Structure from Motion

Computer Vision I

CSE 252A

Lecture 10

Announcements

- Assignment 2 is due Nov 8, 11:59 PM
- Assignment 3 will be released Nov 8
  - Due Nov 22, 11:59 PM

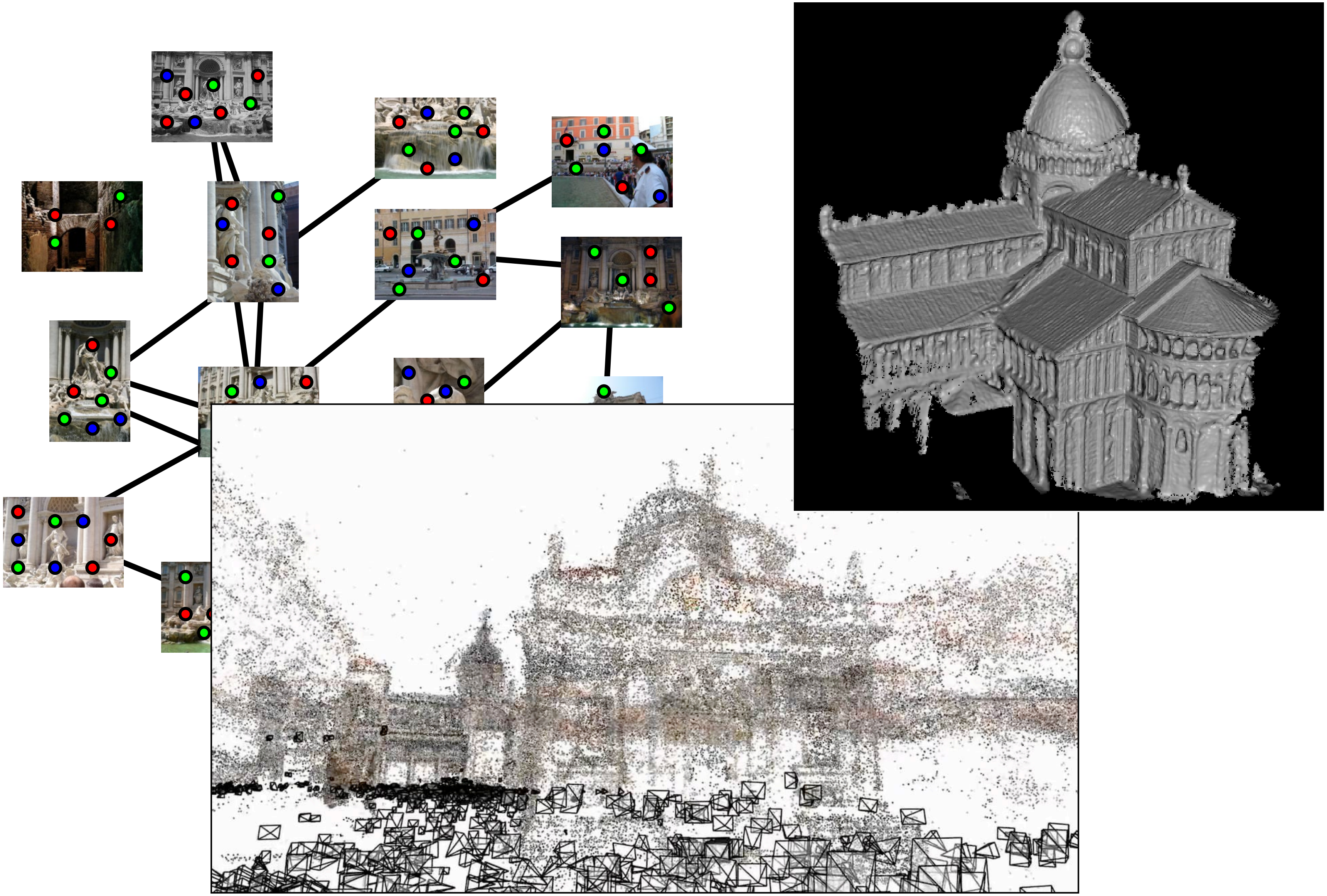
Structure-from-Motion (SFM)

Given two or more images or video without any information on camera position/motion as input, estimate camera motion and 3-D structure of a scene

Two Approaches

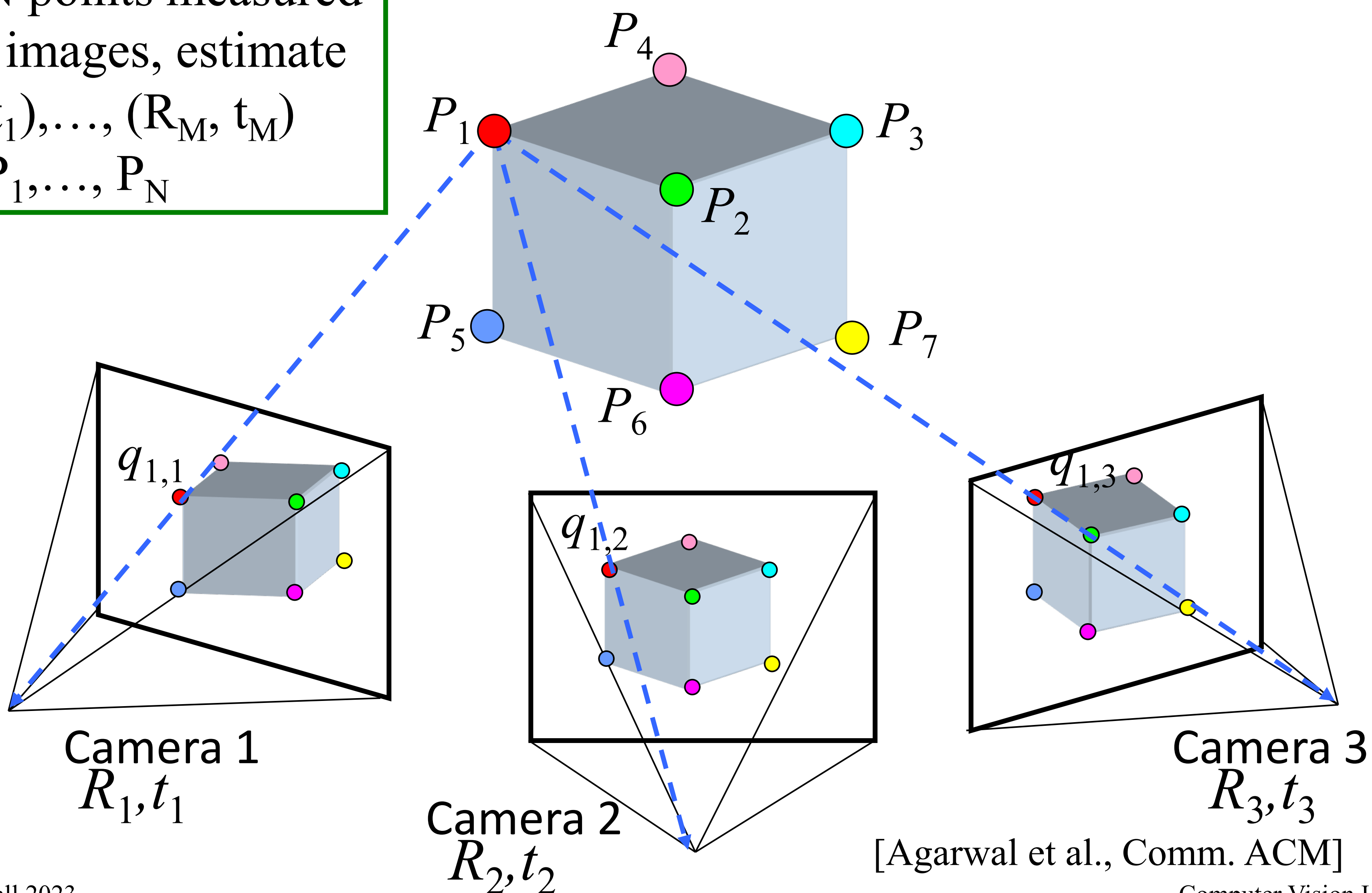
1. Discrete motion (wide baseline)
2. Continuous (Infinitesimal) motion usually from video (covered next week)

Structure from Motion



Structure from motion

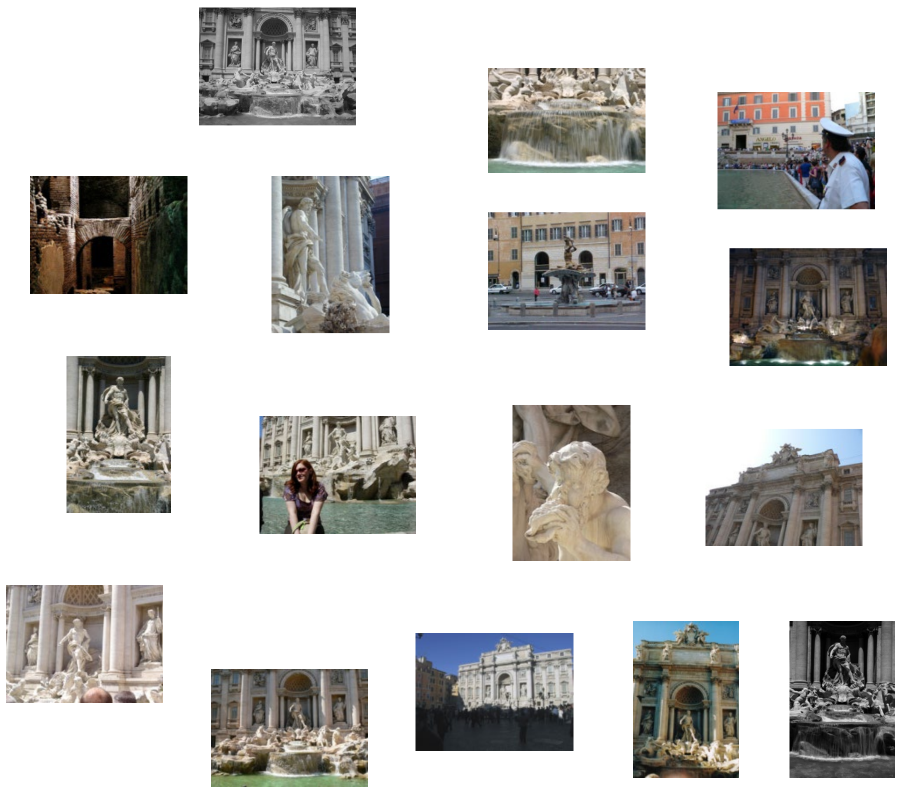
For N points measured in M images, estimate $(R_1, t_1), \ldots, (R_M, t_M)$ and $P_1, \ldots, P_N$



[Agarwal et al., Comm. ACM]

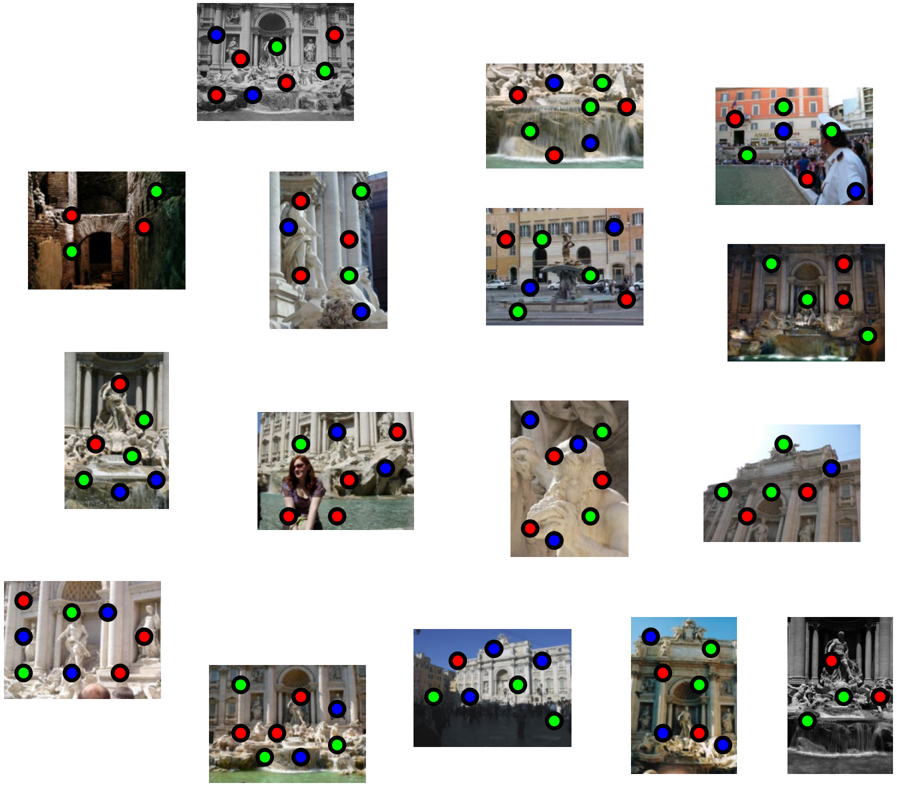
Feature detection

Several images observe a scene from different viewpoints



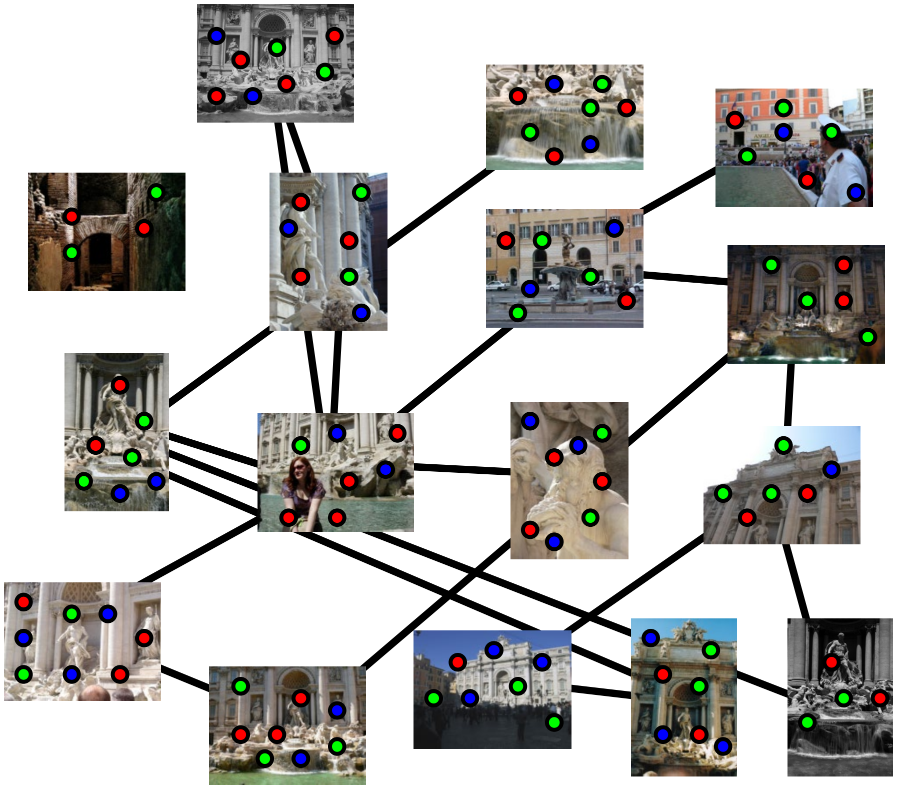
Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004]



Feature matching

Match features between each pair of images

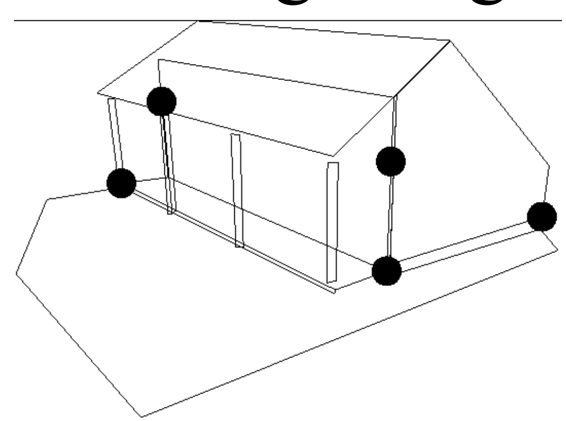


Two views, feature matching

- Greedy Algorithm:
  - Given feature in one image, find best match in second image irrespective of other matches
  - Suitable for small motions, little rotation, small search window
- Otherwise
  - Must compare descriptor over rotation
  - Cannot consider all potential pairings (way too many), so
    - Manual correspondence (e.g., photogrammetry)
    - Use robust outlier rejection (e.g., RANSAC)
    - More descriptive features (line segments, SIFT, larger regions, color)
    - Use video sequence to track, but perform SFM w/ first and last image

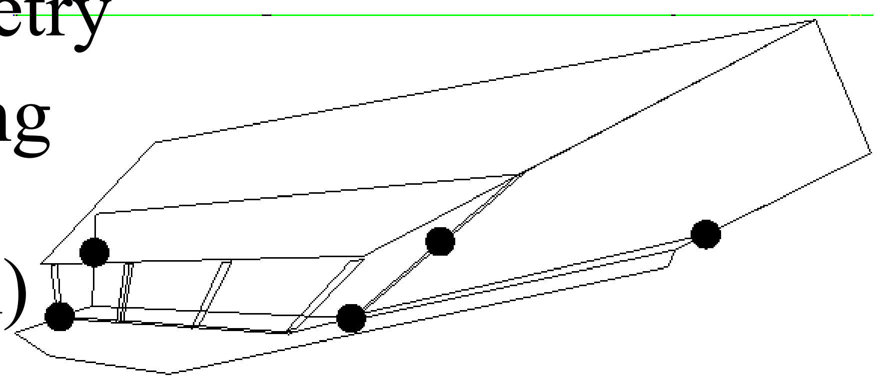
Two views, calibrated cameras

- Input: Two images (or video frames)
- Detect feature points
- Determine feature correspondences
- Compute the essential matrix
- Retrieve the relative camera rotation and translation (to scale) from the essential matrix
- Optional: Perform dense stereo matching using recovered epipolar geometry
- Reconstruct corresponding 3D scene points (to scale)



Two views, uncalibrated cameras

- Input: Two images (or video frames)
- Detect feature points
- Determine feature correspondences
- Compute the fundamental matrix
- Retrieve the relative camera 3D projective transformation from the fundamental matrix
- Optional: Perform dense stereo matching using recovered epipolar geometry
- Reconstruct corresponding 3D scene points (to 3D projective transformation)



Two-View Geometry

Essential Matrix E

- Calibrated
- Normalized coordinates
- Rank 2
  - Two nonzero singular values are equal
- 5 degrees of freedom
  - Camera rotation
  - Direction of camera translation
- Similarity reconstruction

Fundamental Matrix F

- Uncalibrated
- Pixel coordinates
- Rank 2
- 7 degrees of freedom
  - Homogeneous matrix to scale
  - $\det F = 0$
- Projective reconstruction

Essential Matrix (calibrated cameras)

- Number of point correspondences and solutions
  - 5 point correspondences, up to 10 (real) solutions
  - 6 point correspondences, 1 solution
  - 7 point correspondences, 1 or 3 real solutions (and 2 or 0 complex ones)
  - 8 or more point correspondences, 1 solution

Fundamental Matrix (uncalibrated cameras)

- Number of point correspondences and solutions
  - 7 point correspondences, 1 or 3 real solutions (and 2 or 0 complex ones)
  - 8 or more point correspondences, 1 solution

Mathematics

- Essential matrix
  - Linear estimation (8 or more correspondences)
  - Retrieval of normalized camera projection matrices and 3D rotation and translation (to scale)
- Fundamental matrix
  - Linear estimation (8 or more correspondences)
  - Retrieval of camera projection matrices and 3D projective transformation

Essential matrix, linear estimation

$$\hat{\mathbf{x}}_i'^\top \mathtt{E} \hat{\mathbf{x}}_i = 0 \, \forall \, i$$

$$\begin{bmatrix} \hat{x}_i' & \hat{y}_i' & \hat{w}_i' \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{w}_i \end{bmatrix} = 0$$

$$\hat{x}_i \hat{x}_i' e_{11} + \hat{y}_i \hat{x}_i' e_{12} + \hat{w}_i \hat{x}_i' e_{13} + \hat{x}_i \hat{y}_i' e_{21} + \hat{y}_i \hat{y}_i' e_{22} + \hat{w}_i \hat{y}_i' e_{23} + \hat{x}_i \hat{w}_i' e_{31} + \hat{y}_i \hat{w}_i' e_{32} + \hat{w}_i \hat{w}_i' e_{33} = 0$$

$$\begin{bmatrix} \hat{x}_i \hat{x}_i' & \hat{y}_i \hat{x}_i' & \hat{w}_i \hat{x}_i' & \hat{x}_i \hat{y}_i' & \hat{y}_i \hat{y}_i' & \hat{w}_i \hat{y}_i' & \hat{x}_i \hat{w}_i' & \hat{y}_i \hat{w}_i' & \hat{w}_i \hat{w}_i' \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{bmatrix} = 0$$

$$\mathbf{a}_i^\top \mathbf{e} = 0$$

where

$$\mathbf{a}_i = (\hat{x}_i \hat{x}_i', \hat{y}_i \hat{x}_i', \hat{w}_i \hat{x}_i', \hat{x}_i \hat{y}_i', \hat{y}_i \hat{y}_i', \hat{w}_i \hat{y}_i', \hat{x}_i \hat{w}_i', \hat{y}_i \hat{w}_i', \hat{w}_i \hat{w}_i')^\top$$

$$\mathbf{e} = (e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33})^\top$$

Essential matrix, linear estimation

Given $n \geq 8$ point correspondences

$$\begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} \mathbf{e} = 0$$

$$\mathtt{A}\mathbf{e} = 0, \text{ solve for } \mathbf{e}$$

where

$$\mathbf{e} = (e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33})^\top$$

$$\mathtt{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix}$$

$$\mathbf{a}_i = (\hat{x}_i\hat{x}'_i, \hat{y}_i\hat{x}'_i, \hat{w}_i\hat{x}'_i, \hat{x}_i\hat{y}'_i, \hat{y}_i\hat{y}'_i, \hat{w}_i\hat{y}'_i, \hat{x}_i\hat{w}'_i, \hat{y}_i\hat{w}'_i, \hat{w}_i\hat{w}'_i)^\top$$

Essential matrix, linear estimation

$\mathtt{A}\mathbf{e} = 0$, solve for $\mathbf{e}$ using singular value decomposition (SVD)

$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top$$

where

$$\mathtt{U} \text{ and } \mathtt{V} \text{ are orthogonal matrices}$$
$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_9), \text{ where } \sigma_i \geq \sigma_{i+1} \geq 0$$

Columns of $\mathtt{U}$ are left singular vectors corresponding to singular values $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_9)^\top$ and columns of $\mathtt{V}$ are right singular vectors corresponding to singular values.
$\mathbf{e} = (e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33})^\top$ is right singular vector corresponding to smallest singular value (i.e., $\mathbf{e}$ is the last column of $\mathtt{V}$)

$$\mathtt{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

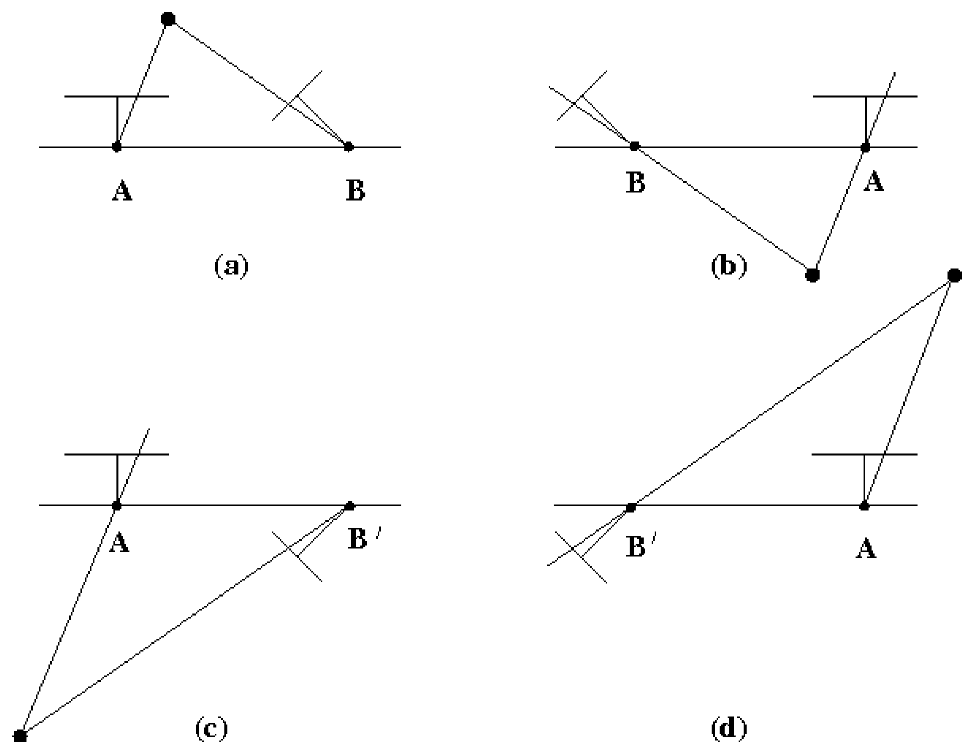
then enforce constraints (rank 2 and other two singular values are equal)

$$\mathtt{E} = \mathtt{U}\Sigma\mathtt{V}^\top$$
$$\mathtt{E} = \mathtt{U}\Sigma'\mathtt{V}^\top, \text{ where } \Sigma' = \text{diag}(1, 1, 0)$$

Two views, calibrated cameras

- Retrieve relative camera rotation and direction of translation from essential matrix $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ $\quad \hat{\mathtt{P}} = [\mathtt{I} \mid \mathbf{0}]$ and $\hat{\mathtt{P}}' = [\mathtt{R} \mid \mathbf{t}]$

Four solutions for **R** and **t**, but only one (a) where reconstructed point is in front of both cameras



Fundamental matrix, linear estimation

Warning: data normalization must be used!

$$\mathbf{x}_i'^\top \mathbf{F} \mathbf{x}_i = 0 \, \forall \, i$$

$$\begin{bmatrix} x_i' & y_i' & w_i' \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = 0$$

$$x_i x_i' f_{11} + y_i x_i' f_{12} + w_i x_i' f_{13} + x_i y_i' f_{21} + y_i y_i' f_{22} + w_i y_i' f_{23} + x_i w_i' f_{31} + y_i w_i' f_{32} + w_i w_i' f_{33} = 0$$

$$\begin{bmatrix} x_i x_i' & y_i x_i' & w_i x_i' & x_i y_i' & y_i y_i' & w_i y_i' & x_i w_i' & y_i w_i' & w_i w_i' \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$$\mathbf{a}_i^\top \mathbf{f} = 0$$

where

$$\mathbf{a}_i = (x_i x_i', y_i x_i', w_i x_i', x_i y_i', y_i y_i', w_i y_i', x_i w_i', y_i w_i', w_i w_i')^\top$$

$$\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^\top$$

Fundamental matrix, linear estimation

Given $n \geq 8$ point correspondences

Warning: data normalization must be used!

$$\begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} \mathbf{f} = 0$$

$$\mathtt{A}\mathbf{f} = 0, \text{ solve for } \mathbf{f}$$

where

$$\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^\top$$

$$\mathtt{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix}$$

$$\mathbf{a}_i = (x_i x_i', y_i x_i', w_i x_i', x_i y_i', y_i y_i', w_i y_i', x_i w_i', y_i w_i', w_i w_i')^\top$$

Fundamental matrix, linear estimation

$\mathtt{A}\mathbf{f} = 0$, solve for $\mathbf{f}$ using singular value decomposition (SVD)

$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top$$

where

$$\mathtt{U} \text{ and } \mathtt{V} \text{ are orthogonal matrices}$$
$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_9), \text{ where } \sigma_i \geq \sigma_{i+1} \geq 0$$

Columns of $\mathtt{U}$ are left singular vectors corresponding to singular values $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_9)^\top$ and columns of $\mathtt{V}$ are right singular vectors corresponding to singular values.
$\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^\top$ is right singular vector corresponding to smallest singular value (i.e., $\mathbf{f}$ is the last column of $\mathtt{V}$)

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

Warning: data normalization must be used!

then enforce constraint (rank 2)

$$\mathtt{F} = \mathtt{U}\Sigma\mathtt{V}^\top, \text{ where } \Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$$
$$\mathtt{F} = \mathtt{U}\Sigma'\mathtt{V}^\top, \text{ where } \Sigma' = \text{diag}(\sigma_1, \sigma_2, 0)$$

Two views, uncalibrated cameras

- Retrieve uncalibrated camera projection matrices from fundamental matrix

$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [[\mathbf{e}']_\times \mathtt{F} + \mathbf{e}'\mathbf{v}^\top \mid \lambda \mathbf{e}']$, where $\mathbf{v}$ is any 3-vector and λ is a non-zero scalar. If $\mathbf{v} = \mathbf{0}$ and $\lambda = 1$, then $\mathtt{P}' = [[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{e}']$.

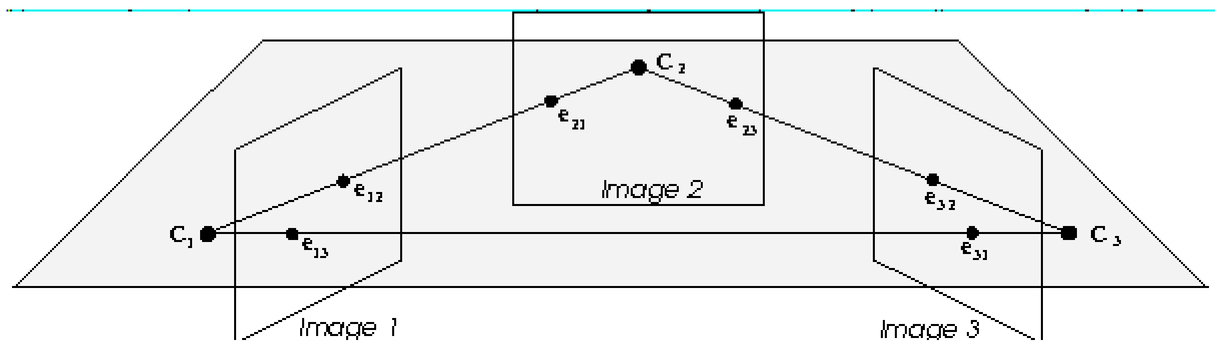
- Retrieve epipoles from fundamental matrix

$$\mathtt{F}\mathbf{e} = \mathbf{0} \quad \mathbf{e} \text{ is (right) null space of } \mathtt{F}$$

$$\mathbf{e}'^\top \mathtt{F} = \mathbf{0}^\top \quad \mathbf{e}' \text{ is left null space of } \mathtt{F}$$

$$(\mathtt{F}^\top \mathbf{e}' = \mathbf{0} \quad \mathbf{e}' \text{ is (right) null space of } \mathtt{F}^\top)$$

Three Views



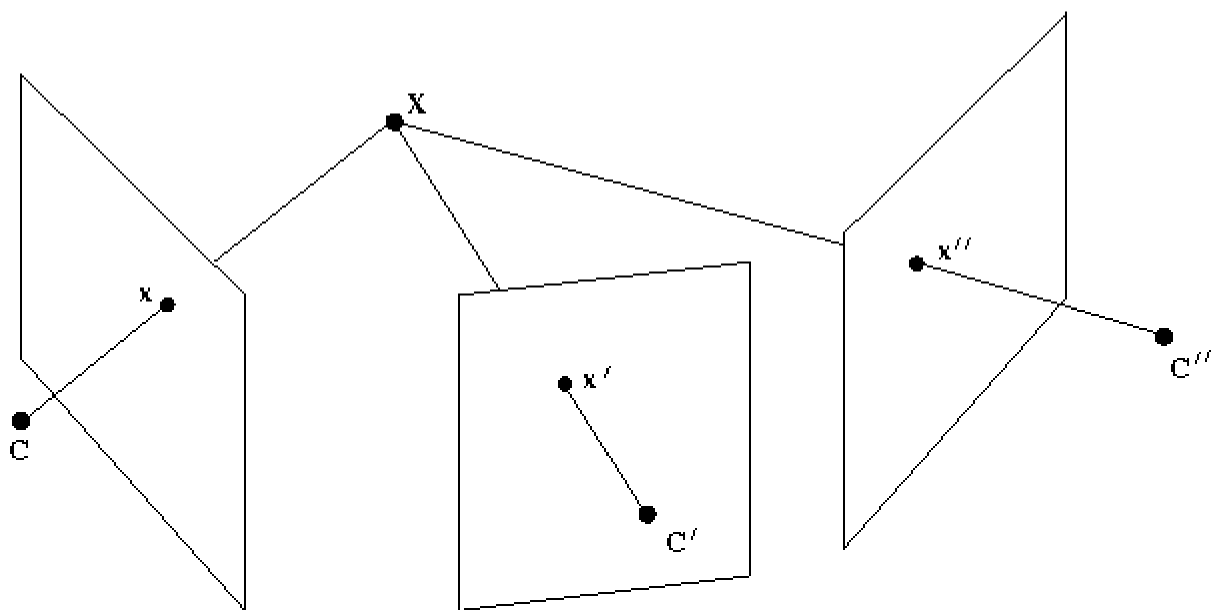
Trifocal plane

Trifocal Tensor

- 3x3x3 tensor
- 27 elements, 18 degrees of freedom
  – 33 degrees of freedom (3 camera projection matrices) minus 15 degrees of freedom (3D projective transformation)
- Uses tensor notation
  – Einstein summation
- Retrieve camera projection matrices

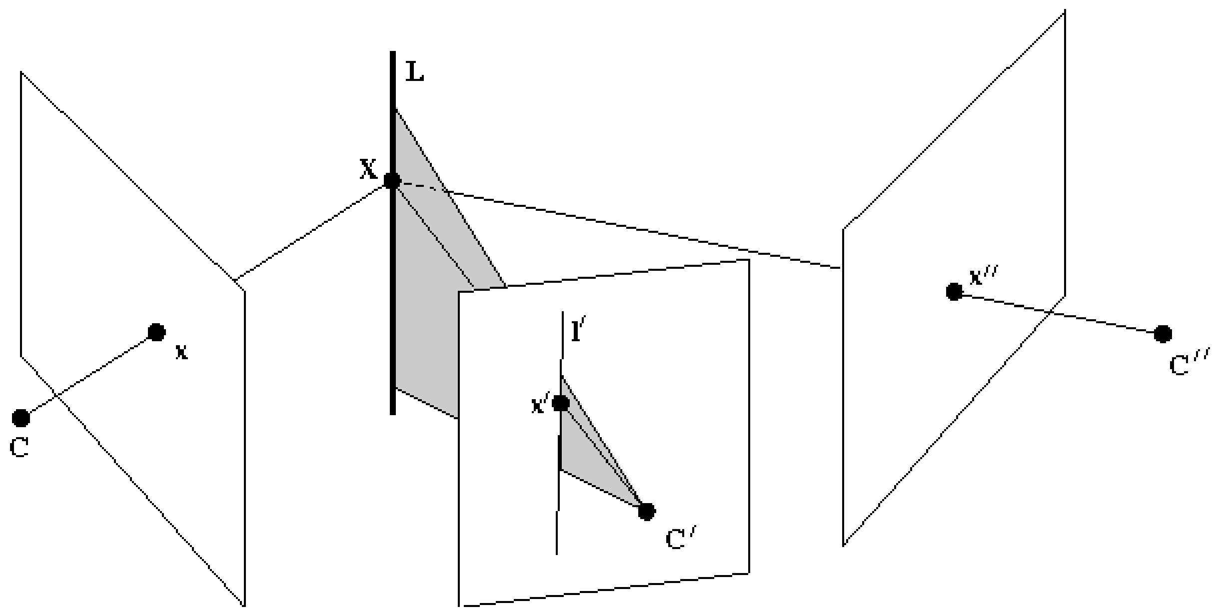
3 Points

- Point-Point-Point



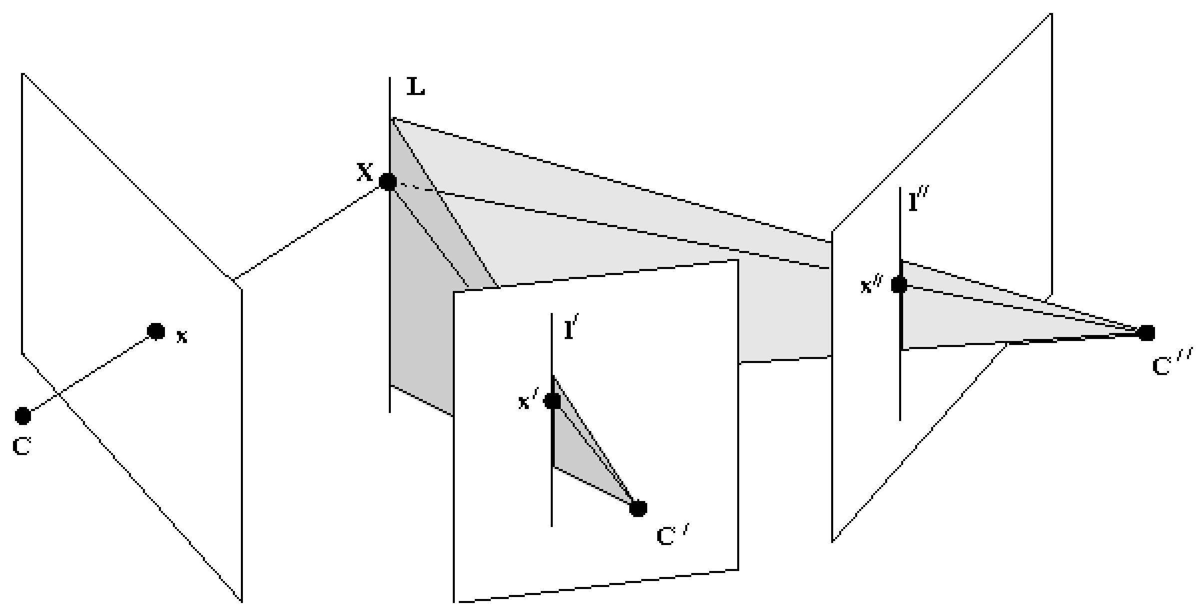
2 Points, 1 Line

- Point-Line-Point
  - Note: image line must pass through corresponding image point



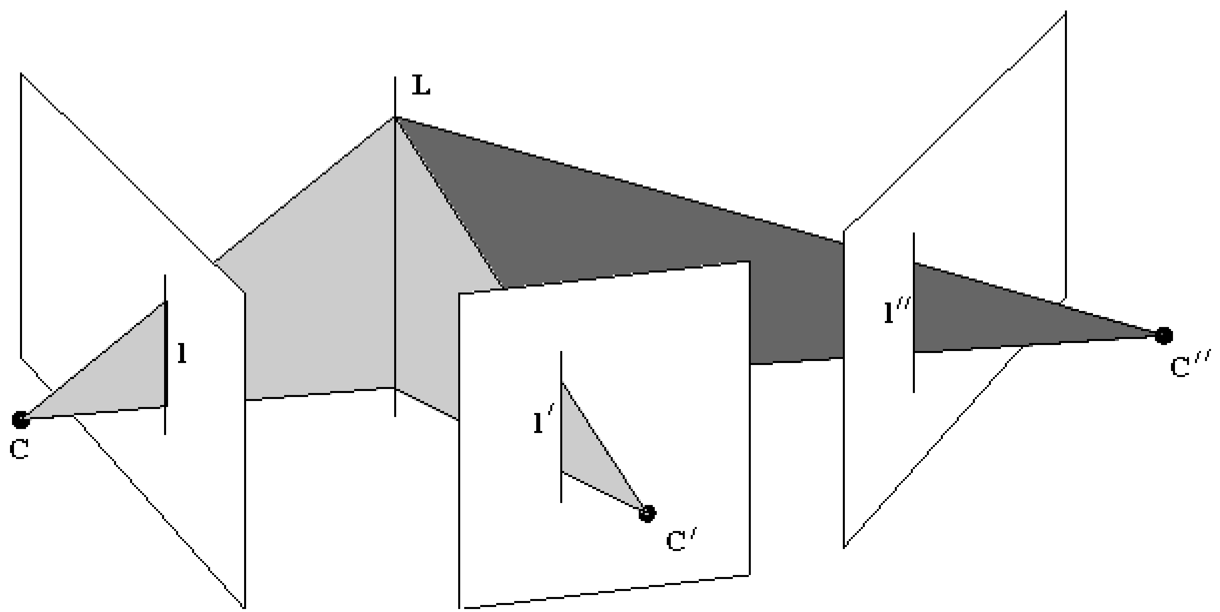
1 Point, 2 Lines

- Point-Line-Line
  – Note: image lines do not need to correspond, but must pass through corresponding image points



3 Lines

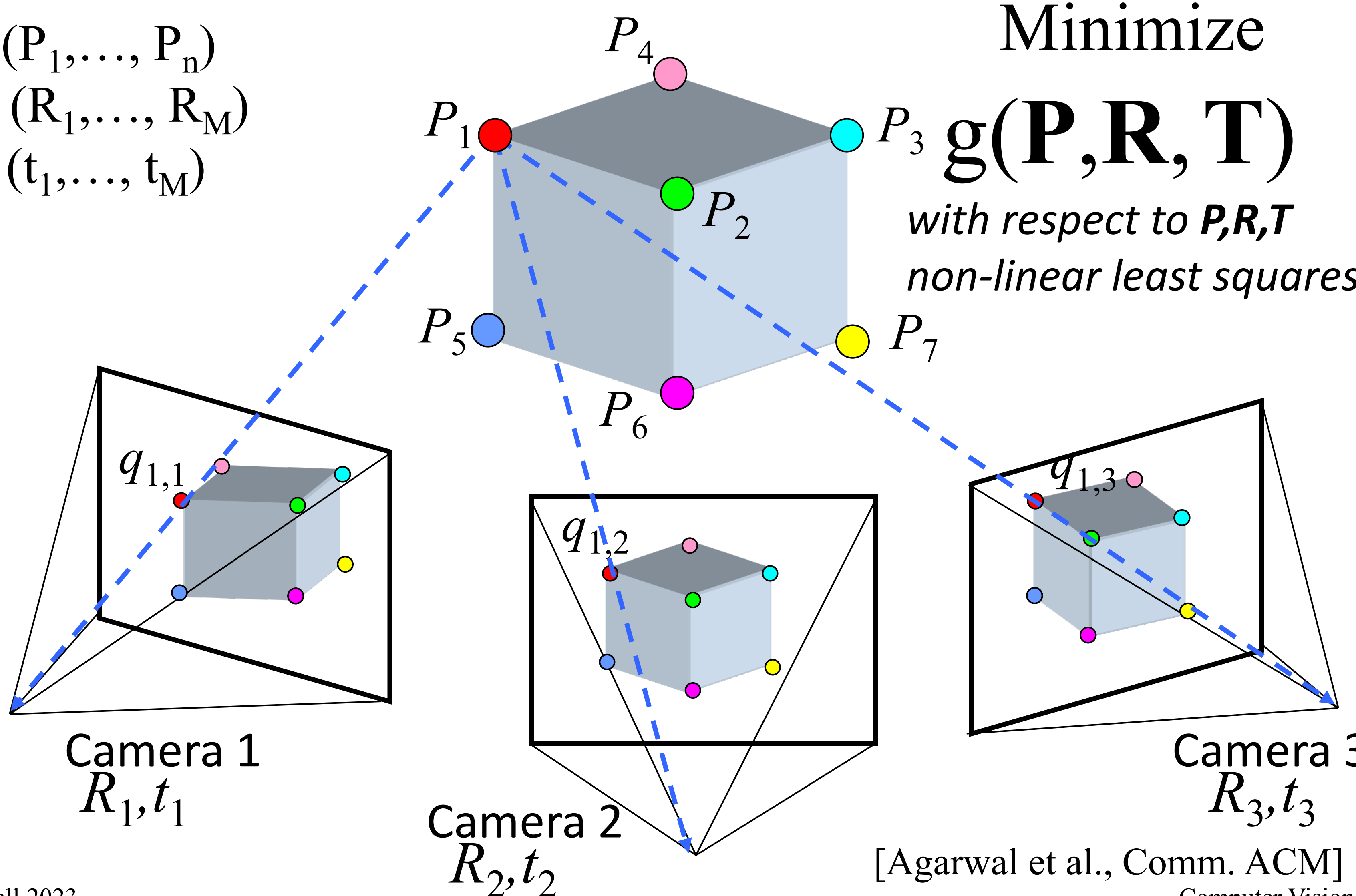
- Line-Line-Line



N-view structure from motion

$\mathbf{P} = (P_1, \ldots, P_n)$
$\mathbf{R} = (R_1, \ldots, R_M)$
$\mathbf{T} = (t_1, \ldots, t_M)$

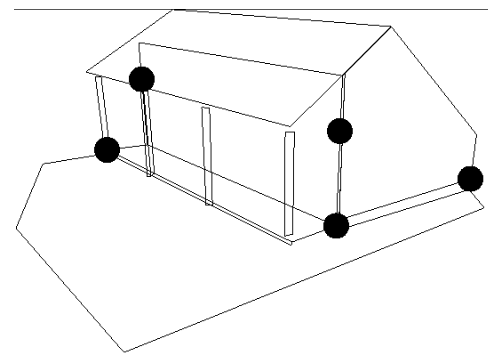
Minimize
$$g(\mathbf{P}, \mathbf{R}, \mathbf{T})$$
with respect to ***P,R,T***
non-linear least squares



[Agarwal et al., Comm. ACM]

Calibrated structure from motion

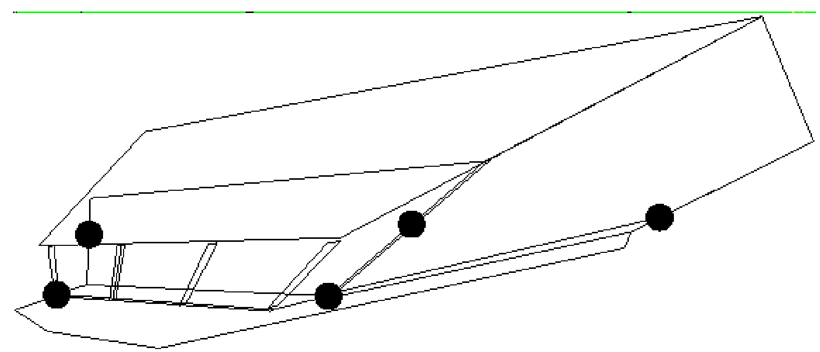
- Consider m images of n points, how many unknowns?
  - Unknowns
    - 3D Structure: $3n$
    - Cameras: $6m - 7$
    - Total: $3n + 6m - 7$
  - Measurements
    - $2nm$
- Solution when $3n + 6m - 7 \leq 2nm$



Up to 3D similarity transformation

Uncalibrated structure from motion

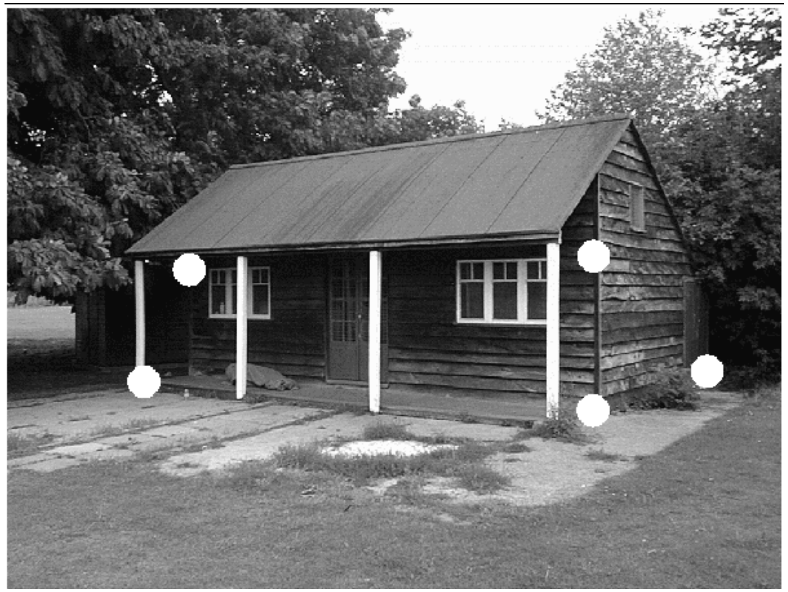
- Consider m images of n points, how many unknowns?
  - Unknowns
    - 3D Structure: $3n$
    - Cameras: $11m - 15$
    - Total: $3n + 11m - 15$
  - Measurements
    - $2nm$
- Solution when $3n + 11m - 15 \leq 2nm$

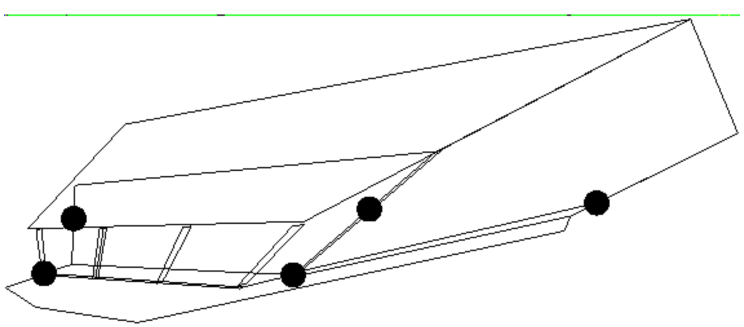


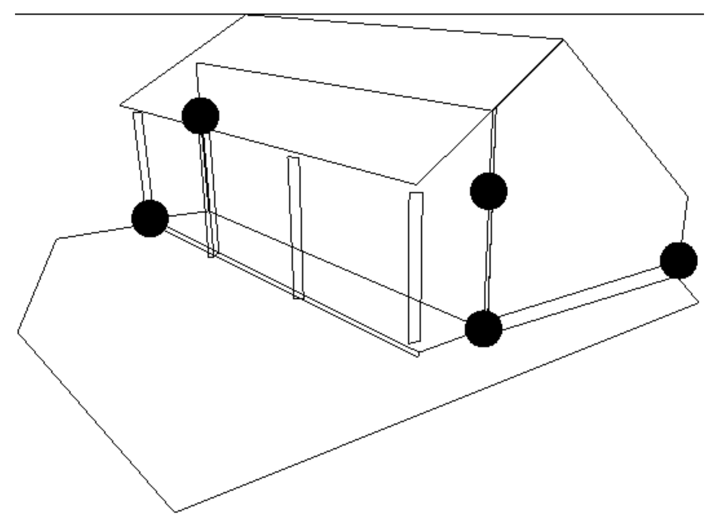
Up to 3D projective transformation

Direct Reconstruction
Projective to Euclidean

5 (or more) points correspondences







N-View Geometry

- Reconstruction
  - Bundle adjustment
    - Simultaneous adjustment of parameters for all cameras and all 3D scene points
    - Minimize reprojection error in all images

$$\min_{\hat{\mathrm{P}}^i, \widehat{\mathbf{X}}_j} \sum_{ij} d(\hat{\mathrm{P}}^i \widehat{\mathbf{X}}_j, \mathbf{x}_j^i)^2$$

    - Reconstruction of cameras and 3D scene points to similarity (calibrated) or projective (uncalibrated) ambiguity
  - Factorization (see textbook)

Bundle adjustment

- Minimize sum of squared reprojection errors:

$$g(\mathbf{P},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij} \left\| P(P_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

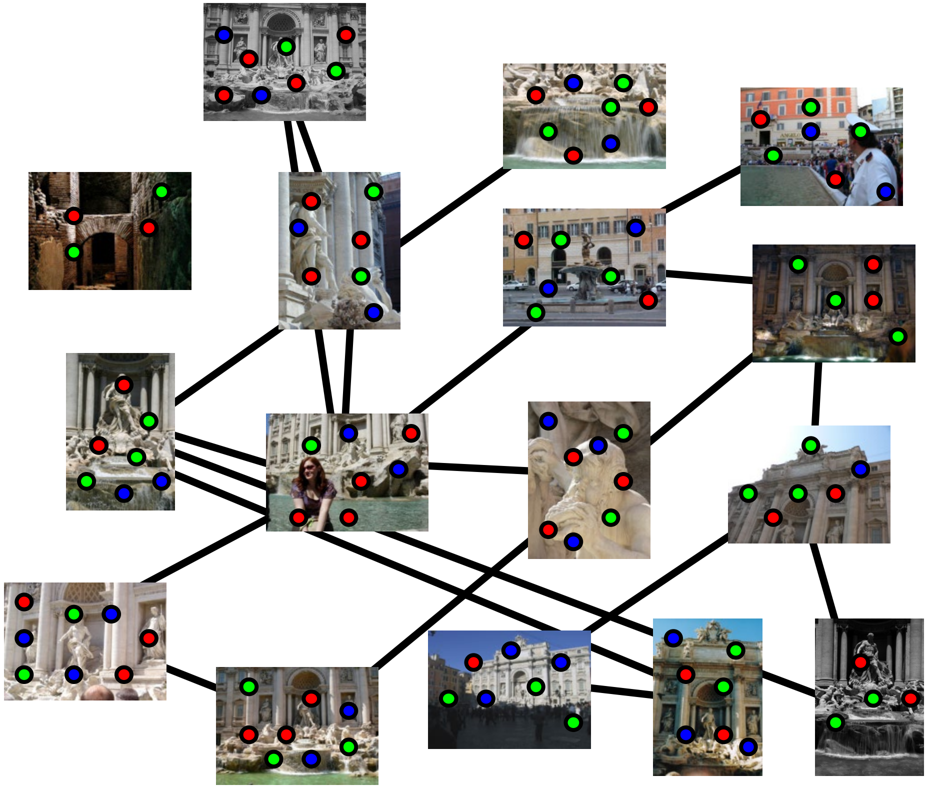
predicted image location

observed image location

indicator variable:
1: If point *i is* visible in image *j*
0: Otherwise

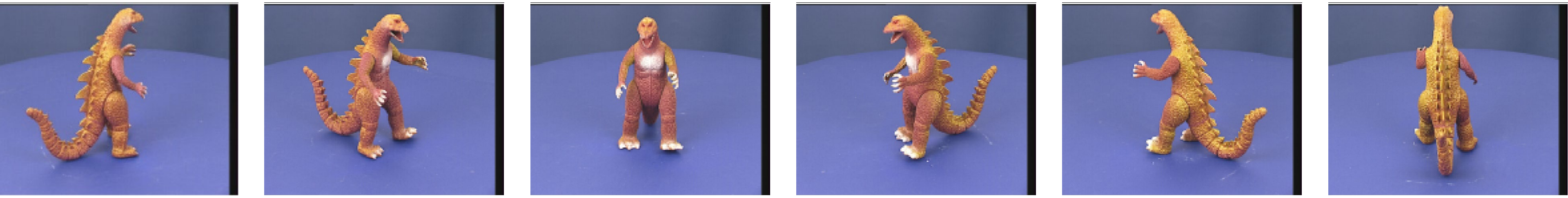
  - g(**P, R, T) is** optimized with non-linear least squares
  - Levenberg-Marquardt is a popular choice
- Practical challenges
  - Initialization (Bootstrap from 2 View Solution between pairs)
  - Outliers (covered in next lecture)

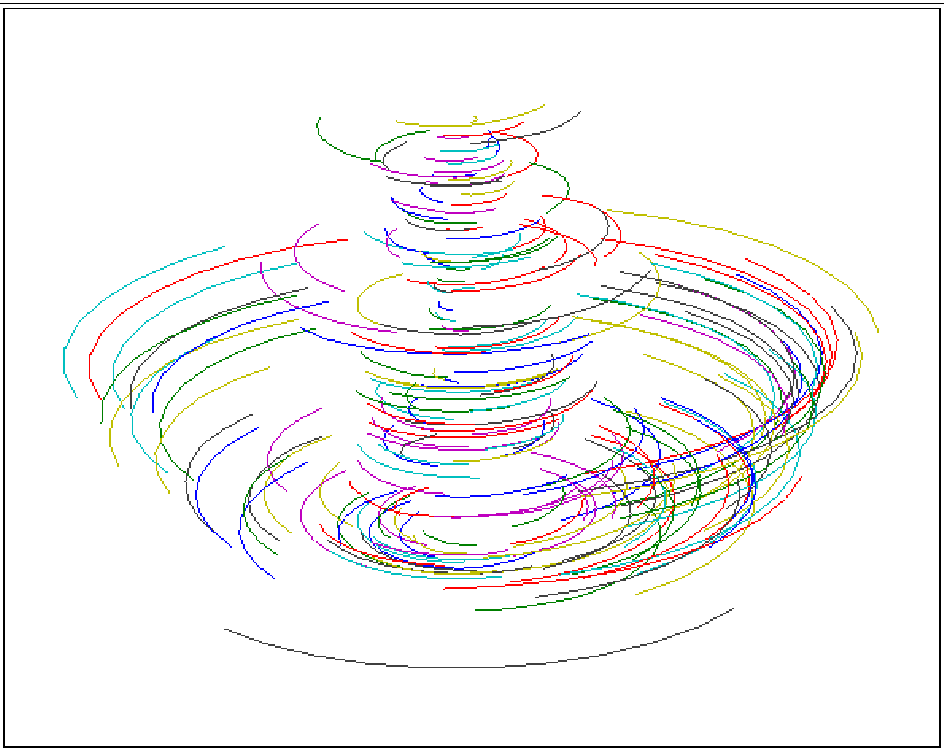
Initial Conditions for Optimization

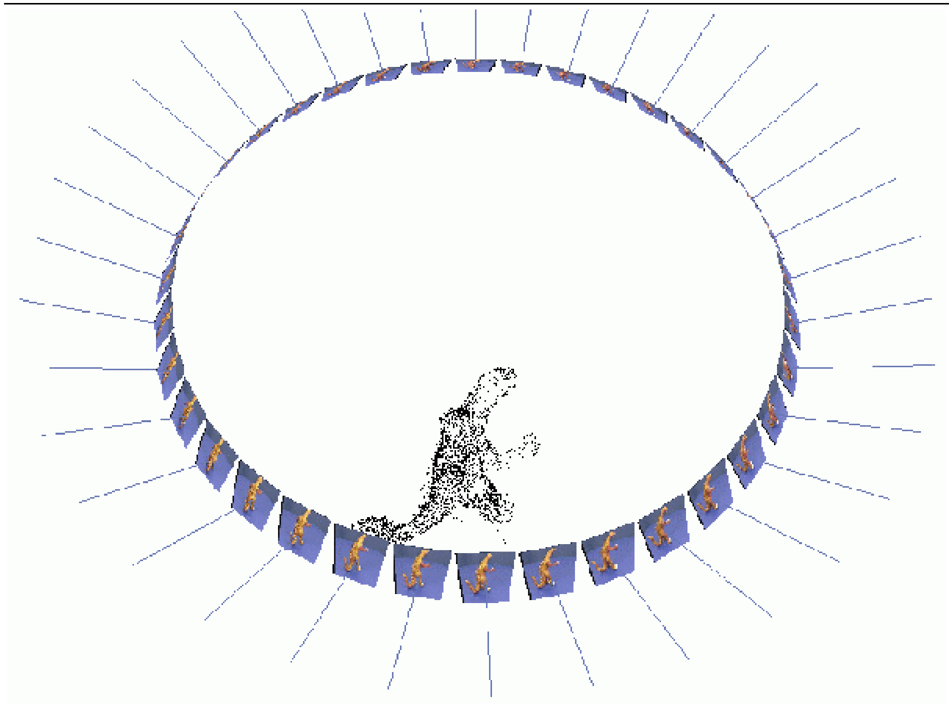


- Perform 2 View SFM for image pairs that have at least 8 matching pairs of features
- This forms a graph
- Can propagate translation and orientation across a spanning tree of the graph
- Can be more accurate if you use full graph

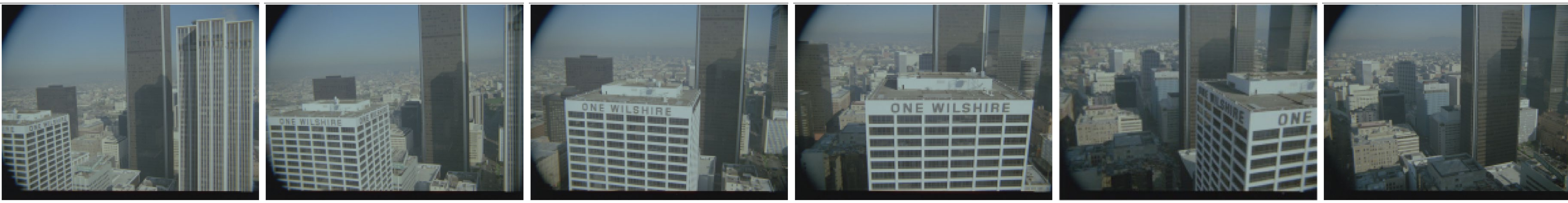
N-View Geometry

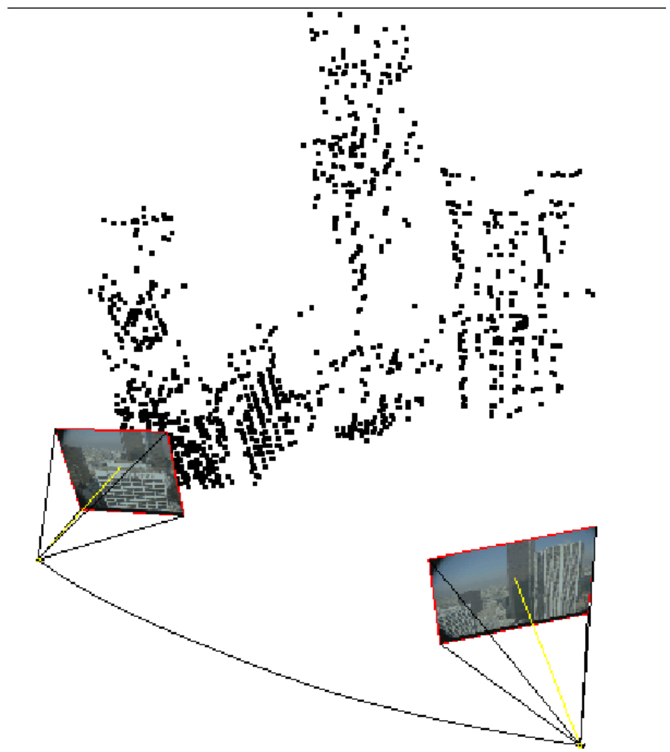






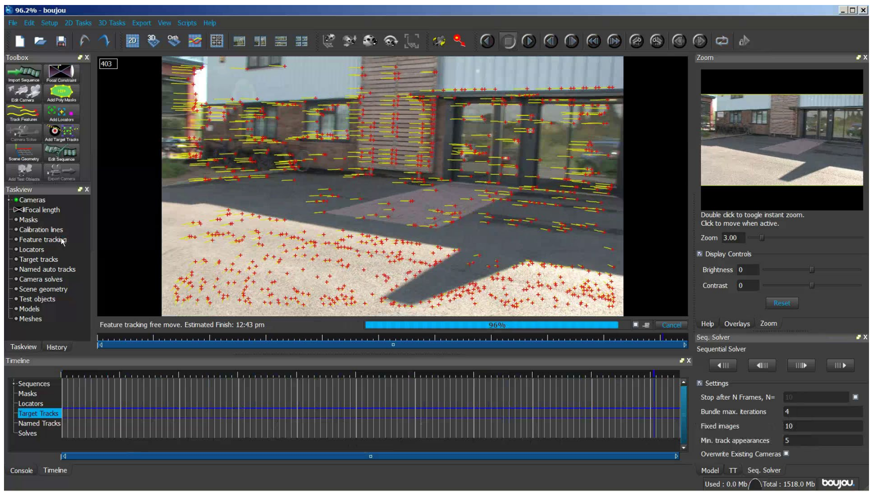
N-View Geometry

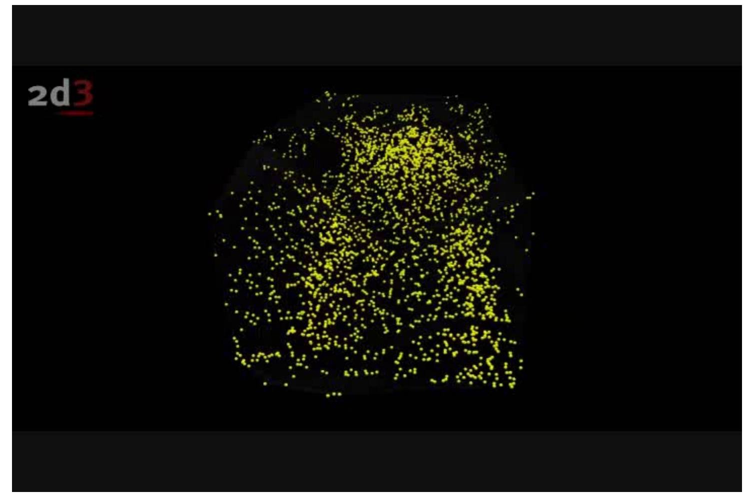




N-View Geometry

- Example results





Next Lecture

- Model fitting