Assignment 4

ECE 181 - Introduction to Computer vision

Kenny Hoang Nguyen
X299187
Section #3

February 2020

1 Discussion

Calibration resulted in the fundamental matrix in 1.
Finding a point in the left image on the android and on the chess board resulted in the epipolar lines on the right image in 1 and 2 respectively. The point on the android (left eye) image was: $(171.42, 299.85)$, and the point on the chessboard image was: $(219.18, 99.76)$. Code for the program is given in the back of this pdf file. I also included a picture showing both lines on the same picture showing they intersect in the epipole.

$8.85785392 \cdot 10^{-6}$	$3.07843326 \cdot 10^{-5}$	$-8.15798560 \cdot 10^{-3}$
$-3.44944817 \cdot 10^{-5}$	$6.08617812 \cdot 10^{-6}$	$1.32520221 \cdot 10^{-2}$
$2.80670273 \cdot 10^{-3}$	$-1.46332656 \cdot 10^{-2}$	1

Table 1: The fundamental matrix for this task

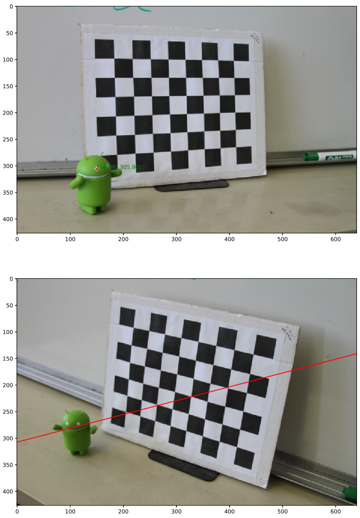

Figure 1: Point on the left image transformed to epipolar line on right image

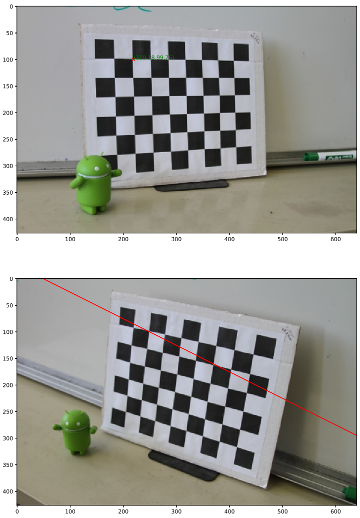

Figure 2: Point on the left image transformed to epipolar line on right image

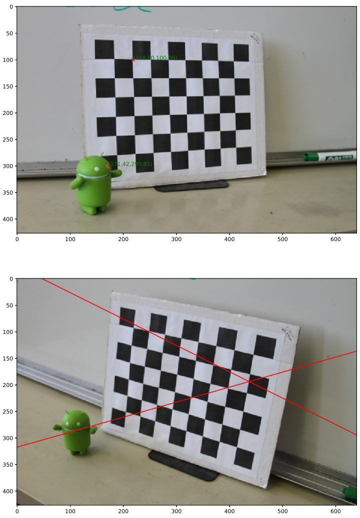

Figure 3: Both lines on the same picture

2 Code for drawing of the epipolar line

```
#!/usr/bin/env python3

import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt


def drawEpipolarLine(x_coord, y_coord):
    # FindFMatrix lets you find points x on the left image and the corresponding
    # points x' on the right image. Which results in the fundamental matrix F
    #F = findFMatrix('left.jpg', 'right.jpg')
    # With the points for chessboard and android we got the F matrix

    F = np.array([[ 8.85785392e-06, 3.07843326e-05, -8.15798560e-03],
                  [-3.44944817e-05, 6.08617812e-06, 1.32520221e-02],
                  [ 2.80670273e-03, -1.46332656e-02, 1.00000000e+00]])

    _, ax = plt.subplots(2)

    # found a b c for line
    # write the point in homogeneous coordinates
    point = np.array([[x_coord], [y_coord], [1]])

    # to find the line l' on the right image l' = F*x
    line = np.matmul(F, point)


    # find two points on the line
    # ax + by + c = 0

    x = np.linspace(0, 639, 1000)
    y = (-line[0]*x-line[2])/line[1]

    ax[0].plot(x_coord, y_coord, 'rx')
    ax[0].text(x_coord, y_coord, f'({x_coord:.2f},{y_coord:.2f})', color = 'green')
    ax[1].plot(x, y, 'r-')

    im1 = plt.imread('left.jpg')
    ax[0].imshow(im1)

    im2 = plt.imread('right.jpg')
    ax[1].imshow(im2)

    plt.show()

def main():
    im1 = plt.imread('left.jpg')
    plt.imshow(im1)
    # Takes in two points can be changed to arbitrary number of points
    picture1 = np.array(plt.ginput(1))
    drawEpipolarLine(picture1[0][0], picture1[0][1])

    return 0

```

```python
if __name__ == "__main__":
    main()
```