# Lecture 16: Structure from Motion

COMP 590/776: Computer Vision

Instructor: Soumyadip (Roni) Sengupta

TA: Mykhailo (Misha) Shvets

Course Website: Scan Me!

# Recap

# Geometry: How do we represent shape of an object?

2.5D representation:

1) Depth & Normal map — Easy to predict with 2D neural networks, efficient but do not give full 3D information.

Explicit representation:

2) Mesh — Hard for neural network but most Graphics pipeline use it. Very efficient with memory.

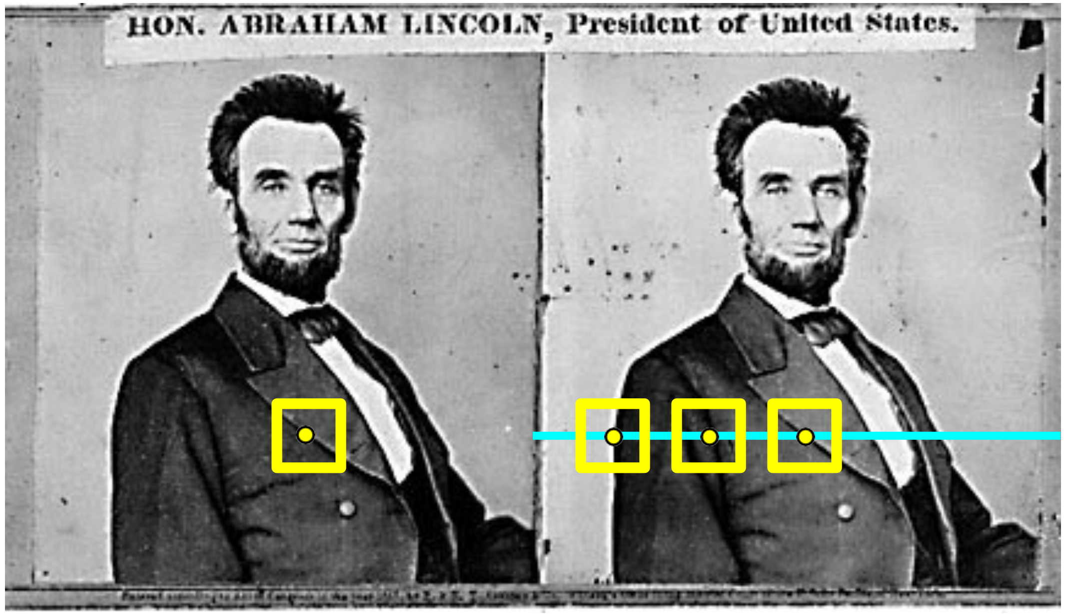
3) Voxels — Easy for neural network but high memory consumption

4) Point Cloud — Output of many RGBD sensors or RGB algorithms

Implicit representation:

5) Surface Representation (SDF) — Memory efficient and deep networks can predict it. But need to convert it to mesh/voxel to be usable in Graphics engines.

# Stereo

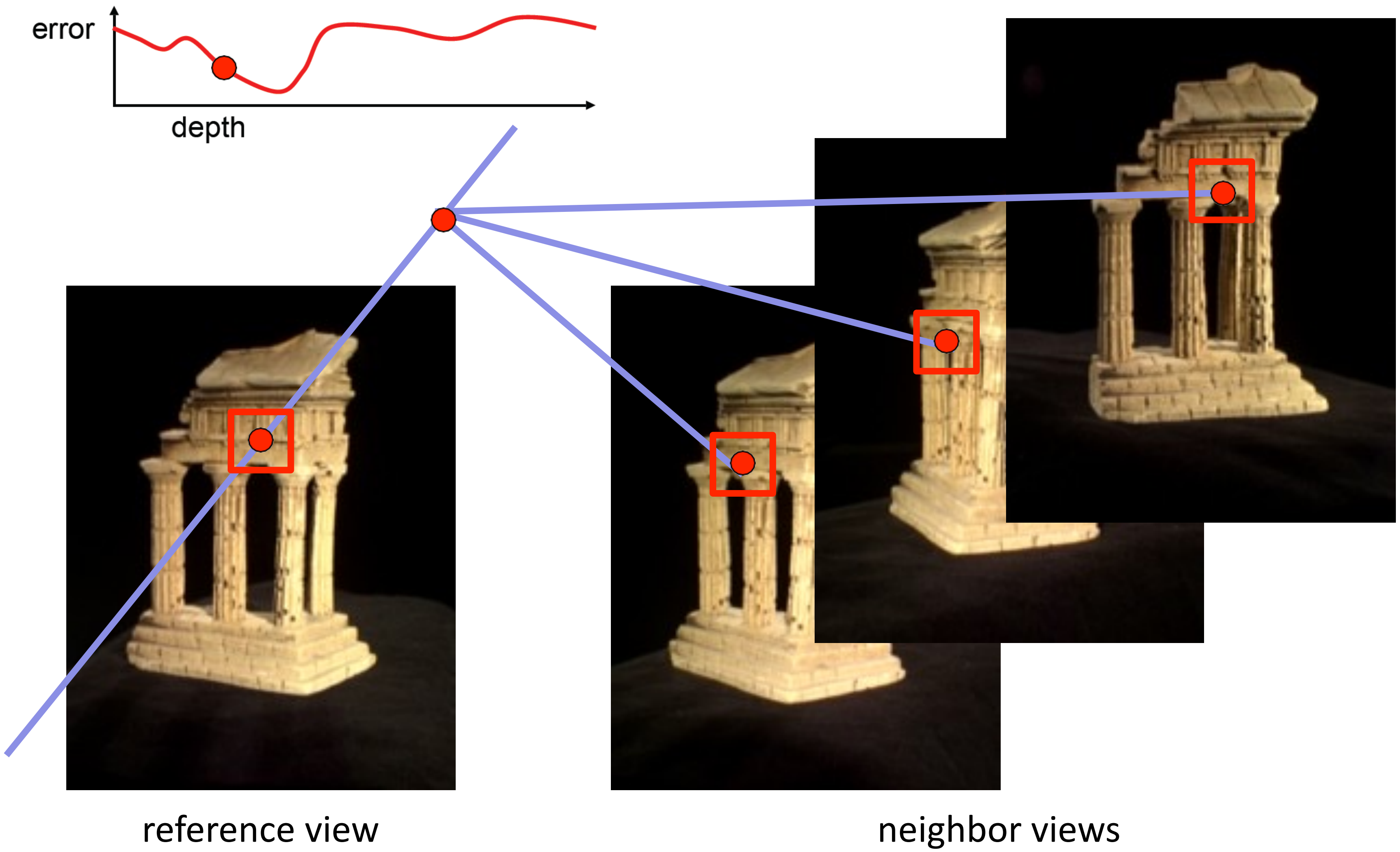



1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

$$Z = \frac{bf}{d}$$

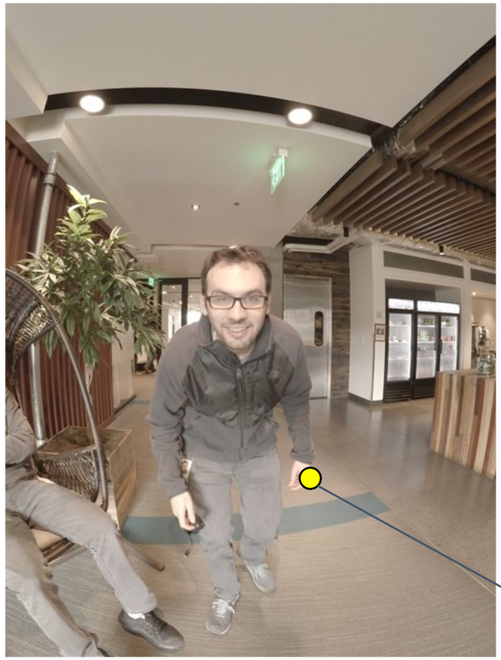
*How can you make the epipolar lines horizontal?*

# Multi-view stereo: Basic idea




Source: Y. Furukawa

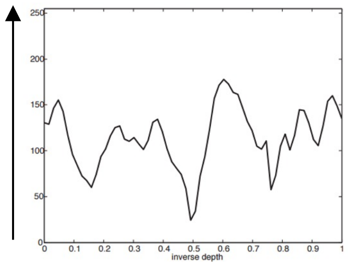
# Plane Sweep Stereo: Cost Volumes -> Depth Maps



Reference image

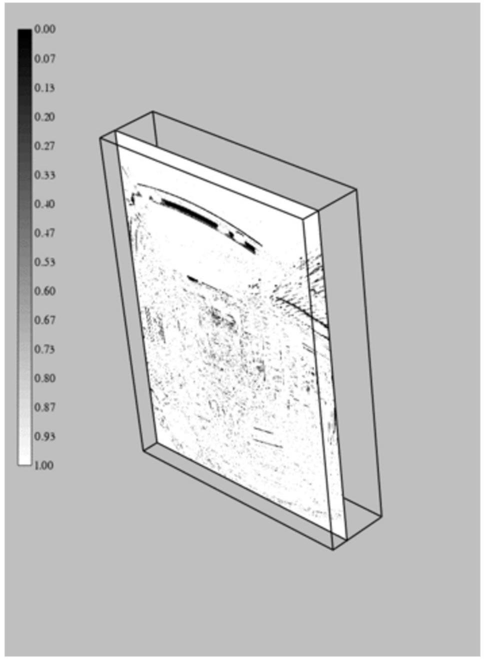
Plane sweep

cost



depth

Single pixel's cost profile



Full cost volume

Depth map solver

(Belief propagation, graph cuts, etc.)

# Plane-Sweep Stereo

- The family of depth planes in the coordinate frame of the reference view

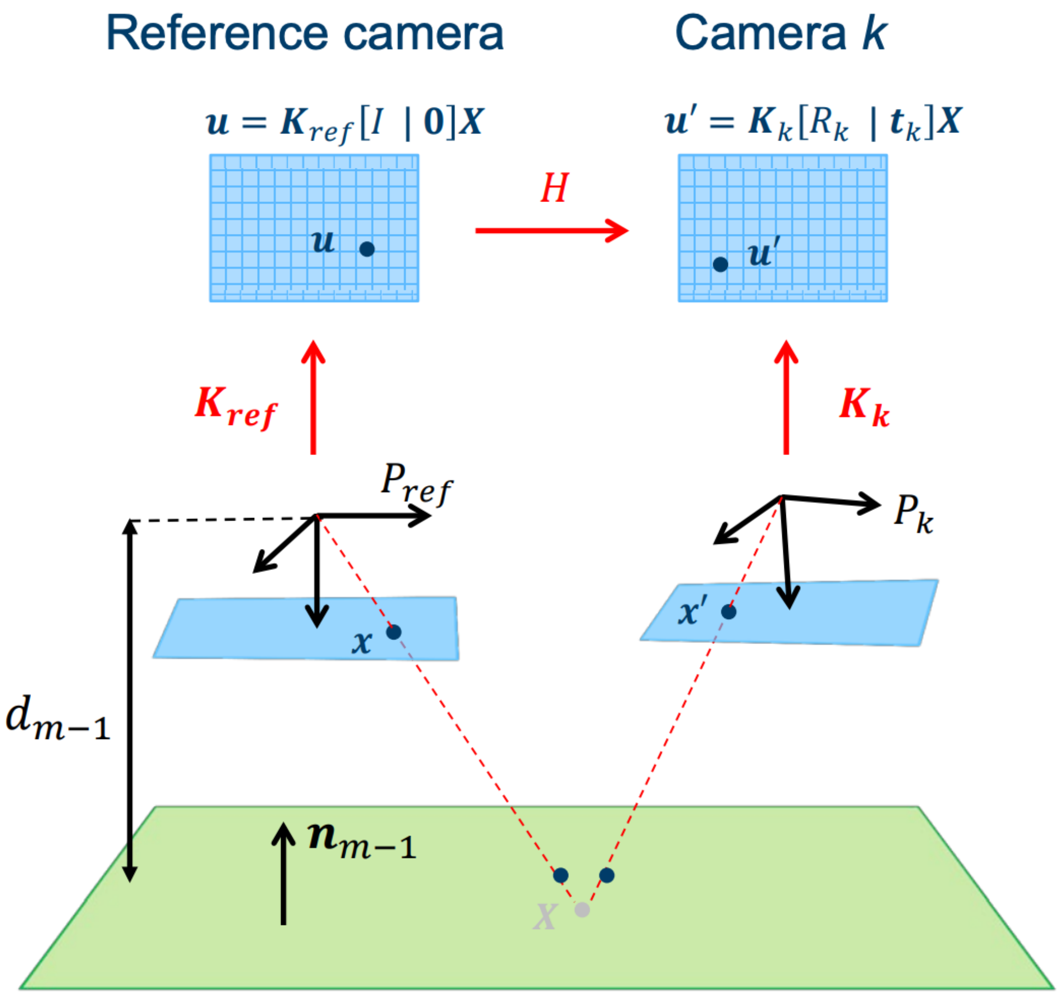
$$\Pi_m = \begin{bmatrix} \boldsymbol{n}_m^T & -d_m \end{bmatrix}$$

- The mapping from the reference camera $P_{ref}$ onto the plane $\Pi_m$ and back to camera $P_k$ is described by the homography induced by the plane $\Pi_m$

$$H_{\Pi_m,P_k} = K_k \left( R_k - \boldsymbol{t}_k \boldsymbol{n}_m^T / d_m \right) K_{ref}^{-1}$$

Try the proof in HW!

- The mapping from $P_k$ to $P_{ref}$ induced by $\Pi_m$ is the inverse homography $H_{\Pi_m,P_k}^{-1}$

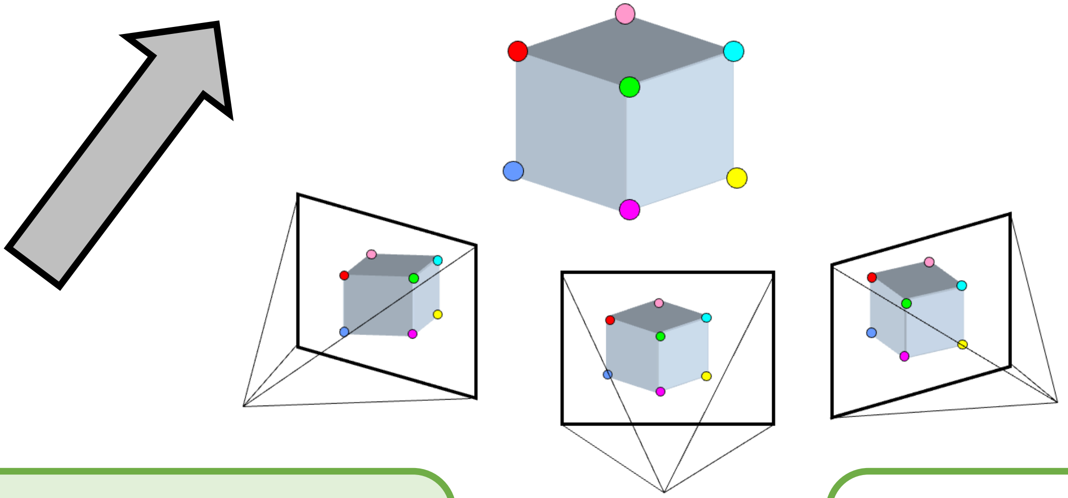



Slight abuse of notation. In equation (x,y) are image co-ordinates, in figure u is image co-ordinate.


Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

# Big picture: 3 key components in 3D

3D Points
(Structure)



Structure from Motion (SfM)

Simultaneous Localization & Mapping (SLAM)

Correspondences

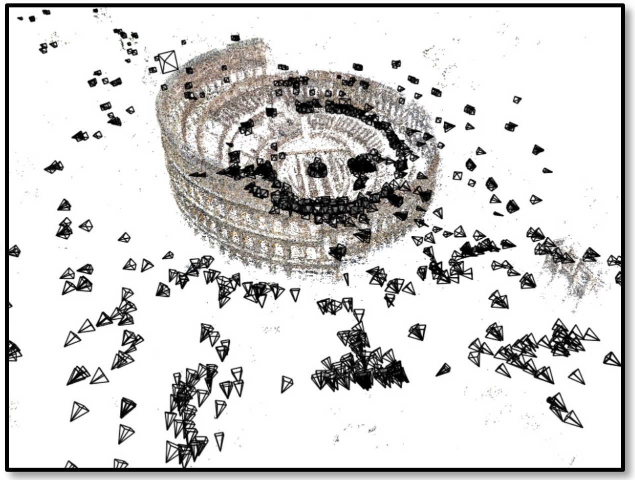
Camera
(Motion)

# Structure from motion

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

# Structure from Motion (SfM)

- Given many images, how can we
  a) figure out where they were all taken from?
  b) build a 3D model of the scene?



This is (roughly) the **structure from motion** problem

# Photo Tourism

Noah Snavely, Steven M. Seitz, Richard Szeliski, "[Photo tourism: Exploring photo collections in 3D]," SIGGRAPH 2006

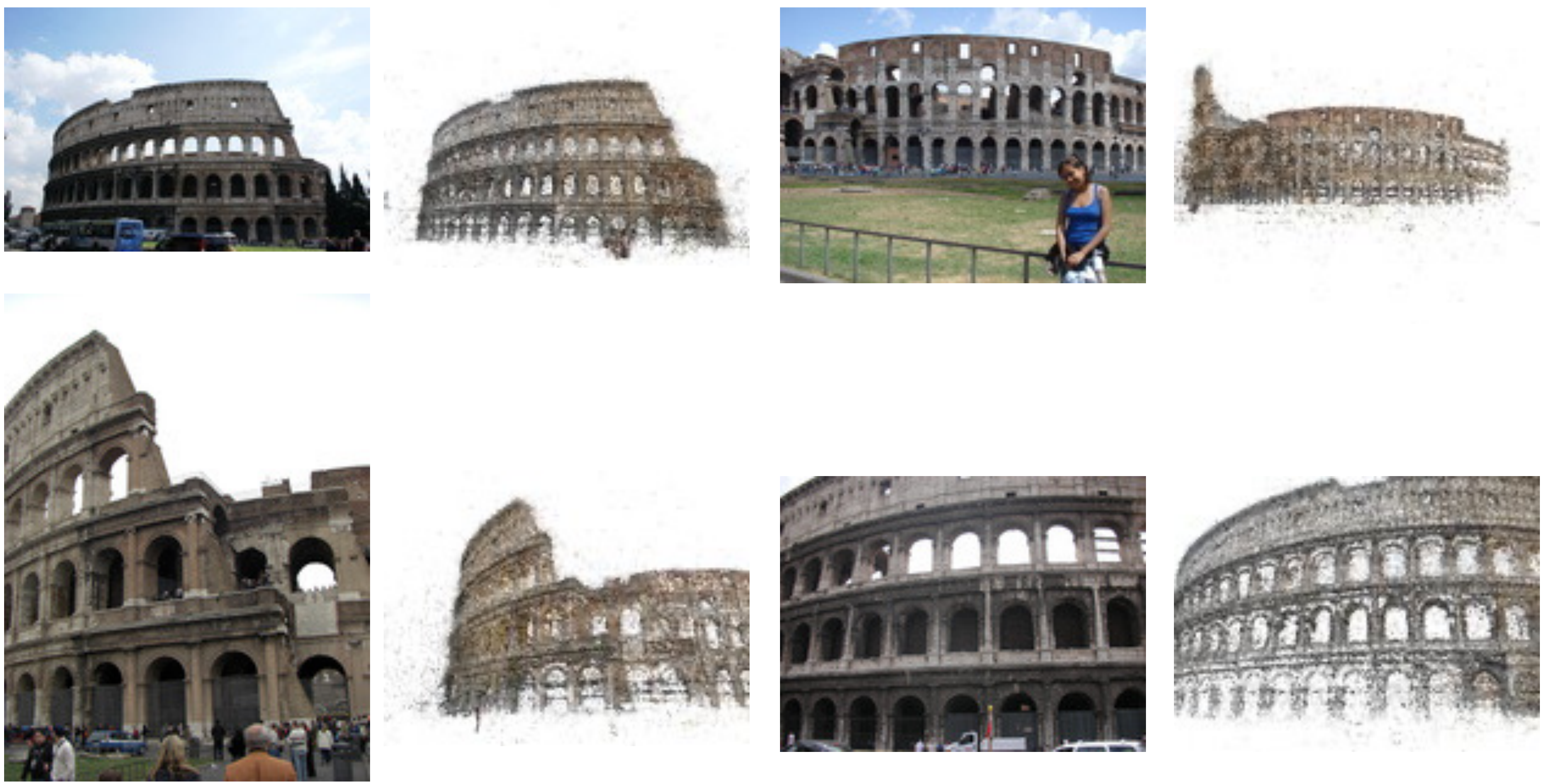
https://youtu.be/mTBPGuPLI5Y

# Large-scale structure from motion

Dubrovnik, Croatia. 4,619 images (out of an initial 57,845).
Total reconstruction time: 23 hours
Number of cores: 352

Building Rome in a Day, Agarwal et al. ICCV 2009

# Large-scale structure from motion



Rome's Colosseum

Building Rome in a Day, Agarwal et al. ICCV 2009

**Reconstructing the World in Six Days,**
Jared Heinly, Johannes L. Schönberger, Enrique Dunn, Jan-Michael Frahm, CVPR 2015.
Work done at UNC CS!

St. Peter’s Basilica, Vatican City
Yahoo Flickr Creative Commons 100M Dataset

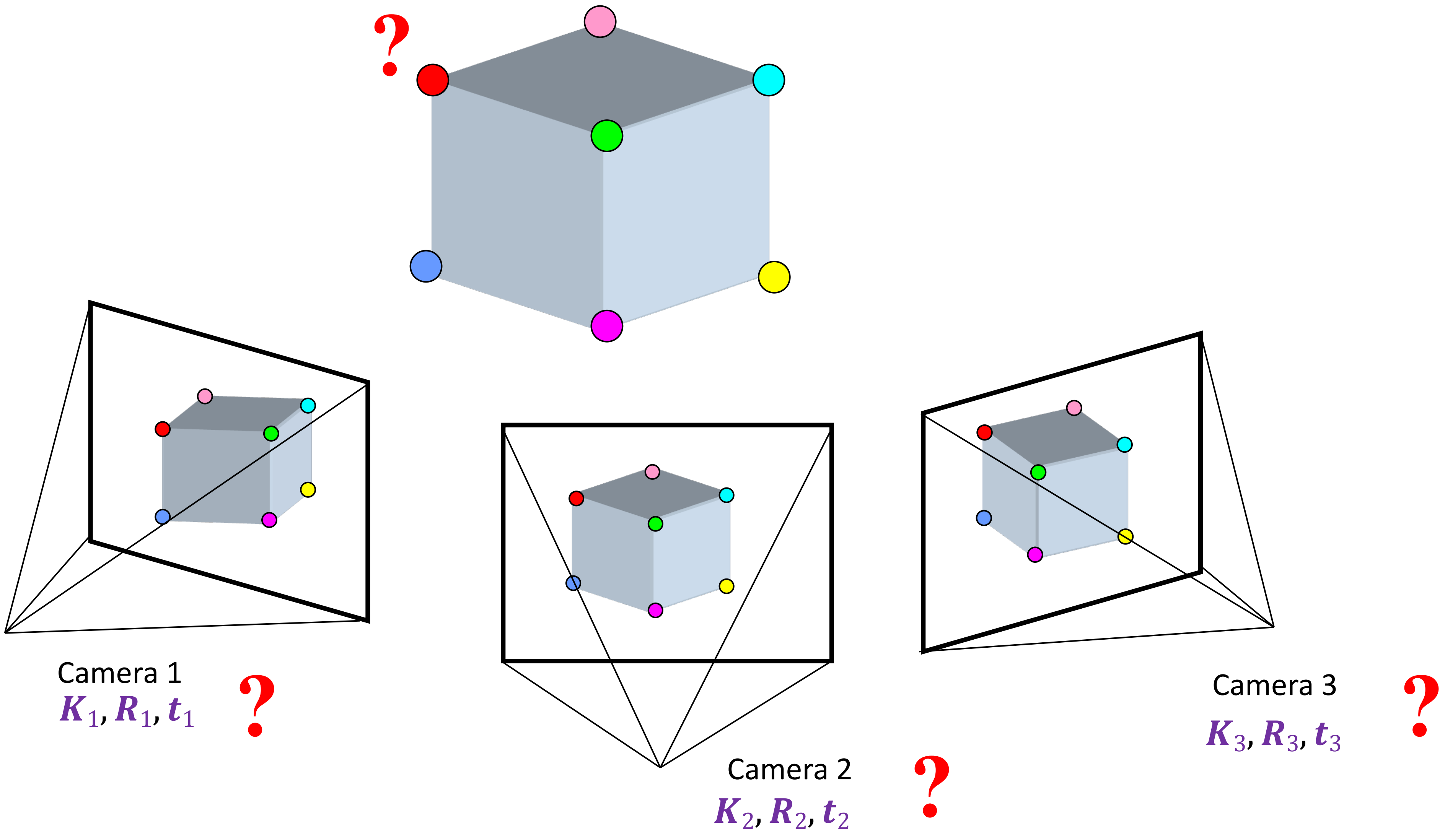
# Today's Class

- Ambiguities in SfM
- Affine SfM
- Projective SfM
  - Global SfM
  - Incremental SfM
- Challenges and Applications

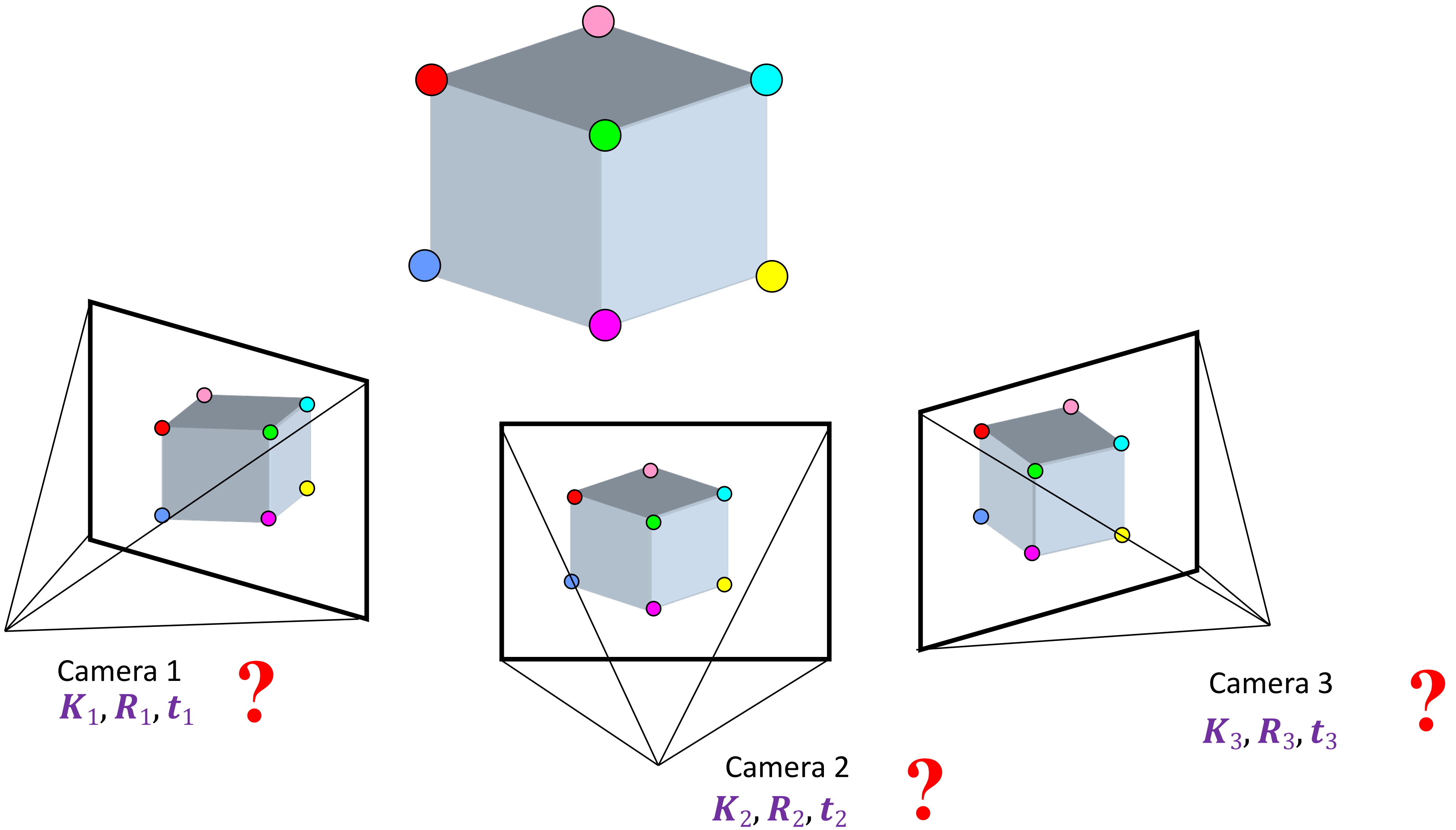
# Today’s Class

- Ambiguities in SfM
- Affine SfM
- Projective SfM
  - Global SfM
  - Incremental SfM
- Challenges and Applications

# Structure from motion

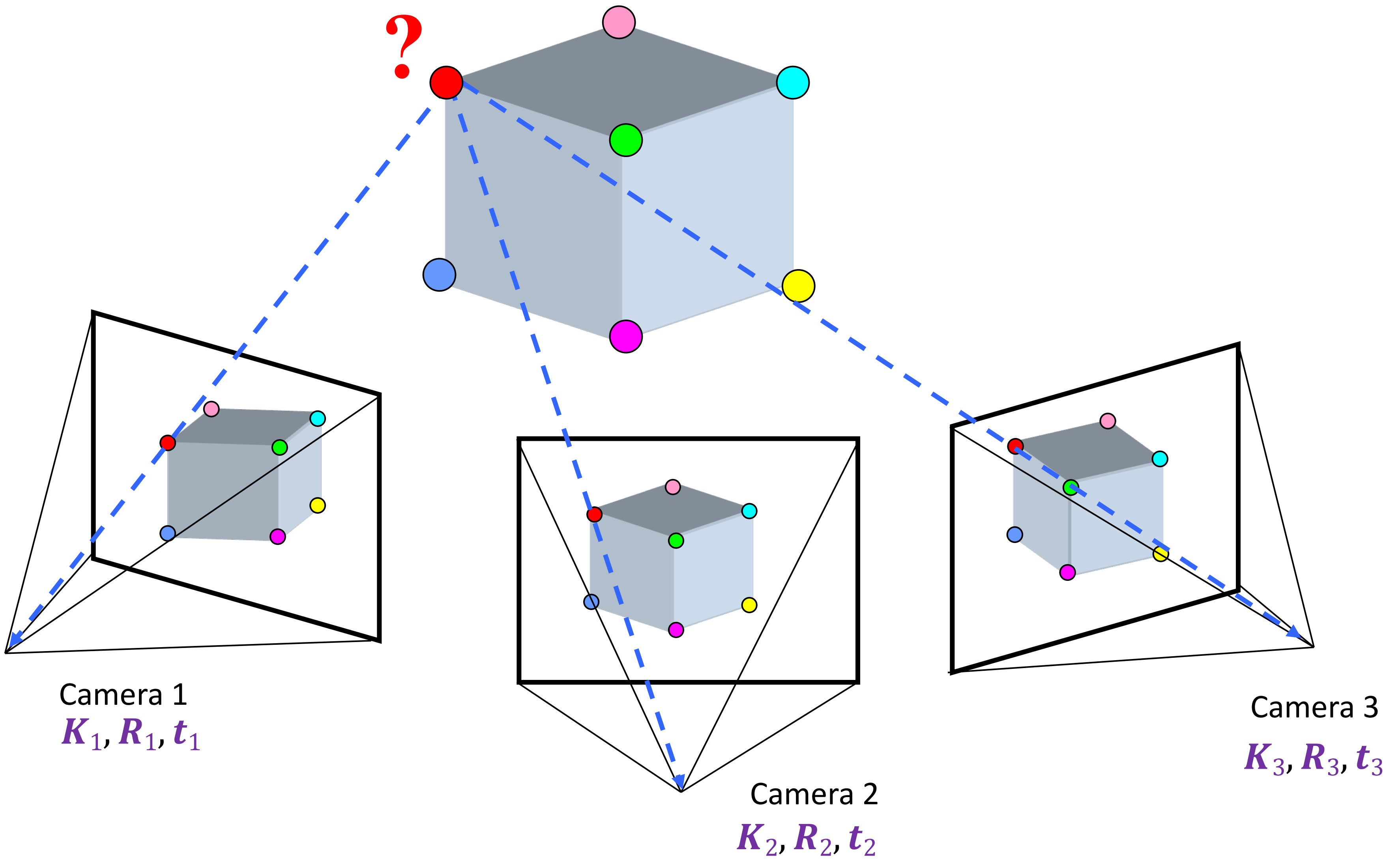


# Recall: Calibration




- Given a set of *known* 3D points seen by a camera, compute the camera parameters

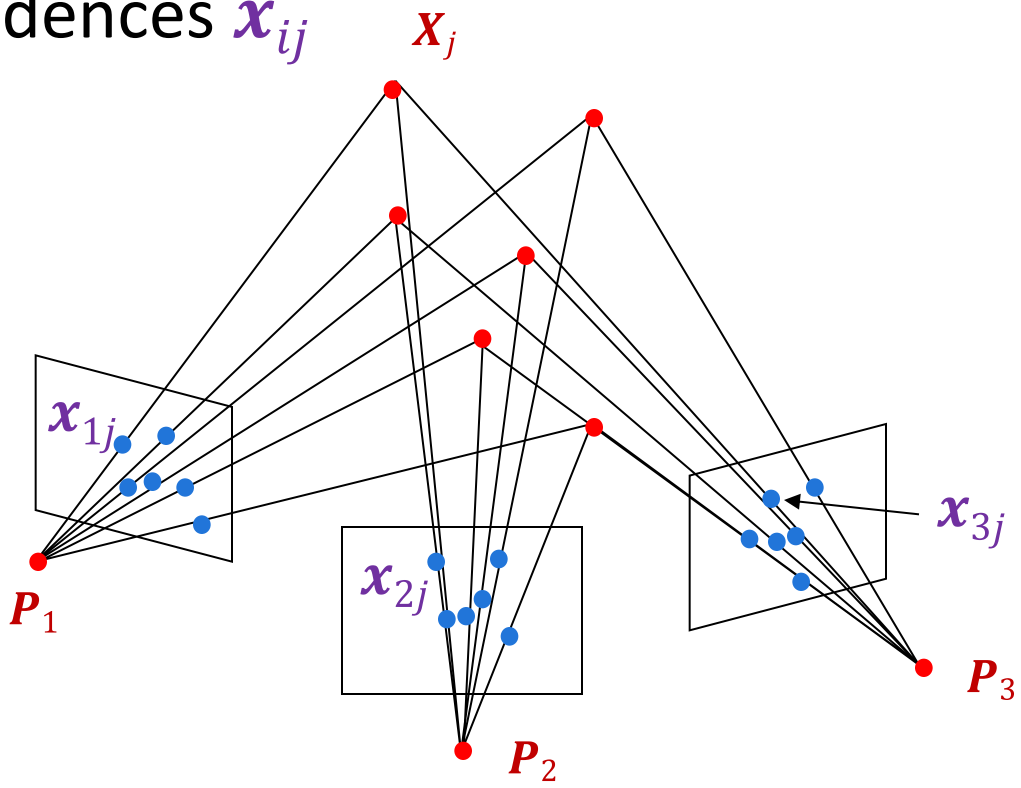
# Recall: Triangulation



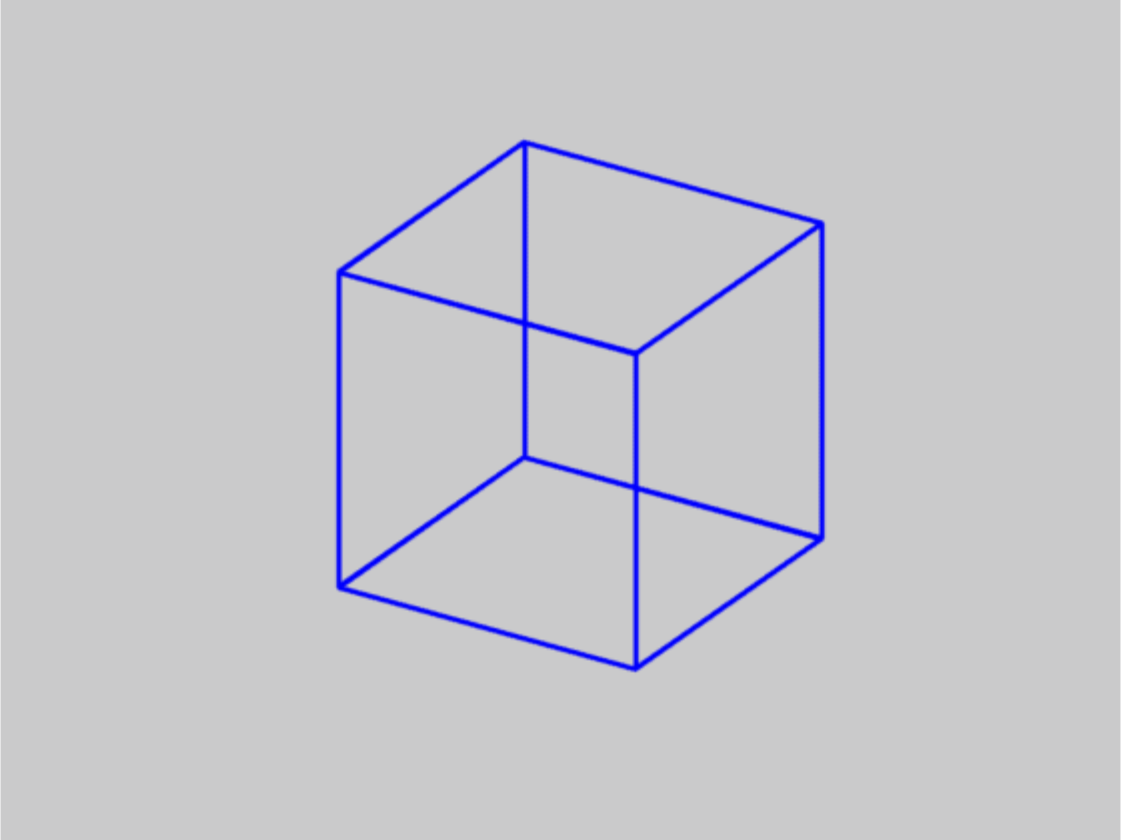

- Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point

# Structure from motion: Problem formulation

- Given: $m$ images of $n$ fixed 3D points such that (ignoring visibility)
  - $\boldsymbol{x}_{ij} \cong \boldsymbol{P}_i \boldsymbol{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- Problem: estimate $m$ projection matrices $\boldsymbol{P}_i$ and $n$ 3D points $\boldsymbol{X}_j$ from the $mn$ correspondences $\boldsymbol{x}_{ij}$



# Is SFM always uniquely solvable?



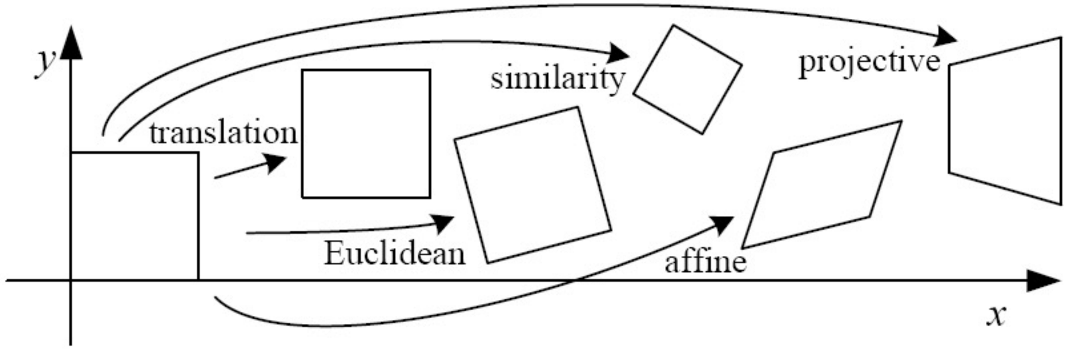
- Necker cube

Source: N. Snavely

# 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 remain exactly the same:
- $\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = \left(\frac{1}{k}\boldsymbol{P}\right)(k\boldsymbol{X})$
- Without a reference measurement, it is impossible to recover the absolute scale of the scene!
- In general, if we transform the scene using a transformation $\boldsymbol{Q}$ and apply the inverse transformation to the camera matrices, then the image observations do not change:
- $\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$

# Recall: 2D image transformations




| Name | Matrix | # D.O.F. | Preserves: | Icon |
|---|---|---|---|---|
| translation | $\left[\ \boldsymbol{I} \mid \boldsymbol{t}\ \right]_{2\times 3}$ | 2 | orientation $+\cdots$ | |
| rigid (Euclidean) | $\left[\ \boldsymbol{R} \mid \boldsymbol{t}\ \right]_{2\times 3}$ | 3 | lengths $+\cdots$ | |
| similarity | $\left[\ s\boldsymbol{R} \mid \boldsymbol{t}\ \right]_{2\times 3}$ | 4 | angles $+\cdots$ | |
| affine | $\left[\ \boldsymbol{A}\ \right]_{2\times 3}$ | 6 | parallelism $+\cdots$ | |
| projective | $\left[\ \tilde{\boldsymbol{H}}\ \right]_{3\times 3}$ | 8 | straight lines | |

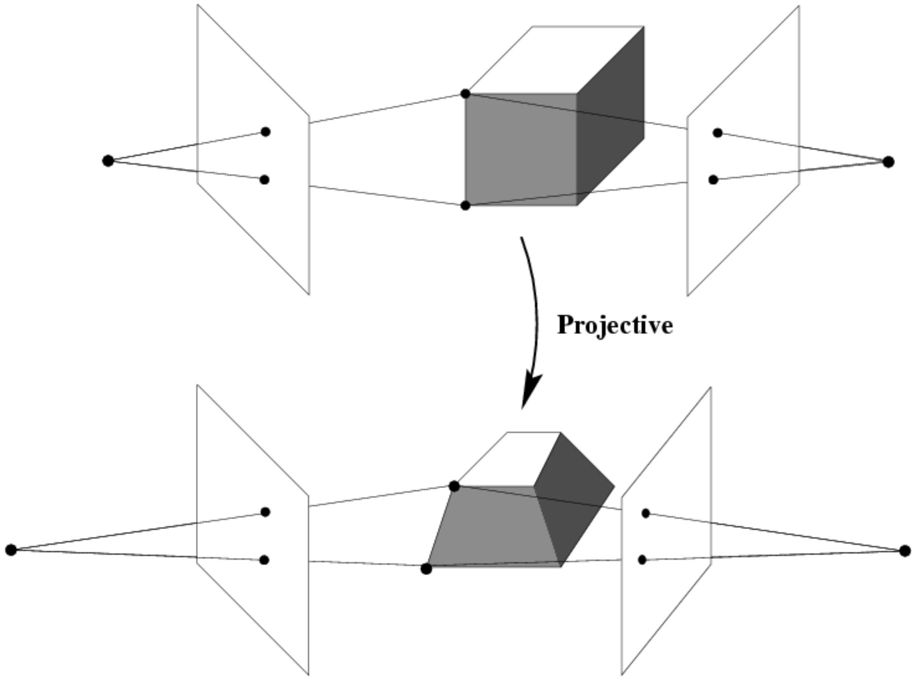
Now, lets extend this to 3D.

# Projective ambiguity

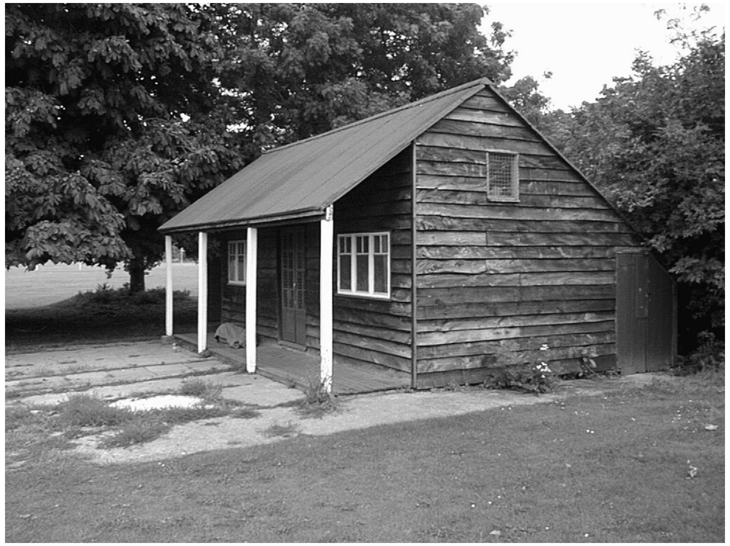
- With no constraints on the camera calibration matrices or on the scene, we can reconstruct up to a *projective* ambiguity:

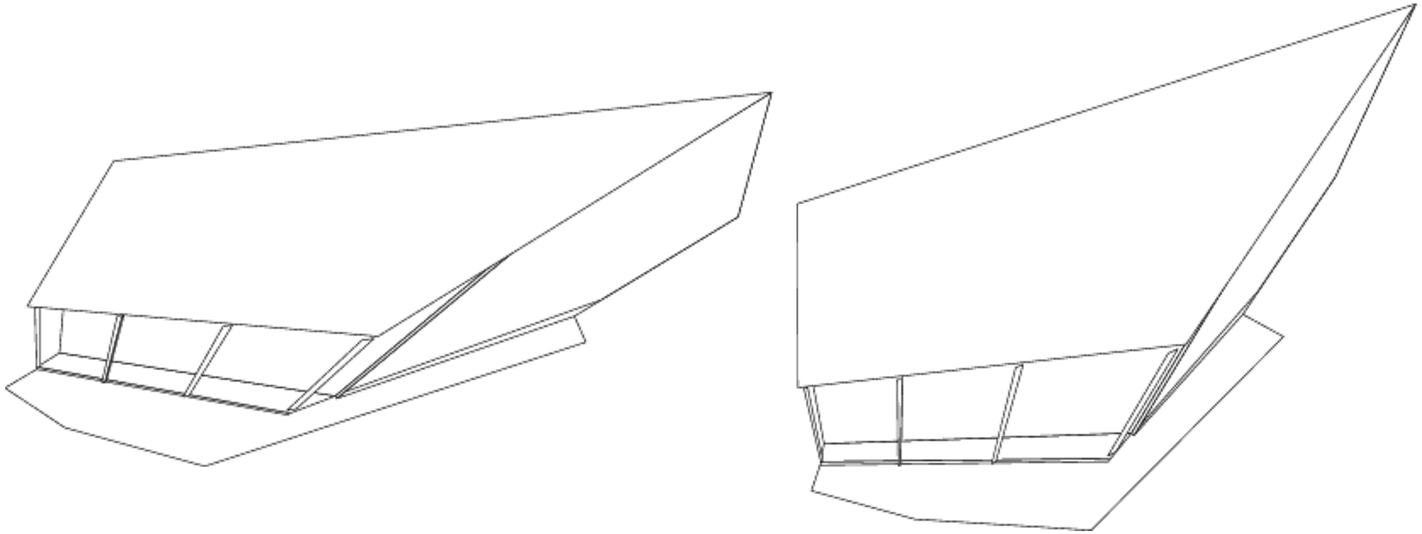
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$$

$\boldsymbol{Q}$ is a general full-rank 4×4 matrix



# Projective ambiguity





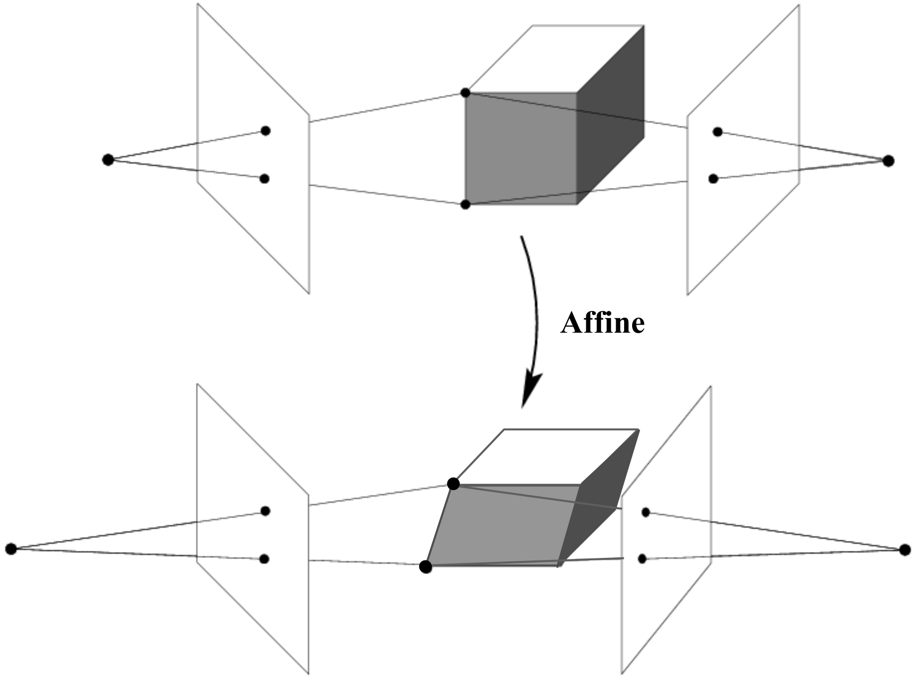
# Affine ambiguity

- If we impose parallelism constraints, we can get a reconstruction up to an *affine* ambiguity:

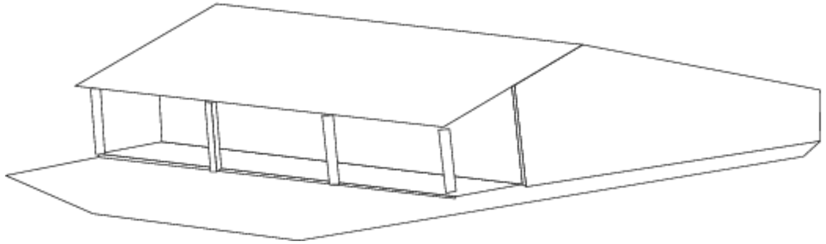
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = \left(\boldsymbol{P}\boldsymbol{Q}_A^{-1}\right)\left(\boldsymbol{Q}_A\boldsymbol{X}\right)$$

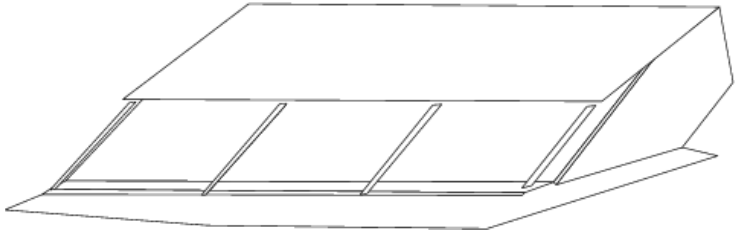
$$\boldsymbol{Q}_A = \begin{array}{cc} \text{3×3 full-rank matrix} & \text{3×1 translation vector} \\ \end{array}$$

$$\boldsymbol{Q}_A = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



# Affine ambiguity





# Similarity ambiguity

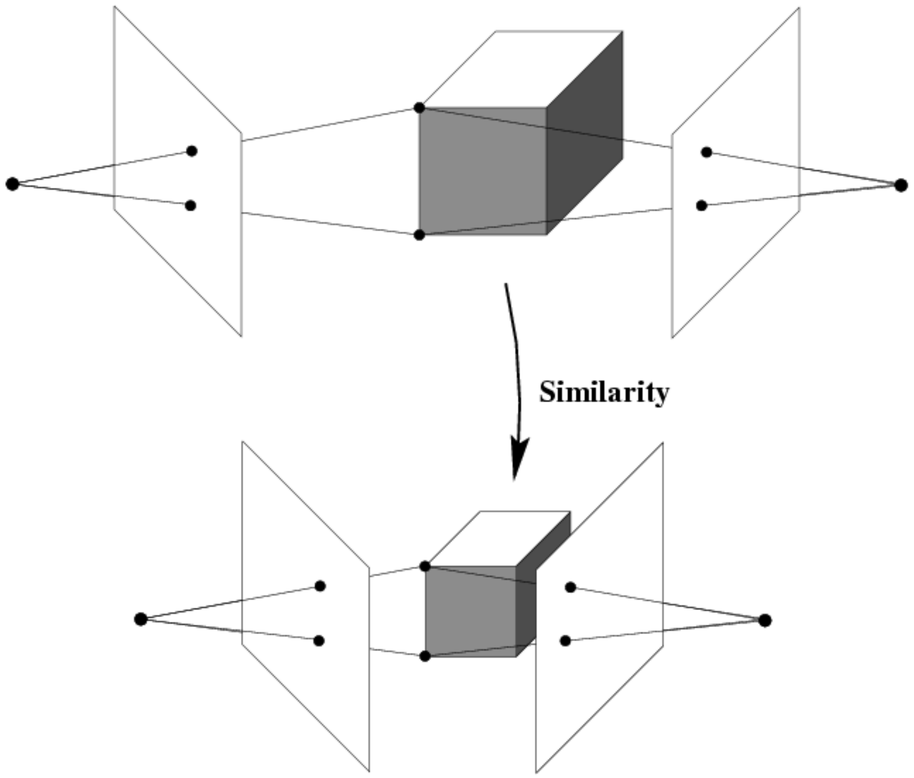
- A reconstruction that obeys orthogonality constraints on camera parameters and/or scene

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

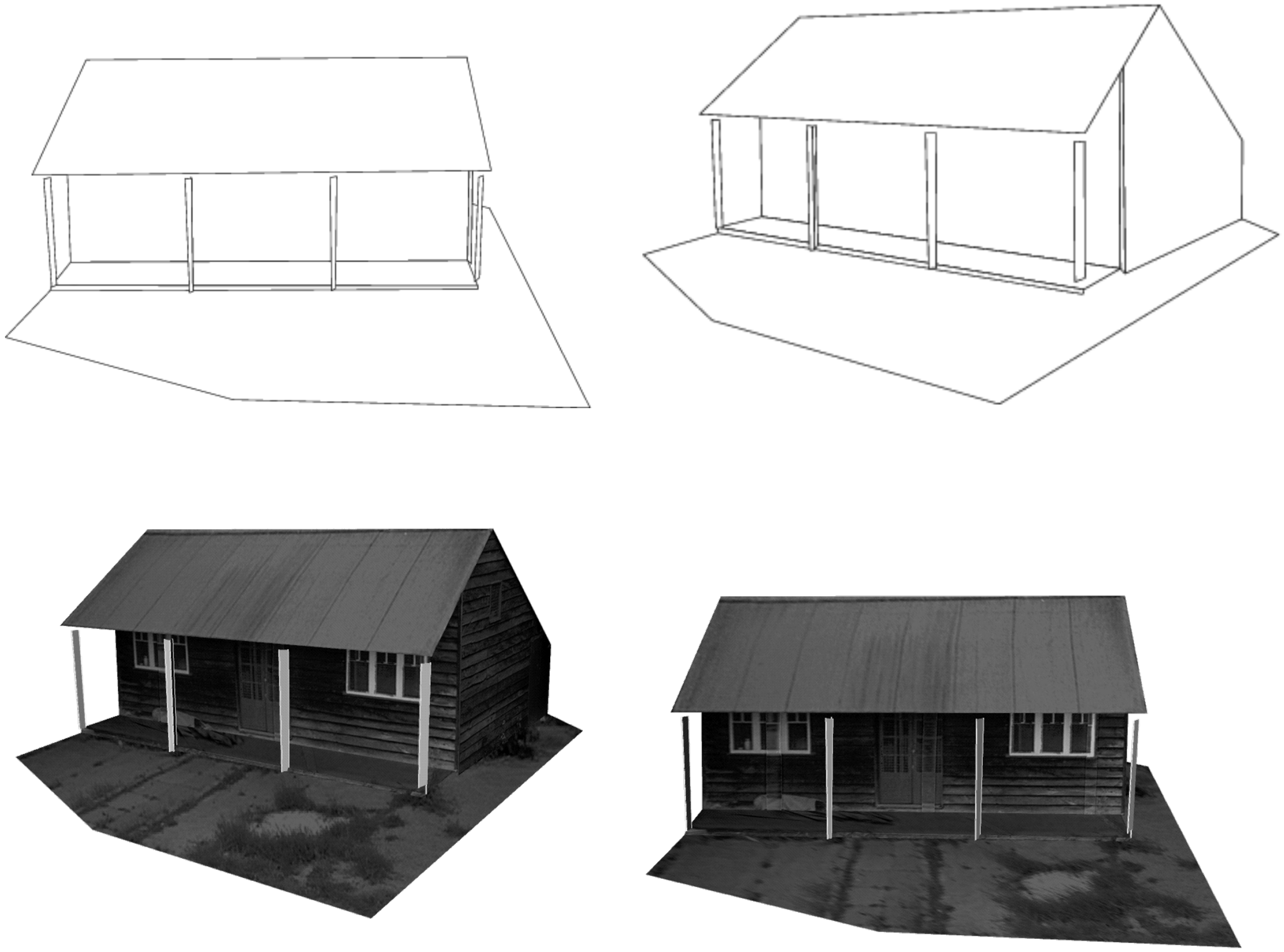
3×3 rotation matrix

3×1 translation vector

$$\boldsymbol{Q}_S = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



# Similarity ambiguity

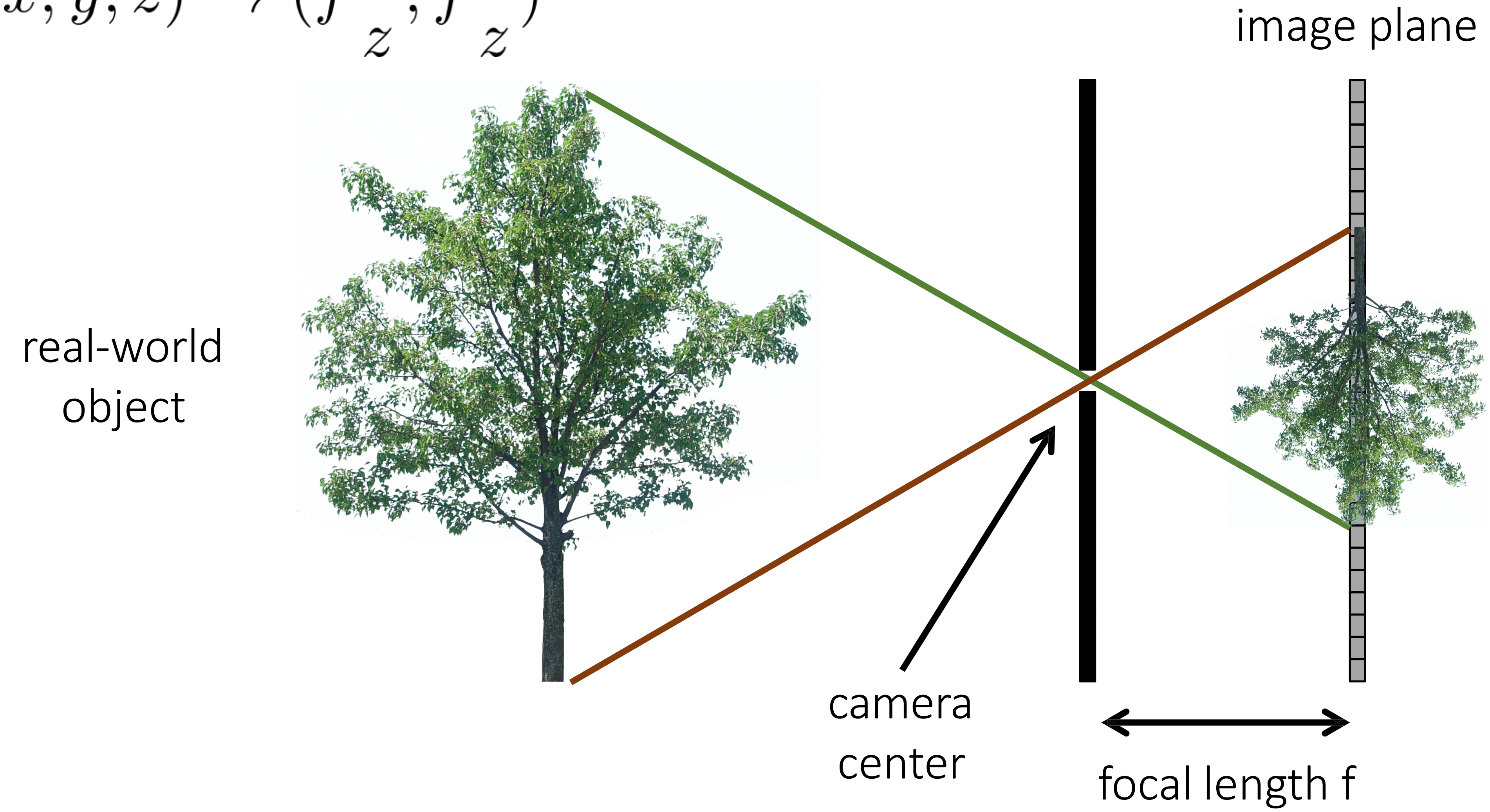


# Today's Class

- Ambiguities in SfM
- Affine SfM
- Projective SfM
  - Global SfM
  - Incremental SfM
- Challenges and Applications

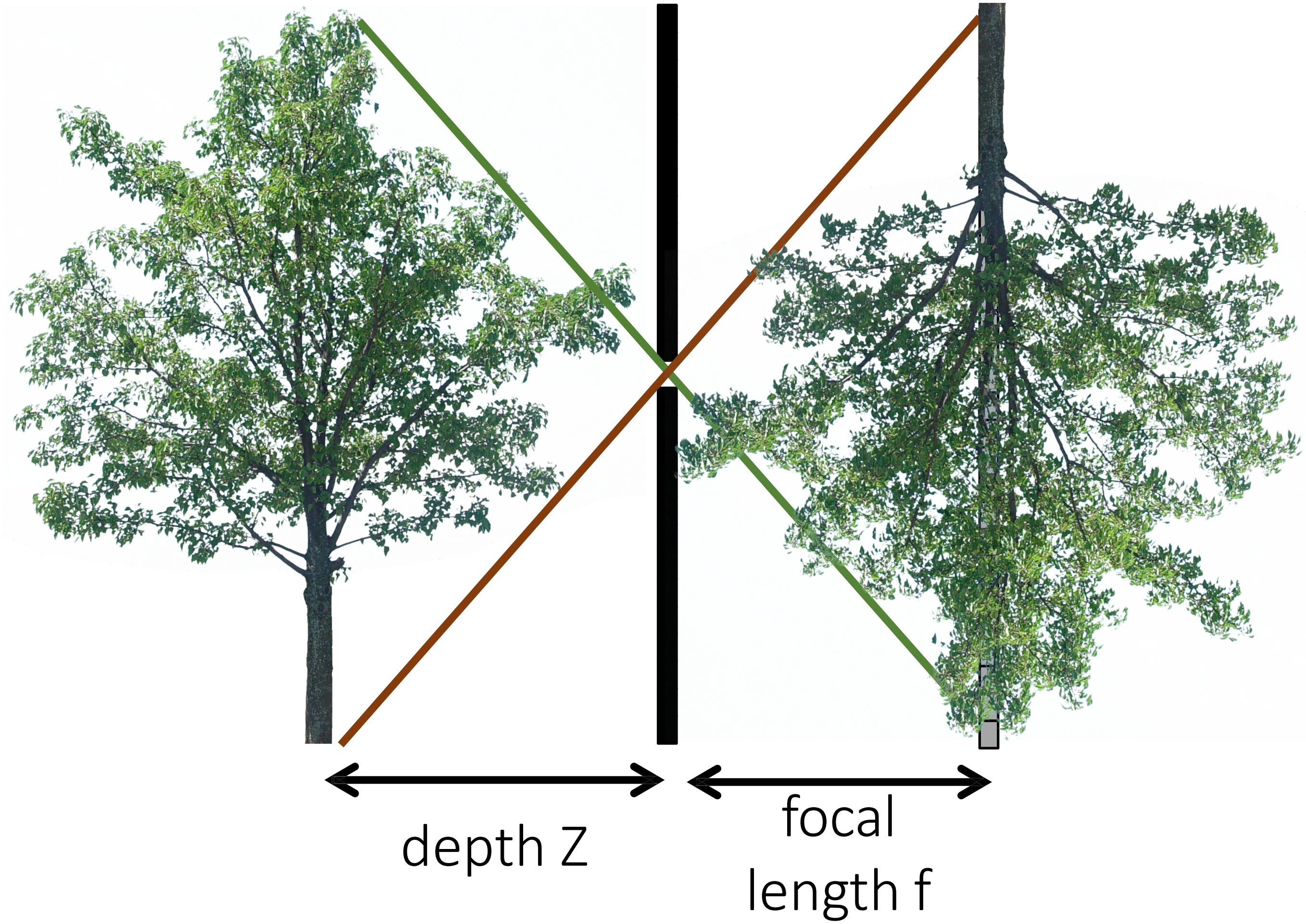
# The pinhole camera

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



# What if…

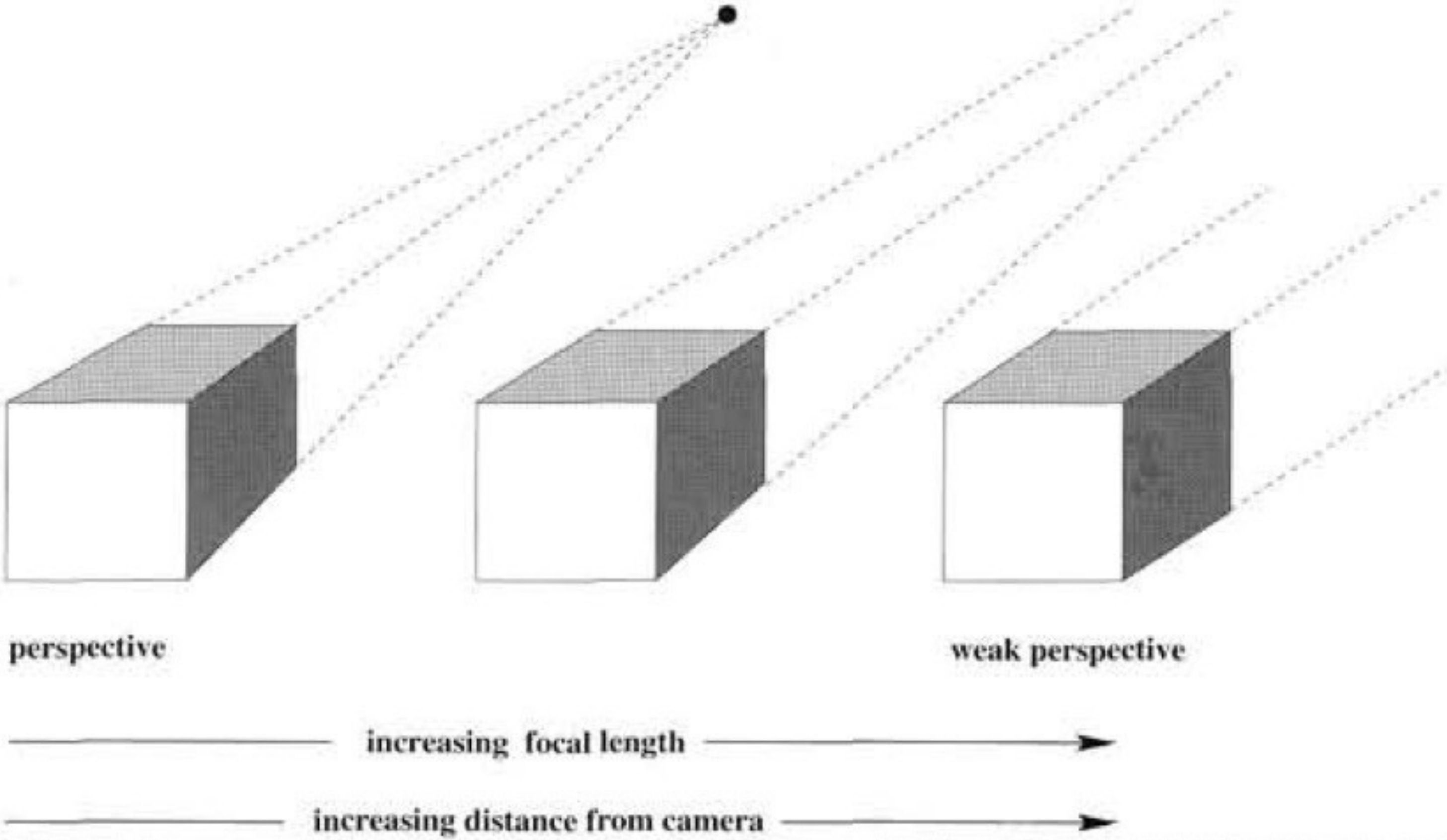
real-world object



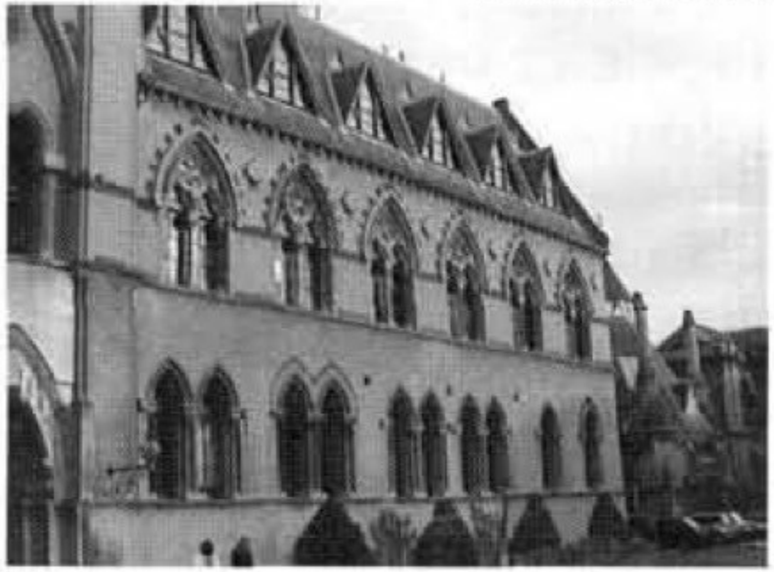

… we continue increasing Z and f while maintaining same magnification?

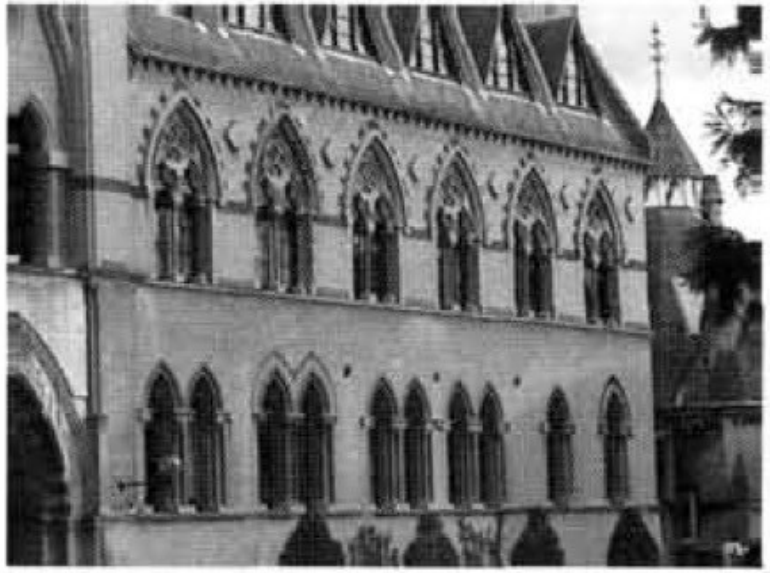
$$f \rightarrow \infty \text{ and } \frac{f}{Z} = \text{constant}$$

camera is *close* to object and has *small* focal length

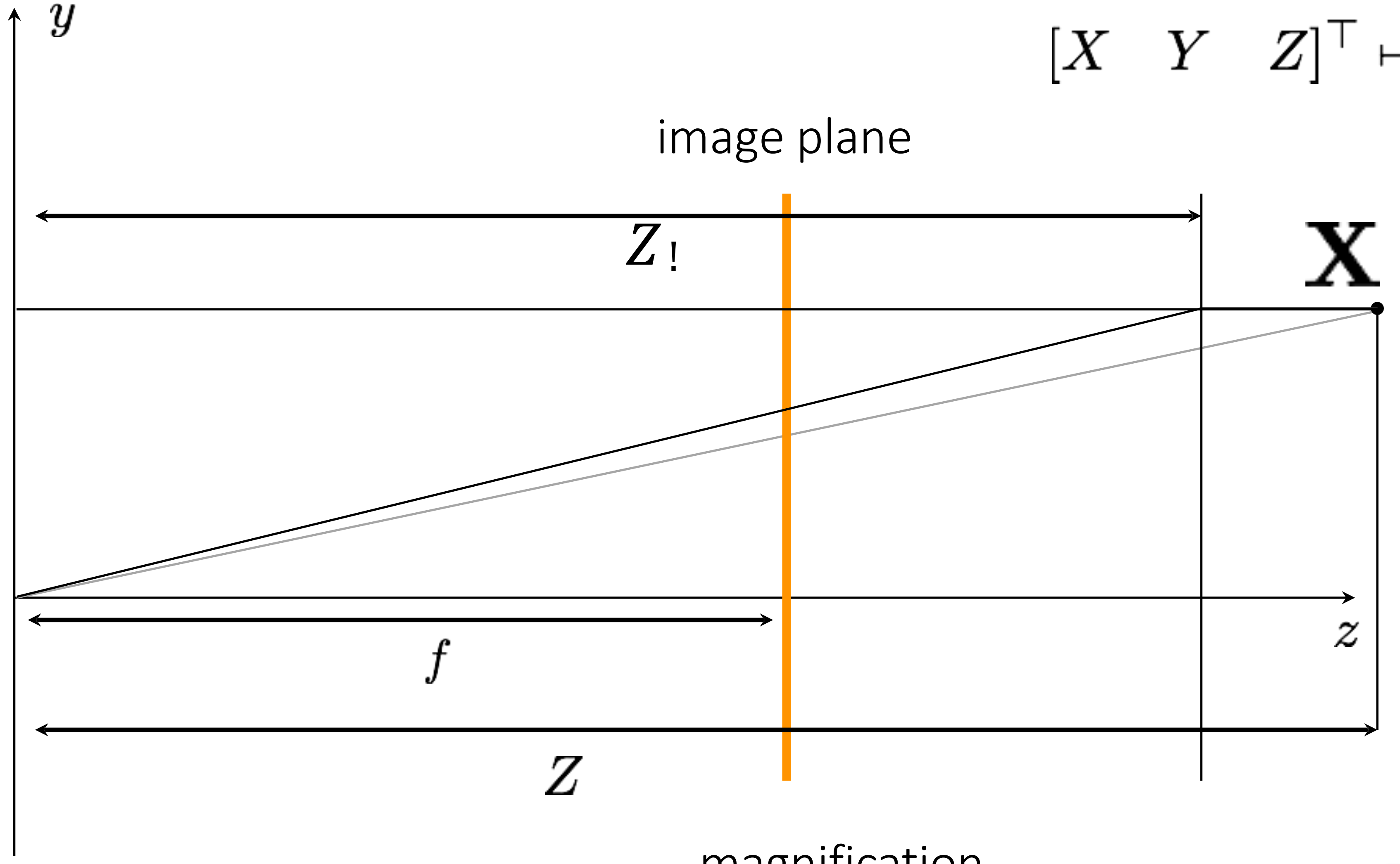



camera is *far* from object and has *large* focal length





# Weak perspective vs perspective camera



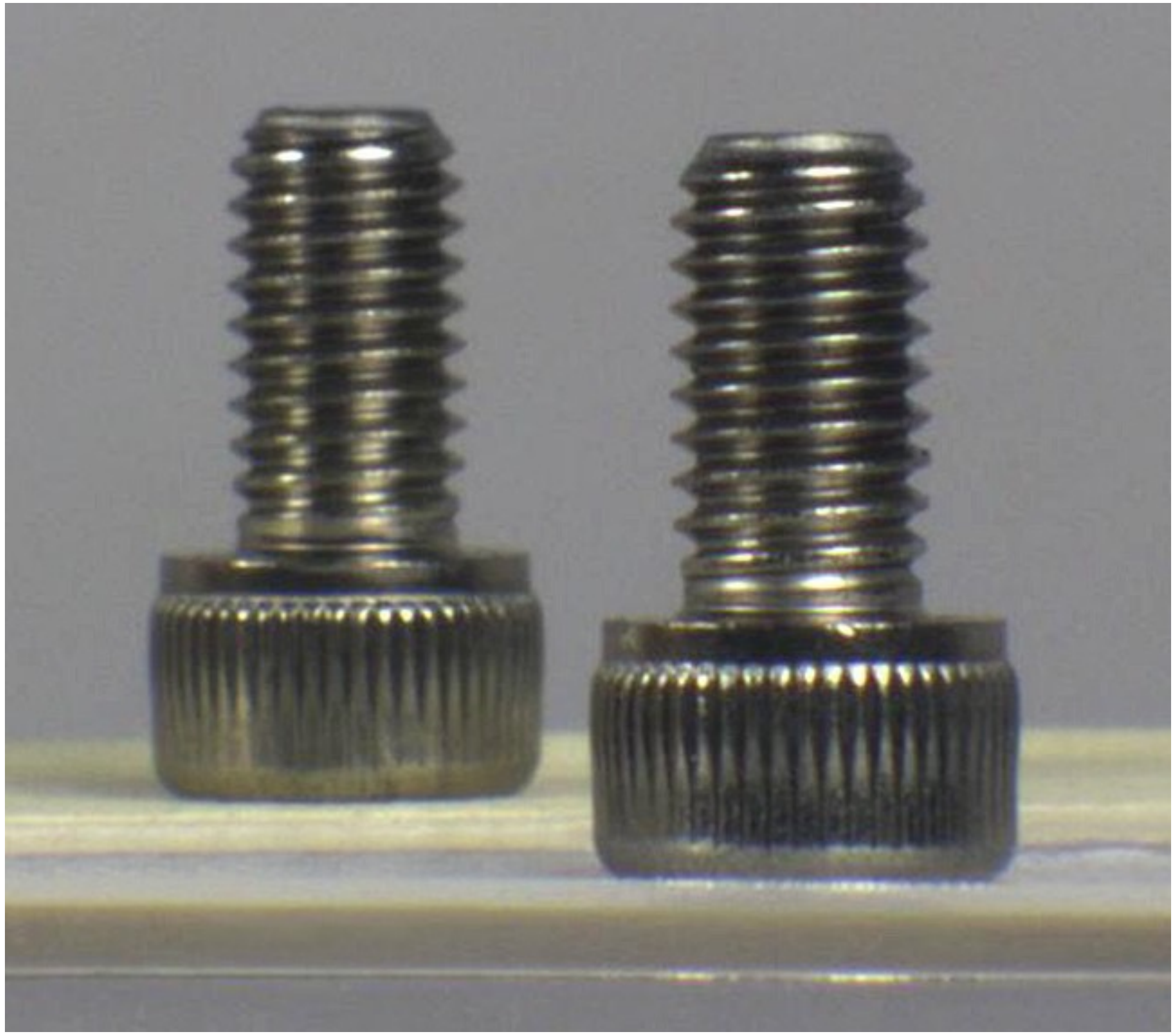

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z_o \quad fY/Z_o]^\top$$

- magnification does not change with depth
- *constant* magnification depending on f and $Z_o$

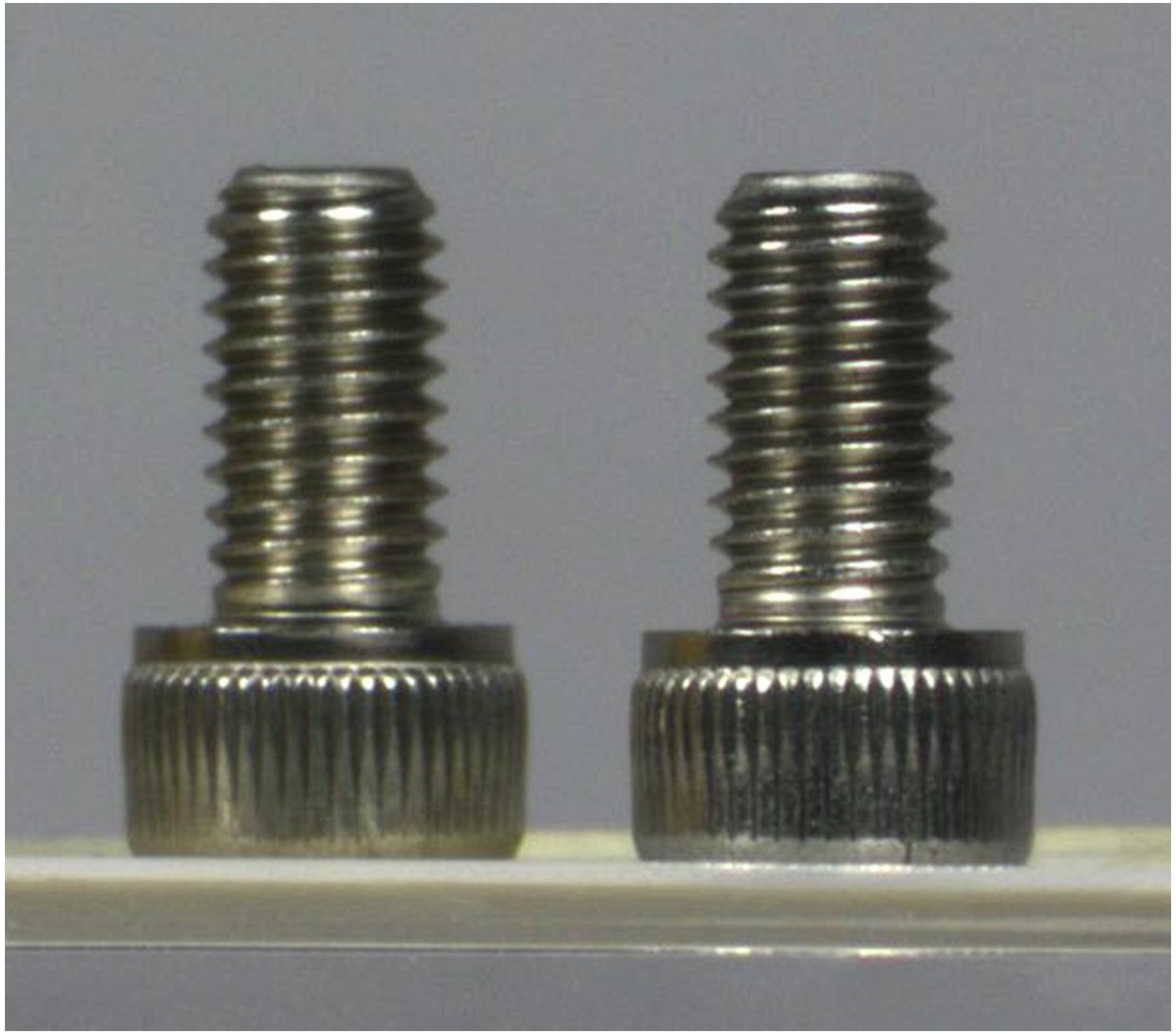
magnification changes with depth

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

# Different cameras



perspective camera



weak perspective camera

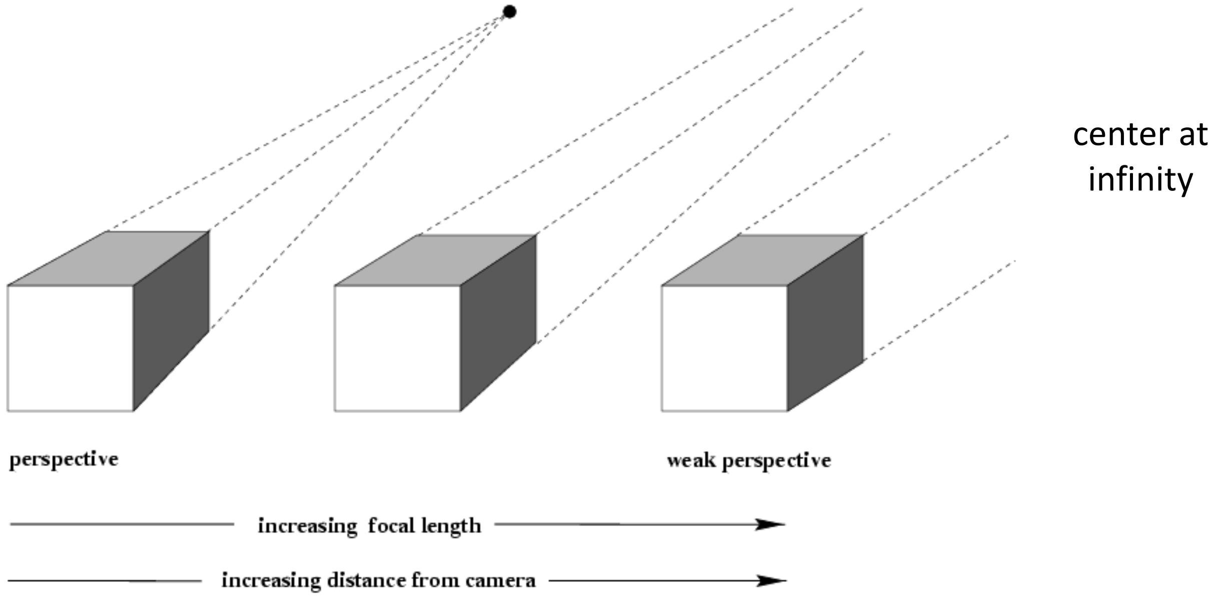
# When can we assume a weak perspective camera?

When the scene (or parts of it) is very far away.

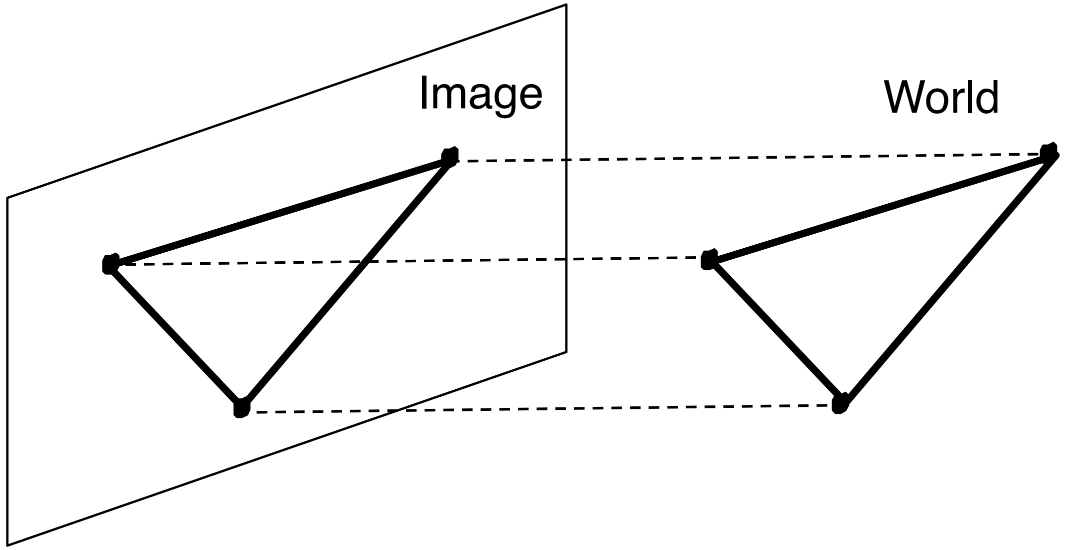
Weak perspective projection applies to the mountains.

# Affine structure from motion

- Let's start with *affine* or *weak perspective* cameras



# Orthographic projection



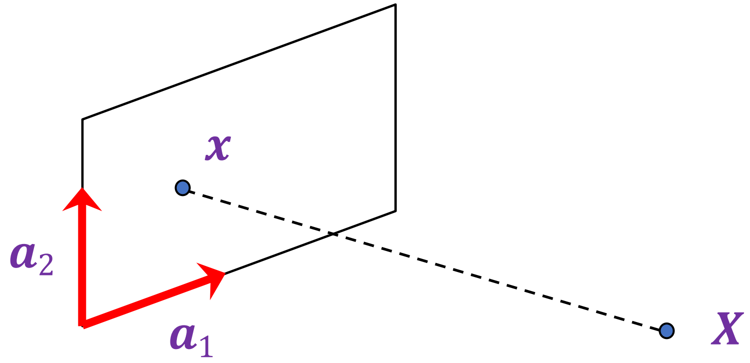

Just drop the $z$ coordinate!

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

# General affine projection

- A general affine projection is a 3D-to-2D linear mapping plus translation:

$$\boldsymbol{P} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$




$\boldsymbol{a}_1, \boldsymbol{a}_2$: rows of projection matrix

- In non-homogeneous coordinates:
- $$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \boldsymbol{AX} + \boldsymbol{t}$$

Projection of
world origin

# Affine structure from motion

- **Given**: $m$ images of $n$ fixed 3D points such that
  - $\boldsymbol{x}_{ij} = \boldsymbol{A}_i \boldsymbol{X}_j + \boldsymbol{t}_i, \quad i = 1, \dots, m, \; j = 1, \dots, n$
- **Problem**: use the $mn$ correspondences $\boldsymbol{x}_{ij}$ to estimate $m$ projection matrices $\boldsymbol{A}_i$ and translation vectors $\boldsymbol{t}_i$, and $n$ points $\boldsymbol{X}_j$
- The reconstruction is defined up to an arbitrary *affine* transformation $\boldsymbol{Q}$ (12 degrees of freedom):

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \boldsymbol{Q}^{-1}, \qquad \begin{pmatrix} \boldsymbol{X}_j \\ 1 \end{pmatrix} \rightarrow \boldsymbol{Q} \begin{pmatrix} \boldsymbol{X}_j \\ 1 \end{pmatrix}$$

- How many knowns and unknowns for $m$ images and $n$ points?
  - $2mn$ knowns and $8m + 3n$ unknowns
  - To be able to solve this problem, we must have $2mn \geq 8m + 3n - 12$ (affine ambiguity takes away 12 dof)
  - E.g., for two views, we need four point correspondences

# Affine structure from motion

- First, center the data by subtracting the centroid of the image points in each view:

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

$$= \boldsymbol{A}_i \boldsymbol{X}_j + \boldsymbol{t}_i - \frac{1}{n}\sum_{k=1}^{n} (\boldsymbol{A}_i \boldsymbol{X}_k + \boldsymbol{t}_i)$$

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

$$= \boldsymbol{A}_i \hat{\boldsymbol{X}}_j$$

# Affine structure from motion

- After centering, each normalized 2D point $\hat{\boldsymbol{x}}_{ij}$ is related to the 3D point by

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

- We can get rid of the need to center the 3D data (and the translation ambiguity) by defining the origin of the world coordinate system as the centroid of the 3D points

# Affine structure from motion

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

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

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

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:

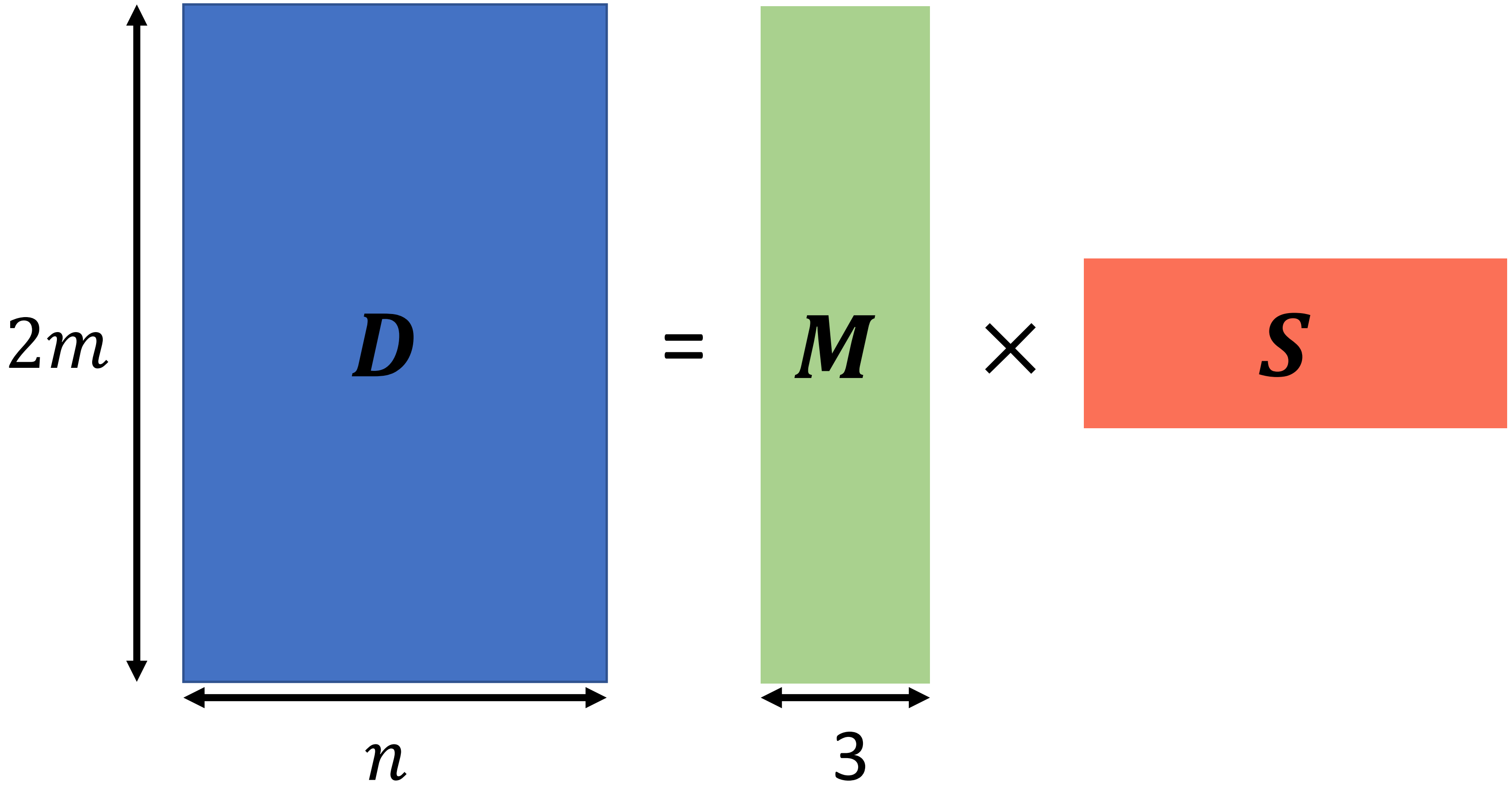
- $$\boldsymbol{D} = \begin{bmatrix} \hat{\boldsymbol{x}}_{11} & \hat{\boldsymbol{x}}_{12} & \cdots & \hat{\boldsymbol{x}}_{1n} \\ \hat{\boldsymbol{x}}_{21} & \hat{\boldsymbol{x}}_{22} & \cdots & \hat{\boldsymbol{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{x}}_{m1} & \hat{\boldsymbol{x}}_{m2}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

- What must be the rank of the measurement matrix $\boldsymbol{D} = \boldsymbol{MS}$?

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

# Factorizing the measurement matrix

- We want:



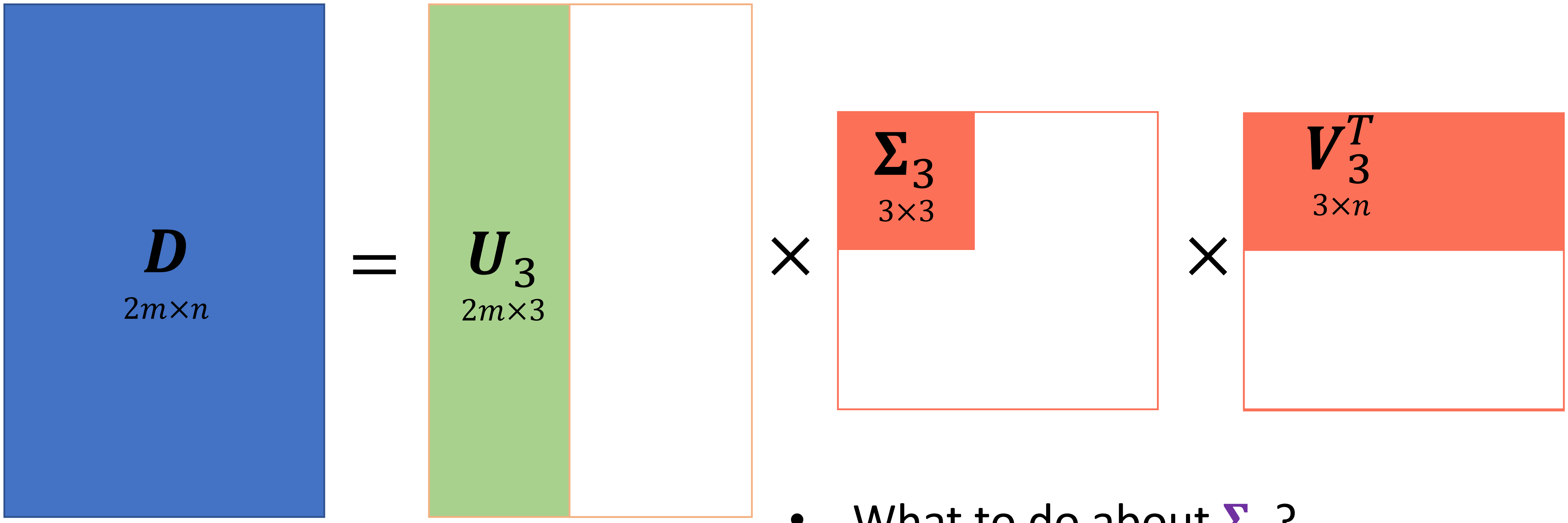
# Factorizing the measurement matrix

- Perform SVD of $\boldsymbol{D}$:

$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times n}{\boldsymbol{U}} \times \underset{n\times n}{\boldsymbol{\Sigma}} \times \underset{n\times n}{\boldsymbol{V}^T}$$

# Factorizing the measurement matrix

- Keep top 3 singular values:
- This is the closest approximation of $\boldsymbol{D}$ with a rank-3 matrix in terms of Frobenius norm




- What to do about $\boldsymbol{\Sigma}_3$?
- One solution: $\boldsymbol{M} = \boldsymbol{U}_3\boldsymbol{\Sigma}_3^{\frac{1}{2}}, \boldsymbol{S} = \boldsymbol{\Sigma}_3^{\frac{1}{2}}\boldsymbol{V}_3^T$

# Factorizing the measurement matrix

- One possible solution:

$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times 3}{\boldsymbol{M}} \times \underset{3\times n}{\boldsymbol{S}}$$

$$\boldsymbol{S} = \boldsymbol{\Sigma}_3^{\frac{1}{2}} \boldsymbol{V}_3^T$$

$$\boldsymbol{M} = \boldsymbol{U}_3 \boldsymbol{\Sigma}_3^{\frac{1}{2}}$$

# Factorizing the measurement matrix

- Other possible solutions (Ambiguity in Reconstruction)

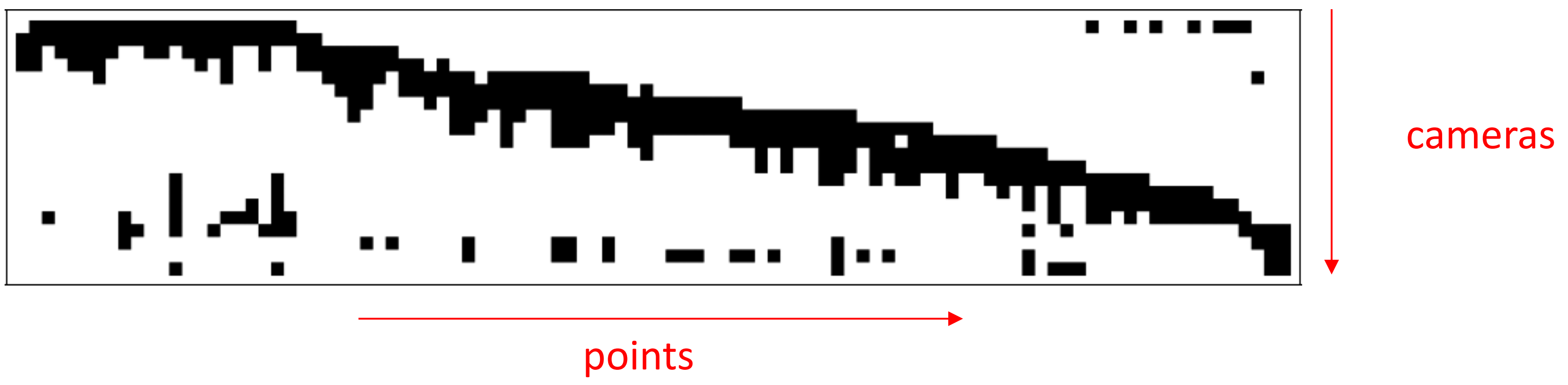
$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times 3}{\boldsymbol{M}} \times \underset{3\times 3}{\boldsymbol{Q}} \times \underset{3\times 3}{\boldsymbol{Q}^{-1}} \times \underset{3\times n}{\boldsymbol{S}}$$

How to eliminate ambiguity?
Assume certain special structure of the projection matrix.
Assume certain conditions about the 3D structure.

We can estimate $\boldsymbol{Q}$ to give the camera matrices in $\boldsymbol{M}$ desirable properties, like orthographic projection

# 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:



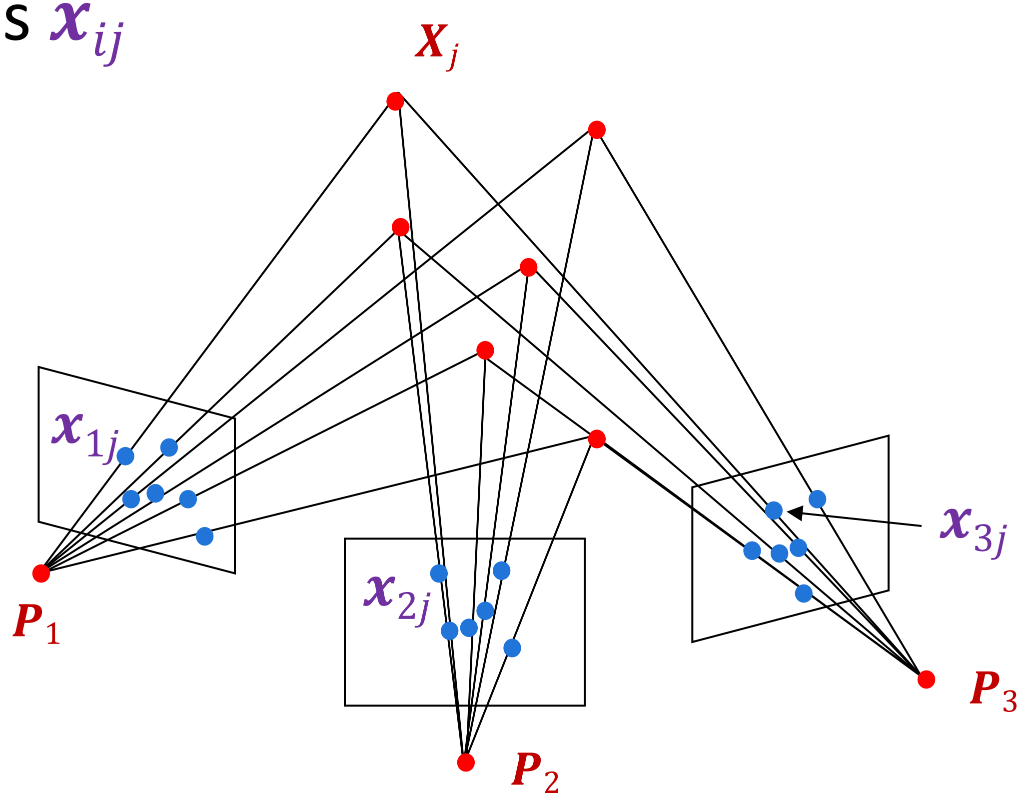

- These kind of problems are called Low-rank Matrix Completion problems (aka the Netflix Problem). Solved with convex/non-convex optimizations.
- Very popular before deep learning era!

# Today's Class

- Ambiguities in SfM
- Affine SfM
- Projective SfM
  - Global SfM
  - Incremental SfM
- Challenges and Applications

# Projective structure from motion

- **Given**: $m$ images of $n$ fixed 3D points such that (ignoring visibility):
  - $\boldsymbol{x}_{ij} \cong \boldsymbol{P}_i \boldsymbol{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$
- **Problem**: estimate $m$ projection matrices $\boldsymbol{P}_i$ and $n$ 3D points $\boldsymbol{X}_j$ from the $mn$ correspondences $\boldsymbol{x}_{ij}$



# Projective structure from motion

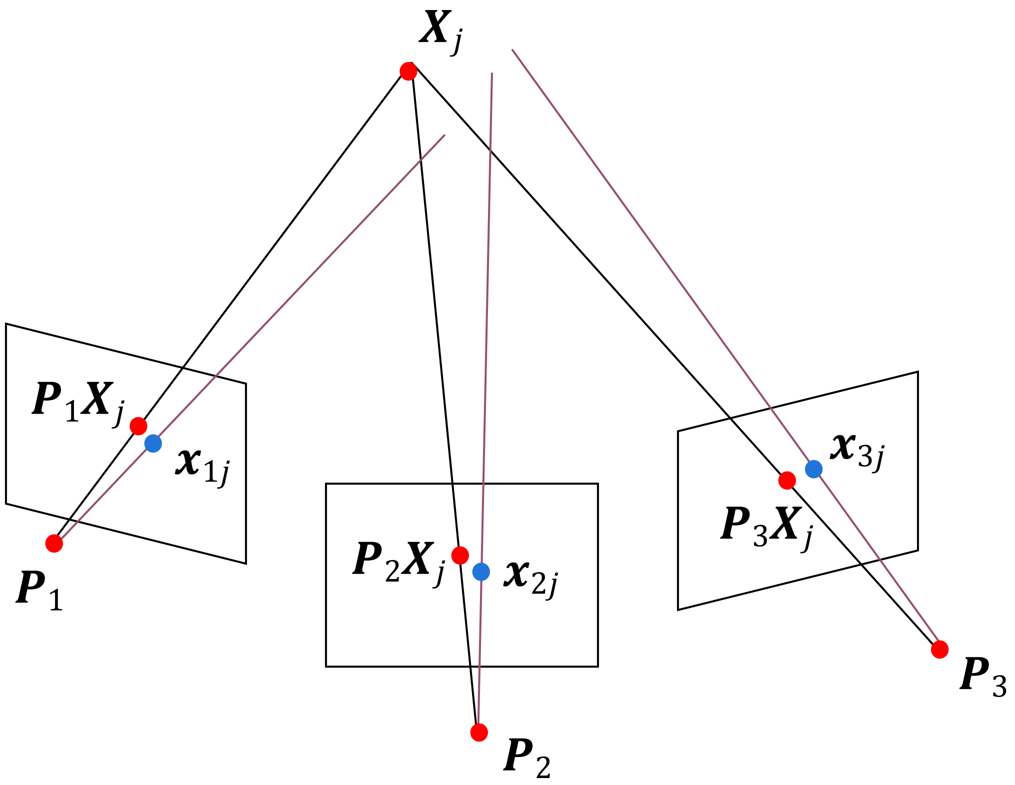
- **Given**: $m$ images of $n$ fixed 3D points such that (ignoring visibility):
  - $\boldsymbol{x}_{ij} \cong \boldsymbol{P}_i \boldsymbol{X}_j, \qquad i = 1, \dots, m, \quad j = 1, \dots, n$
- **Problem**: estimate $m$ projection matrices $\boldsymbol{P}_i$ and $n$ 3D points $\boldsymbol{X}_j$ from the $mn$ correspondences $\boldsymbol{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a $4 \times 4$ projective transformation $\boldsymbol{Q}$:
  - $\boldsymbol{X} \rightarrow \boldsymbol{QX}, \boldsymbol{P} \rightarrow \boldsymbol{PQ}^{-1}$
- We can solve for structure and motion when $2mn \geq 11m + 3n - 15$
- For two cameras, at least 7 points are needed
- You can solve it similar to Affine SfM with matrix factorization.
- Algebraic methods are good for initializing a non-linear optimization problem.

# Bundle adjustment

- Non-linear method for refining structure and motion
- Minimize reprojection error (with lots of bells and whistles):

- $\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} d\left(\boldsymbol{x}_{ij} - \mathrm{proj}(\boldsymbol{P}_i \boldsymbol{X}_j)\right)^2$

visibility flag:
is point $j$ visible in view $i$?




Factorization based SfM works well for very small scenes with limited number of images, even then it produces poor result for most practical purposes.

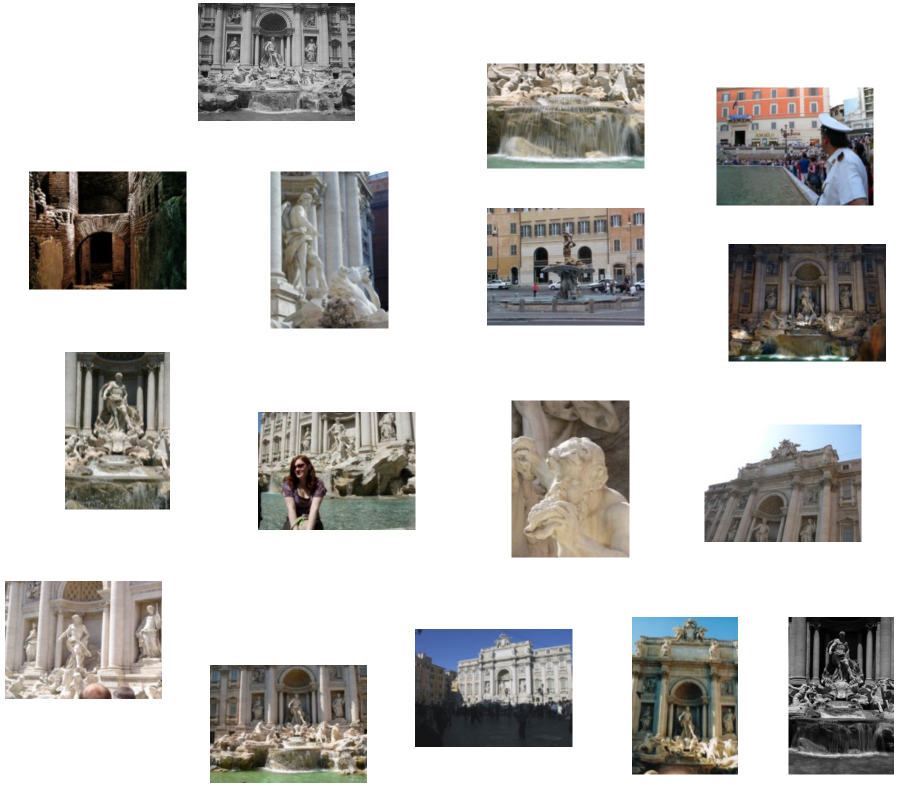
B. Triggs et al. [Bundle adjustment – A modern synthesis](). International Workshop on Vision Algorithms, 1999

# Global Structure from Motion

SfM for large scale scenes

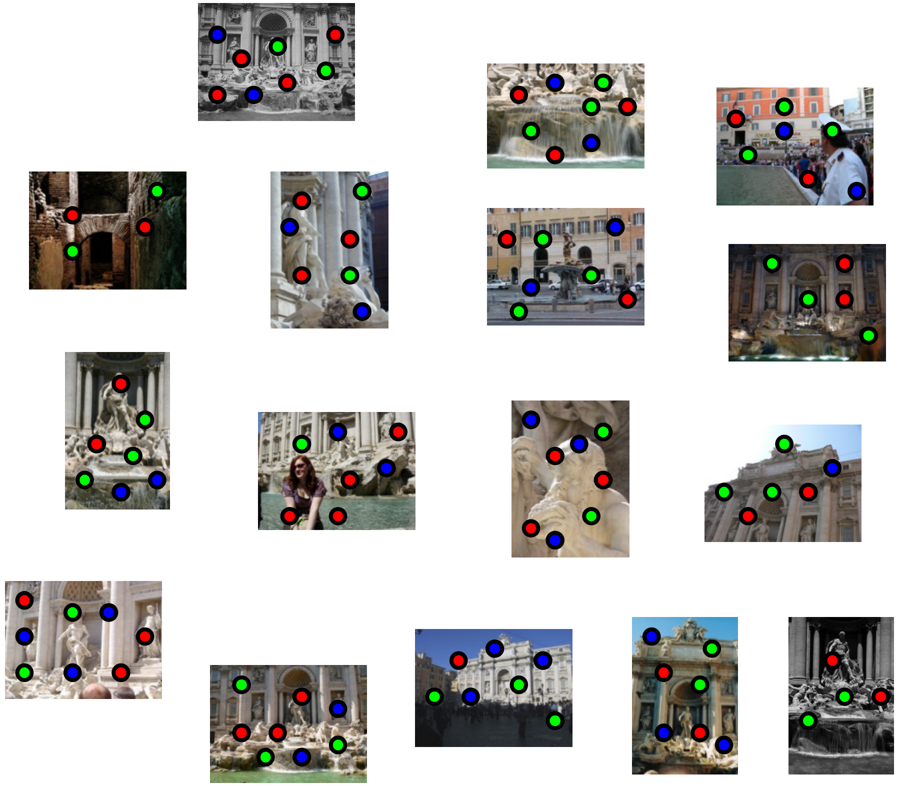
# Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



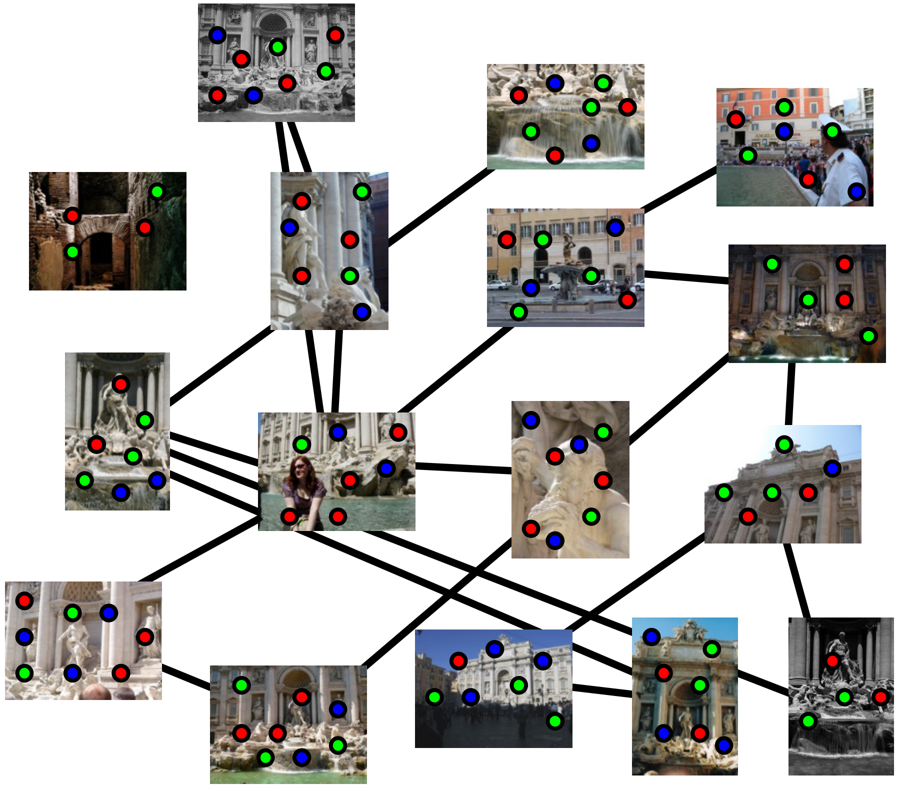
# Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



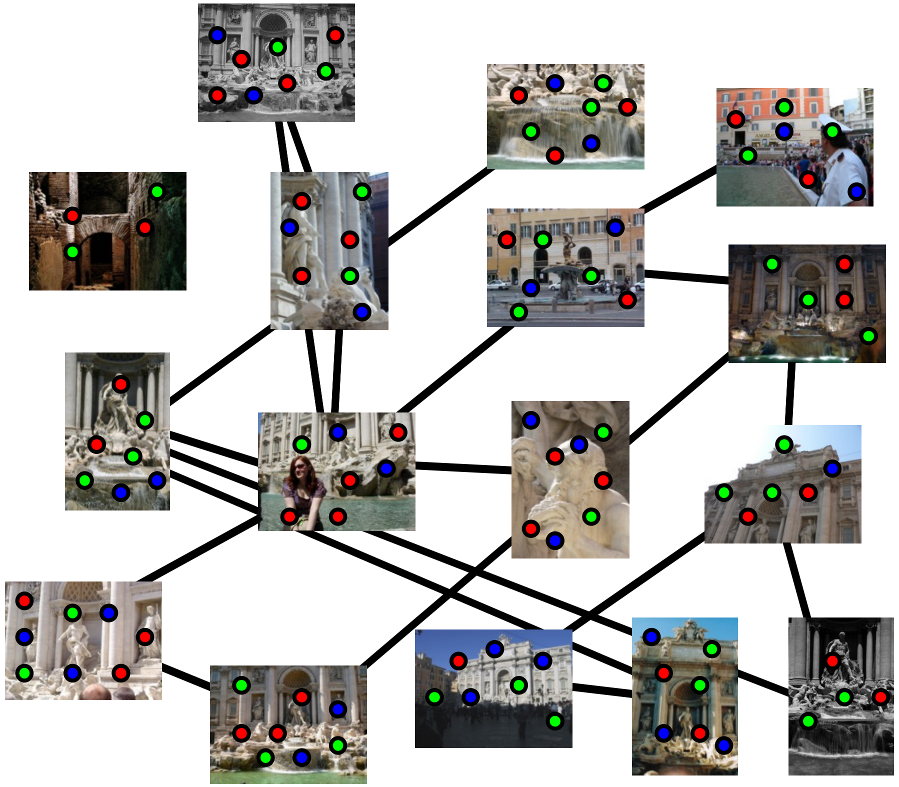
# Feature matching

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

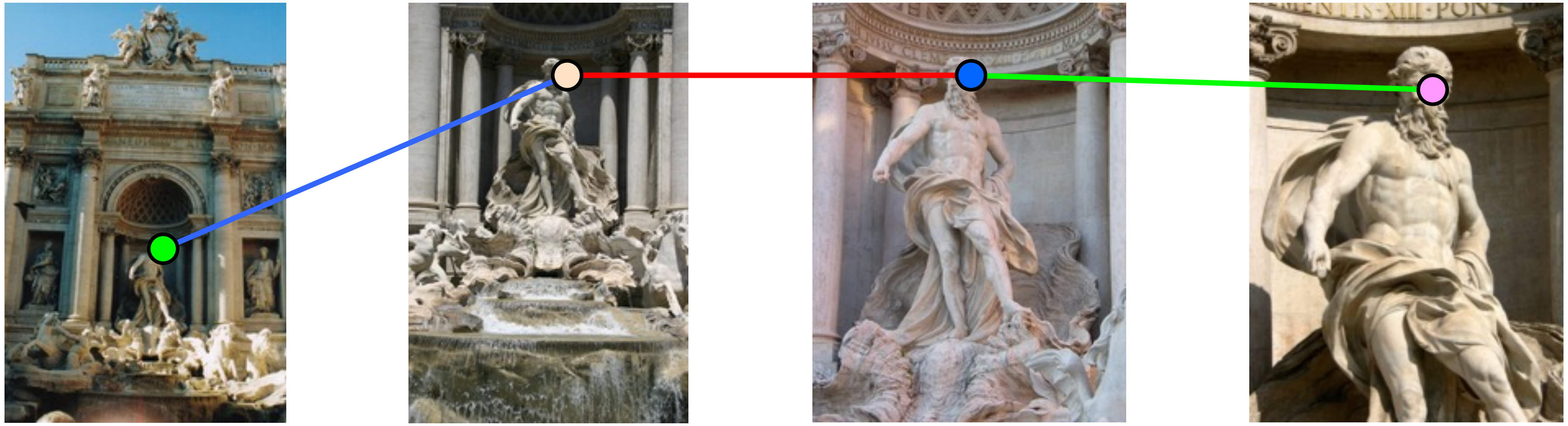


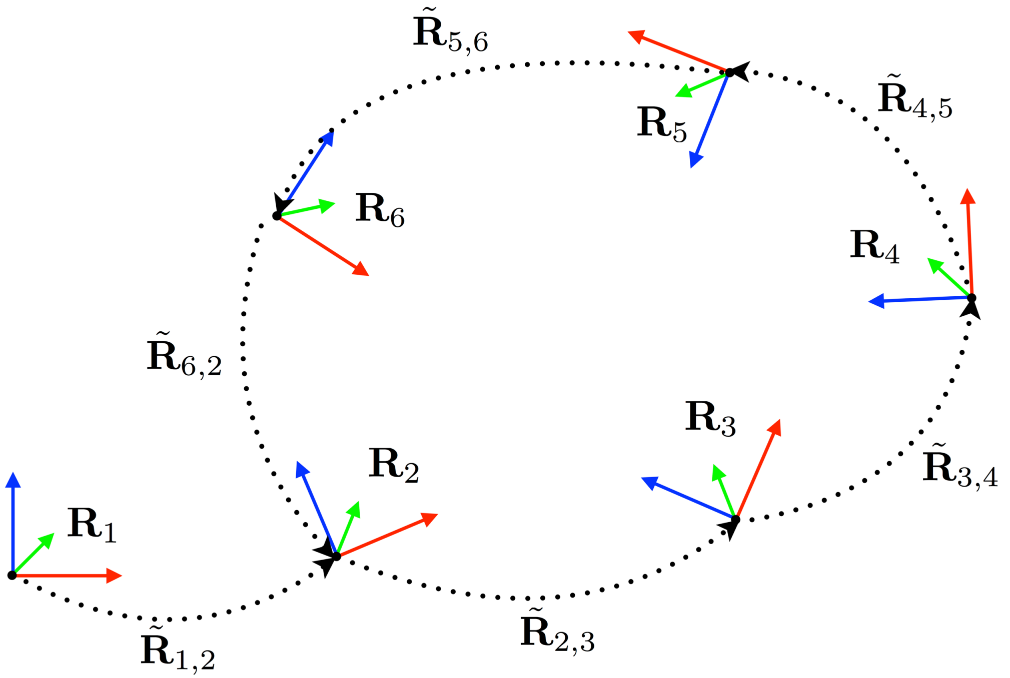
Image 1 Image 2 Image 3 Image 4

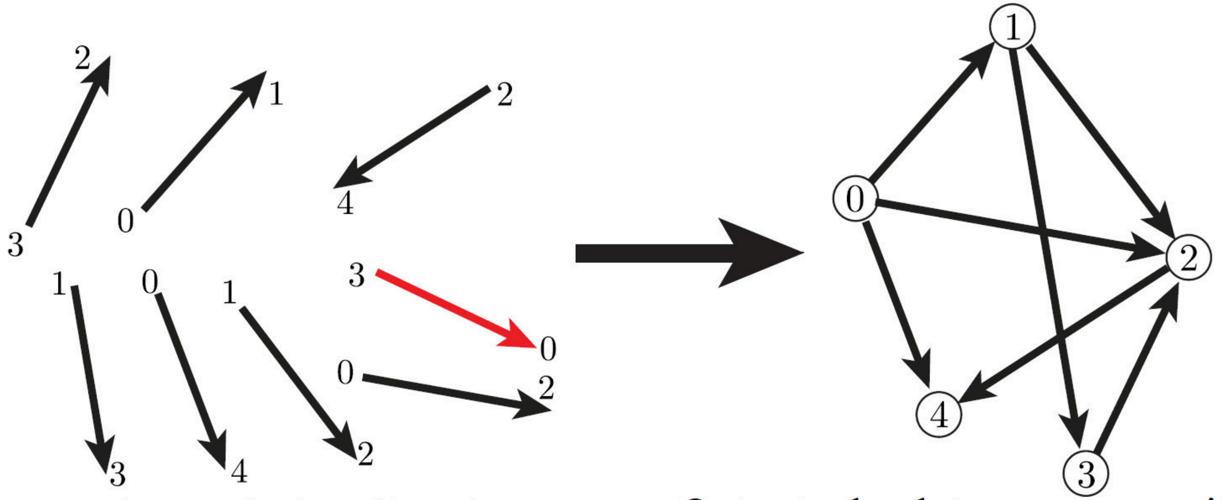
# Global SfM

- Given N images, there are ${}^{N}C_2$ pairs. Many of these pairs will have no overlaps in views and/or Fundamental/Essential matrix between them can not be reliably estimated using RANSAC.
  - Consider we have N0 (N0 < ${}^{N}C_2$) pairs of images with fundamental matrix estimated
- For each N0 pairs of images decompose essential matrix into relative rotation and translation between two cameras: $R_{ij}$ and $t_{ij}$.
- Can we solve for global (world coordinate) rotation and translation of the cameras, given pairwise measurements, i.e.
  - Given $R_{ij}$ and $t_{ij}$ for N0 pairs, find $R_k$ & $T_k$ for N cameras.
- Once we have the cameras we can better initialize the Bundle Adjustment problem.

# Rotation & Translation Averaging

# Given $R_{ij}$ and $t_{ij}$ for N0 pairs, find $R_k$ & $T_k$ for N cameras



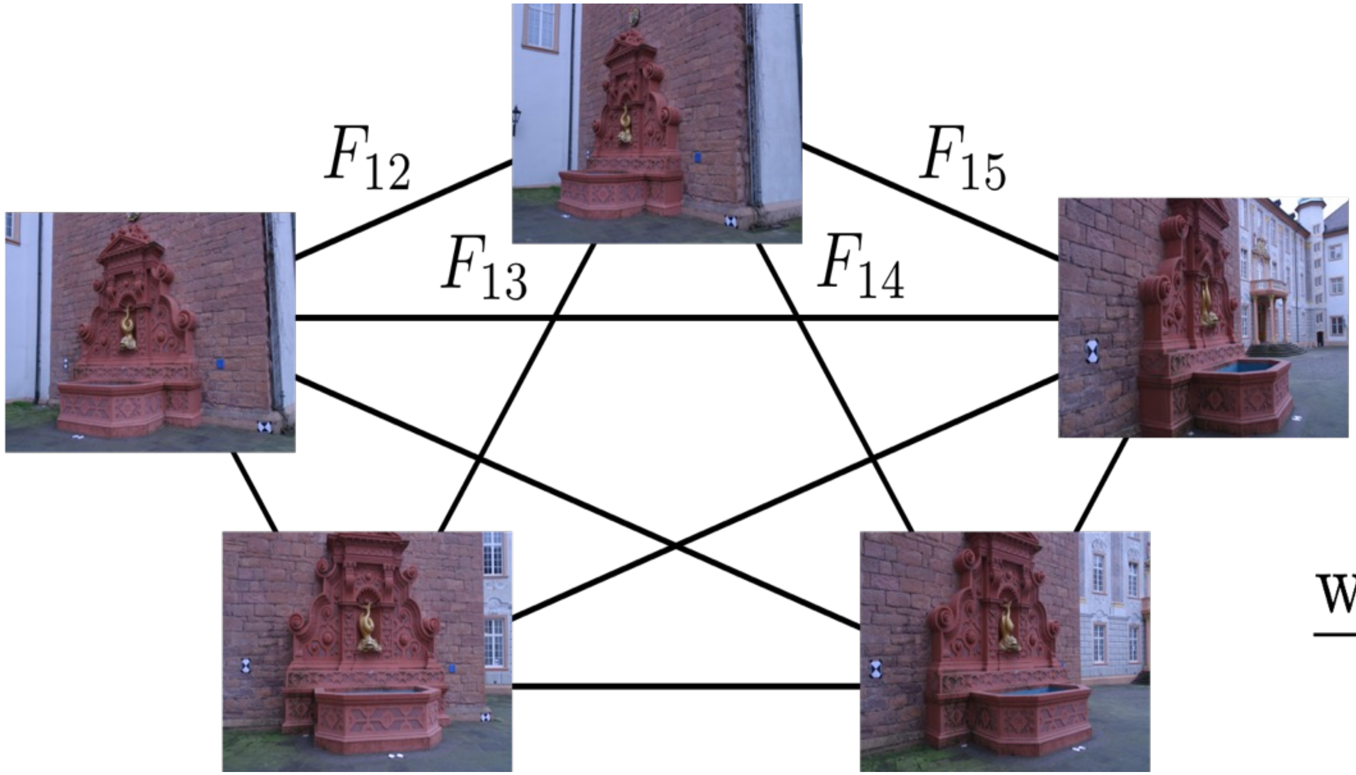



Input unit translation directions

Output: absolute camera positions

Credits: Venu Madhab Govindu

# Camera Pose estimation as matrix completion over Fundamental matrices




$$F = \begin{bmatrix} 0 & F_{12} & F_{13} & F_{14} & F_{15} \\ F_{21} & 0 & F_{23} & F_{24} & F_{25} \\ F_{31} & F_{32} & 0 & F_{34} & F_{35} \\ F_{41} & F_{42} & F_{43} & 0 & F_{45} \\ F_{51} & F_{52} & F_{53} & F_{54} & 0 \end{bmatrix}$$

$$\underline{\text{with } F = A + A^T \text{ and } rank(A) = 3.}$$

- Proves a low-rank property of all the cameras capturing different images of a scene.
- Solves a low-rank camera pose recovery algorithm from Structure from Motion.

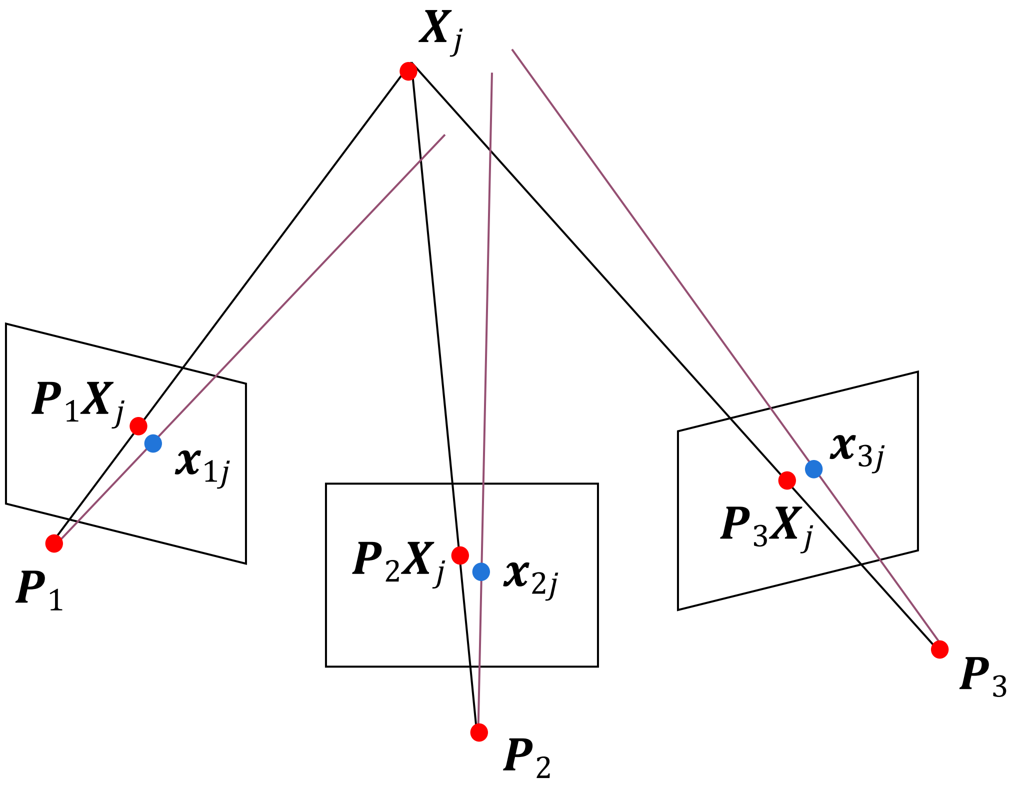


“A New Rank Constraint on Multi-view Fundamental Matrices”, Sengupta et. al. CVPR 2017.

# Bundle adjustment

- Non-linear method for refining structure and motion
- Minimize reprojection error (with lots of bells and whistles):
- 
- $\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} d\left(\boldsymbol{x}_{ij} - \text{proj}(\boldsymbol{P}_i \boldsymbol{X}_j)\right)^2$

visibility flag: is point $j$ visible in view $i$?

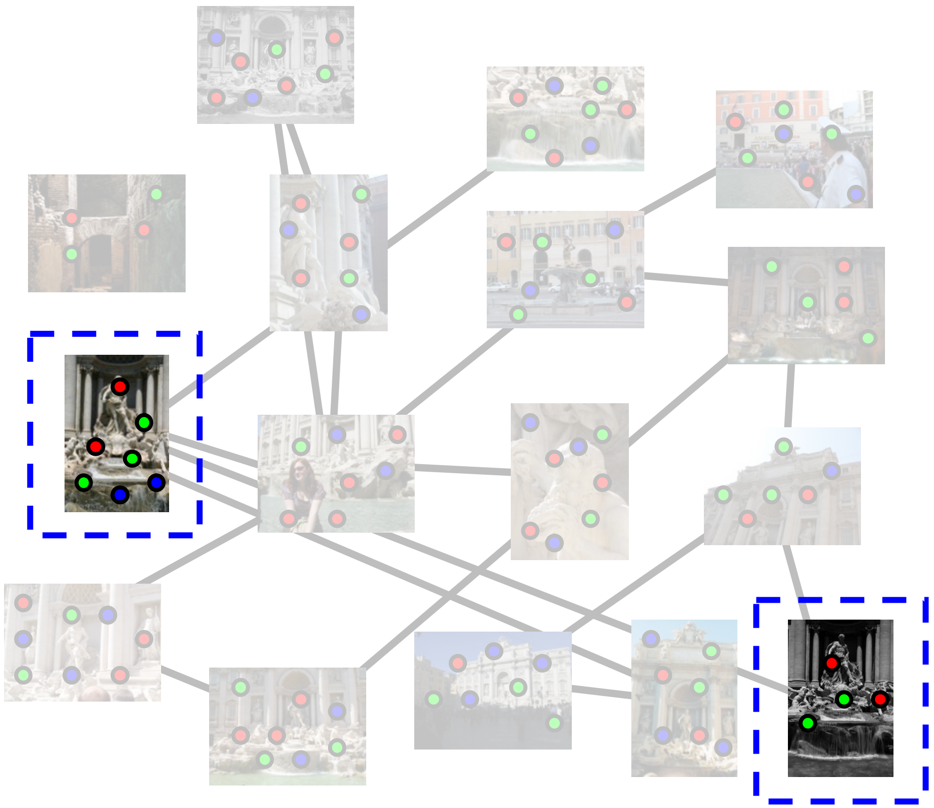



- **Initialize $P_i$'s by solving global SfM**
  - **Rotation Averaging**
  - **Translation Averaging**

B. Triggs et al. Bundle adjustment – A modern synthesis. International Workshop on Vision Algorithms, 1999

# Incremental SfM

Can handle large scale scene, more than Global SfM



- Automatically select an initial pair of images

# 1. Picking the initial pair

- We want a pair with many matches, but which has as large a baseline as possible

✓ lots of matches
✗ small baseline

✓ large baseline
✗ very few matches

✓ large baseline
✓ lots of matches

# Incremental SFM

- Pick a pair of images with lots of inliers (and preferably, good EXIF data)
  - Initialize intrinsic parameters (focal length, principal point) from EXIF
  - Estimate extrinsic parameters ($\boldsymbol{R}$ and $\boldsymbol{t}$) using five-point algorithm (similar to 8-pt algorithm but for essential matrix)
  - Use triangulation to initialize model points
- While remaining images exist
  - Find an image with many feature matches with images in the model
  - Run RANSAC on feature matches to register new image to 3D model points
  - Triangulate new points
  - Perform bundle adjustment to re-optimize everything
  - Optionally, align with GPS from EXIF data or ground control points

# Next Best View Problem

- Choice of next view impacts reconstruction quality
  - almost identical view => high uncertainty in triangulation
  - very different view => low overlap and high camera uncertainty
  - single bad choice may impact the whole reconstruction

- Popular next best view methods:
  - choose view with seeing the most triangulated points
  - minimize reconstruction uncertainty
    - depends on number of observations
    - distribution in the image

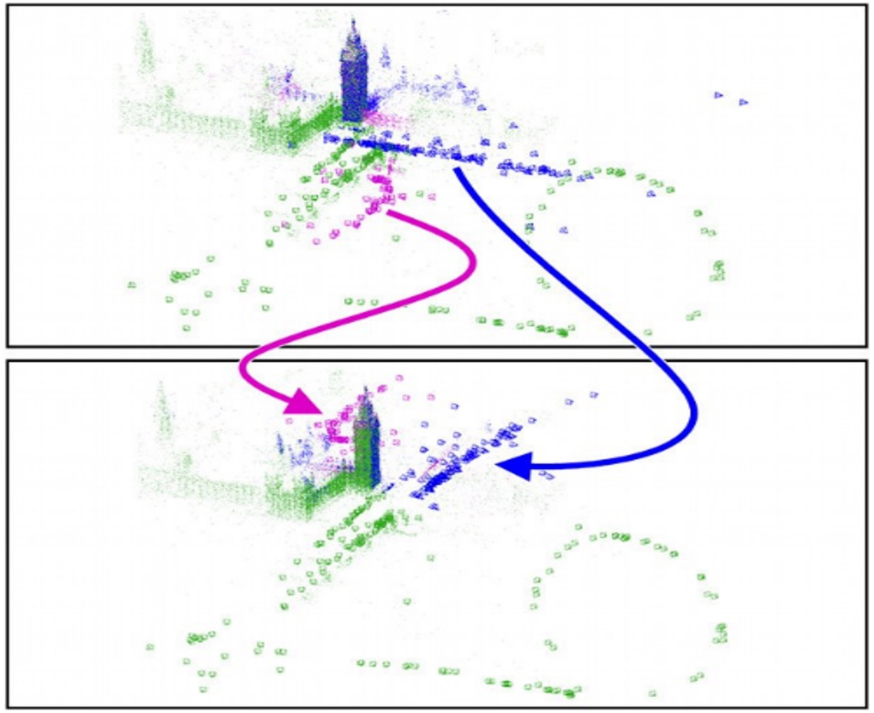
# Today’s Class

- Ambiguities in SfM
- Affine SfM
- Projective SfM
  - Global SfM
  - Incremental SfM
- Challenges and Applications

# The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries

# Repetitive structures cause catastrophic failures



https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

# Repetitive structures cause catastrophic failures


R. Kataria et al. [Improving Structure from Motion with Reliable Resectioning](). 3DV 2020

# The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries
- Reducing error accumulation and closing loops

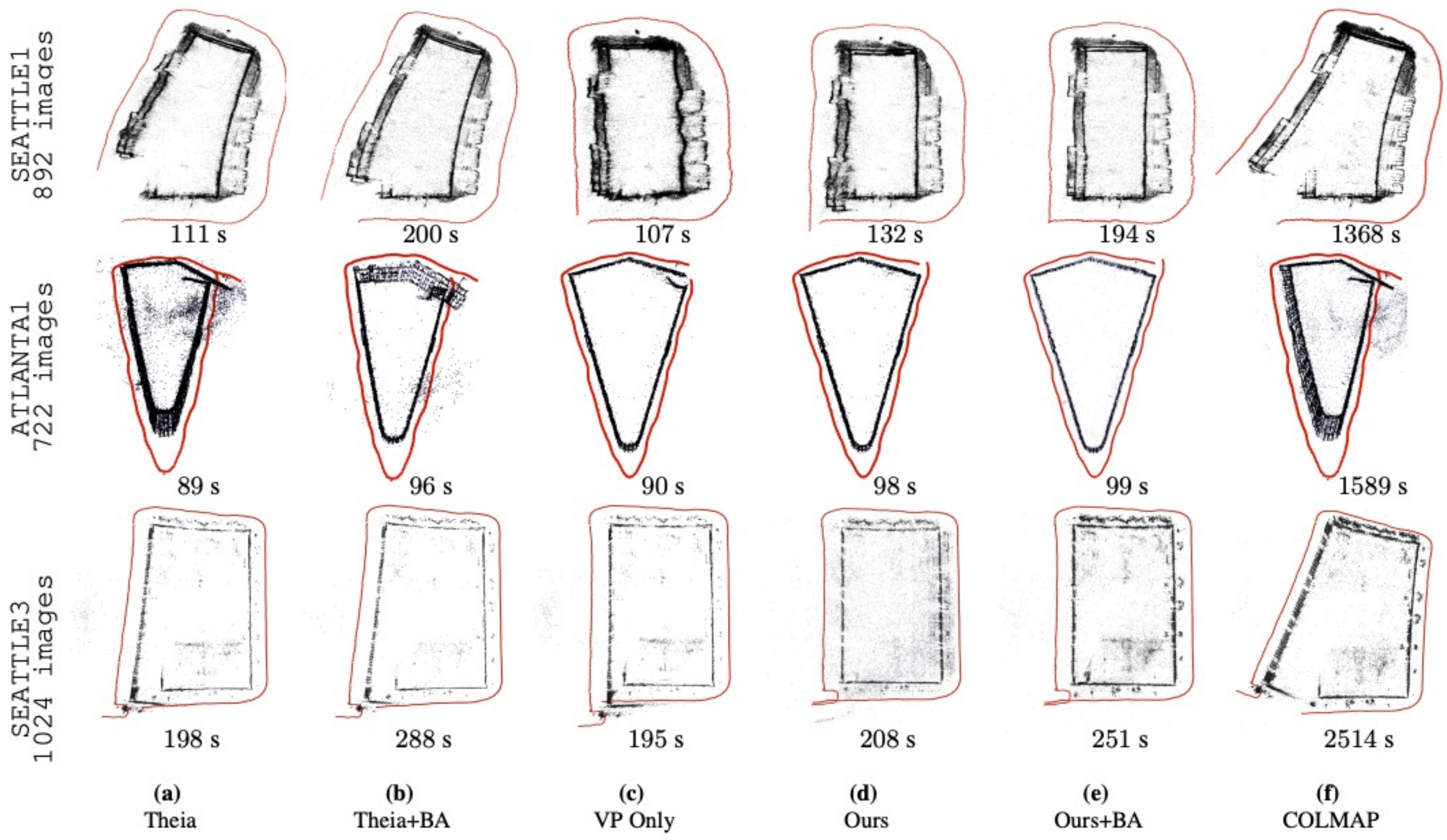
# Loop Detection/Closure

- Problem:
  - Structure from motion is an incremental process
  - Drift accumulates
- Mitigation:
  - Retrieval of long range connections

# Reducing error accumulation and closing loops

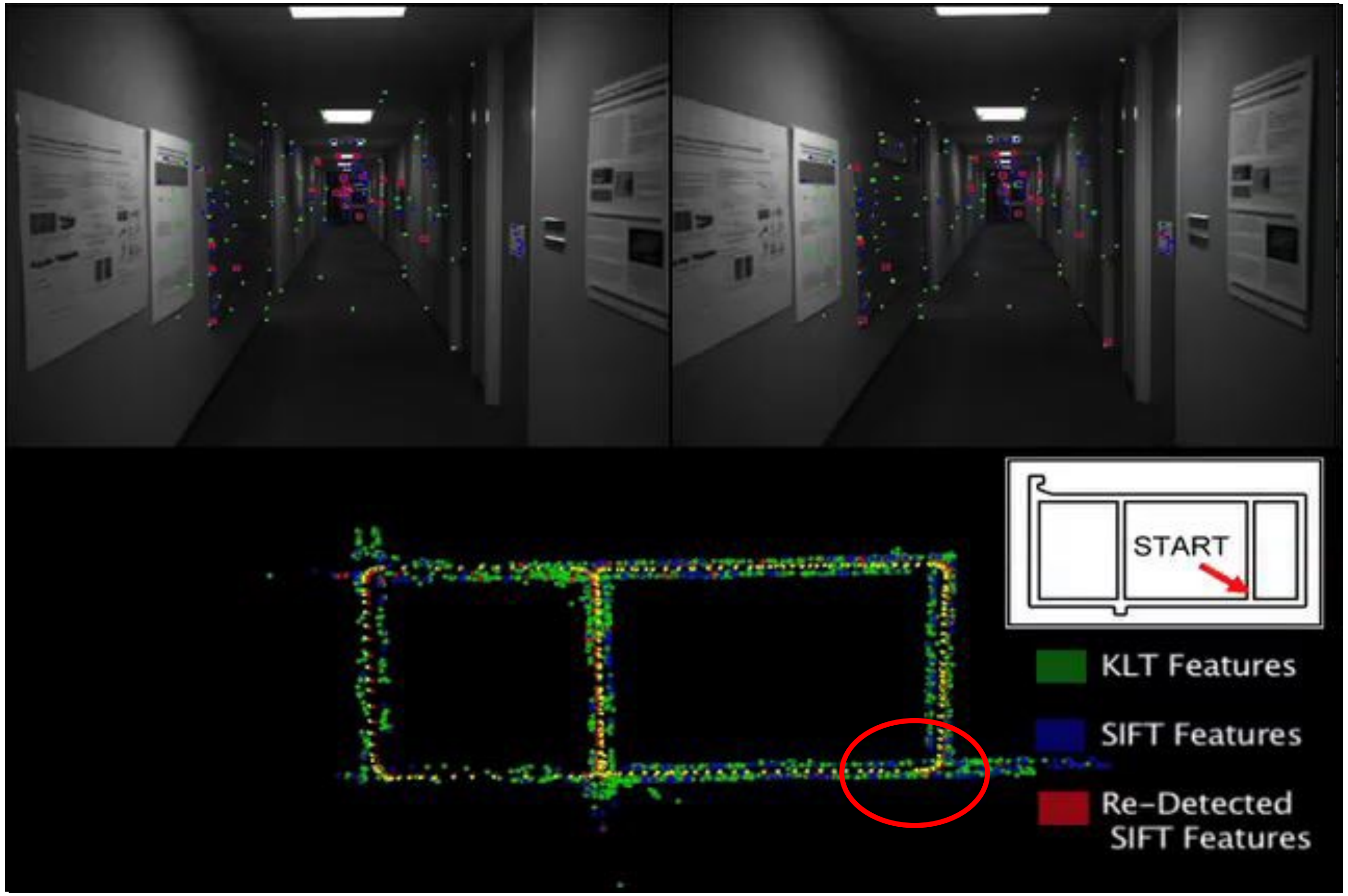

A. Holynski et al. [Reducing Drift in Structure From Motion Using Extended Features](). arXiv 2020

# Reducing error accumulation and closing loops

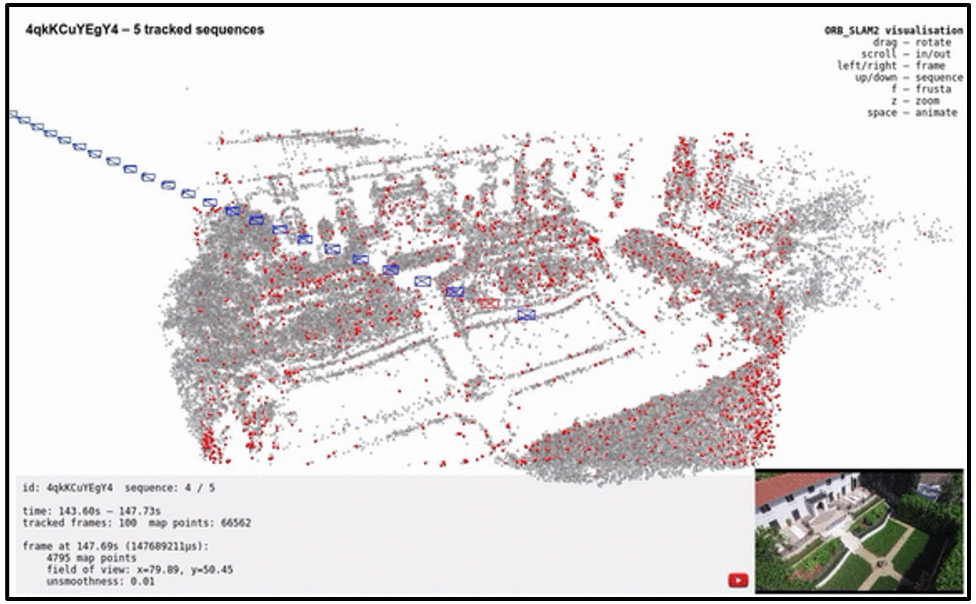



A. Holynski et al. [Reducing Drift in Structure From Motion Using Extended Features](). arXiv 2020

# Loop Closure

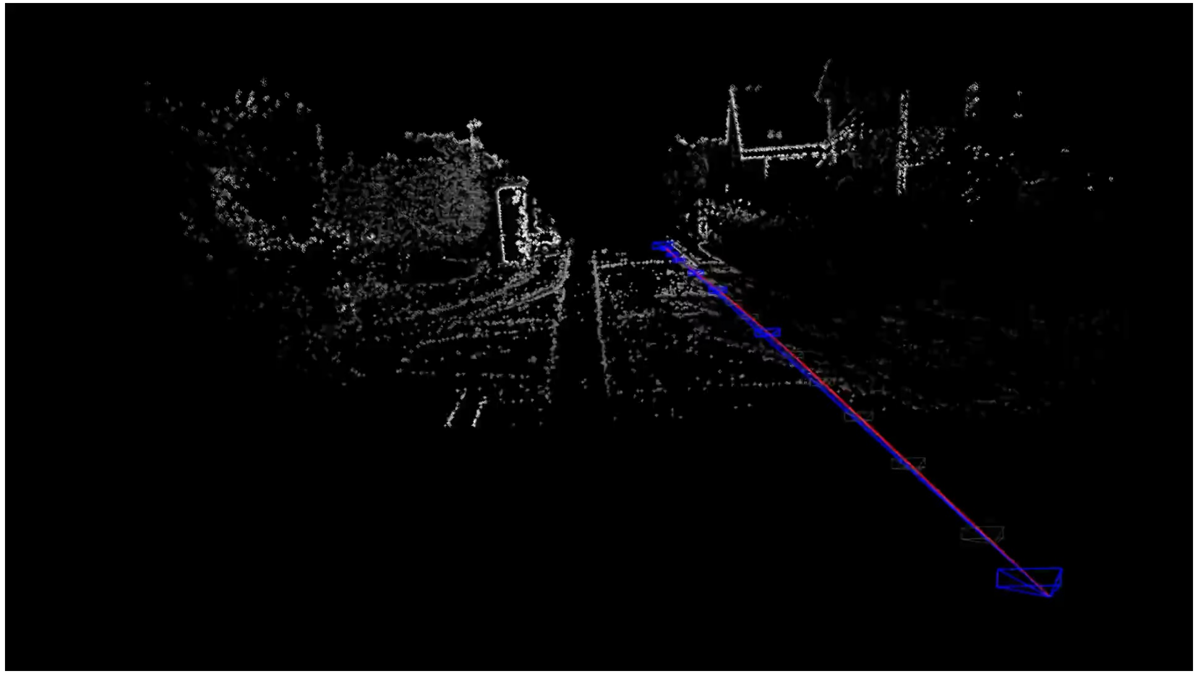


# Can also compute camera poses from video (often called Visual SLAM)



# Visual Simultaneous Localization and Mapping (V-SLAM)

- Main differences with SfM:
  - Continuous visual input from sensor(s) over time
  - Gives rise to problems such as loop closure
  - Often the goal is to be online / real-time



Video from Daniel Cremer's Lab

# SFM software

- Bundler
- OpenSfM
- OpenMVG
- VisualSFM
- COLMAP (Structure-from-motion revisited, JL Schonberger, JM Frahm, CVPR 2016, from UNC!)

- See also Wikipedia's list of toolboxes

# SfM applications

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

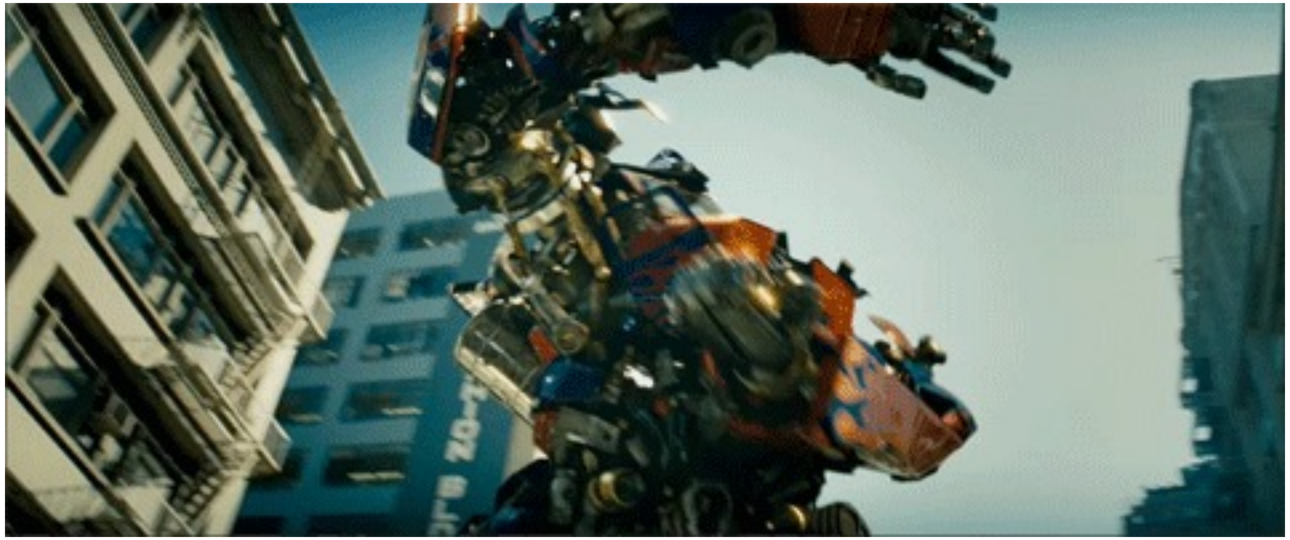
# Applications: Match Moving

Or Motion tracking, solving for camera trajectory

Integral for visual effects (VFX)

Why?





# Applications: Visual Reality & Augmented Reality

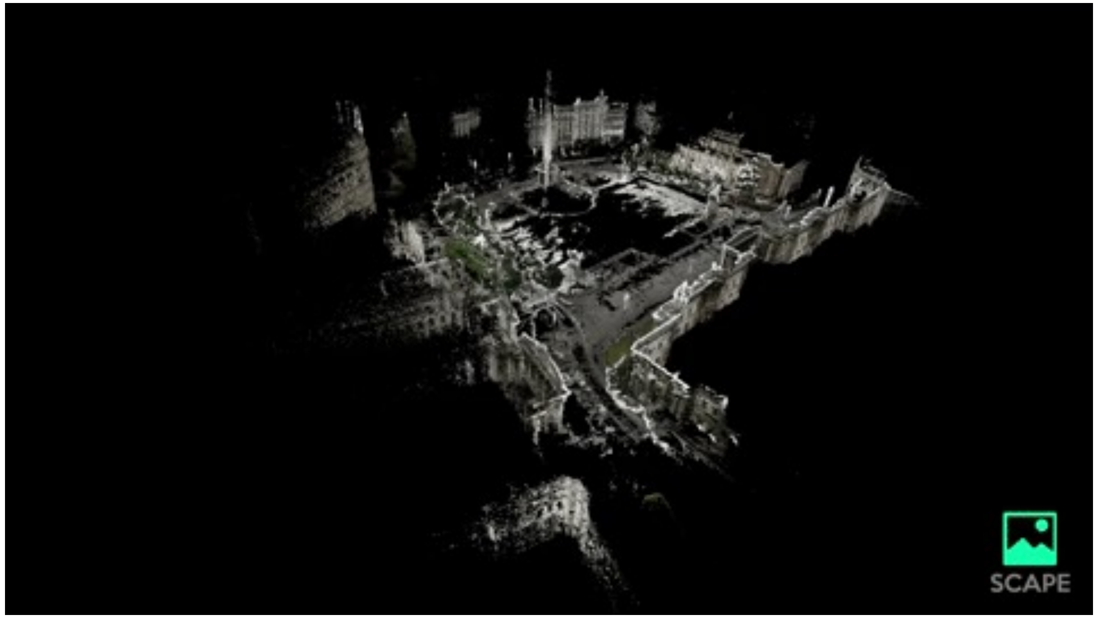
Oculus

https://www.youtube.com/watch?v=KOG7yTz1iTA

Hololens

https://www.youtube.com/watch?v=FMtvrTGnP04

# Applications: Visual Reality & Augmented Reality




Scape: Building the ‘AR Cloud’: Part Three —3D Maps, the Digital Scaffolding of the 21st Century

https://medium.com/scape-technologies/building-the-ar-cloud-part-three-3d-maps-the-digital-scaffolding-of-the-21st-century-465fa55782dd

# Application: AR walking directions


https://www.theverge.com/2019/8/8/20776247/google-maps-live-view-ar-walking-directions-ios-android-feature

# 3D model from video

Spatial ordering

↓ images, 2D feature correspondences

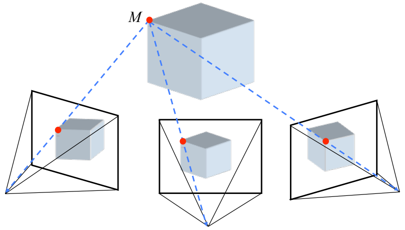
3D points & camera poses

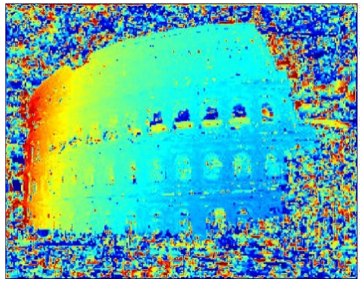
↓ images, camera poses

Dense 3D

↓ images, camera poses, depth maps

Model generation



# Summary: 3D geometric vision

- Fundamentals:
  - Camera Models: Intrinsic & Extrinsics
  - 3D to 2D projections, perspective distortions
  - Vanishing Points & Lines
  - Epipolar Geometry
  - Essential & Fundamental Matrices

- Core problems:
  - Camera calibration: single camera + two camera (estimate E/F matrix)
  - Stereo: depth from two calibrated cameras

- Reconstruction Techniques:
  - Active Stereo
  - Multi-view Stereo
  - Structure from Motion
  - Photometric Stereo (next class)

# Slide Credits

- CS5670, Introduction to Computer Vision, **Cornell Tech, by Noah Snavely.**
- CS 194-26/294-26: Intro to Computer Vision and Computational Photography, **UC Berkeley, by Angjoo Kanazawa.**
- **CS 543 Computer Vision, by Stevlana Lazebnik, UIUC.**
- **COMP 776, by Jan-Michael Frahm, UNC**