Lecture
Reconstruct 3D Geometry

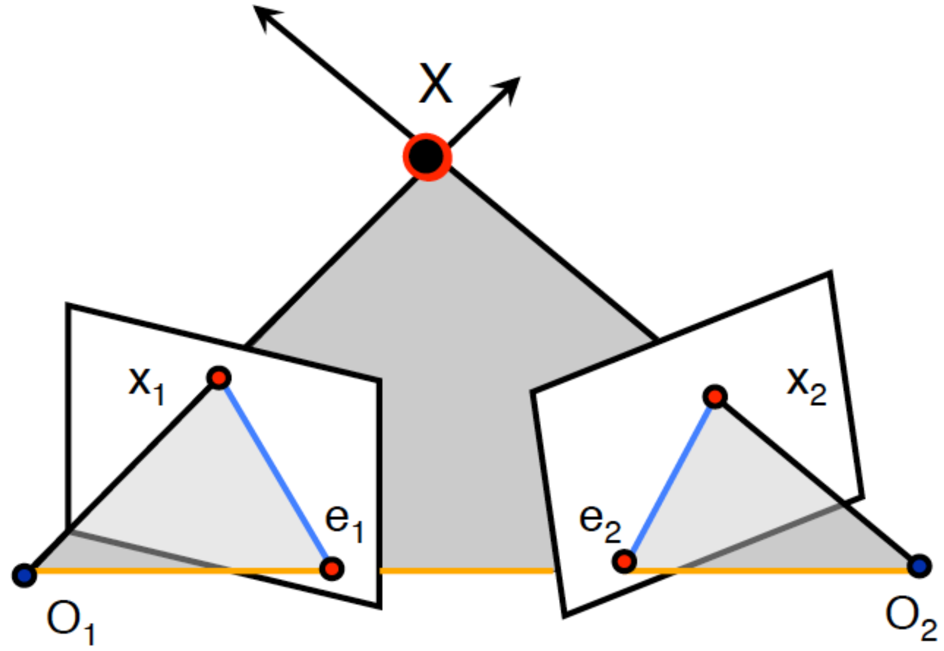
Liangliang Nan

Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

Review of Epipolar Geometry

- Epipolar Geometry
  - Baseline
    - The line between the two camera centers
  - Epipolar plane
    - The plane defined by X, O_1, and O_2
  - Epipoles
    - ∩ of baseline and image plane
    - Projection of the other camera center
  - Epipolar lines
    - ∩ of epipolar plane with the image plane



Review of Epipolar Geometry

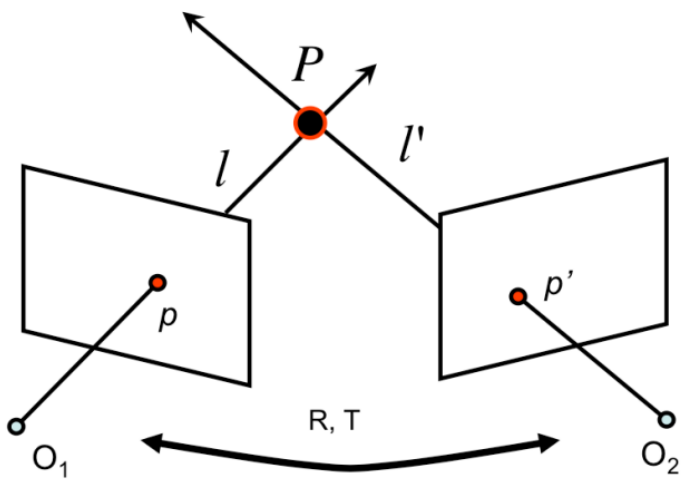
- Essential matrix
  - Canonical camera assumption

$$p'^T E\, p = 0,\ E = [T_X]R \qquad K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Fundamental matrix (most important concept in 3DV)

$$p'^T F\, p = 0,\ F = K'^{-T}[T_\times]RK^{-1}$$

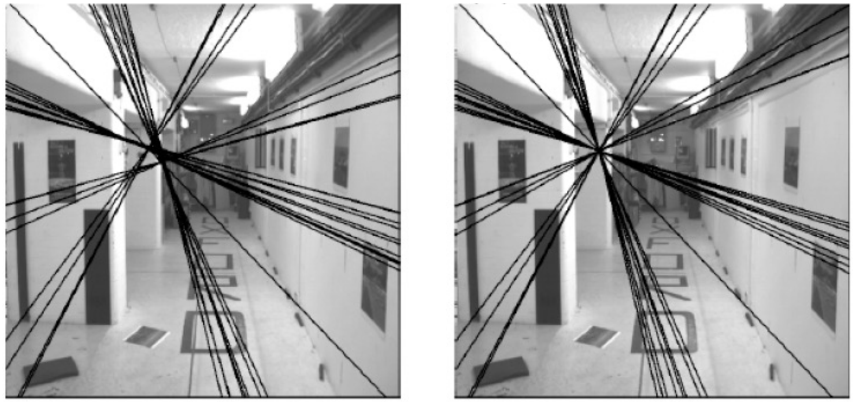
  - Relate matching image points of different views
    - No need 3D location
    - No need camera intrinsic and extrinsic parameters



Review of Epipolar Geometry

- Fundamental matrix
  - 3 by 3
  - homogeneous matrix
  - 7 degrees of freedom
    - 9 elements
    - scale ambiguity (scale doesn’t matter)
    - rank(F) = 2
      - The potential matching point is located on a line

$$p'^T F\, p = 0, \qquad F = K'^{-T}[T_\times]RK^{-1}$$

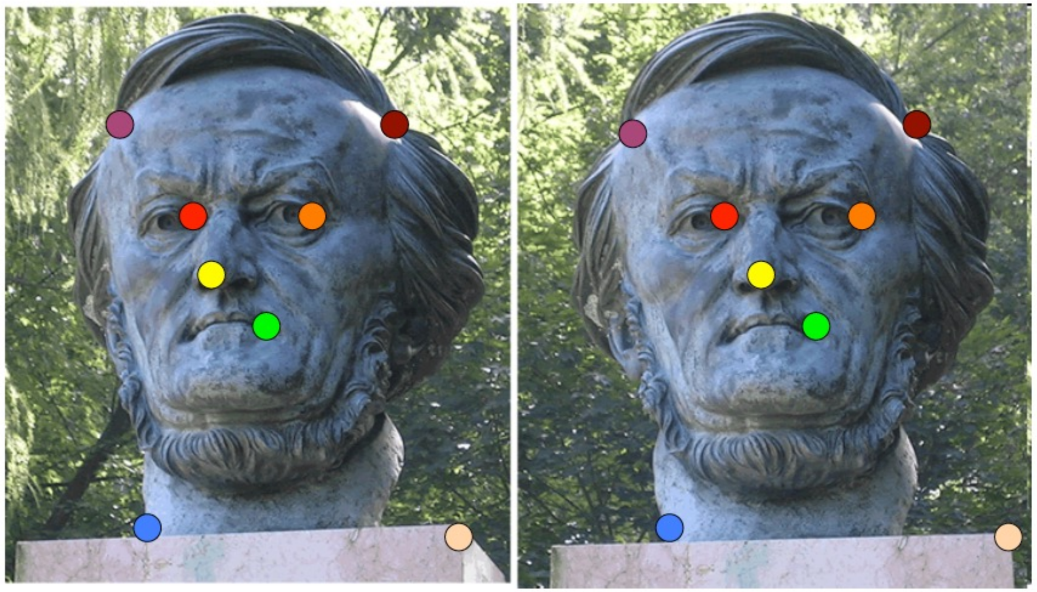


Review of Epipolar Geometry

- Recover F from corresponding image points
  - 8 unknown parameters
  - Each point pair gives a single linear constraint

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

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



Review of Epipolar Geometry

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

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

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

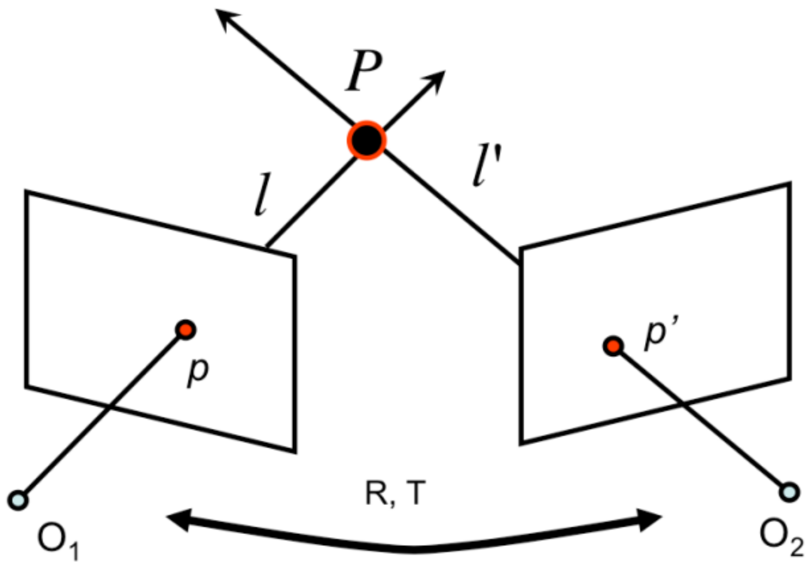
Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

3D from 2 Views

- The general idea

Recover 3D coordinates from corresponding image points

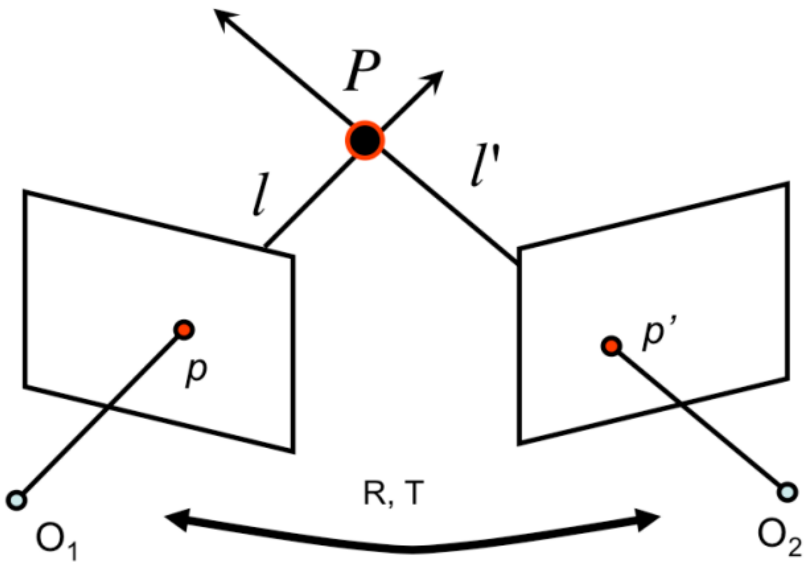


3D from 2 Views

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

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

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



Today’s Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
  - Sensitive to noise
  - Sensitive to origin of coordinates
  - Sensitive to scales

Same scale, different origins

Image taken using different focal lengths

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
  - – Poor numerical conditioning → fix by scaling the data

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
- Normalized 8-point algorithm
  - Idea: normalize before constructing the equations
    - Translation: the centroid of the image points is at origin
    - Scaling: average distance of points from origin is $\sqrt{2}$

$$q_i = T p_i \qquad q'_i = T' p'_i$$

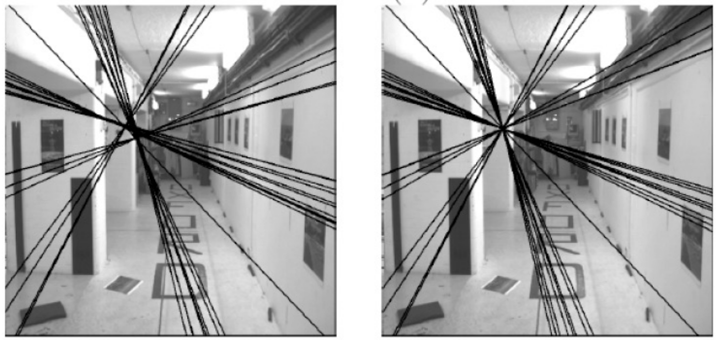
Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
- Normalized 8-point algorithm
  – Idea: normalize before constructing the equations
  – Construct linear system using the normalized points
    - Same as in the original 8-point algorithm
  – Solve using SVD
    - Same as in the original 8-point algorithm

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
- Normalized 8-point algorithm
  - – Idea: normalize before constructing the equations
  - – Construct linear system using the normalized points
  - – Solve using SVD
  - – Constraint enforcement
    - rank(F) = 2

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



Left : Uncorrected F – epipolar lines are not coincident.

Right : Epipolar lines from corrected F.

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
- Normalized 8-point algorithm
  - Idea: normalize before constructing the equations
  - Construct linear system using the normalized points
  - Solve using SVD
  - Constraint enforcement
    - rank(F) = 2

$$\hat{F} = U \Sigma V^T \qquad F = U \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

Recover F from Matched Image Points

- 8-point algorithm (>= 8 point pairs)
- Normalized 8-point algorithm
  - Idea: normalize before constructing the equations
  - Construct linear system using the normalized points
  - Solve using SVD
  - Constraint enforcement
  - De-normalization
    - Apply the **inverse** of the transformation

$$q_i = T p_i \qquad q_i' = T' p_i'$$

$$F = T'^T F_q T$$

See handout on “Epipolar geometry”

Today’s Agenda

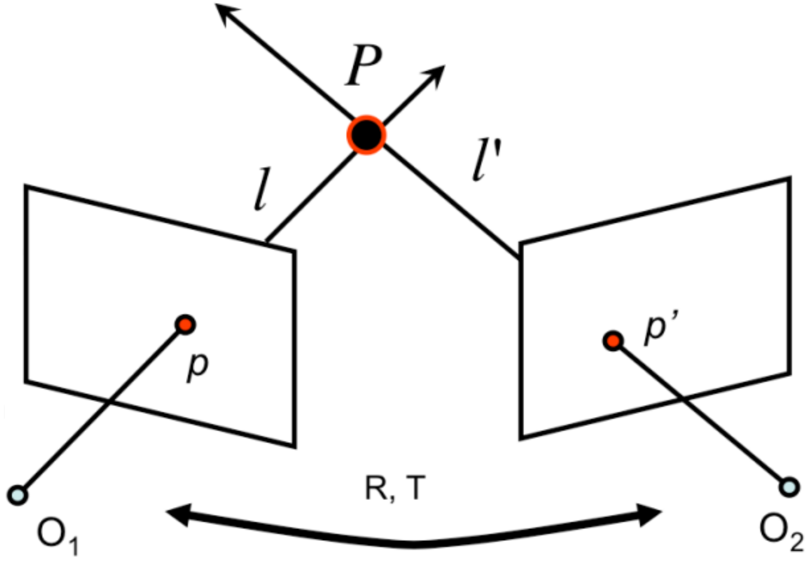
- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
  - Known intrinsic parameters
    - Calibration
    - Estimation + refinement

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

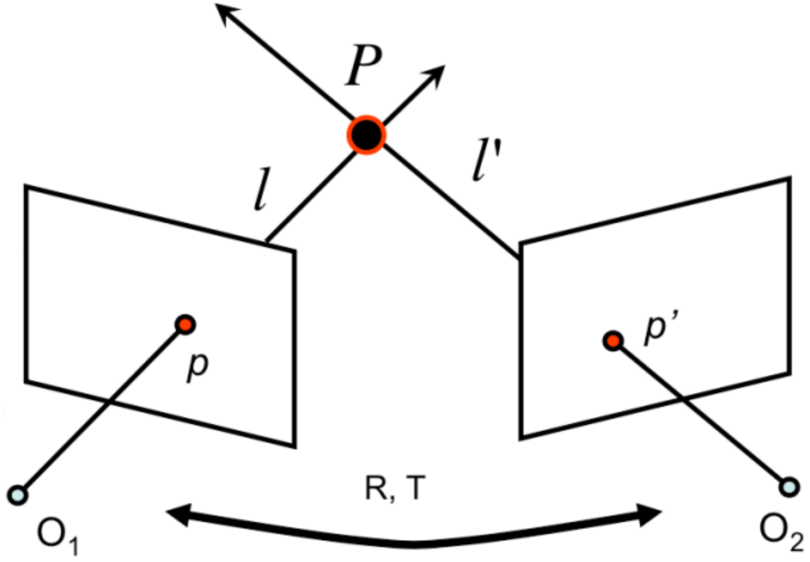
$$E = [t_{\times}]R = K'^{T}FK$$



Relative Pose from Fundamental Matrix

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

$$E = [t_\times]R$$



Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - SVD of E

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

$$E = U \Sigma V^T$$

Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - SVD of E
  - determinant(R) > 0
    - Two potential values

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

$$E = U \Sigma V^T$$

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

Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - SVD of E
  - determinant(R) > 0
    - Two potential values
  - T up to a sign
    - Two potential values
    - Last column of U
    - Corresponds to smallest singular value

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

$$E = U \Sigma V^T$$

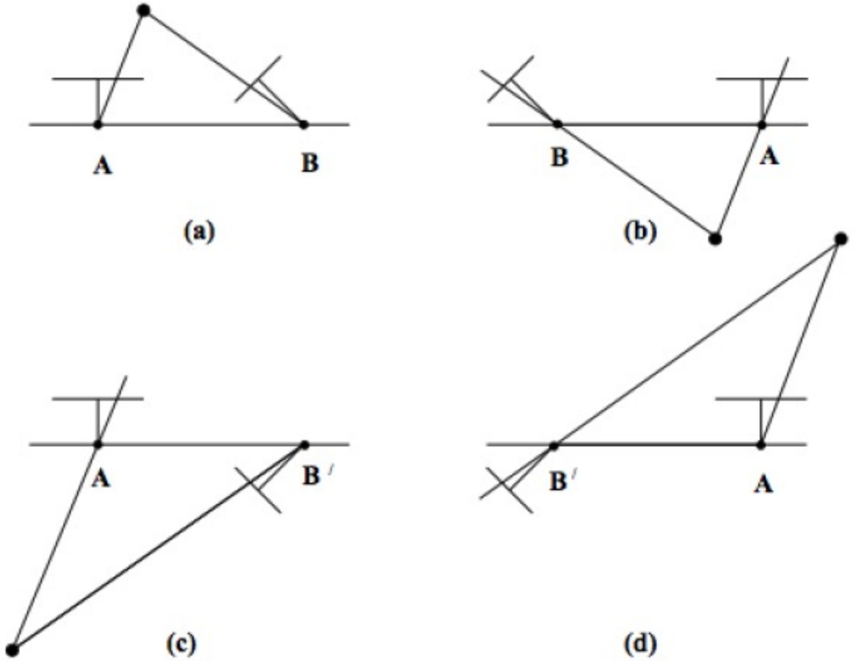
$$R = (\det UWV^T)UWV^T \text{ or } (\det UW^TV^T)UW^TV^T$$

$$t = \pm U \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \pm u_3$$

Relative Pose from Fundamental Matrix

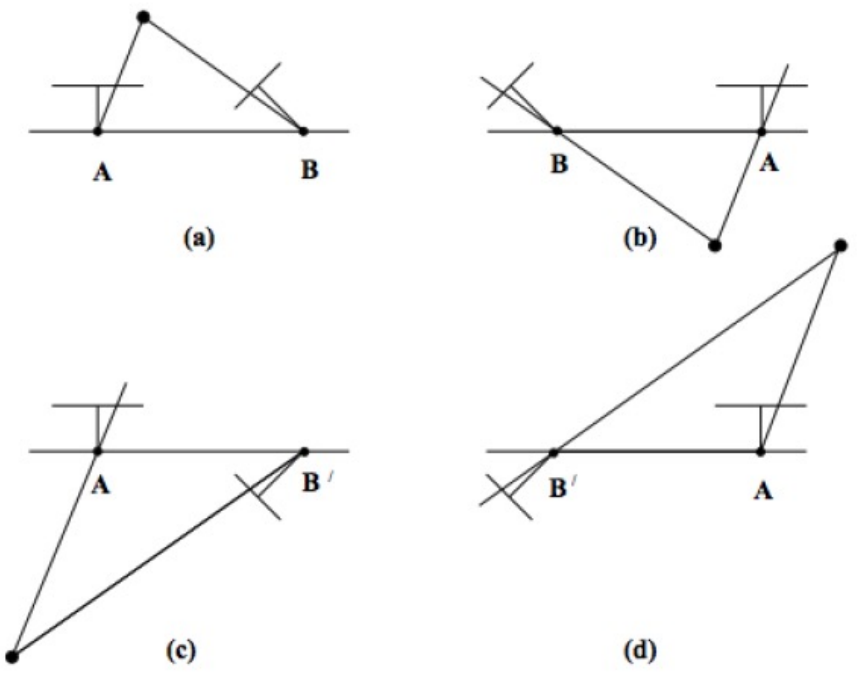
- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - T: two potential values

which is the correct configuration



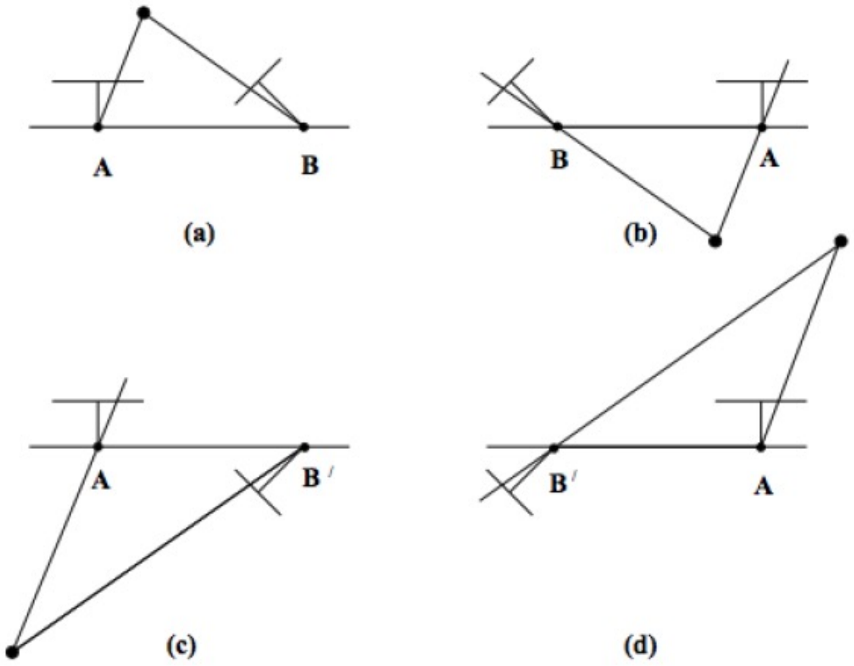
Relative Pose from Fundamental Matrix

- Essential matrix from fundamental matrix
- Relative pose from essential matrix
  - R: two potential values
  - T: two potential values
  - 3D points must be in front of both cameras



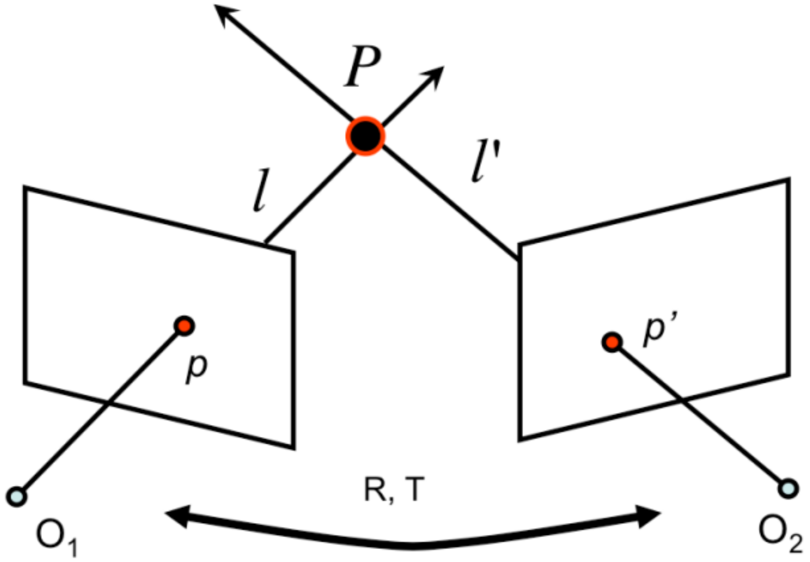
Relative Pose from Fundamental Matrix

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



Relative Pose from Fundamental Matrix

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

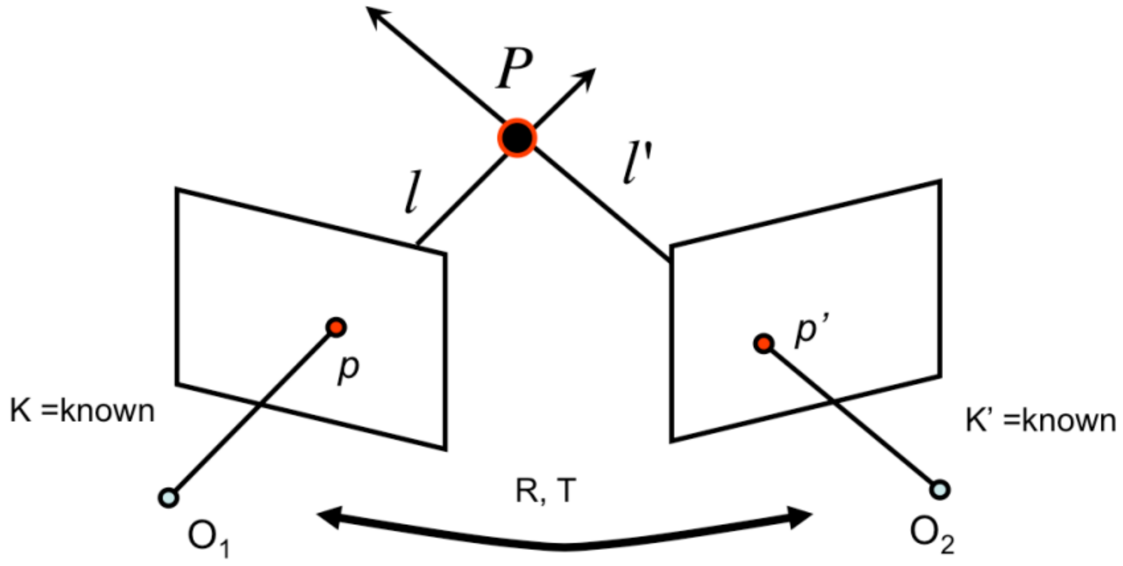


Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

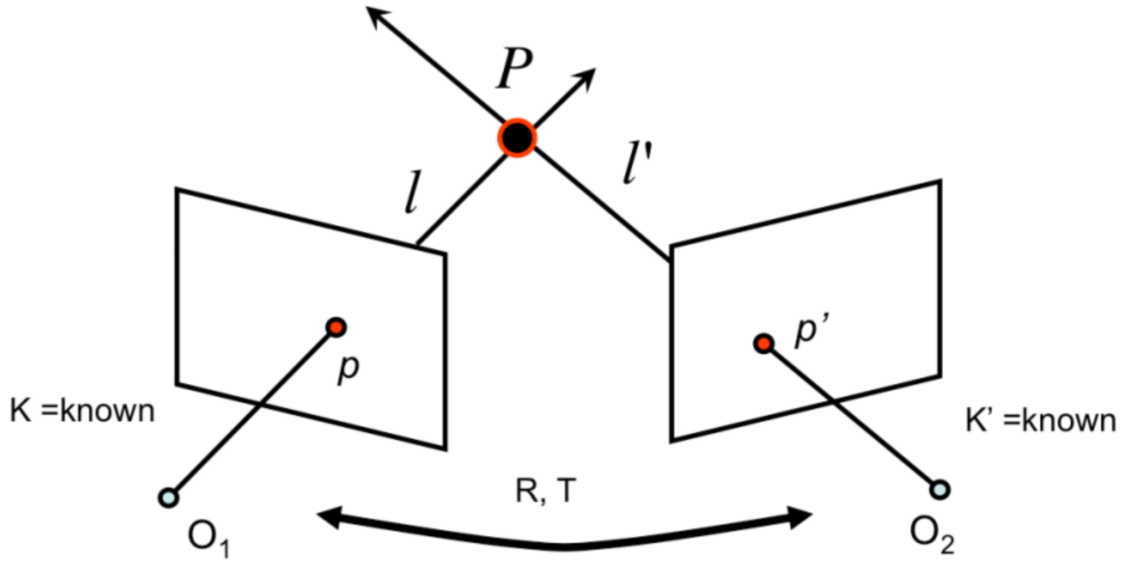
Triangulation

- Find coordinates of 3D point from its projection into two views
  - Known camera intrinsic parameters (K)
  - Known relative orientation (R) and offset (t)



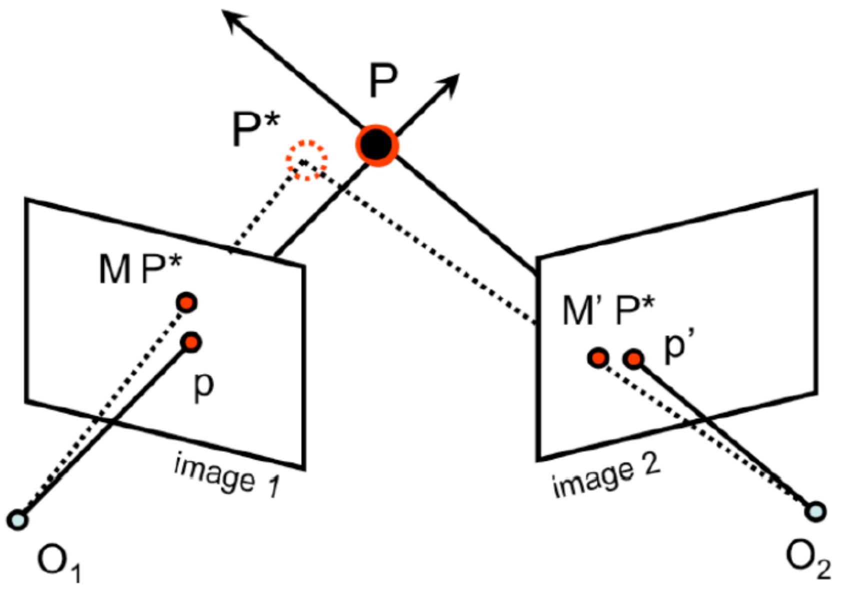
Triangulation

- Find coordinates of 3D point from its projection into two views
  - Known camera intrinsic parameters (K)
  - Known relative orientation (R) and offset (t)
  - In theory, P is ∩ of the two lines of sight



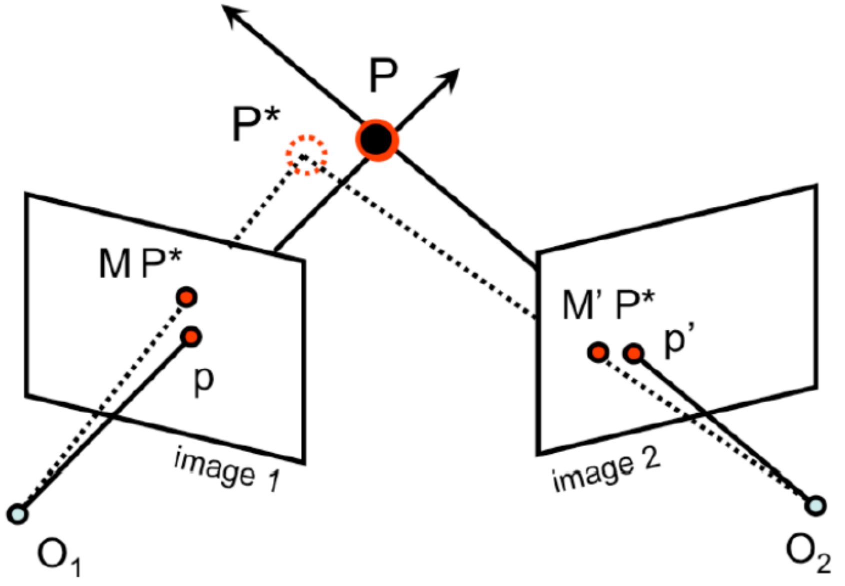
Triangulation

- Find coordinates of 3D point from its projection into two views
  - Known camera intrinsic parameters (K)
  - Known relative orientation (R) and offset (t)
  - In theory, P is ∩ of the two lines of sight
    - Straightforward and mathematically sound
    - Do not work well
      - Noisy in observation
      - K, R, t are not precise



Triangulation

- Find coordinates of 3D point from its projection into two views
  - Known camera intrinsic parameters (K)
  - Known relative orientation (R) and offset (t)
  - In theory, P is ∩ of the two lines of sight
    - Straightforward and mathematically sound
    - Do not work well
      - Noisy in observation
      - K, R, t are not precise
  - Two approaches for triangulation
    - A linear method and
    - A non-linear method



A Linear Method for Triangulation

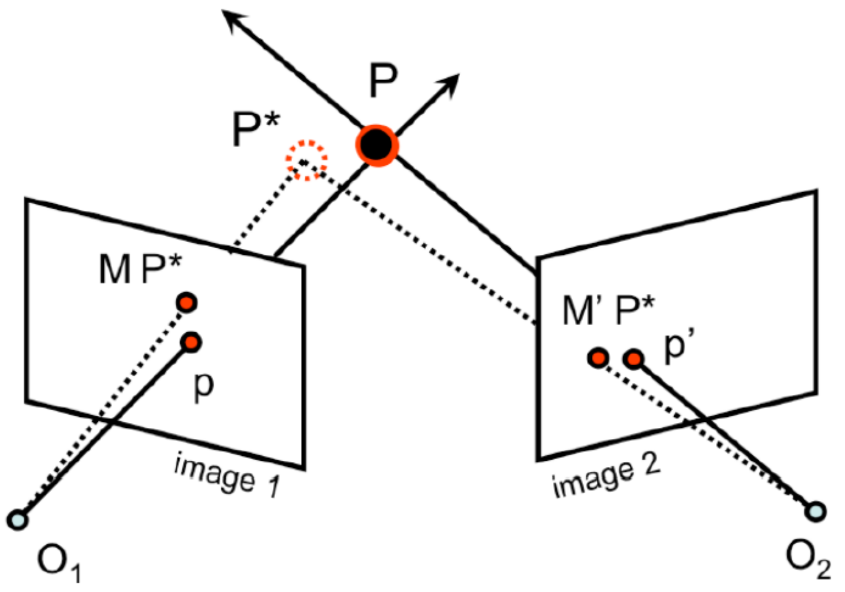
Two image points

$$p = MP = (x, y, 1)$$

$$p' = M'P = (x', y', 1)$$

By the definition of the cross product

$$p \times (MP) = 0$$



A Linear Method for Triangulation

Two image points

$$p = MP = (x, y, 1)$$

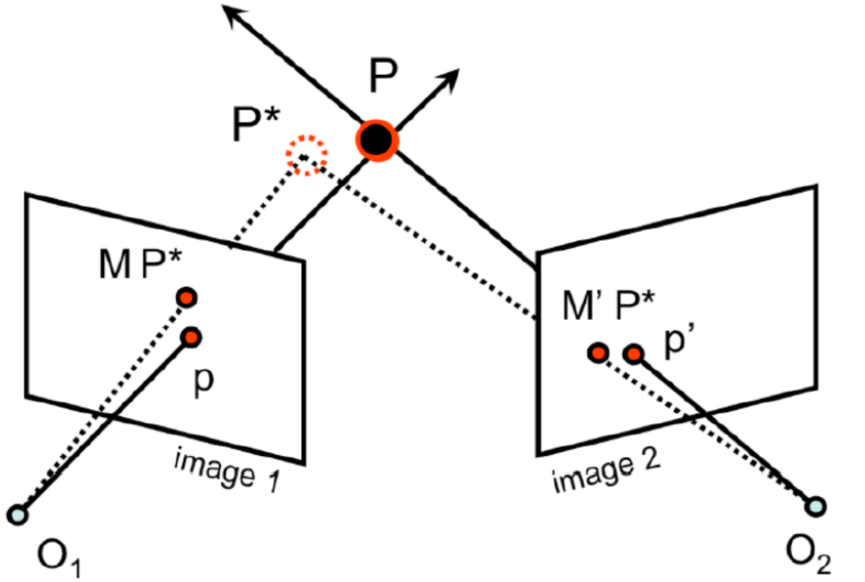
$$p' = M'P = (x', y', 1)$$

By the definition of the cross product

$$p \times (MP) = 0$$

$$x(M_3P) - (M_1P) = 0$$
$$y(M_3P) - (M_2P) = 0$$
$$x(M_2P) - y(M_1P) = 0$$

Solve for P?



A Linear Method for Triangulation

Two image points

$$p = MP = (x, y, 1)$$

$$p' = M'P = (x', y', 1)$$

By the definition of the cross product

$$p \times (MP) = 0$$

Similar constraints can also be formulated for p' and M'.

$$x(M_3P) - (M_1P) = 0$$
$$y(M_3P) - (M_2P) = 0$$
$$x(M_2P) - y(M_1P) = 0$$

$$A = \begin{bmatrix} xM_3 - M_1 \\ yM_3 - M_2 \\ x'M_3' - M_1' \\ y'M_3' - M_2' \end{bmatrix}$$

$$AP = 0$$

A Linear Method for Triangulation

- Advantages
  - Easy to solve and very efficient
  - Any number of corresponding image points
  - Can handle multiple views
  - Used as initialization to advanced methods

$$AP = 0$$

$$A = \begin{bmatrix} xM_3 - M_1 \\ yM_3 - M_2 \\ x'M_3' - M_1' \\ y'M_3' - M_2' \end{bmatrix}$$

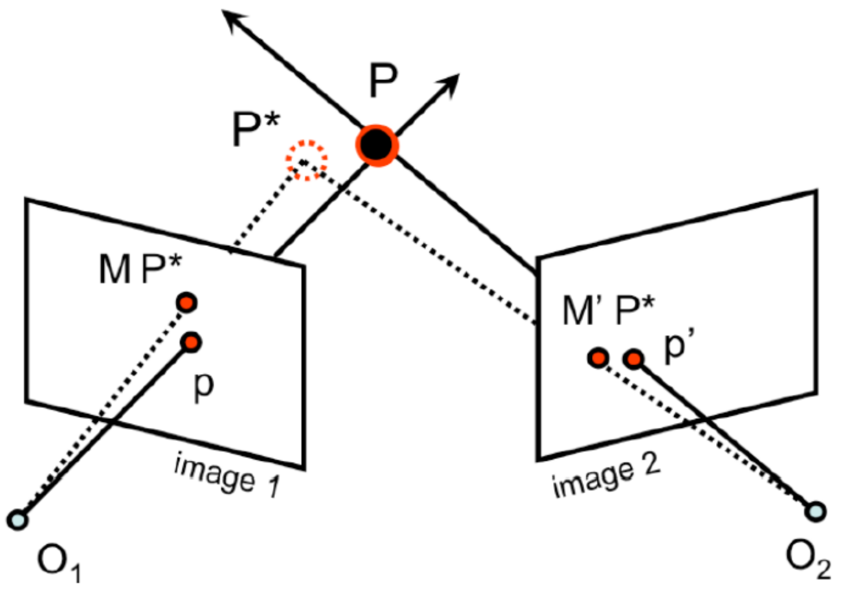
The Non-linear Method for Triangulation

- Minimize the reprojection error

$$\min_{\hat{P}} \sum_i \|M\hat{P}_i - p_i\|^2 + \|M'\hat{P}_i - p_i{}'\|^2$$

Reprojection error

  - Gauss-Newton's method
  - Levenberg-Marquardt

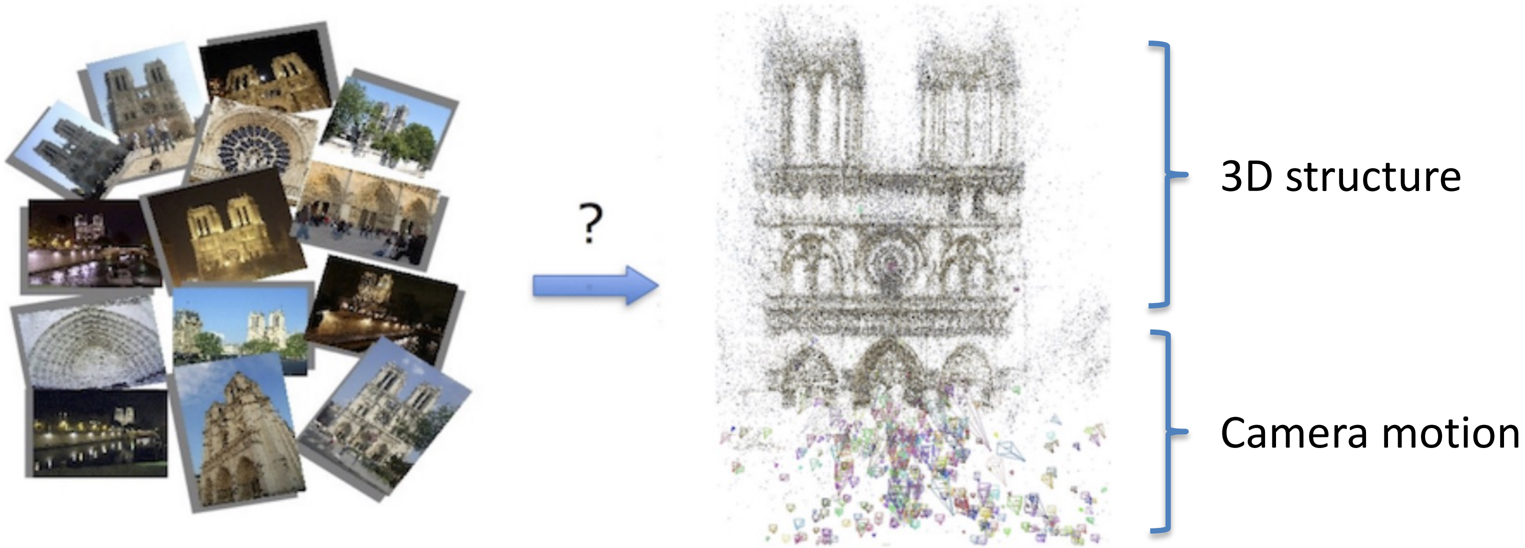


Today’s Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

Structure from Motion

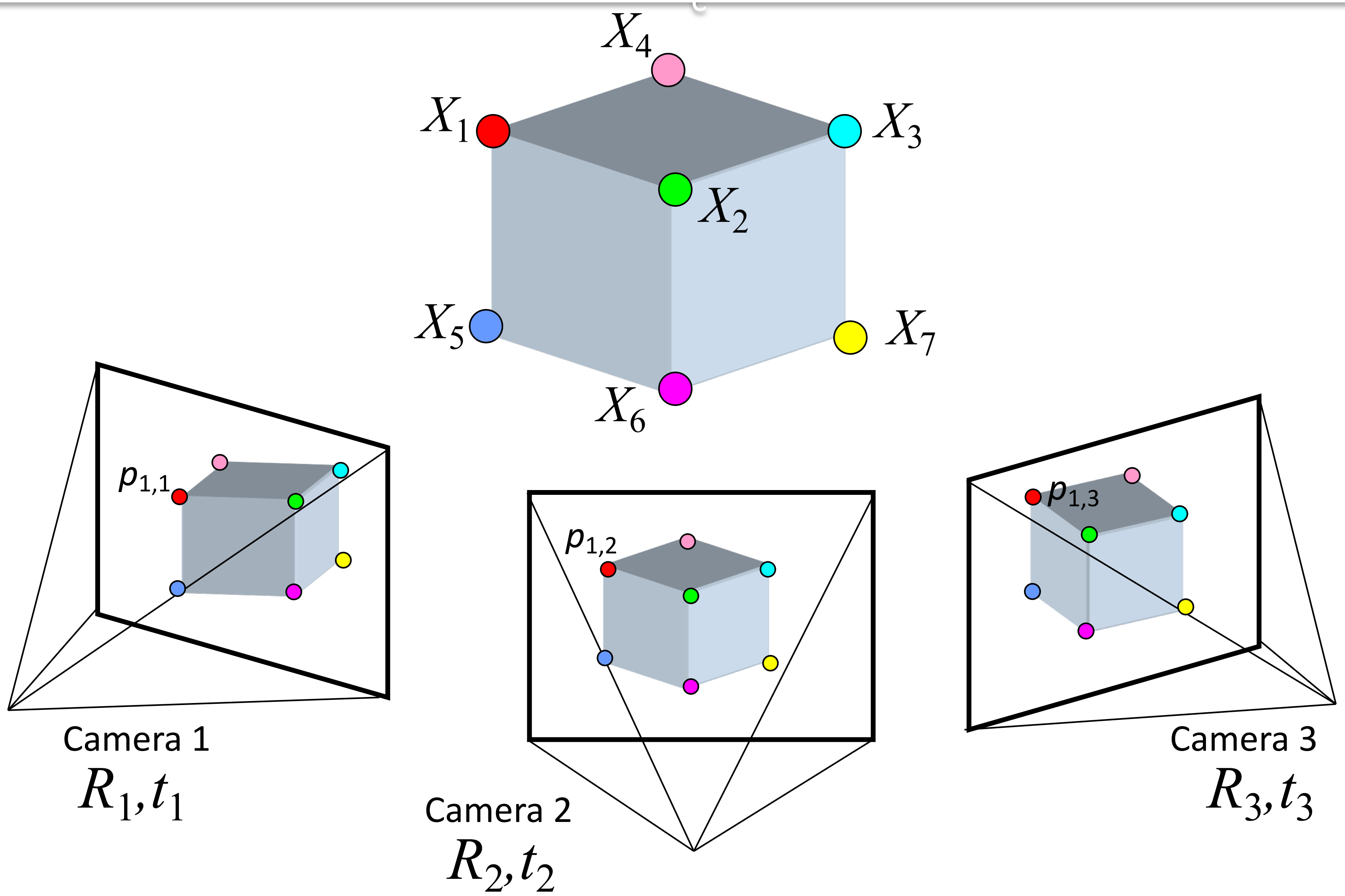
- Structure?
  - 3D geometry of the scene/object
- Motion?
  - Camera locations and orientations



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

Structure from Motion



Bundle Adjustment

- Minimize sum of squared re-projection errors:

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

predicted image points

observed image points

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

Bundle Adjustment

- Minimize sum of squared re-projection errors:

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

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

Bundle Adjustment

- Minimize sum of squared re-projection errors:

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

- Minimizing this function is called *bundle adjustment*
- Initialization
  - From chained 2-view reconstruction
    - Relative motion can be estimated from the corresponding images points
    - 3D points can be estimated from the relative motion using triangulation
  - Global optimization techniques allow poses and 3D structure are initialized arbitrarily.

Bundle Adjustment

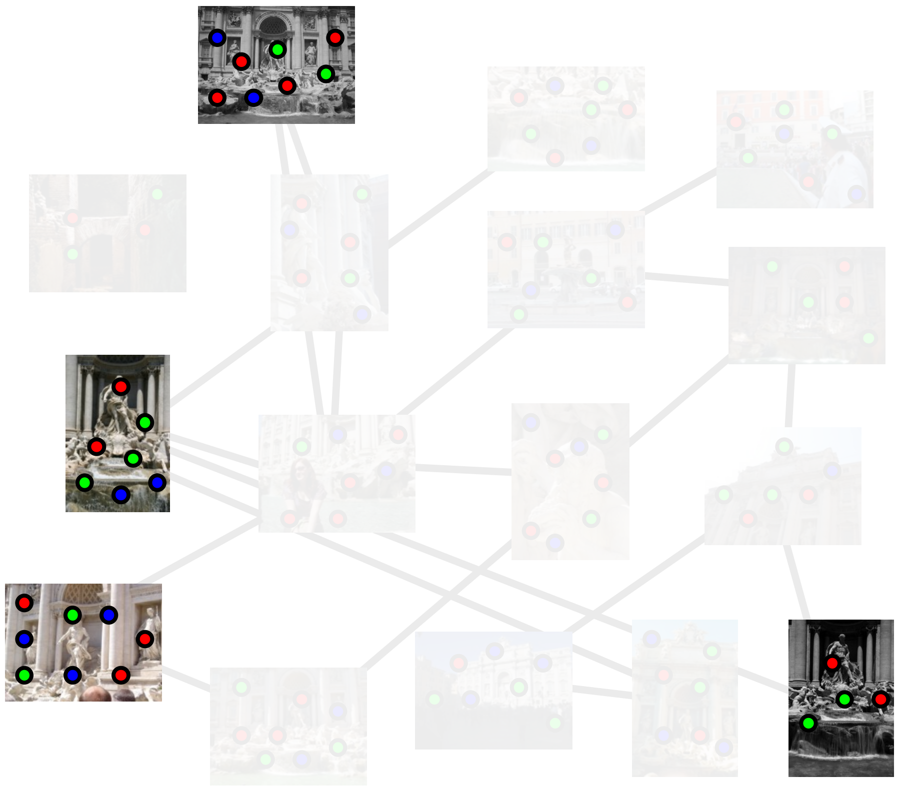
- What are the variables?
  - Camera intrinsic parameters, extrinsic parameters
  - Coordinates of the 3D points
- How many variables per camera?
- How many variables per point?

500 input photos

100,000 3D points

= Very large optimization problem

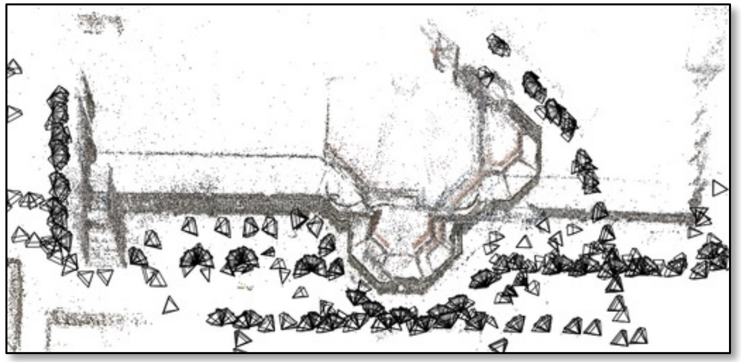
Incremental SfM



Structure from Motion

Failure Cases

- Repetitive structures



Today's Agenda

- Review of Epipolar Geometry
- Reconstruct 3D Geometry
  - 3D from 2 views
    - Estimate fundamental matrix
    - Recover relative pose
    - Triangulation
  - 3D from more views
    - Structure from motion
- Image Matching (ideas only, optional)

Image Matching

- Find Corresponding Image Points
  - Key points
  - Image descriptors
  - Matching

Key Points

- Distinctive locations
- Interesting or stand out
- Remain unchanged
  - Rotate, translate, shrink/expand, distortion …

Descriptors

- The way we **describe** the key points
  - High-dimensional vectors
- Feature (key point + descriptor)
  - e.g., SIFT, SURF

Matching

- Exhaustive search
- RANSAC

Image Matching

- SIFT

David Lowe

FOLLOW

Professor Emeritus, Computer Science Dept., University of British Columbia
Verified email at cs.ubc.ca - Homepage

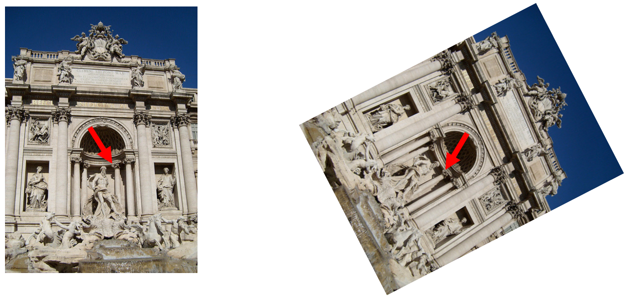
Computer Vision   Object Recognition

TITLE	CITED BY	YEAR
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	66861	2004
Object recognition from local scale-invariant features DG Lowe International Conference on Computer Vision, 1999, 1150-1157	22406	1999

Motivation

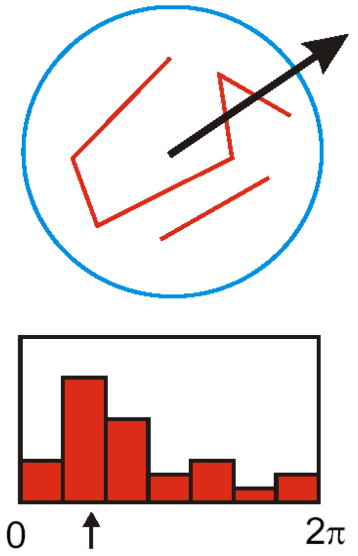
- Scale invariant
  – Decompose the image into multiple scales and describe the key points at each scale

- Rotation invariant
  – Dominant orientation of the gradient directions



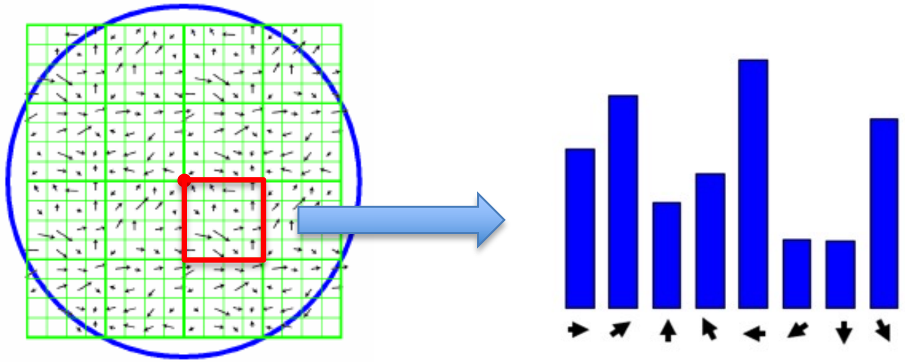
SIFT

- Assign one or more orientations to each key point based on local image gradient directions
  - Create histogram of local gradient directions at the selected scale
  - Assign orientation at the peak of smoothed histogram
  - Each key specifies (x, y, scale, orientation)



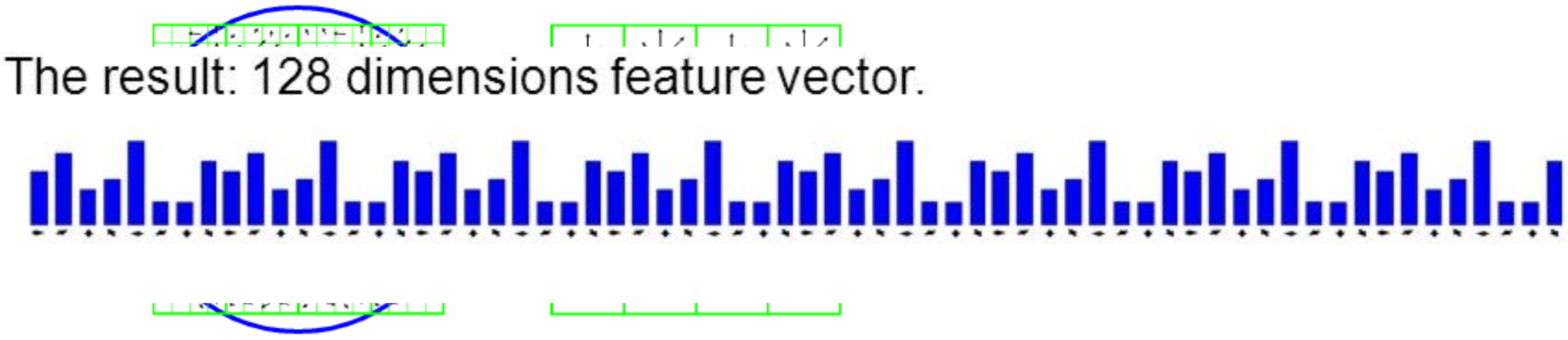
SIFT

- Key point description
  - 16 x 16 neighborhood of the key point
  - Divided into 16 sub-blocks of 4x4 size
  - 8 bin orientation histogram
    - Sum the gradient magnitude in each bin



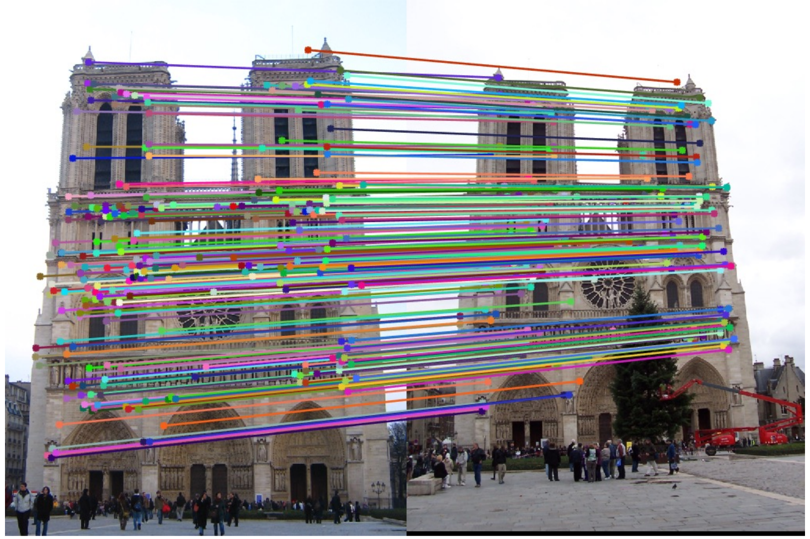
SIFT

- Key point description
  - 16 x 16 neighborhood of the key point
  - Divided into 16 sub-blocks of 4x4 size
  - 8 bin orientation histogram
  - Concatenate histograms of 16 sub-blocks



Feature Matching

- Matching key points
  - Ideal case: find the nearest neighbor
  - Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first



Feature Matching

- Matching key points
  - Ideal case: find the nearest neighbor
  - Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first

$$\text{Reject if } \frac{\text{closest-distance}}{\text{second-closest distance}} > 0.8$$

    - Can eliminate about 90% of false matches while discards only 5% correct matches

Feature Matching

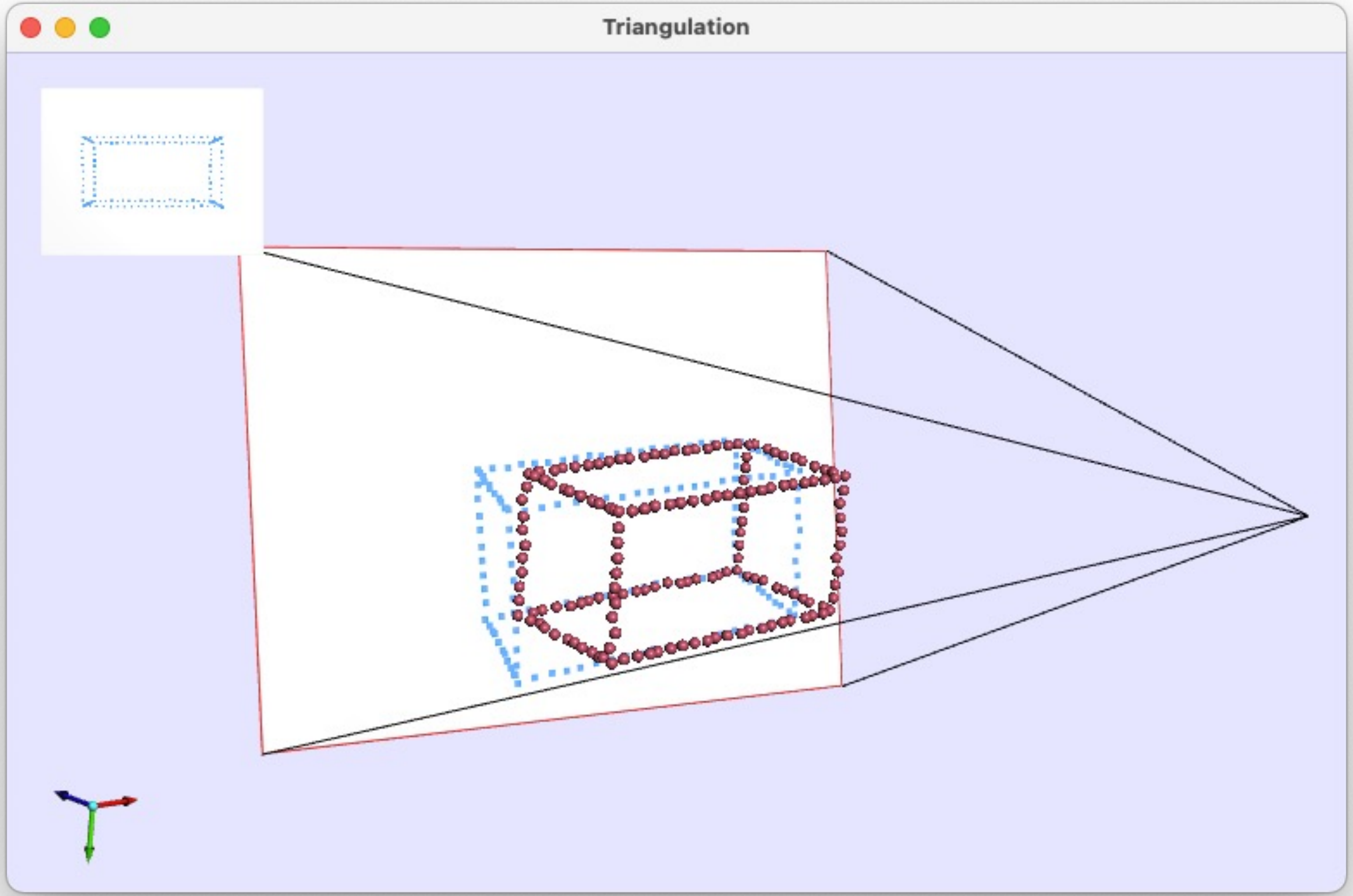
- Matching key points
  - Ideal case: find the nearest neighbor
  - Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first

$$\text{Reject if } \frac{\text{closest-distance}}{\text{second-closest distance}} > 0.8$$

    - Can eliminate about 90% of false matches while discards only 5% correct matches
  - Sophisticate strategies
    - RANSAC

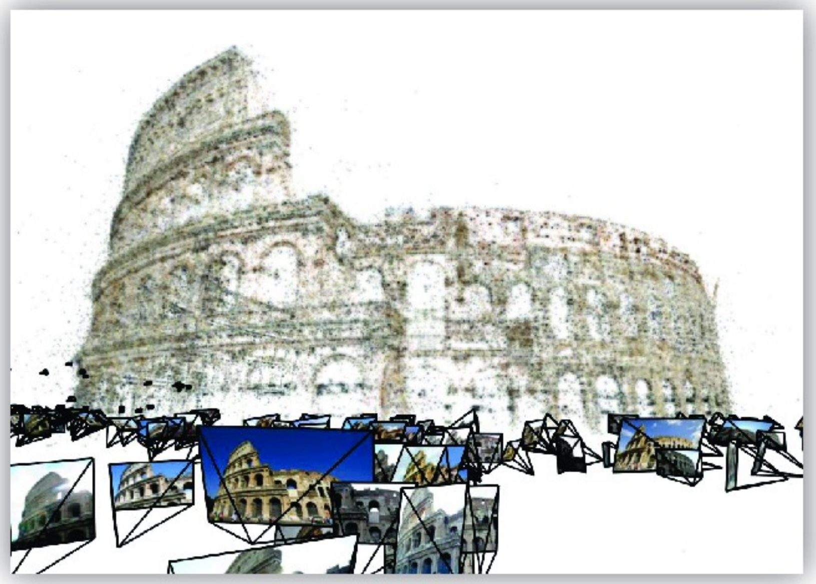
Lab: Image matching

A2: Triangulation



Next Lecture

- Multi-view Stereo
  - Obtaining dense point clouds



Images + camera information

Dense 3d point cloud