CMSC 828T
Vision, Planning and Control of Aerial Robotics
P2Ph1 - Search and Rescue

Venkatraman Narayanan

I. Introduction

This project is aimed at implementing a vision-based Simultaneous Localization and Mapping (SLAM) to genreate a 3D map of the environment using a GTSAM and iSAM packages from GTSAM matlab toolbox. The premise of the project is to find the survivors in a power plant that has been destroyed by a recent earthquake. Due to the hazardous materials, drones (quadcopters) will be used for this mission. There are two parts to the mission. First, building the map of the current condition of the plant using a light weight drone with nothing but a single camera and an IMU. Then use a second drone is capable of carrying other needed supplies for the rescue mission. The second drone will use the map from the first drone and go to the rescue target location with the supplies.

II. The Task

The project has been divided into two modes, first being the SLAM mode and second, the Localization mode. In the SLAM mode, the map of the world is constructed (using GTSAM) given the April-Tag inputs by optimizing the factor graph, while simultaneously localizing the drone in the map. In Localization mode, the using the constructed map and based on observations obtained, the drone is localized in the map using iSAM2 part of GTSAM package.

III. Data Dictionary

The data sources include a $CalibParams.mat$ and 6 sequences which contain the camera images, pixel co-ordinates of corners of tags in the frame, etc.

a) : Camera Intrinsics and Extrinsics are given specifically in CalibParams.mat and has the following parameters:

- K has the camera intrinsics
- TagSize is in size of each AprilTag in meters.
- qIMUToC has the quaternion to transform from IMU to Camera frame (QuaternionW, QuaternionX, QuaternionY, QuaternionZ).
- TIMUToC has the translation to transform from IMU to Camera frame (TransX, TransY, TransZ).

b) : the data sources (in format DataSEQUENCE-TYPE.mat) will contain

- DetAll is a cell array with AprilTag detections per frame.
- IMU has 6 DOF odometry values from IMU sensor
- LeftImgs is a cell array where each cell is a Image.
- TLeftImgs is the Timestamps for LeftImgs.

IV. Background

The SLAM exercise is carried out by using the pixel co-ordinates of AprilTags (image 1) from various poses. The corners of the April-Tags are identified in the pixel co-ordinates, which are later used to estimate the 3D world co-ordinates of the tags.

a) : With the intrinsic calibration parameters, the pixel coordinates of the corner points, and the 3D positions of the tag corners known, the extrinsic calibration can then be determined by minimizing the reprojection error over all camera and rig poses. A nonlinear factor graph optimizer based on GTSAM is used to calibrate the 3D points and poses. For illustration, Figure 2 shows a subgraph of the full factor graph used for extrinsic calibration. Circles represent hidden variables to be optimized, solid squares represent factors connecting one or more variables, and free labels next to factors denote the measured data used.

V. Algorithm

The algorithm for estimating 3D world co-ordinates of camera poses and AprilTags using GTSAM involves 3 steps:

1) Initialisation
   For GTSAM packages to optimize the world co-ordinates properly, the world co-ordinated are to be initialized properly. Knowing that bottom-left corner of tag 10 will be origin in world co-ordinates, the poses of the camera frames consisting of tag 10 can be computed with RPnP approach.

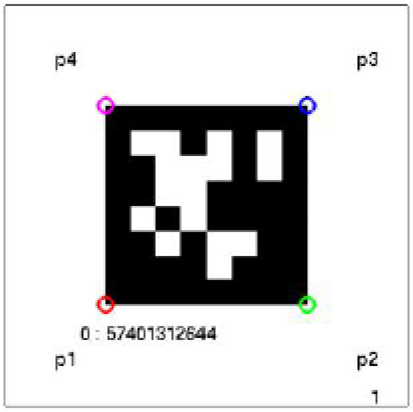

Fig. 1. AprilTag

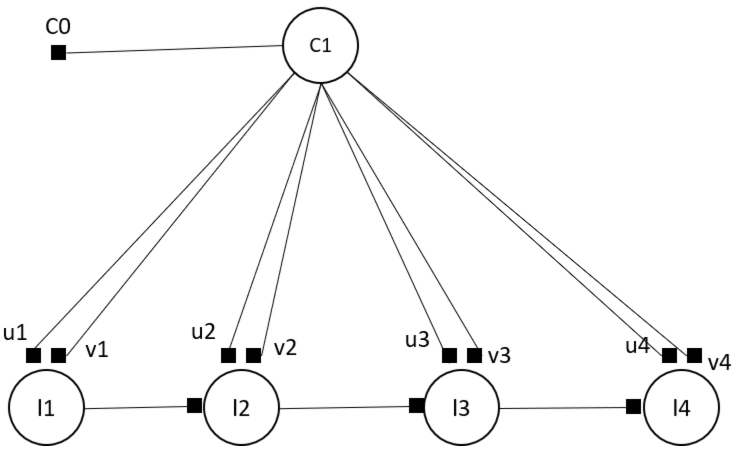

Fig. 2. Part of the Factor graph used for optimizing the 3D points and poses

2) Building the Factor Map
   The factor graph is then built using GTSAM packages for MATLAB, with the initial estimates, measurements in pixel co-ordinates and noise models.
3) Optimising the factor graph
   The factor graph, thus created, is optimized using Levenberg-Marquadt non-linear optimizer.

The iSAM part of the project too follows a similar approach. However, the factor graph are created and optimized in batches (usually 1 frame at a time), as opposed to all the frames at once.

VI. RESULTS

A. DataMapping

GTSAM

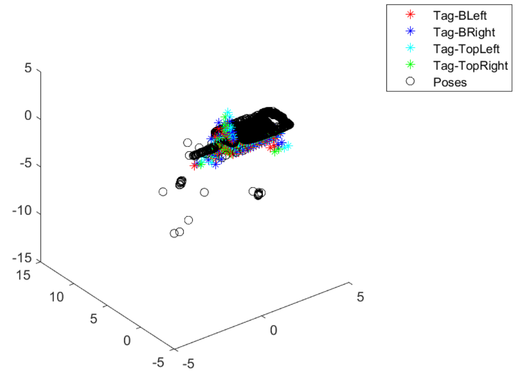

Fig. 3. DataMapping: X,Y,Z plot of poses and Landmarks from GTSAM

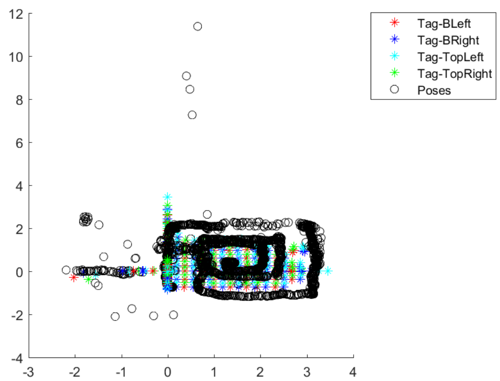

Fig. 4. DataMapping: X,Y plot of poses and Landmarks from GTSAM

B. DataSquare

GTSAM

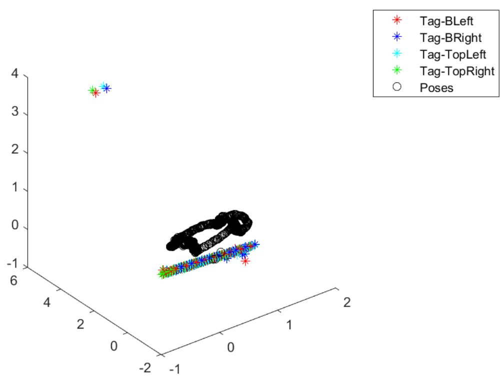

Fig. 5. DataSquare: X,Y,Z plot of poses and Landmarks from GTSAM

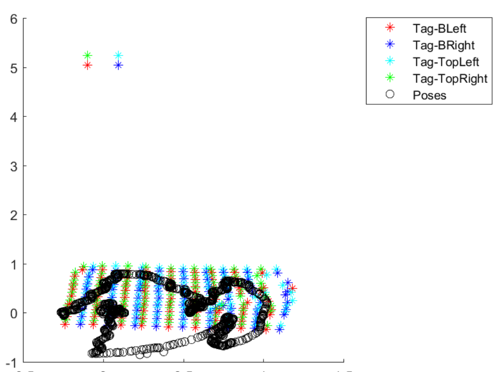

Fig. 6. DataSquare: X,Y plot of poses and Landmarks from GTSAM

iSAM2

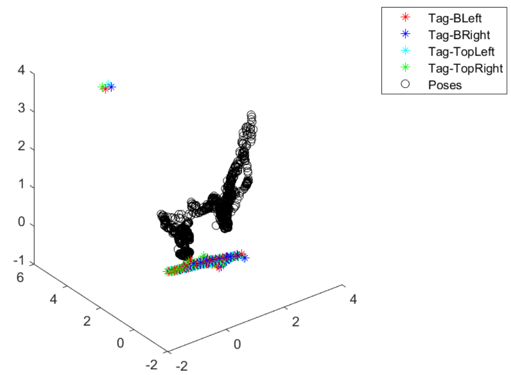

Fig. 7. DataSquare: X,Y,Z plot of poses and Landmarks from iSAM2

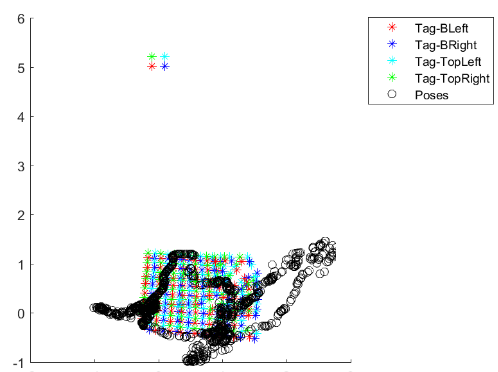

Fig. 8. DataSquare: X,Y plot of poses and Landmarks from iSAM2

C. DataMountain

GTSAM

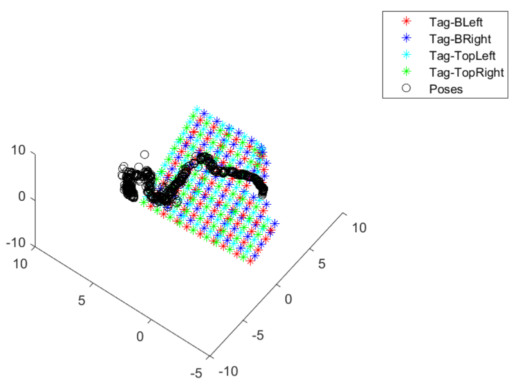

Fig. 9. DataMountain: X,Y,Z plot of poses and Landmarks from GTSAM

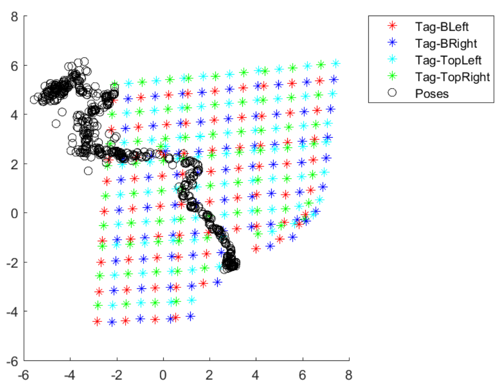

Fig. 10. DataMountain: X,Y plot of poses and Landmarks from GTSAM

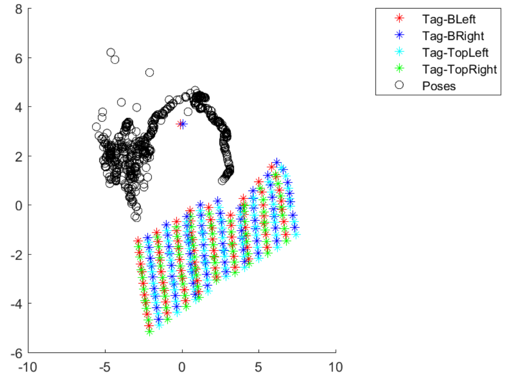

Fig. 11. DataMountain: X,Z plot of poses and Landmarks from GTSAM

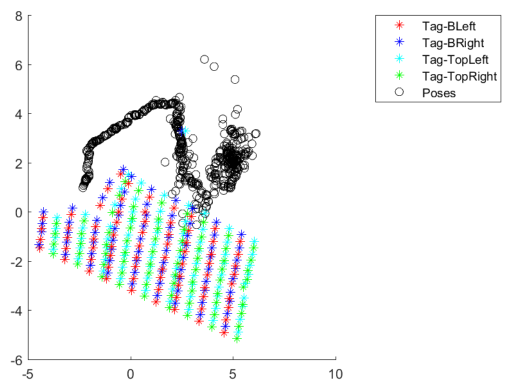

Fig. 12. DataMountain: Y, Z plot of poses and Landmarks from GTSAM

iSAM2

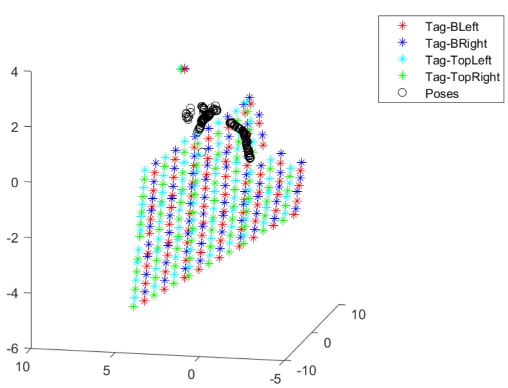

Fig. 13. DataMountain: X,Y,Z plot of poses and Landmarks from iSAM2

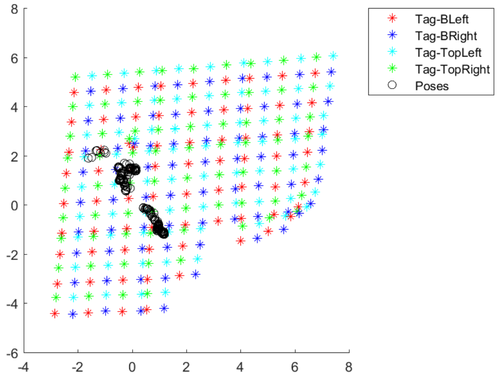

Fig. 14. DataMountain: X,Y plot of poses and Landmarks from iSAM2

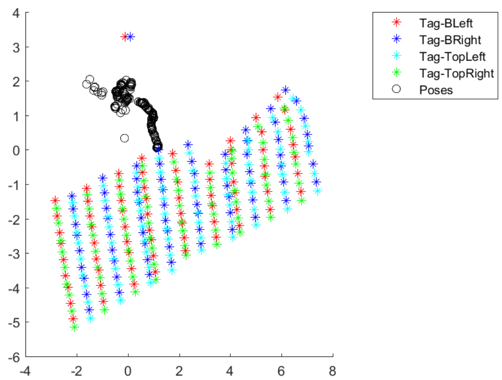

Fig. 15. DataMountain: X,Z plot of poses and Landmarks from iSAM2

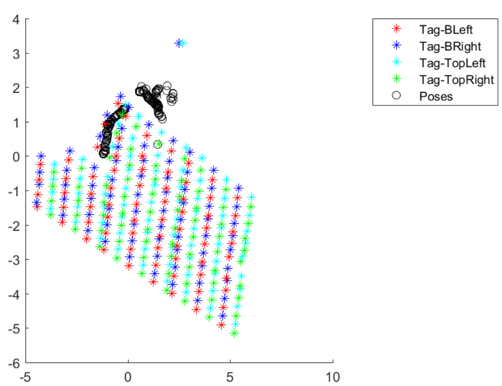

Fig. 16. DataMountain: Y, Z plot of poses and Landmarks from iSAM2

D. DataStraightLine

GTSAM

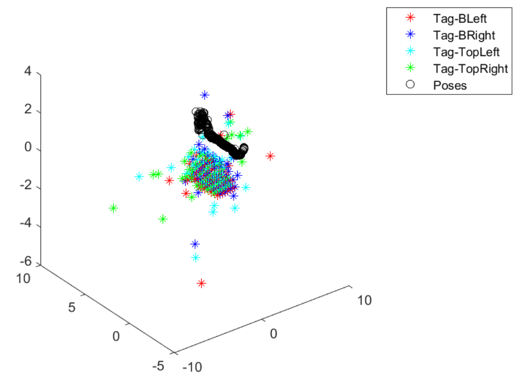

Fig. 17. DataStraightLine: X,Y,Z plot of poses and Landmarks from GTSAM

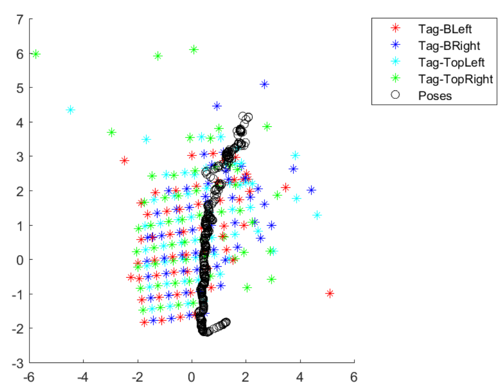

Fig. 18. DataStraightLine: X,Y plot of poses and Landmarks from GTSAM

iSAM2

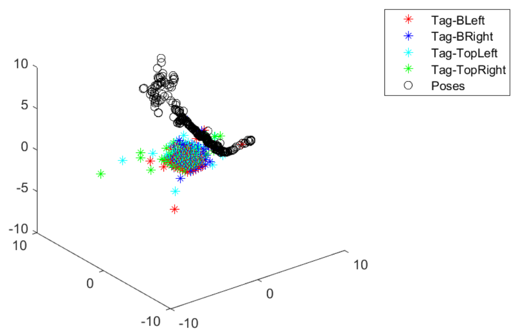

Fig. 19. DataStraightLine: X,Y,Z plot of poses and Landmarks from iSAM2

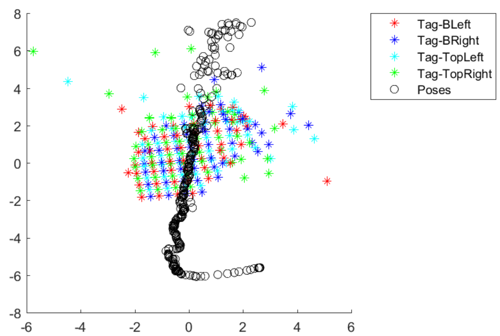

Fig. 20. DataStraightLine: X,Y plot of poses and Landmarks from iSAM2

E. DataSlowCircle

GTSAM

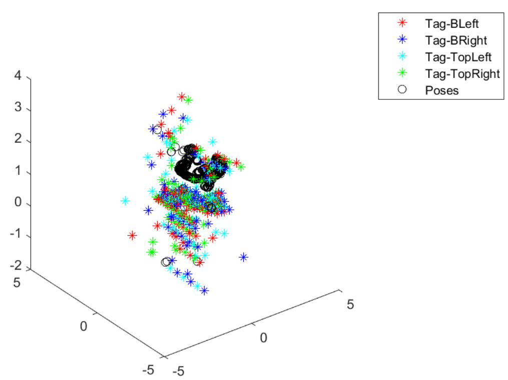

Fig. 21. DataSlowCircle: X,Y,Z plot of poses and Landmarks from GTSAM

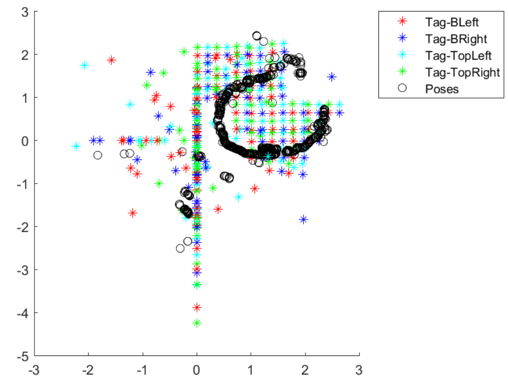

Fig. 22. DataSlowCircle: X,Y plot of poses and Landmarks from GTSAM

iSAM2

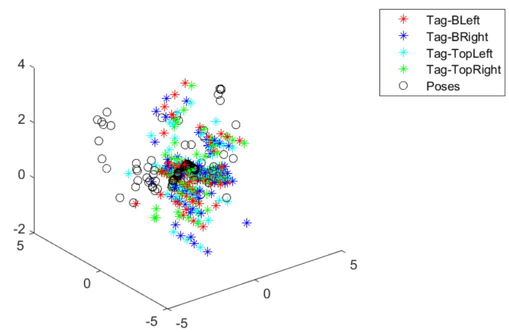

Fig. 23. DataSlowCircle: X,Y,Z plot of poses and Landmarks from iSAM2

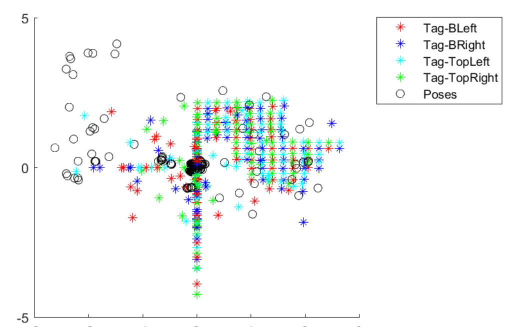

Fig. 24. DataSlowCircle: X,Y plot of poses and Landmarks from iSAM2

F. DataFastCircle

GTSAM

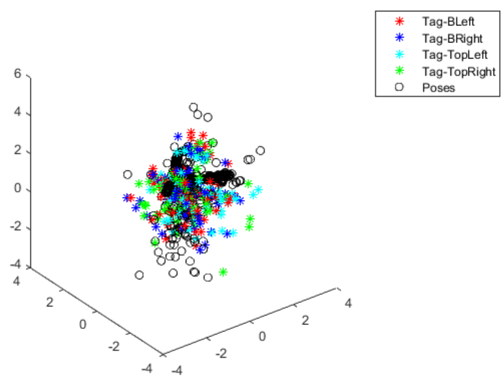

Fig. 25. DataFastCircle: X,Y,Z plot of poses and Landmarks from GTSAM

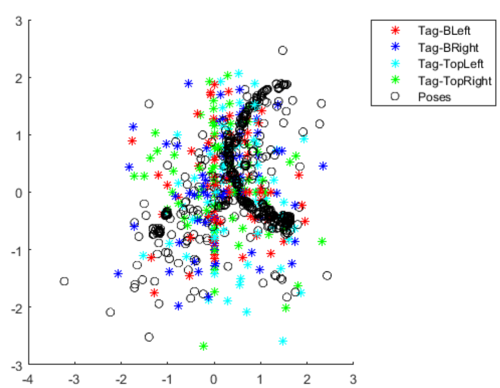

Fig. 26. DataFastCircle: X,Y plot of poses and Landmarks from GTSAM

iSAM2

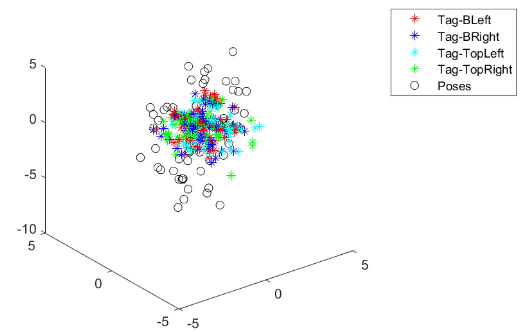

Fig. 27. DataFastCircle: X,Y,Z plot of poses and Landmarks from iSAM2

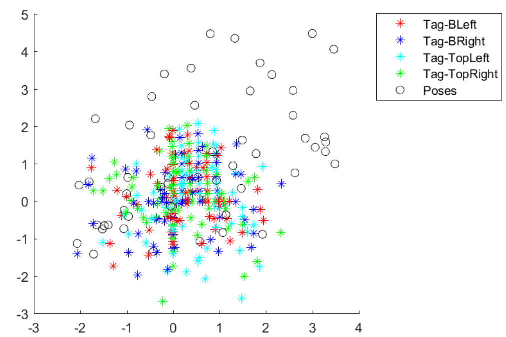

Fig. 28. DataFastCircle: X,Y plot of poses and Landmarks from iSAM2

REFERENCES

[1] Bernd Pfrommer, Nitin Sanket, Kostas Daniilidis, and Jonas Cleveland. Penncosyvio: A challenging visual inertial odometry benchmark. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 38473854, May 2017.

[2] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman lter based visual-inertial odometry using direct photometric feedback. The International Journal of Robotics Research, 36(10):10531072, 2017.