Structure from Motion

Outline

- Bundle Adjustment
- Ambguities in Reconstruction
- Affine Factorization
- Extensions

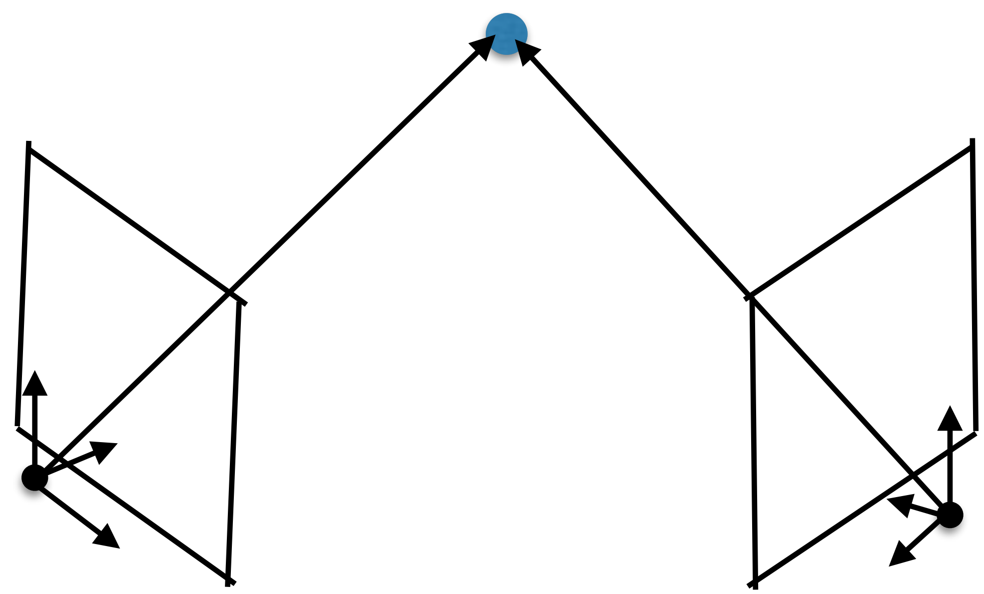
Structure from motion

Recover both 3D scene geometry and camera positions

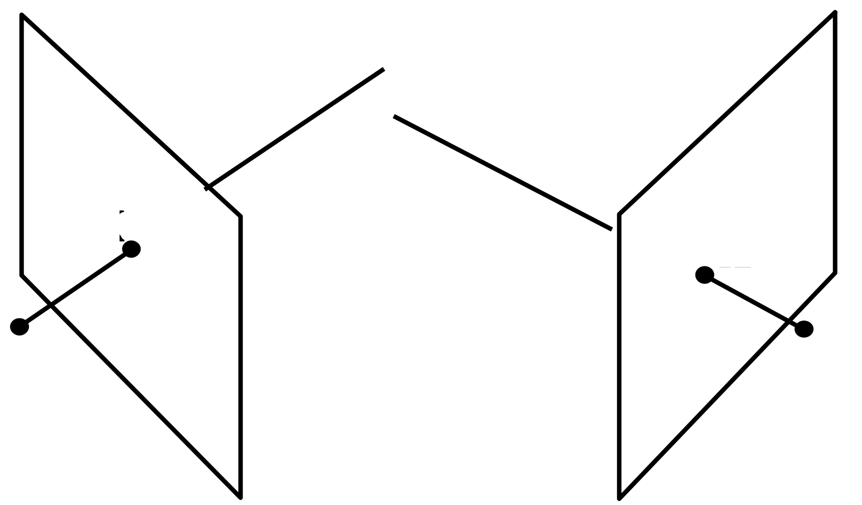


SLAM: Simultaneous Localization and Mapping

Recall: 2-view stereo

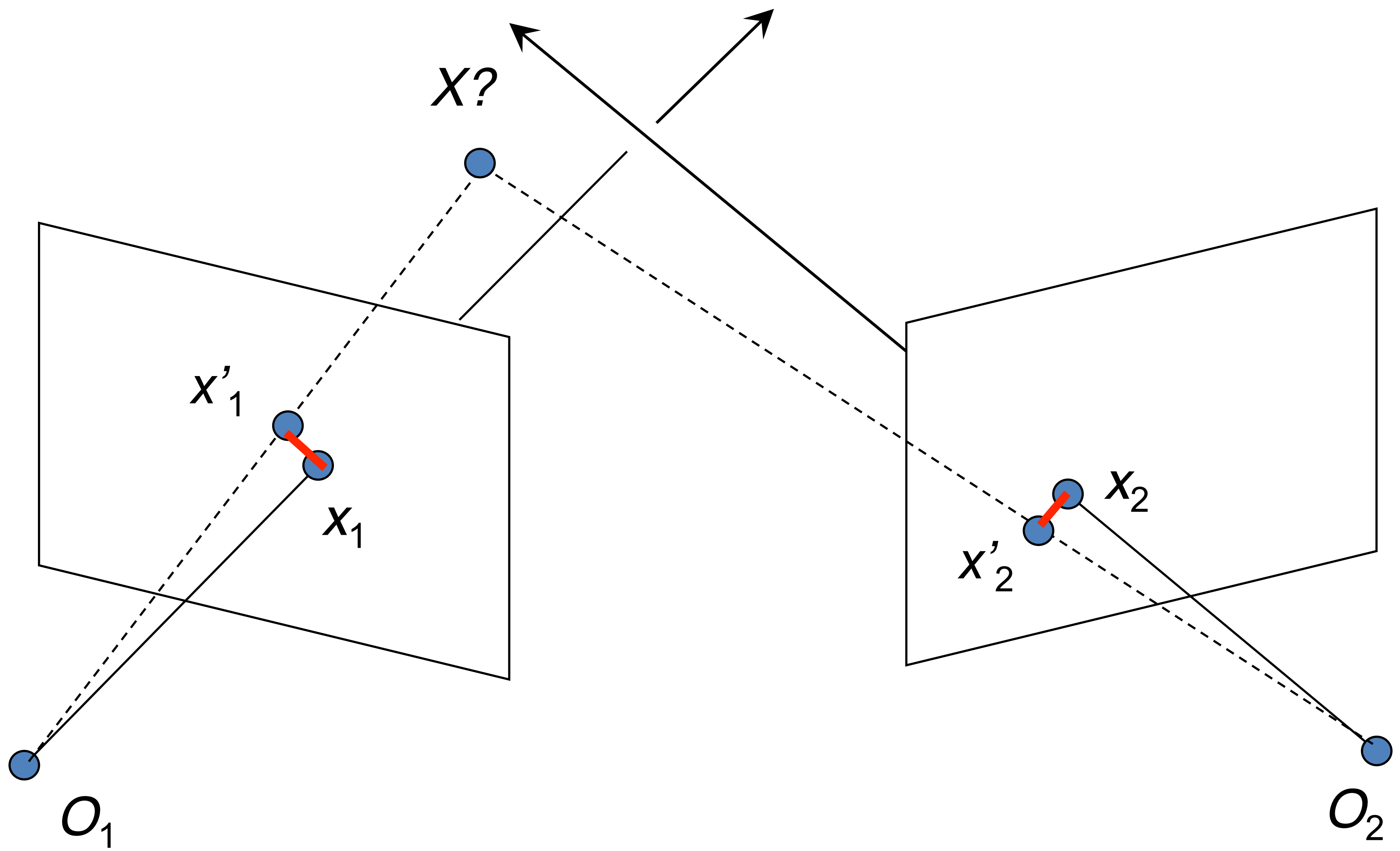


An annoying detail



What to do when ray's don't intersect?

Minimize reprojection error

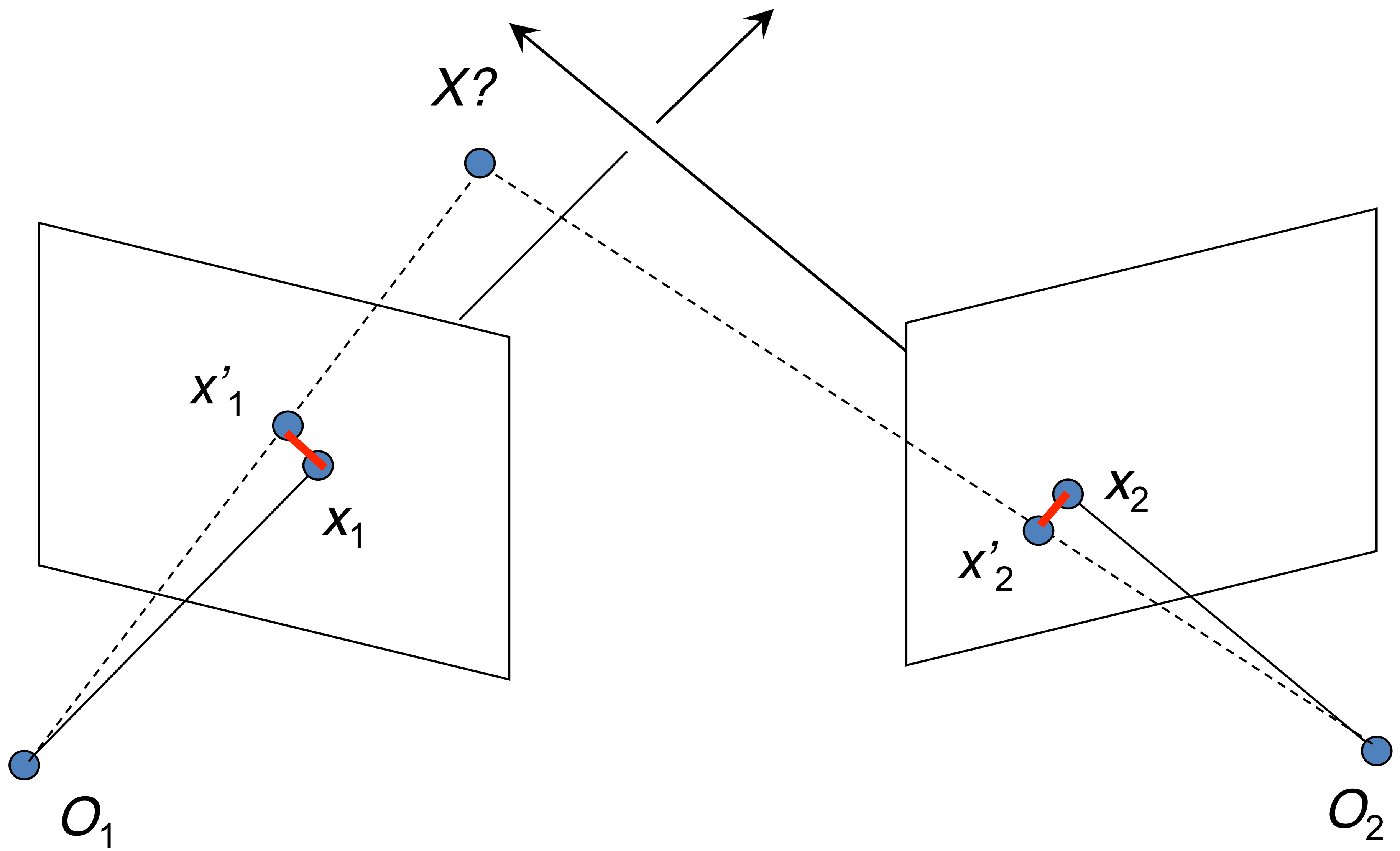


$$\min_{\mathbf{X}} f(\mathbf{X}) = ||\mathbf{x}_1 - Proj(\mathbf{X}, M_1)||^2 + ||\mathbf{x}_2 - Proj(\mathbf{X}, M_2)||^2$$

Perspective projection equations where M_1 = 3x4 matrix of extrinics (R_1,t_1) and intrinsics (f_1) of camera 1

For this equation, its easier to parameterize camera position in absolute coordinates instead of relative ones

Minimize reprojection error

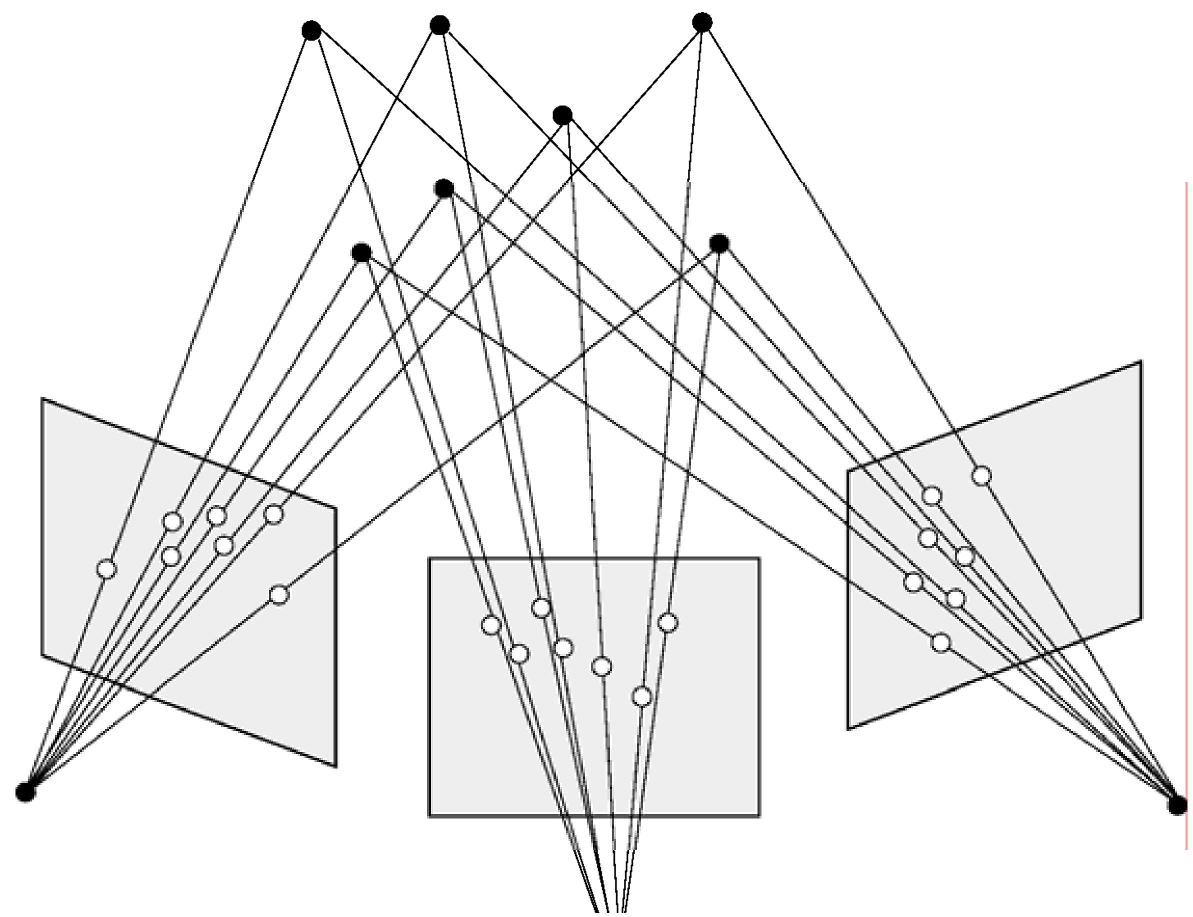


$$\min_{\mathbf{X}} f(\mathbf{X}) = ||\mathbf{x}_1 - Proj(\mathbf{X}, M_1)||^2 + ||\mathbf{x}_2 - Proj(\mathbf{X}, M_2)||^2$$

Perspective projection equations where M_1 = 3x4 matrix of extrinics (R_1,t_1) and intrinsics (f_1) of camera 1

For this equation, its easier to parameterize camera position in absolute coordinates instead of relative ones

Generalize triangulation to multiple points from multiple cameras

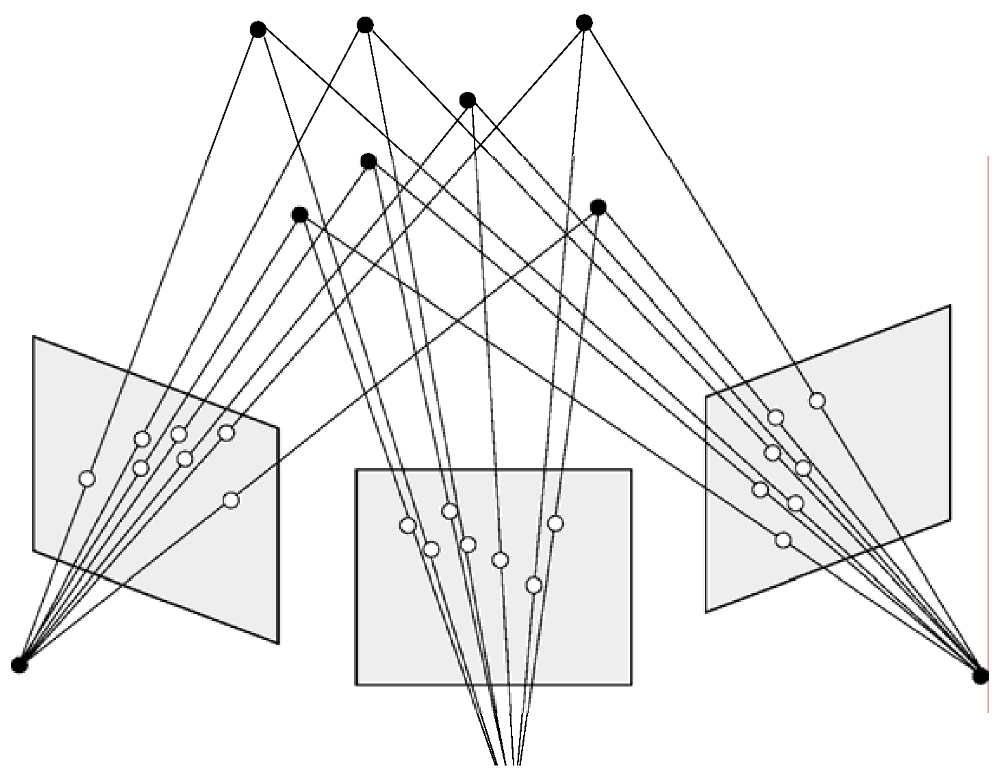


$$\min_{\mathbf{X}_1,\mathbf{X}_2,\ldots} \sum_{i=1}^{m} \sum_{j=1}^{n} ||\mathbf{x}_{ij} - Proj(\mathbf{X}_j, M_i)||^2$$

As written, could this minimization done independantly for each 3D point?

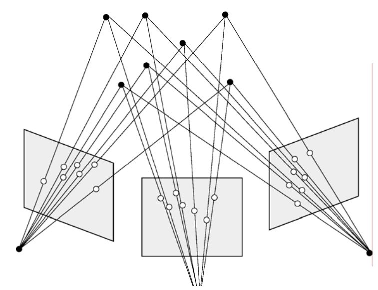
Bundle adjustment

Minimize reprojection error over multiple 3D points and cameras



$$\min_{\mathbf{X}_1,\mathbf{X}_2,\ldots,M_1,M_2,\ldots} \sum_{i=1}^{m} \sum_{j=1}^{n} ||\mathbf{x}_{ij} - Proj(\mathbf{X}_j, M_i)||^2$$

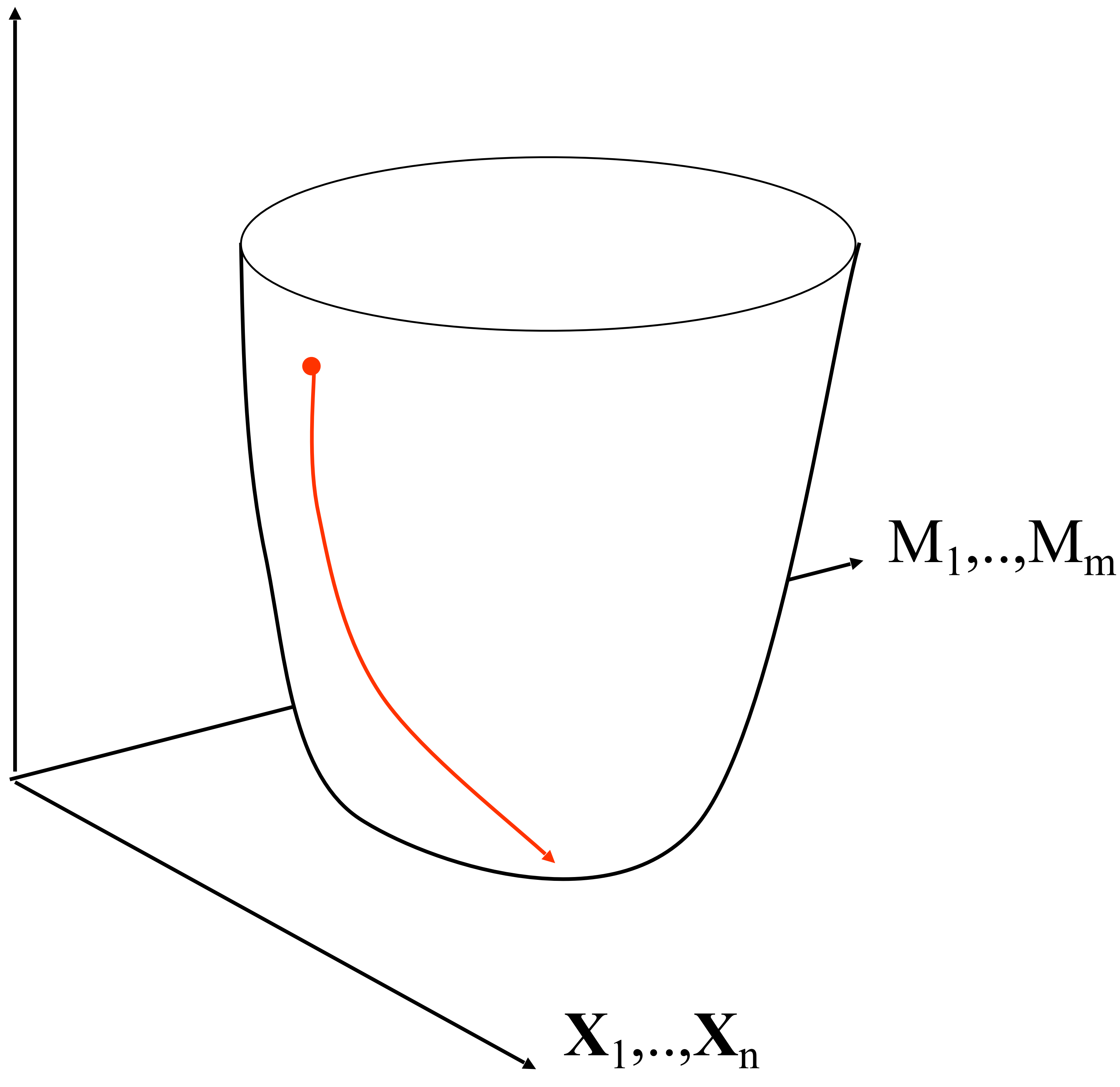
Basic SFM pipeline



1. Find candidate correspondences (interest points + descriptor matches)
2. Select subset that are consistent with epipolar constraints on pairs of images (RANSAC + fundmental matrix)
3. Solve for 3D points and camera that minimize reprojection error

(Lots of variants; e.g., iteratively build map of 3D points and cameras as new images arrive)

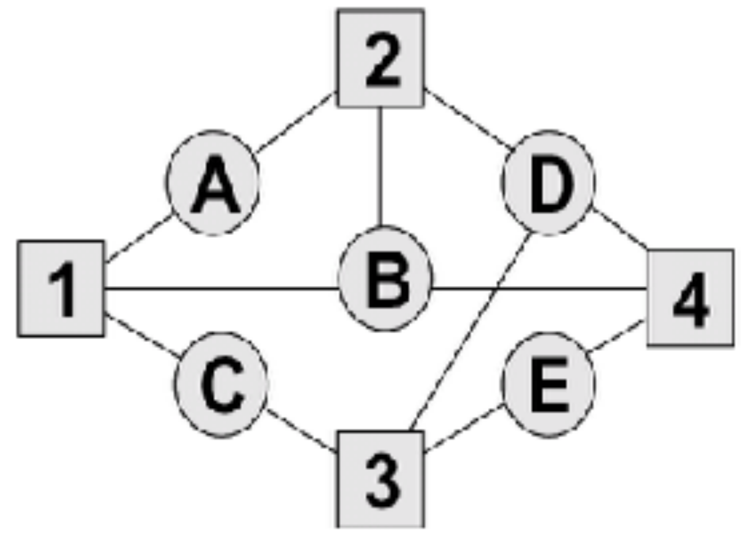
$$\min_{\mathbf{X}_1,\mathbf{X}_2,\ldots,M_1,M_2,\ldots} \sum_{i=1}^{m} \sum_{j=1}^{n} ||\mathbf{x}_{ij} - Proj(\mathbf{X}_j, M_i)||^2$$



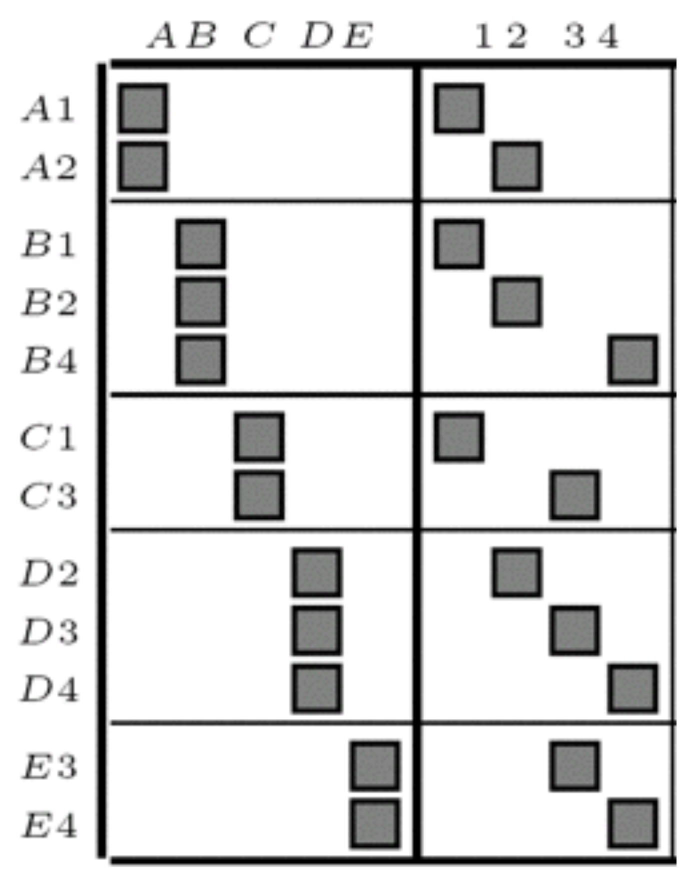
Bundle adjustment: nonlinear least-squares

Encode visibility graphs of what points are seen in what images, i.e.

- Features: A,B,C,D,E
- Images: 1,2,3



Implies jacobian of error function is sparse



Excellent reference

Bundle Adjustment — A Modern Synthesis

Bill Triggs[1], Philip McLauchlan[2], Richard Hartley[3] and Andrew Fitzgibbon[4]

[1] INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/people/ Triggs*

[2] School of Electrical Engineering, Information Technology & Mathematics
University of Surrey, Guildford, GU2 5XH, U.K.
P.McLauchlan@ee.surrey.ac.uk ◇ *http://www.ee.surrey.ac.uk/Personal/P.McLauchlan*

[3] General Electric CRD, Schenectady, NY, 12301
hartley@crd.ge.com

[4] Dept of Engineering Science, University of Oxford, 19 Parks Road, OX1 3PJ, U.K.
awf@robots.ox.ac.uk ◇ *http://www.robots.ox.ac.uk/ awf*

Outline

- Bundle Adjustment
- Ambguities in Reconstruction
- Affine Factorization
- Extensions

Recall camera projection

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= K_{3\times 3} \begin{bmatrix} R_{3\times 3} & T_{3\times 1} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= M_{3\times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{x} \equiv M\mathbf{X}$$

$$x = \frac{m_1^T X}{m_3^T X}$$

$$y = \frac{m_2^T X}{m_3^T X}$$

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 2D homogenous vector remains exactly the same:

$$\mathbf{x} \equiv M\mathbf{X}$$

$$M\mathbf{X} = (\frac{1}{k}M)(k\mathbf{X})$$

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

Structure from motion ambiguity

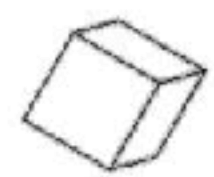
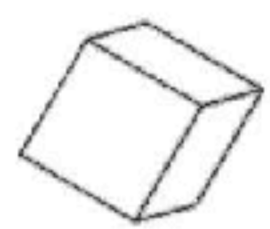
- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the 2D homogenous vector remains exactly the same:

- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, nothing changes

$$\mathbf{x} \equiv M\mathbf{X}$$

$$M\mathbf{X} = (MQ^{-1})(Q\mathbf{X})$$

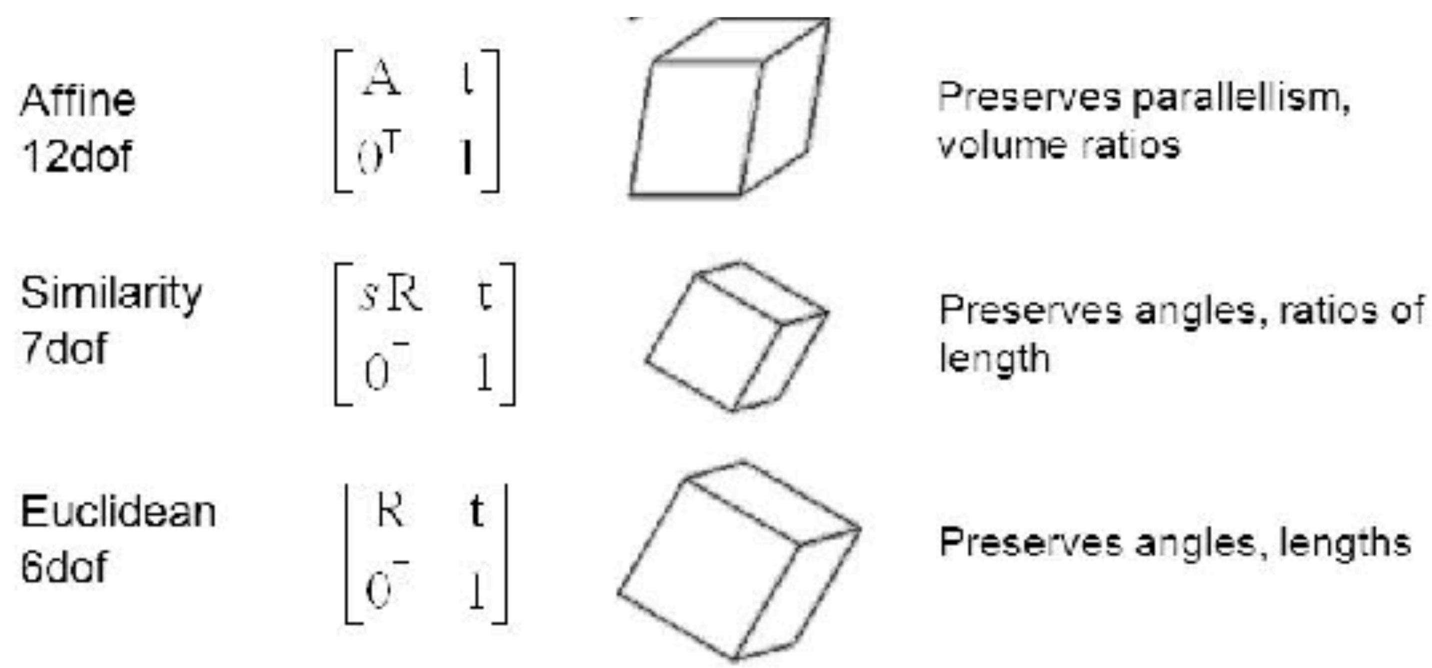
Recall: transformations in 3D

Similarity 7dof	$\begin{bmatrix} s\mathrm{R} & \mathrm{t} \\ 0^{\top} & 1 \end{bmatrix}$		Preserves angles, ratios of length
Euclidean 6dof	$\begin{bmatrix} \mathrm{R} & \mathbf{t} \\ 0^{\top} & 1 \end{bmatrix}$		Preserves angles, lengths

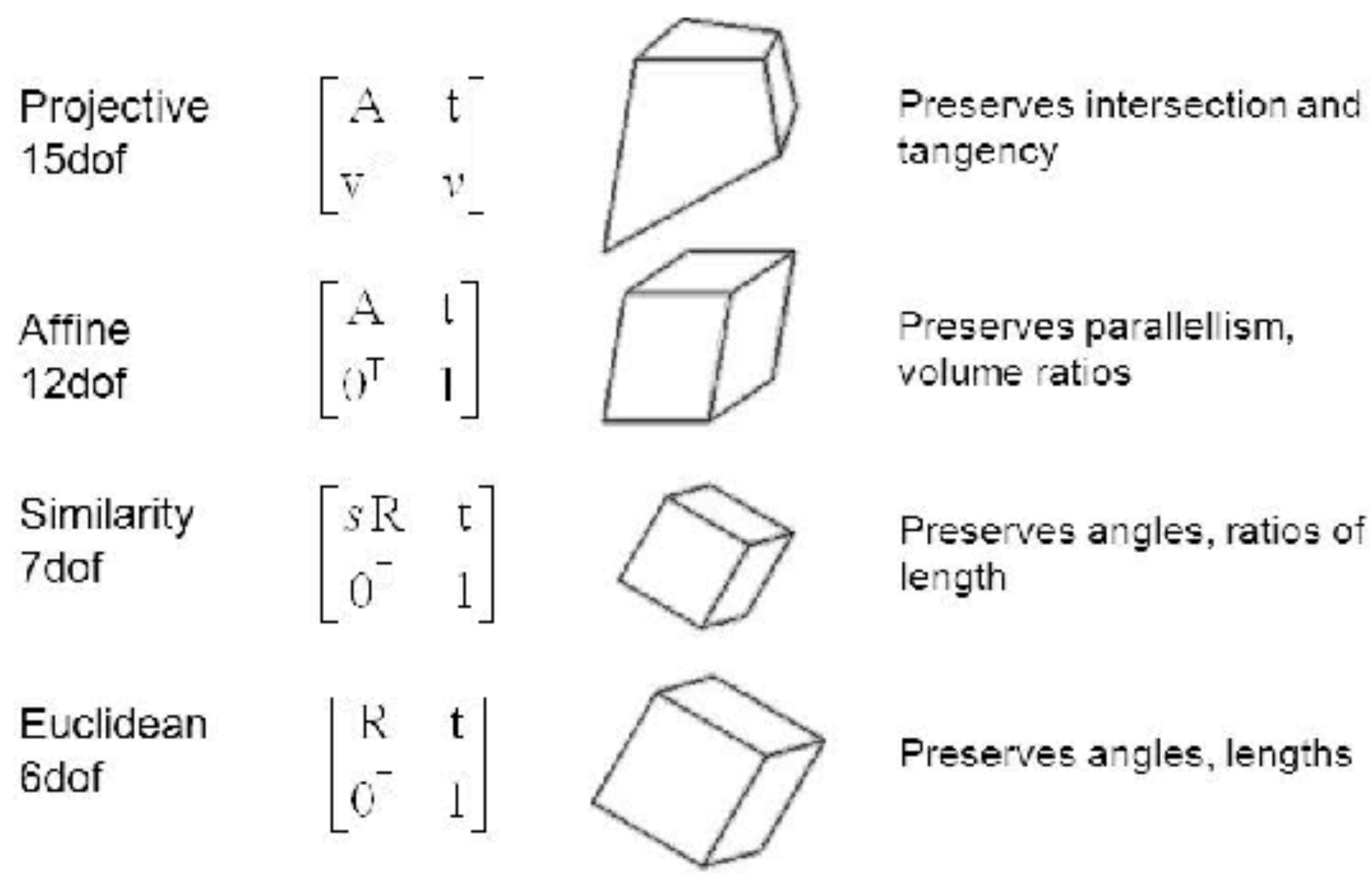
Make use of 3D homogenous coordinates

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Recall: transformations in 3D



Recall: transformations in 3D



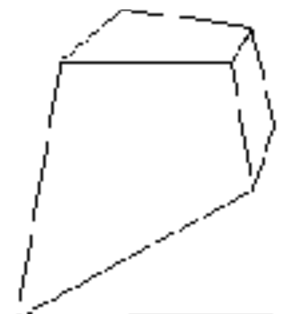
Normalize by last coordinate to recover 3D points

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda z \\ \lambda \end{bmatrix}$$

Back to ambiguities for 3D reconstruction

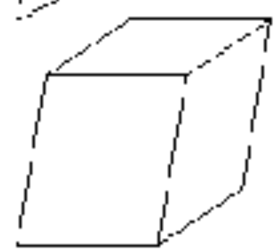
$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

Projective $\begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^\mathrm{T} & v \end{bmatrix}$



Preserves intersection

Affine $\begin{bmatrix} \mathrm{A} & \mathrm{t} \\ 0^\mathrm{T} & 1 \end{bmatrix}$

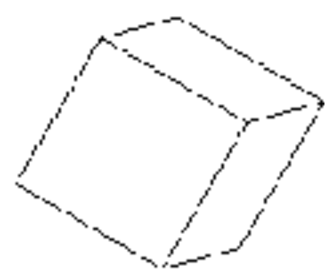


Preserves parallellism,

Similarity $\begin{bmatrix} s\,\mathrm{R} & \mathrm{t} \\ 0^\mathrm{T} & 1 \end{bmatrix}$

Preserves angles, ratios

Euclidean $\begin{bmatrix} \mathrm{R} & \mathrm{t} \\ 0^\mathrm{T} & 1 \end{bmatrix}$

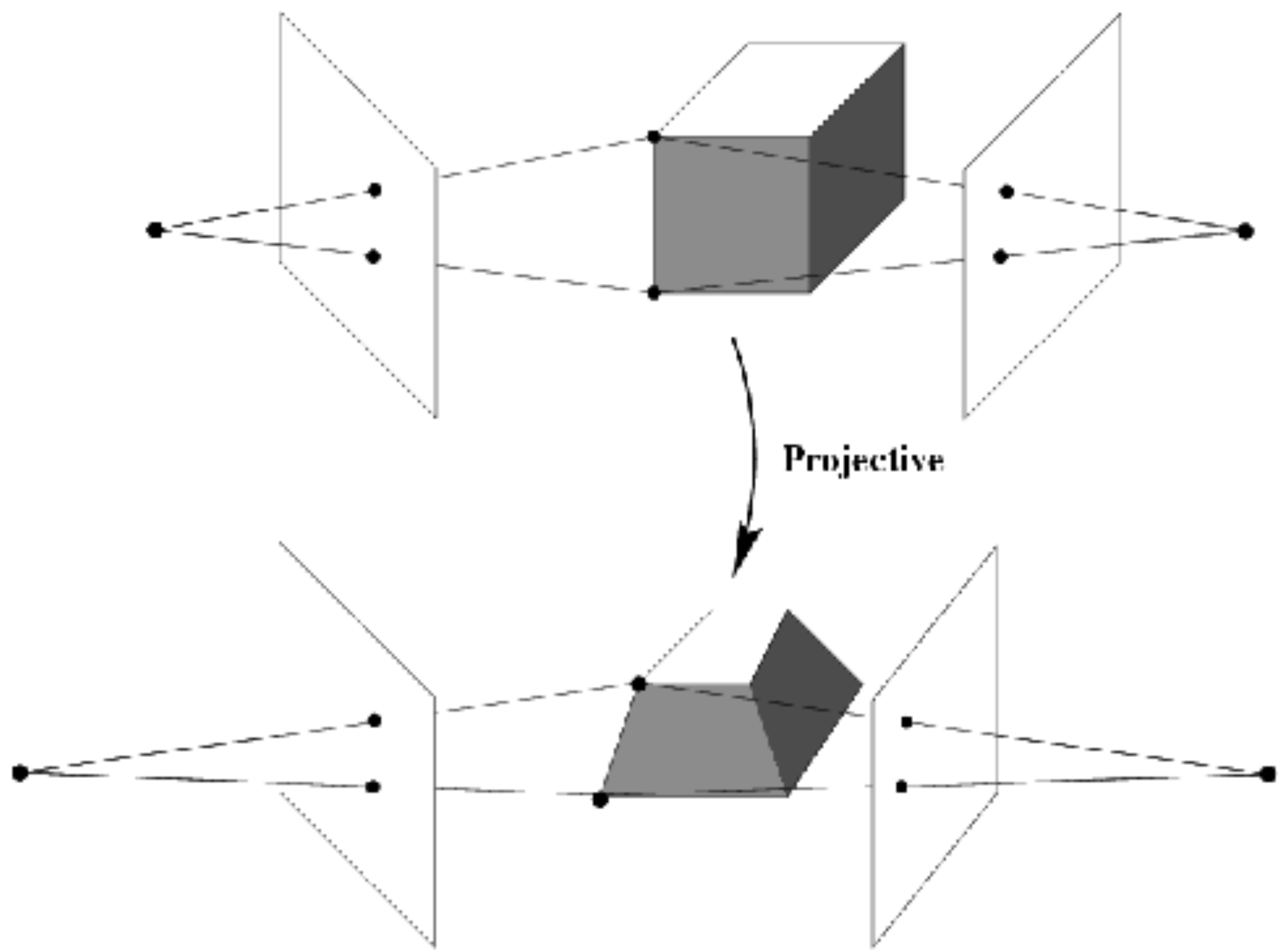


Preserves angles.

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean

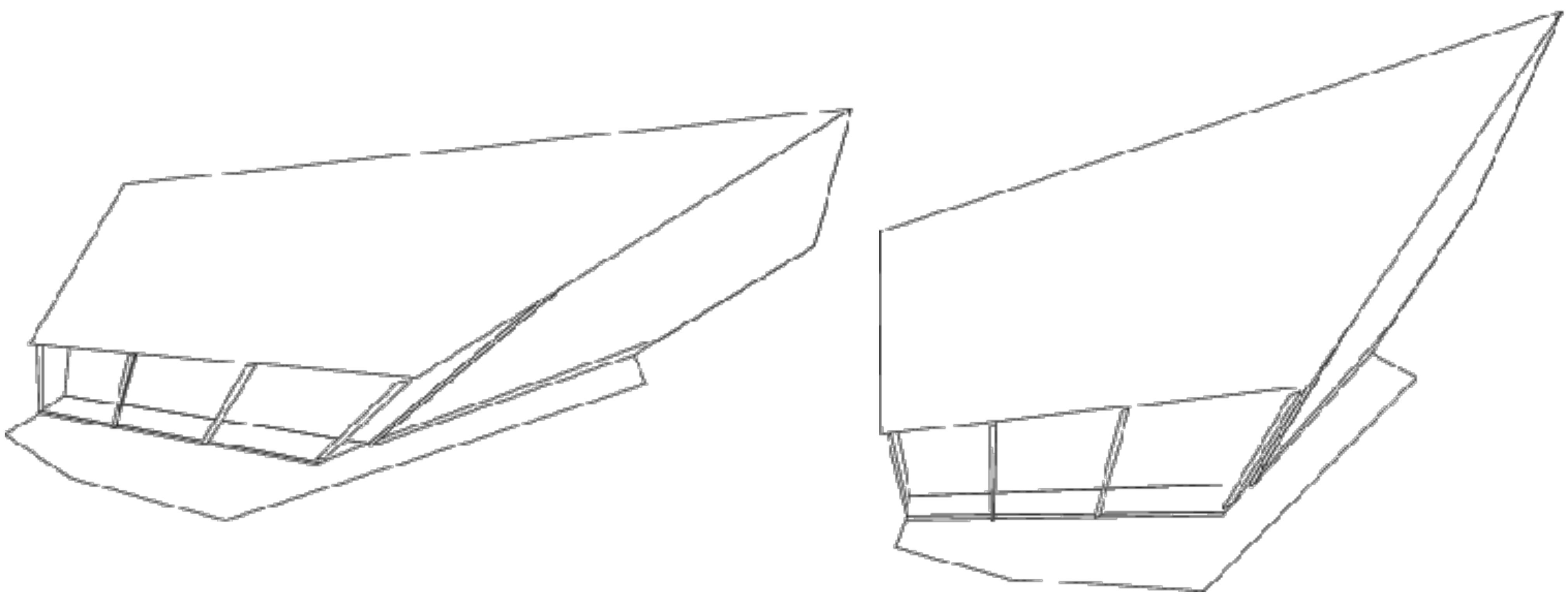
Projective ambiguity

$$x \equiv M\mathbf{X} = (MQ^{-1})(Q\mathbf{X})$$



$$\mathbf{Q_p} = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^{\mathsf{T}} & v \end{bmatrix}$$

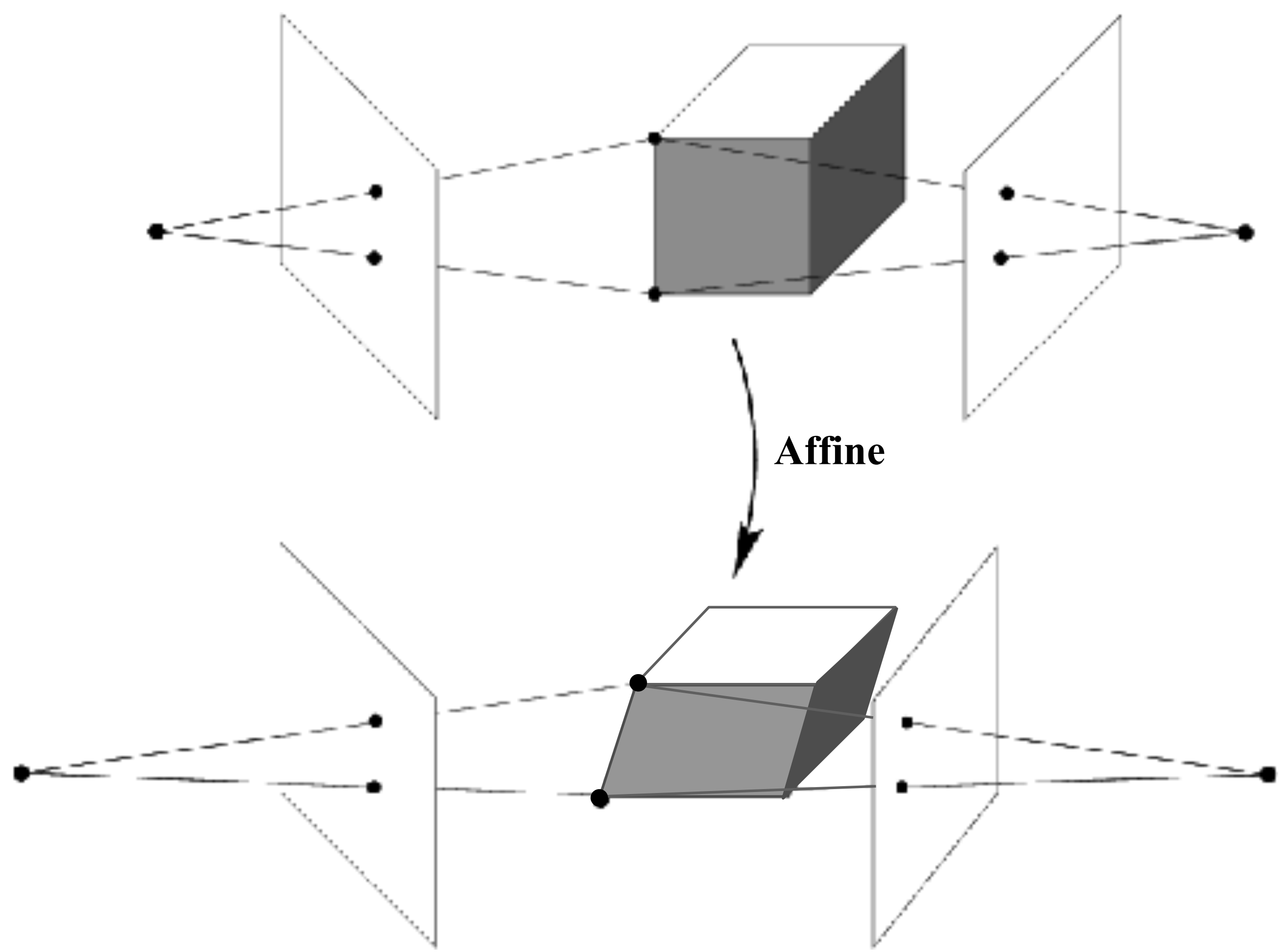
Projective ambiguity



(straight line are preserved)

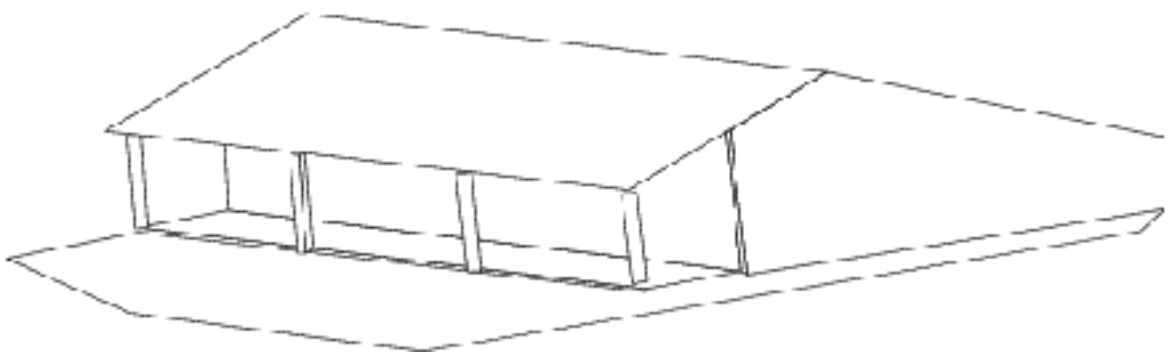
Affine ambiguity

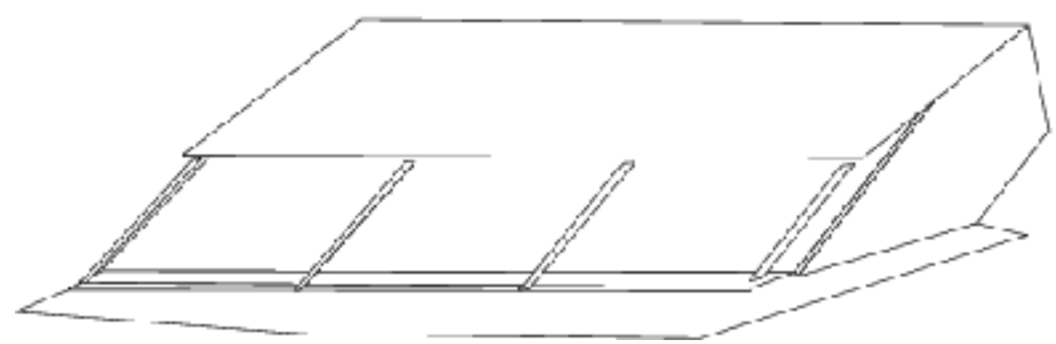
$$x \equiv M\mathbf{X} = (MQ^{-1})(Q\mathbf{X})$$



$$\mathbf{Q}_{\mathbf{A}} = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$$

Affine ambiguity

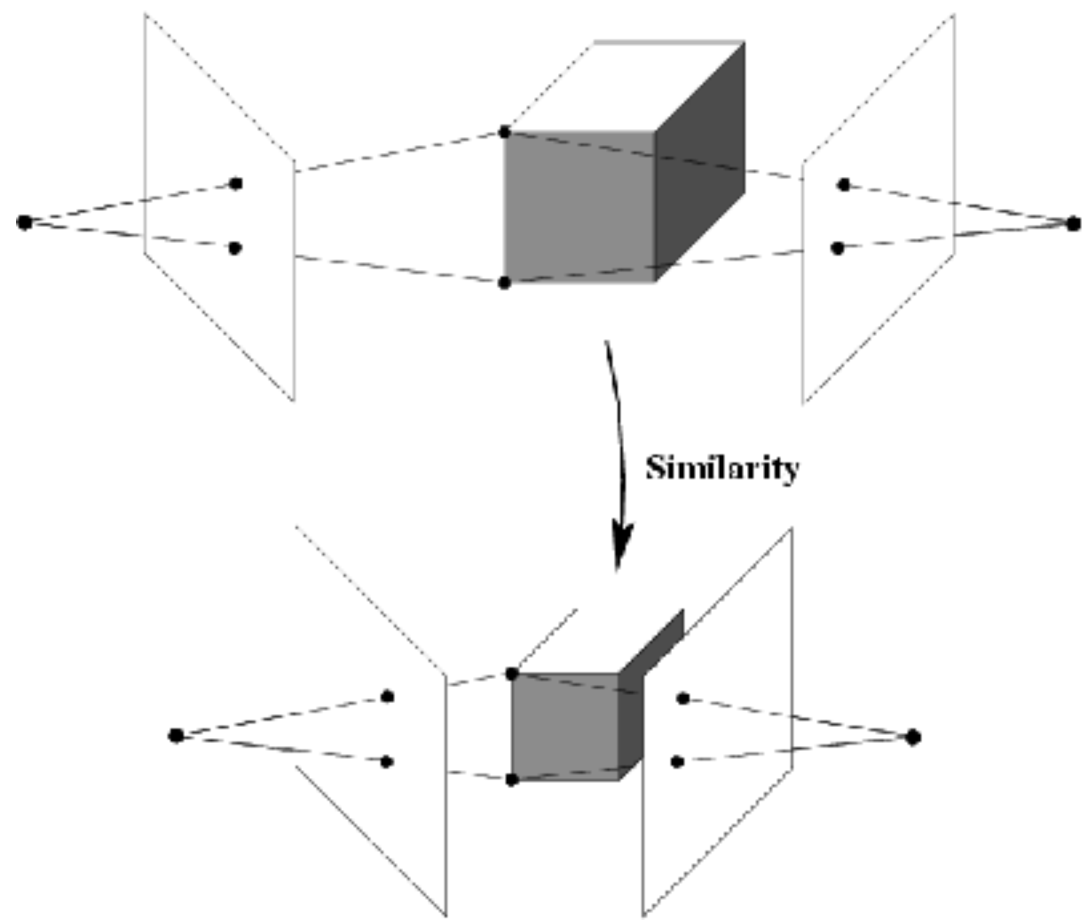




(parallel lines are preserved)

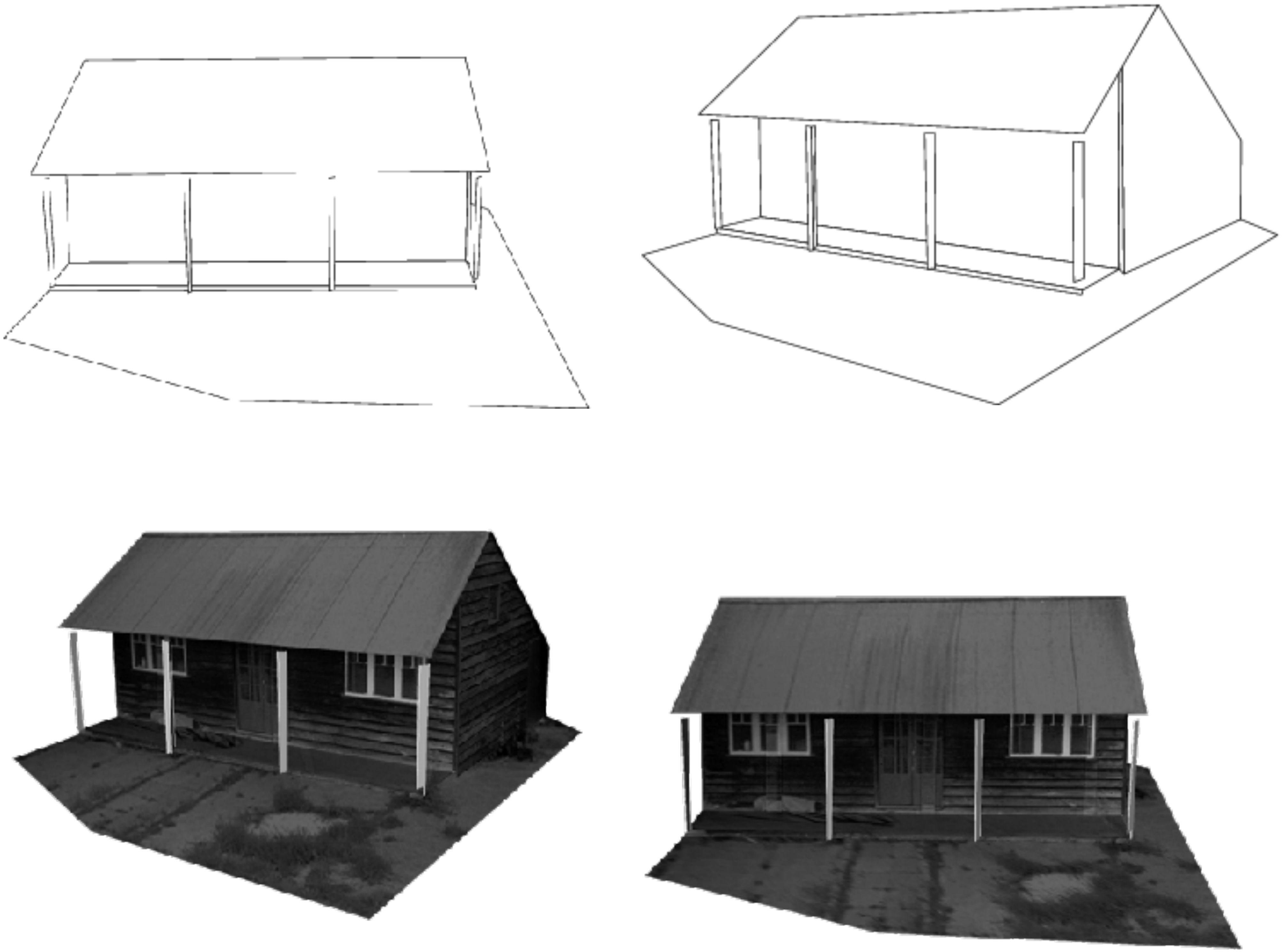
Similarity ambiguity

$$x \equiv M\mathbf{X} = (MQ^{-1})(Q\mathbf{X})$$



$$\mathbf{Q}_s = \begin{bmatrix} s\mathrm{R} & \mathrm{t} \\ 0^{\mathsf{T}} & 1 \end{bmatrix}$$

Similarity ambiguity



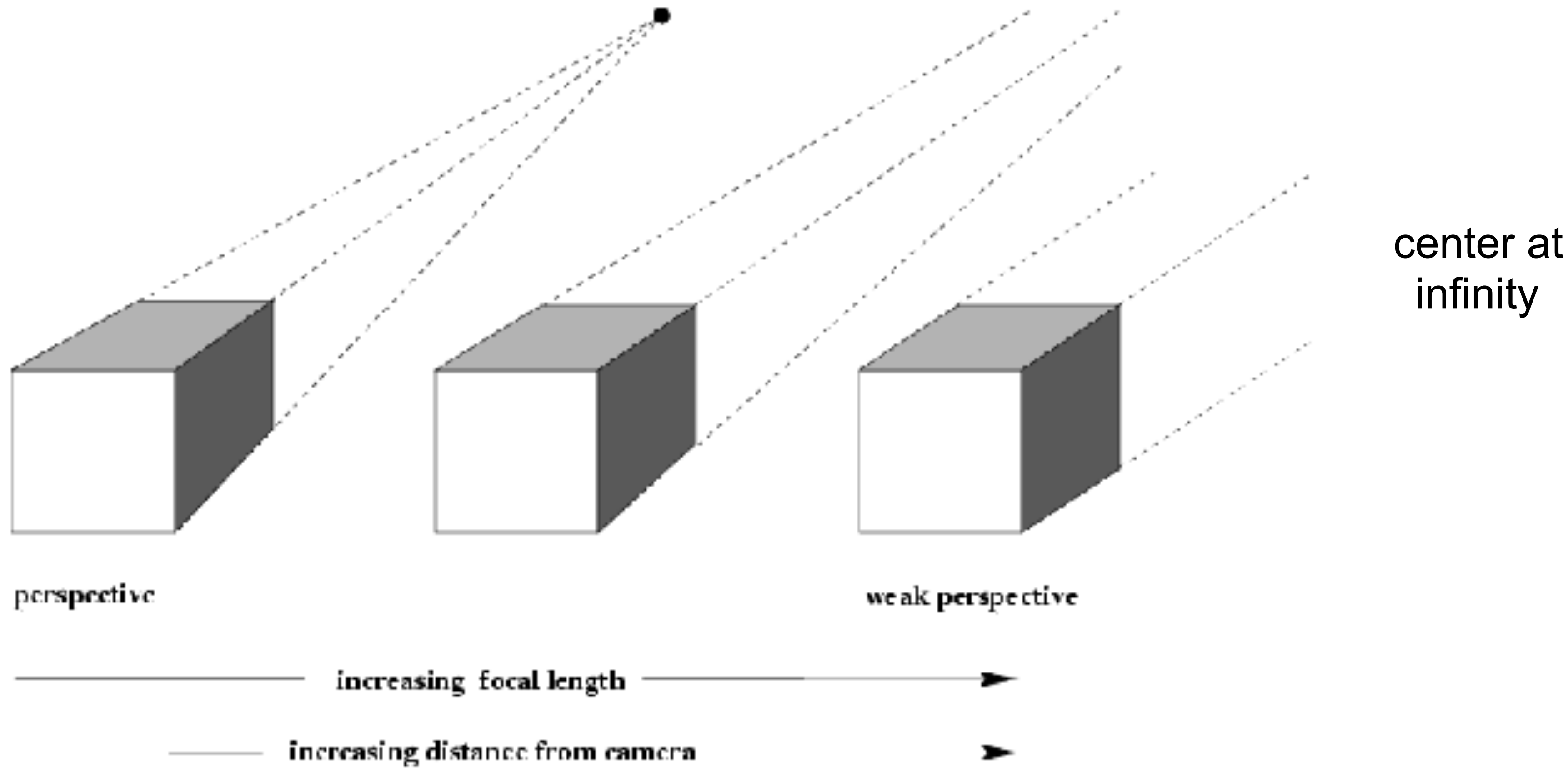
angles+lengths preserved (but can’t recover world coordinate system)

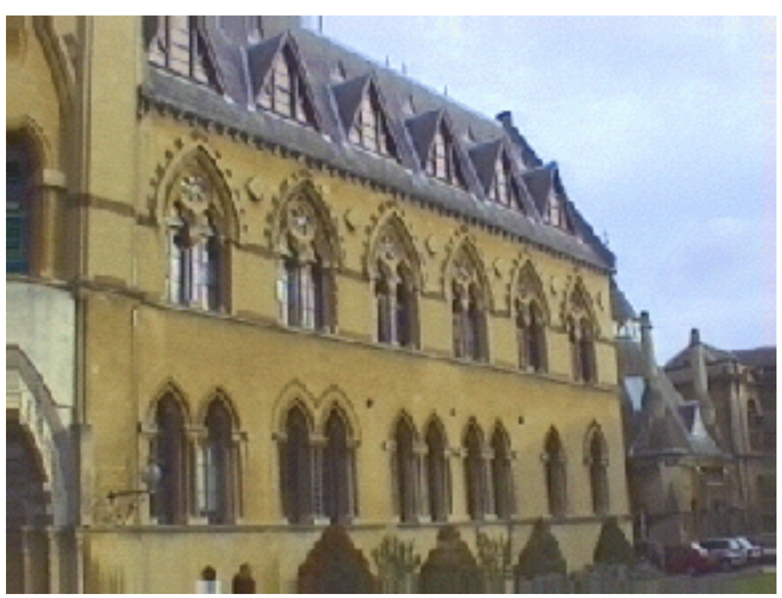
Outline

- Bundle Adjustment
- Ambguities in Reconstruction
- Affine Factorization
- Large-scale SFM

Structure from motion

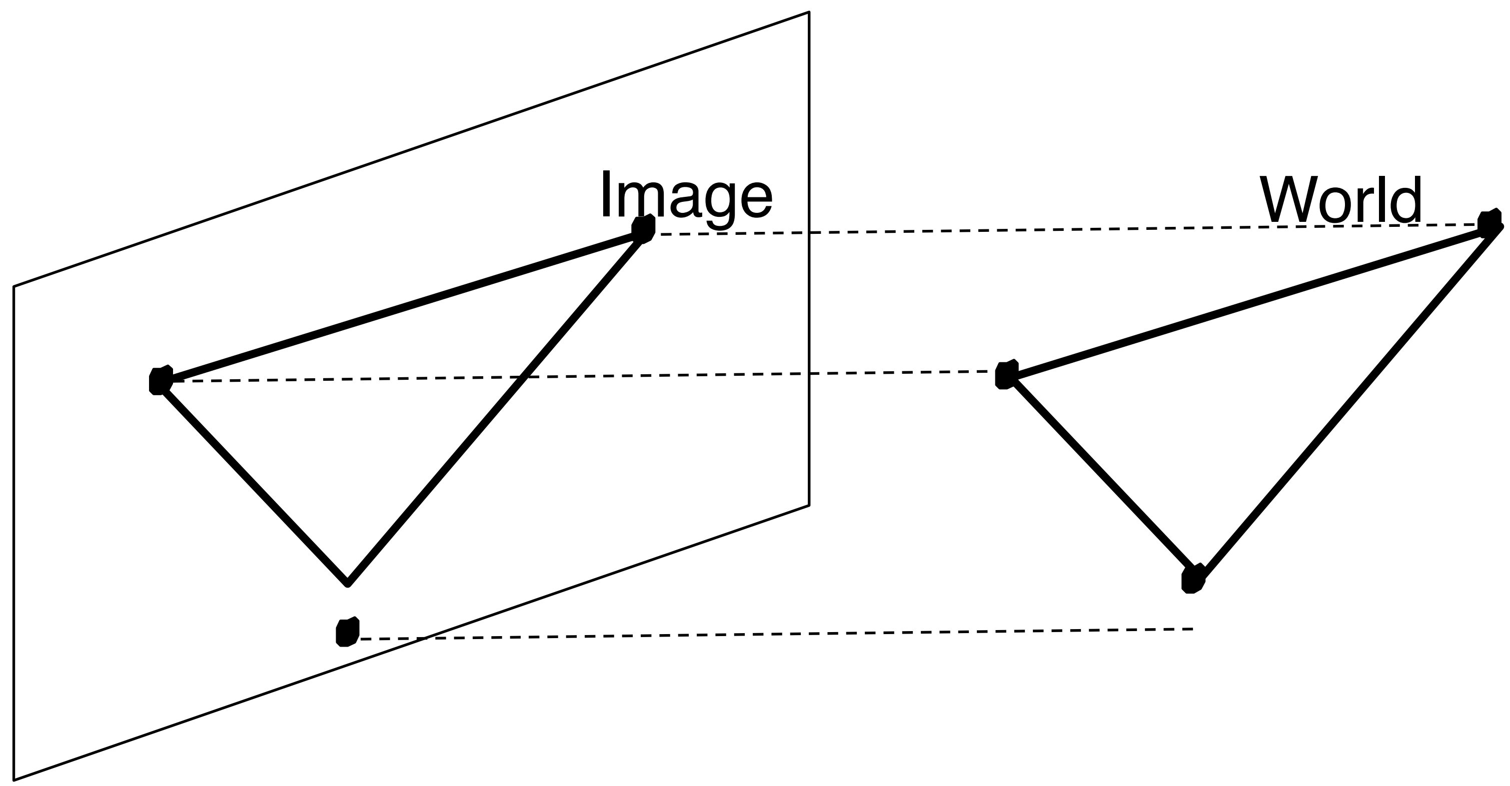
- Let's start with *affine cameras* (the math is easier)





Recall: Orthographic Projection

Special case of perspective projection



- Another common notation for a projection matrix:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Sometimes notationally convenient because 2D homogenous coordinates allow us to write 2D transformations as matrix multiplication

Recall: affine cameras

Model as 3D affine transformation + orthographic projection + 2D affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Affine camera defined by 8 parameters

Affine cameras

Model as 3D affine transformation + orthographic projection + 2D affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ & & & 1 \end{bmatrix} Q_a Q_a^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad Q_a = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & 1 \end{bmatrix}$$

Q_a must be 3D affine transformation (12 parameters) inorder to maintain structure of affine projection matrix

Affine cameras

Model as 3D affine transformation + orthographic projection + 2D affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ & & & 1 \end{bmatrix} Q_a Q_a^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad Q_a = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{A}\mathbf{X} + \mathbf{b}$$

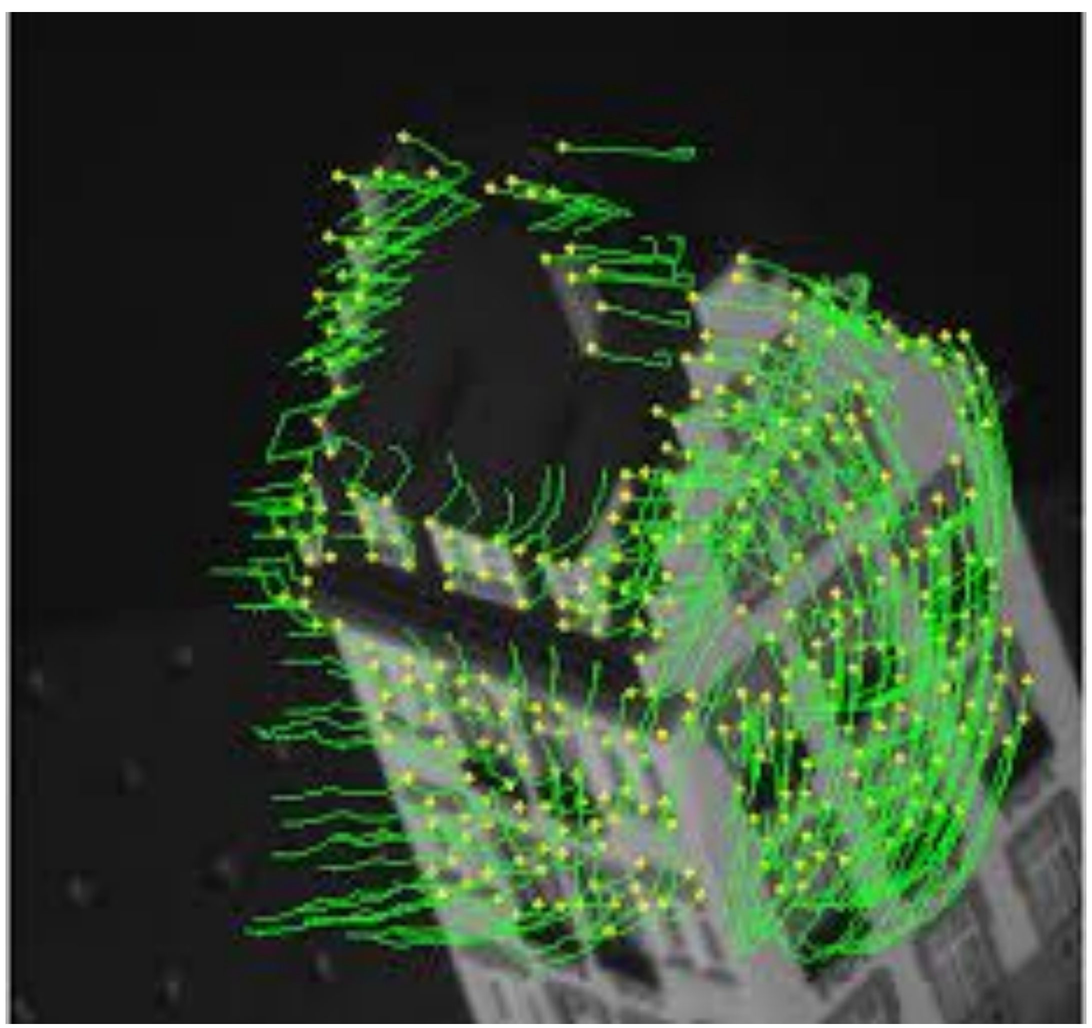
2D points = linear transformation of 3D points + 2D translation

Affine structure from motion (my *favorite* vision algorithm)

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

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

- Problem: use the *mn* correspondences $\mathbf{x}_{ij}$ to estimate *m* projection matrices $\mathbf{A}_i$ and translation vectors $\mathbf{b}_i$, and *n* points $\mathbf{X}_j$



$$\begin{array}{cc} & \text{\# of points} \longrightarrow \\ \text{\# of images} \downarrow & \begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$

Affine structure from motion

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

$$\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i, \qquad i = 1, \dots , m, \; j = 1, \dots , n$$

- Problem: use the *mn* correspondences $\mathbf{x}_{ij}$ to estimate *m* projection matrices $\mathbf{A}_i$ and translation vectors $\mathbf{b}_i$, and *n* points $\mathbf{X}_j$
- The reconstruction is defined up to an arbitrary 3D *affine* transformation **Q** (12 degrees of freedom):

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{Q}^{-1}, \qquad \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \rightarrow \mathbf{Q} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

- How many knowns are there?
- How many unknowns?

Affine structure from motion

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

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

- Problem: use the *mn* correspondences $\mathbf{x}_{ij}$ to estimate *m* projection matrices $\mathbf{A}_i$ and translation vectors $\mathbf{b}_i$, and *n* points $\mathbf{X}_j$
- The reconstruction is defined up to an arbitrary 3D *affine* transformation $\mathbf{Q}$ (12 degrees of freedom):

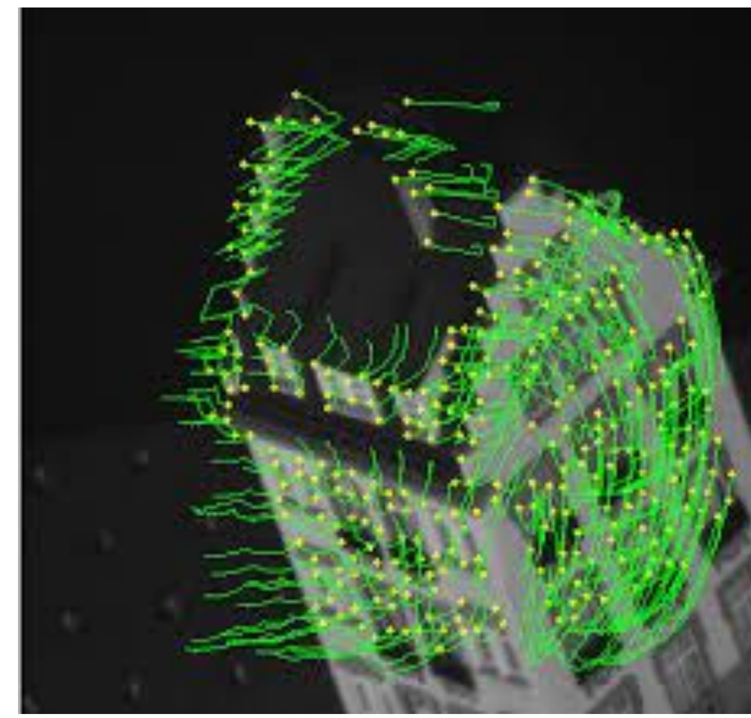
$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{Q}^{-1}, \qquad \begin{pmatrix} \mathbf{X} \\ \mathbf{1} \end{pmatrix} \rightarrow \mathbf{Q} \begin{pmatrix} \mathbf{X} \\ \mathbf{1} \end{pmatrix}$$

- We have 2*mn* knowns and 8*m* + 3*n* unknowns (minus 12 dof for affine ambiguity)
- Thus, we must have 2*mn* >= 8*m* + 3*n* – 12
- For m=2 views, we need n=4 point correspondences

Affine structure from motion

Let's try centering the points in each 2D image

$$\begin{array}{cc} & \text{\# of points} \longrightarrow \\ \begin{array}{c}\text{\# of}\\ \text{images}\end{array}\Big\downarrow & \begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$



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

Plug the following into the above expression: $\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i$

[on board]

Affine structure from motion

- Centering: subtract the centroid of the image points

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

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

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

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

Given a set of 2D correspondences, simply center them in each image
Affine image projection now becomes linear (represent A_i by a 2x3 matrix and $\mathbf{x}\hat{}_{ij}$ is 2 vector)

Affine structure from motion

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

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

cameras ($2m$)

points (n)

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

Affine structure from motion

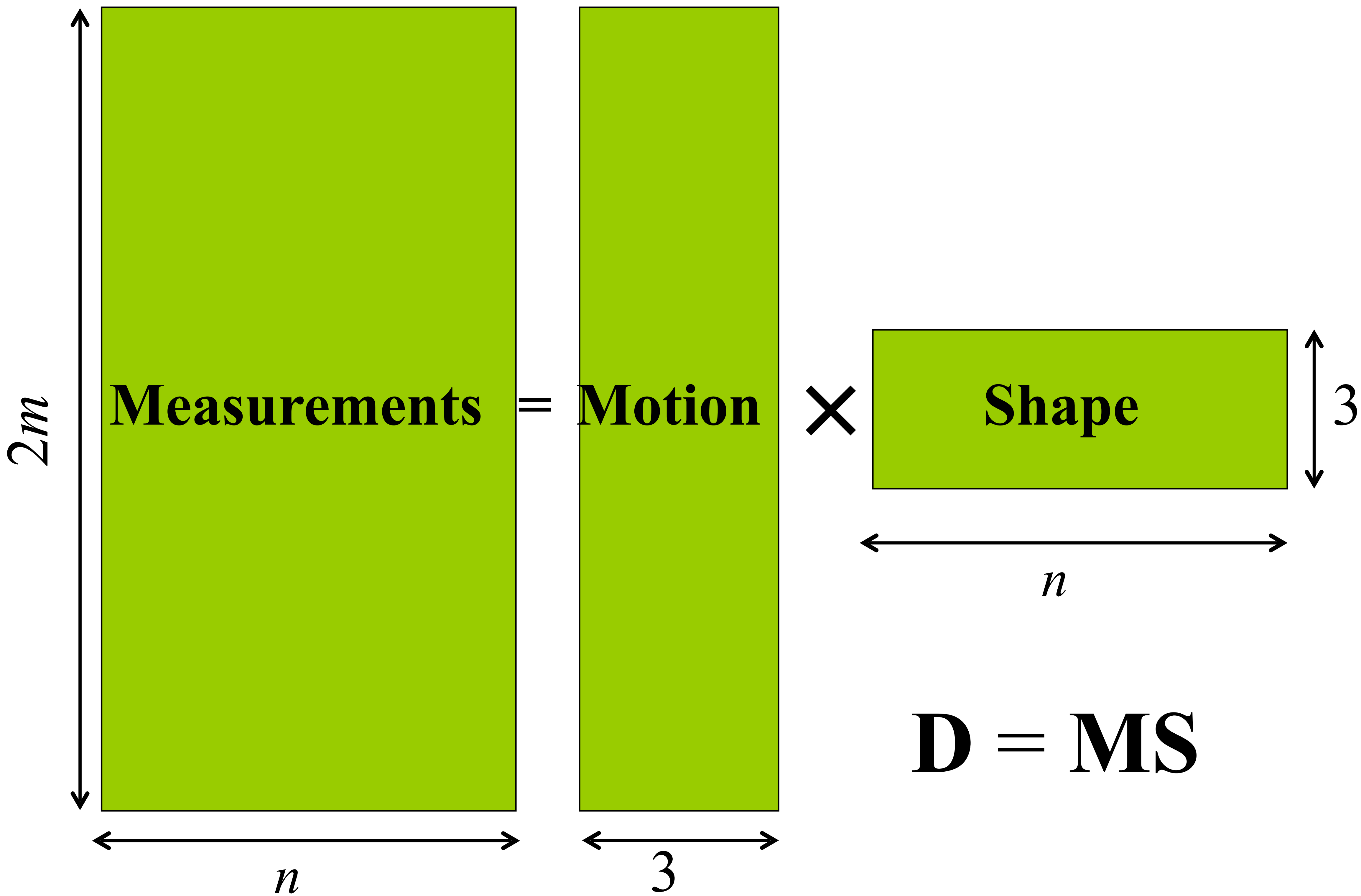
- Let's create a $2m \times n$ data (measurement) matrix:

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

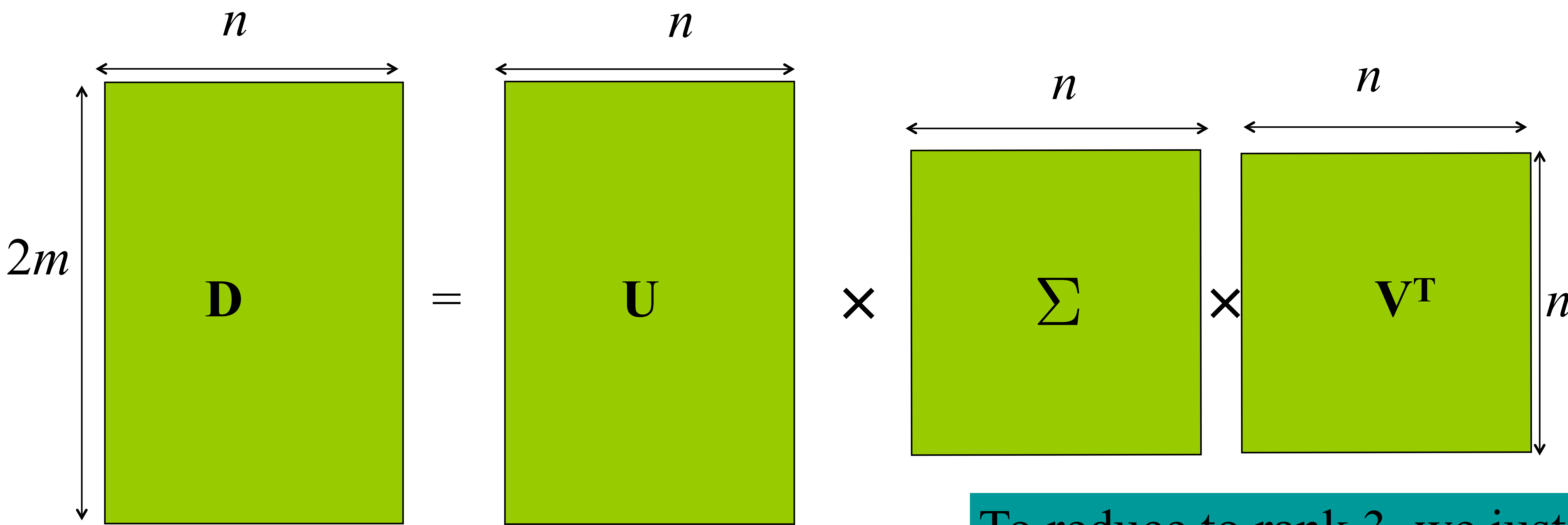
The measurement matrix $\mathbf{D} = \mathbf{MS}$ must have rank 3!

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

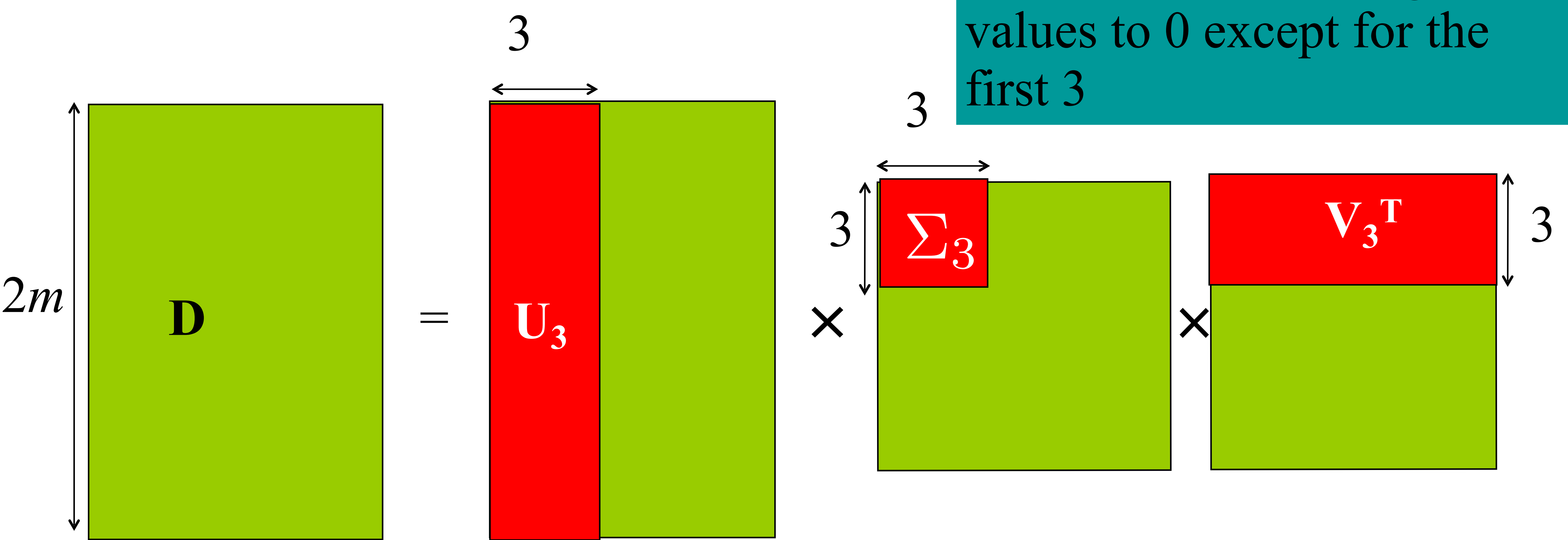
Fundamental Decomposition

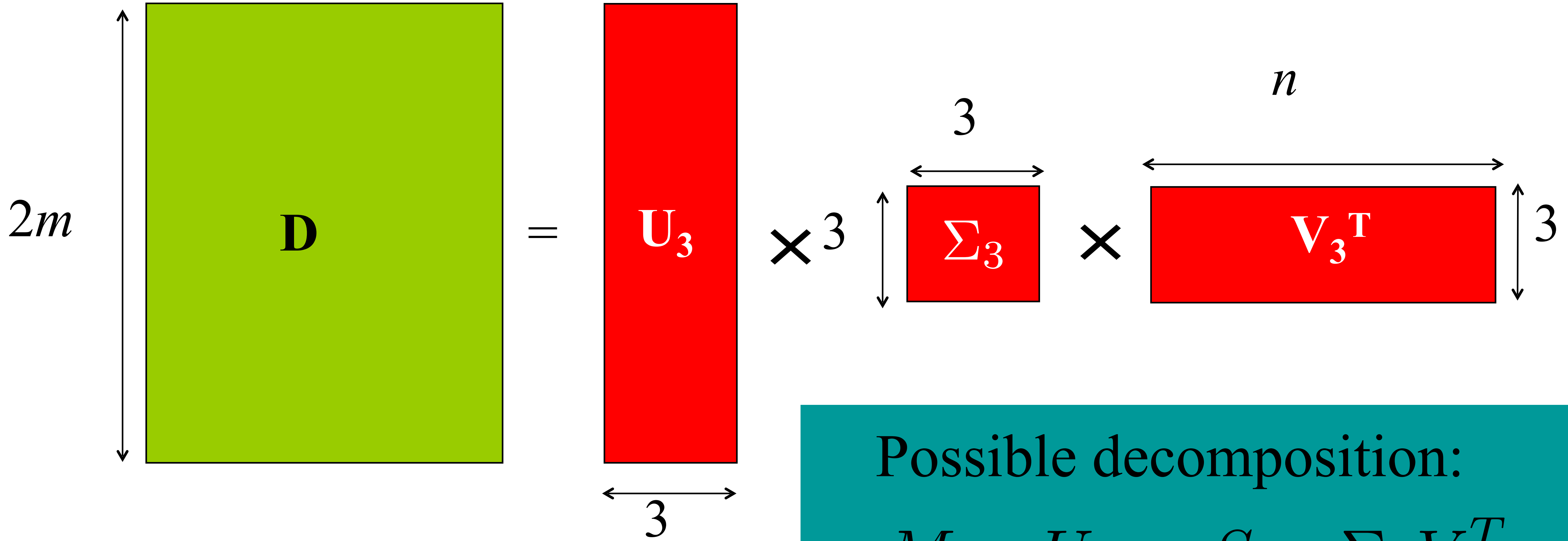


$$\mathbf{D} = \mathbf{M}\mathbf{S}$$



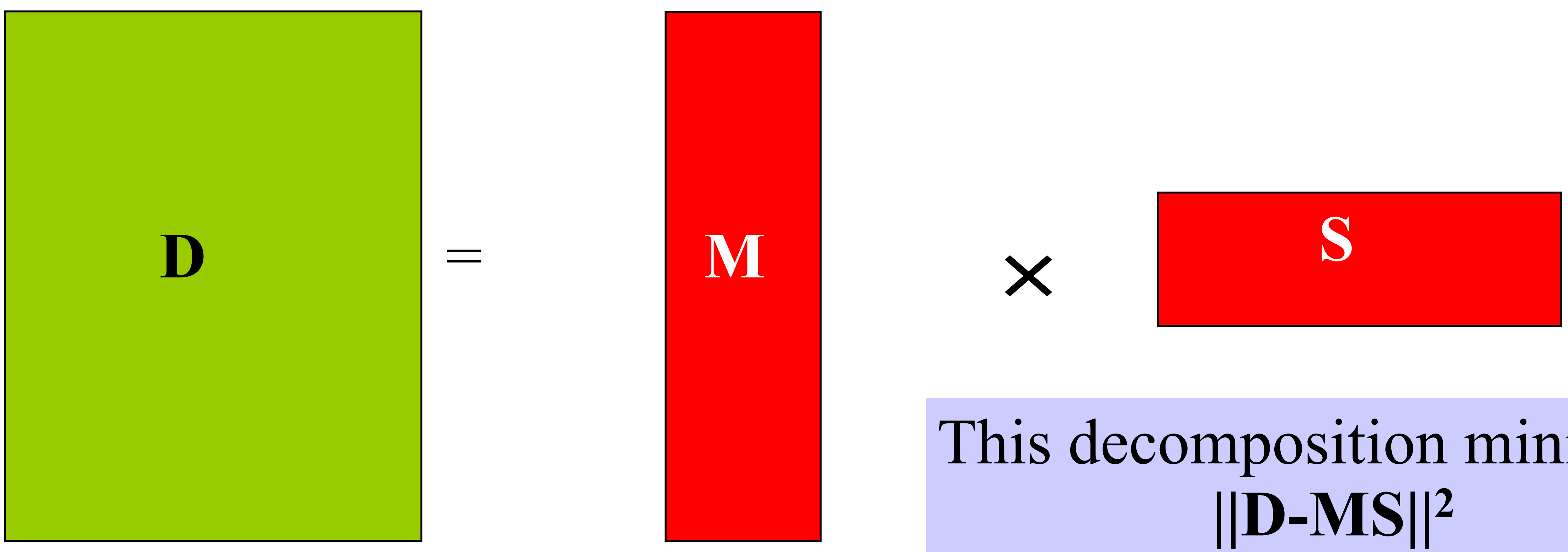
To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3





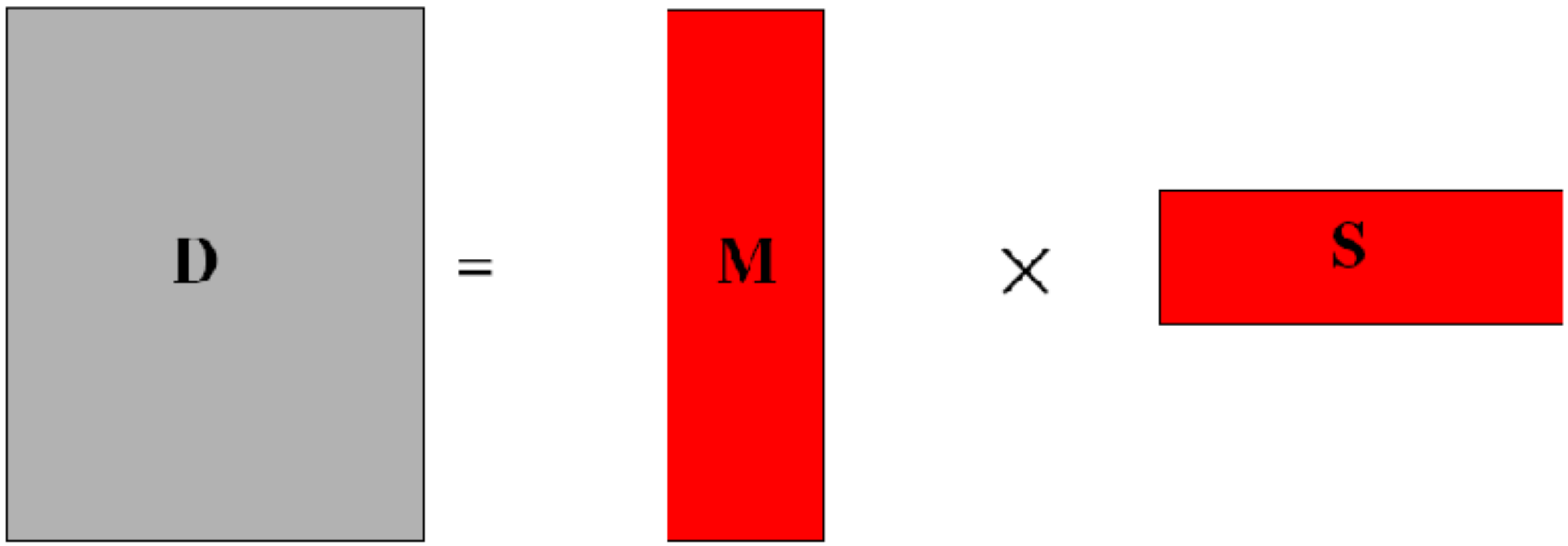
Possible decomposition:

$$M = U_3, \quad S = \Sigma_3 V_3^T$$



This decomposition minimizes $\|\mathbf{D}-\mathbf{MS}\|^2$

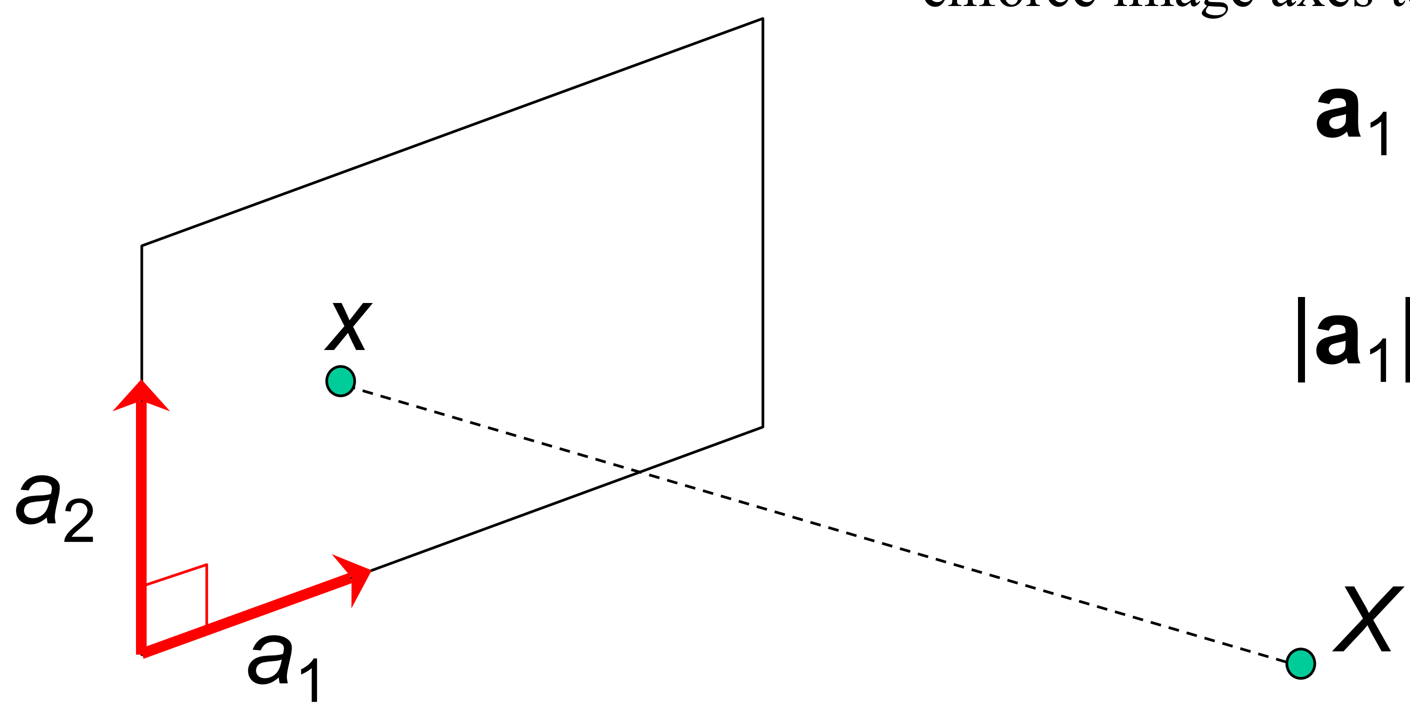
Affine ambiguity



- The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations **M** → **MC, S** →**C**$^{-1}$**S**
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example)

Source: M. Hebert

Eliminating the affine ambiguity



enforce image axes to be ortthonogal and length 1

$$\mathbf{a}_1 \cdot \mathbf{a}_2 = 0$$

$$|\mathbf{a}_1|^2 = |\mathbf{a}_2|^2 = 1$$

$$M = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \end{bmatrix}$$

We want C such that $A_i CC^T A^T_i = Id$

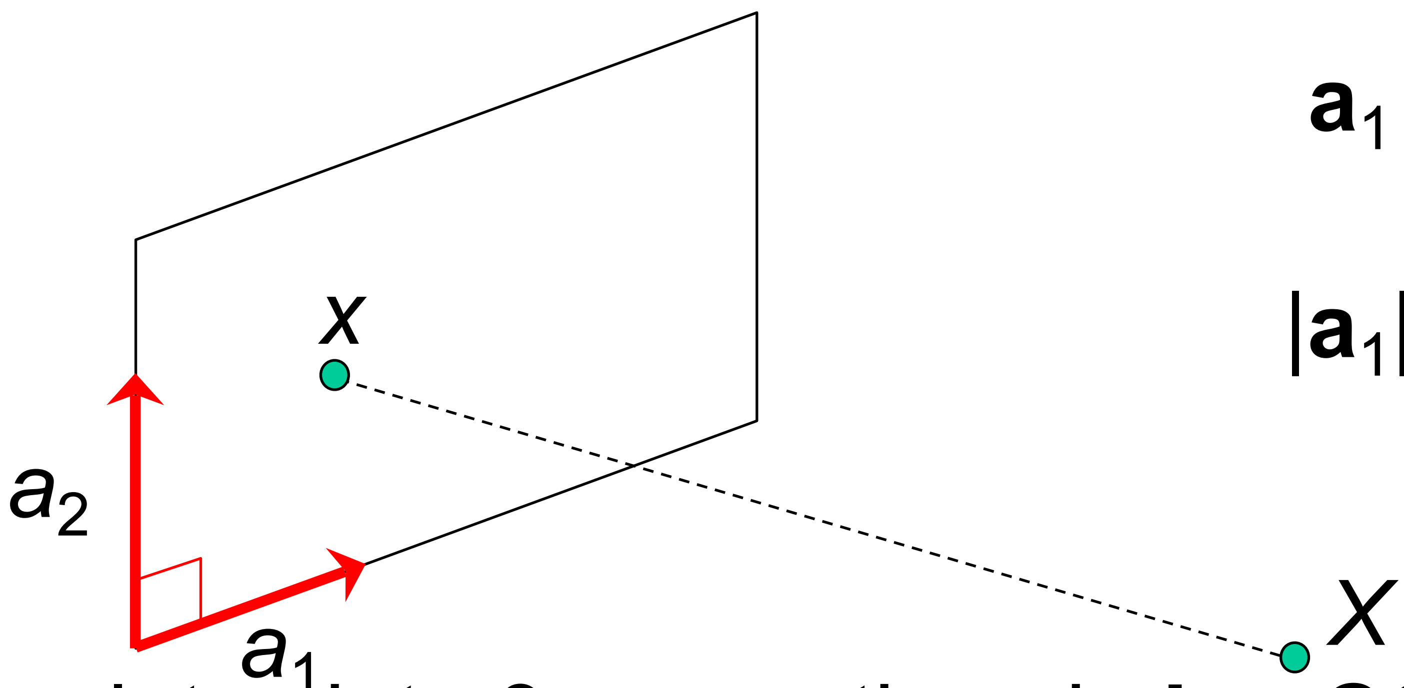
Write out optimization as min

$$\min_L ||A_i L A_i^T - Id||^2$$

Eliminating the affine ambiguity

- Orthographic: image axes are perpendicular and scale is 1

$$M = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \end{bmatrix}$$



$$\mathbf{a}_1 \cdot \mathbf{a}_2 = 0$$

$$|\mathbf{a}_1|^2 = |\mathbf{a}_2|^2 = 1$$

- This translates into $3m$ equations in $\mathbf{L} = \mathbf{C}\mathbf{C}^T$:

$$\mathbf{A_i}\,\mathbf{L}\,\mathbf{A_i}^T = \mathbf{Id}, \qquad i = 1, \ldots, m$$

  - Solve for **L**
  - Recover **C** from **L** by Cholesky decomposition: $\mathbf{L} = \mathbf{C}\mathbf{C}^T$ (in practice, easy to do)
  - Update **M** and **S**: $\mathbf{M} = \mathbf{MC}$, $\mathbf{S} = \mathbf{C}^{-1}\mathbf{S}$

Source: M. Hebert

Algorithm summary

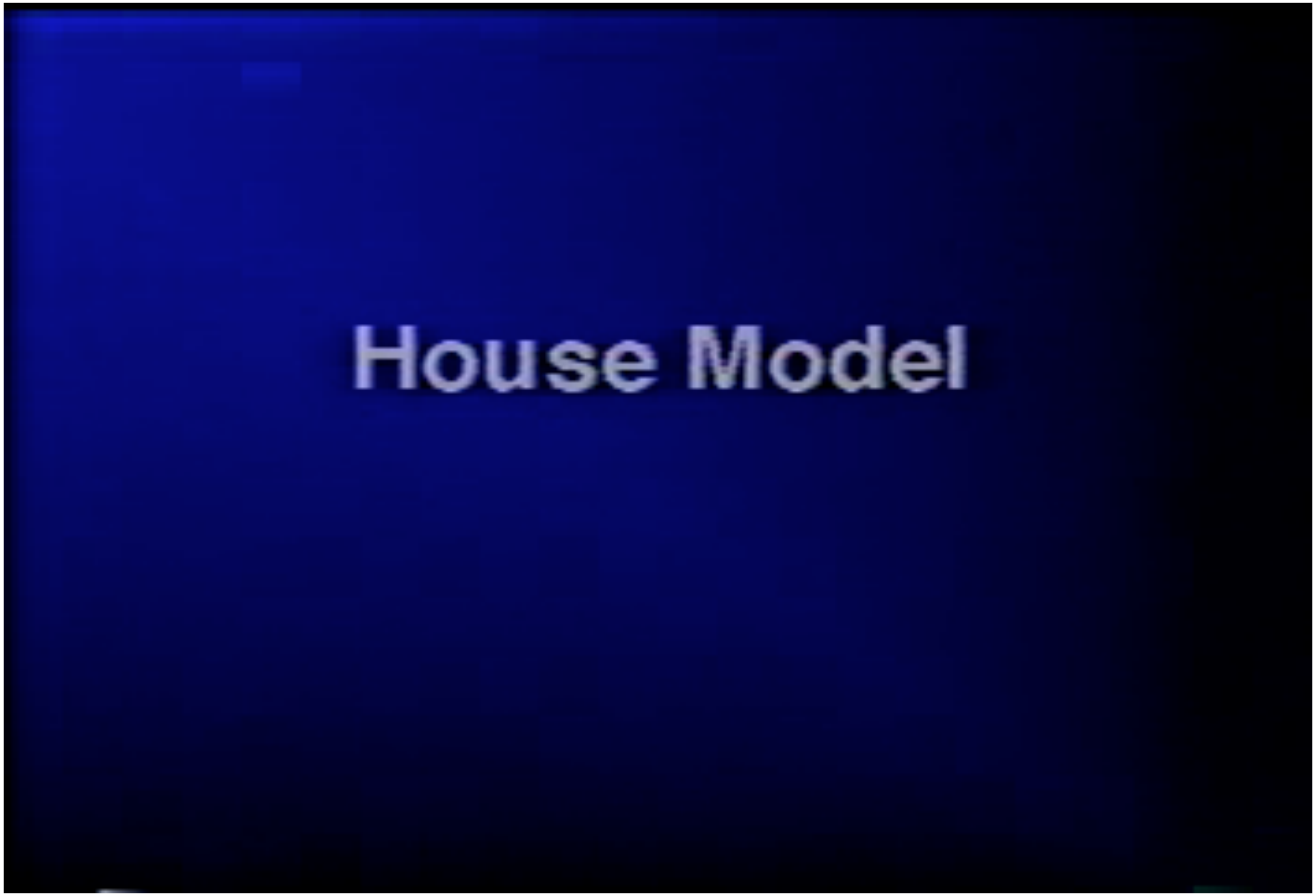
- Given: m images and n features $\mathbf{x}_{ij}$
- For each image i, center the feature coordinates
- Construct a $2m \times n$ measurement matrix **D**:
  - Column j contains the projection of point j in all views
  - Row i contains one coordinate of the projections of all the n points in image i
- 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**
- Create the motion and shape matrices:

$$M = U_3, \quad S = \Sigma_3 V_3^T$$

- Eliminate affine ambiguity

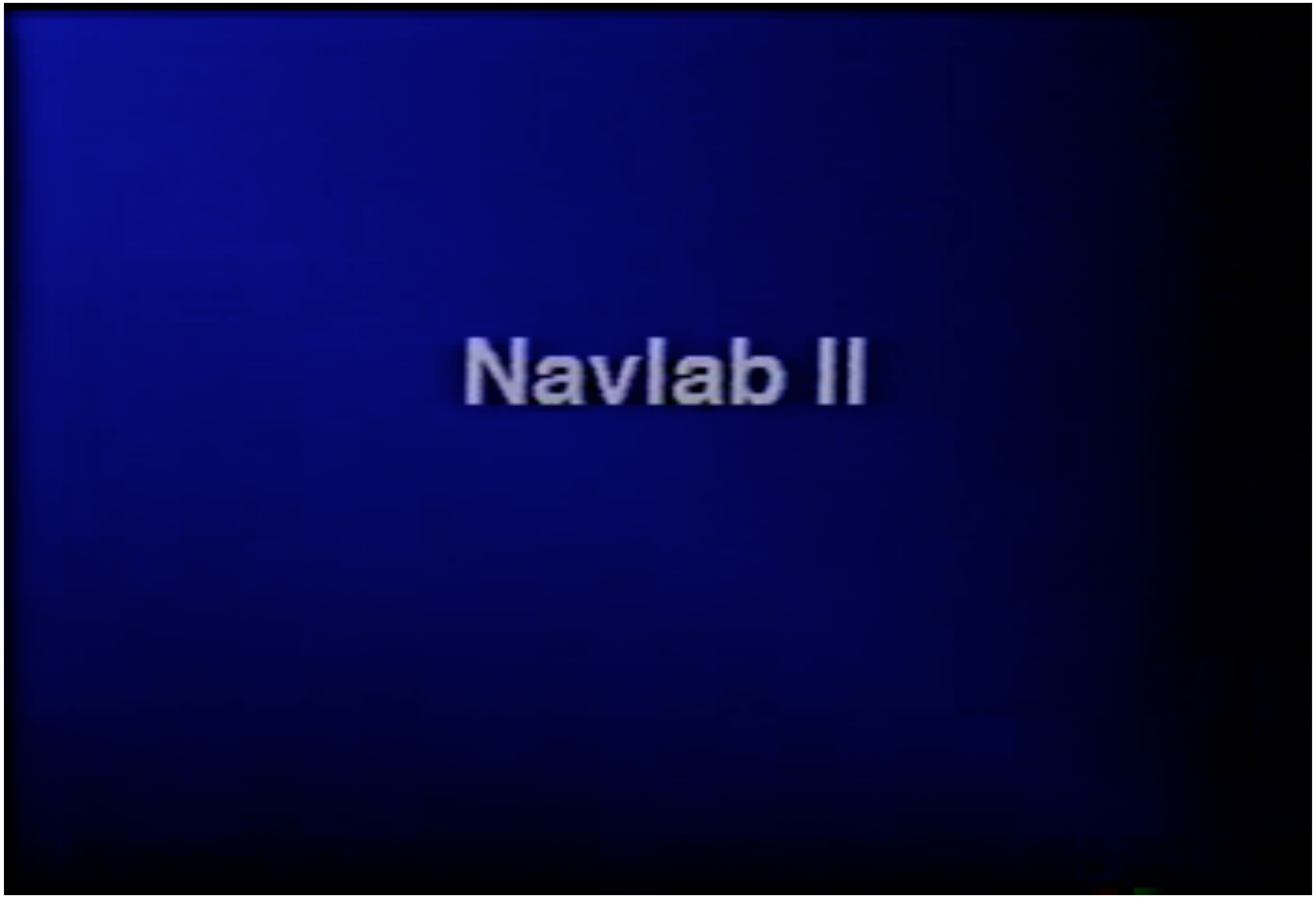
Source: M. Hebert

Results



C. Tomasi and T. Kanade. [Shape and motion from image streams under orthography: A factorization method.]

Results

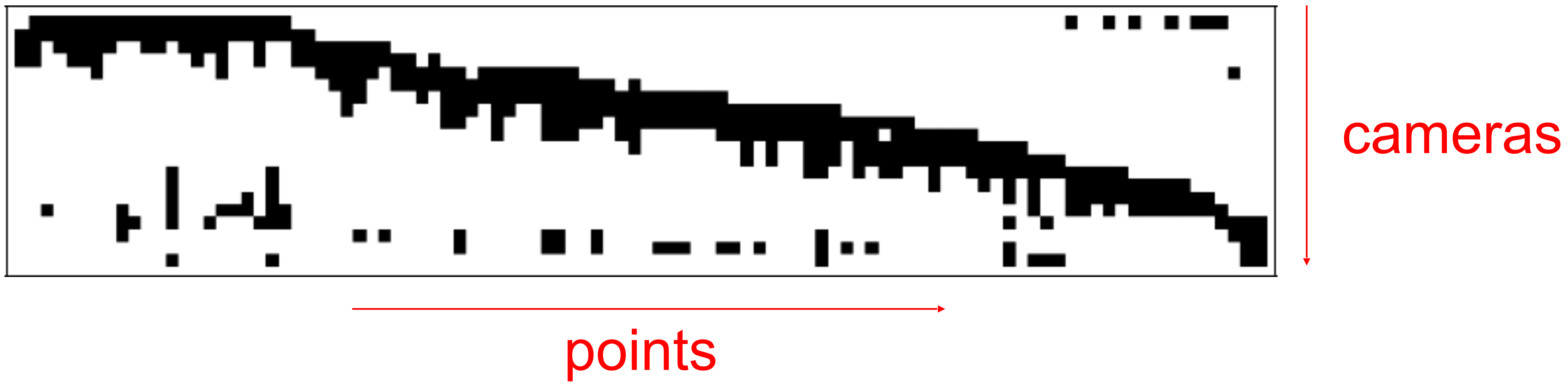


Outline

- Bundle Adjustment
- Ambguities in Reconstruction
- Affine Factorization
- Extensions

Dealing with missing data

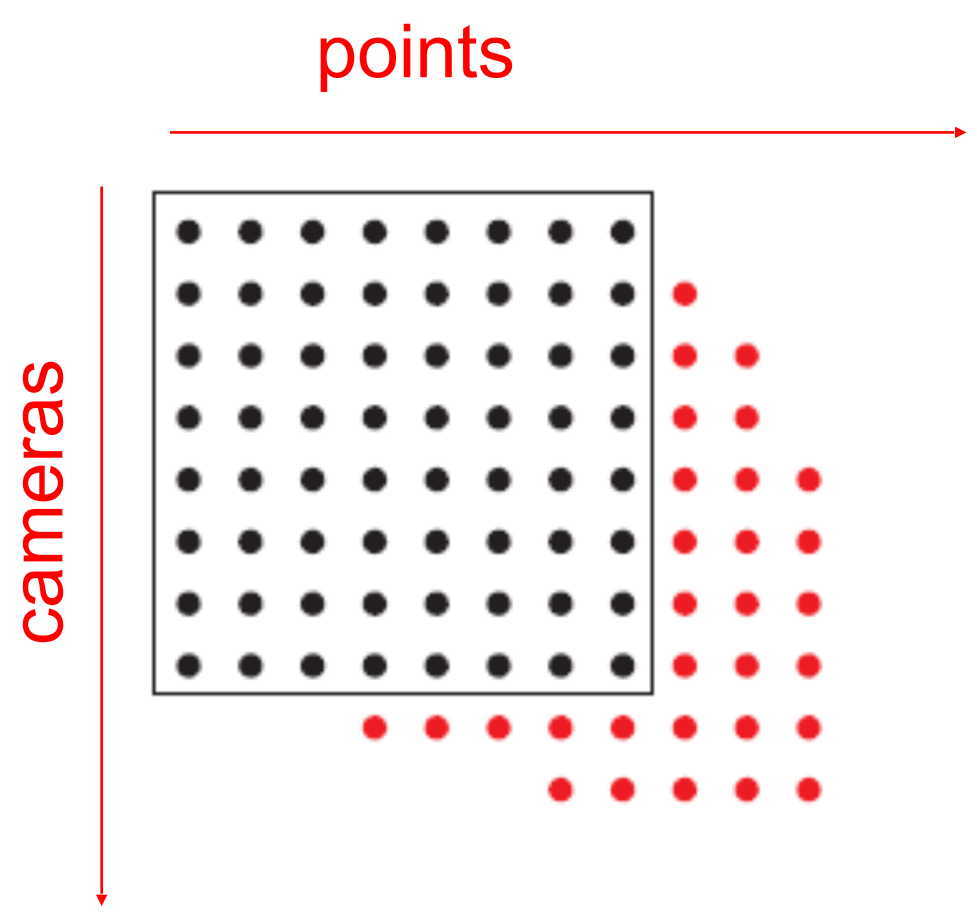
- So far, we have assumed that all points are visible in all views
- In reality, the measurement matrix typically looks something like this:



Sequential structure from motion

- Intuition: exploit low-rank redundancy of matrix
- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
- Incremental bilinear refinement

(1) Perform factorization on a dense sub-block

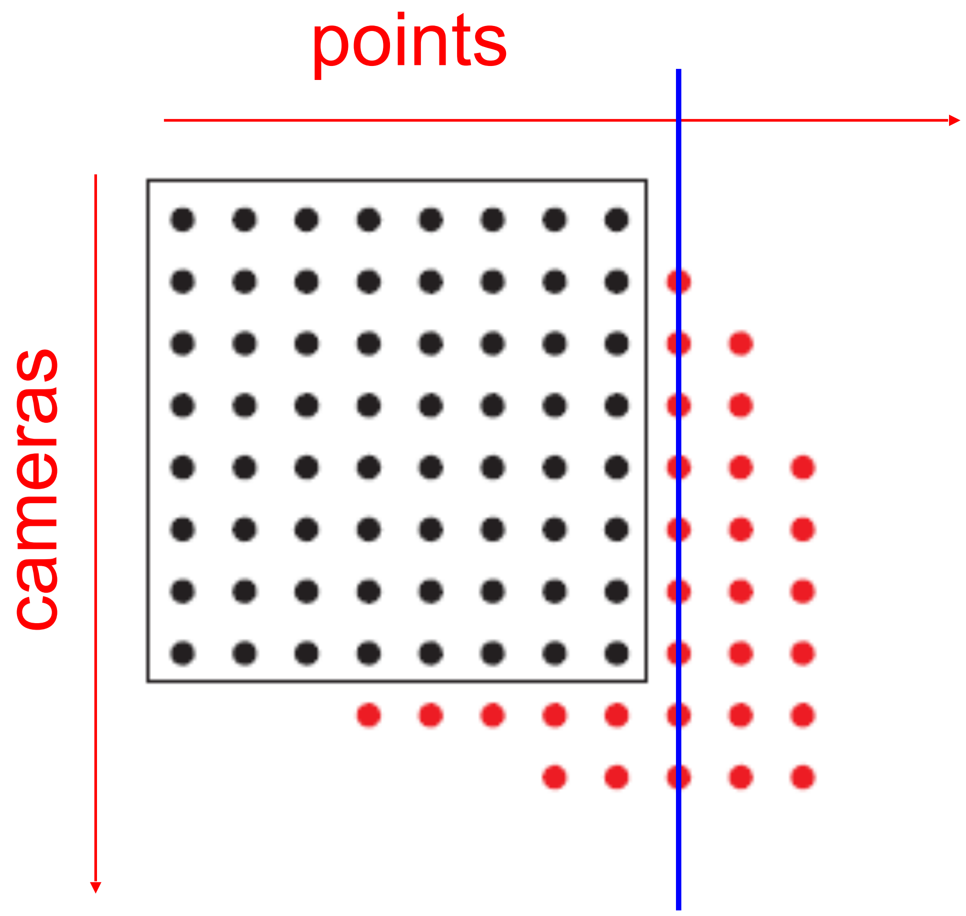


Sequential structure from motion

- Intuition: exploit low-rank redundancy of matrix
- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
- Incremental bilinear refinement

(2) Solve for a new 3D point visible by at least two known cameras (linear least squares)

$$\min_S ||D - MS||^2$$

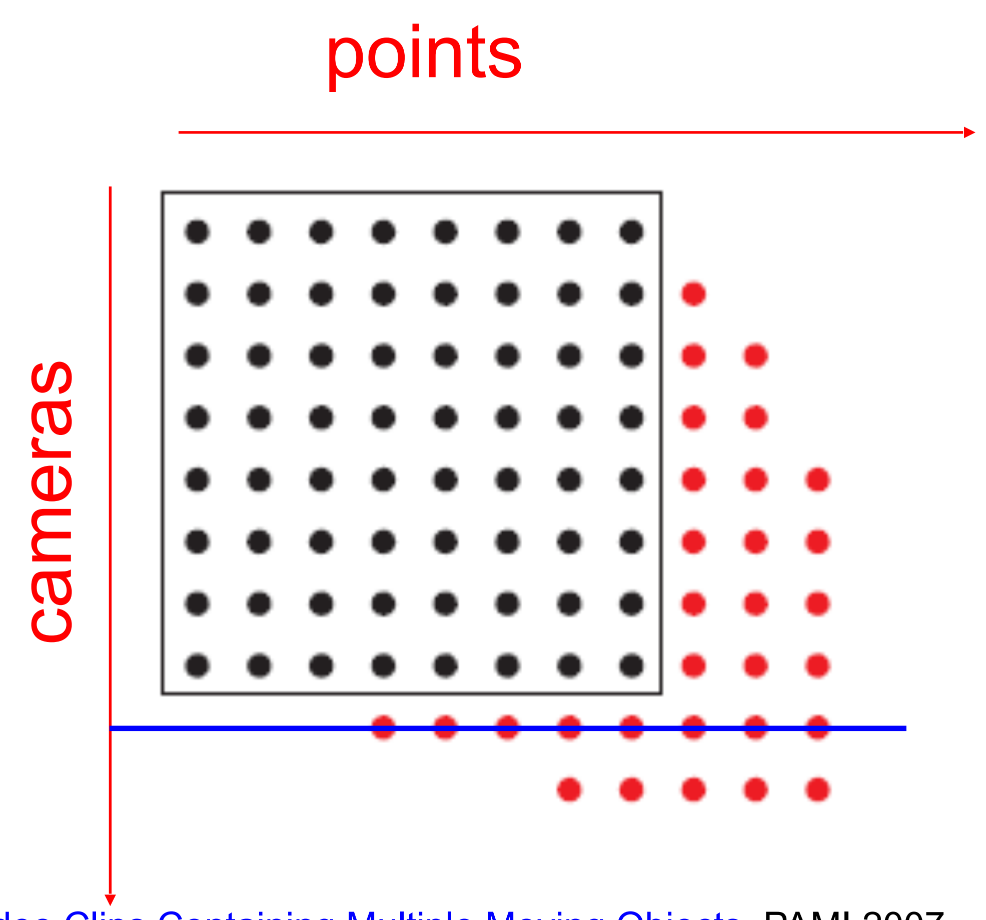


Sequential structure from motion

- Intuition: exploit low-rank redundancy of matrix
- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
- Incremental bilinear refinement

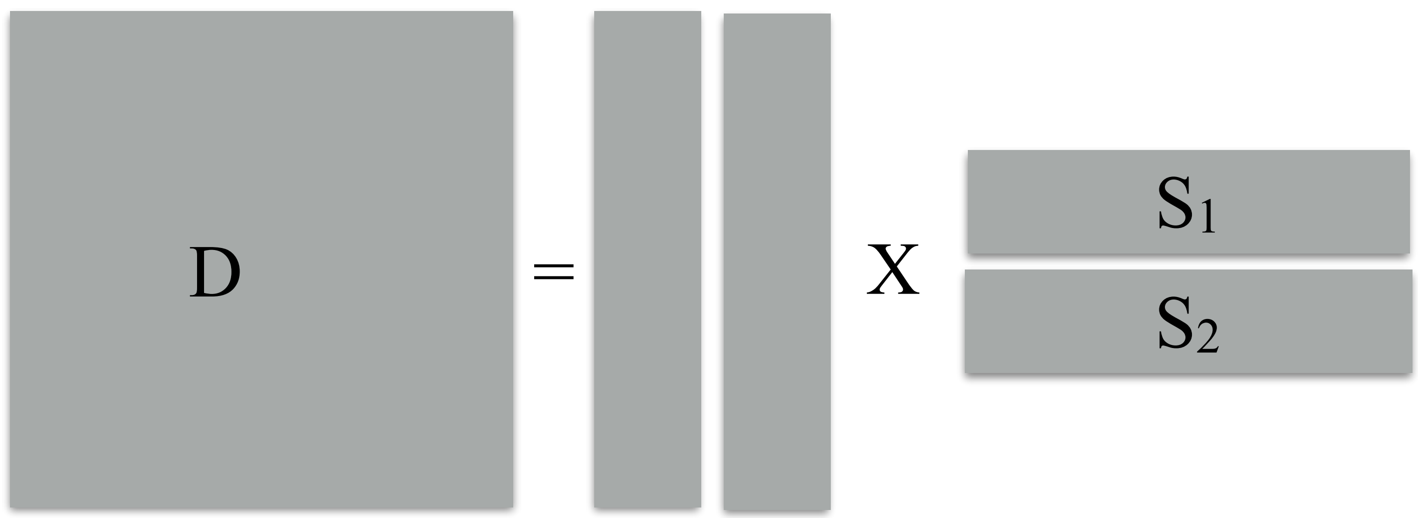
(3) Solve for a new camera that sees at least three known 3D points (linear least squares)

$$\min_{M} ||D - MS||^2$$



F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. [Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects.] PAMI 2007.

Nonrigid structure motion



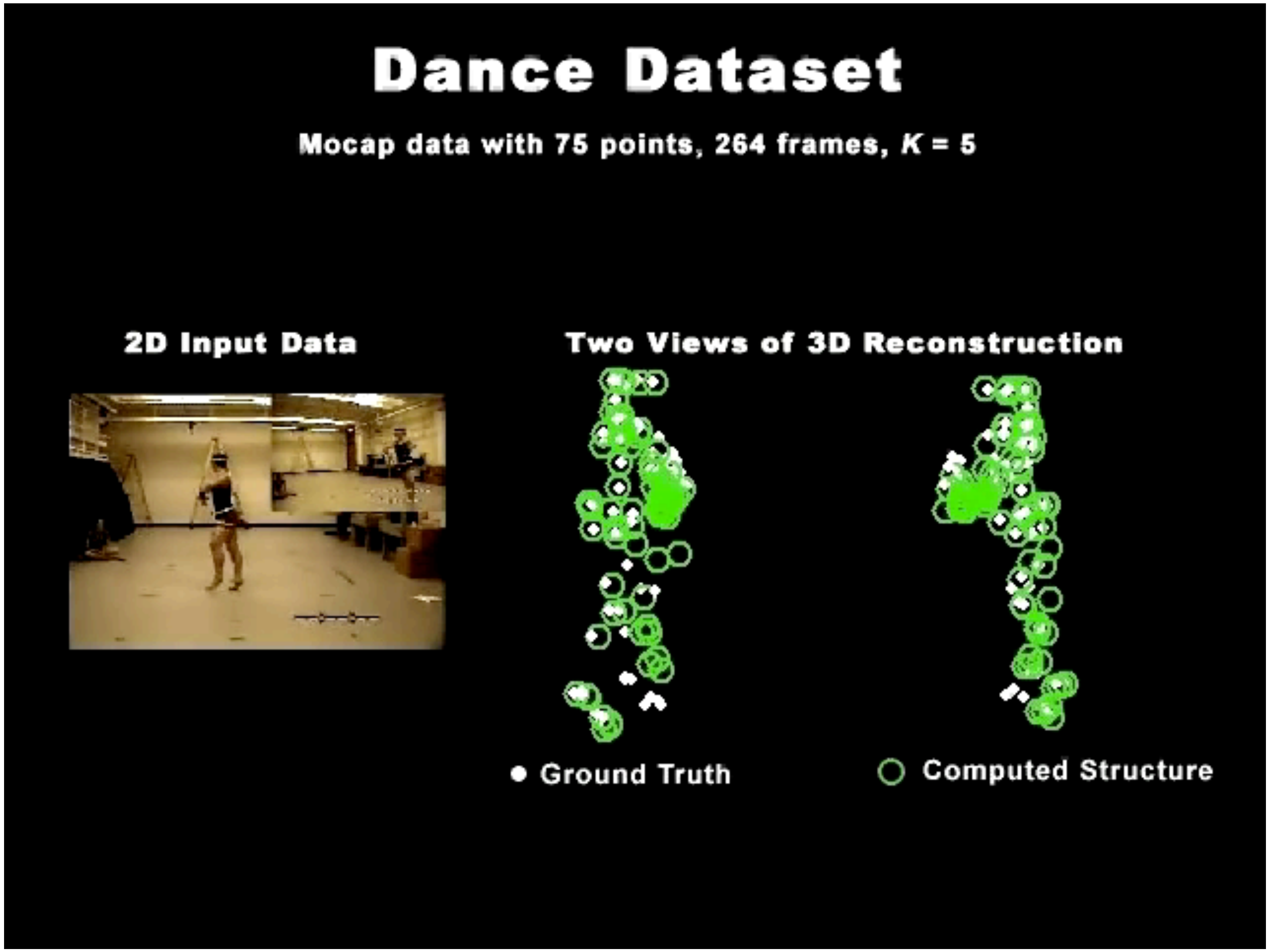
$$\alpha_1 M \;\; \alpha_2 M$$

Assume 3D shapes are a linear combination of K basis shapes

$$S = \sum_{k=1}^{K} \alpha_k S_k$$

For each image, we need to solve for camera matrix *and* scaling coefficients

Simply relax rank constraint on measurement matrix from 3 to 3K!


Dance Dataset
Mocap data with 75 points, 264 frames, K = 5
2D Input Data
Two Views of 3D Reconstruction
Ground Truth
Computed Structure

Projective structure from motion

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

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

- Problem: estimate *m* projection matrices $\mathbf{M}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation **Q**:

$$\mathbf{X} \rightarrow \mathbf{QX}, \mathbf{M} \rightarrow \mathbf{MQ}^{-1}$$

- We can solve for structure and motion when

$$2mn >= 11m + 3n - 15$$

- For two cameras, at least 7 points are needed

Projective SFM: Two-camera case

- Compute fundamental matrix $\mathbf{F}$ between the two views (from 7 or more correspondences)
- First camera matrix: M = $[\mathbf{I}|\mathbf{0}]$
- Second camera matrix: M' = $[\mathbf{A}_{3x3}|\mathbf{b}_{3x1}]$
- Then one can compute **A,b** from F as follows:
  - $\mathbf{b}$ is the right null vector, or epipole ($\mathbf{F}^T\mathbf{b} = 0$)

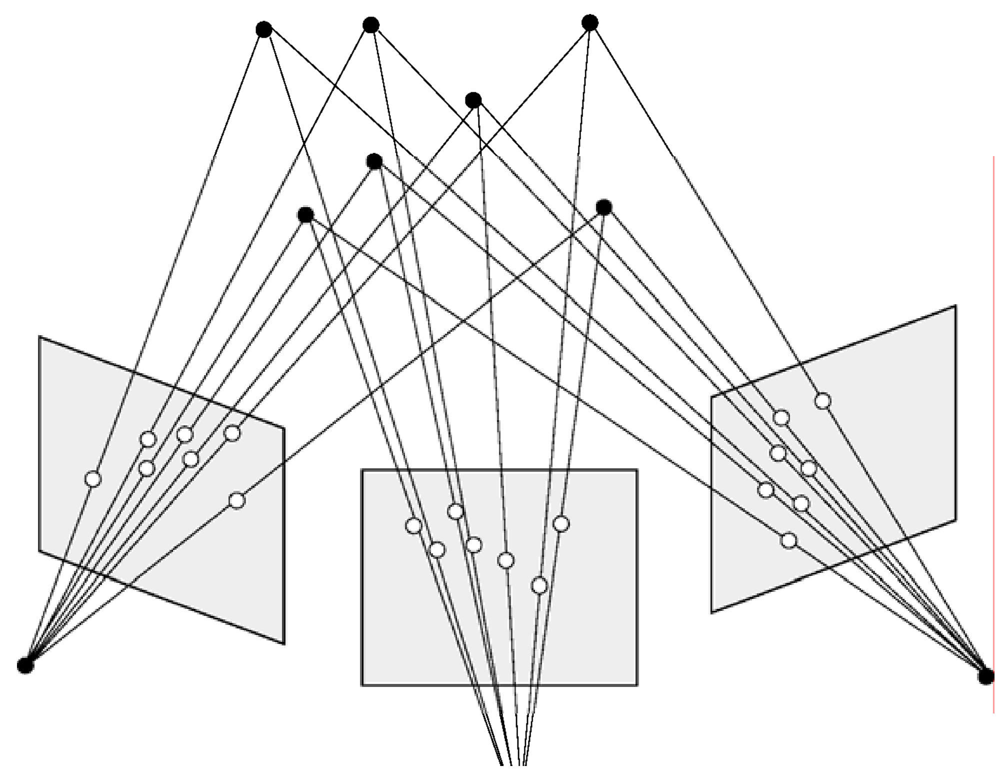
  $A = -\hat{\mathbf{b}}F$ (where hat notation refers to skew symmetric matrix)

For proof, see F & P Sec 8.3

For more than 2 images, things get more implicated

Back to bundle adjustment

Minimize reprojection error over multiple 3D points and cameras



$$\min_{\mathbf{X}_1,\mathbf{X}_2,\ldots,M_1,M_2,\ldots} \sum_{i=1}^{m}\sum_{j=1}^{n} ||\mathbf{x}_{ij} - Proj(\mathbf{X}_j, M_i)||^2$$

Self-calibration

- Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images
- For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images
  - Compute initial projective reconstruction and find 3D projective transformation matrix $\mathbf{Q}$ such that all camera matrices are in the form $\mathbf{M}_i = \mathbf{K}\,[\mathbf{R}_i \mid \mathbf{t}_i]$
- Can use constraints on the form of the calibration matrix: orthogonal image axis
- Can use vanishing points

Review: Structure from motion

- Ambiguity
- Affine structure from motion
  - Factorization
- Dealing with missing data
  - Incremental structure from motion
- Projective structure from motion
  - Bundle adjustment
  - Self-calibration

Summary: 3D geometric vision

- Single-view geometry
  - The pinhole camera model
    - Variation: orthographic projection
  - The perspective projection matrix
  - Intrinsic parameters
  - Extrinsic parameters
  - Calibration
- Multiple-view geometry
  - Triangulation
  - The epipolar constraint
    - Essential matrix and fundamental matrix
  - Stereo
    - Binocular, multi-view
  - Structure from motion
    - Reconstruction ambiguity
    - Affine SFM
    - Projective SFM