
CAMERA CALIBRATION AND FUNDAMENTAL MATRIX ESTIMATION WITH RANSAC

A PREPRINT

 Jimut Bahan Pal*

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Howrah 711202
jimutbahanpal@yahoo.com

May 15, 2020

ABSTRACT

The main aim of this investigation is to study the camera and scene geometry. We will estimate the camera projection matrix also known as 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 2d and 3d points. To estimate the fundamental matrix, the input is corresponding 2d points across the two images. We started out by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we moved on to estimating the fundamental matrix using point correspondences from ORB, which is an alternative to SIFT. We used RANSAC to find the fundamental matrix with the most inliers, we filtered away spurious matches and achieved near perfect point to point matching.

Keywords Epipoles · Camera Calibration · Fundamental matrix · Projection · RANSAC

1 Camera Projection Matrix

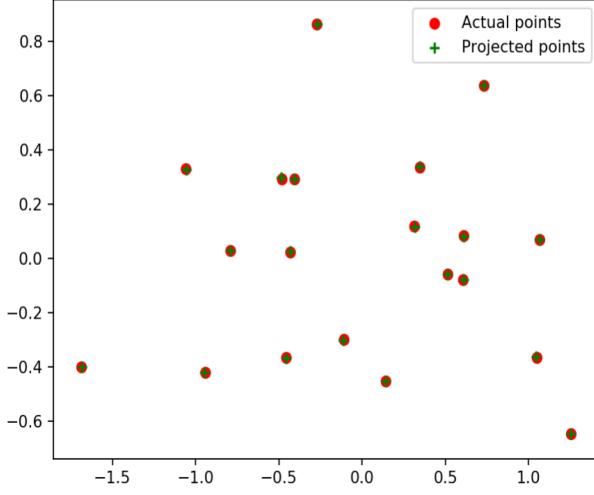
A camera makes a mapping between the 3D world [1] and a 2D image. It projects the 3D world to 2D. The equation of the homogeneous image (3x1), camera matrix (3x4) and a homogeneous world point (4x1) is written as,

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \cong \begin{pmatrix} u * s \\ v * s \\ s \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

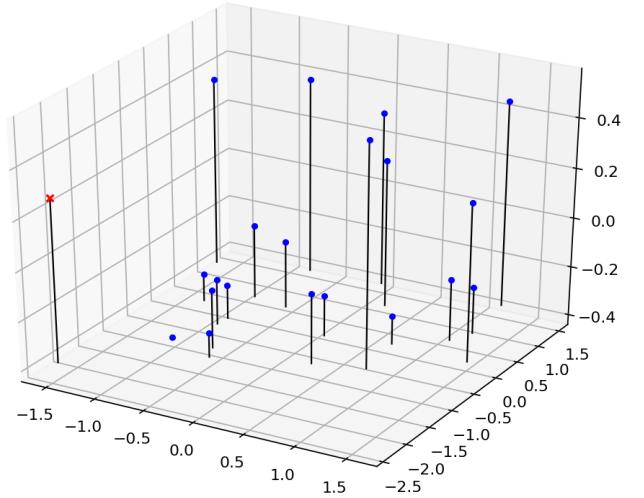
Another way of writing this equation is:

$$\begin{aligned} u &= \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ \rightarrow (m_{31}X + m_{32}Y + m_{33}Z + m_{34})u &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} \\ \rightarrow 0 &= m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}uX - m_{32}uY - m_{33}uZ - m_{34}u \\ v &= \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ \rightarrow (m_{31}X + m_{32}Y + m_{33}Z + m_{34})v &= m_{21}X + m_{22}Y + m_{23}Z + m_{24} \end{aligned}$$

* Jimut Bahan Pal is an alumni of St. Xavier's College. He is currently pursuing M.Sc. in RKMVERI, Belur. This project was done under the supervision of Tamal Maharaj.

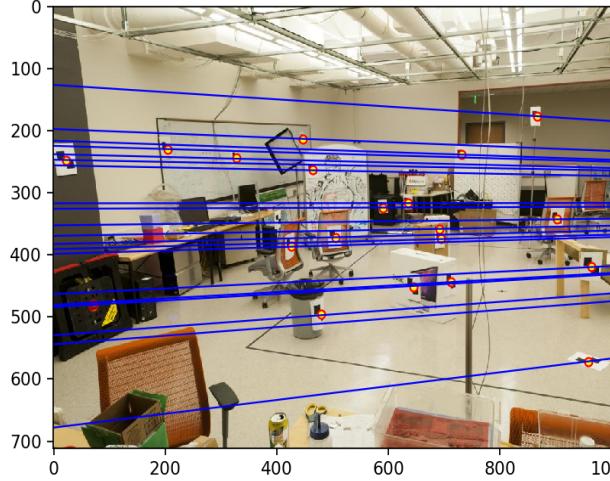


(a) Actual points and the projected points estimated from the projection matrix M .

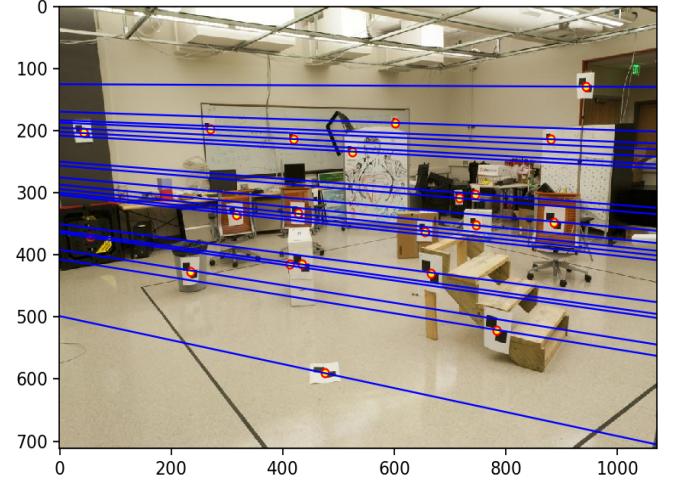


(b) The estimated location of the camera.

Figure 1: Images of Camera Projection matrix calculation resulting in the projected points along with determining the location of the camera.



(a) Epipolar lines drawn on the left image.

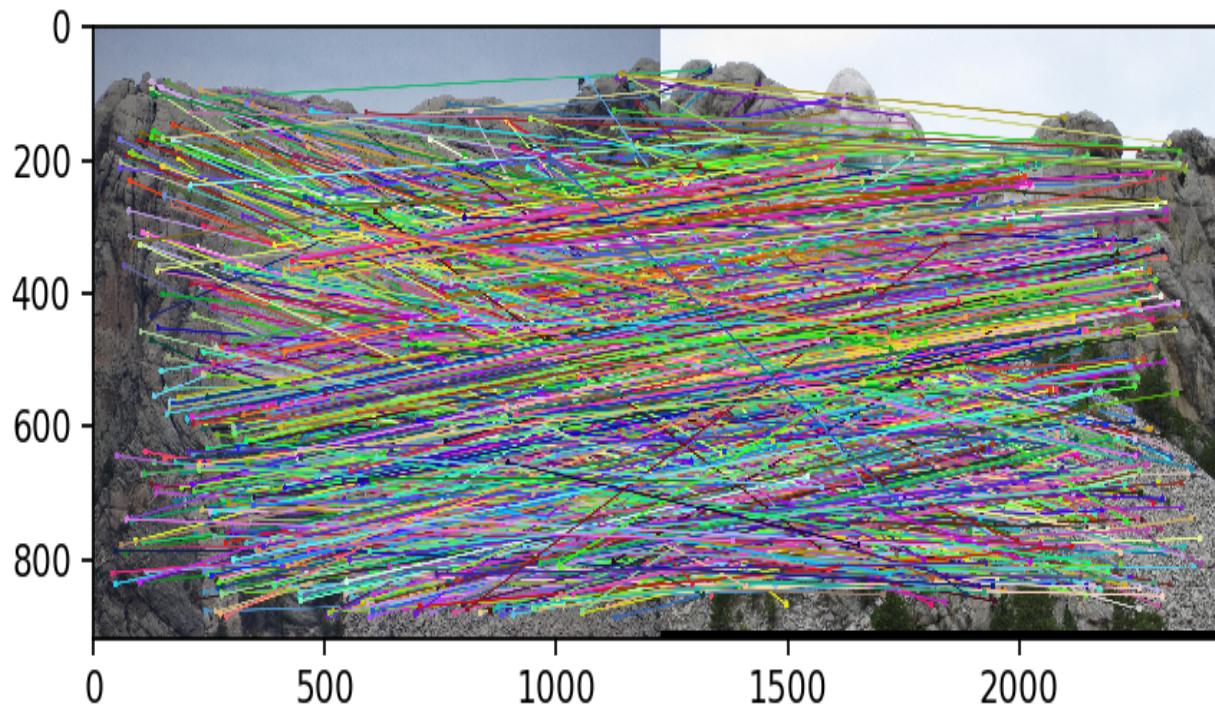


(b) Epipolar lines drawn on the right image.

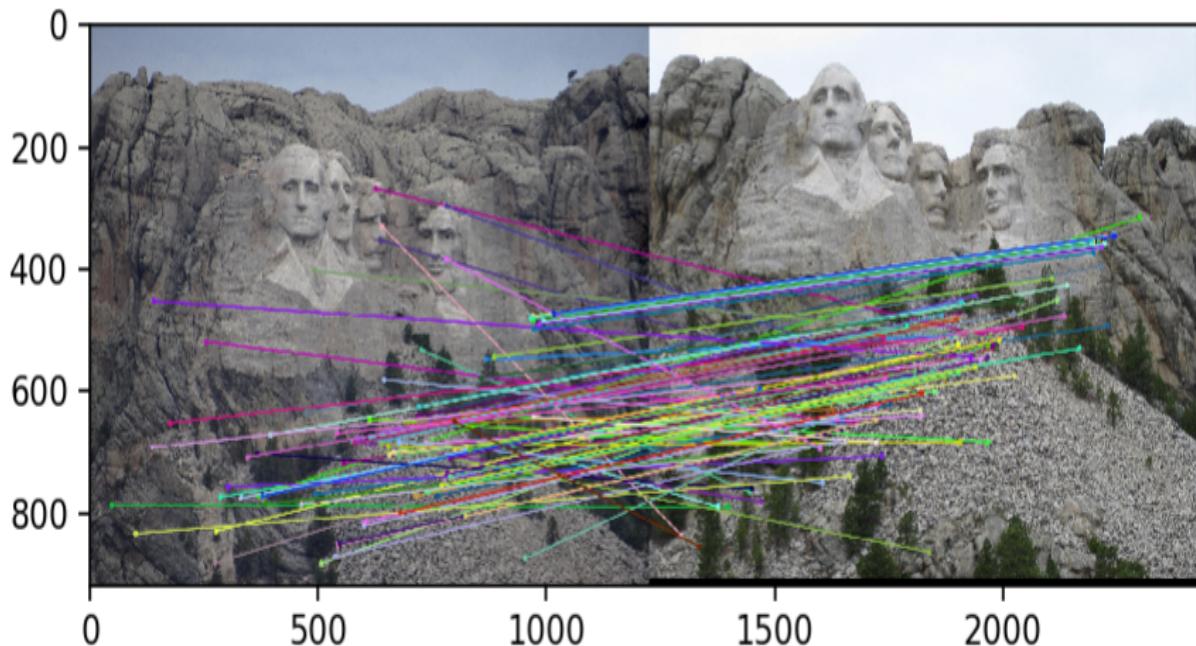
Figure 2: Epipolar lines on two images.

$$\rightarrow 0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}vX - m_{32}vY - m_{33}vZ - m_{34}v$$

We will be able to set up our linear regression to find the elements of the matrix M . The problem with this procedure is that matrix M is only defined up to a scale. In some case it may be that the value of the matrix M will be 0, i.e., all the elements of the matrix M will be 0, which is not very helpful in our context. The way around this is to first fix a scale and do the regression. There are several options for this, we can fix the last element m_{34} to 1 then find the remaining coefficients, or we can use the singular value decomposition to directly solve the constrained optimization problem:

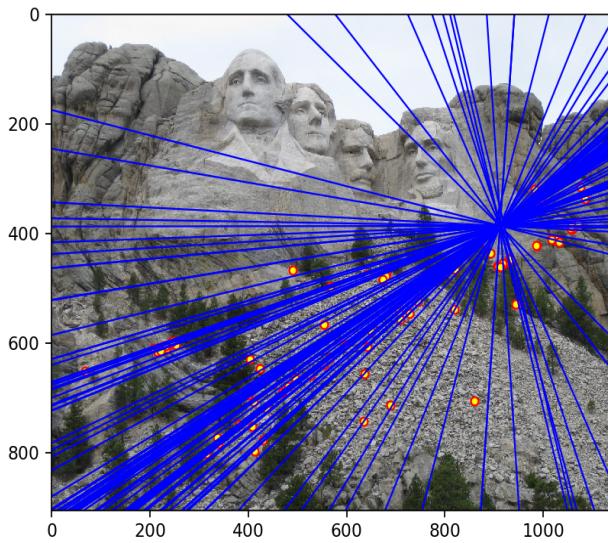


(a) Feature matches by running ORB.

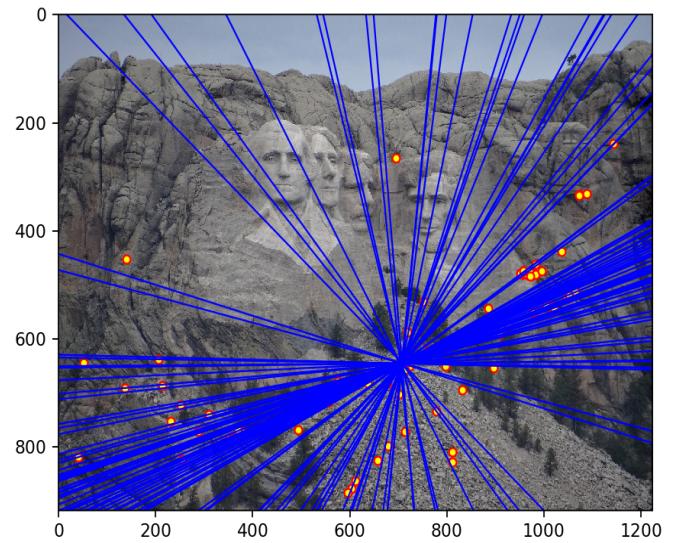


(b) Feature matches by running RANSAC.

Figure 3: Feature matches on both the images of mount Rushmore.

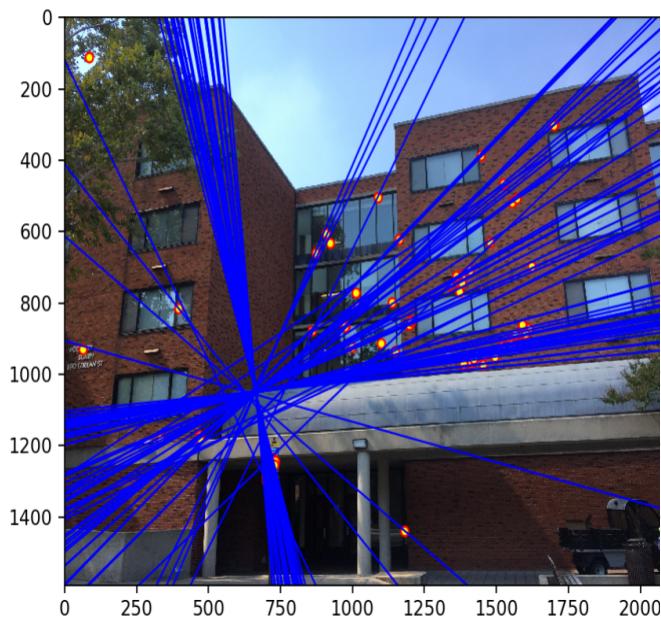


(a) Fundamental matrix estimation for the left image.

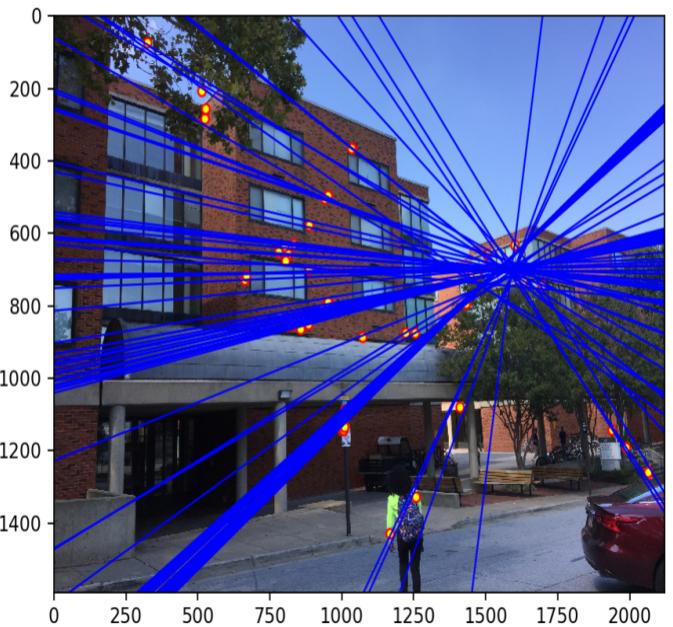


(b) Fundamental matrix estimation for the right image.

Figure 4: Fundamental matrix estimation for mount Rushmore.

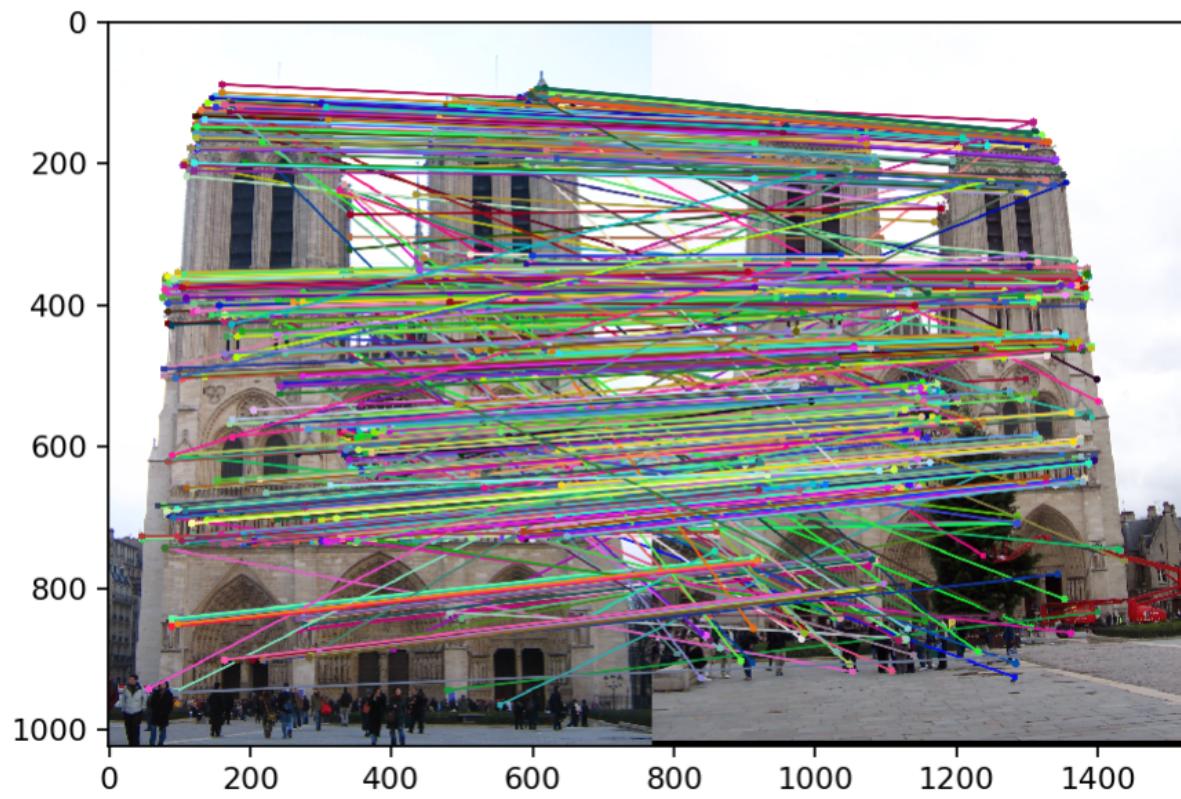


(a) Fundamental matrix estimation for the left image.

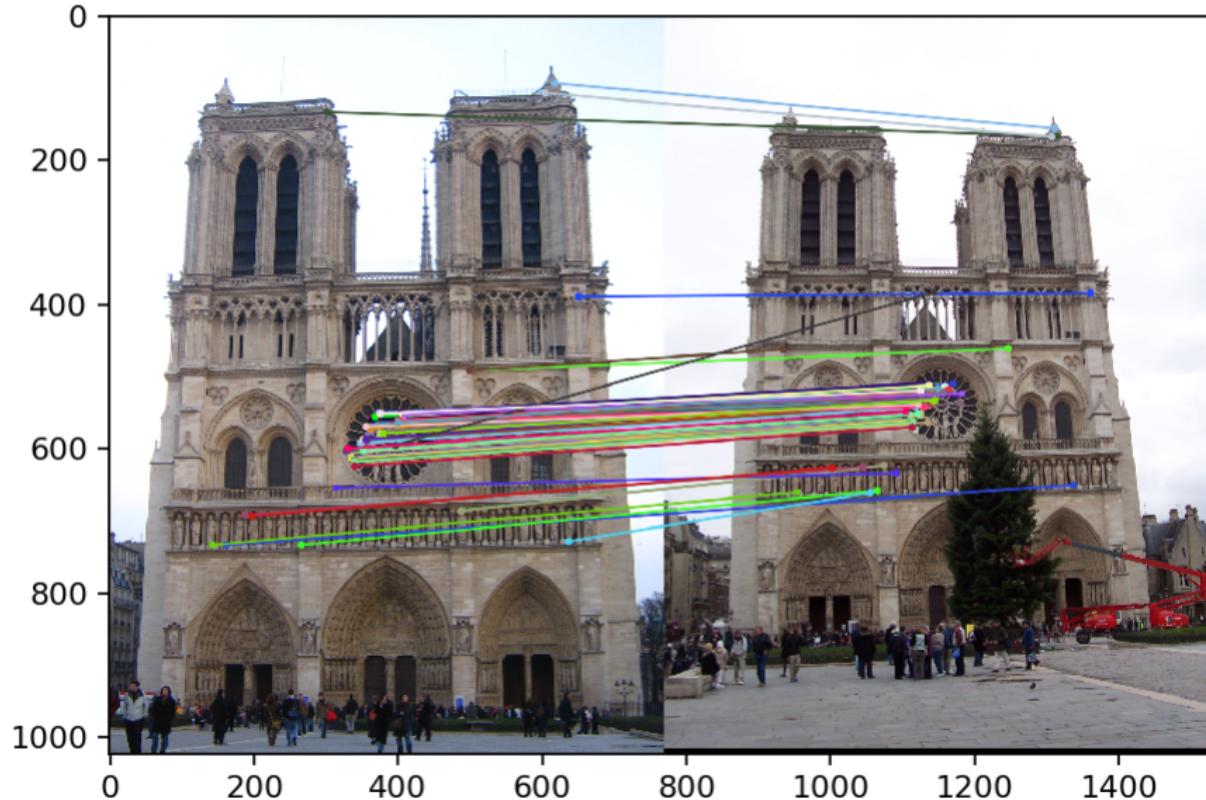


(b) Fundamental matrix estimation for the right image.

Figure 5: Fundamental matrix estimation for Gaudi.



(a) Feature matches by running ORB.



(b) Feature matches by running RANSAC.

Figure 6: Feature matches on both the images of Notre Dame.
5

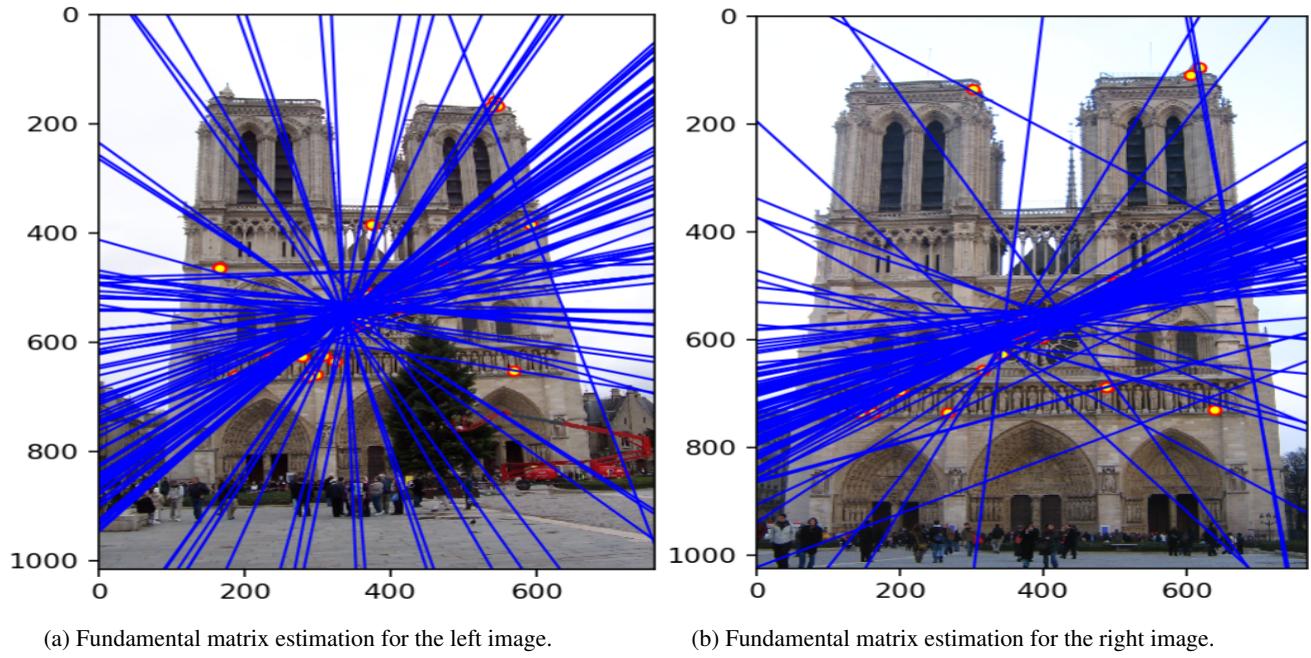


Figure 7: Fundamental matrix estimation for Notre Dame.

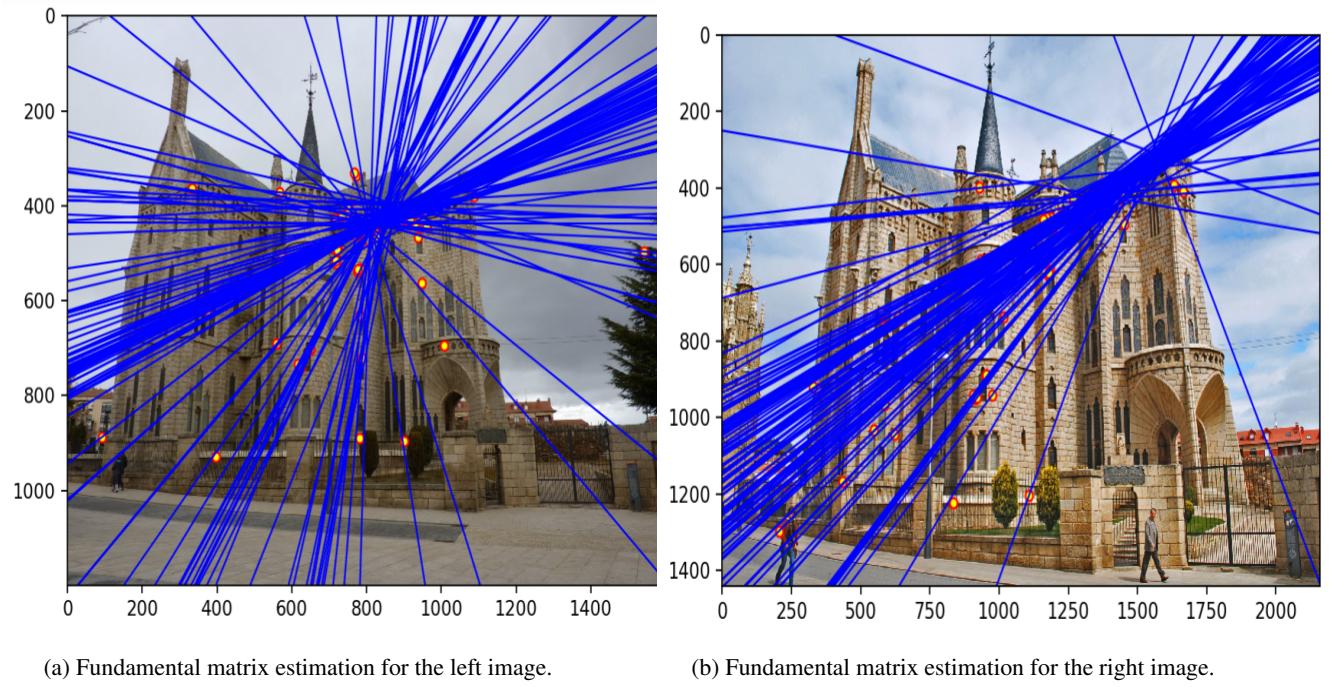
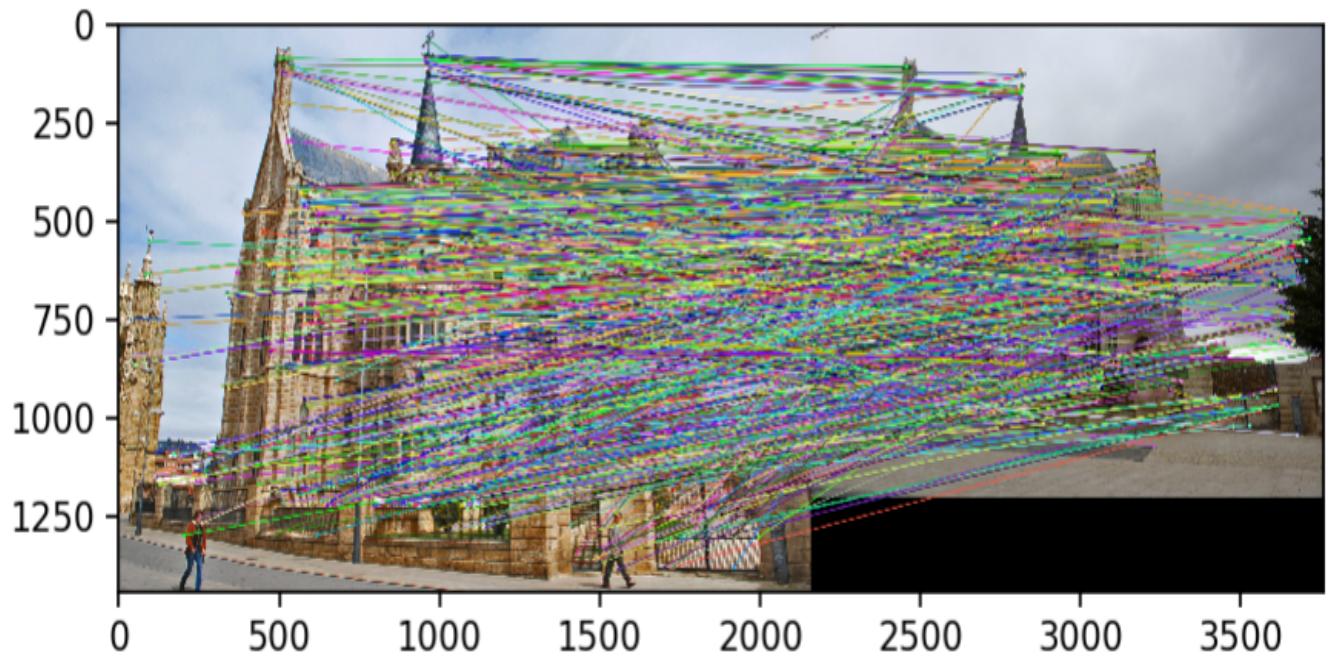
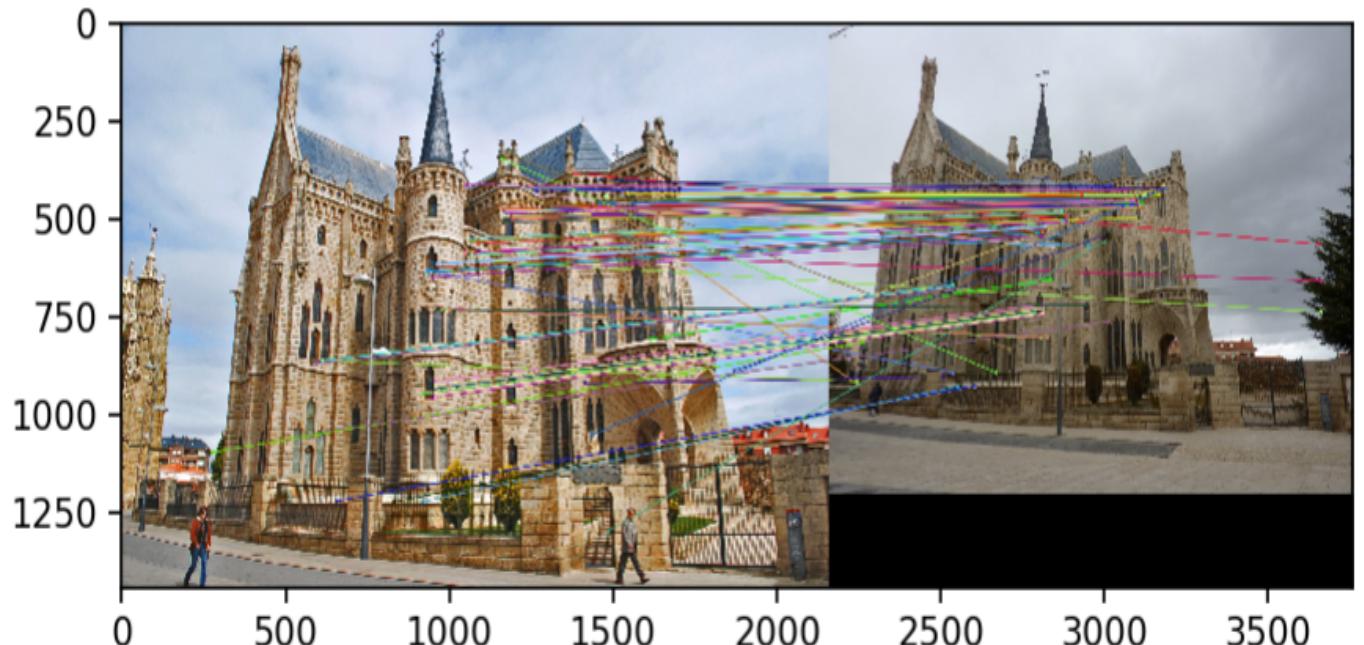


Figure 8: Fundamental matrix estimation for Gaudi.

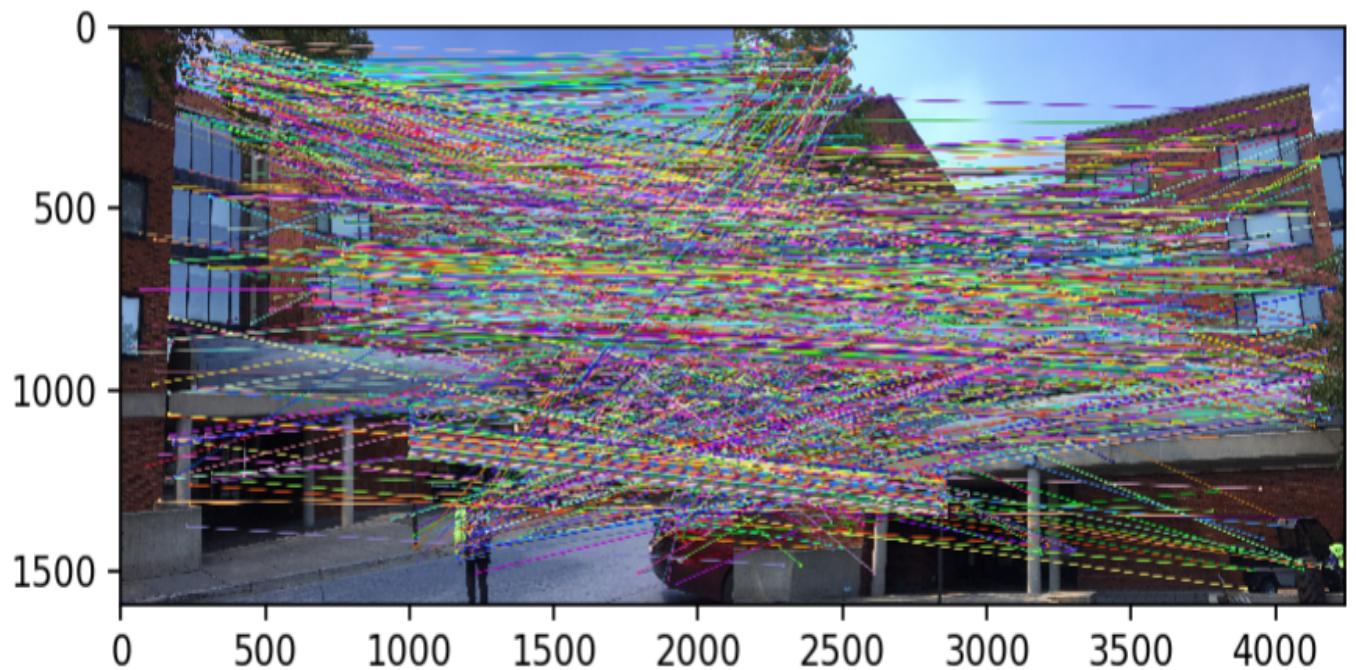


(a) Feature matches by running ORB.

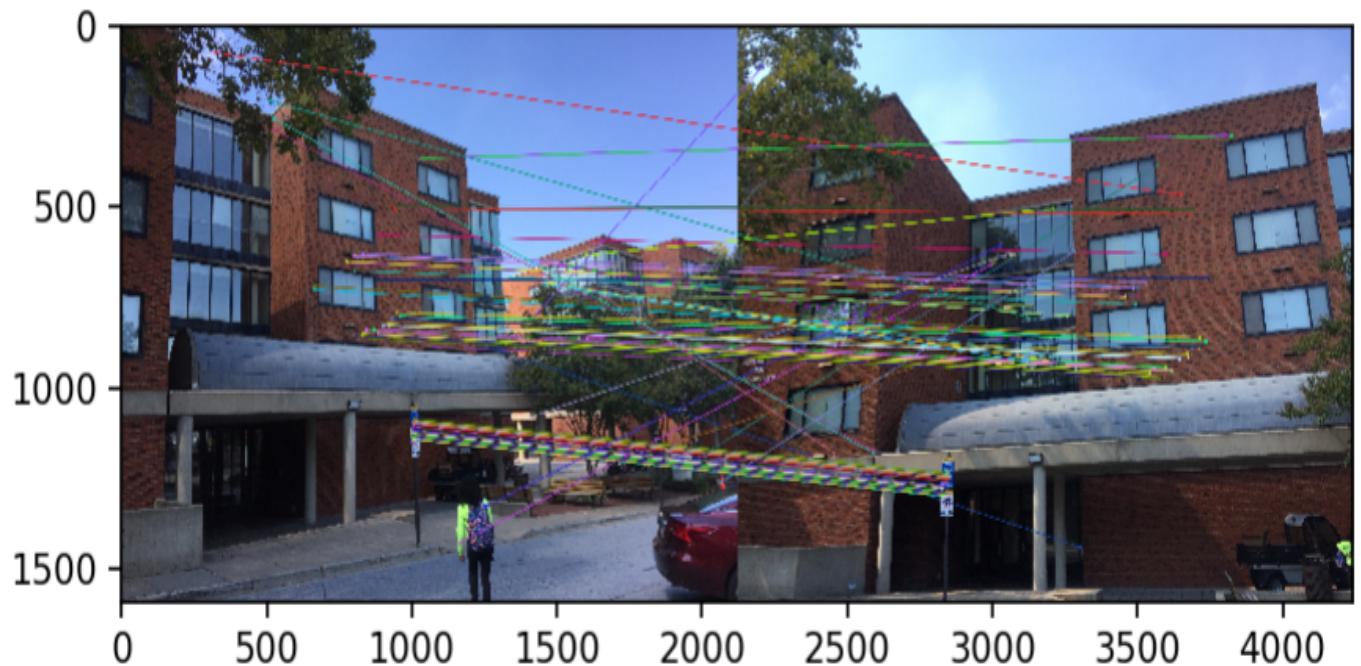


(b) Feature matches by running RANSAC.

Figure 9: Feature matches on both the images of Gaudi.



(a) Feature matches by running ORB.



(b) Feature matches by running RANSAC.

Figure 10: Feature matches on both the images of Gaudi.

$$\min \|Ax\| \text{ such that } \|x\| = 1.$$

For this part, we used the function of the starter code named as *calculate_projection_matrix()* which took two points' list i.e., point's list corresponding to 2D and point's list corresponding to the 3D coordinate system. All we needed to do is we used the least squares solution to solve for M using *np.linalg.inv()* function. We then returned the M matrix by adjusting the parameters accordingly. This is actually done by solving a non-homogeneous system of linear equation, by solving it through least squares, given in the following:

$$\begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & -u_1X_1 - u_1Y_1 - u_1Z_1 - u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 - v_1Y_1 - v_1Z_1 - v_1 \\ \vdots & & & & & & & & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & -u_nX_n - u_nY_n - u_nZ_n - u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n - v_nY_n - v_nZ_n - v_n \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

By doing this we got a projection matrix of :

```
[[ 0.76785834 -0.49384797 -0.02339781  0.00674445]
 [-0.0852134 -0.09146818 -0.90652332 -0.08775678]
 [ 0.18265016  0.29882917 -0.07419242  1.        ]]
```

with a total residual of 0.044535. The actual points and the projected points are shown in Figure 1a. We can further use this step to find the camera center. We can use the equation $C = -Q^{-1}m_4$. Where, we can divide the projection matrix into $M = (Q|m_4)$

The estimated location of camera was found to be coordinate points in 3D in $< -1.5126, -2.3517, 0.2827 >$. The 3D location of all the coordinates is shown in Figure 1b. Here we have used the same function as used before, i.e., *np.linalg.inv()* for finding the inverse of matrix Q.

2 Fundamental Matrix Estimation

We have used a function named *estimate_fundamental_matrix()* to feed in two points, i.e., *points_a* and *points_b*. The first list of points contains the 2D points in image A and the second one contains a list of 2D points in image B. This function returns 3x3 shaped fundamental matrix. We have used the 8 point algorithm for estimating the fundamental matrix. We have used least squares solution using SVD of equations from 8 pairs of correspondences. It can be written as,

$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ \vdots & \vdots \\ u_nu'_n & u_nv'_n & u_n & v_nu'_n & v_nv'_n & v_n & u'_n & v'_n & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{pmatrix} = 0$$

We have used *np.linalg.svd()* to solve for F. The results of the images are shown in Figure 2a and Figure 2b.

3 Fundamental Matrix Estimation with RANSAC

We found the best fundamental matrix estimation using RANSAC on potentially matching points. Our RANSAC loop contains call to the previous function *estimate_fundamental_matrix()*. For uncluttered visualization of points, we needed to return more than 100 points for either left or right images. We did the investigations on the following data:

3.1 Mount Rushmore

This pair was easy and the most of the initial matches are correct. There were 1214 features captured by ORB as shown in Figure 3a. The ransac estimation reduced the number of features as shown in Figure 3b. The epipolar lines that were found are shown in Figure 4a and 4b.

3.2 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. There were 1259 features captured by ORB as shown in Figure 6a. The ransac estimation reduced the number of features as shown in Figure 6b. The epipolar lines that were found are shown in Figure 7a and 7b.

3.3 Gaudi

This pair is difficult and doesn't find many correct matches unless we 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. There were 1031 features captured by ORB as shown in Figure 9a. The ransac estimation reduced the number of features as shown in Figure 9b. The epipolar lines that were found are shown in Figure 8a and 8b.

3.4 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 we got epipolar lines qualitatively similar to part 2 of the project. There were 1160 features captured by ORB as shown in Figure 10a. The ransac estimation reduced the number of features as shown in Figure 10b. The epipolar lines that were found are shown in Figure 5a and 5b.

4 Conclusions

We found ways to apply theoretically 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, 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 computationally 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 reconstructions and other exciting stuffs like point cloud reconstructions etc. in the future.

5 Acknowledgements

It is ritual that scholars express their gratitude to their supervisors. This acknowledgement is very special to me to express my deepest sense of gratitude and pay respect to my supervisor, Tamal Maharaj, Department of Computer Science, for his constant encouragement, guidance, supervision, and support throughout the completion of my project. His close scrutiny, constructive criticism, and intellectual insight has immensely helped me in every stage of my work.

I'm grateful to my father, Dr. Jadab Kumar Pal, Deputy Chief Executive, Indian Statistical Institute, Kolkata for constantly motivating and supporting me to develop this documentation along with the application. Finally, I acknowledge the help received from all my friends and well-wishers whose constant motivation has promoted the completion of this project.

References

- [1] Jimut Bahan Pal. A deeper look into hybrid images. <https://arxiv.org/abs/2001.11302> last accessed on 17.03.2020, 2020.