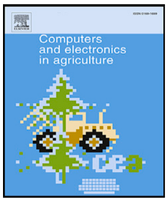
Computers and Electronics in Agriculture 247 (2026) 111687

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Original papers

# Consistent lidar-only SLAM for legged agricultural robots in arboreal environments via robust dimensionality reduction

Paola Nazate-Burgos [a,*], Miguel Torres-Torriti [a,b], Shoudong Huang [c], Fernando Auat Cheein [d,b]

[a] Dept. of Electrical Engineering, School of Engineering, Pontificia Universidad Católica de, Chile
[b] Advanced Center for Electronics and Electrical Engineering, Chile
[c] Robotics Institute, Faculty of Engineering and IT, University of Technology Sydney, Australia
[d] Department of Engineering, Harper Adams University, United Kingdom

## ARTICLE INFO

Dataset link: https://github.com/RAL-UC/RoSA_SLAM, https://github.com/RAL-UC/Pullally_Dataset

*Keywords:*
SLAM
Lidar
Agricultural robotics
Legged robots
Arboreal environments
Dimensionality reduction
Robust distance metrics

## ABSTRACT

Simultaneous localization and mapping (SLAM) in arboreal environments presents unique challenges due to dense foliage, uneven terrain, and GNSS signal degradation. While state-of-the-art 3D lidar SLAM methods exist, they typically require IMU integration and assume smooth platform motion with peak-accuracy as a primary goal, making them unsuitable for legged robots on irregular terrain or when inertial sensors are unreliable. This paper presents a SLAM approach whose goal is consistent localization rather than peak accuracy for robots with non-smooth motion in arboreal environments via dimensionality reduction. The method requires spatially persistent vertical structures like tree trunks. We adapt the modified Hausdorff distance for scan-to-map matching of a 2D projection of 3D lidar data slices as a robust distance metric, eliminating the need for complex feature extraction, IMU integration, or GNSS corrections. This approach is particularly suited for legged robots and handheld mapping systems where discontinuous motion invalidates assumptions of existing methods. The method is less suitable when distinctive vertical structures are absent, such as in very young orchards, espalier systems, or very sparse plantations.

We evaluate our approach across simulated environments, controlled field tests, and three real-world datasets (CitrusFarm, Bacchus, and Pullally), including a novel dataset collected with a quadruped robot. Our comprehensive evaluation results demonstrate that when IMU data is unavailable or unreliable due to impulsive platform motions, the proposed approach maintains positioning accuracy ranging from 0.40 m to 1.34 m across different environments, with particularly robust performance on the challenging Pullally dataset featuring a legged robot (0.40 ± 0.05 m over three trials), while methods designed for IMU integration and dense 3D point cloud matching either achieve superior performance on smooth terrains with wheeled robots (0.16–0.28 m on CitrusFarm) or fail catastrophically on legged platforms (>27 m on Pullally). The proposed method is compared against A-LOAM, LeGO-LOAM, DLO, MOLA and AG-LOAM. Results show that our approach trades peak performance for consistent robustness across datasets and platform types. We release our code and the Pullally dataset to support further research in agricultural robotics: https://github.com/RAL-UC/RoSA_SLAM.

## 1. Introduction

The agricultural sector is undergoing a rapid transformation through the integration of smart farming technologies, with autonomous systems playing a pivotal role in addressing challenges associated with labor shortage, rising labor costs, minimizing environmental impact, as well ensuring economic sustainability for producers (Mahto et al., 2024; Torres-Torriti and Nazate-Burgos, 2022). One important barrier for automating basic agricultural tasks and deploying mobile robots in forestry and fruit tree orchards is the difficulty of implementing autonomous robot navigation strategies because most of the existing simultaneous localization and mapping (SLAM) algorithms have been conceived and optimized for structured indoor or urban environments. The recent literature (Aguiar et al., 2022; Debeunne and Vivet, 2020; Liu et al., 2022; Li and Zhu, 2024) has recognized the applicability of SLAM techniques to the agricultural domain and the

---

* Corresponding author.

*E-mail addresses:* pjnazateo@uc.cl (P. Nazate-Burgos), mtorreto@uc.cl (M. Torres-Torriti), shoudong.huang@uts.edu.au (S. Huang), fauat@harper-adams.ac.uk (F. Auat Cheein).

https://doi.org/10.1016/j.compag.2026.111687
Received 10 December 2025; Received in revised form 16 March 2026; Accepted 17 March 2026

necessity to enhance current SLAM strategies, especially when employed in arboreal environments (Sier et al., 2023; Wang et al., 2024). Environments such as fruit orchards and forest plantations introduce unique constraints, –including uneven terrain, dense and overlapping vegetation, seasonal appearance changes, and highly repetitive structures (Cremona et al., 2022; Debeunne and Vivet, 2020)–, that limit the performance of visual or feature-based methods. Moreover, Global Navigation Satellite System (GNSS) signals are frequently degraded or blocked entirely by foliage and canopy cover, rendering high-precision positioning unreliable. In addition, many SLAM strategies for indoor or urban applications rely heavily on Inertial Measurement Units (IMUs) to compensate odometric drift and motion dynamics (Torres-Torriti and Nazate-Burgos, 2022). However, IMUs induce inertial drift and positioning errors even when fully calibrated and even if filters based on accurate motion models are employed because the strong terrain variability produces strong disturbances in translational acceleration and angular velocity measurements that deteriorate robot pose estimation, and consequently the concurrent mapping process (Cremona et al., 2022; Sier et al., 2023; Wang et al., 2024).

Given these challenges, this work presents a consistent lidar-only SLAM approach suitable for GNSS-denied arboreal environments with uneven terrain, as illustrated in Fig. 1, where dense tree canopies frequently obstruct GNSS signals, rendering high-precision, centimeter-level positioning infeasible. The proposed approach is particularly suited for robots with non-smooth motion in arboreal environments with persistent vertical structures, such as tree trunks, vineyard posts, and plantation stakes. The method is less suitable when salient vertical structures are missing, such as in very young orchards, espalier systems, or very sparse plantations.

The proposed method introduces a dimensionality reduction step by solving the localization employing 2D projections of 3D lidar scans slices of likely tree trunks and lower canopy sections, which are matched to an auxiliary 2D gridmap using the *modified Hausdorff distance* (MHD) (Donoso-Aguirre et al., 2008) as a robust distance metric without requiring explicit feature extraction and data-association stages, nor the matching of large 3D point clouds; see the process overview depicted in Fig. 1. This improves alignment accuracy and robustness by mitigating outliers commonly caused by foliage and dynamic obstacles. The algorithm allows to build the 3D map and operates without reliance on visual features, inertial correction, or GNSS feedback, making it well-suited to outdoor deployment in dense fruit tree plantations or forestry. The approach builds upon earlier work by the authors on robust localization techniques in highly repetitive and geometrically ambiguous environments, such as mining tunnels (Torres-Torriti et al., 2022), coastal navigation (Bustos et al., 2007), and preliminary results presented in Nazate-Burgos et al. (2025-05-23).

### 1.1. Contributions and scope

This work makes the following contributions to agricultural robotics:

(i) Systematic evaluation of SLAM performance when IMU integration is infeasible: We demonstrate that methods designed for IMU-aided operation may fail catastrophically when sensors are unreliable, while simpler approaches maintain consistent performance across diverse conditions. We compare the proposed SLAM approach to A-LOAM (Qin Tong, 2019; Zhang et al., 2014), LeGO-LOAM (Shan and Englot, 2018), DLO (Chen et al., 2022), MOLA (Blanco-Claraco, 2025), and AG-LOAM (Teng et al., 2025). The performance of the different approaches is evaluated in a controlled environment and using three real world datasets: CitrusFarm (Teng et al., 2023), the Bacchus Long-Term (BLT) dataset (Polvara et al., 2024), and the Pullally dataset collected by the authors https://github.com/RAL-UC/Pullally_Dataset. The diversity of arboreal environments and platforms contributes to a comprehensive evaluation of the capabilities and limitations of the proposed approach compared to the benchmark algorithms (Jiang et al., 2025).

(ii) Application of robust distance metrics to agricultural environments: We adapt the modified Hausdorff distance, previously used in mining and coastal navigation, to the specific challenges of arboreal SLAM, including the design of probabilistic height-based filtering for canopy environments.

(iii) Evaluation on legged robotic platforms: Unlike existing agricultural SLAM benchmarks that use wheeled robots, we demonstrate feasibility on a quadruped robot experiencing discontinuous motion dynamics that invalidate standard motion model assumptions.

(iv) Comprehensive performance metrics: three map metrics are reported for simulation and controlled environments, while seven standardized trajectory metrics are evaluated across all datasets, providing more complete assessment than typical agricultural SLAM studies.

(v) Open code and datasets: We release complete code implementation, along with the Pullally dataset featuring quadruped robot data and uneven terrain at https://github.com/RAL-UC/RoSA_SLAM, https://github.com/RAL-UC/Pullally_Dataset.

It is important to note that this work focuses on scenarios where IMU-based approaches are impractical. When high-quality IMU data and smooth platform motion are available, tightly-coupled lidar-inertial methods may achieve superior performance. Our contribution is demonstrating reliable operation in the complementary regime where such assumptions fail.

The proposed SLAM method is designed as a foundational navigation capability that enables a range of downstream tasks. Specifically, a legged robot equipped with reliable localization and mapping in arboreal environments can be deployed for: (i) phenotyping and crop monitoring, by carrying multispectral or RGB-D cameras along tree rows (Yandun Narvaez et al., 2017); (ii) precision spraying or targeted weeding, by navigating to specific georeferenced positions (Calderara-Cea et al., 2024); (iii) yield estimation, by mapping canopy volume along pre-planned trajectories (Gené-Mola et al., 2020; Underwood et al., 2016); and (iv) flower thinning, branches pruning and delicate fruit harvesting, such as cherries, nectarines, table grapes, if endowed with robotic arms and corresponding tools (Kefalas et al., 2025; Navone et al., 2025; Bhattarai et al., 2024). Legged robots are particularly suited for these tasks because they can traverse the uneven, root-disrupted terrain typical of orchard floors that would impede wheeled platforms.

### 1.2. Related works on lidar-based SLAM for arboreal environments

Interest in SLAM for arboreal agricultural environments is relatively recent, and thus very few approaches exist compared to a plethora of SLAM approaches for indoor and structured urban environments (Torres-Torriti and Nazate-Burgos, 2022). Although visual SLAM algorithms are popular, especially in indoor environments due to their suitability for place recognition, their sensitivity to initialization, illumination, and range make them unreliable when used alone to support autonomous navigation, especially in outdoor urban and agricultural environments (Shan et al., 2020). Therefore, the focus of this paper is on lidar-based SLAM methods.

One of the earliest implementations of SLAM for autonomous navigation in agricultural arboreal environments, specifically in olive groves, was proposed in Auat Cheein et al. (2011), which employed a landmark-based SLAM algorithm using an Extended Information Filter (EIF-SLAM) combining 2D lidars and image processing for tree trunks detection.

Since the pioneering work (Auat Cheein et al., 2011) applying SLAM to agricultural orchards, two IMU-aided 3D lidar-based SLAM

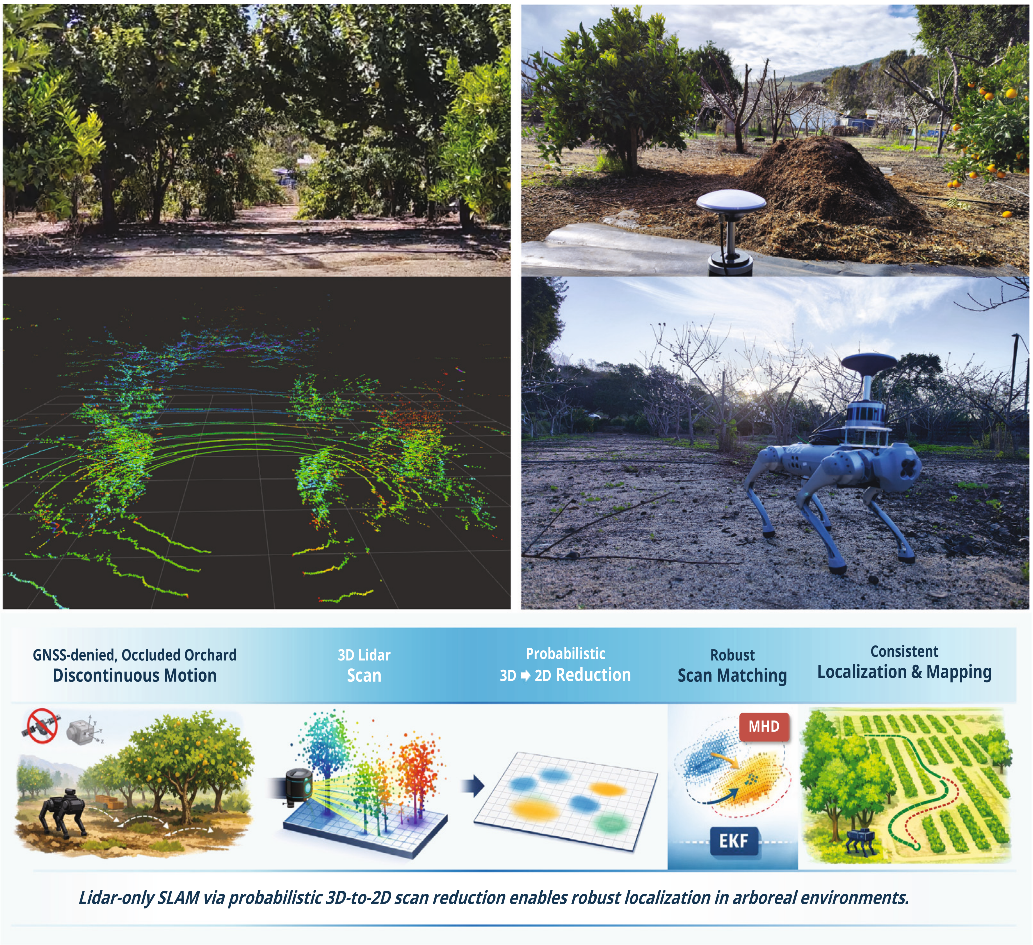


**Fig. 1.** Fruit orchard with dense canopies: RGB camera view in summer (top-left) and corresponding 3D lidar scan (bottom-left). The same orchard row in winter (top-right) and the legged robot with the lidar and GNSS employed in the experiments (bottom-right).

approaches for trellis and tree orchards have been proposed in Aguiar et al. (2022), Xia et al. (2023) and one IMU-less approach is proposed in Nie et al. (2022). The approach proposed in Aguiar et al. (2022), called VineSLAM, extracts point and semiplane features from the 3D point clouds. VineSLAM implements lidar odometry based on the Iterative Closest Point Method (ICP) in a 6-DOF space of translations and rotations. It uses a particle filter for tracking point and plane features in a 3D voxel map, instead of a 2D grid map. VineSLAM achieved RMS positioning errors in the range 0.4–1.6 m outperforming LeGO-LOAM (Shan and Englot, 2018), which yielded root mean squared (RMS) positioning errors between 0.9–22.5 m when tested in the Aveleda vineyard (Portugal). The method proposed in Xia et al. (2023) implements the Normal Distribution Transform (NDT) for 3D point cloud matching and odometry estimation, but combines the 3D lidar odometry with GNSS signals to initialize the estimated pose. The results show that the fusion of lidar odometry with GNSS data improves RTK GNSS positioning accuracy in orchard environments. The IMU-less approach in Nie et al. (2022) also uses the ICP for matching 3D point clouds like (Aguiar et al., 2022), but introduces the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm to fit point clouds to tree trunks modeled as cylinders. The experimental results show that the approach proposed in Nie et al. (2022) yields similar positioning errors to those obtained with LOAM (Zhang et al., 2014), LIO-SAM (Shan et al., 2020), and Cartographer (Hess et al., 2016) with an RMS positioning accuracy of 0.15 ± 0.08 m compared to the 0.13 ± 0.05 m of LOAM and 0.19 ± 0.11 m of Cartographer. LIO-SAM failed to converge in 4 out of 5 test runs.

Concerning popular state-of-the-art 2D lidar-based SLAM approaches originally conceived for indoor environments, the work in Li and Zhu (2024) tested Hector SLAM (Kohlbrecher et al., 2011), GMapping (Grisetti et al., 2007), and Cartographer (Hess et al., 2016) in a simulated orchard environment and also with real world data of the Bacchus dataset (Polvara et al., 2024). The results obtained with the Bacchus dataset show that the approaches achieve RMS positioning errors of 0.29 m, 0.15 m and 0.11 m for Hector, GMapping and Cartographer, respectively. It is to be noted that these approaches yield comparable accuracies to those obtained with 3D lidar-based approaches, but require wheel odometry or IMU measurements, and likewise require smooth motion models for filtering the motion dynamics and pose estimation.

A comprehensive evaluation of 15 representative 3D lidar SLAM methods covering lidar-only, lidar-inertial, and lidar-visual-inertial approaches was conducted in GNSS-denied stacked cage farming houses by Jiang et al. (2025), finding that methods based on direct point cloud registration outperform feature-extraction-based approaches, and reporting pose drift and mapping failures in challenging aisle-like environments similar in geometric repetitiveness to the arboreal rows considered in this work.

Early real-time lidar SLAM systems such as LOAM (Zhang et al., 2014) established a feature-based paradigm by decoupling high-rate odometry from lower-rate mapping through edge and planar feature matching. While effective in structured scenes, this approach may be sensitive to unstable geometric features and can accumulate drift in vegetation-rich environments.

**Table 1**
Comparative summary of lidar odometry and SLAM benchmark methods.

| Method | Year | Scan representation | Registration | Matching strategy | Map type | Loop closure | IMU dependency | Ground segmentation | Dynamic filtering | Computational efficiency | Agricultural validation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LOAM (Zhang et al., 2014) | 2014 | Feature (edge + plane) | Point-to-line & point-to-plane (L–M) | S2S + S2M (dual freq.) | Global voxelized PC | × | Optional | × | × | Medium | × |
| LeGO-LOAM (Shan and Englot, 2018) | 2018 | Feature after cluster segmentation | 2-step decoupled (L–M) | S2S + S2M | Global voxelized PC | ∼[a] | Optional | ✓ | ∼[b] | High | × |
| DLO (Chen et al., 2022) | 2022 | Dense (voxel downsampled) | GICP (NanoGICP) | S2S + S2M (keyframe submap) | Keyframe database | ×[c] | Optional | × | × | Very High | × |
| MOLA (Blanco-Claraco, 2025) | 2025 | Dense (adaptive voxel) | Velocity-integrated ICP | S2M (frame-to-map) | View-based (simple-map) + multi-layer metric | ∼[d] | × | × | × | High | ∼[e] |
| AG-LOAM (Teng et al., 2025) | 2025 | Dense (adaptive voxel) | GICP (frame-to-map) | S2M (cascaded pipeline) | Adaptive consistency map | × | × | × | ✓ | High | ✓ |

**Symbols:** ✓ Yes/Supported; ∼ Partial/Optional; × No/Not supported.
**Abbreviations:** S2S = scan-to-scan; S2M = scan-to-map; PC = point cloud; L–M = Levenberg–Marquardt; GICP = Generalized ICP.

[a] LeGO-LOAM's loop closure mode uses ICP-based scan matching; a descriptor-based detector is absent.

[b] Cluster-based noise rejection filters small isolated objects (e.g. grass blades) but does not explicitly track or remove dynamic vegetation across frames.

[c] DLO is a front-end odometry system; loop closure is flagged as future work by the authors.

[d] MOLA supports GNSS-seeded ICP loop closure; topological descriptor-based detection is identified as future work.

[e] MOLA was tested on Almería forest sequences (backpack, 32-beam lidar), which share geometric properties with plantation agriculture, but no dedicated agricultural benchmark was used.

LeGO-LOAM (Shan and Englot, 2018) improves computational efficiency and robustness by introducing ground segmentation and cluster-based noise rejection, enabling reliable operation on ground vehicles with limited onboard resources. Its two-stage optimization exploits ground constraints to reduce complexity, but its reliance on consistent ground geometry may limit its performance on rough or furrowed agricultural terrain and in densely planted orchards.

More recent methods favor dense scan-matching strategies to avoid brittle feature extraction. Direct lidar Odometry (DLO) proposed in Chen et al. (2022) employs a fully dense Generalized ICP-based front-end with adaptive key-framing and aggressive data structure reuse, achieving high accuracy and real-time performance on resource-constrained platforms. However, DLO lacks explicit mechanisms for handling dynamic vegetation or long-term drift, which are critical in agricultural deployments.

MOLA (Blanco-Claraco, 2025) proposes a modular SLAM framework centered on view-based map representations and tightly coupled velocity estimation within ICP, enabling IMU-free operation and robustness to motion distortion. While highly flexible and validated across diverse datasets, its performance in repetitive row-crop environments remains largely unexplored.

AG-LOAM (Teng et al., 2025) explicitly targets agricultural environments by integrating dense Generalized ICP (GICP) odometry with adaptive map update policies that suppress unreliable measurements during unstable motion and filter dynamic vegetation. Experimental results in real orchards demonstrate superior robustness and accuracy compared to general-purpose SLAM systems, albeit without loop closure.

Overall, the literature indicates a clear shift from feature-based toward dense, adaptive SLAM approaches. For agricultural environments, robustness to geometric repetition, vegetation dynamics, and motion-induced distortion emerges as more critical than map completeness or architectural generality.

A-LOAM, LeGO-LOAM, DLO, MOLA and AG-LOAM have been selected for a comparative evaluation of the performance of the proposed approach. A-LOAM and LeGO-LOAM are widely employed SLAM algorithms, while DLO, MOLA and AG-LOAM are state-of-the-art algorithms that emerge as possible better options for IMU/GNSS-free lidar-based SLAM. A comparative summary of the selected lidar odometry and SLAM benchmark methods is presented in Table 1. On the other hand, a qualitative assessment of the suitability of the selected benchmarking methods for agricultural applications is summarized in Table 2.

## 2. Proposed method

An overview of the proposed approach is depicted in the block diagram of Fig. 2. The approach involves three main stages: 3D scan processing and probabilistic projection, scan matching, and state and map updating. The approach is initialized with the first lidar scan $\mathcal{Z}_0$, stored in the world frame $\mathcal{F}_0$, which is also associated to the global map frame $\mathcal{M}_0$. Subsequent scans in the sensor frame $\mathcal{F}_s$ must be matched to previous ones to obtain relative changes in translation and rotations of the robot platform, which are represented by $\Delta\boldsymbol{x}_t$ in the diagram. The relative state changes in the robot pose are used to update the state estimate $\boldsymbol{x}_t$ and the map, together with the incremental updates of the map $\Delta\mathcal{M}_t$ that correspond to the new occupied areas detected after the scan matching stage. The process is carried out in an iterative incremental fashion, thus the $z^{-1}$ blocks represent memory storage for the robot state and environment map from the previous iteration, $\boldsymbol{x}_{t-1}$ and $\mathcal{M}_{t-1}$, respectively. The next subsections explain in more detail each of the main stages of the proposed SLAM approach for arboreal environments without GNSS nor IMU measurements. The flowchart in Fig. 3 provides a complete end-to-end view of the SLAM pipeline, complementing the mathematical description in Sections 2.1 and 2.2 by covering all major stages of the algorithm: lidar scan acquisition and 3D-to-2D probabilistic projection, EKF state prediction, scan-to-map Hausdorff distance minimization, EKF measurement update, and occupancy grid map update, as well as the decision branches for convergence and initialization.

### 2.1. Scan processing

The 3D lidar measures distances $r^s_{i,j} = r(\theta_i, \phi_k)^s$ to the nearest object in the path of the beam pointed in an azimuth direction $\theta_i = i \cdot \Delta\theta^s$, $i = 1, 2, \ldots, N_\theta$, and elevation angle $\phi_j = i \cdot \Delta\phi^s$, $j = 1, 2, \ldots, N_\phi$ with respect to the plane of the sensor's reference frame $\mathcal{F}_s$, which is typically parallel to the ground plane. Here $\Delta\theta^s$ and $\Delta\phi^s$ are the azimuth and elevation angular resolutions of the lidar sensor, while $N_\theta$ and $N_\phi$ define the corresponding number of samples, respectively. Thus, a 3D lidar scan at time instant $t$ is a set of measurements in polar coordinates:

$$\mathcal{Z}^s_t = \left\{ \boldsymbol{z}^s_{t,i,j} \overset{def}{=} \left(r^s_{t,i,j}, \theta^s_{t,i}, \phi^s_{t,j}\right) \;\middle|\; \theta_i = i \cdot \Delta\theta^s, \phi_j = j \cdot \Delta\phi^s; \right. \\ \left. i = 1, 2, \ldots, N_\theta; j = 1, 2, \ldots, N_\phi \right\} \tag{1}$$

Measurements $\boldsymbol{z}^s_{t,i,j} \in \mathcal{Z}^s_t$ in polar coordinates $\boldsymbol{z}^s_{t,i,j} = \left(r^s_{t,i,j}, \theta^s_{t,i}, \phi^s_{t,j}\right)$ can be expressed in Cartesian coordinates $\boldsymbol{z}^s_{t,i,j} = \left(x^s_{t,i,j}, y^s_{t,i}, z^s_{t,j}\right)$ of

**Table 2**
Agricultural suitability assessment of lidar odometry and SLAM methods considered for comparative benchmarks.

| Method | Vegetation robustness | Uneven terrain handling | GNSS-denied operation | Embedded HW suitability | Single Lidar sufficiency | Key agricultural limitation |
|---|---|---|---|---|---|---|
| LOAM | Low — feature extraction degrades with inconsistent vegetation returns | Moderate — no active distortion correction; sensitive to abrupt motion | ✓ (Lidar only) | Low — exceeds real-time budget on Jetson-class hardware | ✓ | Diverges in feature-sparse or vegetation-dense environments; no loop closure |
| LeGO-LOAM | Moderate — cluster segmentation filters small vegetation but not dense canopy | Moderate — 2-step optimization assumes stable ground plane | ✓ (Lidar only) | High — demonstrated on Jetson TX2 in real time | ✓ | Ground-plane dependency problematic on furrow terrain; tunable $c_{\text{th}}$ requires per-environment adjustment |
| DLO | Moderate — dense matching less sensitive to feature consistency, but no dynamic filtering | Moderate — optional IMU prior helps; no active distortion correction | ✓ (Lidar only; IMU optional) | Very High — <22 ms/scan, <10% CPU (single core) | ✓ | No loop closure; no explicit mechanism to reject dynamic vegetation from map |
| MOLA | Moderate–High — velocity-integrated ICP robust to aggressive motion; forest sequences validated | High — IMU-free via tightly-coupled velocity estimation | ✓ (GNSS optional for loop closure) | High — runs faster-than-realtime on tested hardware | ✓ | No dedicated agricultural evaluation; constant-velocity assumption may be violated at headland turns |
| AG-LOAM | High — adaptive mapper rejects motion-distorted and temporally inconsistent points (dynamic vegetation) | High — motion stability criterion suppresses map updates during rough-terrain traversal | ✓ (lidar only) | High — competitive runtime; cascaded pipeline minimizes mapping overhead | ✓ (360° FOV required) | Requires 360° spinning lidar; no loop closure; not evaluated in row-crop environments without vertical structure |

**Note:** Ratings (Low/Moderate/High) are qualitative assessments synthesized from the reviewed papers' experimental results and design analyses. **Symbols:** ✓ = Yes/Supported; × = No/Not supported.

the *sensor's reference frame* $\mathcal{F}_s$ by the polar-to-Cartesian transformation $C : (r, \theta, \phi) \rightarrow (x, y, z) = (r\cos(\theta)\cos(\phi), r\sin(\theta)\cos(\phi), r\sin(\phi))$:

$$\begin{aligned} x^s_{t,i,j} &= r^s_{t,i,j}\cos(\theta^s_{t,i})\cos(\phi^s_{t,j}),\\ y^s_{t,i,j} &= r^s_{t,i,j}\sin(\theta^s_{t,i})\cos(\phi^s_{t,j}),\\ z^s_{t,i,j} &= r^s_{t,i,j}\sin(\phi^s_{t,j}), \end{aligned}$$

A key step of the proposed approach is the *probabilistic projection* of the 3D scan slices (1) of the trunk and canopy to produce a 2D set of measurements on the ground plane as illustrated in Fig. 4, which shows that 3D points on different trees are discarded or projected on the 2D plane of sensor frame axes $X^s$ and $Y^s$ by setting the measurement vertical coordinate $z^s$ to zero, i.e. $z^s = 0$. For example, points $1, 2, 3$ in Fig. 4 are projected to the ground plane as three different measurements, while points 5 and 6, as well as 8 and 9, are projected as single points 5 and 8 because they share similar locations on the tree trunk, thus reinforcing the confidence about the presence of a tree at that location. In contrast, distant points like 4 or points high in the canopy are discarded and not projected to the ground plane according to the measurements' probability, which will be explained next. The measurement probabilities allow us to discard points that are less reliable and would degrade the localization, either because small angular errors produce larger errors in the correct assignment of points in the Cartesian grid, or because higher points in the canopy are more likely to become occluded to the view of the lidar sensor when the robot moves to a new position.

The measurement probability $p(\mathcal{Z}^s_t | \boldsymbol{x}_t, \mathcal{M}_t)$ can be understood as the correctness of the measurements given the current robot state $\boldsymbol{x}_t$ and the information in the map $\mathcal{M}_t$ (Mutz et al., 2021). In other words, if the robot true location is not $\boldsymbol{x}_t$, or the real world contains different elements than those in the map $\mathcal{M}_t$ it is less likely that the observations are truly corresponding to correct observations from the presumed state and map. To keep calculations tractable, SLAM strategies that employ grid maps typically assume that the cells of the map are independent (Siciliano and Khatib, 2008), and thus the conditional likelihood of the measurements $\boldsymbol{z}^s_{t,i,j} \in \mathcal{Z}^s_t$ given $\boldsymbol{x}_t$ and $\mathcal{M}_t$ can be rewritten as the product of conditional likelihoods of measurements on the corresponding cells:

$$p(\mathcal{Z}^s_t | \boldsymbol{x}_t, \mathcal{M}_t) = \prod_{\substack{z^s_{t,i,j} \in \mathcal{Z}^s_t,\\ i=1,\ldots,N_\theta,\\ j=1,\ldots,N_\phi}} p(\boldsymbol{z}^s_{t,i,j} | \boldsymbol{x}_t, \mathcal{M}_t) \tag{2}$$

where each measurement's likelihood can be computed as the jointly independent conditional probabilities of the lidar beam being at a distance of the nearest object (e.g. tree trunk) and at a certain height, respectively $p_d$ and $p_h$ (defined below):

$$p(\boldsymbol{z}^s_{t,i,j} | \boldsymbol{x}_t, \mathcal{M}_t) = p_d\left(D\left(T^w_s(x^s_{t,i,j}, y^s_{t,i,j}; \boldsymbol{x}_t), \mathcal{M}_t\right) | \boldsymbol{x}_t, \mathcal{M}_t\right) p_h\left(z^s_{t,i,j} | \boldsymbol{x}_t, \mathcal{M}_t\right)$$

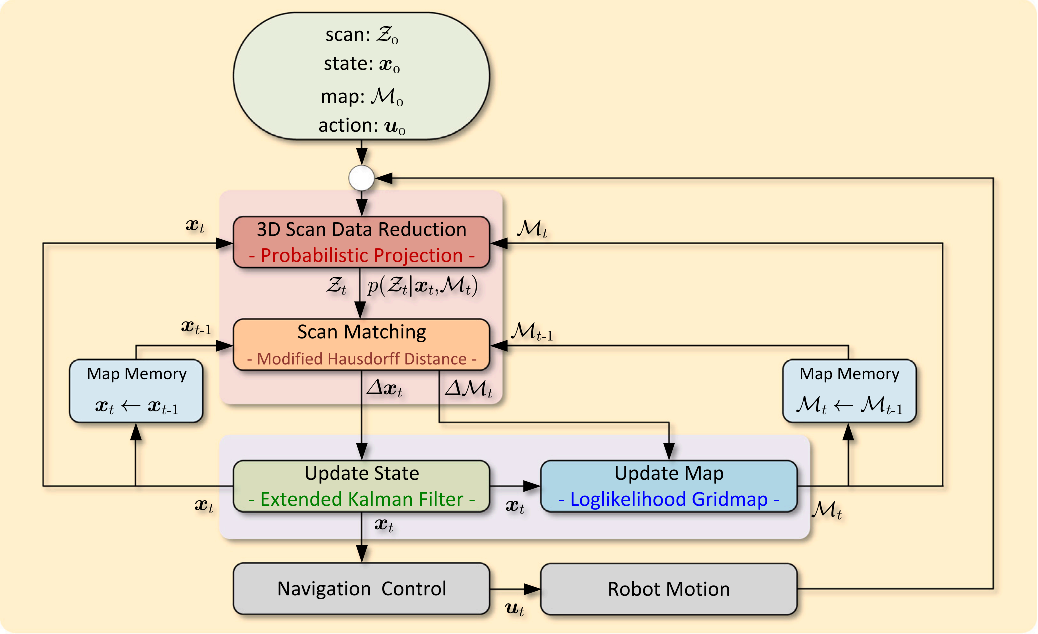


**Fig. 2.** General overview of the proposed lidar-only SLAM for agricultural legged robots in arboreal environments by robust data reduction and matching.

with $D : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ corresponding to the distance transform of the occupied areas or object boundaries $\partial\mathcal{M}_t$ in the map $\mathcal{M}_t$, which is defined as:

$$D(\boldsymbol{p},\mathcal{M}_t) \overset{def}{=} D_{\mathcal{M}_t}(\boldsymbol{p}) \;=\; \min_{\boldsymbol{m}\in\partial\mathcal{M}_t} \|\boldsymbol{p}-\boldsymbol{m}\|_1, \text{ for all } \boldsymbol{p}\in\mathbb{R}^2,$$

and $T_s^w \in SE(2)$ is the 2D transformation of the measurement coordinates $(x^s, y^s)$ in the sensor-frame to world-frame coordinates $(x^w, y^w)$, which can be expressed in terms of a homogeneous transformation matrix $\boldsymbol{T} = [\boldsymbol{R} \in SO(2)|\boldsymbol{t} \in \mathbb{R}^2] \overset{def}{=} \left[\boldsymbol{R}(\theta_t)|\boldsymbol{p}_t\right]$ as the following mapping:

$$T_s^w(x^s, y^s; \boldsymbol{x}_t) : (x^s, y^s) \rightarrow (x^w, y^w) = \boldsymbol{T}\left[x^s \;\; y^s \;\; 1\right]^T = \boldsymbol{R}(\theta_t)\left[x^s y^s\right]^T + \boldsymbol{p}_t, \tag{3}$$

where $\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$ is the 2D rotation matrix. The distance transform $D(\boldsymbol{p},\mathcal{M}_t)$ forms a scalar field over the 2D plane, where each value represents distance to the closest point on the boundary $\partial\mathcal{M}_t$ of the occupied regions corresponding to environment objects in the map $\mathcal{M}_t$, see for example (Danielsson, 1980) for the earliest fast algorithms for computing distance transforms.

The probability $p_d$ of a lidar measurement hitting at distance $d$ from the expected object given the robot state $\boldsymbol{x}_t$ and information in the map is modeled as a zero-mean Gaussian distribution $d \sim N(0, \sigma_s^2)$:

$$p_d(d) = \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{-\frac{d^2}{2\sigma_s^2}}, \tag{4}$$

where $\sigma_s^2$ is the lidar range variance. On the other hand, the probability of the beam hitting a tree trunk at height $z_t^s$ is modeled as an exponential probability density:

$$p_h(z) = \begin{cases} \lambda e^{-\lambda z} & , \quad z \geq 0, \\ 0 & , \quad z < 0, \end{cases} \tag{5}$$

where $\lambda$ is chosen so that the cumulative distribution considers that at least 50% of the measurements fall below the average canopy height. Thus, this parameter may need to be adjusted for the plant species. If the average canopy height is $\bar{z}$, then the cumulative distribution $F(\bar{z},\lambda) = 1 - e^{-\lambda\bar{z}} = 0.5$, implies $\lambda = \ln(2)/\bar{z}$.

In order to discard measurement points $\boldsymbol{z}_t^s$ of lower reliability, a threshold is applied to the measurements according to a selection rule expressed as the following indicator function:

$$\mathbb{1}_{\substack{p_d(\cdot)>\beta_d \\ p_h(\cdot)>\beta_h}}(\boldsymbol{z}_t^s) = \begin{cases} 1 \quad , & \begin{array}{c} \text{if } p_d\left(D\left(T_s^w(x_t^s, y_t^s; \boldsymbol{x}_t), \mathcal{M}_t\right)|\boldsymbol{x}_t, \mathcal{M}_t\right) \;\geq\; \beta_d \\ \text{and } p_h\left(z_{t,i,j}^s|\boldsymbol{x}_t, \mathcal{M}_t\right) \;\geq\; \beta_h, \end{array} \\ 0 \quad , & \text{otherwise,} \end{cases} \tag{6}$$

where the value of $\beta_d$ is given by:

$$\beta_d = \frac{e^{-\frac{1}{2}}}{\sqrt{2\pi\sigma_s^2}} \approx \frac{1}{4\sigma_s},$$

and the value of $\beta_h$ is given by:

$$\beta_h = \frac{\ln(2)}{\bar{z}} e^{-\frac{\ln(2)}{\bar{z}}\bar{z}} = \frac{1}{2}\frac{\ln(2)}{\bar{z}}.$$

The value of $\beta_d$ is set according to the distribution of range measurements (4) so that the selected distance measurements are within $\pm 1\sigma$ of the sensor range error, i.e. 65% of the measurements closest to the expected zero-mean error are selected. The value of $\beta_h$ is computed with $z = \bar{z}$, according to the distribution of height measurements (5), so that measurements are selected whenever the height is below the average canopy height $\bar{z}$.

Finally, the set of 2D measurements in world coordinates for a given robot pose $\boldsymbol{x}_t$ is generated from the 3D measurements using the indicator function (6) and discarding the height information $z_t^s$ as:

$$\overline{\mathcal{Z}}_t^w(\boldsymbol{x}_t) \;=\; \Big\{(x_t^w, y_t^w) \;|\; (x_t^w, y_t^w) = T_s^w(x^s, y^s; \boldsymbol{x}_t), \; \forall\, \boldsymbol{z}_t^s = (x^s, y^s, z^s) \in \mathcal{Z}_t^s, \\ \text{such that } \mathbb{1}_{\substack{p_d(\cdot)>\beta_d \\ p_h(\cdot)>\beta_h}}(\boldsymbol{z}_t^s) = 1\Big\}. \tag{7}$$

### 2.2. Localization by scan-to-map matching using the modified Hausdorff distance

In order to solve the robot pose and find $\boldsymbol{x}_t = (x_t, y_t, \theta)$, the scan-to-map matching problem (Li et al., 2025; Torres-Torriti and Nazate-Burgos, 2022) is solved via the minimization of the modified Hausdorff distance (Torres-Torriti et al., 2022; Donoso-Aguirre et al., 2008) between the measurements $\overline{\mathcal{Z}}_t^w(\boldsymbol{x}_t)$ and the map $\mathcal{M}_t$. This technique is suitable for unstructured environments because it does not

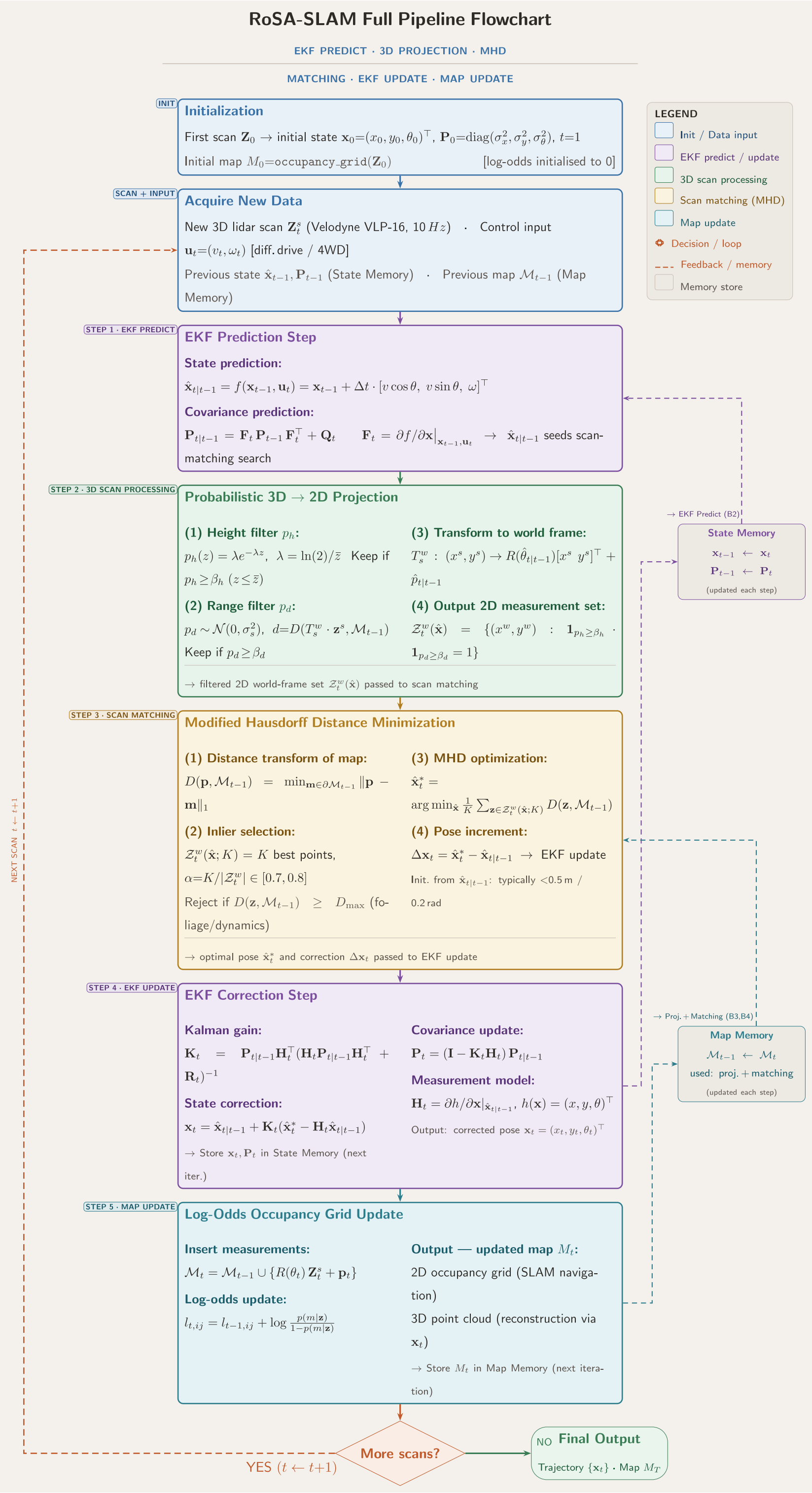


**Fig. 3.** Flowchart of the full SLAM pipeline of the proposed method.

require extracting geometric features from the raw measurements, nor solving feature-to-landmark data association problems.

The robot localization problem can be formulated as the process of finding a translation $\Delta \boldsymbol{p}_t$ and rotation $\Delta\theta_t$ to update the robot's position $\boldsymbol{p}_{t-1}$ and its orientation $\theta_{t-1}$, such that the current measurements $\mathcal{Z}_t^s$ coincide with the previous measurements $\mathcal{Z}_{t-1}^s$ or map $\mathcal{M}_{t-1}$. Formally, the estimated pose vector $\hat{\boldsymbol{x}}_t^*$ is computed as the one that minimizes the matching error between the object boundaries $\partial\mathcal{M}_{t-1}$ in the map $\mathcal{M}_{t-1}$

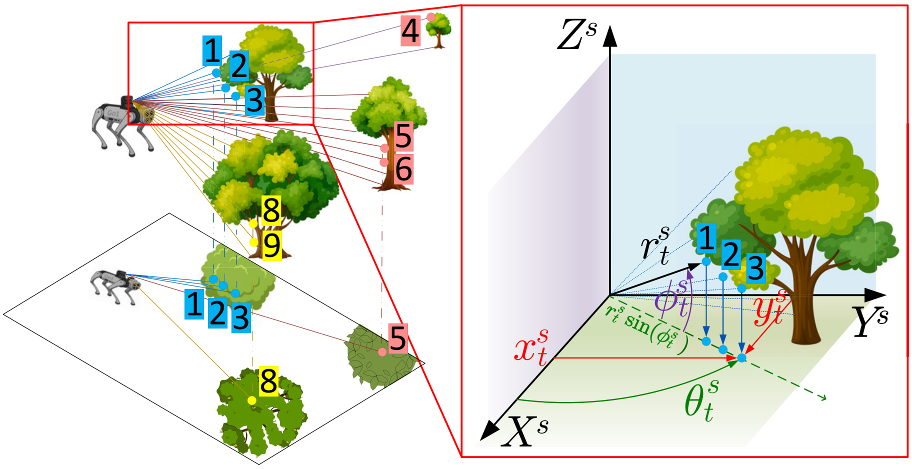


**Fig. 4.** Data reduction of 3D lidar measurements by probabilistic projection of data slices to the 2D ground plane.

and the observed location of the objects' boundaries obtained from the lidar measurements $\overline{\mathcal{Z}}_t^w(\boldsymbol{x}_t)$ (see Eq. (7)):

$$\hat{\boldsymbol{x}}_t^* = (\hat{x}_t^*, \hat{y}_t^*, \hat{\theta}_t^*) \;=\; \arg\min_{\hat{\boldsymbol{x}}=(\hat{x},\hat{y},\hat{\theta})} \bar{h}_K(\mathcal{M}_{t-1}, \overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})) \tag{8}$$

where, for two set of points $\boldsymbol{a}_i \in \mathcal{A}$, $i = 1, 2, \ldots, |\mathcal{A}|$ and $\boldsymbol{b}_j \in \mathcal{B}$, $j = 1, 2, \ldots, |\mathcal{B}|$:

$$\bar{h}_K(\mathcal{A}, \mathcal{B}) = \frac{1}{K} \sum_{i=|\mathcal{B}|}^{|\mathcal{B}|-K} h_i(\mathcal{A}, \mathcal{B}_i),$$

is the so-called *modified Hausdorff distance* and $h_i(\mathcal{A}, \mathcal{B})$ is the $i$th *partial (directed) Hausdorff distance*:

$$h_i(\mathcal{A}, \mathcal{B}_i) \overset{def}{=} \max_{\boldsymbol{b}\in\mathcal{B}_i} \min_{\boldsymbol{a}\in\mathcal{A}} \|\boldsymbol{b} - \boldsymbol{a}\|_1, \tag{9}$$

$$= \max_{\boldsymbol{b}\in\mathcal{B}_i} D(\boldsymbol{b}, \mathcal{A}), \tag{10}$$

and $\mathcal{B}_i$ is the set $\mathcal{B}$ without the first $i-1$ points that solved the directed Hausdorff distance. More specifically, if $\mathcal{B}_1 = \mathcal{B}$, then

$$\mathcal{B}_{i+1} = \mathcal{B}_i - \left\{b_i^*\right\}$$

where

$$b_i^* = \arg\max_{\boldsymbol{b}\in\mathcal{B}_i} D(\boldsymbol{b}, \mathcal{A}).$$

Since $h_i(\mathcal{A}, \mathcal{B}_i) \geq h_j(\mathcal{A}, \mathcal{B}_j)$ for $i < j$ because $\mathcal{B}_i \supset \mathcal{B}_j$, the set $\mathcal{B}_i$ contains $|\mathcal{B}| - i + 1$ points that best match (are closest) to the points in $\mathcal{A}$. It is to be noted that to quickly solve (8), the partial (directed) Hausdorff distance (9) was rewritten using the distance transform in (10). A computationally effective solution to the scan matching problem for localization (8) is obtained by first computing the distance transform of the boundaries $\partial\mathcal{M}_{t-1}$ as shown in Fig. 5 and reformulating (8) as:

$$\hat{\boldsymbol{x}}_t^* = (\hat{x}_t^*, \hat{y}_t^*, \hat{\theta}_t^*) \;=\; \arg\min_{\hat{\boldsymbol{x}}=(\hat{x},\hat{y},\hat{\theta})} \frac{1}{K} \sum_{\boldsymbol{z}\in\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}};K)} D(\boldsymbol{z}, \mathcal{M}_{t-1}) \tag{11}$$

where

$$\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; K) = \cup_{k=1}^K \{\arg \min_{\substack{\boldsymbol{z}\in\overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})\setminus\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}};k-1) \\ s.t.\ D(\boldsymbol{z},\mathcal{M}_{t-1})<D_{max}}} D(\boldsymbol{z}, \mathcal{M}_{t-1})\} \tag{12}$$

and $\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; 0) = \{\emptyset\}$, i.e. $\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; K)$ is the set of the $K$-best matching points of $\overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})$ matching to $\mathcal{M}_t$.

The parameter $K$ in Eqs. (11) and (12) determines how many of the best-matching points are used in the distance computation. This acts as an outlier rejection mechanism: points with large distances (typically foliage, moving objects, or measurement noise) are excluded from the optimization.

The value of $K$ is a dynamically chosen value within an interval, such that the inlier ratio $\alpha$ satisfies:

$$\alpha \overset{def}{=} \frac{K}{|\overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})|} \in [0.7, 0.8].$$

where $|\overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})|$ is the total number of projected measurements. This means the number of best matching points will typically be 70%–80% of the number of projected measurements, while 20%–30% will be considered worst matches and discarded as outliers. A measurement $\boldsymbol{z} \in \overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})$ is said to be an inlier if its distance $D(\boldsymbol{z}, \mathcal{M}_{t-1})$ to the map is smaller than $D_{max}$, where $D_{max}$ is the maximum measurement-to-map discrepancy accepted. $D_{max}$ must be selected in terms of the lidar's range resolution and map resolution. If $\Delta r$ denotes the lidar's range resolution, typically the grid map resolution $\Delta m$, is selected to satisfy $\Delta m \geq \Delta r$, and $D_{max}$ will be chosen to be 2–4 times $\Delta m$, i.e. $\Delta_{max} \in [2, 4]$, to accept all measurement points that are typically $\pm 1$ pixels away from the estimated object boundaries $\partial\mathcal{M}_{t-1} \subset \mathcal{M}_{t-1}$ and provide a reasonable margin for matching errors of single points up to 4 times the map resolution when $\Delta m \approx \Delta r$. This allows to reject outliers that occur either due to measurement errors arising from difficulties in scanning canopies and leaves, or outliers that correspond to new sections of the unexplored environment.

In our experiments, setting the inlier ratio $\alpha \in [0.7, 0.8]$ for all datasets, provided a good balance between robustness to outliers and maintaining sufficient constraints for accurate pose estimation. This range was chosen after a sensitivity analysis which showed that values outside this range degrade the accuracy of localization due to either insufficient outlier rejection ($\alpha$ too high) or over-rejection of valid measurements ($\alpha$ too low).

For clarity of exposition, it is to be noted that the set $\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; K)$ of $K$-best matching points in (12) is expressed as the union of the best matching points found in the original set $\overline{\mathcal{Z}}_t^w(\hat{\boldsymbol{x}})$ to which the set $\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; k-1)$ of $k-1$-best matching points, for $k = 1, 2, \ldots, K$, has been removed. Thus, (12) implicitly describes the recursive construction of the set of best matching points, starting from an initially empty set $\overline{\overline{\mathcal{Z}}}_t^w(\hat{\boldsymbol{x}}; 0) = \{\emptyset\}$. In the practical implementation, this is simply a list sorting operation in ascending order of distances and selecting the points associated to the $K$-smallest distances. It is to be noted that even if the formulation of the scan matching problem in (11) does not appear to be the standard max–min distance (Ryu and Kamata, 2021), the partial Hausdorff distances (9) or (10) involving max–min distance expressions can be sought in ascending order, starting from

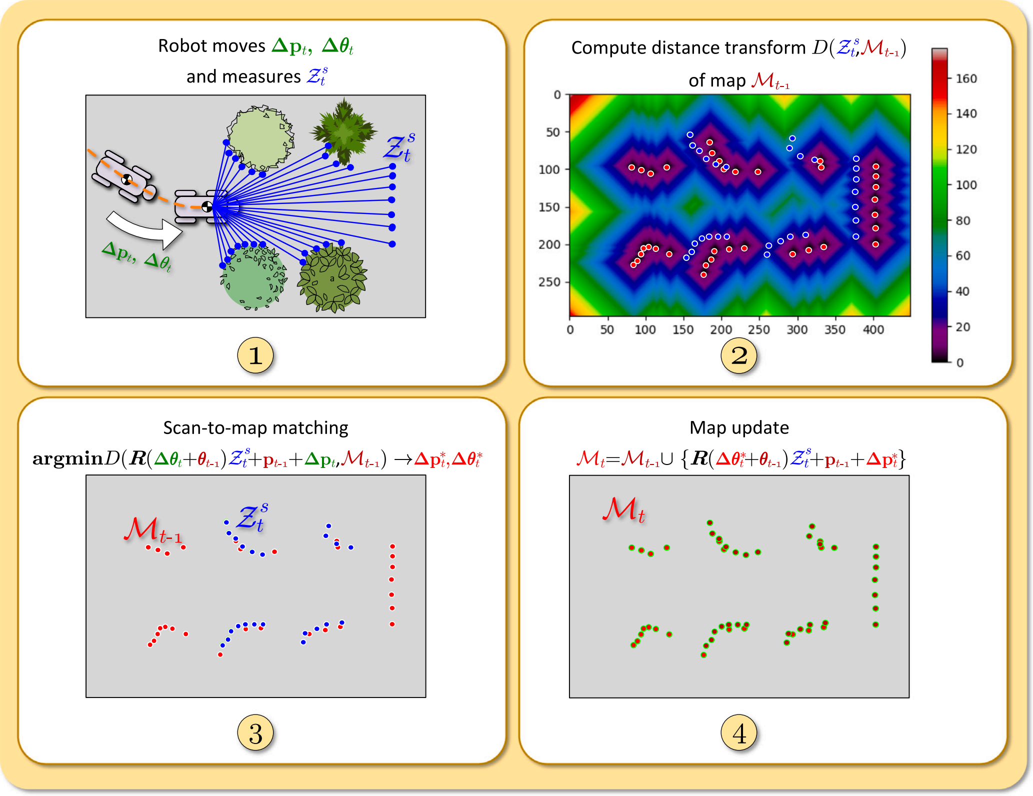


**Fig. 5.** Scan-to map matching using the modified Hausdorff distance: (1) measurement $\mathcal{Z}_t^s$ after robot translational motion $\Delta \boldsymbol{p}_t$ and rotational motion $\Delta\theta_t$; (2) the distance transform of the map $\mathcal{M}_{t-1}$ is computed; (3) minimization of the modified Hausdorff distance yields the estimated motion $\Delta \boldsymbol{p}_t^*$ and $\Delta\theta_t^*$; (4) measurements $\mathcal{Z}_t^s$ are translated and rotated according to the estimated motion to produce an updated map $\mathcal{M}_t$.

the closest points of one set to a point in the other set written in terms of min–min distance expressions of (11) instead of the typical max–min expressions (9) or (10) for the $i$th partial Hausdorff distance computed in the typical descending order (Donoso-Aguirre et al., 2008).

The solution of the scan matching problem (11) is accelerated by initializing the search from the predicted pose $\hat{\boldsymbol{x}}_{t|t-1}$ obtained using the motion model and EKF prediction step; see Peralta-Cabezas et al. (2008) for further details on EKF implementation. This provides a good initial guess for the optimization, typically within 0.5 m and 0.2 rad of the true solution, significantly reducing computation time. On the other hand, the set of high-likelihood 2D measurements in (7) are used to update a log-likelihood 2D occupancy grid; see for example (Thrun et al., 2005; Thrun, 1998) for details on log-odds map representation for incremental update of occupancy grid maps and Torres-Torriti and Nazate-Burgos (2022) for a general introduction of *online SLAM* algorithm implementation. The source code is publicly available at https://github.com/RAL-UC/RoSA_SLAM.

Concerning the motion model and control inputs of the quadruped robot, the evolution of the robot's state $\boldsymbol{x}_t$ is simplified to unicycle-type kinematic model (Morin and Samson, 2008):

$$\dot{\boldsymbol{x}}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} v_t \cos(\theta_t) \\ v_t \sin(\theta_t) \\ \omega_t \end{bmatrix}, \qquad \boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}} \tag{13}$$

where the state vector $\boldsymbol{x}_t = [x_t \; y_t \; \theta_t]^T$ comprises the so-called pose variables, namely the position $\boldsymbol{p}_t = [x_t \; y_t]^t$ and orientation $\theta_t$ of the robot on the 2D plane at time $t$, $v_t$ is the longitudinal velocity of the robot and $\omega_t$ is the angular velocity of the robot. In the kinematic model, $v_t$ and $\omega_t$ are treated as the manipulated variables, and thus the control vector is $\boldsymbol{u}_t = [v_t \; \omega_t]^T$. The model can also represent a differential drive kinematic model considering that control inputs $v_t$ and $\omega_t$ can be expressed in terms of right-wheel and left-wheel angular velocities $\dot{\phi}_r$ and $\dot{\phi}_l$, $v_t = (r\dot{\phi}_r + r\dot{\phi}_l)/2$ and $v_t = (r\dot{\phi}_r + r\dot{\phi}_l)/2$, $\omega_t = (r\dot{\phi}_r - r\dot{\phi}_l)/W$; see Torres-Torriti and Nazate-Burgos (2022). It is to be noted that current quadruped robots, such as the one used in this work, are capable of omnidirectional motion and have independently controllable velocities in the $x_t$, $y_t$ and $\theta_t$ coordinates. Therefore, the kinematic model (13) may be replaced by an omnidirectional-type kinematic model. However, in our experiments the quadruped robot was driven by adjusting only the longitudinal velocity $v_t$ and the angular velocity $\omega_t$ without acting on the lateral velocity (orthogonal to $v_t$). Therefore, the motion filters based on the Extended Kalman filter (EKF) do not yield better position estimates when implemented using the omnidirectional-type kinematic model compared to the simpler model (13). This is explained because the covariance matrices have more parameters to be tuned and, on the other hand, the omnidirectional kinematics may exhibit discontinuous lateral velocities that violate the assumptions that ensure the optimality of the Kalman filter, which require the system to be an ideally linear system with Gaussian distributed zero-mean disturbances or sensor noises. The reader is referred to Peralta-Cabezas et al. (2008) for further details on model discretization and discrete-time EKF implementation. The measurement model is $\boldsymbol{z}_t = \hat{\boldsymbol{x}}_t^* = [\hat{x}_t^*, \hat{y}_t^*, \hat{\theta}_t^*]^T$ because the full state vector is obtained as a result of the scan matching process (11). In other words, the scan matching process can be thought of as an alternative GNSS and compass sensor.

The following aspects concerning the proposed consistent SLAM algorithm: (1) practical implementation guidelines, (2) its generalizability and scope, (3) the influence of environmental characteristics, (4) the step-by-set parameter calibration procedure, and (5) the summary of key tunable parameters is provided as Supplementary Material; see Appendix.

## 3. Experimental methodology and results

The experimental methodology details considered in the validation of the proposed approach and the corresponding results are presented in the following subsections.

### 3.1. Experimental methodology

The proposed approach for navigation and mapping in arboreal environments without GNSS nor IMU signals is evaluated first in simulations presented in Section 3.3, then tested in a controlled outdoor environment with a quadruped robot as presented in Section 3.4, and finally validated in real fruit orchards in Section 3.5 considering the CitrusFarm (Riverside, California, USA) and Bacchus (Epanomi, Greece) datasets; see Teng et al. (2023), Polvara et al. (2024), as well as our dataset for the Pullally experimental fruit orchard in Valparaiso, Chile. In each case (simulation, controlled and real environments) the proposed approach is compared to A-LOAM, LeGO-LOAM, DLO, MOLA and AG-LOAM. As previously mentioned, A-LOAM and LeGO-LOAM are popular and well-established lidar odometry and mapping SLAM algorithms, while DLO, MOLA and AG-LOAM are more recent state-of-the-art algorithms that support IMU/GNSS-free lidar-based SLAM. These algorithms were also selected due to their open-source availability. The reader is referred to Section 1.2 for a discussion on the main characteristics of the selected algorithms for benchmarking purposes and their applicability to the agricultural context. LIO-SAM, which is also a popular algorithm, especially used in structured environments, failed to converge in several runs of the different datasets most likely because it heavily relies on inertial odometry. Therefore, the results for LIO-SAM are not included here for the sake of a fair comparison of the algorithms operating without inertial odometry. The performance metrics employed to evaluate the proposed approach are presented in the following Section 3.2.

Tests in the controlled environment and the Pullally experimental fruit orchard were carried out using a Unitree Go1® quadruped robot. Key parameters include: maximum locomotion speed of 3.7 m/s (running speed), standard operating gait of trotting at 0.3–0.5 m/s during data collection, body mass of approximately 12 kg, and a low-level proprioceptive controller running at 500 Hz onboard. The robot was operated in its standard trot gait throughout the Pullally and controlled environment experiments. The stride length was programmed to a minimum of 0.15 m in order to create the dataset at maximum possible spatial sampling resolution. The stride of the robot would vary typically between 0.15 ± 0.05 m due to the robot's stable balance control perturbations and foot slippage caused by the unevenness of the terrain. The low-level locomotion controllers were employed with the factory settings for gains, duty cycle of a gait pattern and phase offsets as this have already been optimized by the manufacturer of the robot for usability in a wide class of terrains. The duty cycle for trot was 0.5 and the phase offset was 0.5. The duty cycle in this context corresponds to the proportion of the duration of the contact phase to a gait duration, and the phase offsets are relative offsets from the phase of the front-left and rear-right legs of the front-right and rear-left legs. The gait period was configured to 1 stride per second. The intermittent leg–ground contact at this gait frequency is the primary source of the impulsive accelerations that corrupt IMU measurements, as discussed in Sections 1 and 3.5. The nominal life of a fully charged battery is 4 h. In our experiments, the robot battery would last 2 h considering the trajectories and terrain. Thus we swapped the backup battery every 1.5 h. Energy to an external GNSS RTK for ground truth acquisition and a lidar was supplied by an external battery to increase the power autonomy of the whole system. Experiments did not employ the robot internal computers, but an external laptop computer with an Intel® Core™ i7 3.8 GHz CPU with 12 cores, running on its own power. The computer ran Ubuntu 18.04 64-bit operating system with ROS Melodic.

Concerning the sensors, the legged robot is equipped with a Velodyne VLP-16 lidar and an ArduSimple RTK3B GNSS (see Fig. 6(d)). The Velodyne VLP-16 lidar has 16 near-IR channels (903 nm wavelength) providing a vertical field of view of 30° (± 15° above/below the horizontal plane) with a vertical resolution of 2.0°. The horizontal field of view is 360° with adjustable horizontal (azimuth) resolution between 0.1 and 0.4°. The maximum measurement range of the Velodyne VLP-16 is 100 m and has a range accuracy of ±3 cm. The rotation rate is between 4 to 20 Hz depending on the horizontal resolution setting. It produces about 300.000 data points per second. The average power consumption is ∼8 W. The lidar was configured to scan with an horizontal resolution of 0.4°, delivering a full 360° scan every 0.25 s (4 Hz). The ArduSimple RTK3BPro GNSS is based on the Septentrio® Mosaic-X5 receiver chip, which was used to obtain ground truth measurements with centimeter-level positioning accuracy at a 1 Hz sampling rate. The GNSS system was paired with a calibrated Survey GNSS Tripleband+L-band antenna both at the base and mobile receiver.

### 3.2. Performance evaluation metrics

The estimation of the robot pose and the environment's map are intertwined. Therefore, unlike most works that only report trajectory estimation accuracy metrics, here both the trajectory estimation errors and the map accuracy are reported. The results are evaluated using the metrics proposed in Kümmerle et al. (2009), Zi et al. (2020), Sturm et al. (2012), that account for absolute trajectory errors (ATE) and relative displacement/pose errors (RPE) in translation and rotation, and the commonly employed root mean square (RMS) localization error metric (Li and Zhu, 2024). The use of standardized metrics is important to ensure the results presented are comparable with those obtained employing other SLAM strategies or the same algorithms on different datasets. This metrics are implemented in the Python package *evo* (Grupp, 2017) for assessing and comparing trajectories estimated with odometry and SLAM algorithms. The quality of the estimated map is evaluated using the precision and sensitivity metrics. These metrics provide a better indicator of the correctness and completeness of the map as shown in Zi et al. (2020) than the traditional average distance to the nearest neighbor (ADDN) metric. The mathematical definitions of each performance metric is provided as Supplementary Material; see Appendix.

### 3.3. Simulation tests

The performance of the proposed SLAM algorithm was first evaluated in a simulated arboreal environment. The simulation test results are provided as Supplementary Material; see Appendix.

### 3.4. Experimental validation in a controlled environment

Before the experimental validation in real-world arboreal environments, the performance of the proposed approach is evaluated in a controlled outdoor environment with traffic cones representing tree trunks in known locations, as shown in Fig. 6. The experiments are carried out using a Unitree Go1 quadruped robot equipped with a Velodyne VLP-16 lidar, an ArduSimple RTK3B GNSS (see Fig. 6(d)), and a laptop with Ubuntu 18.04 64-bit operating system with ROS Melodic. The GNSS is based on the Septentrio® Mosaic-X5 receiver chip and is used to obtain ground truth measurements with centimeter-level positioning accuracy. The robot was programmed as in the simulation experiments to follow a 6 × 2 m rectangular path. The 16 m long trajectory was repeated by the robot seven times to build the controlled environment dataset and obtain the statistical confidence intervals with the seven realizations, i.e. $N_e = 7$.

The proposed algorithm, described in Section 2, was evaluated using the performance metrics outlined in Section 3.2. For comparison

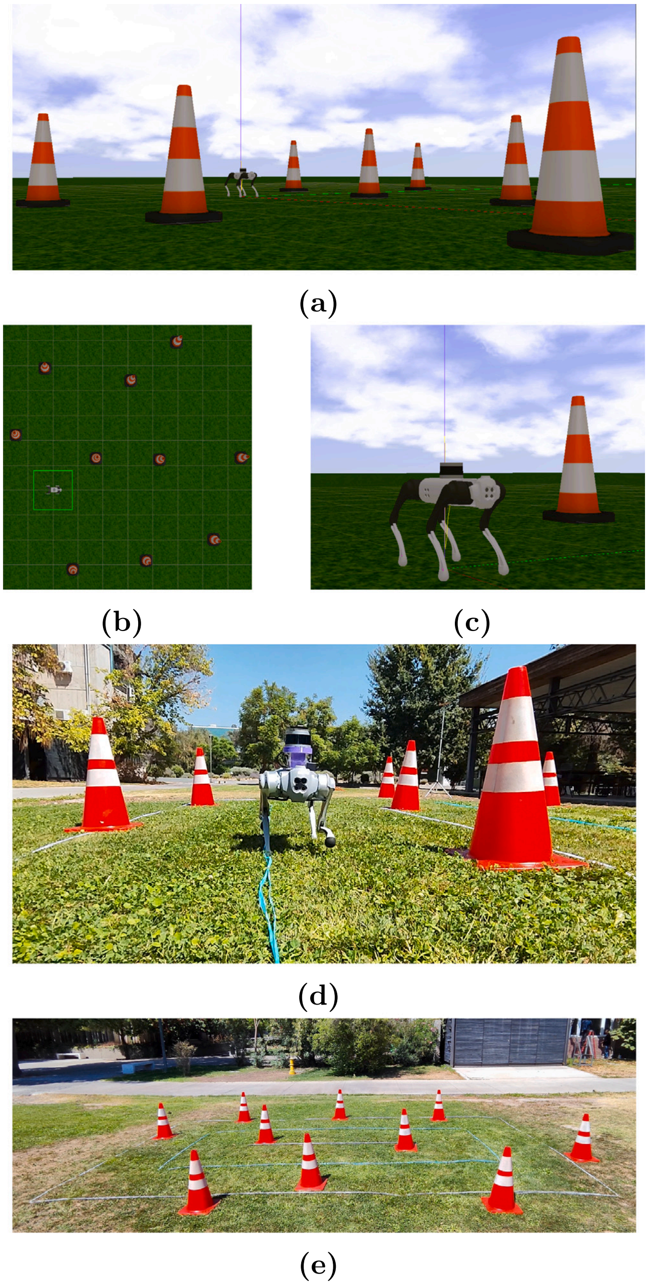


**Fig. 6.** Robot and traffic cones for the controlled field environment experiments: (a) simulation view of the experiment; (b) top view of the traffic cones positions in the field; (c) front view of the quadruped robot; (d) controlled field experiments; (e) perspective view of the traffic cones and the testing rectangular reference path.

purposes, the A-LOAM, LeGO-LOAM, DLO, MOLA and AG-LOAM algorithms were also evaluated without using IMU nor GNSS measurements.

### 3.4.1. *Baseline performance under controlled environment*

The experimental results are presented in Tables 3, 4, Figs. 7 and 8. Compared to A-LOAM, DLO, and MOLA, the proposed method yields significantly lower positional and angular errors when referenced to a global position, indicating superior accuracy and reduced variability. Compared to LeGO-LOAM and AG-LOAM, the proposed approach yields slightly smaller positional and angular errors. All methods have equivalent performance in terms of positional displacement estimation errors of about 1 cm, except for DLO and MOLA. This suggests comparable capability in estimating relative movement. DLO and MOLA failed to properly handle the point cloud of measurements of the traffic cones in certain scans, and this is the likely cause the algorithms started to diverge after the first turn in the rectangular path as shown in Figs. 7(c), 7(d), 8(c) and 8(d). Even if A-LOAM has comparable displacement errors, to those of LeGO-LOAM, AG-LOAM and the proposed method, its angular displacement estimation error is more than 20 times larger that of the proposed approach. This causes A-LOAM to diverge, as can be seen in Figs. 7(a) and 8(a), which show the cumulative errors in orientation and the final distorted map built by A-LOAM. LeGO-LOAM's and AG-LOAM's trajectory estimations are more consistent across multiple runs, although wrong orientation estimates occur in a few runs, especially at the second turn, as shown in Figs. 7(b) and 7(e). Erroneous orientation estimation causes the maps to be incomplete or generate false positives, even if both LeGO-LOAM AG-LOAM are able to recover the correct position estimation in spite of the robot's orientation estimation errors, as shown in Figs. 8(b) and 8(e). In contrast, the proposed approach shows consistency in the estimated trajectories

**Table 3**

Pose estimation errors for the controlled outdoor environment. Position $\boldsymbol{p}$ errors in meters and angular $\theta$ errors in radians.

| Pose estimation metrics | Controlled environment experiment | | | | | |
|---|---|---|---|---|---|---|
| | Proposed approach | A-LOAM | LeGO-LOAM | DLO | MOLA | AG-LOAM |
| $\bar{e}_{\boldsymbol{p}} \pm C_{\boldsymbol{p}}$ | 0.079 ± 0.009 | 0.917 ± 0.137 | 0.097 ± 0.025 | 4.861 ± 0.559 | 1.682 ± 0.165 | 0.190 ± 0.053 |
| $\bar{e}_{\theta} \pm C_{\theta}$ | 0.071 ± 0.015 | 0.439 ± 0.031 | 0.131 ± 0.064 | 1.306 ± 0.106 | 1.420 ± 0.100 | 0.316 ± 0.094 |
| $e_{RMS_p} \pm C_{RMS_p}$ | 0.092 ± 0.011 | 1.289 ± 0.187 | 0.156 ± 0.054 | 6.400 ± 0.719 | 2.430 ± 0.119 | 0.394 ± 0.136 |
| $e_{RMS_\theta} \pm C_{RMS_\theta}$ | 0.091 ± 0.018 | 0.548 ± 0.050 | 0.299 ± 0.189 | 1.708 ± 0.132 | 1.897 ± 0.048 | 0.652 ± 0.220 |
| $\bar{e}_{\Delta\boldsymbol{p}} \pm C_{\Delta\boldsymbol{p}}$ | −0.007 ± 0.005 | −0.013 ± 0.016 | −0.007 ± 0.014 | −2.225 ± 0.480 | 0.343 ± 0.040 | −0.113 ± 0.051 |
| $\bar{e}_{\Delta\theta} \pm C_{\Delta\theta}$ | −0.029 ± 0.021 | 0.001 ± 0.032 | −0.114 ± 0.089 | −0.723 ± 0.204 | 0.190 ± 0.071 | −0.238 ± 0.103 |
| $e_{RMS_{\Delta p}} \pm C_{RMS_{\Delta p}}$ | 0.081 ± 0.017 | 0.080 ± 0.016 | 0.133 ± 0.041 | 3.812 ± 0.320 | 0.464 ± 0.023 | 0.379 ± 0.133 |
| $e_{RMS_{\Delta\theta}} \pm C_{RMS_{\Delta\theta}}$ | 0.136 ± 0.032 | 0.208 ± 0.031 | 0.355 ± 0.199 | 1.220 ± 0.164 | 0.385 ± 0.066 | 0.638 ± 0.217 |
| $e_{\boldsymbol{p}\%} \pm C_{\boldsymbol{p}\%}$ | 0.665 ± 0.073 | 7.772 ± 1.222 | 0.821 ± 0.215 | 41.083 ± 4.597 | 14.238 ± 1.475 | 1.616 ± 0.457 |
| $e_{end} \pm C_{end}$ | 0.119 ± 0.020 | 2.569 ± 0.333 | 0.118 ± 0.071 | 8.373 ± 3.817 | 4.379 ± 0.339 | 0.337 ± 0.296 |

**Table 4**

Map estimation errors for the controlled outdoor environment using a grid map resolution of 0.01 m/pixel.

| Map estimation metrics | Controlled environment experiment | | | | | |
|---|---|---|---|---|---|---|
| | Proposed approach | A-LOAM | LeGO-LOAM | DLO | MOLA | AG-LOAM |
| $Precision \pm C_{Precision}$ | 0.923 ± 0.029 | 0.002 ± 0.003 | 0.218 ± 0.234 | 0.003 ± 0.002 | 0.010 ± 0.007 | 0.250 ± 0.221 |
| $Sensitivity \pm C_{Sensitivity}$ | 0.805 ± 0.035 | 0.013 ± 0.015 | 0.108 ± 0.092 | 0.178 ± 0.281 | 0.018 ± 0.012 | 0.242 ± 0.215 |
| $Accuracy \pm C_{Accuracy}$ | 0.999 ± 0.001 | 0.952 ± 0.018 | 0.992 ± 0.002 | 0.791 ± 0.281 | 0.984 ± 0.003 | 0.992 ± 0.002 |

across multiple runs, as well as a high similarity of the estimated map with the ground truth, as shown in Fig. 7(f) and Fig. 8(f), respectively.

The map estimation performance of the proposed approach is the highest, compared to that of all the benchmark methods, with a precision above 92% and a sensitivity above 80%, as reported in Table 4. LeGO-LOAM and AG-LOAM achieved the second best precision and sensitivity, in the range 10%–25%, whereas A-LOAM, DLO and MOLA have very low precision scores below ∼1%. In this case of imbalanced binary classification of cells as occupied or free, the accuracy metric is a poor indicator because it achieves a high score by simply predicting the majority class, which is in this context the free space. Nonetheless, in relative terms, the proposed method also exhibits a higher accuracy compared to the other approaches.

### 3.5. Experimental evaluation in real fruit orchards

The proposed method was evaluated using three real-world datasets captured in fruit orchard environments: two publicly available datasets CitrusFarm (Teng et al., 2023) and Bacchus (Polvara et al., 2024), and one dataset collected by the authors, the Pullally dataset https://github.com/RAL-UC/Pullally_Dataset. A key criterion for dataset selection was the presence of uneven terrain, which presents significant challenges for robotic navigation and environmental perception by disrupting stable locomotion and accurate sensing. As demonstrated in the simulation study by Li and Zhu (2024), terrain roughness hinders smooth, controlled movement of mobile platforms and introduces noise in sensor data, which negatively affects mapping, object detection, and spatial understanding. The main characteristics of each dataset are summarized next.

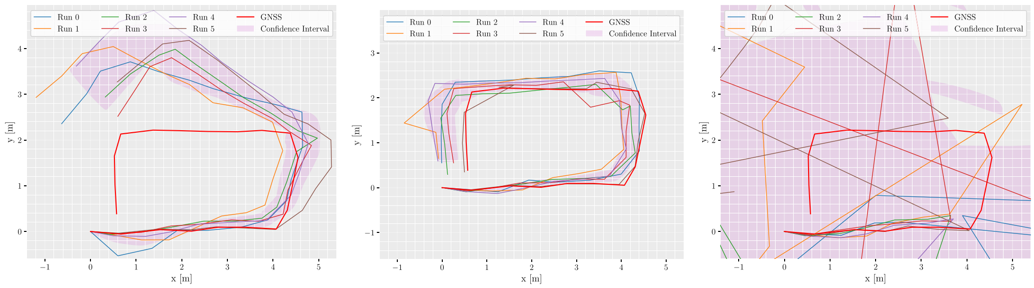
In contrast with the controlled environment experiment, for which the ground-truth map was straightforwardly constructible because the cones' positions were precisely known a priori, none of the three real-world datasets (CitrusFarm, Bacchus, Pullally) include a ground-truth occupancy map against which to compute precision, sensitivity, and accuracy. While RTK-GNSS trajectories are available and used as ground truth for pose evaluation, generating a ground-truth 2D occupancy map of the orchard would require a dedicated, high-accuracy reference survey (e.g., terrestrial laser scanning or aerial lidar) that was not part of the original dataset acquisition campaigns. The generation of reference maps for real fruit tree orchard environments is planned as part of future work.

### 3.5.1. Description of the datasets

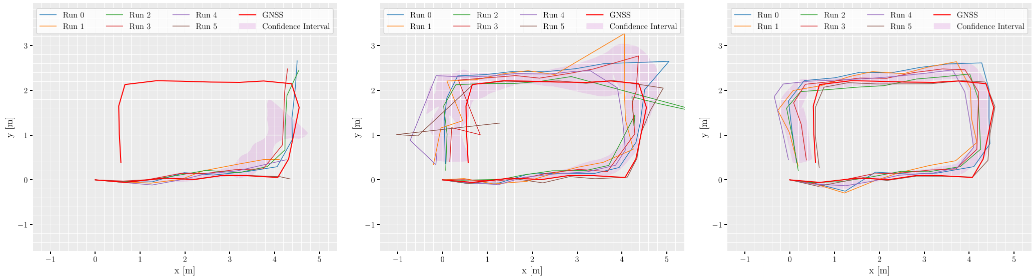
**CitrusFarm Dataset**: This dataset was captured at the Agricultural Experimental Station of the University of California, Riverside, using a Clearpath Jackal wheeled mobile robot (Teng et al., 2023). The platform was equipped with a Velodyne VLP-16 3D lidar, offering a 360° horizontal field of view (FOV) with angular resolution of 0.2°, a 30° vertical FOV with angular resolution of 2°, a range accuracy of ±3 cm and a maximum range of 100 m at 0.1 Hz. A SwiftNav Duro GPS-RTK system provided the ground truth with a spatial positioning error of less than 0.02 m at 0.1 Hz. The platform also included a regular monochrome camera, a stereo RGB camera, a red–green–NIR camera, a thermal camera, wheel odometry and IMU sensors, which are not used here for the comparison of the lidar-based SLAM approaches.

The dataset comprises seven sequences recorded across three citrus fields, featuring trees of different species at various growth stages. It includes 7.5 km of recorded trajectories spanning 1.7 h of operation, totaling 1.3 TB of data. For this study, Sequence 05 was used, which contains an 865.33 m trajectory along 7 rows of citrus trees. This sequence was selected due to its challenging conditions, including tree density and uneven terrain, which are appropriate for benchmarking the proposed algorithm, A-LOAM, LeGO-LOAM, DLO, MOLA and AG-LOAM.

**Bacchus Dataset**: The Bacchus dataset was collected in the Ktima Gerovassiliou vineyard, Epanomi, Greece, using a Saga Robotics Thorvald II mobile robot that has a four-wheel drive and steer (4WD4S) wheel setup (Polvara et al., 2024). The platform was equipped with an Ouster OS1-16 3D lidar (360° horizontal FOV, 32° vertical FOV, 120 m range at 10 Hz), and ground-truth positioning was provided

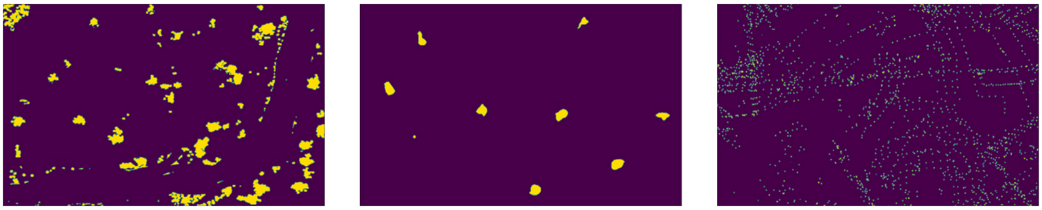


(a) A-LOAM (b) LeGO-LOAM (c) DLO



(d) MOLA (e) AG-LOAM (f) Proposed method

**Fig. 7.** Estimated trajectories for the controlled outdoor environment by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method.



(a) A-LOAM (b) LeGO-LOAM (c) DLO

(d) MOLA (e) AG-LOAM (f) Proposed method

**Fig. 8.** Estimated final maps for the controlled outdoor environment by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method. Map area is 8x5 m with 0.01 m/pixel resolution.

by a Trimble BX992 RTK-GNSS system with centimeter-level accuracy. The robot was also equipped with IMU, 2D lidars and RGB-D cameras that acquired data not used for purposes of benchmarking the proposed algorithm.

The dataset includes a long-term data acquisition campaign conducted over several months to capture the crop's growth. The dataset comprises trajectories along five adjacent vineyard corridors. The acquisition session of June 8th, 2022, with waypoints B→G→F→A→B, which is 102.3 m long, is used in this work for evaluation purposes. This sequence, recorded during late spring, features dense foliage of grapevines growing on a trellis training system, thus creating a challenging environment for unambiguous non-degenerate localization due to the uniformity of the plants creating a wall-like long corridor without salient geometrical features. Similar to the Citrus dataset, the uneven terrain also contributes to degrading the matching accuracy of lidar, thereby degrading the performance of SLAM algorithms.

**Pullally Dataset**: This dataset was collected by the authors at the Pullally experimental fruit orchard located in Valparaiso, Chile, using a Unitree Go1 quadruped robot. The robot was equipped with a Velodyne VLP-16 lidar, like the robot of the Citrus Dataset, and an ArduSimple RTK3BPro GNSS-RTK system paired with a calibrated Survey GNSS Tripleband+L-band antenna. Ground-truth positions were recorded at 1 Hz with sub-centimeter accuracy using a stationary RTK base station.

The robot followed a predefined 150 m trajectory through a mixed-species orchard containing orange, lemon, plum, peach, and almond trees. The recording was conducted in winter, during which citrus trees retained foliage while the others did not. The trajectory was repeated three times in order to obtain the statistical confidence intervals. A satellite image of the orchard section employed for the acquisition campaign showing the robot's reference path marked in red is presented in Fig. 9(a). The main challenges of this dataset include varying tree heights, mixed foliage conditions, and highly irregular terrain. A special feature of this dataset lies in the fact that the lidar point clouds have less positional continuity than in the other datasets because the sensor is mounted on a legged robot. Unlike wheeled robots, legged robots naturally introduce discontinuous leg–ground contact forces that propagate to the body of the robot, thus disturbing IMU measurements which cannot be properly integrated to produce reliably inertial odometry, even when using high-precision calibrated IMUs. This aspect, together with the loss of GNSS signals caused by dense tree canopies, motivates the proposed approach for localization in arboreal environments. Fig. 9(b) shows the 3D reconstruction of the orchard by combining point clouds based on the robot's pose obtained from the GNSS, while Fig. 9(c) shows the 3D reconstruction obtained using the pose estimated by the proposed approach. It is to be noted that, even if the proposed approach is based on the data reduction of 3D lidar measurements by probabilistic projection of 3D lidar scans to a 2D plane and matching in the projection plane, the map reconstruction from the estimated poses adequately preserves the 3D details of tree canopies because of the achieved positioning accuracy, as shown next in the results discussion section.

The main characteristics of the datasets considered are summarized in Table 5. It is to be noted that the lidars employed in the acquisition of each dataset are widely adopted models that have similar characteristics. However, the platform type, motion smoothness and orchard plants are very different, thus contributing to the comprehensive assessment under different real world conditions. For comparison purposes, all maps are reported using an image resolution of 0.01 m/pixel. This is reasonable choice considering the lidars employed by the different datasets have an accuracy of ±0.03 m.

### *3.5.2. Performance under real fruit orchards*

A comparison of the pose estimation metrics obtained with the proposed approach, A-LOAM, LeGO-LOAM, DLO, MOLA, and AG-LOAM applied to the Citrus, Bacchus and Pullally datasets is presented in Tables 6–8. A graphical comparison of the estimated trajectories and maps is shown in Fig. 10, Fig. 11, Fig. 12 and Fig. 13 for the Citrus, Bacchus and Pullally datasets, respectively.

The results indicate that the proposed approach yields accurate pose estimates on average across the different orchard environments. The mean norm of the position estimation errors ranges between 0.40 m and 1.34 m, while the mean orientation error ranges between 0.05 rad and 0.67 rad (2.86°–38.39°) for the proposed approach across the different datasets. The large variation in orientation errors across datasets is attributed not only to the feature of the environments and motion smoothness of the platform that can make point cloud matching more difficult due to the uniformity of the canopies as in the Bacchus datasets, but also to the number of turns of the trajectory and the increased chance of wrongly estimating the robot's heading in each turn, as in the case of the zig-zagging trajectory of the Pullally dataset. All methods have comparatively low positional and angular displacement errors on average. However, the averaged displacement estimation errors may yield low values that do not properly convey the effect of cumulative errors into the absolute pose estimation. It is to be noted that the positional error over the path length $e_{p\%}$ and the end-point estimation error $e_{end}$ provide better measures of the effect of cumulative error. In particular $e_{p\%}$ metric takes into account the length of the trajectory for a fair comparison of cumulative errors across different length paths. In all datasets, the proposed approach yields lower values for positional error over the path length and the end-point estimation error. Hence, these results confirm that the proposed method can robustly solve the localization problem with consistent accuracy across datasets for different orchards, and regardless of the length or shape of the trajectory.

The results also show that the Pullally dataset is the most challenging one. A-LOAM, DLO and MOLA fail in all runs to estimate correctly the robot turns at the end of each row, as shown in Figs. 12(a), 12(c), and 12(d). On the other hand, LeGO-LOAM is able to estimate correctly only one out of three trajectories. LeGO-LOAM is able to reconstruct the trajectory correctly only along the first row in two cases, see Fig. 12(b), but on the second row the small orientation estimation errors start to accumulate. Considerable angular estimation errors are made by LeGO-LOAM in the second turn, before traversing the third row of the orchard, as can be seen in Fig. 12(b). AG-LOAM has a good handling of two of the three runs; see Fig. 12(e). However, an erroneous heading estimation in the second run cause AG-LOAM to wander. Compared to the other approaches, results indicate that AG-LOAM recovers the orientation after a while, but maintains some cumulative errors which prevent it from fulfilling a correct loop closure.

The average *total processing time per scan* and the computation *maximum frame rate* across real world datasets is summarized in Table 9. The *total processing time* includes the different steps of the SLAM algorithm, such as (i) possible data pre-processing for feature extraction or outliers removal, (ii) scan-to-map or scan-to-scan matching, (iii) odometry computation, (iv) mapping. The elapsed time of the complete computation cycle was measured by reading the high-resolution CPU clock with microseconds accuracy before the acquisition of a new lidar scan and at the end of the scan processing pipeline. The difference in CPU ticks between the initial and final clock reading at each cycle is averaged to compute the average total processing time per scan. The maximum frame rate is the inverse of the shortest elapsed time per scan processing cycle. An important caveat is that none of these figures are directly comparable to the reported original authors' work, since they were measured on different hardware, with different datasets, and some differences may arise depending on specific parameters settings related to the density of points considered for point clouds registration. However, these values provide order-of-magnitude indicators and reference indicators for real-time application of the strategies. Real-time processing requirements will also depend on specific application aspects, such as the expected operation speed of the robotic platform. The proposed method, without any special code optimization, requires almost twice the processing time per scan compared to A-LOAM and LeGO-LOAM, and about five times the processing time of AG-LOAM. Ongoing work is concerned with code optimization to achieve higher scan processing rates.

## 4. Discussion

The experimental results presented in Section 3 allow a comprehensive evaluation of the proposed approach relative to the five benchmark methods (A-LOAM, LeGO-LOAM, DLO, MOLA, and AG-LOAM) across four progressively challenging testing contexts: a controlled outdoor environment with traffic cones, and three real fruit orchard datasets (CitrusFarm, Bacchus, and Pullally). The following subsections synthesize the key findings, identify the conditions under which each method performs best, and discuss the trade-offs between accuracy, robustness, and platform generality.

### *4.1. Performance in the controlled environment*

The controlled outdoor environment, comprising a simple rectangular trajectory around traffic cones on flat terrain, provides the clearest comparison of baseline localization accuracy under conditions where all methods operate without the confounds of uneven terrain or variable vegetation. The proposed approach achieves the lowest mean position error ($\bar{e}_p = 0.079 \pm 0.009$ m) and the lowest RMS position error ($e_{RMS_p} = 0.092 \pm 0.011$ m) among all methods (Table 3). LeGO-LOAM and AG-LOAM are the closest competitors, with RMS position errors of

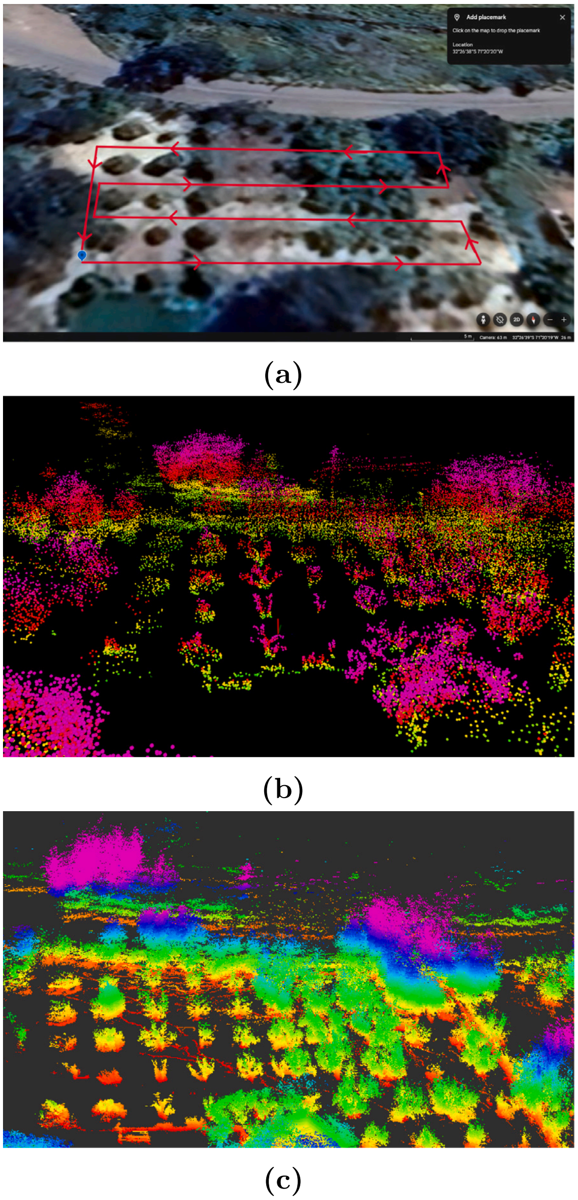


**Fig. 9.** Pullally experimental fruit orchard: (a) Satellite image view with robot's reference path; (b) 3D reconstruction from GNSS ground truth; (c) 3D reconstruction from proposed approach.

0.156 m and 0.394 m respectively, while A-LOAM ($e_{RMS_p}$ = 1.289 m), MOLA (2.430 m), and DLO (6.400 m) exhibit substantially larger errors.

The difficulty experienced by DLO and MOLA in this environment is noteworthy. DLO relies on dense 3D point cloud matching and, in the absence of sufficiently distinctive geometric features (traffic cones are small, sparse targets), the matching degenerates and produces large positional drift ($e_{p\%}$ = 41.08%). MOLA's velocity-integrated ICP, while robust to aggressive platform motion in general, also struggles here, yielding $e_{p\%}$ = 14.24%. Both methods were originally designed and validated primarily on rich, feature-dense outdoor or forest environments, and the sparse cone geometry of the controlled experiment represents an adverse condition for feature-based 3D matching. In contrast, the proposed method's 2D projection and Hausdorff distance matching are well-suited to compact, vertically elongated targets even when sparse, because the projection concentrates the relevant structural information into a stable 2D point set.

AG-LOAM performs reasonably well ($e_{RMS_p}$ = 0.394 m) relative to DLO and MOLA, benefiting from its adaptive map-update mechanism that rejects motion-distorted points during the legged robot's gait cycles. LeGO-LOAM achieves competitive mean errors but with higher variance across runs ($C_{RMS_p}$ = 0.054 m), reflecting its sensitivity to ground-plane segmentation quality on uneven surfaces. The proposed approach is the most consistent across all seven replicate runs in this environment (Fig. 7), confirming that the probabilistic 2D projection and modified Hausdorff distance matching provide a stable and repeatable localization solution even under the challenging conditions imposed by the legged robot's impulsive gait.

### 4.2. *Performance in real orchard environments*

#### 4.2.1. *CitrusFarm dataset*

On the CitrusFarm dataset, collected with a wheeled robot on relatively smooth terrain, A-LOAM ($e_{RMS_p}$ = 0.282 m), LeGO-LOAM

**Table 5**
Comparative summary of the agricultural lidar datasets.

| Feature | CitrusFarm | Bacchus | Pullally |
|---|---|---|---|
| Platform | Clearpath Jackal | Saga Robotics Thorvald II | Unitree Go1 |
| Motion type | Differential drive skid-steer | 4-wheel-drive & 4-wheel-steer (4WD4S) | Legged quadruped robot |
| Motion smoothness | High | Medium | Low |
| Field type | Citrus farm | Vineyard | Mixed citrus, peaches, plums, almond experimental orchard. |
| Path length | ~865.33 m | ~102.3 m | ~150.0 m |
| Run duration | ~12.7 min | ~18.5 min | ~42.0 min |
| Average speed | ~1.0 m/s | ~0.6 m/s | ~0.3 m/s |
| Lidar | Velodyne VLP-16 | Ouster OS1-16 | Velodyne VLP-16 |
| Lidar rate | 10 Hz | 10 Hz | 4 Hz |
| Lidar resolution (H × V) | 0.20° × 2.0° | 0.18° × 2.2° | 0.40° × 2.0° |
| IMU | MicroStrain 3DM-GX5-AHRS | Internal Ouster OS1-16 | None |
| IMU rate | ~200 Hz | ~100 Hz | — |
| GNSS | SwiftNav Duro | Trimble BX992 | ArduSimple RTK3BPro |
| GNSS rate | ~10 Hz | ~10 Hz | ~1 Hz |
| Robot control signal information | No | No | Yes (~85 Hz) |
| Same path runs | 1 | 1 | 3 |
| Test area | 60 × 48 $m^2$ | 50 × 12 $m^2$ | 46 × 28 $m^2$ |
| Dataset size | 6.3 GB | 54 GB | 26.5 GB |

**Table 6**
Pose estimation errors for the CitrusFarm dataset (Sequence 05). Position $p$ errors in meters and angular $\theta$ errors in radians. *Note: CitrusFarm has a single pass ($N_e = 1$) acquisition, thus confidence intervals cannot be computed.*

| Pose estimation metrics | CitrusFarm dataset | | | | | |
|---|---|---|---|---|---|---|
| | Proposed approach | A-LOAM | LeGOLOAM | DLO | MOLA | AGLOAM |
| $\bar{e}_p$ | 1.344 | 0.236 | 0.140 | 0.357 | 26.239 | 0.317 |
| $\bar{e}_\theta$ | 0.050 | 0.032 | 0.025 | 0.042 | 0.415 | 0.052 |
| $e_{RMS_p}$ | 1.725 | 0.282 | 0.160 | 0.407 | 33.252 | 0.360 |
| $e_{RMS_\theta}$ | 0.074 | 0.053 | 0.038 | 0.073 | 0.558 | 0.078 |
| $\bar{e}_{\Delta p}$ | −0.011 | 0.005 | 0.005 | −0.003 | −0.174 | 0.005 |
| $\bar{e}_{\Delta\theta}$ | 0.000 | −0.001 | −0.001 | −0.021 | −0.100 | −0.034 |
| $e_{RMS_{\Delta p}}$ | 0.065 | 0.037 | 0.061 | 0.060 | 1.195 | 0.071 |
| $e_{RMS_{\Delta\theta}}$ | 0.073 | 0.051 | 0.049 | 0.091 | 0.392 | 0.099 |
| $e_{p\%}$ | 0.330 | 0.058 | 0.034 | 0.088 | 6.446 | 0.078 |
| $e_{end}$ | 3.199 | 0.065 | 0.201 | 0.161 | 66.203 | 0.372 |

(0.160 m), DLO (0.407 m), and AG-LOAM (0.360 m) all achieve sub-meter RMS position errors (Table 6), with the proposed approach recording a higher error of 1.725 m. This pattern is consistent with expectations: on smooth wheeled platforms, the 3D matching pipelines of A-LOAM, LeGO-LOAM, and DLO benefit from continuous, distortion-free point clouds, allowing precise feature extraction and matching in full 3D. The proposed approach, which deliberately discards the vertical dimension and retains only 2D trunk-level projections, necessarily trades some accuracy for robustness when 3D information is both abundant and reliable.

MOLA is a notable exception among the benchmark methods, recording a catastrophic RMS error of 33.252 m on CitrusFarm. This failure is attributed to its constant-velocity assumption being violated by the long straight-row segments and abrupt turns of the CitrusFarm trajectory, which produce large velocity discontinuities incompatible with its motion model. The very high end-point error ($e_{end}$ = 66.203 m)

**Table 7**

Pose estimation errors for the Bacchus dataset (Sequence June 8th, 2022). Position $p$ errors in meters and angular $\theta$ errors in radians. *Note: Bacchus dataset has a single pass* ($N_e = 1$) *acquisition, thus confidence intervals cannot be computed.*

| Pose estimation metrics | Bacchus dataset | | | | | |
|---|---|---|---|---|---|---|
| | Proposed approach | A-LOAM | LeGOLOAM | DLO | MOLA | AGLOAM |
| $\bar{e}_p$ | 0.975 | 1.766 | 5.404 | 32.653 | 34.108 | 31.103 |
| $\bar{e}_\theta$ | 0.561 | 0.397 | 0.284 | 0.906 | 0.636 | 0.647 |
| $e_{RMS_p}$ | 1.126 | 2.113 | 5.649 | 35.408 | 36.122 | 33.861 |
| $e_{RMS_\theta}$ | 1.042 | 0.870 | 0.702 | 1.460 | 1.219 | 1.209 |
| $\bar{e}_{\Delta p}$ | 0.022 | 0.013 | 0.084 | 0.482 | 0.106 | 0.389 |
| $\bar{e}_{\Delta\theta}$ | 0.059 | −0.017 | 0.136 | −0.637 | −0.582 | −0.228 |
| $e_{RMS_{\Delta p}}$ | 0.664 | 0.310 | 0.226 | 0.854 | 0.891 | 0.742 |
| $e_{RMS_{\Delta\theta}}$ | 1.048 | 0.872 | 0.644 | 1.488 | 1.393 | 0.998 |
| $e_{p\%}$ | 0.930 | 1.683 | 5.152 | 31.125 | 32.512 | 29.648 |
| $e_{end}$ | 0.564 | 1.236 | 5.191 | 31.900 | 36.740 | 37.925 |

**Table 8**

Pose estimation errors for the Pullally dataset. Position $p$ errors in meters and angular $\theta$ errors in radians. *Note: Pullally results are averaged over three repetitions of the acquisition trajectory* ($N_e = 3$) *with 95% confidence intervals.*

| Pose estimation metrics | Pullally dataset | | | | | |
|---|---|---|---|---|---|---|
| | Proposed approach | A-LOAM | LeGO-LOAM | DLO | MOLA | AG-LOAM |
| $\bar{e}_p \pm C_p$ | 0.402 ± 0.054 | 21.304 ± 2.819 | 2.948 ± 3.106 | 13.552 ± 1.072 | 16.777 ± 3.016 | 1.608 ± 2.154 |
| $\bar{e}_\theta \pm C_\theta$ | 0.674 ± 0.133 | 1.025 ± 0.446 | 0.693 ± 0.076 | 1.114 ± 0.314 | 1.711 ± 0.213 | 0.956 ± 0.301 |
| $e_{RMS_p} \pm C_{RMS_p}$ | 0.470 ± 0.064 | 27.490 ± 5.644 | 4.141 ± 4.027 | 17.729 ± 1.783 | 21.446 ± 3.598 | 2.123 ± 2.513 |
| $e_{RMS_\theta} \pm C_{RMS_\theta}$ | 1.124 ± 0.234 | 1.271 ± 0.402 | 1.105 ± 0.219 | 1.334 ± 0.386 | 2.008 ± 0.134 | 1.392 ± 0.331 |
| $\bar{e}_{\Delta p} \pm C_{\Delta p}$ | −0.003 ± 0.001 | −0.003 ± 0.003 | 0.000 ± 0.002 | −0.065 ± 0.072 | 0.124 ± 0.130 | −0.036 ± 0.027 |
| $\bar{e}_{\Delta\theta} \pm C_{\Delta\theta}$ | 0.043 ± 0.022 | 0.004 ± 0.014 | 0.041 ± 0.030 | −0.144 ± 0.127 | 0.018 ± 0.106 | −0.113 ± 0.083 |
| $e_{RMS_{\Delta p}} \pm C_{RMS_{\Delta p}}$ | 0.080 ± 0.019 | 0.053 ± 0.008 | 0.079 ± 0.019 | 0.326 ± 0.210 | 0.838 ± 0.334 | 0.203 ± 0.066 |
| $e_{RMS_{\Delta\theta}} \pm C_{RMS_{\Delta\theta}}$ | 0.368 ± 0.099 | 0.239 ± 0.043 | 0.384 ± 0.077 | 0.567 ± 0.177 | 0.484 ± 0.053 | 0.487 ± 0.118 |
| $e_{p\%} \pm C_{p\%}$ | 0.268 ± 0.033 | 14.237 ± 2.030 | 1.950 ± 2.040 | 9.042 ± 0.650 | 11.183 ± 1.884 | 1.078 ± 1.448 |
| $e_{end} \pm C_{end}$ | 0.276 ± 0.127 | 53.484 ± 19.259 | 5.286 ± 4.085 | 18.713 ± 12.283 | 23.298 ± 11.952 | 2.799 ± 4.437 |

**Table 9**

Average total processing time per lidar scan and computation rate of the SLAM algorithms.

| Algorithm | Time [ms] | Maximum computation rate [Hz] |
|---|---|---|
| A-LOAM | 284 | 3.6 |
| LeGO-LOAM | 266 | 3.8 |
| DLO | 56 | 17.9 |
| MOLA | 111 | 9.0 |
| AG-LOAM | 93 | 10.8 |
| Proposed method | 543 | 1.8 |

confirms that MOLA diverges on this trajectory rather than simply accumulating slow drift.

Among the benchmark methods that succeed on CitrusFarm, DLO and AG-LOAM achieve results comparable to A-LOAM in terms of RMS error, but with larger cumulative errors as measured by $e_{p\%}$ (DLO: 0.088%; AG-LOAM: 0.078% vs. A-LOAM: 0.058% and LeGO-LOAM: 0.034%), indicating that LeGO-LOAM provides the best drift control on smooth wheeled-platform trajectories in this environment. The proposed approach records $e_{p\%} = 0.330\%$ and $e_{end} = 3.199$ m, reflecting primarily the cumulative heading errors at trajectory turns (Fig. 10), which are a known limitation of the 2D projection approach when turns reduce the number of distinct matched landmarks.

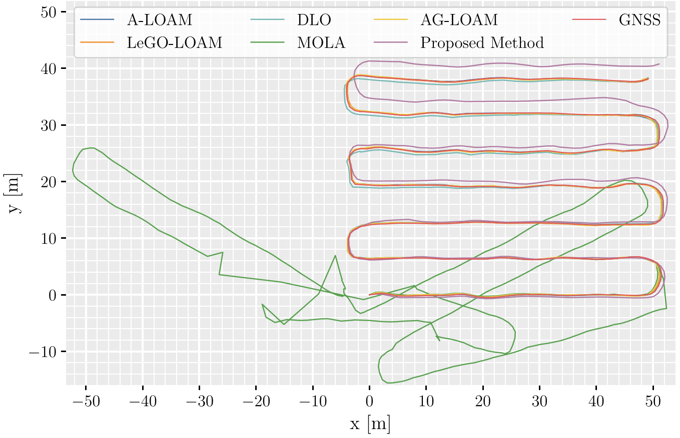
### 4.2.2. Bacchus dataset

The Bacchus vineyard dataset, characterized by a uniform trellis canopy forming long featureless corridors, represents the most challenging environment for feature-based 3D SLAM. The degeneracy of the corridor geometry causes all 3D benchmark methods to produce large errors: DLO ($e_{RMS_p} = 35.408$ m), MOLA (36.122 m), and AG-LOAM (33.861 m) all fail to maintain localization along the vineyard alleys. LeGO-LOAM also diverges substantially ($e_{RMS_p} = 5.649$ m), and even A-LOAM, which performed well on CitrusFarm, degrades significantly ($e_{RMS_p} = 2.113$ m). The proposed approach is the only method to achieve a sub-2 m RMS error ($e_{RMS_p} = 1.126$ m) and the smallest positional error over path length ($e_{p\%} = 0.930\%$), as well as the smallest end-point error ($e_{end} = 0.564$ m vs. 1.236 m for A-LOAM, Table 7).

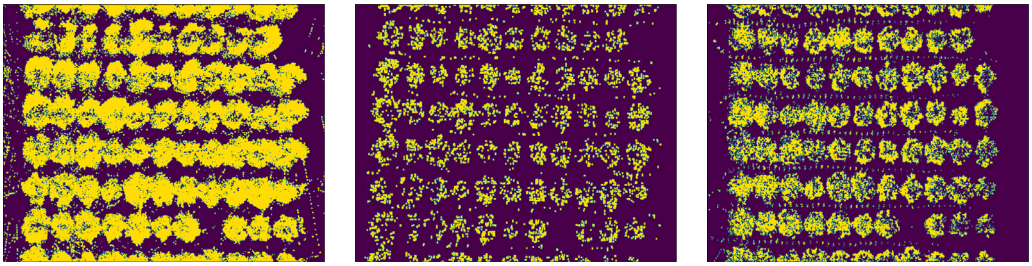
The degradation of DLO, MOLA, and AG-LOAM on Bacchus warrants specific discussion. DLO's dense matching relies on sufficient 3D geometric variability; the planar wall-like structure of a trellis vineyard in late spring with dense foliage provides almost no distinctive geometric features, causing scan matching to degenerate. Similarly, MOLA's ICP-based approach, even augmented by velocity integration, cannot compensate for the absence of salient landmarks in the repetitive corridor geometry. AG-LOAM's adaptive mapper, which rejects motion-distorted points, exacerbates the problem here: in a feature-sparse environment, the already-limited set of distinctive points is further reduced by the adaptive rejection step, leaving insufficient constraints for reliable pose estimation. The proposed method's robustness on Bacchus stems directly from its use of 2D trunk projections: even in a trellis vineyard, the vine posts and trunk bases at ground level produce a distinct, repeatable 2D point pattern that is more discriminative than the 3D foliage returns used by the benchmark methods.

### 4.2.3. Pullally dataset

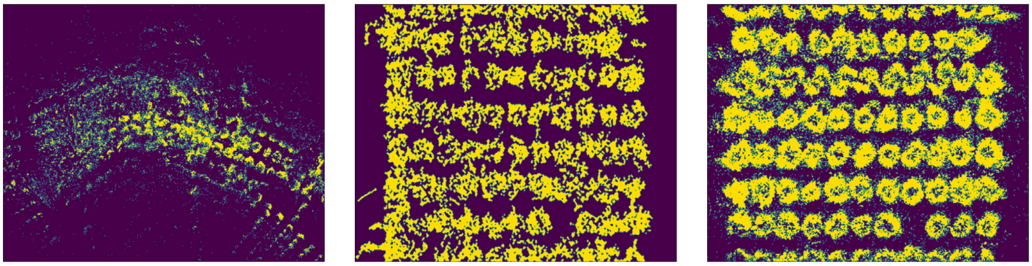
The Pullally orchard dataset, collected with a Unitree Go1 quadruped robot traversing mixed-species orchard rows on highly



**(a)** Estimated trajectories



**(b)** A-LOAM **(c)** LeGO-LOAM **(d)** DLO



**(e)** MOLA **(f)** AG-LOAM **(g)** Proposed method

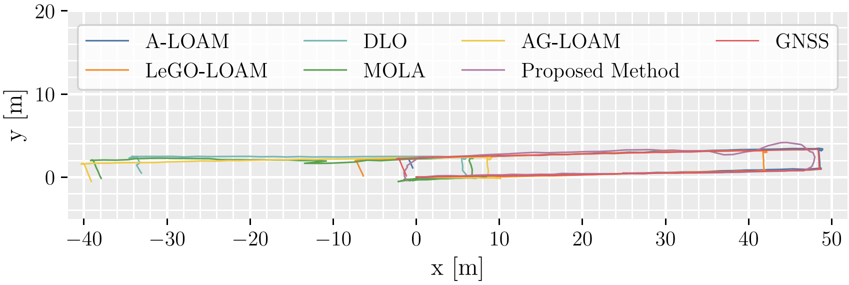
**Fig. 10.** Estimated trajectories and reconstructed maps for the CitrusFarm dataset (Sequence 05) by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method. Map area is 60 × 48 m with 0.01 m/pixel resolution.

uneven terrain with three replicate runs, exposes the most significant divergence between the proposed approach and all benchmark methods. As shown in Table 8, the proposed approach achieves consistent performance across all three runs ($e_{RMS_p} = 0.470 \pm 0.064$ m), while A-LOAM fails catastrophically ($27.490 \pm 5.644$ m), DLO diverges substantially ($17.729 \pm 1.783$ m), and MOLA also fails ($21.446 \pm 3.598$ m). LeGO-LOAM and AG-LOAM show intermediate behavior: LeGO-LOAM achieves a mean RMS error of 4.141 m, but with very high variance (±4.027 m), succeeding in one of three runs, but diverging in the other two, likely due to its dependency on proper ground-plane segmentation; AG-LOAM performs more consistently than LeGO-LOAM but still records $e_{RMS_p} = 2.123 \pm 2.513$ m with large run-to-run variability.

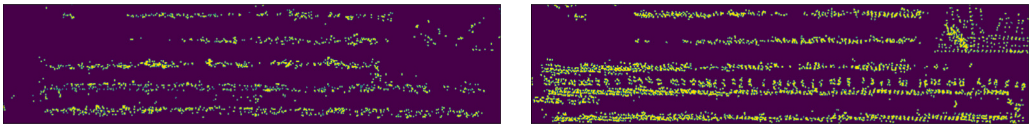
The failure of A-LOAM on the Pullally dataset is explained by its strong dependence on IMU integration for point cloud distortion correction: the impulsive accelerations generated by the robot's trotting gait produce discontinuous lidar sweeps that corrupt feature extraction, causing the scan-to-scan matching to diverge at each turn. DLO, despite operating without IMU integration, relies on dense 3D matching that is similarly sensitive to the inter-scan discontinuities introduced by the legged gait. MOLA's constant-velocity model, while effective at smoothing motion on continuous platforms, is directly violated by the impulsive nature of quadruped locomotion, leading to poor initial alignment estimates for ICP at each scan.

AG-LOAM's motion stability criterion, which suppresses map updates during rough-terrain traversal, partially mitigates the impact of the legged gait and explains its better-than-average performance ($e_{RMS_p} = 2.123$ m) relative to A-LOAM, DLO, and MOLA. However, the large confidence interval (±2.513 m) reveals that AG-LOAM's performance is not reliable across runs, likely because its map-update suppression logic, calibrated for wheeled rough-terrain platforms, is overly conservative for the quadruped's faster gait cycle, intermittently starving the map of updates and causing localization drift at turns.

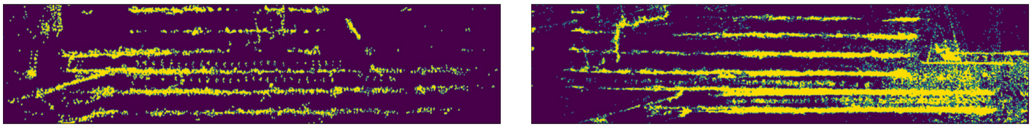
The proposed approach is robust to all of these failure modes because it does not rely on 3D feature extraction, point cloud distortion correction, or inter-scan continuity. The 2D ground-plane projection naturally filters out the vertical perturbations introduced by the legged



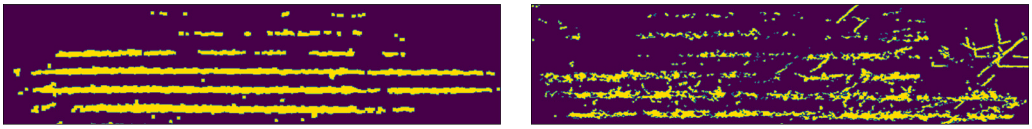
(a) Estimated trajectories



(b) A-LOAM (c) LeGO-LOAM

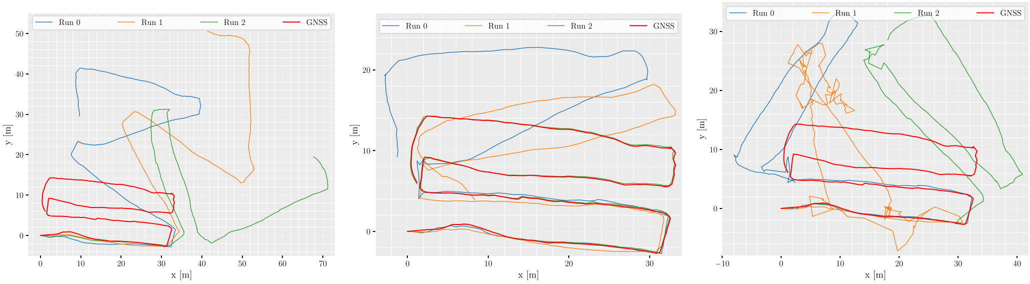


(d) DLO (e) MOLA

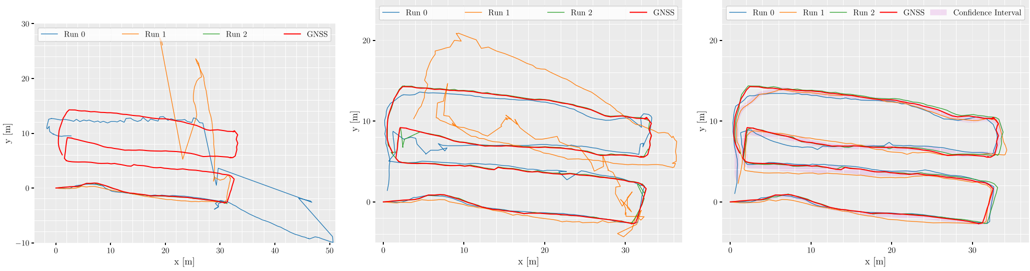


(f) AG-LOAM (g) Proposed method

**Fig. 11.** Estimated trajectories and reconstructed maps for the Bacchus dataset (Sequence June 8th, 2022) by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method. Map area is 50 × 12 m with 0.01 m/pixel resolution.

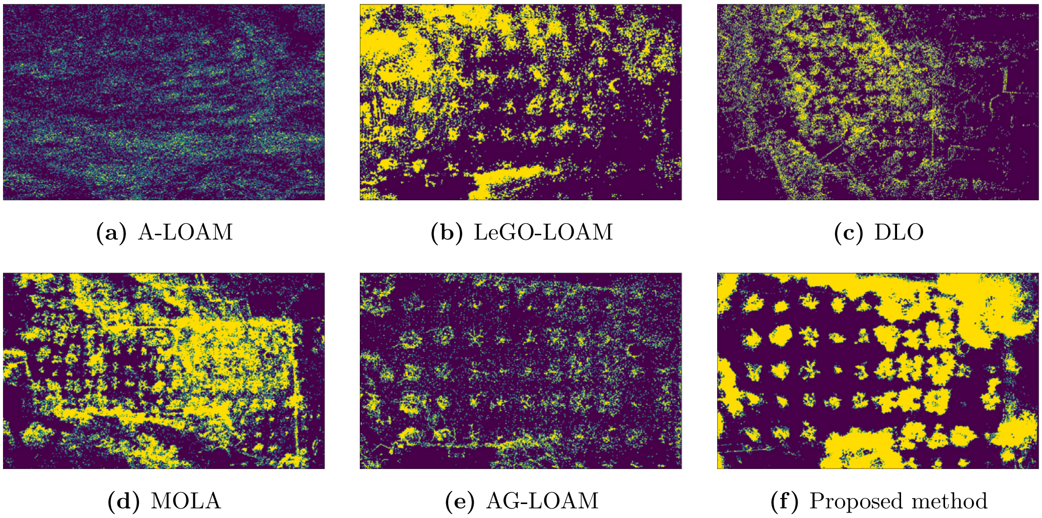


(a) A-LOAM (b) LeGO-LOAM (c) DLO



(d) MOLA (e) AG-LOAM (f) Proposed method

**Fig. 12.** Estimated trajectories for the Pullally dataset by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method.



(a) A-LOAM (b) LeGO-LOAM (c) DLO

(d) MOLA (e) AG-LOAM (f) Proposed method

**Fig. 13.** Estimated final maps for the Pullally dataset by A-LOAM, LeGO-LOAM, DLO, MOLA, AG-LOAM, and the proposed method. Map area is 46x28 m with 0.01 m/pixel resolution.

gait, and the modified Hausdorff distance's robustness to outliers absorbs the residual misalignment in projected scans caused by platform roll and pitch. The resulting $e_{p\%} = 0.268 \pm 0.033\%$ and $e_{end} = 0.276 \pm 0.127$ m confirm that drift is well-controlled across the full 150 m trajectory and across all three repetitions.

### 4.3. Performance comparison across platforms and environments

Taken together, the results across all four experimental contexts reveal a clear and consistent pattern. The proposed approach is the only method that achieves competitive or best performance across *all* evaluated environments and platforms. In contrast, each of the five benchmark methods excels under specific conditions but degrades significantly under others, as summarized below.

A-LOAM and LeGO-LOAM perform best on smooth wheeled platforms in structured environments (CitrusFarm), where continuous point cloud acquisition and reliable IMU-assisted distortion correction enable precise 3D feature matching. Both methods degrade substantially on the Bacchus trellis vineyard and fail on the Pullally legged-robot dataset. DLO shares the same sensitivity to platform continuity and feature availability: it is competitive on CitrusFarm but collapses on Bacchus and Pullally. MOLA fails on the CitrusFarm straight-and-turn trajectory due to velocity-model violations and on Pullally due to legged gait discontinuities, but performs moderately on the controlled environment. AG-LOAM is the most capable of the benchmark methods on the legged-robot Pullally dataset, benefiting from its motion stability criterion, but still shows unacceptably high variance and cannot match the consistency of the proposed approach.

The proposed method shows its principal limitation on the CitrusFarm dataset, where the reduction to 2D projection sacrifices the accuracy achievable through full 3D matching on smooth wheeled platforms. This trade-off is, however, intentional and quantifiable: the CitrusFarm RMS error of 1.725 m is the cost of a design that provides sub-0.5 m errors and low variance on the Pullally legged-robot dataset where all benchmark methods fail, and sub-1.5 m errors on the feature-degenerate Bacchus dataset where all benchmark methods also fail. For practitioners, the key decision criterion is therefore platform dynamics and environment geometry: methods with full 3D matching and IMU integration are preferable for wheeled robots on smooth terrain with feature-rich 3D structure; the proposed approach is preferable for legged, handheld, or humanoid platforms, for environments with uniform or sparse 3D features, or whenever IMU reliability cannot be guaranteed due to impulsive locomotion or GNSS denial.

### 4.4. Limitations and directions for future work

Several limitations of the present study merit explicit discussion. First, the CitrusFarm and Bacchus evaluations are based on single trajectory acquisitions ($N_e = 1$), which precludes the computation of confidence intervals and limits the statistical characterization of performance variability for those datasets. Creating new datasets of different arboreal environments with multiple runs, as available for the Pullally experiments, are planned as part of future work to provide a complete statistical comparison.

Second, the proposed method's 2D projection discards elevation information, which limits its applicability to platforms operating on approximately planar terrain. On very steep slopes or multi-level arboreal environments, the ground-plane projection assumption may introduce systematic errors not fully captured by the EKF process noise model. Extending the approach to a full 6-DOF pose estimation framework with terrain elevation mapping would address this limitation.

Finally, all five benchmark methods and the proposed approach were evaluated without loop closure. Incorporating a loop closure mechanism, particularly for long repetitive orchard traversals, would reduce cumulative drift for all methods and constitutes an important direction for future work.

## 5. Conclusion

This paper has presented a lidar-only SLAM approach for arboreal agricultural environments that eliminates the dependency on inertial odometry and GNSS positioning. The method operates by probabilistically projecting 3D lidar scans onto the ground plane and solving a 2D scan-to-map matching problem using the modified Hausdorff distance, making it well suited to platforms where IMU integration is unreliable: specifically, legged robots, handheld mapping systems, and humanoid platforms where intermittent leg–ground contact forces corrupt inertial measurements. The method was evaluated in simulation, in a controlled outdoor environment, and across three real-world orchard datasets (CitrusFarm, Bacchus, and Pullally), comparing it against five benchmark methods spanning the range from classic feature-based (A-LOAM, LeGO-LOAM) to state-of-the-art dense and agriculturally adapted approaches (DLO, MOLA, AG-LOAM).

The results show that the proposed approach trades peak accuracy on smooth wheeled platforms for consistent robustness across platform types and environments. The 2D ground-plane projection naturally suppresses the vertical perturbations introduced by the quadruped's trotting gait, and the modified Hausdorff distance's robustness to outliers absorbs the residual scan misalignment caused by platform roll and pitch, explaining this consistent advantage under discontinuous motion. While benchmark methods achieve better accuracy on smooth wheeled platforms (CitrusFarm), they fail on discontinuous motion platforms (Pullally), whereas our approach maintains stable performance across both conditions. A limitation of this study is that the CitrusFarm and Bacchus evaluations are based on single trajectory runs due to dataset availability, while the Pullally dataset includes three replicate runs. Ongoing work involves creating new arboreal environment datasets with multiple runs of the same paths across all datasets and ground truth grid maps to better characterize performance variability of the SLAM algorithms.

## CRediT authorship contribution statement

**Paola Nazate-Burgos:** Writing – original draft, Writing – review & editing, Visualization, Methodology, Investigation, Formal analysis, Data curation. **Miguel Torres-Torriti:** Writing – original draft, Writing – review & editing, Supervision, Software, Resources, Project administration, Funding acquisition, Conceptualization. **Shoudong Huang:** Writing – original draft, Supervision, Investigation, Formal analysis. **Fernando Auat Cheein:** Writing – review & editing, Validation, Supervision, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This project has been supported by the National Agency of Research and Development (ANID), Chile under grants Fondecyt 1220140, ANID CIA250006, and Doctoral grant 21212303.

## Appendix A. Supplementary data

The Supplementary material related to this article is available at: https://github.com/RAL-UC/RoSA_SLAM.

## Data availability

The source code is available at https://github.com/RAL-UC/RoSA_SLAM. The dataset is available at: https://github.com/RAL-UC/Pullally_Dataset.

## References

Aguiar, A.S., Neves dos Santos, F., Sobreira, H., Boaventura-Cunha, J., Sousa, A.J., 2022. Localization and mapping on agriculture based on point-feature extraction and semiplanes segmentation from 3D LiDAR data. Front. Robot. AI 9, http://dx.doi.org/10.3389/frobt.2022.832165.

Auat Cheein, F., Steiner, G., Perez Paina, G., Carelli, R., 2011. Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. Comput. Electron. Agric. 78 (2), 195–207. http://dx.doi.org/10.1016/j.compag.2011.07.007.

Bhattarai, U., Zhang, Q., Karkee, M., 2024. Design, integration, and field evaluation of a robotic blossom thinning system for tree fruit crops. J. Field Robot. 41 (5), 1366–1385. http://dx.doi.org/10.1002/rob.22330.

Blanco-Claraco, J.L., 2025. A flexible framework for accurate LiDAR odometry, map manipulation, and localization. Int. J. Robot. Res. 44 (9), 1553–1599. http://dx.doi.org/10.1177/02783649251316881.

Bustos, J., Donoso, F., Guesalaga, A., Torres-Torriti, M., 2007. Matching radar and satellite images for ship trajectory estimation using the Hausdorff distance. IET Radar, Sonar & Navig. 1, 50–58. http://dx.doi.org/10.1049/iet-rsn:20060025.

Calderara-Cea, F., Torres-Torriti, M., Auat Cheein, F., Delpiano, J., 2024. A two-stage deep learning strategy for weed identification in grassfields. Comput. Electron. Agric. 225, 109300. http://dx.doi.org/10.1016/j.compag.2024.109300.

Chen, K., Lopez, B.T., Agha-mohammadi, A.-a., Mehta, A., 2022. Direct LiDAR odometry: Fast localization with dense point clouds. IEEE Robot. Autom. Lett. 7 (2), 2000–2007. http://dx.doi.org/10.1109/LRA.2022.3142739.

Cremona, J., Comelli, R., Pire, T., 2022. Experimental evaluation of visual-inertial odometry systems for arable farming. J. Field Robot. 39 (7), 1121–1135. http://dx.doi.org/10.1002/rob.22099.

Danielsson, P.-E., 1980. Euclidean distance mapping. Comput. Graph. Image Process. 14 (3), 227–248. http://dx.doi.org/10.1016/0146-664X(80)90054-4.

Debeunne, C., Vivet, D., 2020. A review of visual-LiDAR fusion based simultaneous localization and mapping. Sensors 20 (7), http://dx.doi.org/10.3390/s20072068.

Donoso-Aguirre, F., Bustos-Salas, J.-P., Torres-Torriti, M., Guesalaga, A., 2008. Mobile robot localization using the hausdorff distance. Robotica 26 (2), 129–141. http://dx.doi.org/10.1017/S0263574707003657.

Gené-Mola, J., Gregorio, E., Auat Cheein, F., Guevara, J., Llorens, J., Sanz-Cortiella, R., Escolà, A., Rosell-Polo, J.R., 2020. Fruit detection, yield prediction and canopy geometric characterization using LiDAR with forced air flow. Comput. Electron. Agric. 168, 105121. http://dx.doi.org/10.1016/j.compag.2019.105121.

Grisetti, G., Stachniss, C., Burgard, W., 2007. Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Trans. Robot. 23 (1), 34–46. http://dx.doi.org/10.1109/TRO.2006.889486.

Grupp, M., 2017. evo: Python package for the evaluation of odometry and SLAM. https://github.com/MichaelGrupp/evo.

Hess, W., Kohler, D., Rapp, H., Andor, D., 2016. Real-time loop closure in 2D LIDAR SLAM. In: 2016 IEEE International Conference on Robotics and Automation. ICRA, pp. 1271–1278. http://dx.doi.org/10.1109/ICRA.2016.7487258.

Jiang, J., Zhang, T., Li, K., 2025. LiDAR-based 3D SLAM for autonomous navigation in stacked cage farming houses: An evaluation. Comput. Electron. Agric. 230, 109885. http://dx.doi.org/10.1016/j.compag.2024.109885.

Kefalas, A., Kalampokas, T., Vrochidou, E., Papakostas, G.A., 2025. A vision-based pruning algorithm for cherry tree structure elements segmentation and exact pruning points determination. Comput. Electron. Agric. 237, 110735. http://dx.doi.org/10.1016/j.compag.2025.110735.

Kohlbrecher, S., von Stryk, O., Meyer, J., Klingauf, U., 2011. 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. http://dx.doi.org/10.1109/SSRR.2011.6106777.

Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleiner, A., 2009. On measuring the accuracy of SLAM algorithms. Auton. Robots 27, 387–407. http://dx.doi.org/10.1007/s10514-009-9155-6.

Li, Y., An, J., He, N., Li, Y., Han, Z., Chen, Z., Qu, Y., 2025. A review of simultaneous localization and mapping algorithms based on lidar. World Electr. Veh. J. 16 (2), http://dx.doi.org/10.3390/wevj16020056.

Li, Q., Zhu, H., 2024. Performance evaluation of 2D LiDAR SLAM algorithms in simulated orchard environments. Comput. Electron. Agric. 221, 108994. http://dx.doi.org/10.1016/j.compag.2024.108994.

Liu, X., Nardari, G.V., Ojeda, F.C., Tao, Y., Zhou, A., Donnelly, T., Qu, C., Chen, S.W., Romero, R.A.F., Taylor, C.J., Kumar, V., 2022. Large-scale autonomous flight with real-time semantic SLAM under dense forest canopy. IEEE Robot. Autom. Lett. 7 (2), 5512–5519. http://dx.doi.org/10.1109/LRA.2022.3154047.

Mahto, M.K., Srivastava, S.K., Sah, B., 2024. Introduction to smart farming. In: Smart Agritech: Robotics, AI, and Internet of Things (IoT) in Agriculture. John Wiley & Sons, Ltd, pp. 1–28. http://dx.doi.org/10.1002/9781394302994.ch1.

Morin, P., Samson, C., 2008. Motion control of wheeled mobile robots. In: Siciliano, B., Khatib, O. (Eds.), Springer Handbook of Robotics. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 799–826. http://dx.doi.org/10.1007/978-3-540-30301-5_35.

Mutz, F., Oliveira-Santos, T., Forechi, A., Komati, K.S., Badue, C., França, F.M., De Souza, A.F., 2021. What is the best grid-map for self-driving cars localization? An evaluation under diverse types of illumination, traffic, and environment. Expert Syst. Appl. 179, 115077. http://dx.doi.org/10.1016/j.eswa.2021.115077.

Navone, A., Martini, M., Chiaberge, M., 2025. Autonomous robotic pruning in orchards and vineyards: A review. Smart Agric. Technol. 12, 101283. http://dx.doi.org/10.1016/j.atech.2025.101283.

Nazate-Burgos, P., Torres-Torriti, M., Aguilera-Marinovic, S., Arévalo, T., Huang, S., Auat Cheein, F., 2025-05-23. Robust 2D lidar-based SLAM in arboreal environments without IMU/GNSS. In: Workshop "Novel Approaches for Precision Agriculture and Forestry with Autonomous Robots" of the IEEE International Conference on Robots and Automation (ICRA 2025). pp. 1–7, arXiv:2505.10847.

Nie, F., Zhang, W., Wang, Y., Shi, Y., Huang, Q., 2022. A forest 3-D lidar SLAM system for rubber-tapping robot based on trunk center atlas. IEEE/ASME Trans. Mechatronics 27 (5), 2623–2633. http://dx.doi.org/10.1109/TMECH.2021.3120407.

Peralta-Cabezas, J.L., Torres-Torriti, M., Guarini-Hermann, M., 2008. A comparison of Bayesian prediction techniques for mobile robot trajectory tracking. Robotica 26 (5), 571–585. http://dx.doi.org/10.1017/S0263574708004153.

Polvara, R., Molina, S., Hroob, I., Papadimitriou, A., Tsiolis, K., Giakoumis, D., Likothanassis, S., Tzovaras, D., Cielniak, G., Hanheide, M., 2024. Bacchus long-term (BLT) data set: Acquisition of the agricultural multimodal BLT data set with automated robot deployment. J. Field Robot. 41 (7), 2280–2298. http://dx.doi.org/10.1002/rob.22228.

Qin Tong, C.S., 2019. A-LOAM: Advanced implementation of LOAM. https://github.com/HKUST-Aerial-Robotics/A-LOAM.

Ryu, J., Kamata, S., 2021. An efficient computational algorithm for hausdorff distance based on points-ruling-out and systematic random sampling. Pattern Recognit. 114, 107857. http://dx.doi.org/10.1016/j.patcog.2021.107857.

Shan, T., Englot, B., 2018. 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. http://dx.doi.org/10.1109/IROS.2018.8594299.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 5135–5142. http://dx.doi.org/10.1109/IROS45743.2020.9341176.

Siciliano, B., Khatib, O. (Eds.), 2008. Springer Handbook of Robotics. Springer, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-540-30301-5.

Sier, H., Li, Q., Yu, X., Peña Queralta, J., Zou, Z., Westerlund, T., 2023. A benchmark for multi-modal LiDAR SLAM with ground truth in GNSS-denied environments. Remote. Sens. 15 (13), http://dx.doi.org/10.3390/rs15133314.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., 2012. A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580. http://dx.doi.org/10.1109/IROS.2012.6385773.

Teng, H., Wang, Y., Chatziparaschis, D., Karydis, K., 2025. Adaptive LiDAR odometry and mapping for autonomous agricultural mobile robots in unmanned farms. Comput. Electron. Agric. 232, 110023. http://dx.doi.org/10.1016/j.compag.2025.110023.

Teng, H., Wang, Y., Song, X., Karydis, K., 2023. Multimodal dataset for localization, mapping and crop monitoring in citrus tree farms. In: Bebis, G., Ghiasi, G., Fang, Y., Sharf, A., Dong, Y., Weaver, C., Leo, Z., LaViola Jr., J.J., Kohli, L. (Eds.), Advances in Visual Computing. Springer Nature Switzerland, Cham, pp. 571–582. http://dx.doi.org/10.1007/978-3-031-47969-4_44.

Thrun, S., 1998. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99 (1), 21–71. http://dx.doi.org/10.1016/S0004-3702(97)00078-7.

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

Torres-Torriti, M., Nazate-Burgos, P., 2022. SLAM in agriculture. In: Encyclopedia of Smart Agriculture Technologies. Springer International Publishing, Cham, pp. 1–22. http://dx.doi.org/10.1007/978-3-030-89123-7_235-1.

Torres-Torriti, M., Nazate-Burgos, P., Paredes-Lizama, F., Guevara, J., Auat Cheein, F., 2022. Passive landmark geometry optimization and evaluation for reliable autonomous navigation in mining tunnels using 2D lidars. Sensors 22 (8), http://dx.doi.org/10.3390/s22083038.

Underwood, J.P., Hung, C., Whelan, B., Sukkarieh, S., 2016. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. Comput. Electron. Agric. 130, 83–96. http://dx.doi.org/10.1016/j.compag.2016.09.014.

Wang, X., Liang, X., Campos, M., Zhang, J., Wang, Y., 2024. Benchmarking of laser-based simultaneous localization and mapping methods in forest environments. IEEE Trans. Geosci. Remote Sens. 62, http://dx.doi.org/10.1109/TGRS.2024.3439438.

Xia, Y., Lei, X., Pan, J., Chen, L., Zhang, Z., Lyu, X., 2023. Research on orchard navigation method based on fusion of 3D SLAM and point cloud positioning. Front. Plant Sci. 14, http://dx.doi.org/10.3389/fpls.2023.1207742.

Yandun Narvaez, F., Reina, G., Torres-Torriti, M., Kantor, G., Auat Cheein, F., 2017. A survey of ranging and imaging techniques for precision agriculture phenotyping. IEEE/ASME Trans. Mechatronics 22 (6), 2428–2439. http://dx.doi.org/10.1109/TMECH.2017.2760866.

Zhang, J., Singh, S., et al., 2014. LOAM: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems, vol. 2, (no. 9), Berkeley, CA, pp. 1–9.

Zi, B., Wang, H., Santos, J., Zheng, H., 2020. A new metric for assessing the performance of 2D lidar SLAMs. In: Afli, H., Bleimann, U., Burkhardt, D., Loew, R., Regier, S., Stengel, I., Wang, H., Zheng, H.J. (Eds.), Proceedings of the 6th Collaborative European Research Conference. CERC 2020, Belfast, Northern-Ireland, UK, September 10–11, 2020, In: CEUR Workshop Proceedings, vol. 2815, CEUR-WS.org, pp. 64–77, URL: https://ceur-ws.org/Vol-2815/CERC2020_paper04.pdf.