Course: CS55700

Name: Weijian Zheng

Project 1 Report

Camera Calibration

Background and introduction

For pinhole camera model, we need to use projection parameters to model the image formation process. By applying those parameters to a 3-D coordinates, we can compute the 2-D coordinates of an object in the image plane.

There are two kinds of projection parameters. One is called extrinsic parameters. Extrinsic parameters are majorly used to measure the initial position of the camera. More specifically, we use a 3-by-3 rotation matrix *R* and a 3-by-1 translation vector *T*. Another kind of parameters are called intrinsic parameters. Intrinsic parameters are used to measure the internal parameters of the camera itself. It includes focal length, image center and scale factors. The accurate estimate of above parameters is required for many applications in compute vision. This estimation procedure is called camera calibration.

Algorithms and steps

There are many different kinds of calibration methods. For our project, we have two major steps. In the first step, we compute the 3-by-4 projection matrix *P* using 6 given points’ 2-D and 3-D coordinates. Then, we will normalize the projection matrix *P* to recover parameters. In the following part, the more detail of each step will be presented.

1. Firstly, I build a 12-by-12 matrix *B* includes all the coordinates of our 6 collected points. Its form is as follows:

We can see that each point contributes two rows of matrix *B*.

1. After building matrix *B*, I used SVD to infer the projection matrix. More specifically, if we have , the projection matrix *P* is coming from the last column of matrix *H*.
2. After finding the value and reshaping, I had a projection matrix *P*. Matrix *P* is then normalized using the. Please note that here is coming from the first 3 elements of the last column in *P*. After that, we will rename the left three columns of matrix *P* as *l* and right one column as *r*.
3. In this step, I computed the as the matrix A to recover parameter values. In this step, we need to normalize the matrix *A* by element .
4. Then, in this step, I recovered all parameters using normalized matrix *A* using the given formula.
5. In the last step, I used the predicted parameters to test some points. Also, the distance between our predicted 2-D coordinates and actual 2-D coordinates is computed.

Results:

In this project, we assume there is no skew for our calibrated camera. Thus, *s* is zero and I have the following predicted parameter values.

Below is the extrinsic parameter matrix *K*:

[ 2.40450798e+03 0.00000000e+00 2.90717847e+02]

[ 0.00000000e+00 2.45637478e+03 9.03532466e+02]

[ 0.00000000e+00 0.00000000e+00 1.00000000e+00]

Below is the rotation matrix *R*:

[ -9.06198303e-01 4.22851956e-01 -9.27061406e-04]

[ 1.54520254e-02 -3.44457425e-03 -9.99874677e-01]

[ -4.22846808e-01 -9.06194753e-01 -3.41281223e-03]

Below is the translation matrix *T*:

[ 21.06008007 1.32206631 77.32543014]

Verification:

After the estimation of those parameters, I selected 8 points and used their 3-D coordinates to predict their 2-D coordinates in image plane. Then, I computed the distance between predicted position and the actual position.

To verify the correctness of my estimated parameters, I firstly manually selected 8 points and measured their 2-D and 3-D coordinates. Then, I used the 3-D coordinates to compute their 2-D coordinates. Finally, the distance between predicted position and actual position is computed.

In my program, I printed the average distance of 8 points and it is 13.09 in my program. In conclusion, my predicted parameters can be used to approximately map the 3-D coordinates to a 2-D coordinates in image.

Error analysis:

I think there are two major reasons for some errors in the prediction:

1. The manually selection points process is not accurate. For my project, I used the ImageJ software to select some points and record its coordinates manually. This process cannot be very accurate since there was some inevitably tremble of our hand during the click process. That means is the 3-D coordinates may not exactly match.
2. Another possible reason I think might come from the image rig. For the given image rig, I am not exactly sure about the accuracy of it. If there are some inaccurate of them, our prediction also can be affected.

Also, for my project, I firstly thought there might be some skew for camera that we need to take into consideration. However, after some testing, I found that adding skew back did not affect our accuracy a lot.

References:

[1] Remondino, Fabio, and Clive Fraser. "Digital camera calibration methods: considerations and comparisons." International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36.5 (2006): 266-272.