IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Pitch Variation Filter for LiDAR-Only SLAM and Localization in Self-Balancing Mobile Robot

Doyeon Kim[1], Heoncheol Lee[1*], *Member, IEEE*, and Ka Hyung Choi[2]

***Abstract*—This paper addresses the Pitch Variation Problem in Two-Wheeled Self-Balancing (TWSB) robots that use 2D LiDAR for Simultaneous Localization And Mapping (SLAM). The issue arises from sudden accelerations or decelerations, leading to abrupt pitch variations that cause the 2D LiDAR to capture data from unintended surfaces, such as the ground or ceiling, destabilizing the robot's position estimation. To mitigate this, we propose a novel preprocessing method that efficiently removes point clusters affected by pitch variation by leveraging their distinct characteristics, without the need for an alignment process. Experimental results demonstrate that our method reduces errors by at least 24.31% across various scan matching algorithms. Furthermore, as the proposed method operates independently of SLAM, it can be seamlessly integrated into a wide range of systems and has been shown to substantially enhance SLAM performance when used alongside existing algorithms.**

***Index Terms*—2D LiDAR, Localization, Mapping, Scan Matching, Two-Wheeled Self-Balancing Robot**

I. Introduction

OVER the past decades, extensive research has been conducted on SLAM [1]–[7], leading to its widespread application in various industrial settings, including logistics and service robots. Among these applications, TWSB robots have gained attention for their compact design, maneuverability, and balance mechanisms [8]–[10]. However, implementing 2D LiDAR-based SLAM on specialized systems such as TWSB robots, where the LiDAR cannot remain stable, necessitates careful consideration of the system's physical dynamics.

For TWSB robots equipped with 2D LiDAR, achieving accurate localization is particularly challenging. These robots frequently undergo abrupt accelerations or decelerations, leading to sudden pitch variations, which we refer to as the *Pitch Variation Problem*. These pitch changes disrupt the scan-matching process, causing the sensor to capture data from unintended surfaces, such as the ground or ceiling, which we describe as pitch variation points, instead of useful environmental features, as shown in Fig. 1.

This work was supported in part by the Institute of Information Communications Technology Planning Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT)(IITP-2026-RS-2020-II201612, 50%), and in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2026-RS-2024-00438430, 50%) supervised by the IITP(Institute for Information Communications Technology Planning Evaluation), and in part by LG Electronics Inc. (Republic of Korea). *(Corresponding author: Heoncheol Lee.)*

[1]Doyeon Kim and Heoncheol Lee are with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea (e-mail: dodeakim@kumoh.ac.kr; hclee@kumoh.ac.kr).

[2]Ka Hyung Choi is with the Appliance Research Laboratory, LG Electronics Inc., Gasan R&D Campus, 51, Gasan digital 1-ro, Geumcheon-gu, Seoul 153-802, Republic of Korea (e-mail: kahyung.choi@lge.com).

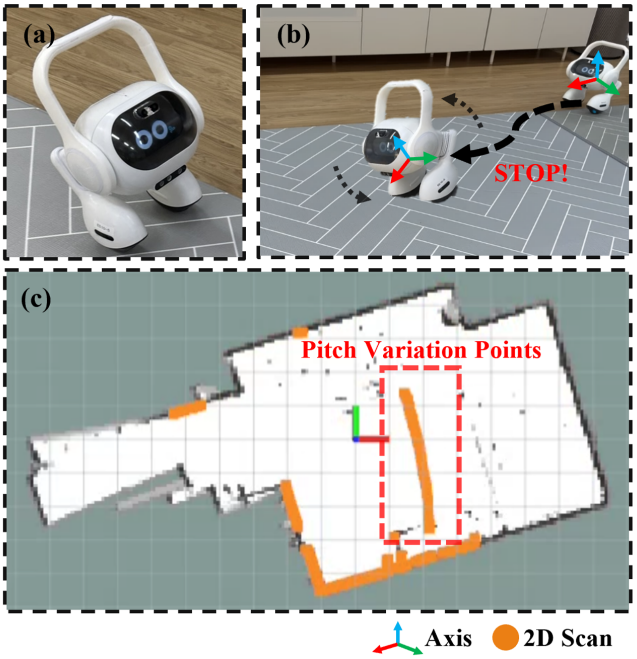

Fig. 1. (a) An example of the TWSB robot. (b) A scenario where the TWSB robot makes a sudden stop, causing the 2D LiDAR to capture the ground. (c) An example of pitch variation points

Due to this limitation, research on localizing TWSB robots using 2D LiDAR has been limited. Most TWSB robots rely on three-dimensional range sensors such as 3D LiDAR or RGB-D cameras [11], [12], which can directly estimate their pitch and handle pitch variation. While these methods enable pitch estimation, they are expensive and require considerable computational resources, making them less suitable for commercial applications. Moreover, vision-based sensors are vulnerable to environmental factors such as changes in illumination and typically have a narrow field of view (FOV), which compromises system stability and reliability.

Given the complexities introduced by the pitch variation problem, a tailored solution is necessary for TWSB robots that use 2D LiDAR. Traditional 2D LiDAR-based SLAM algorithms often assume a stable sensor platform, which does not align with the dynamic nature of TWSB robots. This instability can lead to incorrect localization and mapping, particularly in environments with rapid and unpredictable movements.

In this work, we propose a preprocessing technique for SLAM, called the *Pitch Variation Filter*, to address the pitch variation problem using only a 2D LiDAR. The main contributions of this work are summarized as follows:

- The first preprocessing technique for 2D-LiDAR SLAM in TWSB robots is introduced, which effectively removes pitch variation points from the ground or ceiling to address the pitch variation problem.
- An analysis of the characteristics of the pitch variation problem is conducted, leading to the development of a heuristic method. This method leverages these characteristics to identify clusters affected by pitch variation within the sensor frame without requiring scan-to-scan alignment.
- The proposed method operates efficiently and has demonstrated its effectiveness when applied to various scan matching algorithms. Additionally, since it operates independently of SLAM, it allows for easy integration and has shown improved map consistency.

II. Related Works

In this section, methods for addressing the pitch variation problem are introduced, excluding mechanical methods such as adding stabilizers [13] or supports [14].

A. Motion Compensation

Motion compensation is a technique commonly employed in robotics and SLAM applications to correct for the effects of a robot's movement on sensor data. In the context of TWSB robots using 2D LiDAR, it is particularly useful for addressing the pitch variation problem.

One approach is Hector-SLAM [2], which uses an Extended Kalman Filter (EKF) to estimate the initial pose and project the scan onto the xy-plane. This method effectively compensates for the robot's movements by fusing Inertial Measurement Unit (IMU) data with other sensor inputs, resulting in a consistent 3D solution. Similarly, Level-Plane SLAM [15] employs an IMU with the Madgwick filter to determine the platform's orientation, projecting 2D LiDAR data onto the horizontal plane, and has proven effective when mounted on a helmet in real-world SLAM applications. However, these methods rely on additional sensors, which can introduce challenges such as synchronization issues and unintended latencies.

To avoid the challenges associated with using additional sensors, O. Salem *et al.* [16] introduced a motion compensation method using only 2D LiDAR based on plane-to-plane registration. However, this method performs scan matching during the preprocessing stage, which duplicates the scan matching process that will be performed later in the SLAM stage, and it does not account for orientation in the pitch direction, making it unsuitable for the pitch variation problem.

Although motion compensation methods can help mitigate the pitch variation problem, they often assume that the walls between the ceiling and the floor are flat and do not account for measurements of the floor or ceiling. This assumption breaks down in environments where clutters exist between walls or when the floor or ceiling is measured, leading to inaccurate compensation and preventing a fundamental solution to the problem.

B. Dynamic Object Removal

Another potential solution to the pitch variation problem is to treat the points affected by pitch variations as dynamic obstacles and remove them from the data. This approach can mitigate the adverse effects on SLAM performance and improve the overall accuracy of localization and mapping.

Generally, dynamic object removal methods often utilize 3D LiDAR due to its ability to capture rich spatial information [17]–[19]. In contrast, recognizing objects using 2D LiDAR alone is challenging, necessitating the integration of additional sensors, especially cameras, or the development of solutions tailored to specific targets.

A simple approach, as applied in [20], is to filter LiDAR scans within the bounding boxes provided by vision-based object detection. However, this increases costs and resource requirements, and pitch variations occur independently of specific objects, making this approach unsuitable in such situations.

Another method is to utilize the geometric features of the objects to be detected. For instance, A. Leigh *et al.* [21] proposed a person tracking method using a pre-trained random forest classifier. The 2D LiDAR scan data are classified into human and non-human clusters, allowing the system to distinguish and track individuals. Hanjing Ye *et al.* [22] demonstrated the effectiveness of this person tracker by applying it to SLAM, removing detected individuals to improve overall SLAM accuracy.

Similarly, X. Zhang *et al.* [23] introduced an L-shape fitting method for vehicle detection, which uses the geometric features of 2D LiDAR data to identify and track vehicles in the environment. This method assumes an L-shaped rectangle model and employs the least squares method to determine the best-fitting rectangle without requiring any prior learning.

Although treating pitch variation affected-points as dynamic objects is a promising approach, an effective method for detecting the occurrence of pitch variation and tracking its effects has not yet been developed. Therefore, developing a reliable method for detecting and tracking pitch variations remains an open challenge.

III. Methodology

In this section, the problem definition for addressing the pitch variation problem is briefly introduced, and an overview of the system is provided. Additionally, the operation and interaction of each module within the system are explained.

A. Problem Definition

The objective of addressing the pitch variation problem for TWSB robots is to segment the scan data $z = \{p_i\}_{i=1}^{N}$, captured by a 2D LiDAR sensor, into two distinct subsets: inlier points z_i that correspond to valid environmental features, and outlier points z_o caused by pitch variations. This segmentation is crucial to prevent incorrect associations in the robot's localization process. By effectively segmenting the data, the robot's perception system can maintain an accurate representation of the environment despite the problems introduced by pitch variations.

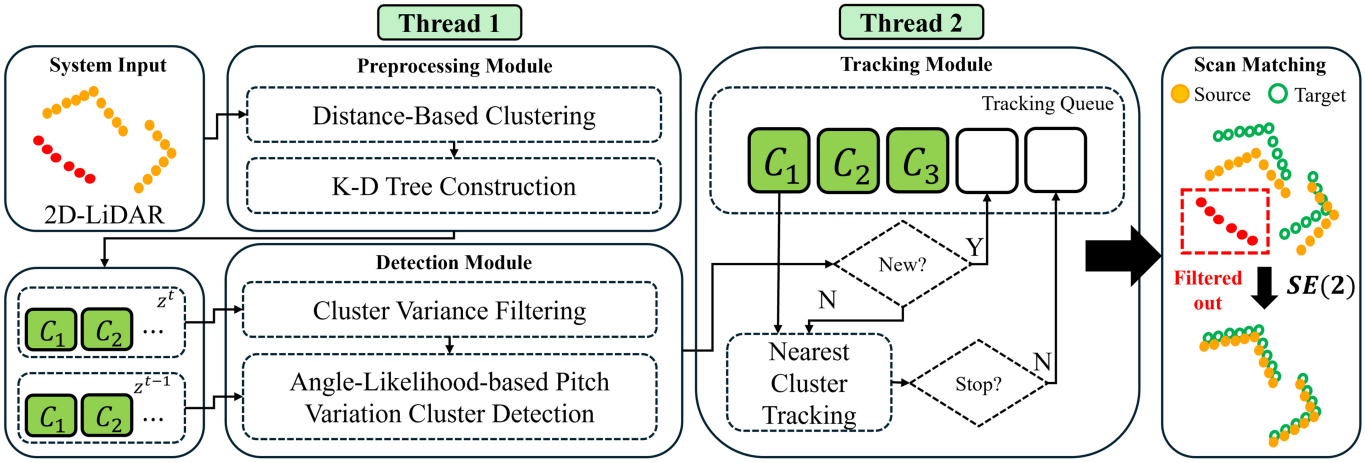

Fig. 2. The proposed method processes 2D scans containing points partially affected by pitch variation (represented as red dots) and outputs scans with these points filtered out. The system operates through three modules: preprocessing, detection, and tracking, running across two separate threads. In the output scan, points influenced by pitch variation (highlighted in the red box) are removed, resulting in cleaner scan matching between the source scans (yellow dots) and target scans (green dots).

B. System overview

The overview of our proposed method is illustrated in Fig. 2. The proposed method consists of a preprocessing module, a detection module, and a tracking module. The preprocessing and detection modules run in a single thread, while the tracking module is executed in a separate thread, with the entire system operating as a single process. All modules in our pipeline operate directly in the sensor frame without requiring any additional alignment.

Initially, the raw 2D LiDAR scan is fed into the preprocessing module, where distance-based clustering is performed, and a K-D Tree is built to prepare the data for subsequent use in the detection module.

Once the preprocessing results for the time step $t-1$ and t are ready, the detection module identifies clusters where pitch variation has occurred. The identified clusters are inserted into a tracking queue, where they are continuously tracked. Clusters that remain in the tracking queue are eventually removed, resulting in an output scan from which clusters affected by pitch variation have been eliminated.

C. Preprocessing Module

Since determining whether each individual point has been affected by pitch variation is difficult, it is assumed that points influenced by pitch variation share similar geometric characteristics. Based on this assumption, clustering is performed to generate seed clusters.

To generate a set of clusters $S_C^t = \{C_1, \ldots, C_i\}$ from the scan z^t at time t, a distance-based clustering method is applied, starting from the first index of z^t. Using Breadth-First Search (BFS), points within a predefined d_{th} are grouped together. If this condition is not met, a new cluster starting point is selected. Finally, clusters with a size larger than m_{th} are considered valid clusters. After clustering, a K-D Tree T_{kd}^t is constructed for z^t to efficiently compute the likelihood of clusters in z^{t+1}. This enables fast nearest neighbor queries between the current clusters and the previous scan, allowing for likelihood evaluation without explicitly constructing a likelihood field on a grid.

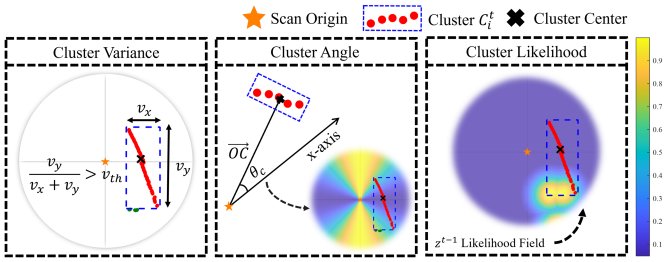

Fig. 3. Illustration of the key characteristics observed during pitch variation between consecutive LiDAR scans.

D. Detection Module

To detect pitch variation clusters among the seed clusters, a heuristic method is employed. These clusters typically capture the ground or ceiling during rapid acceleration or deceleration and exhibit several distinct characteristics, as illustrated in Fig. 3.

1) Cluster Variance: Pitch variation clusters tend to be distributed along a perpendicular axis to the sensor's direction of movement. To quantify this property, we compute the perpendicular variance ratio v_C as follows:

$$v_C = \frac{v_y}{v_x + v_y} \tag{1}$$

where v_x and v_y represent the variances of the cluster points along the x- and y-axes, respectively, assuming the sensor's forward direction corresponds to the x-axis. Clusters with v_C lower than a predefined threshold v_{th} are excluded to reduce the number of candidate clusters, improving computational efficiency.

2) Cluster Angle: The angle θ_C between the sensor origin and the centroid of the cluster is also considered. Pitch variation clusters typically appear either directly in front of or directly behind the sensor, resulting in angles close to 0° or 180°.

To quantify this property, we define the angular confidence as:

$$P(\theta_C) = \exp\left(-\frac{\cos^2(\theta_C)}{2\sigma_\theta^2}\right) \tag{2}$$

where σ_θ denotes the standard deviation of the angular distribution.

3) Cluster Likelihood: Pitch variation clusters often appear in different locations across consecutive scans due to abrupt pitch changes, resulting in low temporal consistency when evaluated within the likelihood field [24] generated from z^{t-1}.

To measure temporal consistency, we assess how well the current cluster aligns with the previous scan z^{t-1} using a likelihood field. For each point p_i^t in cluster C^t, the likelihood is computed as:

$$L(p_i^t \in C^t \mid z^{t-1}) = \exp\left(-\frac{d(p_i^t, \mathrm{NN}(p_i^t, z^{t-1}))^2}{2\sigma_l^2}\right) \tag{3}$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance between p_i^t and its nearest neighbor in z^{t-1}, and σ_l is the standard deviation of the spatial likelihood field. The overall likelihood of the cluster is computed as:

$$P(L_C) = \frac{1}{N}\sum_{i=1}^{N} L(p_i^t \in C^t \mid z^{t-1}) \tag{4}$$

where N is the number of points in C^t.

4) Final Pitch Variation Detection: Since $P(\theta_C)$ and $P(L_C)$ represent independent directional and temporal cues, we combine them multiplicatively. A high product value indicates that the cluster has both a lateral orientation and temporal correspondence with the previous scan, implying a persistent environmental feature. To detect pitch variation clusters, we compute the complement of this product:

$$P(C_{pv}) = 1 - P(\theta_C)\cdot P(L_C) \tag{5}$$

If $P(C_{pv})$ exceeds a predefined threshold P_{th}, the cluster is classified as a pitch variation cluster and added to the tracking queue.

The detailed procedure is described in Algorithm 1.

Algorithm 1: Detection Module

Input: Set of clusters S_C^t, previous clusters S_C^{t-1}, K-D Tree T_{kd}^{t-1}
Output: Set of pitch variation clusters S_{pv}^t

$S_{pv}^t \leftarrow \emptyset$
for $C_i^t \in S_C^t$ **do**
  Extract (x_C, y_C, θ_C), v_C, and N from C_i^t
  if $v_C < v_{th}$
    continue
  $P(\theta_C) \leftarrow \exp\left(-\frac{\cos^2(\theta_C)}{2\sigma_\theta^2}\right)$
  $P(L_C) \leftarrow \frac{1}{N}\sum_{j=1}^{N}\exp\left(-\frac{d(p_j^t, \mathrm{NN}(p_j^t, z^{t-1}))^2}{2\sigma_l^2}\right)$
  $P(C_{pv}) \leftarrow 1 - P(\theta_C)\cdot P(L_C)$
  if $P(C_{pv}) > P_{th}$
    $S_{pv}^t \leftarrow S_{pv}^t \cup \{C_i^t\}$
return S_{pv}^t

Algorithm 2: Tracking Module

Input: Set of clusters S_C^t, pitch variation clusters S_{pv}^t, tracking queue Q^{t-1}
Output: Updated tracking queue Q^t

Initialize $Q^t \leftarrow \emptyset$
for $C_i^t \in S_C^t$ **do**
  Extract (x_C, y_C, θ_C), v_C, and N from C_i^t
  $C_{\text{nearest}}^{t-1} \leftarrow$ `find_nearest`(C_i^t, Q^{t-1})
  if `compute_distance`$(C_i^t, C_{nearest}^{t-1}) > d_{track}$
    continue
  else
    if $v_C < v_{th}$
      continue
    if `compute_distance`$(0, C_i^t) \notin [d_{min}, d_{max}]$
      continue
    if $\left|$`size`$(C_i^t) -$ `size`$(C_{nearest}^{t-1})\right| > \Delta m_{th}$
      continue
    $Q^t \leftarrow Q^t \cup C_i^t$
$Q^t \leftarrow Q^t \cup \left\{C_j^t \mid C_j^t \in S_{pv}^t \wedge C_j^t \notin Q^t\right\}$
return Q^t

E. Tracking Module

Since our detection module relies on the likelihood field constructed at time step $t-1$, it effectively detects pitch variation clusters in the first frame where pitch change occurs. However, in subsequent frames, even if the pitch variation persists, the likelihood field may no longer be sufficient to detect the affected clusters. In addition, conventional probabilistic tracking is not feasible in this context because pitch variation clusters are not actual moving objects but artifacts generated by sudden pitch changes. These clusters often exhibit near-linear shapes but continuously deform and shift unpredictably until the TWSB robot returns to a stable state.

To address this issue, we adopt a nearest cluster tracking approach. For each new seed cluster $C_i^t \in S_C^t$, the nearest cluster in the previous tracking queue Q^{t-1} is identified based on centroid distance. If the distance is within d_{track}, the clusters are temporally associated; otherwise, the association is rejected. After this nearest neighbor association, additional checks are performed to prevent false matches.

1) Perpendicular Variance Check: Unlike the cluster likelihood, the perpendicular variance ratio remains consistent for pitch variation clusters throughout tracking. Therefore, we reuse the same threshold as in the detection module and exclude clusters with $v_C < v_{\text{th}}$ from tracking.

2) Radial Distance Check: The cluster centroid is verified to fall within the physically plausible radial distance range

$[d_{\min}, d_{\max}]$. This constraint reflects the effect of sensor pitch variation, where points from the ground or ceiling are unintentionally captured and tend to appear in specific radial bands. Although this constraint could be used during detection, we intentionally apply it in the tracking phase to conservatively refine the candidates and correct potential false positives.

3) Size Consistency Check: The number of points in the current cluster is compared with the associated previous cluster. If the size difference exceeds the threshold Δm_{th}, the cluster is considered unreliable and removed from the tracking queue.

Unassociated pitch variation clusters detected in the current frame are directly added to the tracking queue. By continuously applying this process, our method tracks and removes pitch variation clusters across frames, producing a corrected scan where such clusters are eliminated.

The details of this process are described in Algorithm 2.

IV. Experimental Results

A. Experiment Setup

To evaluate the performance of the proposed pitch variation filter, three evaluation metrics were used: Precision-Recall Curve (PR Curve), Scan Matching Error, and Absolute Trajectory Error (APE). The details of each evaluation are explained in the following sections. To ensure consistency across all tests, the experiments were conducted using the LG Q9 robot (Qualcomm RB5) equipped with a SLAMTEC C1 LiDAR, following a single parameter configuration as listed in Table I. Due to the LiDAR's mounting position near the robot's leg, only 240° of its 360° FOV captures meaningful data from the surrounding environment.

TABLE I
Parameter Configuration

Module	Parameter	Value
Preprocessing	d_{th}	0.3
	m_{th}	10
Detection	v_{th}	0.65
	σ_θ, σ_l	1.0
	P_{th}	0.8
Tracking	d_{track}	1.5
	d_{min}, d_{max}	[0.3, 2]
	Δm_{th}	$(\frac{1}{2}m,\ 2m)$

B. Parameter Tuning

We explain the key parameters that significantly affect the system's behavior. All parameters were empirically selected to balance detection sensitivity and robustness. In our hardware configuration, the robot is equipped with a low-cost 2D LiDAR sensor, which provides relatively low angular resolution (2 points per degree and 10 Hz) and includes measurement noise. To prevent noise-induced clusters and to ensure meaningful cluster formation, a minimum cluster size threshold m_{th} is used. This value can be set higher when using high-resolution LiDARs.

In particular, v_{th}, σ_θ, and σ_l are tuned according to the TWSB robot's control dynamics. When the robot operates at higher speeds and undergoes larger pitch variations, the resulting clusters exhibit clearer pitch variation characteristics. In this case, v_{th} can be set to a higher value to suppress false positives while still reliably detecting pitch variation clusters.

On the other hand, σ_θ and σ_l control the system's sensitivity. Lower values make the system more responsive to small geometric deviations, but excessively small values may cause overreaction to minor disturbances or noise. Conversely, larger values improve robustness against noise but may cause the system to miss true pitch variation clusters.

These characteristics may vary depending on the operational environment. For example, in environments with carpets or frequent height discontinuities, larger pitch variations tend to occur due to sudden tilting of the robot. In such cases, higher values of v_{th}, σ_θ, and σ_l are recommended to robustly distinguish pitch variation clusters from static environmental structures.

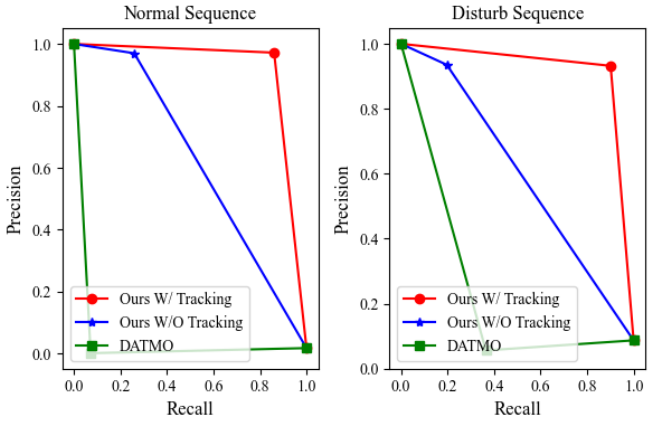

Fig. 4. Precision-Recall curves of the proposed method, with and without the tracking module, and the DATMO baseline. The left column corresponds to the Normal Sequence, while the right column shows results from the Disturb Sequence.

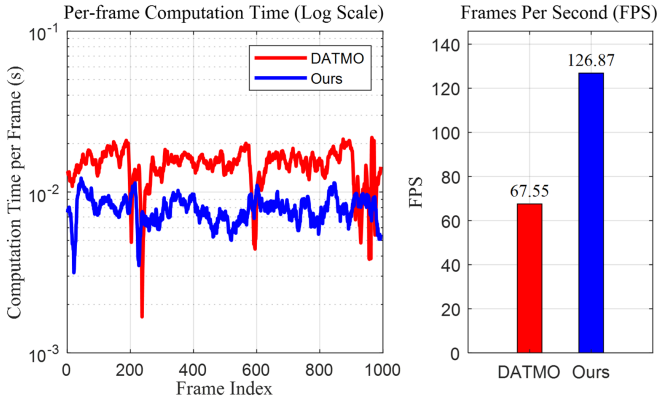

Fig. 5. Computation time analysis. The left graph shows the per-frame computation time in log scale, while the right graph illustrates the overall system throughput in terms of FPS.

C. Precision-Recall Curve and Time Computation

Since pitch variations occur in only a small portion of the entire dataset, we used the PR Curve as an evaluation metric

to ensure a fair and accurate assessment. For the PR Curve, two self-collected and labeled datasets were used to determine whether the proposed method effectively removed data from the ground or ceiling when pitch variation occurred. The first dataset, named the "normal sequence," contains typical robot movements such as moving forward, backward, and turning, collected from a TWSB robot. The second dataset, named the "disturb sequence," includes both normal movements and situations where external disturbances, such as human-induced shaking, occurred.

To compare our method with existing dynamic object removal techniques, we conducted an evaluation against DATMO[1] (Detection and Tracking of Moving Objects), which utilizes a Kalman Filter to track moving objects over time. In the DATMO implementation, objects with an estimated velocity greater than 0.1 m/s (best-performing among 0.1, 0.2, and 0.3 m/s) were considered dynamic and removed from the scan. Since DATMO operates in the world frame, we provided SLAM-based poses (from the same pseudo-ground-truth trajectory described in Section IV-E).

As shown in Fig. 4, the proposed method demonstrated reliable performance in removing pitch variation clusters, achieving an AUC of at least 0.92. Additionally, it was observed that the combined use of the detection and tracking modules resulted in superior performance compared to using the detection module alone. In contrast, DATMO exhibited substantially lower PR curve performance, with the best AUC being only 0.24 in the context of the pitch variation problem. This is because conventional dynamic object removal methods are designed to filter out actual dynamic objects, whereas pitch variation clusters are not real objects but artifacts resulting from sudden changes in the robot's pitch angle. These clusters appear abruptly and deform irregularly, making them difficult to handle with general-purpose dynamic object tracking methods. Furthermore, velocity-based filtering is difficult to tune and prone to false positives when applied to pitch variation clusters.

In terms of computational efficiency, our proposed method achieved 126.87 Hz, as shown in Fig. 5, while DATMO operated at 67.55 Hz, making our method approximately 1.9 times faster in processing speed. Note that this timing excludes the coordinate alignment time required for DATMO; including this overhead would further widen the gap. This faster processing highlights the practical applicability of our method for real-time deployment on commercial platforms, where both accuracy and computational speed are critical.

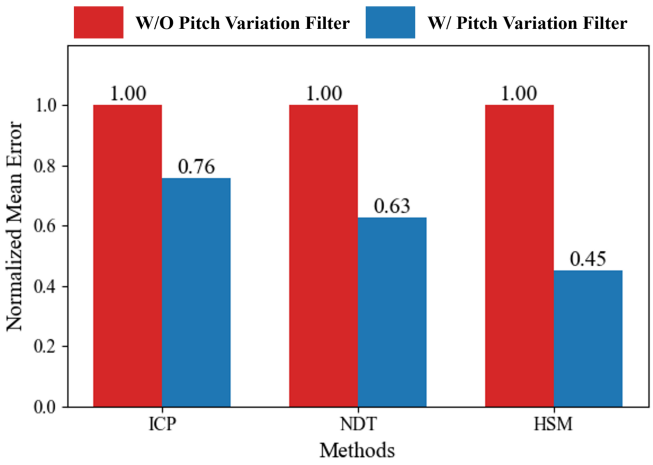

Fig. 6. Comparison of Normalized Mean Chamfer Distance in Scan Matching Methods with and without Pitch Variation Filter. The red bars represent the error without using the Pitch Variation Filter, while the blue bars show the error when the filter is applied.

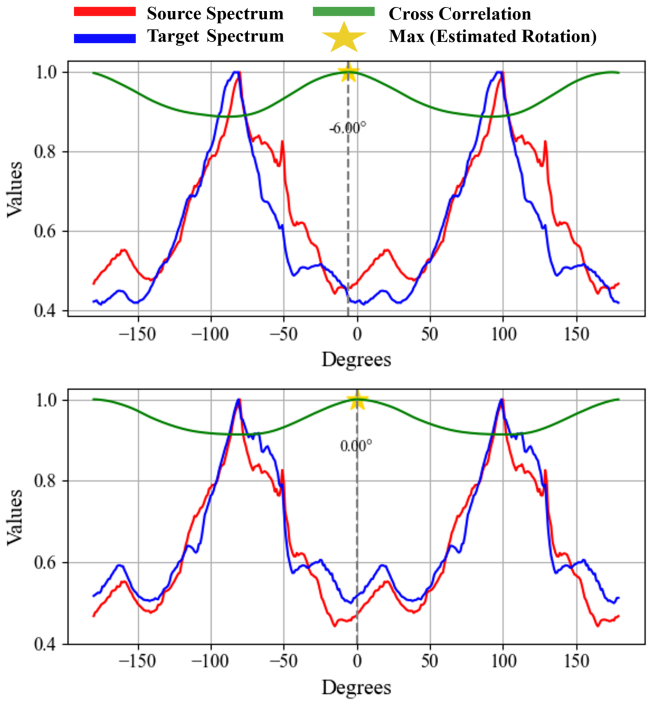

Fig. 7. Comparison of cross-correlation in HSM with and without the Pitch Variation Filter. Without the filter, the estimated rotation is -6°, while with the filter, the rotation is correctly estimated as 0°.

D. Scan Matching Error

To evaluate scan matching errors, we compared the performance of Iterative Closest Point (ICP) [25], Normal Distribution Transform (NDT) [26], and Hough Scan Matcher (HSM) [27]. After each method performed scan alignment between the frame before pitch variation occurred and the frame where it occurred, the Chamfer Distance (CD) between the two scans was calculated using the following equation:

$$CD(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} |x-y|_2^2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} |x-y|_2^2 \tag{6}$$

As shown in Fig. 6, the proposed pitch variation filter substantially reduced scan matching errors across all methods, achieving error reductions of 24.31% for ICP, 37.10% for NDT, and 54.69% for HSM compared to cases where the pitch variation filter was not applied. This highlights the robustness and effectiveness of the proposed approach.

[1]https://github.com/kostaskonkk/datmo

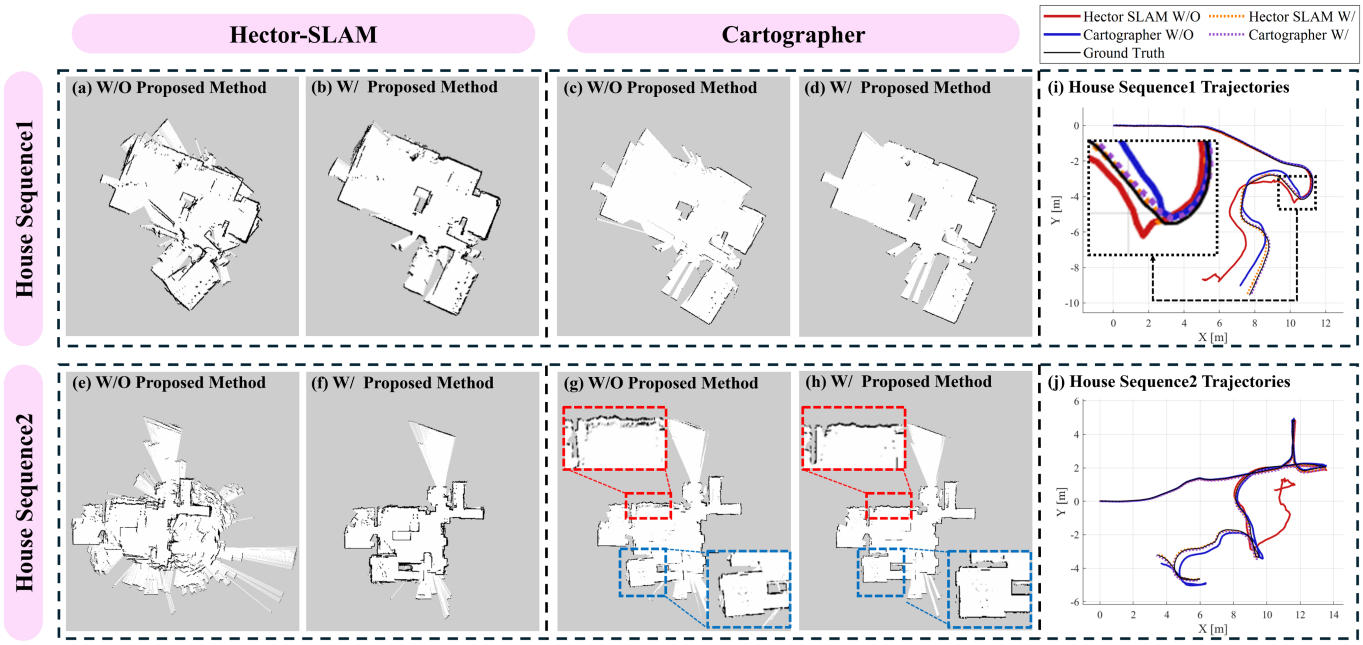

Fig. 8. Mapping results and trajectory comparisons with and without the pitch variation filter are presented. (a) and (b) show Hector-SLAM without and with the proposed method, respectively, while (c) and (d) show Cartographer without and with the method. Similarly, (e) and (f) present Hector-SLAM without and with the method, and (g) and (h) show Cartographer without and with the method. (i) illustrates the trajectories corresponding to (a) through (d), and (j) shows the trajectories corresponding to (e) through (h). The black box indicates the location where pitch variation occurred and highlights the resulting trajectory differences. The red box denotes a duplicated wall caused by pitch variation. The blue box highlights the distortion in the map due to rotational error introduced by pitch variation.

Pitch variation clusters typically appear as straight or nearly straight lines, which tend to degrade the performance of HSM due to its reliance on the Hough Transform. As illustrated in Fig. 7, the presence of a pitch variation cluster introduces noise into the Hough spectrum, resulting in inaccurate rotation estimations. Consequently, the removal of these clusters leads to the most substantial error reduction in HSM compared to the other evaluated methods.

TABLE II
ABSOLUTE TRAJECTORY ERROR (ATE) FOR EACH SLAM METHOD WITH AND WITHOUT THE PROPOSED METHOD

Sequence	Hector-SLAM		Cartographer	
	w/o	w/	w/o	w/
House1	1.0522	**0.1903**	0.3551	**0.1101**
House2	–	**0.0779**	0.2106	**0.0964**

"–" denotes pose tracking failure.

E. SLAM Integration

Furthermore, to observe how our proposed method influences mapping quality and trajectory when applied to SLAM, we integrated it into the open-source SLAM systems Hector-SLAM [2] and Cartographer [3]. The experiments were conducted in a real-world residential environment, referred to as the "House sequence." In these experiments, the robot moved at a maximum linear velocity of 1.0 m/s and an angular velocity of 0.6 rad/s. "House sequence 1" included a single pitch variation, whereas "House sequence 2" featured multiple pitch variations. In addition to qualitative improvements, we quantitatively evaluated the SLAM performance using APE with a pseudo-ground-truth trajectory. This pseudo-ground-truth was generated using Cartographer, configured with carefully fine-tuned parameters to fuse LiDAR, wheel odometry, and IMU data. The results are summarized in Table II.

$$ATE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i^{est} - \mathbf{p}_i^{gt}\|^2} \tag{7}$$

For "House sequence 1," the proposed method reduced the APE from 1.0522 m to 0.1903 m in Hector-SLAM, and from 0.3551 m to 0.1101 m in Cartographer. These numerical improvements correspond to the qualitative mapping results shown in Fig. 8 (a)-(d). Without the proposed method, (a) and (c) exhibited significant map inconsistencies at the points of pitch variation. In contrast, with the proposed method applied, as shown in (b) and (d), the system successfully handled the pitch variation and completed the mapping process.

For "House sequence 2," which involved multiple pitch variations, Hector-SLAM without the proposed method failed in pose tracking, while applying our method resulted in an APE of 0.0779 m. The APE of Cartographer also decreased from 0.2106 m to 0.0964 m with the proposed method. These improvements are visually evident in Fig. 8 (e)–(h). Without the proposed method, (e) failed to achieve consistent mapping due to repeated pitch variations. In contrast, (f), where the pitch variation filter was applied, successfully maintained stable mapping. Meanwhile, (g), which used Cartographer without the proposed method, achieved partial success in mapping thanks to Cartographer's hierarchical scan matching

structure, which incorporates the Correlative Scan Matcher and the Ceres Scan Matcher. This structure provides a prior for scan matching and enables relatively stable mapping even without odometry or IMU data. However, it still exhibited minor mapping inconsistencies, such as duplicated walls and slight structural distortions. In contrast, (h), where the proposed method was applied, demonstrated superior mapping quality, clearly proving the effectiveness of the proposed method.

These results highlight that the proposed filter is particularly well-suited for environments with frequent pitch variations, especially those caused by abrupt acceleration or deceleration. Nevertheless, a known limitation of the method is its potential misclassification of dynamic objects—such as humans or moving furniture—as pitch variation clusters, when they abruptly enter the LiDAR field of view from the front or rear. While these objects are not the intended targets of our method, such misclassifications may interfere with other modules that rely on accurate detection of dynamic entities, such as object tracking or obstacle avoidance.

V. Conclusions

In this paper, we propose a novel preprocessing method to address the pitch variation problem encountered by TWSB robots. The proposed method leverages the distinct characteristics of pitch variation clusters, enabling efficient cluster removal without the need for an alignment process. Additionally, it achieves an onboard processing speed of 126.87 Hz and a minimum AUC of 92%. When applied to scan matching algorithms such as ICP, NDT, and HSM, the method demonstrated error reductions of 24.31%, 37.10%, and 54.69%, respectively, showing improved trajectory results when integrated with SLAM algorithms.

Currently, we have focused on the pitch direction, which has the most significant impact on TWSB robots, but in future research, we plan to also consider the roll and yaw directions.

References

[1] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[2] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 155–160, 2011.

[3] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, 2016.

[4] 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, 2018.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[6] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, 2019.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[8] F. Grasser, A. D'arrigo, S. Colombi, and A. C. Rufer, "Joe: a mobile, inverted pendulum," *IEEE Transactions on industrial electronics*, vol. 49, no. 1, pp. 107–114, 2002.

[9] J.-X. Xu, Z.-Q. Guo, and T. H. Lee, "Design and implementation of integral sliding-mode control on an underactuated two-wheeled mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3671–3681, 2014.

[10] V. Klemm, A. Morra, C. Salzmann, F. Tschopp, K. Bodie, L. Gulich, N. Küng, D. Mannhart, C. Pfister, M. Vierneisel, F. Weber, R. Deuber, and R. Siegwart, "Ascento: A two-wheeled jumping robot," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7515–7521, 2019.

[11] G. Goronzy, M. Pelka, and H. Hellbrück, "Qrpos: Indoor positioning system for self-balancing robots based on qr codes," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2016.

[12] S. Jing, Z. Ma, K. Yi, H. Li, and Z. Li, "Pose optimization of a single ball self-balancing mobile robot based on imu and visual fusion," in *2018 37th Chinese Control Conference (CCC)*, pp. 2436–2441, 2018.

[13] M. Clark and R. C. Roberts, "Autonomous quadrotor terrain-following with a laser rangefinder and gimbal system," in *2017 IEEE SENSORS*, pp. 1–3, 2017.

[14] J. Zhao, J. Li, and J. Zhou, "Research on two-round self-balancing robot slam based on the gmapping algorithm," *Sensors*, vol. 23, no. 5, 2023.

[15] S. Lovett, T. Paquette, B. DeBoon, S. Rajan, and C. Rossa, "Level plane slam: Out-of-plane motion compensation in a globally stabilized coordinate frame for 2d slam," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3355–3360, 2023.

[16] O. A. A. K. Salem, E. Giacomini, L. Brizi, L. Di Giammarino, and G. Grisetti, "Enhancing lidar performance: Robust de-skewing exclusively relying on range measurements," in *International Conference of the Italian Association for Artificial Intelligence*, pp. 310–320, Springer, 2023.

[17] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6529–6536, 2021.

[18] L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, and R. Siegwart, "Dynablox: Real-time detection of diverse dynamic objects in complex environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6259–6266, 2023.

[19] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from lidar point streams," *Nature Communications*, vol. 15, p. 345, Jan 2024.

[20] N. Zimmerman, T. Guadagnino, X. Chen, J. Behley, and C. Stachniss, "Long-term localization using semantic cues in floor plan maps," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 176–183, 2023.

[21] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 726–733, 2015.

[22] H. Ye, G. Chen, W. Chen, L. He, Y. Guan, and H. Zhang, "Mapping while following: 2d lidar slam in indoor dynamic environments with a person tracker," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 826–832, 2021.

[23] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient l-shape fitting for vehicle detection using laser scanners," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 54–59, 2017.

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[25] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[26] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, pp. 2743–2748 vol.3, 2003.

[27] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the hough domain," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2739–2744, 2005.