
Sensation and percetion

Assignement 2

Sami Sellami

September 30, 2018



THE SOURCE CODE IS ATTACHED TO THE PRESENT FILE

TASK 1: Case 4 Estimating proper trajectory using kalman filter

After uploading the dataset we can plot our dataset:

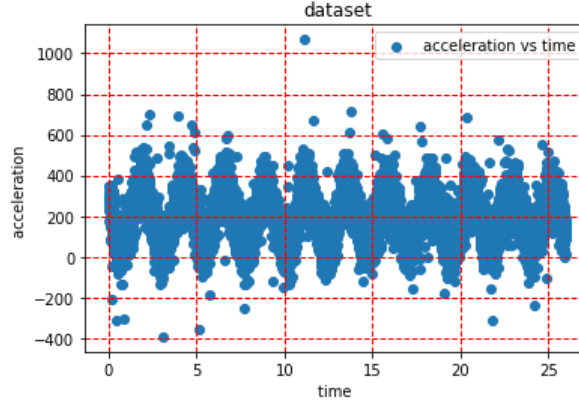


Figure 1: Acceleration in respect of time

For estimating the proper trajectory we implement a code in python for filtering our data using kalman filter. This is the code

```
def kalman (data):  
    x=data_frame.y  
    sigmaEta=x.std()  
    sigmaPsi=1  
    x_opt=[]  
    x_opt.append(x[0])  
    e_opt=[]  
    e_opt.append(sigmaEta)  
    K=[]  
    K.append(1)  
    for t in range (x.size-1):  
        e_opt.append(math.sqrt( (sigmaEta**2 *(e_opt[t]**2 +sigmaPsi**2) )  
        / (sigmaEta**2 + e_opt[t]**2 + sigmaPsi**2) ) )  
        K.append(e_opt[t+1]**2/sigmaEta**2)  
        x_opt.append(K[t+1]*x[t+1] + (1-K[t+1])*x_opt[t])  
    return x_opt
```

And we obtain the following graph after filtering:

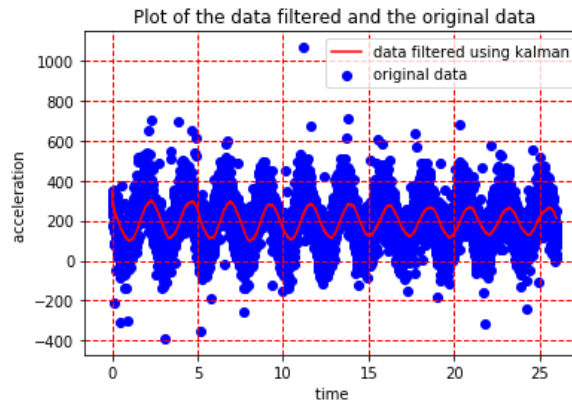


Figure 2: Acceleration in respect of time after using kalman filter

TASK 2: Camera Calibration

We take 30 pictures of our chess board and we compute the algorithm for camera calibration in PYTHON:

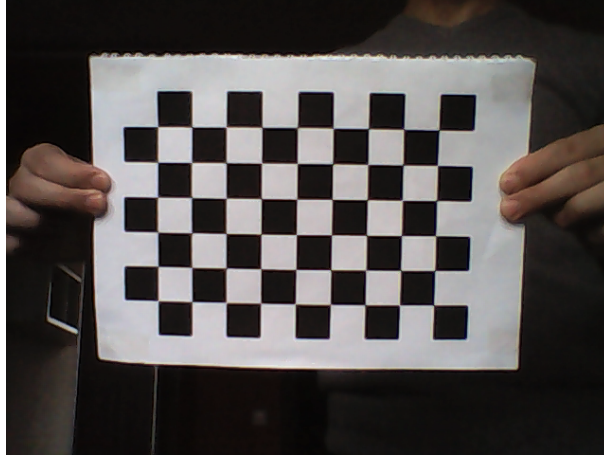


Figure 3: Image of the ChessBoard

we need at least 10 test patterns for camera calibration, we use the function "`cv.findChessboardCorners()`" to find pattern in the chess board and "`cv.drawChessboardCorners()`" to draw the pattern

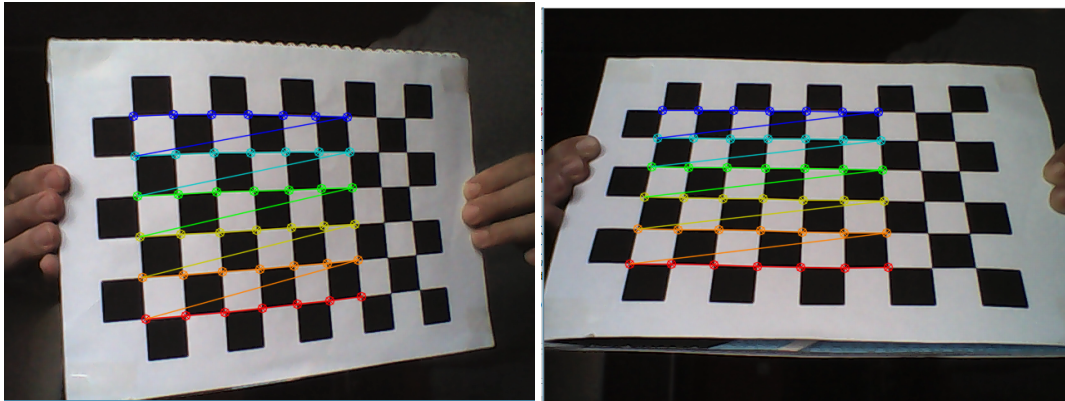


Figure 4: Images with pattern drawn in it

Now we can calibrate the camera and extract the intrinsic and extrinsic parameters:

$$\text{IntrinsicParameters} = \begin{bmatrix} 06884456 & 0 & 266.9266647 \\ 0 & 824.39349912 & 218.14799475 \\ 0 & 0 & 1 \end{bmatrix}$$

The function `calibrateCamera` return extrinsic parameters in the form of rotation vector and translation vector, the rotation is a vector formed of the euler angles representing the rotation, to obtain the rotation matrix, we use the code linked in python file to convert Euler angles to rotation matrix

The extrinsic parameters depend on the position of the chess board, so for each image we have different extrinsic parameters, we are going to represent one matrix for one position of the chess board:

$$\text{RotationMatrix} = \begin{bmatrix} -0.9789908 & 0.00383985 & -0.20386826 \\ 0.10467765 & -0.848553 & -0.51865248 \\ -0.17498457 & -0.52909646 & 0.83032364 \end{bmatrix}$$

$$\text{TranslationVector}_{scale1} = \begin{bmatrix} 3.72721863 \\ 3.12062999 \\ 18.41341474 \end{bmatrix}$$

We have to multiply the result by the size of the chess board square which is $2.4mm$ since our algorithm give us a result in the scale of size of chess board square

$$TranslationVector = \begin{bmatrix} 8.945324712 \\ 7.489511975 \\ 44.192195376 \end{bmatrix}$$

And finally:

$$ExtrinsicParameters = \begin{bmatrix} -0.9789908 & 0.00383985 & -0.20386826 & 8.945324712 \\ 0.10467765 & -0.848553 & -0.51865248 & 7.489511975 \\ -0.17498457 & -0.52909646 & 0.83032364 & 44.192195376 \end{bmatrix}$$

We take a picture of a cup at about 20cm and we propose to estimate its dimensions using the calibrated camera:



Figure 5: Images of a cup

The dimensions of the cup can be computed using the expression above comprising the intrinsic and extrinsic matrices and the coordinates of the cup in the image plane:

$$\mathbf{b} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

We know that $t_1 = t_2 = 0$, $t_3 = 200mm$ and we suppose that there is no rotation, and knowing the resolution of the camera we can obtain $x, y, z = 200$

$$x = x_{image} * resolution / size_{image} = 640 * 4.5 / 11.5$$

$$y = y_{image} * resolution / size_{image} = 480 * 5 / 8.5$$

and we can calculate X, Y

TASK 3: Case 17 Fundamental matrix and disparity map

We have a dataset which contains left and right images from a scene, the task is to use 8 points algorithm in order to find the fundamental matrix:

First we have to find a minimum of 8 corresponding points in both images, for that we use key point detection technique namely the function "load stereoPointPairs" in MATLAB and we obtain the following plot:



Figure 6: matched points in the stereo image

The following plot show the stereo image ater conserving only inliers



Figure 7: matched points in the stereo image after outliers removal

After that we can compute the fundamental matrix :

$$F = \begin{bmatrix} 4,70534018328790e-06 & -0,000360913108080483 & 0,0348275622082616 \\ 0,000364453646511181 & 3,36531699206132e-06 & -0,0937286010924828 \\ -0,0425517566149795 & 0,0993083850885704 & 0,989164068589324 \end{bmatrix}$$

Disparity map: We use matlab and a diparity range of $[-6, 10]$ to compute the disparity map and we obtain the following plot:

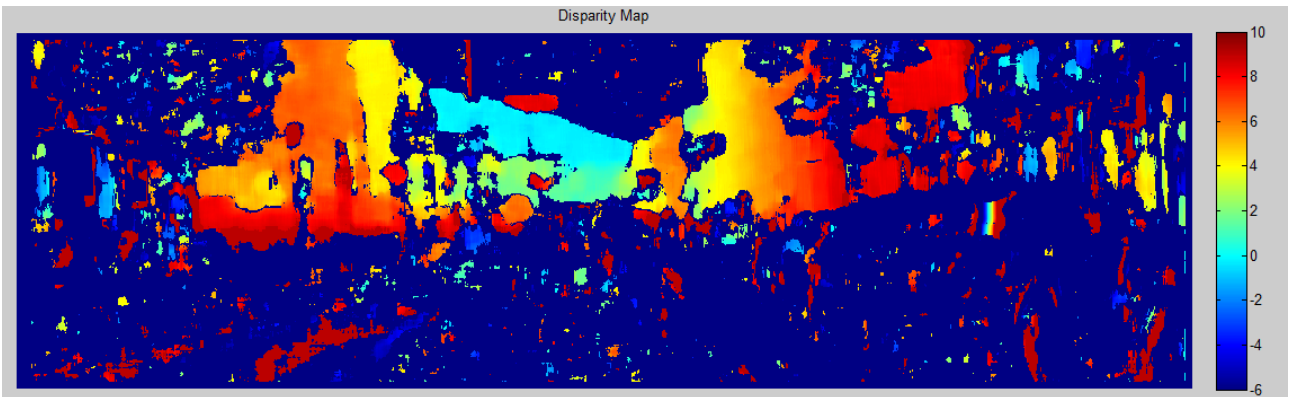


Figure 8: Disparity map