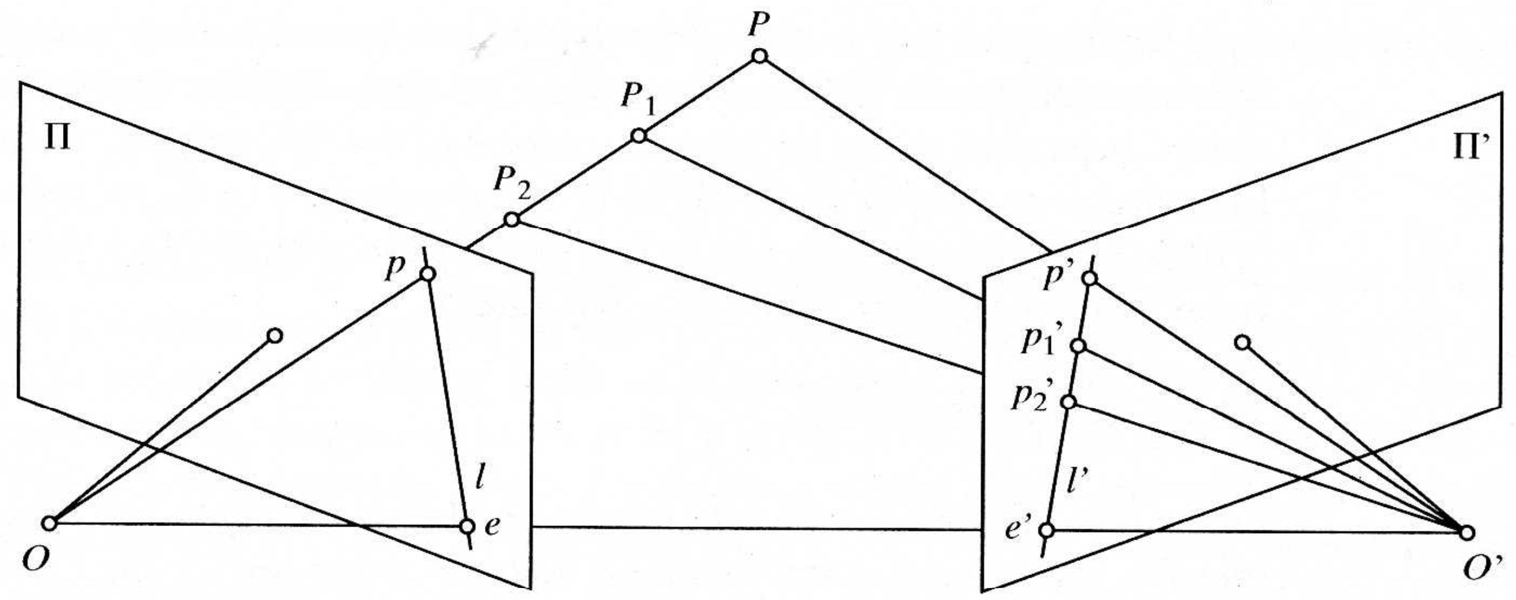
CS 231A Section 5: Midterm Review

Jiahui Shi
Yongwhan Lim
Kevin Tang
Hao Su

Important Concepts

- Epipolar geometry
- Essential and Fundamental Matrix
- 8-point algorithm
- RANSAC
- Hough transform
- Filter and edge detection
- K-Means

Epipolar Geometry



$\overrightarrow{Op}$, $\overrightarrow{O'p'}$, and $\overrightarrow{OO'}$ are coplanar.

- Epipolar lines, Epipolar plane, Epipoles, Baseline
- Epipolar constraint $p^T \cdot [T \times (R\, p')] = 0$

Essential and Fundamental Matrix

Skew symmetric matrix:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}_{\times}]\mathbf{b}$$

$$p^T \cdot [T_{\times}] \cdot Rp' = 0$$

$$E = [T_{\times}] \cdot R$$

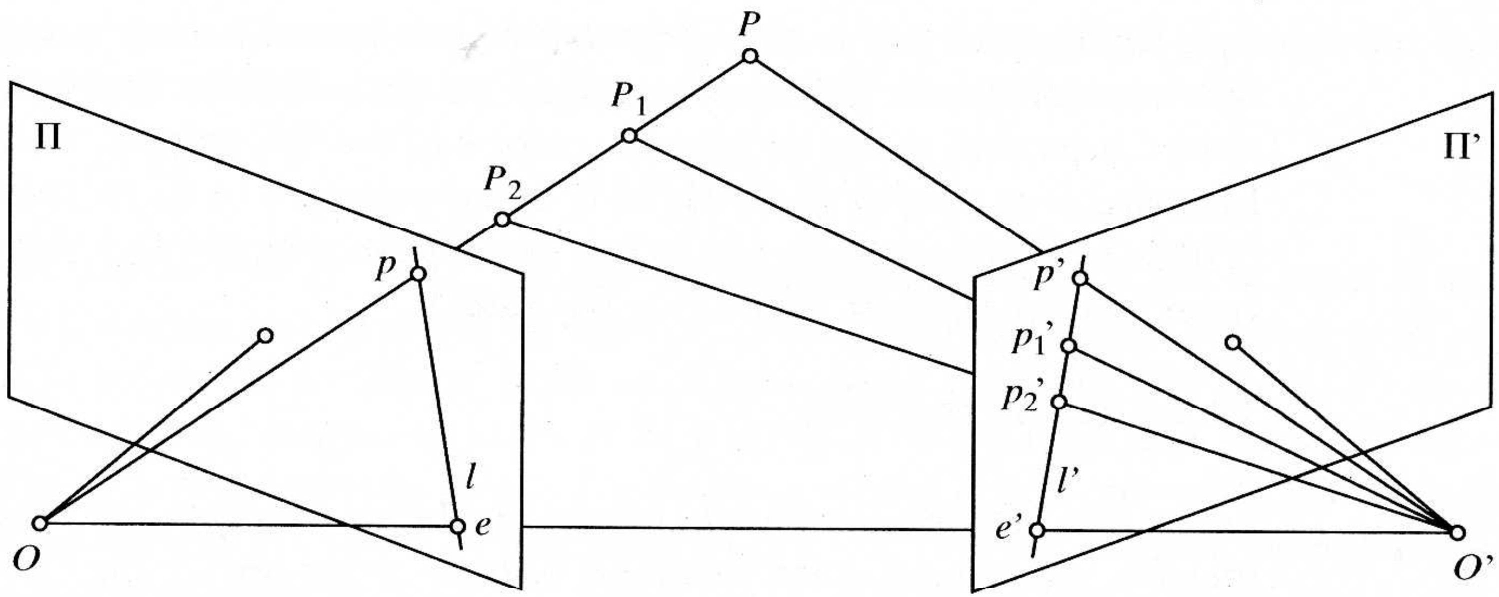
$$p^T E p' = 0$$

$$p^T \; K^{-T} \cdot [T_{\times}] \cdot R \; K'^{-1} \; p' = 0$$

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

$$\mathrm{p}^{\mathrm{T}} \; \mathrm{F} \; \mathrm{p}' = 0$$

A Simple “Trick”



- The fundamental matrix corresponding to a camera pair, $\mathrm{M} = [I\ 0]$ and $\mathrm{M}' = [A\ a]$ is equal to $[a]_x A$

8-point algorithm

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

$$\mathbf{W}\,\mathbf{f} = 0, \quad \|\mathbf{f}\| = 1$$

Given W, find least square solution f by SVD:

f is the left singular vector corresponding to the smallest singular value of W.

8-point Algorithm (normalized)

0. Compute T_i and T_i' (Origin = centroid of image points, Mean square distance of the data points from origin is 2 pixels)

1. Normalize coordinates: $q_i = T_i \, p_i \quad q_i' = T_i' \, p_i'$

2. Use the eight-point algorithm to compute F_q from the points q_i and q_i'

3. Enforce F_q to be rank-2: apply SVD F_q and set the smallest singular value to zero

4. De-normalize F_q: $F = T'^T F_q T$

RANSAC for Model Fitting

RANSAC loop:

1. Randomly select a minimum number of points for model fitting
2. Compute a model from these points
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of model on all of the inliers

- Keep the model with the largest number of inliers

Example: Fitting a plane

- Describe an algorithm that could be used to detect the orientation of a plane in the scene from scene points

RANSAC Pros/Cons

Pros:

- General method suited for a wide range of model fitting problems
- Easy to implement and easy to calculate its failure rate

Cons:

- Only handles a moderate percentage of outliers without cost blowing up
- Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)

Hough Transform

- A voting technique that can be used for model fitting problems
- Main idea:
  1. Record all possible models on which all given points belong to.
  2. Look for models that get many votes.

Hough Tranform Pros/Cons

Pros

- All points are processed independently, so can cope with occlusion
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin
- Can detect multiple instances of a model in a single pass

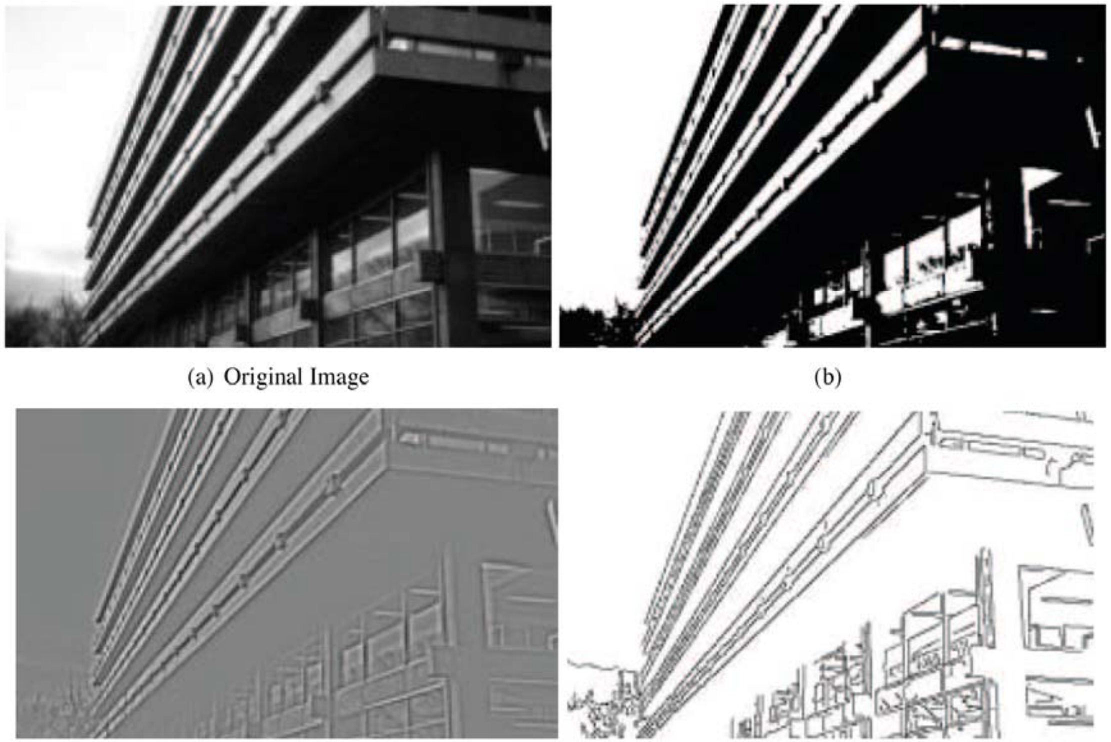
Cons

- Complexity of search time increases exponentially with the number of model parameters
- Non-target shapes can produce spurious peaks in parameter space
- Quantization: hard to pick a good grid size

Linear Filter

- Linear filtering:
  - Form a new image whose pixels are a weighted sum of original pixel values
  - Use the same set of weights at each point
- 1D linear filter and 2D linear filters
- Convolution
  - Properties: commutative, associative, distributive, shift, shift-invariance
- Cross-correlation

Filter and edge detection


(a) Original Image (b) (c) (d)

Possible Operations:

- Difference of Gaussians
- Threshold
- Box
- Fourier
- Inverse Fourier
- Canny
- Gaussian

K-Means

- Basic idea: randomly initialize the *k* cluster centers, and iterate between the two steps we just saw.
    1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
    2. Given cluster centers, determine points in each cluster
        - For each point p, find the closest c_i. Put p into cluster i
    3. Given points in each cluster, solve for c_i
        - Set c_i to be the mean of points in cluster i
    4. If c_i have changed, repeat Step 2

- Properties
    - Will always converge to *some* solution
    - Can be a "local minimum"
        - Does not always find the global minimum of objective function:

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

Slide credit: Steve Seitz

K-Means Pros/Cons

- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error

- Cons/issues
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters only
  - Assuming means can be computed