Yiqing Cai

Team G: The ExcalibR

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

ILR03
Oct. 28, 2016

Individual Progress

Overview

For this stage of project, I was primarily responsible for sensor noise correction (fixed pattern noise) and part of the photometric calibration (the camera response function). By the second progress review, I have done with the sensor noise correction and on halfway of the photometric calibration. We used the emergent vision technology HR-12000 cameras and AEROTECH linear actuator (act as the robot arm) in this stage of work.

Implementation

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

In the last ILR, we were trying to get the gain fixed pattern noise ‘a’ by unify it for each pixel based on the shifted frames. However, we found that this wouldn’t work unless we are able to move the frame one pixel by one pixel, which is not possible to carry out by our existing image capturing system. So finally we implement the method called flat field correction to solve this problem.

So we take 100 frames of dark images, and then take the average of them as the dark noise ‘b’ (offset FPN), which is an image matrix. Then we take 100 frames of white surface images (uniform surface with uniform light source) with the same exposure time, and take the average of them as the pixel value we get ‘y’, which is an image matrix. For the truth pixel value ‘x’, we take the median of ‘y’, and this should be a single value because we use a uniform white surface with uniform light source. And then we can calculate the gain FPN ‘a’ according to the formula, which is also an image matrix, because for each pixel the FPN noise is different.

The comparison of the noisy image and the calibrated image is shown in Figure 1. To make it more intuitive, we calculate the variance of both images, and by theory, each pixel value of the correct image should all be the same. We can see that by removing the FPN noise, the variance of the image is much more smaller, which means the method we applied is effective.

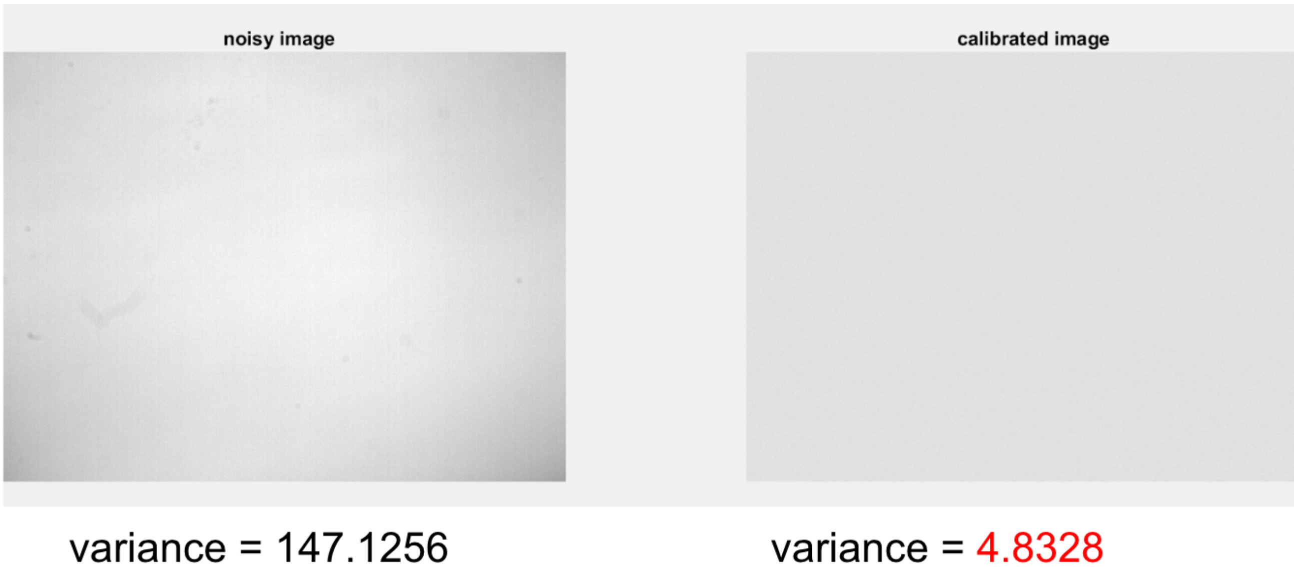

Figure 1. The noisy image and the image with no FPN noise

Then we move on to deal with the camera response function (CRF). This depends on the exposure time. Longer exposure times (or integration times) mean that more light will hit the camera sensor. Hence, pixel values will look brighter. Similarly, if the exposure time is shorter, less light will accumulate on the sensor. Hence, pixel values will appear darker. This change in intensity is usually called the camera response function. Response in this case means how the hardware responds to light. This response is determined as a function per pixel. The idea is that we want to figure out the true pixel color irrespective of the exposure time.

In order words, the hardware response to variations in exposure X (which is the product of the irradiance E the film receives and the exposure time t). We can then implement the algorithm according to the following formula to get the inverse response function and the irradiance for each pixel (the pixel value Z and the exposure time t is known).

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \le \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}$$

Because the algorithm we implemented would take a lot of memory so we first tested on down sampled images, and what we got is shown in Figure 2.

Figure 2 -1. The result of implementing CRF algorithm (at least two images with different exposure time is needed, and more images will produce more accurate results)

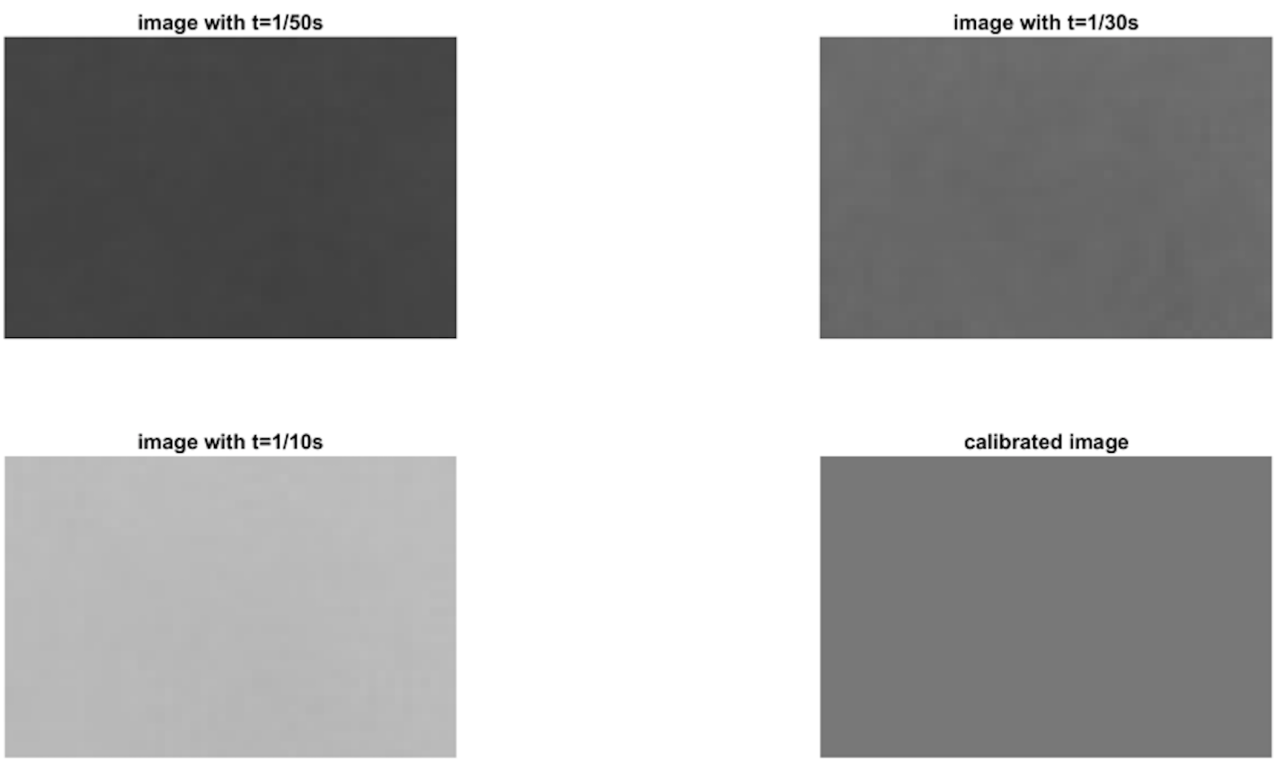

Figure 2 -2. The result of implementing CRF algorithm (at least two images with different exposure time is needed, and more images will produce more accurate results)

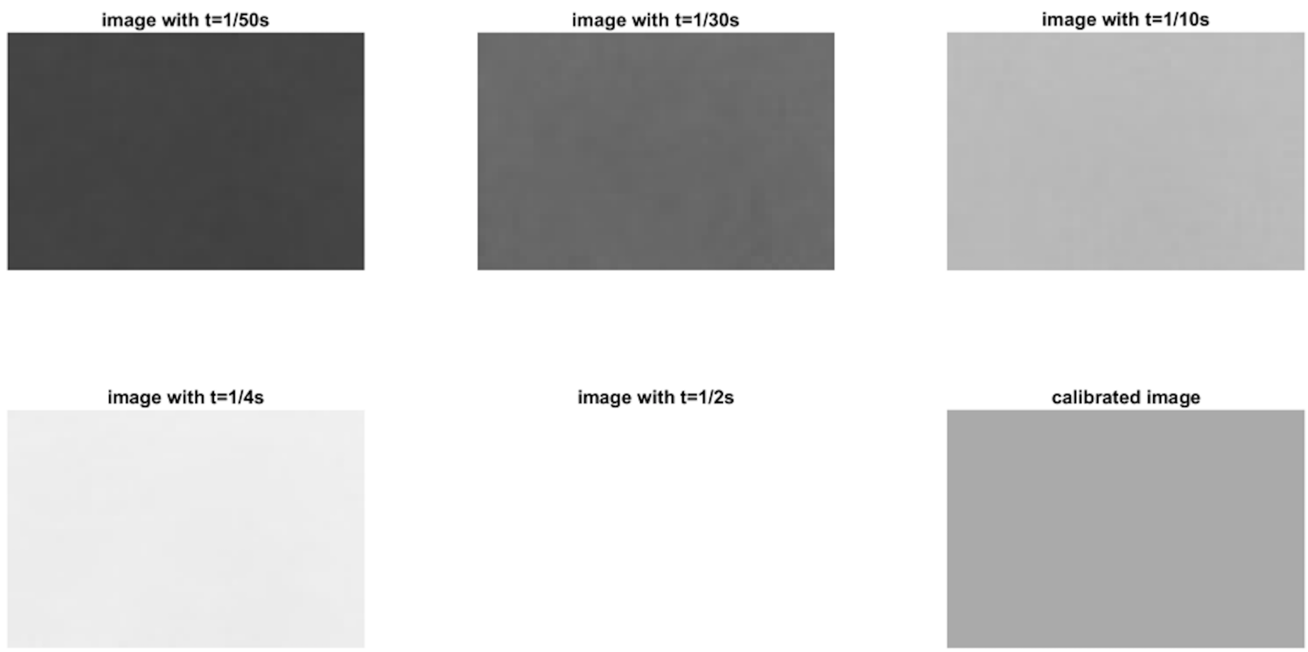

Figure 2 -3. The result of implementing CRF algorithm (at least two images with different exposure time is needed, and more images will produce more accurate results)

Challenges

The main challenges I faced during this task were:

1. The CRF algorithm take a lot of memory so we are not able to test on the images with full resolution. For now we down sampled the images for testing, and we are planning to divide images into small sub-images in order to solve the memory insufficiency problem.

2. When gathering image data for testing, it is hard to obtain 100% uniform surface and 100% uniform light source. We also need many frames of static scenes as the input of the algorithm, and it is also hard to obtain 100% static scenes. As we are expecting outcomes with definitely high accuracy, we need to figure out the error correction methods.

Teamwork

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

Member	Tasks
Huan-Yang Chang	Robot arm motion planning and PSO triggering
Man-Ning Chen	Sensor noise correction and photometric calibration
Yiqing Cai	Sensor noise correction and photometric calibration
Sambuddha Sardar	Robot arm motion planning and PSO triggering
Siddharth Raina	Sensor noise correction and photometric calibration

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. Peter and Sam were working on the Robot (AEROTECH - 3 Prismatic Joint) motion planning and PSO triggering, while Mandy, Sid and me were working on the sensor noise calibration and photometric calibration part. We faced many difficulties but we worked them out eventually as a team.

Future Plans

From now on, my task will be focused on two main areas of our project:

Photometric calibration:
We are now on the halfway of photometric calibration, so in the following days, my main task will still focus on the camera response function. First is to work out the memory insufficiency problem and then we still need to test on more different kind of images which is not the white surface. In order to integrate into the final system, the algorithm needs to be robust enough.

Geometric lens distortion correction:

If I am able to complete the camera response function soon, I will join the geometric calibration part. The plan is to first study the existing algorithms and pipelines of geometric calibration designed by our sponsor Oculus, and then we need to implement it for multi cameras and maybe add some new ideas for improvement.