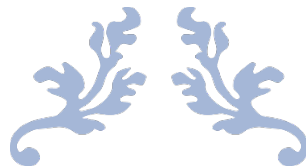




UNIVERSITY OF
MARYLAND



PROJECT 2

VISUAL ODOMETRY

ENPM673 Perception for Autonomous Robots



MARCH 28, 2018

115617018

RajendraMayavan Rajendran Sathyam

Contents

Extraction of camera image	2
Keypoint Correspondence	2
Estimation of Fundamental Matrix	3
Recovering the Rotation and Translation Matrices	4
Plotting the motion of the car	5
Visual Odometry with MATLAB's Computer Vision Toolbox	6

Extraction of camera image

The image is first processed to de-mosaic and undistort the image using the provided function. The undistortion lookup table is provided in along with the Oxford dataset image files in the model folder.

```
fx, fy, cx, cy, G_camera_image, LUT] =  
ReadCameraModel(location_of_images, location_of_LUT);  
  
I = imread(filename);  
J = demosaic(I, 'gbgr');  
undistorted_J = UndistortImage(J, LUT);
```



Figure 1 Undistorted frame

Keypoint Correspondence

In order to find the movement of the camera with two frames, the feature points have to be extracted and they have to be matched with the same feature in the next image. The feature detection algorithm used is SURF. Computer vision toolbox function, `detectSURFFeatures` was used to detect the features for each frame. And the `matchFeatures` function was used to match the features extracted from the two frames.



Figure 2 Matched features

```

I1 = rgb2gray(undistorted_I1);
points1 = detectSURFFeatures(I1);

I2 = rgb2gray(undistorted_I2);
points2 = detectSURFFeatures(I2);

[f1,vpts1] = extractFeatures(I1,points1);
[f2,vpts2] = extractFeatures(I2,points2);

indexPairs = matchFeatures(f1,f2) ;
matchedPoints1 = vpts1(indexPairs(:,1));
matchedPoints2 = vpts2(indexPairs(:,2));

```

Estimation of Fundamental Matrix

The fundamental matrix is the matrix which gives the transformation of the matched points from one frame to another. To estimate the fundamental matrix, 8 - point algorithm was used. Since all the images are of equal size and resolution, normalization is not required.

A matrix A is form with the 8 points from the equation $x^T F x = 0$. The eigen vector of the smallest eigen value of the matrix gives the elements of the 3x3 fundamental matrix in order.

Since the fundamental matrix has a rank restriction of 2, the rank of the F matrix is reduced to 2 by Singular Value Decomposition using the following algorithm.

```
[U,D,V] = svd(Ftemp);
D(3,3)=0;
Ftemp=U*D*V';
```

Since the fundamental matrix is computed with 8 random points taken from the matched points, it is not always accurate as it takes the outliers sometimes. Outliers are the noise in the data which to be eliminated. In order to reduce the effect of imperfection in the feature matching. The whole process is done a 1000 iteration and the fundamental matrix with the least error is taken as the fundamental matrix. This whole process is done by the function findEssentialMatrix.

Recovering the Rotation and Translation Matrices

The essential matrix can be calculated from the fundamental matrix using the equation, $E=K' * F * K$. Where K is the calibration matrix of the camera used to take the picture. This data is given in the model folder in the dataset. The essential matrix $E = [t]_{\times} R$. The matrix T and R are computed using the following algorithm. The calculation of R and T from essential matrix is done by the function findTandR.

```
% Calculating R and T from essential matrix
[U,~,V] = svd(E);

W = [0 -1 0;
     1 0 0;
     0 0 1];

R1 = U*W'*V';
R2 = U*W*V';

if (det(R1) < 0)
    R1=-R1;
end

if (det(R2) < 0)
    R2=-R2;
end

if ((R1(1,1)>0)&&(R1(2,2)>0)&&(R1(3,3)>0))
    R=R1;
else
    R=R2;
end

T = U(:,3);
```

Plotting the motion of the car

Using the values obtained from the essential matrix, the plot for the motion of the car is plotted. But because the fundamental matrix is computed using some points taken in random, the plot is not always the same.



Figure 3 In this figure the last turn was not computed properly and the odometry thinks the car turned around 180 degrees

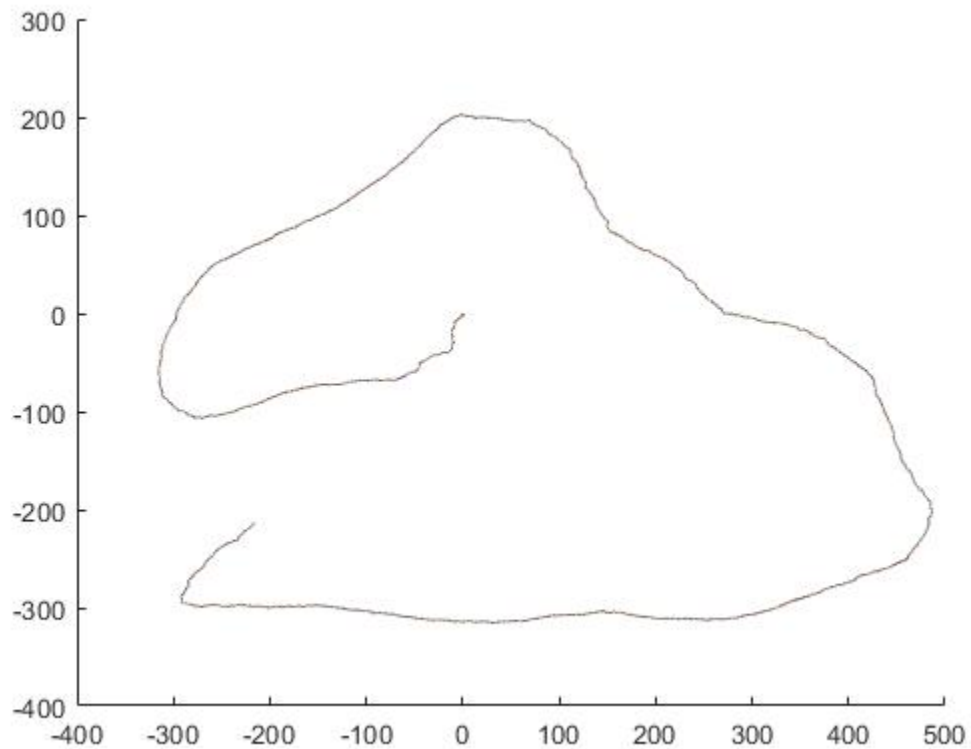


Figure 4 The last turn was again miscalculated but is better than the first iteration.

Visual Odometry with MATLAB's Computer Vision Toolbox

In order to compare the result, the same dataset was computed using the computer vision toolbox function, `estimateFundamentalMatrix`. This function implements RANSAC to separate the inliers from the outliers. The result from using the computer vision toolbox function is shown below.

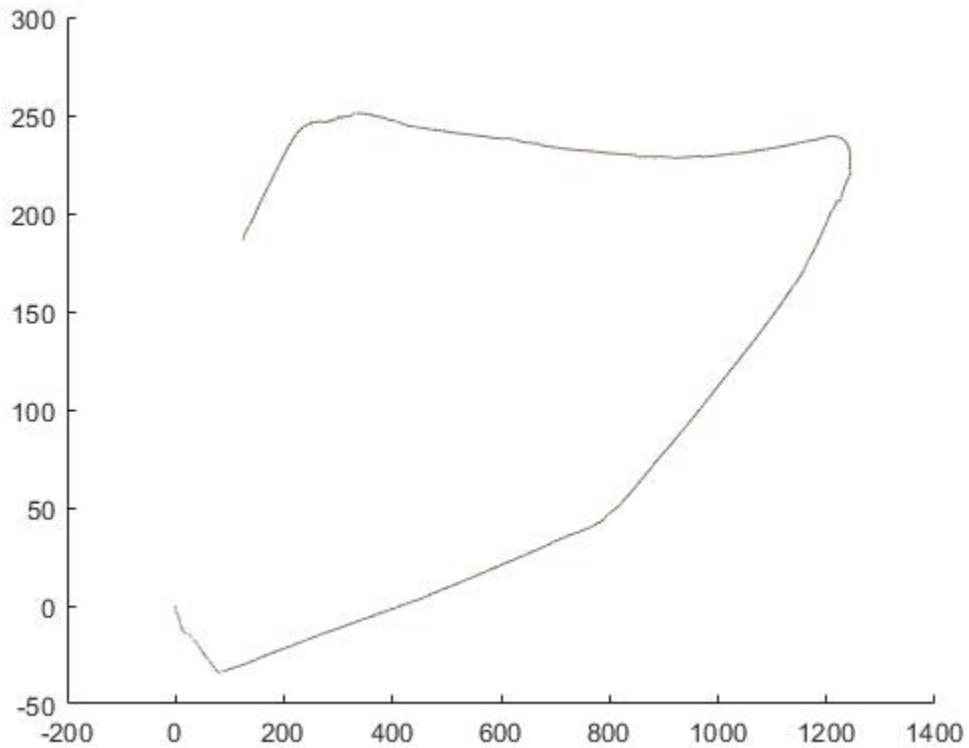


Figure 5 The result from using computer vision toolbox is consistent but not very accurate as well

Instructions on how to run the code is given in the readme file.