

1 What I've done

- Got more data from quadcopter (at 1.5m height, 1.0m radius) (I think the realsense node wasn't publishing for my second batch of data from last week, I may have forgotten to plug the camera in. So I'm not using that one).
- Wrote literature review section of report.
- Added a bit more to the Background Info section of the report on photometric error. (Although it's not finished yet).
- Installed stuff so I can read all of the messages from the rosbags (on my virtual machine).
- Investigated the fact I was getting a different amount of RGB and depth images from the rosbags.
- Wrote script to extract ground truth data
- Aligned ground truth data with images (for second experiment) by finding the point where the quadcopter started moving upwards. Pretty sure it's at least a few frames off though. (See Figure 1).
- Wrote script to extract control and odometry data
- Updated the image to camera frame coordinate transform with camera parameters and a (rough) scaling factor found by seeing the scale of the created point clouds.
- Implemented RANSAC for kabsch, started playing around with thresholds to try get something with decent results, doing ok but not great.

2 Parts of report to look at

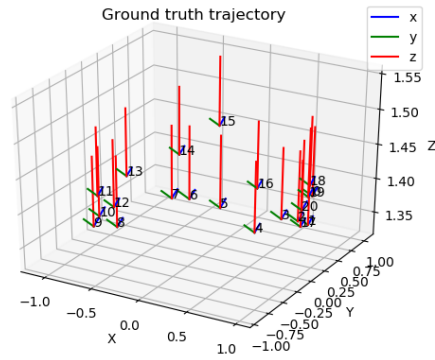
- Literature review (section 3, page 3)
- Background information (section 4, page 9) if you did not look at it from last week. New subsection added 4.6. Registration Methods: Photometric error, although it's not completely finished yet.

3 Questions

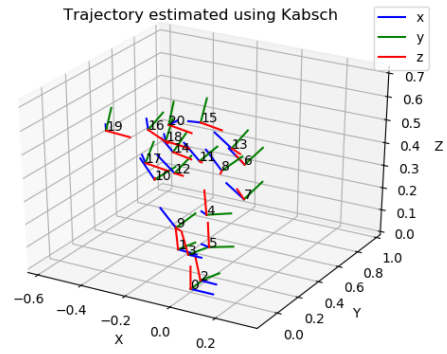
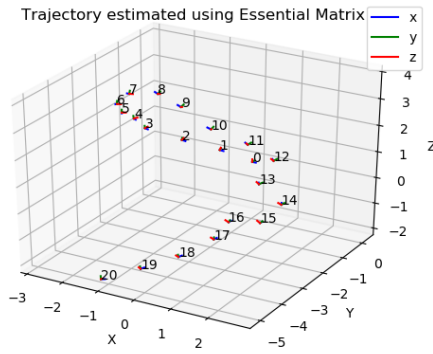
- How do I pick good parameters/thresholds for RANSAC?

4 Comments

- The control and odometry data is not time-stamped (all timestamps are 0), it does have a sequence number but I don't think I can use that to align it. I guess I could try to use the rosbag clock somehow? I should probably talk to Jean Luc and try get a version that has a working timestamp. Although I think the quad might have an older version of the code on it, as the version on git didn't work to play back the messages (Jean-Luc installed it on the quad).
- I got my callbacks to just print out the number of times they'd been called and found that the RGB and depth callback were both being called roughly the same amount (around 2000). When I got them to save the images, the depth one saved about that amount but the RGB one only saved about 500. (This is with new data, not the rosbag I was using last week). I ran them using ApproximateTimeSynchronizer with 0.1 tolerance (TimeSynchronizer didn't work), and got about 500 images for each. Part of the issue may just be that I'm running this on a virtual machine, I could try it on the GPU machine?
- Picking which frames to try register is definitely an important part of this (small enough for good matches, big enough not to run into numerics problems). I'm currently starting from a bit after the quad begins to move (which I estimated from looking at the images) and only doing every 8th frame. Although the amount of motion isn't consistent between frames (this could be partially due to not getting all of the images out of the rosbag as well as the motion of the quad not being smooth).
- It can also be hard to choose a distance threshold for kabsch when the level of motion changes a lot. For now I'm running kabsch again with a higher threshold if I don't get enough points with the lower threshold. It seems to work ok but is very slow.
- I'm getting very inconsistent results for Kabsch, probably need to tune RANSAC better. Also still has some outliers present (see Figure 2).

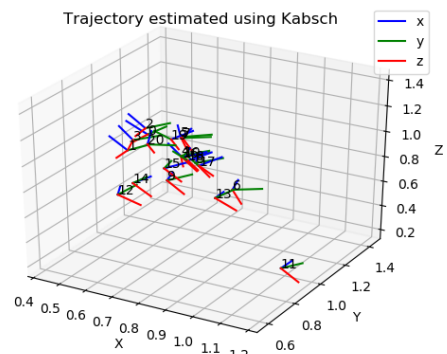
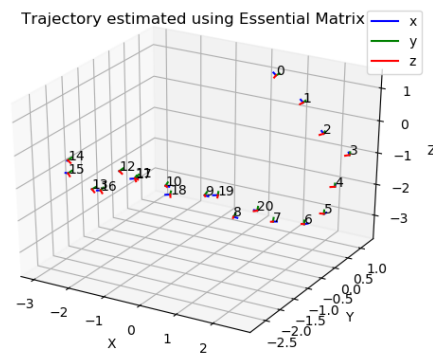


(a) Ground truth from Vicon



(b) Estimated trajectory for Essential Matrix method, start at origin

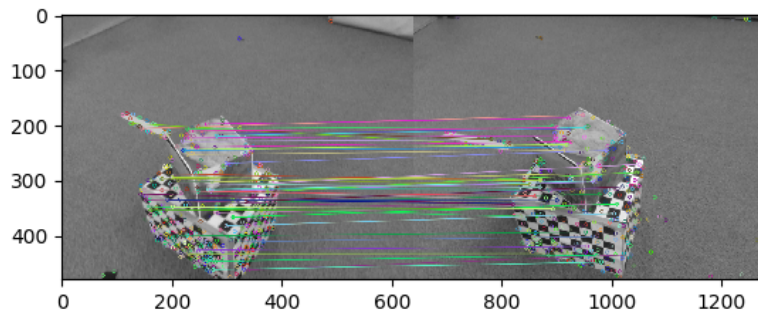
(c) Estimated trajectory for Kabsch method, start at origin



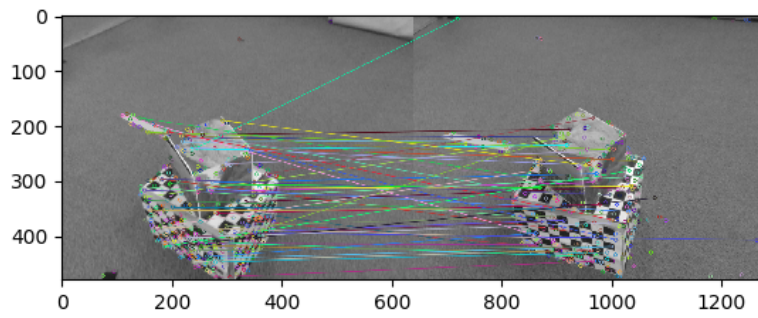
(d) Estimated trajectory for Essential Matrix method, start at ground truth

(e) Estimated trajectory for Kabsch method, start at ground truth

Figure 1: Trajectory visualizations for second quad dataset



(a) Points used for Essential Matrix



(b) Points used for Kabsch

Figure 2: Matches after RANSAC (inliers) between two frames