INDIVIDUAL LAB REPORT 2

Progress Review - 1

Shubham Garg

TEAM H (PHOENIX)

Teammates:

Akshit, Parv, Steve

February 21, 2019

Individual Contribution:

1. Understanding of Kaliber library developed for combined IMU-Camera calibration
2. Setting up Kaliber framework for computing Camera-PX4 transformation matrix
3. Integrating April-tag code detection in ROS motion server framework.
4. ZED sensor mount

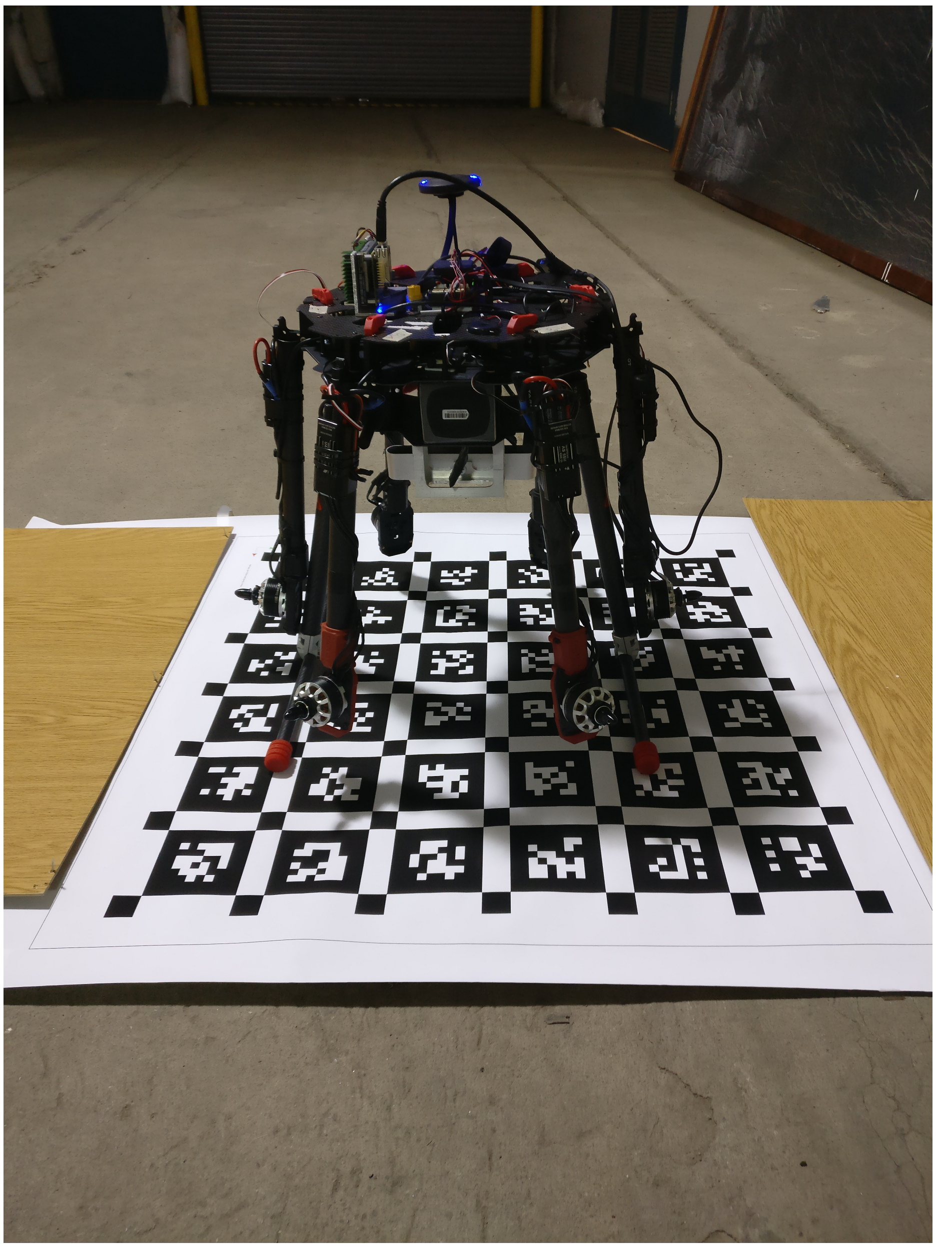
Understanding of Kaliber library developed for combined IMU-Camera calibration:

This library simultaneously computes the homogeneous transformation, **C** between the camera and the world map, and the homogeneous transformation, **I** between IMU and the world map. We used 16 April tags grid (map) as the landmarks. Above two transformations (C & I) can be used to compute the transformation between Camera and IMU ($I.C^{-1}$). Here the IMU is parameterized as 6x1 spline using the three degrees of freedom for translation & three degrees of freedom for orientation. Based on the raw acceleration & angular velocity readings from the Pixhawk's IMU, the acceleration and the angular velocity is computed in terms of **I**.
Above library needs an initial guess of C, homogeneous transformation which is first computed for each April tag using the PnP algorithm. Then the error between the predicted positions of the landmark based on the IMU reading and the observed position of the landmark using camera is minimized using Levenberg Marquardt optimizer. So, this library generates the following output: (i) the transformation between the camera and the IMU, and (ii) the offset between camera time and IMU time, d. (iii) the pose of the IMU w.r.t to world frame, I. (iv) Intrinsic camera calibration matrix, K for the ZED sensor camera.

Setting up Kaliber framework for computing Camera-PX4 transformation matrix:

1. Initially, we printed a sheet of 16 April tag grid and took the measurements of these tags for ground reference. This information is inserted in the YAML file mainly tag size is changed.
2. Further, all the axes of the UAV are translated in x,y & z direction & rotated in all the three directions for the proper calibration. So, to ensure we have sufficient data in all six degrees of freedom, we took the IMU sensor's measurement and camera frame for around 60 seconds.
3. IMU sensor data is subscribed from the MAVROS raw_sensor message & camera

frames are subscribed from the ZED sensor camera node.
4. This two data are combined in a ROSbag which is fed in the Kaliber library to generate the required matrices.
Further, we found that the Rotation matrix generated by this library is correct but the translation between the camera and IMU are computed incorrectly. So, we manually took those measurements and edited the transformation matrix.
Below is the IMU-Camera sensor calibration setup.

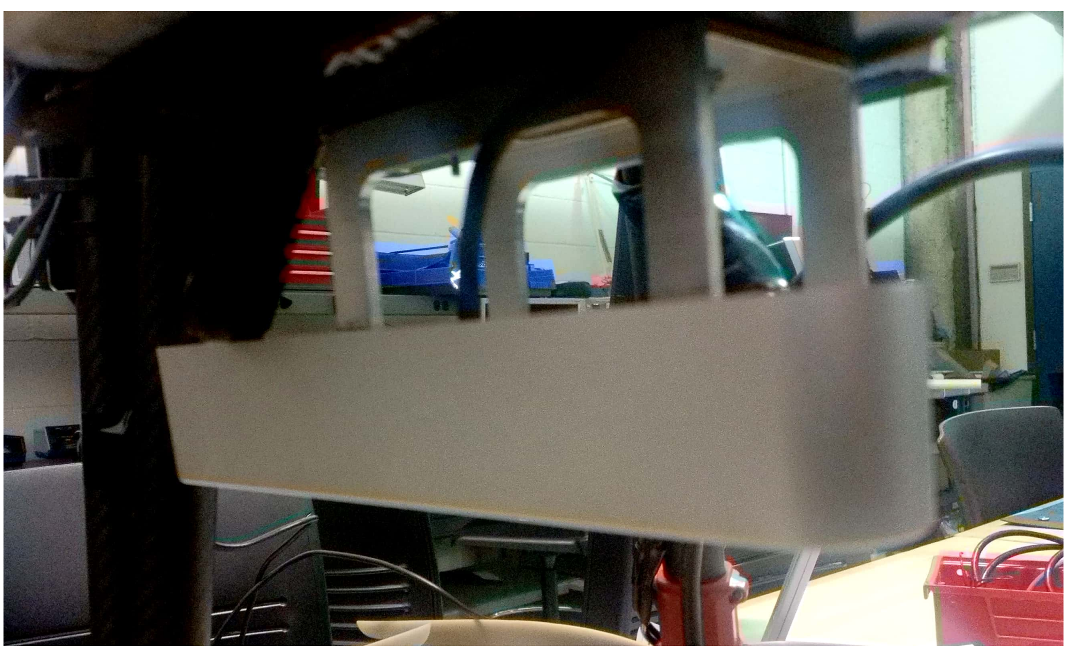


Integrating April-tag code detection in ROS motion server framework:

To improve the reliability of the system we are using ROS action server action which ensures that if any ROS node crashes there is a mechanism to recover from that state. Also, the user might want the ability to cancel the request during execution or get periodic feedback about how the request is progressing. So, my task was to integrate the April tag detection code in the ROS motion server framework.
Since the IMU's position in x & y drifts with time (due to integral errors), we need external sensor modality for accurate position estimation. So, the IMU's pose estimated by the ZED camera is published on MAVROS vision_pose message and PX4's onboard EKF2 will fuse this data with on board IMU data.

ZED sensor mount:

For testing position hold and UAV localization (using SLAM), we are using ZED camera sensor. And this week we mounted and tested ZED camera sensor on board. Since the 3D printer was temporarily unavailable/not-working last week, we had to design the mount for our ZED camera sensor using box aluminum tube. We designed the ZED camera sensor mount such that it can be attached at the bottom of the UAV platform and can also be attached in the front. The weight of the aluminum mount was our primary concern. Tim helped us in removing the excess part of the metal sheet from all the six faces of the mount which reduced mount's weight significantly. So, below is the final mount for the ZED camera.



Challenges:

Understanding the Kaliber library itself was challenging as it involves complex measurement & process model. But it was important for us in debugging and validating the results generated by the library. It took us around 5 to 7 iteration for understanding the proper input format and requirement of the library.

Lessons Learnt:
1. Initially, we were generating ROS bag from incorrect mavros message (local_pose estimated by the IMU) and later realized after going through the library that it takes raw IMU data as input.
2. During testing, we also found that we need to slowly orient the UAV so that a sufficient amount of data is collected in a single position.
3. Also, we had to ensure that most of the camera frame has all the April tags in it.
4. While playing the recorded ROS bag we found that most of the camera frames have shadows in it and we pre-processed all the camera frames before packing it in ROS bag.
Since the translation matrix from the Kaliber library was incorrect, instead of debugging further we manually computed the translation between camera and IMU. We also faced difficulty in computing the Rotation matrix from the Quaternion matrix published by the ZED camera sensor ROS node. There are not many existing python libraries which can be easily integrated. So, we wrote our own function for this conversion.

Teamwork

Akshit/ Shubham: Integrating Alt hold code in ROS motion server and test autonomous Takeoff & Land.
Parv/ Steve: Setting up onboard IMU (calibration, Testing) and commanding husky with onboard IMU for accurate pose estimation.

Future plans

I will be working with Parv on setting up ORB-SLAM2 on our platform. We will try to set it up such that it works for both UAVs and AGVs. Since ORB-SLAM doesn't fuse odometry data for the localization, we will have to modify the ORB SLAM's EKF. For the next progress review, our plan is to have a handheld platform which generates the map of the environment and simultaneously localizes itself.
Further, we are going to test and mount FLIR thermal camera onboard for fire detection. First, we are planning to test basic segmentation method (classical techniques) for fire detection. If it doesn't perform reliably we will try deep learning based networks like YOLO for the fire detection.