Computer Vision

Group 12 | 0786031 廖俊凱 0516044 陳思妤 0856733 黃明翰

👉Animated 3D model results are in /results/3D_model/

Introduction

Structure from Motion (SfM) is a technique which utilizes a series of 2-dimensional images to reconstruct the 3-dimensional structure of a scene or object. SfM can produce point cloud based 3-D models similar to LiDAR. This technique can be used to create high resolution digital surface models (including digital elevation models) and models of objects with consumer grade digital cameras. The relatively new technique has been made possible by advances in computers and digital cameras. Together, these advances have now made it feasible for a wide range of users to be able to generate 3-D models without extensive expertise or expensive equipment.

SfM is based on the same principles as stereoscopic photogrammetry. In stereophotogrammetry, triangulation is used to calculate the relative 3-D positions (x, y, z) of objects from stereo pairs. Traditionally these techniques require expensive specialized equipment and software.

To create a 3-d reconstruction one simply needs many images of an area or an object with a high degree overlap, taken from different angles. The camera doesn't need to be specialized, standard consumer-grade cameras work well for SfM methods. The images are often be taken from a moving sensor, but can also be taken by a person or multiple people are different locations and angles.

In this assignment, we take two examples, (1) extraction of features in images (e.g., points of interest, lines, etc.) and matching these features between images and (2) recovery of the 3D structure using the estimated motion and features.

Implementation Procedure

1. Find out correspondence across images. **function**: **sift**()、**find_match**()

   This part is as same as the feature matching part in HW3. We first detect interest points by SIFT, then exact and match SIFT features between all of the images

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

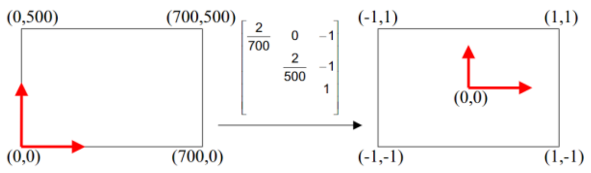
   function: **normalize**()、**fundamental_matrix**()

   We estimate the fundamental matrix based on normalized 8-points algorithm provided by teacher's slide.

   A. Normalize the input

   Make the mean = (0,0,0), std = (1,1,1),

   $\tilde{x} = Tx,\ \tilde{x}' = T'x'$


(0,500)
(700,500)
(0,0)
(700,0)
$\begin{bmatrix} \frac{2}{700} & 0 & -1 \\ & \frac{2}{500} & -1 \\ & & 1 \end{bmatrix}$
(-1,1)
(1,1)
(0,0)
(-1,-1)
(1,-1)

B. Construct the 8-points algorithm function and use RANSAC algorithm to find better F.

I. Find F with random 8 matching points by normalized 8-points algorithm. Initial solution via SVD and enforce $\det(\tilde{F}) = 0$ by SVD.

$$\begin{aligned} n &= x_1.shape[1] \\ A &= [x_2[0]x_1[0], x_2[0]x_1[1], x_2[0], x_2[1]x_1[0], x_2[1]x_1[1], x_2[1], x_1[0], x_1[1], ones((n,1))] \\ U, S, V &= svd(A) \\ F &= V[-1,:].reshape(3,3) \\ U, S, V &= svd(F) \\ F &= Udiag(S[0], S[1], 0)V \end{aligned}$$

II. Find inliers with this $\tilde{F}$ with error < threshold

III. Repeat the step A and step B

IV. Return the best $\tilde{F}$ and inliners

C. Denormalize

$$\mathrm{F} = T'^{T}\tilde{F}T$$

3. Draw the interest points on our found in step.1 in one image and the corresponding epipolar lines in another. **function**: **compute_epilines**()、**draw**()

We use the interesting points(x') in image2 and calculate the epipolar lines in image1 using the equation

Epipolar_line:

$$l' = Fx \qquad l = F^T x'$$

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

function: **essential_matrix**()、**get_4_possible_projection_matrix**()

Since we have fundamental matrix, we can use the given intrinsic matrix K1、K2 to get the essential matrix by the formula:

essential matrix: $\mathrm{E} = K_1^T F K_2$

second camera extrinsic matrix:

$$E = Udiag(1,1,0)V^T \qquad W = \begin{vmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$\begin{aligned} P_2 &= [UWV^T \mid +u_3] \\ P_2 &= [UWV^T \mid -u_3] \\ P_2 &= [UW^TV^T \mid +u_3] \\ P_2 &= [UW^TV^T \mid -u_3] \end{aligned}$$

5. Find out the most appropriate solution of essential matrix.

   function: **get_correct_P**()

   We prefer more 3D points in front of cameras. Thus, we try to use each solution of extrinsic matrix P2 we get in step 4 to get 3D points by triangulation, and then pick the solution of extrinsic matrix P2 with most of 3D points in front of cameras.

   We will discuss the detail of triangulation in step 6. In this step, we focus on how to determine the better extrinsic matrix P2 from 4 candidates.

   Assume we get 3D points X by triangulation from step 6.

   Knowing that extrinsic matrix P equals to [R|T]

   Given X, P1, P2:

$$
\begin{aligned}
R =&\ P_1[:,0:3] \\
t =&\ P_1[:,3] \\
C =&\ -R^T t \\
X\ is\ in\ front\ of\ P1? =&\ ((X - C)\ *\ R[2,:]^T\ >\ 0) \\
R =&\ P_2[:,0:3] \\
t =&\ P_2[:,3] \\
C =&\ -R^T t \\
X\ is\ in\ front\ of\ P2? =&\ ((X - C)\ *\ R[2,:]^T\ >\ 0)
\end{aligned}
$$

6. Apply triangulation to get 3D points. **function**: **triangulation**()

   Using triangulation provided by teacher's slide can reconstruct the 3D points from 2D point and camera matrix.

   Given x1, x2, P1, P2:

$$
x_1 = w\begin{vmatrix} u_1 \\ v_1 \\ 1 \end{vmatrix}, x_2 = w\begin{vmatrix} u_2 \\ v_2 \\ 1 \end{vmatrix}, P_1 = \begin{vmatrix} {p^1}^T_1 \\ {p^1}^T_2 \\ {p^1}^T_3 \end{vmatrix}, P_2 = \begin{vmatrix} {p^2}^T_1 \\ {p^2}^T_2 \\ {p^2}^T_3 \end{vmatrix}, A = \begin{vmatrix} u_1 {p^1}^T_3 - {p^1}^T_1 \\ v_1 {p^1}^T_3 - {p^1}^T_2 \\ u_2 {p^2}^T_3 - {p^2}^T_1 \\ v_2 {p^2}^T_3 - {p^2}^T_2 \end{vmatrix}
$$

$$
\begin{aligned}
&U,S,V = svd(A) \\
&\mathrm{X} = \mathrm{V}[:,-1]
\end{aligned}
$$

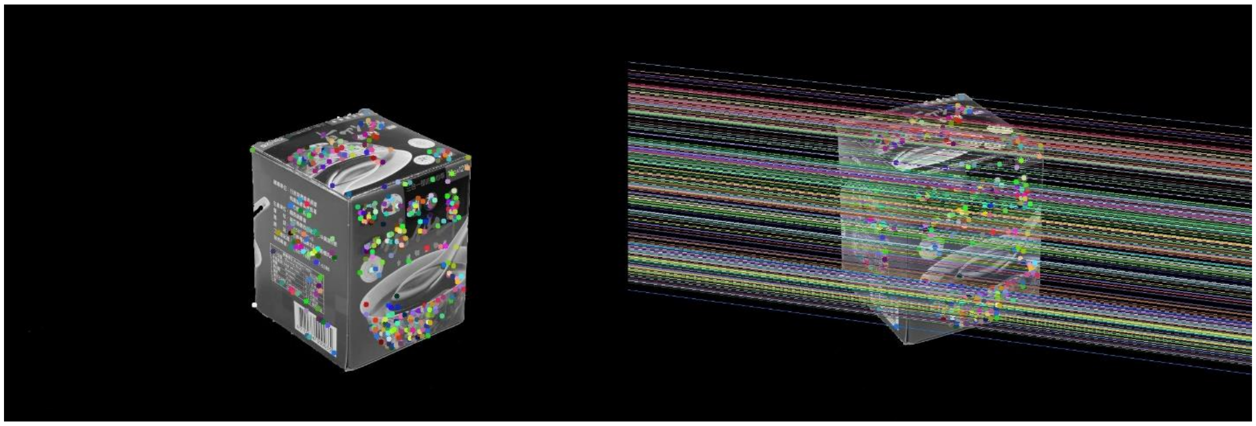
7. Use texture mapping to get a 3D model.

   We use the MATLAB code provided by the Professor Chiu. By given the input image、2D points、3D points and camera matrix, the function will output an .obj file and a .mtl file. And we can use some kind of software to visualize our results.

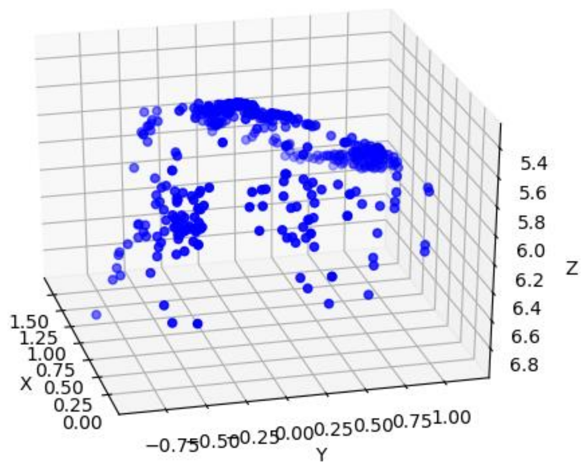
Experiment Result

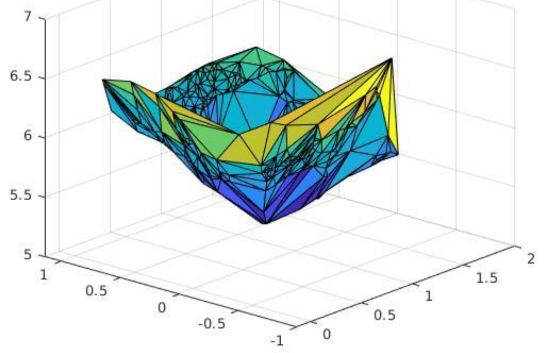
1. Mesona

Epipolar lines

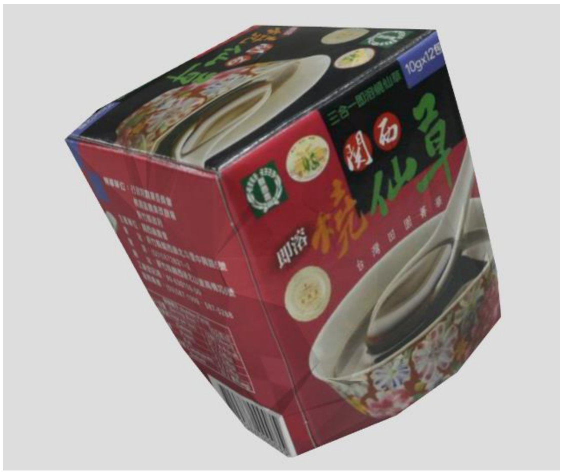


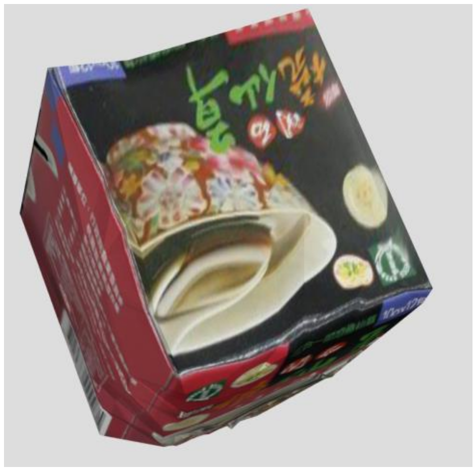
3D points





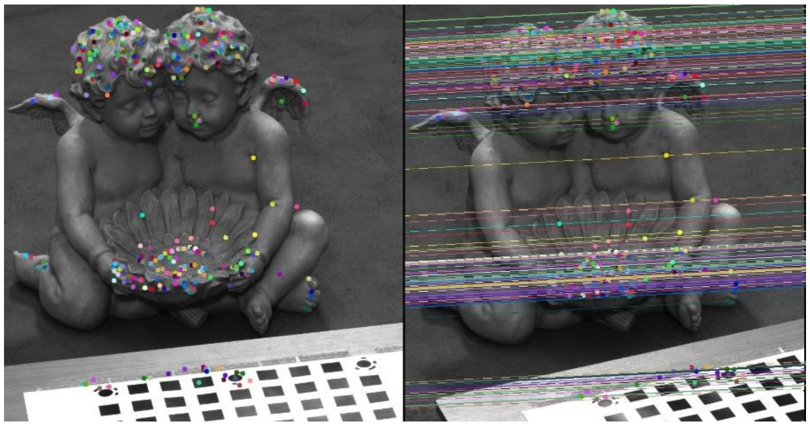
3D model



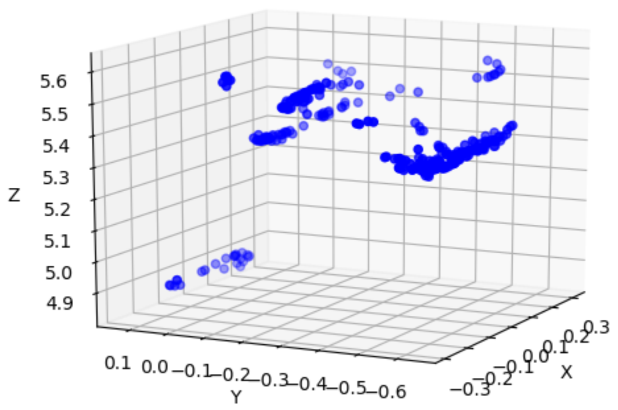


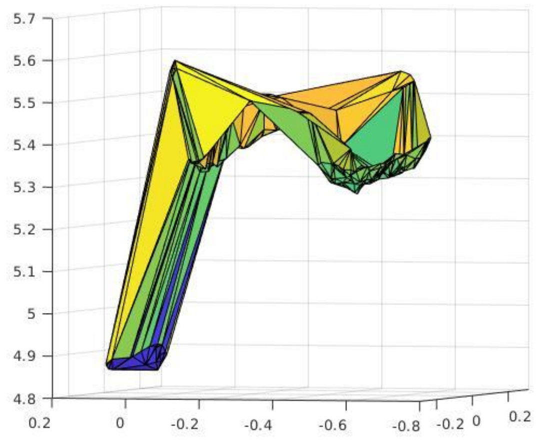
2. Statue

Epipolar lines

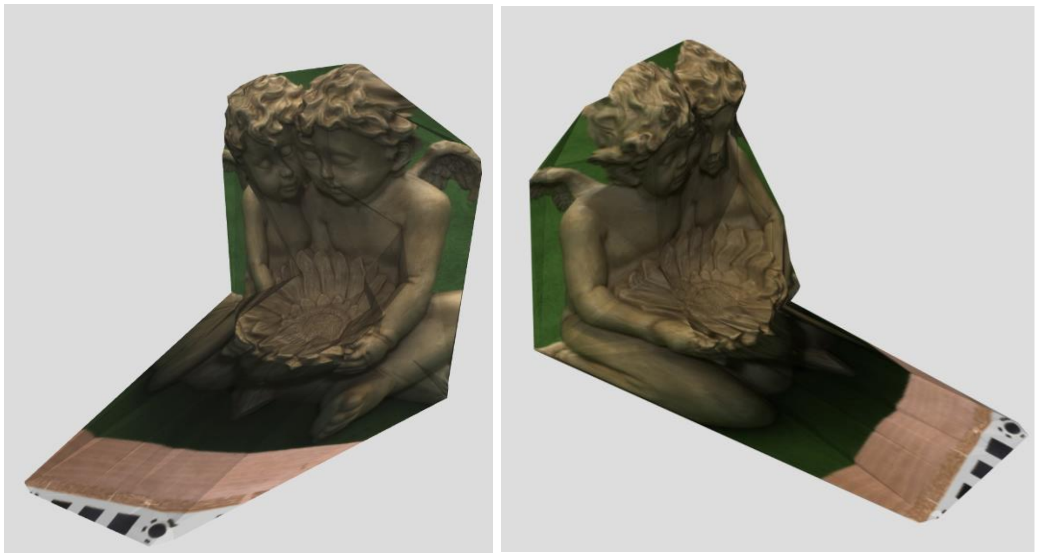


3D points



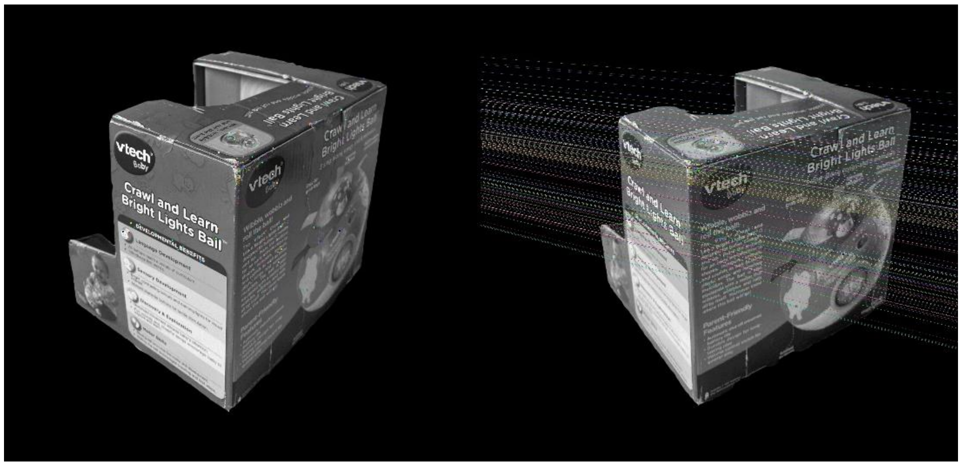


3D model

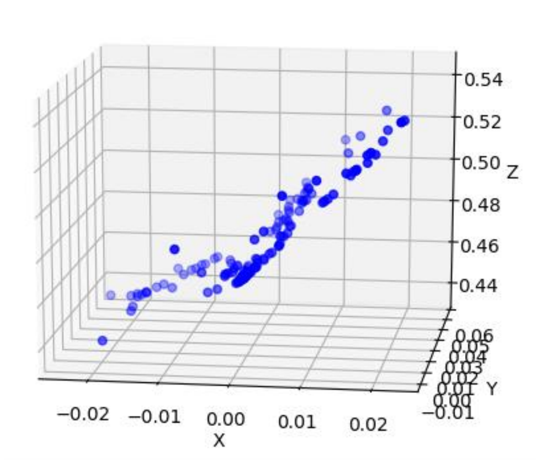


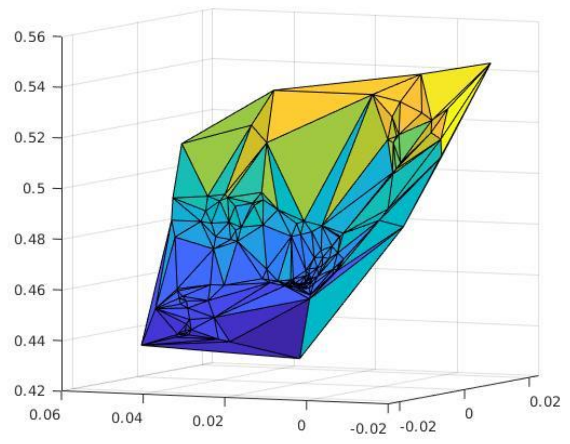
3. Our Data (Box)

Epipolar lines



3D points

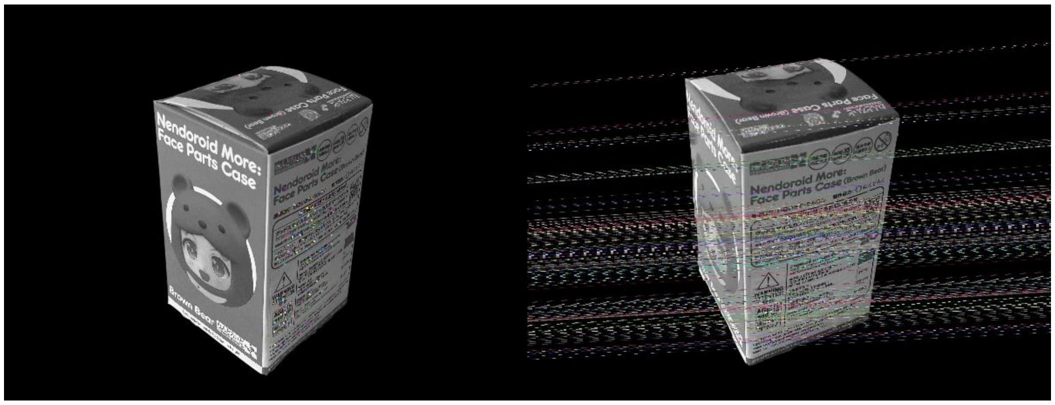




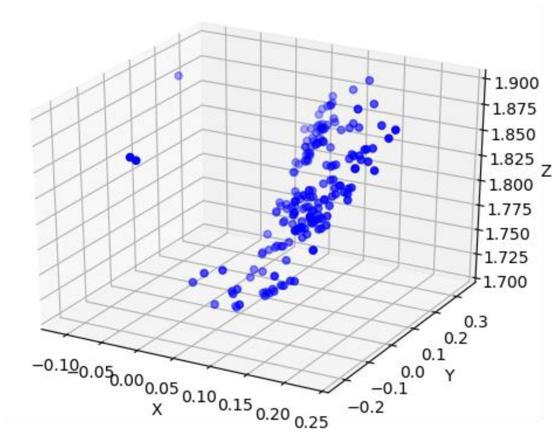
3D model

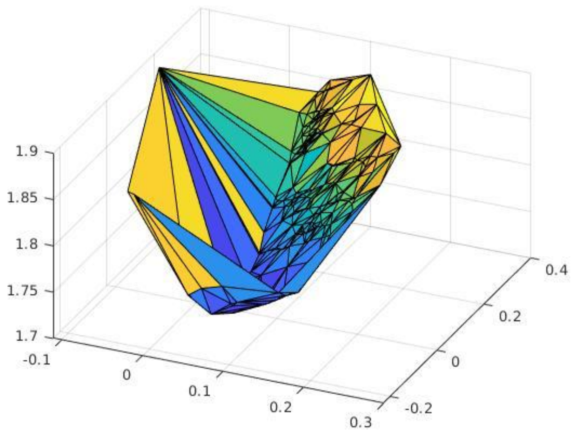
4. Our Data (PVC Box)

Epipolar lines

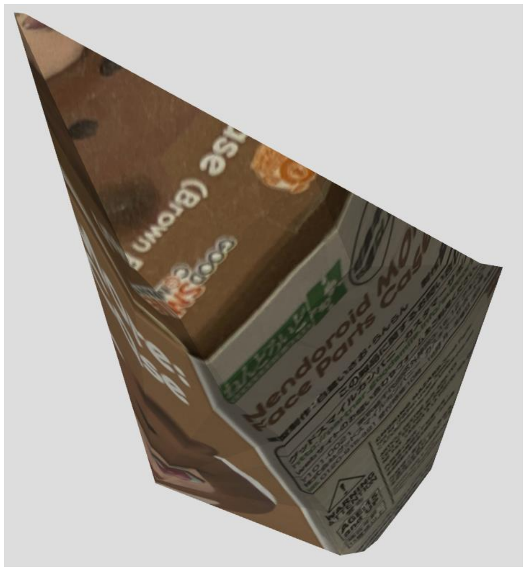


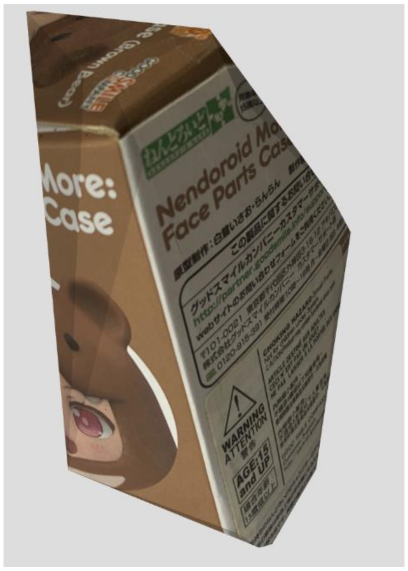
3D points





3D model





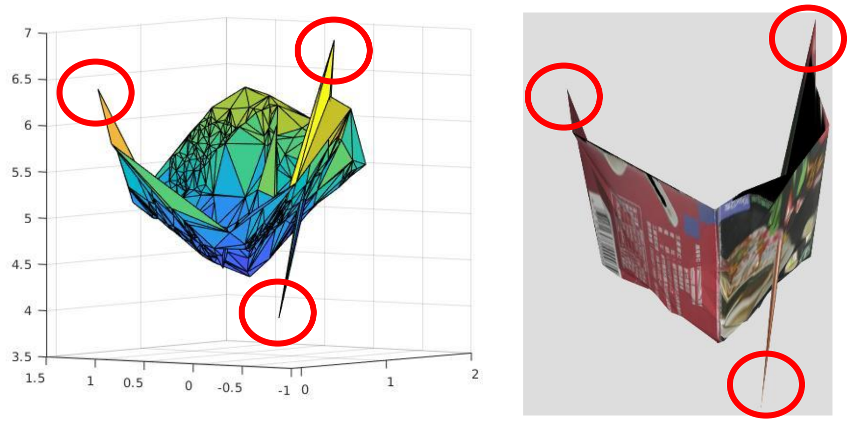
Discussion

1. Fundamental Matrix

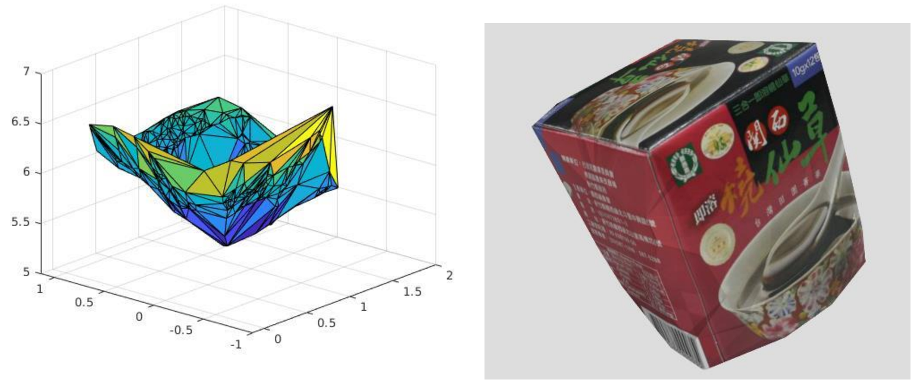
RANSAC

Fundamental matrix estimation is sensitive to quality of matches, there are many algorithms to get the fundamental matrix, here we use the 8-point algorithm + RANSAC to deal with outliers.

We found that it’s very important to set the proper threshold in RANSAC for each dataset. Following is the different outcome before and after we adjust the threshold.



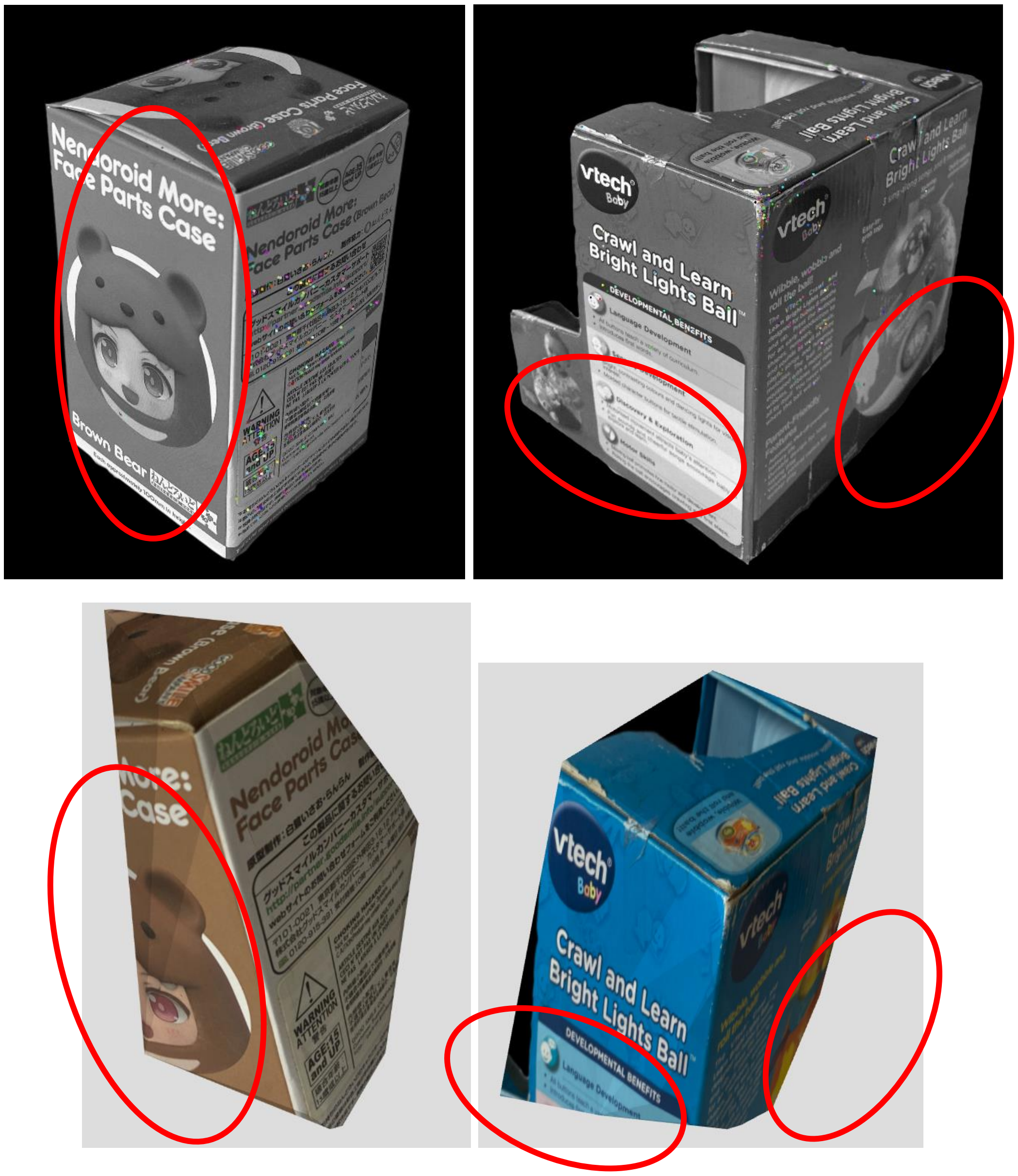
threshold = 0.05, there are still few outliers



threshold = 0.005, better result

In-liners & Out-liners

We found that in our data, the number of the interest points in each side are extremely imbalanced. This causes that many interest points in one side are remained and those in other sides are filtered as out-liners, which results the incompleteness of the results of the 3D model.

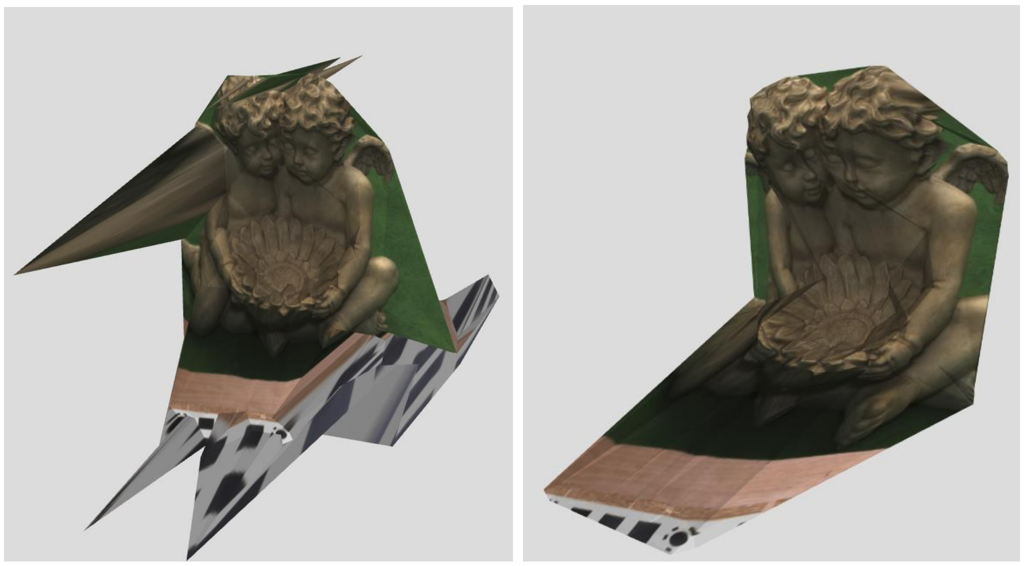


Efficiency

We found that when running our data, it takes longer time to complete the task than using the provided data. It is because the time complexity of RANSAC depends on the number of the correspondence points. Our images has more pixels thus more correspondence points.

2. Camera Matrix

The intrinsic matrix from cv2.calibratecamera and the K1, K2 provided by spec will cause the result very weird, it should be multiplied 0.001.


(left) original intrinsic matrix (right) intrinsic matrix *= 0.001, better result

Conclusion

Our result seems good enough, but there is still something that can be improved. We try our function of finding fundamental matrix with OpenCV's function and our final result is not as perfect as the result using OpenCV. We think the difference is caused by the fact that the parameters we choose in normalized 8-point algorithm are not perfect.

For the future work, we want to

1. Keep trying to finetune the parameters to get a better result
2. Use more input images to construct a more complete 3D structure

Work Assignment Plan Between Team Members

0516044 陳思妤	Data collection + Step 7
0786031 廖俊凱	Step 1~3 + Report
0856733 黃明翰	Step 4~6 + Report