Multicamera Visual SLAM For Vineyard Inspection

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

***Abstract*— Automating the process of collecting samples (e.g., images) from a vineyard can help to monitor the condition of the grapes with precision and prevent the spreading of diseases. A critical part of this task is the development of a robust localization algorithm so that (a) a robot is able to carry out the inspection process and (b) the vine-grower knows exactly which part of the vineyard has been inspected. In this paper, we propose a novel approach for enhancing the robustness of vSLAM by utilizing multiple stereo cameras and a novel method for detecting loops in homogeneous environments based on AprilTags, where state-of-the-art approaches may find it difficult to detect them. We test the accuracy of our method using a wheeled Robotic Platform (RP) in simulation and in a synthetic vineyard developed at CSL, NTUA [1]. The developed method achieves high accuracy in the localization of the RP in the vineyard and robustness even when a featureless object covers a large part of the Field of View of one camera. The developed software is available for testing at the CSL's bitbucket repository [2].**

I. Introduction

Recently, robots started making their first steps towards real-world applications in agriculture and, more specifically, in vineyards [3]. Grapes have to be visually inspected so that their condition is accurately assessed. To automate this task, cm-level position accuracy and the ability to maneuver in tight spaces are required. While solutions such as RTK GPS promise to provide such accuracy levels, vineyards are usually located in remote, GPS-denied, and challenging areas; therefore, an alternative localization method must be utilized. Visual Simultaneous Localization and Mapping (vSLAM) uses camera feedback to determine the robot's position within a map it creates. As a result, it can be used in situations where the sky is obstructed by environmental factors, e.g. high crops, trees, etc. Additionally, it is a less expensive alternative to RTK GPS, as it requires only visual feedback from the cameras mounted on the robot.

In recent years, many vSLAM approaches have been proposed using the indirect or feature method, requiring either monocular [4] - [8], or stereo [8] - [13] feedback. PTAM [5] was the first algorithm to introduce the use of keyframes for mapping. Many monocular vSLAM algorithms build upon PTAM, i.e.: ORB-SLAM [6] and VINS [7]. These algorithms operate in real time, and use a constant velocity model to track features in consecutive images and estimate the camera's trajectory over time. To mitigate any accumulative error and to detect loop closure, the real-time Bundle Adjustment (BA) is used [14]. The BA simultaneously refines the position of the 3D landmarks and the pose of the keyframes. ORBSLAM, VINS and Kimera [8] use DBoW2 [15] for loop detecting, which converts images into a bag-of-word representation for loop closure detection.

Fig. 1: Drone view of the CSL rover in the synthetic vineyard developed in the lab.

In vineyards, the crop is organized in parallel rows. There, the bag-of-word approach could result in incorrect loop detection due to the high similarity between images. As a result, such vSLAM approaches are prone to false positive loop detection in homogeneous environments. To resolve these issues, we propose a multicamera vSLAM approach that utilizes two cameras for trajectory estimation and one for AprilTag [16] & loop closure detection. The synthetic vineyard and the developed platform used in the experiments are presented in Fig. 1.

Our approach achieves robust localization in feature-poor environments since the Field of View (FoV) is doubled, resulting in more features available to track, and in situations in which one of the cameras is obscured by the sun or leaves. In addition, to address the challenge of image similarity, loop closure is performed when the robot detects a registered AprilTag. This technique enables the algorithm to re-localize the camera within the created map without having to necessarily revisit a previous location since the pose of the AprilTag is pre-known. The performance of the developed

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" (Project Number: 2182).

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

framework is compared to ORB-SLAM3 [12], a state-of-the-art vSLAM algorithm. Multicam systems have also been proposed, but they are either not open source or use other lens types (fisheye) [17] [19] [20]. In our approach, with the use of AprilTags the algorithm has the maximum confidence that the same point in the vineyard was visited in the past and uses this information for accurate loop detection.

This paper is organized as follows: In Sect. II the experimental setup is presented, while in Sect. III the developed vSLAM approach is introduced, focusing on feature selection, AprilTag, and loop closure detection. In Sect. IV experimental results are discussed, and the developed approach is compared with a SOTA method and the respective ground truth. Last, in Sect. V we conclude and propose future research directions.

II. Experimental Setup

A. Realistic Grapevine Canopy

To perform experiments easily with varying and controlled light conditions, a vineyard with artificial grapes and leaves was built at CSL (Fig. 1). Each row consists of multiple plants on a trellis system so that the canopy 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 common 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 4-meter-long rows on even terrain (Fig. 1).

B. CSL's rover

A wheeled robotic platform (RP) was used to validate the concept (Fig, 2). The RP is designed and constructed for research purposes, comprising custom-built in-house parts as well as off-the-shelf parts (e.g., aluminum profiles, bearing units, etc.). Its motion system features four mecanum wheels, providing the robot with omnidirectional motion capabilities [21]. The wheels are powered by four Maxon DC motors (RE 35) combined with planetary gearboxes (GP 42) and incremental encoders (HEDL 5540), providing 5 Nm of continuous torque per wheel. GT2 timing belts and pulleys are used to protect actuator shafts from increased robot payloads and to transmit power to the wheels. Two RoboClaw 2x30A motor controllers are used to drive the actuators since each controller can drive two DC brushed motors [22]. The encoders attached to the motors are read by the controllers, which run local PID control schemes that can follow precisely speed commands for all wheels. The two motor controllers are connected via USB to the system's master computer, which is a Raspberry Pi 2 model B (RPi) running the Raspbian OS. The operator can connect to the RPi using WiFi and Secure Shell (SSH) Network Protocol to run a Python program that establishes two serial connections with the motor controllers and sends the desired motion commands. The system is powered by two LiPo batteries for the RoboClaw controllers and a powerbank for the master computer. A Jetson Orin AGX Developer Kit is utilized for the execution of the VO software module and the recording of the experiments [23]. Two ZED2i cameras, one in the front and one in the back of the rover, are connected via USB 3.0 to the carrier board, and one ZEDX mini is on the side for the AprilTag detection part [26]. The third camera is connected via a GMSL2 port to the Jetson. This system is powered by a XP-1 Micro-Start lithium battery [32] and runs JetPack 5.1.1 [24].

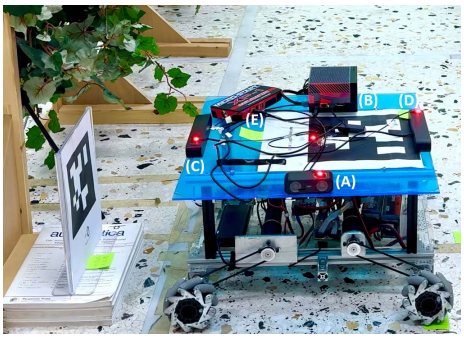


Fig. 2: The robotic platform with the developed vision system. (A) The ZEDX mini used for AprilTag detection, (B) the Jetson Orin AGX, (C) & (D) the two ZED2i cameras, (E) the XP-1 lithium battery that powers the Jetson Orin AGX.

C. Gazebo Environments

To test the performance of the vSLAM algorithm under controlled conditions, a simulation setup was implemented using Gazebo, which includes an accurate model of the RP as well as sensor plugins to simulate multiple stereo cameras and STL CAD descriptions of grapevines acquired from [25]. In addition, ROS was used since it allows replaying the experiments that were recorded in rosbags. As a result, we could assess the performance of our approach against state-of-the-art vSLAM algorithms. One AprilTag was added at the beginning of the first row as illustrated in Fig. 3. Two simulated environments were created; the first accurately resembles the synthetic vineyard at CSL, while the second resembles a typical vineyard with many rows of grapevines.

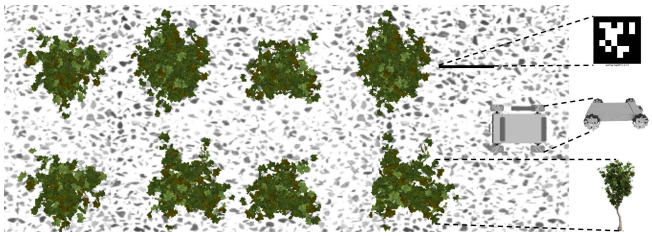


Fig. 3: Gazebo environment that resembles the CSL synthetic vineyard setup.

III. vSLAM Approach

This section presents our approach in detail. Firstly, the map is initialized by extracting features from both stereo

cameras (back and front ZED2i cameras) and by creating 3D landmarks using stereo matching. Features are extracted from each new camera frame and matched with the 3D landmarks. In turn, the camera pose is estimated by minimizing the reprojection error of the matches using the Ceres Solver [27]. Local BA is applied to achieve local consistency of the camera poses and refine the estimation of the 3D landmark positions. Lastly, AprilTag detection aids in identifying the loops, and if one is detected, global BA is performed.

A. Background

Feature extraction for vSLAM applications is a crucial step in providing accurate and robust localization and mapping. In this work, ORB Features were selected, due to (a) their scale and rotation invariance properties, and (b) the fact that their extraction requires low processing power and as a result they are suitable for real-time SLAM applications [28]. The ORB feature extraction algorithm was implemented from scratch to maintain full control over the process. Specifically, in our implementation, for a homogeneous distribution of features across the entire camera frame, each one was separated into grids, and ORB features were extracted on each grid separately. Additionally, to extract features even on less-textured image grids, adaptive fast threshold was also implemented on the feature extraction algorithm. Lastly, suppression via Square Covering (SSC) [29] ensured the selection of the strongest features, while maintaining homogeneity. Each extracted feature is assigned to a grid for accelerated matching.

During the initialization of the system, features are extracted from both stereo camera frames. The left camera frame features are matched with their stereo correspondence on the right frame, respecting the epipolar constraint. The depth value of a stereo match is considered valid, if its estimation is less than 40 times the stereo baseline [30], otherwise it is considered not accurate and is not used in the calculations. In turn, the map is initialized by creating 3D landmarks from the stereo matches. With every new camera frame, matches are searched between the projection of the 3D landmarks and the newly extracted features whose assigned grid is within a predefined radius around it. The system follows a constant velocity model, predicting the position of its 3D landmark on the new frame as if the camera maintained the same velocity (and orientation). If the number of the matches found is insufficient (the minimum number of matches was empirically chosen as 60) the radius is increased and the matching process restarts.

Another critical step for vSLAM applications is the keyframe selection. Keyframes are selected based on the current number of tracked features. If the number falls below a certain threshold or if there is a substantial reduction of tracked features from the previous keyframe, a new one is inserted. To maintain a clear map, only 3D landmarks that are matched with at least 3 keyframes are included.

For computational efficiency, the front-left camera pose is estimated without refining the world coordinates of the 3D landmarks. This is achieved by solving a non-linear optimization problem that minimizes the reprojection error, i.e.: Motion-Only BA. 3D landmarks that were observed from the front-right or back (left/right) lenses, require a transformation to the front-left lens to be factored in the calculations related to the estimation of the camera pose. The equations used for Motion-Only BA are :

$$\boldsymbol{R}, \boldsymbol{t} : min \sum_{i=1}^{M} \rho w_i \boldsymbol{d}(x_i, U(F_i))^2 \tag{1}$$

$$\boldsymbol{F_i} = \boldsymbol{K_C}(\boldsymbol{R_{CV}}(\boldsymbol{R}X_i + \boldsymbol{t}) + \boldsymbol{t_{CV}}) \tag{2}$$

$$\boldsymbol{U(F_i)} = \begin{bmatrix} \boldsymbol{F_i(0)}/\boldsymbol{F_i(2)} \\ \boldsymbol{F_i(1)}/\boldsymbol{F_i(2)} \end{bmatrix} \tag{3}$$

where C denotes the front or rear camera observing the 3D landmark, and V denotes the left or right lens. $\boldsymbol{R}, \boldsymbol{t}$ denotes the rotation matrix and translation displacement of the front-left lens, $\boldsymbol{R_{CV}}, \boldsymbol{t_{CV}}$ is the rotation matrix and translation displacement representing the transformation from the camera observing the 3D landmark, to the front-left lens and $\boldsymbol{K_C}$ is the intrinsics matrix for either the front or the rear camera. The ρ denotes the robust Huber [31] loss function, and w_i is a weight based on the scale of each observation. M represents the total number of the 3D landmarks to be optimized, X_i is the world coordinates of the current 3D landmark, x_i is the matched feature on the image plane, and $\boldsymbol{d(x, y)}$ is the Euclidean distance between vectors x and y.

Local BA is performed each time a new keyframe is added. This process refines the poses of the keyframes that share 3D landmarks with the newly inserted keyframe, and the locations of the 3D landmarks. The Local BA equations are presented in (4) and (5), where k denotes the keyframe and K denotes the total number of keyframes. Together with (3) form the Local BA optimization problem.

$$\boldsymbol{R_k}, \boldsymbol{t_k} : min \sum_{i=1}^{M} \sum_{k=1}^{K} \rho w_i \boldsymbol{d}(x_i, \boldsymbol{U(F_{i,k})})^2 \tag{4}$$

$$\boldsymbol{F_{i,k}} = \boldsymbol{K_C}(\boldsymbol{R_{CV}}(\boldsymbol{R}_k X_i + \boldsymbol{t_k}) + \boldsymbol{t_{CV}}) \tag{5}$$

Global BA is a particular case of Local BA where all the keyframes and 3D locations are refined. This process is performed when a loop closure is detected; in our case when a registered AprilTag is detected.

B. AprilTag detection

AprilTags (Fig. 3) are detected using the apriltag_ros package and the ZEDX mini on the side of the RP (Fig. 2) [34]. The apriltag_ros package outputs the transformation from the camera frame to the detected AprilTag frame. The world pose of the AprilTag is considered known and registered in the Jetson Orin AGX. When it is detected during an experiment, the current camera pose can be accurately calculated using a transformation between the AprilTag pose and the ZEDX mini frame. With an accurate estimate of the current camera pose, the loop closure thread initiates global BA to refine all the previously estimated poses and 3D landmarks.

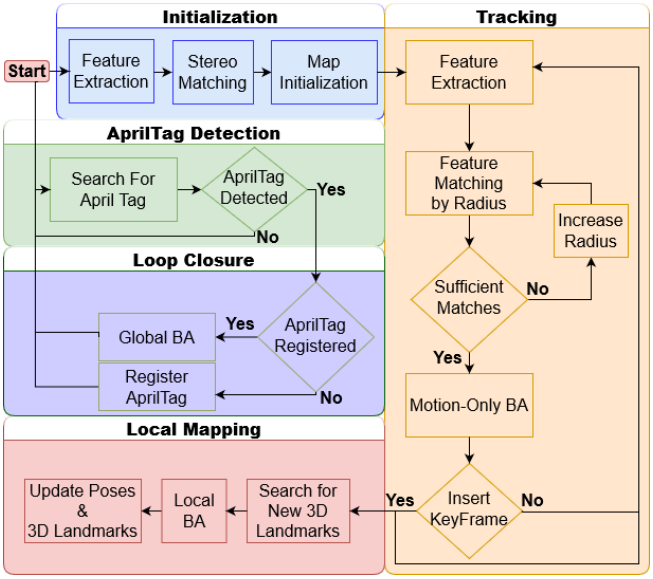


Fig. 4: The vSLAM Pipeline.

C. VSLAM Pipeline

The system is separated into four different processes: tracking, local mapping, loop closing, and AprilTag detection. Initialization is performed once during the system startup. Next, the tracking thread tracks the 3D landmarks in consecutive frames, performs Motion-Only BA, and inserts keyframes if needed. When a keyframe is inserted, the local mapping thread searches for new 3D landmarks between the newly added keyframe and its connected keyframes, and launches the process of the local BA. The loop closing thread performs a global BA when an AprilTag is detected for the second time. Lastly, the AprilTag detection thread detects AprilTags for loop closing. The flowchart of the developed vSLAM pipeline is presented in Fig. 4.

IV. Results

To evaluate the performance of the proposed vSLAM algorithm, experiments were conducted in the simulated vineyard in Gazebo and in the synthetic one at CSL. The results were compared against ORB-SLAM3 and the ground truth trajectory followed by the RP. The motion of the RP was captured using the PhaseSpace camera system [33].

A. Simulation

The RP used in the Gazebo simulation is a model of the RP described in Sect. II-B. Like the physical RP, it is equipped with two stereo cameras, one in the front and one in the back, that are used for camera pose estimation in the vSLAM algorithm, operating at 752x480 resolution, and one mini camera located at the right side of the rover used for AprilTag detection, operating at 1920x1080 resolution. The HD resolution for the mini camera was selected to increase accuracy in the AprilTag detection process. The two stereo cameras have $12cm$ baseline (identical to the ZED2i stereo camera) and operate at $15fps$. Two environments were created; one simulates the realistic environment built at CSL, and the second simulates a typical vineyard with many rows of grapevines.

Two experiments were performed on the first simulation environment. The first consisted of the rover inspecting the vineyard, where, on purpose, we have added a plain brown carton box close to the path that the RP should follow (see Fig. 5). The box has very few features available to track since its surface is homogeneous and monochromatic. The goal was to evaluate the algorithms when a featureless object covers a large part of the FoV of the front camera. ORB-SLAM3 failed to estimate the trajectory of the camera in this environment, and ultimately, the tracking process did reset (Fig. 5). In contrast, the developed algorithm outperformed ORB-SLAM3 & displayed robustness. It continued the camera pose estimation thanks to the features available through the rear camera. Eventually, it performed loop closure optimization when the AprilTag was detected at the end of the inspection process. Loop closure detection further improved the estimation of all previous poses since, at this point, the algorithm had the maximum confidence that the same point in the vineyard was visited in the past (see Fig. 5).

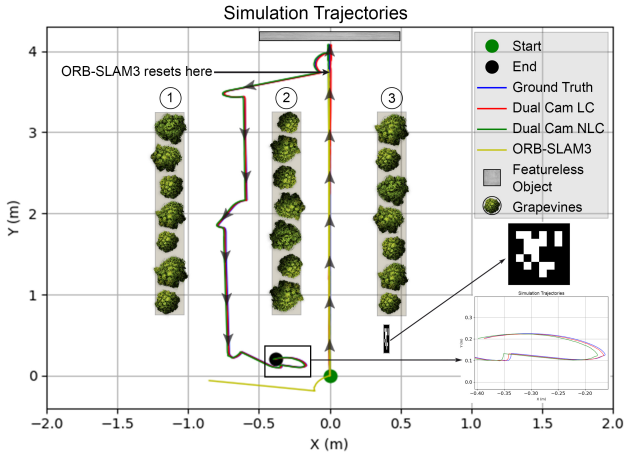


Fig. 5: The first simulation experiment. Dual Cam LC denotes our approach with loop closure, while Dual Cam NLC denotes our approach without loop closure. ORB-SLAM3 resets since there are very few features available to track. Our approach continues thanks to the features available from the rear camera.

To test the error accumulation over time and its effect on the overall accuracy of the proposed algorithm, during the second experiment, the RP was instructed to follow a longer path, map the entire vineyard and arrive at its starting point. Our framework achieved cm-level accuracy in this scenario by detecting the AptilTag at the end of the path and performing loop closure optimization, as presented in Fig. 6.

The second simulation environment was created to test the loop detection accuracy of the bag-of-words approach, on images with high similarity. The robot had to follow a long path in the vineyard as presented in Fig. 7 & return back to its home position. Intentionally, we avoided loops except

while returning to the home position. In this experiment ORB-SLAM3 detected 3 loop closures, two of them being incorrect, due to the similarities in the environment, resulting in highly inaccurate pose tracking. In contrast, the proposed algorithm tracked the path with an impressive cm-level accuracy and detected correctly the loop closure at the end of the path, where the AprilTag is located, ultimately resulting in excellent path tracking. The results are presented in Fig. 8.

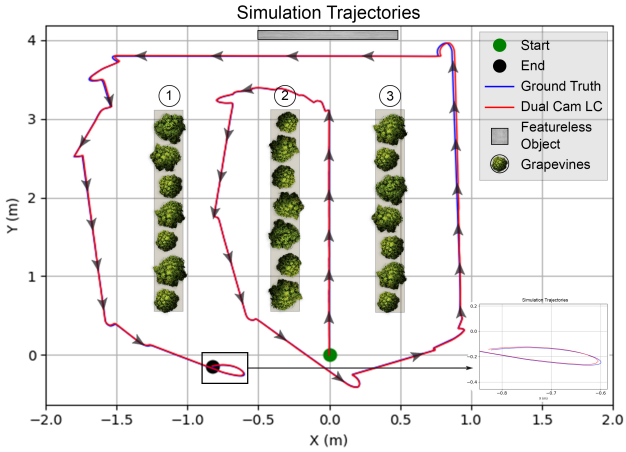


Fig. 6: The second simulation experiment. Our approach achieves cm-level accuracy.

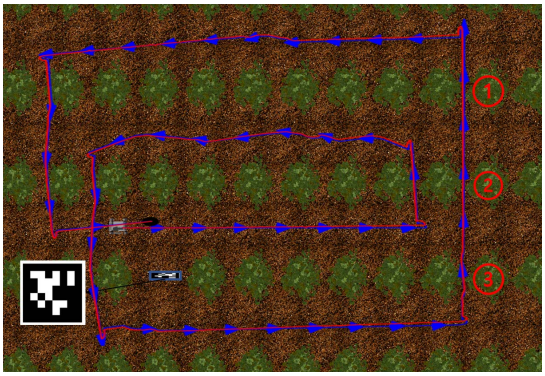


Fig. 7: Followed Path through the simulated Vineyard.

The results of the translation rms error (RMSE) in meters in the simulated environments are presented in Table I. It is evident that the developed algorithm achieves cm-level accuracy even in homogeneous environments where features may be harder to track.

B. Synthetic Vineyard Experiments

Additional experiments were conducted on the developed synthetic vineyard at CSL. The RP presented in Section II-B was utilized for this set of experiments. The two ZED2i cameras (front and rear) were operating at 640x360@15fps resolution while the ZEDX mini was operating at 1920x1080 resolution. To acquire the ground truth position of the rover, the PhaseSpace motion capturing system was used (red LEDs

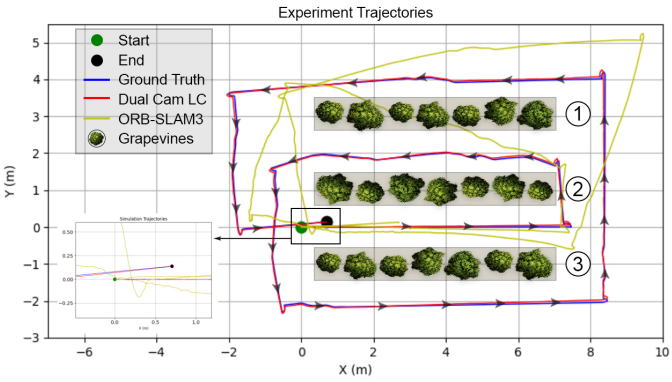
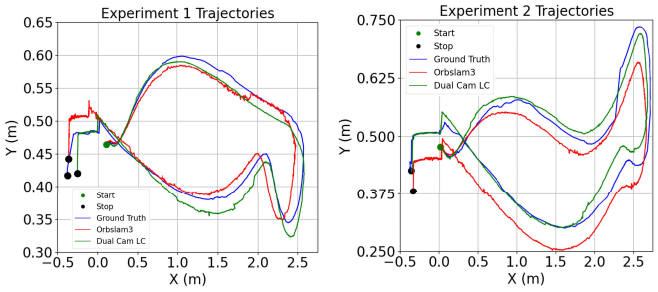


Fig. 8: ORB-SLAM3 incorrectly detects two loop closures, resulting in inaccurate pose tracking, while our approach tracks the path accurately and performs loop closure optimization only when the registered AprilTag is detected at the end of the trajectory.

in Fig. 2). In Fig. 9, the trajectory estimation of our approach and ORB-SLAM3's are presented compared with the ground truth. Our approach outperforms ORB-SLAM3 as evident by the Fig. 9 and by the RMSE results presented in Table I. Lastly, in Fig. 10, the case where a featureless carton box was placed in the middle of the row is presented. As expected, ORB-SLAM3 resets as there were very few features available to track while our approach demonstrated robustness.

TABLE I: Results of Translation RMSE (m) of all experiments in Gazebo (yellow) and CSL (gray).

	RMSE (m)	
Experiment	**Dual Cam LC**	**ORB-SLAM3**
Gazebo 1	0.006930	resets
Gazebo 2	0.015403	resets
Gazebo 3 (typical vineyard)	0.053063	detects wrong LC
CSL 1	0.160043	0.273532
CSL 2	0.123967	0.189264



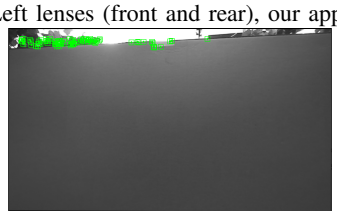
(a) (b)

Fig. 9: Experiment at the synthetic vineyard at CSL. Our approach outperforms ORB-SLAM3 since there are more features to track using the rear camera. Furthermore, Apriltag loop closure significantly enhances trajectory accuracy.

V. Conclusions

In this research work, we propose a novel approach for enhancing the robustness of vSLAM by utilizing multiple

(a) Left lenses (front and rear), our approach



(b) Left lens, ORB-SLAM3

Fig. 10: Feature tracking, (a) When the front camera is covered by a featureless object the algorithm continues tracking the available features from the second camera, (b) ORB-SLAM3 resets when a featureless object covers most of the camera's FoV.

stereo cameras and a new, novel method for detecting loop closures in environments where the bag-of-word loop closure detection methods are highly prone to fail. We display high accuracy in the localization of an RP in a vineyard and robustness even when a featureless object covers a large part of the FoV of one camera. The AptilTag approach for loop detection allows us to accurately detect loop closures when a state-of-the-art vSLAM algorithm fails. Future work includes finding the optimal number of AprilTags to be distributed in the vineyard so that cm-level accuracy can be achieved in even longer paths without the need to return the RP to its home position to perform loop closure.

ACKNOWLEDGMENT

The authors would like to thank TWI-Hellas for providing the PhaseSpace motion capture system and John Zarras for his valuable help in carrying out the experiments.

REFERENCES

[1] https://csl-ep.mech.ntua.gr/
[2] https://bitbucket.org/csl_legged/dc-vslam-med24/
[3] BACCHUS Project EU, 27-Feb-2020. [Online]. Available: https://bacchus-project.eu/ [Accessed: 1-Mar-2023].
[4] Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. 2007, 29, 1052–1067
[5] Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
[6] Mur-Artal, R., Montiel, J. M. M., & Tardós, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. CoRR, abs/1502.00956. http://arxiv.org/abs/1502.00956
[7] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.
[8] Rosinol, A., Abate, M., Chang, Y., & Carlone, L. (2020). Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. IEEE Intl. Conf. on Robotics and Automation (ICRA).
[9] Negre, P. L., Bonin-Font, F., & Oliver, G. (2016). Cluster-based loop closing detection for underwater slam in feature-poor regions. 2016 IEEE International Conference on Robotics and Automation (ICRA), 2589–2595. https://doi.org/10.1109/ICRA.2016.7487416
[10] Bescos, B., Facil, J. M., Civera, J., & Neira, J. (2018). DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. IEEE Robotics and Automation Letters, 3(4), 4076–4083. https://doi.org/10.1109/lra.2018.2860039
[11] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," CoRR, vol. abs/1610.06475, 2016, [Online]. Available: http://arxiv.org/abs/1610.06475
[12] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel and Juan D. Tardós, ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM, IEEE Transactions on Robotics 37(6):1874-1890, Dec. 2021
[13] Qin, Tong Cao, Shaozu Pan, Jie & Shen, Shaojie. (2019). A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors.
[14] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, 2006, pp. 363–370.
[15] Gálvez-López, D., & Tardós, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics, 28(5), 1188–1197. https://doi.org/10.1109/TRO.2012.2197158
[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 3400-3407.
[17] Urban, S., & Hinz, S. (2016). MultiCol-SLAM - A Modular Real-Time Multi-Camera SLAM System. ArXiv Preprint ArXiv:1610.07336.
[18] Urban, S., Wursthorn, S., Leitloff, J., & Hinz, S. (2016). MultiCol Bundle Adjustment: A Generic Method for Pose Estimation, Simultaneous Self-Calibration and Reconstruction for Arbitrary Multi-Camera Systems. International Journal of Computer Vision, 1–19.
[19] Urban, S., Leitloff, J., & Hinz, S. (2015). Improved Wide-Angle, Fisheye and Omnidirectional Camera Calibration. ISPRS Journal of Photogrammetry and Remote Sensing, 108, 72–79.
[20] J. Kuo, M. Muglikar, Z. Zhang and D. Scaramuzza, "Redesigning SLAM for Arbitrary Multi-Camera Systems," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 2116-2122, doi: 10.1109/ICRA40945.2020.9197553.
[21] "4 in. HD Mecanum Wheels" https://www.andymark.com/products/4-in-hd-mecanum-wheel-set-options. Accessed: 2023-01-01.
[22] "RoboClaw Motor Controller." https://www.pololu.com/product/3286.
[23] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/
[24] https://developer.nvidia.com/embedded/jetpack-sdk-511
[25] "Vineyard Models." https://www.turbosquid.com/3d-model/vineyard. Accessed: 2023-01-01.
[26] https://store.stereolabs.com/en-eu/products/zed-2i.
[27] Agarwal, S., Mierle, K., & Team, T. C. S. (2022). Ceres Solver (2.1) [Computer software]. https://github.com/ceres-solver/ceres-solver
[28] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571.
[29] Bailo, O., Rameau, F., Joo, K., Park, J., Bogdan, O., & Kweon, I. (2018). Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. Pattern Recognition Letters, 106. https://doi.org/10.1016/j.patrec.2018.02.020
[30] L. M. Paz, P. PiniÉs, J. D. TardÓs and J. Neira, "Large-Scale 6-DOF SLAM With Stereo-in-Hand," in IEEE Transactions on Robotics, vol. 24, no. 5, pp. 946-957, Oct. 2008, doi: 10.1109/TRO.2008.2004637.
[31] Peter J. Huber. "Robust Estimation of a Location Parameter." The Annals of Mathematical Statistics, 35(1)
[32] https://antigravitybatteries.com/products/micro-starts/xp-1/
[33] https://www.phasespace.com/
[34] Malyuta, D., Brommer, C., Hentzen, D., Stastny, T., Siegwart, R., & Brockers, R. (2019). Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. Journal of Field Robotics, arXiv:1908.06381.