**Cs 512 Assignment 3: report**

**Abstract**

This program separate to 2 parts.

1st part is used to generate the position of corner in world plane and image plane.

2nd part is compute camera parameters with professor noticed in class and implement the RANSAC algorithm for robust estimation.

**Problem statement**

* 1. Using openCV function (e.g. cvFindChessboardCorners, cvFindCornerSubPix) cannot find the corner by the cube image.
  2. Opencv do not have function to detect position of curse. In order to support that function, trying to using Pymouse and others function but seems not continence.
  3. Lost last point record in saved detection information file.
  4. Calibration result is too small with coplanar calibration.
  5. ‘np.loadtxt’ cannot split the coordinate except add signal comma between x coordinate and y coordinate in saved information file.
  6. Cannot understand how to using RANSAC algorithm such as when to exit the while loop and how to calculate the best solution.
  7. The camera parameter S is non 0 but the known parameters which professor provided is 0 and that are different.
  8. With the noisy image, the parameter which calculated have mistaken.

1. **Proposed solution**
   1. CvFindChessboardCorners function seems cannot support 3d plane.
   2. Tk have the function to handle the cursor position and also can draw the input box.
   3. Add Seek() function after calling writelines function.
   4. Cannot found out the root cause.
   5. Using readline function to load points coordinates information.
   6. Thanks for xi zhang`s help to notice me how to understand the algorithm.
   7. The value which calculated is –xxx e-04 which is very small and rounded off to 0.
   8. In my opinion, the mistake is introduced by the noisy points are been calculated in the inlier list.
2. **Implementation details**

Program 1:

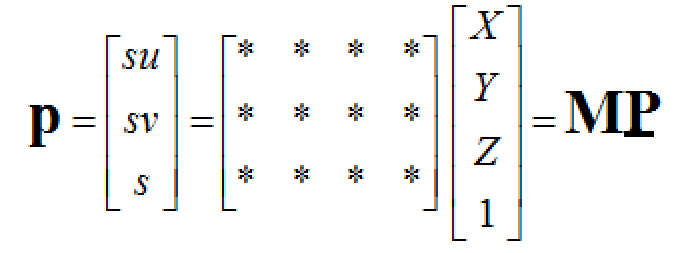
Mark the work plane coordination for the corner on the cube and provide the input box to key the corner coordinate value before click the corresponding point on image.

After that this program will save the world plane coordinates and image plane coordinates value to files.

Program 2:

Step 1; find out the production matrix M

As we know, the camera projection matrix is like below



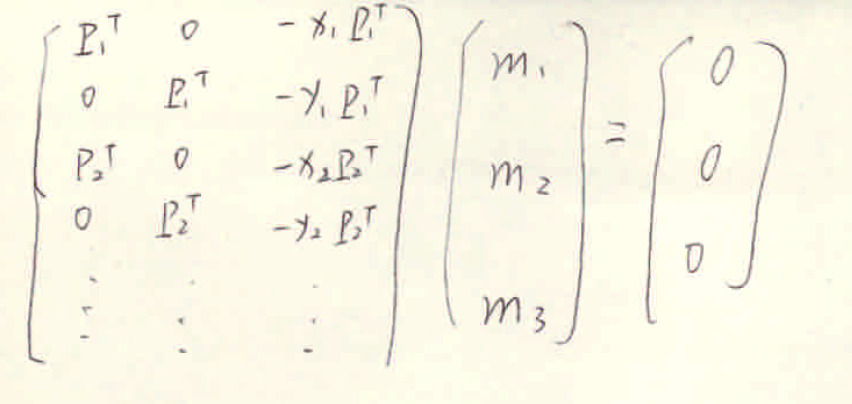
To expand the P and P we can get below equitation 

xi = xi`/wi` = m1tPi / m3tPi

yi = yi`/wi` = m2tPi / m3tPi

m1tPi - xi m3tPi =0

m1 tPi - xi m3tPi =0



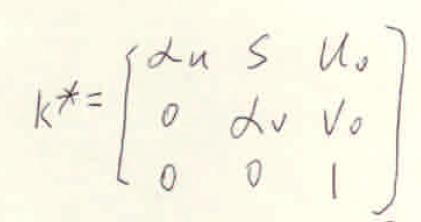
 step 2 find out M\*, T\*, R\*, U0, V0

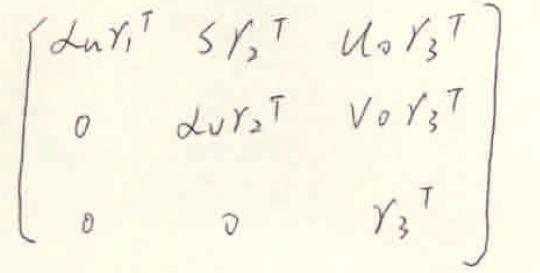
M = K\* [R\*| T\*] = [K\*R\*|K\*T\*]

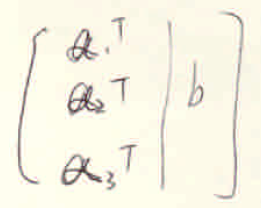
M` = δM

Meanwhile

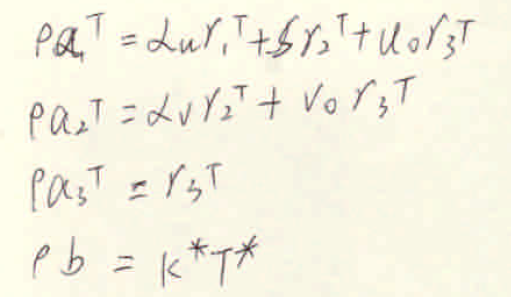
R\* = 

K\*= 

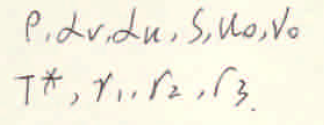
K\*R\* = 

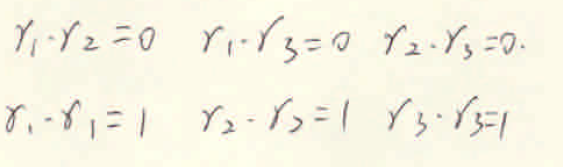
M` = 

So, we can get

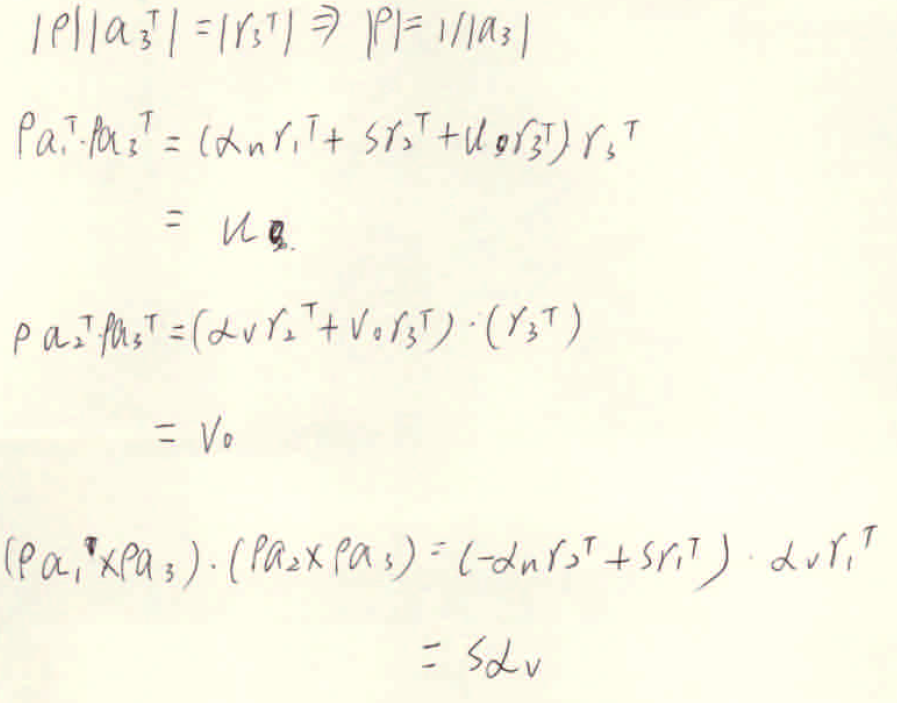


Know: a1,a2,a3,b

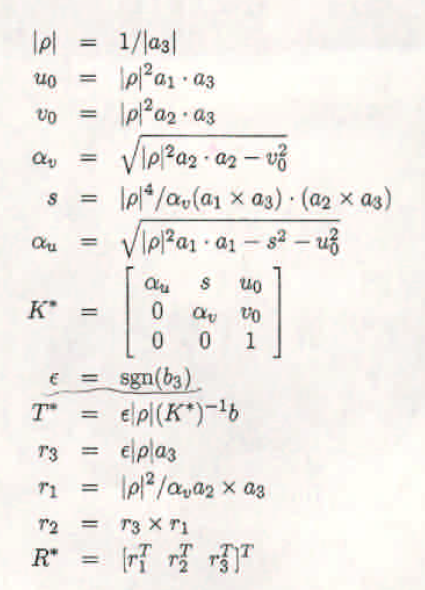
Unknow: 

As 

So



Then we can get below equation



Step 3: Ransac algorism

Below is the step for Ransac algorism.

1. A sample subset containing minimal data items is randomly selected from the points. A fitting model and the corresponding model parameter are computed using the small subset.
2. Algorithm check which elements of the entire points are inlier and using the inlier points to generate a fit model and calculate the corresponding model parameters using the inlier points.
3. Ransac algorithm will repeat the above two steps until k > maxk or find out the best solution.

Below is the code annotation:


k = \frac{\log(1 - p)}{\log(1 - w^n)}
 p = 0.99 w = 0.5 , n = 8 which get from configuration file.

While (i < k)

{

Calculate\_M(8 points)

InlierPoints = FitInlier(M1)

M2 = Calculate M(InlierPoints)

FitInlier(M2)

Calculate w, k and update the precondition of loop

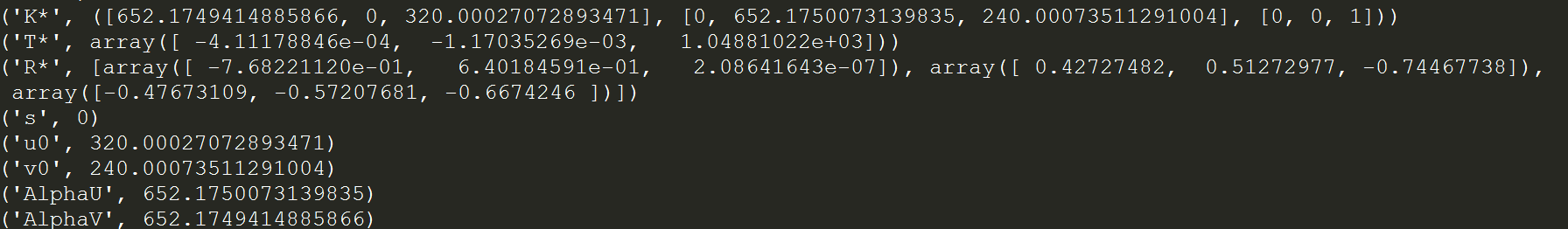
If (k > maxK)

break

}

1. **Results and discussion**





1. **References**

[1] <http://docs.opencv.org/> opencv website.

[2] <https://en.wikipedia.org/wiki/RANSAC> RANSAC wiki

When testing the program on noisy data you will note that RANSAC is not handling well one of the provided cases. Explain the reason for RANSAC not being able to handle this case properly.

Answer： In my opinion, the mistake is introduced by the noisy points are been calculated as the inlier point and inserted in the inlier list.

Digital noise usually represents itself as speckled pixels of color in your images or as a grainy effect, and is generally considered undesirable.

In my opinion, purchase higher-end and newer camera equipment can avoid the nosie.