

**ILR04: Progress Review 3**

Maitreya Naik

Team B: Arcus

Teammates: Logan Wan, Clare Cui, Angad Sidhu

ILR03

Nov. 11, 2016

## 1. Individual Progress

Since the last progress review, I was responsible for the calibration of the camera intrinsic parameters, camera-IMU extrinsic transformation, and verification of the power board layout designed by Logan Wan. I was also responsible for a majority of the electronic component sourcing for the power distribution board and custom cables.

### 1.1. Camera Intrinsic Calibration

Camera Intrinsic Calibration comprises of finding the intrinsic parameter matrix,  $K$ , displayed in Fig 1

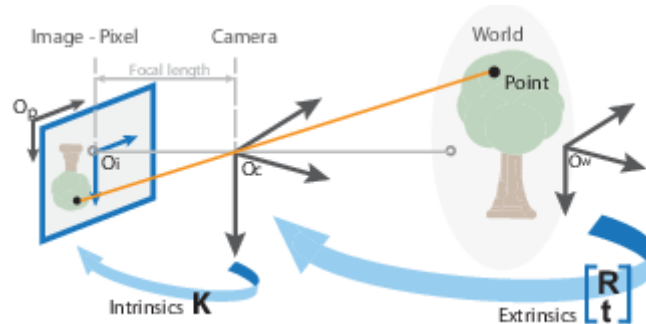


Fig 1: Camera Intrinsic and Extrinsic relations [from [1]]

From Where 
$$K = \begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Here,

$(f_x, f_y)$  — Focal length in pixels.

$$f_x = F / p_x$$

$$f_y = F / p_y$$

$F$  — Focal length in world units, typically expressed in millimeters.

$(p_x, p_y)$  — Size of the pixel in world units.

Since we are using a fish eye lens with about  $180^\circ$  FOV, the standard ROS camera\_calibration package [2] and the standard MATLAB camera calibration app in the Computer Vision Toolbox [3] didn't provide acceptable results (acceptable pixel error  $\epsilon$   $[-0.5px, 0.5px]$ ). Hence, we tried to fix Kalibr [4]. In the meanwhile, we also came across Davide Scaramuzza's OCamCalib Toolbox for Matlab [5] for Fisheye lenses.

For calibration, we either used an Apriltag board or a checkerboard. Each software would be provided with multiple images with the pattern board occupying the maximum area of the lens as well as the individual corners and edges. The software would then look for the corners between the squares that made up the whole pattern.

With OCamCalib, we did get an intrinsic matrix with a pixel error of less than  $0.8px$ , and the toolbox also had a handy function of automatic corner extraction. However, the calibration process was manual in selecting some corners from some images in case they weren't detected

in the provided images. Though reliable, this was a time-consuming task. In case of Kalibr, there was no way of helping the software in finding all the corner points in case it fails to detect them in a certain image, and it eventually discards the image from its calibration calculations. And although it took the same time as OCamCalib, Kalibr could be provided with a rosbag of the calibration image stream and left to do its job on more than 300 images in less than 5 minutes. Fig 2 displays an intrinsic calibration result.

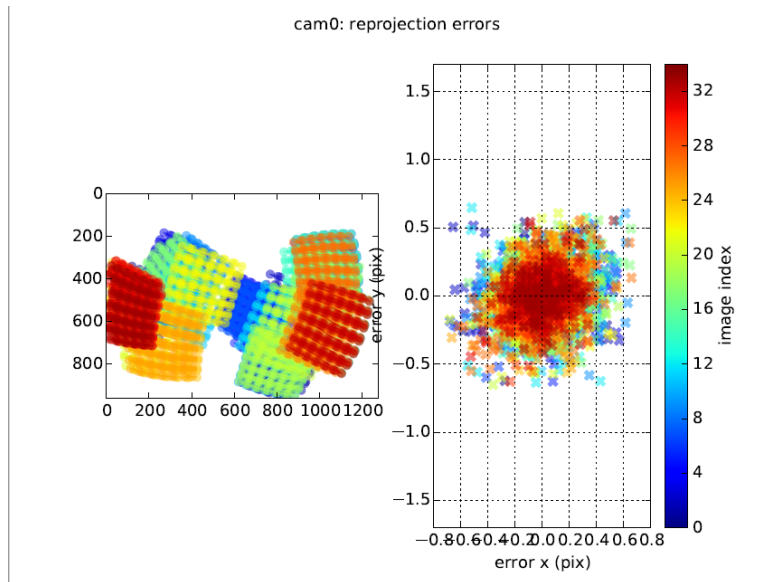


Fig 2: Kalibr camera intrinsic calibration outputs

The plane on the left displays the different apritag boards detected color-coded according to the image index in the rosbag stream. The right plane displays image re-projection errors after Kalibr [4] applies the calculated calibration matrix,  $K$  back to the image stream. As is visible, the pixel error is within  $[-0.8\text{px}, 0.8\text{px}]$  in the x-axis and  $[-0.6, 0.6\text{px}]$  in the y-axis. This was a reliable calibration result and we will try to improve this after the first hardware integration of our drone is complete.

Kalibr [4] also produces a YAML file that contains the calculated intrinsic camera parameters. The YAML syntax is explained in [6]

## 1.2. Camera-IMU Extrinsic Calibration

The camera-IMU extrinsic calibration comprises of calibrating temporal and spatial coordinates of an intrinsically calibrated IMU and a camera. This was executed using Kalibr [4] and required the camera intrinsic YAML produced by the software as well as an IMU noise parameter YAML as described in [6]. The IMU Noise Model can be derived using the technique described in [7]. Fortunately, the IMU Noise Model of the sensor used (VN-100 [8]) was derived by other lab staff of the Planetary Robotics Lab at Carnegie Mellon University where our project is being developed. Hence, the calibration was a matter of making a rosbag of the IMU data using the imu-vn-100 package [9] at 200Hz and the camera data at the same time at 20Hz and running the Kalibr calibration on the same bag file with the intrinsic parameter YAML files. T

The result of the calibration is a report containing estimated IMU accelerometer and gyroscope biases over time, the estimated respective acceleration and angular velocities errors as depicted in Fig 3, and the image reprojection errors as depicted in Fig 4. Since the reprojection errors are within 1px for the later images, another calibration will be performed to acquire reduced errors.

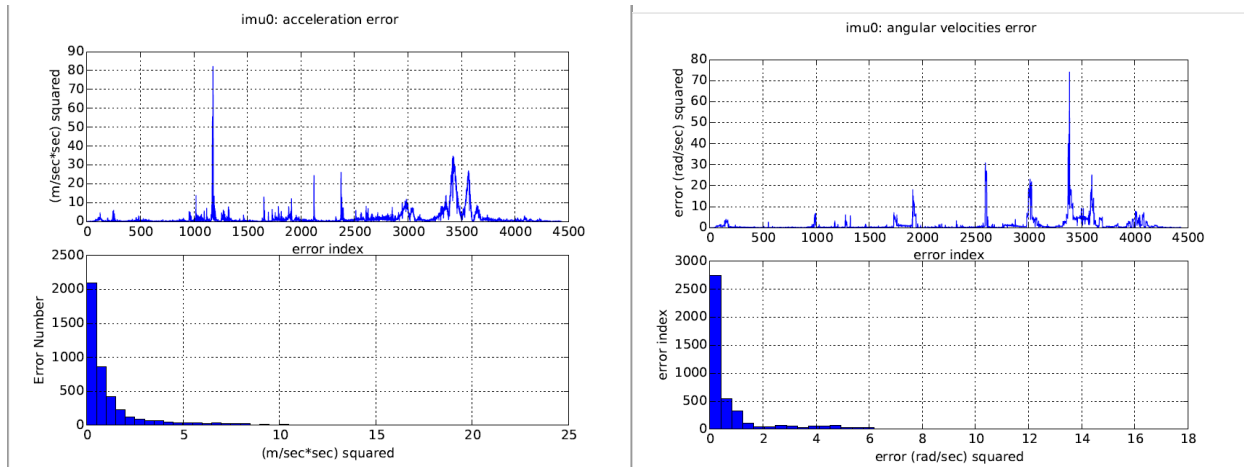


Fig 3: Kalibr Camera-IMU extrinsic calibration. Left: IMU acceleration calculation error; Right: IMU angular velocity calculation error

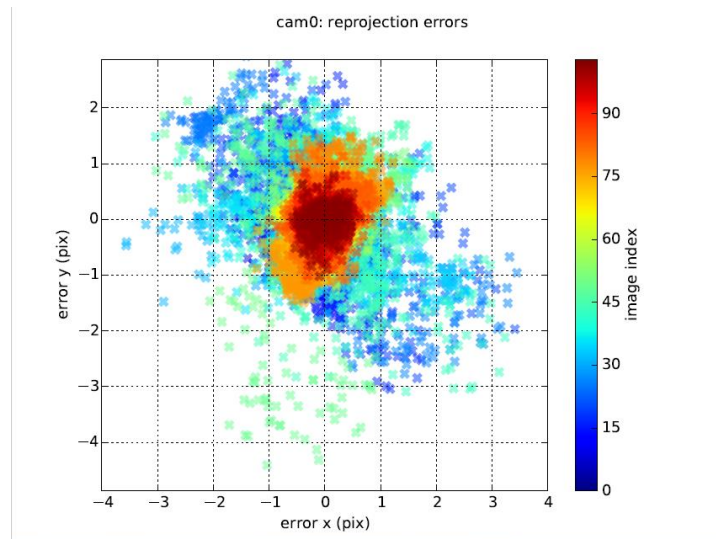


Fig 4: Kalibr Camera-IMU extrinsic calibration. Camera reprojection errors

### 1.3. Power Distribution Layout

After Logan Wan completed the initial design of the Power Distribution board layout, I reviewed it and suggested changes to shorten connections to LEDs, ensure placement of capacitors near appropriate terminals for effective noise removal as well as changing resistor pair values considering the resistor power capacities and ensuring sufficient current flow for LEDs to light up. Specifically, the 820+20 ohm resistor pairs at the 19V DC line were replaced by 2x 470 ohm, 0.2W resistors for 20mA current flow, and the 680+47 resistor pair at the battery input

terminals were replaced by 2x 390 ohm, 0.2W resistors for a minimum of 19mA current flow on 14.8V battery voltage.

Finally, Clare Cui and I sourced appropriate electronic components from Digikey including DC barrel jacks required for connecting the PDS output to our Gigabyte Brix [10] computer.

The final PDS layout is shown in Fig 5 (without the resistor value changes).

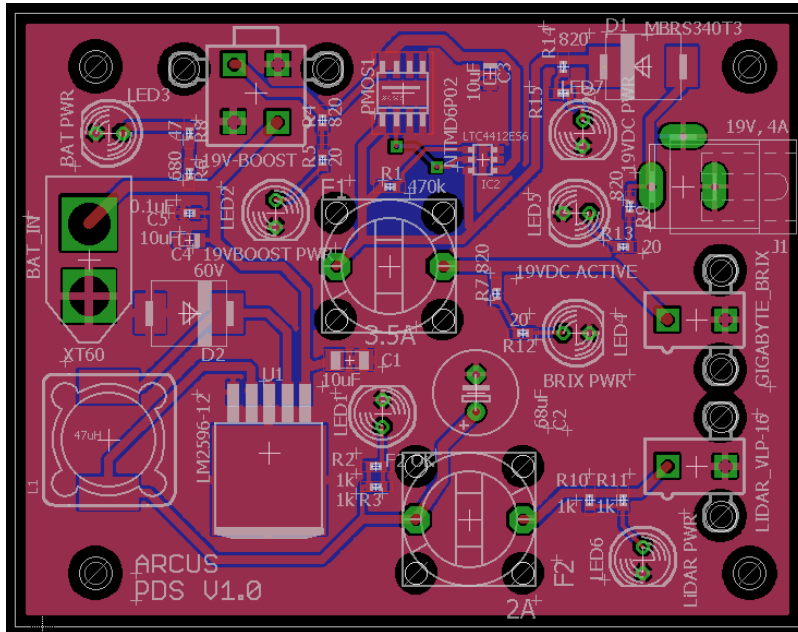


Fig 5: Power Distribution Board Layout [by Logan Wan and Maitreya Naik]

#### 1.4. Custom Cables

Since the IMU VN-100 currently has a 3' long 10-pin Harwin M80 to USB-to-TTL cable, the appropriate connectors were ordered for a custom cable of 20cm. This is because contacting Vectornav, we were given a quote of \$100 for each custom size cable.

Similarly, since we are not sure if the USB end uses a TTL or RS232 communication (VN-100 has pins for both), we ordered a Harwin connector and USB-to-RS232 connector cable set too. Also, we ordered a spare USB-to-TTL connector cable set since we will be connecting the PixHawk's serial4/5 port to the Gigabyte Brix's USB port.

## 2. Challenges

Challenges were faced during camera intrinsic calibration. First, we were using 16-bit grayscale images as input to Kalibr which resulted in the software generating semantic errors. It was then found out that we need to provide 8-bit grayscale images for Kalibr to work. Following this, we were receiving calibration outputs that resulted in pixel errors ranging in 3px. This was because we were taking the rosbag image stream at 4Hz. This probably resulted in Kalibr not being able to calculate the movement of the pattern boards across sequential images. Hence, increasing the image stream frequency to 20Hz led to better calibration outputs. Finally, fixing the

exposure of the camera to a lower but workable value reduced motion blur in images resulting in more images being used by Kalibr in its calibration calculation.

The large error in the IMU-Camera calibration may be a result of insufficient IMU intrinsic calibration. Hence, an effort will be made to run the calibration with a new data stream. If this still does not work, an effort will be made to carry out intrinsic IMU calibration as described in [7], if time permits.

### 3. Teamwork

Clare Cui and Logan Wan successfully CADed the sensor mounts to hold the RGB mv Bluefox camera, the Vectornav VN-100 IMU, the Velodyne Puck LiDAR, and the Piksi GPS Radio. They also built the mount to hold the drone at 10' above the ground in the back of Logan's truck for data acquisition at LaFarge.

Angad Sidhu was responsible for assisting in the software errors coming up in the camera intrinsic calibration and for bringing up the software for the LiDAR-Camera Calibration. He was also integral in developing the launch files for the LaFarge Data Acquisition test

### 4. Plan

By the next Progress Review, we will strive to carry out LiDAR-Camera Extrinsic calibration and find a way to verify the performance of the calibration. I will be responsible for powering up the drone and performing the IMU's Hard and Soft Iron Calibration to account for the electromagnetic fields, followed by a camera-IMU extrinsic calibration.

A test flight will be carried out without the IMU mounted. Following which, the computer and LiDAR protection mount will be integrated on to the drone.

Ricky will continue to work on fixing the loop closures and getting familiarized with the full BLAM! [11] code package while writing a boot script for the on-board computer to start up as soon as power is connected.

If the Power Distribution board and components arrive, the components will be soldered and the board will be tested within 2 days.

## References

- [1] T. M. Inc, "What is Camera Calibration?," [Online]. Available: <https://in.mathworks.com/help/vision/ug/camera-calibration.html>.
- [2] O. S. R. Foundation, "camera\_calibration," [Online]. Available: [http://wiki.ros.org/camera\\_calibration](http://wiki.ros.org/camera_calibration).
- [3] T. M. Inc, "Camera Calibration App," [Online]. Available: <https://in.mathworks.com/videos/camera-calibration-with-matlab-81233.html>.

- [4] A. ETH Zurich, "Kalibr," [Online]. Available: <https://github.com/ethz-asl/kalibr>.
- [5] D. Scaramuzza, "OCamCalib," [Online]. Available: <https://sites.google.com/site/scarobotix/ocamcalib-toolbox>.
- [6] ethz-asl, "Kalibr YAML formats," [Online]. Available: <https://github.com/ethz-asl/kalibr/wiki/yaml-formats>.
- [7] A. ETHZ, "IMU Noise Model," [Online]. Available: <https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model>.
- [8] Vectornav, "Vectornav VN-100 Rugged Industrial IMU," [Online]. Available: <http://www.vectornav.com/products/vn100-rugged>.
- [9] KumarRobotics, "IMU VN-100 ROS Package," [Online]. Available: [https://github.com/KumarRobotics/imu\\_vn\\_100](https://github.com/KumarRobotics/imu_vn_100).
- [10] Gigabyte, "GB-BSi7-6500 (rev. 1.0)," [Online]. Available: <http://www.gigabyte.com/products/product-page.aspx?pid=5692#ov>.
- [11] E. Nelson, "BLAM!," [Online]. Available: <https://github.com/erik-nelson/blam>.