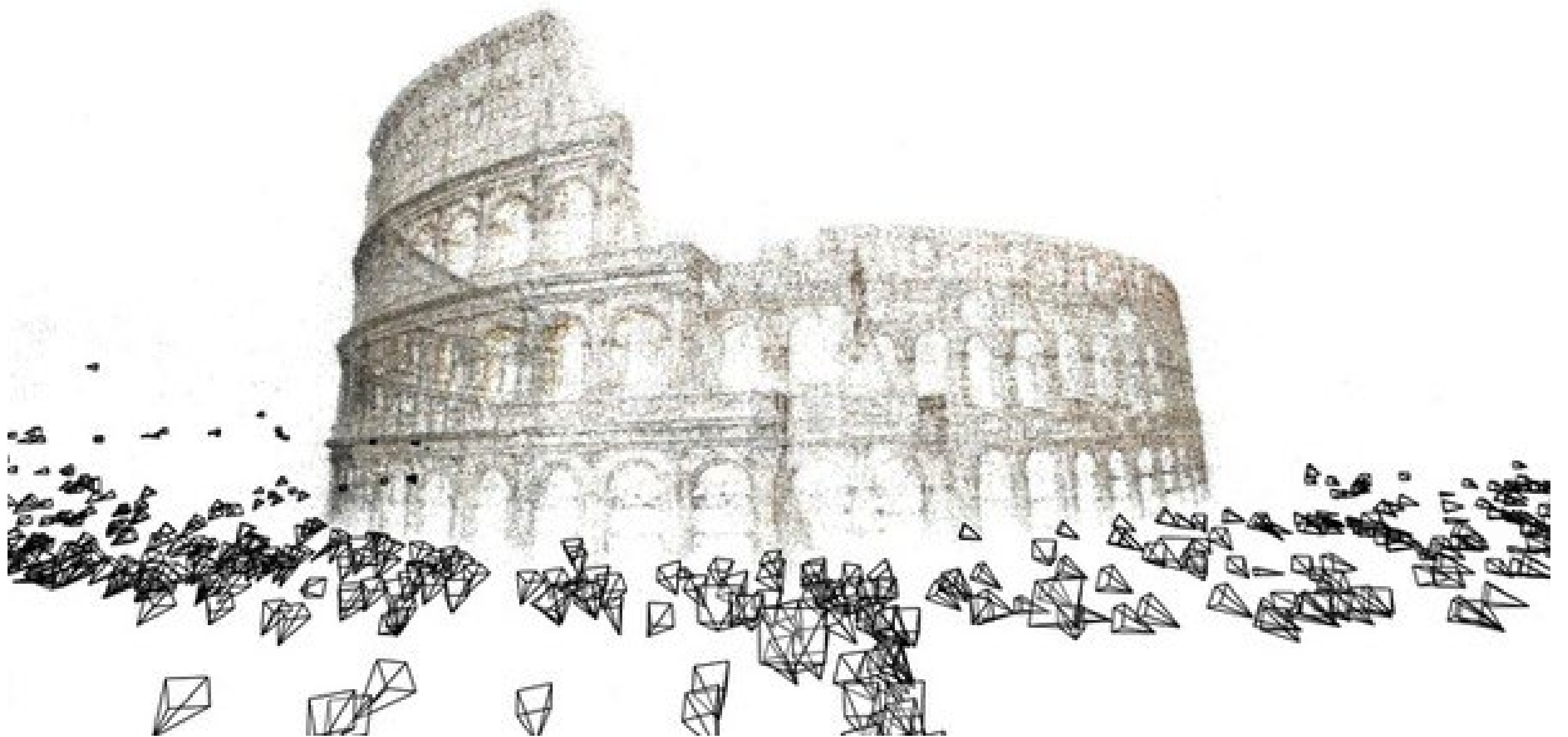


Structure from motion



Course announcements

- Homework assignment 1 will be posted this Friday.

Overview of today's lecture

- Two-view structure from motion.
- Ambiguities in structure from motion.
- Affine structure from motion.
- Multi-view structure from motion.
- Large-scale structure from motion.

Slide credits

Many of these slides were adapted from:

- Kris Kitani (16-385, Spring 2017).
- Noah Snavely (Cornell University).
- Rob Fergus (New York University).

Two-view structure from motion

	Structure (scene geometry)	Motion (camera geometry)	Measurements
Camera calibration	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D correspondences
Reconstruction	estimate	estimate	2D to 2D correspondences

Structure from motion



Драконъ, видимый подъ различными углами зрѣнія
По гравюру на мѣди изъ „Oculus artificialis teleiopicus“ Нана. 1702 года.

Camera calibration & triangulation

- Suppose we know 3D points
 - And have matches between these points and an image
 - How can we compute the camera parameters?
- Suppose we have known cameras, each of which observes a point
 - How can we compute the 3D location of that point?

Structure from motion

- SfM solves both of these problems *at once*
- A kind of chicken-and-egg problem
 - (but solvable)

Reconstruction

(2 view structure from motion)

Given a set of matched points

$$\{\mathbf{x}_i, \mathbf{x}'_i\}$$

Estimate the camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

Reconstruction

(2 view structure from motion)

Given a set of matched points

$$\{x_i, x'_i\}$$

Estimate the camera matrices

$$\mathbf{P}, \mathbf{P}' \leftarrow \text{'motion'}$$

(of the cameras)

Estimate the 3D point

$$\mathbf{X} \leftarrow \text{'structure'}$$

Two-view SfM

1. Compute the Fundamental Matrix $\mathbf{F}$ from points correspondences

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

Two-view SfM

1. Compute the Fundamental Matrix **F** from points correspondences
8-point algorithm

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

Two-view SfM

1. Compute the Fundamental Matrix **F** from points correspondences

8-point algorithm

2. Compute the camera matrices **P** from the Fundamental matrix

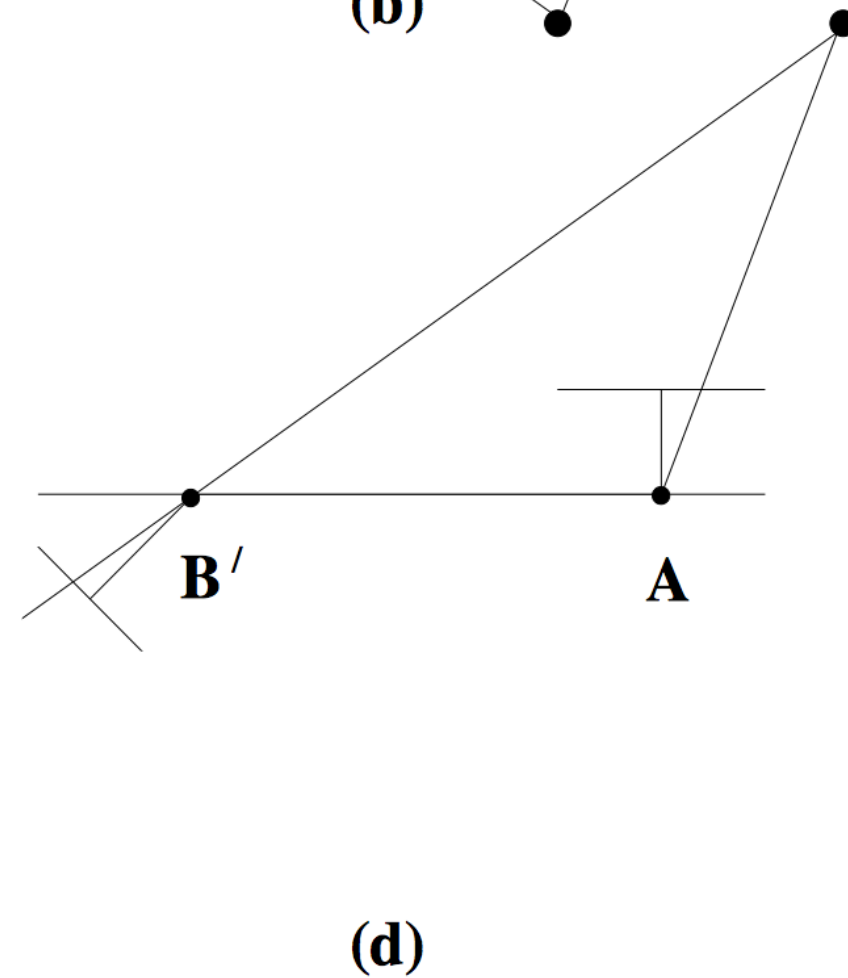
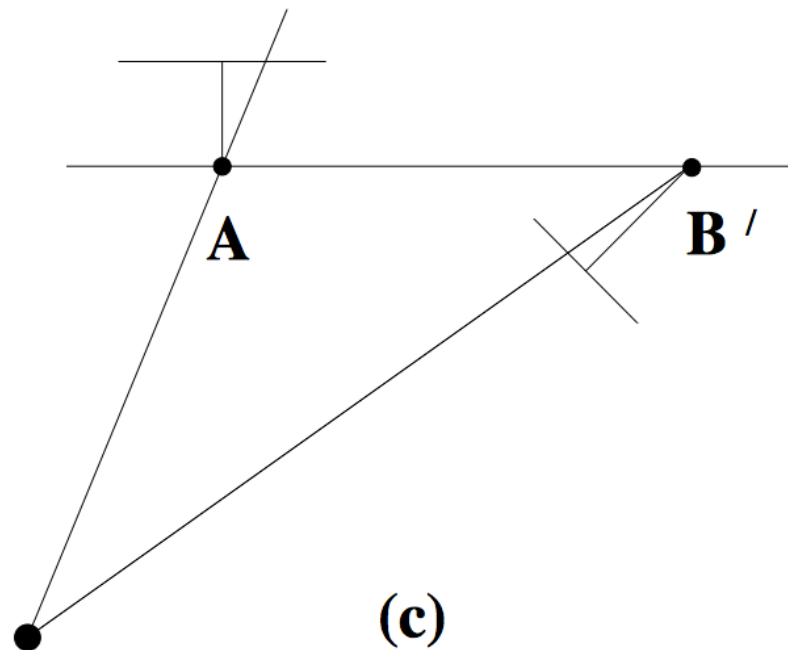
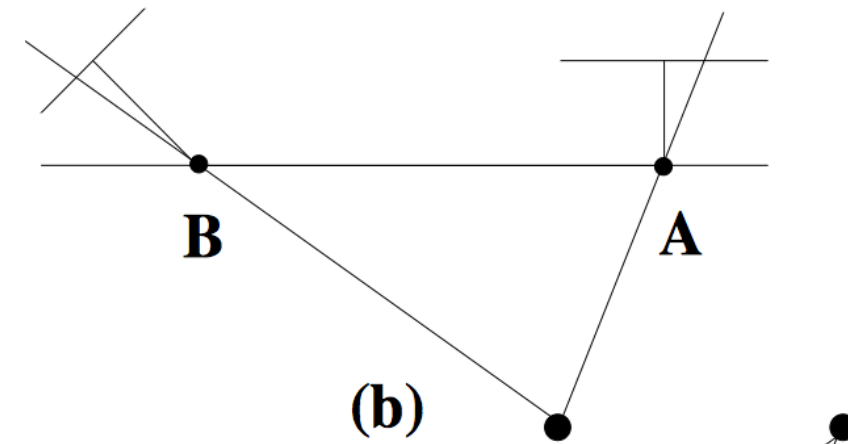
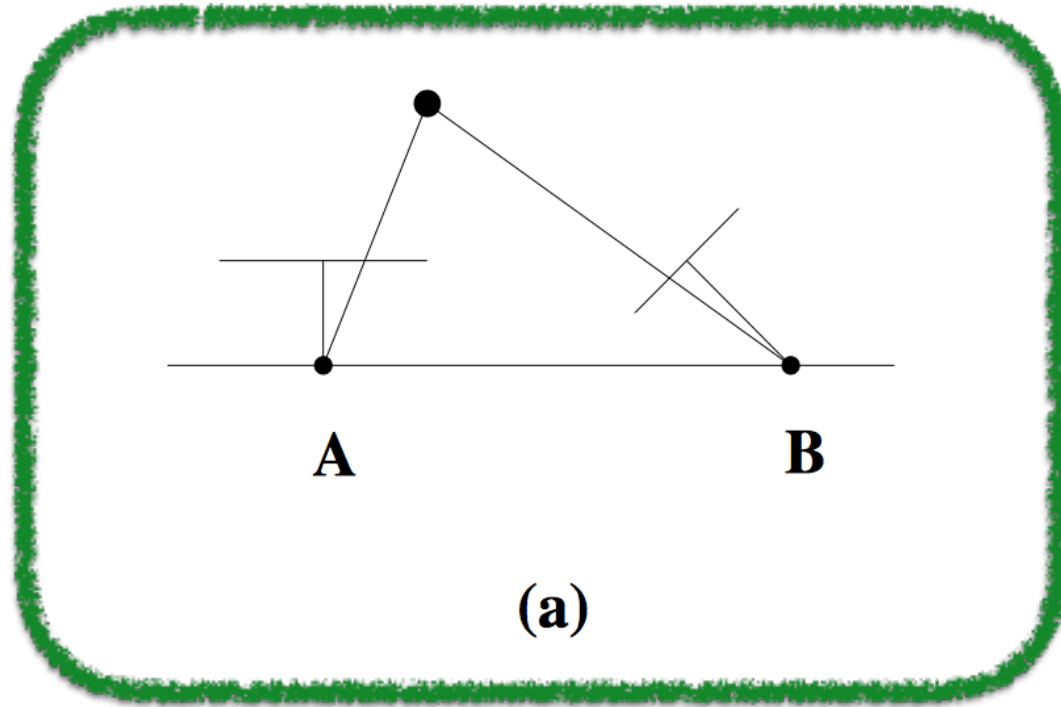
$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \text{ and } \mathbf{P}' = [[\mathbf{e}_x]\mathbf{F} \mid \mathbf{e}']$$

Camera matrices corresponding to the fundamental matrix $\mathbf{F}$ may be chosen as

$$\mathbf{P} = [\mathbf{I} | \mathbf{0}] \quad \mathbf{P}' = [[e_x] \mathbf{F} | e']$$

(See Hartley and Zisserman C.9 for proof)

Find the configuration where the points is in front of both cameras



Two-view SfM

1. Compute the Fundamental Matrix **F** from points correspondences

8-point algorithm

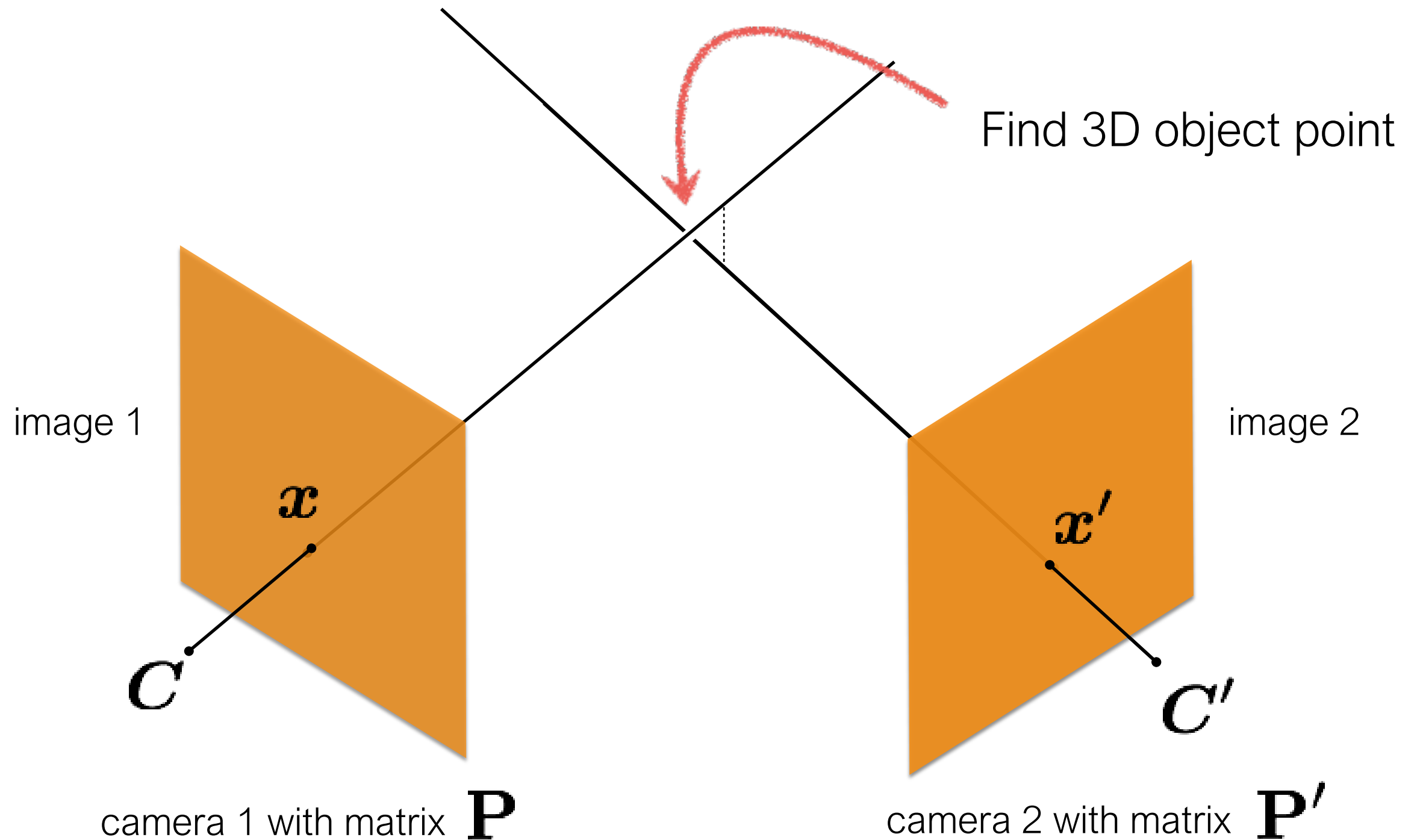
2. Compute the camera matrices **P** from the Fundamental matrix

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \text{ and } \mathbf{P}' = [[\mathbf{e}'_x]\mathbf{F} \mid \mathbf{e}']$$

3. For each point correspondence, compute the point **X** in 3D space (triangulation)

DLT with $\mathbf{x} = \mathbf{P} \mathbf{X}$ and $\mathbf{x}' = \mathbf{P}' \mathbf{X}$

Triangulation



Two-view SfM

1. Compute the Fundamental Matrix **F** from points correspondences

8-point algorithm

2. Compute the camera matrices **P** from the Fundamental matrix

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \text{ and } \mathbf{P}' = [[\mathbf{e}'_x]\mathbf{F} \mid \mathbf{e}']$$

3. For each point correspondence, compute the point **X** in 3D space (triangulation)

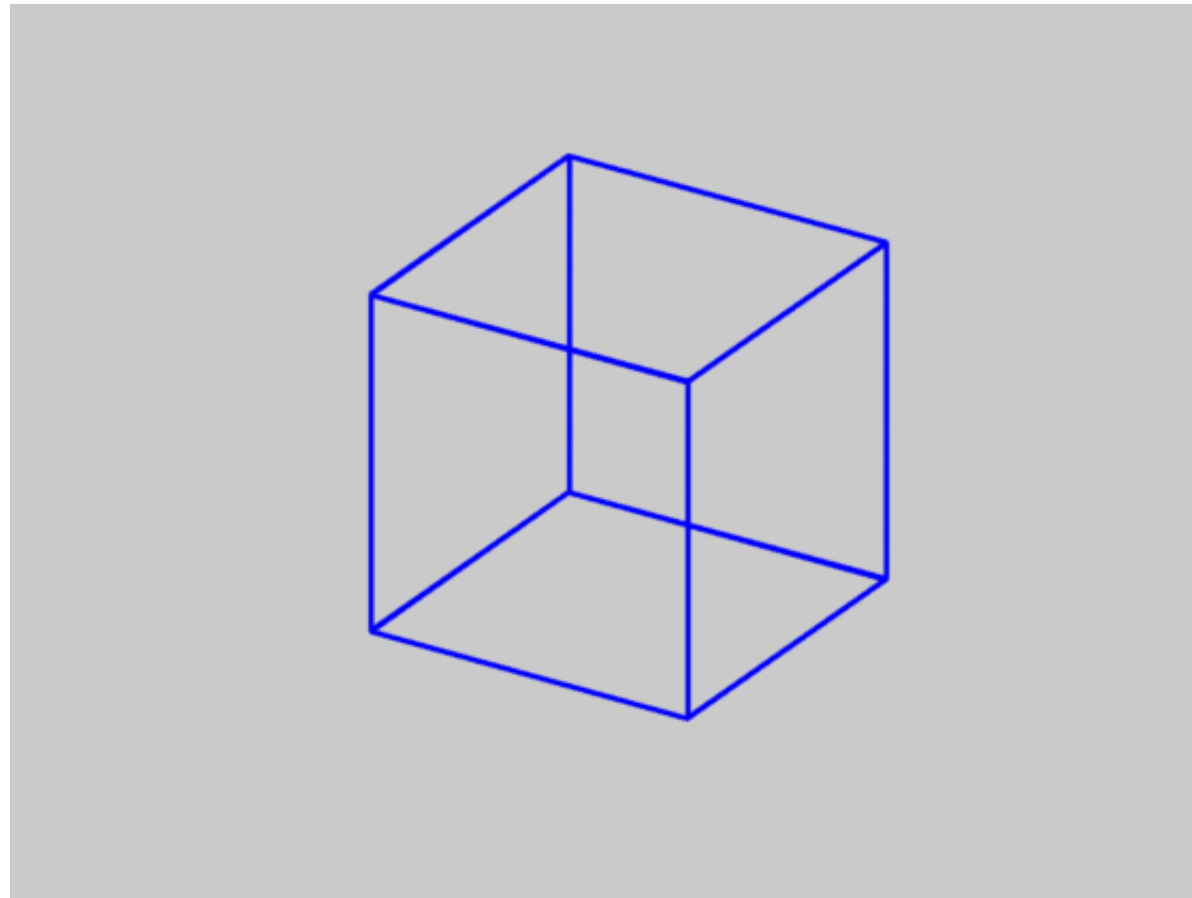
DLT with $\mathbf{x} = \mathbf{P} \mathbf{X}$ and $\mathbf{x}' = \mathbf{P}' \mathbf{X}$

Is SfM always uniquely
solvable?

Ambiguities in structure from motion

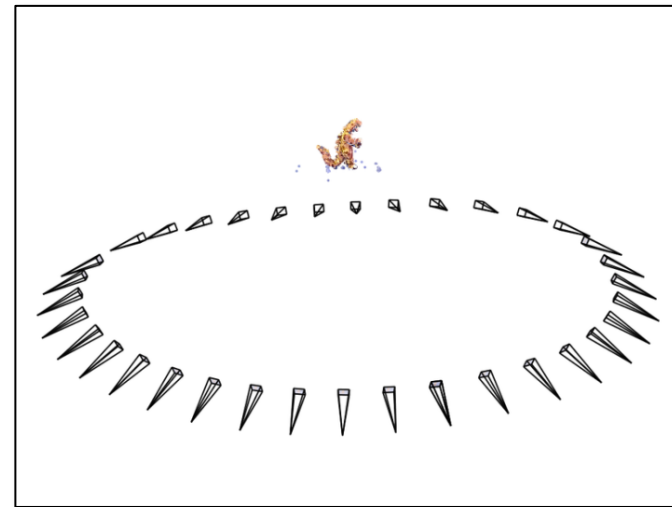
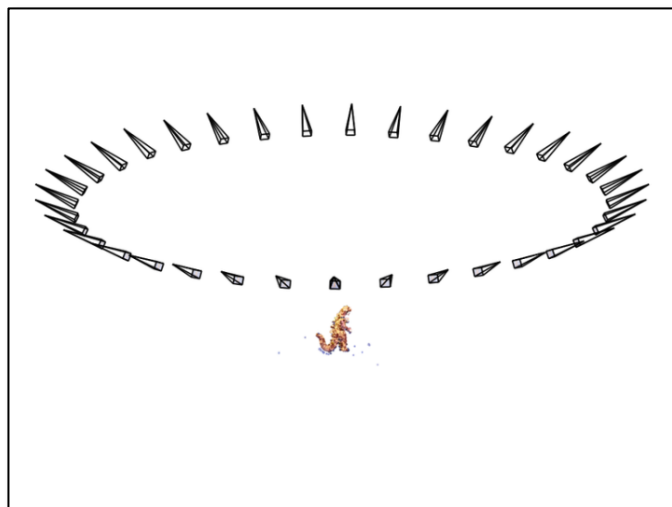
Is SfM always uniquely solvable?

- No....



SfM – Failure cases

- Necker reversal



Projective Ambiguity

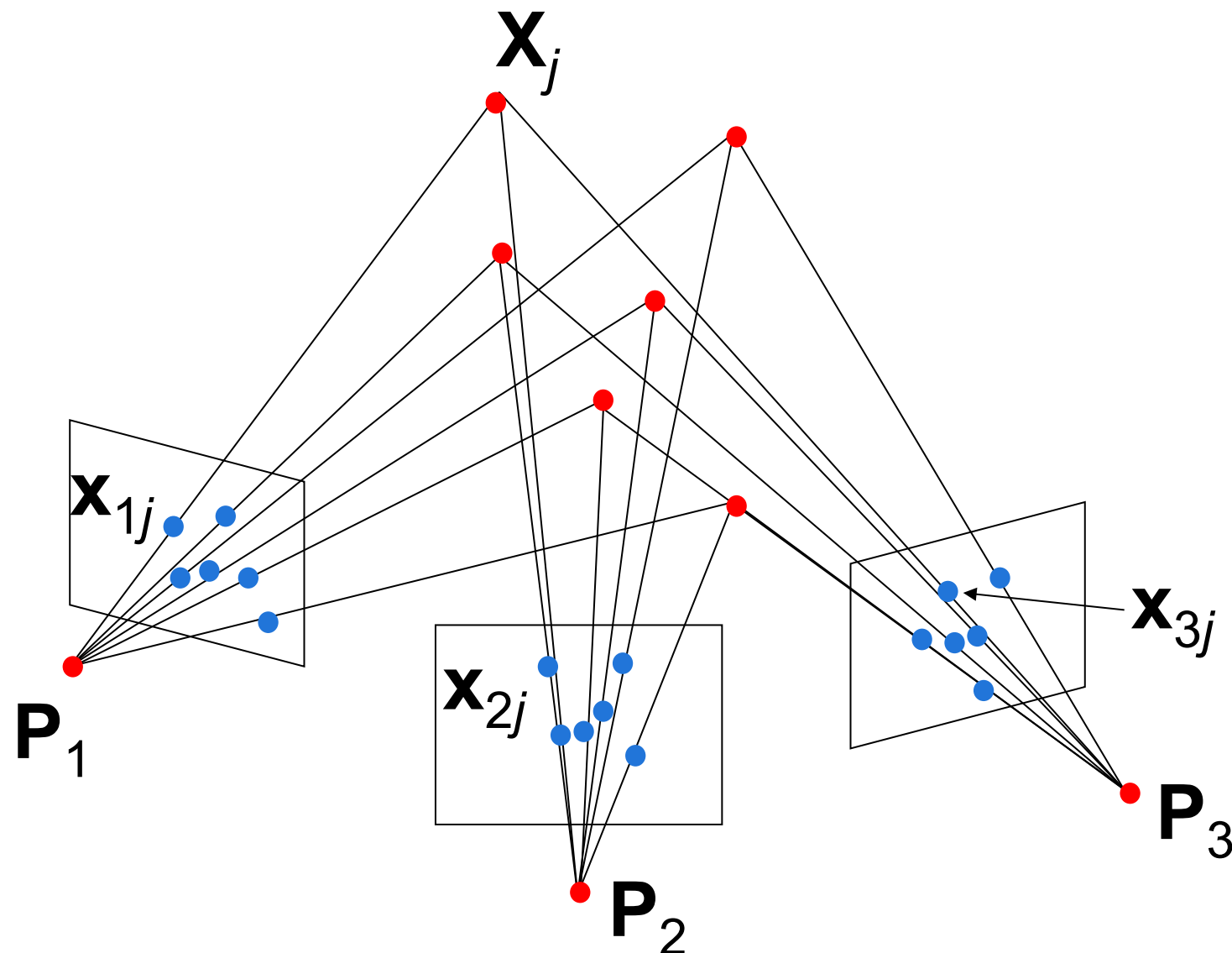
- Reconstruction is ambiguous by an arbitrary 3D projective transformation without prior knowledge of camera parameters

Structure from motion

- Given: m images of n fixed 3D points

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

- Problem: estimate m projection matrices $\mathbf{P}_i$ and n 3D points $\mathbf{X}_j$ from the mn correspondences $\mathbf{x}_{ij}$



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:

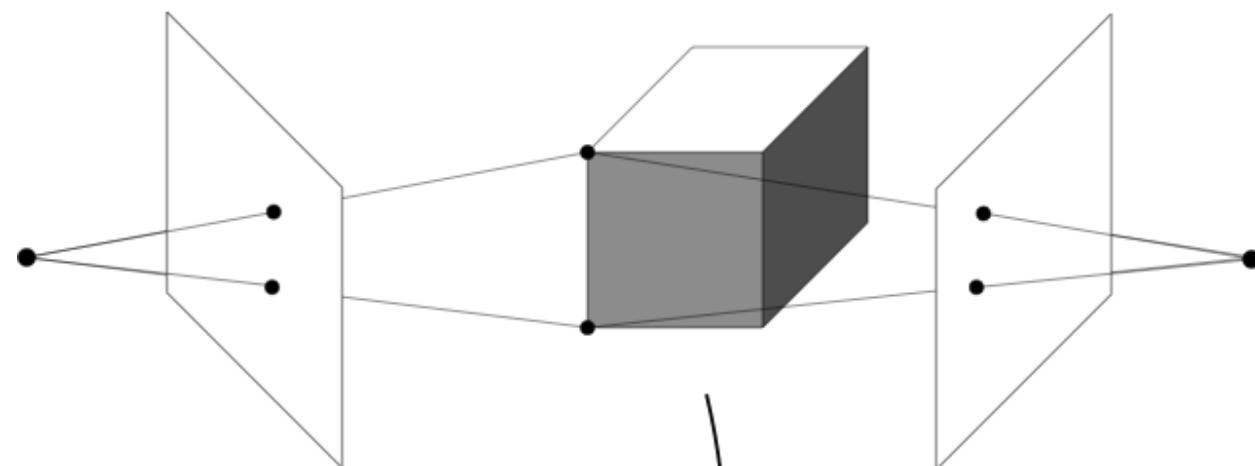
$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\mathbf{X})$$

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

Structure from motion ambiguity

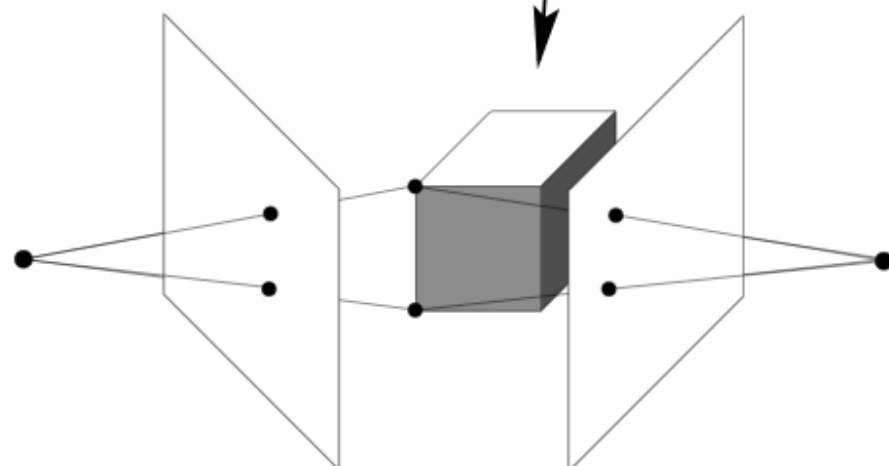
- If we scale the entire scene by some factor k and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same
- More generally: if we transform the scene using a transformation $\mathbf{Q}$ and apply the inverse transformation to the camera matrices, then the images do not change

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

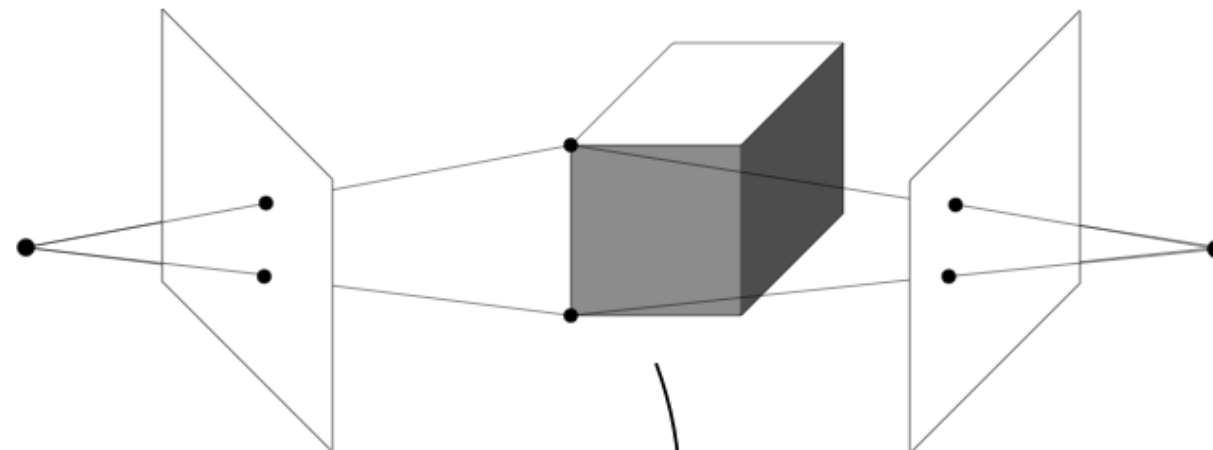


Calibrated cameras
(similarity projection ambiguity)

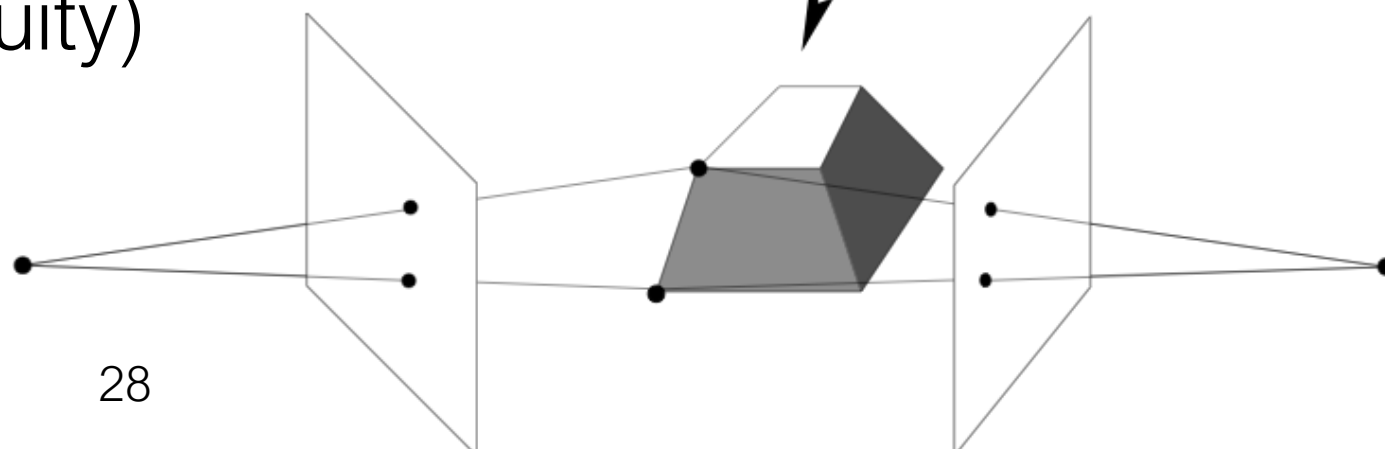
Similarity



Uncalibrated cameras
(projective projection ambiguity)



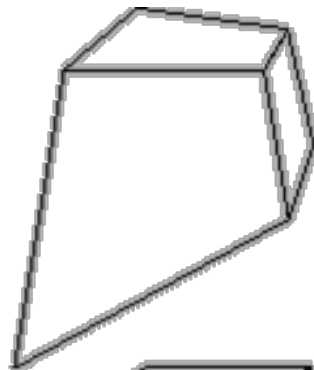
Projective



Types of ambiguity

Projective
15 dof

$$\begin{bmatrix} A & t \\ v^\top & v \end{bmatrix}$$



Preserves intersection and tangency

Affine
12 dof

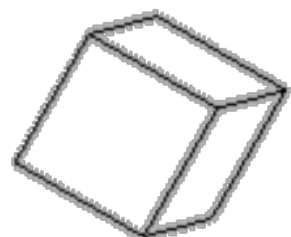
$$\begin{bmatrix} A & t \\ 0^\top & 1 \end{bmatrix}$$



Preserves parallelism, volume ratios

Similarity
7 dof

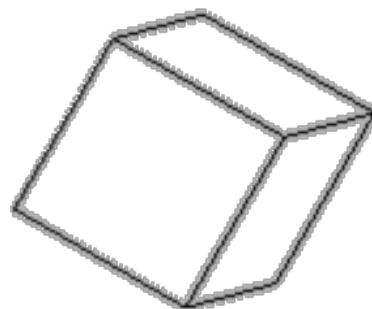
$$\begin{bmatrix} sR & t \\ 0^\top & 1 \end{bmatrix}$$



Preserves angles, ratios of length

Euclidean
6 dof

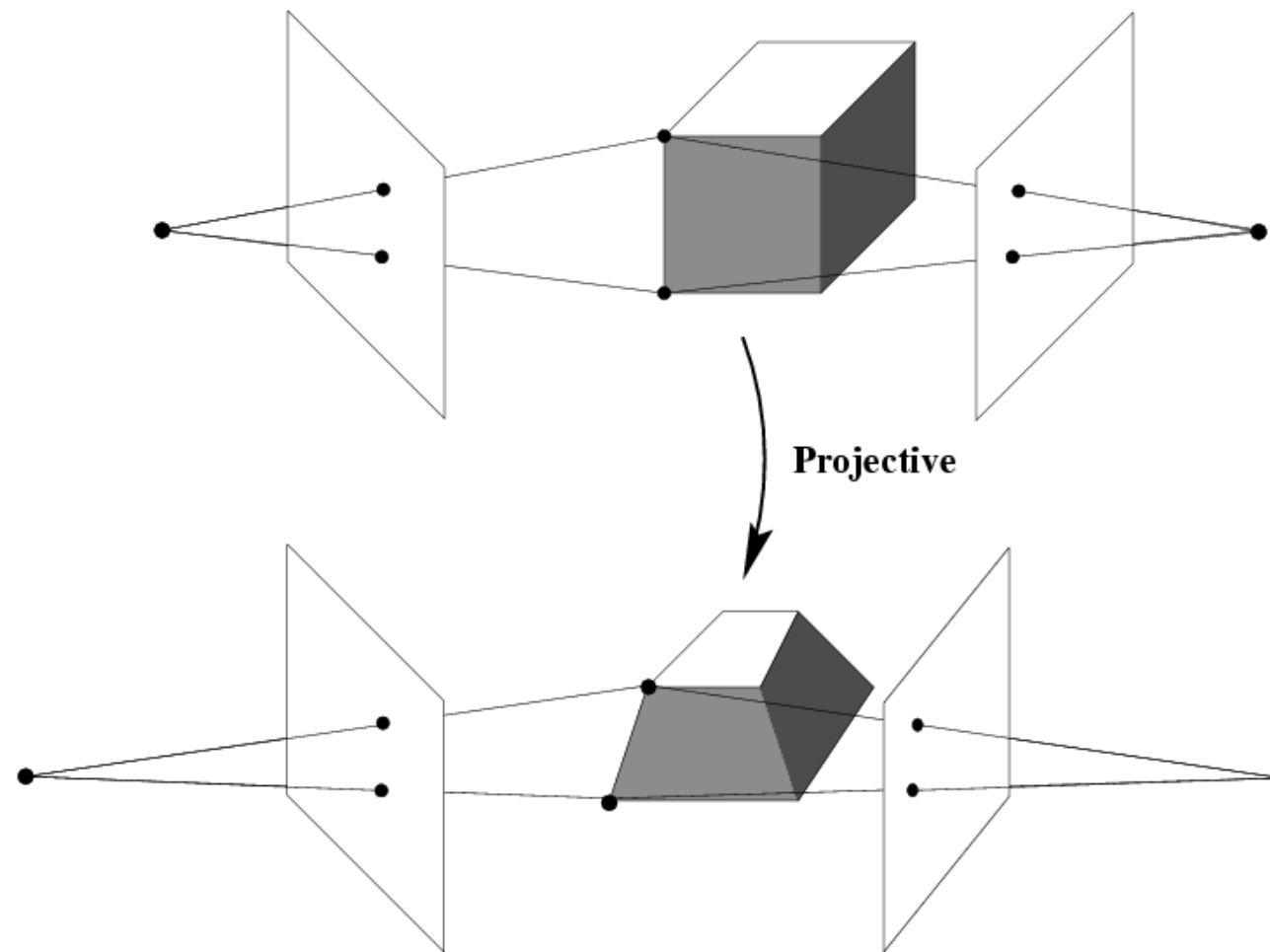
$$\begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix}$$



Preserves angles, lengths

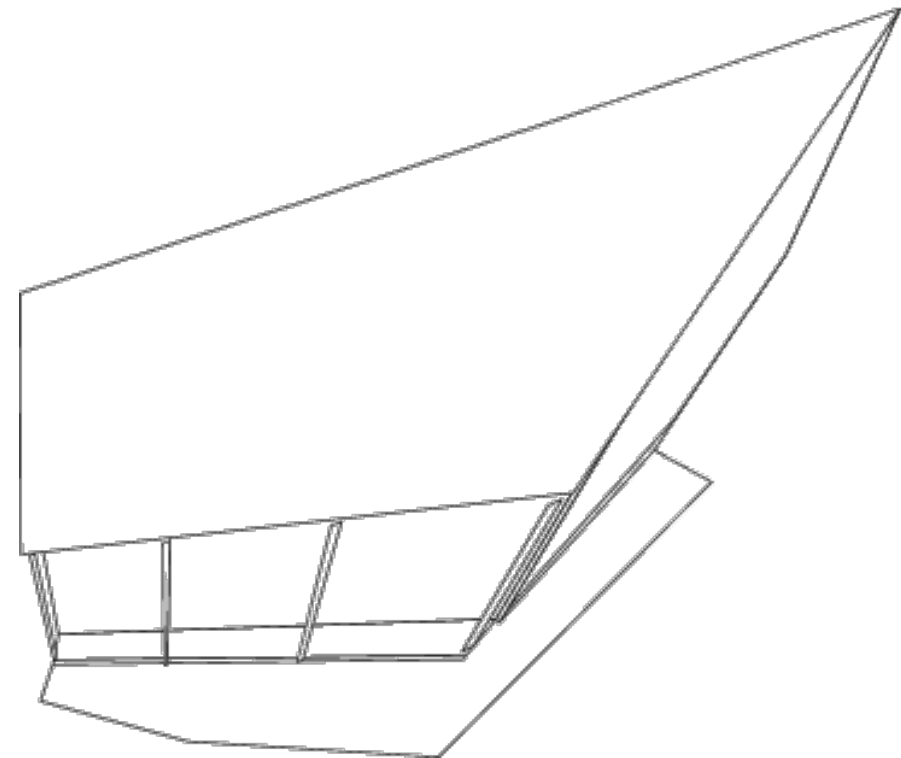
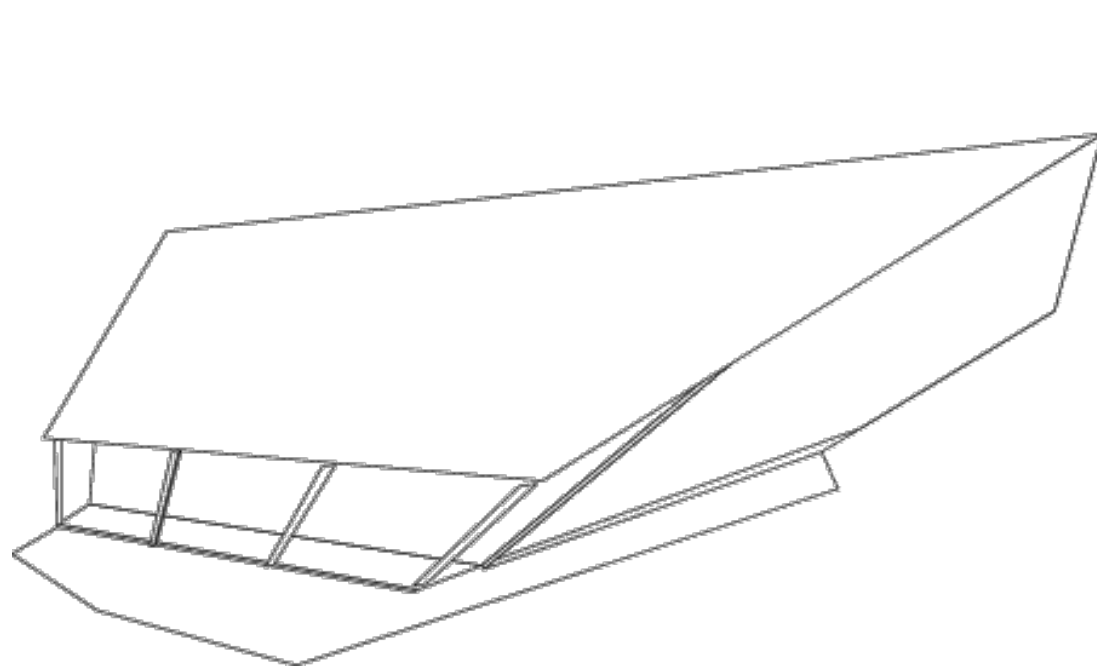
- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean

Projective ambiguity

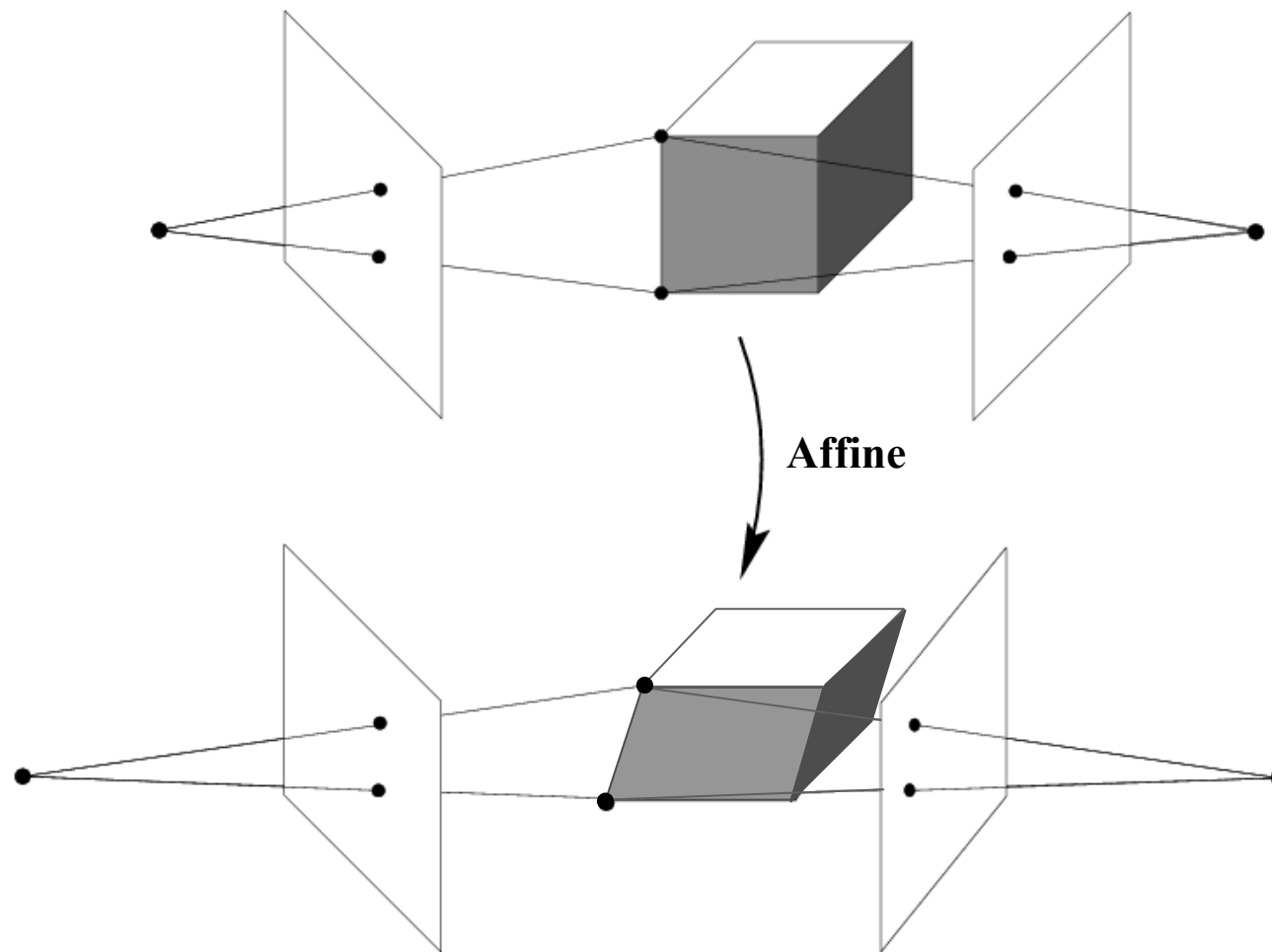


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_P^{-1}\right)\left(\mathbf{Q}_P\mathbf{X}\right)$$

Projective ambiguity

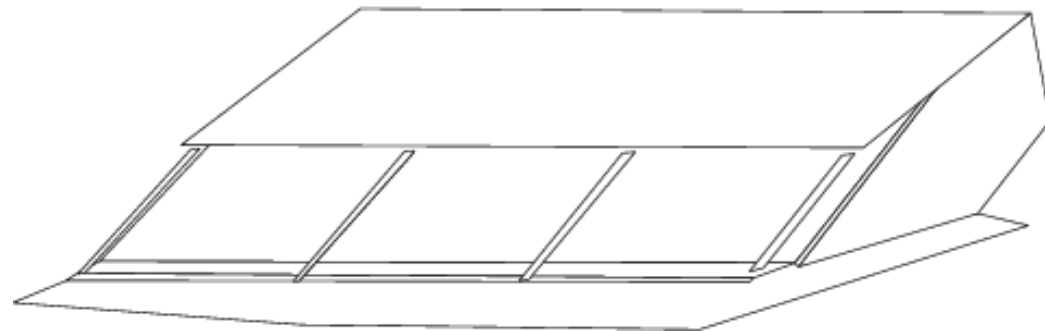
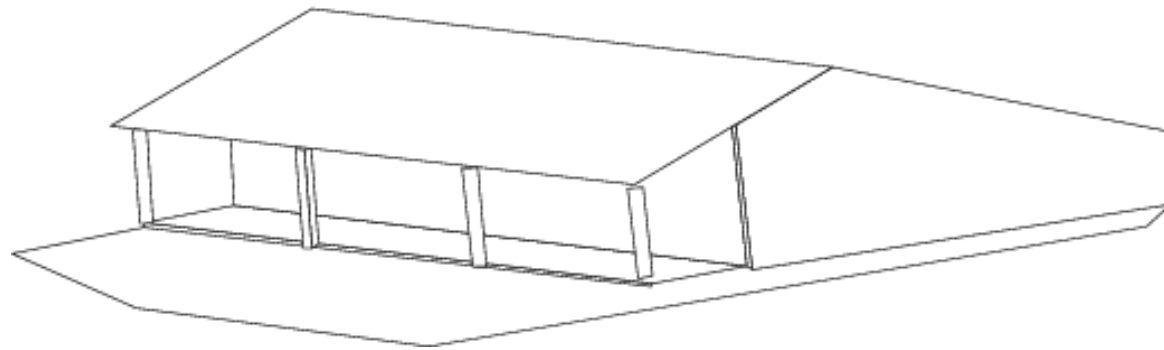


Affine ambiguity

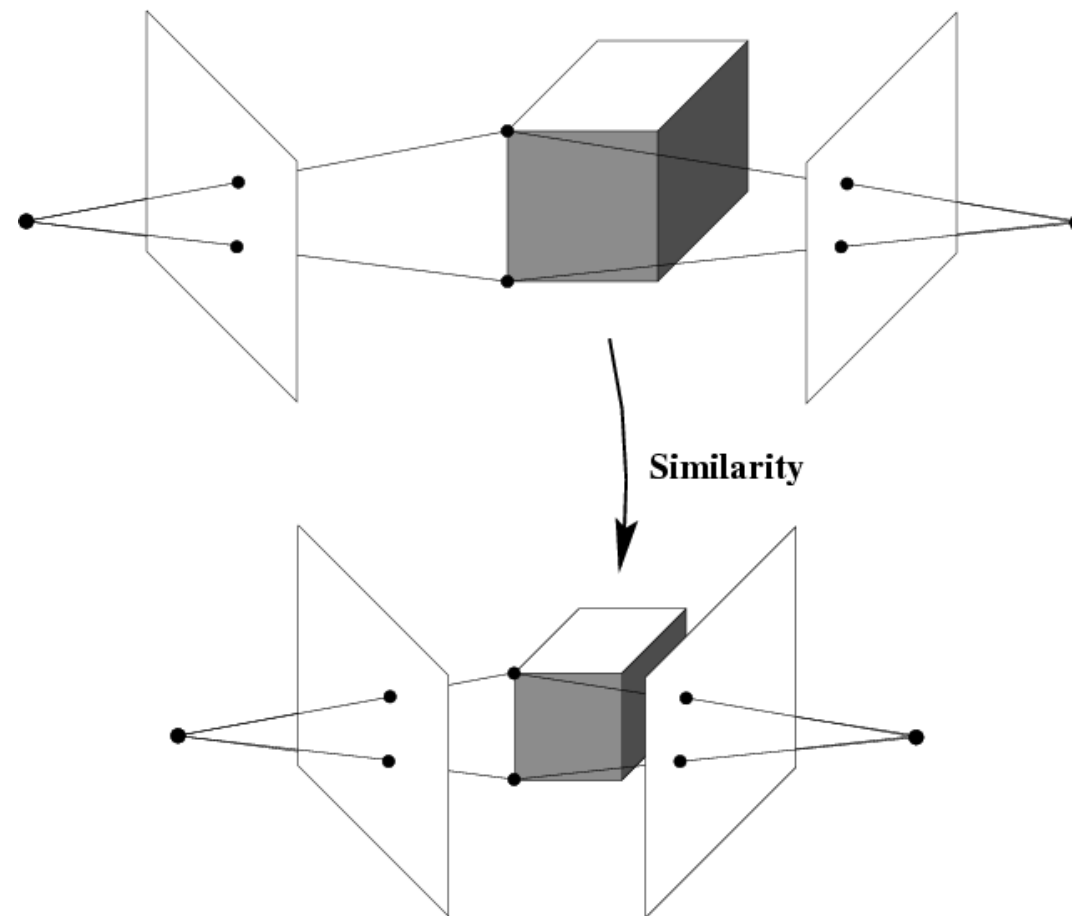


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_A^{-1}\right)\left(\mathbf{Q}_A\mathbf{X}\right)$$

Affine ambiguity

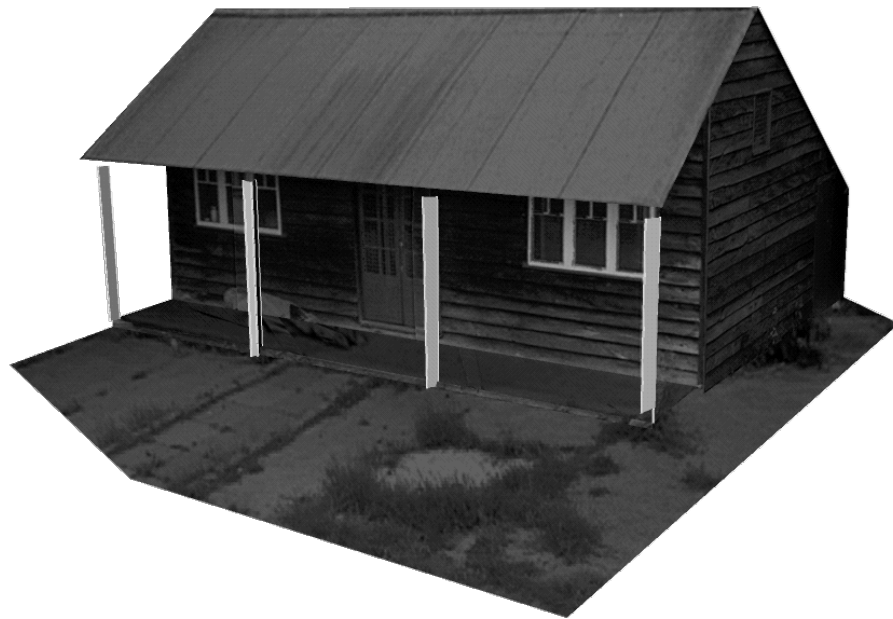
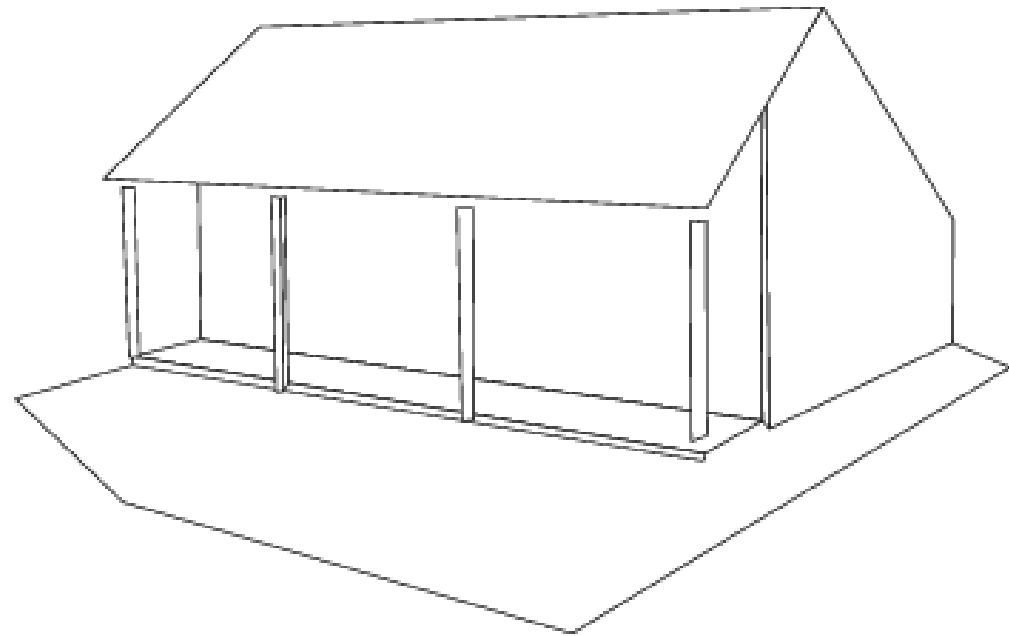
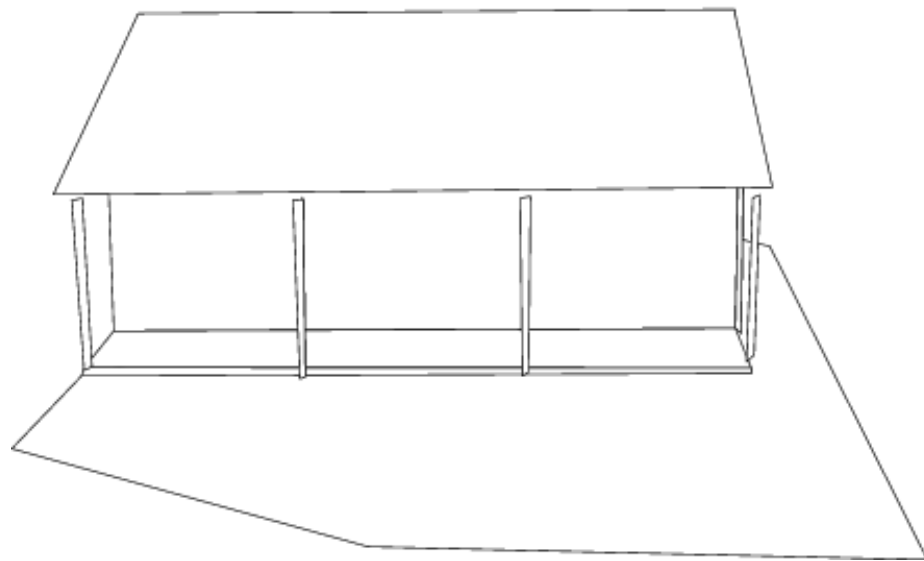


Similarity ambiguity



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

Similarity ambiguity

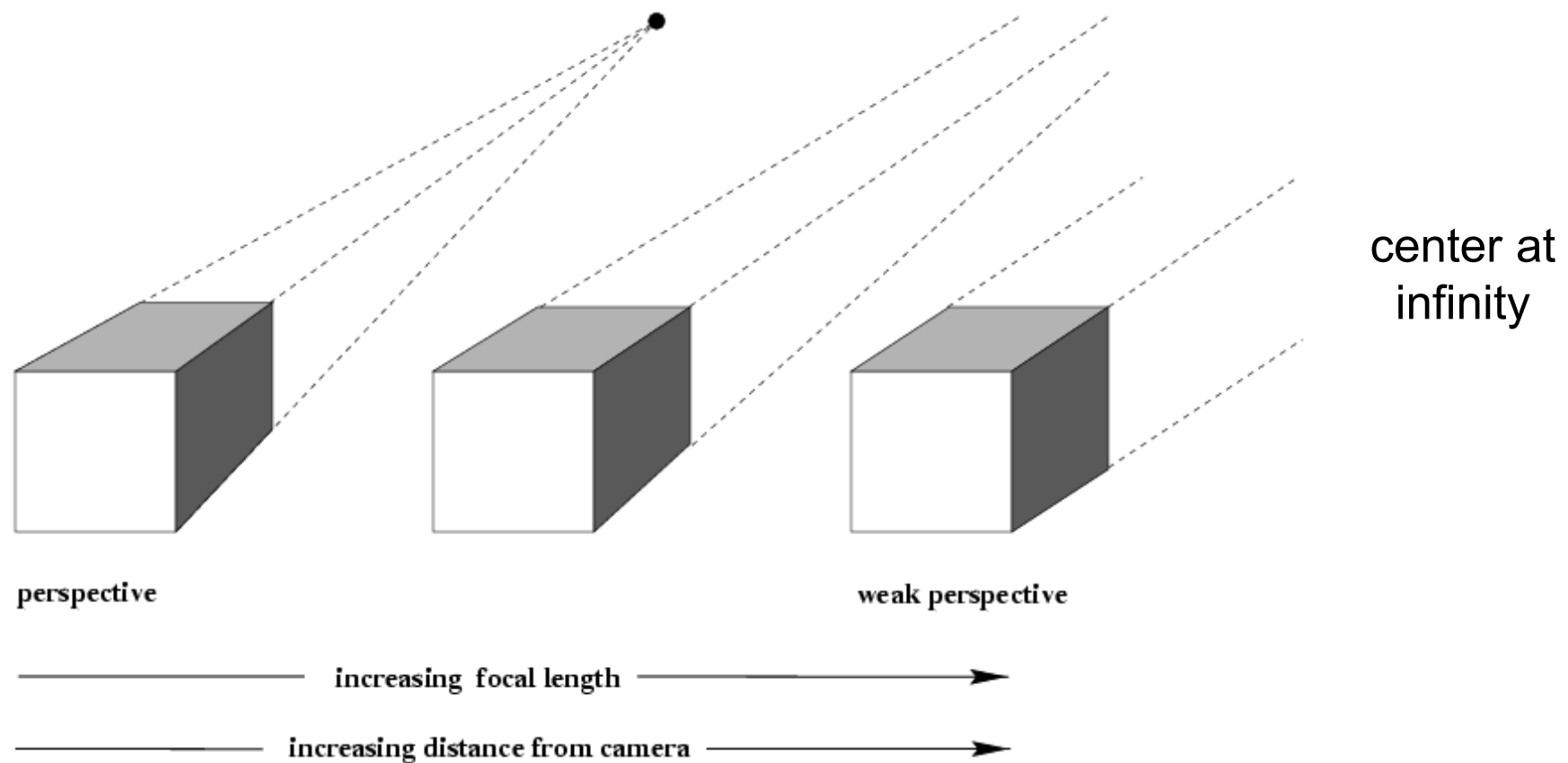


What can we do to remove ambiguities?

Affine structure from motion

Structure from motion

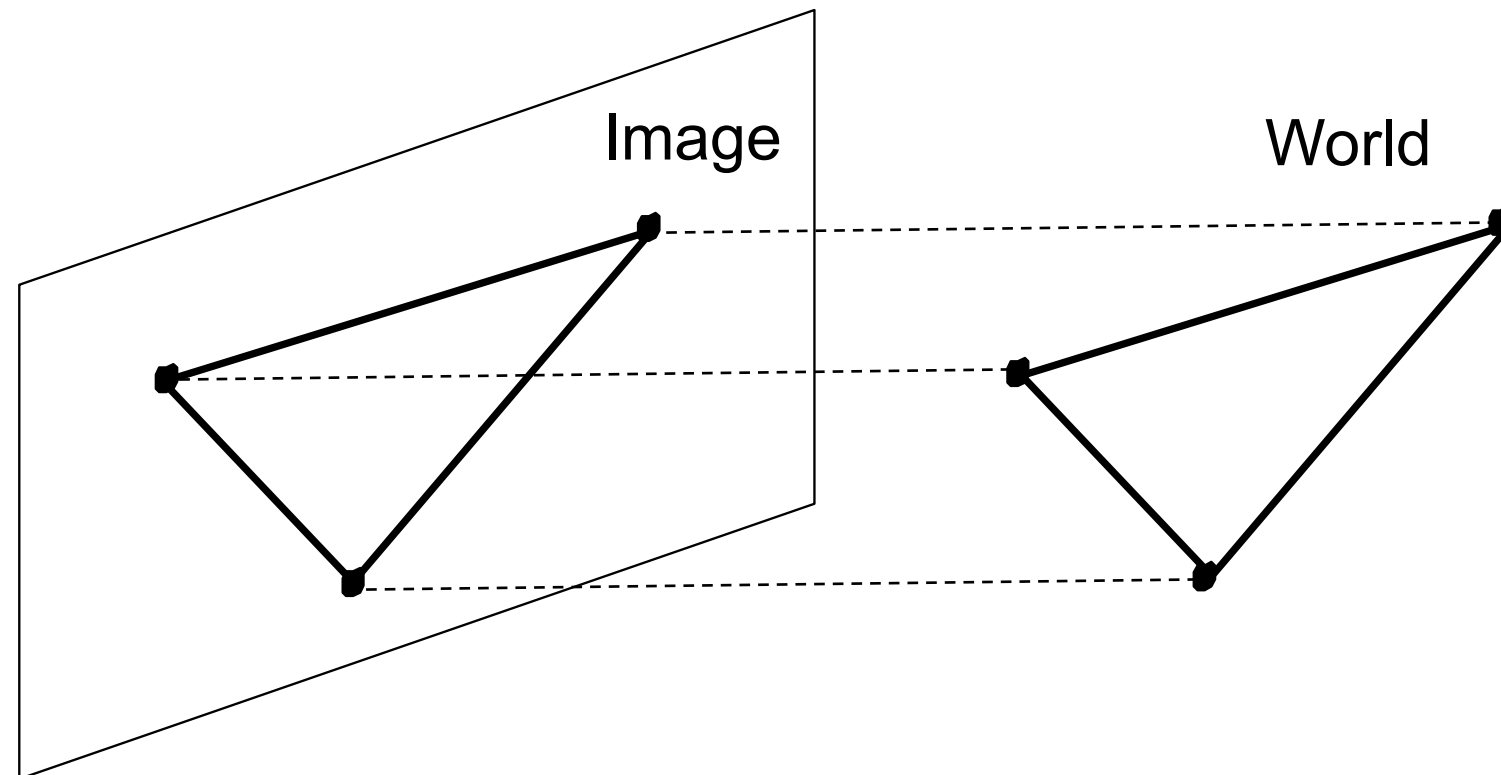
- Let's start with *affine cameras* (the math is easier)



Recall: Orthographic Projection

Special case of perspective projection

- Distance from center of projection to image plane is infinite

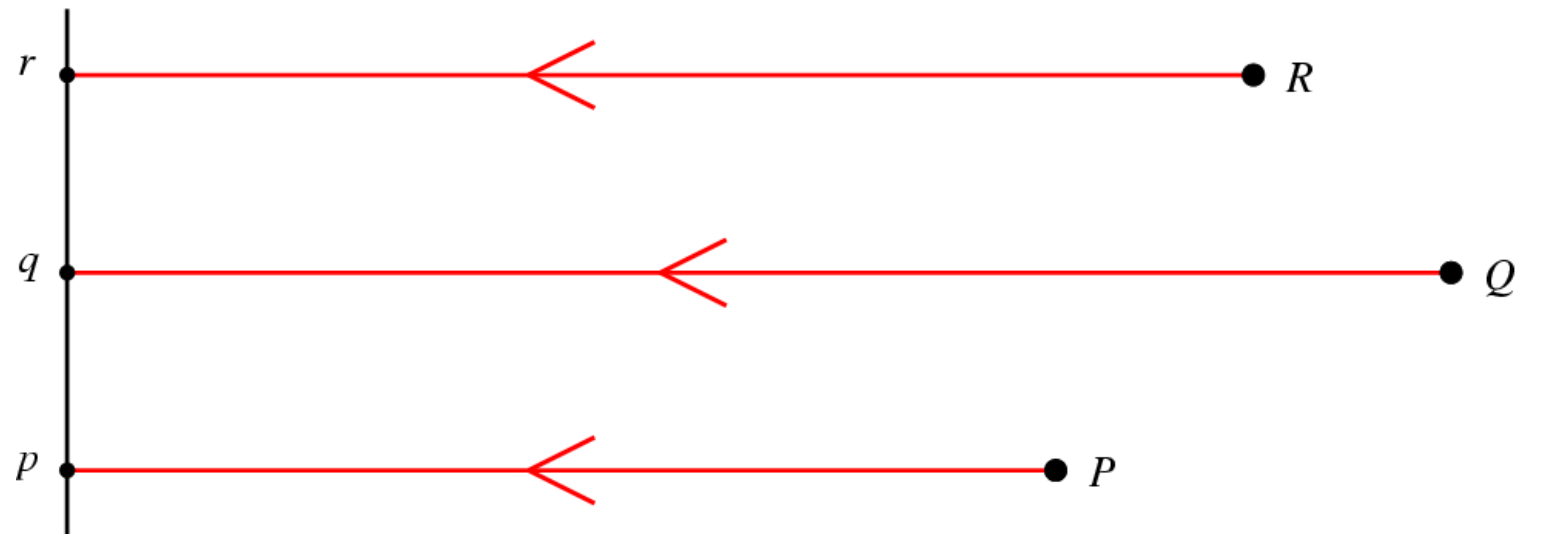


- Projection matrix:

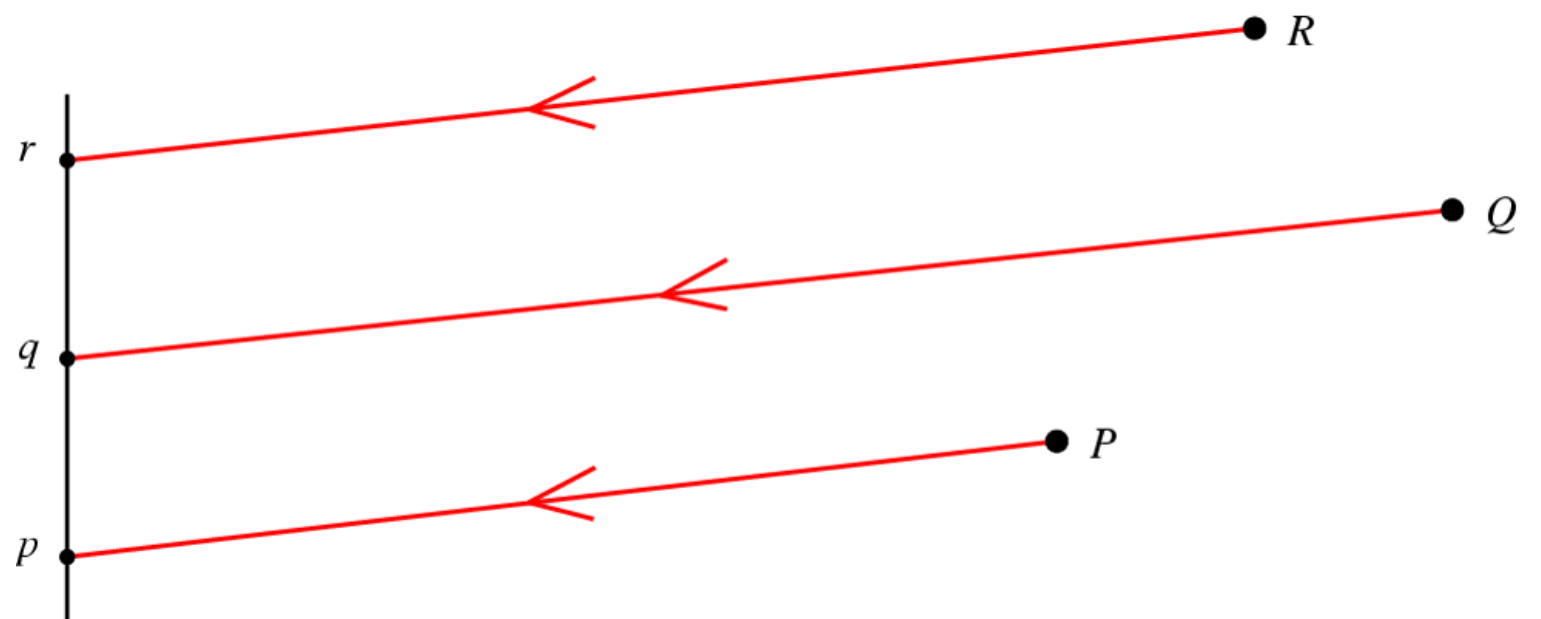
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Affine cameras

Orthographic Projection



Parallel Projection

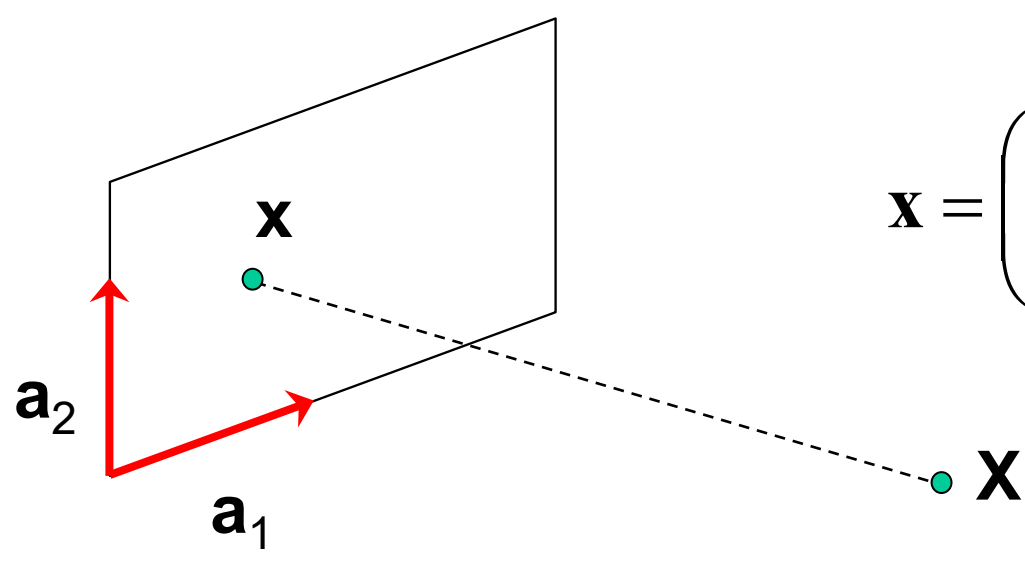


Affine cameras

- A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

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

- Affine projection is a linear mapping + translation in inhomogeneous coordinates



$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{A}\mathbf{X} + \mathbf{b}$$

Projection of world origin

Affine structure from motion

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

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

- 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 *affine* transformation $\mathbf{Q}$ (12 degrees of freedom):

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

- We have $2mn$ knowns and $8m + 3n$ unknowns (minus 12 dof for affine ambiguity)
- Thus, we must have $2mn \geq 8m + 3n - 12$
- For two views, we need four point correspondences

Affine structure from motion

- Centering: subtract the centroid of the image points

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

- For simplicity, 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 $\mathbf{X}_j$ by

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

Affine structure from motion

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

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


points (n)


cameras
($2m$)

Affine structure from motion

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

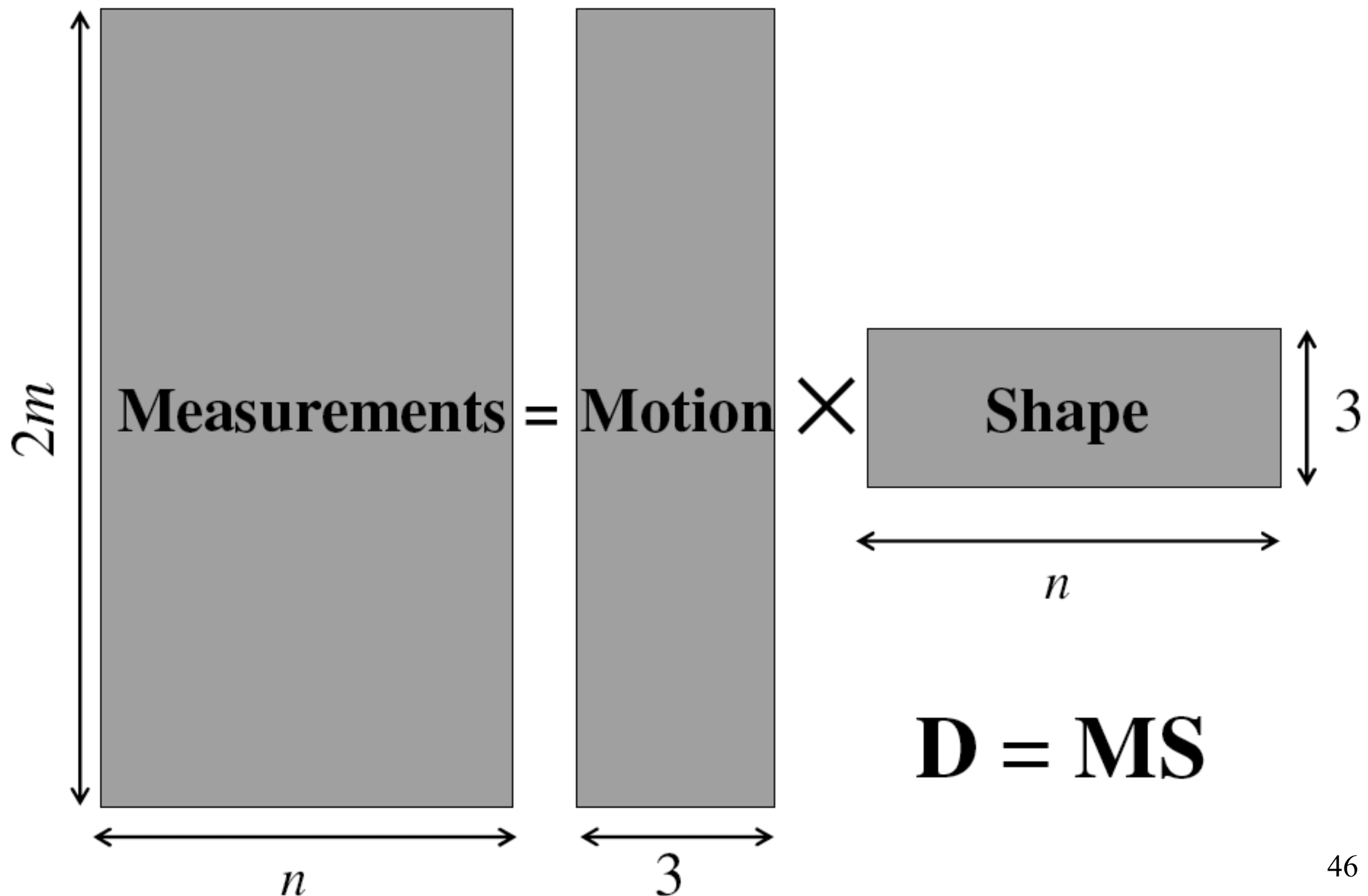
$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn}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

cameras
($2m \times 3$)

points ($3 \times n$)

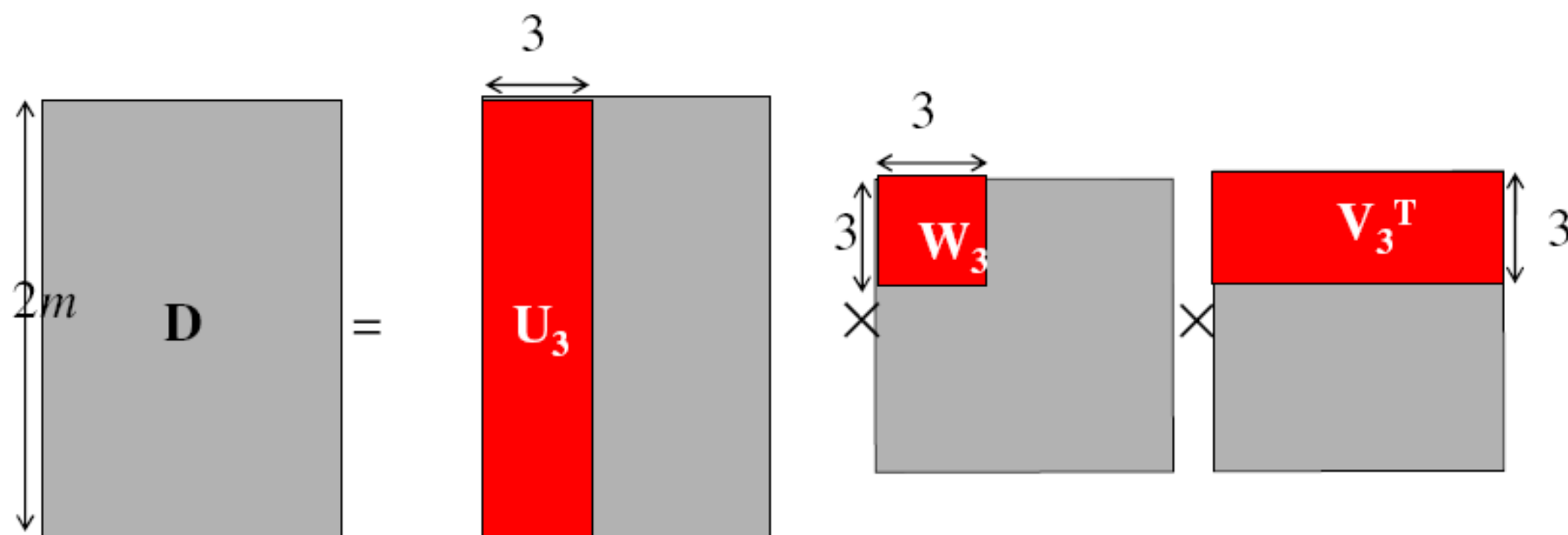
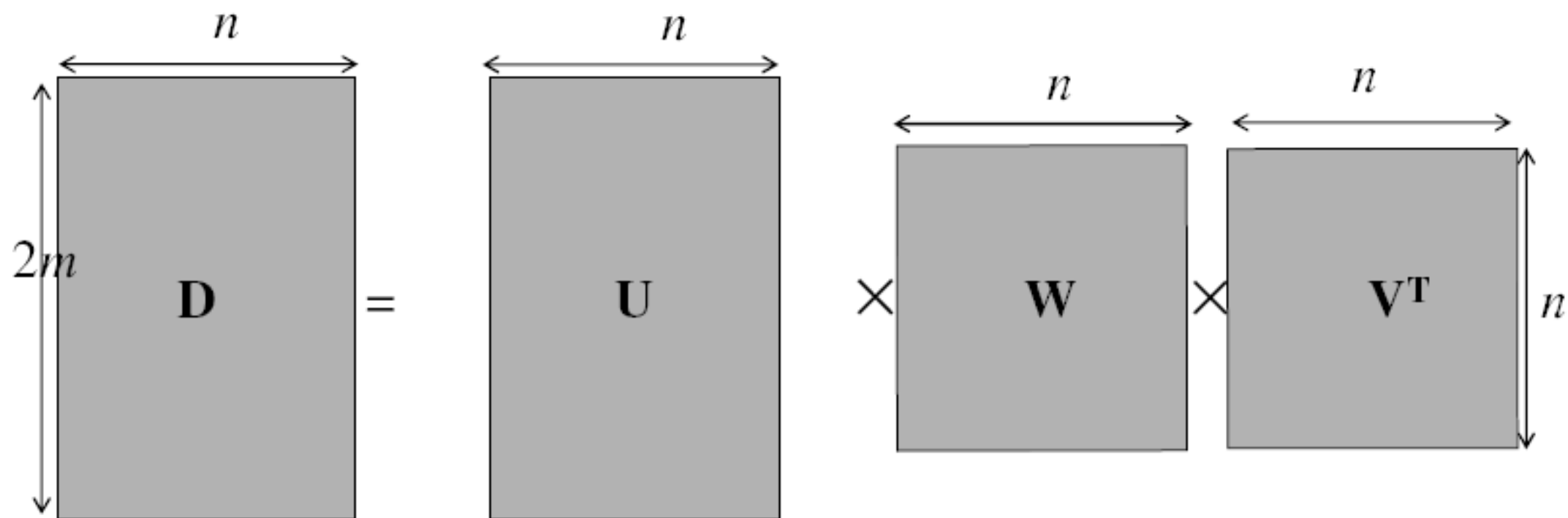
The measurement matrix $\mathbf{D} = \mathbf{MS}$ must have rank 3!

Factorizing the measurement matrix



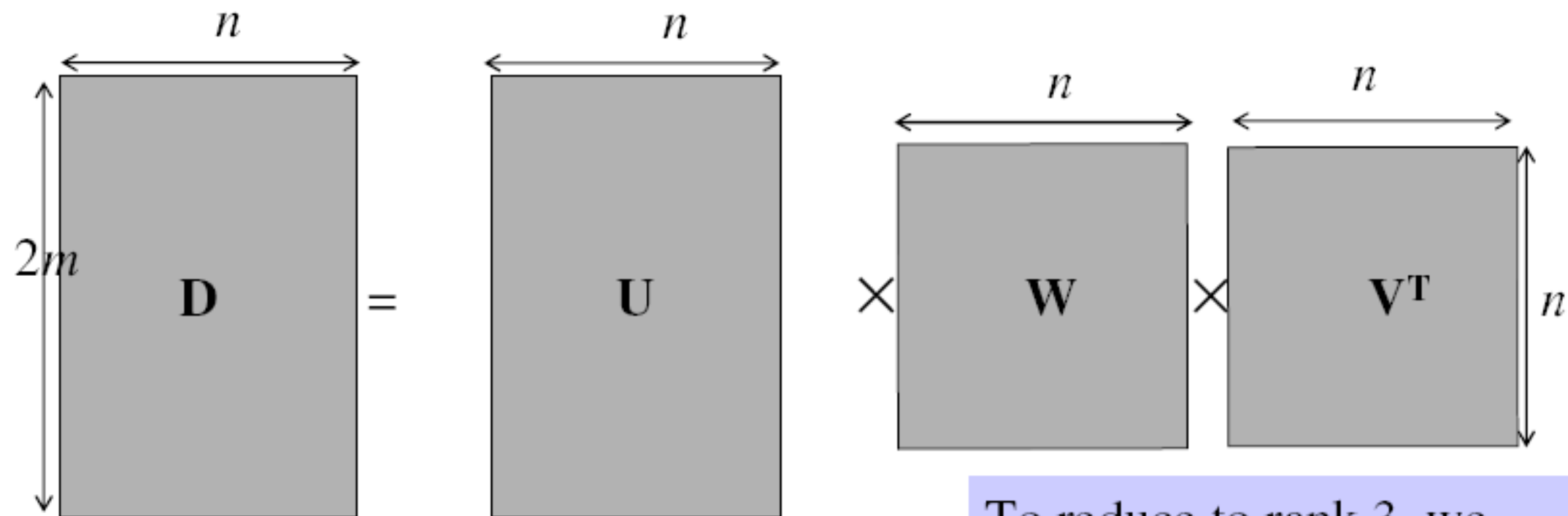
Factorizing the measurement matrix

- Singular value decomposition of D :

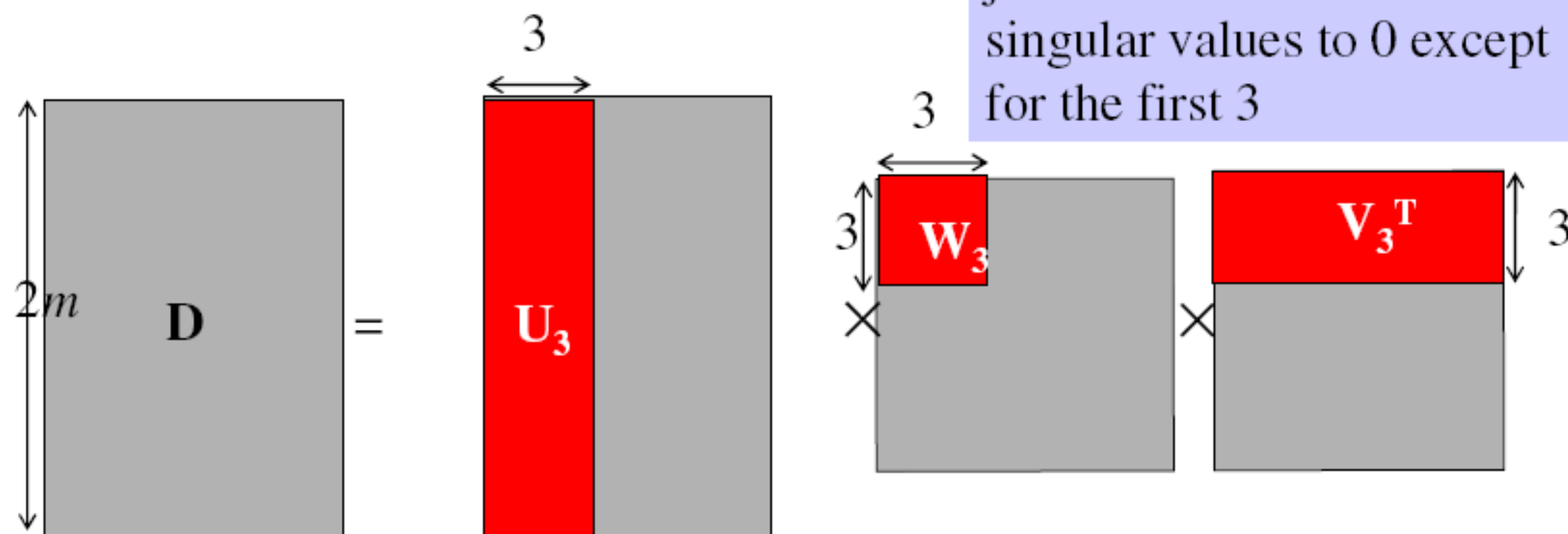


Factorizing the measurement matrix

- Singular value decomposition of D :

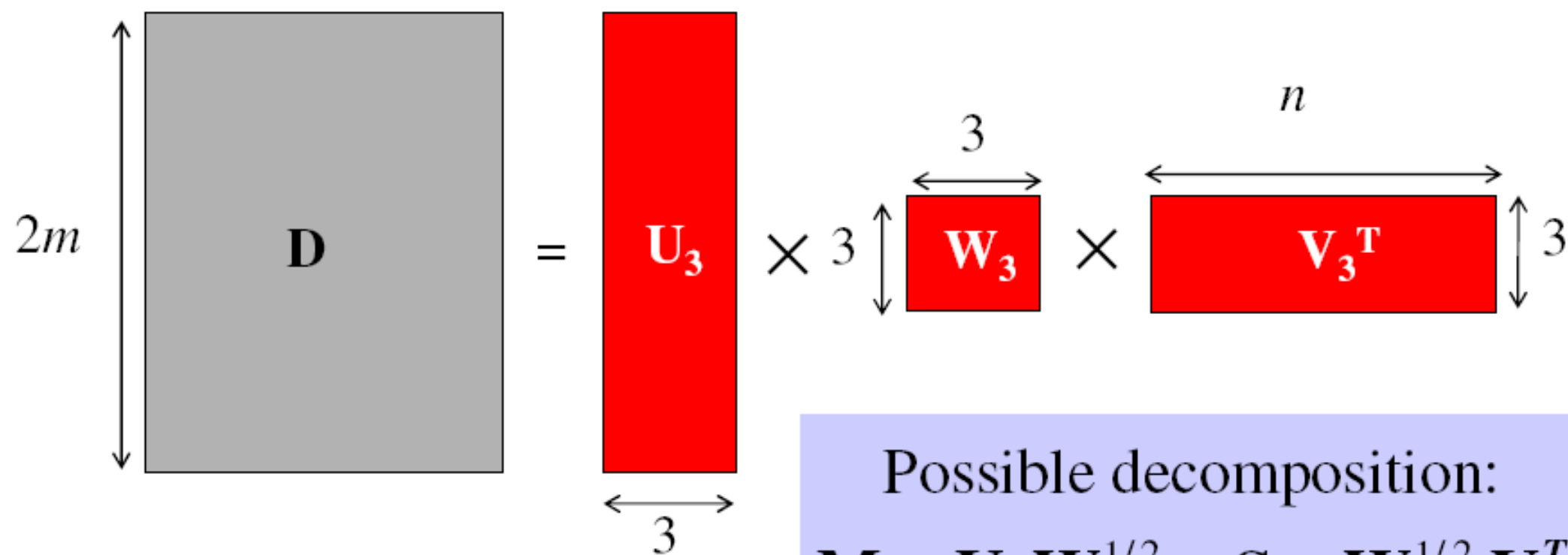


To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3



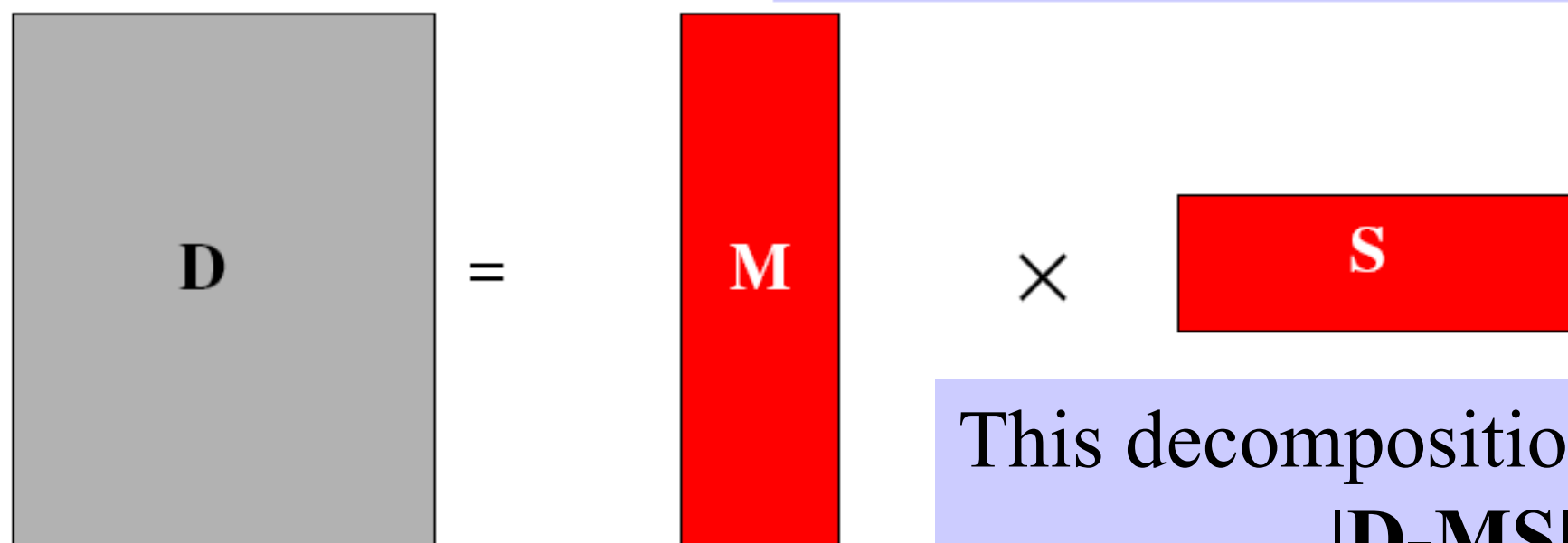
Factorizing the measurement matrix

- Obtaining a factorization from SVD:



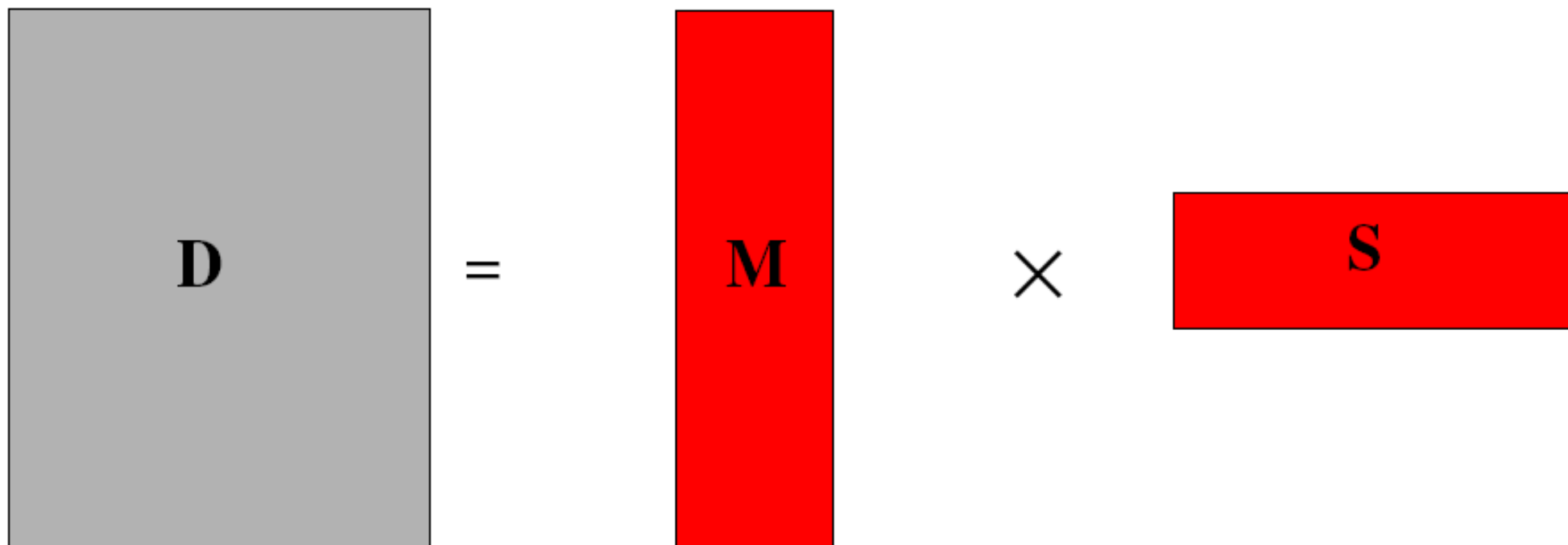
Possible decomposition:

$$\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \quad \mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T$$



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

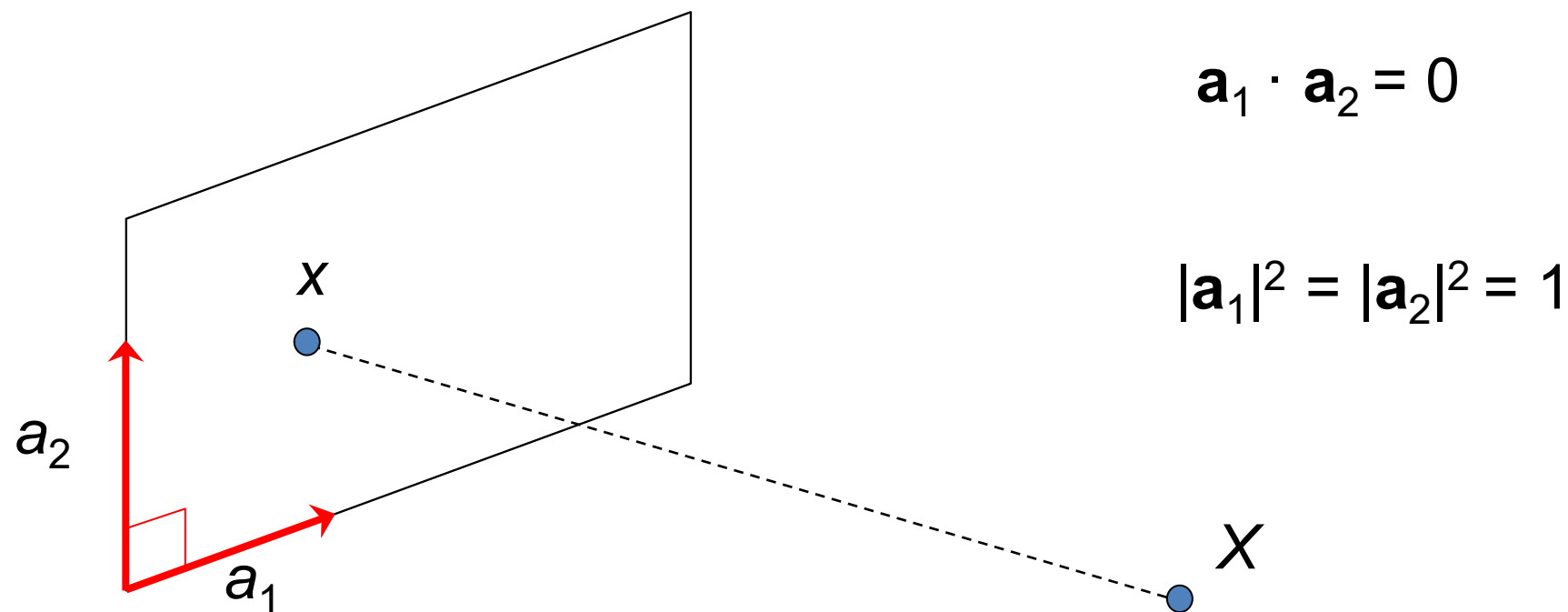
Affine ambiguity


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

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

Eliminating the affine ambiguity

- Orthographic: image axes are perpendicular and of unit length



Solve for orthographic constraints

Three equations for each image i

$$\begin{aligned}\tilde{\mathbf{a}}_{i1}^T \mathbf{C} \mathbf{C}^T \tilde{\mathbf{a}}_{i1} &= 1 \\ \tilde{\mathbf{a}}_{i2}^T \mathbf{C} \mathbf{C}^T \tilde{\mathbf{a}}_{i2} &= 1 \\ \tilde{\mathbf{a}}_{i1}^T \mathbf{C} \mathbf{C}^T \tilde{\mathbf{a}}_{i2} &= 0\end{aligned} \quad \text{where} \quad \tilde{\mathbf{A}}_i = \begin{bmatrix} \tilde{\mathbf{a}}_{i1}^T \\ \tilde{\mathbf{a}}_{i2}^T \end{bmatrix}$$

- Solve for $\mathbf{L} = \mathbf{C} \mathbf{C}^T$
- Recover $\mathbf{C}$ from $\mathbf{L}$ by Cholesky decomposition: $\mathbf{L} = \mathbf{C} \mathbf{C}^T$
- Update $\mathbf{A}$ and $\mathbf{X}$: $\mathbf{A} = \tilde{\mathbf{A}} \mathbf{C}$, $\mathbf{X} = \mathbf{C}^{-1} \tilde{\mathbf{X}}$

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 $\mathbf{D}$:
 - 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 $\mathbf{D}$:
 - Compute SVD: $\mathbf{D} = \mathbf{U} \mathbf{W} \mathbf{V}^T$
 - Create $\mathbf{U}_3$ by taking the first 3 columns of $\mathbf{U}$
 - Create $\mathbf{V}_3$ by taking the first 3 columns of $\mathbf{V}$
 - Create $\mathbf{W}_3$ by taking the upper left 3×3 block of $\mathbf{W}$
- Create the motion and shape matrices:
 - $\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2}$ and $\mathbf{S} = \mathbf{W}_3^{1/2} \mathbf{V}_3^T$ (or $\mathbf{M} = \mathbf{U}_3$ and $\mathbf{S} = \mathbf{W}_3 \mathbf{V}_3^T$)
- Eliminate affine ambiguity

Reconstruction results



1



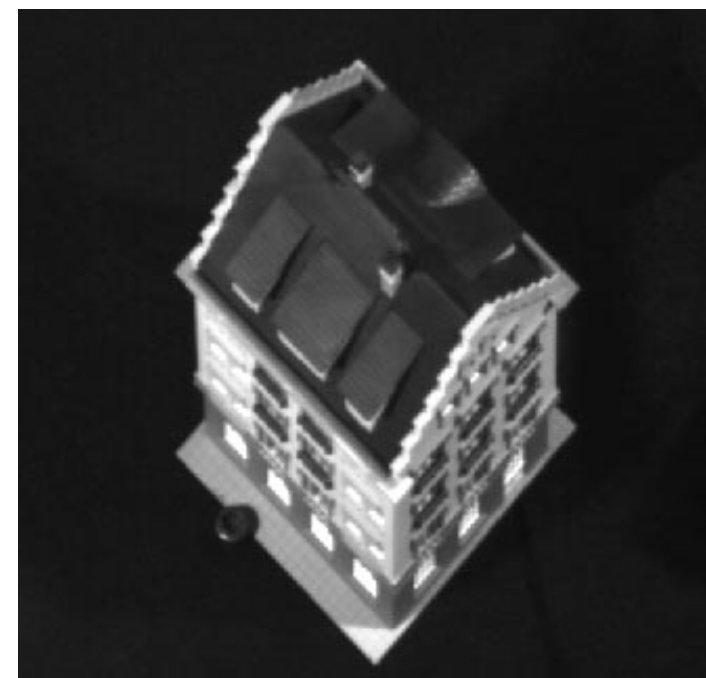
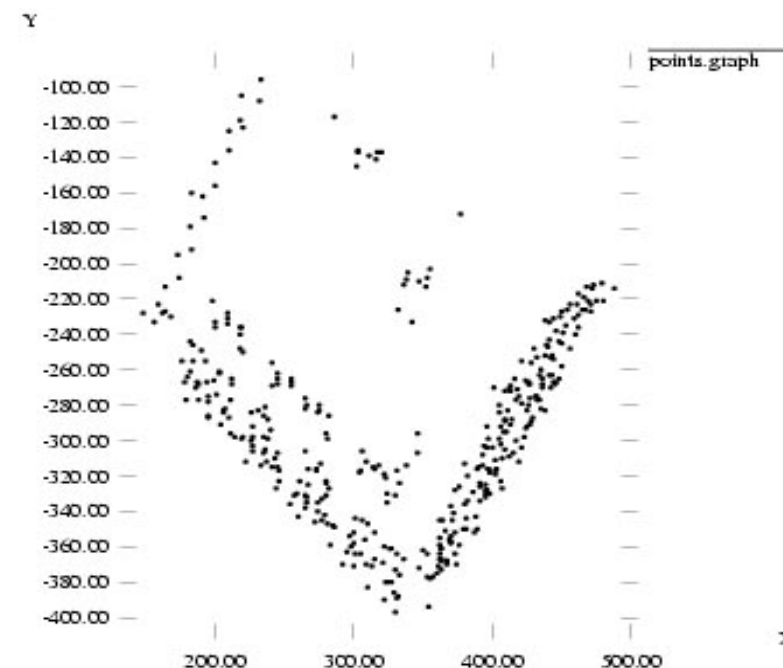
60



120



150



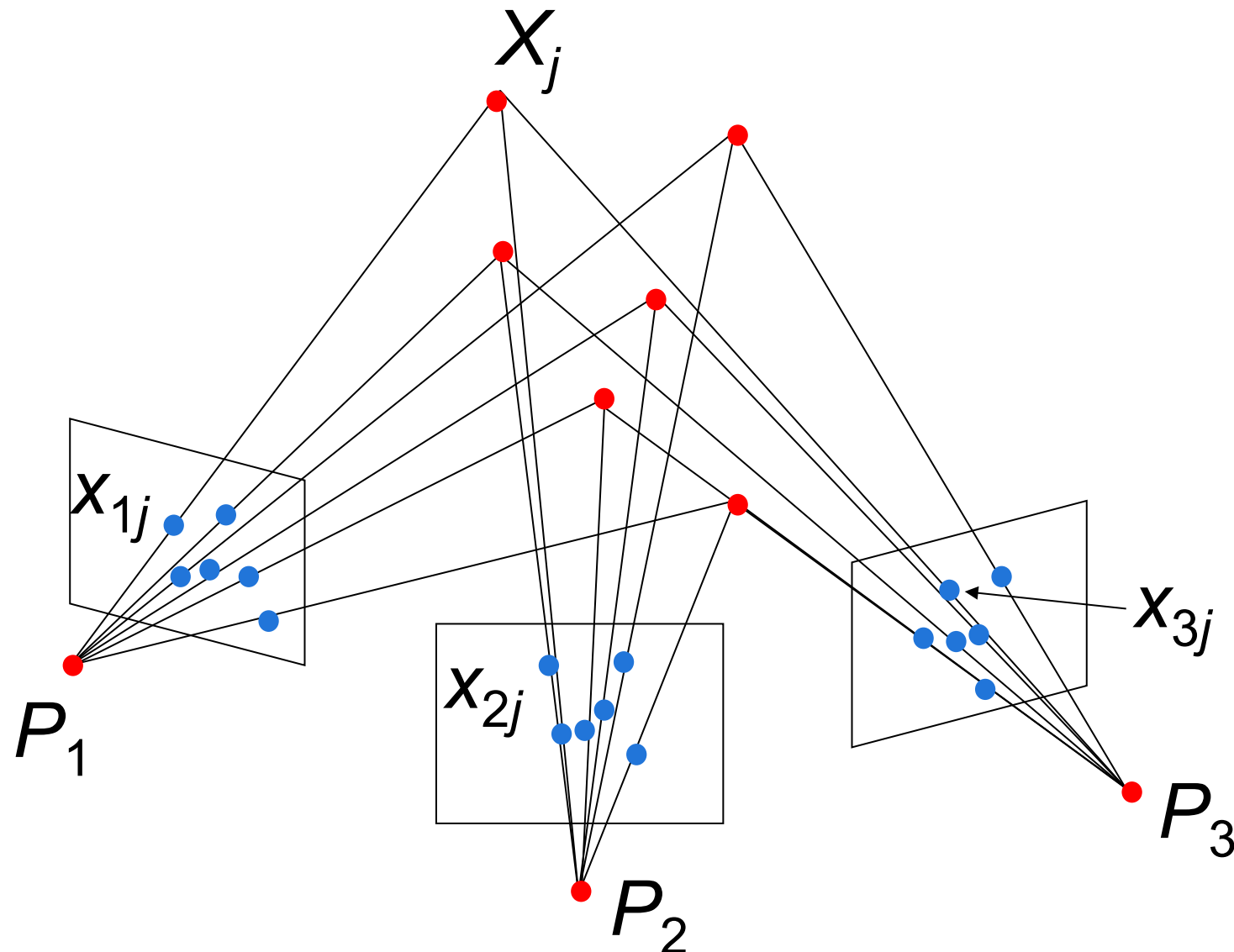
Multi-view projective structure from motion

Projective structure from motion

- Given: m images of n fixed 3D points

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

- Problem: estimate m projection matrices $\mathbf{P}_i$ and n 3D points $\mathbf{X}_j$ from the mn correspondences $\mathbf{x}_{ij}$



Projective structure from motion

- Given: m images of n fixed 3D points

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

- Problem: estimate m projection matrices $\mathbf{P}_i$ 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 $\mathbf{Q}$:

$$\mathbf{X} \rightarrow \mathbf{QX}, \quad \mathbf{P} \rightarrow \mathbf{PQ}^{-1}$$

- We can solve for structure and motion when

$$2mn \geq 11m + 3n - 15$$

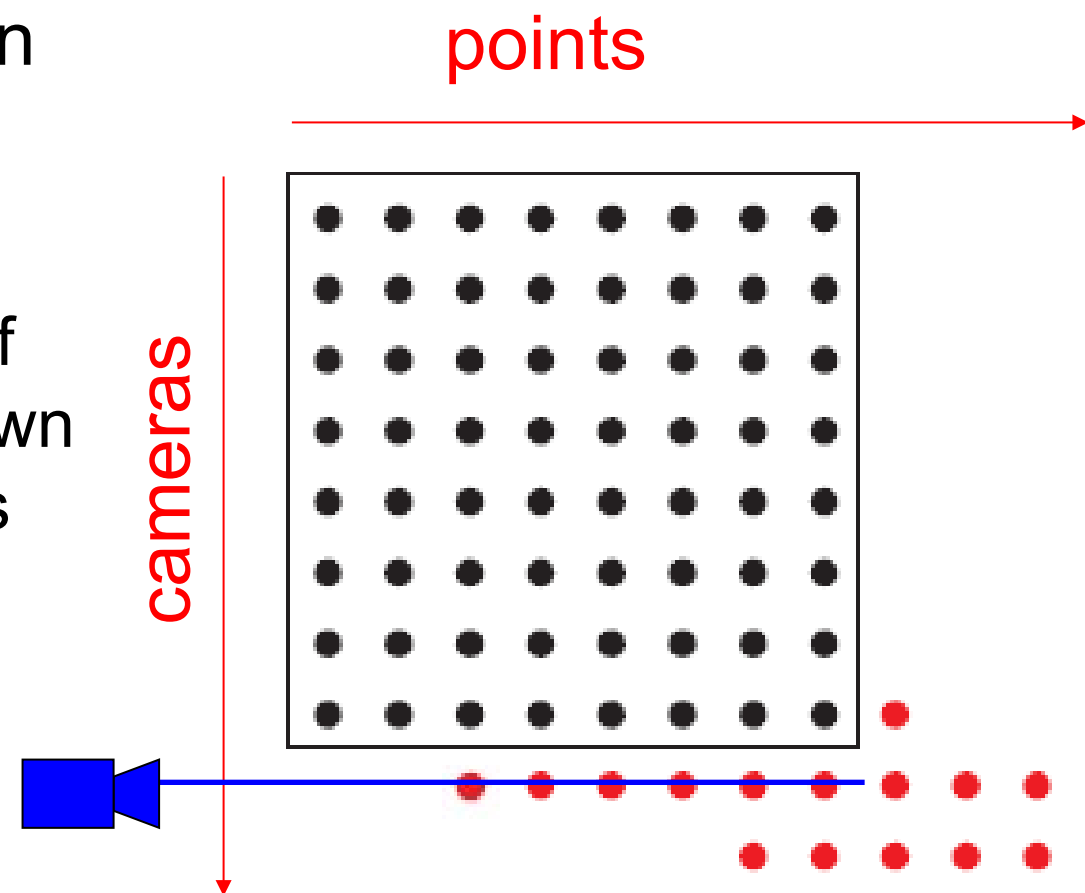
- For two cameras, at least 7 points are needed

Projective SFM: Two-camera case

- Compute fundamental matrix $\mathbf{F}$ between the two views
- First camera matrix: $[\mathbf{I}|\mathbf{0}]$
- Second camera matrix: $[\mathbf{A}|\mathbf{b}]$
- Then $\mathbf{b}$ is the epipole ($\mathbf{F}^T \mathbf{b} = 0$), $\mathbf{A} = -[\mathbf{b}_\times] \mathbf{F}$

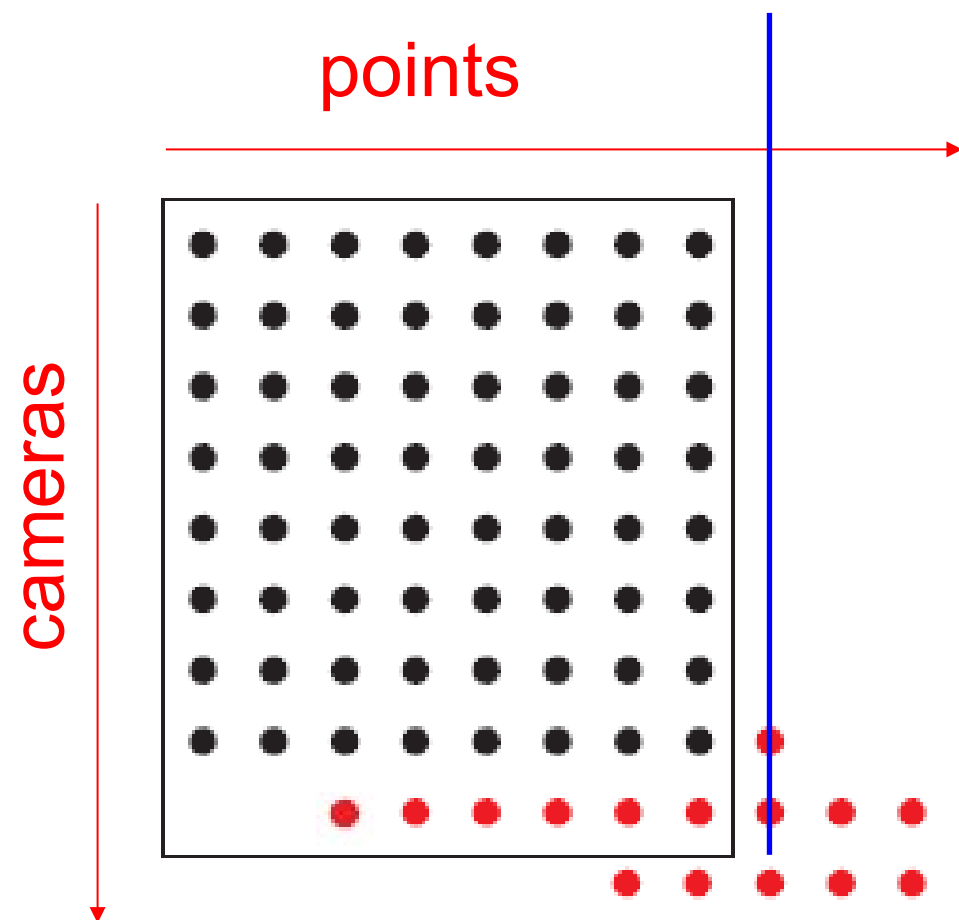
Sequential structure from motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
 - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*



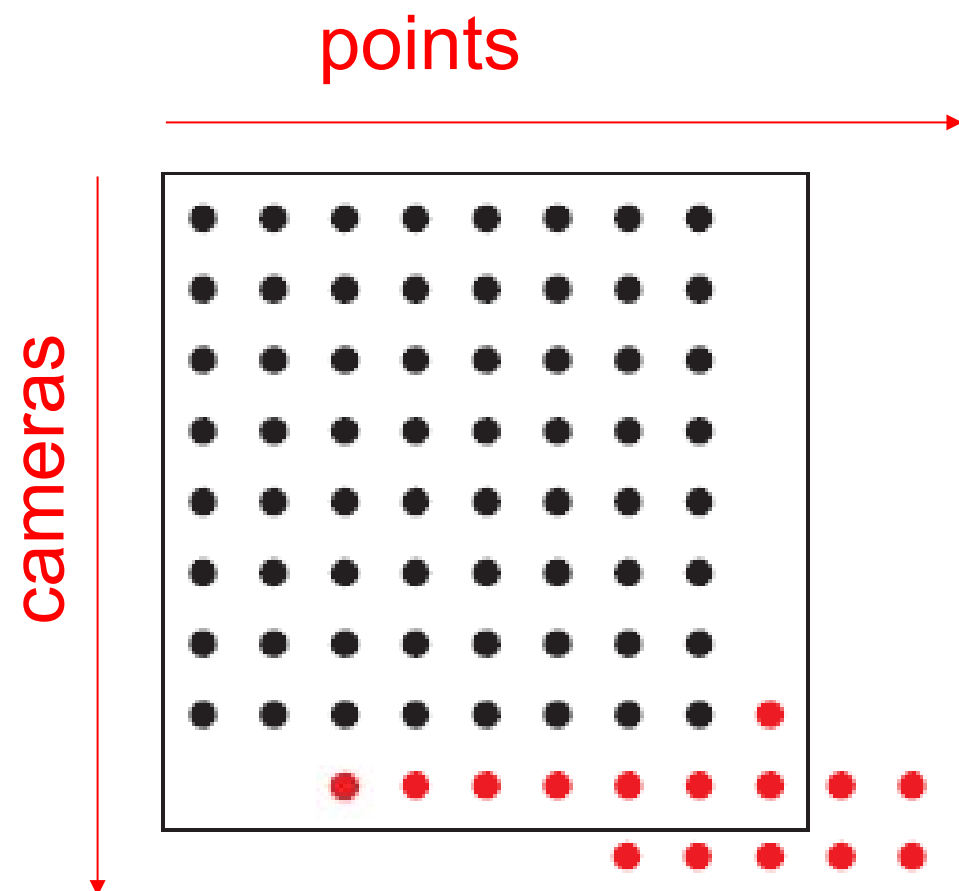
Sequential structure from motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
 - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
 - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*



Sequential structure from motion

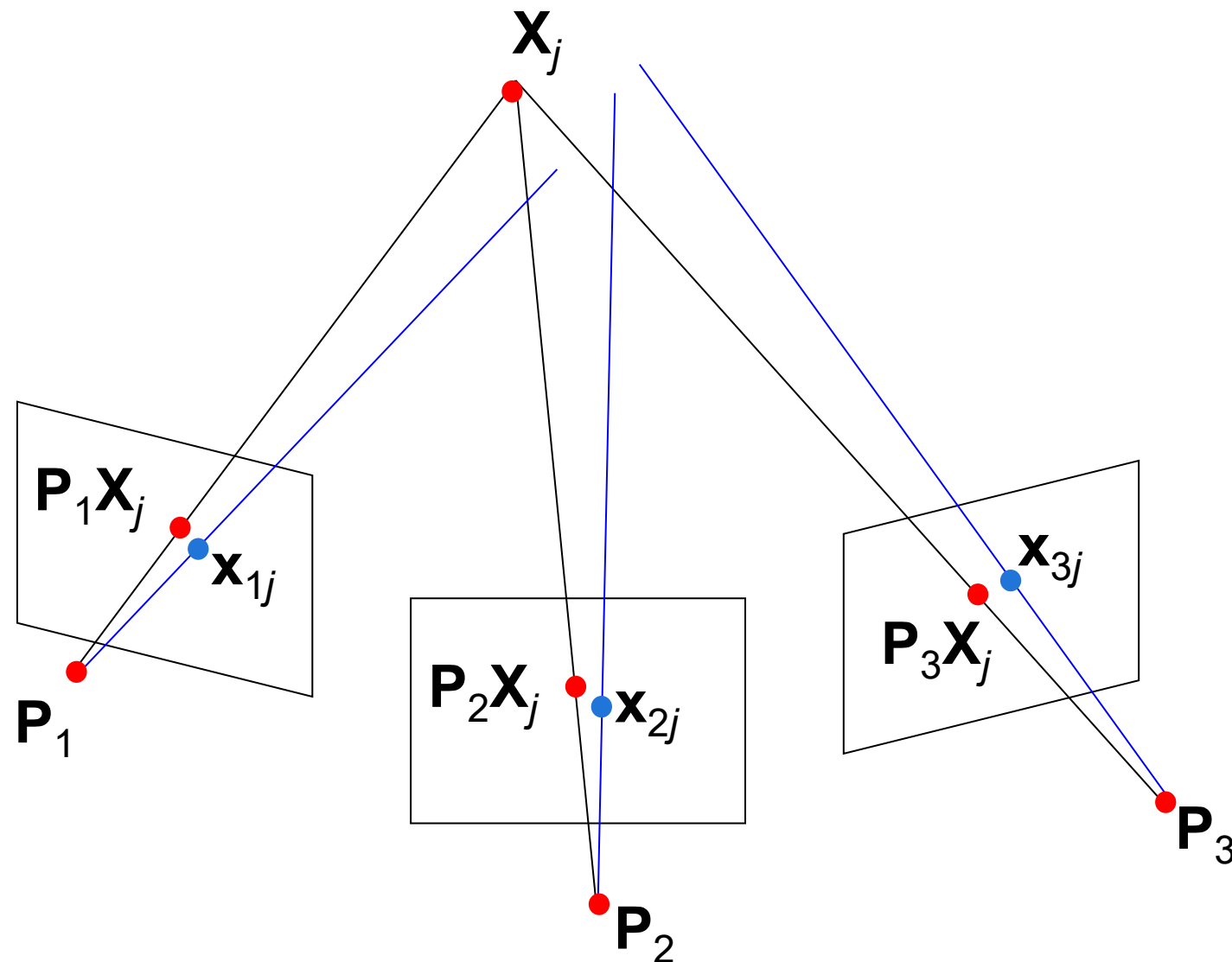
- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
 - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
 - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*
- Refine structure and motion: bundle adjustment



Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

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



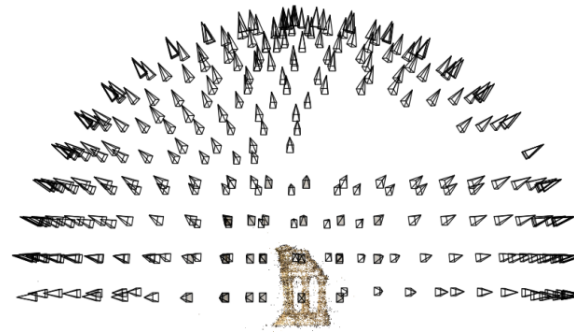
Review: Structure from motion

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

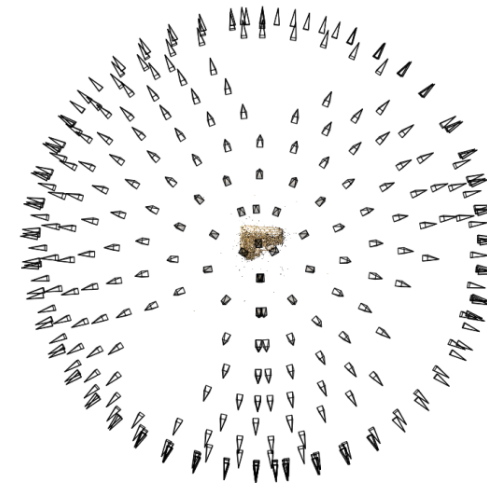
	Structure (scene geometry)	Motion (camera geometry)	Measurements
Camera calibration	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D coorespondences
Reconstruction	estimate	estimate	2D to 2D correspondences

Large-scale structure from motion

Structure from motion



Reconstruction (side)



(top)

- Input: images with points in correspondence
 $p_{i,j} = (u_{i,j}, v_{i,j})$
- Output
 - structure: 3D location $\mathbf{x}_i$ for each point p_i
 - motion: camera parameters $\mathbf{R}_j$, $\mathbf{t}_j$ possibly $\mathbf{K}_j$
- Objective function: minimize *reprojection error*



15,464



37,383



76,389

Standard way to view photos

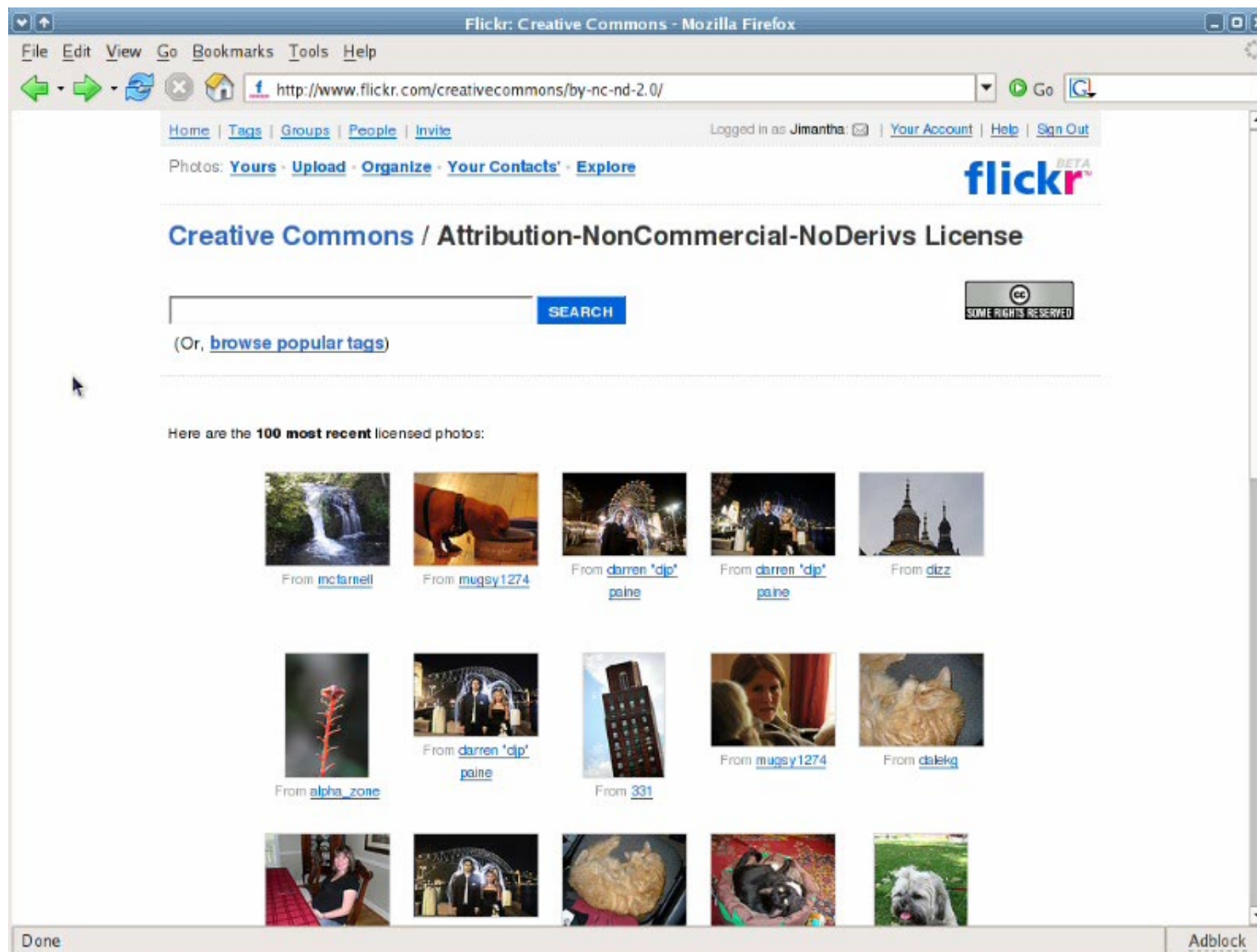
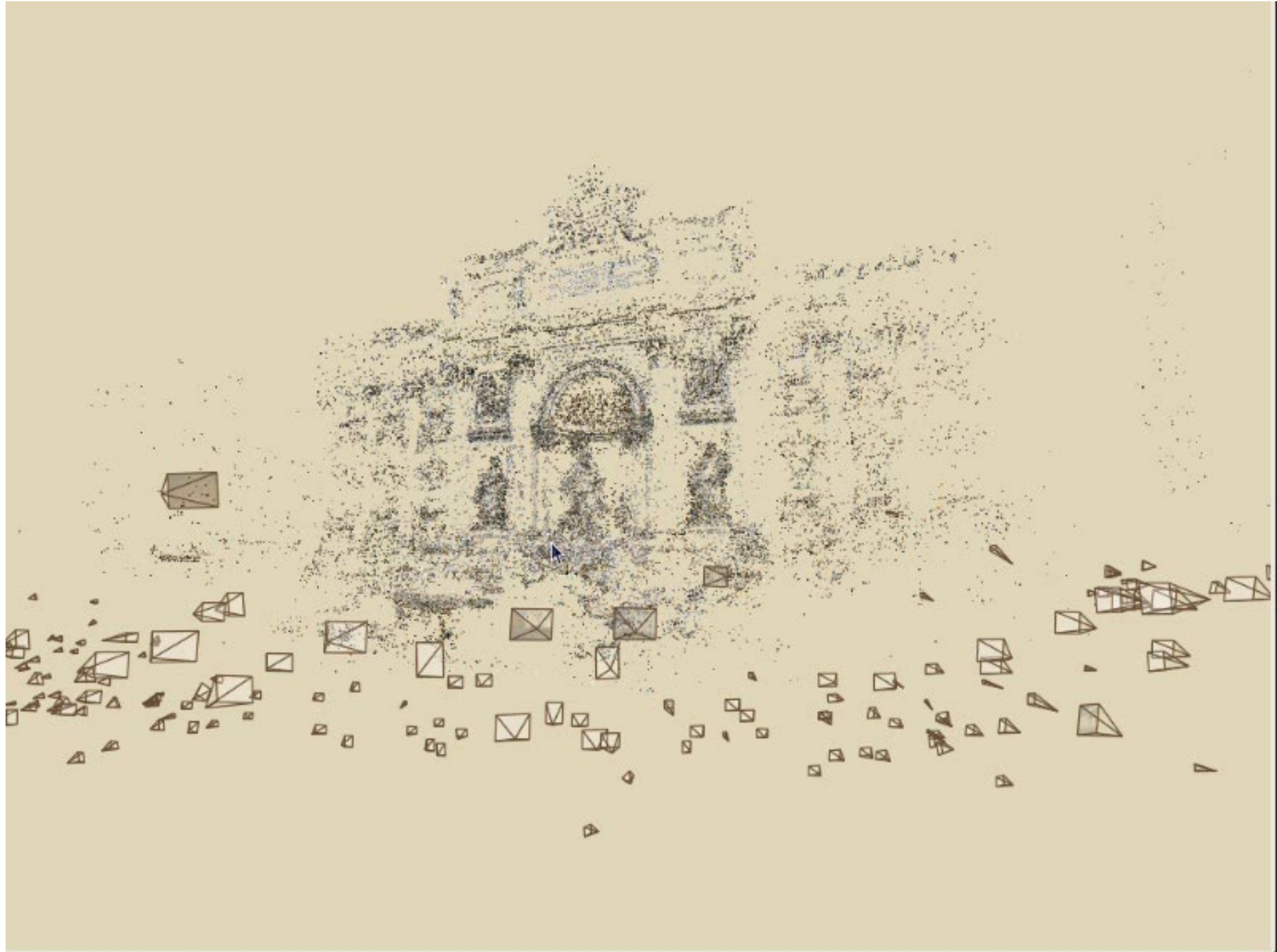
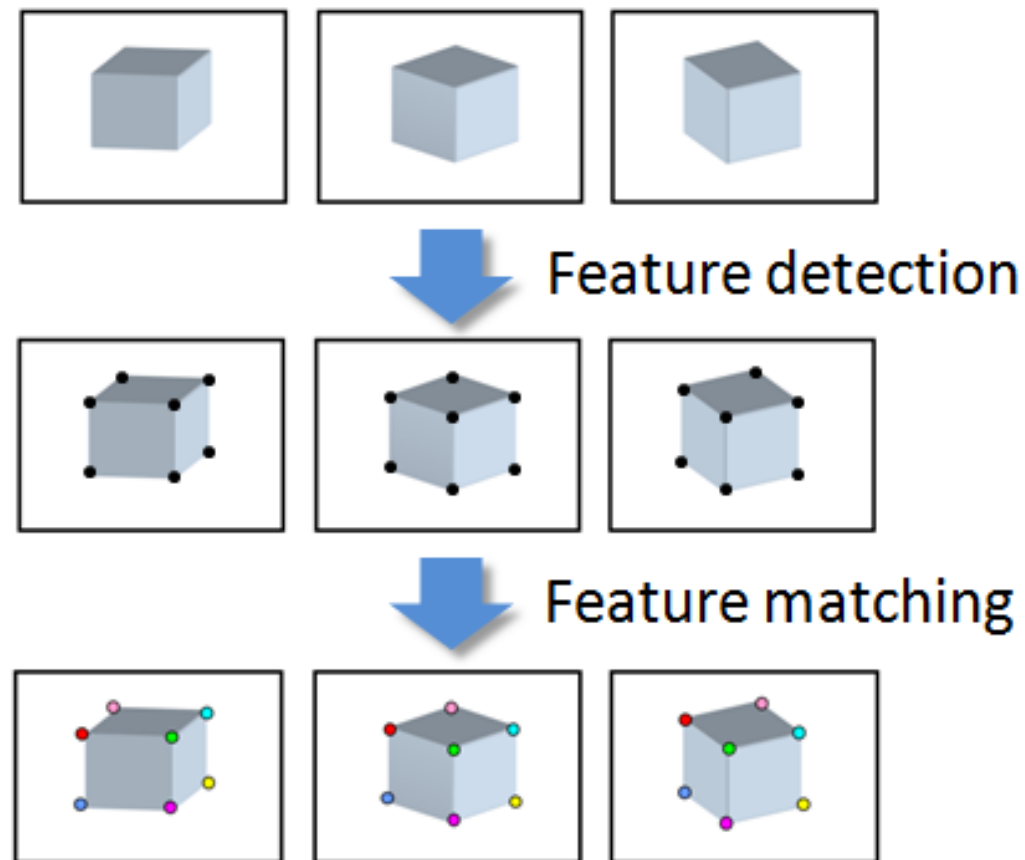


Photo Tourism

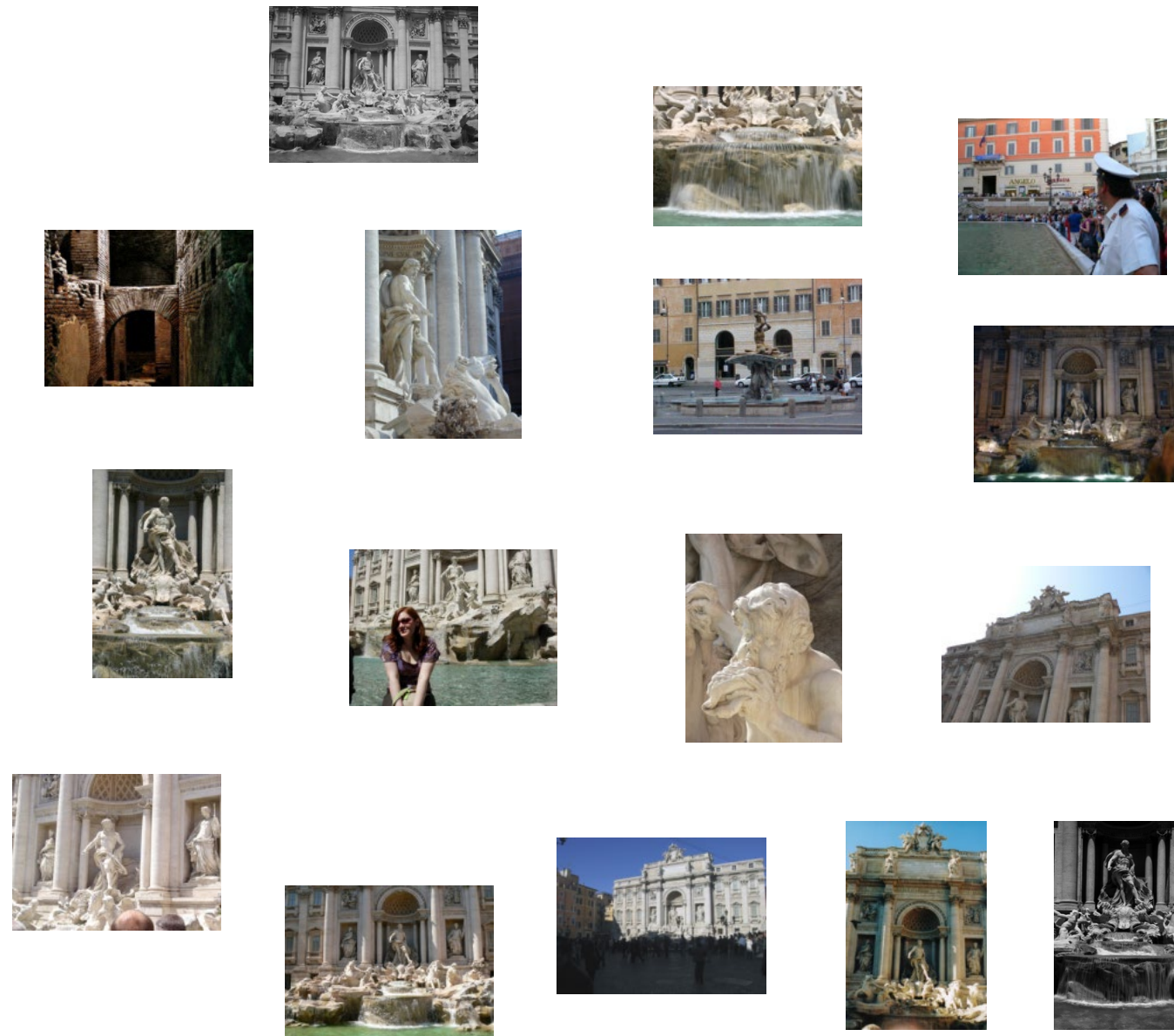


Input: Point correspondences



Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



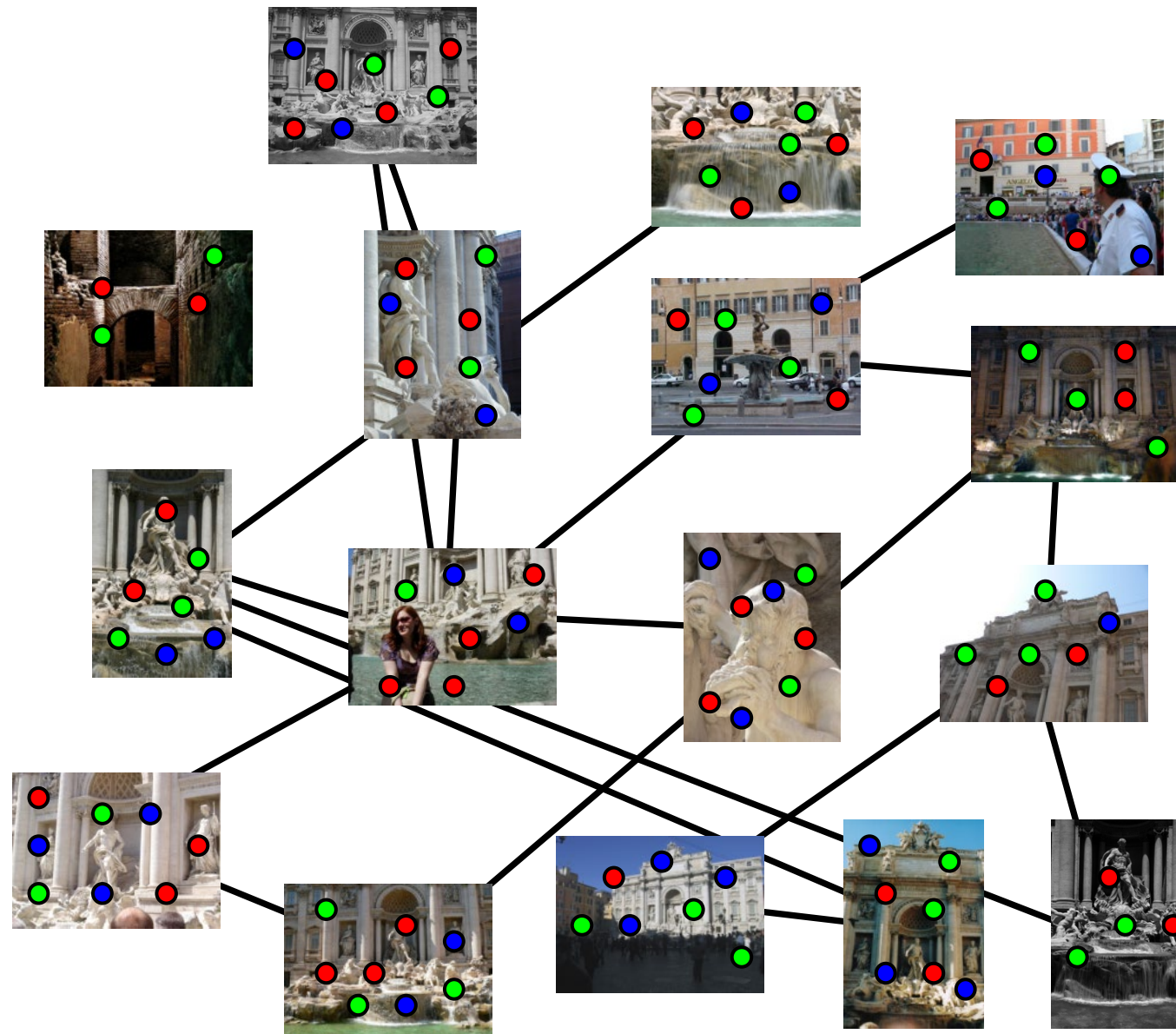
Feature description

Describe features using SIFT [Lowe, IJCV 2004]



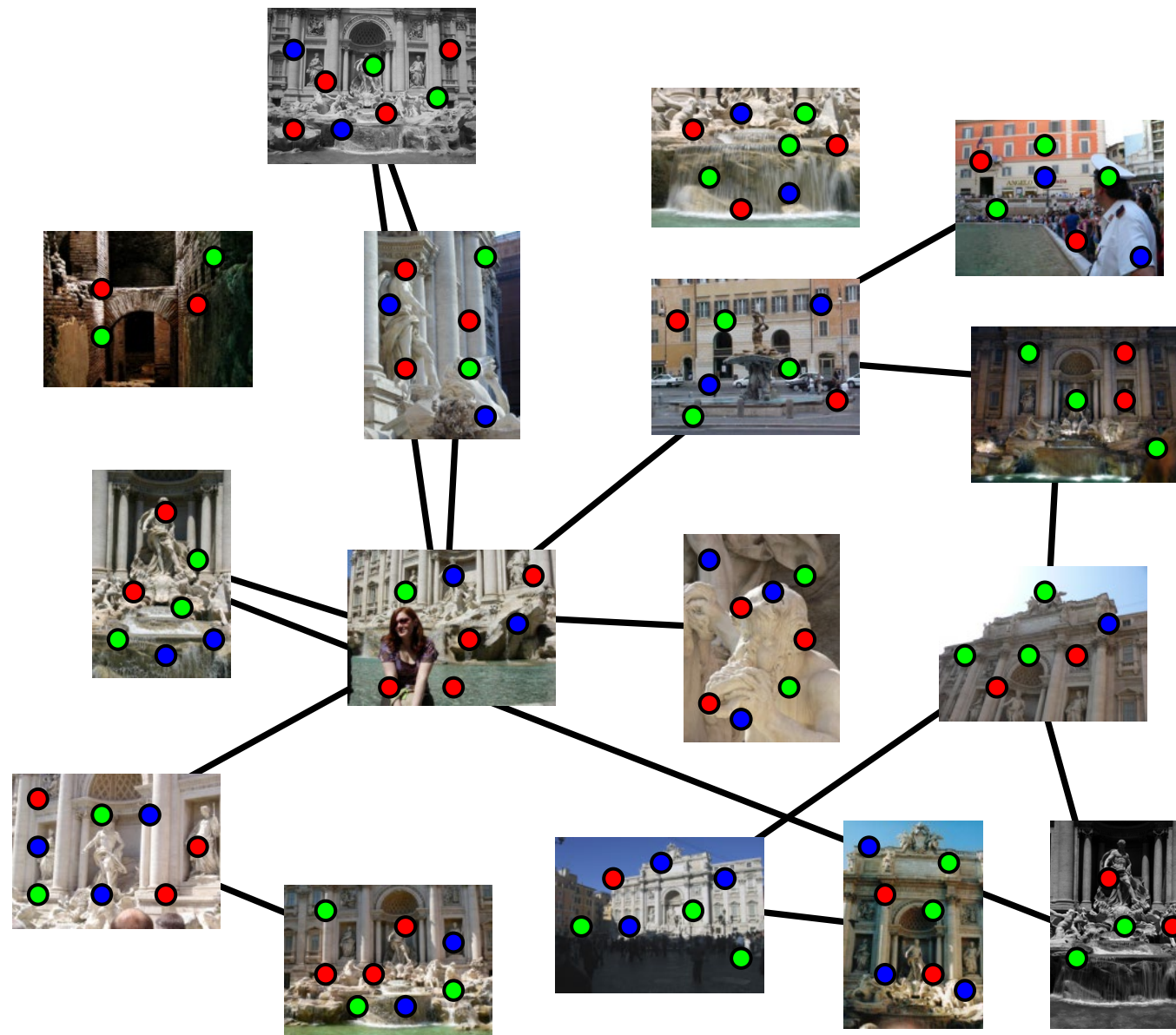
Feature matching

Match features between each pair of images



Feature matching

Refine matching using RANSAC to estimate fundamental matrix between each pair



Correspondence estimation

- Link up pairwise matches to form connected components of matches across several images

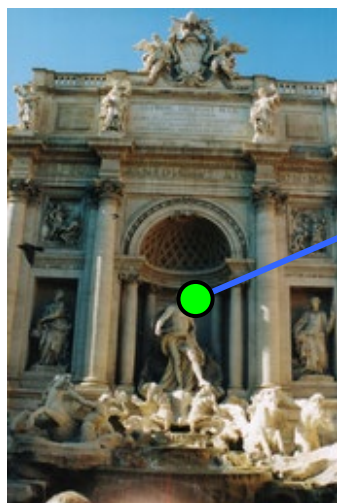


Image 1

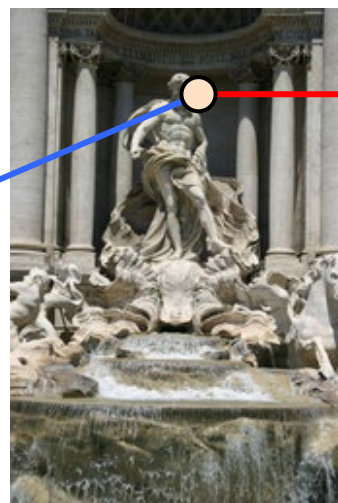


Image 2

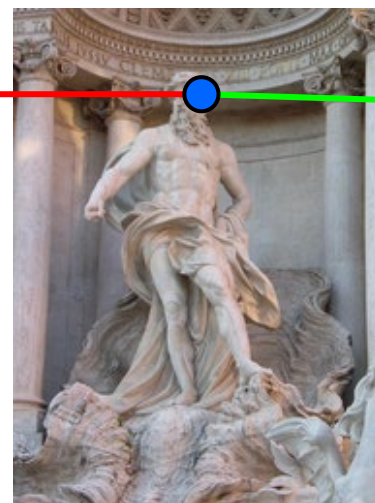


Image 3

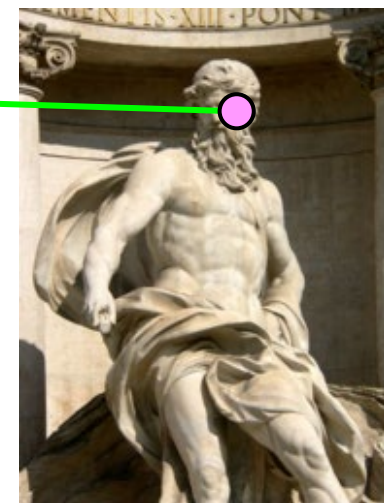


Image 4

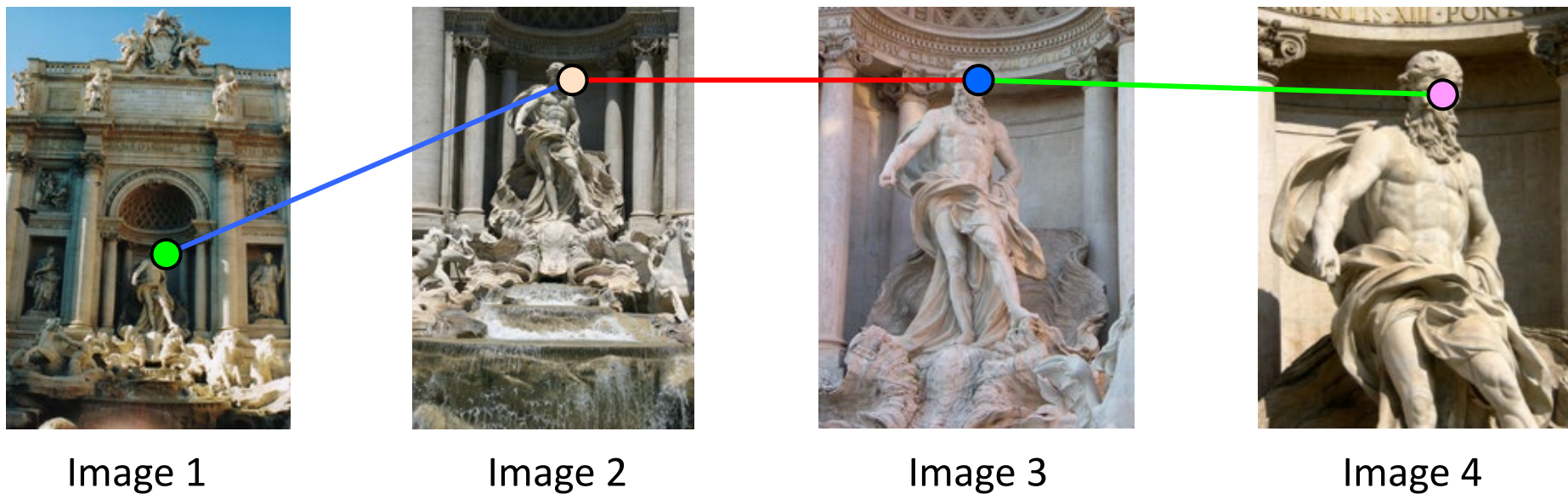
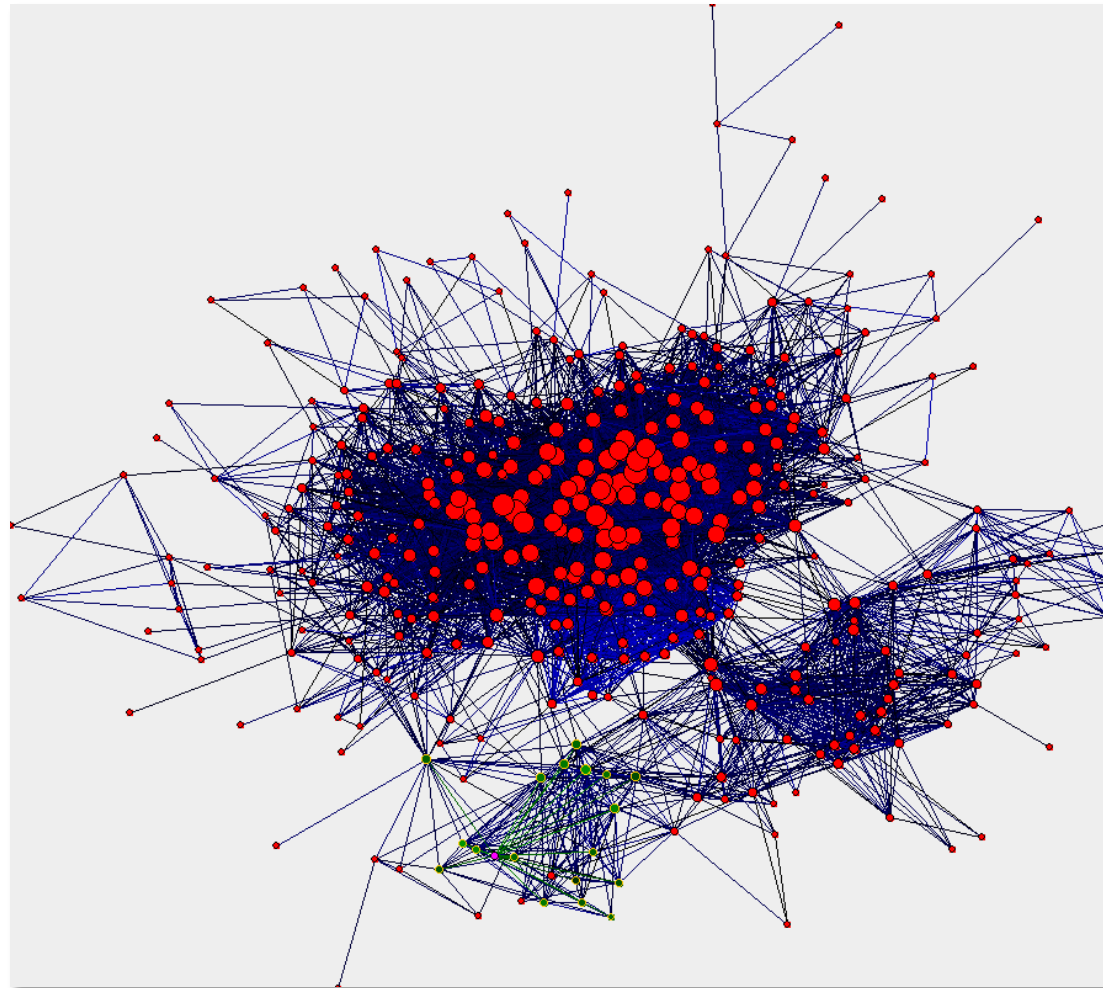
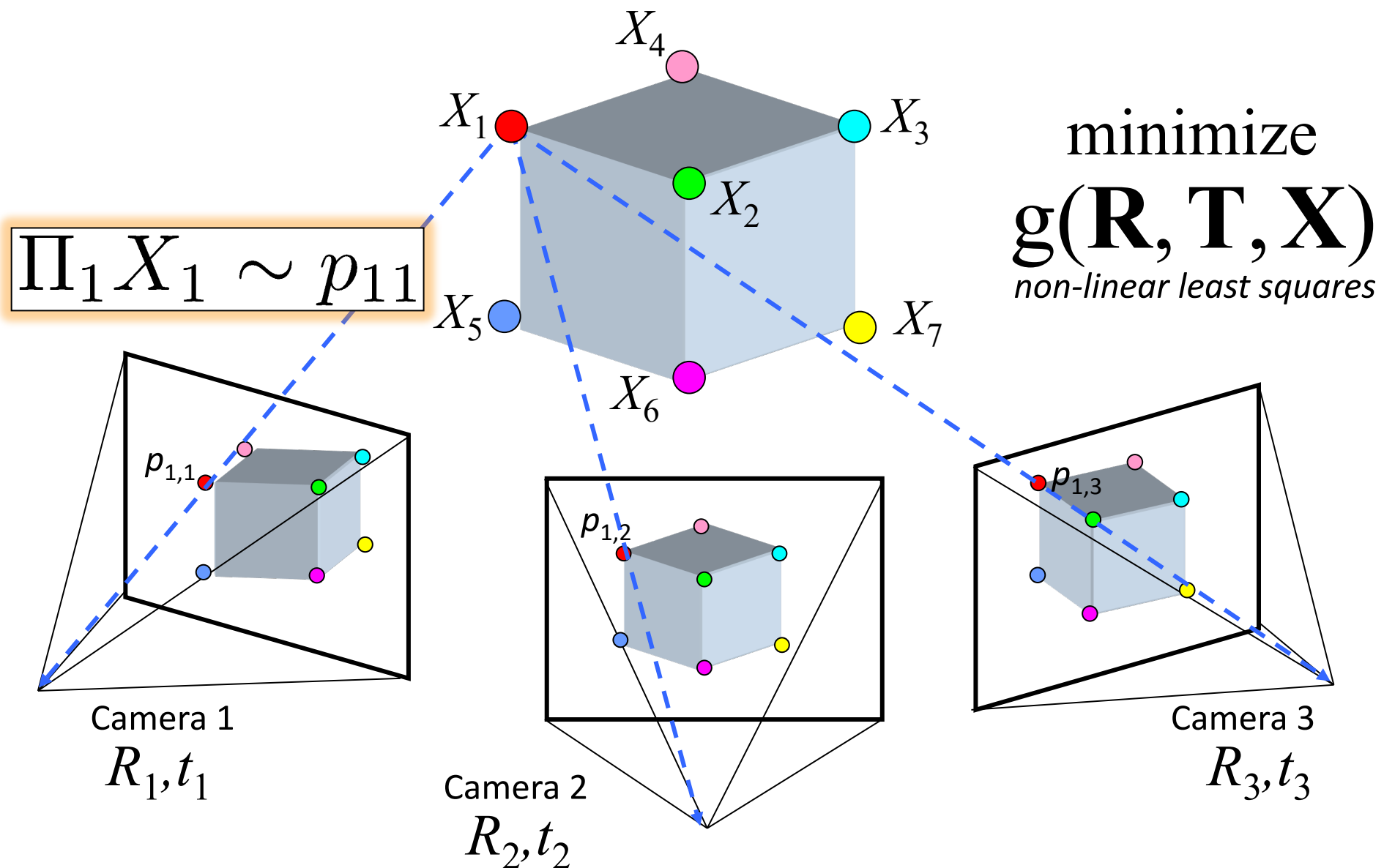


Image connectivity graph



(graph layout produced using the Graphviz toolkit: <http://www.graphviz.org/>)

Structure from motion



Global structure from motion

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^m \sum_{j=1}^n \underbrace{w_{ij}}_{\substack{\text{indicator variable:} \\ \text{is point } i \text{ visible in image } j?}} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\substack{\text{predicted} \\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\text{observed} \\ \text{image location}}} \right\|^2$$

- Minimizing this function is called *bundle adjustment*
 - Optimized using non-linear least squares, e.g. Levenberg-Marquardt

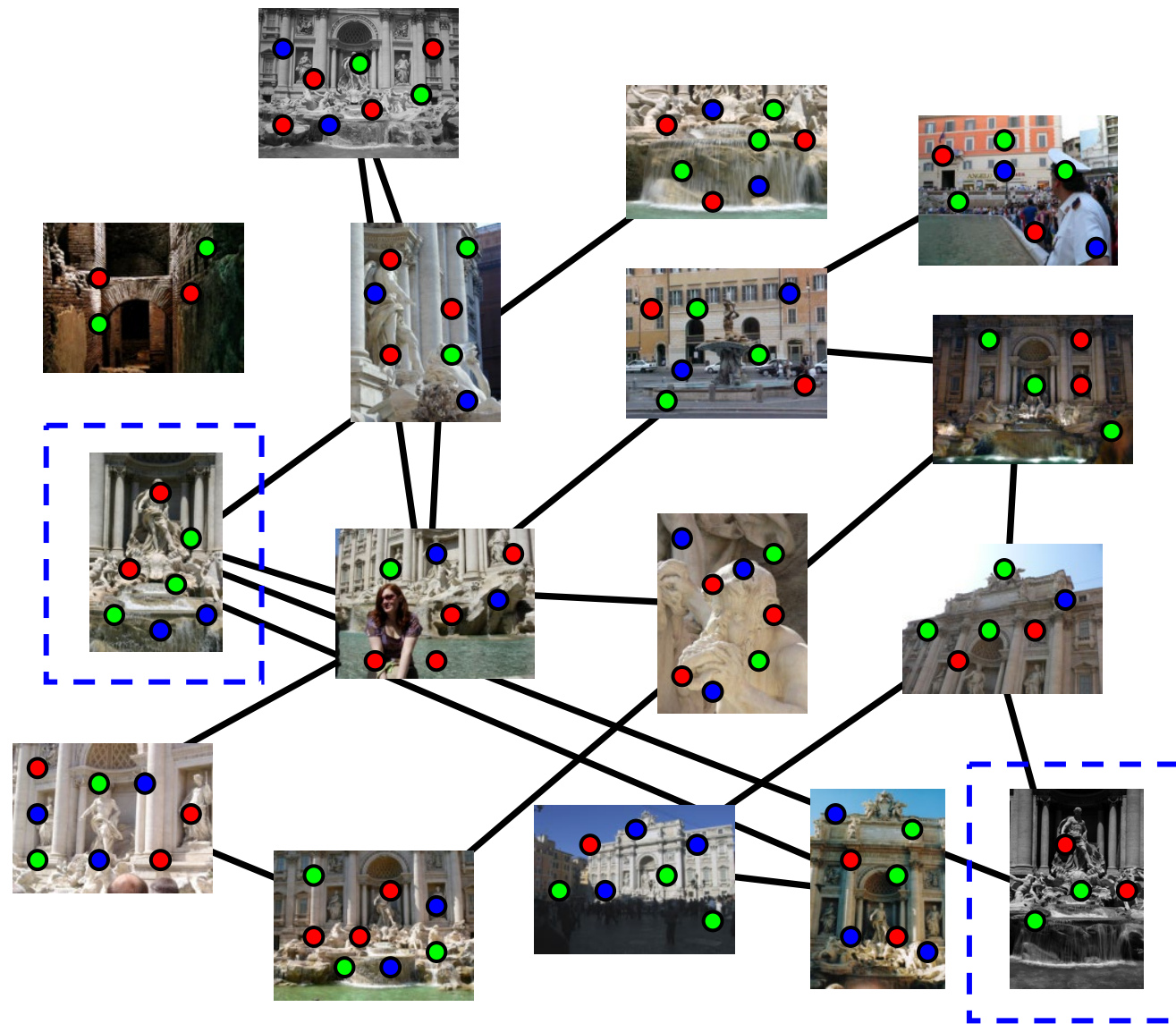
Problem size

- What are the variables?
- How many variables per camera?
- How many variables per point?
- Trevi Fountain collection
 - 466 input photos
 - + > 100,000 3D points
 - = very large optimization problem

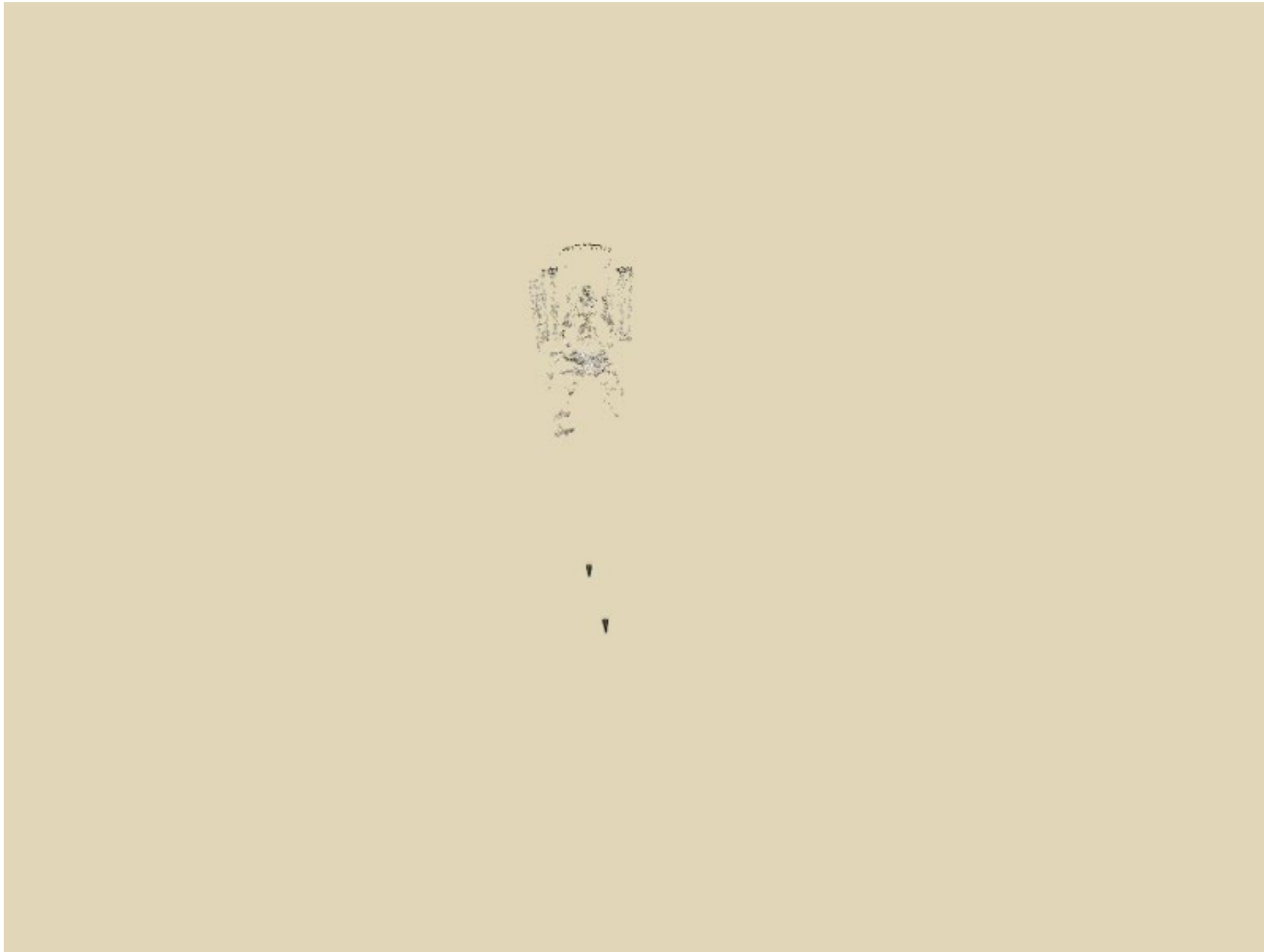
Doing bundle adjustment

- Minimizing g is difficult
 - g is non-linear due to rotations, perspective division
 - lots of parameters: 3 for each 3D point, 6 for each camera
 - difficult to initialize
 - gauge ambiguity: error is invariant to a similarity transform (translation, rotation, uniform scale)
- Many techniques use non-linear least-squares (NLLS) optimization (*bundle adjustment*)
 - Levenberg-Marquardt is one common algorithm for NLLS
 - Lourakis, **The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm**,
<http://www.ics.forth.gr/~lourakis/sba/>
 - http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm

Initialization: Incremental structure from motion



Incremental structure from motion



Final reconstruction



More examples



More examples



More examples









Even larger scale SfM

City-scale structure from motion

- “Building Rome in a day”

<http://grail.cs.washington.edu/projects/rome/>

SfM applications

- 3D modeling
- Surveying
- Robot navigation and mapmaking
- Visual effects (“Match moving”)
 - https://www.youtube.com/watch?v=RdYWp70P_kY

Applications – Photosynth



Applications – Hyperlapse



<https://www.youtube.com/watch?v=SOpwHaQnRSY>

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

References

Basic reading:

- Szeliski textbook, Chapter 7.
- Hartley and Zisserman, Chapter 18.