ECE661 Fall 2024: Homework 5

By – Aayush Bhadani

abhadani@purdue.edu

The goal is to implement a fully automated approach for robust homography estimation and use it to generate a panoramic view by stitching together overlapping views of the same scene.

We start with SIFT detector with a Brute-Force matcher which gives us key points and descriptor vectors. Next, we use RANSAC algorithm for outlier rejection and linear least squares for the initial homography estimate. We subsequently refine this homography estimates using the LM method. After we have the homography matrices we project all the views to a common reference frame that is the middle image. Finally, we stitch together all these projections to form a panoramic view of the scene.

2. Theory Questions

1. Conceptually speaking, how do we differentiate between the inliers and the outliers when using RANSAC for solving the homography estimation problem using the interest points extracted from two different photos of the same scene?

In RANSAC we use a decision threshold 'delta' to differentiate inliers and outliers. We compare estimate with the data point and if the distance between them is within the threshold limit (delta) we take it as an inlier. We start by constructing an initial homography estimate with a few points using which we transform points of image1 to image2 space. We then compare the distance between points in image2 with those transformed to image2 space. If the distance is less than the threshold (delta), we assign it as an inlier or else as an outlier. This process is repeated a number of times until we have an inlier set of desired size without outliers. Finally, we use the entire inlier set for calculating the estimate.

2. As you will see in Lecture 13, the Gradient-Descent (GD) is a reliable method for minimizing a cost function, but it can be excruciatingly slow. At the other extreme, we have the much faster Gauss-Newton (GN) method but it can be numerically unstable. Explain in your own words how the Levenberg-Marquardt (LM) algorithm combines the best of GD and GN to give us a method that is reasonably fast and numerically stable at the same time.

LM algorithm combines the best of GD and GN methods with the help of damping coefficient that adjusts the balance between GD and GN. When we are far from the solution, the damping coefficient is large and LM algorithm behaves more like GD thus ensuring stability & safer convergence guarantee. However, when we get closer to the solution, the algorithm behaves more like GN with a smaller damping coefficient. This results in larger, more confident steps which speed up convergence. This adaptability gives us a fast and numerically stable method for minimizing the cost function.

3.

3.1 Task 1

Linear Least Square:

For a single pair of correspondences, we get the following equation

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \end{bmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x_1' \\ y_1' \end{pmatrix}$$

This is of the form $Ax = b$. When we have 4 pairs of correspondences then matrix A is square and we can simply take the inverse of A to find the unknown vector consisting of the elements of Homography matrix. However, when we make use of more than 4 correspondences we have an over-determined system. To solve for linear least square estimate we take pseudo-inverse here which is given by: $(A^TA)^{-1}A^Tb$.

We finally construct homography matrix as $H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$

RANSAC:

Over a number of trails, the RANSAC algorithm uses some randomly selected data points to construct a homography estimate and then evaluates how well the remaining data supports this estimate. The estimate is accepted when the support exceeds a threshold. The supporting data for an accepted estimate form the inliers and the rest become outliers. Finally, we use the inlier set for refining the estimate.

The following values were used:

$\sigma = 2$

$\delta = 3\sigma$

$\epsilon = 0.3$

$p = 0.99$

$n = 6$

$N = \dfrac{\ln(1-p)}{\ln(1-(1-\epsilon)^n)}$

$M = (1-\epsilon)n_{total}$

$(n_{total}\ is\ total\ numer\ of\ correspondences)$

The Levenberg-Marquardt (LM) method:

It is a very efficient iterative optimization algorithm used for solving non-linear least squares problems. It combines the best of two methods: Gradient Descent and Gauss-Newton, adapting between them to offer a balance of speed and stability. When the solution is far, LM behaves more like Gradient Descent, taking cautious steps to avoid instability. As the solution approaches the minimum, it transitions to the faster Gauss-Newton method, making larger, more confident steps. This dynamic adjustment allows LM to converge quickly while remaining robust to issues like poor initial guesses or ill-conditioned problems. We can use it from scipy.optimize.least_squares.

Set of given 5 image for performing the task:

Image1:

Image2:

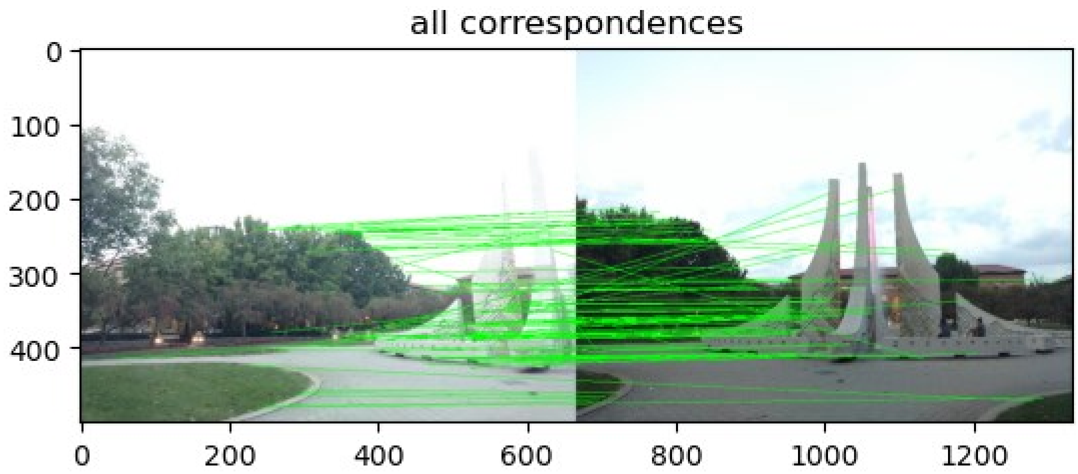
Image3:

Image4:

Image5:

<u>Correspondences:</u>

Image1 & Image2:



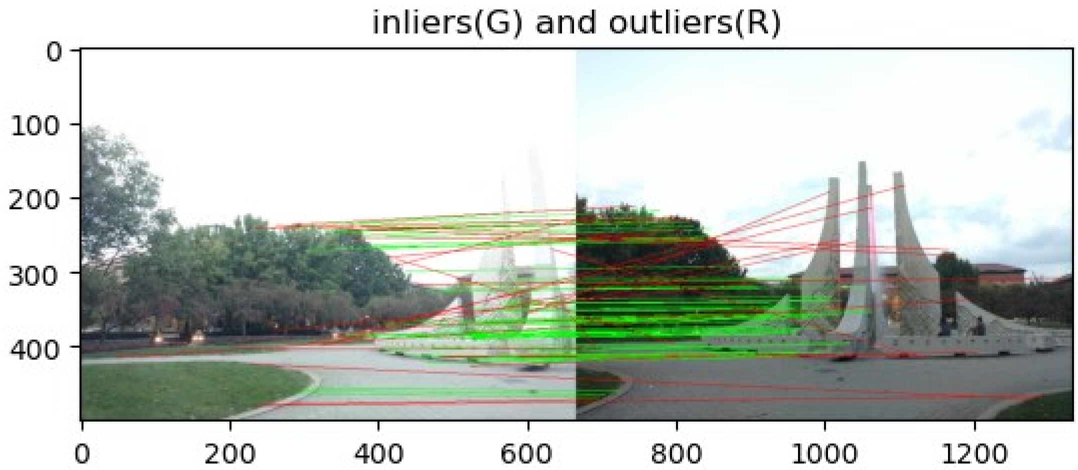
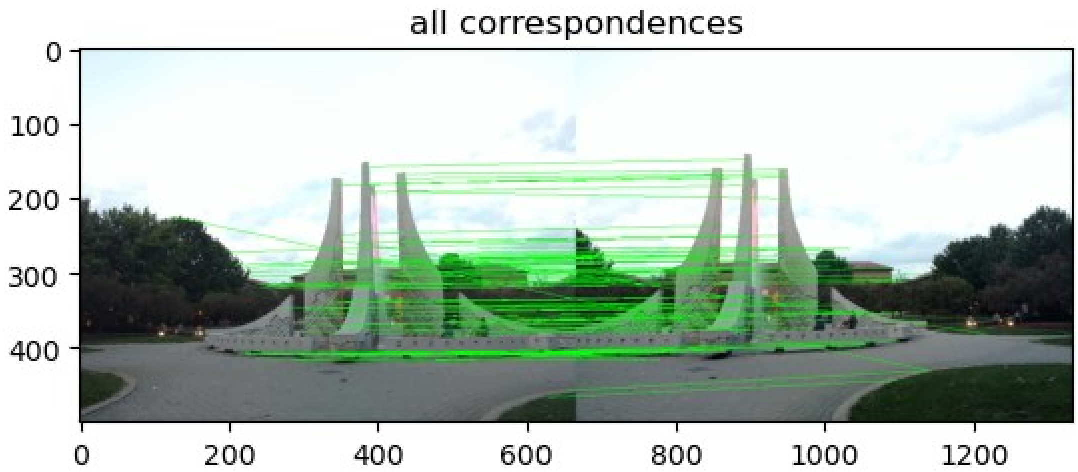


Image2 & Image3:



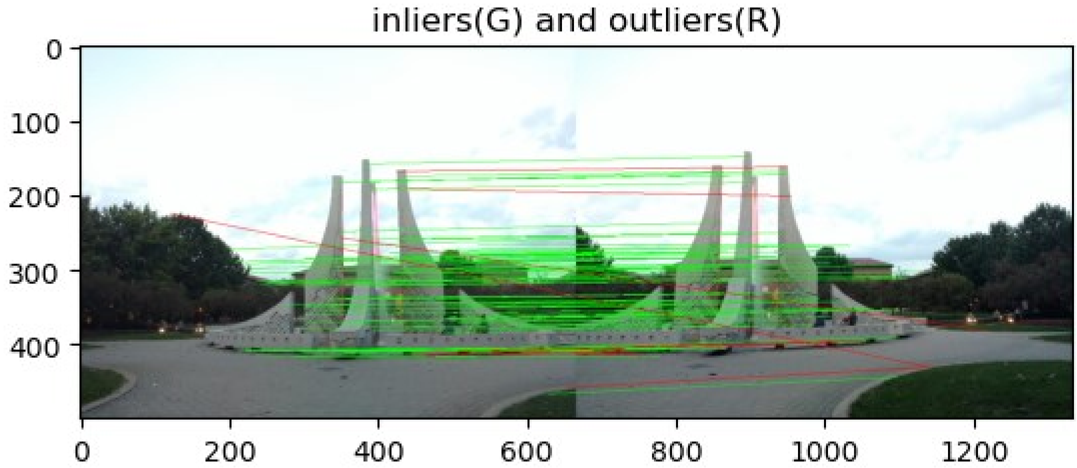
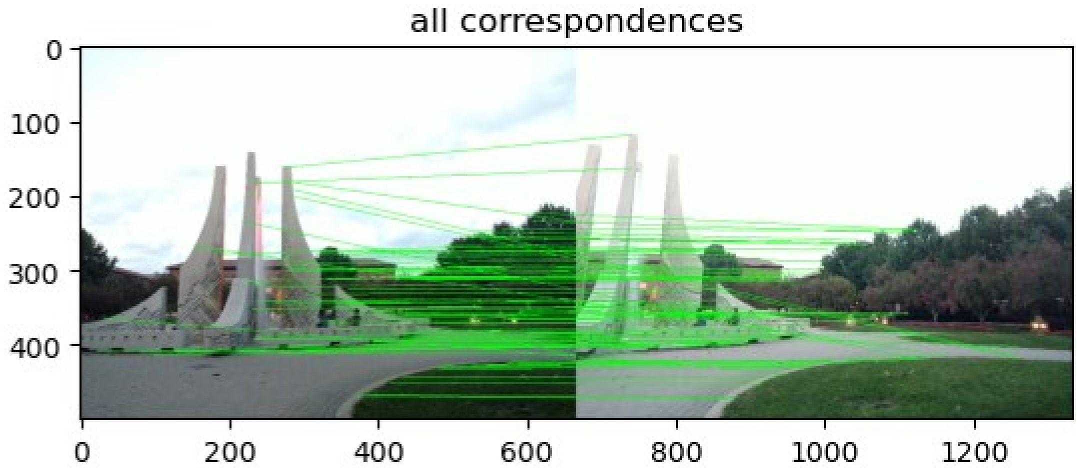


Image3 & Image4:



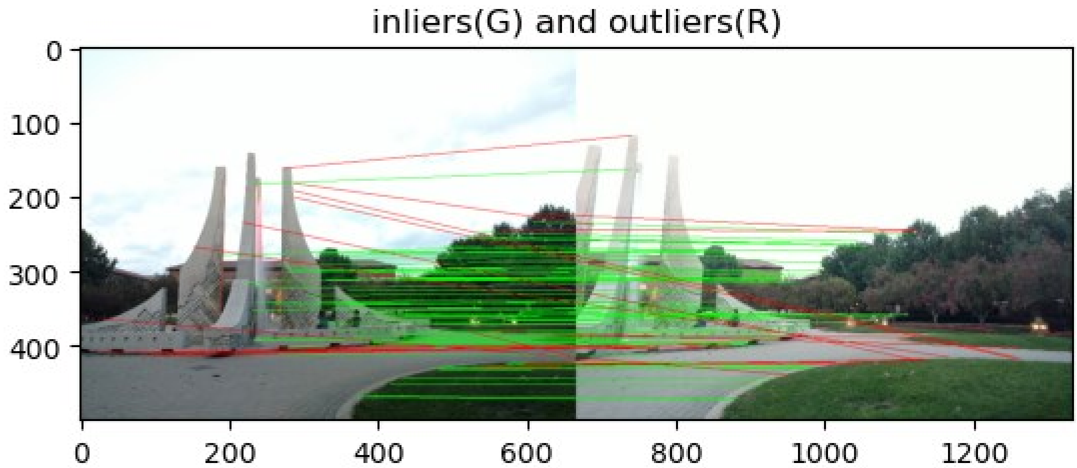
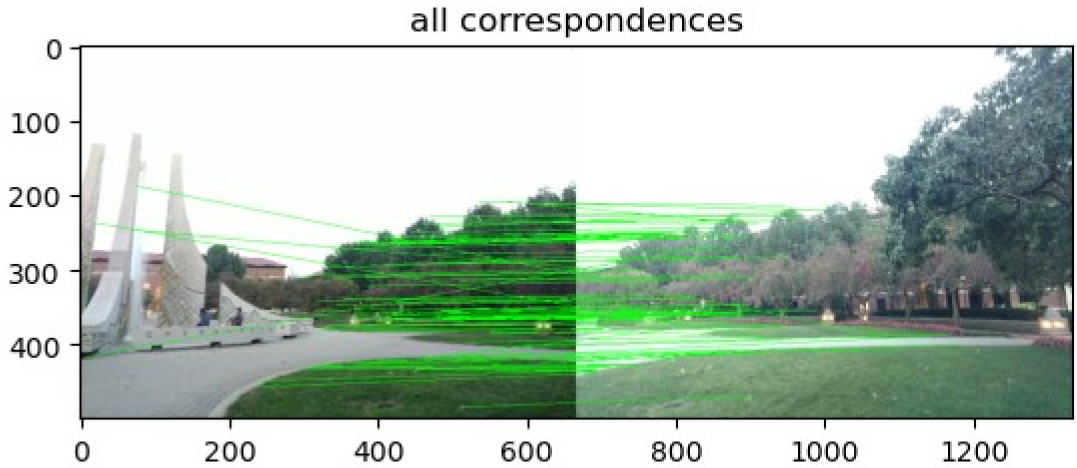
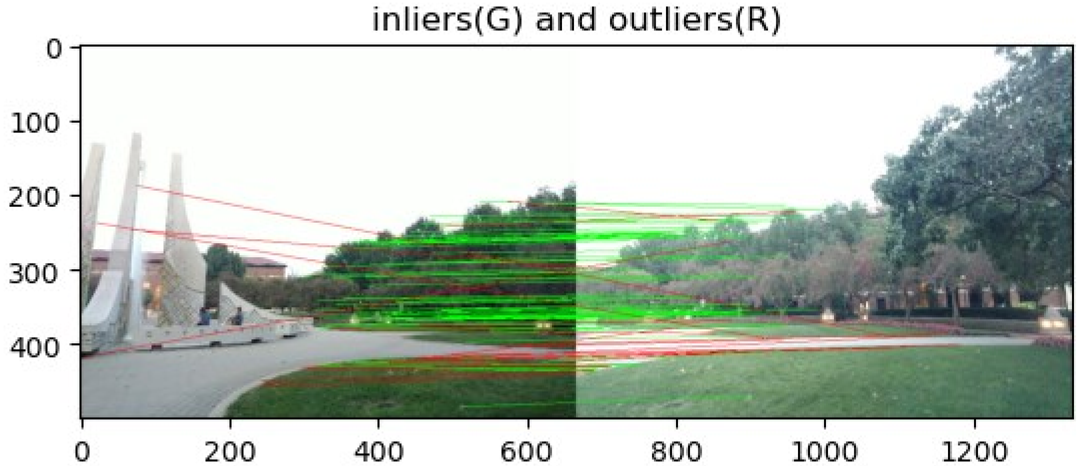
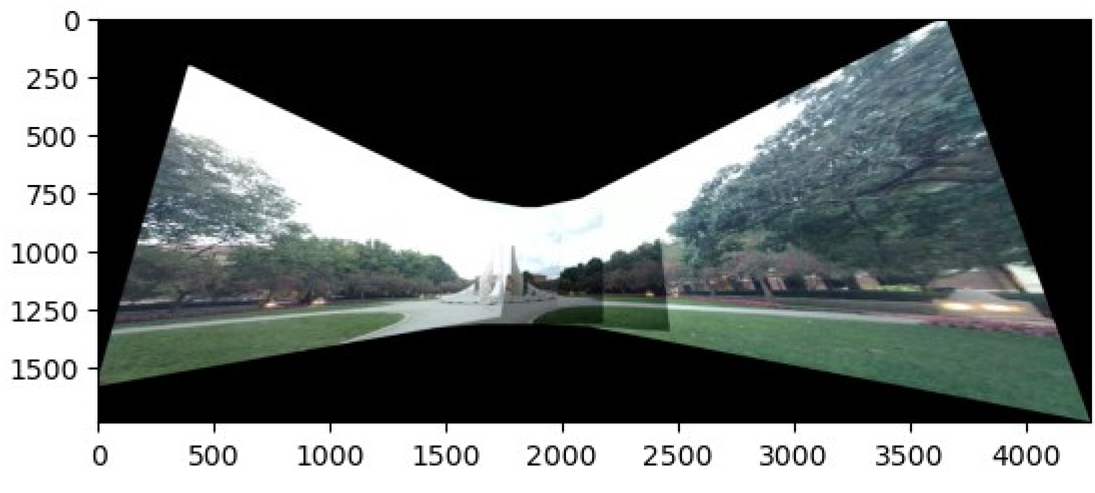


Image4 & Image5:





Panoramic view:



3.2 Task2 : Other set of images used for the task:

Image1:

Image2:

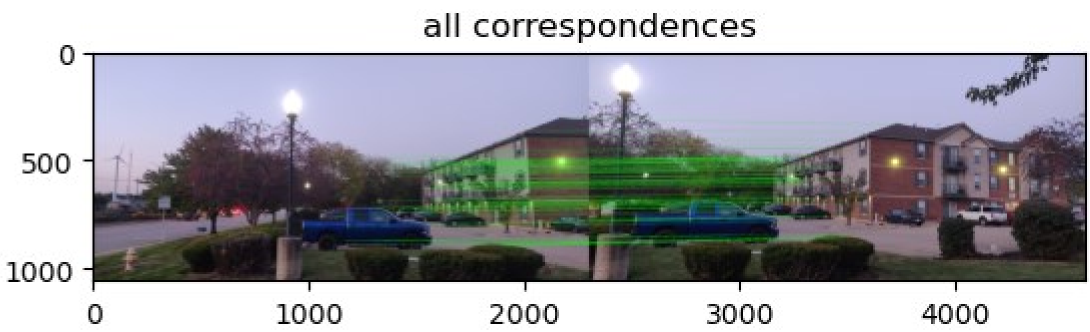
Image3:

Image4:

Image5:

<u>Correspondences</u>

Image1 & Image2:



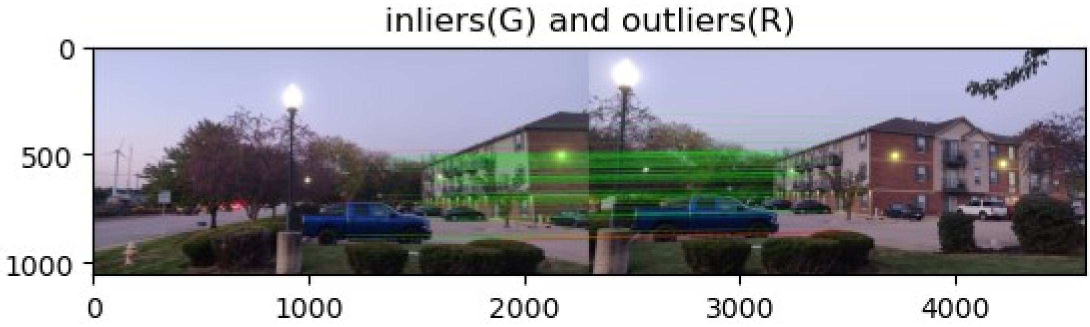
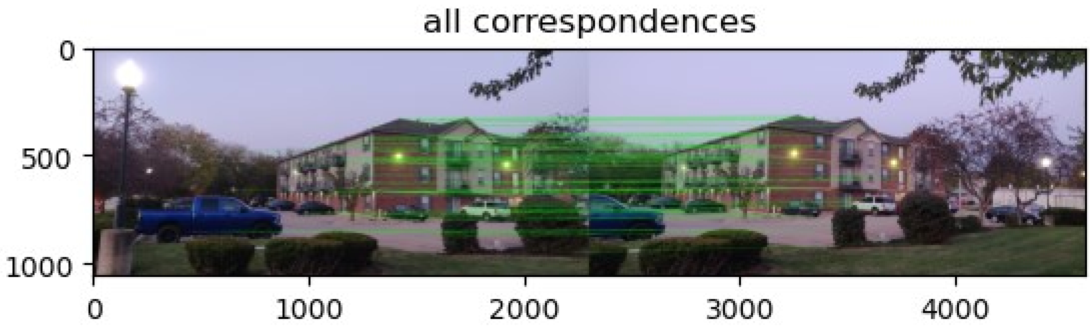


Image2 & Image3:



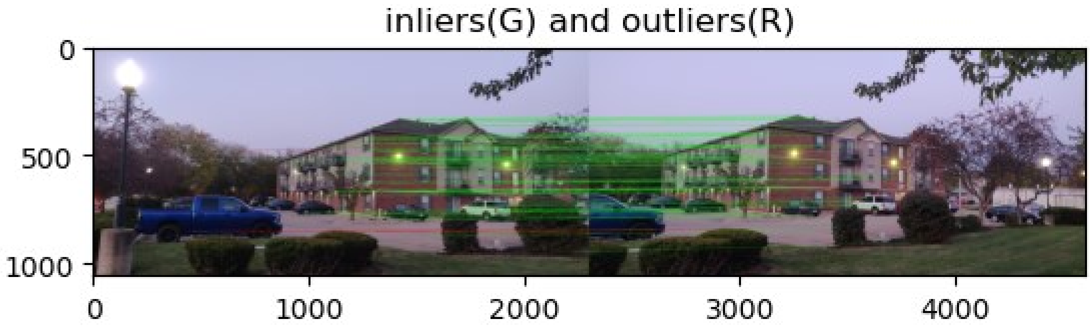
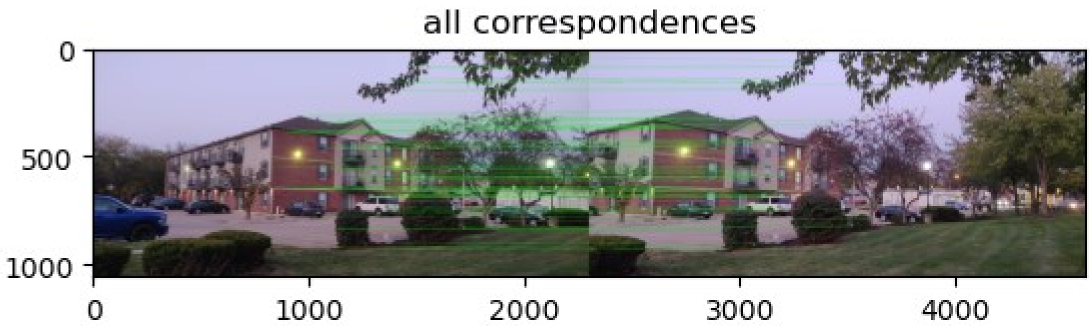


Image3 & Image4:



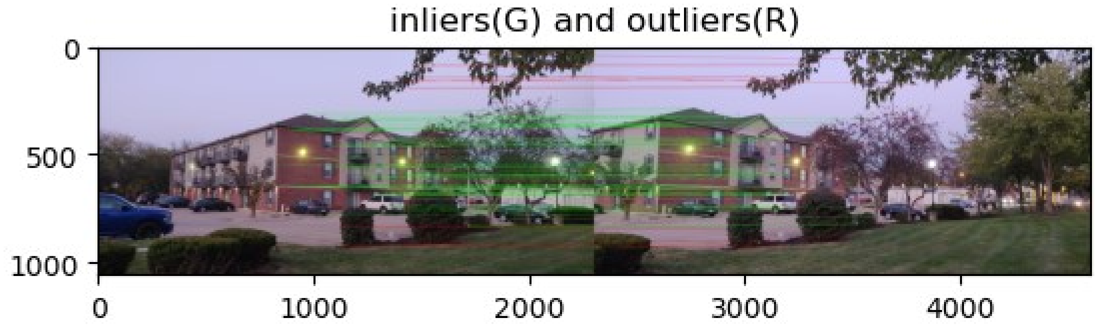
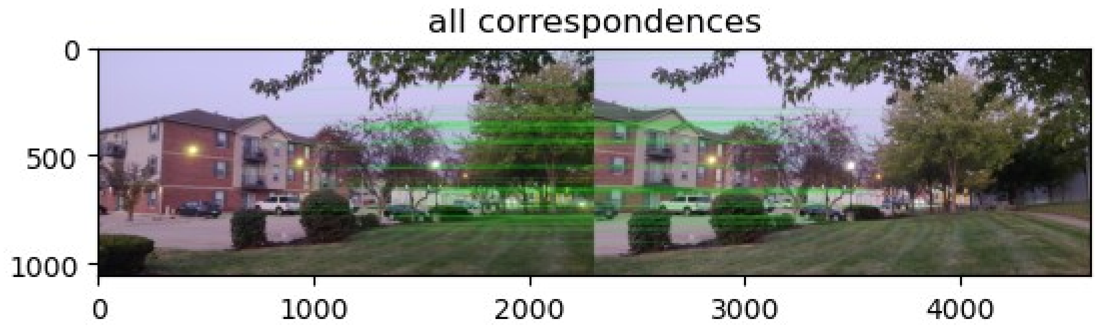


Image4 & Image5:



<u>Panoramic view</u>