Evaluation of LiDAR Inertial Odometry method with 3D LiDAR-based Sensor Pack

Samuel Ogunniyi
Centre for Robotics and Future Production,
Council for Scientific and Industrial Research
Pretoria, South Africa
sogunniyi@csir.co.za

Daniel Withey
Department of Engineering Design and Mathematics,
University of the West of England
Bristol, United Kingdom
dan.withey@uwe.ac.uk

As the 4th industrial revolution emerges at the forefront of South Africa's national strategy, research areas like mapping and localization find importance in more fields than just robotics. The mining industry is well-positioned to be a potential beneficiary of these technological changes. By nature, mining settings could be labeled with a similar status to indoor areas as they are both GPS-denied type environments. Mapping and localization algorithms using Simultaneous Localization and Mapping (SLAM) are proven to function in similar conditions. These SLAM-based algorithms are highly effective at mapping, yet they can be susceptible to registration, motion distortion, and drift issues if provided with no external odometry. Also, using mobile robots may not always be possible in these environment types for practical reasons. Employing a device with a different form factor, such as a mapping sensor pack, could be an option. This study evaluates a Lidar Inertial Odometry solution integrated on a LiDAR-based sensor pack developed for mapping and localization applications. For the chosen LiDAR Inertial Odometry (LIO) solution the Root Mean Squared Error was computed. This was found to be greater than the Root Mean Squared Error computed by the LiDAR sensor pack's Eth-ICP Mapper implementation. However, the LIO solution produces pose estimates at a higher rate, which is beneficial for localization continuity and mapping.

Keywords—LiDAR-Inertial, Odometry, SLAM, Mapping, Localization, ICP, Sensor pack.

I. Introduction

While the research area of Mapping and Localization has long been the purview of mobile robotics, its significance has not gone unnoticed within other industry sectors, such as manufacturing and mining. Applications that make use of 3D maps have emerged for GPS-denied environments [1, 2, 3]. Stereo cameras and Light Detection and Ranging (LiDAR) sensors have found use in mapping applications [4, 5]. Nevertheless, mapping in GPS-denied environments has its own set of challenges. For example, the unreliability of GPS in indoor or underground environments results in unreliable position estimates for the mapping process [6, 7]. Additionally, mining environments are more unforgiving to mapping sensors, having narrow cross-sections, inconsistent illumination, and near-homogeneous surfaces. As a result, camera sensors are rendered less effective in these types of settings [8]. Comparatively speaking, LiDAR sensors are not invulnerable to the effects of mapping in GPS denied environments, owing to low vertical resolution, vulnerability to motion distortion, poor performance in degraded visual environments, and low update rate [11].

Being able to self-localize a mapping sensor within these difficult conditions goes some way towards resolving these issues. Robotics research up to the present day has continued to drive the development of algorithms, which allow the robot to self-localize in the map it is generating, typically managed via the Simultaneous Localization And Mapping (SLAM) methodology [9]. SLAM is typically implemented through point cloud registration. Point cloud registration (also known as geometric registration) is often administered by the popular Iterative Closest Point (ICP) algorithm [2]. The ICP algorithm extracts and processes points in the environment to determine where the LiDAR sensor's frame is relative to the map frame. It is important to note that the LiDAR sensor can typically obtain laser scans at a frequency of about 10Hz [10, 11], due to the rotational speed of the internal mechanism in conventional LiDAR sensors [12]. In addition, there is computational load due to the SLAM algorithm, as the map size increases throughout the mapping process. In the mapping process, if the motion of the mobile robot is too rapid, the LiDAR sensor may not be able to adequately capture environment data in each scan. Missing data points in the LiDAR scan data give rise to motion distortion artifacts and registration issues, in the emerging point cloud. These issues are compounded if the features in the mapped environment are sparse and if the environments have homogenous surfaces, with little texture.

These challenges are typically solved by algorithmically fusing motion data from additional motion sensors, such as Inertial Measurement Units (IMU) and optical wheel encoders, with LiDAR data. The optical encoders provide a measure of the change in position over time known as odometry [6]. Odometry is defined as an output of the following steps. Firstly, the encoders produce measurements relative to an origin position. Secondly, these measurements are converted into position estimates. Lastly, the position estimates are summed. That summed term is known as Odometry [6]. The IMU sensor also provides translation estimations by integrating accelerometer measurements and orientation estimations from gyroscope measurements [13]. Also, calculating the transformation between the sensor body frames and the body frame of a mobile robot allows the odometry to be produced to be relative to the robot. As a result, odometry is computed as the robot moves, in the case of a wheeled robot.

In a lab setting, floor surfaces are usually level. As a result, light encoders record little motion drift due to wheel slippage. The sensor fusion also tempers the IMU's inherent error gyroscope and accelerometer measurements [6]. Furthermore, the IMU and the light encoder update rates provide odometry estimates faster than the SLAM algorithm can provide its position estimates, as it is delayed by its point cloud registration process. In this context, the speed of the mobile robot is usually thresholded.

In some circumstances, it is not always possible to have a mobile robot due to cost or the environment being unsuitable for a wheeled robot. Some examples are areas in underground mines or certain agricultural contexts, where it is rough, uneven terrain. Furthermore, wheel odometry data can be unreliable due to wheel slippage on these types of terrain [6]. While it may not be possible to utilize a mobile robot for each scenario, it may be possible to harness the mapping and motion sensors in a different form factor, such as a sensor pack, a handheld device, or an Unmanned Aerial Vehicle (UAV). These forms provide no access to wheel odometry; hence, a need exists for odometry to be obtained from additional sensors. From a functionality point of view, the LiDAR and IMU sensors used in robotic platforms lend well for usage in these devices, as they could perform double duty for mapping and producing odometry without the added weight of extra sensors.

When employed individually, the LiDAR and IMU sensors have inherent weaknesses, as previously established. However, as previously mentioned, the fusion of the sensor data can be beneficial, compensating for the shortcomings of individual sensors [6, 10]. For example, when calculating odometry from a LiDAR alone, motion distortion can play a role when at speed. Also, the LIDAR odometry depends on an initial position estimate, which is hard to compute without extra sensors [15]. In contrast, the IMU sensor, which is superb at providing initial or prior pose estimates [16], publishes motion data at a rate of an order of magnitude (typically 100Hz and above) to that of the LIDAR. The odometry estimate produced by the fusion of these sensors is at a higher frequency than the LiDAR scan rate to counteract the motion distortion [17, 18, 27]. The type of odometry generated from IMU and LiDAR sensor fusions is termed LiDAR Inertial Odometry. In this paper, we present an evaluation of an open-source tightly coupled LIO software implementation proposed to address these types of issues. The remainder of the paper is structured as follows. Section II presents related works on LiDAR Inertial odometry and a brief overview of relevant 3D mapping techniques. Section III details the system overview of both the hardware and software. Section IV shows the experiments and results. Section V concludes the paper with a brief discussion of the results and further work suggestions.

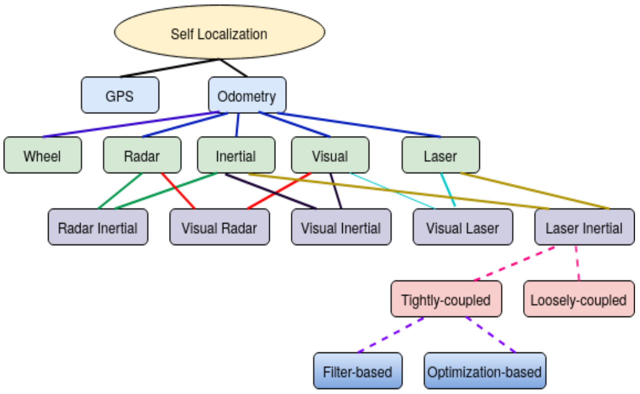


Fig. 1. The hierarchy of self-localization approaches.

II. Related Work

A hierarchy of self-localization strategies has emerged from research into sensor fusions, which are conceptually similar but differ in methodology [6]. Fig. 1 shows an equivalent diagram of the hierarchy of self-localization approaches [6], primarily focusing on the breakdown of experimentation under the LiDAR Inertial Odometry research area

A. LiDAR Inertial Odometry

LiDAR Inertial Odometry (LIO) stems from the intuitive research work conducted on the fusion of LiDAR sensors data with IMU sensor data to derive odometry estimations. In the last decade, LiDAR Inertial Odometry (LIO) research has split into two main research areas, namely tightly coupled and loosely coupled LIO [27].

B. Loosely Coupled LiDAR-Inertial Sensor Fusion

In the loosely coupled category, LiDAR inertial Odometry fuses LiDAR motion estimates and IMU integration results. The following works detail certain aspects of the background of LIO. The most notable works in literature for Loosely coupled LIO are LOAM [20] and LeGO-LOAM [21]. In LOAM [20], the focus is on using the LiDAR odometry obtained from an ICP implementation, extracting features, namely planes or edges. In this case, the IMU is used to deskew point clouds from the rotating LiDAR sensor, provide initial pose estimates, and counteract the LiDAR sensor's motion distortion. LeGO-LOAM [21] builds on [20] by dropping questionable points from the point cloud in a process called point cloud segmentation. In addition, LeGO-LOAM also performs loop-closure. In LOAM and LeGO-LOAM, IMU bias remains, under the assumption that its effect is negligible over a short duration. Other works include using extended Kalman filters to fuse LiDAR and IMU data for 2D AND 3D maps [22, 23, 24]. In general, EKF based methods are computationally efficient. However, they are sensitive to information loss, imprecise estimates, and also capturing outlier data points.

C. Tightly Coupled LiDAR-Inertial Sensor Fusion

The LiDAR and IMU sensors are used in the tightly coupled category to generate odometry based on raw measurements. The tightly coupled methodology is broken down further into the filter-based sensor fusion approaches and the optimization-based sensor fusion approaches [25, 26]. Regarding the area of filter-based sensor fusion, the following works are noteworthy. Firstly, the Gaussian particle filter (GBF) is used to fuse from a planar 2D LiDAR and an IMU [28]. However, Kalman filters are viewed as more favorable to use, as the GBF's computation complexity increases along with the increasing number of features identified in the scanned point cloud [29]. Among the Kalman filters, the Extended Kalman filter (EKF) is the most popular, as seen in LINS [30] and R-LINS [31]. In these works, the error-state Kalman filters are used iteratively with a robocentric model to provide an ego-motion estimation and improve state estimation. [32] presents an implementation of the error-state Kalman filter with IMU bias estimation, used with a Digital Elevation Model (DEM) to provide quality state estimations. Recursive use of the EKF in Fast-LIO [29] reduces the effect of linearization errors and the computational time inherent in LiDAR-IMU fusion. The most prominent optimization method for tightly coupled LiDAR-Inertial sensor fusions is Factor Graph Optimization (FGO). FGO tightly coupled approaches are superior to EKF-based estimators at linking data in successive periods, as they can handle delayed measurements [33]. These delayed measurements can be additional factors to the factor graph. Geneva et al. [34] use IMU pre-integration measurement data, with plane feature points from their LiDAR data, to constrain their FGO implementation, similarly to Yang et al. [35]. Tixiao et al., with their LIO-SAM [36] framework, fuses IMU pre-integration data with LiDAR measurements using FGO. LIO-SAM suffers in narrow self-similar settings, for example, corridors. LIO-SAM is useful for sensor fusion as supplementary sensor measurements can be added to the factor graph as new factors. The use of specific keyframes coinciding with newly computed Lidar odometry minimizes the effect of older LIDAR scans, providing real-time performance. Contrastingly, the LiLi-OM [37] solution uses two separate factor graphs of different sizes. The larger factor graph adds particular keyframes for the pre-integrated IMU data and sliding window optimization. In contrast, the smaller factor graph retrieves the poses from other keyframes within the sliding window.

The Lidar Inertial odometry (LIO) and mapping [11] algorithm termed "LIO-Mapping" evaluated in this study also uses graph optimization. Here, the algorithm uses a maximum a posteriori (MAP) formulation to optimize both LiDAR and IMU measurements jointly within a sliding window. Rotational constraints to improve the final pose within a generated 3D Map. Here the raw imu data is used via state prediction step to deskew the incoming raw LiDAR data and as a pre-integrated input to the joint non-linear optimization cost function. In the next phase, features are extracted from the deskewed LiDAR data to build and update a local map, using previously obtained LiDAR features and transform estimates from the state prediction step. These relative LiDAR measurements are simultaneously acquiring relative LiDAR measurements, along with a LiDAR to world passed along with the previously mentioned pre-integrated IMU measurements to a Mahalanobis-based cost function, which is minimized in an optimization step to compute a MAP estimation of the new IMU state and the transforms between the IMU frame and the LiDAR frame. Furthermore, the transforms, from a previous iteration, namely the imu frame to lidar frame and the world to IMU frame, are used to update the local map, along with the extracted features. Subsequently, the output from the optimization step feeds back into the state prediction step to prevent IMU drift. Lastly, the acquired LiDAR poses are constrained to a fixed world frame by registering the feature points to the global map. In the process, the execution of the steps recursively provides the required odometry. Fig. 2 shows a simplified version of the LIO-mapping framework [11].

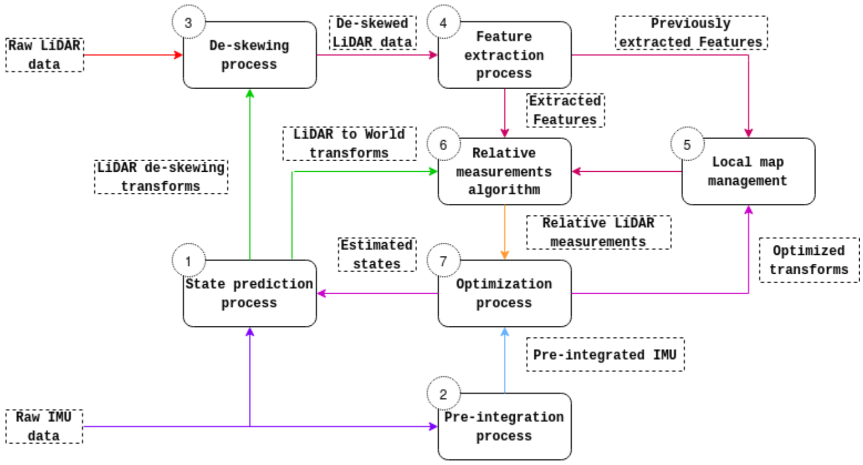


Fig. 2. The LIO-mapping framework diagram.

D. Mapping and Localization

Historically, Mapping and Localization is a well-studied research area [46] [47]. The Simultaneous Localization and Mapping (SLAM) approach [40] is synonymous with Mapping and Localization research and has led to a variety of implementations [43] [44] [45]. The SLAM is a self-reflexive process, which uses a LiDAR or stereo camera-sensor to build a map of an environment and simultaneously determine its position within said map [40]. SLAM comes in different flavors, such as fastSLAM or EKF-SLAM [39] [40] and uses the Iterative closest point (ICP) algorithm [41], a form of geometric registration. The ICP algorithm builds maps by merging smaller point cloud data sets after identifying features and attributes in successive LiDAR point cloud scans. These features pertain to the geometric shapes of objects within the environment and attributes which are key descriptors for the data points in the point cloud. A close to optimum transformation is computed from features in consecutive data sets as the LiDAR sensor scans the environment. The limitations of LiDAR sensors are well documented [10] [11] [12] and indicate a need for external odometry to combat the registration, motion distortion, and drift issues that come with using these sensors on their own.

III. System overview

A. LiDAR Sensor pack Hardware

The hardware employed for this study is a LiDAR-based sensor pack. The sensor pack comprises of the following components: a Velodyne puck (VLP16) LiDAR sensor [46], a Microstrain 3dm-gx3-25 IMU sensor [47], a 3rd generation core i7 CPU with mini-ITX form factor, a DC-to-DC convertor as well as two 12V batteries. The puck sensor has the following specifications: a measurement range of up to 100m, a range accuracy of up at ± 3cm, a vertical Field of view (FOV) of ± 15°, a horizontal FOV of 360°, a vertical angular resolution of 2°, horizontal angular resolution ranging between 0.1° to 0.4° and a rotation rate of between 5 to 10Hz.

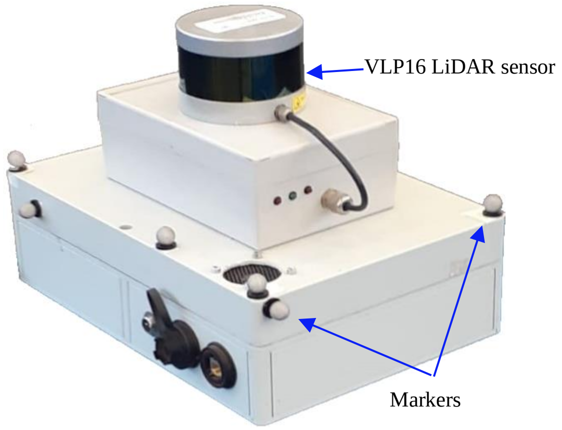


Fig. 3. The LiDAR sensor pack.

The extensive 3dm-gx3-25 sensor specifications are available online [47].

Fig 3. shows an image of the LiDAR sensor pack used for this study. The sensor pack has retro-reflective markers on its body, to track it in the volume of the motion capture system, as explained in the next section.

B. Vicon Motion Capture System

A Vicon Motion Capture System (MCS) [51] with 12 MX-T40 cameras, which has submillimetre accuracy was used to provide ground truth for the LiDAR sensor pack. The MCS can track objects within a 7m × 8m × 3m volume. The object (in this case the LiDAR sensor pack) is defined by placing markers in different positions of the object's body. Figure 4 shows an image of the Vicon MCS setup in the CSIR's Centre for Robotics and Future production Systems (CRFP) lab. Our other work shows an illustration of how the sensor pack is tracked within the Vicon capture volume [2].

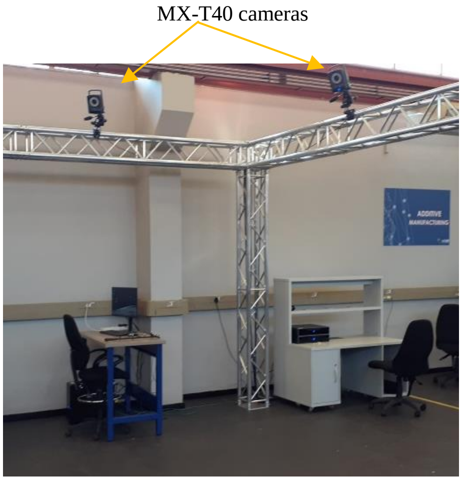


Fig. 4. The Vicon motion capture system

C. Software System Overview

The LiDAR sensor contains a mapping and localization software package, which includes an implementation of the open-source Ethz-asl ICP Mapper [48] package, to perform SLAM. The onboard software package combines subsequent LiDAR point cloud scans to form a map and simultaneously provides position estimates. For the LiDAR Inertial Odometry, an open-source LIO-mapping software package [49] was chosen, which provides auxiliary LIO (odometry) when integrated the with mapping and localization software package.

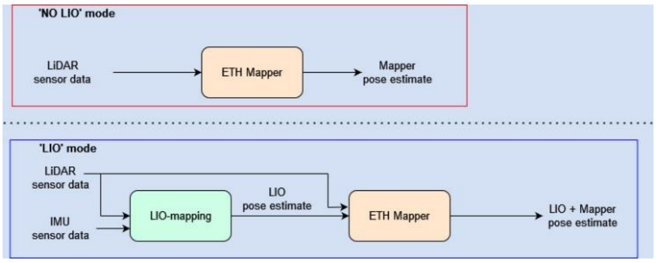


Fig. 5. The flow of data diagram for 'No Odometry' and 'LIO ' modes.

Fig 5. shows a diagram that explains the flow of data from the sensor data inputs to the pose estimate outputs a frequency of 0.5Hz. On the other hand with the LIO enabled the software package provides regular interim position estimates at a frequency of 3.3Hz.

.

D. Frames and Transformations

Fig. 6 shows the main system frames (i.e. the map, odom, and lidar frame), for the integrated system. The above-mentioned Mapping and Localization software package computes the pose estimates of the "odom frame" with the "map frame" via a transformation. The LIO-mapping package provides the LIO (odometry) of the "sensor pack frame" within the "odom frame". Before movement at the starting position, all the frames are aligned, but as the sensor pack is moved the "odom frame" moves relative to the map frame and the sensor pack frame moves relative to the odom frame respectively. The transformations between the frames were previously computed and are available when the package is executed for the mapping and localization process.

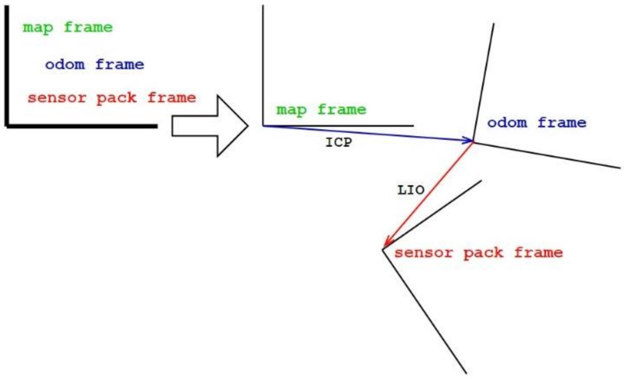


Fig. 6. Main system frames.

IV. Experiments and Results

The LiDAR sensor pack was placed on a trolley and pushed in figure 8 path around two obstacles in the CRFP lab.

A. Pose estimates Tests

The LiDAR sensor pack was placed on a trolley and pushed in figure 8 path around two obstacles in the CRFP lab. The experimental setup is shown in Fig. 7.

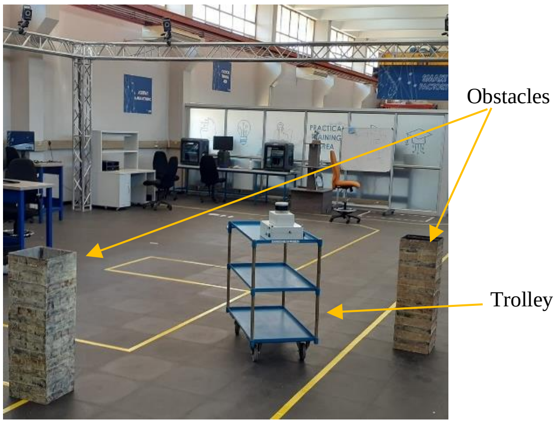


Fig. 7. The experimental setup

B. *Pose estimates Results*

For the figure 8 path test, the position estimates data is from two sources, namely the Vicon MCS and the onboard mapping and localization software package. After the LIO integration, the software package provides position estimates in two modes. The first mode is the ‘No odometry’ mode, which provides Mapper pose estimates. The second mode is the ‘LIO’ mode, which produces both the LIO and the LIO + mapper pose estimates, as illustrated in Fig. 5. The following pose estimates types were captured for the pose estimates test, namely the Mapper, the LIO, the LIO + mapper, and the Vicon pose estimates respectively. For the designated figure 8 path, the LiDAR sensor pack moved at an average velocity of ~0.44 m/s, about the path. The average velocity value was computed using the timestamped Vicon pose estimates. Fig. 8 shows the plots of the pose estimates. The plots of the position estimates show a rougher path produced by the LIO pose estimates, while a smoother path is produced by the Mapper and the LIO+ Mapper pose estimates. The Mapper pose estimates path adheres more closely to the produced Vicon (ground truth) path.

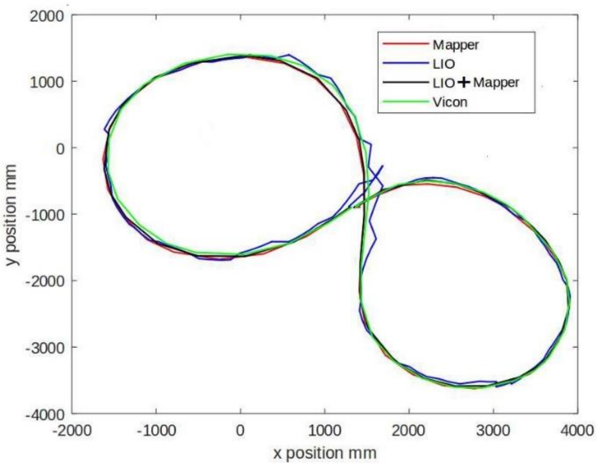


Fig. 8. The plot of x and y position estimates on a figure 8 path

From Fig. 8, it can be seen that the LIO pose estimates do not improve on the Mapper pose estimates position accuracy. However, it produces more than twice the number of pose estimates compared to the Mapper pose estimates in the 'No Odometry' case. Passing the LIO pose estimates to the Eth Mapper produced an improved position accuracy, as seen by the black LIO mapper plot. In the ‘No Odometry‘ and ‘LIO’ modes the 3D point clouds produced were indistinguishable, meaning the LIO mapper position estimates produced were accurate enough for the 3D Mapping process.

C. *Root Mean Square Error Values*

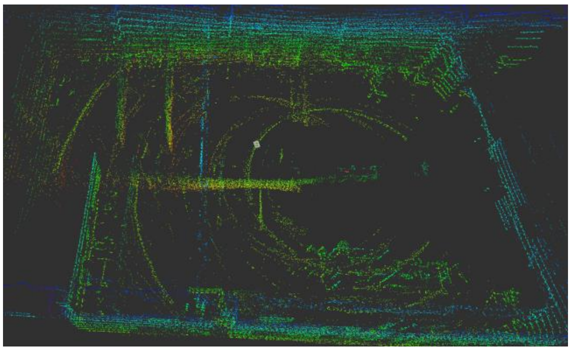
Calculating the Root Mean square error (RMSE) values for the Mapper, the LIO, and the LIO mapper pose estimates relative to the Vicon values indicate that, the software package is more accurate at producing position estimates when in ‘No Odometry’ mode, than when in the ‘LIO’ mode, as shown in TABLE 1. Taking Fig. 8 into consideration along with the RMSE values confirms that the output LIO + mapper pose estimates attain accurate values, though not as accurate as the inherent Eth Mapper implementation when no odometry is available. Essentially, the LIO pose estimates are provided to the Eth Mapper at a faster rate than when in the ‘No Odometry‘ mode, resulting in an improved RMSE value for the LIO mapper pose estimates. While the LIO pose estimates are produced at a higher frequency of ~ 3.3 Hz, the computation times of the Eth Mapper remain the same producing LIO mapper pose estimates at ~ 0.5 Hz.

TABLE I. ROOT MEAN SQUARE ERROR VALUES

Pose estimation sources	**Average Frequency of pose estimates produced in Hz**	**Number of position estimates produced**	**RMSE in metres (m)**
Mapper pose estimate	~0.5 Hz	92	0.19 m
LIO pose estimate	~3.3 Hz	188	0.31 m
LIO + Mapper pose estimate	~0.5 Hz	63	0.23 m

D. *3D Mapping Trolley Speed Tests*

The tests included pushing the LiDAR-based sensor pack through the CRFP lab on a trolley to visualize if possible motion distortion or point cloud registration issues. In the case of the ‘No Odometry’ mode, the resultant 3D point cloud (as seen in Fig. 8) showed motion distortion artifacts and could not accurately localize the sensor pack. In the case when the ‘LIO’ mode was enabled, the resultant 3D point cloud showed no signs of motion distortion or registration issues and could accurately localize the sensor pack. The 3D point cloud with the ‘LIO’ mode is shown in Fig. 9. The 3D point clouds indicate that with the LIO enabled the software package was found to be more speed tolerant, due to its publishing of position estimates at a higher frequency. The 3D point clouds in Fig. 9 and 10 use the same LiDAR and IMU input data. That been said, there is a marked difference in how the Mapping and Localization software handles the 3D map generation at speed when the LIO is enabled. The highest recorded speed was at about 2.4 metres per second (m/s), with the LIO enabled.



3D point cloud generated at speed in the ‘No Odometry’ mode.

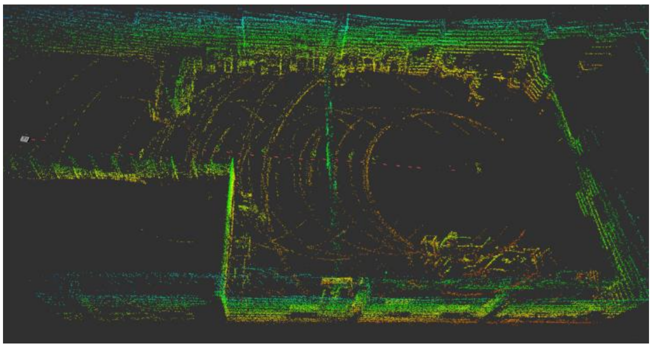


Fig. 9. 3D point cloud generated at speed in 'LIO' mode.

II. Conclusions and Future Work

In the case when the LiDAR Inertial Odometry (LIO) is enabled, it produces decent enough pose estimates, though not as accurate as the pose estimates produced by the sensor pack's Eth-ICP Mapper. However, the position estimates are sufficiently accurate as there is no visible difference in the 3D point clouds produced in both position estimation modes. In addition, the LIO position estimates are produced at a higher frequency, which provides invaluable resistance to motion distortion and registration issues while the sensor pack is at speed. The highest speed recorded from pushing the sensor pack on a trolley was 2.4 metres (m/s). In contrast, the Mapping and Localization software without LIO enabled suffers from motion distortion, registration issues, and can not properly localize itself at these higher recorded speeds. The recommended future works are as follows:

- Running further pose estimate tests, including more complex motions with roll, pitch, and yaw features.
- Conducting the speed tests at higher velocities than possible on a trolley, for example, strapping the sensor pack to a vehicle.

References

[1] H. Qin et al., "Autonomous Exploration and Mapping System Using Heterogeneous UAVs and UGVs in GPS-Denied Environments," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1339-1350, Feb. 2019.

[2] S. Ogunniyi, D. Withey, S. Marais, and G. Crafford, "LiDAR-based 3D mapping and localisation system for ground penetrating radar," in Proc. SAUPEC/RobMech/PRASA Conf., 2020, pp. 1–6.

[3] Z. Ren, L. Wang and L. Bi, "Robust GICP-Based 3D LiDAR SLAM for Underground Mining Environment", Sensors, vol. 19, no. 13, 2019, p. 2915.

[4] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama and S. Mita, "Lidar scan feature for localization with highly precise 3-D map", Proc. IEEE Intell. Vehicle Symposium, pp. 1345-1350, 2014.

[5] T. Caselitz, B. Steder, M. Ruhnke and W. Burgard, "Monocular camera localization in 3D lidar maps", Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys., pp. 1926-1931, 2016.

[6] S. A. S. Mohamed, M. -H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen and J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems," in *IEEE Access*, vol. 7, pp. 97466-97486, 2019

[7] S. Poddar, R. Kottath and V. Karar, "Evolution of Visual Odometry Techniques", arXiv.org, 2021. [Online]. Available: https://arxiv.org/abs/1804.11142. [Accessed: 05- Aug- 2021].

[8] C. Debeunne and D. Vivet, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping," Sensors, vol. 20, no. 7, p. 2068, Apr. 2020.

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006

[10] J.-C. Yang, C.-J. Lin, B.-Y. You, Y.-L. Yan, and T.-H. Cheng, "RTLIO: Real-Time LiDAR-Inertial Odometry and Mapping for UAVs," Sensors, vol. 21, no. 12, p. 3955, Jun. 2021.

[11] H. Ye, Y. Chen and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3144-3150.

[12] "Explore LiDAR Sensors to Improve the Peripheral Perception | Market Prospects", Market-prospects.com, 2021. [Online]. Available: https://www.market-prospects.com/articles/lidar-sensors. [Accessed: 06- Aug- 2021].

[13] J. Shen, D. Tick and N. Gans, "Localization through fusion of discrete and continuous epipolar geometry with wheel and IMU odometry," Proceedings of the 2011 American Control Conference, 2011, pp. 1292-1298

[14] [C. Le Gentil, T. Vidal-Calleja and S. Huang, "IN2LAAMA: Inertial Lidar Localization Autocalibration and Mapping", IEEE Transactions on Robotics, vol. 37, no. 1, pp. 275-290, 2021. Available: 10.1109/tro.2020.3018641.

[15] W. Wen, L.-T. Hsu, and G. Zhang, "Performance Analysis of NDT-based Graph SLAM for Autonomous Vehicle in Diverse Typical Driving Scenarios of Hong Kong," Sensors, vol. 18, no. 11, p. 3928, Nov. 2018.

[16] H. Xue, H. Fu, and B. Dai, "IMU-Aided High-Frequency Lidar Odometry for Autonomous Driving," Applied Sciences, vol. 9, no. 7, p. 1506, Apr. 2019.

[17] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic LiDAR fusion: Dense map-centric continuous-time SLAM,"in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 1206–1213.J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," Autonomous Robots, vol. 41, no. 2, pp. 401–416, 2017.

[18] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping", Autonomous Robots, vol. 41, no. 2, pp. 401-416, 2016.

[19] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 4758-4765

[20] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013, pp. 3923–3929.

[21] J. Tang et al., "Lidar scan matching aided inertial navigation system in gnss-denied environments," Sensors, vol. 15, no. 7, pp. 16 710–16 728, 2015.

[22] M. Demir and K. Fujimura, "Robust Localization with Low-Mounted Multiple LiDARs in Urban Environments," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 3288-3293

[23] S. Thrun, Probabilistic Robotics. Cambridge, Massachusetts: The MIT Press, 2005.

[24] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception", Foundations and Trends in Robotics, vol. 6, no. 1-2, pp. 1-139, 2017.

[25] J. Zhang, W. Wen, F. Huang, X. Chen and L. Hsu, "Coarse-to-Fine Loosely-Coupled LiDAR-Inertial Odometry for Urban Positioning and Mapping", Remote Sensing, vol. 13, no. 12, p. 2371, 2021.

[26] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in 2012 IEEE International Conference on Robotics and Automation. IEEE,2012, pp. 1–8.

[27] W. Xu and F. Zhang, "FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter", IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3317-3324, 2021.

[28] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A lidar-inertial state estimator for robust and efficient navigation," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 8899–8906.

[29] C. Qin, H. Ye, C.E. Pranata, J. Han, S. Zhang, and Ming Liu, "RLINS: A Robocentric Lidar-Inertial State Estimator for Robust and Efficient Navigation," arXiv:1907.02233, 2019.

[30] G. Hemann, S. Singh, and M. Kaess, "Long-range gps-denied aerial inertial navigation with lidar localization," in Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 1659–1666.

[31] W. Wen, T. Pfeifer, X. Bai and L. Hsu, "It is time for Factor Graph Optimization for GNSS/INS Integration: Comparison between FGO and EKF", arXiv.org, 2021. [Online]. Available: https://arxiv.org/abs/2004.10572. [Accessed: 06- Aug- 2021].

[32] P. Geneva, K. Eckenhoff, Y. Yang and G. Huang, "LIPS: LiDAR-Inertial 3D Plane SLAM," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 123-130.

[33] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu and G. Huang, "Tightly-Coupled Aided Inertial Navigation with Point and Plane Features," 2019 International Conference on Robotics and Automation (ICRA), 2019

[34] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020

[35] K. Li, M. Li and U. D. Hanebeck, "Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5167-5174, July 2021

[36] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 5135-5142.

[37] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," Machine Learning, vol. 31, pp. 29–53, Apr 1998.

[38] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): part ii," IEEE Robotics Automation Magazine, vol. 13, pp. 108–117, Sep. 2006.

[39] F. Pomerleau, F. Colas, and R. Siegwart, "A Review of Point Cloud Registration Algorithms for Mobile Robotics," Foundations and Trends in Robotics, vol 4, pp. 1-104, May 2015.

[40] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization,"IEEE Transactions on Robotics and Automation, vol. 17, pp. 125–137, April 2001.

[41] F. Pei, H. Yan, and M. Zhu, "Improved fastslam system using artificial fish-swarm optimized distributed unscented particle filter," in 2017 36th Chinese Control Conference (CCC), pp. 6951–6956, July 2017.

[42] S. Grzonka, C. Plagemann, G. Grisetti, and W. Burgard, "Look-ahead proposals for robust grid-based slam with rao-blackwellized particle filters," I. J. Robotic Res., vol. 28, pp. 191–200, 02 2009.

[43] S. Oh, M. Hahn, and J. Kim, "Simultaneous localization and mapping for mobile robots in dynamic environments," in 2013 International Conference on Information Science and Applications (ICISA), pp. 1–4, June 2013.

[44] Y. Jia, X. Yan and Y. Xu, "A Survey of simultaneous localization and mapping for robot," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019, pp. 857-861,

[45] S. Thrun,, "Robotic Mapping: A Survey", Exploring Artificial Intelligence in the New Millennium, 2002.

[46] "Puck Lidar Sensor, High-Value Surround Lidar | Velodyne Lidar", Velodyne Lidar, 2021. [Online]. Available: https://velodynelidar.com/products/puck/. [Accessed: 07- Aug- 2021].

[47] "3DM-GX3® -25 | LORD Sensing Systems", LORD Sensing Systems | Innovating Together, 2021. [Online]. Available: https://www.microstrain.com/inertial/3dm-gx3-25?qt-product_quicktab=1#qt-product_quicktab. [Accessed: 07- Aug- 2021].

[48] Ethzasl_icp_mapping,2019.[Online]. Available:https://github.com/ethz-asl/ethzasl_icp_mapping. [Accessed: 24- May-2021].

[49] "LIO-mapping: Implementation of Tightly Coupled 3D Lidar Inertial Odometry and Mapping (LIO-mapping)", [Online]. Available: https://github.com/hyye/lio-mapping. [Accessed: 07- Aug- 2021].

[50] M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," in IEEE Transactions on Robotics, vol. 24, no. 6, pp. 1365-1378, Dec. 2008, doi: 10.1109/TRO.2008.2006706.

[51] "The Vicon motion capture system." URL: https://www.vicon.com/about-us/what-is-motion-capture/. Last visited on 2021/08/20.