ENPM662

Project Three

SURIYA SURESH

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

PROBLEM ONE

Given a set of points, the camera intrinsic matrix K had to be found.

1.1 Question One

To find the K matrix, a minimum set of 6 points of image and world points are needed, as the P matrix has 11 degrees of freedom and it means there are 11 unknown values that need to be found. A set of points will allow two values to be found, so 6 set of points are needed to form 12 equations find the P matrix.

1.2 Question Two

The pipeline that is followed:

- Taking pictures of a target to calibrate upon.
- Extracting features for mapping image and world points.
- Mapping features to get image-world coordinate points
- Estimating camera parameters from coordinate points pair.
- Measuring error of estimated points.
- Refining estimated parameters of camera.

1.3 Question Three

The general overview of steps to determine the K matrix is as follows:

- Determine A matrix from given point pairs
- Calculating P Matrix from A
- Determine C matrix from P
- Determine M matrix from P
- RQ decomposition of M to get K and R matrices
- Calculating Re-projection error

Determining A Matrix

After normalizing the world and image points by z score normalization[1], we find the A matrix by looping over all the points that have been defined as a list and we do,

$$\begin{bmatrix} \mathbf{0}_4^{\mathbf{T}} & -w'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} & v'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} \\ w'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} & \mathbf{0}_4^{\mathbf{T}} & -u'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} \\ -v'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} & u'_{\mathbf{i}}\,\mathbf{X}_{\mathbf{i}}^{\mathbf{T}} & \mathbf{0}_4^{\mathbf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

Figure 1.1: The A matrix

We use the relation ,

$$x_i = P * X_i \tag{1.1}$$

As x_i and X_i are proportional ,the dot product of Eq1.1 will be equal to zero. From this relation, we can say that, Fig 1.1 shows the final output of the above equations

$$\mathbf{x}_{\mathbf{i}} \times \mathbf{P}\,\mathbf{X}_{\mathbf{i}} = \begin{bmatrix} v'_{\mathbf{i}}\,\mathbf{p}_3^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} - w'_{\mathbf{i}}\,\mathbf{p}_2^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} \\ w'_{\mathbf{i}}\,\mathbf{p}_1^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} - u'_{\mathbf{i}}\,\mathbf{p}_3^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} \\ u'_{\mathbf{i}}\,\mathbf{p}_2^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} - v'_{\mathbf{i}}\,\mathbf{p}_1^{\mathbf{T}}\mathbf{X}_{\mathbf{i}} \end{bmatrix}$$

Figure 1.2: Cross Product

which is used to find the P matrix.

Calculating P, K, C and R matrices

After the A matrix is found, we do SVD of the A matrix and obtain the P matrix as the last column of the transpose of the V matrix from np.svd.We reshape the column matrix into a 3x4 matrix. This method can be called as direct linear transform.

After we do SVD and obtain the P matrix, the C matrix can be obtained from the SVD of the P matrix,similar to how the P matrix was found, we take the transpose of the last column of the V matrix from SVD and make the last element one by dividing the matrix by the same element.

$$T = -\tilde{C} \tag{1.2}$$

$$\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\widetilde{\mathbf{C}}\right]$$

Figure 1.3: Relation of P, K, C and R matrix

We can find the Translation matrix from eq1.2.
The M matrix is a sub-matrix of the P matrix which is taken by taking the first three elements of the P matrix. We do RQ decomposition by using scipy.linalg.rq [2] to obtain the R and Q matrices which correspond to the K and Rotation matrices respectively. We need to ensure that the last element of the K matrix is unity.
We know that,

$$M = KR \tag{1.3}$$

Using the above relation we can get the K and R matrices as described in the above paragraph.
The scipy.lingalg uses the Gram Schmidt process to do RQ factorization, As seen from numpy docs[3], we can see the qr factorization of that package does the same and by extension so does scipy.RQ factorization is similar to QR factorization , only the order of the matrices vary. So RQ decomposition follows the similar process to QR.
Hence the K,R,P and T matrices have been found.

Question Four and Five

The final results of the P Matrix,K , R and T matrices are shown below.

The re-projection error found was the mean of the x and y errors found by difference of the image points and the estimated points found by multiplying P matrix with the World Points.

```
P
[[ 0.65087998  0.00282762 -0.60261004  0.00880529]
[-0.05953654  0.8794707  -0.05912141 -0.01119818]
[-0.12305918 -0.01144226 -0.13944506  1.        ]]

K
[[ 4.75906963  0.00763496  0.11238334]
[ 0.         -4.73870285  0.15863089]
[ 0.          0.          1.        ]]

R
[[ 0.74951737  0.00624006 -0.66195526]
[ 0.0453193  -0.99809323  0.04190532]
[-0.66043158 -0.06140811 -0.74837101]]

T
[-5.36472297 -0.26529319 -0.04906457 -1.        ]
```

Figure 1.4: P, K, T and R matrix

```
The Error per Point is
0.6456432992961719
0.5098863748518329
0.47594714374074815
0.6569563763332001
0.7887206409888395
0.9397188637556576
0.7547814098777549
0.8831534785705162
```

Figure 1.5: Error of All Points

1.4 Problems Faced

- The error of the images were high so Z Score normalization was done to reduce error.
- RQ factorisation using numpy was giving slightly wrong values after reversing order of matrices so scipy was used to do RQ factorization directly.

Chapter 2

PART TWO

Given a set of images, the camera had to be calibrated for the given set of images that had various views of a checkerboard.From the opencv docs [4], the general steps to do the process and the code was taken.
The general steps followed are:

- Loading images from Folder
- Converting images to Gray-scale
- Finding Corners of Squares on Checkerboard
- Calibrate Camera
- Improving Accuracy of determined Corners
- Output of Determined Corners on Images
- Finding Re-projection Error

After we load the images, we need to define the reference points for the world points so that the image points can be interlinked. We use the dimensions of the given square and a numpy array of orderly increasing (x,y) coordinates of corners of the square.
We find the corners of the checkerboard using cv2.findchessboardcorners and improve corner detection by using cv2.cornersubpix, which sums the gradients in the neighbor points and finds a center.This process continues until the found center stays within a threshold.
Afterwards the corners are drawn using cv2.drawChessboardCorners and the output is displayed in the program. A sample output is attached below.
After the corners are found, using cv2.calibrateCamera which uses world points, image points and size of the image to return a few outputs which includes the intrinsic matrix K.
The re-projection error is found using cv2 functions, Using the output of calibratecamera[5] which gives the rotation and translation vectors, the world points can be

projected to a image plane.This set of points are used to calculate the error using cv2.norm divided by the total points found.

Error and K Matrix

The re-projection error for each image as well as a sample output of the detected corners is shown.

```
list of errors:
0.04409223117723159
0.050312819949747886
0.050920588626377494
0.05596497403044541
0.02887155008050747
0.037426459523077646
0.010766554133792615
0.02576735582963954
0.041555805511274704
0.03043590503124703
0.04480947619840216
0.05648582766649341
0.04229121848640994
K [[681.84652522   0.         255.90180383]
 [  0.         679.29914033 451.1561578 ]
 [  0.           0.           1.        ]]
```

Figure 2.1: List of All Errors and K Matrix

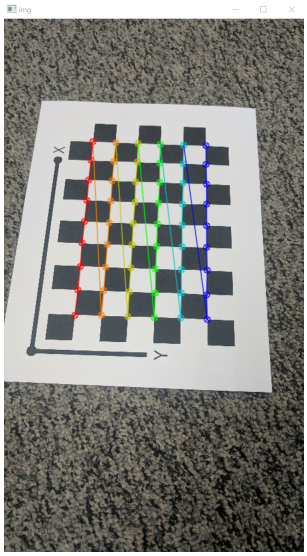

Figure 2.2: Example output of Detected Corners

2.1 Problems Faced

Some of the problems faced were:

- Images were too large when the output was seen so the image was resized.

2.2 Improve Accuracy of K matrix

Some of the possible ways that the K matrix can be improved:

- Ensuring that the calculated values are not rounded too much.
- Taking more images that cover many Point of Views of the Calibration Board
- Ensuring that the calibration target used is level with the ground.
- Ensuring the image taken is with a steady focus.
- Ensuring distortion is reduced as much as possible.
- Ensuring the images taken are clear and have sharp outline of the calibration target.

2.3 References

Additional references used but not mentioned are Class slides.

REFERENCES

[1] Sourabh Gupta. *Z Score Norm*. https://ml-concepts.com/2021/10/08/z-score-normalization/. [Online]. 2022.

[2] Scipy. *RQ*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.rq.html. [Online]. 2020.

[3] Numpy docs. *QR*. https://numpy.org/doc/stable/reference/generated/numpy.linalg.qr.html. [Online]. 2022.

[4] OPEN-CV docs. *Camera Calibration*. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html. [Online]. 2022.

[5] OPEN-CV docs. *calibrate camera fn*. https://docs.opencv.org/4.x//d0c/group__calib3d.html#ga687a1ab946686f0d85ae0363b5af1d7b. [Online]. 2022.