# HW3 - Structure from Motion


Group 28
Wen-Jie Tseng (0556146)
Li-Yang Wang (0656619)
Yi-Chen Lee (0656611)
Computer Vision - 2018 Spring Semester


## 1. INTRODUCTION

Structure from motion (SfM) is a photogrammetric range imaging technique for estimating three-dimensional structures from two-dimensional image sequences. In this assignment, we tried to reconstruct 3D points from two given image pairs. The intrinsic matrix $K$ was given. The key points and descriptors were extracted from a pair of images by Scale-Invariant Feature Transform (SIFT). We used Fast Library for Approximate Nearest Neighbors (FLANN) based matcher and applied ratio test to obtain matches in two images. Then we ran eight-point algorithm with RANSAC to get a robust estimated fundamental matrix. After having fundamental matrix, the epipolar lines and essential matrix can be found. Based on essential matrix, we obtained four possible solution of the second camera matrix. With triangulation, one can convert image pixels into 3D coordinate. At last, we checked which solution has the most points in front of itself and applied given matlab code to get the final .obj file.

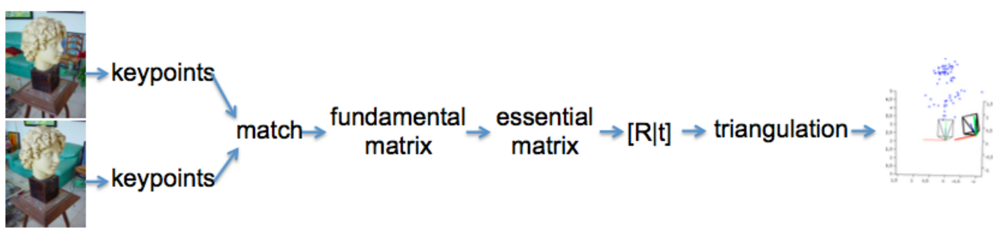


Figure 1. The workflow of 3D reconstruction.

## 2. IMPLEMENTATION DETAILS

### 2.1 Find out Correspondence across Images

We use SIFT to extract features and FLANN based matcher to obtain matches. The implementation are developed in **find_matches** function.

### 2.2 Estimate Fundamental Matrix with RANSAC

First, we introduce eight-point algorithm to estimate fundamental matrix F. The fundamental matrix is defined as $x'^T F x = 0$. This provides a linear equation,

$$[uu^{'} f_{11} + vu^{'} f_{12} + u^{'} f_{13} + uv^{'} f_{21} + vv^{'} f_{22} + v^{'} f_{23} + uf_{31} + vf_{32} + f_{33}] = 0$$

where $x = [u, v, 1]^T$ and $x^{'} = [u^{'}, v^{'}, 1]^T$. Instead of solving $Af = 0$, we seek f to minimize $||Af||$, the least eigenvector of $A^T A$. Since F has rank 2, F is replaced by F' that minimizes $||F - F^{'}||$ subject to $det(F^{'}) = 0$. This can be achieve by Singular Value Decomposition (SVD). Replace the third diagonal element in $\Sigma$. The estimated fundamental matrix will become $F^{'} = U\Sigma^{'} V^T$. One problem of the equation above is that the orders of magnitude difference between column of data matrix will yield poor results. Thus we have to apply normalization before eight-point algorithm, transform image to [-1, 1] x [-1, 1]. After computing fundamental matrix, we have to denormalized it back to original scale. We implement these steps in **normalize_2d_pts**, **constraint_matrix**, and **eight_point_algorithm** functions.

To obtain a robust estimation of $F$, we implemented eight-point algorithm with RANSAC. This pseudocode shows as follows,

1. Sample 8 points to obtain a fundamental matrix F
2. Compute Sampson distance to get test error from rest of the points
3. If a point's distance is less than threshold value: add it into inliers
4. Compute F and the distance of the current inliers
5. If current error less than best error: record current inliers and F as best inliers and best F
6. After a given iterations, we obtain a set of inliers and F

Note that we use Sampson distance as error. Sampson distance can be roughly thought as the squared distance between a point x to the corresponding epipolar line x'F.

$$d(u_i, v_i) = (u_i F v_i)^2[1/((Fu^T)^2 + (Fu^T)^2) + 1/((vF)^2 + (vF)^2)]$$

## 2.3 Draw the Interest Points and the Corresponding Epipolar Lines

We can use fundamental matrix to obtain epipolar line as follows, $l = F^T x'$ and $l' = Fx$. The dimension of l and l' are 3xN, which reprents N $ax + by + c = 0$ lines. Then we draw the keypoints and epipolar lines back to original image, as shown in Figure 2. This part of code was implemented in **draw_epilines**.

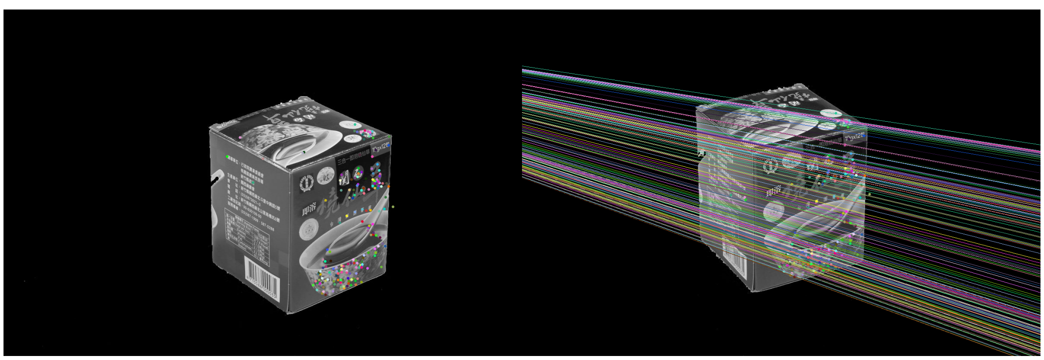

Figure 2. The epipolar lines and interest points of Mesona images.

## 2.4 Get 4 Possible Solutions of Essential Matrix from Fundamental Matrix

Given the intrinsic matrices of two cameras, we can simply compute essential matrix E through,

$$E = U diag(1, 1, 0)V^T$$

The decomposition of essential matrix was implemented in **find_second_camera_mat**. Next, let the first camera matrix be $P_1 = [I \mid 0]$, there will be four possible solutions for the second camera matrix $P_2$, says,

$$P_2 = \left[UWV^T \mid +u_3\right]$$
$$P_2 = \left[UWV^T \mid -u_3\right]$$
$$P_2 = \left[UW^TV^T \mid +u_3\right]$$
$$P_2 = \left[UW^TV^T \mid -u_3\right]$$

where W is,

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

After having four possible camera matrix, we use triangulation to convert image points back to 3D coordinate. To obtain X, we can solve the system of equations via SVD.

$$AX = 0 \qquad A = \begin{bmatrix} up_3^T - p_1^T \\ vp_3^T - p_2^T \\ u^{'} p_3^{'T} - p_1^{'T} \\ v^{'} p_3^{'T} - p_2^{'T} \end{bmatrix}$$

where,

$$x = w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad x^{'} = w \begin{bmatrix} u^{'} \\ v \\ 1 \end{bmatrix} \qquad P = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} \qquad P^{'} = \begin{bmatrix} p_1^{'T} \\ p_2^{'T} \\ p_3^{'T} \end{bmatrix}$$

$$x = PX \qquad x^{'} = P^{'} X$$

We developed the computation above in **triangulation** function.

### 2.5 Find out the Most Appropriate Camera Matrix and Triangulation

Follow the instruction in slides, we can compute camera center C with rotation matrix R and translation t,

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix} = R^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - R^T t = -R^T t$$

For each point in 3D coordinates, we check the angle between camera center to point and the camera's view direction. The camera matrix with the most points in front of itself becomes our candidate to the final stage. We record the 3D and 2D points, use the given matlab function to obtain .obj files (Fig 5 and 8).

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

# 3. RESULT

## 3.1 Mesona images

Figure 3 shows our result if Mesona images and its corresponding epipolar lines, we can extract more feature points.

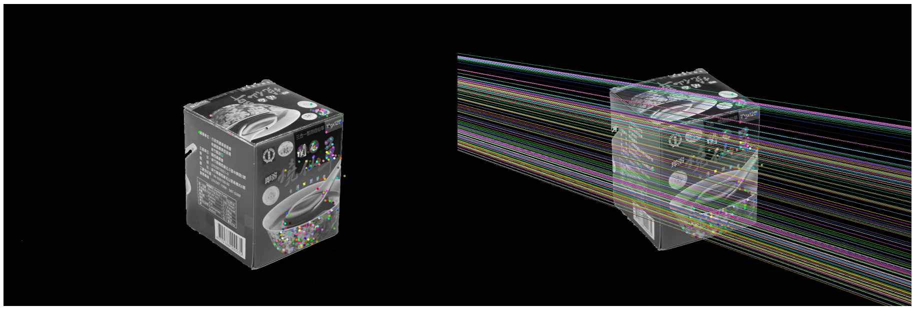

Figure 3. The epipolar lines and interest points of Mesona images

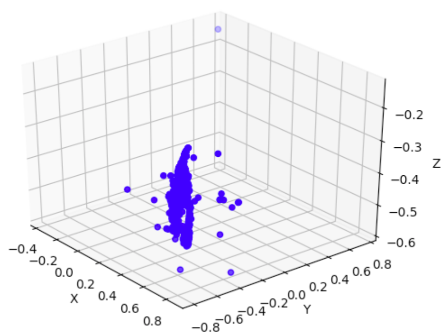


Figure 4. The 3D points from our implementation.

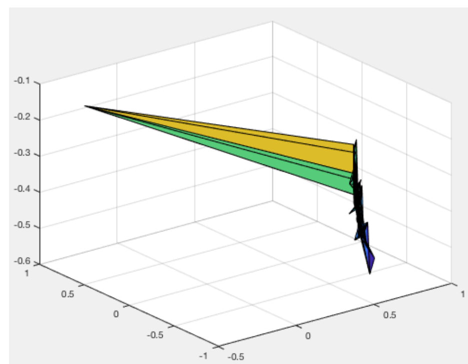

Figure 5. The reconstructed 3D model.

## 3.2 Statue images

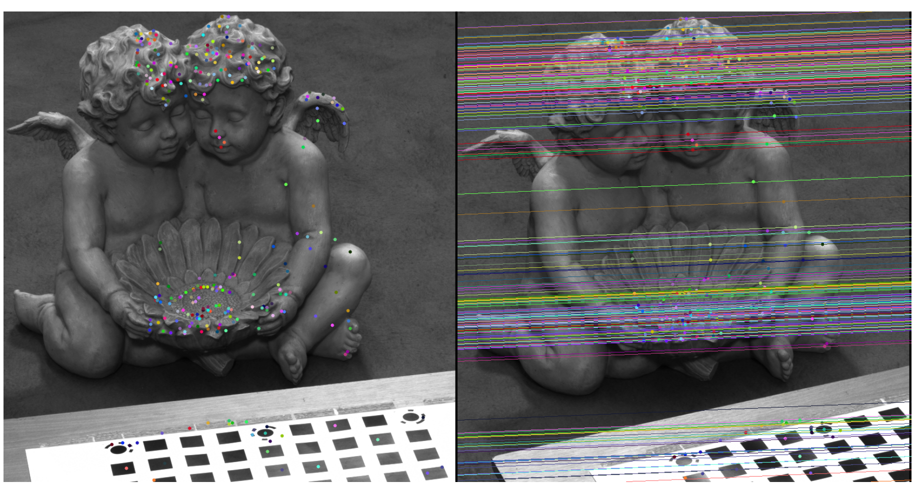

Figure 6. The epipolar lines and interest points of Statue images.

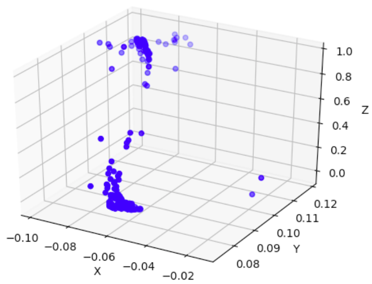


Figure 7. The 3D points from our implementation.

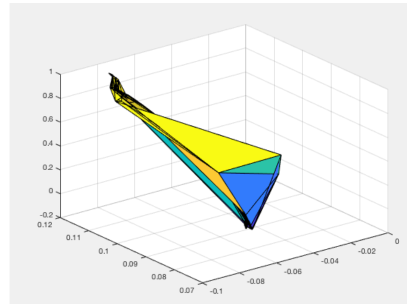

Figure 8. The reconstructed 3D model.

## 4. DISCUSSION

We tried to implement each part of the reconstruction pipeline on our own (provided in our source code). There are still some issue in computing fundamental matrix, so we chose using **cv2.findFundamentalMat** to get the correct result. Figure 9 is the epipolar line computed by our implementation of fundamental matrix. Indeed we made mistake in it and we are expecting to fix it before demo. We also consider to extract more feature points to have a better reconstructed 3D models.

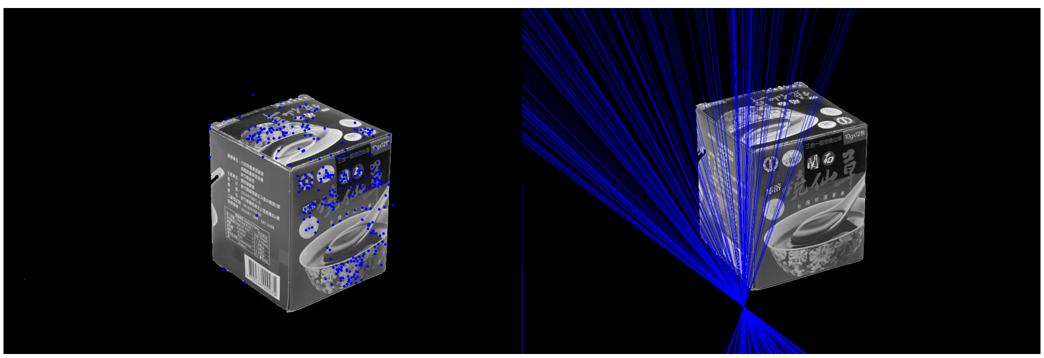

Figure 9. A false trial.

## 5. WORK ASSIGNMENT PLAN

- Li-Yang Wang: essential matrix and find camera matrix
- Yi-Chen Lee: triangulation, reconstruction 3d model with matlab
- Wen-Jie Tseng: feature extract, RANSAC and eight-point algorithm, fundamental matrix

## 6. CODE USAGE

- task_1.py: reconstruct 3D points of Mesona images.

  ```
  python3 task_1.py
  ```

- task_2.py: reconstruct 3D points of Statue images.

  ```
  python3 task_2.py
  ```

- opencv-funcs.py was for confirming our implementation is correct. Outputs of our scripts were stored ing out_3dpts, out_imgs, and out_obj directories.

## REFERENCES


1. Multiple View Geometry. Richard Hartley and Andrew Zisserman (2004).
2. Computer Vision: Algorithm and Applications. Richard Szeliski (2010).
3. OpenCV guide on epipolar geometry.
4. A GitHub reference code.