Korea Advanced Institute of Science and Technology

School of Electrical Engineering

EE735 Computer Vision Fall 2018

Student's Name: Dinh Vu

Student's ID: 20184187

Programming Assignment 2

1. The Given Dataset

1.1. Feature Extraction and Matching

In order to extract and match features in each picture, the Scale-Invariant Feature Transform (SIFT) is applied by using respectively two functions `vl_sift()` and `vl_ubcmatch()` in VLFeat toolbox [1]. Figure 1.1 shows the all features, marked by red dots, in the frame 8 and 9, while Figure 1.2 presents the matching features between them.

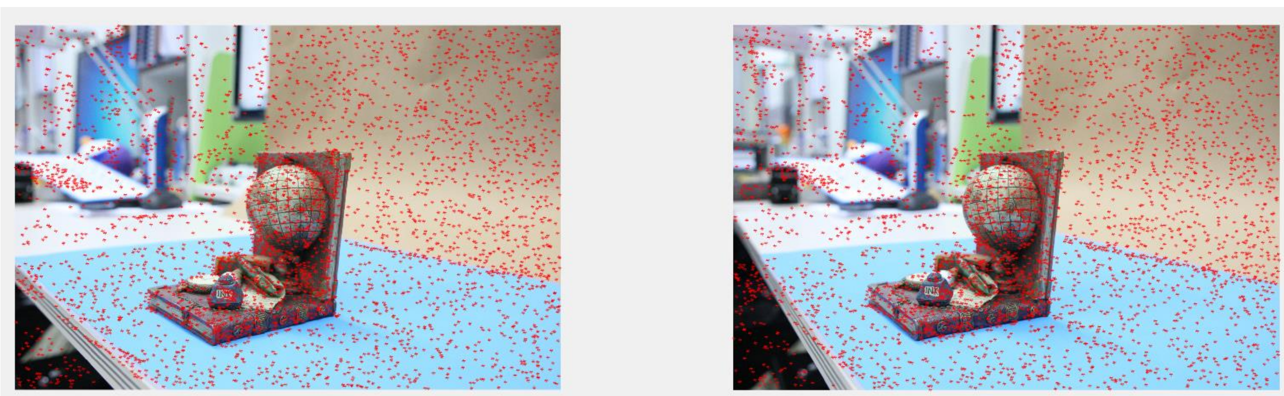

Figure 1.1. All features in the frame 8 (left) and 9 (right)

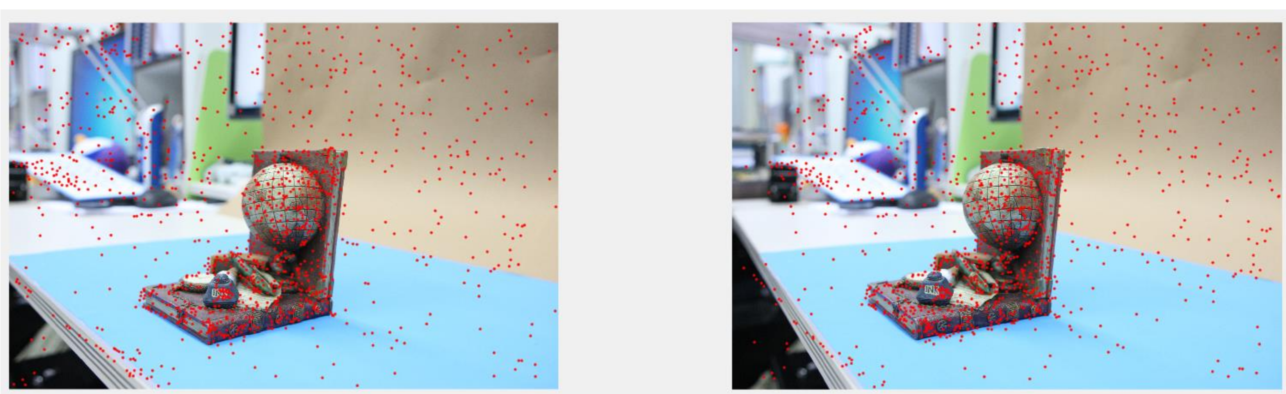

Figure 1.2. The matching features between the frame 8 (left) and 9 (right)

1.2. Initialization Step

The best pair of images is chosen by the maximum number of correspondence between two images. One image is call reference view and other is pair view. For the given dataset, the reference and pair view are the image 8 and 9, respectively. First, the essential matrix E is determined from the coordinates of all correspondent points in the normalized image plane by using 5-point calibrated algorithm [2]. However, there are several cases for the essential matrix, hence, each case is checked with the distance calculated by the following equation:

$$d = \sqrt{\frac{(x_i'^T F x_i)^2}{(Fx_i)_1^2 + (Fx_i)_1^2 + (F^T x_i')_1^2 + (F^T x_i')_2^2}} \quad (1.1)$$

Where:

- d is the distance from the point to the epipolar line
- x and x' are the coordinates of each correspondent point in the normalized image plane
- F is the fundamental matrix, calculated by the equation: $F = (K^T)^{-1} E K^{-1}$ (1.2)

If the distance is less than the threshold, which is chosen to be 1, that point is an inlier, otherwise, it is not. The case has the highest number of inliers is the true essential matrix.

After finding the essential matrix, the camera matrix, $P = [R|T]$, must be estimated by decomposing the essential matrix in `Ess2Cam.m` file. There are always four options for the camera matrix. In order to determine the best option, each option is checked by the following steps:

- *Step 1*: Choose one of four option then process the triangulation where the chosen P corresponds with the pair view, while the reference view has $P = \text{eye}(3,4)$.
- *Step 2*: Calculate the depth of each 3D reconstructed point by the below equation.

  $$\text{depth}(X, P) = \frac{sign(|R|)c}{T\|R_3\|}$$

  Where: c is the camera center and R_3 is the third column of R.
  If the depth is positive, the 3D point exists in real world, otherwise not. Counting the number of the existing points in 3D.
- *Step 3*: Repeat the step 1 and 2 until all options are checked. Then the best P gives the maximum number of the existing points in the 3D world.

After getting the initial camera matrix, the initial view can be displayed in Figure 1.3.

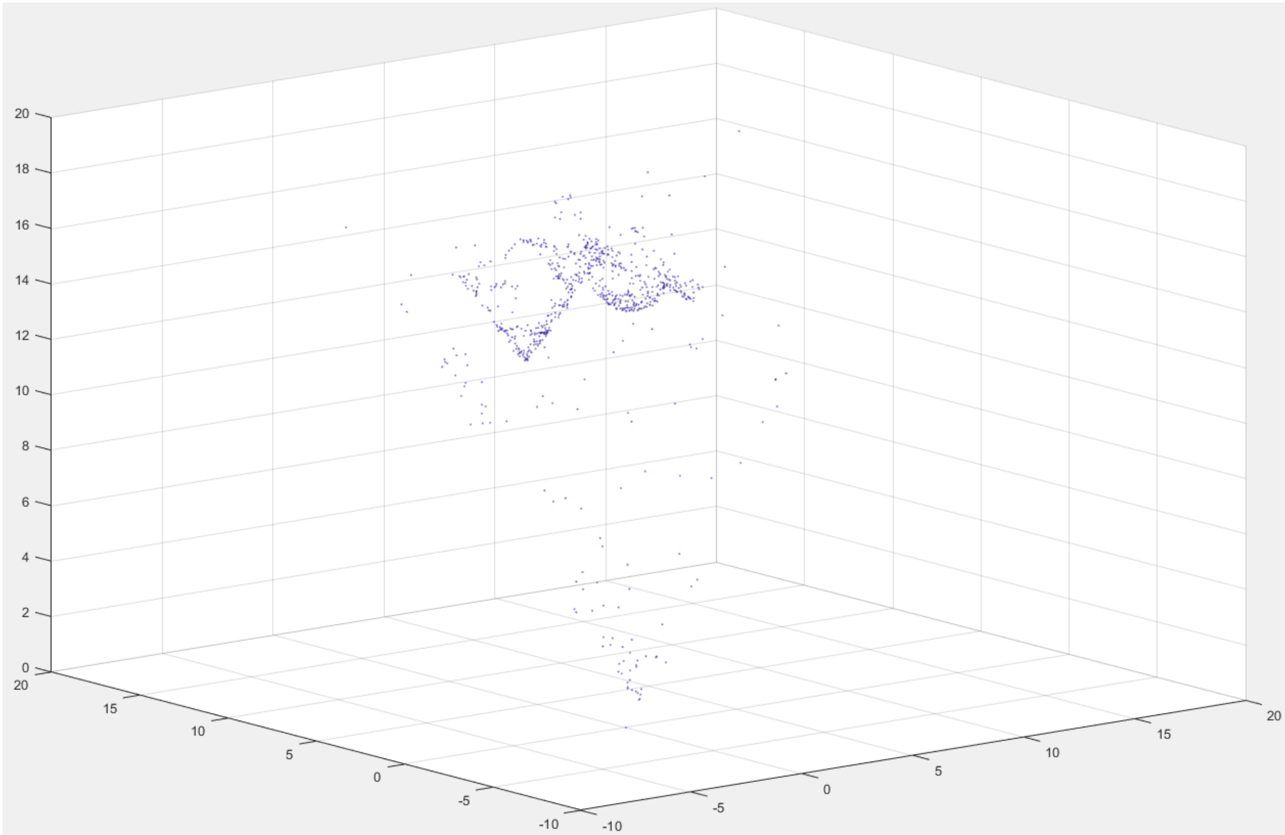

Figure 1.3. The 3D initial view for the given dataset

1.3. Growing Step

In the growing step, all other images must be joined to reconstructed more and more 3D points as possible. The three view geometry in [3] is applied.

- *Step 1*: The new chosen view is the image having greatest number of matching features with the two previous view.
- *Step 2*: All 3D points, which can be projected in all 3 views, is found by simply checking the index in the list of matching features.
- *Step 3*: The new camera matrix is determined from the 3D matching points and the coordinates of their projection on the normalized image plane by using perspective 3-point algorithm [4].
- *Step 4*: The outcome of the given function `PerspectiveThreePoint()` include various camera matrices. The best camera matrix can be found by checking the Sampson distance in the pair view and the new view.

$$d = \sqrt{\left\|x_{pair} - KP_{old}X\right\|^2 + \|x_{new} - KP_{new}X\|^2}$$

Where:

- x_{pair} and x_{new} are the normalized image coordinate of the 3-view matching points
- P_{old} and P_{new} are the camera matrix of the pair view and the new view

- X is the 3D coordinates of each 3-view matching points
- d is the Sampson distance

❖ *Step 5*: If the Sampson distance is less the threshold, also equal to 1, that point is an inlier. Counting the number of inliers.

❖ *Step 6*: Repeat step 1 to 5 until all potential camera matrices are checked. The best new camera matrix has the maximum number of inliers.

❖ *Step 7*: Process triangulation with the best new camera matrix and the matching features between the pair and new view. The new 3D reconstructed coordinates are concatenated with the previous 3D reconstruction.

After finishing the above 7 steps, the pair view become the reference view, the new view become the pair view. The new view is chosen among the rest unselected image. When all images are selected, the final view are achieved as shown in Figure 1.4.

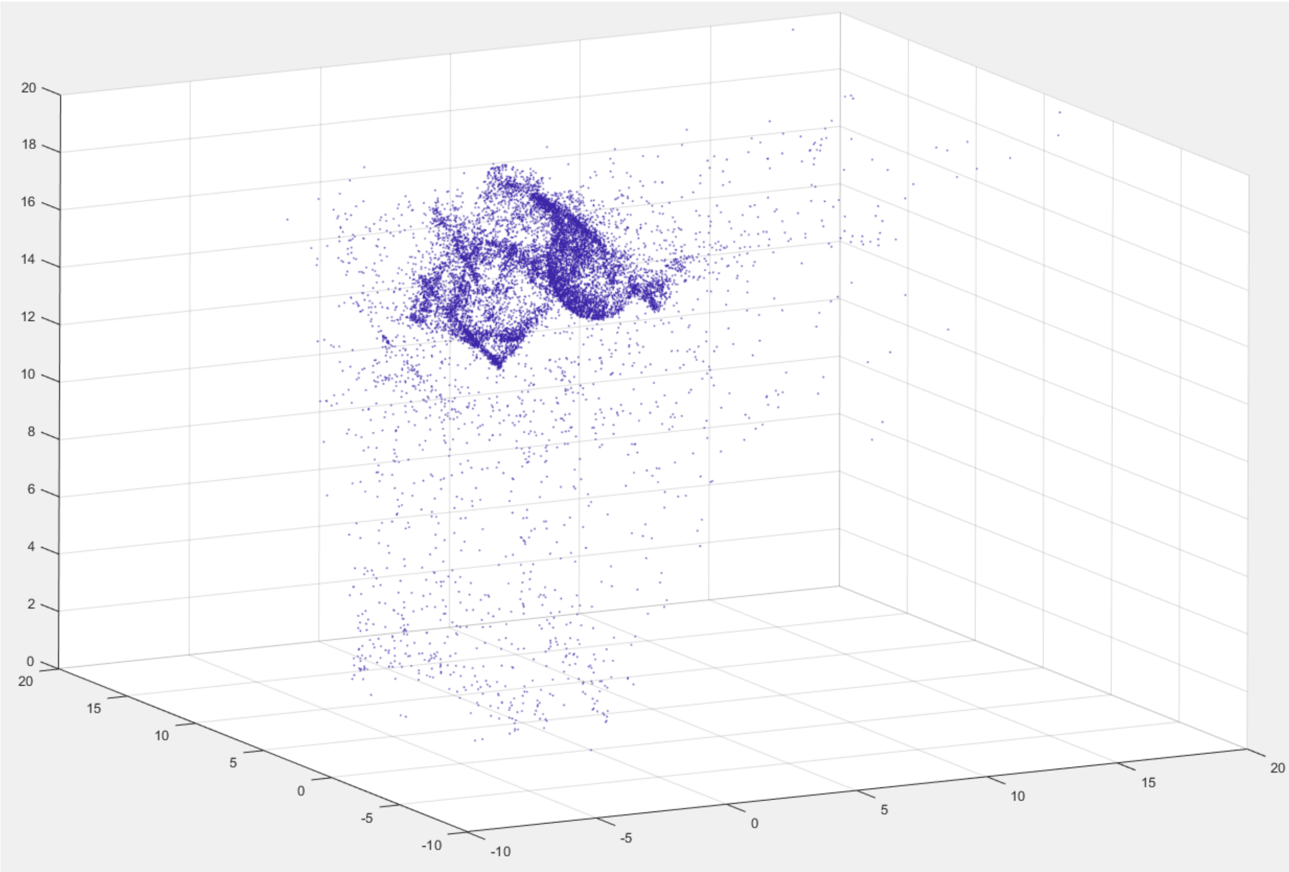

Figure 1.4. The 3D final view of the given dataset

2. My Own Dataset

2.1. Camera Calibration

The Camera Calibration Toolbox for Matlab in [5] is used to find the camera intrinsic matrix K. The checkerboard is taken photos ten times with difference camera positions as in Figure 2.1.

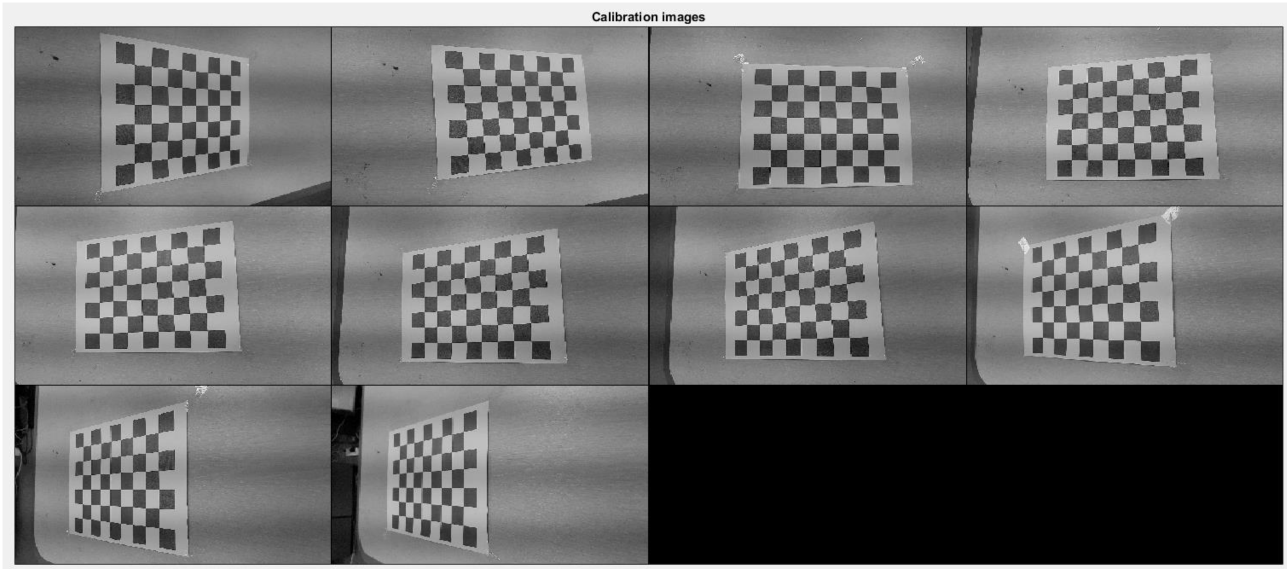

Figure 2.1. The checkerboard images for camera calibration

The next step in camera calibration is extracting corners by using GUI explained in the second link of [5]. Figure 2.2 shows an example of this step.

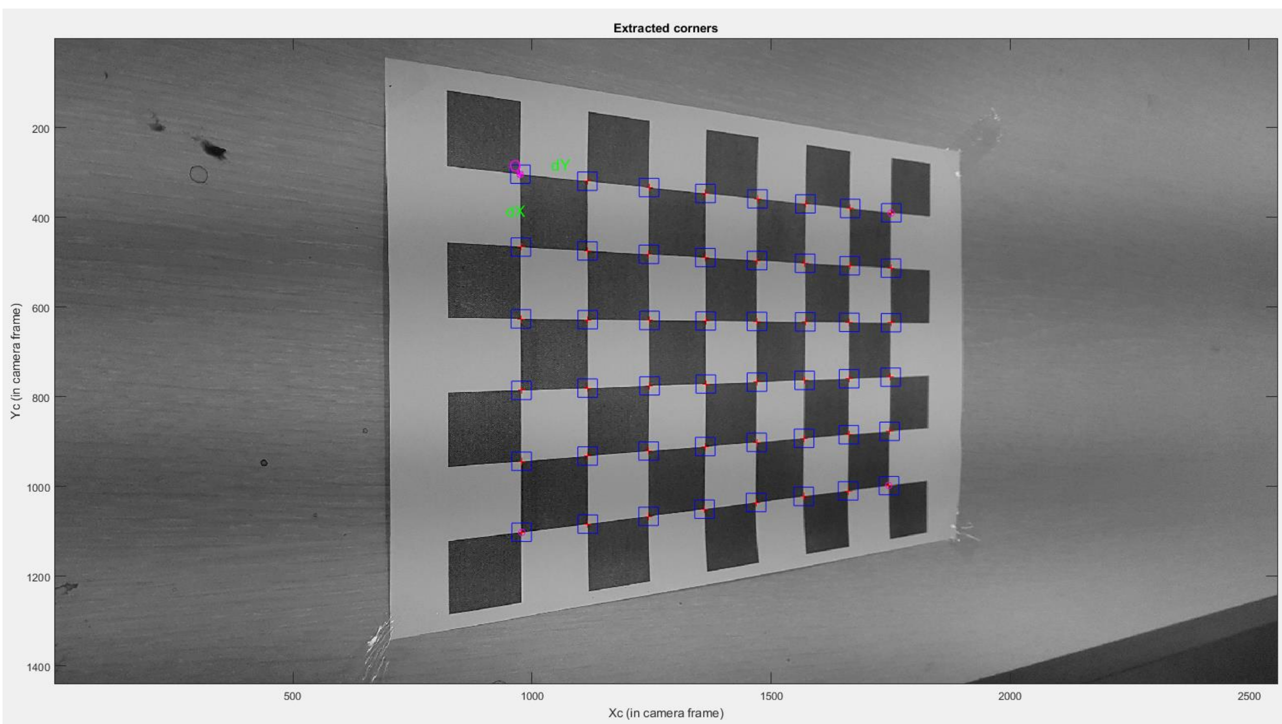

Figure 2.2. Extracting corners

After extracting corners in total ten checkerboard images, the Toolbox will estimate the focal length, coordinates of the principal point and the skew with distortion and pixel error, which are presented in Figure 2.3. The pixel errors in two dimensions are less than one.

```
Calibration parameters after initialization:

Focal Length:          fc = [ 2055.87618   2055.87618 ]
Principal point:       cc = [ 1279.50000   719.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 10
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 2078.44319   2068.51171 ] +/- [ 16.26949   14.85358 ]
Principal point:       cc = [ 1296.59919   715.14629 ] +/- [ 12.94148   11.29765 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.02073   -0.05428   -0.00427   -0.00662  0.00000 ] +/- [ 0.02233   0.13536   0.00200   0.00226
Pixel error:          err = [ 0.59309   0.48578 ]
```

Figure 2.3. Output of the camera calibration

Therefore, the camera intrinsic matrix can be determined as below:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2078.44319 & 0 & 1296.59919 \\ 0 & 2068.51171 & 715.14629 \\ 0 & 0 & 1 \end{bmatrix}$$

2.2. Feature Extraction and Matching

Due to save execution time when operating my dataset, only ten pictures are captured.

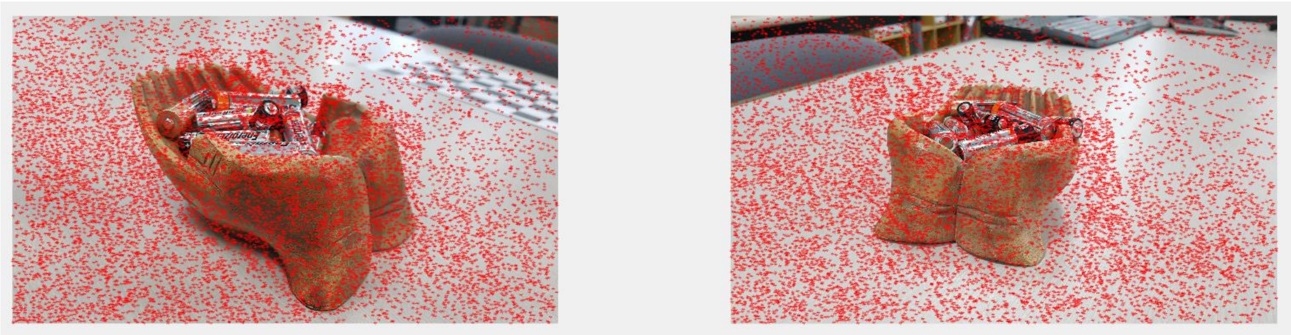

Figure 2.4. The feature extraction in the picture 3 (left) and 4 (right)

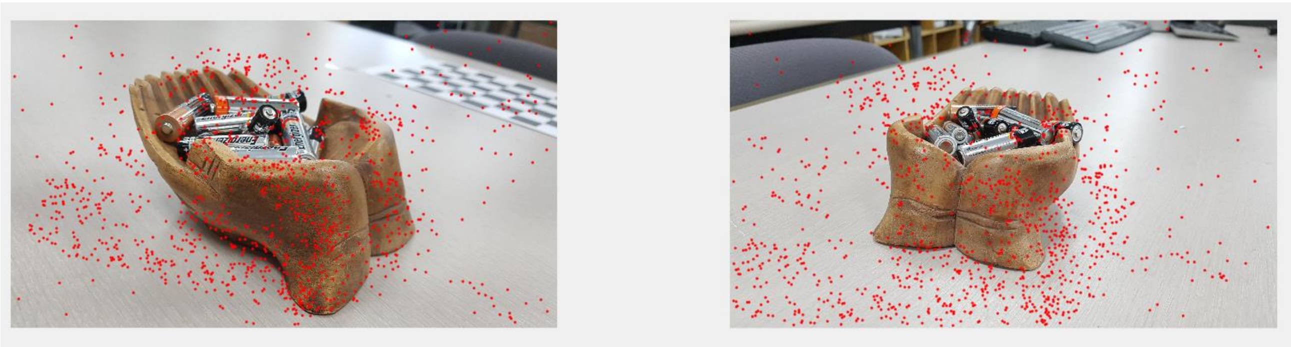

Figure 2.5. The matching features in the picture 3 (left) and 4 (right)

2.3. Initialization Step

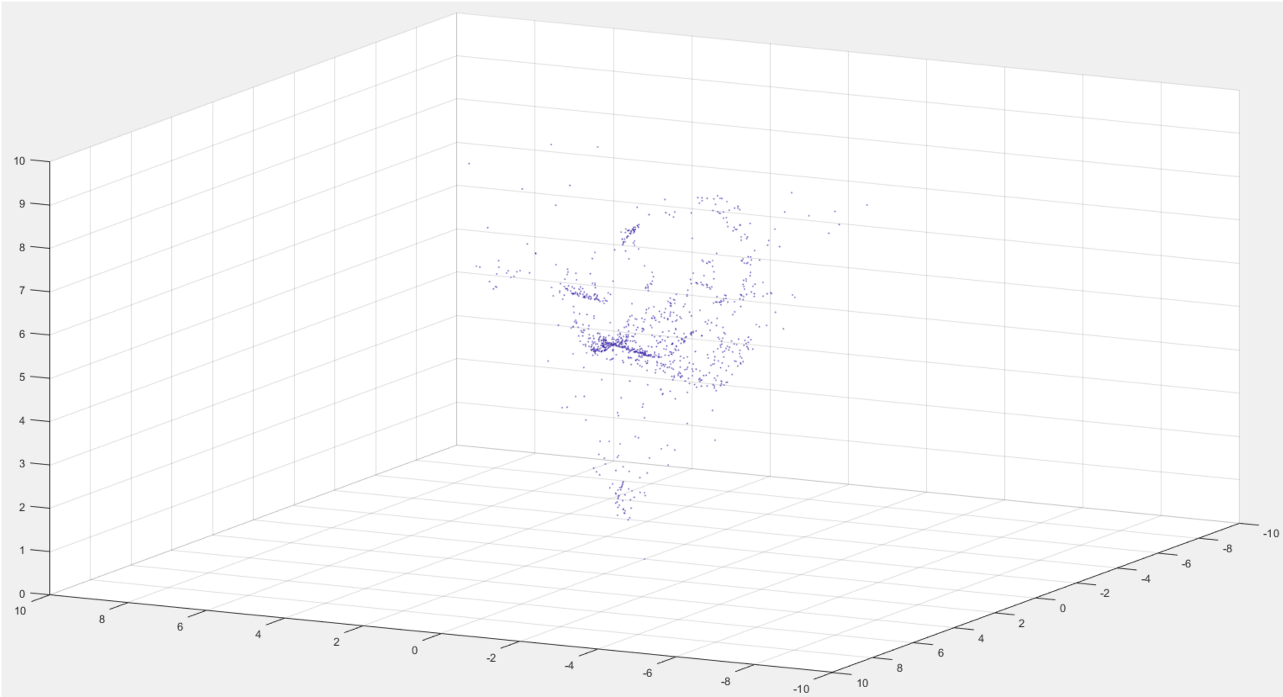

Figure 2.6. The initialization 3D view for my dataset

2.4. Growing Step

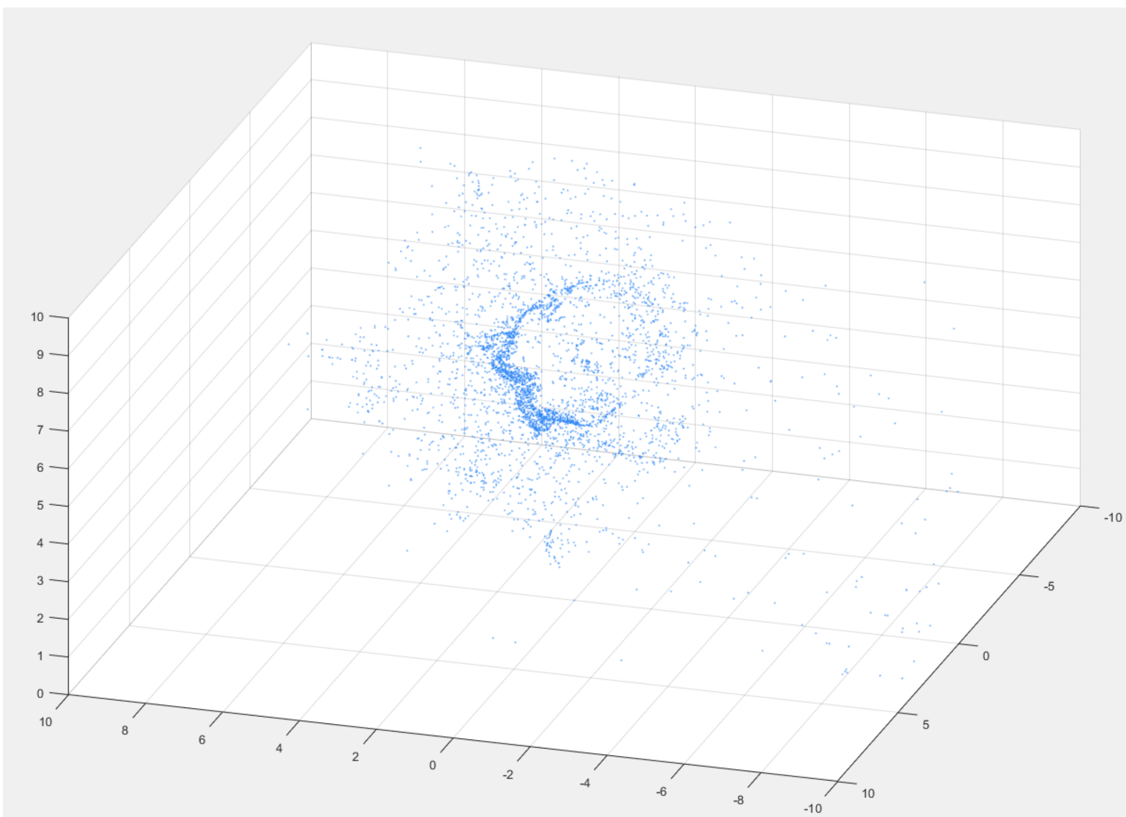

Figure 2.7. The final 3D view for my dataset

Reference

[1] Andrea Vedaldi and Brian Fulkerson, “VLFeat Toolbox”, *The VLFeat Team*, http://www.vlfeat.org

[2] David Nistér, “An Efficient Solution to the Five-Point Relative Pose Problem”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, issue 6, June 2004

[3] Richard Hartley and Andrew Zisserman, “Multiple View Geometry in Computer Vision”, Cambridge University Press, 2003, USA

[4] Robert M. Haralick, Chung-nan Lee, Karsten Ottenberg and Michael Nolle, “Review and Analysis of Solutions of the Three Point Prespective Pose Estimation Problem”, *International Journal of Computer Vision*, volume 13, issue 3, pp 331-356, December 1994

[5] Camera Calibration Toolbox for Matlab:

http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html