CMSC828T (P2Ph1): Search and Rescue

Rishabh Agarwal[1]

Abstract— **This paper describes how the project2 phase1 of CMSC828T class was completed. The paper talks about the structure followed for creating the factor graph used in the SLAMusingGTSAM.m and LocalizationUsingiSAM.m. The factor graph was inspired by the work of [1] about visual inertial odometry benchmark**

I. INTRODUCTION

The first code SLAMusingGTSAM.m takes in the input AprilTag corners coordinates in the camera frame. Based on the structure for motion algorithm I analyzed the change in the positions of the apriltag coordinates in each frame and generated a homography to calculate the image coordinates in global frame. In LocalizationUsingiSAM, I read the located landmarks and read the new set of data to calculate the quadrotor position in global frame. The first part uses GTSAM whereas the second part uses iSAM to iterate each frame one by one.

Section 2 talks about the working on SLAMusingGTSAM in more detail with the result plots, section 3 tries to explain the working of LocalizationUsingiSAM. Followed by section 4 talking about the issues faced and conclusion.

II. SLAMusingGTSAM

A. Factor Graph

So for the factor graph the first part was creating the factors of landmarks and quadrotor positions. For the landmarks I saved the bottom left as the first coordinates followed by anticlockwise motion. After creating the factors the relation between each frame and landmark coordinate was stored as a *GenericProjectionFactor* measurement. After the measurements between frames and coordinates I added the PriorFactors which sort of gives us the initial conditions for our factor graph. I assigned the tag with id 10 as the origin tag with left bottom corner as (0,0,0). Also, a random camera frame position was assigned to the first camera frame which could be later seen as the distinct point in the fig. 1. Then, I added the constraints between the corners of all the tags which leaded to 6 constraints for each tag, square size with length of tagsize and diagonal between two pair of coordinates. For the consecutive frames, a constraint of identity matrix was added to make sure no sudden change in the quadrotor position and orientation. after adding, all these constraints, measurements and factors I finally initialized the graph with initial estimates of the location calculated from the homography part.

*This work was completed with the help of Teaching Assistants of CMSC828T

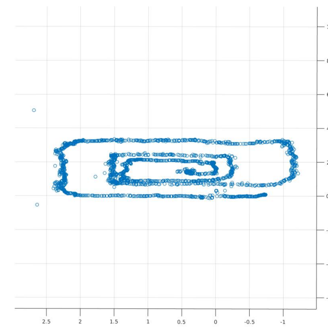

Fig. 1. Camera coordinates Top View

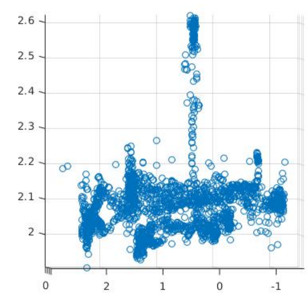

Fig. 2. Camera coordinates Side View

B. Homography

For the homography, I assumed that I know the position of tag 10 in the global frame and based on the comparison between the observed coordinates and the camera instrinsics, I calculated the rest of the coordinates for each frame. These coordinates served as my initial estimates for all the landmarks factors. Based on all these factors and data and implementing LevenbergMarquardtOptimizer for the factor graph I received the results as shown in fig.[1-3].

III. LocalizationUsingiSAM2

For the iSAM, I took the output LandMarksComputed from SLAMusingGTSAM. This input serves as my ground truth for the isam factor graph. I started using the same concept of adding factors in the beginning but this time also stored the ground truth for the comparison between the data read from the camera frame. Next step was odometry which was the pose between the consecutive camera frames.

A. Factor Graphs

For initializing the isam, we have to set the options parameters which was observed by reading the isam2 files

Fig. 3. Landmarks coordinates Side View

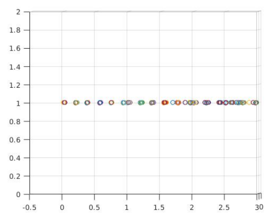

Fig. 4. Landmarks coordinates Side View

from +gtsam folder. But after setting the parameters I started giving the same genericprojection between each frame and landmarks but this time only between two frames at a time and updating the isam for each iterator as the isam operates iteratively and processing frame by frame provides a optimized output. .

IV. Conclusion

So in conclusion, the mapping part worked perfectly fine using the algorithm as mentioned above but when working on iSAM I still faced issues possibly because of the less understanding on the operation of iSAM in general. I tried following the GTSAM pattern but that didn't exactly seemed to work in this case.

References

[1] Pfrommer, B., Sanket, N., Daniilidis, K., and Cleveland, J. (2017). PennCOSYVIO: A challenging Visual Inertial Odometry benchmark. 2017 IEEE International Conference on Robotics and Automation (ICRA). doi:10.1109 icra.2017.7989443

[2] https://www.cc.gatech.edu/dellaert/FrankDellaert/FrankDellaert/Entries/2011/10/2iSAM2JournalPaper.html