Announcements

- Project 3 Due Friday
- Artifact due at the same time

- We are planning to announce new grading guidelines this week

The Fundamental Matrix & Structure from Motion

The Many Meanings of Vectors and Cross Products

- One algebraic form can have many geometric interpretations
- Let's review the algebra, and some of the geometry relevant for today…

Cross-Products: The Algebra

- Compute 3D vector **c** that has a dot product of zero with both 3D vectors **a**, and **b**

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} =$$

$$= (a_2 b_3 - a_3 b_2)\,\mathbf{i} + (a_3 b_1 - a_1 b_3)\,\mathbf{j} + (a_1 b_2 - a_2 b_1)\,\mathbf{k}$$

Cross-Product as Linear Operator

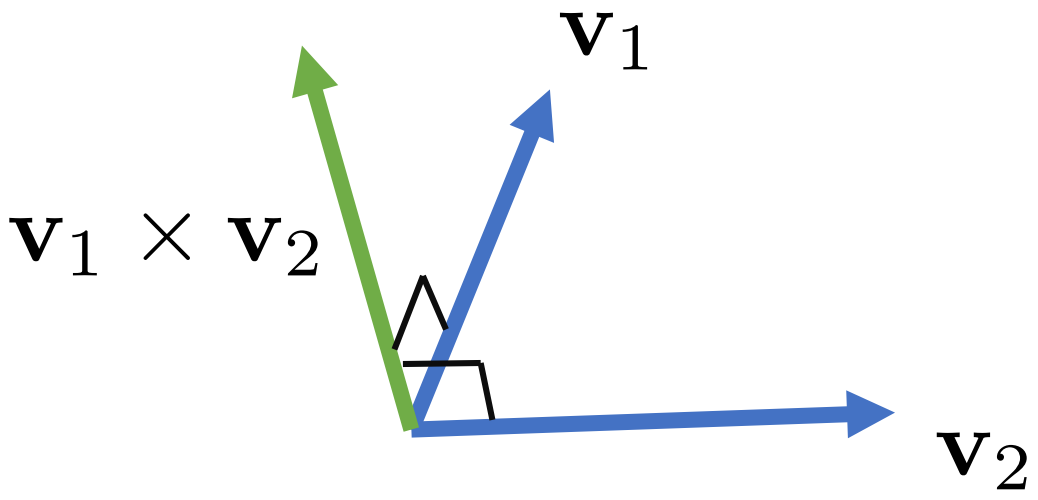
Useful fact: Cross product with a vector **t** can be represented as multiplication with a (*skew-symmetric*) 3x3 matrix

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$$\mathbf{t} \times \tilde{\mathbf{p}} = [\mathbf{t}]_\times \tilde{\mathbf{p}}$$

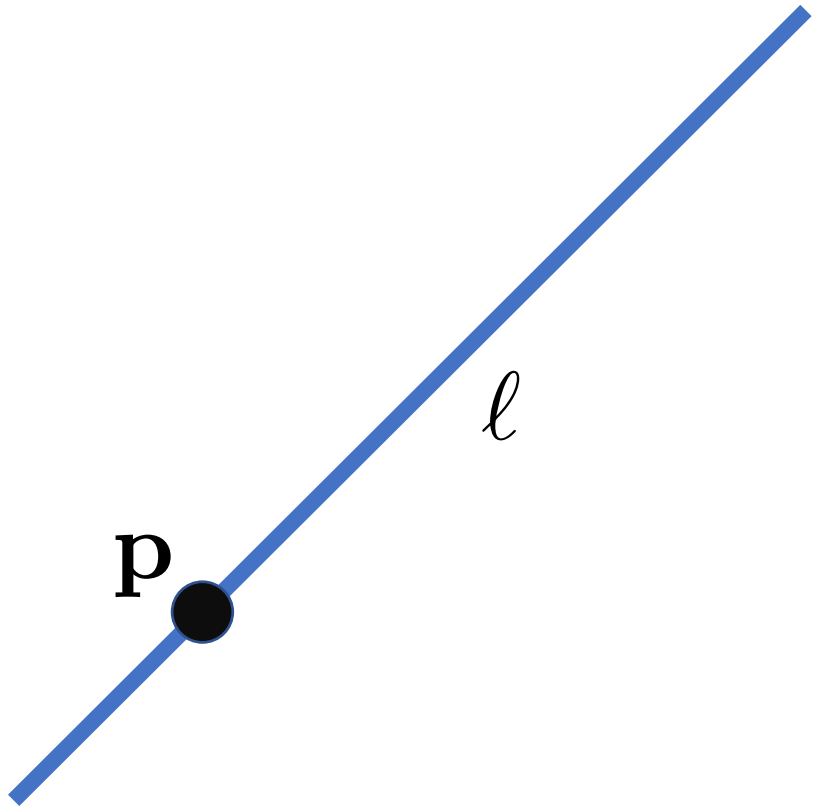
Cross Products: Geometric Interpretation

- The cross product of two vectors is the normal vector to plane between that contains them



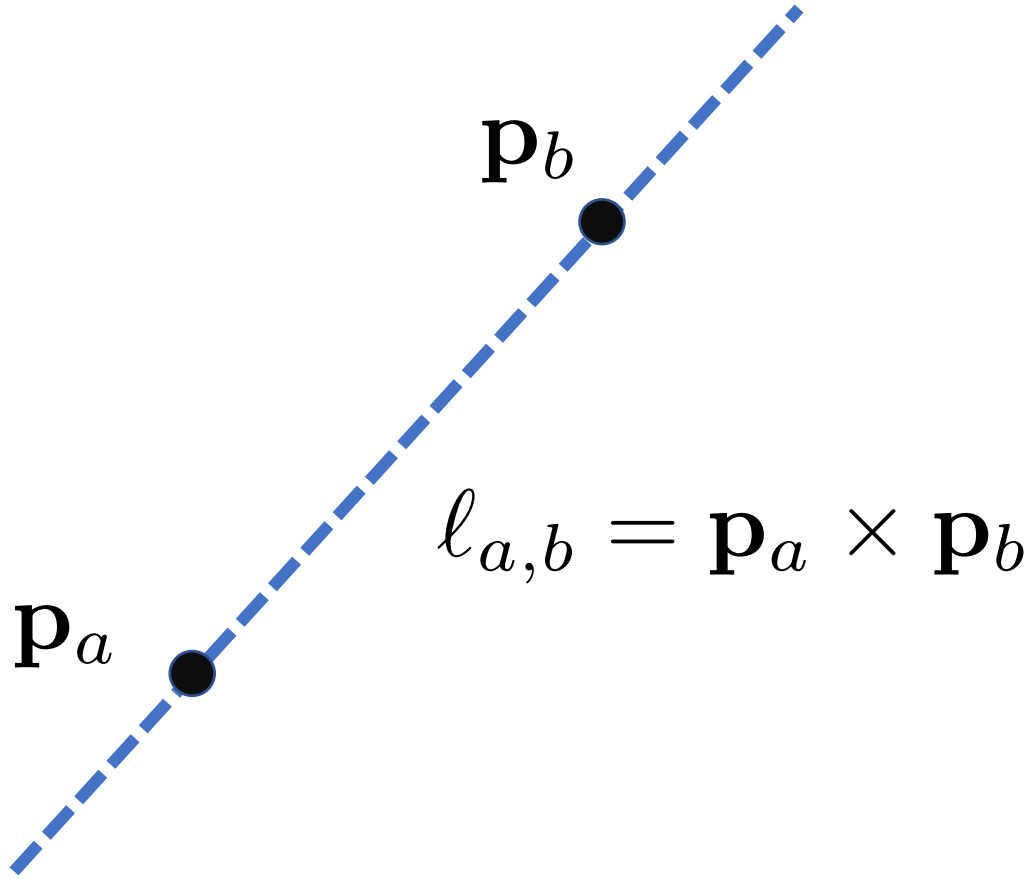
Point-Line Duality and the Cross-Product

- A point in 3D $$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
- A line ℓ in 3D is defined by the equation $\ell^\intercal \mathbf{p} = 0$
- Points and lines are both vectors of the same dimension with a homogeneous coordinate



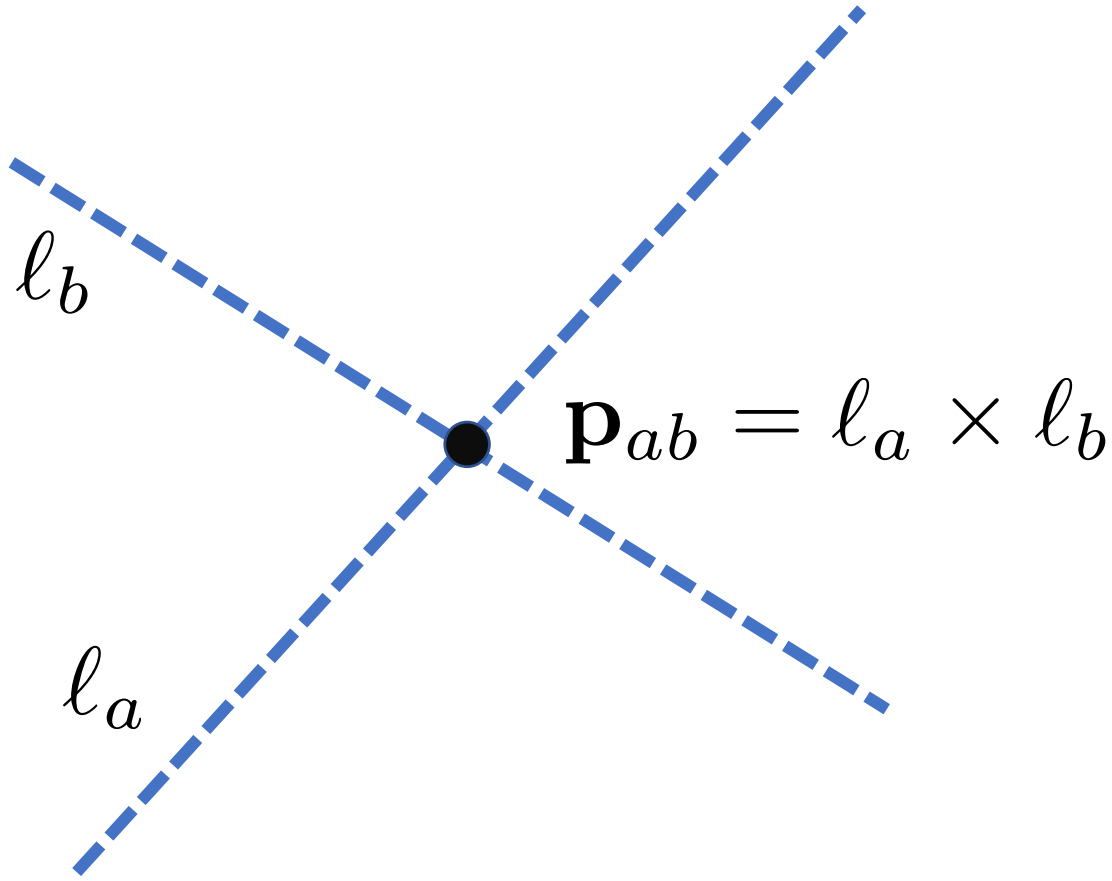
Point-Line Duality and the Cross-Product

- The line between two points is just the cross product of the two points



Point-Line Duality and the Cross-Product

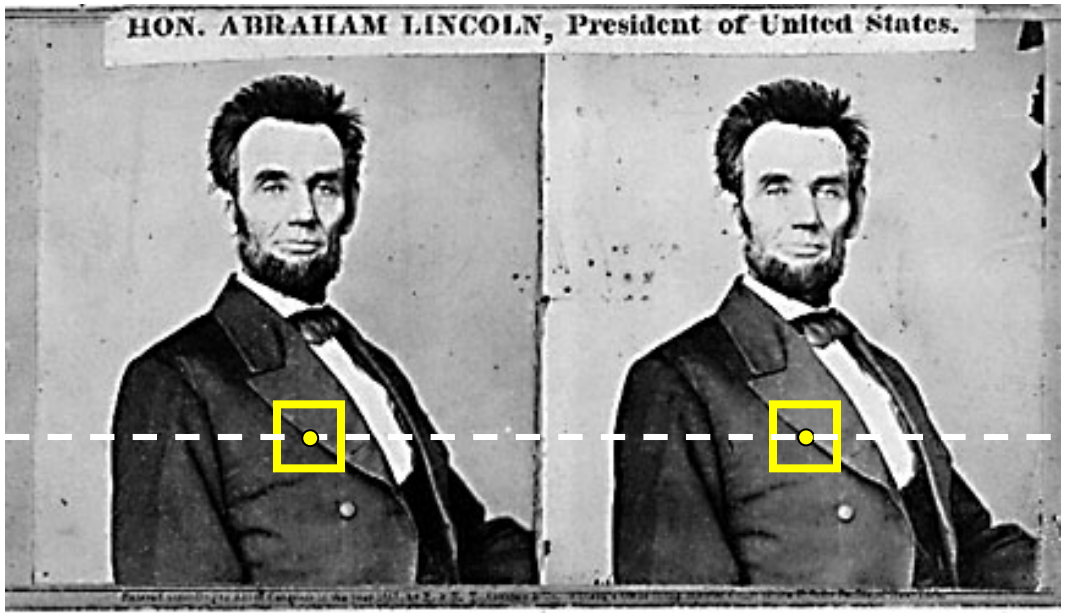
- The intersection of two lines is the cross produce of those lines



Point-Line Duality and the Cross-Product

- One vector can have multiple interpretations
- I suggest:
  - First think about the geometry
  - Then the representation
  - Then the algebra

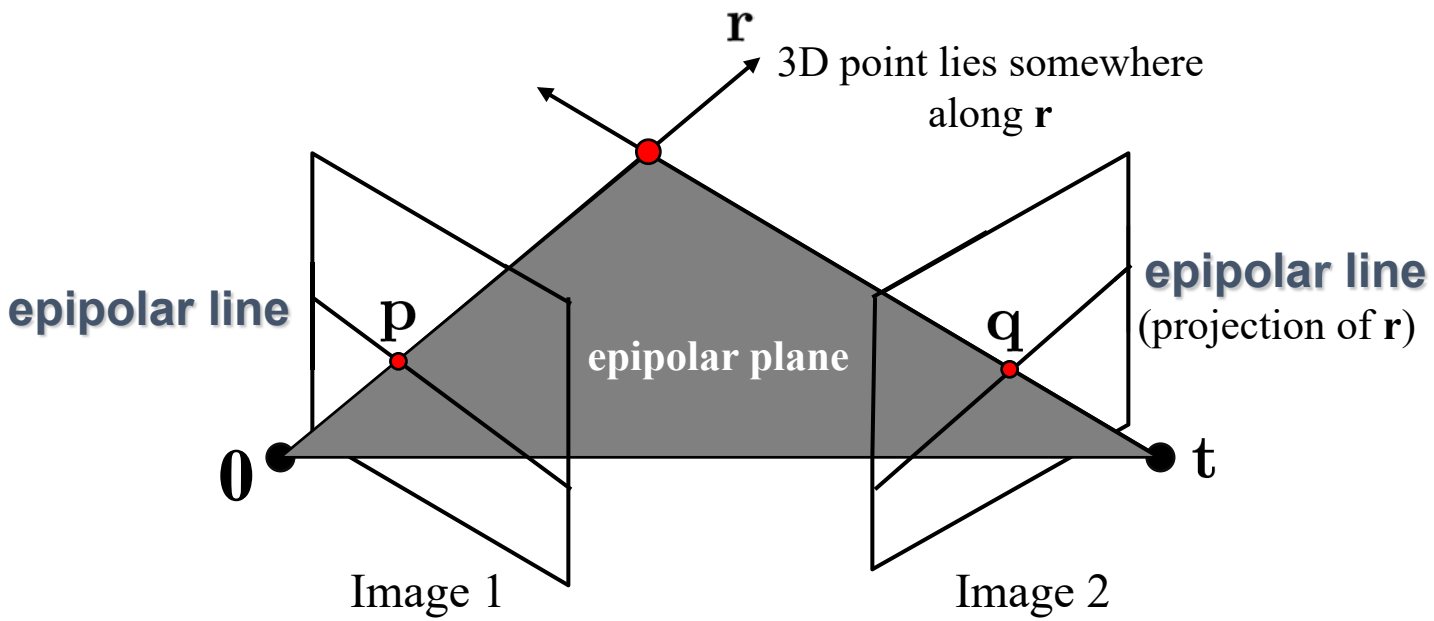
Back to stereo



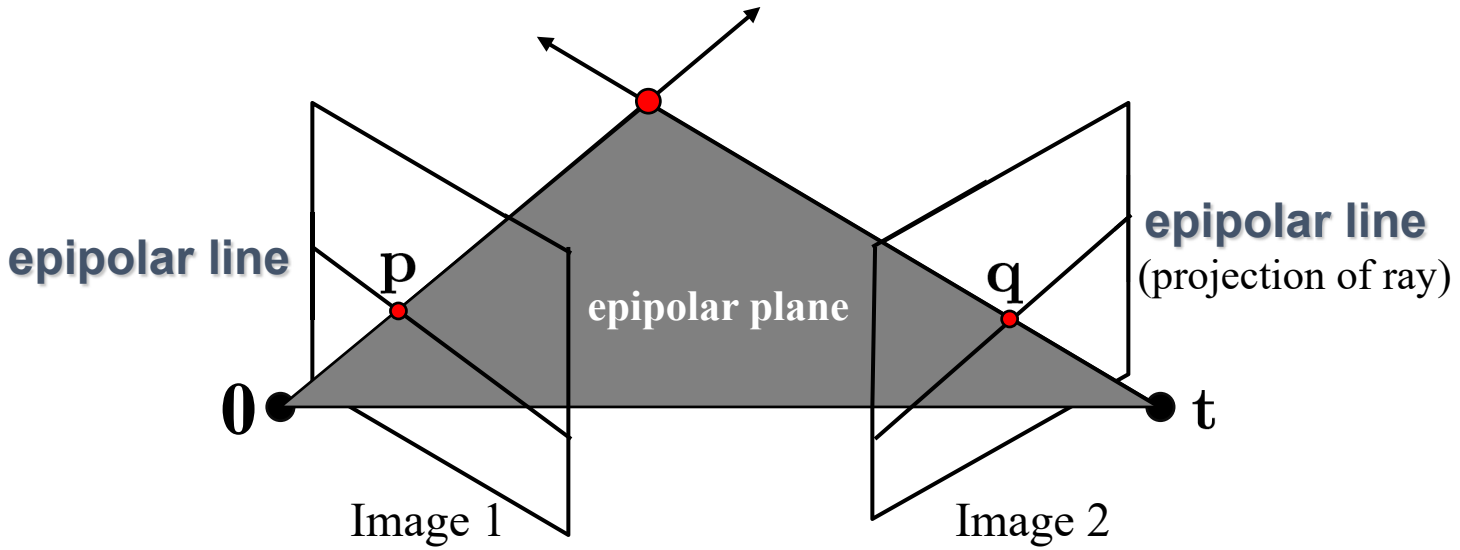
- Where do epipolar lines come from?

Two-view geometry

- Where do epipolar lines come from?



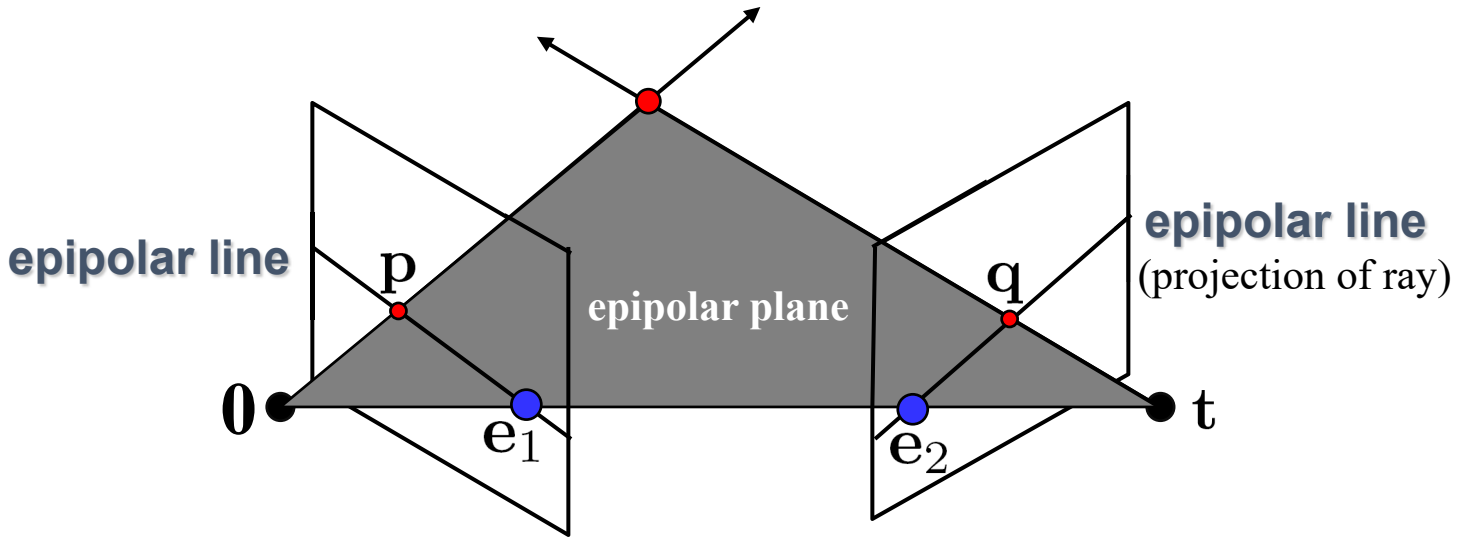
Fundamental matrix



- This *epipolar geometry* of two views is described by a Very Special 3x3 matrix $\mathbf{F}$, called the *fundamental matrix*
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point $\mathbf{p}$ is: $\mathbf{Fp}$

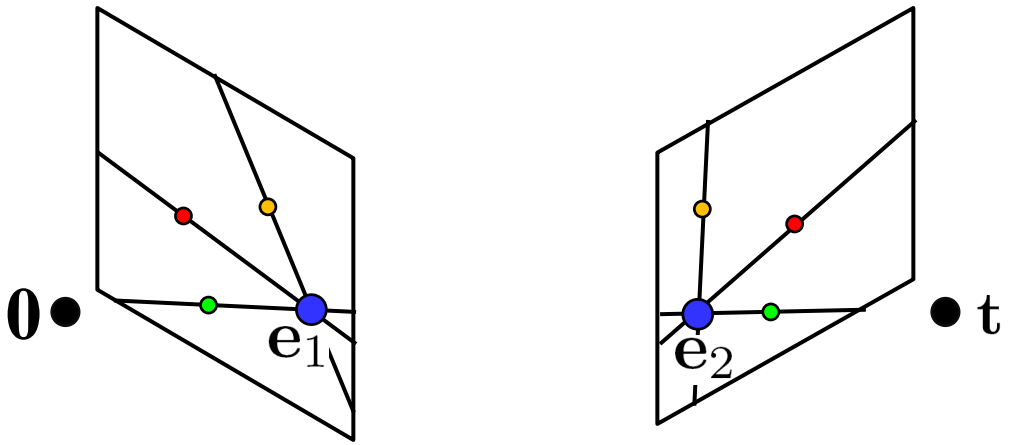
- *Epipolar constraint* on corresponding points: $\mathbf{q}^T\mathbf{Fp} = 0$

Fundamental matrix



- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other

Fundamental matrix



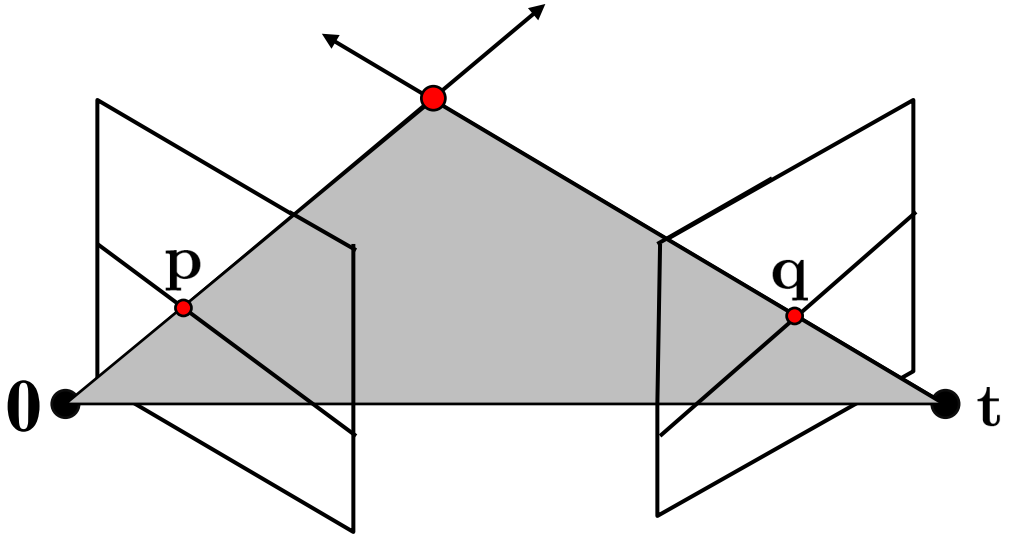
- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole

Epipoles

Properties of the Fundamental Matrix

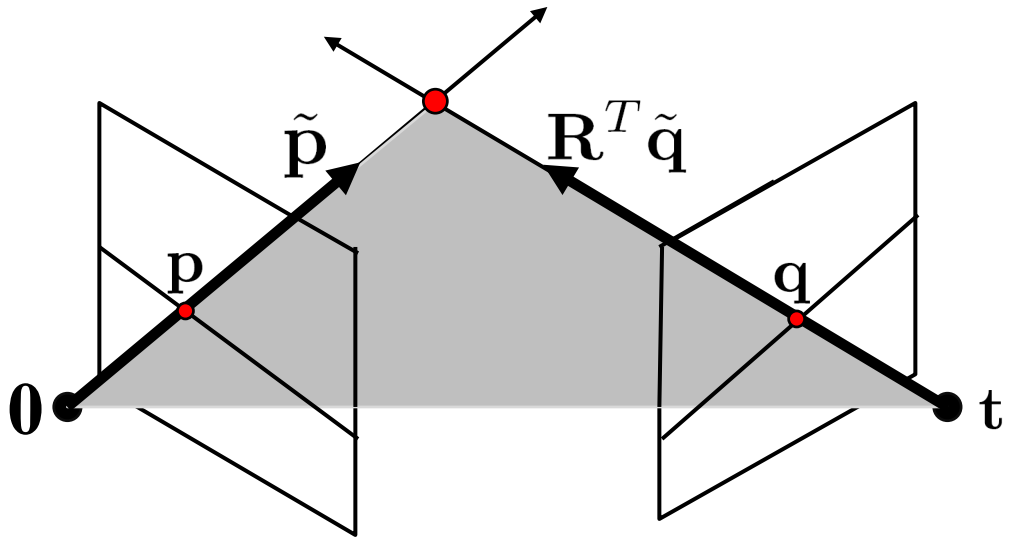
- $\mathbf{F}\mathbf{p}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T\mathbf{q}$ is the epipolar line associated with $\mathbf{q}$
- $\mathbf{F}\mathbf{e}_1 = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}_2 = \mathbf{0}$
- $\mathbf{F}$ is rank 2
- How many degrees of freedom does $\mathbf{F}$ have?

Fundamental matrix



- Why does **F** exist?
- Let's derive it…

Fundamental matrix – calibrated case



$\mathbf{K}_1$: intrinsics of camera 1

$\mathbf{K}_2$: intrinsics of camera 2

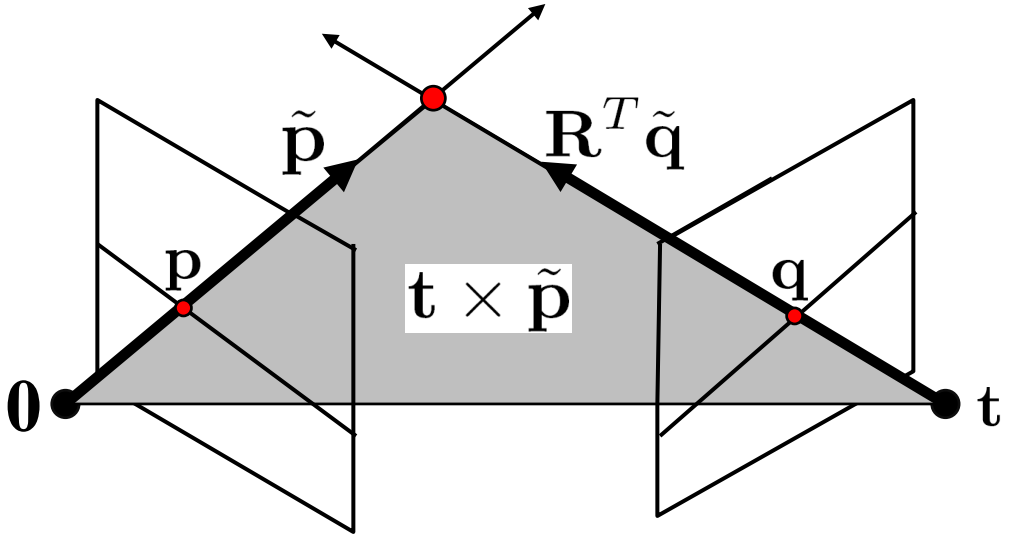
$\mathbf{t}$: translation of cam 2 wrt cam 1

$\mathbf{R}$: rotation of image 2 w.r.t. camera 1

$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1}\mathbf{p}$: ray through **p** in camera 1's (and world) coordinate system

$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1}\mathbf{q}$: ray through **q** in camera 2's coordinate system

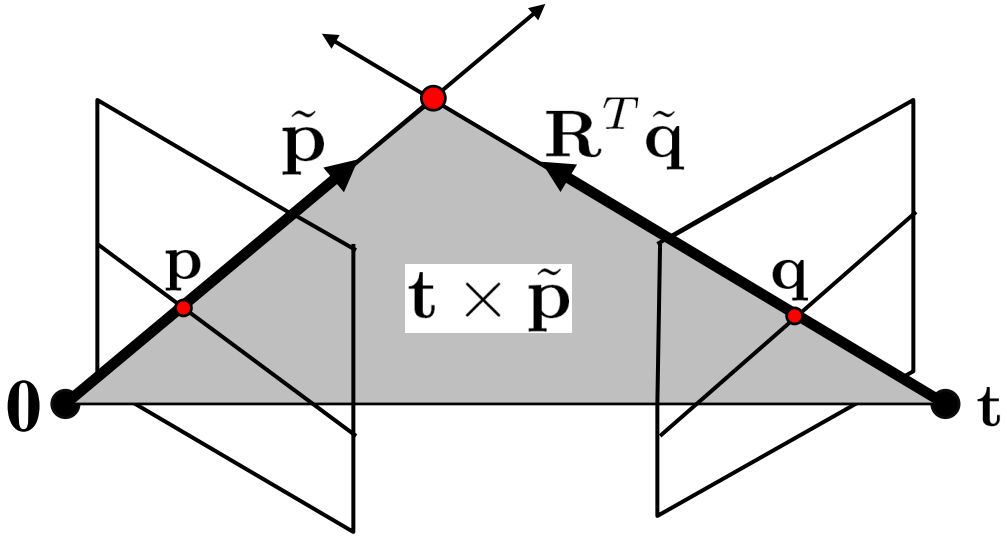
Fundamental matrix – calibrated case



- $\tilde{\mathbf{p}}$, $\mathbf{R}^T\tilde{\mathbf{q}}$, and $\mathbf{t}$ are coplanar
- epipolar plane can be represented as $\mathbf{t}\times\tilde{\mathbf{p}}$

$$(\mathbf{R}^T\tilde{\mathbf{q}})^T(\mathbf{t}\times\tilde{\mathbf{p}})=0$$

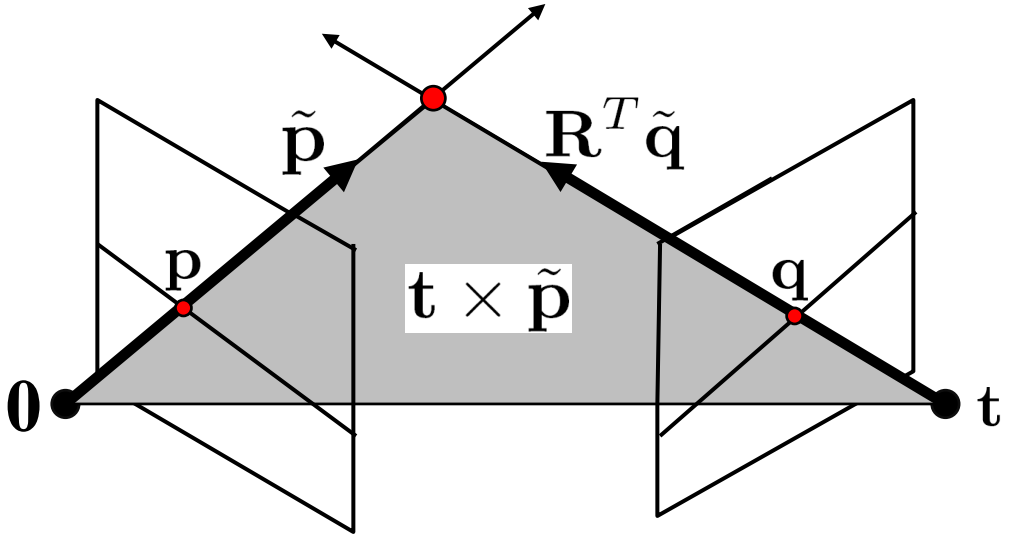
Fundamental matrix – calibrated case



$$(\mathbf{R}^T\tilde{\mathbf{q}})^T(\mathbf{t}\times\tilde{\mathbf{p}})=0$$

$$\tilde{\mathbf{q}}^T\mathbf{R}(\mathbf{t}\times\tilde{\mathbf{p}})=0$$

Fundamental matrix – calibrated case

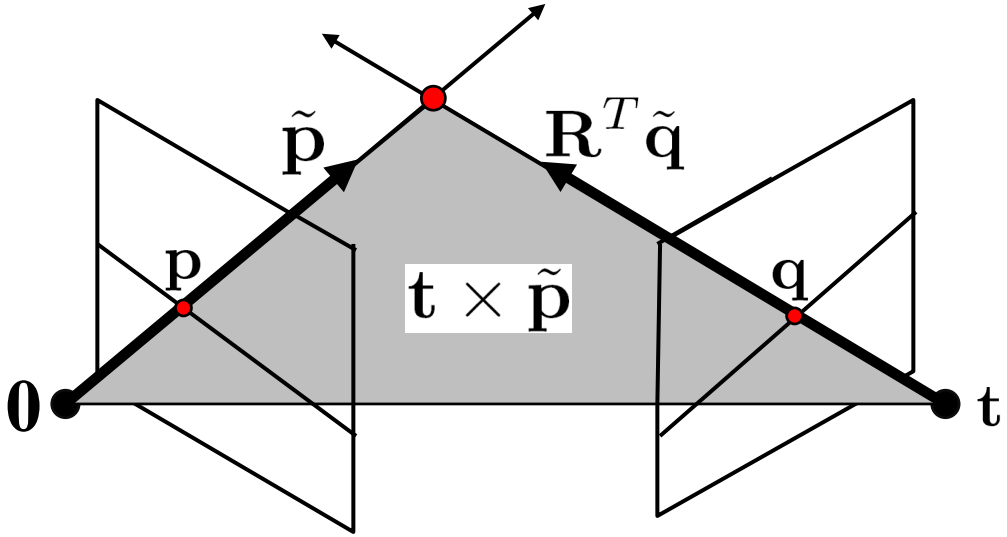


- One more substitution:
  - Cross product with **t** (on left) can be represented as a 3x3 matrix

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$$\mathbf{t}\times\tilde{\mathbf{p}} = [\mathbf{t}]_\times\,\tilde{\mathbf{p}}$$

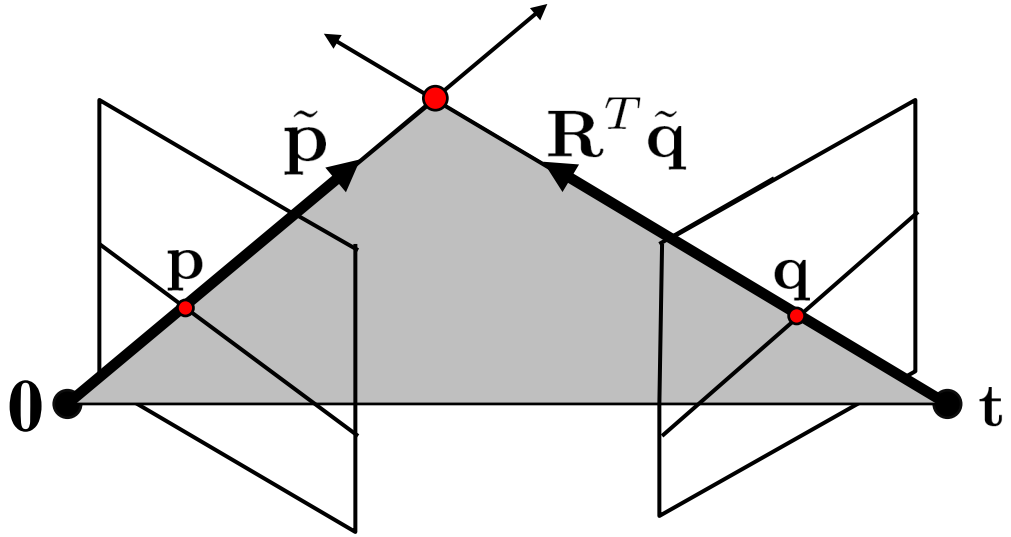
Fundamental matrix – calibrated case



$$\tilde{\mathbf{q}}^T\mathbf{R}(\mathbf{t}\times\tilde{\mathbf{p}})=0$$

$$\tilde{\mathbf{q}}^T\mathbf{R}\,[\mathbf{t}]_\times\,\tilde{\mathbf{p}}=0$$

Fundamental matrix – calibrated case



$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1}\mathbf{p}$: ray through **p** in camera 1's (and world) coordinate system

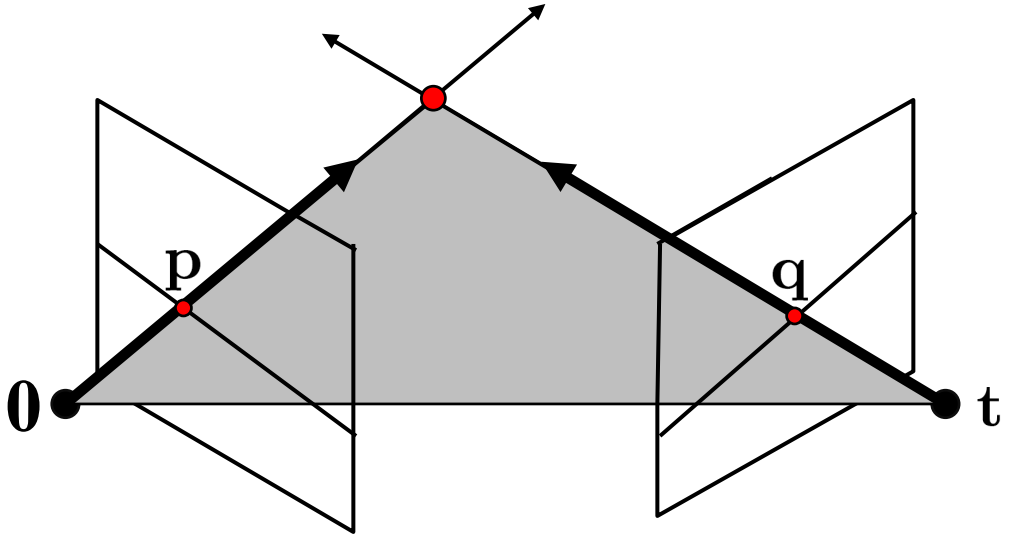
$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1}\mathbf{q}$: ray through **q** in camera 2's coordinate system

$$\tilde{\mathbf{q}}^T \underbrace{\mathbf{R}\,[\mathbf{t}]_\times}_{\mathbf{E}} \tilde{\mathbf{p}} = 0$$

$\mathbf{E}$ ← the "Essential matrix"

$$\tilde{\mathbf{q}}^T \mathbf{E} \tilde{\mathbf{p}} = 0$$

Fundamental matrix – uncalibrated case



$\mathbf{K}_1$: intrinsics of camera 1

$\mathbf{K}_2$: intrinsics of camera 2

$\mathbf{t}$: translation of cam 2 wrt cam 1

$\mathbf{R}$: rotation of image 2 w.r.t. camera 1

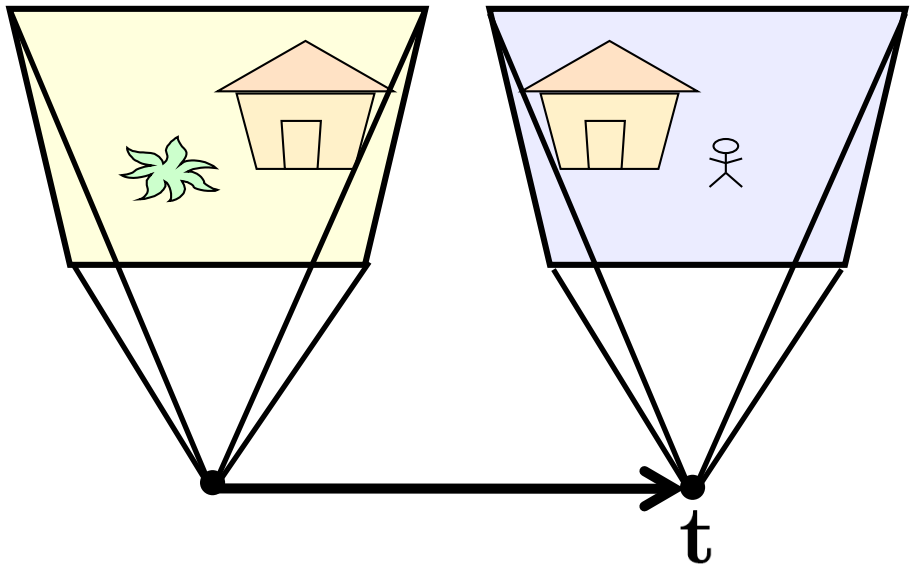
$$\mathbf{q}^T \underbrace{\mathbf{K}_2^{-T} \mathbf{R} \left[\mathbf{t}\right]_\times \mathbf{K}_1^{-1}} \mathbf{p} = 0$$

$\mathbf{F}$ ← the Fundamental matrix

Putting the “Fun” in “Fundamental Matrix”

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

Rectified case



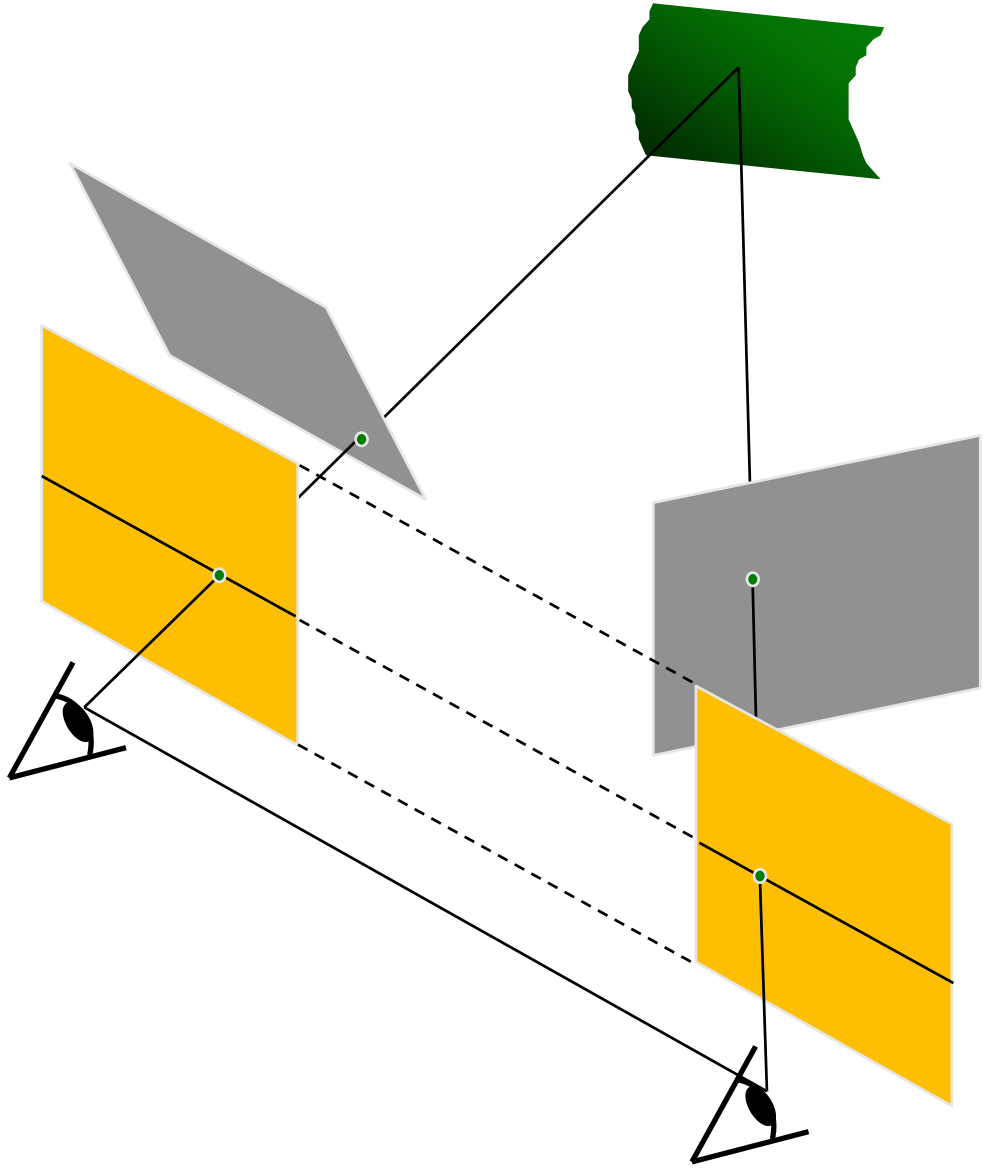
$$\mathbf{R} = \mathbf{I}_{3\times 3}$$

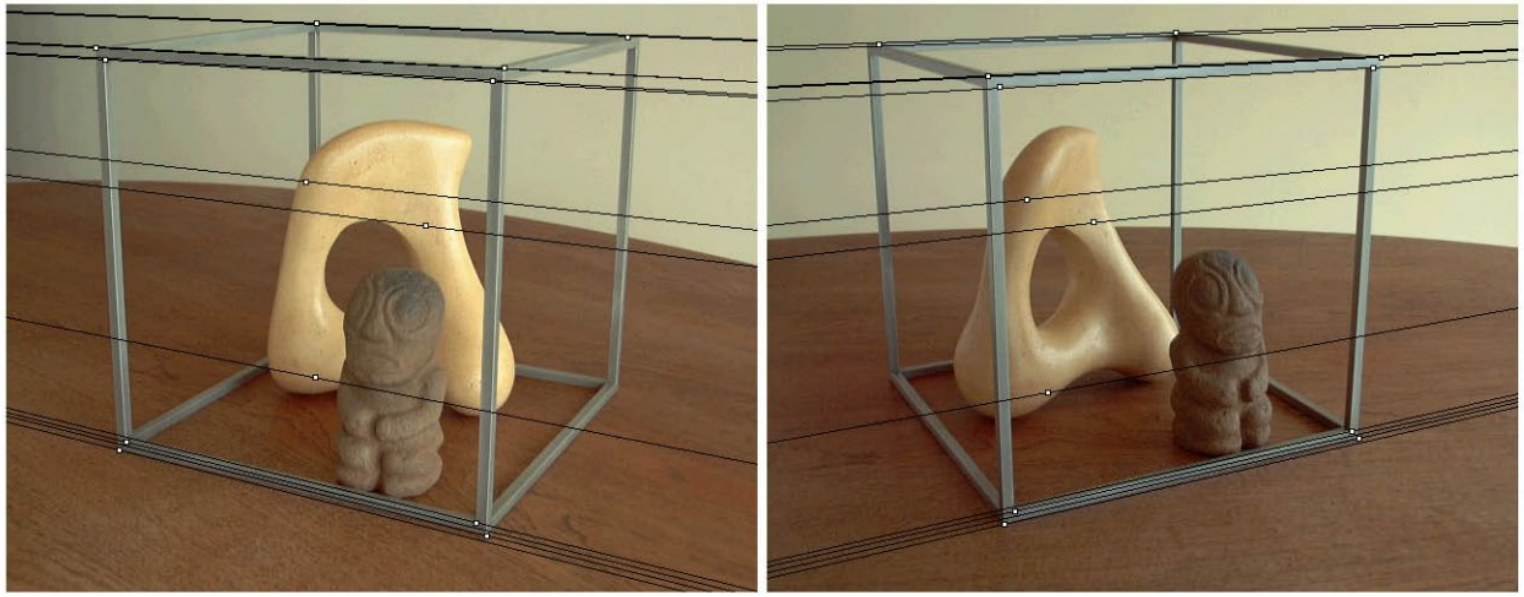
$$\mathbf{t} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

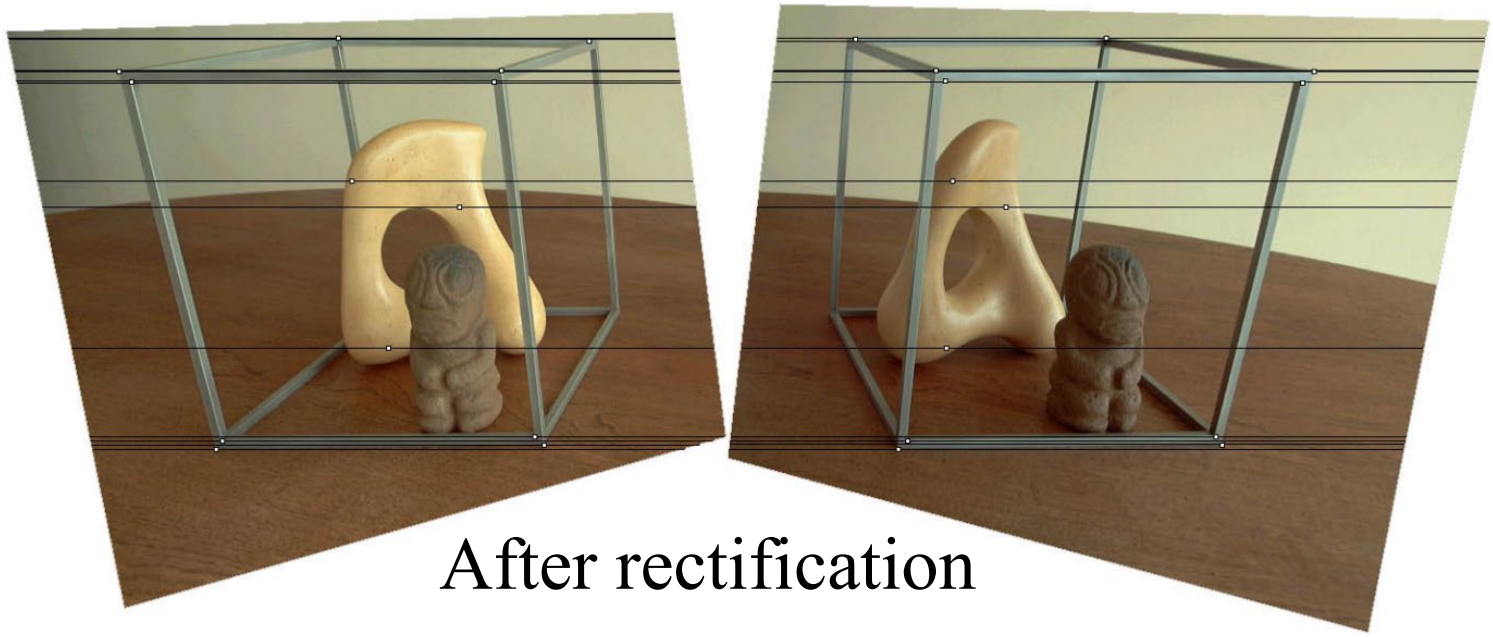
Stereo image rectification

- reproject image planes onto a common plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transform), one for each input image reprojection
- C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



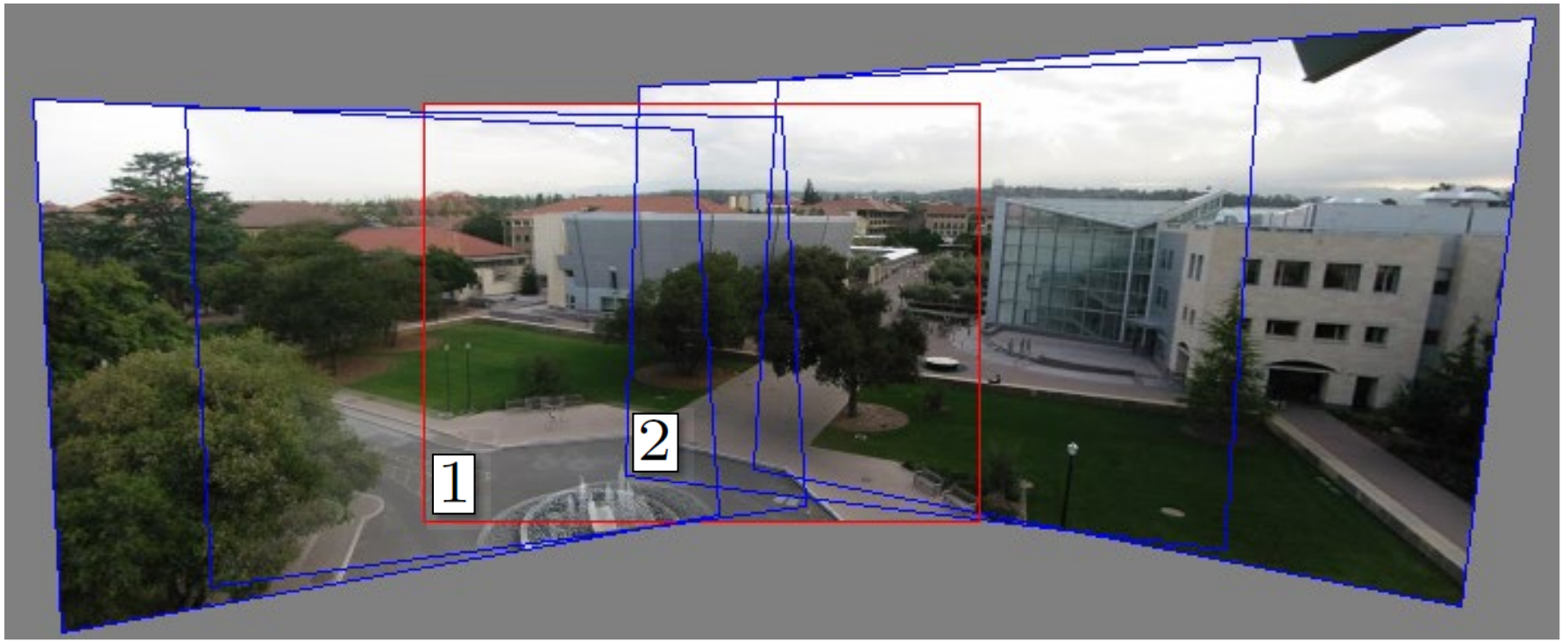


Original stereo pair



After rectification

Relationship with homography?



Images taken from the same center of projection? Use a homography!

Questions?

Estimating F

- If we don't know $\mathbf{K}_1$, $\mathbf{K}_2$, $\mathbf{R}$, or $\mathbf{t}$, can we estimate $\mathbf{F}$ for two images?

- Yes, given enough correspondences

Estimating F – 8-point algorithm

- The fundamental matrix F is defined by

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

for any pair of matches x and x’ in two images.

- Let x=$(u,v,1)^{\mathrm{T}}$ and x’=$(u’,v’,1)^{\mathrm{T}}$, $\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

each match gives a linear 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

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

- Like with homographies, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$, least eigenvector of $\mathbf{A}^T\mathbf{A}$.

8-point algorithm – Problem?

- $\mathbf{F}$ should have rank 2
- To enforce that $\mathbf{F}$ is of rank 2, F is replaced by F' that minimizes $\|\mathbf{F} - \mathbf{F}'\|$ subject to the rank constraint.

- This is achieved by SVD. Let $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \text{ let } \quad \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

then $\mathbf{F}' = \mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}$ is the solution.

8-point algorithm

```
% Build the constraint matrix
A = [x2(1,:)'.*x1(1,:)'  x2(1,:)'.*x1(2,:)'  x2(1,:)' ...
     x2(2,:)'.*x1(1,:)'  x2(2,:)'.*x1(2,:)'  x2(2,:)' ...
     x1(1,:)'            x1(2,:)'            ones(npts,1) ];

[U,D,V] = svd(A);

% Extract fundamental matrix from the column of V
% corresponding to the smallest singular value.
F = reshape(V(:,9),3,3)';

% Enforce rank2 constraint
[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';
```

8-point algorithm

- Pros: it is linear, easy to implement and fast
- Cons: susceptible to noise

Problem with 8-point algorithm

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

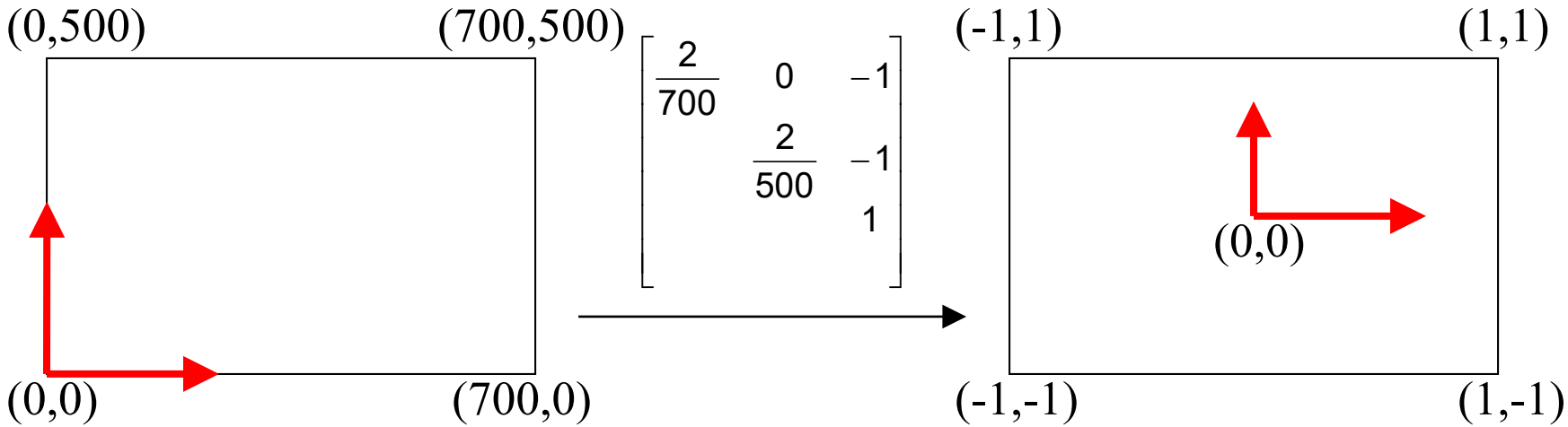
~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

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

Normalized 8-point algorithm

normalized least squares yields good results

Transform image to ~[-1,1]x[-1,1]



Normalized 8-point algorithm

1. Transform input by $\hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i, \quad \hat{\mathbf{x}}'_i = \mathbf{T}\mathbf{x}'_i$
2. Call 8-point on $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$ to obtain $\hat{\mathbf{F}}$
3. $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}$

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

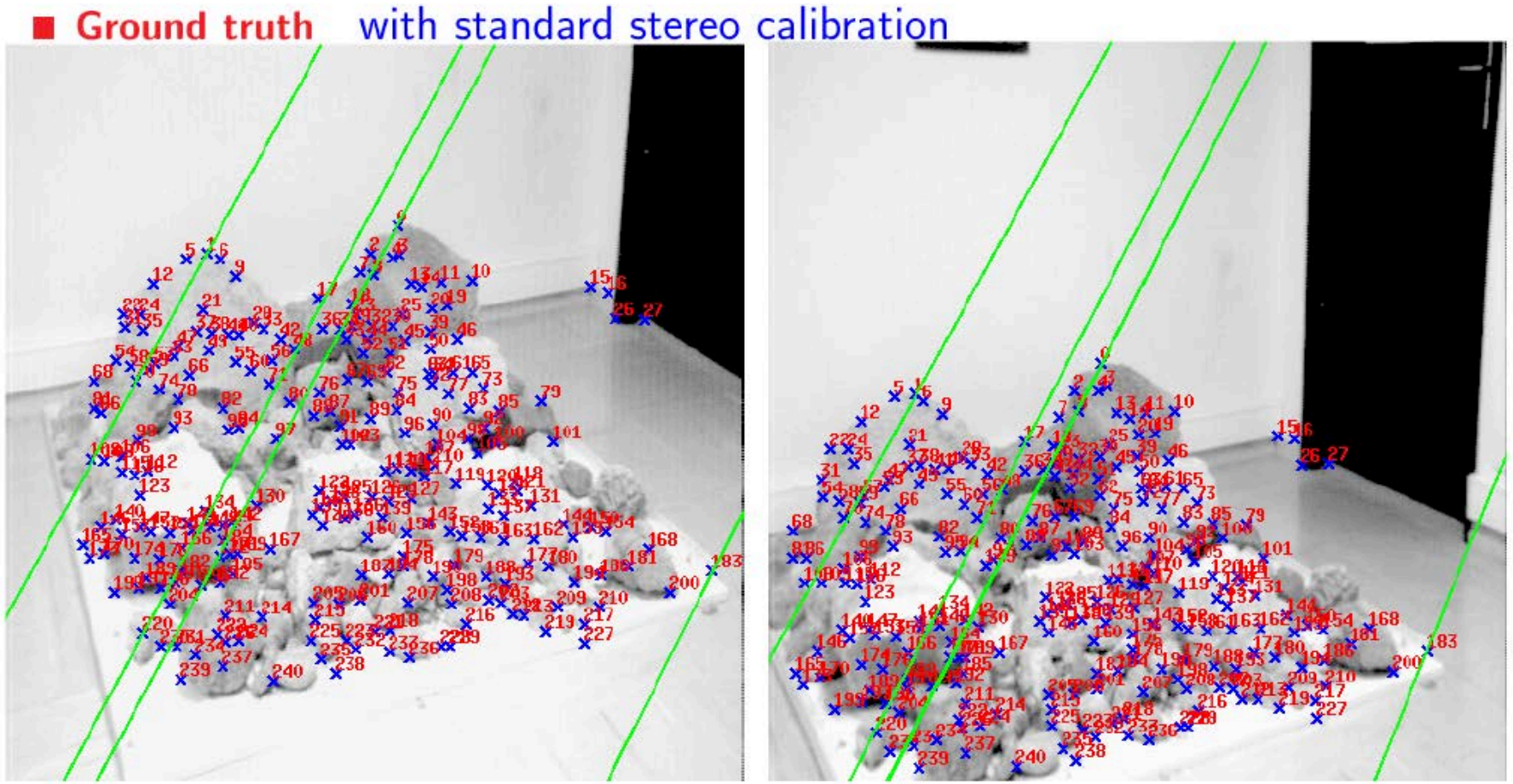
$$\hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{T}'^{-\mathrm{T}}\mathbf{F}\mathbf{T}^{-1}\hat{\mathbf{x}} = 0$$

$$\hat{\mathbf{F}}$$

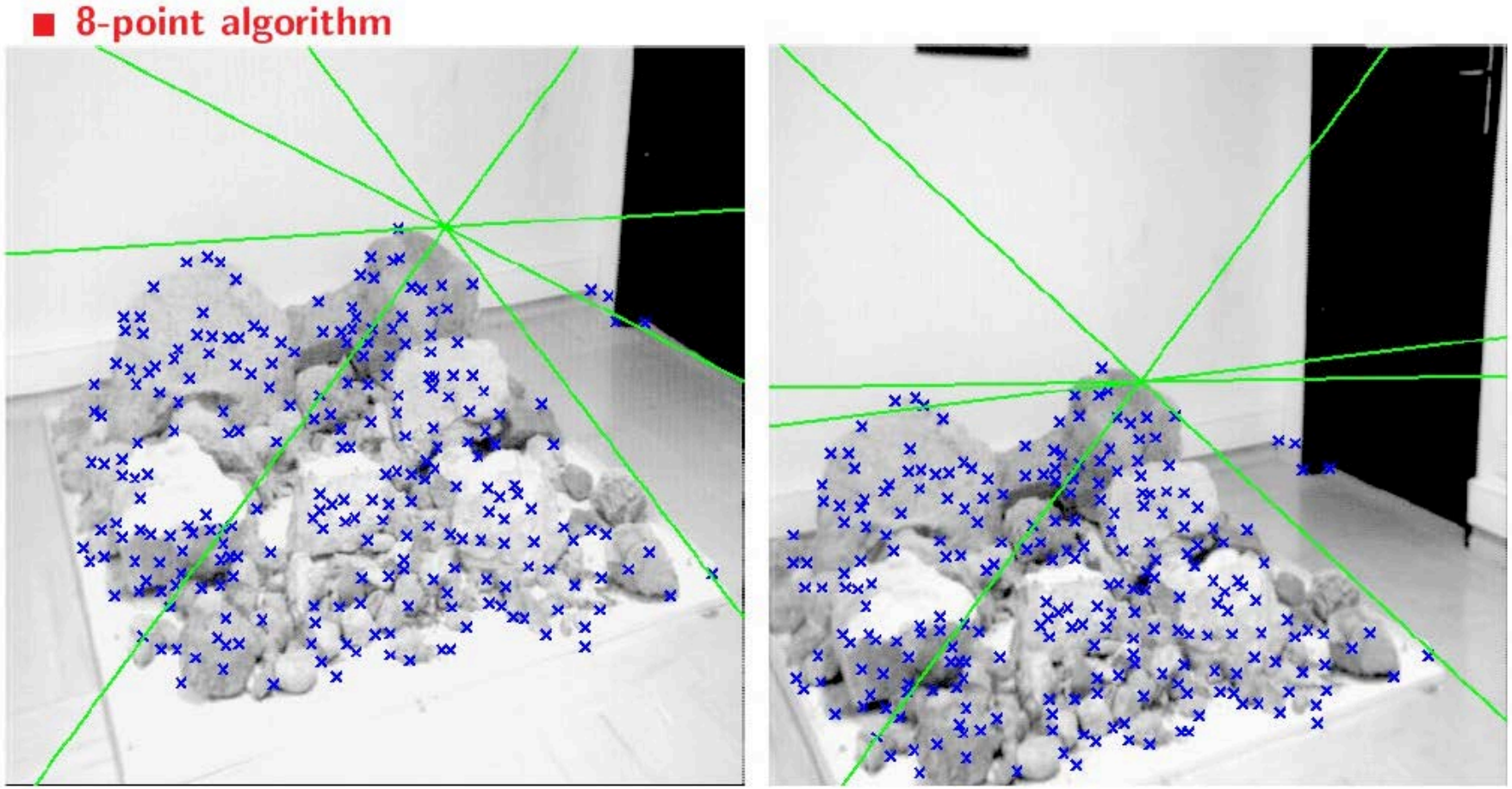
Normalized 8-point algorithm

```
[x1, T1] = normalise2dpts(x1);
[x2, T2] = normalise2dpts(x2);

A = [x2(1,:)'.*x1(1,:)'   x2(1,:)'.*x1(2,:)'  x2(1,:)' ...
     x2(2,:)'.*x1(1,:)'   x2(2,:)'.*x1(2,:)'  x2(2,:)' ...
     x1(1,:)'             x1(2,:)'            ones(npts,1) ];

[U,D,V] = svd(A);

F = reshape(V(:,9),3,3)';

[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';
% Denormalise
F = T2'*F*T1;
```

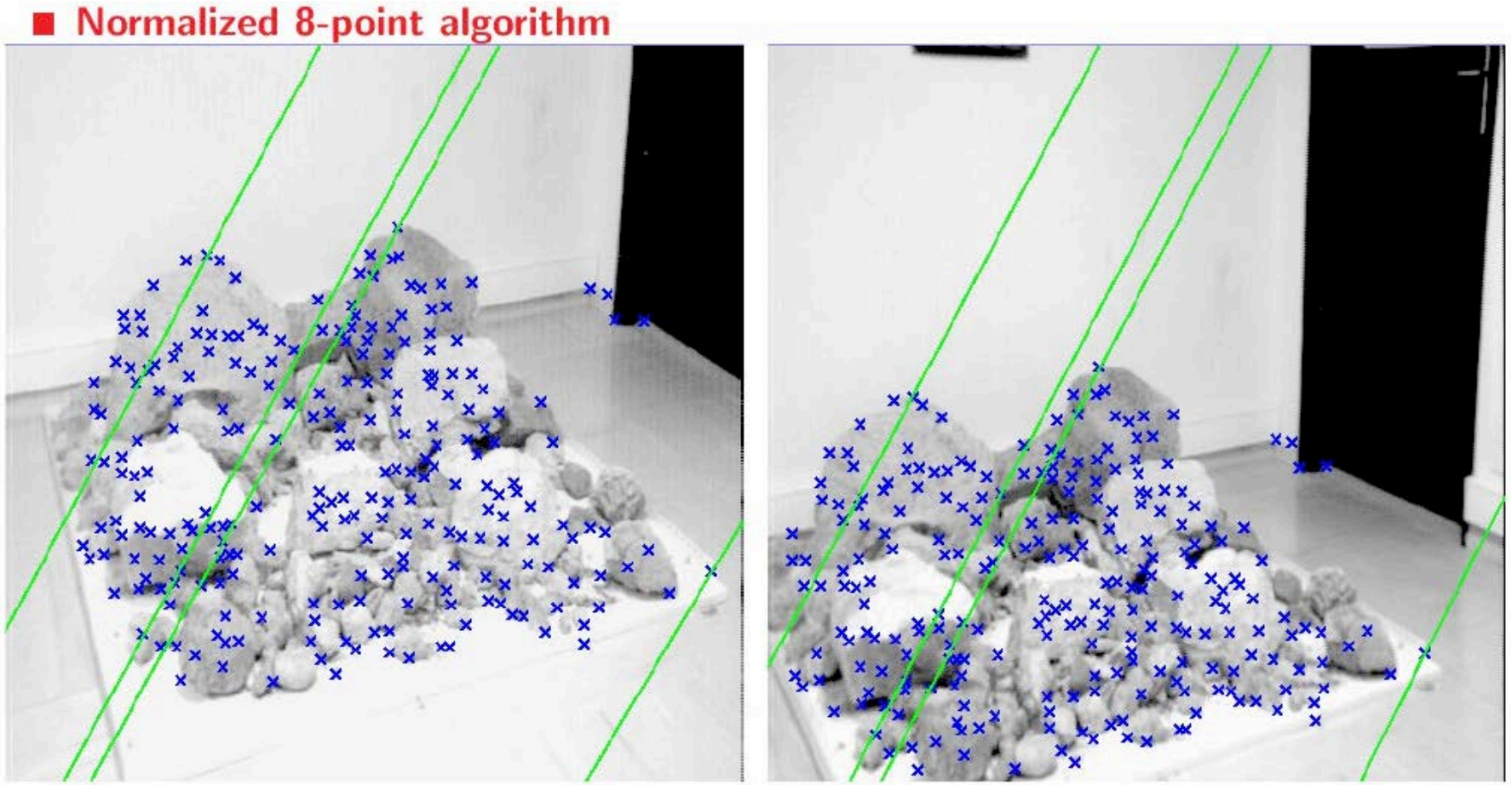
Results (ground truth)



Results (8-point algorithm)



Results (normalized 8-point algorithm)



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…

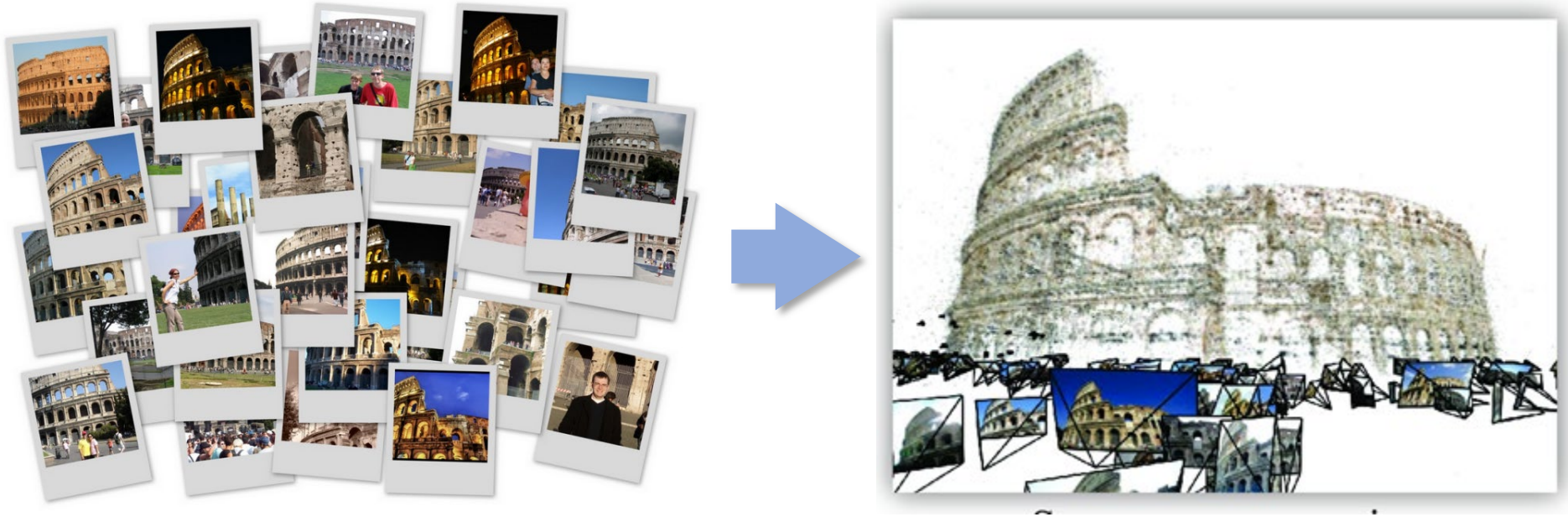
Large-scale structure from motion

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

CS5670: Computer Vision

Abe Davis, Slides by Noah Snavely

Structure from motion



Readings

- Szeliski, Chapter 7.1 – 7.4

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?

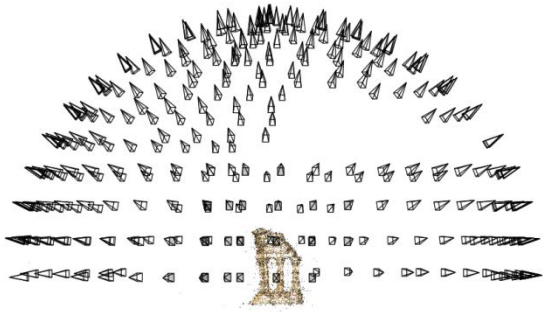
- The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*
- The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*
- After this it starts to get complicated…
  - Instead, we explicitly solve for camera poses and scene geometry

Structure from motion

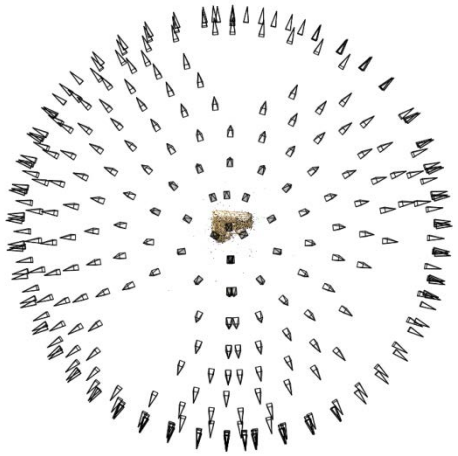
- Given many images, how can we
  - a) figure out where they were all taken from?
  - b) build a 3D model of the scene?

- 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

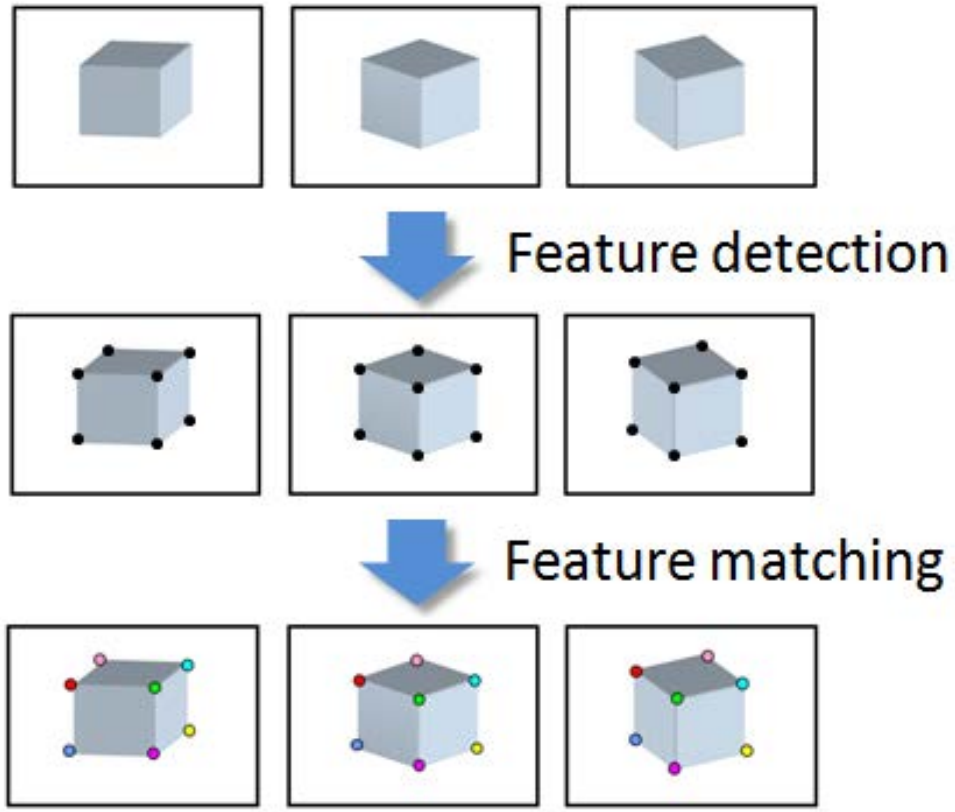
- Translations, affine transformations, homographies…

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 known 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

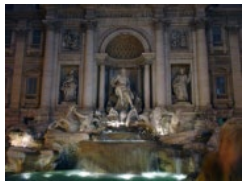
Feature Detection

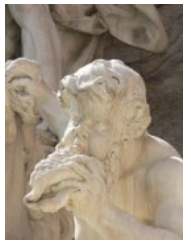
- Detect features using SIFT [Lowe, IJCV 2004]





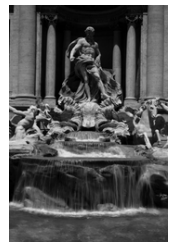




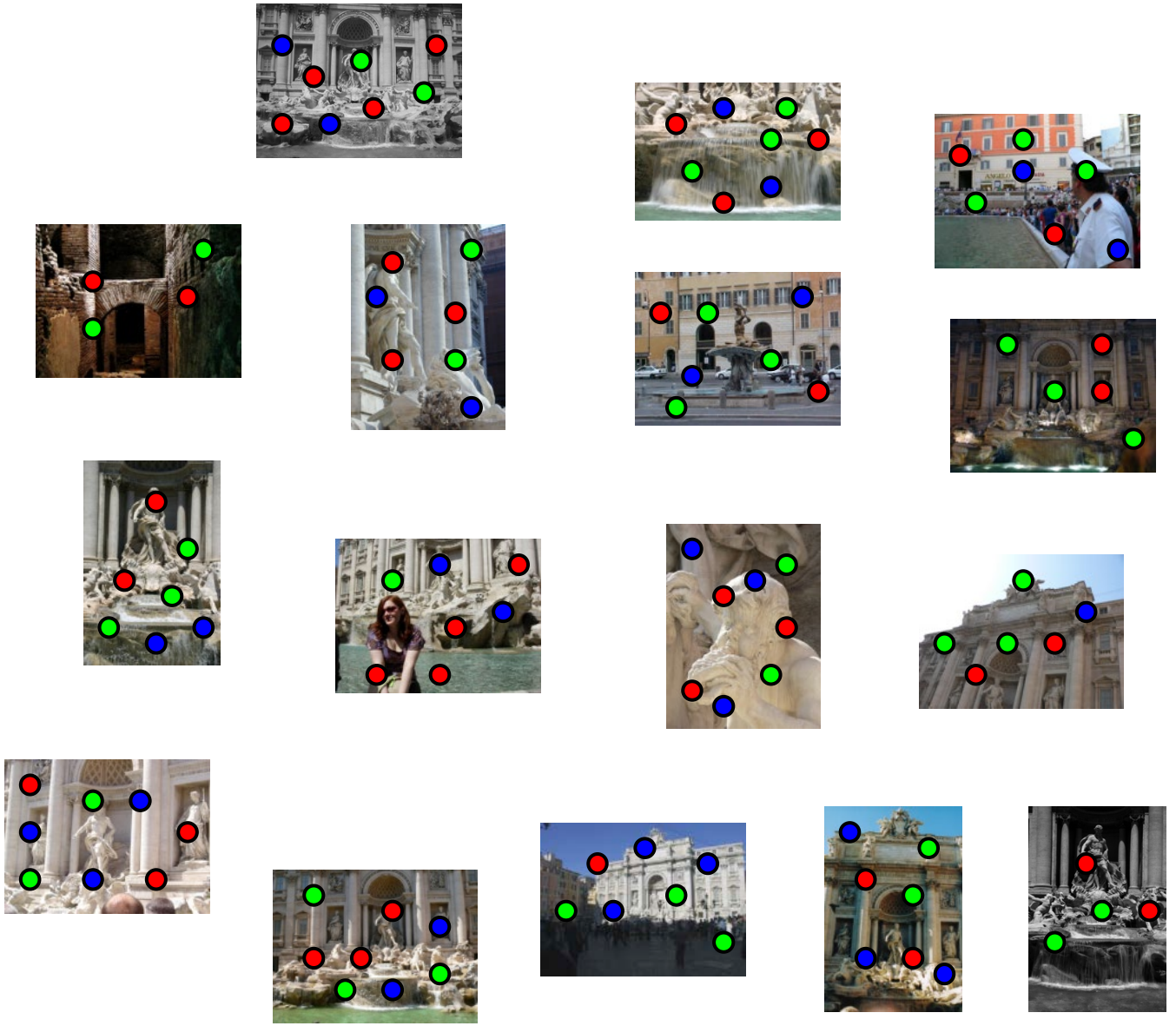






Feature Detection

- Detect features using SIFT [Lowe, IJCV 2004]



Feature Matching

- Detect features using SIFT [Lowe, IJCV 2004]
- Match features between each pair of images
- Refine matching using RANSAC to estimate fundamental matrix between each pair

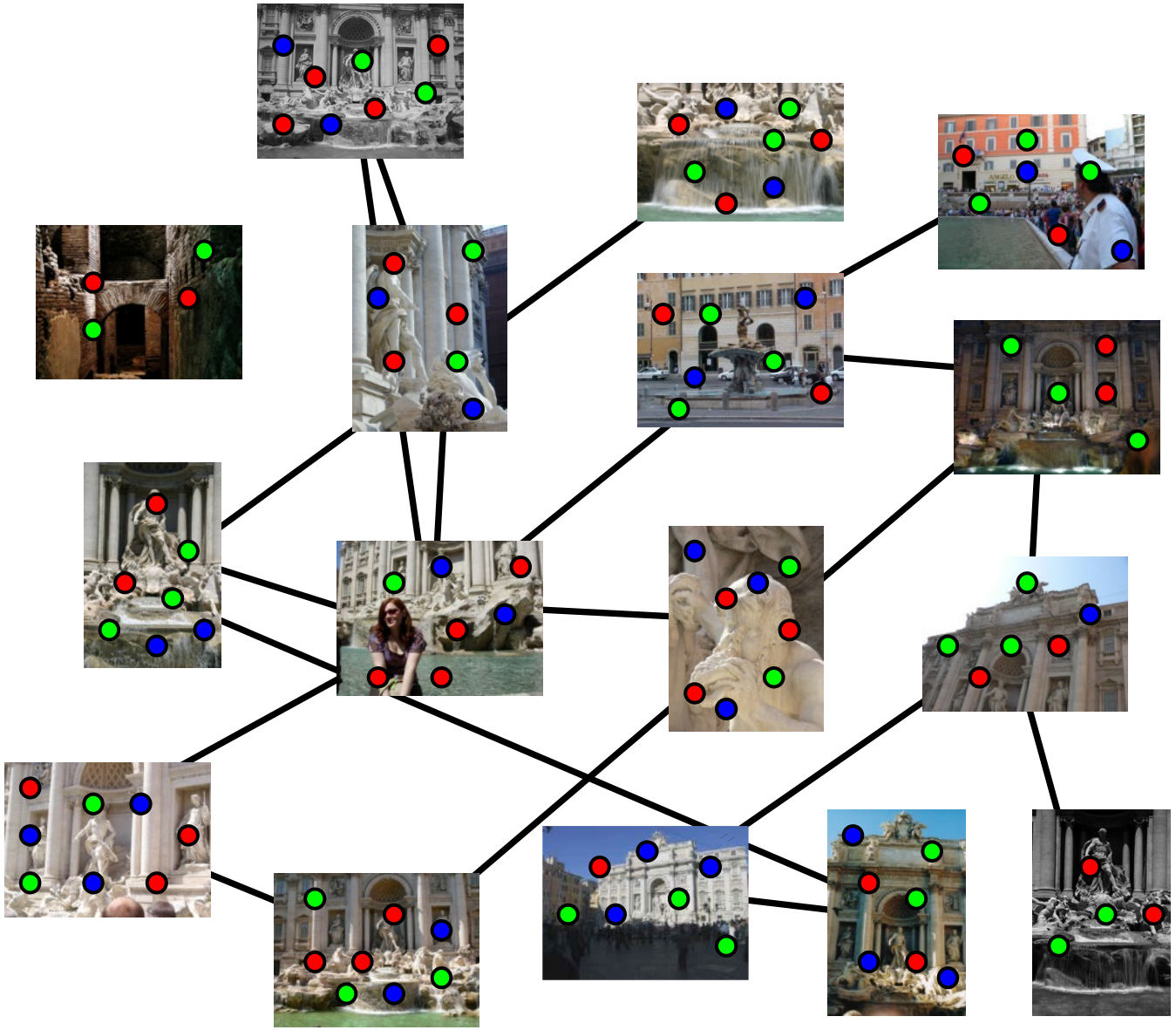
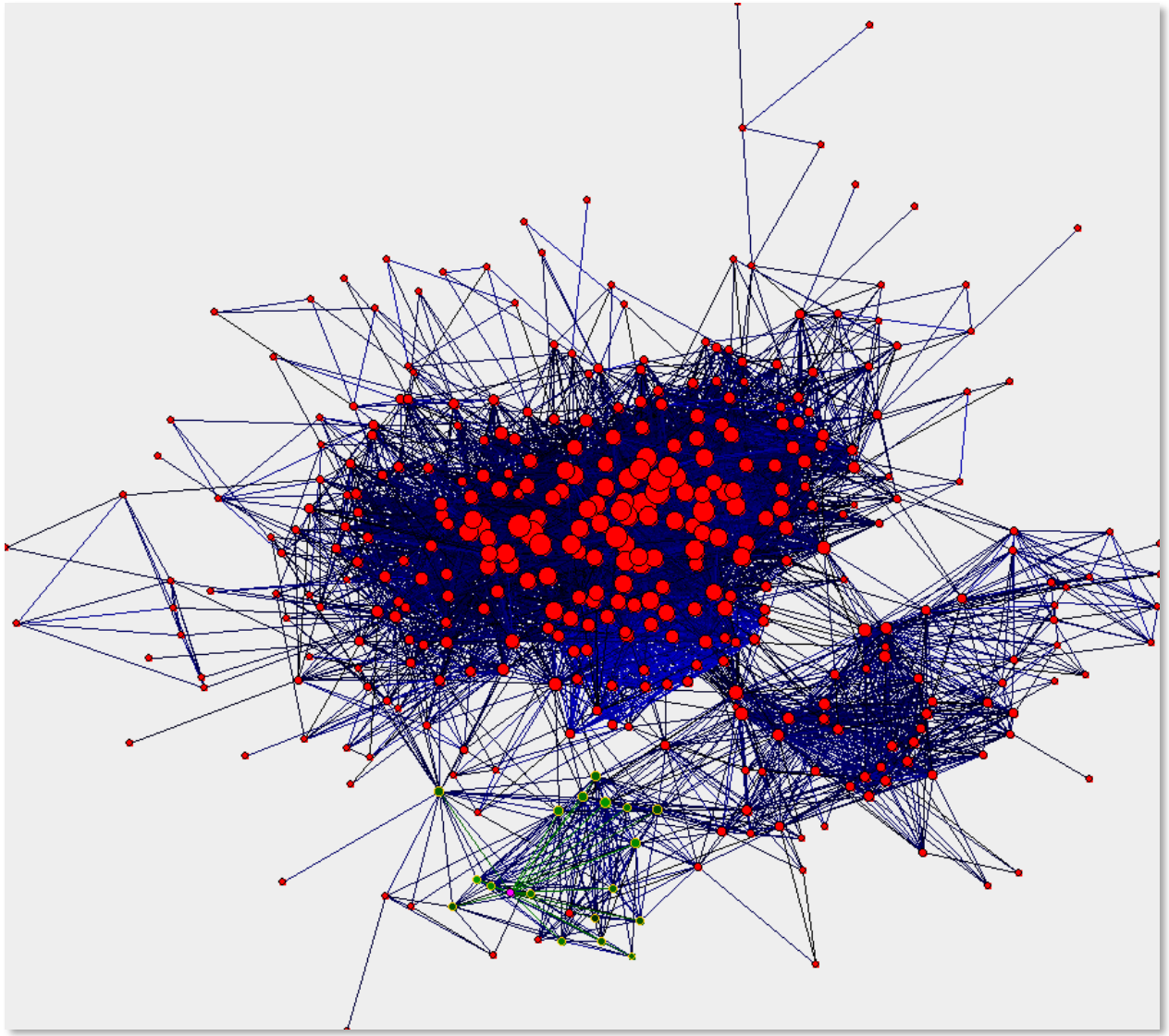


Image connectivity graph

- Graph of connectivity based on matched features



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

Correspondence estimation

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

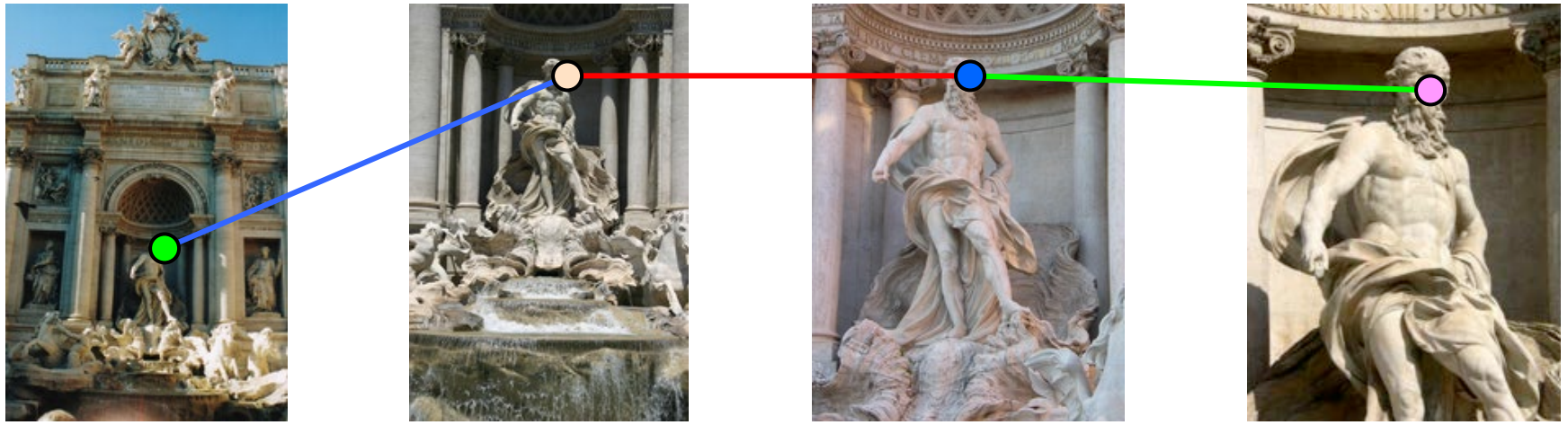
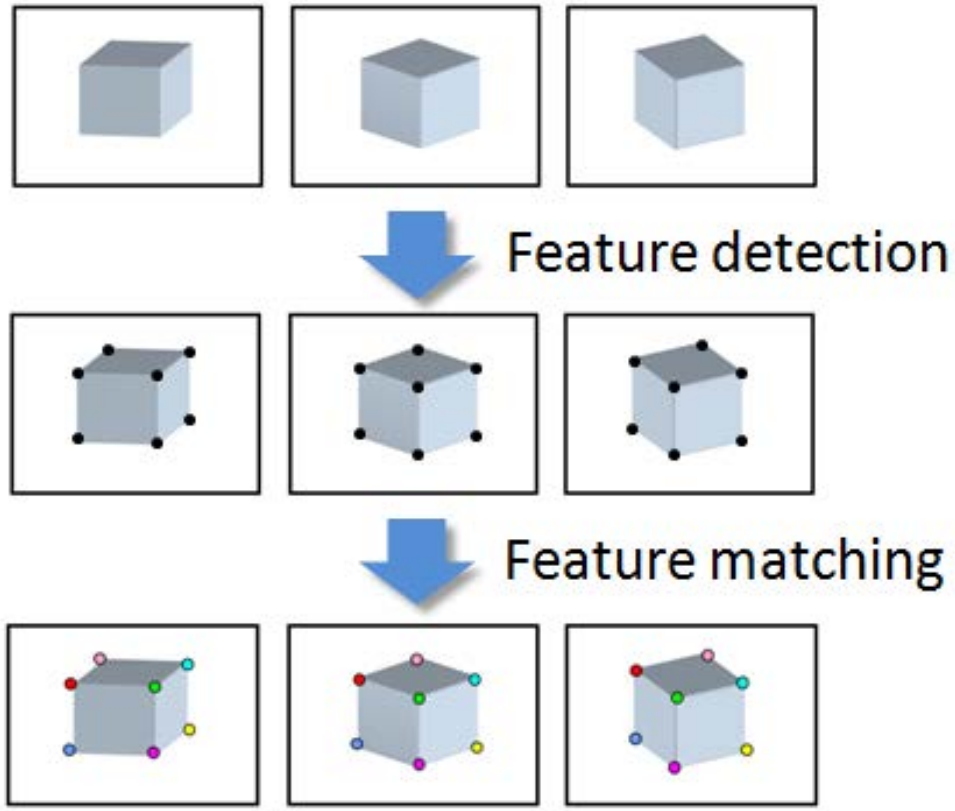
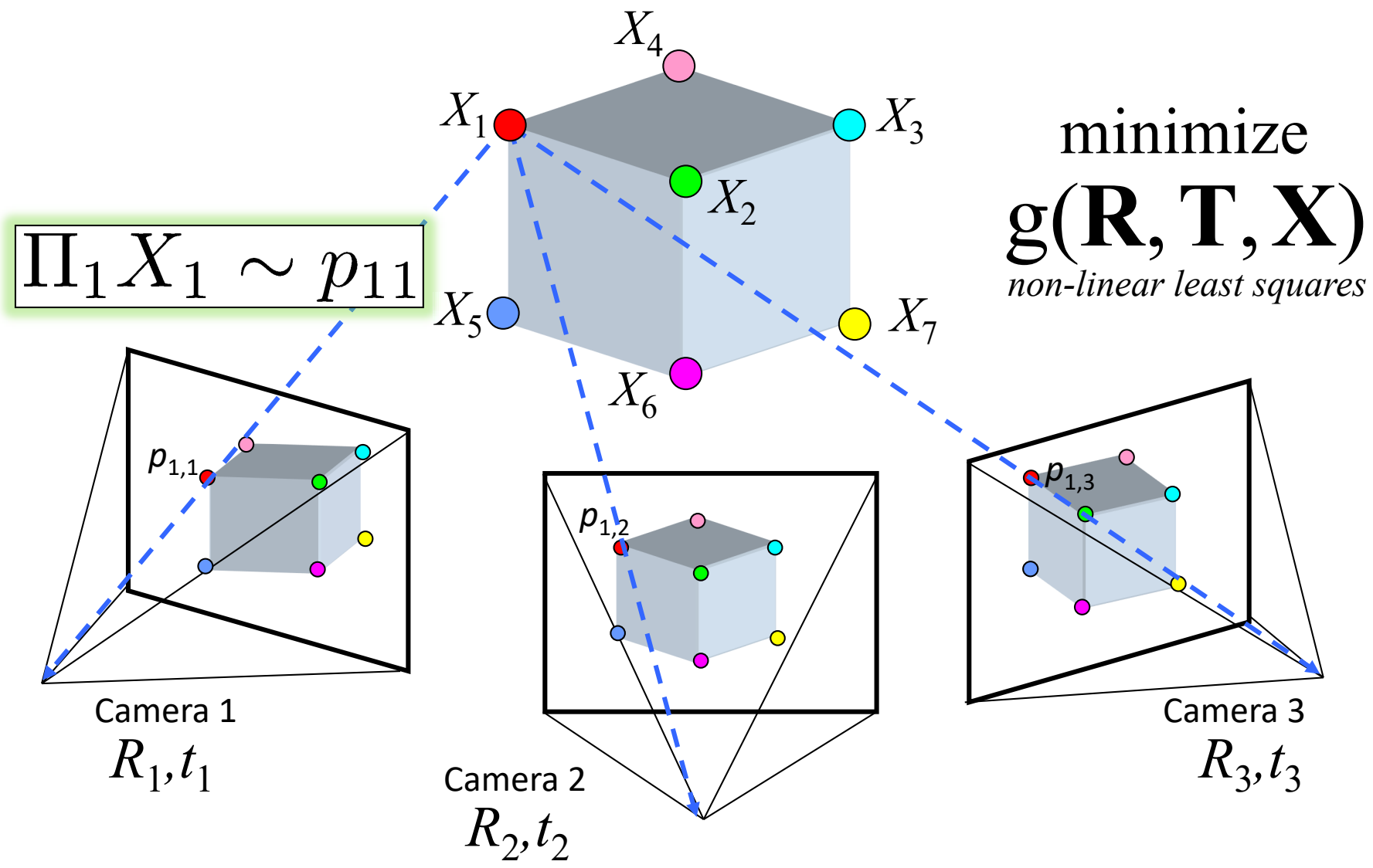


Image 1 Image 2 Image 3 Image 4

Input



Structure from motion



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:

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

predicted image location

observed image location

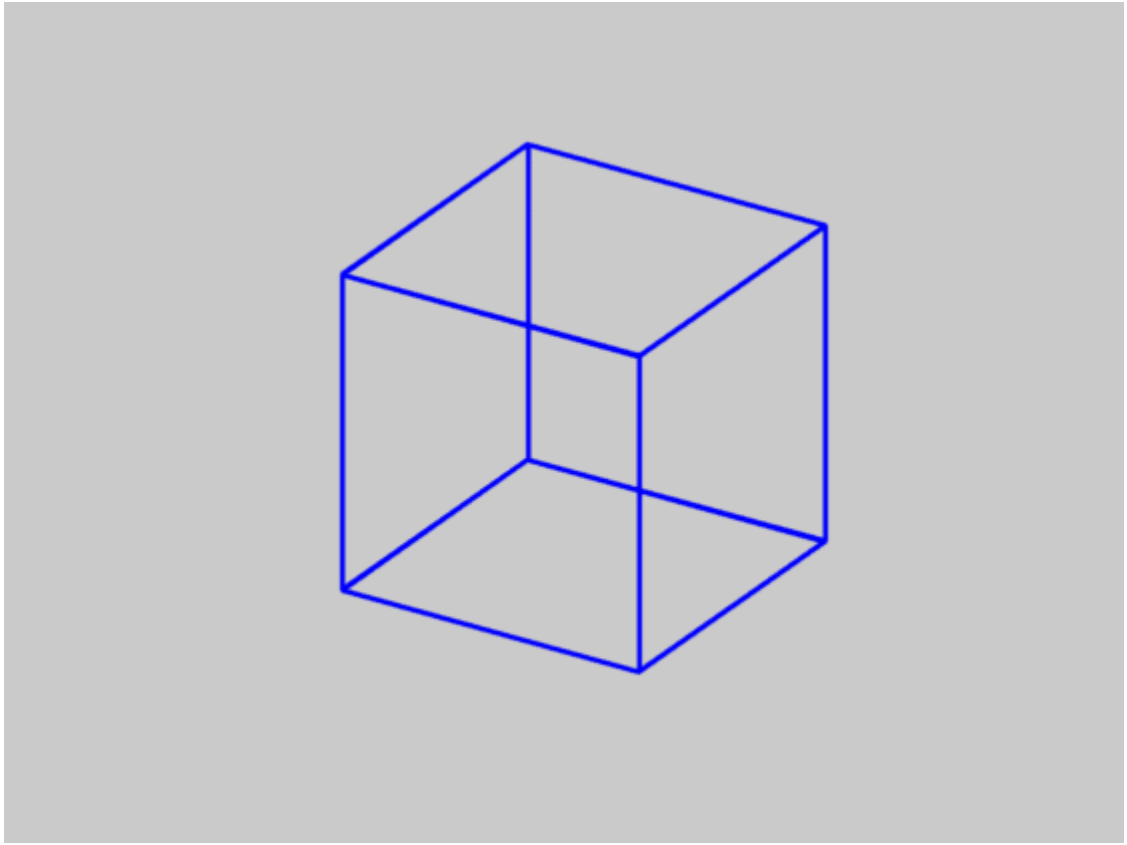
indicator variable: is point *i* visible in image *j* ?

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

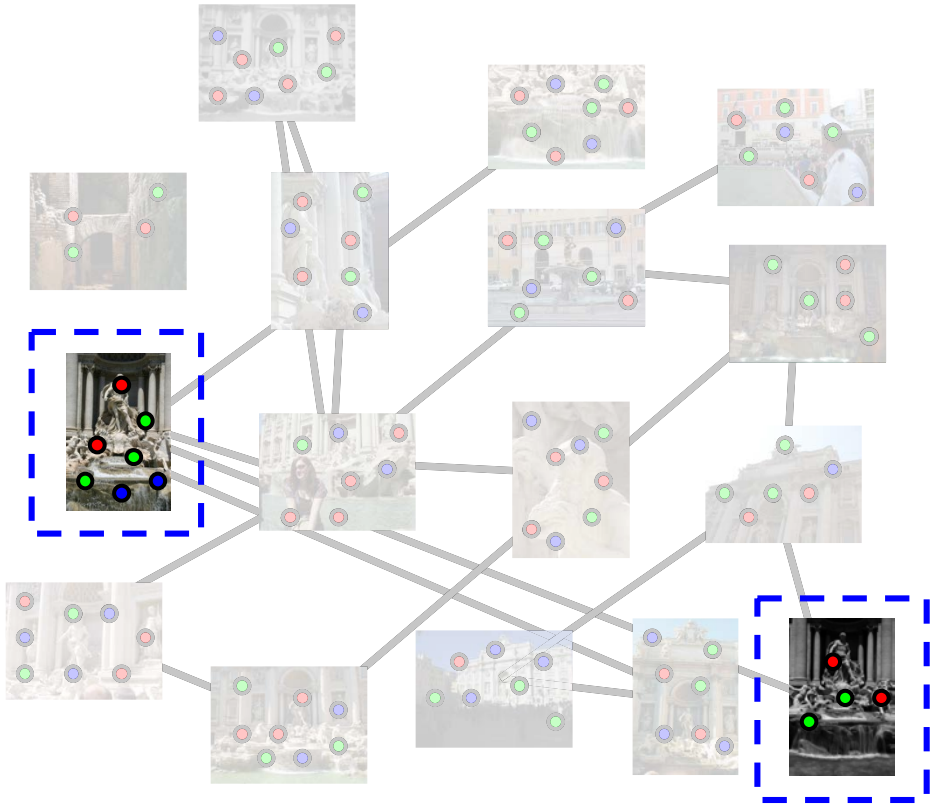
Is SfM always uniquely solvable?

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 (Part of Noah’s PhD work)

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]

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

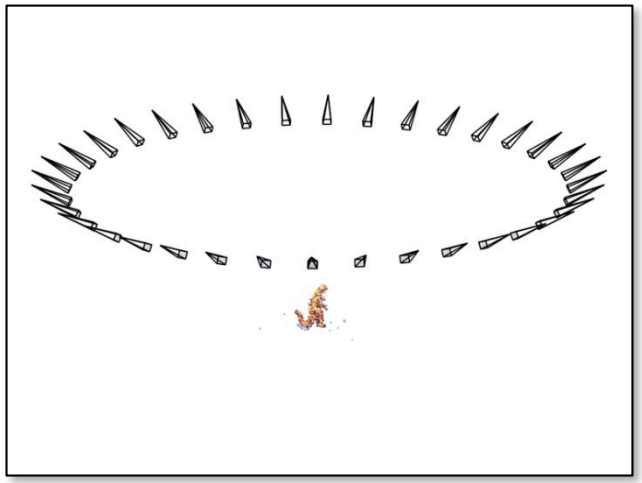
Simulated views of the Moon over one month, demonstrating librations in latitude and longitude. Also visible are

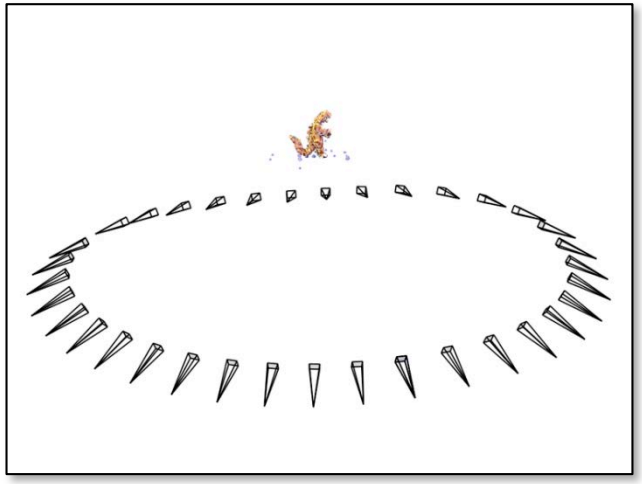
https://en.wikipedia.org/wiki/Libration

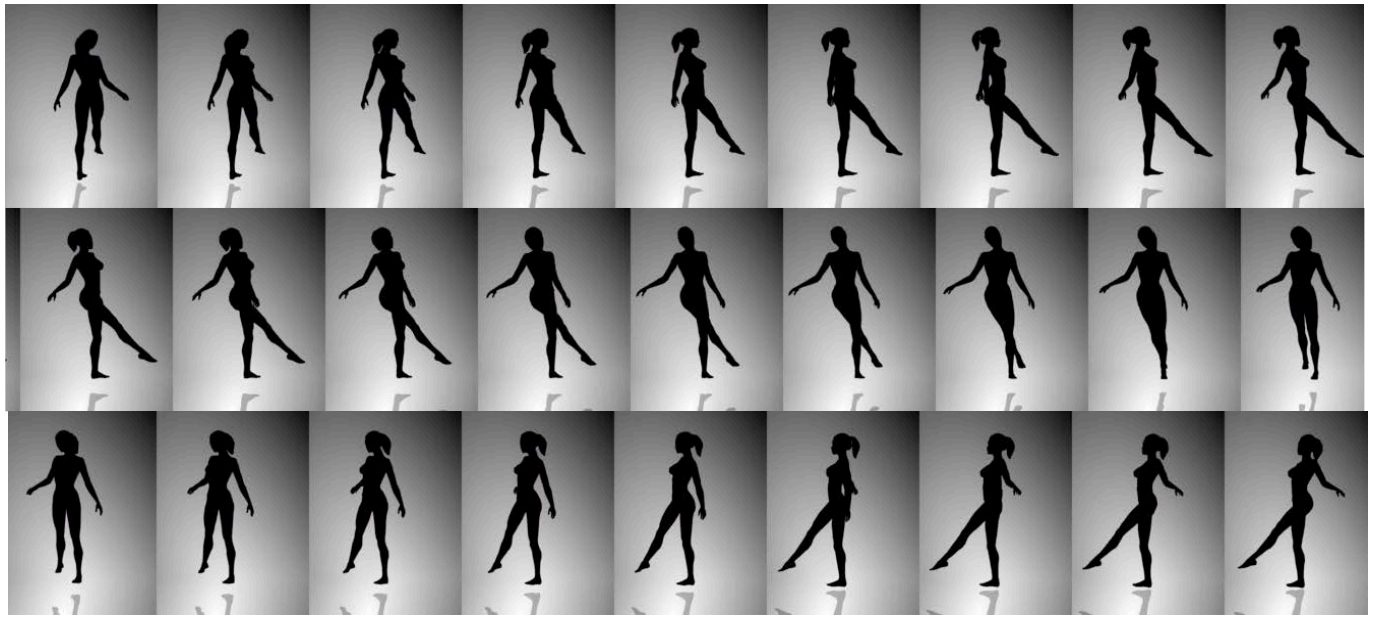
Questions?

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

PhotoTourism

Applications: Visual Reality & Augmented Reality

Oculus

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

Hololens

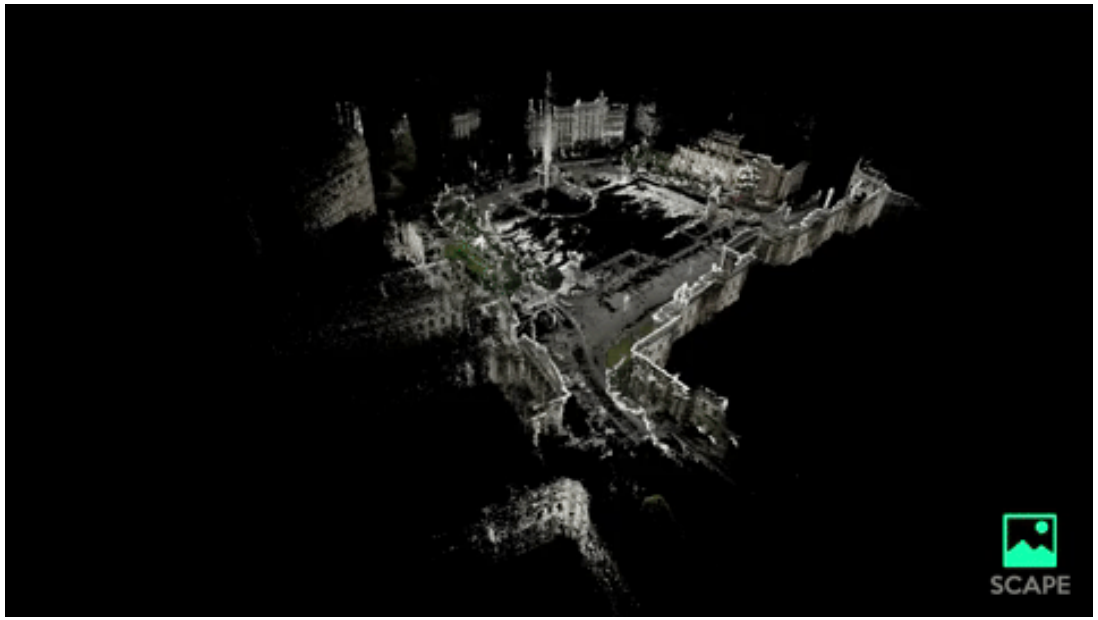
https://www.youtube.com/watch?v=FMtvrTGnP04

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?