Yiqing Cai

Team G: The ExcalibR

Teammates:
Huan-Yang Chang
Man-Ning Chen
Siddharth Raina
Sambuddha Sarka

ILR06
Feb.2, 2017

Individual Progress

Overview

For this stage of project, I was primarily responsible for building the camera parameter models for the following path planning procedure. We planned to do the simulation in RobotStudio for the whole dome setup and calibration process, so building models would definitely be the first step for the testing and debugging.

Implementation

The camera parameters consist of intrinsic matrix (focal length and principal point), and extrinsic matrix(rotational matrix and translation vector). First, I generated 200 points randomly on the sphere, and connected all the points with the center of the sphere.

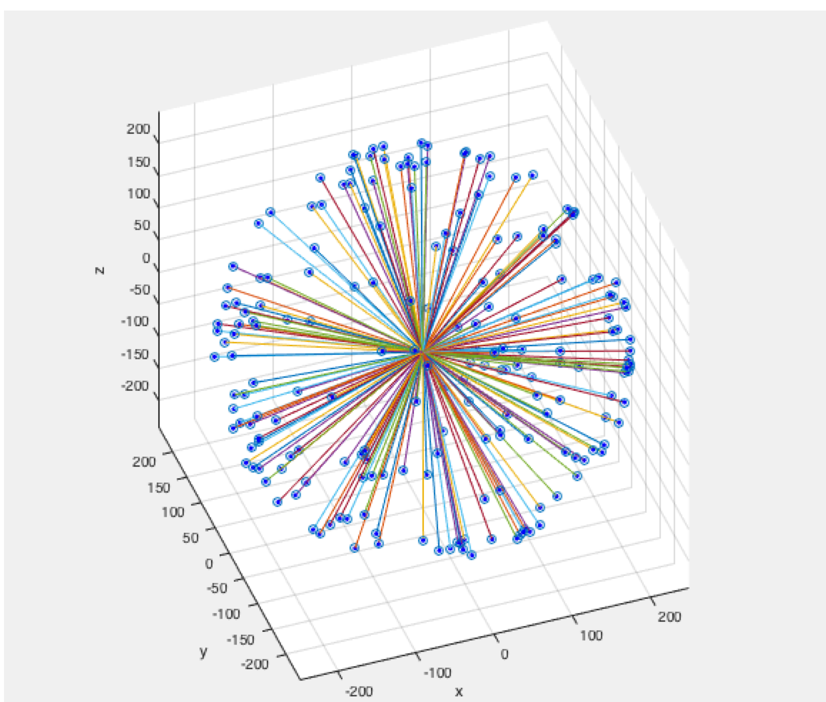

Figure 1. Simplified model of cameras mounted on the dome and looking at the center

This is actually a simplified model of cameras mounted on the dome and looking at the center of the dome, as shown in Figure 1. The dome in real is a sphere with the diameter of 5m. And there will be 100 ~ 200 cameras mounted on any position of the dome surface and looking at the center of the sphere. So in the simplified model, I generate 200 random 3D positions on the sphere, regarding them as the positions of 200 cameras.

For the intrinsic matrix, all the parameters are in the unit of pixels, so I need to convert the unit of focal length and principal point into pixels. The principal point is the center of the image plane and we regard the left top corner as the origin of the image plane. In this model, I ignored all the skews and distortions, as those parameters will increase the complexity of calculation and have little effect on the path planning procedure.

For the extrinsic matrix, I need to compute the rotation and translation between the world frame and the camera frame. First we need to compute the uvw vectors of the camera frame under the world coordinate and then compute their relationships. The model of the camera frame is shown in Figure 2, the w vector is the opposite to the 'look' direction, and the up direction is the absolute up direction under the world coordinates. We can then compute the v vector based on the w vector and up direction, and then compute u as the cross product of vector w and vector v. (Note that all the vectors are unit vectors.)

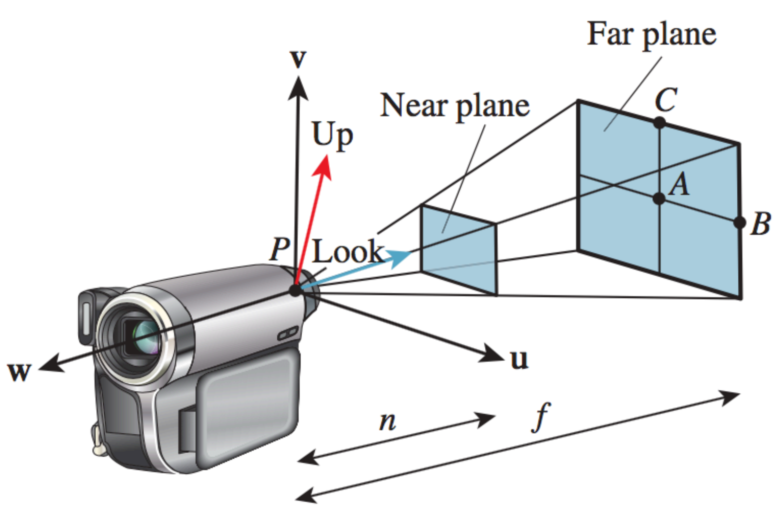

Figure 2. The Camera Coordinate uvw and the 'look' direction

The following equations clearly describe the calculation of the camera frame.

$$\mathbf{w} = \frac{-\mathbf{look}}{||\mathbf{look}||} = \mathcal{S}(\mathbf{look}).$$

$$\bar{\mathbf{v}} = \mathbf{vup} - (\mathbf{vup} \cdot \mathbf{w})\mathbf{w}$$

$$\mathbf{v} = \frac{\bar{\mathbf{v}}}{||\bar{\mathbf{v}}||} = \mathcal{S}(\bar{\mathbf{v}}).$$

$$\mathbf{u} = \mathbf{v} \times \mathbf{w}.$$

For the v vector, we first project the up direction onto the plane perpendicular to w, and hence perpendicular to the look direction as well.

Once we have the uvw vectors, we can get the rotation matrix from world frame to the camera frame as R = [u v w], and the translation vector t as the position of the camera. When we put the [R|t] together and get the inverse of the homogenous form, we can get the extrinsic matrix of the camera.

For the 200 cameras, I did the above calculation and save all the intrinsic and extrinsic matrix in a cell for the following path planning process.

Challenges

The main challenges I faced during this task were:

I spent a lot of time trying to figure out the exact relationship between the camera frame and the world frame. It is not a difficult relationship, but it is a little confusing during the calculation process and needs patience to get the correct answer.

Teamwork

Work undertaken by each team member is as follows (see Table 1):

Member	Tasks
Huan-Yang Chang	Building the model of dome and ABB Robot arm in the Robot Studio
Man-Ning Chen	Segmentation of the color checker for the color calibration
Yiqing Cai	Building the camera parameter models for 200 cameras on the dome
Sambuddha Sardar	Rendering of the calibration target and cameras in Blender
Siddharth Raina	Other Sensor noise Correction

Table 1. Team co-work

Our team worked with great coordination during execution of the second stage of this project. We communicated during the entire task and solved problems together. Sam was working on the rendering of the 3D calibration target and cameras in Blender, trying to generate virtual images of all the cameras. Peter was working on building the model of dome and ABB Robot Arm in the Robot Studio, and trying to generate code for the control of robot arm according to the 3D points for the motion path. Mandy was working on segmentation of the color checker for the color calibration. I was working on building the camera parameter models for 200 cameras on the dome for the following path planning procedure. Sid is trying to work on other kinds of sensor noise except for the FPN and PRNU we dealt with last semester. We faced many difficulties but we worked them out eventually as a team.

Future Plans

From now on, my task will be focused on the designing the path planning code for ABB Robot Arm. As I have completed building camera parameter models, I can calculate the projection of calibration target on the FOV of cameras, with projection at different locations in the dome, I can calculate the necessary 3D points that the robot arm should be able to reach.