

# New LiDAR Based SLAM Systems for Real-Time Campus Tour Robot Navigation

Subhobrata Chakraborty  
Department of Computer Science  
California State University, Northridge  
Northridge, CA, USA  
subhobrata.chakraborty.026@my.csun.edu

Abhishek Verma  
Department of Computer Science  
California State University, Northridge  
Northridge, CA, USA  
abhishek.verma@csun.edu

Amiel Hartman  
Department of Mechanical Engineering  
California State University, Northridge  
Northridge, CA, USA  
amiel.hartman@csun.edu

**Abstract**— LiDAR based SLAM systems are often used because of their high precision and accuracy in 3D environments. LiDAR emits lasers to create detailed point clouds of the environment which helps the robot localize within the environment precisely. It is independent of the ambient light and weather conditions which makes it a perfect choice over visual based SLAM systems. We introduce a challenging campus tour robot navigation dataset that encompasses complex indoor and outdoor scenarios. Data was collected using custom sensor setup on two mobile robots SPOT and Husky. We propose and compare two state-of-the-art real-time LiDAR inertial odometry based SLAM system and only LiDAR based odometry system with the goal to deploy the system on the SPOT and Husky robots for the novel application of giving tours around CSUN university campus via a mobile robot. Experiment results prove that computationally the LiDAR based system took less time than the LiDAR inertial based system, but the LiDAR inertial system was more accurate in terms of the state estimation of the robot.

**Keywords**—LiDAR inertial odometry, LiDAR mapping, point cloud, iterative closest point, Faster-LIO, F-LOAM, SLAM.

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) serves as an integral requirement for mobile robotic systems, which also include unmanned aerial vehicles (UAVs). Common visual odometry methods like stereo visual odometry [1] and monocular visual odometry [2, 3] are often deployed in mobile robotic systems because it is computationally less expensive and cost-effective. Visual odometry provides rich RGB and visual information from the environment but it is challenging to measure the depth of a scene directly. It has also been established to utilize substantial computational resources for the reconstruction of the 3D environment for planning of trajectory. In any kind of odometry system that relies solely on visual input, the system tends to be sensitive towards varying lighting conditions. Light Detection and Ranging (LiDAR) sensors can address these challenges because they are highly tolerant to different surrounding lighting conditions.

Recent advancements in LiDAR technology, particularly solid-state LiDARs that are based on micro-electro-mechanical-system (MEMS) [4] scanning and rotating prisms [5], have emerged as cost-effective, lightweight

alternatives. These LiDARs can produce three dimensional measurements as well as active measurements with an accuracy spanning over a much longer range. This feature is significantly useful for smaller scale UAVs, especially in industrial applications such as aerial mapping or post-disaster search and inspection.

When compared to visual SLAM systems, LiDAR SLAM offers higher accuracy in terms of estimating pose and robustness against variations in the environment, including changes in illumination and weather. Since LiDAR has proved to be very useful, it has led to widespread adoption in different applications in the robotics industry like autonomous driving [6, 7], inspection with drones [8], and manipulation of packages in warehouses [9].

Prior research on LiDAR SLAM [10] has established the fact that it has been performing well on publicly available dataset evaluations. However, practical applications come across certain challenges in terms of robustness while transitioning from an indoor to outdoor or static to dynamic environments. The fact that these systems need to be computationally efficient for easier deployment on resource constrained platforms like UAVs should also be taken into consideration. The most common approach to estimating the transformation of depth data between two scans involves Iterative Closest Point (ICP) method. ICP's prove to be computationally inefficient when dealing with a large number of points. LiDAR Odometry and Mapping (LOAM) is another approach that matches features more efficiently but due to iterative calculation it is computationally expensive.

Our research focuses on testing the Faster LiDAR Inertial Odometry (Faster-LIO) [6, 7] and the Faster LiDAR Odometry and Mapping (F-LOAM) [11] algorithms on real-time dataset. The subsequent sections of this paper are structured as: Section II focuses on the prior relevant works related to our research while section III provides details of our custom dataset. Section IV covers details of the two proposed systems on our custom dataset. Section V presents the findings of the experiments while section VI entails conclusion and future work.

## II. RELATED WORK

Besl et al. [12] introduced an iterative closest point (ICP) algorithm for registration of LiDAR scans that serves as the



(a) Sensor setup on Husky robot.



(b) Sensor setup on SPOT robot.

Fig. 1. Sensor configuration of the robots used for collecting the Campus Tour Robot Navigation (CTRNav) dataset.

foundation for dealing with LiDAR based odometry systems, and it is very effective in handling 3D scans. While handling sparse point clouds from LiDAR measurements it is very difficult to achieve exact point matching. To address this issue, Segal et al. [13] proposed a more generalized approach for the ICP method, which leverages the distance between the points and the planes. It was further improved by Zhang et al. [14] where the previous ICP method was integrated with the distance between the points and edges, subsequently leading to the development of the LiDAR odometry and mapping (LOAM) algorithm. Various variations of the LOAM algorithm have been introduced since then which include Lego-LOAM [15] and LOAM-Livox [9]. Lego-LOAM and LOAM-Livox are effective in structured environments with LiDARs featuring large field of view, but they prove inaccurate in featureless environments or where there is a smaller field of view for the LiDARs.

Incorporation or fusion of inertial measurement units (IMU) to address the challenges of LiDAR degeneration is a common and effective method. The systems where the LiDAR data and the IMU data are processed separately are called loosely coupled LiDAR inertial odometry systems. Even though the LiDAR and IMU data are processed separately, they are fused eventually. The LOAM algorithm when integrated with the IMU data also takes into consideration the positional and orientational measurements from the IMU to account for the initial estimate of the registration of the LiDAR scan. Zhen et al. [16] worked on fusing the IMU data with the output of the Gaussian particle filter of the LiDAR data, eventually using the error state extended Kalman filter. Balazadegan et al. [17] enhanced LiDAR scan registration by incorporation of the gravity model of the IMU for estimating the ego motion involving 6-DOF (degrees of freedom). Zuo et al. [18] makes use of a multi-state constraint Kalman filter for integrating the output of the scan registration with visual and IMU data. A classical approach in the loosely coupled method involves acquiring pose data by new scan registration, which is followed by fusion of the IMU measurements. This reduces computational time by separating the scan registration and data fusion, but it does not consider the robot's velocity along with the new scan pose. In environments where the features are limited, there is a possibility that the scan registration will be poor in certain directions, which could impact the fusion in the later stages.

In the case of tightly coupled LiDAR inertial odometry methods, the raw feature points of the LiDAR and the IMU

data are fused together. Optimization based and particle-based methods are the two main approaches to the tightly couple LiDAR inertial odometry system. Geneva et al. [19] employs graph-based optimization method while pre-integrating constraints of IMU and the plane constraints from the LiDAR. LIOM [20] package utilizes a similar graph-based optimization method on edge and planar features.

### III. DATASET DESCRIPTION

We introduce in this paper the Campus Tour Robot Navigation (CTRNav) dataset, which is a custom dataset gathered at California State University, Northridge (CSUN). The objective of this dataset is to test mobile robots to perform a tour of the CSUN campus. We present the results of our experiments on the CTRNav dataset in Section V. The dataset encompasses diverse environment and covers the following scenarios.

#### A. Indoor Mapping

This subset of CTRNav was specifically curated for indoor mapping using the LiDAR system. It includes various elements such as the robotics lab, conference room, corridor, courtyard, and office spaces. It is about 12 minutes 51 seconds long and the mapping result is represented in Fig. 4 of Section IV.

#### B. Outdoor Mapping

The focus of this subset is to map the outdoor parking lot of the Reseda Annex building at CSUN. The scene elements include two buildings, two alleys and cars parked in the parking lot. It is about 12 minutes 10 seconds long and the mapping result is illustrated in Fig. 5 of Section IV.

#### C. Campus Mapping

Collected on the premises of Jacaranda Hall at CSUN, this subset originated from the HAAS lab, serving as the starting point for the robot's manual operation. The robot covered diverse areas, including corridors, classrooms, courtyard, and the food court. The data collection involved the use of LiDAR and IMU sensors. The dataset is about 1 hour 59 seconds long and the mapping result is illustrated in Fig. 6 of Section V. The HAAS lab is a machine shop on the university premises, which is a challenging warehouse industrial environment with industry grade equipment setup and provided rich features for mapping.

The CTRNav dataset was instrumental in evaluating and validating the performance of the proposed system pipelines in diverse environmental settings, encompassing both indoor and outdoor spaces at the CSUN campus. The dataset was carefully gathered employing both the Husky and the SPOT robots, and the corresponding sensor configuration as illustrated in Fig. 1. This comprehensive setup allowed for the acquisition of diverse data, ensuring a thorough evaluation of the proposed system pipelines across different robotic platforms and sensor arrangements.

### IV. RESEARCH METHODOLOGY

In this paper, we propose the Faster-LIO based system illustrated in Fig. 2 and F-LOAM based system in Fig. 3, for real-time mapping. Faster-LIO system pipeline is described

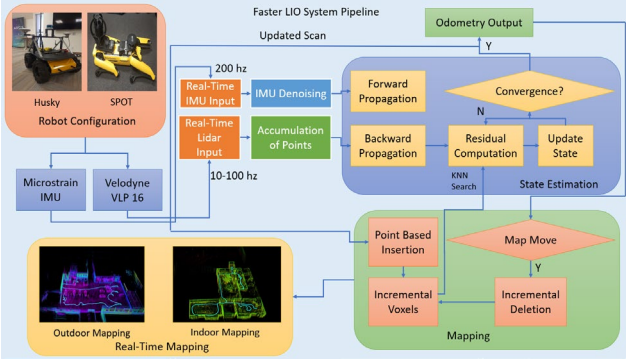


Fig. 2. Proposed Faster-LIO based system pipeline for real-time mapping.

in subsections A-D and F-LOAM system pipeline is described subsections E-G. The two systems were evaluated on the CTRNav dataset.

### A. iVox Data Structure for Faster-LIO

In the iVox system, the initial step is to store the point clouds within the sparse voxels, which are indexed and hashed into an unordered map. Unlike volumetric representations, iVox is based on the utilization of a hash map that is sparse and also capable of storing selected voxels containing one point at a minimum given that the point cloud is sparse in nature. Different algorithms for spatial hashing can be utilized to compute the index of the hash function. The different points contained inside each of the voxels are retained in the form of a vector or inside a structure called the Pseudo Hilbert Curve (PHC). Those type of retention methods are known as linear iVox and iVox-PHC, respectively. The complexity of the k-Nearest Neighbors (k-NN) scan inside every voxel is  $O(n)$  or  $O(k)$  when either linear iVox or iVox-PHC is being used. Here,  $n$  represents the quantity of points retained within each voxel, while  $k$  represents the discrete PHC curve order. During the incremental mapping in the LiDAR inertial odometry, certain measures are taken to avoid insertion of excessive number of points into the same voxel. Equation 1 is the hash function used for the Faster-LIO algorithm,

$$p = [p_a, p_b, p_c]^T, \quad v = \frac{1}{s} [p_a, p_b, p_c]^T, \quad id_v = \text{hash}(v) = (v_a n_a) \text{ xor } (v_b n_b) \text{ xor } (v_c n_c) \text{ mod } N, \quad (1)$$

where  $p_a, p_b, p_c$  determine the coordinates of  $p \in \mathbb{R}^3$ ,  $s$  denotes the voxel size,  $n_a, n_b, n_c$  are very large prime numbers, and  $N$  denotes the size of the hash map.

### B. kNN Search for Faster-LIO

The search for the k-NN algorithm is confined amid a predetermined scope and can be formulated in three consecutive steps. While considering a data structure for iVox  $V$  integrated with a point for query  $P$ , the process is going to involve: 1) Identifying the index of each voxel along with the voxels in close proximity specifically, 18, 26 or 6 voxels, denoted as  $S$ . 2) Iterating across each voxel scanning for  $K$  neighbors at maximum for each voxel. 3) Consolidating the scan outcomes and deciding on the  $K$  optimal neighbors. It is essential to highlight the fact that while step 2 can be independently parallelized in the case of

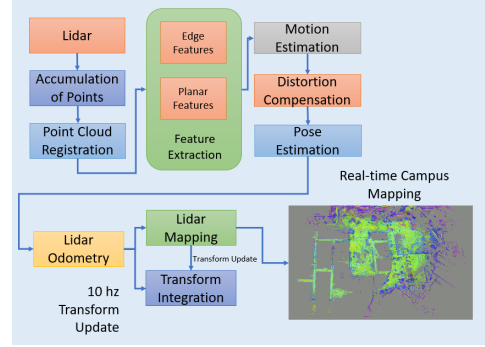


Fig. 3. Proposed F-LOAM based system pipeline for real-time mapping.

each voxel, the existing parallelization for the point clouds renders parallel searching across every voxel unnecessary. The k-NN scan in the case of iVox is characterized by simplicity and effectiveness, although it may not be as strict as tree-like algorithms. Nevertheless, it proves to be sufficient for applications in LIO.

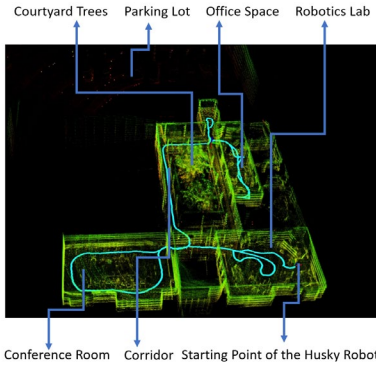
### C. Incremental Mapping for Faster-LIO

The incremental mapping process in iVox involves two main categories: incremental addition and deletion. The addition phase involves the straightforward insertion of new points and the creation of new voxels where necessary. To prevent an excessive accumulation of points within a single voxel, a VoxelGrid-like filter is employed, similar to the approach used in FastLIO2. Leveraging precomputed voxel indices of nearest neighbors, faster LIO refrains from insertion of the recent point while any of the neighboring points are in close proximity to the middle of the grid of the voxel. The pivotal variable for tuning the compromise between precision and efficiency in this filter is the leaf size. A larger leaf size, such as 0.5 m, where  $m$  is meters, generally strikes a balance that works well across various datasets, limiting insertions into a voxel without sacrificing k-NN accuracy.

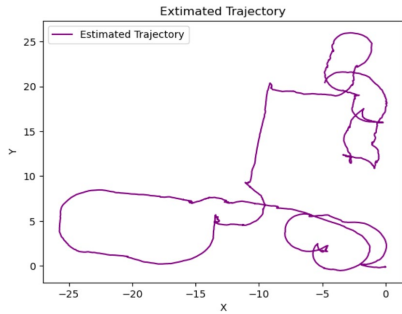
In contrast, incremental deletion differs from the framework of the k-d tree, as navigating through the entire hash map of the voxel proves to be computationally extravagant. Rather than proactively removing points beyond the present Field of View (FoV) within the spatial viewport, a Least Recently Used cache (LRU cache) approach is implemented to handle the management of the local map. Alongside the voxel hashmap, a queue of recently visited voxels is maintained with a predetermined maximum size. Should the number of voxels exceed the threshold limit, the outdated voxels in memory are eliminated. The addition and removal operations in a hash map exhibit  $O(1)$  complexity, thereby it is computationally inexpensive and suitable for real-time LiDAR inertial odometry algorithms.

### D. iVox PHC for Faster-LIO

iVox PHC serves as an alternative rendition of iVox where a modification is introduced by substituting the sequential arrangement layout within every voxel with a pseudo-Hilbert curve (PHC). In the LIO pipeline efforts are made to control the quantity of points within a voxel but the k-NN search performance decreases linearly when there is a



(a) Real-time indoor mapping of the robotics lab, conference room and the office space.



(b) Estimated trajectory of the mobile robot.

Fig. 4. Real-time indoor mapping and estimated trajectory using Faster-LIO on CTRNavi dataset.

steady rise in the total count of points. To tackle this problem, spatial filling curves such as PHC are utilized that offers mapping from a lower dimensional to a higher dimensional space while maintaining the proximity. This characteristic makes them suitable for approximating nearest neighbors within a particular voxel or a complete point cloud.

In this arrangement, a voxel is divided into  $2^{K^3}$  diminutive cubes with  $K$  denoting an adjustable PHC order. It is adjusted based on the physical size of the voxel. Every cube retains the central point of all encompassed points, and the central point undergoes modification when a newly inserted point falls within the cube's range.

#### E. Sensor Model and Feature Extraction for F-LOAM

A LiDAR scan that consists of a large quantity of points presents computational inefficiencies for methods like ICP.

Feature matching proves to be more reliable and efficient in practical scenarios. To enhance both efficiency and matching accuracy, surface and edge features are used to discard noisy or less relevant points. The 3D mechanical LiDAR derived point cloud exhibits scarcity in the vertical dimension and abundance in the horizontal dimension. Horizontal features, therefore become more discernible, minimizing the potential for erroneous feature detection along the horizontal plane. The F-LOAM algorithm focuses on the horizontal plane for each point cloud evaluating the local surface evenness. For flat surfaces the smoothness value is relatively less whereas for the corners and edges the smoothness value is large.

$$\sigma_k^{(m,n)} = \frac{1}{|S_k^{(m,n)}|} \sum_{\mathbf{p}_k^{(m,j)} \in S_k^{(m,n)}} (|\mathbf{p}_k^{(m,j)} - \mathbf{p}_k^{(m,n)}|), \quad (2)$$

The smoothness of the local surface is evaluated using equation 2, where  $S_k^{(m,n)}$  denotes the adjacent set of points for  $\mathbf{p}_k^{(m,n)}$  and  $|S_k^{(m,n)}|$  denotes the number of points for a localized point cloud.  $\mathbf{p}_k$  is the scan plane where feature points at the edge are determined with high  $\sigma$  but for points at the surface a lower  $\sigma$  is used.

#### F. Distortion Compensation and Motion Estimation for F-LOAM

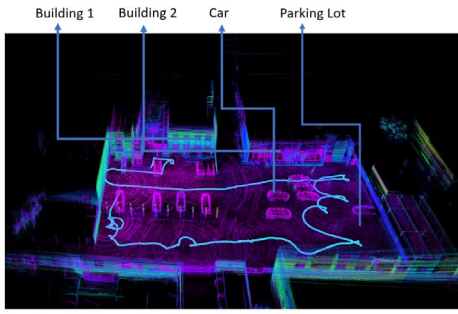
In prior studies like LOAM and LeGO-LOAM, distortion correction is commonly accomplished through scan-to-scan alignment, wherein the iterative estimation of the transformation between a couple of sequential laser scans is employed. This iterative computation for determining the transformation matrix proves to be computationally inefficient. F-LOAM uses a dual phase distortion alleviation method to mitigate the computational costs.

Since most of the current 3D LiDARs function at frequencies surpassing 10 Hz and the time interval between successive LiDAR scans is frequently minimal, F-LOAM leverages the assumption of consistent angular and linear velocity within these brief durations to anticipate motion and rectify distortion in the initial phase. Subsequently, in the phase that follows, distortion is recalculated following the pose estimation procedure, and the newly computed undistorted features are integrated into the final map. F-LOAM demonstrates that the dual phase distortion alleviation method achieves comparable localization accuracy while significantly reducing computational overhead.

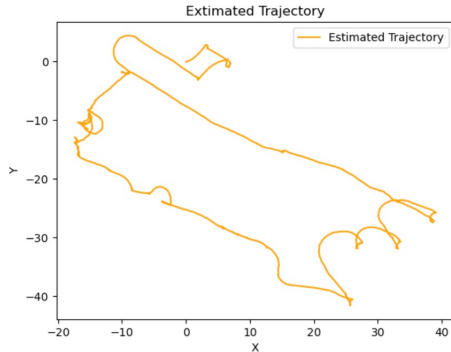
TABLE I. TIME COMPARISON IN MILLISECOND ACROSS INDOOR, OUTDOOR AND CAMPUS MAPPING USING FASTER-LIO ON CTRNAVI DATASET

	Indoor Mapping	Total Time for Indoor Mapping on 7643 seconds	Outdoor Mapping	Total Time for Outdoor Mapping on 7234 seconds	Campus Mapping	Total Time for Campus Mapping on 36041 seconds
Frames Per Second	585	NA	213	NA	417	NA
iVox Addition	0.007 ms	53.501 ms	0.048 ms	347.232 ms	0.120 ms	4324.92 ms
Incremental Mapping	0.036 ms	53.501 ms	0.194 ms	1403.396 ms	0.181 ms	6523.421 ms
Standard Preprocessing	0.351 ms	2682.693 ms	0.383 ms	2770.622 ms	0.387 ms	13947.867 ms
Undistort function in Point Cloud Library (PCL)	0.195 ms	1490.385 ms	0.386 ms	2792.324 ms	0.247 ms	8902.127 ms





(a) Real-time outdoor mapping of the parking lot.



(b) Estimated trajectory of the robot.

Fig. 5. Real-time outdoor mapping and estimated trajectory using Faster-LIO on CTRNavi dataset.

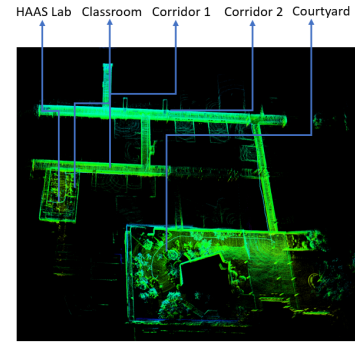
### G. Pose Estimation for F-LOAM

The pose estimation procedure involves the alignment of the undistorted edge characteristics and the planar attributes with the global attribute map. The global attribute map comprises separate representations for planar and edge characteristics which are revised and maintained independently. To improve the computational efficiency during the search process, the organization of planar and edge features is optimized through the utilization of 3D k-d trees. The determination of global lines and planes is achieved by the aggregation of adjacent points from the maps of edge and planar features.

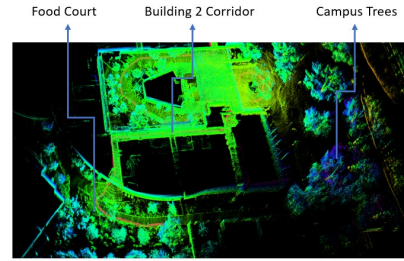
For every edge feature point, the covariance matrix pertaining to the surrounding points is derived from the global attribute map. The regions where the points are aligned in a linear fashion, the covariance matrix displays a singular eigenvalue notably greater than the rest. After acquiring the optimal edges and planes, the link between each feature point to the corresponding global edges or planes is established. This connection is leveraged to deduce the optimized pose between the global map and the present frame by reducing the separation between global planes or edges and feature points.

### H. Mapping for F-LOAM

The overall map comprises of a global planar map and a global edge map, and it undergoes modifications triggered by keyframes. Keyframes are selected when the translational shift surpasses a predefined limit or when a rotational change surpasses a predefined limit. This approach improves computational efficiency when compared to a frame-by-



(a) Campus mapping using Faster-LIO



(b) Food court and building 2 mapping.

Fig. 6. Real-time campus mapping using Faster-LIO based system on CTRNavi dataset.

frame update approach. Distortion correction is dependent on a model of consistent velocity model rather than iterative estimation of motion. This helps in reducing the computational costs but it is less precise than the iterative distortion alleviation utilized in LOAM. In the next stage, distortion is recalibrated depending on the optimization outcome. The newly computed undistorted edge characteristics and planar attributes are eventually refreshed to the global planar map and the global edge map, respectively. After each refresh, the map undergoes downsizing using a 3D voxel grid method to avert overflow of memory.

## V. EXPERIMENT SETUP, RESULTS AND DISCUSSION

Velodyne VLP-16 16 channel spinning LiDAR and MicroStrain IMU were used for the experiments. The LiDAR supports dual return mode that helps in capturing details of complex environments like trees, buildings,

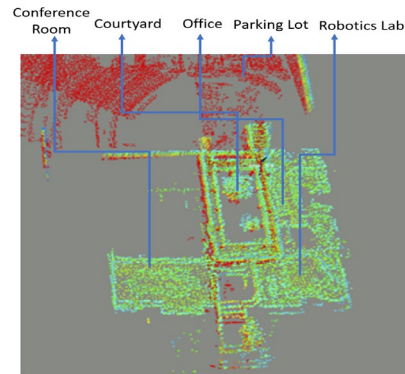


Fig. 7. Indoor mapping using F-LOAM system pipeline on CTRNavi dataset.

TABLE II. AVERAGE ODOMETRY ESTIMATION COMPARISON BETWEEN INDOOR, OUTDOOR AND CAMPUS MAPPING USING F-LOAM ON CTRNAVI DATASET.

	Indoor Mapping	Outdoor Mapping	Campus Mapping
Average Odometry Estimation Time	4.83 ms	13.84 ms	26.82 ms

infrastructures, etc. The advantage of dual return upon single return is that multiple returns significantly increase the number of points available to map, this is true even for smaller objects and non-opaque surfaces. The dataset was collected in the rosbag format. The LiDAR topic and the IMU topic were subscribed during dataset collection. LiDAR topic was published at 10 Hz and the IMU data was published at 100 Hz. The data was collected using the Husky and SPOT robots. The sensor configuration of the robots is illustrated in Fig. 1. The experiments were conducted on a laptop with Ryzen 9 5900 HX (8 cores, 3.3 GHz). The software configuration included Ubuntu 20.04, ROS Noetic. Rviz was used for visualization of the data. Matplotlib and NumPy libraries were used to visualize the trajectory plot.

#### A. Indoor Mapping using Faster-LIO

The robotics lab, conference room, courtyard and the office space were mapped in this scenario. The real-time mapping is illustrated in Fig. 4 along with the estimated trajectory of the robot. Even though the data was collected at 100 Hz for the IMU and 10 Hz for the LiDAR, the mapping was processed at twice the actual clock speed. During the mapping, the LiDAR data was published at 20 Hz and the IMU data was published at 200 Hz.

The average frames per second (FPS) was 585. The LiDAR points need to be added to the iVox data structure in Faster-LIO to update the point cloud. This is crucial for the incremental mapping process. The average time for the addition of each point to the iVox data structure took about 0.007 ms and the incremental mapping process took an average time of 0.036 ms. The standard average preprocessing time took 0.351 ms. Undistort PCL is a process to correct the distortion between LiDAR and IMU sensors, which occurs due to non-linear mapping. The average time it took was 0.195 ms as shown in Table I.

#### B. Outdoor Mapping using Faster-LIO

The outdoor mapping only focuses on the parking lot of

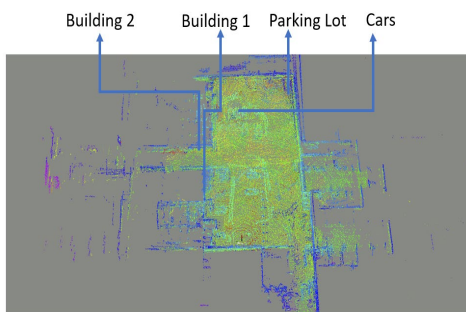


Fig. 8. Outdoor mapping of parking lot using F-LOAM system pipeline on CTRNAvi dataset.

the Reseda Annex building at CSUN. The real-time mapping is illustrated in Fig. 5 where the LiDAR data was published at 20 Hz and the IMU data was published at 200 Hz. The average FPS was 213. The average time for the addition of each point to the iVox data structure took about 0.048 ms and the incremental mapping process took an average time of 0.194 ms. The standard average preprocessing time took 0.383 ms. The average time it took to correct the distortion between the LiDAR and IMU sensors was 0.386 ms.

#### C. Campus Mapping using Faster-LIO

The campus mapping focuses on mapping the corridors, classrooms, HAAS Lab, courtyard, and the food court. It covers a more dynamic environment with different elements and diverse features. The real-time mapping is illustrated in Fig. 6 where the different mapped regions are marked. Like the previous datasets, the LiDAR and IMU data were published at 20 Hz and 200 Hz respectively.

The average FPS was 417. The average time for the addition of each point to the iVox data structure took about 0.120 ms and the incremental mapping process took an average time of 0.181 ms. The standard average preprocessing time took 0.387 ms. The average time it took to correct the distortion between the LiDAR and IMU sensors was 0.247 ms. When the motion of the robot was too abrupt, a few points were dropped in the middle of the mapping process because there were too few points to add to the iVox data structure.

Table I shows the comparison across the average time taken to process the indoor, outdoor and campus mapping for the Faster-LIO based system. It gives a vivid idea about the FPS, distortion correction, iVox addition, and incremental mapping times between the three real-time mappings. It also illustrates the total time taken for the mapping process to complete. When compared with Table II, the average odometry estimation time for F-LOAM is clearly faster than that of Faster-LIO but Faster-LIO is more precise in terms of mapping and state estimation. The distinction between the details captured using Faster-LIO and F-LOAM in terms of the mapping is clearly visible in Fig. 5 and Fig. 8 respectively. In Fig. 5 the cars in the parking lot can be clearly identified whereas in Fig. 8 it is relatively difficult to identify the different elements of the environment even

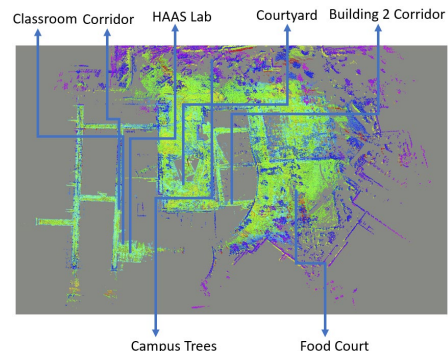


Fig. 9. Campus mapping using F-LOAM system pipeline on CTRNAvi dataset.

though the experiments were conducted in the same outdoor parking lot. Fig. 6 and Fig. 9 illustrate the campus mapping scenario using the Faster-LIO and the F-LOAM algorithms respectively. Faster-LIO has proved to be more robust in terms of mapping and the different elements of the point cloud are more distinguishable than F-LOAM. The state estimation for the robot is also more precise in Faster-LIO because of the fusion of IMU data with the LiDAR data whereas in F-LOAM, the focus is more on improving the computational time while compromising on the quality of the mapping to some extent.

#### D. Indoor mapping using F-LOAM

To further validate the LiDAR inertial odometry, the results were compared with F-LOAM where the mapping is based only on LiDAR and no IMU sensors were used. F-LOAM is a compromise between computational efficiency and quality of mapping of the environment. The indoor mapping is illustrated in Fig. 7 where the different elements of the environment are marked. The average odometry estimation time for each point in the indoor mapping was about 4.83 ms. The pixel size for each point was reduced for easier understanding of the mapping.

#### E. Outdoor Mapping using F-LOAM

The outdoor mapping of the parking lot at Reseda Annex building of CSUN is illustrated in Fig. 8. The average odometry estimation time for each point in the outdoor mapping scenario was about 13.84 ms.

#### F. Campus Mapping using F-LOAM

The campus mapping was also done using the F-LOAM system pipeline to illustrate the different elements of the mapping. The outdoor results proved to be better than the indoor results for the F-LOAM system. Fig. 9 illustrates the campus mapping and each section of the mapping is marked for easier understanding. The average odometry estimation time for each point in the campus mapping was about 26.82 ms. Table II shows the comparison between the odometry estimations between the indoor, outdoor and campus mapping scenarios.

### VI. CONCLUSION AND FUTURE WORK

We introduced a challenging campus tour robot navigation dataset that encompasses complex indoor and outdoor scenarios. Data was collected using custom sensor setup on two mobile robots SPOT and Husky. Two state-of-the-art real-time LiDAR inertial based SLAM system and only LiDAR based system were proposed with the goal to deploy the system on the SPOT and Husky robots for giving tours around campus. Experiment results prove that computationally the LiDAR based system took less time than the LiDAR inertial based system, but the LiDAR inertial system was more accurate in terms of the state estimation of the robot. This research can be further extended to autonomous waypoint-based navigation in the mapped environment along with obstacle avoidance.

### REFERENCES

[1] K. K. M. B. P. Sun, M. Watterson, S. Liu, Y. Mulgaonkar, C. J.

Taylor and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," in *IEEE Robotics and Automation Letters* 3, no. 2 (2018): 965-972.

[2] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15-22. IEEE, 2014.

[3] T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," in *IEEE Transactions on Robotics* 34, no. 4 (2018): 1004-1020.

[4] D. Wang, C. Watkins and H. Xie, "MEMS mirrors for LiDAR: A review," in *Micromachines* 11, no. 5 (2020): 456.

[5] Z. Liu, F. Zhang and X. Hong, "Low-cost retina-like robotic lidars based on incommensurable scanning," in *IEEE/ASME Transactions on Mechatronics* 27, no. 1 (2021): 58-68.

[6] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," in *IEEE Robotics and Automation Letters* 7, no. 2 (2022): 4861-4868.

[7] W. Xu, Y. Cai, D. He, J. Lin and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," in *IEEE Transactions on Robotics* 38, no. 4 (2022): 2053-2073.

[8] A. Gupta and X. Fernando, "Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges," in *Drones* 6, no. 4 (2022): 85.

[9] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3126-3131. IEEE, 2020.

[10] Z. Wang, L. Zhang, Y. Shen and Y. Zhou, "D-liom: Tightly-coupled direct lidar-inertial odometry and mapping," in *IEEE Transactions on Multimedia* (2022).

[11] H. Wang, C. Wang, C.-L. Chen and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4390-4396. IEEE, 2021.

[12] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586-606. Spie, 1992.

[13] A. Segal, D. Haehnel and S. Thrun., "Generalized-icp," in *Robotics: Science and Systems*, vol. 2, no. 4, p. 435. 2009.

[14] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, pp. 1-9. 2014.

[15] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758-4765. IEEE, 2018.

[16] W. Zhen, S. Zeng and S. Sroberer, "Robust localization and localizability estimation with a rotating laser scanner," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6240-6245, 2017.

[17] Y. Balazadegan Sarvrood, S. Hosseinyalamdary and Y. Gao, "Visual-LiDAR odometry aided by reduced IMU," in *ISPRS International Journal of Geo-information*, no. 1 (2016): 3.

[18] X. Zuo, P. Geneva, W. Lee, Y. Liu and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5848-5854. IEEE, 2019.

[19] P. Geneva, K. Eickenhoff, Y. Yang and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 123-130. IEEE, 2018.

[20] H. Ye, Y. Chen and M. Liu., "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3144-3150. IEEE, 2019.