HW4 Structure from Motion

Group 17

Introduction

In this report we will first introduce the Structure from Motion(SfM) and implement procedure in detail. Second, we will show our result of using SfM to two images and get 3D points then constructing 3D model for both TA's testing data and our data. Finally, we have some discussion and conclusion to this report.

Implement procedure

In this section we will show our step by step implementation procedure.

1. Find out correspondence across images

First, we need to find those important features in the image.In our case, we use SIFT method in Opencv to find keypoints(features), where SIFT is invariant to rotation, translation and scale.

After getting the key points of two images, now we have to match their corresponding keypoints. So for each keypoint in the image A, we will iterate all keypoints in image B, and find its best & second best matching keypoints by calculate their L2 norm distance. The time complexity of this process is O(M*N), where M is the number of keypoints in image A and N is the number of keypoints in image B. After matching keypoints between image A, B. We will discard those matching which ratio distance is too large in order to eliminate ambiguous matchings. In this step, we set a **ratio** variable as threshold.

2. Estimate the fundamental matrix across images (normalized 8 points)

With the key points of image A and image B, now we use them to find the fundamental matrix by 8-points algorithm with RANSAC. First we randomly choose 8 points from the key points, and centroid of the points are at the origin and average distance to the origin is equal to sqrt(2). After getting normalize points we need to get an A matrix by 2 set of 8 points from normalize key points of image A and image B. Since we need to find the solution of the equation of Af = 0, we use SVD to decompose A then we get U, S, V matrix. Then we get the first F from the last column of V matrix. But there will be a problem of that f that is the determinant of f is not 0. Therefore we resolve det(f)=0 constraint using SVD, and get the U,S,V matrix from f. We then let the last element of S to zero, and multiply the U,S,V matrix back to get the fundamental matrix. Now we use F and key points to select indices with accepted points, and Sampson distance as error. With the error, we use them to get the inliner index from key points and get the best F of these points.

$$\mathbf{Af} = \begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_n' & y_n'x_n & y_n'y_n & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

3. Draw the interest points on you found in step.1 in one image and the corresponding epipolar lines in another

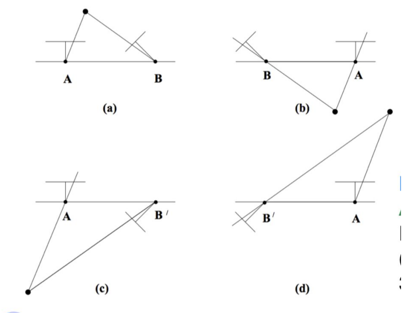
With the fundamental matrix and inliner of image A and image B, now we can use them to draw the epipolar lines.

4. Get 4 possible solutions of essential matrix from fundamental matrix

Now we can simply use fundamental matrix to get essential matrix by matrix multiply the intrinsic matrix of camera 1 and 2 and fundamental matrix. Then we need to find the 4 possible answer of essential matrix. First we use SVD to the essential matrix and get U, S, V matrix. And we use the following method in the image to get 4 possible answer.

$$\mathbf{P}_2 = \left[\mathbf{UWV}^\top \mid +\mathbf{u}_3\right]$$
$$\mathbf{P}_2 = \left[\mathbf{UWV}^\top \mid -\mathbf{u}_3\right]$$
$$\mathbf{P}_2 = \left[\mathbf{UW}^\top\mathbf{V}^\top \mid +\mathbf{u}_3\right]$$
$$\mathbf{P}_2 = \left[\mathbf{UW}^\top\mathbf{V}^\top \mid -\mathbf{u}_3\right]$$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



5. Find out the most appropriate solution of essential matrix and apply triangulation to get 3D points

Now we have 4 possible answer, we need to find the most appropriate answer from them. First we try to use each possible solution to get 3D points(triangulation), and then pick the solution with most of 3D points in front of the camera. The following image shows the method that finding the A matrix and using SVD to get the U, S, V matrix, and the last column of V matrix is the answer that we need.

$$x = w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} \qquad x = w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}X = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} X = \begin{bmatrix} \mathbf{p}_1^\top X \\ \mathbf{p}_2^\top X \\ \mathbf{p}_3^\top X \end{bmatrix}$$

$$x' = w\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad \mathbf{P}' = \begin{bmatrix} \mathbf{p'}_2^\top \\ \mathbf{p'}_2^\top \\ \mathbf{p'}_3^\top \end{bmatrix} \qquad w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u\mathbf{p}_3^\top X \\ v\mathbf{p}_3^\top X \\ \mathbf{p}_3^\top X \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^\top X \\ \mathbf{p}_2^\top X \\ \mathbf{p}_3^\top X \end{bmatrix}$$

$$\Downarrow$$

$$\begin{aligned} u\mathbf{p}_3^\top X - \mathbf{p}_1^\top X &= \left[u\mathbf{p}_3^\top - \mathbf{p}_1^\top\right] X = 0 \\ v\mathbf{p}_3^\top X - \mathbf{p}_2^\top X &= \left[v\mathbf{p}_3^\top - \mathbf{p}_2^\top\right] X = 0 \\ u'\mathbf{p'}_3^\top X - \mathbf{p'}_1^\top X &= \left[u'\mathbf{p'}_3^\top - \mathbf{p'}_1^\top\right] X = 0 \\ v'\mathbf{p'}_3^\top X - \mathbf{p'}_2^\top X &= \left[v'\mathbf{p'}_3^\top - \mathbf{p'}_2^\top\right] X = 0 \end{aligned}$$

$$\mathbf{A}X = 0 \quad A = \begin{bmatrix} u\mathbf{p}_3^\top - \mathbf{p}_1^\top \\ v\mathbf{p}_3^\top - \mathbf{p}_2^\top \\ u'\mathbf{p'}_3^\top - \mathbf{p'}_1^\top \\ v'\mathbf{p'}_3^\top - \mathbf{p'}_2^\top \end{bmatrix} \Leftarrow$$

solve via SVD

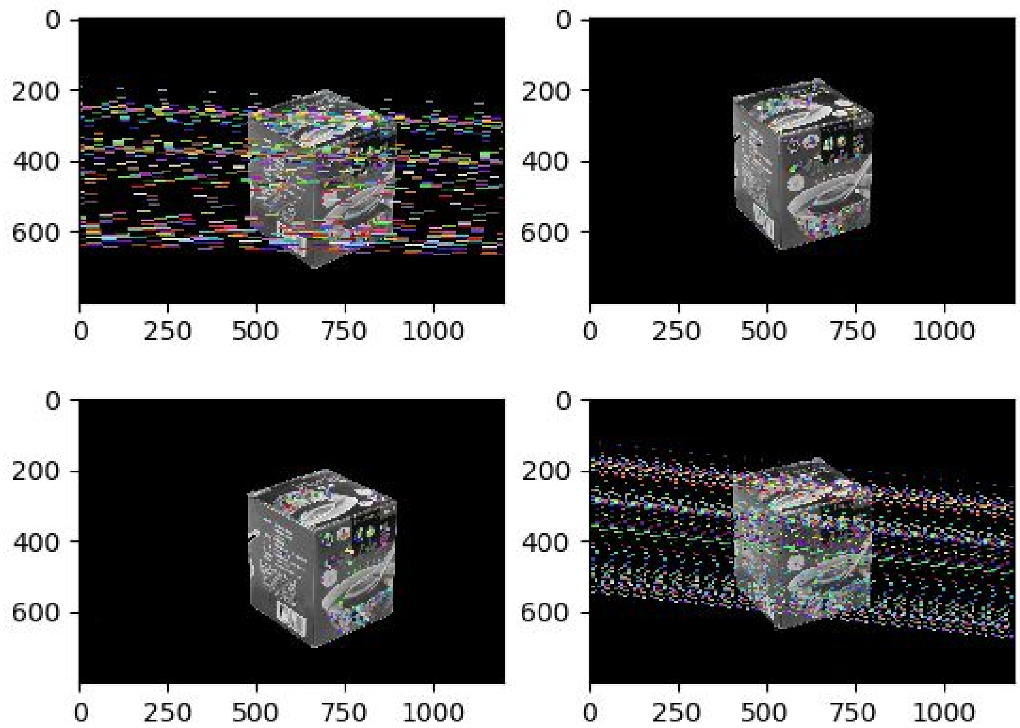
6. Use texture mapping to get a 3D model

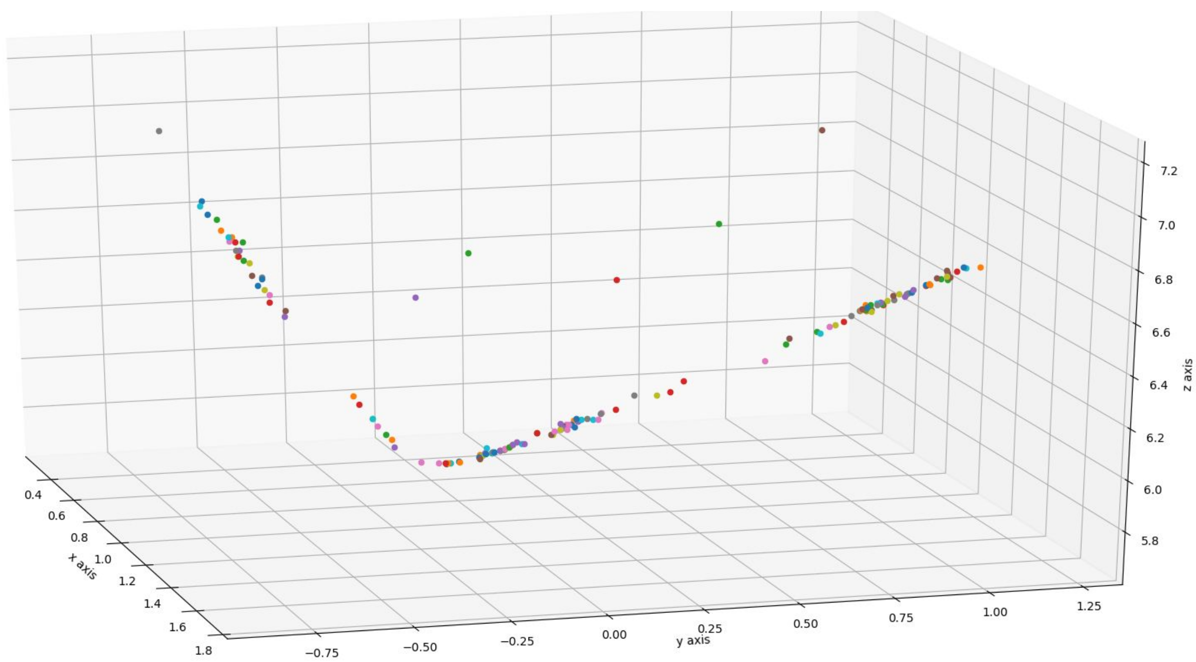
Finally we use the 3D points that in front of the camera and using python to call TA’s matlab code to match the texture of each point. The result will show in below section.

Experiment Result

1. Mesona

   We get 463 interesting point, and after RANSAC,157 of them are inliers.





Fundamental matrix:

```
[[ 9.58368538e-08 -8.27728149e-07  8.46250446e-04]
 [-2.02796005e-07 -8.33010938e-08 -1.72920484e-02]
 [-1.89585775e-03  1.84741063e-02  1.00000000e+00]]
```

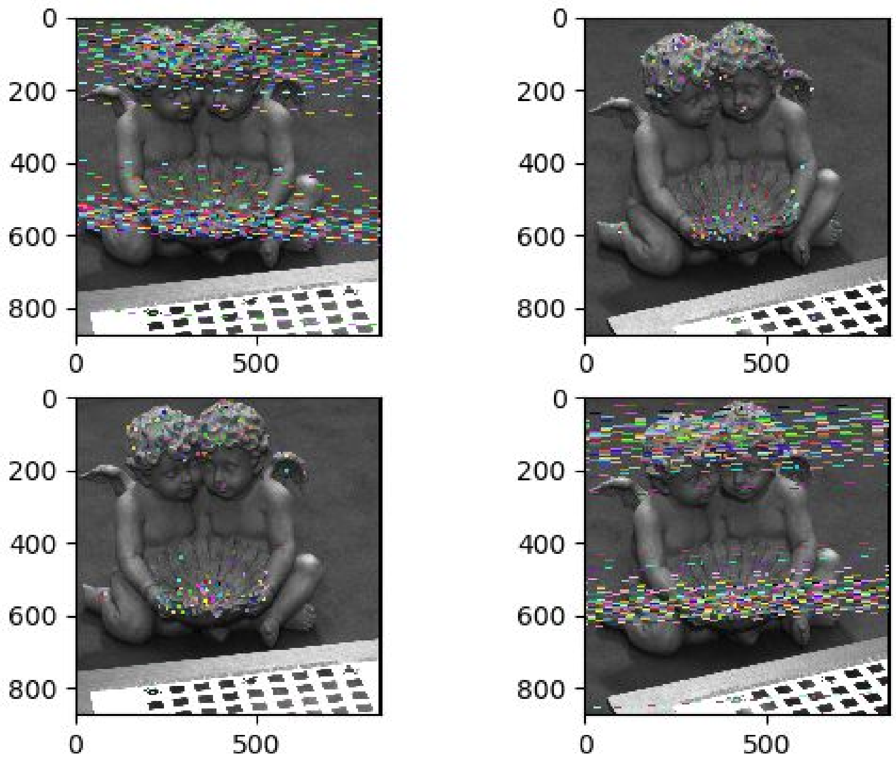
Essential matrix:

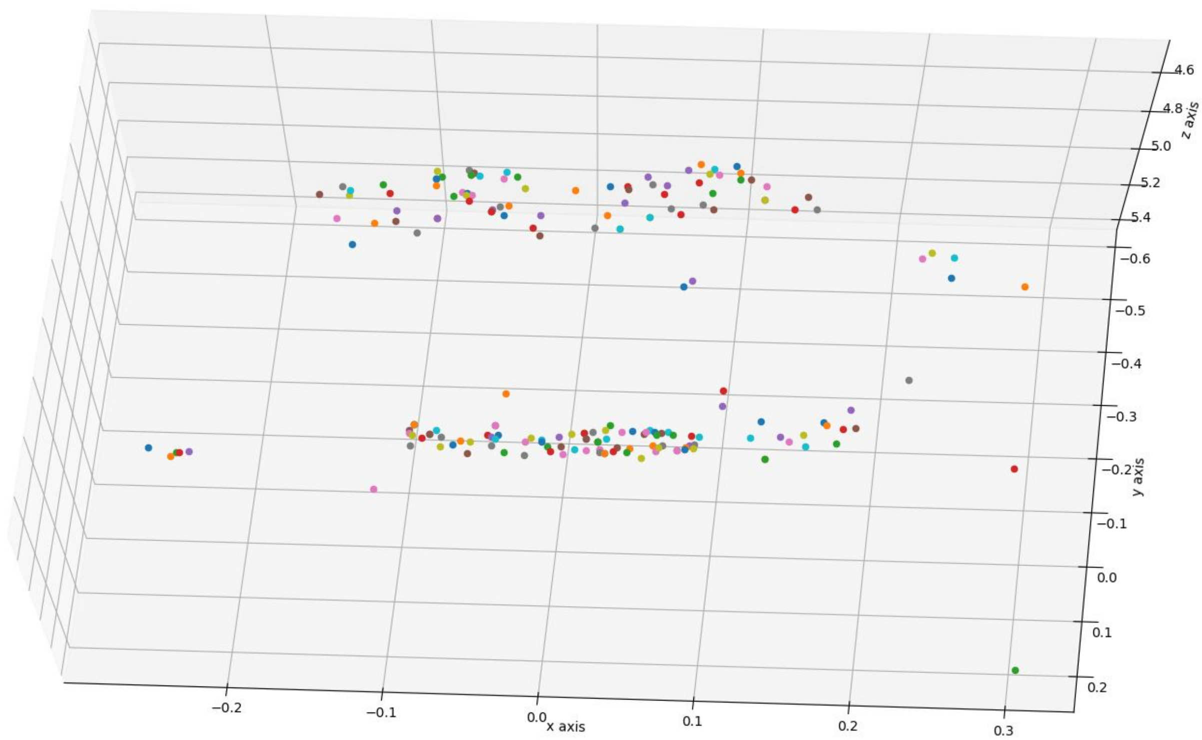
```
[[  0.19376291  -1.67343231   0.82519729]
 [ -0.40994475  -0.16915075 -24.77913722]
 [ -2.73596405  25.62303536   0.72825882]]
```

4 kinds of viewing angle:

2. Statue

We get 434 interesting point, and after RANSAC,177 of them are inliers.





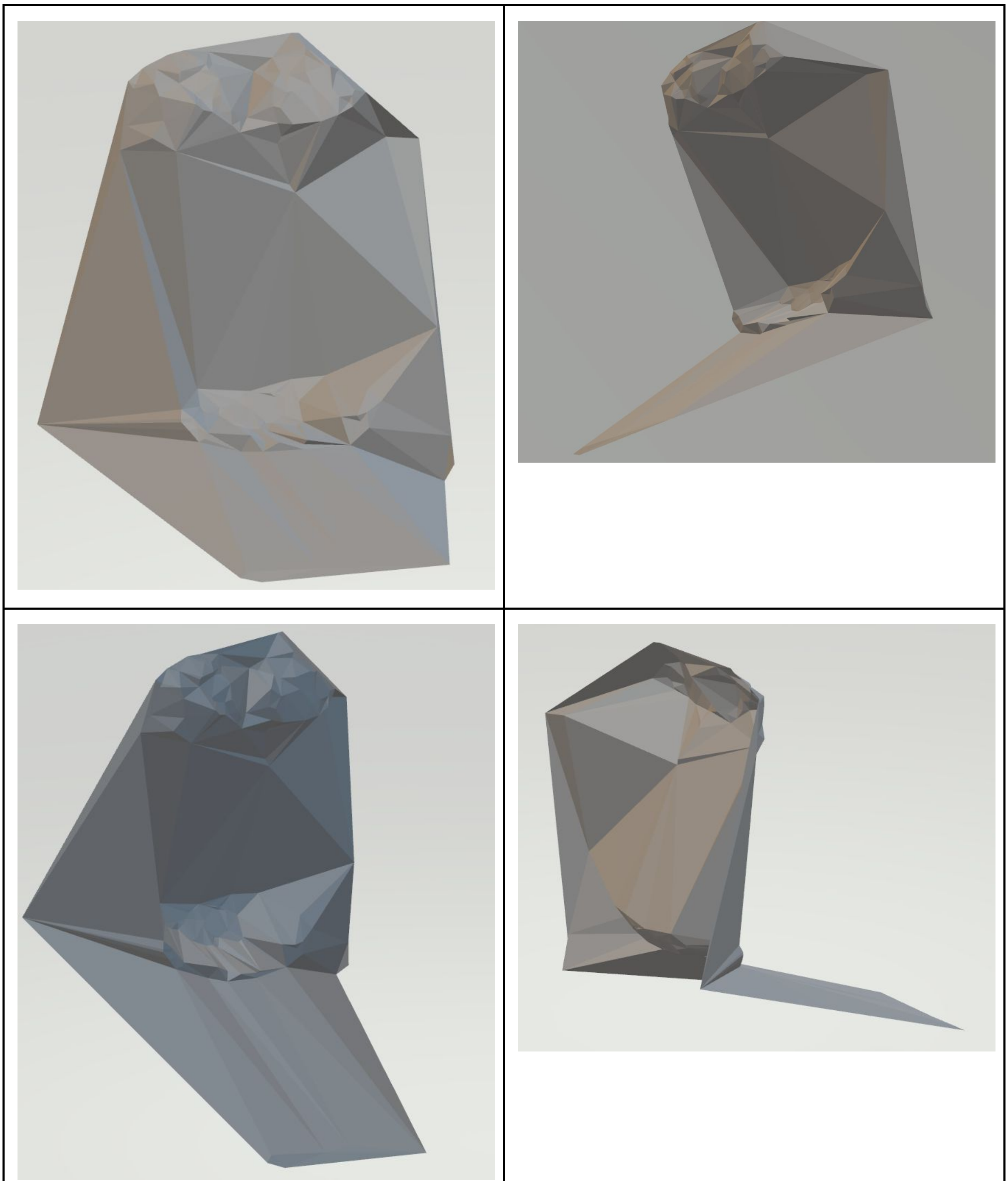
Fundamental matrix:

```
[[ 1.34633484e-08 -3.26839161e-07 -1.08993947e-03]
 [-6.56950964e-07  1.75708284e-07  2.81798273e-02]
 [-1.05558303e-03 -2.76486573e-02  1.00000000e+00]]
```

Essential matrix:

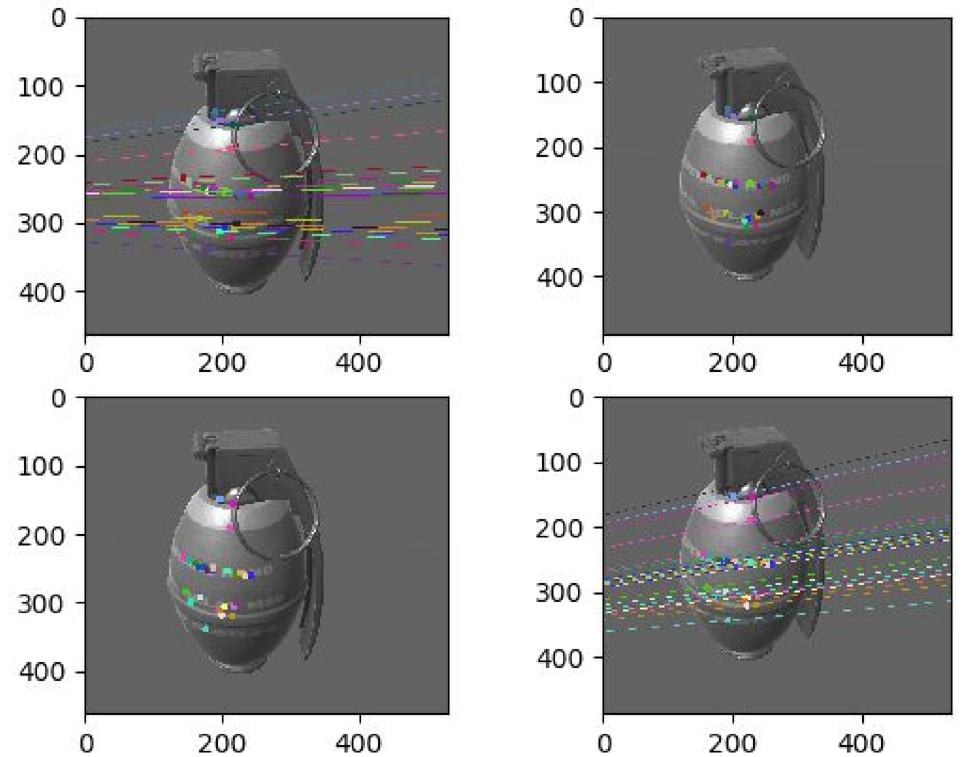
```
[[   0.39646368   -9.6184987    -6.98705872]
 [ -19.33335753    5.16402911  152.03315622]
 [  -8.01757436 -149.9100719     1.19968926]]
```

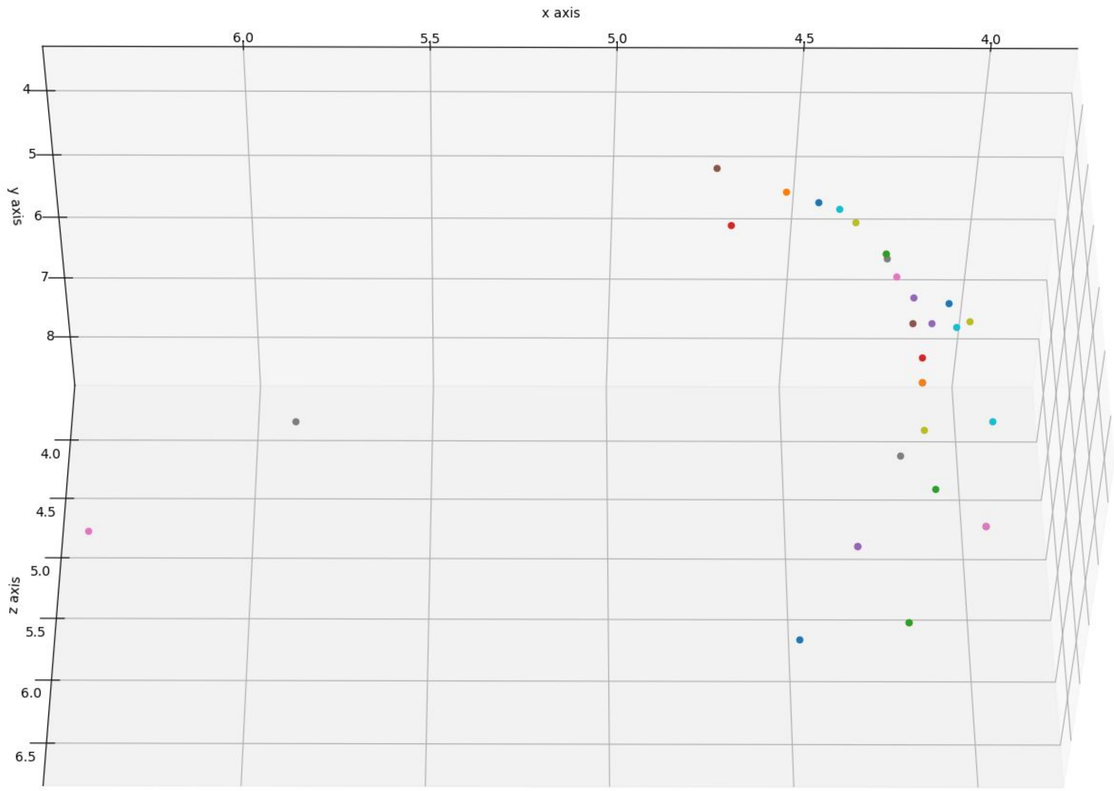
4 kinds of viewing angle:



Our testing Image

We get 72 interesting point, and after RANSAC, 32 of them are inliers.





Fundamental matrix:
[[-8.07324981e-06 -9.65981843e-05 1.85503362e-02]
[8.90776576e-05 5.57601914e-05 5.98991449e-02]
[-2.26628928e-02 -7.61831309e-02 1.00000000e+00]]
Essential matrix:
[[-0.3258429   -3.89878096   3.72255685]
[3.59524641   2.25052649 12.03955779]
[-4.54972041 -15.30683186   0.9959223]]

2 kinds of viewing angle:

Discussion

1. When using RANSAC to find Fundamental matrix, there are a lot of possible choice to choose. We only choose 8 pairs of corresponding feature points from all these correspondence. The best Fundamental matrix varies every time as we rerun the program. So we fixed the seed of numpy.random.RandomState(). The Fundamental matrix then becomes stable.
2. The distance ratio of SIFT and the threshold of inliers in RANSAC are two hyperparameters we can tune for.
   If the distance ratio is big, we will find more feature points, but false inliers will also increase, and finally make the structure graph little bit weird. By experiment, we set distance ratio to 0.7.
   If the threshold of inliers in RANSAC is too big, false inliers will also increase. By experiment, we set distance ratio to 0.01.

Conclusion

In this report we first introduce the implementation step of Structure from Motion. We use SIFT to find the interesting points and 8-points to find the fundamental matrix. Using fundamental matrix to get essential matrix and then we get the four possible result, and choose the best one to be our SfM algorithm result. Finally we use matlab function to map image texture to the points and get our model.

Work assignment plan between team members

- main code 吳承翰
- testing result 謝秉瑾
- report 謝秉瑾,謝宗祐