CSCI 497P/597P: Computer Vision

Lecture 25

Epipolar Geometry and the Fundamental Matrix

Structure From Motion

Multiview Stereo

Announcements

- P3 - partner up by the end of today!
  - Piazza feature for finding a partner, or ask on Discord
- HW4 is out
  - Due next Friday

Goals

- Understand the derivation of the essential matrix and the fundamental matrix.
- Know some properties of 2-camera epipolar geometry and the fundamental matrix:
    - rank deficiency
    - epipolar lines; epipoles
- Understand the Structure From Motion problem and the general idea behind how it is solved.

To the notes!

Two questions:

- We derived F assuming that **K**, **R**, and **t** are known.
- Can we find F without them?
- Can we find **K**, **R**, and **t** if we have **F**?

8-point algorithm

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

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

Let $x=(u,v,1)^T$ and $x'=(u',v',1)^T$,

Each match yields **one** equation:

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-point algorithm

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

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

Let x=$(u,v,1)^{\mathrm{T}}$ and x'=$(u',v',1)^{\mathrm{T}}$,

Each match yields

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

equation:

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

As with homographies, this is has the form Ax = 0

Solve homogeneous system using the SVD.

8-point algorithm: Problem

- **Solution is (generally) not rank 2.**
- Fix: More SVD!

8-point algorithm

- **Solution is (generally) not rank 2.**
- Fix: More SVD!

8-point algorithm: Problem 2

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

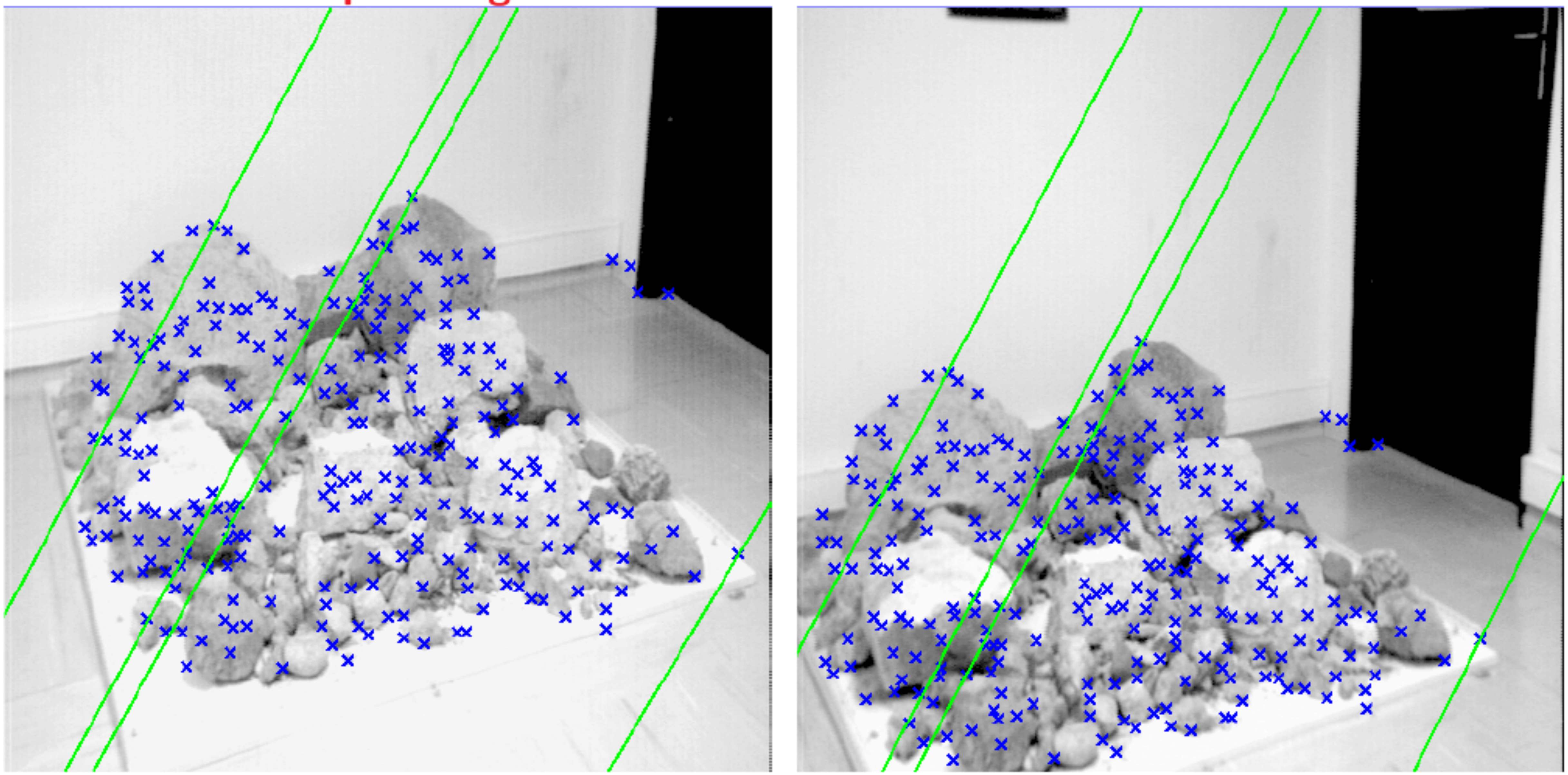
~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

! Orders of magnitude difference
between column of data matrix
→ least-squares yields poor results

Fix: scale image positions to the range [0,1], solve, then scale back.

8-point algorithm: Results

- Normalized 8-point algorithm



The Fundamental Matrix Song

Required viewing:

https://www.youtube.com/watch?v=DgGV3l82NTk

What about more than 2 views?

- 2 views: fundamental matrix
- 3 views: trifocal tensor
- 4 views: *quadrifocal* tensor
- more views: ¯_(ツ)_/¯ (it gets complicated...)

Large-scale structure from motion

- https://www.youtube.com/watch?v=sQegEro5Bfo

Structure from Motion

- Given many photos, reconstruct:
  - positions of the cameras
  - positions of 3D points

Chicken/Egg

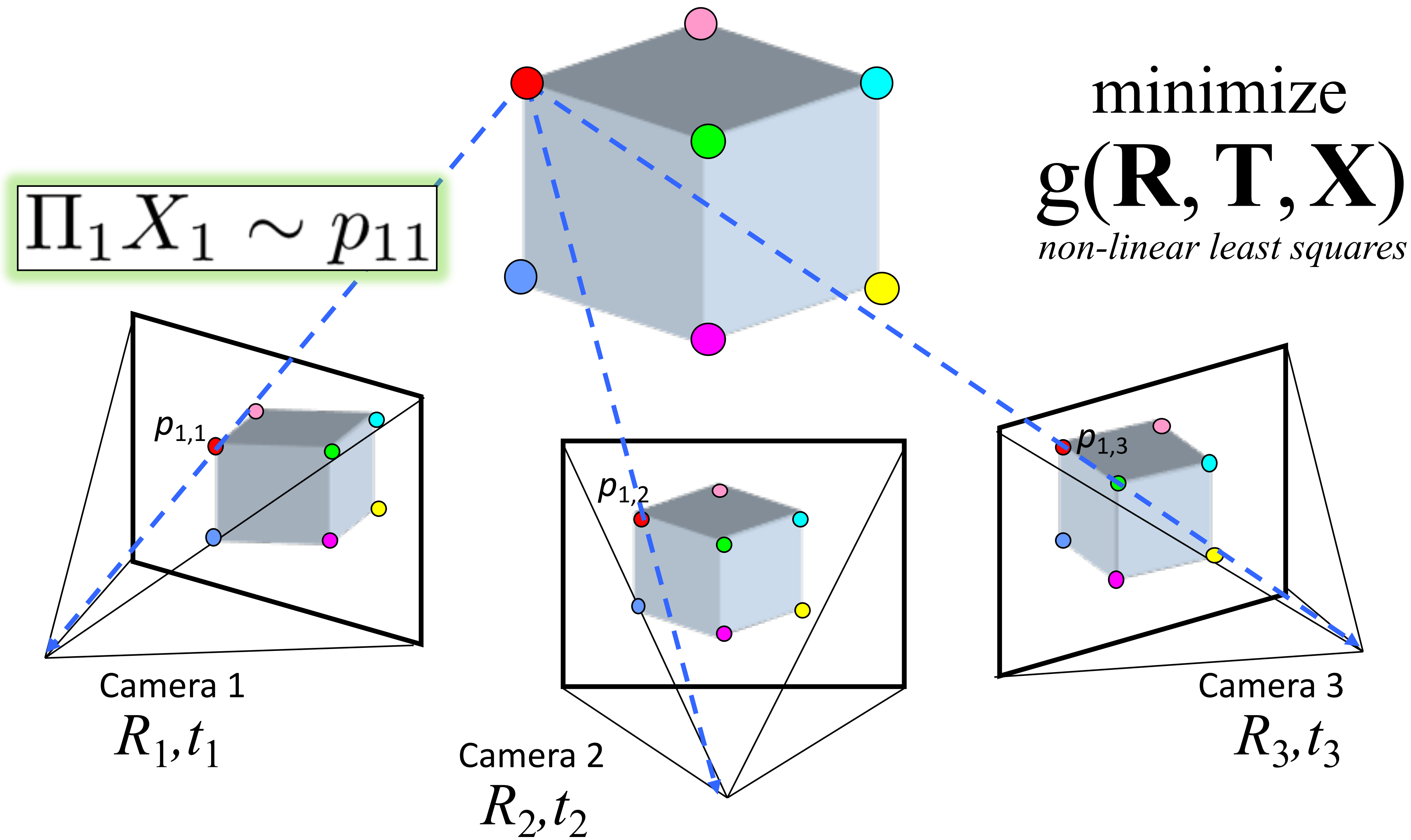
- Step 1: solve for relative pose of pairs (or triples) of cameras using correspondences from feature matching.

- Step 2: alternate between solving:

  - given camera positions, solve for point locations

  - given point locations, solve for camera positions

Structure From Motion



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

indicator variable: is point *i* visible in image *j* ?

predicted image location

observed image location

Applications

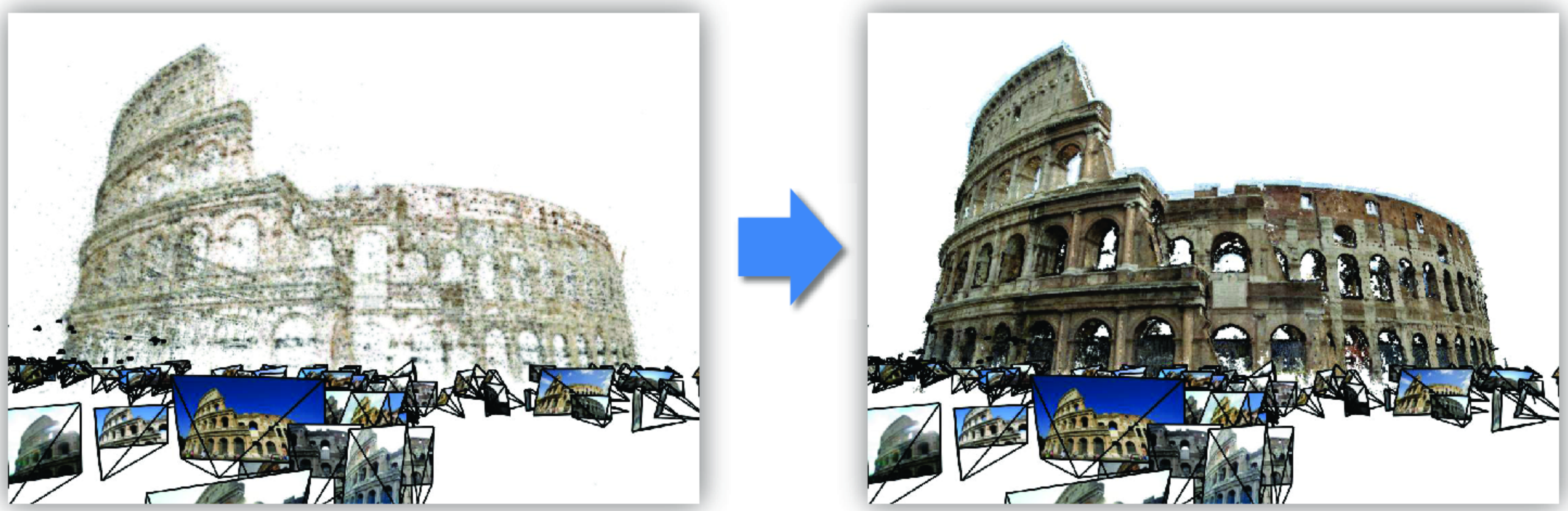
- Hyperlapse https://www.youtube.com/watch?v=SOpwHaQnRSY

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

- Graphics, movies, games, self-driving cars, robots, ...

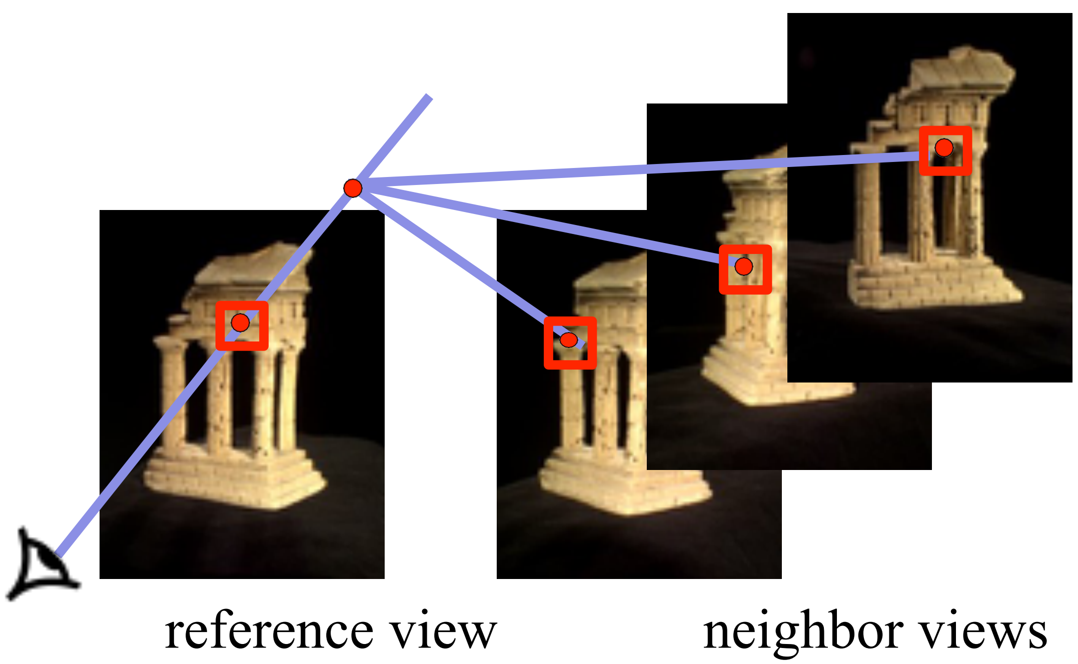
Multiview Stereo

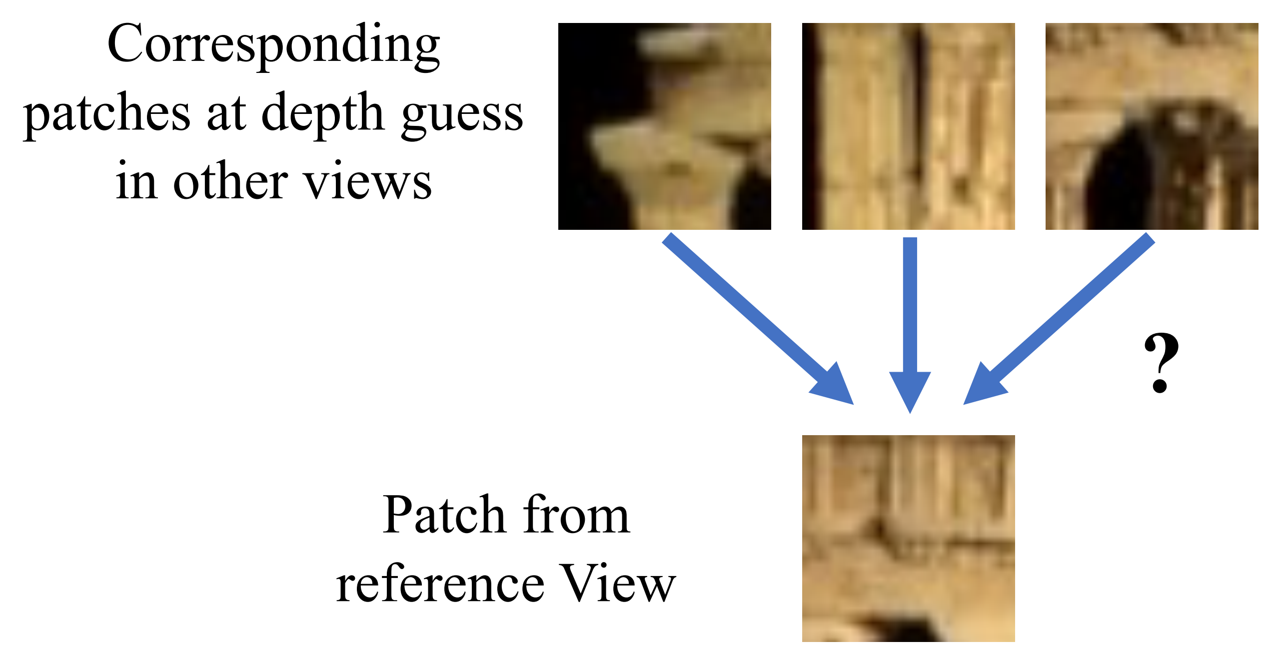
- Once you've solved for all those camera positions, how good a 3D model can you create?



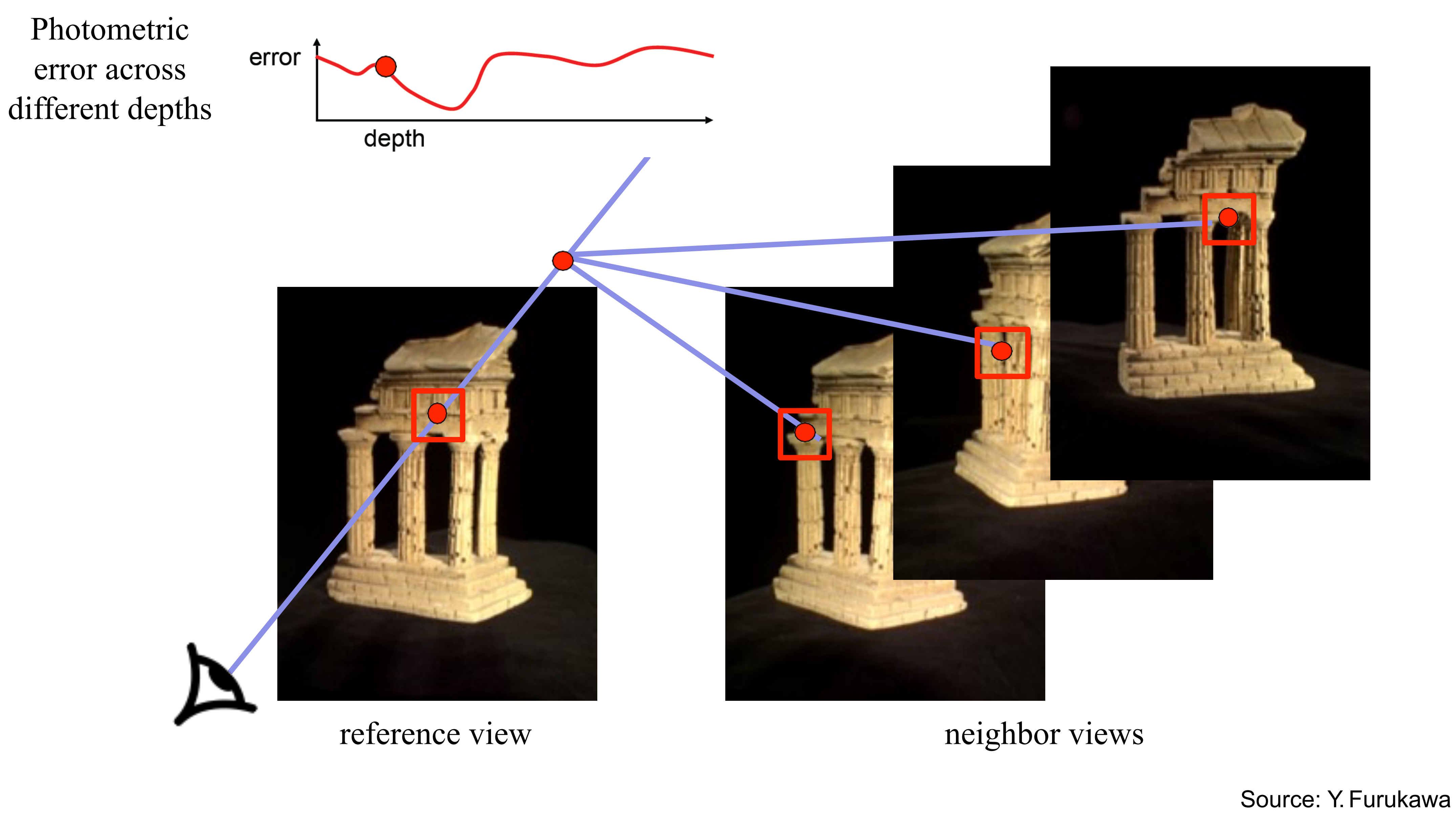
Multiview Stereo: Basic Idea

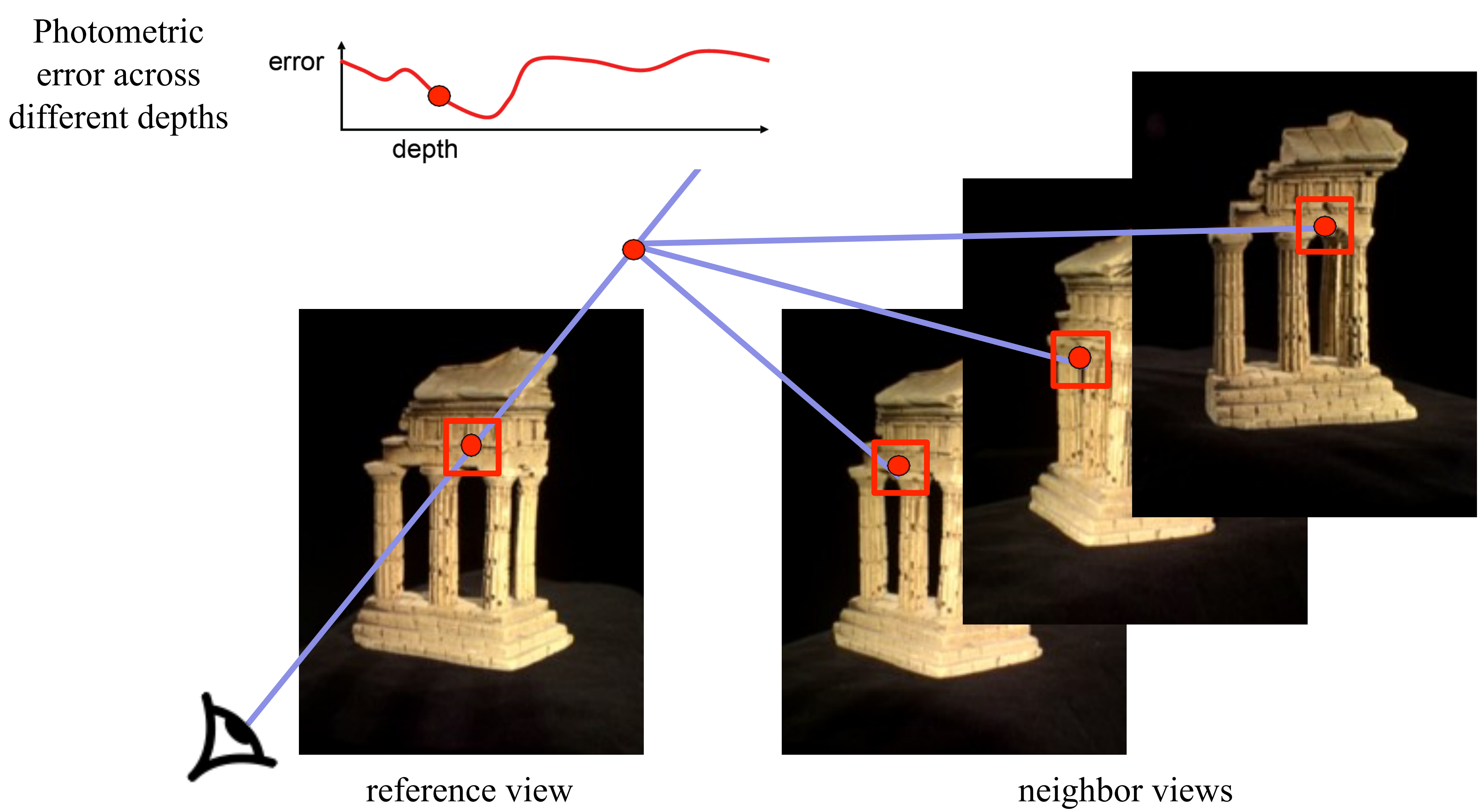
Evaluate the likelihood of geometry at a particular depth for a particular reference patch:





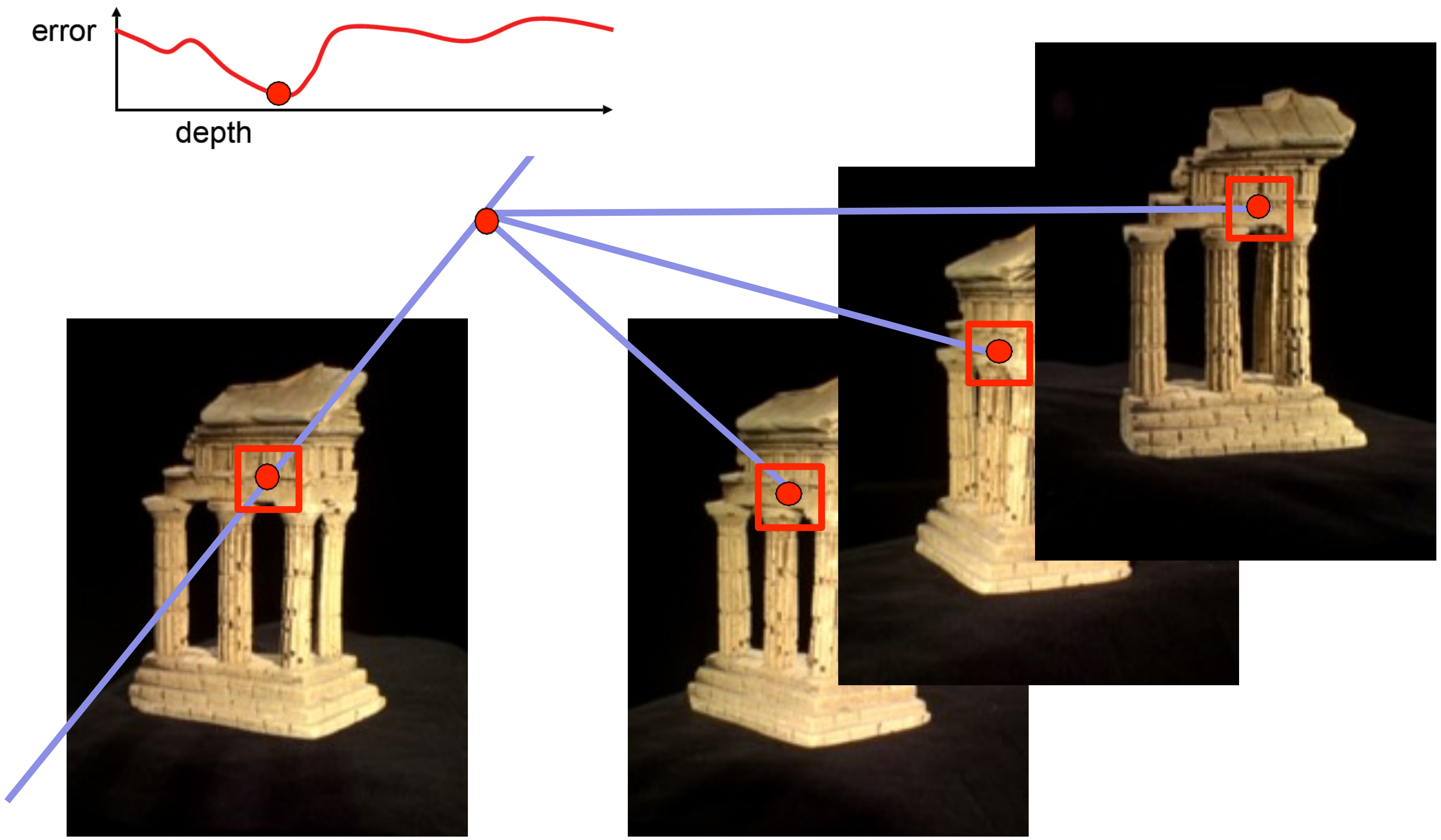
Source: Y. Furukawa


Photometric error across different depths
error
depth
reference view
neighbor views
Source: Y. Furukawa



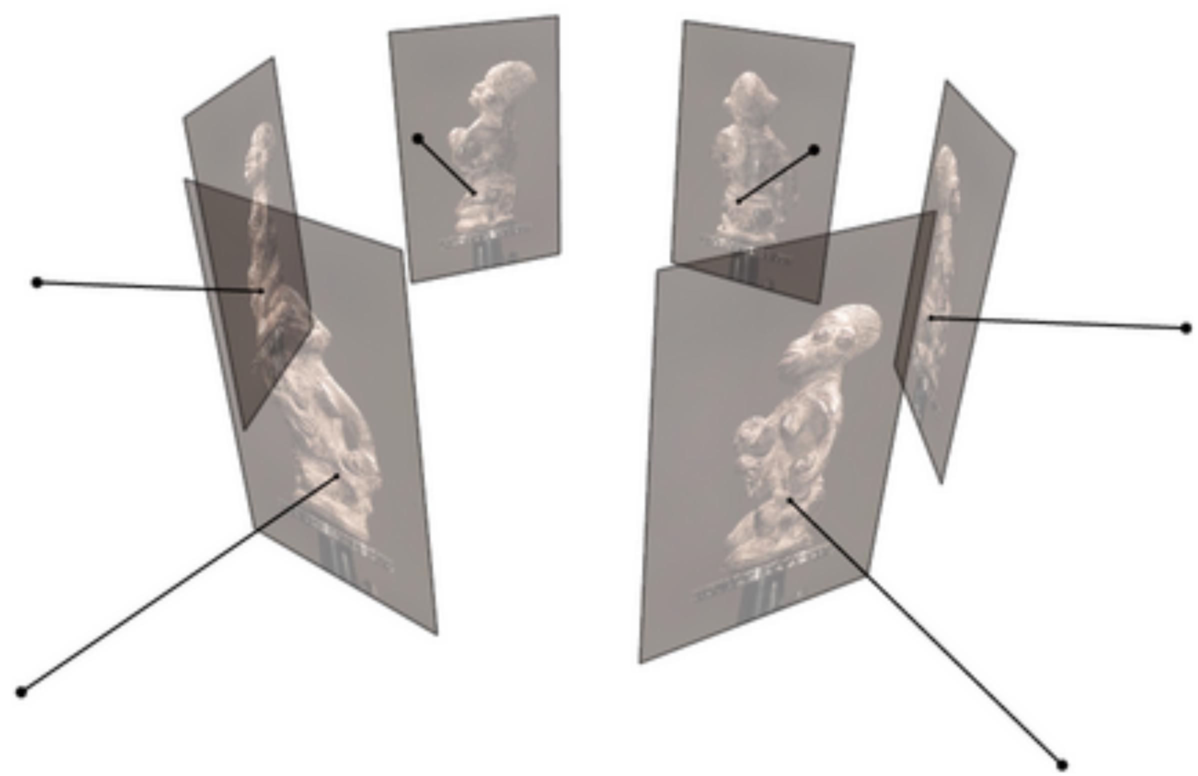
Source: Y. Furukawa

Photometric error across different depths



Depth map fusion

- Compute depth maps for multiple cameras, then fuse them into a 3D model



Figures by Carlos Hernandez

Result

- https://www.youtube.com/watch?v=N6Douyfa7l8
-