HW4: SfM

Group 25

Introduction

Structure from Motion (SfM) is a technique that reconstructs 3D structures by analyzing correspondences between images. It is widely used in fields such as robotics, autonomous vehicles, and augmented reality. This assignment aims to implement the key steps of SfM, including feature matching, estimation of the fundamental and essential matrices, triangulation of 3D points, and generation of a 3D model. Additionally, the pipeline is applied to self-captured images to practice camera calibration and 3D reconstruction.

Implementation procedure

Step 1

To establish correspondences between two images, we used the SIFT (Scale-Invariant Feature Transform) algorithm to detect keypoints and compute their descriptors. The key points were then matched using the BFMatcher with KNN (k-Nearest Neighbors) matching. Lowe's ratio test was applied to filter out unreliable matches. The resulting correspondences are the input for subsequent steps.

Step 2

The fundamental matrix F was estimated using the normalized eight-point algorithm.

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

To improve numerical stability, we first normalized the matched points by translating them to the origin and scaling them to have unit variance. Using RANSAC, we refined FFF by iteratively sampling subsets of points to minimize epipolar constraints and identifying inliers.

Step 3

Using the fundamental matrix F, we computed the epipolar lines corresponding to the matched points in each image. For visualization, we overlaid the epipolar lines on one image and plotted the corresponding matched points on the other. This step verified the accuracy of F by ensuring that the points lie near their corresponding epipolar lines.

Step 4

After calculating the fundamental matrix, we need to compute the essential matrix E using the camera's intrinsic matrix K. The essential matrix describes the relative pose between two cameras (i.e., rotation and translation), allowing us to represent the correspondence of 3D points in two images. The formula for computing the essential matrix is:

$$E = K^T F K$$

Where F is the fundamental matrix calculated from the corresponding points between the two images. By transforming the fundamental matrix into the essential matrix, we incorporate the intrinsic parameters of the cameras, which allows us to infer the relative pose between the two cameras.

Step 5

Next, we need to extract the camera projection matrices from the essential matrix. Projection matrices describe the position and orientation of the camera, helping to map 3D world points into the 2D image plane. In this step, we decompose the essential matrix using Singular Value Decomposition (SVD) to obtain matrices U, S,V^T, and then use the matrix W to form the rotation matrices R and the translation vector t.

Specifically, after performing SVD on the essential matrix, we derive four possible solutions, which include two rotation matrices R1 and R2, as well as two possible directions for the translation vector (t and −t). To ensure the rotation matrices have the correct orientation, we check their determinants and negate them if necessary.

Ultimately, we obtain four potential projection matrices:

1. [R1∣t]
2. [R1∣−t]
3. [R2∣t]
4. [R2∣−t]

These projection matrices represent four possible poses for the second camera.

Step 6

In this step, we need to determine which projection matrix is the most appropriate. To achieve this, we apply triangulation to project the corresponding points from the images into 3D space, thereby reconstructing their 3D coordinates.

First, we use the two projection matrices P1 and P2 to triangulate the corresponding points and calculate the homogeneous coordinates of the 3D points. Here, we construct matrix A and solve it using SVD to find the best 3D point X. We then triangulate the points using each of the possible projection matrices and calculate the depth values (Z) of the reconstructed 3D points relative to both cameras. Finally, we select the projection matrix that yields the highest number of points with positive depth values.

This process ensures that the selected projection matrix is the one that places the maximum number of 3D points in front of both cameras, meaning that these points are visible in the real-world scene for both cameras. Thus, we can determine which set of projection matrices accurately describes the relative position and orientation of the cameras.

Step 7

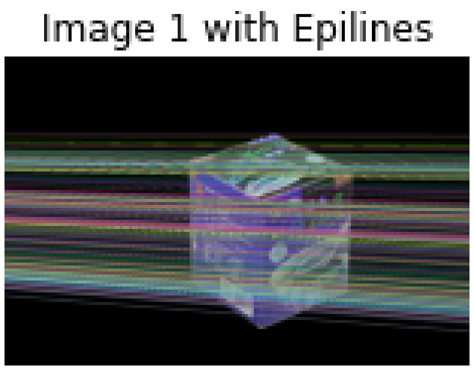
Use The points_3d, inliner_pts1 and K to call the function TA provided in matlab, to get the .mtl and .obj file. (parameter passing by .txt)

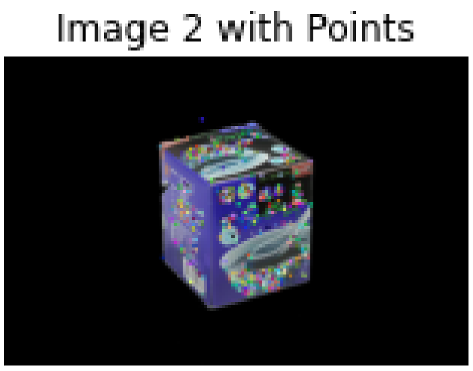
Step 8

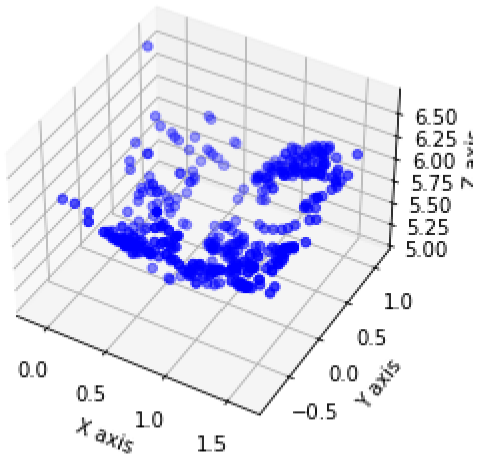
Use blender to reconstruct the model by .mtl and .obj file

Experimental results

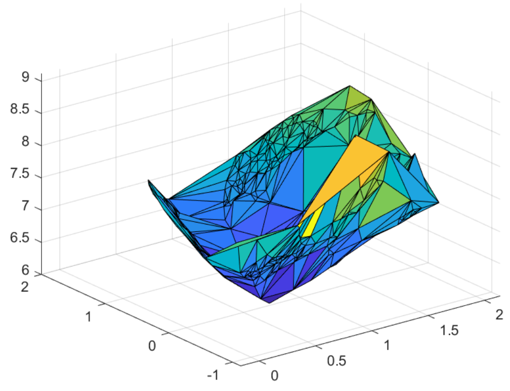
Medona:

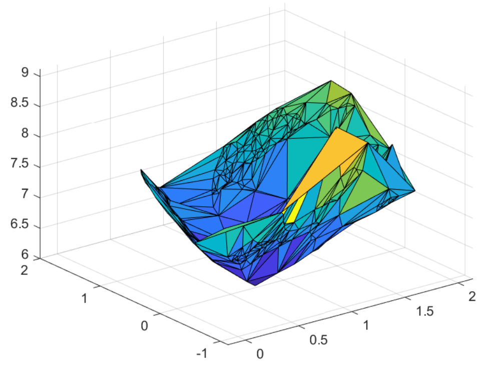




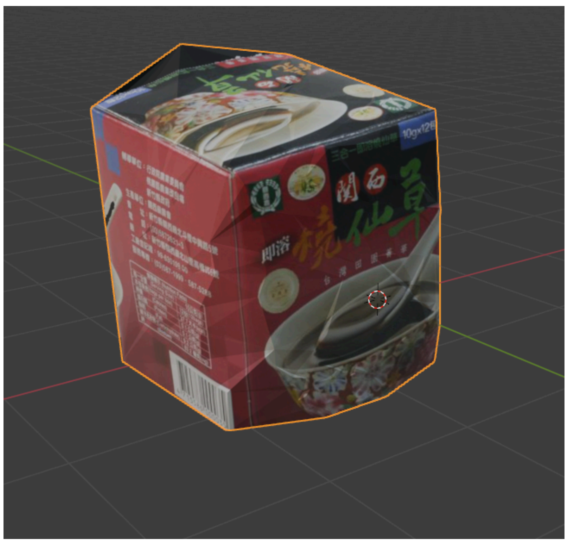


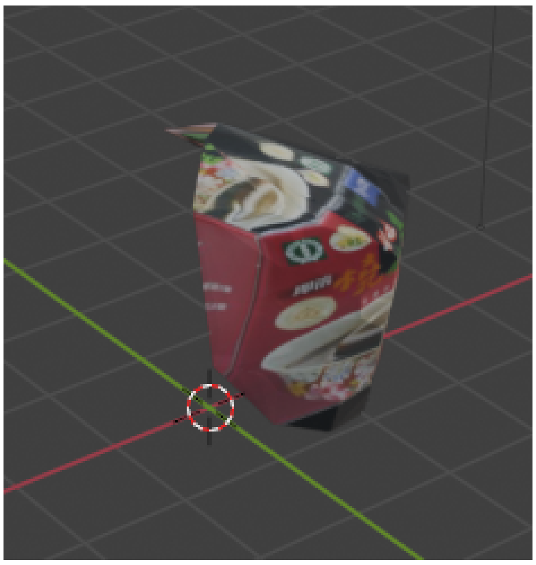
matlab:(Mesona1 and Mesona2 respective)



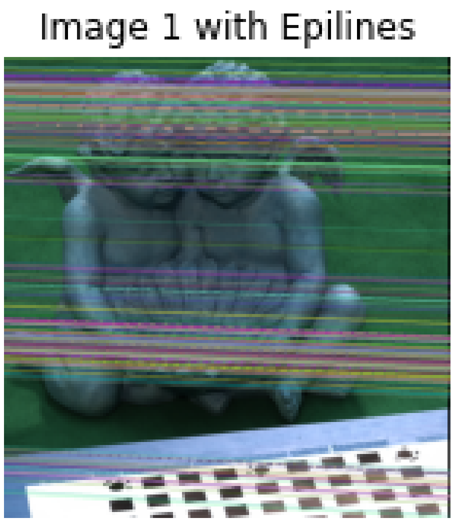


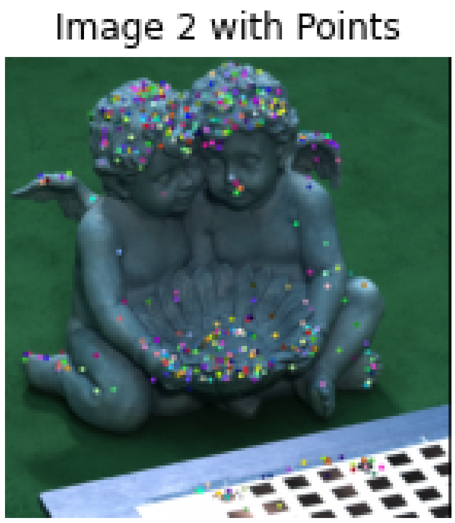
model reconstruct:(Mesona1 and Mesona2 respective)

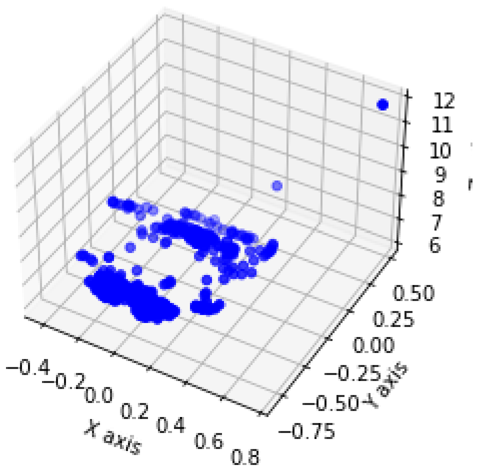




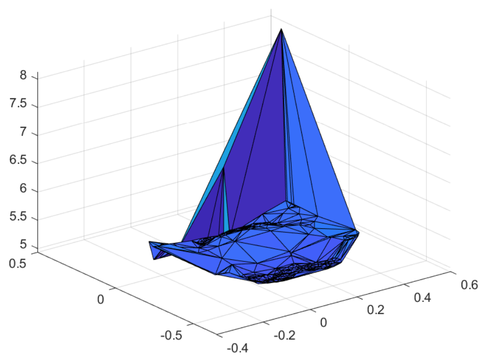
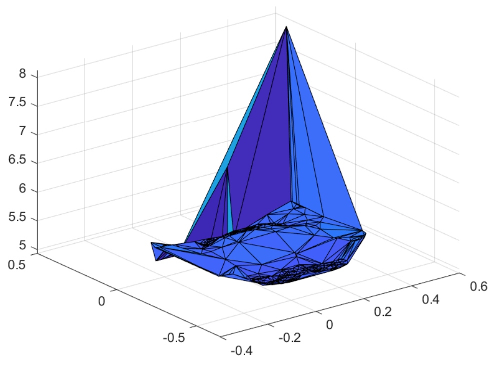
Statue:



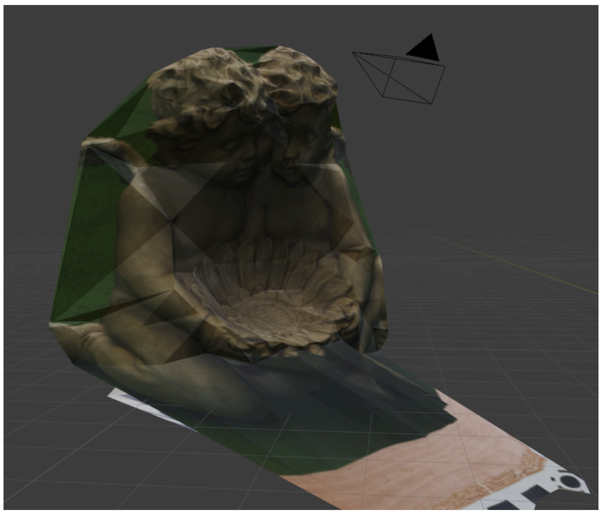
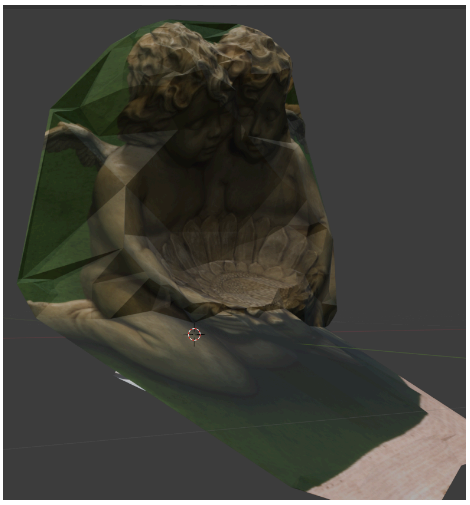




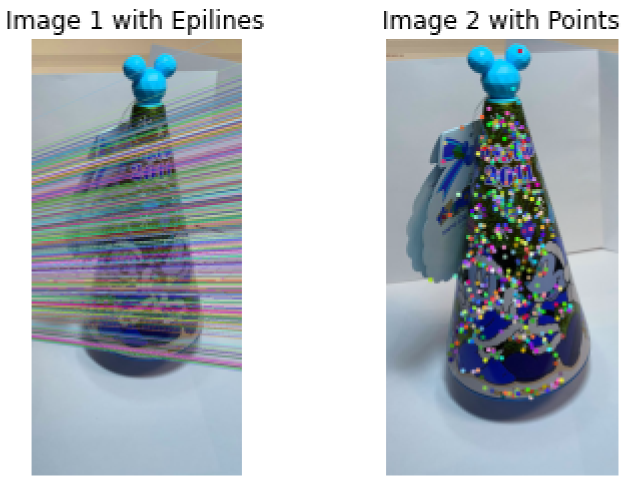
matlab: (Statue1 and Statue2 respective)

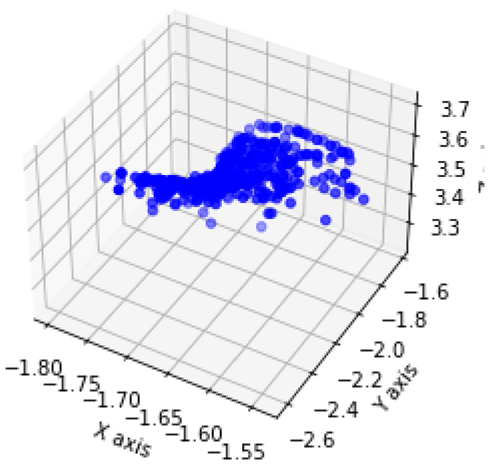



model reconstruct:(Statue1 and Statue2 respective)

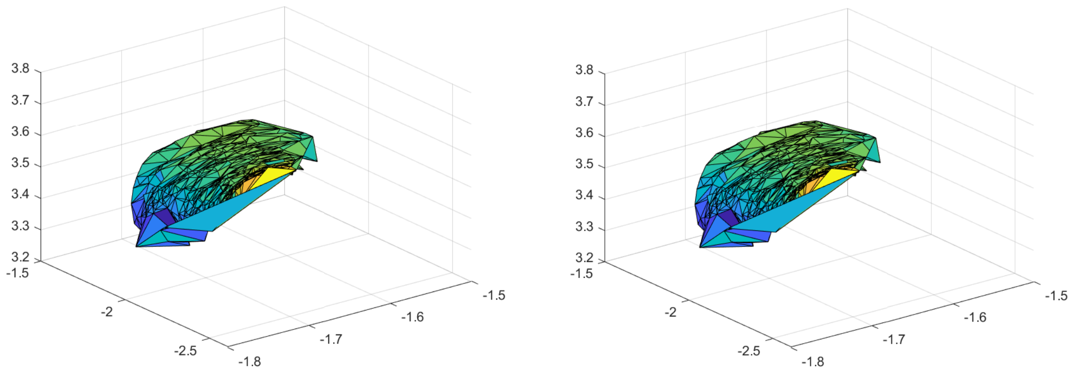



other data:
SantaHat:

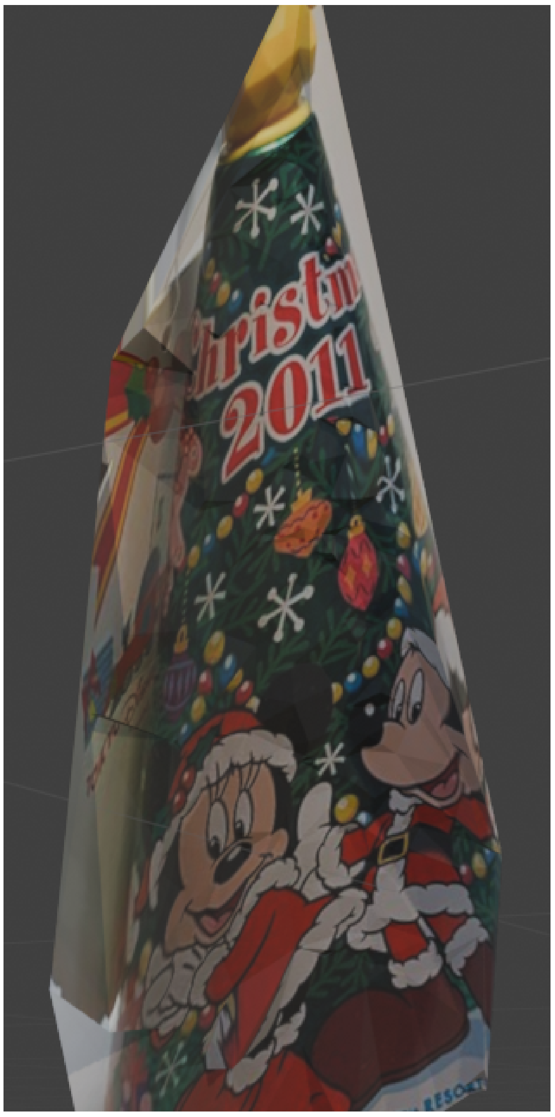


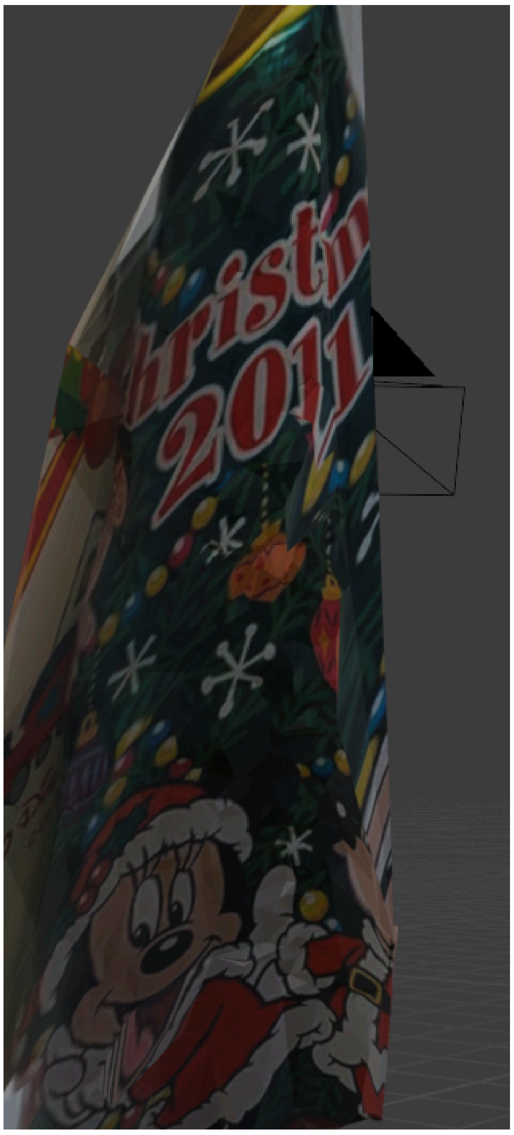


matlab(SantaHat1 and SantaHat2)



model reconstruct(SantaHat1 and SantaHat2)





Discussion

1. **Insufficient feature points in our own data**
   When applying the Structure from Motion (SfM) pipeline to our self-captured images, the number of detected and matched feature points was relatively small. This might be due to insufficient texture or repetitive patterns in the scene, which made it difficult for SIFT to detect distinct keypoints. As a result, the 3D model generated in MATLAB appeared distorted and lacked proper structure.
2. **Improper threshold for the fundamental matrix estimation**
   While estimating the fundamental matrix, the threshold for the epipolar constraint was set too small. This caused the RANSAC algorithm to reject too many inliers, resulting in an inaccurate fundamental matrix. Consequently, the epipolar lines drawn on the images did not align well with the corresponding points. To resolve this, we adjusted the threshold to a more appropriate value.

Conclusion

In this experiment, we successfully reconstructed a 3D point cloud model by combining the computation of the essential matrix, extraction of projection matrices, and triangulation techniques. We began by calculating the essential matrix using the fundamental matrix and the camera's intrinsic matrix, which allowed us to infer the relative pose between the cameras. Then, we used four possible projection matrices for triangulation and selected the one that resulted in the highest number of 3D points in front of both cameras, ensuring the correctness of the camera poses.

This experiment provided us with a deeper understanding of Structure from Motion (SfM) and how to address multi-view geometry problems. These techniques are not only applicable in photogrammetry and computer vision but also lay a solid foundation for future endeavors in 3D reconstruction and augmented reality applications.

Work assignment plan between team members

王唯誠:coding
黃竑睿:coding
鍾任軒:coding + output generate

file structure

root
– data
  – the image TA provided
– my_data
  – the image ourselves
– output
  – object name
    – some data used to transport parameters to Matlab(.txt)
    – matlab output .png
    – model file (.obj & .mlt)
    – blender file (.blend)
– main.py: main code
– py2mat.m: used to read the parameters from python and run the code provided by TA.

– some other file provided by TA