Spatio-Temporal Consistent Semantic Mapping for Robotics Fruit Growth Monitoring

Luca Lobefaro[1] Matteo Sodano[1] Daniel Fusaro[2] Federico Magistri[1]
Meher V. R. Malladi[1] Tiziano Guadagnino[1] Alberto Pretto[2] Cyrill Stachniss[1,3]

***Abstract*—Automatic fruit growth monitoring plays a vital role in advancing precision agriculture. Tracking the evolution of fruits over time is essential to monitor their development and optimize production. The ability to recognize fruits over periods of time, even with drastic scene changes, is a required capability of agricultural robots. This paper presents a system that allows long-term fruit tracking in 3D data. It generates instance-segmented 3D representations of plants at various growth stages over time, utilizing only consumer-grade RGB-D cameras installed on a mobile robot. Our approach first performs instance segmentation on each image in a sequence. Then, by exploiting geometric information and depth maps, we track the same instances throughout the sequence. We produce a 3D point cloud containing instances, exploiting odometry information and 3D semantic mapping. Once our robot performs a new recording at a different plant growth stage, it associates each fruit with the previously built 3D cloud and update the model. We validate the system in a real-world glasshouse environment in Bonn, Germany. Experimental results demonstrate that our system outperforms existing baselines even though it relies only on annotated images and operates at frame-rate, allowing the deployment on a real robot.**

***Index Terms*—Robotics and Automation in Agriculture and Forestry, Mapping**

I. Introduction

THE world population is increasing, and we must increase food production. We need greater efficiency of agricultural production while reducing emissions. One approach toward this is the development and realization of robotic agriculture [20]. In this context, automatic phenotyping of fruits is a relevant topic, and often needs to be performed over time. This is only possible if we are able to recognize fruits in multiple measurements from the same session and among multiple data acquisitions, which is a challenging task. Another important aspect is using sensors easy to adapt to different situations and platforms and at low cost, to facilitate the adoption of such technologies. An example is RGB-D camera, which became popular with the Microsoft Kinect [8] and showed promising results in different fields [18], [27].

Fig. 1: Examples of 3D models generated with our method from two different sequences, showing consistent recognition of the same fruit despite variations in position, structure, and occlusion. The lines connect the same fruits in the two representations.

In this work, we tackle the problem of consistent spatio-temporal instance-segmented mapping of fruits in a glasshouse. Generating a 3D representation of plants with segmented instances poses a challenge when deriving the map from a sequence of images. This process requires us to produce for each frame instances that are consistent with the rest of the sequence. Maintaining consistency over time becomes even more challenging when generating a new representation from a sequence captured days later, the segmentation must handle both intra-sequence and inter-sequence information to ensure that the same instance is consistently recognized across frames and sequences.

The main contribution of this paper is a pipeline that generates spatially aligned 3D point clouds of plants from multiple sequences with temporal consistent fruit instance annotations,

Manuscript received: Apr 15, 2025; Revised: Jun 18, 2025; Accepted: Jul 14, 2025. This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under STA 1051/5-1 within the FOR 5351 (AID4Crops) and by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070405 (DigiForest),

[1]L. Lobefaro, M. Sodano, M. V. R. Malladi, F. Magistri, T. Guadagnino and C. Stachniss are with the Center for Robotics, University of Bonn, Germany. [2]D. Fusaro and A. Pretto are with the University of Padua, Italy. [3]C. Stachniss is additionally with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

Digital Object Identifier (DOI): see top of this page.

Fig. 2: Example of the dataset used (RGB only), where the point of view allows to visualize only the plants. Challenging lighting conditions and the repetitive structure of the environment make instance segmentation and temporal association difficult.

using consumer-grade RGB-D cameras installed on a mobile robot, while running at sensor frame rate. Additionally, our approach can (i) perform image instance segmentation of fruits with predictions consistent among all images in a sequence, i.e., tracking fruit instances within a sequence; (ii) recognize the same instances between different image sequences recorded at different times, even weeks apart, to produce a temporal consistent prediction. Fig. 1 depicts an example of the result. Our experimental evaluation back up these claims. The open-source implementation is available at: https://github.com/PRBonn/semantic-spatio-temporal-mapping

II. Related Work

Approaches to technologies-driven agriculture aim to use advanced technology to optimize resource use and productivity. In this context, automatic fruit monitoring plays a crucial role, making it possible to process great amounts of data in a short amount of time, with accurate results and greater precision if compared to manual labor [3].

For automatic fruit monitoring, we need to equip robots with methods to recognize individual fruit instances. In the scientific literature, there is a great number of works on instance segmentation and object detection using images, e.g. [11], [22]. Different works explore the application in the agricultural domain for fruits [7], [13] and plants detection [24], [29]. In this paper, we rely on Yolo [22] for localizing fruits in the RGB image, because it is suitable for robot adoption, given its ability to work at high frequencies and low memory requirements.

Fruit phenotyping requires us to generate detailed 3D models of plants. Several approaches in the literature aim to produce accurate 3D models of plants consistent in time. Xiang et al. [30] present a two-step approach for plant growth tracking. They focus on spatio-temporal registration of point cloud coming from periodic scan of the same plants. Heiwolt et al. [10] propose a method to encode leaf-shape to facilitate the recognition of the same plant organs at different growing stages, which are helpfull for 4D mapping of plants. Lobefaro et al. [15], Dong et al. [5], and Carlone et al. [2] propose methods for spatial-temporal mapping of growing plants, useful for phenotyping over time. Nevertheless, none can represent instance information about fruits in the models.

To obtain an instance-segmented 3D model, we need to predict consistent annotations between different images of the same sequence, i.e., performing intra-sequence fruit tracking. In this regard, Halstead et al. [7] propose a method to estimate instances consistently through a sequence of images for field-agnostic monitoring, with a focus on multi-tasking learning. Smitt et al. [26] go a step further and propose a method to produce a 3D consistent panoptic representation with consistent instances of sweet peppers. They employ a NeRF-based system to enable 3D panoptic scene understanding. Liu et al. [14] and Meyer et al. [19] use semantic 3D models for fruit counting using only monocular cameras, but they do not target temporal consistency.

For long-term monitoring, we need to track the same fruit across different time sequences [21]. This task presents several challenges: plants constantly change their shape, evolving in a non-predictable way. Across different explorations, fruits might be harvested, fall down, or change color, and some might even be emerged. These dynamic factors make the task particularly difficult. Many studies explore the problem of tracking plant evolution over time. Lobefaro et al. [16] address the data association problem between plant point clouds dealing with noise and dynamics. Magistri et al. [17] associate plant features over time to automatize phenotypic trait tracking. None of these approaches take semantics into account, and they are not able to operate in the wild, without intrusive measurements. Riccardi et al. [23] present a descriptor for temporal consistent fruit instance association, relying on point clouds with pre-integrated semantics. Fusaro et al. [6] also provide instance segmentation but still rely on pre-computed high-resolution point clouds. To the best of our knowledge, no studies provide a method for consistent segmentation between different image sequences to produce consistent 3D models with instance annotations.

In this paper, we propose a complete pipeline for fruit instance segmentation that is consistent both within a single sequence of images and across multiple sequences. We operate ”in the wild”, within a real glasshouse environment, without altering the setting or relying on intrusive methods. In addition, our approach can generate aligned 3D models with instance annotations from two distinctive sequences. It is robust to non-rigid plant evolution and significant changes in fruit structure and position, making it suitable for long-term, real-world monitoring applications.

III. Our Approach

For our approach to fruit growth monitoring over time we assume an RGB-D camera installed on a mobile robot. Starting from an RGB-D data stream, our system first produces a 3D point cloud from a single recording session containing annotations on the fruit instances. For each frame, we compute the associated pose, a pixel-wise instance segmentation on the image, and a point cloud given by the depth. We use this information for online mapping. When the robot moves through the same environment again some time later, with plants at a different growing stage, we need a time-aware instance segmentation to ensure that the predictions remain

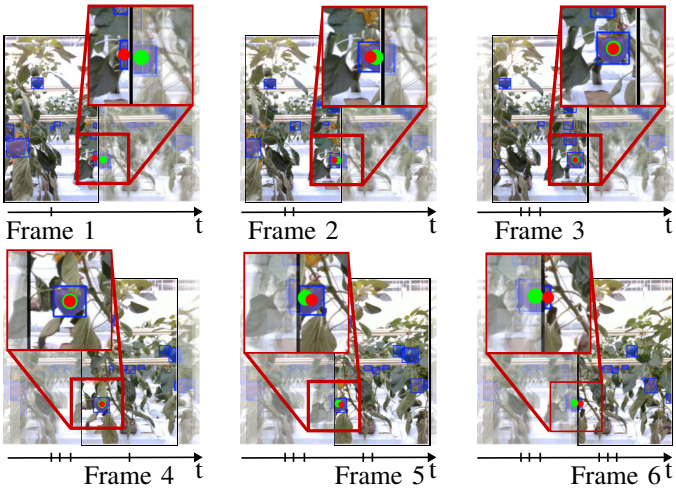

Fig. 3: On the top, the real center of the fruit (green circle) is initially hidden, causing the predicted centroid (red circle) to be inaccurate. As the fruit becomes more visible in frame 2, the predicted centroid aligns better with the true center. In frame 3, with the fruit fully visible, we can accurately predict its centroid. On the bottom, we observe a fruit near the left border. In frame 4, the fruit is still fully visible, but by frame 5, the predicted centroid no longer matches the real one. We update the centroid anyway to facilitate its recognition in frame 6, where its position is closer to the position in frame 5 than to the true center.

consistent with those from the earlier exploration. Additionally, we compute poses aligned with the previous map to facilitate associations and enable the creation of a 3D model aligned with the previous one. In this way, we can produce an updated version of the model containing instances associated with the previous 3D model. To develop the pipeline, we recorded data with a robotics platform designed for agriculture, in which vertically mounted RGB-D cameras can capture side-views of sweet pepper rows in a glasshouse. Fig. 2 shows an example of the input frames.

A. Pose Estimation

Consider the current sequence $\mathcal{S}_t = \{\mathcal{I}_t^1, \mathcal{I}_t^2, \ldots, \mathcal{I}_t^F\}$, where t indicates the time at which the sequence has been recorded and F the number of RGB-D images in the sequence. For each frame in $\mathcal{S}_t$, we compute the associated pose relative to the starting point. This information will serve as a basis for the next steps of the pipeline and for mapping. We rely only on the RGB-D images for motion estimation and do not require wheel odometry, although it can be easily integrated if available. In the absence of prior information as initial guess from another sensor, we use the constant velocity model, similarly to Vizzo et al. [28], as starting point of our motion estimation. For typical robot motion along rows in a glasshouse, this initial guess is sufficient, and thus we use it in our setup. For more complex motions, integrating wheel odometry or an IMU could provide a more accurate initialization and can be incorporated without major changes to the system. Starting from this initial guess, we refine the pose using point-to-point ICP [1] between the current cloud and the local map, computed processing RGB-D data. We generate the current cloud using the classic pinhole camera model [9] and the depth information. To create a point cloud focused only on the plants in the current glasshouse's row, we filter out all the points outside the relevant depth range.

B. Spatially Consistent Instance Segmentation

In parallel to pose estimation, our pipeline simultaneously computes instance information from the image data. We start the process by performing instance segmentation on each image using a Yolo [22] model trained specifically for sweet pepper instance segmentation. This model generates a mask for each image, with pixel-wise instance labels and a bounding box for each fruit. However, these predictions lack consistency across images because standard instance segmentation processes each frame independently, without optimizing instance IDs for tracking across frames. Thus, we refine the predictions, ensuring that the instance annotations are consistent across the sequence.

We start by computing the 3D position of each fruit's centroid for every detected instance. As previously mentioned, we are only interested in fruits from the current row. We want to address this since, from the camera stream, we get observations of rows behind. For each instance prediction, we obtain a segmentation mask and a bounding box. We calculate the average depth of each pixel in the mask and discard those instances with an average depth exceeding a predefined threshold. This threshold is determined based on prior knowledge of the depth range corresponding to the current plants [25]. After filtering out far away instances, we calculate the 3D position of the fruit's centroid using the pixel corresponding to the center of the bounding box and the corresponding depth value, using the pinhole camera model with the appropriate calibration parameters. This centroid represents a viewpoint-specific reference rather than the fruit's geometric center, enabling frame-to-frame association under minimal viewpoint changes. Errors, including occlusions, will not affect the result as similar errors occur in consecutive frames, ensuring correct associations.

Once we obtained the centroid for each instance, we can use it to to associate the corresponding fruit with fruits already seen along the sequence. We maintain a database $\mathcal{C}_{db} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$ of 3D coordinates of fruits' centroids seen so far. For each fruit centroid $\mathcal{C}_i^f$ in the current frame, we search for the nearest one in $\mathcal{C}_{db}$. We use nearest-neighbor search as it aligns well with the sequential nature of our data and the fact that our pipeline relies on pose information. However, our implementation is modular and supports alternative matching algorithms with the same output structure. In case we have an association between $\mathcal{C}_i^f$ and one of the centroids $\mathcal{C}_m \in \mathcal{C}_{db}$, we override the label of the mask corresponding to $\mathcal{C}_i^f$ with the label associated to $\mathcal{C}_m$. Then, we override the value of the centroid $\mathcal{C}_m$ with the one of $\mathcal{C}_i^f$. This final step is crucial, as illustrated in Fig. 3, when the detected fruit instance in the current frame is near the image's border. Due to the limited field of view, we cannot avoid including the instances on image's borders in the analysis. These provide essential information about the fruit's shape, which would be difficult to infer from the portion of the fruit visible when they are in the image's center. Our system deals with two edge cases as

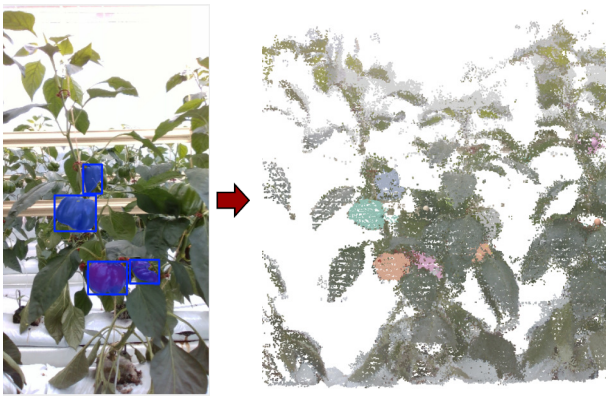

Fig. 4: Example of the instance segmented 3D model produced with our system. From a sequence of images like the one on the left, we can produce a voxel map as the one in the right. Points belonging to fruit instances are in different colors.

follows. The first one is when the fruit instance is on the right side (the fruit enters in the frame as the tobot moves forward) and partially visible. The predicted 3D centroid will not match the real center of a fruit. As we move to the next image, the fruit becomes more visible, and after we update the centroid, this will reflect its actual 3D position, because a larger area of it will be visible. In the second case the fruit instance is on the left side and partially visible (the fruit is leaving the frame as the robot moves forward). The predicted 3D centroid is also inaccurate. Since it will be less visible in the next frame, our update will shift the predicted centroid slightly to the right. This will facilitate the association in the next frame. Finally, if there is no centroid in $\mathcal{C}_{db}$ near enough to $\mathcal{C}_i^f$, we consider it as a new fruit, and we add a new element $\mathcal{C}_{M+1}$ to $\mathcal{C}_{db}$, using the value of $\mathcal{C}_i^f$.

C. Mapping

Once we compute the pose associated with the current frame $\mathcal{I}_t^f$ and the pixel-wise instance annotations, we integrate the point cloud extracted from the frame into the map.

As demonstrated in our previous work on consistent spatio-temporal mapping [15], we use a voxel map representation for our model. This improves performance while minimizing memory consumption for storing the 3D map. Additionally, we maintain a subsampled version of the voxel map that contains only local information, which is particularly effective for fast pose estimation (see Sec. III-A).

Using the unique ID computed for each pixel that represents a fruit, we integrate it into our voxel map. We maintain the associated label for each point and annotate each voxel with the more frequent label among the points it contains. Once we process the entire sequence $\mathcal{S}_t$, we obtain our first plants model, which we will call the reference model $\mathcal{M}_r$. Fig. 4 provides an example of a subsection of the voxel map produced, with distinct colors representing each instance.

D. Temporally-Consistent Pose Estimation

At this point in our approach, we have a 3D reference model $\mathcal{M}_r$ representing the fruits at a given point in time. Returning to model the same plants sometime later, we want to locate the same fruits, recognize them, and assign the same instance. This task is challenging because instance predictions lack temporal consistency, and both the plants and fruits undergo changes in shape and position over time.

Given the new sequence $\mathcal{S}_{t+1} = \{\mathcal{I}_{t+1}^1, \mathcal{I}_{t+1}^2, \dots, \mathcal{I}_{t+1}^Q\}$ measuring the same row in the glasshouse recorded in $\mathcal{S}_t$ and assuming it starts from the same origin of $\mathcal{S}_t$, we want to generate a second map that is aligned with the previous one. Performing frame-to-map registration of each frame on an outdated 3D representation is challenging. The plants changed their appearance between the two sessions. For this reason, following the idea developed in our previous work [16] we use the stable features of the environment as the only valid information for the odometry. In particular, we choose the plants' bases as reliable features, assuming that they remain stationary across successive explorations. These are extracted by applying a height threshold to the reference map, where the threshold is determined based on the observed height at which the plant stem is located. We then operate with the same methodology explained in Sec. III-A.

E. Temporally-Consistent Instance Segmentation

At this point, we need to predict instances on the current sequence consistent with the reference one. We exploit the pose computed on the previous step, aligned with the reference map $\mathcal{M}_r$, which provide information about which plant we are observing.

First, we apply the same instance filtering explained in Sec. III-B. This is essential to consider only instances of the current glasshouse's row. After filtering, we compute the 3D centroids of each fruit, that we use to match with the fruit centroids in the previous map. We maintain two databases: $\mathcal{C}_{curr} = \{\mathcal{C}_1^c, \mathcal{C}_2^c, \dots, \mathcal{C}_N^c\}$ that represent the list of 3D centroids of fruits in the current sequence; and $\mathcal{C}_{db} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_M\}$ that, instead, maintains the list of the 3D centroids seen in the previous map, to perform matching also with the previous sequence. Now, for each fruit's centroid $\mathcal{C}_i^f$ in the current frame, we first search for a match in $\mathcal{C}_{db}$, using the same method presented in Sec. III-B, but this time using a bigger threshold for the nearest neighbor search. This is because we expect that fruits moved more between the two recordings. If we have an association $\mathcal{C}_m \in \mathcal{C}_{db}$ in this step, we update the value of $\mathcal{C}_m$ with the value of $\mathcal{C}_i^f$. Since consecutive frames share similar viewpoints, updating the center with the current frame improves matching, especially under leaf occlusion. An occluded fruit's centroid often more closely resembles that of the previous frame than an average position computed over multiple past frames. Furthermore, it will correct the behavior analyzed in Fig. 3. If we do not have any fruit associated in $\mathcal{C}_{db}$ it means that we do not have the same fruit in the previous map. For this reason, we will perform the association on $\mathcal{C}_{curr}$ to obtain local consistency. If we do not have any association, we add the new centroid $\mathcal{C}_i^f$ to $\mathcal{C}_{curr}$, treating it as a new fruit.

Once we have the associations, we update the labels on the instance mask of the current frame, and we integrate the corresponding point cloud in our voxel map, accumulating

Reference	Current	Approach	Acc	F1
June20-row3	June22-row3	Riccardi [23]	66.67	**60.00**
		Fusaro [6]	-	-
		Ours	**71.43**	**60.00**
June20-row3	July07-row3	Riccardi [23]	33.33	40.0
		Fusaro [6]	**83.0**	0.0
		Ours	70.0	**66.67**
June20-row4	June22-row4	Riccardi [23]	37.04	10.53
		Fusaro [6]	-	-
		Ours	**38.89**	**21.43**
June22-row3	July07-row3	Riccardi [23]	25.0	0.0
		Fusaro [6]	**85.0**	0.0
		Ours	69.23	**66.67**

TABLE I: Accuracy and F1-Score for fruit matching on the 3D models. Better results are in bold. A dash (-) indicates failure.

the information on the instances. This is essential to have the correct labels during mapping.

F. Temporally-Consistent Mapping

For each frame $\mathcal{I}^f_{t+1}$ in the new sequence, we have a pose and pixel-wise instance annotations, together with the point cloud given by the depth associated with the frame. Following the same idea exposed in Sec. III-C, we integrate each frame into a new map, which we will call the current map $\mathcal{M}_c$. Because we computed the pose on the reference map $\mathcal{M}_r$, the new map $\mathcal{M}_c$ is aligned with $\mathcal{M}_r$, making it easy to make measurements on the plants at the different growing stages.

The final result is a second point cloud with the fruit instances annotated by point. These annotations are consistent with those in the reference map, allowing us to recognize the same fruit in the two explorations. Furthermore, the two clouds are aligned, making it easy to measure the fruit. Fig. 1 shows an example of the two clouds, with the same fruits connected by lines and represented with the same color.

IV. Experimental Evaluation

The main focus of this work is to propose a complete pipeline that generates instance-segmented 3D point clouds of plants at different growth stages, using only consumer-grade RGB-D cameras. We present our experiments to show the capabilities of our method. The results of our experiments showcase our main contribution: our method can produce spatially aligned 3D point clouds from different sequences with temporal consistent instances annotations of fruits, all while running at the sensor frame rate. Additionally, we conduct ablation studies to show that our method can (i) perform image instance segmentation of fruit with predictions consistent among all images in a sequence, (ii) recognize the same instances between different image sequences recorded at different times, even weeks apart, to produce a temporal consistent prediction.

A. Experimental Setup

We collected our data in a glasshouse, using the robotic platform presented by Smitt et al. [25]. The robot moves with a speed of approximately $0.2\,\text{m/s}$ between $34\,\text{m}$-long rows of growing sweet peppers. We used an Intel RealSense D435i RGB-D sensor to record data of plants over the course of two weeks. We show an example of the data captured in Fig. 2. In total, we recorded five sequences across three different days: June 20th, June 22nd, and July 7th, 2023, from two different rows of the glasshouse. For the evaluation we created a manually annotated dataset. For each sequence, we manually annotated every sweet pepper instance in each frame, ensuring consistent labeling of the same fruit across different images. Additionally, when annotating different sequences, we maintained consistent labels for the same fruit at various growth stages. Lastly, we used the annotated images to create a ground truth voxel map, where we converted the image labels into 3D points using our mapping system. This ground truth map serves as the basis for evaluating our instance segmentation method and the associations within the 3D models. We do not perform tests on pose estimation as odometry is not our contribution and no ground truth is available on the collected dataset. We use the consistency of the 3D model as an indicator of the quality of the poses since the model is the output of interest.

Sequence	CPU+GPU		Only CPU	
	Hz ↑	ms ↓	Hz ↑	ms ↓
June20-row3	17	61	9	112
June20-row4	19	54	10	105
June22-row3	19	55	10	110
June22-row4	20	51	10	108
July07-row3	19	55	9	116
Average	**19**	**55**	**10**	**110**

TABLE II: Execution time for instance segmentation, odometry and mapping on all the sequences, both for intra-sequence and inter-sequence modeling. The left column shows the experiments performed on the CPU with only Yolo running on the GPU while the right column shows the execution time using only the CPU.

B. 3D Instance Association

In this section, we evaluate the quality of the associations between two models at different growing stages directly on the point clouds. This experiment showcase our main contribution. We evaluate our system against two baselines. The first one, proposed by Fusaro et al. [6], performs instance segmentation of fruits directly on the point cloud. Then, it associates the fruits among two maps, again working only with point clouds. We trained the model on the pairs June20/June22, row3 and June20/June22, row4. We tested the model on the other two pairs of sequences. The second baseline, proposed by Riccardi et al. [23], computes a descriptor for each instance and then uses the Hungarian algorithm [12] to match corresponding fruits. We use the parameter setting suggested in the original implementation. Because this approach relies on instance segmented point clouds to perform the matching, we used as input the instance segmented map produced by our system. Since we use standard approaches to build the point cloud, we do not evaluate its quality for downstream tasks such as 3D segmentation or mapping. Our work focuses primarily on instance association.

The predicted instances do not correspond to the ground truth instances, so the evaluation is carried out using a IoU

Sequence	Approach	Detection				Association			
		Acc [%] ↑	Prec [%] ↑	Rec [%] ↑	F1 [%] ↑	Rec [%] ↑	F1 [%] ↑	TP ↑	FN ↓
June20-row3	Centers2d	48.17	93.44	**49.85**	65.02	87.85	93.53	282	39
	IoU	46.76	93.26	48.40	63.72	87.78	93.49	273	38
	Matches	46.34	93.20	47.96	63.33	74.67	85.50	230	78
	Ours	**48.37**	**94.21**	**49.85**	**65.20**	**96.57**	**98.26**	**310**	**11**
June20-row4	Centers2d	**45.73**	92.58	**47.47**	**62.76**	83.31	90.90	774	155
	IoU	45.43	92.79	47.09	62.47	84.80	91.77	781	140
	Matches	44.25	92.35	45.93	61.35	74.47	85.37	668	229
	Ours	44.70	**92.94**	46.27	61.78	**92.92**	**96.33**	**840**	**64**
June22-row3	Centers2d	47.03	89.35	**49.83**	63.98	84.31	91.49	344	64
	IoU	**47.08**	89.90	49.71	**64.02**	83.78	91.18	341	66
	Matches	46.43	89.41	49.13	63.41	65.67	79.28	264	138
	Ours	46.29	**90.48**	48.66	63.29	**99.50**	**99.75**	**396**	**2**
June22-row4	Centers2d	48.06	90.85	**50.50**	64.92	83.65	91.10	1090	213
	IoU	47.10	90.92	49.43	64.04	83.20	90.83	1060	214
	Matches	45.47	90.63	47.72	62.52	75.26	85.89	925	304
	Ours	**48.11**	**91.39**	50.39	**64.96**	**94.54**	**97.20**	**1230**	**71**
July07-row3	Centers2d	51.22	84.53	**56.52**	**67.75**	89.17	94.27	321	39
	IoU	50.89	**84.97**	55.92	67.45	89.61	94.52	319	37
	Matches	50.48	84.51	55.62	67.09	78.25	87.80	277	77
	Ours	**51.23**	84.88	56.37	**67.75**	**95.82**	**97.87**	**344**	**15**

TABLE III: Ablation study for consistent fruit instance segmentation in a sequence of images. In the column Detection we show the quality of the results of our pre-trained Yolo model. In the column Association we present the quality of the associations between consecutive frames. For each sequence the row indicates the glasshouse row. Best results are outlined in bold.

threshold of 25% for associating the predicted instances ID with the ground truth instances ID. We evaluate the systems with two metrics: F1-score and Accuracy. In particular, to keep into account also true negative matches (fruit correctly not associated), we compute the accuracy value as:

$$\text{Accuracy}_{\text{match}} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{FP} + \text{FN})} \tag{1}$$

Our approach does not operate directly on the point clouds like the other two methods. Instead, it performs instance segmentation on the images while simultaneously generating the 3D model at the sensor's frame rate. As a result, it relies only on local information for its predictions and does not exploit the data from the entire map. Moreover, it does not require training on annotated 3D point clouds but instead relies on the 2D instance segmentation module given by Yolo. Despite this, as we show in Tab. I, our system outperforms the baselines in almost all cases, particularly in F1-score, where, the other methods sometimes fail due to lack of true positives. This is especially true for Fusaro et al. [6], as insufficient training data limited the ability to learn accurate matches, a common constraint of deep learning methods. Riccardi et al. [23], instead, relies only on the 3D point cloud, making it highly dependent on its quality for extracting key features like fruit center and size.

C. Execution Time

To provide evidence that our system can produce the 3D model at frame frequency, Tab. II reports the execution time of our pipeline. We evaluated our system on an Intel Core i9-10980XE CPU with the Yolo model running on a Nvidia RTX A4000 GPU. Additionally, in the same table, we include the results of running also the Yolo instance segmentation model on the CPU. We report the number resulting by averaging the execution times on all the sequences. The results indicate that our model can generate a 3D model with temporal-consistent annotated instances at a rate of 19 Hz on the CPU with Yolo running on the GPU and at 10 Hz when the entire process runs on the CPU.

D. Ablation Study

In this section we conduct ablation studies to showcase the additional contributions of our work.

Intra-Sequence Instance Segmentation Evaluation. The first ablation study analyzes how our system can produce consistent labels within a single sequence and real world glasshouse conditions. Specifically, we tested three methods: (i) Centers2d: this method associates two instances across consecutive frames by finding the nearest centers within the 2D bounding boxes provided by Yolo. (ii) IoU: this approach associates instances by selecting those with the highest intersection over union of their masks. (iii) Matches: this method uses SuperPoint descriptors [4] to compute point-to-point matches in the images. Two instances are then associated based on the number of matching points within the corresponding bounding boxes. The method also influences the detection because it impacts how we filter wrong instances. Then, we compute F1-score, accuracy and recall. For the detection, we define the values as follows: true positive indicates a correctly detected fruit, false negative indicates a fruit in the ground truth but not detected, and false positive indicates a detected fruit not present in the ground truth. For associations between consecutive frames, instead, we consider a match as true positive if the same label is correctly assigned to the same fruit and false negative if a different label is assigned to the same fruit between two frames. False positive is always zero, as it would represent a fruit detected but not in the ground truth, which is treated as a detection error. We show, in Tab. III, that our method

Reference	Current	Approach	Detection		Association			
			Acc/Rec [%] ↑	F1 [%] ↑	Rec [%] ↑	F1 [%] ↑	TP ↑	FN ↓
June20-row3	June22-row3	Centers2d	**53.90**	**70.05**	44.91	61.99	181	222
		IoU	53.20	69.45	51.88	68.32	207	192
		Matches	52.27	68.65	41.43	58.59	162	229
		Ours	53.32	69.55	**86.57**	**92.80**	**348**	**54**
June20-row3	July07-row3	Centers2d	**63.57**	**77.78**	17.20	29.35	59	284
		IoU	62.37	76.82	33.53	50.22	113	224
		Matches	63.42	77.62	12.90	22.86	44	297
		Ours	56.82	72.57	**69.67**	**82.12**	**209**	**91**
June20-row4	June22-row4	Centers2d	**54.10**	**70.21**	3.32	6.43	41	1194
		IoU	52.84	69.14	5.94	11.21	72	1141
		Matches	52.73	69.05	4.68	8.94	56	1141
		Ours	51.50	67.99	**15.11**	**26.26**	**177**	**994**
June22-row3	July07-row3	Centers2d	62.97	77.28	26.95	42.45	90	244
		IoU	62.37	76.82	37.16	54.19	123	208
		Matches	**63.12**	**77.39**	8.43	15.56	28	304
		Ours	58.32	73.67	**62.91**	**77.24**	**190**	**112**

TABLE IV: Ablation study for consistent fruit instance segmentation among different sequences. In the column "Detection" we show the quality of the results of our pre-trained Yolo model on the current sequence. In the column "Association" we present the quality of the associations between the reference and the current sequence. Best results are in bold.

always outperforms the other methods in the ablation study for the association task, even if it does not always achieve the highest accuracy in detection.

Inter-Sequence Instance Segmentation Evaluation. The second ablation study evaluates the ability of our system to produce instance labels consistent among different sequences and the results support our second claim. We perform a similar experiment of the previous section, to test how methods from the most naive to the most complex one, improves the results in temporal instance tracking.

Because the labels predicted by our approach in the reference and the current sequence do not correspond, we first compute a map between each label and the corresponding ground truth label. We associate each predicted fruit with the most overlapping ground truth fruit on the map. Then, we use these mappings both in the reference and current frame to compute our metrics. In particular, we consider true positive (TP) the associated instances with the same ground truth label. We consider false negative (FN) all instances that have not been matched but have the same ground truth label associated with them. In Tab. IV, we report the recall and the F1-score computed on all the methods. Our approach always performs better in the association step. The sequence on the 4th row of the glasshouse is challenging due to severe occlusion, a higher number of fruits and, as shown in Tab. V, higher distance between fruits across time. This reflects the lower performances of all methods. Finally, we evaluate the robustness of our pipeline to depth noise in Tab. VI. We add Gaussian noise with increasing standard deviation to each pixel and randomly invalidate depth pixels with growing probability. The results show stable performance under moderate noise (1 cm), while it degradates over 5 cm.

E. 3D Model Evaluation for Fruit Monitoring

In this section, we quantitatively evaluate the quality of the 3D model for the integration into a system for fruit monitoring. Our system is not designed as a phenotyping framework, but rather designed to enable algorithms that tackle such tasks, producing results that are preliminary to methods such as shape completion pipelines. We demonstrate its applicability by integrating it with a shape completion module [18]. Specifically, we extract each fruit instance from our 3D model and apply the shape completion module presented by Magistri et al. [18] to generate a corresponding mesh. Fig. 5 shows an example of this pipeline applied to one instance from our model. This demonstrates that the instances of our model can be easily extracted to be integrated into pipelines for fruit growth monitoring. The temporal consistency of our approach enables tracking fruit growth over time.

Reference	Current	Distance [cm]
June20-row3	June22-row3	2.95 ± 2.77
June20-row3	July07-row3	4.44 ± 2.15
June20-row4	June22-row4	17.64 ± 13.47
June22-row3	July07-row3	4.88 ± 3.19

TABLE V: Average distance between fruits in time for each sequence.

V. CONCLUSION

In this paper, we presented a pipeline that generates instance-segmented 3D point clouds of plants at various growth stages, enabling fruit tracking over time with a mobile robot. Our approach processes image sequences to obtain consistent fruit instance segmentation simultaneously generating a 3D point cloud with instances. Using this as a reference, we can generate an aligned 3D model from new recordings at different growth stages, with consistent instance recognition. We evaluated our method on a real-world dataset, compared it with existing techniques, and supported our claims through experiments. Our system, which requires no annotated point clouds and runs at sensor frame rate, outperforms prior methods. However, excessive fruit displacement over time can reduce the accuracy. Future work could improve robustness to positional shifts, evaluate performance on smaller, denser fruits, and test robustness to higher speeds by replacing nearest neighbor association with neural fruit descriptors.

Ref.	Curr.	σ [cm]	Inv. [%]	Recall ↑	F1 ↑
June20 (row3)	June22 (row3)	1	5	87.10	93.11
		5	10	32.41	48.95
		10	30	25.77	40.98
June20 (row3)	July07 (row3)	1	5	55.25	71.18
		5	10	33.00	49.62
		10	30	11.79	21.09
June20 (row4)	June22 (row4)	1	5	15.83	27.34
		5	10	7.80	14.48
		10	30	5.10	9.70
June22 (row3)	July07 (row3)	1	5	63.70	77.82
		5	10	24.25	39.04
		10	30	19.63	32.81

TABLE VI: Effect of synthetic noise on association quality. Here, σ denotes Gaussian noise standard deviation added to each depth pixel (cm); Inv. is the probability of treating a pixel as invalid. Results are averaged over 5 runs with different random noise samples. Noise-free results are reported in the "Association" column of Tab. IV.

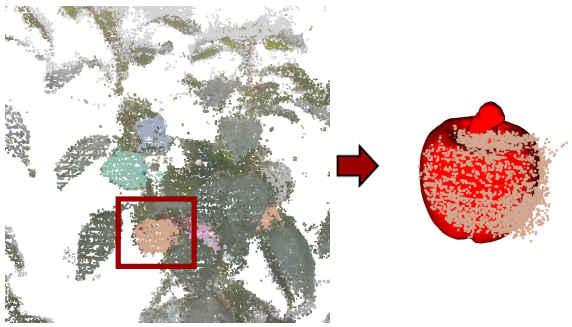

Fig. 5: Example on how our 3D model can be integrated with a shape completion module [18].

REFERENCES

[1] P. Besl and N. McKay. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[2] L. Carlone, J. Dong, S. Fenu, G. Rains, and F. Dellaert. Towards 4d crop analysis in precision agriculture: Estimating plant height and crown radius over time via expectation-maximization. In *Proc. of the ICRA Workshop on Robotics in Agriculture*, 2015.

[3] T. Chen and H. Yin. Camera-based Plant Growth Monitoring for Automated Plant Cultivation with Controlled Environment Agriculture. *Smart Agricultural Technology*, 8:100449, 2024.

[4] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] J. Dong, J. Burnham, B. Boots, G. Rains, and F. Dellaert. 4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[6] D. Fusaro, F. Magistri, J. Behley, A. Pretto, and C. Stachniss. Horticultural Temporal Fruit Monitoring via 3D Instance Segmentation and Re-Identification using Point Clouds. *arXiv preprint*, arXiv:2411.07799, 2024.

[7] M. Halstead, S. Denman, C. Fookes, and C. McCool. Fruit detection in the wild: The impact of varying conditions and cultivar. In *Proc. of Digital Image Comp.: Techniques and Applications (DICTA)*, 2020.

[8] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Trans. on Cybernetics*, 43(5):1318–1334, 2013.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[10] K. Heiwolt, C. Öztireli, and G. Cielniak. Statistical shape representations for temporal registration of plant components in 3D. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.Y. Lo, et al. Segment Anything. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[12] H. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[13] X. Liu, D. Zhao, W. Jia, W. Ji, C. Ruan, and Y. Sun. Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation. *IEEE Access*, 7:139635–139642, 2019.

[14] X. Liu, S.W. Chen, C. Liu, S.S. Shivakumar, J. Das, C.J. Taylor, J. Underwood, and V. Kumar. Monocular camera based fruit counting and mapping with semantic data association. *IEEE Robotics and Automation Letters (RA-L)*, 4(3):2296–2303, 2019.

[15] L. Lobefaro, M. Malladi, T. Guadagnino, and C. Stachniss. Spatio-Temporal Consistent Mapping of Growing Plants for Agricultural Robots in the Wild. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[16] L. Lobefaro, M. Malladi, O. Vysotska, T. Guadagnino, and C. Stachniss. Estimating 4D Data Associations Towards Spatial-Temporal Mapping of Growing Plants for Agricultural Robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[17] F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-Based 4D Registration of Plants Point Clouds for Phenotyping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[18] F. Magistri, E. Marks, S. Nagulavancha, I. Vizzo, T. Läbe, J. Behley, M. Halstead, C. McCool, and C. Stachniss. Contrastive 3D Shape Completion and Reconstruction for Agricultural Robots using RGB-D Frames. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):10120–10127, 2022.

[19] L. Meyer, A. Gilson, U. Schmid, and M. Stamminger. FruitNeRF: A Unified Neural Radiance Field based Fruit Counting Framework. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[20] L.F.P. Oliveira, A.P. Moreira, and M.F. Silva. Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. *Robotics*, 10(2), 2021.

[21] F.J. Pierce and P. Nowak. Aspects of Precision Agriculture. In *Advances in Agronomy*, volume 67, pages 1–85. Academic Press, 1999.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] A. Riccardi, S. Kelly, E. Marks, F. Magistri, T. Guadagnino, J. Behley, M. Bennewitz, and C. Stachniss. Fruit Tracking Over Time Using High-Precision Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[24] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss. Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[25] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool. PATHoBot: A robot for glasshouse crop phenotyping and intervention. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[26] C. Smitt, M. Halstead, P. Zimmer, T. Laebe, E. Guclu, C. Stachniss, and C. McCool. Pag-nerf: Towards fast and efficient end-to-end panoptic 3d representations for agricultural robotics. *arXiv preprint arXiv:2309.05339*, 2023.

[27] M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss. Robust Double-Encoder Network for RGB-D Panoptic Segmentation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[28] 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 Robotics and Automation Letters (RA-L)*, 8(2):1029–1036, 2023.

[29] J. Weyler, F. Magistri, E. Marks, Y.L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley. Phenobench: A large dataset and benchmarks for semantic image interpretation in the agricultural domain. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[30] S. Xiang and D. Li. Research on Plant Growth Tracking Based on Point Cloud Segmentation and Registration. In *Proc. of the Intl. Conf. on Image Processing, Computer Vision and Machine Learning (ICICML)*, 2022.

CERTIFICATE OF REPRODUCIBILITY

The authors of this publication declare that:

1) The software related to this publication is distributed in the hope that it will be useful, support open research, and simplify the reproducability of the results but it comes without any warranty and without even the implied warranty of merchantability or fitness for a particular purpose.

2) *Luca Lobefaro* primarily developed the implementation related to this paper. This was done on Ubuntu 22.04.

3) *Meher V.R. Malladi* verified that the code can be executed on a machine that follows the software specification given in the Git repository available at:

   https://github.com/PRBonn/semantic-spatio-temporal-mapping.git

4) *Meher V.R. Malladi* verified that the experimental results presented in this publication can be reproduced using the implementation used at submission, which is labeled with a tag in the Git repository and can be retrieved using the command:

   ```
   git checkout ral2025
   ```