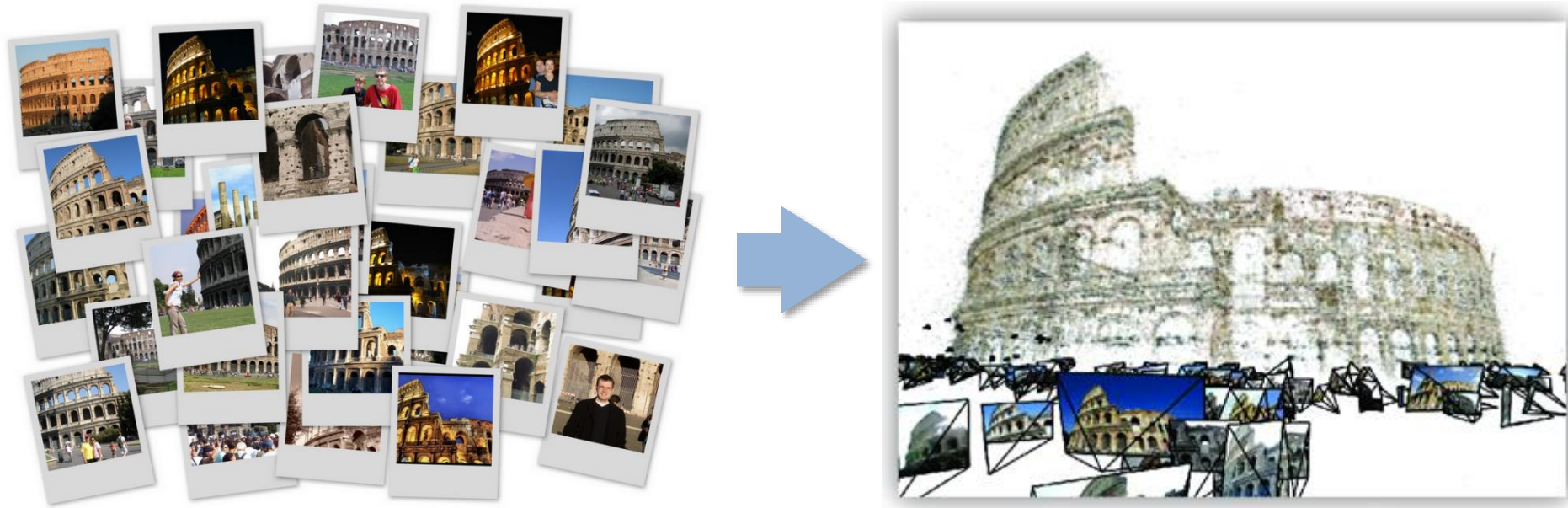


# CS5670: Computer Vision

## Structure from motion

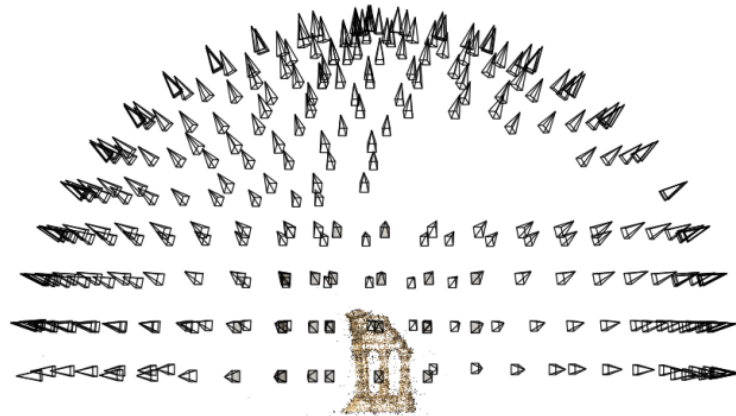


# Readings

- Szeliski (1<sup>st</sup> Edition), Chapter 7.1 – 7.4

# Reminder: multi-view stereo

**Problem formulation:** given several images of the same object or scene, compute a representation of its 3D shape



# Structure from motion

- Multi-view stereo assumes that cameras are calibrated
  - Extrinsic and intrinsic 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  $3 \times 3 \times 3$  tensor called the *trifocal tensor*
- The geometry of four views is described by a  $3 \times 3 \times 3 \times 3$  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

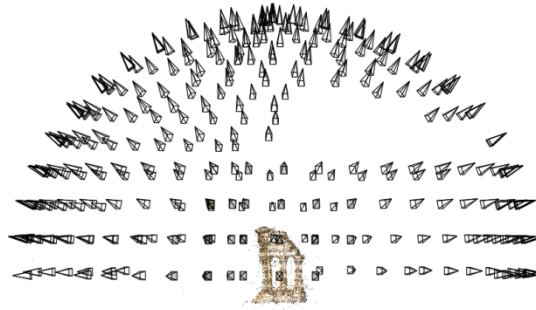
- Given many images, how can we
  - a) figure out where they were all taken from?
  - b) build a 3D model of the scene?



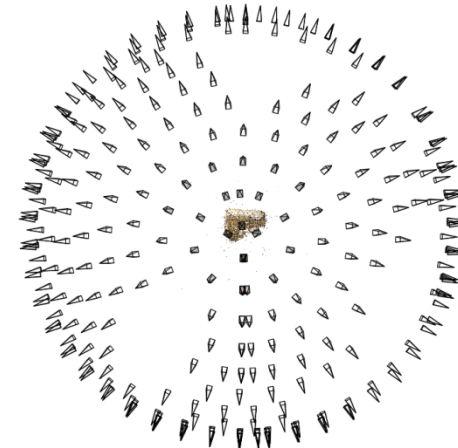
This is the **structure from motion (SfM)** problem



# Structure from motion



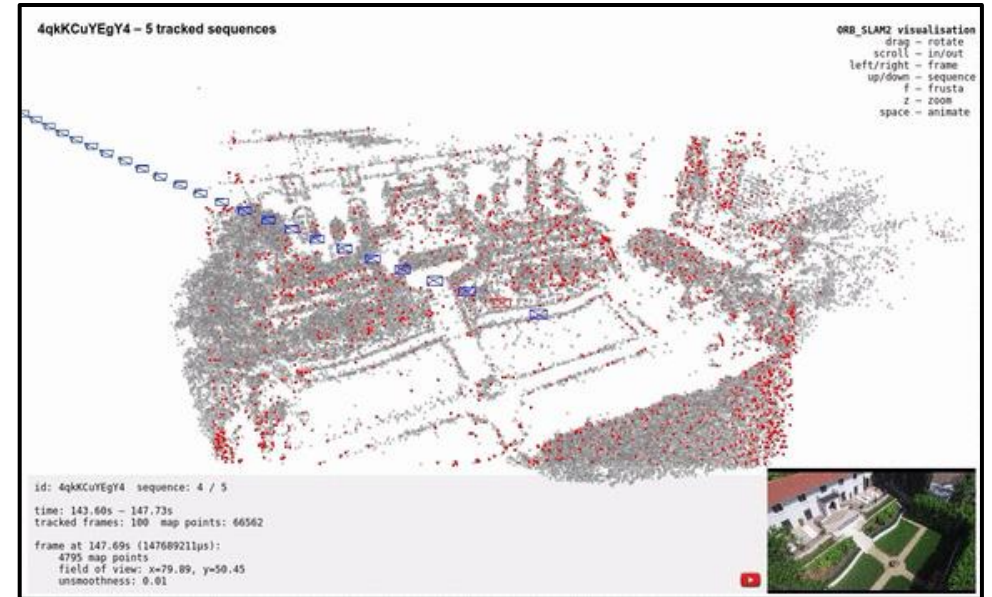
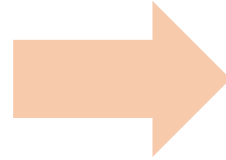
Reconstruction (side)



(top)

- Input: images with 2D points in correspondence  
$$p_{ij} = (u_{ij}, v_{ij})$$
- 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*

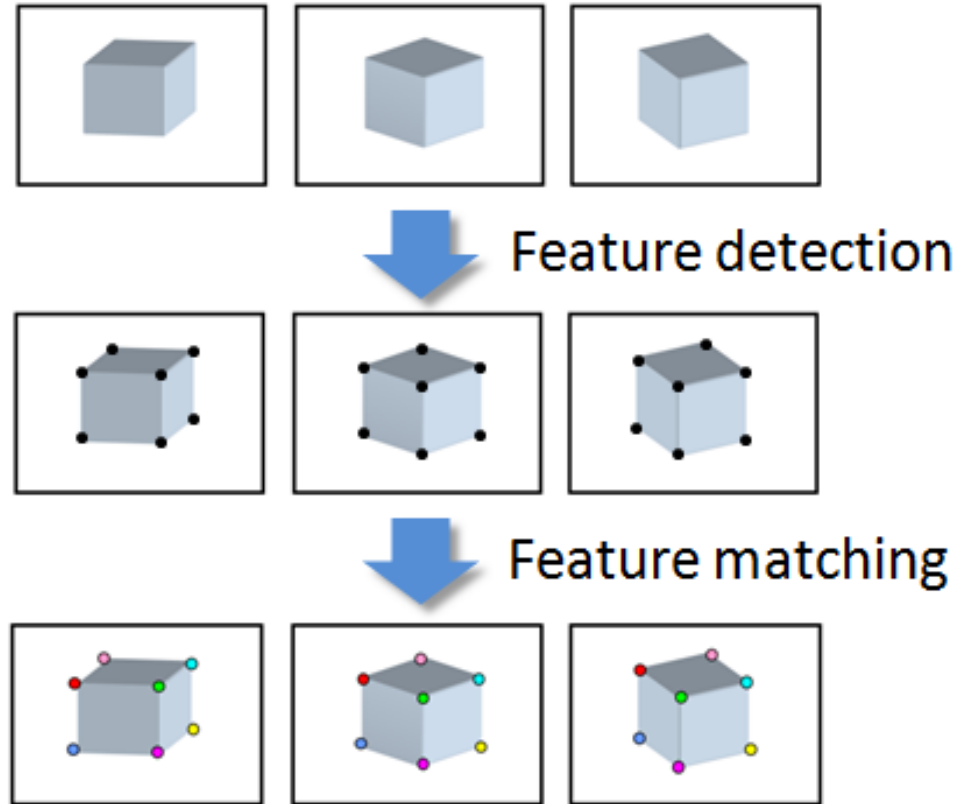
# Also doable from video



# What we've seen so far...

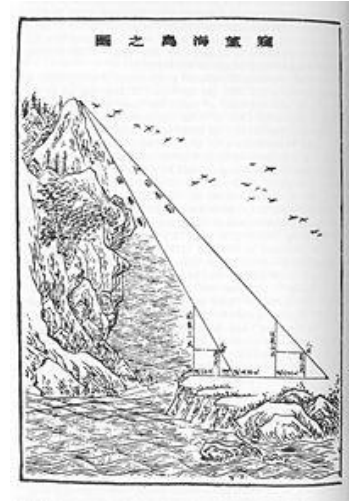
- 2D transformations between images
  - Translations, affine transformations, homographies...
- Fundamental matrices
  - Represent relationships between 2D images in the form of corresponding 2D lines
- **What's new:** Explicitly representing 3D geometry of cameras *and points*

# Input



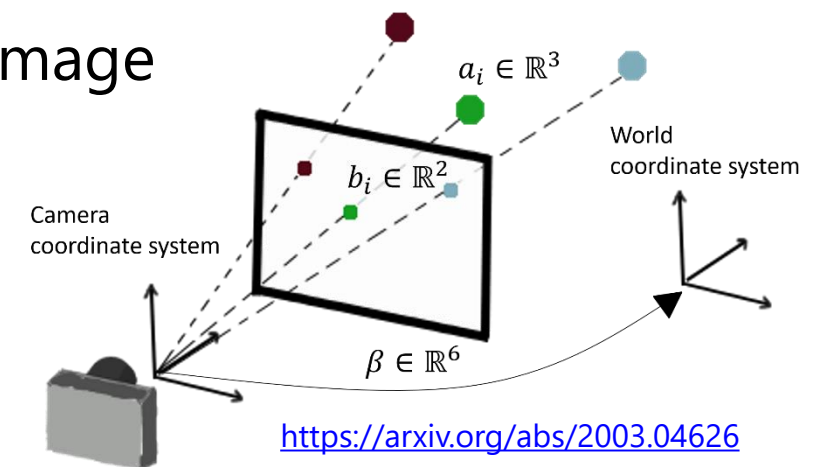
# Triangulation & camera calibration

- Suppose we have known camera parameters, each of which observes a point
  - How can we compute the 3D location of that point?
  - This is called *triangulation* (known since ancient times)



[Liu Hui](#) (c. 263), How to measure the height of a sea island.

- On the other hand: Suppose we have known 3D points
  - And have matches between these points and an image
  - How can we compute the camera parameters?
  - This is called *camera calibration* (or *camera resectioning*)



<https://arxiv.org/abs/2003.04626>

# Structure from motion

- What if we don't know 3D points or camera parameters?
- SfM solves both of these problems *at once*
- A kind of chicken-and-egg problem
  - (but solvable)

# Photo Tourism



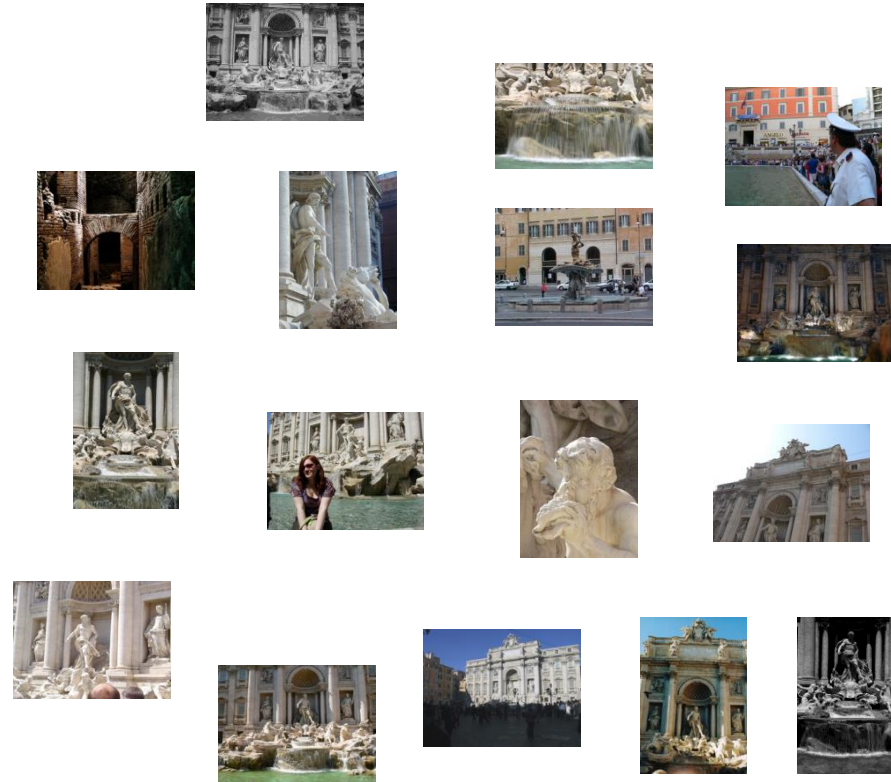
# **First step: how to get correspondence?**

- Feature detection and matching



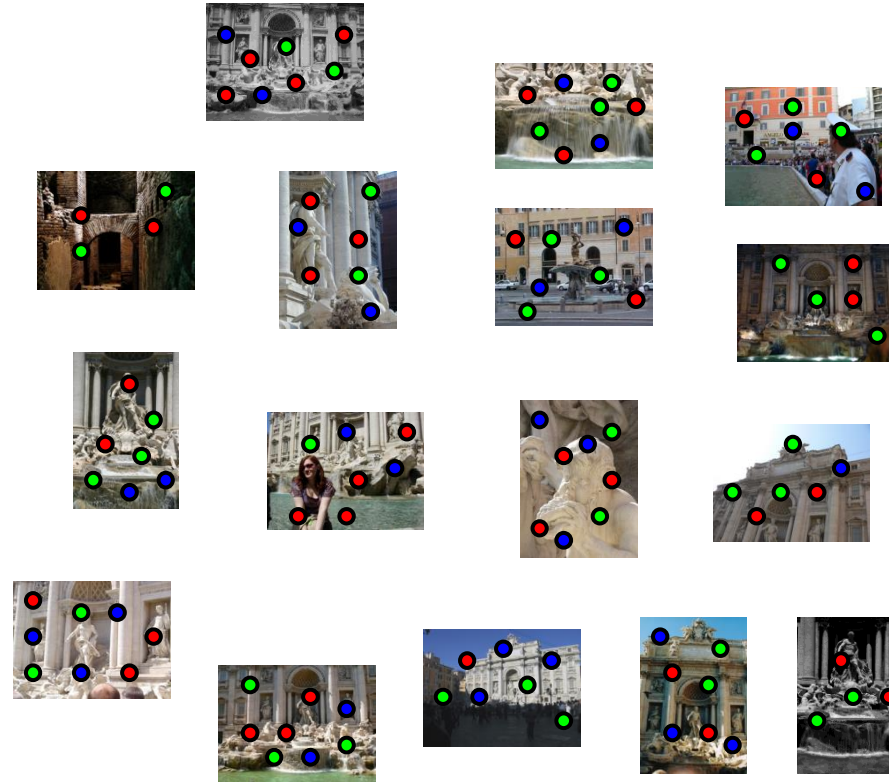
# Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



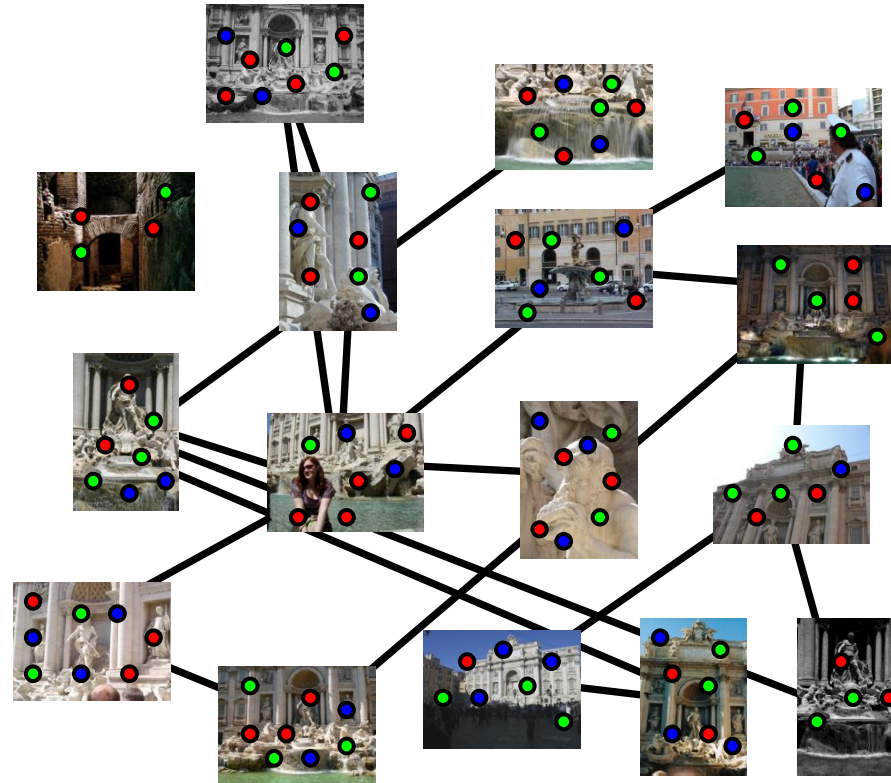
# Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



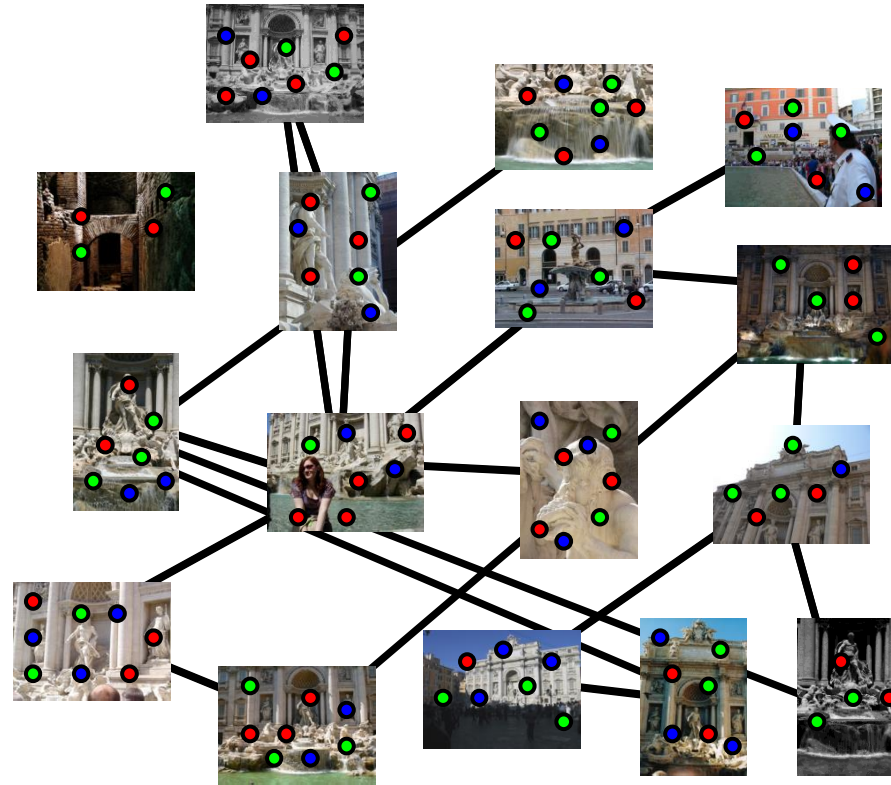
# Feature matching

Match features between each pair of images

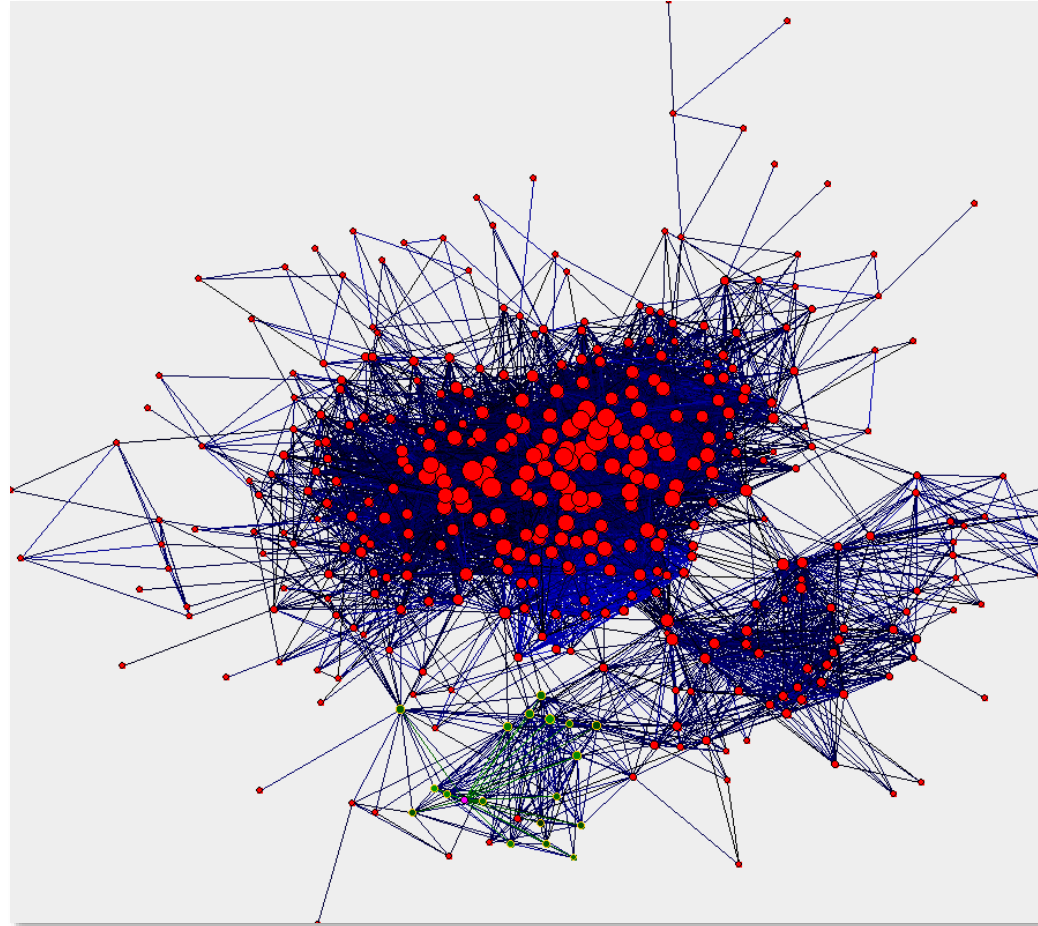


# Feature matching

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

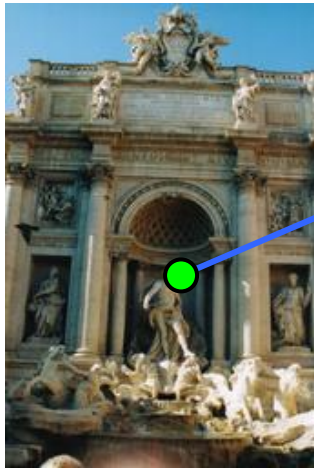


Image 1



Image 2

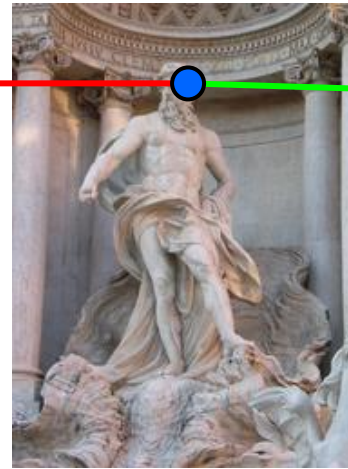


Image 3

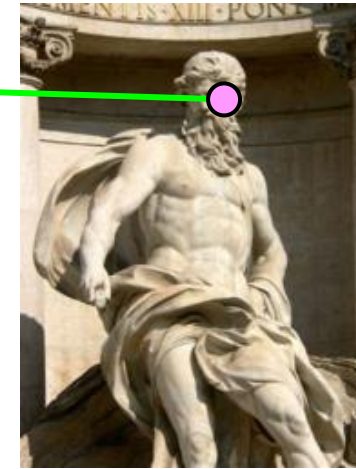
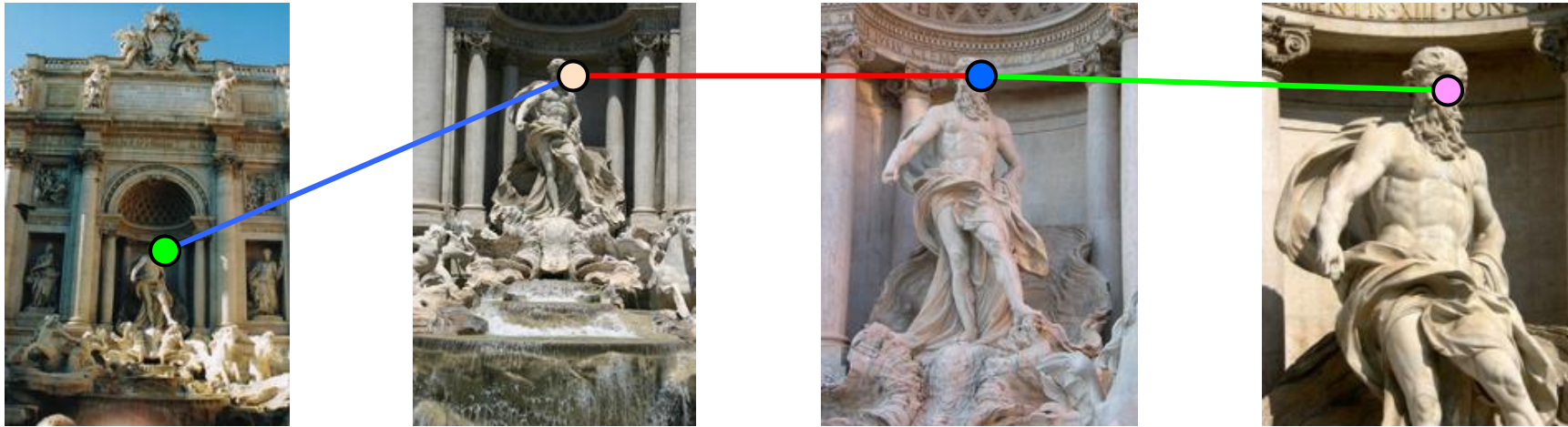
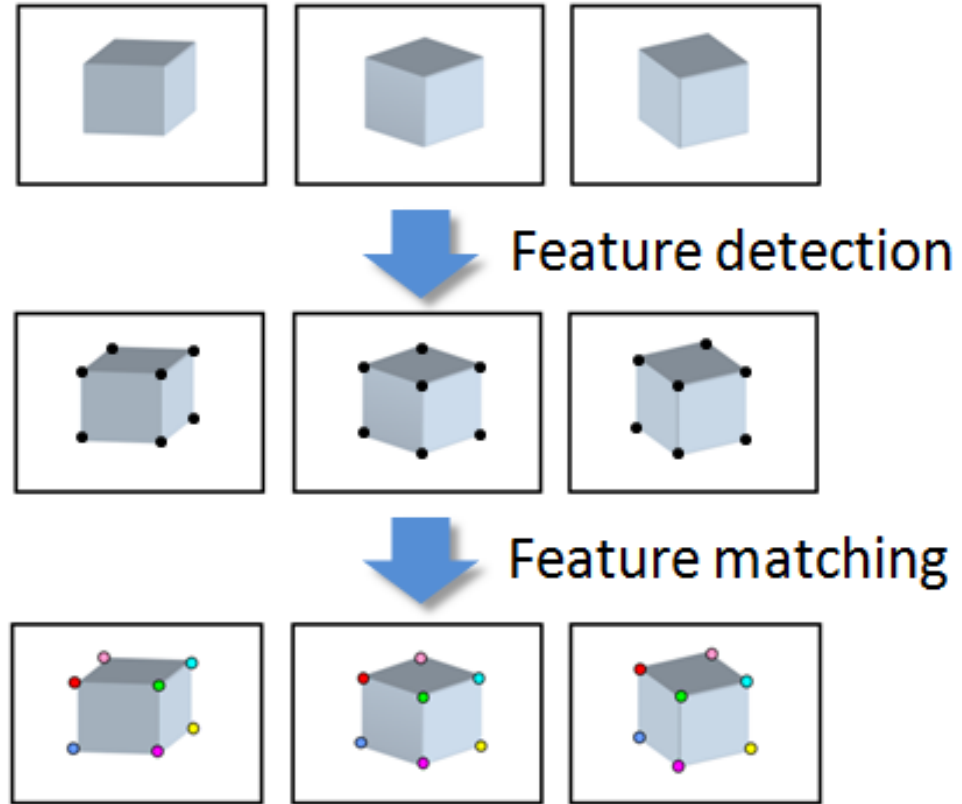


Image 4

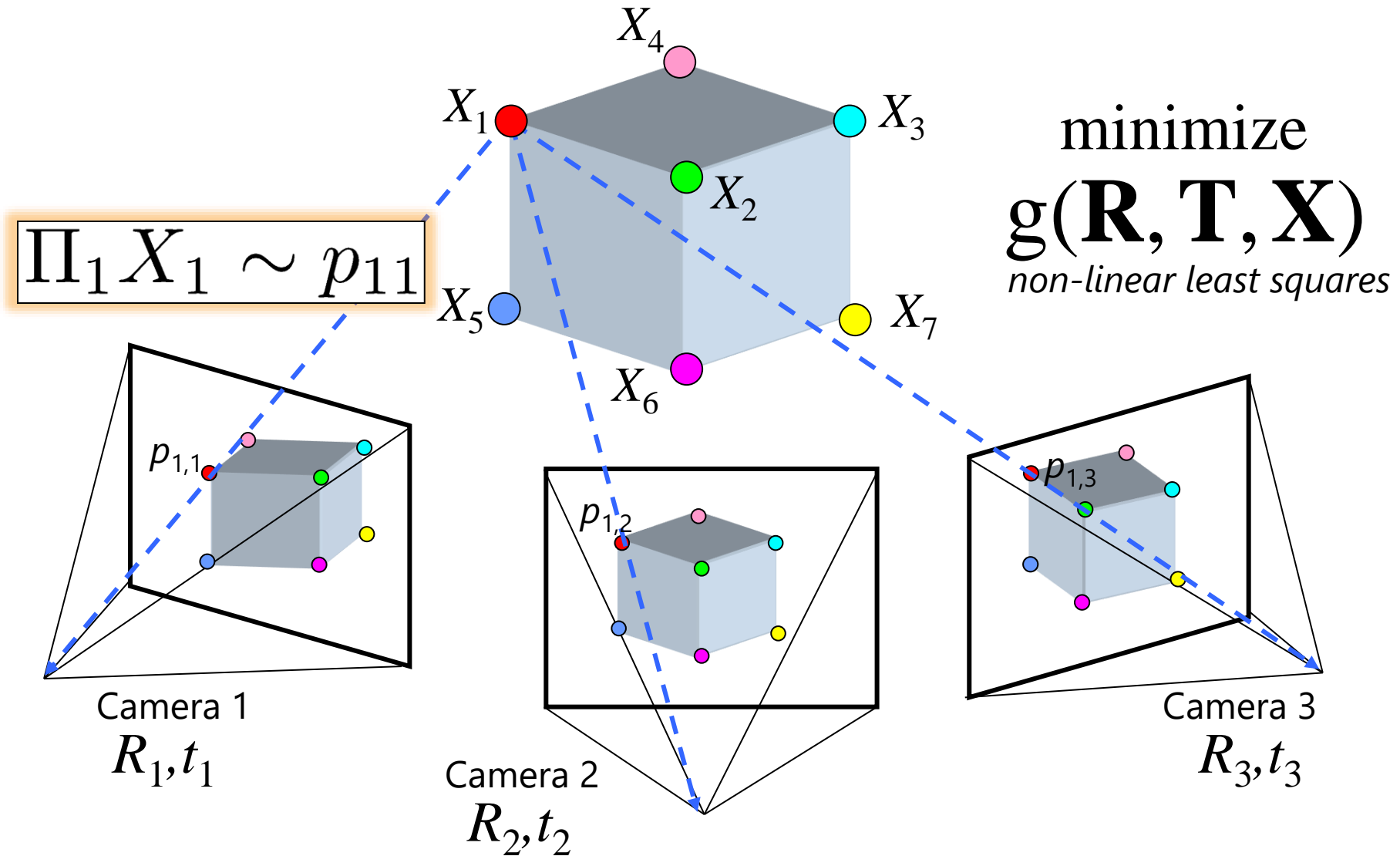


# Input to Structure from Motion





# Structure from motion



# Problem size

- What are the variables?
  - Cameras and points
- How many variables per camera?
  - 6 (if calibrated), more if uncalibrated
- How many variables per point?
  - 3
  
- Trevi Fountain collection
  - 466 input photos
  - + > 100,000 3D points
  - = very large optimization problem

# Structure from motion

- Minimize sum of squared reprojection errors:

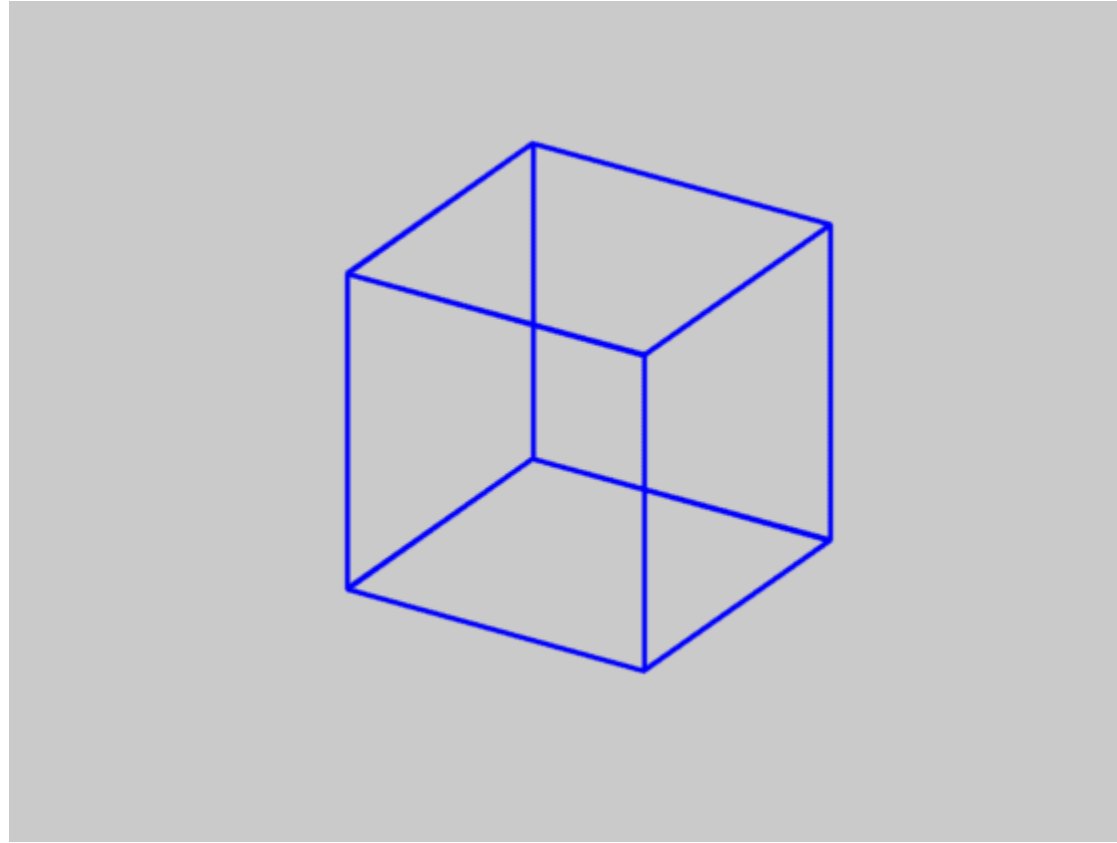
$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^m \sum_{j=1}^n \underbrace{w_{ij}}_{\substack{\text{indicator variable:} \\ \text{is point } i \text{ visible in image } j?}} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\substack{\text{predicted} \\ \text{image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

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

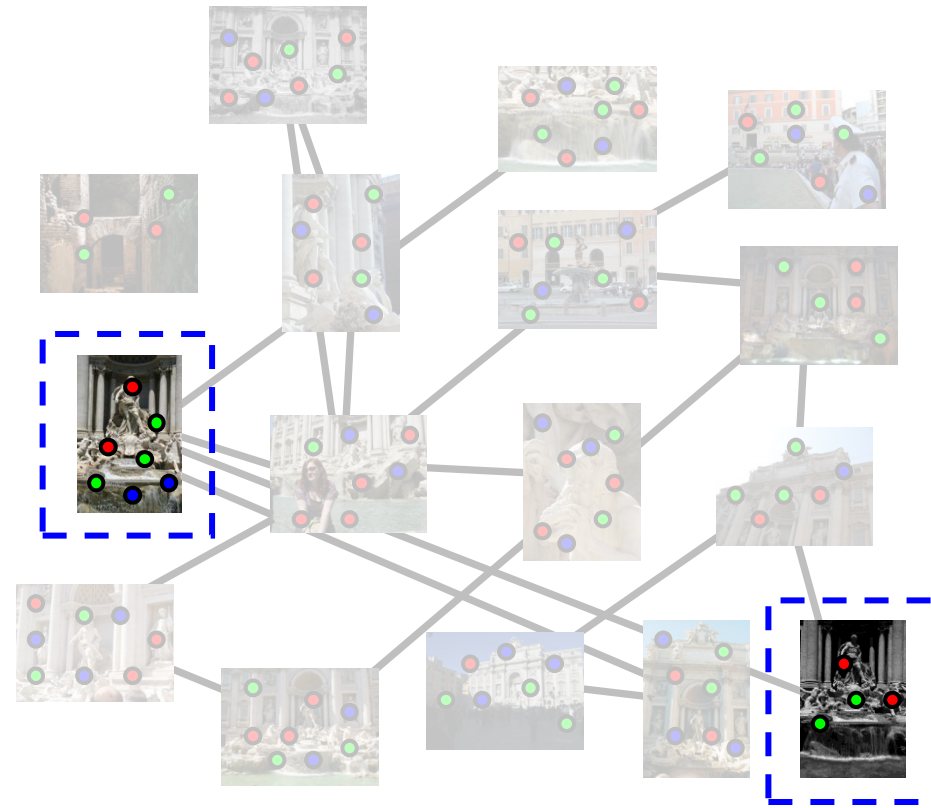
**Is SfM always uniquely solvable?**

# Is SfM always uniquely solvable?

- No...



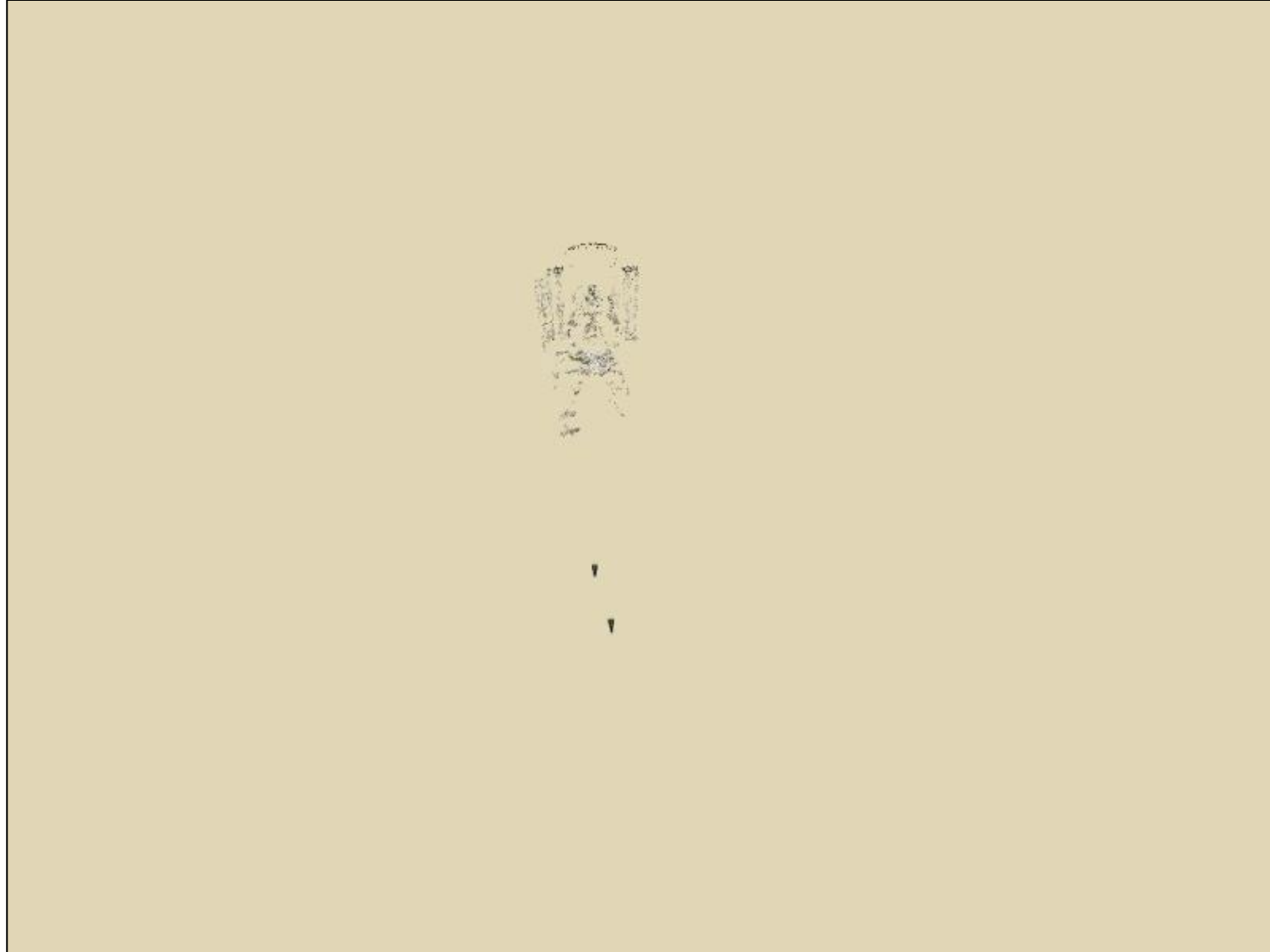
# Incremental structure from motion



# Incremental structure from motion



# Incremental structure from motion



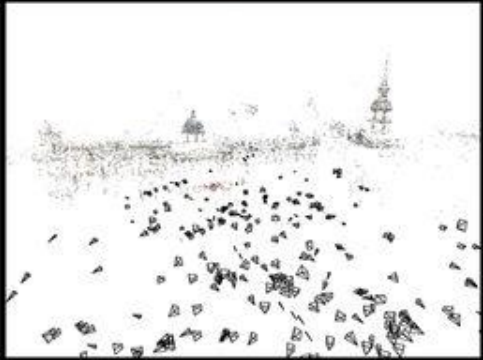
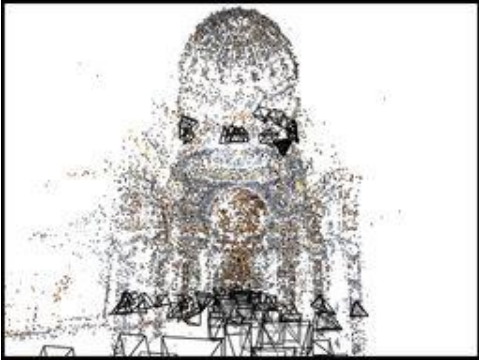
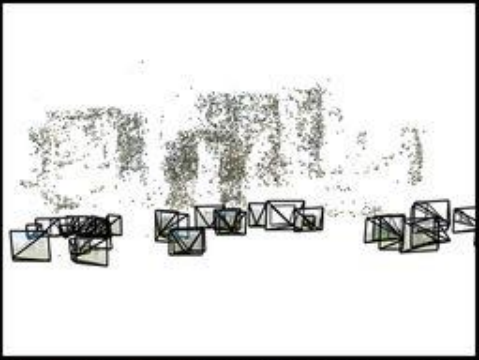
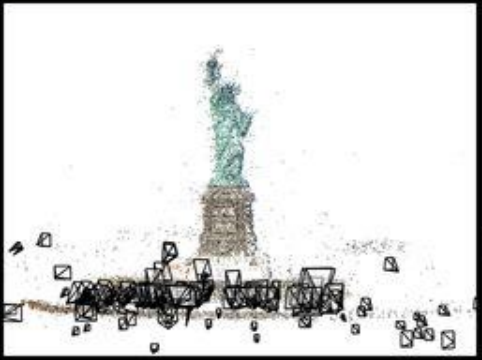
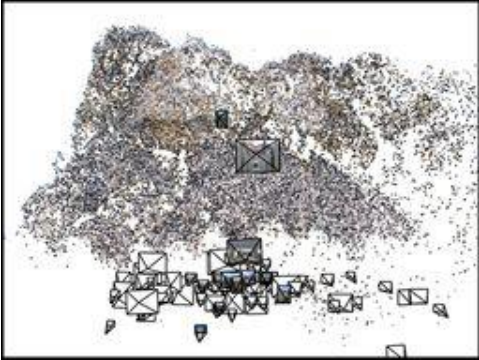
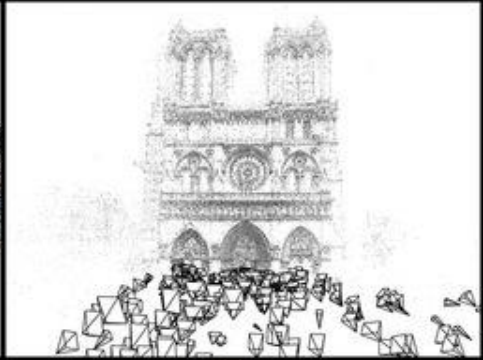
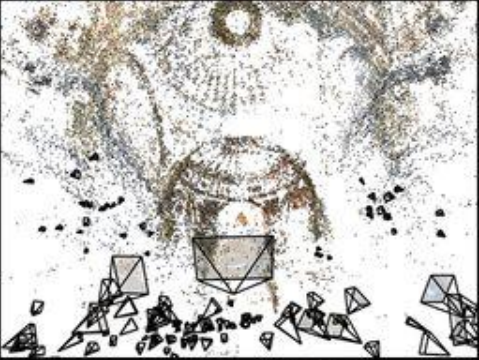
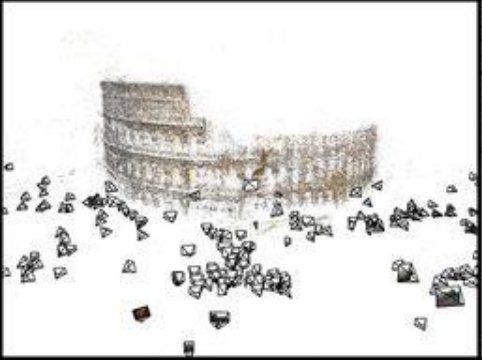


# Incremental structure from motion

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

# Photo Tourism









# 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.

## Contents [hide]

- [Lunar libration](#)
- [Trojan libration](#)
- [See also](#)
- [References](#)
- [External links](#)

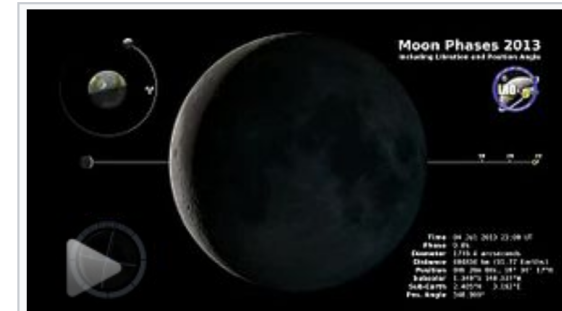
## Lunar libration [edit source]

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.<sup>[1]</sup>

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](#)



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



Simulated views of the Moon over one month, demonstrating librations in [latitude](#) and [longitude](#). Also visible are [edit]

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

# Libration

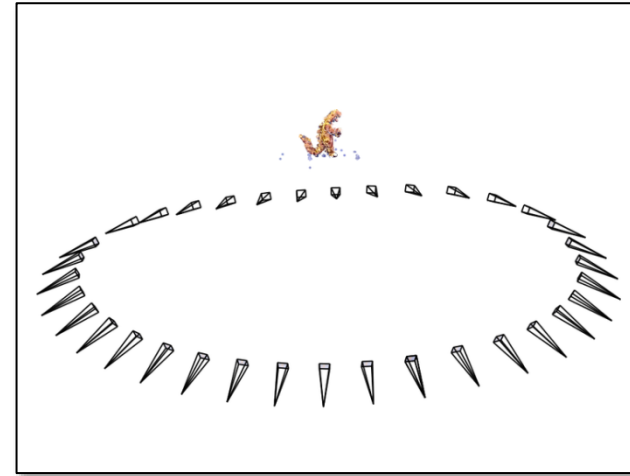
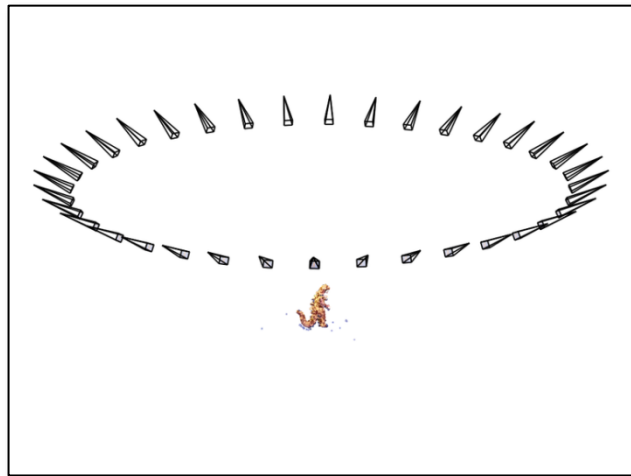


**Questions?**



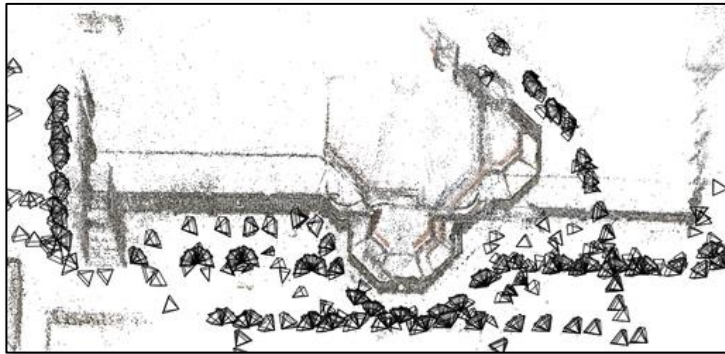
# SfM – Failure cases

- Necker reversal



# Structure from Motion – Failure cases

- Repetitive structures: Symmetries in man-made scenes



# 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](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

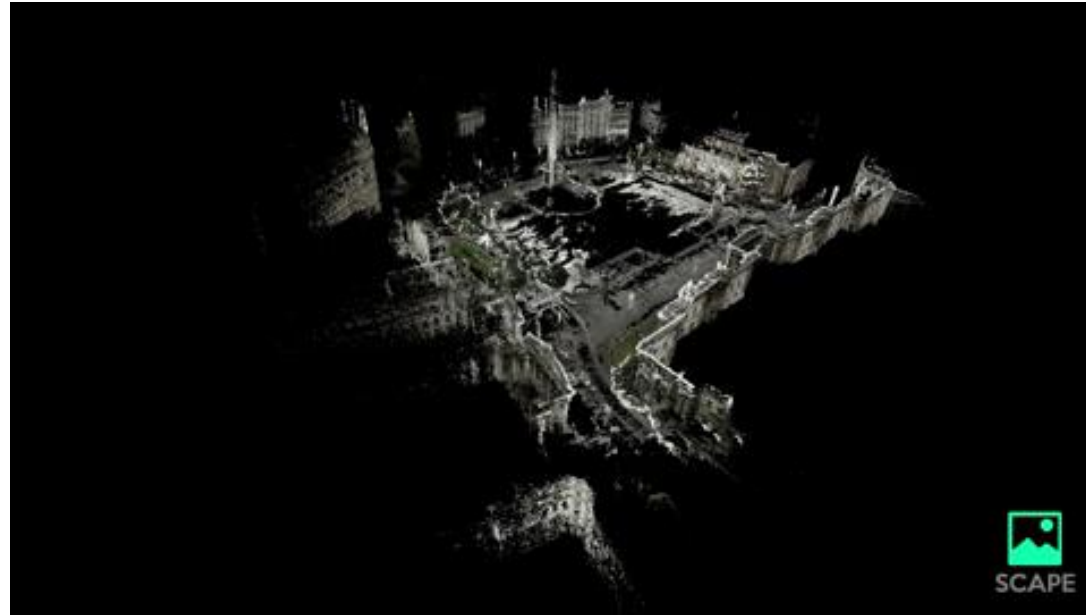
<https://www.youtube.com/watch?v=KOG7yTz1iTA>



Hololens

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

# Applications: 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>

**Questions?**