Yiqing Cai

Team G: The ExcalibR

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

ILR02
Oct. 21, 2016

Individual Progress

Overview

For this stage of project, I was primarily responsible for setting up the cameras and doing sensor calibration. We first used the EVT camera API to trigger multi cameras at the same time by software, and then tried to modify the random noise and fixed pattern noise caused by sensor (CMOS photodetector). We used the emergent vision technology HR-12000 cameras and AEROTECH linear actuator (act as the robot arm) in this stage of work.

Implementation

The first step is to use the EVT camera API and try to trigger multi cameras capturing images of the calibration target. The calibration target is the similar size to the human face and is shown in Figure 1.

The calibration target we built is showed in Figure 1 :

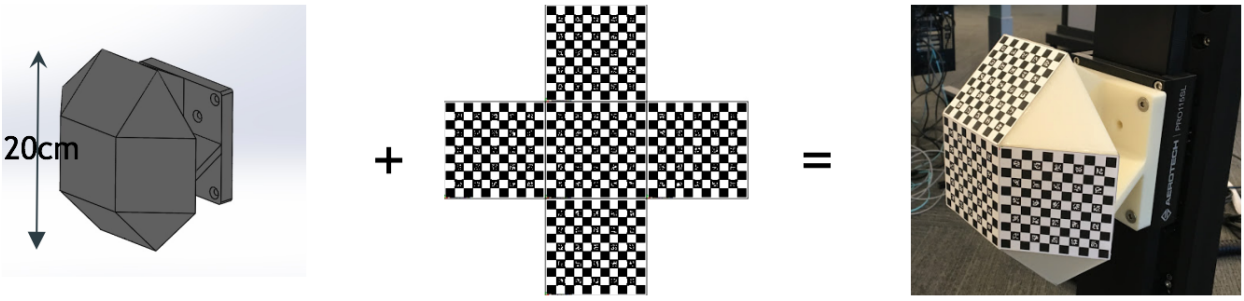


Figure 1. The calibration target we built

For the triggering part, we write C++ code on the Visual Studio platform to trigger multi cameras at the same time. The general process is to make multiple camera open, stream and record all the images. One tricky thing is that the cameras need to be closed by explicating API commands and since our commands for closing camera is put at the last part of our code, the program wouldn't execute the command if it stops before the command. So we need to close

the cameras manually before we rerun the code if it stopped in halfway. We spent some time figuring out this and other parts went well.

The second work is to modify the random noise and fixed pattern noise caused by CMOS. The fixed pattern noise is consists of two parts, the Dark Signal Non-Uniformity(DSNU)——the offset FPN and the Photo Response Non-Uniformity(PRNU)——the gain FPN. DSNU is seen as an offset between pixels in dark, it can be corrected by subtracting a dark frame. PRNU is seen as a responsive variation between pixels under illumination and can be corrected by offset and gain for each pixel.

The formula is y = ax + b, where b is the fixed pattern noise, x is the truth pixel value of the image, a is the gain fixed pattern noise and y the pixel value we get.

For b, we can get the value for each pixel according to the image we took in the dark current. For a, we need to find a way figuring out the calculation method. The way we are trying is to take three frames of images consecutively and then try to unify a for all the pixels in the frame, and make the value proportional to the true value.

For data collection, we use the harris corner detection and BRIEF descriptor to find out the exact pixels the frame has shifted, and then we can do the calculation based on that.

The result of corner detection and match result is shown in Figure 2 and Figure 3:

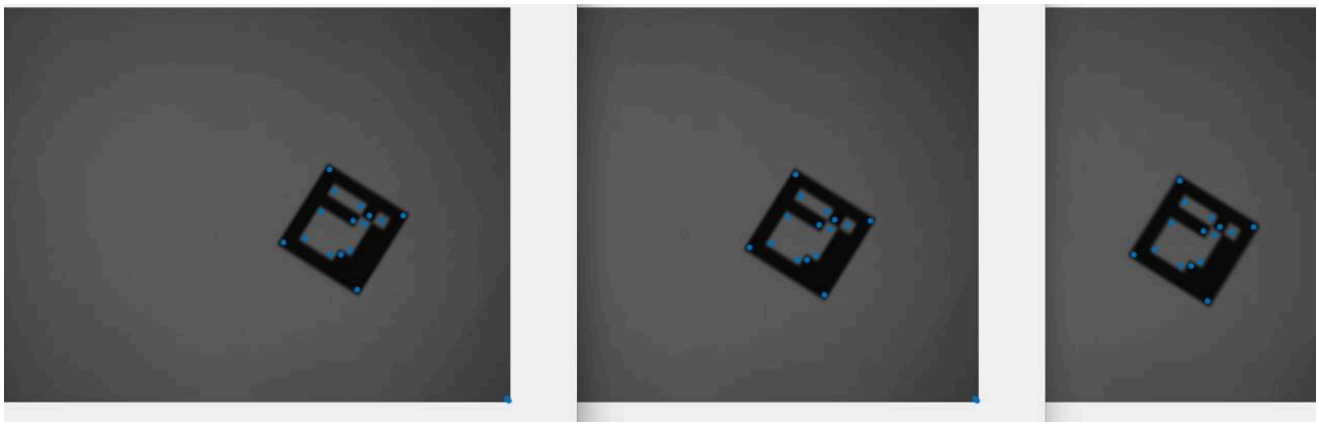


Figure 2. The result of corner detection

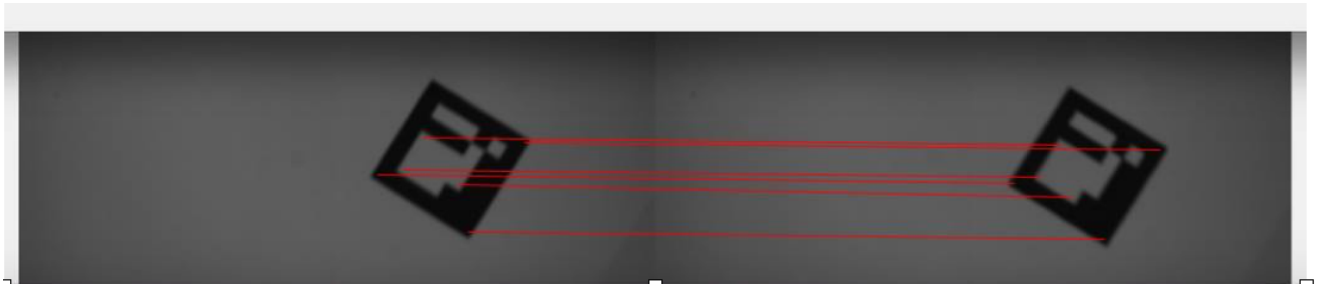
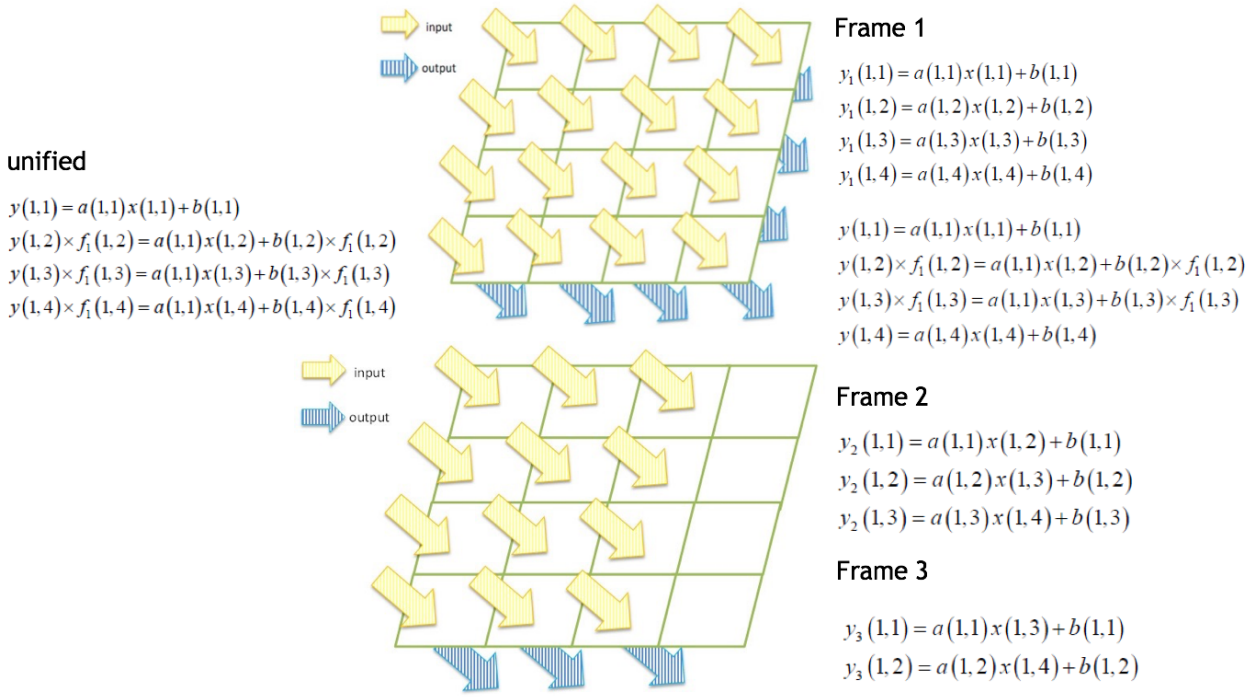


Figure 3. The result of match based on BRIEF descriptor

Once we get the shifted parameters, we can unify the gain FPN——a, as follows:



Once we unify the gain FPN, we can get the pixel values proportional to the true value of each pixels, then the noise problem is solved.

The uniform light source and the camera we use is shown in Figure 4:

Figure 4. The uniform light source and the EVT 12000 camera

Challenges

The main challenges I faced during this task were:

1. Cannot trigger multi-cameras at the same time: We implement EVT API in our code (on visual studio platform) and change it to make multiple camera open, stream and record. However, we found that every time when we stop our program while it was running or it stop itself because of bugs in the program, we would face failure reopening the cameras next time. At first we thought we might be using the API codes wrong. After several tests, we found out that cameras needed to be closed by explicit API commands and since our commands for closing camera is put at the last part of our code, the program wouldn't execute the command if it stop before the command. So we need to make sure the camera is closed before we test the program from the start.

2. Before we can implement geometry calibration, we need to remove the fixed pattern noise (FPN) caused by the photodetector, that is, we need to first implement the sensor calibration. For the offset FPN, we can subtract it in the dark current, however for the gain FPN, we need to find a way to remove it in order to get digital images of better quality. We are trying to get a sequence of images with a marker to indicate the frame shift and to unify the coefficient before the true value of a specific pixel so that we can get better digital images.

3. Failure in corner detection of high resolution images: during our process of calculating the value of gain FPN, we need to firstly find out how much pixels the frame had shifted. However, the Harris corner detector failed to detect corners unless we down-sampling the images. We are guessing that we would need larger kernels for the gaussian pyramids and trying to enlarge kernel size of the detector or using other detection methods. Another way is to implement the apriltag detection library.

Teamwork

Our group mainly divided into two sub-group of robot arm setup and Aerotech control group and camera setup and sensor calibration group. Work undertaken by each team member is as follows (see Table 1):

Member	Tasks
Huan-Yang Chang	Robot arm and Aerotech control setup
Man-Ning Chen	Camera setup and sensor calibration
Yiqing Cai	Camera setup and sensor calibration
Sambuddha Sardar	Robot arm and Aerotech control setup
Siddharth Raina	Camera setup and sensor calibration

Table 1. Team co-work

Our whole team worked with great coordination during execution of the first stage of this project. We communicated during the entire task and solved problems together. Peter and Sam were working on the Robot (AEROTECH - 3 Prismatic Joint) control, while Mandy, Sid and me were working on the camera setup, multi-cameras triggering and sensor calibration part. We faced many difficulties but we worked them out eventually as a group.

Future Plans

From now on, our team will work on two main areas of our project:

Calibration Process:
As we have finished setting up the robotic arm and the multi-camera capturing system, we will begin to collect image data and start to carry out the calibration process. We are now in the progress of implementing the Fixed Pattern Noise (FPN) calibration, trying to compensate for vignette patterns which is caused by photometric staff and lens artifacts. We plan to capture three consecutive images with a fixed frame shift and calculate the gain FPN according to the neighbor pixels. After that, we can calibrate camera responses (photometric) and correct lens distortion (geometric), and estimate the extrinsic parameters between all cameras.

Aerotech Control and Motion planning:

For the motion planning part, we are now able to move the calibration target in a certain predefined pattern, but the Aerotech can only generate pulse along the X axis, but not Y and Z axis. We need to figure out the reason of failure and program to control the robotic arm to generate pulse in a more flexible pattern. We plan to search for the sample code from the vendor and figure out the function's meaning of the PSO output. We hope we can soon trigger multi cameras to capture images whenever and wherever the robotic arm generate a pause.