ENPM673 - Project 3

Naitri Rajyaguru
email: nrajyagu@umd.edu
117361919

I. Calibration

This section deals with estimating fundamental matrix by computing corresponding feature matches in the pair of images. Once, fundamental matrix is obtained, Essential matrix is computed and pose of camera is retrieved.

A. Feature Matching

For feature matching, we first need to locate features. Here, SIFT (Scale Invariant Feature Transform) is used. This detector is scale and rotation invariant. The algorithm is stated below:
1) Convert images into grayscale.
2) Compute SIFT features for both the images.
3) Use Brute Force matcher to match the features. It uses L2 NORM distance for the same.
4) Sort the matches in ascending order of least distance and for this project purpose only 150 matches were used.
5) Inbuilt function of cv2.BFmatcher.match gives list of objects. All points were collected from this object.

- DMatch.distance - Distance between descriptors
- DMatch.trainIdx - Index of the descriptor in image 2
- DMatch.queryIdx - Index of the descriptor in image 1
- DMatch.imgIdx - Index of the image 2

Fig. 1: Feature Matching

B. Fundamental Matrix estimation using RANSAC

We use 8 point algorithm to estimate fundamental matrix. The algorithm is stated as follows:
1) From the matches that are computed, 8 pairs are taken to estimate initial fundamental matrix.
2) Before computing initial F matrix, we normalize image coordinates and compute is done using equations below:

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} *$$

$$\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

3)All of the pairs from both the images are tested against this fundamental matrix $x' * F * x^T$ and error was computed.
4)If the error is below some threshold value then that point pair will be inlier and corresponding Fundamental matrix will be considered as good.
5) The above steps were repeated for 1000 iterations and outliers were discarded.
The fundamental matrix for dataset 3 is:

$$\begin{bmatrix} [\ 2.28956446\text{e}-09 & -4.55839520\text{e}-07 & 3.26177983\text{e}-04] \\ [\ 4.40929723\text{e}-07 & -9.98230781\text{e}-08 & 1.90571349\text{e}-03\ ] \\ [-3.06562032\text{e}-04 & -1.79617428\text{e}-03 & -2.90746697\text{e}-02]] \end{bmatrix}$$

C. Estimation of Essential Matrix

If we have calibrated cameras, we can get essential matrix from which we can easily estimate pose. The Essential Matrix can be estimated by:

$$E = K^T F K$$

The essential matrix for dataset 3 is:

$$\begin{bmatrix} [\ -0.00601983 & -0.42124979 & 0.02146501\ ] \\ [\ 0.41193744 & - & 0.045455820.90920216] \\ [\ -0.05214487 & -0.90516315 & -0.04006178\ ] \end{bmatrix}$$

D. Estimation of Camera Pose

The Essential matrix is composed of Rotation and Translation of camera. If we decompose Essential matrix using singular value decomposition then we get four camera pose configurations. The pose of the camera is:
Rotation matrix:

$$\begin{bmatrix} [\ 0.64972432 & -0.04794257 & -0.75865659\ ] \\ [\ -0.08611159 & -0.99622705 & -0.01079158\ ] \end{bmatrix}$$

$$\begin{bmatrix} -0.75527684 & 0.07234068 & -0.65140135 \end{bmatrix}$$

Translation:

$C_1 = U(:,3)$ and $R_1 = UWV^T$

$C_2 = -U(:,3)$ and $R_2 = UWV^T$

$C_3 = U(:,3)$ and $R_3 = UW^TV^T$

$C_4 = -U(:,3)$ and $R_4 = UW^TV^T$

Trinagulation check for Cheirality condition

$$r_3(\mathbf{X} - \mathbf{C}) > 0$$

and $z > 0$

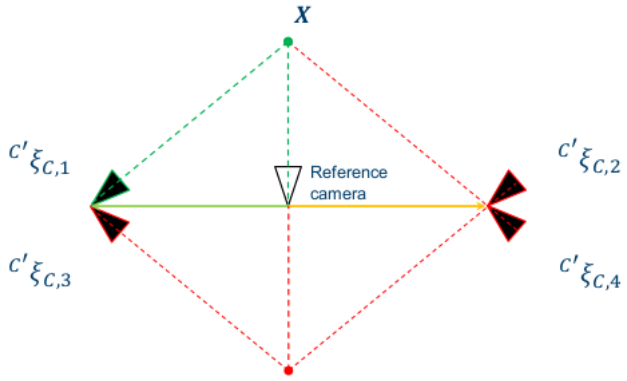

Fig. 2: Four Camera Poses

II. Data

III. Rectification

To compute disparity, we need to have both the images in the same plane so that rotation does not come in to picture. To accomplish this, we will rectify the image pairs and also verify that they are coplanar by drawing epipolar lines. If epipolar lines are parallel then they are considered as coplanar. Inbuilt function of cv2.stereoRectifyUncalibrated() is used to get homography matrices of both the images.

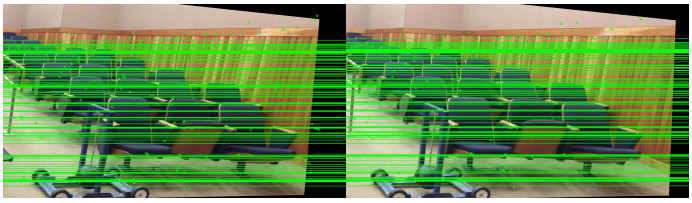

Fig. 3: Parallel Epipolar lines for dataset3

IV. Data

V. Correspondence

The problem of correspondence occurs when images are taken from different point of view. To compute dispairty, we need all the points in the left image to be rightly correlated to the right image. The rectified images are available, and hence for earch corresponding pixel on the epipolar line, we will search for the same in another image. Here, block matching algorithm is used where a fixed window is selected and slided over in the other image epipolar line and Sum of Sqaured Distances is computed. The index of the iwndow with minimum SSD is taken into account for disparity. I have used two methods here. One is for every window, searching for a particular range in another image and computing disparity. This method is extremely slow and takes around 40 minutes for computation and disparity map generation. Another method is to compute all window patches of one row in left and right images. Then take each patch of left image and compare it by taking SSD with all patches in the right. This takes 10 minutes to compute. Results are not very different.

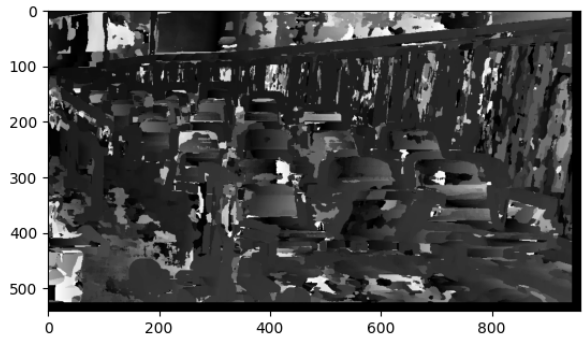

Fig. 4: Disparity Map for dataset3

VI. Compute Depth Maps

As we have disparity map, we can compute depth using the given formula:

$$z = \frac{\text{baseline} * f}{disparity}$$

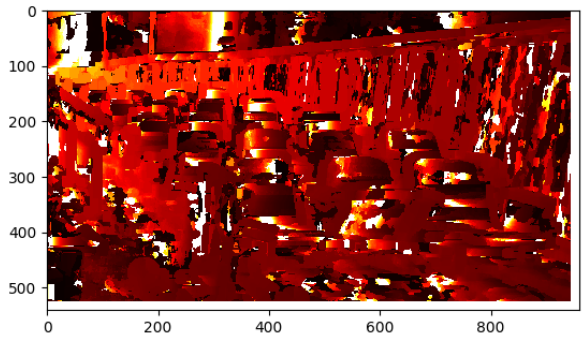

Fig. 5: Depth Map for dataset3

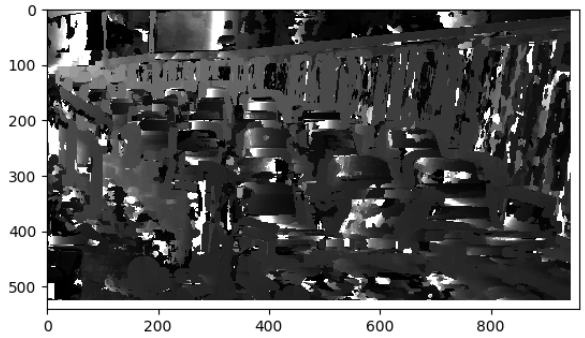

Fig. 6: Depth Map for dataset3

VII. Results

Dataset1:

Estimated F:

[[-5.81987686e-10 2.18455949e-08 1.41745046e-04]
[-2.99667784e-08 -1.80018681e-08 2.06426493e-03]
[-1.29727568e-04 -2.02951252e-03 -2.00631452e-03]]
Estimated E:

[[-4.45743354e-04 1.97780137e-02 7.56443409e-02]
[-2.51736349e-02 -5.08963923e-03 9.96811586e-01]
[-7.22930294e-02 -9.97165123e-01 -5.38935280e-03]]
Rotation matrix:

[[0.99998203 0.00323523 0.00504663]
[-0.00326058 0.99998207 0.00502348]
[-0.00503029 -0.00503985 0.99997465]]
Translation:

[0.9969386 -0.07554499 0.02015911]

Fig. 7: Results for dataset1

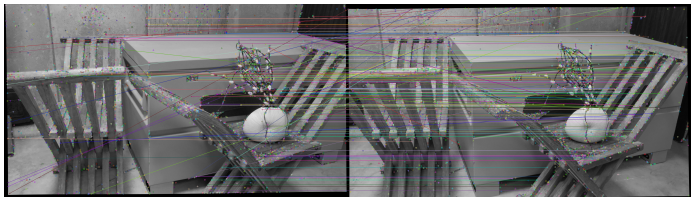

Fig. 8: Feature Matching for dataset1

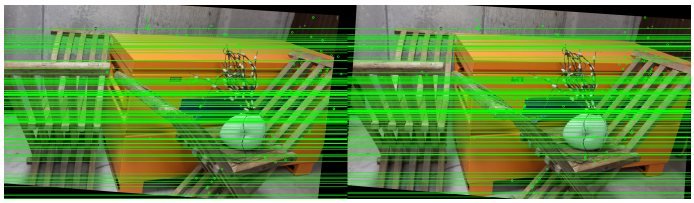

Fig. 9: Parallel epipolar lines for dataset1

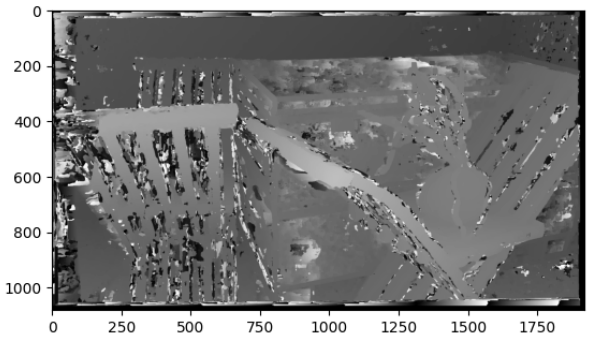

Fig. 10: Disparity map for dataset1

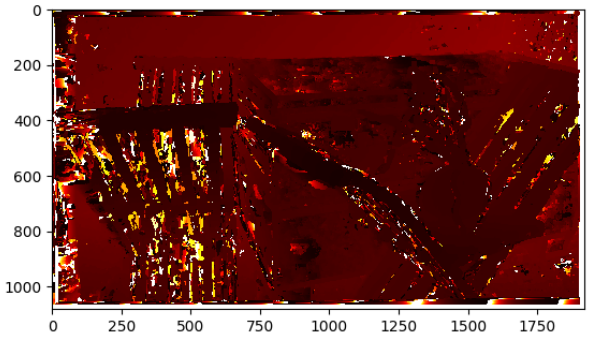

Fig. 11: Depth map for dataset1

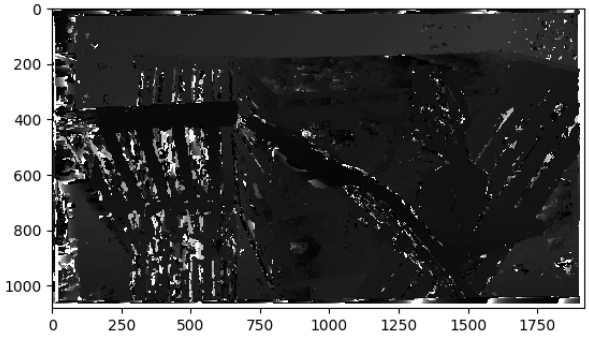

Fig. 12: Depth map for dataset1

Dataset 2:

Estimated F:
[[-2.07103874e-09 1.41010398e-06 -4.79743014e-04]
[-1.39194648e-06 -1.87787941e-07 4.71263279e-03]
[4.75945749e-04 -4.57284207e-03 -1.10317917e-02]]

Estimated E:
[[-0.01359285 0.57068048 0.08401069]
[-0.5668074 -0.03811911 0.82022252]
[-0.04705969 -0.81781476 -0.02359997]]

Rotation matrix:
[[0.99966996 0.02479844 -0.00670895]
[-0.02459075 0.99926382 0.02944668]
[0.00743424 -0.02927198 0.99954384]]

Translation:
[0.81675041 -0.06716636 0.57306845]

Fig. 13: Results for dataset2

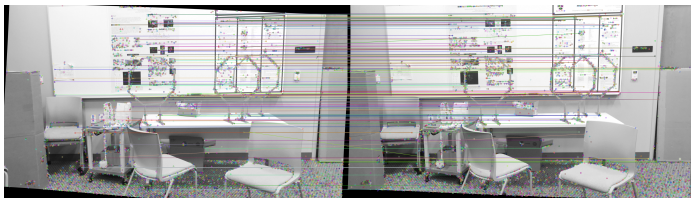

Fig. 14: Feature Matching for dataset2

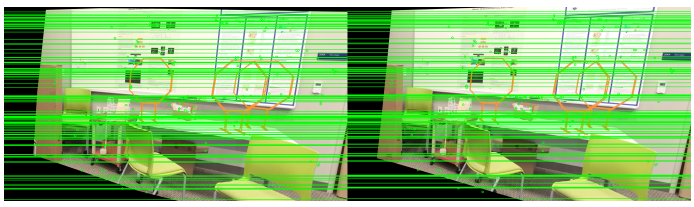

Fig. 15: Parallel epipolar lines for dataset2

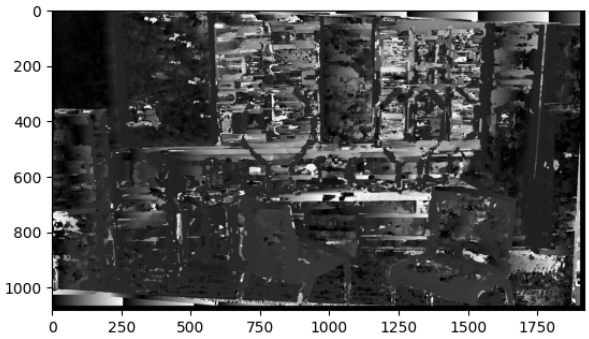

Fig. 16: Disparity map for dataset2

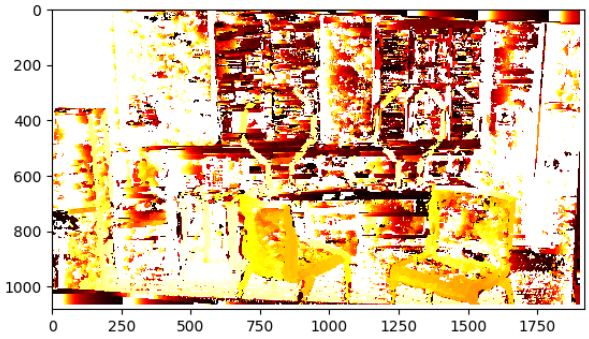

Fig. 17: Depth map for dataset2

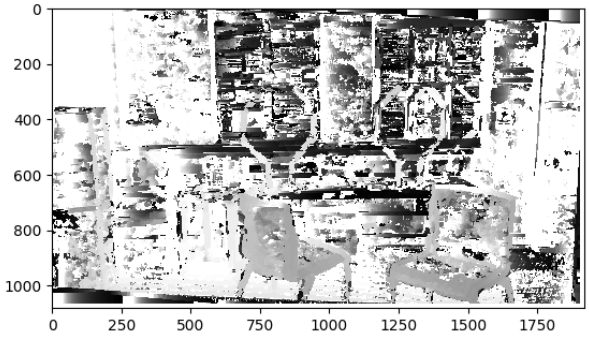

Fig. 18: Depth map for dataset2