Huan-Yang Chang

Team G: Excalibr

Teammates:

Man-ning Chen, Yiqing Cai, Sambuddha Sarkar, Siddharth Raina

EXCALIBR

ILR04

11/10 2016

Individual Progress

1. Implementing the geometry calibration

To make the minimum valuable product, I implemented the off-the-shelf camera calibration in OpenCV first. For the future improvement, I modularize the code to four unit (Fig1).
It would let future improvement easier to conduct because I already understand the input and output of each module unit in the program. In the parameters saving part, out team also discussing using either the "text-formatted(.txt)" or " Extensible Markup Language(.xml)" format as our storage format. For text format, it had easy accessibility. However, we had to parse the format every time we read from the file. For XML, it was a structured document and had a better readability for the human. Finally, we decided to use text file rather than XML file to storage our results because the XML format might need another API to read the file and using txt is convenient for future use.

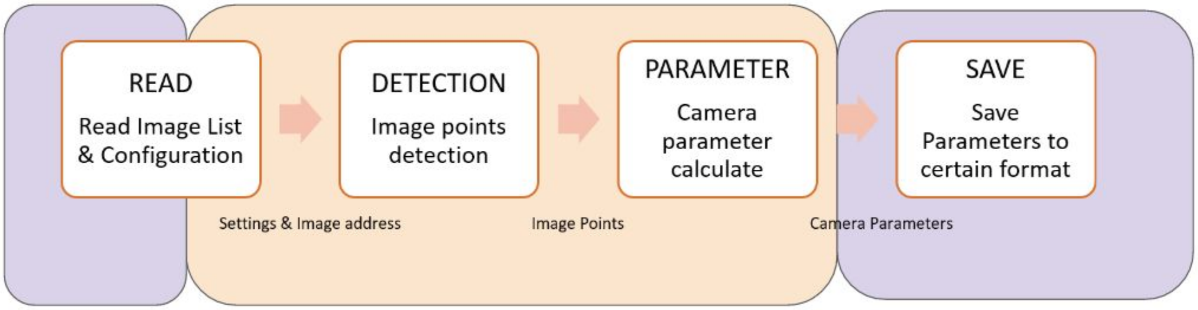

Fig 1. Geometry calibration process

2. Geometry calibration evaluation methods

For evaluating our geometry calibration result, there might two ways to do, one is reprojection error and another one is reconstruction error(Fig 2). In reprojection error, we would first convert the object points in world coordinate to image coordinate by using projection matrix we got, then calculate the distance of the reprojection points and the corresponding points in the image.The another method was using the projection matrix to reconstruct the point in world frames. In our case, we took the movement of Aerotech arm as a ground truth and using reconstruction method to estimate the movement of the target(Fig 3.).Then we can compare the movement to get the accuracy estimation. Table 1 was the current result.

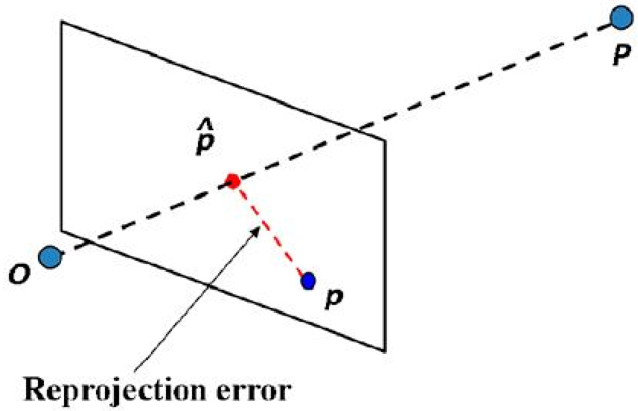

Fig 2. Reprojection error[1]

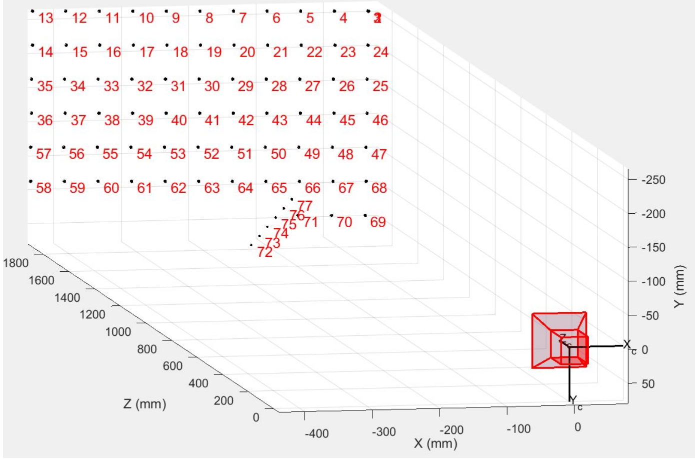

Fig 3. Reconstruction error

[1] Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman (p312). Cambridge University Press, 2003.

Table 1 Current result in geometry calibration.

Title	Value
Reprojection error	0.3 pixels
Reconstruction error	300 **micrometer**

Challenges

1. Expertise in C++ and OpenCV

In the first beginning, I tried to combine the geometry calibration code for 3D objects. However, because of my limited knowledge, it was hard for me to finish this task. By consulting with our project leader in the company, he recommended me try to use the off-the-shelf camera calibration and make the code modular first to make the minimum valuable product. In the future, we could base on this structure to change the code.

2. Slowness in OpenCV camera calibration

After I implemented the geometry calibration, I found that the camera calibration function in OpenCV was designed for general use and was not designed to deal with lots of images and super accurate calibration, hence the speed and accuracy were not satisfied for our requirements. In the next step, I will try to consult the employee in Oculus and I will also analyze the cause of the slowness in the OpenCV. On the other side, we used about 700 images to conduct camera calibration for the whole field of view. However, the In Zhang's method[2], it only took 10~20 images to calibrate the camera. Hence, we may find a way to reduce the number of the image to an accepted amount and at the same time keep the good calibration results.

Teamwork

Work was done in this week	Name
AEROTECH robot trajectory generation, velocity profiling & troubleshooting.	Sam
Implement geometry calibration and validate the accuracy in camera calibration	Peter / Sid

[2] http://research.microsoft.com/en-us/um/people/zhang/calib/

Photometric calibration: Solving memory problem, Convert code from Matlab to C++, create the inverse response parameters	Cece/Mandy
FPN : Convert code from Matlab to C++ and create FPN parameters	Mandy/Cece

In this week, the team tried to make individual work more suitable for integration. I tried to implement the geometry calibration in Linux with OpenCV library and also did literature research to confirm the accuracy evaluation methods. Sid was working on image processing for geometry calibration. Sam was working on the Aerotech arm PSO problem and generated the robot arm trajectory. Cece and Mandy were working on memory problem in photometric calibration. They used the different algorithm to create the inverse response function without the memory issue. Besides, they also converted their code from Matlab to C++ for integration.

Future Plans

1. Complete the geometry calibration pipeline

- I will combine the image processing part into geometry calibration to complete the geometry calibration pipeline.
- Define the Data input and output format in the integration of geometry pipeline.

2. Improve the slowness problem in geometry calibration

- Increase the speed of the geometry calibration :I will try to understand the function “calibrateCamera” to find the cause of slowness and discuss with the sponsor for this problem.
- Reducing the number of the image in calibration : The possible method was to instigate the effect of a single image in calibration and find out the minimum number of the images and required positions of the target.