MACHINE VISON HW3

Soheil Shirvani

810195416

Question 1:

Our aim in this question is to detect keypoints in 2 given images, match them, find the Fundamental matrix between the 2 images, use this fundamental matrix to draw epipolar lines and also find the depth map of the image pair.

Method

The following steps were followed to find the Fundamental matrix and the depth map from disparity between image1 and image2.

1. SIFT was used to compute keypoints and keypoint descriptors in image1 and image2. Keypoints in both images were captured.

2. Using the keypoint descriptors, good matches with distance lesser than 0.75 were filtered. Matching was done using brute force k-NN matcher with k=2 and L2 norm distances.

3. Good matches obtained were used to find the Fundamental matrix, using RANSAC with the find findFundamentalMat() function. 10 inlier pairs were randomly chosen out of this and plotted.

4. Using these inlier points, epilines on the other image for each image in the image pair were drawn and saved. This was done with the help of the computeCorrespondEpilines() function in OpenCV which accepts the feature points, the index of the image to which the feature points belog to, the fundamental matrix.

5. The depth map was calculated using the stereoBM() function. This is a block matching algorithm which matches corresponding windows/blocks in the image pairs and computes the disparity. The number of disparities and block size were both tuned to get an acceptable depth map.

Parallel epilines

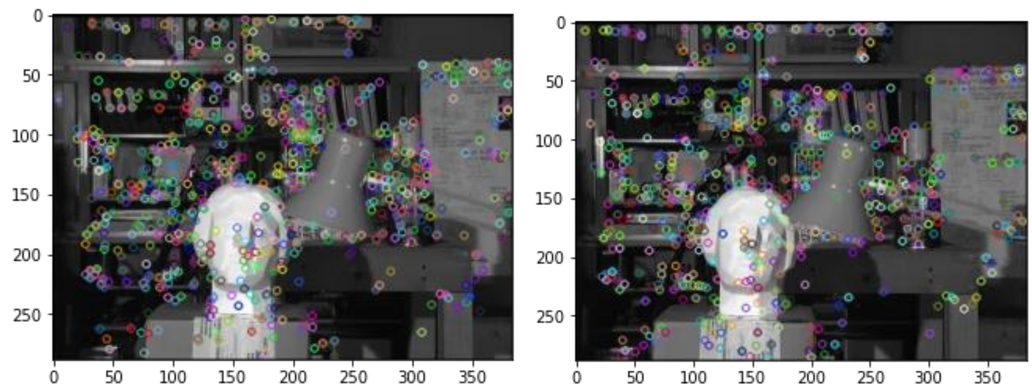
This resulted in epilines that were not parallel (since given images are just x-shifted, epilines should have been parallel), since some keypoint matches returned by SIFT had minor changes in y coordinates. This is shown in figure 4. To overcome this, pixel values were rounded to nearest integer and RANSAC was run with error threshold of 1. Epilines obtained with this setting are shown in figure 3. So, rounding numbers make the epilines parallel so No Rectification was needed.

Results

The Fundamental matrix for the given images was calculated using RANSAC with error threshold of 1 and rounded pixel co-ordinates,

$F =$

```
Fundamental Matrix:
[[ 7.49324246e-05 -3.91483688e-03  3.08538583e-01]
 [ 3.67015957e-03  6.11693262e-06  1.71194073e+13]
 [-3.10120793e-01 -1.71194073e+13  1.00000000e+00]]
```



a) SIFT keypoints detected in the left image. b) SIFT keypoints detected in the right image.

Figure 1: SIFT keypoints detected.

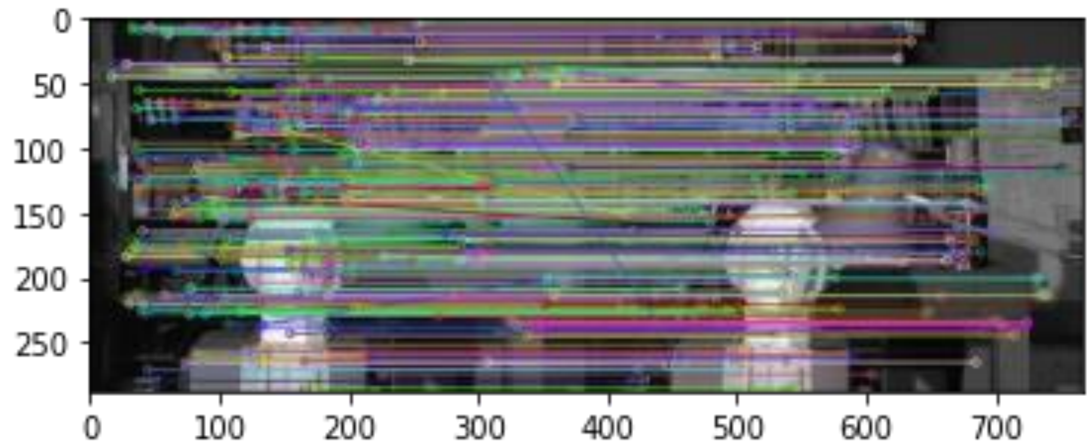
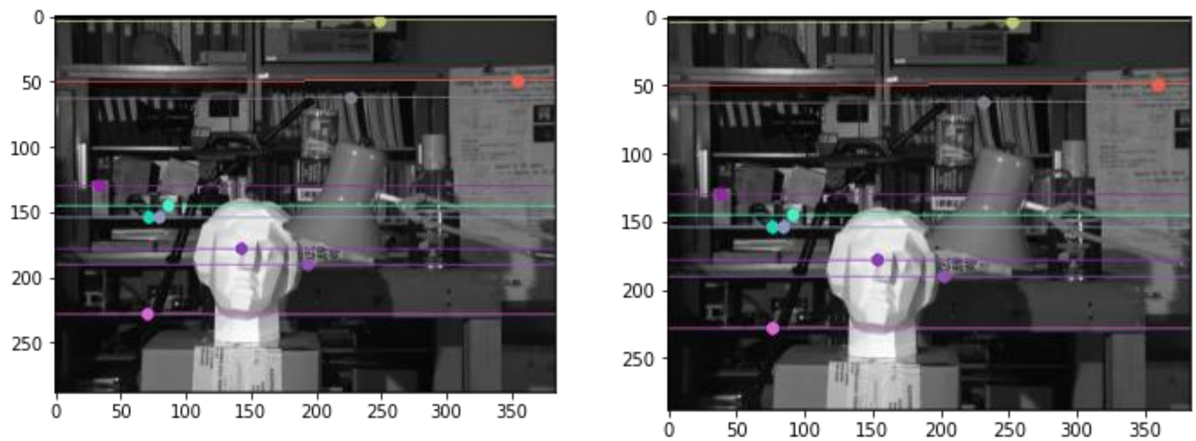


Figure 2: Matches detected with k-NN, with distance lesser than 0.75 between pairs.



a) Epipolar lines for features detected in the right image, drawn on the left image

b) Epipolar lines for features detected in the left image, drawn on the right image.

Figure 3: Epipolar lines, corresponding line pairs are drawn with the same color

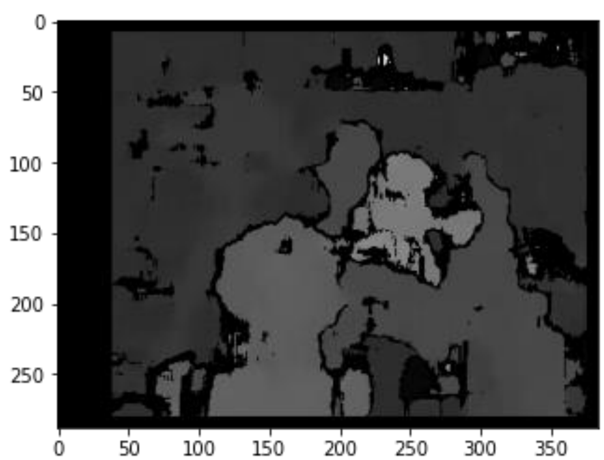


Figure 5: Depth map from disparity computed using block size 17 and number of disparities 32.

Question 2:

Our aim in this question is to calibrate a camera using a 3D calibration object which consists of two orthogonal checkerboard patterns and find the intrinsic and the extrinsic parameters of the camera.

Method

In the setup phase the checkerboard patterns were placed orthogonally for their photo to be taken. To be able to calibrate the camera both world coordinates and their image projections are needed, therefore a coordinate system was created to calculate the world coordinates. In world Coordinates (0, 0, 0) is the conjunction between 3 planes, the right plane is x, left one is y, and the height is z.

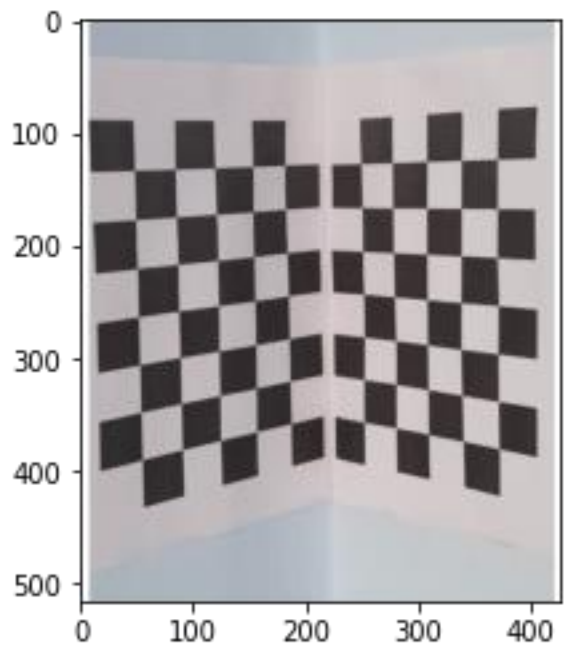


Figure 1: Took photo from 2 orthogonal chess board

When the image is taken Harris corner detector was performed on the picture and 32 corners were selected. Since at least 6 points were needed (12 equations for 11 parameters) this was more than enough. Selected corners can be seen on the picture below:

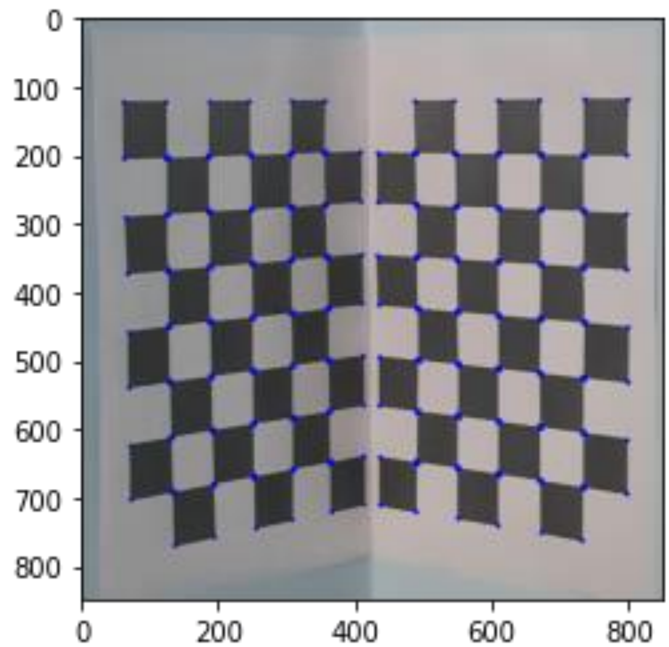


Figure 2: Harris corners detected

Then the positions for these corners were measured with the world coordinate system that is just created. All world coordinates were measured in mm. Then the image coordinates and the world coordinates are saved in 2 vectors named world_coordiate and milimetr_corners for Harris.

After measuring scene Points Pi and their image correspondences pi, the aim is to find the camera matrix M.

Since we know that the product of camera matrix with the world coordinate gives us the image points, we can create a homogenous system to solve for m1, m2 and m3 (which will make up M). 32 points taken created a 64 x 12 matrix for P and we got 12 x 1 for m.

To solve Pm = 0, the singular value decomposition can be used. If P = UVD in the singular value decomposition the column of V with the least eigenvalue will give us m. Therefore, to compute V, the eigenvectors of matrix transpose(P) * P was found. Then the first column was taken, since it was the column with the least eigenvalue. After having the 12 x 1 column of eigen vectors, the data was arranged in 3 x 4 format to construct M.

After constructing M, we can divide it in to 4 matrices A1, A2, A3 and B. A1 will be the first three columns of the first row, A2 will be the first three columns of the second row and A3 will be the first three rows of the third row. B will be the last column of M. Then if we say A = [A1; A2; A3] and A = KR where K is the intrinsic parameter matrix (3 x 3) and R is the rotation matrix (3 x 1) we can apply a RQ factorization.

f = (1/norm(A3)) * A3. e = A2 * transpose(R3)

d = norm (A2 - e * R3)

R2 = (1/d) * (A2 – e*R3)

c = A1 * transpose(R3)

b = A1 * transpose(R2)

a = norm (A1 -(b * R2) - (c * R3))

R1 = (1/a) * (A1 -(b * R2) - (c * R3))

Then every element in the K matrix is divided by f to make f variable 1 as in the original K matrix, since scale is not important. Since K and R is calculated, the only parameter that we need to know is the translation matrix and since T = inverse(K) * B, it was easily calculated, and the project was finished Since we have K and R so the camera matrix was calculated.

Other points were tested since we know that the product of camera matrix and real world coordinates should give us our image corners

Results:

K:

	0	1	2
0	3.515830e-19	9.353418e-07	-0.000281
1	0.000000e+00	6.568830e-06	-0.000160
2	0.000000e+00	0.000000e+00	1.000000

R1, R2, R3, T:

	0
0	-0.886585
1	0.000000
2	0.462566

	0
0	-0.964294
1	0.000000
2	0.264833

	0
0	0.264833
1	0.000000
2	0.964294

	0
0	-0.000002
1	-0.000001
2	0.006604