CSE 152: Computer Vision

Hao Su

Lecture 18: Review (I)

Features

- Filters
  - basic operation of a filter, effects of common filters, basic properties
- SIFT feature
- Bag of words
  - algorithm, limitation, concept of visual vocabulary, k-means clustering, geometric verification for matching

Math basics

- Basic concepts
- Spectral decomposition
- SVD, pseudo-inverse
- Geometric interpretation of linear transformation
- Basic matrix calculus (a cheetsheet may help)
- Solutions of linear equations
  - Under-determined, over-determined

Math basics

- Need to know how to solve least squares
- Need to know how to achieve low-rank matrix approximation
- Basic multi-variable Taylor expansion

Harris Corner

- Define a 2D function that is the appearance variation of a patch under patch location perturbation
- Taylor expansion helps to approximate this function as a quadratic function
- Analyze the Hessian of the quadratic approximation to infer the geometry of the 2D variation function, which is the cornerness score
- Need to know eigenvalues of the Hessian

PCA & RANSAC

- PCA: dimension reduction tool
- Geometric meaning of PCA
- Computation: PCA as SVD
- Optimization objective of PCA
- RANSAC algorithm
- When shall we use RANSAC?
- Pros & Cons of RANSAC

Classification

- K-nearest neighbor
  - basic algorithm
  - Pros & cons
- Bias-variance trade-off
- Cross-validation
- Linear classifier, logistic regression
  - What does the decision boundary like?
  - How to formulate the loss function?
- Comparing k-NN and logistic regression

CNN

- Why do we need deep learning?
- What is activation function and why we need them?
- Basic components of a CNN
  - Convolution layer, fully-connected layer, ReLU layer, pooling layer
  - Stride
  - Dropout layer

Network optimization

- Gradient descent
- Stochastic gradient descent
- Why we need momentum?
- Back-propagation is chain-rule
- Regularization techniques
  - L2 regularizer, dropout, data augmentation

Object detection & segmentation

- RCNN
- UNet

- We do not test any advanced material (e.g., XXX-RCNN)

3D Deep Learning

- Why directly extending 2D convolution to 3D is not a good idea?
- Multiview CNN, view pooling
- Volumetric CNN, common strategy to reduce computational complexity
- PointNet architecture, permutation invariance
- PointNet++

Camera Model

- Pinhole model
- Effect of aperture, lens
- What is intrinsic camera matrix?
- What is extrinsic camera matrix?
- Vanishing line

Multi-view geometry

- Triangulation
- Concepts:
  - Epipolar plane, epipole, epipolar line, baseline
- Epipolar line of parallel image planes

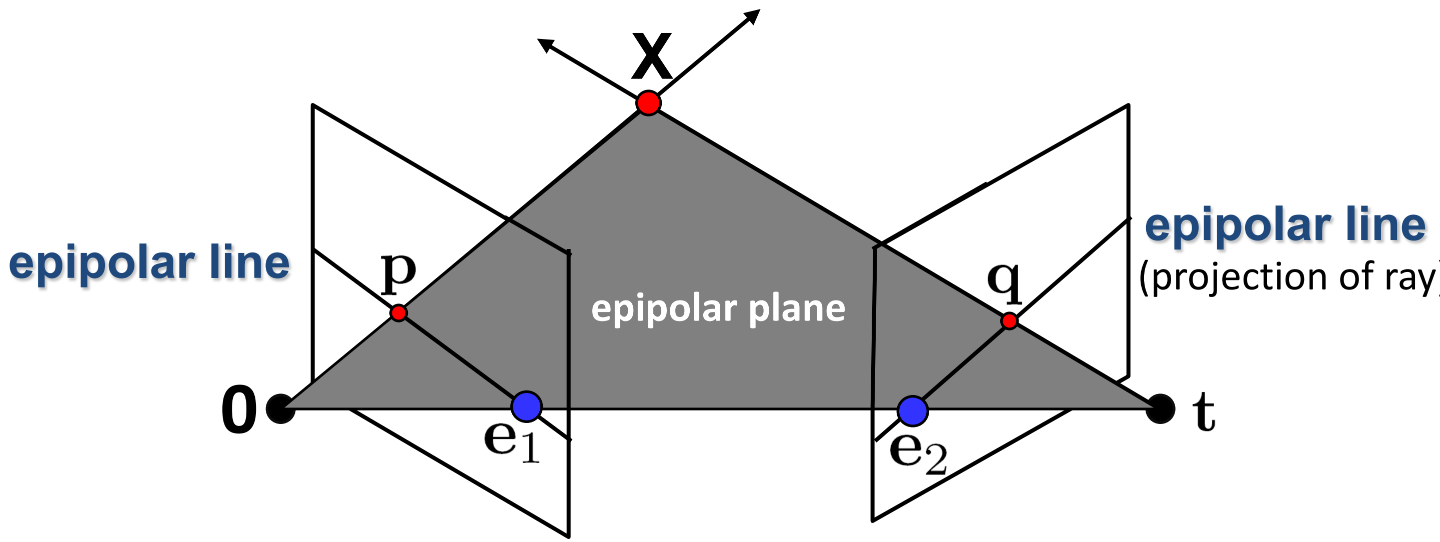
Cross product as matrix multiplication

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

$$[a_\times] = -[a_\times]^T$$

"skew-symmetric matrix"

Essential matrix

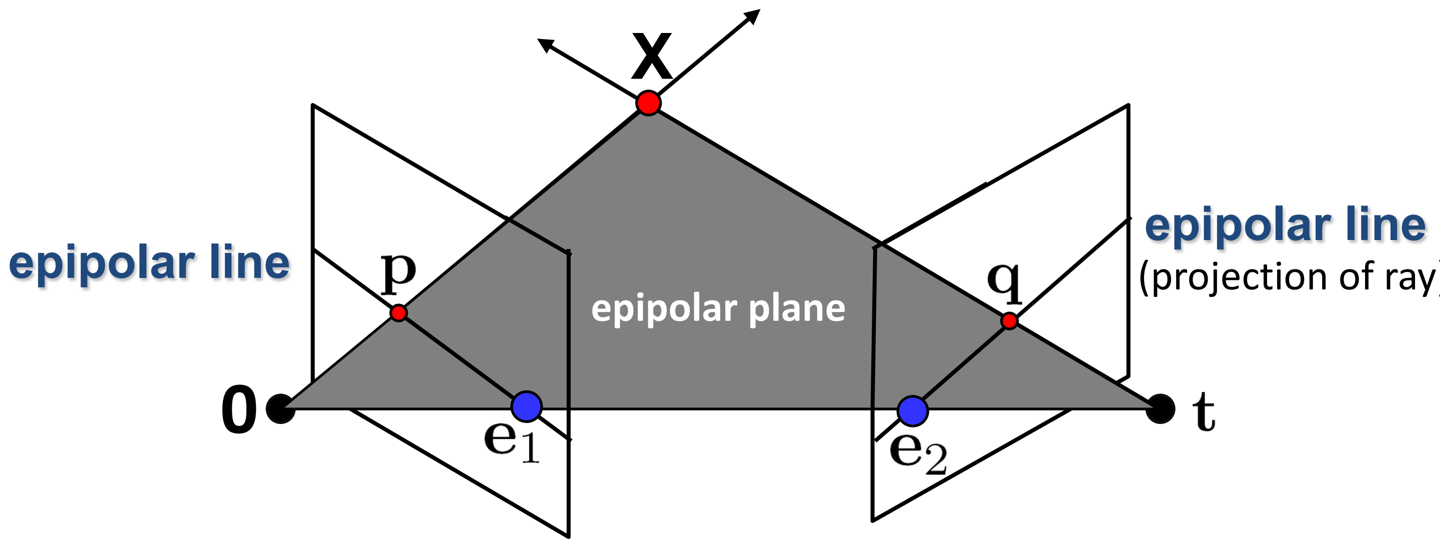


- Assume calibrated cameras with $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{I}_{3x3}$.
- Let camera 1 be [**I**, **0**] and camera 2 be [**R**, **t**].
- In camera 1 coordinates, 3D point **X** is given by $\mathbf{X}_1 = \lambda_1 \mathbf{p}$
- In camera 2 coordinates, 3D point **X** is given by $\mathbf{X}_2 = \lambda_2 \mathbf{q}$
- Since camera 2 is related to camera 1 by rigid-body motion [**R**, **t**]

$$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$$

$$\lambda_2 \mathbf{q} = \lambda_1 \mathbf{R}\mathbf{p} + \mathbf{t}$$

Essential matrix



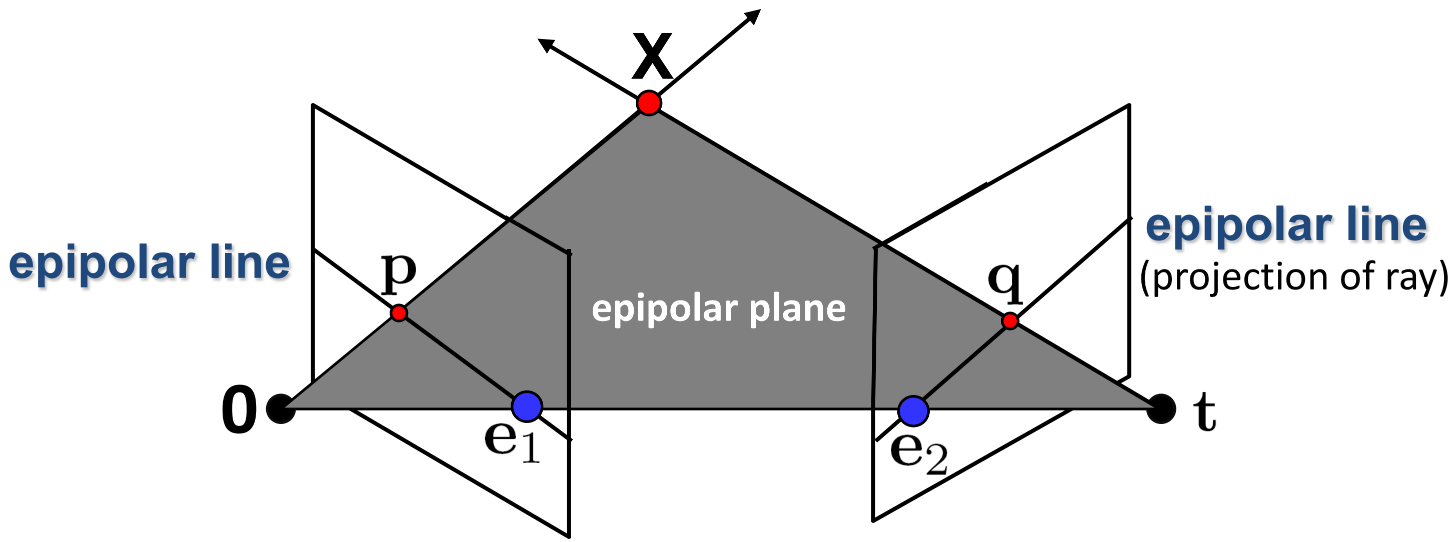
- We have: $\lambda_2 \mathbf{q} = \lambda_1 \mathbf{R}\mathbf{p} + \mathbf{t}$
- Take cross-product with respect to **t**:

$$\lambda_2 [\mathbf{t}]_\times \mathbf{q} = \lambda_1 [\mathbf{t}]_\times \mathbf{R}\mathbf{p}$$

- Take dot-product with respect to **q**:

$$0 = \lambda_1 \mathbf{q}^\top [\mathbf{t}]_\times \mathbf{R}\mathbf{p}$$

Essential matrix



- We have: $\lambda_2 \mathbf{q} = \lambda_1 \mathbf{R} \mathbf{p} + \mathbf{t}$
- Take cross-product with respect to **t**:

$$\lambda_2 [\mathbf{t}]_\times \mathbf{q} = \lambda_1 [\mathbf{t}]_\times \mathbf{R} \mathbf{p}$$

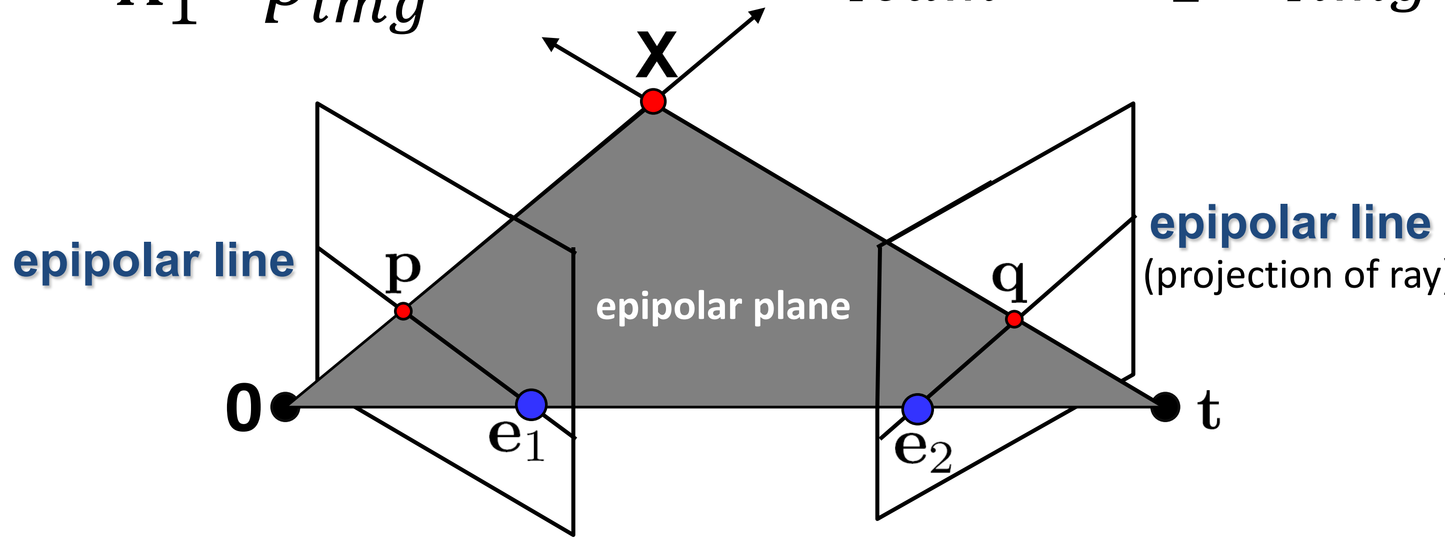
- Take dot-product with respect to **q**:

$$0 = \lambda_1 \mathbf{q}^\top \boxed{[\mathbf{t}]_\times \mathbf{R}} \mathbf{p}$$

E = Essential matrix

Essential matrix

$$p_{cam} = K_1^{-1} p_{img} \qquad q_{cam} = K_2^{-1} q_{img}$$



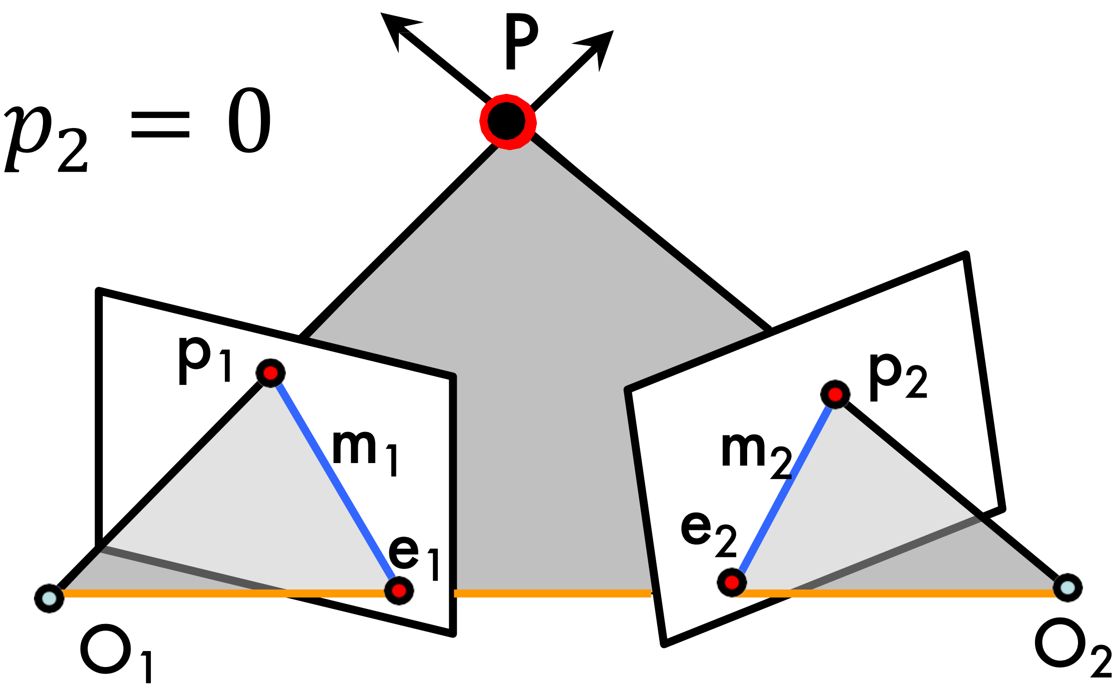
$$q_{cam}^T \, [t]_\times R \, p_{cam} = 0$$

$$q_{img}^T \, \boxed{K_2^{-1} [t]_\times R \, K_1^{-1}} \, p_{img} = 0$$

F = fundamental matrix

Epipolar Constraint

$$p_1^T \cdot F p_2 = 0$$



- $m_1 = F\ p_2$ is the epipolar line associated with p_2
- $m_2 = F^T p_1$ is the epipolar line associated with p_1
- $F\ e_2 = 0$ and $F^T e_1 = 0$
- F is 3x3 matrix; 7 DOF
- F is singular (rank two)

https://stackoverflow.com/questions/49763903/why-does-fundamental-matrix-have-7-degrees-of-freedom
(Won't ask you)

Estimate F: 8-eight algorithm

- Steps:
  - Each pair of correspondence is a linear equation
  - Formulate linear equation system from all pairs
  - Estimate F
    - Rank=8: A non-zero solution (with constraint $\|\mathbf{f}\| = 1$)
    - If N>8: least square
  - Project the estimation F to rank 2 matrix space
    - By SVD, the same as how we treat PCA as low-rank approximation
  - Need normalization
  - May need RANSAC to select points

Multi-view Reconstruction

- Two view: by triangulation
  - Need to first estimate F
  - From F we can infer E (need K1 and K2, from camera calibration, a process we did not cover in this course)
  - From E we can infer R and T (use SVD)
  - With R and T, we can use triangulation to estimate 3D positions of points

Optical Flow

- What is optical flow?
- Key assumptions
- Why we need these assumptions in our derivation?
- Optical flow equation
- How do we address the ambiguity?
- How would 2D transformations help us to address the ambiguity?
- The basic properties of each 2D transformation
- Can follow the derivation of the expanded KLT objective