Lecture 19: Estimating geometry

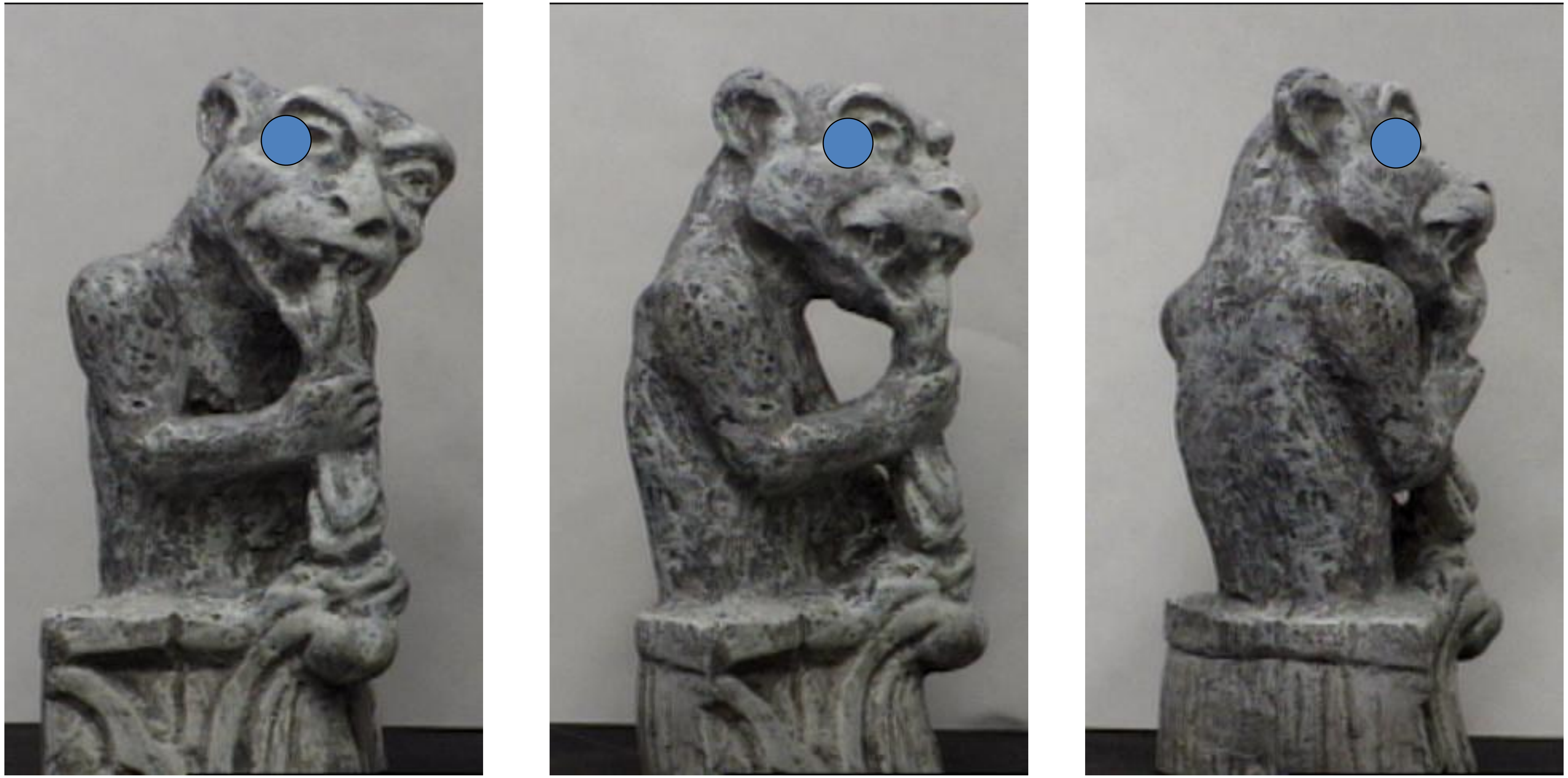
Slides mostly from N. Snavely

Announcements

- Project proposal submission now open
- This week's section: panorama stitching and project office hours
- Thank you for answering questions on Piazza

Recall: triangulation

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

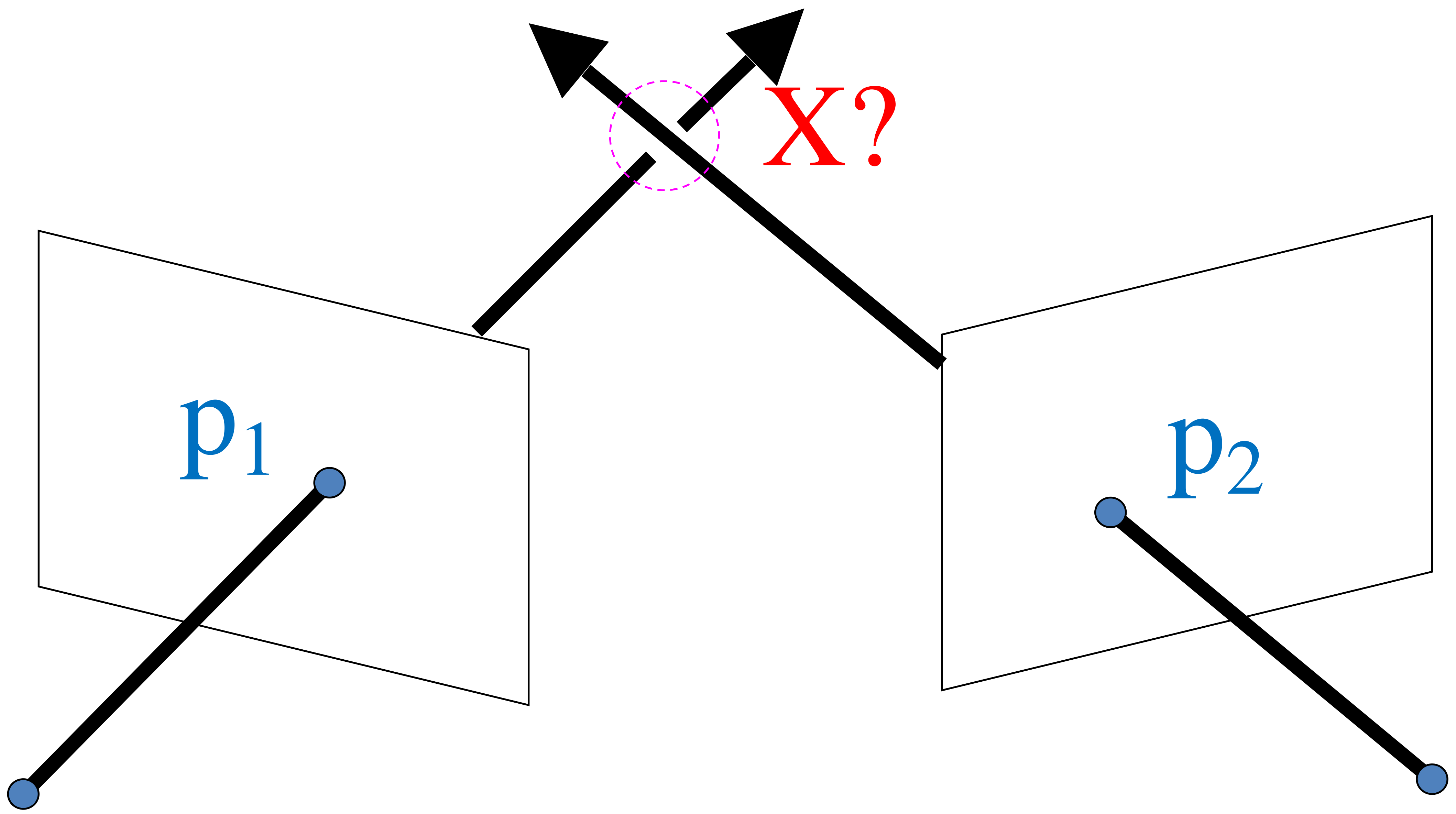


Source: D. Fouhey

Triangulation

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

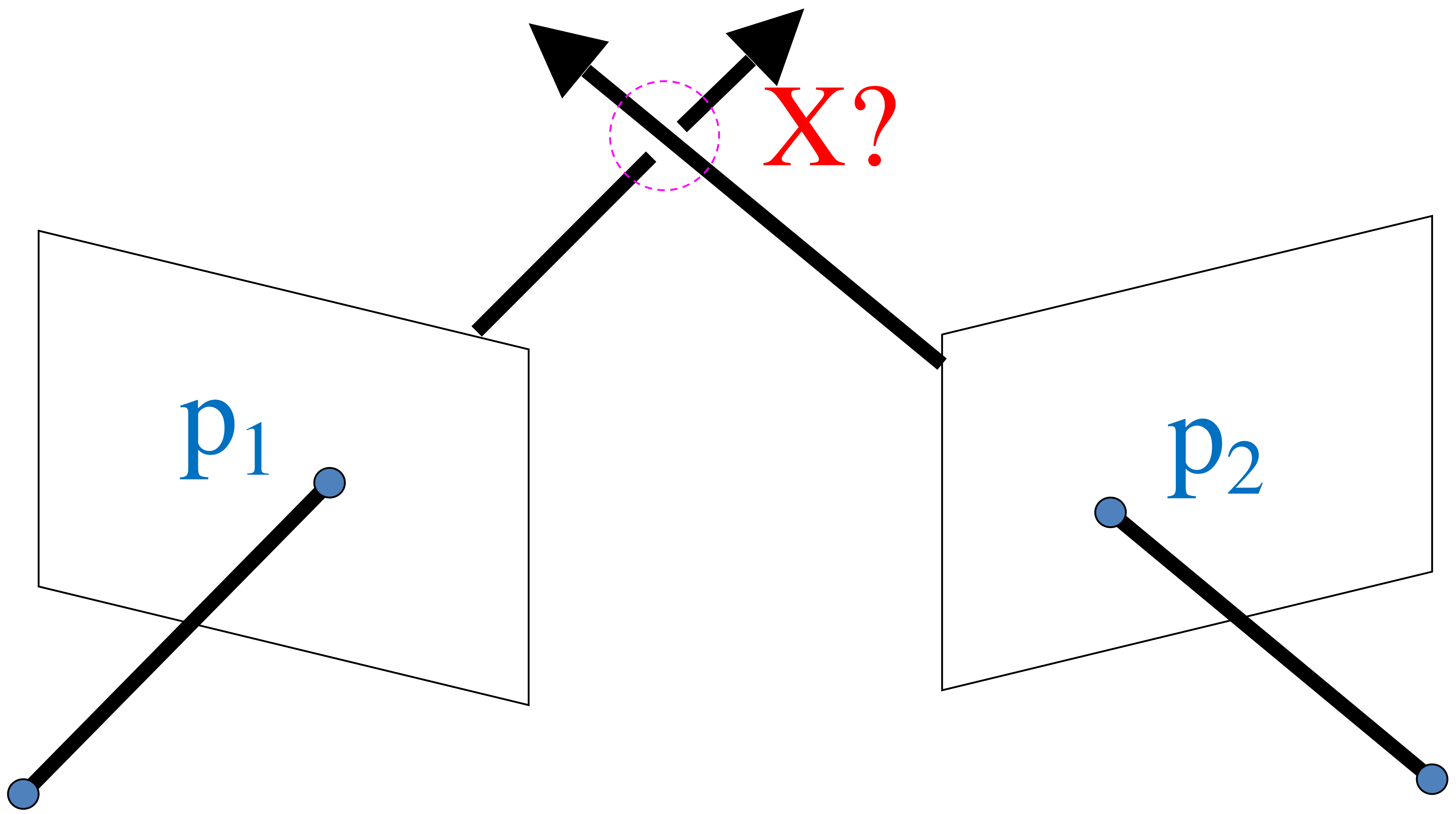
Why is the calibration here important?



Source: D. Fouhey

Triangulation

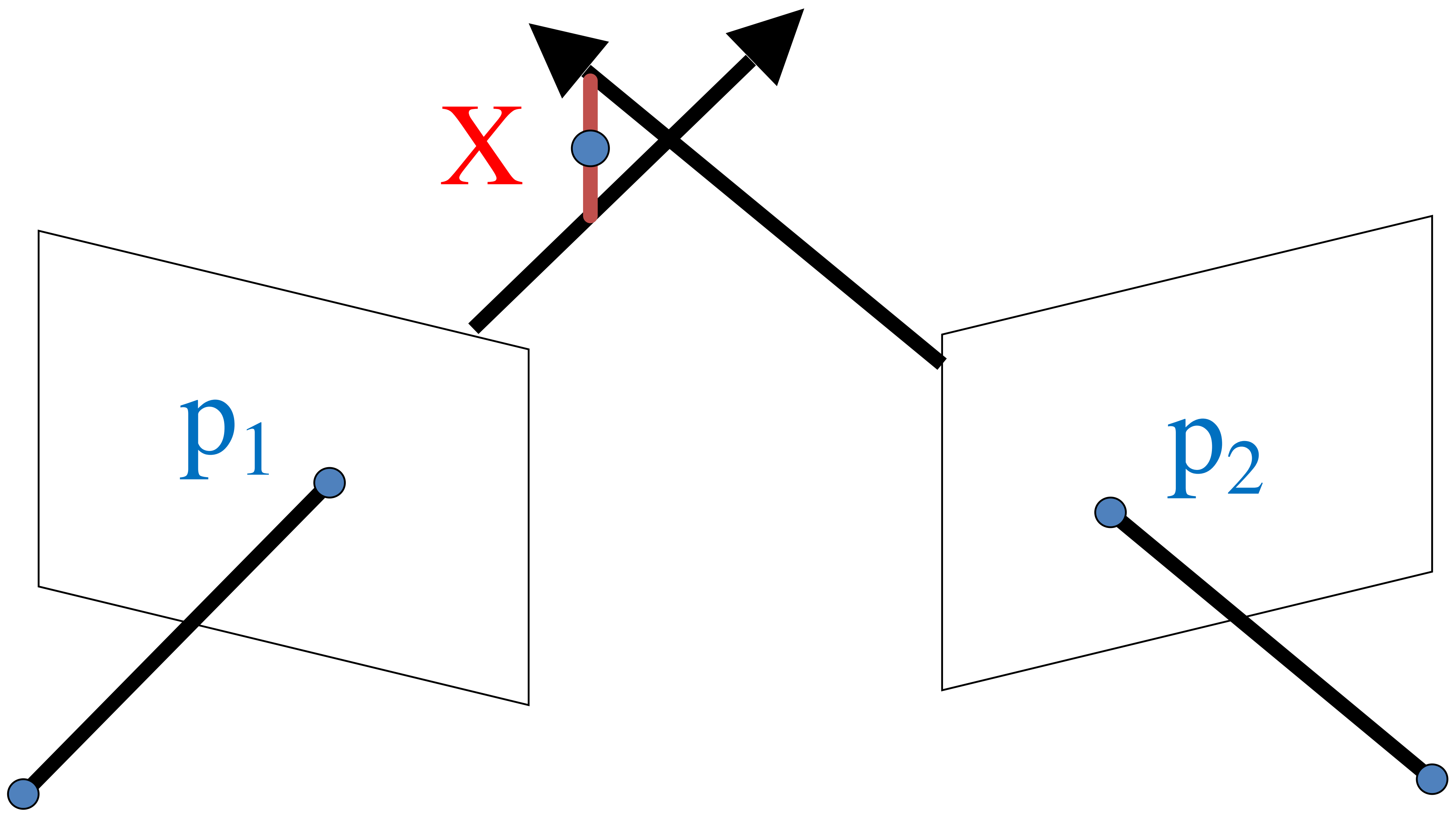
Rays in principle should intersect, but in practice usually don't exactly due to noise, numerical errors.



Source: D. Fouhey

Triangulation – Geometry

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

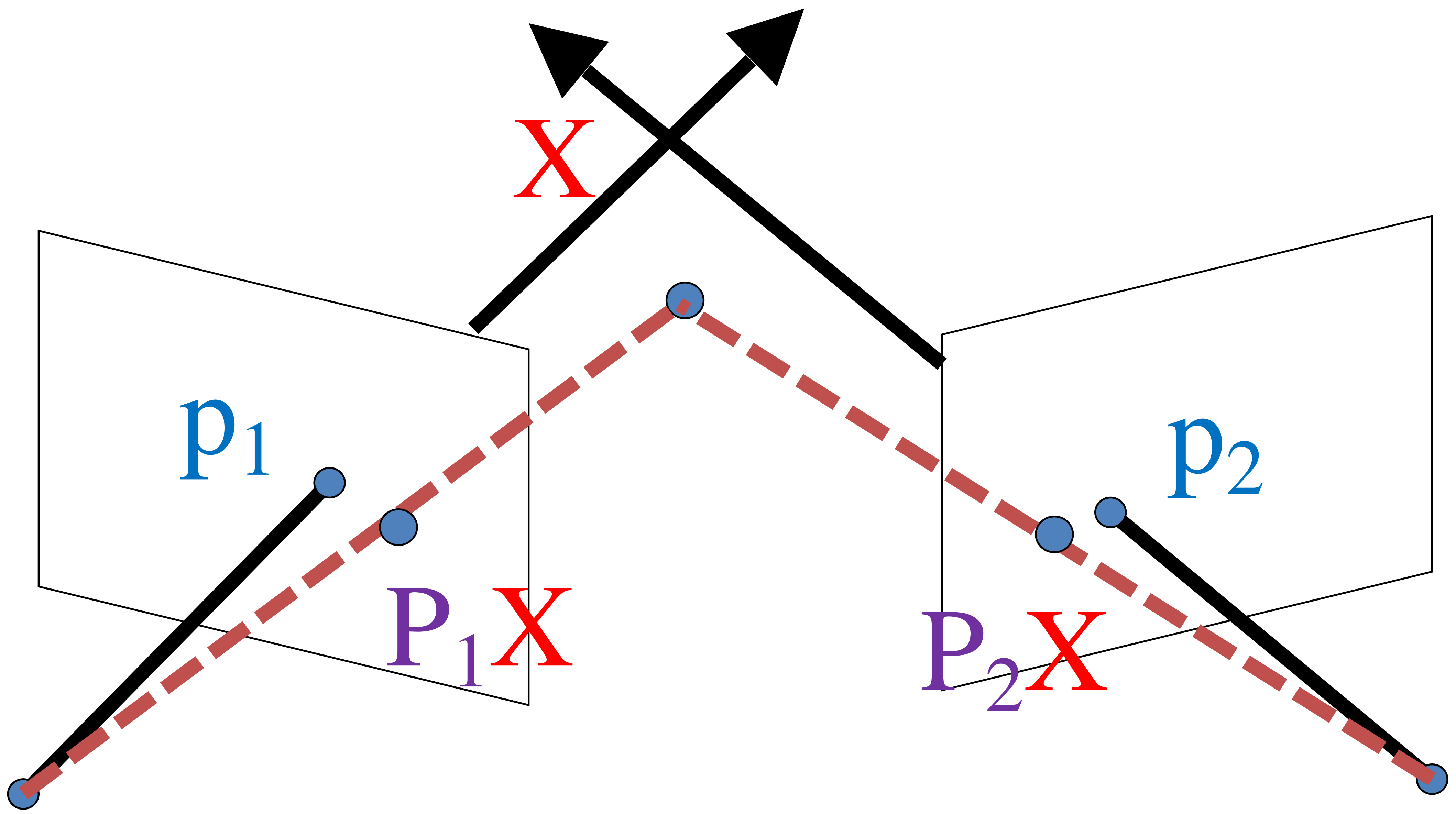


Source: D. Fouhey

Triangulation – Non-linear Optim.

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

where d is distance in image space



Source: D. Fouhey

Estimating 3D structure

- Given many images, how can we…
  1. Figure out where they were all taken from?
  2. Build a 3D model of the scene?

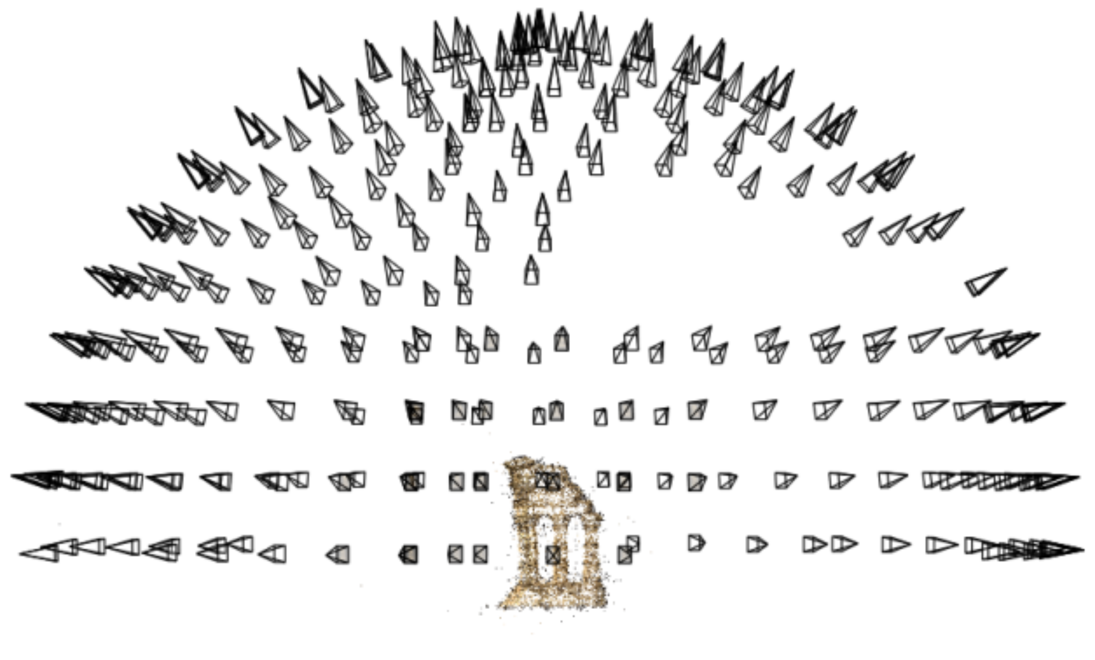
This is the **structure from motion** problem

Source: N. Snavely

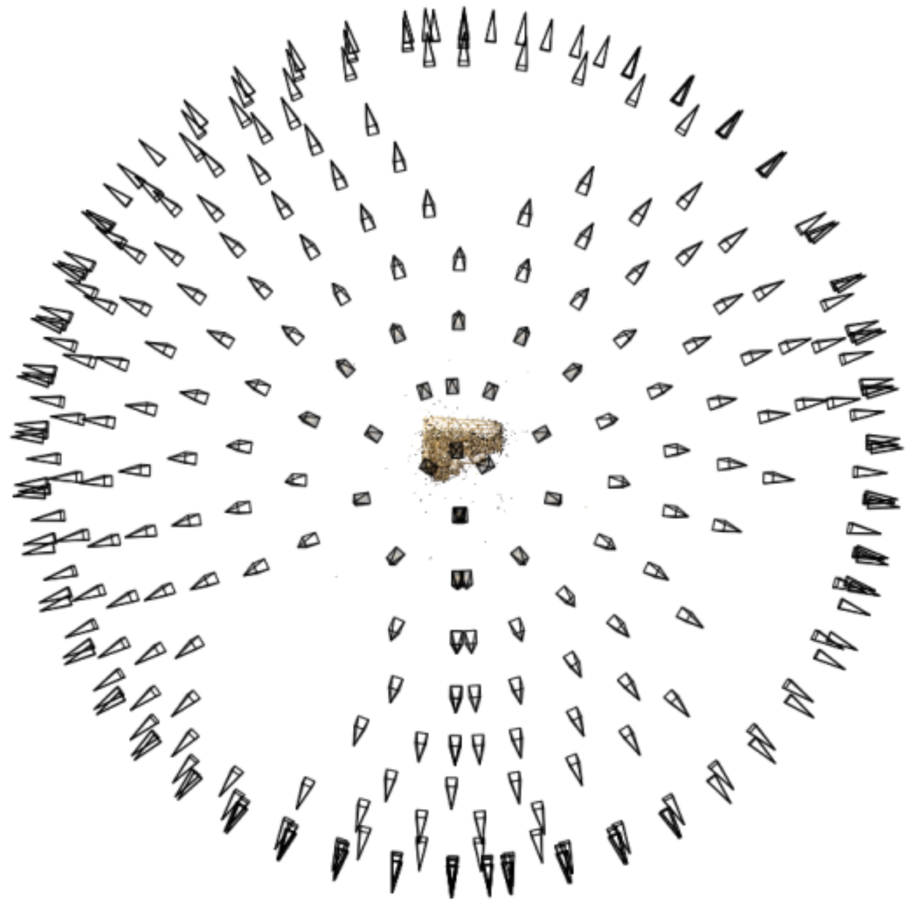
Today

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

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

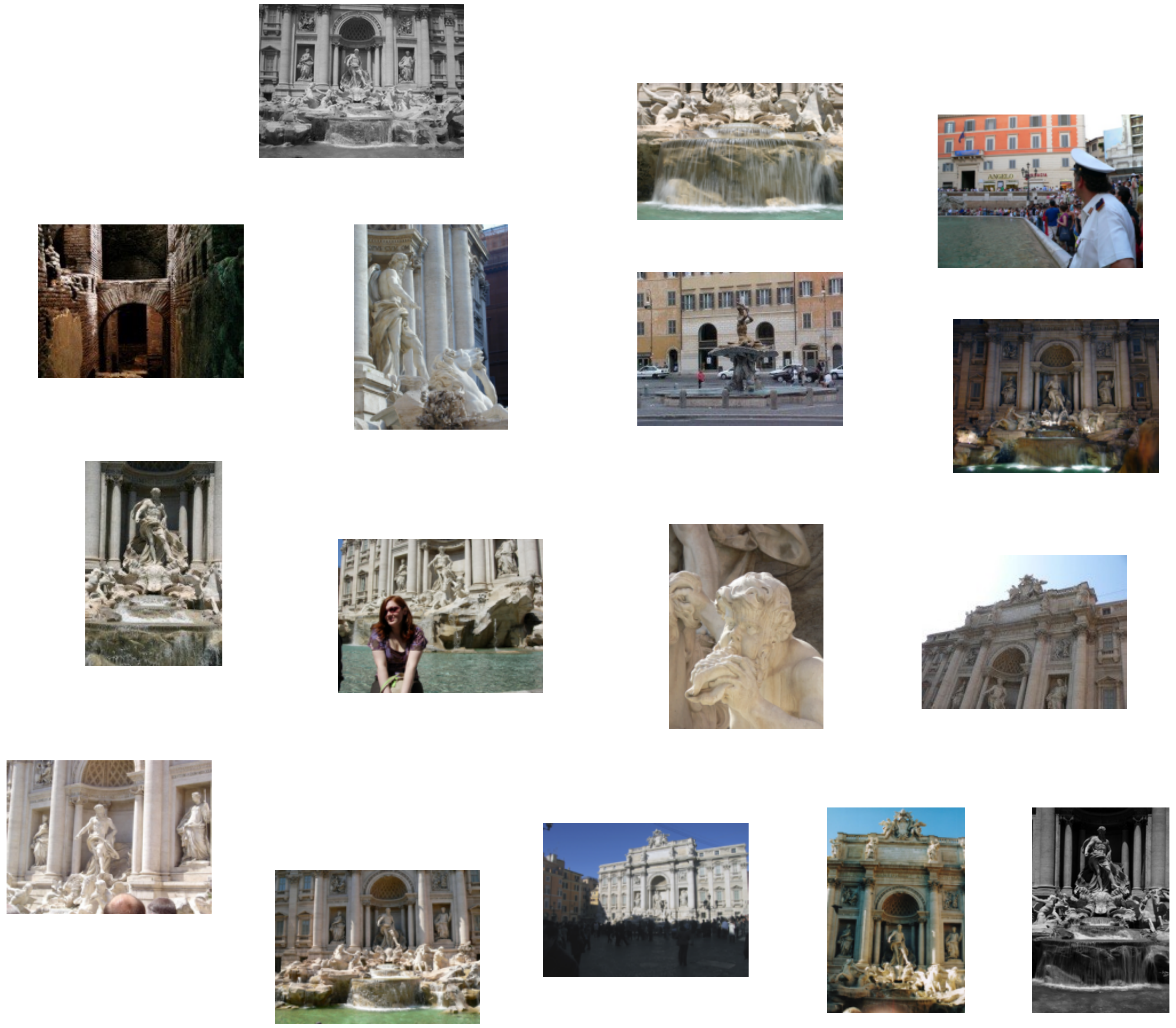
Source: N. Snavely

Example: Photo Tourism

Source: N. Snavely

Feature detection

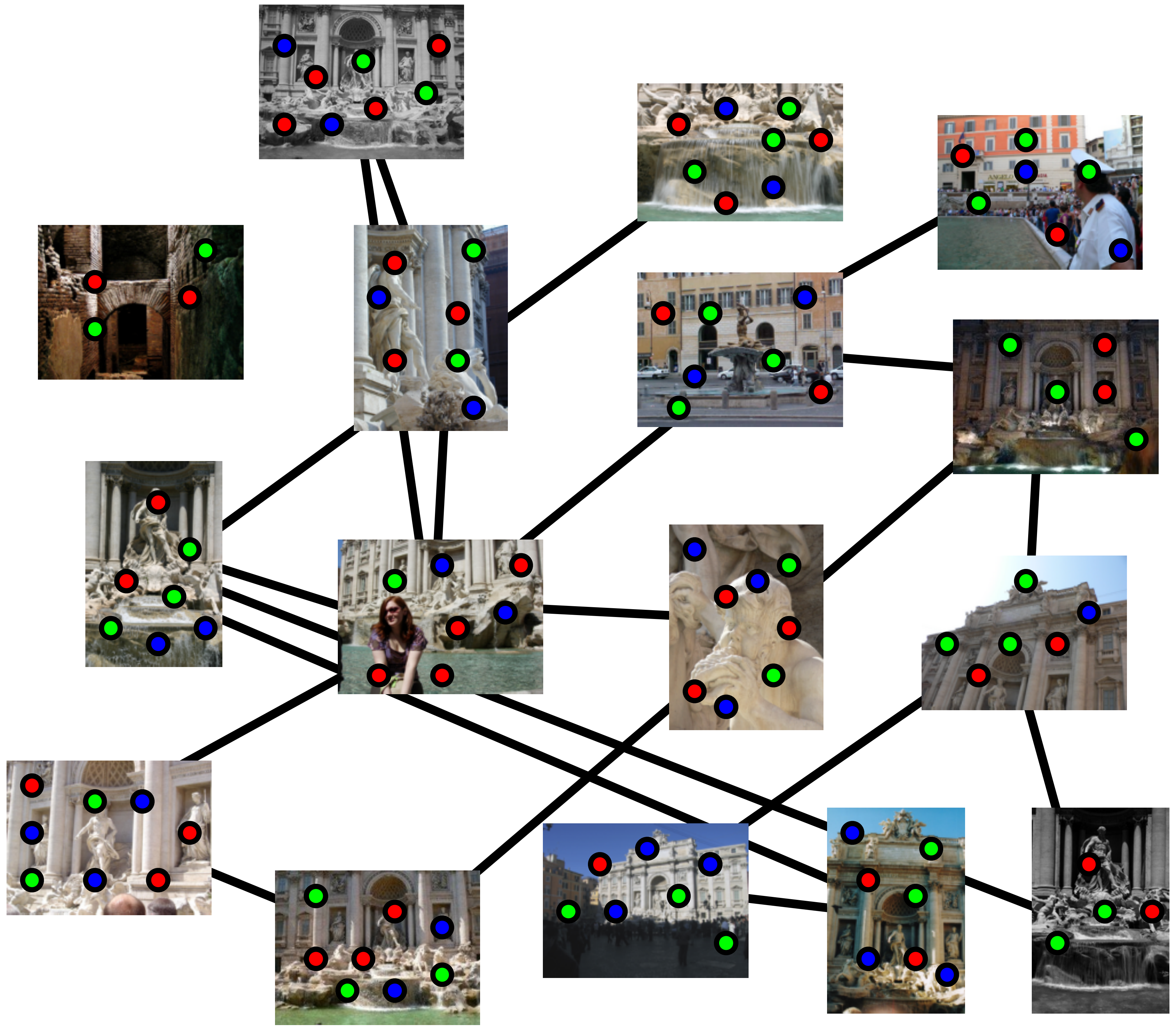
- Same process as with homography estimation
- Detect features using SIFT



Source: N. Snavely

Feature matching

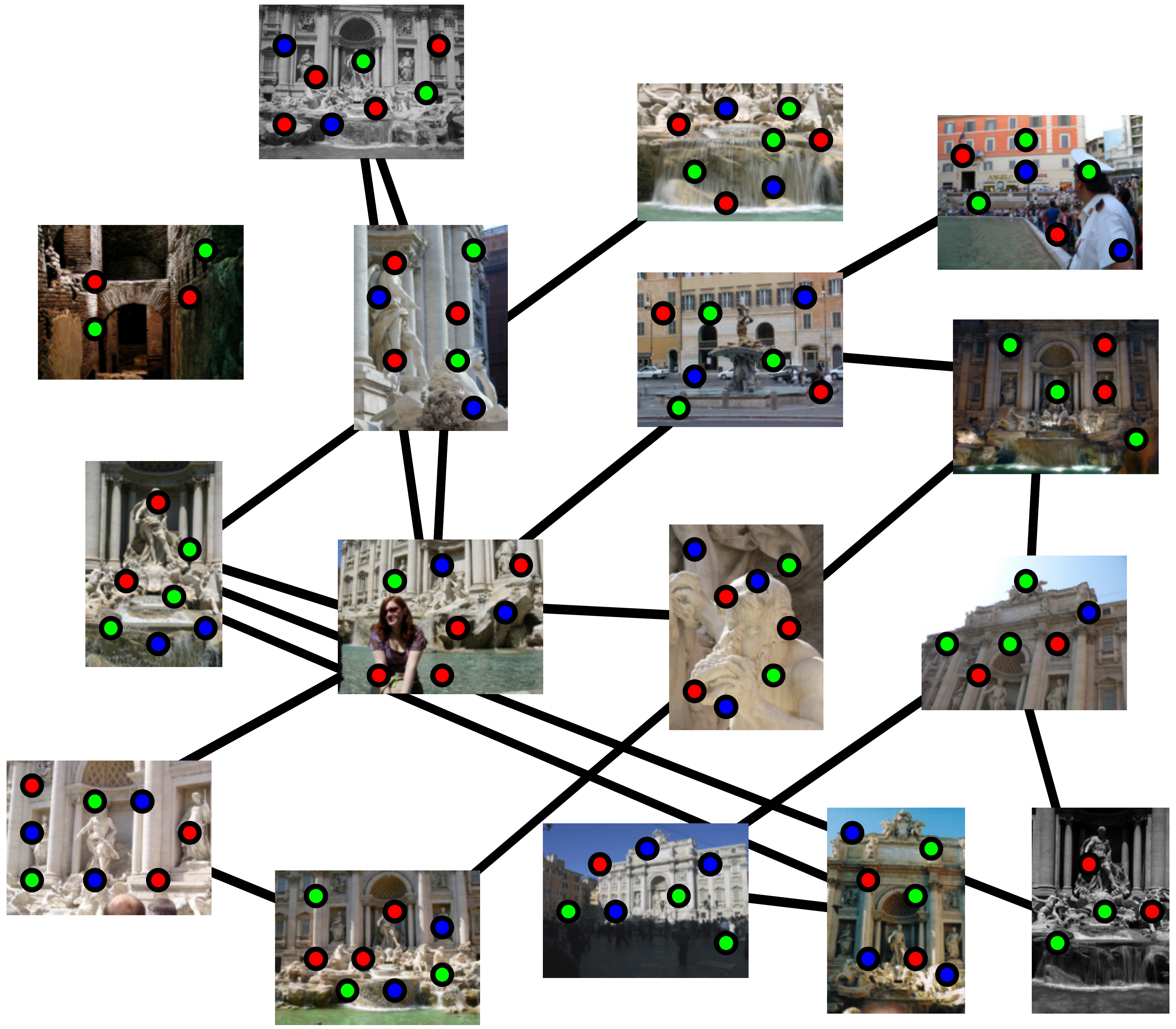
Match features between each pair of images



Source: N. Snavely

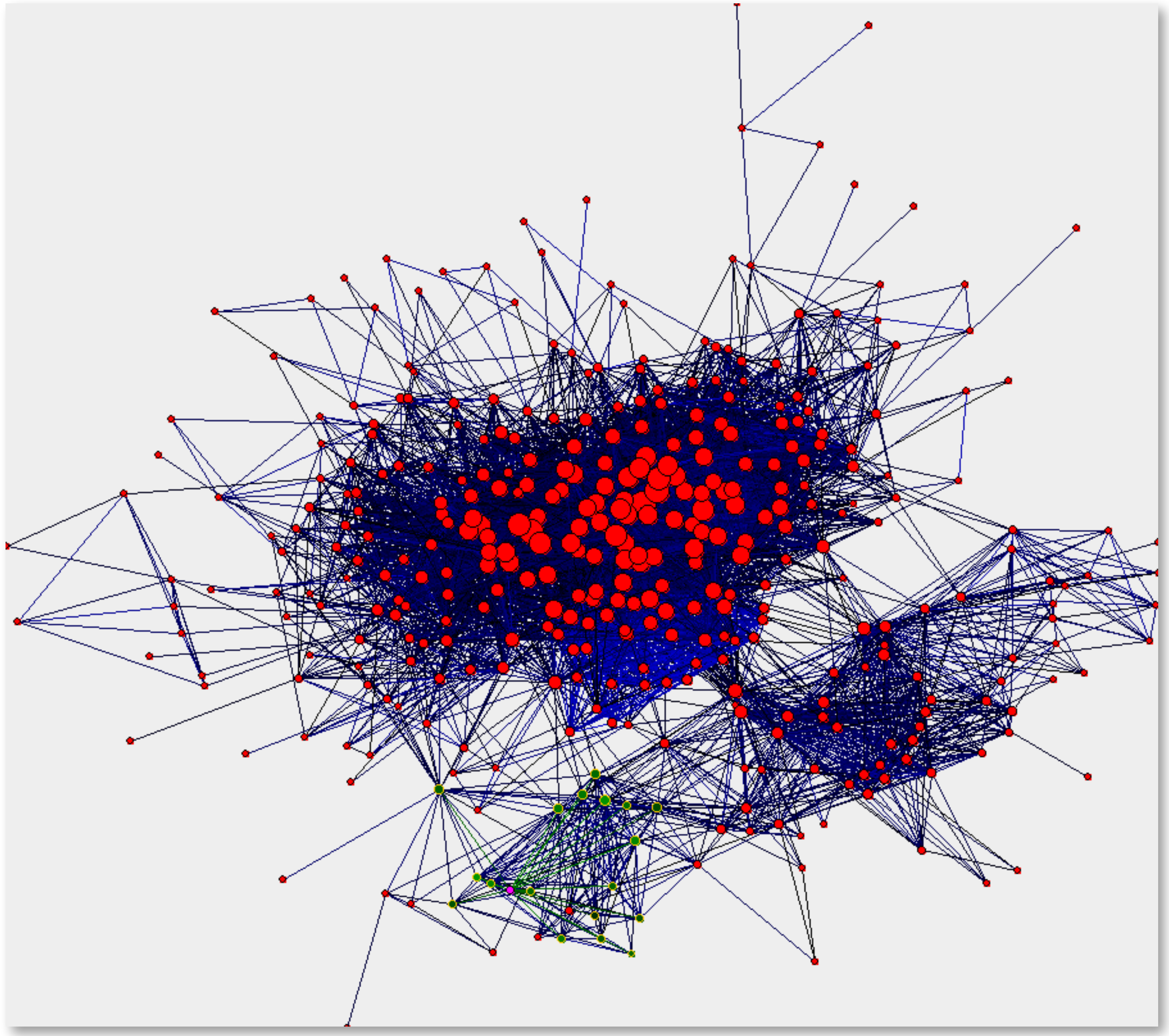
Feature matching

- 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.

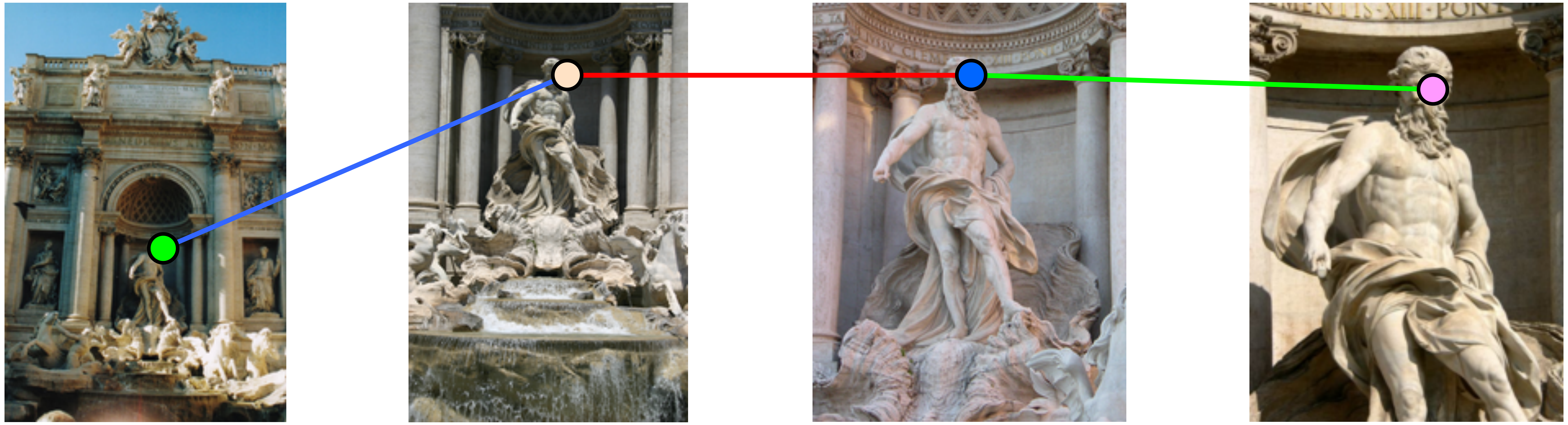
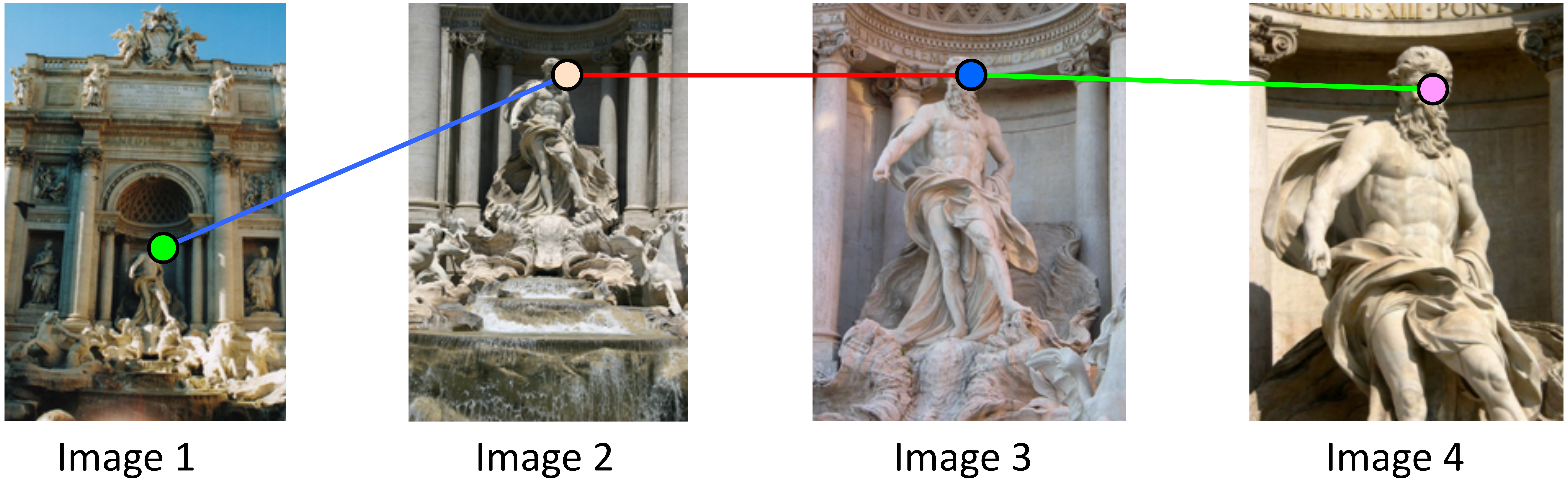


Image 1 Image 2 Image 3 Image 4

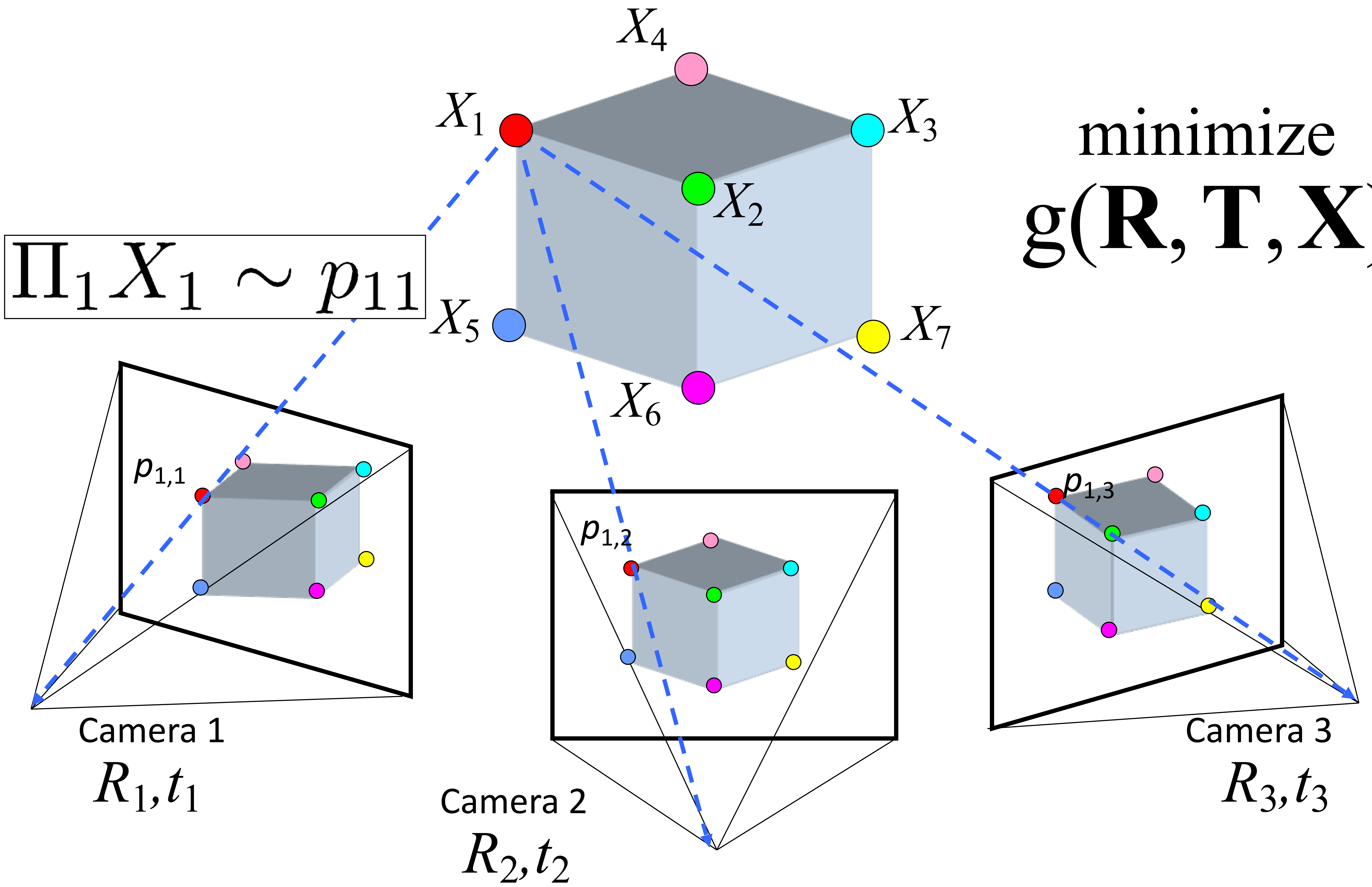
Source: N. Snavely

Correspondence estimation

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



Structure from motion



minimize

$$g(\mathbf{R}, \mathbf{T}, \mathbf{X})$$

Source: N. Snavely

Structure from motion

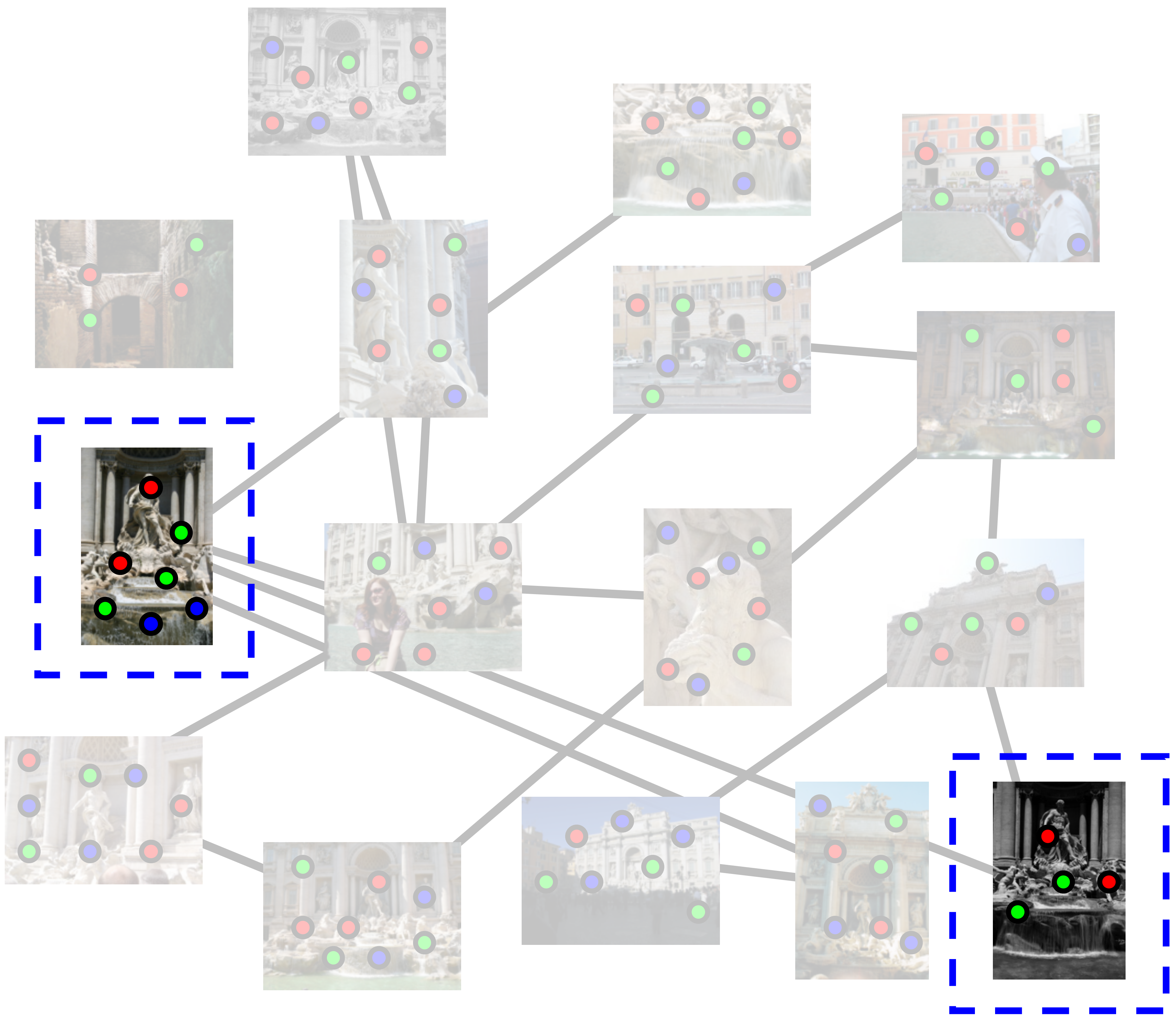
- Minimize sum of squared reprojection errors:

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

- 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

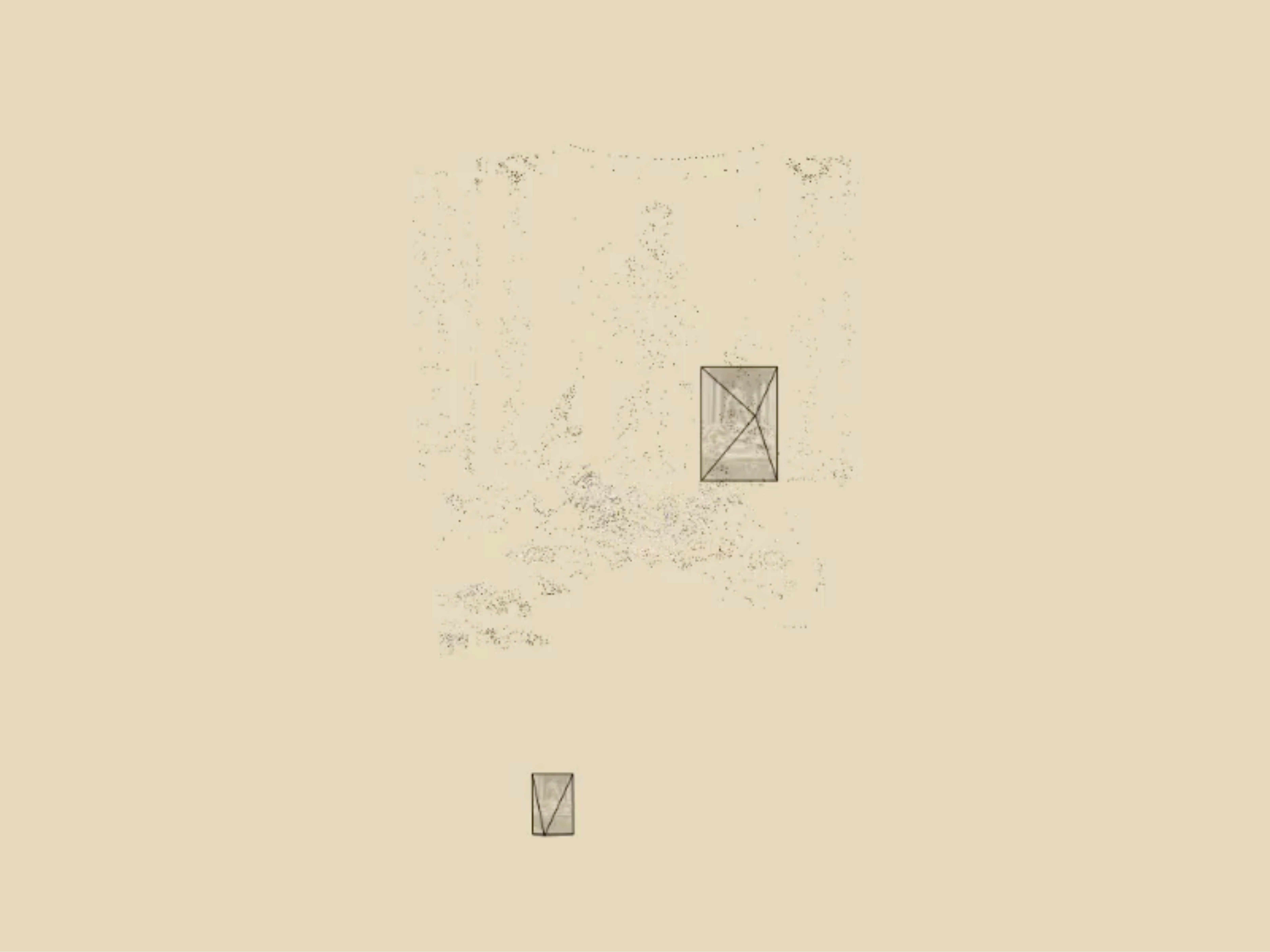
Source: N. Snavely

Incremental structure from motion



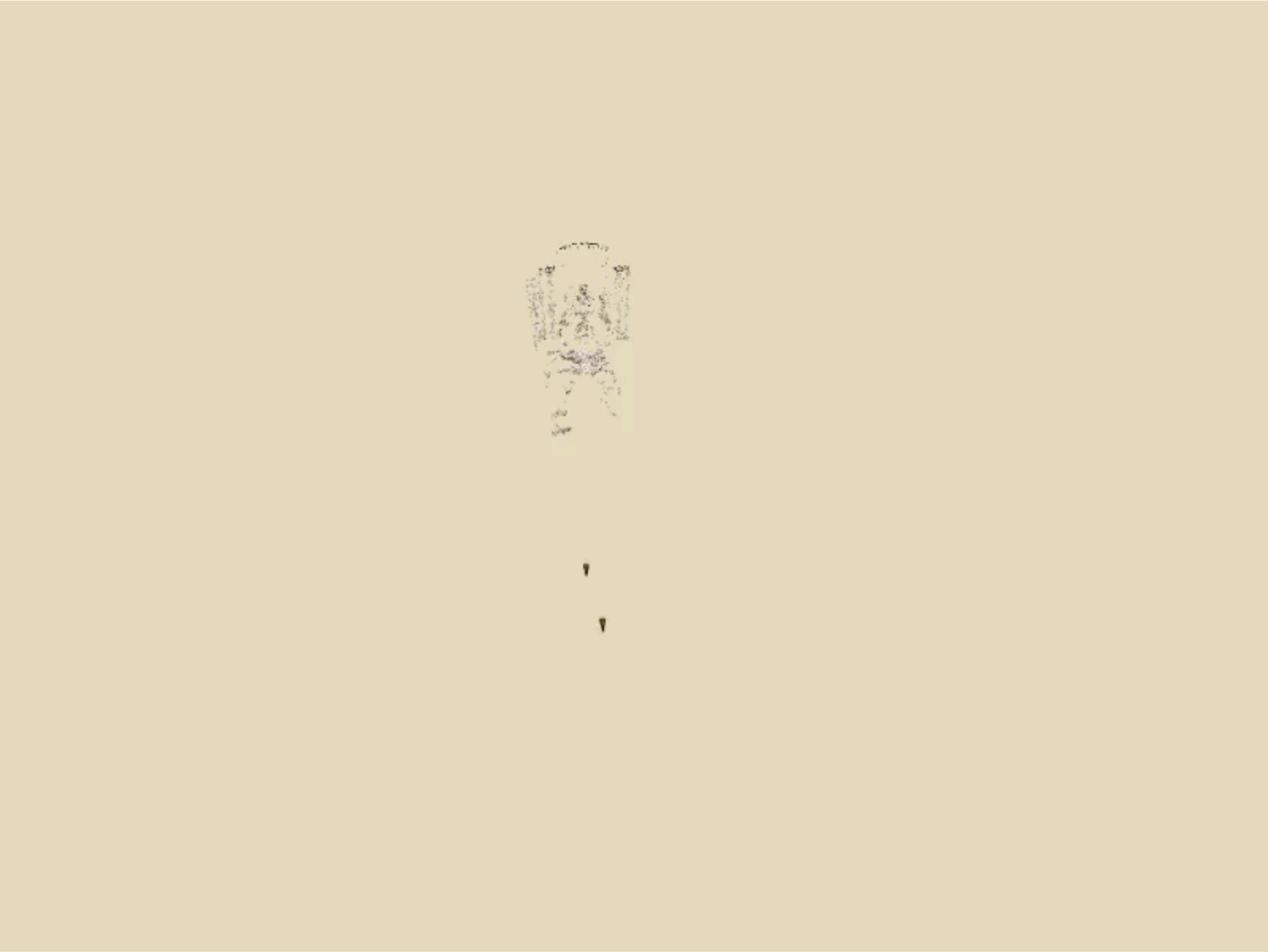
Source: N. Snavely

Incremental structure from motion



Source: N. Snavely

Incremental structure from motion



Source: N. Snavely

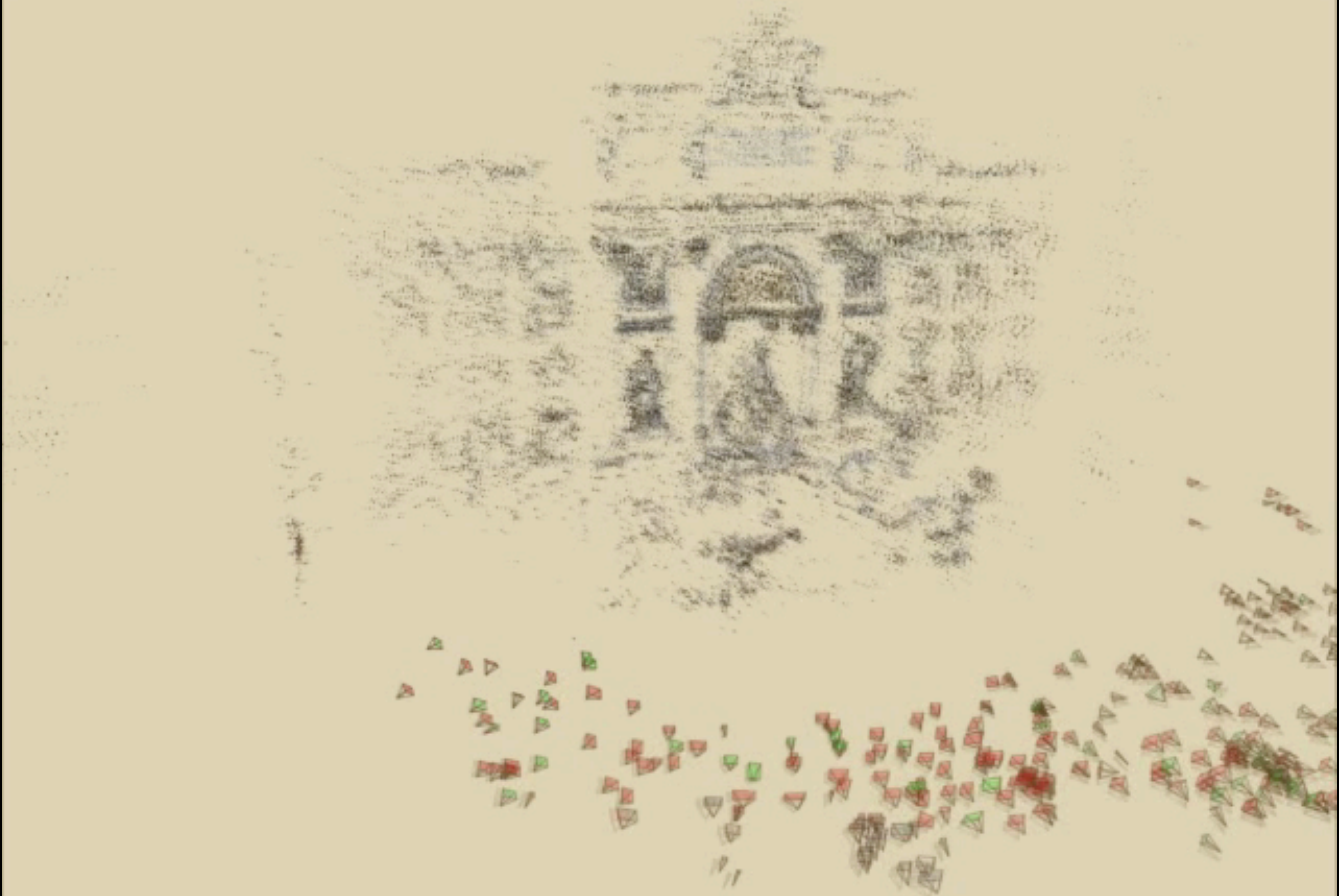
Photo Tourism

Exploring photo collections in 3D

Noah Snavely   Steven M. Seitz   Richard Szeliski

University of Washington   *Microsoft Research*

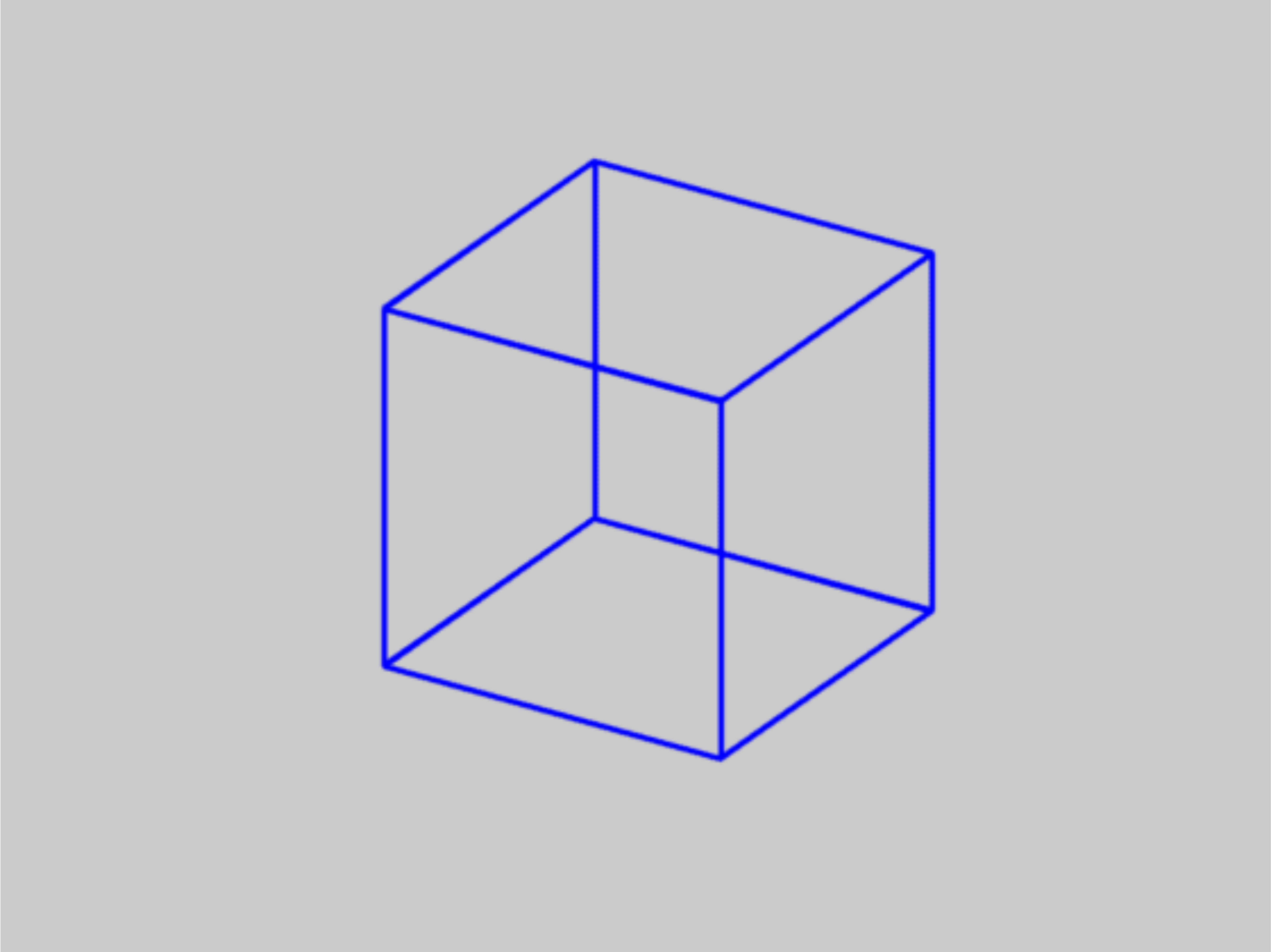
SIGGRAPH 2006




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

Is SfM always uniquely solvable?

No. Consider the Necker cube:



Source: N. Snavely

Is SfM always uniquely solvable?

Two interpretations:

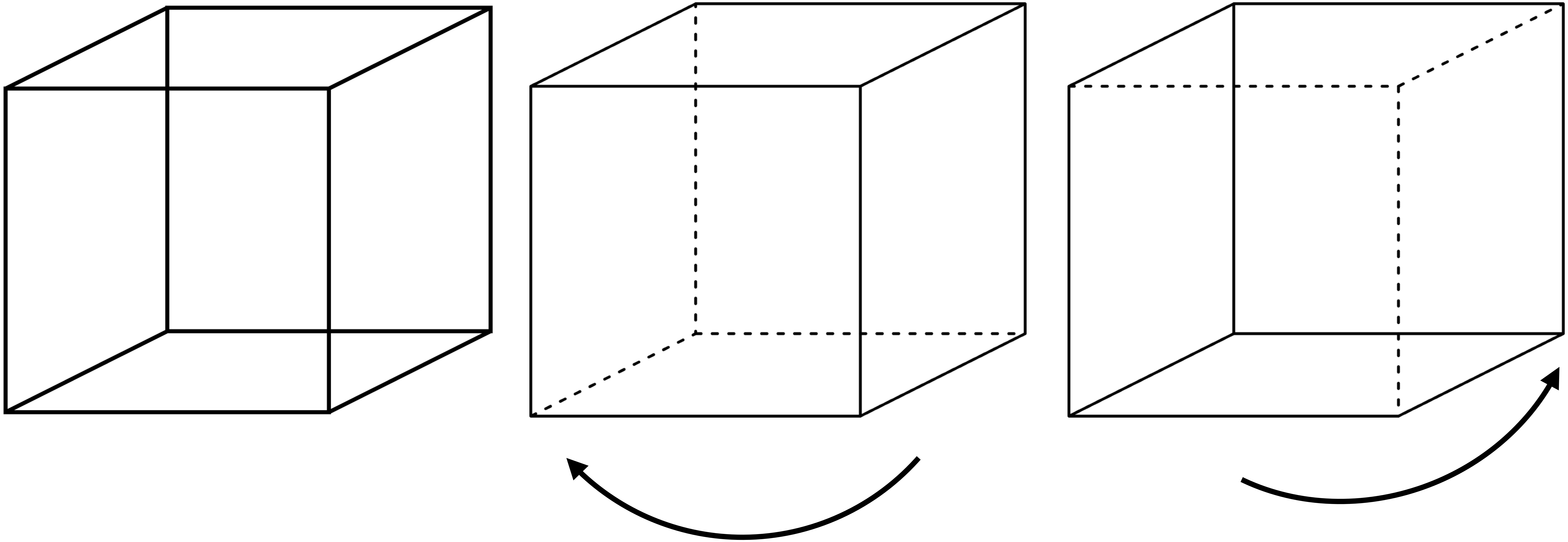


Image source: Wikipedia

Can also reconstruct from video

Source: N. Snavely

Applications: Visual Reality & Augmented Reality

Oculus

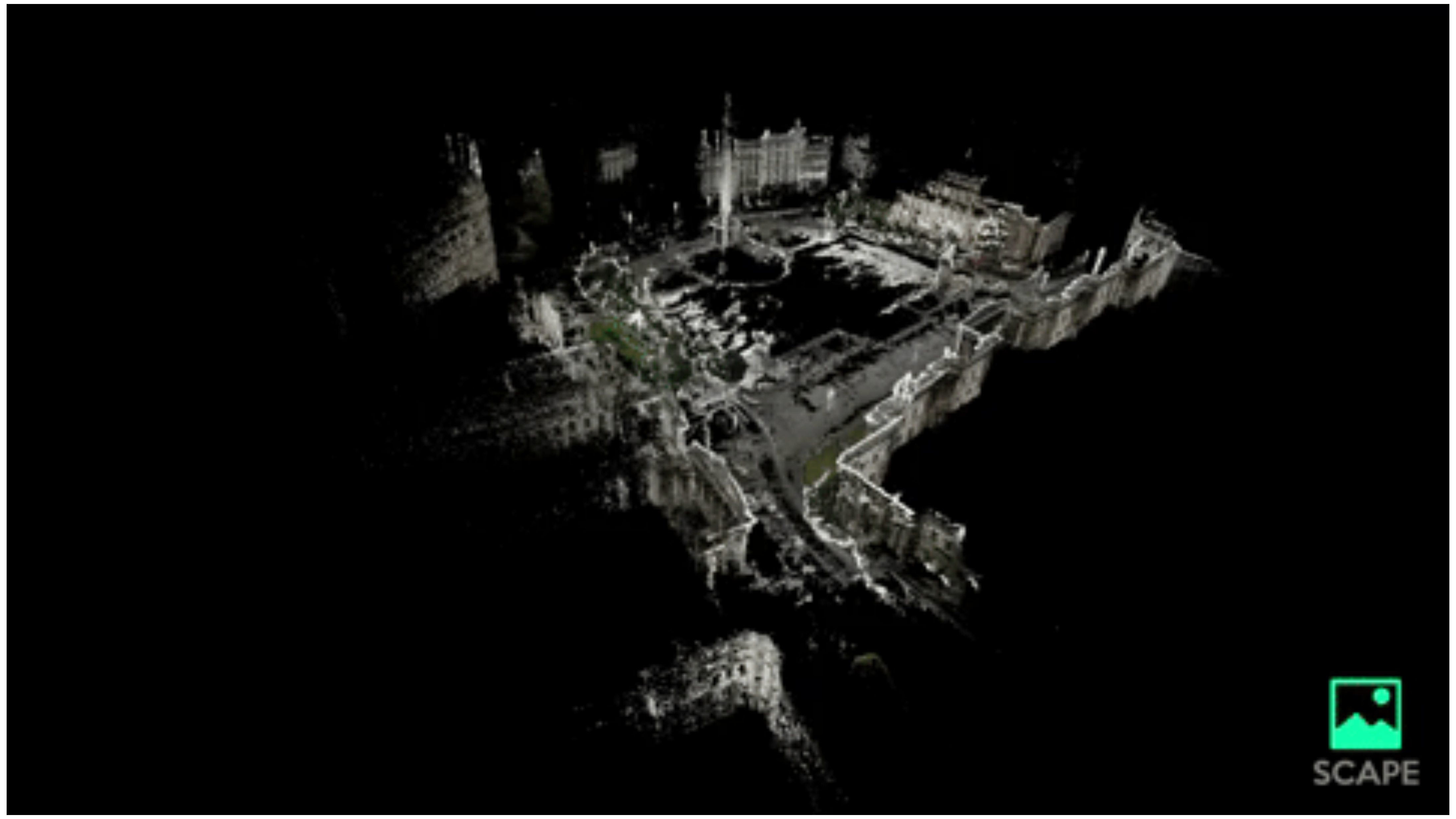
https://www.youtube.com/watch?v=KOG7yTz1iTA

Hololens

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

Source: N. Snavely

Application: Simultaneous localization and mapping (SLAM)



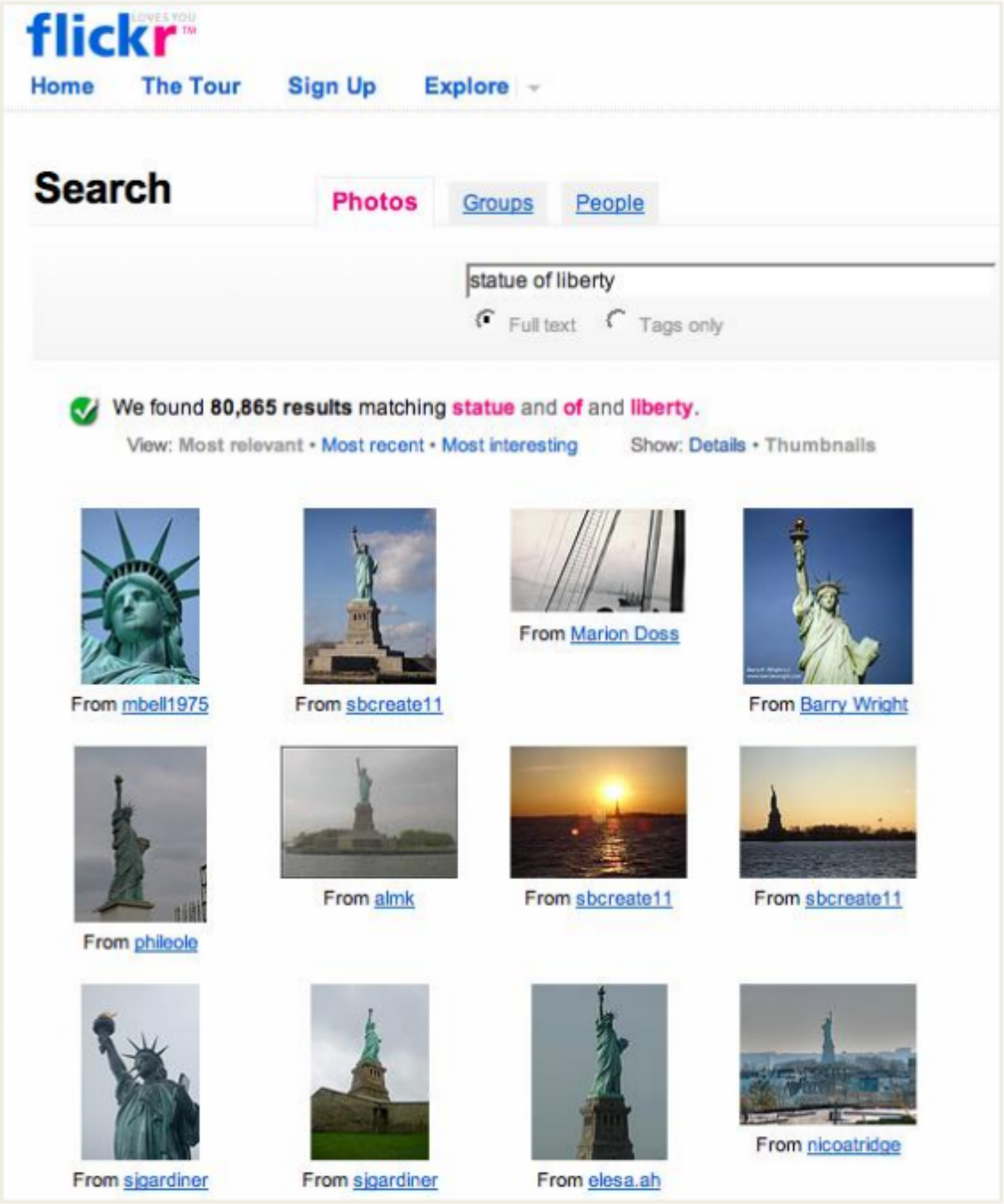
Scape: Building the ‘AR Cloud’: Part Three —3D Maps, the Digital Scaffolding of the 21st Century
https://medium.com/scape-technologies/building-the-ar-cloud-part-three-3d-maps-the-digital-scaffolding-of-the-21st-century-465fa55782dd

Source: N. Snavely

Today

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

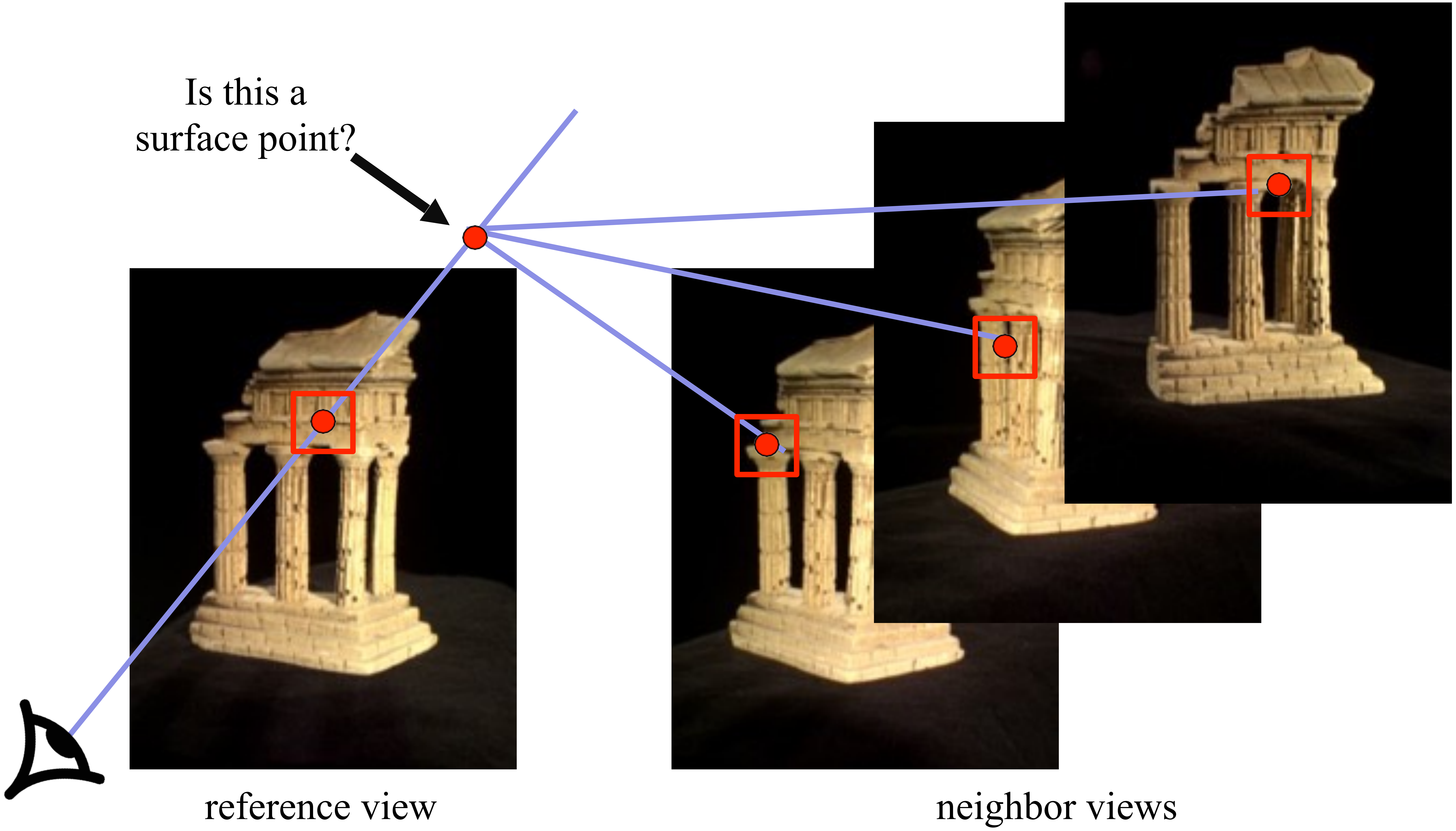
Multi-view stereo



Can we estimate depth, now that we have pose?

Source: N. Snavely

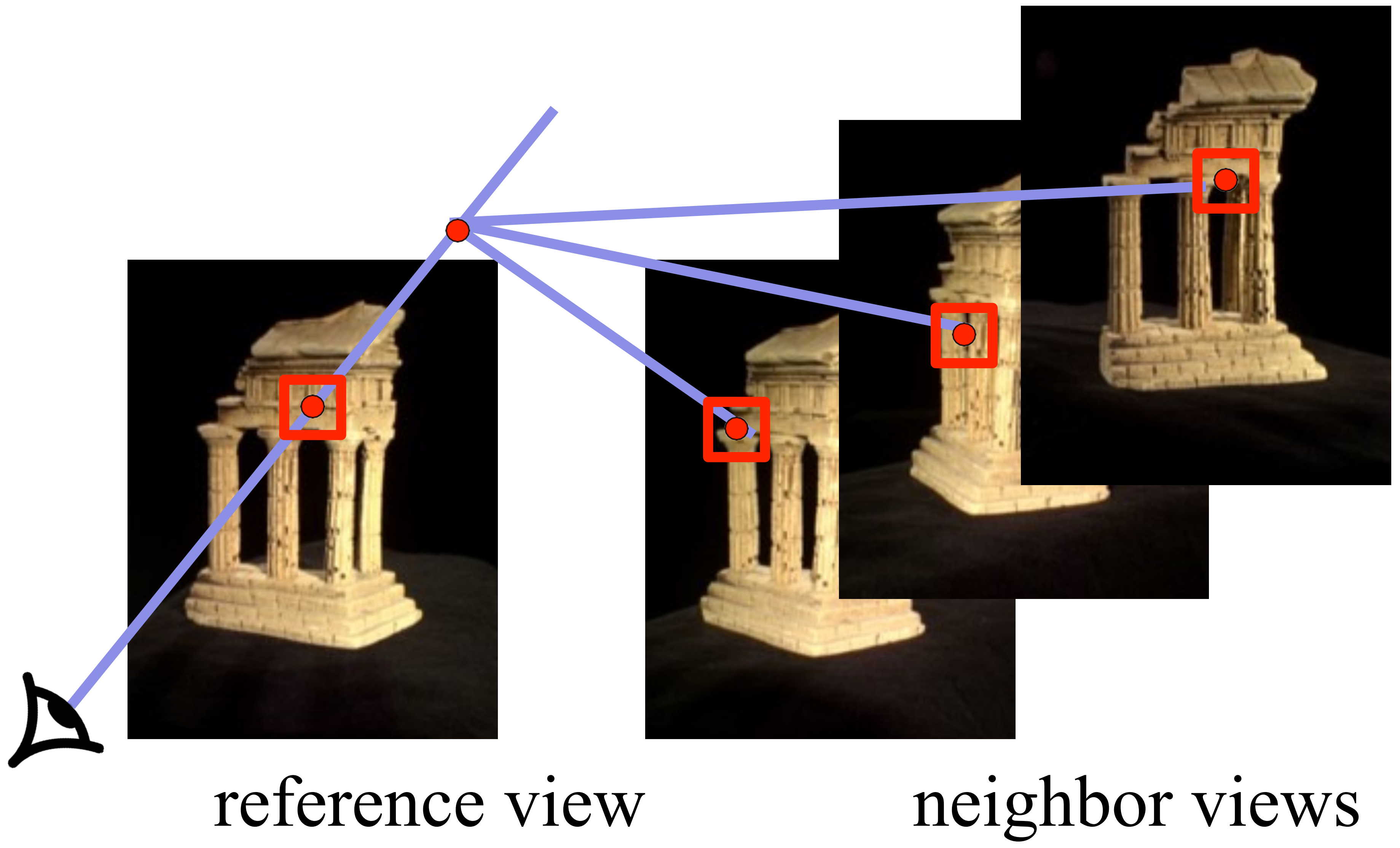
Multi-view stereo

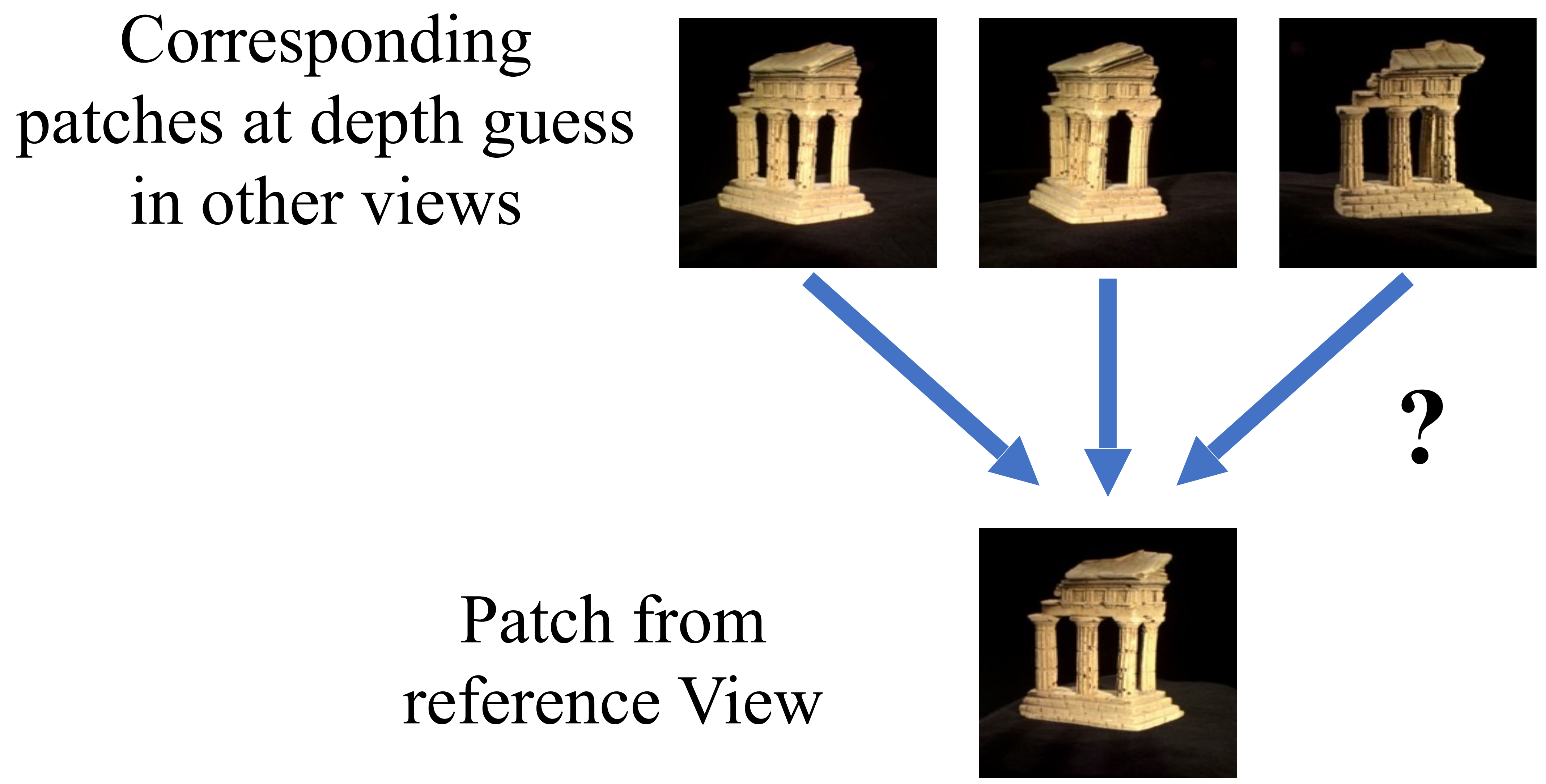


Source: Y. Furukawa

Multi-view stereo

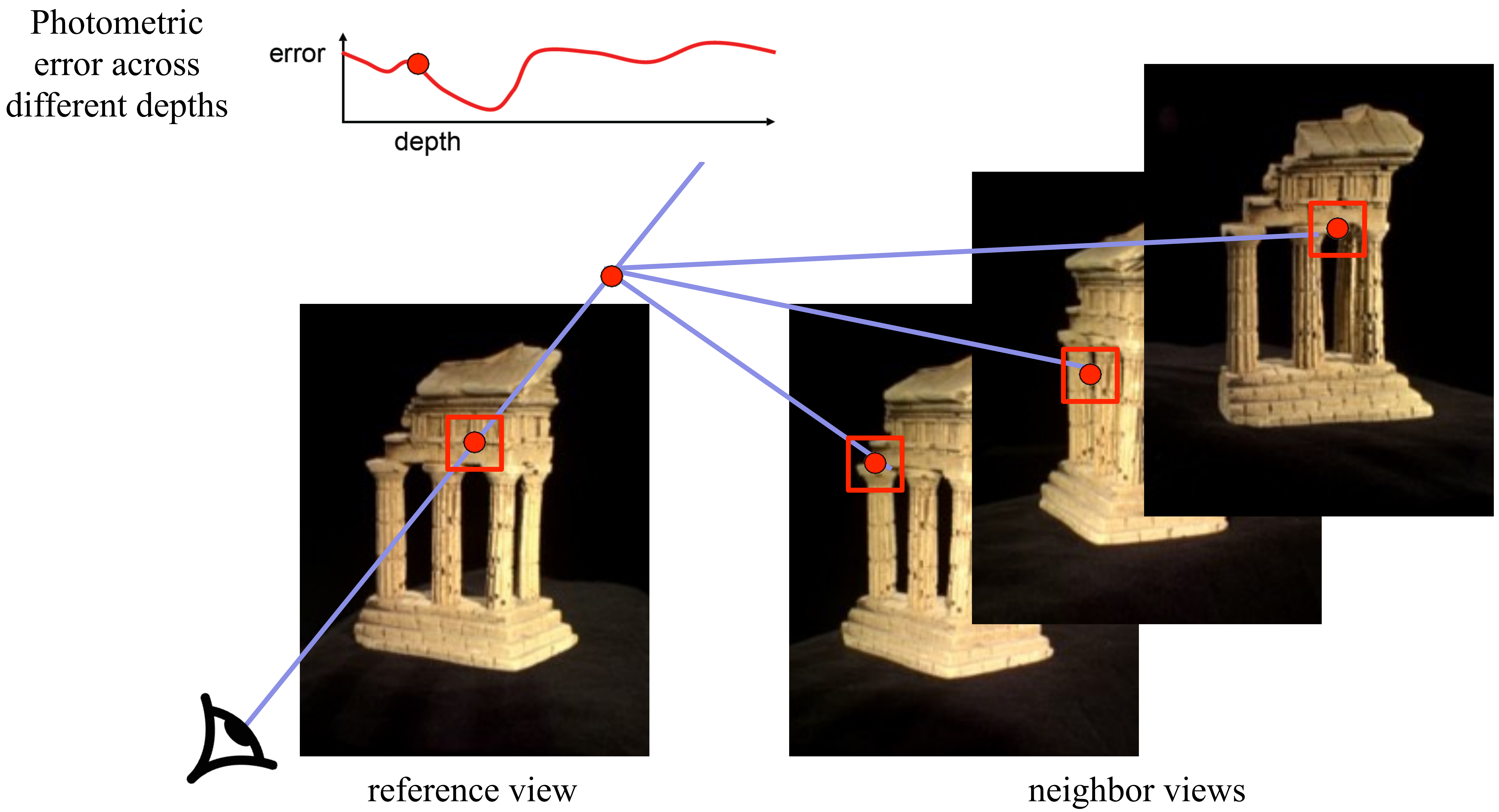
Evaluate the likelihood of a particular depth for a particular reference patch:





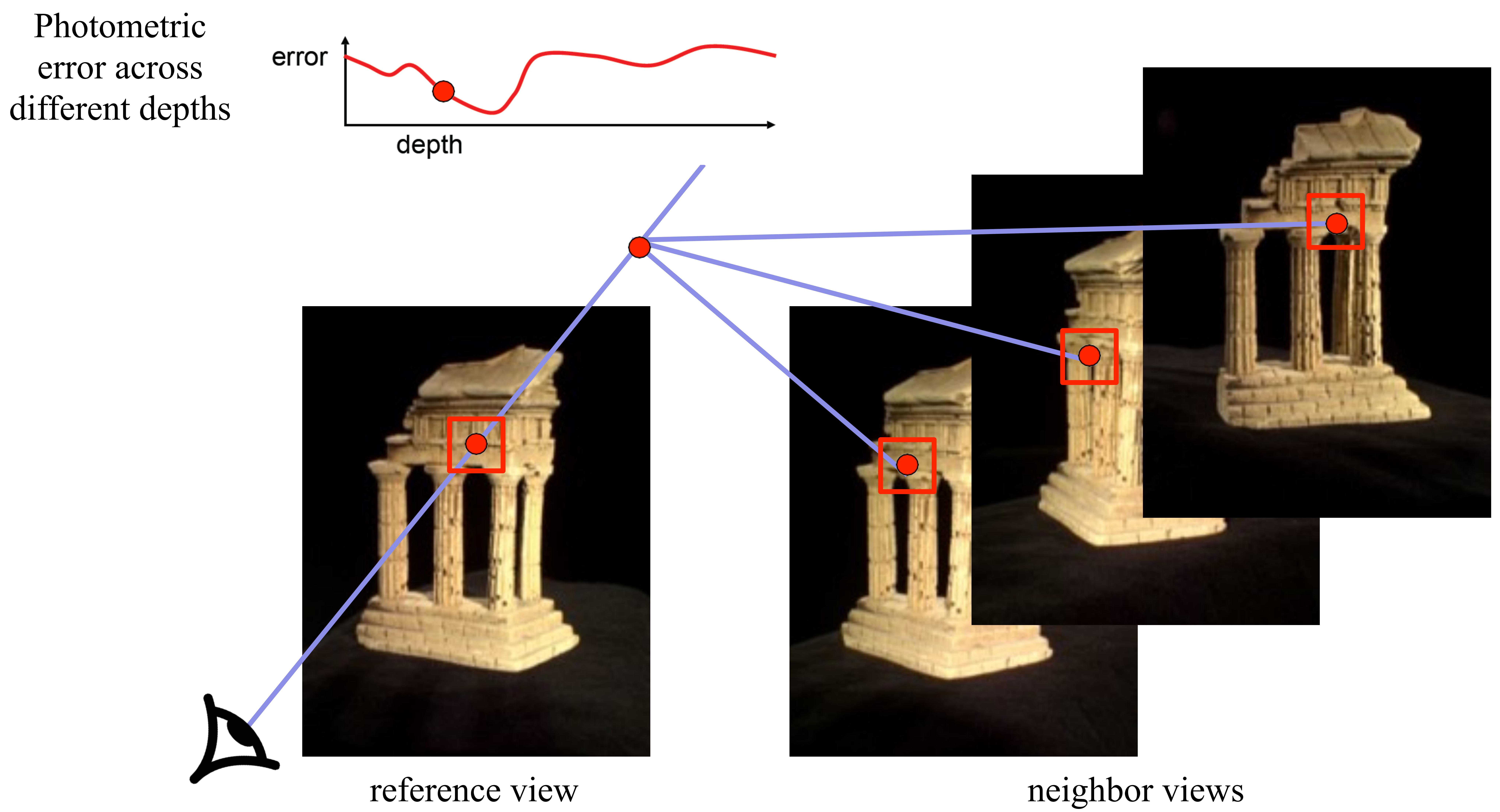
Source: Y. Furukawa

Multi-view stereo



Source: Y. Furukawa

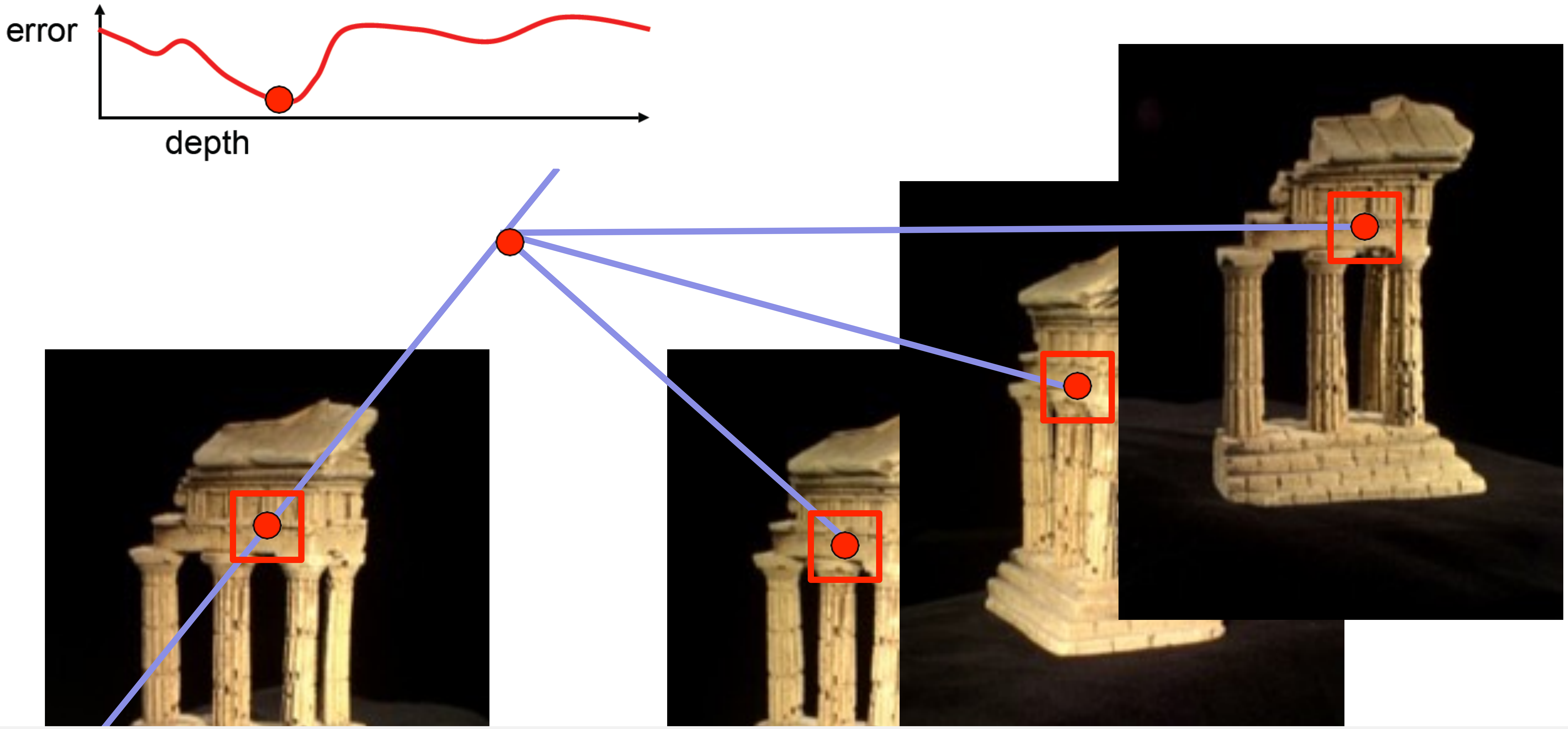
Multi-view stereo



Source: Y. Furukawa

Multi-view stereo

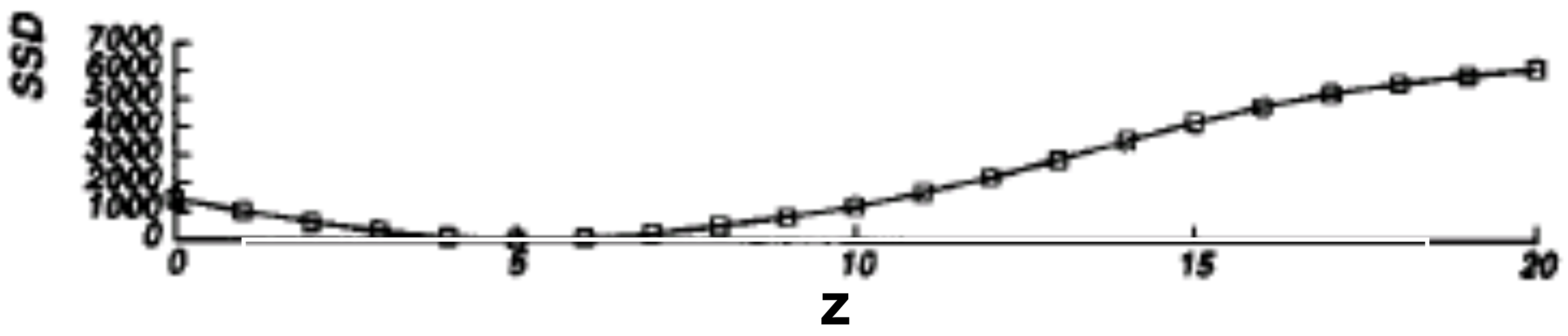
Photometric error across different depths



Solve for a depth map over the whole reference view

Source: Y. Furukawa

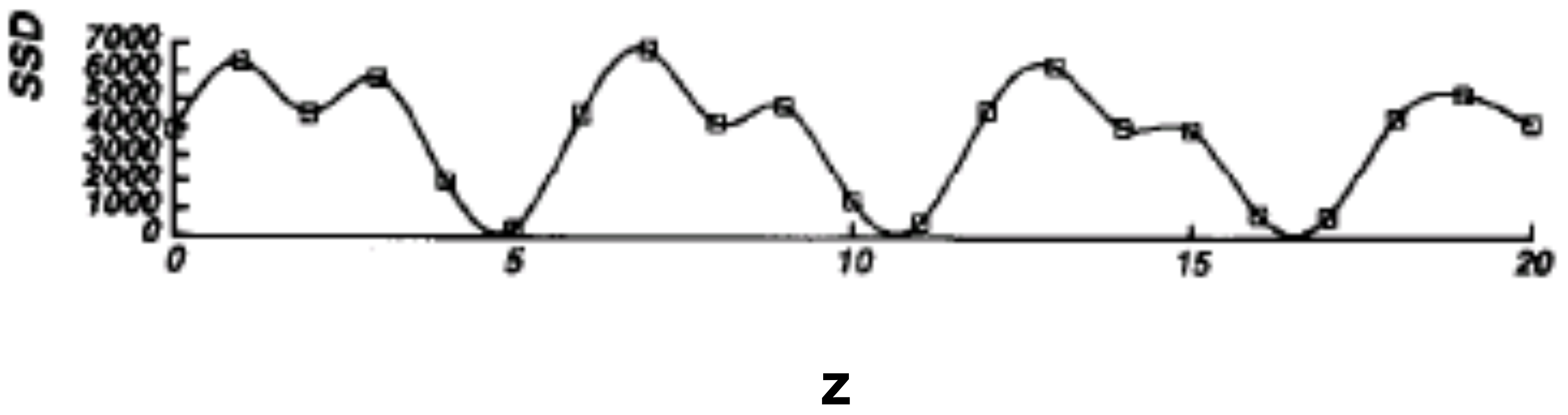
Multiple-baseline stereo



pixel matching score

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

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

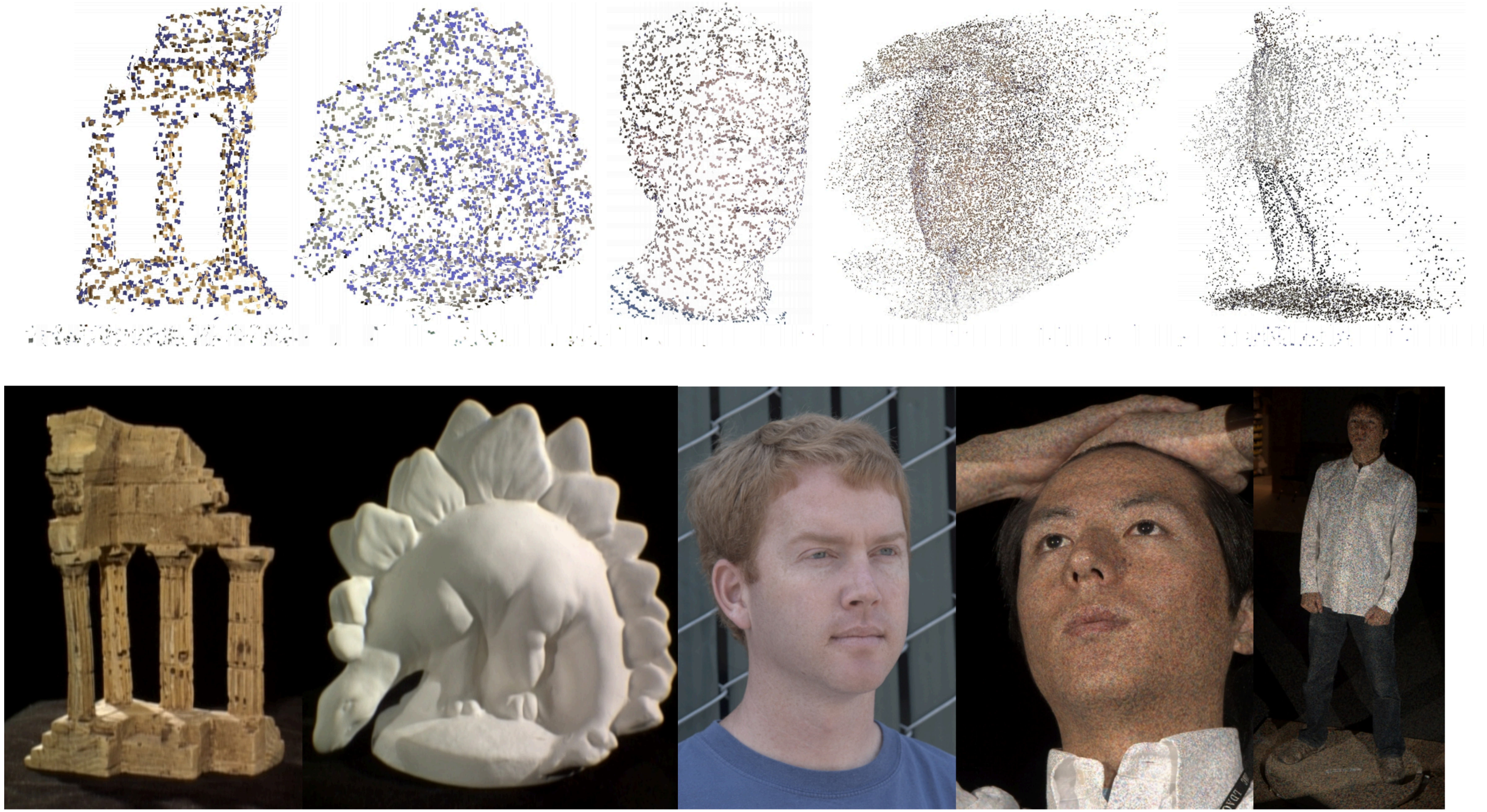


- For larger baselines, must search larger area in second image

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

Source: N. Snavely

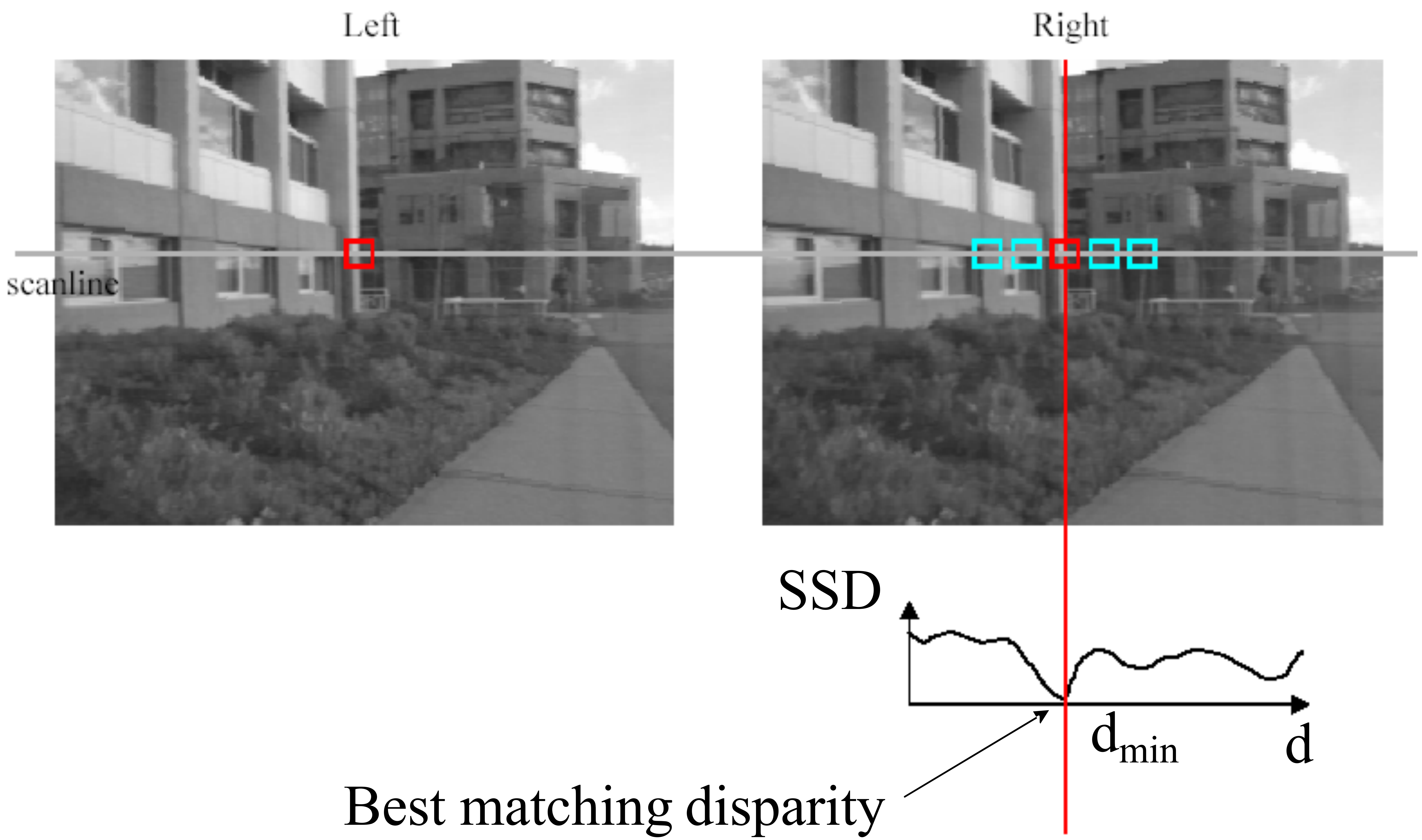
Depth is ambiguous from local information alone!



Today

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

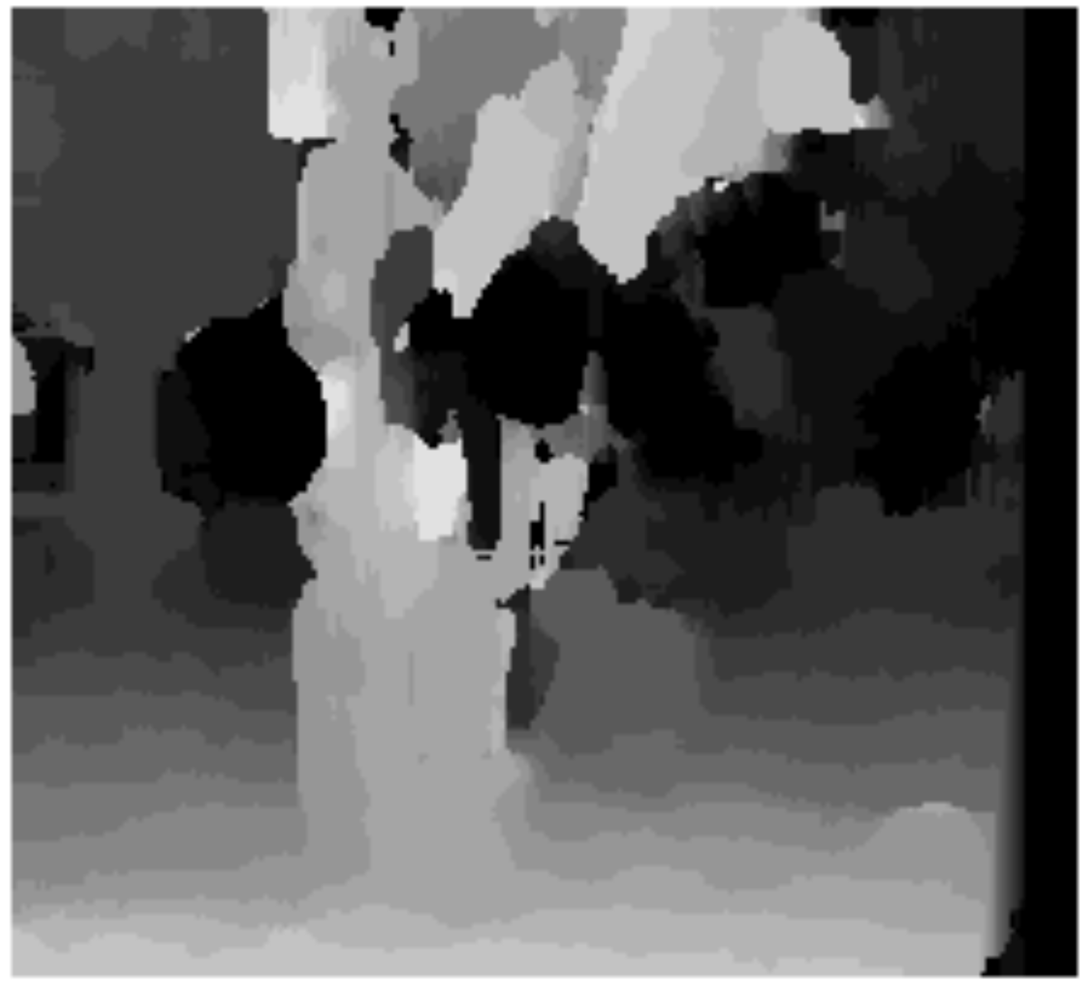
Recall: Stereo matching based on sum-of-squared distance



Source: N. Snavely

Window size

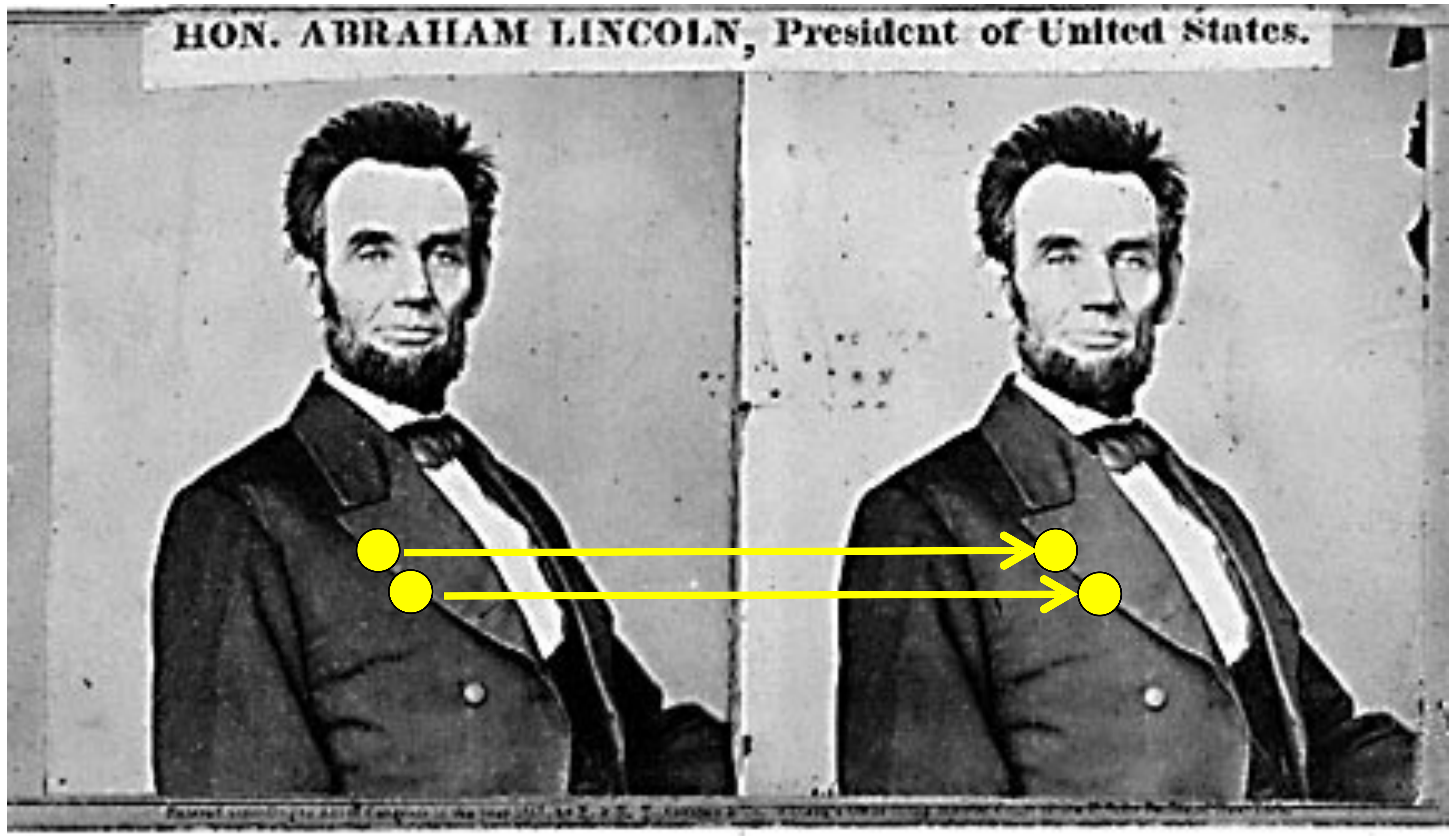
W = 3



W = 20

Source: N. Snavely

Stereo as energy minimization



- What defines a good stereo correspondence?
  1. Match quality
     - Want each pixel to find a good match in the other image
  2. Smoothness
     - If two pixels are adjacent, they should (usually) move about the same amount

Source: N. Snavely

Stereo as energy minimization

- Find disparity map d that minimizes an energy function $E(d)$, where d is disparity.

- Simple pixel / window matching

$$E(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

$C(x, y, d(x, y)) =$ Squared distance between windows I(x, y) and J(x + d(x,y), y)

Source: N. Snavely

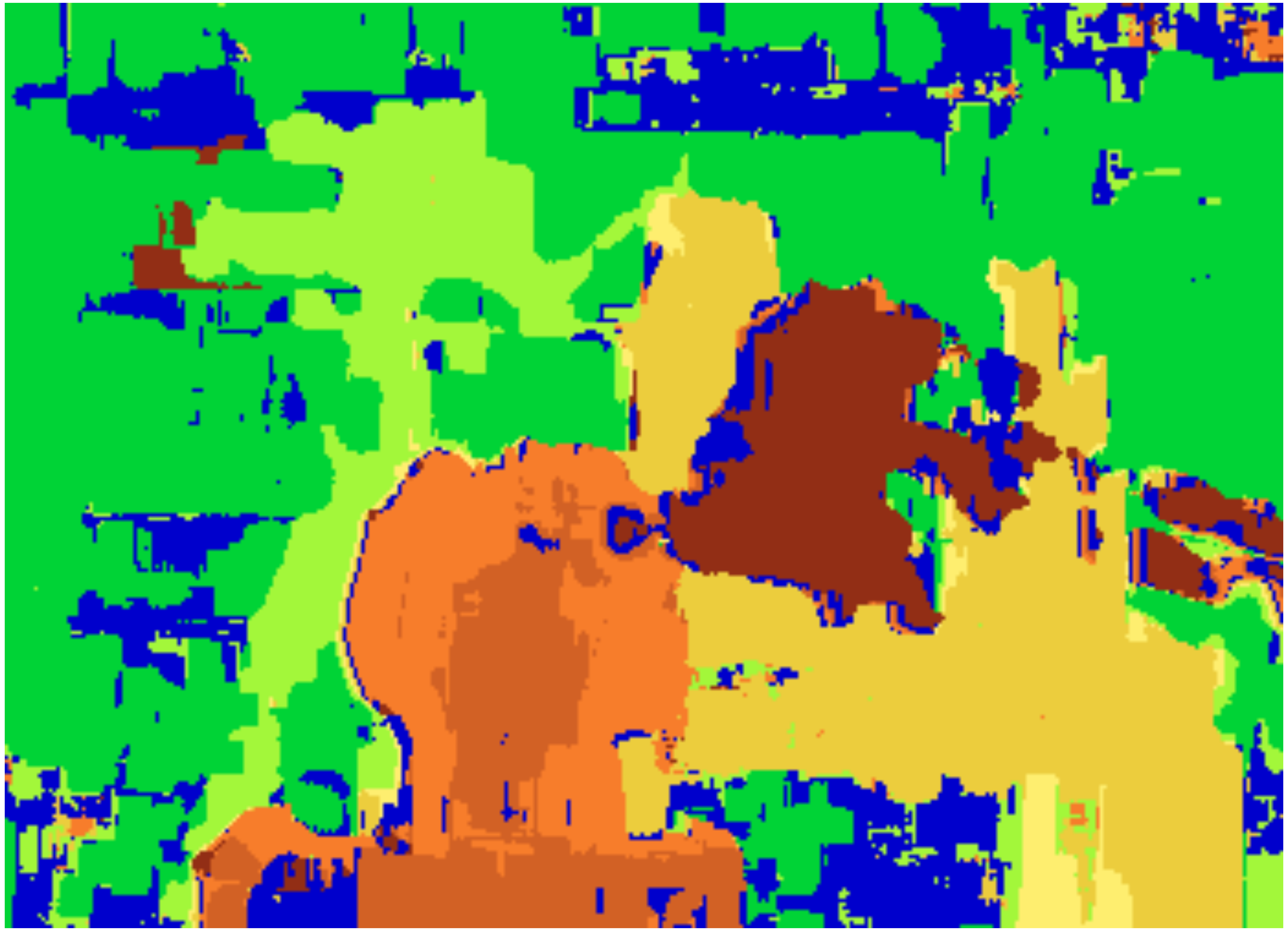
Stereo as energy minimization

Simple pixel / window matching: choose the minimum of each pixel independently

$$d(x,y) = \arg\min_{d'} \; C(x,y,d')$$

Source: N. Snavely

Greedy selection of best match



Source: N. Snavely

Stereo as energy minimization

- Better objective function

$$E(d) = \underbrace{E_d(d)}_{\text{match cost}} + \lambda \underbrace{E_s(d)}_{\text{smoothness cost}}$$

Want each pixel to find a good match in the other image

Adjacent pixels should (usually) move about the same amount

Source: N. Snavely

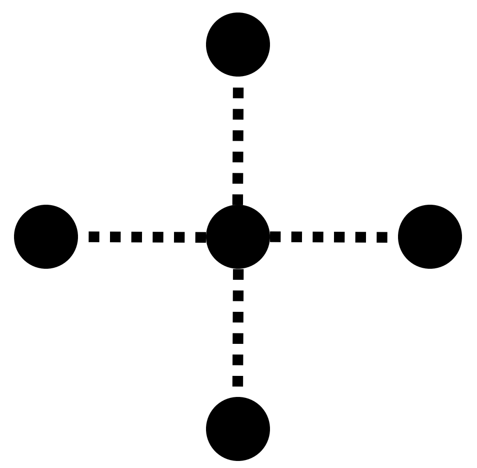
Stereo as energy minimization

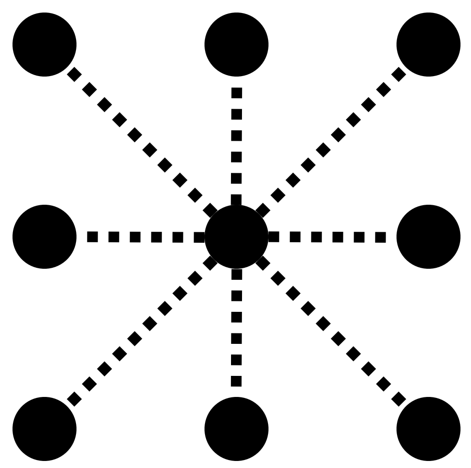
$$E(d) = E_d(d) + \lambda E_s(d)$$

match cost: $$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

smoothness cost: $$E_s(d) = \sum_{(p,q)\in \mathcal{E}} V(d_p, d_q)$$

$\mathcal{E}$: set of neighboring pixels


4-connected neighborhood


8-connected neighborhood

Source: N. Snavely

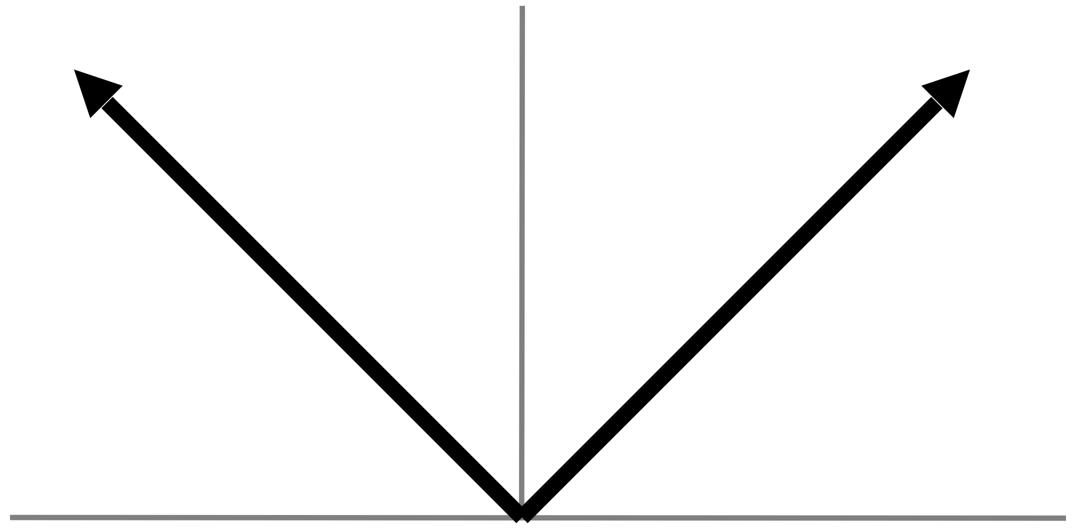
Smoothness cost

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

How do we choose *V*?

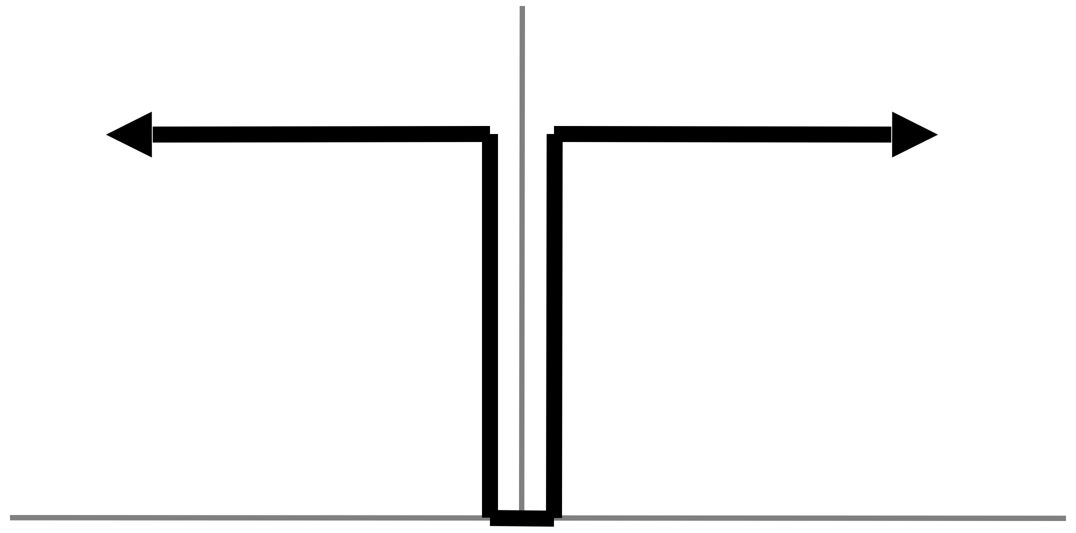
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

"Potts model"

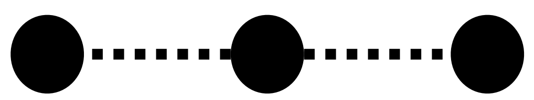


Source: N. Snavely

Dynamic programming

$$E(d) = E_d(d) + \lambda E_s(d)$$

- Can minimize this independently per row scanline using dynamic programming (DP)



- Basic idea: incrementally build a table of costs D one column at a time

$D(x, y, i)$: minimum cost of solution such that d(x,y) = i

Base case: $D(0, y, i) = C(0, y, i), i = 0, \ldots, L$ (L = max disparity)

Recurrence: $D(x, y, i) = C(x, y, i) + \min_{j \in \{0,1,\ldots,L\}} D(x-1, y, j) + \lambda |i - j|$

Source: N. Snavely

Dynamic programming

- Finds “smooth”, low-cost path through cost volume from left to right

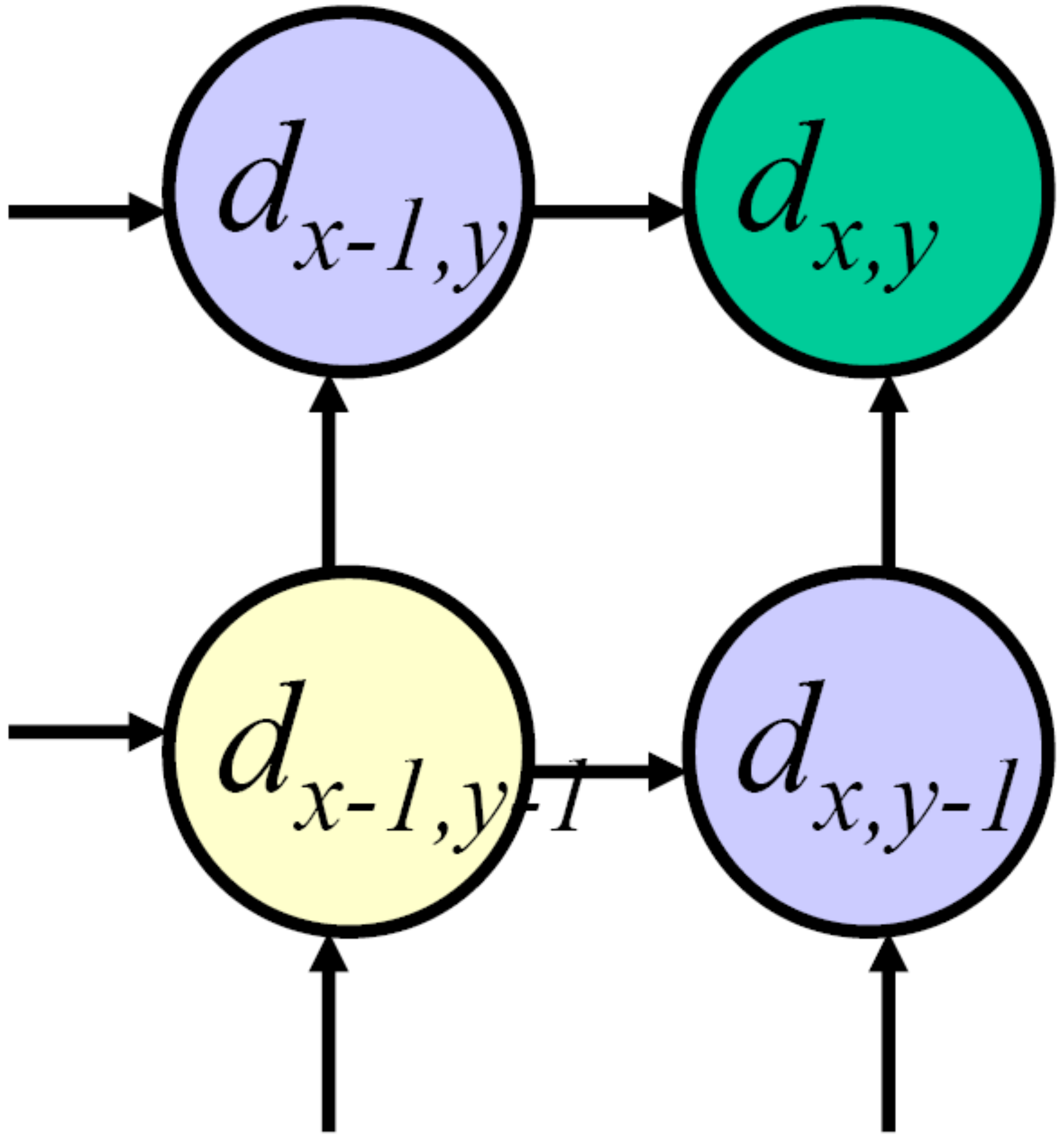
Source: N. Snavely

Dynamic programming

Source: N. Snavely

Dynamic programming

- Can we apply this trick in 2D as well?



- No: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

Source: D. Huttenlocher

Stereo as a minimization problem

$$E(d) = E_d(d) + \lambda E_s(d)$$

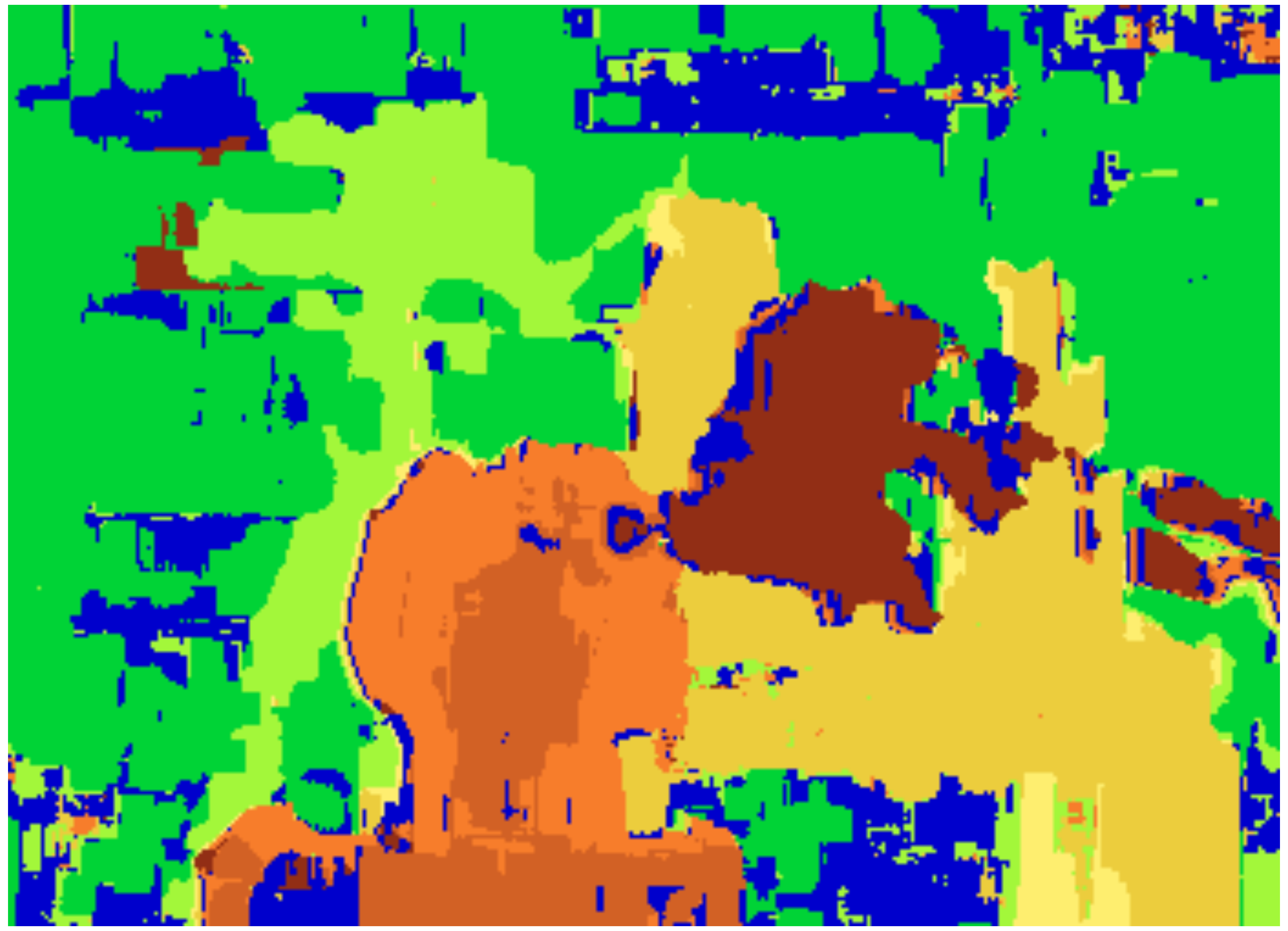
- The 2D problem has many local minima
  - Famous problem: doing inference in a Markov Random Field (MRF)
  - Gradient descent doesn't work well

- And a large search space
  – n x m image w/ k disparities has k^{nm} possible solutions
  – Finding the global minimum is NP-hard in general

- Good approximations exist (see Szeliski textbook):
  - Graph cuts
  - Belief propagation

Source: N. Snavely

Better methods exist...



Greedy selection

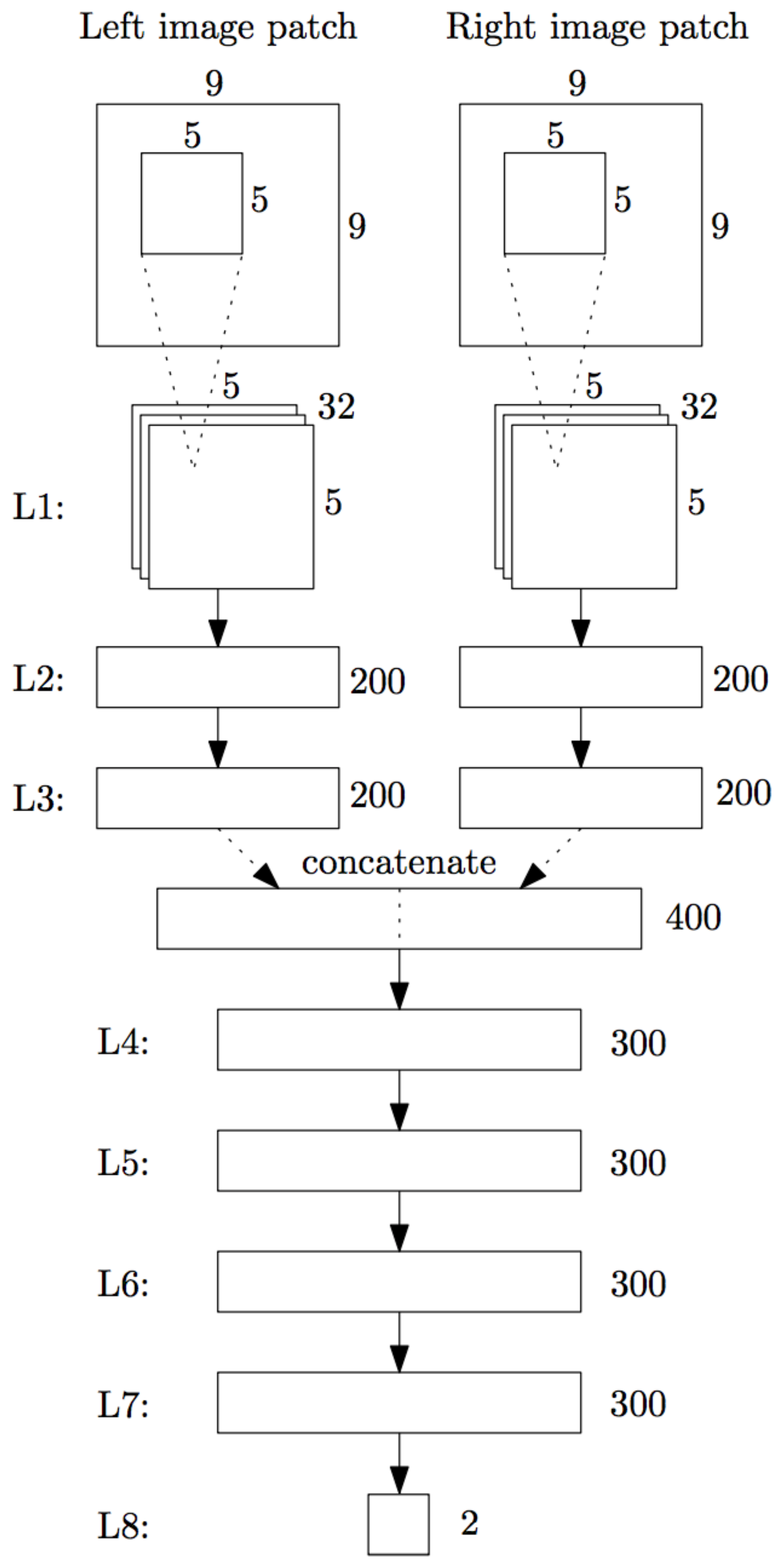
Graph cuts model

Ground truth

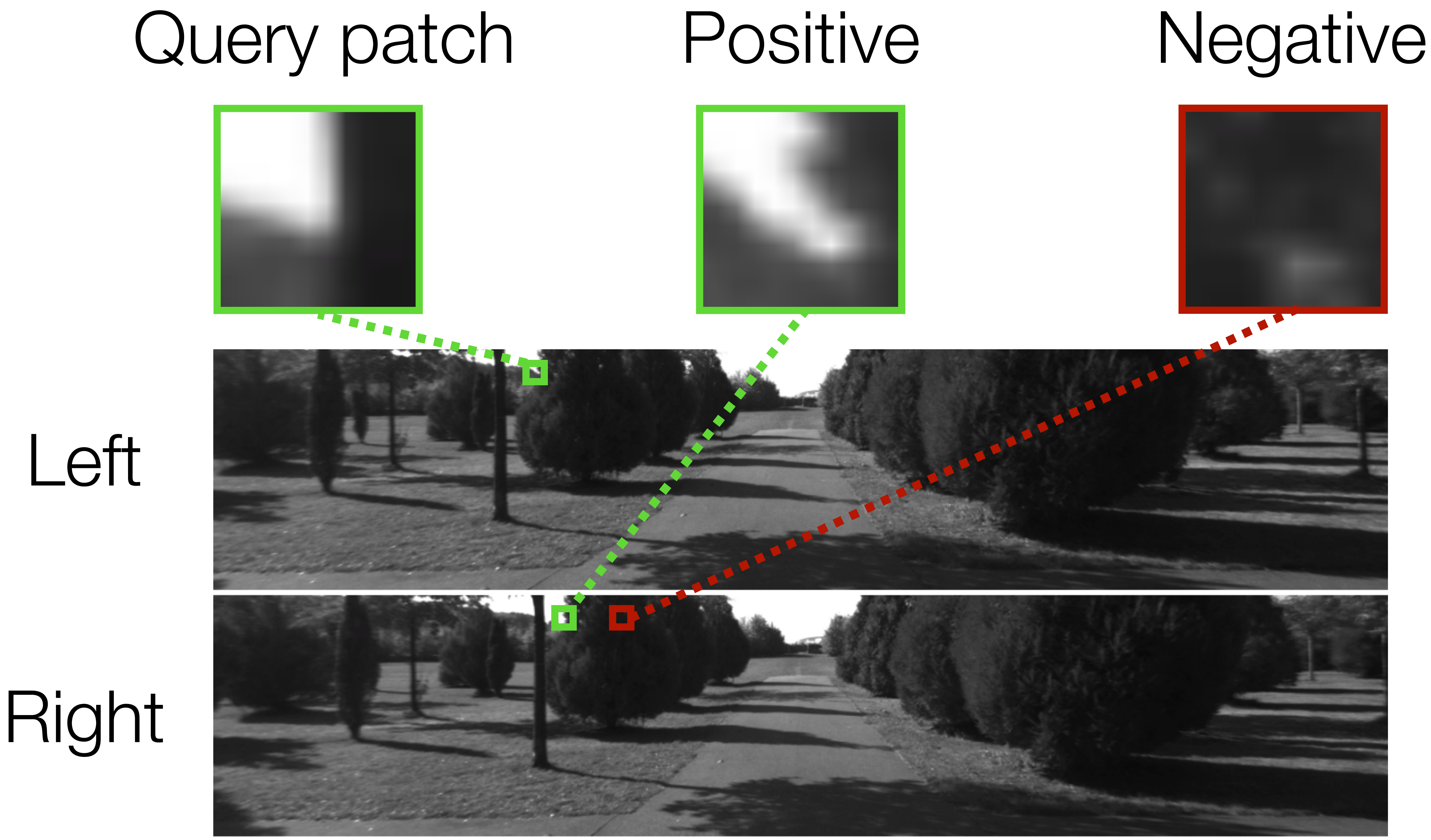
Boykov et al., [Fast Approximate Energy Minimization via Graph Cuts](), International Conference on Computer Vision, September 1999.

Source: N. Snavely

Deep learning + MRF refinement



[Zbontar & LeCun, 2015]



CNN-based matching + refinement

Next lecture: Color, lighting, and shading