Lecture 21: Structure from motion

CS 5670: Introduction to Computer Vision

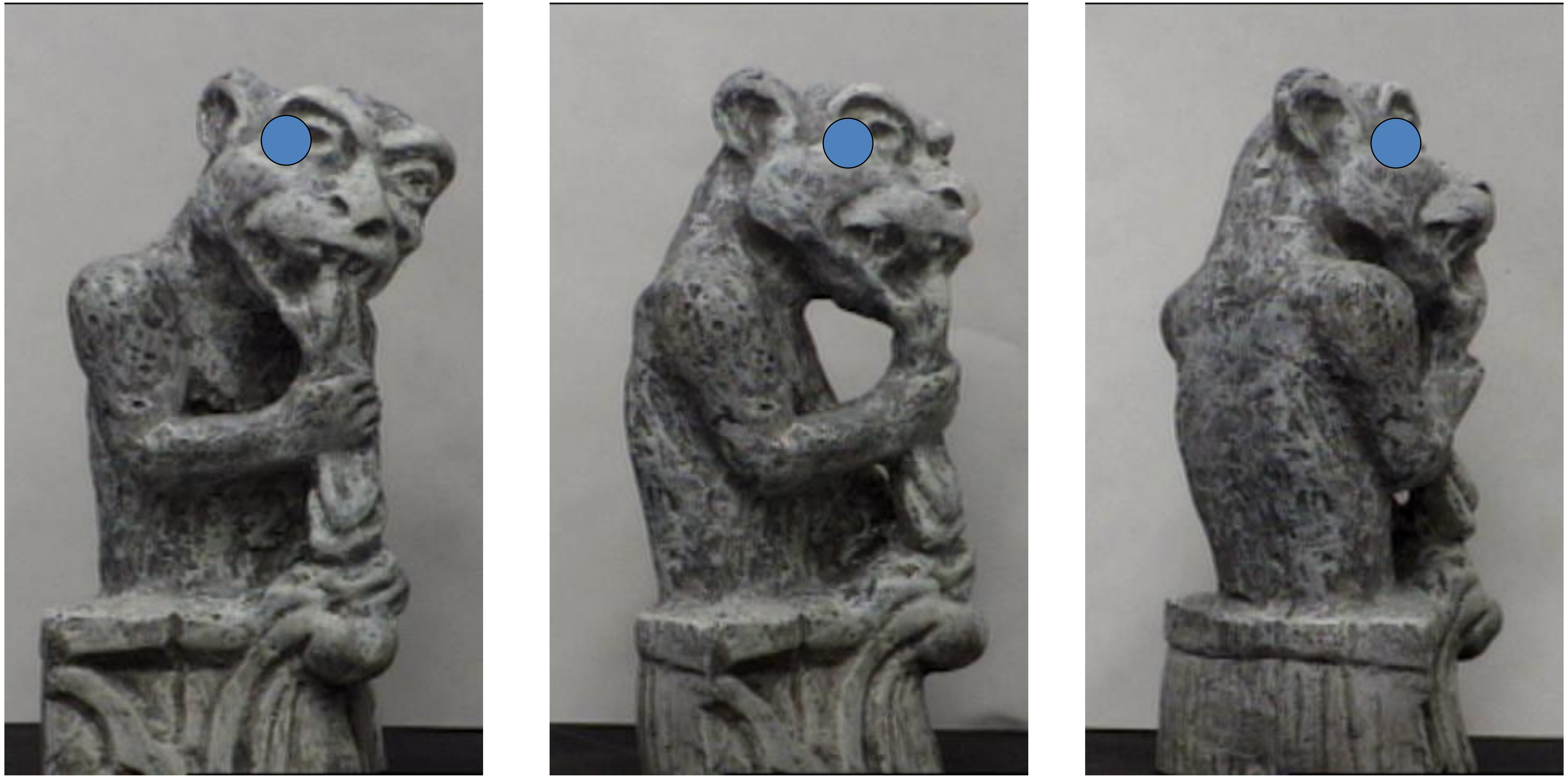
Most slides from Noah Snavely and David Fouhey

Announcements

- PS5 out tonight (panorama stitching)

Triangulation

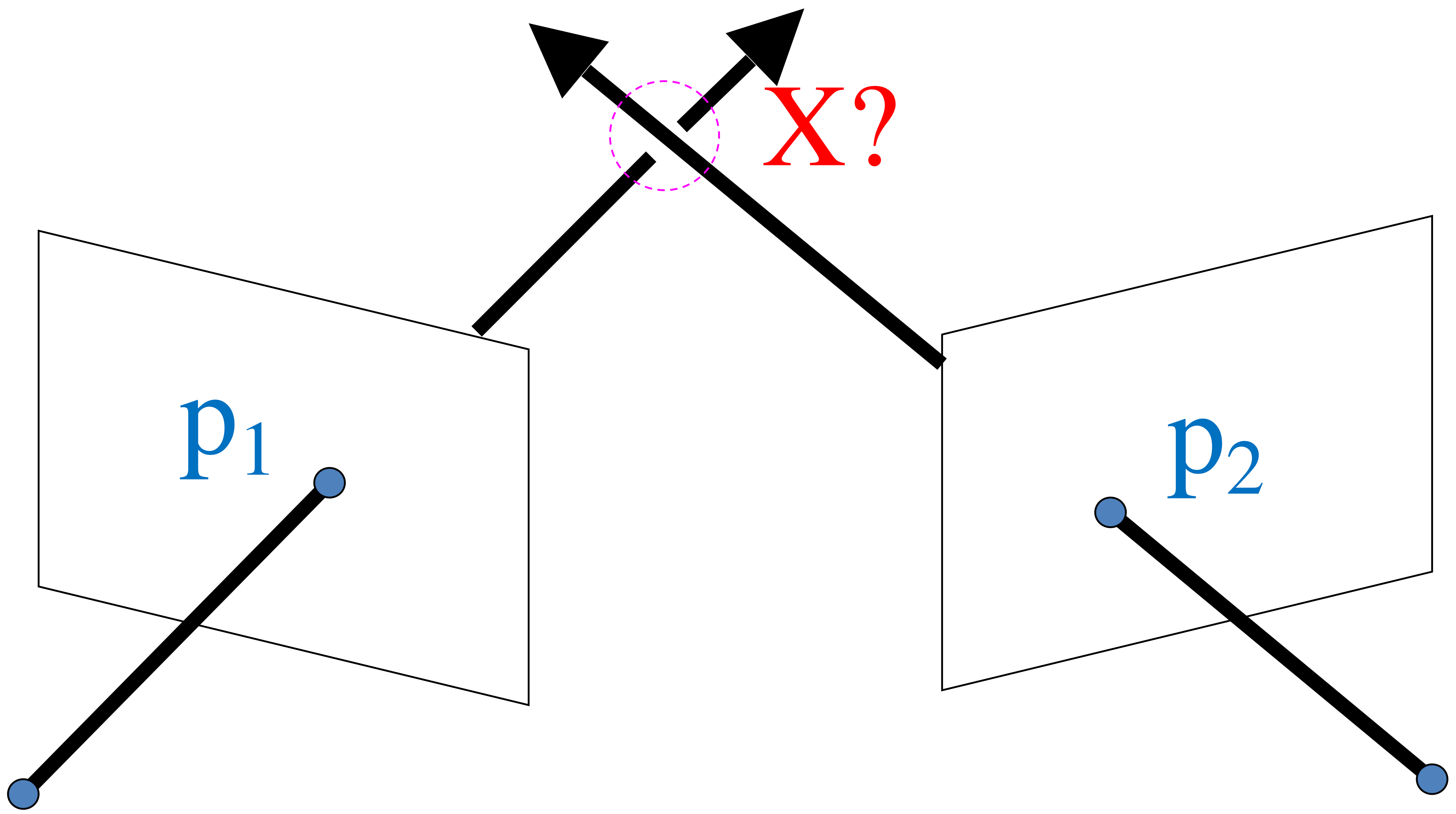
Given projection $\mathbf{p}_i$ of unknown 3D point $\mathbf{X}$ in two or more images (with known cameras $\mathbf{P}_i$), find $\mathbf{X}$



Source: D. Fouhey

Triangulation

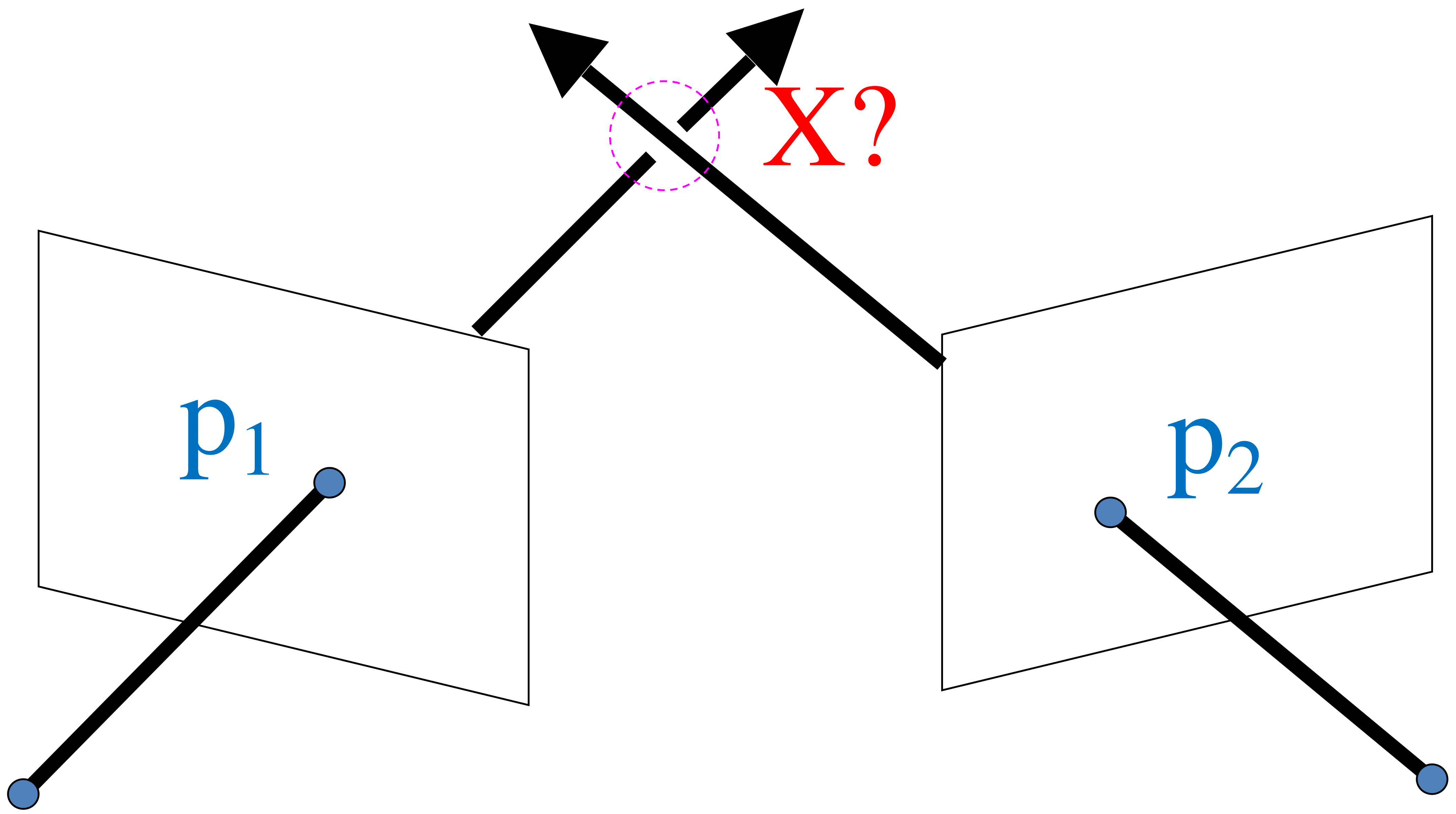
Given projection $\mathbf{p}_i$ of unknown 3D point $\mathbf{X}$ in two or more images (with known camera projection matrices $\mathbf{P}_i$), find $\mathbf{X}$



Source: D. Fouhey

Triangulation

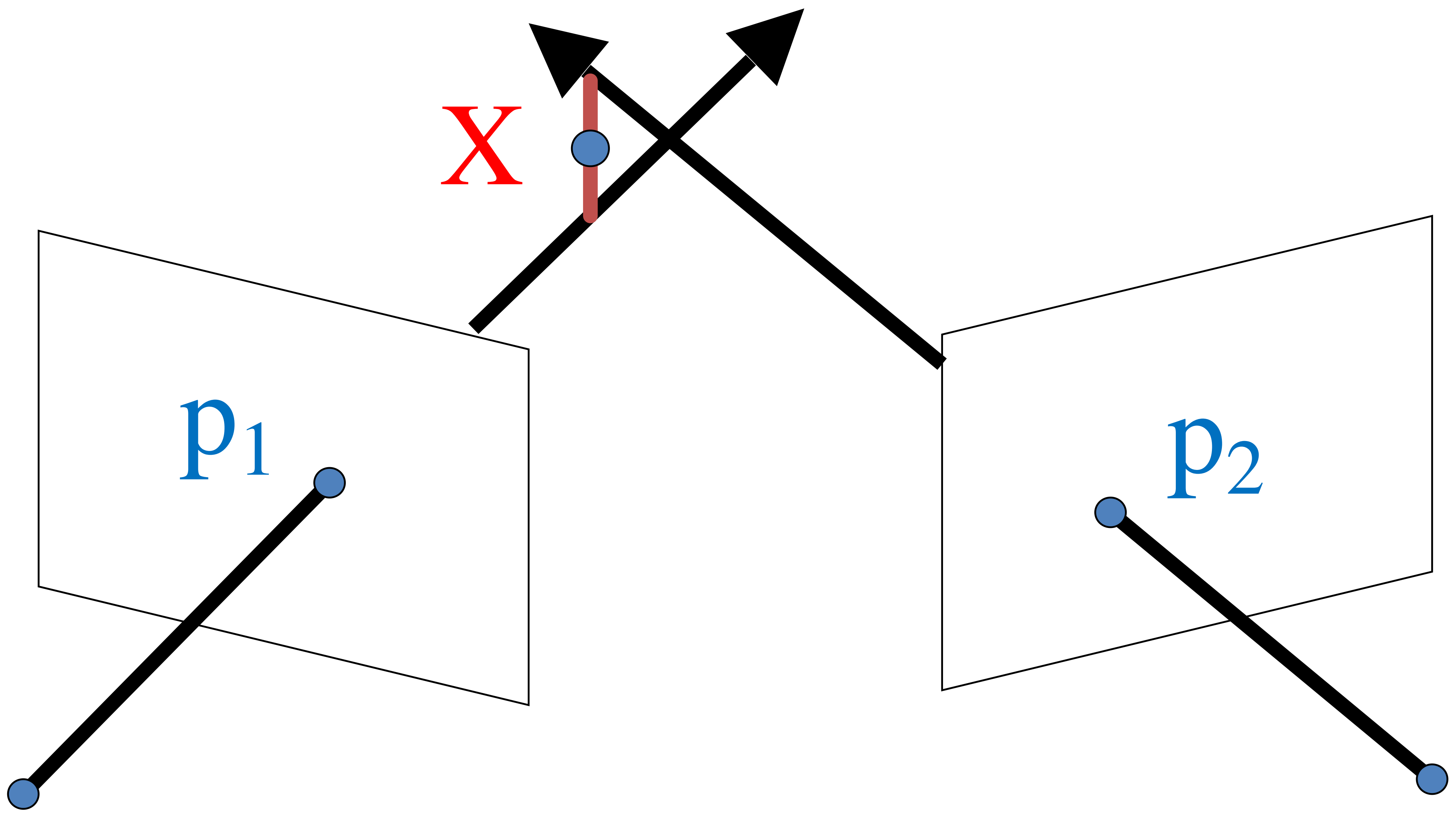
Rays in principle should intersect, but in practice usually don't exactly due to noise, numerical errors.



Source: D. Fouhey

Triangulation: Geometry

Find shortest segment between viewing rays, set **X** to be the midpoint of the segment.



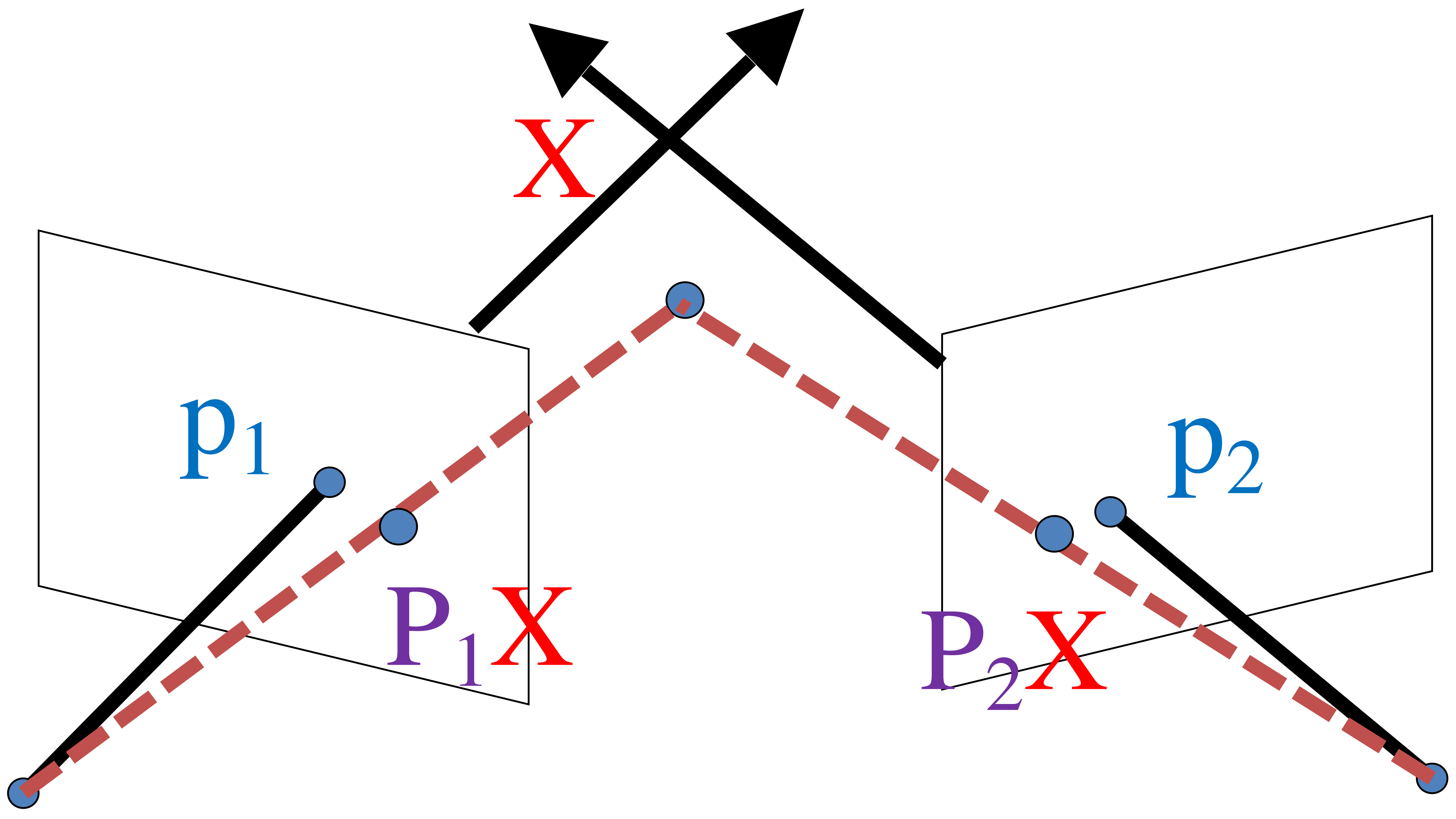
Source: D. Fouhey

Triangulation: Non-linear Optim.

Use nonlinear least squares. Find $\boldsymbol{X}$ that minimizes:

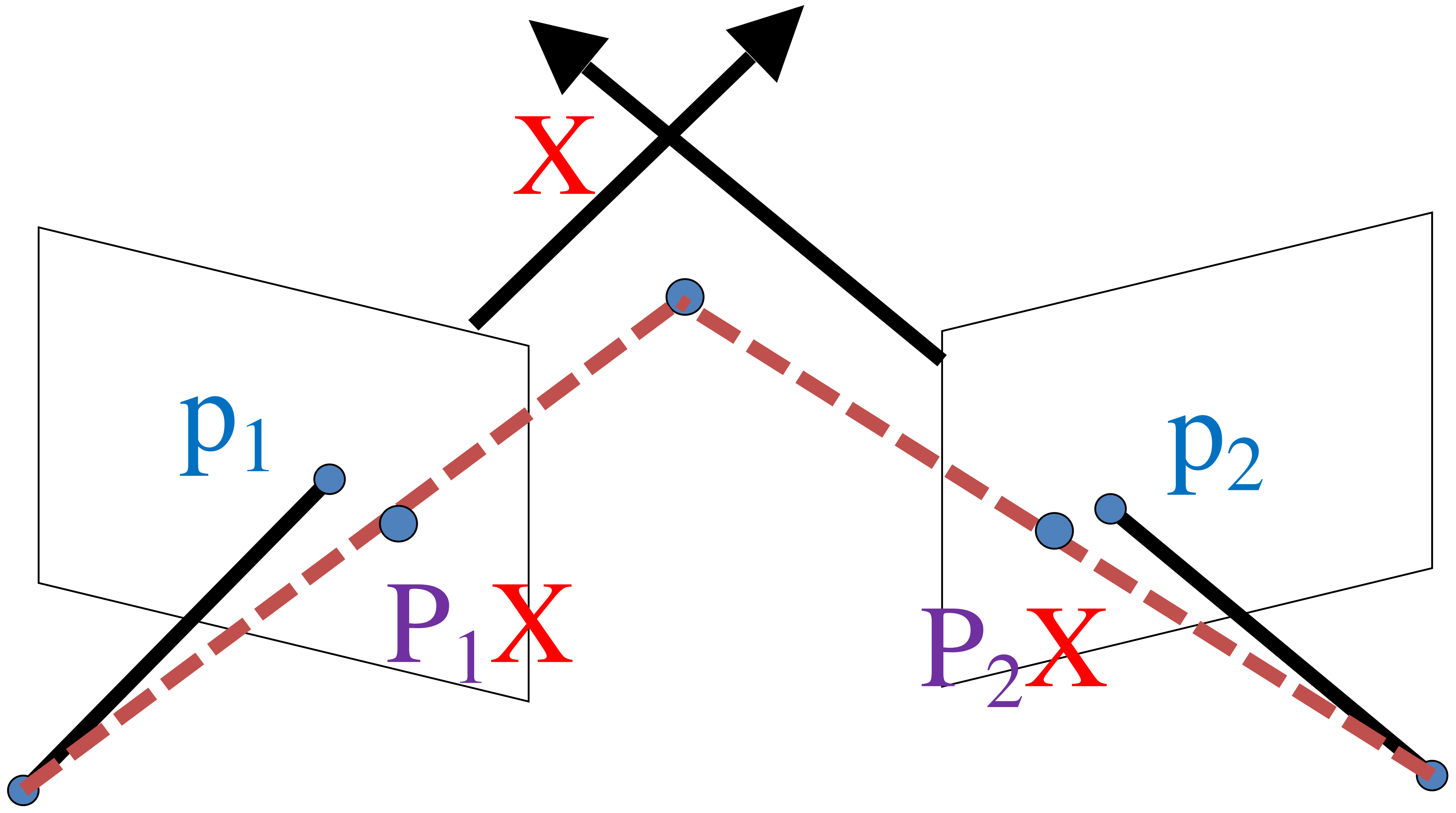
$$d\left(\boldsymbol{p}_1, P_1\boldsymbol{X}\right)^2 + d\left(\boldsymbol{p}_2, P_2\boldsymbol{X}\right)^2$$

where d is distance in image space.



Source: D. Fouhey

Triangulation: Linear method



Source: D. Fouhey

First: A better way to handle homogeneous coordinates in linear optimization

Projection in homogeneous coordinates.

$$\boldsymbol{p}_i \equiv P\boldsymbol{X}_i$$

i.e., **PX_i & p_i** are proportional/scaled copies of each other

$$\boldsymbol{p}_i = \lambda P\boldsymbol{X}_i, \ \ \lambda \neq 0$$

This implies their cross product is **0**, since
$a \times b = \|a\| \|b\| \sin(\theta)$.

$$\boldsymbol{p}_i \times P\boldsymbol{X}_i = \boldsymbol{0}$$

Handles the “divide by 0” issue when solving.

Source: D. Fouhey

Triangulation: Linear method

$$p_1 \equiv P_1 \mathbf{X} \qquad p_1 \times P_1 \mathbf{X} = \mathbf{0} \qquad [p_{1x}] P_1 \mathbf{X} = \mathbf{0}$$

$$p_2 \equiv P_2 \mathbf{X} \qquad p_2 \times P_2 \mathbf{X} = \mathbf{0} \qquad [p_{2x}] P_2 \mathbf{X} = \mathbf{0}$$

Cross product as matrix

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = [\boldsymbol{a}_x] \boldsymbol{b}$$

$$[p_{1x}] P_1 \mathbf{X} = \mathbf{0} \qquad ([p_{1x}] P_1) \mathbf{X} = \mathbf{0}$$

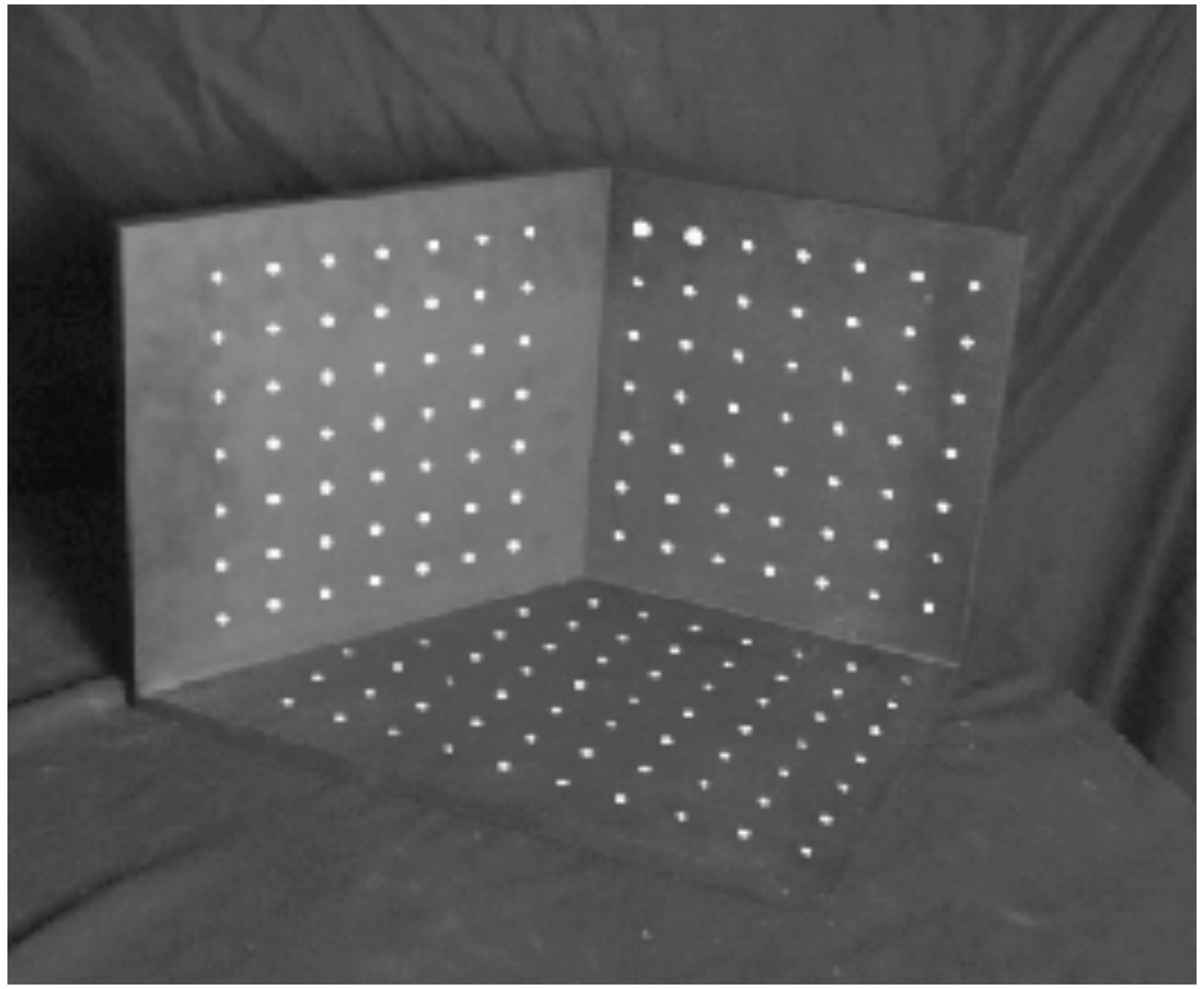
$$[p_{2x}] P_2 \mathbf{X} = \mathbf{0} \qquad ([p_{2x}] P_2) \mathbf{X} = \mathbf{0}$$

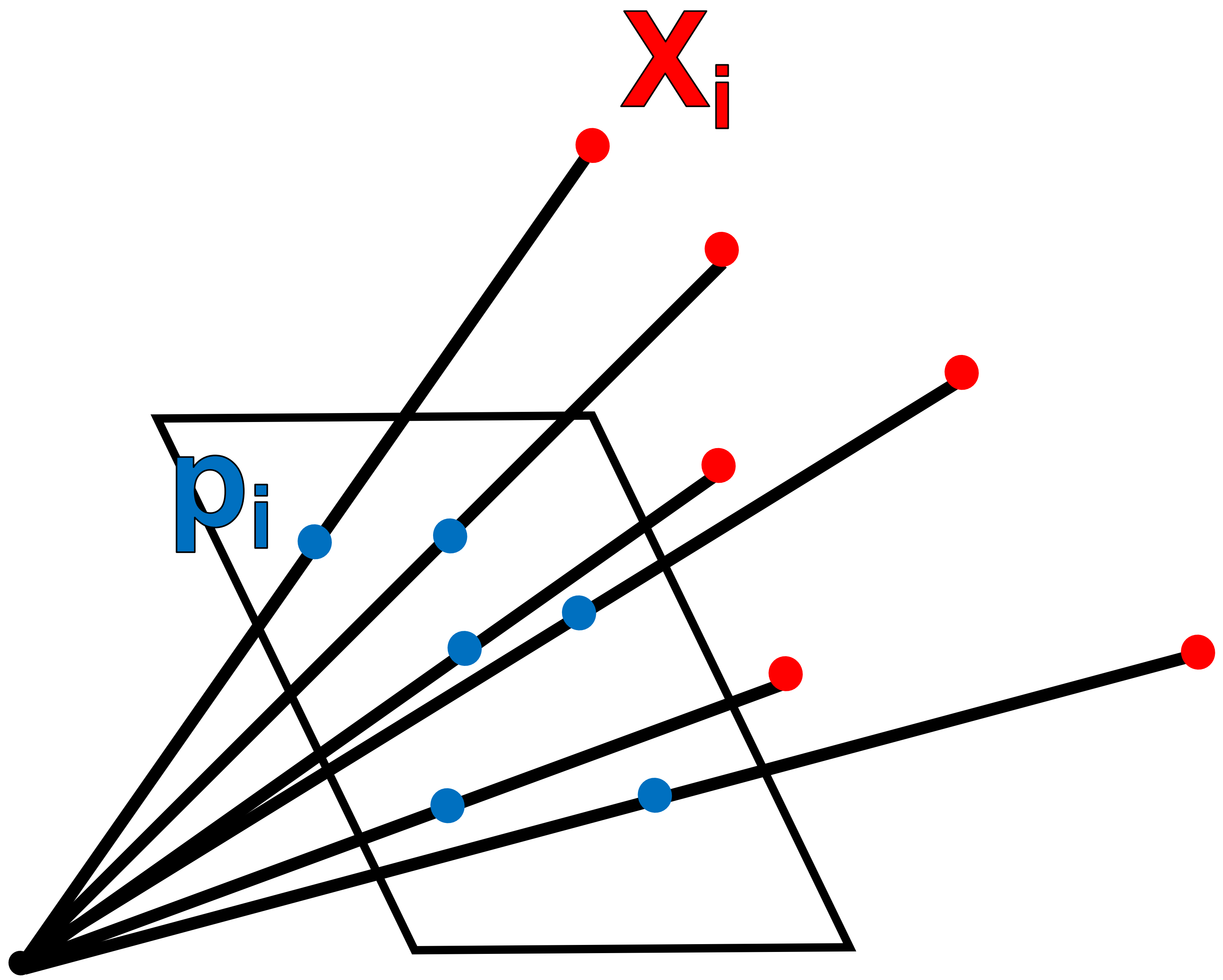
Two equations per camera for 3 unknown in X

Source: D. Fouhey

Camera calibration

- Can we estimate camera pose from points? Known as *camera calibration* or *camera resectioning*.
- Given *n* points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\mathbf{p}_i$, estimate the camera parameters.





Source: S. Lazebnik and D. Fouhey

Camera Calibration: Linear Method

$$\boldsymbol{p}_i \equiv P\boldsymbol{X}_i$$

Remember (from geometry): this implies $\mathbf{MX_i}$ **&** $\mathbf{p_i}$ are proportional/scaled copies of each other

$$\boldsymbol{p}_i = \lambda P\boldsymbol{X}_i, \ \lambda \neq 0$$

Recall that this implies their cross product is **0**

$$\boldsymbol{p}_i \times P\boldsymbol{X}_i = \mathbf{0}$$

Source: D. Fouhey

Camera Calibration: Linear Method

$$\mathbf{p}_i \times P\mathbf{X}_i = \mathbf{0}$$

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1 P_i \\ P_2 \mathbf{X}_i \\ P_3 \mathbf{X}_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Using these constraints, you can derive:

$$\begin{bmatrix} \mathbf{0}^T & -\mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ \mathbf{X}_i^T & \mathbf{0}^T & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where P_i is row i of P

Adapted from D. Fouhey

Camera calibration

- The linear solution does not optimize the right objective function.

- Optimize using nonlinear least squares:

$$\sum \left\| \mathrm{proj}(P\mathbf{X}_i) - [u_i, v_i]^T \right\|_2^2$$

- Can initialize using the linear solution.
- Other advantages: can also add radial distortion, not optimize over known variables, add constraints

Source: D. Fouhey

Structure from motion

- We can estimate points from cameras and cameras from points. Can we do both at once?
- Given many images, how can we…
    1. Figure out where they were all taken from?
    2. Build a 3D model of the scene?

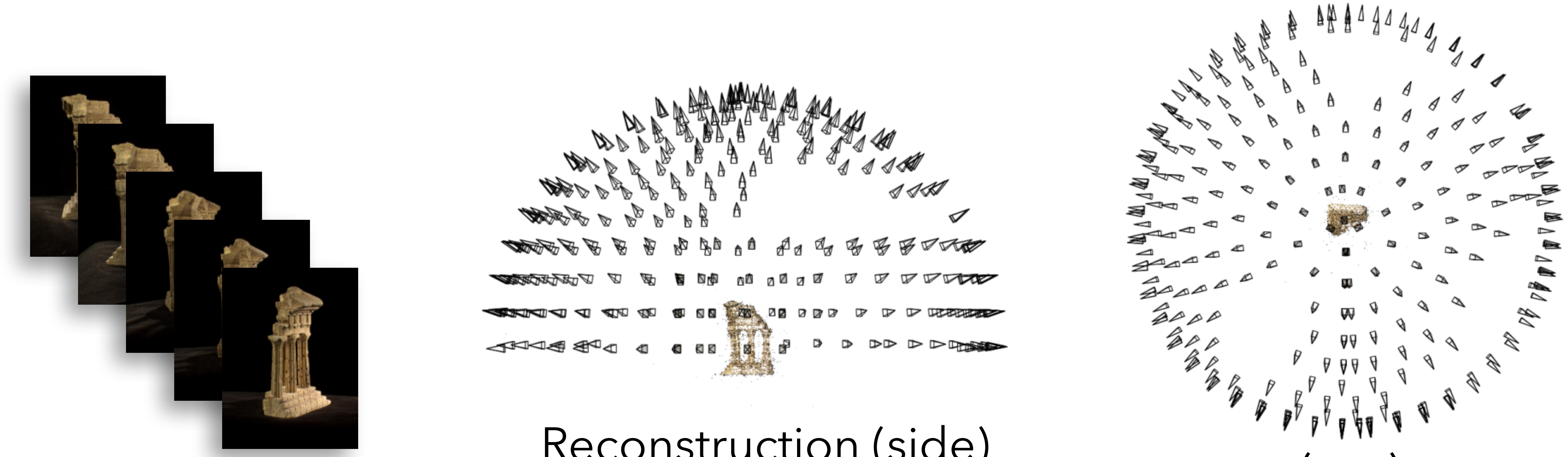
This is the **structure from motion** problem.

Adapted from: N. Snavely

Today

- Structure from motion
- Multi-view stereo
- Radiance fields

Structure from motion



Reconstruction (side)

(top)

- Input: images with pixels 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

Source: N. Snavely

Camera calibration & triangulation

- Suppose we know 3D points
  - And have matches between these points and an image
  - Computing camera parameters similar to homography estimation

- Suppose we have know camera parameters, each of which observes a point
  - We can solve for the 3D location

- Seems like a chicken-and-egg problem, but in SfM we can solve both at once

Source: N. Snavely

Example: Photo Tourism

15,464

37,383

76,389

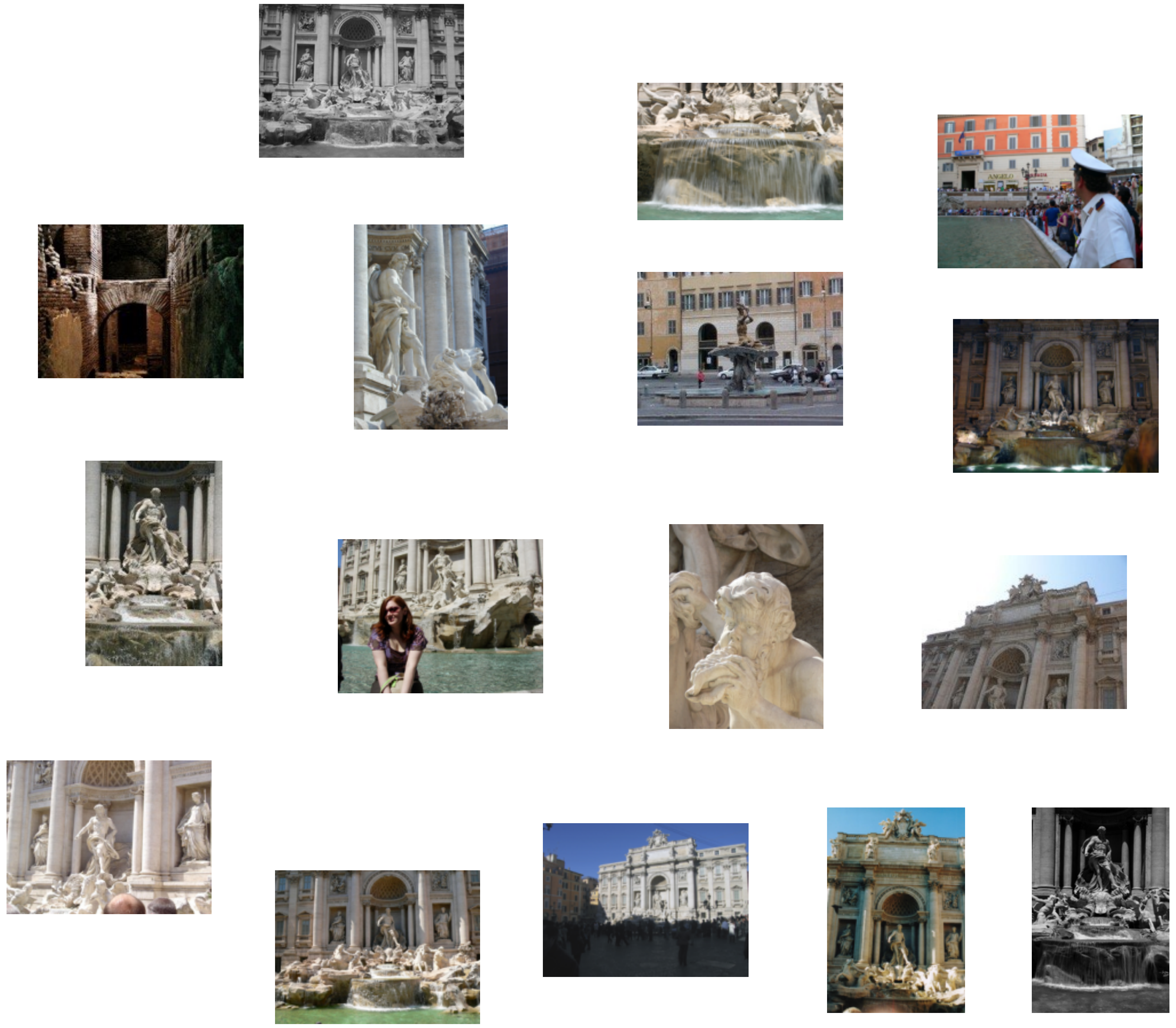
Source: N. Snavely

Example: Photo Tourism

Source: N. Snavely

Feature detection

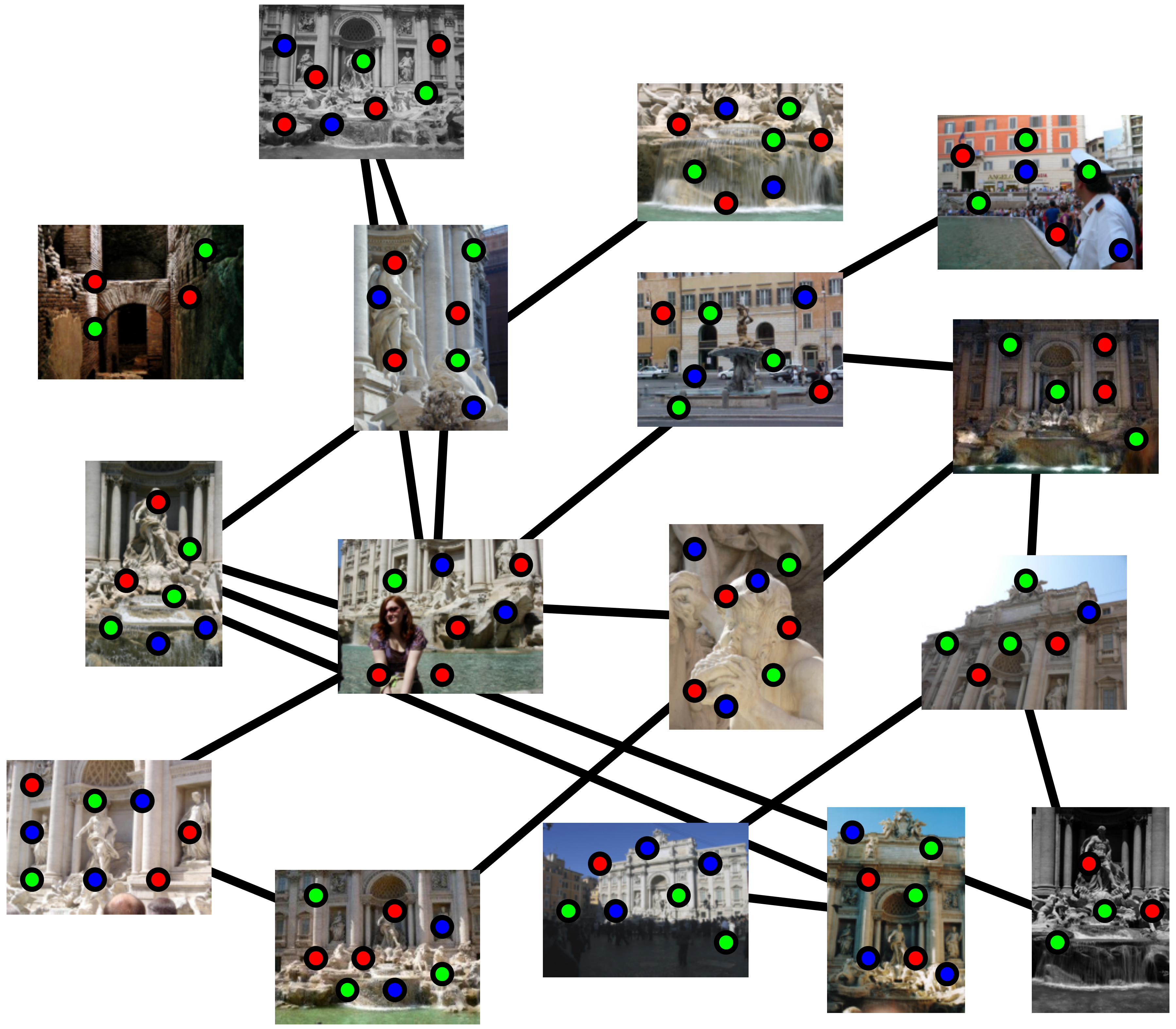
- Same process as with homography estimation
- Detect features using SIFT



Source: N. Snavely

Feature matching

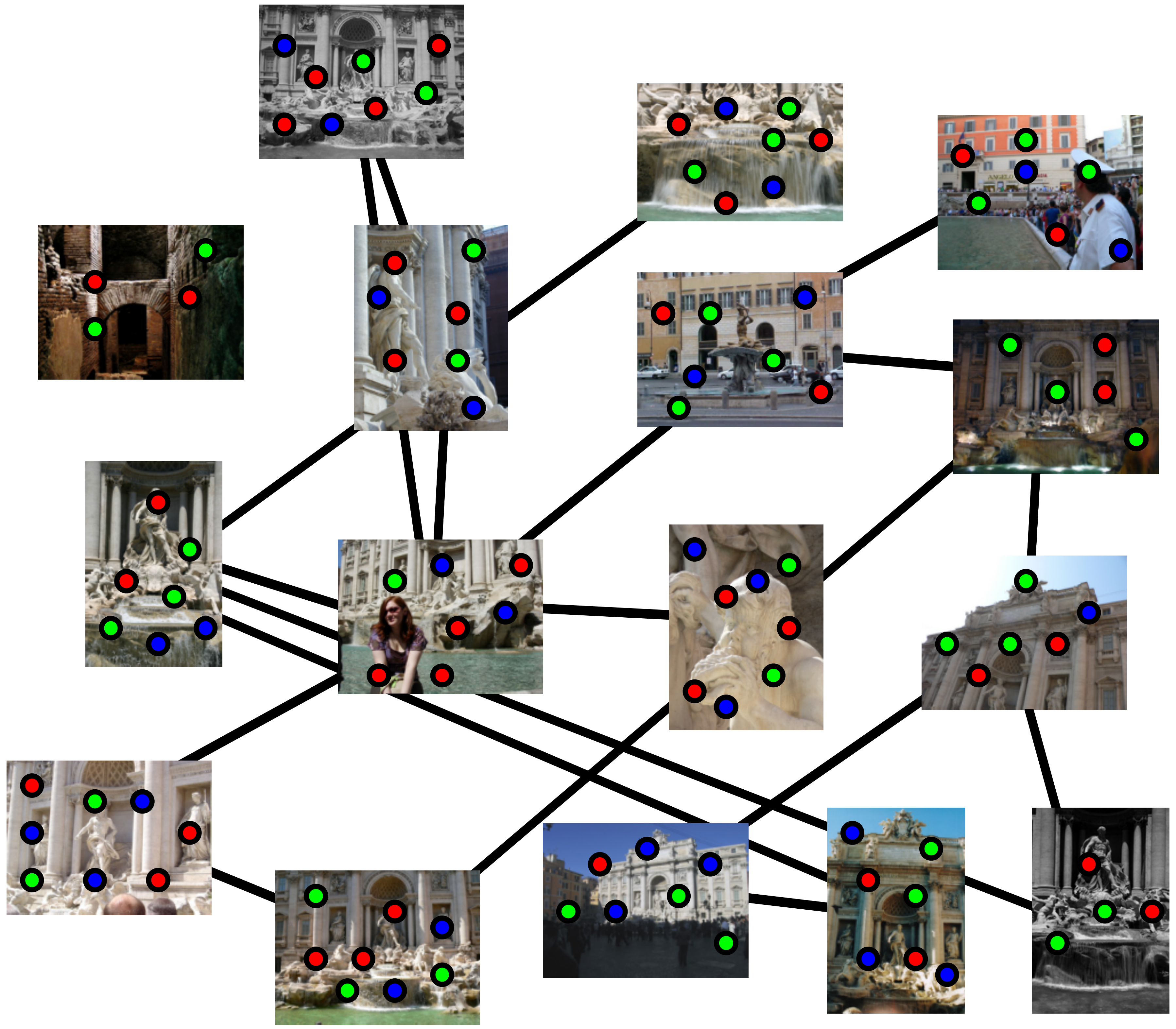
Match features between each pair of images



Source: N. Snavely

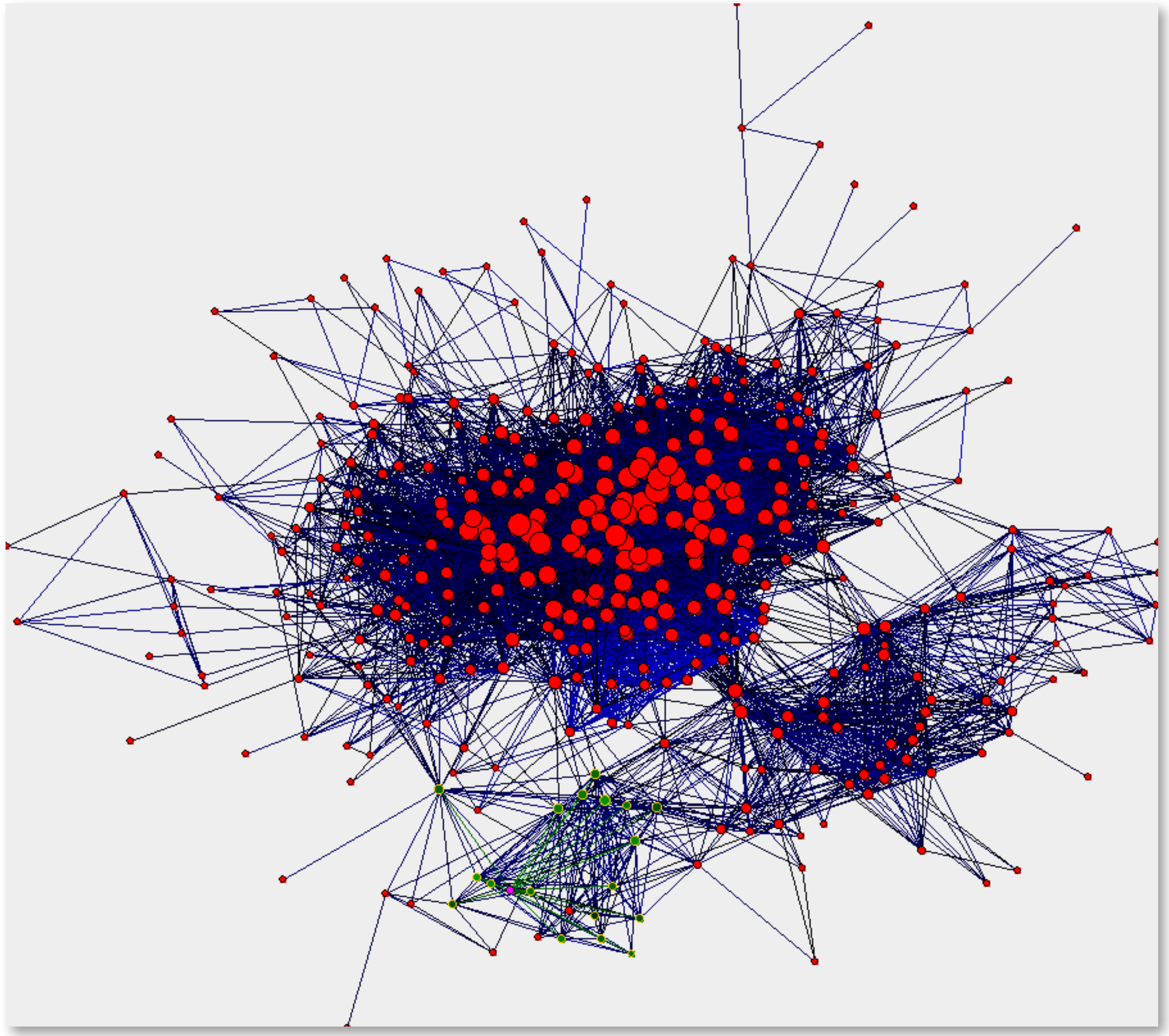
Feature matching

- Remove bad matches using ratio test.
- Other tricks: throw out matches that aren't on epipolar lines.



Source: N. Snavely

Image connectivity graph



Source: N. Snavely

Correspondence estimation

- Track each feature across the dataset.
- Link up pairwise matches to form connected components of matches across several images.

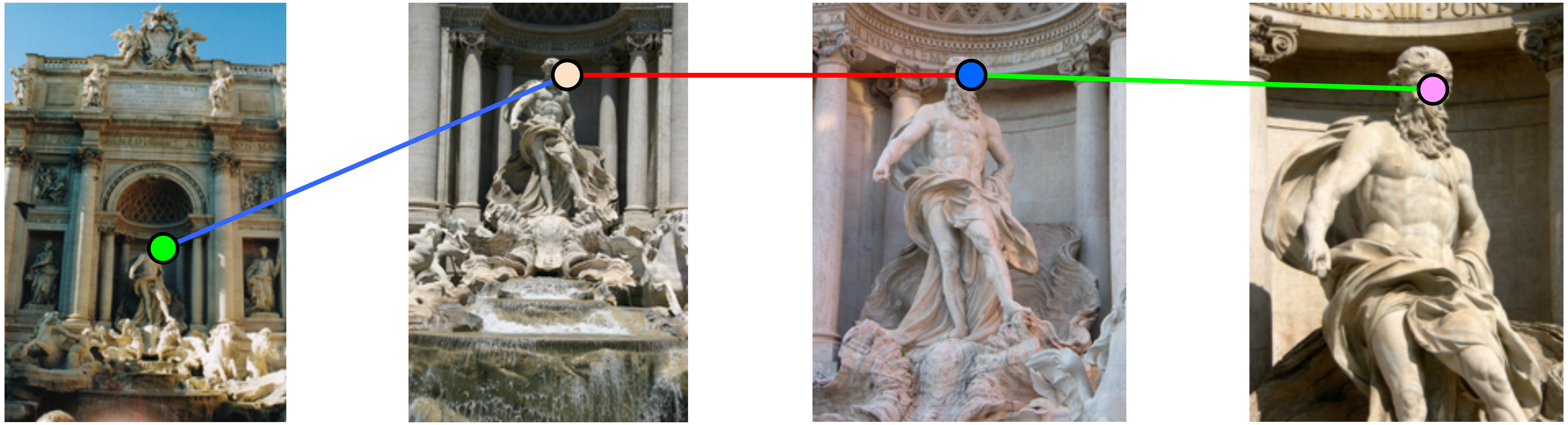


Image 1 Image 2 Image 3 Image 4

Source: N. Snavely

Correspondence estimation

A **point track**: the same 3D point projects to all 4 image positions.

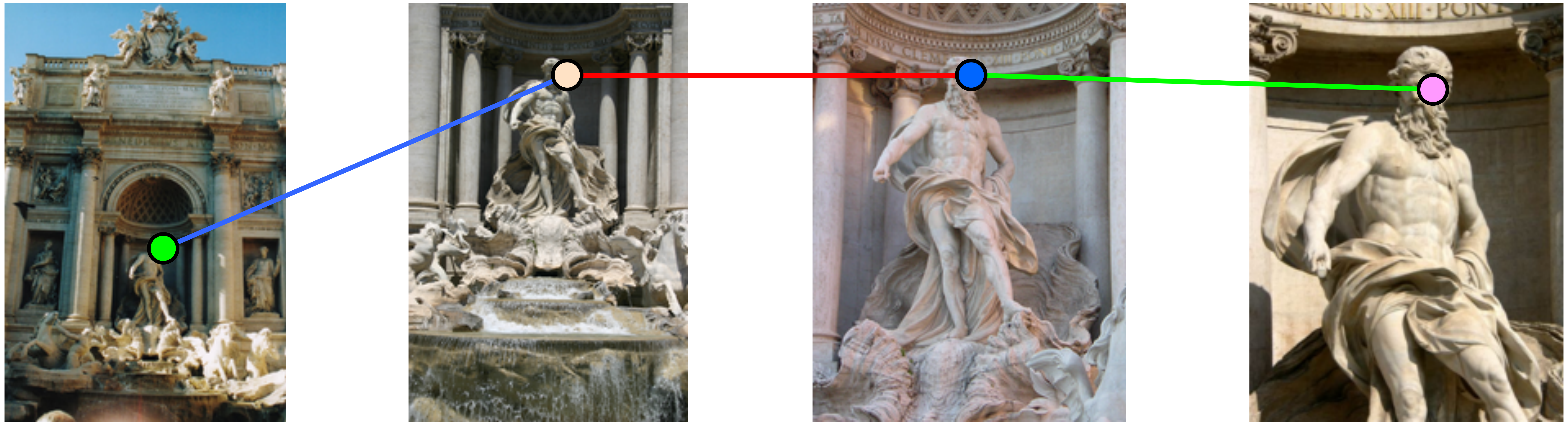
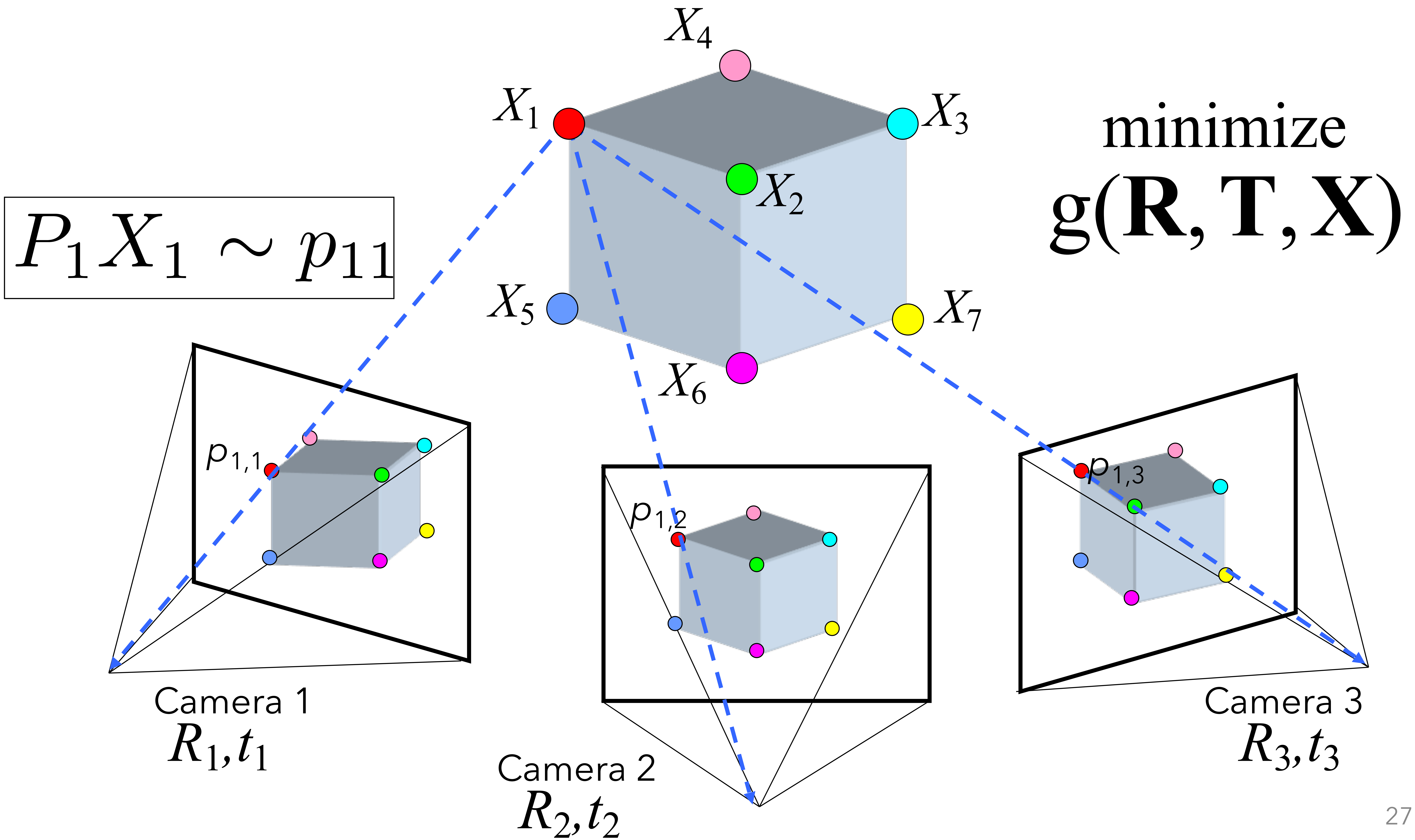


Image 1 Image 2 Image 3 Image 4

Structure from motion



Source: N. Snavely

Structure from motion

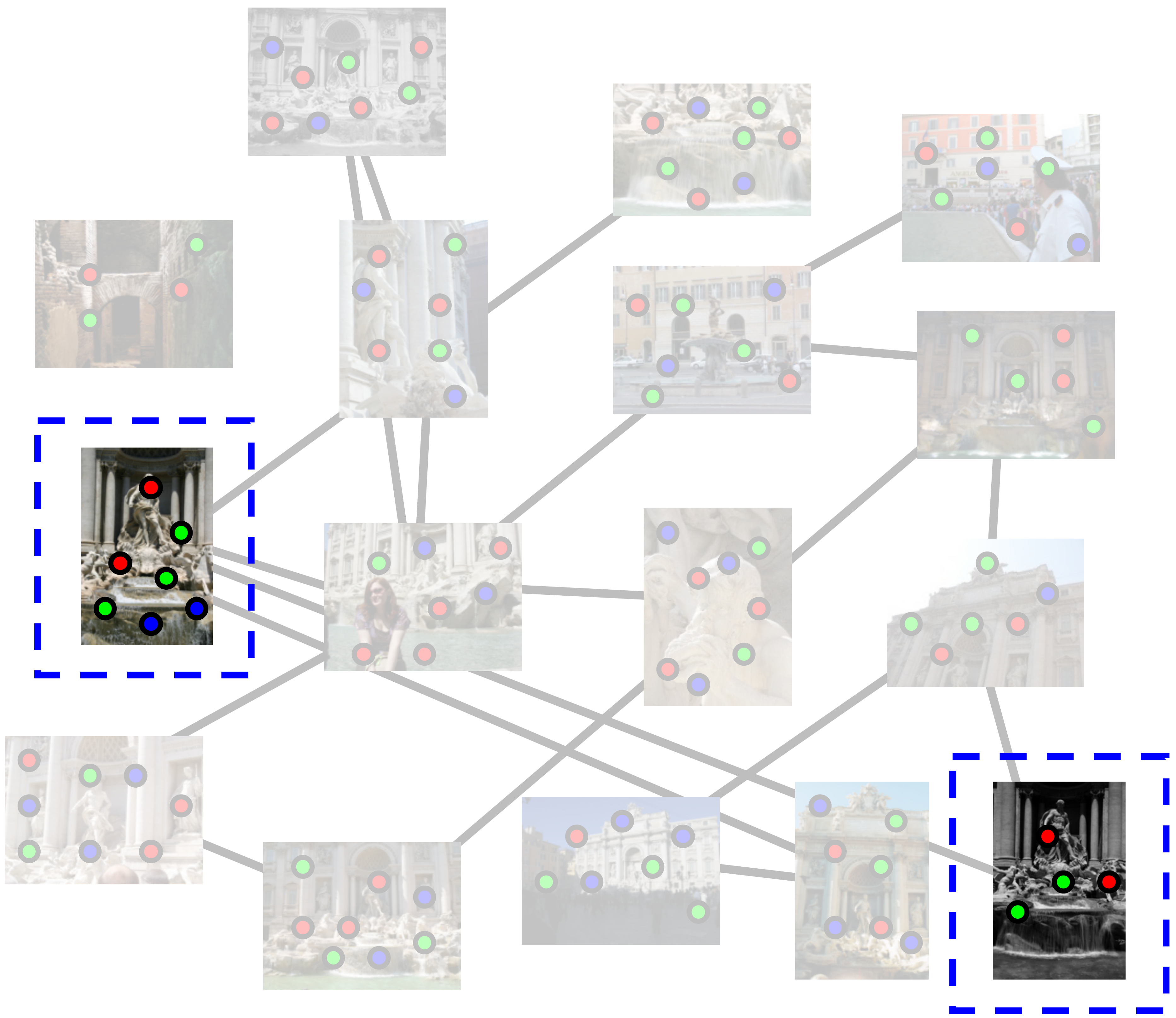
- Minimize sum of squared reprojection errors:

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

- Minimizing this function is called *bundle adjustment*.
  - Optimized using non-linear least squares
- Lots of outliers: use robust loss functions (e.g., Huber) and solve incrementally

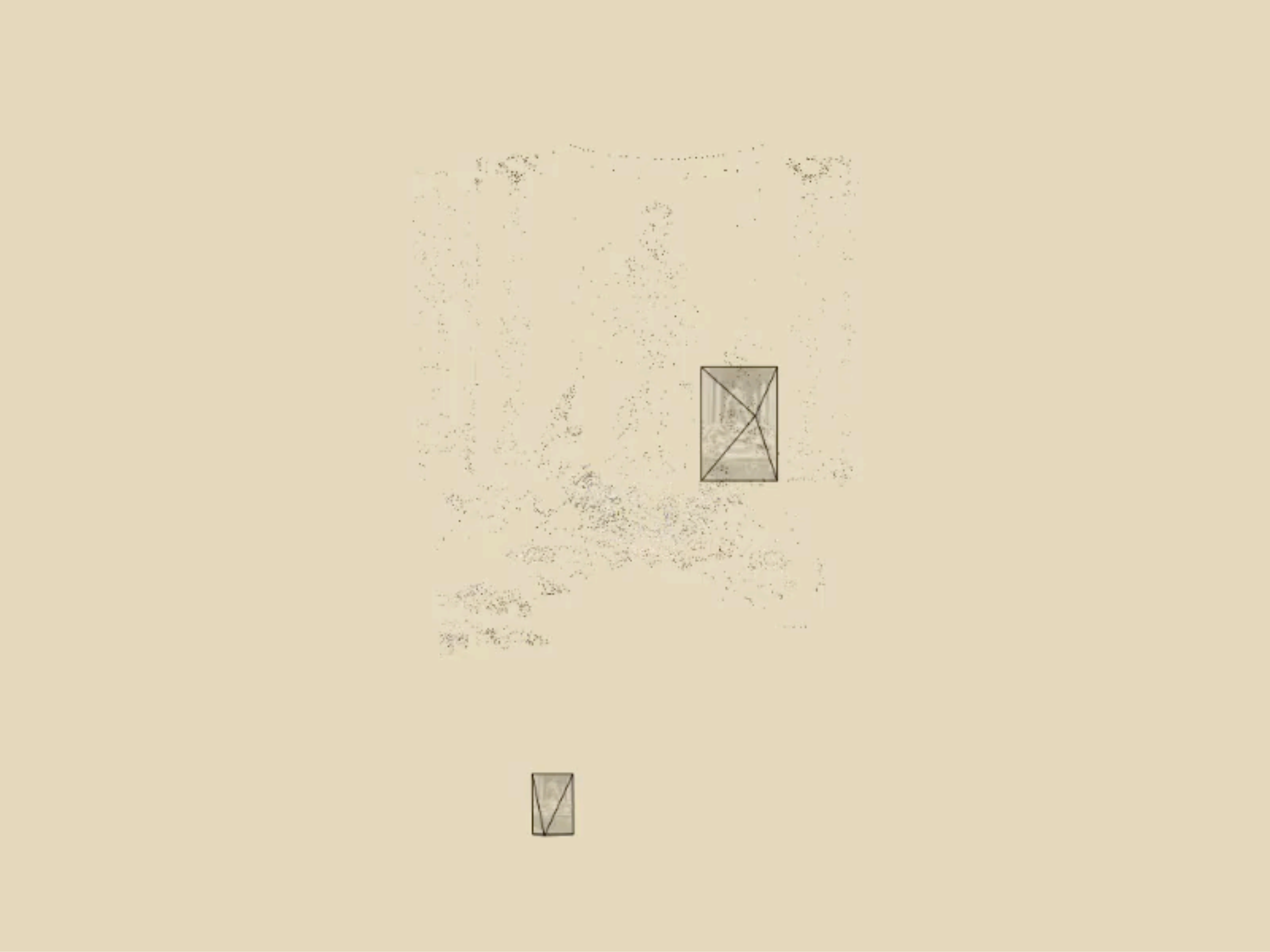
Source: N. Snavely

Incremental structure from motion



Source: N. Snavely

Incremental structure from motion



Source: N. Snavely

Incremental structure from motion

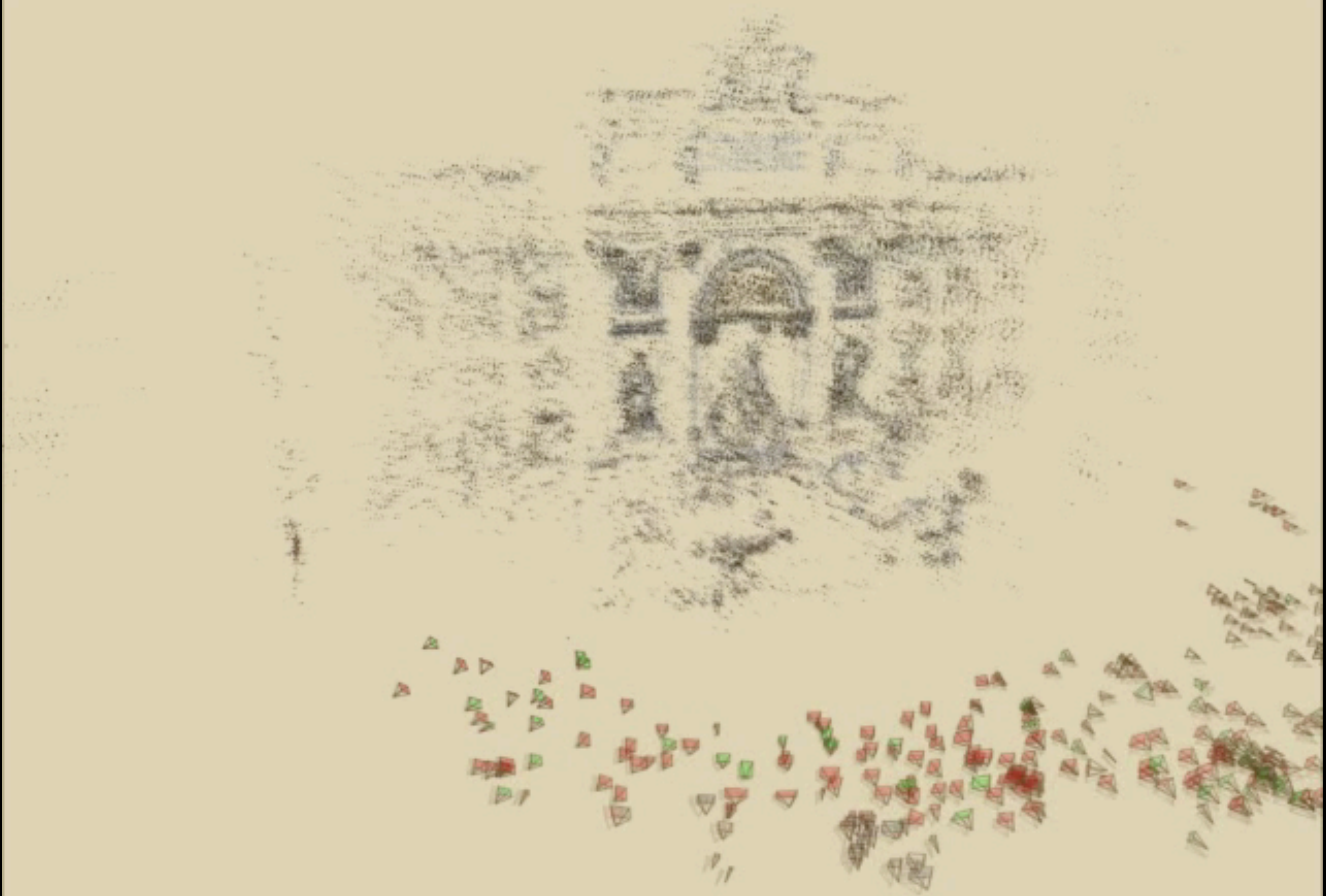
Source: N. Snavely

Photo Tourism

Exploring photo collections in 3D

Noah Snavely Steven M. Seitz Richard Szeliski
University of Washington *Microsoft Research*

SIGGRAPH 2006



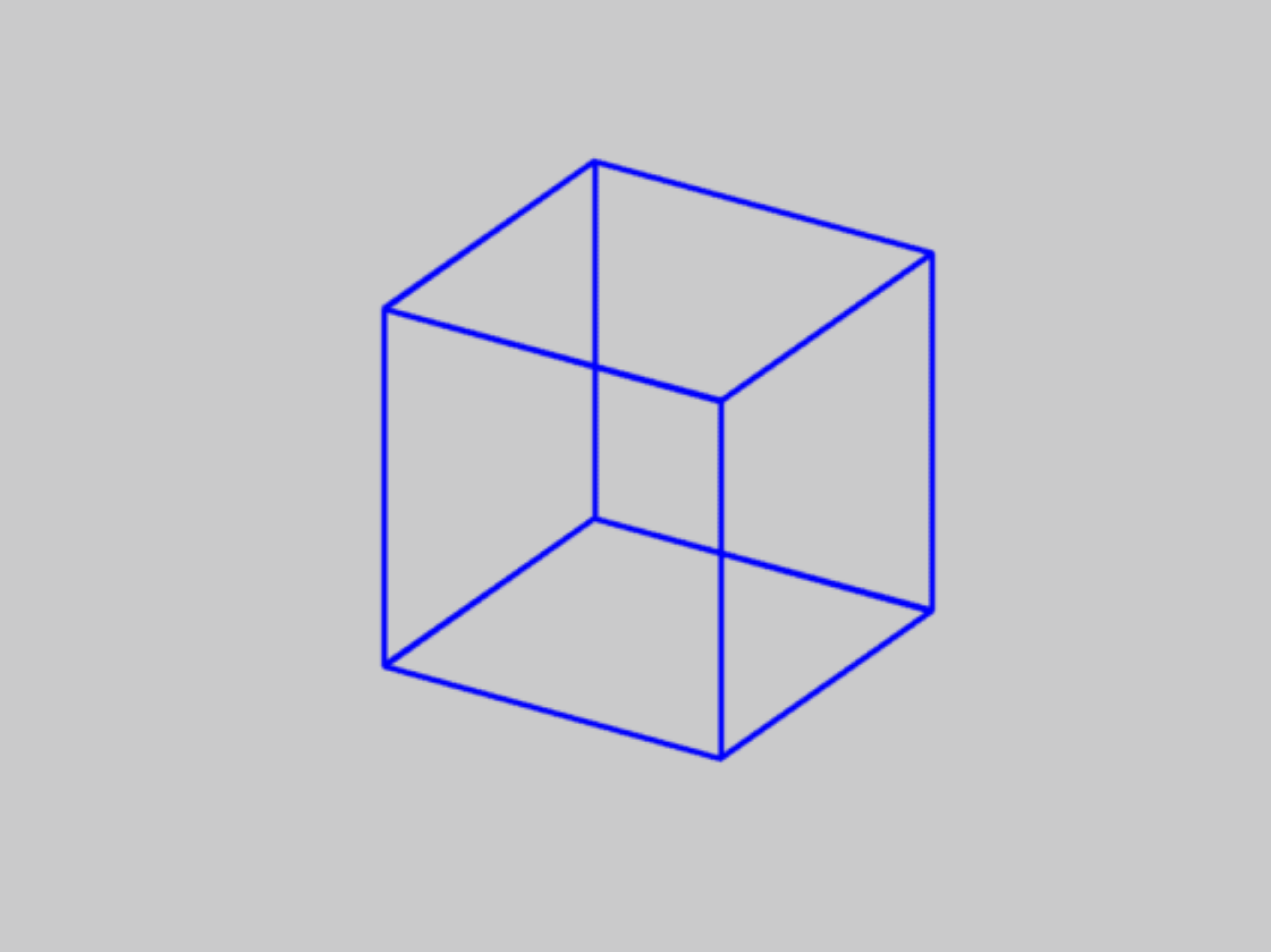

Name: brodo_1608736
Added by: brodo
Date: November 21, 20...

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

Source: N. Snavely

Is SfM always uniquely solvable?

No. Consider the Necker cube:



Source: N. Snavely

Is SfM always uniquely solvable?

Two interpretations:

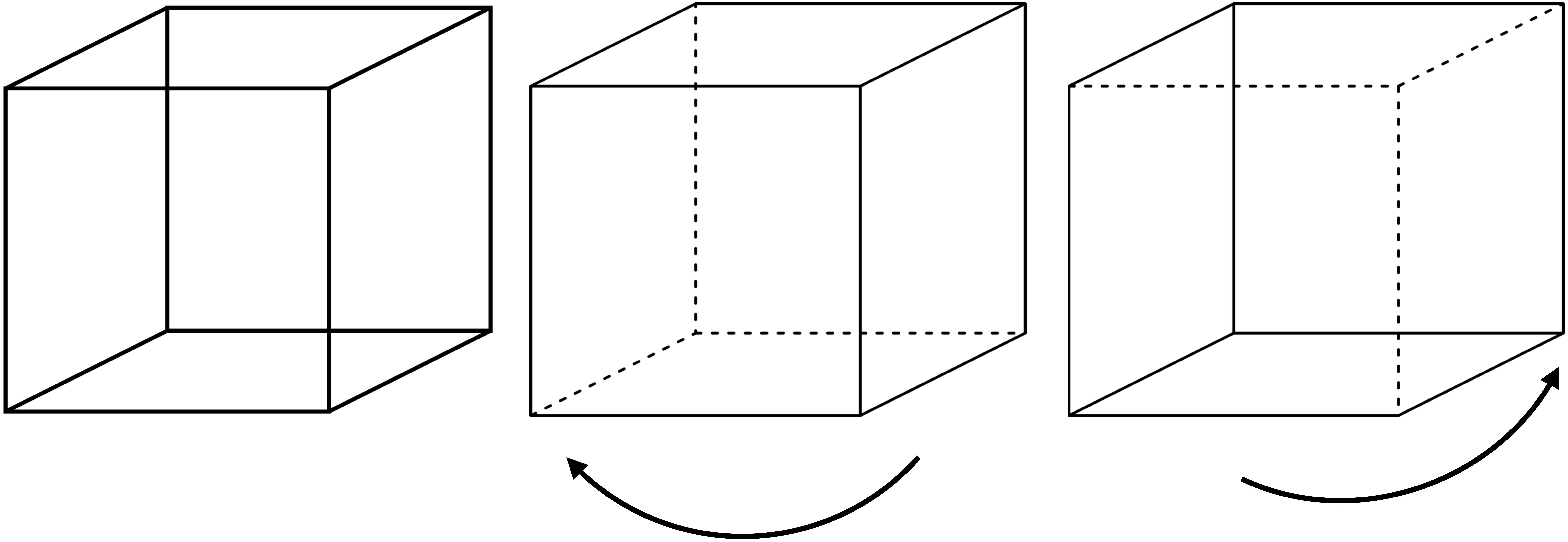
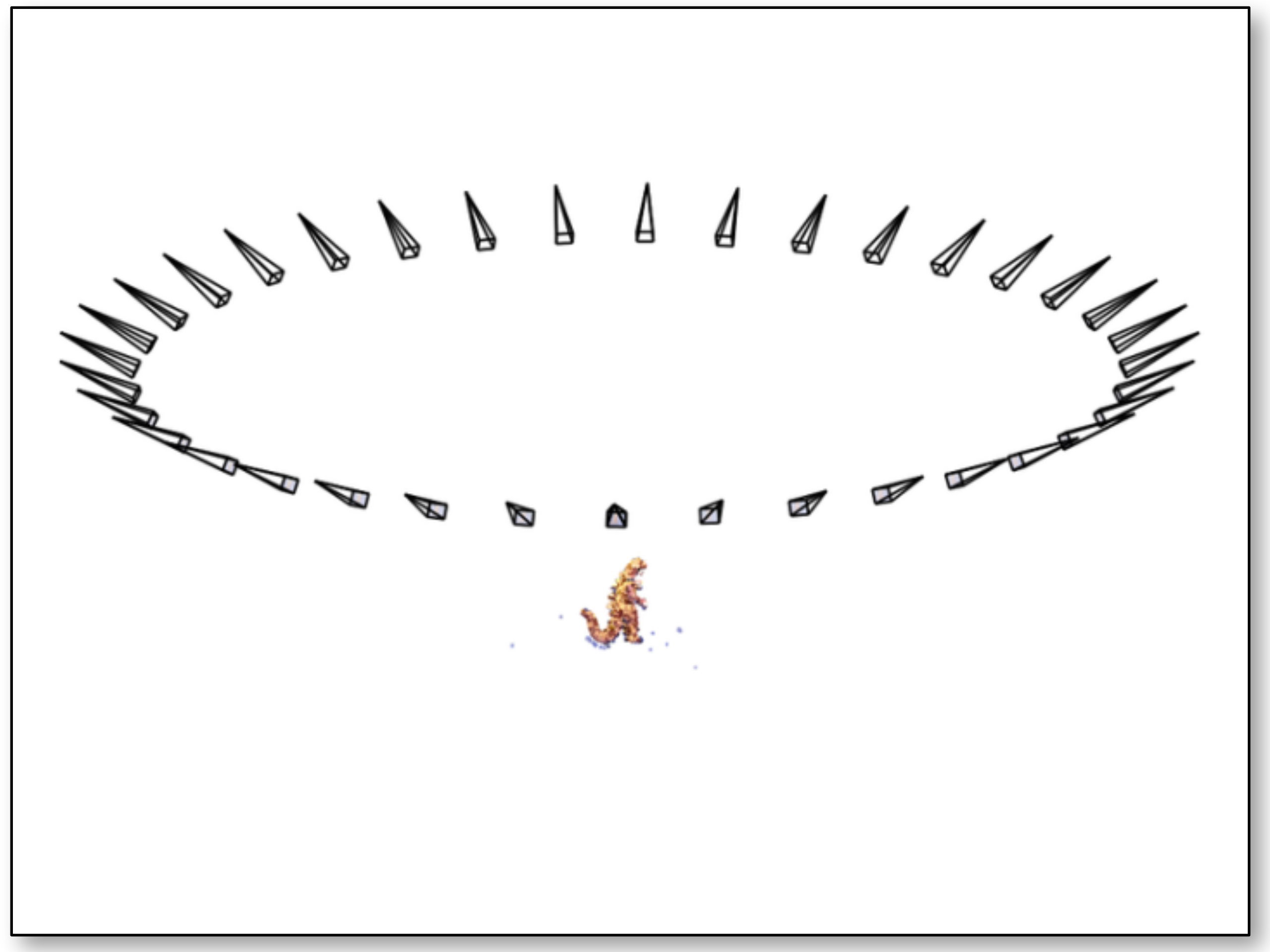
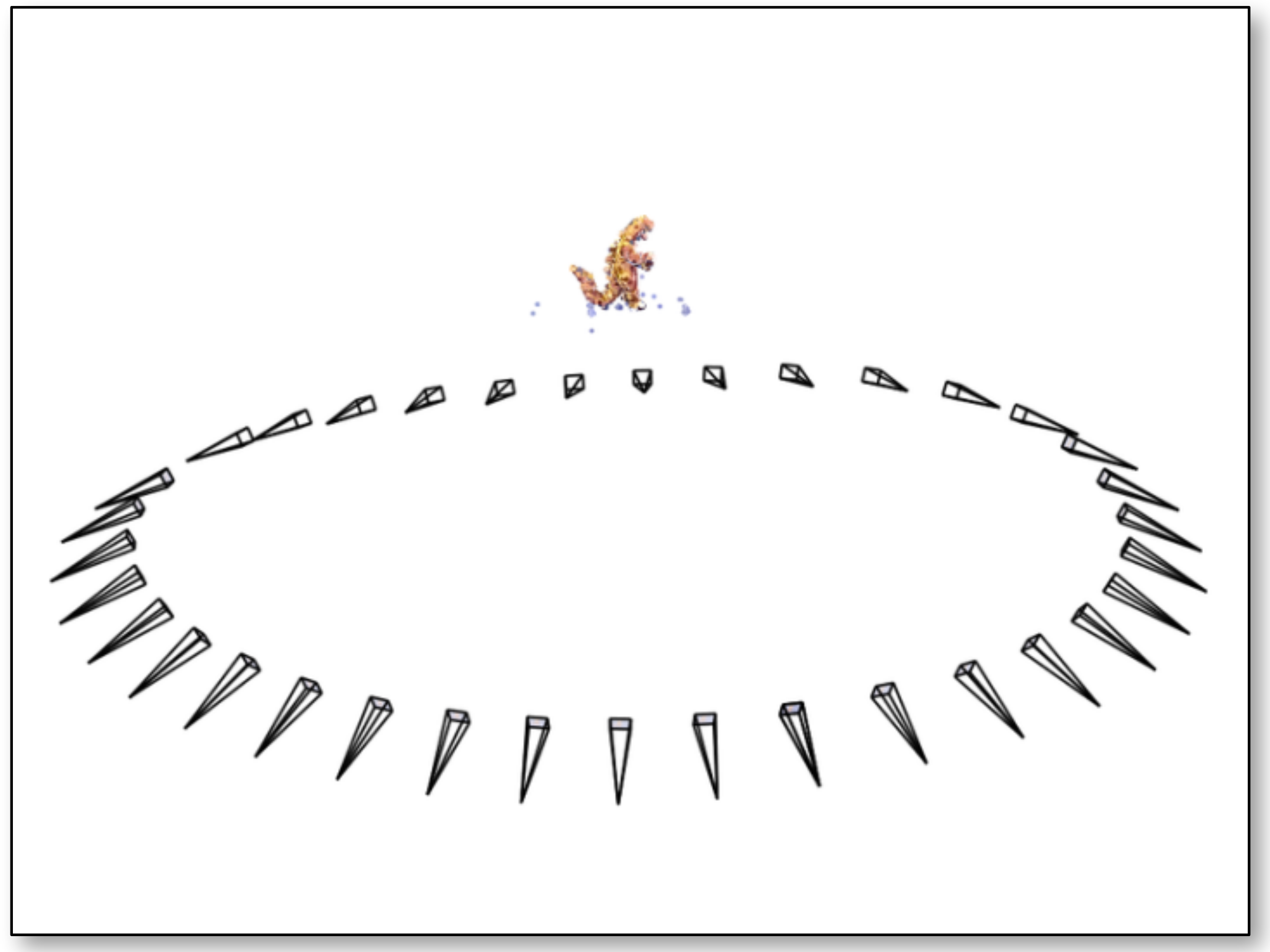


Image source: Wikipedia

Failure case: Necker reversal

- Under orthographic camera, object rotation by θ produces same image as mirror image rotated by $-\theta$.
- Can occur in perspective cameras, e.g., when objects are far from the camera





Adapted from N. Snavely

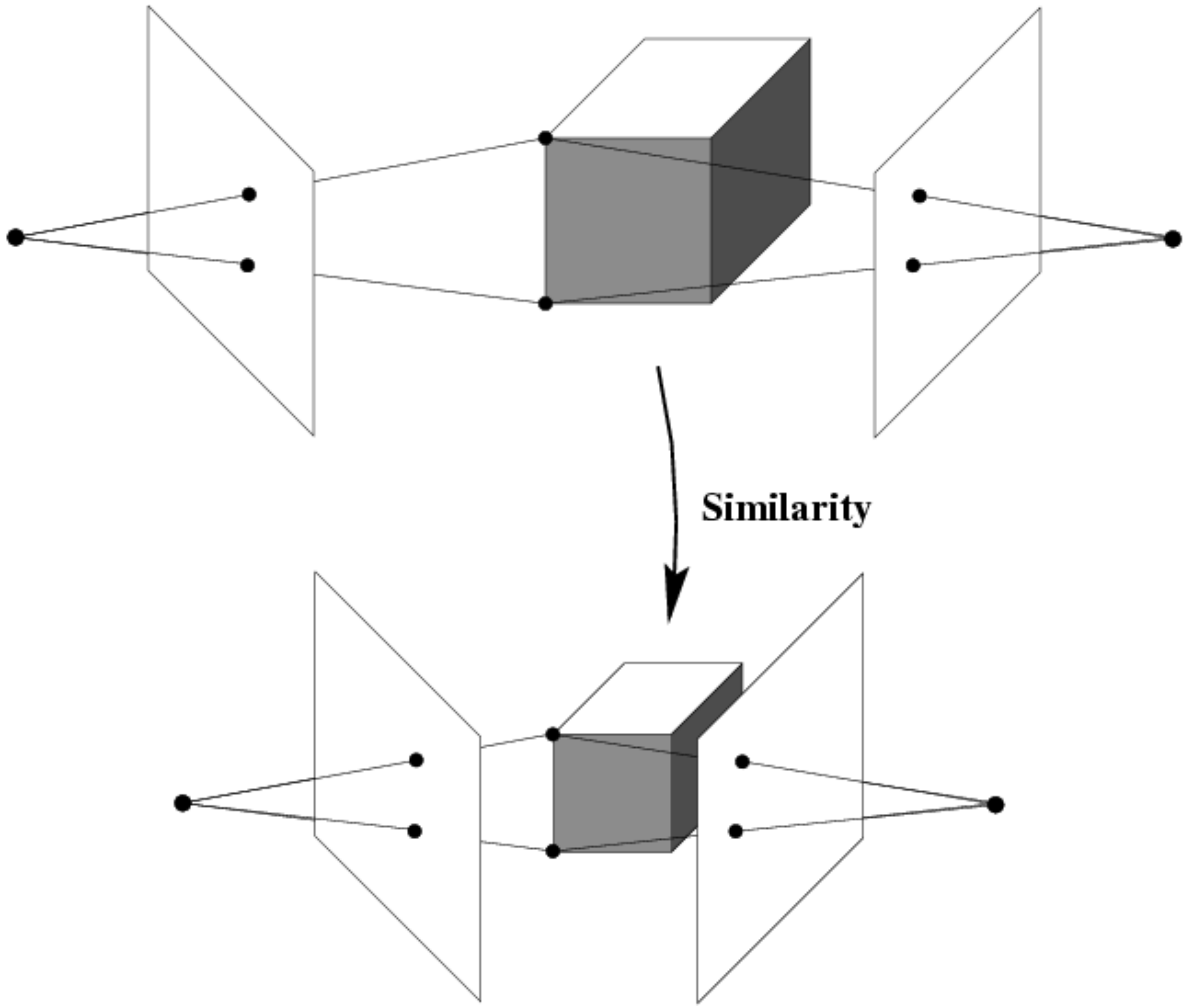
Ambiguity up to similarity transformation

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

3×3
rotation
matrix

3×1
translation
vector

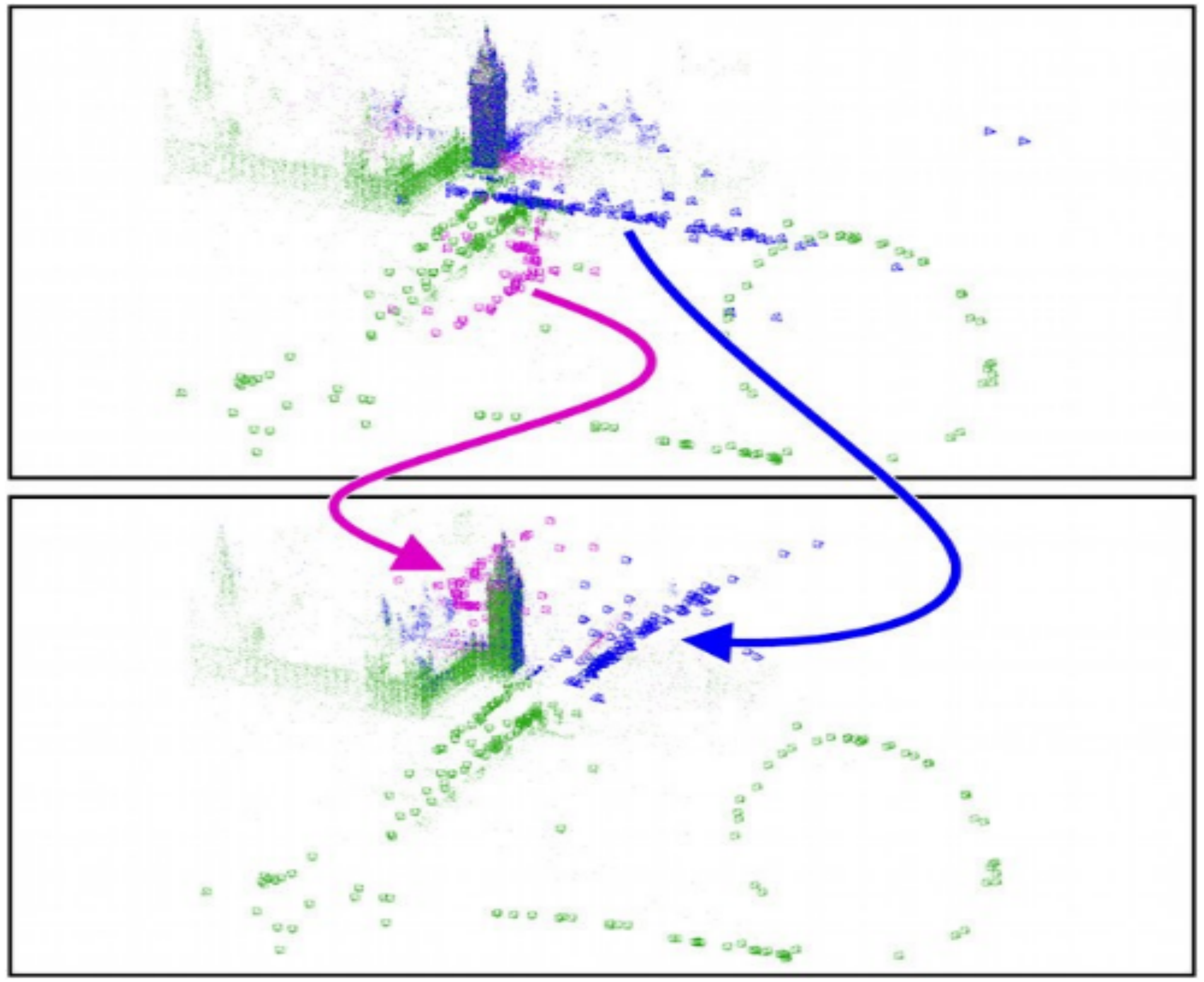
$$\boldsymbol{Q}_S = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



- Special case: scale ambiguity

Adapted from S. Lazebnik

Repetitive structures cause catastrophic failures



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

Source: S. Lazebnik

Repetitive structures cause catastrophic failures

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

Source: S. Lazebnik

Can also reconstruct from video

Source: N. Snavely

Applications: Visual Reality & Augmented Reality

Oculus

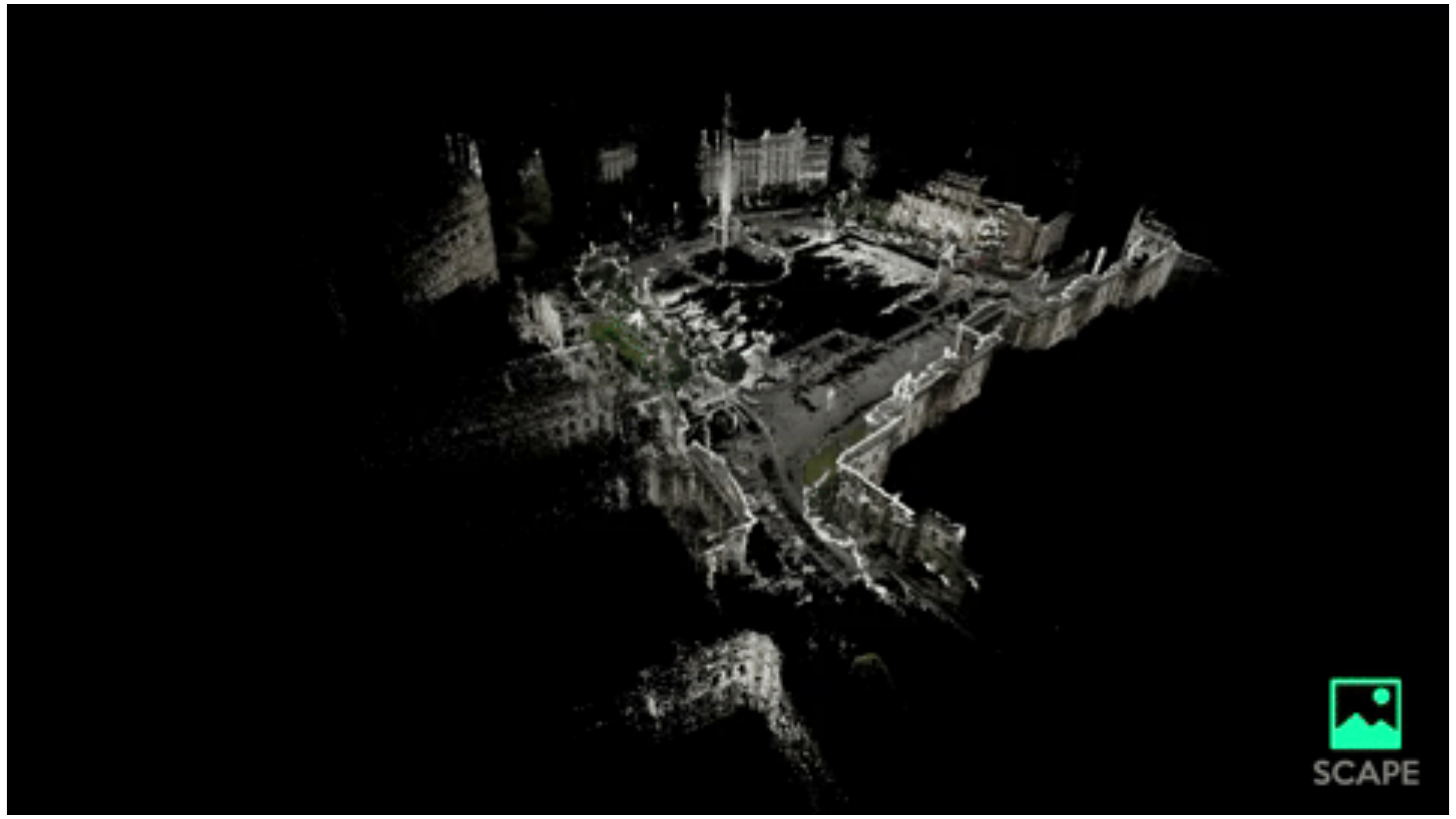
https://www.youtube.com/watch?v=KOG7yTz1iTA

Hololens

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

Source: N. Snavely

Application: Simultaneous localization and mapping (SLAM)



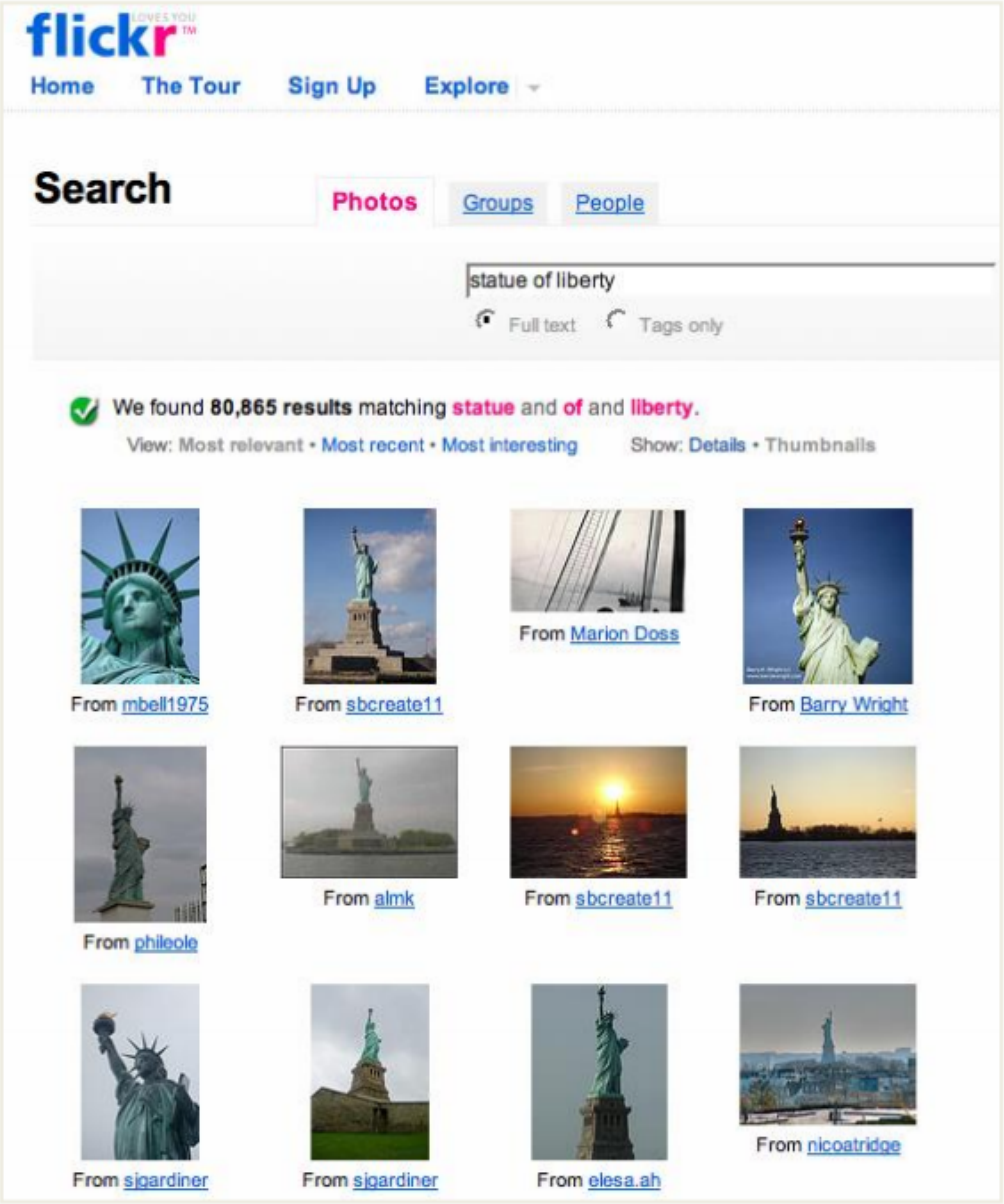
Scape: Building the 'AR Cloud': Part Three –3D Maps, the Digital Scaffolding of the 21st Century
https://medium.com/scape-technologies/building-the-ar-cloud-part-three-3d-maps-the-digital-scaffolding-of-the-21st-century-465fa55782dd

Source: N. Snavely

Today

- Structure from motion
- **Multi-view stereo**
- Stereo matching algorithms

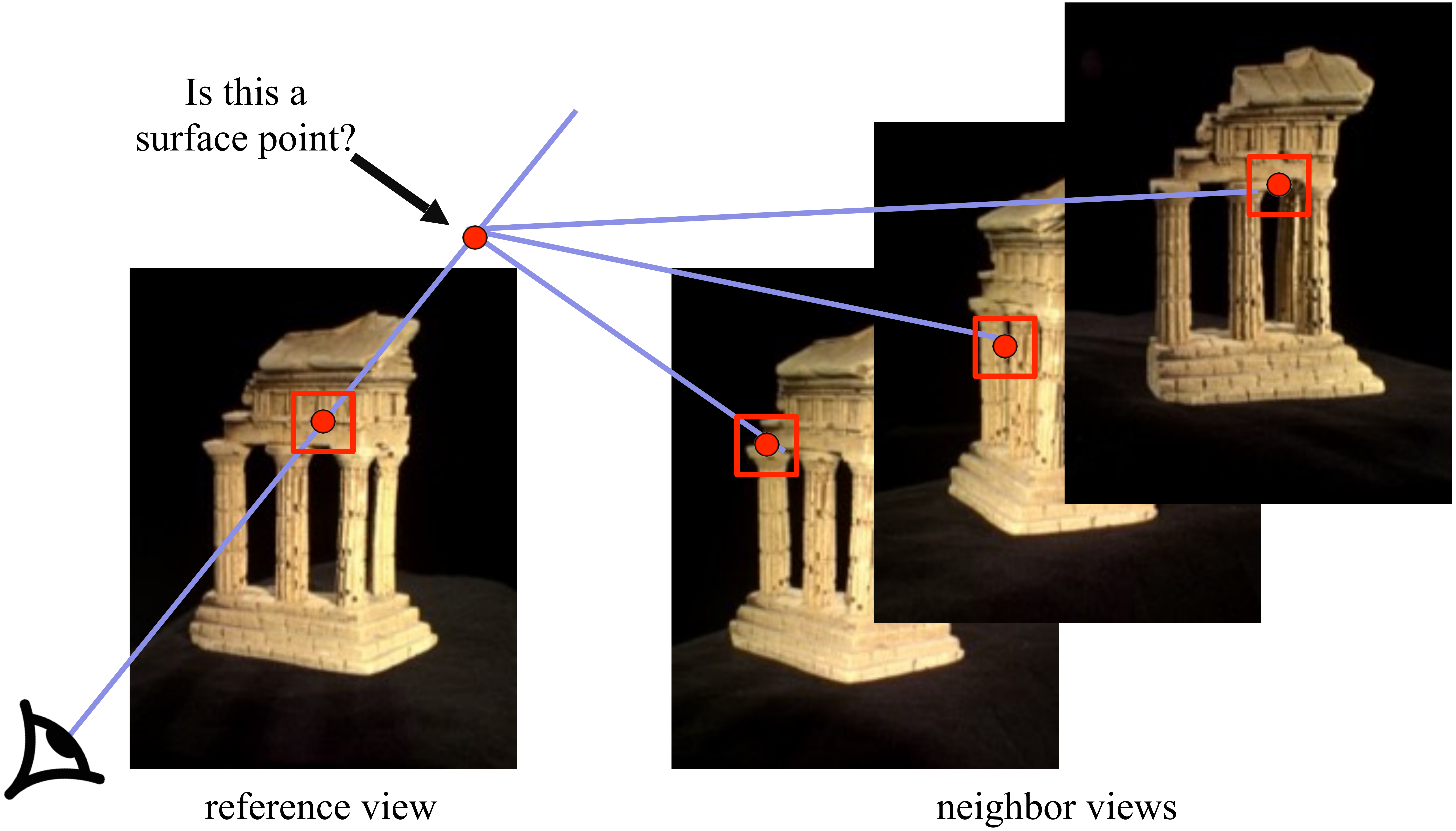
Multi-view stereo



Can we estimate depth, now that we have pose?

Source: N. Snavely

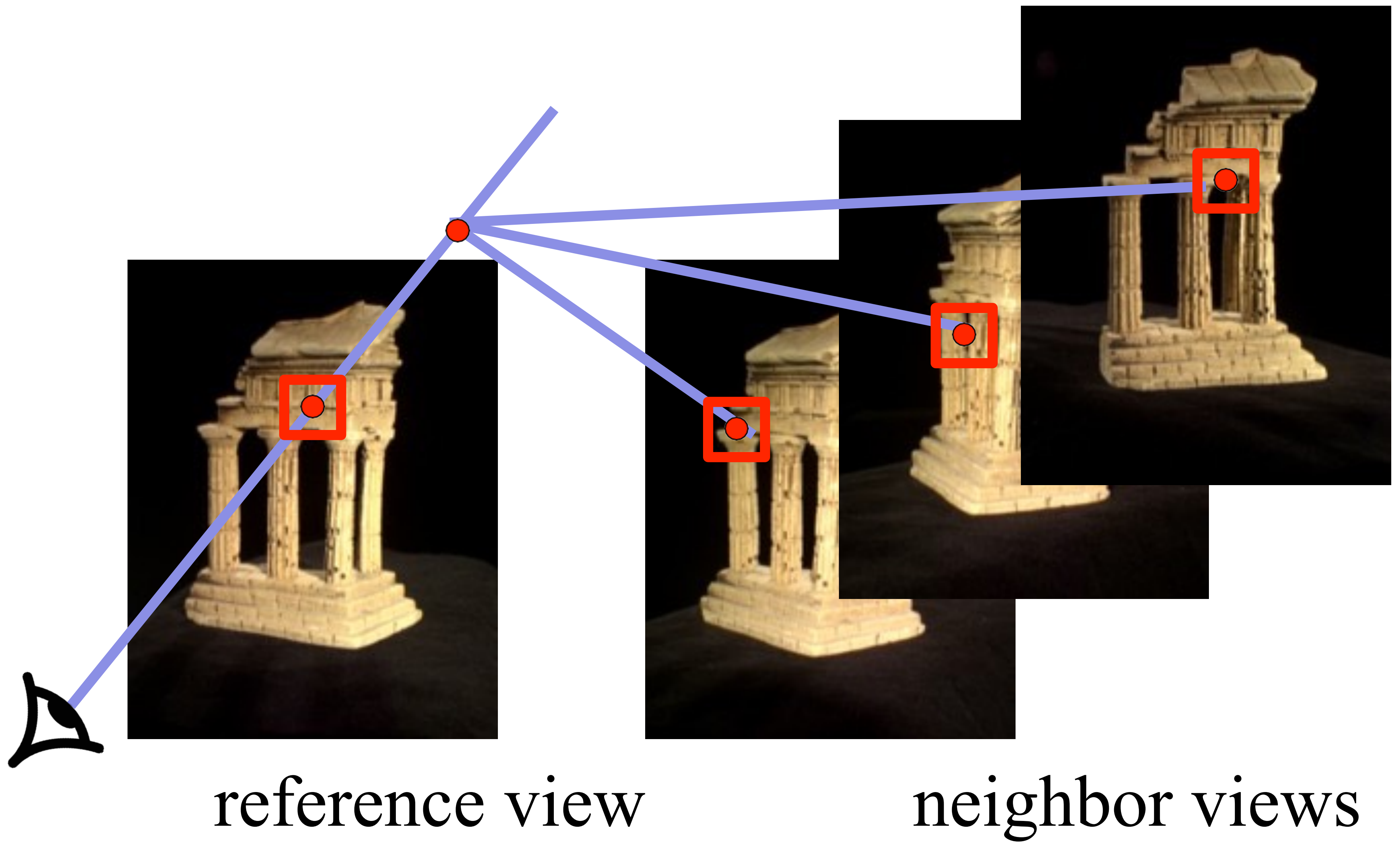
Multi-view stereo

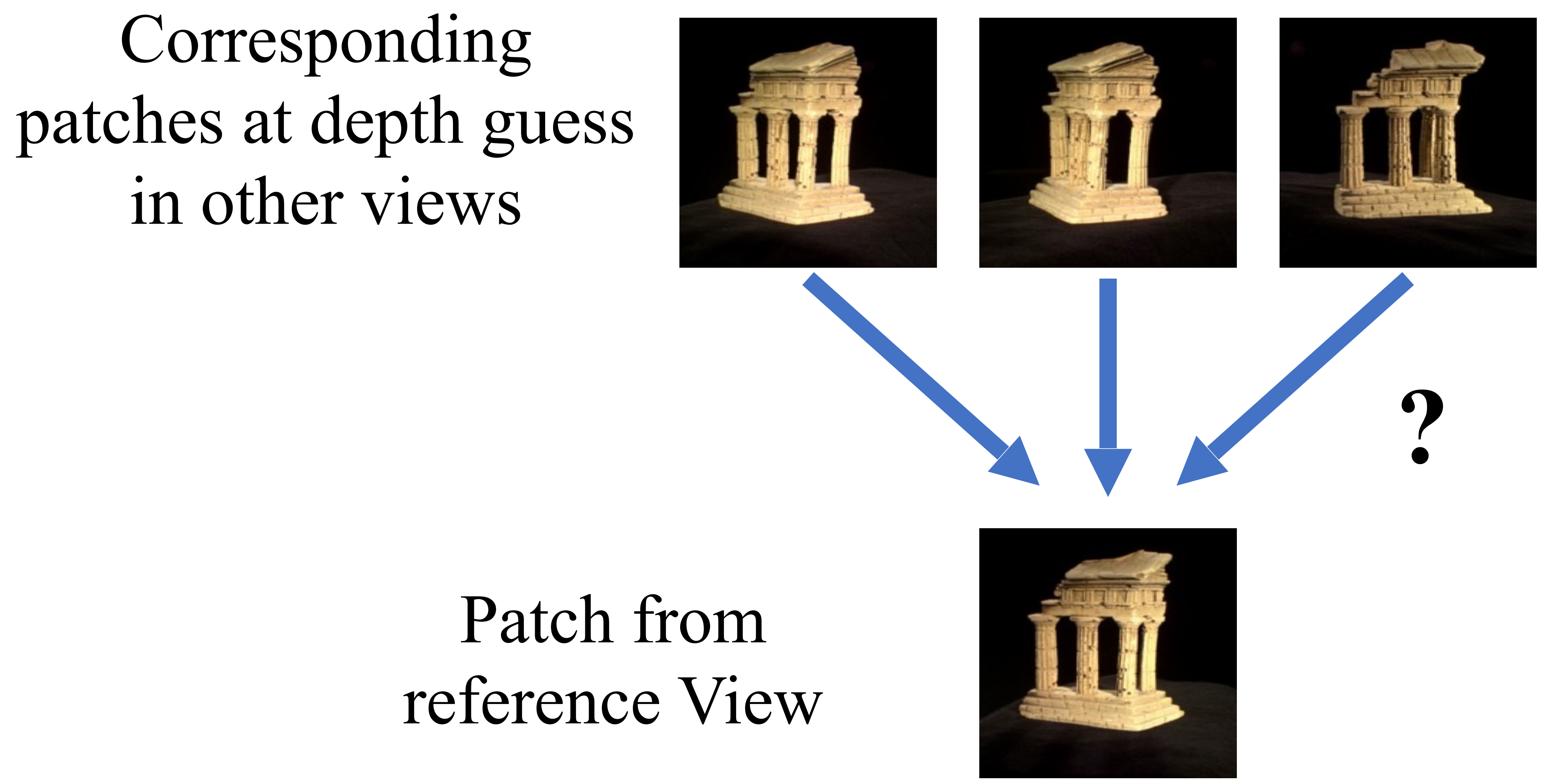


Source: Y. Furukawa

Multi-view stereo

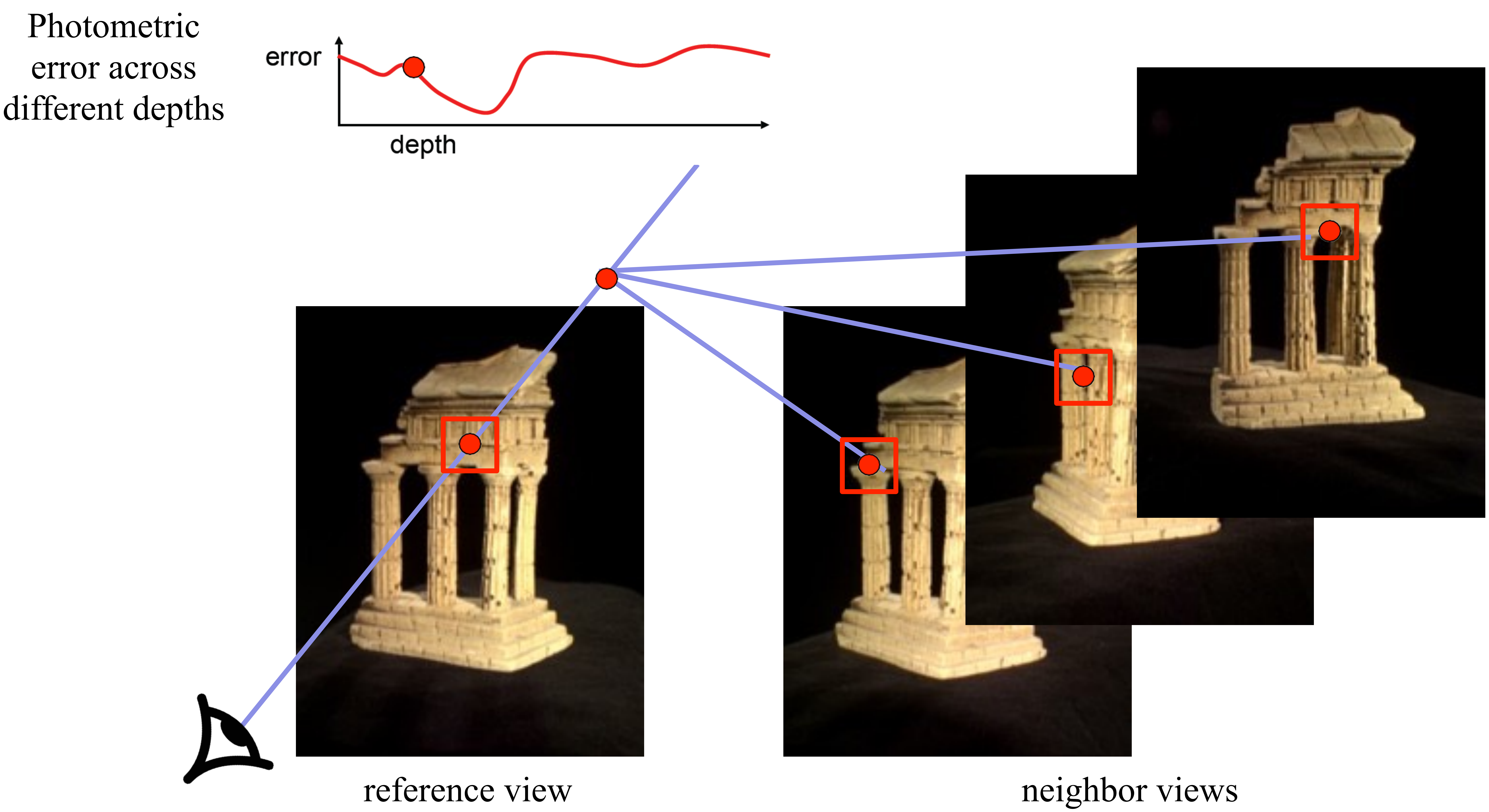
Evaluate the likelihood of a particular depth for a particular reference patch:





Source: Y. Furukawa

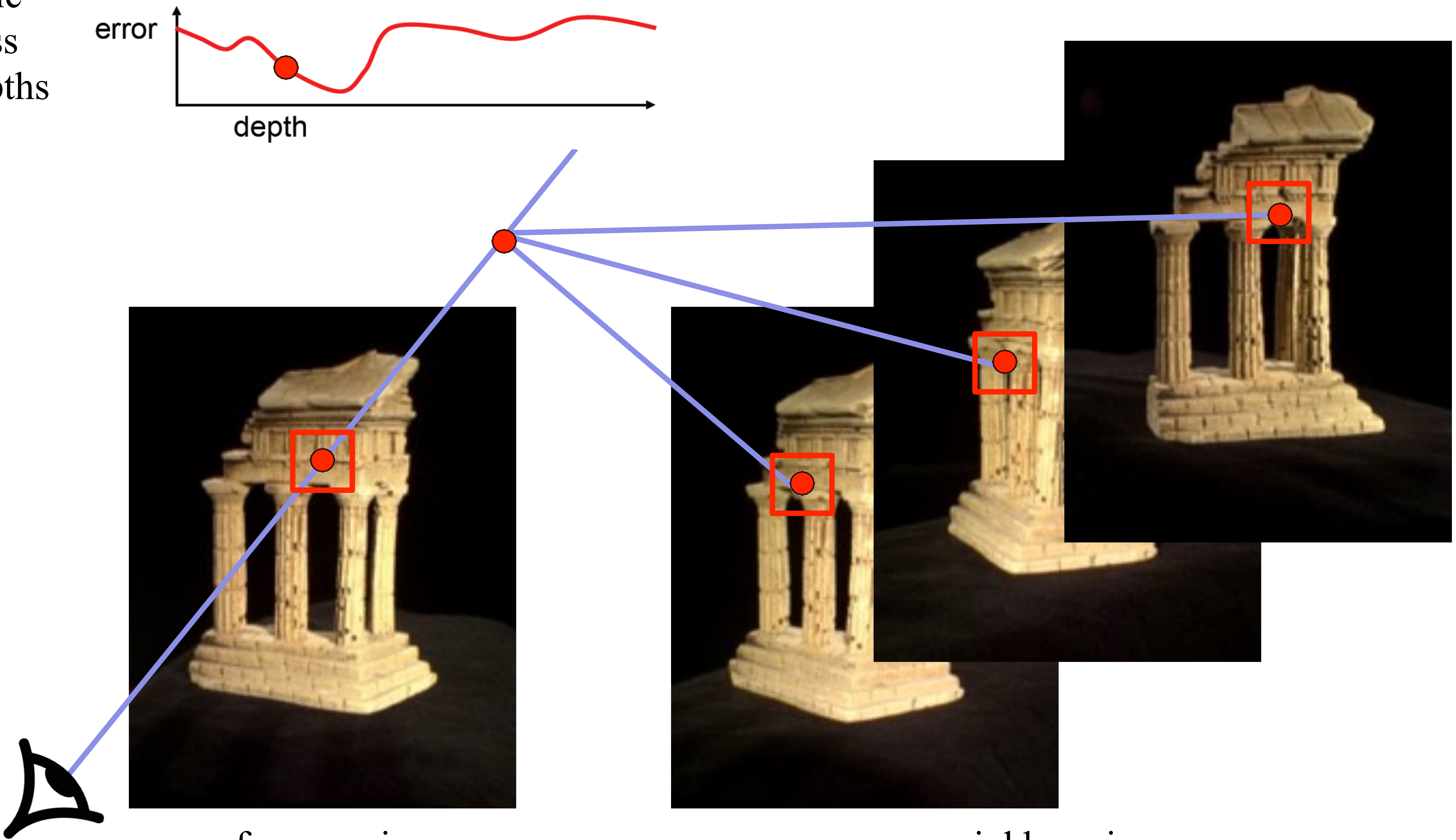
Multi-view stereo



Source: Y. Furukawa

Multi-view stereo

Photometric error across different depths



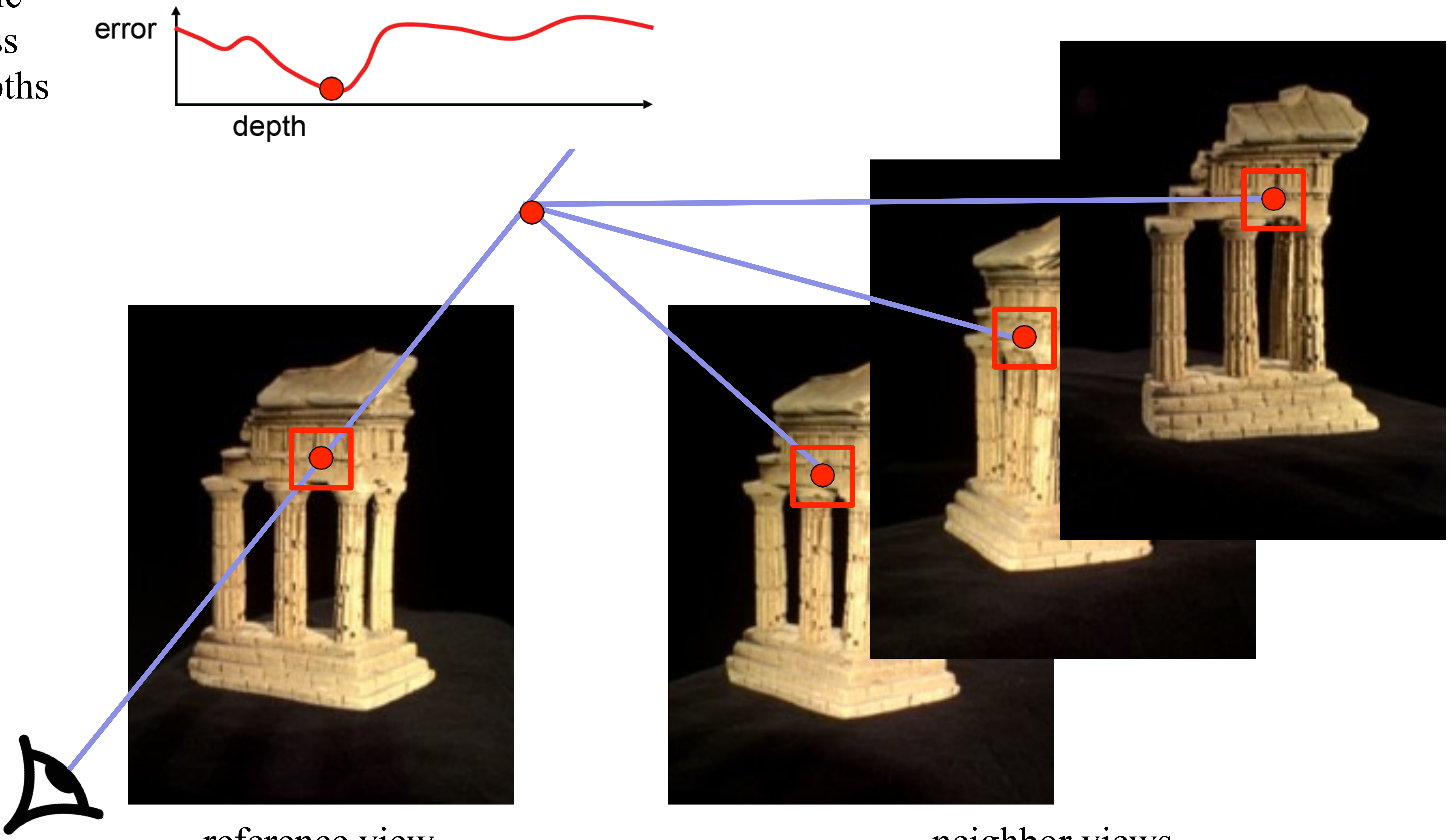
reference view

neighbor views

Source: Y. Furukawa

Multi-view stereo

Photometric error across different depths

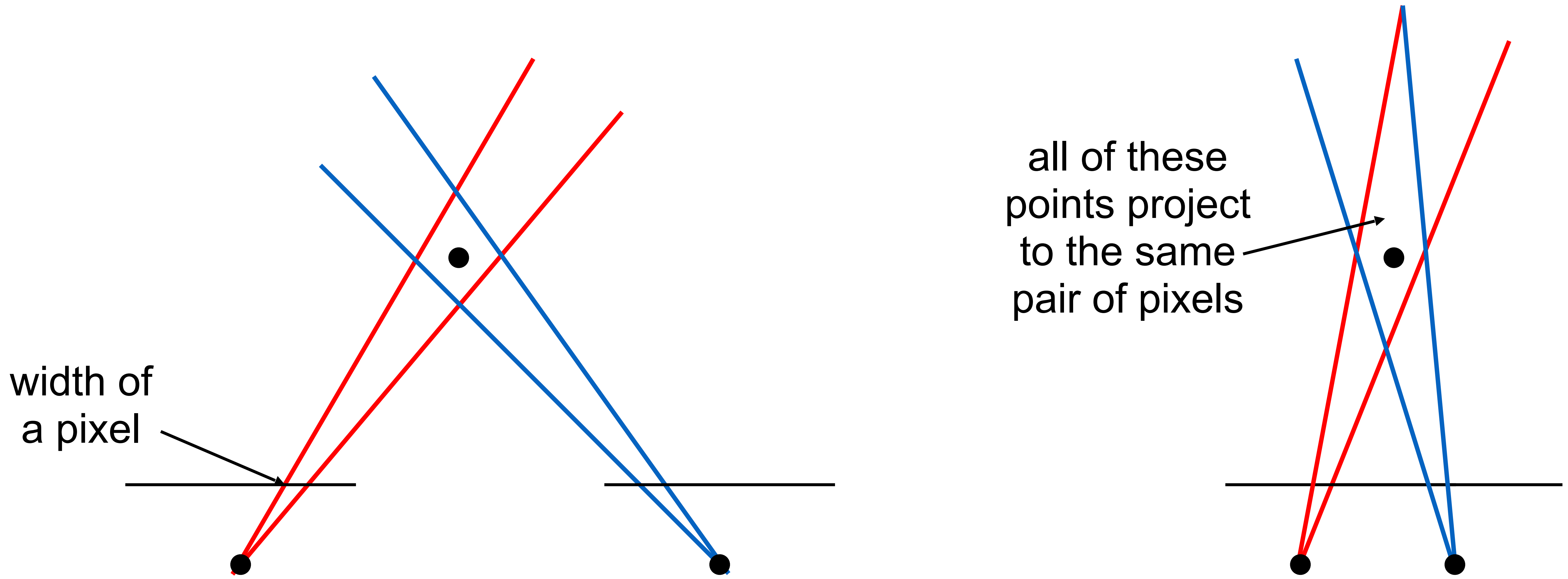


reference view

neighbor views

Source: Y. Furukawa

Multiple-baseline stereo



Large Baseline

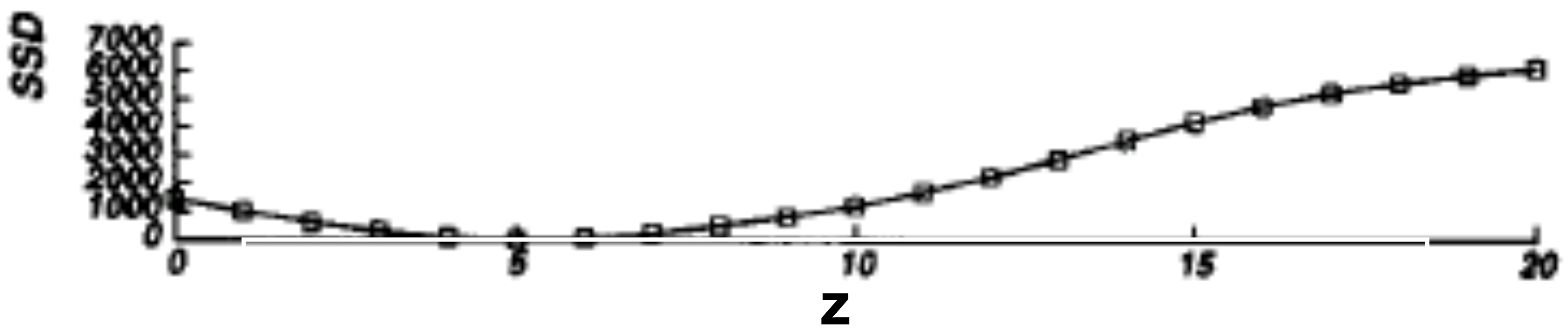
Small Baseline

What's the optimal baseline?

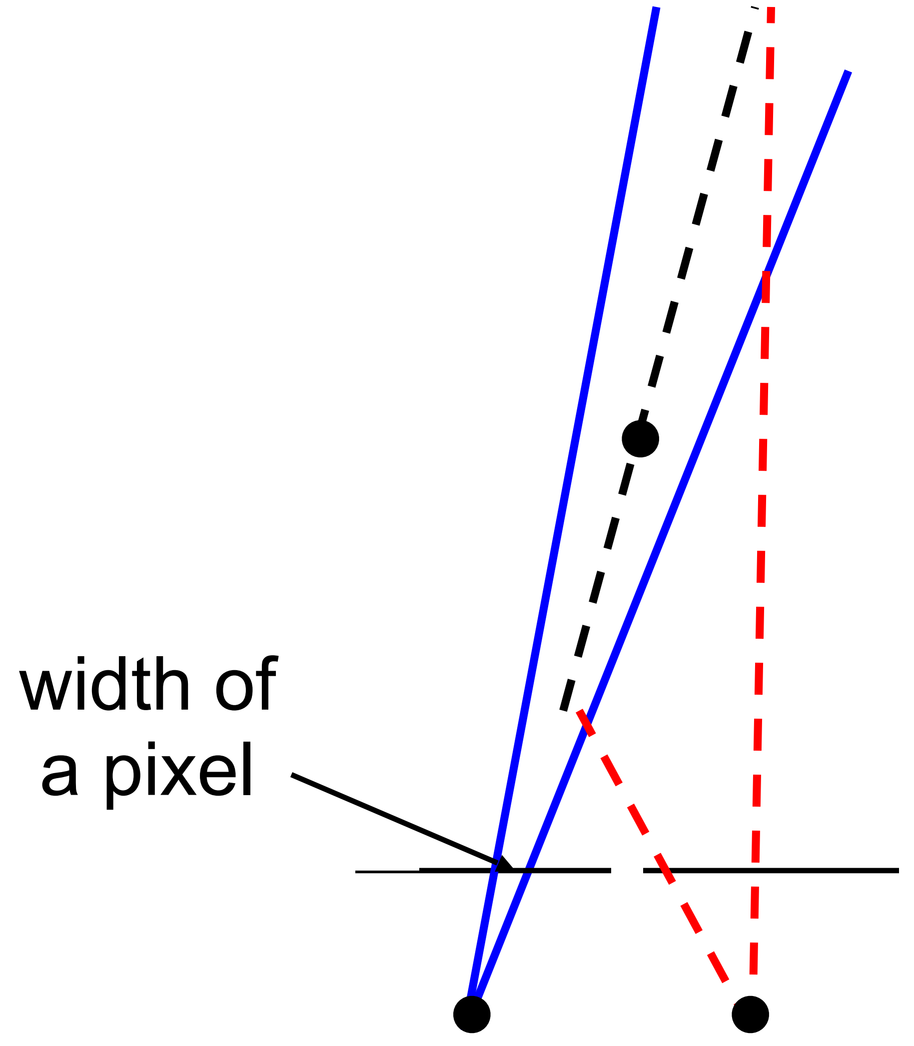
- Too small: large depth error
- Too large: difficult search problem

Source: N. Snavely

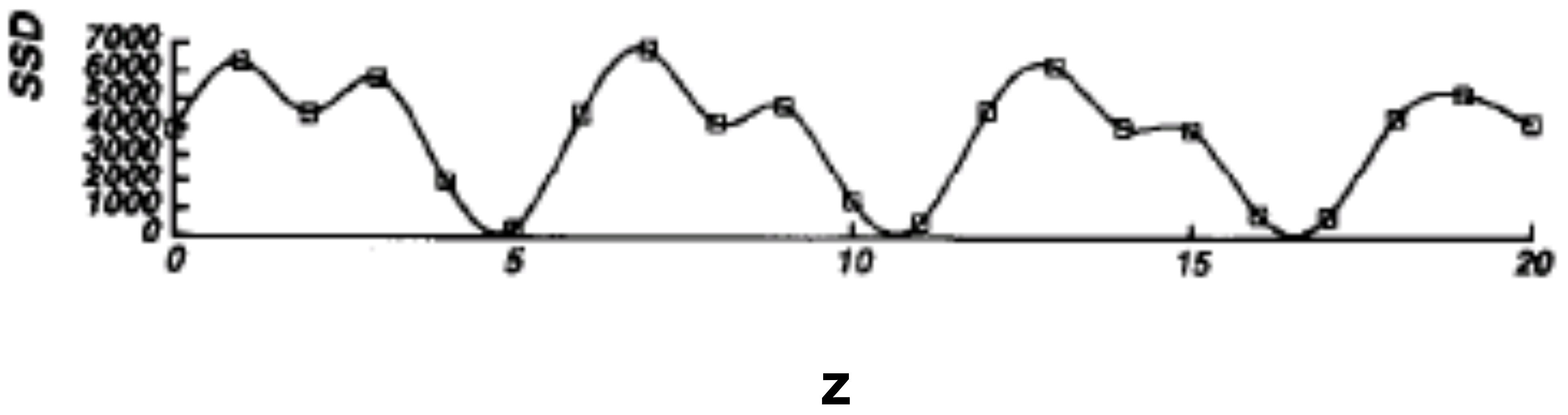
Multiple-baseline stereo

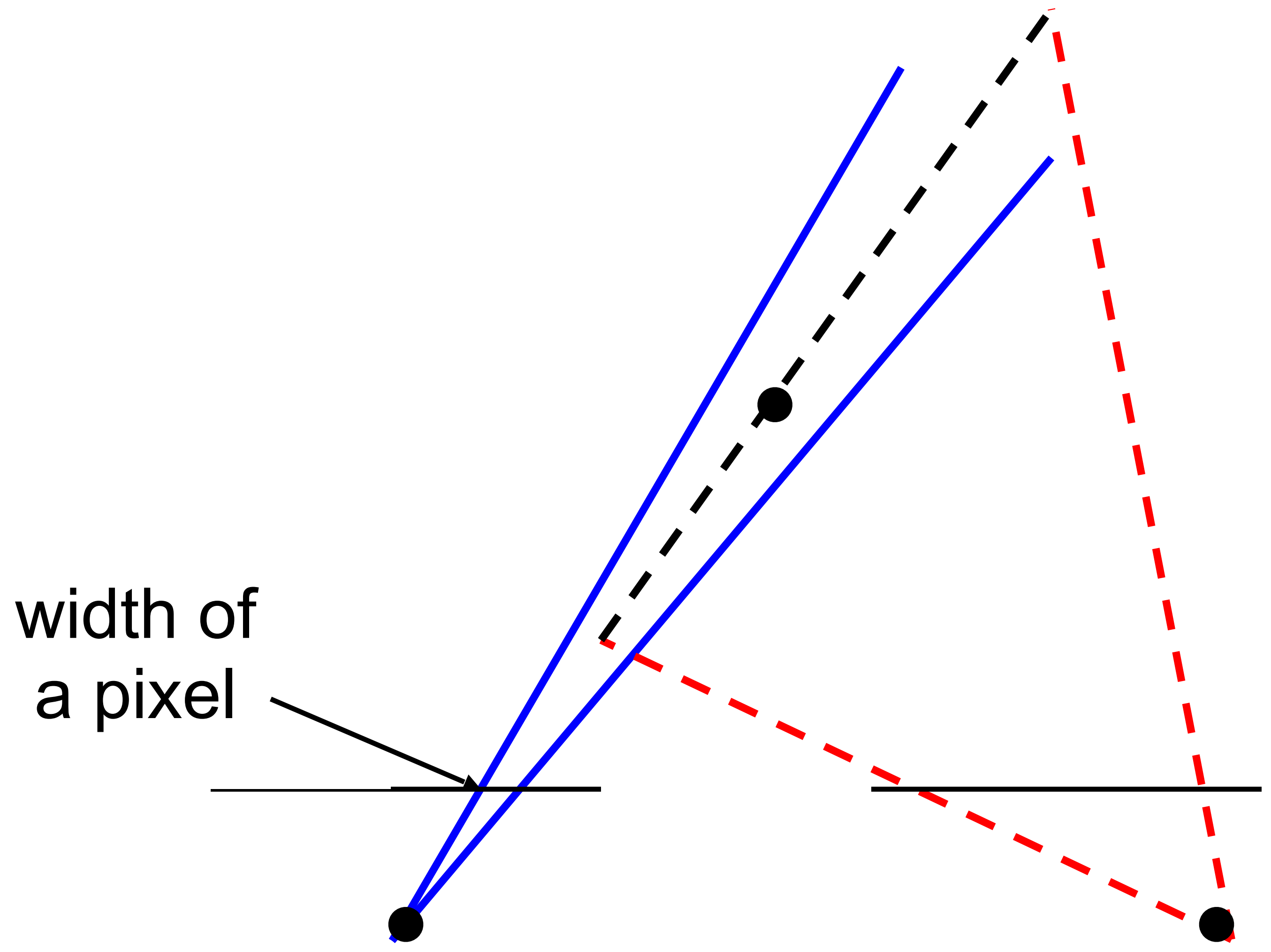


pixel matching score



- For short baselines, estimated depth will be less precise due to narrow triangulation





- For larger baselines, must search larger area in second image

M. Okutomi and T.Kanade, "A Multiple-Baseline Stereo System," IEEE Trans. on Pattern Analysis and Machine Intelligence.

Source: N. Snavely

Next class: neural fields