**REPORT**

Vismaya Veeramanju Kalyan

A20423189

CS 512 FALL-18

Assignment 3

**PROBLEM STATEMENT:**

1. Implement a program to extract feature points from the calibration target and show them on the image. Save the image points detected in a file.
2. Second program to compute the camera parameters using the 3d-2d pair correspondence file. The program must output the camera’s intrinsic and extrinsic parameters.
3. Implement RANSAC algorithm for robust estimation. Parameters used in the RANSAC algorithm should be read from a text file named “RANSAC.config”. Final values of the Estimation should be displayed by the program.

**PROPOSED SOLUTION:**

1. To extract feature points from the calibration target we can use the OpenCV function.

cv2.findChessboardCorners

If the corners are found, add object points and image points after refining them.

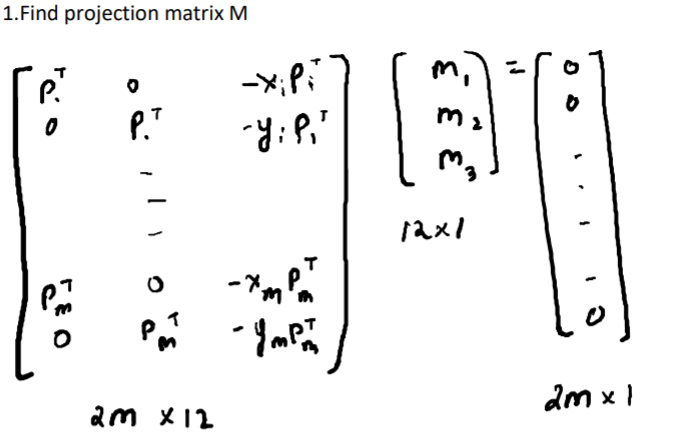
These points are then written into a file.

Here multiple images are being used for planar calibration and these points are written down into different files.

To show the points on the image we use

cv2.drawChessboardCorners

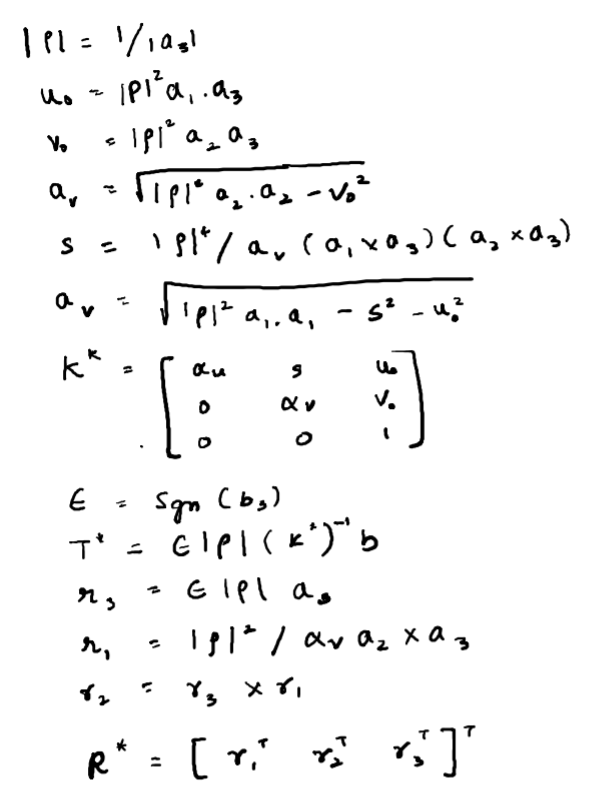
1. I have implemented a program for non-planar in this part using the data given by the professor.



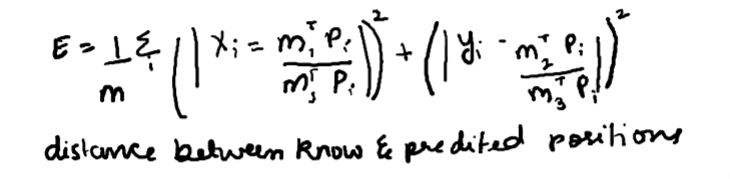
The is a homogenous equation. The projection matrix can be estimated using SVD. A = u d vt Solution is the column of v belonging to zero singular value.

2.Find parameters from M like K\* , R\* and T\*

Equations to solve:



Display the Mean Square Error:



image\_xi = (m1.T.dot(p)) / (m3.T.dot(p))

image\_yi = (m2.T.dot(p)) / (m3.T.dot(p))

mse += ((xi - image\_xi) \*\* 2 + (yi - image\_yi) \*\* 2)

1. Implement the RANSAC algorithm for robust estimation.
2. Principle of RANSAC algorithm:

Repeat k times:

* Draw n points uniformly at random with replacement.
* Fit a model to points.
* Find inliers in entire set with distance less than t.
* Recompute model (if at least d inliers)
* Update parameters (k, t).

Number of points drawn at each attempt should be small in a hope that atleast one set will not have any outliers.

1. Parameters:

N = # points drawn at each evaluation.

D = min # points needed to estimate model ( 6 in our config)

K = # trails

t = distance threshold to identify inliers.

w =

(1-p) = (1 – wn)k  => log( 1 – p) => k log(1-wn) => k

Update w,k every iteration but set a upper bound for k. (1000 in our config)

p = 0.99 in our config

with probability of p at least one experiment does not have outliers.

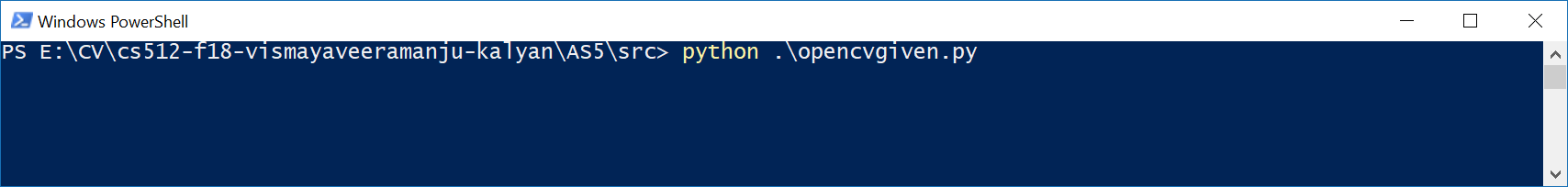
w = 0.5 in our config

The probability that the point is an inlier.

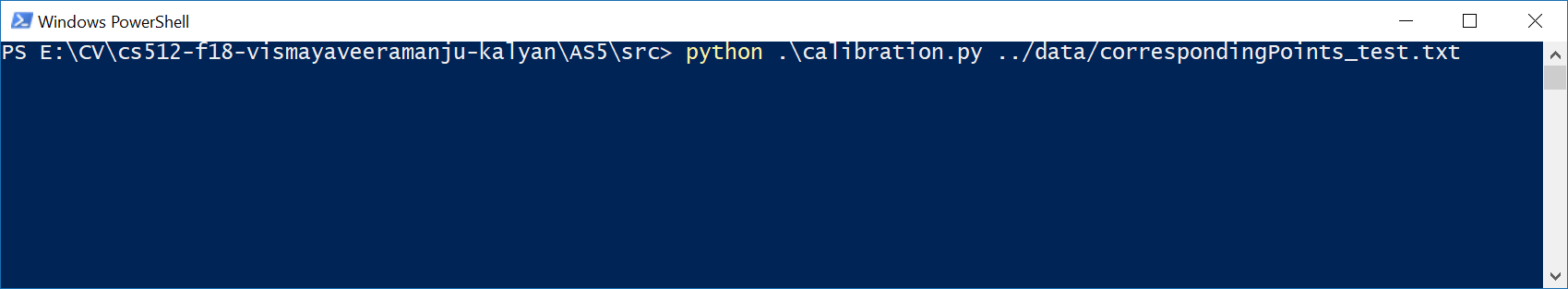
The parameters are stored in the ransac.config file.

**EXECUTION DETAILS:**

1. To run the First program to detect the 2d-3d points and store it to a file.



1. To run the second program.



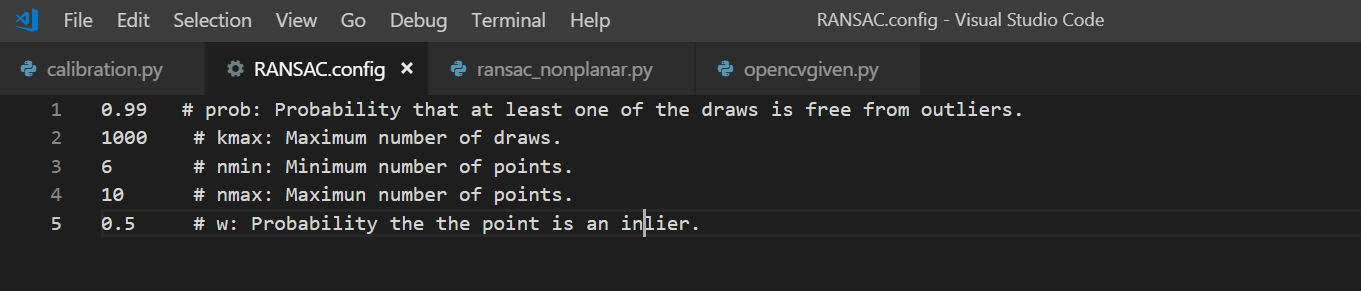
1. To run the RANSAC program

python .\ransac\_nonplanar.py ..\data\correspondingPoints\_test\_noise0.txt \data\RANSAC.config

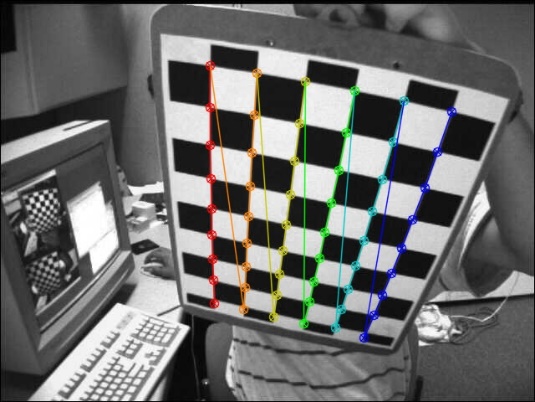
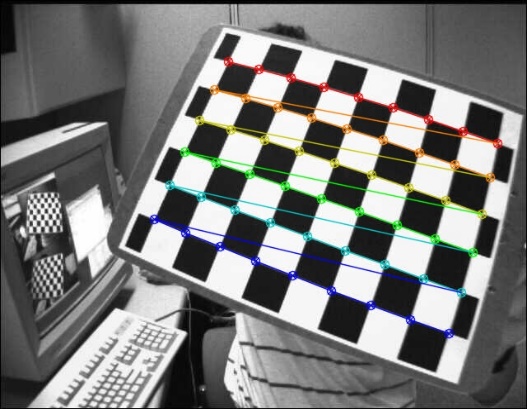
python .\ransac\_nonplanar.py ..\data\correspondingPoints\_test\_noise1.txt \data\RANSAC.config

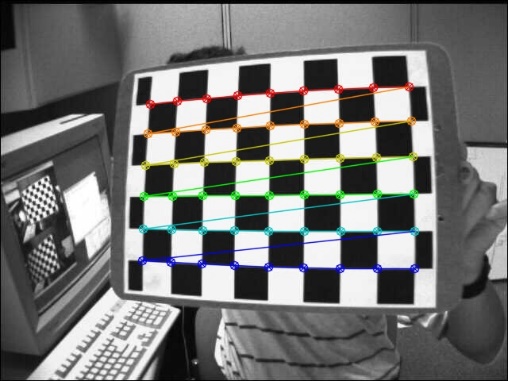
**IMPLEMENTATION DETAILS:**

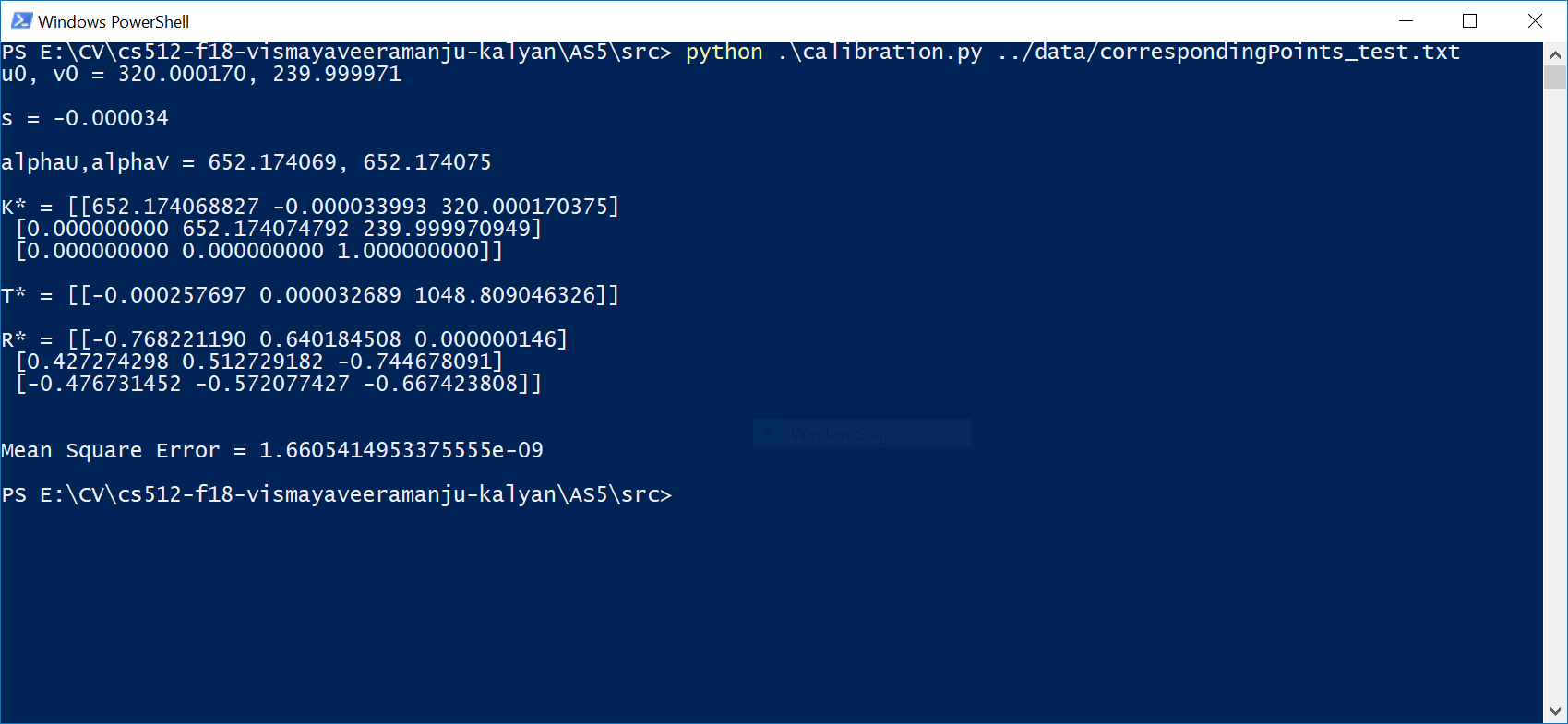
1. The program 1 is done for the planar calibration using the functions provided by the opencv documentation.
2. The second part of the program is done for the non-planar as I had understood the details of the non-planar more thoroughly. Also I Found it hard to implement the planar with the help of the paper provided by the professor.
3. The correspondence file is made by combining the data provided by the professor in the website.
4. The points used for camera calibration is used as an input to the RANSAC algorithm to remove outliers. The output of this is close to the known parameters.
5. The ransac.config file contains the parameters.

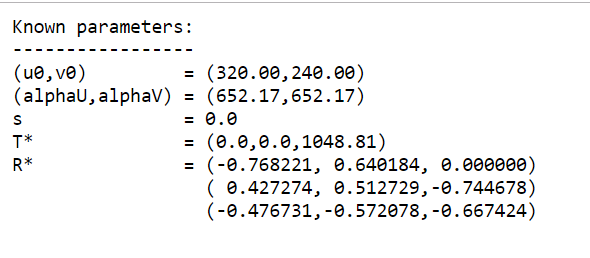


**RESULTS AND DISCUSSION:**

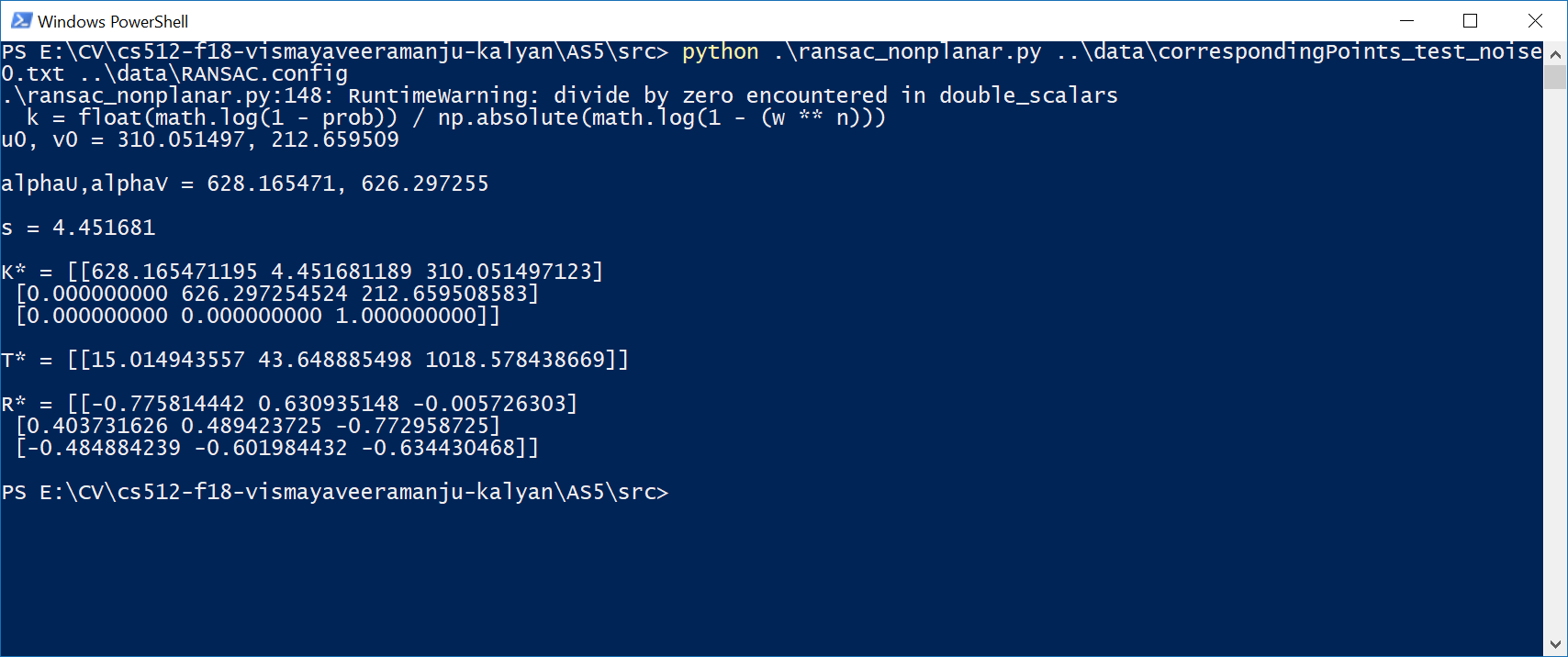
 

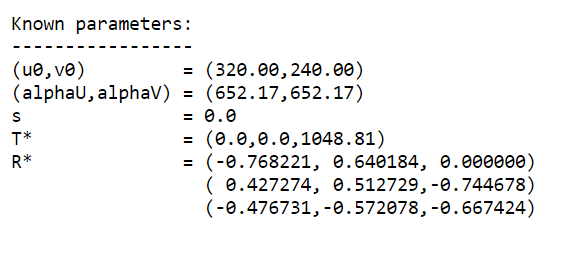


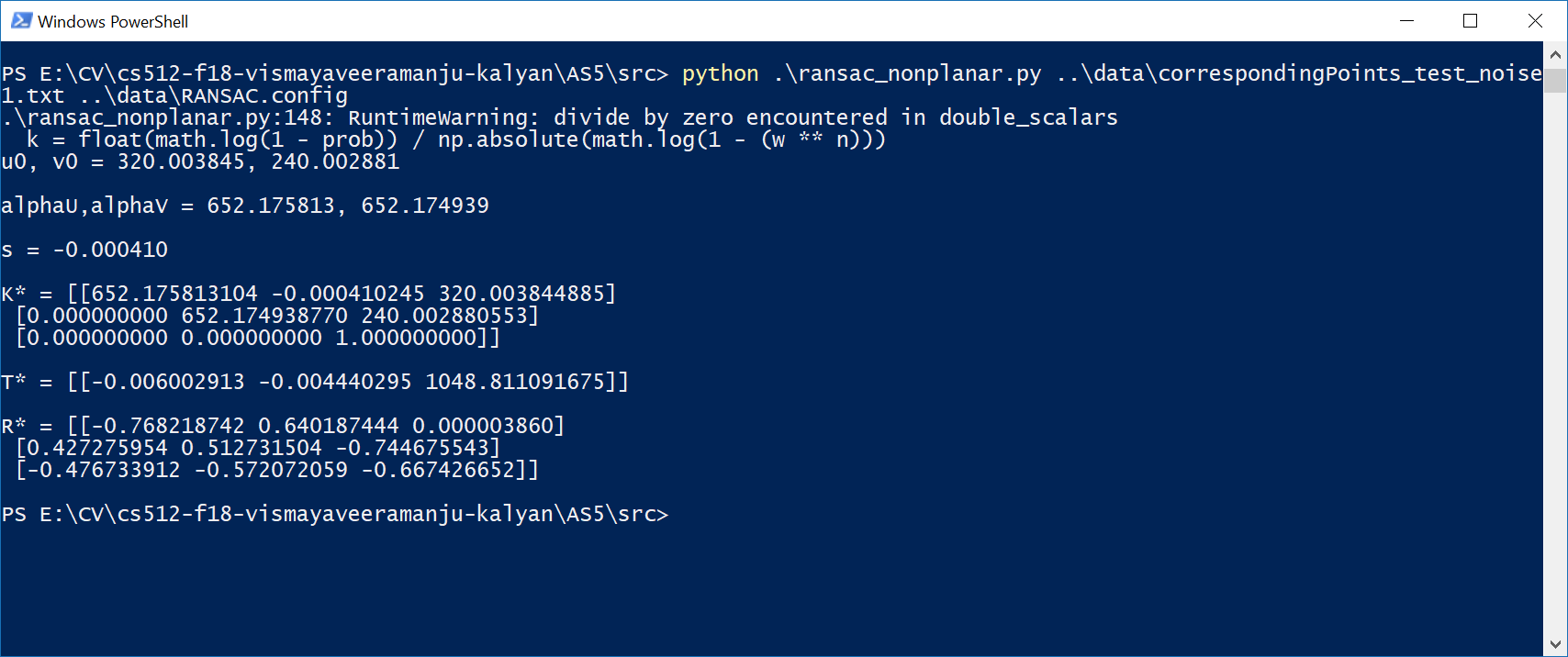


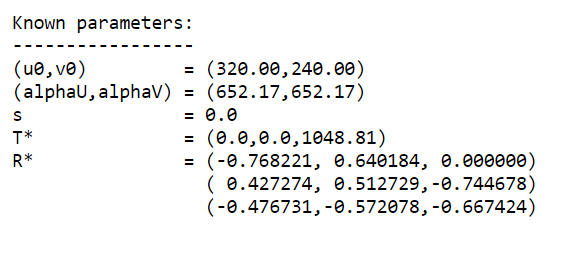


1. Using the noise data.





****



**REFERNCE:**

http://www.cs.iit.edu/~agam/cs512/data/calibration/index.html

<https://github.com/opencv/opencv/tree/master/samples/data>

<https://docs.opencv.org/3.4.3/dc/dbb/tutorial_py_calibration.html>