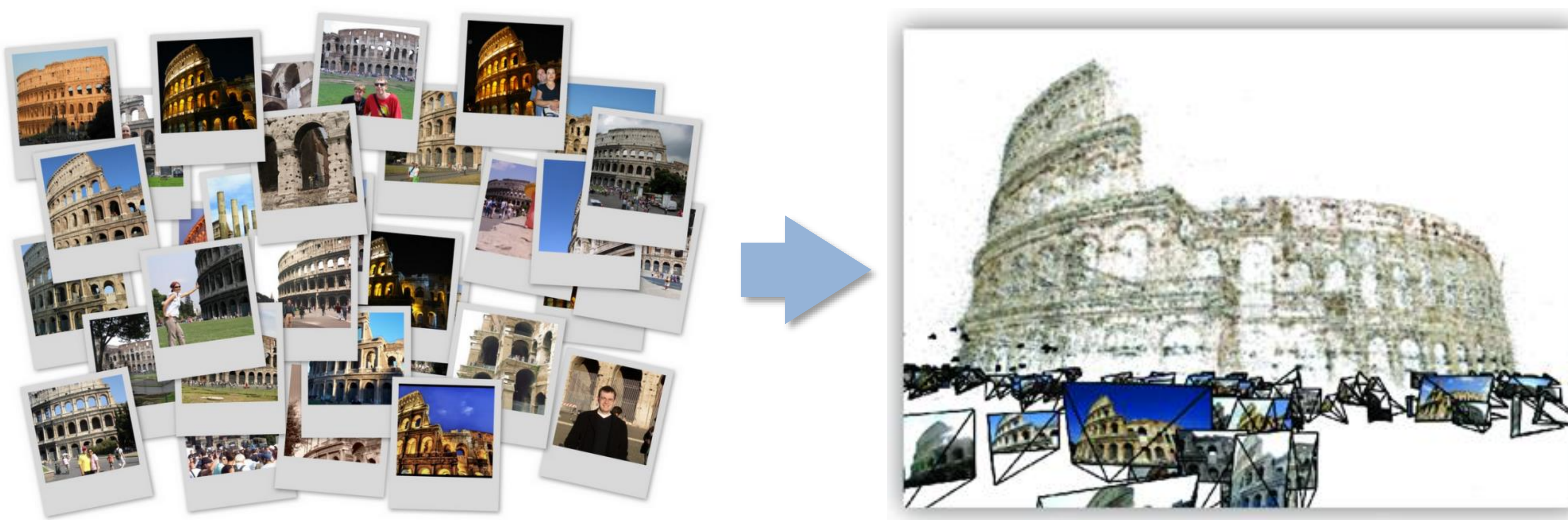
CS5670: Computer Vision

Noah Snavely

Structure from motion



Readings

- Szeliski, Chapter 7.1 – 7.4

Annoncements

- Project 4 due next Monday, 4/22
- Last in-class quiz next Wednesday, 4/24
- Project 5 on deep learning coming up
- Final exam in-class on Monday, 5/6

Project 3 Artifact Prizes

Third Place

Yuxi Sun and Yaohai Xu

Second Place

Aayushi Jain and Keith Yu

First Place

Kulvinder Lotay and Roger Wang

Structure from motion

- Multi-view stereo assumes that cameras are calibrated
  - Extrinsics and intrinsics are known for all views

- How do we compute calibration if we don't know it? In general, this is called *structure from motion*

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

Two views

- Solve for Fundamental matrix / Essential matrix
- Factorize into intrinsics, rotation, and translation

What about more than two views?

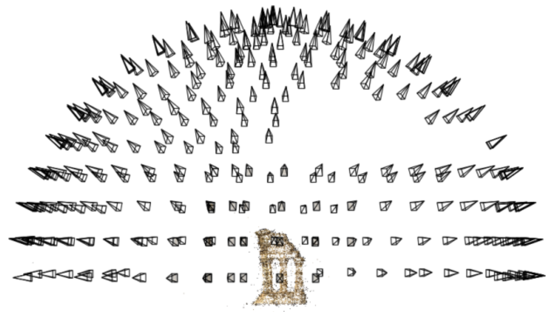
- The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*
- The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*
- After this it starts to get complicated…
  - Instead, we explicitly solve for camera poses and scene geometry

Structure from motion

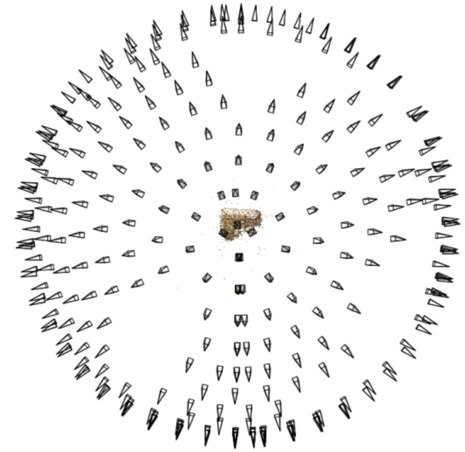
- Given many images, how can we
  a) figure out where they were all taken from?
  b) build a 3D model of the scene?

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

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 $\mathbf{K}_j$

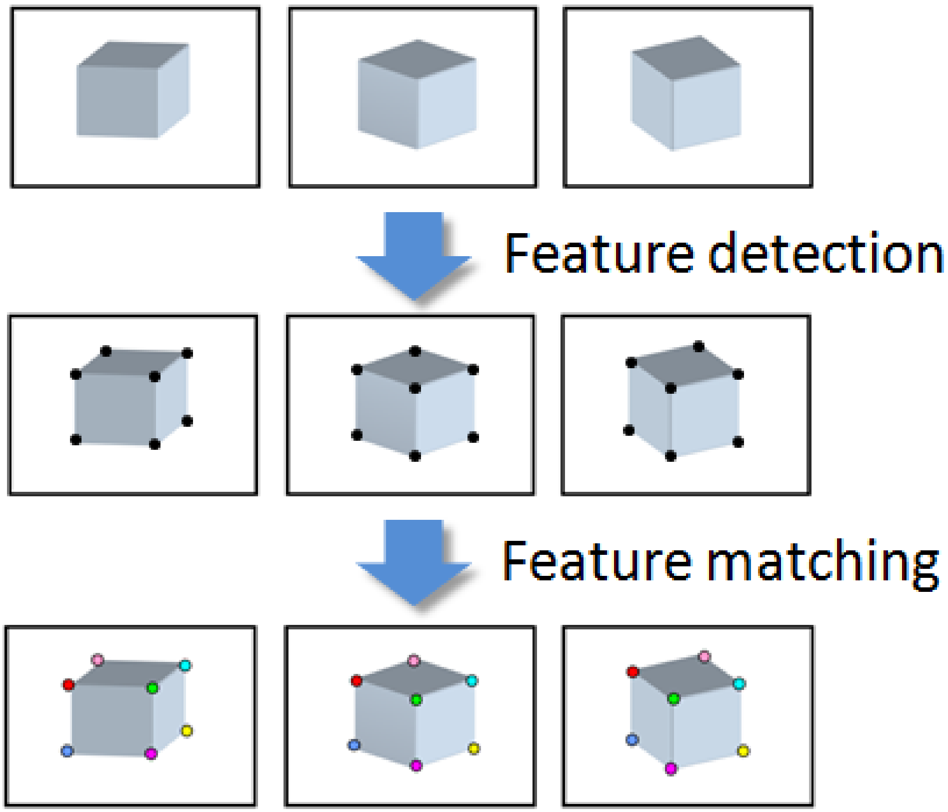
- Objective function: minimize *reprojection error*

Also doable from video

What we've seen so far…

- 2D transformations between images
  - Translations, affine transformations, homographies…
- Fundamental matrices
  - Represent relationships between 2D images in the form of corresponding 2D lines
- **What's new:** Explicitly representing 3D geometry of cameras *and points*

Input



Camera calibration and 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 know camera parameters, 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)

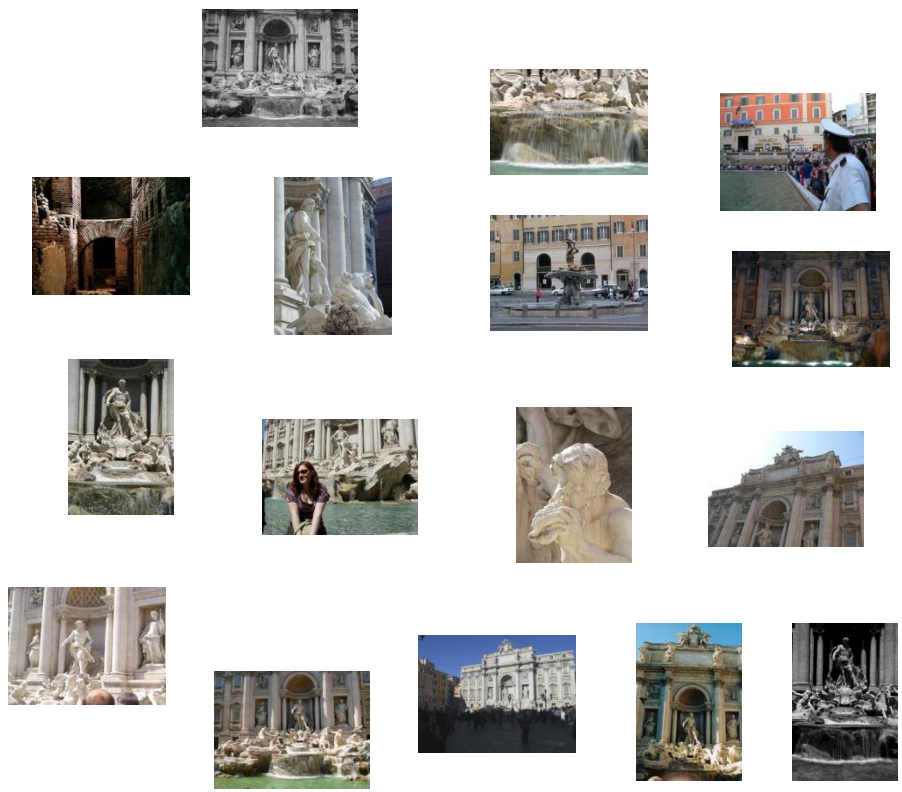
Photo Tourism

First step: how to get correspondence?

- Feature detection and matching

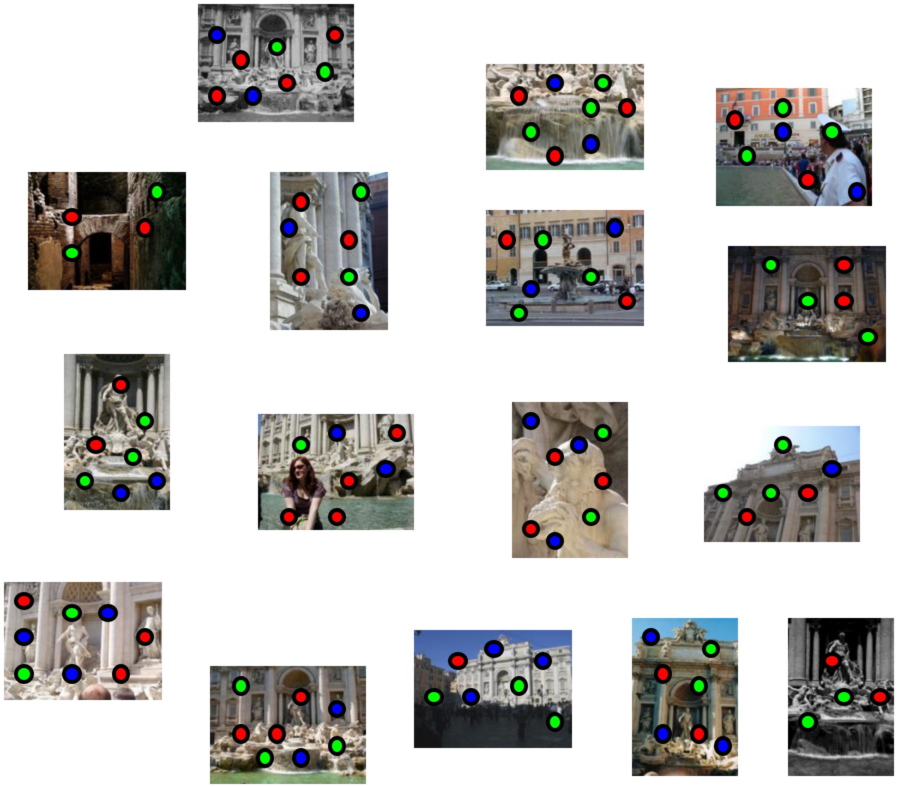
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



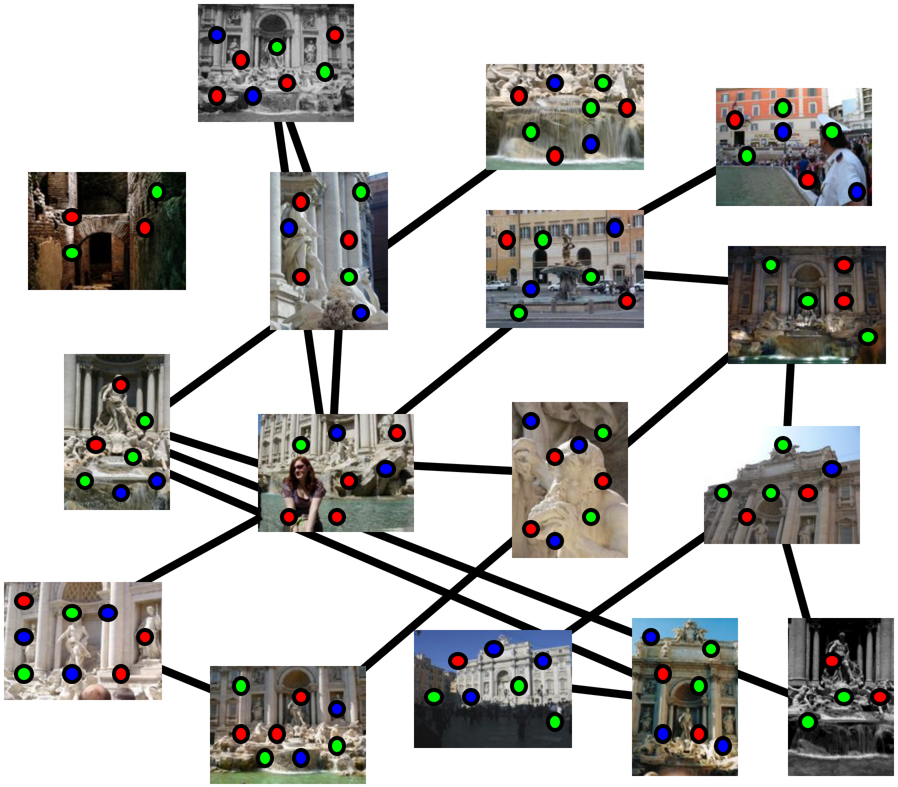
Feature detection

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

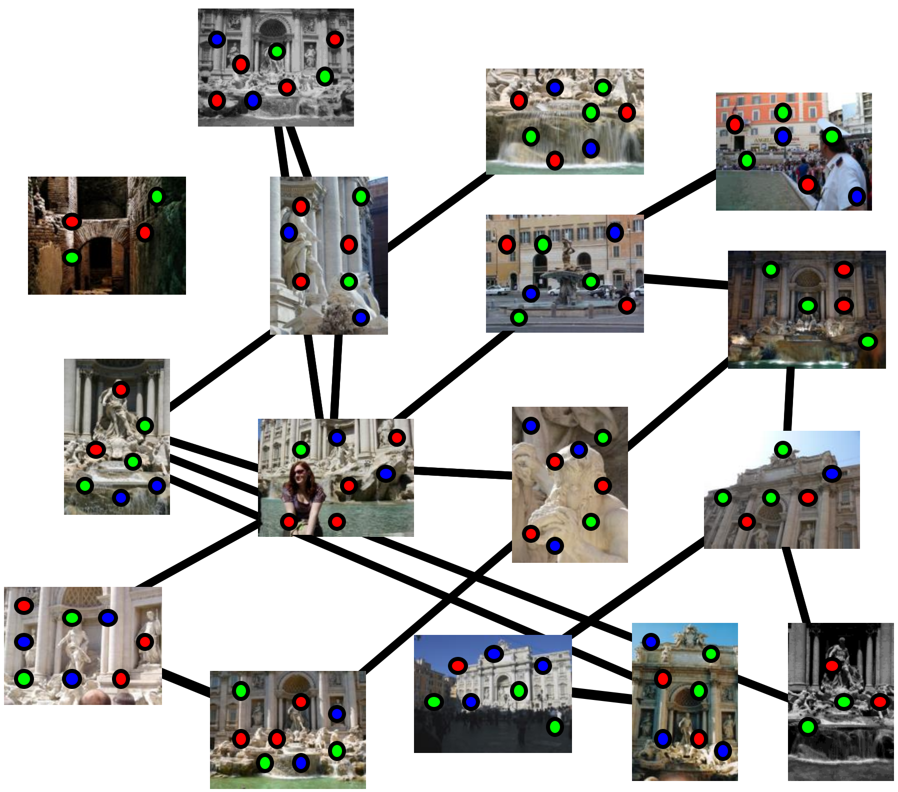
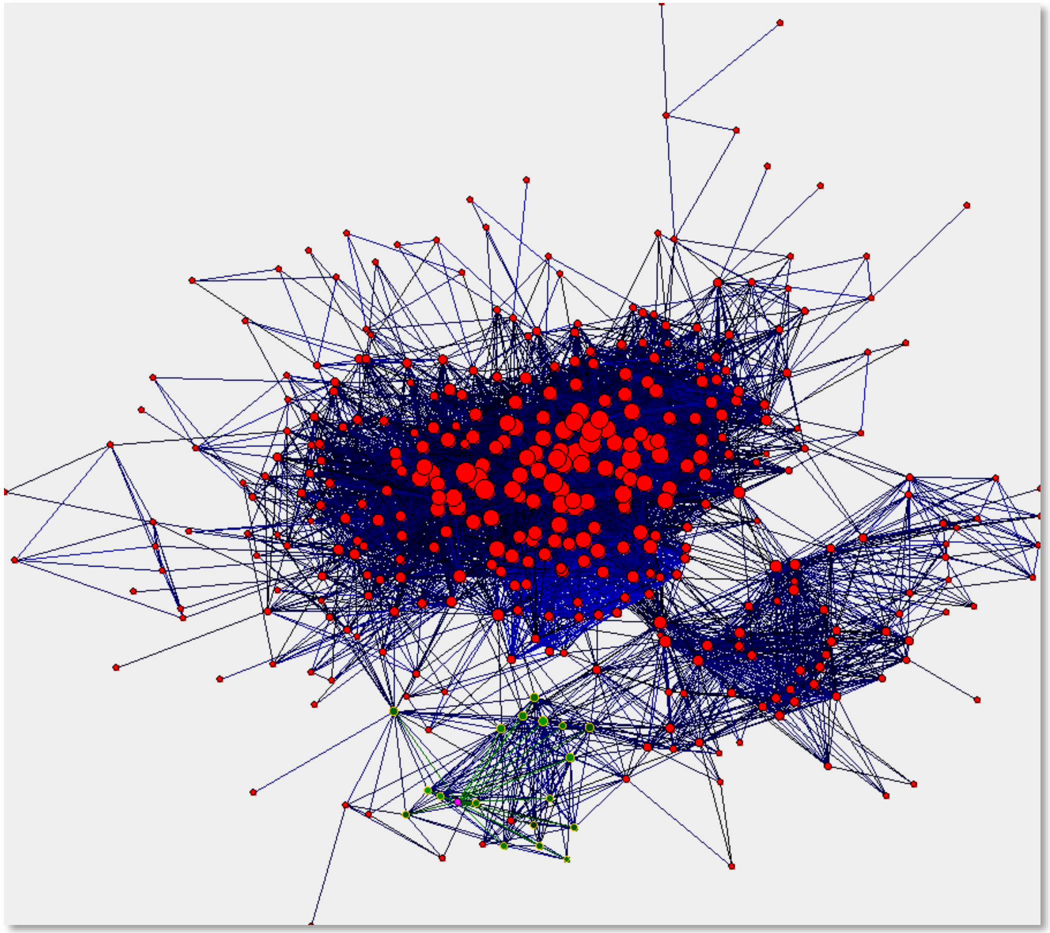


Image connectivity graph

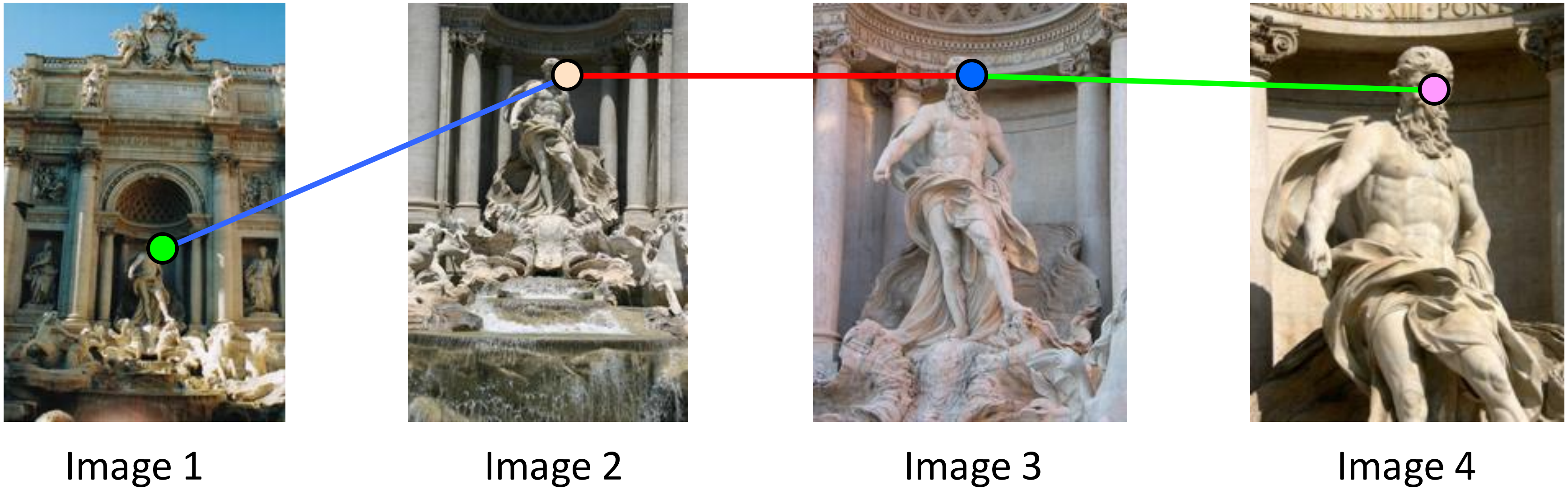


(graph layout produced using the Graphviz toolkit: http://www.graphviz.org/)

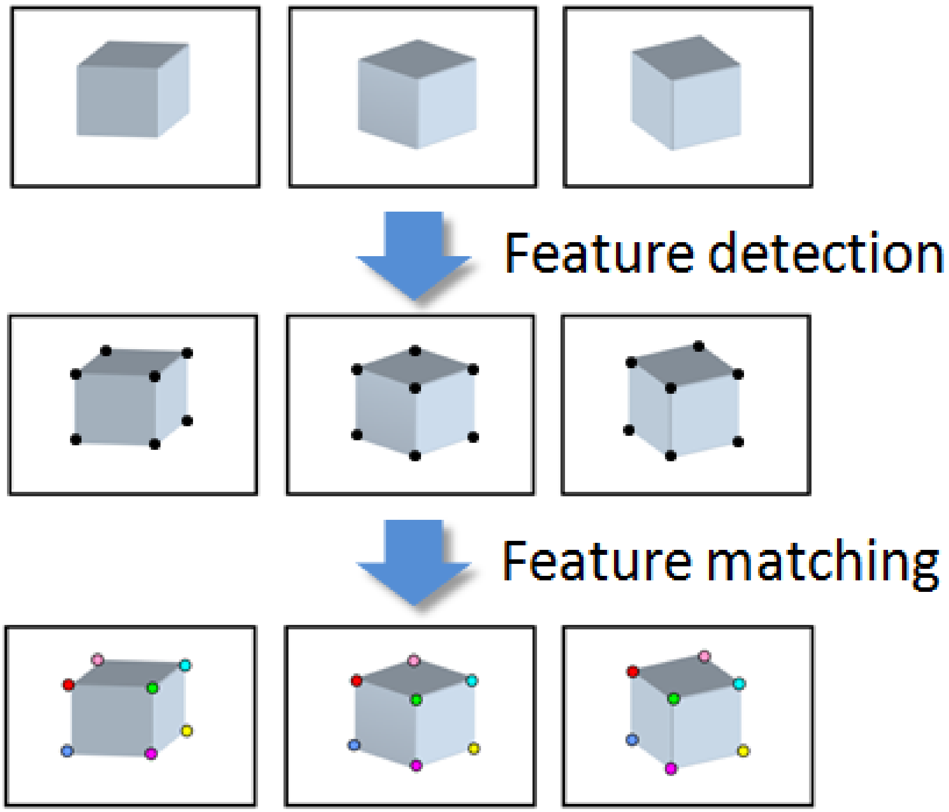
Demo

Correspondence estimation

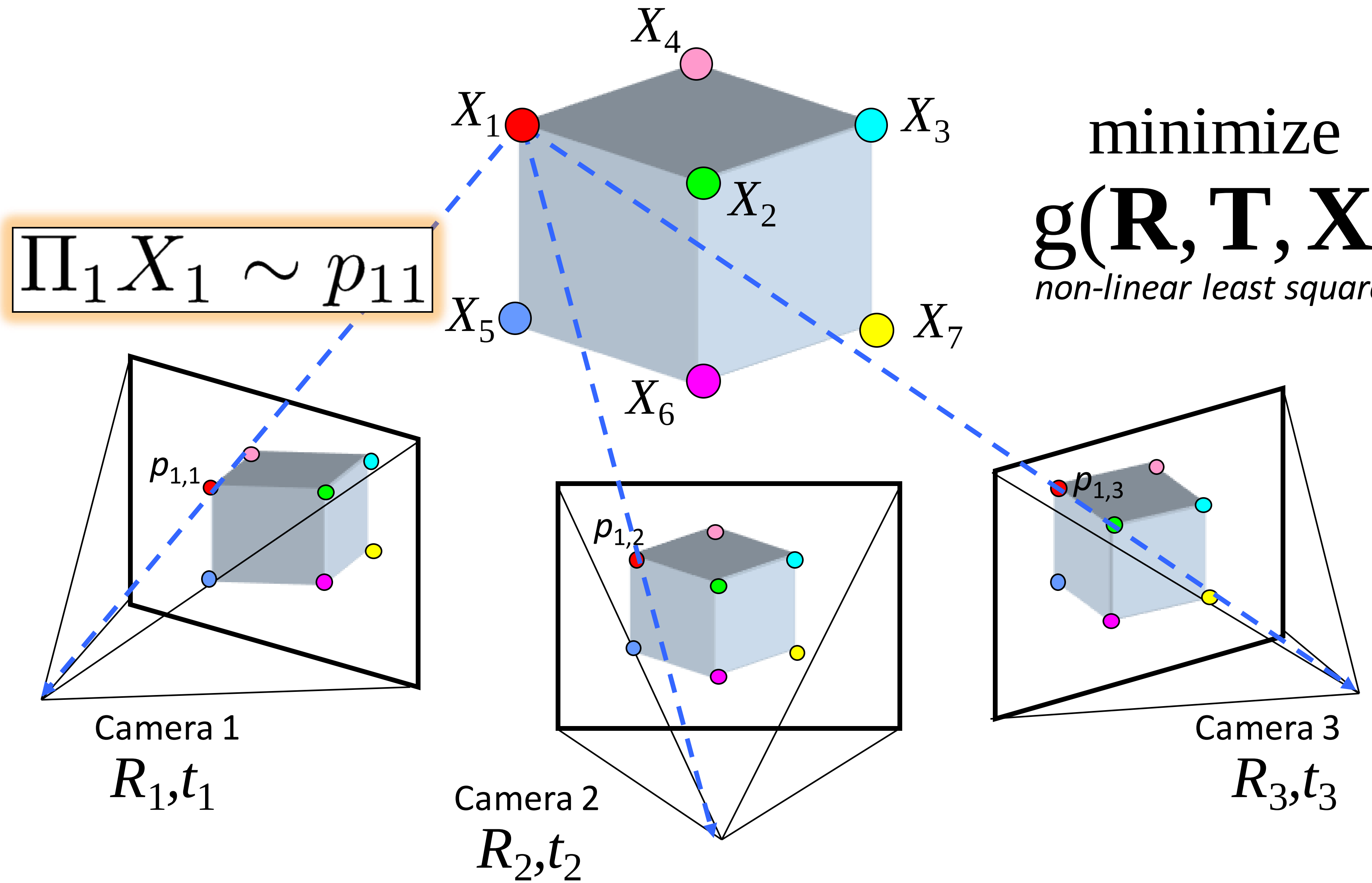
- Link up pairwise matches to form connected components of matches across several images



Input



Structure from motion



minimize

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

non-linear least squares

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

Structure from motion

- Minimize sum of squared reprojection errors:

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

predicted
image location

observed
image location

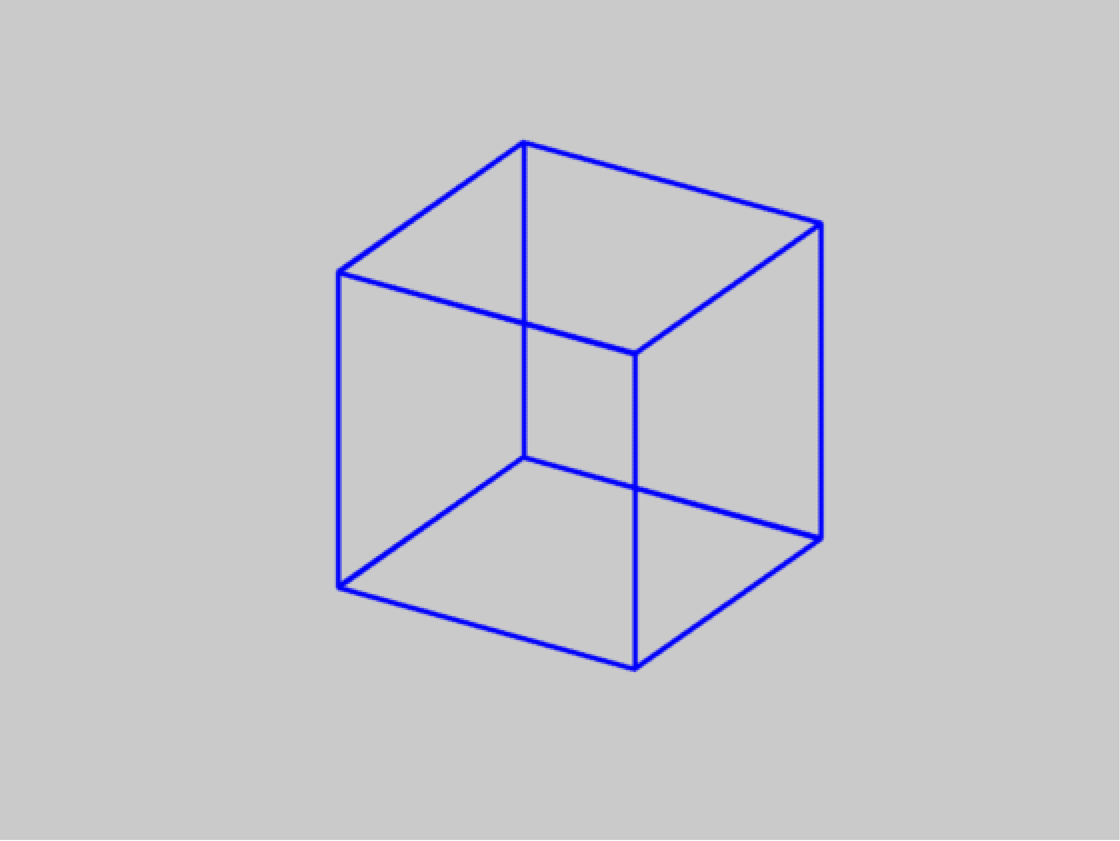
indicator variable:
is point *i* visible in image *j* ?

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

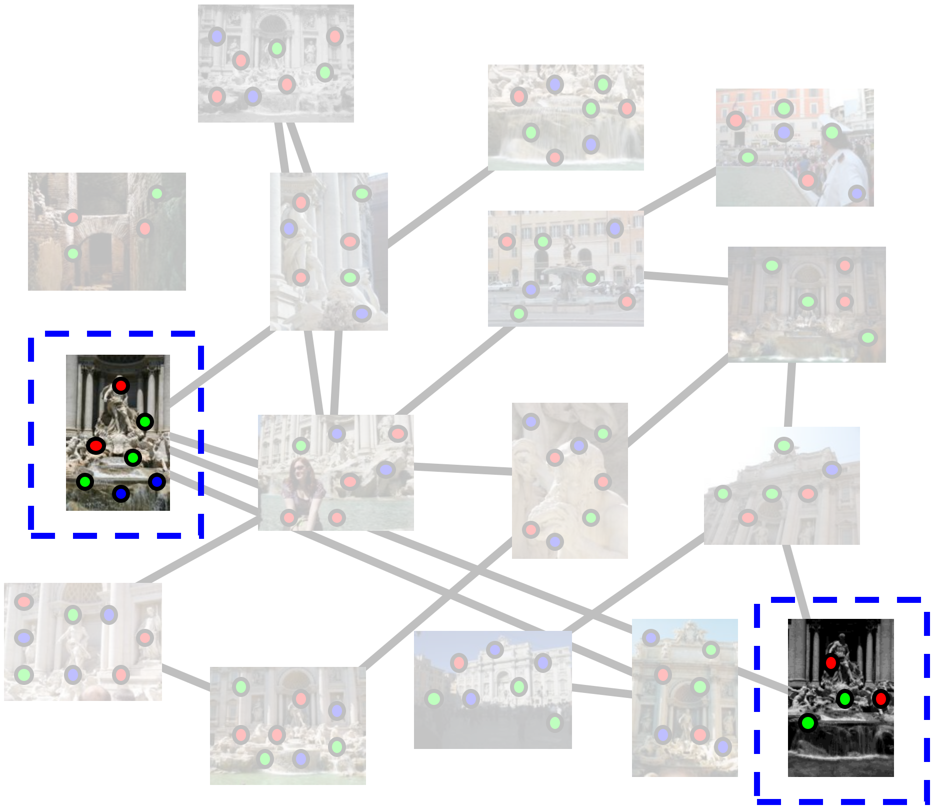
Is SfM always uniquely solvable?

Is SfM always uniquely solvable?

- No…



Incremental structure from motion



Incremental structure from motion

Incremental structure from motion

Incremental structure from motion

Time-lapse reconstruction of Dubrovnik, Croatia, viewed from above

Photo Explorer

Libration

From Wikipedia, the free encyclopedia

This article is about astronomical observations. For molecular motion, see Libration (molecule).

Not to be confused with liberation, libation, or vibration.

In astronomy, **libration** is a perceived oscillating motion of orbiting bodies relative to each other, notably including the motion of the Moon relative to Earth, or of trojan asteroids relative to planets. Lunar libration is distinct from the slight changes in the Moon's apparent size viewed from Earth. Although this appearance can also be described as an oscillating motion, it is caused by actual changes in the physical distance of the Moon because of its elliptic orbit around Earth. Lunar libration is caused by three phenomena detailed below.

The phase and libration of the Moon for 2013 at hourly intervals, with music, titles and supplemental graphics.

Contents [hide]

Lunar libration [edit source]

Simulated views of the Moon over one month, demonstrating librations in latitude and longitude. Also visible are

The Moon keeps one hemisphere of itself facing the Earth, due to tidal locking. Therefore, humans' first view of the far side of the Moon resulted from lunar exploration on October 7, 1959. However, this simple picture is only approximately true: over time, slightly *more* than half (about 59%) of the Moon's surface is seen from Earth due to libration.[1]

Libration is manifested as a slow rocking back and forth of the Moon as viewed from Earth, permitting an observer to see slightly different halves of the surface at different times.

There are three types of lunar libration:

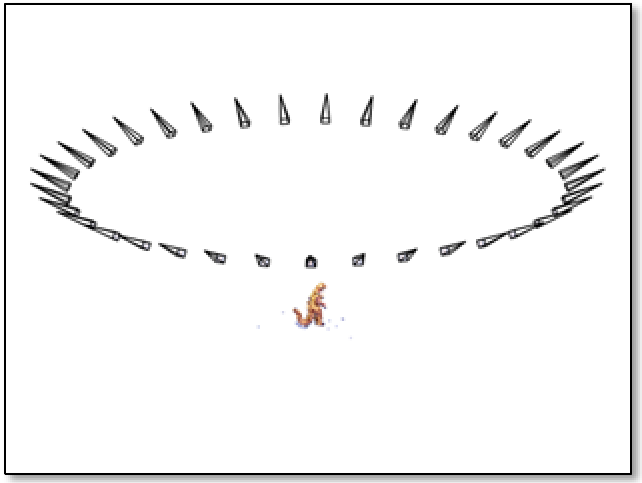
- *Libration in longitude* results from the eccentricity of the Moon's orbit around Earth; the Moon's rotation sometimes leads and sometimes lags its orbital position.
- *Libration in latitude* results from a slight inclination (about 6.7 degrees) between the Moon's axis of rotation and the normal

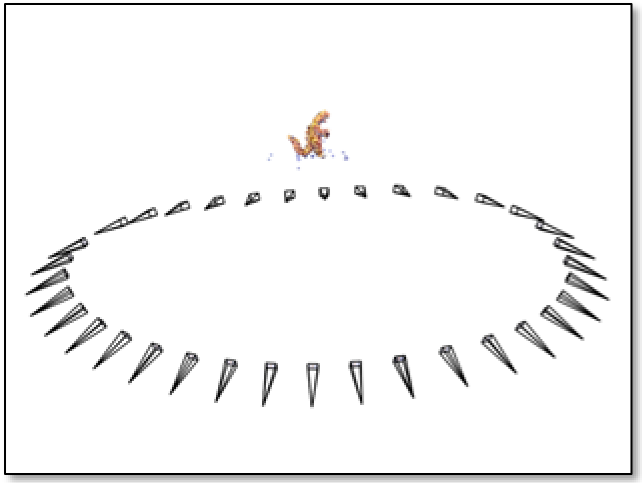
https://en.wikipedia.org/wiki/Libration

Questions?

SfM – Failure cases

- Necker reversal





SfM applications

- 3D modeling
- Surveying
- Robot navigation and mapmaking
- Visual effects...
  - (see video)

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 – Hyperlapse

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

https://www.youtube.com/watch?v=sA4Za3Hv6ng

Applications: Visual Reality & Augmented Reality

Oculus

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

Hololens

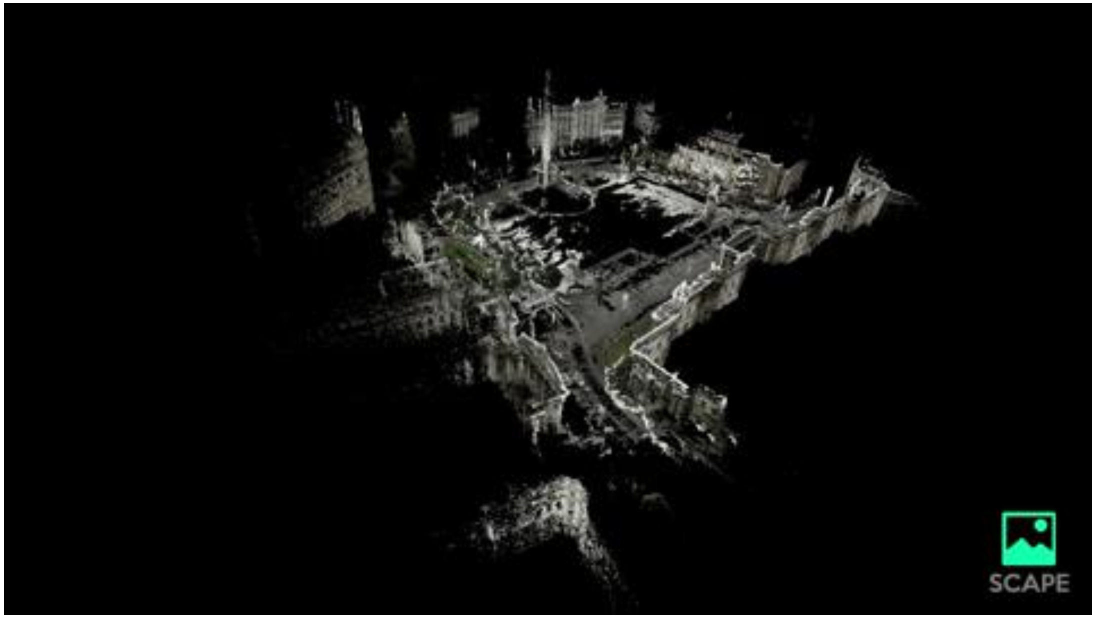
https://www.youtube.com/watch?v=FMtvrTGnP04

Applications: Simultaneous localization and mapping (SLAM)

https://www.youtube.com/watch?v=k43xJs3Roqg

https://www.youtube.com/watch?v=ZR1yXFAslSk

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

Application: AR walking directions

Questions?