COMPUTER VISION PROJECT
ON
Camera Calibration and Fundumental Matrix Estimation Using RANSAC

SUBMITTED By

DIBYENDU DAS
NILABJANAYAN BERA

UNDER THE GUIDANCE OF
TAMAL MJ

Ramakrishna Mission Vivekananda Educational and Research Institute
BELUR

2021

Contents

1 INTRODUCTION :

In this project, we use the geometric relationships between images taken from multiple views to compute camera positions and estimate fundamental matrices for various scenes.Specifically we will estimate the camera projection matrix or calibration matrix, which maps 3D world coordinates to image coordinates, as well as the fundamental matrix, which relates points in one scene to epipolar lines in another. The camera projection matrix and the fundamental matrix can each be estimated using point correspondences. To estimate the projection matrix (camera calibration), the input is corresponding 3d and 2d points. To estimate the fundamental matrix the input is corresponding 2d points across two images. We will start out by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we eill move on to estimating the fundamental matrix using point correspondences from ORB. By using RANSAC to find the fundamental matrix with the most inliers, we can filter away spurious matches and achieve near perfect point to point matching.

2 PROCESS :

2.1 Projection Matrix:

The camera projection matrix maps homogeneous 3D coordinates to 2D image coordinates such that -

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq M \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

Where M is camera projection matrix.

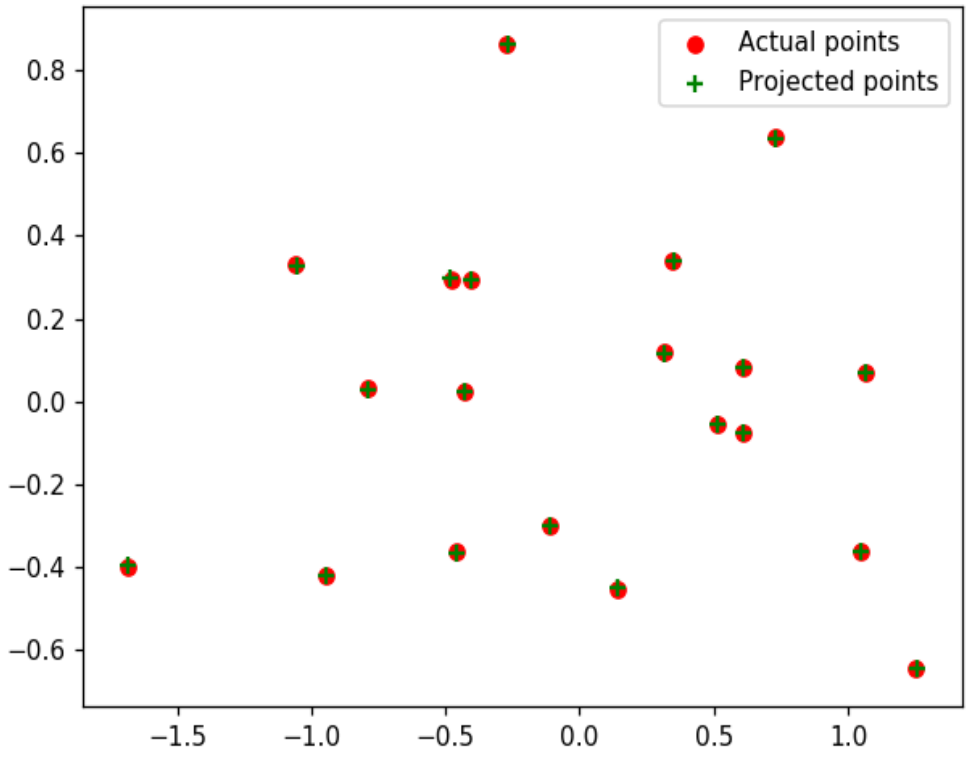
$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix}$$

We can solve for M by setting up a homogeneous system -

$$\begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1Z_1 & -v_1Y_1 & -v_1Z_1 \\ . & . & . & . & . & . & . & . & . & . & . \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nZ_n & -v_nY_n & -v_nZ_n \end{pmatrix} * \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_n \\ v_n \end{pmatrix}$$

M has 12 elements, but since the scaling term m_{34} is arbitrary, we only need to solve for 11 variables. We solved this using least squares with np.linalg.lstsq function.

$$M = \begin{pmatrix} 0.45828095 & -0.29474332 & -0.01396452 & 0.00402529 \\ -0.05085792 & -0.05459096 & -0.54104038 & -0.05237589 \\ 0.10901111 & 0.17835024 & -0.04428027 & 0.59683007 \end{pmatrix}$$



2.2 Camera Center :

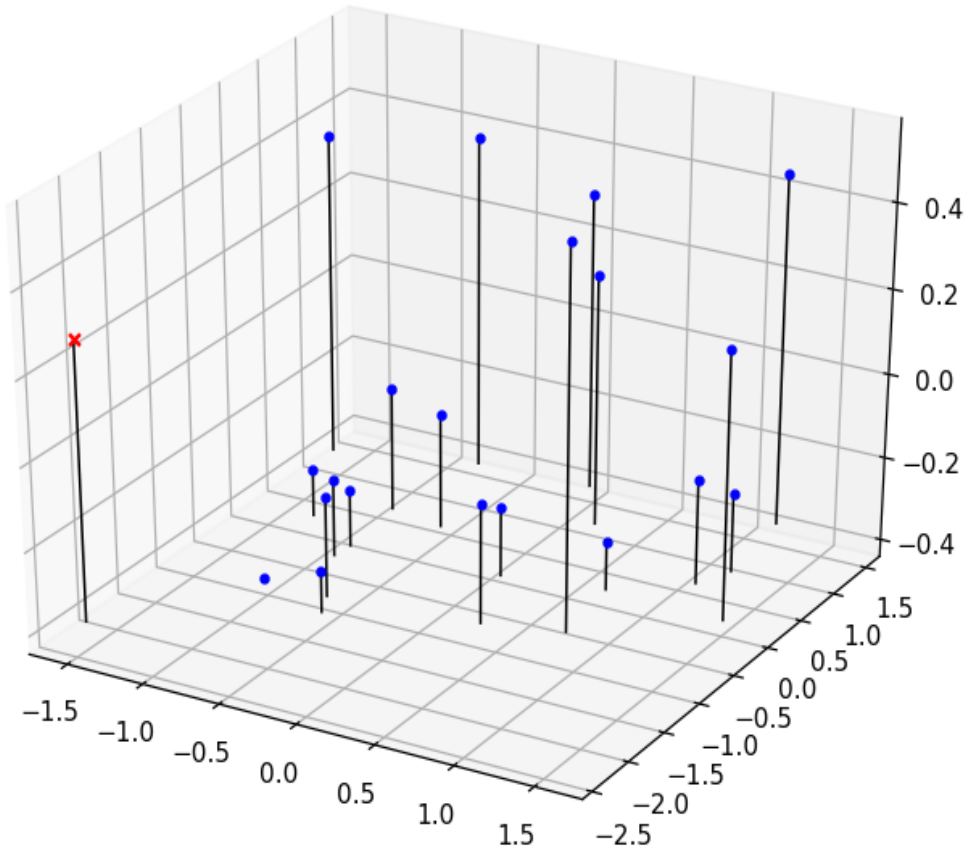
Writing M as the concatenation of a 3x3 matrix Q and a column matrix m4 as such

$$M = (Q|m_4)$$

we can compute the camera center by solving

$$\begin{aligned} C &= -Q^{-1}m_4 \\ &= -\begin{pmatrix} 0.45828095 & -0.29474332 & -0.01396452 \\ -0.05085792 & -0.05459096 & -0.54104038 \\ 0.10901111 & 0.17835024 & -0.04428027 \end{pmatrix}^{-1} \begin{pmatrix} 0.00402529 \\ -0.05237589 \\ 0.59683007 \end{pmatrix} \\ &= \begin{pmatrix} -1.51263977 \\ -2.35165965 \\ 0.28266502 \end{pmatrix} \end{aligned}$$

. The total residual is 0.044535.



2.3 Fundamental Matrix Estimation :

The next part of this project is estimating the mapping of points in one image to lines in another by means of the fundamental matrix.The normalized eight-point algorithm is used to compute the fundamental matrix given point correspondences x = (u, v) and x' = (u', v') in the left and right images, respectively. Each point correspondence generates one constraint on the fundamental matrix F and must satisfy the epipolar constraint equation

$$x'^T F x = 0.$$

Expanding the matrices out by multiplication, we obtain the following equation for n point correspondences

$$\begin{pmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_nu_n & u'_nv_n & u'_n & v'_nu_n & v'_nv_n & v'_n & u_n & v_n & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \mathbf{0}$$

$$\Leftrightarrow \mathbf{Af} = 0.$$

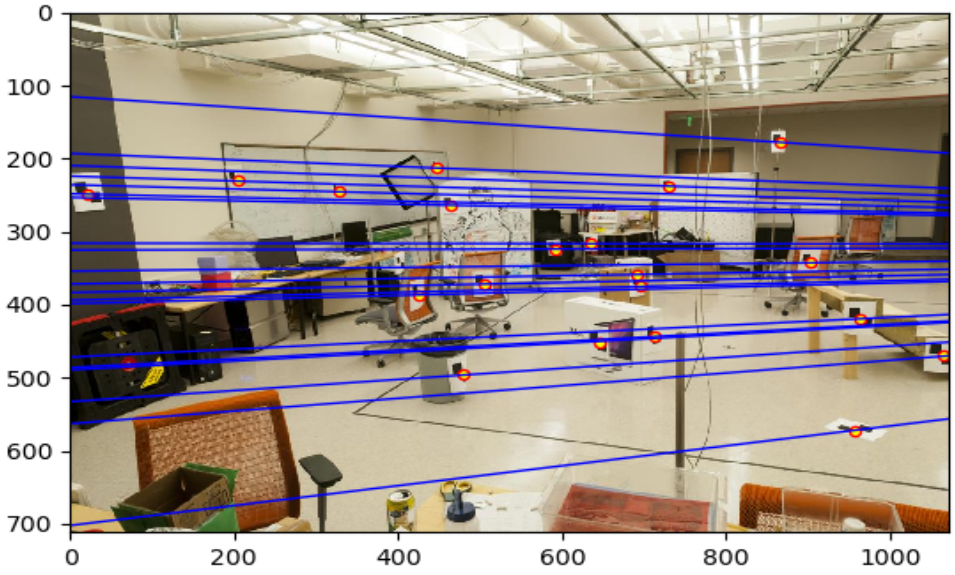
where A is the n x 9 equation matrix, and f is a 9-element column vector containing the entries of the fundamental matrix F. From here, the least-squares solution f is easily computed by performing singular value decomposition (SVD) on the matrix $A = UDV^T$. It is well-known that the vector f that minimizes $||Af||$ such that $||f|| = 1$ can be found along the column of V corresponding to the least singular value. Next, we rearrange the 9 entries of f to create the 3x3 fundamental matrix F. Then, we perform SVD on F to obtain

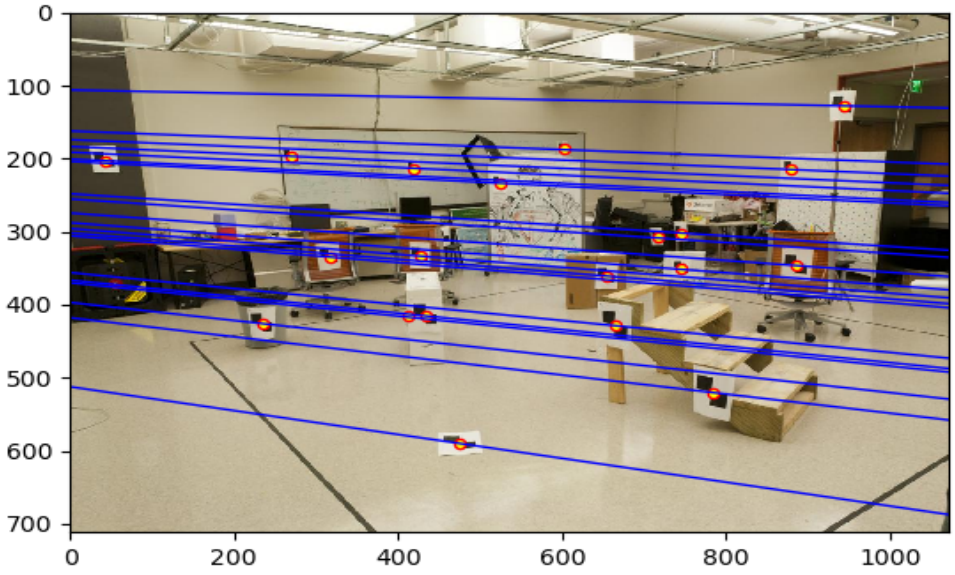
$$F = U_f D_f {V_f}^T$$

We set the smallest singular value of Df to 0 to create matrix D'f, thus reducing the rank of the matrix from 3 to 2, and from there we can recompute the rank-2 fundamental matrix as

$$F = U_f D_f {V_f}^T.$$

Using the version of the eight-point algorithm without prior coordinate normalization to compute the fundamental matrix on the laboratory image pair, we obtain the following fundamental matrix F (truncated to 4 decimal places) and the epipolar lines for both images:





Now let's look at some results we are getting.

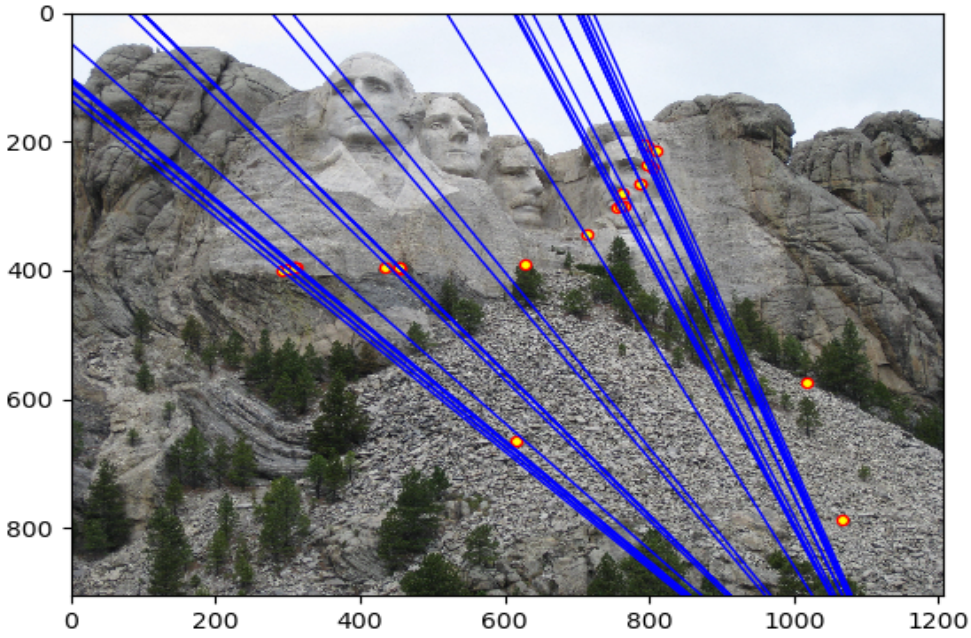


Figure 1: Epipolar Lines in Mount Rushmore Image - 1

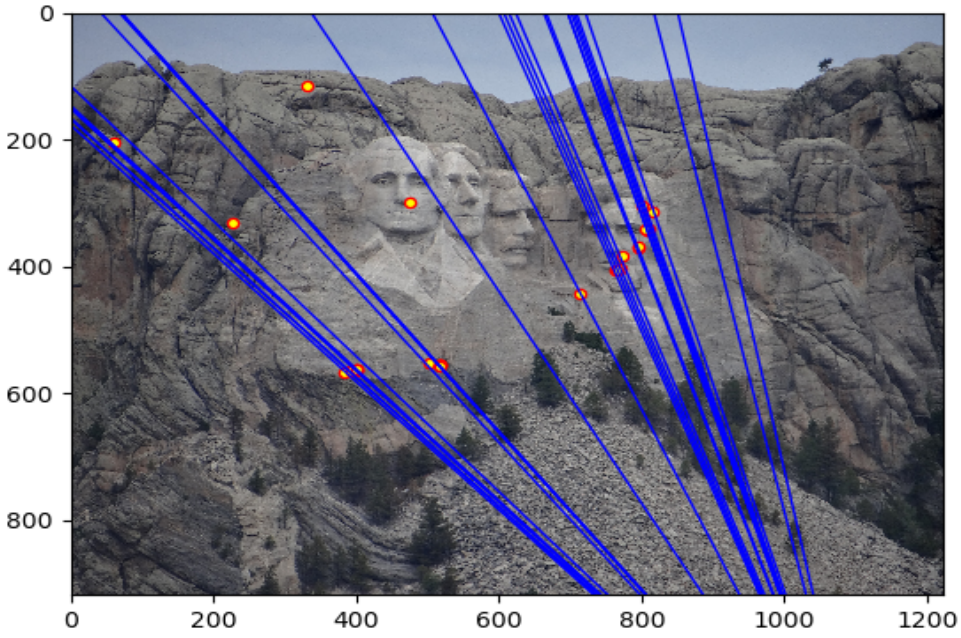


Figure 2: Epipolar Lines in Mount Rushmore Image - 2

If we look at the results, we can notice that the epipolar lines don't pass exactly through the center of all the point correspondences. How can we improve this? However, in order to improve the accuracy of epipolar lines generated here, we have to maintain this: solve for $Af = 0$,such that$||f|| = 1$.Therefore, we want to modify the 'vanilla' eight-point algorithm to recenter the point correspondences xi = (ui, vi) and x'i = (u'i, v'i) in both images to their respective centroids before proceeding to compute the least-squares solution for f. After recentering the image points, we must scale the points to be a fixed squared distance from the origin. The coordinate normalization steps can be summarized in the following steps:
1.Compute the centroid of all corresponding points in a single image:

$$\bar{\mathbf{u}} = \frac{1}{n}\sum_{i=1}^{n} u_i$$
$$\bar{\mathbf{v}} = \frac{1}{n}\sum_{i=1}^{n} v_i$$

2.Recenter by subtracting the mean u and v coordinates from the original point correspondences to obtain

$$\tilde{\mathbf{u}} = \mathbf{u} - \bar{\mathbf{u}}$$
$$\tilde{\mathbf{v}} = \mathbf{v} - \bar{\mathbf{v}}$$

3.Define the scale term s and s' to be the average distances of the centered points from the origin in both the left and right images:

$$\mathbf{s} = \frac{\sqrt{2}}{\left(\frac{1}{n}\sum_{i=1}^{n}(\tilde{u}_i^2 + \tilde{v}_i^2)\right)^{1/2}}$$

$$\mathbf{s}' = \frac{\sqrt{2}}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(\tilde{u'}_i^2 + \tilde{v'}_i^2\right)\right)^{1/2}}$$

4.Construct the transformation matrices Ta and Tb:

$$T_a = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\bar{\mathbf{u}} \\ 0 & 1 & -\bar{\mathbf{v}} \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_b = \begin{pmatrix} s' & 0 & 0 \\ 0 & s' & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\bar{\mathbf{u}}' \\ 0 & 1 & -\bar{\mathbf{v}}' \\ 0 & 0 & 1 \end{pmatrix}$$

5.The normalized correspondence points can be computed from

$$\hat{\mathbf{x}}_\mathbf{i} = T_a\mathbf{x}$$
$$\hat{\mathbf{x}}'_\mathbf{i} = T_b\mathbf{x}$$

6.Solve for the fundamental matrix F by applying the eight-point algorithm on the normalized set of point correspondences computed in the previous step. 7.After obtaining the normalized fundamental matrix F_{norm}, retrieve the fundamental matrix in the original coordinate frame using the following formula

$$F_{orig} = T_b^T * F_{norm} * T_a$$

8.Using the normalized eight-point algorithm on the laboratory image pair, we get this fundumental matrix:

$$F = \begin{pmatrix} -1.17248591e^{-07} & 1.60824663e^{-06} & -4.01980786e^{-04} \\ 1.11212887e^{-06} & -2.73443755e^{-07} & 3.23319884e^{-03} \\ -2.36400817e^{-05} & -4.44404958e{-}^{03} & 1.03455561e^{-01} \end{pmatrix}$$

With RANSAC we will find the best fundumental matrix and will show the results for 4 pairs of different images.

2.4 Fundamental Matrix with RANSAC :

In this part we will find the best fundamental matrix using RANSAC on randomly selected potentially eight matching points from two images of same object.
In practical situations, we will not be given ground truth correspondences between two images, so there must be a way to compute interest points and predict matches between stereo image pairs. We use the ORB descriptors around detected interest points. We will find the best fundumental matrix using RANSAC and also we will find the inliers. With the maximum no. of inliers we will decide which one is the best fundumental matrix. We ran our algorithm on "Mount Rushmore","Notre Dame","Gaudi","Woodruff" these images and got these results.

2.4.1 Mount Rushmore:

This pair is easy, and most of the initial matches are correct. The base fundamental matrix estimation without coordinate normalization will work fine with RANSAC.

Unnormalized:

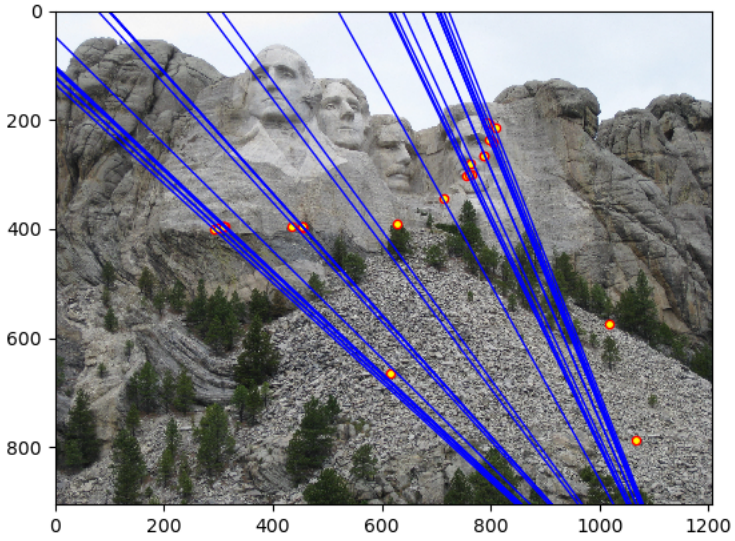


Figure 3: Epipolar Lines in first Image

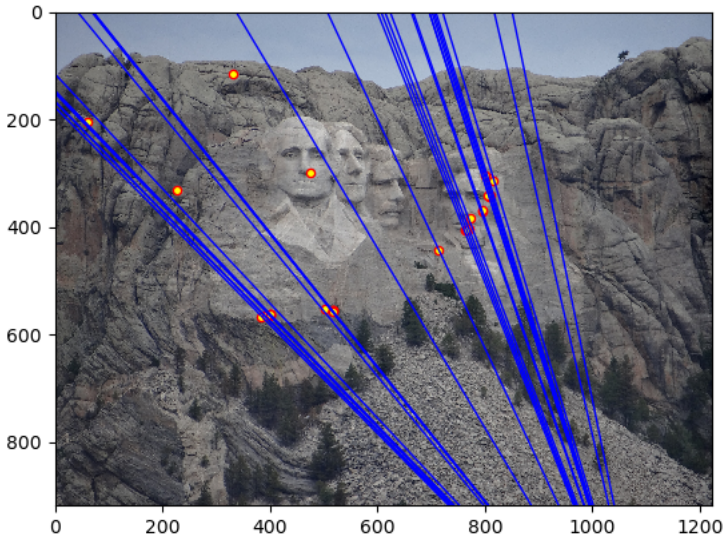


Figure 4: Epipolar Lines in second Image

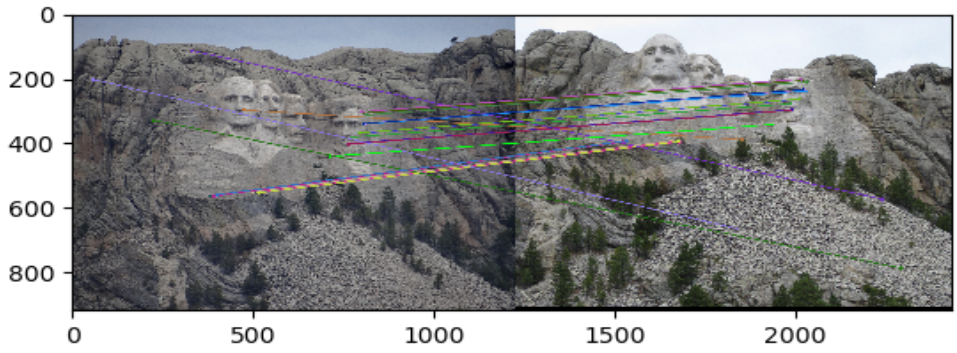


Figure 5: Match Two Images

Normalized:

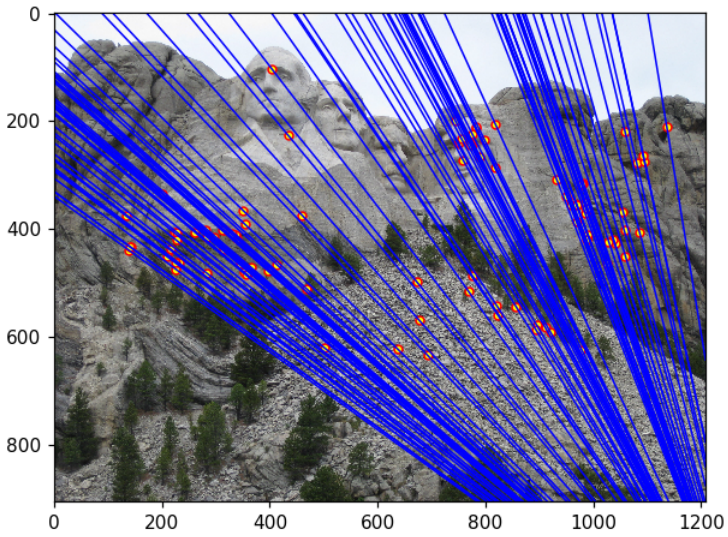


Figure 6: Epipolar Lines in first Image

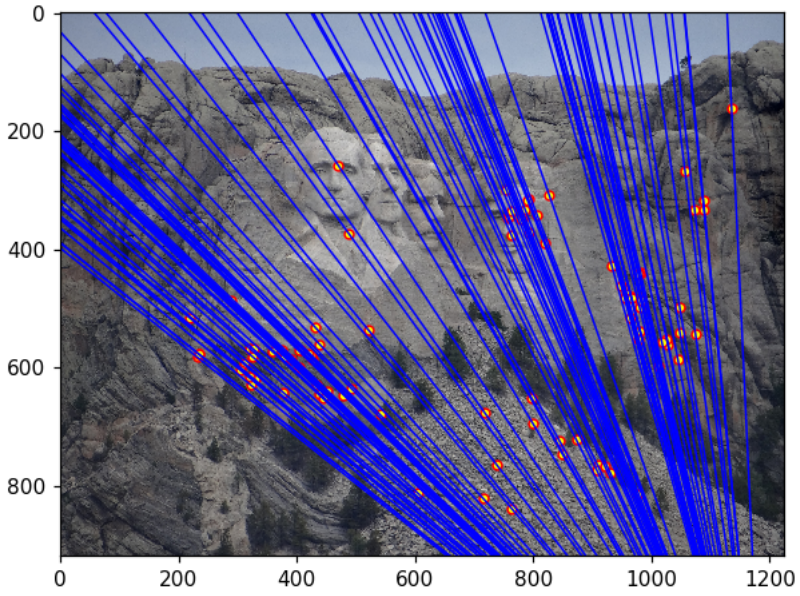


Figure 7: Epipolar Lines in second Image

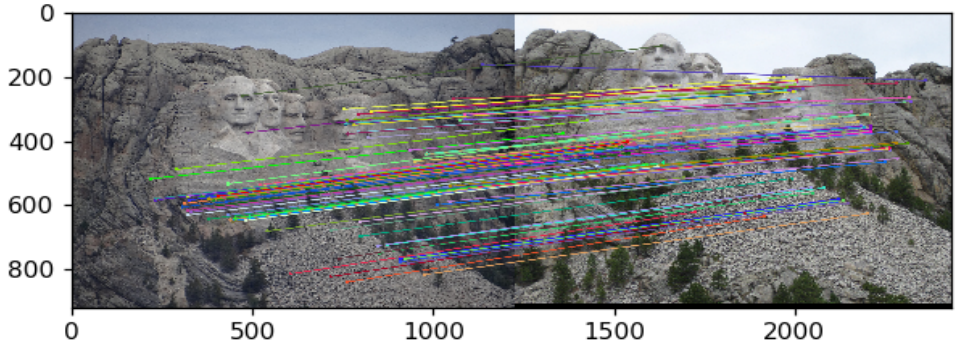


Figure 8: Match Two Images

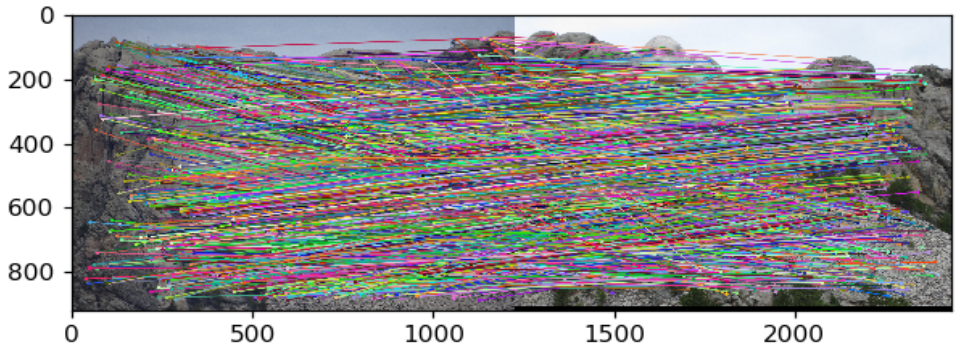


Figure 9: Matching Using ORB

2.4.2 Notre Dame:

This pair is difficult because the keypoints are largely on the same plane. Still, even an inaccurate fundamental matrix can do a pretty good job of filtering spurious matches.

Unnormalized:

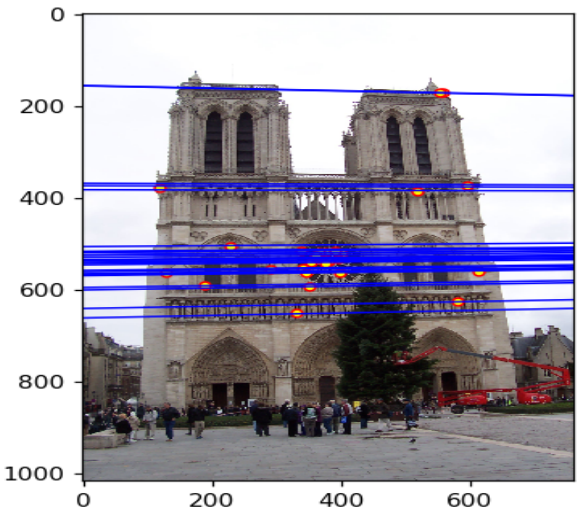


Figure 10: Epipolar Lines in first Image

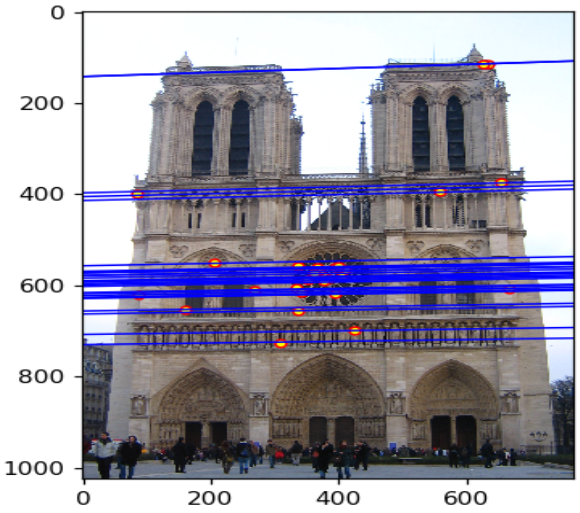


Figure 11: Epipolar Lines in second Image

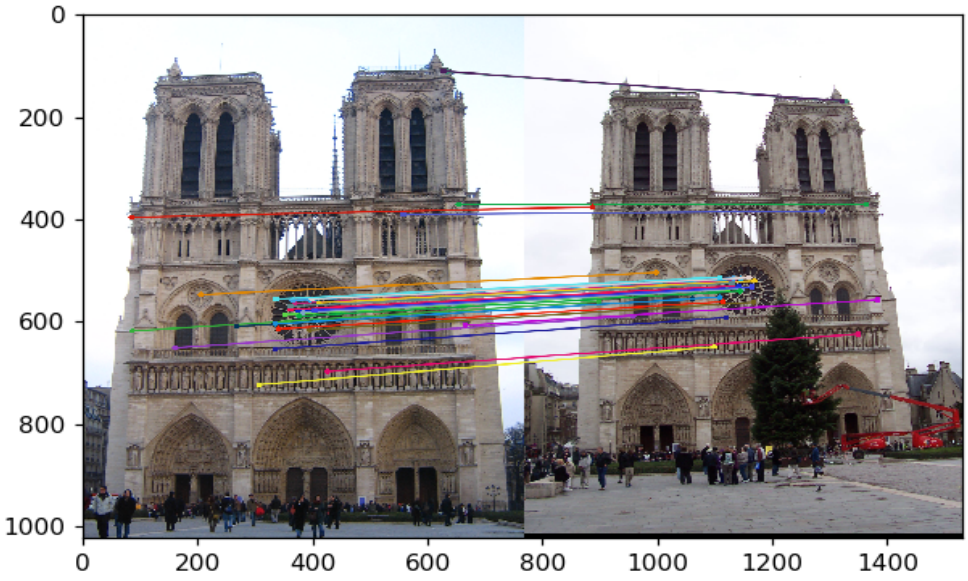


Figure 12: Match Two Images

Normalized:

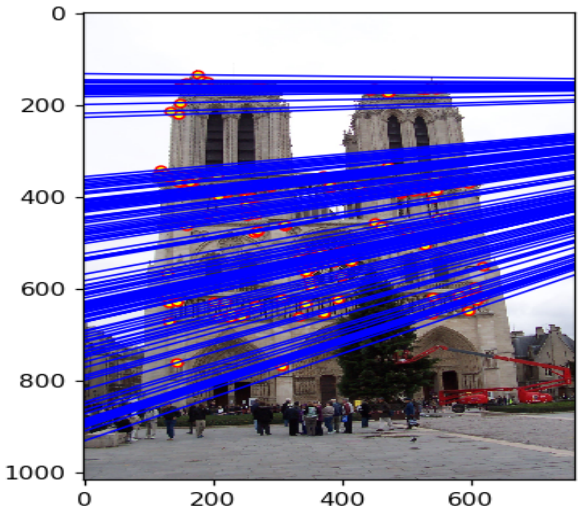


Figure 13: Epipolar Lines in first Image

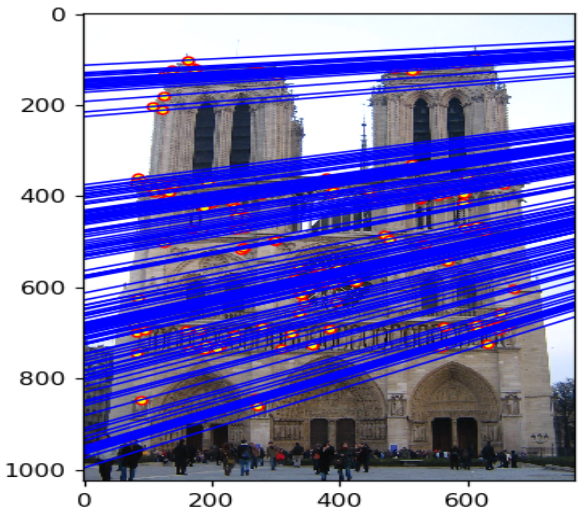


Figure 14: Epipolar Lines in second Image

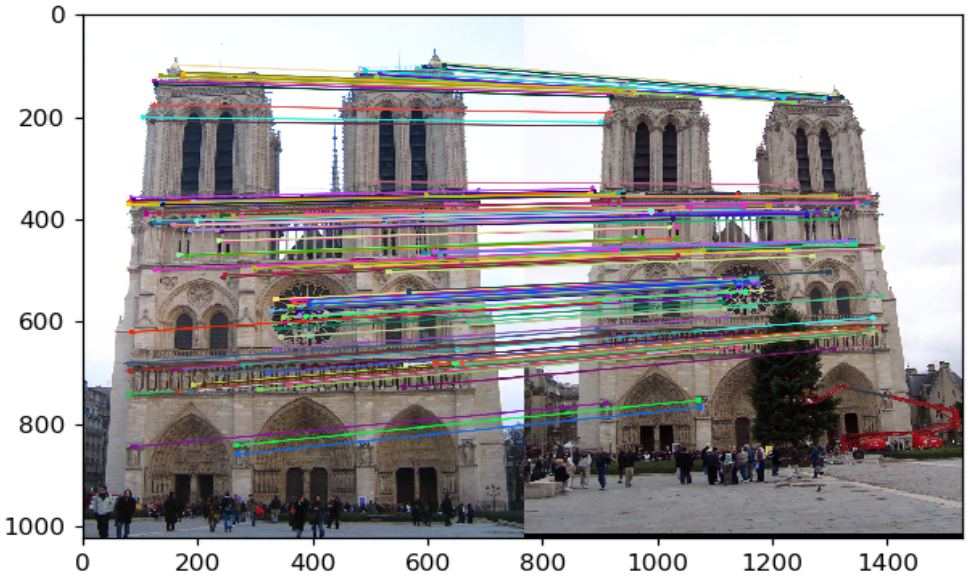


Figure 15: Match Two Images

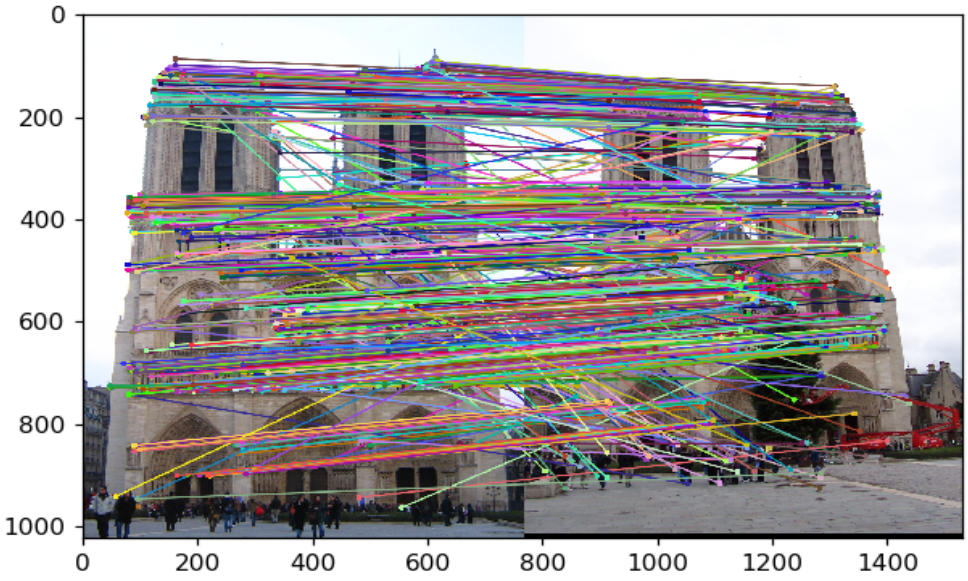


Figure 16: Matching Using ORB

2.4.3 Gaudi:

This pair is difficult and doesn't find many correct matches unless you run at high resolution, but that will lead to tens of thousands of ORB features, which will be somewhat slow to process. Normalizing the coordinates seems to make this pair work much better.

Unnormalized:

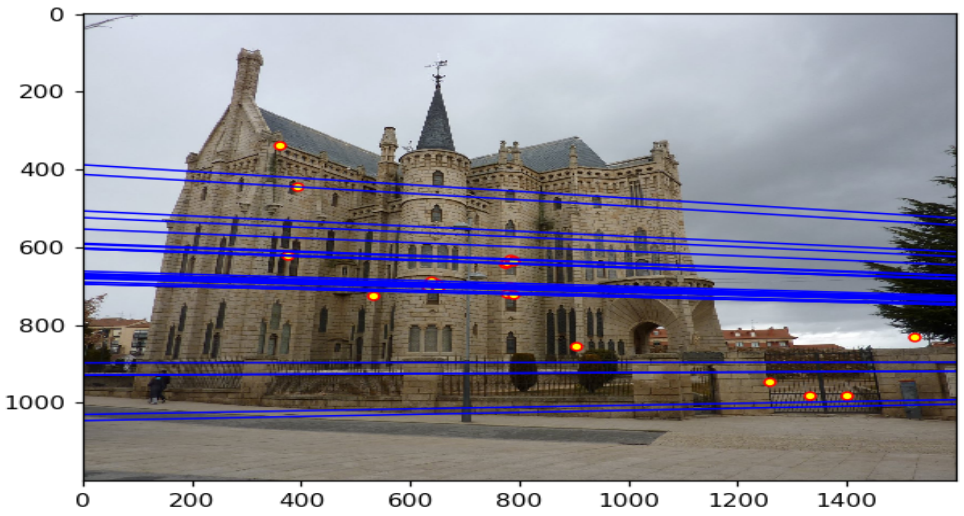


Figure 17: Epipolar Lines in first Image

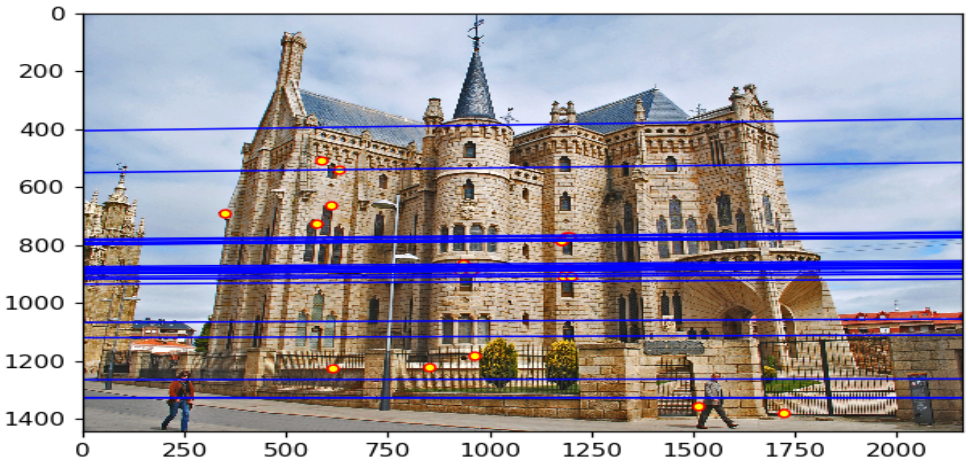


Figure 18: Epipolar Lines in second Image

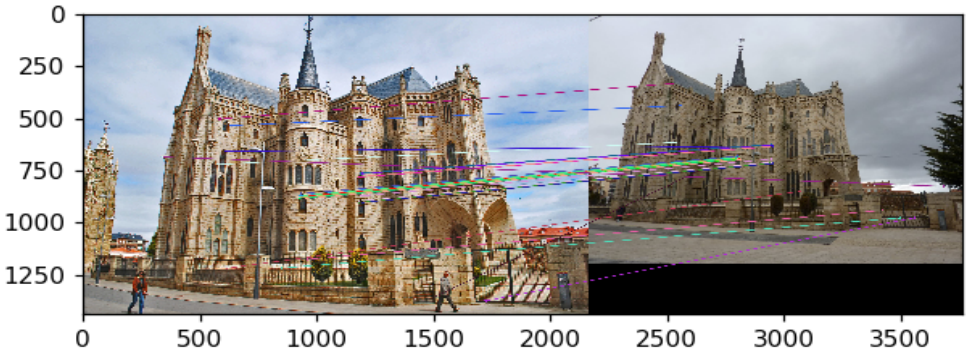


Figure 19: Match Two Images

Normalized:

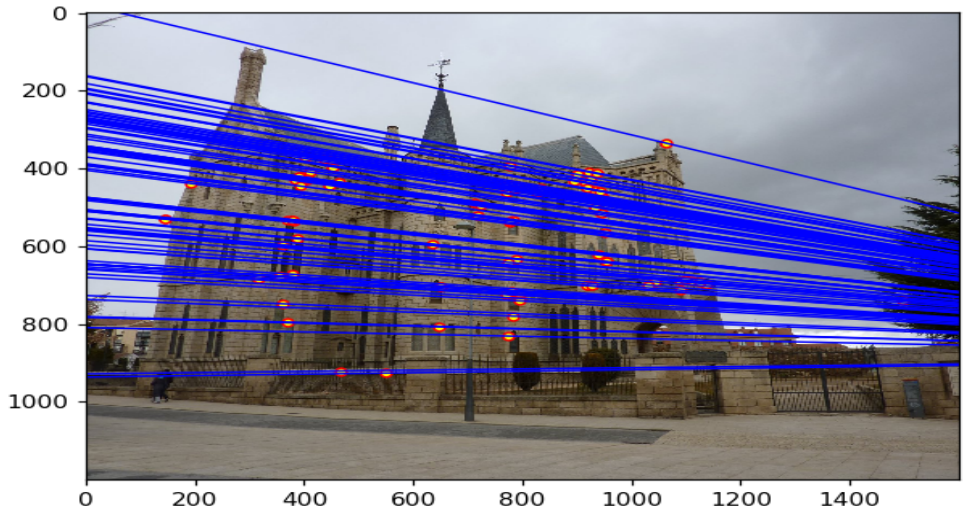


Figure 20: Epipolar Lines in first Image

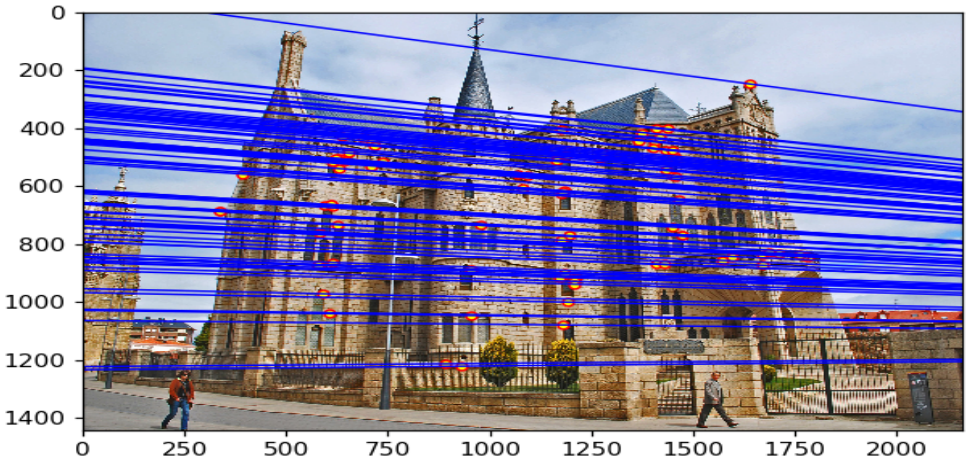


Figure 21: Epipolar Lines in second Image

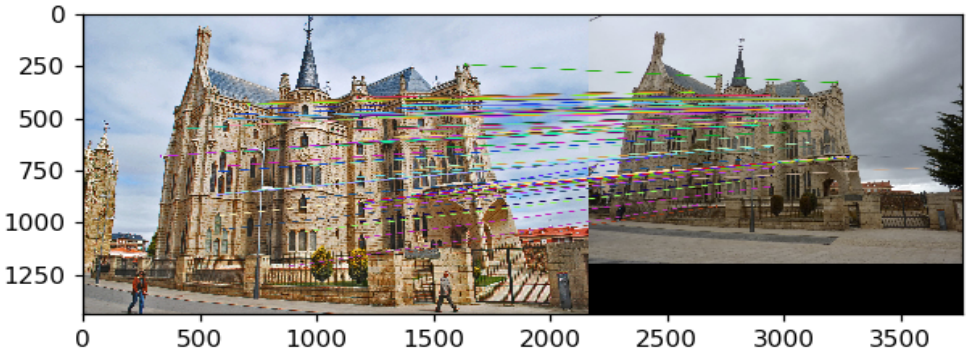


Figure 22: Match Two Images

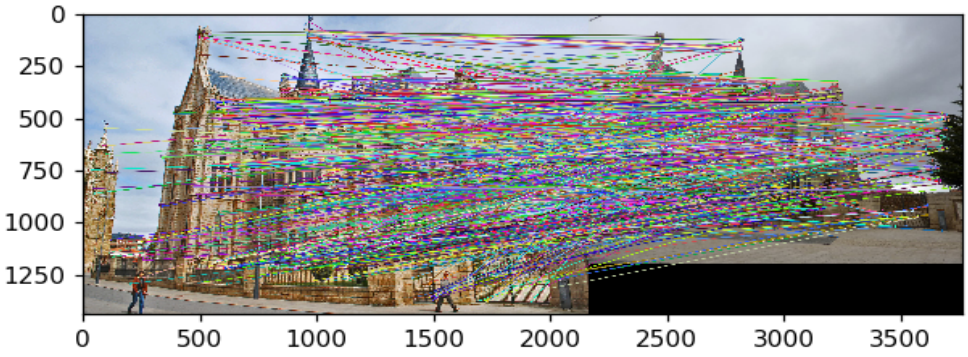


Figure 23: Matching Using ORB

2.4.4 Woodruff:

This pair has a clearer relationship between the cameras (they are converging and have a wide baseline between them). The estimated fundamental matrix is less ambiguous and you should get epipolar lines qualitatively similar to part 2 of the project.

Unnormalized:

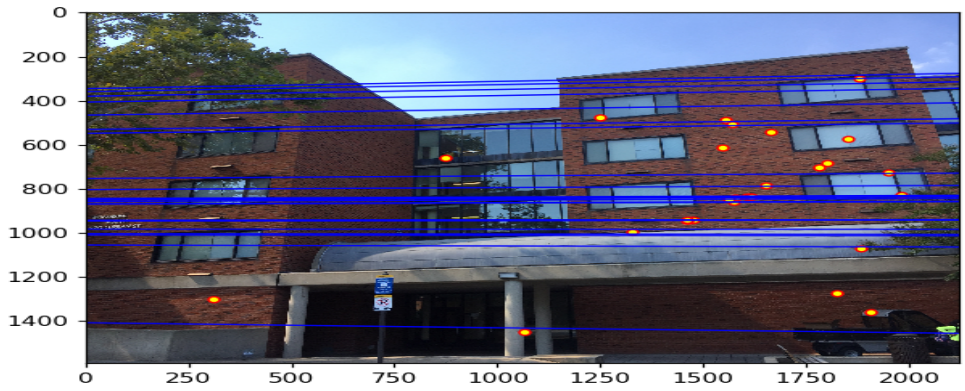


Figure 24: Epipolar Lines in first Image

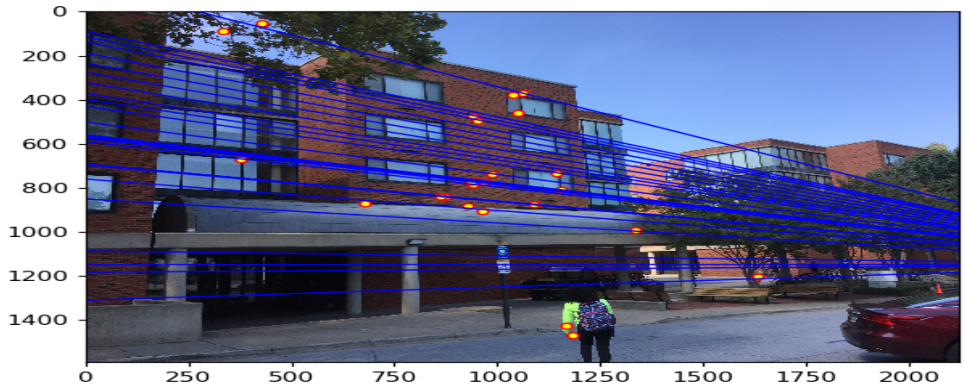


Figure 25: Epipolar Lines in second Image

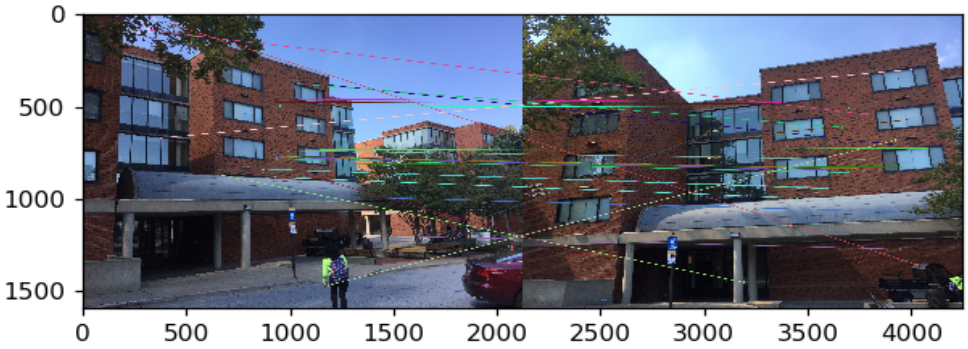


Figure 26: Match Two Images

Normalized:

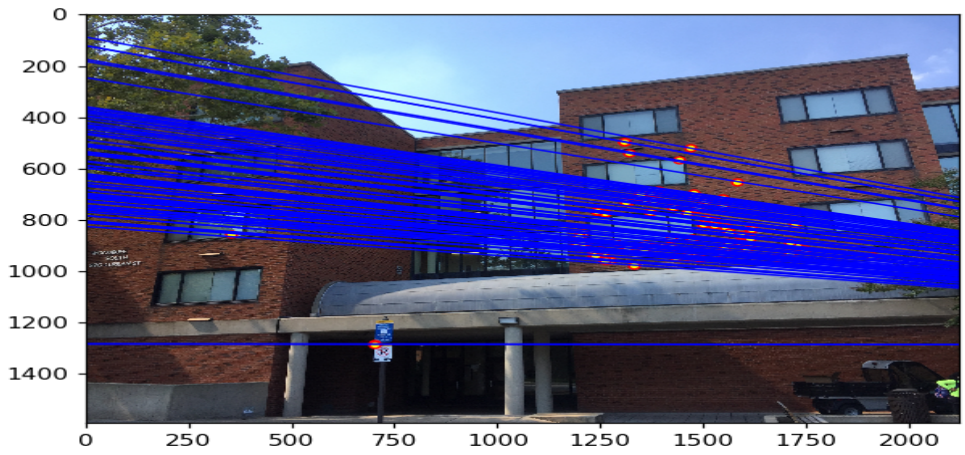


Figure 27: Epipolar Lines in first Image

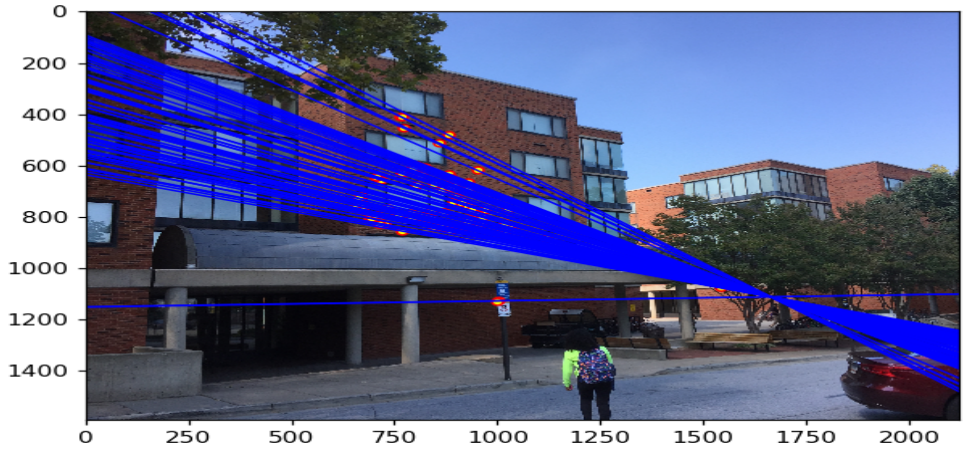


Figure 28: Epipolar Lines in second Image

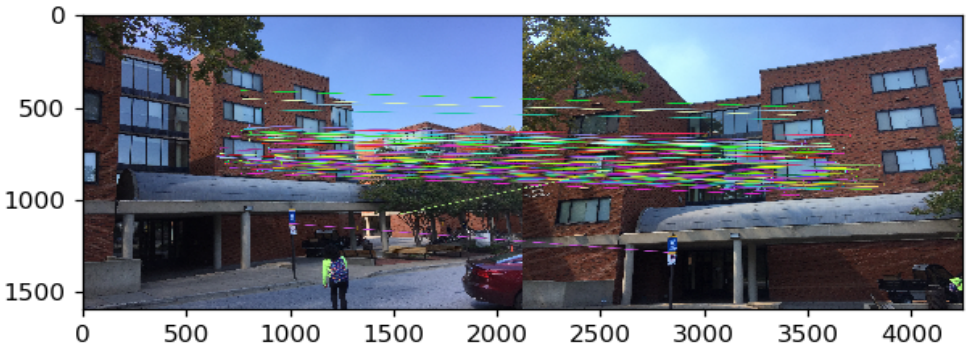


Figure 29: Match Two Images

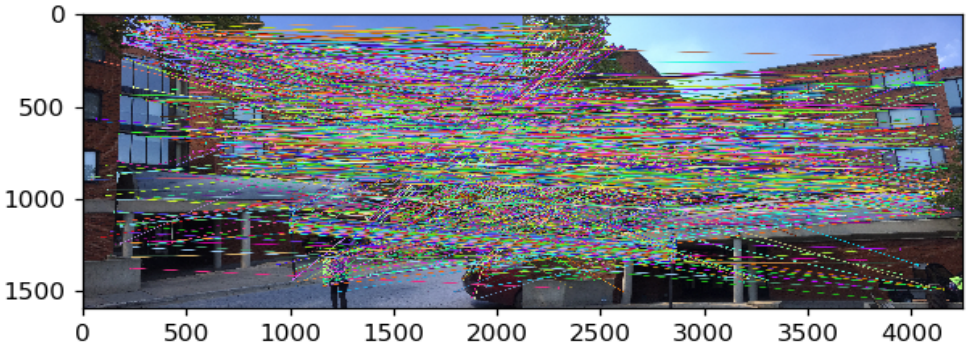


Figure 30: Matching Using ORB

3 CONCLUSION:

We found ways to apply theoritically computable epipolar lines to real world applications. This type of applications can be used for finding any point on an image by performing a linear search, saving time and computational power to process the whole image. RANSAC works pretty fine in most cases when we normalize, but since it is performing randomly, it may not find the best or optimal match everytime. The best way to do this is indeed very compuatationally expensive but, indeed the local minimum of the best points might be similar to the global optimum, just as machine learning problems. So, we can increase this part by further modifying these things for 3D resonstructions and other exciting stuffs like point cloud reconstructions etc. in the future.

4 REFERENCES:

1. $https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj3/html/sdai30/index.html$
2. $https://en.wikipedia.org/wiki/Camera_matrix$