Project 2: Visual Odometry.

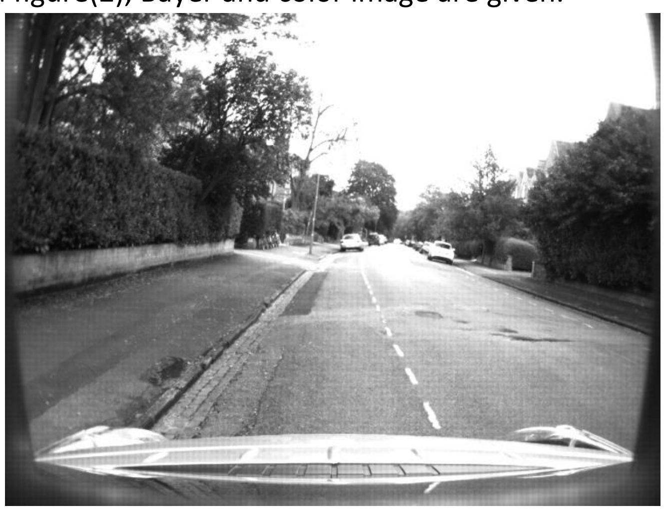
ENPM 673, Robotics Perception

Utsav V Patel

UID – 115816756

Following are the steps that I took to complete this project

1) **Recovering the color images-**
   Here, the given input images are black and white with Bayer format, by applying the demosaic function I recovered the color image. The alignment of the pixels is GBRG. To get the colored images run Demosaic.m script.
   In the figure(1) and figure(2), Bayer and color image are given.



Figure(1) – Bayer Image

Figure(2) Color Image

2) Extracting camera parameters-

In the second step of the project I extracted the camera parameters with the help of ReadCameraModel.m function given in the input dataset. The function extracts the values of fx, fy, cx, cy, G camera image, LUT.

3) Undistorting the images-

In this step, the color images are undistorted with the help of UndistortImage.m function given with the dataset. The input color image and undistorted images are given in the figure(3) and figure(4).

Figure(3) Input color image

Figure(4) – Undistorted Image

4) **<u>Finding point correspondences-</u>**

In the fourth step, for the given two images, corresponding points are found. I used SURF[2] for finding corresponding points. In this step before finding corresponding points with the help of SURF, Gaussian filter[1] is applied so the unwanted high frequency noise can be removed. Application of Gaussian filter prevents the detection of unwanted corners. The found corresponding points are shown in figure(5).

Fig(5) – Image with corresponding points

5) **<u>Estimating Fundamental Matrix-</u>**

In this step fundamental matrix is estimated from the corresponding points found in the previous step. I used normalized 8 points[3] method for finding the fundamental matrix. The function for finding fundamental matrix is defined as FindFundamentalMatrix.m. But before estimating the fundamental matrix from the all corresponding points it is necessary to remove unwanted points and normalize the points to make our results robust.

1) Normalization of points-
   In this process first, the centroid of all the points are found and then by appropriate translation centroid is translated to the origin. In the next step, mean distance of all the points from the origin is found. Now by applying proper scaling factor to the points the mean distance of all the points from the origin is converted to 2^(1/2).
2) RANSAC-
   After applying the normalization operation, from all the normalized corresponding points 8 inlier points which satisfies the fundamental matrix are found. In this process randomly 8 points are selected from the set of corresponding points and the for each 8 points how much the fundamental matrix is satisfied is decided. The best 8 points are selected as the inliers.

   After estimating the fundamental matrix, it is denormalized. The equation for denormalization is as following.
   F = T2' * Fnorm * T1
   T2 = Transformation matrix for normalizing points in right image
   T1 = Transformation matrix for normalizing points in left image
   Fnorm = Normalized fundamental matrix

6) **<u>Estimating Essential Matrix-</u>**
   Now with the help of camera parameters found in step(2) and fundamental matrix in step(5), essential matrix is determined. The equation for the essential matrix is given below.

   E = P' * F * P

   P = Camera Parameter matrix

   F = Fundamental matrix

7) **<u>Extracting R and T from the Essential Matrix-</u>**
   The value of R and T are found by singular value decomposition of the essential matrix. There will be 4 possible solutions for R and T pair. The correct R and T pair is found by performing the triangulation[5] of the points. The four possible solutions can be permuted from the following equations.

```
[U, D, V] = svd(E)
```

$$W = [0 -1\ 0, 1\ 0\ 0, 0\ 0\ 1]$$

$$R1 = U\ W\ V'$$

$$R2 = U\ W'\ V'$$

$$t1 = (:,3)$$

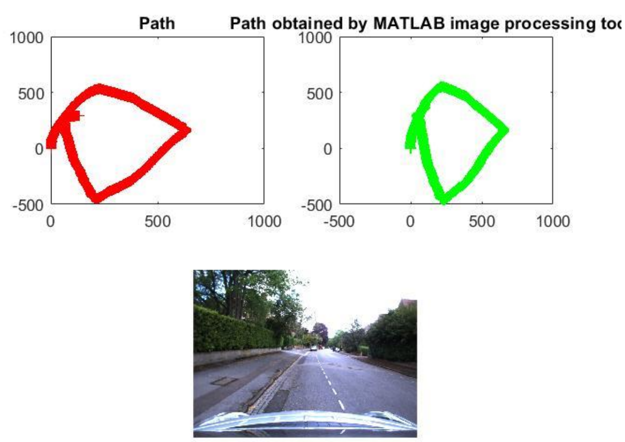
$$t2 = -(:,3)$$

8) <u>Plotting the camera center-</u>

After finding the R and T of the camera. The camera center is obtained and the plotted. Here the trajectory can be found without knowing the exact scale. It is only possible to know the trajectory with the correct scale by integrating other devices like wheel odometer.

9) <u>Extra Credit</u>

Here, I have also found the trajectory of the camera by using MATLAB inbuilt image processing tools. The trajectory with green line is obtained by implementing matlab image processing tool box and the red line trajectory is obtained by customized function. The comparison for both of the trajectories can be seen in the figure(6). It can be seen that the turns detected by both of the trajectories are the same.



Figure(6) Comparison of two outputs

References

1) https://www.mathworks.com/help/images/ref/imgaussfilt.html
2) https://www.mathworks.com/help/vision/ref/detectsurffeatures.html
3) https://www.youtube.com/watch?v=1X93H_0_W5k
4) http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/f07/proj_final/www/amichals/fundamental.pdf
5) Multiple view geometry in computer vision by Richard Hartley and Andrew Zisserman
6) http://cvgl.stanford.edu/teaching/cs231a_winter1415/lecture/lecture10_detector_descriptors_2015.pdf