# EN.520.665 Machine Perception

# Project 2 Report

Team member: Zhiwei Gong, Pupei Zhu

## 1. Introduction

In this project, the factorization method for structure from motion for orthographic camera cases is implemented on two datasets: Castle Sequence and Medusa Head. The task consists of two parts: (i) Using KLT to track feature points. (ii) Using factorization approach to extract shape and motion parameters. The results of the Tomasi-Kanade algorithm show that it can reconstruct the object to some extent, but the effects are limited by the texture and illumination.

## 2. Methodology

### 2.1 KLT feature tracker

In this project, the feature point tracker we use is Kanade–Lucas–Tomasi (KLT) feature tracker [1]. Good features are located by examining the minimum eigenvalue of each 2 by 2 gradient matrix, and features are tracked using a Newton-Raphson method of minimizing the difference between the two windows. Multiresolution tracking allows for relatively large displacements between images.

In our implementation, to decide the tracking points, we use **cv.goodFeaturesToTrack()**. Firstly, we choose the first frame in the video and delete some Shi-Tomasi corner points in it, following that, utilizing Lucas-Kanade optical flow, we iteratively track these points. **cv.calcOpticalFlowPyrLK()** is used to calculate the optical flow. We pass the previous frame, previous points and subsequent frame. It returns next points with binary status numbers, which have a value of 1 if the next point is found and a value of 0 otherwise. These next points are passed as previous points in the following phase recursively.

### 2.2 Factorization method

After tracking the feature points, the factorization method [2] is implemented for estimation of object shape and camera motion from image stream. Through the tracking of P points over F frames of images, the 2FxP measurement matrix W is constructed. In order to use the rank theorem, the rotation and translation are measured with respect to the object's centroid. The rotation matrix is described by: both rotation (R) and translation (T) of the dimensions 2Fx3 and 3xP, respectively. In order to estimate the rotation and translation matrices, we use the factorization approach after obtaining the SVD decomposition. To solve for matrix Q, we use the cholesky decomposition. Finally, we align the first camera reference system with the world reference system.

# 3. Experiment

## 3.1 Experiment Setup

### 3.1.1 Dataset

In this project, two datasets: (i) Castle Sequence; (ii) Medusa Head are used. Castle Sequence consists of 27 images describing the castle shape. Medusa Head is a video consisting of 350 frames.

### 3.1.2 Implementation Details

In our experiments, we first start with image frame reading. For both medusa video and castle images, we use opencv to read the frame by frame images.

Then we use **goodFeatresToTrack** to detect features with high quality, the parameters we set for feature detection and tracking on these two datasets are shown in Table.1. (i) minDistance: minimum possible Euclidean distance between the returned corners. (ii) blockSize: size of an average block for computing a derivative covariation matrix over each pixel neighborhood. (iii) winSize: size of the search window at each pyramid level. (iv) maxLevel: 0-based maximal pyramid level number; if set to 0, pyramids are not used (single level), if set to 1, two levels are used, and so on; if pyramids are passed to input then algorithm will use as many levels as pyramids have but no more than maxLevel. (v) qualityLevel: parameter characterizing the minimal accepted quality of image corners. The parameter value is multiplied by the best corner quality measure, which is the minimal eigenvalue or the Harris function response. The corners with the quality measure less than the product are rejected.

| | Castle Sequence | Medusa Head |
|---|---|---|
| Start Frame | 1 | 100 |
| End Frame | 27 | 260 |
| minDistance | 3 | 3 |
| blockSize | 7 | 7 |
| winSize | (40,40) | (40,40) |
| maxLevel | 3 | 3 |
| quanlityLevel | 0.01 | 0.01 |
| macCorners | 5000 | 5000 |

Table 1: Parameters for KLT feature detection and tracking

After detecting the good features, we start to use optical flow to track the features. Then, we use the tracked features to form the W matrix in the paper.

In the next stage, we implemented the method from the paper, we firstly calculated the R_hat and S_hat, and then used cholesky decomposition to solve the Q, and after that, we obtained the R and S matrix.

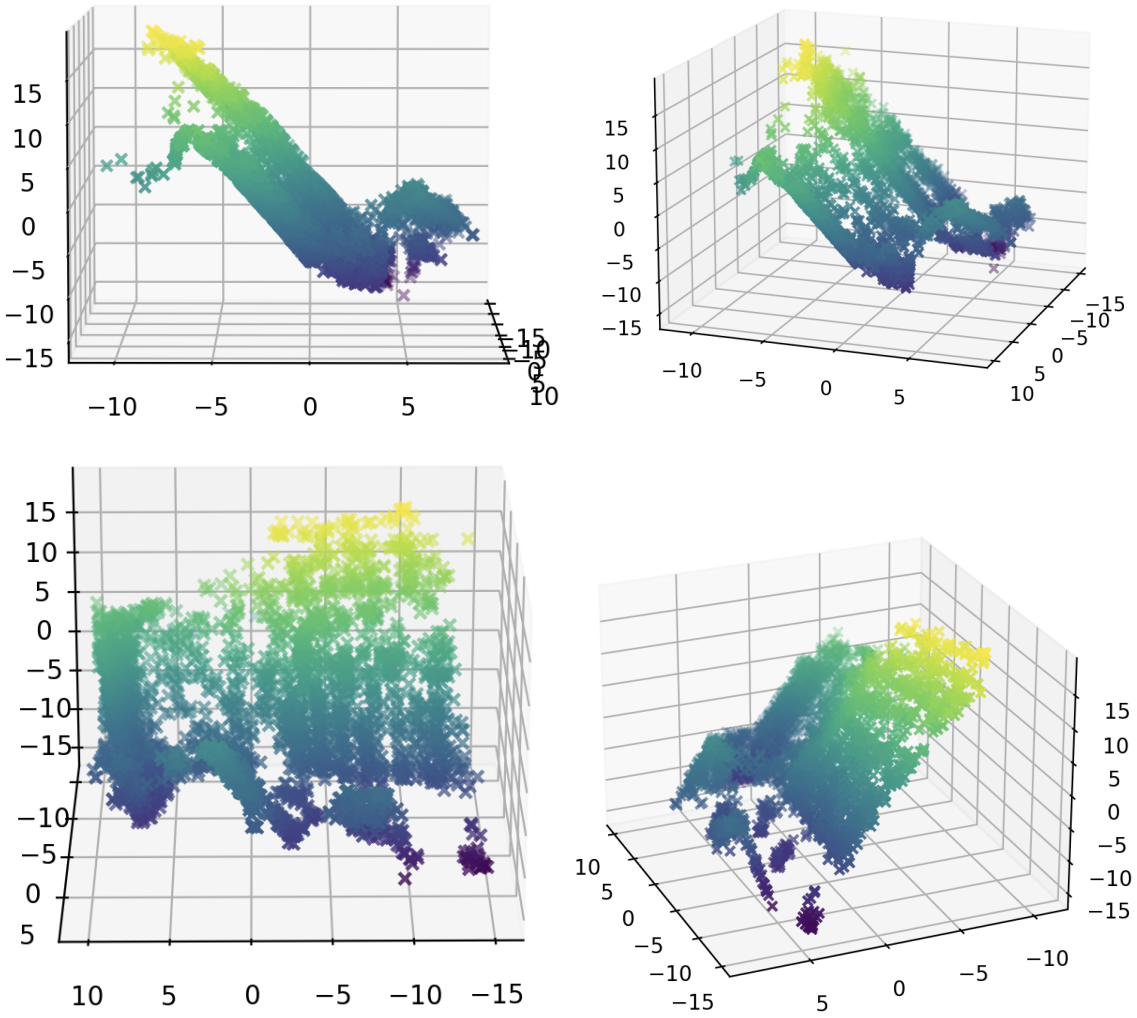
Finally, we plotted the 3D features from the S matrix. And transferred the world coordinates of the feature points to the pose of the first image.

### 3.1.3 Results

#### 3.1.3.1 Castle Sequence

For the Castle Sequence dataset, we set the maximum number of corners to return be 5000, after applying the KLT tracker, we get 3597 remaining feature points. Fig. 1 shows one of the original castle image sequences. According to the results in Fig. 2 and Fig. 3, we can see that the shape of 3D feature points we plot is similar to the castle shape.

Figure 1: Castle image



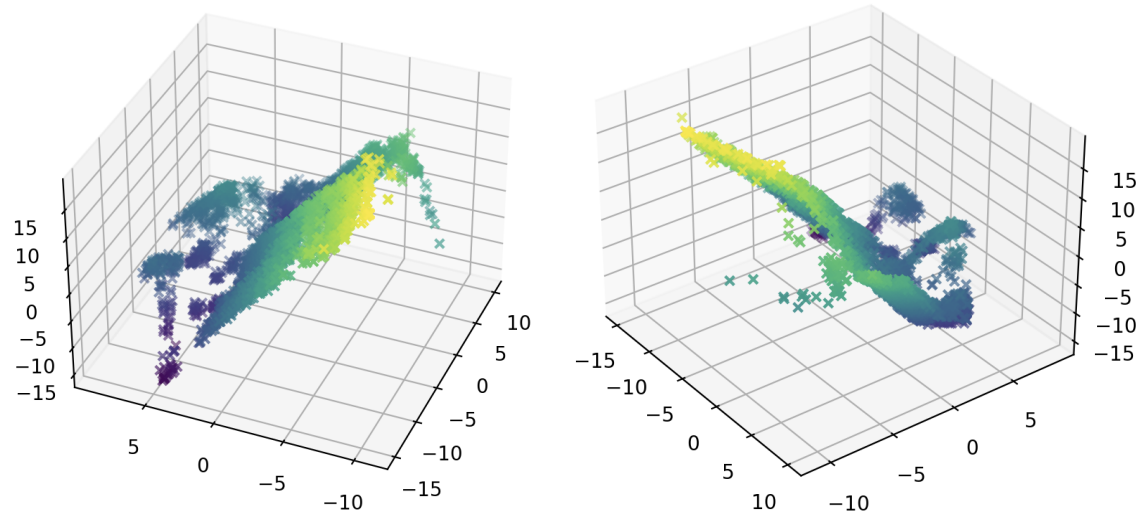


Figure 2: Structure from motion results of Castle image - different views of feature 3D point cloud

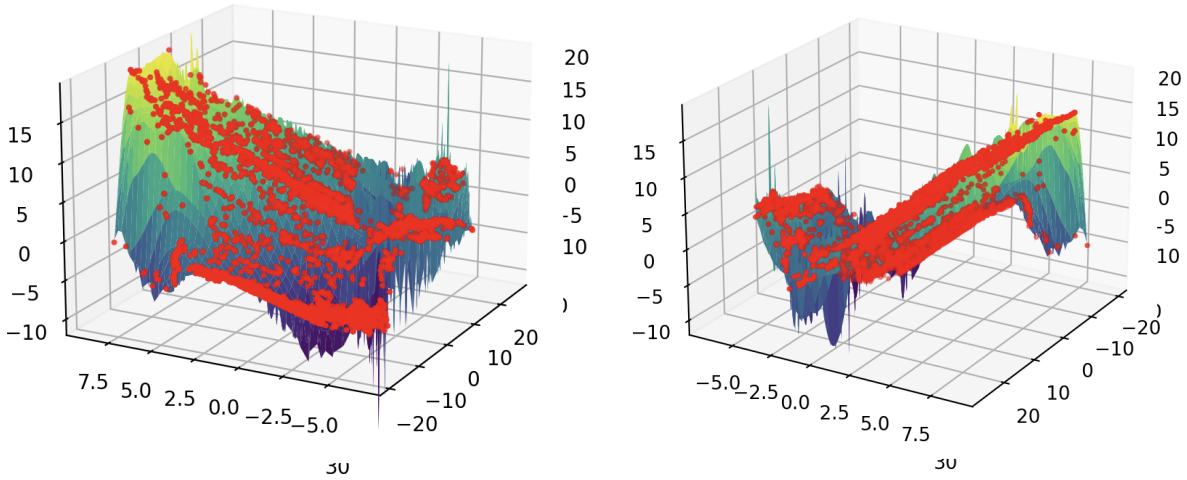


Figure 3: Shape display of Castle after aligning the first camera reference system with the world reference system

### 3.1.3.2 Medusa Head

For the Medusa Head dataset, we set the maximum number of corners to return be 5000, after applying the KLT tracker, we get 3993 remaining feature points. Fig. 4 shows different views of the medusa head sequences. According to the results in Fig. 5 and Fig. 6, we can see that the shape of 3D feature points we plot is similar to the medusa head shape.

Figure 4: Medusa Head images

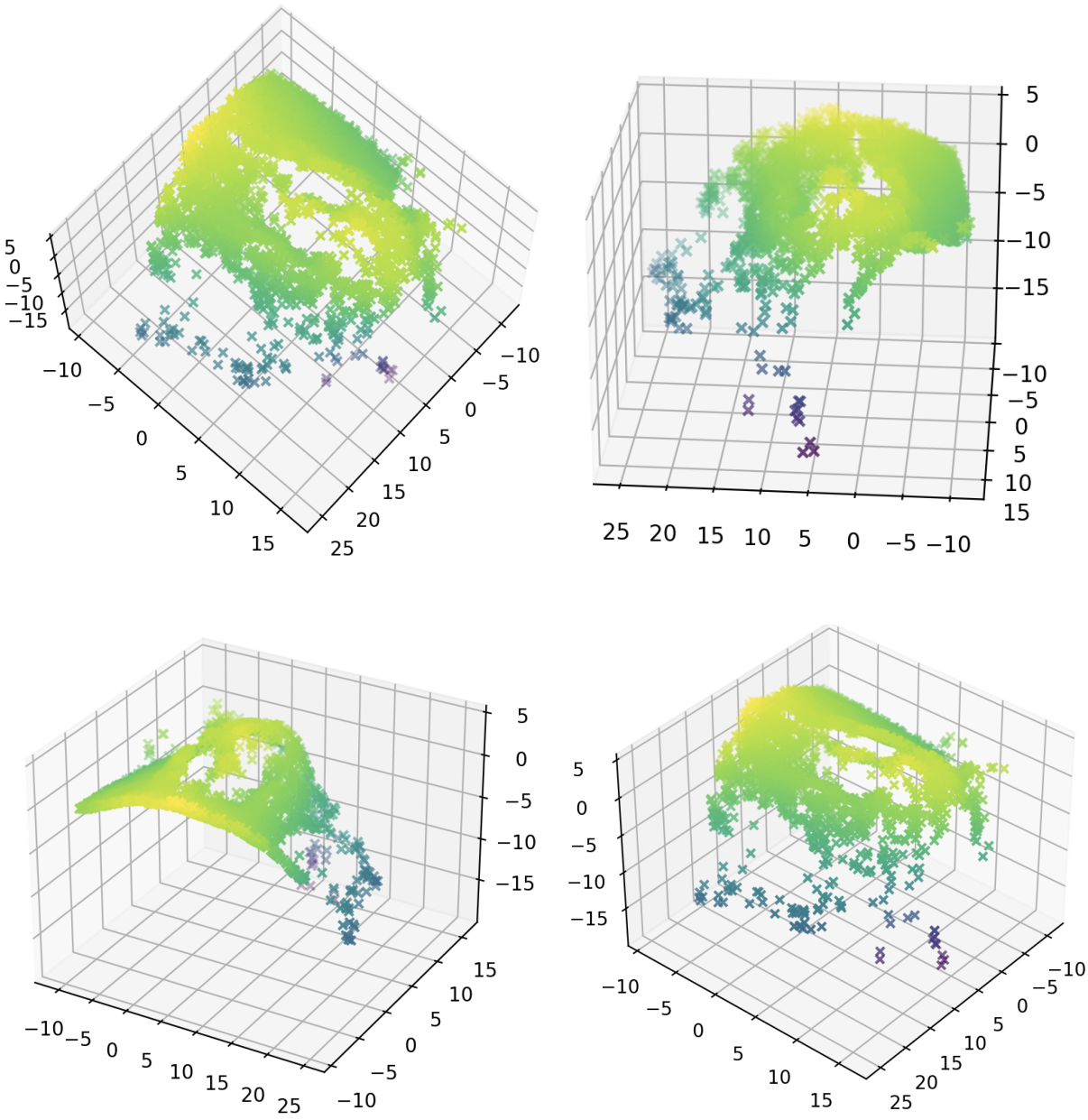


Figure 5: Structure from motion results of Medusa Head - different views of feature 3D point cloud

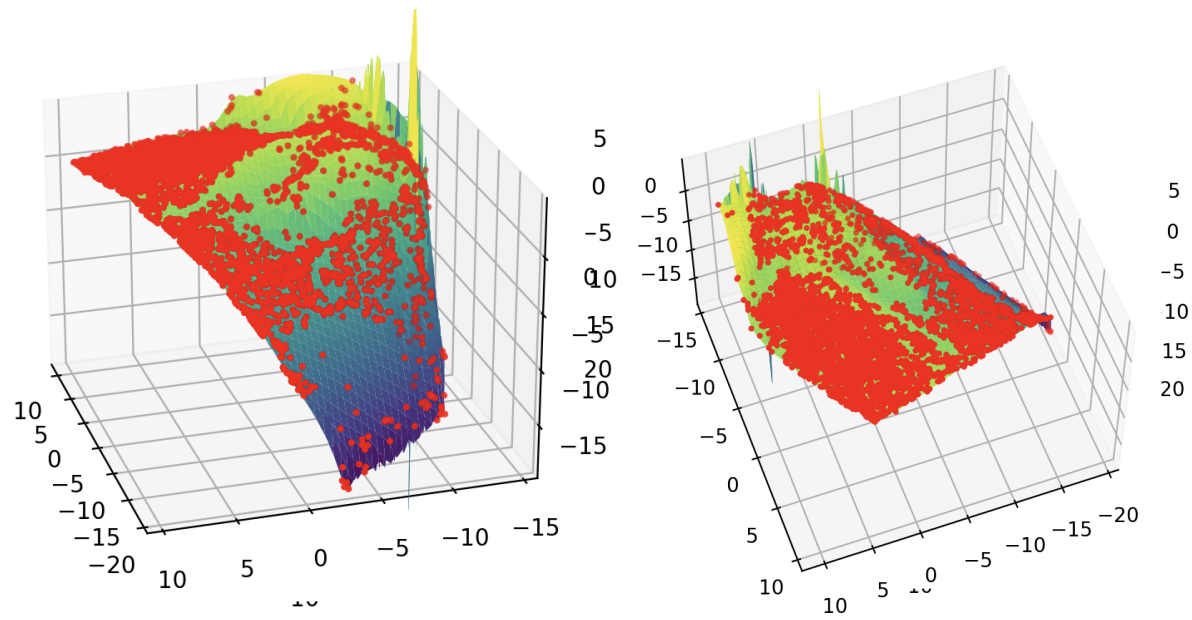


Figure 6: Shape display of medusa head after aligning the first camera reference system with the world reference system

## 4. Discussion and Conclusion

According to the above results, we can conclude that the Tomasi-Kanade algorithm can reconstruct the object to some extent, but the performance on our cases is not good enough. This might be caused by the effects of the texture and illumination. Tomasi-Kanade algorithm is that it assumes a static scene, where the 3D structure of the objects in the scene does not change over time. The algorithm assumes that the 3D structure of the scene is static, but in our cases, there are some dynamic factors in the scene, specially for castle images. This can cause the algorithm to produce inaccurate results. The algorithm also assumes that the 2D images are perfect, but in our cases, images like castle sequence may be noisy or distorted due to various factors such as lighting conditions or camera motion. This can also cause the algorithm to produce inaccurate results. Another potential failure mode of the Tomasi-Kanade algorithm is that it relies on the availability of high-quality images with sufficient texture and detail. For our datasets, they might not contain enough texture, the algorithm may fail to accurately estimate the 3D structure of the scene. Finally, for the medusa video, it looks that the video is concatenated by two part, thus the feature will be lost during the two parts.

## Reference

[1] Jianbo, Shi, and Carlo Tomasi. "Good features to track." In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 593-600. 1994.

[2] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography-- a factorization method," International Journal of Computer Vision, 9(2):137--154, 1992.