Feature Detection and RANSAC

Discussion #6

Written by:
Justin*, Chung Min*

Topics

This discussion covers feature detection and matching.

1 Harris Corner Detection

The Harris corner detector uses the structure tensor (also called the second moment matrix) to identify corners in images. Consider the following three 3×3 image windows:

Window A (Flat)

5	5	5
5	5	5
5	5	5

Window B (Edge)

1	1	9
1	1	9
1	1	9

Window C (Corner)

1	1	9
1	1	9
9	9	9

Corresponding gradients:

A's X gradients (I_x)

0	0	0
0	0	0
0	0	0

$\sum I_x^2$: ____ $\sum I_y^2$: ____

$\sum I_x I_y$: ____

B's X gradients

0	**8**	0
0	**8**	0
0	**8**	0

$\sum I_x^2$: ____ $\sum I_y^2$: ____

$\sum I_x I_y$: ____

C's X gradients

0	**8**	0
0	**8**	0
0	0	0

$\sum I_x^2$: ____ $\sum I_y^2$: ____

$\sum I_x I_y$: ____

A's Y gradients (I_y)

0	0	0
0	0	0
0	0	0

B's Y gradients

0	0	0
0	0	0
0	0	0

C's Y gradients

0	0	0
8	**8**	0
0	0	0

Problem 1.1: ***Structure Tensor Computation***
M is defined as:

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

where the sums are taken over the local window.
Compute the structure tensor M for each of the three windows A, B, and C.

Window A (Flat)

$$M_A = \begin{bmatrix} \_\_\_\_ & \_\_\_\_ \\ \_\_\_\_ & \_\_\_\_ \end{bmatrix}$$

Window B (Edge)

$$M_B = \begin{bmatrix} \_\_\_\_ & \_\_\_\_ \\ \_\_\_\_ & \_\_\_\_ \end{bmatrix}$$

Window C (Corner)

$$M_C = \begin{bmatrix} \_\_\_\_ & \_\_\_\_ \\ \_\_\_\_ & \_\_\_\_ \end{bmatrix}$$

Problem 1.2: ***Determinant and Trace***
For each window, compute the determinant and trace.

- $\det(M) = M_{11} \cdot M_{22} - M_{12}^2$
- $\text{trace}(M) = M_{11} + M_{22}$

Problem 1.3: ***Harris Response***
One common Harris corner response is defined as:

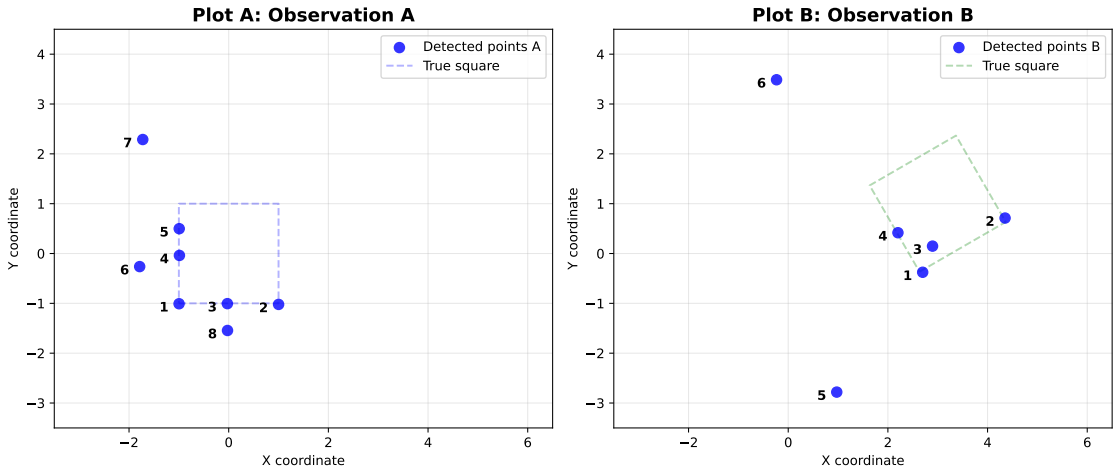
$$R = \frac{\det(M)}{\text{trace}(M)}$$

What is this for each window above?

Problem 1.4: ***Thresholding Rationale***
In practice, we often threshold using a high ratio of $\det(M)$ to $\text{trace}(M)$. Explain why this ratio is meaningful for corner detection.

2 RANSAC for Robust Point-cloud fitting

Suppose we have an object (a square) detected in two different 2D point clouds. We detected some point correspondences between the two pointclouds, but our feature matcher also gave us lots of junk. Our goal is to estimate the rigid transformation describing how this square moved despite the junk. In the figure below, detected matches are indicated by the numbers next to the points.



Problem 2.1: ***What model?*** How many degrees of freedom does the transform we're trying to fit have? How many point correspondences do we need to fully estimate it?

Problem 2.2: ***RANSAC Pseudocode***

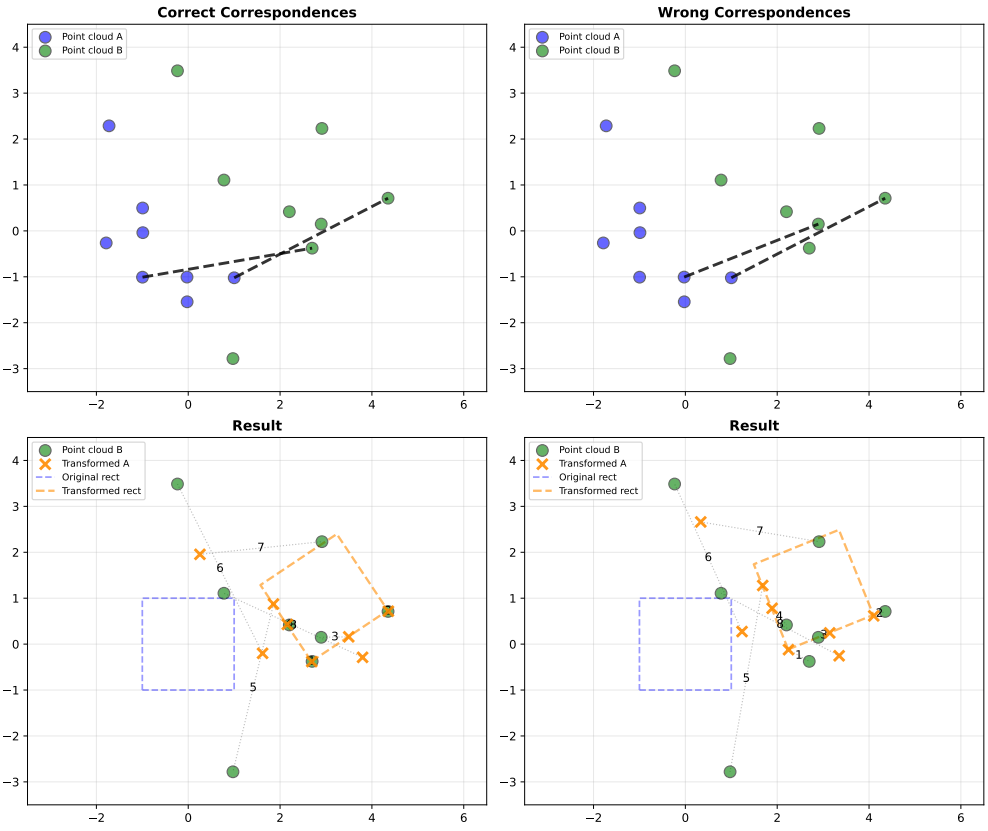
Last week, we learned we can use least-squares to estimate a transform from correspondences. Suppose I give you this oracle function `T = get_T(point_correspondences)` along with `new_points = transform_points(T, points)`. Write pseudocode for how to use these functions inside the RANSAC algorithm. (assume any helper functions you want for this)

Problem 2.3: ***Analyzing RANSAC***

For the given correspondences in Plot A and B, what is the probability we get the correct model from one iteration of RANSAC? (8 total correspondences, 3 correct and 5 wrong)

Problem 2.4: ***Success Probability***
What is the probability we get the correct model after running RANSAC for N iterations?

Problem 2.5: ***Counting Inliers*** Illustrated below are two example iterations of RANSAC, one correct and one wrong pair chosen. The bottom row shows the result of using those correspondences to estimate the transform, and applying it to point cloud A.



For each scenario, how many inliers does the estimated model have? (use the radius of the green dot as inlier radius).