Department of Computer Science

CSC2503H: Foundations of Computer Vision

Assignment #3:

Fundamental Matrix & Homography Estimation

Sanjif Rajaratnam
999091986
rajara24

Course Instructor: Dr. Kyros Kutulakos

December 7th, 2016

Table of Contents

Table of Figures

1.0 Fundamental Matrix Estimation

1.1 Ground-Truth Calculation

Let the subscript L represent the left image, and the subscript R represent the right image. Let the projection matrices for the left and right image be represented by $\boldsymbol{\Pi_L}$ and $\boldsymbol{\Pi_R}$, respectively. The projection matrices can be found via **Equation 1**.

$$\boldsymbol{\Pi_\mu} = (\boldsymbol{M_{int}} \cdot \boldsymbol{M_{ext}})^{\mu} \quad where\ \mu = L, R \qquad (1)$$

The equation for the fundamental matrix ($\boldsymbol{F_0}$) is derived in the class notes. The final equation, from the notes, can be seen in **Equation 2**. The subscripts were changed ($2 \rightarrow L, 1 \rightarrow R$) to match the given notation. This method was chosen because the fundamental matrix ($\boldsymbol{F_1}$) found using RANSAC used the same order of variables.

$$x_L^T \boldsymbol{F} x_R = 0 \ \rightarrow \ \boldsymbol{F} = [\boldsymbol{\Pi_L} m]_x \boldsymbol{\Pi_L} \boldsymbol{\Pi_R^T} \left(\boldsymbol{\Pi_R} \boldsymbol{\Pi_R^T}\right)^{-1} \qquad (2)$$

The $[\boldsymbol{\Pi_L} m]_x$ is a cross-product matrix of the $\boldsymbol{\Pi_L} m$ vector. The m vector is the vector in the nullspace of $\boldsymbol{\Pi_R}$. It was found using the null function within Matlab. The $[\boldsymbol{\Pi_L} m]_x$ cross product matrix was then found using **Equation 3**.

$$[\boldsymbol{\Pi_2} m]_x = \begin{bmatrix} 0 & -(\boldsymbol{\Pi_2} m)[3] & (\boldsymbol{\Pi_2} m)[2] \\ (\boldsymbol{\Pi_2} m)[3] & 0 & -(\boldsymbol{\Pi_2} m)[1] \\ -(\boldsymbol{\Pi_2} m)[2] & (\boldsymbol{\Pi_2} m)[1] & 0 \end{bmatrix} \qquad (3)$$

The ground truth matrix ($\boldsymbol{F_0}$) can then be found by solving for $\boldsymbol{\Pi_L}, \boldsymbol{\Pi_R}$, and $[\boldsymbol{\Pi_L} m]_x$ and plugging these matrices into **Equation 2**. The ground truth fundamental matrix is computed in the 'findFGroundTruth.m' function.

1.2 Epipolar Line Calculation and Error Calculation

Firstly, a subset of regularly spaced points was chosen in the left image. The region was selected due to the high density of features. The region chosen was defined by $x \in [-70, -40]$

and $y \in [0, 30]$. The epipolar line in the left image was found for each feature point projection in the right image that was within the specified region. The epipolar line in the left image was found using both the ground truth fundamental matrix ($\boldsymbol{F_0}$) and the fundamental matrix found using RANSAC ($\boldsymbol{F_1}$). The equation for the epipolar line in the left image can be obtained via **Equation 4**. The methodology described in the manual was then used to find valid endpoints of the epipolar lines found with the ground truth fundamental matrix.

$$Epipolar\ Lines\ in\ Left\ Image = \boldsymbol{F_\phi} * x_R\,, \qquad where\ \phi = 0{,}1 \qquad (4)$$

Then, from the remaining ground truth epipolar endpoint set and the epipolar lines from the $\boldsymbol{F_1}$, the perpendicular error was found at both ends of the region. For, each endpoint set ($\{x_0, y_0\}, \{x_1, y_1\}$), and corresponding epipolar line ($ax + by + c = 0$) from $\boldsymbol{F_1}$, the maximum perpendicular error can be found using **Equation 5** [1].

$$\max \perp distance = max\left(\frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}, \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}\right) \qquad (5)$$

The maximum of these maximum perpendicular distances is the error measure between F_0 and F_1. The 'findErrorInFEst.m' function was used to calculate the error. The error measure obtained is:

$$F_{error} = 4.3648e - 14$$

This is a very small error and it is partially due to floating point errors. The RANSAC fundamental matrix estimation provided a really good estimation of the ground truth fundamental matrix when there was no noise in the measurements.

1.3 Median Error vs. Noise

Here, a mean-zero noise with standard-deviation, σ_n, was added to the left and right image set. The error was found using the same function, 'findErrorInFEst.m', from **Section 1.2.** For each

σ_n, multiple trials were run with different noise samples, and the median error was found. The plot for σ_n vs. the median error can be seen in **Figure 1**.

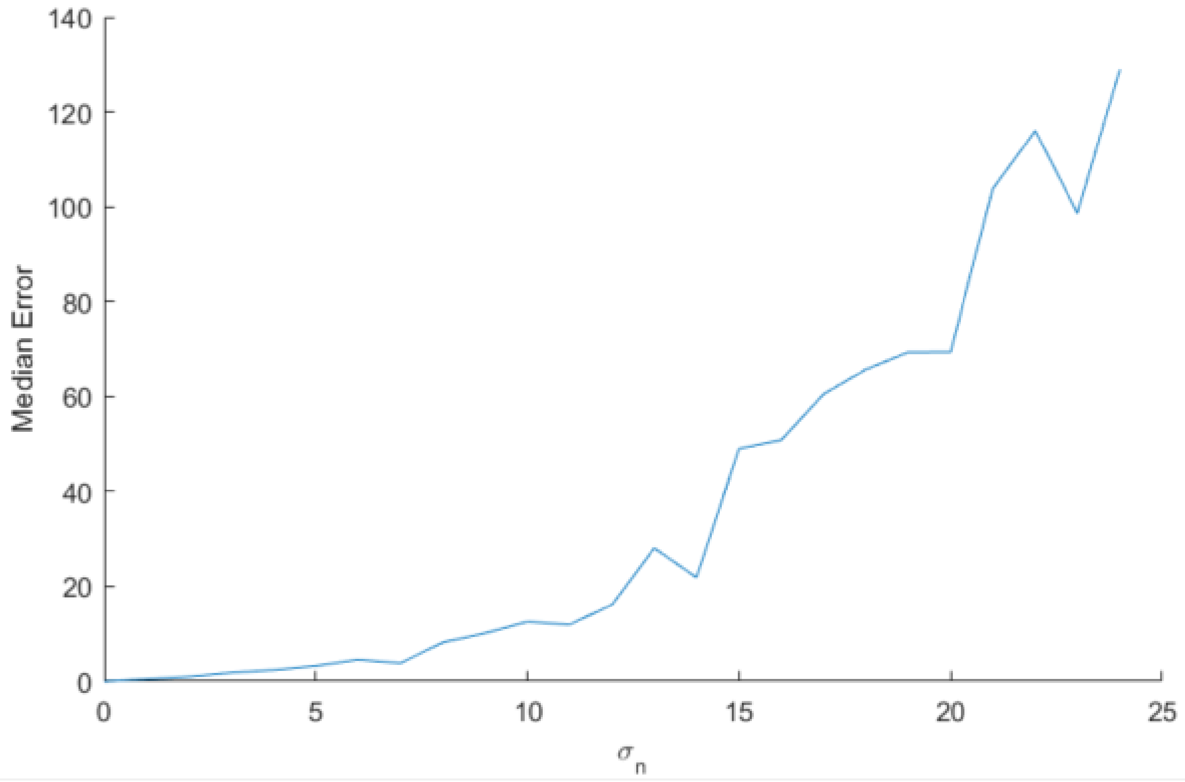

Figure 1: Median Error vs. σ_n

From **Figure 1**, it is seen that at small values of σ_n ($\sigma_n \leq 6$), the median error is near zero and increases approximately linearly with σ_n. However, as σ_n increases beyond 10, the median error exponentially increases. Also, the median error for adjacent values of σ_n can vary by large amounts as σ_n increases. It can be concluded that for $\sigma_n \leq 5$, we can get reasonably small errors for the recovered F-matrix that are proportional to the noise.

1.4 Median Error vs. Noise w/o Hartley Normalization

The median error vs. σ_n plot was computed again with the Hartley Normalization turned off. The resultant plot can be seen in **Figure 2**.

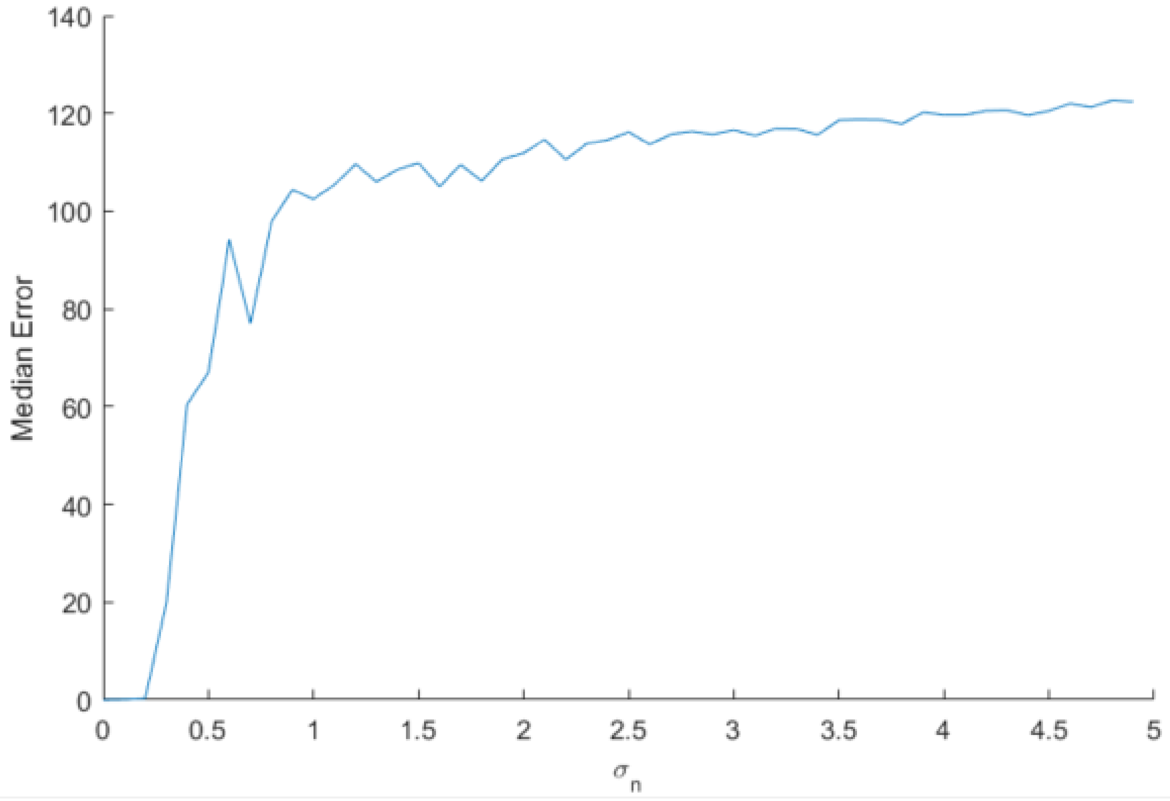

Figure 2: Median Error vs. σ_n w/o Hartley Normalization

From **Figure 2**, it is seen that by $\sigma_n = 0.5$, the median error is already at approximately 100. The computed F-matrix is very unstable when noise is added to the system. Hartley's normalization is used to improves the accuracy of the results [2]. The points $\{\vec{p}_k\}_{k=1}^{K}$ and $\{\vec{q}_k\}_{k=1}^{K}$, have the form $\{x, y, 1\}_{k=1}^{K}$. Without normalization, the x and y terms for each point are several orders of magnitude higher than 1. This causes the terms of the **A** matrix in the calculation for F to have some dominant elements, which makes the square sum of differen ces of **A** large. By normalizing, the terms in A are brought closer together and sum square of differences of A is decreased significantly [2]. This means that changing previously small terms of **A** will have a much greater effect than changing previously large terms of **A** [2]. Hartley's normalization is essential because it ensures that all elements of **A** have similar magnitudes [2].

1.5 Median Error vs. Noise w/ sclZ = 0.1

The median error vs. σ_n plot was computed for a third time with the Z component of the Dino (sclZ) changed to 0.1. The resultant plot can be seen in **Figure 3**.

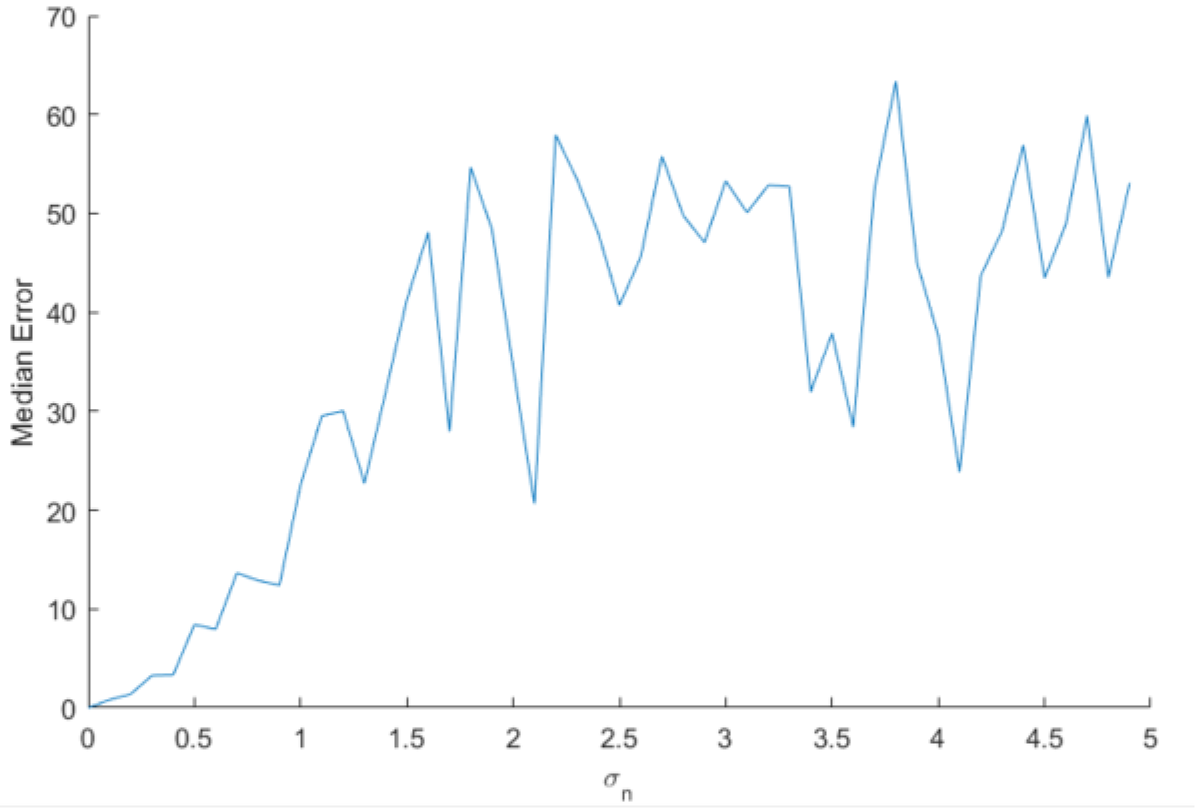

Figure 3: Median Error vs. σ_n w/ sclZ = 0.1

Here, the true **F**-matrix does not change when sclZ is changed because the parameters in the scene have no effect on the true **F**-matrix. The true **F**-matrix depends only on the relative positions and internal parameters of the camera. The **F**-matrix has no dependence on the scene structure. The estimation of **F**-matrix is less tolerant to image position noise than the case from **Section 1.3**. This is because the F matrix and it's corresponding epipolar lines are calculated via projective geometric relationships which are affected by noise. The smaller scale means the Dino in the real world is further away so it has a smaller projection onto both image frames.

Noise affects the correspondence of closer points more greatly, so the error increases more quickly with increasing noise.

1.6 Median Error vs. Noise w/ Smaller Camera Separation

The median error vs. σ_n plot was computed with the distance between the cameras decreased to 10. The resultant plot can be seen in **Figure 4**.

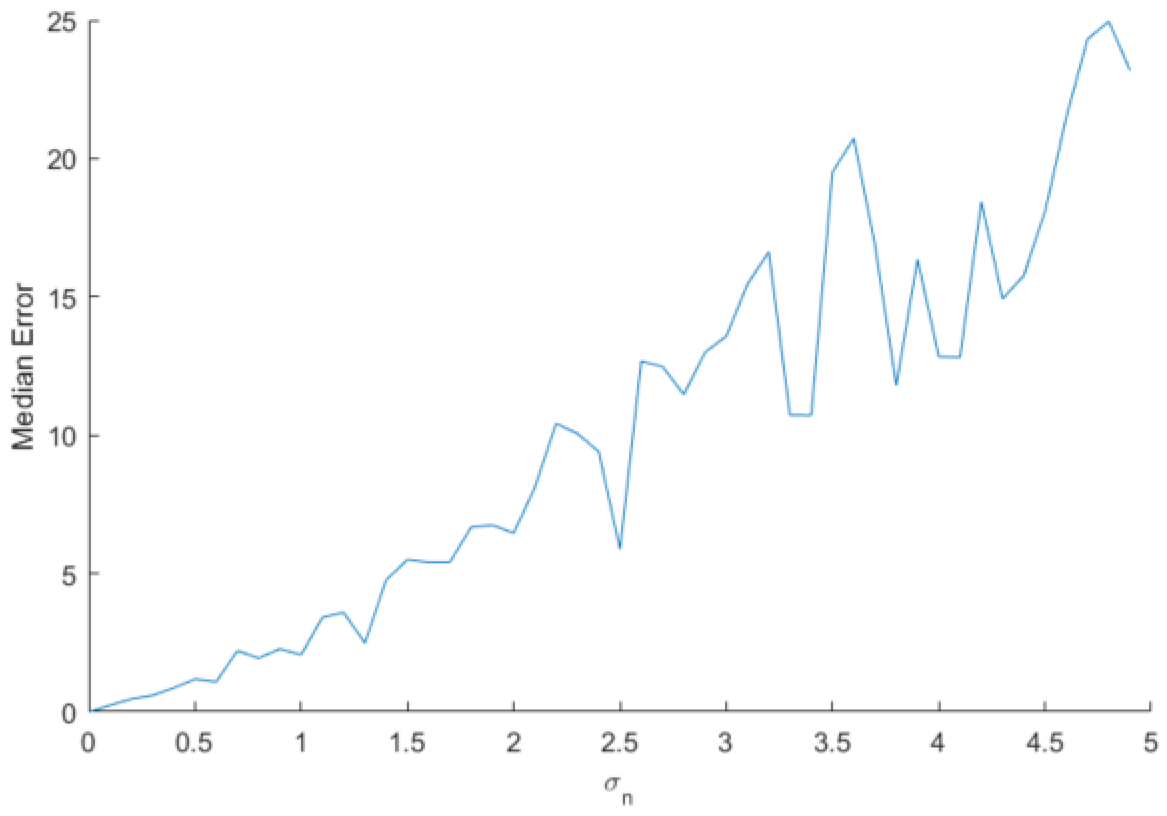

Figure 4: Median Error vs. σ_n w/ Reduced Camera Distance

The estimation of **F** is less tolerant to image noise when the cameras are closer together. When the cameras are really close together their epipolar lines are close to parallel. The ray that connects a point in the scene to the camera has a smaller projection onto the other camera's image frame, and less information can be derived from the scene (e.g. depth) since the images are almost from the same vantage point and provide more similar information. This error is similar to the scaling error as the points in one projection to another are closer together so the effect of noise on them is greater.

2.0 Homography Estimation

2.1 H-Estimator Function

We need to solve for $\boldsymbol{H}$ by minimizing the error equation (**Equation 6**). Firstly, let $\vec{q}_k$ and $\vec{p}_k$ be homogenous coordinates defined by $[u, v, 1]_k^T$ and $[x, y, 1]_k^T$, respectively. Let $\boldsymbol{H}$ be the 3x3 Homography matrix.

$$\mathcal{O} = \arg\min_{\mathrm{H}} \sum_{k=1}^{K} \sum_{j=1}^{2} [\vec{q}_k \times (\boldsymbol{H}\vec{p}_k)]_j^2 \qquad (6)$$

Firstly, the solution of the cross product can be seen in **Equation 7**.

$$\vec{q}_k \times \boldsymbol{H}\vec{p}_k = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_k \times \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_k = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_k \times \begin{bmatrix} x_k H_{11} + y_k H_{12} + H_{13} \\ x_k H_{21} + y_k H_{22} + H_{23} \\ x_k H_{31} + y_k H_{32} + H_{33} \end{bmatrix}$$

$$= \begin{bmatrix} v_k x_k H_{31} + v_k y_k H_{32} + v_k H_{33} - x_k H_{21} - y_k H_{22} - H_{23} \\ x_k H_{11} + y_k H_{12} + H_{13} - u_k x_k H_{31} - u_k y_k H_{32} - u_k H_{33} \\ u_k x_k H_{21} + u_k y_k H_{22} + u_k H_{23} - v_k x_k H_{11} - v_k y_k H_{12} - v_k H_{13} \end{bmatrix} \quad (7)$$

Here, the last row can be ignored because it is dependent on the other two rows. The cross product that can be written as the product of an $\boldsymbol{A}$ matrix and a h vector as seen in **Equation 8**.

$$\sum_{j=1}^{2}[\vec{q}_k \times (H\vec{p}_k)]_j =$$

$$\begin{bmatrix} 0 & 0 & 0 & -x_k & -y_k & -1 & v_k x_k & v_k y_k & v_k \\ x_k & y_k & 1 & 0 & 0 & 0 & -u_k x_k & -u_k y_k & -u_k \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ H_{33} \end{bmatrix} = \boldsymbol{A_k} h \qquad (8)$$

Plugging this back into **Equation 6** and writing it in matrix notation yields:

$$O = \arg\min_{\mathrm{h}} \sum_{k=1}^{K} (\boldsymbol{A_k} h)^2 = \arg\min_{\mathrm{h}} (\boldsymbol{A_{2K\times 9}} h_{9\times 1}) = \arg\min_{\mathrm{h}} (h_{1\times 9}^T \boldsymbol{A_{9\times 2K}^T} \boldsymbol{A_{2K\times 9}} h_{9\times 1}) \qquad (9)$$

Each correspondence k contribute two rows to the $\boldsymbol{A}$ matrix. Now, to minimize this objective, we take the derivative of this equation and set it equal to 0 as seen below:

$$\frac{\partial O}{\partial h} = \boldsymbol{A_{9X2K}^T} \boldsymbol{A_{2Kx9}} h_{9x1} = 0 \qquad (10)$$

This is a homogeneous least squares problem that can be solved by finding the Singular Value Decomposition (SVD) of $\boldsymbol{A^T A}$ (i.e $\boldsymbol{A^T A} = \boldsymbol{U \Sigma V^*}$). Then h will equal the column of $\boldsymbol{V}$ with the smallest eigenvalue in $\boldsymbol{\Sigma}$, and $\boldsymbol{\Sigma}$ is arranged in descending order along the diagonal so the solution will be the last column of $\boldsymbol{V}$.

For numerical stability, the coordinates used in the $\boldsymbol{H}$ calculations are normalized. The following steps would be taken to solve for $\boldsymbol{H}$.

1. A similarity transform matrix, $\boldsymbol{T_p}$, will be computed and applied to the homogeneous $\vec{p}_k$ coordinates to get the normalized $\tilde{\vec{p}}_k$ coordinates, such that the centroid of the

normalized point is at the coordinate origin, and all the points have an average distance of $\sqrt{2}$ from the origin [2]. This was done in the provided code.

2. A similarity transform matrix, $\boldsymbol{T_q}$, will be computed and applied to the homogeneous $\vec{q}_k$ coordinates to get the normalized $\tilde{\vec{q}}_k$ coordinates in a similar fashion. This was also done in the provided code.
3. Compute a 2×9 $\widetilde{\boldsymbol{A}}_{\boldsymbol{k}}$ matrix for each normalized point correspondence given
4. Assemble K 2×9 $\widetilde{\boldsymbol{A}}_{\boldsymbol{k}}$ matrices into one $2K\times9$ matrix $\widetilde{\boldsymbol{A}}$
5. Compute the SVD of $\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}}$
6. The homography vector $\tilde{h}$, as explained above, is the equal to the rightmost column of the $\widetilde{\boldsymbol{V}}$ matrix from the SVD solution of $\widetilde{\boldsymbol{A}}^T\widetilde{\boldsymbol{A}}$**.** The normalized Homography matrix ($\boldsymbol{H_0}$) can be found by remapping $\tilde{h}$ to a 3×3 matrix.
7. Finally, the Homography matrix in the original coordinate system can be found by undoing the transformations applied in steps 1 and 2 via this formula:

$$\boldsymbol{H} = \boldsymbol{T_q^{-1}H_0T_p} \qquad \textbf{(11)}$$

8. Finally, rescale the Homography matrix so that the sum of squares of its 9 elements is one. This can be done by dividing each element by the square root of its current sum of squares.

2.2 RANSAC to Optimize Homography

The RANSAC algorithm used is as follows:

1. Pick 4 random points and compute the Homography using the 'linEstH.m' function.
2. Solve for the corresponding q_k in the right image using the calculated homography, and all points in the left image
3. Normalize the q_k vectors by dividing all its elements by its third element
4. Compute the absolute Euclidean distance between the calculated q_k and the measured value from the right image. This will give the absolute value of the error, also known as the L_1 error.
5. Inliers were defined as all points that fell within an absolute distance of 1.5 pixels.

6. Steps 1-5 were repeated 10 times and the set with the most inliers was chosen to continue with the algorithm
7. Next, another 10 sets of iterations were done to entrain the inliers. The Homography matrix was calculated using all the inliers remaining.
8. The error was solved for, and the new number of inliers were found. If this number was equal to the previous number of inlier, the loop terminated, and the Homography matrix was considered entrained. Otherwise, step 7 was completed with the new set of inliers, until 10 iterations were completed

Figure 5 shows the original left image, and **Figure 6** shows the right image warped to the left image. **Figure 7** shows the original right image, and **Figure 8** shows the left image warped to the right image.

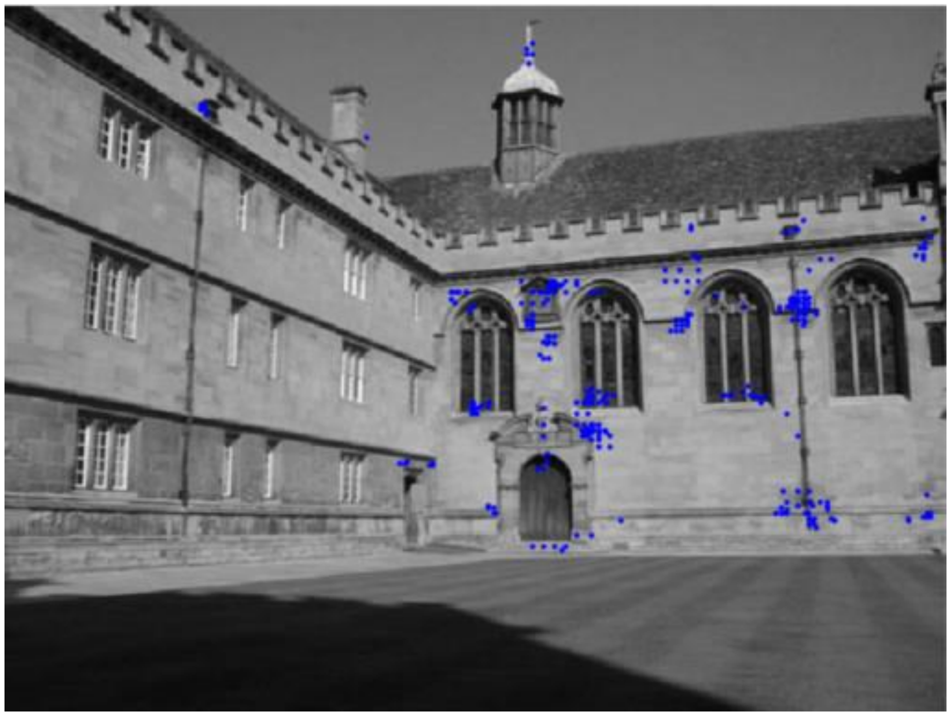

Figure 5: Original Left Image of Wadham College

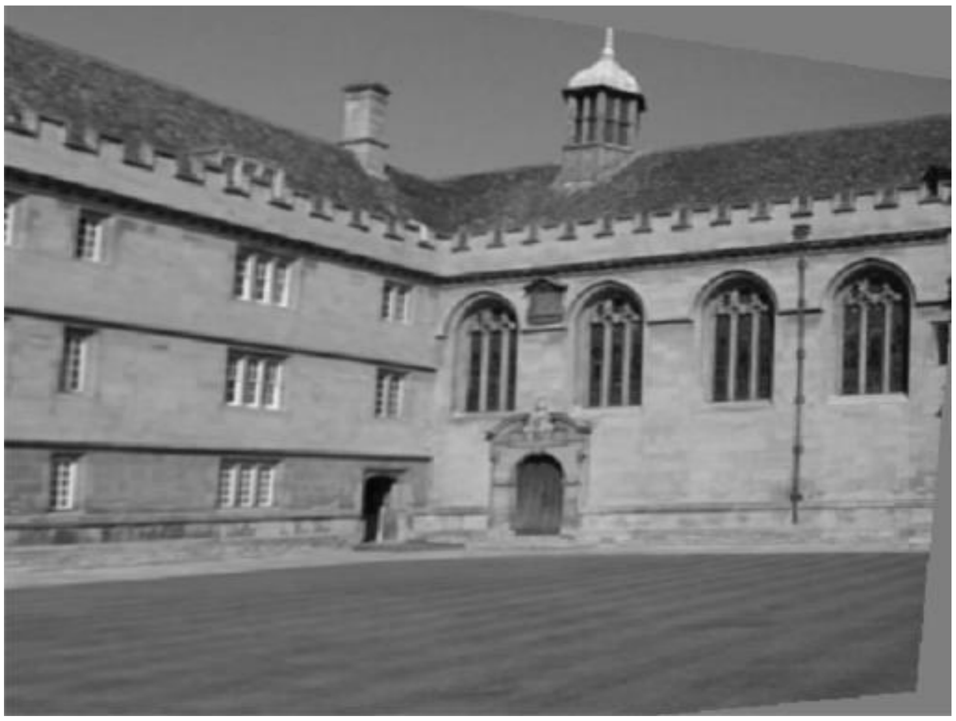

Figure 6: Right Image Warped to Left Image of Wadham College

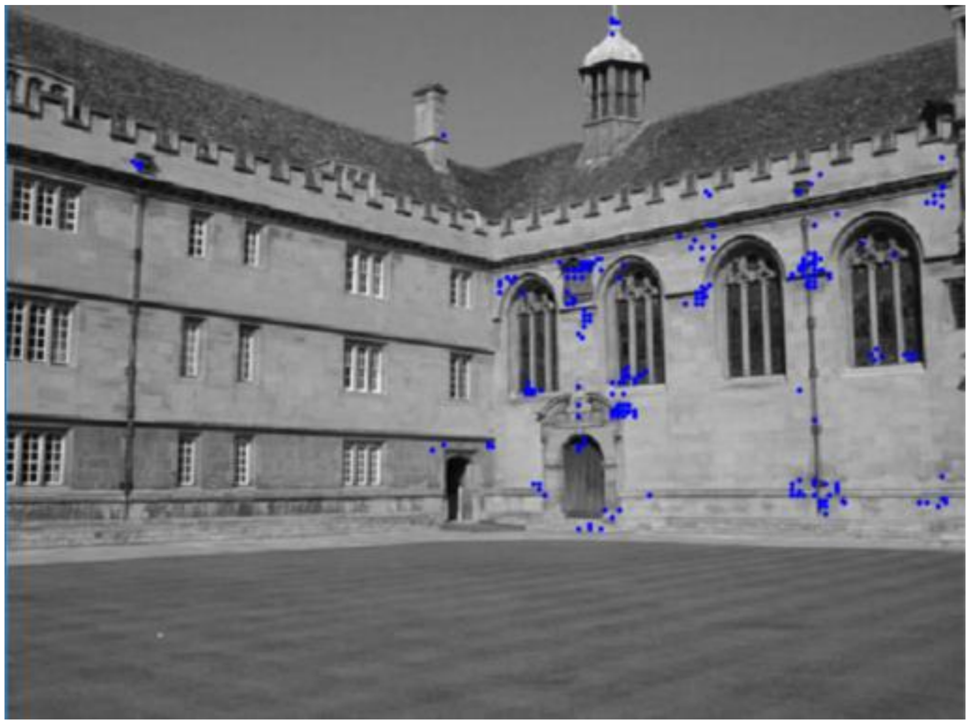

Figure 7: Original Right Image of Wadham College

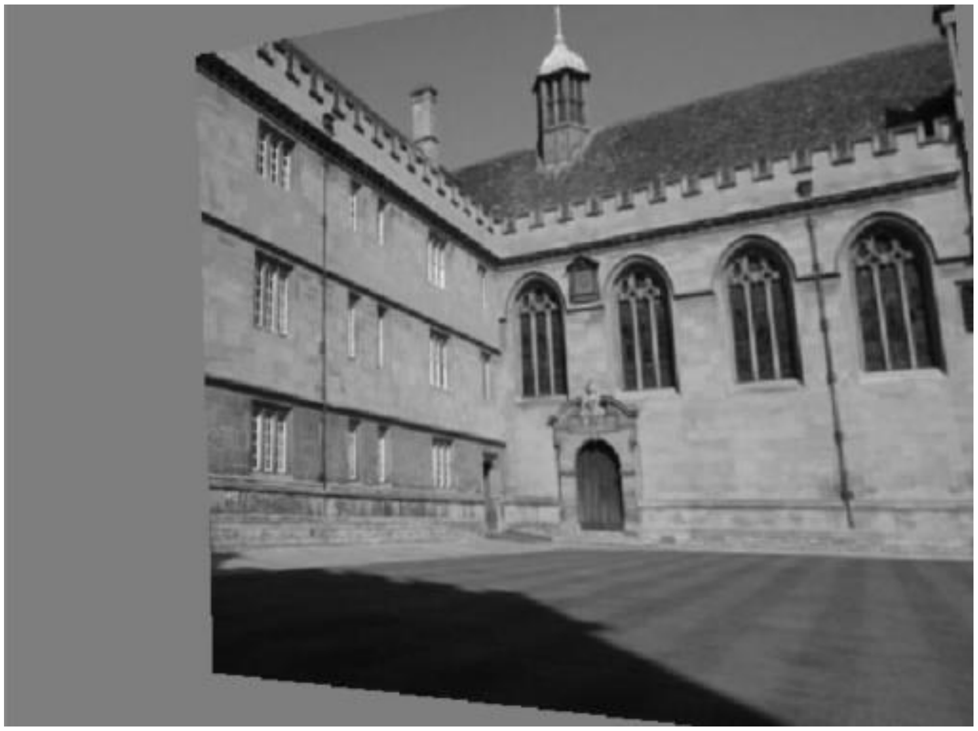

Figure 8: Left Image Warped to Right Image of Wadham College

From the above figures, it is seen that the right-hand wall matches up well in both sets of images because that was the plane used to calculate the Homography. The left-hand wall however does not line up well. This is because the Homography matrix is different for both walls. A Homography matrix relates points between two images that fall on a specific 3D plane. The right-hand wall also had a high probability of being chosen from RANSAC because it had significantly more inliers points than the left-hand wall.

3.0 References

[1] https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line#Line_defined_by_an_equation

[2] Richard Hartley and Andrew Zisserman. 2003. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press, New York, NY, USA.