Loop Closure Detection and SLAM in Vineyards with Deep Semantic Cues

Alexios Papadimitriou[1,2], Ioannis Kleitsiotis[1,2], Ioannis Kostavelis[1],
Ioannis Mariolis[1], Dimitrios Giakoumis[1], Spiridon Likothanassis[1,2] and Dimitrios Tzovaras[1]

***Abstract*— Automation of vineyards cultivation necessitates for mobile robots to retain accurate localization system. The paper introduces a stereo vision-based Graph-Simultaneous Localization and Mapping (Graph-SLAM) pipeline custom-tailored to the specificities of vineyard fields. Graph-SLAM is reinforced with a Loop Closure Detection (LCD) based on semantic segmentation of the vine trees. The Mask R-CNN network is applied to segment the trunk regions of images, on which unique visual features are extracted. These features are used to populate the bag of visual words (BoVWs) retained on the formulated graph. A nearest neighbor search is applied to each query trunk-image to associate each unique feature descriptor with the corresponding node in the graph using a voting procedure. We apply a probabilistic method to select the most suitable loop closing pair and, upon an LCD appearance, the 3D points of the trunks are employed to estimate the loop closure constraint to the graph. The traceable features on trunk segments drastically reduce the number of retained BoVWs, which in turn expedites significantly the loop closure and graph optimization, rendering our method suitable for large scale mapping in vineyards. The pipeline has been evaluated on several data sequences gathered from real vineyards, in different seasons, when the appearance of vine trees vary significantly, and exhibited robust mapping in long distances.**

I. INTRODUCTION

Vineyards are used for the production of grapes, the growing of which constitutes a significant fragment in orchards farming that requires dedicated agricultural operations [1]. Vineyard agriculture comprises procedures such as cultivation, inspection, spraying, pruning and harvesting which are currently performed manually. The deployment of autonomous robots in vineyards, capable of undertaking such procedures [2], can significantly minimize the harsh and long working conditions experienced by grapes growers, while at the same time can offer precise and extend control on monitoring, planning and prediction of farmers yield [3].

However, the deployment of existing robotic technologies in vineyards is not straightforward, since there are major challenges [4]. The various illumination conditions that influence the vision systems' performance, the poor GPS (Global Positioning System) availability due to the signal blockage or multi-reflections in vineyards located in mountains or in canopies created in blossom seasons, the dead-reckoning systems failure due to harsh terrains, the patterns' repeatability and the seasonal variation of trees' morphology that influence the robot's localization, are some of these challenges that agricultural robots need to cope with. Robot mobility is essential for vineyards exploration thus, accurate SLAM that deals with such challenges is essential for each robot to be deployed in such environments [5].

Fig. 1. On top, an aerial instance of a vineyard. On bottom, three instances of the same vineyard in different seasons exhibit the variation of the scenery.

The visual SLAM in vineyards has been already studied from different perspectives. Some approaches focus on the use of cameras in GPS-denied environments [6], [7], [8] to incrementally create a map relied on vineyards structured features. High resolution 3D Lidar sensors are also used to provide dense vineyards reconstruction [9], yet the high-cost of such sensors still hinders the adoption of such solutions. Semantic SLAM methods have also been presented, tailored to vineyards mapping needs, but tested on limited ranges and data [10], mainly due to excessive demands on computational resources. LCD approaches are typically used in long travelled distances to correct the significant drift caused by incremental SLAM methods [11], some of which have been applied to agriculture applications [12], [13]. Though, the uniformity of vineyards makes the detection of places, which the robot has passed, so challenging that requires dedicated approaches to distinguish among grape trees and structures. To the best of our knowledge, a semantic-based LCD method for SLAM optimization in vineyard environments has not been presented yet. The majority of the existing methods rely on the existence of artificial landmarks for LCD or employ the GPS measurements, yet placing landmarks in

[1]Centre for Research and Technology-Hellas, Information Technologies Institute (CERTH / ITI)

[2]Large Scale Machine Learning and Cloud Data Engineering Lab, Computer Engineering and Informatics, University of Patras, Greece

ACKNOWLEDGMENT: This work has been supported by the EU Horizon 2020funded project namely: "BACCHUS (MoBile Robotic PlAtforms for ACtive InspeCtion and Harvesting in AgricUlturalAreaS)" under the grant agreement with no: 871704.

all vineyards areas is not possible neither practical, while the presence of accurate GPS measurements is sometimes questionable, since it depends on the vineyard's location.

We introduce a Graph-SLAM method accompanied by an LCD strategy that capitalizes on deep learning-based semantic segmentation of grape trees to isolate their trunk regions upon which traceable and unique visual words (VWs) are computed. We tackle the uniformity of vineyard environments (see Fig. 1) by focusing on the trunk regions of grape trees, since they slightly vary among seasonal alterations. Candidate frames for LCD are inferred in a vote-wise probabilistic manner, exploiting the (trunk-based) semantic VWs, which in turn used for graph optimization when the robot revisits previously seen places in the vineyard. Our method relies only on stereo visual input, maintaining the required perception resources as minimal as possible thus, the main contributions are as follows:

- a semantic segmentation of grape trees based on Mask R-CNN neural network and the formalization of visual words with local descriptors on the salient regions (i.e., the trunks)
- a feature tracking visual odometry (VO) strategy relied on a double-filtering approach to remove outliers among consecutive frames
- an LCD method that build upon semantic visual words to calculate appearance based probability among stored nodes in the graph
- data accumulation and fusion using Graph-SLAM in order to map, correct and smooth the robot's long trajectories

II. RELATED WORK

We focus on recent works related to mapping techniques in orchard related environments, emphasizing on works conducted for vineyards applications. In addition, we discuss the existing works oriented towards the semantics of vineyards either for semantic mapping and robot localization.

A. Mapping and Localization in Vineyards

In many agricultural related robotics applications, the ORB-SLAM and ORB-SLAM2 [14], [15] approaches are widely used since they are suitable for outdoor applications, and prune redundant key-frames maintaining a bounded-size map. Authors in [16] utilized ORB-SLAM for vineyards, yet slightly modified to recover key-frames for further classification approaches. Authors in [17] extensively tested monocular ORB-SLAM2 in Rosario [18] agriculture dataset and found that the heuristic threshold used for initializing monocular tracking and the number of ORB features extracted impact the robustness of the system significantly. VineSLAM [8], constitutes a 2D SLAM solution tailored to vineyard environments. The authors employed geometrical attributes of the grape trees and their relevant distances to create dedicated natural features, namely ViTruDe [19]. ViTruDe features combined with RFID tags placed on the edges of each row of the vineyard to ease LCD in row transitions can alleviate extended Kalman filter from increased complexity. In [20], an LCD in a commercial orchard is performed with artificial landmarks made of retro-reflective tapes located at the end of each row, facilitating the detection of the pre-visited location with a 3D Lidar. However, such methods require the existence of a large number of artificial landmarks to cover an entire vineyard and in many cases could be impractical. Authors in [21], found GMapping to be the most reliable among KartoSLAM [22] and Google-Cartographer [23] although, these algorithms are considered to be efficient only on planar fields. In more recent work described in [9], a SLAM method based on a 3D Lidar data fused with GPS measurements, created 3D maps for the canopy density estimation. The method exhibited accurate results for long travelled distance. However, in such cases GPS signal reception could be sparse and the utilization of 3D Lidar sensors would significantly increase the cost of the solution. Another work that follows a similar philosophy to ours is the one described in [10]. The authors used semantic features, which represented grape spheres, extracted from 2D images and by exploiting depth information the corresponding 3D point clouds were utilized for the camera motion estimation in consecutive frames with singular value decomposition, followed by an ICP variation. This method provides accurate SLAM for few meters travelled distance, although it does not account for long distances in vineyards and does not cope with the LCD and localization error drift. Moreover, relying only on grapes detection ends up to ephemeral mapping and localization abilities, since grapes are present on the trees only for one season annually and the rest period the map becomes obsolete.

B. Deep Vine Part Detection

Over the last decade, the computer vision community has increasingly moved away from the paradigm of handcrafted features on images, towards richer feature representations extracted from deep Convolutional Neural Networks (CNN). This trend has greatly affected the state-of-the-art methods in the tasks of semantic segmentation, bounding box regression and instance segmentation in viticulture. The authors of [24] detected grape bunches in their manually annotated dataset, evaluating YOLOv2 [25], YOLOv3 [26], in the tasks of grape bunch bounding box detection and Mask R-CNN[27] in the task of instance segmentation. The authors of [28] utilized a location sensitive variant of the HTC [29] instance segmentation network to detect grape bunches on RGB images and estimated the number of individual berries inside them. Although these works achieved impressive results in the extraction of grape bunch masks, they did not concern themselves with parts of the plant that remain unchanged throughout the year. In [30] bounding box detection was performed on an RGB vine trunk dataset with Faster R-CNN [31], YOLOv3 and YOLOv5 [32]. However, they did not predict trunk masks in their dataset, which consist more precise representations of trunks' location. The authors of [28] trained SegNet [33] and FCN [34] to semantically segment trunks and cordons for the task of

cordon trajectory estimation. Due to the heavy occlusions of cordons from shoot/leaves, in [35] the same authors utilized the Faster-RCNN architecture to detect all the visible cordon parts and the FCN [34] architecture to segment shoots/leaves. Afterwards, they predicted the centroid of an occluded cordon segment between two visible cordon segments by leveraging the geometric information from the occluding shoots/leaves. However, a forward-looking camera in a vineyard corridor adds perspective distortion to the scenery, resulting in whole plant cordons being occluded, and different plant cordons occupying significantly different pixel area, thus, making the regression of cordon trajectory less robust. The works most similar to ours are those presented in [36] and [37], which both captured their datasets with a frontal stereo camera mounted on an AgRob V16, in conjunction with thermal and blue-infrared-filtered cameras respectively. In [36] the authors evaluated two SSD [38] bounding box detector variants on their manually annotated datasets. Similarly, the authors of [37] evaluate the SSD architecture with a MobileNetv1 and MobileNetv2 backbone, as well as the TinyYOLOv3 [26] architecture, for the bounding box detection of vine trunks, which are projected in corresponding registered depth images to facilitate the construction of a vineyard corridor 2D map. However, both works considered only the bounding boxes surrounding the trunks, which according to our experiments were inadequate for the extraction of salient features throughout the seasons.

III. METHODOLOGY

A. Structured Motion in Vineyards

For harvesting and inspection tasks executed in an orchard-like environments such as vineyards, the robot follows structured motions in order to complete its mission. Such motions are typical in outdoor applications where robots cover long distances to visit all places of interest [39]. We selected a variation of Boustrophedon-like motion in our mapping approach based on which, each row is traversed twice. In the first row passing the robot moves close to the trees on the left side of the row and in the second row passing, the robot moves close to trees on the right side of the row. This is a reasonable motion for harvesting applications where the robot moves closer to vine trees to execute the manipulation tasks. This motion is also convenient for localization, since passing from the same row twice will favour the loop closure detection.

B. Trunk Detection Network

Our approach for leveraging visual information for the accurate robot SLAM is to isolate trunk regions of RGB images and compute on these segments representative features that uniquely describe the observed place and will be traceable in a potential robot revisit of the same area. The semantic module of our pipeline extracts the binary masks defining only the regions of vine trunks and returns the "strongest" features that constitute the visual words (VWs). These unique features extracted from semantic regions of vine trees contribute to the formulation of a Bag of Visual Words (BoVWs) framework utilized for the loop closure searching strategy.

Mask extraction is actualized through Deep Convolutional Neural Networks trained on our hand-tailored dataset of vineyard scenery. The relevant tasks associated with our problem are either semantic segmentation or instance segmentation, with the latter providing more information, as it additionally provides information regarding the existence of individual objects on the image, instead of a per-pixel mask of the image. We investigated both approaches using state-of-the-art architectures representative of each category - PSPNet [40] for semantic segmentation and Mask R-CNN [27] for instance segmentation. Mask R-CNN estimates masks for each instance of trunks existing in an RGB image, therefore, we created the total trunks mask by applying the OR logical operator to all the predicted masks.

The experimental results indicated the superior performance of Mask R-CNN in our dataset. Therefore, we chose it for the task of estimating the binary masks of the trunks $\mathbf{V}$ for the query RGB image $\mathbf{I}_Q$. Following the estimation of $\mathbf{V}$, we extract trunk information from $\mathbf{I}_Q$ as follows:

$$\mathbf{I}_i^e = \begin{cases} \mathbf{I}_i^Q & \text{if } \mathbf{V}_i = 1 \\ 0 & \text{if } \mathbf{V}_i = 0 \end{cases}$$

where $\mathbf{I}^e$ is the extracted image, $\mathbf{I}_i^Q$ the value of $\mathbf{I}_Q$ at pixel i and $\mathbf{V}_i$ is the value of the trunk mask $\mathbf{V}$ at pixel i. Afterwards, we calculate the ORB features for $\mathbf{I}^e$, where we limit the maximum number of the requested features on the detected trunks to 1000. The set of the ORB features' descriptors is denoted as the $\mathbf{M_Q}$ and corresponds to the semantic VWs of the query image. We showcase the complete VWs computation pipeline with a sample from our dataset in Fig. 2.

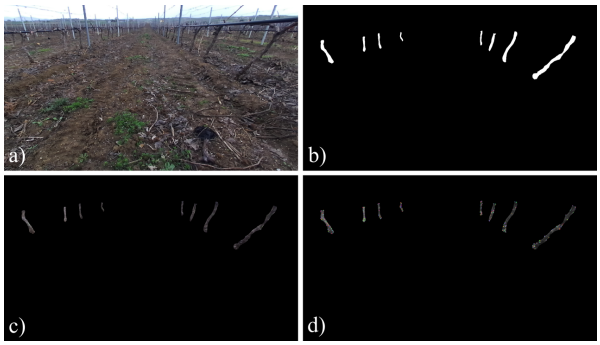


Fig. 2. The visual words computation pipeline: a) The original RGB image; b) The trunks binary mask; c) The extracted RGB trunks from the original image; d) The ORB features' positions on the grayscale extracted trunks.

C. Semantic Graph-SLAM

Our complete framework for the proposed stereo vision-based Graph-SLAM is graphically illustrated in Fig. 3. The front-end, is a VO scheme, where the estimated robot poses constitute the nodes of a pose graph. The VWs are also computed and associated with each node in the graph. The

back-end refers to the LCD and to the graph optimization approach.

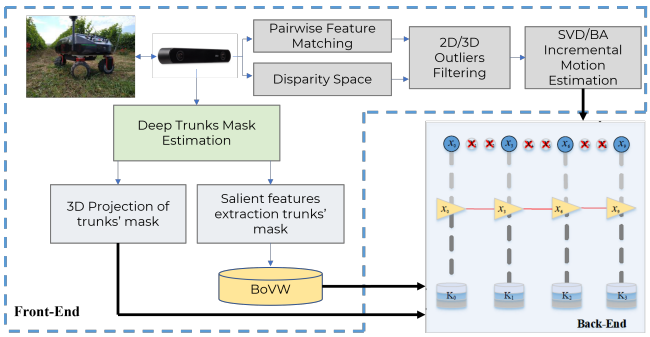


Fig. 3. The proposed Graph-SLAM pipeline; the front-end comprises to the robot incremental motion estimation and the creation of key-frames based on the semantic segmentation of trunks, while the back-end comprises the LCD and graph-optimization modules.

1) Visual odometry (VO) (front-end): We employ a two-step features tracking and filtering method in order to discard as many outliers as possible, aiming at obtaining a reliable rigid transformation among camera poses. The ORB features [41] are selected, which are computational inexpensive and account for viewpoint changes. In particular, let $\mathbf{Q_t}$ and $\mathbf{Q_{t-1}}$ be the sets of 2D correspondences of the frames at time t and $t-1$, respectively. The feature matching is performed in brute force manner by exploiting Hamming distance. On the 2D correspondences, we estimate the planar rotation matrix H using RANSAC, with the Direct Linear Transform (DLT) as the hypothesis of the RANSAC loop. The correspondences which do not satisfy the transformation $\mathbf{Q_{t-1}} = H\mathbf{Q_t}$ are discarded as outliers. The depth image is then calculated using the corresponding disparity maps. The matched inliers are then projected to 3D coordinates using the depth image formulating two sets of $\mathbf{P_t}$ and $\mathbf{P_{t-1}}$. Subsequently, for each 3D point $p_t \in \mathbf{P_t}$, the rate $\rho_z = \frac{p_t^z}{p_{t-1}^z}$ is calculated, where p_t^z denotes the z coordinate of the point p_t, and if there is an extreme deviation from 1 the point is rejected as outlier [42]. The transformation T which best aligns the remaining 3D correspondences is computed through singular value decomposition (SVD). We apply a local Sparse Bundle Adjustment (SBA) [43] on a sliding window of three consecutive frames to optimize the estimated transformation T by minimizing the re-projection error. The camera motion transformation T is expressed to the robot base frame and the final robot transformation τ_t among consecutive robot poses x_t, x_{t-1} is obtained.

2) Pose Graph formulation (back-end): In our graph each node corresponds to a robot pose x_t, which is added as a new node, when $d(x_t, x_{t-m}) > D$, where m, is the last pose added in the graph and D is an euclidean distance threshold large enough to maintain the sparse nature of the problem. The edges correspond to the accumulative transformation $T_g = \tau_{t-m}, \tau_{t-m+1}, ..., \tau_t$, between the nodes in the graph. Each node is also associated with a key-frame $\mathbf{K_Q}$ which contains the information of the VWs $\mathbf{M_Q}$, as described in Sect. III-B. Using the trunks image $\mathbf{I^e}$ and the depth information of the disparity maps, the point cloud is calculated and included in $\mathbf{K_Q}$. The concatenated descriptors $\mathbf{M}$ for each processed key-frame formulate the BoVWs used for searching for LCD when a new query image $\mathbf{I}_Q$ appears.

In Fig. 3 a formulated pose graph with 4 nodes is graphically illustrated, where each node is also associated with a key-frame $\mathbf{K}_n$ and contains the respective VWs.

Each key-frame $\mathbf{K}_n$ contains also the point clouds corresponding to segmented trunks that inherit the transformation T_g. This allows global map refinements when an LCD triggers a graph optimization.

3) Loop closure detection and optimization: When a new node is to be added in the graph, we trigger an LCD scheme as proposed in [44]. Let $\mathbf{I}_Q$ be the captured image of the query node and $\mathbf{M}_Q \subset \mathbf{K}_Q$ containing N_Q ORB features on the detected trunks of the query image $\mathbf{I}_Q$. A k-nearest neighbor (k-NN) search based on hamming distance is performed with $k = 1$, having as a train set all the pre-visited nodes' descriptors $\mathbf{M}_i$, $i \in (0, L-l)$ of N_i features. L denotes the number of already existing nodes in the pose graph and l is the set of most recently added nodes. A histogram of votes is created where each bin represents the k_i NNs of the node i. Efficient search for NNs is achieved through a Locality Sensitive Hashing (LSH) algorithm. To avoid definition of heuristics and thresholds for each node i we estimate the binomial probability mass function (PMF):

$$P_{r,i}(X = k_i) = \binom{N_i}{k_i} p_i^{k_i} (1 - p_i)^{N_i - k_i}, \tag{1}$$

where, $0 < p_i = \frac{N_i}{\sum_i N_i} < 1$.

By exploiting the "Law of rare events" [45] an LCD event is considered when the following criteria are met:

$$0 \leq P_{r,i}(X = k_i) < \lambda, \text{ where } \lambda \approx 0, \tag{2}$$

and

$$k_i > E[Bin(N_i, p_i)]. \tag{3}$$

To avoid false detection, we consider that a loop actually exists if there are at least three consecutive LCDs. For each candidate node c we estimate the transformation T_c which best aligns the point clouds that correspond to the projected features of VW, $\mathbf{P}_c$ and $\mathbf{P}_Q$ with the method described in Sect. III-C.1. Should the emerged transformation's T_{c_v} translation norm is smaller than D, this transformation is added as a constraint between the nodes Q and c_v. The graph update and optimization follows by exploiting the ISAM2[46].

IV. EXPERIMENTAL EVALUATION

A. Experimental Protocol

The evaluation dataset for both the vine trunk detection and LCD is consisted of stereo RGB images with 1280×720 pixels captured by a ZED 2 camera and collected during three visits to a northern Greece vineyard in the

autumn of 2020, the winter and the summer of 2021[1], with significant differences in the ambient illumination during the data collection. The collected data sequences followed the trajectory described in Sect. III-A for 2 adjacent rows. In the time interval between our first two visits, the vineyard had been defoliated and no withered leaves had remained on the ground, while in our last visit the trees had blossomed, contributing to an important qualitative change in the captured scenery. Among the collected sequences we selected 399 frames from the first visit, 105 frames from the second visit, and 114 frames from the last visit to manually annotate the four closest to the camera trunks from the left and right rows of visible trees. Samples of the captured data are exhibited in Fig. 4.

B. Evaluation

1) Trunk Detection performance: We trained our algorithms from samples drawn randomly from all visits, but evaluated them separately. We made a $70-10-20\%$ train-validation-test split of the data captured from each visit, and augmented our training set ten times per training image, following the augmentation schema proposed in [24]. We zero-padded the images fed into Mask R-CNN to a $1:1$ aspect ratio at the largest image dimension, i.e., 1280 pixels.

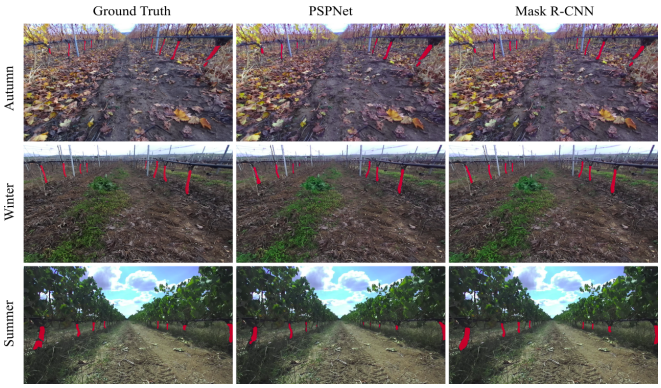


Fig. 4. Qualitative comparison of PSPNet and Mask R-CNN detection of trunks, with respect to the ground truth annotated data, for an autumn, winter and summer RGB sample. In all cases Mask R-CNN produces more accurate trunk masks than PSPNet, especially for the third and fourth trunk in a row. The reader should note the qualitative differences between the scenes corresponding to different collection sessions.

Regarding the training of the trunk mask prediction networks, we generally kept most of the default parameters of the Mask R-CNN implementation [47] and PSPNet implementation [48] [49]. We utilized the validation dataset for the selection of the optimal learning rate, number of training iterations and network backbone. For Mask R-CNN the metric optimized was mAP@0.5:0.95 IoU, while for PSPNet it was mean trunk class IoU (Trunk IoU). Additionally for Mask R-CNN we found the optimal confidence threshold value by maximizing the F-measure at 0.3 IoU. We trained our algorithms on a Nvidia Tesla K40m up to the largest batch size and backbone network possible with respect to our hardware. We evaluated the two methods with Trunk IoU and the mean IoU of the background class (BG IoU) in our three datasets, and present our experimental results on Table I, highlighting with bold letters the best evaluation metric for each method. Qualitative results comparing the two methods with the ground truth can be seen in Fig. 4.Mask R-CNN constituted the best method for our datasets and, therefore, we employed it for the LCD method.

[1]We thank Ktima Gerovassiliou winery for providing us with access to its vineyard for our data collection and experiments

TABLE I
EVALUATION OF MASK R-CNN AND PSPNET FOR THE THREE VISITS.

Method / Dataset	Trunk IoU	BG IoU
Mask R-CNN / autumn	**0.683**	**0.998**
PSPNet / autumn	0.641	0.996
Mask R-CNN / winter	**0.723**	**0.998**
PSPNet / winter	0.659	0.996
Mask R-CNN / summer	**0.625**	**0.999**
PSPNet / summer	0.474	0.997

2) Mapping and Optimization performance: In order to achieve real-time performance, the front and back-end of proposed SLAM assigned to two separate threads. As for the parameters described in Sect. III-C.1 and Sect. III-C.2, we found $D = 0.5$m and $l = 160$ to be sufficient for the respective dataset. For the evaluation of the proposed SLAM algorithm, we extracted 5 pairs of rows from the total available scenes, 1 from autumn (A01), 2 from the winter (W01, W02) and 2 from the summer (S01, S02). The S01 and S02 also contain ground truth (GT) data provided by a Real Time kinematic positioning (RTK) system and an absolute heading Inertial Measurement Unit (IMU). A summary of the datasets details are presented in Table II.

TABLE II
SLAM DATASET SUMMARY.

Dataset Name	Row Length(m)	Total Distance(m)	GT	FPS
A01	80	175	No	2
W01	80	327	No	2
W02	80	317	No	2
S01	35	102	Yes	5
S02	45	122	Yes	5

First, we evaluate our method regarding the loop closure performance. To evaluate how the VWs stemming from our trunk semantic segmentation approach contribute to the efficient detection of loop closures, we compare it against the case where 20% of the strongest ORB features obtained from the entire image are added as VWs in each key-frame. The results are summarized in Table III for our method and in Table IV for the base line method, respectively.

In our method, from the total amount of nodes added to the graph while traversing the row at second time, $\approx 50\%$ are identified as LCD events. However, for the baseline approach less than 1/3 of nodes are correctly identified as LCD events. The latter means that the extracted VWs based on semantic trunk segmentation significantly contribute to the LCD since those VWs are traceable when the robot passes from the same

TABLE III

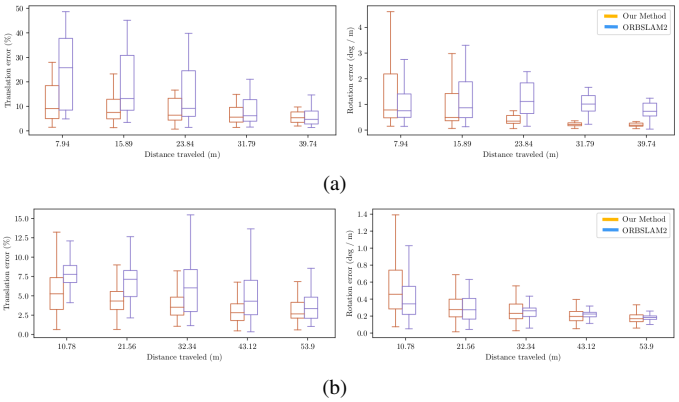
EVALUATION OF LCD - SEMANTIC TRUNKS MASKS.

Dataset	#Nodes	Common Nodes	True LCD	False LCD
A01	305	11	5	1
W01	436	145	71	11
W02	427	137	74	4
S01	197	62	42	2
S02	240	137	91	4

TABLE IV

EVALUATION OF LCD - 20% BASELINE METHOD.

Dataset	#Nodes	Common Nodes	True LCD	False LCD
A01	305	11	2	5
W01	436	145	45	18
W02	427	137	39	11
S01	197	62	27	14
S02	240	137	22	17



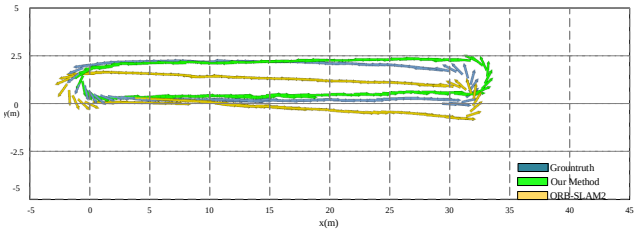
(a)

(b)

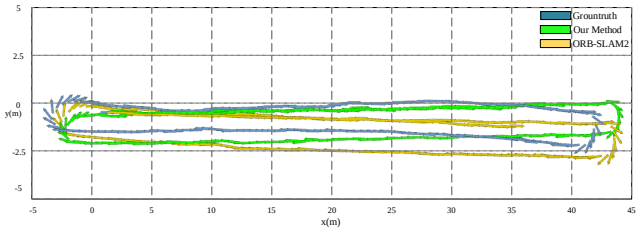
Fig. 6. Translation and rotation RMSE comparison of our method (orange) against ORB-SLAM2 (blue) in datasets a) S01 and b) S02.

place and can coped with the uniformity of the observed scenes within the same row.

Since the GT data were only available in the summer datasets, we evaluate the trajectory against the GT and the ORB-SLAM2 [15] approach. The error metric used to evaluate the output trajectory are the Absolute Trajectory Error (ATE), which is defined as the average deviation from the GT, where for S01 and S02 our method achieved 0.667 and 0.76 for each sequence respectively, against the 0.976 and 1.082 of the ORB-SLAM2. The trajectory comparison is shown in Fig. 5 and the root mean square error (RMSE) of the translation and rotation vector w.r.t. the travelled distance of the robot is exhibited in Fig. 6.



(a)



(b)

Fig. 5. Evaluation of our proposed implementation (green) with the ORB-SLAM2 (yellow) and the GT (blue) in the summer datasets a) S01 and b) S02.

The results shows that our method performs better in the challenging environment of the vineyard based both on the qualitative comparison and the resulted errors. The efficacy of the LCD and the graph optimization, is indicated in Fig. 6 by the significant drop on the translation and rotation error after the first row passing.

In order to further evaluate the global consistency of the generated map, we utilized a statistical methodology. First, we extracted the 3D points corresponding to the mask of each trunk i denoted as $\mathbf{T}_i$, translated by the respective optimized robot's pose, and calculated the trunk's points mean μ_i. Then, we created clusters, where each cluster q contains all the $\mathbf{T}_i$ in a radius r around the trunk i. The radius was selected to be 0.7m, since two consecutive trunks are mostly 0.8m apart. For each cluster we calculated its center and estimated the standard deviation σ_i, of the trunks belonging to q. The mean $\frac{1}{n}\sum_i \sigma_i$, where n is the total number of detected trunks found to be $0.29m$ for A01, $0.19m$ for W01, $0.13m$ for W02, $0.09m$ for S01 and $0.12m$ for S02. This metric indicates that the the existence of large amount of LCDs and the triggering of the graph-optimization contribute to the formulation of consistent 3D metric maps, avoiding duplicate registration of trunks.

V. CONCLUSION

A Graph-SLAM method for robot operation in vineyard environments has been presented. For the front-end, a custom stereo VO tailored to the specificities of such environments has been designed. Nodes and edges are added to the pose graph based on robot motion strategy while the salient information in each key-frame is obtained through unique features from the segmented trunk regions on the image. Trunk segmentation is applied with a Mask R-CNN network and on the extracted masks ORB features are calculated. These features, formulate a representative BoVWs that can discriminate among the uniform scenes within the rows of the vineyard, highly contributing to the efficient loop closures, the detection of which is relied on the "law of rare events" calculated as the lowest probability in the binomial distribution. The 3D projected trunk regions are further utilized for the optimized graph-update with an accurate loop closure constraint. Experimental results on real vineyards proved the ability of the method to produce accurate robot motion estimation and consistent metric maps.

REFERENCES

[1] S. V. Wandkar, Y. C. Bhatt, H. Jain, S. M. Nalawade, and S. G. Pawar, "Real-time variable rate spraying in orchards and vineyards: a review," *Journal of The Institution of Engineers (India): Series A*, vol. 99, no. 2, pp. 385–390, 2018.

[2] D. Sarri, L. Martelloni, and M. Vieri, "Development of a prototype of telemetry system for monitoring the spraying operation in vineyards," *Computers and Electronics in Agriculture*, vol. 142, pp. 248–259, 2017.

[3] J. Lowenberg-DeBoer, I. Y. Huang, V. Grigoriadis, and S. Blackmore, "Economics of robots and automation in field crop production," *Precision Agriculture*, vol. 21, no. 2, pp. 278–299, 2020.

[4] J. Billingsley, A. Visala, and M. Dunn, "Robotics in agriculture and forestry," 2008.

[5] M. S. A. Mahmud, M. S. Z. Abidin, A. A. Emmanuel, and H. S. Hasan, "Robotics and automation in agriculture: present and future applications," *Applications of Modelling and Simulation*, vol. 4, pp. 130–140, 2020.

[6] F. N. Dos Santos, H. M. P. Sobreira, D. F. B. Campos, R. Morais, A. P. G. M. Moreira, and O. M. S. Contente, "Towards a reliable monitoring robot for mountain vineyards," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*. IEEE, 2015, pp. 37–43.

[7] S. Marden and M. Whitty, "Gps-free localisation and navigation of an unmanned ground vehicle for yield forecasting in a vineyard," in *Recent Advances in Agricultural Robotics, International workshop collocated with the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, 2014.

[8] F. N. Dos Santos, H. Sobreira, D. Campos, R. Morais, A. P. Moreira, and O. Contente, "Towards a reliable robot for steep slope vineyards monitoring," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3, pp. 429–444, 2016.

[9] T. Lowe, P. Moghadam, E. Edwards, and J. Williams, "Canopy density estimation in perennial horticulture crops using 3d spinning lidar slam," *Journal of Field Robotics*, 2020.

[10] A. K. Nellithimaru and G. A. Kantor, "Rols: Robust object-level slam for grape counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[11] R. Comelli, T. Pire, and E. Kofman, "Evaluation of visual slam algorithms on agricultural dataset," pp. 0–0, 2020.

[12] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1373–1378.

[13] F. A. Cheein, G. Steiner, G. P. Paina, and R. Carelli, "Optimized eif-slam algorithm for precision agriculture mapping based on stems detection," *Computers and electronics in agriculture*, vol. 78, no. 2, pp. 195–207, 2011.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[15] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[16] T. T. Santos, L. H. Bassoi, H. Oldoni, and R. L. Martins, "Automatic grape bunch detection in vineyards based on affordable 3d phenotyping using a consumer webcam." in *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*. In: CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 11., 2017, Campinas. Ciência de ..., 2017.

[17] F. Shu, P. Lesur, Y. Xie, A. Pagani, and D. Stricker, "Slam in the field: An evaluation of monocular mapping and localization on challenging dynamic agricultural environment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1761–1771.

[18] T. Pire, M. Mujica, J. Civera, and E. Kofman, "The rosario dataset: Multisensor data for localization and mapping in agricultural environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 633–641, 2019.

[19] J. Mendes, F. N. Dos Santos, N. Ferraz, P. Couto, and R. Morais, "Vine trunk detector for a reliable robot localization system," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2016, pp. 1–6.

[20] J. Zhang, S. Maeta, M. Bergerman, and S. Singh, "Mapping orchards for autonomous navigation," in *2014 Montreal, Quebec Canada July 13–July 16, 2014*. American Society of Agricultural and Biological Engineers, 2014, p. 1.

[21] F. Roure, G. Moreno, M. Soler, D. Faconti, D. Serrano, P. Astolfi, G. Bardaro, A. Gabrielli, L. Bascetta, and M. Matteucci, "Grape: Ground robot for vineyard monitoring and protection," in *Iberian Robotics Conference*. Springer, 2017, pp. 249–260.

[22] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2d mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 22–29.

[23] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[24] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, p. 105247, 2020.

[25] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] Y. Majeed, M. Karkee, Q. Zhang, L. Fu, and M. D. Whiting, "Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks," *Computers and Electronics in Agriculture*, vol. 171, p. 105308, 2020.

[29] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[30] E. Badeka, T. Kalampokas, E. Vrochidou, K. Tziridis, G. A. Papakostas, T. Pachidis, and V. G. Kaburlasos, "Real-time vineyard trunk detection for a grapes harvesting robot via deep learning," in *13th International Conference on Machine Vision (ICMV 2020)*, 2020 (Accepted).

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[32] G. Jocher, A. Stoken, and J. B. et.al., "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration," Jan. 2021.

[33] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[34] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1–1, 05 2016.

[35] P. Buayai, K. R. Saikaew, and X. Mao, "End-to-end automatic berry counting for table grape thinning," *IEEE Access*, vol. 9, pp. 4829–4842, 2021.

[36] A. S. Aguiar, N. N. Monteiro, F. N. d. Santos, E. J. Solteiro Pires, D. Silva, A. J. Sousa, and J. Boaventura-Cunha, "Bringing semantics to the vineyard: An approach on deep learning-based vine trunk detection," *Agriculture*, vol. 11, no. 2, 2021.

[37] A. S. Aguiar, F. N. D. Santos, A. J. M. De Sousa, P. M. Oliveira, and L. C. Santos, "Visual trunk detection using transfer learning and a deep learning-based coprocessor," *IEEE Access*, vol. 8, pp. 77 308–77 320, 2020.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[39] D. Longo, A. Pennisi, R. Bonsignore, G. Muscato, G. Schillaci, *et al.*, "A multifunctional tracked vehicle able to operate in vineyards using gps and laser range-finder technology," in *International Conference Ragusa SHWA2010-September 16-18 2010 Ragusa Ibla Campus-Italy" Work safety and risk prevention in agro-food and forest systems*, 2010.

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.

[41] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[42] I. Kostavelis, E. Boukas, L. Nalpantidis, and A. Gasteratos, "Stereo-based visual odometry for autonomous robot navigation," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 21, 2016.

[43] M. I. Lourakis and A. A. Argyros, "Sba: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 1, pp. 1–30, 2009.

[44] K. A. Tsintotas, P. Giannis, L. Bampis, and A. Gasteratos, "Appearance-based loop closure detection with scale-restrictive visual features," in *International Conference on Computer Vision Systems*. Springer, 2019, pp. 75–87.

[45] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*. Cambridge university press, 2013, vol. 53.

[46] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[47] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[48] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal on Computer Vision*, 2018.

[49] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.