HW04 Report

1. Introduction

Homework 4 is to explore fundamental steps in Structure from Motion (SfM) using stereo images and intrinsic camera calibration data. The objective is to implement and analyze critical tasks such as detecting correspondences between two images, estimating the fundamental matrix, and visualizing epipolar geometry through epipolar lines.

2. Implementation

a. Find out correspondence across images

We employ the SIFT (Scale-Invariant Feature Transform) algorithm to establish correspondences between two images. The SIFT detector is robust to scale, rotation, and illumination changes. After extracting keypoints and descriptors for both images, we use the FLANN (Fast Library for Approximate Nearest Neighbors) matcher to find potential matches. Parameters such as trees and checks in FLANN are adjusted to balance speed and accuracy.

1. Keypoint Detection:

- SIFT is applied to each grayscale image to detect local features.

2. Feature Matching:

- FLANN is configured with KD-Tree indexing (trees=5) and search (checks=50) parameters.
- Lowe's ratio test is applied to filter ambiguous matches.

b. Estimate the fundamental matrix across images

Using the correspondences from the previous step, we calculate the fundamental matrix (F) through the normalized 8-point algorithm. The fundamental matrix encapsulates the epipolar geometry between two views, mapping points in one

image to their corresponding epipolar lines in the other image.

1. Normalize the coordinates of matching points for better numerical stability.
2. Solve the homogeneous equation using Singular Value Decomposition (SVD) to find F.
3. Enforce the rank-2 constraint by setting the smallest singular value of F to zero.

c. Draw the interest points you found in Step 1 in one image and the corresponding epipolar lines in another.

To visualize the epipolar geometry, we draw:

1. Epipolar Lines:
   - Lines corresponding to interest points in one image are drawn in the other.
2. Interest Points:
   - Matching points are marked on both images.

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

In *get_essential_matrix()* function, We can obtain the essential matrix by the formula below

$$\mathbf{E} = \mathbf{K}_1^T \mathbf{F} \mathbf{K}_2$$

and by applying the function, *np.linalg.svd()*, we get the matrix U, S, V_T. Change the matrix S into diag(1, 1, 0) and return the essential matrix.

In *get_camera_matrix_choices()* function, the first camera matrix, **P1** = [**I** | **0**], there are four possible choices for the second camera matrix **P2**,

$$\mathbf{P}_2 = \left[\mathbf{U}\mathbf{W}\mathbf{V}^T \middle| +\mathbf{u}_3\right]$$

$$\mathbf{P}_2 = \left[\mathbf{U}\mathbf{W}\mathbf{V}^T \middle| -\mathbf{u}_3\right]$$

$$\mathbf{P}_2 = \left[\mathbf{U}\mathbf{W}^T\mathbf{V}^T \middle| +\mathbf{u}_3\right]$$

$$\mathbf{P}_2 = \left[\mathbf{U}\mathbf{W}^T\mathbf{V}^T \middle| -\mathbf{u}_3\right]$$

and matrix W is

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

so the function finally return the **R2 array** and the **t2 array**, where the **R2 array** includes UWV^T, UWV^T, UW^TV^T, UW^TV^T and the **t2 array** inclueds +u3, -u3, +u3, -u3.

e. Find out the most appropriate solution of essential matrix

After obtaining the four possible solutions, we apply the *get_solution()* function to find the solution. By using the *triangulation()* function, the corresponding 3D points and the number of 3D points in front of cameras can be obtained. We compare the number of 3D points in front of cameras in different **P2** matrices and choose the matrix with the most 3D points in front of cameras. Finally, return the 3D points, the number of 3D points in front of cameras, **P1** and **P2**.

f. Apply triangulation to get 3D points

The process of the linear solution in *triangulation()* function is that for the given points1, points2, P1 and P2, get the prediction points and projection matrices. Next, create the matrix **A** as the formula below

$$\mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^T - \mathbf{p}_1^T \\ v\mathbf{p}_3^T - \mathbf{p}_2^T \\ u'\mathbf{p}_3'^T - \mathbf{p}_1'^T \\ v'\mathbf{p}_3'^T - \mathbf{p}_2'^T \end{bmatrix}$$

and apply the *np.linalg.svd()* function to obtain **V_T** and then get the positions in 3D space.

To calculate the number of 3D points in front of the cameras, we only need to check whether the output result conform to the formula below

$$(\mathbf{X} - \mathbf{C}) \cdot \mathbf{R}(3,:)^T > 0$$

and return the corresponding results.

g. **Use texture mapping to get a 3D model**
We directly used the provided function **obj_main** to achieve this. Here we use matlab.engine to directly call matlab functions from python.

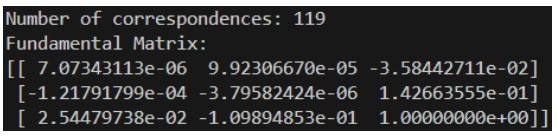
h. **Use 3d software like Blender to visualize the resulting 3D model**
Here we use the *fig.add_subplot(111, projection = '3d')* function in *plot_points3d()* to plot the 3D points on the 3D coordinates.

3. Experimental result

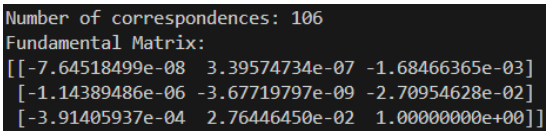
Using the SIFT detector and FLANN matcher, [N] correspondences were successfully identified between the two images.
Lowe's ratio test ensured robust matching by removing ambiguous matches.

Estimate Fundamental Matrix

The calculated fundamental matrix is:



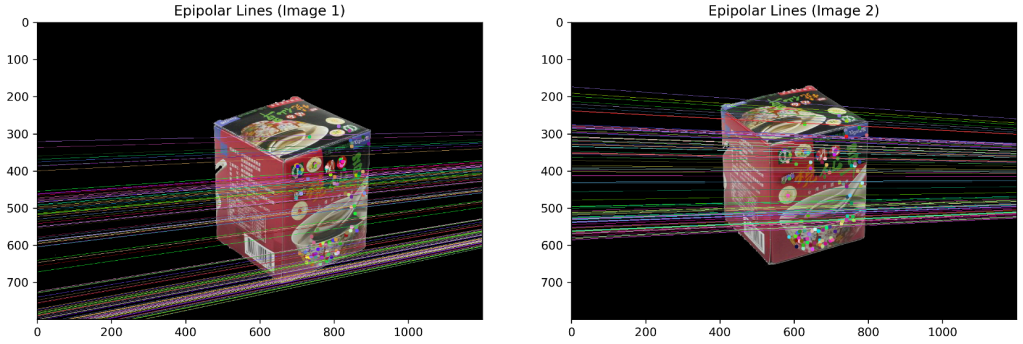
fundamental_matrix_Mesona.png



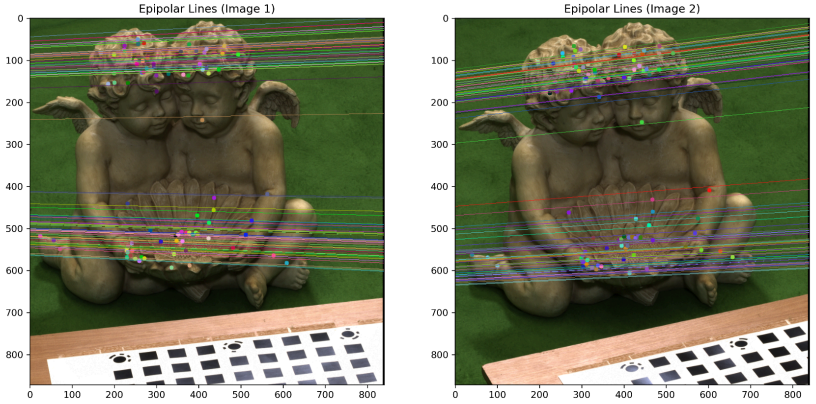
fundamental_matrix_Statue.png

Epipolar Lines

The following figure shows the detected keypoints and corresponding epipolar lines:
Left Image: Keypoints with their corresponding epipolar lines in the right image.
Right Image: Keypoints with their corresponding epipolar lines in the left image.

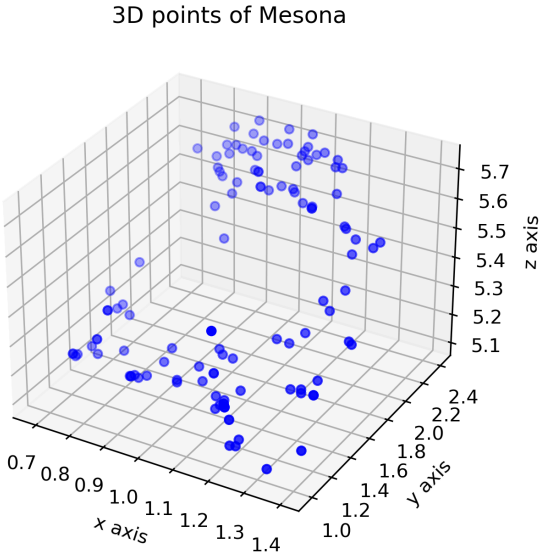


Mesona_epipolar-lines.png

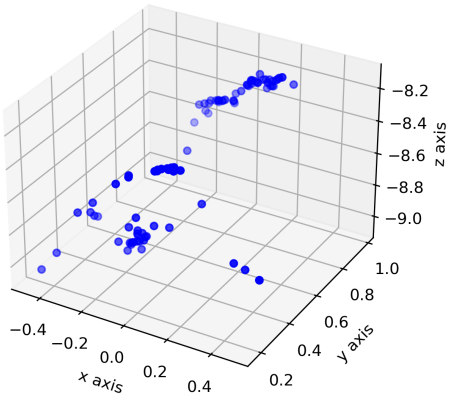


Statue_epipolar-lines.png

After finding out the most appropriate solution of the essential matrix, we obtain the 3D points of the images and plot them on the 3D coordinates.

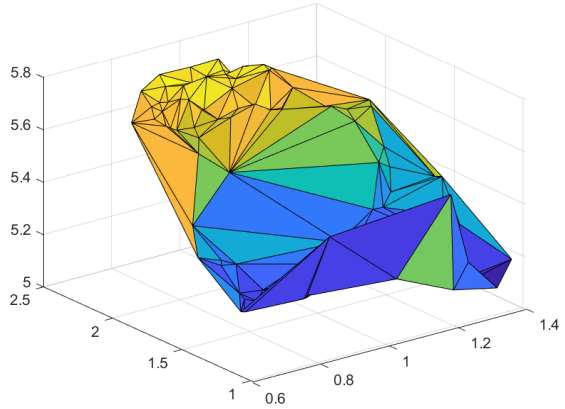


Mesona_3Dpoints.png

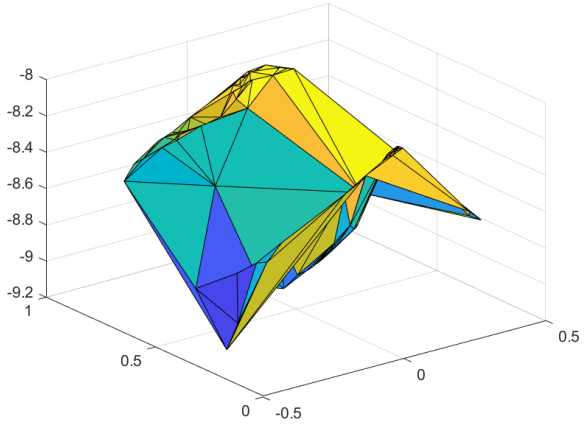


Statue_3Dpoints.png

By implementing the file, obj_main.m, we obtain the mesh-triangulation images



mesh_triangulation_Mesona.png



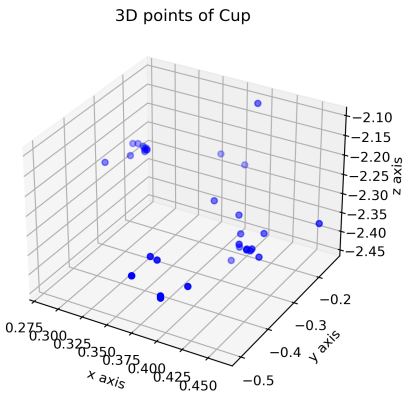
mesh_triangulation_Statue.png

Using our data as the input, we get

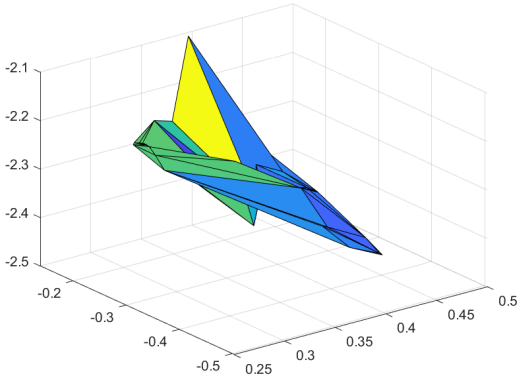
```
Number of correspondences: 35
Fundamental Matrix:
[[-1.06748260e-08  2.19489680e-07 -4.78982501e-04]
 [-9.80089421e-08  1.46491128e-08  1.36639770e-03]
 [ 2.93309061e-04 -1.81015556e-03  1.00000000e+00]]
```

fundamental_matrix_Cup.png

Cup_epipolar-lines.png



Cup_3Dpoints.png



mesh_triangulation_Cup.png

4. Discussion

- During the calculation of the fundamental matrix, we noticed that some noise in the correspondences we detected caused some deviations. To address this, we adjusted the threshold in Lowe's ratio test to filter out unreliable matches, improving the robustness.
- The parameters in FLANN, like the number of trees and checks, significantly affected the matching accuracy. If not properly tuned, the results could be unreliable. To fix this, we spent time fine-tuning these parameters for different datasets, finding configurations that produced the best matches.
- Matrix computation was unstable because of the unbalanced or large range of feature points. To solve this, we normalized the keypoints, making the computation much more stable and reliable.
- Testing on a single dataset wasn't enough to validate the method's generalizability. To overcome this, we tested the implementation on multiple datasets to see how it performs in different scenarios. This helped us refine the parameters and ensure consistent results.
- In the triangulation*()* function, it should be careful when obtaining the matrix **A**. Since the corresponding terms may be wrong in coding which causes wrong results.
- In the *texture_mapping()*, when compiling the *eng.obj_main()*, we also get the error in the begining. Since we didn't change the variable into the *matlab.double()* type.
- It takes a lot of time to complete the homework this time, since the process in SFM is quite complex, and we spent a long time understanding the purpose of each step.

5. Conclusion

In this homework, we implemented the pipeline of Structure from Motion (SfM) . Using SIFT and FLANN, we can find the keypoints and their correspondences in two stereo images. The accuracy of which is then verified by visualizing epipolar lines and keypoints. By the fundamental matrix and essential matrix we calculated through the correspondences, we know the camera matrices of the cameras, which are then used in triangulation to get 3D points. Finally, we conduct texture mapping to construct the model and visualize it using Blender.

Overall, this approach worked really well for exploring the basics of 3D reconstruction, and the results aligned very well with what we expected in theory after some fine tuning.

6. Work assignment plan between team members.

彭宇楨:34% , 洪義翔:33% , 張宥楨: 33%