Directed Studies: Assignment 4

Navid Zarrabi
navid.zarrabi@torontomu.ca

Toronto Metropolitan University — June 16, 2023

1

The objective of this course is to learn how to distinguish inliers from outliers in statistical inference. For more information, refer to 'Ch 5.3 Handling Outliers' of your textbook.

Given the source and destination 2d points below, compute a homography that maps the source points to the destination. a minimum of 10 inliers needed to be detected by the RANSAC algorithm. The re-projection error of the inliers need to be less than 0.005.

To learn about homography, see

https://www.cs.toronto.edu/ lindell/teaching/420/slides/lecture8.pdf

or read 'Multiple View Geometry in Computer Vision' textbook.

Report your normalized homography transformation. Write your code in Python, do not use OpenCV. Plot a scatter plot, showing inliers (marked by o) and outliers (marked by x), where inliers of source and destination are connected via a line. Include the plot in your report. You should only use the following modules:

import numpy as np
import random
import matplotlib.pyplot as plt

RANSAC (Random Sample Consensus) is an iterative algorithm used for robust estimation in the presence of outliers [1]. It is commonly used in computer vision and other fields to solve problems such as point cloud registration and image feature matching [2]. RANSAC randomly selects a minimal sample from the data, fits a model to that sample, and then determines the number of inliers that are consistent with the model within a certain threshold. This process is repeated multiple times, and the model with the largest number of inliers is considered the best fit to the data. RANSAC is effective at handling data with outliers and can provide reliable estimations in the presence of noise.

1.1 Solution

First, we randomly sample four points from the source and destination point sets using sample function from random library. Then, we need to compute the homography matrix using the sampled points. This matrix (named A) is $2n \times 9$ in dimension where n is the number of points.

$$
\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1}
$$

A Singular Value Decomposition is needed to extract right singular vector (called V) from the A matrix. The smallest singular vector (or the last row of matrix V) is stored as $v_{\hat{i}}$. This $v_{\hat{i}}$ is then reshaped to get the homorgraphy transformation H. Using the computed homography, all other points are transformed to a predicted destination. Euclidean distance between the transformed points and the Destination points are calculated to generate an error matrix. If the error is below the 0.005 treshold, then the point can be called an inlier. Otherwise, it is an outlier.
The above process is repeated until we get enough number of inliers and make sure the transformation is correct. The iteration with largest number of inliers is chosen as the best transformation.

1.2 Implementation

In order to see and visualize data, first source and destination points are plotted as shown in Figure 1. Red asterisk shows the source, blue asterisk depicts destination, and the dashed line is the transformation between these points. Most of the red points are mapped into a line of blue points and the noisy transformations is obvious in this example. Now, we will use RANSAC and Homography to eliminate these outliers and their corresponding transformations.
Three main functions are defined. First one looks for inliers using algorithm explained above. Second function calculates Homography matrix and returns it. Third function plots the inliers, outliers, and transformations for inliers as shown in Figure 2. As we see in Figure 3, the red transformations are eliminated.

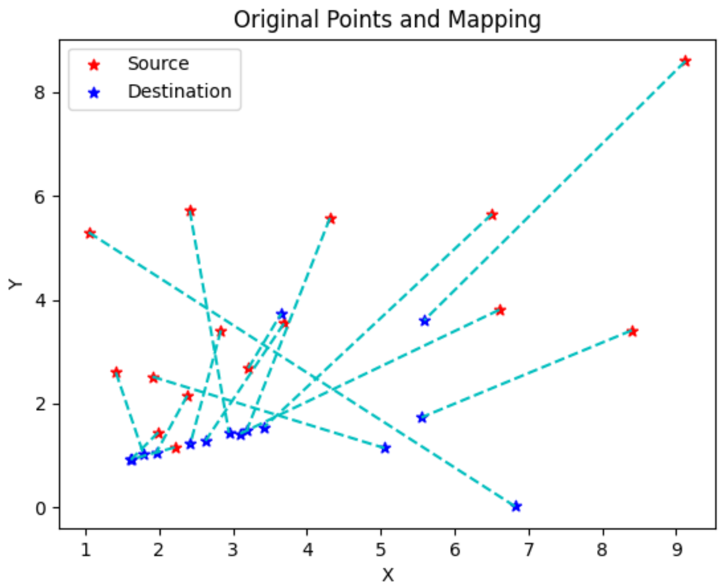

Figure 1: Source, Destination, and Transformations

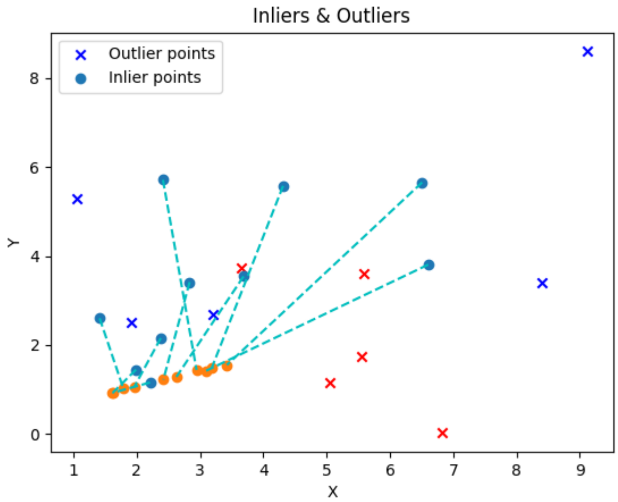

Figure 2: Inliers, outliers, and inlier transformations

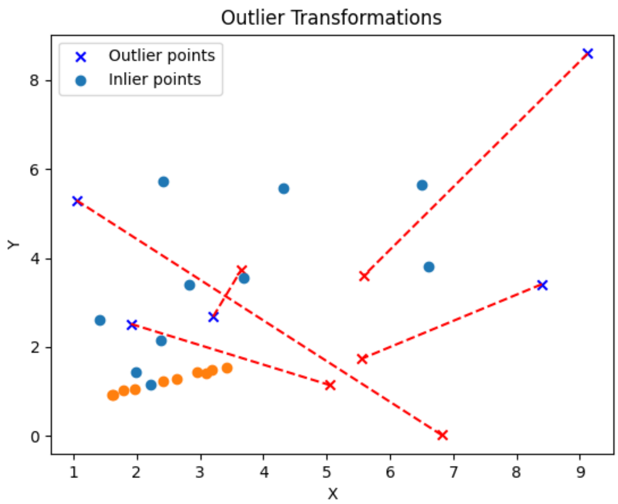

Figure 3: Inliers, outliers, and outlier transformations

References

[1] Barfoot, Timothy D. State estimation for robotics. Cambridge University Press, 2017.

[2] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.