Lecture
Reconstruct 3D Geometry

Liangliang Nan

Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Extracting corresponding image points (next lecture)
    - Recover camera motion
    - Triangulation
  - 3D from more views
    - Structure from motion

Review of Epipolar Geometry

- Essential matrix
  - Canonical camera assumption

$$p'^{T} E\, p = 0, \quad E = [\mathbf{t}_{\times}]R \qquad K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Fundamental matrix (most important concept in 3DV)

$$p'^{T} F\, p = 0, \quad F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

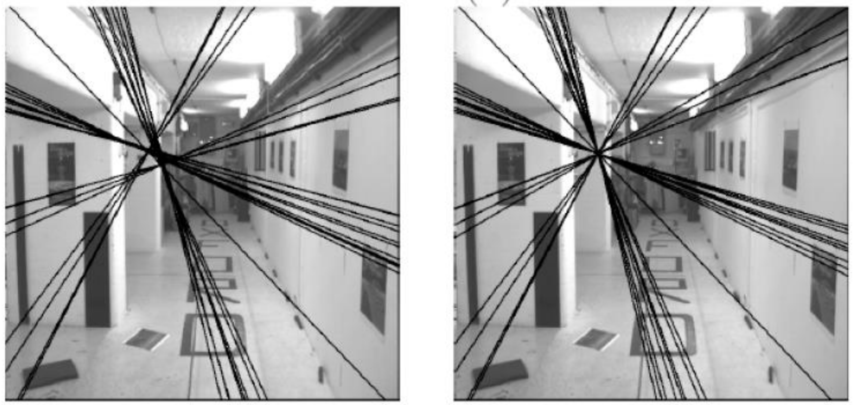
  - Relates matching image points of different views
    - No known 3D location
    - No known camera intrinsic and extrinsic parameters

Review of Epipolar Geometry

- Fundamental matrix
  - 3 by 3
  - homogeneous (has scale ambiguity)
  - rank(F) = 2
    - The potential matching point is located on a line
  - 7 degrees of freedom

$$\mathbf{p}'^{T}F\mathbf{p} = 0 \qquad F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.



Left : Uncorrected F – epipolar lines are not coincident.

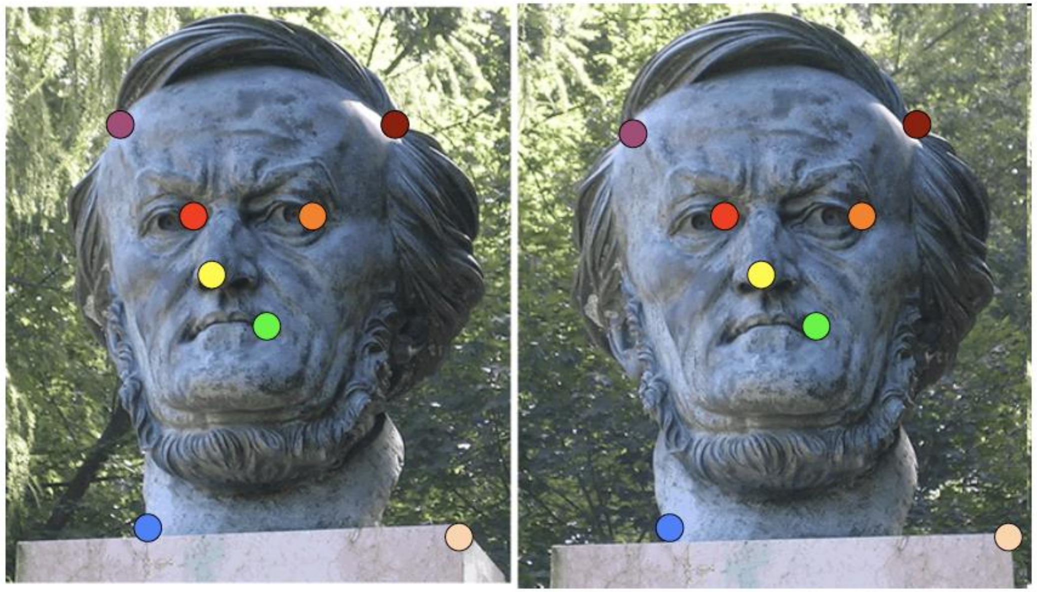
Right : Epipolar lines from corrected F.

Review of Epipolar Geometry

- Recover F from corresponding image points
  - 8 unknown parameters to recover (scale ambiguity)
  - Each point pair gives a single linear constraint

$$\begin{cases} \mathbf{p}_i = (u_i, v_i, 1) \\ \mathbf{p}'_i = (u'_i, v'_i, 1) \end{cases} \qquad \mathbf{p}'^T F \, \mathbf{p} = 0$$

$$\begin{bmatrix} u_i u'_i & v_i u'_i & u'_i & u_i v'_i & v_i v'_i & v'_i & u_i & v_i & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$



Review of Epipolar Geometry

- Recover F from corresponding image points
  - 8 unknown parameters to recover (scale ambiguity)
  - Each point pair gives a single linear constraint
  - 7-point algorithm does exist but less popular
  - ~~8-point algorithm (>= 8 pairs)~~ → Normalized 8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ u_3u_3' & v_3u_3' & u_3' & u_3v_3' & v_3v_3' & v_3' & u_3 & v_3 & 1 \\ u_4u_4' & v_4u_4' & u_4' & u_4v_4' & v_4v_4' & v_4' & u_4 & v_4 & 1 \\ u_5u_5' & v_5u_5' & u_5' & u_5v_5' & v_5v_5' & v_5' & u_5 & v_5 & 1 \\ u_6u_6' & v_6u_6' & u_6' & u_6v_6' & v_6v_6' & v_6' & u_6 & v_6 & 1 \\ u_7u_7' & v_7u_7' & u_7' & u_7v_7' & v_7v_7' & v_7' & u_7 & v_7 & 1 \\ u_8u_8' & v_8u_8' & u_8' & u_8v_8' & v_8v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$W\mathbf{f} = 0$$

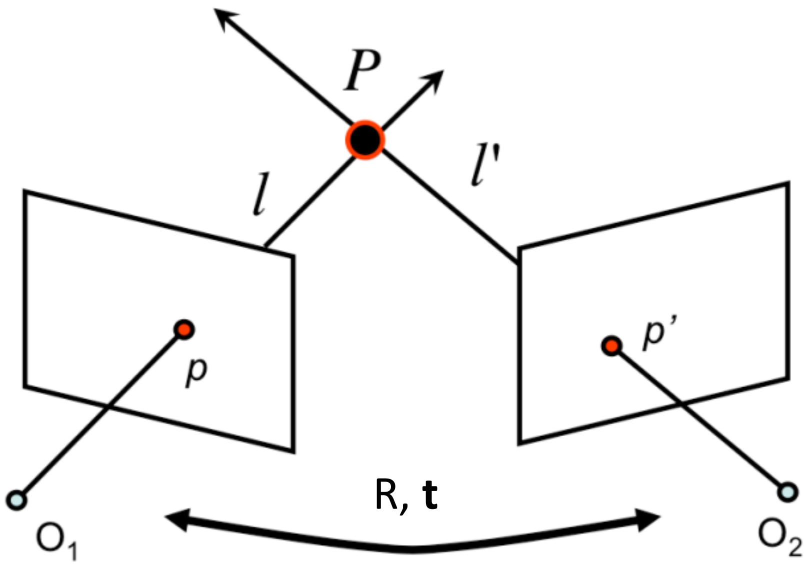
Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Extracting corresponding image points (next lecture)
    - Recover camera motion
    - Triangulation
  - 3D from more views
    - Structure from motion

3D from 2 Views

- The general idea

Recover 3D coordinates from corresponding image points
(assume camera parameters are known)

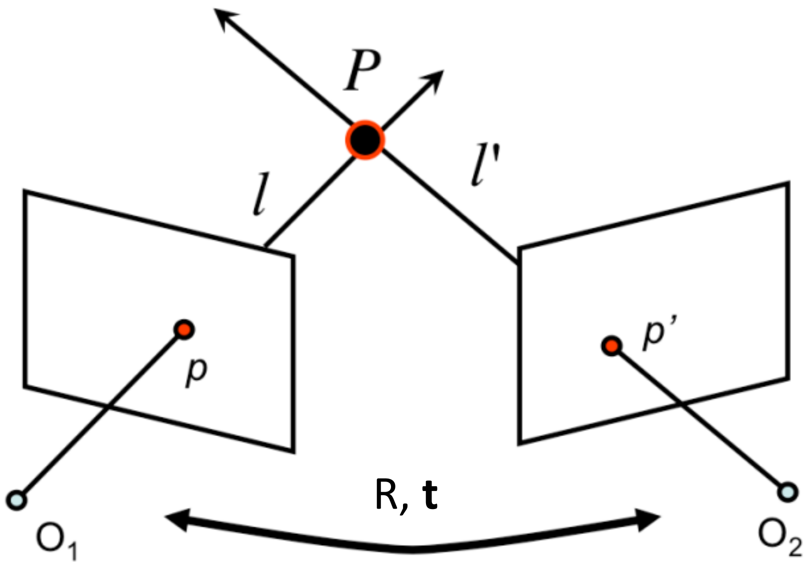


3D from 2 Views

- What information is needed?
  - Corresponding image points (next lecture) ✓
    - Image matching techniques
  - Intrinsic camera parameters ✓
    - Camera calibration
  - Extrinsic camera parameters?
    - Recover from image points

$$p'^T F\, p = 0,$$

$$F = K'^{-T}[\mathbf{t}_\times] R K^{-1}$$



Today’s Agenda

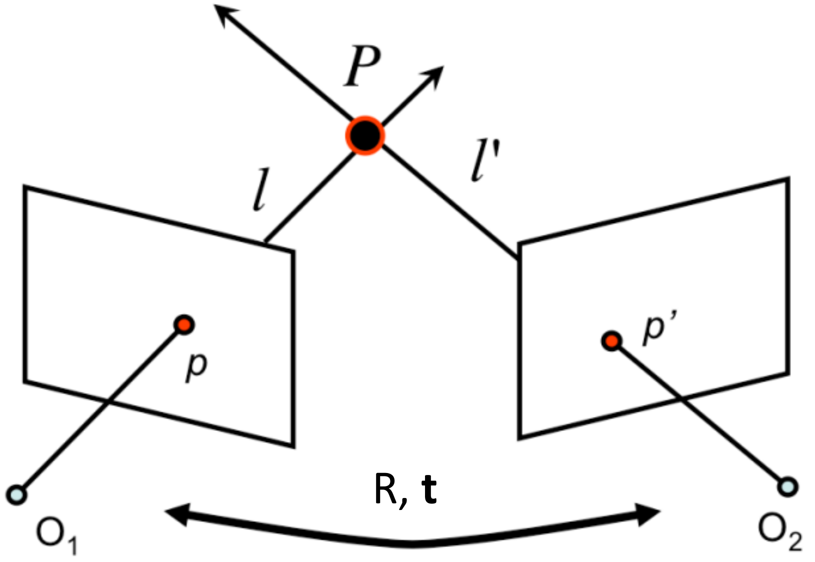
- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Extracting corresponding image points (next lecture)
    - Recover camera motion
    - Triangulation
  - 3D from more views
    - Structure from motion

Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
  - Known intrinsic parameters
    - From calibration

$$F = K'^{-T}[\mathbf{t}_\times]RK^{-1}$$

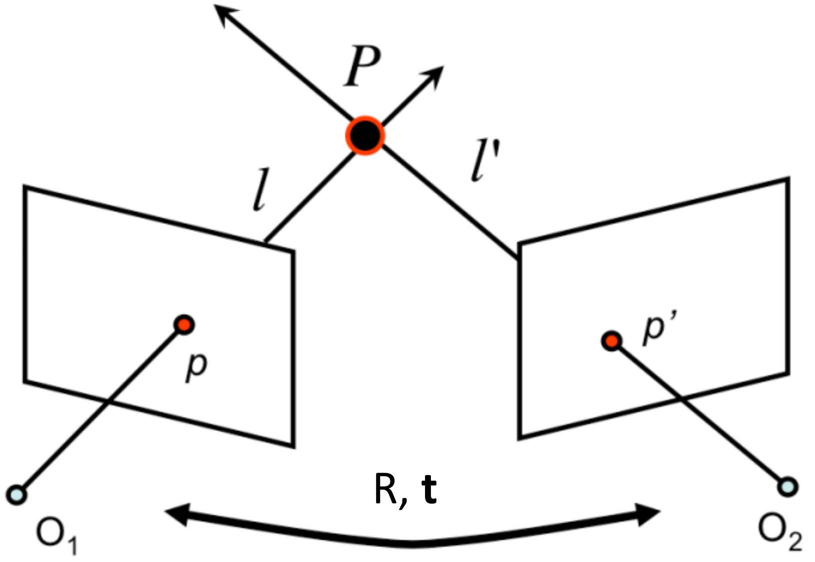
$$E = [\mathbf{t}_\times]R = K'^{T}FK$$



Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix

$$E = [\mathbf{t}_{\times}]R$$



Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - SVD of E $\quad E = UDV^{\mathrm{T}}$
  - determinant(R) > 0
    - Two potential values

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

$$R = (\det UWV^T)UWV^T \text{ or } (\det UW^TV^T)UW^TV^T$$

Relative Pose from Fundamental Matrix

TU Delft 3Dgeoinfo

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - SVD of E $\quad E = UDV^{\mathrm{T}}$
  - determinant(R) > 0
    - Two potential values

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

$$R = (\det UWV^T)UWV^T \text{ or } (\det UW^TV^T)UW^TV^T$$

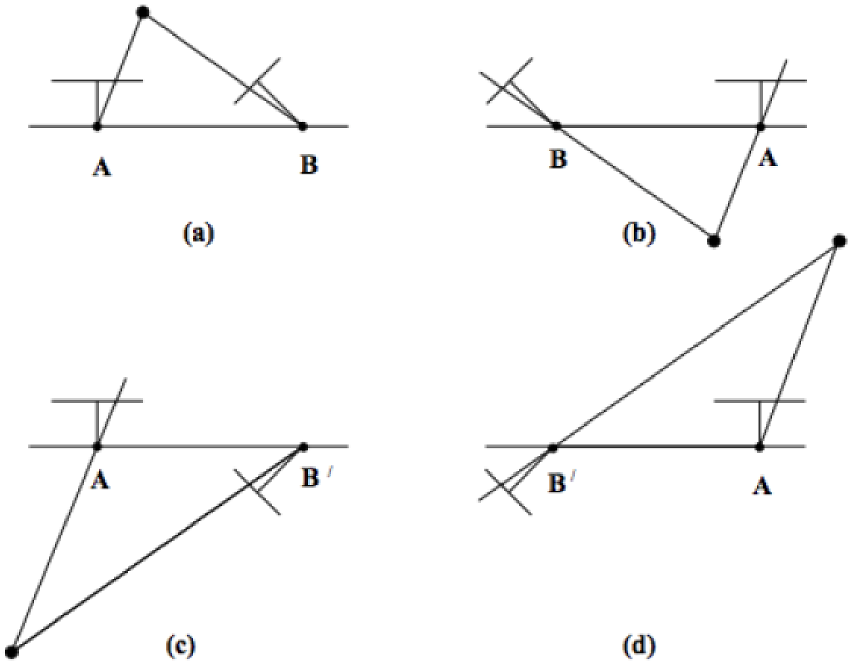
  - $\mathbf{t}$ up to a sign
    - Two potential values
    - $\mathbf{u}_3$: last column of U
    - Corresponds to the smallest singular value

$$\mathbf{t} = \pm U \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \pm \mathbf{u}_3$$

Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - $\mathbf{t}$: two potential values

Which is the correct configuration

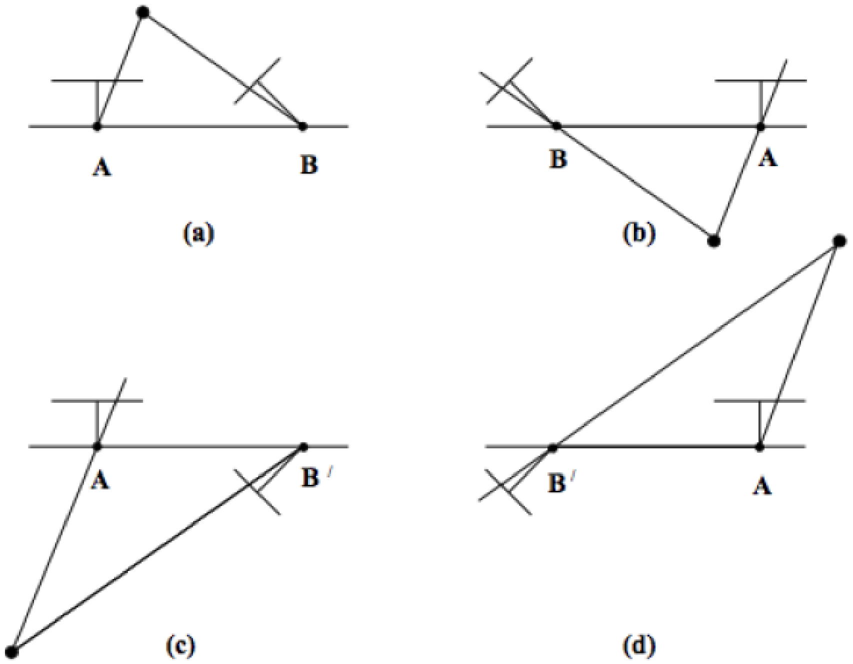


B and B’ rotate cameras in opposite directions

Relative Pose from Fundamental Matrix

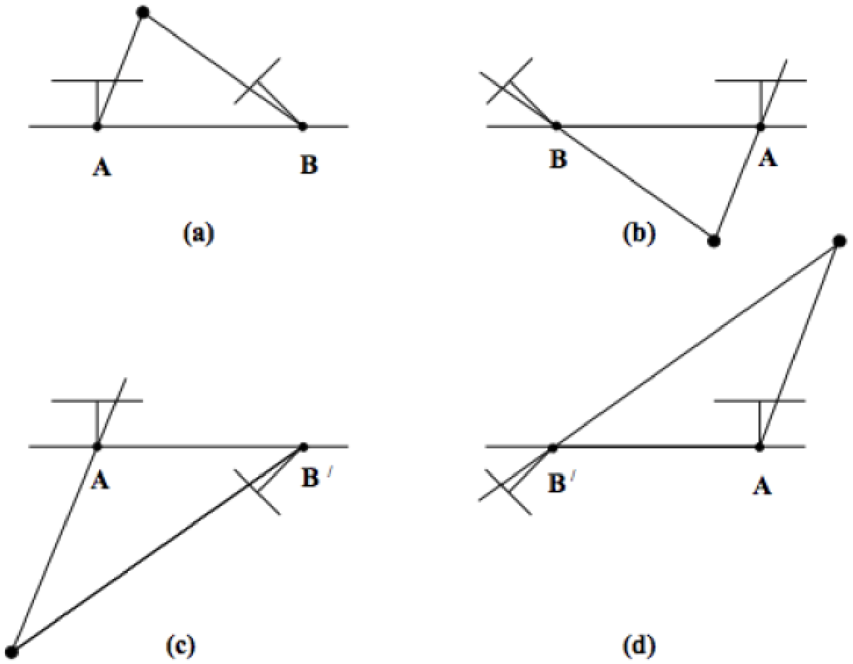
- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - $\mathbf{t}$: two potential values
  - 3D points must be in front of both cameras

But 3D points are not known yet?



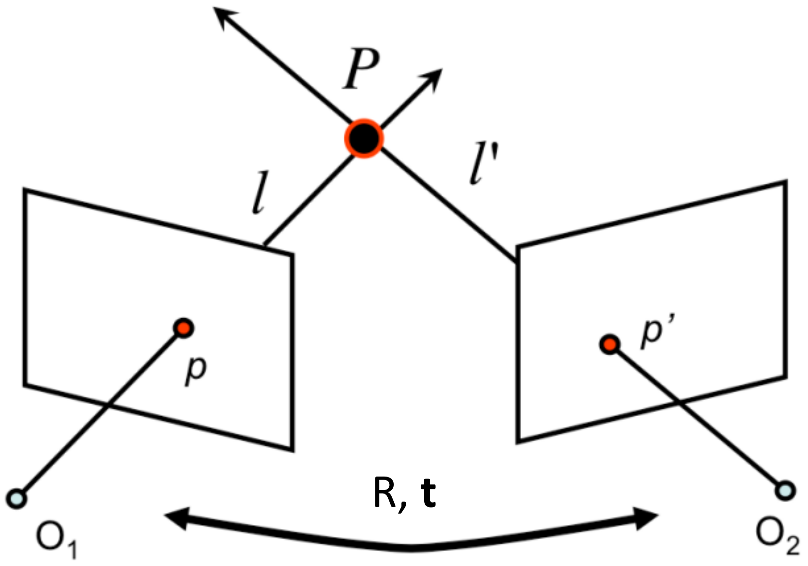
Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - $\mathbf{t}$: two potential values
  - 3D points must be in front of both cameras
    - Reconstruct 3D points
      - using all potential pairs of R and $\mathbf{t}$
    - Count the number of points in front of cameras
    - The pair giving max number is correct



Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - $\mathbf{t}$: two potential values
  - 3D points must be in front of both cameras
    - First camera
      - $P.z > 0$?
    - Second camera
      - P in 2nd camera's coordinate system: $Q = R * P + \mathbf{t}$
      - $Q.z > 0$?

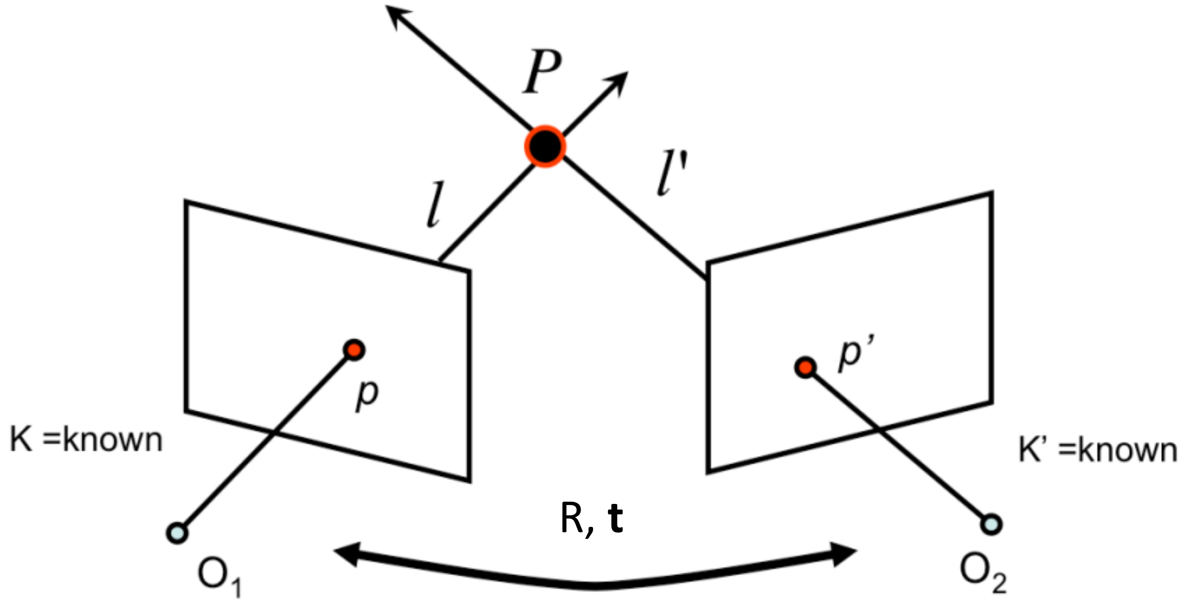


Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Extracting corresponding image points (next lecture)
    - Recover camera motion
    - Triangulation
  - 3D from more views
    - Structure from motion

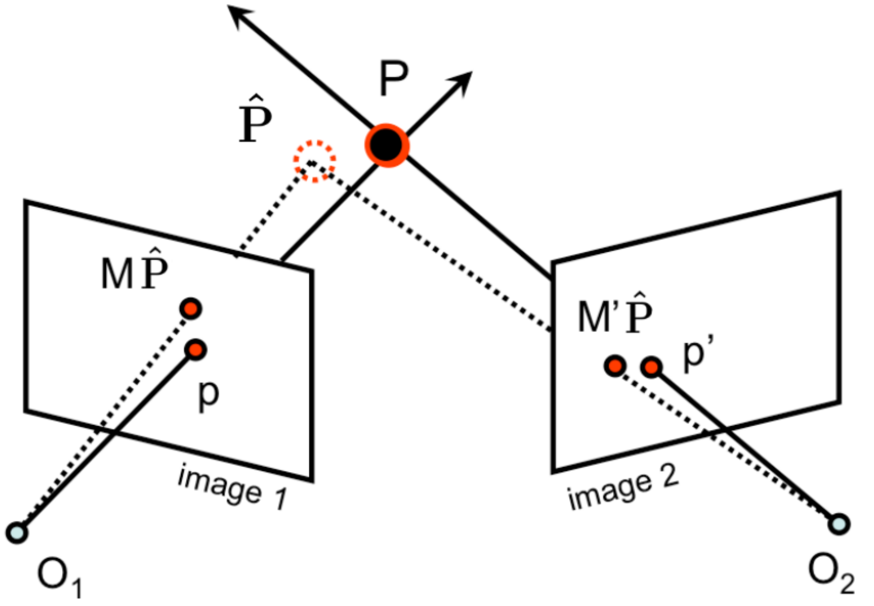
Triangulation

- 3D point from its projection into two views
  - Compute two lines of sight from K, R, and $\mathbf{t}$
  - In theory, P is the ∩ of the two lines of sight



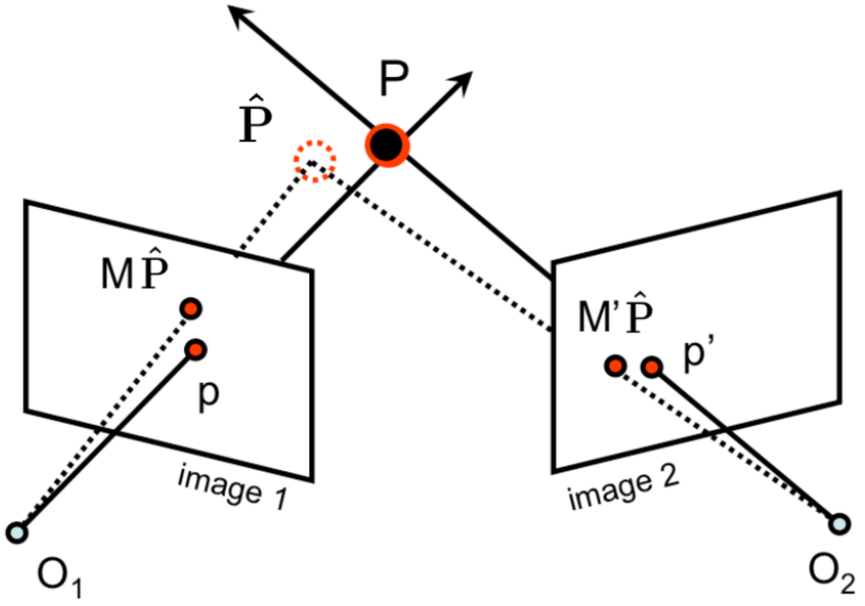
Triangulation

- 3D point from its projection into two views
  - Compute two lines of sight from K, R, and $\mathbf{t}$
  - In theory, P is the ∩ of the two lines of sight
    - Straightforward and mathematically sound
    - Does not work well
      - Noise in observation
      - Discreet pixel representation
      - Inaccuracy in K, R, $\mathbf{t}$



Triangulation

- 3D point from its projection into two views
  - Compute two lines of sight from K, R, and $\mathbf{t}$
  - In theory, P is the ∩ of the two lines of sight
    - Straightforward and mathematically sound
    - Does not work well
      - Noise in observation
      - Discreet pixel representation
      - Inaccuracy in K, R, $\mathbf{t}$
  - Two approaches for triangulation
    - A linear method
    - A non-linear method



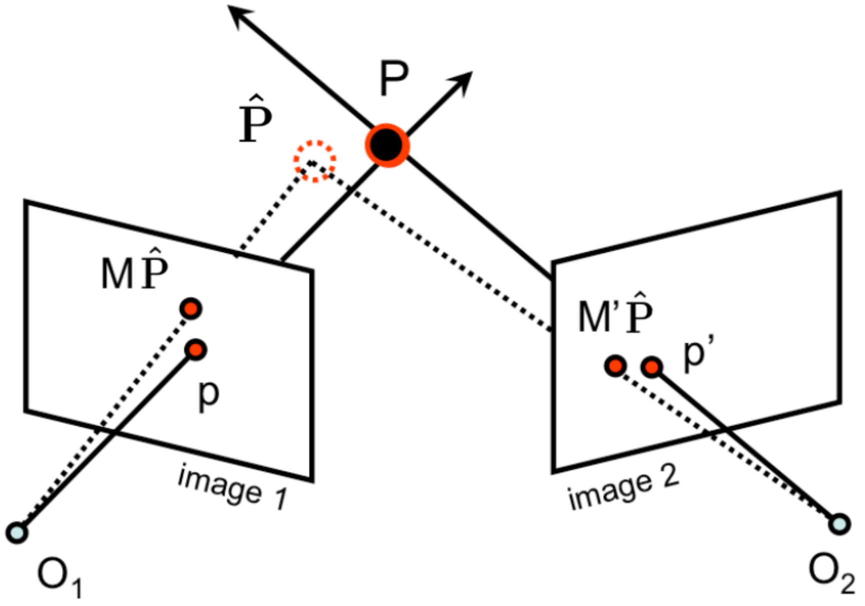
A Linear Method for Triangulation

Two image points

$$\mathbf{p} = M\mathbf{P} = (x, y, 1) \text{ and } \mathbf{p}' = M'\mathbf{P} = (x', y', 1)$$

By the definition of the cross product

$$\mathbf{p} \times (M\mathbf{P}) = 0$$



A Linear Method for Triangulation

Two image points

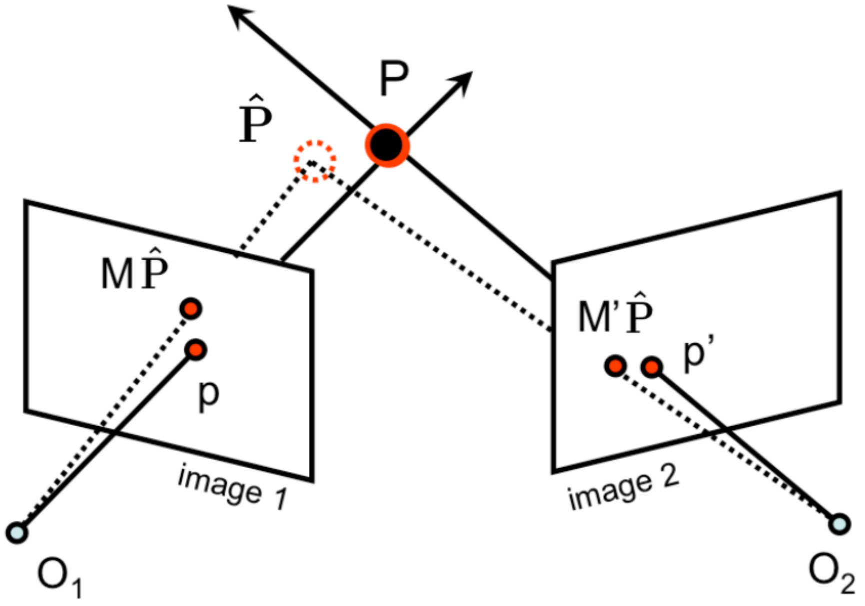
$$\mathbf{p} = M\mathbf{P} = (x, y, 1) \text{ and } \mathbf{p}' = M'\mathbf{P} = (x', y', 1)$$

By the definition of the cross product

$$\mathbf{p} \times (M\mathbf{P}) = 0$$

$$x(\mathbf{m}_3^T\mathbf{P}) - (\mathbf{m}_1^T\mathbf{P}) = 0$$
$$y(\mathbf{m}_3^T\mathbf{P}) - (\mathbf{m}_2^T\mathbf{P}) = 0$$
$$x(\mathbf{m}_2^T\mathbf{P}) - y(\mathbf{m}_1^T\mathbf{P}) = 0$$

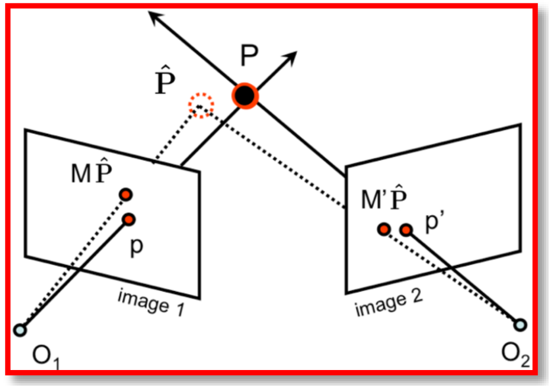
Solve for P?



A Linear Method for Triangulation

Two image points

$$\mathbf{p} = M\mathbf{P} = (x, y, 1) \text{ and } \mathbf{p}' = M'\mathbf{P} = (x', y', 1)$$



By the definition of the cross product

$$\mathbf{p} \times (M\mathbf{P}) = 0$$

Similar constraints from p' and *M'*

$$x(\mathbf{m}_3^T\mathbf{P}) - (\mathbf{m}_1^T\mathbf{P}) = 0$$
$$y(\mathbf{m}_3^T\mathbf{P}) - (\mathbf{m}_2^T\mathbf{P}) = 0$$
$$x(\mathbf{m}_2^T\mathbf{P}) - y(\mathbf{m}_1^T\mathbf{P}) = 0$$

$$A = \begin{bmatrix} x\mathbf{m}_3^T - \mathbf{m}_1^T \\ y\mathbf{m}_3^T - \mathbf{m}_2^T \\ x'\mathbf{m}_3'^T - \mathbf{m}_1'^T \\ y'\mathbf{m}_3'^T - \mathbf{m}_2'^T \end{bmatrix}$$

$$AP = 0$$

A Linear Method for Triangulation

- Advantages
  - Easy to solve and very efficient
  - Can handle multiple views
  - Used as initialization to advanced methods (e.g., non-linear methods and SfM)

$$AP = 0$$

$$A = \begin{bmatrix} x\mathbf{m}_3^T - \mathbf{m}_1^T \\ y\mathbf{m}_3^T - \mathbf{m}_2^T \\ x'\mathbf{m}_3'^T - \mathbf{m}_1'^T \\ y'\mathbf{m}_3'^T - \mathbf{m}_2'^T \end{bmatrix}$$

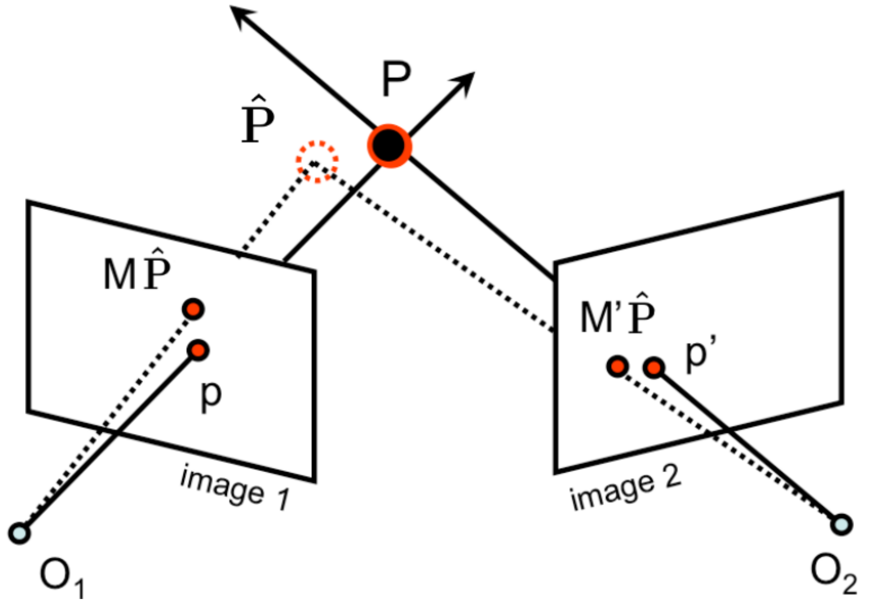
The Non-linear Method for Triangulation

- Formulation

$$\min_{\hat{\mathbf{P}}} \|M\hat{\mathbf{P}} - \mathbf{p}\|^2 + \|M'\hat{\mathbf{P}} - \mathbf{p}'\|^2$$

Reprojection error

- Solving it
  - Methods
    - Levenberg-Marquardt
    - Gauss-Newton's method
  - Requires good initialization
    - 3D coordinates from the linear method

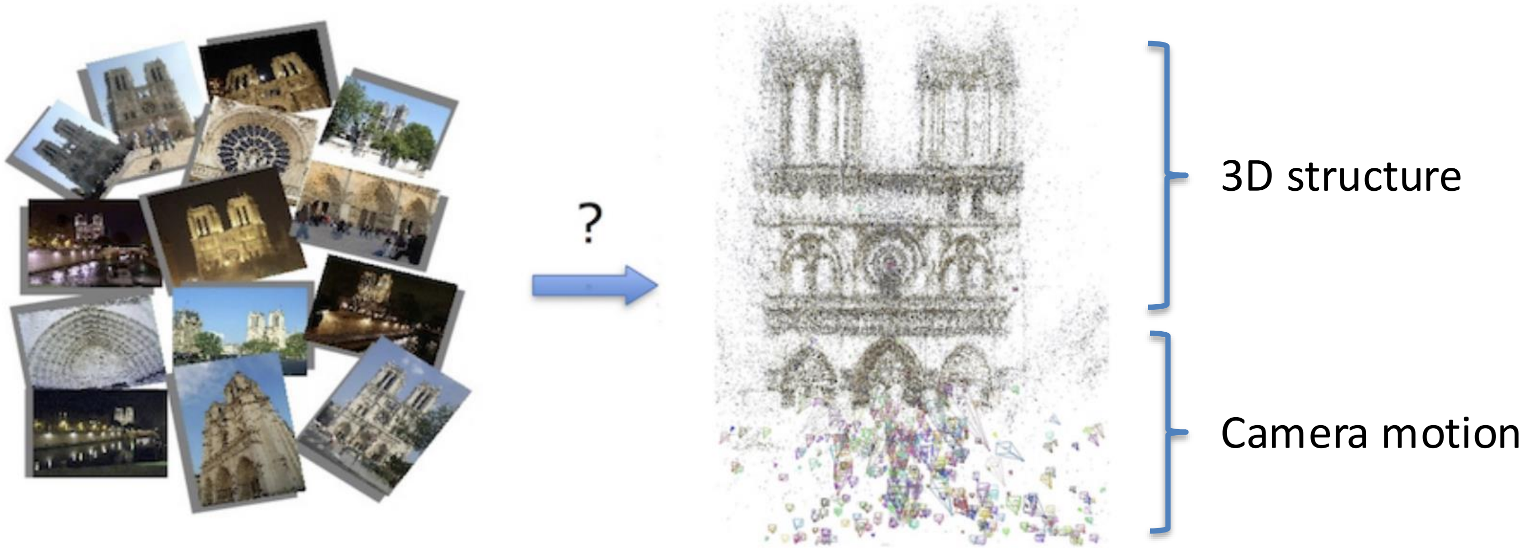


Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Extracting corresponding image points (next lecture)
    - Recover camera motion
    - Triangulation
  - 3D from more views
    - Structure from motion

Structure from Motion

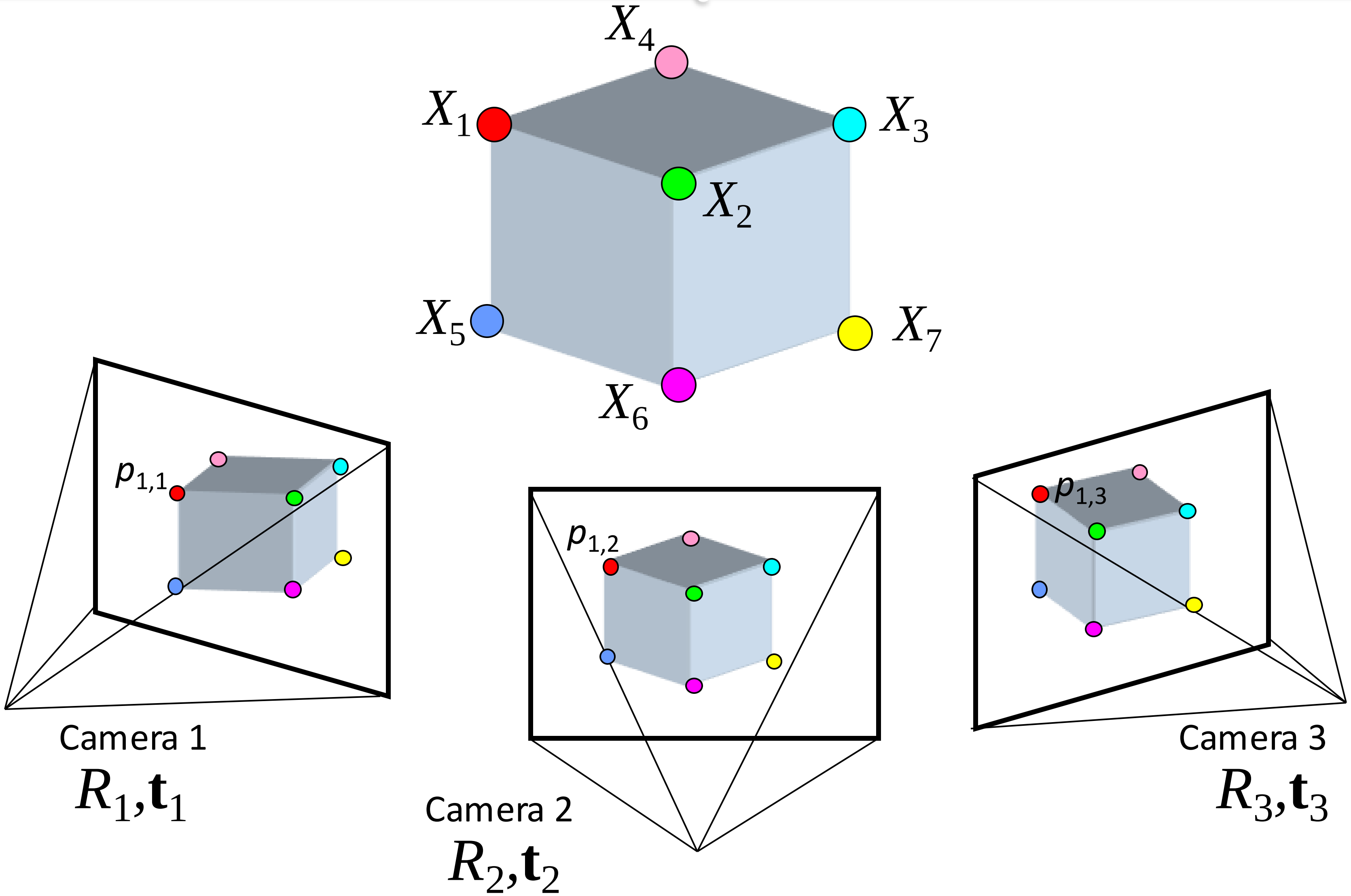
- Structure?
  – 3D geometry of the scene/object
- Motion?
  – Camera locations and orientations



Structure from Motion

- Structure
  - 3D geometry of the scene/object
- Motion
  - Camera locations and orientations
- Structure from Motion
  - Compute the geometry from moving cameras
  - Simultaneously recovering structure and motion
  - Extension of 2-view reconstruction to multiple views

Structure from Motion



Bundle Adjustment

- Minimize total re-projection error:

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

How is it different from the non-linear method for triangulation?

Bundle Adjustment

- Minimize total re-projection error:

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

- Optimized using non-linear least squares
  - e.g., Levenberg-Marquardt

Bundle Adjustment

- Minimize total re-projection error:

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

- Optimized using non-linear least squares
- Initialization
  - From chained 2-view reconstruction
    - Relative motion can be estimated from the corresponding image points
    - 3D points can be estimated from the relative motion using triangulation
  - Global optimization techniques allow poses and 3D structures to be initialized arbitrarily.

Bundle Adjustment

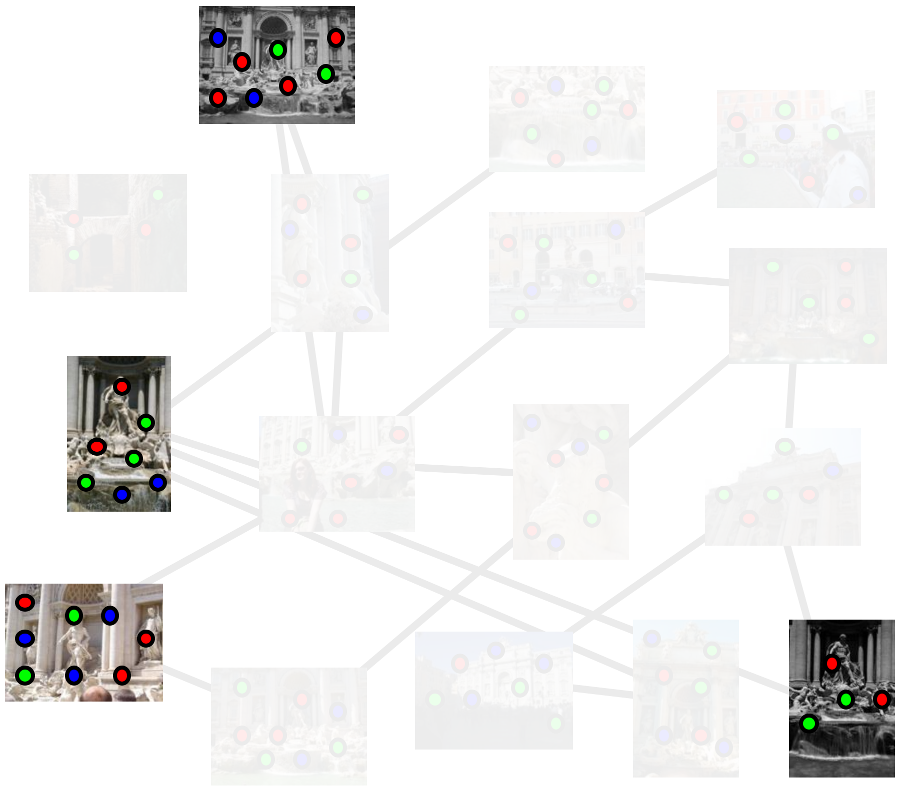
$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

- What are the variables?
  - Camera extrinsic parameters (and intrinsic parameters)
  - Coordinates of the 3D points
- How many variables per camera?
- How many variables per point?

Example: 100 input photos, 100,000 3D points

Very large optimization problem

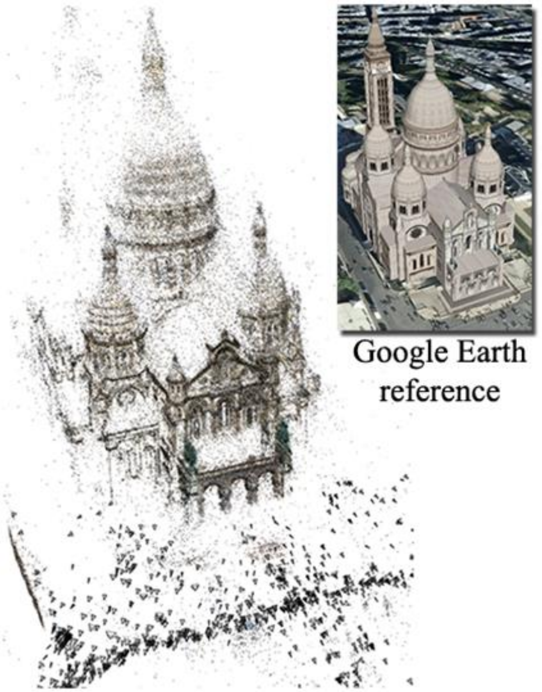
Incremental SfM



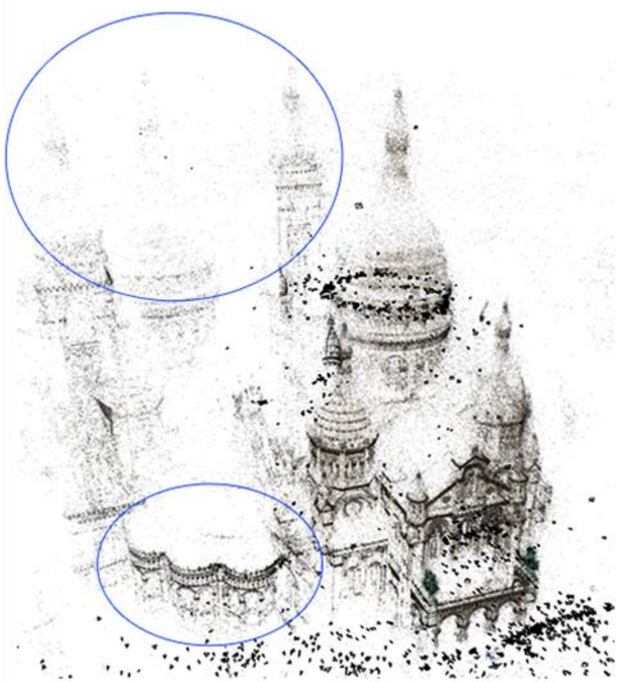
Structure from Motion

Failure Cases

- Repetitive structures

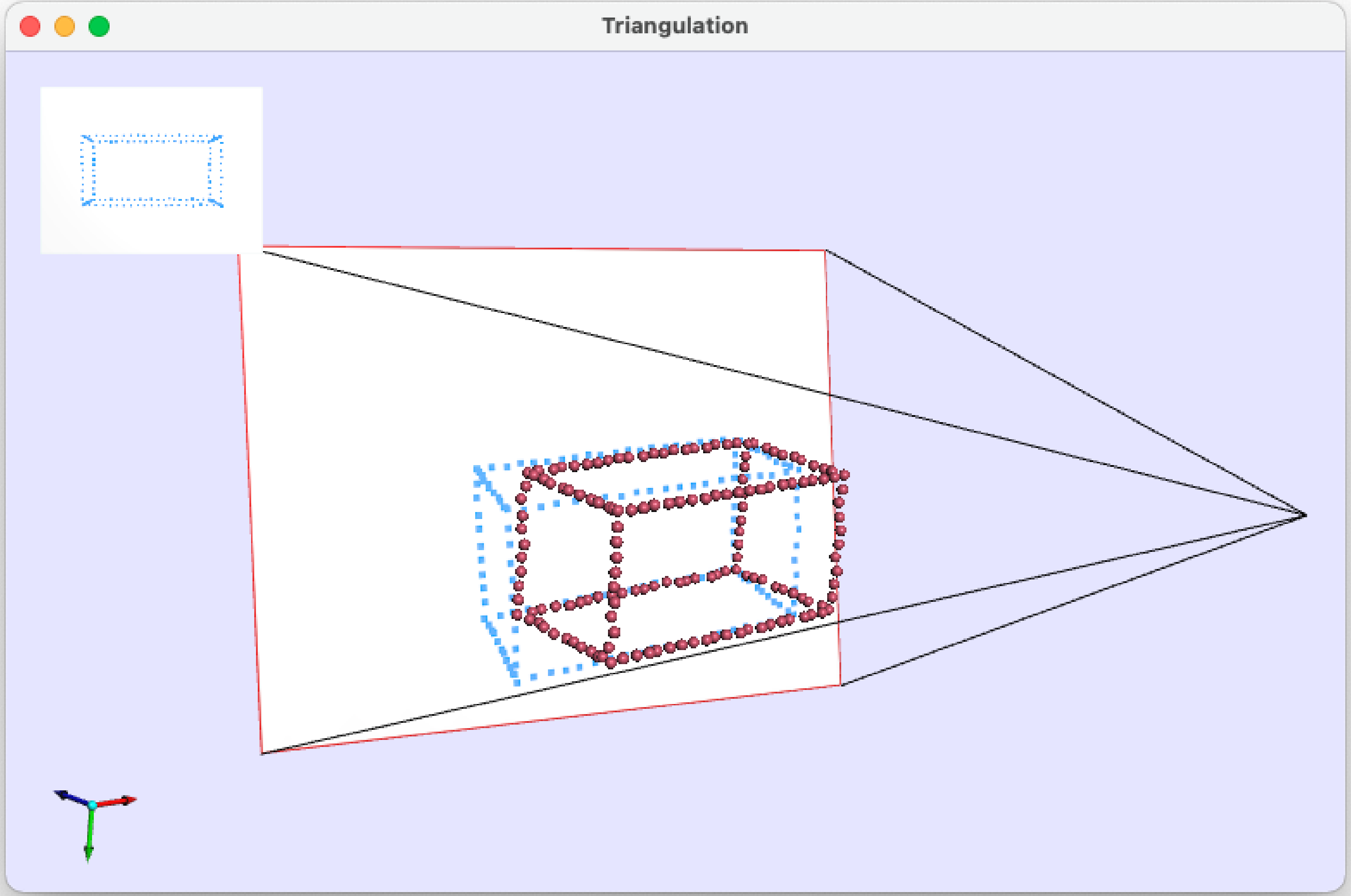


Ground truth



Broken model

A2: Triangulation



Next Lecture

- Image matching
  - Obtaining corresponding image points