**NCTU 2019 CV HW4:**

**Structure from Motion (SfM)**

**Hsin-Yu Chen, Yuan-Syun Ye**

**May 2019**

**1 Introduction**

In HW4, we need to compute the camera parameters and the 3D points coordinates by corresponding points in two or more images. First, we calculate feature points by SIFT implemented in HW3. Next, we use RANSAC to ﬁnd the fundamental matrix of the least outliner by 8-point Algorithm, then we can draw the epipolar lines. Finally, we compute four possible camera matrix then pick the right one to project 2D points back to 3D by Linear Triangulation and Inner product.

**2 Implementation**

**2.1 Feature matching by SIFT features**

Like what we have done in HW3. We use SIFT to find the feature points, and use the brute-force matcher method and ratio test to ﬁnd the corresponding points of the two images. The code is Figure 1. The result is Figure 2.

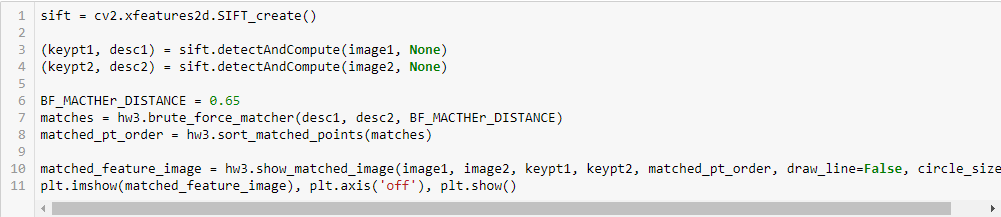


Figure using SIFT feature to match feature

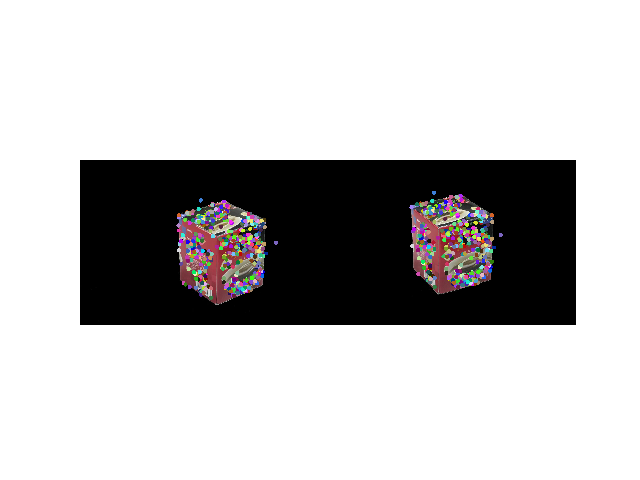


Figure SIFT Reuslt

**2.2 Estimate Fundamental matrix and Essential matrix by RANSAC**

**2.2.1 Normalize image coordinates**

Before computing Fundamental matrix, we need to Normalize image coordinates first in order to improve the stability of the calculation. The code is in Fig 3

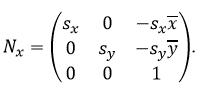
Let points in both images’ mean = 0 a standard deviation ~= (1,1,1)

X and X’ are pair of corresponding points in two images

And we need to compute Nx andNx’

x = Nx x and x’ = Nx’x’

We set normalized matrix 𝑁𝑥 as:



Where x, y is the average of the original points,σis variances of the original points



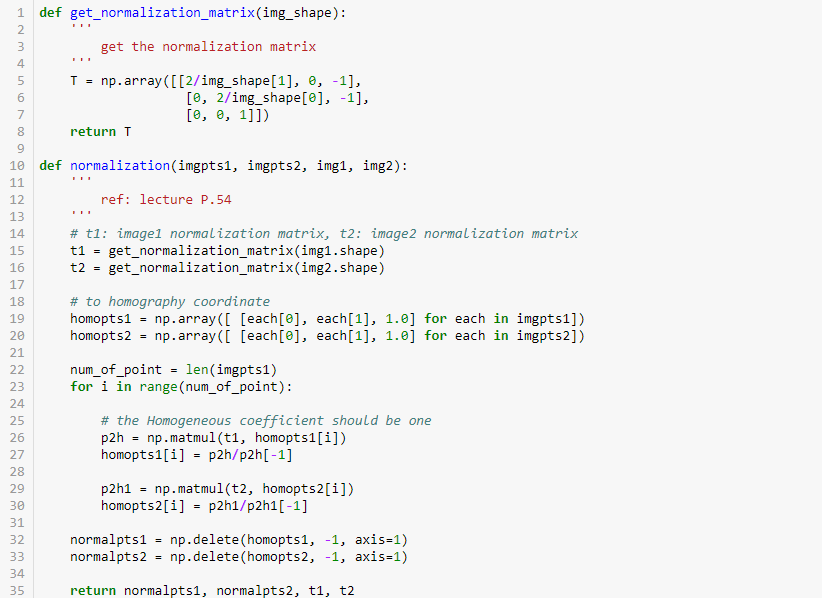


Figure normalization process

**2.2.2 Compute Fundamental matrix by 8 points Algorithm and De-normalize it**

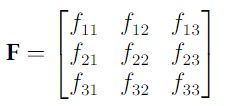
X and X’ are pair of corresponding points in two images.

The fundamental matrix is defined by the equation:



That is, while

X = (x, y, 1) and X’ = (x, y, 1)



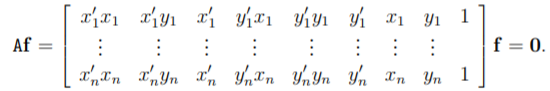
The equation corresponding to a pair of points (x, y, 1) and (x, y, 1) is:



Then change equation to vector inner product form:



It can also be written as the follow form:



Then, we compute Fundamental matrix by **8-point Algorithm**:

Linear solution: solve f from Af = 0 using SVD

Constraint enforcement: Resolve det(F)=0 constraint using SVD

**De-normalize** by normalized matrix 𝑁𝑥

F = Nx’TFNx

The code is in Figure 4



Figure de-normalize fundamental matrix and compute fundamental matrix process

**2.2.3 Compute Essential matrix**

The defining equation for the essential matrix is:



Where



So it can be written as:



And



That is, relationship between the fundamental and essential matrices is:



And we have already Estimate Fundamental matrix by RANSAC so we can compute essential matrices. The code is in Figure 5.

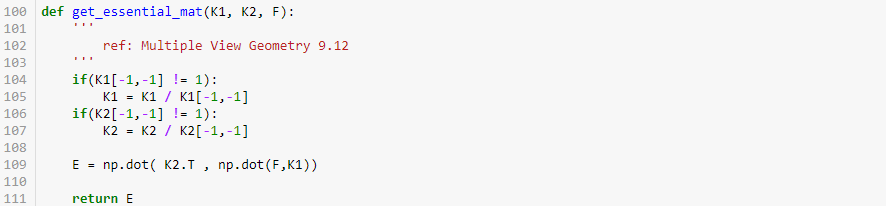
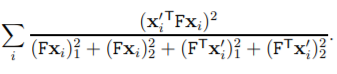


Figure essential matrix process

**2.2.4 RANSAC**

RANSAC is a way of randomly sampling feature points and uses the sampled feature points to calculate fundamental matrix, while the more inliner number of fundamental matrix is the best solution. Here we choose 8 for the number of random samples. The Sampson distance is used to compute geometric error for each fundamental matrix, which is defined as:



We determine the inliner threshold to 0.25, and to repeat 5000 times. The code is in Figure 6 and Figure 7.

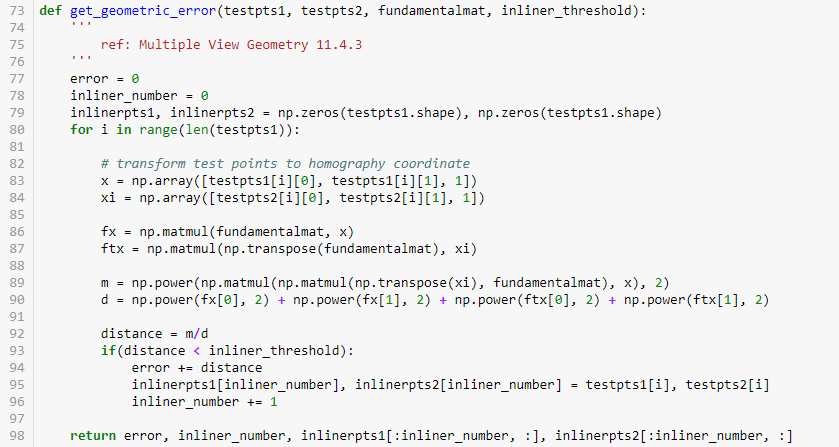


Figure compute inliner threshold by Sampson distance

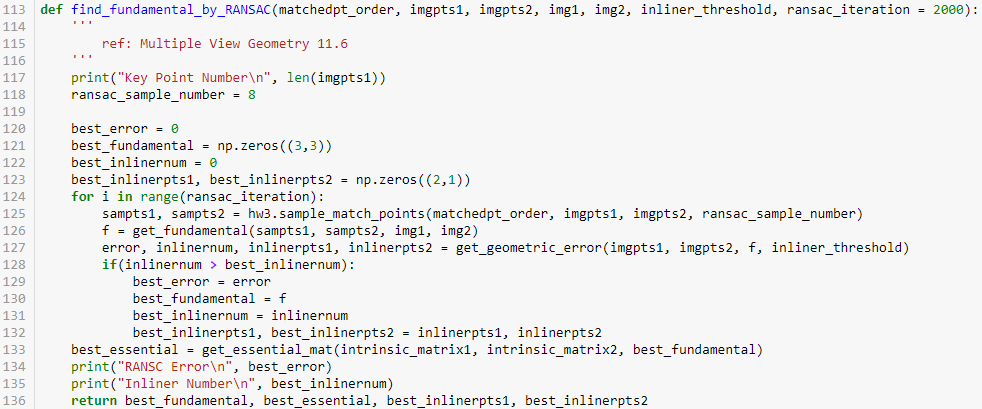
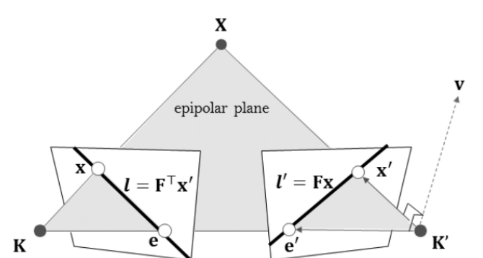


Figure using RANSAC to find the most inliner number of fundamental matrix

**2.3 Draw epipolar lines**



The function of epipolar lines is defined as:

and 

Now, we have the equation of epipolar lines of both image, then we substituting two points into equations and draw the line on the image. The code is in Figure 8.

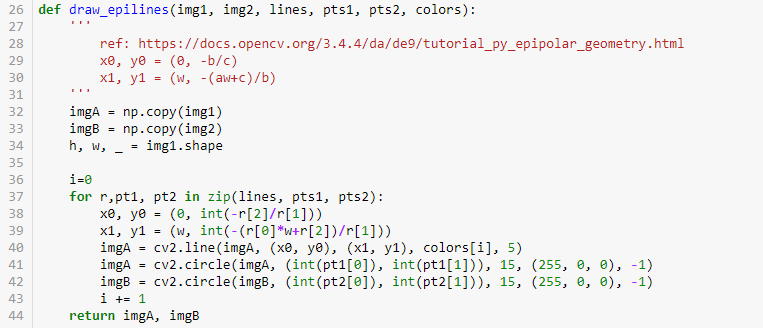


Figure draw epipolar lines process

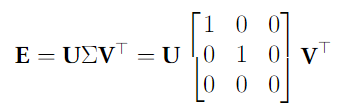
(epipolar lines result)

**2.4 Estimate four possible camera matrix**

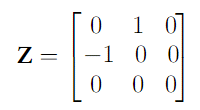
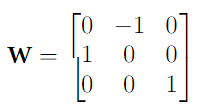
Essential matrix E can presented by ***R*** and ***t***:



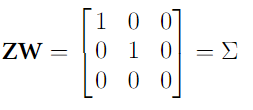
Due to the lack of rank caused by the outer product operation and the orthogonality of the rotation matrix, the singular value decomposition of the essential matrix ***E*** has a special structure, which is:



First, we defined:

Where



Then we can write Essential matrix ***E*** as:



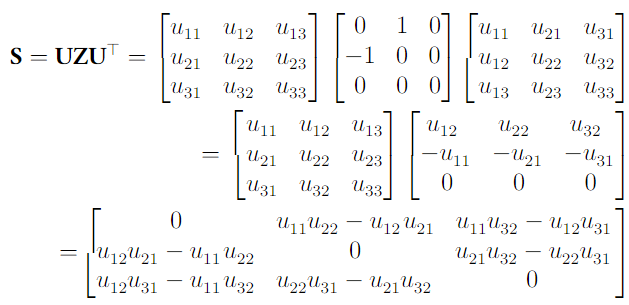
***U*** is orthogonal matrix (U⊤U=UU⊤=I), so we get the equation:



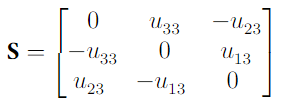
And define ***S*** and ***R’*** as:

and 

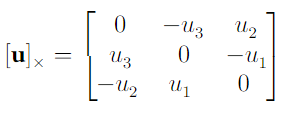
Now, we can write equation as: ***E = SR’***, which is similar with ***E = [t]xR***, we want to check whether they are same:



***U*** is orthogonality, where u∗3=u∗1×u∗2 (u∗i is ith column of U). S can be written as:



The outer product of the vector can be described by a 3 × 3 matrix, defined as:



And ***R'*** is an orthogonal matrix because:



So

***[t]x*** ***= S*** and ***R = R’***

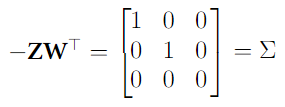
According to the relationship between the outer product operation and the antisymmetric matrix, we can get ***t***:

***t= (s32, s13, s21) T= (u31, u32, u33) T***

Because ***–E = − [t]x R= [-t]x R*** is also an essential matrix equivalent to ***E***, there are two possible solutions of ***t***, where:

***t1 = (s32, s13, s21) T= (u31, u32, u33) T*** and ***t2 = −t1***

Also,***Σ***can be write as another way:



So Essential matrix ***E*** can also be defined as:

***E = −UZUT UWTVT***

And get another set of solutions:

***S = −UZUT and R’ = UWTVT***

That is, we get two possible solutions of ***R***:

***R1=UWVT and R2=UWTVT***

So the four sets of possible solutions ***(R1, t1), (R1, t2), (R2, t1) and (R2, t2)*** are generated.

That is, for a given essential matrix ***E = UΣVT***, and first camera matrix ***P = [I | 0]***, there are four possible choices for the second camera matrix P , namely:

***P = [UWVT | +u3] or [UWVT | −u3] or [UWTVT | +u3] or [UWTVT | −u3].***

The code is in Fig 8.

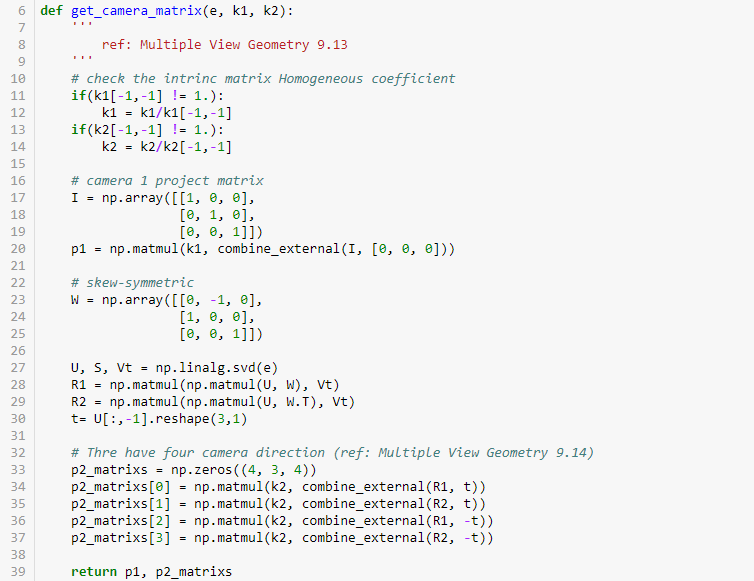
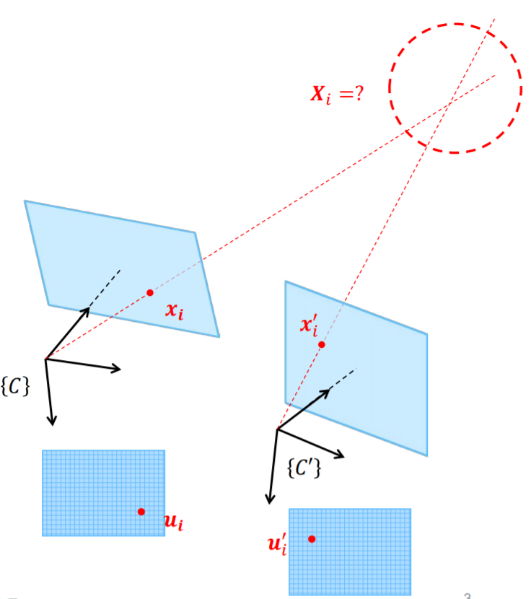


Figure estimate four camera matrixs

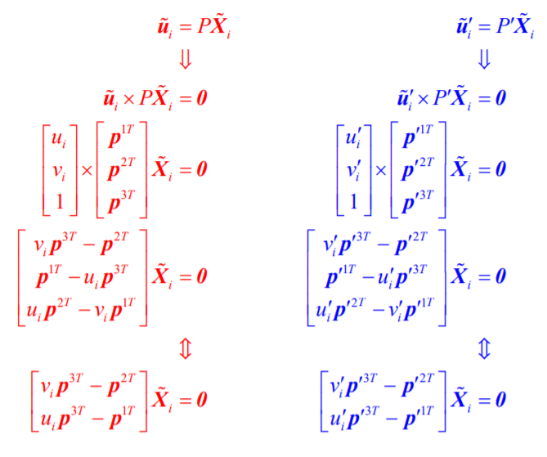
**2.5 Project 2D points back to 3D**

**2.5.1 Triangulation**

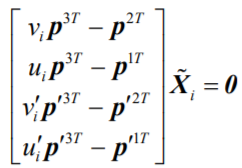
Now we have camera matrices 𝑃, 𝑃’ and 2D correspondences 𝒖𝑖 ↔ 𝒖𝑖′, and we want to reconstructed 3D point 𝑿i.



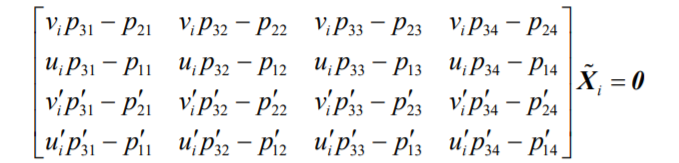
In the method of Linear Triangulation, we get two equations from each perspective camera model. And we need to solve this system of equations with SVD to reconstructed 3D point Xi.



Combining these equations:



That is,



Then we defined the matrix as A and solve this system of equations by SVD.



The code is in Fig 9.

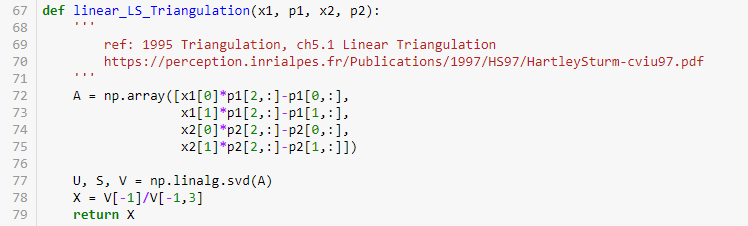


Figure linear triangulation process

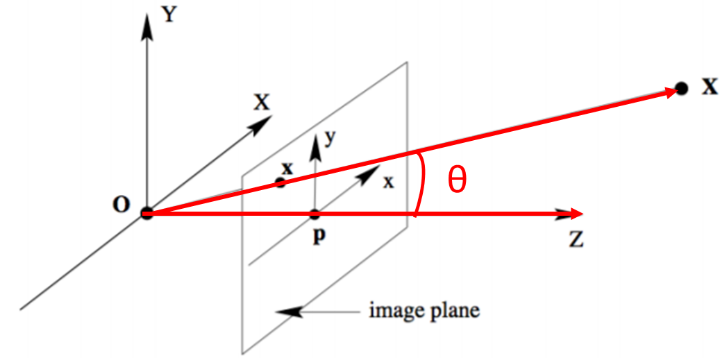
**2.5.2 Pick the solution with most of 3D points in front of cameras**

We want to pick the solution with most of 3D points in front of cameras. The method we check whether a 3D points ***X*** is in front of cameras is calculate Inner product of View Direction and vector from Camera Center to 3D points ***X***.

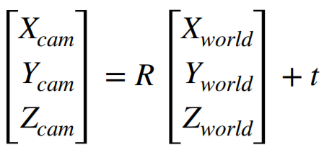
Because Inner product of two vector ***U*** and ***V*** is:



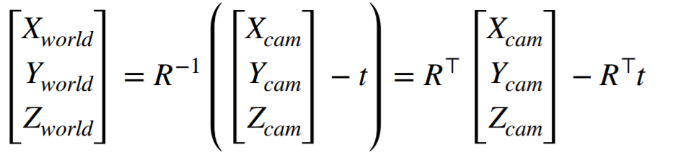
And cosθ is positive whileθis between 90 to -90 degrees. That is, if the value of Inner product is positive then this point X is in front of the cameras.



First, we have ***R*** and ***t***, which can turn points in world coordinate system into camera coordinate system by the function:

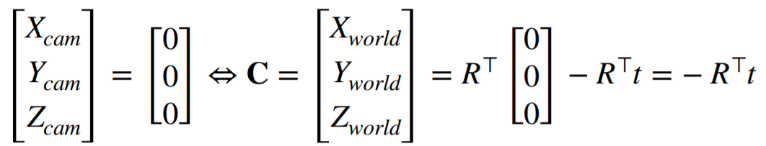


And we need to turn points in camera coordinate system into world coordinate system, so change the function into:

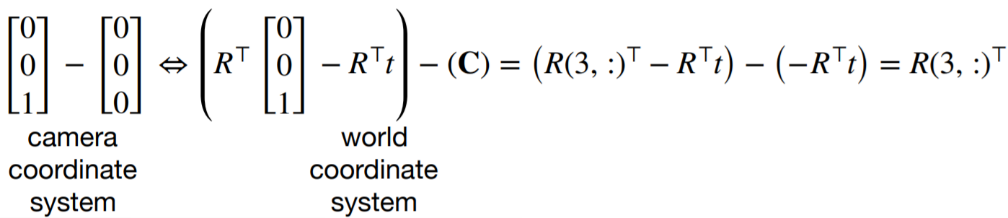


Then we can compute Camera Center and View Direction in world coordinate system.

Camera Center ***C***:



View Direction



Next, calculate the Inner product and check whether it is positive.



We have to calculate number of points in front of cameras for each solution ***(R1, t1), (R1, t2), (R2, t1) and (R2, t2)*** and pick the solution with most points.

The code is in Fig 10.

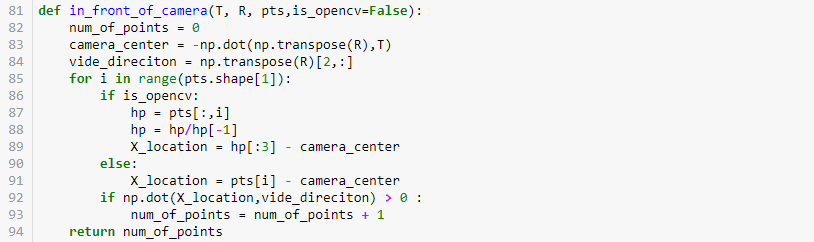


Figure check project points is in front camera

**3 Experimental**

3.1 TA Images

The TA provides two test materials, namely, Mesona, and Statue1, which correspond to Fig 11, Fig 12 , respectively.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

3.2 Our Images

3.3 Compare with Open CV

To check the results of our implementation, we compare the result with Open CV in Fig 14 and also show the Fundamental matrix estimated by each way. It can be found that our Fundamental matrix and the matrix Open CV computed are diﬀerent, probably because our RANSAC did not completely select all the inliners. Comparing the calculated Fundamental matrix, ours is:

And Open CV is:

It can be seen that there is a diﬀerence between the two.

**4 Discussion**

There are two methods to estimate Fundamental matrix, 8-point algorithm and 7-point algorithm respectively, where 8-point algorithm is simplest method, just need to compute least squares solution of a set of linear equations using SVD, and that is what we did in the homework. Compare with 8-point algorithm, 7-point algorithm is faster (need fewer points) and could be more robust (fewer points), but there would have one or three real solutions, cause more case need to deal with, also the method is more complex.

Another thing to note is that intrinsic matrix K given for first image TA provide. The k33 is 0.001，we need to divide intrinsic matrix K by 0.001 to turn k33 to 1.

**5 Conclusion**

We implement Structure from Motion(SfM), used SIFT to ﬁnd feature points, and used the Brute-Force method to pair feature points, then use 8-point algorithm to estimate fundamental matrix, ﬁnd the best fundamental matrix by RANSAC and draw the epipolar lines. Then, we compute four possible camera matrix and do Linear Triangulation to reconstructed 3D respectively. Finally, we pick the solution with most of 3D points in front of cameras and project 2D points back to 3D.

**6 Work Assignment Plan**

This homework divided into two parts. Yuan-Syun Ye is responsible for the part of cording and checks this report. Hsin-Yu Chen is responsible for the writing of the report.

**References**

http://cvrs.whu.edu.cn/downloads/ebooks/Multiple%20View%20Geometry%20in%20Computer%20Vision%20(Second%20Edition).pdf

https://www.researchgate.net/publication/303522230\_qiantanjichujuzhenbenzhijuzhenyuxiangjiyidong\_Beginner%27s\_Guide\_to\_Fundamental\_Matrix\_Essential\_Matrix\_and\_Camera\_Motion\_Recovery?fbclid=IwAR3chaM6yPcNxyrD2vTfzR3tgcejHuzF-jZXXgi8\_B0AFNWeLa7tdWzbnHg