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

B's X gradients

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

C's X gradients

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

A's Y gradients (I_y)

0	0	0
0	0	0
0	0	0

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

$\sum I_x I_y$: ______

B's Y gradients

0	0	0
0	0	0
0	0	0

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

$\sum I_x I_y$: ______

C's Y gradients

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

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

$\sum I_x I_y$: ______

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)	**Window B (Edge)**	**Window C (Corner)**
$M_A = \begin{bmatrix} \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ \end{bmatrix}$	$M_B = \begin{bmatrix} \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ \end{bmatrix}$	$M_C = \begin{bmatrix} \_\_\_ & \_\_\_ \\ \_\_\_ & \_\_\_ \end{bmatrix}$

Window A (Flat):

$$M_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Window B (Edge):

$$M_B = \begin{bmatrix} 192 & 0 \\ 0 & 0 \end{bmatrix}$$

(Note: $\sum I_x^2 = 8^2 + 8^2 + 8^2 = 192$)
Window C (Corner):

$$M_C = \begin{bmatrix} 128 & 64 \\ 64 & 128 \end{bmatrix}$$

(Note: $\sum I_x^2 = 8^2 + 8^2 = 128$, $\sum I_y^2 = 8^2 + 8^2 = 128$, $\sum I_x I_y = 8 \times 8 = 64$)

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

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

Window A: $\det(M_A) = 0$, $\text{trace}(M_A) = 0$
Window B: $\det(M_B) = 192 \cdot 0 - 0^2 = 0$, $\text{trace}(M_B) = 192$
Window C: $\det(M_C) = 128 \cdot 128 - 64^2 = 12288$, $\text{trace}(M_C) = 256$

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?

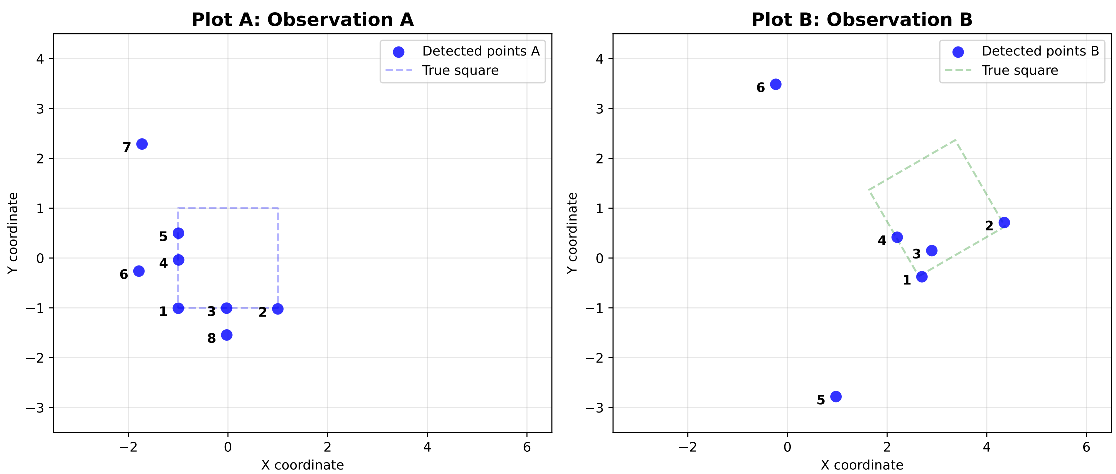
Window A: $R_A = \frac{0}{0}$ (undefined/0)
Window B: $R_B = \frac{0}{192} = 0$
Window C: $R_C = \frac{12288}{256} = 48$

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.

The ratio $\frac{\det(M)}{\text{trace}(M)}$ is meaningful because the determinant captures the product of the eigenvalues $\lambda_1\lambda_2$ while the trace captures their sum $\lambda_1 + \lambda_2$.
For **corners**, both eigenvalues are large (gradients in multiple directions), giving a large determinant and thus a high ratio.
For **edges**, one eigenvalue is large and one is small (gradient in only one direction), making the determinant small and the ratio low.
For **flat regions**, both eigenvalues are small, giving near-zero determinant and ratio.
Thus, a high ratio reliably identifies corners where gradients are significant in all directions.

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?

A 2D rigid transformation has **3 degrees of freedom**: 2 for translation (t_x, t_y) and 1 for rotation (θ).
We need **2 point correspondences** to fully estimate it (each point gives 2 equations for x and y coordinates, giving us 4 equations for 3 unknowns).

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)

```
best_T = None
best_inlier_count = 0

for iteration in range(N):
    sample = randomly_sample_2_correspondences(all_correspondences)

    # Fit model to sample
    T = get_T(sample)

    # Count inliers using the fitted model
    inlier_count = count_inliers(T, all_correspondences, threshold)

    # Keep best model
    if inlier_count > best_inlier_count:
        best_inlier_count = inlier_count
        best_T = T

return best_T

# Afterwards, *refit* model to all inliers: final_T = get_T(inliers)
```

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)

From the figure, there are 3 correct correspondences and 5 incorrect ones (8 total).
We need to sample 2 correspondences. The probability that both are correct is:

$$p = \frac{\binom{3}{2}}{\binom{8}{2}} = \frac{3}{28} \approx 0.107$$

This is the probability of randomly selecting 2 points from the 3 correct matches out of all possible pairs.

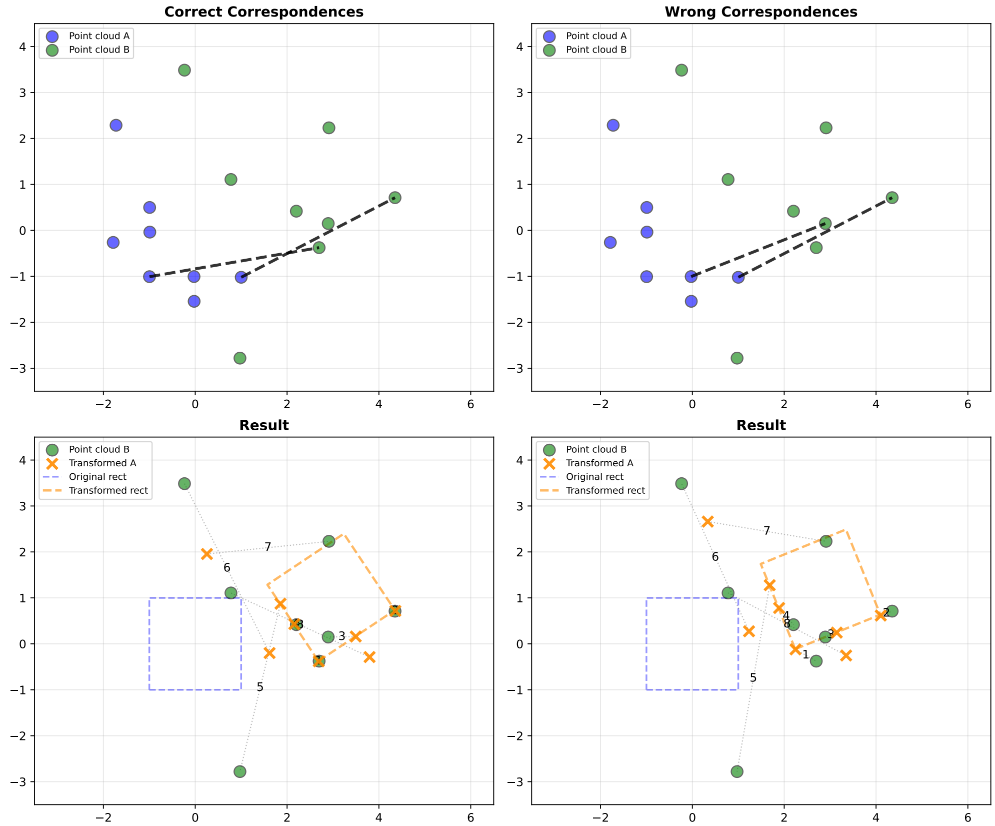
Problem 2.4: ***Success Probability***
What is the probability we get the correct model after running RANSAC for N iterations?

From 2.3, $p = \frac{3}{28}$ is the probability of success on one iteration.
The probability of failing all N iterations is $\left(1 - \frac{3}{28}\right)^N = \left(\frac{25}{28}\right)^N$.
Therefore, the probability of success in at least one iteration is:

$$P(\text{success}) = 1 - \left(\frac{25}{28}\right)^N$$

Examples: With $N = 20$: $P(\text{success}) \approx 0.52$ (52%). With $N = 100$: $P(\text{success}) \approx 0.9997$ (99.97%).

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).

Left: 3 inliers, right: 0 inliers. Note how even with very few correct correspondences, RANSAC can still find the correct model!