2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Vinymap: a Vineyard Inspection and 3D Reconstruction Framework for Agricultural Robots

Ioannis Zarras[1], Athanasios Mastrogeorgiou[1], Konstaninos Machairas[1],
Konstantinos Koutsoukis[1], *Student members, IEEE,* and Evangelos Papadopoulos,[1] *Life Fellow, IEEE*

***Abstract*— Efficient and thorough vineyard inspection is crucial for optimizing yield and preventing disease from spreading. Manual approaches are labor-intensive and prone to human error, motivating the development of automated solutions. Precision viticulture benefits greatly from access to photo-realistic 3D vineyard maps and from capturing intricate visual details necessary for accurate canopy and grape health assessment. Generating such maps efficiently proves challenging, particularly when employing cost-effective equipment. This paper presents a novel vineyard inspection and 3D reconstruction framework implemented on a Robotic Platform (RP) equipped with three stereo cameras. The framework's performance was evaluated on an experimental synthetic vineyard developed at NTUA. This testing setup allowed experimentation under diverse lighting conditions, ensuring the system's robustness under realistic scenarios. Unlike existing solutions, which often focus on specific aspects of the inspection, our framework offers a top-down approach, encompassing autonomous navigation, high-fidelity 3D reconstruction, and canopy growth assessment. The developed software is available at the Control Systems Laboratory's (CSL) bitbucket repository [1].**

I. INTRODUCTION

Earth's growing population and the ever-present climate change pose a significant challenge to global food availability. Precision agriculture (PA) practices and automated crop inspection have emerged as key strategies to make agriculture more efficient and sustainable. One key sector is vineyards due to their structured layouts and the high value of the crops [2]. However, several challenges must be addressed before robots can be integrated fully into vineyard management. One significant obstacle is the obstructed satellite signal in mountainous vineyard regions. This renders Global Positioning Systems (GPS), an essential component for robot navigation, unreliable. Uneven terrain within the vineyards and obstacles like rocks, irrigation equipment, and workers necessitate developing robust and adaptable path-planning solutions. Furthermore, cost-effectiveness remains a critical concern when it comes to widespread adoption. Many existing solutions in vineyard robotics rely heavily on GPS [3], rendering them ineffective in GPS-denied [21] environments. Other approaches address specific aspects of automation like spraying [4], grape counting [5], or disease detection [6], lacking a holistic solution. The former do not offer a comprehensive 3D representation of the vine trees, which is valuable for crop inspection. Also, research often neglects the crucial aspects of autonomous navigation and obstacle avoidance within the vineyard [22], [23]. This fragmentation limits their effectiveness in providing a complete picture for the vine grower. Recent research demonstrates the possibility of generating photo-realistic 3D maps utilizing 3D Gaussians [7]. While valuable, such solutions typically require immense computing power [8], [9], making them prohibitively expensive for many growers and hampering real-time capability, which is essential since fast decision-making is crucial for farmers to react to sudden weather events [10], [11].

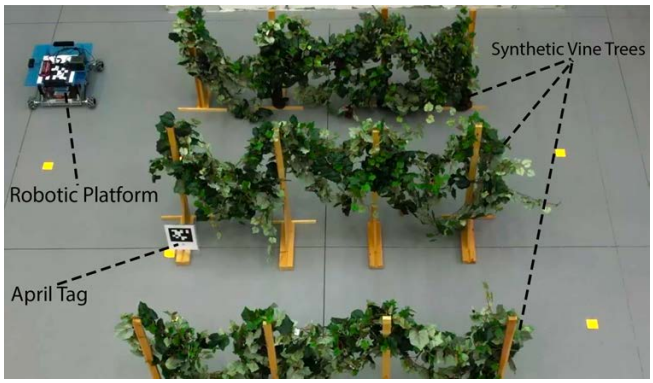

Fig. 1. Drone view of the rover in the synthetic vineyard developed at the Control Systems Lab of the National Technical University of Athens.

Given these limitations, we present Vinymap: a Vineyard Inspection And 3D Reconstruction Framework For Agricultural Robots. Vinymap employs a robust dual-camera visual odometry (VO) system designed explicitly for GPS-denied environments with limited visual features available [12]. This redundancy ensures continuous operation even when one camera is obstructed by sunlight or texture-less obstacles. We enhance its effectiveness by improving loop closure with AprilTag Detection by adding a visual servoing algorithm that ensures proper alignment with the AprilTag before initiating loop closure corrections. Moreover, Vinymap leverages a path-planning algorithm that uses polytopic decomposition to create obstacle-free corridors [13]. This approach, implemented for the first time on a real robot, benefits from the vineyard's corridor-like layout. Vinymap delivers comprehensive vineyard inspection through two key algorithms: 1) A photo-realistic 3D mapping algorithm that generates fast

[1]The authors are with the Control Systems Lab (CSL), School of Mechanical Engineering, National Technical University of Athens, Greece. e-mails: janniszar@gmail.com, {amast, kmach, kkoutsou, egpapado}@central.ntua.gr.

This work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 2182).

The authors would like to thank K. S. Narkhede, PhD Candidate at the University of Delaware, for providing the code of [13].

visual representations of the vineyard. 2) A canopy density assessment algorithm that provides vital insights into grape health and quality, as dense foliage can limit sunlight and air circulation around the grape clusters [24]. Vinymap's effectiveness was initially evaluated in realistic simulations. We subsequently constructed a synthetic vineyard at CSL to extensively test the framework on an actual robotic platform under realistic conditions.

II. Experimental Setup

A. Realistic Grapevine Canopy

A vineyard with artificial grapes and leaves was built at CSL to conduct controlled and repeatable experiments with varying light conditions. Each row consists of multiple plants on a trellis system, so the vegetation form resembles a natural canopy. The basic vineyard row parameters, such as the distance between plants ($\sim 1m$) and grapes' minimum height ($0.60m$), are based on standard viticulture practices in Greece. The artificial grapes' grid features varying density and grape size, creating different visibility conditions since some grapes are partly covered with leaves, whereas others lie on the front plane. The vineyard consists of three 3-meter-long rows on even terrain.

B. CSL Rover

A custom robotic platform (RP) was used to validate and test the proposed software framework. The RP was designed and constructed for research purposes, comprising of in-house built and off-the-shelf parts (e.g., aluminum profiles, bearing units, etc.). Its motion system features four mecanum wheels [14] to provide the robot with omnidirectional motion capabilities. The wheels are powered by four Maxon DC motors (RE 35) combined with planetary gearboxes (GP 42) and incremental encoders (HEDL 5540), providing $5Nm$ of continuous torque per wheel.

The perception system of the RP was designed to host a dual-camera VO algorithm [12], as well as the Vinymap framework. Thus, the RP had three stereo cameras (Fig. 2). The front and rear cameras are Stereolabs ZED2 cameras. They are utilized for VO, while the right side camera is a ZED X Mini Camera with a $2.2mm$ focal length. It is used for the 3D reconstruction and canopy density assessment tasks. The ZED X Mini is also fitted with polarizing lenses, which increase the RP's ability to operate under diverse lighting conditions.

The software framework runs on the NVIDIA Jetson Orin AGX. The ZED2 cameras are connected to the device via USB 3.0, while the ZED X Mini is connected via a GMSL2 port. The Orin is powered by a $12V$, $1200mAh$ XP-1 lithium battery, providing power for more than 30 minutes.

C. Simulation Environment

Before deployment within the synthetic vineyard, our framework was validated through a set of simulation experiments. Leveraging the Gazebo simulation platform, we constructed a replica of the synthetic vineyard, the CSL rover, and the ZED Depth cameras. The use of appropriate simulation plugins and accurate STL files ensured high fidelity to the actual hardware configuration.

D. Experimental Pipeline

Our framework employs a pipeline to achieve navigation and concurrent 3D reconstruction within the vineyard. Fig. 4 illustrates the pipeline in detail. The process unfolds as follows:

Objective Initiation: The robot begins its trajectory at a predetermined starting location within the vineyard. A user-defined target position within the designated vineyard perimeter serves as the navigation goal.

Trajectory Planning: The path planning algorithm [13] uses a pre-existing vineyard map to decompose the environment and compute an obstacle-free trajectory for the RP to follow.

Real-Time Localization: The dual-camera VO algorithm developed at CSL [12] continuously estimates the robot's position within the vineyard. This VO algorithm incorporates the loop closure functionality, which is triggered whenever the robot encounters an AprilTag. The AprilTags are periodically placed within the vineyard.

Velocity Control: To ensure the robot follows the planned trajectory, a tracking-PID controller is employed. Utilizing the desired waypoints and the robot's current position, it outputs the appropriate velocity commands published using the Robot Operating System (ROS).

Command Execution & Motion: The published velocity messages are received via a UDP interface on the robot's onboard Raspberry Pi. Subsequently, they are transmitted to the Roboclaw microcontrollers through serial communication, dictating the robot's motion and ensuring the planned trajectory is followed.

Continuous 3D Reconstruction & Inspection: The platform utilizes its dedicated ZED X camera to capture visual data throughout the navigation process. The data are used from our inspection and 3D reconstruction software, to create a 3D map with valuable spatial information.

E. Loop Closure with AprilTags

The dual-camera VO algorithm implements a loop closure mechanism that leverages AprilTags placed periodically within the environment to address potential drift associated

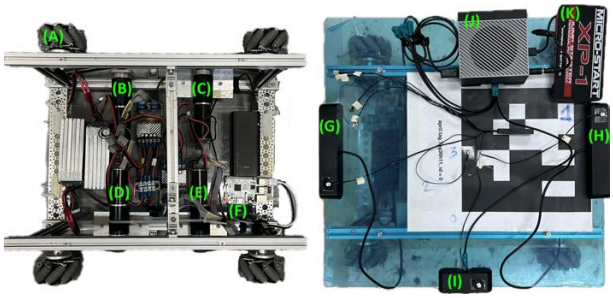

Fig. 2. The robotic platform and the developed vision system: (A) mecanum wheel, (B), (C), (D), (E) Maxon Motors, (F) Raspberry Pi, (G), (H) ZED2 Cameras, (I) ZED X Mini Camera, (J) Nvidia Orin AGX, (K) XP-1 Battery.

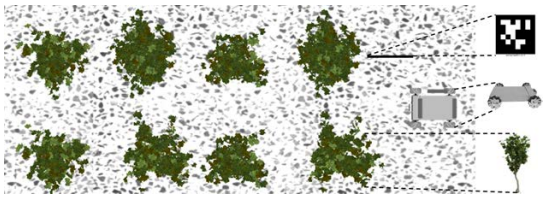

Fig. 3. Simulated Environment in Gazebo.

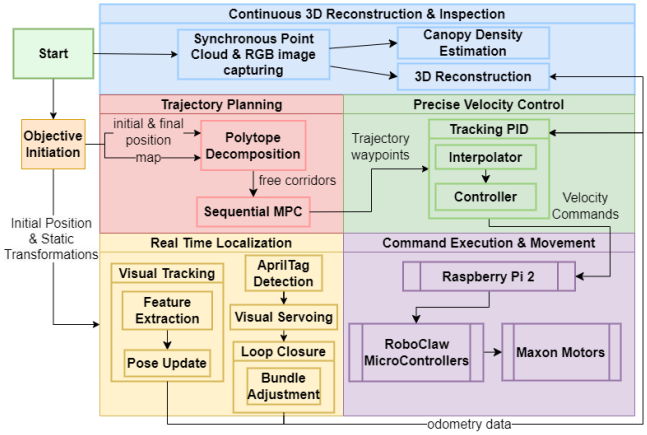

Fig. 4. Experimental Pipeline.

with visual navigation. These fiducial markers are placed in a priori known positions and orientations, allowing the robot to determine its absolute pose upon detection with the ZED X camera. We observed performance degradation in loop closure accuracy when AprilTags were detected at oblique angles or from distances greater than $50cm$. In such scenarios, the tag information yielded inaccurate pose estimates, worsening odometry drift after loop closure. We implemented a corrective process triggered upon AprilTag detection to mitigate this effect. The robot performs visual servoing to reach a pre-defined distance ($30cm$) & ensures a viewing angle close to the AprilTag normal. This pose facilitates an accurate inference of the robot's transformation relative to the tag, enabling subsequent loop closure & pose-graph optimization to refine the odometry measurements effectively. The application of this algorithm resulted in a reduction in loop closure errors & improved VO performance.

To investigate the impact of both radial distance and viewing angle on the accuracy of pose estimation, we conducted several experiments as illustrated in Fig. 5a. Two radial distances from the AprilTag were evaluated: $30cm$ and $50cm$. Each semicircle was subdivided into 12 sections, representing view angle deviations of $15°$. The findings, presented in Fig. 5b, demonstrate a significant reduction in transformation error as the viewing angle approached $90°$. Furthermore, a radial distance of $30cm$ yielded more accurate pose estimations compared to a $50cm$ radial distance. The transformation error at a $90°$ view angle and at a radial distance of $30cm$ is zero.

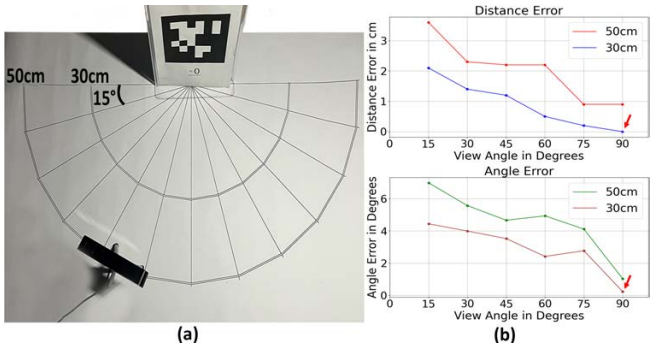

Fig. 5. (a) The experimental setup for identifying the optimal camera view angle & distance from the AprilTag for loop closure. (b) Distance and Angle error of AprilTag pose from various viewing angles. The transformation error at a 90° view angle and a radial distance of 30cm is zero.

III. THE VINYMAP FRAMEWORK

The Vinymap framework leverages synchronized point clouds and images captured by the ZED X camera to comprehensively represent the vineyard canopy and assess its vegetative state. It is implemented in Python with Open3D and OpenCV, and its key functionalities include a Point Cloud Quality Enhancement pipeline, a Canopy Density Assessment algorithm, and a photo-realistic 3D Reconstruction algorithm.

A. Point Cloud Quality Enhancement

Vinymap involves cleaning the raw input point clouds through a series of statistical methods based on [25]. This step eliminates noise and outliers, ensuring the integrity and reliability of subsequent processing stages. This process: 1) Loads and formats the raw point clouds. It removes invalid points and downsamples for memory efficiency. 2) Crops point clouds using bounding boxes to exclude areas near the boundaries of ZED X's field of view, which were noisy. 3) Removes points with a low number of neighboring points within a local radius, targeting areas with high noise. 4) Removes clusters identified by DBSCAN clustering [15] that fall below a size threshold, indicating noise. 5) Estimates surface normals for each point using covariance analysis of neighboring points. The point cloud quality enhancement algorithm is always called before any other process.

B. Canopy Density Assessment

Our framework incorporates a novel strategy for identifying gaps within the vineyard canopy. This method capitalizes on the inherent color properties of the vine tree leaves, as well as the topological properties of surface reconstruction with alpha shapes [16]. We implement a Canopy Density Assessment (CDA) algorithm to output a comprehensive Canopy Density Index (CDI) for each input point cloud batch. This index enables vine growers to pinpoint increased foliage density that could restrict sunlight and air circulation around the grape clusters or decreased foliage density, which might indicate impaired growth. After the quality of the point clouds is enhanced, CDA calls the following functions:

PointCloudRegistration: This function merges individual point cloud scans captured from successive viewpoints as the RP traverses the vineyard and leverages KISS-ICP [17]. The function implements a tiered ICP approach utilizing three distinct point correspondence distances. This sequential strategy, employing a coarse-to-fine logic, progressively refines the alignment between successive scans. The function's primary output is a unified point cloud map fm encompassing all the input batch's scans stitched together through the registration process. Additionally, the function provides a list of individual transformations tr. Each transformation corresponds to a specific input point cloud and details the precise positional and rotational adjustments necessary to integrate it into the overall map.

CanopySeparation: This function operates on the fused point cloud map fm generated by the PointCloudRegistration function. It isolates points within the map exhibiting green hues. This targeted color-based segmentation filters out extraneous elements, resulting in a distinct point cloud ic, that exclusively encompasses the vineyard's green canopy.

FullCanopyEstimation: Reconstructs a mesh m from the ic point cloud acquired by the CanopySeparation function. The reconstruction excludes all gaps and irregularities by employing a large alpha parameter.

CanopyGapExtraction: The mesh m produced from the FullCanopyEstimation is sampled, and a point cloud fc representing the gapless full canopy is acquired. The initial canopy point cloud ic is subtracted from the full canopy point cloud using distance-based subtraction. The resulting point cloud gc represents the gaps in the canopy.

The CDA algorithm then calculates the CDI by dividing the number of points of the output point cloud from the CanopyGapExtraction gc by the number of points of the full canopy point cloud fc, as seen in (1), after downsampling both of the clouds using the same voxel size. Fig. 6 illustrates the CDA process in detail.

$$CDI = 1 - \frac{NumberOfPoints(gc)}{NumberOfPoints(fc)} \tag{1}$$

C. Photo-Realistic Vineyard Reconstruction

To generate a visually informative and spatially accurate representation of the vineyard, Vinymap utilizes projective texture mapping. The Photo-realistic Vineyard Reconstruction (PVR) algorithm receives a set of point clouds P and a set of images I as its input. Each point cloud $p_i \in P$ must be captured simultaneously with a corresponding image $I_i \in I$ as the robot moves. This is achieved with ROS's synchronization policy. PVR effectively projects high-resolution RGB images onto meshes reconstructed from the raw point cloud data using alpha shapes. The meshes are then registered and aligned, resulting in a photorealistic 3D representation of the vineyard. This process is presented in Algorithm 1 and illustrated in Fig. 6. After enhancing the point clouds using point cloud quality enhancement, PVR calls the following functions:

SurfaceReconstruction: This function creates a triangle mesh m_i that accurately represents a source point cloud p_i. This is achieved by leveraging the ball pivoting algorithm (BPA) [18] and fusing its results with the alpha shapes [16] reconstruction algorithm.

ProjectiveTextureMapping: This function receives a triangle mesh m_i and an RGB image I_i as input. For each triangle vertex $v \in m_i$, the function projects it to the 2D image plane of I_i and matches it with a pixel $q \in m_i$ with coordinates $(u, v) \in I_i$, according to (2).

$$\boldsymbol{q} = \boldsymbol{M}\boldsymbol{v}, \tag{2}$$

where:

$$\boldsymbol{q} = \begin{bmatrix} u \\ v \end{bmatrix}, \boldsymbol{M} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{v} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

The $\boldsymbol{M}$ matrix is the intrinsic matrix of the camera that captured I_i, in our case, the ZED X mini camera. The (u,v) coordinates are used to apply texture to the so-far colorless mesh m_i. Provided that the RGB image I_i and the point cloud p_i were captured simultaneously, the resulting texture application produces a photo-realistic 3D mesh r_i. The sum of all the photorealistic meshes is symbolized with R.

PointCloudRegistration: Merges individual point cloud scans captured from successive viewpoints as the RP traverses the vineyard. Described in section III B.

MeshFusion: The output transformations tr from the PointCloudRegistration are used to transform each reconstructed mesh $r_i \in R$ identically. Thus, the ICP cloud registration is used to register the meshes accurately. The aligned and registered meshes are then fused, and the function outputs the result of a photo-realistic vineyard mesh y.

Algorithm 1 Photo-realistic Vineyard Reconstruction

```
Given: P = {p_1, ..., p_k}, I = {I_1, ..., I_k}, alpha, M
B = PointCloudQualityEnhancement(P)
for p_i in B do
    m_i = SurfaceReconstruction(p_i, alpha)
    r_i = ProjectiveTextureMapping(m_i, I_i, M)
end for
fm, tr = PointCloudRegistration(B)
y = MeshFusion(tr)
return y
```

IV. EXPERIMENTAL RESULTS

A. Canopy Density Assessment

To test the performance of our canopy density assessment algorithm, we conducted experiments in both the simulated and the synthetic vineyard setting. The simulation offered a highly controlled environment with almost no canopy density variations. We tested the CDA on dense, perfect canopies, resulting in very high-density index scores, as expected. The laboratory setting provided a more realistic yet controlled environment. We tested two distinct scenarios:

Moderately sparse canopy: CDA returned a density index within an expected range, aligning with the visual assessment of the canopy.

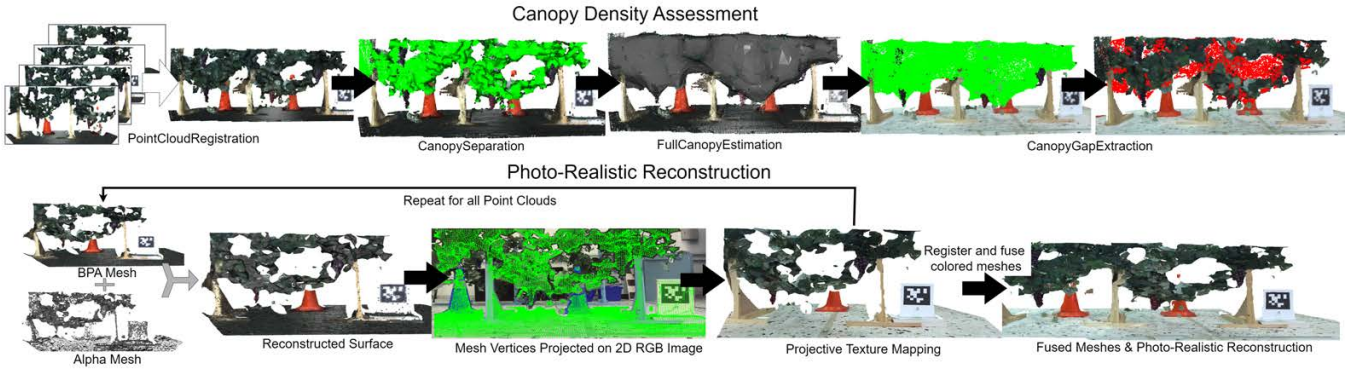

Fig. 6. Pipeline illustrations of the (a) Canopy density assessment, (b) Photorealistic vineyard Reconstruction.

Sparse canopy: The presence of gaps resulted in a significantly lower density index, validating CDA's ability to detect and quantify vegetation irregularities.

Laboratory tests were repeated under natural and artificial lighting. All tests yielded accurate and intuitively useful results, reflecting CDA's simplicity and effectiveness in assessing canopy density. Table 1 and Fig. 7 display the results in detail.

Fig. 7. Canopy Density Assessment Scenarios.

B. Reconstruction Quality

Regarding visual fidelity and information richness, the PVR algorithm is compared to the raw point cloud data and Stereolabs' ZED Spatial Mapping [19] algorithm. A direct comparison with the raw point cloud data readily reveals the difference. While the raw data offers an essential structural representation, our reconstructed mesh delivers a significantly clearer depiction of the vineyard. As presented in Fig. 8, the AprilTag exhibits enhanced clarity, and individual grapes within grape clusters are more readily discernible.

Compared to the high-quality mesh output of Stereolabs' ZED Spatial Mapping algorithm at its highest settings, our reconstruction emerges as the superior solution. The projective texture mapping approach generates meshes featuring significantly greater visual detail, enabling vine growers to inspect individual leaves and grapes meticulously. This enhanced texture facilitates more informed decision-making in remote precision agriculture.

TABLE I. CANOPY DENSITY INDEX OUTPUT.

Experiment Type		Canopy Type	CDI
Simulated		Dense	0.742
Laboratory	Artificial Lighting	Moderately Sparse	0.512
	Natural Lighting		0.569
Laboratory	Artificial Lighting	Sparse	0.450
	Natural Lighting		0.461

Fig. 8. Reconstruction Comparison.

C. Real-Time Viability

To evaluate the real-time capabilities of our algorithm, we captured data using ROSbags and SVO files from ZED cameras, allowing for offline review, timing analysis, and benchmarking against state-of-the-art methods. This assessment was conducted on the NVIDIA Jetson Orin platform. We compared the Vinymap framework with RTAB-map [20] and ZED Spatial Mapping [19].

Our algorithm demonstrates real-time processing capabilities, performing on par with the ZED Spatial Mapping algorithm at its high settings. The time needed for reconstructing

a single vineyard row is $\sim 68s$ with ZED Spatial Mapping and $\sim 51s$ with Vinymap. While RTAB-Map exhibits faster execution ($\sim 30s$ for a single row), it occasionally experiences delays due to loop closure estimation calculations. In contrast, our approach strategically triggers loop closure only upon AprilTag detection, eliminating these delays.

The CDA and PVR algorithms achieve real-time viability. Notably, the desired output quality of Vinymap can influence the robot's operational speed. Vine growers can select the optimal solution for their needs by carefully balancing robot speed and desired reconstruction detail. In the case of Vinymap, a 40% overlap between captured point clouds is required for effective ICP registration. Assuming the robot moves in a straight line perpendicular to the canopy, (3) yields the maximum robot speed which achieves the 40% overlap,

$$s = fps * fov * (1 - overlap) \tag{3}$$

where s is the maximum movement speed, fps is the frequency of point cloud and RGB image capturing, fov is the length of the visible canopy at each frame in meters and is dependent on the distance between the robot and the canopy, and $overlap$ is the required overlap in volume percentage between the captured point clouds. With the capability of running with $fps = 2$, $fov = 0.9m$, and a necessary overlap of 0.4, Vinymap is currently suitable for running in real-time on robots with movement speeds lower than $1.08m/s$. The RP on which our framework was tested was moving at about $0.12m/s$, comfortably within the limit.

V. CONCLUSION

This paper presented a novel framework for 3D reconstruction and vineyard canopy analysis. Demonstrated in a synthetic vineyard, our approach: (a) enhanced VO localization by utilizing AprilTags and applying a visual servoing step before triggering loop closure optimization, (b) achieved real-time viability, (c) achieved crisp 3D reconstruction compared to the ZED spatial mapping approach, (d) demonstrated a complete inspection solution integrating state-of-the-art path planning approaches based on polytope decomposition applied for the first time on a real robotic platform. Regarding future research directions, integrating deep learning and automated reasoning techniques could unlock complex inspection tasks, such as detecting diseases or highlighting potential risks for the vine grower to further examine. Additionally, extending autonomous inspection could enable complete vineyard coverage and monitoring without needing pre-defined start and finish points. We strongly believe this framework is a solid first step towards a complete solution for precision viticulture.

REFERENCES

[1] https://bitbucket.org/csl_legged/vinymap_iros24/src/main/

[2] G. Pappalardo, A. Scienza, G. Vindigni, and D. Mario, "Profitability of wine grape growing in the EU member states," Journal of Wine Research, vol. 24, Mar. 2013, doi: 10.1080/09571264.2012.724392.

[3] J. M. Bengochea-Guevara, J. Conesa-Muñoz, D. Andújar, and A. Ribeiro, "Merge Fuzzy Visual Servoing and GPS-Based Planning to Obtain a Proper Navigation Behavior for a Small Crop-Inspection Robot," Sensors, vol. 16, no. 3, Art. no. 3, Mar. 2016.

[4] F. P. Terra, G. R. A. da Rosa, J. J. P. Prado, and P. L. J. Drews-, "A Low-Cost Prototype to Automate Agricultural Sprayers∗," IFAC-PapersOnLine, vol. 53, no. 2, pp. 15835–15840, Jan. 2020.

[5] L. Shen et al., "Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s," Computers and Electronics in Agriculture, vol. 206, p. 107662, Mar. 2023.

[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Front. Plant Sci., vol. 7, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering." arXiv, Aug. 08, 2023. doi: 10.48550/arXiv.2308.04079.

[8] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting." arXiv, Dec. 06, 2023. doi: 10.48550/arXiv.2312.10070.

[9] N. Keetha et al., "SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM." arXiv, Dec. 04, 2023.

[10] T. Thoai, R. J. Ranola, and L. Camacho, "The Importance of Weather Forecasts and Meteorological Information in Adaptation to Climate Change in Agricultural Production: Some Preliminary Findings," Philippine Agricultural Scientist, vol. 101, pp. 377–392, Dec. 2018.

[11] U. Kumar et al., "Role of Information in Farmers' Response to Weather and Water Related Stresses in the Lower Bengal Delta, Bangladesh," Sustainability, vol. 12, no. 16, Art. no. 16, Jan. 2020.

[12] C. Kokas, A. Mastrogeorgiou, K. Machairas, K. Koutsoukis and E. Papadopoulos, "Multicamera Visual SLAM For Vineyard Inspection," 2024 32nd Mediterranean Conference on Control and Automation (MED), Chania - Crete, Greece, 2024, pp. 75-80, doi: 10.1109/MED61351.2024.10566198.

[13] K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, and I. Poulakakis, "A Sequential MPC Approach to Reactive Planning for Bipedal Robots." arXiv, Apr. 30, 2022. doi: 10.48550/arXiv.2205.00156.

[14] M. Mutalib and N. Azlan, "Prototype Development of Mecanum Wheels Mobile Robot: A review," Applied Research and Smart Technology (ARSTech), vol. 1, pp. 71–82, Nov. 2020.

[15] D. Deng, "DBSCAN Clustering Algorithm Based on Density," in 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Sep. 2020, pp. 949–953.

[16] S. Asaeedi, F. Didehvar, and A. Mohades, "Alpha Convex Hull, a Generalization of Convex Hull," Sep. 2013.

[17] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," IEEE Robot. Autom. Lett., vol. 8, no. 2, pp. 1029–1036, Feb. 2023.

[18] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pp. 349–359, Jul. 1999, doi: 10.1109/2945.817351.

[19] "Spatial Mapping Overview - Stereolabs." Accessed: Mar. 09, 2024. [Online]. Available: https://www.stereolabs.com/docs/spatial-mapping

[20] "RTAB-Map," RTAB-Map. Accessed: Mar. 09, 2024. [Online]. Available: http://introlab.github.io/rtabmap/

[21] A. Costley and R. Christensen, "Landmark Aided GPS-Denied Navigation for Orchards and Vineyards," in 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Apr. 2020, pp. 987–995.

[22] S. Kelly et al., "Target-Aware Implicit Mapping for Agricultural Crop Inspection," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 9608-9614.

[23] L. Srinivas, A. Bharathy, S. Ramakuri, A. Sethy, and R. Kumar, "An optimiZED machine learning framework for crop disease detection," Multimedia Tools and Applications, vol. 83, pp. 1–20, May 2023.

[24] S. E. Spayd, J. M. Tarara, D. L. Mee, and J. C. Ferguson, "Separation of Sunlight and Temperature Effects on the Composition of Vitis vinifera cv. Merlot Berries," Am J Enol Vitic., vol. 53, no. 3, pp. 171–182, Jan. 2002, doi: 10.5344/ajev.2002.53.3.171.

[25] H. Balta, J. Velagic, W. Bosschaerts, G. De Cubber, and B. Siciliano, "Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments," IFAC-PapersOnLine, vol. 51, no. 22, pp. 348–353, Jan. 2018, doi: 10.1016/j.ifacol.2018.11.566.