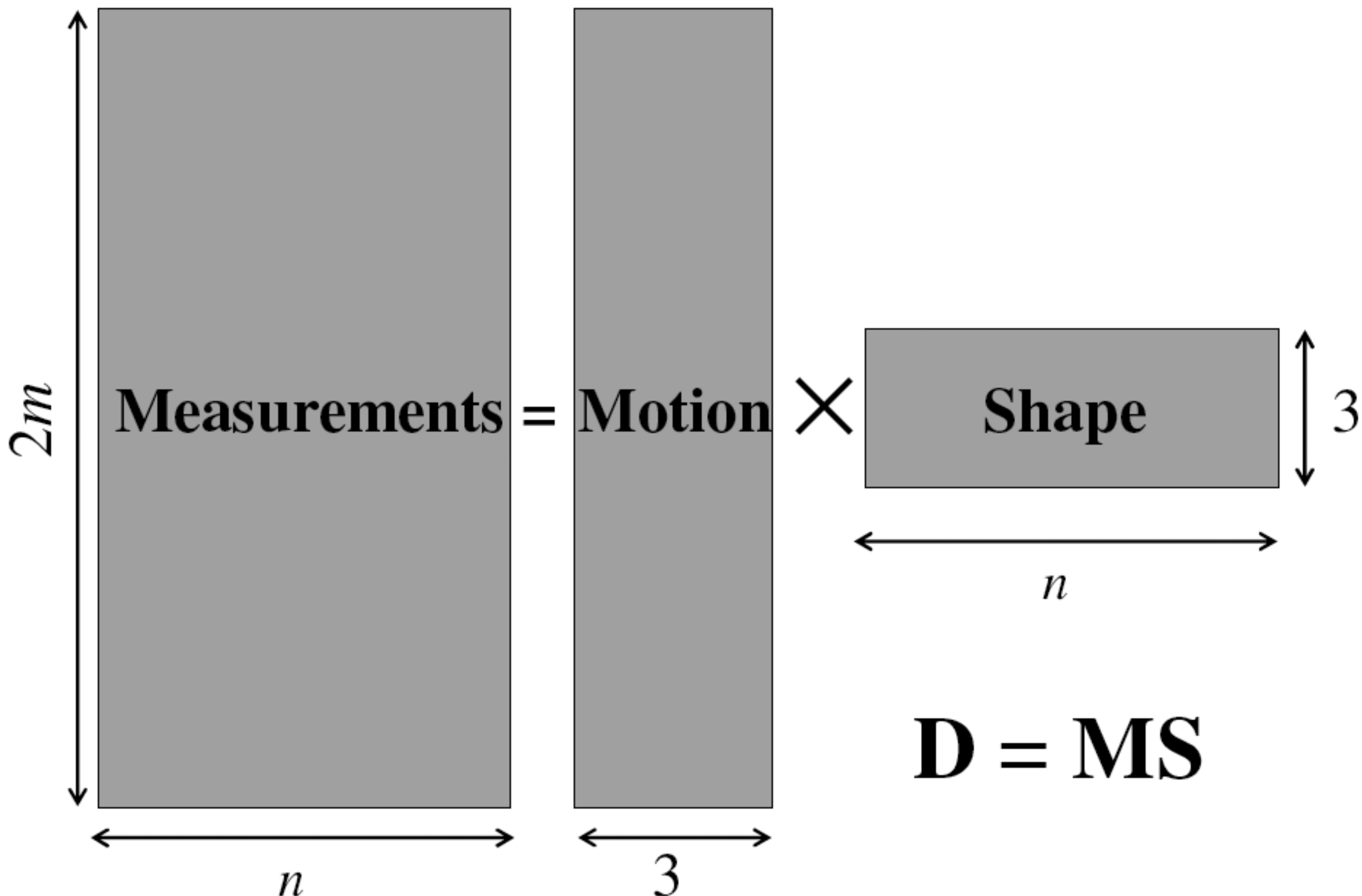
cameras $(2m \times 3)$

points $(3 \times n)$

The measurement matrix **D = M S** has rank 3

(it's a product of a 2mx3 matrix and 3xn matrix)

Factorizing the measurement matrix

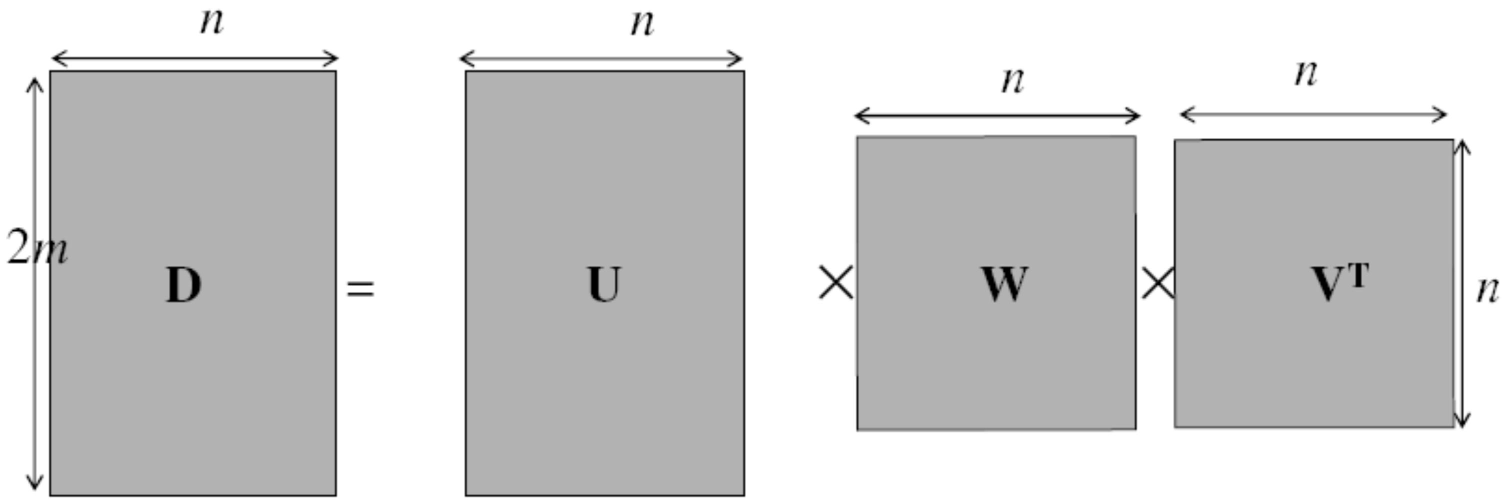


$$\mathbf{D} = \mathbf{M}\mathbf{S}$$

Source: M. Hebert

Factorizing the measurement matrix

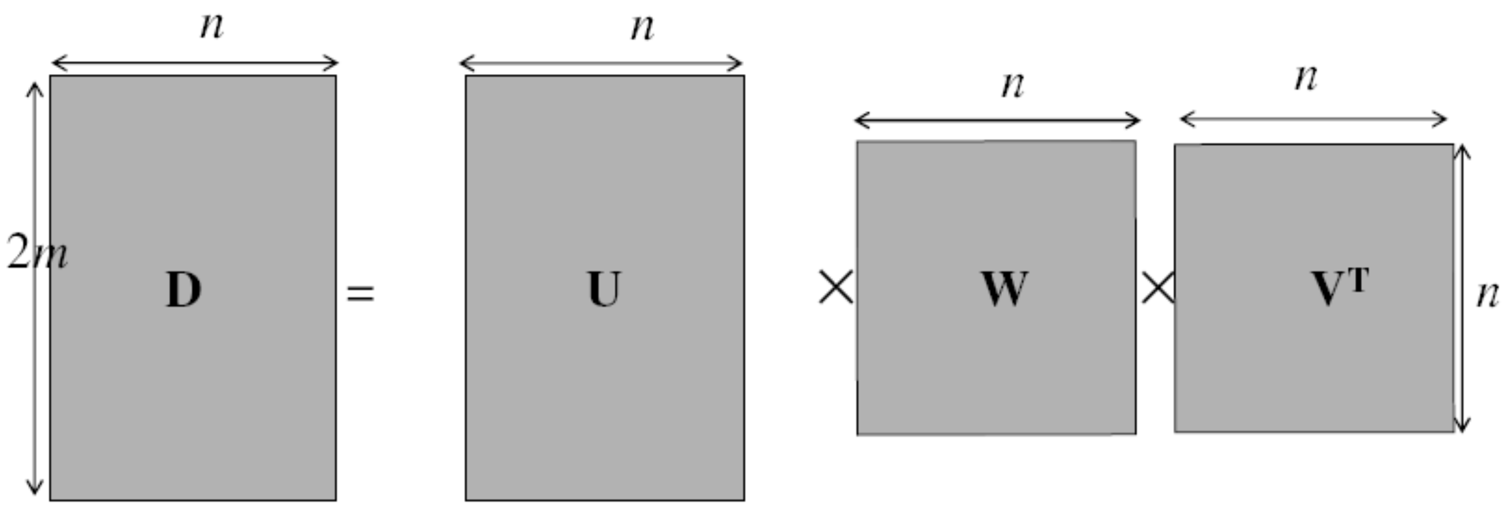
Singular value decomposition of D:



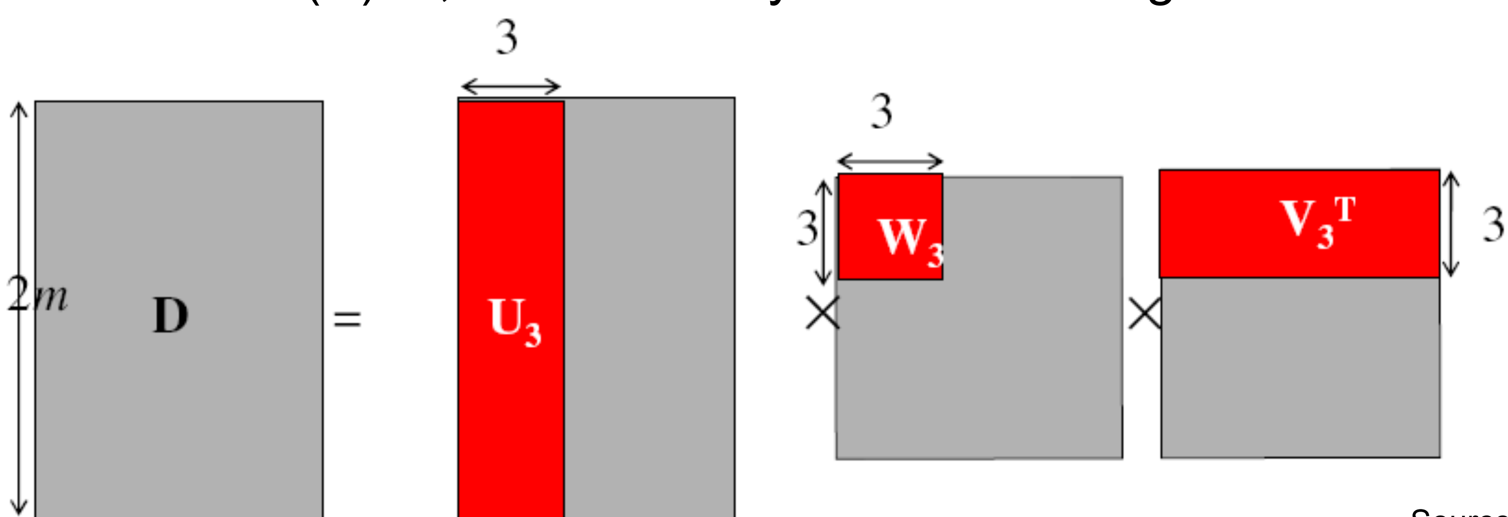
Source: M. Hebert

Factorizing the measurement matrix

Singular value decomposition of D:



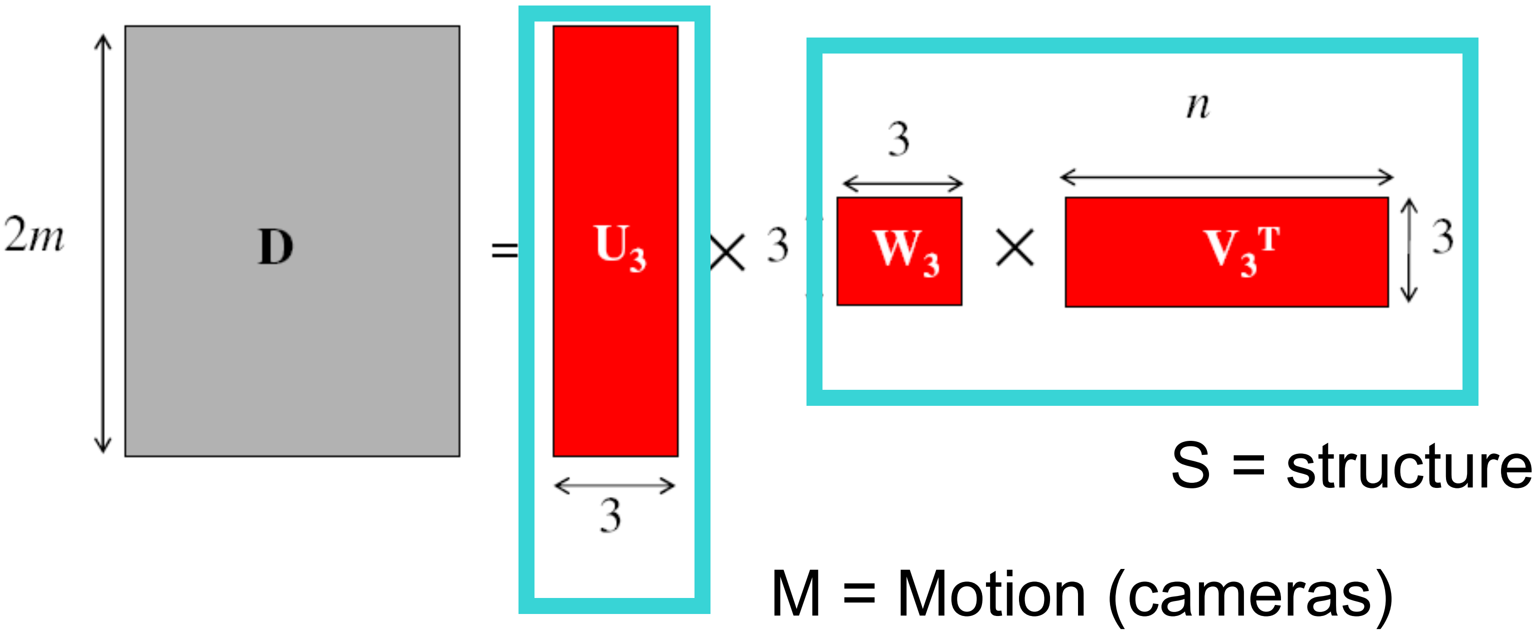
Since rank (D)=3, there are only 3 non-zero singular values



Source: M. Hebert

Factorizing the measurement matrix

Obtaining a factorization from SVD:

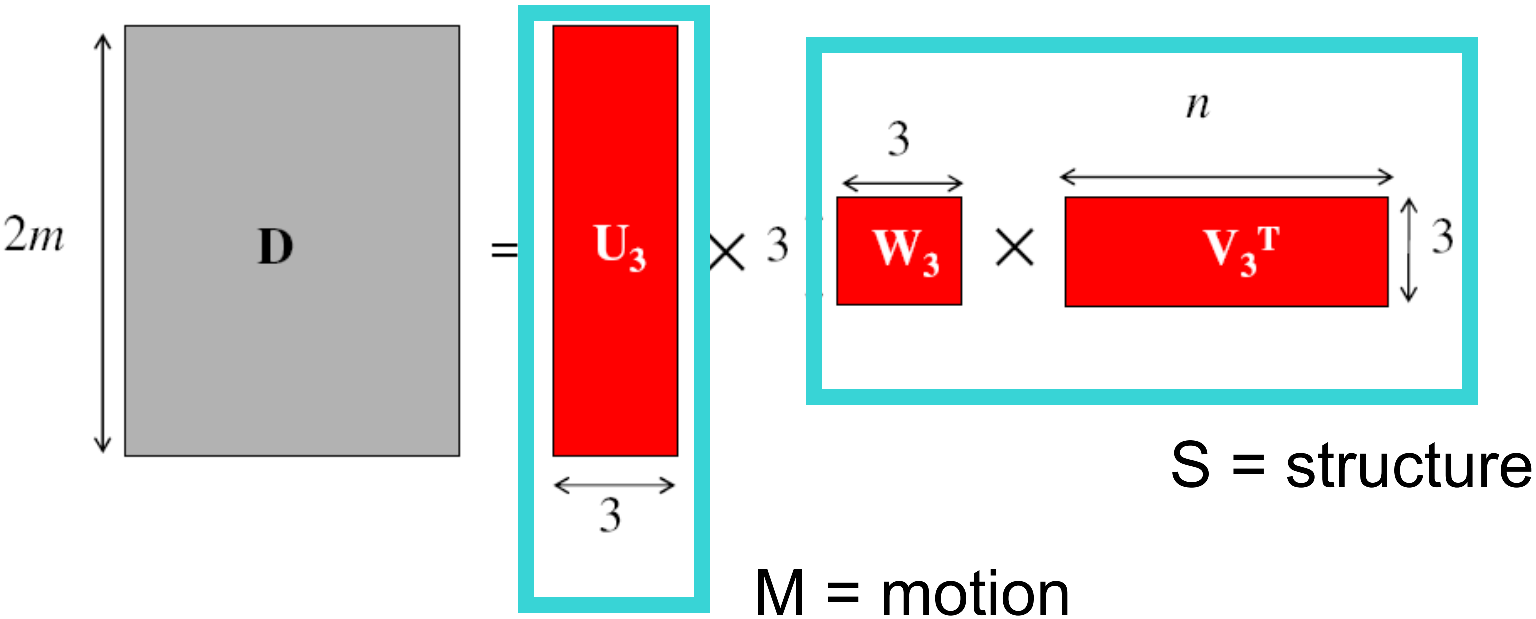


What is the issue here?

D has rank>3 because of - measurement noise
- affine approximation

Factorizing the measurement matrix

Obtaining a factorization from SVD:

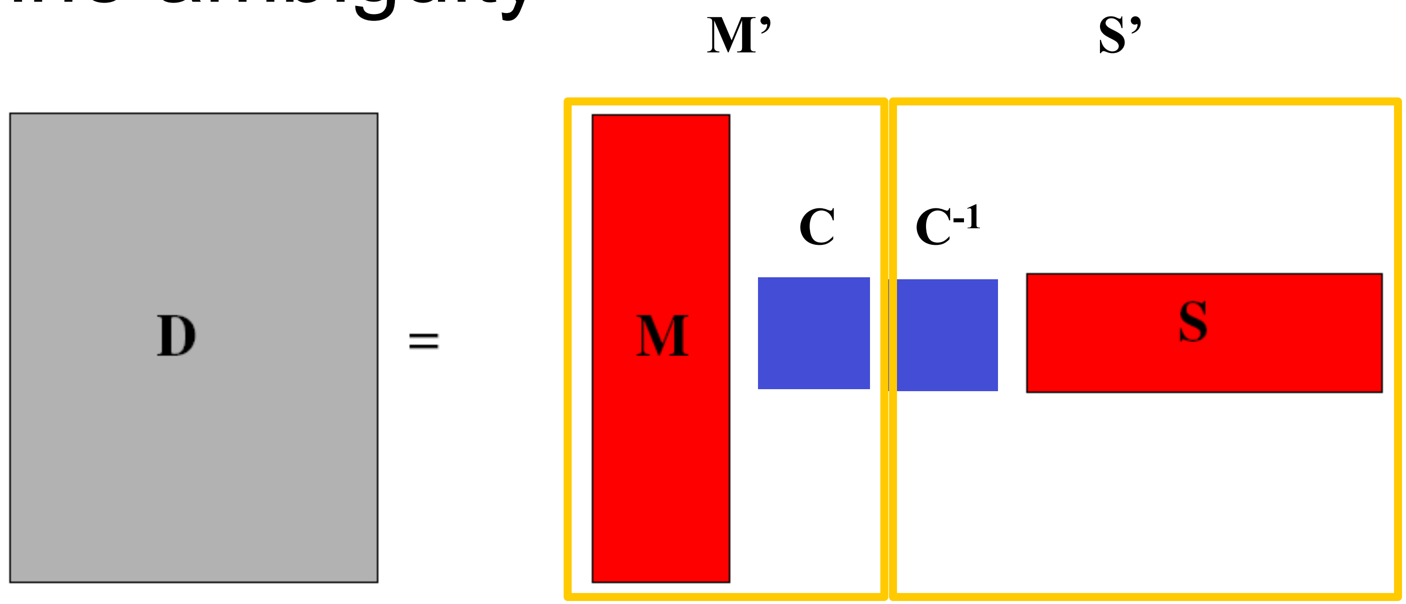


Theorem: When D has a rank greater than p, $\mathcal{U}_p\mathcal{W}_p\mathcal{V}_p^T$ is the best possible rank-p approximation of D in the sense of the Frobenius norm.

$$\mathcal{D} = \mathcal{U}_3\mathcal{W}_3\mathcal{V}_3^T$$

$$\boxed{\begin{cases} \mathcal{A}_0 = \mathcal{U}_3 \\ \mathcal{P}_0 = \mathcal{W}_3\mathcal{V}_3^T \end{cases}}$$

Affine ambiguity



The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations **M** → **MC**, **S** → $\mathbf{C}^{-1}\mathbf{S}$

We can enforce some Euclidean constraints to resolve this ambiguity (more on next lecture!)

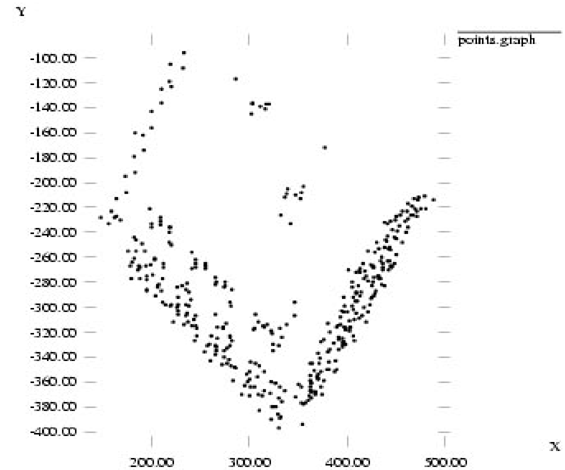
Algorithm summary

1. Given: *m* images and *n* features $\mathbf{x}_{ij}$
2. For each image *i, c*enter the feature coordinates
3. Construct a 2*m* × *n* measurement matrix **D**:
   - Column *j* contains the projection of point *j* in all views
   - Row *i* contains one coordinate of the projections of all the *n* points in image *i*
4. Factorize **D**:
   - Compute SVD: $\mathbf{D} = \mathbf{U}\ \mathbf{W}\ \mathbf{V}^{\mathbf{T}}$
   - Create $\mathbf{U}_3$ by taking the first 3 columns of **U**
   - Create $\mathbf{V}_3$ by taking the first 3 columns of **V**
   - Create $\mathbf{W}_3$ by taking the upper left 3 × 3 block of **W**
5. Create the motion and shape matrices:
   - $\mathbf{M} = \mathbf{M} = \mathbf{U}_3$ and $\mathbf{S} = \mathbf{W}_3\mathbf{V}_3^{T}$ (or $\mathbf{U}_3\mathbf{W}_3^{½}$ and $\mathbf{S} = \mathbf{W}_3^{½}\ \mathbf{V}_3^{T}$)
6. Eliminate affine ambiguity

Reconstruction results

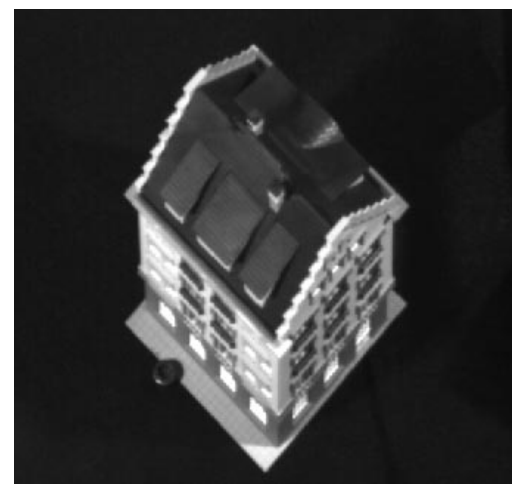
1

60

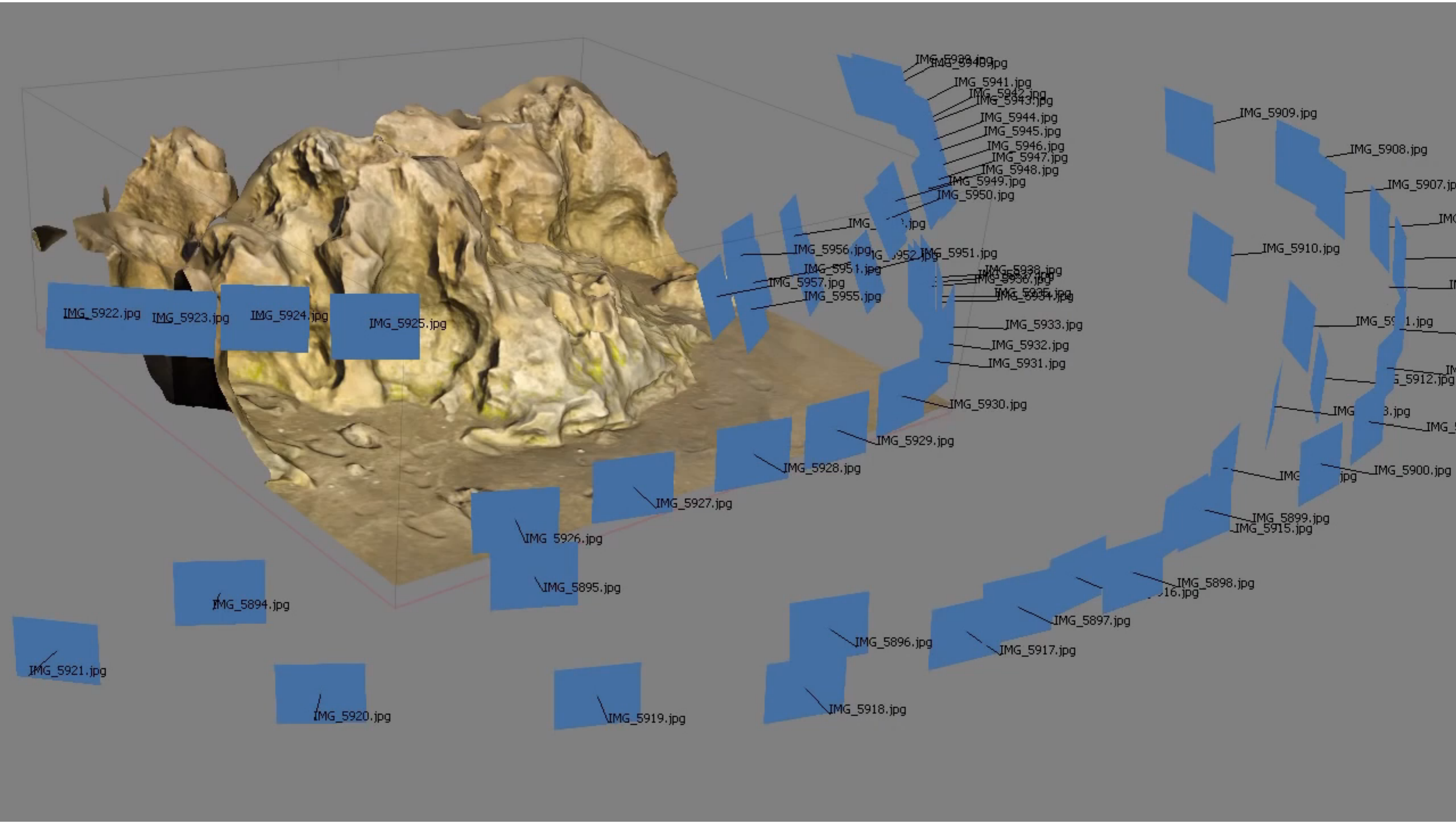


120

150



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


IMG_5941.jpg
IMG_5944.jpg
IMG_5945.jpg
IMG_5946.jpg
IMG_5947.jpg
IMG_5948.jpg
IMG_5949.jpg
IMG_5950.jpg
IMG_5909.jpg
IMG_5908.jpg
IMG_5956.jpg
IMG_5951.jpg
IMG_5910.jpg
IMG_5957.jpg
IMG_5955.jpg
IMG_5922.jpg
IMG_5923.jpg
IMG_5924.jpg
IMG_5925.jpg
IMG_5933.jpg
IMG_5932.jpg
IMG_5931.jpg
IMG_5930.jpg
IMG_5929.jpg
IMG_5928.jpg
IMG_5927.jpg
IMG_5926.jpg
IMG_5900.jpg
IMG_5899.jpg
IMG_5915.jpg
IMG_5895.jpg
IMG_5894.jpg
IMG_5898.jpg
IMG_5897.jpg
IMG_5896.jpg
IMG_5917.jpg
IMG_5921.jpg
IMG_5920.jpg
IMG_5919.jpg
IMG_5918.jpg

Next lecture

Multiple view geometry

Perspective structure from Motion