2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 1-4, 2023, Oahu, Hawaii, USA

Level Plane SLAM: Out-of-plane Motion Compensation in a Globally Stabilized Coordinate Frame for 2D SLAM

Samuel Lovett[1], Tyler Paquette[2], Brayden DeBoon[2], Sreeraman Rajan[1], and Carlos Rossa[1]

***Abstract*—Two-dimensional (2D) simultaneous localization and mapping (SLAM) using a LIDAR is a method used to track the position and orientation of a moving platform. 2D-SLAM assumes that the platform translates in a 2D plane and can only rotate about an axis perpendicular to that plane. However, the assumption of no out-of-plane (OOP) motion does not hold true for platforms experiencing motion in six degrees-of-freedom (6-DOF), such as wearable technologies that have no 3D LIDAR. This paper proposes a new algorithm, called the Level Plane for SLAM (LPS) for removing OOP motion from 2D-LIDAR scans generated on platforms experiencing 6-DOF without requiring scan-matching in 3D. Like other existing methods, an IMU is combined with a 2D-LIDAR to determine the platform's orientation, capture OOP motion, and generate a scan in 3D. Unlike other methods, OOP motion is removed by projecting scans onto a globally stabilized coordinate frame in 2D where both scan matching and map alignment take place. The proposed algorithm is validated over a series of experiments with different levels of induced and observed OOP motion. Experimental results show that LPS is able to handle more OOP motion than other algorithms and run in real-time.**

***Index Terms*—SLAM, scan matching, signal filtering, LIDAR odometry, Lidar-IMU fusion, Hector SLAM**

I. Introduction

In order to localize a moving platform, a map is required and to generate a map, information about the platform's location is required. These are the fundamental principles behind simultaneous localization and mapping (SLAM) [1]. SLAM is widely used in the fields of mobile robotics, autonomous vehicles, augmented reality, tracking, path planning, and navigation [2]–[4].

SLAM generates a map and localizes the platform within that map in real-time, and as such accurately estimates the motion of the platform. This is done by measuring the distance between the platform and relevant landmarks in the environment from different locations. A common type of sensor used in SLAM is a 2D-LIDAR. A 2D-LIDAR sends out beams of light that are reflected off objects and returned to the sensor in the LIDAR. By observing the time of flight (ToF) or how long the beam takes to return to the sensor, the distance to the object can be calculated [5]. A LIDAR scan is then made by rotating the beam generator about its axis and recording the distances generated over one rotation. Since a 2D-LIDAR only has one beam generator, it is only able to measure the ToF in the plane of the beam.

After the data is acquired, the next step in SLAM is to recover a map of the environment surrounding the sensor. Consider a LIDAR scan containing the distances between the sensor and its environment, at position A. The LIDAR then moves to position B, and a second scan is taken. The translation and rotation required to align the scans at A and B represent the encoded motion of the LIDAR between the scans.

Scan matching is the technique of decoding motion from two partially overlapping scans [6]–[8]. Scan matching becomes challenging when only a few points between the scans overlap. If scans have no overlap, the scan matching will incorrectly align them and create an inaccurate map, which can no longer be used for platform tracking, path planning, or navigation. 2D-SLAM avoids this problem of divergent scans by constraining the 2D-LIDAR's platform motion to movement on a plane and rotation about an axis normal to that plane. This assumes that all the LIDAR scans are kept level.

The assumption that the LIDAR has been kept level presents an issue for platforms that experience motion in 6-DOF, such as robots in rough terrains, drones, and wearable technology. If the 2D-SLAM algorithm does not account for the out-of-plane (OOP) motion, it assumes that all scans overlap, even when they do not. This results in an inaccurate map of the environment. In Fig. 1 this problem is illustrated by two LIDAR scans A and B observing the same wall from the same location. In A, the distance d_1 to the wall is reported correctly; however, in B the LIDAR is not kept level and the distance is incorrectly reported as $d_2 > d_1$, thus, inaccurately representing two distinct walls.

A common solution to account for OOP motion is to perform scan matching in 3D using a 3D-LIDAR. Although 3D scan matching captures the platform's OOP movement, 3D-LIDARs are bulky, expensive, and have slower scan rates than 2D-LIDARs. To avoid using 3D-LIDAR a 2D-LIDAR can be transformed into a 3D-LIDAR by mounting its base to a servo

We acknowledge the support of Teaching City Oshawa, the City of Oshawa, the Oshawa Fire Department, and Hibou Systems Inc..

This research is supported by the Queen Elizabeth II Scholarship in Science and Technology. Cette recherche est appuyée par la bourse d'études supérieures en sciences et technologie de la reine Élizabeth II.

[1]S. Lovett (corresponding author), S. Rajan, and C. Rossa are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. E-mail: samuellovett@cmail.carleton.ca; sreeramanr@sce.carleton.ca; rossa@sce.carleton.ca.

[2]T. Paquette and B. Deboon are with Hibou Systems Inc., Springwater, ON, Canada. E-mail: tyler.paquette@hibousystems.com, brayden.deboon@hibousystems.com

motor [9], [10]. A 3D point cloud is generated by transforming the 2D scans into 3D at the angle of the servo motor. In [11] an active mechanical stabilization system is presented ensuring the 2D-LIDAR is kept in the same plane. While all the above are cheaper than a 3D-LIDAR, they are also bulky and add additional weight, making them challenging to use in drone or wearable technology applications.

Another approach to account for OOP motion is Hector SLAM. Hector SLAM transforms 2D LIDAR scans into 3D using the platform's orientation and creates a locally stabilized coordinate frame that changes with the platform's orientation in real-time by using an inertial measurement unit (IMU) [12]. Hector SLAM combines the IMU's linear acceleration and angular velocity using an extended Kalman filter (EKF) to find the platform's orientation, from which a set of 2D scans can be represented in 3D. Scan-matching then occurs in 3D and this set of scans is merged with an existing 2D map through a Gauss-Newton style optimization of scan-map alignments. The 2D map generated from a 2D-LIDAR experiencing OOP motion is computationally expensive and may not be accurate since the locally stabilized coordinate frames are subject to measurement inaccuracies from the IMU.

For pure 2D motion, these inaccuracies impact the location of points within a scan, but subsequent scans are still guaranteed to be within the same coordinate frame. However, the same cannot be said for 3D motion. OOP motion injected into the locally stabilized coordinate frames rotates the coordinate frames with respect to one another, meaning that they are no longer guaranteed to be correlated with the 2D map. This issue of scan-map convergence is clearly stated in [12]. To avoid this issue, scans could be matched and aligned with the map in a globally stabilized coordinate frame.

This paper proposes a real-time method of removing OOP motion from 2D-LIDAR scans generated on platforms experiencing 6-DOF without requiring scan-matching in 3D. The proposed method, henceforth referred to as Level Plane for SLAM (LPS), uses an IMU combined with a magnetometer to determine the platform's orientation using a faster filtering technique than an EKF [13]. In Hector SLAM, the OOP motion is captured via an IMU, transforming the 2D-LIDAR scan into a *locally stabilized* 3D space before scan matching. In contrast, here each scan is projected separately into the same 3D coordinate frame using the measured orientation of each scan and immediately projected onto a *globally stabilized* 2D plane. Scan matching and map alignment are then performed in this globally stabilized 2D plan. The result is a method that is more robust to OOP motion, significantly faster than the time a LIDAR takes to generate a single scan, and leverages additional information from the IMU to reduce the scan-matching/map-alignment complexity from 3-DOF to 2-DOF. To the best of the author's knowledge, the concept of global stabilization of OOP motion has not been introduced before.

The paper is organized as follows: Section II provides an overview of the proposed mathematical framework. Section III details the experimental validation of the LPS algorithm. It is first validated through static mapping in Section III-A

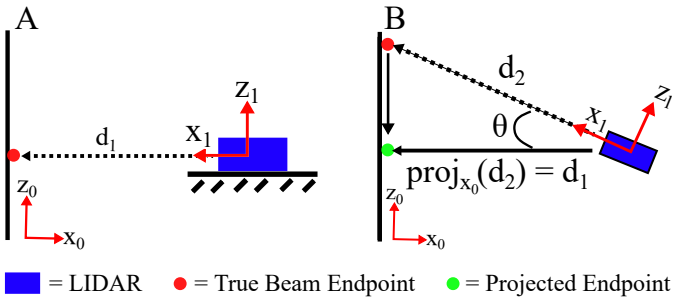

Fig. 1: LIDAR scans A and B are taken OOP of one another. In A, the platform is kept level and returns distance d_1, whereas in B the platform is not level and the measured distance is $d_2 > d_1$. The projection of d_2 by angle θ onto X_0 is d_2 with the OOP motion removed.

to demonstrate its effectiveness at removing the OOP motion component from a single static laser scan. In Section III-B & III-C the LPS is evaluated alongside Hector SLAM in real-world indoor mapping scenarios of varying length with different levels of OOP motion present. The results show that LPS improves robustness to OOP motion compared to Hector SLAM. The LPS software is publicly available[1].

II. IMU-Based Global Stabilization for OOP Motion Compensation

In this paper, a matrix is represented in bold upper case where the superscript defines the parent coordinate frame, and the subscript defines the child coordinate frame. A vector is represented in bold lowercase, where the superscript represents its current coordinate frame. The coordinate frames are defined as follows:

- The world coordinate frame RF_0 has its origin at the starting location of the platform, an x-axis in the positive direction of the platform's motion, and a z-axis pointing out of the ground. A zero in the subscript or superscript indicates this frame.
- The LIDAR coordinate frame RF_1 has its origin at the centre of the LIDAR, and axes inline with the world frame axes. A one in the subscript or superscript indicates this frame.
- The IMU coordinate frame RF_2 has its origin at the centre of the IMU, and coincident with the origin of RF_0, an x-axis inline with the world x-axis, and a z-axis pointing out of the sensor normal to its base. A two in the subscript or superscript indicates this frame.

A 2D-scan ${}^k\mathbf{s}$ viewed in a 3D coordinate frame k is a collection of LIDAR beam end points ${}^k\mathbf{p}_{i=1,\dots,n}$ such that,

$$ {}^k\mathbf{s} = \begin{bmatrix} {}^k\mathbf{p}_i & \dots & {}^k\mathbf{p}_n \end{bmatrix} \tag{1} $$

$$ {}^k\mathbf{p}_i = \begin{bmatrix} {}^kx_i & {}^ky_i & {}^kz_i \end{bmatrix}^T \tag{2} $$

where n is the total number of points in the scan, ${}^kx_i$ and ${}^ky_i$ are the location of the i^{th} point along the k^{th} frame's x

[1]https://github.com/samuelLovett/level-plane-SLAM.git

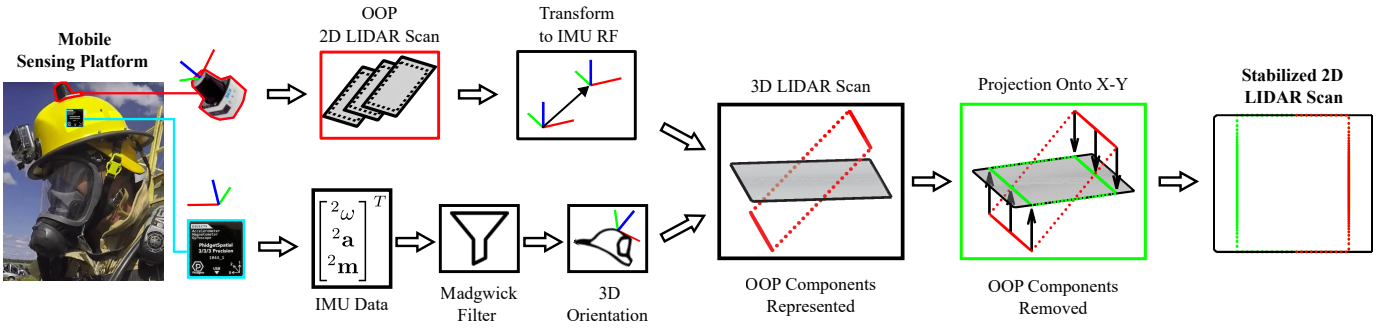

Fig. 2: Data flow of the proposed global stabilization for OOP motion method (LPS). 2D LIDAR scans are combined with the platform's orientation to form a 3D scan, which is projected onto a consistent globally stabilized coordinate frame. Scan matching and map alignment are then performed in the consistent globally stabilized coordinate frame.

axis and y axis, respectively. ${}^{k}z_i = 0 \; \forall \; i$, as demonstrated in Fig. 1 A.

The IMU measurement data is defined as,

$$ {}^{2}\mathbf{U} = \begin{bmatrix} {}^{2}\boldsymbol{\omega} & {}^{2}\boldsymbol{a} & {}^{2}\boldsymbol{m} \end{bmatrix} \tag{3} $$

where ${}^{2}\boldsymbol{\omega}$ is the angular rate of change of the IMU about the RF_2 axes respectively, ${}^{2}\boldsymbol{a}$ is the linear acceleration of the IMU about the RF_2 axes respectively, and ${}^{2}\boldsymbol{m}$ is the magnetic field vector acting on the RF_2 axes respectively. These measurements are then combined to determine the IMU's orientation, allowing the LIDAR scan to later be transformed into the world coordinate frame RF_0.

A. IMU Sensor Fusion

LIDAR-IMU sensor fusion is widely used in SLAM [14]–[16]. Extracting position and orientation from an IMU requires double integration of the acceleration with respect to time, which leads to continuous accumulation of error [17]. To account for this, (3) is filtered using a Madgwick filter [17] to limit the measurement drift by adaptively combining the gyroscope, accelerometer, and magnetometer measurements. It outputs the platform's orientation in quaternions which can then be converted into a set of Euler angles,

$$ {}^{0}\mathbf{\Phi} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{T} \tag{4} $$

where ϕ, θ, ψ are the platform's yaw, pitch, and roll in RF_0, respectively. The Madgwick filter uses less computational memory and has a faster compute time for this process than an EKF [13].

Once (1) has been transformed into the IMU coordinate frame RF_2, (4) allows the scan to be further transformed into RF_0. By transforming (1) to RF_0, the OOP orientation information from (3) is explicitly represented within the scan. The scan is now in 3D with 6-DOF and can be decomposed into the desired and stable $X-Y$ components, Fig. 2.

To perform these transformations, a series of homogeneous frame transformations are conducted. The LIDAR scan is made homogeneous $\left({}^{k}\tilde{\mathbf{S}}\right)$ for mathematical convenience,

$$ {}^{k}\mathbf{s} = \begin{bmatrix} {}^{k}\mathbf{p}_i & \dots & {}^{k}\mathbf{p}_n \end{bmatrix} \Rightarrow {}^{k}\tilde{\mathbf{S}} = \begin{bmatrix} {}^{k}\mathbf{p}_i & \dots & {}^{k}\mathbf{p}_n \\ 1 & 1 & 1 \end{bmatrix} \tag{5} $$

The 4×4 homogeneous transformation matrix ${}^{j}\mathbf{H}_k$ is defined as

$$ {}^{j}\mathbf{H}_k = \begin{bmatrix} & {}^{j}\mathbf{R}_k & & {}^{j}\mathbf{o}_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

where j is the desired coordinate frame, and k is the coordinate frame which contains the LIDAR scan ${}^{k}\mathbf{s}$. The 3×3 rotation matrix ${}^{j}\mathbf{R}_k$ is defined as

$$ {}^{j}\mathbf{R}_k = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\gamma) \tag{7} $$

where $\mathbf{R}_z(\alpha)$, $\mathbf{R}_y(\beta)$, and $\mathbf{R}_x(\gamma)$ are the 3×3 elementary rotations by angles α, β, and γ required to align coordinate frame j to coordinate frame k. Additionally, ${}^{j}\mathbf{o}_k = (t_x, t_y, t_z)^T$ is the translation vector that aligns the origin of coordinate frame j to the origin of coordinate frame k.

With (5) and (6) in homogeneous form, the scan ${}^{k}\tilde{\mathbf{S}}$ can be transformed from coordinate frame k to coordinate frame j by computing,

$$ {}^{j}\tilde{\mathbf{S}} = {}^{j}\mathbf{H}_k \cdot {}^{k}\tilde{\mathbf{S}}. \tag{8} $$

In order to explicitly represent the OOP within the LIDAR scan, the scan must first have the same coordinate frame as the IMU (RF_2). To perform the required transformation from the LIDAR frame to the IMU frame, j and k, are defined as 2, and 1, respectively, in (5), (6), and (7). In (7) α, β, and γ, are defined as the angles required to align RF_1 to RF_2.

With the scan transformed into RF_2, it can further be transformed into RF_0 by repeating the above process with ${}^{2}\tilde{\mathbf{S}}$, $j = 0$, $k = 2$, and (7) as the rotation matrix containing the Euler angles extracted from (4), which represents the platform's orientation when ${}^{1}\mathbf{s}$ was created. The result is a LIDAR scan transformed into RF_0 containing the OOP motion information,

$$ {}^{0}\tilde{\mathbf{S}} = \mathbf{H}_2 \cdot {}^{2}\tilde{\mathbf{S}}. \tag{9} $$

The homogenous portion of (9) is now removed, since its mathematical convenience is no longer needed:

$$ {}^{0}\tilde{\mathbf{S}} = \begin{bmatrix} {}^{0}\mathbf{p}_i & \dots & {}^{0}\mathbf{p}_n \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow {}^{0}\mathbf{s} = \begin{bmatrix} {}^{0}\mathbf{p}_i & \dots & {}^{0}\mathbf{p}_n \end{bmatrix} \tag{10} $$

TABLE I: Comparison of absolute pose error for different levels of synthetic out of plane motion (sOOPM). The range of sOOPM is given in degrees, all other units are in metres.

sOOPM	Algorithm	RMSE	Mean	STD	min / max	SSE
0	Hector	0.01	0.01	0.01	0 / 0.04	0.01
	LPS	0.01	0.01	0.01	0 / 0.02	0
(-5, 5)	Hector	0.25	0.24	0.06	0.12 / 0.31	6.48
	LPS	0.01	0.01	0.01	0 / 0.04	0.01
(-10, 10)	Hector	0.34	0.33	0.09	0.24 / 0.59	10.46
	LPS	0.05	0.04	0.02	0.01 / 0.08	1.13

In the above ${}^0\mathbf{s}$ represents the LIDAR scan within the world frame, whose orientation corresponds to the platform's orientation at the time of the scan's creation, see Fig. 2. It is important to note that ${}^0\mathbf{s}$ is not translated in (9) since the origin of RF_2 is coincident with RF_0.

B. Creating the Globally Stabilized Coordinate Frame

With ${}^k\mathbf{s}$ mapped to the world frame, the ${}^0z$ component of ${}^0\mathbf{p}_i$ is no longer zero. Therefore, in (2), ${}^0z_i \neq 0 \ \forall \ i$, it is now the location of the i^{th} point along the z-axis of the world frame.

By defining the points within a laser scan using a Cartesian coordinate frame in (2), removal of the OOP components is trivial. The projection of ${}^0\mathbf{s}$ onto the $X-Y$ plane is,

$$ {}^0\mathbf{s} = \begin{bmatrix} {}^0\mathbf{p}_i & \dots & {}^0\mathbf{p}_n \end{bmatrix} \tag{11} $$

$$ {}^0\mathbf{p}_i = \begin{bmatrix} {}^0x_i & {}^0y_i & 0 \end{bmatrix}^T \tag{12} $$

By projecting onto the globally stabilized $X-Y$ plane, the z-component containing all the OOP motion caused by the platform's pitch and roll angles (θ, ψ) is removed. The motion contained within the LIDAR scan has now been reduced from 6-DOF to 3-DOF, as standard 2D SLAM algorithms require.

LPS also reduces the scan-matching complexity from 3-DOF to 2-DOF. In standard 2D scan matching, the motion encoded within the LIDAR scan (x-translation, y-translation, and rotation about the LIDAR's z-axis) is relative to the LIDAR coordinate frame. However, the scan matching algorithm must estimate the platform's motion in the world coordinate frame (x-translation, y-translation, and rotation ϕ about the z-axis). This requires the additional step of transforming the relative angular heading of the LIDAR into its absolute heading within the world frame. In the proposed algorithm, ϕ is already known via (4) and encoded within the LIDAR scan via (10). Therefore, the scan matching algorithm only needs to calculate the translations along x and y axes. This improves mapping accuracy and robustness to OOP motion, as shown in the following sections.

III. Experimental Validation

To test the effectiveness of LPS, the experimental setup exhibiting 6-DOF motion shown in Fig. 2 is used. The platform is composed of a firefighting helmet modified to house a 2D LIDAR (SICK Tim 581) and an IMU (PhidgetSpatial Precision 3/3/3 High Resolution). The SICK Tim 581 has a scanning speed of approximately 6.67 ms (15 scans per second), an angular resolution of 0.33° [18], and is located atop the helmet such that it is level when a user looks forward. It communicates over micro-USB using TCP/IP and is powered by an external Lynxmotion 3s 11.1V 4000mAh lithium polymer battery. The IMU is mounted 5 cm below the base of the LIDAR near the user's head to ensure it is close to the centre of rotation. The data from both sensors is processed on a laptop computer using a Linux virtual machine with a 2.30 GHz single core CPU and 4 GB of RAM, on robot operating system (ROS) [19]. Three experiments are conducted: 1) rotated static mapping, 2) small-scale SLAM with platform level, and 3) large-scale SLAM with OOP motion.

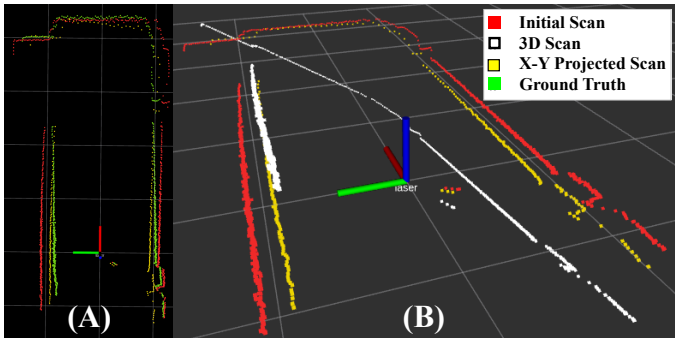

Fig. 3: Rotated Static Mapping experiment. (A) shows a top-down comparison between the initial scan (red), the projected scan (yellow), and the ground truth scan (green). (B) shows this again from an isometric view.

A. Rotated Static Mapping

For the static indoor test, the platform in Fig. 2 is placed in the centre of a hallway at a roll angle of 30°, replicating what is seen in Fig. 1B. The hallway's ground truth measurement is obtained by placing the platform level in the same location and recording a laser scan without LPS implemented.

Fig. 3 overlays the stages a laser scan undergoes as it traverses LPS, first from a top-down perspective in (A) and then from an isometric angle in (B). The isometric angle shows how the initial scan (red lines) is transformed into 3D (white lines) and then projected down onto the globally stable plane (yellow lines). The top-down perspective includes the ground truth scan (green line). The top-down view shows that the initial scan is wider than the projected scan and the ground truth. If not accounted for, this can generate inaccurate maps. By removing the OOP motion, the projected scan is a near-perfect replica in scale of the ground truth scan. The scans do not overlap perfectly because the LIDAR changes pose between the collection of the tilted scan and the ground truth scan.

B. Small-Scale SLAM with Platform Level

The OOP motion component of a scan is caused by the pitch and roll of the platform's orientation given in (4). For the second experiment, LPS was evaluated alongside Hector

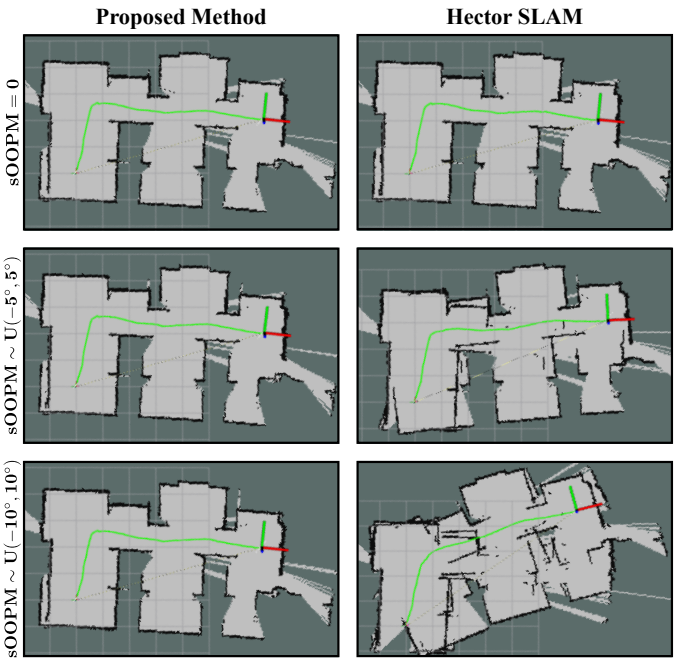

Fig. 4: A comparison of maps generated by Hector SLAM and LPS with increasing levels of synthetic OOP motion (sOOPM) added as shown in the figure. In the third row, during trial three when the randomly sampled values of the sOOPM range from $-10°$ and $10°$ Hector SLAM is no longer able to make a usable map.

SLAM over multiple trials of the same recorded small-scale mapping dataset with different levels of synthetic OOP motion (sOOPM) added to (4). The original dataset has a total path length of 9.641 m, and was recorded by placing the platform shown in Fig. 2, level on a cart, and traversing an office space. The cart ensured that little OOP motion was contained within the dataset, allowing sOOPM to be added to (4) during each trial. The amount of sOOPM added was given by randomly sampling a uniform distribution $sOOPM \sim U(a, b)$ where a and b are the upper and lower bounds, respectively. In total, three trials were conducted with trial one adding no sOOPM, trial two adding $sOOPM \sim U(-5°, 5°)$, and trial three adding $sOOPM \sim U(-10°, 10°)$. After the sOOPM was added to the original dataset, the modified dataset was given to LPS and Hector SLAM to generate two independent maps.

The performance of each algorithm across each trial is compared qualitatively by observing the generated map, Fig. 4, and quantitatively by using the absolute pose error (APE) as is standard [20], see Table I. The APE compares the estimated trajectory provided by Hector SLAM and LPS to the known ground truth trajectory across the entire trajectory. To calculate the APE the EVO evaluation of odometry and SLAM Python package was used [20].

As expected when observing trial one, Fig. 4 row one, where no sOOPM is present, both maps converge to the true map. The results shown in Table I confirm this observation, showing a mean APE of approximately zero for both LPS and Hector SLAM. In trial two the limitations of Hector SLAM start to become apparent with the scan-matcher misaligning the left side of the map in Fig. 4 column Hector SLAM, row two. The APE for this trial reflects the effect of the added sOOPM noise on Hector SLAM, showing an average deviation from the ground truth of 24 cm. On the other hand, LPS shows no change between trial one and trial two. In trial three, Fig. 4 row three, when $sOOPM \sim U(-10°, 10°)$, Hector SLAM can no longer generate an explorable map for path-planning or navigation. However, LPS is hardly impacted and continues to show an accurate map. This observed level of robustness to sOOPM is due to the immediate projection of all OOP motion onto the globally stabilized coordinate frame. Hector SLAM, on the other hand, transforms scans into a locally stabilized coordinate frame, which is not guaranteed to be correlated with the global map.

C. Large-scale SLAM with OOP Motion

For the third experiment, LPS and Hector SLAM were evaluated over one trial of the same real-time indoor SLAM scenario having motion characteristics shown in Fig. 6. The data was collected by having a person wear the platform in Fig. 2 as they navigated a 75m long corridor in 75 seconds. The person was instructed to walk without modifying their gait in an attempt to keep the helmet level; which can be seen when observing the platform's motion in Fig. 6. The jagged and overlapping angular speed curves in Fig. 6 describe just how much rotation and OOP motion occurred while a person was simply walking. If the person had modified their gait to keep the platform level the motion curves would be much smoother showing a more constant rate of change.

Fig. 5 shows the map LPS and Hector SLAM generated over the data containing OOP motion. As expected from the results of Section III-B, and observing the motion characteristics of the platform, Hector SLAM is not able to account for the OOP motion induced by a human's gait cycle, resulting in a map that is not usable for path planning or navigation. LPS significantly improves the robustness of 2D-SLAM to OOP motion generating a usable map by removing the OOP motion within the scans, and aligning them within a globally stabilized coordinate frame.

Throughout this experiment, LPS took an average of 4.76 milliseconds to process a single scan, with a median value of 3.70 milliseconds. The processing time is defined as the difference in time between a scan's initial creation and that scan's projection onto the globally stabilized coordinate frame. The SICK Tim 581 has a scanning speed of approximately 6.67 ms. Thus, the proposed algorithm is well-suited for real-time applications involving dense LIDAR scans.

IV. Conclusion

In 2D-SLAM, most algorithms assume that motion only occurs in a 2D plane. This assumption does not hold for platforms encountering rough terrain, drones, and wearable technology. Past solutions to overcome this problem include the

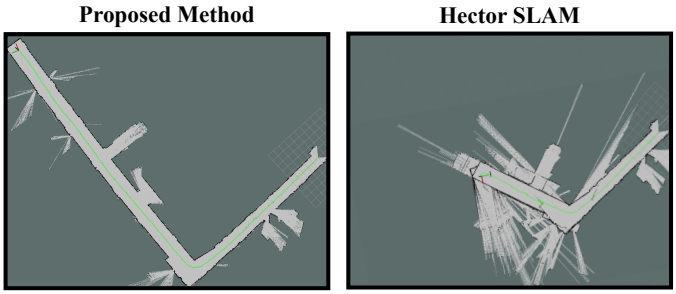

Fig. 5: A comparison of maps generated by LPS and Hector SLAM while navigating a corridor. The dataset is collected from a person walking with the platform on their head. Both maps are created using the same dataset.

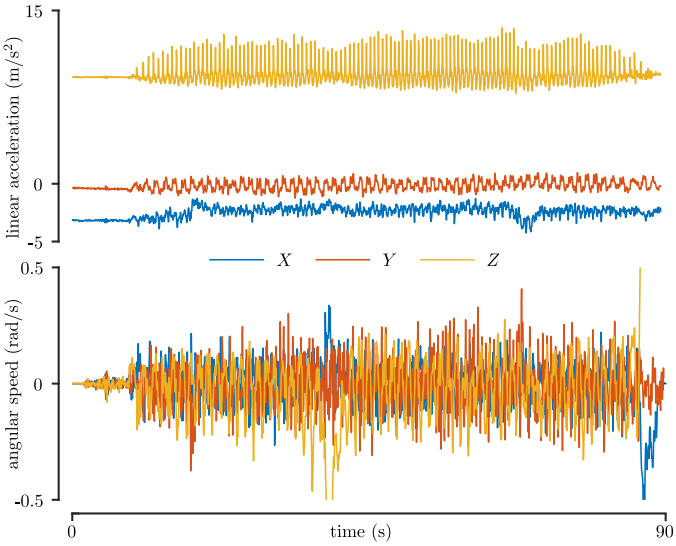

Fig. 6: IMU measurements obtained during the large-scale SLAM with OOP Motion experiment showing the motion experienced by the platform while being worn by the user.

use of 3D-LIDAR, mechanical intervention such as [9], [10], and LIDAR-only odometry, but they are expensive, bulky, and have a high sensitivity to noise [10]. This paper proposes LPS an algorithm that processes 2D laser scans to remove the inaccuracies caused by OOP motion for 2D-SLAM. Using information about the platform's orientation gathered by an IMU, LIDAR scans are transformed into 3D. The 3D scans are then projected onto a globally stabilized 2D coordinate frame. The transformation captures the previously unaccounted-for platform motion, and the projection from 3D to 2D removes these components entirely.

The method is validated in a static and two dynamic mapping scenarios. The results demonstrate that LPS is consistent with using Hector SLAM when no OOP motion is present. However, when OOP motion is present at different levels, LPS outperforms Hector SLAM and is able to maintain good accuracy across the platform's complete trajectory. LPS is fast enough to run in real-time and can be used to improve the robustness of 2D SLAM algorithms subjected to OOP motion.

While LPS improves the robustness of the 2D maps, further improvements can be made. In cases where the platform's pitch or roll cause the LIDAR scan to observe the ground or roof, the projection of these points onto the globally stabilized reference plane will result in false laser endpoints (artifacts) within the scan. These artifacts hamper scan-matching in the same way OOP motion does; causing the scan-matcher to incorrectly align scans and create inaccurate maps, which can no longer be used for platform tracking, path planning, or navigation. Future work involving the use of adaptive filters or deep learning techniques to classify and remove these artifacts would further improve the robustness of 2D-SLAM.

References

[1] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," *Proc. IROS 1991*, pp. 1442–1447 vol.3, 1991.

[2] M. Achmad *et al.*, "A ROS-based human-robot interaction for indoor exploration and mapping," *Int. J. of Adv. and Appl. Sciences*, vol. 3, pp. 88–92, 05 2016.

[3] Z. Chong *et al.*, "Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment," *Proc. - IEEE Int. Conf. on Robot. and Auton.*, pp. 1554–1559, 2013.

[4] A. Khoyani and M. Amini, "A Survey on Visual SLAM Algorithms Compatible for 3D Space Reconstruction and Navigation," in *2023 IEEE Int. Conf. on Consum. Electronics (ICCE)*. IEEE, 2023, pp. 01–06.

[5] N. Mehendale and S. Neoge, "Review on LiDAR technology," *SSRN 3604309*, 2020.

[6] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[7] M. C. Bosse, "ATLAS: a framework for large scale automated mapping and localization," 2004.

[8] P. Biber and W. Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," vol. 3, 2003, pp. 2743–2748.

[9] M. Ocando *et al.*, "Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS," in *2017 Lat. Amer. Robot. Symp. (LARS)*, 2017, pp. 1–6.

[10] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Proc. of Robot.: Sci. and Syst. (RSS '14)*, July 2014.

[11] J. Kim, H. Jeong, and D. Lee, "Single 2D lidar based follow-me of mobile robot on hilly terrains," *J. of Mech. Sci. and Tech.*, vol. 34, no. 9, pp. 3845–3854, Sep 2020.

[12] S. Kohlbrecher *et al.*, "A flexible and scalable SLAM system with full 3D motion estimation," *IEEE Int. Symp. on Saf., Secur., and Rescue Robot.*, pp. 155–160, 2011.

[13] N. Büscher *et al.*, "On the functional and extra-functional properties of imu fusion algorithms for body-worn smart sensors," *Sensors*, vol. 21, no. 8, 2021.

[14] Z. Wang *et al.*, "IMU-Assisted 2D SLAM Method for Low-Texture and Dynamic Environments," *Applied Sciences*, vol. 8, no. 12, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/12/2534

[15] M. Baglietto *et al.*, "Human navigation and mapping with a 6DOF IMU and a laser scanner," *Robotics and Auton. Sys.*, vol. 59, no. 12, pp. 1060–1069, 2011.

[16] C. Debeunne and D. Vivet, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping," *Sensors*, vol. 20, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/7/2068

[17] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," 2010.

[18] SICK AG, *Operating Instructions for TIM5x/56x/57x/58x-2D LiDAR Sensors*, SICK AG, Waldkirch, Germany, 2021. [Online]. Available: https://cdn.sick.com/media/docs/2/02/602/operating_instructions_tim55x_56x_57x_58x_2d_lidar_sensors_en_im0097602.pdf

[19] Q. Morgan *et al.*, "ROS: an open-source Robot Operating System," in *Proc. of the IEEE Intl. Conf. on Robot. and Automat. (ICRA)*, Kobe, Japan, may 2009.

[20] *evo: Python package for the evaluation of odometry and SLAM.* (2017), Michael Grupp [Online]. Available: https://github.com/MichaelGrupp/evo.